Code

Opportunists and Organizers: How Urban Regimes and Social Capital build Solar Cities in Japan

Replication Code

Timothy Fraser, Northeastern University Political Science PhD Candidate

January 19, 2021

This document contains all code necessary for reproducing datasets, analyses, and visuals for the paper, “Opportunists and Organizers: How Urban Regimes and Social Capital build Solar Cities in Japan.”

0. Packages

First, let’s load necessary packages.

Hide

```
# Load packages
library(tidyverse)
library(readxl)
library(tidygraph)
library(ggraph)

#library(conflicted)
# Make sure the select and filter functions have no complications
select <- dplyr::select
filter <- dplyr::filter

# Mapping Packages
library(sf)
library(rgdal)
# Network Packages
library(network)
library(intergraph)
library(GGally)

# https://spatialreference.org/ref/esri/asia-north-albers-equal-area-conic/
equalarea <- "+proj=aea +lat_1=15 +lat_2=65 +lat_0=30 +lon_0=95 +x_0=0 +y_0=0 +ellps=WGS
84 +datum=WGS84 +units=m +no_defs"

# Let's also create a folder for holding visualizations.
dir.create("viz")
```

1. Format Japanese Renewable Energy Data

Japan’s Agency for Natural Resources and Energy compiled records on every Feed-in Tariff subsidized solar project in Japan. Let’s access these records, stored in “company_network/japanese_solar_plants”, and geocode them.

Hide

```
# Let's write a quick function to extract columns from
# each prefecture-month dataset on solar adoption.
get_data = function(file){
  # Print name of prefecture from which data was extracted
    pref <- paste(
      file %>%
        str_remove_all(
          pattern = "company_network/japanese_solar_plants/|_201907.xlsx"))
    print(pref)

  file %>%
    # Grab the data on FiT certified plants (認定設備),
    # as these are the most reliable and time-recorded data
    # on solar in Japan
    readxl::read_excel(skip = 2, sheet = 1) %>% slice(-1) %>%
    select(id = `設備ID`,
           plant = `発電事業者名`,
           rep_name = `代表者名`,
           office_address = `事業者の住所`,
           office_phone = `事業者の電話番号`,
           type = `発電設備区分`,
           size_kw = `発電出力（kW）`,
           plant_address = `発電設備の所在地`,
           date_certified = `新規認定日`,
           total_output_kw = `太陽電池の合計出力（kW）`,
           units = 9,
           disposal_costs_reserved = `廃棄費用の積立状況`) %>%
    # Make sure each column keeps the following format
    mutate_at(vars(id, plant, rep_name,
                   office_address, office_phone,
                   type, plant_address, date_certified,
                   disposal_costs_reserved),
              funs(as.character(.))) %>%
    mutate_at(vars(size_kw, total_output_kw, units),
              funs(as.numeric(.))) %>%
    mutate(pref = pref) %>%
    return()

}

# Run this function on all files in the following folder, then bind them together
# and filter to just plants that are actually online. This is a useful check to eliminat
e projects which are proposed but never really come to fruition.
dat <- dir("company_network/japanese_solar_plants", full.names = TRUE) %>%
  lapply(get_data) %>%
  bind_rows() %>%
    # Filter out any plants that have not actually begun producing power
  filter(!is.na(total_output_kw))
```

Hide

```
# Now, let's reformat the data, making the office addresses clearer.
dat <- dat %>%
  # Now, let's edit the office addresses so that they use the same formatting
    # First, extract the numeric portion of the address
  mutate(numbers = office_address %>%
           str_extract_all(pattern = "([0-9|一|二|三|四|五|六|七|八|九|十|１|２|３|４|５|６|
７|８|９|０|零|〇]+)(番地|丁目|号|-|ー|).*") %>%
  sapply(function(x){paste(x, collapse = "")}) %>%
    na_if(y = "NA")) %>%
  # Now convert the number portion of the address
  mutate(fixed = numbers %>%
           str_replace_all(pattern = c(
             "番地|番" = "-", "丁目" = "-", "号" = "-", "一" = "-", "ー" = "-",
             "一|１" = "1", "二|２" = "2", "三|３" = "3", "四|４" = "4", "五|５" = "5",
             "六|６" = "6", "七|７" = "7", "八|８" = "8", "九|９" = "9", "十|１０" = "10",
             "零|〇|０" = "0")) %>%
           # Now, if a "号" not followed by anything else got replaced,
           # it will leave behind a "-".
           # We now remove the "-" from any strings ending in "-"
           str_remove(pattern = "-$")) %>%
  # Now, find the string we originally extracted from office_address,
  # which we saved in the vector numbers,
  # and replace it with the edited string we saved in fixed
  mutate(office_address = office_address %>%
           str_replace(pattern = numbers, replacement = fixed) %>%
           # Remove any open blank spaces, either large "  " or small " ".
           str_remove_all(pattern = "  | ")) %>%
  select(-numbers, -fixed)


# Save edited results
dat %>%
  # save
  saveRDS("company_network/jp_solar_plants.rds")
# This totals 400983 installations
remove(dat)
```

Finally, let's geocode each of these plants (so to speak), by identifying the prefecture and city of every plant and assigning that plant the unique five-digit municipality code for the municipality which hosts it.

Hide

```
library(tidyverse)
library(zipangu)
# Next, adjust the address to include the prefecture name
dat <- read_rds("company_network/jp_solar_plants.rds") %>%
  mutate(pref = pref %>% str_remove("[0-9]{2}[.]")) %>%
  # Now check if the prefecture name is present in the plant address. If not, fill it i
n.
  mutate(plant_address = if_else(str_detect(string = plant_address, pattern = pref),
           true = plant_address,
           false = paste(pref, plant_address, sep = "")))

muni <- read_csv("jp_solar/muni_code.csv")

# Append the muni-code to the row
geocode = function(i){
  # Print completion rate
  print(paste(i, " complete"))

  dat %>%
    # filter just to the prefecture of interest
    filter(pref == muni$pref[i]) %>%
    # Now capture the municipality of any plant address that contains
    # that municipality code's municipality name and prefecture name
    mutate(geo = if_else(
      str_detect(plant_address, pattern = muni$pref[i]) &
        str_detect(plant_address, pattern = muni$muni[i]),
      true = muni$muni_code[i], false = NA_character_)) %>%
    # Now filter just to those plants that received a code
    filter(!is.na(geo)) %>%
    # return for bundling with others and saving
    return()
}

system.time(
  1:nrow(muni) %>%
    lapply(geocode) %>%
    bind_rows() %>%
    saveRDS("company_network/jp_plants_geocoded.rds")
  # Remaining 94,451 out of 100,079
)

remove(dat, muni, geocode)
```

2. Matching Companies

2.1 Simplifying Names

Next, I consolidate the project names for each solar farm into simpler forms where able, removing phrases like “Ltd.” or “Corp” and identifying the use of company company names in the field, like Kyocera, Hitachi, or Softbank.

Hide

```
#install.packages("fuzzyjoin")
library(fuzzyjoin)
library(tidyverse)
library(zipangu)
select <- dplyr::select
```

Hide

```
super_recode = function(data){
  require(zipangu)
  require(stringdist)
  require(stringr)
  require(tidyverse)
  # Extra patterns to remove
  removalpattern <- c(" ","  ", "ー", "-","・",
                      "株式会社","有限会社","合同会社","(株)",
                      "株式会","有限責任", "事業組合", "(有)",
                      "一般社団法人")

  # Edit plant names to try and identify companies
  data %>%
    str_conv(encoding = "UTF-8") %>%
    str_conv_normalize(to = "nfkc") %>%
    tolower() %>%
    # Remove spaces and the word company and equivalents
    str_remove_all(pattern = " ") %>%
    str_remove_all(pattern = paste(
      removalpattern, collapse = "|")) %>%
    # Find Japan Names
    str_replace_all(
       pattern = "日本|ニッポン|にっぽん|にほん|ニホン|nihon", replacement = "日本") %>%
    # Improve names
    str_replace_all(
      c(
        ".*京セラ.*" = "kyocera", ".*kyocera.*" = "kyocera",
        ".*ntt.*" = "ntt",
        ".*サンワ.*" = "sanwa", ".*sanwa.*" = "sanwa",
        ".*オリックス.*" = "orix", ".*orix.*" = "orix",
        ".*sget.*" = "sget", ".*toprunner.*" = "toprunner",
        ".*qvou.*" = "qvou", ".*freshup.*" = "freshup",
        ".*ecostyle.*" = "ecostyle", ".*lohas.*" = "lohas",
        ".*t&m.*" = "t&m", ".*t&j.*" = "t&j",
        ".*a&c.*" = "a&c", ".*ティーダ.*" = "tida",
        ".*ワタミ.*" = "watami energy",
        ".*jfe.*" = "jfe",
        ".*ヴェオリア.*" = "veolia",
        ".*大阪瓦斯.*" = "osaka gas",
        ".*日本製紙.*" = "nippon paper industries",
        ".*上組.*" = "kamigumi",
        ".*dmm*" = "dmm.com",
        ".*大和.*" = "daiwa",
        ".*三菱.*" = "mitsubishi",
        ".*中部電力.*" = "chubu electric / ctech",
        ".*ctech*" = "chubu electric / ctech",
        ".*東北電力.*" = "tohoku electric",
        ".*関西電力.*" = "kansai electric",
        ".*九州電力.*" = "kyushu electric",
        ".*四国電力.*" = "shikoku electric",
        ".*四電工.*" = "shikoku electric",
        ".*四電.*" = "shikoku electric",
```

```
".*九電工.*" = "kyushu electric",
".*東電工.*" = "tokyo electric",
".*東京発電.*" = "tokyo electric",
".*東電.*" = "tokyo electric",
".*東京電力.*" = "tokyo electric",
".*北海道電力.*" = "hokkaido electric",
".*中国電力.*" = "chugoku electric",
".*関電工.*" = "kansai electric",
".*関電.*" = "kansai electric",
".*シービーエス.*" = "cbs",
".*daiwa.*" = "daiwa",
".*いちごeco.*" = "ichigoeco",
".*csd.*" = "csd", ".*itd.*" = "itd",
".*ツジデン.*" = "tsujiden",
".*arca.*" = "arca",
".*greensystem.*" = "greensystem", ".*jai.*" = "jai",
".*jre.*"  = "jre", ".*jps.*" = "jps",
".*izumi.*" = "izumi", ".*sb.*" = "sb",
".*we＆kenergy.*" = "we&kenergy",
".*rsalpetroleum*" = "universalpetroleum",
".*soleil.*" = "soleil", ".*acty.*" = "acty",
".*a-style.*" = "a-style", ".*未来創電|mirai創電.*" = "sojitz",
".*ゼンショー.*" = "zensho", ".*lvc.*" = "lvc",
".*advance.*" = "advance", ".*ntt.*" = "ntt",
".*ロジカルバリュー.*" = "logicalvalue",
".*orsolar.*" = "orsolar", ".*レオパレス.*" = "レオパレス",
".*アーステック.*" = "earth tech",
".*滋賀総合建設.*" = "shiga construction",
".*滋賀建.*" = "shiga construction",
".*インテグリティパートナーズ.*" = "ip",
".*アドニス.*" = "adonis",
".*富士ジーワイ商事.*" = "fuji gy trading",
".*共栄.*" = "kyoyei",
".*アツギ.*" = "atsugi",
".*全農.*" = "zenno energy",
".*日昇つくば.*" = "nissho tsukuba",
".*東鉄工業.*" = "totestu kogyo",
".*シーテック.*" = "chubu electric / CTECH",
".*西部瓦斯.*" = "saibu gas",
".*エネサンス.*" = "enessance holdings",
".*日鉄.*" = "nippon steel",
".*東光電気.*" = "toko electrical works",
".*岩田地崎.*" = "iwata chizaki group",
".*タクマ.*" = "takuma",
".*新日本電工.*" = "shin-nippon electric works",
".*三井物産フォーサイト.*" = "mitsui real estate",
".*サイサン.*" = "saisan",
".*楽天.*" = "rakuten", ".*らくてん.*" = "rakuten",
".*ラクテン.*" = "rakuten",
".*遠州トラック.*" = "enshu truck",
".*東京不動産管理.*" = "tokyo real estate",
".*テクノ菱和.*" = "techno ryowa",
".*タケエイ.*" = "takeei",
".*jr九州.*" = "jr kyushu",
```

```
        ".*ダイオーエンジニアリング.*" = "daio engineering",
        ".*ユーエスエス.*" = "uss",
        ".*西日本技術開発.*" = "west japan engineering",
        ".*東洋産業.*" = "toyo industries",
        ".*ユニティー.*" = "unity",
        ".*岐建.*" = "giken",
        ".*三谷商.*" = "mitani",
        ".*日本重化学工業.*" = "japan metals & chemicals",
        ".*三交不動産.*" = "sanko real estate",
        ".*朝日興業.*" = "asahi kogyo",
        ".*西部石油.*" = "seibu oil",
        ".*スバル興業.*" = "subaru kogyo",
        ".*国際石油開発帝石.*" = "inpex",
        ".*グリムス.*" = "gremz",
        ".*坪井工業.*" = "tsuboi industries",
        ".*ヤマト電機.*" = "yamato denki",
        ".*アドバンテック.*"= "advantech",
        ".*スマートテック.*" = "smart tech",
        ".*日本空調.*" = "japan air conditioning",
        ".*メディオテック.*" = "mediotec",
        ".*丸住製紙.*" = "marusumi paper",
        ".*三谷セキサン.*" = "mitani sekisan",
        ".*大分石油.*" ="oita oil",
        ".*みすず.*" = "misuzu corporation",
        ".*東京ガス.*" = "tokyo gas",
        ".*日本国土開発.*" = "jdc",
        "ソーラー" = "ソラー",
        "ソラー" = "ソラー",
        ".*未来創造.*" = "mirai sozo"))  %>%
    return()
}
```

Finally, let's save these plants as the "population" dataset, zooming into just solar farms with an installed capacity of at least 10 kilowatts (the minimum size for being designated utility-scale solar).

Hide

```
read_rds("company_network/jp_plants_geocoded.rds") %>%
  # Filter out any plants that have not actually begun producing power
  filter(!is.na(total_output_kw)) %>%
  # Bad news 154,791 of the total 400,983 plants do NOT have a company address
  # This makes it very hard to tell if this plant actually exists.
  # Filter to plants that actually have a phone, office address, plant address,
  # and representative name
  # These plants are most likely to have actually been built.
  filter(
    #!is.na(rep_name),
    #!is.na(office_address),
         !is.na(plant_address),
    #     !is.na(office_phone),
         !is.na(type),
         !is.na(size_kw),
         !is.na(date_certified)) %>%
  # Filter to just solar farms over 10 kilowatts in installed capacity
  filter(type == "太陽光", size_kw >= 10) %>%
  # Now gather the main details
  select(id, plant, rep_name, office_address, office_phone, type, plant_address,
         size_kw, date_certified, pref, geo) %>%
  mutate(plant = plant %>% super_recode()) %>%
  saveRDS("company_network/jp_plants_population.rds")
```

2.2 Join plants and companies

Next, let's import the list of solar company names I obtained by webscraping the corporate intelligence website BaseConnect.

Hide

```
# Take list of webscraped company names;
# Create simplified versions of their names,
# Using the same rules as I applied to simplifying plant names
companies <- read_rds("webscraping/company_traits.rds") %>%
  pivot_wider(id_cols = id, names_from = label, values_from = result) %>%
  # Grab those rows where we have estimates on how many employees there are
  # these are the most likely to be real companies, not shells
  select(company_id = id, company_name, employees = `従業員数`) %>%
 # filter(!is.na(employees)) %>%
  select(-employees) %>%
  mutate(company_name = company_name %>% super_recode())
```

Now, let's write and implement a quick function, broken into chunks of 2000 plants, since there are a lot of plants (400,000+), which will match the names of any of our webscraped companies to our population of solar farms.This produces a sample of ~25,000 solar farms, approximately 5% of the original dataset where we know which company they represent.

Hide

```
get_match = function(data){

  print(data$segment[1])

  data %>%
      fuzzy_join(y = companies, by = c("plant" = "company_name"),
             match_fun = str_detect, mode = "inner") %>%
    return()
}

# For each segment of 2000 solar farms,
read_rds("company_network/jp_plants_population.rds") %>%
  # grab just the plant id and name
  select(id, plant) %>%
  # create an id for each round
  mutate(segment = round(1:n() / 2000)) %>%
  split(.$segment) %>%
  # For each segment,
  # fuzzy join in the appropriate company names
  map_dfr(. %>% get_match(), .id = "segment") %>%
  saveRDS("company_network/jp_plant_names.rds")

# This produces a sample of ~25,000 solar farms, approximately 5% of the original datase
t
# where we know which company they represent
#read_rds("company_network/jp_plant_names.rds")
```

3. Analyze Sampling

Next, we want to investigate traits of our sample to see if it is remotely representative of the population. (Good news; it is surprisingly representative), with some expected differences, since we essentially took a convenience sample.

3.1 Gather Sample

First, let's import our cities, companies, and population of solar farms, and derive from it the sample of solar farms.

Hide

```
# Import towns
muni <- read_csv("company_network/census_jp/muni_code.csv") %>%
  left_join(by = "muni_code",
            y = read_rds("company_network/census_jp/jp_dataset.rds") %>%
              ungroup() %>%
              select(pop, muni_code) %>%
              distinct()) %>%
  # filter to our study portion
  filter(pop >= 30000)

# Write a quick function that we will apply to the population and sample
# in order to assess their traits
get_tally = function(data){
  data %>%
    # Consolidate the plants into an edgelist
    group_by(muni_code = geo, pref, year = str_sub(date_certified, 1,4)) %>%
    # Preserve total number of plants and total size of plants in kw per town-company pa
ir
    summarize(plants = n(),
              size_kw = sum(size_kw, na.rm = TRUE)) %>%
    ungroup() %>%
    return()
}


# Generate population data.frame
pop <- read_rds("company_network/jp_plants_population.rds") %>%
  # Filter just to solar farms in cities with populations over 30,000
  filter(geo %in% muni$muni_code)

# Get companies with sufficient data
companies <- read_rds("webscraping/company_traits.rds") %>%
  filter(label == "従業員数" & !is.na(result)) %>%
  select(id) %>% unlist()

# Generate sample data frame
sample <- read_rds("company_network/jp_plants_population.rds") %>%
  # Filter just to solar farms in cities with populations over 30,000
  filter(geo %in% muni$muni_code)  %>%
  # Join in company names,
  # and keep only those plants with company names
  inner_join(by = "id",
            y = read_rds("company_network/jp_plant_names.rds") %>%
              select(id, company_id))



# Generate sample size of samples against which to compare
n = read_rds("company_network/jp_plants_population.rds") %>%
  # Filter just to solar farms in cities with populations over 30,000
  filter(geo %in% muni$muni_code)  %>%
  # Join in company names,
  # and keep only those plants with company names
```

```
  inner_join(by = "id",
             y = read_rds("company_network/jp_plant_names.rds") %>%
               select(id, company_id)) %>%
  # Get sample size
  nrow()


# Export the companies in this sample
read_rds("webscraping/company_traits.rds") %>%
  pivot_wider(
    id_cols = id,
    names_from = label,
    values_from = result) %>%
  # Filter to just those companies with sufficient traits
  filter(!is.na(`従業員数`)) %>%
  # Filter to those companies in our sample that we can look at
  filter(id %in% sample$company_id) %>%
  write_csv("company_network/jp_companies.csv")



# Gather sample, focusing JUST on plants from companies for which we have sufficient dat
a
read_rds("company_network/jp_plants_population.rds") %>%
  # Filter just to solar farms in cities with populations over 30,000
  filter(geo %in% muni$muni_code)  %>%
  # Join in company names,
  # and keep only those plants with company names
  inner_join(by = "id",
             y = read_rds("company_network/jp_plant_names.rds") %>%
               select(id, company_id)) %>%
  # Zoom into just plants from our 565 or so companies
  filter(company_id %in% unique(read_csv("company_network/jp_companies.csv")$id)) %>%
  saveRDS("company_network/jp_plants_sample.rds")

# Import
sample <- read_rds("company_network/jp_plants_sample.rds")
```

3.2 T-Tests

Next, let's use a series of one-sample t-tests, where we compare the sample mean to the population mean, to verify what traits about our sample are within the sample ballpark as our population.

Hide

```
# How similar/different is the sample of plants from these specific companies
# compared to the population of all plants?

# When we just straight-forwardly compare the mean size of installations,
# We find that the average size of plants in our sample is quite a bit larger, averaging
at ~285 compared to 140; that different is statistically significant.
result_overall <- sample %>%
  filter(company_id %in% companies) %>%
  select(size_kw) %>%
  t.test(
    mu = mean(pop$size_kw,na.rm = TRUE),
    alternative = "two.sided") %>%
  broom::tidy() %>%
  bind_cols(data.frame(pop_mean =mean(pop$size_kw,na.rm = TRUE)))

# However, this shouldn't invalidate the sample right off the bat.

# Let's assess how much specific sizes of plants differ in the sample
# compared to the population, on average.

# On average, looking at plants 10-49 kilowatts in size,
# the sample differs only a tiny bit - the average size of small plants is 42 kW in the
 population but 45 kW in the sample (p < 0.001).
result_10_49 <- sample %>%
  filter(company_id %in% companies) %>%
  select(size_kw) %>%
  filter(size_kw >= 10 & size_kw < 50) %>%
  t.test(
    mu = pop %>%
      select(size_kw) %>%
      filter(size_kw >= 10 & size_kw < 50) %>%
      summarize(mean = mean(size_kw, na.rm = TRUE)) %>% unlist(),
    alternative = "two.sided") %>%
  broom::tidy() %>%
  bind_cols(pop %>%
      select(size_kw) %>%
      filter(size_kw >= 10 & size_kw < 50) %>%
      summarize(pop_mean = mean(size_kw, na.rm = TRUE)))

# On average, looking at plants 50-499 kW in size,
# the sample differs slightly - the average size of medium plants is 247 kW in the popul
ation, but 270 kW in the sample (p < 0.001).
result_50_499 <- sample %>%
  filter(company_id %in% companies) %>%
  select(size_kw) %>%
  filter(size_kw >= 50 & size_kw < 499) %>%
  t.test(
    mu = pop %>%
      select(size_kw) %>%
      filter(size_kw >= 50 & size_kw < 499) %>%
      summarize(mean = mean(size_kw, na.rm = TRUE)) %>% unlist(),
    alternative = "two.sided") %>%
```

```
  broom::tidy() %>%
  bind_cols(pop %>%
      select(size_kw) %>%
      filter(size_kw >= 50 & size_kw < 499) %>%
      summarize(pop_mean = mean(size_kw, na.rm = TRUE)))

# On average, looking at plants 500-1999 kW in size,
# the sample differs slightly - the average size of large plants is 1109 kW in the popul
ation compared to 1187 kW in the sample (p < 0.001).
result_500_1999 <- sample %>%
  filter(company_id %in% companies) %>%
  select(size_kw) %>%
  filter(size_kw >= 500 & size_kw < 1999) %>%
  t.test(
    mu = pop %>%
      select(size_kw) %>%
      filter(size_kw >= 500 & size_kw < 1999) %>%
      summarize(mean = mean(size_kw, na.rm = TRUE)) %>% unlist(),
    alternative = "two.sided") %>%
  broom::tidy() %>%
  bind_cols(pop %>%
      select(size_kw) %>%
      filter(size_kw >= 500 & size_kw < 1999) %>%
      summarize(pop_mean = mean(size_kw, na.rm = TRUE)))

# Finally, on average, looking at plants 2 megawatts in size or greater,
# the sample differs greatly, but in the opposite direction. The average size of mega-so
lar plants in 18180 kW in the population, but just 13709 kW in the sample (p < 0.001).
result_2000 <- sample %>%
  filter(company_id %in% companies) %>%
  select(size_kw) %>%
  filter(size_kw >= 2000) %>%
  t.test(
    mu = pop %>%
      select(size_kw) %>%
      filter(size_kw >= 2000) %>%
      summarize(mean = mean(size_kw, na.rm = TRUE)) %>% unlist(),
    alternative = "two.sided") %>%
  broom::tidy() %>%
  bind_cols(pop %>%
      select(size_kw) %>%
      filter(size_kw >= 2000) %>%
      summarize(pop_mean = mean(size_kw, na.rm = TRUE)))

# This shows that the sample of plants obtained from these companies involve installatio
ns that are slightly larger in size in small, medium, and large size categories on avera
ge, but slightly smaller in size in the mega-solar category.

bind_rows(
  result_overall,
  result_10_49,
  result_50_499,
  result_500_1999,
  result_2000, .id = "id") %>%
```

```
  pivot_longer(
    cols = c(estimate, pop_mean),
    names_to = "type",
    values_to = "value") %>%
  mutate(id = id %>% recode_factor(
    "1" = "Overall",
    "2" = "Solar (10-49 kW)",
    "3" = "Solar (50-499 kW)",
    "4" = "Solar (500-1999 kW)",
    "5" = "Solar (2000 kW)")) %>%
  mutate(type = type %>% recode_factor("estimate" = "Sample", "pop_mean" = "Population"
)) %>%
  ggplot(mapping = aes(x = id, y = value / 1000, ymin = conf.low / 1000, ymax = conf.hig
h / 1000, color = type)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  geom_linerange(size = 1.25, color = "firebrick", alpha = 0.75) +
  geom_point(size = 3, alpha = 0.75) +
  scale_y_log10() +
  scale_color_manual(values = c("firebrick", "steelblue")) +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        plot.caption = element_text(hjust = 0.5),
        legend.position = "bottom") +
  labs(y = "Average Megawatts\nof Solar (log)",
       x = "Type of Solar", color = "Group",
       caption = "Bands depict 95% confidence intervals.This shows that the sample of pl
ants obtained\nfrom these companies involve installations that are slightly larger in si
ze\nin small, medium, and large size categories on average,\nbut slightly smaller in siz
e in the mega-solar category.") +
  ggsave("viz/fig_A1_one_sample_ttests.png", width = 8, height = 4, dpi = 500)
```

4. Build Datasets

Next, let's build our datasets. This requires processing our company traits and city traits into formats that we can model.

Hide

```
rm(list = ls())
```

4.1 Company Traits

First, let's process our company traits, by turning various dates written in kanji or in the Showa/Heisei/Reiwa format into the western calendar.

Hide

```
# Let's write a quick function to convert Japanese dates to the western calendar.
fix_date = function(original_date){

  data.frame(original_date = original_date) %>%
  # Convert start-date to English date format
  mutate(start_year = original_date %>% str_extract("[0-9]{4}年") %>% parse_number(),
         start_month = original_date %>% str_remove("[0-9]{4}年") %>% str_remove("月"),
         start_day = original_date %>% str_remove("[0-9]{4}年[0-9]+月") %>% str_remove(
"日"),

         # If the start month has no characters in it, set that to NA
         start_month = if_else(start_month == "", NA_real_, as.numeric(start_month)),
                  # If the start year is present but the start month is empty, set it to
January
         start_month = if_else(is.na(start_month), 1, start_month),
         # If the start month has no characters in it, set that to NA
         start_day = if_else(start_day == "", NA_real_, as.numeric(start_day)),
         start_day = if_else(is.na(start_day), 1, start_day),
         original_date = lubridate::make_date(year = start_year, month = start_month, da
y = start_day)) %>%
  select(original_date) %>%
    .[,1] %>%
    return()
}

# Convert value
# Extract key financial patterns

# Let's also extract monetary values from Japanese formats
fix_money = function(value){

data.frame(value = value) %>%
  mutate(
    cho = str_extract(value, pattern = "[0-9]+兆円|[0-9]+兆"),
    oku = str_extract(value, pattern = "[0-9]+億円|[0-9]+億"),
         man = str_extract(value, pattern = "[0-9]+万円|[0-9]+万"),
         yen = str_extract(value, pattern = "[0-9]+円")) %>%
  mutate_at(vars(cho,oku,man,yen), funs(. %>% str_remove_all("兆|億|万|円") %>% parse_num
ber())) %>%
  mutate(
    # 1 trillion
    cho = cho * 1000000000000,
    # 100 million
    oku = oku * 100000000,
    # 10 thousand
         man = man * 10000,
    # 1 yen
         yen = yen) %>%
  rowwise() %>%
  mutate(value = sum(cho, oku, man, yen, na.rm = TRUE)) %>%
  # Finally, go back and fill these in as NA, not ZERO, if they were missing all kinds o
f money
```

```
  mutate(value = if_else(is.na(cho) & is.na(oku) & is.na(man) & is.na(yen), NA_real_, va
lue)) %>%
  select(value) %>%
  unlist() %>%
  return()
}

# Finally, let's import our company dataset,
# implementing both of these techniques for adapting variables
companies <- read_csv("company_network/jp_companies.csv") %>%
  select(company_id = id, company_name,
         attribute,
         start_date = `設立年月`,
         value = `資本金`,
         sales = `売上高`,
         growth_rate_2018 = `売上高増加率`,
         employees = `従業員数`,
         employee_growth_rate = `従業員増加率`,
         offices = `事業所数`,
         facilities = `工場数`,
         market = `上場市場`,
         closing_month = `決算月`,
         ceo_name = `名前`,
         ceo_birthdate = `生年月日`,
         ceo_uni = `出身大学`)  %>%
  # Fix up these
  mutate(start_date = fix_date(start_date)) %>%
  mutate(value = fix_money(value)) %>%
  mutate(sales = sales %>% str_remove("[(][0-9]{4}年度[)]") %>% fix_money()) %>%
  mutate(growth_rate_2018 = growth_rate_2018 %>% str_remove("[(][0-9]{4}年度[)]") %>% pa
rse_number()) %>%
  mutate(employees = employees %>% str_extract("[0-9]+人") %>% parse_number()) %>%
  mutate(employee_growth_rate = employee_growth_rate %>% str_remove("[(][0-9]{4}年度[)]"
) %>% parse_number()) %>%
  mutate(offices = offices %>% recode_factor(
    "0~3箇所" = "0-3",
    "4~10箇所" = "4-10",
    "11箇所以上" = "11 or more")) %>%
  mutate(facilities = facilities %>% recode_factor(
    "0~5箇所" = "0-5",
    "6~20箇所" = "6-20",
    "21~50箇所" = "21-50",
    "51箇所以上" = "51 or more")) %>%
  mutate(market = market %>% recode_factor(
    "東証１部" = "Tokyo Stock Exchange 1st section",
    "東証２部" = "Tokyo Stock Exchange 2nd section",
    "東証マザーズ" = "Tokyo Stock Exchange MOTHERS",
    "JASDAQ" = "JASDAQ",
    "未上場" = "unlisted")) %>%
  mutate(ceo_birthdate = ceo_birthdate %>% fix_date())

# Let's identify the classificatiosn of each company, assigned by BaseConnect,
# and classify each company into them using a preferential order that highlights involve
ment in key industries of interest.
```

```
attributes <- companies %>%
  select(company_id, attribute) %>%
  separate(col = attribute, sep = ", ", into = paste("col", 1:16, sep = "_")) %>%
  pivot_longer(
    cols = c(col_1:col_16),
    names_to = "column",
    values_to = "entry") %>%
  select(-column) %>%
  filter(!is.na(entry)) %>%
  mutate(entry = str_remove(entry, ",| ")) %>%

  mutate(entry = entry %>% str_replace_all(c(
    ".*再生.*" = "renewables",
    ".*再エネ" = "renewables",
    ".*太陽光.*" = "renewables",
    ".*風力.*" = "renewables",

    ".*電力.*" = "energy",
    ".*ガス.*" = "energy",
    ".*燃料.*" = "energy",
    ".*ガソリン.*" = "energy",
    ".*エネルギー.*" = "energy",

    ".*建.*" = "construction/engineering",
    ".*工.*" = "construction/engineering",
    ".*ゼネコン.*" = "construction/engineering",
    ".*電気設備工事.*" = "construction/engineering",
    ".*工事.*" = "construction/engineering",
    ".*エンジニアリング.*" = "construction/engineering",
    ".*リフォーム.*" = "construction/engineering",

    ".*不動産.*" = "real estate",
    ".*アパート.*" = "real estate",
    ".*マンション.*" = "real estate",
    ".*住宅.*" = "real estate",
    ".*テナント.*" = "real estate",
    ".*レンタル.*" = "real estate",
    ".*リース.*" = "real estate",
    ".*ホテル.*" = "real estate",
    ".*旅館.*" = "real estate",

    ".*廃棄.*" = "waste",
    ".*リサイクル.*" = "waste",

    ".*送.*" = "transport",
    ".空運.*" = "transport",
    ".*海運.*" = "transport",

    ".*倉庫.*" = "storage",

    ".*電子." = "manufacturing/electronics",
    ".*電池.*" = "manufacturing/electronics",

    ".*樹脂.*" = "manufacturing/electronics",
```

```
    ".*プラスチック.*" = "manufacturing/electronics",
    ".*製造.*" = "manufacturing/electronics",
    ".*機械.*" = "manufacturing/electronics",
    ".*鉄鋼.*" = "manufacturing/electronics",
    ".*家電.*" = "manufacturing/electronics",
    # there are not enough telecoms to have a distinct communications level
        ".*テレコム.*" = "manufacturing/electronics",
    ".*電信.*" = "manufacturing/electronics",
    ".*電話.*" = "manufacturing/electronics",
    ".*テレビ.*" = "manufacturing/electronics",
    ".*通信.*" = "manufacturing/electronics",

    ".*自動車.*" = "auto",
    ".*車.*" = "auto",
    ".*乗り物.*" = "auto",


    ".*売.*" = "commerce",
    ".*売買.*" = "commerce",
    ".*販売.*" = "commerce",
    ".*eコマース.*" = "commerce",



    ".*農.*" = "food/agriculture",
    ".*食.*" = "food/agriculture",
    ".*カフェ.*" = "food/agriculture",
    ".*喫茶店.*" = "food/agriculture",
    ".*肉.*" = "food/agriculture",
    ".*酒.*" = "food/agriculture",
    ".*水産.*" = "food/agriculture",

    ".*貸金.*" = "finance"))) %>%
  mutate(entry = if_else(!entry %in% c("renewables", "energy", "construction/engineerin
g",
                                       "real estate", "waste", "transport",
                                       "manufacturing/electronics", "auto", "commerce",
                                       "communications", "food/agriculture", "finance"),
                         true = "other", false = entry)) %>%
  group_by(company_id) %>%
  summarize(entry = paste(entry, collapse = ";")) %>%
  # finally, reclassify in the following order:
  ungroup() %>%
  mutate(entry = if_else(str_detect(entry, "construction/engineering"),
                         "construction/engineering", entry),
         entry = if_else(str_detect(entry, "manufacturing/electronics"),
                         "manufacturing/electronics", entry),
         entry = if_else(str_detect(entry, "auto"), "auto", entry),
         entry = if_else(str_detect(entry, "real estate"), "real estate", entry),
         #entry = if_else(str_detect(entry, "communications"), "communications", entry),
         entry = if_else(str_detect(entry, "transport"), "transport", entry),
         entry = if_else(str_detect(entry, "waste"), "waste", entry),
         entry = if_else(str_detect(entry, "food/agriculture"), "food/agriculture", entr
y),
```

```
         entry = if_else(str_detect(entry, "finance"), "finance", entry),
         # Finally, if none of the others apply,
         # then they truly originally are energy companies
         entry = if_else(str_detect(entry, "energy"), "energy", entry),
         entry = if_else(str_detect(entry, "renewables"), "renewables", entry))

# Finally, let's save this as the cleaned company datase.
companies %>%
  select(-attribute) %>%
  left_join(by = "company_id", y = attributes %>% rename(company_type = entry)) %>%
  # Remove unusable traits, or those with high levels of missing data
  select(-closing_month, -sales,  -growth_rate_2018,
         -employee_growth_rate, -facilities,
         -ceo_name, -ceo_birthdate, -ceo_uni) %>%
  mutate(company_type = company_type %>% factor() %>% relevel(ref = "renewables")) %>%
  # Consolidate all stock exchanges into one; (because we have very small categories for
each)
  mutate(market = if_else(market != "unlisted",
                          true = "listed", false = "unlisted",
                          missing = NA_character_) %>%
           factor() %>% relevel(ref = "unlisted")) %>%
  saveRDS("company_network/jp_companies_cleaned.rds")

remove(attributes)
```

4.1 City Traits

Next, let’s build a city-level dataset, drawing from lots of census data and external datasets.

A. Census Data

First, let’s combine data obtained from E-Stat, Japan’s census data portal, into a more workable format.

Hide

```
# Take our two data.frames full of lots of data
bind_rows(
  # Import the first, making its municipality code 5 digits long
  read_csv("company_network/census_jp/stats_a.csv") %>%
    mutate(`AREA Code` = `AREA Code` %>% str_pad(width = 5, side = "left", pad = "0")),
  # Import the second, making its municipality code 5 digits long
  read_csv("company_network/census_jp/stats_b.csv") %>%
    mutate(`AREA Code` = `AREA Code` %>% str_pad(width = 5, side = "left", pad = "0")))
 %>%
  # Now select and rename just the following variables
  select(
    muni_code = `AREA Code`,
    variable = ITEMS,
    year = YEAR,
    value = Value) %>%
  # Remove commas and make the value field into a numeric variable
  mutate(value = value %>% str_remove_all("[,]") %>% as.numeric) %>%
  # Take the variable field and recode variable names into simple, R-friendly names
  mutate(variable = variable %>% recode(
    "A5101_Number of in-migrants from other municipalities[person]" = "inmigrants",
    "A5102_Number of out-migrants to other municipalities[person]" = "outmigrants",
    "B1103_Inhabitable area[ha]" = "area_inhabitable",
    "C120110_Taxable income[thousand yen]" = "income_thous",
    "D1202_Number of general personnel (municipalities)[person]" = "employees_muni",
    "C3401_Value of manufactured goods shipments, etc.[million yen]" = "value_manuf_mil
l",
    "D2201_Financial strength index (municipalities)[-]" = "fin_str_index",
    "D2202_Real term balance of revenue to expenditure (municipalities)[%]" = "ratio_rev
_exp",
    "D3203031_Social welfare expenditure (municipalities)[thousand yen]" = "exp_social_w
elfare",
    "D320304_Health expenditure (municipalities)[thousand yen]" = "exp_health",
    "D320308_Public-works expenditure (municipalities)[thousand yen]" = "exp_public_work
s",
    "D320309_(Expenditure) Fire service (local public finance, municipalities)[thousand
 yen]" = "exp_fire",
    "D320310_Education expenditure (municipalities)[thousand yen]" = "exp_education",
    "A1101_Total population (Both sexes)[person]" = "pop",
    "A110102_Total population (Female)[person]" = "pop_women",
    "A1303_Total population (65 and over)[person]" = "pop_age_65_plus",
    "C3501_Annual sales of commercial goods (wholesale and retail trade)[million yen]" =
"value_commerce_mill",
    "F1101_(Population census) Population in labour force[person]" = "pop_labor_force",
    "F1107_(Population census) Unemployed persons[person]" = "pop_unemployed",
    "C2210_Number of persons engaged (Primary industry) (Economic Census for Business Fr
ame)[person]" = "employees_primary",
    "C2211_Number of persons engaged (Secondary industry) (Economic Census for Business
 Frame)[person]" = "employees_secondary",
    "C2212_Number of persons engaged (Tertiary industry) (Economic Census for Business F
rame)[person]" = "employees_tertiary",
    "E9106_Population by last school completed (University and graduate school)[person]"
= "pop_college",
```

```
    "C5401_Average price for land transaction (Residential quarter)[yen/m2]" = "land_pri
ce_res",
    "C3101_Gross agricultural product[million yen]" = "value_agr_mill")) %>%
  # export to file
  saveRDS("company_network/census_jp/stats_long.rds")

# now add in additional spending measures
read_rds("company_network/census_jp/stats_long.rds") %>%
  bind_rows(
    read_csv("company_network/census_jp/census_spending.csv", skip = 8)  %>%
      select(year = YEAR,
             muni_code = `AREA Code`,
             exp_agr_fish_for  = "D320306_Agriculture, forestry and fishery expenditure
 (municipalities)[thousand yen]",
             exp_commerce_manuf = "D320307_Commerce and manufacturing expenditure (munic
ipalities)[thousand yen]") %>%
      pivot_longer(
        cols = -c(year, muni_code),
        names_to = "variable",
        values_to = "value")) %>%
  saveRDS("company_network/census_jp/stats_long.rds")
```

Next, let's identify which years of data were available.

Hide

```
# Now great a grid of all 1741 towns and the years 2000-2017
expand_grid(read_csv("company_network/census_jp/muni_code.csv"), year = 2000:2017) %>%
  # Let's join in
  left_join(by = c("muni_code", "year"),
            y = read_rds("company_network/census_jp/stats_long.rds") %>%
              pivot_wider(id_cols = c(muni_code, year),
                          names_from = variable,
                          values_from = value)) %>%
  saveRDS("company_network/census_jp/stats_wide.rds")


# Calculate available variable-years
read_rds("company_network/census_jp/stats_wide.rds") %>%
  pivot_longer(
    cols = -c(muni_code, pref_code, pref, muni, year),
    names_to = "variable",
    values_to = "value") %>%
  group_by(variable, year) %>%
  summarize(missing = sum(if_else(is.na(value), 1, 0)) / n()) %>%
  # Use a 92% threshold - we only work with data where we have at least 92% of data poin
ts per year
  # We expand this to 92% because residential land price is missing 8% of data points
  filter(missing <= 0.08) %>%
  ungroup() %>%
  mutate(variableyear = paste(variable, year, sep = "_")) %>%
  # Export
  saveRDS("company_network/census_jp/available.rds")

read_rds("company_network/census_jp/available.rds")
```

B. Fill in Missing Years

Next, let’s fill in missing years using the most recently available years of data. Let’s also add in a few extra variables, like disaster damage and social capital.

Hide

```
# Fill in years with value variable values
read_rds("company_network/census_jp/stats_wide.rds") %>%
  pivot_longer(
    cols = -c(muni_code, pref_code, pref, muni, year),
    names_to = "variable",
    values_to = "value") %>%
  # Blank out any values from years that are missing over 95% of data points data
  mutate(variableyear = paste(variable, year, sep = "_")) %>%

  mutate(value = if_else(
    variableyear %in% read_rds("company_network/census_jp/available.rds")$variableyear,
    value, NA_real_)) %>%
  # Remove any NaNs or Inf responses
  mutate(value = if_else(!is.nan(value) & !is.infinite(value), value, NA_real_)) %>%
  select(-variableyear) %>%
  # Now pivot wider again so we can fill in values
  pivot_wider(
    id_cols = c(muni_code, pref_code, pref, muni, year),
    names_from = variable,
    values_from = value)  %>%

  # First, merge in Tim Fraser's social capital and social vulnerability indices
  #left_join(y = read_csv("raw_data/census_jp/indices.csv") %>%
  #            select(muni_code, year, social_capital,
  #                   bonding, bridging, linking, vulnerability),
  #          by = c("muni_code", "year")) %>%
  # Next, merge in Daniel Aldrich's Black Wave data on 2011 disaster damage
  left_join(y = readxl::read_excel("company_network/census_jp/fraser_2019_ssq_data.xlsx"
) %>%
              select(muni_code = Code,
                     pref_en = Prefecture_English,
                     muni_en = Municipality_Prefecture_English,
                     death_2011 = death11, destroy_2011 = destroy11, damage_2011 = damag
ed11,
                     tsunami = coast, exclusion_zone = Fukushima_Exclusion_Zone),
            by = "muni_code") %>%

  # Now arrange so that rows look like Tokyo 2017, Tokyo 2016, Tokyo 2015...
  arrange(muni_code, desc(year)) %>%
  group_by(muni_code) %>%
  # Fill in each value with the most recent year's value
  fill(inmigrants:value_agr_mill, .direction = "downup") %>%
  # I'm going to make a controversial choice here,
  # which is to fill in some values missing in the past with FUTURE values.
  # We only do this for control variables,
  # because we expect that these values remain pretty constant over time.

  # Next, with our filled in population values,
  # we will calculate per capita measures for each here
  mutate(
    pop_women = pop_women / pop * 100,
    pop_age_65_plus = pop_age_65_plus / pop * 100,
```

```
    employees_muni = employees_muni / pop * 100,
    employees_primary = employees_primary / pop_labor_force * 100,
    employees_secondary = employees_secondary / pop_labor_force * 100,
    employees_tertiary = employees_tertiary / pop_labor_force * 100,
    pop_college = pop_college / pop * 100,

    total_migration_rate = (inmigrants + outmigrants) / pop * 1000,
    unemployment = pop_unemployed / pop_labor_force * 1000,

    income_per_capita = income_thous / pop,

    value_manuf_mill = value_manuf_mill / pop,
    value_commerce_mill = value_commerce_mill / pop,
    value_agr_mill = value_agr_mill / pop,

    exp_social_welfare = exp_social_welfare / pop,
    exp_education = exp_education / pop,
    exp_health = exp_health / pop,
    exp_public_works = exp_public_works / pop,
    exp_fire = exp_fire / pop,
    exp_commerce_manuf = exp_commerce_manuf / pop,
    exp_agr_fish_for = exp_agr_fish_for / pop) %>%

  saveRDS("company_network/census_jp/stats_wide_filled.rds")
```

C. Fixing coding for missing values

Finally, let's fix coding issues for missing values, making sure that NaN or Inf are correctly classified as NA so that the mean can fix them.

Hide

```
vars <- c("pop", "pop_women",
          "pop_age_65_plus", "area_inhabitable",
          # Socioeconomics
          "income_per_capita", "unemployment",
          "employees_primary", "employees_secondary",
          "employees_tertiary", "pop_college",
          "land_price_res",
          # Governance Capacity
          "employees_muni", "fin_str_index", "ratio_rev_exp",
          # Urban Regime Outputs
          "exp_commerce_manuf", "exp_agr_fish_for",
          "exp_social_welfare", "exp_health", "exp_public_works",
          "exp_fire", "exp_education",
          # Economic Outputs
          "value_manuf_mill", "value_agr_mill", "value_commerce_mill",
          # Community Resources
          "social_capital", "bonding", "bridging", "linking",
          "vulnerability", "total_migration_rate", "inmigrants", "outmigrants",
          # Disaster data
          "fukushima", "tsunami", "death_2011", "destroy_2011", "damage_2011", "exclusio
n_zone", "disaster_damage")


# Now impute missing values with the mean for each variable-year
read_rds("company_network/census_jp/stats_wide_filled.rds") %>%
  # Calculate disaster damage
  mutate_at(vars(exclusion_zone, destroy_2011, damage_2011),
            funs(if_else(is.na(.), 0, as.numeric(.)))) %>%
  mutate(disaster_damage = (destroy_2011 + damage_2011) / pop * 1000) %>%
  # Make a few extra variables
  mutate(fukushima = if_else(year >= 2011, 1, 0)) %>%
  # For each year, if observations are INF or NAN or NA, make them NA
  # Then replace them with the mean value for that variable for that year
  group_by(year) %>%
  mutate_at(vars(any_of(vars)), funs(if_else(is.na(.) | is.nan(.) | is.infinite(.), NA_r
eal_, .))) %>%
  #  mutate_at(vars(vars),
  #            funs(if_else(is.na(.) | is.nan(.) | is.infinite(.), mean(., na.rm = TRU
E), .))) %>%

  # Now arrange so that rows look like Tokyo 2017, Tokyo 2016, Tokyo 2015...
  arrange(muni_code, desc(year)) %>%
  group_by(muni_code) %>%
  # create a few extra variables

  # export to file
  saveRDS("company_network/census_jp/stats_wide_filled_imputed.rds")
```

Let's also format photovoltaic output, which I separately measured for each Japanese municipality. I took the average of all raster cells located within each municipality, borrowing data from the World Bank's Global Solar Atlas. See more information here: https://globalsolaratlas.info/download (https://globalsolaratlas.info/download)

Hide

```
# Gather average photovoltaic output from Japanese municipalities from Fraser 2020's 7-c
ountry study of solar power
readxl::read_excel("company_network/census_jp/Japan_pop_v2.xlsx") %>%
  select(muni_code = Code, pvout = Avg_PV_output, windspeed = Windspeed_2018) %>%
  saveRDS("company_network/census_jp/jp_pvout.rds")
```

Finally, let’s combine all of these datasets together into one final datafile, “jp_dataset.rds”.

Hide

```
# Import towns
read_csv("jp_solar/muni_code.csv") %>%
  # Join in photovoltaic output per town
  left_join(y = read_rds("company_network/census_jp/jp_pvout.rds"),
            by = "muni_code") %>%
  # Join in demographic data by year and town
  left_join(y = read_rds("company_network/census_jp/stats_wide_filled_imputed.rds") %>%
              filter(year == 2010) %>% select(-pref, -pref_code, -muni, -year),
            by = c("muni_code")) %>%
  # Join in social cpaital indices
  left_join(y  = read_csv("company_network/census_jp/indices_V2_2020_10_28.csv") %>%
  filter(year == 2010) %>%
  select(muni_code, social_capital, bonding, bridging, linking, vulnerability),
  by = "muni_code") %>%

  # Now fix the year to how it should be
  arrange(muni_code) %>%
  # Now grab just specific variables, named accordingly
  select(
    pref,
    pref_code,
    muni,
    muni_code,

    # Environmental Governance Capacity
    # How much did that city change its emissions from 1990 to 2005?
    # Baseline measure of long-term environmental governance capacity
    #emissions_1990_2005,

    # Renewable Resources
    pvout, windspeed,

    # Demographics & Basic Traits
    pop, pop_women,
    pop_age_65_plus,
    area_inhabitable,
    # Socioeconomics
    income_per_capita,
    unemployment,
    employees_primary, employees_secondary,
    employees_tertiary, pop_college,
    land_price_res,
    # Governance Capacity
    employees_muni,fin_str_index,ratio_rev_exp,
    # Urban Regime Outputs
    exp_commerce_manuf,
    exp_agr_fish_for,
    exp_social_welfare,
    exp_health, exp_public_works,
    exp_fire, exp_education,
    # Economic Outputs
    value_manuf_mill, value_agr_mill, value_commerce_mill,
```

```
    # Community Resources
    social_capital, bonding,
    bridging, linking,
    vulnerability,
    total_migration_rate,
    inmigrants,
    outmigrants,
    # Disaster data
    fukushima, tsunami,
    death_2011, destroy_2011,
    damage_2011,
    exclusion_zone,
    disaster_damage) %>%
  arrange(muni_code) %>%
  saveRDS("company_network/census_jp/jp_dataset.rds")
```

5. Build Network Edgelist

Next, we want to build our network edgelist, which means adapting the original format of the population and sample into a tidygraph object we can easily analyze and visualize.

Hide

```
companies <- read_rds("company_network/jp_companies_cleaned.rds") %>%
  # Fill in the following data points, which are missing 7 % of data points
  mutate_at(vars(start_date, value),
            funs(if_else(is.na(.), mean(., na.rm = TRUE), .))) %>%
  # Now fill in offices with the modal category, 0-3
  mutate(offices = if_else(is.na(offices), "0-3", as.character(offices)) %>%
           factor(levels = c("0-3", "4-10", "11 or more")))


# Import towns
muni <- read_csv("company_network/census_jp/muni_code.csv") %>%
  left_join(by = "muni_code",
            y = read_rds("company_network/census_jp/jp_dataset.rds") %>%
              ungroup() %>%
              select(pop, muni_code) %>%
              distinct()) %>%
  # filter to our study portion
  filter(pop >= 30000)


# Bind together the node traits into one dataframe
nodes <- bind_rows(muni %>%
                     mutate(id = paste("city", muni_code, sep = "_")) %>%
                     mutate(mode = "city"),
                   companies %>%
                     mutate(id = paste("company", company_id, sep = "_")) %>%
                     mutate(mode = "company"))

nodes %>%
  saveRDS("company_network/jp_nodes.rds")

# Import plants
edgelist <- read_rds("company_network/jp_plants_sample.rds") %>%
  mutate(company_id = paste("company", company_id, sep = "_"),
         muni_code = paste("city", geo, sep = "_")) %>%
  group_by(company_id, muni_code) %>%
  summarize(size_kw = sum(size_kw, na.rm = TRUE),
            plants = n()) %>%
  ungroup()

expand_grid(
  company_id = paste("company",
                     companies$company_id %>% unique() %>% sort(),
                     sep = "_"),
  muni_code = paste("city",
                    muni$muni_code %>% unique() %>% sort(),
                    sep = "_")) %>%
  left_join(by = c("company_id", "muni_code" ), y = edgelist) %>%
  # Join in company size
  left_join(by = c("company_id" = "id"), y = nodes %>%
              select(id, employees)) %>%
  # Join in population
  left_join(by = c("muni_code" = "id"), y = nodes %>%
```

```
              select(id, pop)) %>%
  # Calculate number of solar farms per 100,000 residents
  mutate(outcome = plants / pop * 100000) %>%
  mutate_at(vars(size_kw, plants, outcome), funs(if_else(is.na(.), 0, as.numeric(.) )))
 %>%
  saveRDS("company_network/jp_edgelist.rds")

remove(edgelist)
remove(muni, companies)
```

5.1 Make Tidygraph object

Let's gather the observed dataset here.

Hide

```
# Remove data
#rm(list = ls())


read_rds("company_network/jp_edgelist.rds") %>%
  filter(plants > 0) %>%
  select(-size_kw) %>%
    # Create the object in tidygraph, which is great for its versality
  # it imports all vectors in the node data.frame and edge data.frame
  tidygraph::tbl_graph(
    edges = .,
    nodes = nodes,
    directed = FALSE, node_key = "id") %>%
  # drop unneded extra ids
  select(-company_id, -muni_code) %>%
  # Join in a dditional traits
  left_join(by = "id", y = bind_rows(
  read_rds("company_network/jp_edgelist.rds") %>%
    filter(plants > 0) %>%
    group_by(id = company_id) %>%
    summarize(total_plants = sum(plants)),

  read_rds("company_network/jp_edgelist.rds") %>%
    filter(plants > 0) %>%
    group_by(id = muni_code) %>%
    summarize(total_plants = sum(plants)))) %>%
  # And create some extra variables
    mutate(total_plants_cat = cut(total_plants,
                                  breaks = c(0, 50, 100, 500, 1000, 2000),
                                  labels = c("0-45", "50-99", "100-499",
                                             "500-999", "1000-2000"))) %>%
  # create a population controlled edge-weight measure
  #activate("edges") %>%
  #mutate(outcome = plants / .N()$pop[from] * 1000) %>%
  activate("nodes") %>%

  saveRDS("company_network/jp_network.rds")
```

5.2 Visualize

We can visualize this dataset here. It ends up looking like a giant solar donut, though, so it is not particularly helpful as an analytical tool.

Hide

```
library(tidyverse)
library(tidygraph)
library(ggraph)

g1 <- read_rds("company_network/jp_network.rds")  %>%

  activate("nodes") %>%
  filter(!node_is_isolated()) %>%

  activate("edges") %>%
  rename(`Solar Farms\nper 1000 residents` = outcome) %>%
  ggraph(layout = "stress") +
  geom_edge_diagonal(aes(width = `Solar Farms\nper 1000 residents`),
                     color = "grey", alpha = 0.5) +
  geom_node_point(aes(size = log(total_plants + 0.0001) + 1),
                  color = "white") +
  geom_node_point(aes(color = mode, size = log(total_plants + 0.0001) ), alpha = 0.5) +

  scale_color_viridis(option = "plasma", begin = 0.2, end = 0.8, discrete = TRUE)+
  theme_graph() +
  theme(legend.position = 'bottom', legend.box = "horizontal",
        plot.caption = element_text(hjust = 0.5),
        panel.border = element_rect(fill = NA, color = "black"),
        plot.subtitle =  element_text(hjust = 0.5)) +
  labs(
    subtitle = "Observed\n \n",
    color = "Type",
    size = "Total Solar Farm Rate (log)\nper City/Company")

g1 +
  ggsave("viz/figure_1.png", height = 6, width = 10, dpi = 500)
```

Hide

```
rm(list=ls())
```

5.3 Finalize Network Dataset

Make Dataset! This means filling in missing data with the mean and mode too.

Hide

```
rm(list = ls())

library(tidyverse)
library(tidygraph)

# write a function to get the modal category in a vector
get_mode = function(data){
  data.frame(value = data) %>%
    filter(!is.na(value)) %>%
    group_by(value) %>%
    summarize(count = n()) %>%
    top_n(n = 1) %>%
    select(value) %>%
    unlist() %>%
    unname() %>%
    return()
}



get_grid = function(graph){
  edges <- graph %>%
    activate("edges") %>%
    as.data.frame() %>%
    select(from, to, outcome, plants)

  nodes <- graph %>%
    activate("nodes") %>%
    as.data.frame() %>%
    # create a unique ID in order,
    # to match the numbering in the edgelist.
    mutate(number = 1:n())


  # Rebuild the complete grid of company-city node pairs
  expand_grid(from_id = nodes %>%
                filter(mode == "city") %>%
                select(id) %>% unlist(),
              to_id = nodes %>%
                filter(mode == "company") %>%
                select(id) %>% unlist()) %>%
    # Join in the edgelist, using understandable node ids
    left_join(by = c("from_id", "to_id"),
              y = edges %>%
                # Join in the right ids
                left_join(by = c("from" = "number"),
                          y = nodes %>% select(number, from_id = id)) %>%
                left_join(by = c("to" = "number"),
                          y = nodes %>% select(number, to_id = id)) %>%
                select(from_id, to_id, plants, outcome)) %>%
    # Fill in all pairs that weren't in the short edgelist with 0,
    # since they had no ties and were not expected to have anymore.
    mutate_at(vars(outcome, plants),
```

```
              funs(if_else(is.na(.), 0, as.numeric(.)))) %>%
  # Join in the node traits,
    left_join(by = c("from_id" = "id"),
              y = nodes %>%
                select(id, pvout:exclusion_zone, disaster_damage)) %>%
    left_join(by = c("to_id" = "id"),
              y = nodes %>%
                select(id,
                       start_date, value, employees,
                       offices, market, company_type)) %>%
    # Reorder columns
    select(from_id, to_id, plants, outcome,
           start_date:company_type,
           pvout:exclusion_zone, disaster_damage) %>%
    return()
}


read_rds("company_network/jp_network.rds")  %>%
  # Merged in nodes
  activate("nodes") %>%
  left_join(by = c("id" = "muni_code"),
            y = read_rds("company_network/census_jp/jp_dataset.rds") %>%
              mutate(muni_code = paste("city", muni_code, sep = "_")) %>%
              select(-pref_code, -pref, -muni) %>%
              rename(pop_2010 = pop) %>%
        mutate_at(vars(pvout:outmigrants, disaster_damage),
          funs(if_else(is.na(.), mean(., na.rm = TRUE), .)))) %>%
# Impute mean for these variables, all of which are missing less than 7% of data points
  saveRDS("company_network/jp_network_ready.rds")


# Transform into the big grid
read_rds("company_network/jp_network_ready.rds") %>%
  # Build the grid
  get_grid() %>%
  # Save the grid
  saveRDS("company_network/jp_grid.rds")
```

These are our final variables.

Hide

```
read_rds("company_network/jp_grid.rds") %>% names()
```

5.4 Map

Finally, let's also map this network. You can't strictly map a bipartite network if the companies are not rooted in a single location; so instead, I'll make a coaffiliation network of cities, to highlight among cities, which share the most companies' plants.

Hide

```
# Process: Let's join in geometry to each edge
# use st_union to make "LINESTRING"
```

Let’s map this coaffiliation network! To do so, we use a shapefile of municipal political boundaries from the Ministry of Land, Infrastructure, and Transportation.

First, we gather the data necessary for some projections.

Hide

```
# This data uses the North East Asia Albers Equal Distance Conic Projection
eqdc = "+proj=eqdc +lat_0=0 +lon_0=0 +lat_1=15 +lat_2=65 +x_0=0 +y_0=0 +ellps=WGS84 +dat
um=WGS84 +units=m +no_defs"

# and also the North East Asia Albers Equal Area Conic Project
eqac = "+proj=aea +lat_1=15 +lat_2=65 +lat_0=30 +lon_0=95 +x_0=0 +y_0=0 +ellps=WGS84 +da
tum=WGS84 +units=m +no_defs"
```

Second, we create a shapefile of municipalities.

Hide

```
library(tidyverse)
library(sf)
library(rgdal)

# First, let's read in our shapefile of municipal political boundaries
read_sf("map/shapes/polbnda_jpn.shp") %>%
  # Transform to Asia North Albers Equal Area Conic projection
  # Obtained here: https://spatialreference.org/ref/esri/102025/
  st_transform(CRS(eqac)) %>%
  # Let's make one row/shape per municipality code
  group_by(muni_code = adm_code) %>%
  summarize(geometry = st_union(geometry)) %>%
  ungroup() %>%
  # Transform to an Azimuthal Equidistant Projection,
  # rotated 180 degrees to allow easy visualization (to make Japan sideways)
  st_transform(crs = CRS(paste0("+proj=aeqd +ellps=sphere +lat_0=90 +lon_0=", 180)))  %
>%
  saveRDS("map/muni_shapes.rds")


# Now generate prefectural boundaries
read_rds("map/muni_shapes.rds") %>%
  group_by(pref_code = str_sub(muni_code, 1,2)) %>%
  summarize(geometry = st_union(geometry)) %>%
  ungroup() %>%
  saveRDS("map/pref_shapes.rds")

# Now let's alter the shape of the bounding box
# to ensure that our plots are more comfortably sizes
box <- read_rds("map/muni_shapes.rds") %>%
  st_bbox() %>% c() %>%
  data.frame(area = .) %>%
  tibble::rownames_to_column(var = "type") %>%
  mutate(area = if_else(type == "xmin", area - area / 5.4, area),
         area = if_else(type == "xmax", area + area / 7.3, area),
         area = if_else(type == "ymin", area - area / 3.5, area),
         area = if_else(type == "ymax", area + area / 80, area)) %>%
  pivot_wider( names_from = type, values_from = area) %>%
  unlist() %>%
  st_bbox()

# Now crop the shapes to just this box
shapes <- read_rds("map/muni_shapes.rds") %>%
  st_crop(y = box)

# Now generate centroids for each municipality
shapes %>%
  group_by(muni_code) %>%
  summarize(geometry = st_centroid(geometry)) %>%
  ungroup() %>%
  st_crop(y = box) %>%
  saveRDS("map/muni_centroids.rds")
```

```
# Get centroids as points, which we can easily join together
centroids <- read_rds("map/muni_centroids.rds") %>%
  mutate(x = st_coordinates(geometry)[,1],
         y = st_coordinates(geometry)[,2]) %>%
  as.data.frame() %>% select(-geometry)

# Now join into that data.frame the points from each
edgelist %>%
  left_join(by =  c("from" = "muni_code"),
            y = centroids %>%
              rename(from_x = x,
                     from_y = y)) %>%
  left_join(by =  c("to" = "muni_code"),
            y = centroids %>%
              rename(to_x = x,
                     to_y = y)) %>%
  saveRDS("map/edges_geocoded.rds")

remove(edgelist, city, company, centroids, nodes)

# Now import the edges again
read_rds("map/edges_geocoded.rds") %>%
  # Filter out any geocodes that didn't make it
  filter(!is.na(from_x) & !is.na(from_y) & !is.na(to_x) & !is.na(to_y)) %>%
  # Create line-springs in C really really fast
  mutate(geometry = sprintf("LINESTRING(%s %s, %s %s)",
                            from_x, from_y,
                            to_x, to_y)) %>%
  # Set the crs and tell the object to become sf format using the geometry column
  st_as_sf(crs = CRS(paste0("+proj=aeqd +ellps=sphere +lat_0=90 +lon_0=", 180)),
           wkt = "geometry") %>%
  select(from, to, weight, geometry) %>%
  saveRDS("map/edges_lines.rds")

# Thanks to Michael Dorman for this lifesaving response on Stack Overflow
# https://stackoverflow.com/questions/51918536/r-create-linestring-from-two-points-in-same-row-in-dataframe
```

Hide

```
ggplot() +
  # Add water barrier
  geom_sf(data = pref, fill = NA, color = "darkgrey", size = 5) +
  # Add dark blue background
  geom_sf(data = pref, fill = "#002A3E", color = NA) +
  # Add edges, colored a beautiful blue
  geom_sf(data = st_as_sf(net, "edges"), color = "#648FFF",
          size = 0.1, alpha = 0.05) +
  # Add sample cities, colored orange and sized by degree
  geom_sf(data = st_as_sf(net, "nodes"),
          mapping = aes(size = degree), color = "#FFB000", alpha = 0.25) +
  # Overlay some prefectural boundaries
  geom_sf(data = pref, fill = NA, color = "black") +
 # Get case studies in there too
  geom_sf(data = cases, color = "black", size = 5) +
    geom_sf(data = cases, color = "white", size = 3.5) +
   theme_void(base_size = 14) +
#  viridis::scale_color_viridis(option = "plasma", discrete = TRUE, na.value = NA_charac
ter_) +
  theme(legend.position = "bottom", legend.box = "vertical") +
#  guides(color = guide_colorbar(barheight = 0.5, barwidth = 10)) +
  labs(size = "Large-N Sample Cities, sized by total # of companies\n shared with other
 cities, weighted by # of Plants Linked") +
  ggsave("map/map_national.png", height = 5, width = 9, dpi = 500)
```

```
Error in grid.newpage() :
  could not open file '/cloud/project/.Rproj.user/shared/notebooks/A83CE304-japan_solar_
network/1/D54C308B422be8b0/cdfsmrqixb94b_t/_rs_chunk_plot_001.png'
```

6. Edge Analysis:

Finally, let's analyze this network! Early tests showed that if we looked at the entire incidence matrix of town-company pairs, we explain barely 1% of the variation. This is very bad. This, combined with experiences in the field, pointed towards the idea that there might be multiple, potentially contrasting pathways to getting lots of solar adoption in your city. As a result, a single parsimonious model isn't really the best way to model this data. Instead, below, I try to wean out city-company pairs that aren't comparable. Here are three quick aspects that might make two sets of city-company pairs not comparable.

1. One pair has lots of solar, while another pair has none. That's not very comparable; there were probably very different experiences in these two different sets of cities and companies. It would be a little better to zoom into just the edgelist - just city-company pairs that actually adopted any solar. By looking at these, we can figure out what drives city-company pairs to adopt *lots* of solar, as opposed to just a little.
2. One pair is located in rural areas vs. urban areas. My past work shows that rural areas experience the solar boom very differently from urban areas, applying few constrains to companies that want to develop there, whereas cities might use matching programs, invitations, community engagement, and lots of other techniques to promote renewables. As a result, I zoom into just cities with a population of 30,000 or more.
3. Still, some company-city pairs may experience very different a) technical conditions, like sunlight and real estate costs or b) socioeconomics and governance conditions. If we really want to test the effects of certain community level interventions and urban regime typologies, we want to look at communities as similar as

possible in terms of traits a and b before running this analysis. So, we use coarsened exact matching to identify a set of good matches.

Goal: Test the intervening effect of community resources on relevant factors

Hide

```
library(tidyverse)
library(tidygraph)
library(MatchIt)
library(cem)
library(Zelig)

# First, we need to be sure that extraneous factors are not at work.
# Let's zoom into just communities whose inhabitable area, land prices, population, and
 pvoutput are above the mean.

# MOST SOLAR OCCURS IN
dat <- read_rds("company_network/jp_grid.rds") %>%
  # Transform measures
  mutate(#income_per_capita = log(income_per_capita),
    #fin_str_index = fin_str_index^2,
    value = log(value),
    employees = log(employees)) %>%
  # Rescale
  mutate_at(vars(start_date, value,  employees,
                 #offices, market, company_type,
                 pvout, pop_2010, area_inhabitable, land_price_res,
                 income_per_capita, unemployment,
                 disaster_damage, #exclusion_zone,
                 employees_muni,
                 exp_agr_fish_for, exp_commerce_manuf,
                 exp_social_welfare, exp_health, exp_education, exp_public_works,
                 #bonding, bridging, linking,
                 total_migration_rate),
            funs(scale(.) %>% as.numeric())) %>%
  # Eliminate cases where people didn't build solar.
  # These are probably pretty different
  filter(plants > 0) %>%
  # I hypothesize that cities with stronger bridging and linking ties
  # build more solar
  mutate(value_cat = ntile(value, 4) %>% factor(),
         employees_cat = ntile(employees, 4) %>% factor()) %>%
  mutate(tech = (pvout + area_inhabitable + -1*land_price_res +
                   exp_public_works) / 4) %>%
  # Consolidate municipal spending into three distinct scaled indicators:
  #mutate(regime_dev = (exp_agr_fish_for + exp_commerce_manuf) / 2,
  #       regime_opp = (exp_social_welfare),
  #       regime_prog = (exp_education + exp_health) / 2) %>%
  select(-exp_agr_fish_for, -exp_commerce_manuf,
         -exp_social_welfare, -exp_education, -exp_health) %>%
  # Calculate a capacity index
  mutate(capacity = (fin_str_index + ratio_rev_exp + employees_muni) / 3) %>%
  mutate(socioeconomics = (income_per_capita + -1*unemployment) / 2) %>%
  # Join in new urban regime scores
  left_join(by = c("from_id" =  "muni_code"),
            y = read_csv("company_network/urban_regimes_scores.csv") %>%
              filter(year == 2010) %>%
              select(-year, -muni) %>%
              mutate(muni_code = paste("city", muni_code, sep = "_"))) %>%
```

```
# Finally, rescale all of these from 1 to 10 for standardized effects,
# which can be easily multiplied, log transformed, etc.
mutate_at(vars(start_date, value,  employees,
               #offices, market, company_type,
               pvout, pop_2010, area_inhabitable, land_price_res,
               income_per_capita, unemployment,
               disaster_damage, #exclusion_zone,
               employees_muni,
               bonding, bridging, linking,
               regime_dev, regime_opp, regime_prog, capacity, socioeconomics,
               total_migration_rate),
          funs(. %>% scales::rescale(to = c(1,10))))
```

6.1 Social Capital Experiment

First, let's identify similar samples in terms of technical resources, socioeconomics, and governance, and let's test the effects of social capital in these samples.

Hide

```
match0 <- dat %>%
  mutate(treatment = (ntile(bridging, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m0 <- dat %>%
  mutate(treatment = (ntile(bridging, 2) - 1)) %>%
  mutate(weights = match0$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + company_type,
     weights = "weights", model = "ls")


match1 <- dat %>%
  mutate(treatment = (ntile(linking, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m1 <- dat %>%
  mutate(treatment = (ntile(linking, 2) - 1)) %>%
  mutate(weights = match1$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + company_type,
     weights = "weights", model = "ls")


## 6.2 Bridging * Linking

match5 <- dat %>%
   mutate(treatment = (ntile(linking, 2) - 1) * (ntile(bridging, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
```

```
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m5 <- dat %>%
  mutate(treatment = (ntile(linking, 2) - 1) * (ntile(bridging, 2) - 1)) %>%
  mutate(weights = match5$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging * linking +
        regime_dev + regime_opp + regime_prog +
        # controls
        capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
       pvout + area_inhabitable + land_price_res + exp_public_works +
                value + employees + start_date + offices + market + # colinearity
        company_type,
      weights = "weights", model = "ls")

#m5 %>% car::vif() # a fair amount of colinearity due to interaction
```

6.2 Urban Regimes

Second, let’s identify similar samples in terms of technical resources, socioeconomics, and governance, and let’s test the effects of urban regimes in these samples.

Hide

```
### Developmental

match6 <- dat %>%
   mutate(treatment = (ntile(regime_dev, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m6 <- dat %>%
   mutate(treatment = (ntile(regime_dev, 2) - 1)) %>%
  mutate(weights = match6$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")

#m6 %>% car::vif() # a fair amount of colinearity due to interaction

#### with Linking

match7 <- dat %>%
   mutate(treatment = (ntile(linking, 2) - 1) * (ntile(regime_dev, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m7 <- dat %>%
   mutate(treatment = (ntile(linking, 2) - 1) * (ntile(regime_dev, 2) - 1)) %>%
  mutate(weights = match7$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking * regime_dev +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")
```

```
### with bridging and linking

match7b <- dat %>%
   mutate(treatment = (ntile(linking, 2) - 1) * (ntile(bridging, 2) - 1) * (ntile(regime
_dev, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m7b <- dat %>%
   mutate(treatment = (ntile(linking, 2) - 1) * (ntile(bridging, 2) - 1) * (ntile(regime
_dev, 2) - 1)) %>%
  mutate(weights = match7b$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging * linking * regime_dev +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + #pop_college +
       #total_migration_rate + #disaster_damage)
     # pvout +
     #  area_inhabitable +
       land_price_res  +
      #  exp_public_works
      value + employees + start_date + offices + market + # colinearity
       company_type,
    weights = "weights", model = "ls")
# So much colinearity, that I need to drop several controls

### Opportunist

match8 <- dat %>%
   mutate(treatment = (ntile(regime_opp, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m8 <- dat %>%
   mutate(treatment = (ntile(regime_opp, 2) - 1)) %>%
  mutate(weights = match8$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
```

```
     weights = "weights", model = "ls")

#m8 %>% car::vif() # a fair amount of colinearity due to interaction

#### with linking

match9 <- dat %>%
   mutate(treatment = (ntile(linking,2)-1)*(ntile(regime_opp, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m9 <- dat %>%
   mutate(treatment = (ntile(linking,2)-1)* (ntile(regime_opp, 2) - 1)) %>%
  mutate(weights = match9$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking * regime_opp +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")


#### with linking

match9b <- dat %>%
   mutate(treatment = (ntile(linking,2)-1)* (ntile(bridging, 2) - 1) *(ntile(regime_opp,
2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m9b <- dat %>%
   mutate(treatment = (ntile(linking,2)-1)* (ntile(bridging, 2) - 1) *(ntile(regime_opp,
2) - 1)) %>%
  mutate(weights = match9b$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging * linking * regime_opp +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
```

```
       company_type,
     weights = "weights", model = "ls")

### Progressive

match10 <- dat %>%
   mutate(treatment = (ntile(regime_prog, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m10 <- dat %>%
   mutate(treatment = (ntile(regime_prog, 2) - 1)) %>%
  mutate(weights = match10$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")

#### with linking

match11 <- dat %>%
   mutate(treatment = (ntile(linking, 2)-1) * (ntile(regime_prog, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m11 <- dat %>%
   mutate(treatment = (ntile(linking, 2)-1) * (ntile(regime_prog, 2) - 1)) %>%
  mutate(weights = match11$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging + linking * regime_prog +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")
```

```
match11b <- dat %>%
   mutate(treatment = (ntile(linking, 2)-1) * (ntile(bridging, 2)-1) * (ntile(regime_pro
g, 2) - 1)) %>%
  # Match by technocratic criteria
  # and socioeconomic criteria
  # and governance criterai
  matchit(formula = treatment ~ pvout + area_inhabitable +
            land_price_res + pop_2010 + exp_public_works +
            socioeconomics + capacity, method = "cem")

m11b <- dat %>%
   mutate(treatment = (ntile(linking, 2)-1) * (ntile(bridging, 2)-1) * (ntile(regime_pro
g, 2) - 1)) %>%
  mutate(weights = match11b$weights) %>%
  filter(weights != 0) %>%
  zelig(formula = log(outcome) ~ bonding + bridging * linking * regime_prog +
       regime_dev + regime_opp + regime_prog +
       # controls
       capacity + socioeconomics + pop_college + total_migration_rate + disaster_damage
 +
      pvout + area_inhabitable + land_price_res + exp_public_works +
               value + employees + start_date + offices + market + # colinearity
       company_type,
     weights = "weights", model = "ls")
```

6.3 Texreg Table

Next, let's put these models into a model table!

Hide

```
#library(devtools)
#library(remotes)
#install_github("leifeld/texreg")
library(texreg)

texreg::htmlreg(
  list(m0,m1,m5,m6,#m3,m3b,m4,m4b,m2,m2b,
       m7,m7b,m8,m9,m9b,m10,m11,m11b),
  custom.header = list("Linking x Bridging Social Capital" = 1:3,
                       "Developmental Urban Regimes" = 4:6,
                       "Opportunist Urban Regimes" = 7:9,
                       "Progressive Urban Regimes" = 10:12),
  custom.model.names = c("Bridging", "Linking",
                         "Linking x Bridging",
                         "Developmental", "Developmental x Linking", "Developmental x Li
nking x Bridging",
                         "Opportunist", "Opportunist x Linking", "Opportunist x Linking
 x Bridging",
                         "Progressive", "Progressive x Linking", "Progressive x Linking
 x Bridging"),
  custom.coef.map = list(
    # Social Capital
    "bonding" = "Bonding Social Capital",
    "bridging" = "Bridging Social Capital",
    "linking" = "Linking Social Capital",
    "bridging:linking" = "Linking x Bridging",
    # Urban Regime
    "regime_dev" = "Developmental Regime",
    "linking:regime_dev" = "Developmental x Linking",
    "bridging:regime_dev" = "Developmental x Bridging",
    "bridging:linking:regime_dev" = "Developmental x Linking x Bridging",
    "regime_opp" = "Opportunist Regime",
    "linking:regime_opp" = "Opportunist x Linking",
     "bridging:regime_opp" = "Opportunist x Bridging",
    "bridging:linking:regime_opp" = "Opportunist x Linking x Bridging",
    "regime_prog" = "Progressive Regime",
    "linking:regime_prog" = "Progressive x Linking",
     "bridging:regime_prog" = "Progressive x Bridging",
    "bridging:linking:regime_prog" = "Progressive x Linking x Bridging",
    # Company Traits
    "employees" = "Company Employees (log)",
    "linking:employees" = "Company Employees x Linking",
    "bridging:linking:employees" = "Company Employees x Linking x Bridging",
    "value" = "Company Value (log)",
    "start_date" = "Date Founded",
    "marketlisted" = "Publicly Traded (1/0)",
    "offices11 or more" = "# of Branches: 11 or more",
    "offices4-10" = "# of Branches: 4-10",
    "company_typeauto" = "Sector: Auto",
    "company_typeconstruction/engineering" = "Sector: Construction/Engineering",
    "company_typeenergy" = "Sector: Energy",
    "company_typefood/agriculture" = "Sector: Food/Agriculture",
    "company_typemanufacturing/electronics" = "Sector: Manufacturing/Electronics",
```

```
    "company_typereal estate" = "Sector: Real Estate",
    "company_typetransport" = "Sector: Transport",
    "company_typewaste" = "Sector: Waste",
    # Controls
    "capacity" = "Government Capacity Index",
    "socioeconomics" = "Socioeconomic Status Index",
    "pop_college" = "% College Educated",
    "total_migration_rate" = "Total Migration Rate",
    "disaster_damage" = "Disaster Damage Rate",
    "pvout" = "Sunlight",
    "area_inhabitable" = "Inhabitable Area",
    "land_price_res" = "Cost of Real Estate",
    "exp_public_works" = "Public Works Quality",
    "(Intercept)" = "Constant"),
  groups = list(
    "<b>Social Capital</b>" = 1:4,
    "<b>Urban Regimes</b>" = 5:16,
    "<b>Company Traits</b>" = 17:32,
    "<b>Controls</b>" = 33:40),
  stars = c(0.001, 0.01, 0.05, 0.10),
  bold = 0.10,
  file = "viz/matching_tables.html",
  custom.note = "Since network data suffers from extreme heteroskedasticity, p-values ar
e suspect. Beta coefficients are reliable, but simulation or permutation methods, which
 are not affected by heteroskedasticity, should be used to assess confidence intervals a
nd type II error.",
  caption = "<b>OLS models of Solar Farms per 100,000 residents (log transformed right-s
kewed outcome)</b><br><i>Beta Coefficients used for Simulation Tests</i>",
  caption.above = TRUE
)
```

6.4 Simulated Effects

Because city-company pairs are naturally interrelated, modeling standard errors are suspect, and we need to evaluate the relevance of these observations a different way. Below, I estimate simulated effects, which use statistical simulation of predicted values using a multivariate normal distribution to side-step the issue of standard errors and p-values entirely. Simulation works fine even on very heteroskedastic (interdependent) data.

Hide

```
# Get first differences between the 25th and 75th percentiles
f0 <- m0 %>%
  sim(x = setx(., linking = quantile(dat$bridging, probs = .25, na.rm = TRUE)),
      x1 = setx(., linking = quantile(dat$bridging, probs = .75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Bridging",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))

# Get first differences between the 25th and 75th percentiles
f1 <- m1 %>%
  sim(x = setx(., linking = quantile(dat$linking, probs = .25, na.rm = TRUE)),
      x1 = setx(., linking = quantile(dat$linking, probs = .75, na.rm = TRUE))) %>%
   with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Linking",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))



# Linking x Bridging interaction
f5 <- m5 %>%
  sim(x = setx(., linking = quantile(dat$linking, probs = .25, na.rm = TRUE),
               bridging = quantile(dat$bridging, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., linking = quantile(dat$linking, probs = .75, na.rm = TRUE),
                bridging = quantile(dat$bridging, probs = 0.75, na.rm = TRUE))) %>%
    with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Linking x Bridging",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))



# Development
f6 <- m6 %>%
```

```
  sim(x = setx(., regime_dev = quantile(dat$regime_dev, probs = .25, na.rm = TRUE)),
      x1 = setx(., regime_dev = quantile(dat$regime_dev, probs = .75, na.rm = TRUE))) %
>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Developmental",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))



# Developmentx Linking
f7 <- m7 %>%
  sim(x = setx(., regime_dev = quantile(dat$regime_dev, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_dev = quantile(dat$regime_dev, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Developmental x Linking",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))

# Development x Linking x Bridging
f7b <- m7b %>%
  sim(x = setx(., regime_dev = quantile(dat$regime_dev, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE),
               bridging = quantile(dat$bridging, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_dev = quantile(dat$regime_dev, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE),
                bridging = quantile(dat$bridging, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Developmental x Linking x Bridging",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))

# Opportunist
f8 <- m8 %>%
```

```
  sim(x = setx(., regime_opp = quantile(dat$regime_opp, probs = .25, na.rm = TRUE)),
      x1 = setx(., regime_opp = quantile(dat$regime_opp, probs = .75, na.rm = TRUE))) %
>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Opportunist",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))


# Opportunist x Linking
f9 <- m9 %>%
  sim(x = setx(., regime_opp = quantile(dat$regime_opp, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_opp = quantile(dat$regime_opp, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Opportunist x Linking",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))

# Opportunist x Linking x Bridging
f9b <- m9b %>%
  sim(x = setx(., regime_opp = quantile(dat$regime_opp, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE),
               bridging = quantile(dat$bridging, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_opp = quantile(dat$regime_opp, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE),
                bridging = quantile(dat$bridging, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Opportunist x Linking x Bridging",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))



# Progressive
```

```
f10 <- m10 %>%
  sim(x = setx(., regime_prog = quantile(dat$regime_prog, probs = .25, na.rm = TRUE)),
      x1 = setx(., regime_prog = quantile(dat$regime_prog, probs = .75, na.rm = TRUE)))
 %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Progressive",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))


# Progressive x Linking
f11 <- m11 %>%
  sim(x = setx(., regime_prog = quantile(dat$regime_prog, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_prog = quantile(dat$regime_prog, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Progressive x Linking",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))

# Progressive x Linking
f11b <- m11b %>%
  sim(x = setx(., regime_prog = quantile(dat$regime_prog, probs = .25, na.rm = TRUE),
               linking = quantile(dat$linking, probs = 0.25, na.rm = TRUE),
               bridging = quantile(dat$bridging, probs = 0.25, na.rm = TRUE)),
      x1 = setx(., regime_prog = quantile(dat$regime_prog, probs = .75, na.rm = TRUE),
                linking = quantile(dat$linking, probs = 0.75, na.rm = TRUE),
                bridging = quantile(dat$bridging, probs = 0.75, na.rm = TRUE))) %>%
  with(sim.out) %>%
  with(x1) %>%
  with(fd) %>%
  data.frame() %>%
  rename(fd = 1) %>%
  mutate(fd = exp(fd)) %>%
  summarize(term = "Progressive x Linking x Bridging",
            qi_median = median(fd),
            qi_low = quantile(fd, probs = 0.025),
            qi_high = quantile(fd, probs = 0.975))


# Visualize!
```

```
bind_rows(f0, f1,#f2,f2b,f3,f3b,f4,f4b,
          f5,f6,f7,f7b,f8,f9,f9b,f10,f11, f11b, .id = "order") %>%
  mutate(term = if_else(term == "Developmental x Linking x Bridging",
                        "Developmental x Linking x Bridging*", term)) %>%
  # Fix confidence interval for this one, shortening and
  # leaving an asterisk to a note about it.
  # Since it is so large, we need to shorten it to make others visible.
  #But, I will mention this in the text;
  mutate(qi_high = if_else(term == "Developmental x Linking x Bridging*",
                        2, qi_high)) %>%
  mutate(term = term %>% str_replace("x Linking x Bridging", "x\nLinking x Bridging")) %
>%
  mutate(type = case_when(
    str_detect(term, "Developmental") ~ "Developmental Regimes x Social Capital",
    str_detect(term, "Progressive") ~ "Progressive Regimes x Social Capital",
    str_detect(term, "Opportunist") ~ "Opportunist Regimes x Social Capital",
    str_detect(term, "Developmental|Opportunist|Progressive", negate = TRUE) ~ "Social C
apital") %>%
      factor(levels = c("Social Capital", "Developmental Regimes x Social Capital",
                        "Progressive Regimes x Social Capital",
                        "Opportunist Regimes x Social Capital"))) %>%
  # This is a super weird effect, from a model with high colinearity,
  # so lets get rid of it for now.
 # filter(term != "Developmental x Linking x Bridging") %>%
#    filter(term != "Developmental x Linking") %>%
  mutate(order = as.numeric(order)) %>%
  mutate(significant = if_else(qi_low < 0 & qi_high > 0, "Insignificant", "p < 0.05") %
>%
           factor(levels = c( "p < 0.05", "Insignificant"))) %>%
  ggplot(mapping = aes(x = reorder(term, -order), y = qi_median, ymin = qi_low, ymax = q
i_high,
                       color = type)) +
  geom_hline(yintercept=  0, linetype = "dashed", color = "darkgrey") +
  geom_linerange(size = 1.25) +
  geom_point(size = 4.5) +
  geom_point(size = 2.5, color = "white", show.legend = TRUE) +
  coord_flip() +
  scale_color_manual(values = c("#DC267F", "#648FFF", "#785EF0", "#FFB000")) +
  labs(y = "Difference in Expected Solar Farms\nper 100,000 residents",
       x = "",
       color = "Statistical\nSignificance") +
#       caption = "First differences compare outcomes for a predict greater than 75% of
 network edges,\n compared to greater than 25% of edges. Points reflect median, while ba
nds reflect 95% confidence intervals based on 1000 simulations, holding all other predic
tors constant at their means or modes. Upper interval for Developmental x Linking X Brid
ging (18.9) shorted to 2 for visibility.") +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        plot.caption = element_text(hjust = 0.5),
        legend.position = "right") +
  facet_wrap(~type, scales = "free_y", ncol = 1, ) +
  guides(color = FALSE) +
  ggsave("viz/fig_5_matching_fd.png", width = 6, height = 6, dpi = 500)
```

6.5 Matching Comparisons

Finally, let’s generate some helpful visuals demonstrating how much better our matched samples are than the original samples would have been.

Hide

```
library(tidyverse)
library(cem)
library(MatchIt)
# Write a short function to extract quantities of interest,
# like absolute standardized mean difference
get_stats = function(model){
  bind_rows(
    summary(model)$sum.all %>%
      as.data.frame() %>%
      tibble::rownames_to_column(var = "term") %>%
      magrittr::set_colnames(value = names(.) %>%
                               tolower() %>% str_replace_all("[.]", "") %>%
                               str_replace_all(" ", "_")) %>%
      mutate(type = "Unmatched"),

    summary(model)$sum.matched %>%
      as.data.frame() %>%
      tibble::rownames_to_column(var = "term") %>%
      magrittr::set_colnames(value = names(.) %>%
                               tolower() %>% str_replace_all("[.]", "") %>%
                               str_replace_all(" ", "_")) %>%
      mutate(type = "Matched")) %>%
  return()
}

out <- list(match0, match1, #match2, match2b,
            #match3, match3b, match4, match4b,
            match5, match6, match7, match7b,
             match10, match11, match11b,
            match8, match9, match9b) %>%
  map_dfr(~get_stats(.), .id = "model") %>%
  mutate(model = model %>% recode_factor(
    "1" = "Bridging", "2" = "Linking",  "3" = "Linking\nx Bridging",
    #"3" = "Company Size\nx Linking","4" = "Company Size\nx Linking x Bridging",
    #"5" = "Small Companies\nwith Linking", "6" = "Small Companies\nwith Linking x Bridg
ing",
    #"7" = "Big Companies\nwith Linking", "8" = "Big Companies\nwith Linking x Bridgin
g",
    "4" = "Developmental",
    "5" = "Developmental\nx Linking",  "6" = "Developmental\nx Linking x Bridging",
    "7" = "Progressive",
    "8" = "Progressive\nx Linking", "9" = "Progressive\nx Linking x Bridging",
    "10" = "Opportunist", "11" = "Opportunist x Linking",
    "12" = "Opportunist\nx Linking x Bridging", )) %>%
  mutate(term = term %>% recode_factor(
    "pop_2010" = "Population",
    "area_inhabitable" = "Area",
    "pvout" = "Sunlight",
    "land_price_res" = "Cost of Real Estate",
        "exp_public_works" = "Public Works",
    "socioeconomics" = "Socioeconomics",
    "capacity" = "Gov't Capacity"))
```

```
out %>%
  left_join(by = c("model", "term"),
            y = out %>%
              filter(type == "Unmatched") %>%
              mutate(order = abs(std_mean_diff)) %>%
              select(model, term, order)) %>%
  ggplot(mapping = aes(x = reorder(term, order),
                       y = abs(std_mean_diff), color = type, group = type)) +
  geom_hline(yintercept = 0, linetype = "solid", color = "black") +
  geom_hline(yintercept = 0.1, linetype = "dashed", color = "black") +
  geom_line() +
  geom_point(size = 2.5) +
  geom_point(size = 1.5, color = "white", show.legend = TRUE) +
#  coord_flip() +
  facet_wrap(~model, ncol = 6, scales = "free_y") +
  theme_classic(base_size = 12) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(hjust = 1, angle = 30),
        plot.subtitle = element_text(hjust = 0.5)) +
  labs(x = "Matching Covariates", subtitle = "Covariate Balance Before/After Coarsened E
xact Matching",
       color = "Sample", y = "Absolute Standardized Mean Difference") +
  ggsave("viz/fig_A4_balance.png", width = 13, height = 5.5, dpi = 500)
```

6.6 Pathways

Last, let's make a descriptive visual for assessing the interaction effect of bridging and linking social capital on each urban regime.

Hide

```
dat %>%
  mutate(weights = match7b$weights) %>%
  filter(weights != 0) %>%
  ggplot(mapping = aes(x = regime_dev*bridging*linking, y = outcome)) +
  #geom_jitter() +
  geom_point() +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour= NA) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  scale_x_log10()

bind_rows(
  dat %>%
    mutate(weights = match7b$weights) %>%
    filter(weights != 0) %>%
    rename(regime = regime_dev) %>%
    mutate(num = nrow(.)) %>%
    mutate(type = paste("Developmental", " (n = ", num, ")", sep = "")),
  dat %>%
    mutate(weights = match9b$weights) %>%
    filter(weights != 0) %>%
    rename(regime = regime_opp) %>%
    mutate(num = nrow(.)) %>%
    mutate(type = paste("Opportunist", " (n = ", num, ")", sep = "")),
  dat %>%
    mutate(weights = match11b$weights) %>%
    filter(weights != 0) %>%
    rename(regime = regime_prog) %>%
    mutate(num = nrow(.)) %>%
    mutate(type = paste("Progressive", " (n = ", num, ")", sep = ""))) %>%
  ggplot(mapping = aes(x = regime*bridging*linking*weights, y = outcome*weights)) +
  #geom_jitter() +
  geom_point(color = "black") +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour= NA, alpha = 0.75) +
  geom_smooth(method = "lm", se = FALSE, color = "white") +
  scale_y_log10() +
  scale_x_log10() +
  viridis::scale_fill_viridis(option = "plasma") +
  facet_wrap(~reorder(type, num)) +
  theme_minimal(base_size = 14) +
  theme(panel.border =element_rect(fill = NA, color = "black"),
        panel.grid.minor = element_blank(),
        plot.caption = element_text(hjust = 0.5)) +
  labs(fill = "Cumulative\nDensity",
       x = "Urban Regime Index x Bridging x Linking Social Capital",
       y = "Solar Farms per 100,000 residents",
       caption = "Both axes weighted by Coarsened Exact Matching Weights.") +
  ggsave("viz/fig_6_regimes_pathways.png", width = 8, height = 4, dpi = 500)
```

7. Descriptive Visuals

Can we find multiple distinct pathways to adopting solar based on the *people*, actors, and conditions involved? Does this match up with the story about urban regimes and social capital described above?

First, let's import a population dataset and our medium-N sample dataset.

Hide

```
library(tidyverse)

# Import all 1741 municipality traits
pop <- read_csv("company_network/census_jp/muni_code.csv") %>%
  select(muni_code) %>%
  # Tally and join in number of solar farms
  left_join(by = "muni_code",
            y = read_rds("company_network/jp_plants_population.rds")  %>%
  filter(type == "太陽光") %>%
    filter(size_kw >= 10) %>%
  group_by(muni_code = geo) %>%
  summarize(plants = n(),
            size_kw = sum(size_kw, na.rm = TRUE))) %>%
  mutate_at(vars(plants:size_kw), funs(if_else(is.na(.), 0, as.numeric(.)))) %>%
  # Join in population traits
  left_join(by = "muni_code", y = read_rds("company_network/census_jp/jp_dataset.rds"))
 %>%
  mutate(pop_density = pop / area_inhabitable * 1000,
         outcome = plants / pop * 1000) %>%

   # Join in new urban regime scores
  left_join(by = c( "muni_code"),
            y = read_csv("company_network/urban_regimes_scores.csv") %>%
              filter(year == 2010) %>%
              select(-year, -muni))

# Import qualitative analysis data
dat <- read_rds("company_network/qca_dataset.rds") %>%
  select(pref, pref_jp, muni, muni_en = english, summary,muni_bureaucrat:public_re, don
e) %>%
  filter(done == "x") %>%
  # Join in municipality codes
  left_join(by = c("pref_jp" = "pref", "muni"),
            y = read_csv("company_network/census_jp/muni_code.csv")  %>%
              select(pref, muni, muni_code)) %>%
  # Fix two missing municipality codes
  mutate(muni_code = if_else(muni_en == "Kanmaki-cho", "29424", muni_code),
         muni_code = if_else(muni_en == "Tomakomai-shi", "01213", muni_code)) %>%
  # Join in basic population traits
  left_join(by = "muni_code", y = pop %>% select(-pref, -muni))
```

7.1 Other Pathways

Let's try visualizing the pathways one more way.

Hide

```
out <- dat %>%
  mutate(pop = if_else(pop >= 30000, "yes", "no")) %>%
  mutate(outcome = if_else(outcome > median(outcome, na.rm = TRUE), "yes", "no")) %>%
  mutate_at(vars(bonding,bridging,linking,regime_opp, regime_prog, regime_dev),
            funs(if_else(. >= median(., na.rm = TRUE), true = "yes", false ="no"))) %>%
  select(muni_code, outcome, muni_bureaucrat:public_re, pop,
         bonding,bridging,linking, regime_opp, regime_prog, regime_dev)

# y - axis = outcome,
# x - axis = bridging
# color = regime
library(ggpubr)


out <- pop %>%
  filter(pop >= 30000) %>%
  mutate_at(vars(regime_dev, regime_prog, regime_opp),
            funs(if_else(. > median(., na.rm = TRUE), 1, 0))) %>%
  mutate(type = case_when(
    regime_dev == 1 & regime_opp == 0 & regime_prog == 0 ~ "Developmental",
    regime_dev == 0 & regime_opp == 1 & regime_prog == 0 ~ "Opportunist",
    regime_dev == 0 & regime_opp == 0 & regime_prog == 1 ~ "Progressive",
    regime_dev == 0 & regime_opp == 0 & regime_prog == 0 ~ "Caretaker",
    regime_dev == 1 & regime_opp == 1 & regime_prog == 1 ~ "All Three Regimes")) %>%
  mutate(type = if_else(is.na(type), "Hybrid", type))


out %>%
  mutate(type = type %>% factor(levels = c("Developmental", "Opportunist", "Progressive"
,
                                           "Caretaker", "Hybrid", "All Three Regimes")))
%>%
  ggplot(mapping = aes(x = bridging*linking, y = outcome)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour= NA) +
  #geom_point(color = "white", alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_log10() +
  facet_wrap(~type) +
  theme_minimal(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        panel.spacing = unit(0.5, "cm"),
        plot.subtitle = element_text(hjust = 0.5))  +
  viridis::scale_fill_viridis(option = "plasma") +
  labs(x = "Bridging x Linking Social Capital",
       y = "Solar Farms per 1,000 residents (log)", fill = "Cumulative\nDensity",
       subtitle = "Solar Adoption Pathways by Urban Regime & Social Capital (n = 805)")
 +
  guides(fill = guide_colorbar(barwidth = 0.5, barheight = 10, frame.colour = "black"))
 +
  ggsave("viz/other_pathways.png", width = 8, height = 4, dpi = 500)
```

7.2 Medium-N Sample Tests

Let’s example our sample using the same one-sample t-test as before.

Hide

```
library(tidyverse)

vars <- dat %>%
  select(plants:size_kw,outcome, pvout:total_migration_rate, disaster_damage, pop_densit
y) %>%
  names()

mytest = function(myvariable){
  print(myvariable)
  # Our sample of towns with any population is remarkably good
  t.test(x = dat %>% select(myvariable) %>% unlist(),
         mu = mean(x = pop %>% select(myvariable) %>% unlist(), na.rm = TRUE),
         alternative = "two.sided", conf.level = 0.95, paired = FALSE) %>%
    broom::tidy() %>%
    mutate(pop_mean = pop %>% select(myvariable) %>% unlist() %>% mean(na.rm = TRUE)) %
>%
    return()
}

out <- data.frame(term = vars) %>%
  split(.$term) %>%
  map_dfr(~mytest(.$term), .id = "term") %>%
  mutate(p.value = round(p.value, 3)) %>%

    mutate(type = term %>% dplyr::recode(
    "value_agr_mill" = "Economics",
    "linking" = "Community Resources",
    "bridging" = "Community Resources",
    "social_capital" = "Community Resources",
    "fin_str_index" = "Solar/Governance",
    "bonding" = "Community Resources",
    "vulnerability" = "Demographics",
    "employees_muni" = "Solar/Governance",
    "employees_primary" = "Economics",
    "value_commerce_mill" = "Economics",
    "value_manuf_mill" = "Economics",
    "pvout" = "Technical Resources",
    "ratio_rev_exp" = "Solar/Governance",
    "outcome" = "Solar/Governance",
    "windspeed" = "Technical Resources",
    "disaster_damage" = "Demographics",
    "pop_college" = "Demographics",
    "exp_commerce_manuf" = "Spending",
    "exp_fire" = "Spending",
    "employees_secondary" = "Economics",
    "pop_age_65_plus" = "Demographics",
    "exp_social_welfare" = "Spending",
    "exp_agr_fish_for" = "Spending",
    "pop_women" = "Demographics",
    "exp_health" = "Spending",
    "employees_tertiary" = "Economics",
    "exp_public_works" = "Technical Resources",
    "unemployment" = "Demographics",
```

```
    "exp_education" = "Spending",
    "total_migration_rate" = "Demographics",
    "plants" = "Solar/Governance",
    "income_per_capita" = "Demographics",
    "area_inhabitable" = "Technical Resources",
    "pop_density" = "Technical Resources",
    "size_kw" = "Solar/Governance",
    "land_price_res" = "Technical Resources",
    "pop" = "Technical Resources")) %>%
    mutate(term = term %>% dplyr::recode(
    "value_agr_mill" = "Agricultural Output (millions)",
    "linking" = "Linking Social Capital",
    "bridging" = "Bridging Social Capital",
    "social_capital" = "Social Capital",
    "fin_str_index" = "Financial Strength Index",
    "bonding" = "Bonding Social Capital",
    "vulnerability" = "Social Vulnerability",
    "employees_muni" = "Municipal Employees per 1000 residents",
    "employees_primary" = "% Primary Sector Employment",
    "value_commerce_mill" = "Commercial Output (millions)",
    "value_manuf_mill" = "Manufacturing Output (millions)",
    "pvout" = "Photovoltaic Output",
    "ratio_rev_exp" = "Ratio of Revenues to Expenditures",
    "outcome" = "Solar Farms per 1000 residents",
    "windspeed" = "Average Windspeed",
    "disaster_damage" = "Disaster Damaged Buildings\nper 1000 residents",
    "pop_college" = "% College Educated",
    "exp_commerce_manuf" = "Spending on Commerce/Manufacturing",
    "exp_fire" = "Spending on Fire Departments",
    "employees_secondary" = "% Secondary Sector Employment",
    "pop_age_65_plus" = "% Age 65 and over",
    "exp_social_welfare" = "Spending on Social Welfare",
    "exp_agr_fish_for" = "Spending on Agriculture, Fish, & Forestry",
    "pop_women" = "% Women",
    "exp_health" = "Spending on Health",
    "employees_tertiary" = "% Tertiary Sector Employment",
    "exp_public_works" = "Spending on Public Works",
    "unemployment" = "Unemployment Rate",
    "exp_education" = "Spending on Education",
    "total_migration_rate" = "Total Migration Rate",
    "plants" = "Solar Farms",
    "income_per_capita" = "Income per capita",
    "area_inhabitable" = "Inhabitable Area",
    "pop_density" = "Population Density",
    "size_kw" = "Plant Size (kW)",
    "land_price_res" = "Cost of Real Estate",
    "pop" = "Population"))

out %>%
  ggplot(mapping = aes(x = reorder(term, -pop_mean), y = estimate,
                       ymin = conf.low, ymax = conf.high)) +
  geom_linerange(color = "steelblue", alpha = 0.5) +
  geom_point(alpha = 0.5, color = "steelblue", size = 2) +
  geom_point(data = out, mapping = aes(x = reorder(term, -pop_mean), y = pop_mean),
```

```
            color = "firebrick", show.legend = TRUE, size = 2, alpha = 0.5) +
  coord_flip() +
  scale_y_log10() +
  facet_wrap(~type, scales = "free", ncol = 2, strip.position = "right") +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        plot.caption = element_text(hjust = 0.5)) +
  labs(x = "", y = "", caption = "Red dot displays population mean, while blue dot displ
ays sample mean, with 95% confidence intervals.") +
  ggsave("viz/fig_A6_mediumn_sample_traits.png", height = 6, width = 10, dpi = 500)
# Our towns are pretty similar in terms of population density,
# employment traits, spending on public works (mostly),
# governance, income, land prices, linking social ties, solar,
# gender, solar resources, social capital, total migration rates,
# unemployment, agricultural and manufacturing output, vulnerability, and windspeeds.
```

Hide

```
# Our sample of towns with a population over 30,000 is remarkably good
t.test(dat %>%
         filter(pop >= 30000) %>%
         select(plants) %>% unlist(),
       mu = pop %>% filter(pop >= 30000) %>%
         select(plants) %>% unlist() %>% mean(na.rm = TRUE),
       alternative = "two.sided", conf.level = 0.95, paired = FALSE)
```

Hide

```
# Our sample of towns with a population under 30,000 is remarkably good
t.test(dat %>%
         filter(pop < 30000) %>%
         select(plants) %>% unlist(),
       mu = pop %>% filter(pop < 30000) %>%
         select(plants) %>% unlist() %>% mean(na.rm = TRUE),
       alternative = "two.sided", conf.level = 0.95, paired = FALSE)
```

7.3 Histogram

Hide

```
# The median number of solar farms is 161 solar farms

dat %>%
  ggplot(mapping = aes(x = outcome)) +
  geom_histogram(mapping = aes(y = ..density..), color = "black", fill = "darkgrey") +
  geom_density(fill = "purple", alpha = 0.5) +
    geom_vline(xintercept = median(dat$outcome), color = "black", linetype = "dashed") +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        plot.caption = element_text(hjust = 0.5)) +
  labs(x = "Solar Farms per 1,000 residents in Sample",
       y = "% of Sample Cities (n = 90)",
       caption = "Dashed line indicates median rate of solar farms\nper 1000 residents i
n sample (3.1).") +
  ggsave("viz/fig_A7_sample_histogram.png", dpi = 500, width = 6, height = 4)
```

7.4 Actors Bar Chart

Hide

```
dat %>%
  # Make a label for high or low plants
  mutate(type = if_else(outcome > median(outcome,na.rm = TRUE), "High", "Low")) %>%
  mutate(num = if_else(type == "High",
                       true = sum(outcome > median(outcome,na.rm = TRUE)),
                       false = sum(outcome <= median(outcome, na.rm = TRUE)))) %>%
  mutate(type = paste(type, " (n = ", num, ")", sep = "")) %>%
  # Pivot for tidy calculation
  pivot_longer(
    cols = c(muni_bureaucrat:public_re),
    names_to = "measure",
    values_to = "value") %>%
  mutate(measure = measure %>% dplyr::recode(
    "governors" = "Governors",
    "utility" = "Utility",
    "promotion_agency" = "Business\nPromotion Agency",
    "local_benefactor" = "Local Benefactor",
    "matching" = "Matching",
    "mayor_council" = "Mayor or\nCity Council",
    "company_driven" = "Key Company",
    "pref_bureaucrat" = "Prefectural\nBureaucrats",
    "muni_bureaucrat" = "Municipal\nBureaucrats",
    "public_re" = "Public Renewable\nEnergy Program",
    "residents" = "Residents")) %>%
  # For each type and measure of interest,
  # tally number of towns with each type of governance
  # and calculate the mean number of plants for each
  group_by(type, measure) %>%
  summarize(count = sum(value == "yes"),
            outcome = mean(if_else(value == "yes", outcome, 0))) %>%
  ggplot(mapping = aes(x = reorder(measure, -count), y = count,
                       fill = log(outcome), color = type)) +
  geom_col(position = "stack", width = 0.75) +
#  geom_text(color = "white", check_overlap = TRUE) +
  coord_flip() +
  scale_color_manual(values = c("white", "black")) +
  viridis::scale_fill_viridis(option = "plasma", discrete = FALSE, begin = 0.1, end = 0.
8) +
  labs(y = "# of Cities in Sample",
       x = "", subtitle =  "Key Actors in Solar Power Siting",
       fill = "Mean Solar\nFarm Rate (log)",
       color = "Adoption") +
  theme_minimal(base_size = 14) +
  theme(panel.border = element_rect(fill = NA ,color = "black"),
        plot.subtitle = element_text(hjust = 0.5)) +
  guides(fill = guide_colorbar(frame.colour = "black", barwidth = 0.5, barheight = 10))
 +
  ggsave("viz/fig_9_actors_bar.png", dpi = 500, width = 7, height = 4.5)

# AVERAGE NUMBER OF SOLAR FARMS, not TOTAL?
```

8. QCA

After statistical analysis, I turn to Qualitative Comparative Analysis by Charles Ragin. This aims to consolidate the medium-N dataset into a set of multiple distinct pathways to adopting solar based on the *people*, actors, and conditions involved. Then, I assess how these pathways match up with urban regimes and social capital.

Let's conduct QCA below using the CCubes algorithm.

Hide

```
#install.packages("QCA")
# Load QCA Package
library(QCA)
library(tidyverse)

# Format a truth table
tab <- dat %>%
  mutate(pop = if_else(pop >= 30000, "yes", "no")) %>%
  mutate(outcome = if_else(outcome > median(outcome, na.rm = TRUE), "yes", "no")) %>%
  select(muni_en, outcome, muni_bureaucrat:public_re, pop) %>%
  mutate_at(vars(outcome, muni_bureaucrat:public_re, pop), funs(if_else(. == "yes", 1, 0
))) %>%
  tibble::column_to_rownames(var = "muni_en") %>%
  magrittr::set_colnames(value = names(.) %>% str_sub(1,3) %>% toupper()) %>%
  # remove extremely infrequent conditions

  QCA::truthTable(outcome = "OUT",  sort.by = "incl", show.cases = TRUE, incl.cut = 1)
# Remove any rows which are contradictory


# Boolean Minimalization
result <- tab %>%
  QCA::minimize(details = TRUE, sol.cov = .9, include = "C")
```

Now, let's format this table to make a pathways visual.

Hide

```
out <- result$tt$tt %>%
  as.data.frame() %>%
  select(-PRI, -incl) %>%
  arrange(desc(n)) %>%
  mutate(id = 1:n()) %>%
  mutate(OUT = as.numeric(OUT))

out %>%
  pivot_longer(
    cols = -c(id, n, cases),
    names_to = "measure",
    values_to = "value") %>%
  mutate(measure = measure %>% tolower() %>% recode_factor(
    "gov" = "Governor",
    "uti" = "Utility",
    "pro" = "Business\nPromotion Agency",
    "loc" = "Local Benefactor",
    "mat" = "Matching",
    "may" = "Mayor or\nCity Council",
    "com" = "Key Company",
    "pre" = "Prefectural\nBureaucrats",
    "mun" = "Municipal\nBureaucrats",
    "pub" = "Public Renewable\nEnergy Program",
    "res" = "Residents",
    "pop" = "Urban",
    "out" = "Solar Farms")) %>%
  mutate(total = sum(value == "Yes")) %>%
  mutate(highlight = if_else(measure == "Solar Farms", 1, 0)) %>%
  left_join(by = "id", y = out %>%
              select(id, solar = OUT)) %>%
  mutate(solar = solar*n) %>%
  mutate(value = if_else(value == 1, "Yes", "No") %>%
           factor(levels = c("Yes", "No"))) %>%
  ggplot(mapping = aes(x = reorder(cases, -solar), y = reorder(measure, -total), fill =
 factor(value), color = factor(highlight) )) +
  geom_tile(size = 1.15) +
  scale_color_manual(values = c("black", "white")) +
  scale_fill_manual(values = c("#7E03A8FF", "black")) +
  guides(color = FALSE) +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
      plot.subtitle = element_text(hjust = 0.5),
    axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  labs(subtitle = "Solar Adoption Pathways by Key Actors",
    y = "", fill = "Condition\nPresent/\nAbsent",
    x = "Unique Pathways to Solar in 90 Sample Cities\n(Sorted by Outcome and No. of Cit
ies Explained, from most (9) to least (1))",
    caption = "Solar Farm refers to adoption above (Yes) or below (No) the sample media
n.") +
  ggsave("viz/fig_4_pathways_individual.png",width = 10, height = 8, dpi = 500)
```

Let’s adjust this further to classify each of the reduced set of cases into a pathway by theme.

Hide

```
colors <- wesanderson::wes_palette("Zissou1", n = 2,type = "continuous")
# Visualize
viz <- result$primes %>%
  as.data.frame() %>%
  mutate(cases = result$IC$incl.cov$cases) %>%
  mutate(count = cases %>% str_count(pattern = ";")) %>%
  mutate(cases = result$IC$incl.cov$cases %>% str_replace_all(";", "\n")) %>%
  mutate_at(vars(MUN:POP),
            funs(. %>% factor() %>%
                   recode_factor("2" = "Yes", "1" = "No", "0" = "N/A"))) %>%
  mutate(id = 1:n())%>%
  mutate(OUT = "Yes") %>%
  pivot_longer(
    cols = -c(id, cases, count),
    names_to = "measure",
    values_to = "value") %>%
  mutate(measure = measure %>% tolower() %>% recode_factor(
    "out" = "Solar Farms",
    "gov" = "Govern.",
    "uti" = "Utility",
    "pro" = "Business\nPromotion Agency",
    "loc" = "Local Benefactor",
    "mat" = "Matching",
    "may" = "Mayor or\nCity Council",
    "com" = "Key Company",
    "pre" = "Prefectural\nBureaucrats",
    "mun" = "Municipal\nBureaucrats",
    "pub" = "Public Renewable\nEnergy Program",
    "res" = "Residents",
    "pop" = "Urban")) %>%
  group_by(measure) %>%
  mutate(total = sum(value == "Yes")) %>%
  ungroup() %>%
  mutate(highlight = if_else(measure == "Solar Farms", 1, 0))  %>%
  # Classify into themes
  mutate(class = cases %>% recode_factor(
    "Osaki-shi" = "Public Renewable\nEnergy Programs\n& Matching",
    "Higashi-Matsushima-shi" = "Public Renewable\nEnergy Programs\n& Matching",
    "Futtsu-shi\n Hamamatsu-shi" = "Public Renewable\nEnergy Programs\n& Matching",
    "Awaji-shi" = "Public Renewable\nEnergy Programs\n& Matching",

    "Toyohashi-shi\n Nihonmatsu-shi" = "Public Renewable\nEnergy Programs",
    "Arao-shi" = "Public Renewable\nEnergy Programs",
    "Ono-cho" = "Public Renewable\nEnergy Programs",
    "Miyama-shi" = "Public Renewable\nEnergy Programs",

    "Tahara-shi" = "Matching",
    "Minamisoma-shi" = "Matching",

    "Watari-cho\n Ishinomaki-shi" = "Local Bureaucrats\n& Benefactors",
    "Unzen-shi\n Ube-shi" = "Local Bureaucrats\n& Benefactors",
    "Inami-cho\n Arida-shi" = "Local Bureaucrats\n& Benefactors",
    "Abira-cho" = "Rural\nDevelopment",
```

```
    "Kisosaki-cho" = "Rural\nDevelopment",
    "Nakagawa-machi\n Mukawa-cho,Daigo-machi" = "Rural\nDevelopment",
    "Tsubetsu-cho\n Naganuma-cho" = "Rural\nDevelopment",
    "Nakagawa-machi\n Hioki-shi" = "Rural\nDevelopment")) %>%
  mutate(color = if_else(value == "Yes", true = as.character(class), false = value)) %>%
  mutate(color = factor(color,
                        levels = c("N/A", "No", "Rural\nDevelopment",
                                   "Local Bureaucrats\n& Benefactors",
                                   "Matching",
                                   "Public Renewable\nEnergy Programs",
                                   "Public Renewable\nEnergy Programs\n& Matching")))
```

Plot simplified pathways, first normally and then by theme!

Hide

```
# Plot normally
viz %>%
  ggplot(mapping = aes(x = reorder(measure, -total), y = factor(cases),
                       color = factor(highlight),
                       fill = value, label = value)) +
  geom_tile(size = 1.15) +
  geom_text(color = "white") +
  scale_color_manual(values = c("black", "white")) +
  scale_fill_manual(values = c("darkgrey", "black", "#7E03A8FF")) +
  theme_classic(base_size = 14) +
  scale_x_discrete(position = "top")  +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        axis.text.x = element_text(hjust = 0, angle = 25),
        plot.caption = element_text(hjust = 0.5)) +
  guides(fill = FALSE, color = FALSE) +
  labs(x = "Pathways to High Solar Adoption", y = "Pathways (identified by constituent c
ases)",
       caption = "Covers 58% of the whole sample of 90 towns.") +
  ggsave("viz/other_pathways_table.png", width = 13, height = 8, dpi = 400)


# Let's repeat that, but this time, let's classify these pathways further into general,
 interpretable themes.
viz %>%
  ggplot(mapping = aes(x = reorder(measure, -total), y = factor(cases),
                       color = factor(highlight),
                       fill = color, label = value)) +
  geom_tile(size = 1.15) +
  geom_text(color = "white") +
  scale_color_manual(values = c("black", "white")) +
  scale_fill_manual(values = c("darkgrey", "black",# "#7E03A8FF",
                               "#648FFF", "#785EF0", "#DC267F", "#FE6100","#EDA400")) +
                               #"#FFB000"

  theme_classic(base_size = 14) +
  scale_x_discrete(position = "top")  +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        axis.text.x = element_text(hjust = 0, angle = 25),
        plot.caption = element_text(hjust = 0.5),
        panel.spacing = unit(0.5, "cm"),
        strip.text.y = element_text(angle = 0)) +
  guides(fill = FALSE, color = FALSE) +
  labs(x = "Pathways to High Solar Adoption", y = "Pathways (identified by constituent c
ases)",
       caption = "Covers 58% of the whole sample of 90 towns. Pathways classified by the
me.") +
  facet_grid(rows = vars(class), scales = "free", shrink = FALSE, space = "free") +
   ggsave("viz/fig_9_pathways_table_classes.png", width = 13, height = 8, dpi = 400)
```

Urban Regimes comparison

Next, let’s assess qualitatively which urban regime each of these pathway cities fit into, in 2010 and 2017.

2017

Hide

```
# Take the sets of cases per pathway,
paths <- data.frame(cases = result$IC$incl.cov$cases) %>%
  # give each pathway a unique id
  mutate(id = 1:n()) %>%
  # separate the bundles of cases into multiple columns
  separate(col = cases, into = paste("case", 1:3, sep = "_"),
           sep = ";|,", remove = FALSE) %>%
  # Then coalesce them into a tidy data.frame of pathway-case pairs
  pivot_longer(cols = contains("case_"), values_to = "muni_en") %>%
  select(-name) %>%
  filter(!is.na(muni_en)) %>%
  mutate(muni_en = str_remove_all(muni_en, " ")) %>%
  # and now for each pathway, join in the city demographic data
  #for the corresponding case
  left_join(by = "muni_en", y = dat %>%
              mutate(bonding = ntile(bonding, 2) - 1,
                     bridging = ntile(bridging, 2) - 1,
                     linking = ntile(linking, 2) - 1)) %>%
  # Including urban regime scores and categories
  left_join(by = "muni_code",
            y = read_csv("company_network/urban_regimes_categories.csv") %>%
              filter(year == 2017) %>%
              select(-muni)) %>%
  # Let's classify towns based on bonding, bridging, and linking social capital
  mutate(social_capital = case_when(
    bonding == 1 & bridging == 1 & linking == 1 ~ "All Three",
    bonding == 0 & bridging == 1 & linking == 1 ~ "Bridging x Linking",
    bonding == 0 & bridging == 0 & linking == 1 ~ "Linking",
    bonding == 0 & bridging == 1 & linking == 0 ~ "Bridging",
    bonding == 1 & bridging == 0 & linking == 0 ~ "Bonding",
    bonding == 1 & bridging == 1 & linking == 0 ~ "Bonding x Bridging",
    bonding == 0 & bridging == 0 & linking == 0 ~ "None")) %>%
  # Now join in the pathway themes
  left_join(by = c("id"), y = viz %>% rename(cases_formatted = cases)) %>%
  select(id, muni_en, regime = type, social_capital, pathway = class)

# Let's make a tile plot,
# where we situate the PATHWAY NAMES of cities

# Option 1: Let's make a geom_tile
# where x is social capital categories,
# where y is urban regime categories
# and tile contains names of municipalities sorted into these

colors <- c("#648FFF", "#785EF0", "#DC267F", "#FE6100", "#FFB000")

tally <- paths %>%
  # Reclassify Regimes into a More Easily Visible Set
  mutate(regime = regime %>% recode_factor(
    "Caretaker" = "Caretaker",
    "Developmental" = "Developmental",
    "Progressive" = "Progressive",
    "Opportunist" = "Opportunist\n(including\nOpportunist Progressive/\nOpportunist Deve
```

```
lopmental)",
    "Opportunist Progressive" = "Opportunist\n(including\nOpportunist Progressive/\nOppo
rtunist Developmental)",
    "Opportunist Developmental" = "Opportunist\n(including\nOpportunist Progressive/\nOp
portunist Developmental)",
        "All Three" = "Hybrid (All Three)")) %>%
  # Reclassify Social Capital
  mutate(social_capital = social_capital %>% recode_factor(
    "All Three" = "Bonding,\nBridging,\n& Linking",
    "Bridging x Linking" = "Bridging\n & Linking",
    "Bridging" = "Bridging",
    "Bonding x Bridging" = "Bonding\n & Bridging",
    "Linking" = "Linking\n or Bonding",
    "Bonding" = "Linking\n or Bonding",
    "None" = "None")) %>%
  group_by(regime, pathway) %>%
  summarize(count = n(),
            muni_en = paste(sort(unique(muni_en)), collapse = ",\n"))
```

2010

Hide

```
# Take the sets of cases per pathway,
paths <- data.frame(cases = result$IC$incl.cov$cases) %>%
  # give each pathway a unique id
  mutate(id = 1:n()) %>%
  # separate the bundles of cases into multiple columns
  separate(col = cases, into = paste("case", 1:3, sep = "_"),
           sep = ";|,", remove = FALSE) %>%
  # Then coalesce them into a tidy data.frame of pathway-case pairs
  pivot_longer(cols = contains("case_"), values_to = "muni_en") %>%
  select(-name) %>%
  filter(!is.na(muni_en)) %>%
  mutate(muni_en = str_remove_all(muni_en, " ")) %>%
  # and now for each pathway, join in the city demographic data
  #for the corresponding case
  left_join(by = "muni_en", y = dat %>%
              mutate(bonding = ntile(bonding, 2) - 1,
                     bridging = ntile(bridging, 2) - 1,
                     linking = ntile(linking, 2) - 1)) %>%
  # Including urban regime scores and categories
  left_join(by = "muni_code",
            y = read_csv("company_network/urban_regimes_categories.csv") %>%
              filter(year == 2010) %>%
              select(-muni)) %>%
  # Let's classify towns based on bonding, bridging, and linking social capital
  mutate(social_capital = case_when(
    bonding == 1 & bridging == 1 & linking == 1 ~ "All Three",
    bonding == 0 & bridging == 1 & linking == 1 ~ "Bridging x Linking",
    bonding == 0 & bridging == 0 & linking == 1 ~ "Linking",
    bonding == 0 & bridging == 1 & linking == 0 ~ "Bridging",
    bonding == 1 & bridging == 0 & linking == 0 ~ "Bonding",
    bonding == 1 & bridging == 1 & linking == 0 ~ "Bonding x Bridging",
    bonding == 0 & bridging == 0 & linking == 0 ~ "None")) %>%
  # Now join in the pathway themes
  left_join(by = c("id"), y = viz %>% rename(cases_formatted = cases)) %>%
  select(id, muni_en, regime = type, social_capital, pathway = class)

# Let's make a tile plot,
# where we situate the PATHWAY NAMES of cities

# Option 1: Let's make a geom_tile
# where x is social capital categories,
# where y is urban regime categories
# and tile contains names of municipalities sorted into these

colors <- c("#648FFF", "#785EF0", "#DC267F", "#FE6100", "#FFB000")

tally <- paths %>%
  # Reclassify Regimes into a More Easily Visible Set
  mutate(regime = regime %>% recode_factor(
    "Caretaker" = "Caretaker",
    "Developmental" = "Developmental",
    "Progressive" = "Progressive",
    "Opportunist" = "Opportunist\n(including\nOpportunist Progressive/\nOpportunist Deve
```

```
lopmental)",
    "Opportunist Progressive" = "Opportunist\n(including\nOpportunist Progressive/\nOppo
rtunist Developmental)",
    "Opportunist Developmental" = "Opportunist\n(including\nOpportunist Progressive/\nOp
portunist Developmental)",
        "All Three" = "Hybrid (All Three)")) %>%
  # Reclassify Social Capital
  mutate(social_capital = social_capital %>% recode_factor(
    "All Three" = "Bonding,\nBridging,\n& Linking",
    "Bridging x Linking" = "Bridging\n & Linking",
    "Bridging" = "Bridging",
    "Bonding x Bridging" = "Bonding\n & Bridging",
    "Linking" = "Linking\n or Bonding",
    "Bonding" = "Linking\n or Bonding",
    "None" = "None")) %>%
  group_by(regime, pathway) %>%
  summarize(count = n(),
            muni_en = paste(sort(unique(muni_en)), collapse = ",\n"))
```

Change in Social Capital 2010-2017 per pathway

Next, let's visualize the change in social capital scores for each city in each pathway.

Hide

```
# Take the sets of cases per pathway,
paths <- data.frame(cases = result$IC$incl.cov$cases) %>%
  # give each pathway a unique id
  mutate(id = 1:n()) %>%
  # separate the bundles of cases into multiple columns
  separate(col = cases, into = paste("case", 1:3, sep = "_"),
           sep = ";|,", remove = FALSE) %>%
  # Then coalesce them into a tidy data.frame of pathway-case pairs
  pivot_longer(cols = contains("case_"), values_to = "muni_en") %>%
  select(-name) %>%
  filter(!is.na(muni_en)) %>%
  mutate(muni_en = str_remove_all(muni_en, " ")) %>%
  # Now join in the pathway themes
  left_join(by = c("id"), y = viz %>% rename(cases_formatted = cases)) %>%
  left_join(by = c("muni_en"),
            y = dat %>% select(muni_en, muni_code)) %>%
  # Now join in the social capital scores for 2010 and 2017
  left_join(by = c("muni_code"),
            y = read_csv("company_network/census_jp/indices_V2_2020_10_28.csv") %>%
              select(muni_code, year, bonding, bridging, linking) %>%
              pivot_longer(
                cols = c(bonding, bridging, linking),
                names_to = "index",
                values_to = "score") %>%
              # And rescale each number to 1 to 10
              # COMPARED TO THE ENTIRE COUNTRY from 2000 to 2017
              group_by(index) %>%
              mutate(score = scales::rescale(score, to = c(1,10))) %>%
              ungroup() %>%
              # then filter to just 2010 and 2017
              filter(year %in% c(2010, 2017))) %>%
  # Now pivot wider and subtract the difference between 2010 and 2017
  pivot_wider(
    id_cols = -c(year, score),
    names_from = "year",
    values_from = score) %>%
  mutate(diff = `2017` - `2010`) %>%
  # Let's also identify the largest score for each, so as to arrange them easily
  rowwise() %>%
  mutate(order = max(`2010`, `2017`)) %>%
  ungroup()


colors <- c("#648FFF", "#785EF0", "#DC267F", "#FE6100", "#FFB000")

paths %>%
  mutate(index = index %>% recode_factor(
    "bonding" = "Bonding\nSocial Capital",
    "bridging" = "Bridging\nSocial Capital",
    "linking" = "Linking\nSocial Capital")) %>%
  ggplot(mapping = aes(x = reorder(muni_en, order),
                       ymin = `2010`, ymax = `2017`, color = index)) +
```

```
  geom_linerange(position = position_dodge(width = 0.5)) +
  geom_point(mapping = aes(y = `2010`), size = 0.5, position = position_dodge(width = 0.
5)) +
  geom_point(mapping = aes(y = `2017`), size = 2, position = position_dodge(width = 0.5
)) +
  coord_flip() +
  facet_grid(class~index, scales = "free_y", space = "free") +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
        strip.text.y = element_text(angle = 0)) +
  guides(color = FALSE) +
  labs(y = "Index Score (rescaled 1 to 10)", x = "") +
  scale_color_manual(values = colors[c(1,3,5)]) +
  ggsave("viz/fig_10_pathways_social_capital_lines.png", width = 8, height = 6, dpi = 50
0)

# I want...
# Split panels by pathway
# Split panels by index, because this is too hard to read.
```

Pathway Comparison

Hide

```
colors <- c("#ACE3EB", "#648FFF", "#785EF0", "#DC267F", "#FE6100", "#FFB000")

# Take the sets of cases per pathway,
paths <- data.frame(cases = result$IC$incl.cov$cases) %>%
  # give each pathway a unique id
  mutate(id = 1:n()) %>%
  # separate the bundles of cases into multiple columns
  separate(col = cases, into = paste("case", 1:3, sep = "_"),
           sep = ";|,", remove = FALSE) %>%
  # Then coalesce them into a tidy data.frame of pathway-case pairs
  pivot_longer(cols = contains("case_"), values_to = "muni_en") %>%
  select(-name) %>%
  filter(!is.na(muni_en)) %>%
  mutate(muni_en = str_remove_all(muni_en, " ")) %>%
  # Now join in the pathway themes
  left_join(by = c("id"), y = viz %>% rename(cases_formatted = cases)) %>%
  left_join(by = c("muni_en"),
            y = dat %>% select(muni_en, muni_code, bonding, bridging, linking,
                               regime_prog, regime_opp, regime_dev)) %>%
  select(id, muni_en, regime_dev, regime_prog, regime_opp,
         bonding, bridging, linking, pathway = class) %>%
  mutate_at(vars(bonding, bridging, linking, regime_dev, regime_prog, regime_opp),
            funs(scales::rescale(., to = c(1,10))))

paths %>%
  pivot_longer(cols = c(regime_dev, regime_prog, regime_opp,
                        bonding, bridging, linking),
               names_to = "measure",values_to = "value") %>%
  mutate(measure = measure %>% recode_factor(
    "bonding" = "Bonding\nSocial Capital",
    "bridging" = "Bridging\nSocial Capital",
    "linking" = "Linking\nSocial Capital",
    "regime_dev" = "Developmental\nRegime",
        "regime_prog" = "Progressive\nRegime",
    "regime_opp" = "Opportunist\nRegime")) %>%
  ggplot(mapping = aes(x = pathway, y = value, fill = measure)) +
  geom_violin() +
  geom_point(data = . %>%
               group_by(measure, pathway) %>%
               summarize(median = median(value)),
             mapping = aes(x = pathway, y = median),
             size = 3) +
    geom_point(data = . %>%
               group_by(measure, pathway) %>%
               summarize(median = median(value)),
             mapping = aes(x = pathway, y = median),
             size = 2, color = "white", show.legend = TRUE) +
  coord_flip() +
  facet_grid(~measure) +
  scale_fill_manual(values = c(colors)) +
  guides(fill = FALSE) +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black"),
```

```
      panel.spacing = unit(0.5, "cm")) +
scale_y_continuous(breaks = c(2.5, 5, 7.5), labels = c(2.5, 5, 7.5)) +
labs(y = "Index Score (rescaled 1 to 10)",
     subtitle = "Range of Pathway Cities by Social Capital and Urban Regime",
     x = "") +
ggsave("viz/other_range_pathways.png", width = 9.5, height = 4, dpi = 500)
```

Social Capital Descriptives

Let’s also break down our medium-N sample by social capital; this provides more evidence that high adopters tended to have some combination of bridging ties and any other form of high social capital.

Hide

```
colors <- c("#648FFF", "#785EF0", "#DC267F", "#FE6100", "#FFB000")

# Let's find out what percentage of towns in the medium N sample had above average socia
l capital
viz <- dat %>%
  mutate(outcome = if_else(outcome > median(outcome,na.rm = TRUE), "High", "Low") %>%
           factor(levels = c("Low", "High"))) %>%
    # Let's classify towns based on bonding, bridging, and linking social capital
  mutate_at(vars(bonding, bridging, linking), funs(ntile(.,2) - 1)) %>%
  mutate(social_capital = case_when(
    bonding == 1 & bridging == 1 & linking == 1 ~ "Bonding, Bridging,\n& Linking\n",
    bonding == 0 & bridging == 1 & linking == 1 ~ "Bridging & Linking\n",
    bonding == 0 & bridging == 1 & linking == 0 ~ "Bridging\n",
    bonding == 1 & bridging == 1 & linking == 0 ~ "Bonding & Bridging\n",
    bonding == 0 & bridging == 0 & linking == 1 ~ "Bonding\nand/or Linking\n",
    bonding == 1 & bridging == 0 & linking == 0 ~ "Bonding\nand/or Linking\n",
    bonding == 1 & bridging == 0 & linking == 1 ~ "Bonding\nand/or Linking\n",
    bonding == 0 & bridging == 0 & linking == 0 ~ "None\n"))   %>%
  # Tally up number of cities per outcome-social capital type pair
  group_by(outcome, social_capital) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  # Now tally up total cases per social capital type
  group_by(social_capital) %>%
  mutate(total_cases = sum(count)) %>%
  ungroup()  %>%
  mutate(social_capital = paste(social_capital, "(n = ", total_cases, ")\n", sep = ""))
 %>%
  # Now tally up total cases per outcome
  group_by(outcome) %>%
  mutate(outcome_cases = sum(count)) %>%
  ungroup() %>%
  mutate(outcome = paste(outcome, "\n(n = ", outcome_cases, ")", sep = ""))  %>%
  # Now order them
  mutate(outcome = outcome %>% factor(levels = c("Low\n(n = 45)","High\n(n = 45)")),
         social_capital = social_capital %>% factor(levels = c(
           "Bonding, Bridging,\n& Linking\n(n = 15)\n",
           "Bridging & Linking\n(n = 20)\n",
           "Bridging\n(n = 5)\n",
           "Bonding & Bridging\n(n = 5)\n",
           "Bonding\nand/or Linking\n(n = 28)\n",
           "None\n(n = 17)\n")))

viz %>%
  ggplot(mapping = aes(x = outcome, y = count, fill = social_capital)) +
  geom_col(position = "fill", color = "white", size = 1.15) +
  coord_flip()   +
  scale_fill_manual(values = c(rev(colors), "darkgrey"))  +
  theme_classic(base_size = 14) +
  theme(panel.border = element_rect(fill = NA, color = "black")) +
  scale_y_continuous(breaks = c(0,.25,.5,.75,1), labels = c(0,25,50,75,100)) +
  labs(y = "% of Cities in Medium-N Sample (n = 90)",
       x = "Solar Adoption Rate\n(above or below median level\nof 3.1 farms per 1,000 re
```

```
sidents)",
       fill = "Social Capital",
       caption = "Social capital categorized as 'Bridging'\nif above sample median level
of Bridging Social Capital, for example.") +
  ggsave("viz/fig_B5_medium_n_social_capital.png", width = 8, height = 4, dpi = 500)
```

9. Case Studies

Last but not least, let’s gather some data for our qualitative case studies.

Hide

```
# Import towns
muni <- read_csv("company_network/census_jp/muni_code.csv") %>%
  left_join(by = "muni_code",
            y = read_rds("company_network/census_jp/jp_dataset.rds") %>%
              ungroup() %>%
              select(pop, muni_code) %>%
              distinct()) %>%
  # filter to our study portion
  filter(pop >= 30000)


# Generate population data.frame
pop <- read_rds("company_network/jp_plants_population.rds") %>%
  # Filter just to solar farms in cities with populations over 30,000
  filter(geo %in% muni$muni_code)

# These are our case studies
muni %>%
  filter(muni %in% c("淡路市", "東松島市"))


# Get number of solar installations of each size bracket
dat <- pop %>%
  filter(geo %in% c("28226", # Awaji
                    "04214")) %>%# Higashi-Matsushima
  mutate(muni = geo %>% recode_factor(
    "28226" = "Awaji",
    "04214" = "Higashi-Matsushima")) %>%
  group_by(muni) %>%
  summarize(count = n(),
            size_10_49 = sum(size_kw >= 10 & size_kw < 50),
            size_50_499 = sum(size_kw >= 50 & size_kw < 500),
            size_500_1999 = sum(size_kw >= 500 & size_kw < 2000),
            size_2000 = sum(size_kw >= 2000)) %>%
  ungroup() %>%
  pivot_longer(cols = c(count:size_2000),
               names_to = "measure", values_to = "value") %>%
  mutate(type = if_else(measure == "count", "Solar Overall", "Solar by Size"))

# What percentile
rates <- pop %>%
  group_by(geo) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  left_join(by = c("geo" = "muni_code"), y = muni) %>%
  mutate(solar_rate = count / pop * 1000)

#5.57
#17

rates %>%
  summarize(quantile(solar_rate, probs = 0.95))
```