EDA-Base-Script

PSTAT 296A

2025-10-15

Import packages

```
library(tidyverse)
library(ggplot2)
library(dplyr)
library(tseries)
library(forecast)
library(ggfortify)
library(strucchange)
```

Background Research

(source: https://www.sciencedirect.com/science/article/pii/S0955395919300180)

Each wave is driven by a surge in popularity for a certain type of opioid substance, followed by restricter regulations and monitoring programs that drive death rates to go down a little, before the cycle repeats again. Each wave has a different target demographic and will be found to be more prevalent in a certain age group, geographic area, gender, etc.

Wave 1 (~ 1990s - 2010): Rise of prescription opioids including Oxycontin

Wave 2 (~ 2010 - 2013): Rise of heroin

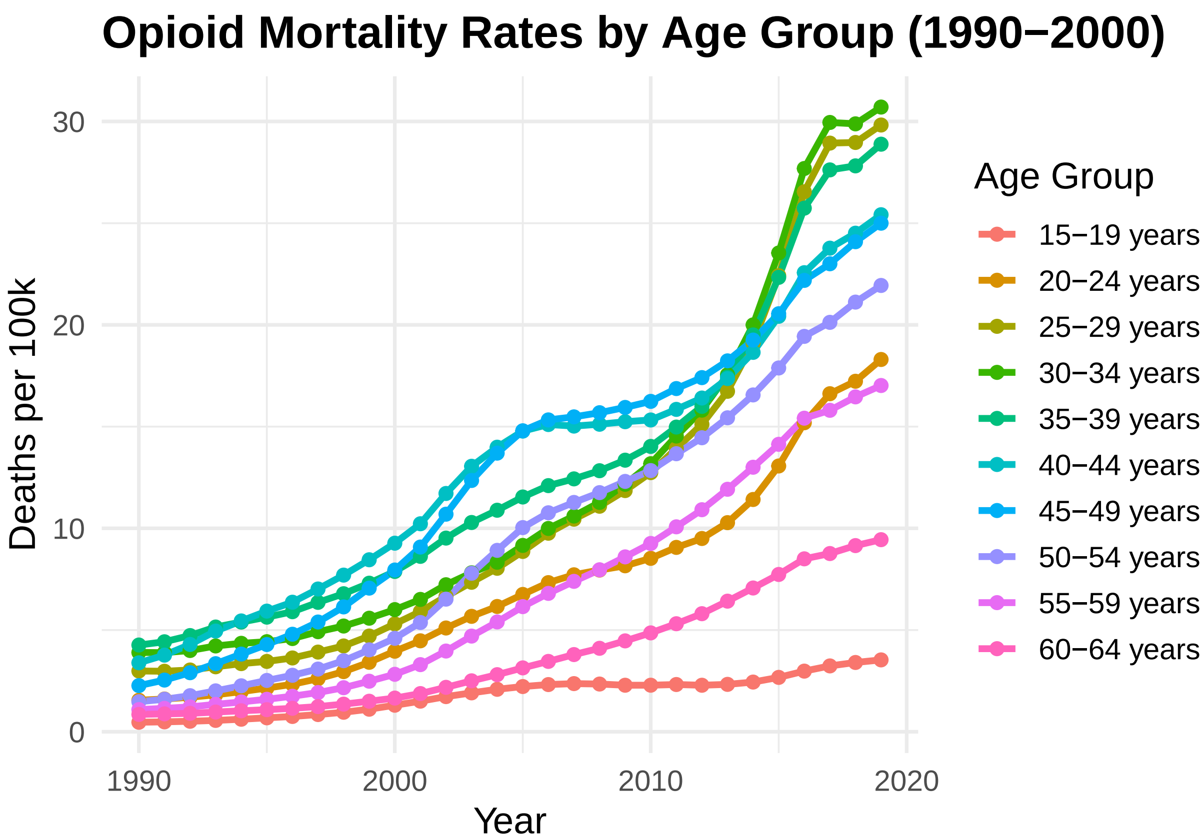
Wave 3 (~ 2013 - present): Rise of synthetic opioids like Fentanyl

EDA - Age

```
# Analyze age distribution (US opioid rate age.csv)
age_data <- read.csv("Data/US opioid rate age.csv")

# Check unique values of age bins
unique(age_data$age)
```

```
##  [1] "15-19 years" "20-24 years" "25-29 years" "30-34 years" "35-39 years"
##  [6] "40-44 years" "45-49 years" "50-54 years" "55-59 years" "60-64 years"
```

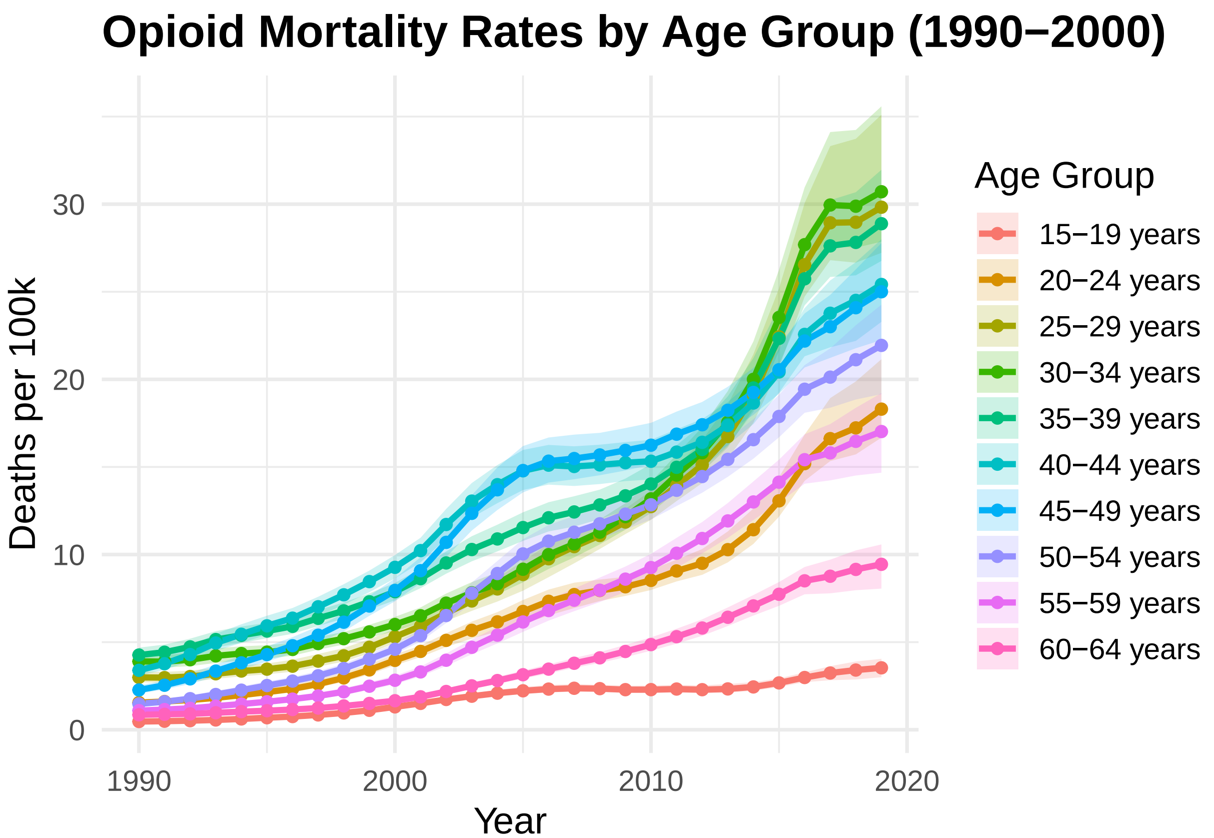
```r
# Plot the distribution of opioid death rates by age
ggplot(age_data, aes(x = year, y = val, color = age)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 1.8) +
  labs(
    title = "Opioid Mortality Rates by Age Group (1990-2000)",
    x = "Year",
    y = "Deaths per 100k",
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    plot.title = element_text(face = "bold")
  )
```



Ages from 25-39 years old show a similar trend with the highest opioid death rates. Overall, opioid death rate is trending up since 1990.

```r
# Add highlights around the confidence interval
ggplot(age_data, aes(x = year, y = val, color = age, fill = age)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1.1) +
  geom_point(size = 1.5) +
  labs(
    title = "Opioid Mortality Rates by Age Group (1990-2000)",
    x = "Year",
```

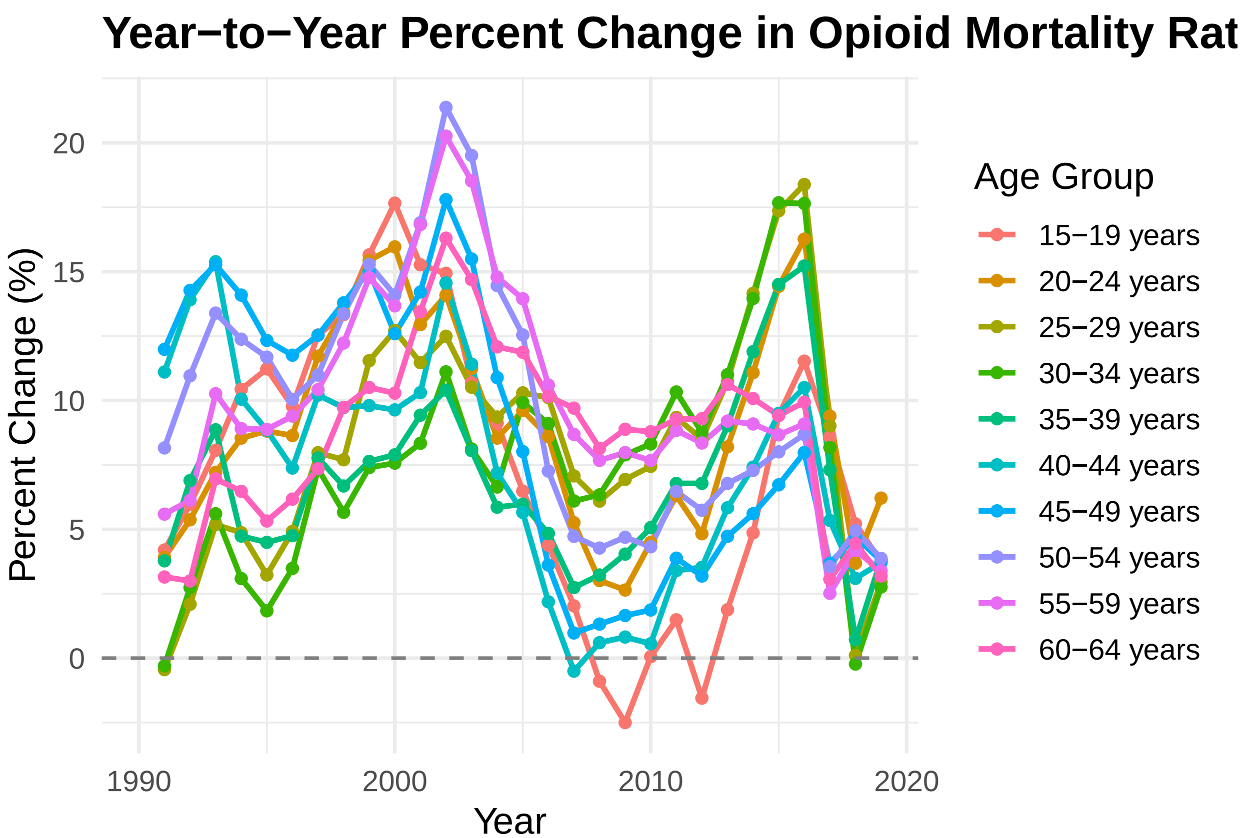
```
    y = "Deaths per 100k",
    color = "Age Group",
    fill = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    plot.title = element_text(face = "bold")
  )
```



There is much more variability for opioid death rates for ages 25-39, especially in the more recent years.

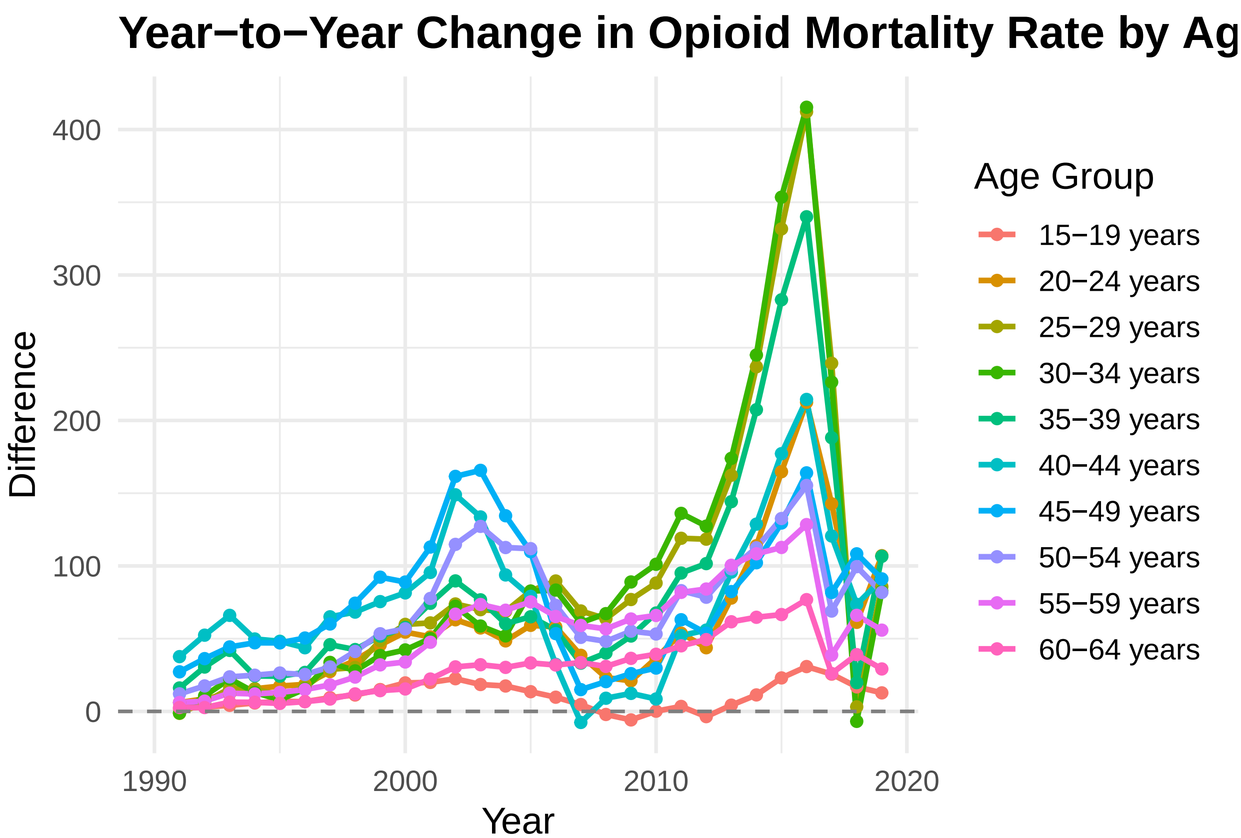
```
# Compute year-to-year percent change per age group
age_change <- age_data %>%
  group_by(age) %>%
  arrange(year, .by_group = TRUE) %>%
  mutate(
    pct_change = 100 * (val - lag(val)) / lag(val)
  ) %>%
  ungroup()

ggplot(age_change, aes(x = year, y = pct_change, color = age)) +
  geom_line(linewidth = 1) +
  geom_point(size = 1.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50") +
  labs(
    title = "Year-to-Year Percent Change in Opioid Mortality Rate by Age Group",
```

```r
    x = "Year",
    y = "Percent Change (%)",
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    plot.title = element_text(face = "bold")
  )
```




```r
# Compute year-to-year percent change per age group
age_diff <- age_data %>%
  group_by(age) %>%
  arrange(year, .by_group = TRUE) %>%
  mutate(
    pct_change = 100 * (val - lag(val))
  ) %>%
  ungroup()

ggplot(age_diff, aes(x = year, y = pct_change, color = age)) +
  geom_line(linewidth = 1) +
  geom_point(size = 1.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50") +
  labs(
    title = "Year-to-Year Change in Opioid Mortality Rate by Age Group",
    x = "Year",
    y = "Difference",
```

```r
    color = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    plot.title = element_text(face = "bold")
  )
```



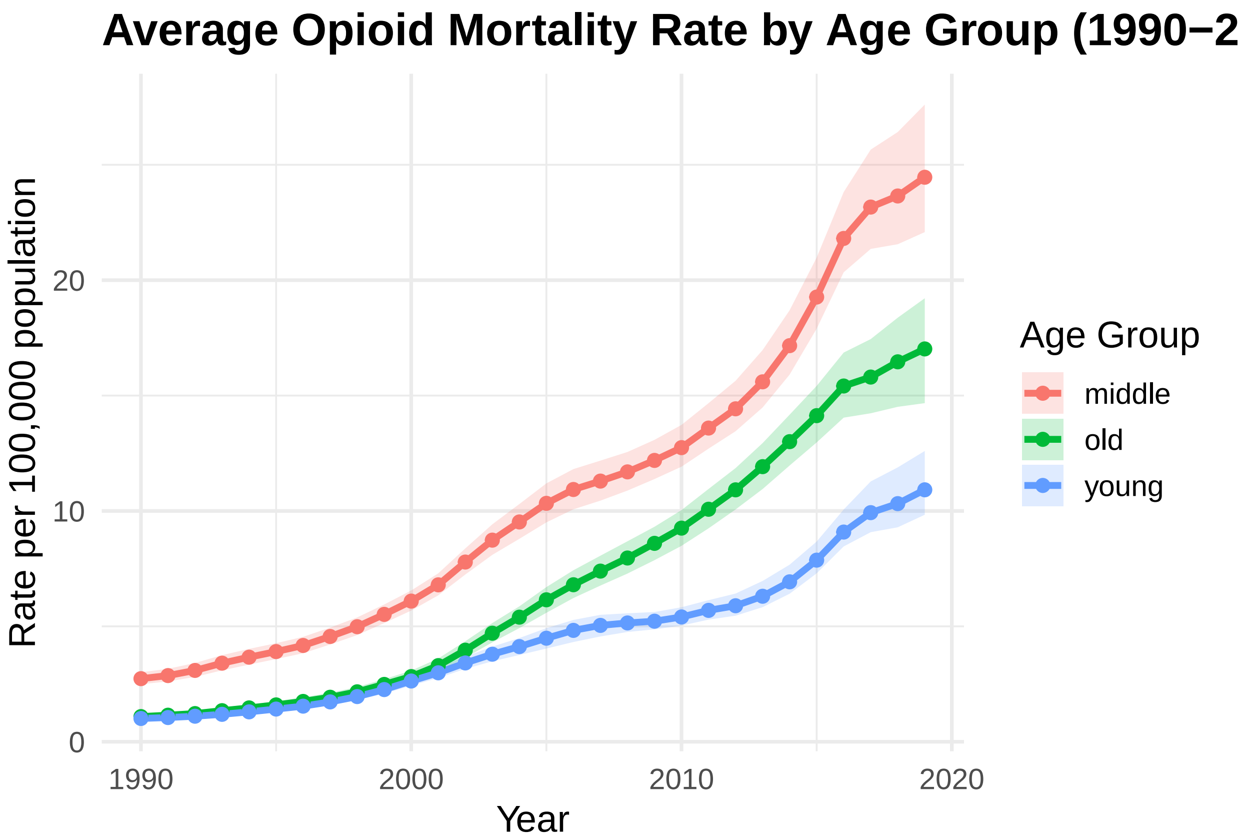

```r
# Group the age buckets into young, middle, and old.
age_data <- age_data %>%
  mutate(
    age_group = case_when(
      age %in% c("15-19 years", "20-24 years") ~ "young",
      age %in% c("55-59 years", "59-64 years") ~ "old",
      TRUE ~ "middle"
    )
  )
head(age_data)
```

```
##   measure                     location  sex       age                 cause metric
## 1  Deaths United States of America Both 15-19 years Opioid use disorders   Rate
## 2  Deaths United States of America Both 20-24 years Opioid use disorders   Rate
## 3  Deaths United States of America Both 25-29 years Opioid use disorders   Rate
## 4  Deaths United States of America Both 30-34 years Opioid use disorders   Rate
## 5  Deaths United States of America Both 35-39 years Opioid use disorders   Rate
## 6  Deaths United States of America Both 40-44 years Opioid use disorders   Rate
```

```
##   year       val     upper     lower age_group
## 1 1990 0.4670483 0.5106493 0.4277849     young
## 2 1990 1.5493897 1.7051303 1.4148146     young
## 3 1990 2.9873181 3.3271826 2.7056838    middle
## 4 1990 3.8990845 4.2795801 3.5592602    middle
## 5 1990 4.2629531 4.6842460 3.8918969    middle
## 6 1990 3.3901412 3.7341517 3.0834375    middle
```

The only limitation to this approach is that rates are given in deaths per 100k, thus that when age groups are aggregated it makes the assumption that each age group has the same number of people, which is obviously not true. If population data was available, rescaling can be done to give weights to mortality rates for each age group

```r
age_data %>%
  group_by(year, age_group) %>%
  summarise(
    middle_rate = mean(val, na.rm = TRUE),
    low_rate = mean(lower, na.rm = TRUE),
    high_rate = mean(upper, na.rm = TRUE)
  ) %>%
  ggplot(aes(x = year, y = middle_rate, color = age_group, fill = age_group)) +
  geom_ribbon(aes(ymin = low_rate, ymax = high_rate), alpha = 0.2, color = NA) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 1.8) +
  labs(
    title = "Average Opioid Mortality Rate by Age Group (1990-2000)",
    x = "Year",
    y = "Rate per 100,000 population",
    color = "Age Group",
    fill = "Age Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    plot.title = element_text(face = "bold")
  )
```



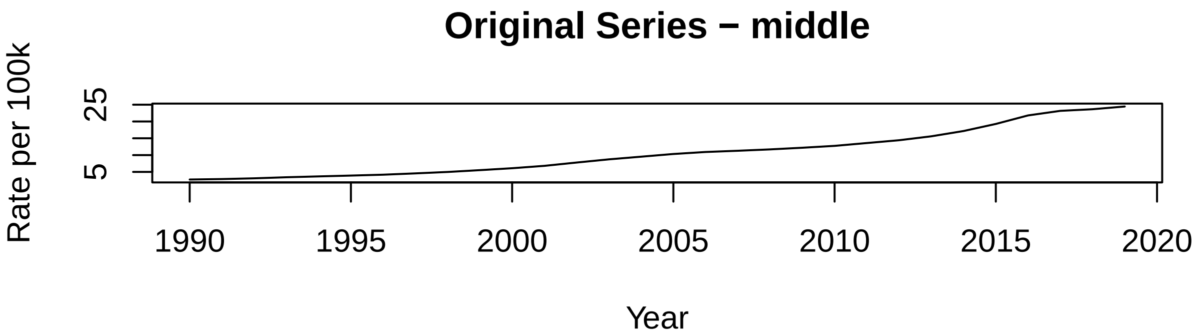

```r
age_grouped <- age_data %>%
  group_by(year, age_group) %>%
  summarise(
    middle_rate = mean(val, na.rm = TRUE),
    low_rate = mean(lower, na.rm = TRUE),
    high_rate = mean(upper, na.rm = TRUE)
  ) %>%
  ungroup()
```

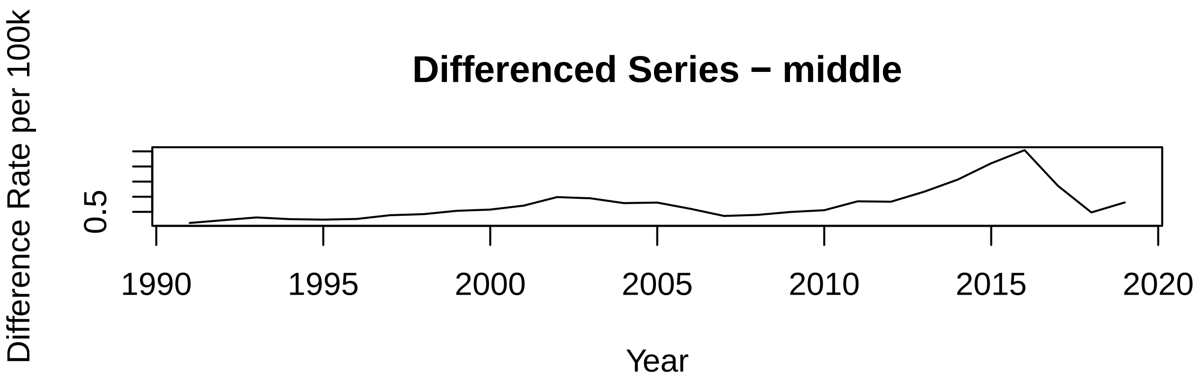
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
library(tidyverse)
library(forecast)
library(tseries)

# Loop through each age group
for (group in unique(age_grouped$age_group)) {

  cat("\n--------------------------------------\n")
  cat("Results for:", group, "\n")

  # Subset for that group
  sub <- age_grouped %>%
    filter(age_group == group) %>%
    arrange(year)
```

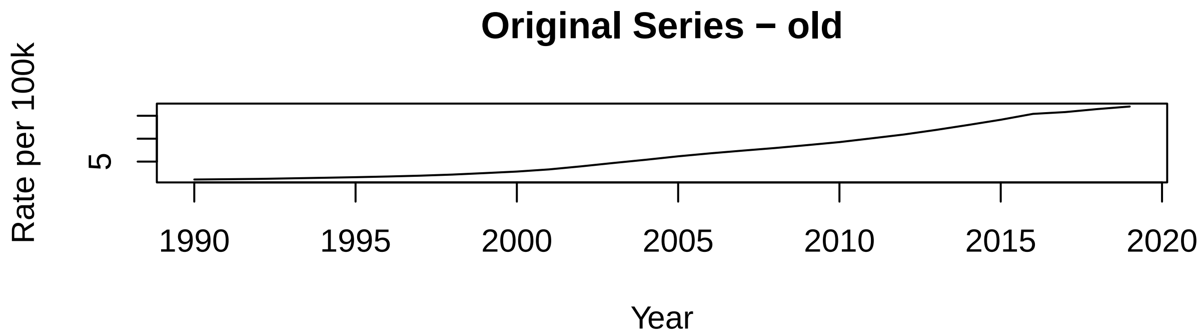
```r
  # Create time series
  ts_obj <- ts(sub$middle_rate, start = min(sub$year), frequency = 1)

  # Run ADF test on original
  adf_orig <- adf.test(ts_obj)
  cat("ADF (original series) p-value:", round(adf_orig$p.value, 4), "\n")

  # Difference the series
  ts_diff <- diff(ts_obj)

  # Run ADF test on differenced series
  adf_diff <- adf.test(ts_diff)
  cat("ADF (differenced series) p-value:", round(adf_diff$p.value, 4), "\n")

  # Optional: Plot both series
  par(mfrow = c(2, 1))  # two plots per group
  plot(ts_obj, main = paste("Original Series -", group),
       ylab = "Rate per 100k", xlab = "Year")
  plot(ts_diff, main = paste("Differenced Series -", group),
       ylab = "Difference Rate per 100k", xlab = "Year")

  cat("---------------------------------------\n")
}
```

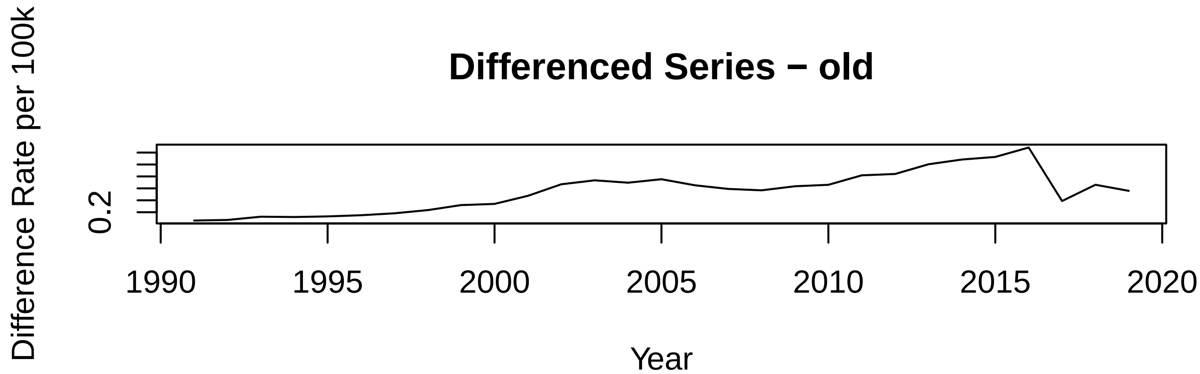
```
##
## ---------------------------------------
## Results for: middle
## ADF (original series) p-value: 0.9784
## ADF (differenced series) p-value: 0.2848
```



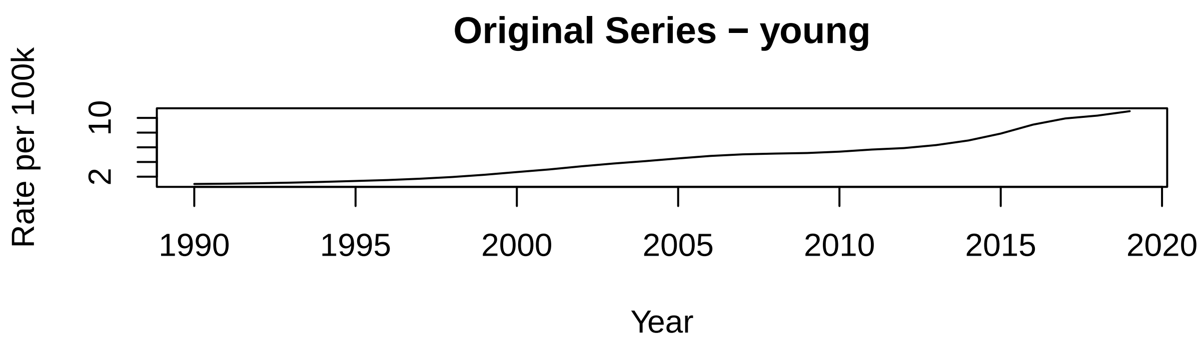


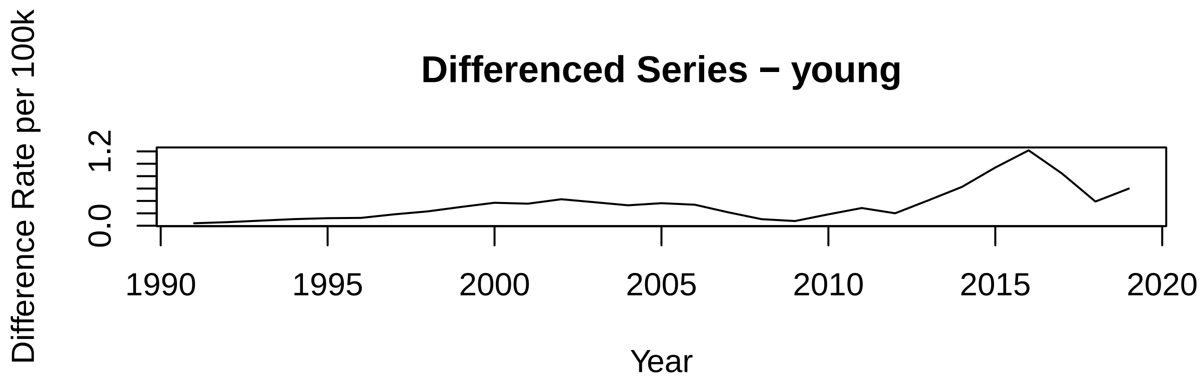

```
## -------------------------------------
##
## -------------------------------------
## Results for: old
## ADF (original series) p-value: 0.5482
## ADF (differenced series) p-value: 0.2194
```






```
## --------------------------------------
##
## --------------------------------------
## Results for: young
## ADF (original series) p-value: 0.9749
## ADF (differenced series) p-value: 0.5816
```






```
## --------------------------------------
```

```r
breakpoints(middle_rate ~ 1, data = filter(age_grouped, age_group == "middle"))
```

```
##
##   Optimal 6-segment partition:
##
## Call:
## breakpoints.formula(formula = middle_rate ~ 1, data = filter(age_grouped,
##     age_group == "middle"))
##
## Breakpoints at observation number:
## 8 12 16 22 26
##
## Corresponding to breakdates:
## 0.2666667 0.4 0.5333333 0.7333333 0.8666667
```

```r
breakpoints(middle_rate ~ 1, data = filter(age_grouped, age_group == "old"))
```

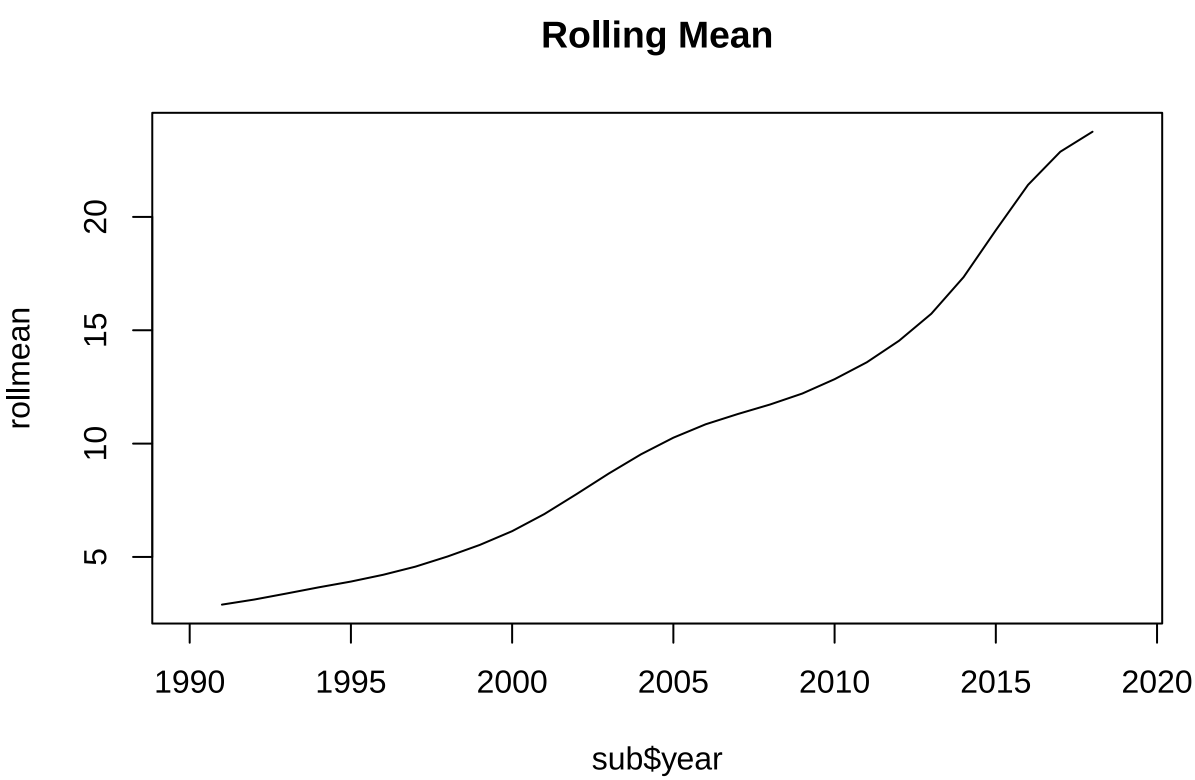
```
##
##   Optimal 6-segment partition:
##
## Call:
## breakpoints.formula(formula = middle_rate ~ 1, data = filter(age_grouped,
##     age_group == "old"))
##
## Breakpoints at observation number:
## 10 14 18 22 26
```

```
##
## Corresponding to breakdates:
## 0.3333333 0.4666667 0.6 0.7333333 0.8666667
```

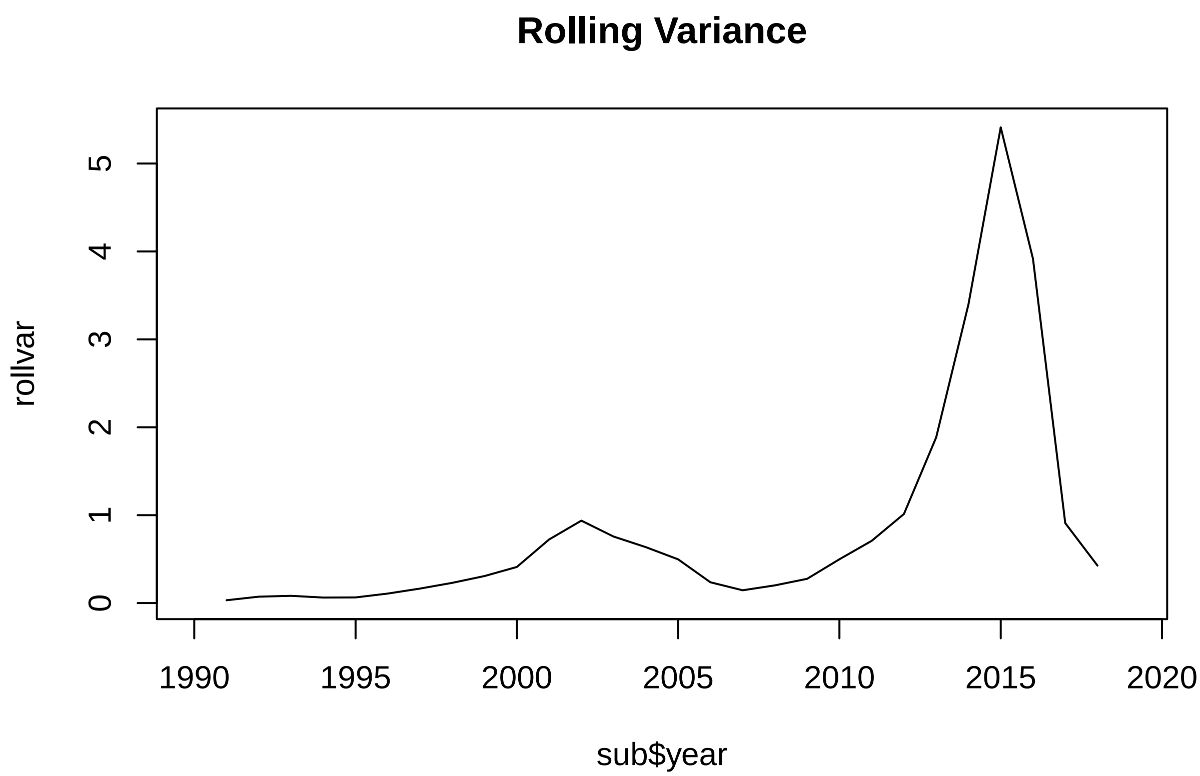
```r
breakpoints(middle_rate ~ 1, data = filter(age_grouped, age_group == "young"))
```

```
##
##   Optimal 6-segment partition:
##
## Call:
## breakpoints.formula(formula = middle_rate ~ 1, data = filter(age_grouped,
##     age_group == "young"))
##
## Breakpoints at observation number:
## 8 12 16 22 26
##
## Corresponding to breakdates:
## 0.2666667 0.4 0.5333333 0.7333333 0.8666667
```

```r
library(zoo)
sub <- filter(age_grouped, age_group == "middle")
rollmean <- rollapply(sub$middle_rate, 3, mean, fill = NA)
rollvar  <- rollapply(sub$middle_rate, 3, var, fill = NA)
plot(sub$year, rollmean, type = "l", main = "Rolling Mean")
```




```r
plot(sub$year, rollvar, type = "l", main = "Rolling Variance")
```


Rolling Variance
rollvar
0
1
2
3
4
5
1990
1995
2000
2005
2010
2015
2020
sub$year