Development Economics: Project Data Preparation

Contents

Pour Ryan: 1) Variable “poverty”. Comme resultat final on a besoin de “share of people in poverty”. Pour le moment la variable est = 1, 2, 3. Selon si pas pauvre, pauvre, ou très pauvre (jsp quel ordre). Faut transformer en dummy qui dit = 1 pour pauvre, et très pauvre et égal a 0 si pas pauvre. Faire gaffe que selon les années ils peuvent changer la definition de si pov=3 veut dire très pauvre ou pas pauvre. Mettre dans les dataset “hh_dataset” et “dataset”

2) Variable “urban” qui decrit si le household vis en region urbaine ou pas. Selon les années contient les valuers “1”,“0”,“Yerevan”. Jsp pk ils font la diff avec Yerevan mais nous on la veut pas. Decouvrir si Yerevan veut urban (je suis 90% sûr que oui). Résultat final: urban = 1 si urban. Mettre dans les dataset “hh_dataset”, “dataset”, “dataset_quartiles”, et “dataset_deciles”. Pour ces trois derniers, la variable va représenter la quantité de personne dans un district qui vive dans urban ou non. Possiblement elle sera egale a 1 pour les district urban et 0 pour ceux ruraux, mais ça ça dépend si un district a des régions urbaines et rurales, ou si les districts sont uniquement urbain ou ruraux.

3) Variable “agric_income”. Celle ci c’est un cauchemar. Pas seulement elle change de nom a travers les années dans les data excel, mais elle prend le nom d’autre variables. Ce qui veut dire qu’on pas juste dire prend le nom x,y,z et considere que ça fait tout parti de cette variable (comme je fais pour urban et poverty par exemple) et il faut a la place regarder chaque ptn de fichier excel et selectionner pour chaque année le nom correct. Faire ça en ouvrant les dataset excels et en selectionnant individuellement chaque année dans la liste de dataset “raw_data_list”. Faire attention que quand une fichier excel est ouvert sur ton ordi, R ne peut pas le load. Ducoup vraiment faire etape par etape: ouvrir une année excel, trouver la variable qui reporte “income from production and sale of agricultural products (including evaluated barter)”, trouver le nom qu’ils lui donnent cette année la, FERMER EXCEL, ouvrir R et load le data, et ecrire le code qui prend que cette variable la de cette année dans “raw_data_list” et la mettre dans “cleaned_data_list”, idealement en transformant les “NA” en “0”. Ensuite essayer le plus possible de s’assurer que y a pas de probleme dans le dataset final.

4) Faire des graphes similaires à ceux qui existent déja dans ce doc pour visualiser un peu ce qui arrive à ces variables a travers le temps, entre groupe de riche vs pauvres, en groupes de “traités” (drought) et “non-traités”, etc.

5) Regarder si il y a d’autre variables intéressantes dans les fichiers excel. Si t’en trouves, faut après survivre la migraine que t’auras en essayant de comprendre comment ces Armeniens gèrent leur ptn de données de la façon la plus incohérente possible.

6) Si t’es vrmnt motivé et que tu veux un petit cadeau dans les toilettes, trouvez des autres données lieés a l’agriculture en Armenie au niveau regional et ensuite les ajouter aux 4 dataset qu’on creeer dans ce doc.

- poverty (0 = non, 1 = yes)

settlement - 2004 à 2017 : 0 yerevan, 1 urban, 2 rural - 2018 : 1 yerevan, 2 urban, 3 rural - 2019 : 1 urban, 2 rural - 2020 : 1 yerevan, 2 urban, 3 rural - 2021 : 1 urban , 2 rural - 2022 à 2023 : -999999999 missing, 1 urban, 2 rural

a part Yerevan qui la capital y a aucune raison qu’un district entier soit complétement considéré comme rural ou urbain j’ai été check (uniquement avec data de 2012) et y a pas un district qui n’est pas à la fois rural et
urbain Pour Yerevan, j’ai regarder 2019 et 2022 (qui sont uniquement rural et urbain) je n’ai trouvé aucune observation ou une personne vivait a Yerevan et rural (uniquement urbain)

Agri income 2004: y1_3drm.10 2005 : y1_3drm.10 2006 : y1_3drm.10 2007 : y1_3drm.10 2008 : y1_3drm.10 2009 : y1_3drm.10 2010 : y1_3drm.10 2011 : y1_3drm.10 2012 : y1_3drm.10 2013 : y1_3drm.10 2014 : y1_3drm.10 2015 : y1_3drm.13 (y1_3drm.10 is unemp) 2016 : y1_3drm.3 (y1_3drm.10 is income child) 2017 : y1_3drm.3 (y1_3drm.10 is income child) 2018 : y1_3drm.3 (y1_3drm.10 is income child) 2019 : y1_3amd.3.00 (y1_3drm.10 don’t exist) 2020 : y1_3amd.3.00 (y1_3drm.10 don’t exist) 2021 :

y1_3amd.3.00 (y1_3drm.10 don't exist) 2022 :y1_3amd.3.00 (y1_3drm.10 don't exist) 2023 : (y1_3drm.10 don't exist) - WOW : wow wow wow wow - je croyais que income prod était missing en 2020, 2022, 2023, parce que y avait pas de variable dans la liste. mais en fait MDR pour 2020, 2022 il s'y trouve (enfin je crois mais ça a l'air) mais c'est pas noté dans la ptn de liste, pour 2023 j'ai pas trouvé

la variable poverty est hyper bizarre, certains sont considérés pauvres alors qu'il touche 80k et d'autre touche 35k et le sont pas. Et c'est pas mon code ça s'observe clairement dans les excels

Data

Loading the excel data

```
# Load Drought Data
droughtdata = read.csv("household_data/district_yearly_drought_treatment_long.csv")

# Load Agriculture Raw Dataset
raw_agri_data = as.data.frame(read_xlsx("agriculture data/agri.xlsx"))

# 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

```
# Drought: Naming Conventions
colnames(droughtdata)[colnames(droughtdata) == 'district_name'] <- 'district'
droughtdata <- droughtdata %>%
  mutate(district = if_else(district == "Erevan", "Yerevan", district)) %>%
  mutate(district = if_else(district == "Lori", "Lor", district)) %>%
  mutate(district = if_else(district == "Vayots Dzor", "Vayots dzor", district))
colnames(droughtdata)[colnames(droughtdata) == 'dummy_drought_consecutive'] <- 'yearly_flag'

# 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
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
```

```r
# 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
```

Merging and Aggregating

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

```r
# 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 and final touches
dataset <- left_join(dataset, droughtdata, by = c("year", "district"))
colnames(dataset)[which(names(dataset) == "yearly_flag")] <- "drought"

dataset_deciles <- left_join(dataset_deciles, droughtdata, by = c("year", "district"))
colnames(dataset_deciles)[which(names(dataset_deciles) == "yearly_flag")] <- "drought"

dataset_quartiles <- left_join(dataset_quartiles, droughtdata, by = c("year", "district"))
colnames(dataset_quartiles)[which(names(dataset_quartiles) == "yearly_flag")] <- "drought"

hh_dataset <- left_join(hh_dataset, droughtdata, by = c("year", "district"))
colnames(hh_dataset)[which(names(hh_dataset) == "yearly_flag")] <- "drought"

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

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

#Temp 2010 fix (is NA in Jonas' dataset)
dataset$drought[is.na(dataset$drought)] <- 0
dataset_deciles$drought[is.na(dataset_deciles$drought)] <- 0
dataset_quartiles$drought[is.na(dataset_quartiles$drought)] <- 0
hh_dataset$drought[is.na(hh_dataset$drought)] <- 0
```

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

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

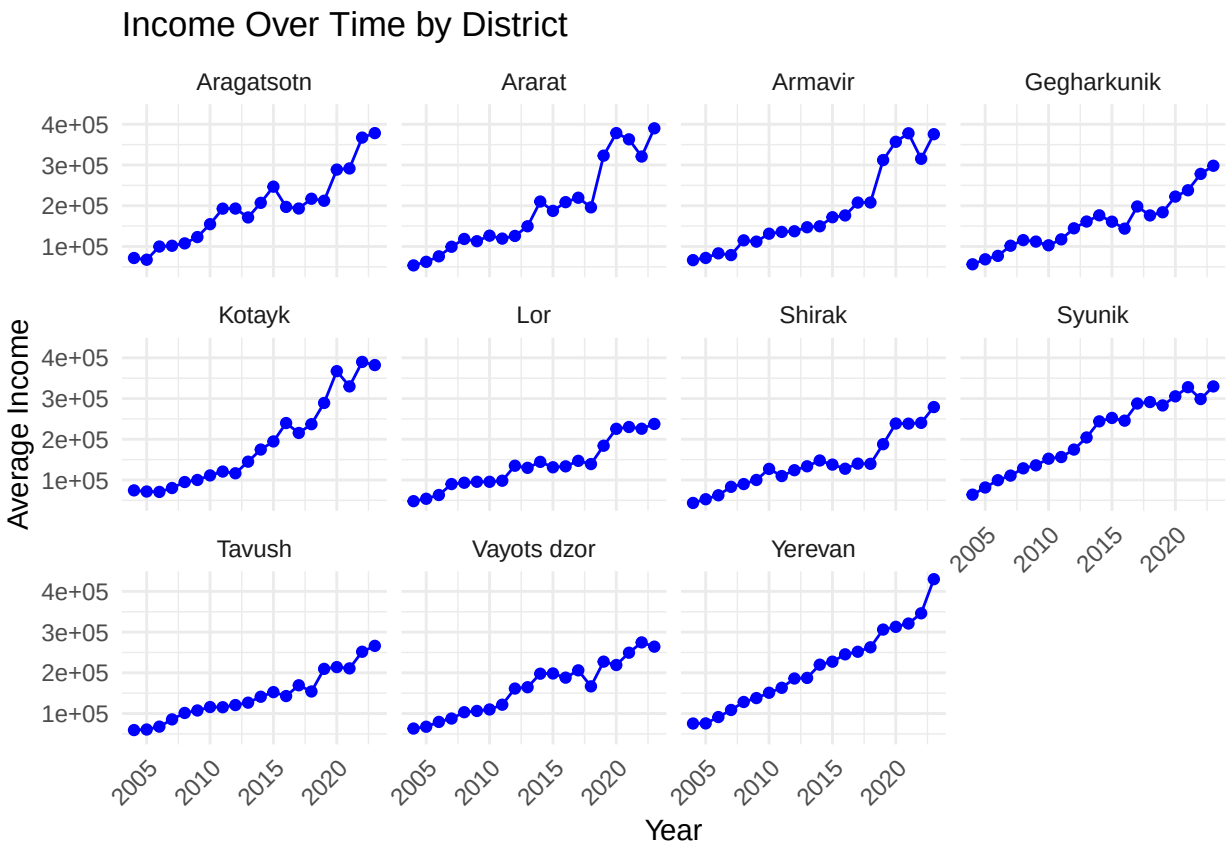
```
## [1] TRUE
```

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

# Save
save(dataset, file = "final_data.Rdata")
```

Descriptive Evidence

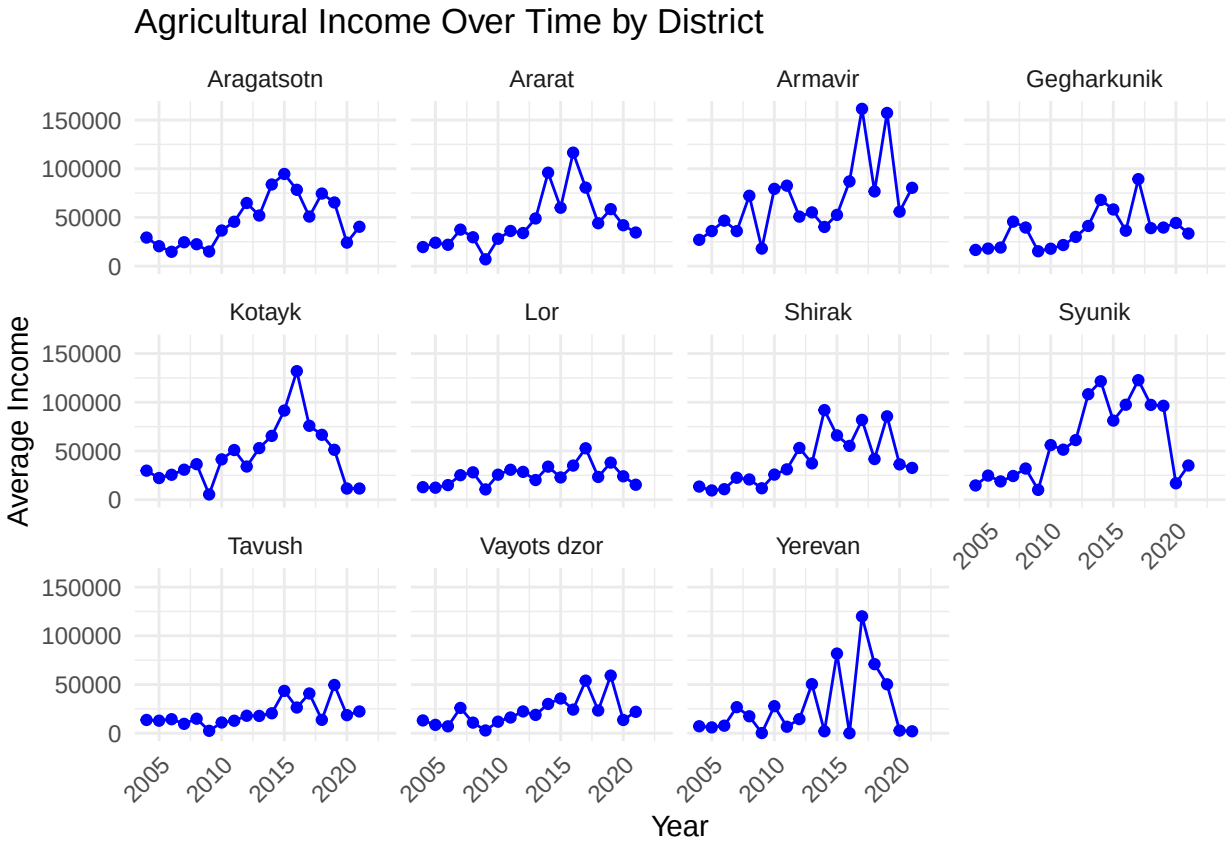
By District

```
# Income trend over time, faceted by district
ggplot(dataset, aes(x = year, y = income)) +
  geom_line(aes(group = district), color = "blue") + # Group by district to connect dots
  geom_point(color = "blue") +

  # Creates a chart for each district
  facet_wrap(~ district) +
  labs(
    title = "Income Over Time by District",
    x = "Year",
    y = "Average Income") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



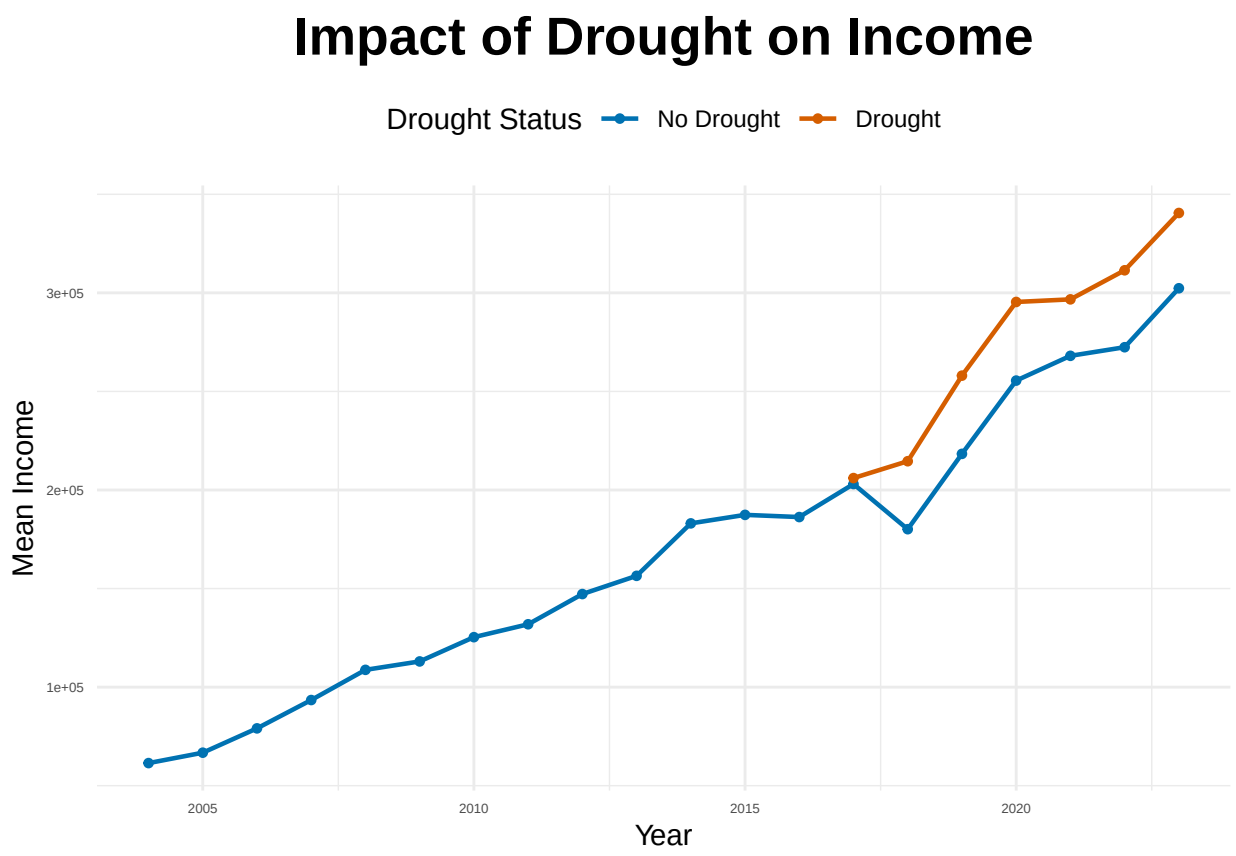

```
# Agriculture Income trend over time, faceted by district
ggplot(dataset, aes(x = year, y = agric_income)) +
  geom_line(aes(group = district), color = "blue") + # Group by district to connect dots
  geom_point(color = "blue") +

  # Creates a chart for each district
```

```
  facet_wrap(~ district) +
  labs(
    title = "Agricultural Income Over Time by District",
    x = "Year",
    y = "Average Income") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



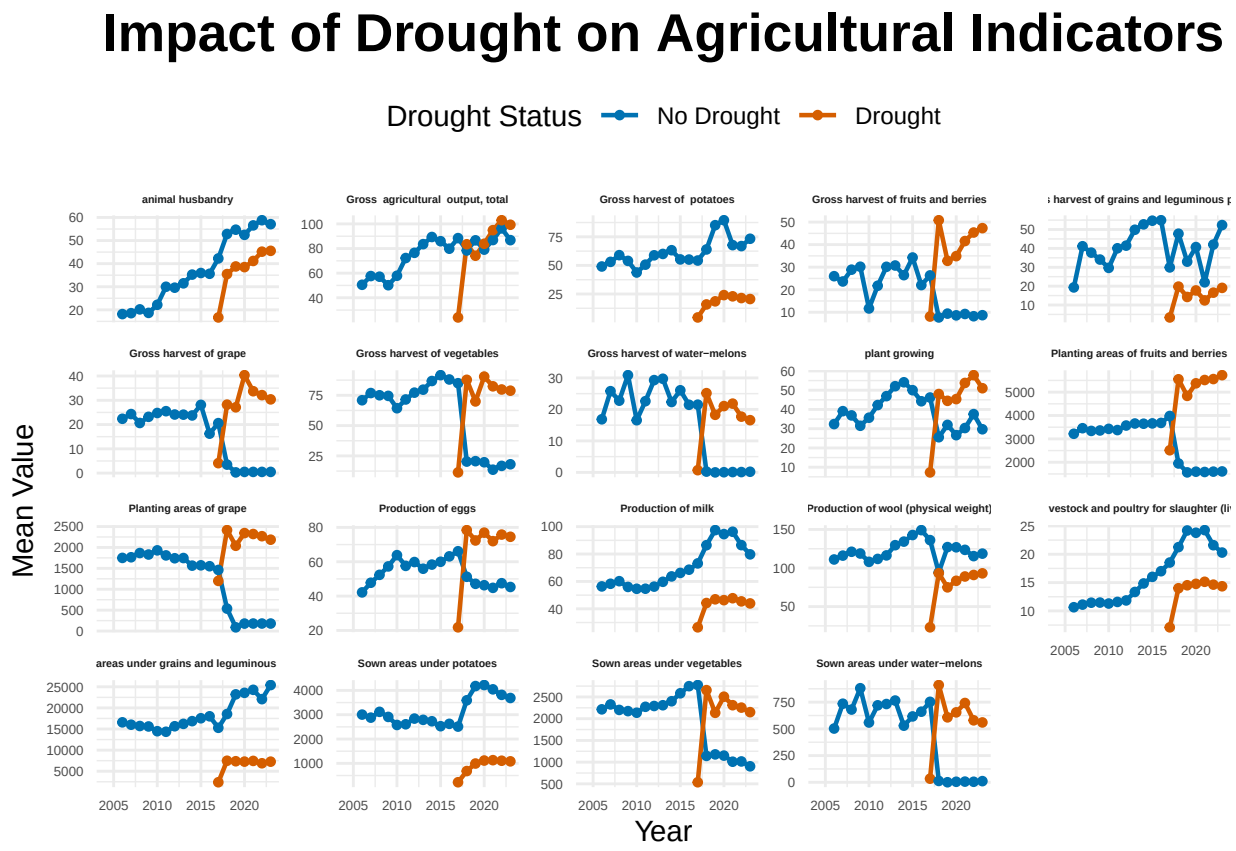

```
# Compute, for each years, the mean of income between districts
#  that experience drought vs no drought

inc_plot <- dataset %>%
  group_by(year, drought) %>%
  summarise(income = mean(income, na.rm = TRUE), n_district = n())

ggplot(inc_plot, aes(x = year, y = income,
                     color = factor(drought), group = factor(drought))) +
  geom_line(linewidth = 0.8) +
  geom_point(size = 1.2) +
  scale_color_manual(values = c("0" = "#0072B2", "1" = "#D55E00"),
                     labels = c("No Drought", "Drought")) +
  labs(x = "Year", y = "Mean Income", color = "Drought Status",
       title = "Impact of Drought on Income") +
  theme_minimal() +
  theme(legend.position = "top",
```

```
    axis.text.x = element_text(size = 5),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
    strip.text = element_text(size = 4, face = "bold"))
```




```
# take variables list
vars_agri <- c("Gross  agricultural  output, total", "plant growing", "animal husbandry",
           "Sown areas under grains and leguminous plants", "Sown areas under potatoes",
           "Sown areas under vegetables", "Sown areas under water-melons",
           "Planting areas of fruits and berries", "Planting areas of grape",
           "Gross harvest of grains and leguminous plants", "Gross harvest of  potatoes",
           "Gross harvest of vegetables", "Gross harvest of water-melons",
           "Gross harvest of fruits and berries", "Gross harvest of grape",
           "Realized livestock and poultry for slaughter (live weight)",
           "Production of milk", "Production of eggs", "Production of wool (physical weight)")

# Compute, for each years, the mean of all agriculture variables between region
#that experience drought vs no drought
agri_plot <- dataset %>%
  group_by(year, drought) %>%
  summarise(across(all_of(vars_agri), ~ mean(.x, na.rm = TRUE), .names = "{.col}"),
            .groups = "drop") %>%
  pivot_longer(-c(year, drought), names_to = "variable", values_to = "mean_value")

ggplot(agri_plot, aes(x = year, y = mean_value,
```

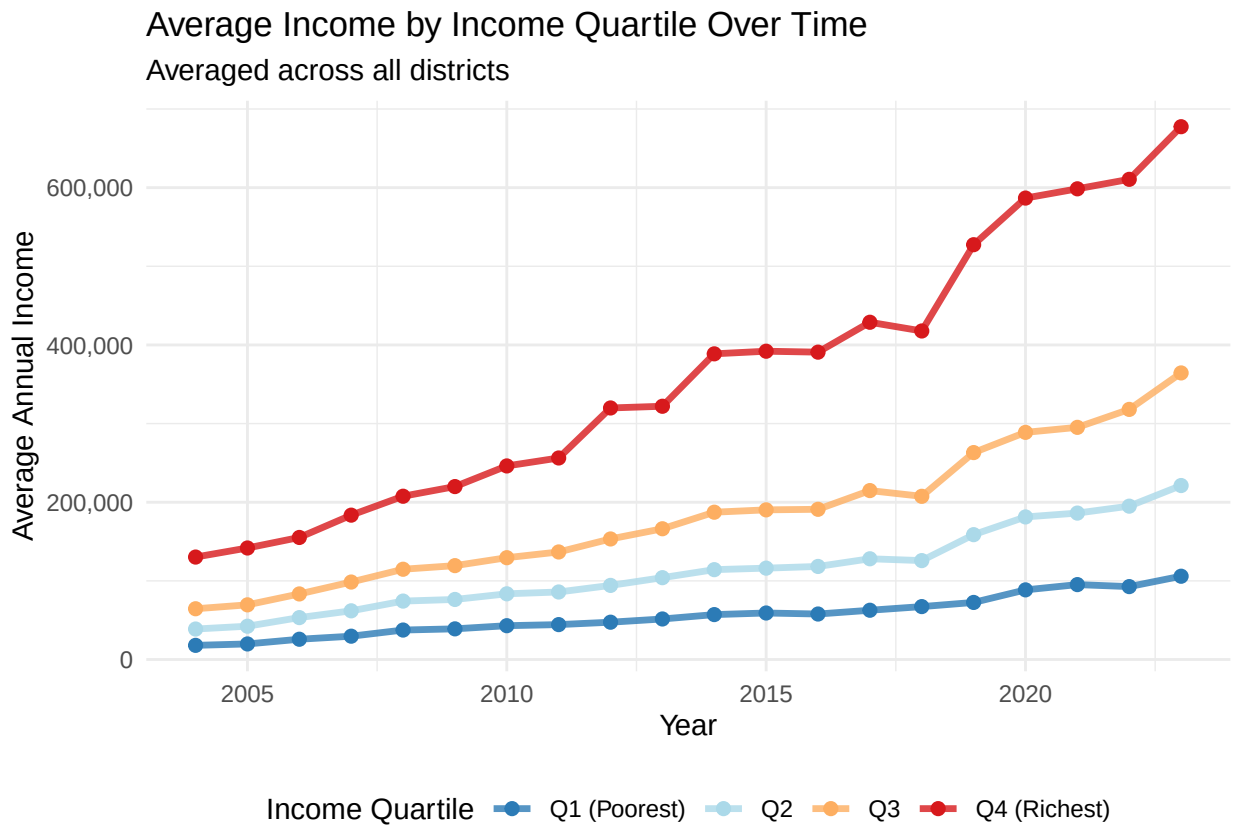
```
                      color = factor(drought), group = factor(drought))) +
  geom_line(linewidth = 0.8) +
  geom_point(size = 1.2) +
  scale_color_manual(values = c("0" = "#0072B2", "1" = "#D55E00"),
                     labels = c("No Drought", "Drought")) +
  facet_wrap(~ variable, scales = "free_y") +
  labs(x = "Year", y = "Mean Value", color = "Drought Status",
       title = "Impact of Drought on Agricultural Indicators") +
  theme_minimal() +
  theme(legend.position = "top",
    axis.text.x = element_text(size = 5),
    axis.text.y = element_text(size = 5),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
    strip.text = element_text(size = 4, face = "bold"))
```



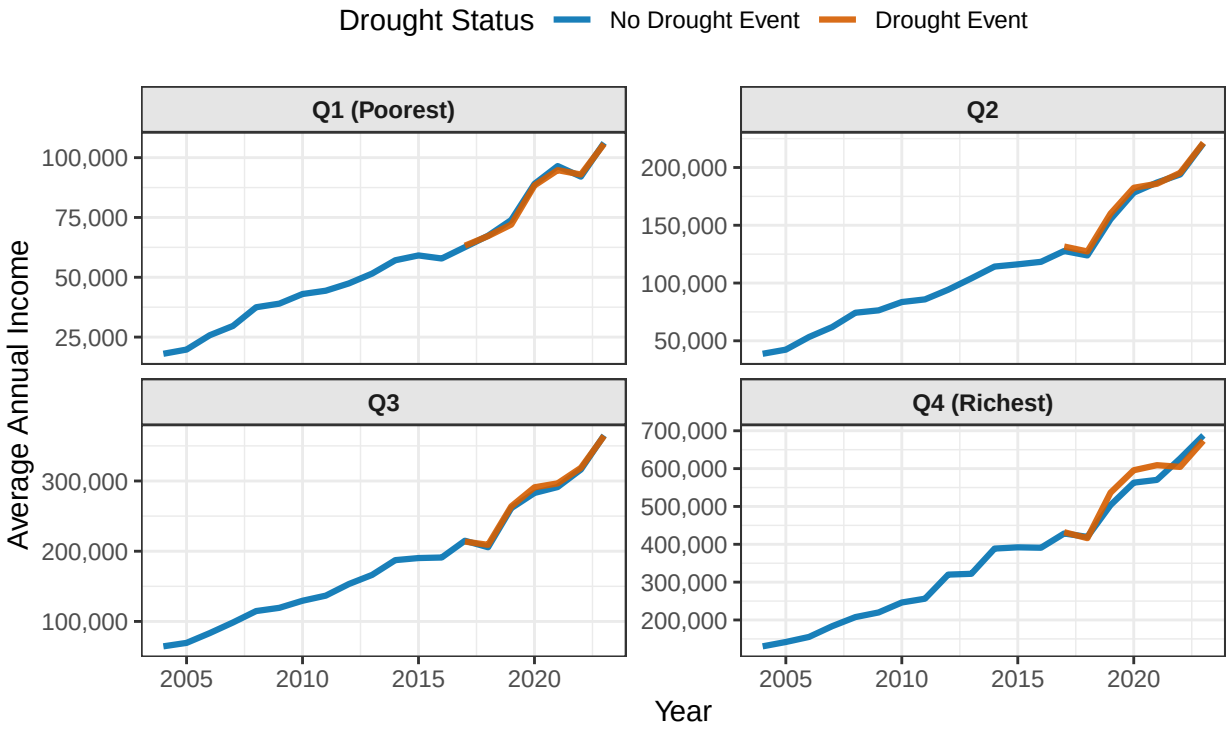
With Quartiles

```
# Data Prep
dataset_prepped_q <- dataset_quartiles %>%
  mutate(drought_status = factor(drought,
                                 levels = c(0, 1),
                                 labels = c("No Drought Event", "Drought Event")),
         income_quartile = factor(national_quartile,
```

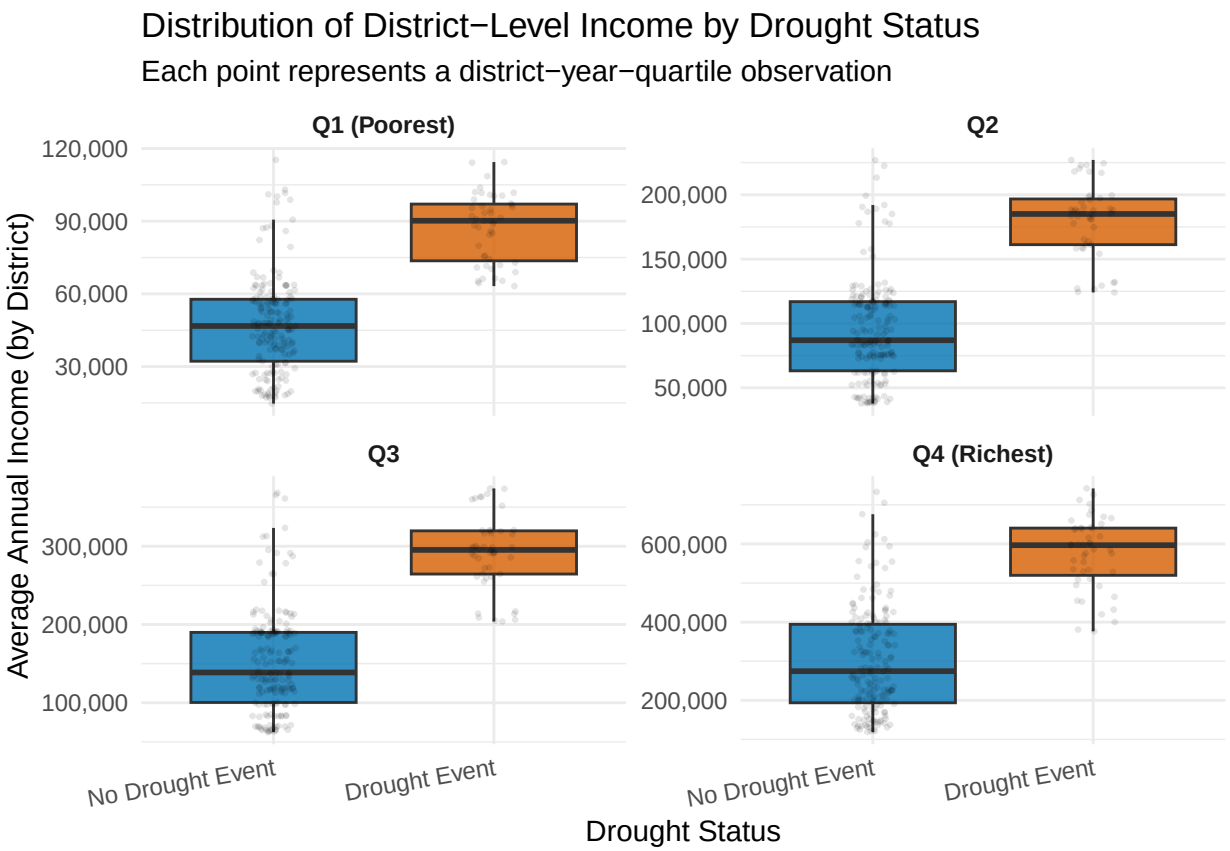
```
                          levels = c(1, 2, 3, 4),
                          labels = c("Q1 (Poorest)", "Q2", "Q3", "Q4 (Richest)")))

# A1. Aggregate the data: Find the mean income for each year and quartile
plot1_data_q <- dataset_prepped_q %>%
  group_by(year, income_quartile) %>%
  summarize(avg_income = mean(income, na.rm = TRUE),
            avg_agr_income = mean(agric_income, na.rm = TRUE), .groups = 'drop')


# A2. Create the plot q
ggplot(plot1_data_q, aes(x = year, y = avg_income, color = income_quartile, group = income_quartile)) +
  geom_line(linewidth = 1.2, alpha = 0.8) +
  geom_point(size = 2) +

  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) + # Formats y-axis labels (e.g., 50,000)
  scale_color_brewer(palette = "RdYlBu", direction = -1) +
  labs(
    title = "Average Income by Income Quartile Over Time",
    subtitle = "Averaged across all districts",
    x = "Year",
    y = "Average Annual Income",
    color = "Income Quartile") +
  theme_minimal() + theme(legend.position = "bottom")
```



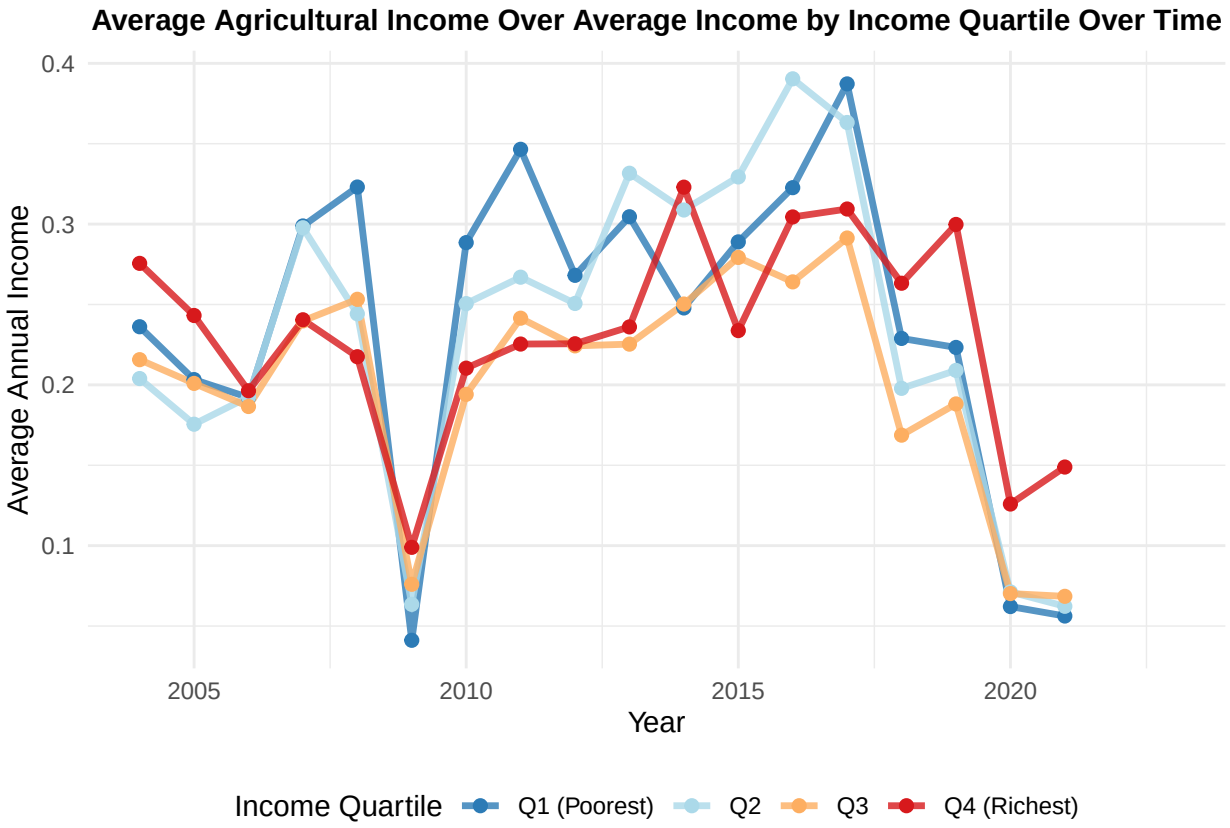
```r
# B1. Aggregate data: Mean income by year, quartile, AND drought status
plot2_data_q <- dataset_prepped_q %>%
  group_by(year, income_quartile, drought_status) %>%
  summarize(avg_income = mean(income, na.rm = TRUE),
            avg_agr_income = mean(agric_income, na.rm = TRUE), .groups = 'drop')


# B2. Create the faceted plot q
ggplot(plot2_data_q, aes(x = year, y = avg_income, color = drought_status, group = drought_status)) +
  geom_line(linewidth = 1.1, alpha = 0.9) +

  # Create 4 separate plots, one for each 'income_quartile'
  facet_wrap(~ income_quartile, scales = "free_y") +

  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Impact of Drought Events on Income, by Income Quartile",
    subtitle = "Average income trends faceted by income group",
    x = "Year",
    y = "Average Annual Income",
    color = "Drought Status") +
  theme_bw() + # A clean theme
  theme(
    legend.position = "top",
    strip.background = element_rect(fill = "grey90"), # Style the facet labels
    strip.text = element_text(face = "bold") )
```



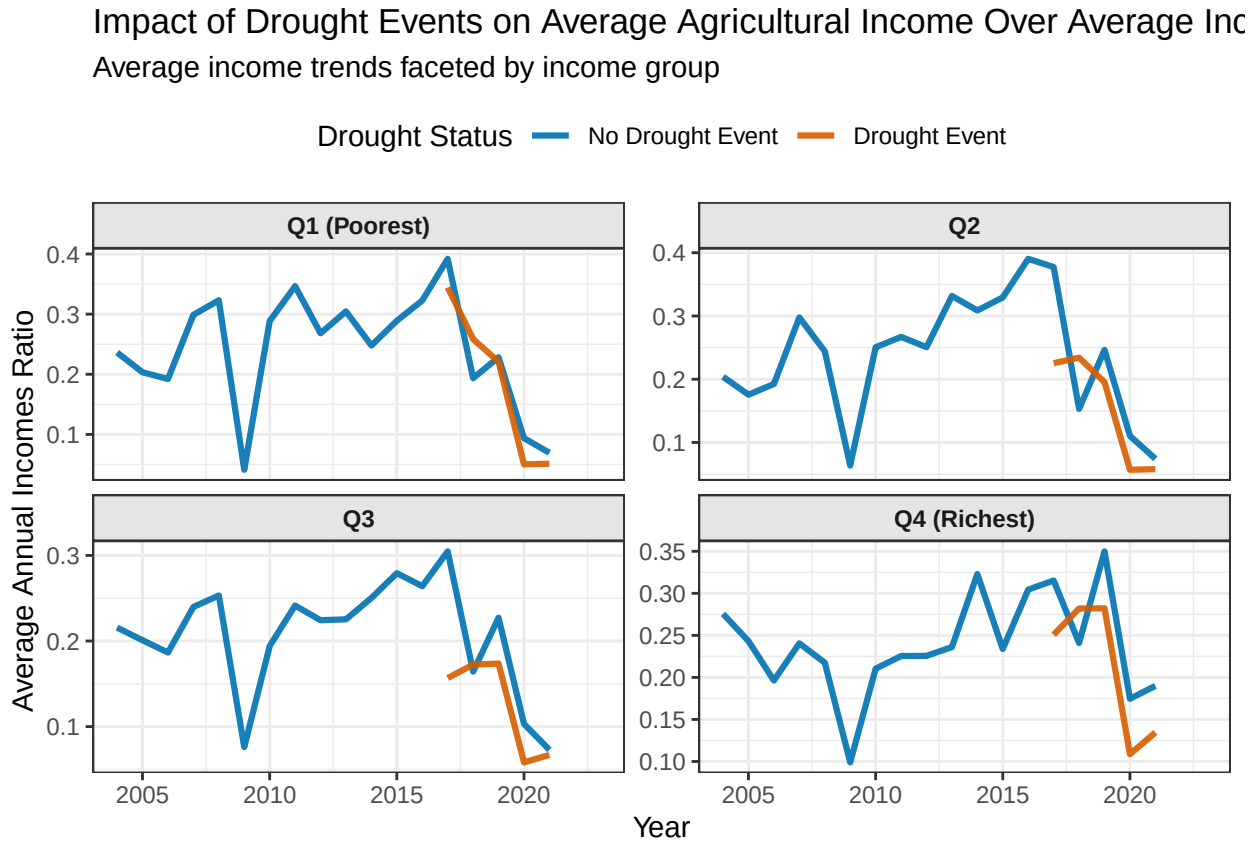

```r
#plot C: boxplots
ggplot(dataset_prepped_q, aes(x = drought_status, y = income, fill = drought_status)) +
  geom_boxplot(alpha = 0.8, outlier.shape = NA) + # 'outlier.shape = NA' hides outliers for now
  geom_jitter(width = 0.1, alpha = 0.1, size = 0.5) +

  # --- Faceting ---
  facet_wrap(~ income_quartile, scales = "free_y") +

  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Distribution of District-Level Income by Drought Status",
    subtitle = "Each point represents a district-year-quartile observation",
    x = "Drought Status",
    y = "Average Annual Income (by District)",
    fill = "Drought Status") +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 10, hjust = 1),
        strip.text = element_text(face = "bold"))
```



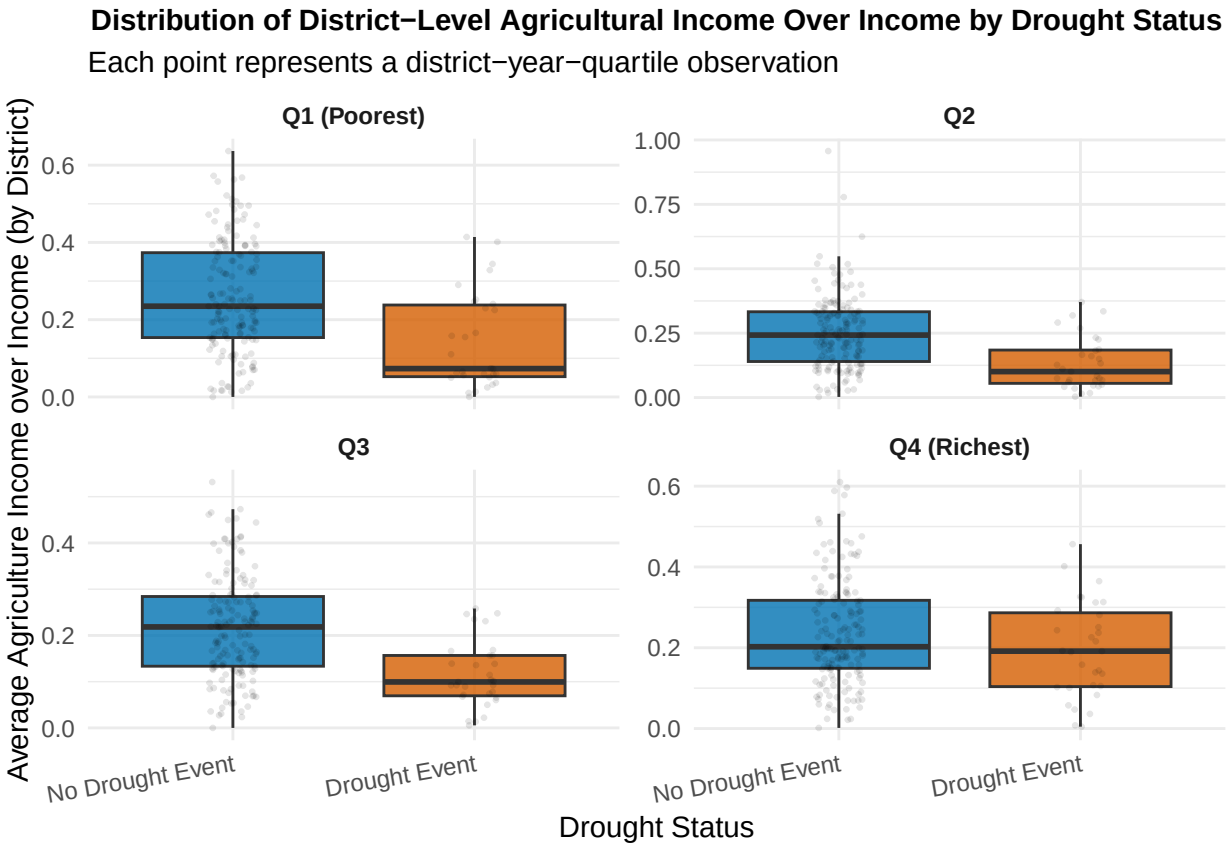

```r
# A1.2 Plot q ratio of Average Agriculture Income over Average Income
ggplot(plot1_data_q, aes(x = year, y = avg_agr_income/avg_income, color = income_quartile, group = incor
  geom_line(linewidth = 1.2, alpha = 0.8) +
  geom_point(size = 2) +
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) + # Formats y-axis labels (e.g., 50,000)
  scale_color_brewer(palette = "RdYlBu", direction = -1) +
  labs(
    title = "Average Agricultural Income Over Average Income by Income Quartile Over Time",
    x = "Year",
    y = "Average Annual Income",
    color = "Income Quartile") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, face = "bold", hjust = 0.5))
```



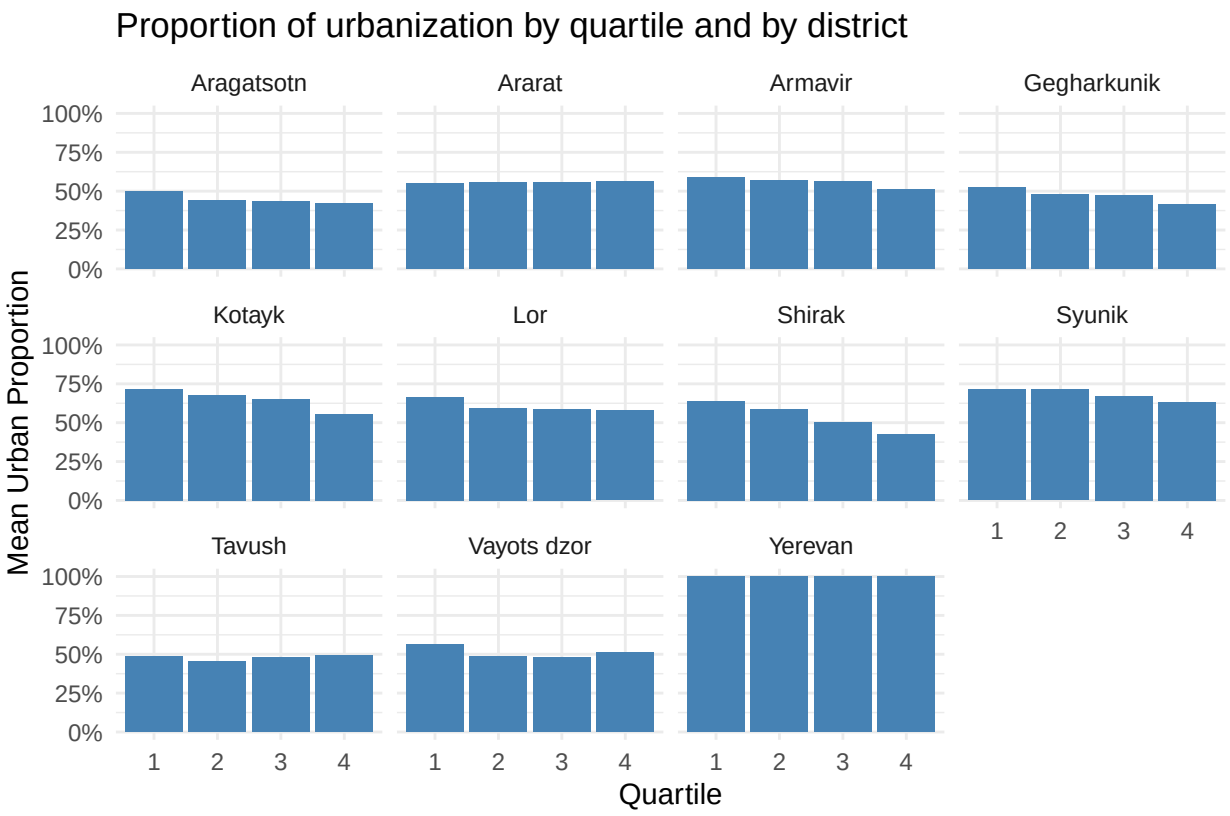

```r
# B2.2 faceted Plot q ratio of Average Agriculture Income over Average Income
ggplot(plot2_data_q, aes(x = year, y = avg_agr_income/avg_income, color = drought_status, group = drough
  geom_line(linewidth = 1.1, alpha = 0.9) +
  # Create 4 separate plots, one for each 'income_quartile'
  facet_wrap(~ income_quartile, scales = "free_y") +
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Impact of Drought Events on Average Agricultural Income Over Average Income, by Income Qua
    subtitle = "Average income trends faceted by income group",
    x = "Year",
    y = "Average Annual Incomes Ratio",
    color = "Drought Status") +
  theme_bw() + # A clean theme
  theme(
    legend.position = "top",
    strip.background = element_rect(fill = "grey90"), # Style the facet labels
    strip.text = element_text(face = "bold"))
```




```r
#plot C.2: boxplots
ggplot(dataset_prepped_q, aes(x = drought_status, y = agric_income/income, fill = drought_status)) +
  geom_boxplot(alpha = 0.8, outlier.shape = NA) + # 'outlier.shape = NA' hides outliers for now
  geom_jitter(width = 0.1, alpha = 0.1, size = 0.5) +
  # --- Faceting ---
  facet_wrap(~ income_quartile, scales = "free_y") +
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Distribution of District-Level Agricultural Income Over Income by Drought Status",
    subtitle = "Each point represents a district-year-quartile observation",
    x = "Drought Status",
    y = "Average Agriculture Income over Income (by District)",
    fill = "Drought Status") +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 10, hjust = 1),
        strip.text = element_text(face = "bold"),
        plot.title = element_text(size = 11, face = "bold", hjust = 0.5))
```



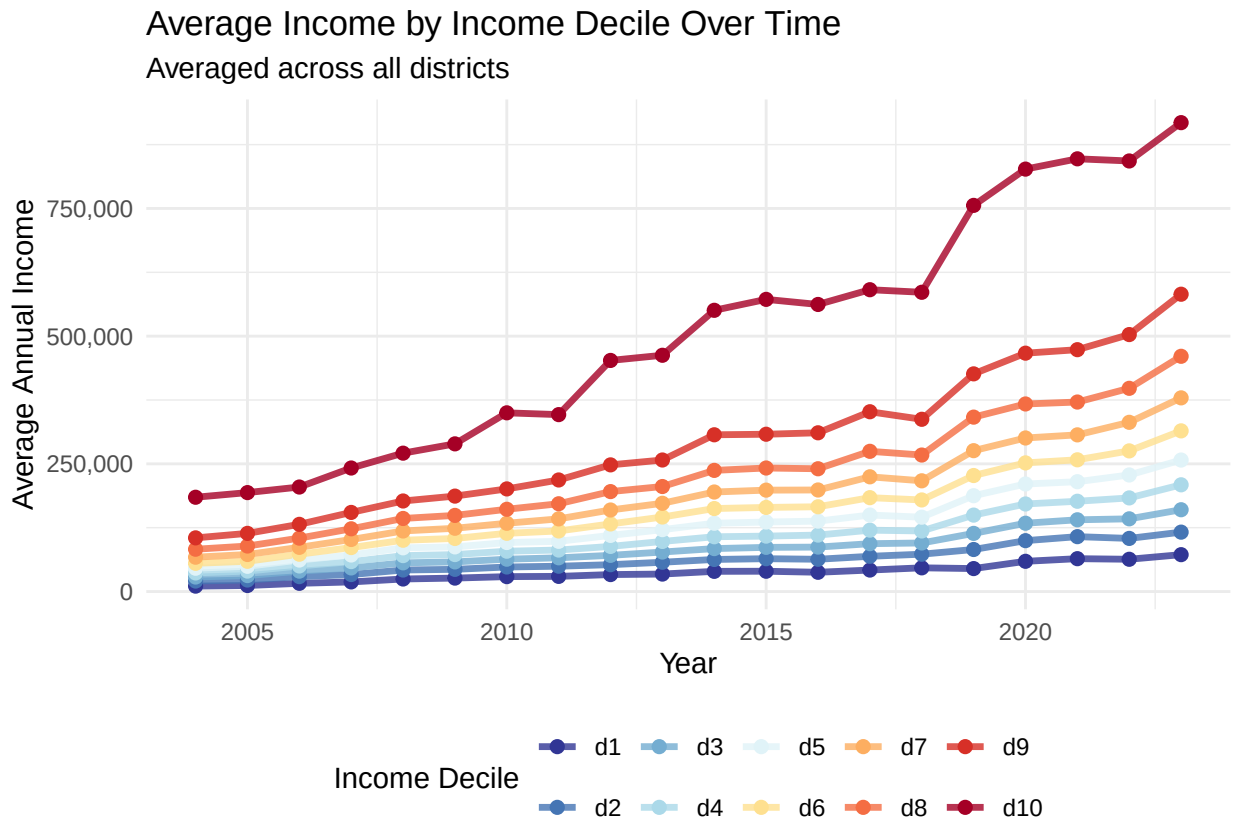

```
#
dataset_prepp_urb = dataset_quartiles %>%
  group_by(district, national_quartile) %>%
  summarize(mean_urban = mean(urban, na.rm = TRUE), .groups = 'drop')

ggplot(dataset_prepp_urb, aes(x = factor(national_quartile), y = mean_urban)) +
  geom_col(fill = "steelblue") +
  facet_wrap(~district) +
  labs(
    title = "Proportion of urbanization by quartile and by district",
    x = "Quartile",
    y = "Mean Urban Proportion",
    caption = "1 = urban, 0 = rural"
  ) +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```



With Deciles

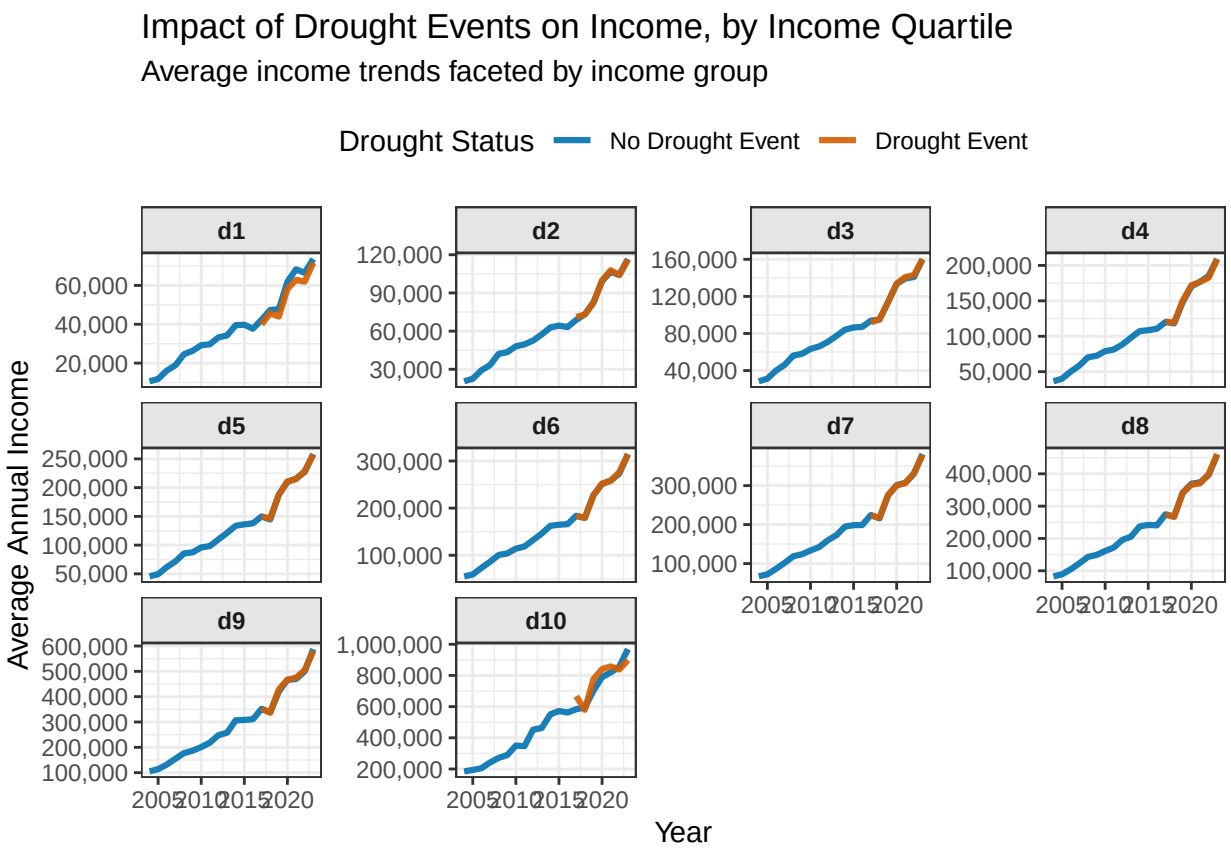
```
dataset_prepped_d <- dataset_deciles %>%
  mutate(
    drought_status = factor(drought,
                            levels = c(0, 1),
                            labels = c("No Drought Event", "Drought Event")),
    income_decile = factor(national_decile,
                            levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
                            labels =
                            c("d1","d2","d3","d4","d5","d6","d7","d8","d9","d10")))

# A1. Aggregate the data: Find the mean income for each year and by decile
plot1_data_d <- dataset_prepped_d %>%
  group_by(year, income_decile) %>%
  summarize(avg_income = mean(income, na.rm = TRUE),
            avg_agr_income = mean(agric_income, na.rm = TRUE), .groups = 'drop')


# A2. Create the plot d
ggplot(plot1_data_d, aes(x = year, y = avg_income, color = income_decile, group = income_decile)) +
  geom_line(linewidth = 1.2, alpha = 0.8) +
  geom_point(size = 2) +
```

```r
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) + # Formats y-axis labels (e.g., 50,000)
  scale_color_brewer(palette = "RdYlBu", direction = -1) +
  labs(
    title = "Average Income by Income Decile Over Time",
    subtitle = "Averaged across all districts",
    x = "Year",
    y = "Average Annual Income",
    color = "Income Decile"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



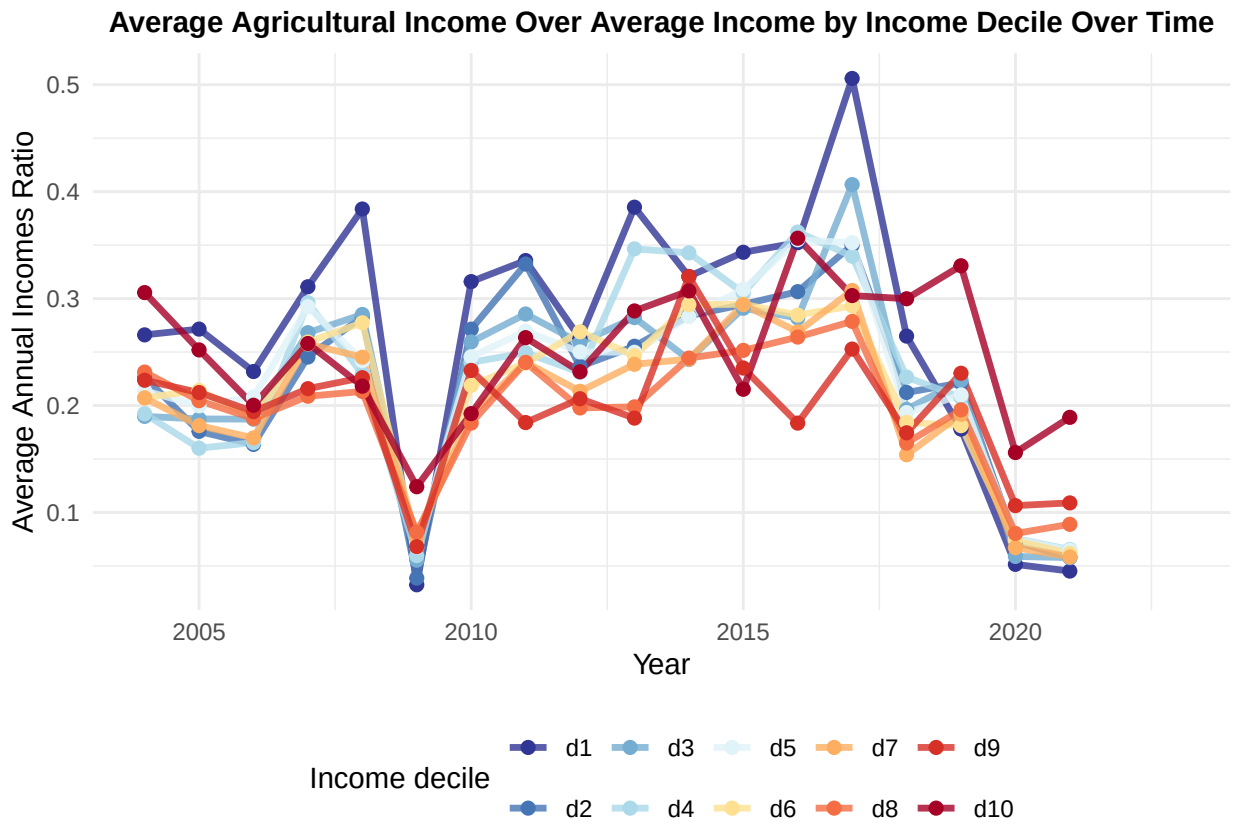

```r
# B1. Aggregate data: Mean income, agriculture income and production by year, decile, AND drought statu
plot2_data_d <- dataset_prepped_d %>%
  group_by(year, income_decile, drought_status) %>%
  summarize(avg_income = mean(income, na.rm = TRUE),
            avg_agr_income = mean(agric_income, na.rm = TRUE), .groups = 'drop')

# B2. Create plot 2 with deciles
ggplot(plot2_data_d, aes(x = year, y = avg_income, color = drought_status, group = drought_status)) +
  geom_line(linewidth = 1.1, alpha = 0.9) +

  # Create 10 separate plots, one for each 'income_decile'
  facet_wrap(~ income_decile, scales = "free_y") +
```

```
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Impact of Drought Events on Income, by Income Quartile",
    subtitle = "Average income trends faceted by income group",
    x = "Year",
    y = "Average Annual Income",
    color = "Drought Status") +
  theme_bw() + # A clean theme
  theme(legend.position = "top",
        strip.background = element_rect(fill = "grey90"),
        strip.text = element_text(face = "bold") )
```

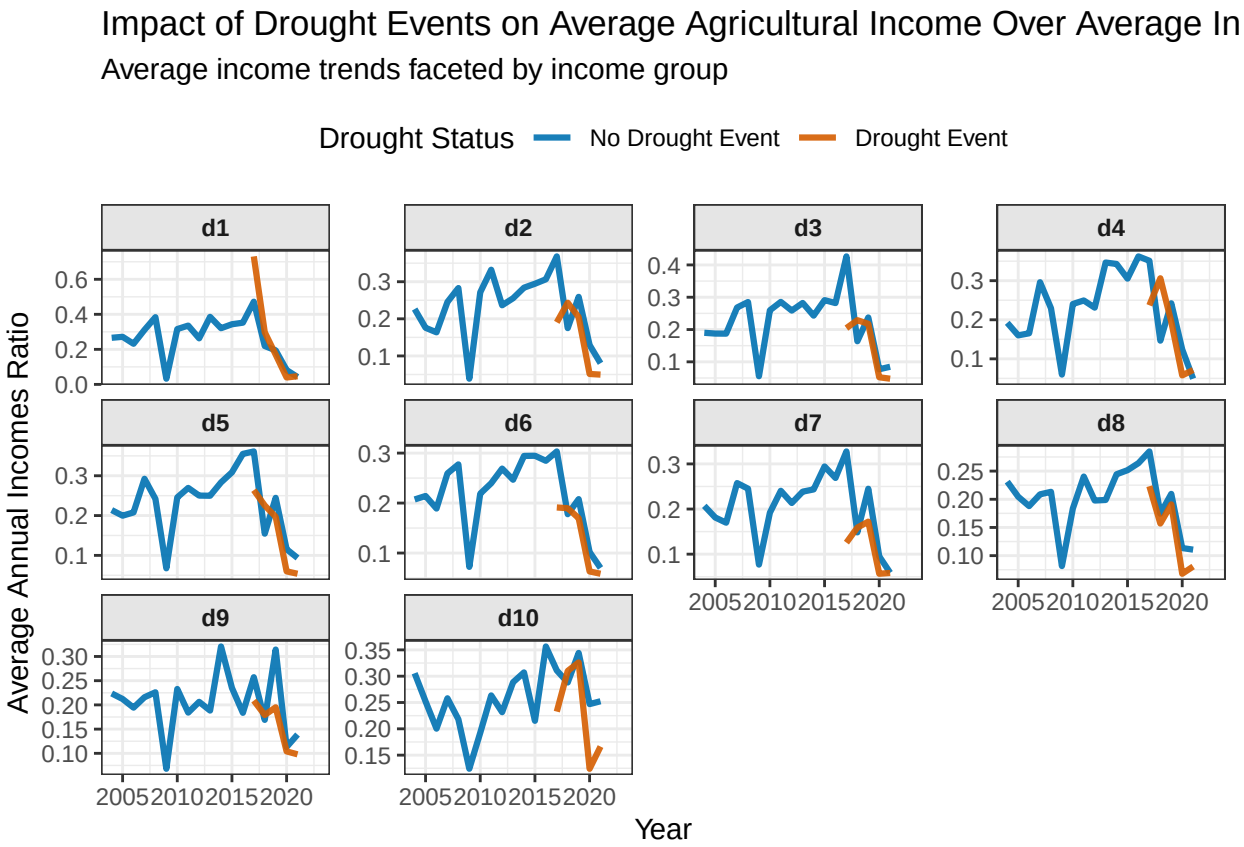



```
# A2. Create the plot d
ggplot(plot1_data_d, aes(x = year, y = avg_agr_income/avg_income, color = income_decile, group = income_
  geom_line(linewidth = 1.2, alpha = 0.8) +
  geom_point(size = 2) +
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) + # Formats y-axis labels (e.g., 50,000)
  scale_color_brewer(palette = "RdYlBu", direction = -1) +
  labs(
    title = "Average Agricultural Income Over Average Income by Income Decile Over Time",
    x = "Year",
    y = "Average Annual Incomes Ratio",
```

```
    color = "Income decile") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, face = "bold", hjust = 0.5))
```




```
# B2. Create the faceted plot d
ggplot(plot2_data_d, aes(x = year, y = avg_agr_income/avg_income, color = drought_status, group = droug
  geom_line(linewidth = 1.1, alpha = 0.9) +
  facet_wrap(~ income_decile, scales = "free_y") +
  # --- Aesthetics & Labels ---
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = c("No Drought Event" = "#0072B2", "Drought Event" = "#D55E00")) +
  labs(
    title = "Impact of Drought Events on Average Agricultural Income Over Average Income, by Income Dec
    subtitle = "Average income trends faceted by income group",
    x = "Year",
    y = "Average Annual Incomes Ratio",
    color = "Drought Status") +
  theme_bw() + # A clean theme
  theme(
    legend.position = "top",
    strip.background = element_rect(fill = "grey90"), # Style the facet labels
    strip.text = element_text(face = "bold"))
```


Impact of Drought Events on Average Agricultural Income Over Average In
Average income trends faceted by income group
Drought Status
No Drought Event
Drought Event
d1
d2
d3
d4
d5
d6
d7
d8
d9
d10
Average Annual Incomes Ratio
Year
2005
2010
2015
2020