Assignment 0

Applied Forecasting in Complex Systems 2022

Week 1

On this course we use R-software. R is a widely used and freely distributed programming language that is particularly suitable for statistical analysis. The exercises follow the book Forecasting: Principles and Practice, 2021 (https://otexts.com/fpp3) and can be solved by using the basic R-software, but you can use Python (you'll need to code by yourself some function that are not built in), which you can download free of charge to your personal computer. There exists many different integrated development environments (IDE) for the R programming language. We recommend that you use the one called RStudio. RStudio is also free of charge.
This first assignment is focus on:

- applying how to make ts object and deal with it using several functions like autoplot(), frequency(), etc.
- applying how to draw several types of plots like seasonal plot, seasonal subseries plot, scatter plots and lag plot, etc.
- understanding what is autocorrelation and learn how to draw autocorrelation function (ACF) plot.
- understanding what is white noise and learn how to discern white noise series using ACF plots.

For practice exercises of Week 1:

- Make sure you are familiar with R, RStudio and the tidyverse packages.
- Otherwise: read the first four chapters of "ModernDive": [`moderndive.netlify.com`] (https://moderndive.netlify.com/)

Introduction to working with R in Jupyter Notebook

Running R in Jupyter With The R Kernel

To work with R, you'll need to load the IRKernel and activate it to get started on working with R in the notebook environment.

install some packages

1) Open regular R terminal or RStudio terminal:

a) install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))

b) devtools::install_github('IRkernel/IRkernel')

c) IRkernel::installspec(user = FALSE)

2) Via Command Prompt: conda create -n my-r-env -c r r-essentials

Source: [`Installing the R kernel in Jupyter Lab`](https://github.com/IRkernel/IRkernel)

This course will also follow Library(fpp3) that has the following packages:

- `tibble`, for tibbles, a modern re-imagining of data frames.
- `dplyr`, for data manipulation.
- `tidyr`, to easily tidy data using spread() and gather().
- `lubridate`, for date/times.
- `ggplot2`, for data visualisation.
- `tsibble`, for tsibbles, a time series version of a tibble.
- `tsibbledata`, various time series data sets in the form of tsibbles.
- `feasts`, for features and statistics of time series.
- `fable`, for fitting models and producing forecasts.

Also, we will use some series from FPP2 package content: ['Package 'fpp2'](https://cran.r-project.org/web/packages/fpp2/fpp2.pdf)

```
fpp2_packages()
```

```
## [1] "cli"        "crayon"     "expsmooth"  "fma"        "forecast"
## [6] "ggplot2"    "magrittr"   "purrr"      "rstudioapi"
```

FPP3 package content: ['Package 'fpp3'](https://cran.r-project.org/web/packages/fpp3/fpp3.pdf)

```
fpp3_packages()
```

```
##  [1] "cli"         "crayon"      "dplyr"       "fable"       "fabletools"
##  [6] "feasts"      "ggplot2"     "lubridate"   "magrittr"    "purrr"
## [11] "rstudioapi"  "tibble"      "tidyr"       "tsibble"     "tsibbledata"
## [16] "urca"
```

Other

Google is a powerful tool if you want to find new information regarding R-programming. See for example, http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/

Exercises from Forecasting: Principles and Practice, 2021

"Throughout this course, you'll have to make comments, justifications and reflections out of why the results/implementations based on your analysis and assumptions of the method are appropiate for the problem at stake, hence critically reflect on drawing conclusions."

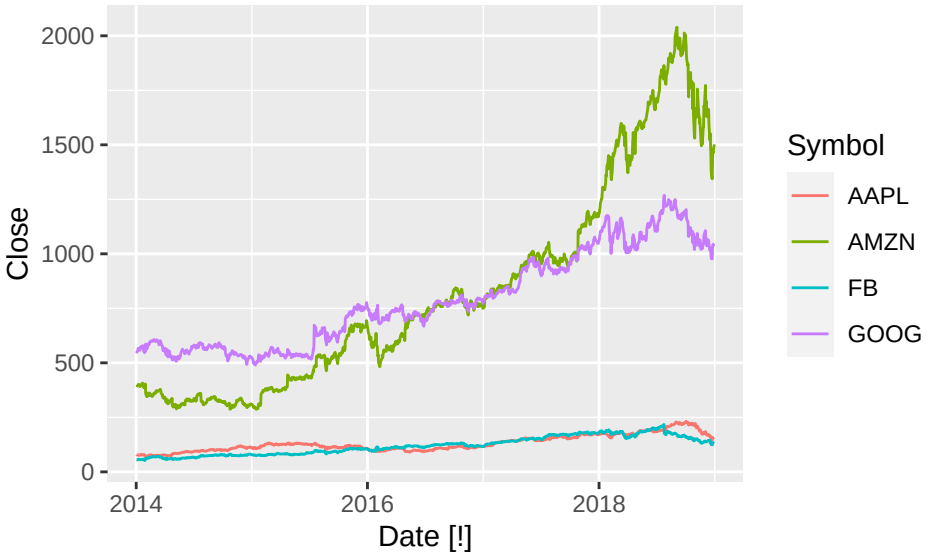
Exercise 1

Use the help function to explore what the series `gafa_stock`, `PBS`, `vic_elec` and `pelt` represent.

a. Use `autoplot()` to plot some of the series in these data sets.
b. What is the time interval of each series?

gafa_stock

```
gafa_stock %>%
  autoplot(Close)
```



Stock prices for these technology stocks have risen for most of the series, until mid-late 2018.

```
gafa_stock
```

```
## # A tsibble: 5,032 x 8 [!]
## # Key:       Symbol [4]
##    Symbol Date        Open  High   Low Close Adj_Close    Volume
##    <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
```

```
##  1 AAPL   2014-01-02  79.4  79.6  78.9  79.0      67.0  58671200
##  2 AAPL   2014-01-03  79.0  79.1  77.2  77.3      65.5  98116900
##  3 AAPL   2014-01-06  76.8  78.1  76.2  77.7      65.9 103152700
##  4 AAPL   2014-01-07  77.8  78.0  76.8  77.1      65.4  79302300
##  5 AAPL   2014-01-08  77.0  77.9  77.0  77.6      65.8  64632400
##  6 AAPL   2014-01-09  78.1  78.1  76.5  76.6      65.0  69787200
##  7 AAPL   2014-01-10  77.1  77.3  75.9  76.1      64.5  76244000
##  8 AAPL   2014-01-13  75.7  77.5  75.7  76.5      64.9  94623200
##  9 AAPL   2014-01-14  76.9  78.1  76.8  78.1      66.1  83140400
## 10 AAPL   2014-01-15  79.1  80.0  78.8  79.6      67.5  97909700
## # ... with 5,022 more rows
```

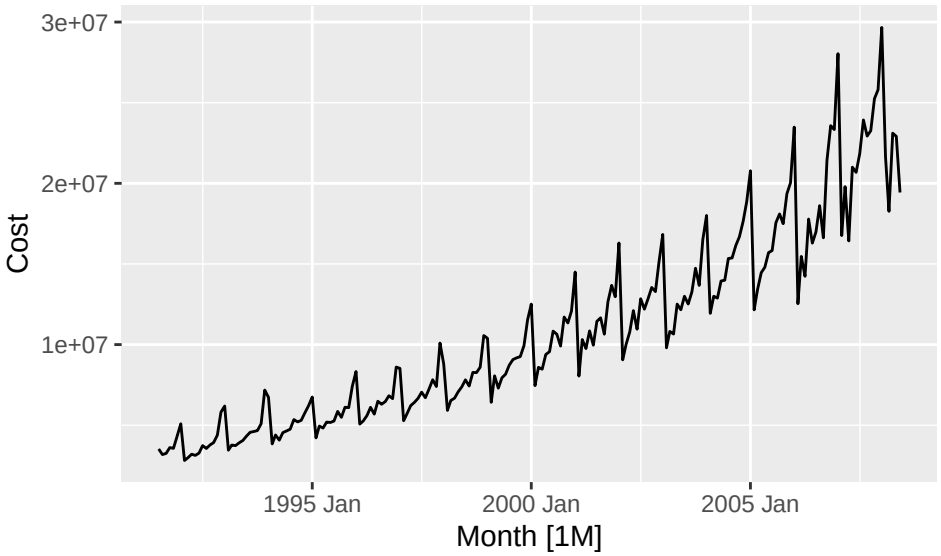
Interval is daily. Looking closer at the data, we can see that the index is a Date variable. It also appears that observations occur only on trading days, creating lots of implicit missing values.

PBS

There are too many series to plot. Let's focus on aggregate A10 expenditure.

```
a10 <- PBS %>%
  filter(ATC2 == "A10") %>%
  summarise(Cost = sum(Cost))
```

```
a10 %>%
  autoplot(Cost)
```



Appears to have upward trend (perhaps exponential), and seasonality which varies proportionately to the level of the series.

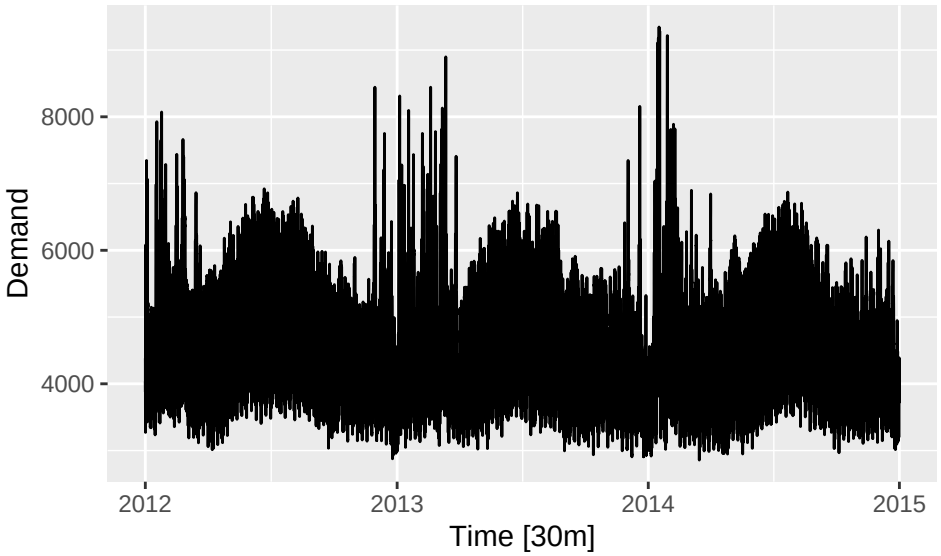
```
a10
```

```
## # A tsibble: 204 x 2 [1M]
##       Month    Cost
##       <mth>   <dbl>
##  1 1991 Jul 3526591
##  2 1991 Aug 3180891
##  3 1991 Sep 3252221
##  4 1991 Oct 3611003
##  5 1991 Nov 3565869
##  6 1991 Dec 4306371
##  7 1992 Jan 5088335
##  8 1992 Feb 2814520
##  9 1992 Mar 2985811
## 10 1992 Apr 3204780
## # ... with 194 more rows
```

Observations are made once every month.

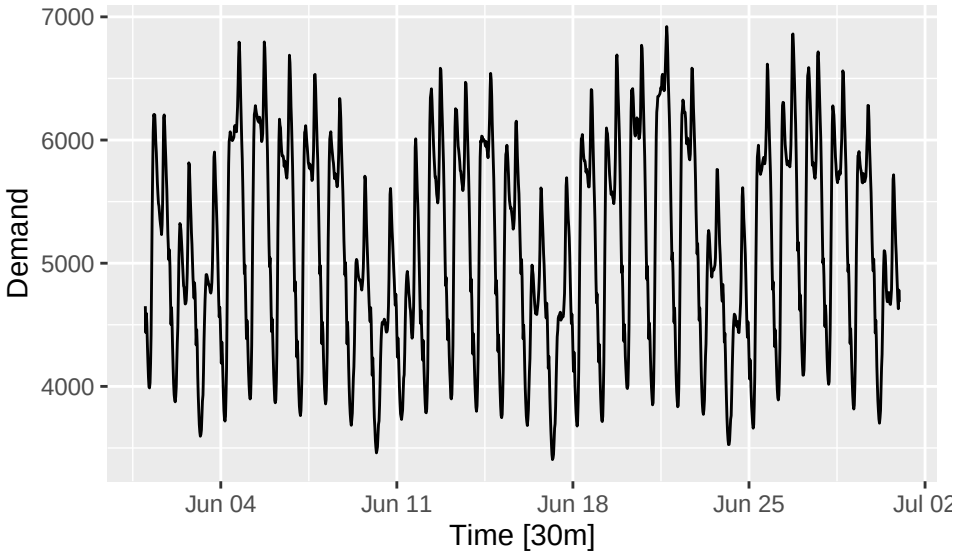
vic_elec

```
vic_elec %>%
  autoplot(Demand)
```



Appears to have an annual seasonal pattern, where demand is higher during summer and winter. Can't see much detail, so let's zoom in.

```
vic_elec %>%
  filter(yearmonth(Time) == yearmonth("2012 June")) %>%
  autoplot(Demand)
```



Appears to have a daily pattern, where less electricity is used overnight. Also appears to have a working day effect (less demand on weekends and holidays).

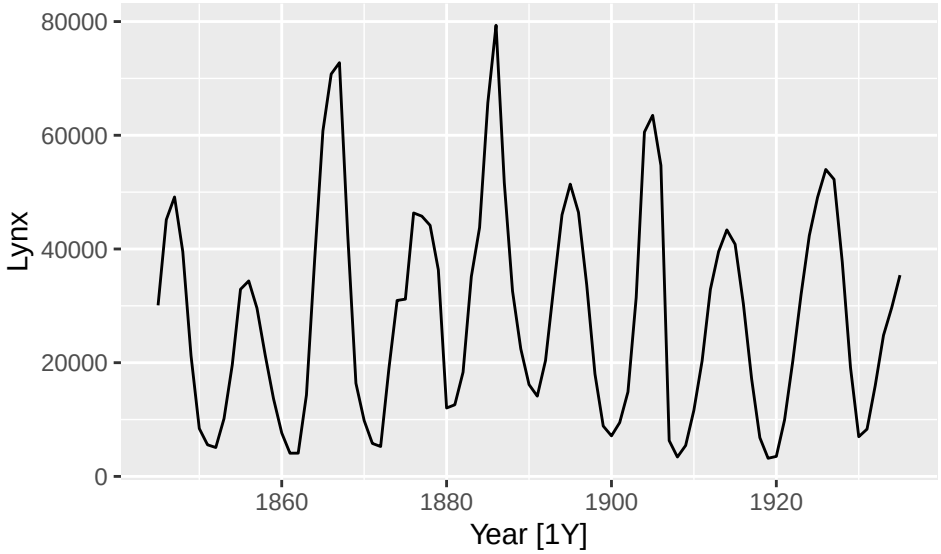
```
vic_elec
```

```
## # A tsibble: 52,608 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
##  7 2012-01-01 03:00:00  3694.        20.1 2012-01-01 TRUE
##  8 2012-01-01 03:30:00  3562.        19.6 2012-01-01 TRUE
##  9 2012-01-01 04:00:00  3433.        19.1 2012-01-01 TRUE
## 10 2012-01-01 04:30:00  3359.        19.0 2012-01-01 TRUE
## # ... with 52,598 more rows
```

Data is available at 30 minute intervals.

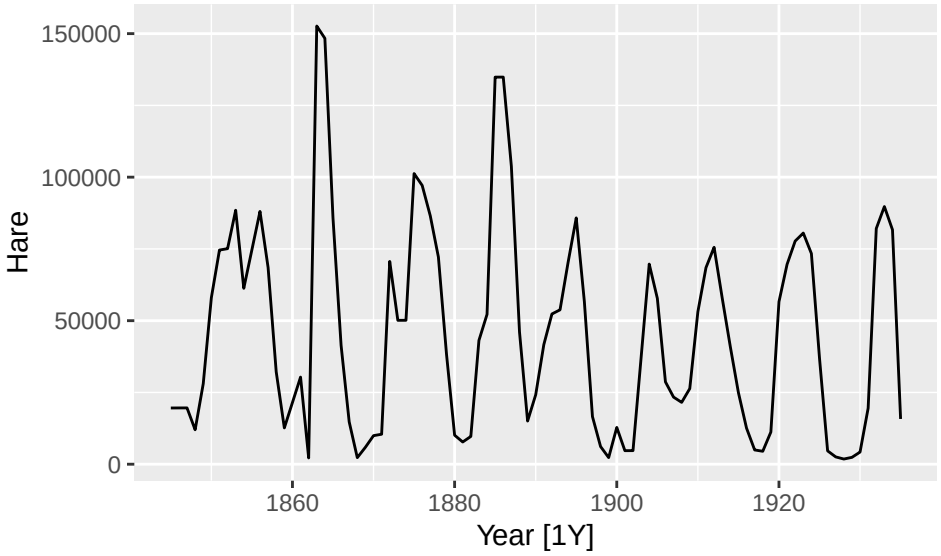
pelt

```
pelt %>% autoplot(Lynx)
```



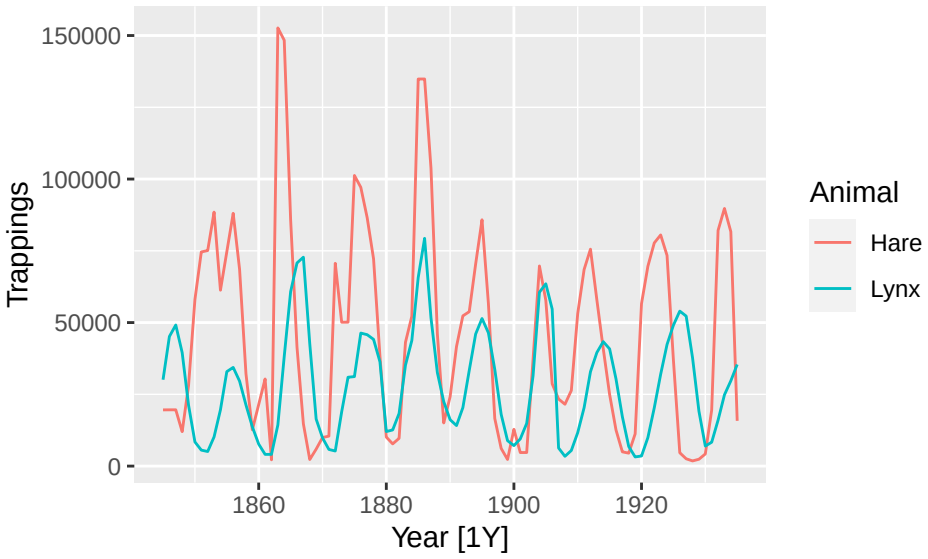
Canadian lynx trappings appears to be cyclic, as the extent of peak trappings is unpredictable, and the spacing between the peaks is irregular.

```
pelt %>% autoplot(Hare)
```



Similar can be said for snowshoe hare trappings, although this series appears more erratic.

```
pelt %>%
  pivot_longer(Hare:Lynx, names_to="Animal", values_to="Trappings") %>%
  autoplot(Trappings)
```



Plotting both Lynx and Hare trappings, it appears that the peaks in Canadian Lynx trappings occur shortly after peaks in Snowshoe Hare trappings. This relationship is due to the Canadian Lynx being specialised hunters of the Snowshoe Hare, resulting in a strong predator-prey relationship.

```
interval(pelt)
```

```
## <interval[1]>
## [1] 1Y
```

Observations are made once per year.

Exercise 2

Use `filter()` to find what days corresponded to the peak closing price for each of the four stocks in `gafa_stock`.

```
gafa_stock %>%
  group_by(Symbol) %>%
  filter(Close == max(Close)) %>%
  ungroup() %>%
  select(Symbol, Date, Close)
```

```
## # A tsibble: 4 x 3 [!]
## # Key:       Symbol [4]
##   Symbol Date       Close
##   <chr>  <date>     <dbl>
## 1 AAPL   2018-10-03  232.
## 2 AMZN   2018-09-04 2040.
## 3 FB     2018-07-25  218.
## 4 GOOG   2018-07-26 1268.
```

Exercise 3

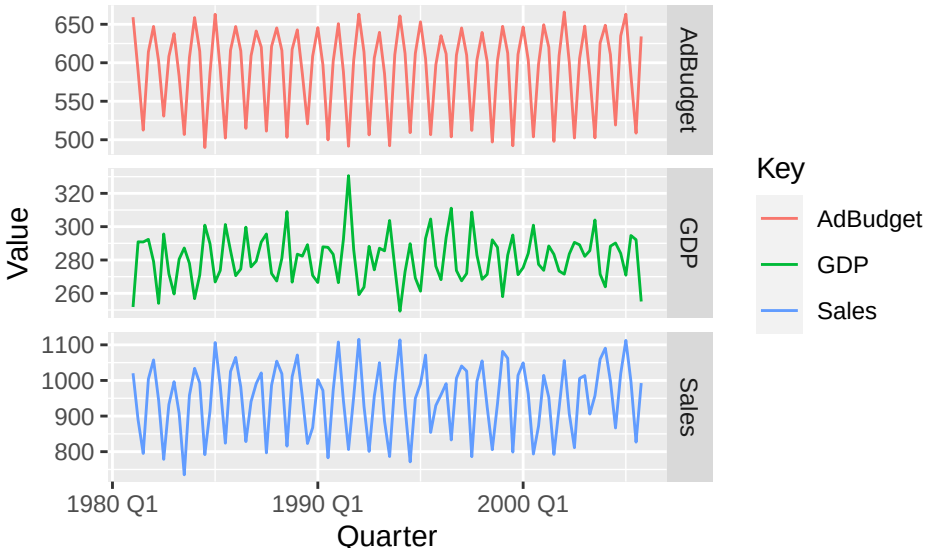
Download the file `tute1.csv` from [the book website](http://OTexts.com/fpp3/extrafiles/tute1.csv), open it in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information. Columns B through D each contain a quarterly series, labelled Sales, AdBudget and GDP. Sales contains the quarterly sales for a small company over the period 1981-2005. AdBudget is the advertising budget and GDP is the gross domestic product. All series have been adjusted for inflation.

```
download.file("http://OTexts.com/fpp3/extrafiles/tute1.csv",
              tute1 <- tempfile())
tute1 <- readr::read_csv(tute1)
```

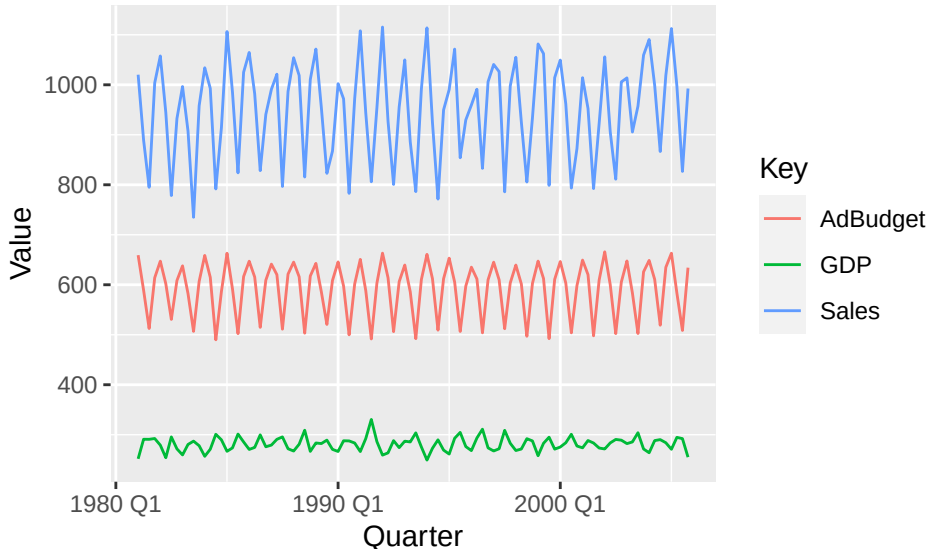
```
## Rows: 100 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (3): Sales, AdBudget, GDP
## date (1): Quarter
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(tute1)
mytimeseries <- tute1 %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(index = Quarter)

mytimeseries %>%
  pivot_longer(-Quarter, names_to="Key", values_to="Value") %>%
  ggplot(aes(x = Quarter, y = Value, colour = Key)) +
    geom_line() +
    facet_grid(vars(Key), scales = "free_y")
```



```
# Without faceting:
mytimeseries %>%
  pivot_longer(-Quarter, names_to="Key", values_to="Value") %>%
  ggplot(aes(x = Quarter, y = Value, colour = Key)) +
    geom_line()
```

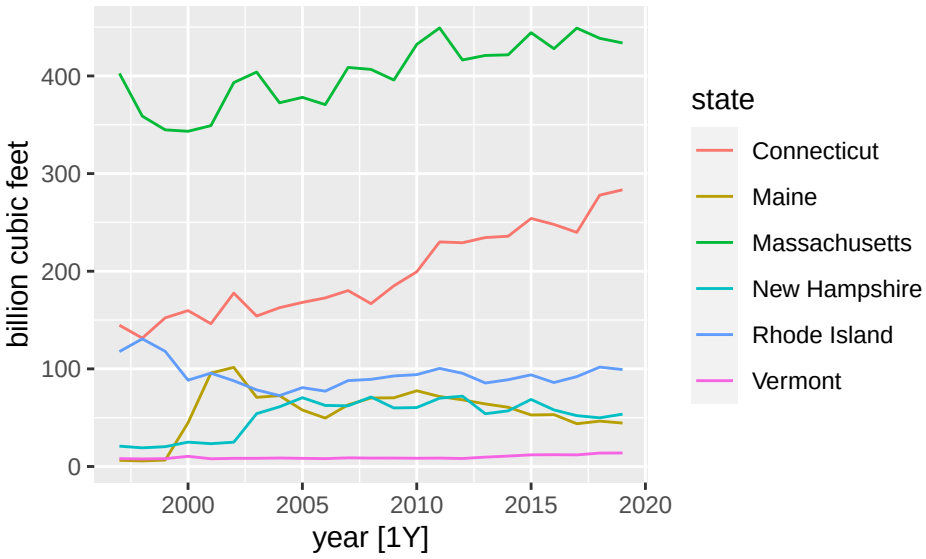


Exercise 4

The `USgas` package contains data on the demand for natural gas in the US.

a. Install the ‘USgas‘ package.
b. Create a tsibble from ‘us_total‘ with year as the index and state as the key.
c. Plot the annual natural gas consumption by state for the New England area (comprising the st

```
#install.packages("USgas")
library(USgas)
us_tsibble <- us_total %>%
  as_tsibble(index=year, key=state)
# For each state
us_tsibble %>%
  filter(state %in% c("Maine", "Vermont", "New Hampshire", "Massachusetts",
                      "Connecticut", "Rhode Island")) %>%
  autoplot(y/1e3) +
  labs(y = "billion cubic feet")
```

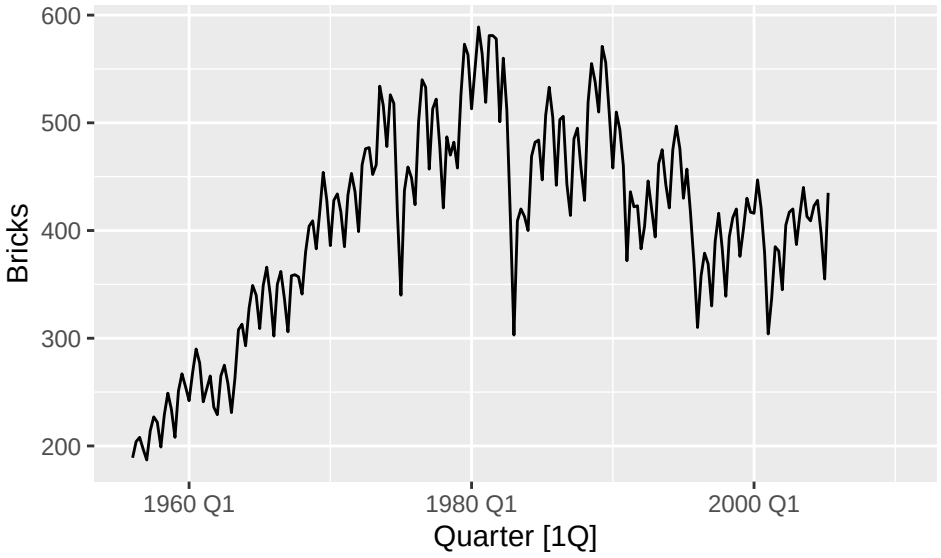


Exercise 5

Create time plots of the following four time series: `Bricks` from `aus_production`, `Lynx` from `pelt`, `Close` from `gafa_stock`, `Demand` from `vic_elec`. + Use `?` (or `help()`) to find out about the data in each series. + For the last plot, modify the axis labels and title.

Bricks

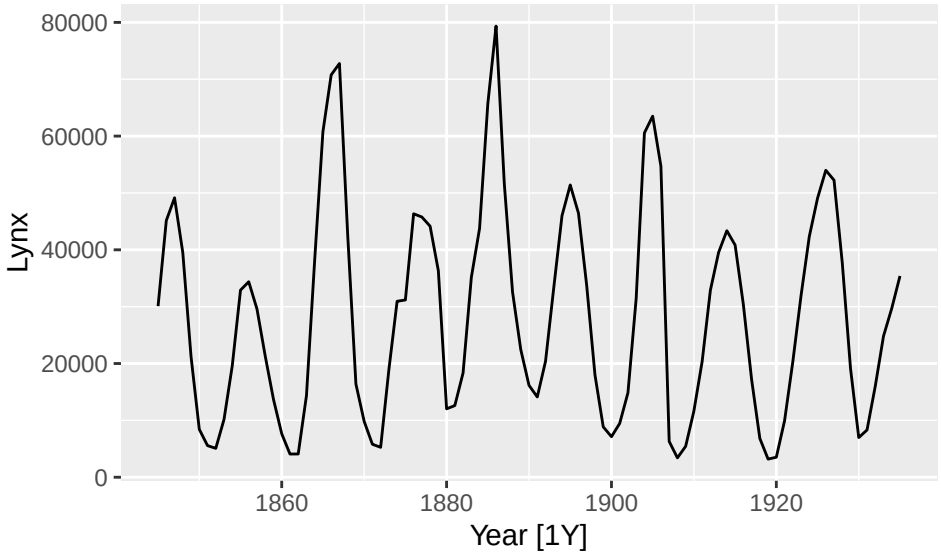
```
aus_production %>% autoplot(Bricks)
```



An upward trend is apparent until 1980, after which the number of clay bricks being produced starts to decline. A seasonal pattern is evident in this data. Some sharp drops in some quarters can also be seen.

Lynx

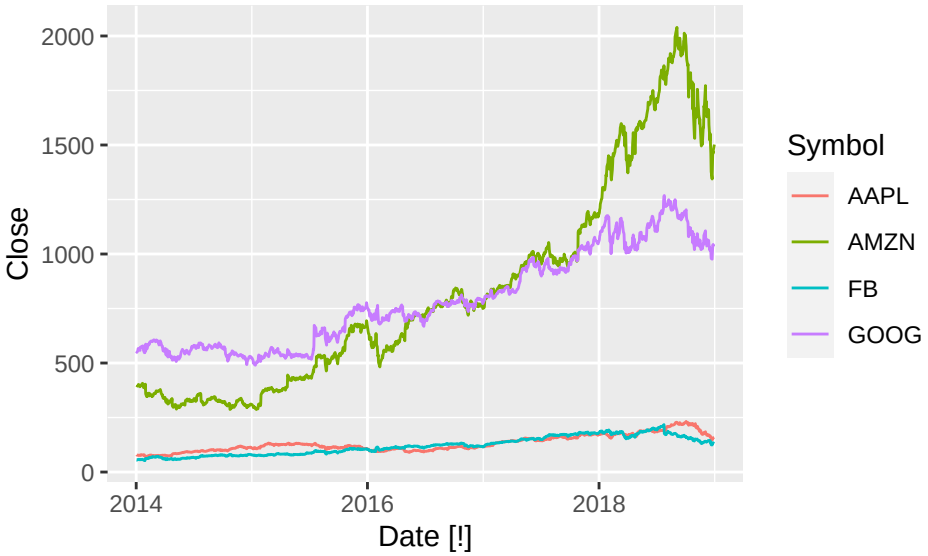
```
pelt %>% autoplot(Lynx)
```



Canadian lynx trappings are cyclic, as the spacing between the peaks is irregular but approximately 10 years.

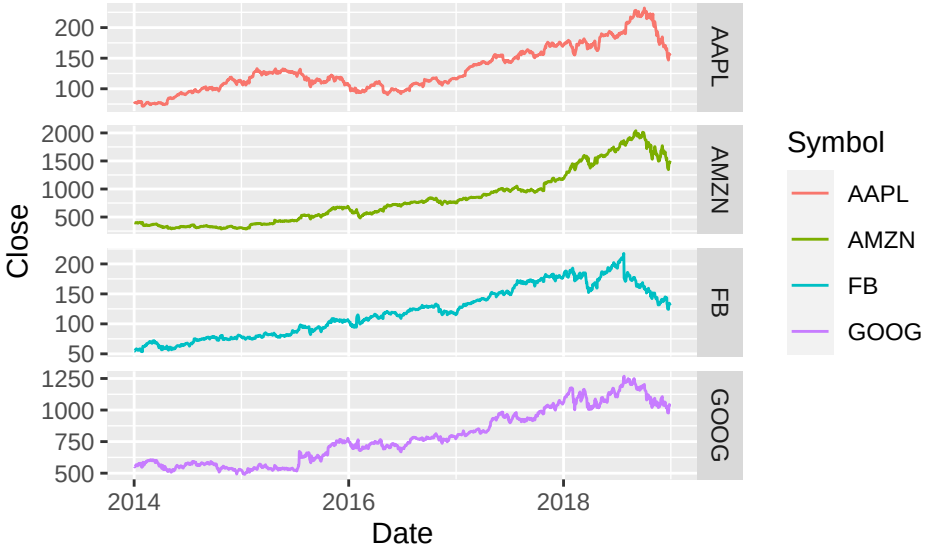
Close

```
gafa_stock %>% autoplot(Close)
```



The four stocks are on different scales, so they are not directly comparable. A plot with faceting would be better.

```
gafa_stock %>%
  ggplot(aes(x=Date, y=Close, group=Symbol)) +
  geom_line(aes(col=Symbol)) +
  facet_grid(Symbol ~ ., scales='free')
```

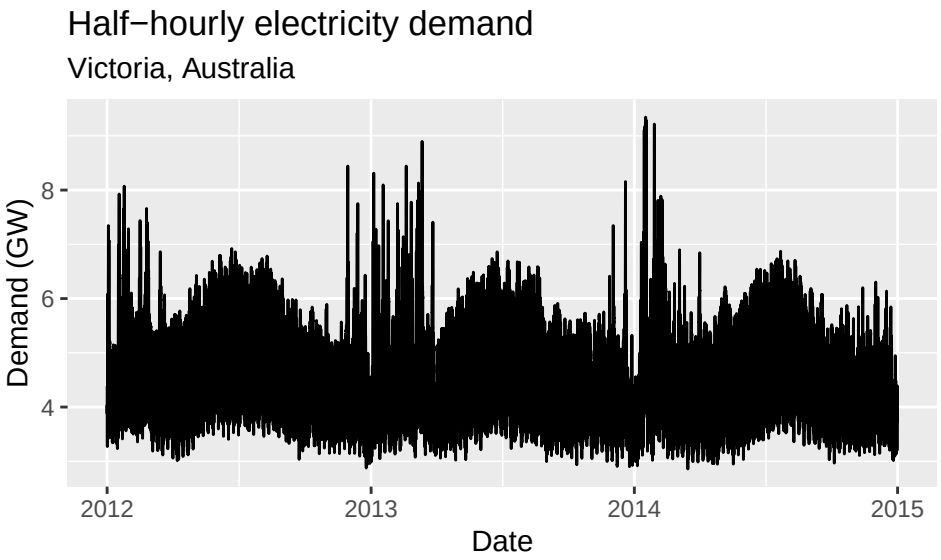


The downturn in the second half of 2018 is now very clear, with Facebook taking a big drop (about 20%) in the middle of the year.

The stocks tend to move roughly together, as you would expect with companies in the same industry.

Demand

```
vic_elec %>% autoplot(Demand/1e3) +
  labs(
    x = "Date",
    y = "Demand (GW)",
    title = "Half-hourly electricity demand",
    subtitle = "Victoria, Australia"
  )
```

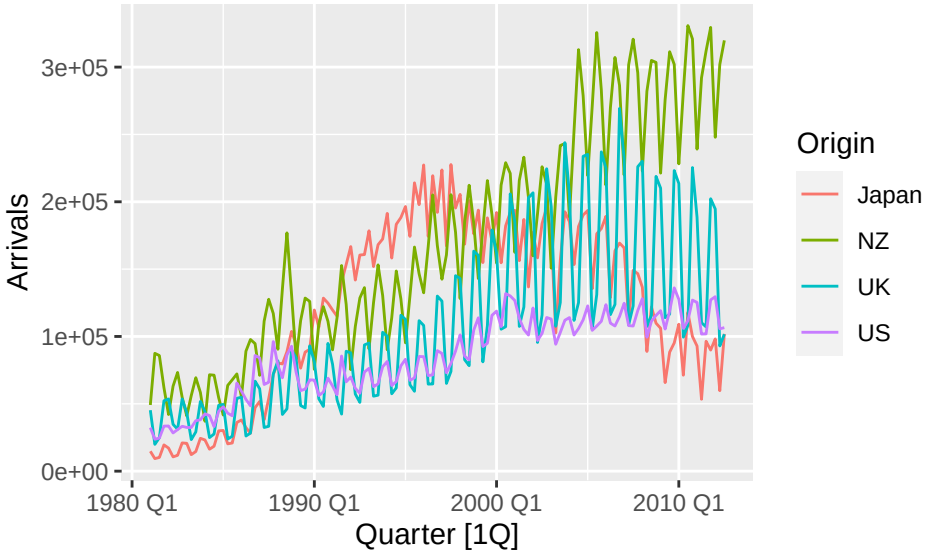


Here the annual seasonality is clear, with high volatility in summer, and peaks in summer and winter. The weekly seasonality is also visible, but the daily seasonality is hidden due to the compression on the horizontal axis.

Exercise 6

The `aus_arrivals` data set comprises quarterly international arrivals (in thousands) to Australia from Japan, New Zealand, UK and the US. Use `autoplot()`, `gg_season()` and `gg_subseries()` to compare the differences between the arrivals from these four countries. Can you identify any unusual observations?

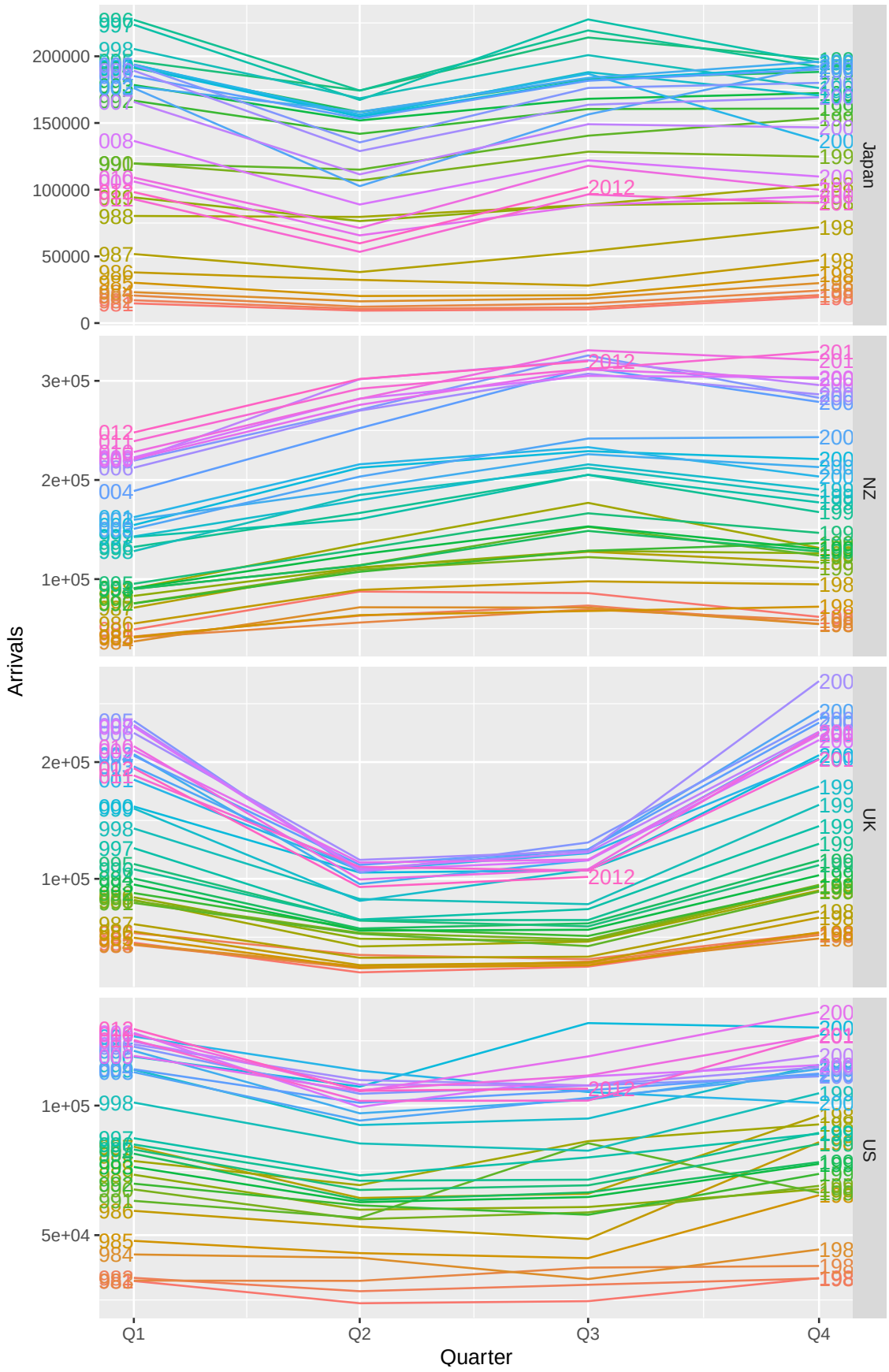
```
aus_arrivals %>% autoplot(Arrivals)
```



Generally the number of arrivals to Australia is increasing over the entire series, with the exception of Japanese visitors which begin to decline after 1995. The series appear to have a seasonal pattern

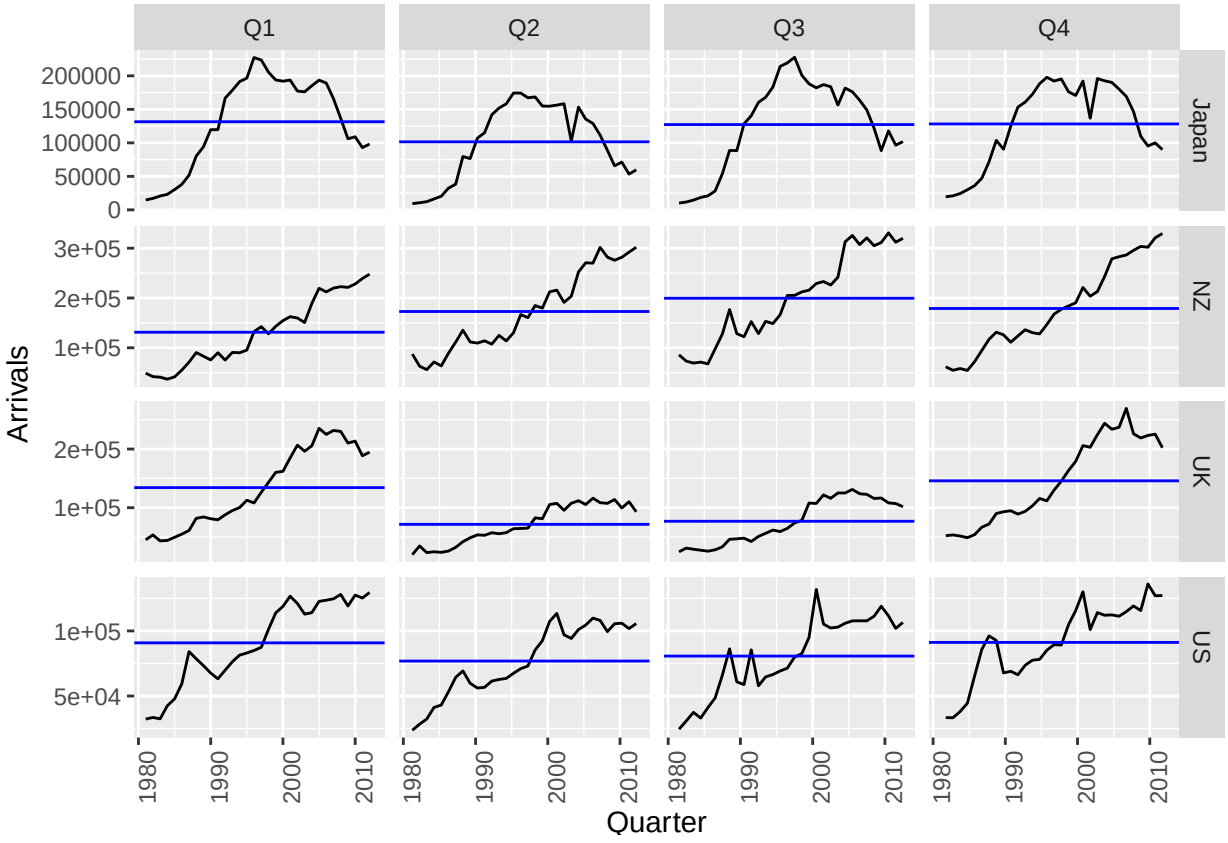
which varies proportionately to the number of arrivals. Interestingly, the number of visitors from NZ peaks sharply in 1988. The seasonal pattern from Japan appears to change substantially.

```
aus_arrivals %>% gg_season(Arrivals, labels = "both")
```


Arrivals
Quarter
Q1
Q2
Q3
Q4
Japan
NZ
UK
US
2012
0
50000
100000
150000
200000
1e+05
2e+05
3e+05
5e+04

The seasonal pattern of arrivals appears to vary between each country. In particular, arrivals from the UK appears to be lowest in Q2 and Q3, and increase substantially for Q4 and Q1. Whereas for NZ visitors, the lowest period of arrivals is in Q1, and highest in Q3. Similar variations can be seen for Japan and US.

```
aus_arrivals %>% gg_subseries(Arrivals)
```



The subseries plot reveals more interesting features. It is evident that whilst the UK arrivals is increasing, most of this increase is seasonal. More arrivals are coming during Q1 and Q4, whilst the increase in Q2 and Q3 is less extreme. The growth in arrivals from NZ and US appears fairly similar across all quarters. There exists an unusual spike in arrivals from the US in 1992 Q3.

Unusual observations:

- 2000 Q3: Spikes from the US (Sydney Olympics arrivals)
- 2001 Q3-Q4 are unusual for US (9/11 effect)
- 1991 Q3 is unusual for the US (Gulf war effect?)

Exercise 7

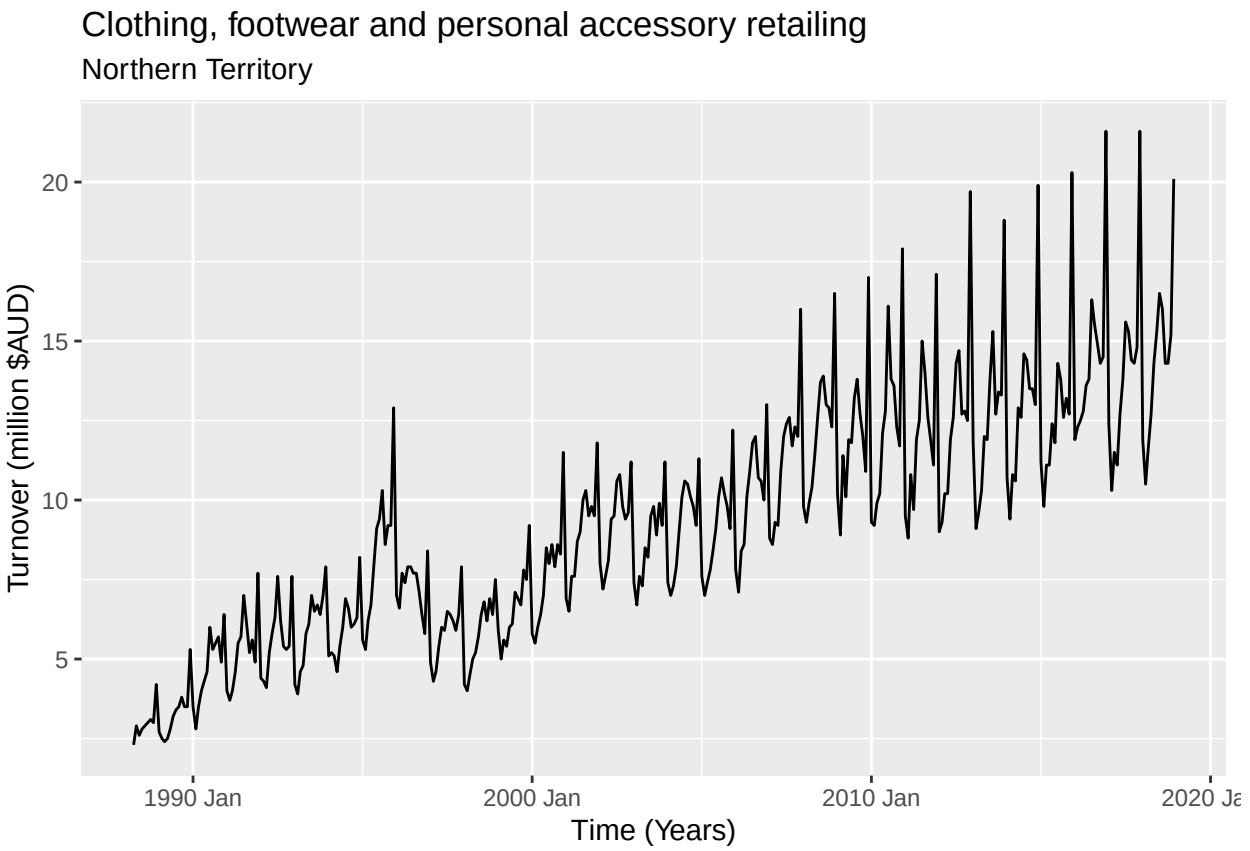
Monthly Australian retail data is provided in aus_retail. Select one of the time series as follows (but choose your own seed value):

```
set.seed(12345678)
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()
```

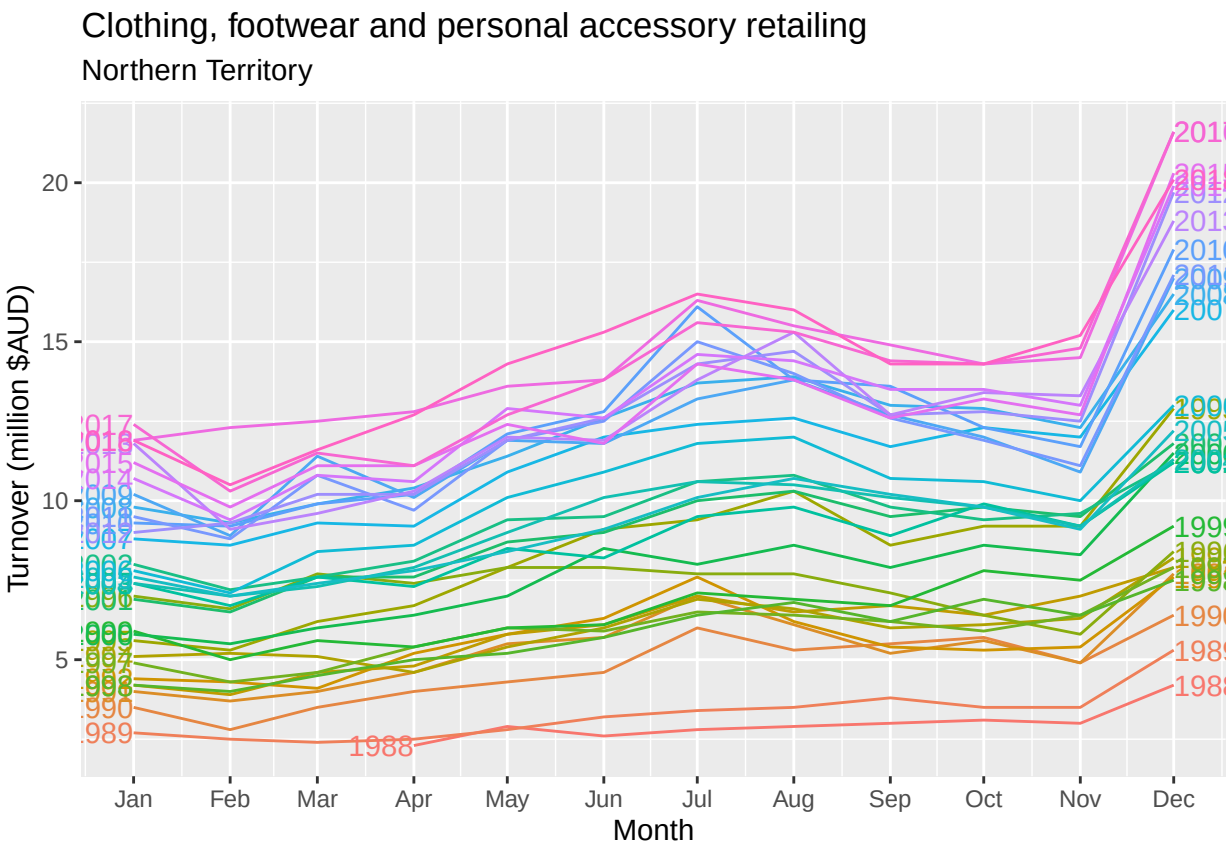
```
set.seed(12345678)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
myseries %>%
  autoplot(Turnover) +
  labs(y = "Turnover (million $AUD)", x = "Time (Years)",
       title = myseries$Industry[1],
       subtitle = myseries$State[1])
```



The data features a non-linear upward trend and a strong seasonal pattern. The variability in the data appears proportional to the amount of turnover (level of the series) over the time period.

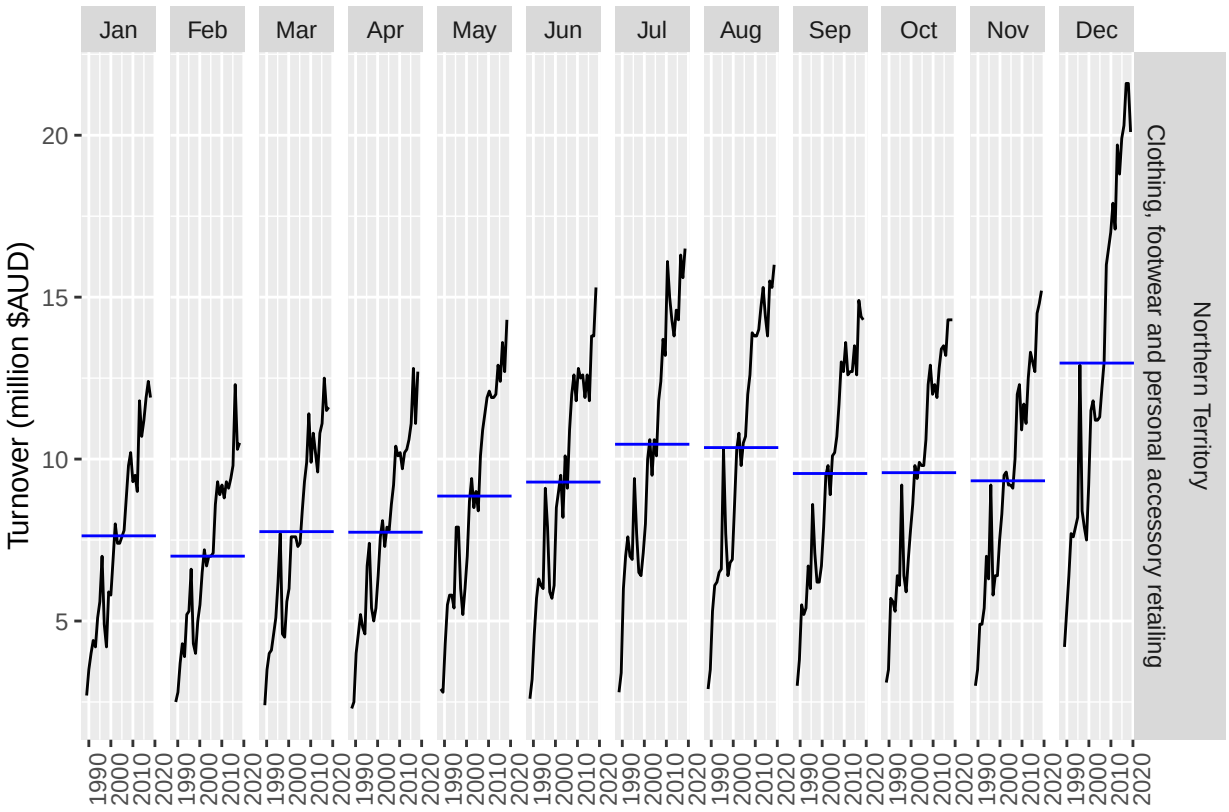
```
myseries %>%
  gg_season(Turnover, labels = "both") +
```

```
  labs(y = "Turnover (million $AUD)",
       title = myseries$Industry[1],
       subtitle = myseries$State[1])
```



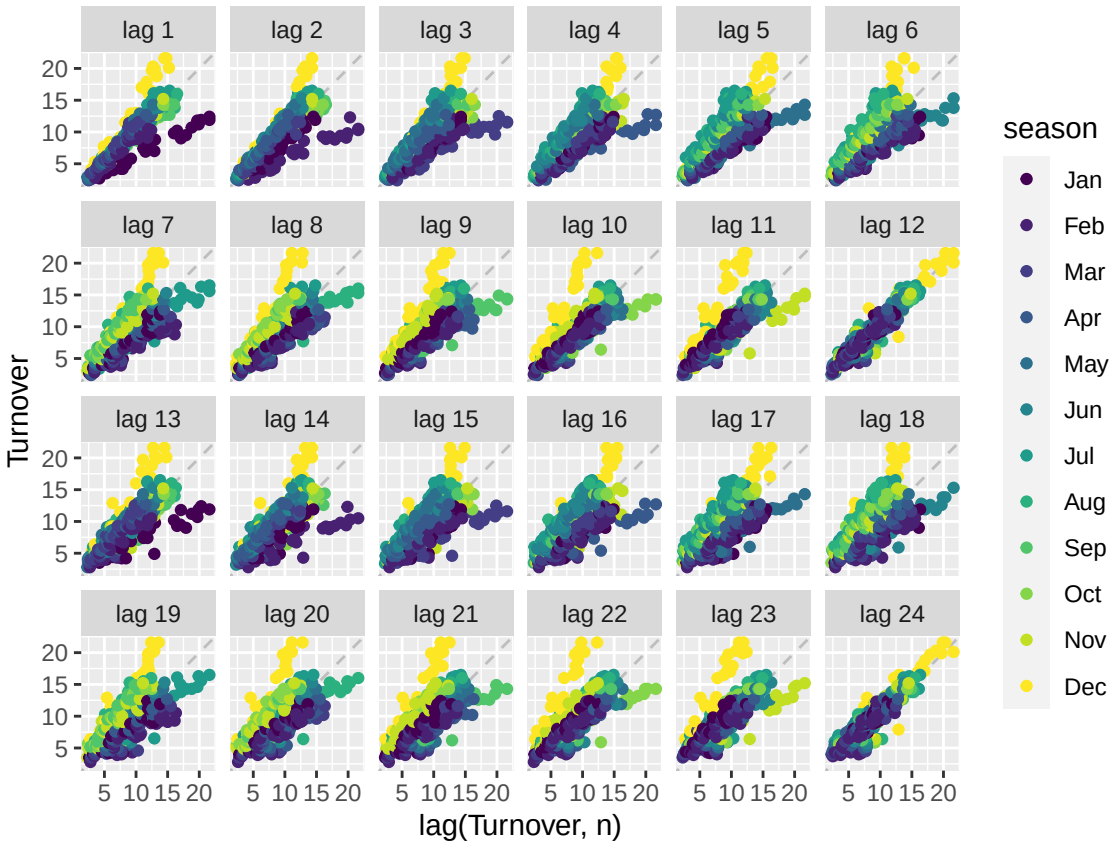
Strong seasonality is evident in the season plot. Large increases in clothing retailing can be observed in December (probably a Christmas effect). There is also a peak in July that appears to be getting stronger over time. 2016 had an unusual pattern in the first half of the year.

```
myseries %>%
  gg_subseries(Turnover) +
  labs(y = "Turnover (million $AUD)", x="")
```

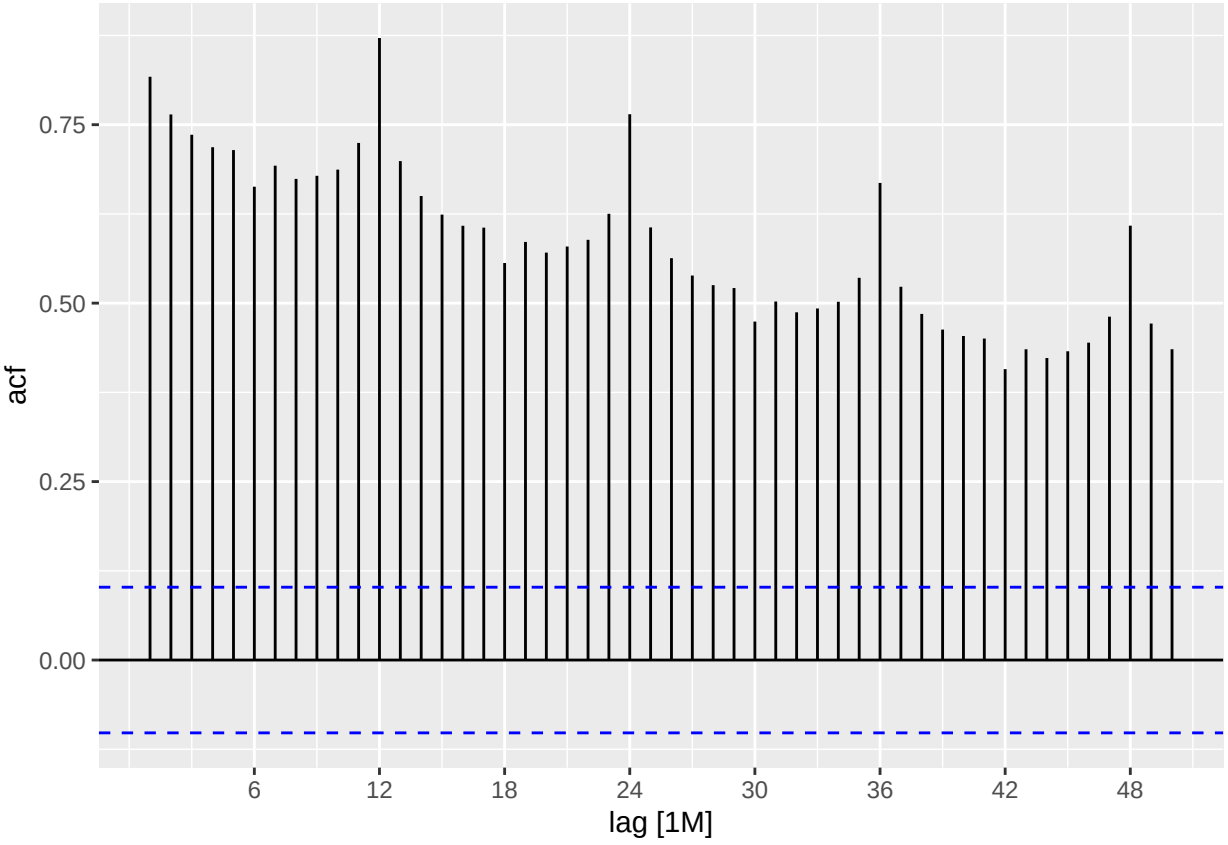


There is a strong trend in all months, with the largest trend in December and a larger increase in July and August than most other months.

```
myseries %>%
  gg_lag(Turnover, lags=1:24, geom='point') + facet_wrap(~ .lag, ncol=6)
```




```
myseries %>%
  ACF(Turnover, lag_max = 50) %>%
  autoplot()
```



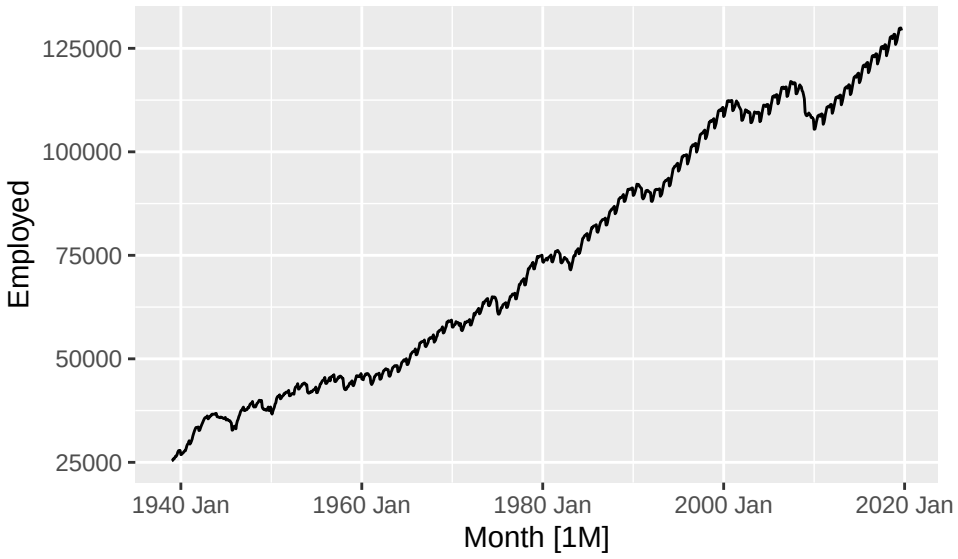
Exercise 8

Use the following graphics functions: `autoplot()`, `gg_season()`, `gg_subseries()`, `gg_lag()`, `ACF()` and explore features from the following time series: "Total Private" `Employed` from `us_employment`, `Bricks` from `aus_production`, `Hare` from `pelt`, "H02" `Cost` from `PBS`, and `us_gasoline`.

- Can you spot any seasonality, cyclicity and trend?
- What do you learn about the series?
- What can you say about the seasonal patterns?
- Can you identify any unusual years?

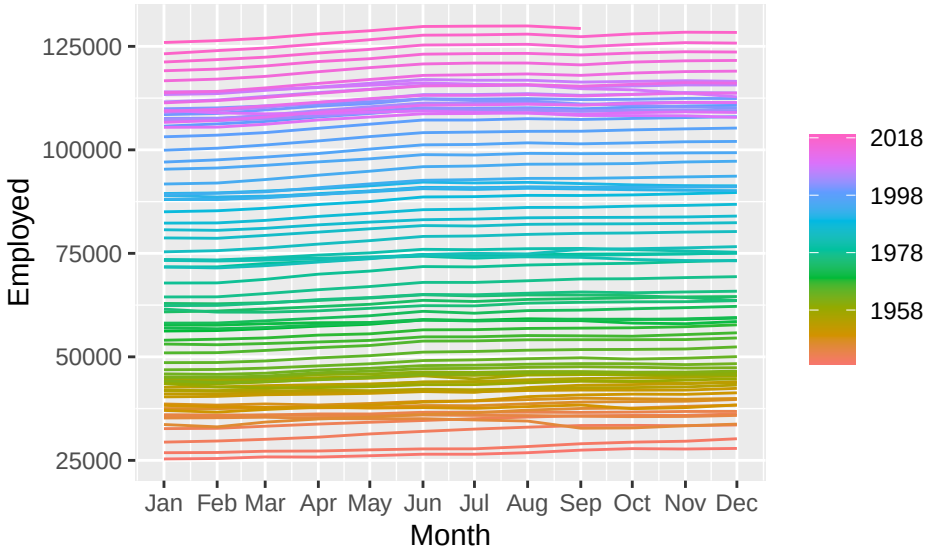
Total Private Employment in the US

```
us_employment %>%
  filter(Title == "Total Private") %>%
  autoplot(Employed)
```

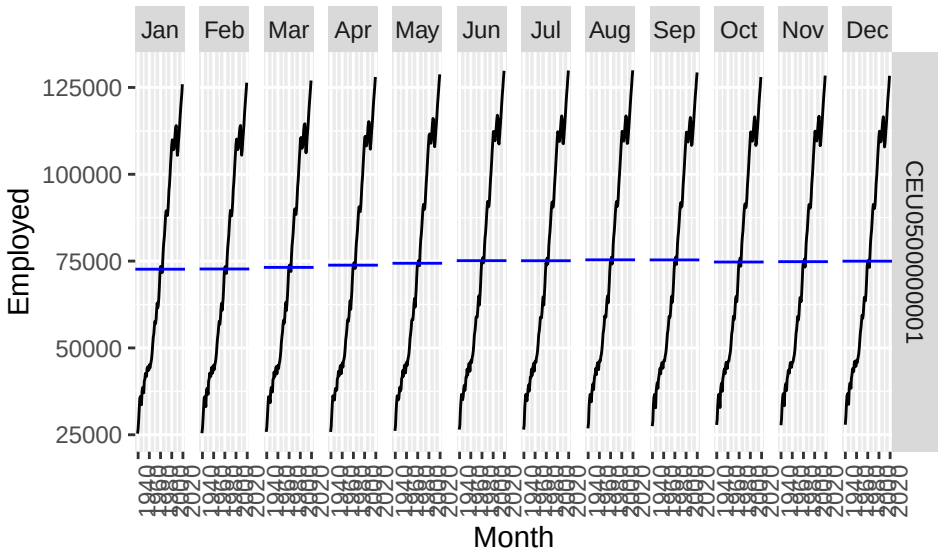


There is a strong trend and seasonality. Some cyclic behaviour is seen, with a big drop due to the global financial crisis.

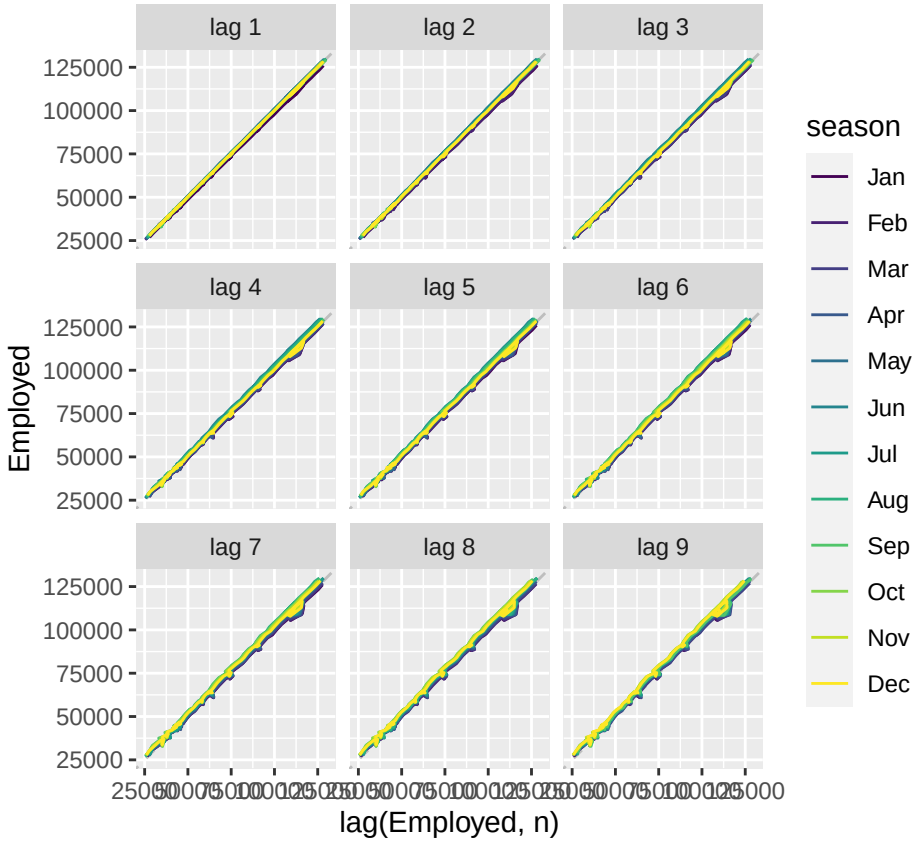
```
us_employment %>%
  filter(Title == "Total Private") %>%
  gg_season(Employed)
```




```
us_employment %>%
  filter(Title == "Total Private") %>%
  gg_subseries(Employed)
```

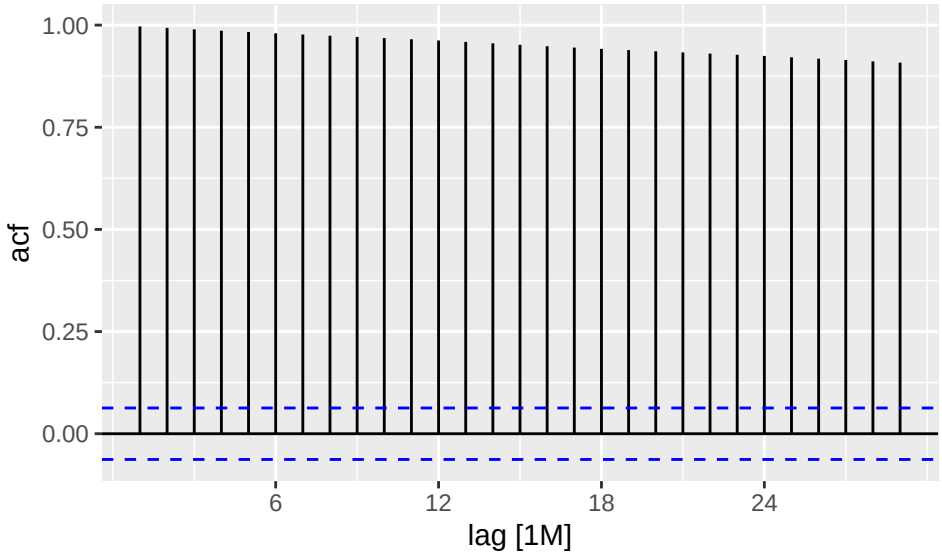



```
us_employment %>%
  filter(Title == "Total Private") %>%
  gg_lag(Employed)
```




```
us_employment %>%
  filter(Title == "Total Private") %>%
```

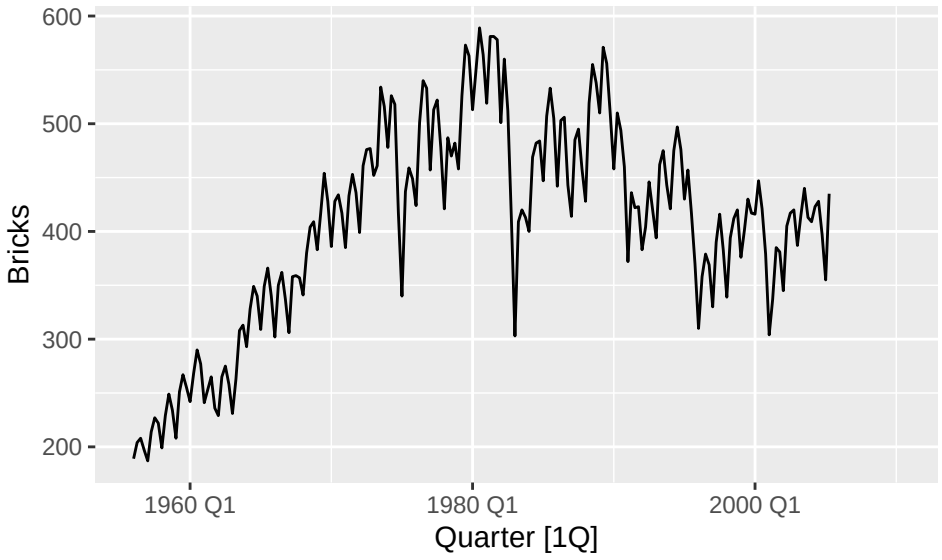
```
ACF(Employed) %>%
autoplot()
```



In all of these plots, the trend is so dominant that it is hard to see anything else. We need to remove the trend so we can explore the other features of the data.

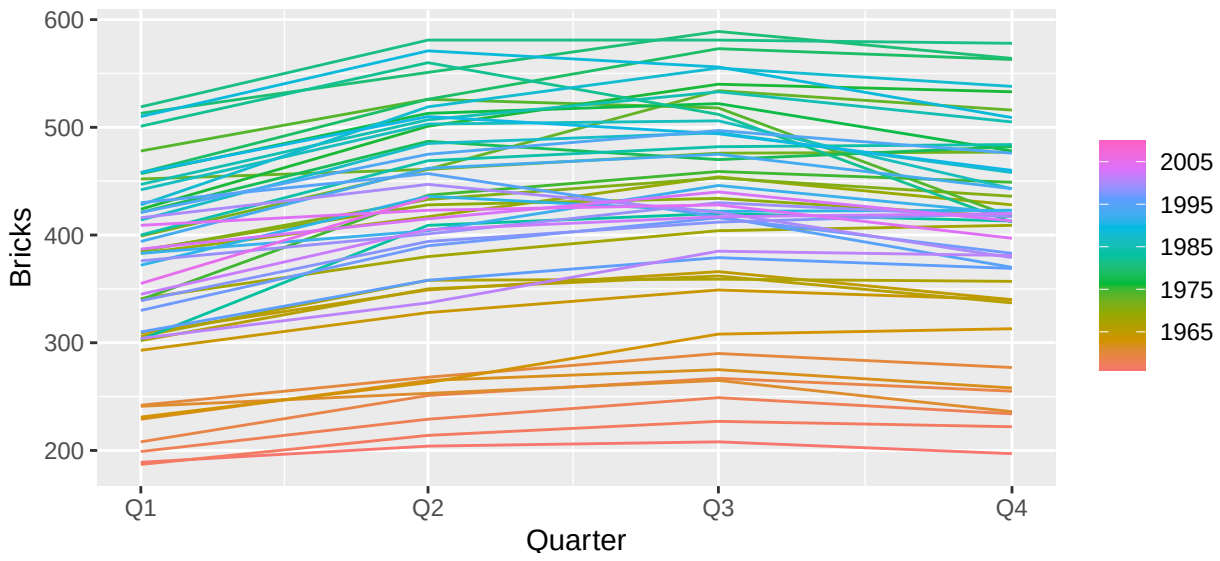
Brick production in Australia

```
aus_production %>%
  autoplot(Bricks)
```



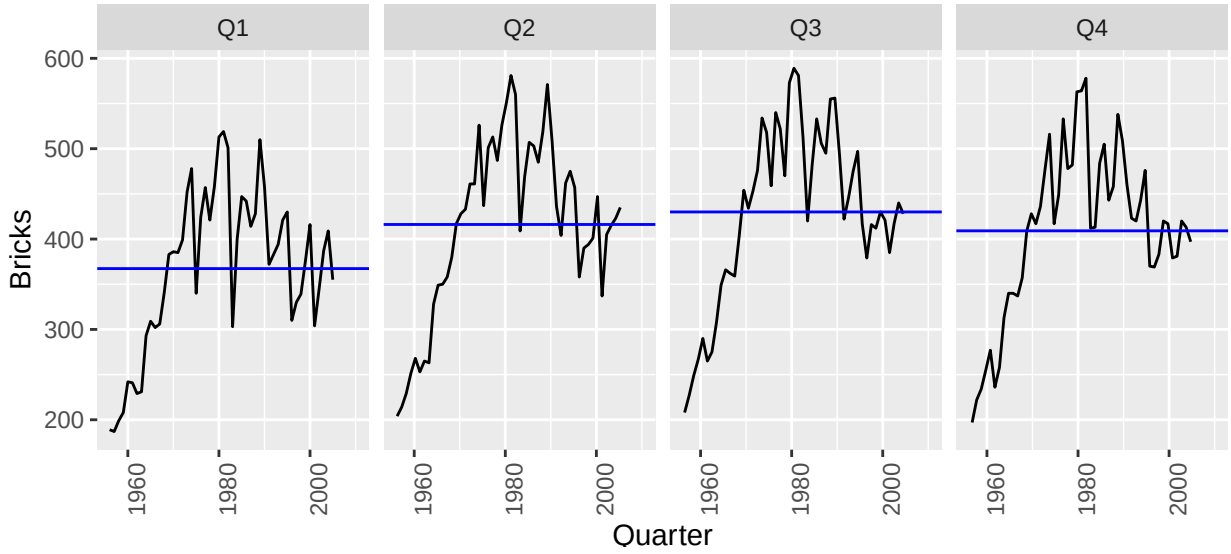
A positive trend in the first 20 years, and a negative trend in the next 25 years. Strong quarterly seasonality, with some cyclicity – note the recessions in the 1970s and 1980s.

```
aus_production %>%
  gg_season(Bricks)
```



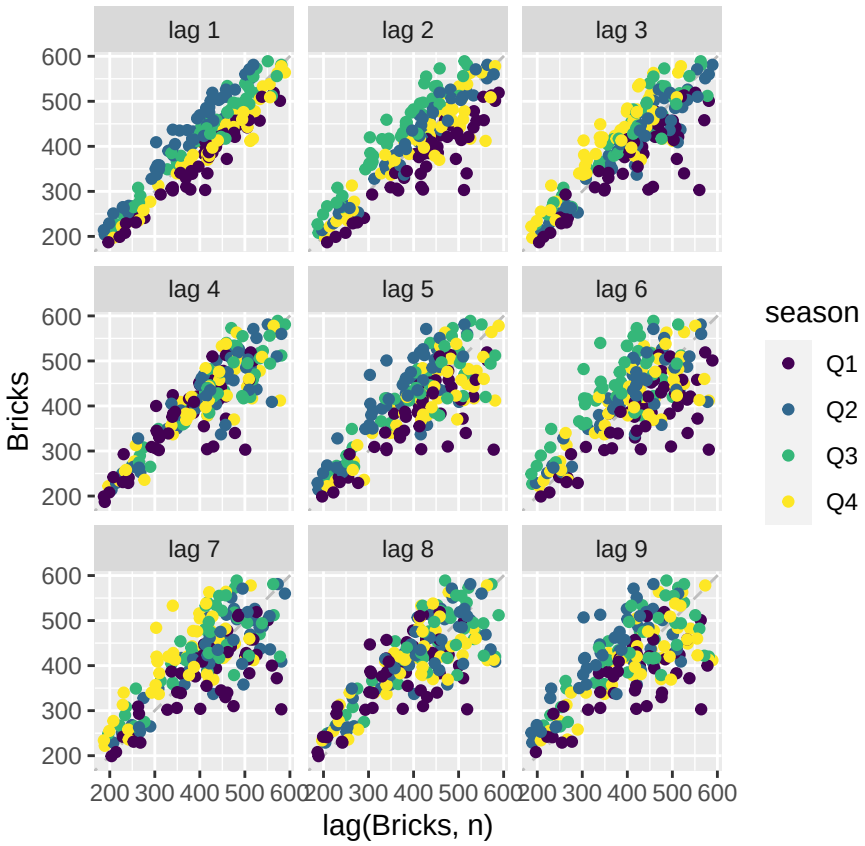
Brick production tends to be lowest in the first quarter and peak in either quarter 2 or quarter 3.

```
aus_production %>%
  gg_subseries(Bricks)
```

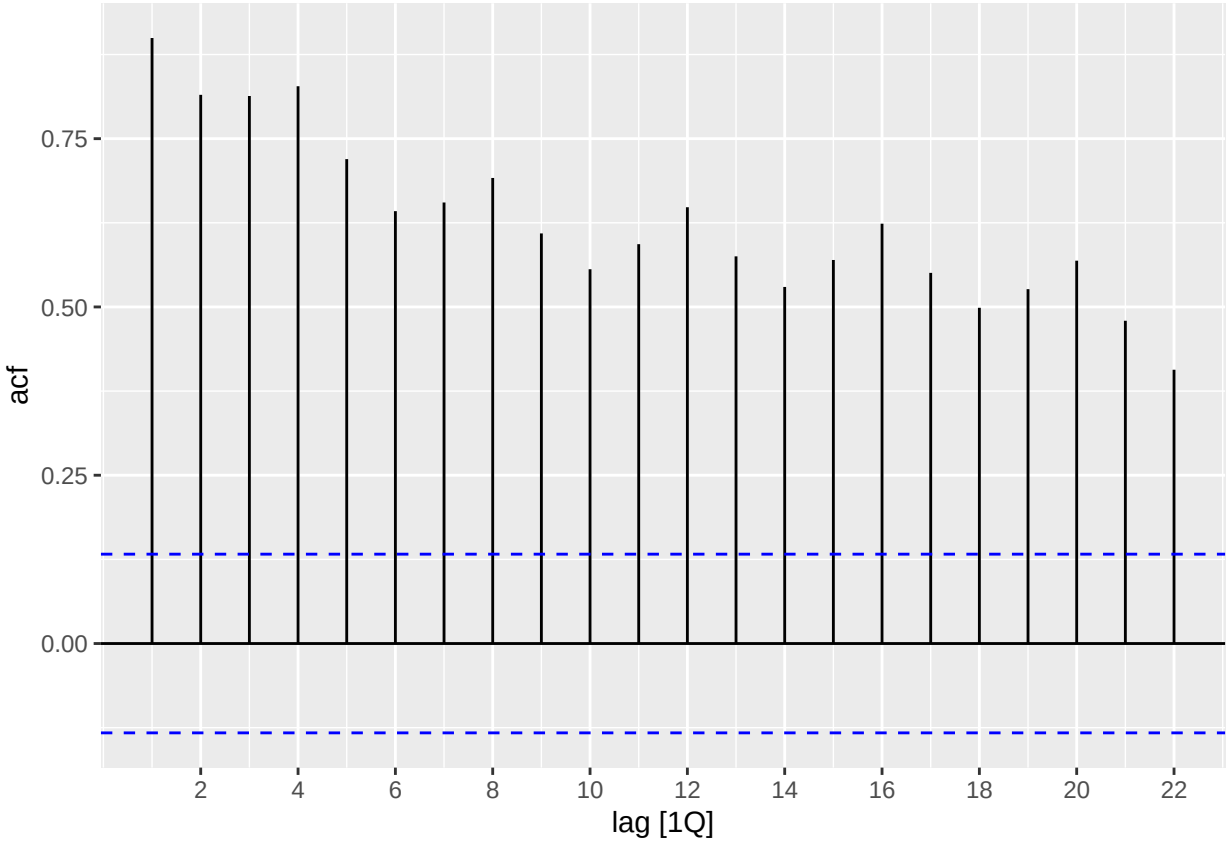


The decrease in the last 25 years has been weakest in Q1.

```
aus_production %>%
  gg_lag(Bricks, geom='point')
```



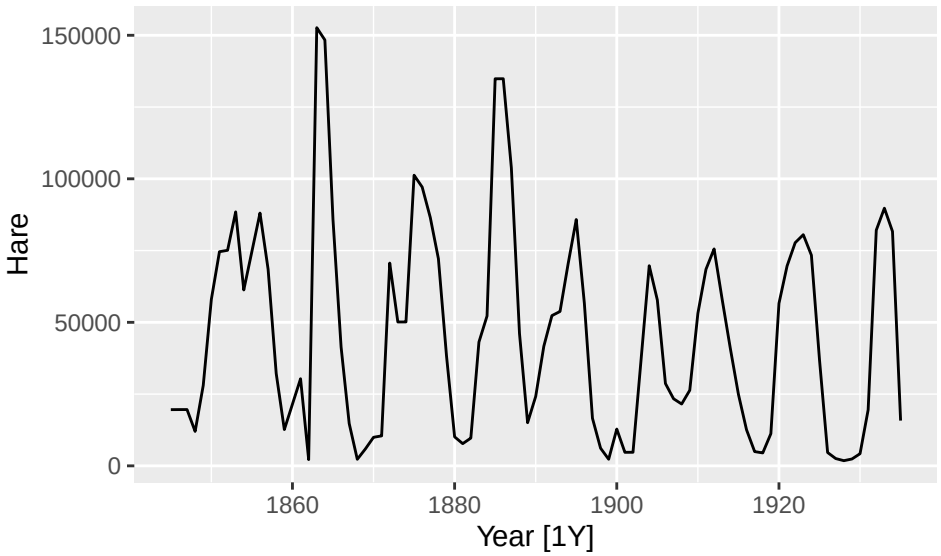

```
aus_production %>%
  ACF(Bricks) %>% autoplot()
```



The seasonality shows up as peaks at lags 4, 8, 12, 16, 20, ... The trend is seen with the slow decline on the positive side.

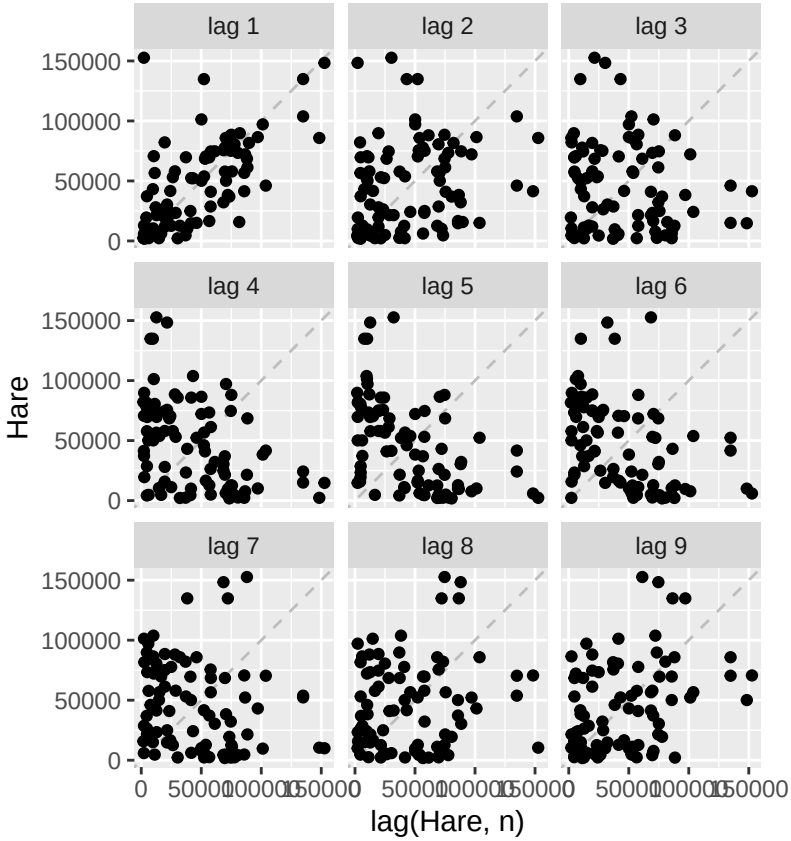
##3 Snow hare trappings in Canada

```
pelt %>%
  autoplot(Hare)
```

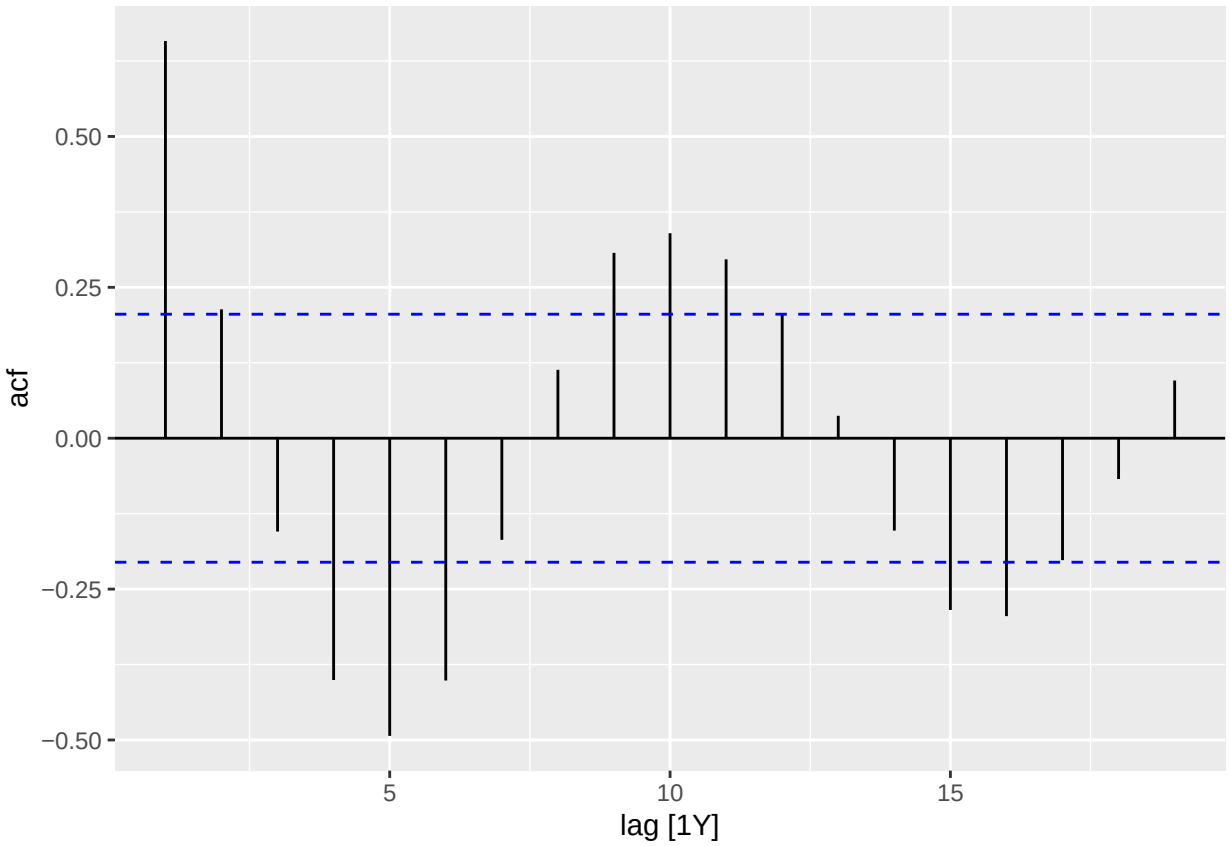


There is some cyclic behaviour with substantial variation in the length of the period.

```
pelt %>%
  gg_lag(Hare, geom='point')
```




```
pelt %>%
  ACF(Hare) %>% autoplot()
```



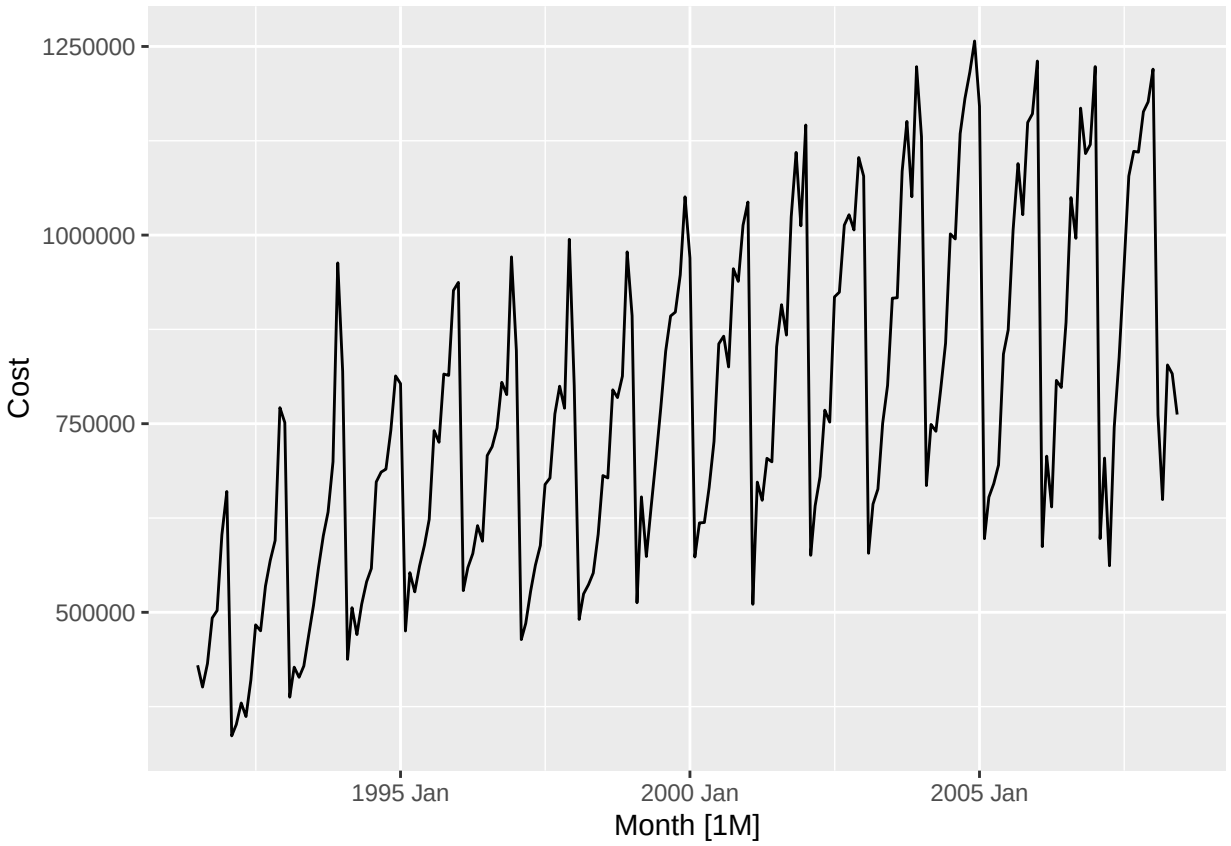
The cyclic period seems to have an average of about 10 (due to the local maximum in ACF at lag 10).

H02 sales in Australia

There are four series corresponding to H02 sales, so we will add them together.

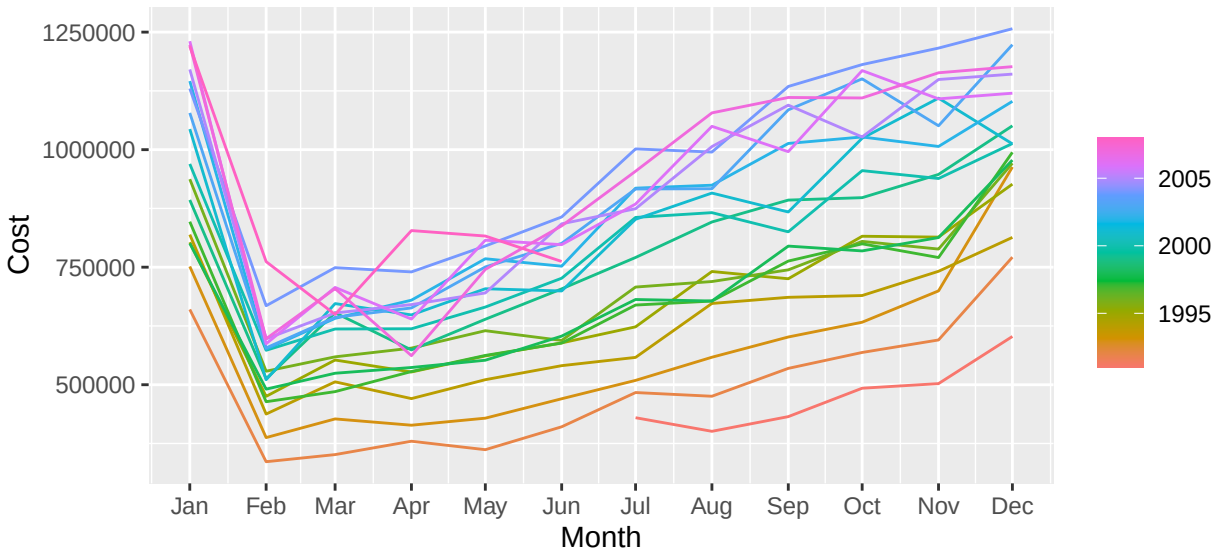
```
h02 <- PBS %>%
  filter(ATC2 == "H02") %>%
  group_by(ATC2) %>%
  summarise(Cost = sum(Cost)) %>%
  ungroup()
```

```
h02 %>%
  autoplot(Cost)
```

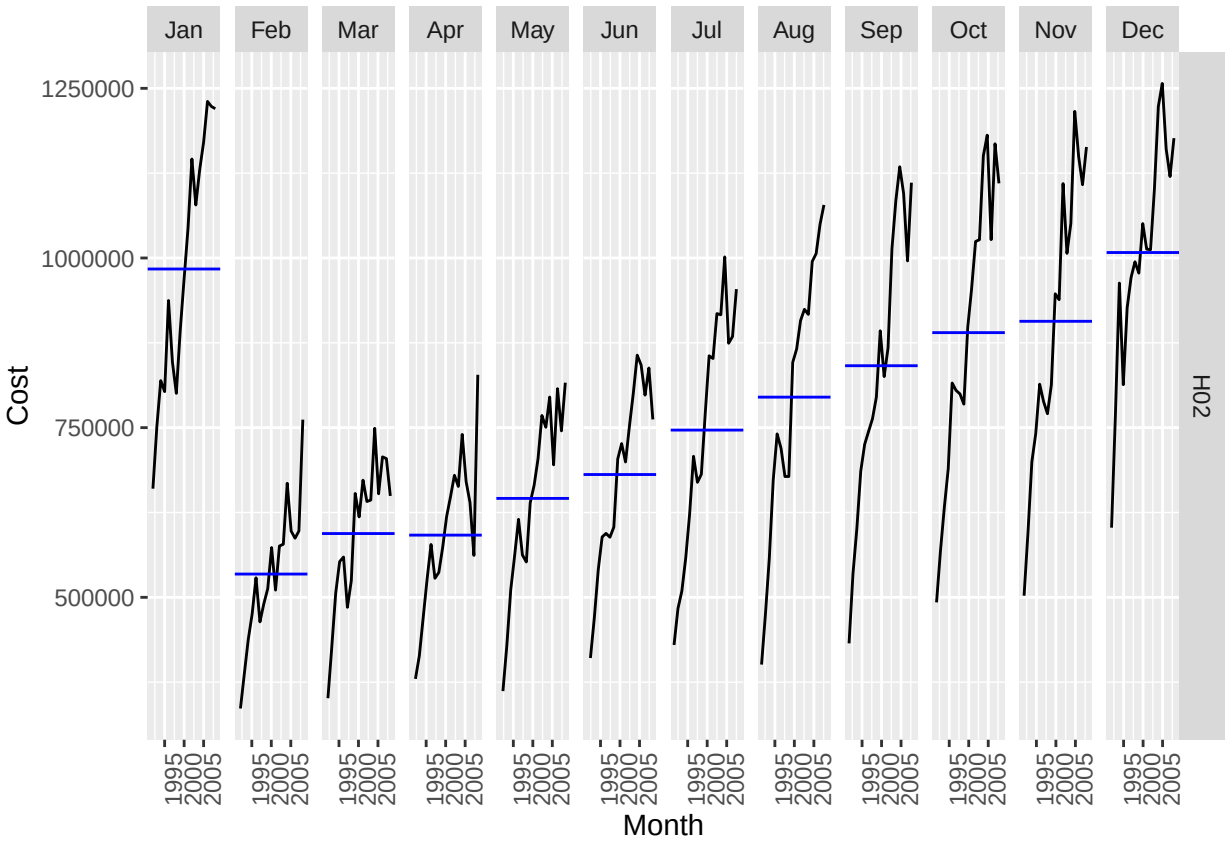


A positive trend with strong monthly seasonality, dropping suddenly every February.

```
h02 %>%
  gg_season(Cost)
```

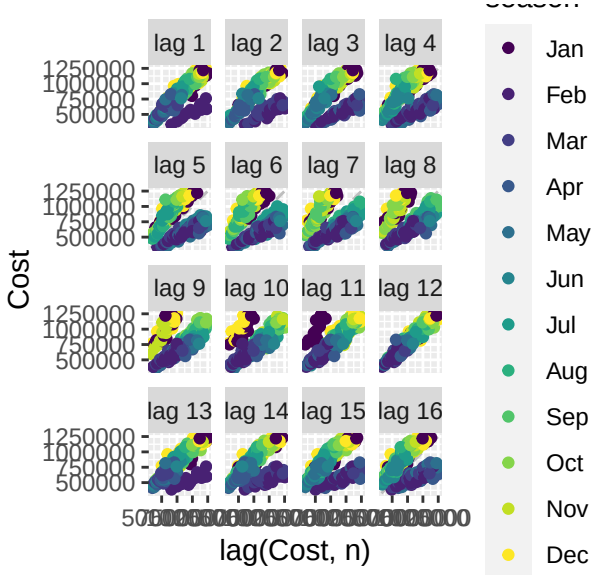



```
h02 %>%
  gg_subseries(Cost)
```

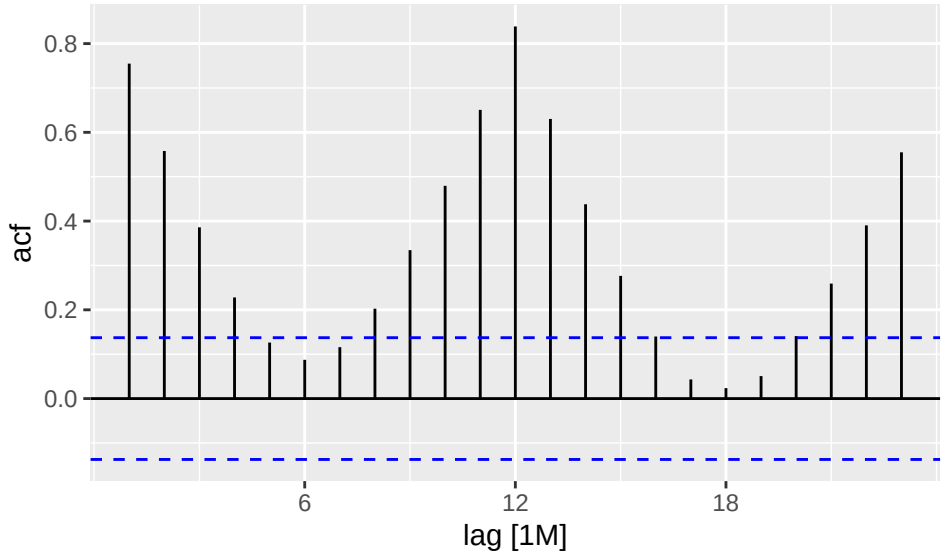


The trends have been greater in the higher peaking months – this leads to increasing seasonal variation.

```
h02 %>%
  gg_lag(Cost, geom='point', lags=1:16)
```



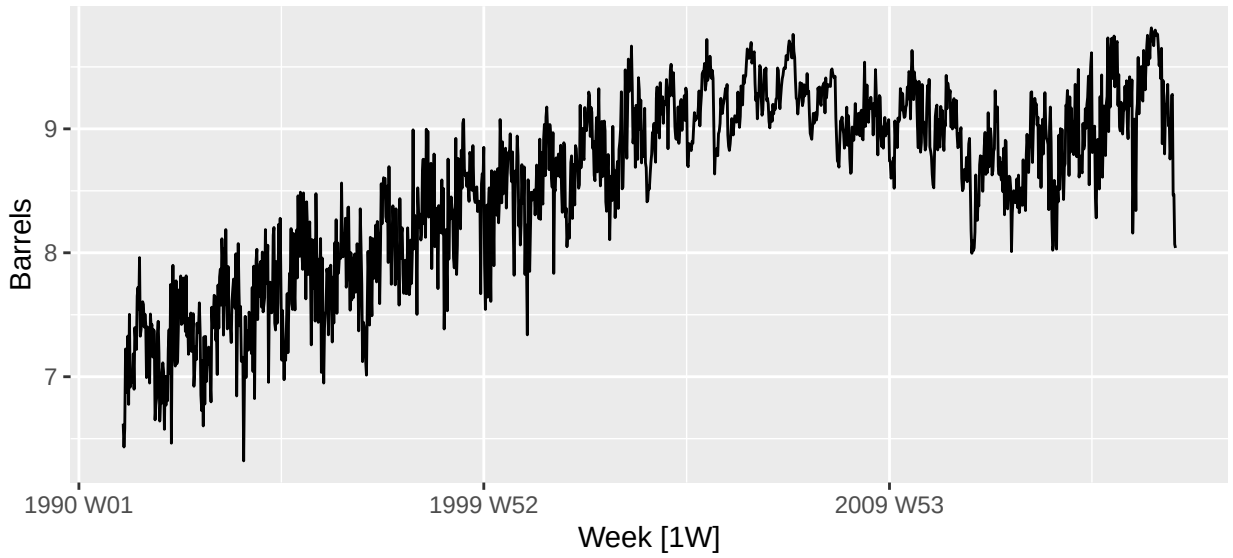
```
h02 %>%
  ACF(Cost) %>% autoplot()
```



The large January sales show up as a separate cluster of points in the lag plots. The strong seasonality is clear in the ACF plot.

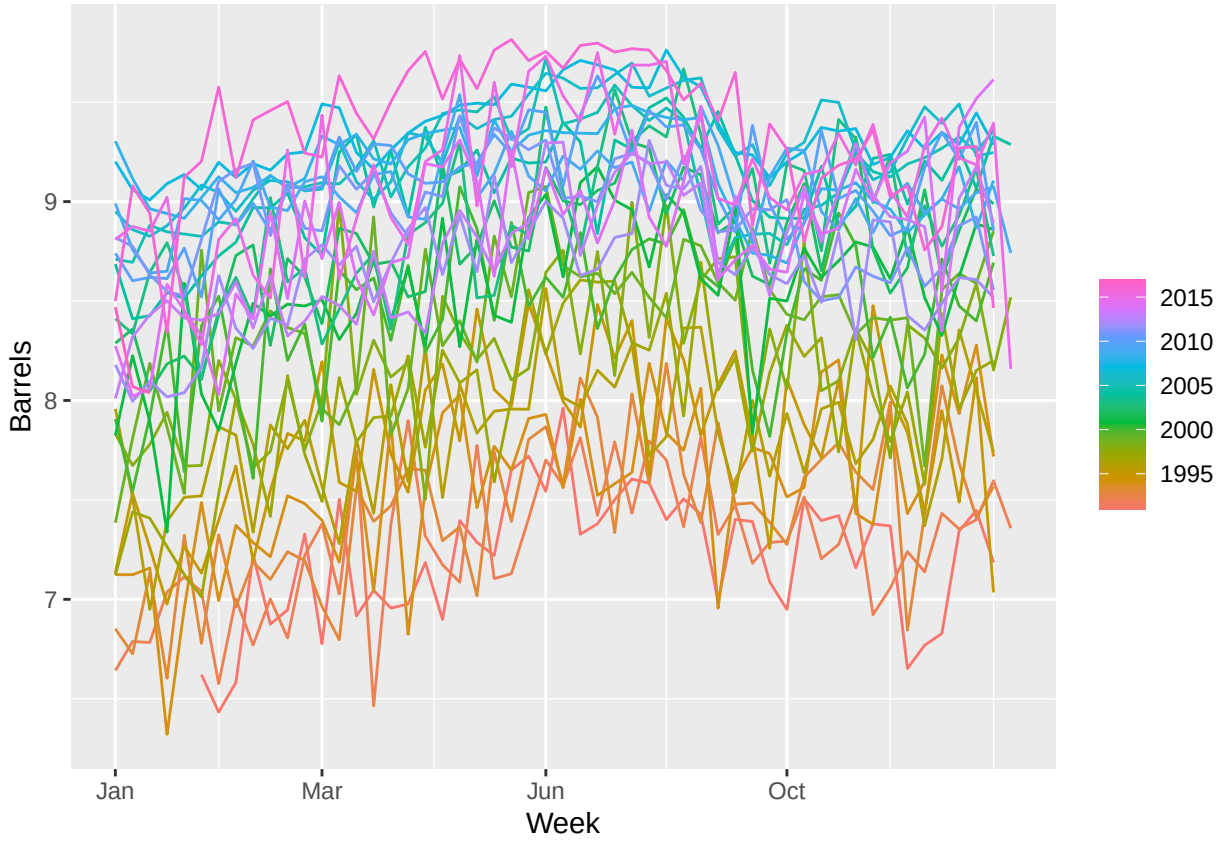
US gasoline sales

```
us_gasoline %>%
  autoplot(Barrels)
```



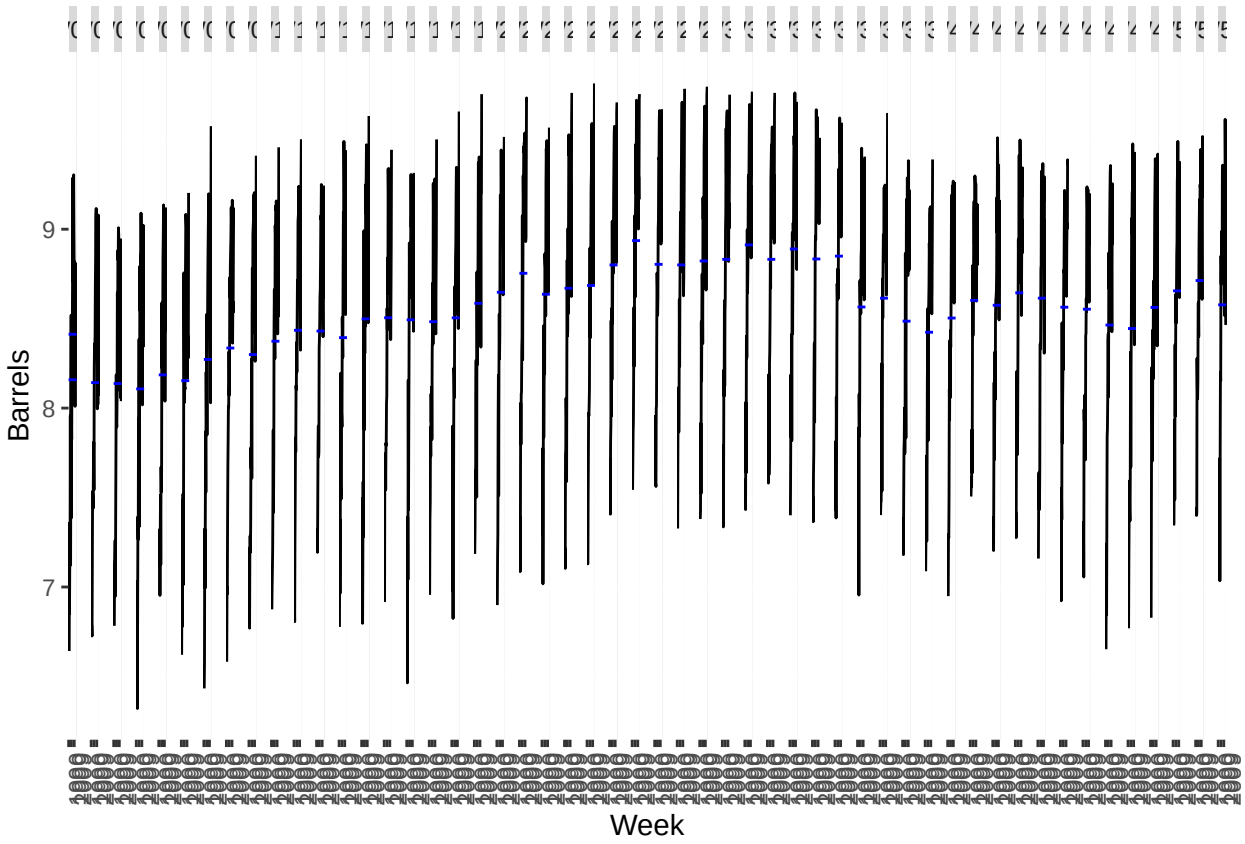
A positive trend until 2008, and then the global financial crisis led to a drop in sales until 2012. The shape of the seasonality seems to have changed over time.

```
us_gasoline %>%
  gg_season(Barrels)
```



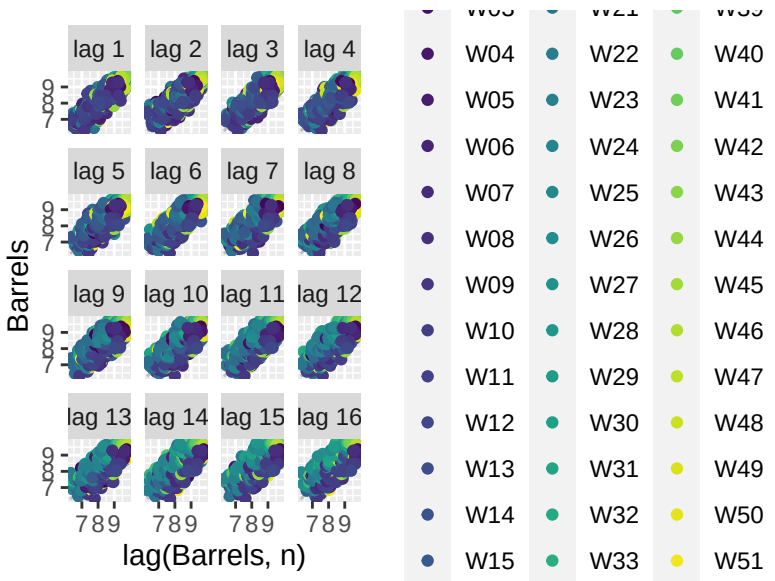
There is a lot of noise making it hard to see the overall seasonal pattern. However, it seems to drop towards the end of quarter 4.

```
us_gasoline %>%
  gg_subseries(Barrels)
```

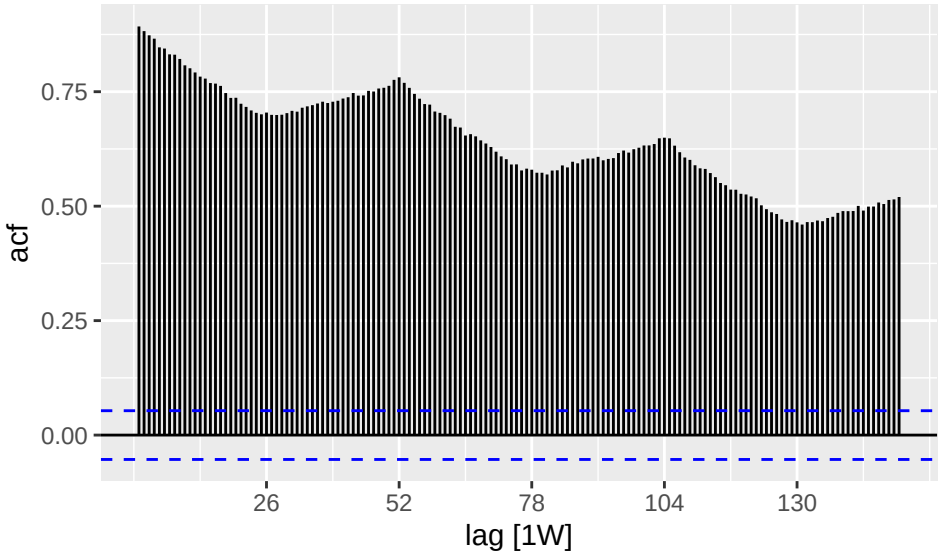


The blue lines are helpful in seeing the average seasonal pattern.

```
us_gasoline %>%
  gg_lag(Barrels, geom='point', lags=1:16)
```



```
us_gasoline %>%
  ACF(Barrels, lag_max = 150) %>% autoplot()
```



The seasonality is seen if we increase the lags to at least 2 years (approx 104 weeks)