DATA 624 - Homework 2

Richie Rivera

Question 3.1

```r
# Importing the library
library("fpp3")
```

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?

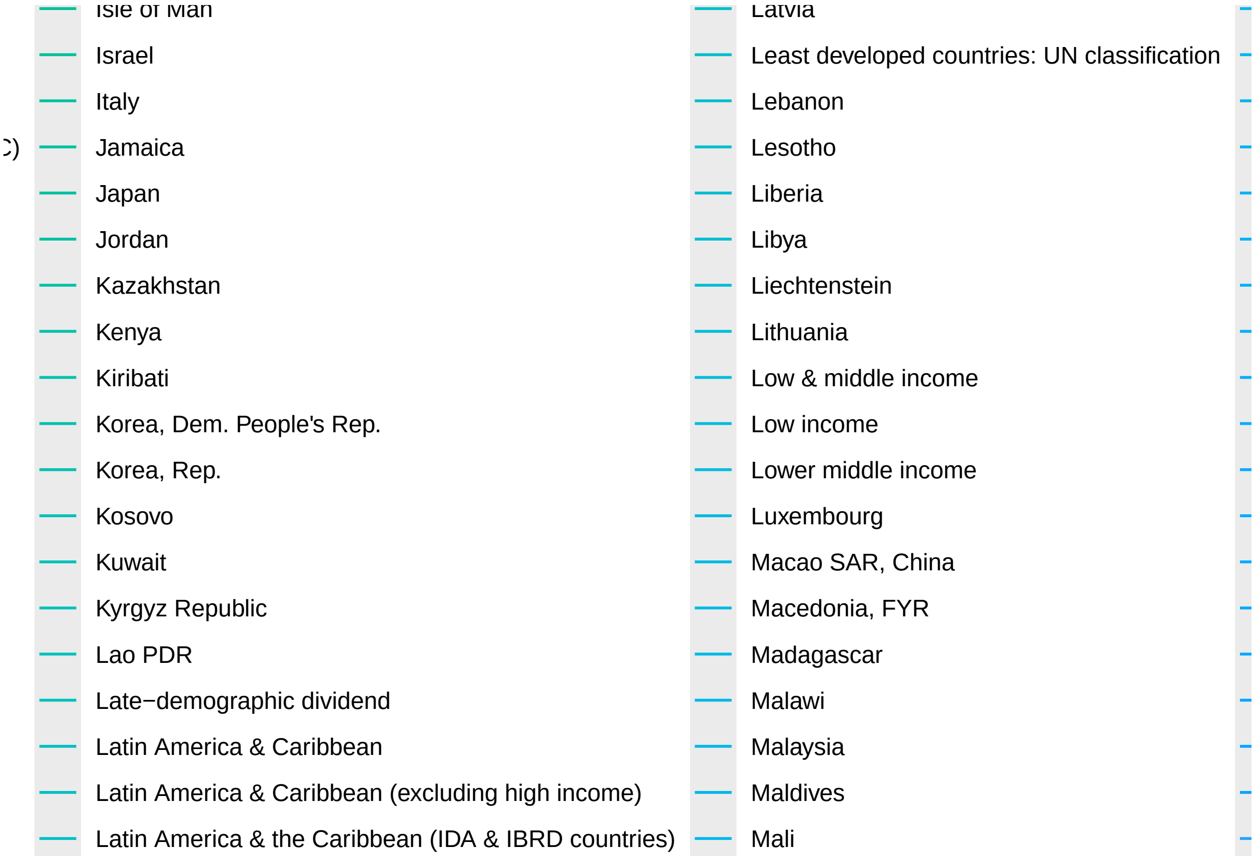
```
## Warning: package 'fpp3' was built under R version 4.3.3

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

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

## v tibble      3.2.1     v tsibble     1.1.5
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.0     v feasts      0.3.2
## v lubridate   1.9.3     v fable       0.3.4
## v ggplot2     3.5.1     v fabletools  0.4.2

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tsibble' was built under R version 4.3.3

## Warning: package 'tsibbledata' was built under R version 4.3.3

## Warning: package 'feasts' was built under R version 4.3.3

## Warning: package 'fabletools' was built under R version 4.3.3

## Warning: package 'fable' was built under R version 4.3.3

## -- 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()
```

```r
# Loading in the dataset
data(global_economy)

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
global_economy <- global_economy |>
  mutate(
    GDPpC = GDP / Population
  )

autoplot(
  global_economy,
  GDPpC
)
```

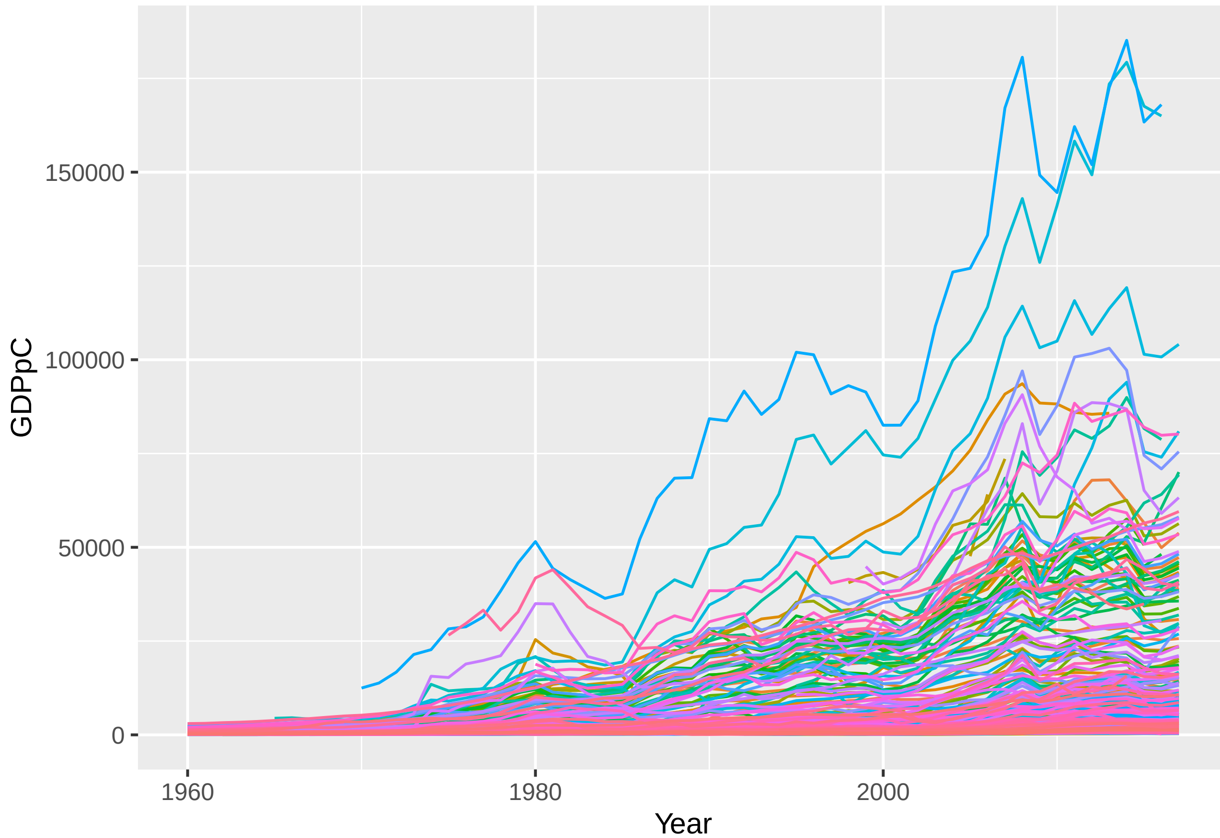
```
## Warning: Removed 3242 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



This plot has so many timeseries in it that the legend takes up the whole plot. The easiest way I found to get around this is to remove the legend:

```
autoplot(
  global_economy,
  GDPpC,
  show.legend = FALSE
) +
  xlab("Year") +
  ylab("GDPpC")
```

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



From this chart, we can see that there is one line that historically is much greater than the rest. We'll filter the dataset to 1994 where the line is clearly the country with the greatest GDP per capita. Towards the end of the timeseries there is a change and it becomes a bit hard to tell which one is on top:

```
global_economy |>
  filter(
    Year == 1994
  ) |>
  arrange(
    desc(GDPpC)
  ) |>
  head()
```

```
## # A tsibble: 6 x 10 [1Y]
## # Key:       Country [6]
##   Country     Code   Year      GDP Growth   CPI Imports Exports Population  GDPpC
##   <fct>       <fct> <dbl>    <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>  <dbl>
## 1 Monaco      MCO    1994 2.72e 9  2.22    NA      NA      NA          30427 89404.
## 2 Liechtenst~ LIE    1994 1.95e 9  6.87    NA      NA      NA          30365 64157.
## 3 Luxembourg  LUX    1994 1.83e10  3.82    72.8    80.8   100.        402925 45482.
## 4 Switzerland CHE    1994 2.93e11  1.27    86.9    36.2    40.7      6993795 41844.
## 5 Japan       JPN    1994 4.91e12  0.993 101.       7.10    9.00   124961000 39269.
## 6 Bermuda     BMU    1994 1.87e 9  0.600   NA      NA      NA          59320 31476.
```

We can see from the first few rows that the highest GDP per Capita is typically Monaco. From the most recent year:

```r
global_economy |>
  filter(
    Year == max(global_economy$Year)
  ) |>
  arrange(
    desc(GDPpC)
  ) |>
  head()
```

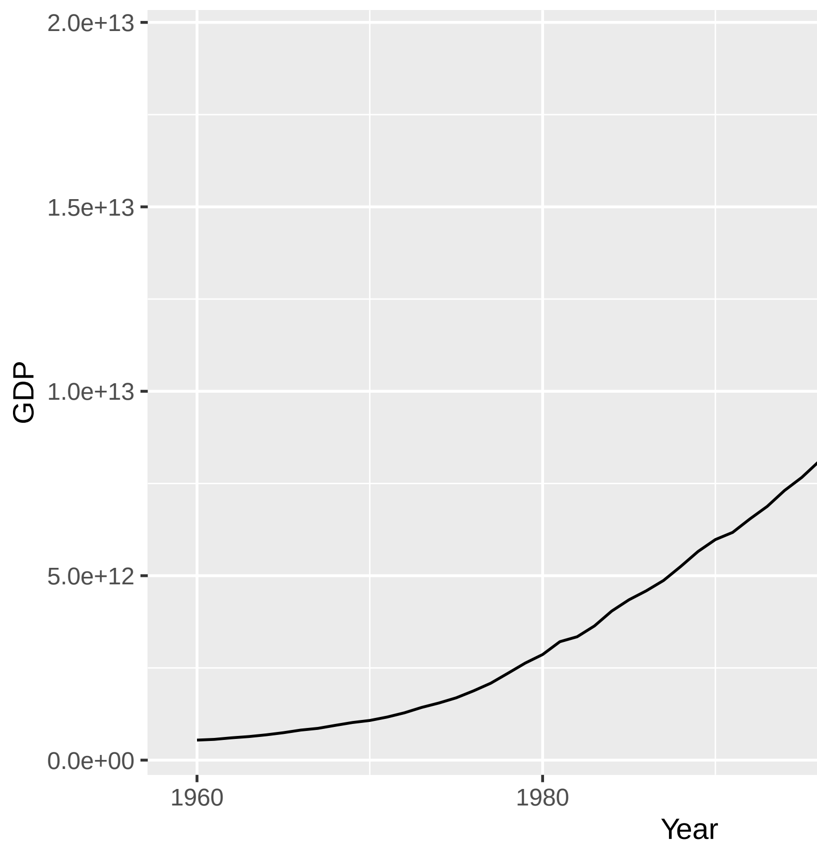
```
## # A tsibble: 6 x 10 [1Y]
## # Key:       Country [6]
##   Country     Code   Year     GDP Growth   CPI Imports Exports Population  GDPpC
##   <fct>       <fct> <dbl>   <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>  <dbl>
## 1 Luxembourg  LUX    2017 6.24e10   2.30 111.    194.    230.      599449 1.04e5
## 2 Macao SAR,~ MAC    2017 5.04e10   9.10 136.     32.0    79.4     622567 8.09e4
## 3 Switzerland CHE    2017 6.79e11   1.09  98.3    53.9    65.0    8466017 8.02e4
## 4 Norway      NOR    2017 3.99e11   1.92 115.     33.1    35.5    5282223 7.55e4
## 5 Iceland     ISL    2017 2.39e10   3.64 122.     42.8    47.0     341284 7.01e4
## 6 Ireland     IRL    2017 3.34e11   7.80 105.     87.9   120.     4813608 6.93e4
```

In the latest year of data, it's Luxemburg with a GDP per capita quite a bit higher than Monaco's.

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.

```r
autoplot(
  global_economy |>
    filter(
      Code == "USA"
    ),
  GDP
) +
  xlab("Year") +
  ylab("GDP")
```



A. United States GDP from global_economy.

For the US GDP, we can see that the growth seems to be exponential with a single exception that seems to be around 2008.

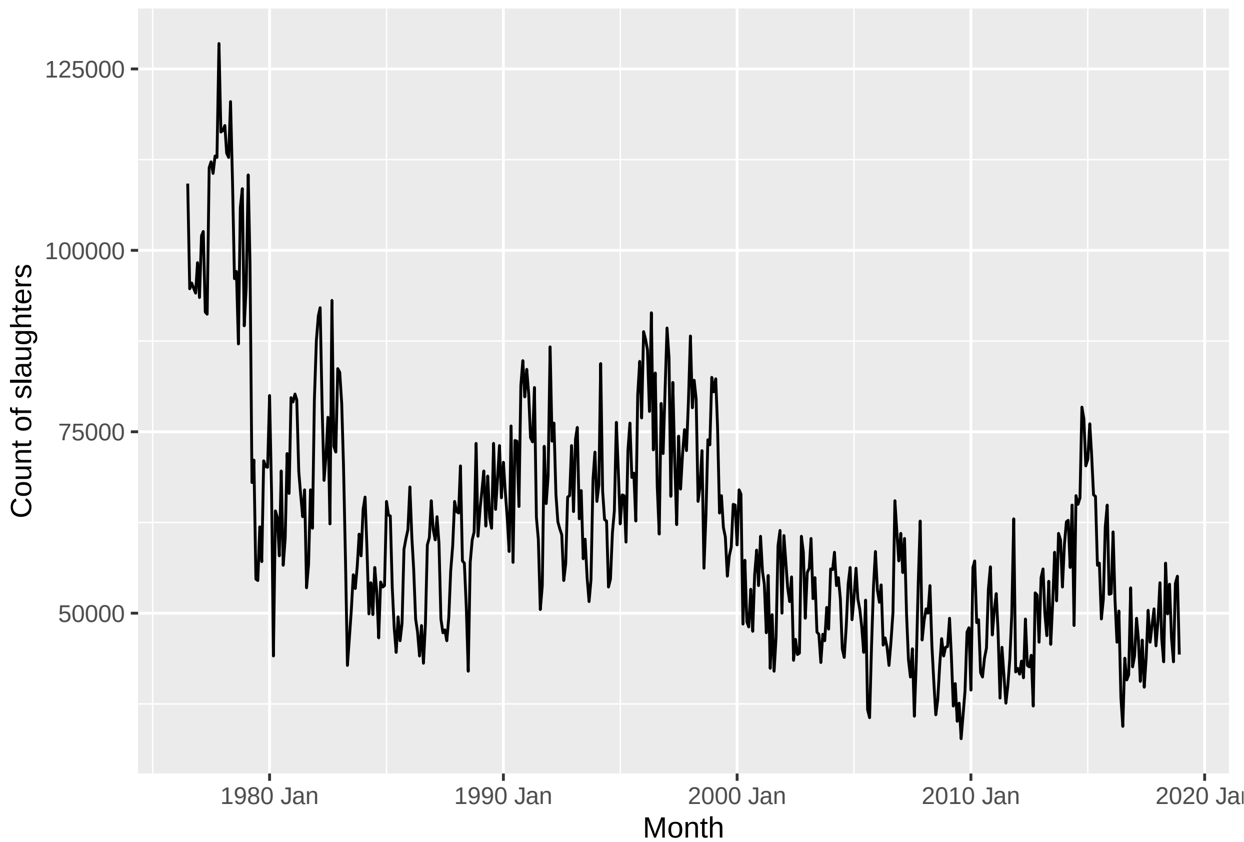
```r
data(aus_livestock)
head(aus_livestock)
```

B. Slaughter of Victorian "Bulls, bullocks and steers" in 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
## 5 1976 Nov Bulls, bullocks and steers Australian Capital Territory  2100
## 6 1976 Dec Bulls, bullocks and steers Australian Capital Territory  1800
```

```r
autoplot(
  aus_livestock |>
    filter(
      Animal == "Bulls, bullocks and steers",
```

```
      State == "Victoria"
    ),
  Count
) +
  xlab("Month") +
  ylab("Count of slaughters")
```



This graph has what seems to be seasonal peaks and valleys but we can see that the trend is generally downward.

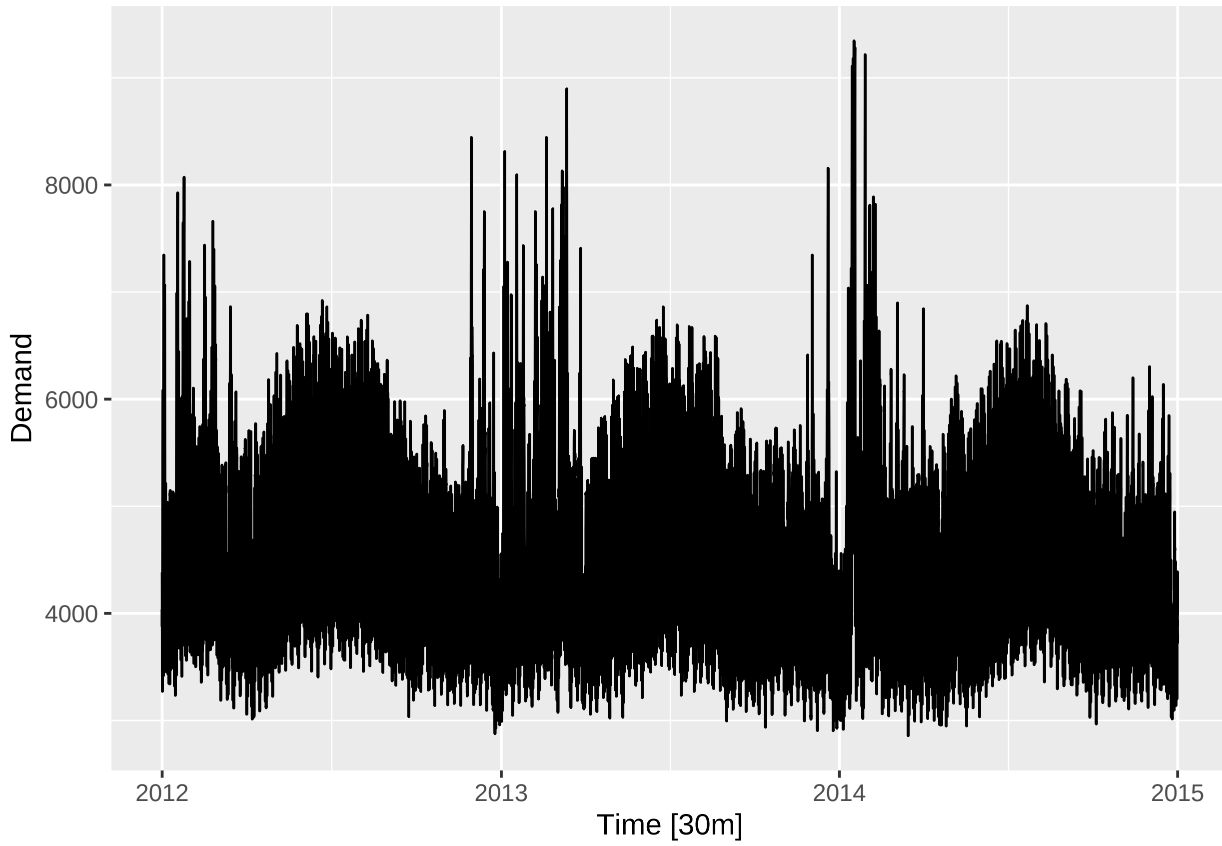
```
data(vic_elec)
head(vic_elec)
```

C. Victorian Electricity Demand from 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
```

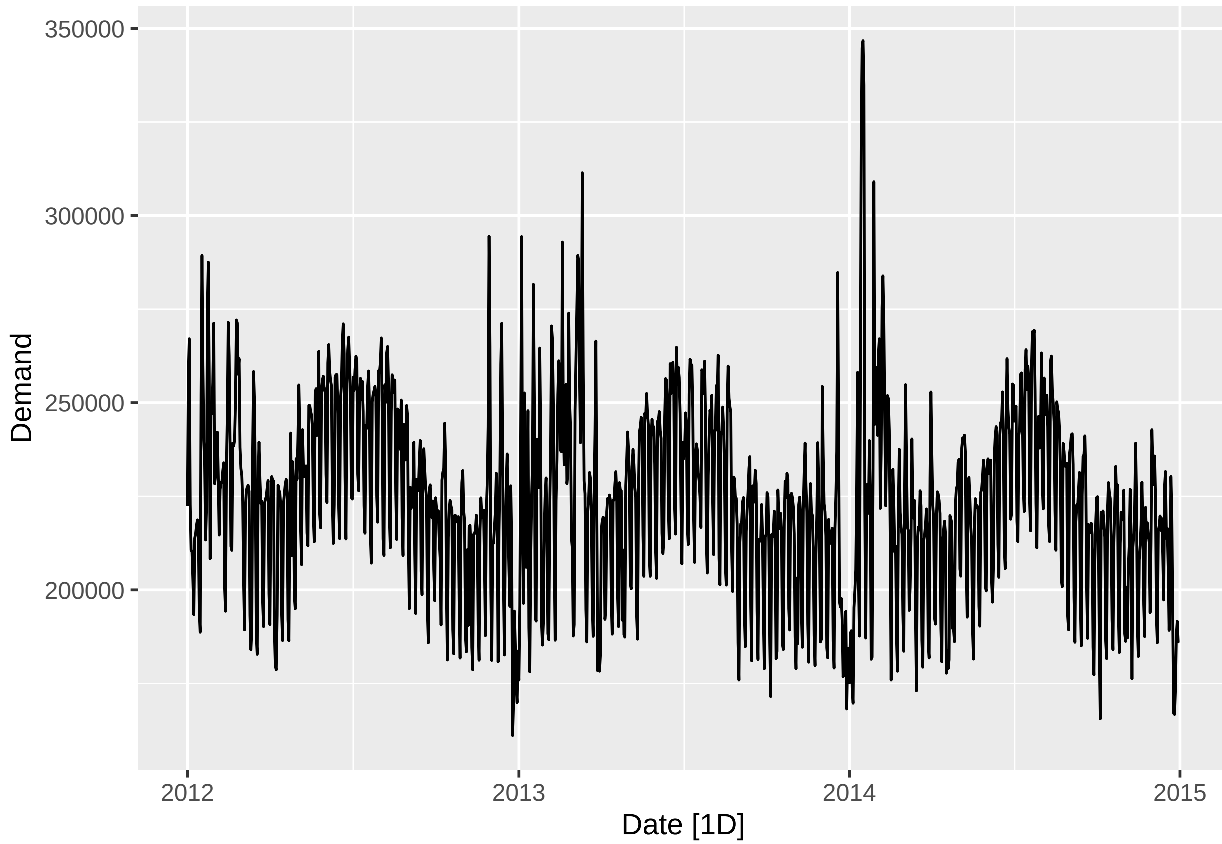
```r
autoplot(
  vic_elec,
  Demand
)
```



This graph has so much granularity on the x axis that we'll need to modify it to make it more presentable.

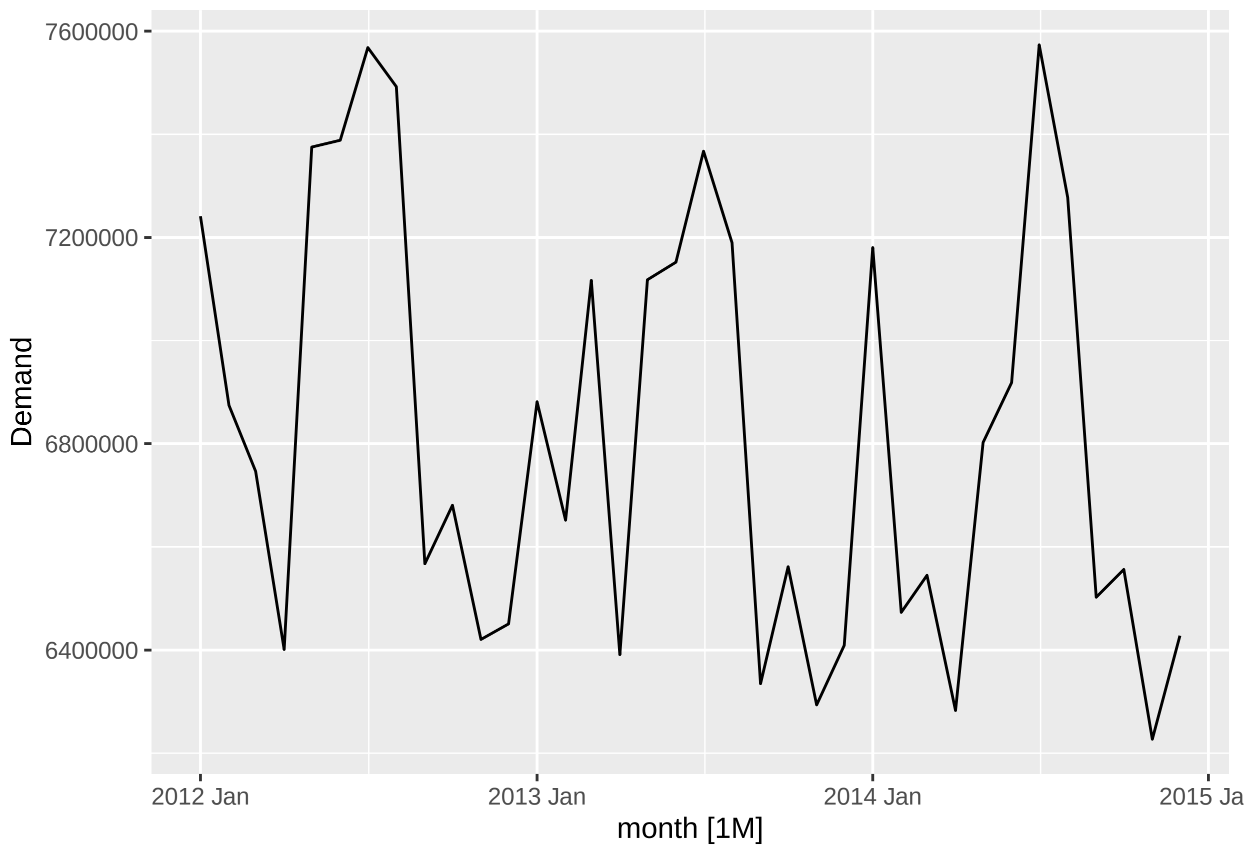
```r
daily_demand <- vic_elec |>
  group_by(Date) |>
  mutate(
    Demand = sum(Demand)
  ) |>
  distinct(
    Date,
    Demand
  ) |>
  as_tsibble(
    index = Date
  )

autoplot(
  daily_demand,
  Demand
```

```
)
```



This plot is a bit better as we can see that it's highly seasonal data. Trying one last group with month:

```
monthly_demand <- vic_elec |>
  mutate(
    month = yearmonth(Date)
  ) |>
  group_by(month) |>
  mutate(
    Demand = sum(Demand)
  ) |>
  distinct(
    month,
    Demand
  ) |>
  as_tsibble(
    index = month
  )

autoplot(
  monthly_demand,
  Demand
)
```



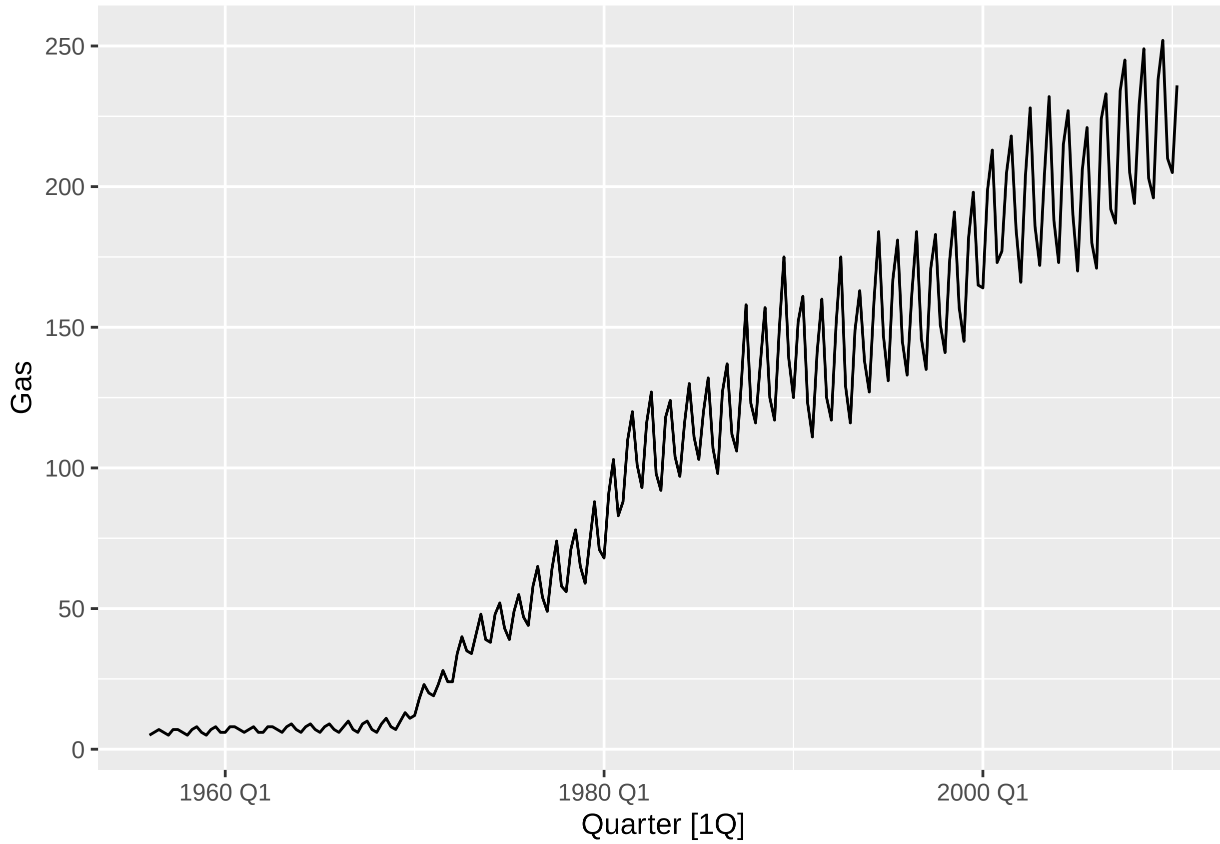
Looking at it monthly we can see some clear drops in demand and a general drop in demand over time.

```r
data(aus_production)
head(aus_production)
```

D. Gas production from 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
```

```r
autoplot(
  aus_production,
  Gas
)
```



From here we can see that there is some seasonality to gas production which seems to increase in range as time goes on but the general trend is increasing as well.

Question 3.3

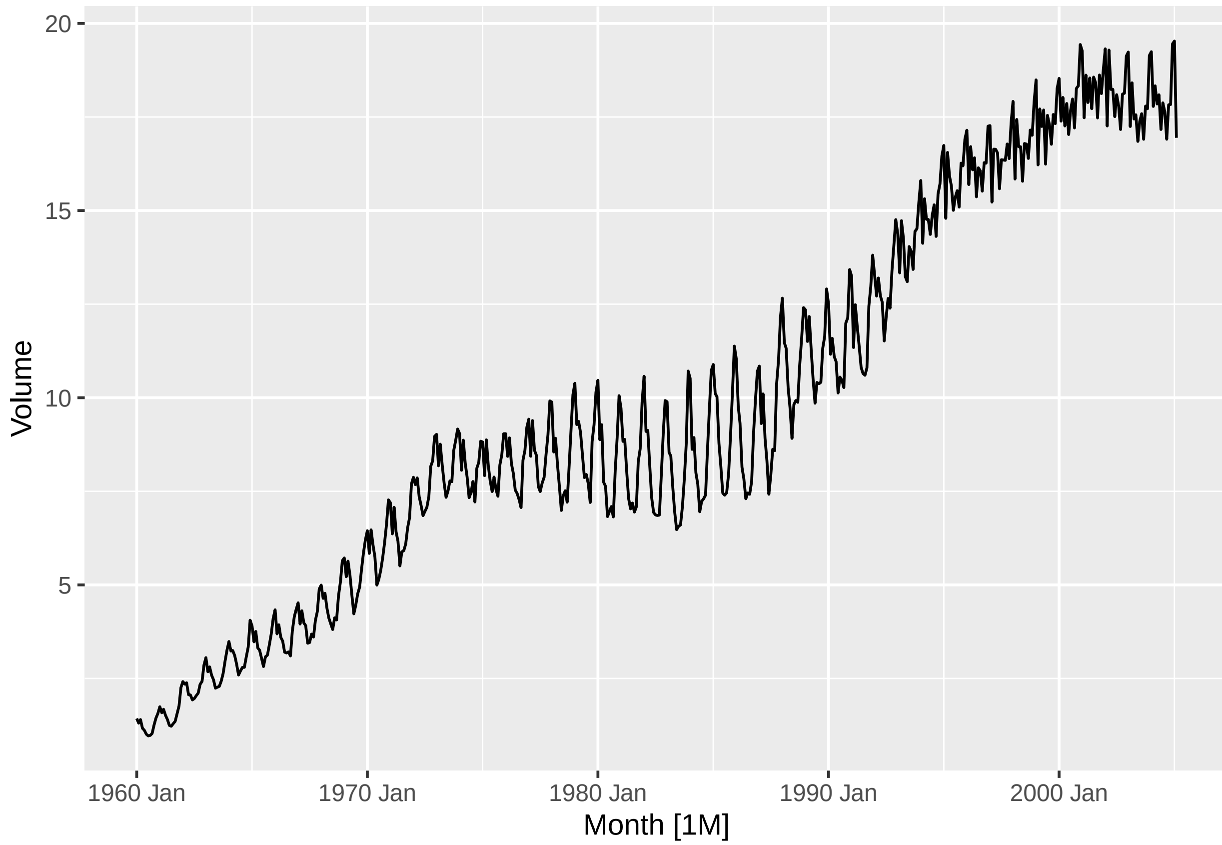
Why is a Box-Cox transformation unhelpful for the canadian_gas data?

We'll start by plotting the data as is and then plot it with a Box-Cox transformation. From the chapter, we can use the guerrero feature to select a lambda:

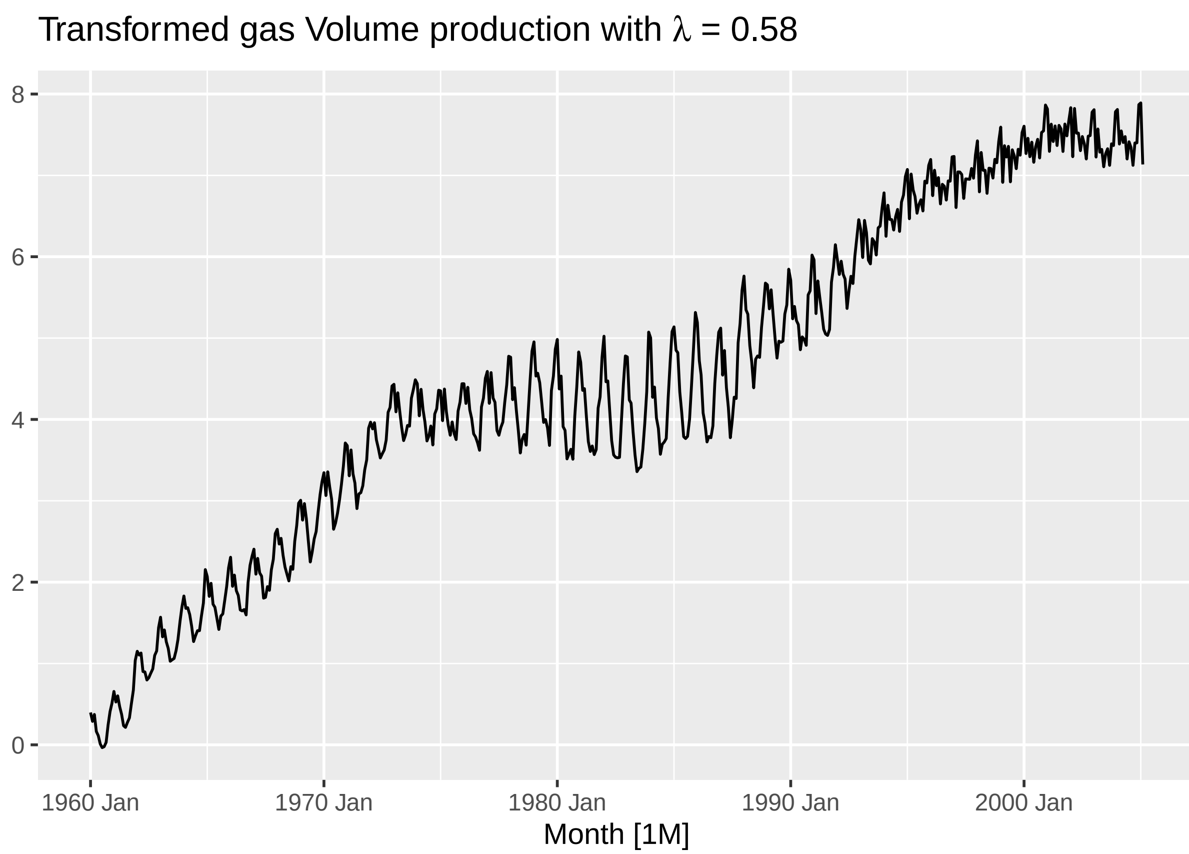
```r
data(canadian_gas)
head(canadian_gas)
```

```
## # 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
```

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




```r
lambda <- canadian_gas |>
  features(Volume, features = guerrero) |>
  pull(lambda_guerrero)

canadian_gas |>
  autoplot(box_cox(Volume, lambda)) +
  labs(
    y = "",
    title = latex2exp::TeX(
      paste0(
        "Transformed gas Volume production with $\\lambda$ = ",
        round(lambda, 2)
      )
    )
  )
```



The point of a Box-Cox transformation is to make the size of seasonal variation similar across the series. From these two plots, it seems like the transformation didn't smooth out the seasonal variation very much. This could be due to that period of "stagnation" that occurs from around 1975 through the late 1980s.

Question 3.4

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

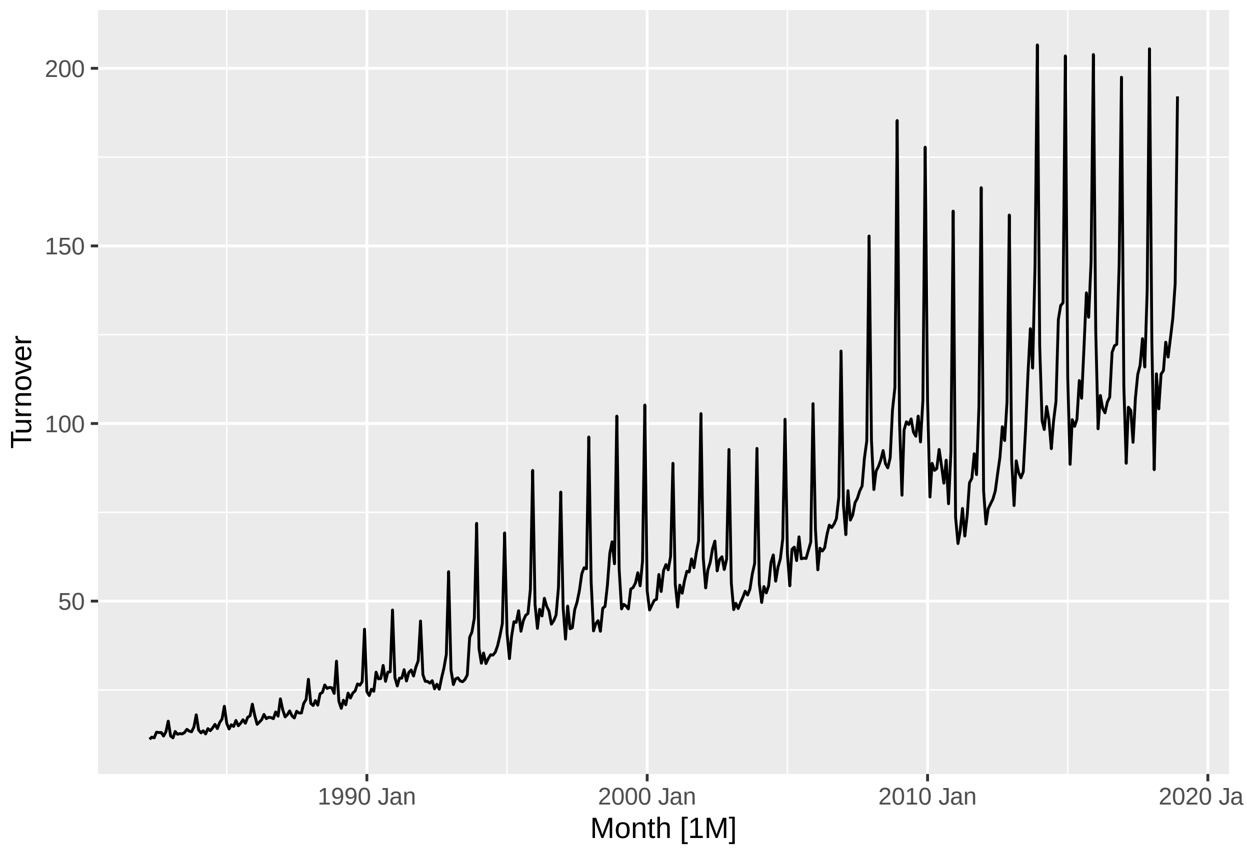
Importing data from exercise 7 in section 2.10:

```
set.seed(2111994)

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

head(myseries)
```

```
## # A tsibble: 6 x 5 [1M]
## # Key:       State, Industry [1]
##   State      Industry                            `Series ID`    Month Turnover
##   <chr>      <chr>                               <chr>          <mth>    <dbl>
## 1 Queensland Other recreational goods retailing A3349480L   1982 Apr     11.1
## 2 Queensland Other recreational goods retailing A3349480L   1982 May     11.7
## 3 Queensland Other recreational goods retailing A3349480L   1982 Jun     11.5
## 4 Queensland Other recreational goods retailing A3349480L   1982 Jul     13.1
```

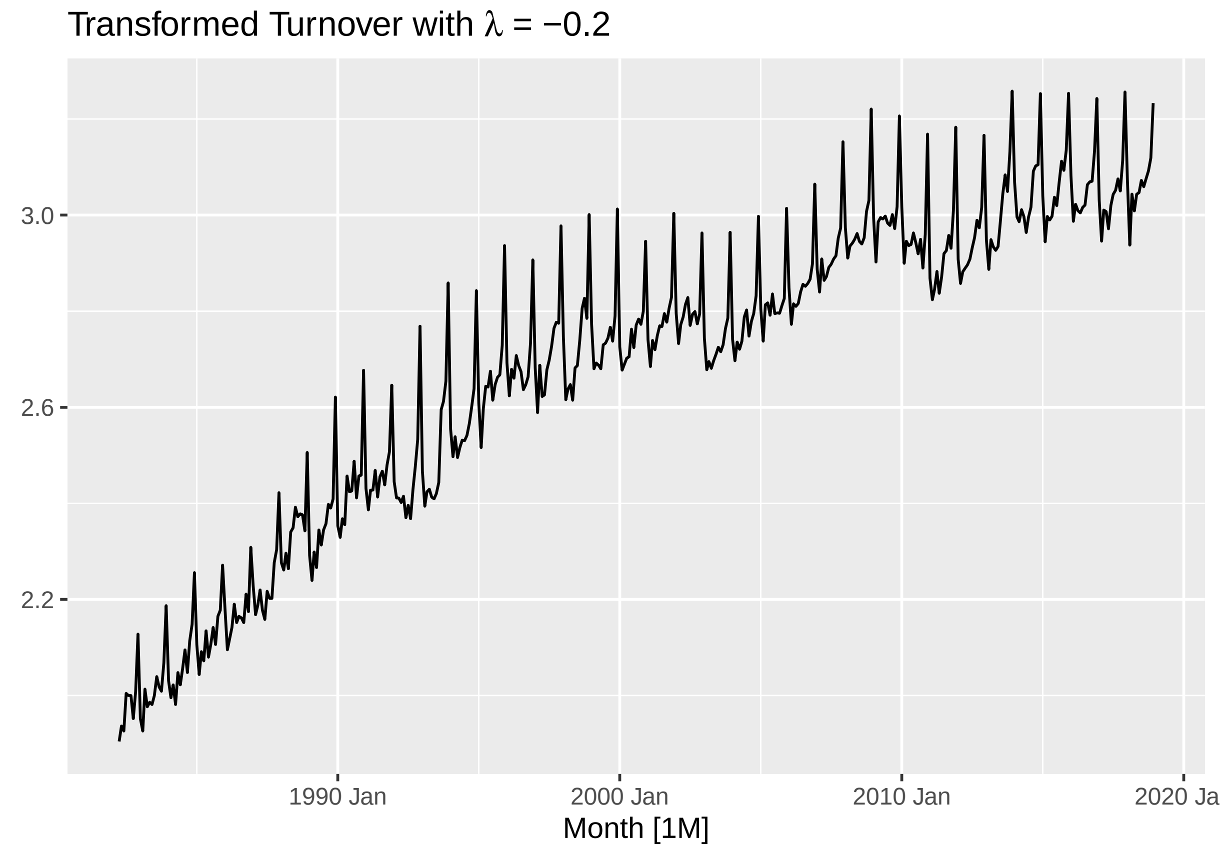
```
## 5 Queensland Other recreational goods retailing A3349480L   1982 Aug     13
## 6 Queensland Other recreational goods retailing A3349480L   1982 Sep     13
```

```
autoplot(
  myseries,
  Turnover
)
```



I would start by using guerrero to pick a lambda for the transformation:

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

myseries |>
  autoplot(box_cox(Turnover, lambda)) +
  labs(
    y = "",
    title = latex2exp::TeX(
      paste0(
        "Transformed Turnover with $\\lambda$ = ",
        round(lambda, 2)
      )
    )
  )
```



This did a much better job at smoothing out the seasonal variations as each season seems to be around the same size.

Question 3.5

For the following series, find an appropriate Box-Cox transformation in order to stabilize the variance.

I honestly don't see much of a reason to not use guerrero to help pick a lambda value. Because of this, I'm going to create 2 functions. One to get the lambda and the other to create a plot with it.

```r
data(aus_production)
head(aus_production)
```

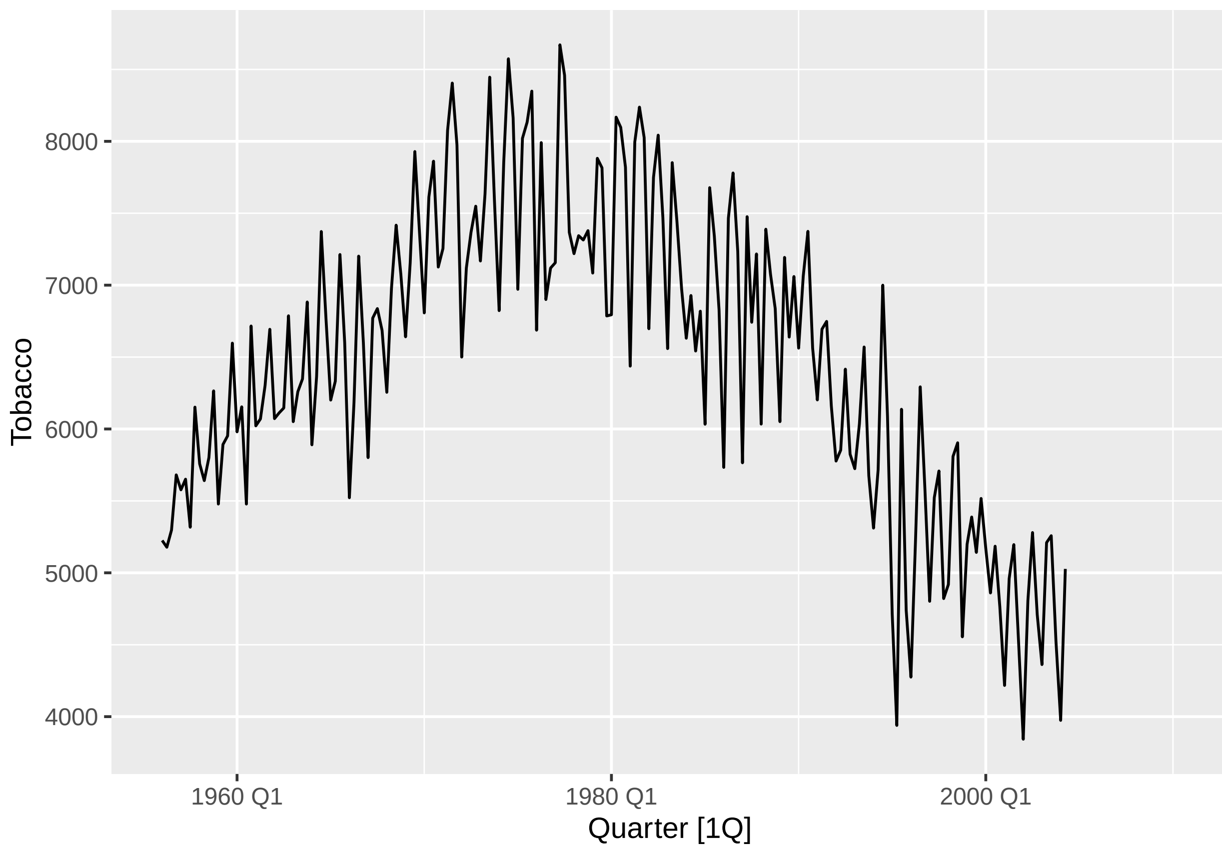
Tobacco from aus_production

```
## # 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
```

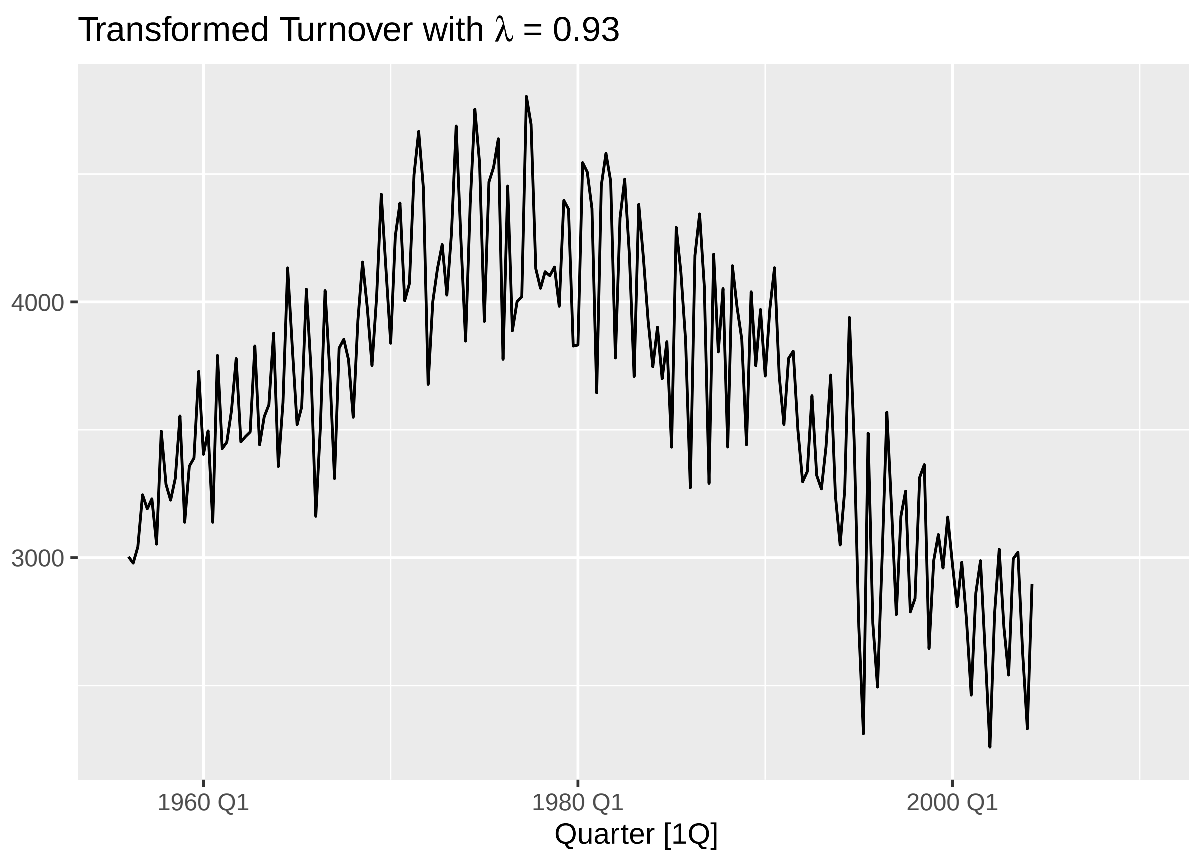
```
get_guerrero_lambda <- function(ts, column) {
  lambda <- ts |>
    features({{column}}, features = guerrero) |>
    pull(lambda_guerrero)

  return(lambda)
}

plot_box_tranformed_graph <- function(ts, column){
  lambda <- get_guerrero_lambda(
    ts,
    {{column}}
  )

  ts |>
  autoplot(box_cox({{column}}, lambda)) +
  labs(
    y = "",
    title = latex2exp::TeX(
      paste0(
        "Transformed Turnover with $\\lambda$ = ",
        round(lambda, 2)
      )
    )
  )
}

lambda3a <- get_guerrero_lambda(
  aus_production,
  Tobacco
)

autoplot(
  aus_production,
  Tobacco
)
```

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




```
plot_box_tranformed_graph(
  aus_production,
  Tobacco
)
```

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



The lambda here is 0.93 which is pretty close to 1 signaling that a Box-Cox transformation isn't very useful.

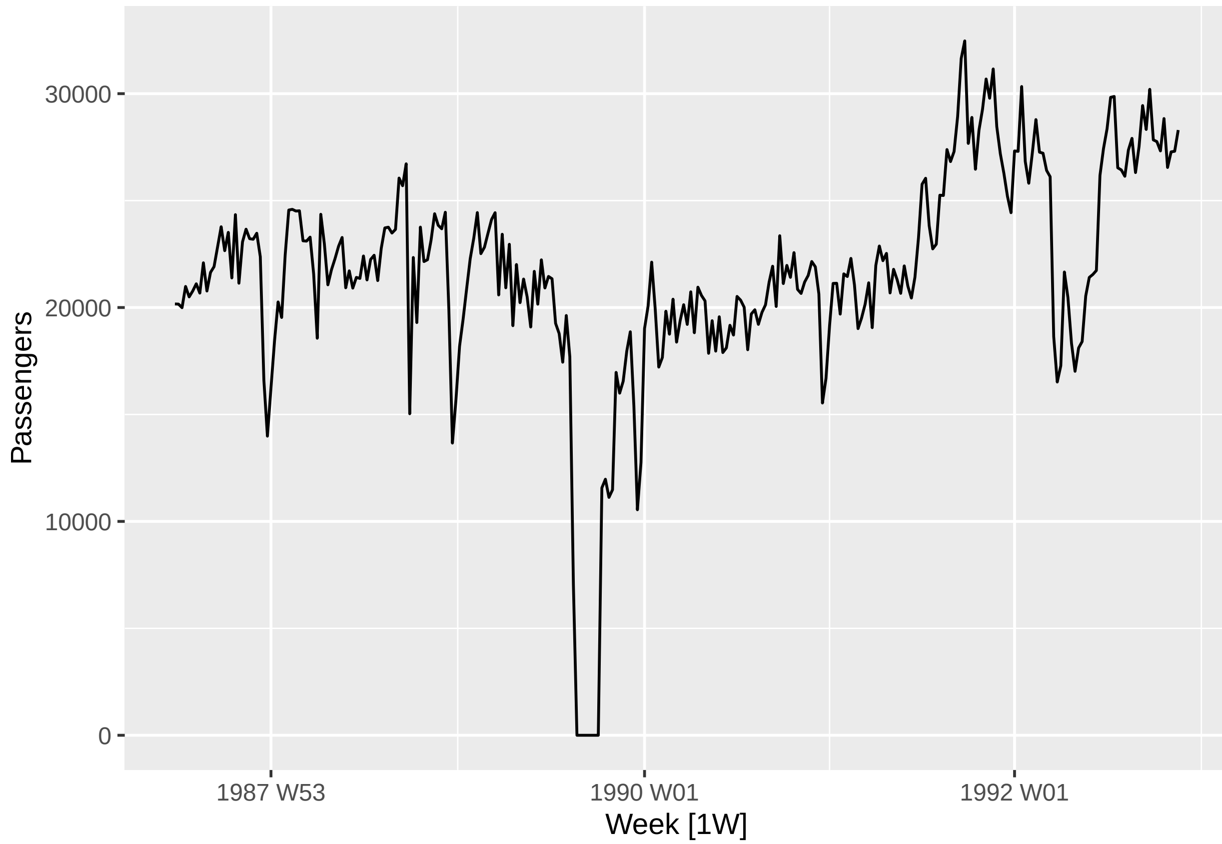
```r
data(ansett)
head(ansett)
```

Economy class passengers between Melbourne and Sydney from ansett

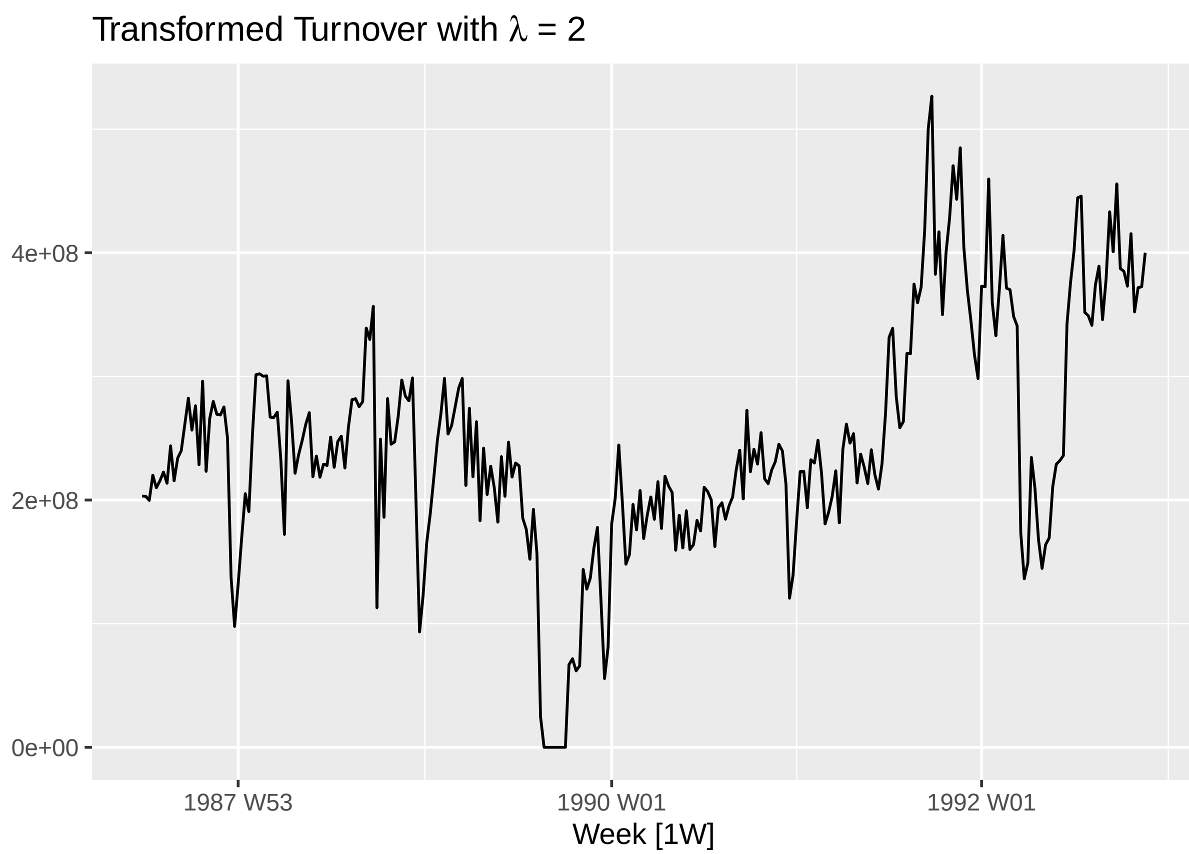
```
## # A tsibble: 6 x 4 [1W]
## # Key:       Airports, Class [1]
##       Week Airports Class    Passengers
##     <week> <chr>    <chr>         <dbl>
## 1 1989 W28 ADL-PER  Business        193
## 2 1989 W29 ADL-PER  Business        254
## 3 1989 W30 ADL-PER  Business        185
## 4 1989 W31 ADL-PER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

```r
melsyd <- ansett |>
  filter(
    Airports == "MEL-SYD",
    Class == "Economy"
  )
```

```r
lambda3b <- get_guerrero_lambda(
  melsyd,
  Passengers
)

autoplot(
  melsyd,
  Passengers
)
```




```r
plot_box_tranformed_graph(
  melsyd,
  Passengers
)
```



The lambda here is 2 which means that the data shows better seasonal variations when it undergoes the equation below with $\lambda = 2$

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(y), & \text{if } \lambda = 0 \end{cases}$$

```r
data(pedestrian)
head(pedestrian)
```

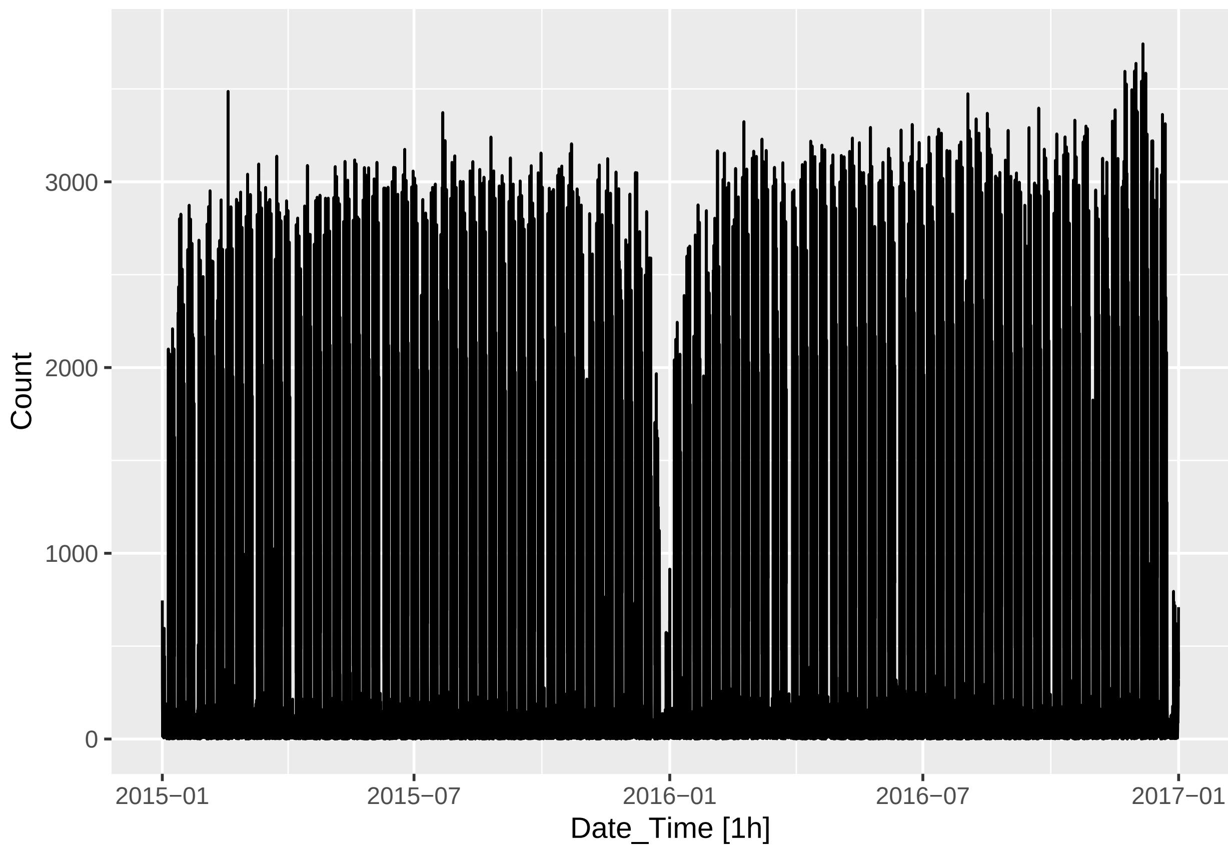
Pedestrian counts at Southern Cross Station from pedestrian.

```
## # 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
```

```r
unique(pedestrian$Sensor)
```

```
## [1] "Birrarung Marr"                "Bourke Street Mall (North)"
## [3] "QV Market-Elizabeth St (West)" "Southern Cross Station"
```

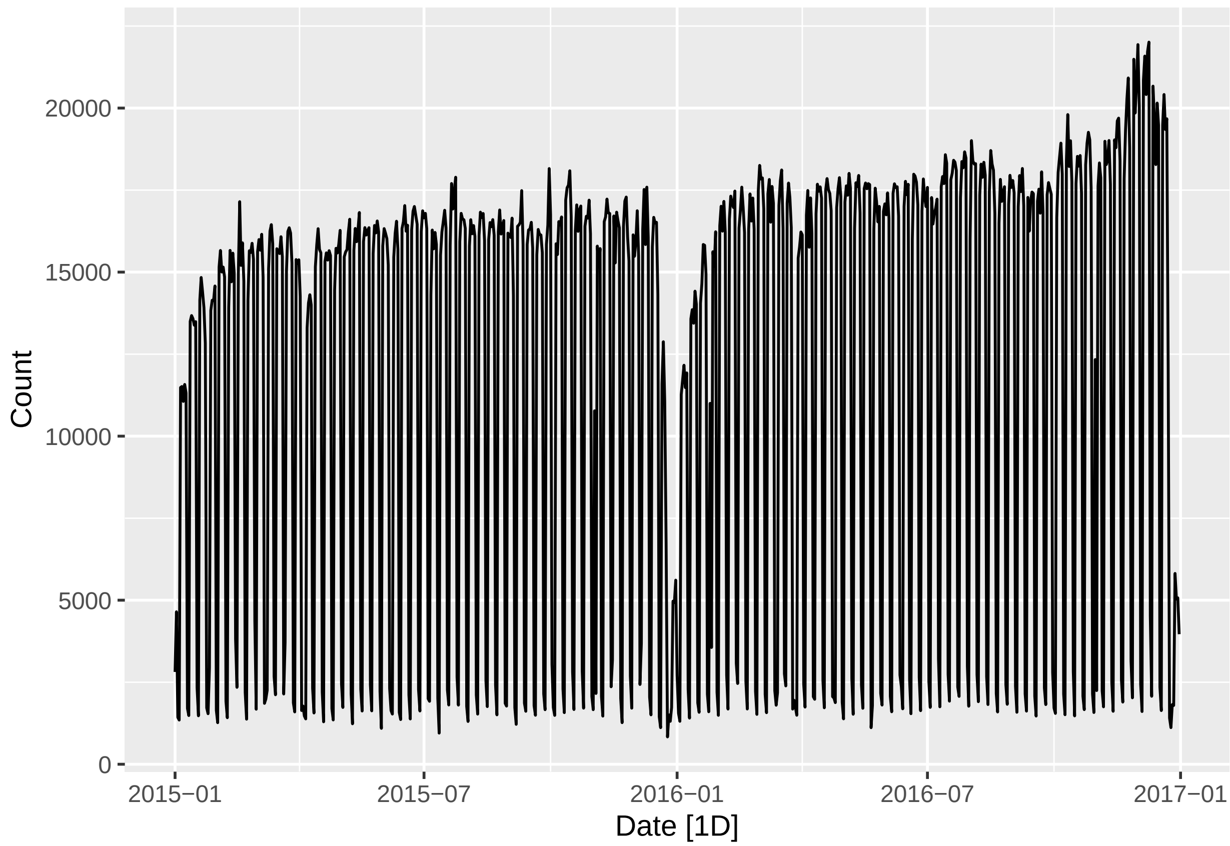
```r
scross <- pedestrian |>
  filter(
    Sensor == "Southern Cross Station"
  )

autoplot(
  scross,
  Count
)
```



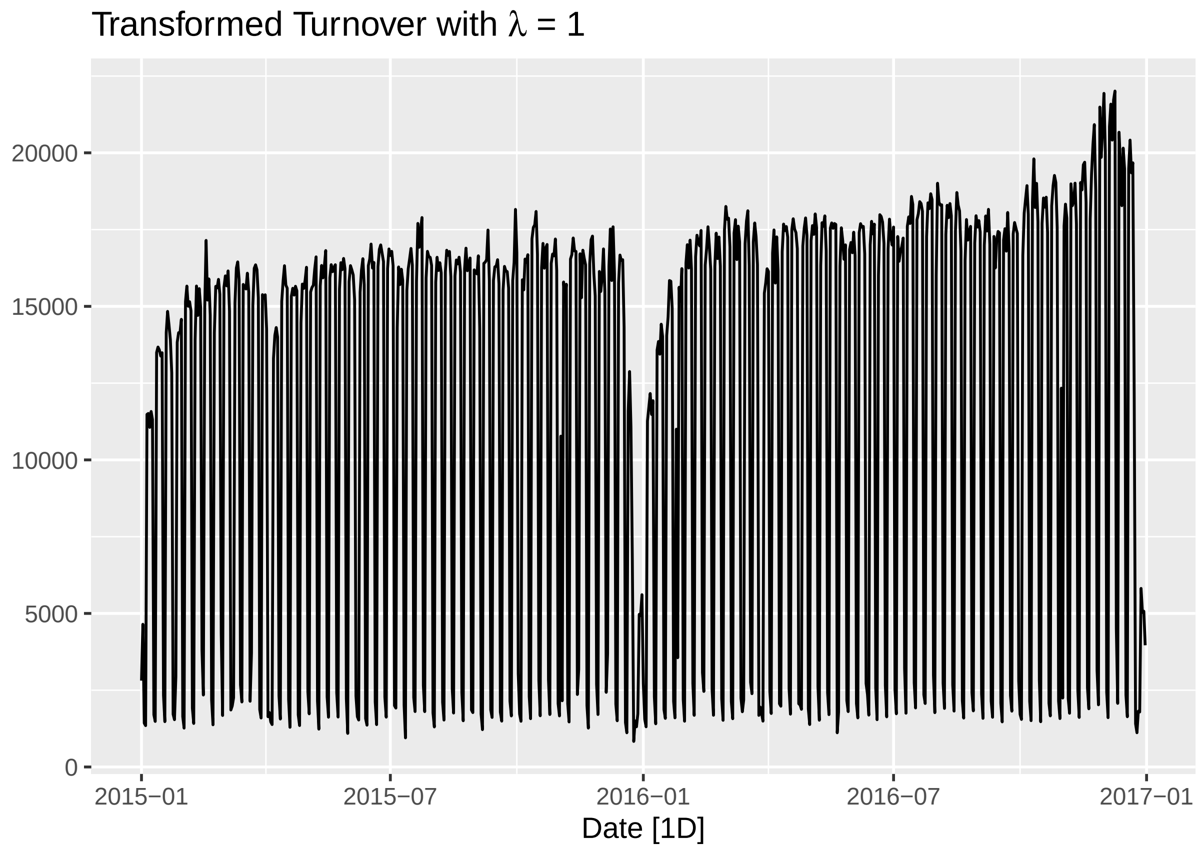
The `pedestrian` dataset is 2 years of hourly count data which is incredibly noisy. So let's group on day:

```r
dly_scross <- scross |>
  group_by(
    Date
  ) |>
  mutate(
    Count = sum(Count)
  ) |>
  distinct(
```

```
    Date,
    Count
  ) |>
  as_tsibble(
    index = Date
  )

lambda3c <- get_guerrero_lambda(
  dly_scross,
  Count
)

autoplot(
  dly_scross,
  Count
)
```



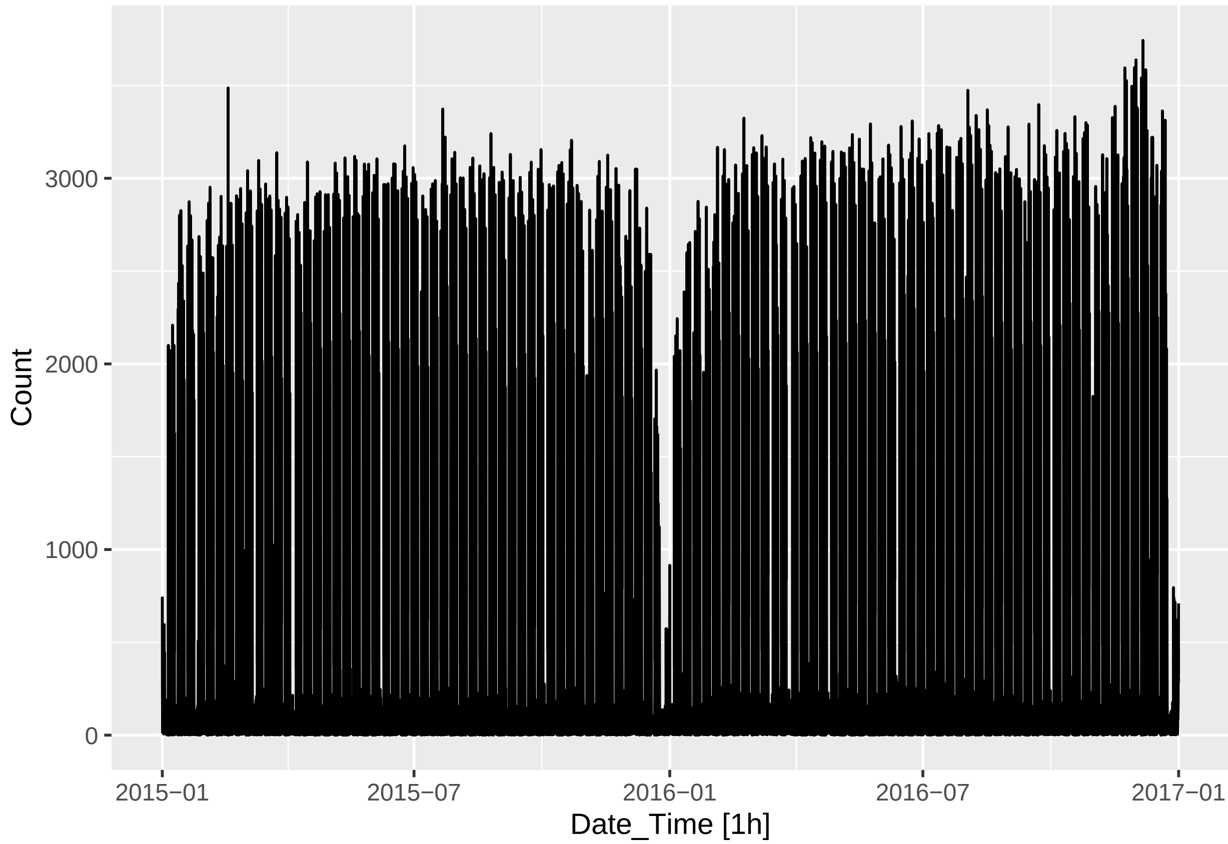

```
plot_box_tranformed_graph(
  dly_scross,
  Count
)
```



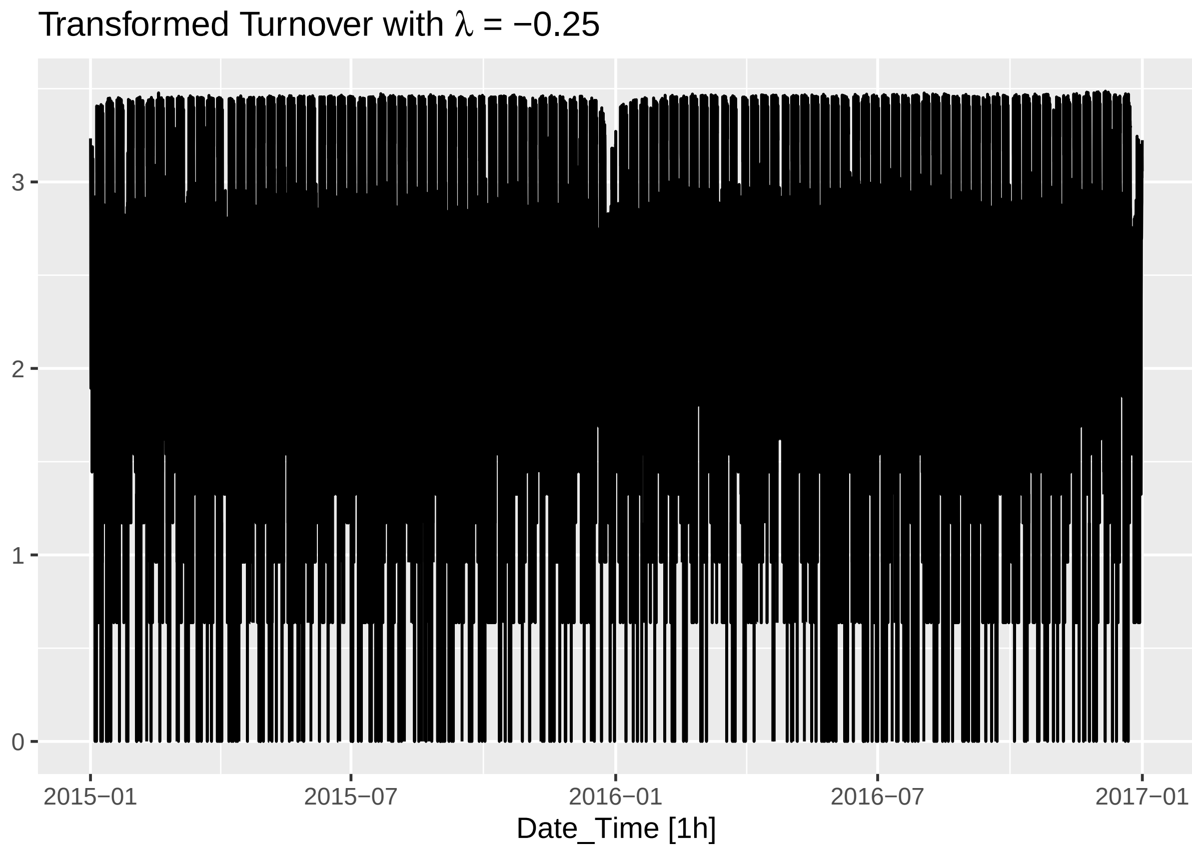
Here Lambda is 1 again, meaning that a box-cox transform isn't very helpful for this data daily. Although the data does look fairly consistent with the magnitude of the peaks and valleys.

Trying again with the hourly data:

```
lambda3d <- get_guerrero_lambda(
  scross,
  Count
)

autoplot(
  scross,
  Count
)
```




```
plot_box_tranformed_graph(
  scross,
  Count
)
```



The hourly data has a huge mass of values so tightly together that it's difficult to extract much but we do have a lambda value of -0.25 which we can see in the second chart by the new location of the mass of values that there was a significant change.

Question 3.7

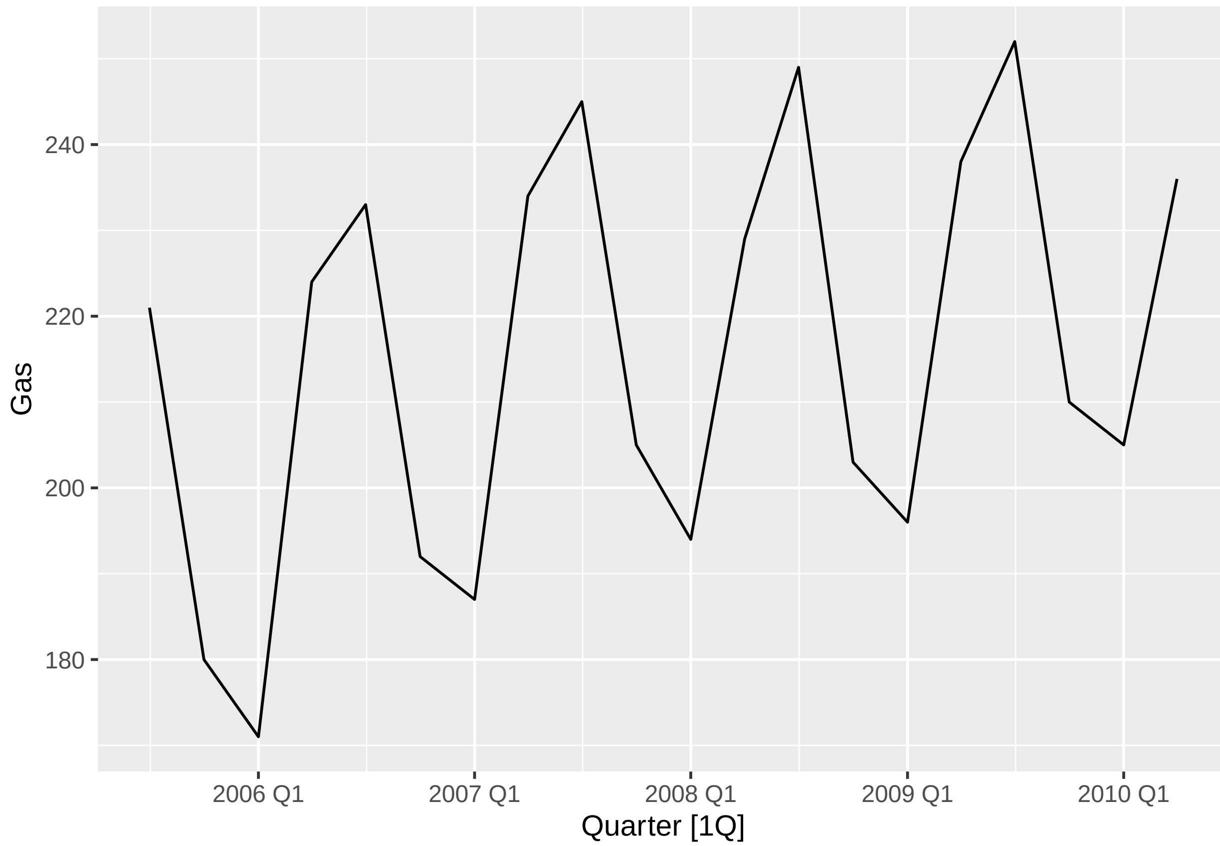
Consider the last five years of the Gas data from aus_production.

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

```r
autoplot(
  gas,
  Gas
)
```

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

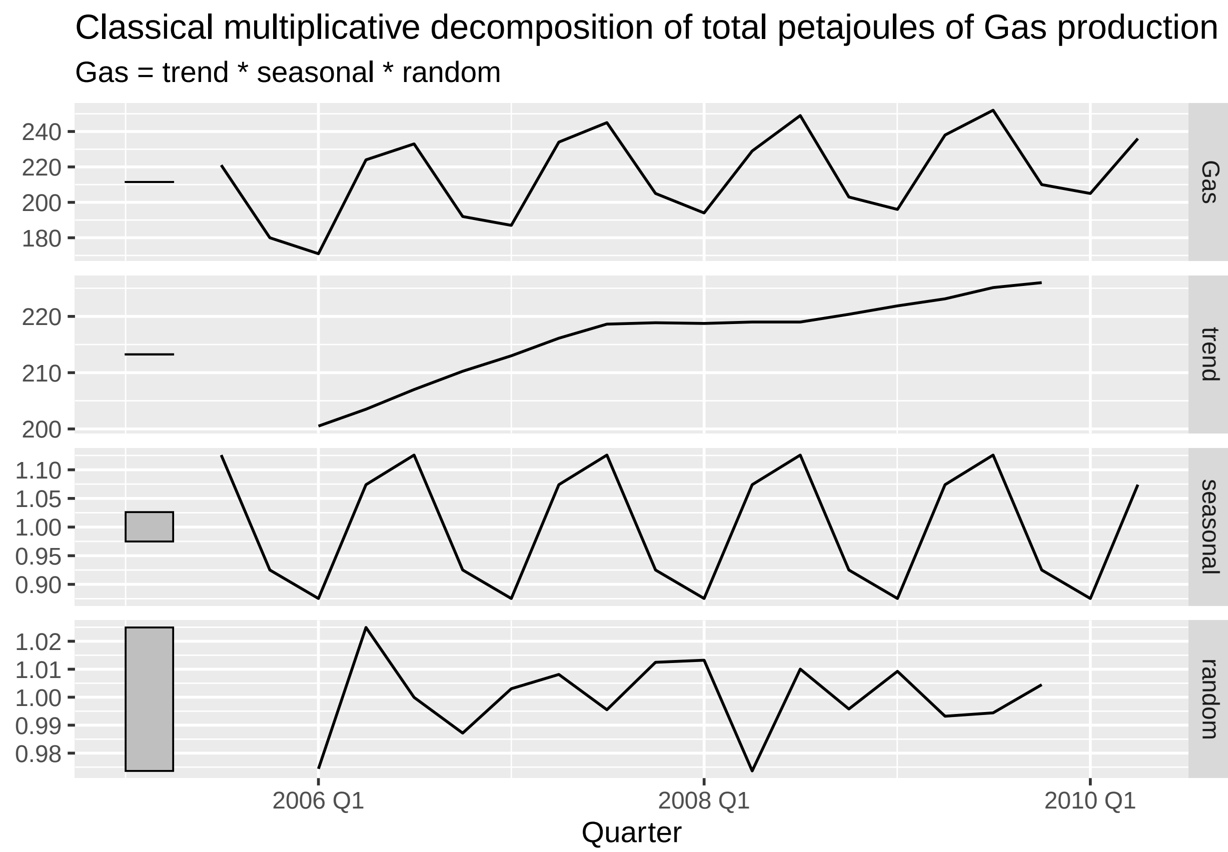


From the first graph we can clearly see that there is some cyclicality to this chart where the values seem to increase from Q1 to Q3 and decrease from Q3 to Q4. It can also be seen by comparing peaks to each other that there is an increasing trend over time.

```r
gas |>
  model(
    classical_decomposition(Gas, type = "multiplicative")
  ) |>
  components() |>
  autoplot() +
  labs(
    title = "Classical multiplicative decomposition of total petajoules of Gas production"
  )
```

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

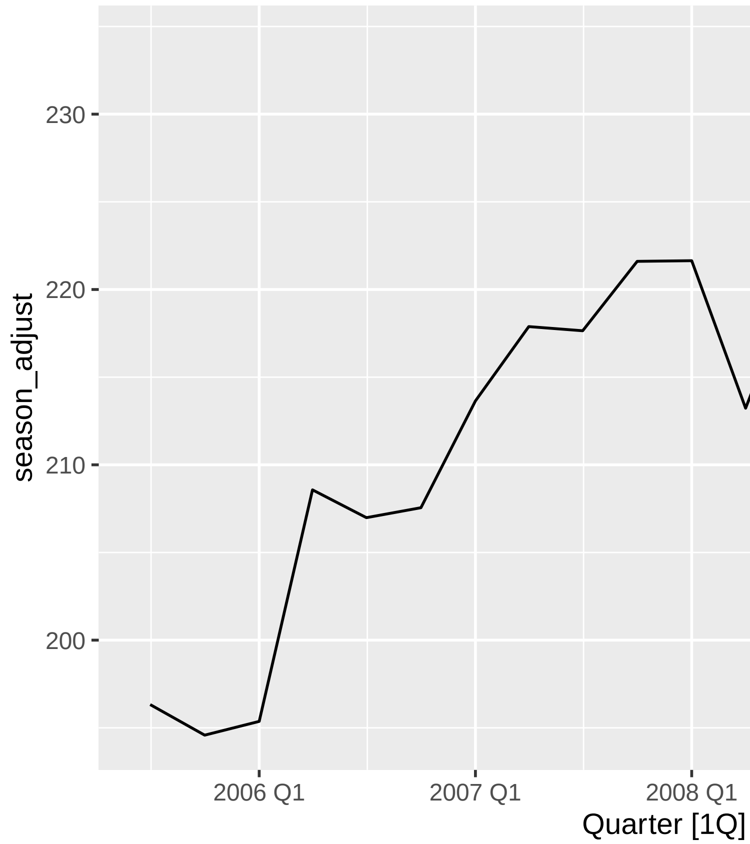
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



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

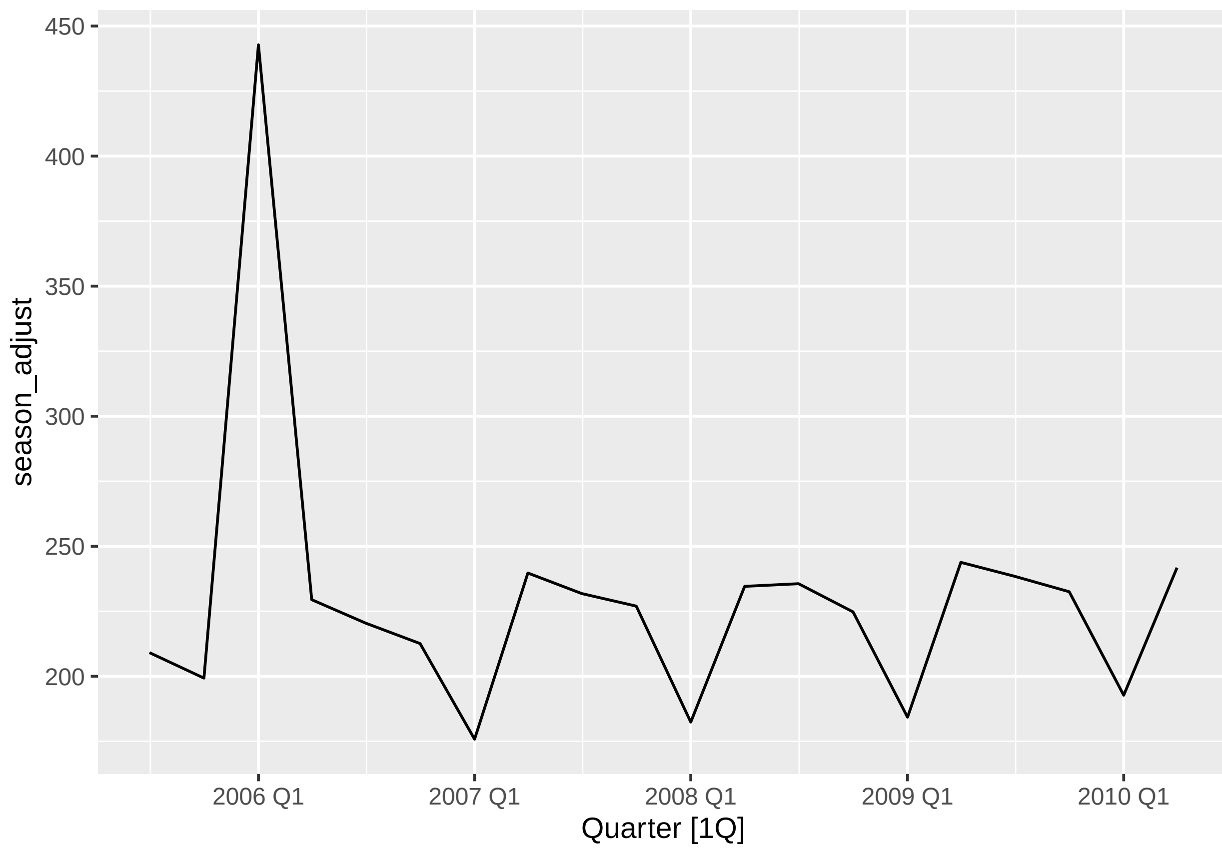
Yes they do. There a seasonal component that increases from Q1 to Q3 and decreases from Q3 to Q4. Lastly it also has a generally upward trend.

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

autoplot(
  components(gas_decomp) |>
    as_tsibble(
      index = Quarter
    ),
  season_adjust
)
```



D. Compute and plot the seasonally adjusted 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?

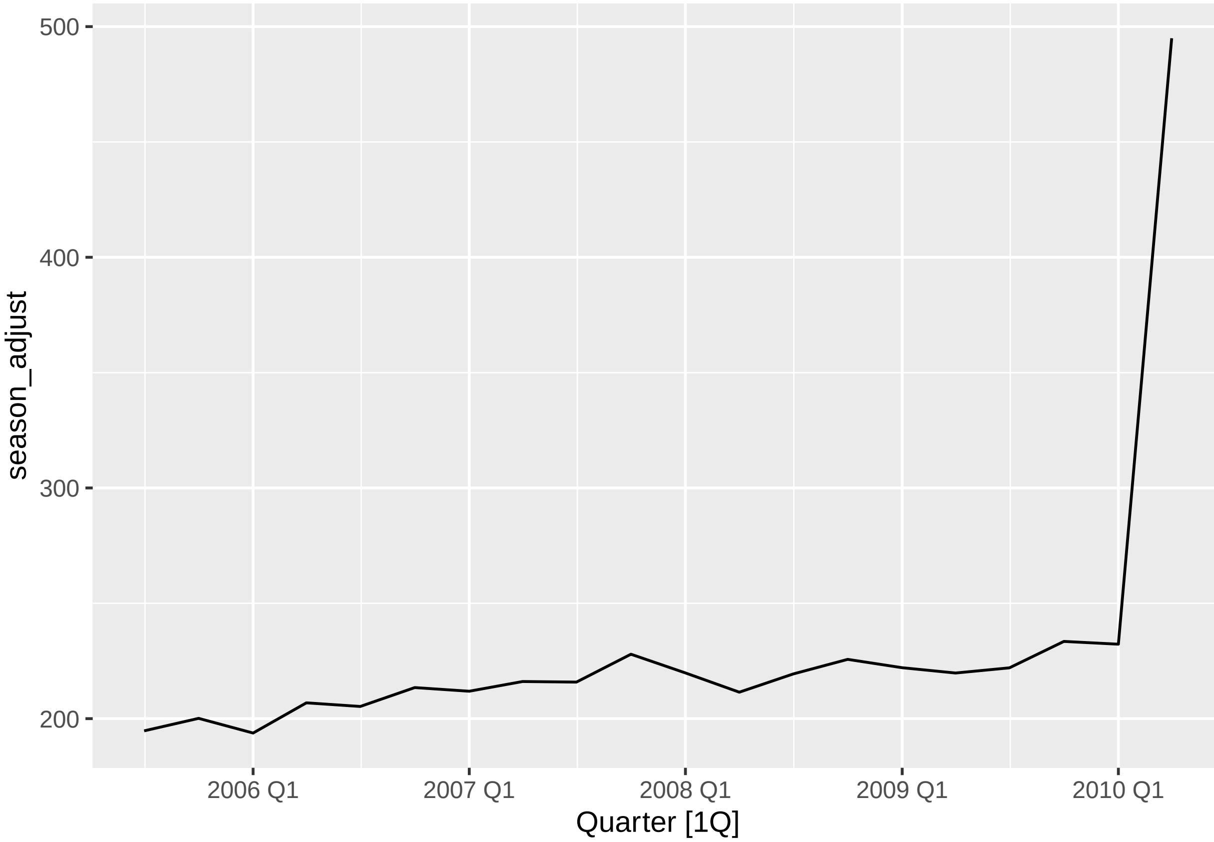
```
gas |>
  mutate(
    Gas =  Gas + ifelse(
      Gas == min(gas$Gas), 300, 0
    )
  ) |>
  model(
    classical_decomposition(
      Gas,
      type = "multiplicative"
    )
  ) |>
  components() |>
  as_tsibble() |>
  autoplot(season_adjust)
```



By Adding 300 to the minimum gas production we can see that Q1 2006 becomes an outlier and the chart now essentially shows the seasonality of the dataset.

```
gas |>
  mutate(
    Gas =  Gas + ifelse(
      Gas == 236, 300, 0
    )
  ) |>
  model(
    classical_decomposition(
      Gas,
      type = "multiplicative"
    )
  ) |>
  components() |>
  as_tsibble() |>
  autoplot(season_adjust)
```

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



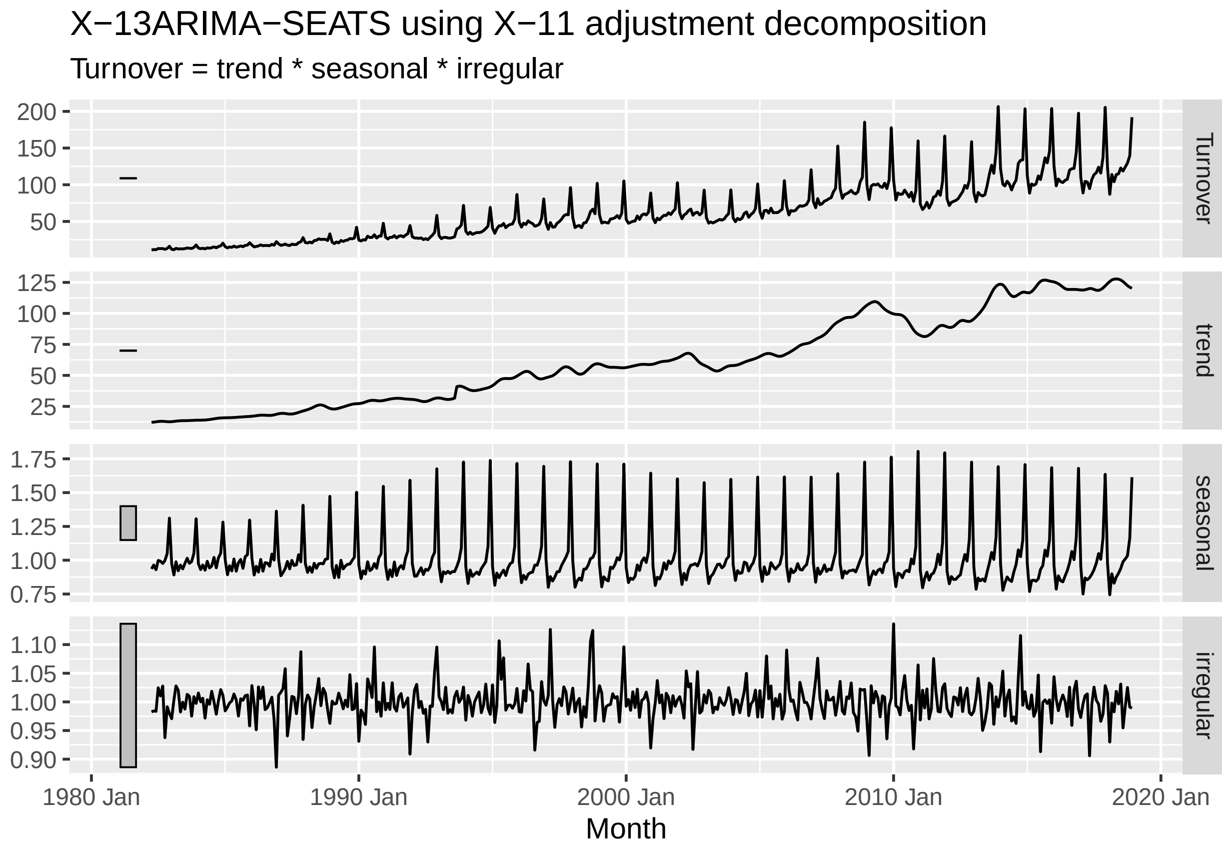
series?

By adding 300 to the last datapoint in the series, we can see that the seasonally adjusted data skyrockets but the trend does seem to follow more closely the trend we noticed before any observation was modified.

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?

```
retail_x11_dcmp <- myseries |>
  model(x11 = X_13ARIMA_SEATS(Turnover ~ x11())) |>
  components()

autoplot(
  retail_x11_dcmp,
  Turnover
)
```



By using the x11 decomposition, I can see that there is a strong seasonal component as well as a generally upwards trend which seems to experience strong growth from around 2005 to 2008. Understandably, during the financial crisis of 2008, there is a sharp drop-off and a relatively slow growth afterwards. It can also be noted that the seasonal trend using x11 has much more variation within it too.

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.

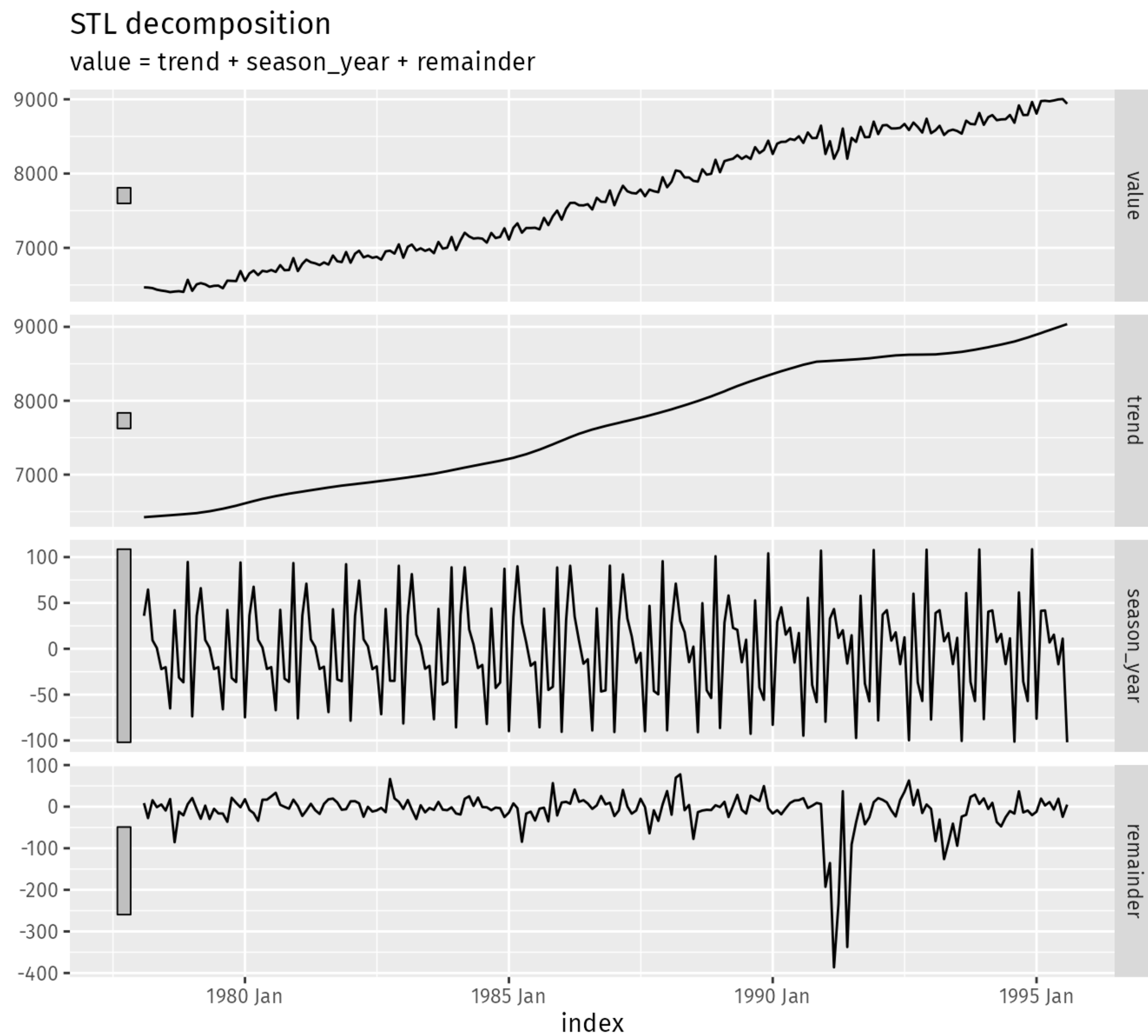
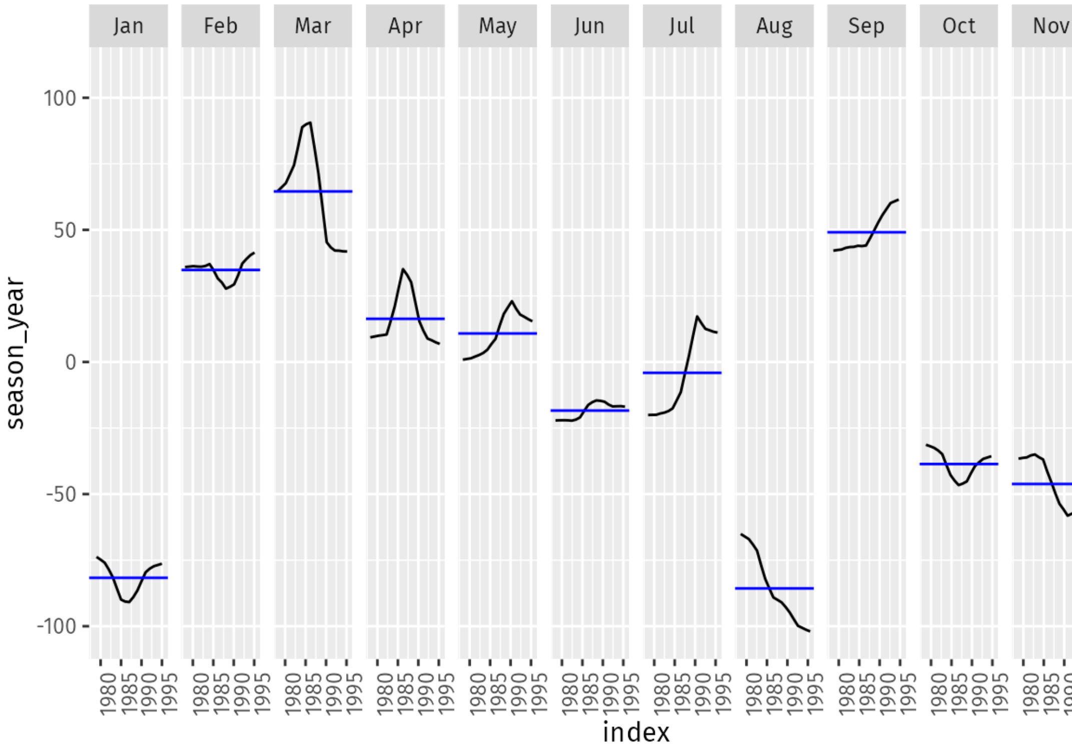


Figure 3.19: Decomposition of the number of persons in the civilian labour force in Australia each month from



February 1978 to August 1995.
Figure 3.20: 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.

Firstly, the trend is fairly obviously increasing as time goes on. This trend is pretty apparent from the raw timeseries values. Looking at the seasonal component, it seems that this timeseries is pretty regular with its seasonality. Looking at the figure 3.20, we can see that January and August are two months of very low turnover but December and March are very high months for turnover. This can be seen by loooking at the Blue lines for each month. Aside from those two months.

B. Is the recession of 1991/1992 visible in the estimated components? The recession in 1991/1992 is very visible in the remainder section of the decomposition. We can see that during these years there is a massive increase in the magnitude of the remainder meaning that this period did not fit any other observable seasonality/trend.