Data 624 Homework 2 Chapter 3.7

Enid Roman

2024-09-14

1. Consider the GDP information in global_economy. Plot the GDP per capita for each country over time. Which country has the highest GDP per capita? How has this changed over time?

```r
# Load required packages

library(fpp3)
library(tsibble)
library(ggplot2)
library(dplyr)
#install.packages("forecast")
library(forecast)
library(zoo)
#install.packages("slider")
library(slider)
#install.packages("seasonal")
#install.packages("Rtools")
```

```r
# View the first few rows of the aus_production dataset
head(global_economy)
```

```
## # A tsibble: 6 x 9 [1Y]
## # Key:        Country [1]
##   Country     Code   Year         GDP Growth   CPI Imports Exports Population
##   <fct>       <fct> <dbl>       <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
## 2 Afghanistan AFG    1961  548888896.     NA    NA    8.10    4.45    9166764
## 3 Afghanistan AFG    1962  546666678.     NA    NA    9.35    4.88    9345868
## 4 Afghanistan AFG    1963  751111191.     NA    NA   16.9     9.17    9533954
## 5 Afghanistan AFG    1964  800000044.     NA    NA   18.1     8.89    9731361
## 6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4    11.3     9938414
```

```r
# Create a time series plot of GDP Per Capita for Each Country
global_economy %>%
  autoplot(GDP/Population,show.legend = FALSE) +
  labs(
    title = "GDP Per Capita for Each Country",
    x = "Year",
    y = "GDP Per Capita",
    )
```

GDP Per Capita for Each Country
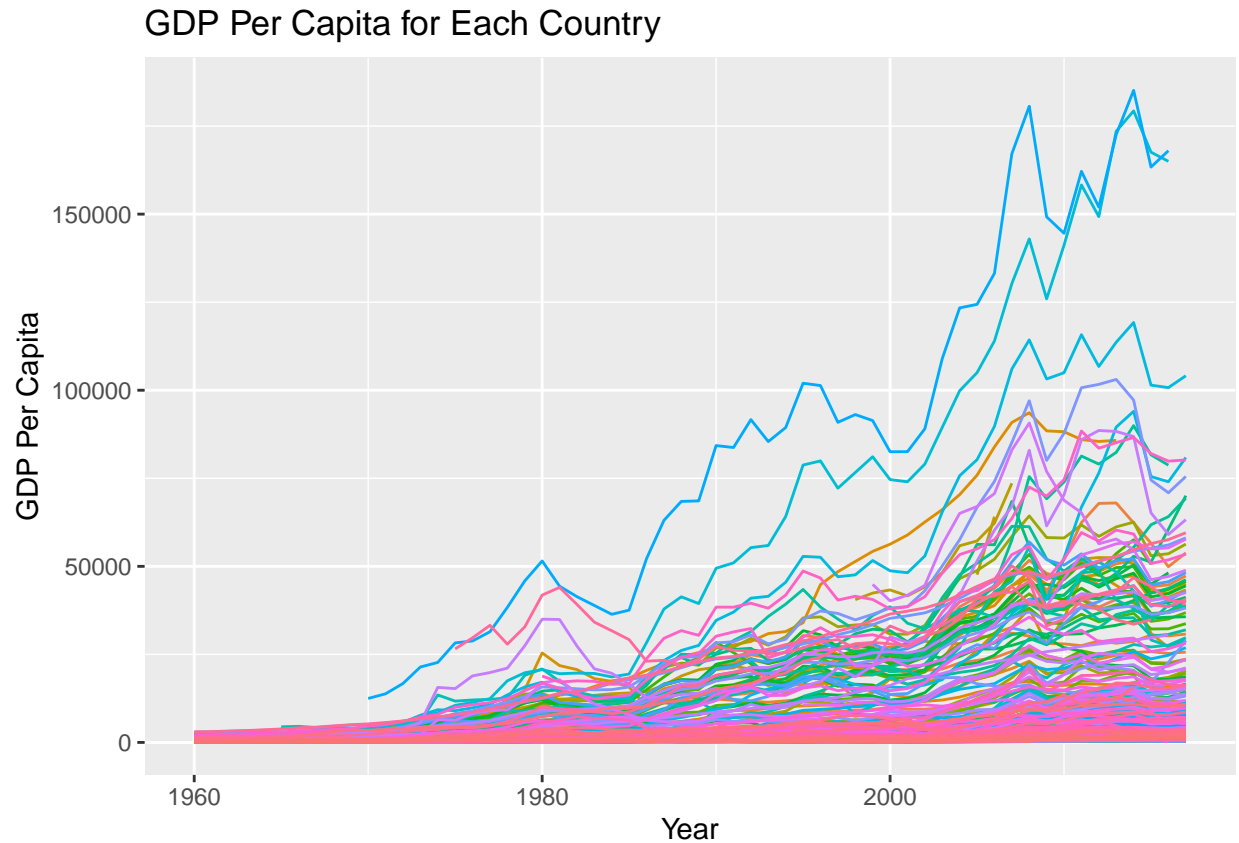

```r
# Find the country with the highest GDP per capita
global_economy %>%
  mutate(GDP_per_capita = GDP / Population) %>%  # Calculate GDP per capita
  filter(GDP_per_capita == max(GDP_per_capita, na.rm = TRUE)) %>%  # Find the highest GDP per capita
  select(Country, Year, GDP_per_capita)  # Select relevant columns (Country, Year, GDP per capita)
```

```
## # A tsibble: 1 x 3 [1Y]
## # Key:         Country [1]
##   Country  Year GDP_per_capita
##   <fct>   <dbl>          <dbl>
## 1 Monaco   2014        185153.
```

Monaco has the highest GDP per capita with 18512.5 in 2014.

```r
# Filter the data for Monaco and plot the GDP per capita over time
global_economy %>%
  filter(Country == "Monaco") %>%
  mutate(GDP_per_capita = GDP / Population) %>%
  ggplot(aes(x = Year, y = GDP_per_capita)) +
  geom_line(color = "blue") +
  labs(
    title = "GDP Per Capita Over Time for Monaco",
    x = "Year",
    y = "GDP Per Capita"
  )
```

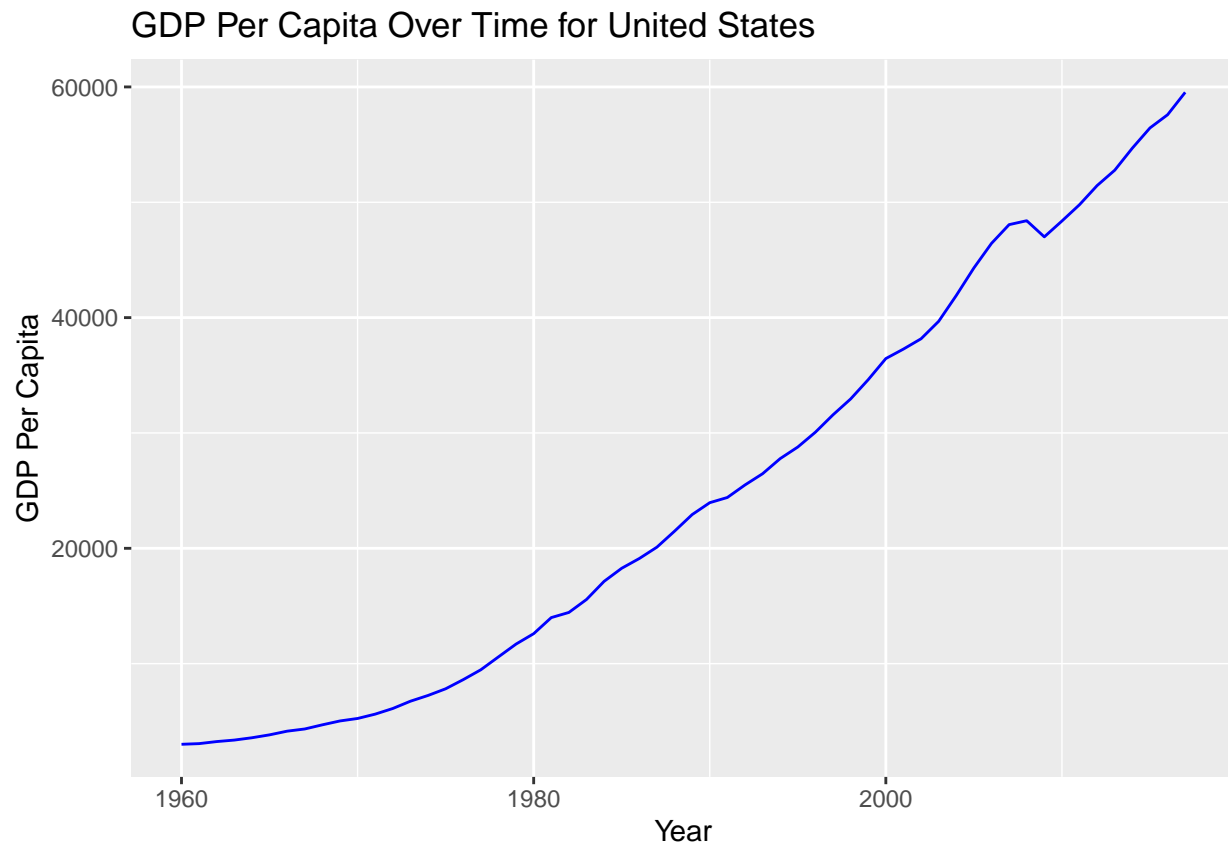# GDP Per Capita Over Time for Monaco
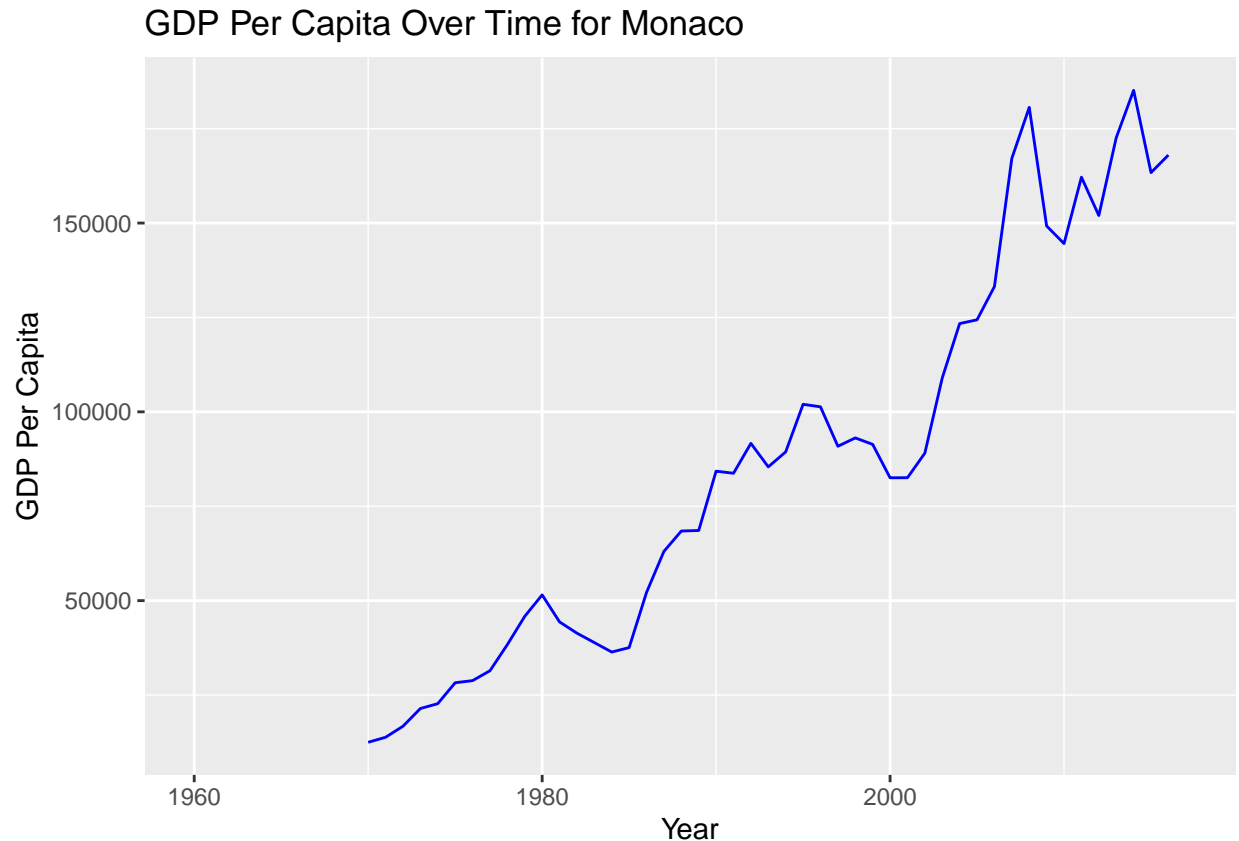
This graph gives a clear visual representation of Monaco's economic performance relative to its population over time. You can see that Monaco increase in GDP per capita, the line is trending upward. This suggests that Monaco's economy has been growing faster than its population. There is a bit fluctuation. There are significant rises and falls, which might indicate economic shocks, policy changes, or fluctuations in Monaco's population.

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.

```
# Filter the data for United States and plot the GDP per capita over time
global_economy %>%
  filter(Country == "United States") %>%
  mutate(GDP_per_capita = GDP / Population) %>%
  ggplot(aes(x = Year, y = GDP_per_capita)) +
  geom_line(color = "blue") +
  labs(
    title = "GDP Per Capita Over Time for United States",
    x = "Year",
    y = "GDP Per Capita"
  )
```

GDP Per Capita Over Time for United States

This graph gives a clear visual representation of United States economic performance relative to its population over time. You can see that United States increase in GDP per capita, the line is trending upward. This suggests that United States economy has been growing faster than its population. There is a bit fluctuation. There is a slight rise and fall in 2007 - 2009, which might indicate economic shocks, policy changes, or fluctuations in United States population. The United States experienced a severe recession from December 2007 to June 2009, also known as the Great Recession. The economy began to grow again in mid-2009 after the passage of the American Recovery and Reinvestment Act and the Troubled Asset Relief Program (TARP).

There are transform needed for GDP Per Capita Over Time for United States. An increase in GDP per capita can be a sign of economic growth.

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

```
# View the first few rows of the aus_production dataset
head(aus_livestock)
```

```
## # A tsibble: 6 x 4 [1M]
## # Key:       Animal, State [1]
##      Month Animal                   State                        Count
##      <mth> <fct>                    <fct>                        <dbl>
## 1 1976 Jul Bulls, bullocks and steers Australian Capital Territory  2300
## 2 1976 Aug Bulls, bullocks and steers Australian Capital Territory  2100
## 3 1976 Sep Bulls, bullocks and steers Australian Capital Territory  2100
## 4 1976 Oct Bulls, bullocks and steers Australian Capital Territory  1900
```

4

```
## 5 1976 Nov Bulls, bullocks and steers Australian Capital Territory   2100
## 6 1976 Dec Bulls, bullocks and steers Australian Capital Territory   1800
```

```
# Filter the data and plot
aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers", State == "Victoria") %>%
  autoplot(Count) +
  labs(title = "Australian Livestock Slaughter: Bulls, bullocks and steers in Victoria")
```


Australian Livestock Slaughter: Bulls, bullocks and steers in Victoria
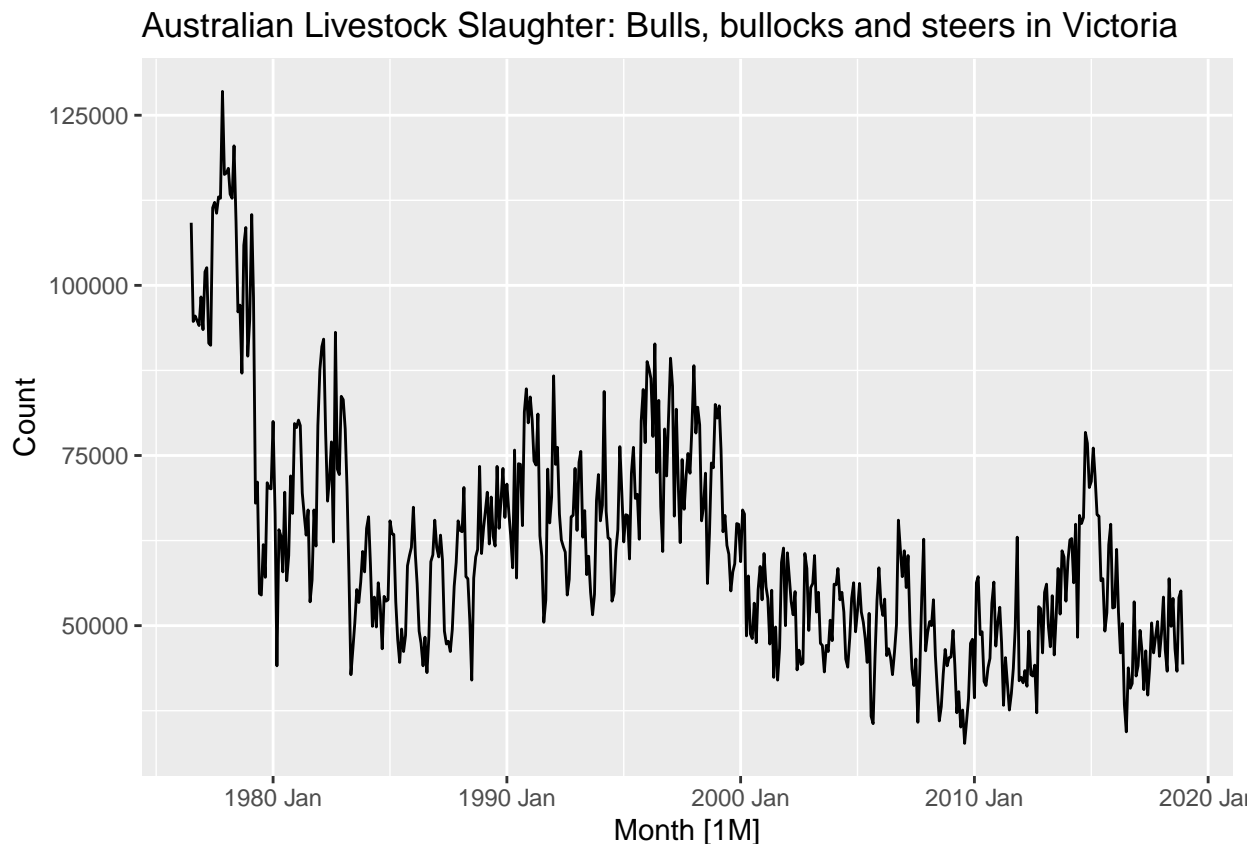
The livestock dataset appears to exhibit excessive fluctuations, alongside an overall decreasing trend. These fluctuations can obscure any underlying patterns or long-term trends, making it difficult to interpret the data or draw meaningful conclusions. The fluctuations might be caused by short-term variability in the data, seasonal effects, or random noise. Meanwhile, the downward trend suggests a consistent decline in livestock numbers over time.

To better analyze and interpret this data, it may be helpful to apply a transformation or smoothing technique. For example, using a moving average could reduce the noise and highlight the underlying trend more clearly. Additionally, seasonal decomposition might help separate any seasonal patterns from the overall trend, making it easier to focus on the general decrease in livestock counts over time.

```
# Applying a moving average
aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers", State == "Victoria") %>%
  mutate(Count_MA = zoo::rollmean(Count, k = 12, fill = NA)) %>%
  autoplot(Count_MA) +
  ggtitle("12-Month Moving Average of Livestock Count in Victoria")
```

12–Month Moving Average of Livestock Count in Victoria
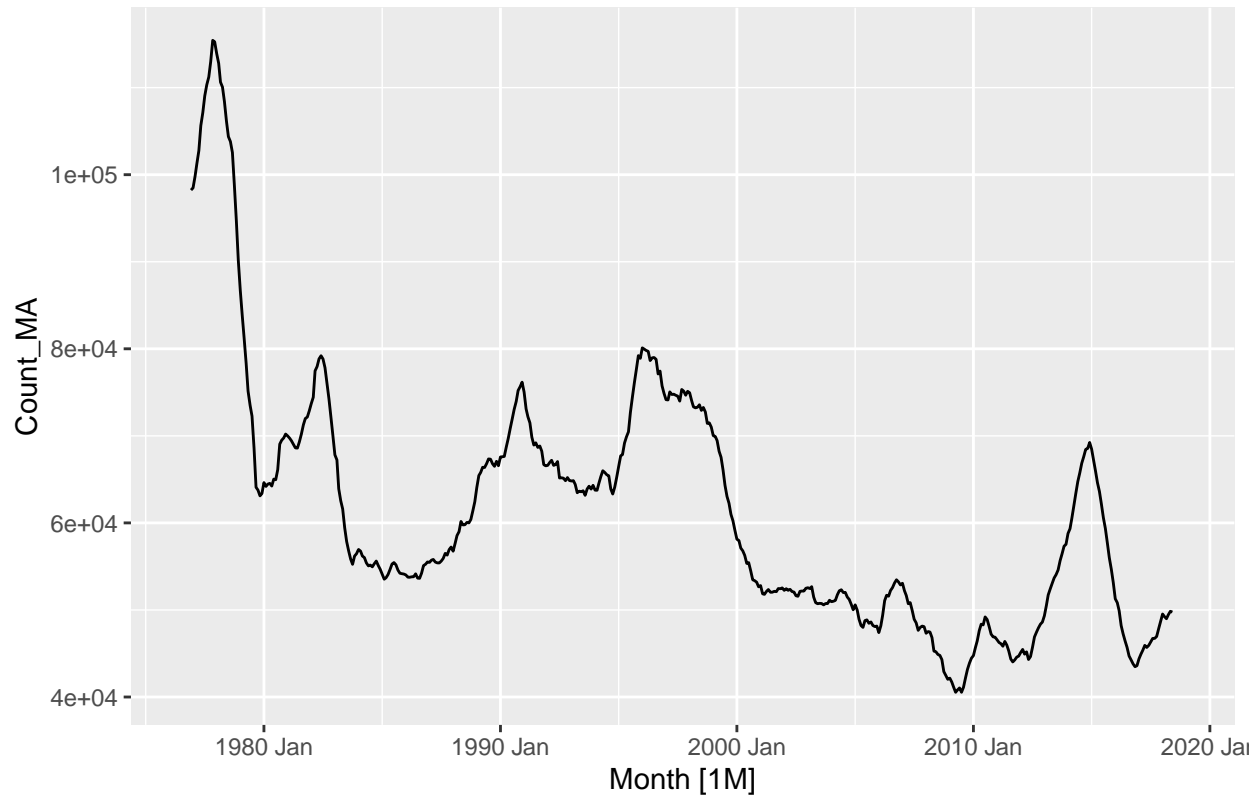
Here I transformed the monthly data into monthly average to make the pattern more consistent.

```r
# Filter the data for "Bulls, bullocks and steers" in Victoria
aus_livestock_vic <- aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers", State == "Victoria") %>%
  mutate(Count = as.numeric(Count)) %>%
  filter(Count > 0)

# Step 1: Calculate optimal lambda for Box-Cox transformation
lambda <- BoxCox.lambda(aus_livestock_vic$Count)

# Step 2: Apply Box-Cox transformation
aus_livestock_vic <- aus_livestock_vic %>%
  mutate(Count_BoxCox = BoxCox(Count, lambda))

# Step 3: Apply first differencing to remove the trend
aus_livestock_vic <- aus_livestock_vic %>%
  mutate(Count_diff1 = difference(Count_BoxCox, lag = 1))

# Step 4: Apply seasonal differencing (for seasonality removal, e.g., 12 months)
aus_livestock_vic <- aus_livestock_vic %>%
  mutate(Count_diff_seasonal = difference(Count_diff1, lag = 12))

# Step 5: Apply a 12-month moving average after all transformations
aus_livestock_vic <- aus_livestock_vic %>%
  mutate(Count_MA_diff_seasonal = zoo::rollmean(Count_diff_seasonal, k = 12, fill = NA))
```
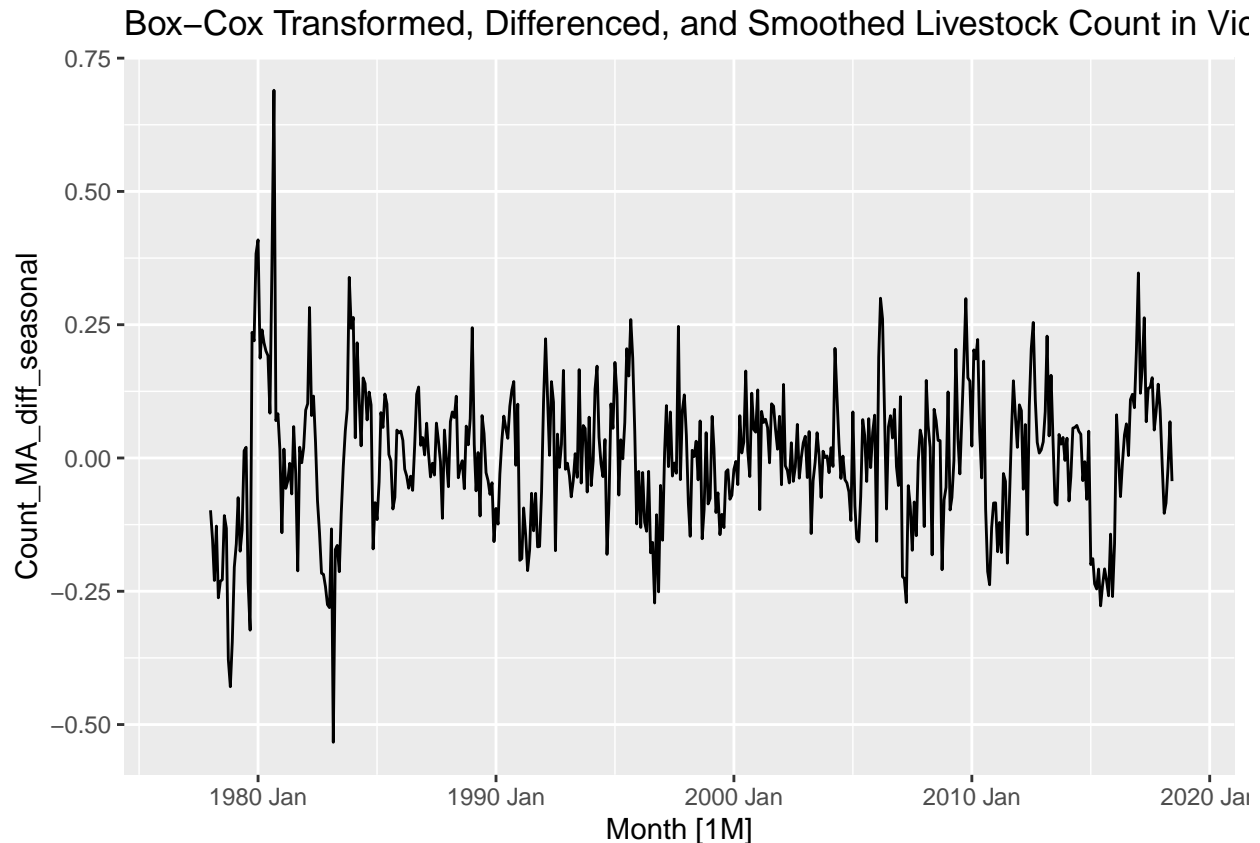
```
# Step 6: Plot the final result
autoplot(aus_livestock_vic, Count_MA_diff_seasonal) +
  ggtitle("Box-Cox Transformed, Differenced, and Smoothed Livestock Count in Victoria")
```

Box–Cox Transformed, Differenced, and Smoothed Livestock Count in Vic

To better analyze and interpret this data, I was able to apply moving average, which is much more readable. I tried everything to be able to transform the constant downward trend in which it was impossible to have both the excessive fluctuation and downward trend transformed together. (I am opened to suggestions.)

It's possible that the downward trend in livestock data could reflect broader societal shifts, including changes in dietary habits, such as the rising popularity of vegetarianism or veganism in Australia. Over recent years, there has been growing awareness around plant-based diets, driven by health concerns, environmental issues, and ethical considerations surrounding animal welfare. This could influence demand for meat and subsequently lead to a decline in livestock numbers being raised or slaughtered.

However, there could be other factors influencing the downward trend, such as:

Economic Changes: A decrease in demand for meat due to economic factors, such as reduced consumer spending or changes in export demand, could affect livestock numbers. Technological Improvements: Advances in farming efficiency may lead to fewer animals being raised but with higher productivity per animal. Regulatory or Environmental Policies: Changes in agricultural or environmental policies could influence livestock farming, including incentives to reduce carbon emissions or land use. Global Market Shifts: Fluctuations in global meat markets could lead to reduced livestock production in Australia, especially if exports decline or competitors become more efficient. Climate Impact: Droughts and other climate-related factors might affect grazing land availability, leading to lower livestock numbers over time. It's important to analyze multiple possible factors alongside the data to understand the full context of the trend. You could also look into broader economic, social, and environmental reports from Australia during the period covered by your dataset to assess whether dietary shifts are a primary driver, or if other factors are at play.

Victorian Electricity Demand from vic_elec

```
# View the first few rows of the aus_production dataset
head(vic_elec)
```

```
## # A tsibble: 6 x 5 [30m] <Australia/Melbourne>
##   Time                Demand Temperature Date       Holiday
##   <dttm>               <dbl>       <dbl> <date>     <lgl>
## 1 2012-01-01 00:00:00  4383.        21.4 2012-01-01 TRUE
## 2 2012-01-01 00:30:00  4263.        21.0 2012-01-01 TRUE
## 3 2012-01-01 01:00:00  4049.        20.7 2012-01-01 TRUE
## 4 2012-01-01 01:30:00  3878.        20.6 2012-01-01 TRUE
## 5 2012-01-01 02:00:00  4036.        20.4 2012-01-01 TRUE
## 6 2012-01-01 02:30:00  3866.        20.2 2012-01-01 TRUE
```

```
# Filter the data and plot
vic_elec %>%
  autoplot() +
  labs(title = "Half-hourly electricity demand for Victoria, Australia")
```
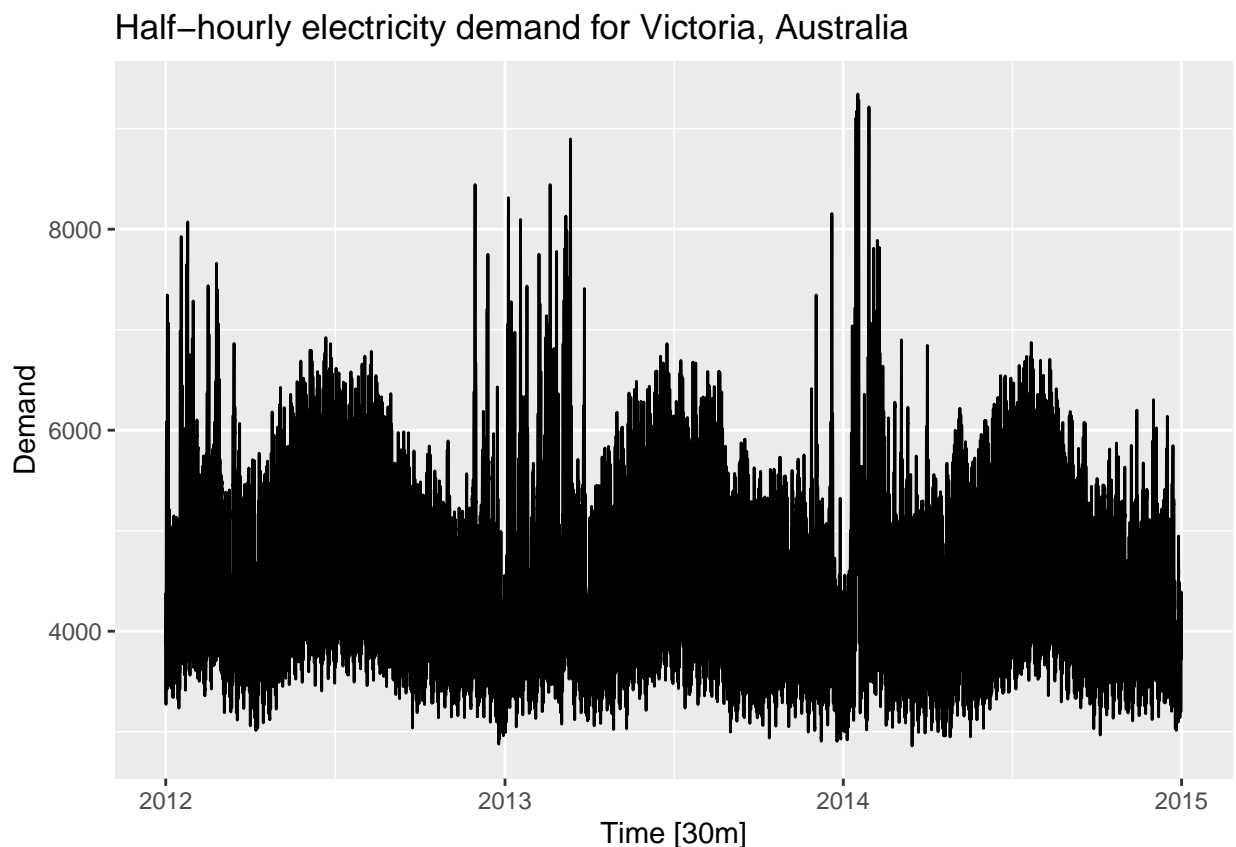

The excessive fluctuations in the vic_elec dataset are most likely due to a combination of predictable seasonal patterns, random short-term noise, and possible long-term trends. By applying transformations like moving averages, seasonal decomposition, and Box-Cox transformations, you can reduce the noise and focus on the meaningful underlying patterns in the data. These transformations are necessary to help make the dataset more interpretable and to improve the accuracy of any subsequent analysis or forecasting.

Since the data is collected at a high frequency (half-hourly), it captures more detail, including short-term fluctuations. These fluctuations are often smoothed out when data is aggregated over longer time periods (e.g., daily or monthly). The high frequency of the data makes it prone to exhibit higher levels of noise.

```r
# Convert the half-hourly data to a daily average
vic_elec_daily_avg <- vic_elec %>%
  index_by(Date = as_date(Time)) %>%  # Group by Date
  summarise(Daily_Avg_Demand = mean(Demand))  # Calculate the daily average of Demand

# Plot the daily average electricity demand
vic_elec_daily_avg %>%
  autoplot(Daily_Avg_Demand) +
  ggtitle("Daily Average Electricity Demand in Victoria") +
  xlab("Date") +
  ylab("Electricity Demand (MW)")
```
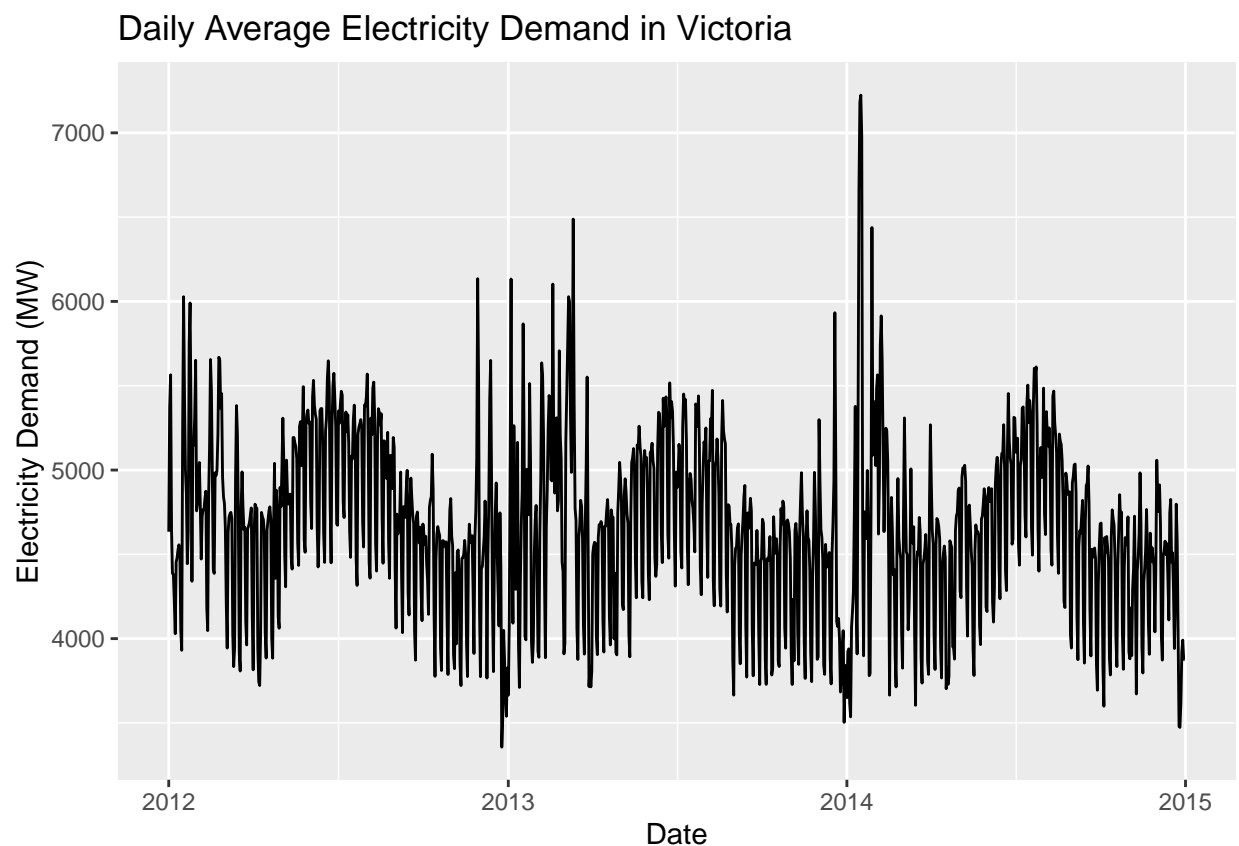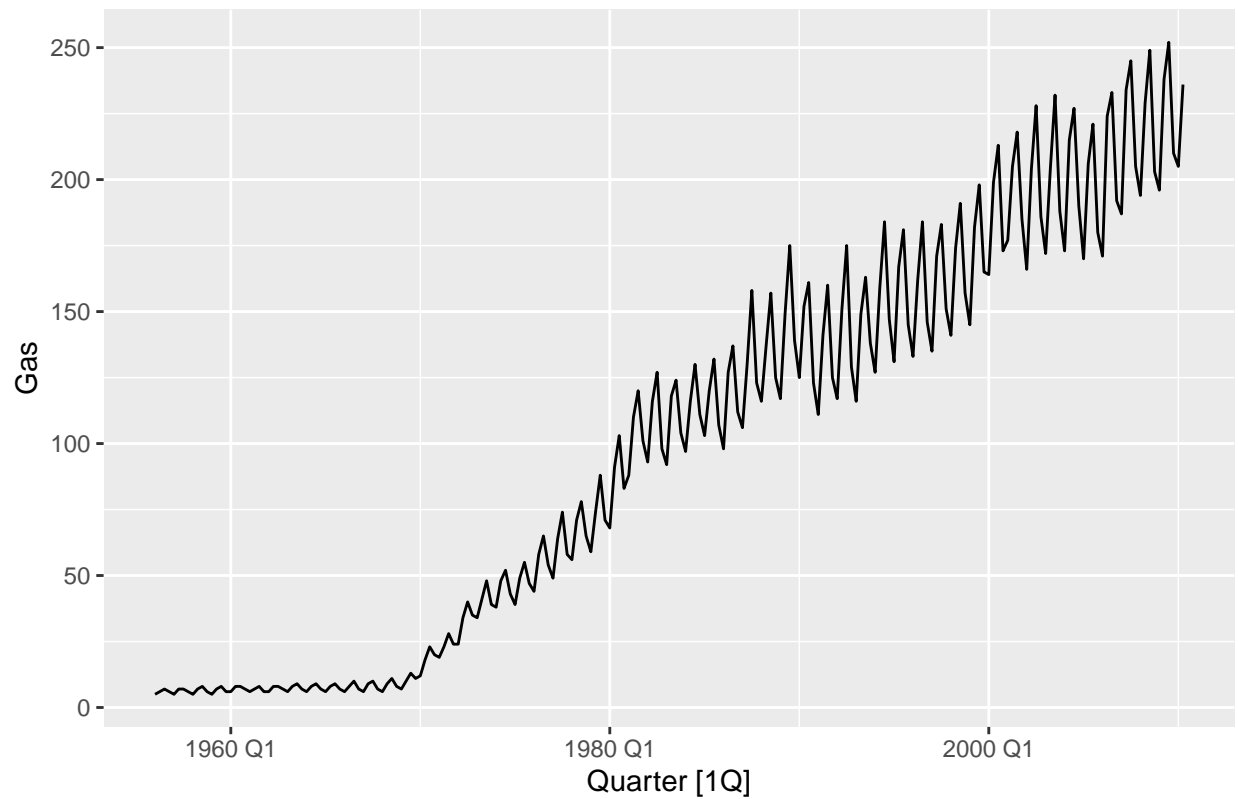

This method aggregates the half-hourly data into daily averages, providing a clearer view of the overall trend and reducing short-term fluctuations.

## Gas production from aus_production.

```r
# Filter the data and plot
aus_production %>%
  autoplot(Gas) +
  labs(title = "Quarterly production of Gas Production in Australia")
```

Quarterly production of Gas Production in Australia


```r
lambda <- aus_production |>
  features(Gas, features = guerrero) |>
  pull(lambda_guerrero)
aus_production |>
  autoplot(box_cox(Gas, lambda)) +
  labs(y = "",
       title = paste("Transformed Gas Production with Lambda = ", round(lambda,2)))
```

Transformed Gas Production with Lambda = 0.11
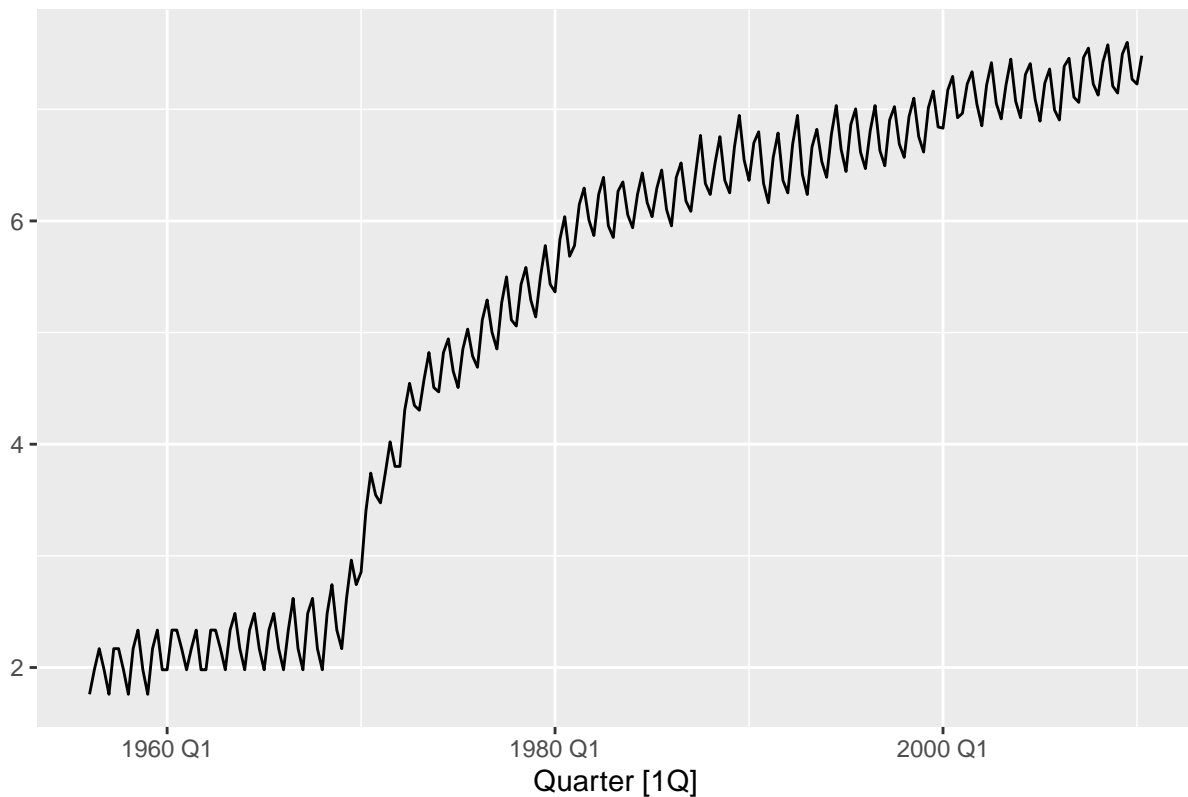
The Box-Cox transformation is useful for stabilizing the variance in time series data, making the series more linear and suitable for analysis. It is particularly helpful when you have a non-constant variance or non-normal distribution in the original data. I transformed Australian quarterly gas production with the lambda parameter chosen using the Guerrero method. (This code was provided in the Forcasting: Principles and Practice text, Chapter 3.1)
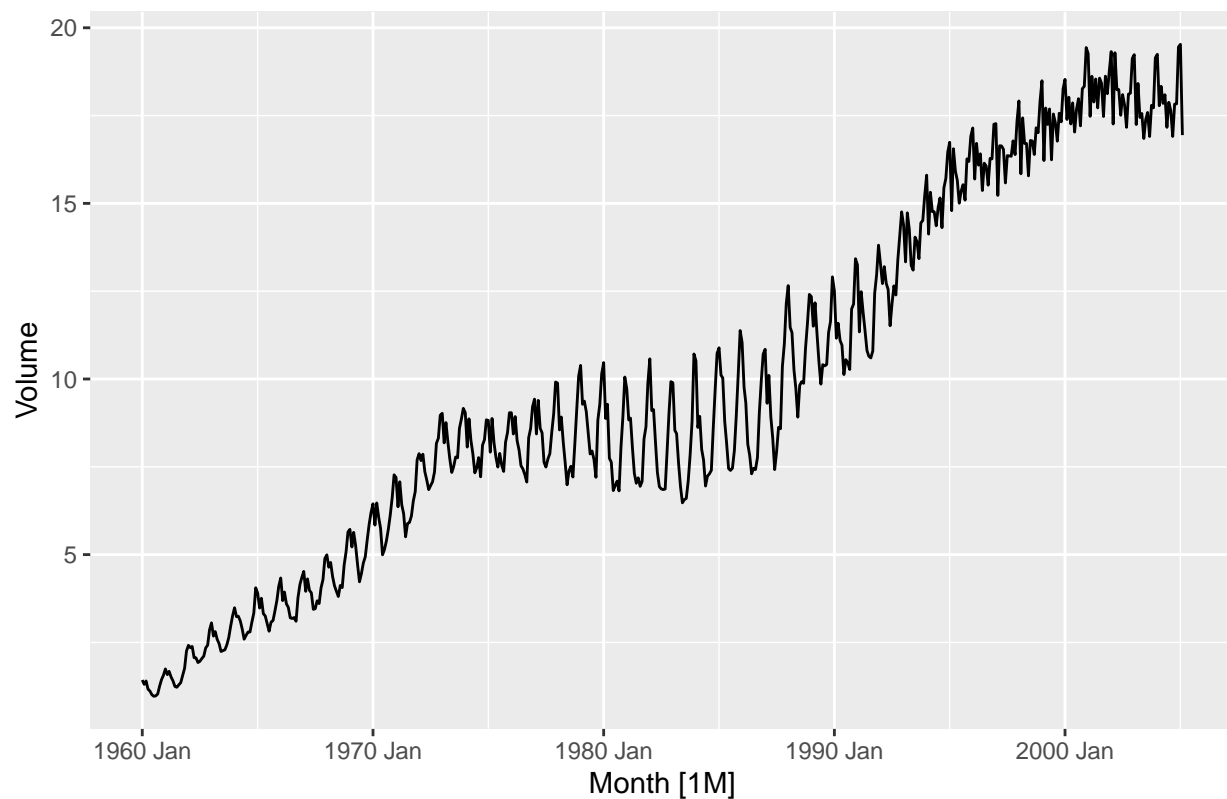
3. Why is a Box-Cox transformation unhelpful for the canadian_gas data?

```
head(canadian_gas)
```

```
## # A tsibble: 6 x 2 [1M]
##       Month Volume
##       <mth>  <dbl>
## 1 1960 Jan    1.43
## 2 1960 Feb    1.31
## 3 1960 Mar    1.40
## 4 1960 Apr    1.17
## 5 1960 May    1.12
## 6 1960 Jun    1.01
```

```
# Filter the data and plot
canadian_gas %>%
  autoplot() +
  labs(title = "Monthly Canadian gas production")
```

Monthly Canadian gas production


```
lambda <- canadian_gas %>%
  features(Volume, features = guerrero) %>%
  pull(lambda_guerrero)
canadian_gas %>%
  autoplot(box_cox(Volume, lambda)) +
  labs(y = "",
       title = paste("Transformed Canadian Gas volume with lambda = ", round(lambda,2)))
```

Transformed Canadian Gas volume with lambda = 0.58
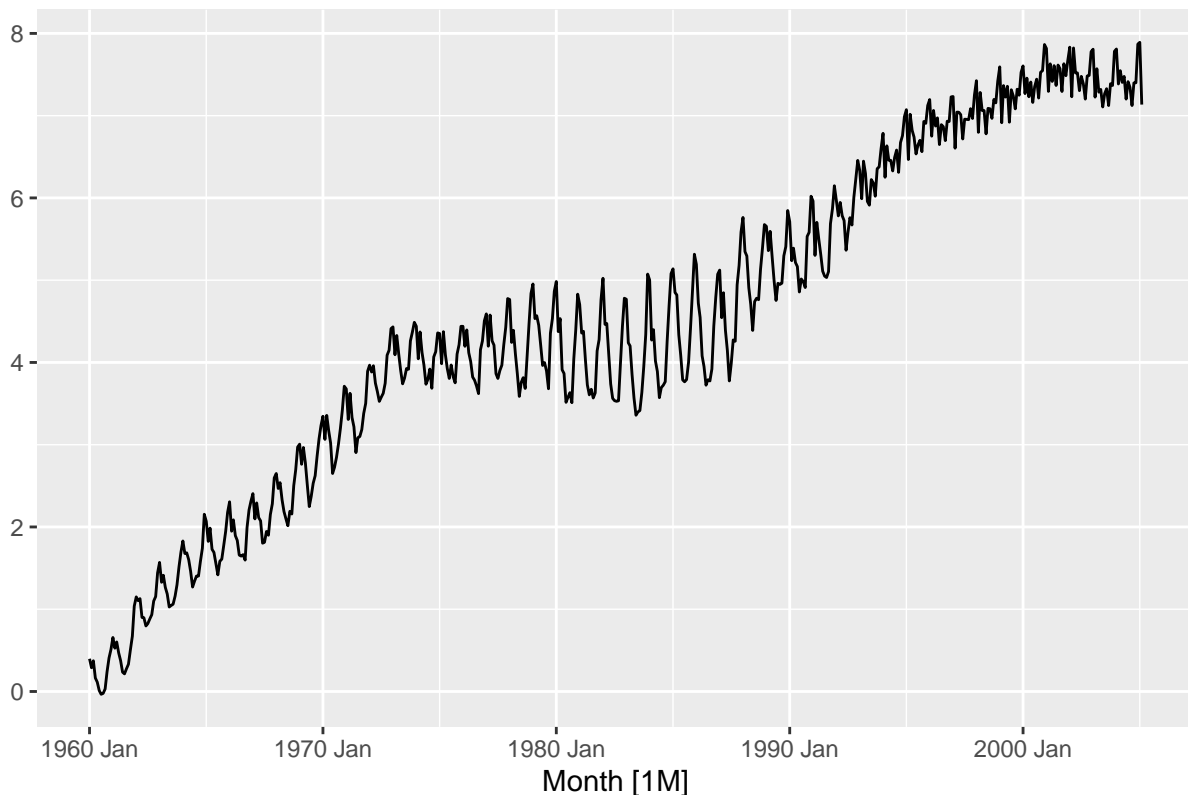
A Box-Cox transformation is generally considered unhelpful for the "canadian_gas" data because the variance first increases then decreases, it suggests a non-monotonic relationship between the variance and the mean, which is not something the Box-Cox transformation can handle effectively. The transformation assumes a simpler relationship, and it won't be able to fully capture or correct this behavior.

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

```
# View the first few rows of the aus_production dataset
head(aus_production)
```

```
## # A tsibble: 6 x 7 [1Q]
##    Quarter  Beer Tobacco Bricks Cement Electricity   Gas
##      <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
## 1 1956 Q1    284    5225    189    465        3923     5
## 2 1956 Q2    213    5178    204    532        4436     6
## 3 1956 Q3    227    5297    208    561        4806     7
## 4 1956 Q4    308    5681    197    570        4418     6
## 5 1957 Q1    262    5577    187    529        4339     5
## 6 1957 Q2    228    5651    214    604        4811     7
```
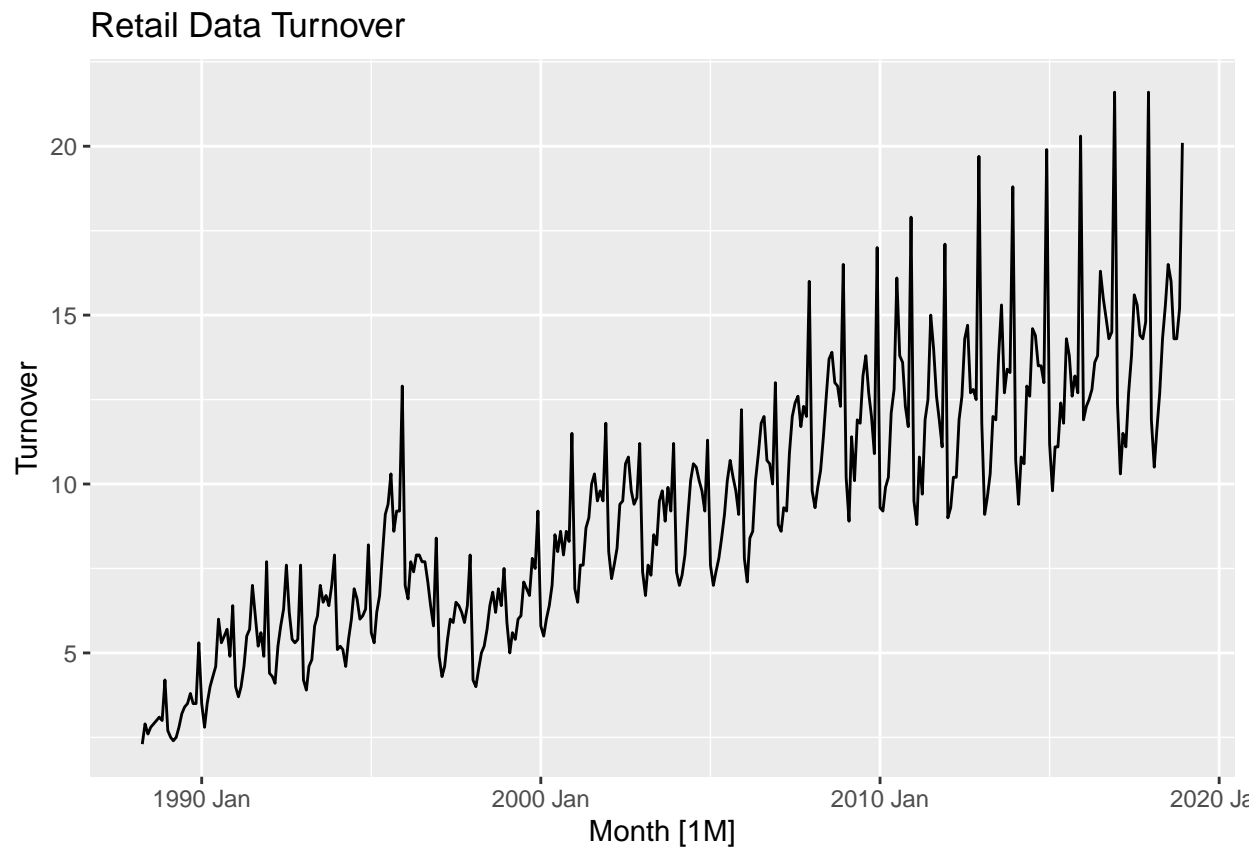
```
set.seed(12345678)
myseries <- aus_retail |>
```

```
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
autoplot(myseries, Turnover) +
  labs(title = "Retail Data Turnover")
```

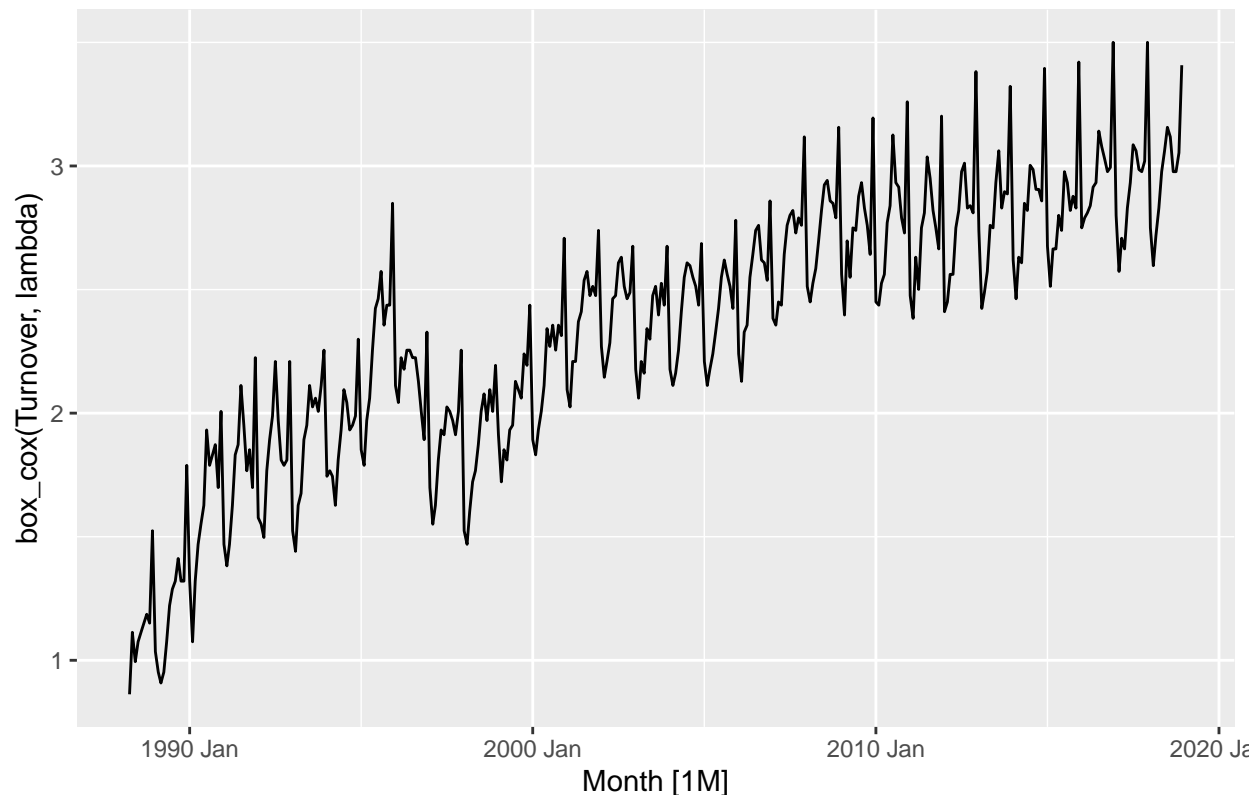## Retail Data Turnover


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

myseries %>% autoplot(box_cox(Turnover, lambda))+
  labs(title = paste("Transformed Retail Turnover with lambda round(lambda, 2)"))
```

## Transformed Retail Turnover with lambda round(lambda, 2)

With the Box-Cox transformation with lambda = 0.8, we can see a more uniform seasonal variation. Box-Cox transformation was used since it uses a natural logarithm for exponential growth as well as using Guerrero found a good value of lambda to help making the forecasting simpler.

5. For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance. Tobacco from aus_production, Economy class passengers between Melbourne and Sydney from ansett, and Pedestrian counts at Southern Cross Station from pedestrian.

```
#Tobacco from `aus_production`
aus_production %>% autoplot(Tobacco) +
  labs(y = "Production in Tones",
       title = "Quarterly production of Tobacco in Australia.")
```

Quarterly production of Tobacco in Australia.
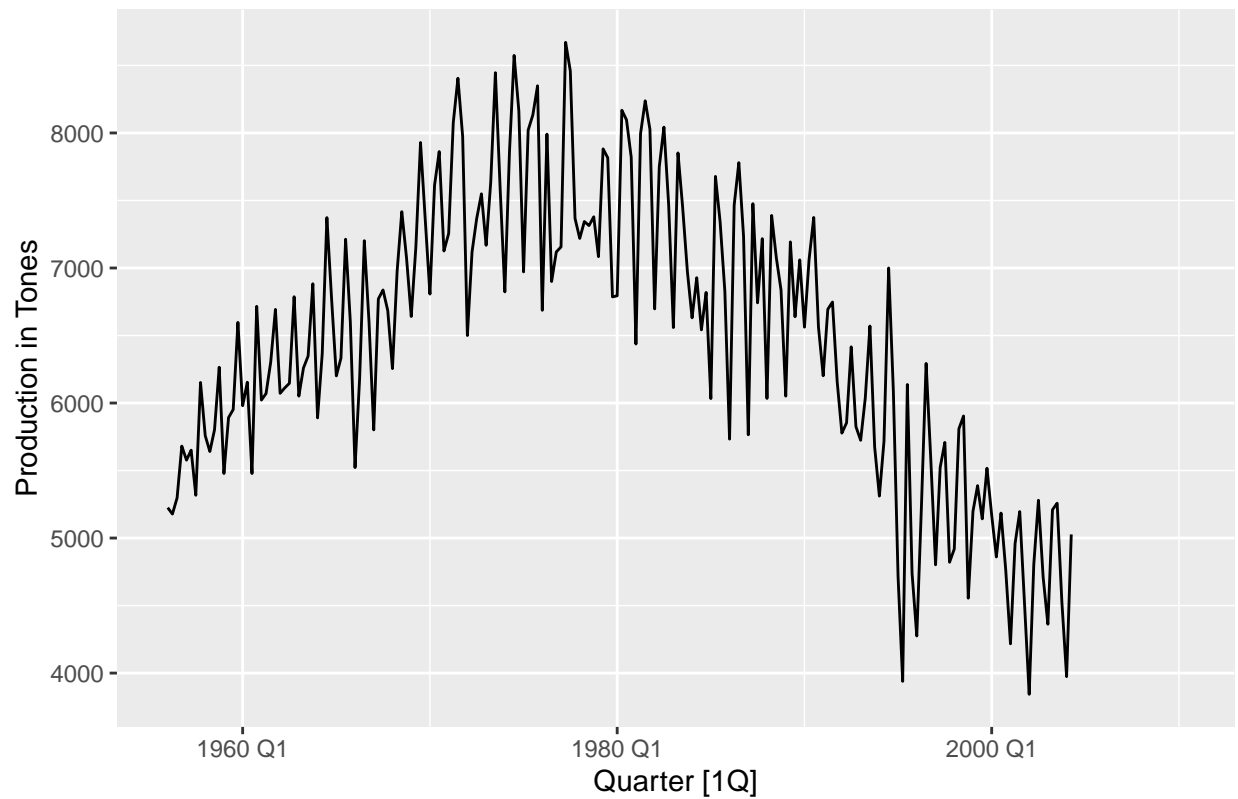

```
lambda <- aus_production %>%
  features(Tobacco, features = guerrero) %>%
  pull(lambda_guerrero)
aus_production %>%
  autoplot(box_cox(Tobacco, lambda)) +
  labs(y = "",
       title = paste("Box-Cox Transformation with lambda = ", round(lambda,2)))
```

Box−Cox Transformation with lambda = 0.93
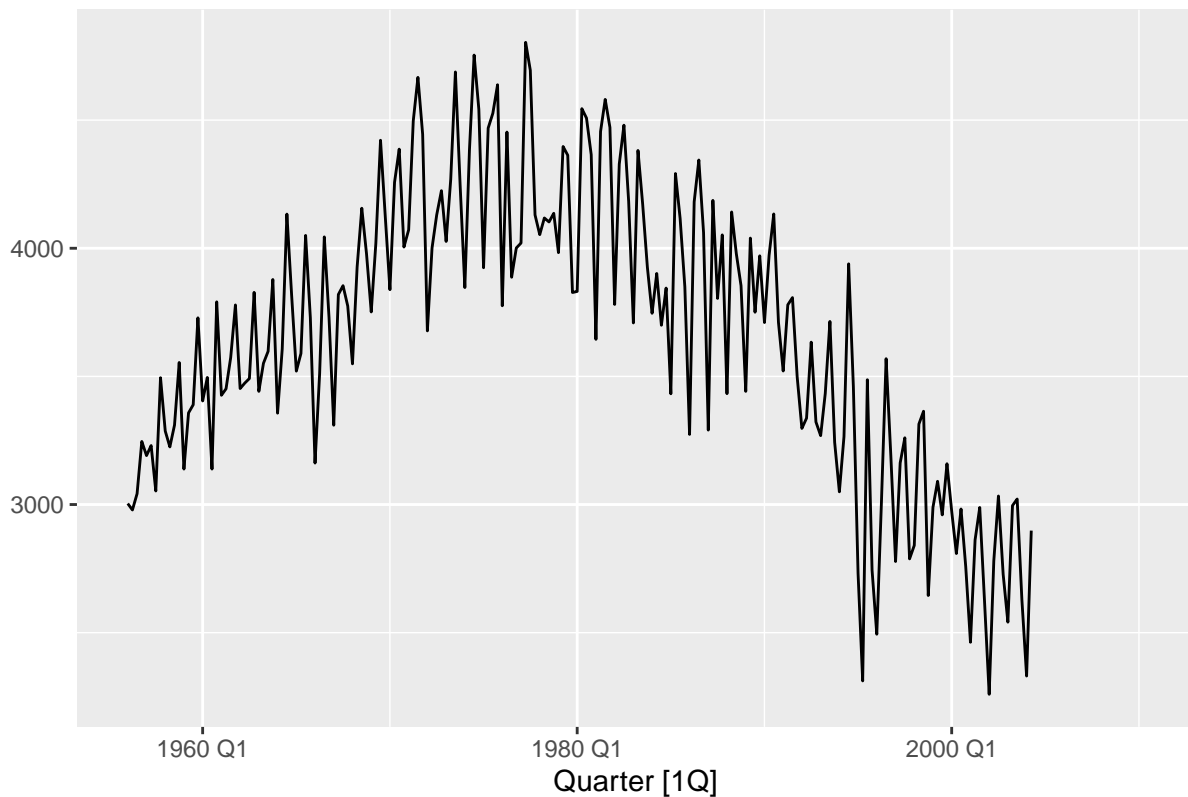
For the Tobacco from aus_production dataset, we see that Box-Cox transformation we have a lambda value of 0.93 which signifies that there was barely a transformation in data.

```
head(ansett)
```

```
## # A tsibble: 6 x 4 [1W]
## # Key:       Airports, Class [1]
##       Week Airports Class    Passengers
##     <week> <chr>    <chr>         <dbl>
## 1 1989 W28 ADL-PER  Business        193
## 2 1989 W29 ADL-PER  Business        254
## 3 1989 W30 ADL-PER  Business        185
## 4 1989 W31 ADL-PER  Business        254
## 5 1989 W32 ADL-PER  Business        191
## 6 1989 W33 ADL-PER  Business        136
```

```
eco_mel_syd <- ansett %>%
  filter(Class == "Economy",
         Airports == "MEL-SYD")

autoplot(eco_mel_syd, Passengers)+
  labs(title = "Economy Class Passengers Between Melbourne and Sydney")
```
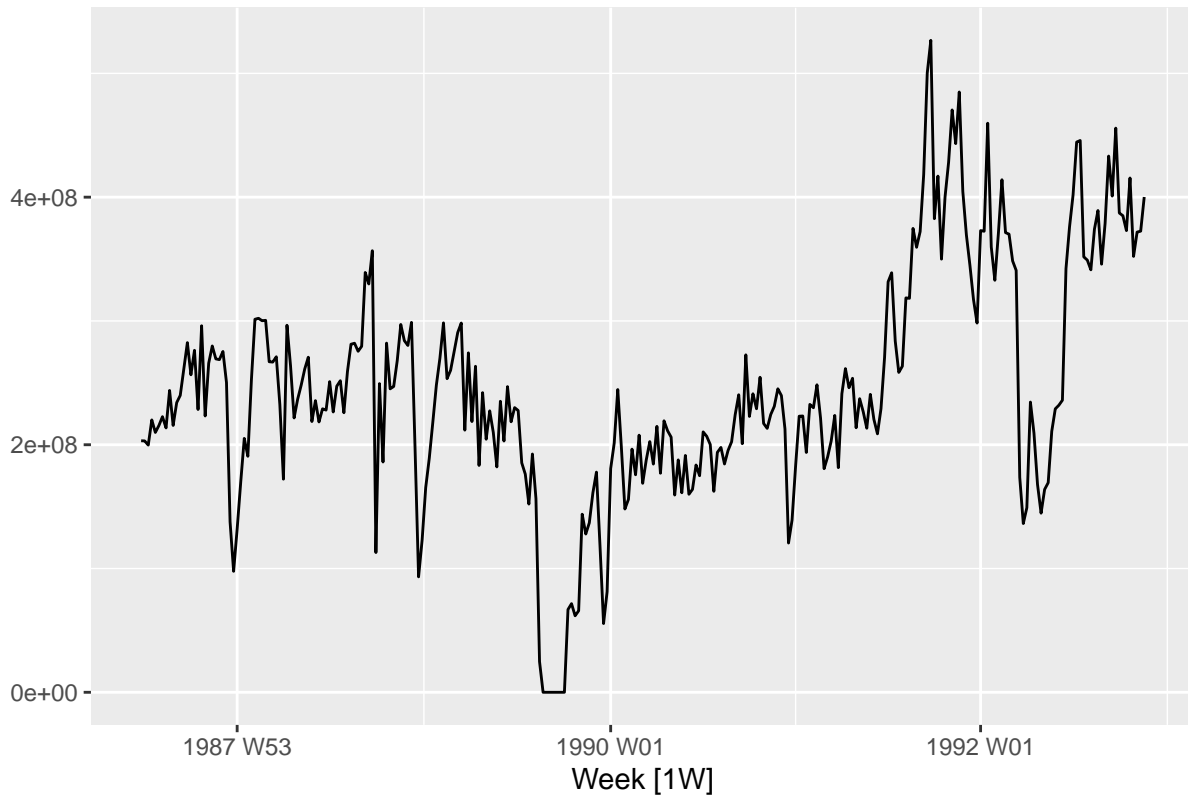
Economy Class Passengers Between Melbourne and Sydney
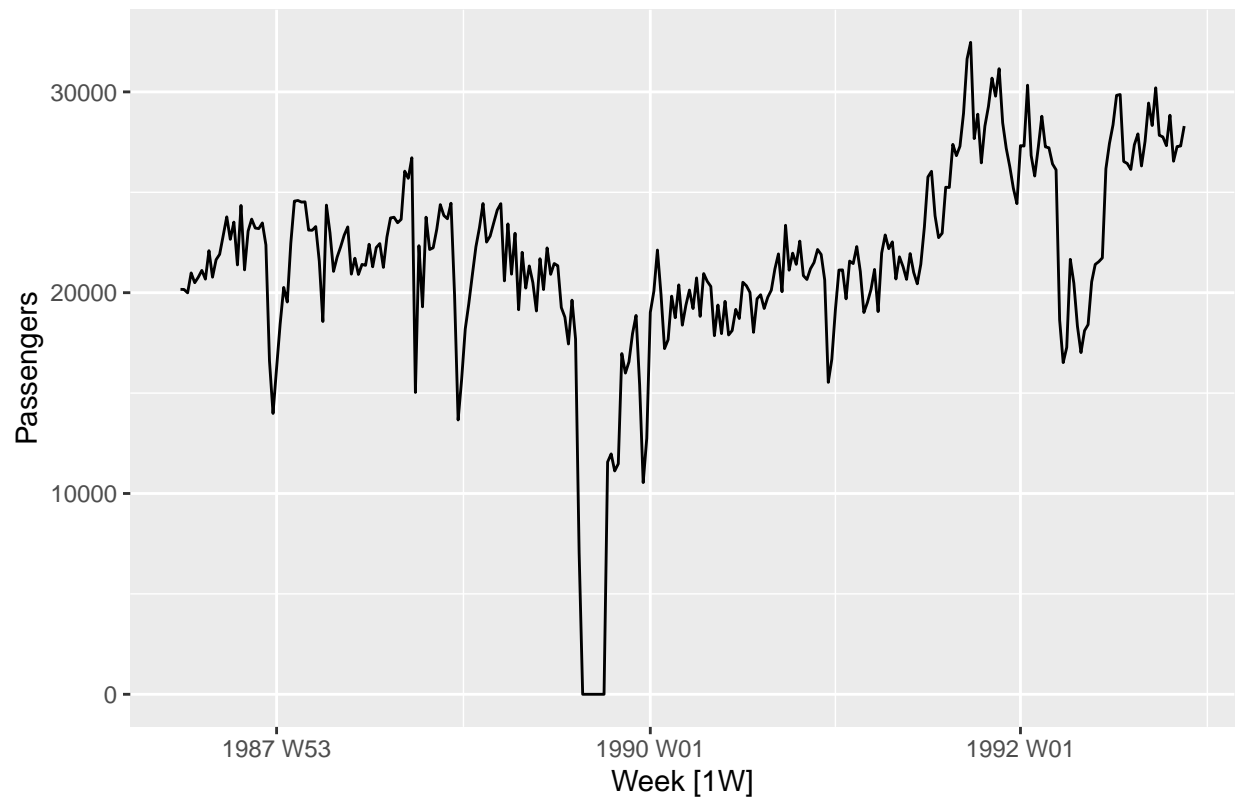

```r
lambda <- eco_mel_syd %>%
  features(Passengers, features = guerrero) %>%
  pull(lambda_guerrero)
eco_mel_syd %>%
  autoplot(box_cox(Passengers, lambda)) +
  labs(y = "",
       title = paste("Box-Cox Transformation with lambda = ", round(lambda,2)))
```

Box–Cox Transformation with lambda = 2

In the Economy class passengers between Melbourne and Sydney from ansett dataset, we have a Box-Cox transformation with a lambda value of 2 indicating that the data is Squared in order to show variations better.

```
head(pedestrian)
```

```
## # A tsibble: 6 x 5 [1h] <Australia/Melbourne>
## # Key:        Sensor [1]
##   Sensor         Date_Time           Date       Time Count
##   <chr>          <dttm>              <date>     <int> <int>
## 1 Birrarung Marr 2015-01-01 00:00:00 2015-01-01     0  1630
## 2 Birrarung Marr 2015-01-01 01:00:00 2015-01-01     1   826
## 3 Birrarung Marr 2015-01-01 02:00:00 2015-01-01     2   567
## 4 Birrarung Marr 2015-01-01 03:00:00 2015-01-01     3   264
## 5 Birrarung Marr 2015-01-01 04:00:00 2015-01-01     4   139
## 6 Birrarung Marr 2015-01-01 05:00:00 2015-01-01     5    77
```
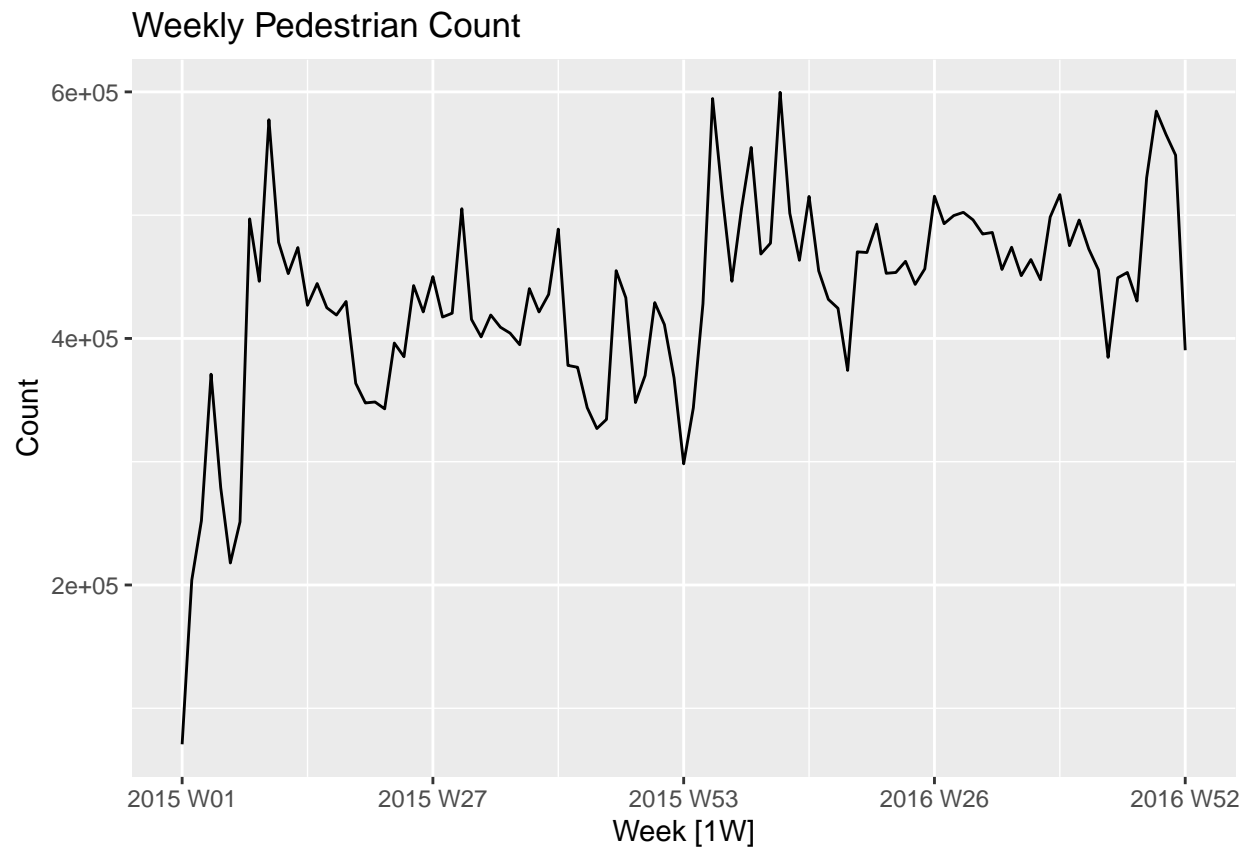
```
pedestrian %>% filter(Sensor =='Southern Cross Station') %>% autoplot(Count) +
  labs(title = "Pedestrian Counts at Southern Cross Station")
```

Pedestrian Counts at Southern Cross Station
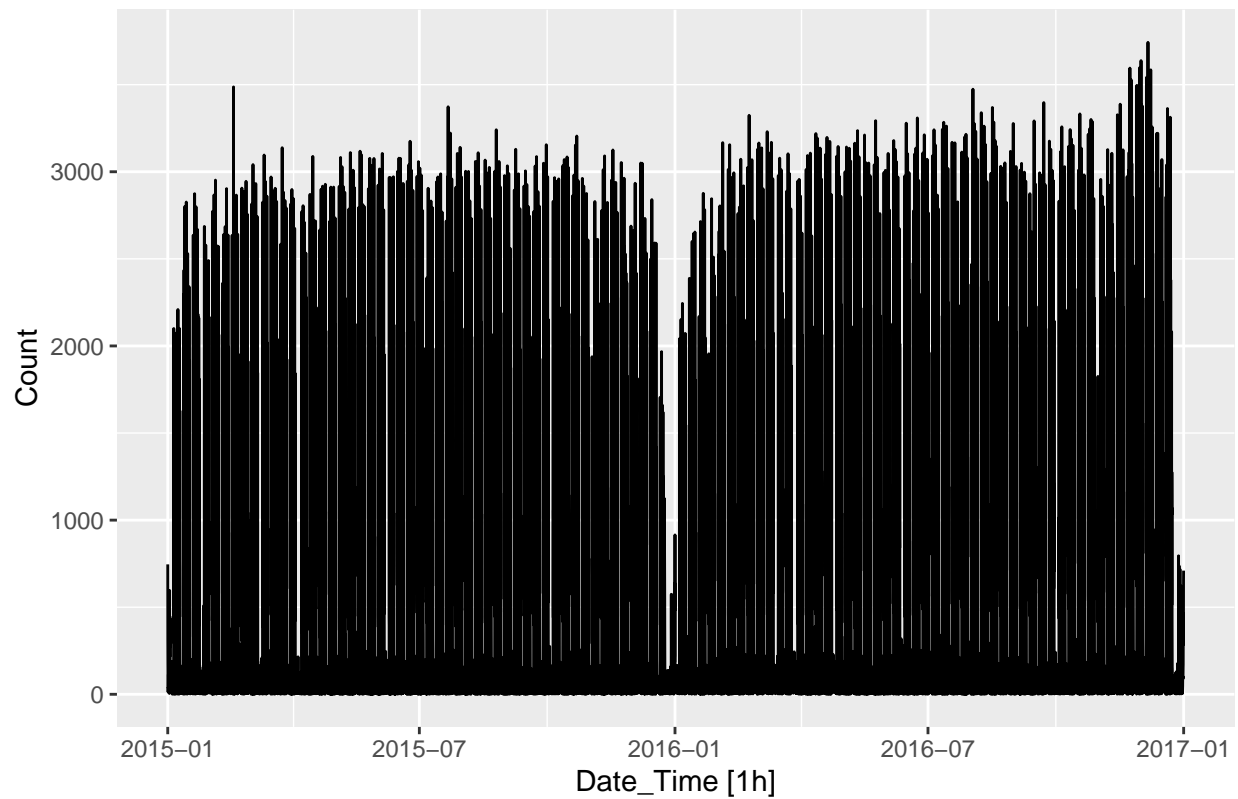


```
weekly <- pedestrian %>%
  mutate(Week = yearweek(Date)) %>%
  index_by(Week) %>%
  summarise(Count = sum(Count))

weekly %>% autoplot(Count)+
  labs(title = "Weekly Pedestrian Count")
```

Weekly Pedestrian Count


```
lambda <- weekly %>%
  features(Count, features = guerrero) %>%
  pull(lambda_guerrero)

weekly %>% autoplot(box_cox(Count,lambda)) +
  labs(title = paste("Transformed Weekly Pedestrian Count with lambda =", round(lambda, 2)))
```

Transformed Weekly Pedestrian Count with lambda = 2
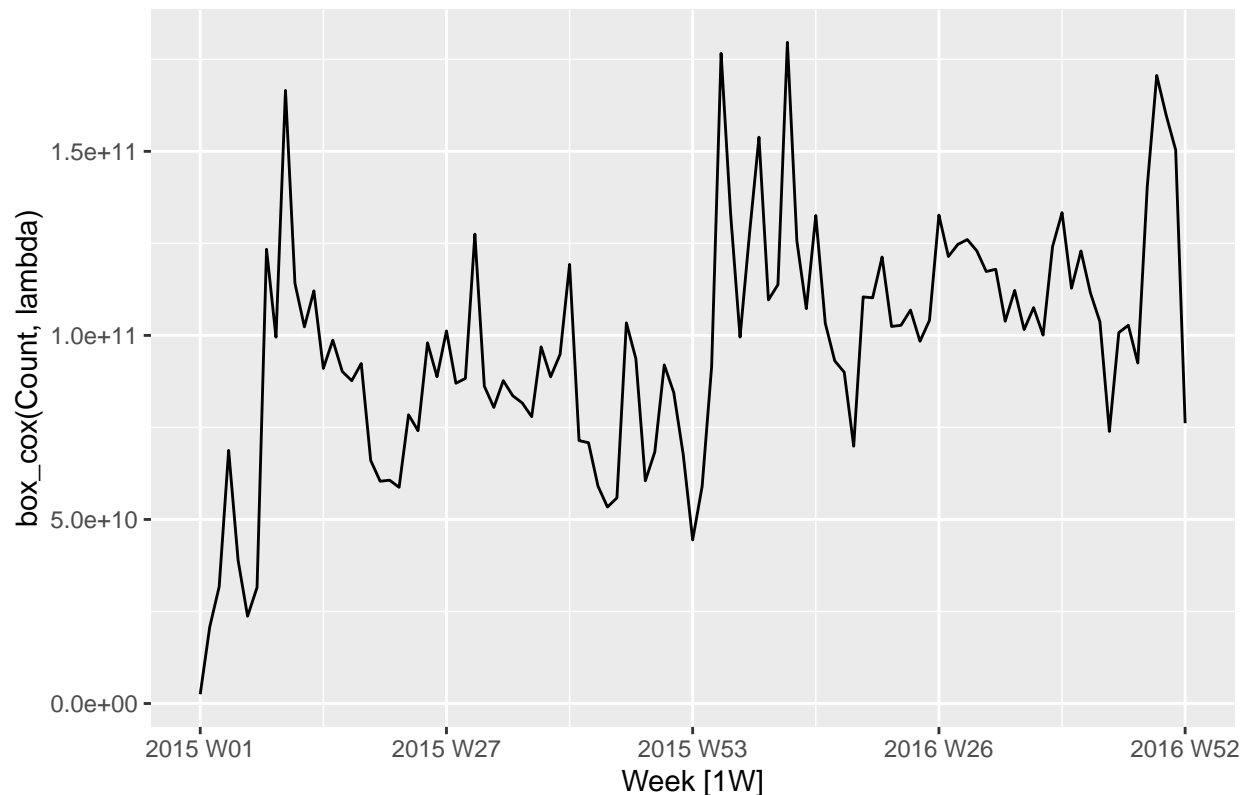
The Box-Cox Transformation appears to apply a power transformation with a lambda value of 2. However, the significant adjustment was aggregating the data from an hourly pedestrian count to a weekly count, as the overall timeframe spans from January 2015 to December 2016. The original dataset had an excessive number of data points per row, leading to implied variance and considerable noise.

7. Consider the last five years of the Gas data from aus_production.

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

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

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

```
gas %>% autoplot(Gas) +
  labs(title = "Australia Gas Production", y = "Petajoules")
```

Australia Gas Production

The seasonal fluctuations display a recurring pattern where Q1 starts off low, followed by an increase in Q2, peaking in Q3, and then decreasing again in Q4. This cycle appears to repeat annually. Despite these seasonal variations, there is an overall upward trend, suggesting consistent growth alongside the predictable cyclical shifts throughout the quarters.

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

```
class_decomp <- gas %>%
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) %>%
  components()
class_decomp %>% autoplot() +
  labs(title = "Classical multiplicative decomposition of Australia
                Gas Production")
```

Classical multiplicative decomposition of Australia Gas Production

Gas = trend * seasonal * random
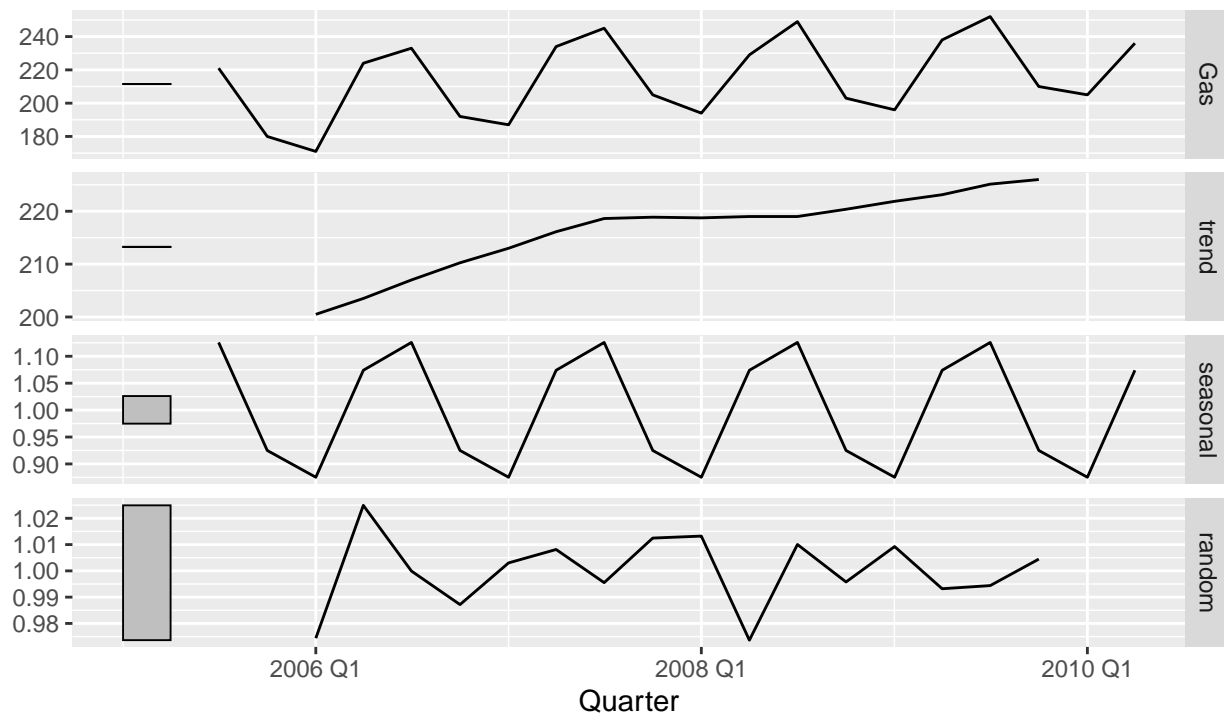
c. Do the results support the graphical interpretation from part a?

Yes, the results from the decomposition support the graphical interpretation from part a. The decomposition analysis breaks down the time series into its components—trend, seasonality, and residuals—confirming the upward trend and the recurring seasonal pattern that were observed visually. The trend component clearly shows a consistent upward trajectory, while the seasonal component captures the repetitive cyclical fluctuations across specific periods (e.g., quarters). These findings align with the initial visual observation, reinforcing the presence of both long-term growth and periodic variations in the data.

d. Compute and plot the seasonally adjusted data.

```
as_tsibble(class_decomp) %>%
  autoplot(season_adjust) +
  labs(title = "Seasonally Adjusted Data")
```
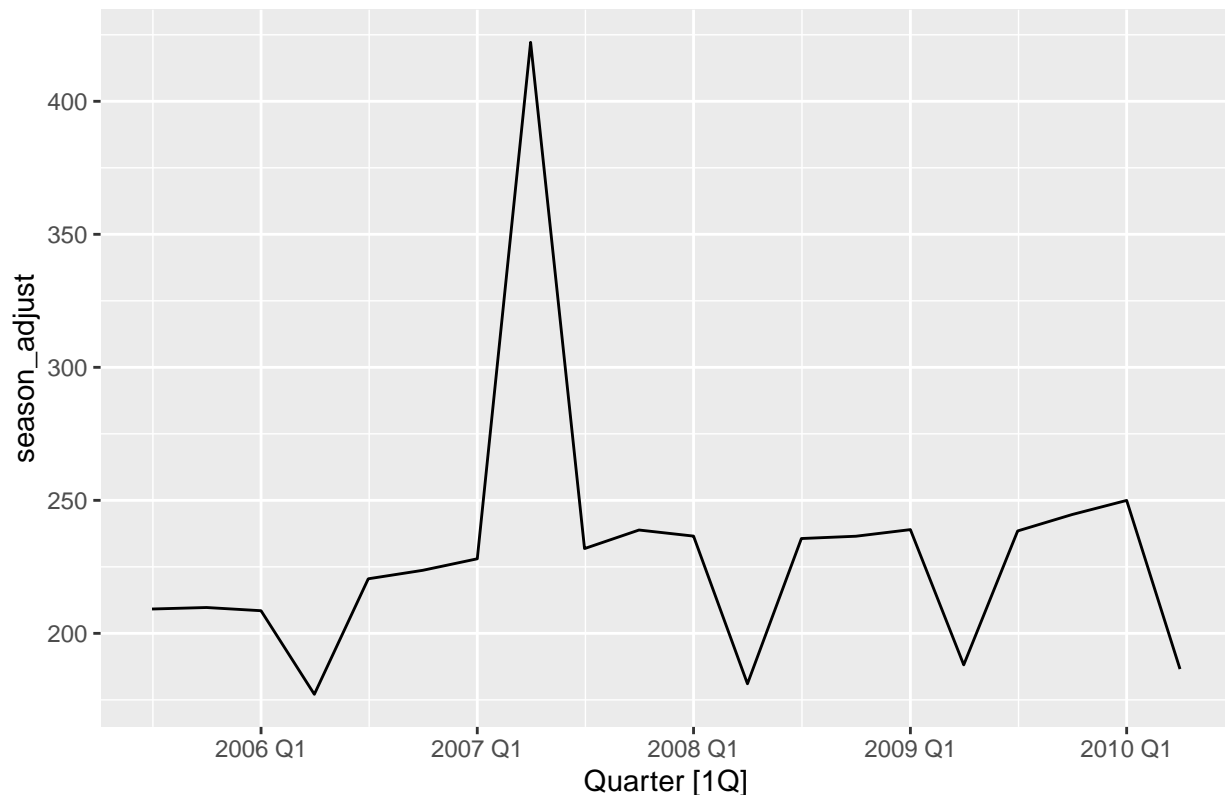
Seasonally Adjusted Data
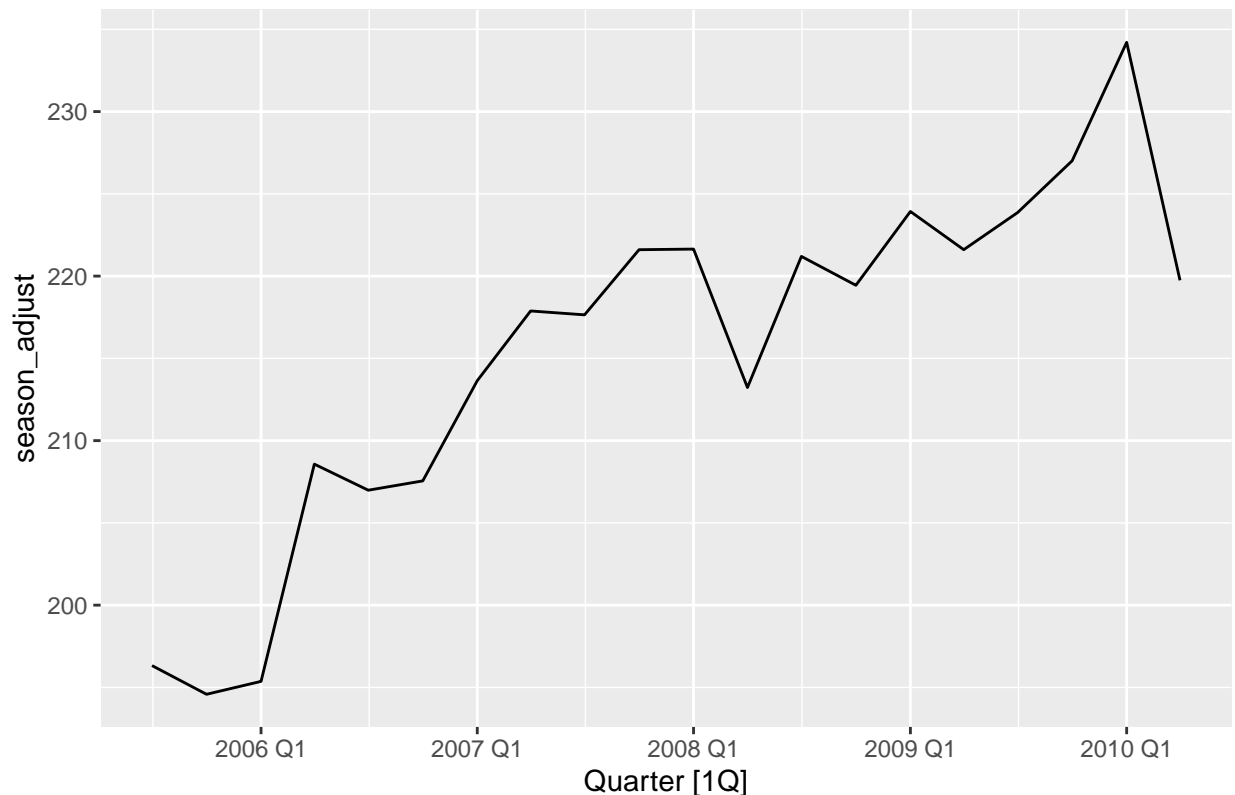
The seasonally adjusted data does show significantly less variation, which is exactly what we would expect. By removing the seasonal component, we are left with data that highlights the underlying trend and residuals, minimizing the periodic fluctuations that occur at regular intervals. This reduction in variation allows for a clearer analysis of the trend and any irregular patterns, making it easier to identify long-term changes and any anomalies in the data.

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?

```
gas %>%
  mutate(Gas = if_else(Quarter==yearquarter("2007Q2"), Gas + 300, Gas)) %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components() %>%
  as_tsibble() %>%
  autoplot(season_adjust) +
  labs(title = 'Seasonally Adjusted Data with 300 added to "2007 Q2"')
```

## Seasonally Adjusted Data with 300 added to "2007 Q2"

Adding the 300 outlier caused a significant spike in both the raw and seasonally adjusted data, which in turn impacted the overall trend of the dataset. This artificial increase disrupted the natural flow of the data, making the long-term trend appear more volatile than it actually is. Outliers like this can distort the analysis by creating a misleading representation of the underlying pattern, emphasizing the importance of handling such anomalies carefully in time series analysis.

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

```r
gas %>%
  mutate(Gas = if_else(Quarter==yearquarter("2010Q4"), Gas + 300, Gas)) %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components() %>%
  as_tsibble() %>%
  autoplot(season_adjust) +
  labs(title = 'Seasonally Adjusted Data with 300 added to "2010 Q4"')
```

## Seasonally Adjusted Data with 300 added to "2010 Q4"
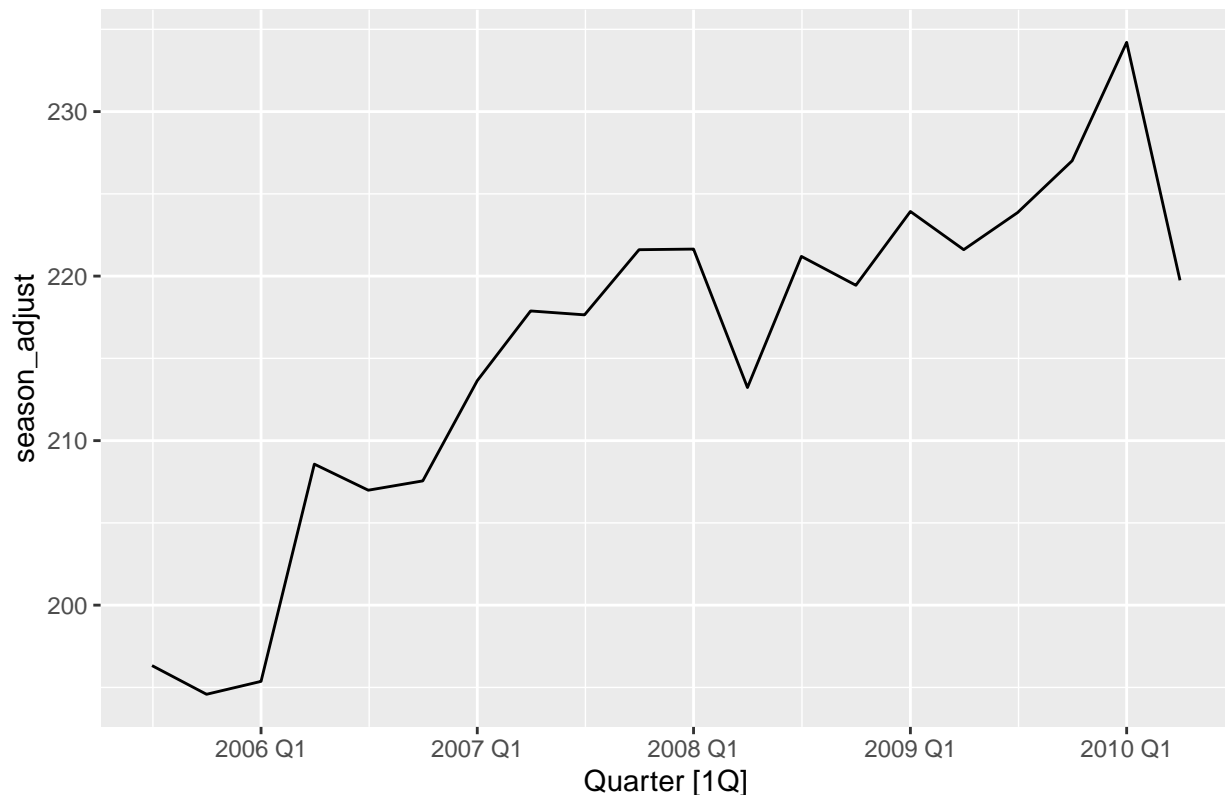
Yes, adding an outlier near the end of the time series rather than in the middle does make a difference. When the outlier is positioned towards the end, it can cause a more pronounced impact on the recent data, creating an extreme peak that distorts the pattern moving forward. In this case, placing the outlier at the beginning of Q4 results in an exaggerated upward trend, which may mislead interpretations of the overall trend and future projections. Outliers in later periods can have a stronger influence on trend analysis since they directly affect the recent data and can skew predictions based on the end of the series.

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?

```
x11_dcmp <- myseries %>%
  model(x11 = X_13ARIMA_SEATS(Turnover ~ x11())) %>%
  components()
autoplot(x11_dcmp) +
  labs(title = "Decomposition of total US retail employment using X-11.")
```

Decomposition of total US retail employment using X−11.

Turnover = trend * seasonal * irregular
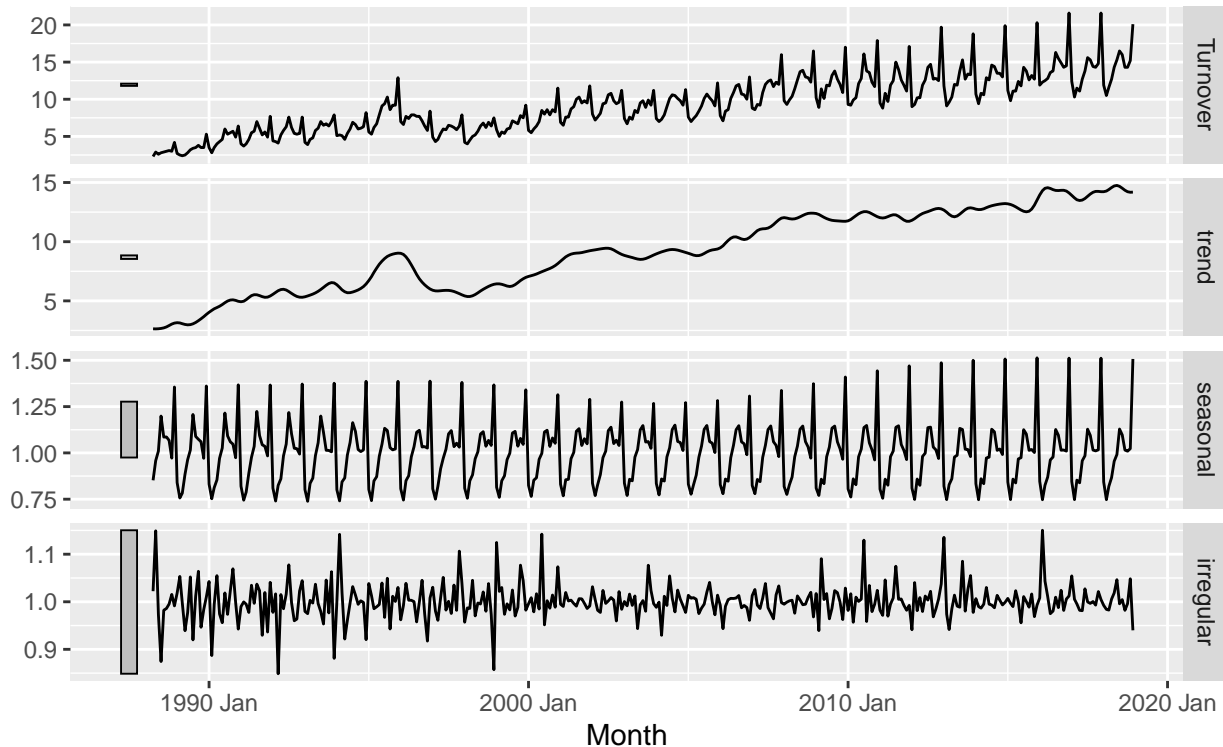
Decomposing the series using X-11 reveals some interesting outliers and unusual features that may not have been immediately apparent. Specifically, the irregular component of the decomposition shows a notable increase in noise around the early 1990s and again toward the end of the 2000s. These periods of heightened variability coincide with economic recessions, which likely caused the unexpected fluctuations in the data. Aside from these outliers, the decomposition presents a clear and consistent trend line, as well as a recognizable seasonal pattern throughout the turnover, reinforcing the cyclical nature of the data. The X-11 method helps in isolating these unusual features, allowing us to understand the external factors influencing the data more clearly.

9. Figures 3.19 and 3.20 show the result of decomposing the number of persons in the civilian labour force in Australia each month from February 1978 to August 1995.

Decomposition of the number of persons in the civilian labour force in Australia each month from February 1978 to August 1995.

Seasonal component from the decomposition shown in the previous figure.

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 decomposition of the civilian labor force in Australia from 1978 to 1995 reveals a stable upward trend, indicating steady growth over the years. The seasonal component appears consistent, with regular fluctuations occurring at similar intervals each year, suggesting that seasonality remained stable during this period.

However, the most striking feature is the sharp dip around 1991 and 1992, which aligns with a known recession during that time. This dip significantly impacted the overall trend, though the upward movement resumed after the recession passed. The scales of the graphs highlight these key changes, particularly the magnitude of the recession's impact.

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

Yes, the recession of 1991/1992 is visible in the estimated components. In figure 3.20, the seasonal component shows a sharp decrease during the March to August period of the early 1990s, which aligns with the recession. This seasonal dip is further supported by the overview provided in figure 3.19, where the STL decomposition reveals a significant decrease in the remainder (or noise) component during the 1990–1991 period. This drop in the remainder indicates unusual fluctuations that coincide with the broader economic downturn, reinforcing the observation of a recession during that time.