Development Economics: Project Data Preparation

Contents

Data

Loading Drought Data

```
# Load SPEI index data and drought dummies
droughtdata1 = read.csv("drought_data_output/spei3_yearly_master.csv")
colnames(droughtdata1)[which(names(droughtdata1) == "avg_spei3")] <- "spei"
droughtdata1$spei = droughtdata1$spei * (-1)
colnames(droughtdata1)[which(names(droughtdata1) == "consec_drought")] <- "drought_dummy"
colnames(droughtdata1)[which(names(droughtdata1) == "total_droughts")] <- "drought_dummy2"

# Load VIH stress data
droughtdata2 = read.csv("drought_data_output/agri_stress_long.csv")
colnames(droughtdata2)[which(names(droughtdata2) == "PROVINCE")] <- "district"
colnames(droughtdata2)[which(names(droughtdata2) == "YEAR")] <-"year"
colnames(droughtdata2)[which(names(droughtdata2) == "DATA")] <- "agric_stress"

droughtdata = left_join(droughtdata1, droughtdata2, by=c("year","district"))

# Load temperature data (world bank)
tempdata = read_xlsx("rain_data/cru-x0.5_timeseries_tas_timeseries_annual_1901-2024_mean_historical_cru
colnames(tempdata)[which(names(tempdata) == "name")] <- "district"

# Load raw agriculture data
raw_agri_data = as.data.frame(read_xlsx("agriculture data/agri.xlsx"))
```

Load Household Data

```
# Load Household Survey Data
household_data <- list()
years_to_load <- 2004:2023
for (year in years_to_load) {
  if (year <= 2014) {
    file_extension <- ".xls"
    read_function <- readxl::read_xls} else {
    file_extension <- ".xlsx"
    read_function <- readxl::read_xlsx}
  file_path <- file.path("household_data", paste0("Household-", year, file_extension))
  if (file.exists(file_path)) {
    loaded_data <- try(read_function(file_path), silent = TRUE)
    if (!inherits(loaded_data, "try-error")) {
      household_data[[as.character(year)]] <- loaded_data
      cat("Successfully loaded:", file_path, "\n")} else {
      warning(paste("Could not read", file_path, "- skipping. Error:", as.character(loaded_data)))
      }} else {
    warning(paste("File not found:", file_path, "- skipping."))}}
```

```
## Successfully loaded: household_data/Household-2004.xls
## Successfully loaded: household_data/Household-2005.xls
## Successfully loaded: household_data/Household-2006.xls
```

```
## Successfully loaded: household_data/Household-2007.xls
## Successfully loaded: household_data/Household-2008.xls
## Successfully loaded: household_data/Household-2009.xls
## Successfully loaded: household_data/Household-2010.xls
## Successfully loaded: household_data/Household-2011.xls
## Successfully loaded: household_data/Household-2012.xls
## Successfully loaded: household_data/Household-2013.xls
## Successfully loaded: household_data/Household-2014.xls
## Successfully loaded: household_data/Household-2015.xlsx
## Successfully loaded: household_data/Household-2016.xlsx
## Successfully loaded: household_data/Household-2017.xlsx
## Successfully loaded: household_data/Household-2018.xlsx
## Successfully loaded: household_data/Household-2019.xlsx
## Successfully loaded: household_data/Household-2020.xlsx
## Successfully loaded: household_data/Household-2021.xlsx
## Successfully loaded: household_data/Household-2022.xlsx
## Successfully loaded: household_data/Household-2023.xlsx
```

```
cat("\nProcess complete. All available Excel datasets have been loaded into the 'household_data' list.\n
```

```
##
## Process complete. All available Excel datasets have been loaded into the 'household_data' list.
```

```
rm(loaded_data)
```

Dealing with Names

```
# Raw (Household) Data List
raw_data_list = household_data

# Variables change names through the years, so we must go through the excel sheets
#  and find all the different types of names variables have
name_map <- list(
  district = c("hh_02", "MARZ", "marz"),
  hh_size = c("members", "MEMBERS"),
  urban = c("settlement", "SETTLEMENT", "SETTLEME"),
  poverty = c("poverty", "POVERTY", "pov", "POV"),
  exp = c("expend", "EXPEND"),
  income = c("totincome", "TOTINCOME", "TOTINCOM"),
  fdcons = c("fdcons", "FDCONS"),
  fdpurch = c("fdpurch", "FDPURCH"))

# There is an overlap of the variable name for agricultural income
# In some years, the variable is called y1_3drm.10, but in others it has a
# different name. And in some, y1_3drm.10 exists but describes another variable.
# We thus need to change the name of those variables to select precisely agric
for (year in names(household_data)) {
  year_num <- as.integer(year)
  if (year_num >= 2015 & year_num <= 2018) {
    if (year_num == 2015) {
      household_data[[year]] <- household_data[[year]] %>%
```

```
        rename_with(~"agric_income_temp", .cols = matches("y1_3drm.13"))
    } else if (year_num %in% c(2016, 2017, 2018)) {
      household_data[[year]] <- household_data[[year]] %>%
        rename_with(~"agric_income_temp", .cols = matches("y1_3drm.3"))}
    household_data[[year]] <- household_data[[year]] %>%
      rename_with(~"income_child_ignore", .cols = matches("y1_3drm.10"))}}

name_map$agric_income <- c("y1_3amd.3.00", "y1_3drm.10", "agric_income_temp")

# Fix the different names and set common ones established above
cleaned_data_list <- list()
final_columns <- names(name_map)
for (year in names(household_data)) {
  yearly_data <- household_data[[year]]
  current_names <- names(yearly_data)
  for (standard_name in names(name_map)) {
    possible_old_names <- name_map[[standard_name]]
    name_to_replace <- intersect(possible_old_names, current_names)
    if (length(name_to_replace) > 0) {
      yearly_data <- rename(yearly_data, !!standard_name := all_of(name_to_replace))}}
    yearly_data <- yearly_data %>%
    mutate(year = as.integer(year)) %>%
    select(year, any_of(final_columns))
    cleaned_data_list[[year]] <- yearly_data}

# Merge the different years
hh_dataset <- bind_rows(cleaned_data_list)

# Rename the districts according to their codes and set common ones
hh_dataset <- hh_dataset %>%
  mutate(district = case_when(
    district == "Yerevan"  ~ "Yerevan",
    district == "Aragatsotn"  ~ "Aragatsotn",
    district == "Ararat"  ~ "Ararat",
    district == "Armavir"  ~ "Armavir",
    district == "Gegharkunik"  ~ "Gegharkunik",
    district == "Lor"  ~ "Lor",
    district == "Kotayk"  ~ "Kotayk",
    district == "Shirak"  ~ "Shirak",
    district == "Syunik"  ~ "Syunik",
    district == "Vayots dzor" ~ "Vayots dzor",
    district == "Tavush" ~ "Tavush",
    district == 1  ~ "Yerevan",
    district == 2  ~ "Aragatsotn",
    district == 3  ~ "Ararat",
    district == 4  ~ "Armavir",
    district == 5  ~ "Gegharkunik",
    district == 6  ~ "Lor",
    district == 7  ~ "Kotayk",
    district == 8  ~ "Shirak",
    district == 9  ~ "Syunik",
    district == 10 ~ "Vayots dzor",
    district == 11 ~ "Tavush",
```

```
    district == "TAVUSH" ~ "Tavush",
    district == "YEREVAN" ~ "Yerevan",
    district == "ARAGATSOT" ~ "Aragtsotn",
    district == "ARARAT" ~ "Ararat",
    district == "ARMAVIR" ~ "Armavir",
    district == "GEGHARKUNIK" ~ "Gegharkunik",
    district == "KOTAYK" ~ "Kotayk",
    district == "SHIRAK" ~ "Shirak",
    district == "SYUNIK" ~ "Syunik",
    district == "VAYOTS DZOR" ~ "Vayots dzor",
    district == "LORI" ~ "Lor",
    district == "Lori" ~ "Lor",
    district == "rural" ~ "Armavir",
    district == "Sjunik" ~ "Syunik",
    district == "Vayots Dzor" ~ "Vayots dzor",
    district == "other urban" ~ "Yerevan",
    TRUE ~ NA_character_ ))

# Change poverty level for a dummy variable
hh_dataset <- hh_dataset %>%
  mutate(poverty = case_when(
    poverty == 1  ~ 0,
    poverty == 2  ~ 1,
    poverty == 3  ~ 1,
    TRUE ~ NA_real_))

# Change settlement values for a dummy variable (urbanization)
hh_dataset <- hh_dataset %>%
  mutate(urban = case_when(
    year <= 2017 & urban == 0 ~ 1,
    year <= 2017 & urban == 1 ~ 1,
    year <= 2017 & urban == 2 ~ 0,
    year %in% c(2018,2020) & urban == 1 ~ 1,
    year %in% c(2018,2020) & urban == 2 ~ 1,
    year %in% c(2018,2020) & urban == 3 ~ 0,
    year %in% c(2019,2021) & urban == 1 ~ 1,
    year %in% c(2019,2021) & urban == 2 ~ 0,
    year %in% c(2019,2021) & urban == 1 ~ 1,
    year >= 2022 & urban == 1 ~ 1,
    year >= 2022 & urban == 2 ~ 0,
    year >= 2022 & urban == -999999999 ~ NA_real_,
    TRUE ~ NA_real_ ))
```

```
# Harmonize the dataset by linking year, district and agricultural variables
colnames(raw_agri_data) <- raw_agri_data[3, ]
districts_row <- as.character(raw_agri_data[2, ])

# Associate agricultural variables with their districts (with a forward fill)
districts_filled <- fill(data.frame(district = districts_row), district, .direction = "down")$district
raw_agri_data[2, ] <- districts_filled
raw_agri_data <- raw_agri_data[-c(1, 3, 22:40), ] # Cut empty row

# Extract district names
```

```r
districts_row <- as.character(raw_agri_data[1, -1])
data_only <- raw_agri_data[-1, ]
clean_colnames <- gsub("\\.\\d+$", "", colnames(data_only)) # harmonize variables names

# Create a new dataset for row binding the observation in an appropriate way
stacked_data <- data.frame()

for (i in 2:ncol(data_only)) {
  temp_df <- data.frame(
    district = districts_row[i-1],
    year = data_only[[1]],
    variable = clean_colnames[i],
    value = as.numeric(data_only[[i]]),
    stringsAsFactors = FALSE
  )
  stacked_data <- rbind(stacked_data, temp_df)
}

# Pivot in order to put variables as a columns
agric_data <- stacked_data %>%
  pivot_wider(
    names_from = variable,
    values_from = value
  ) %>%
  arrange(district, year)

# Harmonize district names with other dataset
agric_data <- agric_data %>%
  mutate(district = case_when(
    district == "Yerevan City"  ~ "Yerevan",
    district == "Aragatsotn Marz"  ~ "Aragatsotn",
    district == "Ararat Marz"  ~ "Ararat",
    district == "Armavir Marz"  ~ "Armavir",
    district == "Gegharkunik Marz"  ~ "Gegharkunik",
    district == "Lori Marz"  ~ "Lor",
    district == "Kotayk Marz"  ~ "Kotayk",
    district == "Shirak Marz"  ~ "Shirak",
    district == "Syunik Marz"  ~ "Syunik",
    district == "Vayots Dzor Marz" ~ "Vayots dzor",
    district == "Tavush Marz" ~ "Tavush"))

#Withdraw some variables
agric_data = agric_data %>%
  select(-c(12:17)) %>%
  slice(-c(109:126))

# Replace ".." and "-" observations by NA
agric_data[agric_data == ".."] <- NA
agric_data[agric_data == "-"] <- NA

# For Temperature data
tempdata = subset(tempdata, select = -code )
tempdata_long <- tempdata %>%
```

```
    pivot_longer(
    cols = -district,
    names_to = "year",
    values_to = "temperature"  ) %>%
  mutate(district = recode(district,
      "Gergharkunik" = "Gegharkunik",
      "Lori"         = "Lor",
      "Vayots Dzor"  = "Vayots dzor"),
      year = str_extract(year, "\\d{4}"))
tempdata_long = subset(tempdata_long, year >= 2000)
```

Merging and Aggregating

```
# Household-level dataset is ready
hh_dataset = hh_dataset

# District-level dataset by aggregating HH data
dataset <- hh_dataset %>%
  group_by(district, year) %>%
  summarise(income = mean(income), poverty = mean(poverty),
            exp = mean(exp), fdcons = mean(fdcons), fdpurch = mean(fdpurch),
              agric_income = mean(agric_income, na.rm=T), urban = mean(urban), n_households = n())

# Split income into deciles, i.e. assign each HH to an income decile with dummies
dataset_deciles <- hh_dataset %>%
  group_by(year) %>%
  mutate(national_decile = ntile(income, 10)) %>%
  ungroup() %>%
  group_by(district, year, national_decile) %>%
    summarise(income = mean(income), poverty = mean(poverty),
              exp = mean(exp), fdcons = mean(fdcons), fdpurch = mean(fdpurch),
              agric_income = mean(agric_income, na.rm=T), urban = mean(urban), n_households = n())

# Split income into deciles, i.e. assign each HH to an income decile with dummies
dataset_quartiles <- hh_dataset %>%
  group_by(year) %>%
  mutate(national_quartile = ntile(income, 4)) %>%
  ungroup() %>%
  group_by(district, year, national_quartile) %>%
    summarise(income = mean(income), poverty = mean(poverty),
              exp = mean(exp), fdcons = mean(fdcons), fdpurch = mean(fdpurch),
              agric_income = mean(agric_income, na.rm=T), urban = mean(urban), n_households = n())


# Add drought data
dataset <- left_join(dataset, droughtdata, by = c("year", "district"))

dataset_deciles <- left_join(dataset_deciles, droughtdata, by = c("year", "district"))

dataset_quartiles <- left_join(dataset_quartiles, droughtdata, by = c("year", "district"))
```

```r
hh_dataset <- left_join(hh_dataset, droughtdata, by = c("year", "district"))

# Add agriculture data
agric_data$year = as.integer(agric_data$year)
dataset = left_join(dataset, agric_data, by = c("year", "district"))

# Add temperature data
tempdata_long$year = as.integer(tempdata_long$year)
dataset = left_join(dataset, tempdata_long, by = c("year", "district"))

# Clean useless datasets
rm(cleaned_data_list, household_data, name_map, yearly_data, droughtdata, raw_agri_data, stacked_data,

# Replace Yerevan NA with 0
dataset <- dataset %>% mutate(agric_income = case_when(district == "Yerevan" & year == "2016" & is.na(ag

# Change long names
colnames(dataset)[colnames(dataset) == "Gross  agricultural  output, total"] <- 'agric_output'
colnames(dataset)[colnames(dataset) == "Gross harvest of grains and leguminous plants"] <- 'grains_harv
colnames(dataset)[colnames(dataset) == "Gross harvest of  potatoes"] <- 'potatoes_harvest'
colnames(dataset)[colnames(dataset) == "Gross harvest of fruits and berries"] <- 'fruits_harvest'
colnames(dataset)[colnames(dataset) == "Gross harvest of water-melons"] <- 'watermelon_harvest'
colnames(dataset)[colnames(dataset) == "Gross harvest of grape"] <- 'grapes_harvest'
colnames(dataset)[colnames(dataset) == "Gross harvest of vegetables"] <- 'vegetables_harvest'
colnames(dataset)[colnames(dataset) == "Sown areas under vegetables"] <- 'vegetables_area'
colnames(dataset)[colnames(dataset) == "Sown areas under potatoes"] <- 'potatoes_area'
colnames(dataset)[colnames(dataset) == "Sown areas under water-melons"] <- 'watermelon_area'
colnames(dataset)[colnames(dataset) == "Sown areas under grains and leguminous plants"] <- 'grains_area
colnames(dataset)[colnames(dataset) == "Planting areas of fruits and berries"] <- 'fruits_area'
colnames(dataset)[colnames(dataset) == "Planting areas of grape"] <- 'grapes_area'

# Adjust unit (before: 1 = 1000 tons -> after: 1 = 1 ton)
dataset$agric_output = dataset$agric_output * 1000
dataset$potatoes_harvest = dataset$potatoes_harvest * 1000
dataset$fruits_harvest = dataset$fruits_harvest * 1000
dataset$grains_harvest = dataset$grains_harvest * 1000
dataset$vegetables_harvest = dataset$vegetables_harvest * 1000
dataset$watermelon_harvest = dataset$watermelon_harvest * 1000
dataset$grapes_harvest = dataset$grapes_harvest * 1000

# Adjust agric_stress unit (is in %)
dataset$agric_stress = dataset$agric_stress / 100

# Add all crop-related agricultural output variables
dataset$crops_output = dataset$potatoes_harvest + dataset$fruits_harvest + dataset$grains_harvest + data

# Add all crop-related agricultural "agricultural land" variables
dataset$crops_land = dataset$potatoes_area + dataset$grains_area + dataset$vegetables_area

# Get harvest per area

# 1. Crop mapping: harvest column -> area column
crops <- list(
```

```
  melon = c("watermelon_harvest", "watermelon_area"),
  grape = c("grapes_harvest", "grapes_area"),
  vegetables = c("vegetables_harvest", "vegetables_area"),
  fruits = c("fruits_harvest", "fruits_area"),
  grains = c("grains_harvest", "grains_area"),
  potatoes = c("potatoes_harvest", "potatoes_area"))

# 2. Create output per field
for(crop in names(crops)){
  harvest_col <- crops[[crop]][1]
  area_col <- crops[[crop]][2]
  dataset <- dataset %>%
    mutate(
      !!paste0("output_per_field_", crop) := ifelse(.data[[area_col]] > 0,
                .data[[harvest_col]] / .data[[area_col]], NA)) }
```

Additional data

```
marz_rename_map <- c(
  "Yerevan city" = "Yerevan",
  "Lory Marz" = "Lor",
  "Kotayk Marz" = "Kotayk",
  "Shirak Marz" = "Shirak",
  "Tavush Marz" = "Tavush",
  "Vayots Dzor Marz" = "Vayots dzor",
  "Syunik Marz" = "Syunik",
  "Gegharkunuk Marz" = "Gegharkunik",
  "Ararat Marz" = "Ararat",
  "Armavir Marz" = "Armavir",
  "Aragatsotn Marz" = "Aragatsotn")

watersupply <- read_excel("drought_data_output/watersupply.xlsx", skip = 2, col_names = FALSE)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
## * `` -> `...15`
## * `` -> `...16`
```

```
new_headers <- as.character(watersupply[1, ])
new_headers[1] <- "Marz_Full"
new_headers[2] <- "Junk_Column"
colnames(watersupply) <- new_headers
watersupply <- watersupply[3:13, ]
watersupply = watersupply[,-2]

watersupply <- watersupply %>%
  mutate(Marz = marz_rename_map[Marz_Full]) %>%
  pivot_longer(
    cols = `2010`:`2023`,
    names_to = "Year",
    values_to = "WaterSupply") %>%
  select(Marz, Year, WaterSupply) %>%
  mutate(Year = as.integer(Year))

watersupply$WaterSupply = as.numeric(watersupply$WaterSupply)
names(watersupply)[names(watersupply) == 'Year'] <- 'year'
names(watersupply)[names(watersupply) == 'Marz'] <- 'district'

dataset <- left_join(dataset, watersupply, by = c("year", "district"))
```

Treatment Lags

```
#building lags
dataset <- dataset %>%
    arrange(district, year) %>%
    group_by(district) %>%
  mutate(
    drought_dummy_lag1 = lag(drought_dummy, n = 1, default = 0),
    drought_dummy_lag2 = lag(drought_dummy, n = 2, default = 0),
    drought_dummy2_lag1 = lag(drought_dummy2, n = 1, default = 0),
    drought_dummy2_lag2 = lag(drought_dummy2, n = 2, default = 0),
    spei_lag1 = lag(spei, n = 1, default = 0),
    spei_lag2 = lag(spei, n = 2, default = 0),
    share_lag1 = lag(share, n = 1, default = 0),
    share_lag2 = lag(share, n = 2, default = 0),
    agric_stress_lag1 = lag(agric_stress, n = 1, default = 0),
    agric_stress_lag2 = lag(agric_stress, n = 2, default = 0),
    drought_dummy_lag3 = lag(drought_dummy, n = 3, default = 0),
    drought_dummy_lag4 = lag(drought_dummy, n = 4, default = 0),
    drought_dummy2_lag3 = lag(drought_dummy2, n = 3, default = 0),
    drought_dummy2_lag4 = lag(drought_dummy2, n = 4, default = 0),
    spei_lag3 = lag(spei, n = 3, default = 0),
    spei_lag4 = lag(spei, n = 4, default = 0),
    share_lag3 = lag(share, n = 3, default = 0),
    share_lag4 = lag(share, n = 4, default = 0),
    agric_stress_lag3 = lag(agric_stress, n = 3, default = 0),
    agric_stress_lag4 = lag(agric_stress, n = 4, default = 0),
    drought_dummy_lag5 = lag(drought_dummy, n = 5, default = 0),
    drought_dummy_lag6 = lag(drought_dummy, n = 6, default = 0),
```

```
    drought_dummy2_lag5 = lag(drought_dummy2, n = 5, default = 0),
    drought_dummy2_lag6 = lag(drought_dummy2, n = 6, default = 0),
    spei_lag5 = lag(spei, n = 5, default = 0),
    spei_lag6 = lag(spei, n = 6, default = 0),
    share_lag5 = lag(share, n = 5, default = 0),
    share_lag6 = lag(share, n = 6, default = 0),
    agric_stress_lag5 = lag(agric_stress, n = 5, default = 0),
    agric_stress_lag6 = lag(agric_stress, n = 6, default = 0),
    temperature_lag1 = lag(temperature, n = 1, default = 0),
    temperature_lag2 = lag(temperature, n = 2, default = 0),
    temperature_lag3 = lag(temperature, n = 3, default = 0),
    temperature_lag4 = lag(temperature, n = 4, default = 0),
    temperature_lag5 = lag(temperature, n = 5, default = 0),
    temperature_lag6 = lag(temperature, n = 6, default = 0)) %>%
    ungroup()

dataset_deciles <- dataset_deciles %>%
    arrange(district, year) %>%
    group_by(district) %>%
  mutate(
    drought_dummy_lag1 = lag(drought_dummy, n = 1, default = 0),
    drought_dummy_lag2 = lag(drought_dummy, n = 2, default = 0),
    drought_dummy2_lag1 = lag(drought_dummy2, n = 1, default = 0),
    drought_dummy2_lag2 = lag(drought_dummy2, n = 2, default = 0),
    spei_lag1 = lag(spei, n = 1, default = 0),
    spei_lag2 = lag(spei, n = 2, default = 0),
    share_lag1 = lag(share, n = 1, default = 0),
    share_lag2 = lag(share, n = 2, default = 0),
    agric_stress_lag1 = lag(agric_stress, n = 1, default = 0),
    agric_stress_lag2 = lag(agric_stress, n = 2, default = 0) ) %>%
    ungroup()

dataset_quartiles <- dataset_quartiles %>%
    arrange(district, year) %>%
    group_by(district) %>%
  mutate(
    drought_dummy_lag1 = lag(drought_dummy, n = 1, default = 0),
    drought_dummy_lag2 = lag(drought_dummy, n = 2, default = 0),
    drought_dummy2_lag1 = lag(drought_dummy2, n = 1, default = 0),
    drought_dummy2_lag2 = lag(drought_dummy2, n = 2, default = 0),
    spei_lag1 = lag(spei, n = 1, default = 0),
    spei_lag2 = lag(spei, n = 2, default = 0),
    share_lag1 = lag(share, n = 1, default = 0),
    share_lag2 = lag(share, n = 2, default = 0),
    agric_stress_lag1 = lag(agric_stress, n = 1, default = 0),
    agric_stress_lag2 = lag(agric_stress, n = 2, default = 0) ) %>%
    ungroup()

hh_dataset <- hh_dataset %>%
    arrange(district, year) %>%
    group_by(district) %>%
  mutate(
    drought_dummy_lag1 = lag(drought_dummy, n = 1, default = 0),
```

```
    drought_dummy_lag2 = lag(drought_dummy, n = 2, default = 0),
    drought_dummy2_lag1 = lag(drought_dummy2, n = 1, default = 0),
    drought_dummy2_lag2 = lag(drought_dummy2, n = 2, default = 0),
    spei_lag1 = lag(spei, n = 1, default = 0),
    spei_lag2 = lag(spei, n = 2, default = 0),
    share_lag1 = lag(share, n = 1, default = 0),
    share_lag2 = lag(share, n = 2, default = 0),
    agric_stress_lag1 = lag(agric_stress, n = 1, default = 0),
    agric_stress_lag2 = lag(agric_stress, n = 2, default = 0) ) %>%
    ungroup()
```

```
# One by one, for each column, check if there are NA
#any(is.na(dataset$agric_income))

# Find which rows have the NAs, if any found above
#dataset[is.na(dataset$fdcons), ]

# Save
save(dataset, file = "final_data.Rdata")
save(dataset_deciles, file = "final_data_deciles.Rdata")
save(dataset_quartiles, file = "final_data_quartiles.Rdata")
save(hh_dataset, file = "final_data_household.Rdata")
```