FPP Chapter 3 HW: Time Series Decomposition

Alex Ptacek

Load Packages

```
library(fpp3)
library(tidyverse)
library(seasonal)
library(countrycode)
library(ggpubr)
```

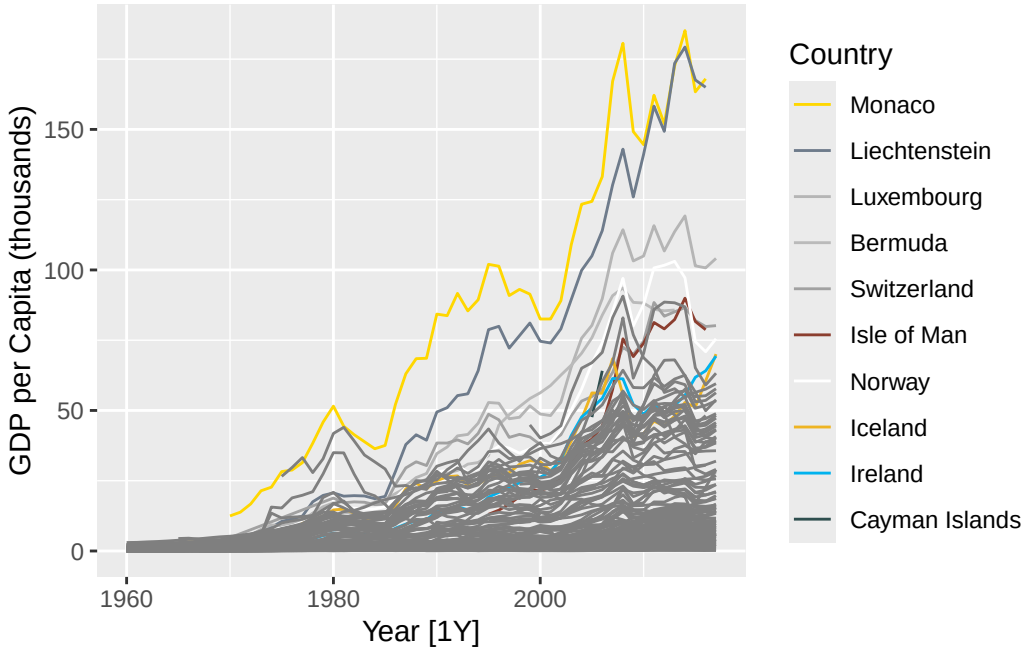
Question 3.1

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: After the below transformations, we can see that Monaco has the highest GDP per capita, followed closely by Liechtenstein. Monaco's data starts in 1970 (10 years later than the beginning of the time series) and has trended upwards over time, more rapidly relative to most of the other time series. There are also heavy cyclic troughs about every 5-10 years.

```
# This code plots GDP per capita for each country over time but is impossible to
# read because there are 263 countries in this dataset.
# There is also an issue of groupings such the EU or the High-Income in the
# country variable
# global_economy |>
#   mutate(gdp_per_capita = GDP/Population) |>
#   autoplot(gdp_per_capita)
```

```r
# Let's trim this down by filtering the global_economy data to countries in the
# countryname_dict data from the countrycode package.
# A quick comparison shows there are some valid countries in global_economy that
# are not in countryname_dict due to naming issues such as Czechia vs Czech
# Republic. We will save this cleaning process for another project. We also filter
# out all the null values from our variables of interest to make factoring and
# overall analysis simpler for the next steps.
trimmed_glob_econ <- global_economy |>
  filter(!is.na(Country) & !is.na(Year) & !is.na(GDP) & !is.na(Population)) |>
  filter(Country %in% countryname_dict$country.name.en) |>
  mutate(gdp_per_capita = GDP/Population)

# Let's further clean the plot by trimming the legend to only top 10 GDP per
# capita, because we're only interested in the highest performers here.

top_countries <- trimmed_glob_econ |>
  group_by(Country) |>
  filter(Year == max(Year)) |>   # Using the latest year as the scale for factoring
  ungroup() |>
  arrange(desc(gdp_per_capita)) |>
  slice_head(n = 10) |>
  pull(Country)


#Improve color contrast for easier visibility of top countries
set.seed(624)
random_colors <- sample(colors(), length(top_countries))


trimmed_glob_econ |>
  mutate(Country = fct_reorder2(Country, Year, gdp_per_capita),
         gdp_per_capita = gdp_per_capita/1e3) |>
  autoplot(gdp_per_capita) +
  scale_color_manual(values = setNames(random_colors, top_countries)) +
  ylab("GDP per Capita (thousands)")
```



Question 3.2

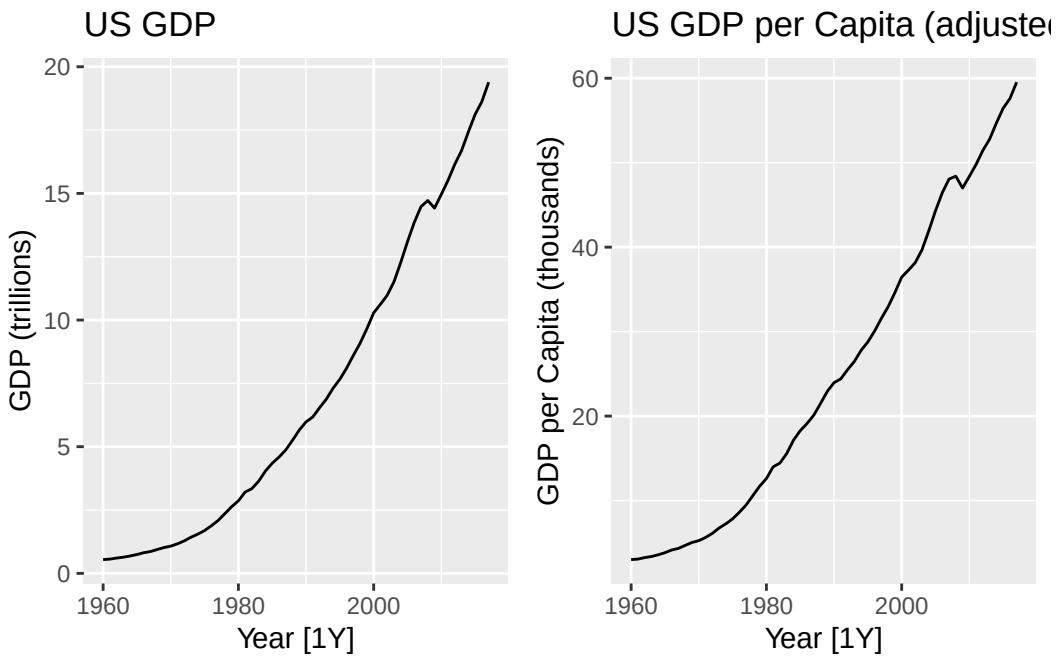
For each of the following series, make a graph of the data. If transforming seems appropriate, do so and describe the effect.

a. United States GDP from global_economy.

Answer: Since GDP is directly affected by population size, it makes sense to transform GDP to GDP per Capita. Interestingly, the plots look nearly identical, showing that population size growth has actually not had a noticeable effect on the patterns GDP growth.

```
ge_plot <- global_economy |>
  filter(Country == "United States") |>
  mutate(GDP = GDP/1e12) |>
  autoplot(GDP) +
  ylab("GDP (trillions)") +
  ggtitle("US GDP")

adj_ge_plot <- global_economy |>
  mutate(gdp_pc = GDP/Population,
```

```
         gdp_pc = gdp_pc/1e3) |>
  filter(Country == "United States") |>
  autoplot(gdp_pc) +
  ylab("GDP per Capita (thousands)") +
  ggtitle("US GDP per Capita (adjusted)")

ggarrange(ge_plot, adj_ge_plot)
```

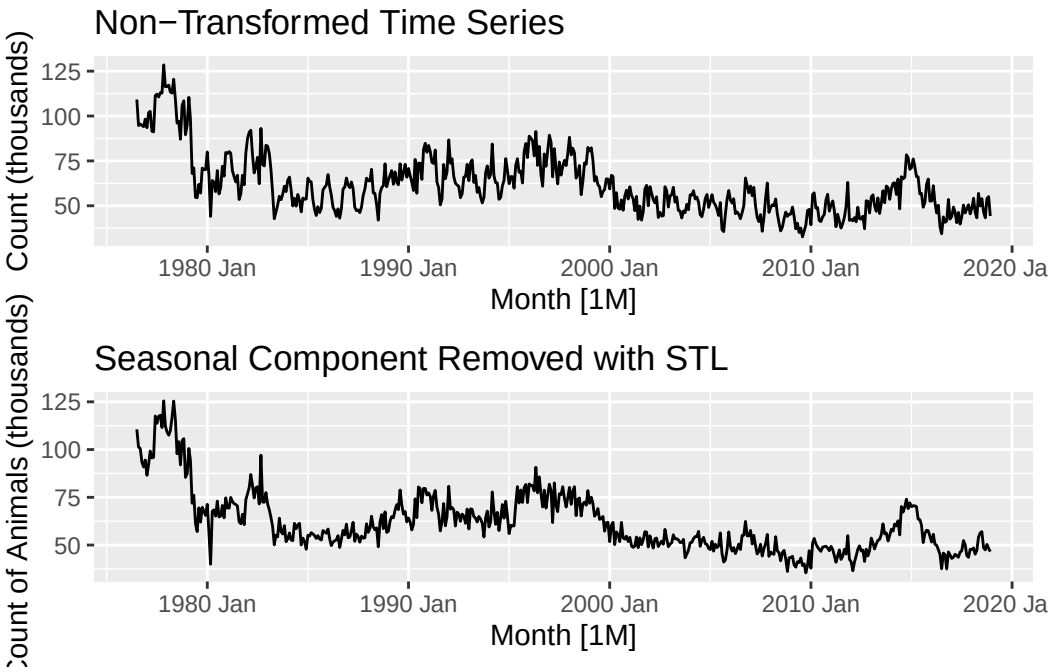


b. Slaughter of Victorian "Bulls, bullocks and steers" in aus_livestock.

Answer: The amount of work days when animals can be slaughtered varies by month due to the months having different number of days. For this reason, it makes sense to try a calendar adjustment. Below, I performed an additive decomposition with the STL model and plotted the data without seasonality. Overall, the shape of the plot looks nearly identical. One noticeable difference is that the adjusted plot has a deeper trough on Jan 1980, telling us that part of the reason this dip wasn't as noticeable in the original plot is because it took place during a seasonal upswing.

```
livestock_plot <- aus_livestock |>
  filter(Animal == "Bulls, bullocks and steers" & State == "Victoria") |>
  mutate(Count = Count/1e3) |>
  autoplot(Count) +
```

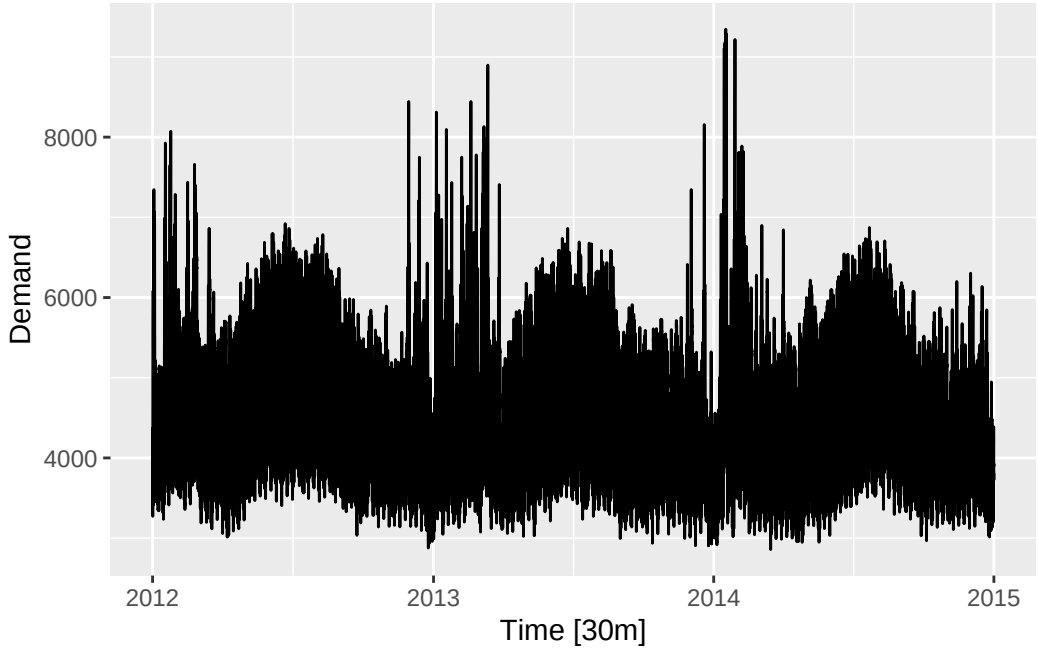
```
  ylab("Count (thousands)") +
  ggtitle("Non-Transformed Time Series")

adj_livestock_plot <- aus_livestock |>
  filter(Animal == "Bulls, bullocks and steers" & State == "Victoria") |>
  mutate(Count = Count/1e3) |>
  model(stl = STL(Count)) |>
  components() |>
  as_tibble() |>
  as_tsibble(index = Month, key = c(Animal, State)) |>
  autoplot(season_adjust) +
  ylab("Count of Animals (thousands)") +
  ggtitle("Seasonal Component Removed with STL")

ggarrange(livestock_plot, adj_livestock_plot, ncol = 1)
```



c. Victorian Electricity Demand from vic_elec.

Answer: I don't think this plot needs an adjustment.

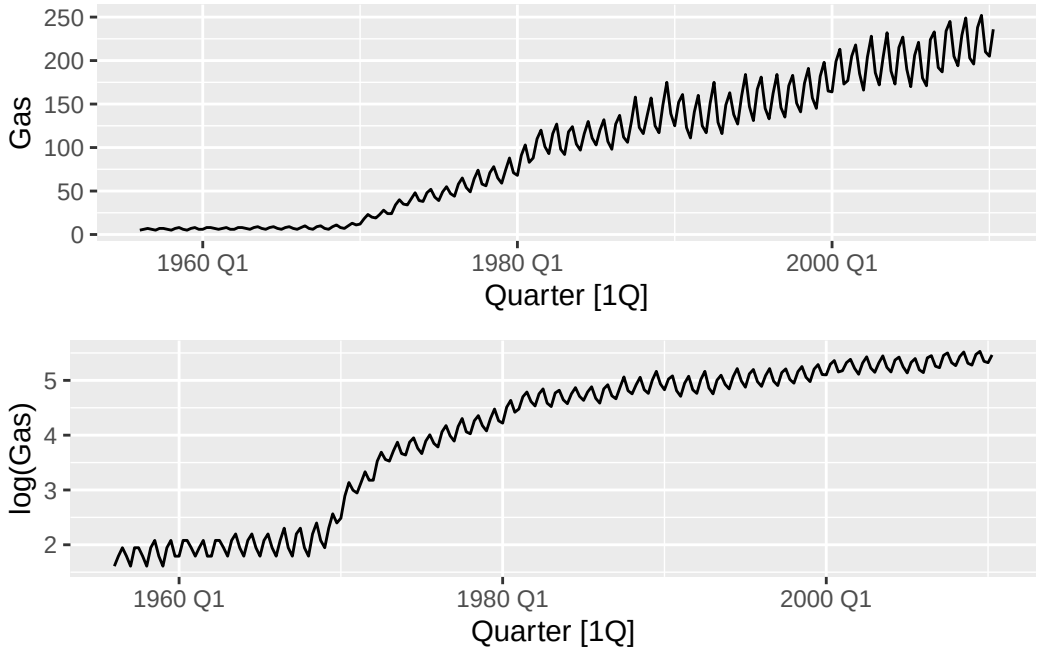
```
vic_elec |>
  autoplot(Demand)
```



d. Gas production from aus_production.

Answer: In the unadjusted plot we can see that the seasonal variation increases relative to scale. This calls for a mathematical adjustment. After a little guess and check with some simple adjustment operators, I found that taking the log of gas did a good job of normalizing the variation. The resulting plot has a more even seasonal pattern. This helps us see that the seasonal pattern is likely very consistent over time.

```
aus_prod_plot <- aus_production |>
  autoplot(Gas)

adj_aus_prod_plot <- aus_production |>
  autoplot(log(Gas))

ggarrange(aus_prod_plot, adj_aus_prod_plot, ncol = 1)
```

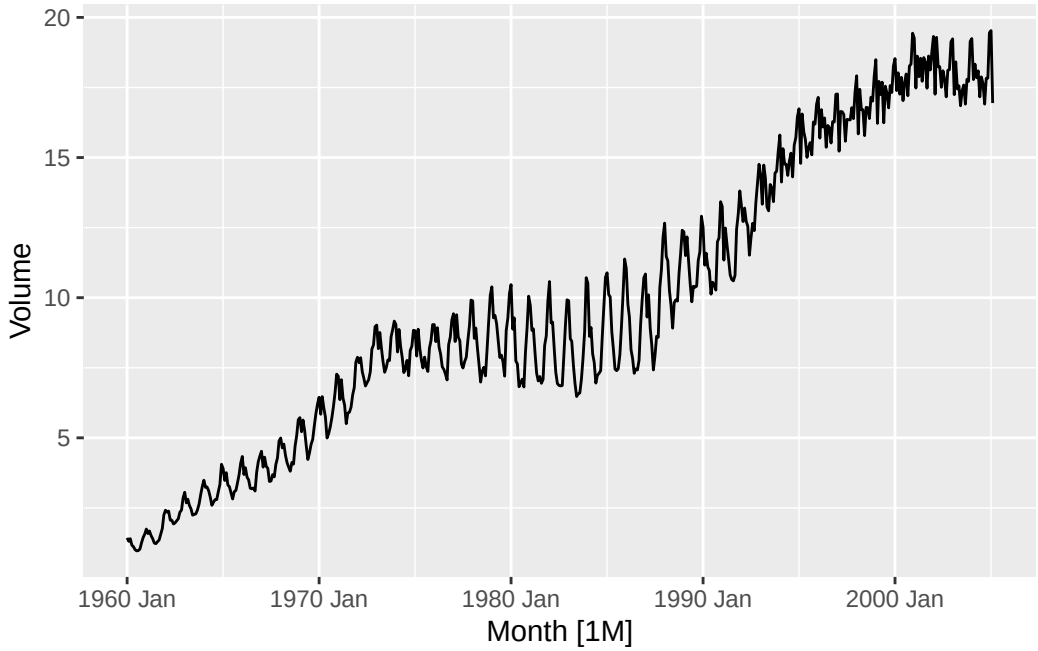


Question 3.3

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

Answer: The Box-Cox transformation is a type of mathematical transformation that can be useful when the goal is to get the most normalized version of the data, instead of opting for a "close-enough" log or power function. Mathematical transformations as a whole are only helpful with when the variation is in the data is relative to the scale of the data. This does not apply to canadian gas because the variation is not always relative to scale. At earlier times, Volume is low and variation is small; in the middle period, Volume is higher and variation has increased by a lot; but the later times have the highest Volume and small variation. Therefore, the variation is not relative to scale.

```
canadian_gas |>
  autoplot()
```



Question 2.7 & 3.4

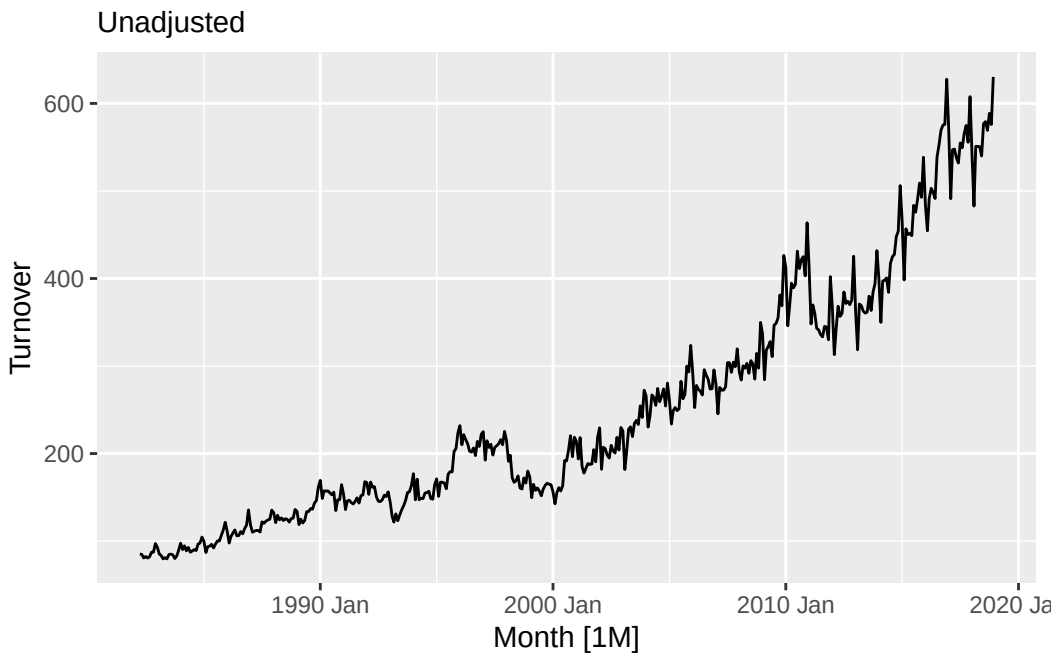
a. 2.7: Monthly Australian retail data is provided in aus_retail. Select one of the time series as follows (but choose your own seed value):

Answer: My time series, Takeaway food services, has an upward trend over time. There appears to be a seasonal pattern, as well as a cyclic jump followed by a corrective dip that brings it back on track. Turnover generally has its lowest point in February and gradually builds up to its peak in December before dropping in January and then more notable again in February. However, several years deviate quite significantly from this pattern. Based on the lag and ACF plots, the data seems highly autocorrelated.

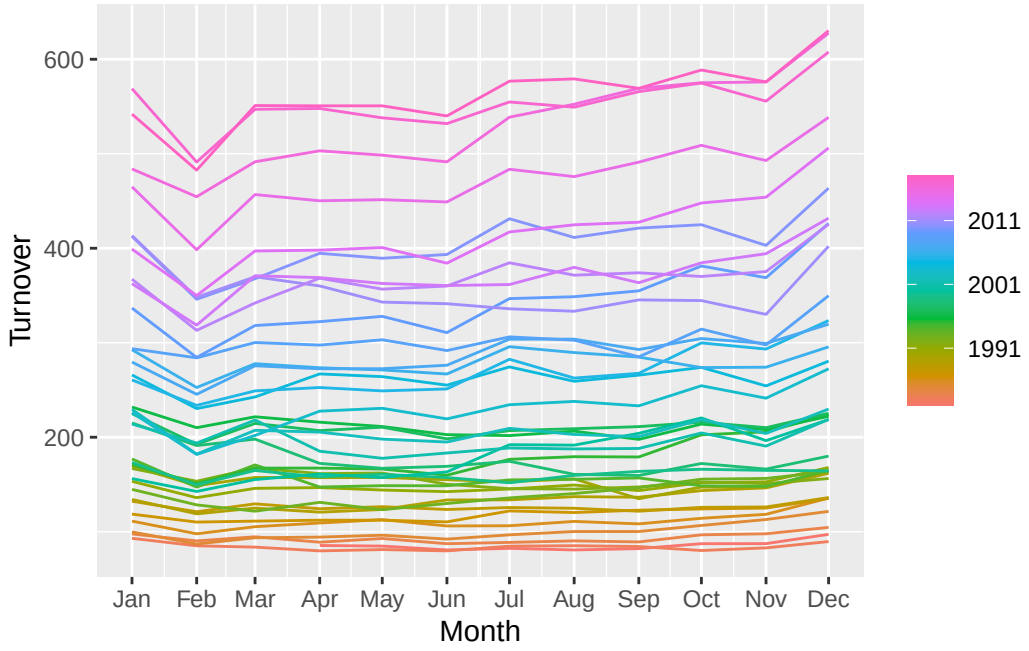
```
set.seed(624)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`, 1))
```

Explore your chosen retail time series using the following functions: autoplot(), gg_season(), gg_subseries(), gg_lag(), ACF() |> autoplot()

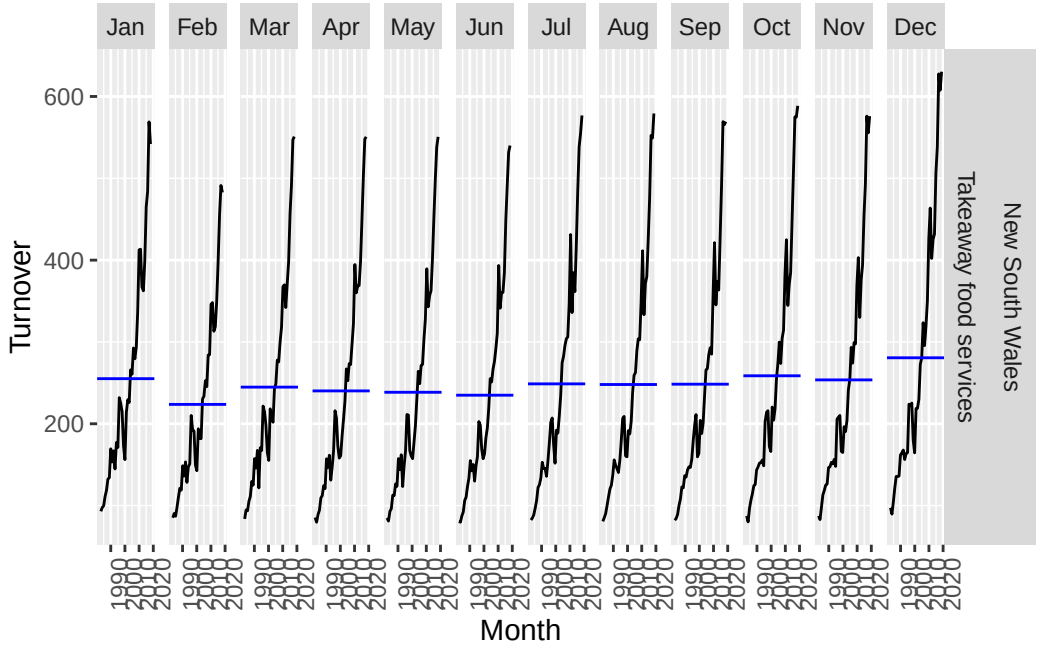
```
myseries_plot <- myseries |>
  autoplot(Turnover) +
  labs(subtitle = "Unadjusted")
myseries_plot
```



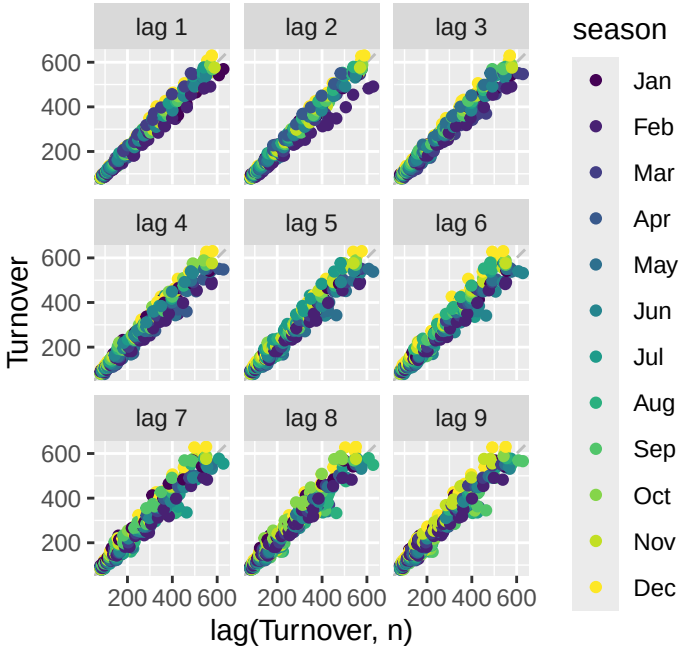

```
myseries |>
  gg_season(Turnover)
```



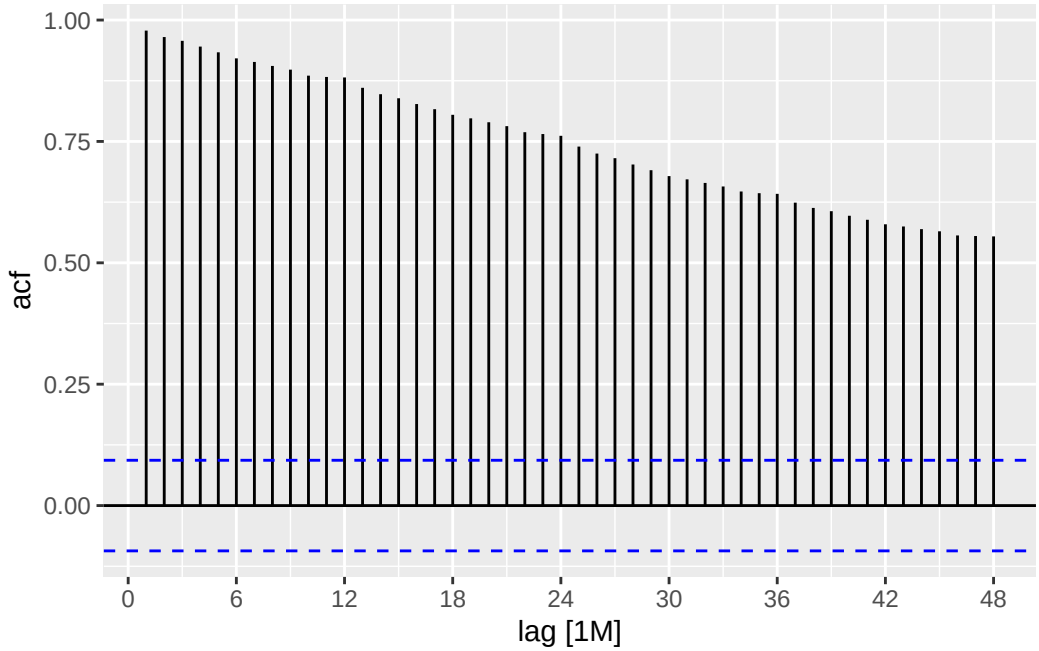

```
myseries |>
  gg_subseries(Turnover)
```



```
myseries |>
  gg_lag(Turnover, geom = "point")
```




```
myseries |>
  ACF(Turnover, lag_max = 48) |>
  autoplot()
```



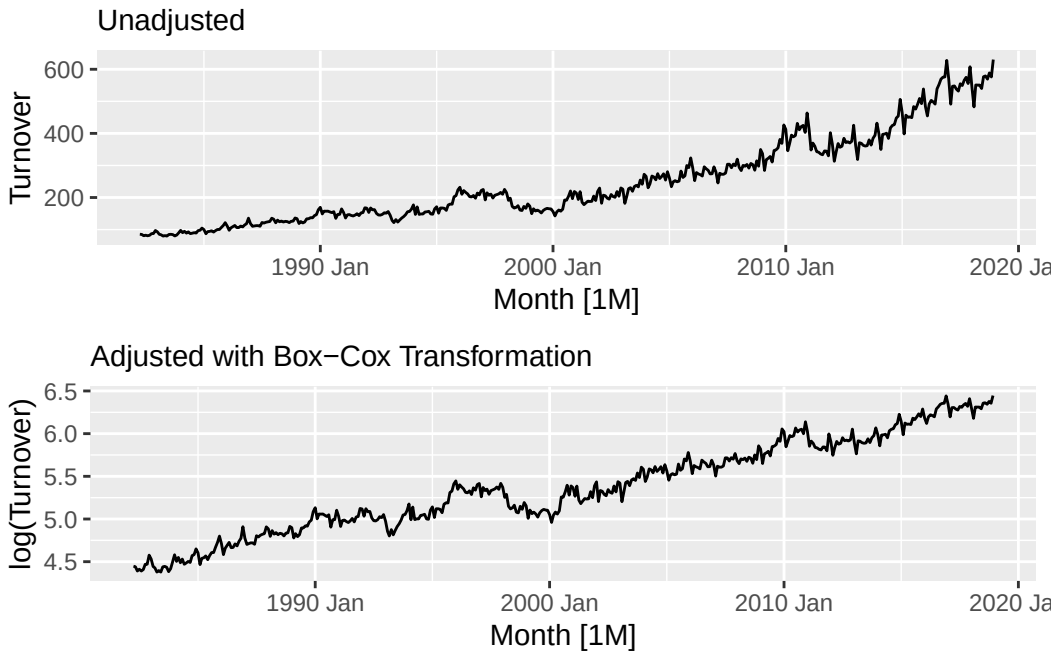
b. 3.4: What Box-Cox transformation would you select for your retail data (from Exercise 7 in Section 2.10)

Answer: Below, I computed the lambda for my series using the Guerrero transformation. Since, the lambda is very close to 0, I would use a log transformation on my data. As we can see, this has done a great job of normalizing the variation.

```
myseries |>
  features(Turnover, features = guerrero)
```

```
# A tibble: 1 x 3
  State           Industry                lambda_guerrero
  <chr>           <chr>                             <dbl>
1 New South Wales Takeaway food services          0.00214
```

```
adj_myseries_plot <- myseries |>
  autoplot(log(Turnover)) +
  labs(subtitle = "Adjusted with Box-Cox Transformation")

ggarrange(myseries_plot, adj_myseries_plot, ncol = 1)
```



Question 3.5

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.

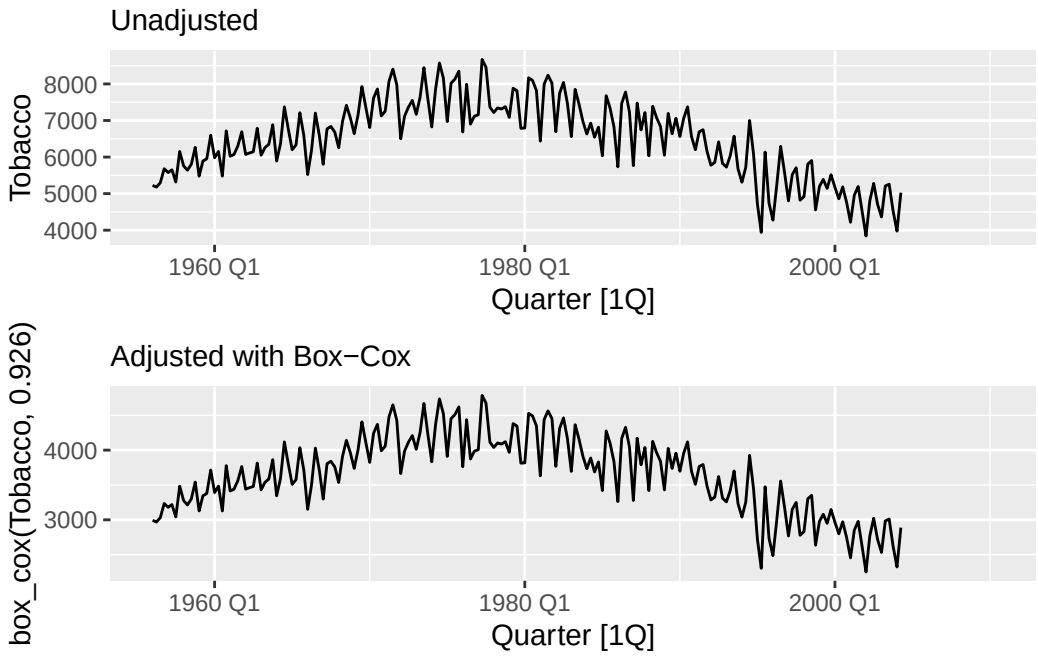
a. Tobacco from aus_production.

Answer: Since the lambda from the Guerrero transformation is so close to 1, we don't necessarily need a Box-Cox transformation. Comparing the two plots, there isn't much difference.

```
aus_production |>
  features(Tobacco, features = guerrero)
```

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

```
tobacco_plot <- aus_production |>
  autoplot(Tobacco) +
  labs(subtitle = "Unadjusted")

adj_tobacco_plot <- aus_production |>
  autoplot(box_cox(Tobacco, 0.926)) +
  labs(subtitle = "Adjusted with Box-Cox")

ggarrange(tobacco_plot, adj_tobacco_plot, ncol = 1)
```

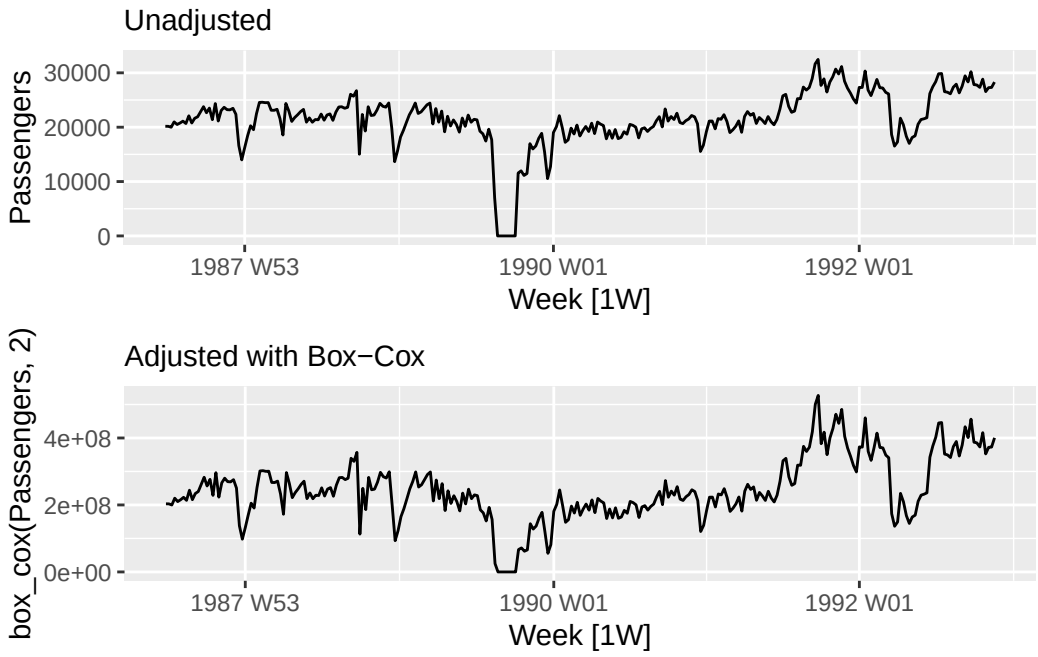


b. Economy class passengers between Melbourne and Sydney from ansett

```
ansett_filtered <- ansett |>
  filter(Class == "Economy" & Airports == "MEL-SYD")

ansett_filtered |>
  features(Passengers, features = guerrero)
```

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

```
ansett_plot <- ansett_filtered |>
  autoplot(Passengers) +
  labs(subtitle = "Unadjusted")

adj_ansett_plot <- ansett_filtered |>
  autoplot(box_cox(Passengers, 2)) +
  labs(subtitle = "Adjusted with Box-Cox")

ggarrange(ansett_plot, adj_ansett_plot, ncol = 1)
```

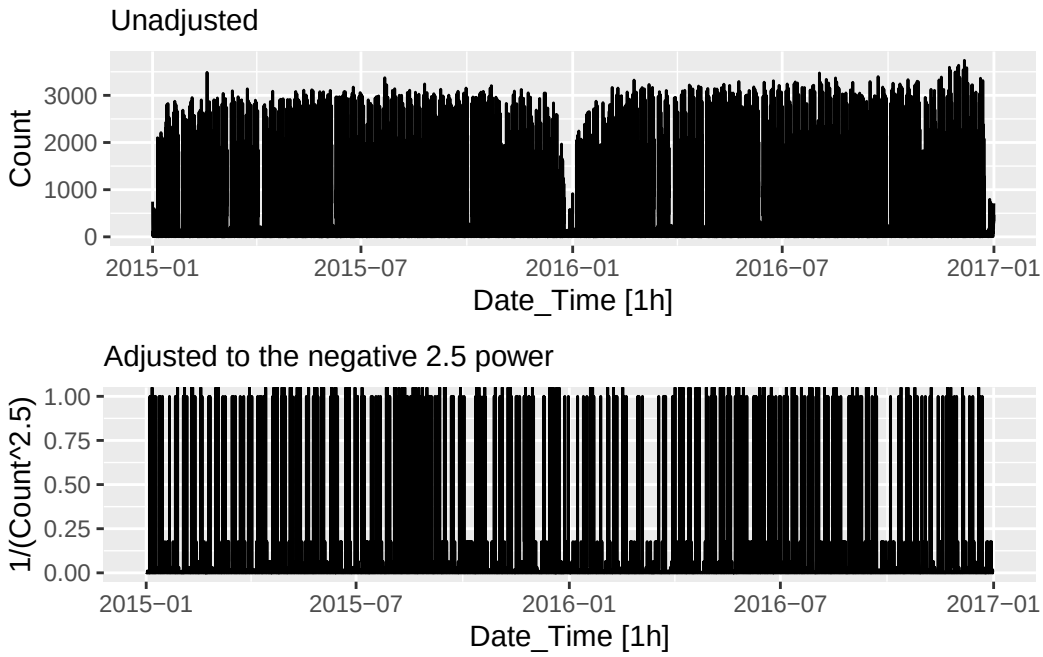


c. Pedestrian counts at Southern Cross Station from pedestrian

```
pedestrian_filtered <- pedestrian |>
  filter(Sensor == "Southern Cross Station")

pedestrian_filtered |>
  features(Count, features = guerrero)
```

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

```
pedestrian_plot <- pedestrian_filtered |>
  autoplot(Count) +
  labs(subtitle = "Unadjusted")

adj_pedestrian_plot <- pedestrian_filtered |>
  autoplot(1/(Count^2.5)) +
  labs(subtitle = "Adjusted to the negative 2.5 power")

ggarrange(pedestrian_plot, adj_pedestrian_plot, ncol = 1)
```



Question 3.7

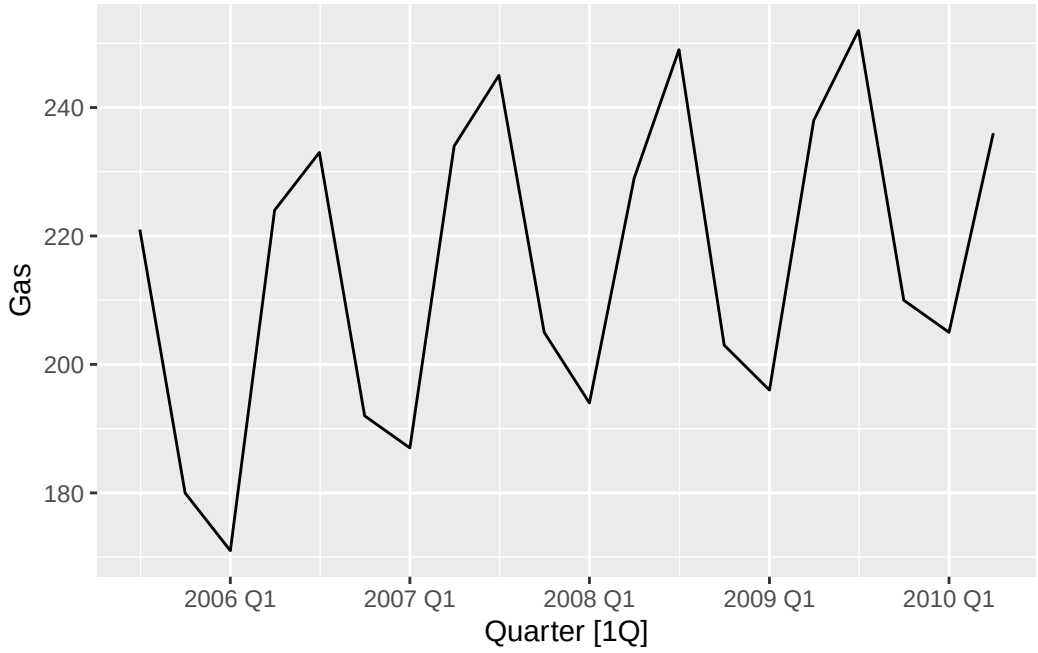
Consider the last five years of the Gas data from aus_production:

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

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

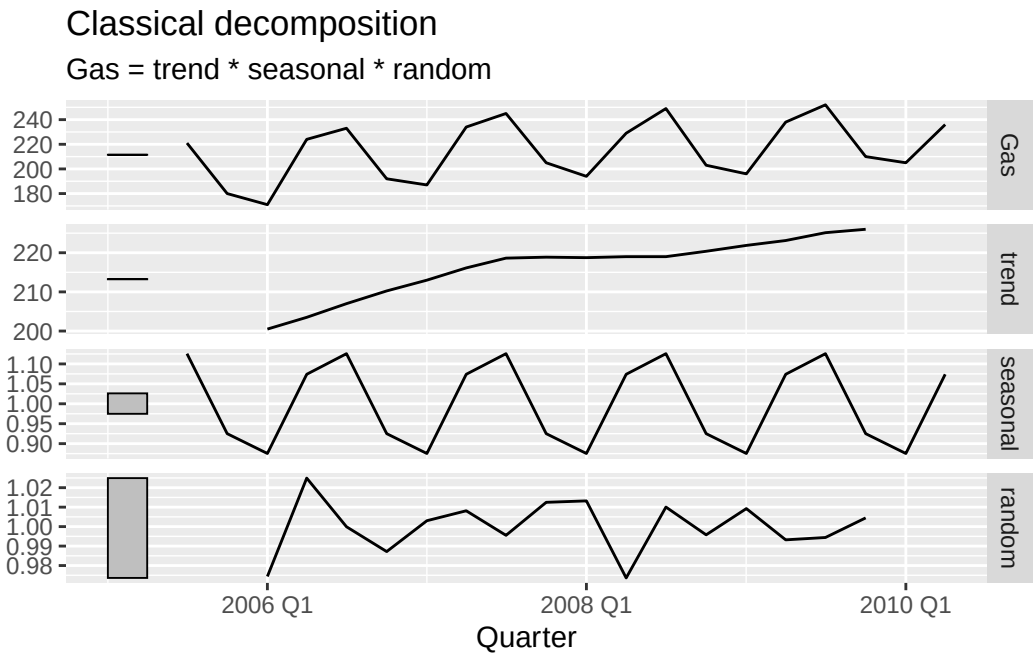
Answer: Gas production has a seasonal pattern that peaks in Q2/Q3 and troughs in Q1/Q4. There is also a positive trend.

```
gas |>
  autoplot(Gas)
```



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

```
gas |>
  model(classical_decomposition(Gas, type = "multiplicative")) |>
  components() |>
  autoplot()
```

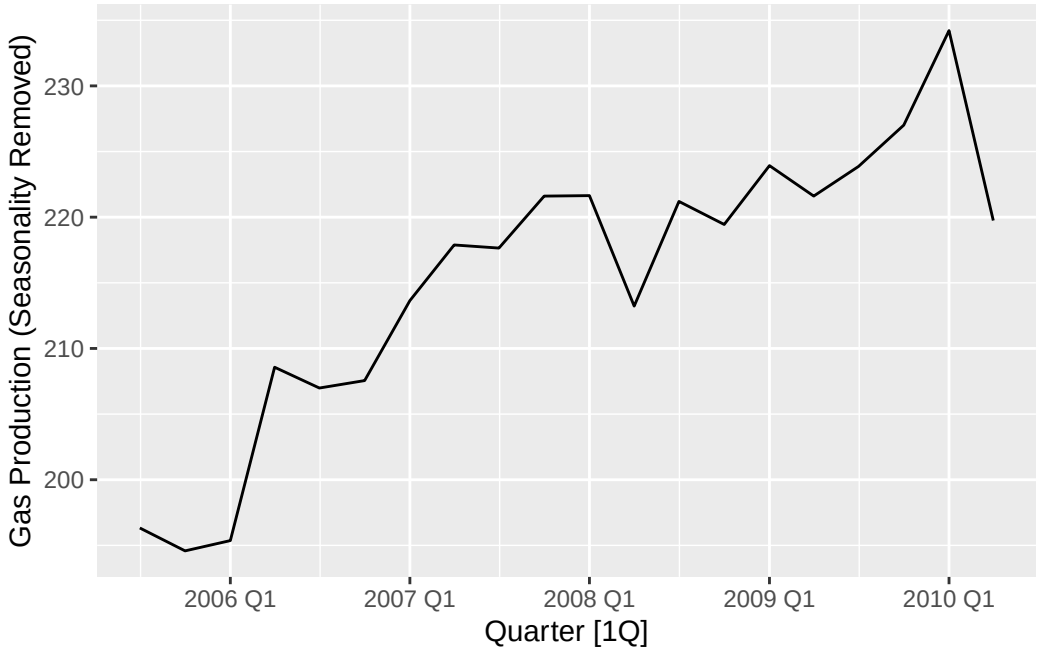


c. Do the results support the graphical interpretation from part a?

Answer: Yes. We can see in the plots of the trend and seasonal components that there is a positive trend and bi-quarterly seasonal component.

d. Compute and plot the seasonally adjusted data.

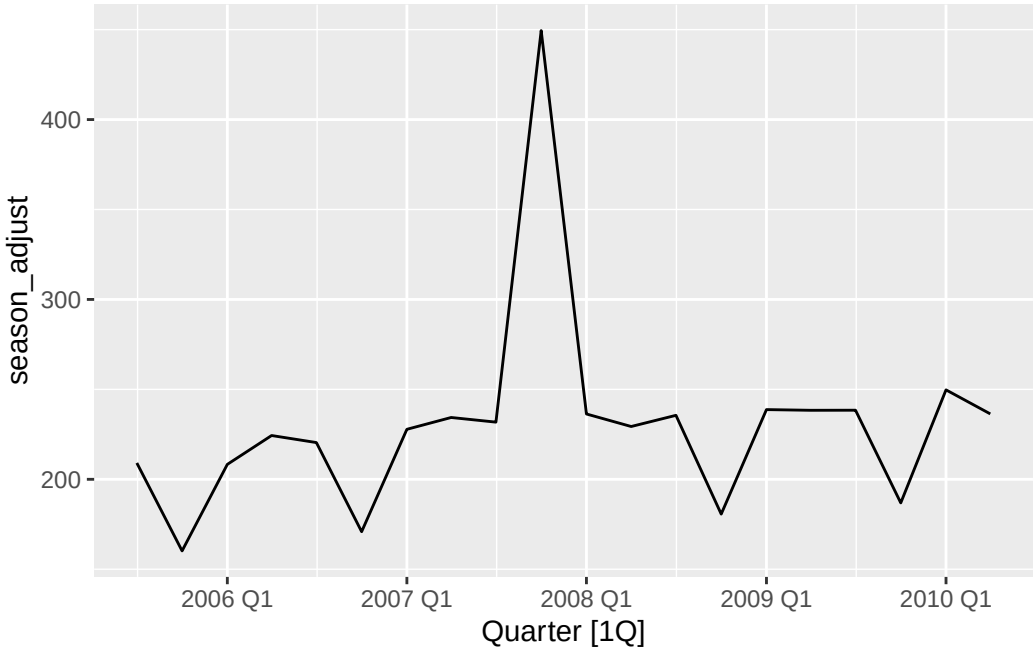
```
gas |>
  model(classical_decomposition(Gas, type = "multiplicative")) |>
  components() |>
  update_tsibble(key = NULL) |>
  autoplot(season_adjust) +
  ylab("Gas Production (Seasonality Removed)")
```



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?

Answer: The addition of the outlier makes it hard to see any trend in the seasonally adjusted data.

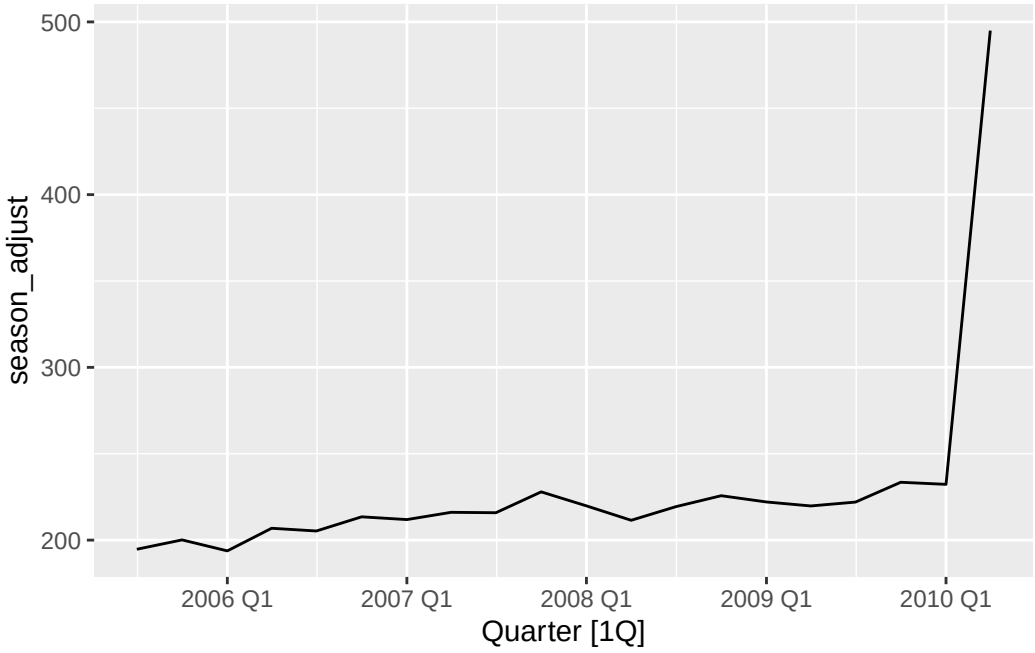
```
gas |>
  mutate(Gas = ifelse(row_number() == 10, Gas + 300, Gas)) |>
  model(classical_decomposition(Gas, type = "multiplicative")) |>
  components() |>
  update_tsibble(key = NULL) |>
  autoplot(season_adjust)
```



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

Answer: Yes. Putting an outlier at the beggining or end causes there to be less variation from our variable to be attributed to the seasonally adjusted data.

```
gas |>
  mutate(Gas = ifelse(row_number() == 20, Gas + 300, Gas)) |>
  model(classical_decomposition(Gas, type = "multiplicative")) |>
  components() |>
  update_tsibble(key = NULL) |>
  autoplot(season_adjust)
```

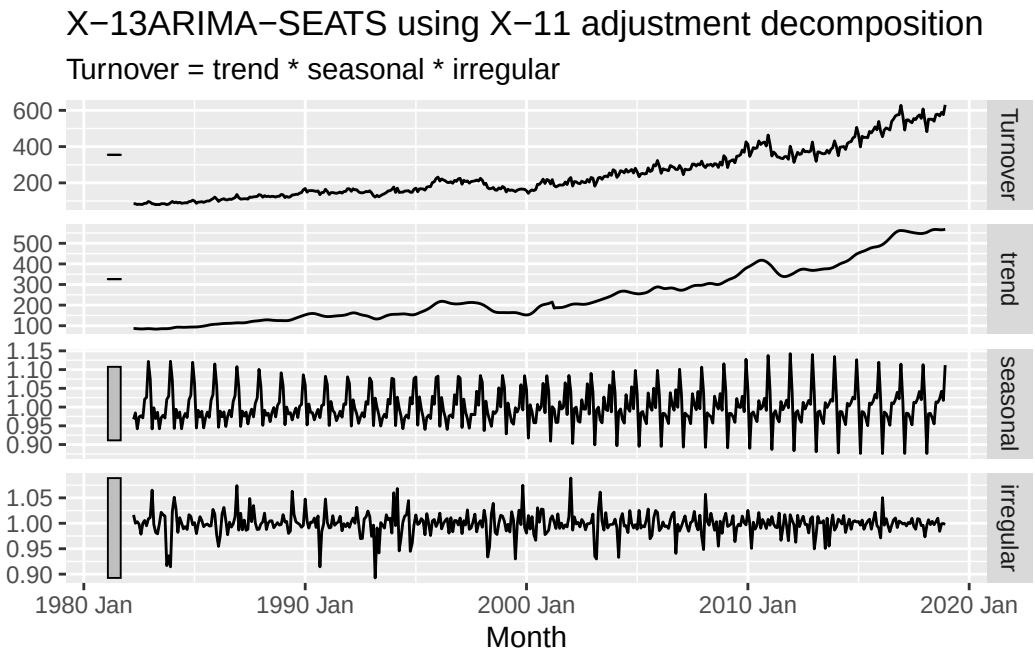


Question 3.8

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?

Answer: The seasonal component has changed in scale and shape over time. Over time, the random component has decreased variation.

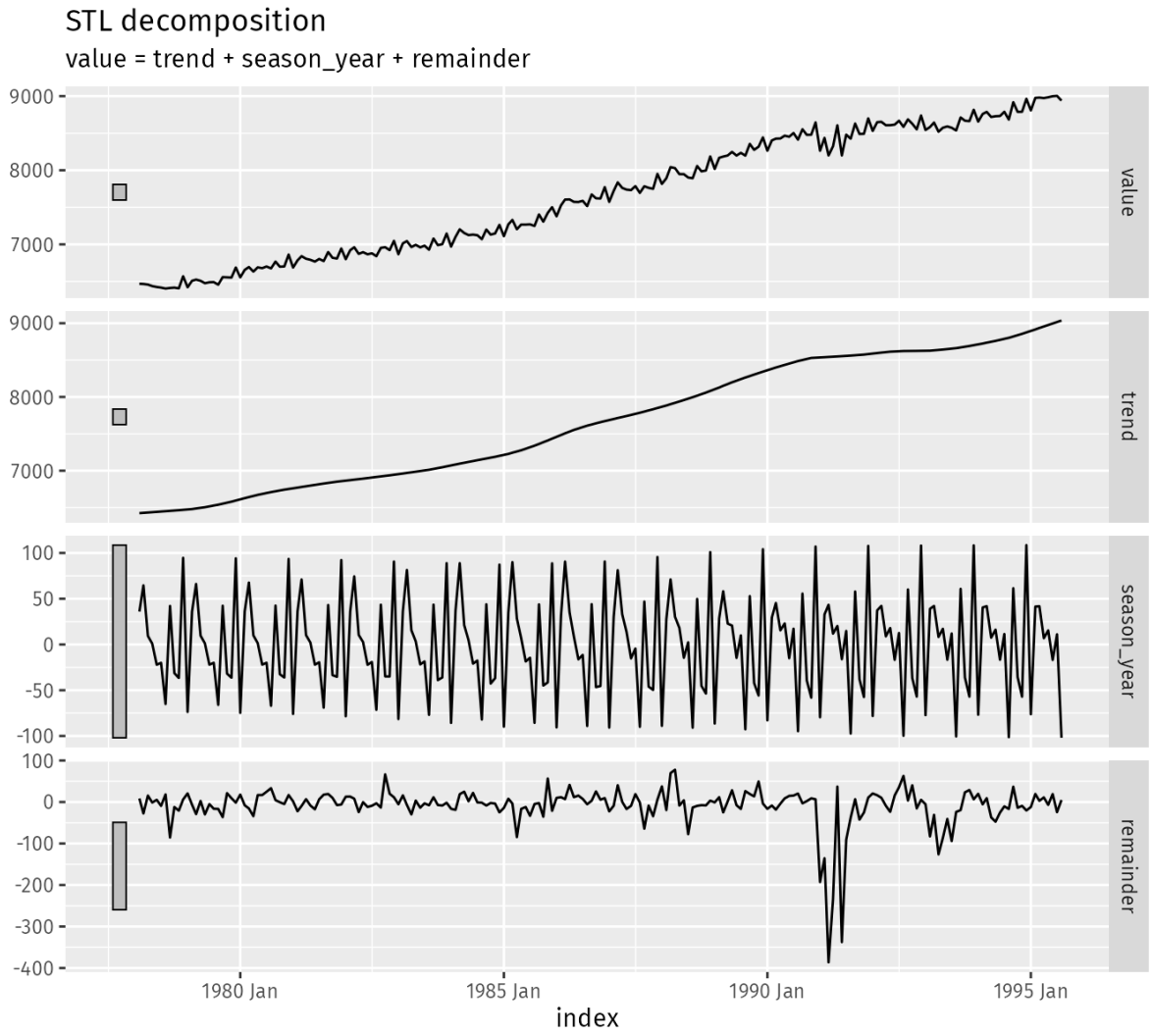
```
myseries |>
  model(x11 = X_13ARIMA_SEATS(Turnover ~ x11())) |>
  components() |>
  autoplot()
```



Question 3.9

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.

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.



Answer: The results of the decomposition highlights the trend that's already pretty visible in the original data. For the seasonal component, we can see the shape of the plot has changed over time, indicating a change to the seasonal pattern over time. Furthermore, the seasonal subseries plots give us additional insight into the shape of the seasonality and which months did not have consistent seasonal patterns (i.e. March goes up and over time). Lastly, the remainder plot shows us that there are some huge outliers around 1992-1993.

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

Answer: Yes. As stated in part a. of this question, the recession appears as an outlier in the remainder plot.