```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> ## ------------------------------------------------------------------------
> ##
> ## Regress ninka-dantai outcomes on raid damage
> ## Note:
> ##
> ## ------------------------------------------------------------------------
> ## environment setting
> Sys.setenv(LANGUAGE="en")
> gc();gc()
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 2427401 129.7    4087080 218.3  4087080 218.3
Vcells 4037173  30.9    8388608  64.0  7069351  54.0
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 2427356 129.7    4087080 218.3  4087080 218.3
Vcells 4037116  30.9    8388608  64.0  7069351  54.0
> rm(list = ls())
> options(scipen = 999)   ## Disable scientific notation
>
> ## ------------------------------------------------------------------------
> ##
> ## Initial settings
> ##
> ## ------------------------------------------------------------------------
> ## Set directory
```

```
> ## ---------------------------------------------------------------------
> ## Load packages and functions
> root_dir = "E:/Dropbox/429_MDA_replication/Replication"
> source(file.path(root_dir, "2_Code/1_PackagesFunctions.R"))
> ## ---------------------------------------------------------------------
> ## Data directory
> data_dir = root_dir      %>%    file.path("1_Data") %>%    dir_create()
> ## Output directory
> output_dir = root_dir    %>%    file.path("3_Result")      %>%    dir_create()
> figure_dir = root_dir    %>%    file.path("4_Figures")     %>%    dir_create()
> ## Working directory
> working_dir = root_dir   %>%    file.path("5_Working")     %>%    dir_create()
> ## ---------------------------------------------------------------------
>
>
> ## ---------------------------------------------------------------------
> ## Load data
> ## ---------------------------------------------------------------------
> ## --- Census data
> tokyo_census_tbl = data_dir     %>%    file.path("census_data4.csv")     %>%
+   ##census_data4
+   read_csv()       %>%
+   filter(year == 2015) %>%
+   mutate(key_code = as.character(key_code))
Rows: 17977 Columns: 68
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (10): zipcode, rating, min_nghb_rating, max_nghb_rating, prewar_district, group, ResidRatio...
dbl (57): key_code, year, pop, r_kyodojutaku6up, r_unemp, rm_unemp, rf_unemp, r_proexe, rmale00...
lgl  (1): HazardAvailable

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
>
> ## --- All years
> tokyo_ninka_tbl = data_dir      %>%    file.path("test_ninka_2.dta")     %>%    read_dta()    %>%
+   rename(
+     dummy_ninka = d_ninka,
+     ln_ninka_days = lnninka_days
+   )
> ## --- Since 1993
> tokyo_ninka_tbl_1993 = data_dir %>%    file.path("test_ninka_sinceFY1993_2.dta")      %>%    read_dta()    %>%
+   rename(
```

```
+    dummy_ninka = d_ninka,
+    ln_ninka_days = lnninka_days
+  )
> ## --- Sports
> # sports_tbl = root_dir  %>%    file.path("@@SUBMISSION/JOP_RR1/supporting_materials/R2_Q5a/sports_club_list.csv")
      %>%
> sports_tbl = data_dir    %>%    file.path("sports_club_list.csv") %>%
+  read_csv(col_types = cols())  %>%
+  rename(
+    club_name = 1, cityName = ku_name,
+    chochoazaName = address_jp) %>%
+  mutate(
+    chochoazaName = if_else(
+      str_detect(chochoazaName, pattern = "¥¥d"),
+      true = str_c(chochoazaName, "丁目"),
+      false = chochoazaName),
+    chochoazaName = stringi::stri_trans_general(chochoazaName, id = "Halfwidth-Fullwidth"),
+    SportClubDummy = 1,
+    zipcode = str_remove_all(zipcode, pattern = "-")
+  )
>
> ## ----------------------------------------------------------------------
> ##
> ## Prepare regression-related objects
> ##
> ## ----------------------------------------------------------------------
> ## Link function
> binom_link = binomial(link = logit)
> count_link = quasipoisson
> ## ----------------------------------------------------------------------
> ## Outcome and treatment variables
> ## ----------------------------------------------------------------------
> depvar_vec_combined = c("ninka_days", "ln_ninka_days", "dummy_ninka", "n_ninka")
> placebo_depvar_vec = c("SportClubDummy", "LocalSportClubDummy")
> treatment_vec = c("ratio_damage", "ln_damage_ratio", "binary_damage_ratio")
> robust_treatment_vec = c(treatment_vec[2:3], str_c("as.factor(", treatment_vec[1],")"))
> ## ----------------------------------------------------------------------
> ## Covariates
> ## ----------------------------------------------------------------------
> ## District FE
> fe_term = "DistrictFE"
> ## Geographical features
> spatial_covariates = c(
```

```
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway and train stations
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## -----------------------------------------------------------------------
> ## Location variables
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> polynomial_degree = 5
> ## --- Polynomials and splines (single dimension)
> distance_term_specification = list(
+   ## --- Polynomials
+   str_c("poly(", distance_termz_base, ", degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   str_c("s(", distance_termz_base, ")"),
+   ## --- none (comparison purposes)
+   "",
+   ## --- 3rd order polynomials
+   str_c("poly(", distance_termz_base, ", degree = 3, simple = TRUE, raw = TRUE)")
+ )
> ## --- Lon-lat (two-dimensional) term
> spatial_term_specification = c(
+   ## --- Polynomials
+   str_c("poly(z_lon, z_lat, degree = ",  polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   "te(longitude, latitude)",
+   ## --- none (comparison purposes)
+   "",
+   ## --- 3rd order polynomials
+   "poly(z_lon, z_lat, degree = 3, simple = TRUE, raw = TRUE)"
+ )
> ## -----------------------------------------------------------------------
> ## Extra (two-dimensional) polynomials
> extra_covariates_specification = c(
+   ## --- none (baseline)
+   "",
+   ## --- with extra terms (robustness check)
+   str_c("poly(ratio_residential, PopDensity_1939_km2_ln, degree = ", polynomial_degree, ", simple = TRUE, raw = FALSE)"))
> ## -----------------------------------------------------------------------
```

```
>
> ## ------------------------------------------------------------------------
> ## Combine the post-1993 records and full records
> ## ------------------------------------------------------------------------
> depvar_1993_tbl = tokyo_ninka_tbl_1993      %>%
+   select(
+     key_code, chochoazaName, all_of(depvar_vec_combined)
+   )  %>%
+   rename_at(vars(ninka_days:n_ninka), ~ str_c(., "_1993"))
> ## ------------------------------------------------------------------------
> tmp_combined_tbl = tokyo_census_tbl   %>%
+   left_join(
+     tokyo_ninka_tbl      %>%
+       select(key_code, chochoazaName, all_of(depvar_vec_combined))
+   )  %>%
+   left_join(depvar_1993_tbl, by = c("key_code", "chochoazaName"))%>%
+   left_join(sports_tbl, by = c("cityName", "chochoazaName", "zipcode")) %>%
+   # left_join(full_poly, by = "key_code")    %>%
+   rownames_to_column(var = "row_count")      %>%
+   mutate(
+     DistrictFE = as.factor(prewar_district),
+     SportClubDummy = if_else(is.na(SportClubDummy), true = 0, false = SportClubDummy)
+   )  %>%
+   select(row_count, everything())
Joining, by = c("key_code", "chochoazaName")
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Spatially-lagged variables
> ## ------------------------------------------------------------------------
> full_poly = data_dir     %>%    file.path("RaidShp_May2020b.rds")%>%
+   read_rds()       %>%
+   filter(row_id!=88) %>% ## drop palace polygon
+   select(key_code, ratio_damage)       %>%
+   left_join(tmp_combined_tbl    %>%    select(key_code, SportClubDummy))%>%
+   mutate(
+     SportClubDummy = if_else(is.na(SportClubDummy), true = 0, false = SportClubDummy)
+   )
Joining, by = "key_code"
> ## Use different SWM
> full_contig_nb = poly2nb(full_poly, queen = TRUE)
> full_contig_listw = nb2listw(full_contig_nb, style = "W", zero.policy = TRUE) ## 1 neighborhood without neighbors
> ## Lagged damage SWM
```

```
> full_contig_nb_damage = poly2nb(full_poly    %>%    drop_na(ratio_damage), queen = TRUE)
> full_contig_listw_damage = nb2listw(full_contig_nb_damage, style = "W", zero.policy = TRUE)##   1   neighborhood   without
neighbors
> # lapply(full_contig_listw_damage$weights, length)  %>%    unlist()      ## N neighbor cells
> ## ------------------------------------------------------------------------
> ## Lagged damage
> lagged_damage_tbl = full_poly   %>%    drop_na(ratio_damage)      %>%
+   mutate(
+     LaggedDamage = lag.listw(full_contig_listw_damage, var = full_poly   %>%    drop_na(ratio_damage)      %>%
        pull(ratio_damage),
+                              zero.policy = TRUE)
+   )   %>%
+   as_tibble()      %>%
+   select(key_code, LaggedDamage)
> ## Construct lagged outcome
> lagged_variables_tbl = full_poly      %>%
+   mutate(
+     LocalSportClubDummy = lag.listw(full_contig_listw, var = full_poly$SportClubDummy, zero.policy = TRUE, NAOK = TRUE),
+     LocalSportClubDummy = if_else(SportClubDummy + LocalSportClubDummy > 0, true = 1, false = 0)
+   )   %>%
+   as_tibble()      %>%
+   select(key_code, SportClubDummy, LocalSportClubDummy)    %>%
+   left_join(lagged_damage_tbl, by = "key_code")
> lagged_variables_tbl     %>%    summary()
   key_code         SportClubDummy LocalSportClubDummy  LaggedDamage
 Length:2286        Min.   :0     Min.   :0          Min.   :0.0000
 Class :character   1st Qu.:0     1st Qu.:0          1st Qu.:0.1839
 Mode  :character   Median :0     Median :0          Median :0.5143
                    Mean   :0     Mean   :0          Mean   :0.4816
                    3rd Qu.:0     3rd Qu.:0          3rd Qu.:0.7667
                    Max.   :0     Max.   :0          Max.   :1.0000
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Combine and save
> ## ------------------------------------------------------------------------
> combined_tbl = tmp_combined_tbl %>%    left_join(lagged_variables_tbl)   %>%
+   mutate(
+     zipcode = str_c(str_sub(zipcode, start = 1, end = 3), "-", str_sub(zipcode, start = 4, end = 7)),
+     zipcode = if_else(zipcode == "962-0838", true = "173-0027", false = zipcode)
+   )   %>%
+   select(
+     row_id, row_count:pop, zipcode, -year,
```

```
+    all_of(depvar_vec_combined), contains("_1993"),
+    all_of(placebo_depvar_vec), all_of(treatment_vec), LaggedDamage,
+    all_of(fe_term), all_of(spatial_covariates), all_of(distance_termz_base),
+    z_lon, z_lat, longitude, latitude,
+    palace_dist)   %>%
+  drop_na(ratio_damage)  %>%
+  write_csv(file = data_dir     %>%    file.path("ANA_data.csv"))
Joining, by = c("key_code", "SportClubDummy")
>
> ## Read zipcode-level data
> zipcode_tbl = data_dir   %>%    file.path("ANA_zipcode_data.csv") %>%    read_csv(col_types = cols())
>
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ## Baseline formula objects
> ## ------------------------------------------------------------------------
> mdl_lst = list(
+  ## NOT IN USE, Polynomial: lm()-compatible specification
+  polynomial_base = chr2fml(
+    "outcome",
+    idv_list = list(
+      treatment_vec[1],
+      spatial_covariates, fe_term,
+      distance_term_specification[[1]],
+      spatial_term_specification[1],
+      extra_covariates_specification[1]## none
+    )
+  ),
+  ## Polynomial: felm()-compatible specification
+  polynomial_base_felm = chr2fml_felm(
+    "outcome",
+    idv_list = list(
+      treatment_vec[1],
+      spatial_covariates,
+      distance_term_specification[[1]],
+      spatial_term_specification[1],
+      extra_covariates_specification[1]## none
+    ),
+    fe_list = fe_term,
+    se_cluster = "row_count"    ## Robust SE
```

```
+     # se_cluster = fe_term      ## Cluster SE
+   ),
+   ## Spline: gam()-compatible specification
+   gam_base = chr2fml(
+     "outcome",
+     idv_list = list(
+       treatment_vec[1],
+       spatial_covariates, fe_term,
+       distance_term_specification[[2]],
+       spatial_term_specification[2],
+       extra_covariates_specification[1]## none
+     )
+   ),
+   ## gam with spatial 5th-order poly (for tab a9)
+   gam_sp5th = chr2fml(
+     "outcome",
+     idv_list = list(
+       treatment_vec[1],
+       spatial_covariates, fe_term,
+       distance_term_specification[[1]],
+       spatial_term_specification[1],
+       extra_covariates_specification[1]## none
+     )
+   )
+ )
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table 1
> ## Result I: Baseline regressions
> ##
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ----------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
```

```
+    felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]      %>%    update(str_c(depvar_vec_combined[3], " ~ ."))  %>%
+    gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%    update(str_c(depvar_vec_combined[3], "_1993 ~ ."))    %>%
+    gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]     %>%    update(str_c(depvar_vec_combined[2], " ~ ."))  %>%
+    # update(str_c(depvar_vec_combined[1], " ~ ."))  %>%
+    gam(data = combined_tbl),
+   ## ------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%    update(str_c(depvar_vec_combined[4], " ~ ."))  %>%
+    gam(data = combined_tbl, family = count_link)
+ )
>
> ## ------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------
> ## Table 1: Panel A
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                        Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm         GAM         GAM           GAM     gam: quasipoisson
                                (logistic)   (logistic)   (continuous)    link = log
                       (1)         (2)         (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage          -0.088**     -0.578**     -0.537**      -0.859**      -0.433***
                     (0.040)      (0.241)      (0.245)       (0.337)       (0.135)

---------------------------------------------------------------------------------------------
Observations            2,152        2,152        2,099         2,152         2,152
```

```
R2                      0.253
Adjusted R2             0.225        0.241        0.234          0.230          0.270
Log Likelihood                    -1,113.950    -1,070.932    -5,798.659
UBRE                              0.035        0.020         12.829          0.692
Residual Std. Error 0.411 (df = 2075)
=================================================================================================
Note:                                                          *p<0.1; **p<0.05; ***p<0.01
>
> ## ------------------------------------------------------------------------
> ## Panel B: Dummy regressions
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst_dummy = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(
+         depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3]
+       )      %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]       %>%
+     update(
+       str_c(
+         depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3]
+       )
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(
```

```
+      str_c(
+        depvar_vec_combined[2], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3]
+      )
+    ) %>%
+    # update(str_c(depvar_vec_combined[1], " ~ .")) %>%
+    gam(data = combined_tbl),
+   ## ----------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+    update(
+      str_c(
+        depvar_vec_combined[4], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3]
+      )
+    ) %>%
+    gam(data = combined_tbl, family = count_link)
+   ## ----------------------------------------------------------------------
+ )
> ## ---------------------------------------------------------------------
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> ## Table 1: Panel B
> stargazer(
+   rslt_lst_dummy,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )

=====================================================================================================
                                                Dependent variable:
                        ---------------------------------------------------------------------------
                              dummy_ninka         dummy_ninka_1993 ln_ninka_days     n_ninka
                             felm         GAM          GAM           GAM     gam: quasipoisson
                                       (logistic)    (logistic)    (continuous)    link = log
                              (1)          (2)          (3)           (4)           (5)
-----------------------------------------------------------------------------------------------------
as.factor(ratio_damage)0.1     -0.029        -0.306       -0.350        -0.327         -0.172
                             (0.044)      (0.236)      (0.242)        (0.349)        (0.148)

as.factor(ratio_damage)0.2     -0.055       -0.477*      -0.508*        -0.612         -0.309*
                             (0.048)      (0.271)      (0.275)        (0.393)        (0.168)

as.factor(ratio_damage)0.3      0.005        -0.111       -0.075        -0.059         -0.190
                             (0.054)      (0.284)      (0.286)        (0.424)        (0.171)
```

```
as.factor(ratio_damage)0.4     -0.075       -0.593*      -0.587*        -0.861*        -0.329*
                            (0.055)      (0.312)      (0.319)        (0.452)        (0.187)

as.factor(ratio_damage)0.5    -0.103**      -0.770**     -0.793**       -0.974**       -0.516***
                            (0.053)      (0.325)      (0.332)        (0.454)        (0.200)

as.factor(ratio_damage)0.6    -0.109**     -0.824***    -0.842***       -1.020**       -0.450**
                            (0.052)      (0.319)      (0.326)        (0.448)        (0.190)

as.factor(ratio_damage)0.7     -0.096*      -0.727**    -0.848***       -0.930**       -0.422**
                            (0.051)      (0.300)      (0.308)        (0.426)        (0.179)

as.factor(ratio_damage)0.8    -0.138***    -0.906***    -0.883***       -1.299***      -0.636***
                            (0.051)      (0.302)      (0.306)        (0.425)        (0.184)

as.factor(ratio_damage)0.9    -0.103**      -0.708**     -0.687**       -1.046***      -0.551***
                            (0.047)      (0.276)      (0.280)        (0.394)        (0.167)

as.factor(ratio_damage)1       -0.063       -0.497       -0.452         -0.711*       -0.400**
                            (0.052)      (0.306)      (0.309)        (0.425)        (0.179)

-------------------------------------------------------------------------------------------------
Observations                   2,152        2,152        2,099          2,152          2,152
R2                            0.256
Adjusted R2                    0.225        0.241        0.234          0.230          0.268
Log Likelihood                            -1,117.871   -1,073.862     -5,802.847
UBRE                                      0.039        0.023         12.883          0.694
Residual Std. Error       0.411 (df = 2066)
=================================================================================================
Note:                                                              *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------
> ##
> ## Table A2
> ## Descriptive Statistics (1 of 2)
> ## See 3_CensusMainTxt&AppxA for the descriptive stats for
> ## the outcome variables created from census.
> ##
```

```
> ## ------------------------------------------------------------------------
> ## Summary statistics
> filtered_spatial_covariates <- spatial_covariates %>%
+   as.data.frame() %>%
+   filter(. != "ratio_residential") %>%
+   t() %>%
+   as.vector()
> tableContinuous(
+   vars = combined_tbl    %>%
+     select(
+       all_of(treatment_vec[1]), LaggedDamage,
+       ln_ninka_days, dummy_ninka, dummy_ninka_1993, n_ninka,
+       all_of(placebo_depvar_vec),
+       ratio_residential, all_of(distance_termz_base),
+       all_of(filtered_spatial_covariates),
+       longitude, latitude)       %>%    as.data.frame(),
+   group = combined_tbl$ratio_damage > median(combined_tbl$ratio_damage),
+   stats = c("n", "mean", "s", "median", "iqr"),
+   prec = 3)
% latex table generated in R 4.2.2 by xtable 1.8-4 package
% Tue Apr 25 17:06:45 2023
¥begingroup¥footnotesize
¥begin{longtable}{llrrrrr}
 ¥textbf{Variable}  &  ¥textbf{Levels}  &  $¥mathbf{n}$  &  $¥mathbf{¥bar{x}}$  &  $¥mathbf{s}$  &  $¥mathbf{¥widetilde{x}}$  &
¥textbf{IQR} ¥¥
  ¥hline
ratio¥_damage & FALSE & 1127 &   0.157 & 0.171 &   0.100 & 0.300 ¥¥
   & TRUE & 1028 &   0.838 & 0.129 &   0.900 & 0.200 ¥¥
   ¥hline
 & all & 2155 &   0.482 & 0.373 &   0.500 & 0.800 ¥¥
   ¥hline
LaggedDamage & FALSE & 1127 &   0.257 & 0.240 &   0.200 & 0.402 ¥¥
   & TRUE & 1028 &   0.730 & 0.181 &   0.767 & 0.252 ¥¥
   ¥hline
 & all & 2155 &   0.483 & 0.319 &   0.520 & 0.600 ¥¥
   ¥hline
ln¥_ninka¥_days & FALSE & 1127 &   2.720 & 4.013 &   0.000 & 8.187 ¥¥
   & TRUE & 1025 &   2.748 & 4.000 &   0.000 & 8.286 ¥¥
   ¥hline
 & all & 2152 &   2.734 & 4.006 &   0.000 & 8.218 ¥¥
   ¥hline
dummy¥_ninka & FALSE & 1127 &   0.318 & 0.466 &   0.000 & 1.000 ¥¥
   & TRUE & 1025 &   0.325 & 0.469 &   0.000 & 1.000 ¥¥
```

```
   ¥hline
 & all & 2152 &   0.321 & 0.467 &   0.000 & 1.000 ¥¥
   ¥hline
dummy¥_ninka¥_1993 & FALSE & 1098 &   0.300 & 0.458 &   0.000 & 1.000 ¥¥
   & TRUE & 1001 &   0.309 & 0.462 &   0.000 & 1.000 ¥¥
   ¥hline
 & all & 2099 &   0.304 & 0.460 &   0.000 & 1.000 ¥¥
   ¥hline
n¥_ninka & FALSE & 1127 &   0.374 & 0.605 &   0.000 & 1.000 ¥¥
   & TRUE & 1025 &   0.393 & 0.633 &   0.000 & 1.000 ¥¥
   ¥hline
 & all & 2152 &   0.383 & 0.618 &   0.000 & 1.000 ¥¥
   ¥hline
SportClubDummy & FALSE & 1127 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   & TRUE & 1028 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   ¥hline
 & all & 2155 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   ¥hline
LocalSportClubDummy & FALSE & 1127 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   & TRUE & 1028 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   ¥hline
 & all & 2155 &   0.000 & 0.000 &   0.000 & 0.000 ¥¥
   ¥hline
ratio¥_residential & FALSE & 1127 &   0.628 & 0.324 &   0.700 & 0.600 ¥¥
   & TRUE & 1028 &   0.874 & 0.184 &   0.900 & 0.200 ¥¥
   ¥hline
 & all & 2155 &   0.746 & 0.293 &   0.900 & 0.400 ¥¥
   ¥hline
palace¥_dist¥_ln & FALSE & 1127 &   2.147 & 0.491 &   2.304 & 0.410 ¥¥
   & TRUE & 1028 &   1.753 & 0.514 &   1.798 & 0.620 ¥¥
   ¥hline
 & all & 2155 &   1.959 & 0.539 &   2.079 & 0.733 ¥¥
   ¥hline
minTargetDistance¥_ln & FALSE & 1127 &   1.029 & 0.848 &   1.174 & 1.323 ¥¥
   & TRUE & 1028 &   0.464 & 0.711 &   0.506 & 1.022 ¥¥
   ¥hline
 & all & 2155 &   0.760 & 0.834 &   0.804 & 1.218 ¥¥
   ¥hline
poly¥_area¥_ln & FALSE & 1127 &  -1.772 & 0.487 &  -1.716 & 0.572 ¥¥
   & TRUE & 1028 &  -2.047 & 0.604 &  -1.963 & 0.762 ¥¥
   ¥hline
 & all & 2155 &  -1.903 & 0.563 &  -1.810 & 0.665 ¥¥
   ¥hline
```

```
n¥_neighbor & FALSE & 1127 &   6.109 & 1.388 &   6.000 & 2.000 ¥¥
   & TRUE & 1028 &   6.187 & 1.361 &   6.000 & 2.000 ¥¥
   ¥hline
 & all & 2155 &   6.146 & 1.375 &   6.000 & 2.000 ¥¥
   ¥hline
PopDensity¥_1939¥_km2¥_ln & FALSE & 1127 &   9.027 & 1.095 &   9.078 & 1.571 ¥¥
   & TRUE & 1028 &   9.995 & 0.892 &  10.107 & 0.821 ¥¥
   ¥hline
 & all & 2155 &   9.489 & 1.114 &   9.725 & 1.429 ¥¥
   ¥hline
mean¥_elevation¥_ln & FALSE & 1127 &   3.241 & 0.632 &   3.445 & 1.278 ¥¥
   & TRUE & 1028 &   2.995 & 0.540 &   2.958 & 1.036 ¥¥
   ¥hline
 & all & 2155 &   3.124 & 0.603 &   3.245 & 1.182 ¥¥
   ¥hline
mean¥_slope¥_ln & FALSE & 1127 &   2.367 & 0.067 &   2.342 & 0.080 ¥¥
   & TRUE & 1028 &   2.370 & 0.074 &   2.335 & 0.097 ¥¥
   ¥hline
 & all & 2155 &   2.368 & 0.071 &   2.339 & 0.087 ¥¥
   ¥hline
river¥_distance¥_ln & FALSE & 1127 &  -0.861 & 0.901 &  -0.759 & 1.266 ¥¥
   & TRUE & 1028 &  -0.642 & 0.885 &  -0.547 & 1.342 ¥¥
   ¥hline
 & all & 2155 &  -0.757 & 0.900 &  -0.656 & 1.319 ¥¥
   ¥hline
railway¥_length¥_ln & FALSE & 1127 &   0.142 & 0.215 &   0.000 & 0.279 ¥¥
   & TRUE & 1028 &   0.181 & 0.228 &   0.069 & 0.320 ¥¥
   ¥hline
 & all & 2155 &   0.160 & 0.222 &   0.000 & 0.297 ¥¥
   ¥hline
n¥_stations & FALSE & 1127 &   0.289 & 0.668 &   0.000 & 0.000 ¥¥
   & TRUE & 1028 &   0.421 & 0.792 &   0.000 & 1.000 ¥¥
   ¥hline
 & all & 2155 &   0.352 & 0.732 &   0.000 & 0.000 ¥¥
   ¥hline
SpLag¥_railway¥_length¥_ln & FALSE & 1127 &   0.666 & 0.446 &   0.680 & 0.614 ¥¥
   & TRUE & 1028 &   0.851 & 0.448 &   0.856 & 0.611 ¥¥
   ¥hline
 & all & 2155 &   0.754 & 0.456 &   0.763 & 0.628 ¥¥
   ¥hline
SpLag¥_n¥_stations & FALSE & 1127 &   1.690 & 2.034 &   1.000 & 2.000 ¥¥
   & TRUE & 1028 &   2.842 & 2.629 &   2.000 & 3.000 ¥¥
   ¥hline
```

```
 & all & 2155 &   2.240 & 2.406 &   1.000 & 3.000 ¥¥
   ¥hline
longitude & FALSE & 1127 & 139.720 & 0.078 & 139.702 & 0.117 ¥¥
   & TRUE & 1028 & 139.749 & 0.047 & 139.737 & 0.077 ¥¥
   ¥hline
 & all & 2155 & 139.734 & 0.066 & 139.728 & 0.097 ¥¥
   ¥hline
latitude & FALSE & 1127 &  35.688 & 0.058 &  35.695 & 0.093 ¥¥
   & TRUE & 1028 &  35.682 & 0.053 &  35.695 & 0.058 ¥¥
   ¥hline
 & all & 2155 &  35.685 & 0.056 &  35.695 & 0.077 ¥¥
   ¥hline
¥hline
¥caption{}
¥label{}
¥end{longtable}
¥endgroup
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A4
> ## Result V: >median damage dummy
> ##
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ratio_damage + binary_damage_ratio")      %>%    as.formula() ##  formula  obj
to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[3], "~ . - ratio_damage + binary_damage_ratio")) %>%
+     gam(data = combined_tbl, family = binom_link),
```

```
+   ## ----------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . - ratio_damage + binary_damage_ratio")
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ----------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[2], "~ . - ratio_damage + binary_damage_ratio")) %>%
+     gam(data = combined_tbl),
+   ## ----------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+     update(str_c(depvar_vec_combined[4], "~ . - ratio_damage + binary_damage_ratio")) %>%
+     gam(data = combined_tbl, family = count_link)
+   ## ----------------------------------------------------------------------
+ )
> ## ----------------------------------------------------------------------
> ## Print regression estimates
> ## ----------------------------------------------------------------------
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                        Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm         GAM         GAM           GAM     gam: quasipoisson
                                (logistic)   (logistic)   (continuous)    link = log
                       (1)         (2)         (3)           (4)           (5)
---------------------------------------------------------------------------------------------
binary_damage_ratio    -0.055**      -0.323**     -0.328**      -0.490**       -0.219***
                      (0.025)      (0.151)      (0.155)       (0.215)        (0.084)

---------------------------------------------------------------------------------------------
Observations            2,152        2,152        2,099         2,152          2,152
R2                     0.252
```

```
Adjusted R2             0.225        0.240        0.233          0.230          0.267
Log Likelihood                    -1,114.453   -1,071.047    -5,799.266
UBRE                              0.036        0.021         12.836          0.693
Residual Std. Error 0.411 (df = 2075)
================================================================================================
Note:                                                          *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A5
> ## Result IV: >70 residential ratio subsample
> ##
> ## ------------------------------------------------------------------------
> ## Panel A: Damage ratio
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula() ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl    %>%    filter(ratio_residential >= 0.7), keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]       %>%
+     update(str_c(depvar_vec_combined[3], " ~ "))    %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ .")
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[2], " ~ "))    %>%
+     # update(str_c(depvar_vec_combined[1], " ~ .")) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7)),
```

```
+   ## ------------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+     update(str_c(depvar_vec_combined[4], " ~ ."))    %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = count_link)
+   ## ------------------------------------------------------------------------
+ )
> ## ------------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------------
> ## Table A5: Panel A
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                          Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm         GAM         GAM           GAM     gam: quasipoisson
                                (logistic)   (logistic)   (continuous)    link = log
                       (1)         (2)         (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage           -0.081*      -0.584**     -0.578**      -0.841**       -0.467***
                      (0.046)      (0.274)      (0.281)       (0.392)        (0.144)

---------------------------------------------------------------------------------------------
Observations            1,522        1,522        1,483         1,522          1,522
R2                     0.278
Adjusted R2             0.240        0.259        0.248         0.243          0.274
Log Likelihood                      -820.426     -795.572     -4,141.291
UBRE                               0.078        0.073        13.534         0.725
Residual Std. Error 0.419 (df = 1445)
=============================================================================================
Note:                                                        *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
> ## Panel B: Dummy regressions
> ## ------------------------------------------------------------------------
> rslt_lst_dummy = list(
+   ## ------------------------------------------------------------------------
```

```
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])   %>%    as.formula() ##
formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl    %>%    filter(ratio_residential >= 0.7), keepCX = TRUE, psdef = FALSE),
+   ## -------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = binom_link),
+   ## -------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = binom_link),
+   ## -------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(
+       str_c(depvar_vec_combined[2], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7)),
+   ## -------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[4], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(ratio_residential >= 0.7), family = count_link)
+   ## -------------------------------------------------------------------
+ )
> ## -------------------------------------------------------------------
> ## Print regression estimates
> ## -------------------------------------------------------------------
> ## Table A5: Panel B
> stargazer(
+   rslt_lst_dummy,
```

```
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec)
+ )

==========================================================================================
                                        Dependent variable:
                      ---------------------------------------------------------------------
                            dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                           felm         GAM         GAM           GAM     gam: quasipoisson
                                    (logistic)   (logistic)   (continuous)    link = log
                            (1)         (2)         (3)           (4)           (5)
------------------------------------------------------------------------------------------
as.factor(ratio_damage)0.1     -0.055       -0.356      -0.404       -0.459        -0.131
                           (0.062)      (0.313)     (0.323)       (0.486)       (0.187)

as.factor(ratio_damage)0.2     -0.052       -0.357      -0.362       -0.483        -0.167
                           (0.067)      (0.347)     (0.355)       (0.534)       (0.202)

as.factor(ratio_damage)0.3     -0.009       -0.145      -0.090       -0.103        -0.169
                           (0.075)      (0.372)     (0.378)       (0.571)       (0.211)

as.factor(ratio_damage)0.4     -0.090       -0.630      -0.575       -0.947        -0.294
                           (0.076)      (0.385)     (0.395)       (0.590)       (0.222)

as.factor(ratio_damage)0.5     -0.134*     -0.827**    -0.881**      -1.076*      -0.483**
                           (0.071)      (0.399)     (0.409)       (0.595)       (0.233)

as.factor(ratio_damage)0.6     -0.102      -0.718*     -0.738*       -0.912        -0.354
                           (0.071)      (0.383)     (0.392)       (0.574)       (0.216)

as.factor(ratio_damage)0.7     -0.096      -0.708*     -0.872**      -0.891*       -0.377*
                           (0.066)      (0.362)     (0.374)       (0.538)       (0.208)

as.factor(ratio_damage)0.8    -0.132**     -0.855**    -0.883**     -1.156**      -0.566***
                           (0.067)      (0.368)     (0.376)       (0.544)       (0.213)

as.factor(ratio_damage)0.9     -0.106*     -0.708**    -0.728**     -1.023**      -0.524***
                           (0.063)      (0.342)     (0.350)       (0.510)       (0.197)

as.factor(ratio_damage)1       -0.086      -0.632*      -0.587       -0.903*      -0.422**
                           (0.066)      (0.372)     (0.379)       (0.546)       (0.210)
```

```
-------------------------------------------------------------------------------------------------------
Observations                   1,522        1,522        1,483          1,522          1,522
R2                           0.280
Adjusted R2                    0.238        0.255        0.244          0.240          0.270
Log Likelihood                             -826.969     -800.928     -4,148.097
UBRE                                      0.087        0.080         13.664          0.731
Residual Std. Error       0.420 (df = 1436)
=======================================================================================================
Note:                                                                 *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A6
> ## Result III: Close-palace subsample (PalaceDist <= 10 km)
> ##
> ## ------------------------------------------------------------------------
> ## Panel A: Damage ratio
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula() ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl    %>%    filter(palace_dist <= 10), keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[3], " ~ "))    %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ .")
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]     %>%    update(str_c(depvar_vec_combined[2], " ~ "))  %>%
```

```
+    # update(str_c(depvar_vec_combined[1], " ~ .")) %>%
+    gam(data = combined_tbl     %>%    filter(palace_dist <= 10)),
+   ## ---------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%    update(str_c(depvar_vec_combined[4], " ~ .")) %>%
+    gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = count_link)
+   ## ---------------------------------------------------------------------
+ )
> ## ---------------------------------------------------------------------
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> ## Table A6: Panel A
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                         Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm         GAM         GAM           GAM     gam: quasipoisson
                                (logistic)   (logistic)   (continuous)    link = log
                       (1)         (2)         (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage           -0.076*      -0.548*      -0.533*       -0.726**       -0.429***
                     (0.044)      (0.283)      (0.287)        (0.369)        (0.155)

---------------------------------------------------------------------------------------------
Observations            1,467        1,467        1,444          1,467          1,467
R2                     0.299
Adjusted R2             0.261        0.284        0.277          0.265          0.332
Log Likelihood                     -724.807     -707.770     -3,912.795
UBRE                              -0.012       -0.020        12.151          0.656
Residual Std. Error 0.398 (df = 1392)
=============================================================================================
Note:                                                       *p<0.1; **p<0.05; ***p<0.01
>
> ## ---------------------------------------------------------------------
> ## Panel B: Dummy regressions
> ## ---------------------------------------------------------------------
> rslt_lst_dummy = list(
```

```
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])   %>%   as.formula() ##
formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl    %>%    filter(palace_dist <= 10), keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]     %>%
+     update(
+       str_c(depvar_vec_combined[2], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10)),
+   ## ------------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[4], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3])
+     ) %>%
+     gam(data = combined_tbl     %>%    filter(palace_dist <= 10), family = count_link)
+   ## ------------------------------------------------------------------------
+ )
> ## ------------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------------
> ## Table A6: Panel B
> stargazer(
```

```
+   rslt_lst_dummy,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )
```

	Dependent variable:				
	dummy_ninka		dummy_ninka_1993	ln_ninka_days	n_ninka
	felm	GAM (logistic)	GAM (logistic)	GAM (continuous)	gam: quasipoisson link = log
	(1)	(2)	(3)	(4)	(5)
as.factor(ratio_damage)0.1	-0.050	-0.439	-0.387	-0.454	-0.317
	(0.067)	(0.350)	(0.355)	(0.493)	(0.208)
as.factor(ratio_damage)0.2	-0.103	-0.723*	-0.668*	-0.877*	-0.437*
	(0.069)	(0.380)	(0.385)	(0.528)	(0.228)
as.factor(ratio_damage)0.3	-0.015	-0.179	-0.074	-0.049	-0.216
	(0.076)	(0.395)	(0.399)	(0.552)	(0.228)
as.factor(ratio_damage)0.4	-0.084	-0.607	-0.505	-0.773	-0.356
	(0.076)	(0.416)	(0.423)	(0.575)	(0.241)
as.factor(ratio_damage)0.5	-0.159**	-1.140***	-1.093***	-1.348**	-0.740***
	(0.068)	(0.416)	(0.423)	(0.551)	(0.255)
as.factor(ratio_damage)0.6	-0.150**	-1.155***	-1.097***	-1.323**	-0.610**
	(0.067)	(0.412)	(0.420)	(0.540)	(0.244)
as.factor(ratio_damage)0.7	-0.123*	-0.888**	-0.944**	-1.043**	-0.495**
	(0.067)	(0.379)	(0.389)	(0.515)	(0.221)
as.factor(ratio_damage)0.8	-0.165**	-1.151***	-1.124***	-1.480***	-0.740***
	(0.066)	(0.378)	(0.386)	(0.509)	(0.225)
as.factor(ratio_damage)0.9	-0.123*	-0.856**	-0.781**	-1.104**	-0.643***
	(0.064)	(0.357)	(0.363)	(0.486)	(0.211)
as.factor(ratio_damage)1	-0.072	-0.530	-0.466	-0.694	-0.489**
	(0.068)	(0.390)	(0.396)	(0.516)	(0.226)

```
-------------------------------------------------------------------------------------------------------
Observations                    1,467        1,467        1,444          1,467          1,467
R2                             0.305
Adjusted R2                     0.264        0.284        0.277          0.268          0.325
Log Likelihood                              -726.360     -708.962     -3,914.315
UBRE                                        -0.010       -0.018        12.185          0.658
Residual Std. Error       0.398 (df = 1383)
=======================================================================================================
Note:                                                                 *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A7
> ## Result VI: with spatially lagged Damage
> ##
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . + LaggedDamage") %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[3], "~ . + LaggedDamage"))   %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . + LaggedDamage")
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]     %>%
+     update(str_c(depvar_vec_combined[2], "~ . + LaggedDamage"))   %>%
```

```
+    gam(data = combined_tbl),
+   ## ---------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+    update(str_c(depvar_vec_combined[4], "~ . + LaggedDamage"))   %>%
+    gam(data = combined_tbl, family = count_link)
+   ## ---------------------------------------------------------------------
+ )
> ## ---------------------------------------------------------------------
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> ## Table A7: Panel A
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec, "LaggedDamage")
+ )

=============================================================================================
                                         Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm         GAM         GAM           GAM     gam: quasipoisson
                                (logistic)   (logistic)   (continuous)    link = log
                       (1)         (2)         (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage           -0.059       -0.332      -0.296       -0.565       -0.304**
                      (0.046)      (0.275)     (0.280)       (0.389)       (0.153)

LaggedDamage           -0.095      -0.910*     -0.872*       -1.003        -0.513*
                      (0.076)      (0.481)     (0.487)       (0.655)       (0.280)

---------------------------------------------------------------------------------------------
Observations            2,152       2,152       2,099         2,152         2,152
R2                     0.253
Adjusted R2             0.225       0.243       0.235         0.230         0.271
Log Likelihood                    -1,113.433   -1,070.556    -5,798.657
UBRE                              0.035       0.020        12.829         0.691
Residual Std. Error 0.411 (df = 2074)
=============================================================================================
Note:                                                      *p<0.1; **p<0.05; ***p<0.01
>
```

```
> ## ------------------------------------------------------------------------
> ## Panel B: Dummy regressions
> ## ------------------------------------------------------------------------
> rslt_lst_dummy = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+         str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3], " + LaggedDamage")
       %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3], " + LaggedDamage")
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[3], "_1993 ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3], " + LaggedDamage")
+     ) %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(
+       str_c(depvar_vec_combined[2], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3], " + LaggedDamage")
+     ) %>%
+     gam(data = combined_tbl),
+   ## ------------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%
+     update(
+       str_c(depvar_vec_combined[4], " ~ . - ", treatment_vec[1], " + ", robust_treatment_vec[3], " + LaggedDamage")
+     ) %>%
+     gam(data = combined_tbl, family = count_link)
+   ## ------------------------------------------------------------------------
+ )
> ## ------------------------------------------------------------------------
```

```
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> ## Table A7: Panel B
> stargazer(
+   rslt_lst_dummy,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec)
+ )

=================================================================================================
                                                Dependent variable:
                        ---------------------------------------------------------------------------
                              dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                             felm         GAM         GAM           GAM     gam: quasipoisson
                                       (logistic)   (logistic)   (continuous)    link = log
                              (1)         (2)         (3)           (4)           (5)
-------------------------------------------------------------------------------------------------
as.factor(ratio_damage)0.1     -0.025       -0.277      -0.319       -0.288        -0.156
                             (0.044)      (0.237)     (0.242)       (0.349)       (0.148)

as.factor(ratio_damage)0.2     -0.046       -0.399      -0.428       -0.521        -0.257
                             (0.048)      (0.273)     (0.278)       (0.397)       (0.170)

as.factor(ratio_damage)0.3      0.016       -0.016      0.025         0.056        -0.134
                             (0.055)      (0.288)     (0.291)       (0.430)       (0.174)

as.factor(ratio_damage)0.4     -0.060       -0.461      -0.451       -0.710        -0.250
                             (0.056)      (0.318)     (0.326)       (0.461)       (0.191)

as.factor(ratio_damage)0.5     -0.085      -0.623*     -0.645*       -0.792*      -0.436**
                             (0.055)      (0.333)     (0.340)       (0.468)       (0.204)

as.factor(ratio_damage)0.6     -0.090*     -0.672**    -0.684**      -0.828*       -0.365*
                             (0.054)      (0.328)     (0.335)       (0.464)       (0.195)

as.factor(ratio_damage)0.7     -0.076      -0.578*     -0.693**      -0.732*       -0.343*
                             (0.053)      (0.309)     (0.318)       (0.443)       (0.184)

as.factor(ratio_damage)0.8    -0.113**     -0.690**    -0.661**     -1.049**      -0.523***
                             (0.054)      (0.320)     (0.326)       (0.452)       (0.194)

as.factor(ratio_damage)0.9     -0.074       -0.467      -0.438       -0.759*      -0.422**
```

```
                            (0.052)      (0.302)      (0.307)       (0.431)       (0.181)

as.factor(ratio_damage)1       -0.028       -0.203       -0.149       -0.364        -0.244
                            (0.058)      (0.340)      (0.345)       (0.476)       (0.198)

-------------------------------------------------------------------------------------------------
Observations                  2,152        2,152        2,099         2,152         2,152
R2                           0.256
Adjusted R2                   0.226        0.243        0.236         0.230         0.269
Log Likelihood                          -1,117.177   -1,073.173    -5,802.701
UBRE                                    0.038        0.023        12.882         0.694
Residual Std. Error       0.411 (df = 2065)
=================================================================================================
Note:                                                          *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A9
> ## Result I: ANA result---continuous damage, 5th poly
> ##
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]      %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula() ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[4]]      %>%    update(str_c(depvar_vec_combined[3], " ~ "))  %>%
+     glm(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[4]]    %>%    update(str_c(depvar_vec_combined[3], "_1993 ~ "))    %>%
+     glm(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
```

```
+   NinkaDays_Linear = mdl_lst[[4]]      %>%    update(str_c(depvar_vec_combined[2], " ~ ."))  %>%
+     # update(str_c(depvar_vec_combined[1], " ~ ."))  %>%
+     glm(data = combined_tbl),
+   ## ---------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[4]]    %>%    update(str_c(depvar_vec_combined[4], " ~ ."))  %>%
+     glm(data = combined_tbl, family = count_link)
+   ## ---------------------------------------------------------------------
+ )
Warning messages:
1: glm.fit: fitted probabilities numerically 0 or 1 occurred
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> ## ---------------------------------------------------------------------
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> ## Table 1: Panel A
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                        Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                      felm       logistic     logistic       normal    glm: quasipoisson
                                                                        link = log
                       (1)         (2)          (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage          -0.088**      -0.583**     -0.532**      -0.779**       -0.447***
                     (0.040)      (0.240)      (0.244)       (0.337)        (0.132)

---------------------------------------------------------------------------------------------
Observations            2,152        2,152        2,099         2,152          2,152
R2                    0.253
Adjusted R2             0.225
Log Likelihood                    -1,040.723   -1,003.765    -5,731.173
Akaike Inf. Crit.                  2,235.447    2,161.529     11,616.350
Residual Std. Error 0.411 (df = 2075)
=============================================================================================
Note:                                                      *p<0.1; **p<0.05; ***p<0.01
>
```

```
> ## ------------------------------------------------------------------------
> ##
> ## Table A10
> ## Result XI: Baseline regressions without some of AP and IP polynomials
> ##
> ## ------------------------------------------------------------------------
>
> ## --- Polynomials and splines only IP
> distance_term_specification_IP = list(
+   str_c("poly(", distance_termz_base[1], ", degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   str_c("s(", distance_termz_base[1], ")")
+ )
>
> ## --- Polynomials and splines only AP
> distance_term_specification_AP = list(
+   str_c("poly(", distance_termz_base[2], ", degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   str_c("s(", distance_termz_base[2], ")")
+ )
>
>
> mdl_lst = c(mdl_lst,
+             ## ----------------------------------------
+             ## Additional models for this table
+             ## ----------------------------------------
+             ## Without polynomials of longitude & latitude
+             polynomial_base_wo_lonlat = chr2fml_felm(
+               "outcome",
+               idv_list = list(
+                 treatment_vec[1],
+                 spatial_covariates,
+                 distance_term_specification[[1]]
+               ),
+               fe_list = fe_term,
+               se_cluster = "row_count"
+             ),
+             ## Without polynomials of Aiming Points
+             polynomial_base_wo_AP = chr2fml_felm(
+               "outcome",
+               idv_list = list(
+                 treatment_vec[1],
+                 spatial_covariates,
+                 distance_term_specification_IP[[1]],
+                 spatial_term_specification[[1]]
```

```
+             ),
+             fe_list = fe_term,
+             se_cluster = "row_count"
+           ),
+           ## Without polynomials of Imperial Palace
+           polynomial_base_wo_IP = chr2fml_felm(
+             "outcome",
+             idv_list = list(
+               treatment_vec[1],
+               spatial_covariates,
+               distance_term_specification_AP[[1]],
+               spatial_term_specification[[1]]
+             ),
+             fe_list = fe_term,
+             se_cluster = "row_count"
+           ),
+           ## Without polynomials of AP and IP
+           polynomial_base_wo_APIP = chr2fml_felm(
+             "outcome",
+             idv_list = list(
+               treatment_vec[1],
+               spatial_covariates,
+               spatial_term_specification[[1]]
+             ),
+             fe_list = fe_term,
+             se_cluster = "row_count"
+           ),
+           ## Without polynomials of lon&lat, AP, IP
+           polynomial_base_wo_LonLatAPIP = chr2fml_felm(
+             "outcome",
+             idv_list = list(
+               treatment_vec[1],
+               spatial_covariates
+             ),
+             fe_list = fe_term,
+             se_cluster = "row_count"
+           )
+ )
>
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
```

```
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## 1b. DV = dummy_ninka, LPM
+   ## wo_LonLat
+   Dummy_Ninka_LPM = mdl_lst[[5]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## 1c. DV = dummy_ninka, LPM
+   ## wo_AP
+   Dummy_Ninka_LPM = mdl_lst[[6]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## 1d. DV = dummy_ninka, LPM
+   ## wo_IP
+   Dummy_Ninka_LPM = mdl_lst[[7]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## 1e. DV = dummy_ninka, LPM
+   ## wo_APIP
+   Dummy_Ninka_LPM = mdl_lst[[8]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## 1f. DV = dummy_ninka, LPM
+   ## wo_LonLatAPIP
+   Dummy_Ninka_LPM = mdl_lst[[9]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE)
+ )
```

```
>
> ## ------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec, robust_treatment_vec)
+ )

=================================================================================================================
=====
                                                Dependent variable:
                 ---------------------------------------------------------------------------------------------------
--
                                                   dummy_ninka
                       (1)            (2)            (3)            (4)            (5)            (6)
-----------------------------------------------------------------------------------------------------------------
-----
ratio_damage          -0.088**       -0.096**       -0.082**       -0.103***      -0.094**       -0.119***
                     (0.040)        (0.039)        (0.039)        (0.039)        (0.039)        (0.037)

-----------------------------------------------------------------------------------------------------------------
-----
Observations           2,152          2,152          2,152          2,152          2,152          2,152
R2                    0.253          0.228          0.248          0.249          0.246          0.220
Adjusted R2            0.225          0.208          0.222          0.223          0.222          0.203
Residual Std. Error 0.411 (df = 2075) 0.416 (df = 2095) 0.412 (df = 2080) 0.412 (df = 2080) 0.412 (df = 2085) 0.417 (df =
2105)
=================================================================================================================
=====
Note:                                                                          *p<0.1; **p<0.05; ***p<0.01
>
> ## ------------------------------------------------------------------
> ##
> ## Table A.11
> ## Result II: Regression with 5th ordered polynomials of prewar covariates
> ##
> ## ------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------
> rslt_lst = list(
```

```
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst[[2]]       %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ . + ", extra_covariates_specification[2])     %>%    as.formula() ##  formula  obj
to feed felm()
+     ) %>%
+     felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst[[3]]       %>%    update(str_c(depvar_vec_combined[3],       "       ~       .       +       ",
extra_covariates_specification[2]))     %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst[[3]]    %>%    update(str_c(depvar_vec_combined[3],     "_1993    ~     .     +     ",
extra_covariates_specification[2]))     %>%
+     gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst[[3]]      %>%
+     update(str_c(depvar_vec_combined[2], " ~ . + ", extra_covariates_specification[2]))     %>%
+     # update(str_c(depvar_vec_combined[1], " ~ . + ", extra_covariates_specification[2]))   %>%
+     gam(data = combined_tbl),
+   ## ------------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst[[3]]    %>%    update(str_c(depvar_vec_combined[4], " ~ . + ", extra_covariates_specification[2]))
       %>%
+     gam(data = combined_tbl, family = count_link)
+ )
Warning message:
In chol.default(mat, pivot = TRUE, tol = tol) :
  the matrix is either rank-deficient or indefinite
>
> ## ------------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------------
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )
```

```
=================================================================================================
                                        Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                     felm         GAM         GAM           GAM     gam: quasipoisson
                               (logistic)   (logistic)   (continuous)    link = log
                      (1)         (2)         (3)           (4)           (5)
-------------------------------------------------------------------------------------------------
ratio_damage          -0.100**     -0.617**     -0.581**      -0.937***      -0.443***
                     (0.040)      (0.248)      (0.252)       (0.343)        (0.136)

-------------------------------------------------------------------------------------------------
Observations           2,152       2,152       2,099         2,152          2,152
R2                    0.262
Adjusted R2            0.229       0.254       0.247         0.232          0.277
Log Likelihood                   -1,111.010   -1,065.961    -5,804.488
UBRE                             0.033       0.016        12.907          0.690
Residual Std. Error 0.410 (df = 2057)
=================================================================================================
Note:                                                        *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A12
> ## Result VIII: Baseline regressions with zip code-level data
> ##
> ## ------------------------------------------------------------------------
> ## Baseline formula objects
> ## ------------------------------------------------------------------------
> mdl_lst_zipcode = mdl_lst
>
> ## ------------------------------------------------------------------------
> ## Panel A: Damage ratio
> ## ------------------------------------------------------------------------
> rslt_lst_zipcode = list(
+   ## ---------------------------------------------------------------------
+   ## 1a. DV = dummy_ninka, LPM
+   Dummy_Ninka_LPM = mdl_lst_zipcode[[2]]     %>%
+     update(
+       str_c(depvar_vec_combined[3], " ~ .")   %>%    as.formula()  ## formula obj to feed felm()
+     ) %>%
```

```
+    felm(data = zipcode_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ---------------------------------------------------------------------
+   ## 1b: DV = dummy_ninka, probit/logit link
+   Dummy_Ninka_GAM = mdl_lst_zipcode[[3]]     %>%    update(str_c(depvar_vec_combined[3], " ~ ."))  %>%
+    gam(data = zipcode_tbl, family = binom_link),
+   ## ---------------------------------------------------------------------
+   ## 1c: DV = dummy_ninka (post-1993 period), probit/logit link
+   Dummy_Ninka_GAM_post1993 = mdl_lst_zipcode[[3]]   %>%    update(str_c(depvar_vec_combined[3], "_1993 ~ ."))    %>%
+    gam(data = zipcode_tbl, family = binom_link),
+   ## ---------------------------------------------------------------------
+   ## 2. DV = log(ninka_days), Gaussian link
+   NinkaDays_Linear = mdl_lst_zipcode[[3]]    %>%    update(str_c(depvar_vec_combined[2], " ~ ."))  %>%
+    # update(str_c(depvar_vec_combined[1], " ~ ."))  %>%
+    gam(data = zipcode_tbl),
+   ## ---------------------------------------------------------------------
+   ## 3. DV = n_ninka, (quasi-)Poisson link
+   N_Ninka_GAM = mdl_lst_zipcode[[3]]   %>%    update(str_c(depvar_vec_combined[4], " ~ ."))  %>%
+    gam(data = zipcode_tbl, family = count_link)
+   ## ---------------------------------------------------------------------
+ )
> ## ---------------------------------------------------------------------
> ## Print regression estimates
> ## ---------------------------------------------------------------------
> stargazer(
+   rslt_lst_zipcode,
+   type = "text",
+   align = TRUE, keep = c(treatment_vec, robust_treatment_vec)
+ )

=============================================================================================
                                        Dependent variable:
                 ---------------------------------------------------------------------------
                       dummy_ninka        dummy_ninka_1993 ln_ninka_days     n_ninka
                     felm         GAM          GAM           GAM     gam: quasipoisson
                               (logistic)    (logistic)    (continuous)    link = log
                     (1)          (2)          (3)           (4)           (5)
---------------------------------------------------------------------------------------------
ratio_damage          -0.072      -0.955*       -0.805        -0.810        -0.711**
                    (0.081)      (0.549)      (0.527)        (0.624)        (0.292)

---------------------------------------------------------------------------------------------
Observations           718          718          714           718           718
R2                   0.425
```

```
Adjusted R2            0.357        0.395        0.365          0.357          0.328
Log Likelihood                     -360.815     -362.734     -1,885.616
UBRE                              0.005        0.016        11.248          0.551
Residual Std. Error 0.394 (df = 641)
================================================================================================
Note:                                                        *p<0.1; **p<0.05; ***p<0.01
>
>
>
>
> ## EOF
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Regress census outcomes on raid damages
> ## Note:
> ##
> ## ------------------------------------------------------------------------
>
> ## environment setting
> Sys.setenv(LANGUAGE="en")
> gc();gc()
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5084876 271.6    7833997 418.4  7833997 418.4
Vcells 12320452  94.0   25848289 197.3 25848289 197.3
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5084879 271.6    7833997 418.4  7833997 418.4
Vcells 12320475  94.0   25848289 197.3 25848289 197.3
> rm(list = ls())
> options(scipen = 999)   ## Disable scientific notation
>
> ## ------------------------------------------------------------------------
> ##
> ## Initial settings
> ##
> ## ------------------------------------------------------------------------
> ## Set directory
> ## ------------------------------------------------------------------------
> root_dir = "E:/Dropbox/429_MDA_replication/Replication"
> ## Load packages and functions
> source(file.path(root_dir, "2_Code/1_PackagesFunctions.R"))
>
```

```
> ## ------------------------------------------------------------------------
> ## Data directory
> data_dir = root_dir      %>%    file.path("1_Data") %>%    dir_create()
> ## Output directory
> output_dir = root_dir    %>%    file.path("3_Result")      %>%    dir_create()
> figure_dir = root_dir    %>%    file.path("4_Figures")     %>%    dir_create()
> ## Working directory
> working_dir = root_dir   %>%    file.path("5_Working")     %>%    dir_create()
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Prepare variables
> ##
> ## ------------------------------------------------------------------------
> ## Dependent variables
> ## ------------------------------------------------------------------------
> ## Logged DVs
> depvar_vec = c(
+   "lnr_unemp", "lnrm_unemp", "lnrf_unemp",
+   "lnr_proexe", "lnave_lvlength", "lnave_eduyr"
+ )
> ## ------------------------------------------------------------------------
> ## Logged negative controls
> negative_cntrlz = c("lnrmale0004")
> ## ------------------------------------------------------------------------
> ## Dependent variable vector (all DVs and negative controls combined): 22 DVs
> depvar_vec_combined = c(depvar_vec, negative_cntrlz)
> ## Dependent variables: for alternative treatment indicators
> depvar_vec_alternative_treatment = c(depvar_vec, negative_cntrlz)
>
> ## ------------------------------------------------------------------------
> ## Treatment variables
> ## ------------------------------------------------------------------------
> treatment_vec = c("ratio_damage", "ln_damage_ratio", "binary_damage_ratio")
> robust_treatment_vec = c(treatment_vec[2:3], str_c("as.factor(", treatment_vec[1],")"))
> ## ------------------------------------------------------------------------
> ## Pretreatment covariates
> ## ------------------------------------------------------------------------
> ## Distance variables
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> ## Polynomial
```

```
> polynomial_degree = 5
> # distance_termz_poly = polynomial_varname(distance_termz_base, degree = polynomial_degree)
> distance_termz_poly = str_c("poly(", distance_termz_base, ", degree = ", polynomial_degree, ", simple = TRUE, raw =
TRUE)")
> distance_term_specification = list( distance_termz_poly, str_c("s(", distance_termz_base, ")") )
> ## Geographical features
> spatial_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## Flammability score
> flame_scorez = c("hazard", "max_nghb_hazard")
> ## ------------------------------------------------------------------------
> ## Spatial spline/polynomials and fixed effects
> ## ------------------------------------------------------------------------
> ## District FEs
> district_fe = "as.factor(prewar_district)"   ## 35-district FE
> ## Spatial terms
> spatial_term_lbl = c("Polynomial", "GAM")    ## This indicates polynomial or spline specification in the csv files
> lonlat_base = c("z_lon", "z_lat")
> spatial_polynomial = str_c("poly(z_lon, z_lat, degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)")
> spatial_term_specification = list(spatial_polynomial, "te(longitude, latitude)")
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Prepare models
> ##
> ## ------------------------------------------------------------------------
> ## 1. Polynomial model
> base_polynomial = chr2fml("outcome",
+                       idv_list = list(
+                         treatment_vec[1], spatial_covariates, district_fe,
+                         distance_term_specification[[1]], spatial_term_specification[[1]]
+                       )
```

```
+ )
> ## ------------------------------------------------------------------------
> ## 2. GAM with additional spline terms
> base_gam = chr2fml("outcome",
+                 idv_list = list(
+                   treatment_vec[1], spatial_covariates, district_fe,
+                   distance_term_specification[[2]], spatial_term_specification[[2]]
+                 )
+ )
> ## ------------------------------------------------------------------------
> ## 3. Factor raid
> factor_polynomial = base_polynomial   %>%    update(str_c(". ~ . -", treatment_vec[1], "+ as.factor(", treatment_vec[1],
")"))
> ## ------------------------------------------------------------------------
> ## Model list obj
> model_list = list(base_polynomial, base_gam)
> model_lbl = c("Polynomial", "GAM")
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Prepare data objects
> ##
> ## ------------------------------------------------------------------------
> ## Read data
> tokyo2keep = data_dir    %>%
+   file.path("census_data4.csv") %>%
+   read_csv(col_types = cols()) %>%
+   mutate(group = "FullSample")
>
>
> ## ------------------------------------------------------------------------
> ## Nested data
> ## --- Full sample
> full_sample = tokyo2keep %>% group_by(year, group) %>% nest()
> ## --- Residential ratio subsamples
> resid_nested = tokyo2keep      %>%    mutate(group = str_c("Resid_", ResidRatioSubsampleLbl))     %>%
+   group_by(year, group)  %>%    nest()
> ## --- 10km subsamples
> distance_nested = tokyo2keep    %>%
+   mutate(
+     group = str_c("Dist_", PalaceDistanceSubsampleLbl)
```

```
+   )  %>%
+   group_by(year, group)  %>%    nest()
> ## --- Subsamples with and without prewar hazard rating
> hazard_nested = tokyo2keep     %>%
+   mutate(group = str_c("HazardAvailable_", HazardAvailable))     %>%
+   group_by(year, group)  %>%    nest()
> ## ------------------------------------------------------------------------
> ## Combined and nested data
> tokyo_nested = resid_nested     %>%
+   bind_rows(hazard_nested)      %>%
+   bind_rows(distance_nested)    %>%
+   bind_rows(full_sample) %>%
+   select(year, group, data)
> ## ------------------------------------------------------------------------
> ## Subsample vector
> subsample_vec = tokyo_nested   %>%    pull(group)  %>%    unique()      ## Subsample indicators
> subsample_vec
[1] NA                    "Resid_(0.6,1]"       "Resid_(0.3,0.6]"      "Resid_[0,0.3]"
[5] "HazardAvailable_FALSE" "HazardAvailable_TRUE"  "Dist_<10km"           "Dist_>10km"
[9] "FullSample"
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
> ## ------------------------------------------------------------------------
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district    %>%    unique()
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Prepare output matrix
> ##
> ## ------------------------------------------------------------------------
> ## Output matrix # columns
> ncol_tmp_mtrx = 8
>
> ## create empty output matrix
> tmp_mtrx0 = matrix(NA, nrow = 30, ncol = ncol_tmp_mtrx)
>
> ## assing names to columns
> colnames(tmp_mtrx0)[1:ncol_tmp_mtrx] = c("DepVar",
```

```
+                                     "year",
+                                     "Sample",
+                                     "tau",
+                                     "se",
+                                     "ci_lower",
+                                     "ci_upper",
+                                     "N_obs")
>
> ## -----------------------------------------------------------------------
> ##
> ## Prepare graph parameters --- All DVs by year (annual subsamples), w/o 6F+ building
> ##
> ## -----------------------------------------------------------------------
> ## Figure parameters
> fig_height = 4.25
> base_cex = .85
> fig_mrgn = c(3.25, 2.4, 1.65, 1) + 0.1   ## figure margin: bottom, left, top, right
> plot_col = c("lightsteelblue4", "lightsteelblue1", "magenta2")
> pt_cex = .6
> polygon_alpha = 0.25
> x_tick = 1:30
> year_lbl = "Census Year"
> y_tick_lim = c(-4, 4)
> y_range = c(-0.15, 0.15)
> y_tick = seq(y_range[1], y_range[2], by = 0.05) %>% round(2)
> y_tick_sub = seq(-1, 1, by = 0.01) %>% round(2)
> y_tick_subsub = seq(-1, 1, by = 0.01)
> separation_linez = c(5, 10, 15, 20, 23, 25) + 0.5
> ## -----------------------------------------------------------------------
> ## Outcome labels
> outcome_pos = c(3, 8, 13, 18, 22, 24.5, 28)
> outcome_pos_1 = c(1, 3, 5, 7)
> outcome_pos_2 = c(2, 4, 6)
> outcome_lbl = c("% Unemp", "% Unemp (Male)", "% Unemp (Female)", "% Prof Exec", "Ave Res Yrs", "Ave Edu Yrs", "% Male <6
y/o")
> outcome_lbl = str_c("ln ", outcome_lbl)
> ## -----------------------------------------------------------------------
> ## Rectangle (border) width
> rect_width = 0.5
> rect_lwd = 0.5
> point_est_lwd = 1.25
> ## -----------------------------------------------------------------------
> ## Colors
```

```
> colz_vec = c("lightsteelblue4", "lightsteelblue4", "gray65", "magenta2")
> alpha_vec = c(0.6, 0.3, 1)
> ## ---------------------------------------------------------------------
>
>
>
>
> ## ---------------------------------------------------------------------
> ##
> ## Figure 6
> ## Result I: Socioeconomic results---GAM
> ##
> ## ---------------------------------------------------------------------
> ## Setting before loop
> ## ---------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
>
> ## initial row number for empty output matrix
> r = 1
>
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
>
> ## ---------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ---------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_gam %>% update(str_c(tmp_depvar, " ~ ."))## Update DV
+
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+
+     ## select year
+     tmp_year = vec_year[y]
+
+     ## Prepare temp data
```

```
+    tmp_data = tokyo_nested     %>%
+      filter(group=="FullSample") %>%
+      filter(
+        year == tmp_year
+      )      %>%
+      unnest(cols = c(data))     %>%
+      drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+
+    ## enter information to the output matrix
+    tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+    tmp_mtrx[r,2] = tmp_year                  ## Census year
+    tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+
+
+    ## Do estimation only if the data exist in a given year
+    if (nrow(tmp_data) > 0) {
+      ## ------------------------------------------------------------------------
+      ## Estimation
+      tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+
+      ## Record outputs
+      tmp_info = model_info_gam(tmp_est, x2extract = "ratio_damage")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "ratio_damage"  ## Add predictor (model specification)
+  )  %>%
+  drop_na()
>
>
```

```
> ## ------------------------------------------------------------------------
> ## Preparing figure
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+   mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+     point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+   ) %>%
+   mutate(row_id = row_number())
>
>
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
> tmp_figure_path = file.path(figure_dir, "fig_6.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
>  segments(x0  =  tmp_plot_dat$row_id-rect_width/2,  x1  =  tmp_plot_dat$row_id+rect_width/2,  y0  =  tmp_plot_dat$tau,  lwd  =
point_est_lwd)
>
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
```

```
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure 7
> ## Result I: Socioeconomic results---factorized damage 2010
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
>
> ## initial row number for empty output matrix
> r = 1
> m = 2 #GAM
>
```

```
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_alternative_treatment)) { ## Loop over outcomes
+   ## Set dependent variable
+   tmp_depvar = depvar_vec_alternative_treatment[i]
+   ## ------------------------------------------------------------------------
+   ## Loop over census years
+   ## Census year
+   tmp_year = 2010
+   ## Output matrix
+   tmp_mtrx = matrix(NA, nrow = 10, ncol = ncol_tmp_mtrx)
+   tmp_mtrx[,1] = i
+   tmp_mtrx[,2] = tmp_year
+   tmp_mtrx[,3] = rep(3, 10)
+   ## ------------------------------------------------------------------------
+   ## Loop over treatment specifications
+   ## Print progress
+         cat(str_c("Spatial   specification:   ",   spatial_term_lbl[m],   "   ---   dependent   variable:   ",   i,   "/",
length(depvar_vec_alternative_treatment), "¥r")); flush.console()
+   ## ------------------------------------------------------------------------
+   ## Pull out subset
+   tmp_data = tokyo_nested%>%
+     filter(group=="FullSample") %>%
+     filter(year == tmp_year) %>%
+     unnest(cols = c(data)) %>%
+     drop_na(all_of(tmp_depvar))
+   ## Treatment setting
+   tmp_treatment = "as.factor(ratio_damage)"
+   ## Set model specification
+   tmp_mod = model_list[[m]]  %>% update(str_c(". ~ . - ratio_damage + ", tmp_treatment)) %>%
+     update(str_c(tmp_depvar, " ~ ."))
+   ## Estimate and pull out coef and SEs
+   if (nrow(tmp_data) > 0) {
+     ## GAM
+     tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+     tmp_info = model_info_gam(tmp_est, x2extract = tmp_treatment)
+     ## Store estimates
+     tmp_mtrx[1:10,4:8] = cbind(tmp_info, nrow(tmp_data))   ## beta, SE, and CIs + N obs
+   }
+   ## ------------------------------------------------------------------------
+   ## Combine output matrix
+   if (i == 1) {
```

```
+     out_mtrx = tmp_mtrx
+   }  else   {
+     out_mtrx = rbind(out_mtrx, tmp_mtrx)
+   }
+ }    ## Treatment loop ends here
> atial specification: GAM --- dependent variable: 7/7
>
> ## -----------------------------------------------------------------------
> ## Modify outputs
> colnames(out_mtrx) = c("DepVar", "year", "Treatment", "tau", "se", "ci_lower", "ci_upper", "N_obs")
> out_dat = out_mtrx       %>%
+   as_tibble %>%
+   mutate(
+     DepVar = depvar_vec_alternative_treatment[DepVar],
+     Treatment = robust_treatment_vec[Treatment],
+     Predictor = as.character(tmp_mod[3])
+   )   %>%
+   drop_na()
>
>
> ## -----------------------------------------------------------------------
> ## Preparing figure
> ## -----------------------------------------------------------------------
> ## Figure parameters
> outcome_lbl_wo6f = outcome_lbl[!str_detect(outcome_lbl, "Apartment")]
> tmp_fig_height = 7
> tmp_fig_mrgn = c(7.75, 2.5, 2.25, .65) + 0.1    ## figure margin: bottom, left, top, right
> separation_linez_2010 = seq(10, 40, by = 10) + 0.5
> year_pos = seq(10, 50, by = 10) - 4.5
> ## Read estimates
> out_dat_master = out_dat
> ## Pull out DV (names)
> tmp_depvar_vec = out_dat_master$DepVar %>% unique
>
> ## -----------------------------------------------------------------------
> ## Summary plot for each census years
> ## -----------------------------------------------------------------------
> # tmp_depvar_vec = tmp_depvar_vec[!str_detect(tmp_depvar_vec, "lnrm_unemp|lnrf_unemp")]
> tmp_depvar_mat = tibble(DepVar = tmp_depvar_vec, FigLabel = outcome_lbl_wo6f[1:7])  %>%
+   ## Drop male/female unemployment rates
+   filter(
+     tmp_depvar_vec != "lnrm_unemp",
+     tmp_depvar_vec != "lnrf_unemp"
```

```
+   )
> ## Pull out estimates
> out_dat = out_dat_master   %>%
+   filter(str_detect(Treatment, "as.factor"))  %>%
+   drop_na %>%
+   mutate(
+     Treatment = str_c(Treatment, rep(seq(10, 100, by = 10), times = 7)),
+     Treatment = str_replace_all(Treatment, "as.factor¥¥(ratio_damage¥¥)", "RaidDamageDummy_")
+   ) %>%
+   filter(
+     year == 2010,
+     DepVar %in% tmp_depvar_mat$DepVar
+   )  %>% drop_na %>%
+   mutate(
+     row_id = row_number(),
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+     point_col = ifelse(ci_lower > 0 & ci_upper > 0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower < 0 & ci_upper < 0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col),
+     ## Dummy labels for figure
+     dummy_label = str_replace_all(Treatment, "RaidDamageDummy_", "Raid Damage ") %>% str_c("%")
+   )   %>%
+   left_join(tmp_depvar_mat)
Joining, by = "DepVar"
>
> ## ---------------------------------------------------------------------
> ## Creating Plot
> ## ---------------------------------------------------------------------
> tmp_figure_path = file.path(figure_dir, "fig_7.pdf")
> pdf(tmp_figure_path, width = plt_ratio(tmp_fig_height), height = tmp_fig_height)
> # dev.new(width = plt_ratio(tmp_fig_height), height = tmp_fig_height)
> x_range = c(0.25, (nrow(out_dat)+0.75))
> y_range_tmp = c(-0.15, 0.16)
> par(cex = base_cex, mar = tmp_fig_mrgn, lend = "square")
> plot(out_dat$tau, y = out_dat$row_id, ylim = y_range_tmp, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs = "i",
xaxs = "i", xlim = x_range)
> ## Separation lines between census years
> abline(v = separation_linez_2010)
> ## Rectangles
> abline(v = out_dat$row_id, col = colz_vec[3], lwd = .65, lty = "dotted")
>  rect(xleft  =  out_dat$row_id-rect_width/2,  xright  =  out_dat$row_id+rect_width/2,  ybottom  =  out_dat$ci_lower,  ytop  =
out_dat$ci_upper, col = out_dat$point_col, border = "black", lwd = rect_lwd)
> segments(out_dat$row_id-rect_width/2, x1 = out_dat$row_id+rect_width/2, y0 = out_dat$tau, lwd = point_est_lwd)
```

```
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(out_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez_2010, label = NA, tck = -0.3)
> ## --- Dummy label on the horizontal axis
> mtext(out_dat$dummy_label, side = 1, line = .5, at = out_dat$row_id, cex = 0.8, las = 2)
> ## --- DV label
> dv_lbl = out_dat   %>% pull(FigLabel)  %>% unique
> dv_lbl[1] = str_c("DV: ", dv_lbl[1])
> mtext(dv_lbl, side = 3, line = 0, at = year_pos, las = 1, cex = 1)   ## census year
> mtext(str_c("Year: 2010"), side = 3, line = 1.15, at = 0.2, las = 1, cex = 1.35, adj = 0)
> ## Vertical axis
> for (j in c(2,4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+   axis(j, line = 0, at = y_tick_subsub, label = NA, tck = -0.005)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 1)
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
> ## Close and save
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A2
> ## Descriptive Statistics (2 of 2)
> ## See 2_NinkaRegression_MainTxt&AppxA for the descriptive stats for
> ## the neighborhood association outcome variables and covariates.
> ##
> ## ------------------------------------------------------------------------
> ## Prepare the object
> ## ------------------------------------------------------------------------
> descriptive_tbl = tokyo2keep   %>%
+   select(
+     key_code, year, binary_damage_ratio,
+     all_of(depvar_vec), negative_cntrlz) %>%
```

```
+   filter(!is.na(binary_damage_ratio)) %>%
+   pivot_wider(names_from = year, values_from = c(depvar_vec, negative_cntrlz)) %>%
+   discard(~ all(is.na(.x)))     ## Some columns only available in limited years
Warning messages:
1: Using an external vector in selections was deprecated in tidyselect 1.1.0.
ℹ Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(negative_cntrlz)

  # Now:
  data %>% select(all_of(negative_cntrlz))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
2: Using an external vector in selections was deprecated in tidyselect 1.1.0.
ℹ Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(depvar_vec)

  # Now:
  data %>% select(all_of(depvar_vec))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> ## ------------------------------------------------------------------------
> ## Table A.2: Descriptive Statistics
> ## ------------------------------------------------------------------------
> tableContinuous(
+   vars = descriptive_tbl %>%    select(-key_code, -binary_damage_ratio) %>%    as.data.frame(),
+   group = descriptive_tbl$binary_damage_ratio,
+   stats = c("n", "mean", "s", "median", "iqr"),
+   prec = 3)
% latex table generated in R 4.2.2 by xtable 1.8-4 package
% Tue Apr 25 17:09:07 2023
¥begingroup¥footnotesize
¥begin{longtable}{llrrrrr}
 ¥textbf{Variable}  &  ¥textbf{Levels}  &  $¥mathbf{n}$  &  $¥mathbf{¥bar{x}}$  &  $¥mathbf{s}$  &  $¥mathbf{¥widetilde{x}}$  &
¥textbf{IQR} ¥¥
  ¥hline
lnr¥_unemp¥_1995 & 0 & 1126 & -2.940 & 0.273 & -2.919 & 0.306 ¥¥
   & 1 & 1024 & -2.938 & 0.351 & -2.911 & 0.365 ¥¥
```

```
   ¥hline
 & all & 2150 & -2.939 & 0.313 & -2.915 & 0.327 ¥¥
   ¥hline
lnr¥_unemp¥_2000 & 0 & 1128 & -2.958 & 0.275 & -2.938 & 0.291 ¥¥
   & 1 & 1030 & -2.955 & 0.312 & -2.924 & 0.326 ¥¥
   ¥hline
 & all & 2158 & -2.956 & 0.293 & -2.930 & 0.303 ¥¥
   ¥hline
lnr¥_unemp¥_2005 & 0 & 1129 & -2.816 & 0.310 & -2.830 & 0.355 ¥¥
   & 1 & 1029 & -2.798 & 0.343 & -2.780 & 0.378 ¥¥
   ¥hline
 & all & 2158 & -2.807 & 0.326 & -2.803 & 0.370 ¥¥
   ¥hline
lnr¥_unemp¥_2010 & 0 & 1128 & -2.758 & 0.250 & -2.751 & 0.299 ¥¥
   & 1 & 1030 & -2.779 & 0.300 & -2.754 & 0.345 ¥¥
   ¥hline
 & all & 2158 & -2.768 & 0.275 & -2.752 & 0.320 ¥¥
   ¥hline
lnr¥_unemp¥_2015 & 0 & 1127 & -3.109 & 0.254 & -3.089 & 0.290 ¥¥
   & 1 & 1028 & -3.152 & 0.301 & -3.119 & 0.352 ¥¥
   ¥hline
 & all & 2155 & -3.130 & 0.278 & -3.101 & 0.318 ¥¥
   ¥hline
lnrm¥_unemp¥_1995 & 0 & 1126 & -2.928 & 0.328 & -2.904 & 0.400 ¥¥
   & 1 & 1024 & -2.911 & 0.405 & -2.863 & 0.445 ¥¥
   ¥hline
 & all & 2150 & -2.920 & 0.367 & -2.885 & 0.423 ¥¥
   ¥hline
lnrm¥_unemp¥_2000 & 0 & 1128 & -2.937 & 0.322 & -2.919 & 0.364 ¥¥
   & 1 & 1030 & -2.930 & 0.378 & -2.883 & 0.392 ¥¥
   ¥hline
 & all & 2158 & -2.934 & 0.349 & -2.901 & 0.382 ¥¥
   ¥hline
lnrm¥_unemp¥_2005 & 0 & 1129 & -2.754 & 0.368 & -2.758 & 0.439 ¥¥
   & 1 & 1029 & -2.734 & 0.403 & -2.704 & 0.458 ¥¥
   ¥hline
 & all & 2158 & -2.744 & 0.385 & -2.731 & 0.449 ¥¥
   ¥hline
lnrm¥_unemp¥_2010 & 0 & 1128 & -2.685 & 0.308 & -2.670 & 0.366 ¥¥
   & 1 & 1030 & -2.706 & 0.376 & -2.661 & 0.424 ¥¥
   ¥hline
 & all & 2158 & -2.695 & 0.342 & -2.667 & 0.392 ¥¥
   ¥hline
```

```
lnrm¥_unemp¥_2015 & 0 & 1127 & -3.053 & 0.325 & -3.021 & 0.366 ¥¥
   & 1 & 1028 & -3.101 & 0.371 & -3.065 & 0.407 ¥¥
   ¥hline
 & all & 2155 & -3.076 & 0.348 & -3.042 & 0.395 ¥¥
   ¥hline
lnrf¥_unemp¥_1995 & 0 & 1126 & -2.986 & 0.317 & -2.935 & 0.293 ¥¥
   & 1 & 1024 & -3.014 & 0.372 & -2.958 & 0.369 ¥¥
   ¥hline
 & all & 2150 & -2.999 & 0.344 & -2.944 & 0.332 ¥¥
   ¥hline
lnrf¥_unemp¥_2000 & 0 & 1128 & -3.014 & 0.316 & -2.977 & 0.302 ¥¥
   & 1 & 1030 & -3.024 & 0.326 & -2.986 & 0.344 ¥¥
   ¥hline
 & all & 2158 & -3.019 & 0.321 & -2.981 & 0.321 ¥¥
   ¥hline
lnrf¥_unemp¥_2005 & 0 & 1129 & -2.938 & 0.332 & -2.915 & 0.317 ¥¥
   & 1 & 1029 & -2.922 & 0.347 & -2.887 & 0.340 ¥¥
   ¥hline
 & all & 2158 & -2.930 & 0.339 & -2.903 & 0.326 ¥¥
   ¥hline
lnrf¥_unemp¥_2010 & 0 & 1128 & -2.889 & 0.264 & -2.867 & 0.280 ¥¥
   & 1 & 1030 & -2.921 & 0.301 & -2.895 & 0.320 ¥¥
   ¥hline
 & all & 2158 & -2.904 & 0.283 & -2.882 & 0.296 ¥¥
   ¥hline
lnrf¥_unemp¥_2015 & 0 & 1127 & -3.218 & 0.280 & -3.196 & 0.304 ¥¥
   & 1 & 1028 & -3.262 & 0.333 & -3.227 & 0.363 ¥¥
   ¥hline
 & all & 2155 & -3.239 & 0.307 & -3.208 & 0.328 ¥¥
   ¥hline
lnr¥_proexe¥_1995 & 0 & 1126 & -1.523 & 0.329 & -1.489 & 0.475 ¥¥
   & 1 & 1026 & -1.576 & 0.352 & -1.591 & 0.455 ¥¥
   ¥hline
 & all & 2152 & -1.549 & 0.341 & -1.545 & 0.482 ¥¥
   ¥hline
lnr¥_proexe¥_2000 & 0 & 1128 & -1.531 & 0.322 & -1.504 & 0.471 ¥¥
   & 1 & 1030 & -1.581 & 0.347 & -1.597 & 0.430 ¥¥
   ¥hline
 & all & 2158 & -1.555 & 0.335 & -1.554 & 0.475 ¥¥
   ¥hline
lnr¥_proexe¥_2005 & 0 & 1129 & -1.509 & 0.308 & -1.474 & 0.422 ¥¥
   & 1 & 1029 & -1.542 & 0.334 & -1.557 & 0.419 ¥¥
   ¥hline
```

```
 & all & 2158 & -1.525 & 0.321 & -1.515 & 0.426 ¥¥
  ¥hline
lnr¥_proexe¥_2010 & 0 & 1128 & -1.513 & 0.278 & -1.484 & 0.373 ¥¥
  & 1 & 1030 & -1.527 & 0.292 & -1.545 & 0.368 ¥¥
  ¥hline
 & all & 2158 & -1.520 & 0.285 & -1.516 & 0.374 ¥¥
  ¥hline
lnr¥_proexe¥_2015 & 0 & 1127 & -1.415 & 0.253 & -1.385 & 0.328 ¥¥
  & 1 & 1028 & -1.424 & 0.268 & -1.422 & 0.329 ¥¥
  ¥hline
 & all & 2155 & -1.420 & 0.260 & -1.403 & 0.333 ¥¥
  ¥hline
lnave¥_lvlength¥_2000 & 0 & 1128 &  2.699 & 0.197 &  2.681 & 0.212 ¥¥
  & 1 & 1030 &  2.711 & 0.229 &  2.724 & 0.233 ¥¥
  ¥hline
 & all & 2158 &  2.705 & 0.213 &  2.703 & 0.231 ¥¥
  ¥hline
lnave¥_lvlength¥_2010 & 0 & 1128 &  2.708 & 0.172 &  2.713 & 0.197 ¥¥
  & 1 & 1030 &  2.705 & 0.196 &  2.726 & 0.205 ¥¥
  ¥hline
 & all & 2158 &  2.707 & 0.184 &  2.720 & 0.201 ¥¥
  ¥hline
lnave¥_lvlength¥_2015 & 0 & 1127 &  2.740 & 0.164 &  2.746 & 0.172 ¥¥
  & 1 & 1028 &  2.713 & 0.186 &  2.734 & 0.201 ¥¥
  ¥hline
 & all & 2155 &  2.727 & 0.175 &  2.740 & 0.186 ¥¥
  ¥hline
lnave¥_eduyr¥_2000 & 0 & 1128 &  2.599 & 0.060 &  2.609 & 0.092 ¥¥
  & 1 & 1030 &  2.584 & 0.057 &  2.583 & 0.075 ¥¥
  ¥hline
 & all & 2158 &  2.592 & 0.059 &  2.596 & 0.087 ¥¥
  ¥hline
lnave¥_eduyr¥_2010 & 0 & 1128 &  2.644 & 0.058 &  2.658 & 0.087 ¥¥
  & 1 & 1030 &  2.637 & 0.055 &  2.639 & 0.074 ¥¥
  ¥hline
 & all & 2158 &  2.640 & 0.057 &  2.647 & 0.083 ¥¥
  ¥hline
lnrmale0004¥_1995 & 0 & 1112 & -0.686 & 0.139 & -0.674 & 0.141 ¥¥
  & 1 & 1014 & -0.681 & 0.162 & -0.675 & 0.153 ¥¥
  ¥hline
 & all & 2126 & -0.684 & 0.150 & -0.674 & 0.146 ¥¥
  ¥hline
lnrmale0004¥_2000 & 0 & 1128 & -0.680 & 0.134 & -0.676 & 0.132 ¥¥
```

```
   & 1 & 1030 & -0.686 & 0.165 & -0.672 & 0.157 ¥¥
   ¥hline
 & all & 2158 & -0.683 & 0.150 & -0.675 & 0.144 ¥¥
   ¥hline
lnrmale0004¥_2005 & 0 & 1120 & -0.682 & 0.135 & -0.672 & 0.138 ¥¥
   & 1 & 1022 & -0.687 & 0.166 & -0.674 & 0.151 ¥¥
   ¥hline
 & all & 2142 & -0.684 & 0.151 & -0.674 & 0.144 ¥¥
   ¥hline
lnrmale0004¥_2010 & 0 & 1128 & -0.681 & 0.118 & -0.674 & 0.135 ¥¥
   & 1 & 1030 & -0.683 & 0.147 & -0.679 & 0.152 ¥¥
   ¥hline
 & all & 2158 & -0.682 & 0.133 & -0.676 & 0.141 ¥¥
   ¥hline
lnrmale0004¥_2015 & 0 & 1120 & -0.680 & 0.110 & -0.680 & 0.132 ¥¥
   & 1 & 1026 & -0.686 & 0.128 & -0.675 & 0.145 ¥¥
   ¥hline
 & all & 2146 & -0.683 & 0.119 & -0.678 & 0.138 ¥¥
   ¥hline
¥hline
¥caption{}
¥label{}
¥end{longtable}
¥endgroup
> ## ------------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A4
> ## Result I: Socioeconomic results---5th poly
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
```

```
>
> ## initial row number for empty output matrix
> r = 1
>
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
>
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_polynomial %>% update(str_c(tmp_depvar, " ~ .")) ## Update DV
+
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+
+     ## select year
+     tmp_year = vec_year[y]
+
+     ## Prepare temp data
+     tmp_data = tokyo_nested     %>%
+       filter(group=="FullSample") %>%
+       filter(
+         year == tmp_year
+       )      %>%
+       unnest(cols = c(data))     %>%
+       drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+
+     ## enter information to the output matrix
+     tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+     tmp_mtrx[r,2] = tmp_year                  ## Census year
+     tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+
+
+     ## Do estimation only if the data exist in a given year
+     if (nrow(tmp_data) > 0) {
+       ## ------------------------------------------------------------------
+       ## Estimation
+       tmp_mod_woFE = update(tmp_mod, .~. - as.factor(prewar_district))
+       tmp_est = lm_robust(tmp_mod_woFE, data = tmp_data, weights = tmp_data$pop, fixed_effects = ~prewar_district)
```

```
+      ## Record outputs
+      tmp_info = model_info_robustlm(tmp_est, x2extract = "ratio_damage")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "ratio_damage"  ## Add predictor (model specification)
+  )  %>%
+  drop_na()
>
>
> ## ------------------------------------------------------------------------
> ## Preparing figure
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+  mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+    point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+    point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+  ) %>%
+  mutate(row_id = row_number())
>
>
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
```

```
> tmp_figure_path = file.path(figure_dir, "fig_a4.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
> segments(x0 = tmp_plot_dat$row_id-rect_width/2, x1 = tmp_plot_dat$row_id+rect_width/2, y0 = tmp_plot_dat$tau, lwd =
point_est_lwd)
>
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> ## Close device
```

```
> dev.off()
null device
        1
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A5
> ## Result I: Socioeconomic results---GAM, binary_damage_ratio
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
>
> ## initial row number for empty output matrix
> r = 1
>
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
>
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_gam %>% update(str_c(". ~ . - ratio_damage + binary_damage_ratio")) %>%
+     update(str_c(tmp_depvar, " ~ ."))
+
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+
+     ## select year
+     tmp_year = vec_year[y]
+
```

```
+    ## Prepare temp data
+    tmp_data = tokyo_nested     %>%
+      filter(group=="FullSample") %>%
+      filter(
+        year == tmp_year
+      )      %>%
+      unnest(cols = c(data))     %>%
+      drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+
+    ## enter information to the output matrix
+    tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+    tmp_mtrx[r,2] = tmp_year                   ## Census year
+    tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+
+
+    ## Do estimation only if the data exist in a given year
+    if (nrow(tmp_data) > 0) {
+      ## ------------------------------------------------------------------------
+      ## Estimation
+      tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+
+      ## Record outputs
+      tmp_info = model_info_gam(tmp_est, x2extract = "binary_damage_ratio")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "binary_damage_ratio"  ## Add predictor (model specification)
+  )  %>%
+  drop_na()
>
```

```
> ## ------------------------------------------------------------------------
> ## Preparing figure
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+   mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+     point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+   ) %>%
+   mutate(row_id = row_number())
>
>
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
> tmp_figure_path = file.path(figure_dir, "fig_a5.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
>  #  tmp_figure_path  =  file.path(tmp_dir,  str_c("AllDepVars_6Fdropped_",  logged_or_not[m],  "_Sample_",  tmp_sample_lbl,
".png"))
> # png(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height, res = 480, units = "in")
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
>  segments(x0  =  tmp_plot_dat$row_id-rect_width/2,  x1  =  tmp_plot_dat$row_id+rect_width/2,  y0  =  tmp_plot_dat$tau,  lwd  =
point_est_lwd)
>
```

```
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A6
> ## Result I: Socioeconomic results---GAM, subsample >70% residential
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
```

```
>
> ## initial row number for empty output matrix
> r = 1
>
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
>
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_gam %>% update(str_c(tmp_depvar, " ~ .")) ## Update DV
+
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+
+     ## select year
+     tmp_year = vec_year[y]
+
+     ## Prepare temp data
+     tmp_data = tokyo_nested     %>%
+       filter(group=="Resid_(0.6,1]") %>%
+       filter(
+         year == tmp_year
+       )      %>%
+       unnest(cols = c(data))     %>%
+       drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+
+     ## enter information to the output matrix
+     tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+     tmp_mtrx[r,2] = tmp_year                  ## Census year
+     tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+
+
+     ## Do estimation only if the data exist in a given year
+     if (nrow(tmp_data) > 0) {
+       ## ------------------------------------------------------------------
+       ## Estimation
+       tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+
```

```
+      ## Record outputs
+      tmp_info = model_info_gam(tmp_est, x2extract = "ratio_damage")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "ratio_damage"  ## Add predictor (model specification)
+  )  %>%
+  drop_na()
>
>
> ## ------------------------------------------------------------------------
> ## Preparing figure
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+  mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+    point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+    point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+  ) %>%
+  mutate(row_id = row_number())
>
>
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
```

```
> tmp_figure_path = file.path(figure_dir, "fig_a6.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
>  #  tmp_figure_path  =  file.path(tmp_dir,  str_c("AllDepVars_6Fdropped_",  logged_or_not[m],  "_Sample_",  tmp_sample_lbl,
".png"))
> # png(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height, res = 480, units = "in")
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
>  segments(x0  =  tmp_plot_dat$row_id-rect_width/2,  x1  =  tmp_plot_dat$row_id+rect_width/2,  y0  =  tmp_plot_dat$tau,  lwd  =
point_est_lwd)
>
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
```

```
> box()
>
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A7
> ## Result I: Socioeconomic results---GAM, within 10km radius
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
>
> ## initial row number for empty output matrix
> r = 1
>
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
>
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_gam %>% update(str_c(tmp_depvar, " ~ ."))## Update DV
+
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+
+     ## select year
```

```
+    tmp_year = vec_year[y]
+
+    ## Prepare temp data
+    tmp_data = tokyo_nested     %>%
+      filter(group=="Dist_<10km") %>%
+      filter(
+        year == tmp_year
+      )      %>%
+      unnest(cols = c(data))     %>%
+      drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+
+    ## enter information to the output matrix
+    tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+    tmp_mtrx[r,2] = tmp_year                  ## Census year
+    tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+
+
+    ## Do estimation only if the data exist in a given year
+    if (nrow(tmp_data) > 0) {
+      ## ------------------------------------------------------------------------
+      ## Estimation
+      tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+
+      ## Record outputs
+      tmp_info = model_info_gam(tmp_est, x2extract = "ratio_damage")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "ratio_damage"  ## Add predictor (model specification)
+  )  %>%
```

```
+   drop_na()
>
> ## ------------------------------------------------------------------------
> ## Preparing figure
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+   mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+     point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+   ) %>%
+   mutate(row_id = row_number())
>
>
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
> tmp_figure_path = file.path(figure_dir, "fig_a7.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
>  #  tmp_figure_path  =  file.path(tmp_dir,  str_c("AllDepVars_6Fdropped_",  logged_or_not[m],  "_Sample_",  tmp_sample_lbl,
".png"))
> # png(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height, res = 480, units = "in")
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
>  segments(x0  =  tmp_plot_dat$row_id-rect_width/2,  x1  =  tmp_plot_dat$row_id+rect_width/2,  y0  =  tmp_plot_dat$tau,  lwd  =
```

```
point_est_lwd)
>
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A8: Socioeconomic results---GAM, ctrl 5th poly of prewar cov
> ##
> ## ------------------------------------------------------------------------
> ## Prepare variables
> ## ------------------------------------------------------------------------
> ## Prewar additional terms
> prewarcov_plus_base = c("ratio_residential", "PopDensity_1939_km2_ln")
>
```

```
> prewarcov_plus_poly = c(
+   polynomial_varname(prewarcov_plus_base, degree = polynomial_degree),
+   "I(ratio_residential*PopDensity_1939_km2_ln)",
+   "I(ratio_residential^2*PopDensity_1939_km2_ln)",
+   "I(ratio_residential*PopDensity_1939_km2_ln^2)",
+   "I(ratio_residential^2*PopDensity_1939_km2_ln^2)",
+   "I(ratio_residential^3*PopDensity_1939_km2_ln)",
+   "I(ratio_residential*PopDensity_1939_km2_ln^3)",
+   "I(ratio_residential^4*PopDensity_1939_km2_ln)",
+   "I(ratio_residential*PopDensity_1939_km2_ln^4)",
+   "I(ratio_residential^3*PopDensity_1939_km2_ln^2)",
+   "I(ratio_residential^2*PopDensity_1939_km2_ln^3)"
+ )
>
> ## ---------------------------------------------------------------------
> ## Prepare models
> ## ---------------------------------------------------------------------
> ## model = GAM
> ## add.feature = prewarcov_plus
> base_gam_prewarcov_plus_poly = chr2fml("outcome",
+                                        idv_list = list(
+                                          "ratio_damage",
+                                          spatial_covariates,
+                                          district_fe,
+                                          distance_term_specification[[2]],
+                                          spatial_term_specification[[2]],
+                                          prewarcov_plus_poly
+                                        ))
>
> ## ---------------------------------------------------------------------
> ## Prepare data obj
> ## ---------------------------------------------------------------------
> ## --- Adding newly defined covariates to the dataset
> full_sample = tokyo2keep %>%
+   drop_na(ratio_damage, pop)    %>%
+   mutate_at(vars(all_of(prewarcov_plus_base)), list(scale2)) %>%
+   group_by(year)   %>%
+   nest
>
> ## ---------------------------------------------------------------------
> ## Analysis
> ## ---------------------------------------------------------------------
> ## Setting before loop
```

```
> ## ------------------------------------------------------------------------
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
> 
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
> 
> ## initial row number for empty output matrix
> r = 1
> 
> ## create empty output matrix
> tmp_mtrx = tmp_mtrx0
> 
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+ 
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   # tmp_mod = base_gam %>%      update(str_c(tmp_depvar, " ~ .")) ## Update DV
+   tmp_mod = base_gam_prewarcov_plus_poly %>% update(str_c(tmp_depvar, " ~ .")) ## Update DV
+ 
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+ 
+     ## select year
+     tmp_year = vec_year[y]
+ 
+     ## Prepare temp data
+     tmp_data = full_sample      %>%
+       filter(
+         year == tmp_year
+       )      %>%
+       unnest(cols = c(data))     %>%
+       drop_na(all_of(tmp_depvar))      ## unnest and drop NAs in DV
+ 
+     ## enter information to the output matrix
+     tmp_mtrx[r,1] = i                                    ## Dependent variable indicator
+     tmp_mtrx[r,2] = tmp_year                  ## Census year
+     tmp_mtrx[r,3] = "FullSample" ## (Sub)sample indicator
+ 
+ 
```

```
+    ## Do estimation only if the data exist in a given year
+    if (nrow(tmp_data) > 0) {
+      ## ------------------------------------------------------------------------
+      ## Estimation
+      tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+
+      ## Record outputs
+      tmp_info = model_info_gam(tmp_est, x2extract = "ratio_damage")
+      tmp_mtrx[r,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+      ## update row number
+      r = r+1
+
+    } ## if ends here
+  } ## year loop ends here
+ } ## outcome loop ends here
>
> ## format outputs for plot
> out_dat = as.data.frame(tmp_mtrx) %>%
+  ## set as numeric
+  mutate_at(vars(-Sample), as.numeric) %>%
+  as_tibble %>%
+  mutate(
+    DepVar = depvar_vec_combined[as.numeric(DepVar)],      ## Dependent variable label
+    Predictor = "ratio_damage"  ## Add predictor (model specification)
+  )  %>%
+  drop_na()
>
>
> ## ------------------------------------------------------------------------
> ## Plot estimates --- All DVs by year (annual subsamples)
> ## ------------------------------------------------------------------------
> ## Pull out DV names
> tmp_depvar_vec = out_dat$DepVar %>% unique
>
> ## Prepare estimates
> tmp_plot_dat = out_dat   %>%
+  mutate(
+         point_col  =  ifelse(tau>0,  adjustcolor(colz_vec[1],  alpha  =  alpha_vec[2]),  adjustcolor(colz_vec[2],  alpha  =
alpha_vec[2])),
+    point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+    point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col)
+  ) %>%
+  mutate(row_id = row_number())
```

```
>
>
> ## Creating Plot
> tmp_figure_path = file.path(figure_dir, "fig_a8.pdf")
> pdf(tmp_figure_path, width = plt_ratio(fig_height), height = fig_height)
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(tmp_plot_dat)+0.75))
> plot(x = tmp_plot_dat$row_id, y = tmp_plot_dat$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs
= "i", xaxs = "i", xlim = x_range)
>
> ## Background for male ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
>
> ## Separation lines between DVs
> abline(v = separation_linez)
> abline(v = 1:nrow(tmp_plot_dat), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
>
> ## Rectangles
> rect(xleft = tmp_plot_dat$row_id-rect_width/2, xright = tmp_plot_dat$row_id+rect_width/2, ybottom = tmp_plot_dat$ci_lower,
ytop = tmp_plot_dat$ci_upper, col = tmp_plot_dat$point_col, border = "black", lwd = rect_lwd)
> segments(x0 = tmp_plot_dat$row_id-rect_width/2, x1 = tmp_plot_dat$row_id+rect_width/2, y0 = tmp_plot_dat$tau, lwd =
point_est_lwd)
>
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(tmp_plot_dat), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(tmp_plot_dat$year, side = 1, line = 0.5, at = 1:nrow(tmp_plot_dat), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
>
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
>
> ## Showing outcome variables in rows.
> mtext(outcome_lbl[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
>
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
```

```
> box()
>
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure A9: Socioeconomic results---GAM, unit zipcode
> ##
> ## ------------------------------------------------------------------------
> ## Data
> census_zipcode_tbl = data_dir  %>%    file.path("census_zipcode_data.csv")    %>%    read_csv(col_types = cols())
> ## Outcomes
> depvar_vec_tmp = c(depvar_vec, negative_cntrlz)
> ## Prepare list obj for baseline result table
> baseline_2010_gam = list_along(depvar_vec_tmp)
> names(baseline_2010_gam) = depvar_vec_tmp
> baseline_2010_wls = baseline_2010_gam
>
> ## ------------------------------------------------------------------------
> ## Loop over spatial settings --- polynomials, spline, and distance variable spline
> ## Loop over outcomes (indexed by i)
> for (i in seq_along(depvar_vec_tmp)) {
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_tmp[i]
+   tmp_mod = model_list[[2]]     %>%    update(str_c(tmp_depvar, " ~ .")) ## Update DV
+   ## ---------------------------------------------------------------------
+   ## Loop over census years
+   for (y in seq_along(vec_year)) {
+     ## Census year
+     tmp_year = vec_year[y]
+     ## Prepare output matrix: tau, SE, etc.
+     tmp_mtrx = matrix(NA, nrow = 1, ncol = ncol_tmp_mtrx)
+     tmp_mtrx[,1] = i                                    ## Dependent variable indicator
+     tmp_mtrx[,2] = tmp_year                             ## Census year
+     ## Prepare output matrix: EDFs
+     tmp_n_spline = str_count(as.character(tmp_mod)[3], "s¥¥(|te¥¥(")    %>%    as.numeric + 1      ##  +1  for  the  geo-
```

```
coordinate spline
+     tmp_edf_mtrx = matrix(NA, nrow = 1, ncol = tmp_n_spline)
+     ## ------------------------------------------------------------------------
+     ## Prepare subsample
+     tmp_data = census_zipcode_tbl      %>%    filter(year == tmp_year)   %>%    drop_na(all_of(tmp_depvar))
+     ## ------------------------------------------------------------------------
+     ## Estimate and pull out coefficients and SEs
+     if (nrow(tmp_data) > 0) {
+
+       ## GAM
+       tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+       tmp_info = model_info_gam(tmp_est, x2extract = treatment_vec[1])
+       ## Extract the effective degrees of freedom
+       tmp_edf = pen.edf(tmp_est)
+       tmp_edf_mtrx[1,] = tmp_edf
+       colnames(tmp_edf_mtrx) = names(tmp_edf)
+       ## Store estimates
+       tmp_mtrx[1,4:8] = c(tmp_info, nrow(tmp_data))  ## beta, SE, and CIs + N obs
+     }
+     ## ------------------------------------------------------------------------
+     ## Combine output matrix
+     tmp_mtrx = cbind(tmp_mtrx, tmp_edf_mtrx)
+     if (y == 1 & i == 1) {
+       out_mtrx = tmp_mtrx
+     } else   {
+       out_mtrx = rbind(out_mtrx, tmp_mtrx)
+     }
+   }  ## year loop ends here
+ }    ## outcome loop ends here
> ## ------------------------------------------------------------------------
> ## Modify outputs
> colnames(out_mtrx)[1:ncol_tmp_mtrx] = c("DepVar", "year", "Sample", "tau", "se", "ci_lower", "ci_upper", "N_obs")
> estimate_tbl = out_mtrx  %>%    as_tibble()   %>%
+   mutate(
+     Sample = "zipcode_polygon",
+     DepVar = depvar_vec_tmp[DepVar],   ## Dependent variable label
+     Predictor = as.character(tmp_mod[3])      ## Add predictor (model specification)
+   )  %>%
+   drop_na()
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Preparing Figure
```

```
> ## ------------------------------------------------------------------------
> ## Figure parameters
> fig_height = 4.25
> base_cex = .85
> fig_mrgn = c(3.25, 2.4, 1.65, 1) + 0.1       ## figure margin: bottom, left, top, right
> y_range = c(-0.2, 0.2)
> y_tick = seq(y_range[1], y_range[2], by = 0.05) %>% round(2)
> y_tick_sub = seq(-1, 1, by = 0.01)    %>%    round(2)
> separation_linez = c(5, 10, 15, 20, 23, 25) + 0.5
> ## ------------------------------------------------------------------------
> ## Outcome labels
> outcome_pos = c(3, 8, 13, 18, 22, 24.5, 28)
> outcome_pos_1 = c(1, 3, 5, 7)
> outcome_pos_2 = c(2, 4, 6)
> outcome_lbl_wo6f = outcome_lbl[!str_detect(outcome_lbl, "Apartment")]
> ## ------------------------------------------------------------------------
> ## Read estimates
> csv_path_vec = estimate_tbl
> ## ------------------------------------------------------------------------
> ## Loop over models: WLS and GAM
> ## Pull out regression estimates
> estimate_tbl = estimate_tbl    %>%
+   mutate(
+     row_id = row_number(),
+        point_col = ifelse(tau > 0, adjustcolor(colz_vec[1], alpha = alpha_vec[2]), adjustcolor(colz_vec[2], alpha =
alpha_vec[2])),
+     point_col = ifelse(ci_lower > 0 & ci_upper > 0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower < 0 & ci_upper < 0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col))
> ## ------------------------------------------------------------------------
> ## Creating Plot
> ## ------------------------------------------------------------------------
> ## Open device
> pdf(
+   figure_dir      %>%    file.path("fig_a9.pdf"),
+   width = plt_ratio(fig_height), height = fig_height)
> ## Plot
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(estimate_tbl) + 0.75))
> plot(estimate_tbl$row_id, y = estimate_tbl$tau, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n", yaxs =
"i", xaxs = "i", xlim = x_range)
> ## Background for the boy ratio
> rect(xleft = separation_linez[6], xright = 40, ytop = 1, ybottom = -1, col = adjustcolor("gray", alpha = 0.2))
> ## Separation lines between DVs
```

```
> abline(v = separation_linez)
> abline(v = 1:nrow(estimate_tbl), lwd = 0.35, lty = "dotted", col = colz_vec[3]) ## horizontal guides
> ## Rectangles
>   rect(xleft   =   estimate_tbl$row_id   -   rect_width/2,   xright   =   estimate_tbl$row_id   +   rect_width/2,   ybottom   =
estimate_tbl$ci_lower, ytop = estimate_tbl$ci_upper, col = estimate_tbl$point_col, border = "black", lwd = rect_lwd)
> segments(x0 = estimate_tbl$row_id - rect_width/2, x1 = estimate_tbl$row_id + rect_width/2, y0 = estimate_tbl$tau, lwd =
point_est_lwd)
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(estimate_tbl), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -0.115)
> mtext(estimate_tbl$year, side = 1, line = 0.5, at = 1:nrow(estimate_tbl), las = 2, cex = 0.85)
> mtext(year_lbl, side = 1, line = 2.45, cex = 0.8)
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
> mtext(outcome_lbl_wo6f[outcome_pos_1], side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex = 0.85)
> mtext(outcome_lbl_wo6f[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.85)
> mtext("Raid Effect (95% CIs)", side = 2, line = 1.65, cex = 0.8)
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
> ## Close device
> dev.off()
null device
          1
> ## ------------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table A3
> ## Result: Socioeconomic results---GAM 2010
> ##
> ## ------------------------------------------------------------------------
> ## Setting before loop
> ## ------------------------------------------------------------------------
> ## Prepare list obj for baseline result table
> baseline_2010_gam = list_along(depvar_vec_combined)
```

```
> names(baseline_2010_gam) = depvar_vec_combined
>
> ## Census year loop
> vec_year = seq(1995, 2015, by = 5)    ## Census years
>
> ## District vector: Unique (35) districts
> vec_district = tokyo2keep$prewar_district  %>% unique
>
> ## initial row number for empty output matrix
> r = 1
>
> ## ------------------------------------------------------------------------
> ## Loop over outcomes (indexed by i)
> ## ------------------------------------------------------------------------
> for (i in seq_along(depvar_vec_combined)) {
+   ## Set dependent variable and model specification
+   tmp_depvar = depvar_vec_combined[i]
+   tmp_mod = base_gam %>% update(str_c(tmp_depvar, " ~ .")) ## Update DV
+
+   ## select year
+   tmp_year = 2010
+
+   ## Prepare temp data
+   tmp_data = tokyo_nested%>%
+     filter(group=="FullSample") %>%
+     filter(
+       year == tmp_year
+     ) %>%
+     unnest(cols = c(data))      %>%
+     drop_na(all_of(tmp_depvar)) ## unnest and drop NAs in DV
+
+   ## Do estimation only if the data exist in a given year
+   if (nrow(tmp_data) > 0) {
+     ## ------------------------------------------------------------------------
+     ## Estimation
+     tmp_est = gam(tmp_mod, data = tmp_data, weights = tmp_data$pop)
+     ## Store models 2010
+     baseline_2010_gam[[i]] = tmp_est
+     ## update row number
+     r = r+1
+
+   } ## if ends here
+ } ## outcome loop ends here
```

```
>
> ## ------------------------------------------------------------------------
> ## <<<----<<< Table: Baseline (2010 General Census) estimates >>>---->>>
> ## ------------------------------------------------------------------------
> ## Table for baseline regression estimates, 2010
> gam_2010 = baseline_2010_gam
> ## Pull out DVs to be reported
> gam_2010 = gam_2010[str_detect(names(gam_2010), "^ln")]          ## Logged DVs
> gam_2010 = gam_2010[!str_detect(names(gam_2010), "kyodojutaku6up")]     ## Drop 6F+ ratio
> names(gam_2010)
[1] "lnr_unemp"     "lnrm_unemp"    "lnrf_unemp"    "lnr_proexe"    "lnave_lvlength" "lnave_eduyr"
[7] "lnrmale0004"
> ## Compute sample means (raw (non-logged) mean values)
> trans_varz = c("lnr_unemp", "lnrm_unemp", "lnrf_unemp", "lnr_proexe")
> mean_tbl = gam_2010      %>%
+   ## Pull out data matrix from model objects
+   map(extract_vec("model"))     %>%    map(as_tibble)      %>%
+   ## Pull out DV vectors
+   map(~.x %>% pull(1))    %>%
+   ## Transform back to raw scale
+   map(exp)  %>%
+   enframe(name = "Outcome", value = "Value") %>%
+   group_by(Outcome)      %>%
+   ## Transform values
+   mutate(
+     ValueRaw = if_else(
+       Outcome %in% trans_varz,
+       lapply(Value, function(x) {x = x -0.01}),
+       Value)
+   )  %>%
+   ## Means and logged means
+   mutate(
+     Mean_Value = Value    %>%    map(mean),
+     Mean_ValueRaw = ValueRaw     %>%    map(mean)
+   )  %>%
+   unnest(c(Mean_Value, Mean_ValueRaw)) %>%
+   mutate(ln_Mean_ValueRaw = log(Mean_ValueRaw))
> mean_tbl
# A tibble: 7 × 6
# Groups:  Outcome [7]
  Outcome       Value        ValueRaw     Mean_Value Mean_ValueRaw ln_Mean_ValueRaw
  <chr>         <list>       <list>           <dbl>       <dbl>          <dbl>
1 lnr_unemp     <dbl [2,158]> <dbl [2,158]>    0.0651      0.0551         -2.90
```

```
2 lnrm_unemp    <dbl [2,158]> <dbl [2,158]>    0.0712       0.0612         -2.79
3 lnrf_unemp    <dbl [2,158]> <dbl [2,158]>    0.0568       0.0468         -3.06
4 lnr_proexe    <dbl [2,158]> <dbl [2,158]>    0.228        0.218          -1.52
5 lnave_lvlength <dbl [2,158]> <dbl [2,158]>   15.2         15.2            2.72
6 lnave_eduyr   <dbl [2,158]> <dbl [2,158]>   14.0         14.0            2.64
7 lnrmale0004   <dbl [2,158]> <dbl [2,158]>    0.510        0.510          -0.674
>
> ## ------------------------------------------------------------------
> ## Stargazer
> stargazer(
+   gam_2010,
+   type = "text",
+   keep = c("ratio_damage"),
+   single.row = FALSE, digits = 4,
+   align = TRUE, model.names = FALSE, model.numbers = FALSE,
+   dep.var.labels.include = TRUE
+ )

=============================================================================================
                                      Dependent variable:
             ------------------------------------------------------------------------------
             lnr_unemp lnrm_unemp lnrf_unemp lnr_proexe lnave_lvlength lnave_eduyr lnrmale0004
-------------------------------------------------------------------------------------------
ratio_damage   0.0028   -0.0016    0.0073   -0.0553***   -0.0551***   -0.0060**   -0.0019
             (0.0164)  (0.0206)  (0.0180)  (0.0146)    (0.0127)    (0.0027)   (0.0095)

-------------------------------------------------------------------------------------------
Observations    2,158    2,158     2,158     2,158       2,158       2,158      2,158
Adjusted R2    0.4780    0.4576    0.2890    0.6903      0.3818      0.7565     0.0076
Log Likelihood 553.5228  59.9267   355.4536  792.3942   1,096.2390  4,461.0590 1,602.6820
UBRE          88.8229  140.3294  106.7176  71.1731     53.7028      2.3752     33.5652
=============================================================================================
Note:                                                     *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------------
> ##
> ## Table A8
> ## Result: Socioeconomic results---GAM, ctrl NE damage
> ##
```

```
> ## ------------------------------------------------------------------------
> ## Use object "gam_2010" from the above analysis.
> ## ------------------------------------------------------------------------
> ## Prepare vectors
> key_code_vec = tokyo2keep      %>%    filter(year == 2010) %>%
+   drop_na(ratio_damage) %>%
+   pull(key_code)
> pop2010_vec = pop = tokyo2keep %>%    filter(year == 2010) %>%
+   drop_na(ratio_damage) %>%
+   pull(pop)
> ## ------------------------------------------------------------------------
> ## Pull out residuals as a tibble object
> DV_lblz = names(gam_2010)
> gam_residualz_baseline = gam_2010     %>%    map(residuals, type = "response") %>%
+   enframe(name = "Outcome", value = "Residuals")    %>%
+   unnest(cols = c(Residuals))   %>%
+   ## Dummy ID for spread() function
+   mutate(
+     key_code = as.character(rep(key_code_vec, length(gam_2010))),
+     population2010 = rep(pop2010_vec, length(gam_2010))
+   )  %>%
+   spread(key = Outcome, value = Residuals)   %>%
+   select(key_code, population2010, all_of(DV_lblz))
> ## ------------------------------------------------------------------------
> ## Load and prepare polygon (moved above)
> ## --- Full polygons
> full_poly = data_dir %>%
+   file.path("RaidShp_May2020b.rds") %>%
+   read_rds() %>%
+   filter(row_id!=88) ## drop palace polygon
>
> full_contig_nb = poly2nb(full_poly, queen = TRUE)
> full_contig_listw = nb2listw(full_contig_nb, style = "W", zero.policy = TRUE) ## 1 neighborhood without neighbors
> # lapply(full_contig_listw$weights, length)   %>% unlist  ## N neighbor cells
> full_poly = full_poly    %>%
+   ## Add spatially-lagged Damage
+   mutate(
+     lagged_damage = lag.listw(full_contig_listw, full_poly$ratio_damage, zero.policy = TRUE)
+   )
> ## ------------------------------------------------------------------------
> ## --- Neighborhoods in the 2010 regressions
> residual_poly = full_poly      %>%
+   left_join(
```

```
+    gam_residualz_baseline,
+    by = "key_code"
+  )  %>%
+  drop_na(lnr_unemp)     ## As in the 2010 regression data
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Models w/ spatially-lagged Damage
> ## ------------------------------------------------------------------------
> ## Prepare a list object
> splag_mod_list = list_along(gam_2010)
> names(splag_mod_list) = names(gam_2010)
> ## Loop over DVs
> for(i in seq_along(gam_2010)) {
+  ## Pull out and prepare data
+  tmp_data = gam_2010[[i]]$model      %>%
+    as_tibble()    %>%
+    rename(
+      prewar_district = 'as.factor(prewar_district)',
+      population_weight = '(weights)'
+    ) %>%
+    mutate(
+      ## This can be replaced with "tokyo_census_tbl" now
+      lagged_damage = residual_poly$lagged_damage
+    )
+  ## Estimate
+      splag_mod_list[[i]]  =  gam(update(gam_2010[[i]]$formula,  .~.  +  lagged_damage),  data  =  tmp_data,  weights  =
population_weight)
+ }
>
> ## ------------------------------------------------------------------------
> ## tidy and skim the results
> gam_2010   %>%    map(tidy, parametric = TRUE)     %>%
+  enframe   %>%    unnest(cols = c(value))    %>%
+  filter(str_detect(term, pattern = "damage"))
# A tibble: 7 × 6
  name          term        estimate std.error statistic   p.value
  <chr>         <chr>          <dbl>    <dbl>     <dbl>     <dbl>
1 lnr_unemp     ratio_damage  0.00284   0.0164     0.173  0.863
2 lnrm_unemp    ratio_damage -0.00155   0.0206    -0.0754 0.940
3 lnrf_unemp    ratio_damage  0.00729   0.0180     0.406  0.685
4 lnr_proexe    ratio_damage -0.0553    0.0146    -3.79   0.000154
5 lnave_lvlength ratio_damage -0.0551    0.0127    -4.33   0.0000154
```

```
6 lnave_eduyr   ratio_damage -0.00595  0.00267  -2.23  0.0259
7 lnrmale0004   ratio_damage -0.00195  0.00950  -0.205  0.838
> splag_mod_list    %>%    map(tidy, parametric = TRUE)      %>%
+  enframe   %>%    unnest(cols = c(value))    %>%
+  filter(str_detect(term, pattern = "ratio_damage"))
# A tibble: 7 × 6
  name         term        estimate std.error statistic  p.value
  <chr>        <chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 lnr_unemp     ratio_damage  0.00211  0.0164    0.129  0.898
2 lnrm_unemp    ratio_damage -0.00192  0.0206   -0.0931 0.926
3 lnrf_unemp    ratio_damage  0.00604  0.0179    0.337  0.736
4 lnr_proexe    ratio_damage -0.0551   0.0146   -3.77   0.000166
5 lnave_lvlength ratio_damage -0.0549   0.0127   -4.32   0.0000165
6 lnave_eduyr   ratio_damage -0.00593  0.00267  -2.22   0.0265
7 lnrmale0004   ratio_damage -0.00223  0.00951  -0.234  0.815
> splag_mod_list    %>%    map(tidy, parametric = TRUE)      %>%
+  enframe   %>%    unnest(cols = c(value))    %>%
+  filter(str_detect(term, pattern = "lagged_damage"))
# A tibble: 7 × 6
  name         term         estimate std.error statistic p.value
  <chr>        <chr>           <dbl>    <dbl>    <dbl>   <dbl>
1 lnr_unemp     lagged_damage  0.0377   0.0196     1.92  0.0547
2 lnrm_unemp    lagged_damage  0.0182   0.0247     0.737 0.461
3 lnrf_unemp    lagged_damage  0.0673   0.0215     3.13  0.00174
4 lnr_proexe    lagged_damage -0.0158   0.0176    -0.899 0.369
5 lnave_lvlength lagged_damage -0.00837  0.0153    -0.548 0.584
6 lnave_eduyr   lagged_damage -0.00123  0.00321   -0.383 0.702
7 lnrmale0004   lagged_damage -0.0130   0.0117    -1.11  0.268
> ## ---------------------------------------------------------------------
> ## Stargazer
> stargazer(
+  splag_mod_list,
+  type = "text",
+  keep = c("ratio_damage"),
+  single.row = FALSE, digits = 4,
+  align = TRUE, model.names = FALSE, model.numbers = FALSE,
+  dep.var.labels.include = TRUE
+ )

=================================================================================
                                Dependent variable:
            ---------------------------------------------------------------------
                                     formula
```

```
-------------------------------------------------------------------------------------
ratio_damage    0.0021  -0.0019   0.0060  -0.0551*** -0.0549*** -0.0059**   -0.0022
             (0.0164) (0.0206) (0.0179)  (0.0146)   (0.0127)   (0.0027)   (0.0095)

-------------------------------------------------------------------------------------
Observations    2,158    2,158    2,158    2,158     2,158     2,158     2,158
Adjusted R2    0.4787   0.4574   0.2919    0.6903    0.3816    0.7564    0.0077
Log Likelihood 554.4038 59.1918  359.4612  791.8095  1,095.3890 4,460.1300 1,602.3050
UBRE          88.7537  140.4297 106.3251  71.2145    53.7470    2.3774    33.5777
=====================================================================================
Note:                                              *p<0.1; **p<0.05; ***p<0.01
> ## ------------------------------------------------------------------
>
>
> ## EOF
>
> ## ------------------------------------------------------------------
> ##
> ## Draw maps: Figure 1, 5, A3
> ##
> ## ------------------------------------------------------------------
>
> ## environment setting
> Sys.setenv(LANGUAGE="en")
> gc();gc()
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5151544 275.2    7833997 418.4  7833997 418.4
Vcells 18336111 139.9   31101692 237.3 31101689 237.3
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5151547 275.2    7833997 418.4  7833997 418.4
Vcells 18336134 139.9   31101692 237.3 31101689 237.3
> rm(list = ls())
> options(scipen = 999)   ## Disable scientific notation
>
> ## ------------------------------------------------------------------
> ##
> ## Initial settings
> ##
> ## ------------------------------------------------------------------
> ## Set directory
> ## ------------------------------------------------------------------
> root_dir = "E:/Dropbox/429_MDA_replication/Replication"
> ## Load packages and functions
```

```
> source(file.path(root_dir, "2_Code/1_PackagesFunctions.R"))
> needs(lubridate)
>
> ## ------------------------------------------------------------------------
> ## Data directory
> data_dir = root_dir      %>%    file.path("1_Data") %>%    dir_create()
>
> ## Output directory
> output_dir = root_dir    %>%    file.path("3_Result")      %>%    dir_create()
> figure_dir = root_dir    %>%    file.path("4_Figures")     %>%    dir_create()
> ## Working directory
> working_dir = root_dir   %>%    file.path("5_Working")     %>%    dir_create()
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------
> ## Load and prepare data objects
> ## ------------------------------------------------------------
> ## Census data
> tokyo2keep = data_dir    %>%
+   file.path("census_data4.csv") %>%
+   read_csv(col_types = cols()) %>%
+   mutate(group = "FullSample")
>
> ## Full polygons
> tokyo_full_poly = data_dir %>% file.path("RaidShp_May2020b.rds") %>% read_rds()
> background_poly = tokyo_full_poly  %>%
+   filter(
+     row_id != 941 & row_id != 974 &
+       row_id != 1302 & row_id != 168
+   )  %>%
+   st_union()       %>%
+   as("Spatial")
> ## Raid polygons w/ prewar census population density
> raid_poly = tokyo_full_poly %>%
+   filter(!is.na(ratio_damage))  %>%
+   as("Spatial")
> ## ------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------
> ##
> ## Figure 1: Draw residential ratio map
```

```
> ##
> ## -------------------------------------------------------------
> ## Target locations
> target_locationz = read_rds(file.path(data_dir, "TargetLocationsCombined.rds"))
> ## -------------------------------------------------------------
> ## Pull out the imperial palace polygon
> palace = tokyo_full_poly   %>%
+   filter(row_id == 88)    %>%
+   st_centroid %>%
+   st_transform(3095)  ## EPSG:3095 --- http://spatialreference.org/ref/epsg/tokyo-utm-zone-54n/
Warning message:
In st_centroid.sf(.) :
  st_centroid assumes attributes are constant over geometries of x
> ## Buffer of 10km
> palace_buffer = st_buffer(palace, dist = 10*10^3)   %>%
+   as("Spatial")  %>%
+   spTransform(prj)
Warning message:
PROJ support is provided by the sf and terra packages among others
>
> ## Buffer of 10km
> palace_buffer_2km = st_buffer(palace, dist = 2*10^3)   %>%
+   as("Spatial")   %>%
+   spTransform(prj)
Warning message:
PROJ support is provided by the sf and terra packages among others
> ## -------------------------------------------------------------
> ## Set polygon colors
> ## -------------------------------------------------------------
> background_poly_col = "ivory2";
> bord_col = "gray10";
> residential_colz = c("white","gray40")  ## gray scale
> nn = 10
> fxBrks = seq(0, 1, by = 0.1)
> nn = length(fxBrks)
> palz = colorRampPalette(residential_colz, space = "rgb")(nn)
> raid_poly$col = palz[raid_poly$ratio_residential*10+1]
> circle_col = c("orangered2", "dodgerblue2", "white")
> dot_col = "chartreuse2"
>
> ## -------------------------------------------------------------
> ## Plot
> ## -------------------------------------------------------------
```

```
> fig_height = 6
> # dev.new(height = fig_height, width = 8)
> png(file.path(figure_dir, "fig_1"), width = 6.15, height = fig_height, units = "in", res = 240)
> par(mar=rep(0,4), oma=rep(0,4), bg="white", family = "Helvetica", lend="square")
> plot(background_poly, col = background_poly_col, border = bord_col, lwd = .25, xaxs = "i", yaxs = "i") ## Background
> plot(raid_poly, col=raid_poly$col, border = bord_col, lwd=.35, add = TRUE)      ## square w/ colors
> ## 10km Buffer
> lines(palace_buffer, lwd = 3, col = circle_col[3])
> lines(palace_buffer, lwd = 1, col = circle_col[1])
> ## 2km Buffer
> lines(palace_buffer_2km, lwd = 3, col = circle_col[3])
> lines(palace_buffer_2km, lwd = 1, col = circle_col[2])
> # text(coordinates(palace_buffer_2km)[,1], y = coordinates(palace_buffer_2km)[,2], labels = "r=2km", col = "maroon2")
> ## Target locations
> points(target_locationz, pch = 21, bg = adjustcolor(dot_col, alpha = 0.85), col = "gray10", cex = 1)
> ## Add legend
> legend(
+   title = "Residential Area (%)",
+   "bottomright",
+   inset=c(.001,.001),
+   bg = "white", box.lwd = 1,
+   ncol=2, cex=1,
+   col = c(palz, background_poly_col),
+   legend = c(str_c(c(seq(0, 100, by = 10)), "%"), "NA"),
+   lwd=12, seg.len=.85
+ )
Warning message:
In (function (s, units = "user", cex = NULL, font = NULL, vfont = NULL,  :
  font family not found in Windows font database
> ## Close
> dev.off()
null device
          1
> ## ------------------------------------------------------------
>
>
>
>
> ## ------------------------------------------------------------
> ##
> ## Figure 5a: Draw raid damage map
> ## Raw human coding
> ##
```

```
> ## ------------------------------------------------------------
> ## Set some colors
> # background_poly_col = "gray90"
> background_poly_col = "ivory2";
> sea_col = "azure3";
> land_col = "gray96";
> bord_col = "gray10";
> fig_height = 6
> ## ------------------------------------------------------------
> ## Set polygon colors
> damage_colz = c("white","orange","firebrick2","darkred")
> nn = 10
> fxBrks = seq(0, 1, by = 0.1)
> nn = length(fxBrks)
> palz = colorRampPalette(damage_colz, space = "rgb")(nn)
> raid_poly$col = palz[raid_poly$ratio_damage*10+1]
> ## ------------------------------------------------------------
> ## Plot base map
> ## ------------------------------------------------------------
> # dev.new(height = fig_height, width = 8)
> png(file.path(figure_dir, "fig_5a.png"), width = 6.15, height = fig_height, units = "in", res = 240)
> par(mar=rep(0,4), oma=rep(0,4), bg="white", family = "Helvetica", lend="square")
> plot(background_poly, col = background_poly_col, border = bord_col, lwd = .25, xaxs = "i", yaxs = "i") ## Background
> plot(raid_poly, col=raid_poly$col, border = bord_col, lwd=.35, add = TRUE)      ## square w/ colors
> ## Scalebar
> # scalebar_rev(loc = c(139.585, 35.54), length = .09)
> ## Add legend
> legend(
+   title = "Raid Damage (%)",
+   "bottomright",
+   inset=c(.001,.001),
+   bg = "white", box.lwd = 1,
+   ncol=2, cex=1,
+   col = c(palz, background_poly_col),
+   legend = c(str_c(c(seq(0, 100, by = 10)), "%"), "NA"),
+   lwd=12, seg.len=.85
+ )
Warning message:
In (function (s, units = "user", cex = NULL, font = NULL, vfont = NULL,  :
  font family not found in Windows font database
> ## Close
> dev.off()
null device
```

```
       1
> ## ------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure 5b: Draw raid damage map
> ## Residualized human coding
> ##
> ## ------------------------------------------------------------------------
> ## Prepare data and residualized raid scores
> ## ------------------------------------------------------------------------
> ## Treatment
> treatment_vec = "ratio_damage"
> ## ------------------------------------------------------------------------
> ## X: Pretreatment covariates
> ## ------------------------------------------------------------------------
> ## Distance variables
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> ## Polynomial
> polynomial_degree = 5
> distance_termz_poly = polynomial_varname(distance_termz_base, degree = polynomial_degree)
> distance_term_specification = list( distance_termz_poly, str_c("s(", distance_termz_base, ")") )
> ## Geographical features
> spatial_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## Flammability score
> flame_scorez = c("hazard", "max_nghb_hazard")
> ## ------------------------------------------------------------------------
> ## Spatial spline/polynomials and fixed effects
> ## ------------------------------------------------------------------------
> ## District FEs
> district_fe = "as.factor(prewar_district)"   ## 35-district FE
> ## Spatial terms
```

```
> spatial_term_lbl = c("Polynomial", "GAM")   ## This indicates polynomial or spline specification in the csv files
> lonlat_base = c("z_lon", "z_lat")
> spatial_polynomial = c(
+   polynomial_varname(lonlat_base, degree = polynomial_degree),
+   "I(z_lon*z_lat)", "I(z_lon^2*z_lat)", "I(z_lon*z_lat^2)","I(z_lon^2*z_lat^2)", "I(z_lon^3*z_lat)", "I(z_lon*z_lat^3)",
+   "I(z_lon^4*z_lat)", "I(z_lon*z_lat^4)", "I(z_lon^3*z_lat^2)", "I(z_lon^2*z_lat^3)"
+ )
> spatial_term_specification = list(spatial_polynomial, "te(longitude, latitude)")
>
> ## ------------------------------------------------------------------------
> ## Prepare models
> ## ------------------------------------------------------------------------
> ## 1. Polynomial model
> remove_polynomial = chr2fml(treatment_vec,
+                         idv_list = list(
+                           spatial_covariates,
+                           distance_term_specification[[1]], spatial_term_specification[[1]],
+                           district_fe  ## ADDED: Prewar 35-district FEs
+                         )
+ )
> ## 2. GAM with additional spline terms
> remove_gam = chr2fml(treatment_vec,
+                    idv_list = list(
+                      spatial_covariates,
+                      distance_term_specification[[2]], spatial_term_specification[[2]],
+                      district_fe  ## ADDED: Prewar 35-district FEs
+                    )
+ )
> ## ------------------------------------------------------------------------
> ## Pull out variable names
> xvarz = fm2xvarWspline(remove_polynomial)
> xvarz = xvarz[!str_detect(xvarz, c("I¥¥(|¥¥)"))]
> varz2keep = c(
+   "key_code", "year", treatment_vec,
+   # depvar_vec_combined,
+   xvarz,
+   "palace_dist", "pop", "longitude", "latitude"
+ )    %>%
+   unique
> ## ------------------------------------------------------------------------
> ## Extract variables and add subsample index
> tokyo2keep = tokyo2keep  %>%
+   filter(year == 2010)   %>%
```

```
+   drop_na(ratio_damage, pop)    %>%
+   ## Standardize for polynomial terms
+   mutate_at(vars(all_of(distance_termz_base)), list(scale2))
>
> ## ---------------------------------------------------------------------
> ## Note: gam.residuals() function
> ## --- Response residuals are the raw residuals (data minus fitted values)
> ## --- Scaled Pearson residuals are raw residuals divided by the standard deviation of the data according to the model
mean variance relationship and estimated scale parameter.
> ## --- Pearson residuals are the same, but multiplied by the square root of the scale parameter
> ## --- Deviance residuals simply return the deviance residuals defined by the model family.
> ## --- Working residuals are the residuals returned from model fitting at convergence.
> ## ---------------------------------------------------------------------
> ## Estimate models to obtain residuals
> ## --- WITH 35-district FEs
> remove_wls_2010 = lm(remove_polynomial, data = tokyo2keep, weights = tokyo2keep$pop)
> remove_gam_2010 = gam(remove_gam, data = tokyo2keep, weights = tokyo2keep$pop)
> tokyo2keep$res_ply_raid_wFE = residuals(remove_wls_2010)
> tokyo2keep$res_gam_raid_wFE = residuals(remove_gam_2010, type = "response")
> ## ---------------------------------------------------------------------
> tokyo_resid = tokyo2keep %>%    select(key_code, res_ply_raid_wFE, res_gam_raid_wFE)
>
> # Extract the data frame from the SpatialPolygonsDataFrame
> raid_poly_df <- as.data.frame(raid_poly) %>%
+   mutate(key_code = as.numeric(as.character(key_code)))
>
> # Perform the join
> raid_poly_resid_df <- raid_poly_df %>%
+   left_join(tokyo_resid, by = "key_code")
>
> # Create a new SpatialPolygonsDataFrame with the joined data
> raid_poly_resid <- SpatialPolygonsDataFrame(raid_poly, data = raid_poly_resid_df)
>
> ## ---------------------------------------------------------------------
> ## Map
> ## ---------------------------------------------------------------------
> ## Set baseline polygon colors
> background_poly_col = "ivory2";
> sea_col = "azure3";
> land_col = "gray92";
> bord_col = "gray10";
> nn = 11
> fig_height = 6
```

```
> varz_vec = c("res_gam_raid_wFE")
> fig_lbl = c("gam_residual_w35dFE")
> map_dir = figure_dir
> ## ---------------------------------------------------------------------
> ## Residualized raid index
> var2plot = raid_poly_resid@data %>%    pull(varz_vec[1])
> var2plot = rescaleIt01(var2plot)      ## Rescale to a [0,1] interval
> ## ---------------------------------------------------------------------
> ## Color intervals
> fxBrks = seq(min(var2plot, na.rm = TRUE), max(var2plot, na.rm = TRUE), length = nn)
> palz = colorRampPalette(c("white","orange","firebrick2","darkred"), space = "rgb")(nn)
> raid_poly_resid$col = palz[var2plot*10+1]
> ## ---------------------------------------------------------------------
> ## Plot map
> # dev.new(width = 6.15, height = fig_height)
> png(file.path(map_dir, "fig_5b.png"), width = 6.15, height = fig_height, units = "in", res = 240)
> par(mar = rep(0,4), oma = rep(0,4), bg="white", family = "Helvetica", lend="square")
> # plot(background_poly, col = "gray90", border = NA, lwd = .35)    ## square w/ colors
> plot(background_poly, col = background_poly_col, border = bord_col, lwd = .25, xaxs = "i", yaxs = "i")   ## Background
> plot(raid_poly_resid, col = raid_poly_resid$col, border = bord_col, lwd =.25, add = TRUE)                      ## Square
w/ colors
> ## Add legend
> legend(
+   title = "Raid Damage (%)",
+   "bottomright",
+   inset = c(.001,.001),
+   bg = "white", box.lwd = 1,
+   ncol=2, cex=1,
+   col = c(palz, background_poly_col),
+   legend = c(str_c(c(seq(0, 100, by = 10)), "%"), "NA"),
+   lwd = 12, seg.len = .85
+ )
Warning message:
In (function (s, units = "user", cex = NULL, font = NULL, vfont = NULL,  :
  font family not found in Windows font database
> dev.off()
null device
        1
> ## ---------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------
```

```
> ##
> ## Figure A3: Prewar population density
> ##
> ## -------------------------------------------------------------
> ## Set some colors
> background_poly_col = "ivory2";
> sea_col = "azure3";
> land_col = "gray96";
> bord_col = "gray10"
>   ## -------------------------------------------------------------
> ## Set Colors: Prewar Population density
> ## -------------------------------------------------------------
> fxBrks = c(18, 10^3, 5*10^3, 10^4, 3*10^4)
> nn = length(fxBrks)
> classes_km = classIntervals(
+   raid_poly$PopDensity_1939_km2, style = "fixed", n = nn,
+   fixedBreaks = c(fxBrks, max(raid_poly$PopDensity_1939_km2))
+ )
> # classes_km$brks = round(classes_km$brks, 2)
> classes_km$brks = round(classes_km$brks)
> pal = colorRampPalette(brewer.pal(nn, "BuGn"))(nn)
> # pal = palz[[1]]   ## customized palette:
> # pal = pal[2:length(pal)]    ## drop white
> cols = findColours(classes_km, pal)    ## Set cols
> names(attr(cols,"table"))[length(names(attr(cols,"table")))] = str_c("[", fxBrks[length(fxBrks)], ",]")
>
> ## -------------------------------------------------------------
> ## Draw Palace Distance
> ## -------------------------------------------------------------
> png(file.path(figure_dir, "fig_a3.png"), width = 6.15, height = fig_height, units = "in", res = 240)
> par(mar=rep(0,4), oma=rep(0,4), bg="white", family = "Helvetica", lend="square")
> plot(background_poly, col = background_poly_col, border = bord_col, lwd = .25, xaxs = "i", yaxs = "i") ## Background
> plot(raid_poly, col=cols, border = bord_col, lwd=.25, add = TRUE)    ## square w/ colors
> ## Add legend
> legend(
+   title = "Population Density (km2, 1939)",
+   "bottomright",
+   inset=c(.001,.001),
+   bg = "white", box.lwd = 1,
+   ncol = 1, cex=1,
+   col=attr(cols,"palette"),
+   legend=names(attr(cols,"table")),
+   lwd=12, seg.len=.85)
```

```
Warning message:
In (function (s, units = "user", cex = NULL, font = NULL, vfont = NULL,  :
  font family not found in Windows font database
>
> dev.off()
null device
          1
> ## ------------------------------------------------------------
>
> ##EOF
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure 8: Sequential g-estimation
> ##
> ## ------------------------------------------------------------------------
>
> ## environment setting
> Sys.setenv(LANGUAGE="en")
> gc();gc()
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5415943 289.3   10193134 544.4  7833997 418.4
Vcells 11747553  89.7   31101692 237.3 31101689 237.3
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5415946 289.3   10193134 544.4  7833997 418.4
Vcells 11747576  89.7   31101692 237.3 31101689 237.3
> rm(list = ls())
> options(scipen = 999)   ## Disable scientific notation
>
> ## ------------------------------------------------------------------------
> ##
> ## Initial settings
> ##
> ## ------------------------------------------------------------------------
> ## Set directory
> ## ------------------------------------------------------------------------
> root_dir = "E:/Dropbox/429_MDA_replication/Replication"
> ## Load packages and functions
> source(file.path(root_dir, "2_Code/1_PackagesFunctions.R"))
>
> ## ------------------------------------------------------------------------
> ## Data directory
> data_dir = root_dir      %>%    file.path("1_Data") %>%    dir_create()
```

```
> ## Output directory
> output_dir = root_dir    %>%    file.path("3_Result")      %>%    dir_create()
> figure_dir = root_dir    %>%    file.path("4_Figures")     %>%    dir_create()
> ## Working directory
> working_dir = root_dir   %>%    file.path("5_Working")     %>%    dir_create()
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Data and model preparation
> ##
> ## ------------------------------------------------------------------------
> ## ------------------------------------------------------------------------
> ## Data and model preparation
> ## ------------------------------------------------------------------------
> ## Load census data w/ raid damage variable
> tokyo_census_tbl = data_dir     %>%
+   file.path("census_data4.csv") %>%
+   read_csv(col_types = cols())
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ## Variables
> ## ------------------------------------------------------------------------
> ## Dependent variables (t)
> ## ------------------------------------------------------------------------
> depvar_vec = c(
+   "lnr_unemp", "lnrm_unemp", "lnrf_unemp",    ## Unemployment rate
+   "lnr_proexe", "lnave_lvlength", "lnave_eduyr", "lnrmale0004"   ## Prop boys
+ )
> ## Lagged dependent variables (t-1)
> lagged_depvar_vec = str_c(depvar_vec, "_lagged")
> ## DV labels for figures
> outcome_lbl = c("ln % Unemp", "ln % Unemp (Male)", "ln % Unemp (Female)", "ln % Prof Exec", "ln Ave Res Yrs", "ln Ave Edu
Yrs", "ln % Male <6 y/o")
> outcome_lbl = outcome_lbl       %>%
+   as.data.frame    %>%
+   mutate(depvar = depvar_vec)
> names(outcome_lbl)[1] = "outcome_lbl"
> ## ------------------------------------------------------------------------
> ## Treatment and mediator variables
```

```
> ## ------------------------------------------------------------------------
> ## --- Treatment
> treatment = "ratio_damage"      ## New treatment variable Dec 2018
> ## --- New population
> mediator = "lnr_newpop"         ## logged new population ratio
> ## --- 6F+ apartment
> mediator2 = "lnr_kyodojutaku6up"
> ## ------------------------------------------------------------------------
> ## X: Pretreatment covariates
> ## ------------------------------------------------------------------------
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> ## Polynomial
> polynomial_degree = 5
> distance_termz_poly = polynomial_varname(distance_termz_base, degree = polynomial_degree)
> pretreatment_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Distance terms
+   distance_termz_poly,
+   ## Terrain terms
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway terms
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## ------------------------------------------------------------------------
> ## Z: Intermediate confounders
> ## ------------------------------------------------------------------------
> z_base = c(
+   "r_popyoung", "r_popold", "ave_age", "r_popforeign",
+   "r_singlehh", "r_myhouse", "r_kyodojutaku",
+   "r_worker3sec", "r5pop", "r5r_worker2sec", "PopDensity"
+ )
> ## Logged Z variables
> z_base_ln = str_c(z_base, "_ln")
> z_lagged_ln = str_c(z_base_ln, "_lagged")    ## these are included in the dmediation function
> ## ------------------------------------------------------------------------
> ## Spatial polynomial and fixed effects
> ## ------------------------------------------------------------------------
> fixed_effect = "as.factor(prewar_district)"  ## 35-district FE
> lonlat_base = c("z_lon", "z_lat")
> polynomialz = c(
```

```
+   polynomial_varname(lonlat_base, degree = polynomial_degree),
+   "I(z_lon*z_lat)", "I(z_lon^2*z_lat)", "I(z_lon*z_lat^2)","I(z_lon^2*z_lat^2)",
+           "I(z_lon^3*z_lat)",    "I(z_lon*z_lat^3)",    "I(z_lon^4*z_lat)",    "I(z_lon*z_lat^4)",    "I(z_lon^3*z_lat^2)",
"I(z_lon^2*z_lat^3)"
+ )
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ## Prepare data
> ## ------------------------------------------------------------------------
> tokyo4gest_20002010 = tokyo_census_tbl       %>%    filter(year == 2000|year == 2010) %>%
+   drop_na(all_of(treatment))    %>%
+   ## compute logged Z
+   mutate_at(
+     vars(all_of(z_base)), list(ln = ~log(. + 0.01))
+   )   %>%
+   ## grouping
+   group_by(key_code)  %>%
+   ## compute lagged Y and Z
+   mutate_at(
+     vars(all_of(depvar_vec), all_of(z_base_ln)), list(lagged = ~lag(.))
+   )   %>% ungroup %>%
+   ## Standardize for polynomial terms
+   mutate_at(
+     vars(all_of(distance_termz_base)), list(scale2)
+   )   %>%
+   mutate_at(
+     vars(all_of(distance_termz_base)), list(as.numeric)
+   )   %>%
+   ## drop raw and simply logged (but not lagged) Z variables
+   select(-all_of(z_base), -all_of(z_base_ln))
>
> tokyo4gest_19952000 = tokyo4gest_20002010
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Sequential g-estimation
> ##
> ## ------------------------------------------------------------------------
> ## Bootstrap settings
> n_boot = 1000*5
```

```
> boot_loop = 1:n_boot
> ## ------------------------------------------------------------------------
> ## Output matrix master
> master_matrix = matrix(NA, nrow = n_boot, ncol = 2) %>% as.data.frame
> names(master_matrix) = c("outcome", "boot_est")
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ## Loop over mediator values
> ## mediator = "lnr_newpop"
> ## ------------------------------------------------------------------------
> set.seed(1234)
> for (i in 1:length(depvar_vec)) {
+   ## ------------------------------------------------------------------------
+   ## Set models
+   tmp_depvar = depvar_vec[i]     ## set DV
+   tmp_lagged_depvar = lagged_depvar_vec[i]  ## set mediator (lagged DV)
+   ##  --- ATE model simply conditioning on pretreatment covariates
+   mod_ATE = chr2fml(tmp_depvar, idv_list = list(treatment, pretreatment_covariates, fixed_effect, polynomialz))
+   ##  --- naive ACDE model simply conditioning on pretreatment covariates and mediator
+     mod_withMwoZ  =  chr2fml(tmp_depvar,  idv_list  =  list(treatment,  mediator,  pretreatment_covariates,  fixed_effect,
polynomialz))
+   ##  --- naive ACDE model fully conditioning on pretreatment covariates, mediator, and intermediate confounders
+    mod_wMandZ  =  chr2fml(tmp_depvar,  idv_list  =  list(treatment,  mediator,  pretreatment_covariates,  tmp_lagged_depvar,
z_lagged_ln, fixed_effect, polynomialz))
+   ##  ---- ACDE model
+   acde_mod = update(mod_ATE, demediated_outcome ~ .)
+   ## ------------------------------------------------------------------------
+
+   ## ------------------------------------------------------------------------
+   ## Prepare data obj for estimation
+   tokyo4gest = tokyo4gest_19952000   ## 1995 and 2000 datasets -> replaced with 2000-2010 data, see above
+   if (str_detect(tmp_depvar, "lnave")) tokyo4gest = tokyo4gest_20002010   ## replace data: 2000 and 2010 datasets
+   tmp_reg_data = get_all_vars(update(mod_wMandZ, . ~ . + year + pop), data = tokyo4gest) %>% drop_na
+   ## ------------------------------------------------------------------------
+   ## Output matrix
+   tmp_ACDE_matrix = master_matrix
+   tmp_ACDE_matrix[,1] = depvar_vec[i]
+   tmp_ATE_matrix = tmp_woZ_matrix = tmp_naive_matrix = tmp_ACDE_matrix
+   ## ------------------------------------------------------------------------
+   ## Bootstrap
+   for (b in boot_loop) {
```

```
+     ## ------------------------------------------------------------------------
+     ## Print progress
+     cat(str_c("Outcome: ", i, "/", length(depvar_vec), " bootstrap ", b, "/", n_boot, " done. ¥r"))
+     ## ------------------------------------------------------------------------
+     ## Bootstrap sample
+     boot_sample = sample_frac(tmp_reg_data, size = 1, replace = TRUE)
+     ## ------------------------------------------------------------------------
+     ## Set mediator value (recentering)
+     boot_sample[,mediator] = boot_sample[,mediator] - mean(boot_sample[,mediator])
+     ## ------------------------------------------------------------------------
+     ## Run models
+     est_ATE = speedlm(mod_ATE, data = boot_sample, weights = boot_sample$pop)
+     est_withMwoZ = speedlm(mod_withMwoZ, data = boot_sample, weights = boot_sample$pop)
+     est_withMandZ = speedlm(mod_wMandZ, data = boot_sample, weights = boot_sample$pop)  ## <- first stage model
+     ## ------------------------------------------------------------------------
+     ## Demediate the outcome
+     boot_sample$demediated_outcome = boot_sample[,tmp_depvar] - coef(est_withMandZ)[mediator]*boot_sample[,mediator]
+     ## Second-stage estimate
+     direct_est = speedlm(acde_mod, data = boot_sample, weights = boot_sample$pop)
+     ## ------------------------------------------------------------------------
+     ## Store results (current round)
+     tmp_ATE_matrix[b,2] = coef(est_ATE)[treatment]
+     tmp_woZ_matrix[b,2] = coef(est_withMwoZ)[treatment]
+     tmp_naive_matrix[b,2] = coef(est_withMandZ)[treatment]
+     tmp_ACDE_matrix[b,2] = coef(direct_est)[treatment]
+     ## ------------------------------------------------------------------------
+   }
+   ## ------------------------------------------------------------------------
+   ## Combine bootstrap results (all outcomes)
+   if (i > 1) {
+     ATE_matrix = bind_rows(ATE_matrix, tmp_ATE_matrix)
+     woZ_matrix = bind_rows(woZ_matrix, tmp_woZ_matrix)
+     naive_matrix = bind_rows(naive_matrix, tmp_naive_matrix)
+     ACDE_matrix = bind_rows(ACDE_matrix, tmp_ACDE_matrix)
+   }  else   {
+     ATE_matrix = tmp_ATE_matrix
+     woZ_matrix = tmp_woZ_matrix
+     naive_matrix = tmp_naive_matrix
+     ACDE_matrix = tmp_ACDE_matrix
+   }
+   ## ------------------------------------------------------------------------
+ }
> tcome: 7/7 bootstrap 5000/5000 done.
```

```
> ## ------------------------------------------------------------------------
> ## Add estimator column
> ATE_matrix$estimator = "ATE"
> woZ_matrix$estimator = "NaiveACDE_withoutZ"
> naive_matrix$estimator = "NaiveACDE_withZ"
> ACDE_matrix$estimator = "ACDE_gest"
> ## ------------------------------------------------------------------------
> ## Combine
> boot_fullresult = ATE_matrix   %>%
+   as_tibble %>%
+   bind_rows(ACDE_matrix) %>%
+   bind_rows(woZ_matrix)  %>%
+   bind_rows(naive_matrix)%>%
+   arrange(outcome, estimator)   %>%
+   mutate(mediator = mediator)
> ## ------------------------------------------------------------------------
> ## Cancel comment-out to save the bootstrap results
> ## Bootstrap output directory
> # boot_dir_sub = file.path(output_dir, str_c(mediator, "_boot_n_", n_boot, "_censusmediation")) %>%      dir_create()
> ## ------------------------------------------------------------------------
> ## Save result
> # file_name = str_c("bootstrap", n_boot, "treatment", treatment, "mediator", mediator, "result.csv", sep = "_")
> # write_csv(boot_fullresult, path = file.path(boot_dir_sub, file_name))
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ## Loop over mediator values
> ## mediator = "lnr_kyodojutaku6up"
> ## ------------------------------------------------------------------------
> set.seed(1234)
> for (i in 1:length(depvar_vec)) {
+   ## ------------------------------------------------------------------------
+   ## Set models
+   tmp_depvar = depvar_vec[i]     ## set DV
+   tmp_lagged_depvar = lagged_depvar_vec[i]  ## set mediator (lagged DV)
+   ##  --- ATE model simply conditioning on pretreatment covariates
+   mod_ATE = chr2fml(tmp_depvar, idv_list = list(treatment, pretreatment_covariates, fixed_effect, polynomialz))
+   ##  --- naive ACDE model simply conditioning on pretreatment covariates and mediator
+     mod_withMwoZ  =  chr2fml(tmp_depvar,  idv_list  =  list(treatment,  mediator2,  pretreatment_covariates,  fixed_effect,
polynomialz))
+   ##  --- naive ACDE model fully conditioning on pretreatment covariates, mediator, and intermediate confounders
```

```
+    mod_wMandZ = chr2fml(tmp_depvar, idv_list = list(treatment, mediator2, pretreatment_covariates, tmp_lagged_depvar,
z_lagged_ln, fixed_effect, polynomialz))
+   ##  ---- ACDE model
+   acde_mod = update(mod_ATE, demediated_outcome ~ .)
+   ## ------------------------------------------------------------------------
+
+   ## ------------------------------------------------------------------------
+   ## Prepare data obj for estimation
+   tokyo4gest = tokyo4gest_19952000   ## 1995 and 2000 datasets -> replaced with 2000-2010 data, see above
+   if (str_detect(tmp_depvar, "lnave")) tokyo4gest = tokyo4gest_20002010   ## replace data: 2000 and 2010 datasets
+   tmp_reg_data = get_all_vars(update(mod_wMandZ, . ~ . + year + pop), data = tokyo4gest) %>% drop_na
+   ## ------------------------------------------------------------------------
+   ## Output matrix
+   tmp_ACDE_matrix = master_matrix
+   tmp_ACDE_matrix[,1] = depvar_vec[i]
+   tmp_ATE_matrix = tmp_woZ_matrix = tmp_naive_matrix = tmp_ACDE_matrix
+   ## ------------------------------------------------------------------------
+   ## Bootstrap
+   for (b in boot_loop) {
+     ## ------------------------------------------------------------------------
+     ## Print progress
+     cat(str_c("Outcome: ", i, "/", length(depvar_vec), " bootstrap ", b, "/", n_boot, " done. ¥r"))
+     ## ------------------------------------------------------------------------
+     ## Bootstrap sample
+     boot_sample = sample_frac(tmp_reg_data, size = 1, replace = TRUE)
+     ## ------------------------------------------------------------------------
+     ## Set mediator value (recentering)
+     boot_sample[,mediator2] = boot_sample[,mediator2] - mean(boot_sample[,mediator2])
+     ## ------------------------------------------------------------------------
+     ## Run models
+     est_ATE = speedlm(mod_ATE, data = boot_sample, weights = boot_sample$pop)
+     est_withMwoZ = speedlm(mod_withMwoZ, data = boot_sample, weights = boot_sample$pop)
+     est_withMandZ = speedlm(mod_wMandZ, data = boot_sample, weights = boot_sample$pop)  ## <- first stage model
+     ## ------------------------------------------------------------------------
+     ## Demediate the outcome
+     boot_sample$demediated_outcome = boot_sample[,tmp_depvar] - coef(est_withMandZ)[mediator2]*boot_sample[,mediator2]
+     ## Second-stage estimate
+     direct_est = speedlm(acde_mod, data = boot_sample, weights = boot_sample$pop)
+     ## ------------------------------------------------------------------------
+     ## Store results (current round)
+     tmp_ATE_matrix[b,2] = coef(est_ATE)[treatment]
+     tmp_woZ_matrix[b,2] = coef(est_withMwoZ)[treatment]
+     tmp_naive_matrix[b,2] = coef(est_withMandZ)[treatment]
```

```
+     tmp_ACDE_matrix[b,2] = coef(direct_est)[treatment]
+     ## ------------------------------------------------------------------------
+   }
+   ## ------------------------------------------------------------------------
+   ## Combine bootstrap results (all outcomes)
+   if (i > 1) {
+     ATE_matrix = bind_rows(ATE_matrix, tmp_ATE_matrix)
+     woZ_matrix = bind_rows(woZ_matrix, tmp_woZ_matrix)
+     naive_matrix = bind_rows(naive_matrix, tmp_naive_matrix)
+     ACDE_matrix = bind_rows(ACDE_matrix, tmp_ACDE_matrix)
+   }   else    {
+     ATE_matrix = tmp_ATE_matrix
+     woZ_matrix = tmp_woZ_matrix
+     naive_matrix = tmp_naive_matrix
+     ACDE_matrix = tmp_ACDE_matrix
+   }
+   ## ------------------------------------------------------------------------
+ }
> tcome: 7/7 bootstrap 5000/5000 done.
> ## ------------------------------------------------------------------------
> ## Add estimator column
> ATE_matrix$estimator = "ATE"
> woZ_matrix$estimator = "NaiveACDE_withoutZ"
> naive_matrix$estimator = "NaiveACDE_withZ"
> ACDE_matrix$estimator = "ACDE_gest"
> ## ------------------------------------------------------------------------
> ## Combine
> boot_fullresult2 = ATE_matrix  %>%
+   as_tibble %>%
+   bind_rows(ACDE_matrix) %>%
+   bind_rows(woZ_matrix)  %>%
+   bind_rows(naive_matrix)%>%
+   arrange(outcome, estimator)   %>%
+   mutate(mediator = mediator2)
> ## ------------------------------------------------------------------------
> ## Cancel comment-out to save the bootstrap results
> ## Bootstrap output directory
> # boot_dir_sub2 = file.path(output_dir, str_c(mediator2, "_boot_n_", n_boot, "_censusmediation")) %>%    dir_create()
> ## ------------------------------------------------------------------------
> ## Save result
> # file_name = str_c("bootstrap", n_boot, "treatment", treatment, "mediator", mediator2, "result.csv", sep = "_")
> # write_csv(boot_fullresult2, path = file.path(boot_dir_sub2, file_name))
> ## ------------------------------------------------------------------------
```

```
>
> ## ------------------------------------------------------------------------
> ## ------------------------ Estimations end here ------------------------
> ## ------------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------------
> ## <<<----------------------<<< Plot estimates >>>---------------------->>>
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ##
> ## Plot: Summary Comparison Plot
> ## mediator = lnr_newpop
> ##
> ## ------------------------------------------------------------------------
> ## Cancel comment-out to read the saved estimates
> # boot_fullresult_path = list.files(boot_dir_sub, pattern = ".csv", full.names = TRUE)
> # boot_fullresult = read.csv(boot_fullresult_path)
> ## ------------------------------------------------------------------------
> ## Set alpha
> alpha_level = 0.95
> crit_valz = c((1-alpha_level)/2, 1-(1-alpha_level)/2)
> ## ------------------------------------------------------------------------
> ## Pull out outcome and estimator vectors
> outcome_vec = boot_fullresult$outcome   %>% unique
> estimator_vec = boot_fullresult$estimator   %>% unique
> n_estimatorz = estimator_vec    %>% length
> n_outcomez = outcome_vec    %>% length
> ## ------------------------------------------------------------------------
> ## Colors
> # colz_vec = c("blue2", "firebrick2", "gray65", "magenta2")
> # alpha_vec = c(0.6, 0.15, 1)
> colz_vec = c("lightsteelblue4", "lightsteelblue4", "gray65", "magenta2")
> alpha_vec = c(0.6, 0.3, 1)
> ## ------------------------------------------------------------------------
> ## Convert for comparison
> ## ------------------------------------------------------------------------
> ## Compute delta = ATE - ACDE
> plot_summary = boot_fullresult  %>%
+  mutate(
+    boot_id = rep(
```

```
+      seq(1, (nrow(boot_fullresult)/n_outcomez/n_estimatorz), by = 1
+      ),
+      n_outcomez*n_estimatorz)
+   )   %>%
+   ## convert to wide-format
+   spread(key = estimator, value = boot_est)   %>%
+   mutate(
+     delta_unbiased = ATE - ACDE_gest
+   )
> ## ---------------------------------------------------------------------
> ## Bootstrap means and CIs
> boot_meanz = plot_summary   %>%
+   select(-boot_id)    %>%
+   ## convert to long-format
+   gather(
+     key = estimator, value = boot_est, -(outcome:mediator)
+   )   %>%
+   select(-mediator)   %>%
+   group_by(outcome, estimator)    %>%
+   summarize(
+     boot_mean = mean(boot_est),
+     ci_lower = boot_mean - 1.96*sd(boot_est),
+     ci_upper = boot_mean + 1.96*sd(boot_est)
+   )   %>% ungroup
`summarise()` has grouped output by 'outcome'. You can override using the `.groups` argument.
> ## ---------------------------------------------------------------------
> ## Combine means and CIs, then set colors
> boot_summary = boot_meanz   %>%
+   # bind_cols(boot_ciz)  %>%
+   mutate(
+       point_col = ifelse(boot_mean>0, adjustcolor(colz_vec[1], alpha = alpha_vec[2]), adjustcolor(colz_vec[2], alpha =
alpha_vec[2])),
+     point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+     point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col),
+     estimator_order = rep(c(3, 2, 1, 4, 5), n_outcomez)
+   )   %>%
+   arrange(estimator_order)
> boot_summary = outcome_lbl  %>%
+   left_join(boot_summary, by = c("depvar" = "outcome"))   %>%
+   mutate(row_id = row_number())
> ## ---------------------------------------------------------------------
> ## Figure parameters
> fig_width = 4.85
```

```
> base_cex = .85
> fig_mrgn = c(8.95, 2.4, 1.65, .5) + 0.1   ## figure margin: bottom, left, top, right
> y_range = c(-0.12, 0.12)
> # y_tick = seq(y_range[1], y_range[2], by = 0.05)
> y_tick = seq(-0.1, y_range[2], by = 0.05)
> y_tick_sub = seq(-1, 1, by = 0.01)
> separation_linez = c(seq(5, 30, by = 5) + 0.5, 40)
> separation_linez_sub = seq(3, 33, by = 5) + 0.5
> rect_width = 0.5
> point_est_lwd = 1.5
> ## ---------------------------------------------------------------------
> ## Outcome label
> outcome_pos = seq(3, 33, by = 5)
> outcome_pos_1 = c(1, 3, 5, 7)
> outcome_pos_2 = c(2, 4, 6)
> ## ---------------------------------------------------------------------
> ## Estimator and outcome labels
> estimator_lbl = c("ATE-ACDE", "ATE (conditioning on X)", "ACDE (g-estimator)", "Naive model w/ M", "Naive model w/ M &
Z")
> outcome_lbl_vec = boot_summary  %>%
+   select(outcome_lbl) %>%
+   unique  %>% as.matrix   %>% as.vector
> ## ---------------------------------------------------------------------
> ## Plot
> pdf(file.path(figure_dir, str_c("fig_8a.pdf")), width = plt_ratio(fig_width), height = fig_width)
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(boot_summary)+0.75))
> plot(x = boot_summary$row_id, y = boot_summary$boot_mean, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n",
yaxs = "i", xaxs = "i", xlim = x_range)
> ## Background
> for (i in 1:length(separation_linez_sub)) {
+   rect(xleft = separation_linez_sub[i], xright = separation_linez[i], ytop = 10, ybottom = -10, col = adjustcolor("gray",
alpha = 0.2), lwd = NA)
+ }
> ## Separation lines
> abline(v = separation_linez)
> abline(v = separation_linez_sub, lty = "dashed", lwd = .5)
> ## Rectangles
> rect(xleft = boot_summary$row_id-rect_width/2, xright = boot_summary$row_id+rect_width/2, ybottom = boot_summary$ci_lower,
ytop = boot_summary$ci_upper, col = boot_summary$point_col, border = "black", lwd = 0.5)
> ## Point estimates
> # segments(x0 = boot_summary$row_id-0.32, x1 = boot_summary$row_id+0.32, y0 = boot_summary$boot_mean)
> segments(x0 = boot_summary$row_id-0.2, x1 = boot_summary$row_id+0.2, y0 = boot_summary$boot_mean, lwd = point_est_lwd)
```

```
> ## ------------------------------------------------------------------
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(boot_summary), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -1)
> # axis(1, line = 0, at = separation_linez_sub, label = NA, tck = -1, lty = "dotted")
> ## Estimator label
> colz = c(rep("black", 3), rep("gray60", 2))
> mtext(estimator_lbl, side = 1, line = 0.5, at = 1:nrow(boot_summary), las = 2, cex = 0.8, col = colz)
> ## Outcome labels
> outcome_prefix = c("DV: ", rep("", times = (length(outcome_pos_1)-1)))
> mtext(str_c(outcome_prefix, outcome_lbl_vec[outcome_pos_1]), side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex =
0.8)
> mtext(outcome_lbl_vec[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.8)
> ## ------------------------------------------------------------------
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
> mtext("Effect Estimates (95% bootstrap CIs)", side = 2, line = 1.65, cex = 0.8)
> ## Zero-reference
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> dev.off()
null device
          1
> ## ------------------------------------------------------------------
>
>
>
> ## ------------------------------------------------------------------
> ##
> ## Plot: Summary Comparison Plot
> ## mediator = lnr_kyodojutaku6up
> ##
> ## ------------------------------------------------------------------
> ## Cancel comment-out to read the saved estimates
> # boot_fullresult2_path = list.files(boot_dir_sub2, pattern = ".csv", full.names = TRUE)
> # boot_fullresult2 = read.csv(boot_fullresult2_path)
> ## ------------------------------------------------------------------
> ## Set alpha
```

```
> alpha_level = 0.95
> crit_valz = c((1-alpha_level)/2, 1-(1-alpha_level)/2)
> ## -----------------------------------------------------------------------
> ## Pull out outcome and estimator vectors
> outcome_vec = boot_fullresult2$outcome   %>% unique
> estimator_vec = boot_fullresult2$estimator   %>% unique
> n_estimatorz = estimator_vec    %>% length
> n_outcomez = outcome_vec    %>% length
> ## -----------------------------------------------------------------------
> ## Colors
> # colz_vec = c("blue2", "firebrick2", "gray65", "magenta2")
> # alpha_vec = c(0.6, 0.15, 1)
> colz_vec = c("lightsteelblue4", "lightsteelblue4", "gray65", "magenta2")
> alpha_vec = c(0.6, 0.3, 1)
> ## -----------------------------------------------------------------------
> ## Convert for comparison
> ## -----------------------------------------------------------------------
> ## Compute delta = ATE - ACDE
> plot_summary = boot_fullresult2  %>%
+   mutate(
+     boot_id = rep(
+       seq(1, (nrow(boot_fullresult2)/n_outcomez/n_estimatorz), by = 1
+       ),
+       n_outcomez*n_estimatorz)
+   )   %>%
+   ## convert to wide-format
+   spread(key = estimator, value = boot_est)   %>%
+   mutate(
+     delta_unbiased = ATE - ACDE_gest
+   )
> ## -----------------------------------------------------------------------
> ## Bootstrap means and CIs
> boot_meanz = plot_summary   %>%
+   select(-boot_id)    %>%
+   ## convert to long-format
+   gather(
+     key = estimator, value = boot_est, -(outcome:mediator)
+   )   %>%
+   select(-mediator)   %>%
+   group_by(outcome, estimator)    %>%
+   summarize(
+     boot_mean = mean(boot_est),
+     ci_lower = boot_mean - 1.96*sd(boot_est),
```

```
+    ci_upper = boot_mean + 1.96*sd(boot_est)
+  )   %>% ungroup
`summarise()` has grouped output by 'outcome'. You can override using the `.groups` argument.
> ## ---------------------------------------------------------------------
> ## Combine means and CIs, then set colors
> boot_summary = boot_meanz   %>%
+   # bind_cols(boot_ciz)  %>%
+   mutate(
+      point_col = ifelse(boot_mean>0, adjustcolor(colz_vec[1], alpha = alpha_vec[2]), adjustcolor(colz_vec[2], alpha =
alpha_vec[2])),
+    point_col = ifelse(ci_lower>0 & ci_upper>0, adjustcolor(colz_vec[1], alpha = alpha_vec[1]), point_col),
+    point_col = ifelse(ci_lower<0 & ci_upper<0, adjustcolor(colz_vec[2], alpha = alpha_vec[1]), point_col),
+    estimator_order = rep(c(3, 2, 1, 4, 5), n_outcomez)
+  )   %>%
+   arrange(estimator_order)
> boot_summary = outcome_lbl  %>%
+   left_join(boot_summary, by = c("depvar" = "outcome"))   %>%
+   mutate(row_id = row_number())
> ## ---------------------------------------------------------------------
> ## Figure parameters
> fig_width = 4.85
> base_cex = .85
> fig_mrgn = c(8.95, 2.4, 1.65, .5) + 0.1   ## figure margin: bottom, left, top, right
> y_range = c(-0.12, 0.12)
> # y_tick = seq(y_range[1], y_range[2], by = 0.05)
> y_tick = seq(-0.1, y_range[2], by = 0.05)
> y_tick_sub = seq(-1, 1, by = 0.01)
> separation_linez = c(seq(5, 30, by = 5) + 0.5, 40)
> separation_linez_sub = seq(3, 33, by = 5) + 0.5
> rect_width = 0.5
> point_est_lwd = 1.5
> ## ---------------------------------------------------------------------
> ## Outcome label
> outcome_pos = seq(3, 33, by = 5)
> outcome_pos_1 = c(1, 3, 5, 7)
> outcome_pos_2 = c(2, 4, 6)
> ## ---------------------------------------------------------------------
> ## Estimator and outcome labels
> estimator_lbl = c("ATE-ACDE", "ATE (conditioning on X)", "ACDE (g-estimator)", "Naive model w/ M", "Naive model w/ M &
Z")
> outcome_lbl_vec = boot_summary  %>%
+   select(outcome_lbl) %>%
+   unique  %>% as.matrix   %>% as.vector
```

```
> ## ------------------------------------------------------------------
> ## Plot
> pdf(file.path(figure_dir, str_c("fig_8b.pdf")), width = plt_ratio(fig_width), height = fig_width)
> par(cex = base_cex, mar = fig_mrgn, lend = "square")
> x_range = c(0.25, (nrow(boot_summary)+0.75))
> plot(x = boot_summary$row_id, y = boot_summary$boot_mean, ylim = y_range, axes = FALSE, xlab = NA, ylab = NA, type = "n",
yaxs = "i", xaxs = "i", xlim = x_range)
> ## Background
> for (i in 1:length(separation_linez_sub)) {
+   rect(xleft = separation_linez_sub[i], xright = separation_linez[i], ytop = 10, ybottom = -10, col = adjustcolor("gray",
alpha = 0.2), lwd = NA)
+ }
> ## Separation lines
> abline(v = separation_linez)
> abline(v = separation_linez_sub, lty = "dashed", lwd = .5)
> ## Rectangles
> rect(xleft = boot_summary$row_id-rect_width/2, xright = boot_summary$row_id+rect_width/2, ybottom = boot_summary$ci_lower,
ytop = boot_summary$ci_upper, col = boot_summary$point_col, border = "black", lwd = 0.5)
> ## Point estimates
> # segments(x0 = boot_summary$row_id-0.32, x1 = boot_summary$row_id+0.32, y0 = boot_summary$boot_mean)
> segments(x0 = boot_summary$row_id-0.2, x1 = boot_summary$row_id+0.2, y0 = boot_summary$boot_mean, lwd = point_est_lwd)
> ## ------------------------------------------------------------------
> ## Horizontal axis
> axis(1, line = 0, at = 1:nrow(boot_summary), label = NA, tck = -0.01)
> axis(1, line = 0, at = separation_linez, label = NA, tck = -1)
> # axis(1, line = 0, at = separation_linez_sub, label = NA, tck = -1, lty = "dotted")
> ## Estimator label
> colz = c(rep("black", 3), rep("gray60", 2))
> mtext(estimator_lbl, side = 1, line = 0.5, at = 1:nrow(boot_summary), las = 2, cex = 0.8, col = colz)
> ## Outcome labels
> outcome_prefix = c("DV: ", rep("", times = (length(outcome_pos_1)-1)))
> mtext(str_c(outcome_prefix, outcome_lbl_vec[outcome_pos_1]), side = 3, line = 0, at = outcome_pos[outcome_pos_1], cex =
0.8)
> mtext(outcome_lbl_vec[outcome_pos_2], side = 3, line = 0.85, at = outcome_pos[outcome_pos_2], cex = 0.8)
> ## ------------------------------------------------------------------
> ## Vertical axis
> for (j in c(2, 4)) {
+   axis(j, line = 0, at = y_tick, label = NA)
+   axis(j, line = 0, at = y_tick_sub, label = NA, tck = -0.01)
+ }
> mtext(y_tick, side = 2, line = 0.5, at = y_tick, cex = 0.8)
> mtext("Effect Estimates (95% bootstrap CIs)", side = 2, line = 1.65, cex = 0.8)
> ## Zero-reference
```

```
> abline(h = 0, col = adjustcolor(colz_vec[4], alpha = alpha_vec[3]))
> box()
>
> dev.off()
null device
        1
> ## ------------------------------------------------------------------------
>
>
>
>
>
>
> ## EOF
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Analyses for Appendix B thru K
> ##
> ## ------------------------------------------------------------------------
>
> ## environment setting
> Sys.setenv(LANGUAGE="en")
> gc();gc()
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5235636 279.7   10193134 544.4 10193134 544.4
Vcells 13752164 105.0   31266007 238.6 31266007 238.6
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  5235528 279.7   10193134 544.4 10193134 544.4
Vcells 13752002 105.0   31266007 238.6 31266007 238.6
> rm(list = ls())
> options(scipen = 999)   ## Disable scientific notation
>
> ## ------------------------------------------------------------------------
> ##
> ## Initial settings
> ##
> ## ------------------------------------------------------------------------
> ## Set directory
> ## ------------------------------------------------------------------------
> root_dir = "E:/Dropbox/429_MDA_replication/Replication"
> ## Load packages and functions
```

```
> source(file.path(root_dir, "2_Code/1_PackagesFunctions.R"))
>
> ## ------------------------------------------------------------------------
> ## Data directory
> data_dir = root_dir      %>%    file.path("1_Data") %>%    dir_create()
> ## Output directory
> output_dir = root_dir    %>%    file.path("3_Result")      %>%    dir_create()
> figure_dir = root_dir    %>%    file.path("4_Figures")     %>%    dir_create()
> ## Working directory
> working_dir = root_dir   %>%    file.path("5_Working")     %>%    dir_create()
> ## ------------------------------------------------------------------------
>
> ## additional packages for this file
> needs("ppcor") ## for correlation matrix
> needs("readxl") ## to read xlsx
> needs("lsr") ## for correlate function
> needs("GGally") ## for ggpairs
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table B1
> ## Regress aiming point dummies on the covariates
> ##
> ## ------------------------------------------------------------------------
> ## Load data
> ## ------------------------------------------------------------------------
> ## --- Census data
> tokyo_census_tbl = data_dir     %>%
+   # file.path("TokyoCensusRaid_May2020.rds") %>%
+   # read_rds()     %>%
+   file.path("census_data4.csv") %>%
+   read_csv(col_types = cols()) %>%
+   select(row_id, everything())  %>%
+   filter(year == 2010)    %>%    drop_na(ratio_damage)      %>%
+   mutate(key_code = as.character(key_code)) %>%
+   ## --- Combine the aiming point table
+   left_join(
+     ## --- Aiming points
+     data_dir %>%   file.path("PairwiseDistance_Poly2AP.csv")      %>%
+       read_csv(col_types = cols())     %>%    select(-row_id)     %>%
+       mutate(key_code = as.character(key_code)),
```

```
+    by = "key_code")      %>%
+   rownames_to_column(var = "row_count")      %>%
+   mutate(DistrictFE = as.factor(prewar_district))
>
> ## ------------------------------------------------------------------------
> ##
> ## Variables and models
> ##
> ## ------------------------------------------------------------------------
> ## Outcome
> outcome = "AimingPointDummy"
> ## ------------------------------------------------------------------------
> ## Covariates
> ## ------------------------------------------------------------------------
> ## District FE
> fe_term = "DistrictFE"
> ## Geographical features
> spatial_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway and train stations
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## ------------------------------------------------------------------------
> ## AP-related variables
> far_ap_dummy = c("MoreThanMaxMinAP_PairwiseDist")
> ap_smoothing_termz = list(
+   str_c("poly(", c("log(AimingPointDistance_km_1)", "log(AimingPointDistance_km_2)"), ", degree = 2, simple = TRUE, raw =
TRUE)"),
+   str_c("poly(", c("log(AimingPointDistance_km_1)", "log(AimingPointDistance_km_2)"), ", degree = 3, simple = TRUE, raw =
TRUE)")
+ )
> ## ------------------------------------------------------------------------
> ## Location variables
> distance_termz_base = c(
+   # "minTargetDistance_ln",
+   "palace_dist_ln"
+ )
> polynomial_degree = 5
```

```
> ## --- Polynomials and splines (single dimension)
> distance_term_specification = list(
+   ## --- Polynomials
+   str_c("poly(", distance_termz_base, ", degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   str_c("s(", distance_termz_base, ")"),
+   ## --- none (comparison purposes)
+   "")
> ## --- Lon-lat (two-dimensional) term
> spatial_term_specification = c(
+   ## --- Polynomials
+   str_c("poly(z_lon, z_lat, degree = ",  polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   "te(longitude, latitude)")
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Formula objects
> ## ------------------------------------------------------------------------
> mdl_lst = list(
+   ## Polynomial: felm()-compatible specification
+   polynomial_base_felm = chr2fml_felm(
+     outcome,
+     idv_list = list(
+       spatial_covariates,
+       distance_term_specification[[1]],
+       spatial_term_specification[1]),
+     fe_list = fe_term,
+     se_cluster = "row_count"    ## Robust SE
+     # se_cluster = fe_term      ## Cluster SE
+   ),
+   ## Polynomial: felm()-compatible specification
+   polynomialsquared_felm = chr2fml_felm(
+     outcome,
+     idv_list = list(
+       far_ap_dummy, ap_smoothing_termz[[1]],
+       spatial_covariates,
+       distance_term_specification[[1]],
+       spatial_term_specification[1]),
+     fe_list = fe_term,
+     se_cluster = "row_count"    ## Robust SE
+     # se_cluster = fe_term      ## Cluster SE
+   ),
```

```
+   ## Polynomial: felm()-compatible specification
+   polynomialcubic_felm = chr2fml_felm(
+     outcome,
+     idv_list = list(
+       far_ap_dummy, ap_smoothing_termz[[2]],
+       spatial_covariates,
+       distance_term_specification[[1]],
+       spatial_term_specification[1]),
+     fe_list = fe_term,
+     se_cluster = "row_count"    ## Robust SE
+     # se_cluster = fe_term      ## Cluster SE
+   )
+ )
>
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1: Baseline model w/o any AP info
+   simple_mod = mdl_lst[[1]]     %>%    felm(data   =   tokyo_census_tbl,   keepCX   =   TRUE,   psdef   =   FALSE,   weights   =
tokyo_census_tbl$pop),
+   ## ------------------------------------------------------------------------
+   ## 2: w/ far AP dummy and squared terms
+   squared_mod = mdl_lst[[2]]    %>%    felm(data   =   tokyo_census_tbl,   keepCX   =   TRUE,   psdef   =   FALSE,   weights   =
tokyo_census_tbl$pop),
+   ## ------------------------------------------------------------------------
+   ## 3: w/ far AP dummy and cubic terms
+   cubic_mod = mdl_lst[[3]]      %>%    felm(data   =   tokyo_census_tbl,   keepCX   =   TRUE,   psdef   =   FALSE,   weights   =
tokyo_census_tbl$pop)
+   ## ------------------------------------------------------------------------
+ )
> ## ------------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------------
> stargazer(
+   rslt_lst,
+   type = "text",
+   single.row = TRUE,
+   align = TRUE,
+   covariate.labels = c(
+     "MoreThanMaxMinAP_PairwiseDist" = "Beyond max(Shortest distance between neighboring aiming points)",
+     "poly(log(AimingPointDistance_km_1), degree = 2, simple = TRUE, raw = TRUE)1"="ln(Aiming point distance)",
```

```
+    "poly(log(AimingPointDistance_km_1), degree = 2, simple = TRUE, raw = TRUE)2"="ln(Aiming point distance)sq",
+    "poly(log(AimingPointDistance_km_2), degree = 2, simple = TRUE, raw = TRUE)1"="ln(Second nearest AP distance)",
+    "poly(log(AimingPointDistance_km_2), degree = 2, simple = TRUE, raw = TRUE)2"="ln(Second nearest AP distance)sq",
+    "poly(log(AimingPointDistance_km_1), degree = 3, simple = TRUE, raw = TRUE)1"="ln(Aiming point distance)",
+    "poly(log(AimingPointDistance_km_1), degree = 3, simple = TRUE, raw = TRUE)2"="ln(Aiming point distance)sq",
+    "poly(log(AimingPointDistance_km_1), degree = 3, simple = TRUE, raw = TRUE)3"="ln(Aiming point distance)cu",
+    "poly(log(AimingPointDistance_km_2), degree = 3, simple = TRUE, raw = TRUE)1"="ln(Second nearest AP distance)",
+    "poly(log(AimingPointDistance_km_2), degree = 3, simple = TRUE, raw = TRUE)2"="ln(Second nearest AP distance)sq",
+    "poly(log(AimingPointDistance_km_2), degree = 3, simple = TRUE, raw = TRUE)3"="ln(Second nearest AP distance)cu",
+    "ratio_residential" ="Residential ratio",
+    "poly_area_ln" ="ln(Area)",
+    "n_neighbor" ="Number of neighbor polygons",
+    "PopDensity_1939_km2_ln" ="ln(Population density (1939))",
+    "mean_elevation_ln" ="ln(Mean elevation)",
+    "mean_slope_ln" ="ln(Mean slope)",
+    "river_distance_ln" ="ln(River distance)",
+    "railway_length_ln" ="ln(Railway length)",
+    "n_stations" ="Number of stations",
+    "SpLag_railway_length_ln" ="ln(Railway length), spatial lag",
+    "SpLag_n_stations" ="Number of stations, spatial lag",
+    "poly(palace_dist_ln, degree = 5, simple = TRUE, raw = TRUE)1" ="ln(Imperial Palace distance)",
+    "poly(palace_dist_ln, degree = 5, simple = TRUE, raw = TRUE)2" ="ln(Imperial Palace distance)sq",
+    "poly(palace_dist_ln, degree = 5, simple = TRUE, raw = TRUE)3" ="ln(Imperial Palace distance)cu",
+    "poly(palace_dist_ln, degree = 5, simple = TRUE, raw = TRUE)4" ="ln(Imperial Palace distance)4th",
+    "poly(palace_dist_ln, degree = 5, simple = TRUE, raw = TRUE)5" ="ln(Imperial Palace distance)5th"
+  )
+ )

==================================================================================================================
                                                                  Dependent variable:
                                                        ---------------------------------------------------
                                                                   AimingPointDummy
                                                              (1)            (2)            (3)
------------------------------------------------------------------------------------------------------------------
Beyond max(Shortest distance between neighboring aiming points)                -0.034*** (0.005) -0.014*** (0.004)
ln(Aiming point distance)                                                  -0.058*** (0.013)
ln(Aiming point distance)sq                                                0.093*** (0.013)
ln(Second nearest AP distance)                                              -0.007 (0.019)
ln(Second nearest AP distance)sq                                            -0.016* (0.008)
ln(Aiming point distance)                                                                  -0.030* (0.017)
ln(Aiming point distance)sq                                                               0.067*** (0.013)
ln(Aiming point distance)cu                                                               -0.023*** (0.006)
ln(Second nearest AP distance)                                                              -0.087 (0.060)
```

ln(Second nearest AP distance)sq			0.071 (0.066)
ln(Second nearest AP distance)cu			-0.019 (0.021)
Residential ratio	-0.001 (0.007)	-0.004 (0.007)	-0.0002 (0.006)
ln(Area)	0.011** (0.005)	0.012*** (0.004)	0.007* (0.004)
Number of neighbor polygons	-0.0004 (0.001)	-0.001 (0.001)	-0.001 (0.001)
ln(Population density (1939))	0.005 (0.003)	0.002 (0.002)	0.001 (0.002)
ln(Mean elevation)	0.020 (0.015)	-0.010 (0.011)	-0.006 (0.010)
ln(Mean slope)	-0.039 (0.048)	0.024 (0.030)	0.011 (0.029)
ln(River distance)	0.001 (0.002)	0.001 (0.002)	0.0004 (0.002)
ln(Railway length)	0.005 (0.009)	0.012* (0.007)	0.008 (0.006)
Number of stations	-0.002 (0.003)	-0.003 (0.002)	-0.001 (0.002)
ln(Railway length), spatial lag	-0.008 (0.006)	0.0004 (0.005)	-0.003 (0.004)
Number of stations, spatial lag	0.0004 (0.001)	-0.0003 (0.001)	0.0003 (0.001)
ln(Imperial Palace distance)	0.190 (0.277)	-0.532** (0.253)	-0.259 (0.229)
ln(Imperial Palace distance)sq	-0.452 (0.673)	1.362** (0.586)	0.788 (0.549)
ln(Imperial Palace distance)cu	0.500 (1.020)	-2.125** (0.919)	-0.895 (0.875)
ln(Imperial Palace distance)4th	-0.226 (0.548)	1.124** (0.498)	0.402 (0.477)
ln(Imperial Palace distance)5th	0.053 (0.169)	-0.293** (0.148)	-0.078 (0.143)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)1.0	0.342 (1.532)	-1.901 (1.230)	-0.300 (1.210)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)2.0	-0.685 (3.219)	3.941 (2.606)	0.594 (2.558)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)3.0	-0.019 (0.085)	0.105* (0.058)	0.010 (0.057)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)4.0	0.015 (0.081)	-0.071 (0.061)	0.0004 (0.059)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)5.0	0.001 (0.004)	-0.009*** (0.002)	-0.002 (0.002)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)0.1	-0.249 (1.199)	1.480 (0.962)	0.230 (0.945)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)1.1	-0.002 (0.036)	0.076*** (0.026)	0.003 (0.030)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)2.1	0.029 (0.071)	-0.135*** (0.047)	-0.001 (0.050)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)3.1	0.0003 (0.006)	-0.008* (0.004)	-0.002 (0.005)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)4.1	-0.008 (0.006)	0.020*** (0.004)	-0.00004 (0.004)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)0.2	-0.560 (2.666)	3.274 (2.163)	0.499 (2.123)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)1.2	-0.043 (0.067)	0.073 (0.047)	0.005 (0.046)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)2.2	0.017 (0.135)	-0.122 (0.104)	-0.008 (0.102)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)3.2	0.005 (0.005)	-0.001 (0.005)	0.003 (0.004)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)0.3	0.007 (0.055)	-0.045 (0.037)	-0.008 (0.037)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)1.3	0.004 (0.004)	-0.029*** (0.005)	0.001 (0.005)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)2.3	0.005 (0.007)	0.004 (0.005)	-0.00000 (0.005)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)0.4	0.007 (0.055)	-0.030 (0.042)	-0.002 (0.040)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)1.4	0.005 (0.003)	-0.011*** (0.003)	0.00004 (0.003)
poly(z_lon, z_lat, degree = 5, simple = TRUE, raw = TRUE)0.5	-0.00003 (0.002)	0.005*** (0.001)	0.001 (0.001)
Observations	2,158	2,158	2,158
R2	0.031	0.449	0.486
Adjusted R2	-0.002	0.429	0.467
Residual Std. Error	4.451 (df = 2087)	3.361 (df = 2082)	3.247 (df = 2080)

```
=================================================================================================================
Note:                                                                    *p<0.1; **p<0.05; ***p<0.01
>
> ## ---------------------------------------------------------------------
> ##
> ## Figure C123: balance checks
> ##
> ## ---------------------------------------------------------------------
>
> ## ---------------------------------------------------------------------
> ## Prepare data objects
> ## ---------------------------------------------------------------------
> ## Read data
> tokyo2keep = data_dir    %>%
+   file.path("census_data4.csv") %>%
+   read_csv(col_types = cols()) %>%
+   drop_na(ratio_damage)
>
> ## ---------------------------------------------------------------------
> ## Prepare variables
> ## ---------------------------------------------------------------------
> ## Dependent variables
> ## ---------------------------------------------------------------------
> ## Logged DVs
> depvar_vec = c(
+   "lnr_unemp", "lnrm_unemp", "lnrf_unemp",
+   "lnr_proexe", "lnave_lvlength", "lnave_eduyr"
+ )
> ## ---------------------------------------------------------------------
> ## Logged negative controls
> negative_cntrlz = c("lnrmale0004")
> ## ---------------------------------------------------------------------
> ## Dependent variable vector (all DVs and negative controls combined): 22 DVs
> depvar_vec_combined = c(depvar_vec, negative_cntrlz)
> ## Dependent variables: for alternative treatment indicators
> depvar_vec_alternative_treatment = c(depvar_vec, negative_cntrlz)
>
> ## ---------------------------------------------------------------------
> ## Treatment variables
> ## ---------------------------------------------------------------------
> treatment_vec = c("ratio_damage", "ln_damage_ratio", "binary_damage_ratio")
> robust_treatment_vec = c(treatment_vec[2:3], str_c("as.factor(", treatment_vec[1],")"))
> ## ---------------------------------------------------------------------
```

```
> ## Pretreatment covariates
> ## ------------------------------------------------------------------------
> ## Distance variables
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> ## Polynomial
> polynomial_degree = 5
> # distance_termz_poly = polynomial_varname(distance_termz_base, degree = polynomial_degree)
> distance_termz_poly = str_c("poly(", distance_termz_base, ", degree = ", polynomial_degree, ", simple = TRUE, raw =
TRUE)")
> distance_term_specification = list( distance_termz_poly, str_c("s(", distance_termz_base, ")") )
> ## Geographical features
> spatial_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## Flammability score
> flame_scorez = c("hazard", "max_nghb_hazard")
> ## ------------------------------------------------------------------------
> ## Spatial spline/polynomials and fixed effects
> ## ------------------------------------------------------------------------
> ## District FEs
> district_fe = "as.factor(prewar_district)"   ## 35-district FE
> ## Spatial terms
> spatial_term_lbl = c("Polynomial", "GAM")    ## This indicates polynomial or spline specification in the csv files
> lonlat_base = c("z_lon", "z_lat")
> spatial_polynomial = str_c("poly(z_lon, z_lat, degree = ",  polynomial_degree, ", simple = TRUE, raw = TRUE)")
> spatial_term_specification = list(spatial_polynomial, "te(longitude, latitude)")
> ## ------------------------------------------------------------------------
> ## Pretreatment covariates to plot for balance checking
> balance_varz = c(spatial_covariates, distance_termz_base, spatial_polynomial[1:2])
>
>
>
> ## ------------------------------------------------------------------------
> ## Figure parameters
> ## ------------------------------------------------------------------------
> set.seed(123)
```

```
> fig_height = 3.85
> base_cex = .75
> ax_lbl_cex = 0.8
> fig_mrgn = c(3, 3, 1, 0) + 0.3   ## figure margin: bottom, left, top, right
> boxplot_colz = c(
+   adjustcolor("lightsteelblue2", alpha = 0.5),
+   adjustcolor("lightsteelblue4", alpha = 0.85),
+   adjustcolor("maroon2", alpha = 0.65)
+ )
> ## ---------------------------------------------------------------------
> ## Prepare data
> ## ---------------------------------------------------------------------
> boxplot_data = tokyo2keep %>%
+   ## 2010 observation because the spatial variables are based on the 2010 census polygons
+   filter(year == 2010)    %>%
+   select(-year)
> boxplot_obs = boxplot_data  %>%
+   group_by(ratio_damage)  %>%
+   summarize(n_obs = n())  %>%
+   select(n_obs)   %>%
+   as.matrix   %>%
+   as.vector
> ## ---------------------------------------------------------------------
> ## Figure C2 & C3: Balance Check ---
> ## pretreatment cov & sp-lagged pretreatment cov
> ## * See below for Figure C1
> ## ---------------------------------------------------------------------
> balance_varz_lbl = c(
+   "Residential Ratio", "ln Area", "N Neighboring Census Units", "ln Elevation", "ln Slope",
+   "ln River Distance (km)", "ln Railway Length (km)", "ln N Stations", "ln Railway Length (km) Spatial Lag", "ln N Stations
Spatial Lag",
+   "ln Palace Distance (km)", "ln min(Target Distance) (km)",
+   "Longitude", "Latitude"
+    # "ln Palace Distance (km, standardized)", "ln min(Target Distance) (km, standardized)", "Longitude (standardized)",
"Latitude (standardized)"
+ )
>
> ## ---------------------------------------------------------------------
> ## Residualized plot
> ## ---------------------------------------------------------------------
> png_resolution = 360
> fig_height = 3.85
> fig_mrgn = c(2.45, 2.35, 0.3, 0.3) + 0.3   ## figure margin: bottom, left, top, right
```

```
> xlimz = c(-3,3)
> ylimz = c(-6,6)
> base_cex = 1.2
> ax_lbl_cex = 1.25
> ## Pretreatment covariates
> pretreatment_combined = c(spatial_covariates, distance_term_specification[[1]], spatial_term_specification[[1]])
>
> ## Figure name label
> bc_label = c("fig_c2_a", "fig_c2_e", "fig_c3_a", "fig_c2_b", "fig_c2_f", "fig_c2_g", "fig_c2_h", "fig_c2_i", "fig_c2_j",
"fig_c3_b", "fig_c3_c", "fig_c2_c", "fig_c2_d")
>
> ## Loop over covariates
> for (i in 1:(length(balance_varz)-2)) {
+
+   ## Print progress
+   cat(str_c("Pretreatment covariate: ", i, "/", (length(balance_varz)-2), " --- done. ¥r") )
+   ## -----------------------------------------------------------------------
+   ## Pull out covariate
+   tmp_covariate = balance_varz[i] ## ith covariate
+    tmp_pretreatment = pretreatment_combined[!str_detect(pretreatment_combined, tmp_covariate)]   ## Drop ith pretreatment
covariate
+   ## Prepare data
+   residual_raid_fml = chr2fml("ratio_damage", idv_list = list(tmp_pretreatment))
+   residual_x_fml = chr2fml(tmp_covariate, idv_list = list(tmp_pretreatment))
+   ## Pull out residuals
+   residual_raid = lm(residual_raid_fml,
+                      data = boxplot_data %>%
+                        mutate_at(
+                          vars("ratio_damage", tmp_covariate),
+                          funs((.-mean(.))/sd(.))
+                        ),
+                      weights = pop
+   )   %>% residuals
+   residual_x = lm(residual_x_fml,
+                   data = boxplot_data %>%
+                     mutate_at(
+                       vars("ratio_damage", tmp_covariate),
+                       funs((.-mean(.))/sd(.))
+                     ),
+                   weights = pop
+   )   %>% residuals
+   ## -----------------------------------------------------------------------
+   ## Figure path
```

```
+   # tmp_path = file.path(figure_dir, str_c(i, "_", tmp_covariate, "_residualized.png"))
+   tmp_path = file.path(figure_dir, str_c(bc_label[i], ".png"))
+   png(tmp_path, width = fig_height, height = sqrt(2)*fig_height, units = "in", res = 240)
+   # dev.new(width = fig_height, height = sqrt(2)*fig_height)
+   par(cex = base_cex, mar = fig_mrgn)
+   ## ------------------------------------------------------------------------
+   ## Scatter plot
+   pt_cex_weight = boxplot_data$pop/1000
+   plot(residual_raid, y = residual_x,
+        pch = 21, bg = boxplot_colz[1], col = boxplot_colz[2], lwd = 0.25,
+        cex = .25*pt_cex_weight,
+        xlim = xlimz, ylim = ylimz,
+        xaxs = "i", yaxs = "i", axes = FALSE, xlab = NA, ylab = NA
+   )
+   ## ------------------------------------------------------------------------
+   ## Regress and add notations
+   ## Zero-reference
+   abline(h=0, col = adjustcolor("black", alpha = 0.5), lwd = 0.65)
+   ## Regression slope
+   tmp_res_line = lm(residual_x ~ residual_raid, weights = boxplot_data$pop)
+   abline(tmp_res_line, col = "maroon2")
+   tmp_info = model_info_lm(tmp_res_line, x2extract = "residual_raid")
+   tmp_res_lbl = tmp_info[1] %>% round(3)
+   tmp_res_lbl = str_c("Slope = ", tmp_res_lbl)
+   text(x = max(residual_raid)*0.1, y = 5.25, labels = tmp_res_lbl, cex = 1, col = "maroon2", pos = 3)
+
+   ## Axis etc.
+   for (j in 1:2) {
+     # tick = seq(-2, 2, by = 0.2)
+     axis(j, line = 0, label = NA)
+     axis(j, line = 0, label = NA, at = seq(-6,6, by = 0.5), tck = -0.02)
+     axis(j, line = 0, label = NA, at = seq(-6,6, by = 0.1), tck = -0.01)
+     axis(j, line = -0.5, tick = FALSE, at = seq(-6,6, by = 1))
+   }
+   mtext("Raid Damage (residualized)", side = 1, line = 1.65, cex = ax_lbl_cex)
+   mtext(str_c(balance_varz_lbl[i], " (residualized)"), side = 2, line = 1.65, cex = ax_lbl_cex)
+   box()
+   ## Close
+   dev.off()
+ }
Warning messages:riate: 13/13 --- done.
1: Using an external vector in selections was deprecated in tidyselect 1.1.0.
ℹ Please use `all_of()` or `any_of()` instead.
```

```
  # Was:
  data %>% select(tmp_covariate)

  # Now:
  data %>% select(all_of(tmp_covariate))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
2: `funs()` was deprecated in dplyr 0.8.0.
i Please use a list of either functions or lambdas:

# Simple named list: list(mean = mean, median = median)

# Auto named with `tibble::lst()`: tibble::lst(mean, median)

# Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
>
> ## ------------------------------------------------------------------------
> ## Figure C1: Balance Check --- Fire hazard
> ## ------------------------------------------------------------------------
> hazard_varz = c("hazard", "min_nghb_hazard", "max_nghb_hazard")
> ## Pretreatment covariates --- EXCLUDING the neighbor hazard ratings
> pretreatment_combined = c(spatial_covariates, distance_term_specification[[1]], spatial_term_specification[[1]])
> ## Prepare data
> hazard_data = tokyo2keep   %>%
+   ## 2010 observation because the spatial variables are based on the 2010 census polygons
+   filter(
+     year == 2010, !is.na(hazard),
+     !is.na(min_nghb_hazard), !is.na(max_nghb_hazard)
+   )   %>%
+   select(-year)
> ## Variable names in figures
> hazard_varz_lbl = c("Fire Hazard", "min(Neighbor Fire Hazard)", "max(Neighbor Fire Hazard)")
>
> ## Figure name label
> bc_label2 = c("fig_c1_a", "fig_c1_b", "fig_c1_c")
>
> ## Loop over covariates
> for (i in 1:length(hazard_varz)) {
+   ## Print progress
```

```
+   cat(str_c("Pretreatment covariate: ", i, "/", length(hazard_varz), " --- done. ¥r") )
+   ## -------------------------------------------------------------------------
+   ## Pull out covariate
+   tmp_covariate = hazard_varz[i] ## ith covariate
+   tmp_pretreatment = pretreatment_combined[!str_detect(pretreatment_combined, tmp_covariate)]   ## Drop ith pretreatment
covariate
+   ## Prepare data
+   residual_raid_fml = chr2fml("ratio_damage", idv_list = list(tmp_pretreatment))
+   residual_x_fml = chr2fml(tmp_covariate, idv_list = list(tmp_pretreatment))
+   ## Pull out residuals
+   residual_raid = lm(residual_raid_fml,
+                      data = hazard_data %>%
+                        mutate_at(
+                          vars("ratio_damage", tmp_covariate),
+                          funs((.-mean(.))/sd(.))
+                        ),
+                      weights = pop
+   )   %>% residuals
+   residual_x = lm(residual_x_fml,
+                   data = hazard_data %>%
+                     mutate_at(
+                       vars("ratio_damage", tmp_covariate),
+                       funs((.-mean(.))/sd(.))
+                     ),
+                   weights = pop
+   )   %>% residuals
+   ## -------------------------------------------------------------------------
+   ## Figure path
+   # tmp_path = file.path(figure_dir, str_c(i + length(balance_varz)-2, "_", tmp_covariate, "_residualized.png"))
+   tmp_path = file.path(figure_dir, str_c(bc_label2[i], ".png"))
+   png(tmp_path, width = fig_height, height = sqrt(2)*fig_height, units = "in", res = 240)
+   # dev.new(width = fig_height, height = sqrt(2)*fig_height)
+   par(cex = base_cex, mar = fig_mrgn)
+   ## -------------------------------------------------------------------------
+   ## Scatter plot
+   pt_cex_weight = hazard_data$pop/1000
+   plot(residual_raid, y = residual_x,
+        pch = 21, bg = boxplot_colz[1], col = boxplot_colz[2], lwd = 0.25,
+        cex = .25*pt_cex_weight,
+        xlim = xlimz, ylim = ylimz,
+        xaxs = "i", yaxs = "i", axes = FALSE, xlab = NA, ylab = NA
+   )
+   ## -------------------------------------------------------------------------
```

```
+   ## Regress and add notations
+   ## Zero-reference
+   abline(h=0, col = adjustcolor("black", alpha = 0.5), lwd = 0.65)
+   ## Regression slope
+   tmp_res_line = lm(residual_x ~ residual_raid, weights = hazard_data$pop)
+   abline(tmp_res_line, col = "maroon2")
+   tmp_info = model_info_lm(tmp_res_line, x2extract = "residual_raid")
+   tmp_res_lbl = tmp_info[1] %>% round(3)
+   tmp_res_lbl = str_c("Slope = ", tmp_res_lbl)
+   ## Add text
+   text(x = max(residual_raid)*0.1, y = 5.25, labels = tmp_res_lbl, cex = 1, col = "maroon2", pos = 3)
+   ## Raw regression
+   # tmp_res_line = lm(chr2fml(tmp_covariate, idv_list = list("ratio_damage")), data = hazard_data)
+   # abline(tmp_res_line, col = boxplot_colz[3], lty = "dashed")
+   ## Axis etc.
+   for (j in 1:2) {
+     # tick = seq(-2, 2, by = 0.2)
+     axis(j, line = 0, label = NA)
+     axis(j, line = 0, label = NA, at = seq(-6,6, by = 0.5), tck = -0.02)
+     axis(j, line = 0, label = NA, at = seq(-6,6, by = 0.1), tck = -0.01)
+     axis(j, line = -0.5, tick = FALSE, at = seq(-6,6, by = 1))
+   }
+   mtext("Raid Damage (residualized)", side = 1, line = 1.65, cex = ax_lbl_cex)
+   mtext(str_c(hazard_varz_lbl[i], " (residualized)"), side = 2, line = 1.65, cex = ax_lbl_cex)
+   box()
+   ## Close
+   dev.off()
+ }
> etreatment covariate: 3/3 --- done.
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure E1: Histogram of the number of pictures taken on each day.
> ## Actual figure was produced by Stata and this code produces equivalent one.
> ##
> ## ------------------------------------------------------------------------
>
> ## LOAD MASTER DATA
> data = data_dir %>%
+   file.path("Aerial_Photo_Freq.dta")   %>%
+   read_dta()
>
> # Create the histogram using ggplot2
```

```
> plot <- ggplot(data, aes(x = date2)) +
+   geom_histogram(binwidth = 2, fill = "steelblue", color = NA) +
+   labs(title = "Number of Photos in the Dataset",
+        x = "Date",
+        y = "Frequency") +
+   theme_minimal() +
+   theme(plot.background = element_rect(fill = "white"),
+         panel.background = element_rect(fill = "white"),
+         panel.grid = element_blank(),
+         axis.line = element_line(color = "black"))
>
> # Save the plot as a PDF
> ggsave(figure_dir %>% file.path("figure_e1.pdf"), plot, width = 8, height = 5)
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure H3
> ## correlation matrix for the intercoder validity
> ##
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Load data
> ## ------------------------------------------------------------------------
> ## human coding shape file
> raid_shape = data_dir    %>%
+   list.files(pattern = "raid_shape.shp$", full.names = TRUE)      %>%
+   read_sf() %>%
+   rename(key_code = key_cod) %>%
+   ## create damage from map variable
+   mutate(dmg_map = NA)
>
> ## WARINING: THE FOLLOWING LOOP TAKES ABOUT ONE HOUR TO FINISH.
> ## loop over 3 layers RGB
> for (b in 1:3){
+   ## crop using raster package
+   for (i in 1:dim(raid_shape)[1]){
+     raid_shape[i,9] =
+       data_dir %>%
+       list.files(pattern = "georeffed_red.tif$", full.names = TRUE) %>%
+       ## load red part of damage map
```

```
+      raster(band = b) %>%
+      crop(extent(raid_shape[i,])) %>%
+      mask(raid_shape[i,]) %>%
+      ## need to use raster package's as.matrix()
+      raster::as.matrix() %>%
+      as.numeric() %>%
+      mean(na.rm = T)
+    cat("i =", i, "¥r")
+  }
+  write_rds(raid_shape,file = file.path(working_dir, str_c("raid_shape_dmg_map_band", b,".rds")))
+ }
> = 2324
> ## human coding data file
> humancoder = data_dir    %>%
+  list.files(pattern = "eval1.xlsx$", full.names = TRUE)   %>%
+  read_excel(sheet="eval1") %>%
+  mutate(dmg_coder1 = damage/10) %>%
+  select(row_id, dmg_coder1)
>
> ## master data file
> raid_data = data_dir     %>%
+  file.path("census_data4.csv") %>%
+  read_csv(col_types = cols()) %>%
+  mutate(key_code = as.character(key_code))
>
> ## trim master data
> raid_data_s = raid_data %>%
+  filter(year==2010) %>%
+  select(key_code, row_id, ratio_damage, ratio_residential) %>%
+  left_join(humancoder, by = "row_id")
>
> ## map based damage variable
> raid_merged =
+  ## band=1
+  working_dir      %>%
+  file.path("raid_shape_dmg_map_band1.rds")  %>%
+  read_rds() %>%
+  select(key_code,dmg_map) %>%
+  rename(dmg_map_b1 = dmg_map) %>%
+  ## merge band=2
+  left_join(
+    working_dir    %>%
+      file.path("raid_shape_dmg_map_band2.rds")      %>%
```

```
+      read_rds() %>%
+      as_tibble() %>%
+      select(key_code,dmg_map) %>%
+      mutate(dmg_map = -1*dmg_map) %>%
+      rename(dmg_map_b2 = dmg_map),
+    by = "key_code"
+  ) %>%
+  ## merge band=3
+  left_join(
+    working_dir    %>%
+      file.path("raid_shape_dmg_map_band3.rds")      %>%
+      read_rds() %>%
+      as_tibble() %>%
+      select(key_code,dmg_map) %>%
+      mutate(dmg_map = -1*dmg_map) %>%
+      rename(dmg_map_b3 = dmg_map),
+    by = "key_code"
+  ) %>%
+  ## adding up all color elements
+  mutate(dmg_map_b123 = dmg_map_b1+dmg_map_b2+dmg_map_b3) %>%
+  ## drop duplicated entries
+  group_by(key_code,dmg_map_b123) %>%
+  slice_head() %>%
+  ungroup() %>%
+  ## merge human code data
+  left_join(raid_data_s, by = "key_code") %>%
+  ## map based
+  mutate(dmg_map_b123_0to1 = (dmg_map_b123-min(dmg_map_b123))/(max(dmg_map_b123)-min(dmg_map_b123))) %>%
+  ## original
+  mutate(ratio_dmg = ratio_damage * ratio_residential) %>%
+  ## student
+  mutate(ratio_dmg_coder1 = dmg_coder1 * ratio_residential)
>
>
> # delete unnecessary variables to make ggpair
> raid_merged_s = raid_merged %>%
+  as_tibble() %>%
+  select(ratio_dmg, dmg_map_b123_0to1, ratio_dmg_coder1)
>
>
>
> ## ------------------------------------------------------------------------
> ## CALCULATE CORRELATION AND GRAPH
```

```
> ## ------------------------------------------------------------------------
>
> ## produce correlogram in PDF with ggpairs
> file_cor_mat = figure_dir %>%
+   file.path(str_c("cor_mat.pdf"))
> pdf(file_cor_mat, height = 3, width = 3)
> g <- ggpairs(raid_merged_s,
+              lower = list(
+                continuous = wrap("points",
+                                  size = .5,
+                                  color = "blue",
+                                  alpha = 0.05)),
+              columnLabels = c("Author", "Map", "Student")) +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> print(g)
Warning messages:
1: Removed 69 rows containing non-finite values (`stat_density()`).
2: In ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,  :
  Removed 69 rows containing missing values
3: In ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,  :
  Removed 69 rows containing missing values
4: Removed 69 rows containing missing values (`geom_point()`).
5: In ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,  :
  Removed 69 rows containing missing values
6: Removed 69 rows containing missing values (`geom_point()`).
7: Removed 69 rows containing missing values (`geom_point()`).
8: Removed 69 rows containing non-finite values (`stat_density()`).
> dev.off()
null device
          1
>
> ## Code for partial correlations in Appendix H
> raid_merged_s2 = raid_merged_s %>%
+   filter(!is.na(ratio_dmg)) %>%
+   filter(!is.na(dmg_map_b123_0to1)) %>%
+   filter(!is.na(ratio_dmg_coder1))
>
> pcor(raid_merged_s2)$estimate
                ratio_dmg dmg_map_b123_0to1 ratio_dmg_coder1
ratio_dmg       1.0000000        0.6810820       0.6577084
dmg_map_b123_0to1 0.6810820        1.0000000      -0.1021951
ratio_dmg_coder1  0.6577084       -0.1021951       1.0000000
>
```

```
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Figure I1
> ##
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Analysis for Ordinary household population
> ## ------------------------------------------------------------------------
>
> ## Load data set
> df = data_dir     %>%
+   file.path("keishicho-tokeisho.xlsx") %>%
+   readxl::read_xlsx(sheet="1940-1944")
>
> ##
> ## Male population
> ##
>
> ## Calculate correlations between 1940 & 1944
> cor(df$futsupopm1940,df$futsupopm1944)
[1] 0.9515412
>
> ## Creating scatter plots
> ggplot(df, aes(x=futsupopm1940, y=futsupopm1944)) +
+   geom_smooth(method=lm, se=T, color='magenta', size=.1) +
+   geom_point(alpha=0.33, color='skyblue3', stroke=0, size=3) +
+   ylim(-5000, 120000) +
+   xlim(0, 125000) +
+   labs(x="Population as of December 1940",
+        y="Population as of December 1944") +
+   theme(legend.title=element_blank(),
+         legend.key=element_rect(fill='NA')) +
+   annotate("text", x = 90000, y = 2500,
+            label = expression("N=80, " ~ rho ~ " = 0.95"))
`geom_smooth()` using formula = 'y ~ x'
Warning messages:
1: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
This warning is displayed once every 8 hours.
```

```
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
2: In is.na(x) :
  is.na() applied to non-(list or vector) of type 'expression'
>
> ggsave(file.path(figure_dir, "a_male_ordinaryHH.pdf"),
+       width = 4,
+       height = 4)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
> ##
> ## Female population
> ##
>
> ## Calculate correlations between 1940 & 1944
> cor(df$futsupopf1940,df$futsupopf1944)
[1] 0.9613205
>
> ## Creating scatter plots
> ggplot(df, aes(x=futsupopf1940, y=futsupopf1944)) +
+   geom_smooth(method=lm, se=T, color='magenta', size=.1) +
+   geom_point(alpha=0.33, color='skyblue3', stroke=0, size=3) +
+   ylim(-5000, 120000) +
+   xlim(0, 125000) +
+   labs(x="Population as of December 1940",
+        y="Population as of December 1944") +
+   theme(legend.title=element_blank(),
+         legend.key=element_rect(fill='NA')) +
+   annotate("text", x = 90000, y = 2500,
+            label = expression("N=80, " ~ rho ~ " = 0.96"))
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
> ggsave(file.path(figure_dir, "b_female_ordinaryHH.pdf"),
+       width = 4,
+       height = 4)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
>
```

```
> ## ------------------------------------------------------------------------
> ## Analysis for Non-ordinary household population
> ## ------------------------------------------------------------------------
>
> ## calculate non-ordinary household popuation
> df = df %>%
+   mutate(junpopm1940 = popm1940 - futsupopm1940) %>%
+   mutate(junpopf1940 = popf1940 - futsupopf1940) %>%
+   mutate(junpopm1944 = popm1944 - futsupopm1944) %>%
+   mutate(junpopf1944 = popf1944 - futsupopf1944)
>
> ##
> ## Male population
> ##
>
> ## Calculate correlations between 1940 & 1944
> cor(df$junpopm1940,df$junpopm1944)
[1] 0.8695896
>
> ## Creating scatter plots
> ggplot(df, aes(x=junpopm1940, y=junpopm1944)) +
+   geom_smooth(method=lm, se=T, color='magenta', size=.1) +
+   geom_point(alpha=0.33, color='skyblue3', stroke=0, size=3) +
+   # ylim(0, 250000) +
+   # xlim(0, 250000) +
+   labs(x="Population as of December 1940",
+        y="Population as of December 1944") +
+   theme(legend.title=element_blank(),
+         legend.key=element_rect(fill='NA')) +
+   annotate("text", x = 9000, y = 600,
+            label = expression("N=80, " ~ rho ~ " = 0.87")) +
+   ## adding label for outliers: Kamata
+   geom_label(
+     label="Kamata",
+     x=10000,
+     y=20000,
+     label.padding = unit(0.3, "lines"), # Rectangle size around label
+     label.size = 0.35,
+     color = "black",
+     fill="#FFFFFF",
+     size = 3
+   )
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
> ggsave(file.path(figure_dir, "c_male_nonordinaryHH.pdf"),
+       width = 4,
+       height = 4)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
> ##
> ## Female population
> ##
>
> ## Calculate correlations between 1940 & 1944
> cor(df$junpopf1940,df$junpopf1944)
[1] 0.6453443
>
> ## Creating scatter plots
> ggplot(df, aes(x=junpopf1940, y=junpopf1944)) +
+   geom_smooth(method=lm, se=T, color='magenta', size=.1) +
+   geom_point(alpha=0.33, color='skyblue3', stroke=0, size=3) +
+   # ylim(0, 250000) +
+   # xlim(0, 250000) +
+   labs(x="Population as of December 1940",
+        y="Population as of December 1944") +
+   theme(legend.title=element_blank(),
+         legend.key=element_rect(fill='NA')) +
+   annotate("text", x = 2750, y = -300,
+            label = expression("N=80, " ~ rho ~ " = 0.65")) +
+   ## adding label for outliers: Kamata
+   geom_label(
+     label="Kamata",
+     x=2680,
+     y=7800,
+     label.padding = unit(0.3, "lines"), # Rectangle size around label
+     label.size = 0.35,
+     color = "black",
+     fill="#FFFFFF",
+     size = 3
+   ) +
+   ## adding label for outliers: Kameido
+   geom_label(
```

```
+    label="Kameido",
+    x=2950,
+    y=400,
+    label.padding = unit(0.3, "lines"), # Rectangle size around label
+    label.size = 0.35,
+    color = "black",
+    fill="#FFFFFF",
+    size = 3
+  )
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
> ggsave(file.path(figure_dir, "d_female_nonordinaryHH.pdf"),
+       width = 4,
+       height = 4)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
In is.na(x) : is.na() applied to non-(list or vector) of type 'expression'
>
>
>
> ## ------------------------------------------------------------------------
> ##
> ## Table J1
> ## Result VII: Placebo regressions (Sport)
> ##
> ## ------------------------------------------------------------------------
>
> ## ------------------------------------------------------------------------
> ## Load data
> ## ------------------------------------------------------------------------
> tokyo_census_tbl = data_dir     %>%    file.path("census_data4.csv")     %>%
+   read_csv(col_types = cols())  %>%
+   mutate(postwar_district23 = cityName ) %>%
+   mutate(key_code = as.character(key_code)) %>%
+   mutate(zipcode = str_replace(zipcode, "-", "")) %>%
+   filter(year == 2015)
>
>
> ## --- All years
> tokyo_ninka_tbl = data_dir     %>%    file.path("test_ninka_2.dta")     %>%    read_dta()    %>%
+   rename(
```

```
+    dummy_ninka = d_ninka,
+    ln_ninka_days = lnninka_days
+  )
> ## --- Since 1993
> tokyo_ninka_tbl_1993 = data_dir %>%    file.path("test_ninka_sinceFY1993_2.dta")      %>%    read_dta()    %>%
+  rename(
+    dummy_ninka = d_ninka,
+    ln_ninka_days = lnninka_days
+  )
> ## --- Sports
> # sports_tbl = root_dir  %>%    file.path("@@SUBMISSION/JOP_RR1/supporting_materials/R2_Q5a/sports_club_list.csv")
      %>%
> sports_tbl = data_dir    %>%    file.path("sports_club_list.csv") %>%
+  read_csv(col_types = cols())  %>%
+  rename(
+    club_name = 1, cityName = ku_name,
+    chochoazaName = address_jp) %>%
+  mutate(
+    chochoazaName = if_else(
+      str_detect(chochoazaName, pattern = "¥¥d"),
+      true = str_c(chochoazaName, "丁目"),
+      false = chochoazaName),
+    chochoazaName = stringi::stri_trans_general(chochoazaName, id = "Halfwidth-Fullwidth"),
+    SportClubDummy = 1,
+    zipcode = str_remove_all(zipcode, pattern = "-")
+  )
>
>
>
>
> ## ------------------------------------------------------------------------
> ## Link function
> ## ------------------------------------------------------------------------
> binom_link = binomial(link = logit)
> count_link = quasipoisson
> ## ------------------------------------------------------------------------
> ## Outcome and treatment variables
> ## ------------------------------------------------------------------------
> depvar_vec_combined = c("ninka_days", "ln_ninka_days", "dummy_ninka", "n_ninka")
> placebo_depvar_vec = c("SportClubDummy", "LocalSportClubDummy")
> treatment_vec = c("ratio_damage", "ln_damage_ratio", "binary_damage_ratio")
> ## ------------------------------------------------------------------------
> ## Covariates
```

```
> ## ------------------------------------------------------------------------
> ## District FE
> fe_term = "DistrictFE"
> ## Geographical features
> spatial_covariates = c(
+   ## Residential ratio and geographical area
+   "ratio_residential", "poly_area_ln", "n_neighbor",
+   ## Prewar population density
+   "PopDensity_1939_km2_ln",
+   ## Terrain variables
+   "mean_elevation_ln", "mean_slope_ln", "river_distance_ln",
+   ## Railway and train stations
+   "railway_length_ln", "n_stations", "SpLag_railway_length_ln", "SpLag_n_stations"
+ )
> ## ------------------------------------------------------------------------
> ## Location variables
> distance_termz_base = c("palace_dist_ln", "minTargetDistance_ln")
> polynomial_degree = 5
> ## --- Polynomials and splines (single dimension)
> distance_term_specification = list(
+   ## --- Polynomials
+   str_c("poly(", distance_termz_base, ", degree = ", polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   str_c("s(", distance_termz_base, ")"),
+   ## --- none (comparison purposes)
+   "",
+   ## --- 3rd order polynomials
+   str_c("poly(", distance_termz_base, ", degree = 3, simple = TRUE, raw = TRUE)")
+ )
> ## --- Lon-lat (two-dimensional) term
> spatial_term_specification = c(
+   ## --- Polynomials
+   str_c("poly(z_lon, z_lat, degree = ",  polynomial_degree, ", simple = TRUE, raw = TRUE)"),
+   ## --- Splines
+   "te(longitude, latitude)",
+   ## --- none (comparison purposes)
+   "",
+   ## --- 3rd order polynomials
+   "poly(z_lon, z_lat, degree = 3, simple = TRUE, raw = TRUE)"
+ )
>
>
> ## ------------------------------------------------------------------------
```

```
> ## Combine the post-1993 records and full records
> ## ------------------------------------------------------------------------
> depvar_1993_tbl = tokyo_ninka_tbl_1993      %>%
+   select(
+     key_code, chochoazaName, all_of(depvar_vec_combined)
+   )  %>%
+   rename_at(vars(ninka_days:n_ninka), ~ str_c(., "_1993"))
> ## ------------------------------------------------------------------------
> tmp_combined_tbl = tokyo_census_tbl   %>%
+   left_join(
+     tokyo_ninka_tbl      %>%
+       select(key_code, chochoazaName, all_of(depvar_vec_combined))
+   )  %>%
+   left_join(depvar_1993_tbl, by = c("key_code", "chochoazaName"))%>%
+   left_join(sports_tbl, by = c("cityName", "chochoazaName", "zipcode")) %>%
+   rownames_to_column(var = "row_count")      %>%
+   mutate(
+     DistrictFE = as.factor(prewar_district),
+     SportClubDummy = if_else(is.na(SportClubDummy), true = 0, false = SportClubDummy)
+   )  %>%
+   select(row_count, everything())
Joining, by = c("key_code", "chochoazaName")
>
> ## ------------------------------------------------------------------------
> ## Spatially-lagged variables
> ## ------------------------------------------------------------------------
> full_poly = data_dir     %>%    file.path("RaidShp_May2020b.rds") %>%
+   read_rds()       %>%    select(key_code, ratio_damage)    %>%
+   left_join(tmp_combined_tbl    %>%    select(key_code, SportClubDummy)) %>%
+   mutate(
+     SportClubDummy = if_else(is.na(SportClubDummy), true = 0, false = SportClubDummy)
+   )
Joining, by = "key_code"
> ## Use different SWM
> full_contig_nb = poly2nb(full_poly, queen = TRUE)
> full_contig_listw = nb2listw(full_contig_nb, style = "W", zero.policy = TRUE) ## 1 neighborhood without neighbors
> ## Lagged damage SWM
> full_contig_nb_damage = poly2nb(full_poly    %>%    drop_na(ratio_damage), queen = TRUE)
> full_contig_listw_damage = nb2listw(full_contig_nb_damage, style = "W", zero.policy = TRUE)##   1   neighborhood   without
neighbors
> ## ------------------------------------------------------------------------
> ## Lagged damage
> lagged_damage_tbl = full_poly   %>%    drop_na(ratio_damage)      %>%
```

```
+   mutate(
+     LaggedDamage = lag.listw(full_contig_listw_damage, var = full_poly  %>%    drop_na(ratio_damage)      %>%
       pull(ratio_damage),
+                          zero.policy = TRUE)
+   )  %>%
+   as_tibble()      %>%
+   select(key_code, LaggedDamage)
> ## Construct lagged outcome
> lagged_variables_tbl = full_poly      %>%
+   mutate(
+     LocalSportClubDummy = lag.listw(full_contig_listw, var = full_poly$SportClubDummy, zero.policy = TRUE, NAOK = TRUE),
+     LocalSportClubDummy = if_else(SportClubDummy + LocalSportClubDummy > 0, true = 1, false = 0)
+   )  %>%
+   as_tibble()      %>%
+   select(key_code, SportClubDummy, LocalSportClubDummy)    %>%
+   left_join(lagged_damage_tbl, by = "key_code")
> lagged_variables_tbl     %>%    summary()
   key_code        SportClubDummy    LocalSportClubDummy  LaggedDamage
 Length:2287       Min.   :0.00000   Min.   :0.0000     Min.   :0.0000
 Class :character   1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.:0.1839
 Mode  :character   Median :0.00000   Median :0.0000     Median :0.5143
                 Mean   :0.02011   Mean   :0.1321     Mean   :0.4816
                 3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.:0.7667
                 Max.   :1.00000   Max.   :1.0000     Max.   :1.0000
                                                   NA's   :1
>
>
> ## ------------------------------------------------------------------------
> ## Combine and save
> ## ------------------------------------------------------------------------
> combined_tbl = tmp_combined_tbl %>%    left_join(lagged_variables_tbl)  %>%
+   mutate(
+     zipcode = str_c(str_sub(zipcode, start = 1, end = 3), "-", str_sub(zipcode, start = 4, end = 7)),
+     zipcode = if_else(zipcode == "962-0838", true = "173-0027", false = zipcode)
+   )  %>%
+   select(
+     row_id, row_count:pop, zipcode, -year,
+     all_of(depvar_vec_combined), contains("_1993"),
+     all_of(placebo_depvar_vec), all_of(treatment_vec), LaggedDamage,
+     all_of(fe_term), all_of(spatial_covariates), all_of(distance_termz_base),
+     z_lon, z_lat, longitude, latitude,
+     palace_dist)   %>%
+   drop_na(ratio_damage)
```

```
Joining, by = c("key_code", "SportClubDummy")
>
>
> ## ------------------------------------------------------------------------
> ## Baseline formula objects
> ## ------------------------------------------------------------------------
> mdl_lst = list(
+   ## Polynomial: felm()-compatible specification
+   polynomial_base_felm = chr2fml_felm(
+     "outcome",
+     idv_list = list(
+       treatment_vec[1],
+       spatial_covariates,
+       distance_term_specification[[1]],
+       spatial_term_specification[1]
+     ),
+     fe_list = fe_term,
+     se_cluster = "row_count"    ## Robust SE
+     # se_cluster = fe_term      ## Cluster SE
+   ),
+   ## Spline: gam()-compatible specification
+   gam_base = chr2fml(
+     "outcome",
+     idv_list = list(
+       treatment_vec[1],
+       spatial_covariates, fe_term,
+       distance_term_specification[[2]],
+       spatial_term_specification[2]
+     )
+   )
+ )
> ## ------------------------------------------------------------------------
>
>
> ## ------------------------------------------------------------------------
> ## Result list object
> ## ------------------------------------------------------------------------
> rslt_lst = list(
+   ## ------------------------------------------------------------------------
+   ## 1a. DV = Sport, LPM
+   Sport_LPM = mdl_lst[[1]]      %>%
+     update(
+       str_c(placebo_depvar_vec[1], " ~ .")    %>%    as.formula()  ## formula obj to feed felm()
```

```
+    ) %>%
+    felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 1b: DV = Sport, probit/logit link
+   Sport_GAM = mdl_lst[[2]]      %>%
+    update(str_c(placebo_depvar_vec[1], " ~ ."))    %>%
+    gam(data = combined_tbl, family = binom_link),
+   ## ------------------------------------------------------------------------
+   ## 2a: DV = Sport (own or neighbors), LPM
+   LocalSport_LPM = mdl_lst[[1]] %>%
+    update(
+      str_c(placebo_depvar_vec[2], " ~ .")    %>%    as.formula()  ## formula obj to feed felm()
+    ) %>%
+    felm(data = combined_tbl, keepCX = TRUE, psdef = FALSE),
+   ## ------------------------------------------------------------------------
+   ## 2b: DV = Sport (own or neighbors), probit/logit link
+   LocalSport_GAM = mdl_lst[[2]] %>%
+    update(str_c(placebo_depvar_vec[2], " ~ ."))    %>%
+    gam(data = combined_tbl, family = binom_link)
+   ## ------------------------------------------------------------------------
+ )
> ## ------------------------------------------------------------------------
> ## Print regression estimates
> ## ------------------------------------------------------------------------
> stargazer(
+   rslt_lst,
+   type = "text",
+   align = TRUE,
+   keep = c(treatment_vec)
+ )

=====================================================================
                                 Dependent variable:
                           ---------------------------------------
                            SportClubDummy   LocalSportClubDummy
                           felm      GAM      felm     GAM
                                 (logistic)          (logistic)
                            (1)      (2)      (3)      (4)
---------------------------------------------------------------------
ratio_damage                 0.006   -0.292    -0.013   -0.047
                           (0.010)  (0.691)  (0.029)  (0.331)

---------------------------------------------------------------------
```

```
Observations                    2,155    2,155    2,155    2,155
R2                             0.033            0.135
Adjusted R2                    -0.003    0.008    0.103    0.185
Log Likelihood                          -245.474          -723.832
UBRE                                    -0.772            -0.328
Residual Std. Error (df = 2078)  0.145            0.328
================================================================================
Note:                                    *p<0.1; **p<0.05; ***p<0.01
> ## ----------------------------------------------------------------------
>
>
> ## EOF
>
```