Replication materials

Household trajectories of resilience to acute malnutrition in the Kenyan drylands

Ravi Bhavnani, Nina Schlager, Mirko Reul and Karsten Donnay

This file contains replication information for the analysis performed in Bhavnani et al. (2023). *'Household trajectories of resilience to acute malnutrition in the Kenyan drylands'*, Frontiers in Sustainable Food Systems, DOI: 10.3389/fsufs.2023.1091346

There are 3 distinct steps in the analysis:

1. Data preparation and resilience calculation
    - Geo-coding of wards
    - Ward resilience to acute malnutrition
    - External stressor intensity
2. Latent Class Mixed Model (LCMM)
3. Cox Regression Analysis

Please also see `readme.txt` for additional information on the data inputs and variable coding. Note that most illustrations featured in the published paper have been re-processed in *Inkscape* to maximize clarity.

1. Data preparation and resilience calculation

1.1 Geo-coding of wards

The raw NDMA data used is `MUAC_all.csv`. All observations are geocoded at the ward level (adm3). We therefore compare ward names provided by the NDMA with spatial information from the GADM project. `geocoding_wardnames_all.txt` lists all differences. The adjusted spatial information is stored in `KEN_adm3-ndma.shp` and can be found in the sub-directory `gadm36_KEN_shp`. `helper_functions_geocoding.R` includes a function to check the number of incorrectly coded observations in a sample.

```
# load helper function
source("code/helper_functions_geocoding.R")

# load MUAC data
df <- read.csv("data/MUAC_all.csv")

# re-code county (adm1) names
df$adm1 <- dplyr::recode(df$adm1,
                                "Tana river" = "Tana River",
                                "West pokot" = "West Pokot",
                                "Tharaka Nithi" = "Tharaka-Nithi")

# re-assign county (adm1) names
df <- df %>% mutate(
```

```
      adm1 = replace(adm1, adm3 == "Kamarandi", "Embu"),
      adm1 = replace(adm1, adm3 == "Nanighi", "Garissa"),
      adm1 = replace(adm1, adm3 == "Kapenguria", "West Pokot"),
      adm1 = replace(adm1, adm3 == "Riwo", "Baringo")
    )

# re-code ward (adm3) names
df$adm3 <- dplyr::recode(df$adm3,
                                  "Komolion" = NA_character_,
                                  "Ribko" = "Ribkwo",
                                  "Orus" = NA_character_,
                                  "Saimo Soi" = "Saimo/Soi",
                                  "Akoret" = NA_character_,
                                  "Kanyuambora" = NA_character_,
                                  "Kamarandi" = NA_character_,
                                  "Abakayle" = "Abakaile",
                                  "Oldonyiro" = "Oldo/Nyiro",
                                  "Charri" = "Chari",
                                  "Bula Pesa" = "Bulla Pesa",
                                  "Loodokilani" = "Iloodokilani",
                                  "Ewuaso Kidong" = "Ewuaso Oonkidong'I",
                                  "Mbirikani" = "Mbirikani/Eselenkei",
                                  "Oldonyonyokie" = "Keekonyokie",
                                  "KANYANGI" = "Kanyangi",
                                  "Chengoni-Samburu" = "Chengoni/Samburu",
                                  "Waa-Ng'ombeni" = "Waa",
                                  "Kasemeni" = "Kinango",
                                  "Mukogodo East" = "Mugogodo East",
                                  "Mukogodo West" = "Mugogodo West",
                                  "Baharini" = "Bahari",
                                  "NGUU/MASAMBA" = "Nguu/Masumba",
                                  "KASIKEU" = "Kasikeu",
                                  "MUKAA" = "Mukaa",
                                  "Simbir Fatuma" = "Shimbir Fatuma",
                                  "Banisa" = "Banissa",
                                  "Warankara" = "Waranqara",
                                  "Rhamu Dimtu" = "Rhamu-Dimtu",
                                  "Sagante" = "Sagante/Jaldesa",
                                  "Korr" = "Korr/Ngurunit",
                                  "Heilu Manyatta" = "Heillu/Manyatta",
                                  "Akithi" = "Akithii",
                                  "Amwitha" = "Amwathi",
                                  "Naroosura" = "Majimoto/Naroosura",
                                  "Thegu river" = "Thegu River",
                                  "Suguta Mar Mar" = "Suguta Marmar",
                                  "Wusi" = "Wusi/Kishamba",
                                  "Challa" = "Chala",
                                  "Wumingu" = "Wumingu/Kishushe",
                                  "Tunyai" = NA_character_,
                                  "Kamarandi" = NA_character_,
                                  "Kalemng'orok" = "Katilu",
                                  "Lokori/Kachodin" = "Lokiriama/Lorengippi",
                                  "Lorugum" = "Turkwel",
```

```
                                "Nachukui" = "Kalokol",
                                "Hadado" = "Hadado/Athibohol",
                                "Lagbogol" = "Lagboghol South",
                                "Barwaqo" = "Barwago",
                                "Dasheq" = "Tarbaj",
                                "Elnur" = "Elnur/Tula Tula",
                                "Riwo" = "Lokori/Kochodin",
                                "Masol" = "Masool")

# check if number of observations dropped = 0
df <- geocode_df(df)
```

```
# load spatial data
# - unzip if shape files are not unpacked
if (!file.exists("shapefiles/gadm36_KEN_1.shp")){
  unzip('shapefiles/gadm36_KEN_1.zip')
}
# - load Kenya GADM ADM-1 shape file
Kenya1 <- sf::st_read("shapefiles/gadm36_KEN_1.shp")
# - unzip if shape files are not unpacked
if (!file.exists("shapefiles/KEN_adm3-ndma.shp")){
  unzip("shapefiles/KEN_adm3-ndma.zip")
}
# - load Kenya NDMA ADM-3 shape file
Kenya3_ndma <- sf::st_read("shapefiles/KEN_adm3-ndma.shp")

# add spatial information
Kenya3_ndma <- merge(Kenya3_ndma, df[!duplicated(df$adm3),],
                     by.x = "NAME_3", by.y = "adm3")
Kenya1_ndma <- merge(Kenya1, df[!duplicated(df$adm1),],
                     by.x = "NAME_1", by.y = "adm1")

# list arid vs semi-arid counties
arid <- Kenya1[Kenya1$NAME_1 %in%
                 c("Baringo", "Garissa", "Isiolo", "Mandera",
                   "Marsabit", "Samburu", "Tana River",
                   "Turkana", "Wajir"),]

semi <- Kenya1[Kenya1$NAME_1 %in%
                 c("Embu", "Kajiado", "Kilifi", "Kitui", "Kwale",
                   "Laikipia", "Lamu", "Makueni", "Meru", "Narok",
                   "Nyeri", "Taita Taveta", "West Pokot"),]

# list arid vs semi-arid wards
arid.adm3 <- Kenya3_ndma[Kenya3_ndma$NAME_1 %in%
                           c("Baringo", "Garissa", "Isiolo", "Mandera",
                             "Marsabit", "Samburu", "Tana River",
                             "Turkana", "Wajir"),]

semi.adm3 <- Kenya3_ndma[Kenya3_ndma$NAME_1 %in%
                 c("Embu", "Kajiado", "Kilifi", "Kitui", "Kwale",
                   "Laikipia", "Lamu", "Makueni", "Meru", "Narok",
                   "Nyeri", "Taita Taveta", "West Pokot"),]
```

```r
# plot
fig1 = ggplot() +
  geom_sf(data = Kenya1,  inherit.aes = FALSE,
          fill = "white", color= "#5c666f",
          linewidth = .25, alpha = .001) +
  geom_sf(data = semi.adm3, inherit.aes = FALSE,
          fill = "#ff6161", color= "#5c666f",
          linewidth = .25, alpha= .65) +
  geom_sf(data = arid.adm3, inherit.aes = FALSE,
          fill = "#ffb3de", color= "#5c666f",
          linewidth = .25, alpha= .65) +
  geom_sf_text(data = arid, aes(label = NAME_1),
               check_overlap = TRUE, size = 3,
               face = "bold", color = "gray15") +
  geom_sf_text(data = semi, aes(label = NAME_1),
               check_overlap = TRUE, size = 3,
               face = "bold", color = "gray15") +
  theme_void() +
  theme(legend.position = "bottom",
        legend.box = "horizontal",
        legend.box.spacing = element_blank(),
        legend.text = element_text(size=14)) +
  coord_sf(xlim = c(32, 45), ylim = c(-6, 6)) +
  labs(x = "", y = "")

plot(fig1)
```

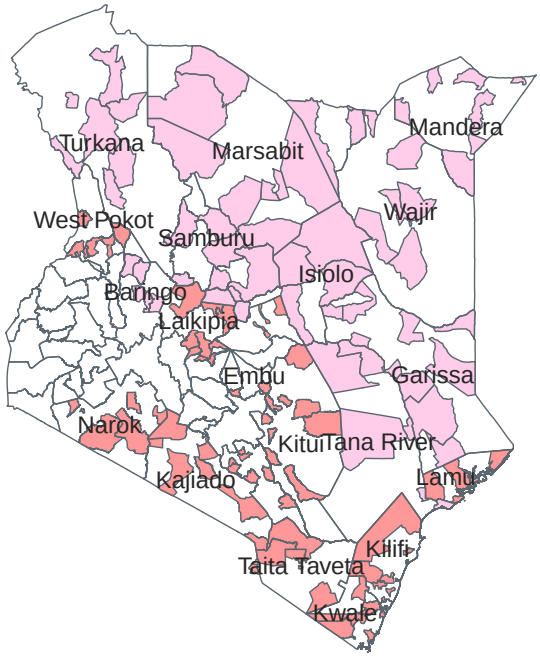


Figure 1: Map of 141 arid- and semi-arid study sites (Figure 1)

1.2 Ward resilience to acute malnutrition

We calculate ward resilience (*r_adm3*) on the basis of monthly MUAC information of individual children in 22 counties between 2016 and 2020.

```
# define MUAC threshold in mm (Risk of acute malnutrition, WHO)
THRESHOLD = 135

# calculate ward resilience
resilience_adm3 <- df
resilience_adm3 <- resilience_adm3 %>%
  group_by(adm3, date) %>%
  dplyr::mutate(id_r = cur_group_id())
excluded_hh = 0
dfs <- list()
for(i in 1:length(unique(resilience_adm3$id_r))) {

  # subset by adm3
  dat <- subset(resilience_adm3,
                resilience_adm3$id_r==
                  unique(resilience_adm3$id_r)[i])

  # calculate MUAC distance to threshold
  # - difference calculation is entry-wise
  above <- subset(dat, dat$MUAC > THRESHOLD)
  below <- subset(dat, dat$MUAC <= THRESHOLD)
  above$diff = above$MUAC - THRESHOLD
  below$diff = THRESHOLD - below$MUAC
  # construct joint data frame
  if(nrow(below) > 0 && nrow(above) > 0) {
    dat <- rbind(above, below)
  } else {
    if(nrow(above) > 0)
      dat <- above
    if(nrow(below) > 0)
      dat <- below
  }

  # - integrate/sum over the differences above & below threshold
  # - relative weight not up/down-ward biased due to value range!
  if (nrow(above)>0){
    upper = sum(above$diff)
  }else{
    upper = 0
  }
  if (nrow(below)>0){
    lower = sum(below$diff)
  }else{
    lower = 0
  }

  r = 1/2*((upper-lower)/(upper+lower)+1)

  # construct data frame
  dfs[[i]] <- data.frame(
```

```
    "adm1" = unique(dat$adm1),
    "adm3" = unique(dat$adm3),
    "adm3ID" = unique(dat$adm3ID),
    "date" = unique(dat$date),
    "r_adm3" = r,
    "GAM_adm1" = unique(dat$GAM_adm1),
    "n_obs_adm3" = nrow(dat),
    "n_obs_adm3_above" = nrow(above),
    "n_obs_adm3_below" = nrow(below))
}

resilience_adm3 <- do.call(rbind, dfs)

df <- data.frame(resilience_adm3)
```

```
# calculate malnutrition prevalence per adm3-month
df <- df %>%
  dplyr::group_by(adm3, date)%>%
  dplyr::mutate(adm3_prevalence = (n_obs_adm3_below/n_obs_adm3)*100)

# calculate mean resilience per adm3
adm3 <- df %>%
  dplyr::group_by(adm3)%>%
  dplyr::summarize(resilience.mean = mean(r_adm3, na.rm = TRUE),
                   prev.mean = mean(adm3_prevalence, na.rm = TRUE))

df <- merge(df, adm3, by = "adm3")

# re-code to categorical
df$resilience.cat <- as.factor(
  ifelse(df$resilience.mean < 0.25,
    1,
    ifelse(df$resilience.mean>= 0.25 & df$resilience.mean < 0.5,
      2,
      ifelse(df$resilience.mean >= 0.5 & df$resilience.mean < 0.75,
        3,
        4))))

# code ward malnutrition prevalence as IPC categories
df$GAM_adm1 <- df$GAM_adm1*100
df$ipc <- as.factor(
  ifelse(df$prev.mean < 5,
    1, # "Phase 1"
    ifelse(df$prev.mean >= 5 & df$prev.mean < 10,
      2, # "Phase 2"
      ifelse(df$prev.mean >= 10 & df$prev.mean < 15,
         3, # "Phase 3"
         ifelse(df$prev.mean >= 15 & df$prev.mean < 30,
            4, # "Phase 4"
            5))))) # Phase 5

df <- data.frame(df)

# add spatial information
```

```
Kenya3_ndma <- merge(Kenya3_ndma, df[!duplicated(df$adm3),],
                     by.x = "NAME_3", by.y = "adm3")
Kenya1_ndma <- merge(Kenya1, df[!duplicated(df$adm1),],
                     by.x = "NAME_1", by.y = "adm1")

# plot right side of figure 2

fig2.2 = ggplot() +
  geom_sf(data = Kenya1,  inherit.aes = FALSE, fill = "white",
          color= "#5c666f", linewidth = .25, alpha = .001) +
  geom_sf(data = Kenya3_ndma,
          aes(fill = resilience.cat), alpha= .65) +
  stat_sf_coordinates(data = Kenya3_ndma,
                      aes(color = ipc), size = 2.5, alpha = .9) +
  scale_fill_manual(name = "Resilience average",
                    values = c("#78e027",
                               "#12cc6a","#00917c"),
                    labels = c("0.25 - 0.49",
                               "0.50 - 0.74", "> 0.75")) +
  guides(colour=guide_legend(override.aes = list(size = 3),
                             title.position = "top", nrow = 1), size = FALSE) +
  scale_color_manual(labels = c("< 5.0", "5.0 - 9.9",
                                "10.0 - 14.9", "15.0 - 29.9",
                                "< 30.0"),
                     name = "Malnutrition prevalence (%)" ,
                     values = c("#6693f5", "#91AEEF",
                                "#a3a9fd","#6048c0",
                                "#491e96")) +
  theme_void() +
  theme(legend.position = "right",
        legend.box = "vertical",
        legend.box.spacing = element_blank(),
        legend.title = element_text(size=10, face = "bold"),
        legend.text = element_text(size=10)) +
  coord_sf(xlim = c(32, 45), ylim = c(-6, 6)) +
  labs(x = "", y = "")

plot(fig2.2)
```

1.3 External stressor intensity

We include information on climate, conflict and food price stressor intensity at the county-month-level to identify typical ward-resilience trajectories.

- `kenya_climate_monthadm1.RData`: The climate data are based on the "MOD13A3 v006" NDVI data available here. For each county-month in our sample, we calculate spatially averaged NDVI.
- `kenya_markets_monthadm1.RData`: Information regarding market price developments for 1kg maize are provided by the NDMA via expert interviews. We calculate monthly deviation rates from a three-year average and categorize each month as stressed or not, where stressor intensity is defined as the mean price increase above the long-term average.
- `kenya_conflict_monthadm1.RData`: Lastly, to account for the intensity of conflict, we draw on the Uppsala Conflict Data Program's Georeferenced Events Dataset (see Sundberg & Melander, 2013).

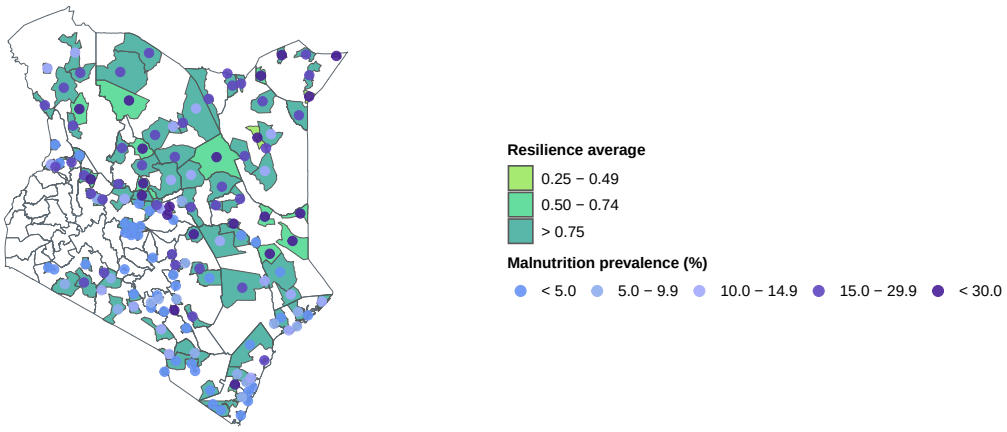


Figure 2: Average resilience for the period between 2016 and 2020 with smaller values indicating lower resilience. Blue dots display the mean proportion of malnourished children relative to the population size per ward for the same period(Figure 2, right)

```
# load data
load("data/kenya_markets_monthadm1.RData")
load("data/kenya_conflict_monthadm1.RData")
load("data/kenya_climate_monthadm1.RData")

# prepare for plotting
dfplot2 <- Reduce(function(x, y) merge(x, y,
                                       by=c("adm1", "year", "month"),
                                       all=TRUE),
                  list(conflict_month, climate_month, markets_month))
dfplot2$date <- as.Date(paste(dfplot2$year, dfplot2$month, "01", sep="-"))
dfplot2[is.na(dfplot2$stressor_conflict),]$stressor_conflict<-0
dfplot2[is.na(dfplot2$stressor_conflict_intensity),]$stressor_conflict_intensity<-0

dfplot2 <- dfplot2 %>%
  group_by(adm1, date) %>%
  mutate(stressors = sum(stressor_conflict_intensity,
                         stressor_markets_intensity,
                         stressor_climate_intensity))
dfplot2 <- subset(dfplot2, !is.na(dfplot2$stressors))

dfplot3 <- aggregate(cbind(stressor_climate_intensity,
                           stressor_conflict_intensity,
                           stressor_markets_intensity,
                           stressors) ~ date, data=dfplot2, FUN=mean)

dfplot3 <- dfplot3[c("date", "stressor_climate_intensity",
                     "stressor_conflict_intensity",
                     "stressor_markets_intensity",
                     "stressors")]
dfplot3 <- reshape2::melt(dfplot3, id.vars=c("date"),
                          measure.vars=c("stressor_climate_intensity",
                                         "stressor_conflict_intensity",
                                         "stressor_markets_intensity"))
```

```r
# plot bottom of figure 3

fig3.1 <- ggplot2::ggplot(dfplot2, aes(x=date, fill="adm")) +
  geom_area(data=dfplot3, aes(y=value, fill=variable), position = 'stack',
            color='black', size=0.2, alpha=0.8) +
 scale_x_date(breaks=c(seq.Date(min(dfplot3$date),
                                max(dfplot3$date),
                                by="quarter"), as.Date("2020-03-01")),
              labels=format(c(seq.Date(min(dfplot3$date),
                                       max(dfplot3$date),
                                       by="quarter"),
                              as.Date("2020-03-01")), "%b %Y"),
              expand = c(0, 0.02)) +
   scale_y_continuous(breaks = c(3*0.05,3*0.10,3*0.15,3*0.20,
                                 3*0.25,3*0.3,3*0.5,3*0.75,3),
                      limits = c(0, 3),
                      labels=c("5%", "10%", "15%", "20%", "25%",
                               "30%", "50%", "75%", "100%"),
                      expand = c(0, 0)) +
   scale_fill_manual(name="stressor",
                     breaks=c("stressor_climate_intensity",
                              "stressor_conflict_intensity",
                              "stressor_markets_intensity"),
                     labels=c("climate", "conflict", "markets"),
                     values=c("#fdebf3" ,"#ccff66", "#513223")) +
   theme(panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.background = element_blank(),
         axis.line =  element_line(colour = "black"),
         legend.position="right", legend.box = "vertical",
         legend.title = element_text(size=10),
         legend.text = element_text(size=10)) +
   labs(y="Stressor Intensity", x="Month")

plot(fig3.1)
```

2. Latent class mixed model (LCMM)

We fit a Latent Class Mixed Model (LCMM) to identify latent patterns in the observed ward resilience and use the fitted model to classify wards into groups with similar resilience trajectories.

We include below the code we used to identify the best model based on the smallest BIC, depending on alternative link functions and 1-10 classes. Note that as a function of the slightly different, anonymised input data, it's impossible to directly replicate the merger for the ward-level resilience trajectories and figures as reported in the article. To save computing time, we provide this information, i.e. the resilience trajectory class of each household (excluding ward-names) in a separate file called `household_all.csv`.

```r
# 1. explore alternative link functions

set.seed(14)

# beta
b1 <- lcmm(fixed=r_adm3~ time + stressor_climate_intensity
```

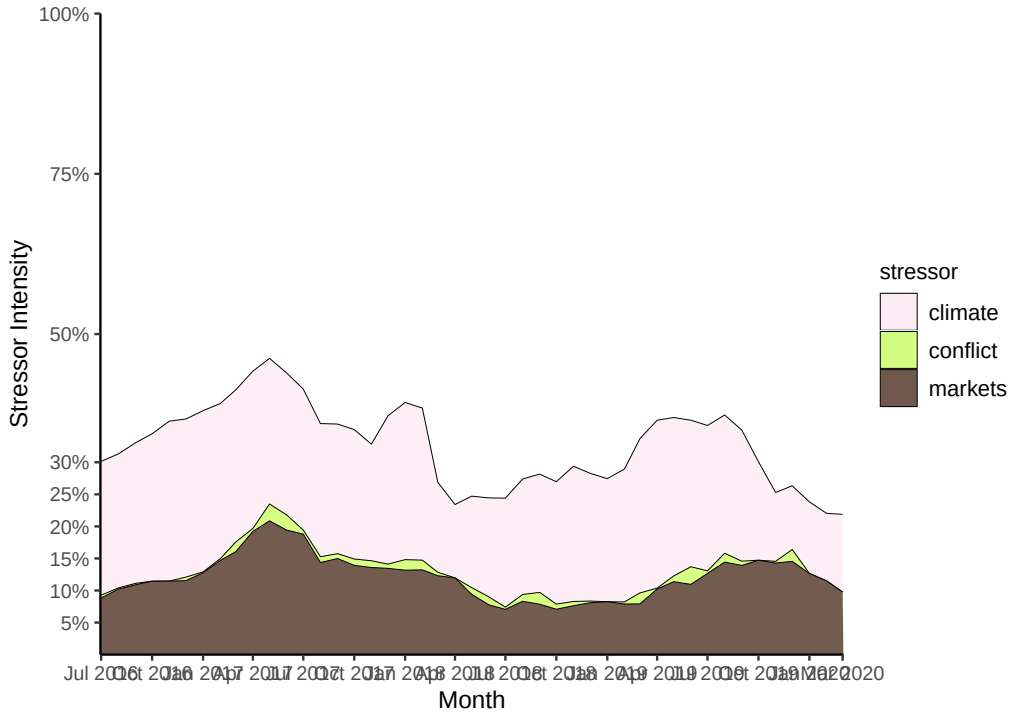


Figure 3: Kenya stressor intensity for the period between 2016 and 2020 (Figure 3, bottom)

```
           + stressor_conflict_intensity + stressor_markets_intensity
           ,random=~time, subject = 'adm3ID', link = "beta"
           ,ng = 1, data = df)
summary(b1)

# splines
s1 <- lcmm(fixed=r_adm3~ time + stressor_climate_intensity
           + stressor_conflict_intensity + stressor_markets_intensity
           ,random=~time, subject = 'adm3ID', link = "splines"
           ,ng = 1, data = df)
summary(s1)

# 2. explore alternative numbers of classes

# initial model with ng=1 for the random initial values
s1 <- lcmm(fixed=r_adm3~ time + stressor_climate_intensity
           + stressor_conflict_intensity + stressor_markets_intensity
           ,random=~time, subject = 'adm3ID', link = "splines"
           ,ng = 1, data = df)
summary(s1)

# model with ng=2
s2 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
                    ,random =~time
                    ,mixture=~time
                    ,subject = 'adm3ID', link = "splines"
                    ,ng = 2, data = df
```

```
                    ,B = s1
                    ,maxiter=100)
summary(s2)

# model with ng=3
s3 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 3, data = df
            ,B = s1
            ,maxiter=250)
summary(s3)

# model with ng=4
s4 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 4, data = df
            ,B = s1
            ,maxiter=250)
summary(s4)

# model with ng=5
s5 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 5, data = df
            ,B = s1
            ,maxiter=250)
summary(s5)

# model with ng=6
s6 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 6, data = df
            ,B = s1
            ,maxiter=250)
summary(s6)

# model with ng=7
s7 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
```

```
            ,subject = 'adm3ID', link = "splines"
            ,ng = 7, data = df
            ,B = s1
            ,maxiter=250)
summary(s7)

# model with ng=8
s8 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 8, data = df
            ,B = s1
            ,maxiter=250)
summary(s8)

# model with ng=9
s9 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
            + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 9, data = df
            ,B = s1
            ,maxiter=250)
summary(s9)

# model with ng=10
s10 <-  lcmm(fixed=r_adm3~ time + stressor_climate_intensity
             + stressor_conflict_intensity + stressor_markets_intensity
            ,random =~time
            ,mixture=~time
            ,subject = 'adm3ID', link = "splines"
            ,ng = 10, data = df
            ,B = s1
            ,maxiter=250)
summary(s10)

# compare model fit based on smallest BIC
fit <- summarytable(s1, s2, s3, s4, s5, s6, s7, s8, s9, s10)
```

Next, we use the model to categorize wards into one of the four classes in terms of their resilience trajectory.

```
# residuals
plot(s4)

# postprob plot
plot(s4,which="postprob")

# post prob's for 4 classes
postprob(s4)
round(summary(as.numeric(s4$pprob[s4$pprob[,"class"]==1,"prob1"])),2)
round(summary(as.numeric(s4$pprob[s4$pprob[,"class"]==2,"prob2"])),2)
```

```
round(summary(as.numeric(s4$pprob[s4$pprob[,"class"]==3,"prob3"])),2)
round(summary(as.numeric(s4$pprob[s4$pprob[,"class"]==4,"prob4"])),2)

# assign classes to observations
df$adm3ID <- as.character(df$adm3ID)
people4 <- as.data.frame(s4$pprob[,1:2])
people4$adm3ID <- as.character(people4$adm3ID)
df <- merge(df, people4, by = "adm3ID", all.x = TRUE)
```

```
# load household data
df <- read.csv("data/household_all.csv", stringsAsFactors = FALSE)

# add spatial information
Kenya3_ndma <- merge(Kenya3_ndma, df[!duplicated(df$adm3ID),],
                     by.x = "adm3ID.y", by.y = "adm3ID")

# plot left side of figure 2
legend_size <- c(1,2,3,4,5)

fig2.1 = ggplot() +
  geom_sf(data = Kenya1,  inherit.aes = FALSE, fill = "white",
          color= "#5c666f", linewidth = .25, alpha = .001) +
  geom_sf(data = Kenya3_ndma, aes(fill = class), alpha= .65) +
  stat_sf_coordinates(data = Kenya3_ndma, aes(color = ipc),
                      size = 2.5, alpha = .9) +
  scale_fill_manual(name = "Resilience trajectory",
                    values = c("#ffb3de" ,"#ff6699", "#ff6161","#de2414")) +
  guides(colour=guide_legend(override.aes = list(size = 5),
                             title.position = "top", nrow = 1), size = FALSE) +
  scale_color_manual(labels = c("< 5.0", "5.0 - 9.9", "10.0 - 14.9",
                                "15.0 - 29.9", "< 30.0"),
                     name = "Malnutrition prevalence (%)" ,
                     values = c("#6693f5", "#91AEEF","#a3a9fd",
                                "#6048c0", "#491e96")) +
  theme_void() +
  theme(legend.position = "right",
        legend.box = "vertical",
        legend.box.spacing = element_blank(),
        legend.title = element_text(size=10, face = "bold"),
        legend.text = element_text(size=10)) +
  coord_sf(xlim = c(32, 45), ylim = c(-6, 6)) +
  labs(x = "", y = "")

plot(fig2.1)
```

```
fig3_top <- ggplot(df, aes(date, r_adm3, group=adm3ID, color=class)) +
          geom_smooth(data=df,
                      aes(y=r_adm3, x=date, colour="gray93"),
                      method="auto", size=0.3, se=FALSE, alpha=0.1) +
          geom_smooth(data=df[grepl("chronic", df$class),],
                      aes(y=r_adm3, x=date, group=class,
                          colour=REDS[4]), method="auto",se=T,
                      size=1.2, alpha=0.1) +
          geom_smooth(data=df[grepl("decrease", df$class),],
```

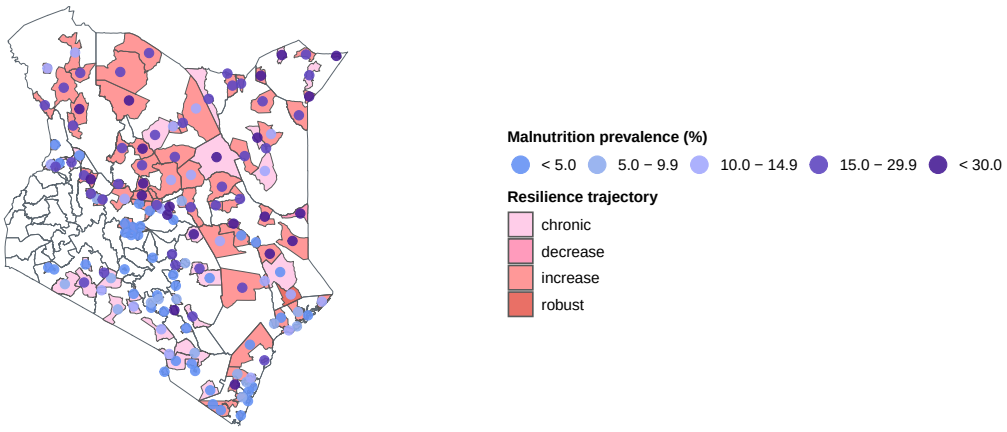


Figure 4: Ward resilience trajectory assigned using the latent-class mixed model. Blue dots display the mean proportion of malnourished children relative to the population size per ward for the same period (Figure 2, left)

```
                aes(y=r_adm3, x=date, group=class,
                    colour=REDS[2]), method="auto",se=T,
                size=1.2, alpha=0.1) +
  geom_smooth(data=df[grepl("increase", df$class),],
                aes(y=r_adm3, x=date, group=class,
                    colour=REDS[3]), method="auto", se=T,
                size=1.2, alpha=0.1) +
  geom_smooth(data=df[grepl("robust", df$class),],
                aes(y=r_adm3, x=date, group=class,
                    colour=REDS[1]), method="auto", se=T,
                size=1.2, alpha=0.1) +
  scale_colour_manual(name = 'Resilience Trajectory',
                values = c('gray93', REDS[1], REDS[2],
                           REDS[3], REDS[4]),
                breaks = c('gray93', REDS[1], REDS[2],
                           REDS[3], REDS[4]),
                labels = c('Ward', 'Chronic', 'Decrease',
                           'Increase', 'Robust')) +
scale_y_continuous(limits = c(0, 1.0),
                   breaks=c(0.00, 0.25, 0.5, 0.75, 1.0),
                   labels=c("0.00", "0.25", "0.5", "0.75", "1.00")) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        legend.position="bottom",
        legend.box = "horizontal",
        legend.title = element_text(size=10, face="bold"),
        legend.text = element_text(size=10),
        plot.title = element_text(hjust = 0.5, face="bold", size=10),
        axis.title.y = element_text(size=10,face="bold",
                                    margin=margin(r=15)),
        axis.title.x = element_text(size=10, face="bold",
                                    margin=margin(t=15))) +
```

```
          labs(y="Resilience", x="Year")

plot(fig3_top)
```

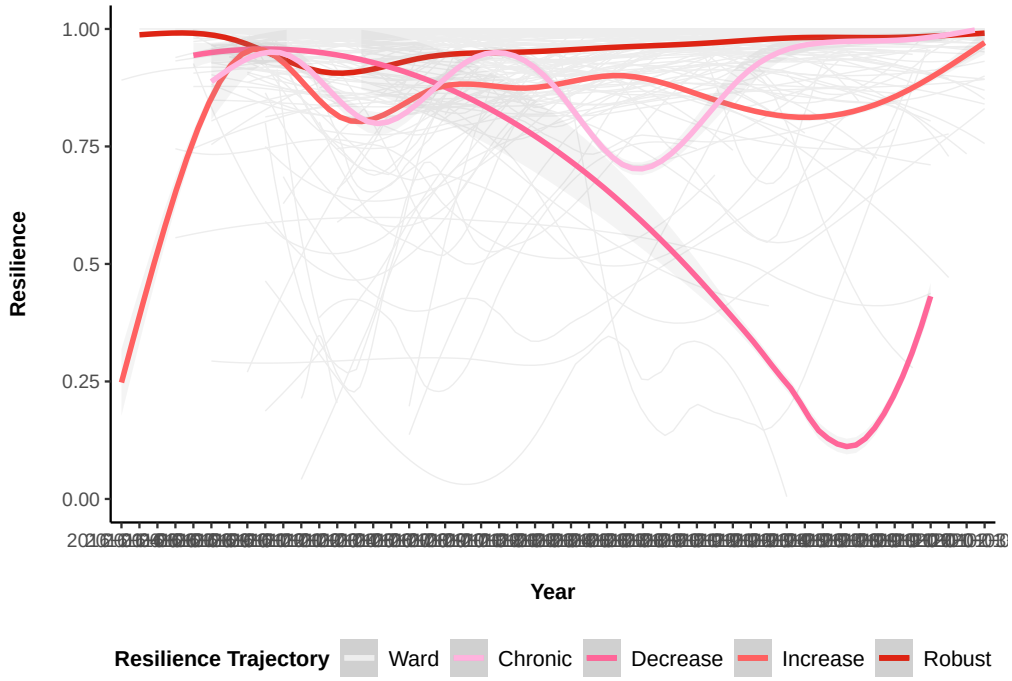


Figure 5: Observed trajectories of ward resilience to acute malnutrition at monthly intervals (Figure 3, top)

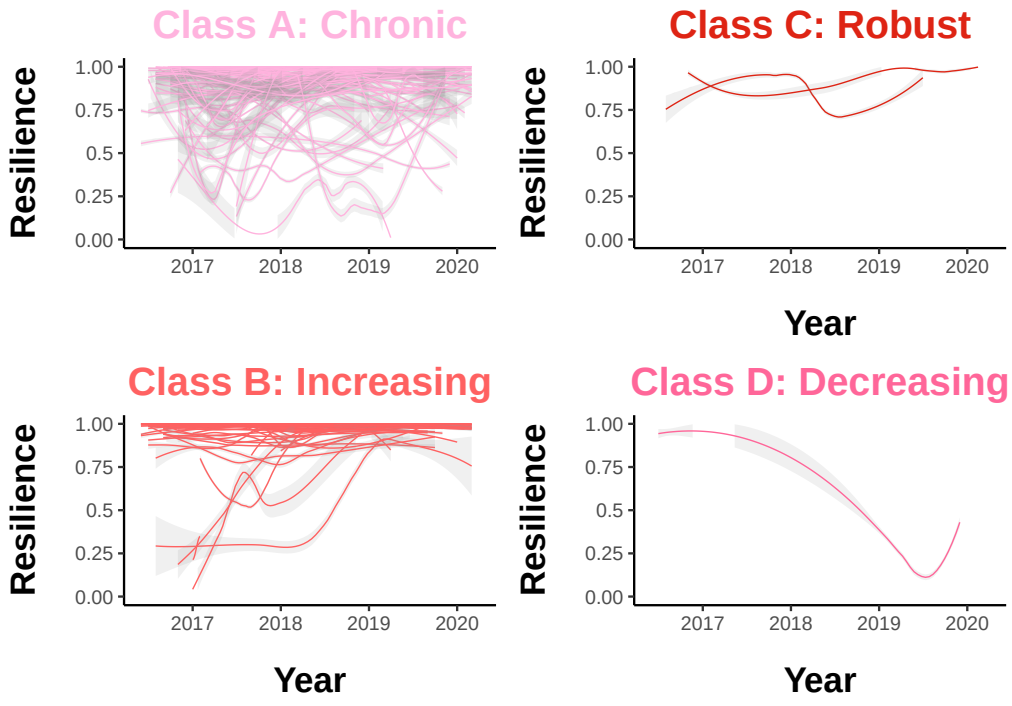


Figure 6: Resilience trajectories per assigned latent class for Kenyan wards (Figure 4)

3. Cox regression

Second, we use cox regression analysis to explore household risk depending on ward-resilience trajectory, while controlling for a minimal set of control variables.

The raw data used is `household_all.csv`.

To perform the cox regression analysis, we provide the following variables

- *AMN_dummy*: binary variable indicating that a household is malnourished, i.e. if min 1 observation is MUAC $<$ 125mm
- *class*: assigned ward-resilience trajectory (increase, decrease, chronic, robust)
- *time*: number of survey months
- *switch_below*: binary variable indicating that a household becomes malnourished at time t
- *switch_recover*: binary variable indicating that a household recovers from malnutrition at time t
- *status*: categorical variable indicating household switches from normal nutrition to undernourished and vice versa
- *tevent:* time in months until switch in status

Note that the above calculations are rather data-intensive and were processed separately. In addition, we include the following fixed household characteristics as background variables:

- *gender household head* (female, male)
- *education household head* (true, false)
- *livelihood* (pastoralist, other)
- *water source* (safe, unsafe)

`rules_coding_agg.txt` provides a summary of the coding and aggregation rules.

```
# load household data
df <- read.csv("data/household_all.csv", stringsAsFactors = TRUE)

# use chronic as a reference category
df$class = relevel(df$class, ref = "chronic")

# recode status
df$status2 <- as.integer(df$status)
summary(df$status2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   2.000   2.000   2.004   2.000   3.000   28928
```

```
# null model with time switch below nutrition threshold as DV
fit2 <- survfit(Surv(tevent, status2) ~ class, data=df)

break_values <- c(0, 0.25, 0.5, 0.75, 1)
summary(fit2)
```

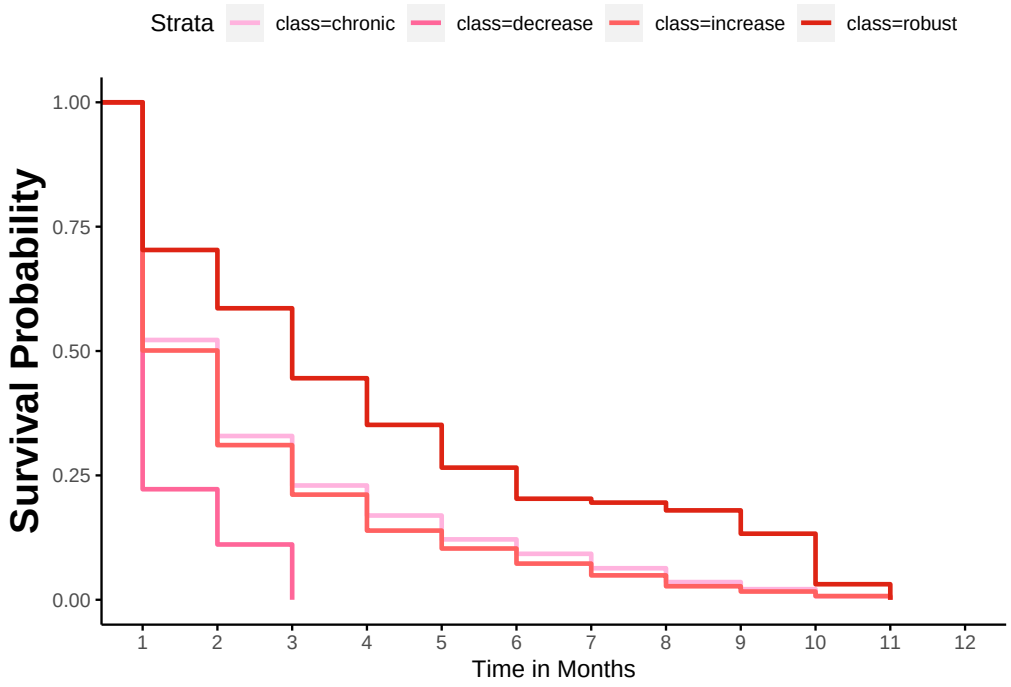
```
## Call: survfit(formula = Surv(tevent, status2) ~ class, data = df)
##
## 50179 observations deleted due to missingness
##                 class=chronic
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1   1689     807   0.5222 0.01215      0.49892       0.5466
```

```
##     2    882     326   0.3292 0.01143      0.30752       0.3524
##     3    556     168   0.2297 0.01024      0.21051       0.2507
##     4    388     102   0.1693 0.00913      0.15236       0.1882
##     5    286      81   0.1214 0.00795      0.10676       0.1380
##     6    205      49   0.0924 0.00705      0.07954       0.1073
##     7    156      49   0.0634 0.00593      0.05274       0.0761
##     8    107      47   0.0355 0.00450      0.02771       0.0455
##     9     60      24   0.0213 0.00351      0.01543       0.0294
##    10     36      24   0.0071 0.00204      0.00404       0.0125
##    11     12      12   0.0000     NaN           NA           NA
##
##                 class=decrease
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1      9       7    0.222   0.139       0.0655        0.754
##     2      2       1    0.111   0.105       0.0175        0.705
##     3      1       1    0.000     NaN           NA           NA
##
##                 class=increase
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1   2834    1414  0.50106 0.00939      0.48298       0.5198
##     2   1420     539  0.31087 0.00869      0.29429       0.3284
##     3    881     282  0.21136 0.00767      0.19685       0.2269
##     4    599     205  0.13903 0.00650      0.12685       0.1524
##     5    394     102  0.10303 0.00571      0.09243       0.1149
##     6    292      86  0.07269 0.00488      0.06373       0.0829
##     7    206      67  0.04905 0.00406      0.04171       0.0577
##     8    139      62  0.02717 0.00305      0.02180       0.0339
##     9     77      30  0.01658 0.00240      0.01249       0.0220
##    10     47      26  0.00741 0.00161      0.00484       0.0113
##    11     21      21  0.00000     NaN           NA           NA
##
##                 class=robust
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    128      38   0.7031  0.0404       0.6283        0.787
##     2     90      15   0.5859  0.0435       0.5065        0.678
##     3     75      18   0.4453  0.0439       0.3670        0.540
##     4     57      12   0.3516  0.0422       0.2779        0.445
##     5     45      11   0.2656  0.0390       0.1991        0.354
##     6     34       8   0.2031  0.0356       0.1441        0.286
##     7     26       1   0.1953  0.0350       0.1374        0.278
##     8     25       2   0.1797  0.0339       0.1241        0.260
##     9     23       6   0.1328  0.0300       0.0853        0.207
##    10     17      13   0.0312  0.0154       0.0119        0.082
##    11      4       4   0.0000     NaN           NA           NA
```

```r
# time-to-event-plot
cox.uni <- ggsurvplot(fit2, data = df,
                      risk.table = FALSE, font.y = c(18, "bold", "black"),
                      xlim = c(1,12),  xlab = "Time in Months", break.time.by = 1,
                      ylim = c(0,1),  ylab = "Survival Probability",
                      break.by = break_values,
                      censor.shape="|", censor.size = 1,
                      palette =c(REDS[1], REDS[2], REDS[3], REDS[4]),
                      ggtheme= theme(panel.grid.major = element_blank(),
```

```
                                    panel.grid.minor = element_blank(),
                                    panel.background = element_blank(),
                                    axis.line = element_line(colour = "black")))

cox.uni
```




```
# univariate regression results
res.cox <- coxph(Surv(tevent, status2) ~ class, data = df)
res.cox
```

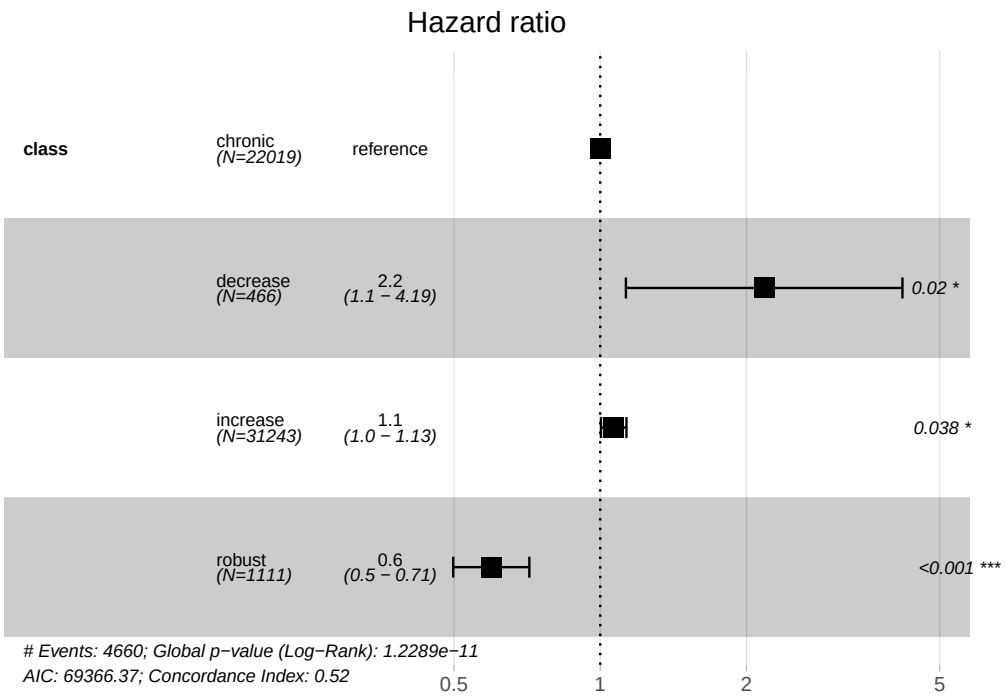
```
## Call:
## coxph(formula = Surv(tevent, status2) ~ class, data = df)
##
##                   coef exp(coef) se(coef)      z      p
## classdecrease  0.77710   2.17514  0.33440  2.324 0.0201
## classincrease  0.06389   1.06597  0.03076  2.077 0.0378
## classrobust   -0.51644   0.59664  0.09202 -5.612  2e-08
##
## Likelihood ratio test=53.81  on 3 df, p=1.229e-11
## n= 4660, number of events= 4660
##    (50179 observations deleted due to missingness)
```

```
summary(res.cox)
```

```
## Call:
## coxph(formula = Surv(tevent, status2) ~ class, data = df)
##
##   n= 4660, number of events= 4660
##    (50179 observations deleted due to missingness)
##
##                   coef exp(coef) se(coef)      z Pr(>|z|)
## classdecrease  0.77710   2.17514  0.33440  2.324   0.0201 *
## classincrease  0.06389   1.06597  0.03076  2.077   0.0378 *
```

```
## classrobust   -0.51644   0.59664  0.09202 -5.612    2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               exp(coef) exp(-coef) lower .95 upper .95
## classdecrease    2.1751     0.4597    1.1294    4.1892
## classincrease    1.0660     0.9381    1.0036    1.1322
## classrobust      0.5966     1.6760    0.4982    0.7146
##
## Concordance= 0.522  (se = 0.006 )
## Likelihood ratio test= 53.81  on 3 df,   p=1e-11
## Wald test            = 47.47  on 3 df,   p=3e-10
## Score (logrank) test = 48.73  on 3 df,   p=1e-10
```

```
cox.uni.plot <- ggforest(res.cox, data = df)
cox.uni.plot
```




```
# rename
df <- df %>%
  dplyr::rename("resilience trajectory" = "class",
                "water source" = "water_source")

# multivariate regression
res.cox2 <- coxph(Surv(tevent, status2) ~ `resilience trajectory` + livelihood
                  +  gender + education + `water source` , data = df)
res.cox2
```

```
## Call:
## coxph(formula = Surv(tevent, status2) ~ `resilience trajectory` +
##     livelihood + gender + education + `water source`, data = df)
##
##                                    coef exp(coef) se(coef)      z        p
## `resilience trajectory`decrease  1.11416   3.04699  0.50297  2.215  0.02675
```

```
## `resilience trajectory`increase  0.07951   1.08275  0.03934  2.021  0.04327
## `resilience trajectory`robust   -0.86627   0.42052  0.12812 -6.761 1.37e-11
## livelihoodpastoralist            0.20345   1.22562  0.04452  4.569 4.89e-06
## gendermale                       0.02182   1.02206  0.04302  0.507  0.61201
## educationyes                    -0.12359   0.88374  0.04180 -2.957  0.00311
## `water source`unsafe            -0.05564   0.94588  0.04161 -1.337  0.18115
##
## Likelihood ratio test=118.1  on 7 df, p=< 2.2e-16
## n= 3158, number of events= 3158
##    (51681 observations deleted due to missingness)
```

```
summary(res.cox2)
```

```
## Call:
## coxph(formula = Surv(tevent, status2) ~ `resilience trajectory` +
##     livelihood + gender + education + `water source`, data = df)
##
##   n= 3158, number of events= 3158
##    (51681 observations deleted due to missingness)
##
##                                    coef exp(coef) se(coef)      z Pr(>|z|)
## `resilience trajectory`decrease  1.11416   3.04699  0.50297  2.215  0.02675 *
## `resilience trajectory`increase  0.07951   1.08275  0.03934  2.021  0.04327 *
## `resilience trajectory`robust   -0.86627   0.42052  0.12812 -6.761 1.37e-11 ***
## livelihoodpastoralist            0.20345   1.22562  0.04452  4.569 4.89e-06 ***
## gendermale                       0.02182   1.02206  0.04302  0.507  0.61201
## educationyes                    -0.12359   0.88374  0.04180 -2.957  0.00311 **
## `water source`unsafe            -0.05564   0.94588  0.04161 -1.337  0.18115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                 exp(coef) exp(-coef) lower .95 upper .95
## `resilience trajectory`decrease    3.0470     0.3282    1.1370    8.1658
## `resilience trajectory`increase    1.0828     0.9236    1.0024    1.1695
## `resilience trajectory`robust      0.4205     2.3780    0.3271    0.5406
## livelihoodpastoralist              1.2256     0.8159    1.1232    1.3374
## gendermale                         1.0221     0.9784    0.9394    1.1120
## educationyes                       0.8837     1.1316    0.8142    0.9592
## `water source`unsafe               0.9459     1.0572    0.8718    1.0263
##
## Concordance= 0.569  (se = 0.008 )
## Likelihood ratio test= 118.1  on 7 df,   p=<2e-16
## Wald test            = 105.3  on 7 df,   p=<2e-16
## Score (logrank) test = 109.3  on 7 df,   p=<2e-16
```

```
fig5 <- ggforest(res.cox2, data = df, fontsize = 1)
plot(fig5)
```

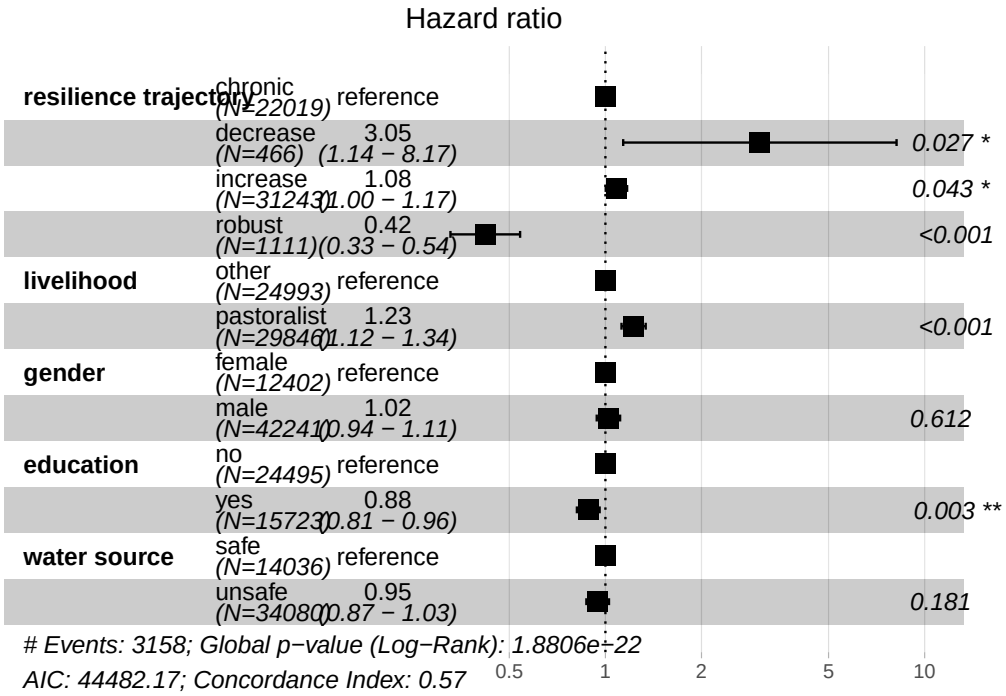


Figure 7: Multivariate Cox regression results for household risk (Figure 5).