Data 624 Homework 2

Joao De Oliveira

2026-02-15

```
library(fpp3)
library(feasts)
```

Introduction

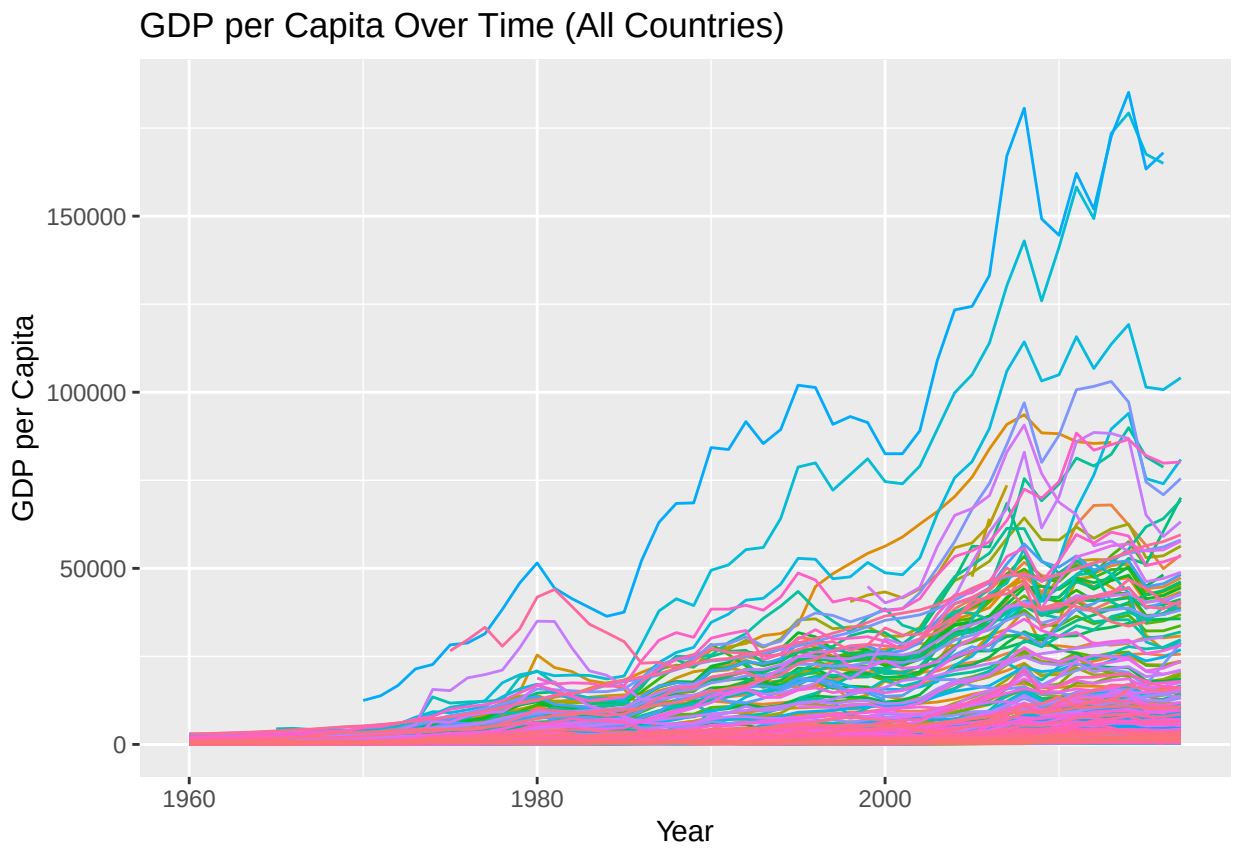
In this report, I will be answering questions 1, 2, 3, 4, 5, 7, 8 and 9 from the 3.7 section from the Hyndman online Forecasting book

Question 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?

Answer 1

```
# plot gdp per capita over time
# Create GDP per capita column (use a clear column name)
gdp_data <- global_economy %>%
  mutate(gdp_per_capita = GDP / Population)

# I removed the legend otherwise I couldn't see the plot
gdp_data %>%
  autoplot(gdp_per_capita) +
  labs(title = "GDP per Capita Over Time (All Countries)",
       x = "Year", y = "GDP per Capita") +
  theme(legend.position = "none")
```



The plot is extremely crowded. So, I will compute the highest GDP per capita and then I will plot the top 10 GDP per capita.

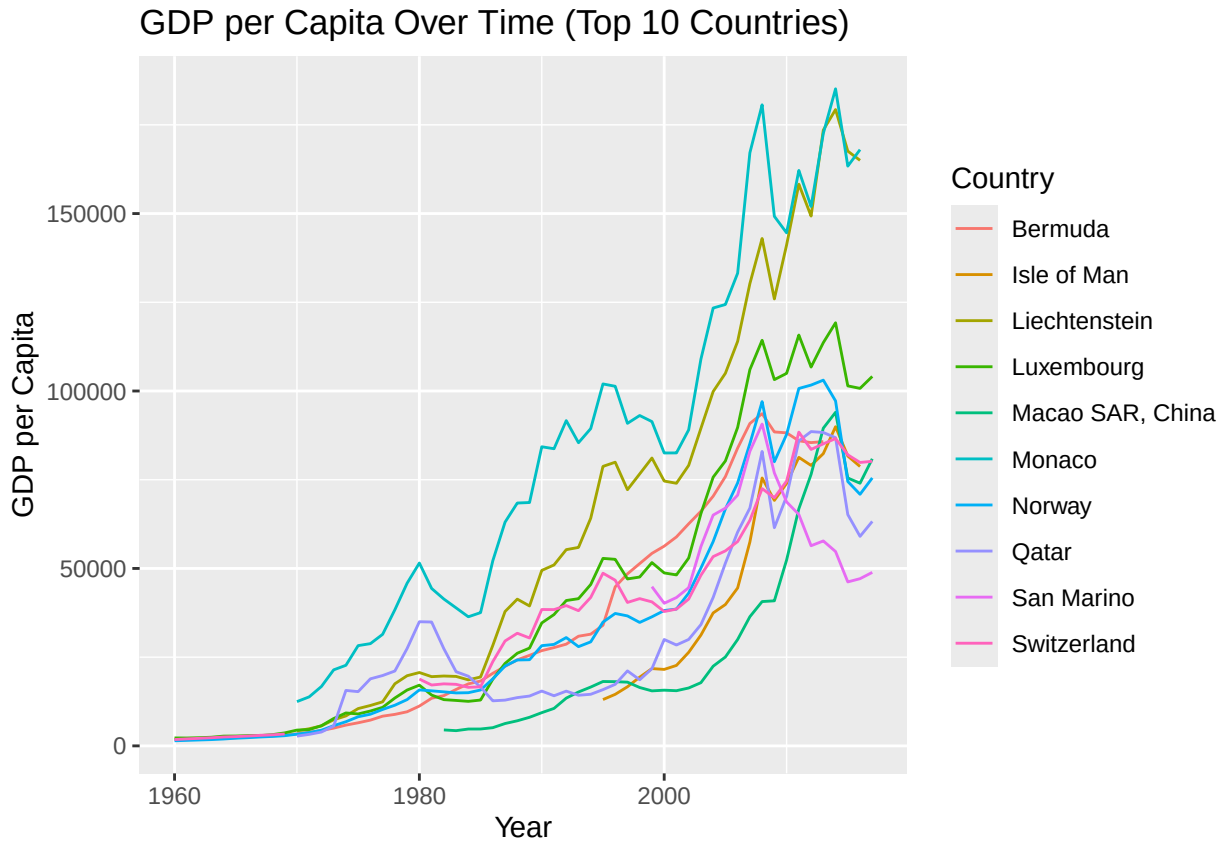
```
# highest gdp per capita
gdp_data %>%
  filter(gdp_per_capita == max(gdp_per_capita, na.rm = TRUE)) %>%
  select(Country, Year, gdp_per_capita) %>%
  arrange(desc(gdp_per_capita))
```

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

```
# Find the top 10 countries
top_countries <- gdp_data %>%
  as_tibble() %>%  # drop tsibble behavior for this summary step
  group_by(Country) %>%
  summarise(max_pc = max(gdp_per_capita, na.rm = TRUE), .groups = "drop") %>%
  filter(is.finite(max_pc)) %>%
  slice_max(max_pc, n = 10) %>%
  pull(Country)

print(top_countries)
```

```
##  [1] Monaco           Liechtenstein    Luxembourg       Norway
##  [5] Macao SAR, China Bermuda          San Marino       Isle of Man
##  [9] Qatar            Switzerland
## 263 Levels: Afghanistan Albania Algeria American Samoa Andorra ... Zimbabwe
```

```
# Plot top 10 countries
gdp_data %>%
  filter(Country %in% top_countries) %>%
  autoplot(gdp_per_capita) +
  labs(title = "GDP per Capita Over Time (Top 10 Countries)",
       x = "Year", y = "GDP per Capita")
```



We can see that **Monaco** is the country with the highest GDP per capita. GDP per capita generally increases over time for most countries, reflecting long-term economic growth. The country with the highest GDP per capita in the dataset is Monaco. Over time, Monaco's GDP per capita has grown steadily and remains significantly higher than most other countries.

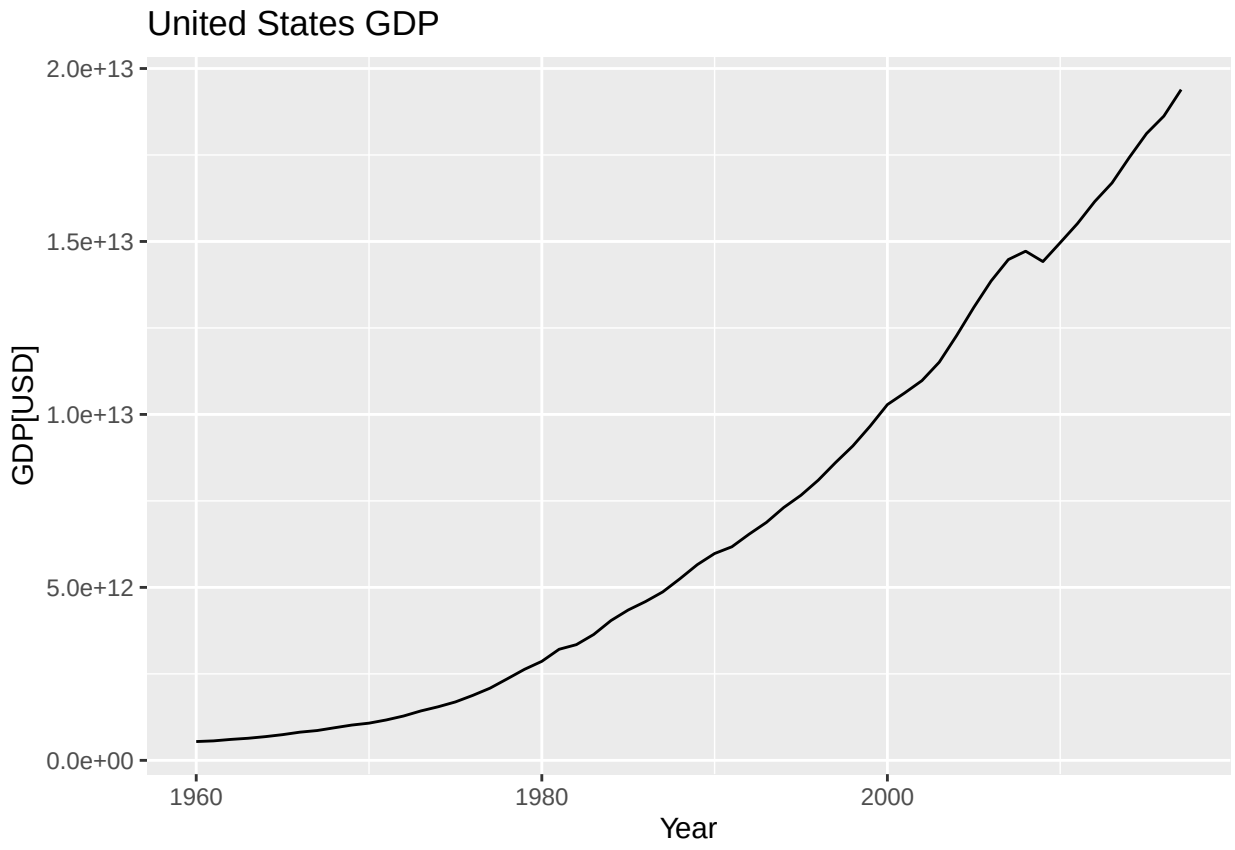
We notice that most countries with the highest GDP are small countries in terms of population, which greatly impacts the per capita GDP (not the GDP).

Question 2

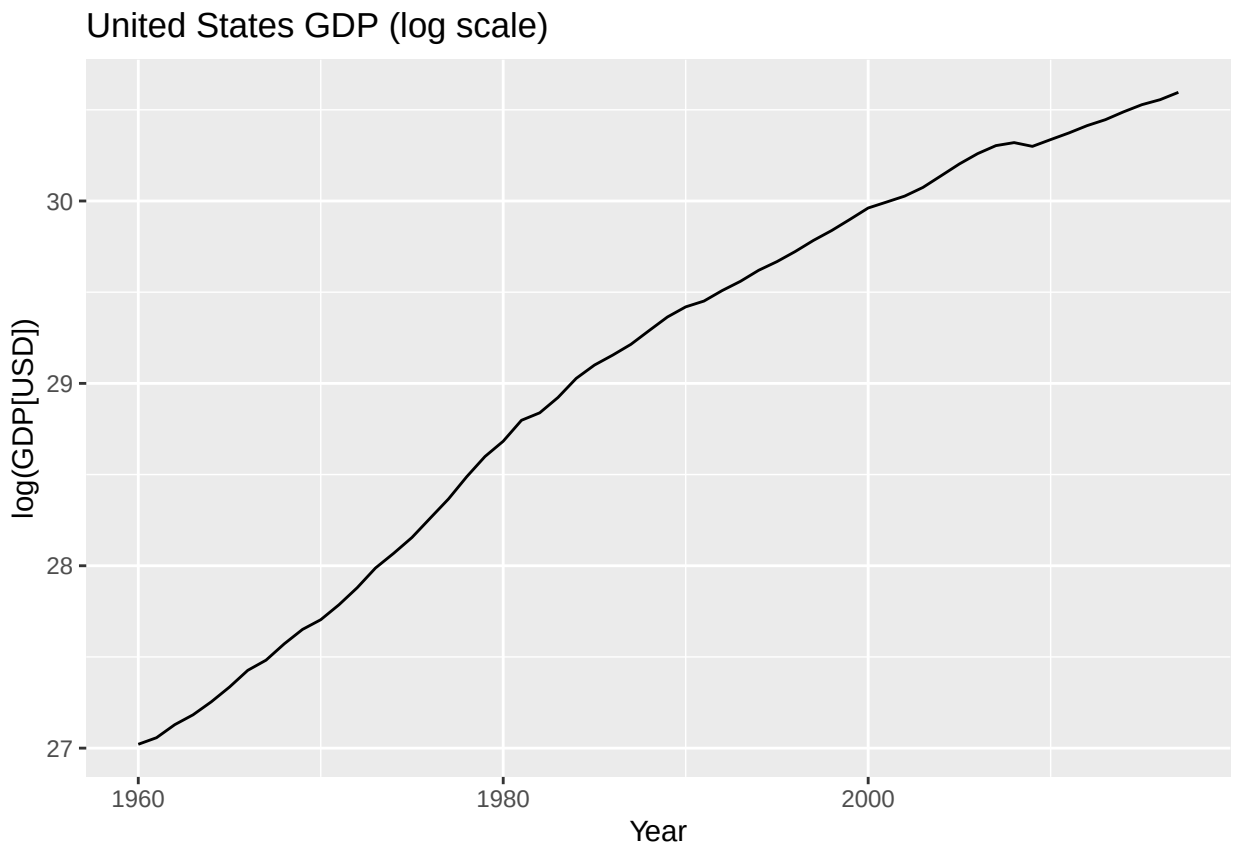
For each of the following series, make a graph of the data. If transforming seems appropriate, do so and describe the effect.

United States GDP from global_economy. Slaughter of Victorian "Bulls, bullocks and steers" in aus_livestock. Victorian Electricity Demand from vic_elec. Gas production from aus_production.

```r
us_gdp <- global_economy |>
  filter(Country == "United States")

# original scale
autoplot(us_gdp, GDP) +
  labs(title = "United States GDP", x = "Year", y = "GDP[USD]")
```

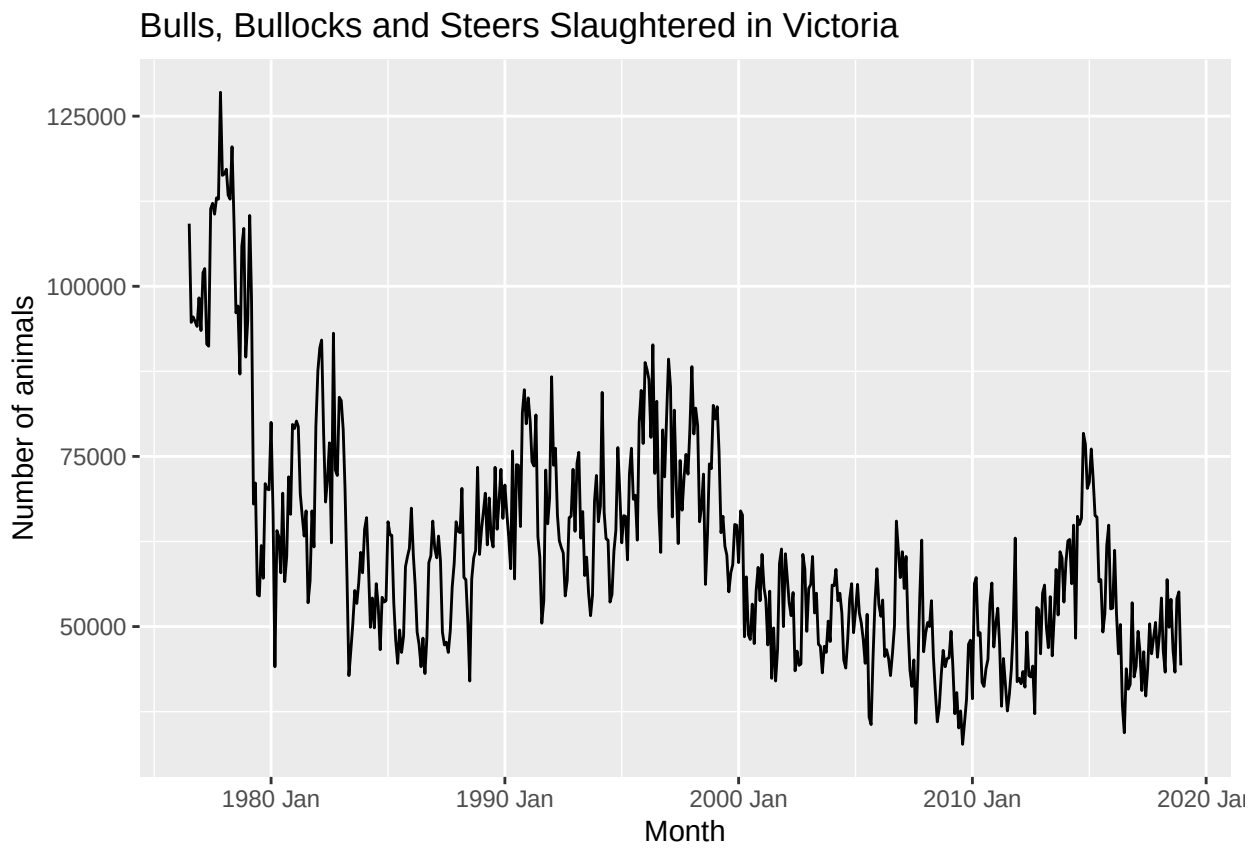



```r
# log transformation since exponential growth in the dataset
autoplot(us_gdp, log(GDP)) +
  labs(title = "United States GDP (log scale)", x = "Year", y = "log(GDP[USD])")
```

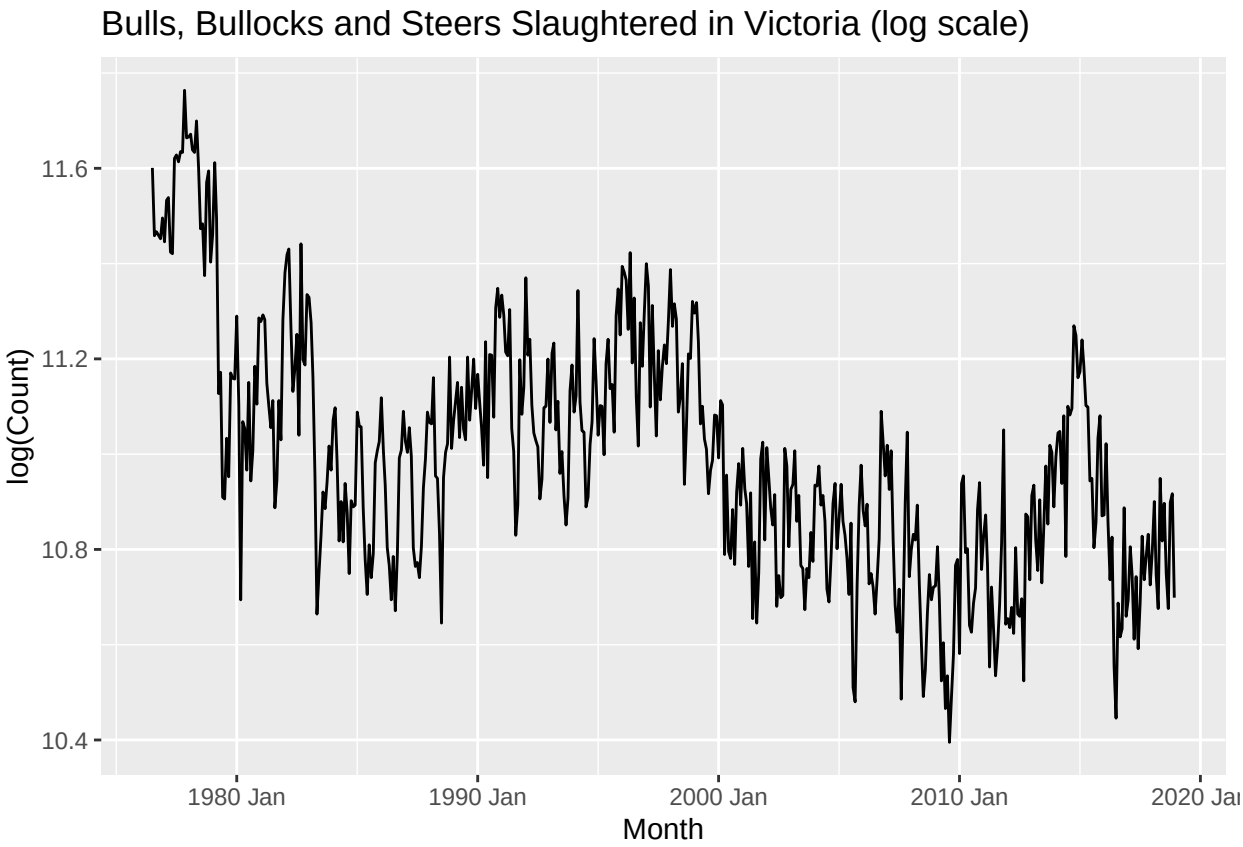


The first graph shows that the data has an exponential growth. Applying a log transformation linearizes the trend, making the growth rate appear more constant over time and stabilizing the increasing variability.

```
animals <- aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers", State == "Victoria")

# original scale
autoplot(animals, Count) +
  labs(title = "Bulls, Bullocks and Steers Slaughtered in Victoria",
       x = "Month", y = "Number of animals")
```

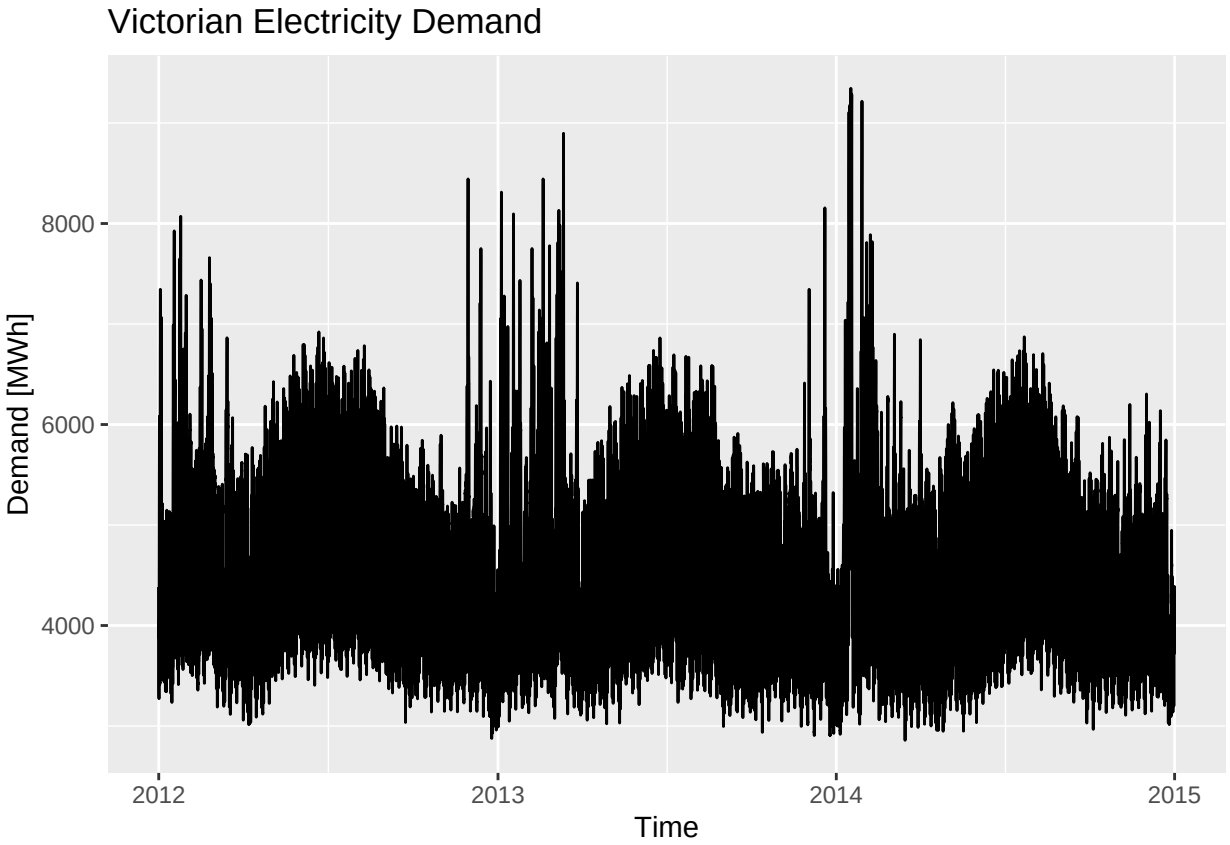



```
# log transform since the data has increasing variability
autoplot(animals, log(Count)) +
  labs(title = "Bulls, Bullocks and Steers Slaughtered in Victoria (log scale)",
       x = "Month", y = "log(Count)")
```

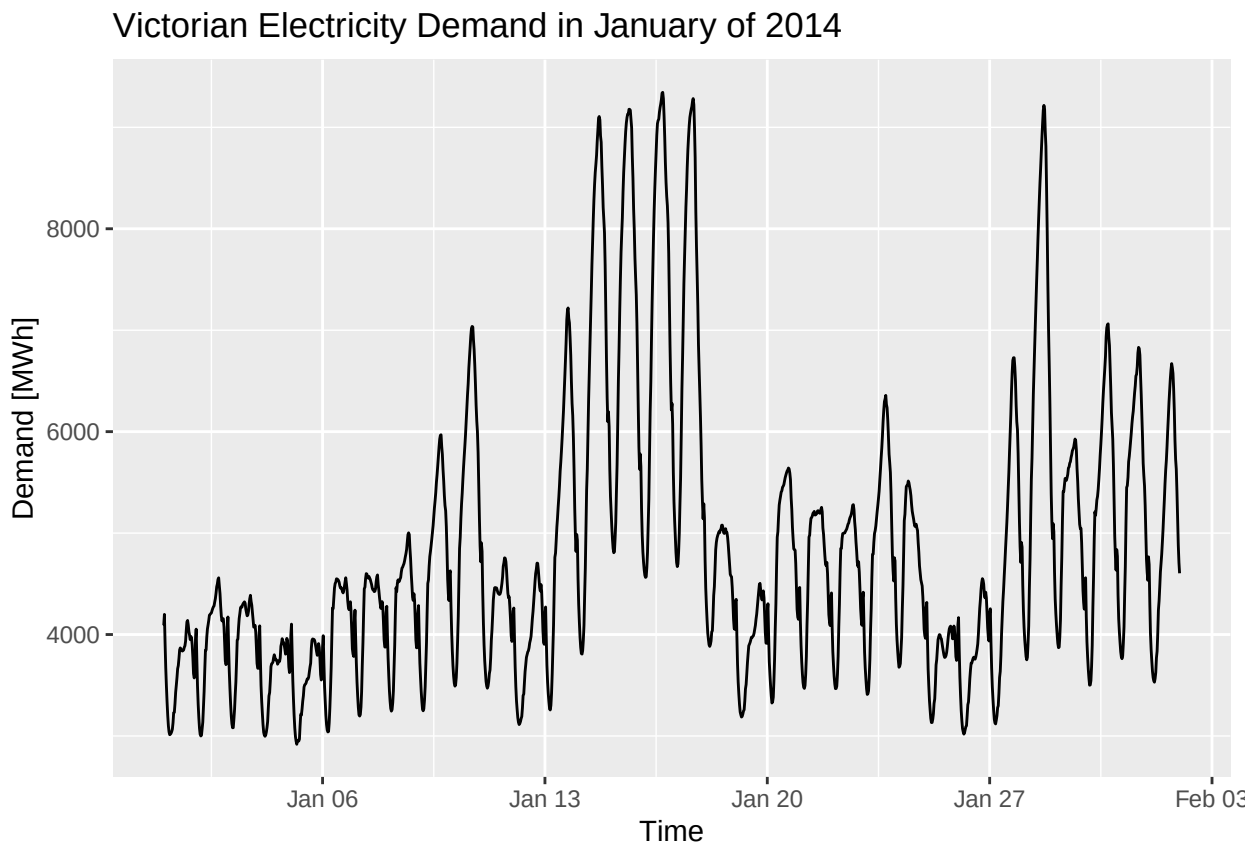


The bulls, bullocks, and steers graph varies over time and can show changing variability. So, I decided to do a log transformation to reduce the impact of long peaks and makes fluctuations easier to compare comparable.

```
# Original scale plot
autoplot(vic_elec, Demand) +
  labs(title = "Victorian Electricity Demand", x = "Time", y = "Demand [MWh]")
```

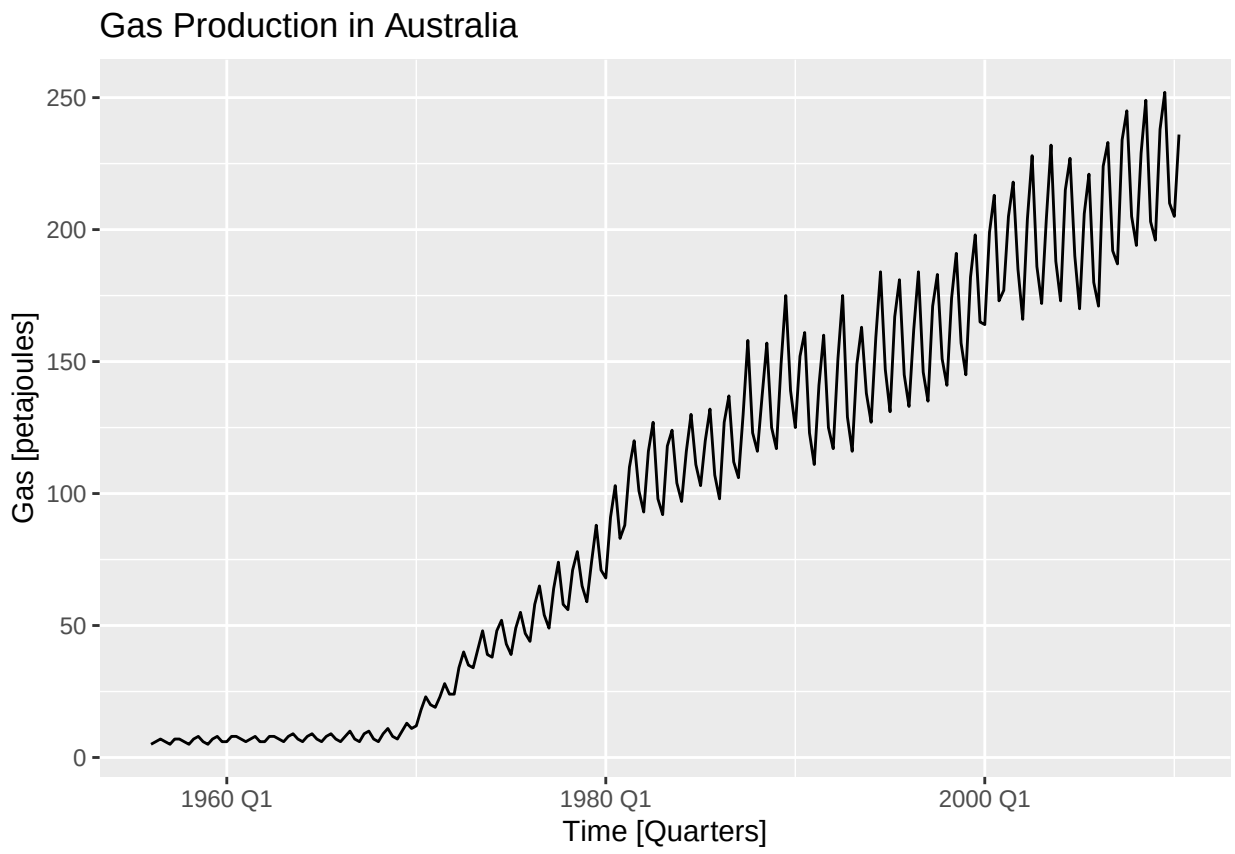



```
# splitting to a shorter window so patterns are visible
vic_elec %>%
  filter_index("2014-01-01" ~ "2014-02-01") %>%
  autoplot(Demand) +
  labs(title = "Victorian Electricity Demand in January of 2014",
       x = "Time", y = "Demand [MWh]")
```

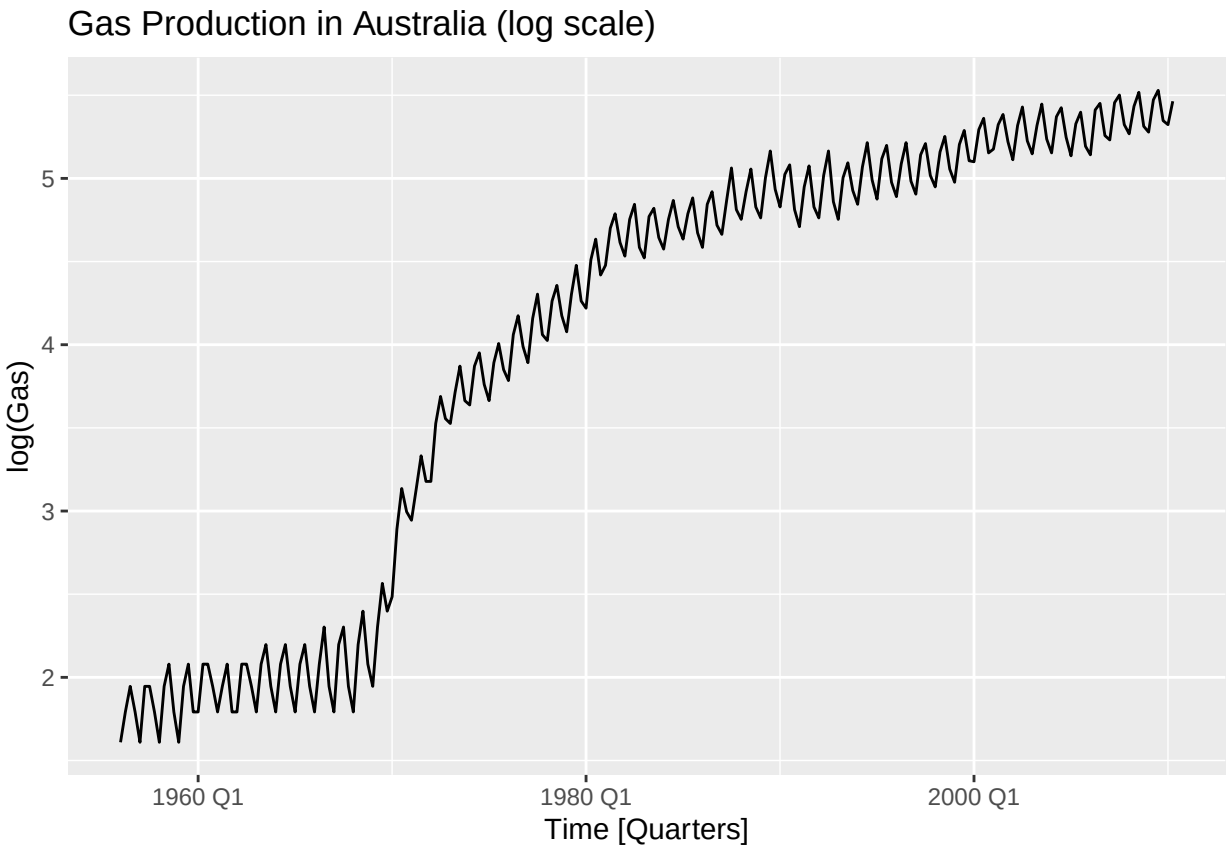


The original graph is extremely dense, with a lot of data points in the same plot, which is not surprising since data points are collected every 30min. I tried to transform it using log but it doesn't really change much. So, I decided to focus on a specific time interval (whole month of January) and analyze the trend in that month. It is much easier to understand daily and weekly patterns.

```
# original scale plot
autoplot(aus_production, Gas) +
  labs(title = "Gas Production in Australia", x = "Time [Quarters]", y = "Gas [petajoules]")
```




```
# log transform since multiplicative growth
autoplot(aus_production, log(Gas)) +
  labs(title = "Gas Production in Australia (log scale)", x = "Time [Quarters]", y = "log(Gas)")
```



The dataset with the original scale has a geometric growth (multiplicative), so I applied the log transformation.

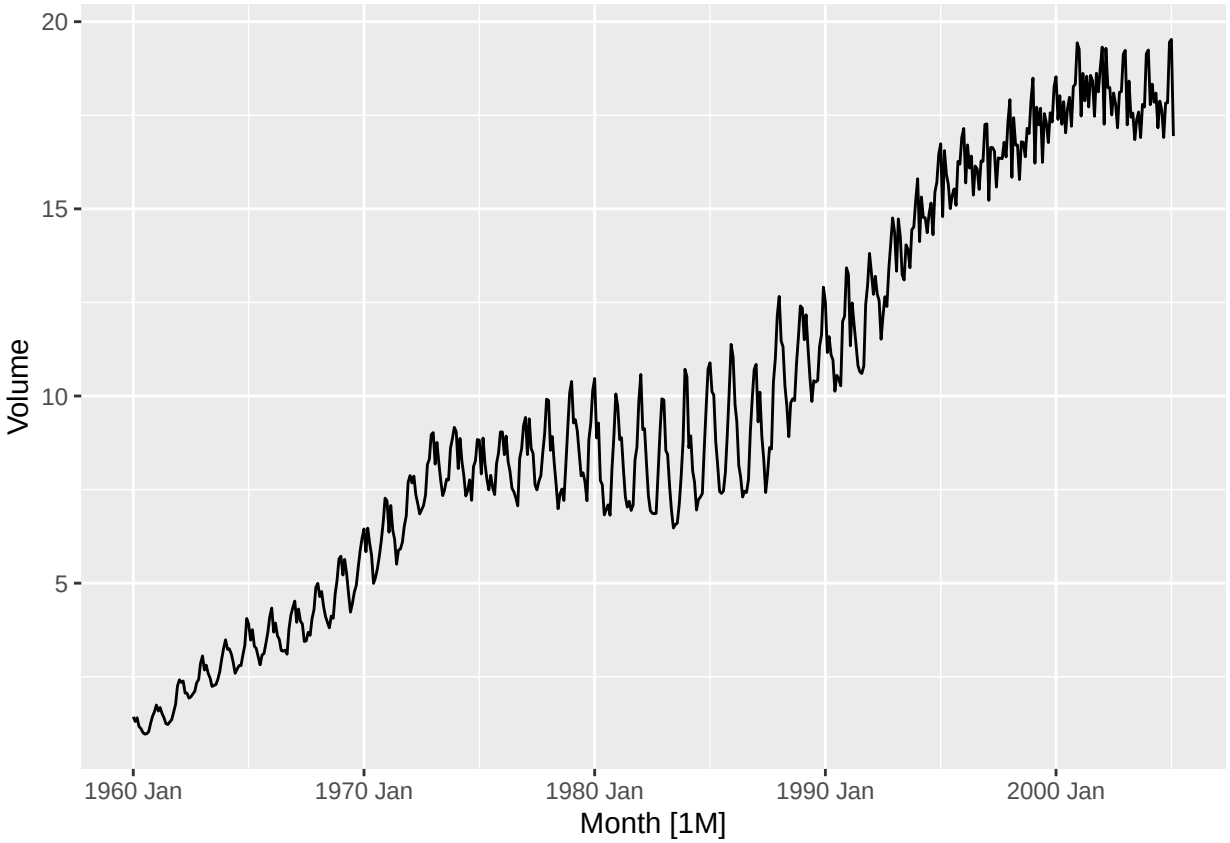
Question 3

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

```
canadian_gas %>%
  as_tibble() %>%
  summarise(min_gas = min(Volume, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   min_gas
##     <dbl>
## 1   0.966
```

```
autoplot(canadian_gas, Volume)
```



A Box–Cox transformation is unhelpful for the Canadian gas series because the variability doesn't increase proportionally with time. The dominant feature of the series is seasonality and not than multiplicative growth, and the variance appears relatively stable over time (not increasing with time). So, using a Box–Cox transformation would not be helpful to better understand the data.

Question 4

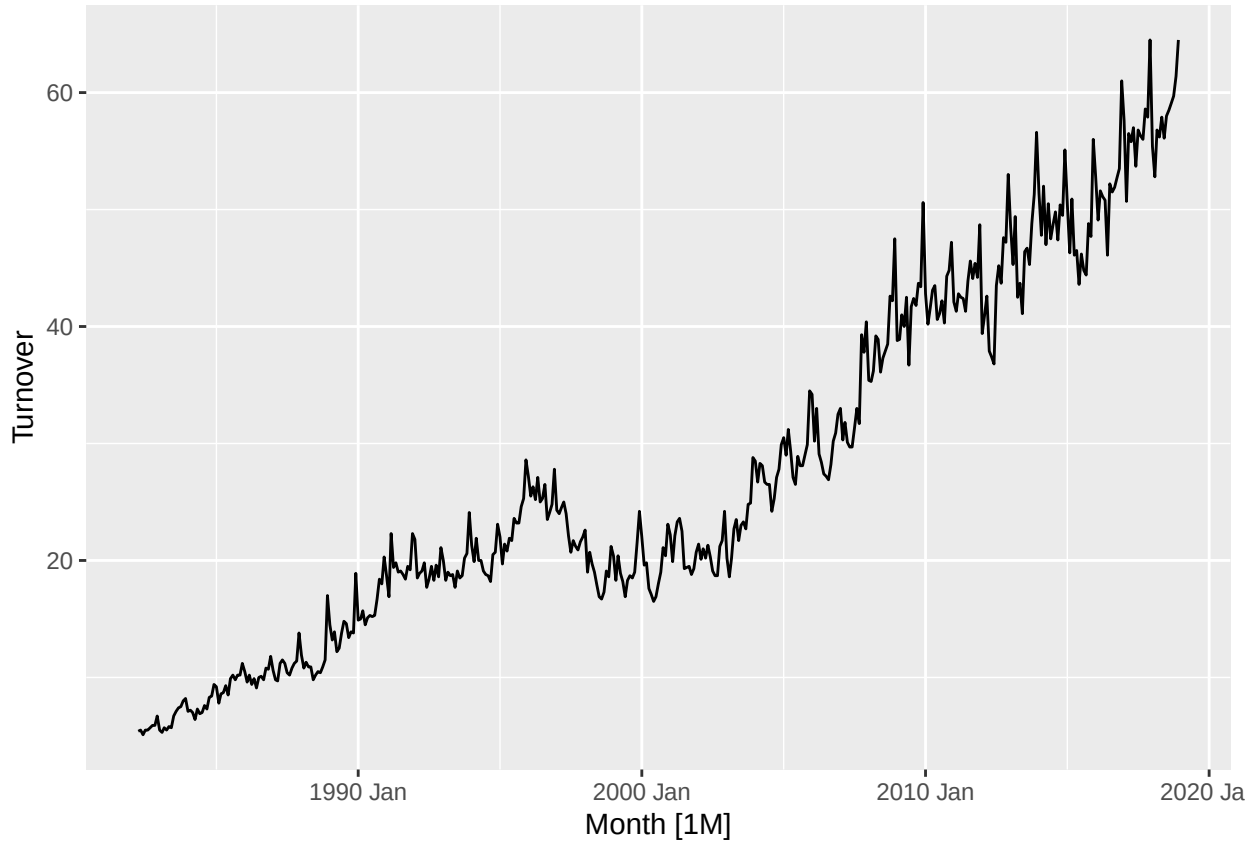
What Box-Cox transformation would you select for your retail data (from Exercise 7 in Section 2.10)?

```
set.seed(1234)

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

myseries
```

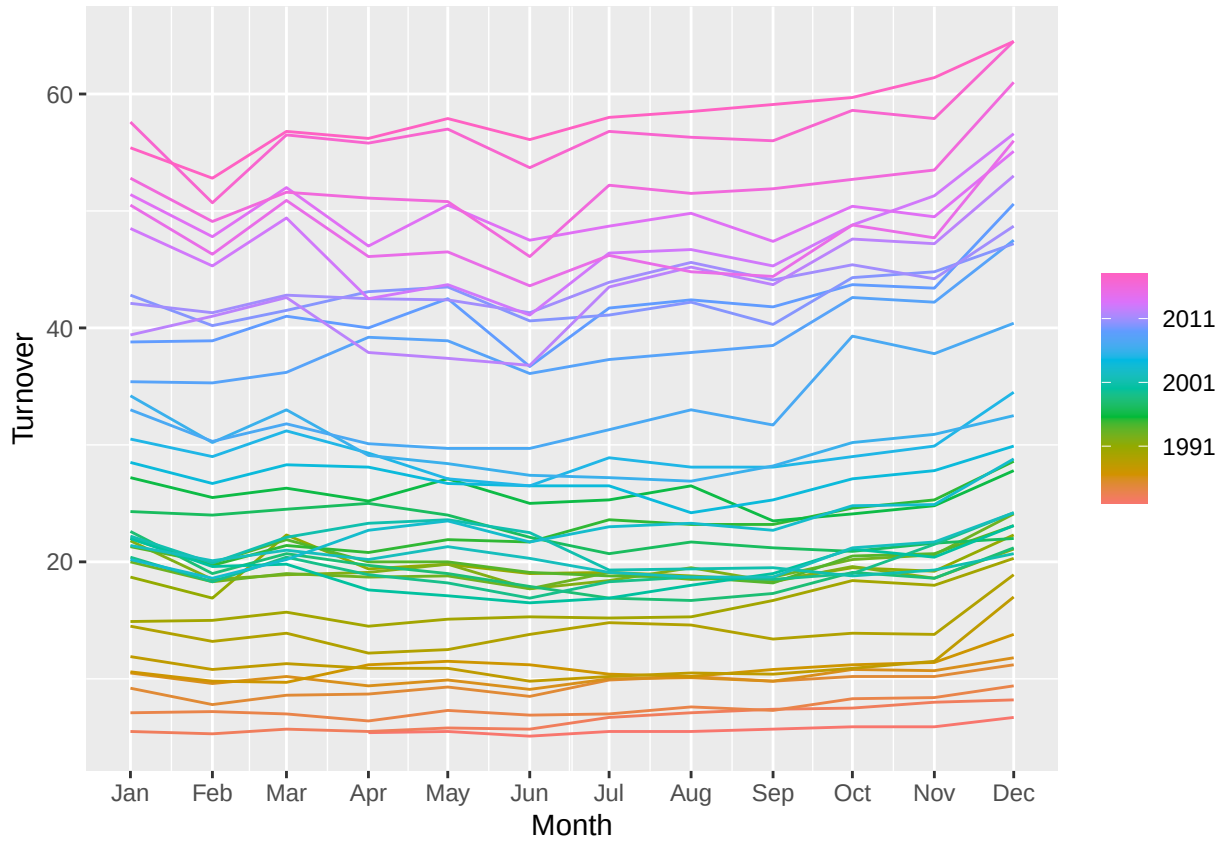
```
## # A tsibble: 441 x 5 [1M]
## # Key:       State, Industry [1]
##    State    Industry                                `Series ID`    Month Turnover
##    <chr>    <chr>                                   <chr>          <mth>    <dbl>
##  1 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Apr      5.4
##  2 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 May      5.5
##  3 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Jun      5.1
##  4 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Jul      5.5
##  5 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Aug      5.5
```

```
##  6 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Sep      5.7
##  7 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Oct      5.9
##  8 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Nov      5.9
##  9 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1982 Dec      6.7
## 10 Tasmania Cafes, restaurants and takeaway food ~ A3349520V   1983 Jan      5.5
## # i 431 more rows
```

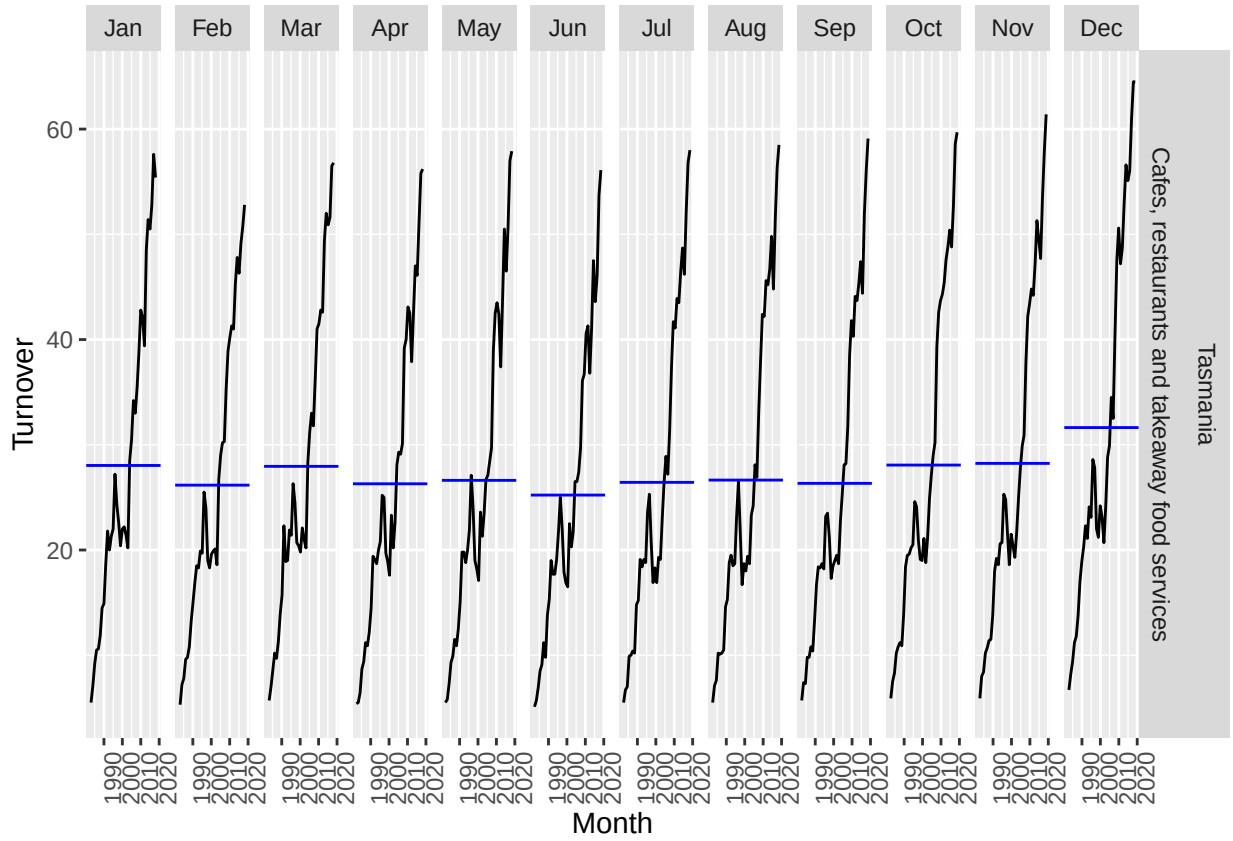
```r
# exploring the series
autoplot(myseries, Turnover)
```



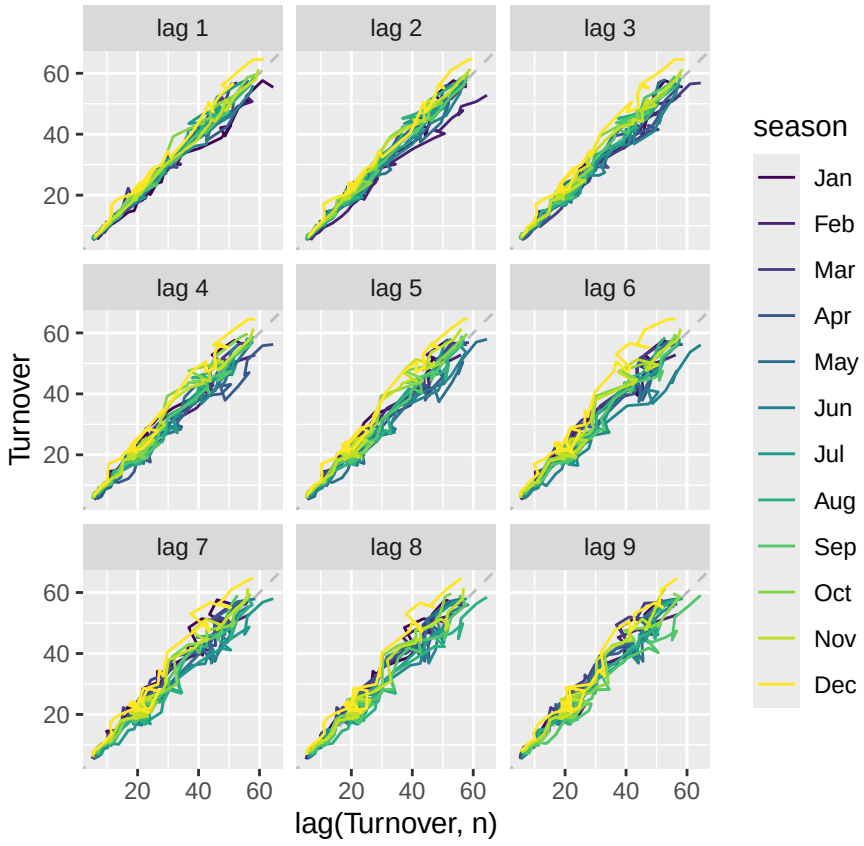

```r
gg_season(myseries, Turnover)
```



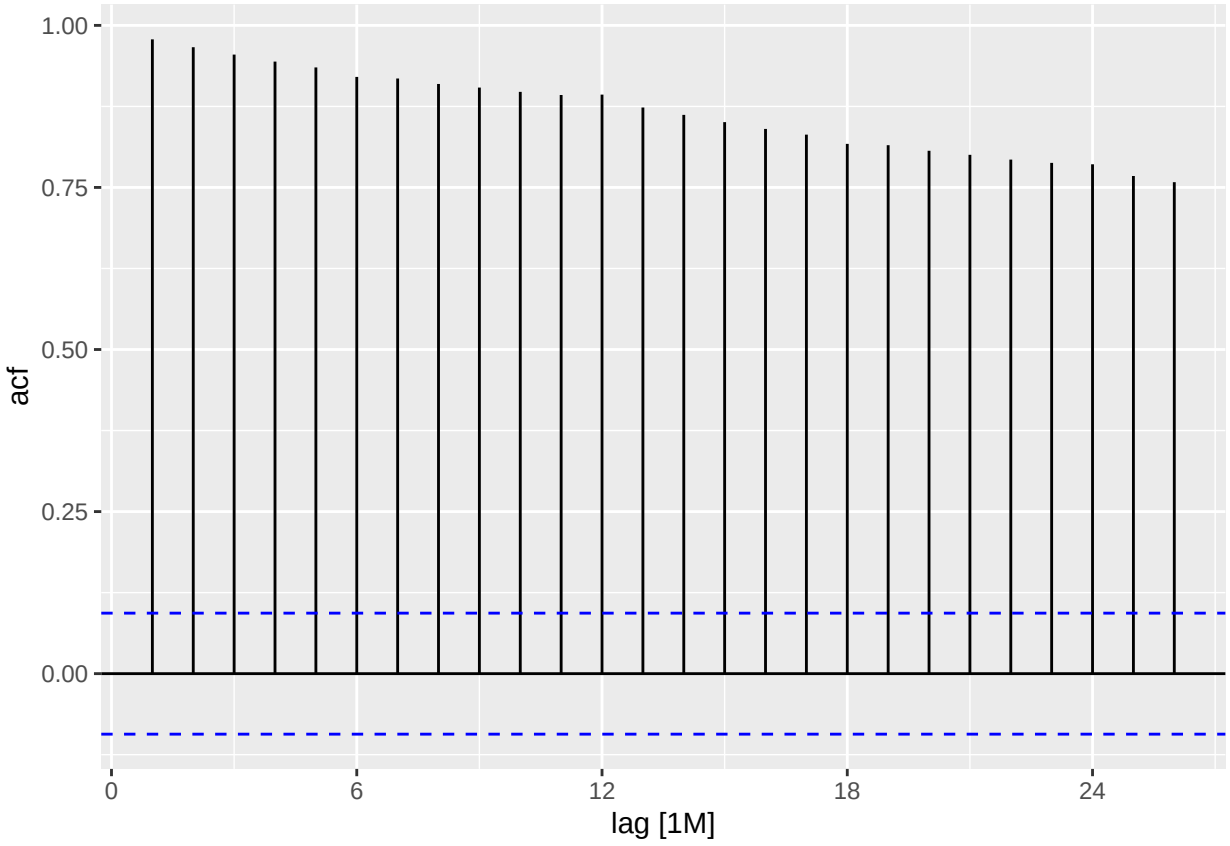

```
gg_subseries(myseries, Turnover)
```




```
gg_lag(myseries, Turnover)
```




```
myseries |> ACF(Turnover) |> autoplot()
```



Optimal Box-Cox lambda

```
# lambda
myseries |>
  features(Turnover, guerrero)
```

```
## # A tibble: 1 x 3
##   State    Industry                                    lambda_guerrero
##   <chr>    <chr>                                                 <dbl>
## 1 Tasmania Cafes, restaurants and takeaway food services          0.320
```

The estimated Box–Cox parameter using the Guerrero method (since it's a time series) is lambda = 0.32. Since this value is substantially below 1 and closer to 0, it indicates that a transformation is appropriate to stabilize the variance. A log transformation (lambda 0) would work and would help convert the multiplicative seasonal pattern into a more additive one. Since we have the optimal lambda value, we could select a Box–Cox transformation with lambda around 0.3, or use a more straightforward approach with log transformation.

Question 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.

```r
# Tobacco series
lambda_tobacco <- aus_production %>%
  features(Tobacco, guerrero)

lambda_tobacco
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1           0.926
```

```r
# Economy class passengers series
lambda_ansett <- ansett %>%
  filter(Class == "Economy", `Airports` == "MEL-SYD") %>%
  features(Passengers, guerrero)

lambda_ansett
```

```
## # A tibble: 1 x 3
##   Airports Class   lambda_guerrero
##   <chr>    <chr>             <dbl>
## 1 MEL-SYD  Economy            2.00
```

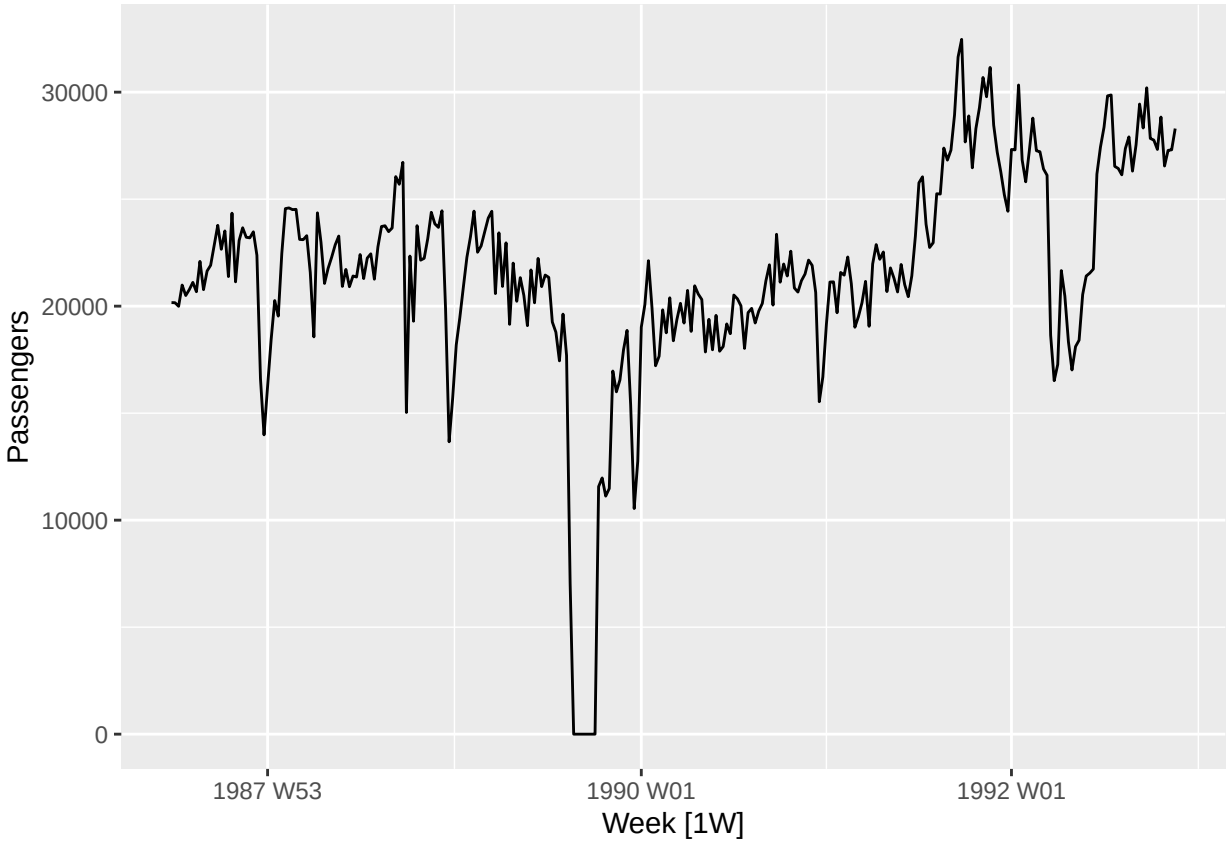
```r
# Pedestrian counts
lambda_ped <- pedestrian %>%
  filter(Sensor == "Southern Cross Station") %>%
  features(Count, guerrero)

lambda_ped
```

```
## # A tibble: 1 x 2
##   Sensor                 lambda_guerrero
##   <chr>                            <dbl>
## 1 Southern Cross Station          -0.250
```

The ideal lambdas are: **Tobacco series**: 0.93 **Economy class passengers**: 2.00 **Pedestrian counts**: -0.25

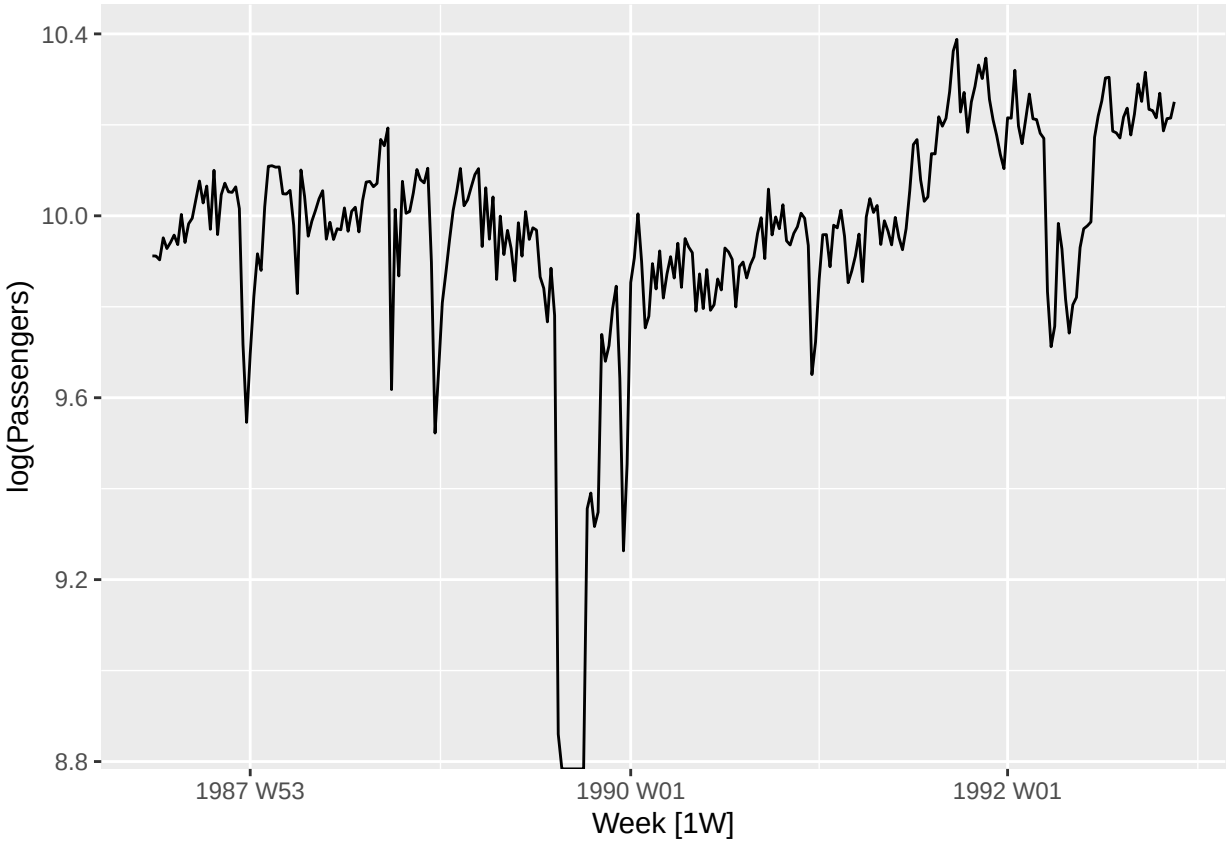
For Tobacco, the Guerrero lambda estimate is 0.93, which is pretty close to 1, suggesting the variance is somewhat stable, so we only need a mild transformation.

```r
# plot to better understand why lambda is 2
ansett_economy <- ansett |>
  filter(Class == "Economy", `Airports` == "MEL-SYD")

autoplot(ansett_economy, Passengers)
```



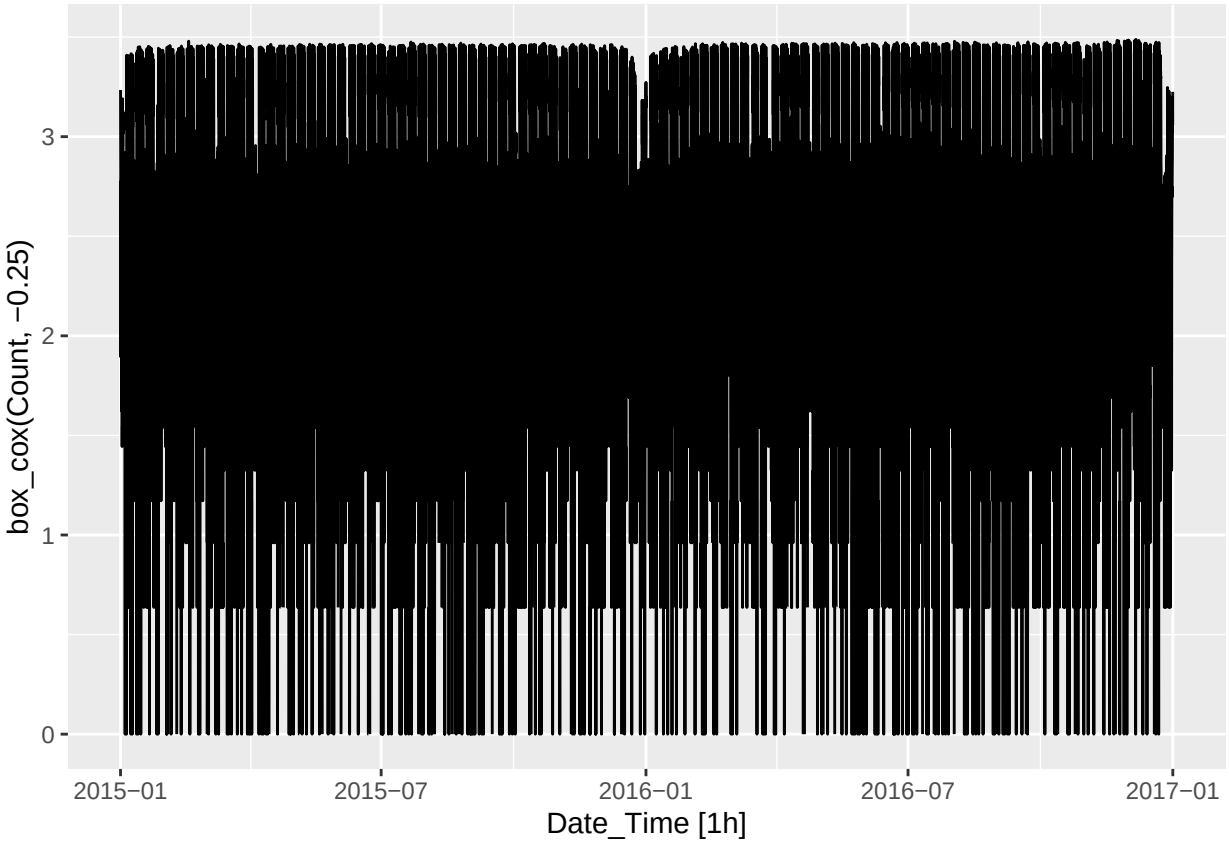
The Guerrero method suggests lambda is 2.00, however, the series (plot above) has sharp structural breaks and basically zero values around 1990, which distort the automatic estimation. The data seems to have increasing variance with level and multiplicative seasonal behavior typical of passenger counts. So, a log transformation is more appropriate in practice, as it stabilizes variance and will give us an easier to understand and interpret series.

```
autoplot(ansett_economy, log(Passengers))
```



For pedestrian counts, we got a lambda equal to -0.25, which indicates the need for a transformation that stabilizes variance. A negative lambda corresponds to an inverse power transform, which heavily compresses large spikes and reduces the difference in variance, making the series more suitable for modeling.

```
pedestrian %>%
  filter(Sensor == "Southern Cross Station") %>%
  autoplot(box_cox(Count, -0.25))
```



Question 7

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

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

```
## # A tsibble: 20 x 2 [1Q]
##      Gas Quarter
##    <dbl>   <qtr>
##  1   221 2005 Q3
##  2   180 2005 Q4
##  3   171 2006 Q1
##  4   224 2006 Q2
##  5   233 2006 Q3
##  6   192 2006 Q4
##  7   187 2007 Q1
##  8   234 2007 Q2
##  9   245 2007 Q3
## 10   205 2007 Q4
## 11   194 2008 Q1
## 12   229 2008 Q2
## 13   249 2008 Q3
## 14   203 2008 Q4
```

```
## 15   196 2009 Q1
## 16   238 2009 Q2
## 17   252 2009 Q3
## 18   210 2009 Q4
## 19   205 2010 Q1
## 20   236 2010 Q2
```

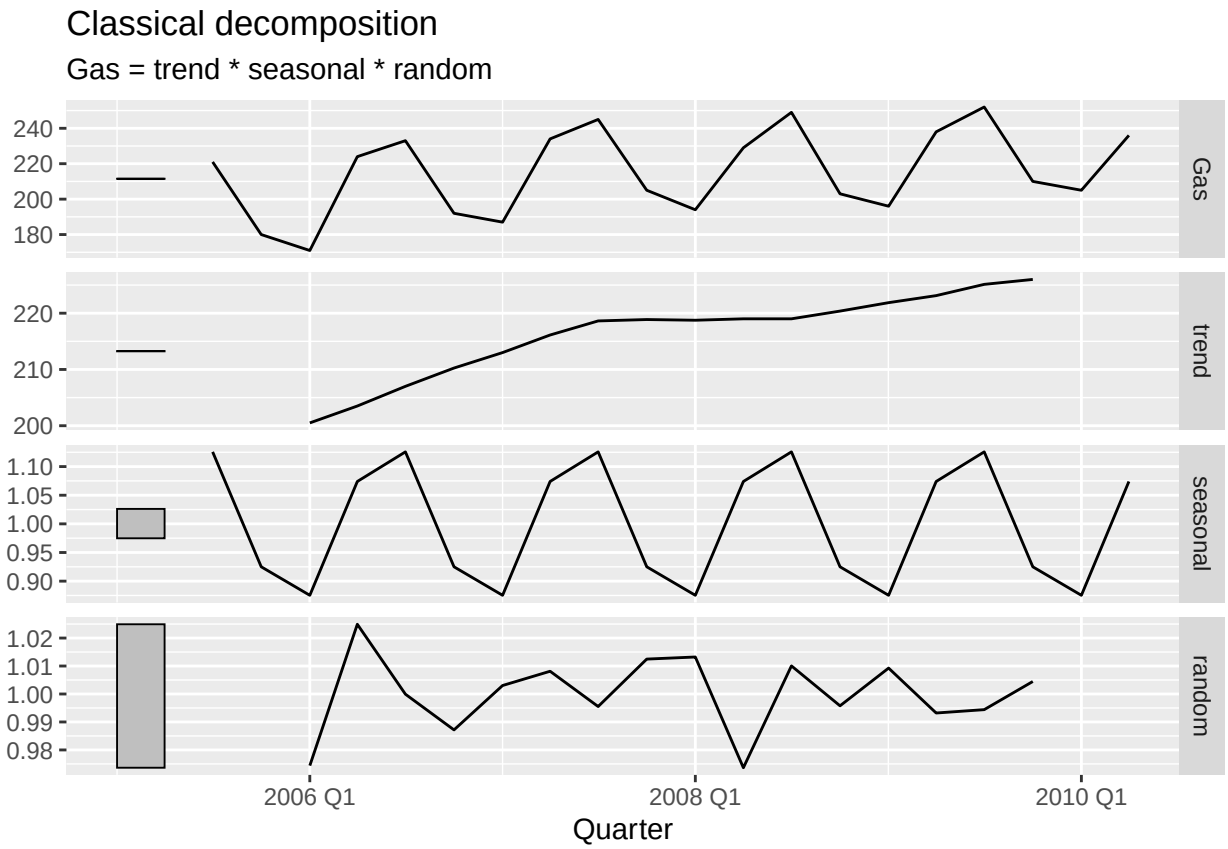
a. Plot the time series. Can you identify seasonal fluctuations and/or a trend-cycle?

```
# plot
autoplot(gas, Gas) +
  labs(title = "Gas Production in the last 5 years", x = "time", y = "Gas [petajoules]")
```

Answer (7a): The series shows clear quarterly seasonality with repeating annual peaks and declines, along with a mild upward trend over time. Seasonal fluctuations increase slightly as the level rises, indicating multiplicative seasonality.

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

```
gas_decomp <- gas |>
  model(classical_decomposition(Gas, type = "multiplicative"))

components(gas_decomp) |> autoplot()
```

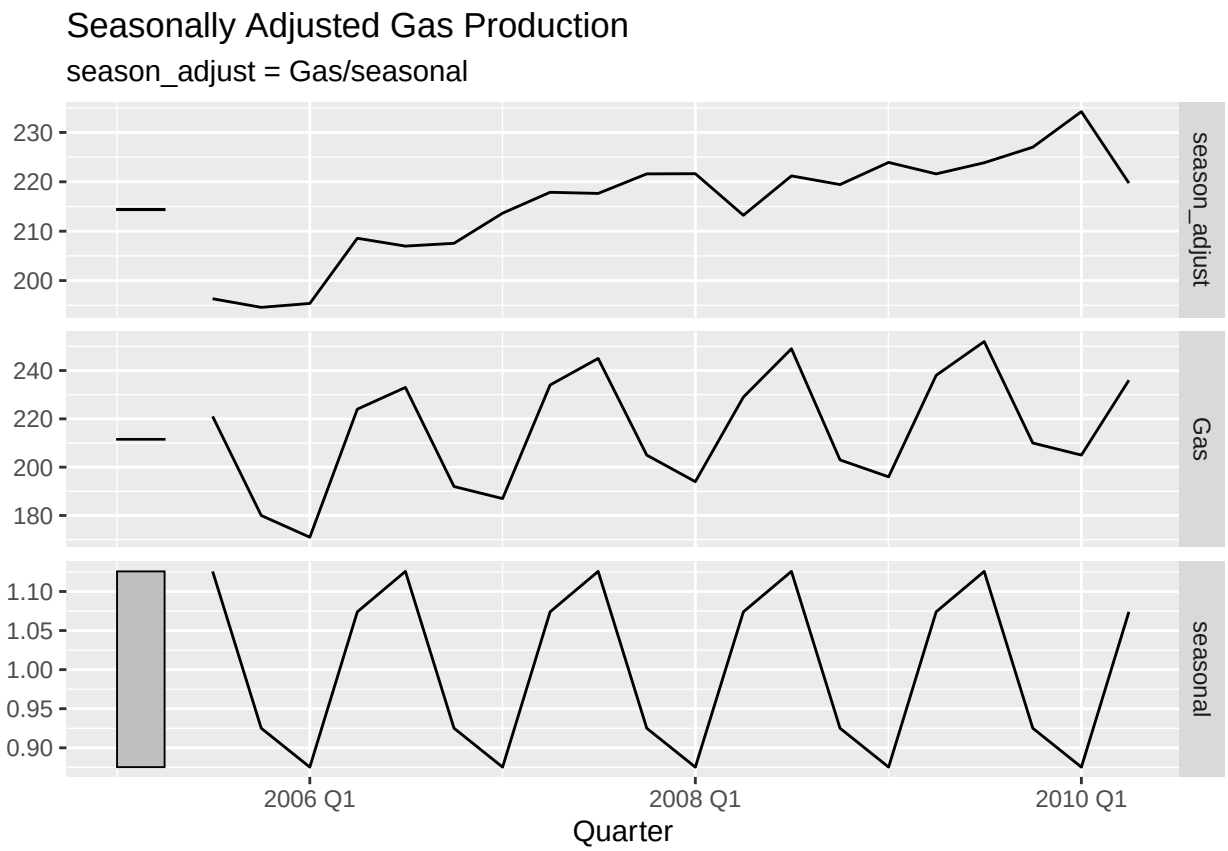


c. Do the results support the graphical interpretation from part a?

Answer (7b and c): The classical multiplicative decomposition shows a smooth upward trend-cycle during this 5 year interval, showing a gradual increase in gas production as seen in part (a). The seasonal indices display a stable and repeating pattern (quarterly), with consistent peaks and declines each year, indicating strong seasonality. Most of the variation in the series iswell explained by the trend-cycle and seasonal components. So, the decomposition supports the graphical interpretation from part (a), confirming both an upward trend and quarterly seasonal fluctuations.

d. Compute and plot the seasonally adjusted data.

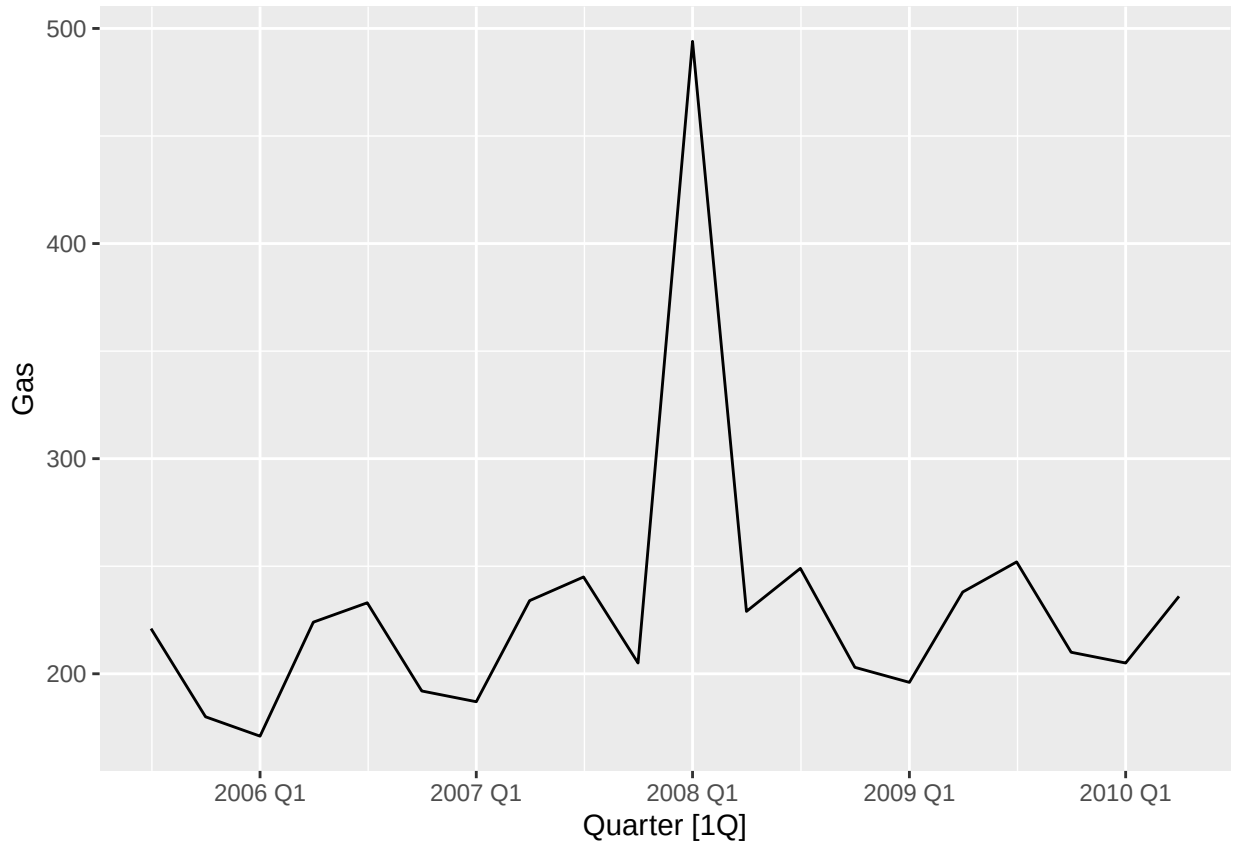
```
gas_components <- components(gas_decomp)

gas_components |>
  autoplot(season_adjust) +
  labs(title = "Seasonally Adjusted Gas Production")
```



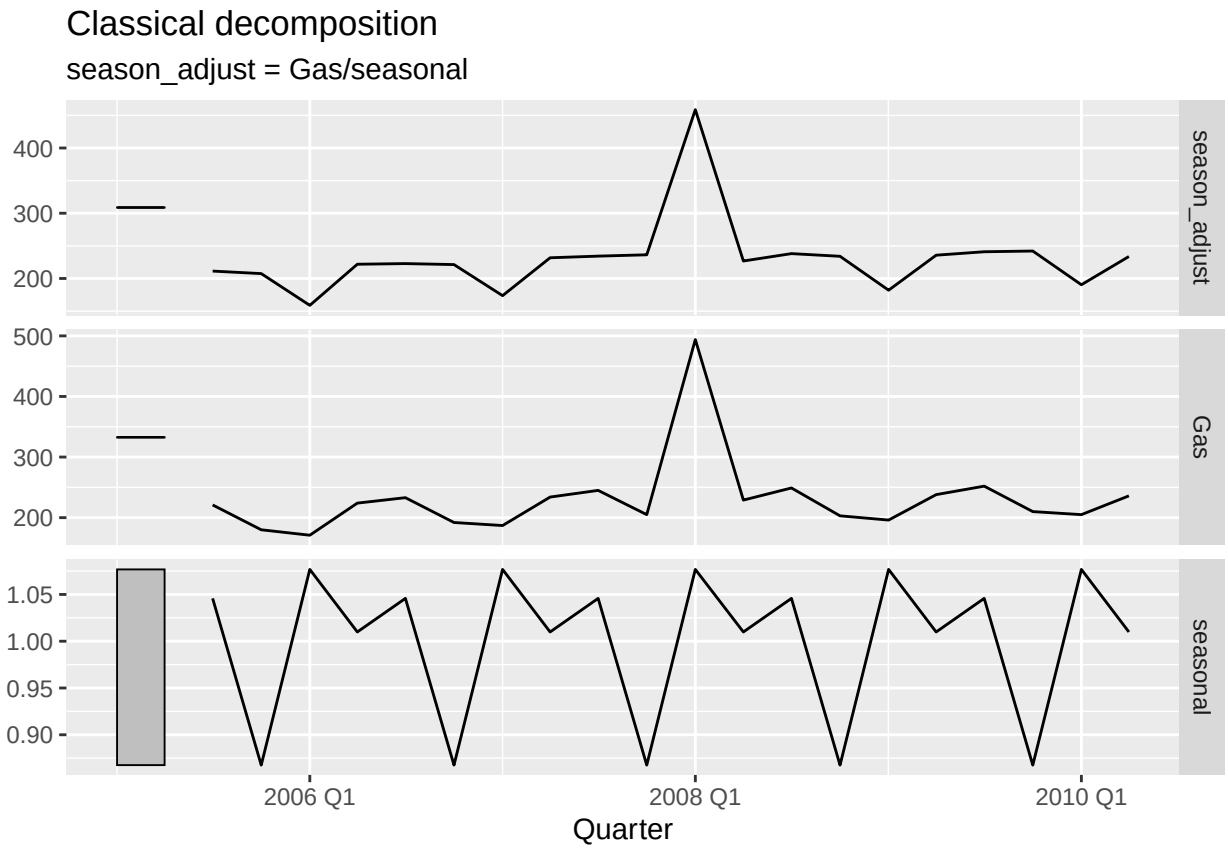
Answer (7d): The seasonally adjusted series removes the quarterly pattern and shows a smoother upward trend. This confirms that seasonality accounts for most short-term variation, leaving a much more clear trend-cycle.

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_outlier <- gas
gas_outlier$Gas[11] <- gas_outlier$Gas[11] + 300

autoplot(gas_outlier, Gas)
```




```
gas_out_decomp <- gas_outlier |>
  model(classical_decomposition(Gas, type = "multiplicative"))

components(gas_out_decomp) |>
  autoplot(season_adjust)
```



Answer (7e): After adding an outlier to one observation, the gas series shows a large spike, which is reflected in both the trend-cycle and the seasonally adjusted components. The seasonally adjusted series now has a pronounced peak at the outlier location, and nearby observations are also kind of distorted. This happens because classical decomposition relies on moving averages to estimate the trend-cycle, so the outlier will greatly impact those moving averages. The seasonal indices shift a little and the trend-cycle is pulled upward around the outlier. So, classical decomposition is not robust to outliers: one extreme value can contaminate the trend, seasonal, and seasonally adjusted components.

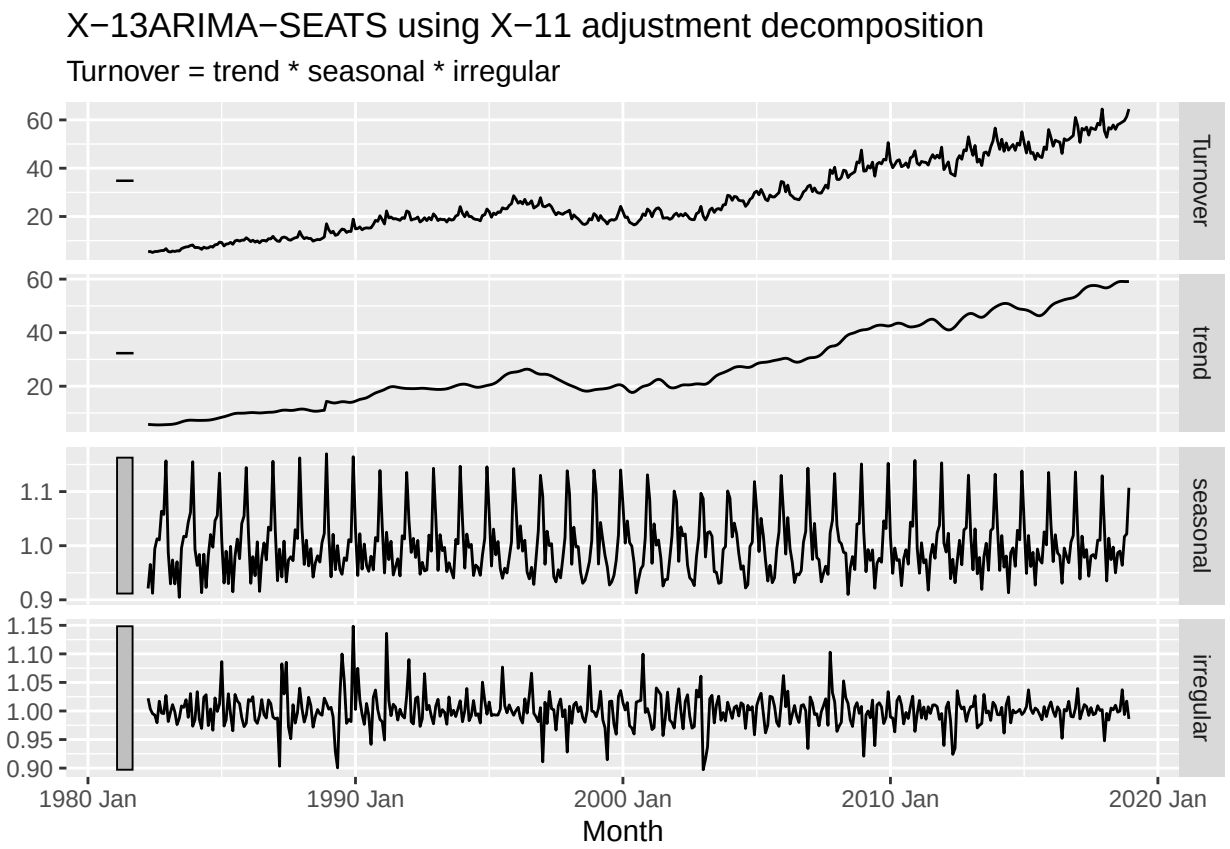
f. Does it make any difference if the outlier is near the end rather than in the middle of the time series?

Answer (7f): An outlier in the middle of the series impacts several surrounding (smaller or bigger) observations because moving averages are centered. When the outlier is near the end, there is less distortion since fewer neighboring values are involved/impacted in the moving average calculation.

Question 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?

```
fit_x11 <- myseries %>%
  model(X11 = X_13ARIMA_SEATS(Turnover ~ x11()))

components(fit_x11) %>% autoplot()
```



Answer 8: The X-11 decomposition confirms the strong seasonal pattern and upward trend seen before. The irregular component highlights several sharp spikes, indicating possible outliers that were not as obvious in the original series. Separating the trend and seasonal components makes these abnormal situations much easier to identify.

Question 9

Answer (9a) and (9b): The STL decomposition shows a strong upward trend in the Australian civilian working force, increasing from about 6.5 million to nearly 9 million over the period. The seasonal component is relatively small in magnitude (approximately ± 100 thousand) compared to the overall series level, indicating not big but stable monthly seasonality. The seasonal pattern remains consistent over time, as shown in the seasonal subseries plot. The remainder component exhibits increased volatility around 1991–1992, corresponding to the recession period, which appears mainly as a temporary flattening in the trend and negative deviations in the remainder rather than a change in seasonality. The recession is not noticeable in the raw value or trend panels, because the trend's scale (6,500–9,000) makes the recession impact look much smaller. It is only by examining the remainder on its own scale that the recession (and impact on the job market) shock becomes apparent.