DATA 624 Homework 2

Fomba Kassoh

2024-14-03

{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)

Load Required Libraries

```
options(repos = c(CRAN = "https://cloud.r-project.org"))

# Function to install a package if not already installed
install_if_needed <- function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg)

  }
}

# List of packages to check and install if necessary
required_packages <- c("fpp3", "dplyr", "ggplot2", "lubridate", "tsibble",
                       "tsibbledata", "feasts", "fable", "fabletools",
                       "curl", "USgas", "readxl", "readr", "tidyr", "forecast",
                       "seasonal", "patchwork", "LaTeX")

# Loop through the list and install packages only if needed
for (pkg in required_packages) {
  install_if_needed(pkg)
}
```

```
## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Installing package into 'C:/Users/RemoteUser/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)

## Warning: package 'LaTeX' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```r
# Function to suppress package startup messages
suppressPackageStartupMessages({
  library(fpp3)
  library(dplyr)
  library(ggplot2)
  library(lubridate)
  library(tsibble)
  library(tsibbledata)
  library(feasts)
  library(fable)
  library(fabletools)
  library(readr)
  library(readxl)
  library(tidyr)
  library(forecast)
  library(seasonal)
  library(patchwork)
})
```

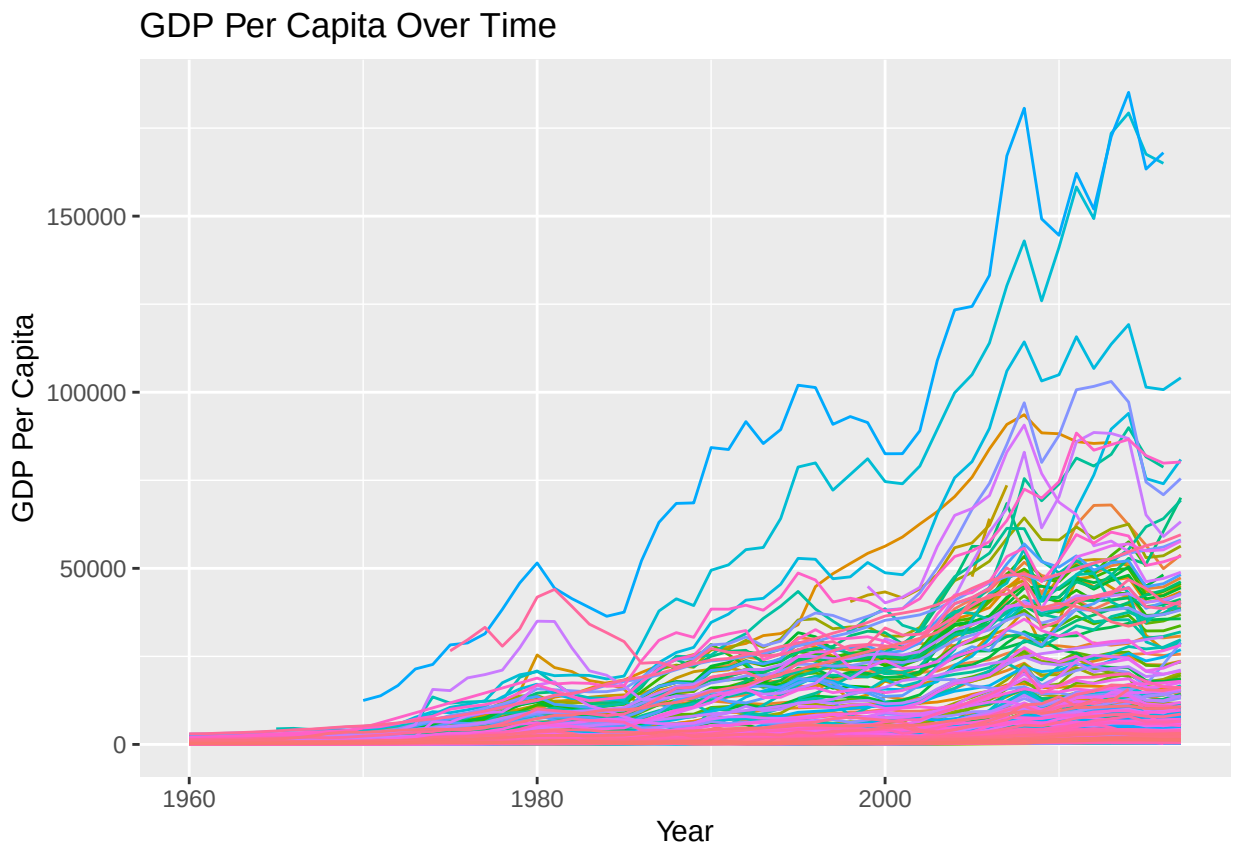
Exercise 3.7.1

Plot the GDP per capita for each country over time. Which country has the highest GDP per capita? How has this changed over time?

Plot the GDP per capita for all countries

Below are the plots.

```r
global_economy |>
  # 1. Filter rows where both GDP and Population are available (no missing values)
  filter(!is.na(GDP), !is.na(Population)) |>

  # 2. Create a new column 'GDP_per_capita' by dividing GDP by Population
  mutate(GDP_per_capita = GDP / Population) |>

  # 3. Plot the 'GDP_per_capita' over time using autoplot
  autoplot(GDP_per_capita) +

  # 4. Customize the plot labels:
  labs(title = "GDP Per Capita Over Time",
       x = "Year",
       y = "GDP Per Capita") +

  # 5. Remove the legend from the plot (as there are too many for this plot to display)
  theme(legend.position = "none")
```



What's the country with highest GDP per Capita

The country with the highest `GDP per capital` is Monaco in `Year` 2014

```
# 1. Filter for Non-Missing Values
# Remove rows where GDP or Population is NA (missing data)
highest_gdp_per_capita <- global_economy |>
  filter(!is.na(GDP), !is.na(Population)) |>

# 2. Calculate GDP per Capita
# Create a new column 'GDP_per_capita' by dividing GDP by Population
mutate(GDP_per_capita = GDP / Population) |>

# 3. Filter for Maximum GDP per Capita
# Keep only the row(s) where GDP per capita is the highest
filter(GDP_per_capita == max(GDP_per_capita, na.rm = TRUE)) |>

# 4. Select Relevant Columns
# Select only Country, Year, and GDP_per_capita columns for output
select(Country, Year, GDP_per_capita)

# 5. Display the Result
highest_gdp_per_capita
```

```
## # A tsibble: 1 x 3 [1Y]
```

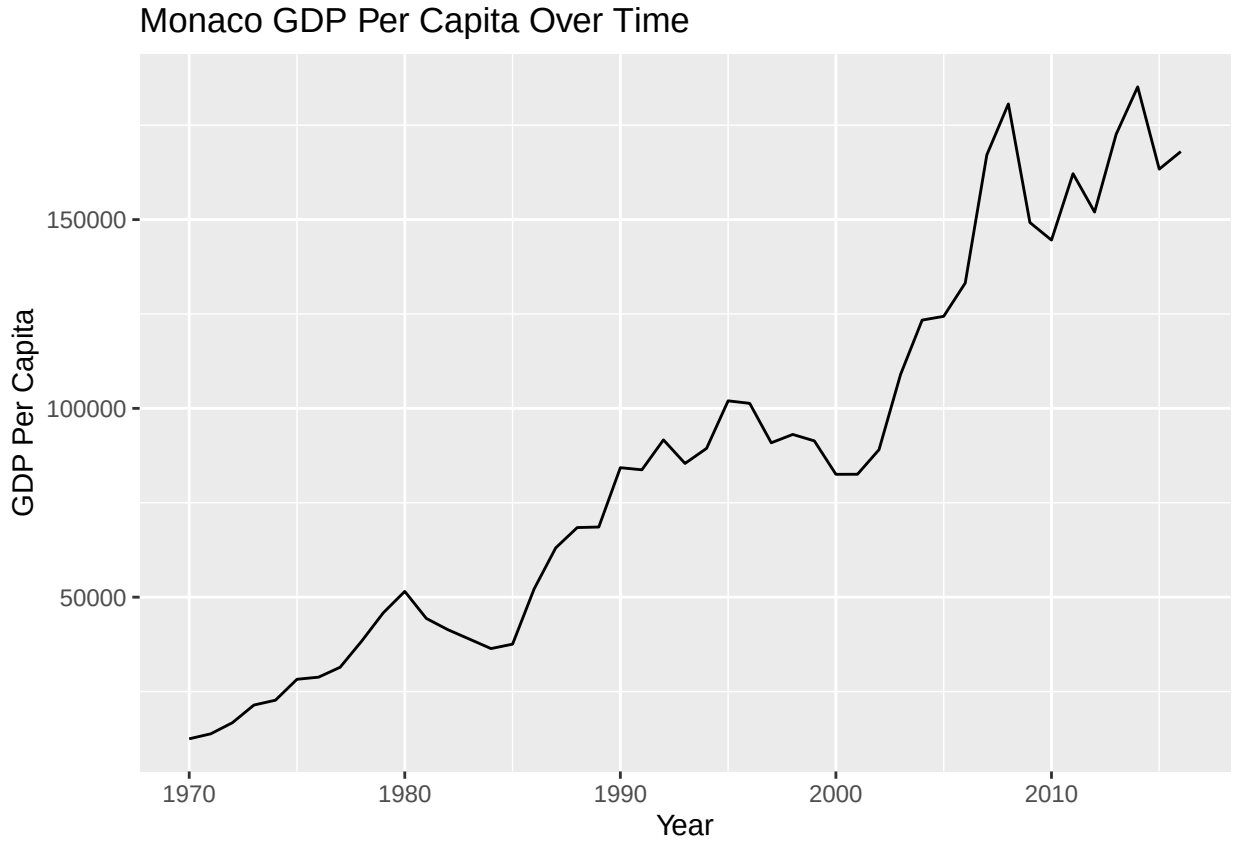
```
## # Key:       Country [1]
##   Country  Year GDP_per_capita
##   <fct>   <dbl>          <dbl>
## 1 Monaco   2014        185153.
```

How has Monaco's GDP per Capita changed over time

Monaco's GDP per capita has seen notable changes over time.

1. From the 1970s to the early 1980s, there was steady, moderate growth.
2. In the mid-1980s to 1990s, the GDP per capita stabilized with minor fluctuations.
3. A dip in the late 1990s was followed by a rapid rise in the early 2000s, continuing through the decade.
4. Between 2000 and 2010, GDP per capita experienced the most significant growth.
5. In the 2010s, some fluctuations occurred, but the overall GDP per capita remained high compared to earlier years.

```
global_economy |>
  filter(!is.na(GDP), !is.na(Population)) |>
  filter(Country == "Monaco") |>
  mutate(GDP_per_capita = GDP / Population) |>
  autoplot(GDP_per_capita)+
  labs(title = "Monaco GDP Per Capita Over Time",
       x = "Year",
       y = "GDP Per Capita")
```



Exercise 3.7.2 For each of the following series, make a graph of the data. If transforming seems appropriate, do so and describe the effect.

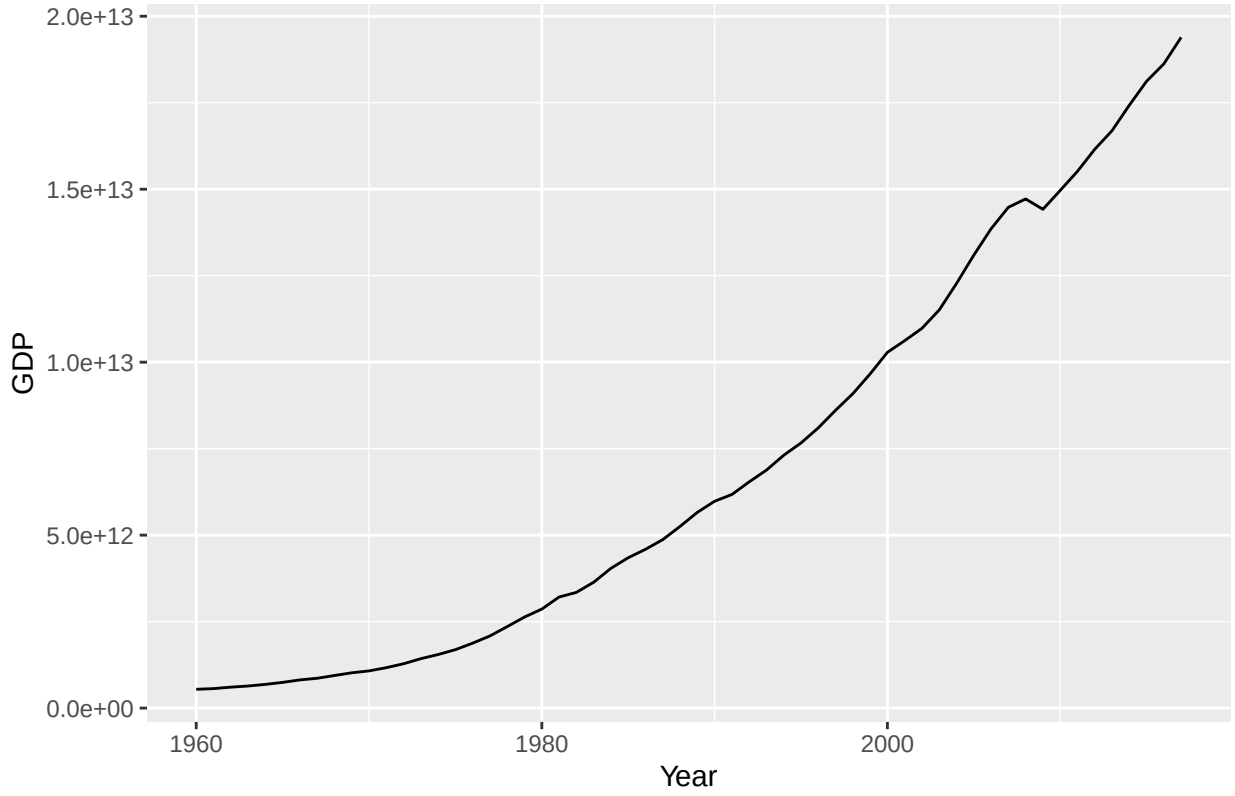
United States GDP from global_economy:**

The graph shows the United States GDP over time, and it appears to exhibit exponential growth.The data does not have constant mean or variance over time, and it is non-stationary. The Box-Cox and logarithmic transforms are appropriate.

As shown in Figure 2, the Box-Cox transformation has a greater effect in terms of linearizing the data over time. This makes it more suitable for modeling purposes.

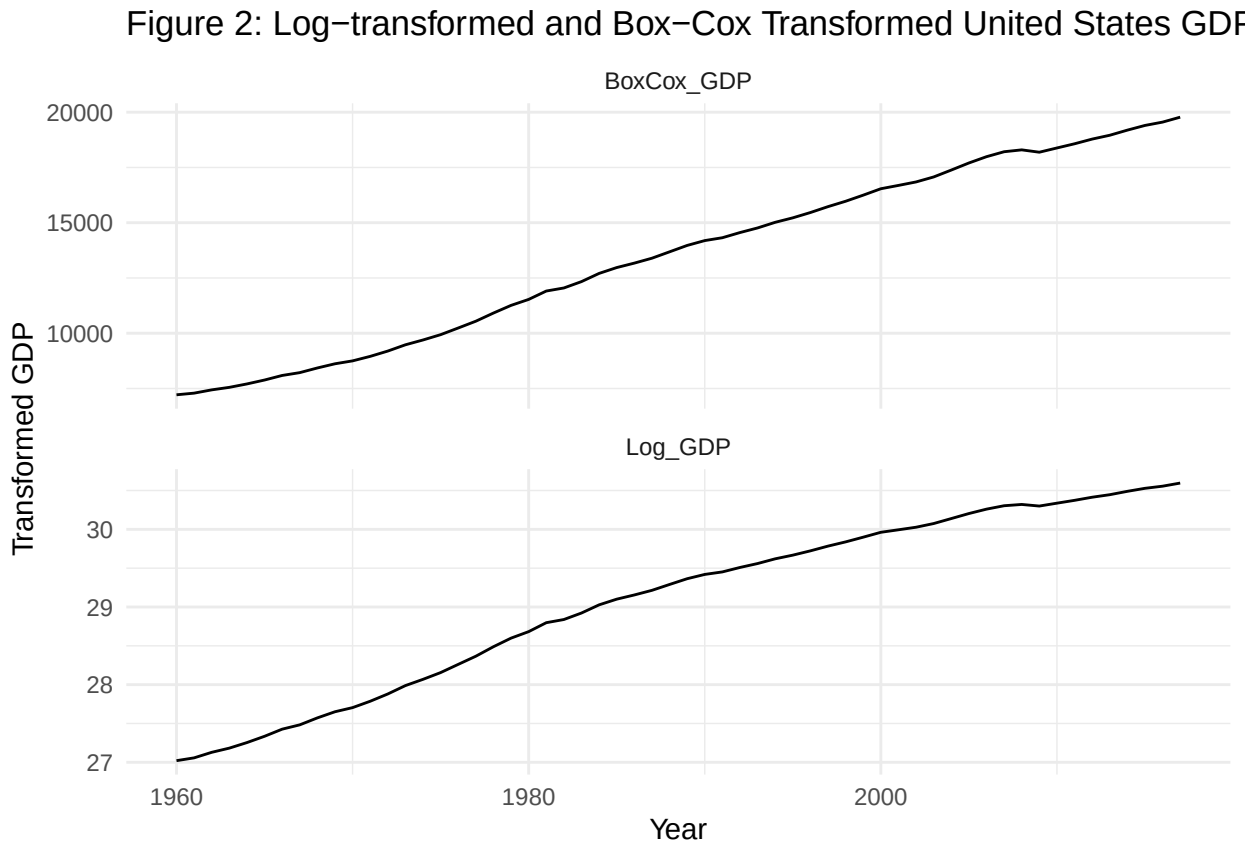
```
# Step-by-step explanation:

us_gdp <- global_economy |>
  # 1. Filter rows where both GDP and Population are available (no missing values)
  filter(!is.na(GDP), !is.na(Population)) |>

  # 2. Further filter the data to only include rows where Country is "United States"
  filter(Country == "United States") |>

  # 3. Select only the Country, Year, and GDP columns for the United States
  select(Country, Year, GDP)

# Plot the GDP of the United States over time
us_gdp |>
  # 4. Plot the GDP over time using autoplot
  autoplot(GDP) +

  # 5. Customize the plot labels:
  labs(title = "Figure 1: United States GDP Over Time",  # Add a title
       x = "Year",
       y = "GDP")
```

Figure 1: United States GDP Over Time




```
# Step 1: Log Transformation and Plot
log_gdp_plot <- ggplot(us_gdp, aes(x = Year, y = log(GDP))) +

# Plot the log-transformed GDP as a line chart
geom_line() +

# Add a title and axis labels to the plot
labs(title = "Log-transformed United States GDP Over Time",  # The plot title
      x = "Year",  # Label the x-axis as "Year"
      y = "Log of GDP")  # Label the y-axis as "Log of GDP"


# Step 2: Find the optimal lambda for Box-Cox transformation using Guerrero method
lambda <- us_gdp |>
  features(GDP, features = guerrero) |>
  pull(lambda_guerrero)

# Step 3: Apply the Box-Cox transformation using the optimal lambda
us_gdp <- us_gdp |>
  mutate(BoxCox_GDP = box_cox(GDP, lambda))

# Step 4: Box-Cox Transformed Plot
boxcox_gdp_plot <- us_gdp |>
  autoplot(BoxCox_GDP) +
  labs(y = "Box-Cox Transformed GDP",
       x = "Year",
```

```
       title = latex2exp::TeX(paste0(
         "Box-Cox Transformed GDP"))) +
  theme_minimal()

# Step 5: Combine both plots using facet_wrap
us_gdp_long <- us_gdp |>
  mutate(Log_GDP = log(GDP)) |>
  pivot_longer(cols = c("BoxCox_GDP", "Log_GDP"),
               names_to = "Transformation",
               values_to = "Value")

ggplot(us_gdp_long, aes(x = Year, y = Value)) +
  geom_line() +
  facet_wrap(~ Transformation, scales = "free_y", ncol = 1) +
  labs(title = "Figure 2: Log-transformed and Box-Cox Transformed United States GDP",
       x = "Year",
       y = "Transformed GDP") +
  theme_minimal()
```



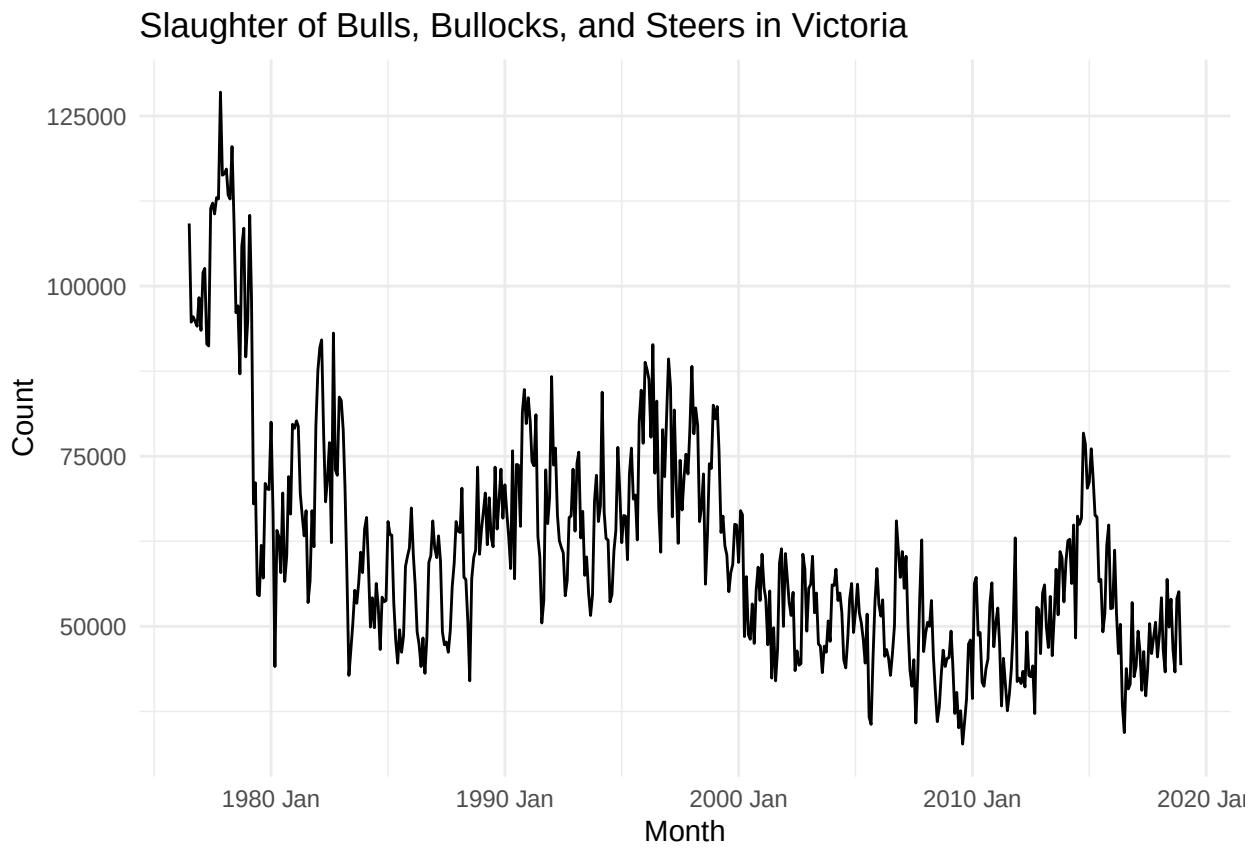
Slaughter of Victorian "Bulls, bullocks and steers" in aus_livestock

As shown in the second plot, the transformations (Box-Cox and Log) applied to the `Count` data while they stabilized the variance (especially in the 1980s) are not appropriate because they:

1. Did not result in a clearly linear relationship.

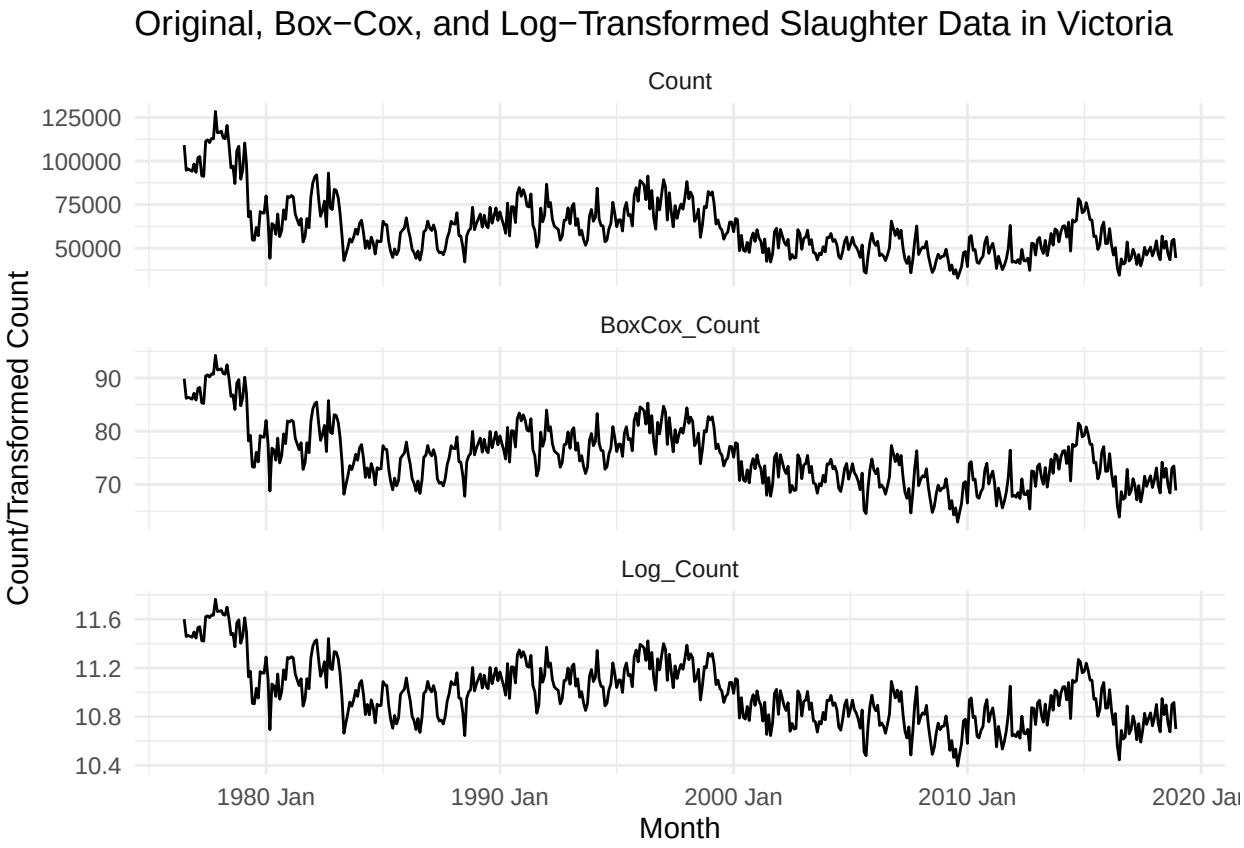
2. Does not support a graphical interpretation.

```
# Step 1: Filter the data for Victoria and Bulls, bullocks and steers
victoria_bulls <- aus_livestock %>%
  filter(State == "Victoria", Animal == "Bulls, bullocks and steers")

# Step 2: Plot the graph for the filtered data
ggplot(victoria_bulls, aes(x = Month, y = Count)) +
  geom_line() +
  labs(title = "Slaughter of Bulls, Bullocks, and Steers in Victoria",
       x = "Month",
       y = "Count") +
  theme_minimal()
```




```
# Step 1: Ensure the Count is listed first in the transformation order
victoria_bulls_long <- victoria_bulls |>
  mutate(BoxCox_Count = box_cox(Count, lambda),
         Log_Count = log(Count)) |>
  pivot_longer(cols = c("Count", "BoxCox_Count", "Log_Count"),  # Place "Count" first
               names_to = "Transformation",
               values_to = "Value") |>
  mutate(Transformation = factor(Transformation, levels = c("Count", "BoxCox_Count", "Log_Count")))  # A

# Step 2: Plot with Count at the top using facet_wrap
ggplot(victoria_bulls_long, aes(x = Month, y = Value)) +
```

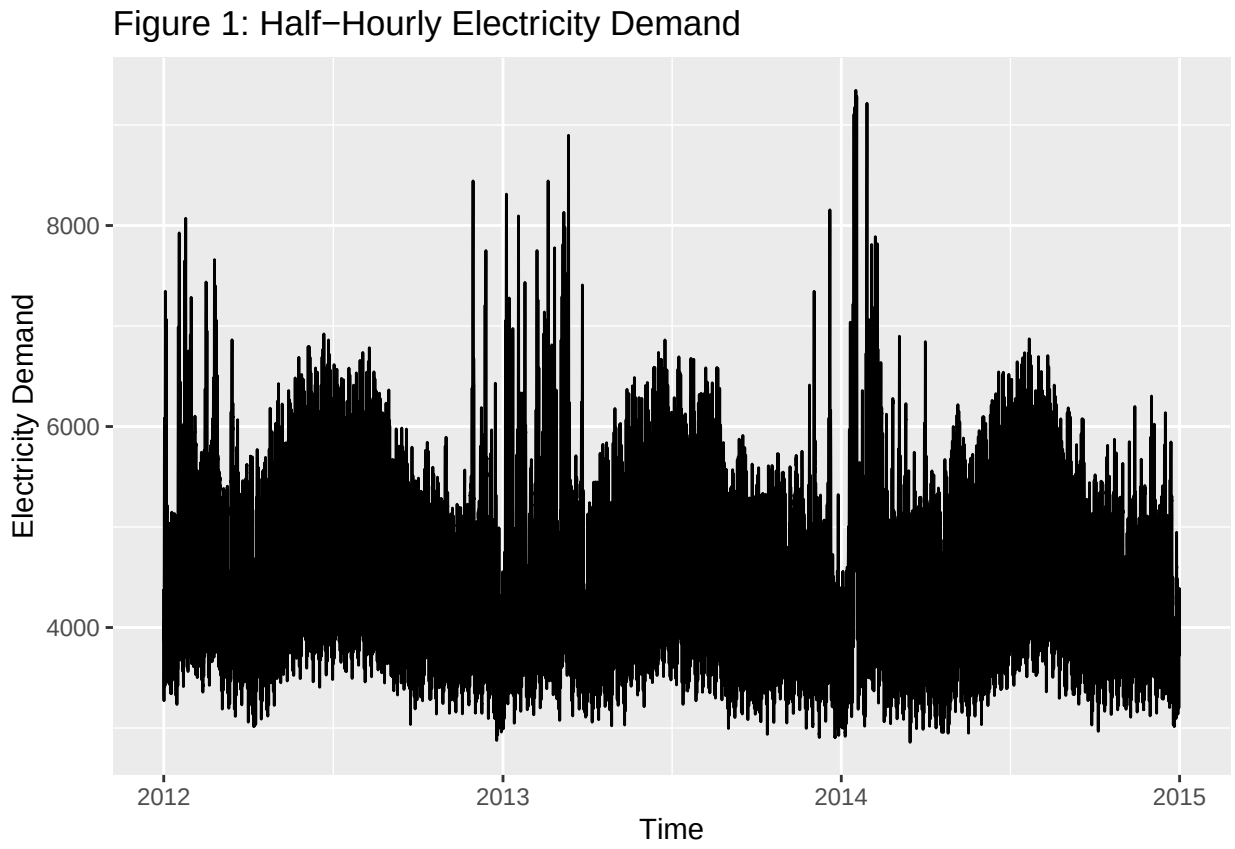
```
  geom_line() +
  facet_wrap(~ Transformation, scales = "free_y", ncol = 1) +  # Use facet wrap to show original, Box-C
  labs(title = "Original, Box-Cox, and Log-Transformed Slaughter Data in Victoria",
       x = "Month",
       y = "Count/Transformed Count") +
  theme_minimal()
```



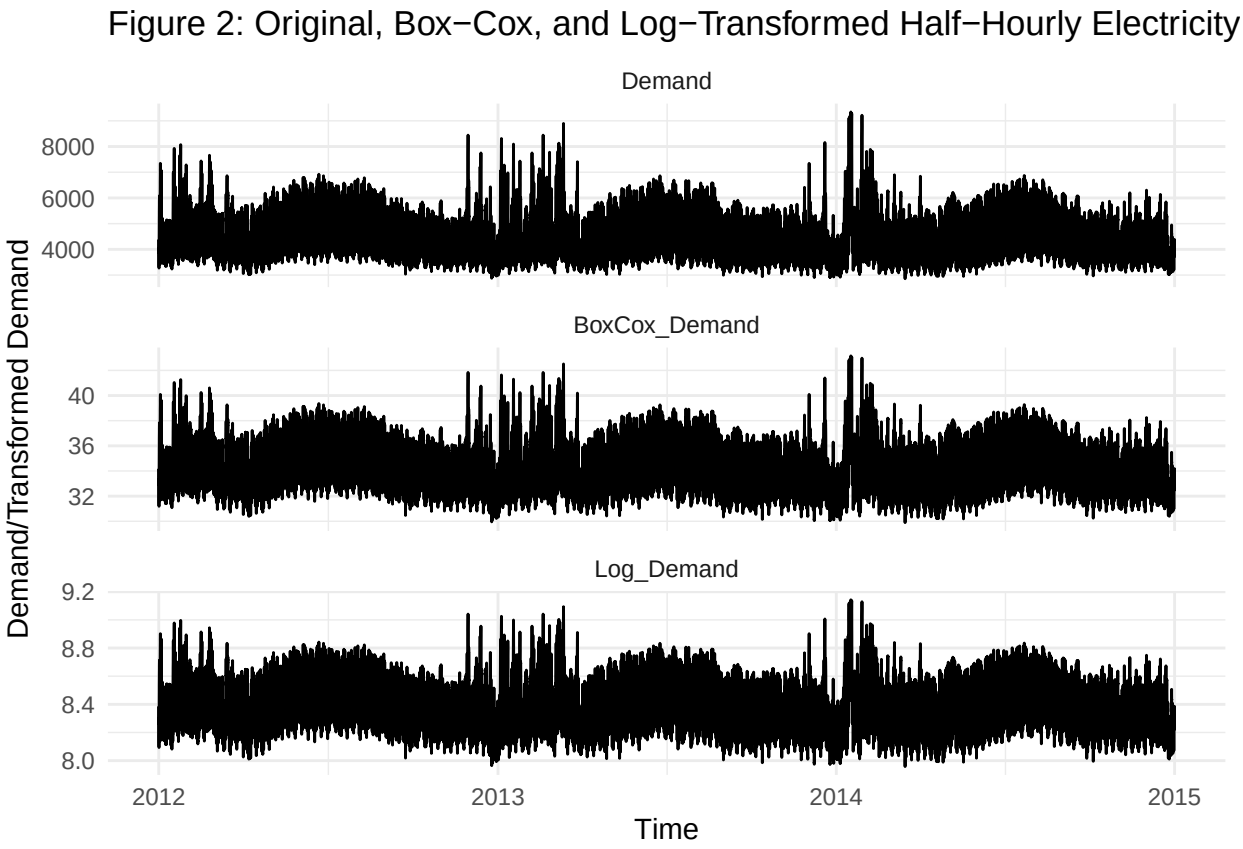
Victorian Electricity Demand from vic_elec: As shown in the second plot, the transformations (Box-Cox and Log) applied to the Half-Hourly Electricity Demand are not appropriate because they:

1. Did not result in a clearly linear relationship.
2. Does not support a graphical interpretation.

```
vic_elec |>
  autoplot(Demand) +  # Explicitly specify 'Demand' column
  labs(x = "Time",
       y = "Electricity Demand",
       title = "Figure 1: Half-Hourly Electricity Demand")
```




```
# Step 1: Ensure the Demand is listed first in the transformation order
vic_elec_long <- vic_elec |>
  mutate(BoxCox_Demand = box_cox(Demand, lambda),  # Box-Cox transformation
         Log_Demand = log(Demand)) |>  # Log transformation
  pivot_longer(cols = c("Demand", "BoxCox_Demand", "Log_Demand"),  # Place "Demand" first
               names_to = "Transformation",
               values_to = "Value") |>
  mutate(Transformation = factor(Transformation, levels = c("Demand", "BoxCox_Demand", "Log_Demand")))

# Step 2: Plot with Demand at the top using facet_wrap
ggplot(vic_elec_long, aes(x = Time, y = Value)) +
  geom_line() +
  facet_wrap(~ Transformation, scales = "free_y", ncol = 1) +  # Use facet wrap to show original, Box-C
  labs(title = "Figure 2: Original, Box-Cox, and Log-Transformed Half-Hourly Electricity Demand",
       x = "Time",
       y = "Demand/Transformed Demand") +
  theme_minimal()
```

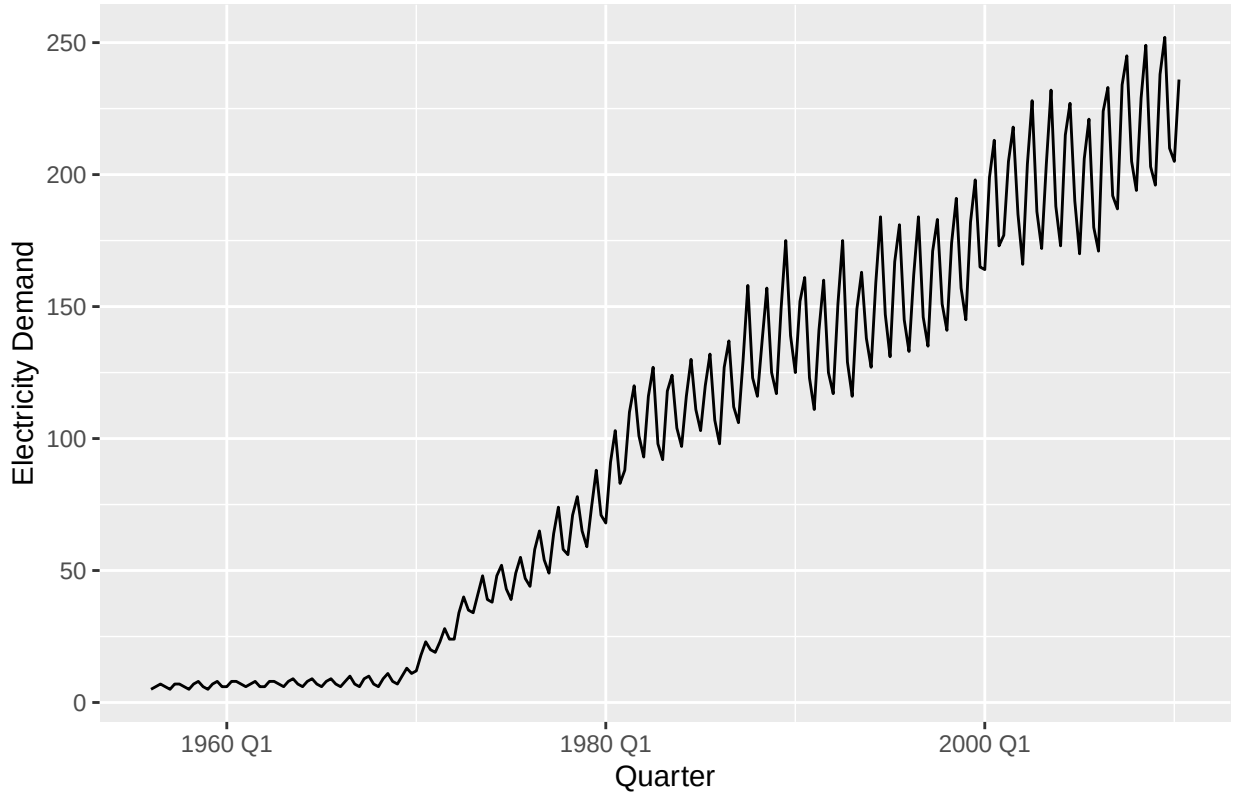


Gas production from aus_production:

As shown in Figure 2, the Box-Cox transformation has a greater effect in terms of stabilizing the variance and linearizing the data over time. This makes it more suitable for modeling purposes, particularly when the variability in the data increases with the magnitude of the observations.

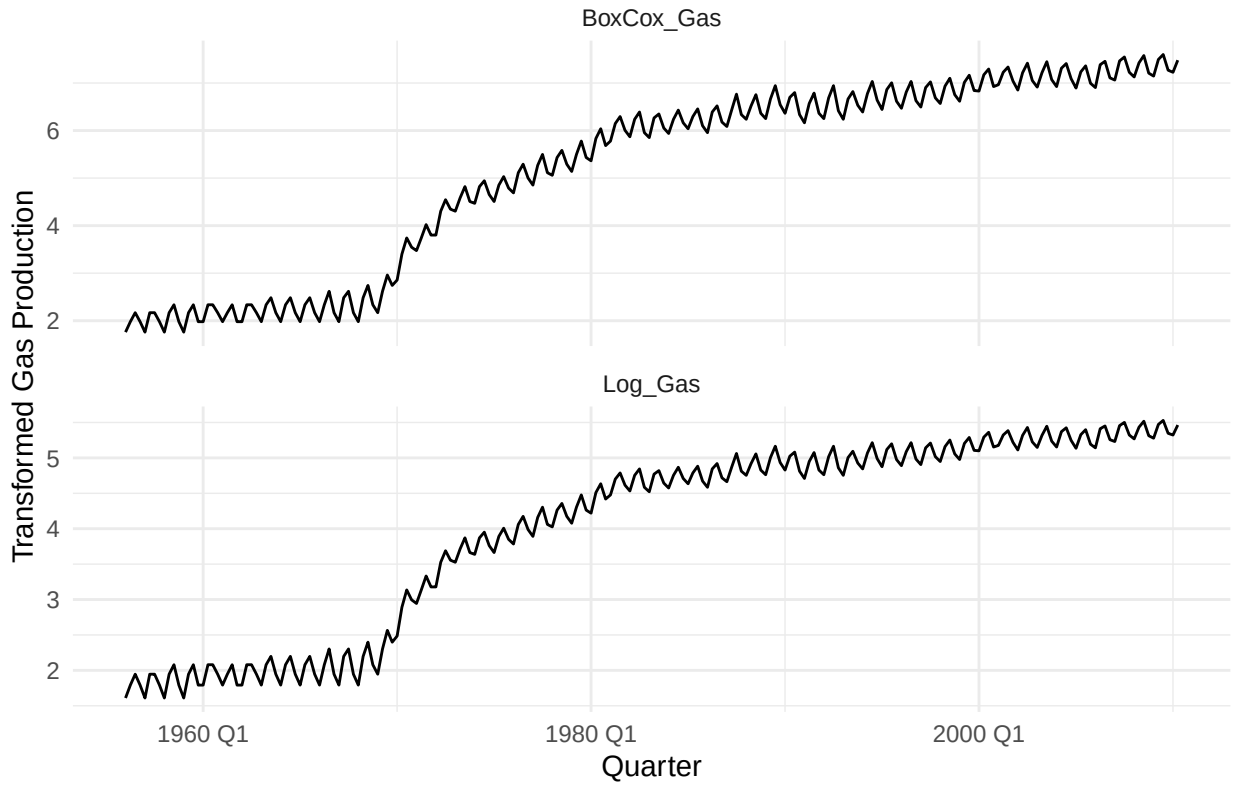
```
aus_production |>
  autoplot(Gas) +
   labs(x = "Quarter",
       y = "Electricity Demand",
       title = "Figure 1: Quarterly Gas Production")
```

Figure 1: Quarterly Gas Production




```
# Step 1: Calculate the optimal lambda using Guerrero's method for Gas
lambda_gas <- aus_production |>
  features(Gas, features = guerrero) |>
  pull(lambda_guerrero)

# Step 2: Apply the Box-Cox and Log transformations for the Gas variable
aus_production_transformed_gas <- aus_production |>
  mutate(BoxCox_Gas = box_cox(Gas, lambda_gas),  # Apply Box-Cox transformation
         Log_Gas = log(Gas)) |>
  pivot_longer(cols = c("BoxCox_Gas", "Log_Gas"),  # Pivot the Box-Cox and Log transformations
               names_to = "Transformation",
               values_to = "Value")

# Step 3: Plot the Box-Cox and Log-transformed data for Gas
ggplot(aus_production_transformed_gas, aes(x = Quarter, y = Value)) +
  geom_line() +
  facet_wrap(~ Transformation, scales = "free_y", ncol = 1) +  # Facet wrap to show Box-Cox and Log tra
  labs(title = "Figure 2: Box-Cox and Log-Transformed Quarterly Gas Production",
       x = "Quarter",
       y = "Transformed Gas Production") +
  theme_minimal()
```

Figure 2: Box−Cox and Log−Transformed Quarterly Gas Production



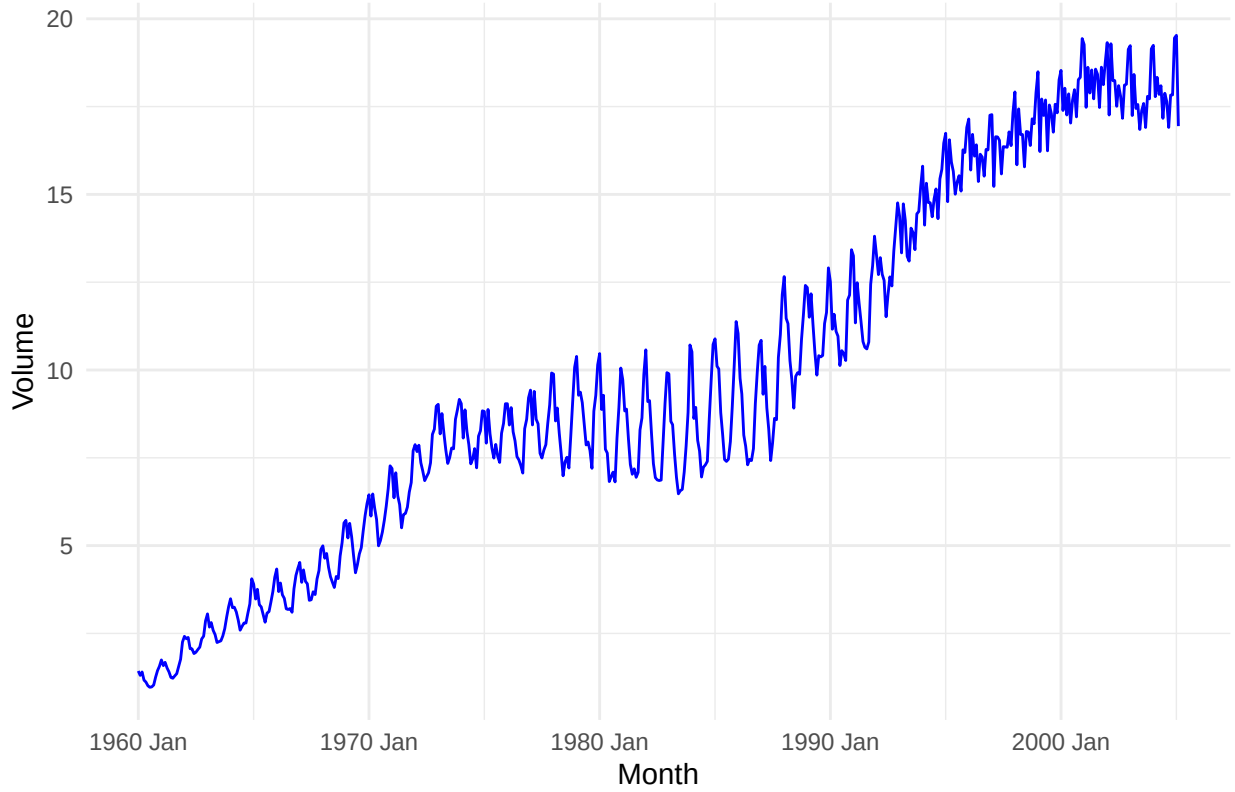
Exercise 3.7.3

Why is a Box-Cox transformation unhelpful for the canadian_gas data?

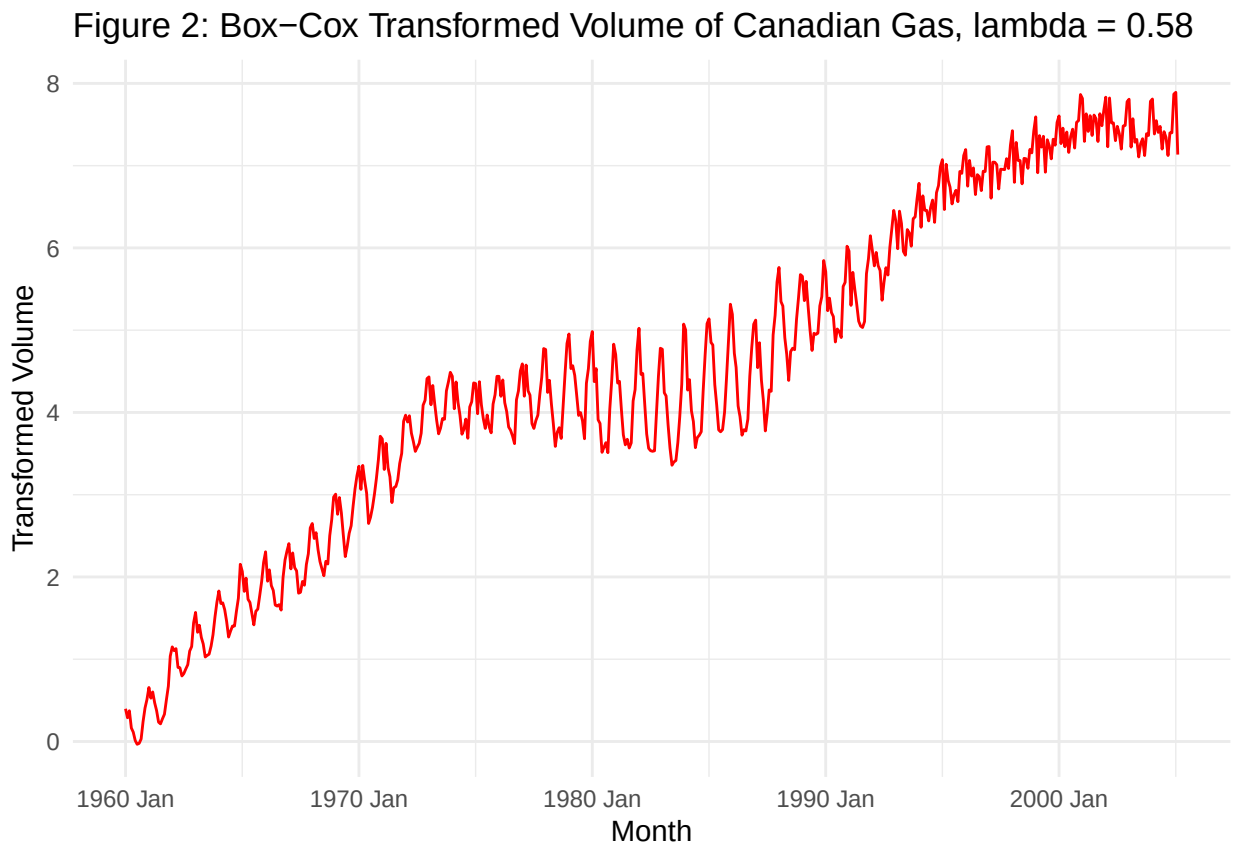
Already Stabilized Variance: A Box-Cox transformation is commonly used to stabilize variance and make data more normally distributed, especially for datasets exhibiting non-constant variance. However, the original time series data (Volume) seems to already show a relatively stable variance, especially after a certain period. The fluctuation in volume is proportional to the overall trend, making the *__Box-Cox transformation not unhelpful__

```
# Original data plot
original_plot <- ggplot(canadian_gas, aes(x = Month, y = Volume)) +
  geom_line(color = "blue") +
  labs(title = "Figure 1: Original Volume of Canadian Gas", x = "Month", y = "Volume") +
  theme_minimal()

original_plot
```

Figure 1: Original Volume of Canadian Gas




```
library(ggplot2)
library(forecast) # For Box-Cox transformation
library(tsibble)
library(feasts)

# Find the optimal lambda for the Box-Cox transformation
lambda <- canadian_gas |>
  features(Volume, features = guerrero) |>
  pull(lambda_guerrero)

# Apply the Box-Cox transformation
canadian_gas <- canadian_gas |>
  mutate(BoxCox_Volume = box_cox(Volume, lambda))

# Box-Cox transformed data plot
boxcox_plot <- ggplot(canadian_gas, aes(x = Month, y = BoxCox_Volume)) +
  geom_line(color = "red") +
  labs(title = "Figure 2: Box-Cox Transformed Volume of Canadian Gas, lambda = 0.58",
       x = "Month",
       y = "Transformed Volume") +
  theme_minimal()

print(boxcox_plot)
```



Exercise 3.7.4 What Box-Cox transformation would you select for your retail data (from Exercise 7 in Section 2.10)?

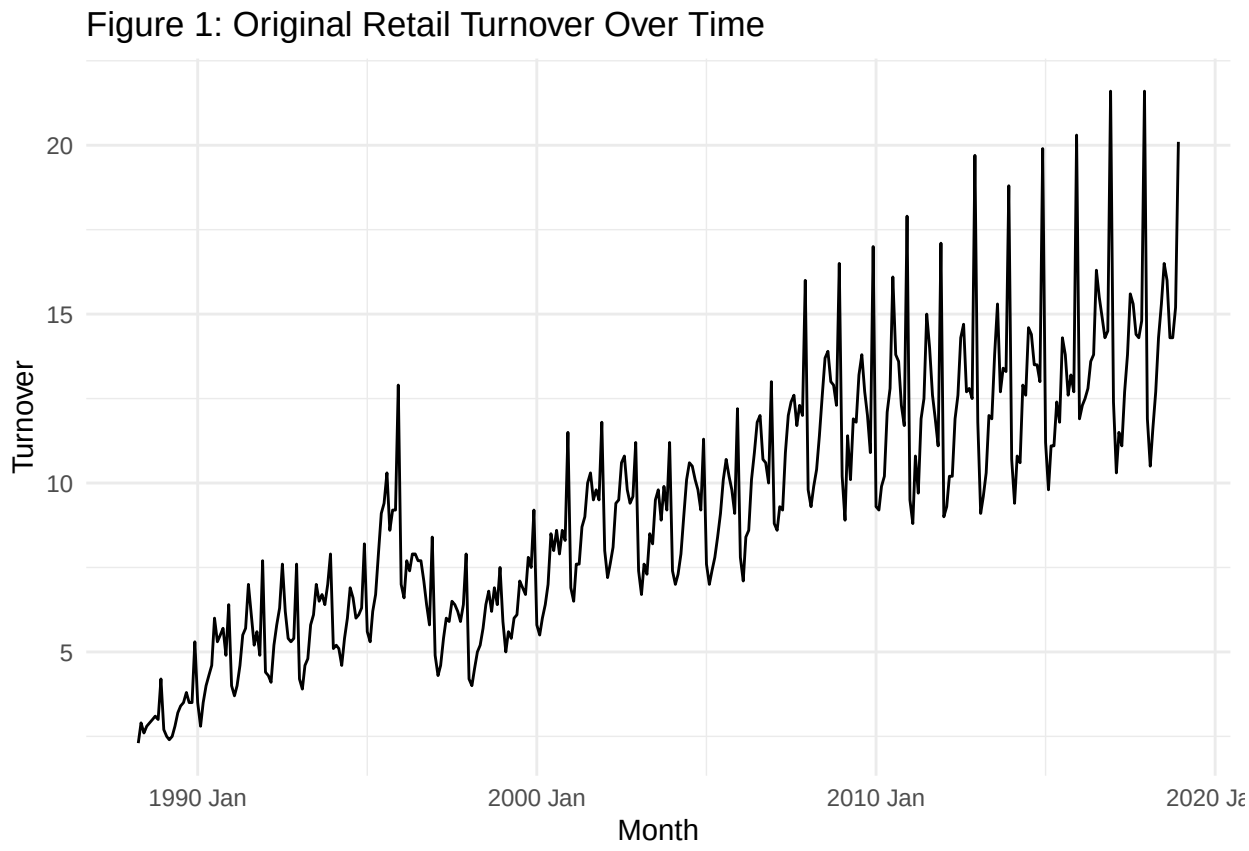
To determine the lambda for the Box-Cox transformation for the retail data from, below are the steps I would follow:

- **Fit a Model to Determine the Best Lambda:** Use the Guerrero method to determine the optimal lambda parameter for the Box-Cox transformation. This lambda is crucial to choose the most appropriate transformation that stabilizes the variance.
- **Apply the Transformation:** Once you have the optimal lambda, apply the Box-Cox transformation to the Turnover column.

Figure 2 two shows the effect of the chosen Box-Cox transformation

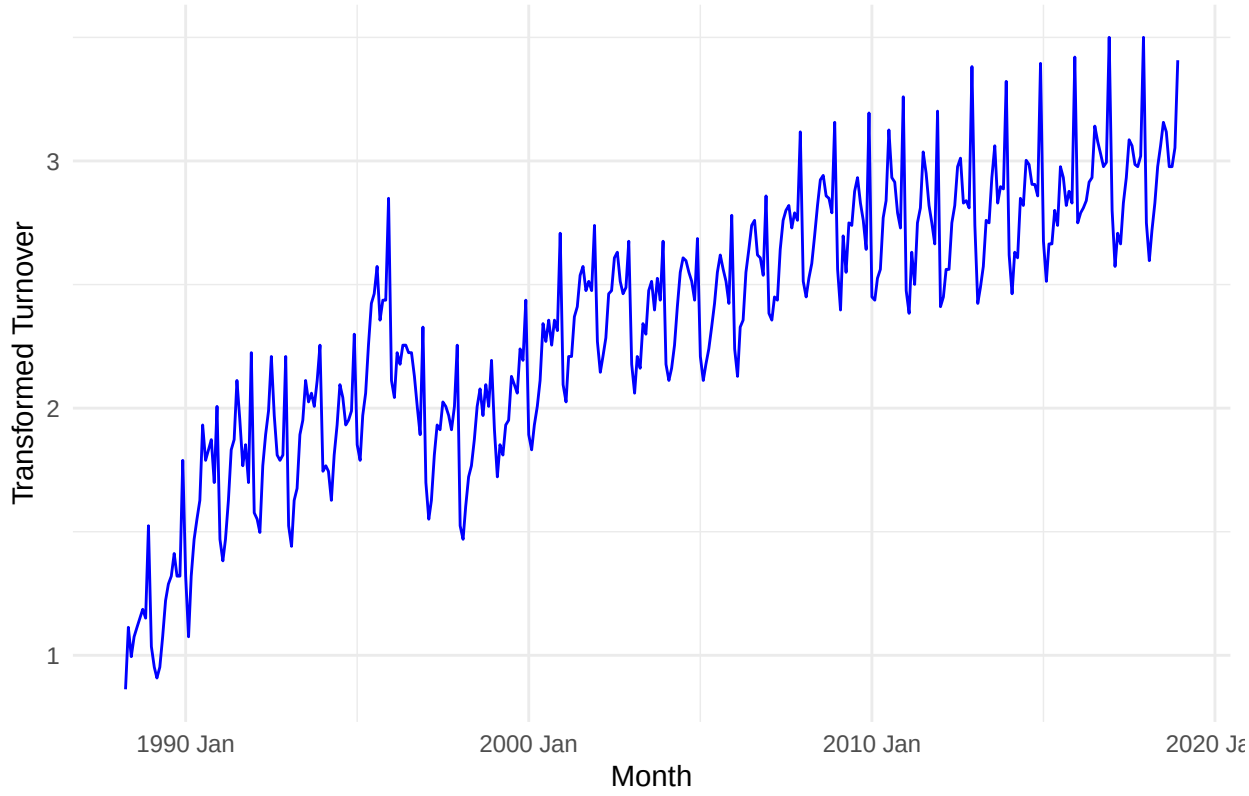
```
# Load necessary libraries
library(fabletools)
library(tsibble)
library(dplyr)

set.seed(12345678)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

# Plot the original turnover data
autoplot(myseries, Turnover) +
```

```
  # Customize the plot labels:
  labs(title = "Figure 1: Original Retail Turnover Over Time",
       x = "Month",
       y = "Turnover") +

  # Add a minimal theme for a cleaner look
  theme_minimal()
```




```
lambda <- myseries |>
  features(Turnover, features = guerrero) |>
  pull(lambda_guerrero)

# Apply the Box-Cox transformation with the optimal lambda
myseries_transformed <- myseries |>
  mutate(BoxCox_Turnover = box_cox(Turnover, lambda))

# Plot the transformed data
library(ggplot2)
ggplot(myseries_transformed, aes(x = Month, y = BoxCox_Turnover)) +
  geom_line(color = "blue") +
  labs(title = paste("Figure 2: Box-Cox Transformed Turnover, lambda = 0.08"),
       x = "Month",
       y = "Transformed Turnover") +
  theme_minimal()
```

Figure 2: Box−Cox Transformed Turnover, lambda = 0.08



Exercise 3.7.5

To get the optimal lambda values,

- Calculate the optimal lambda for Box-Cox transformation using the `guerrero` method.
- After determining the lambda, you can apply the transformation to stabilize the variance of the data.

The results are compiled into a data frame

```
# Find the optimal lambda for Tobacco from aus_production
lambda_tobacco <- aus_production |>
  features(Tobacco, features = guerrero) |>
  pull(lambda_guerrero)

# Apply the Box-Cox transformation with the optimal lambda
tobacco_transformed <- aus_production |>
  mutate(BoxCox_Value = box_cox(Tobacco, lambda_tobacco))

# ------------------------------------------------------------------------
# Filter the Economy class passengers data between Melbourne and Sydney
ansett_data <- ansett %>%
  filter(Class == "Economy", Airports == "MEL-SYD")

# Find the optimal lambda for the Economy class passengers between Melbourne and Sydney from ansett
```

```
lambda_passengers <- ansett_data |>
  features(Passengers, features = guerrero) |>
  pull(lambda_guerrero)

# Apply the Box-Cox transformation with the optimal lambda
ansett_transformed <- ansett_data |>
  mutate(BoxCox_Passengers = box_cox(Passengers, lambda_passengers))

# ------------------------------------------------------------------------
# Filter the Pedestrian counts data at Southern Cross Station
pedestrian_data <- pedestrian %>%
  filter(Sensor == "Southern Cross Station")

# Find the optimal lambda for the Pedestrian counts at Southern Cross Station from pedestrian
lambda_pedestrian <- pedestrian_data |>
  features(Count, features = guerrero) |>
  pull(lambda_guerrero)

# Apply the Box-Cox transformation with the optimal lambda
pedestrian_transformed <- pedestrian_data |>
  mutate(BoxCox_Count = box_cox(Count, lambda_pedestrian))

# ------------------------------------------------------------------------
# Create a data frame for lambda values along with dataset names
lambda_values <- data.frame(
  Dataset = c('aus_production', 'ansett', 'pedestrian'),
  Series = c('Tobacco Production', 'Economy Passengers MEL-SYD', 'Pedestrian Counts Southern Cross'),
  Lambda = c(lambda_tobacco, lambda_passengers, lambda_pedestrian)
)

# Print the data frame
print(lambda_values)
```

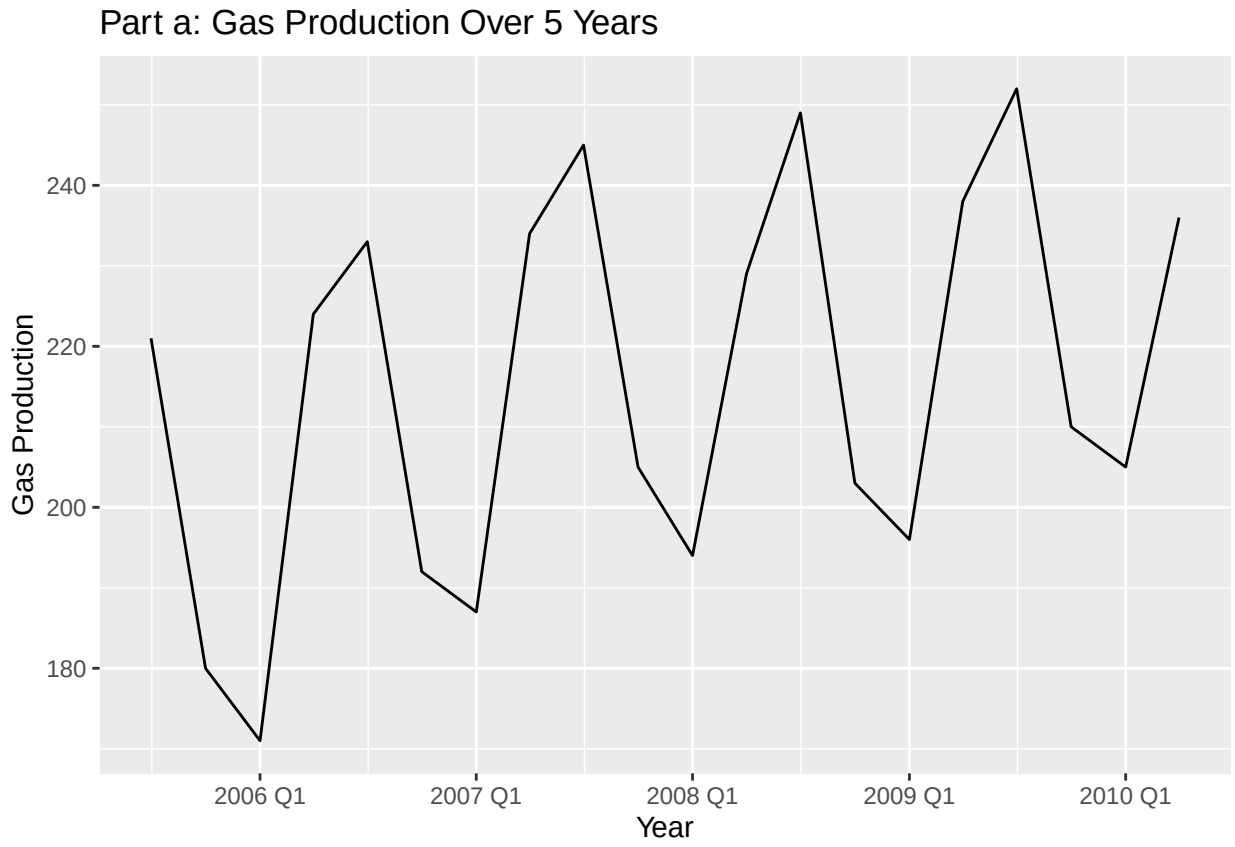
```
##          Dataset                          Series     Lambda
## 1 aus_production              Tobacco Production  0.9264636
## 2         ansett       Economy Passengers MEL-SYD  1.9999268
## 3     pedestrian Pedestrian Counts Southern Cross -0.2501616
```

Exercise 3.7.7

a. Plot the time series. Can you identify seasonal fluctuations and/or a trend-cycle?

From the plot, it is evident that there are clear seasonal fluctuations in the gas production over the five years, with a recurring pattern each year. The production tends to rise and fall in a consistent manner, likely indicating seasonality in the data.

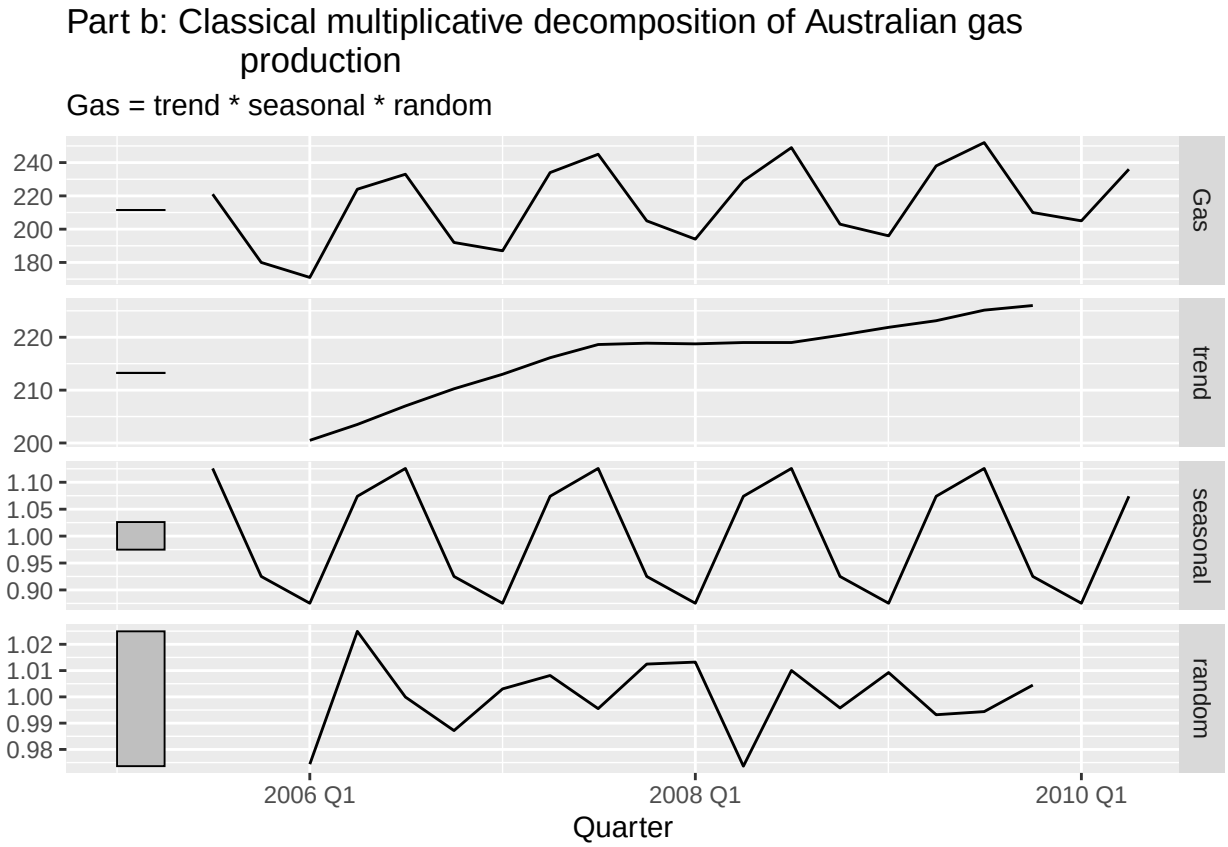
```
gas <- tail(aus_production, 5*4) |> select(Gas)

# Step 2: Plotting the time series
gas |> autoplot(Gas) +
  labs(title = paste("Part a: Gas Production Over 5 Years"),
       x = "Year",
       y = "Gas Production")
```



b. Use classical_decomposition with type=multiplicative to calculate the trend-cycle and seasonal indices.

```
decomp_gas <- gas |>
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) |>
  components() |>
  autoplot() +
  labs(title = "Part b: Classical multiplicative decomposition of Australian gas
                 production")

decomp_gas
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



c. Do the results support the graphical interpretation from part a?

Yes, the results from the classical multiplicative decomposition in part b support the graphical interpretation from part a.

- In part a, the original plot of gas production shows clear seasonal fluctuations. The series exhibits a repeating pattern every year, which suggests a seasonal component. Additionally, the overall level of the series seems to be increasing slightly over time, indicating a trend.
- In part b, the decomposition results confirm this interpretation:
  - The **seasonal component** is evident and captures the repeating seasonal fluctuations observed in the original series.
  - The **trend component** shows a slight upward movement over time, matching the small overall increase visible in the original plot.
  - The **random component** reflects the residual noise or irregularities after accounting for the trend and seasonal patterns.

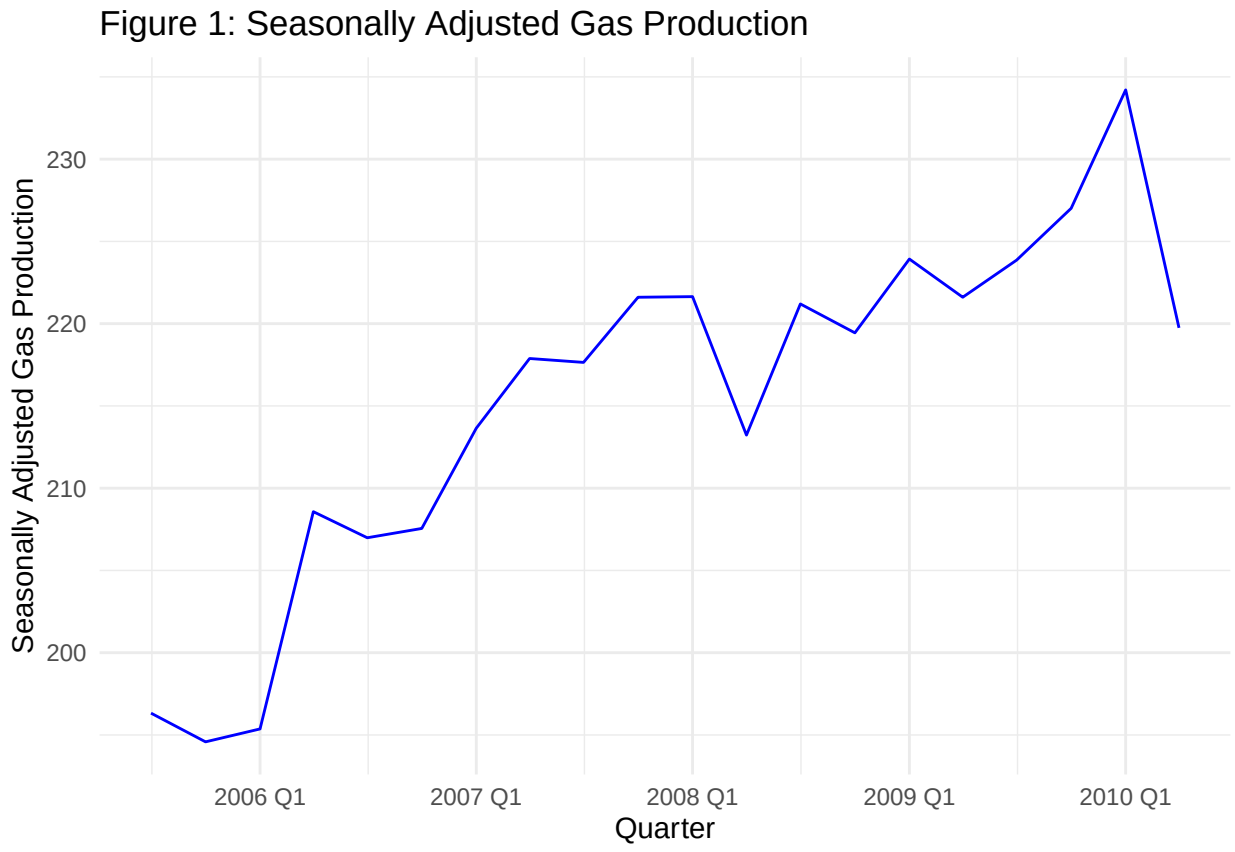
Thus, the decomposition aligns well with the initial visual assessment of the series in part a.

d. Compute and plot the seasonally adjusted data.

The seasonal adjustment happens when you decompose the data using classical_decomposition(). The seasonal component is isolated, and the seasonally adjusted data is the original series divided by the seasonal

component is accessed through the season_adjust column. This column is automatically created by the components() function after performing the decomposition.

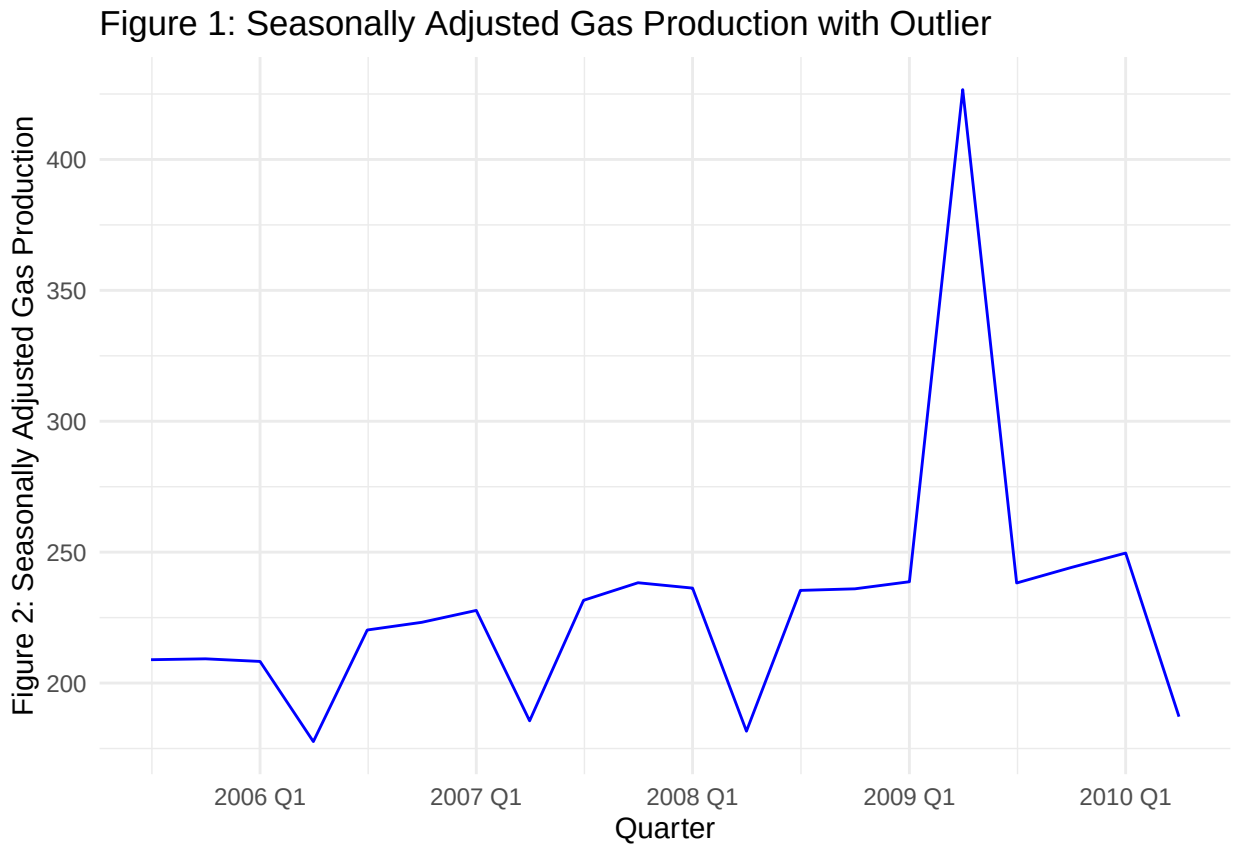
```
# Decompose the gas data using classical decomposition (multiplicative)
decomp_gas <- gas |>
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) |>
  components()

# Plot the seasonally adjusted data
ggplot(decomp_gas, aes(x = Quarter, y = season_adjust)) +
  geom_line(color = "blue") +
  labs(title = "Figure 1: Seasonally Adjusted Gas Production",
       x = "Quarter",
       y = "Seasonally Adjusted Gas Production") +
  theme_minimal()
```



e. Change one observation to be an outlier (e.g., add 300 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

Effect of the Outlier - As shown in Figure 2, the outlier distorts the trend and seasonal components, causing a significant spike in the adjusted data.

```r
# Copy of original data
gas_outlier <- gas

# Introduce an outlier by adding 300 to the observation in 2009 Q2
gas_outlier$Gas[gas_outlier$Quarter == yearquarter("2009 Q2")] <- gas_outlier$Gas[gas_outlier$Quarter ==

# Classical decomposition with the outlier
decomp_gas_outlier <- gas_outlier |>
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) |>
  components()

# Plot the seasonally adjusted data
ggplot(decomp_gas_outlier, aes(x = Quarter, y = season_adjust)) +
  geom_line(color = "blue") +
  labs(title = "Figure 1: Seasonally Adjusted Gas Production with Outlier",
       x = "Quarter",
       y = "Figure 2: Seasonally Adjusted Gas Production") +
  theme_minimal()
```

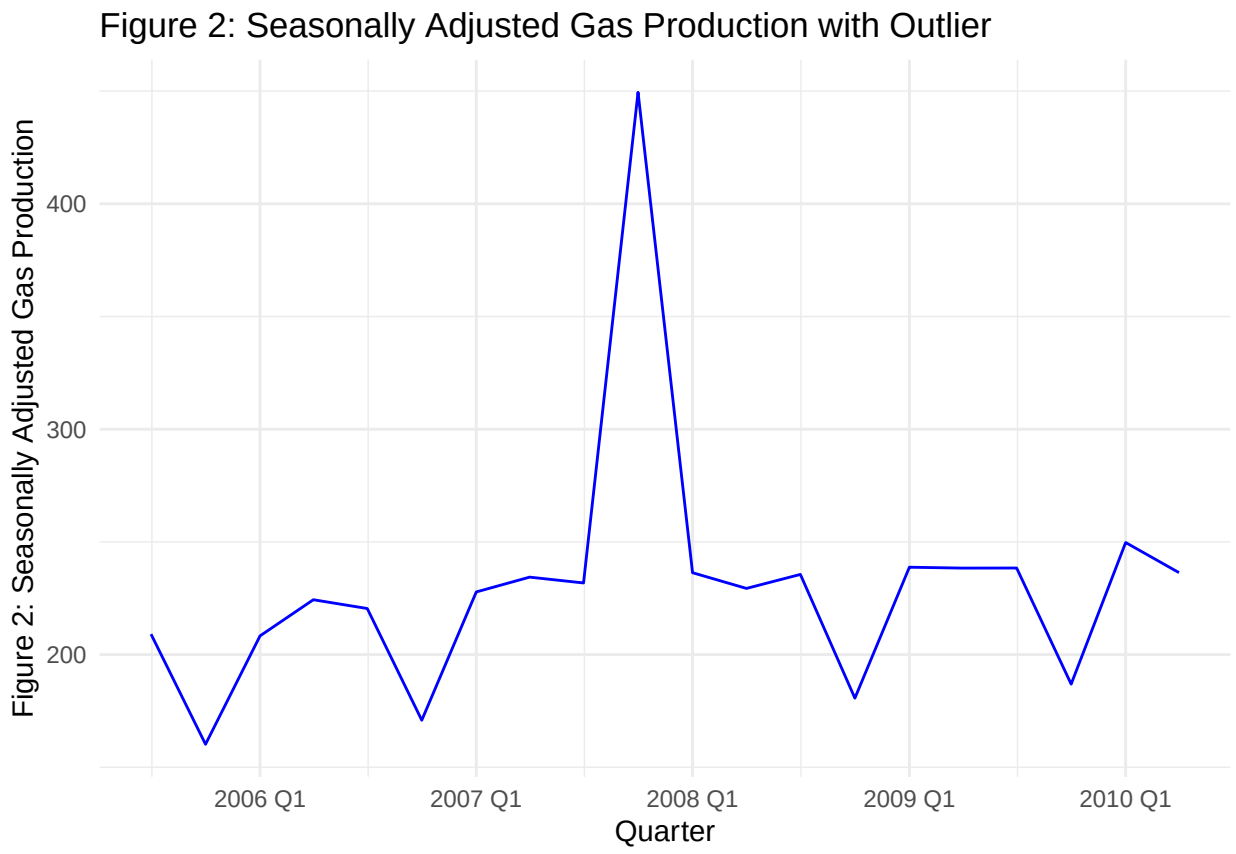


Does it make any difference if the outlier is near the end rather than in the middle of the time series?

Based on the Figure 1 above and Figure 2 below, it appears that the location of the outlier does make a difference in how it affects the seasonally adjusted gas production. When the outlier is near the middle of the time series (2008 Q2 in Figure 2), it causes a noticeable spike that distorts the surrounding data, pushing the seasonal adjustment upwards during the middle period. Similarly, when the outlier is near the end (2009 Q2 in Figure 1), a sharp increase can be seen towards the end of the series, followed by a steep drop.

In both cases, the outlier creates an unnatural spike, but its placement can affect how subsequent values appear, as the seasonal and trend adjustments rely on the relative structure of the data points.

```
# Copy of original data
gas_middle_outlier <- gas

# Introduce an outlier by adding 300 to the observation in 2007 Q4
gas_middle_outlier$Gas[gas_middle_outlier$Quarter == yearquarter("2007 Q4")] <- gas_middle_outlier$Gas[

# Classical decomposition with the outlier
decomp_gas_outlier <- gas_middle_outlier |>
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) |>
  components()

# Plot the seasonally adjusted data
ggplot(decomp_gas_outlier, aes(x = Quarter, y = season_adjust)) +
  geom_line(color = "blue") +
  labs(title = "Figure 2: Seasonally Adjusted Gas Production with Outlier",
       x = "Quarter",
       y = "Figure 2: Seasonally Adjusted Gas Production") +
  theme_minimal()
```



Exercise 3.7.8: Recall your retail time series data (from Exercise 7 in Section 2.10). Decompose the series using X-11. Does it reveal any outliers, or unusual features that you had not noticed previously?

From the X-11 decomposition in Figure 1 and the seasonal plots in Figures 2a, 2b and 2c, the X-11 decomposition reveals more detailed insights into the structure of the time series that might not be easily spotted in the seasonal plots:

1. **Trend**: The decomposition clearly shows a smooth trend, highlighting a growth pattern over time that could have been harder to distinguish just by looking at the seasonal plots. The trend line also captures a period of stagnation or slight decline in the early 2000s that isn't as apparent in the other plots.

2. **Seasonal Component**: The X-11 decomposition provides a clear seasonal component, confirming the presence of consistent seasonal fluctuations. This confirms the patterns observed in the seasonal subseries plot, but the decomposition makes it easier to isolate and visualize.

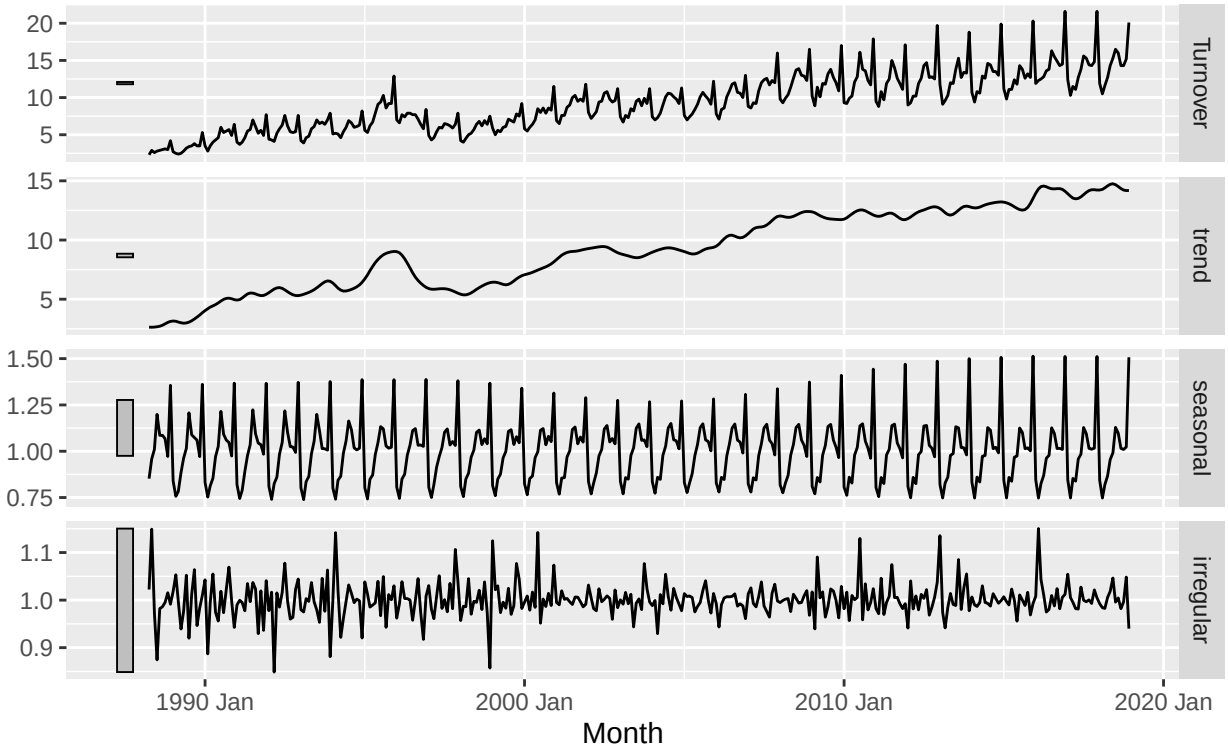
3. **Irregular Component**: The irregular component (residuals) indicates periods where there were unusual fluctuations or deviations from the expected trend. In contrast, the seasonal plots do not show such a clear irregular or random component.

4. **Outliers**: While the seasonal plots do not reveal any clear outliers, the irregular component from the X-11 decomposition reveals some unusual features or spikes, which may correspond to outliers or unpredicted events within the dataset. These irregular fluctuations are especially prominent toward the end of the time series.

```r
set.seed(12345678)

myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

x11_dcmp <- myseries |>
  model(x11 = X_13ARIMA_SEATS(Turnover ~ x11())) |>
  components()

autoplot(x11_dcmp) +
  labs(title =
    "Figure 1: Decomposition of total US retail employment using X-11.")
```

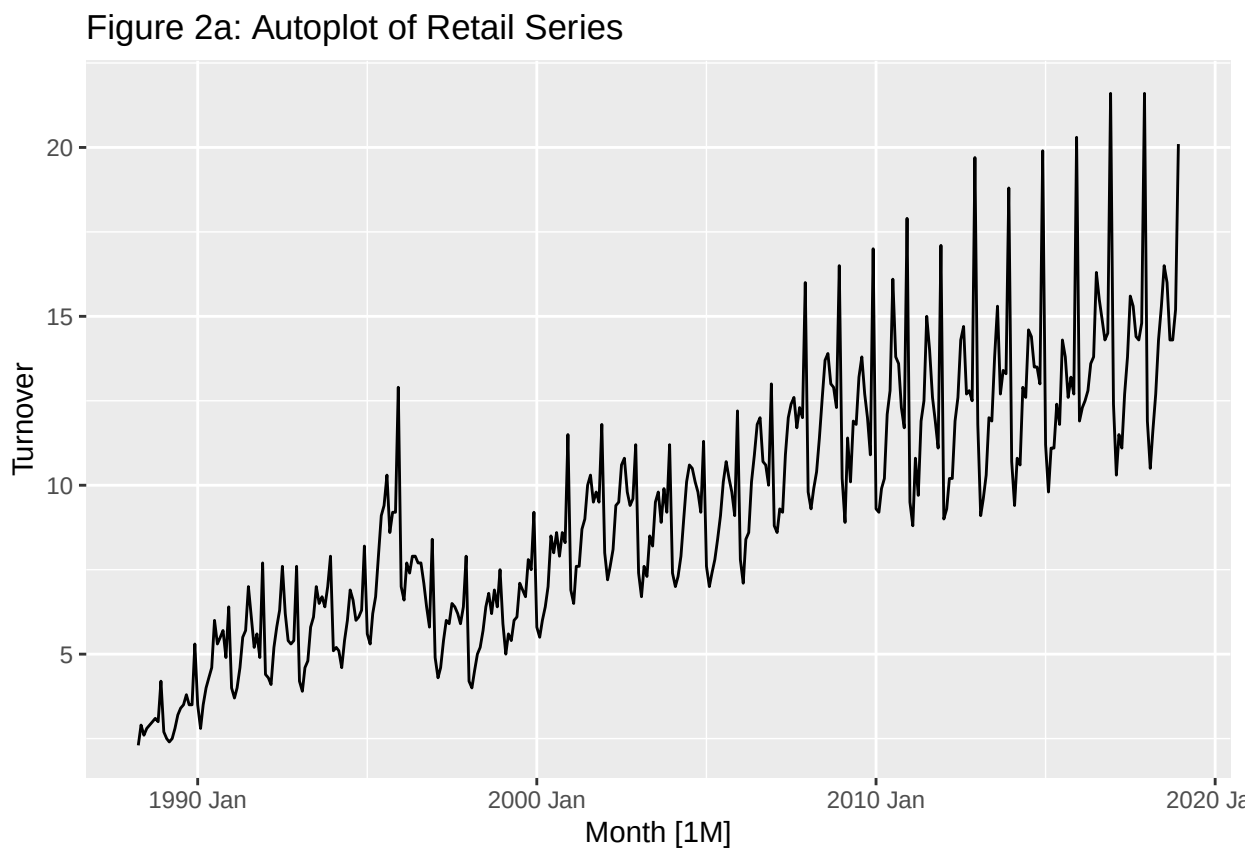
Figure 1: Decomposition of total US retail employment using X−11.

Turnover = trend * seasonal * irregular



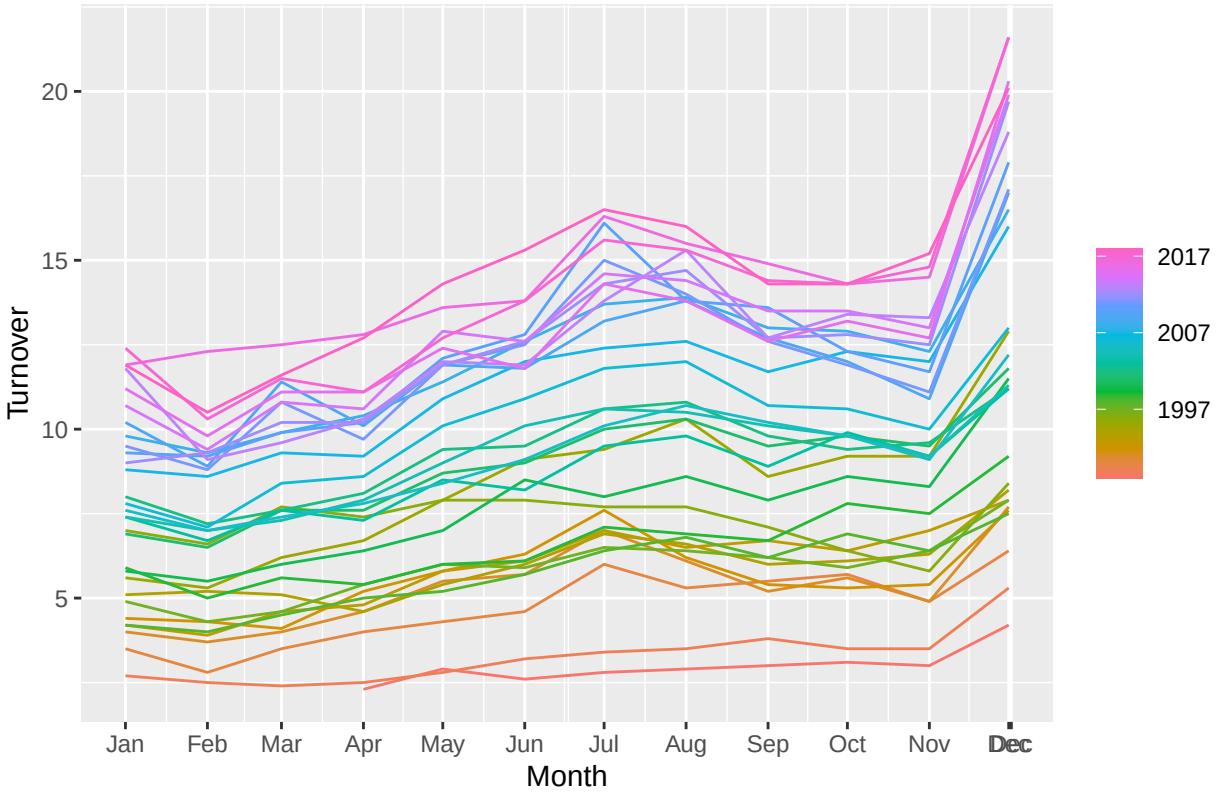

```r
set.seed(12345678)

# Filter and select a series
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`, 1))

# Autoplot of the time series
p1 <- autoplot(myseries, Turnover) +
  labs(title = "Figure 2a: Autoplot of Retail Series", y = "Turnover")

# gg_season for seasonality
p2 <- gg_season(myseries, Turnover) +
```

```
  labs(title = "Figure 2b: Seasonal Plot of Retail Series", y = "Turnover")

# gg_subseries for subseries plot
p3 <- gg_subseries(myseries, Turnover) +
  labs(title = "Figure 2c: Subseries Plot of Retail Series", y = "Turnover")

p1
```

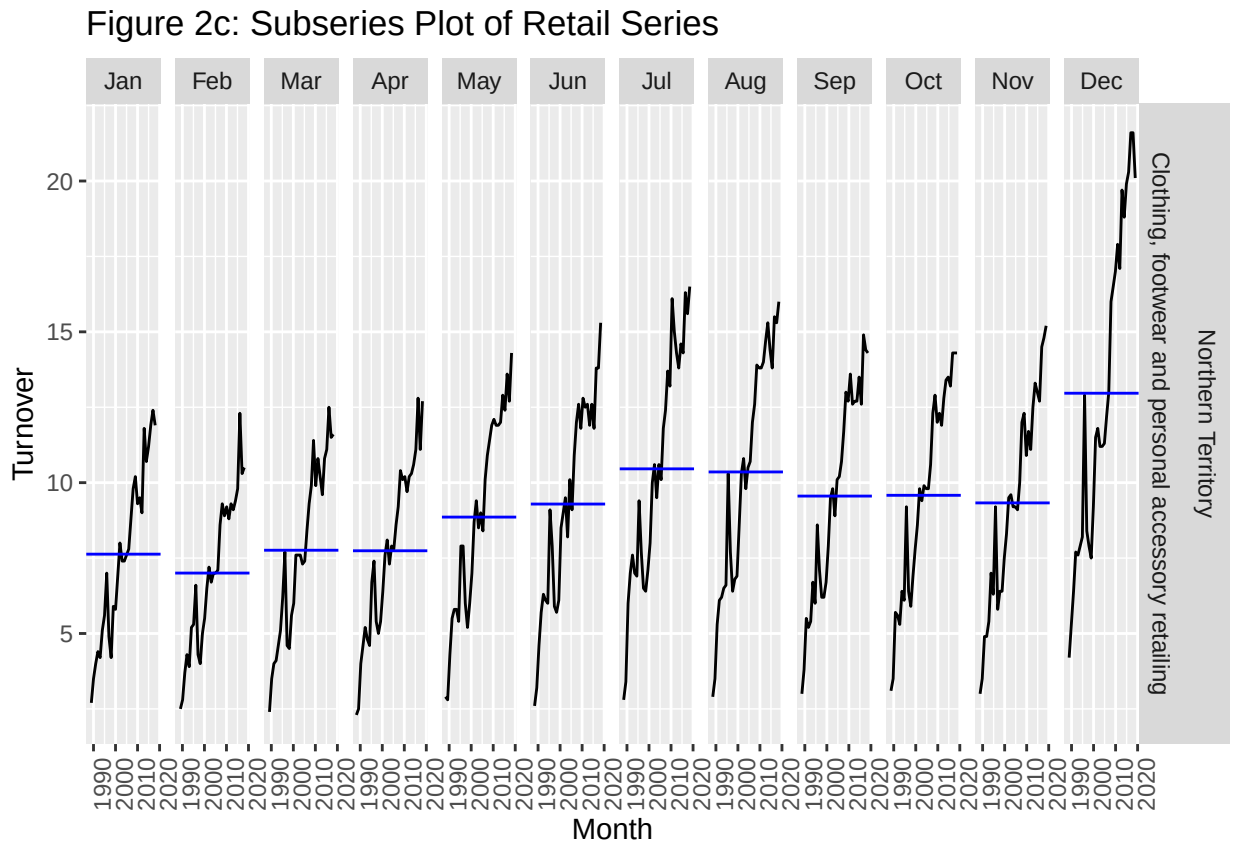



```
p2
```

Figure 2b: Seasonal Plot of Retail Series




```
p3
```



Exercise 3.7.9

a. Write about 3–5 sentences describing the results of the decomposition. Pay particular attention to the scales of the graphs in making your interpretation.

The STL decomposition of the civilian labor force in Australia from February 1978 to August 1995 reveals several important patterns. The "value" plot shows a clear upward trend, with an increase in the overall labor force over time, indicating long-term growth in labor participation. The "trend" component also highlights this gradual increase but smooths out shorter fluctuations. The "season_year" plot shows recurring seasonal variations, which fluctuate consistently throughout the year, revealing cyclical patterns in labor force changes.

b. Is the recession of 1991/1992 visible in the estimated components?

The "remainder" component indicates residual fluctuations or irregular movements in the data that are not captured by the trend or seasonal components. These include short-term shocks and anomalies. The recession of 1991/1992 is somewhat visible in the remainder and trend components, as there is a noticeable dip during that period, especially in the remainder plot, suggesting that the recession had an impact on the labor force.