The relationship between unemployment and crime in the Netherlands

Romeo Schoonderbeek, 2868407 Paul Claessens, 2859603 Furkan Kadir Öztürk, 2852640

2025-06-24

Set-up your environment

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
require(cbsodataR)
```

```
## Loading required package: cbsodataR
```

```
require(sf)
```

```
## Loading required package: sf
## Linking to GEOS 3.13.0, GDAL 3.5.3, PROJ 9.5.1; sf_use_s2() is TRUE
```

Part 1 - Identify a Social Problem

Use APA referencing throughout your document. Here's a link to some explanation.

1.1 Describe the Social Problem

Include the following:

- Why is this relevant?
- ...

Part 2 - Data Sourcing

2.1 Load in the data

```
# provincelevel
DS_Panel_Unemployment <- read.csv("data/raw_data/unemployment_data_transform.csv")
DS_Panel_Crime <- read.csv("data/raw_data/panel_data_crime.csv")
DS_Population <- read.csv("data/population_data.csv")
area_data_working <- read.csv("data/raw_data/area_data.csv")

# aggragate

agg_unemployment <- read.csv("data/raw_data/aggragate_total_unemployment.csv")
agg_crime <- read.csv("data/raw_data/aggragate_total_crime.csv")
agg_population <- read.csv("data/raw_data/population_data_raw.csv")

# subpopulation

criminaliteit_per_leeftijd <- read.csv("data/raw_data/criminaliteit_per_leeftijd.csv")

unemployment_age <- read.csv("data/raw_data/unemployment_per_age.csv")
```

2.2 Provide a short summary of the dataset(s)

In this case we see 28 variables, but we miss some information on what units they are in. We also don't know anything about the year/moment in which this data has been captured.

```
inline_code = TRUE
```

These are things that are usually included in the metadata of the dataset. For your project, you need to provide us with the information from your metadata that we need to understand your dataset of choice.

2.3 Describe the type of variables included

Think of things like:

- Do the variables contain health information or SES information?
- Have they been measured by interviewing individuals or is the data coming from administrative sources?

For the sake of this example, I will continue with the assignment...

Part 3 - Quantifying

3.1 Data cleaning

CBS datasets already come in panel data form. Data cleaning will include removing unnecessary rows and columns, adjusting data frames and merging data sets. 3 main data sets are used. Province panel data, aggregate panel data and data sets used for sub population analysis.

```r
# province panel data cleaning

merged_data <- cbind(DS_Panel_Unemployment, DS_Panel_Crime [, 5])

merged_data <-cbind(merged_data, DS_Population [, 4])

merged_data <- merged_data [, -c(1)]
colnames(merged_data) <- c("Year", "province", "Unemployment_Rate", "total_Crimes", "Population")

merged_data <- mutate(merged_data, crimes_per_capita = (total_Crimes/Population)*100)

write.csv(merged_data, "data/working_data/merged_panel_data_final.csv")

area_data_working <- read.csv("data/raw_data/area_data.csv")

colnames(area_data_working) = c("X", "province", "Area(KM2)")

area_data_working <- area_data_working[6:17 ,]

area_data_working[[3]]  = as.numeric(area_data_working[[3]])

area_data_working[, 3] = area_data_working[, 3]/ 100

flevoland <- data.frame(name = "Flevoland", value = 1417 )
colnames(flevoland) = c("province", "Area(KM2)")

area_data_working <- bind_rows(
  area_data_working[1:5, ], flevoland,
  area_data_working[6:nrow(area_data_working), ]
)
```

```
## New names:
## New names:
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
```

```r
area_data_working <- area_data_working %>%
  slice(rep(1:n(), each = 12))

merged_panel_data_final <- cbind(merged_data, area_data_working[, 3])

colnames(merged_panel_data_final) <- c("year", "province", "unemployment_rate", "total_crimes", "popula

merged_panel_data_final <- mutate(merged_panel_data_final, poplation_density = population/area_km2)

# aggregate data cleaning

agg_unemployment <- agg_unemployment[-c(51,52) ,]
agg_unemployment <- agg_unemployment[, -1]

agg_crime <- agg_crime[-82 , ]
agg_crime <- agg_crime[-c(1:31) ,]
```

```r
agg_crime <- agg_crime[, -1]

colnames(agg_unemployment) = c("year", "unemployment")
colnames(agg_crime) = c("year", "total_crimes")

agg_merged <- cbind(agg_unemployment, agg_crime[, 2])

colnames(agg_merged) = c("year", "unemployment", "total_crime")

agg_merged <- agg_merged %>%
  mutate(total_crime = as.numeric(total_crime)) %>%
  mutate(year = as.numeric(year))

str(agg_merged)
```

```
## 'data.frame':    50 obs. of  3 variables:
##  $ year        : num  1975 1976 1977 1978 1979 ...
##  $ unemployment: num  3.5 4.2 4.3 4.2 4.1 4.3 5.7 8 9.6 9.6 ...
##  $ total_crime : num  486670 564025 590880 612370 668235 ...
```

```r
agg_population <-agg_population[, -4]
agg_population <- agg_population[, -1]
agg_population <- agg_population[-c(1:75) ,]

agg_merged <- cbind(agg_merged, agg_population[, 2])

colnames(agg_merged) = c("year", "unemployment", "total_crime", "population")

str(agg_merged)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ year        : num  1975 1976 1977 1978 1979 ...
##  $ unemployment: num  3.5 4.2 4.3 4.2 4.1 4.3 5.7 8 9.6 9.6 ...
##  $ total_crime : num  486670 564025 590880 612370 668235 ...
##  $ population  : int  13600000 13730000 13810000 13900000 13990000 14090000 14210000 14290000 143400
```

```r
agg_merged <- agg_merged %>%
  mutate(crime_per_capita = (total_crime/population)*100)

# subpopulation

crime2021 <- criminaliteit_per_leeftijd %>% filter(Perioden == "2021")

crime2021 <- crime2021 %>% filter(Geboorteland == "Totaal")

crime2021 <- crime2021[, -c(1:2, 4:8)]

colnames(crime2021) <- c("Age", "Crime")

crime2021$Crime = (crime2021$Crime/10000)*100

unemployment_age <- read.csv("data/raw_data/unemployment_per_age.csv")
```

```r
unemployment_age <- unemployment_age[-c(1:6, 19) ,]

unemployment_age <- unemployment_age[, -1]

colnames(unemployment_age) = c("Age", "Unemployment")

unemployment_age$Unemployment <- gsub(",", ".", unemployment_age$Unemployment)

unemployment_age$Unemployment <- as.numeric(unemployment_age$Unemployment)
```

Please use a separate 'R block' of code for each type of cleaning. So, e.g. one for missing values, a new one for removing unnecessary variables etc.

3.2 Generate necessary variables

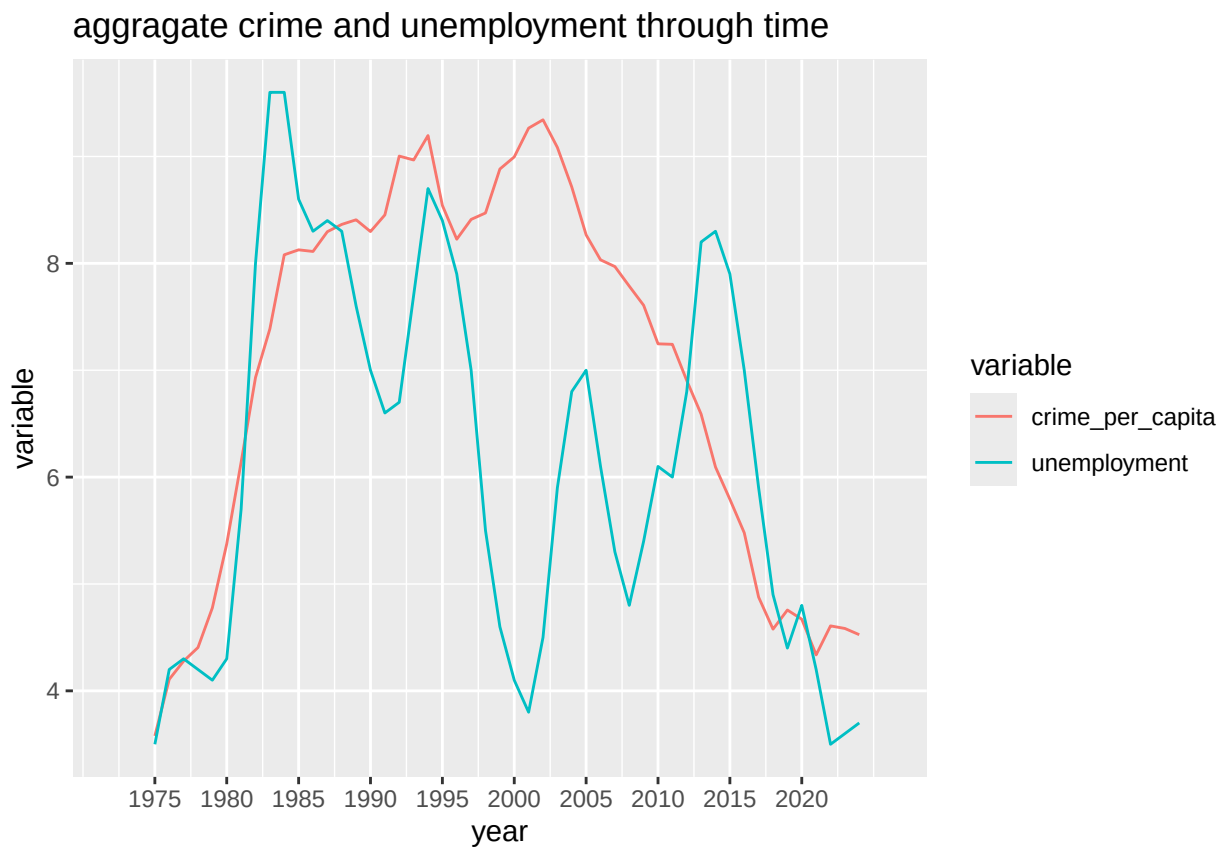
Variable 1

Variable 2

```r
crime2021$Age <- ifelse(crime2021$Age %in% c("12 tot 18 jaar", "18 tot 23 jaar"),
                                    "Youth (12-23)", "Adult+ (23+)")

unemployment_age$Age <- ifelse(unemployment_age$Age %in% c("15 tot 20 jaar", "20 tot 25 jaar"),
                               "Youth (15-25)", "Adult (25+)")
```

3.3 Visualize temporal variation

```r
pivot_agg_data <- agg_merged %>%
  pivot_longer(cols = c(unemployment, crime_per_capita), names_to = "variable", values_to = "value")

# then make the plot

ggplot(pivot_agg_data, aes(x = year, y = value, colour = variable )) +
  geom_line() +
  labs(title = "aggragate crime and unemployment through time", x = "year", y = "variable") +
  scale_x_continuous(breaks = seq(1975, 2024, by = 5), limits = c(1972, 2026))
```



3.4 Visualize spatial variation

```
cbs_maps <- cbs_get_maps()
# the layout of the data.frame is:
str(cbs_maps)
```

```
## 'data.frame':    620 obs. of  4 variables:
##  $ region: chr  "arbeidsmarktregio" "arbeidsmarktregio" "arbeidsmarktregio" "arbeidsmarktregio" ...
##  $ year  : int  2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 ...
##  $ wgs84 : chr  "https://cartomap.github.io/nl/wgs84/arbeidsmarktregio_2014.geojson" "https://cartoma
##  $ rd    : chr  "https://cartomap.github.io/nl/rd/arbeidsmarktregio_2014.geojson" "https://cartomap.g
```

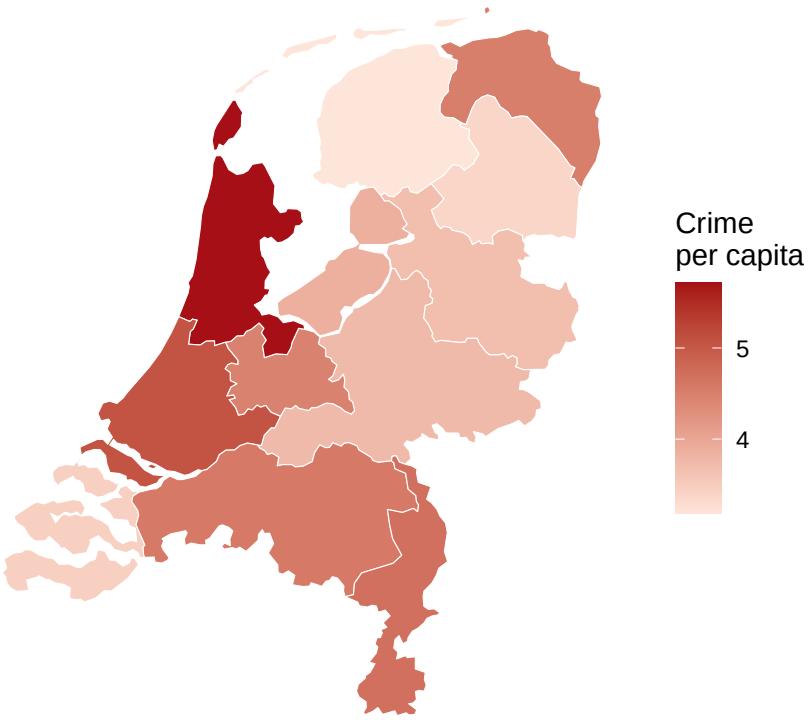
```
#read the GeoJSON straight from PDOK
nl_prov <- st_read(
  "https://cartomap.github.io/nl/wgs84/provincie_2014.geojson",
  quiet = TRUE
)
```

```
#Reading the CSV  →  2023 only
crime <- merged_panel_data_final |>
  filter(year == 2023) |>
  mutate(
    province = sub(" \\(PV\\)$", "", province),
```

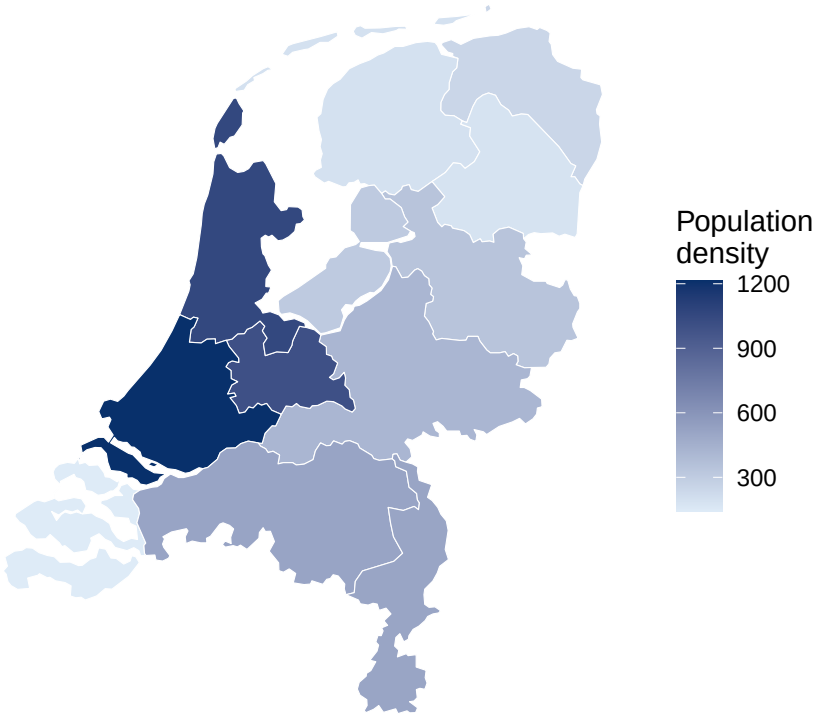
```
    province = recode(province, "Fryslân" = "Friesland")
  ) |>
  filter(!province %in% "Nederland")
```

```
# Province shapes (CBS 2014)
prov_shapes <- read_sf("https://cartomap.github.io/nl/wgs84/provincie_2014.geojson")


# Join data → map
nl_crime <- prov_shapes |>
  left_join(crime, by = c("statnaam" = "province"))
```

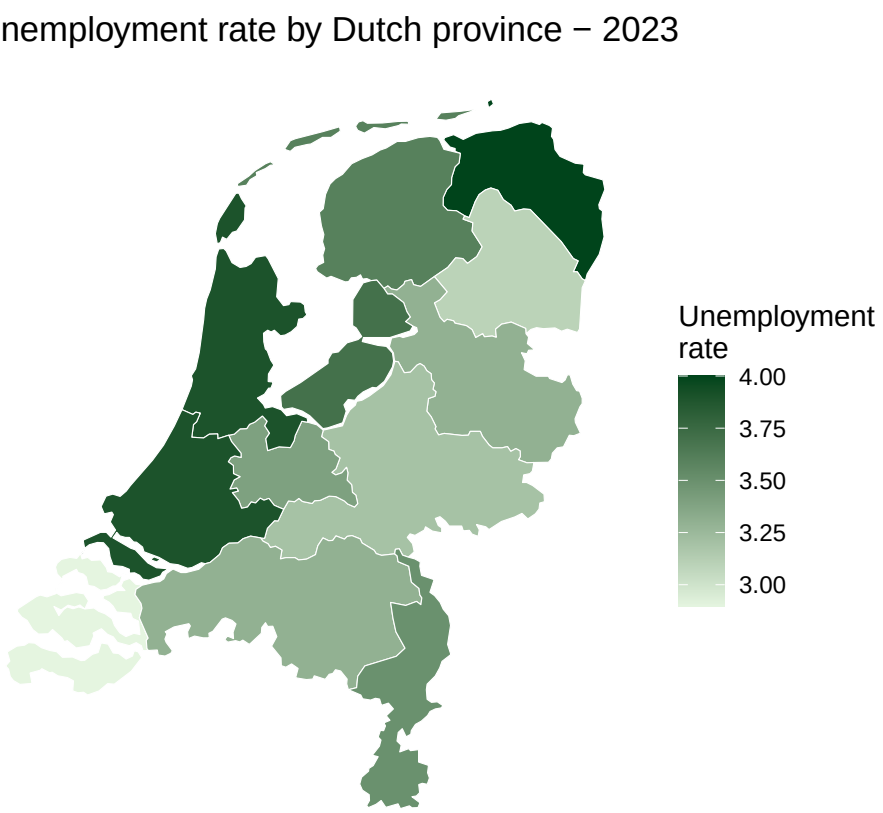
```
ggplot(nl_crime) +
  geom_sf(aes(fill = crimes_per_capita), color = "white", size = 0.2) +
  scale_fill_gradient(
    low  = "#fee5d9",  # pale red
    high = "#a50f15",  # dark red
    na.value = "grey90",
    name = "Crime\nper capita"
  ) +
  labs(
    title = "Crime per capita by Dutch province - 2023"
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.text  = element_blank(),
    axis.ticks = element_blank()
  )
```




```
ggplot(nl_crime) +
  geom_sf(aes(fill = poplation_density), color = "white", size = 0.2) +
  scale_fill_gradient(
    low  = "#deebf7",  # pale blue
    high = "#08306b",  # dark blue
    na.value = "grey90",
    name = "Population\ndensity"
  ) +
  labs(
    title = "Population density by Dutch province - 2023"
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.text  = element_blank(),
    axis.ticks = element_blank()
  )
```




```
ggplot(nl_crime) +
  geom_sf(aes(fill = unemployment_rate), color = "white", size = 0.2) +
  scale_fill_gradient(
    low  = "#e5f5e0",  # pale green
    high = "#00441b",  # dark green
    na.value = "grey90",
    name = "Unemployment\nrate"
  ) +
  labs(
    title = "Unemployment rate by Dutch province - 2023"
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    axis.text  = element_blank(),
    axis.ticks = element_blank()
  )
```

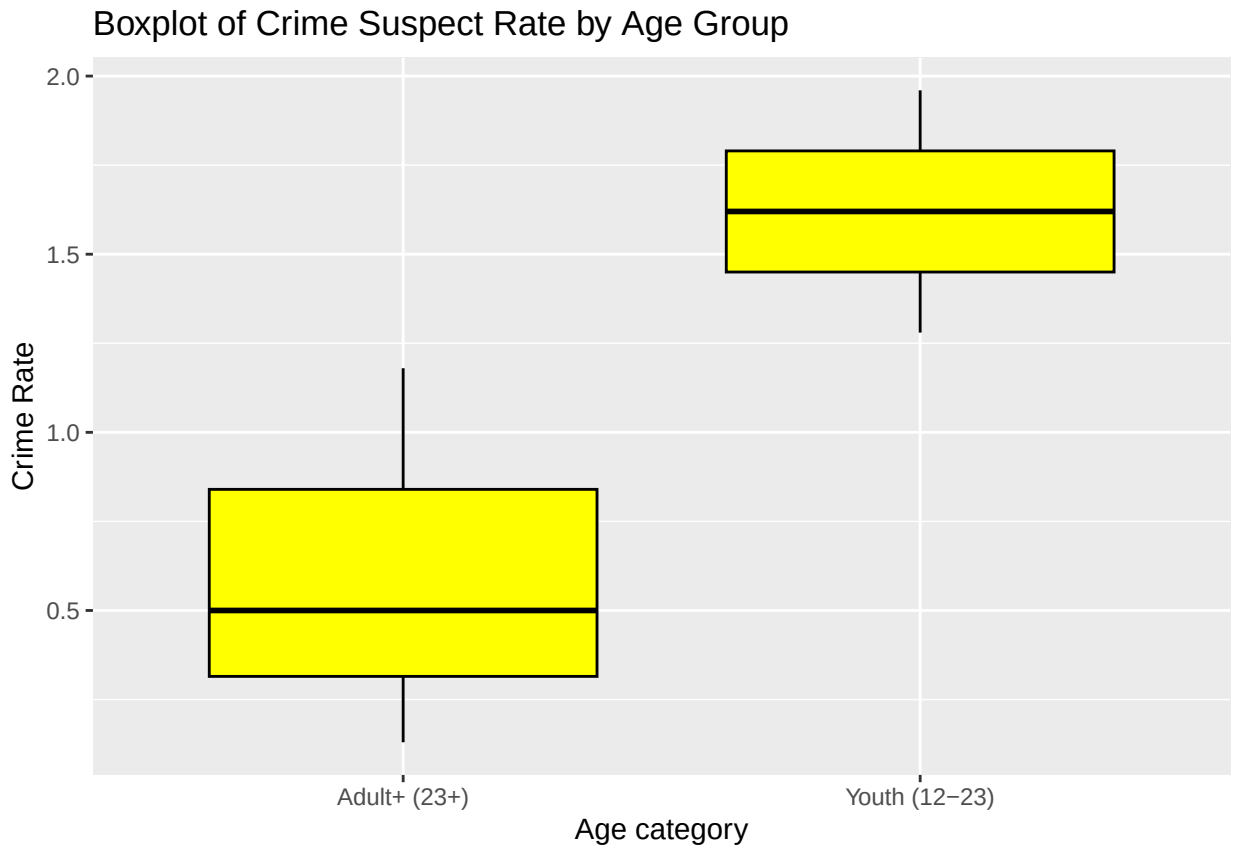


Here you provide a description of why the plot above is relevant to your specific social problem.

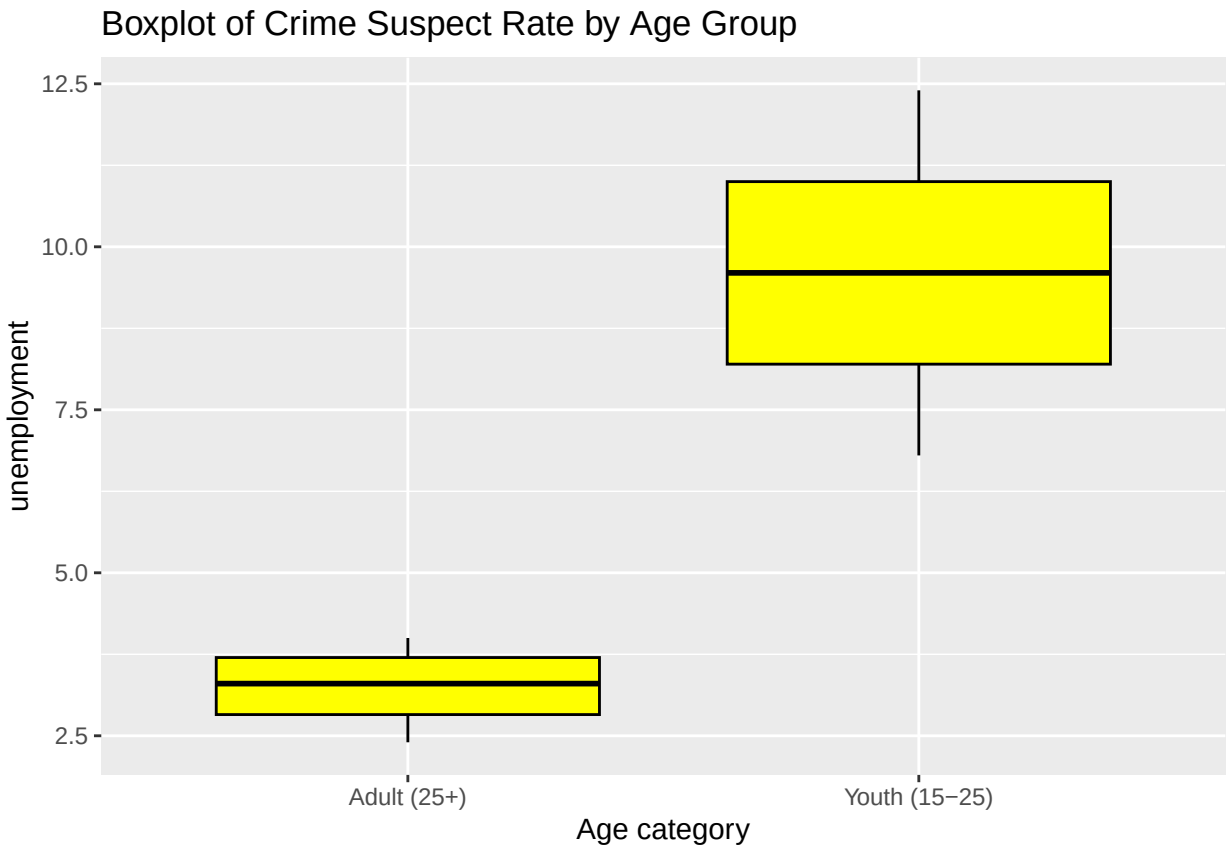
3.5 Visualize sub-population variation

What is the poverty rate by state?

```
# crime

ggplot(crime2021, aes(x = Age, y = Crime)) +
  geom_boxplot(fill = "yellow", color = "black") +
  labs(
    title = "Boxplot of Crime Suspect Rate by Age Group",
    x = "Age category",
    y = "Crime Rate"
  )
```




```
ggplot(unemployment_age, aes(x = Age, y = Unemployment)) +
  geom_boxplot(fill = "yellow", color = "black") +
  labs(
    title = "Boxplot of Crime Suspect Rate by Age Group",
    x = "Age category",
    y = "unemployment"
  )
```



Here you provide a description of why the plot above is relevant to your specific social problem.

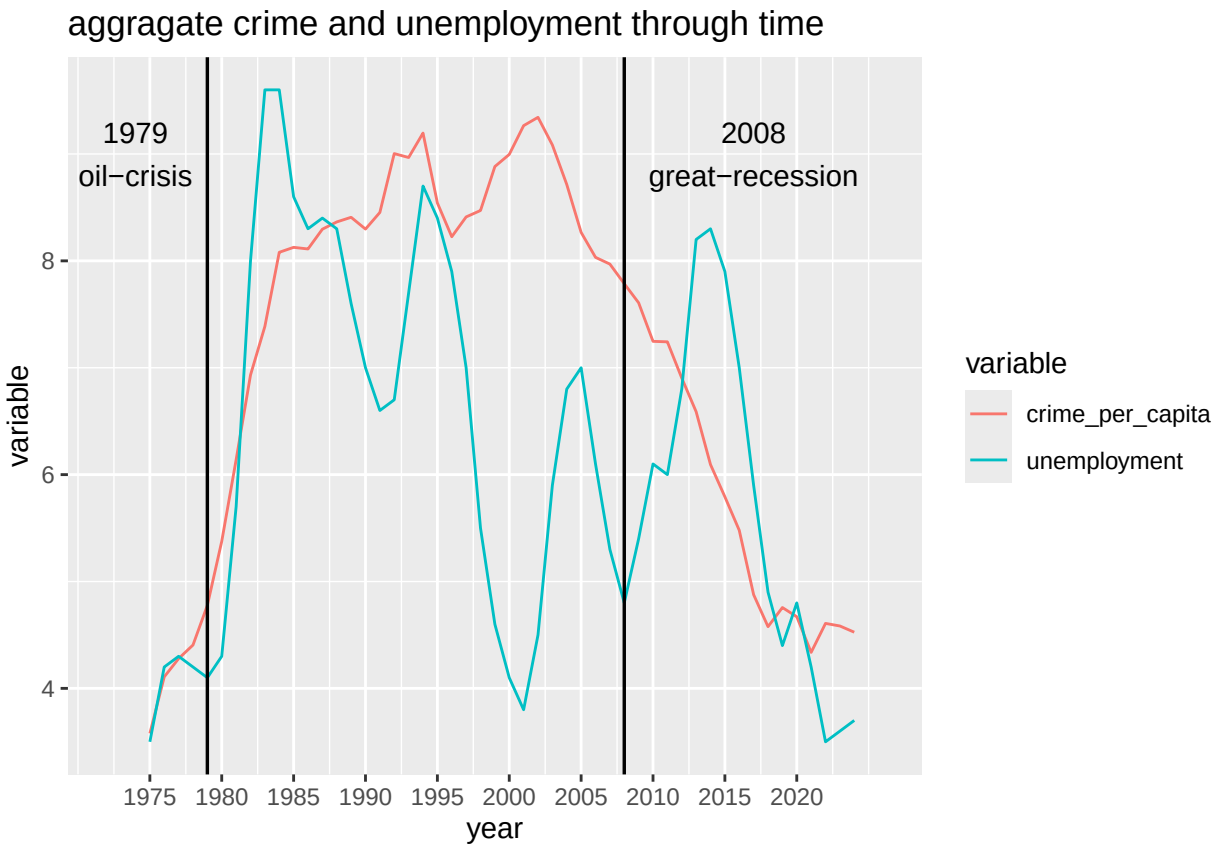
3.6 Event analysis

Analyze the relationship between two variables.

```
pivot_agg_data <- agg_merged %>%
  pivot_longer(cols = c(unemployment, crime_per_capita), names_to = "variable", values_to = "value")

# then make the plot

ggplot(pivot_agg_data, aes(x = year, y = value, colour = variable )) +
  geom_line() +
  labs(title = "aggragate crime and unemployment through time", x = "year", y = "variable") +
  scale_x_continuous(breaks = seq(1975, 2024, by = 5), limits = c(1972, 2026)) +
  geom_vline(xintercept = c(1979, 2008),
             linetype = "solid",
             color = "black",
             size = 0.6) +
  annotate("text", x = 1974, y = 9, label = "1979\noil-crisis") +
  annotate("text", x = 2017, y = 9, label = "2008\ngreat-recession")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
```

```
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Here you provide a description of why the plot above is relevant to your specific social problem.

Part 4 - Discussion

4.1 Discuss your findings

Part 5 - Reproducibility

5.1 Github repository link

Provide the link to your PUBLIC repository here: ...

5.2 Reference list

Use APA referencing throughout your document.