Homework 1

Forecasting: Principles and Practice - Time Series Graphics

Paul Perez

2/14/2021

```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------------------- fpp3 0.4.0 --

## v tibble      3.0.6     v tsibble     0.9.3
## v dplyr       1.0.4     v tsibbledata 0.2.0
## v tidyr       1.1.2     v feasts      0.1.7
## v lubridate   1.7.4     v fable       0.3.0
## v ggplot2     3.3.3

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

```
library(ggplot2)
```

Exercise 2.10 - 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

The data in `gafa_stock` represents historical stock prices ($USD) for Google, Amazon, Facebook, and Apple from 2014 - 2018.

```
gafa_stock
```

```
## # A tsibble: 5,032 x 8 [1D]
## # 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
```

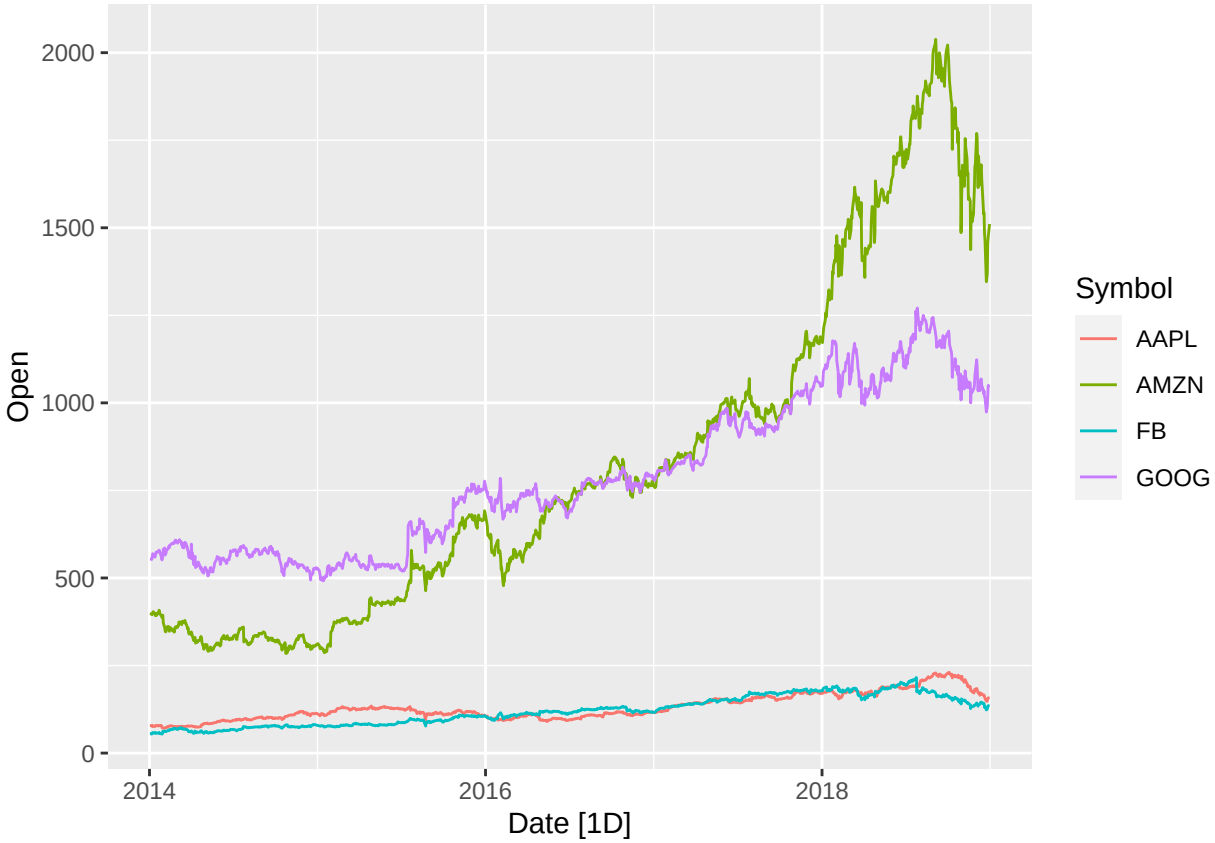
Looking at the `tsibble` object above, we have 5,032 records with 8 values per record. Those are `Symbol`, `Date`, `Open`, `High`, `Low`, `Adj_Close`, and `Volume`. The data [1D] shows that the data is at a daily level, and the Symbol [4] shows that we have four tickers.

```
gafa_stock %>% distinct(Symbol)
```

```
## # A tibble: 4 x 1
##   Symbol
##   <chr>
## 1 AAPL
## 2 AMZN
## 3 FB
## 4 GOOG
```

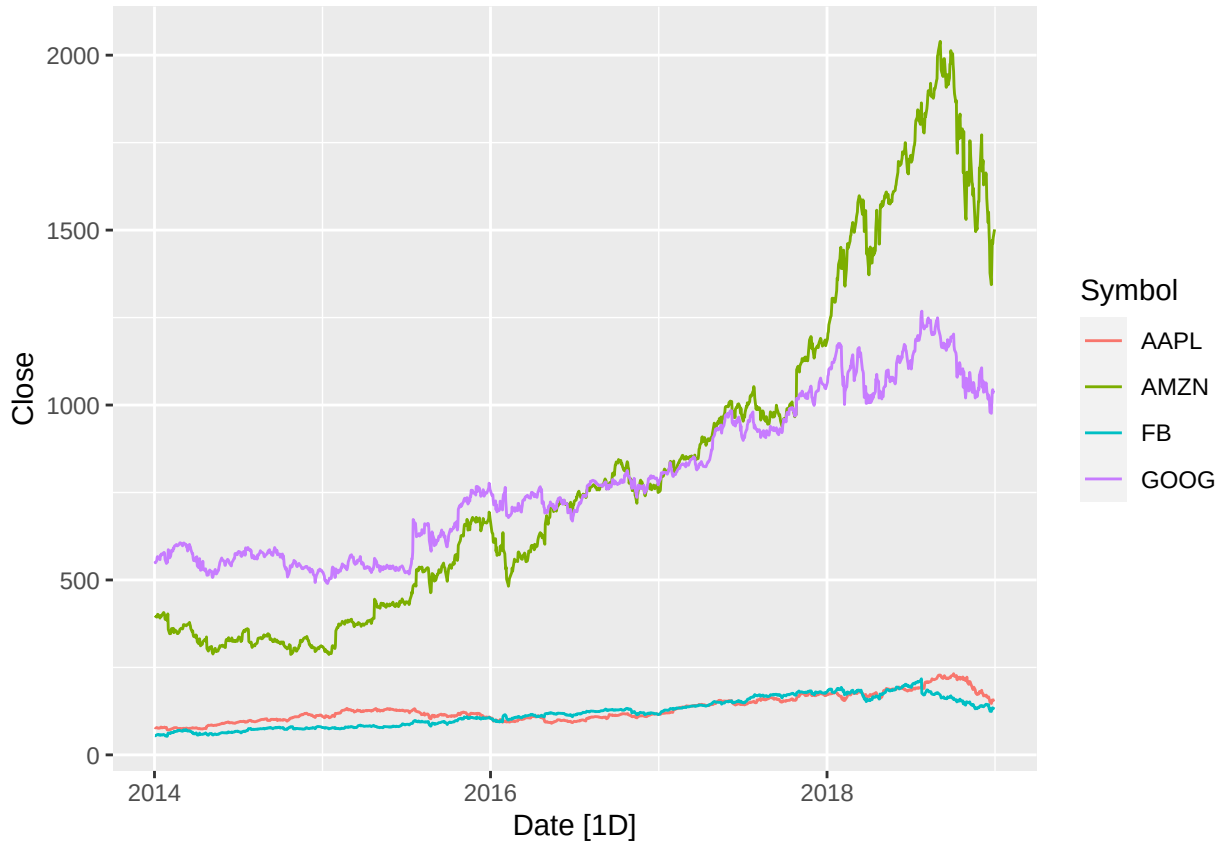
```
autoplot(gafa_stock)
```

```
## Plot variable not specified, automatically selected `.vars = Open`
```



Using `autoplot()` without specifying a variable to plot, we can see the four stock tickers plotted along the same x-axis using the `Open` variable which is the stocks opening price. Below, we can specifically plot the `Close` price by adding it as the second argument in the `autoplot()` function.

```
autoplot(gafa_stock, Close)
```



PBS

The data in 'PBS represents monthly medicare prescription data in Australia.

```
PBS
```

```
## # A tsibble: 65,219 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [336]
##       Month Concession  Type   ATC1  ATC1_desc   ATC2  ATC2_desc   Scripts  Cost
##       <mth> <chr>       <chr>  <chr> <chr>       <chr> <chr>         <dbl> <dbl>
##  1 1991 Jul Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   18228 67877
##  2 1991 Aug Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15327 57011
##  3 1991 Sep Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   14775 55020
##  4 1991 Oct Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15380 57222
##  5 1991 Nov Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   14371 52120
##  6 1991 Dec Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15028 54299
##  7 1992 Jan Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   11040 39753
##  8 1992 Feb Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15165 54405
##  9 1992 Mar Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   16898 61108
## 10 1992 Apr Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   18141 65356
## # ... with 65,209 more rows
```

Looking at the `tsibble` object above, we have 65,219 records with 9 values per record. Those are `Month`, `Concession`, `Type`, `ATC1`, `ATC1_desc`, `ATC2`, `ATC2_desc`, `Scripts`, `Cost`. The data [1M] shows that the data

is at a monthly level, and the Concession, Type, ATC1, and ATC2 [336] shows that we have each of those values have 336 unique values.

Since there are various levels in this nested `tsibble` object, we'll make sure to filter and aggregate this data accordingly.

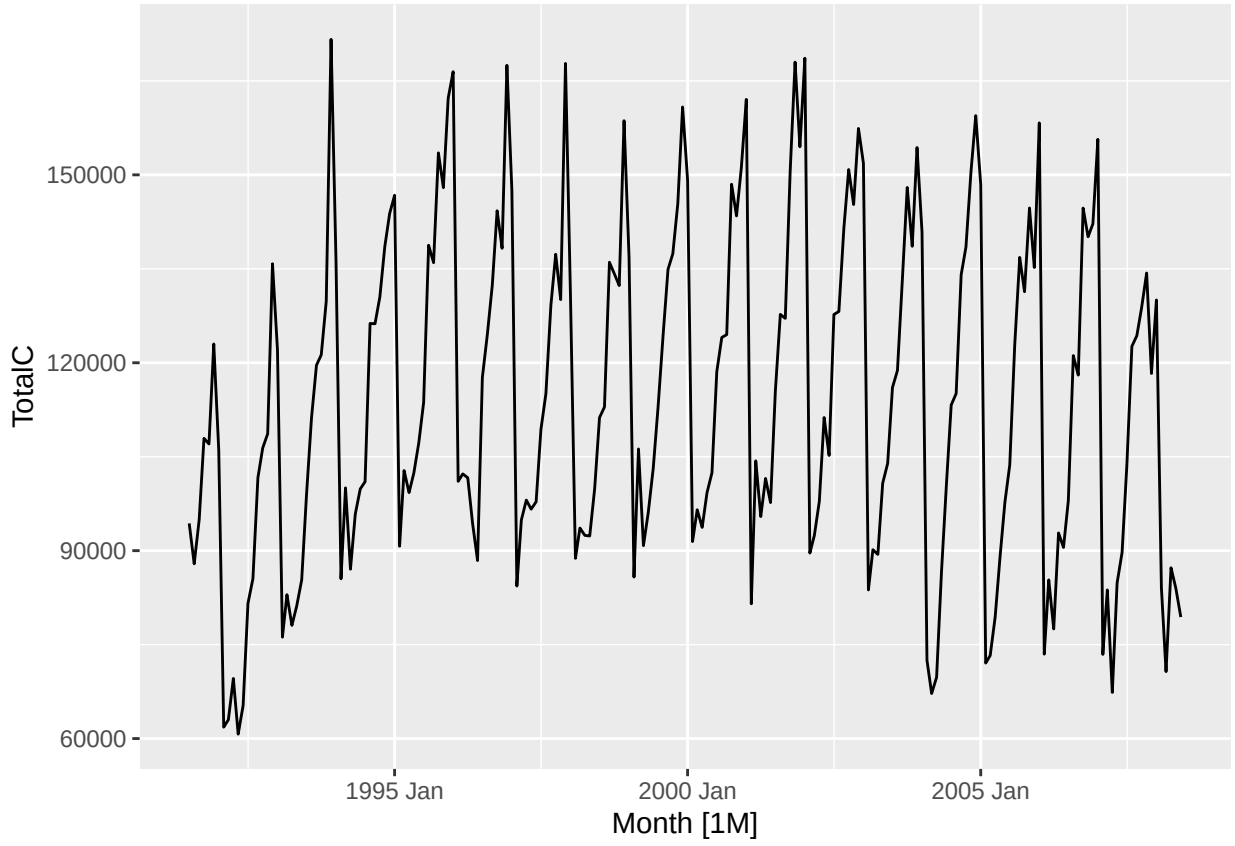
```
PBS %>% distinct(ATC2)
```

```
## # A tibble: 84 x 1
##    ATC2
##    <chr>
##  1 A01
##  2 A02
##  3 A03
##  4 A04
##  5 A05
##  6 A06
##  7 A07
##  8 A09
##  9 A10
## 10 A11
## # ... with 74 more rows
```

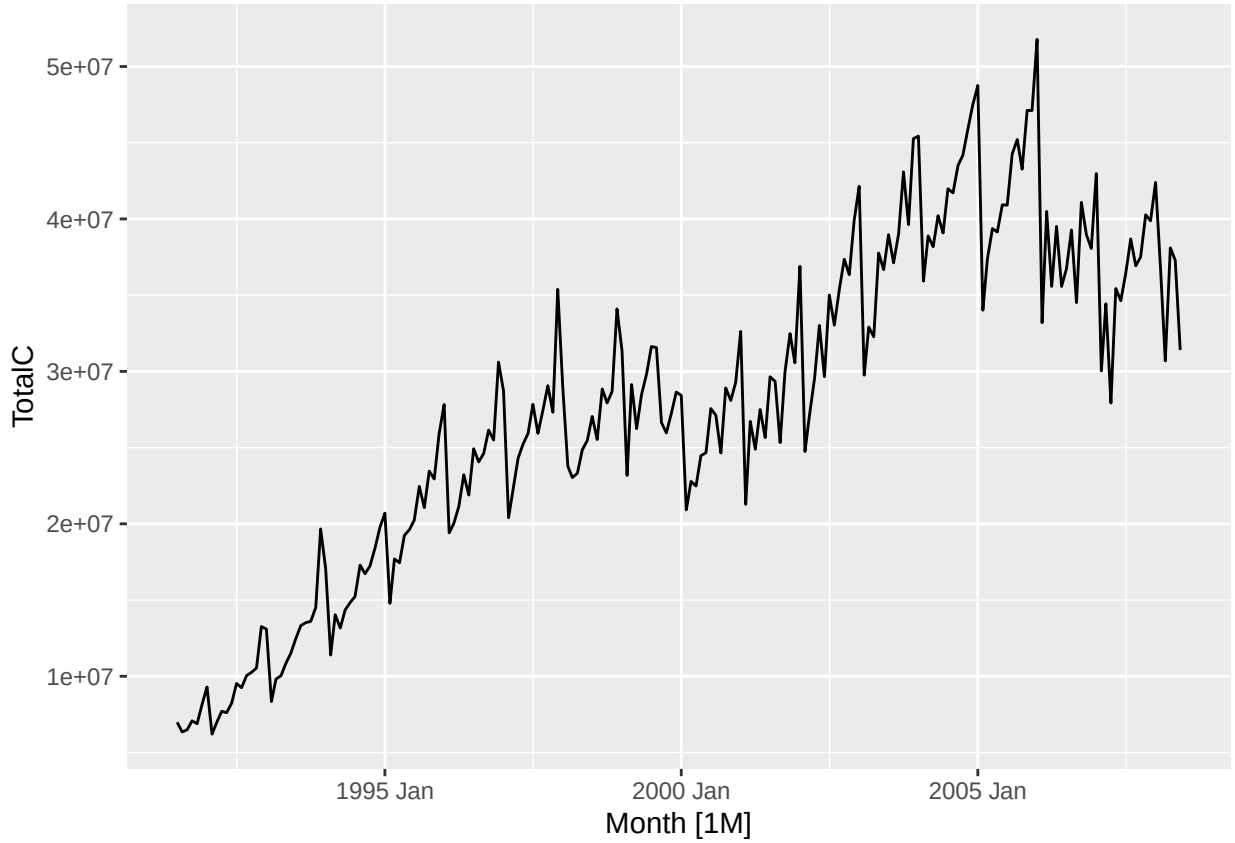
```
PBS %>%
  filter(ATC2 == "A01") %>%
  select(Month, Concession, Type, Cost) %>%
  summarise(TotalC = sum(Cost)) -> a01


PBS %>%
  filter(ATC2 == "A02") %>%
  select(Month, Concession, Type, Cost) %>%
  summarise(TotalC = sum(Cost)) -> a02
```

```
autoplot(a01, TotalC)
```




```
autoplot(a02, TotalC)
```



vic_elec

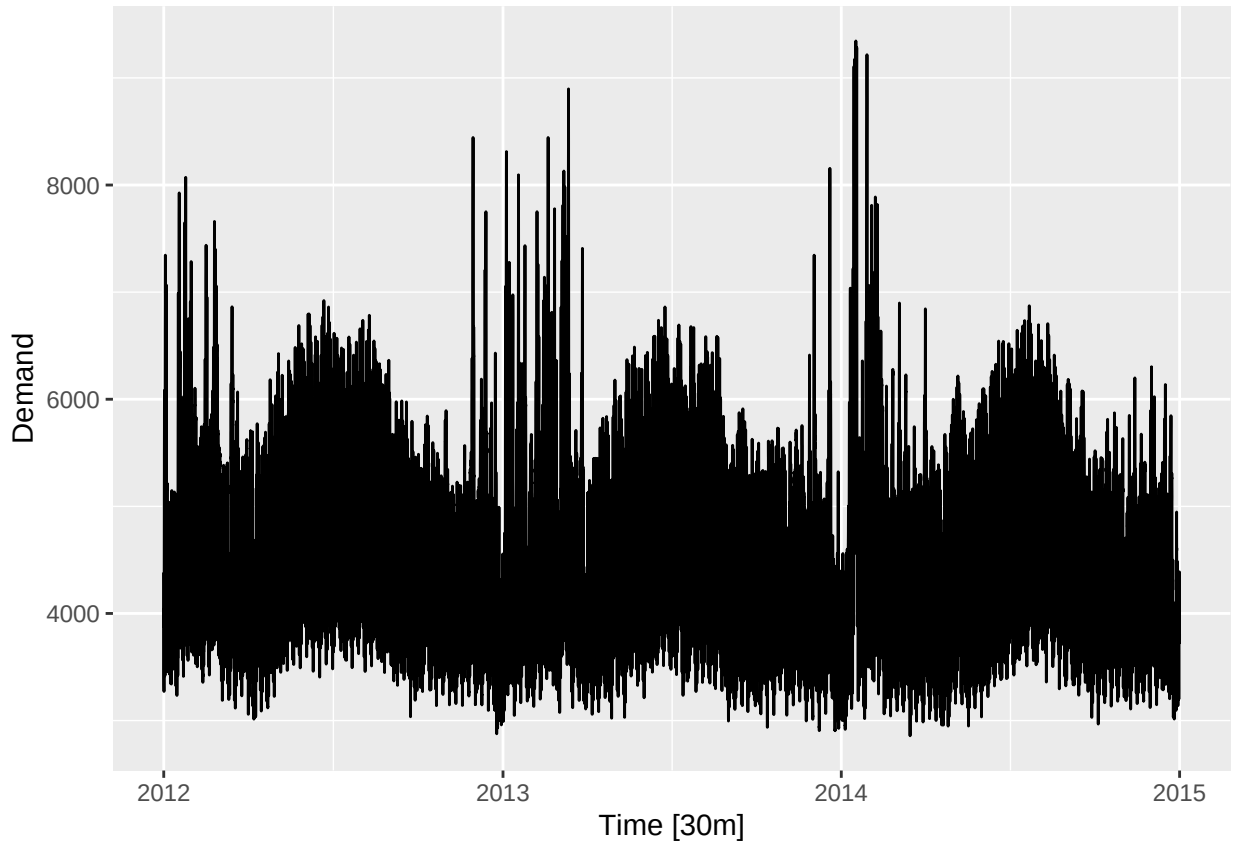
The data in `vic_elec` represents half-hourly electicity demand for Victoria, Australia given three values; Demand, Temperature, and Holiday.

```
vic_elec
```

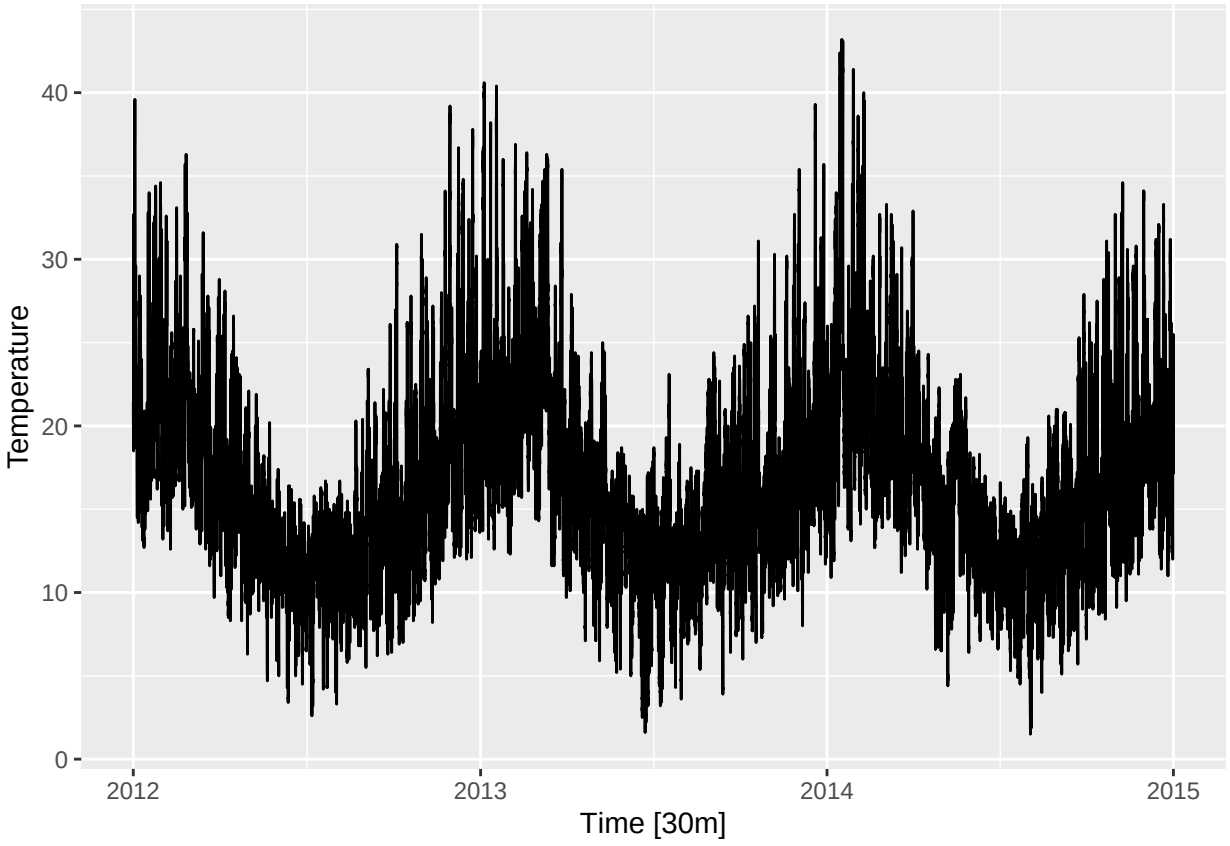
```
## # A tsibble: 52,608 x 5 [30m] <Australia/Melbourne>
##    Time                Demand Temperature Date       Holiday
##    <dttm>               <dbl>       <dbl> <date>     <lgl>
##  1 2012-01-01 00:00:00  4383.        21.4 2012-01-01 TRUE
##  2 2012-01-01 00:30:00  4263.        21.0 2012-01-01 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
```

Looking at the `tsibble` object above, we have 52,608 records with 5 values per record. Those are `Time`, `Demand`, `Temperature`, `Date`, and `Holiday`. The data [30m] shows that the data is at a half-hour time period.

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




```
autoplot(vic_elec, Temperature)
```

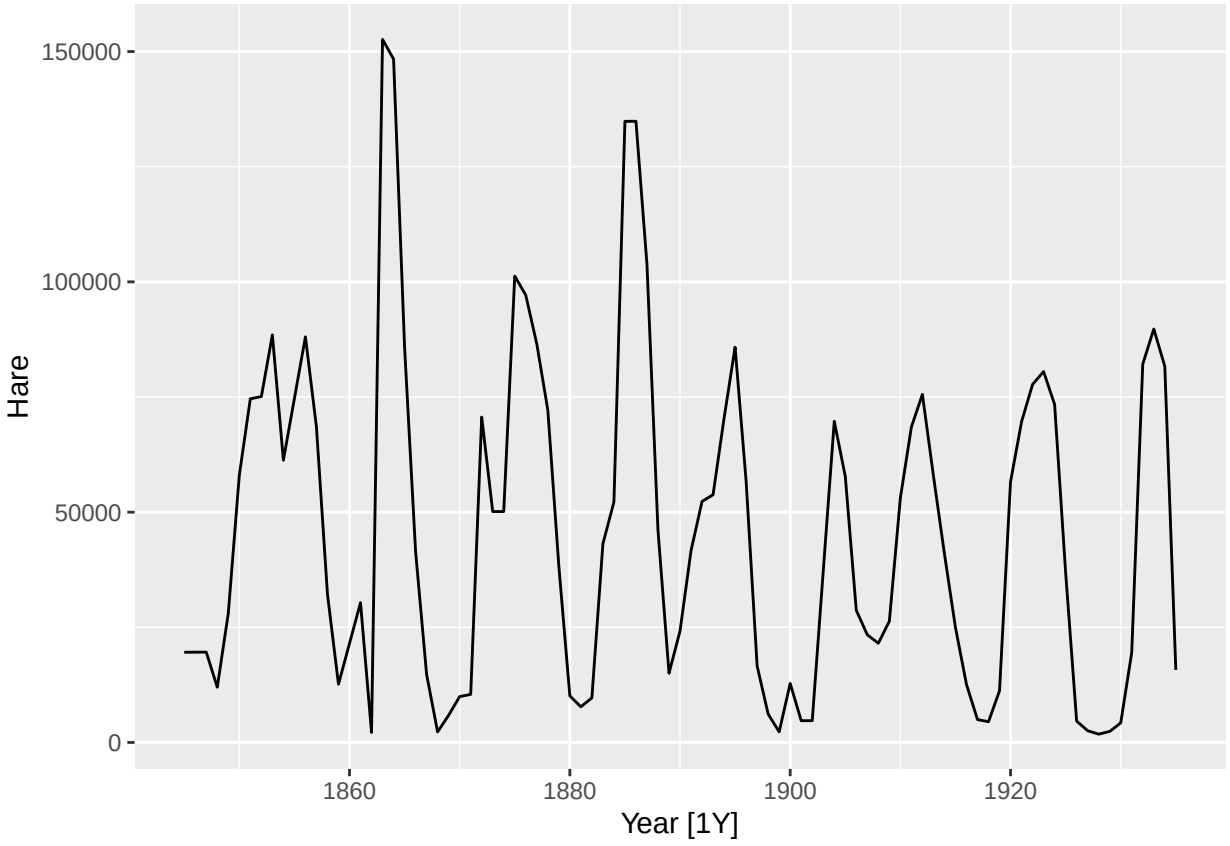


pelt

The data in `pelt` represents Hudson Bay Company trading records for Snowshow Hare and Canadian Lynx furs from 1845 to 1935.

```
autoplot(pelt)
```

```
## Plot variable not specified, automatically selected `.vars = Hare`
```



Looking at the `tsibble` object above, we have 52,608 records with 5 values per record. Those are `Time`, `Demand`, `Temperature`, `Date`, and `Holiday`. The data [30m] shows that the data is at a half-hour time period.

Exercise 2.10 - 2

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

Exercise 2.10 - 3

Download the file `tute1.csv` from the book website, open in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information. Columns B through D each contain 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.

a.

You can read the data into R with the following script:

```
tute1 <- readr::read_csv("tute1.csv")
```

```
## Parsed with column specification:
## cols(
##   Quarter = col_date(format = ""),
##   Sales = col_double(),
##   AdBudget = col_double(),
##   GDP = col_double()
## )
```

```
View(tute1)
```

b.

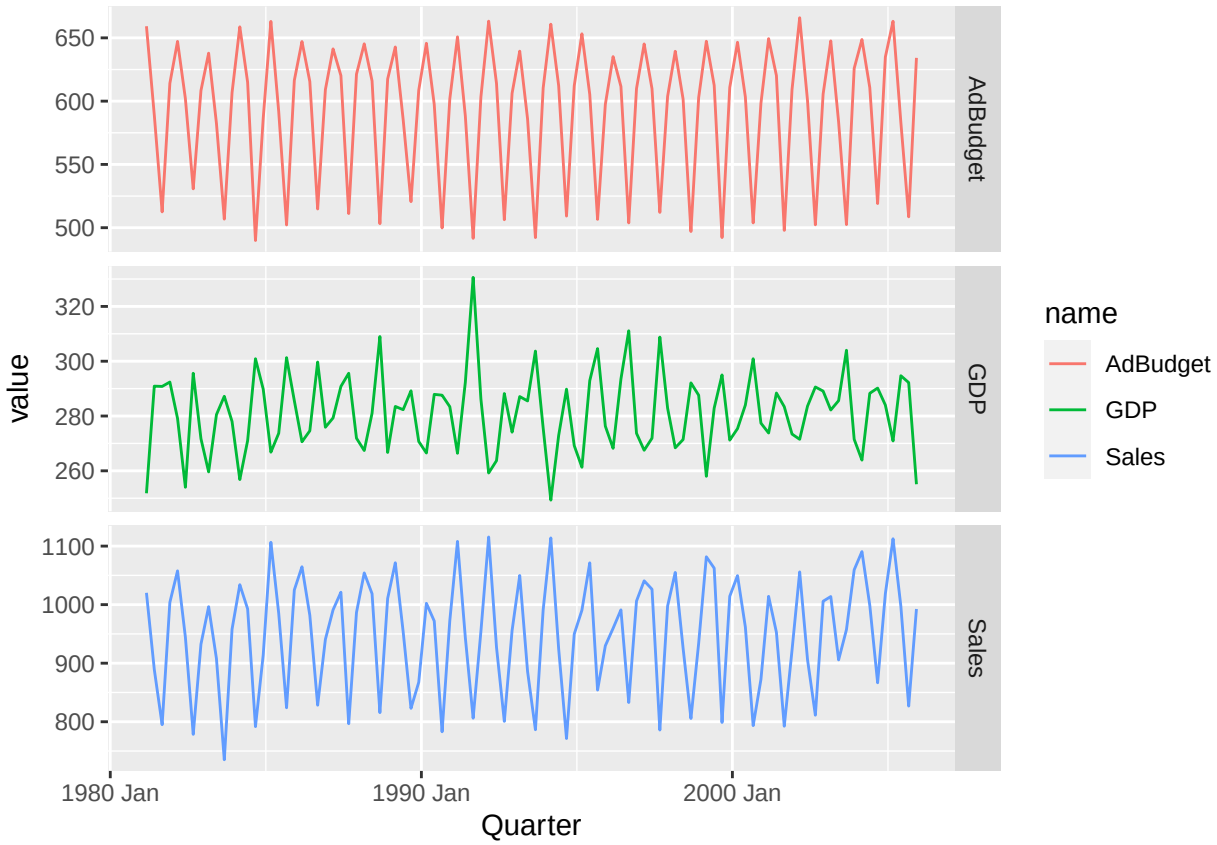
Convert the data to time series

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

c.

Construct time series plots of each of the three series

```
mytimeseries %>%
  pivot_longer(-Quarter) %>%
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line() +
  facet_grid(name ~ ., scales = "free_y")
```



Check what happens when you don't include `facet_grid()`.

Exercise 2.10 - 6

Create time plots of the following four time series: `Bricks` from `aus_production`, `Lynx` from `pelt`, `Close` from `gafa_stock`, `Demand` from `vic_elect`.

- Use `?` (or `help()`) to find out about the data in each series.
- For the last plot_modify the axis labels and title.