# Data 624 - HW2 (Fall 2024)

Khyati Naik

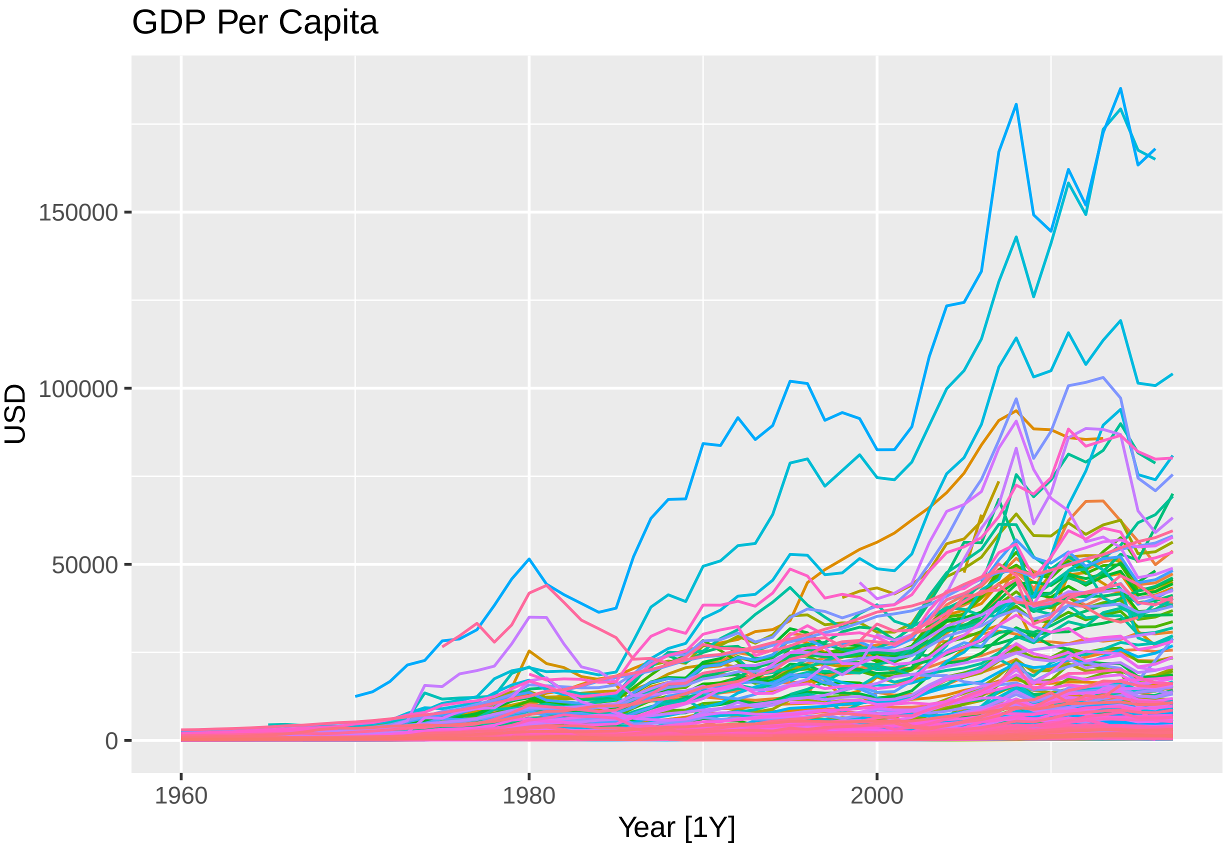
```
library(fpp3)
```

```
## -- Attaching packages -------------------------------------------- fpp3 1.0.0 --

## v tibble      3.2.1     v tsibble     1.1.3
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.0     v feasts      0.3.2
## v lubridate   1.9.2     v fable       0.3.4
## v ggplot2     3.5.1     v fabletools  0.4.2

## -- Conflicts -------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

### 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?

```
data("global_economy")

# Assuming global_economy dataset has columns: Country, Year, GDP, Population
# Calculate GDP per capita
global_economy <- global_economy %>%
  mutate(GDPperCapita = GDP / Population)

# Plot GDP per capita over time for each country
global_economy %>% autoplot(GDPperCapita, show.legend =  FALSE) +
  labs(title= "GDP Per Capita", y = "USD")
```

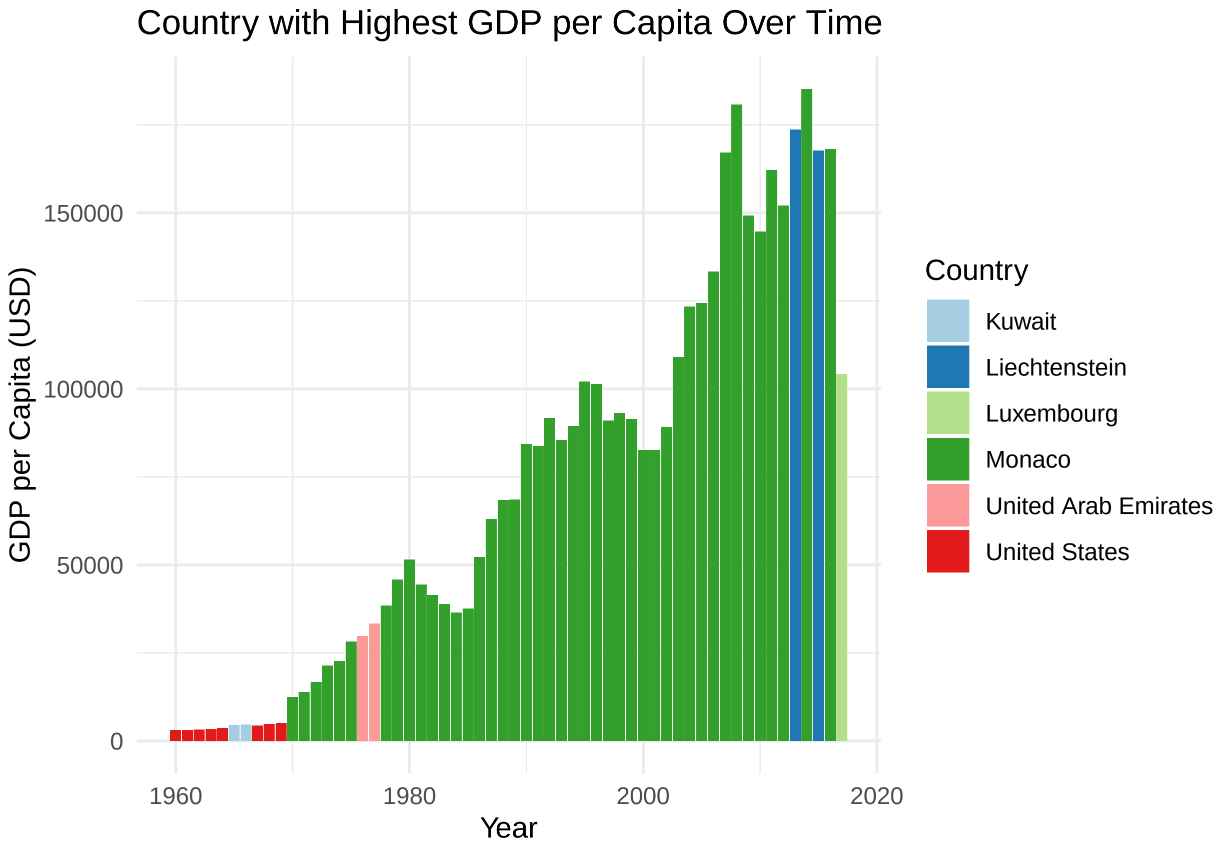
```
## Warning: Removed 3242 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



```r
# Find the country with the highest GDP per capita in each year
highest_gdp_per_capita <- global_economy %>%
  index_by(Year) %>%
  filter(GDPperCapita == max(GDPperCapita, na.rm = TRUE)) %>%
  select(Year, Country, GDPperCapita)%>%arrange(Year,desc(GDPperCapita))

#Which country has the highest GDP per capita?
head(highest_gdp_per_capita  %>%
  arrange(desc(GDPperCapita)), n = 5)
```

```
## # A tsibble: 5 x 3 [1Y]
## # Key:       Country [2]
## # Groups:    @ Year [5]
##    Year Country       GDPperCapita
##   <dbl> <fct>                <dbl>
## 1  2014 Monaco             185153.
## 2  2008 Monaco             180640.
## 3  2013 Liechtenstein      173528.
## 4  2016 Monaco             168011.
## 5  2015 Liechtenstein      167591.
```

```r
# Plot the country with the highest GDP per capita over time
ggplot(highest_gdp_per_capita, aes(x = Year, y = GDPperCapita, fill = Country)) +
  geom_bar(stat = "identity") +  # Use geom_bar with stat = "identity" to show actual values
  labs(title = "Country with Highest GDP per Capita Over Time",
      x = "Year",
      y = "GDP per Capita (USD)") +
  theme_minimal() +
  scale_fill_brewer(palette = "Paired")  # Use a color palette for better visualization
```



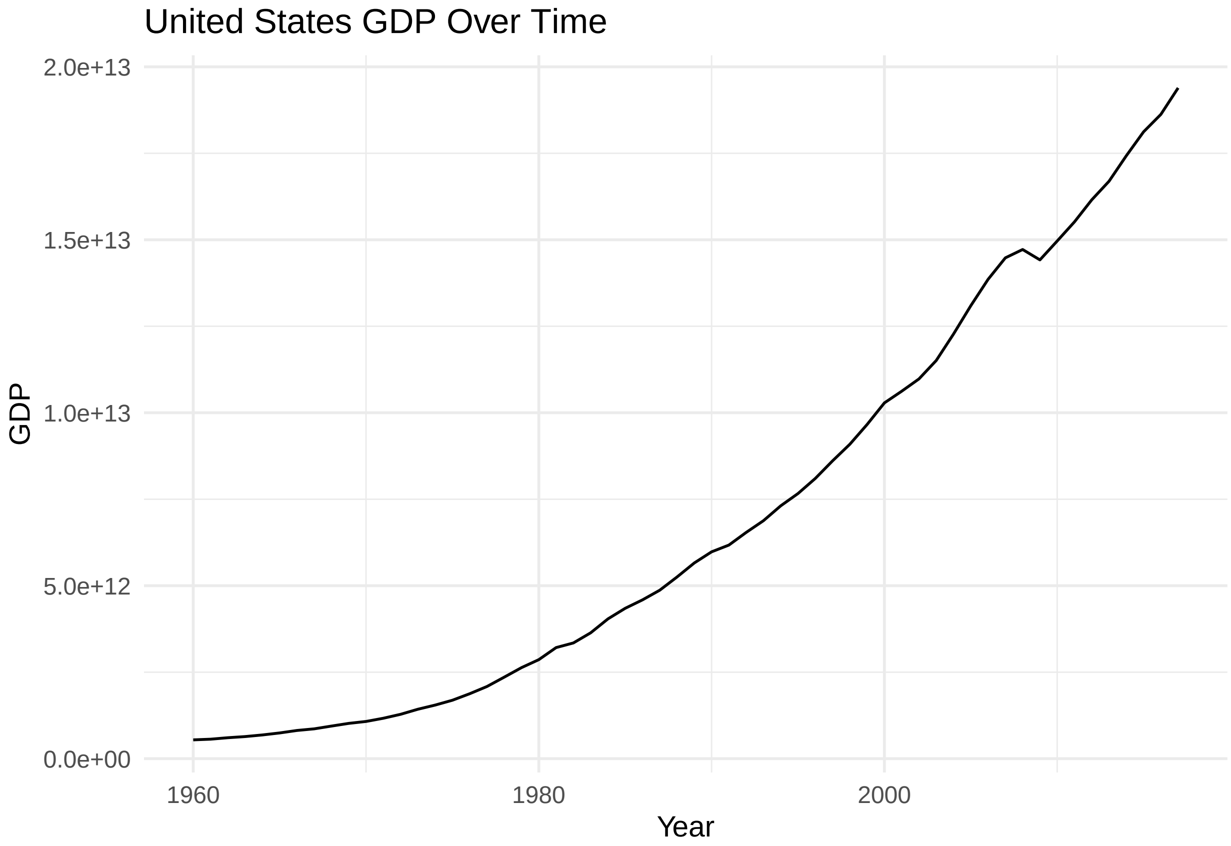
Monaco had the highest GDP per capita in 2014, among all countries from 1960 to 2017. Additionally, Monaco holds the record for having the highest GDP per capita the most number of times during this period.

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

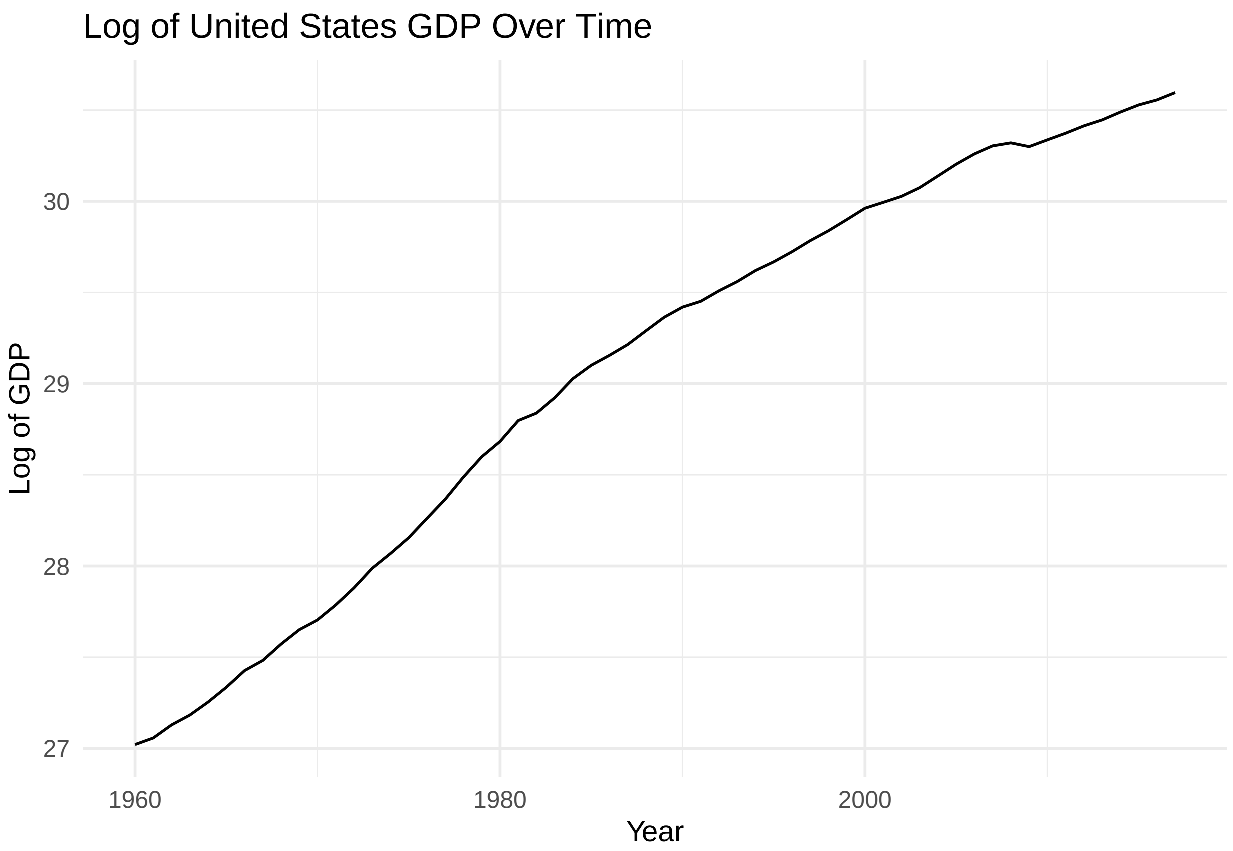
- United States GDP from global_economy.
- Slaughter of Victorian "Bulls, bullocks and steers" in aus_livestock.
- Victorian Electricity Demand from vic_elec.
- Gas production from aus_production.

```
# 1. United States GDP from global_economy
# Assuming global_economy is a tsibble with key Country and index Year
# Filter for United States GDP
us_gdp <- global_economy %>%
  filter(Country == "United States") %>%
  select(Year, GDP)
```

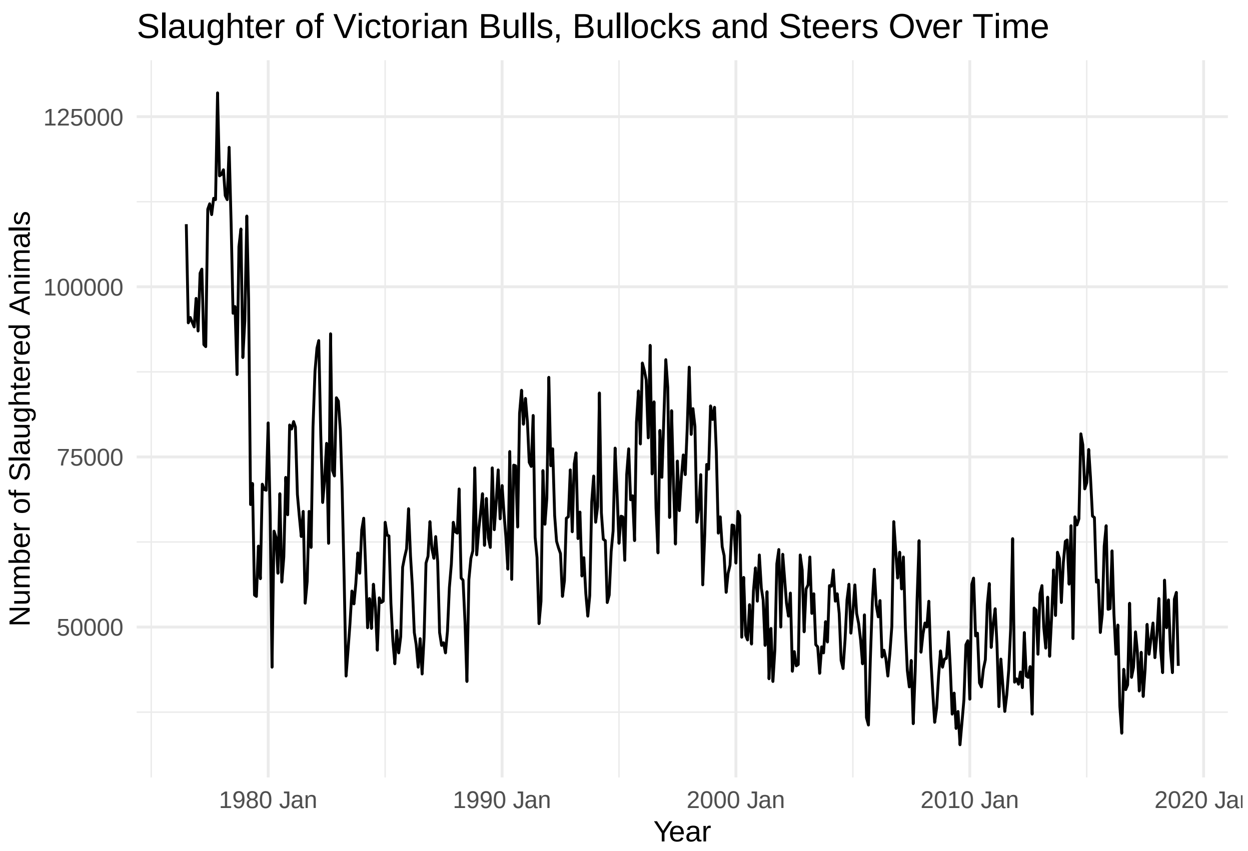
```r
# Plot United States GDP over time
ggplot(us_gdp, aes(x = Year, y = GDP)) +
  geom_line() +
  labs(title = "United States GDP Over Time",
       x = "Year", y = "GDP") +
  theme_minimal()
```



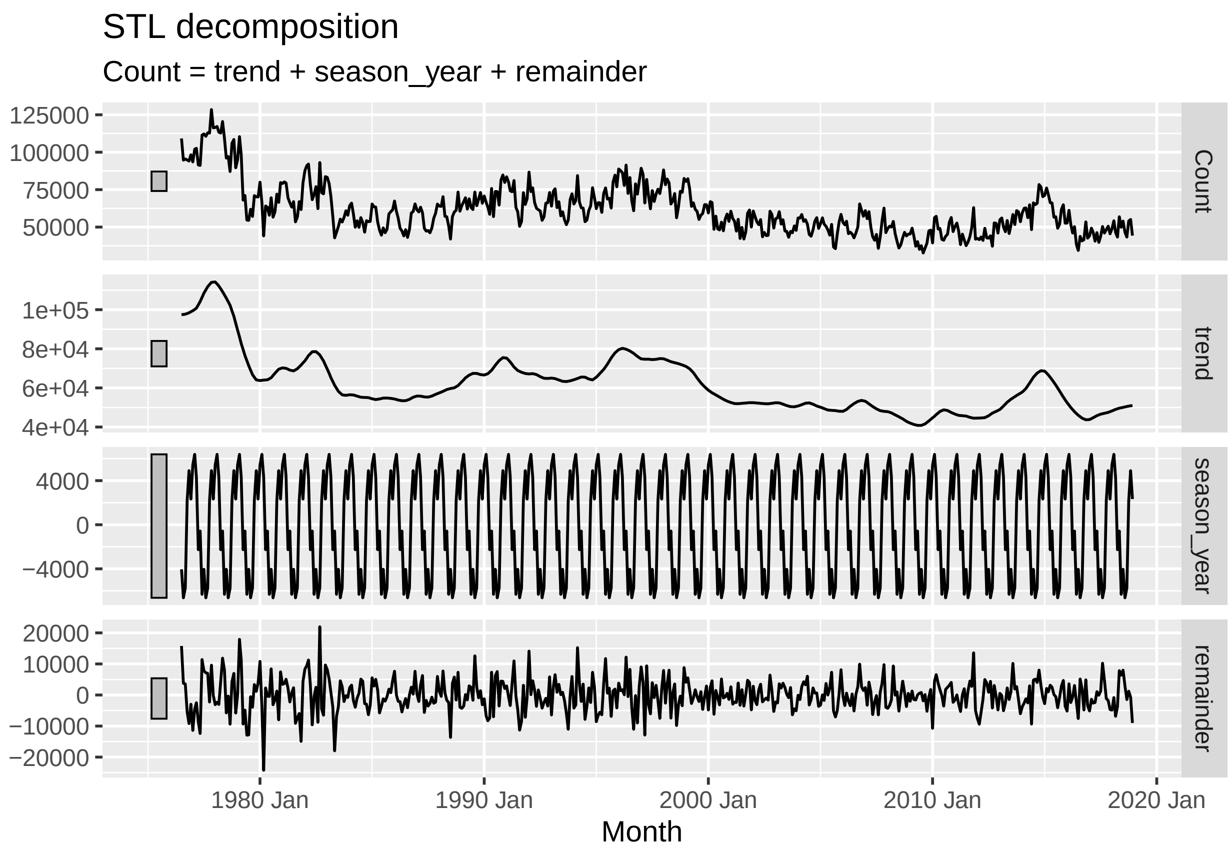
```r
# Log transformation might be useful for GDP to stabilize variance
us_gdp <- us_gdp %>%
  mutate(log_GDP = log(GDP))

# Plot log-transformed United States GDP
ggplot(us_gdp, aes(x = Year, y = log_GDP)) +
  geom_line() +
  labs(title = "Log of United States GDP Over Time",
       x = "Year", y = "Log of GDP") +
  theme_minimal()
```



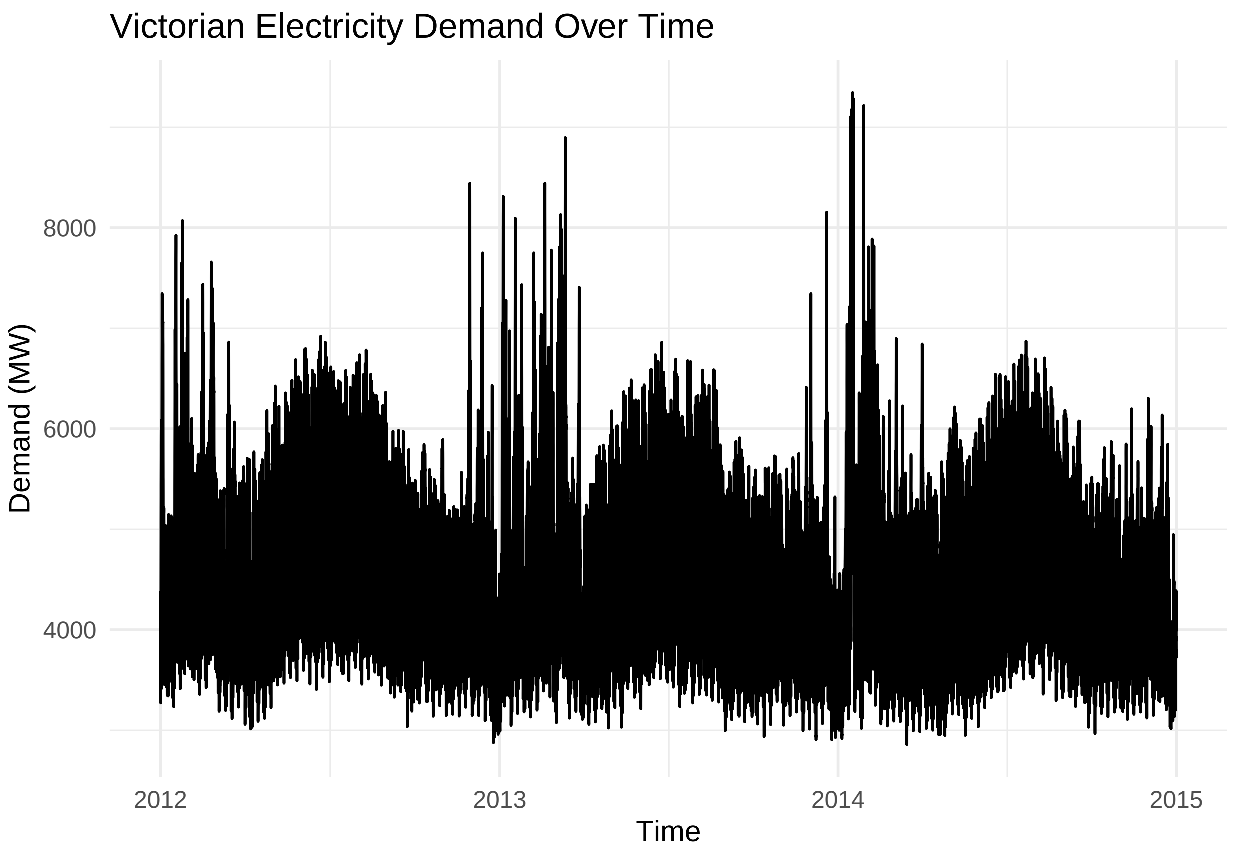
```r
# 2. Slaughter of Victorian "Bulls, bullocks and steers" in aus_livestock -----
# Assuming aus_livestock is a tsibble with key Livestock and index Month
# Filter for "Bulls, bullocks and steers"
vic_bulls <- aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers", State == "Victoria")

# Plot slaughter of Victorian "Bulls, bullocks and steers" over time
ggplot(vic_bulls, aes(x = Month, y = Count)) +
  geom_line() +
  labs(title = "Slaughter of Victorian Bulls, Bullocks and Steers Over Time",
       x = "Year", y = "Number of Slaughtered Animals") +
  theme_minimal()
```



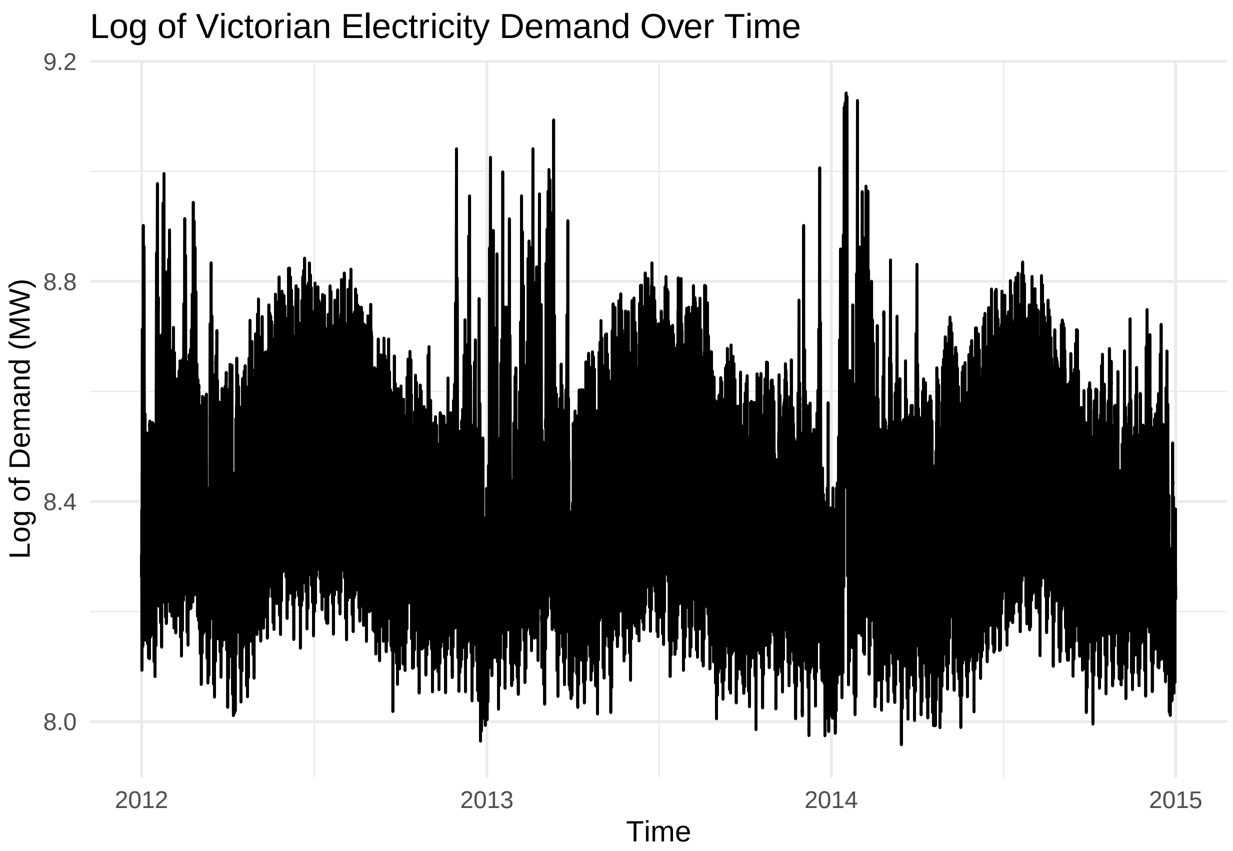
```
# Seasonal decomposition might be appropriate here
vic_bulls_decomp <- vic_bulls %>%
  model(STL(Count ~ season(window = "periodic"))) %>%
  components()

autoplot(vic_bulls_decomp)
```



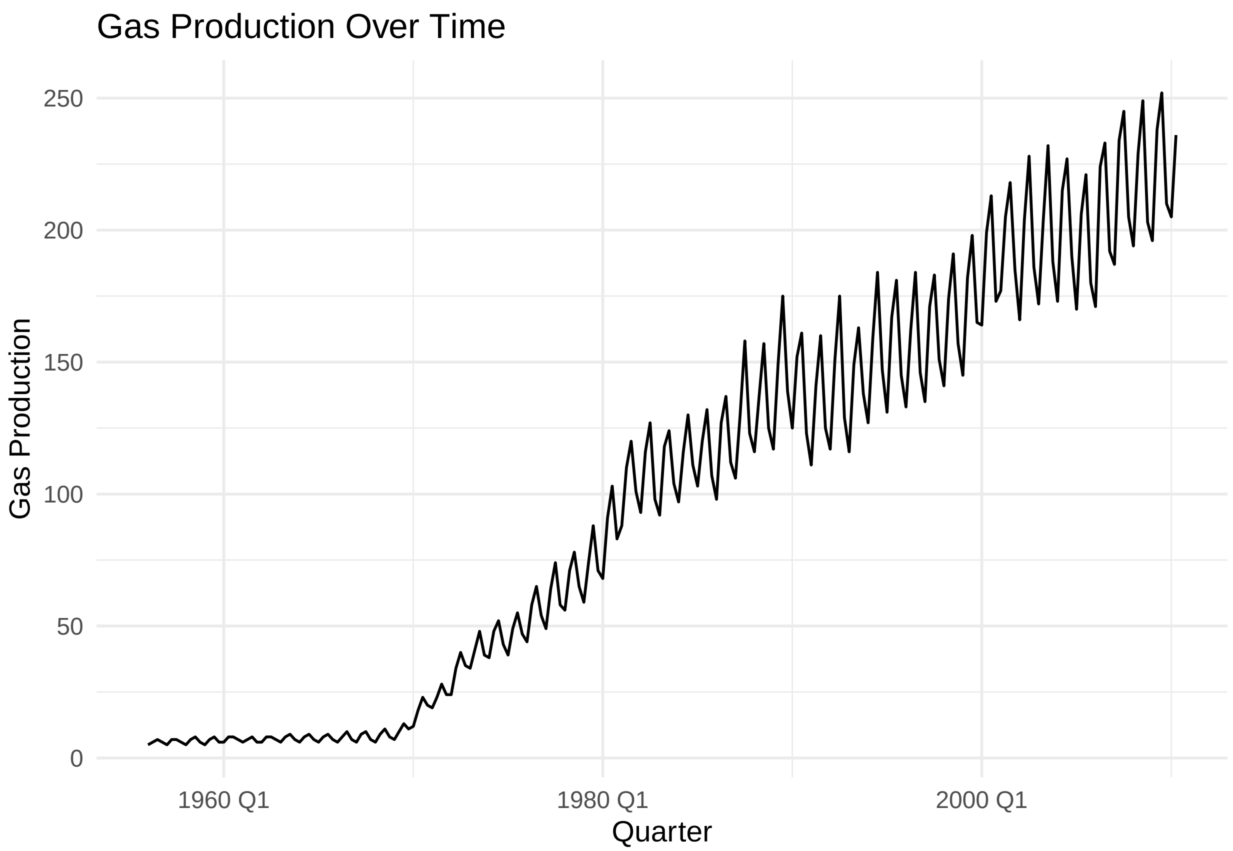
```r
# 3. Victorian Electricity Demand from vic_elec -----
# Assuming vic_elec is a tsibble with key Demand and index Time
# Plot Victorian electricity demand
ggplot(vic_elec, aes(x = Time, y = Demand)) +
  geom_line() +
  labs(title = "Victorian Electricity Demand Over Time",
       x = "Time", y = "Demand (MW)") +
  theme_minimal()
```



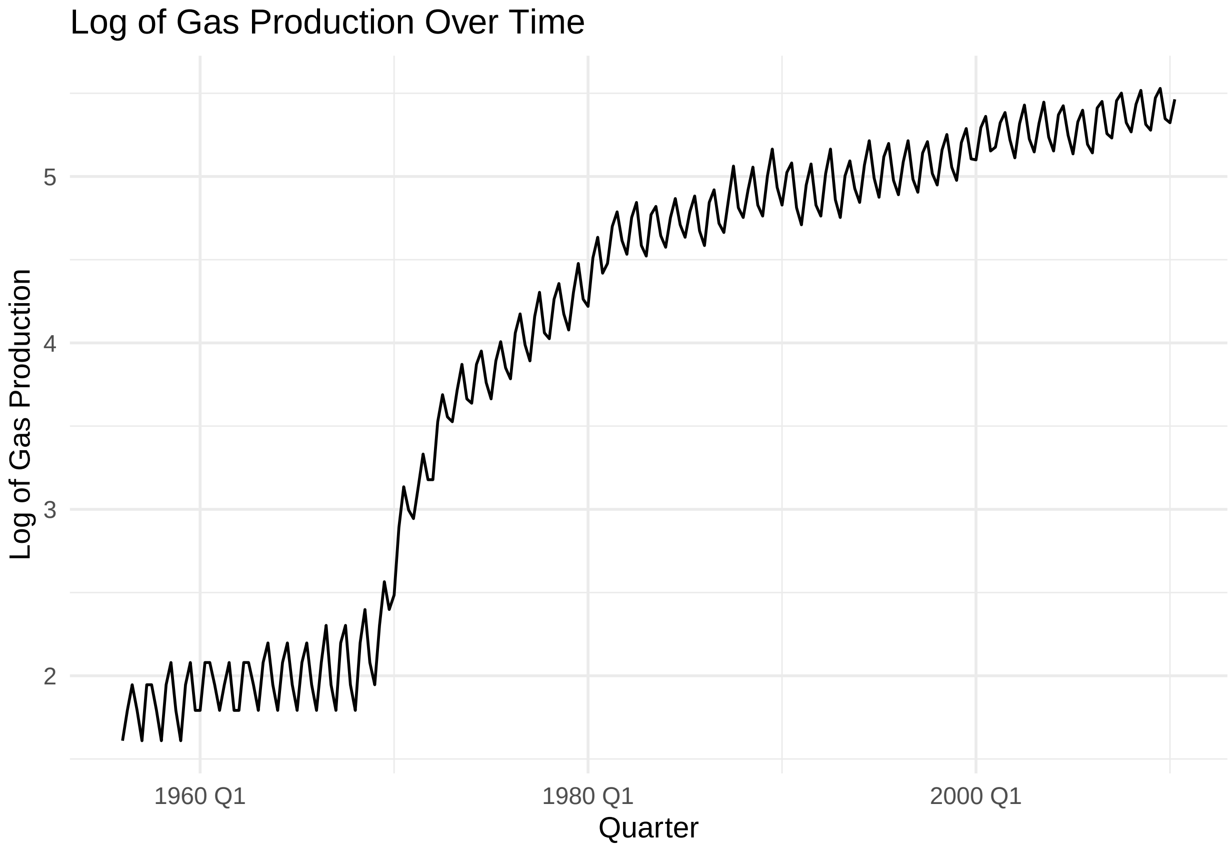
```r
# Log transformation could stabilize variance
vic_elec <- vic_elec %>%
  mutate(log_Demand = log(Demand))

# Plot log-transformed Victorian electricity demand
ggplot(vic_elec, aes(x = Time, y = log_Demand)) +
  geom_line() +
  labs(title = "Log of Victorian Electricity Demand Over Time",
       x = "Time", y = "Log of Demand (MW)") +
  theme_minimal()
```



```
# 4. Gas production from aus_production -----
# Assuming aus_production is a tsibble with key Gas and index Quarter
# Filter for gas production
gas_production <- aus_production %>%
  select(Quarter, Gas)

# Plot gas production over time
ggplot(gas_production, aes(x = Quarter, y = Gas)) +
  geom_line() +
  labs(title = "Gas Production Over Time",
       x = "Quarter", y = "Gas Production") +
  theme_minimal()
```

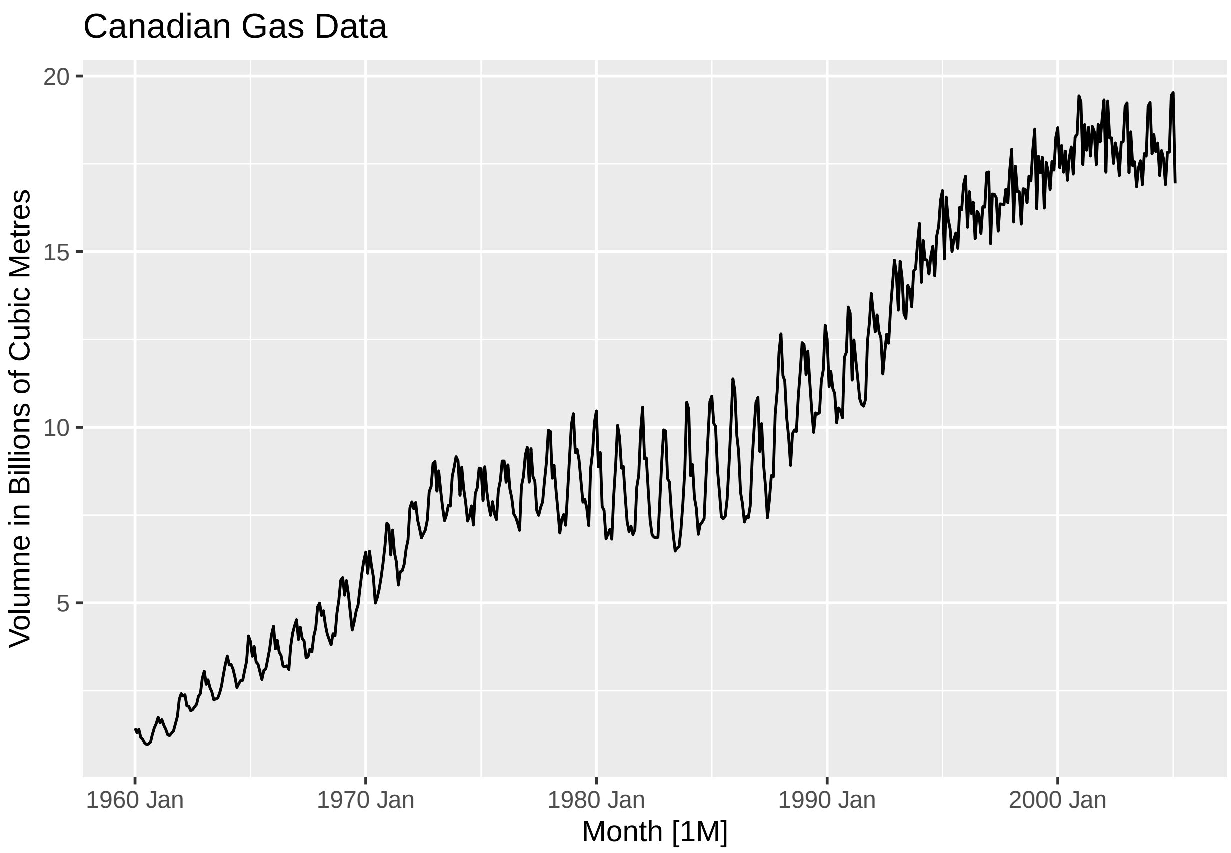


```
# Log transformation for gas production
gas_production <- gas_production %>%
  mutate(log_Gas = log(Gas))

# Plot log-transformed gas production
ggplot(gas_production, aes(x = Quarter, y = log_Gas)) +
  geom_line() +
  labs(title = "Log of Gas Production Over Time",
       x = "Quarter", y = "Log of Gas Production") +
  theme_minimal()
```



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

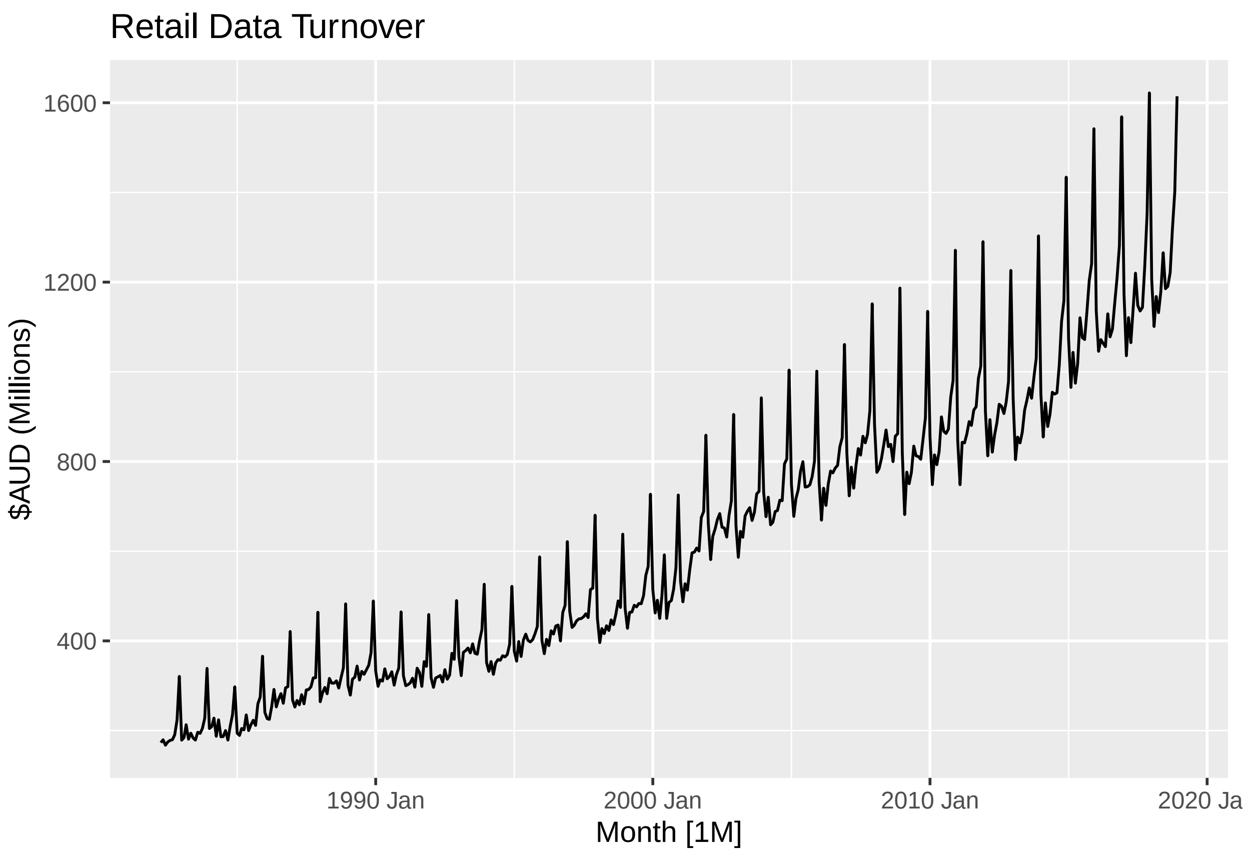
```
canadian_gas %>% autoplot(Volume) +
  labs(title = "Canadian Gas Data", y ="Volumne in Billions of Cubic Metres")
```



The Box-Cox transformation is designed to stabilize variance and make data more normally distributed by transforming the data with a parameter lambda. Since the range of the canadian gas data is very small (approx 0.9 to 19.5), the Box-Cox transformation might not be very effective.

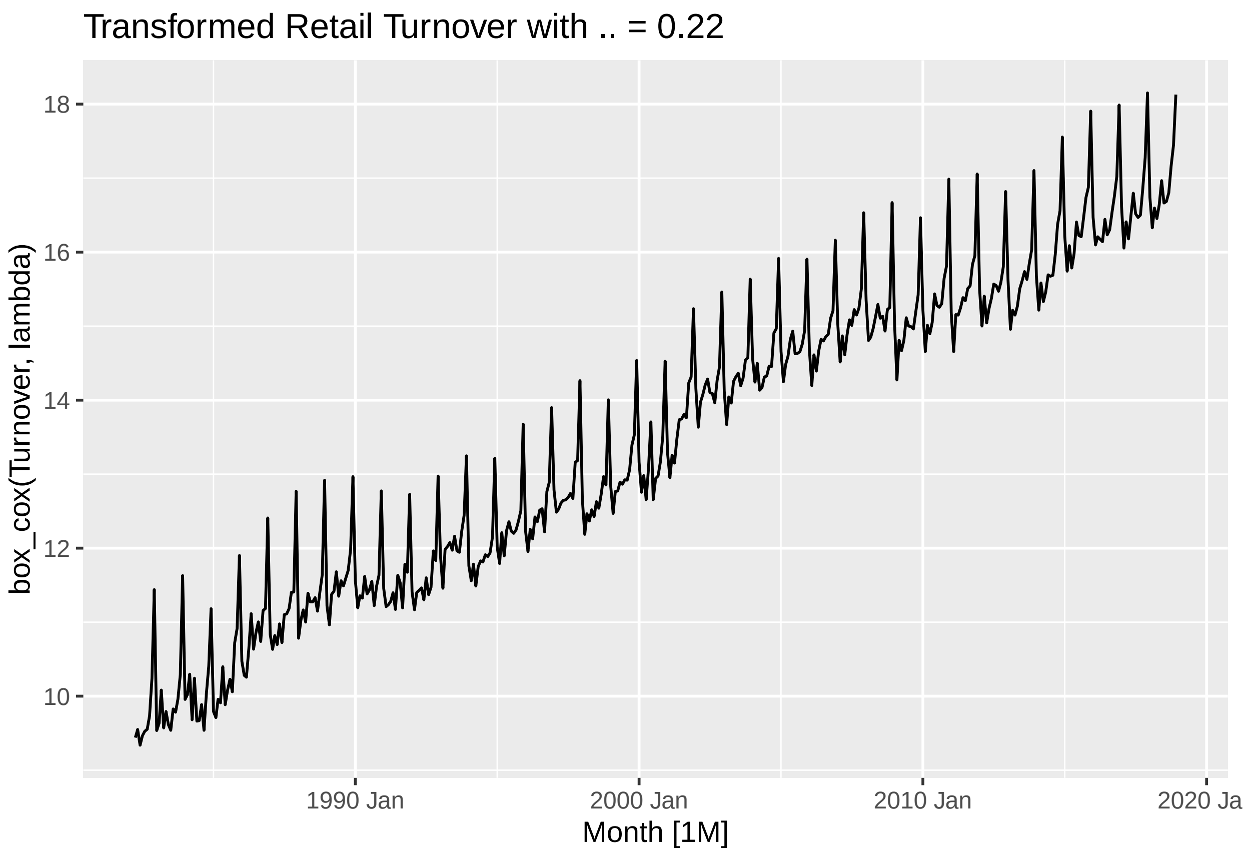
### 3.4. What Box-Cox transformation would you select for your retail data (from Exercise 7 in Section 2.10)?

```
set.seed(123)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
myseries %>% autoplot(Turnover) +
  labs(title = "Retail Data Turnover",
       y = "$AUD (Millions)")
```



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

myseries %>% autoplot(box_cox(Turnover, lambda))+
  labs(title = paste("Transformed Retail Turnover with \u03BB =", round(lambda, 2)))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Transformed Retail Turnover with   = 0.22' in
## 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Transformed Retail Turnover with   = 0.22' in
## 'mbcsToSbcs': dot substituted for <bb>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Transformed Retail Turnover with   = 0.22' in
## 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Transformed Retail Turnover with   = 0.22' in
## 'mbcsToSbcs': dot substituted for <bb>
```

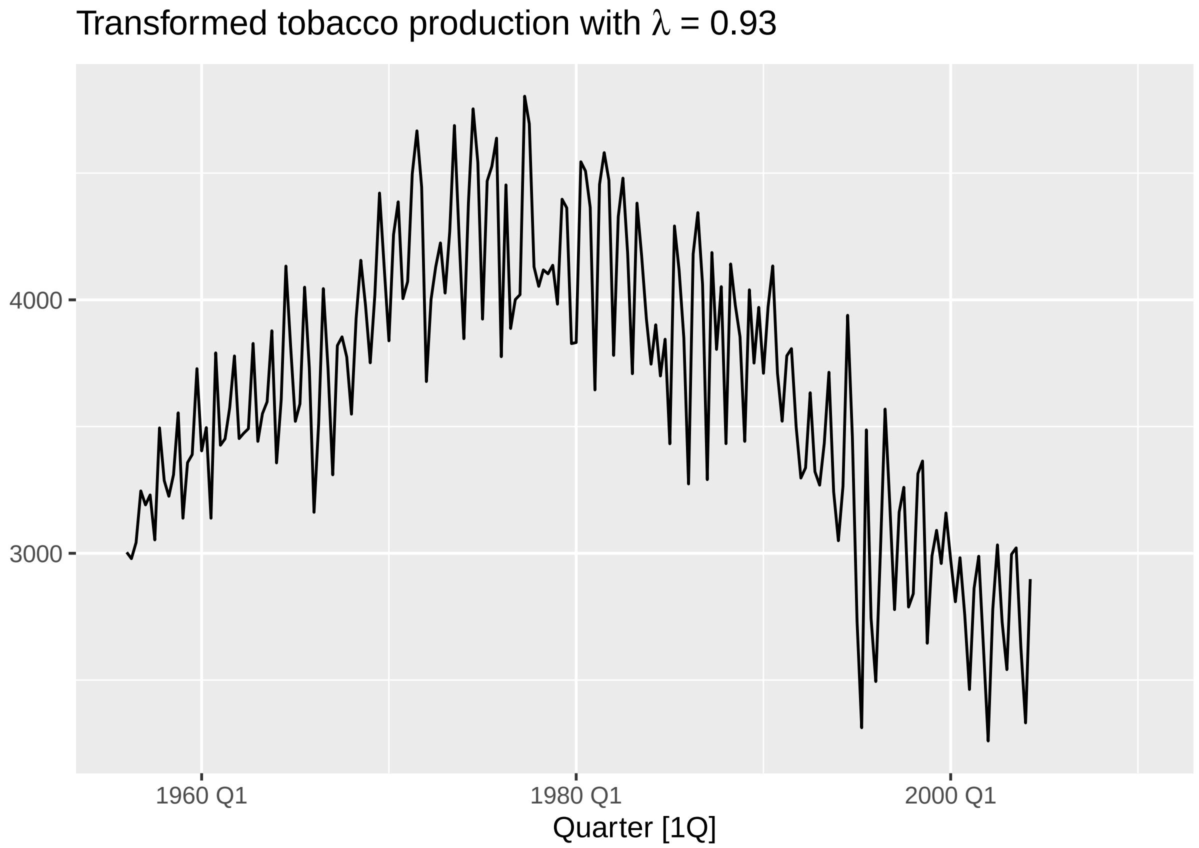


Applying the Box-Cox transformation with a lambda value of 0.22 has improved the consistency of seasonal patterns in the data. This transformation is beneficial for handling data with exponential growth by using a natural logarithm to stabilize variance and normalize the distribution. The lambda value of 0.22 was selected based on Guerrero's research, which identified it as effective for simplifying forecasting. This optimization makes the data more suitable for linear modeling and enhances prediction accuracy. Overall, the transformation has led to a more uniform seasonal variation, facilitating better forecasting and analysis.

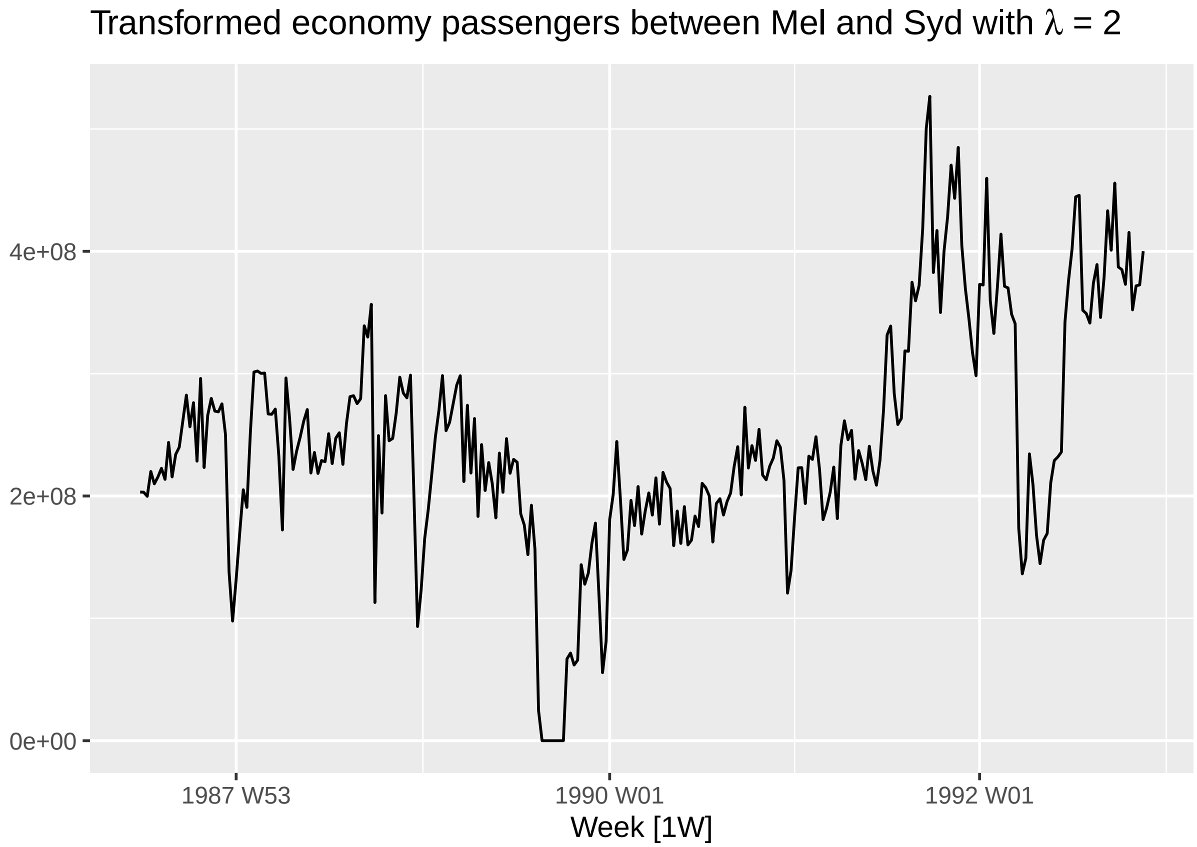
### 3.5. For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance. Tobacco from aus_production, Economy class passengers between Melbourne and Sydney from ansett, and Pedestrian counts at Southern Cross Station from pedestrian.

```
# Tobacco
lambda <- aus_production %>%
  features(Tobacco, features = guerrero) %>%
  pull(lambda_guerrero)
aus_production %>%
  autoplot(box_cox(Tobacco, lambda)) +
  labs(y = "",
       title = latex2exp::TeX(paste0(
         "Transformed tobacco production with $\\lambda$ = ",
         round(lambda,2))))
```

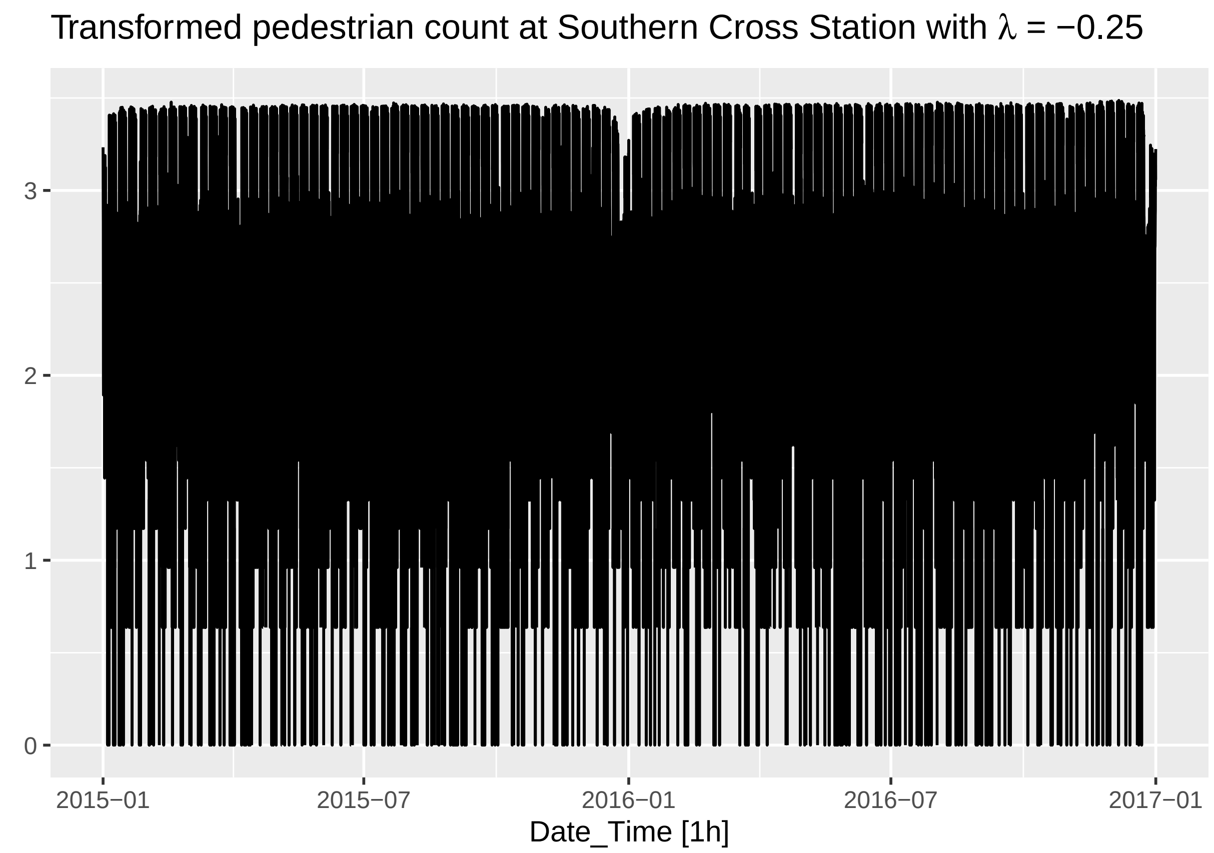
```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



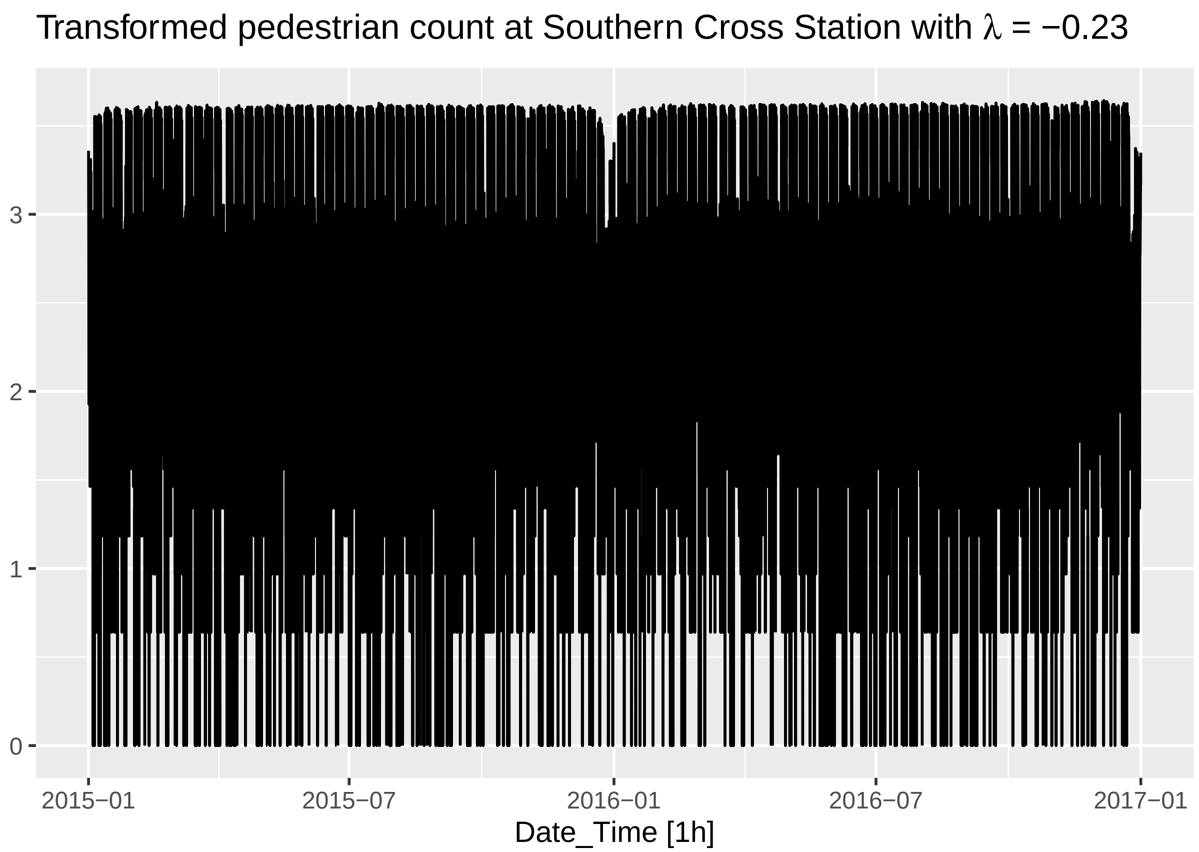
```
# Economy class passengers between Melbourne and Sydney
lambda <- ansett %>%
  filter(Airports == 'MEL-SYD' &
           Class == 'Economy') %>%
  features(Passengers, features = guerrero) %>%
  pull(lambda_guerrero)

ansett %>%
  filter(Airports == 'MEL-SYD' &
           Class == 'Economy') %>%
  autoplot(box_cox(Passengers, lambda)) +
  labs(y = "",
       title = latex2exp::TeX(paste0(
         "Transformed economy passengers between Mel and Syd with $\\lambda$ = ",
         round(lambda,2))))
```



```
# Pedestrian counts at Southern Cross Station
lambda <- pedestrian %>%
  filter(Sensor == 'Southern Cross Station') %>%
  features(Count, features = guerrero) %>%
  pull(lambda_guerrero)

pedestrian %>%
  filter(Sensor == 'Southern Cross Station') %>%
  autoplot(box_cox(Count, lambda)) +
  labs(y = "",
       title = latex2exp::TeX(paste0(
         "Transformed pedestrian count at Southern Cross Station with $\\lambda$ = ",
         round(lambda,2))))
```



```
lambda <- pedestrian %>%
  filter(Sensor == 'Southern Cross Station' &
           Date < '2015-12-01') %>%
  features(Count, features = guerrero) %>%
  pull(lambda_guerrero)

pedestrian %>%
  filter(Sensor == 'Southern Cross Station') %>%
  autoplot(box_cox(Count, lambda)) +
  labs(y = "",
       title = latex2exp::TeX(paste0(
         "Transformed pedestrian count at Southern Cross Station with $\\lambda$ = ",
         round(lambda,2))))
```



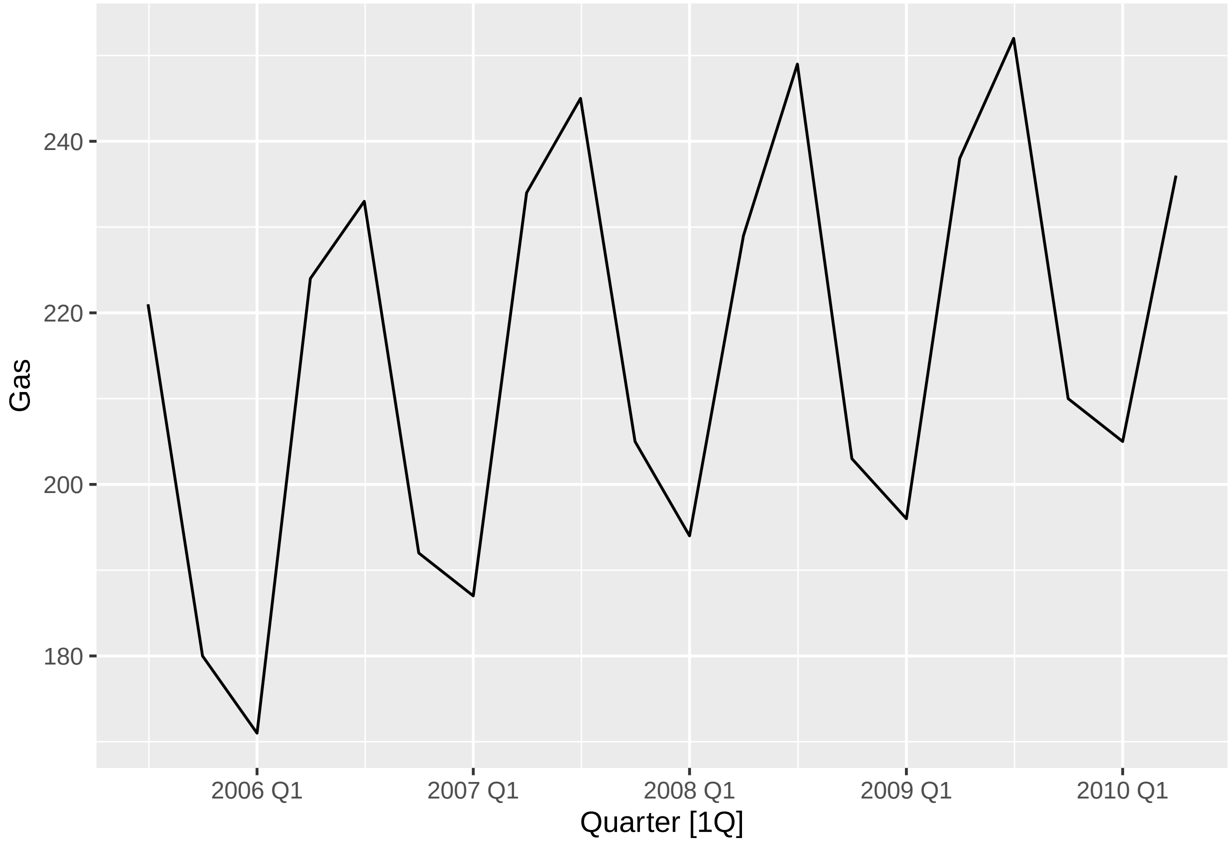
### 3.7. Consider the last five years of the Gas data from aus_production.

```
# Extract the last 5 years of quarterly gas production data from 'aus_production'
gas <- tail(aus_production, n = 5 * 4) |> select(Gas)

# Display the first few rows of the extracted gas data
head(gas)
```

```
## # A tsibble: 6 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
```

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

```
gas %>%  autoplot(Gas)
```

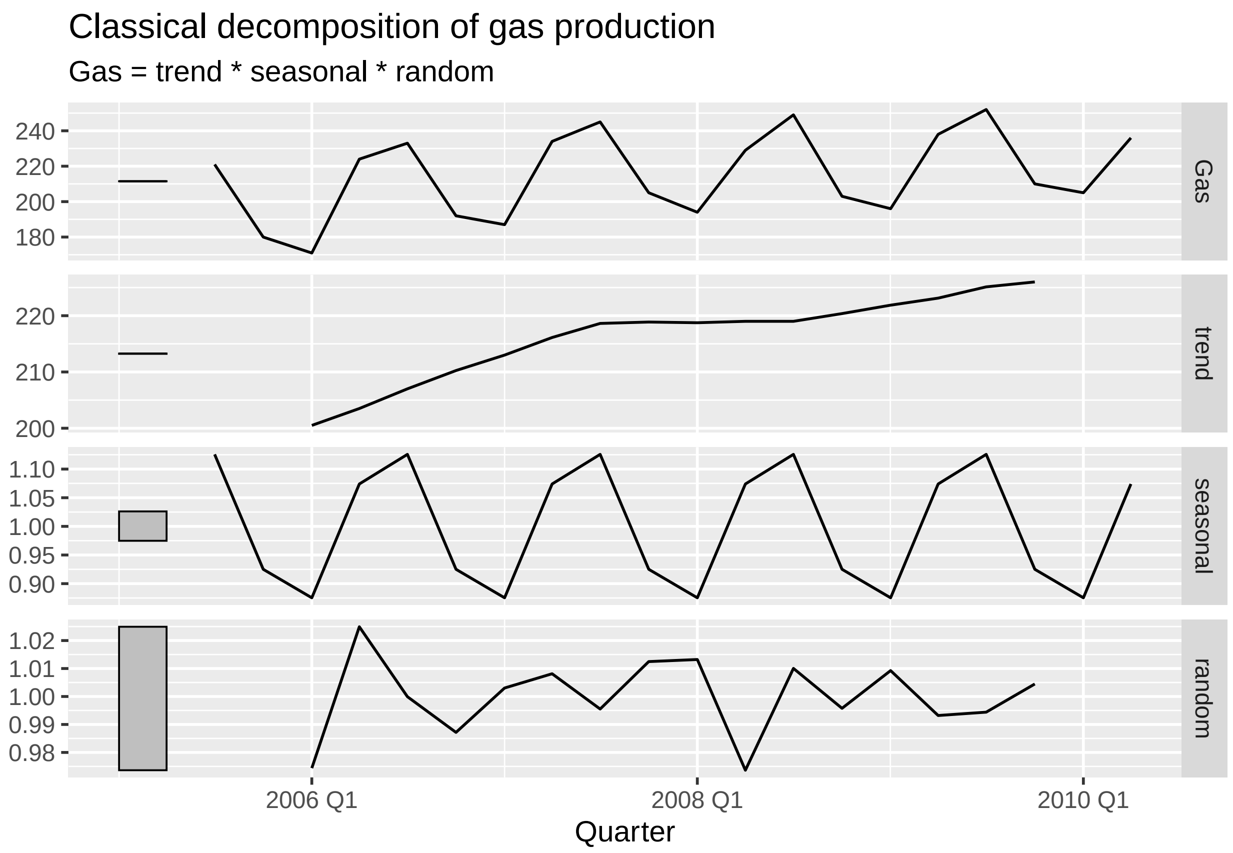


The data shows a clear seasonal trend: manufacturing production usually declines to its lowest levels by the end of the fourth quarter. In contrast, production often peaks around the middle of the year, typically near the end of the second quarter.

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

```
# From section 3.4
clas_decomp <- gas %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components()

clas_decomp %>%
  autoplot() +
  labs(title = "Classical decomposition of gas production")
```

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



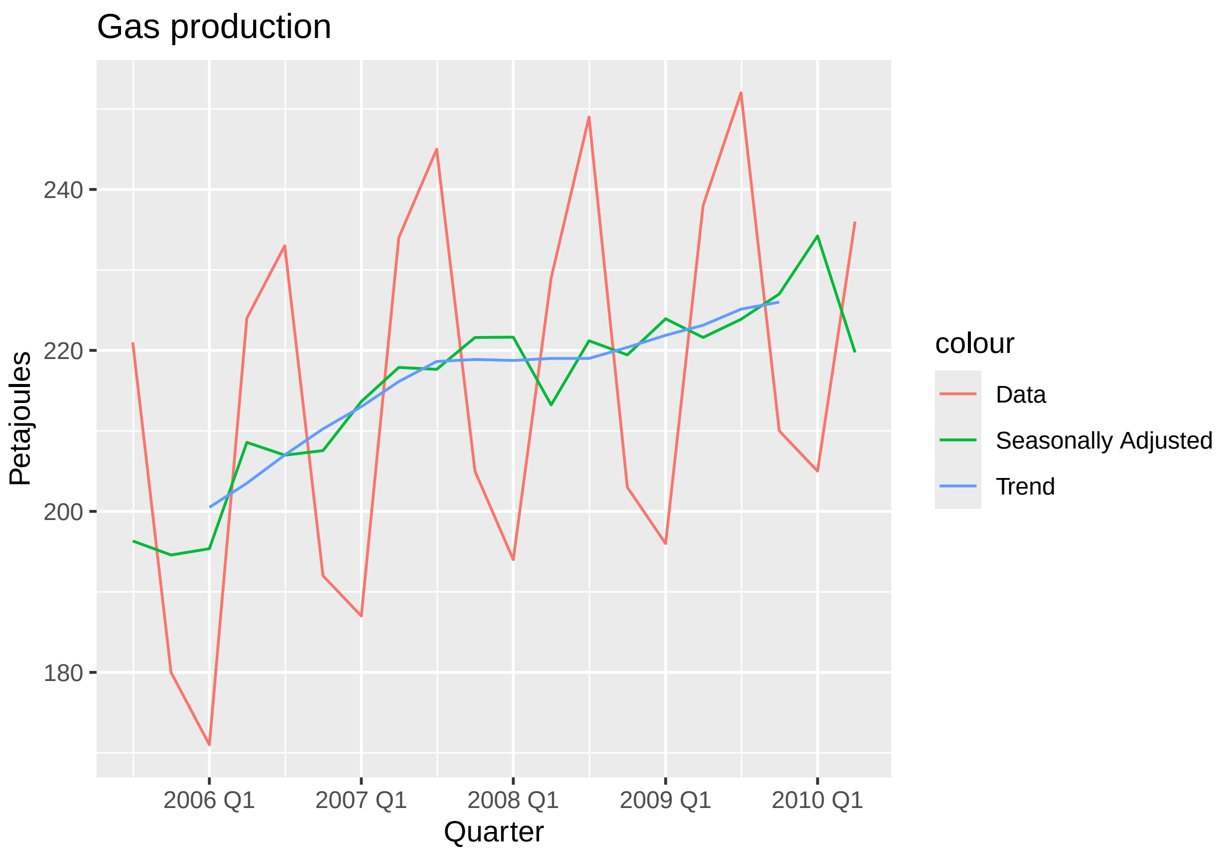
**Do the results support the graphical interpretation from part a?**

Yes, the trend line demonstrates a steady upward progression from left to right, with a stable phase in the middle. Additionally, the seasonal indices reflect a nearly flawless seasonal variation across the five-year span.

**Compute and plot the seasonally adjusted data.**

```
clas_decomp %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Gas, colour = "Data")) +
  geom_line(aes(y = season_adjust,
                colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  labs(y = "Petajoules",
       title = "Gas production")
```

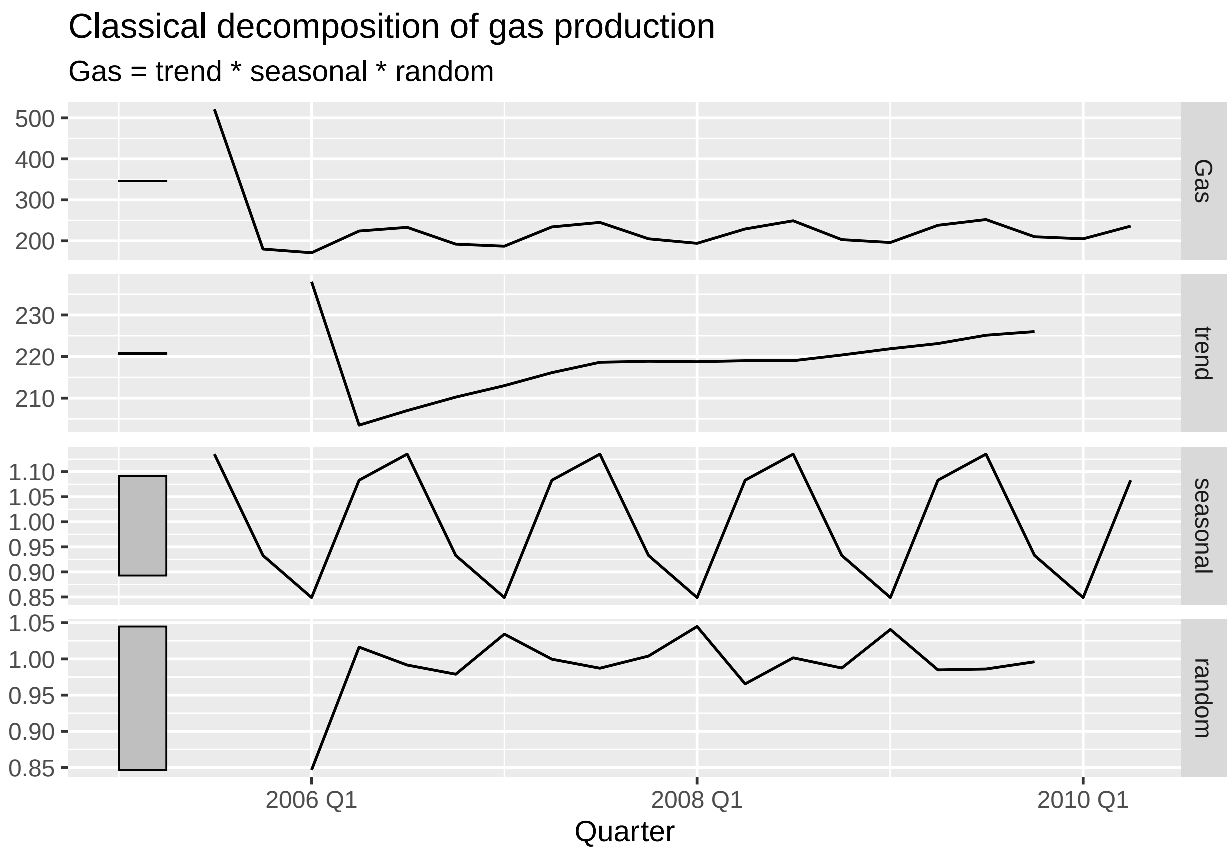
```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



**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?**

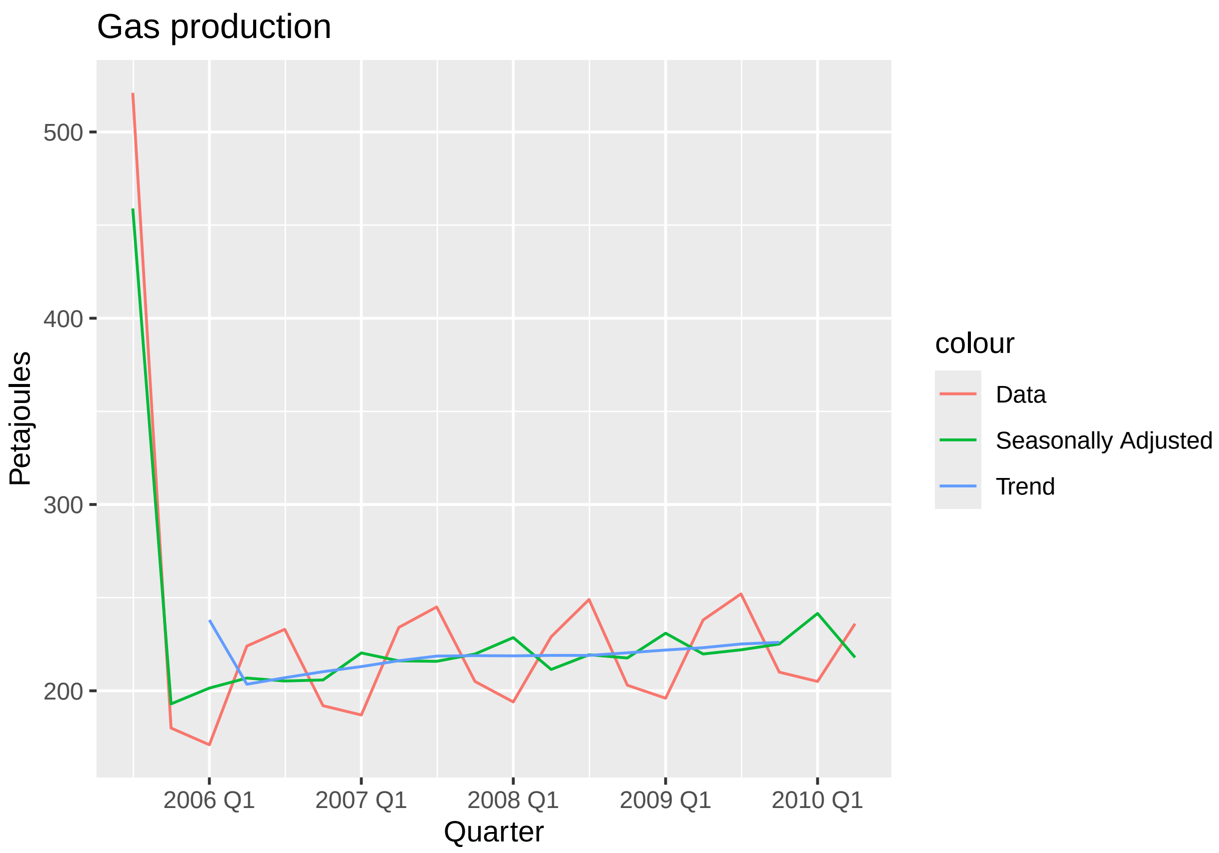
```r
# Outlier in beginning

gas_OutFront <- gas
gas_OutFront$Gas[1] <- gas_OutFront$Gas[1] + 300

of_clas_decomp <- gas_OutFront %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components()

of_clas_decomp %>%
  autoplot() +
  labs(title = "Classical decomposition of gas production")
```

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



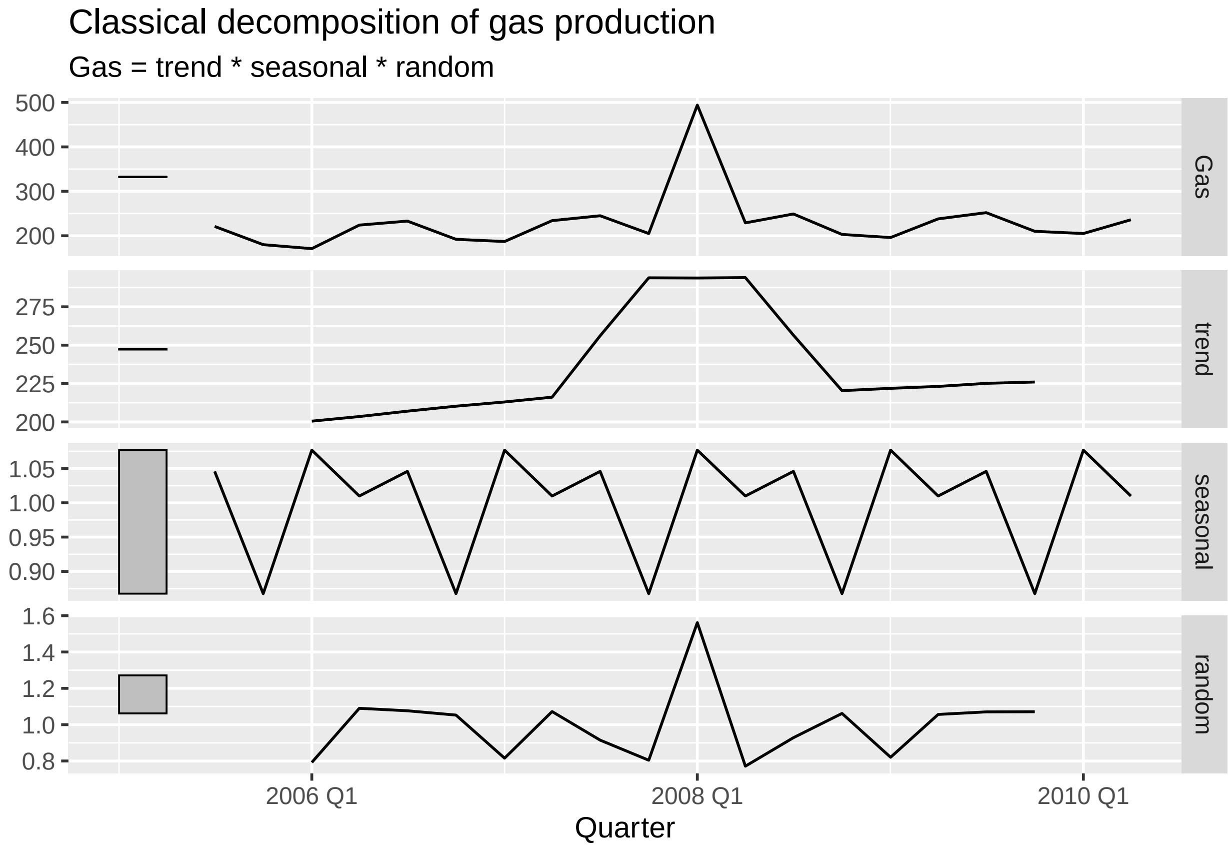
```
of_clas_decomp %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Gas, colour = "Data")) +
  geom_line(aes(y = season_adjust,
               colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  labs(y = "Petajoules",
       title = "Gas production")
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



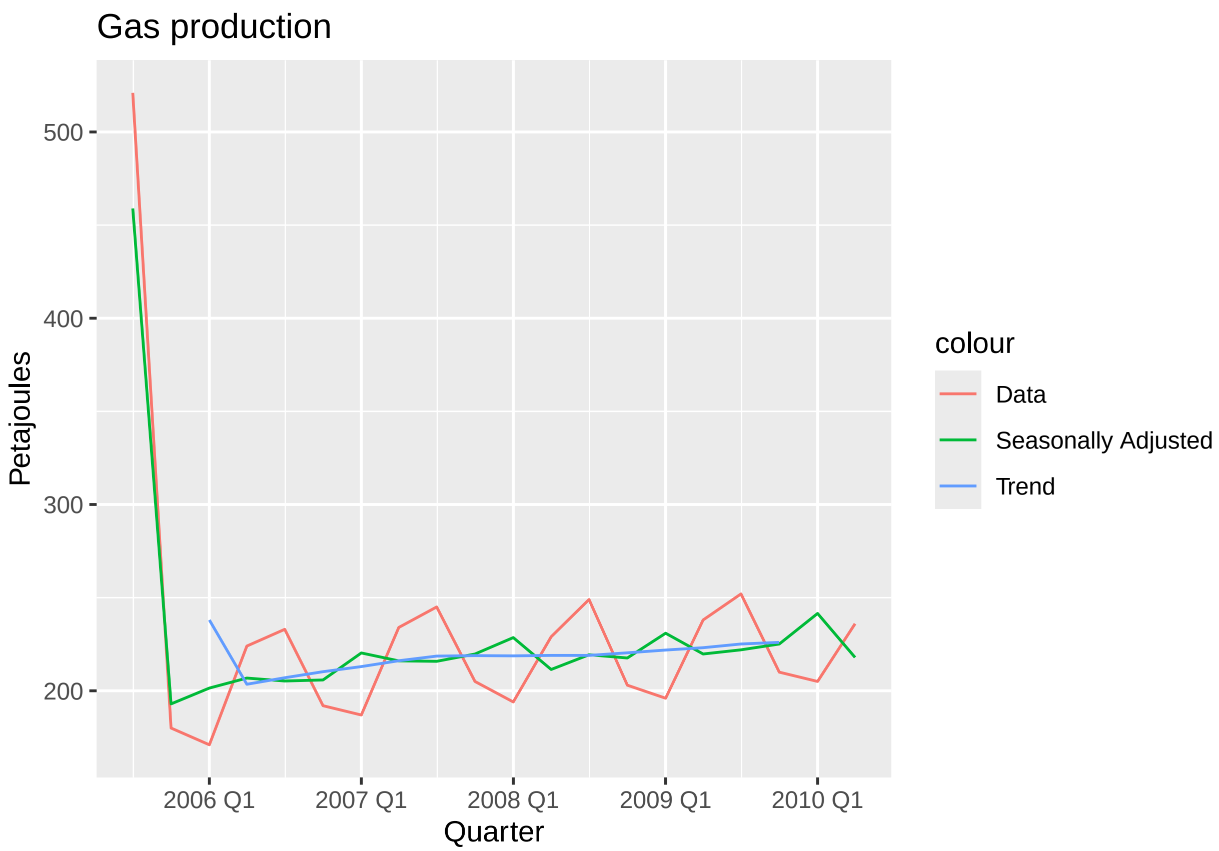
```
# Outlier in middle
gas_OutMid <- gas
gas_OutMid$Gas[11] <- gas_OutMid$Gas[11] + 300

om_clas_decomp <- gas_OutMid %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components()

om_clas_decomp %>%
  autoplot() +
  labs(title = "Classical decomposition of gas production")
```

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



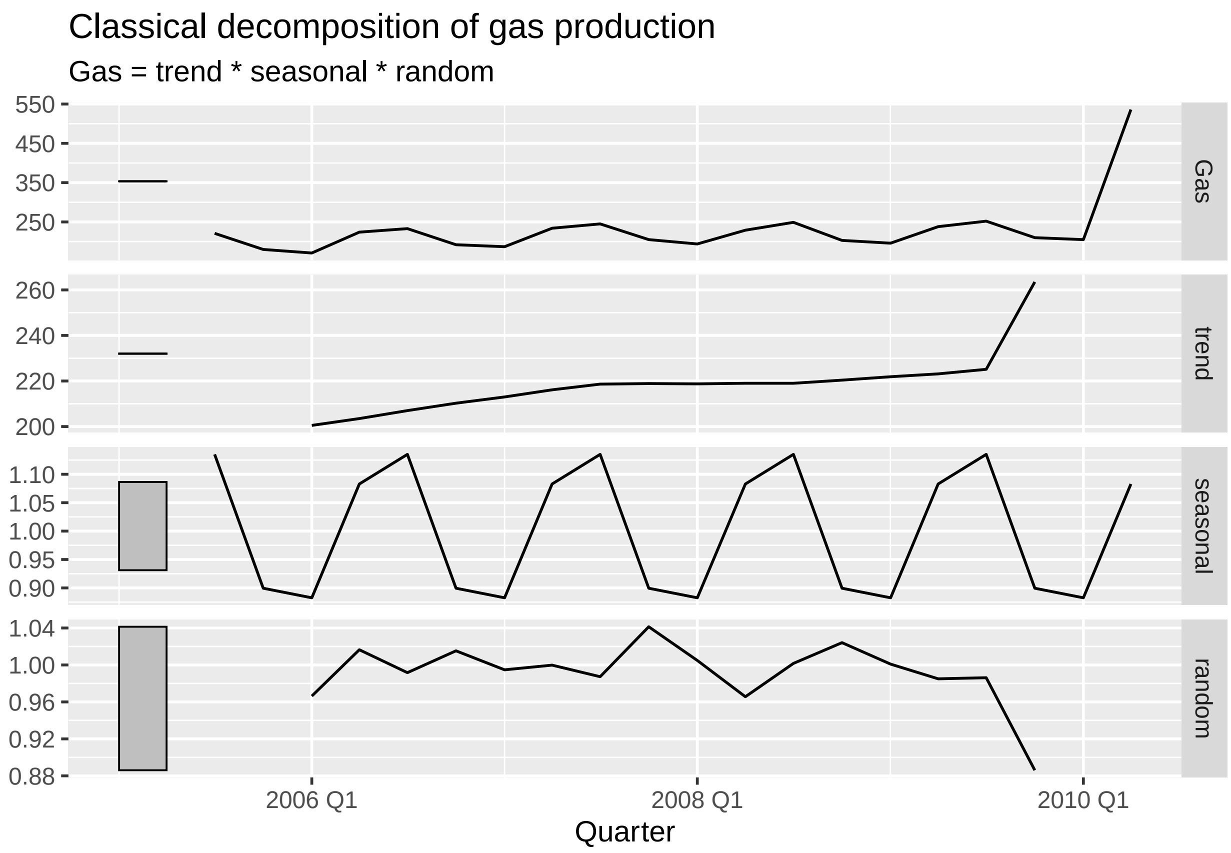
```
of_clas_decomp %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Gas, colour = "Data")) +
  geom_line(aes(y = season_adjust,
                colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  labs(y = "Petajoules",
       title = "Gas production")
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



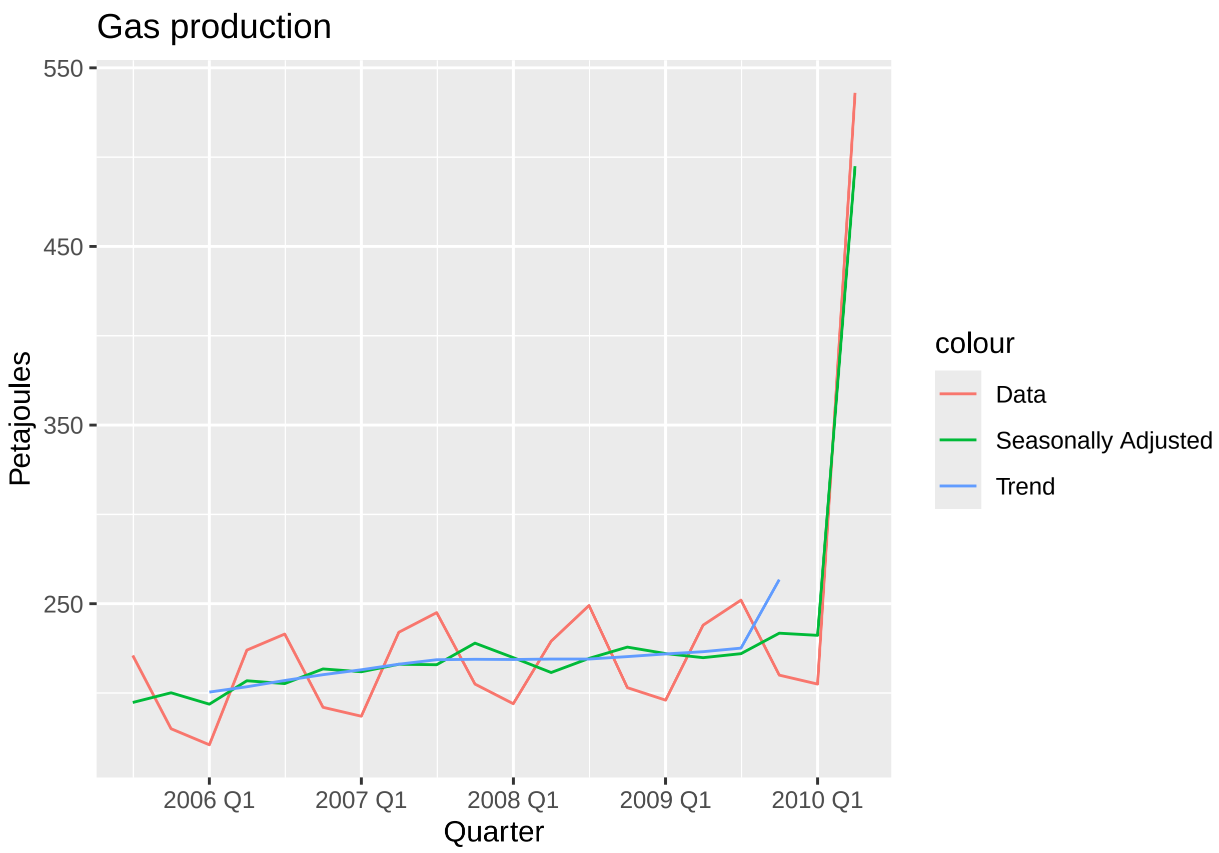
```
# Outlier in back
gas_OutBack <- gas
gas_OutBack$Gas[20] <- gas_OutBack$Gas[20] + 300

ob_clas_decomp <- gas_OutBack %>%
  model(classical_decomposition(Gas, type = "multiplicative")) %>%
  components()

ob_clas_decomp %>%
  autoplot() +
  labs(title = "Classical decomposition of gas production")
```

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



```
ob_clas_decomp %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Gas, colour = "Data")) +
  geom_line(aes(y = season_adjust,
                colour = "Seasonally Adjusted")) +
  geom_line(aes(y = trend, colour = "Trend")) +
  labs(y = "Petajoules",
       title = "Gas production")
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



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

Examining the plots, the first outlier at the beginning highlights a clear anomaly in the seasonally adjusted data, though the rest of the adjusted plot matches the original data well. The outlier at the end also shows an anomaly but otherwise, the seasonally adjusted plot remains close to the original. For the outlier in the middle, the seasonally adjusted plot tracks the data's seasonal patterns more accurately. However, this middle outlier suggests that the seasonally adjusted data might not fully capture the underlying trends.

### 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?

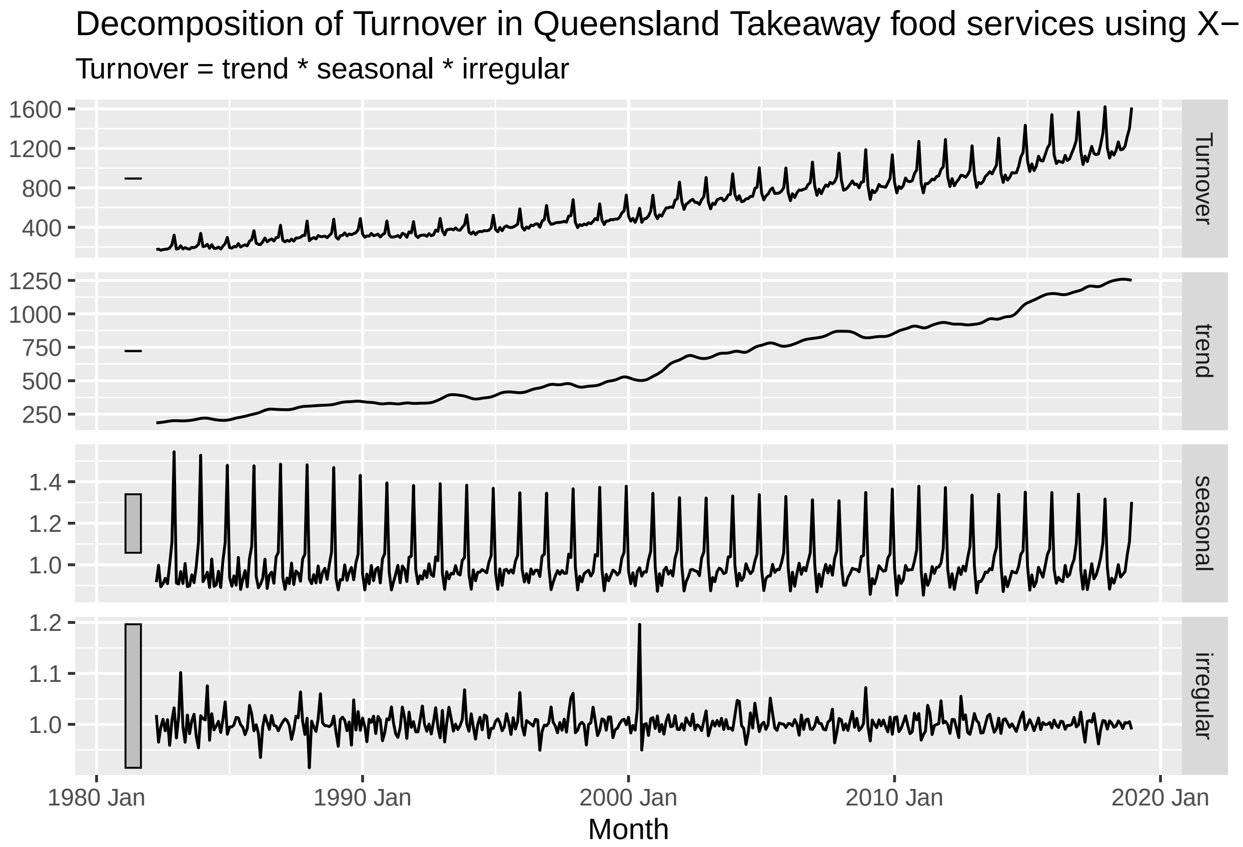
```r
library(seasonal)
```

```
## Warning: package 'seasonal' was built under R version 4.3.3
```

```
##
## Attaching package: 'seasonal'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```r
set.seed(123)

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 =
    "Decomposition of Turnover in Queensland Takeaway food services using X-11.")
```



The seasonal variance changes over time, as shown in the chart. Early on, there are spikes in higher turnover, but later, the spikes occur with lower turnover values. The plot also reveals a few outliers, as noted in the "irregular" chart. Overall, the trend plot doesn't reveal any unexpected patterns in the data.

### 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.

**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 results show that the trend line accurately reflects the overall data pattern. Interestingly, the scales of the season_year and remainder components reveal some key insights. Typically, I would expect the season_year chart to have a smaller gray bar, suggesting it plays a more significant role in the data. However, the decomposition indicates that the remainder component has a greater impact, likely due to the recession of 1991/92, which significantly affected the data. Additionally, the sub-seasonal chart shows notable fluctuations in turnover during March, July, August, November, and December.

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

Yes, the recession of 1991/1992 is clearly visible in the estimated components. The remainder component, in particular, highlights significant outliers during these years, which have a larger impact compared to the season_year component, as reflected in the scales. While the overall data and trend show a slight decline, the remainder chart reveals a more dramatic drop during this period, underscoring the recession's effect.