Data 624 Homework 1

Steven Gonzalez

2/9/2025

Load Packages

```
library(fpp3)
library(USgas)
```

Exercise 1

Explore the following four time series: `Bricks` from `aus_production`, `Lynx` from `pelt`, `Close` from `gafa_stock`, `Demand` from `vic_elec`.

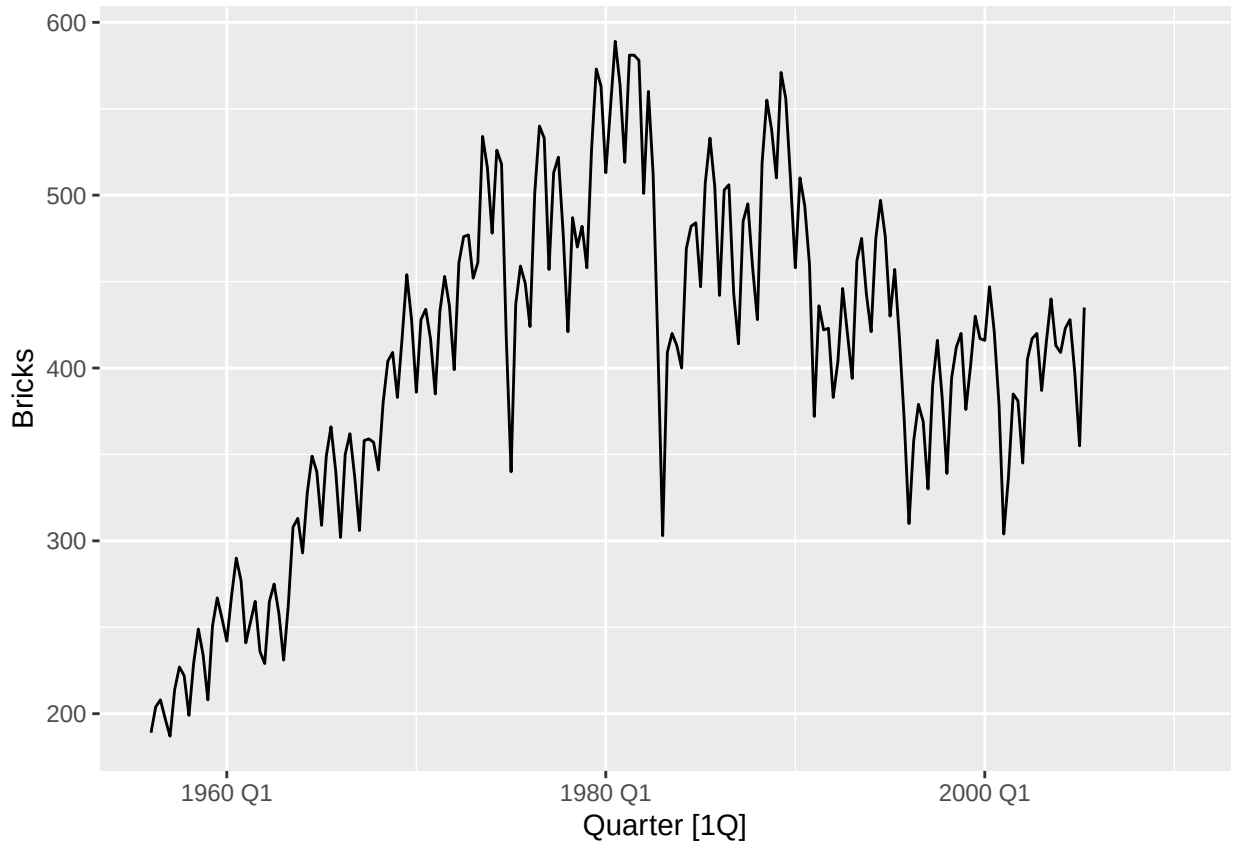
Use `?` (or `help()`) to find out about the data in each series. What is the time interval of each series? Use `autoplot()` to produce a time plot of each series. For the last plot, modify the axis labels and title.

```
?aus_production
aus_production #used to get further familiarized with the data
```

```
## # A tsibble: 218 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
##  7 1957 Q3   236    5317    227    603        5259     7
##  8 1957 Q4   320    6152    222    582        4735     6
##  9 1958 Q1   272    5758    199    554        4608     5
## 10 1958 Q2   233    5641    229    620        5196     7
## # i 208 more rows
```

As can be seen from the results above, the `Bricks` time series from `aus_production` has a quarterly time interval. Below is the time plot illustrating this using `autoplot()`.

```
autoplot(aus_production, Bricks)
```

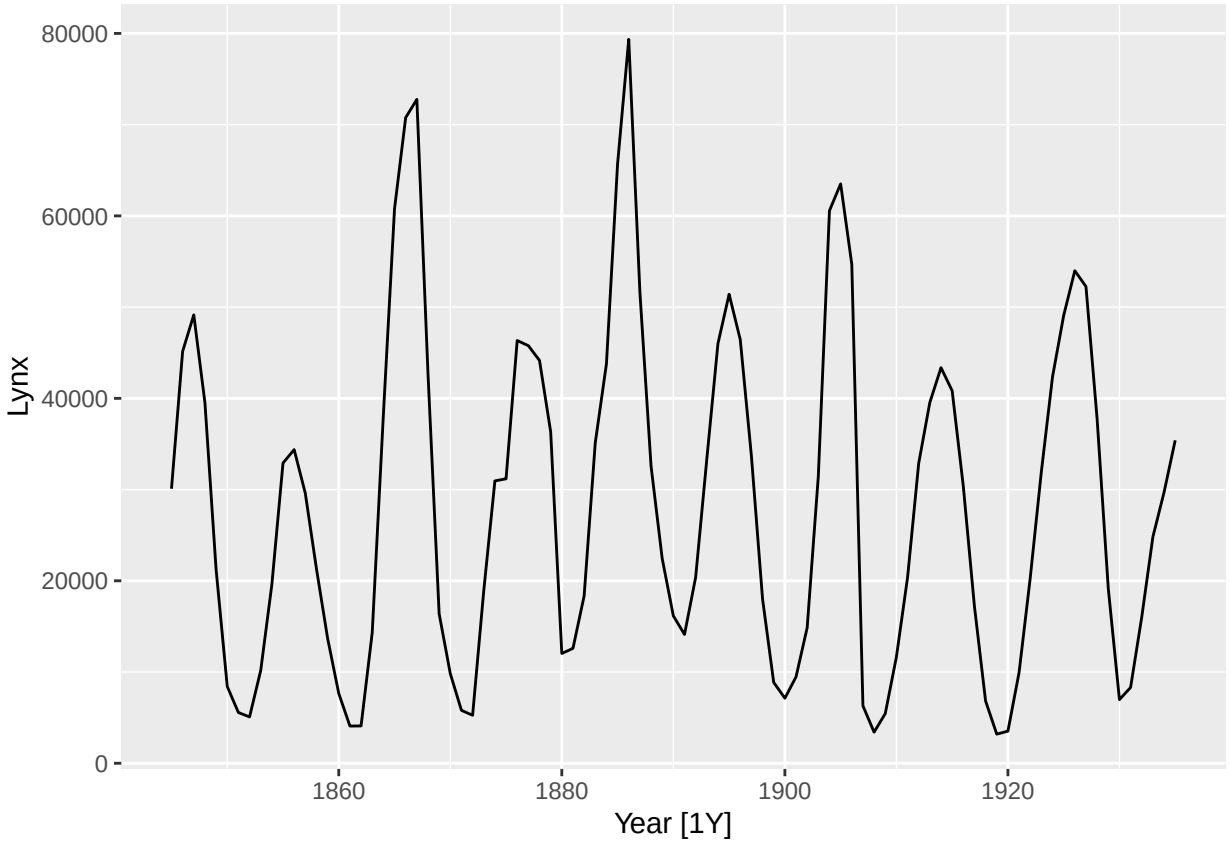



```
?pelt
pelt #used to get further familiarized with the data
```

```
## # A tsibble: 91 x 3 [1Y]
##     Year  Hare  Lynx
##    <dbl> <dbl> <dbl>
##  1  1845 19580 30090
##  2  1846 19600 45150
##  3  1847 19610 49150
##  4  1848 11990 39520
##  5  1849 28040 21230
##  6  1850 58000  8420
##  7  1851 74600  5560
##  8  1852 75090  5080
##  9  1853 88480 10170
## 10  1854 61280 19600
## # i 81 more rows
```

As can be seen from the results above, the `Lynx` time series from `pelt` has an annual time interval. Below is the time plot illustrating this using `autoplot()`.

```
autoplot(pelt, Lynx)
```

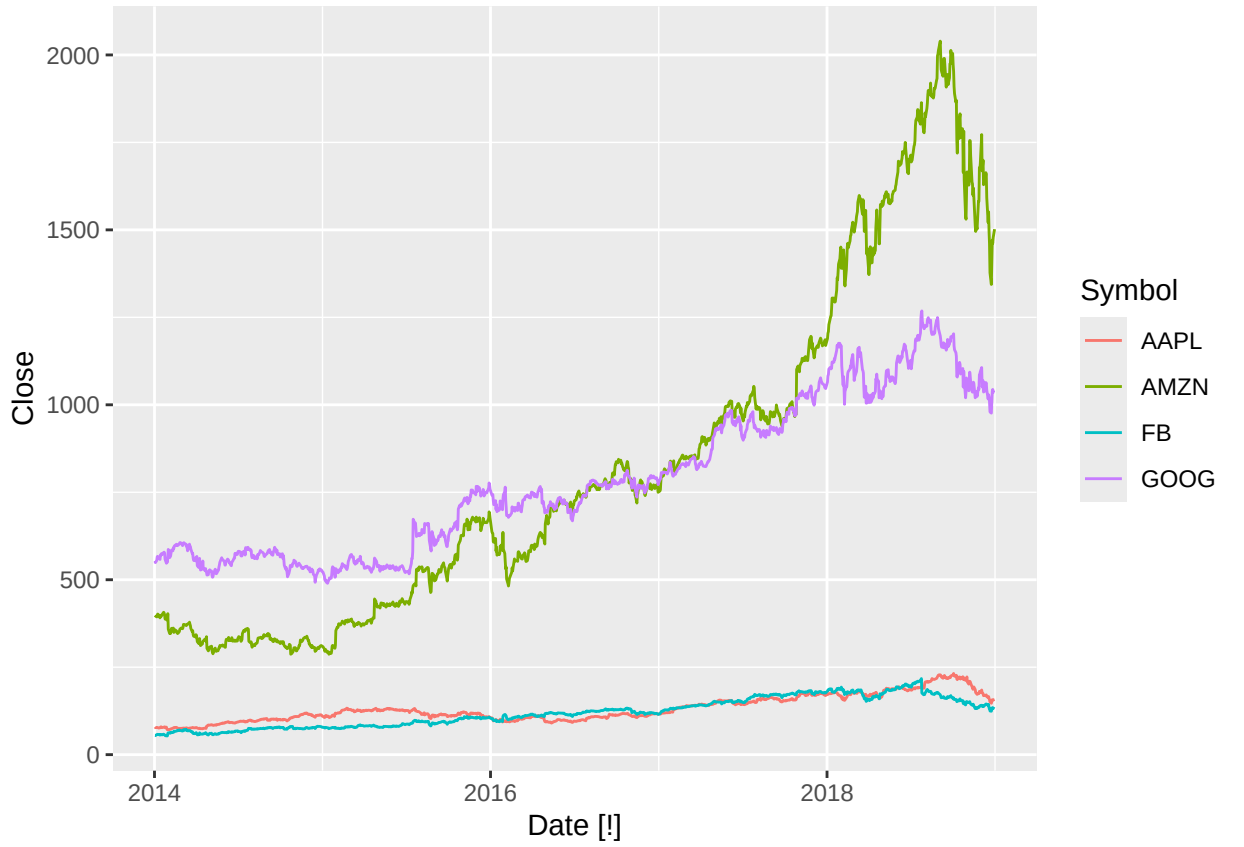



```
?gafa_stock
gafa_stock #used to get further familiarized with the data
```

```
## # 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
## # i 5,022 more rows
```

As can be seen from the results above, the `Close` time series from `gafa_stock` has a time interval with specific dates that seem to be business days, which would make sense given that it is a data set on stock prices. Below is the time plot illustrating this using `autoplot()`.

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

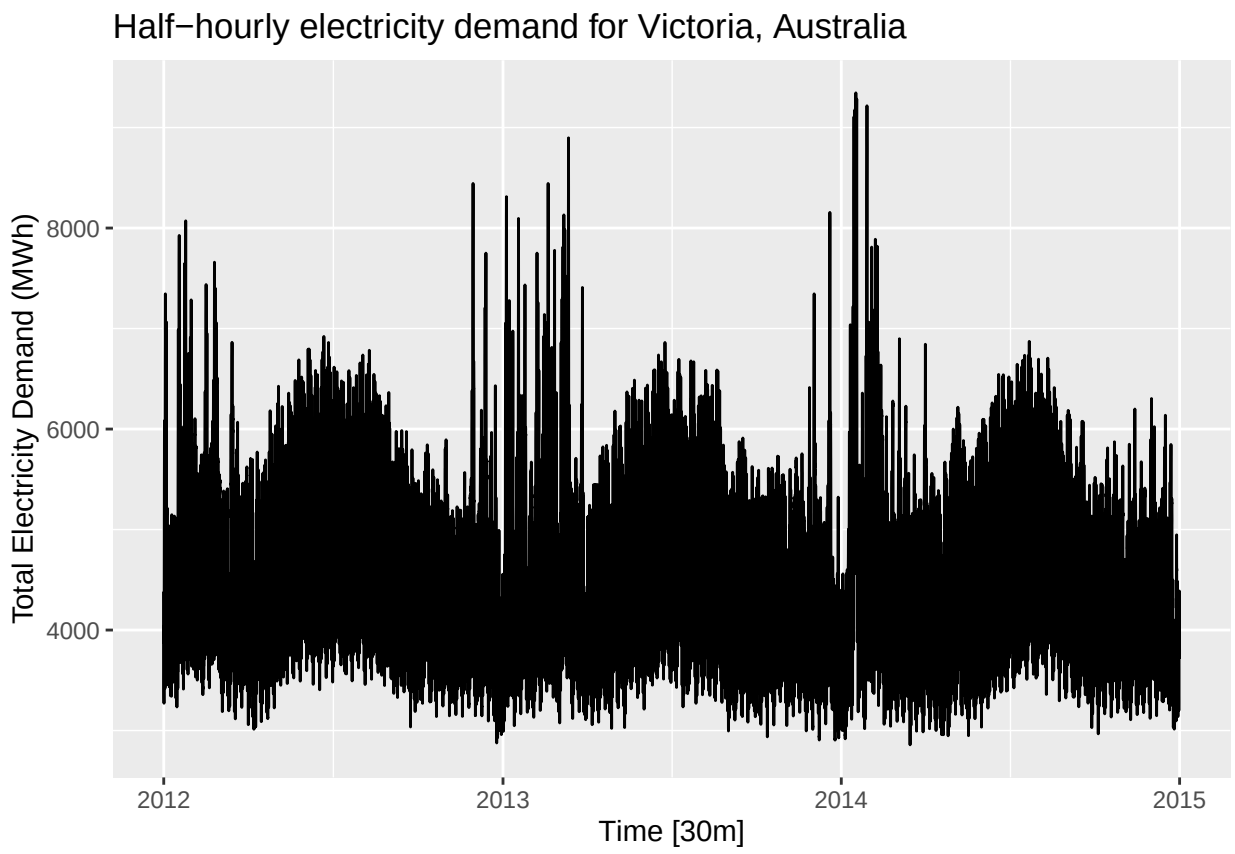



```
?vic_elec
vic_elec #used to get further familiarized with the data
```

```
## # 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
## # i 52,598 more rows
```

As can be seen from the results above, the `Demand` time series from `vic_elec` has a half-hourly time interval. Below is the time plot illustrating this using `autoplot()` with modified title and axis labels.

```
autoplot(vic_elec, Demand) +
  labs(title = "Half-hourly electricity demand for Victoria, Australia",
       y = "Total Electricity Demand (MWh)")
```



Exercise 2

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

```
aapl_peak <- gafa_stock %>%
  filter(Symbol == "AAPL") %>%
  select(Symbol, Date, Close) %>%
  slice_max(Close, n = 1)
aapl_peak
```

```
## # A tsibble: 1 x 3 [!]
## # Key:       Symbol [1]
##   Symbol Date       Close
##   <chr>  <date>     <dbl>
## 1 AAPL   2018-10-03  232.
```

```
amzn_peak <- gafa_stock %>%
  filter(Symbol == "AMZN") %>%
```

```
  select(Symbol, Date, Close) %>%
  slice_max(Close, n = 1)
amzn_peak
```

```
## # A tsibble: 1 x 3 [!]
## # Key:       Symbol [1]
##   Symbol Date       Close
##   <chr>  <date>     <dbl>
## 1 AMZN   2018-09-04 2040.
```

```
fb_peak <- gafa_stock %>%
  filter(Symbol == "FB") %>%
  select(Symbol, Date, Close) %>%
  slice_max(Close, n = 1)
fb_peak
```

```
## # A tsibble: 1 x 3 [!]
## # Key:       Symbol [1]
##   Symbol Date       Close
##   <chr>  <date>     <dbl>
## 1 FB     2018-07-25  218.
```

```
goog_peak <- gafa_stock %>%
  filter(Symbol == "GOOG") %>%
  select(Symbol, Date, Close) %>%
  slice_max(Close, n = 1)
goog_peak
```

```
## # A tsibble: 1 x 3 [!]
## # Key:       Symbol [1]
##   Symbol Date       Close
##   <chr>  <date>     <dbl>
## 1 GOOG   2018-07-26 1268.
```

Exercise 3

Download the file `tute1.csv` from the book website, 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.

You can read the data into R with the following script: `tute1 <- readr::read_csv("tute1.csv") View(tute1)`

Convert the data to time series `mytimeseries <- tute1 |> mutate(Quarter = yearquarter(Quarter)) |> as_tsibble(index = Quarter)`

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()`.

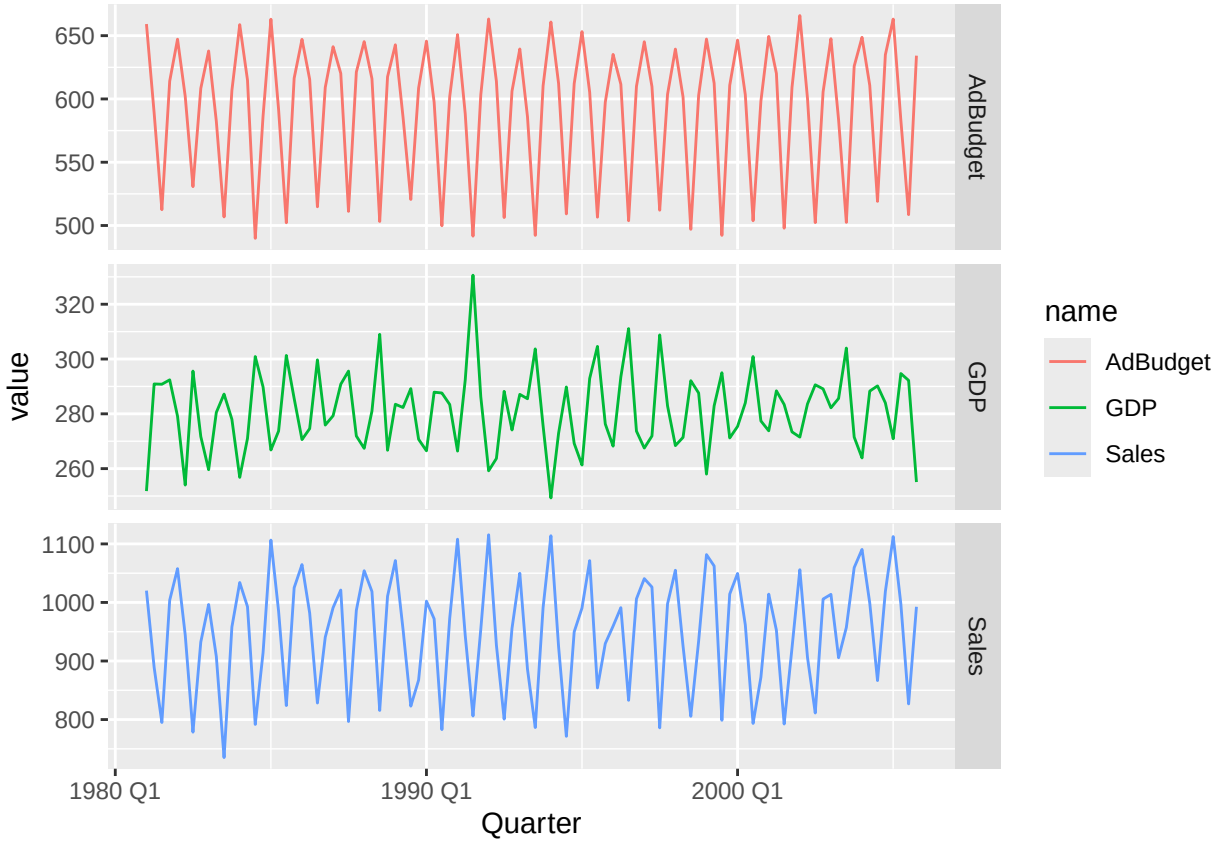
```
url <- "https://raw.githubusercontent.com/Stevee-G/Data624/refs/heads/main/tute1.csv"
tute1 <- readr::read_csv(url) #Had to modify the command in order to make the RMD reproducible
```

```
## 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.
```

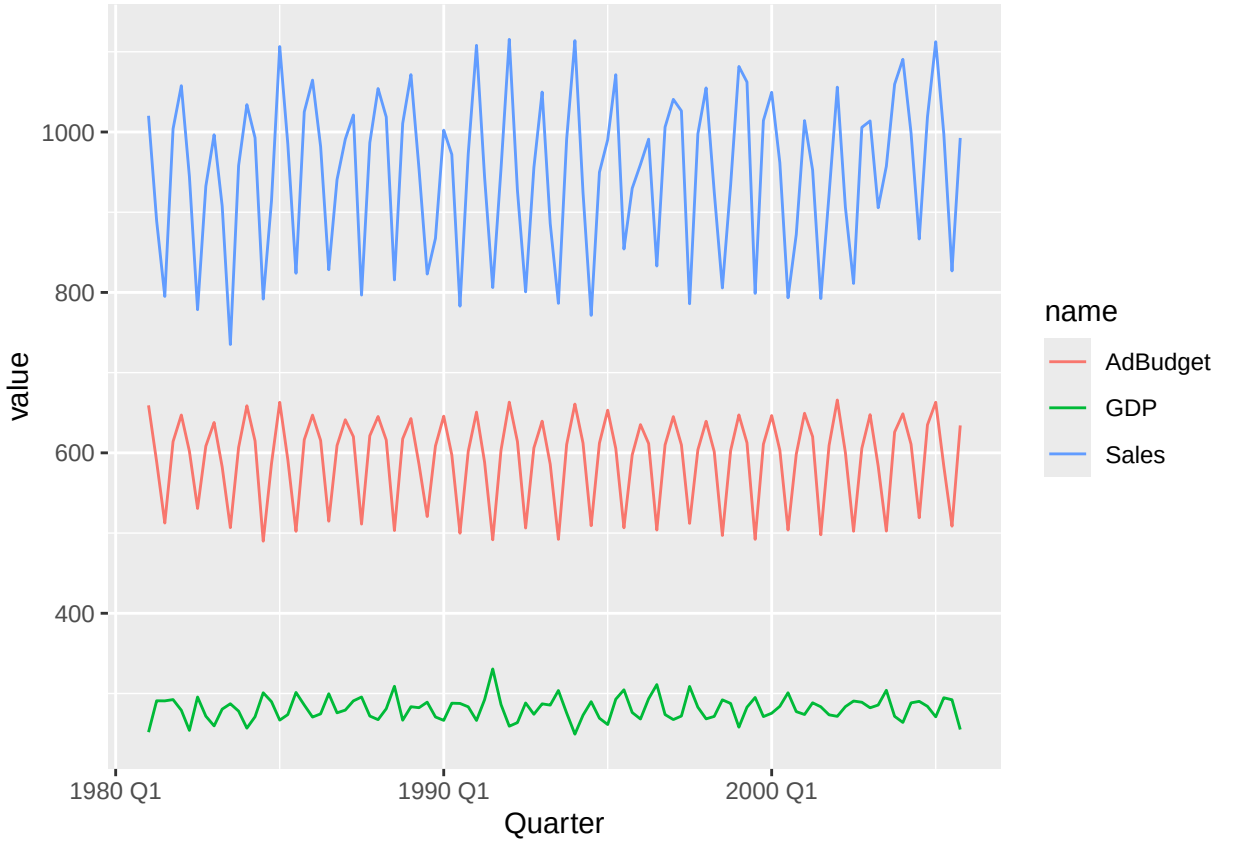
```
View(tute1)

mytimeseries <- tute1 %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(index = Quarter) #Modified the pipe due to personal preference

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



```
mytimeseries %>%
  pivot_longer(-Quarter) %>%
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line()
```



Exercise 4

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

Install the `USgas` package. Create a `tsibble` from `us_total` with year as the index and state as the key. Plot the annual natural gas consumption by state for the New England area (comprising the states of Maine, Vermont, New Hampshire, Massachusetts, Connecticut and Rhode Island).

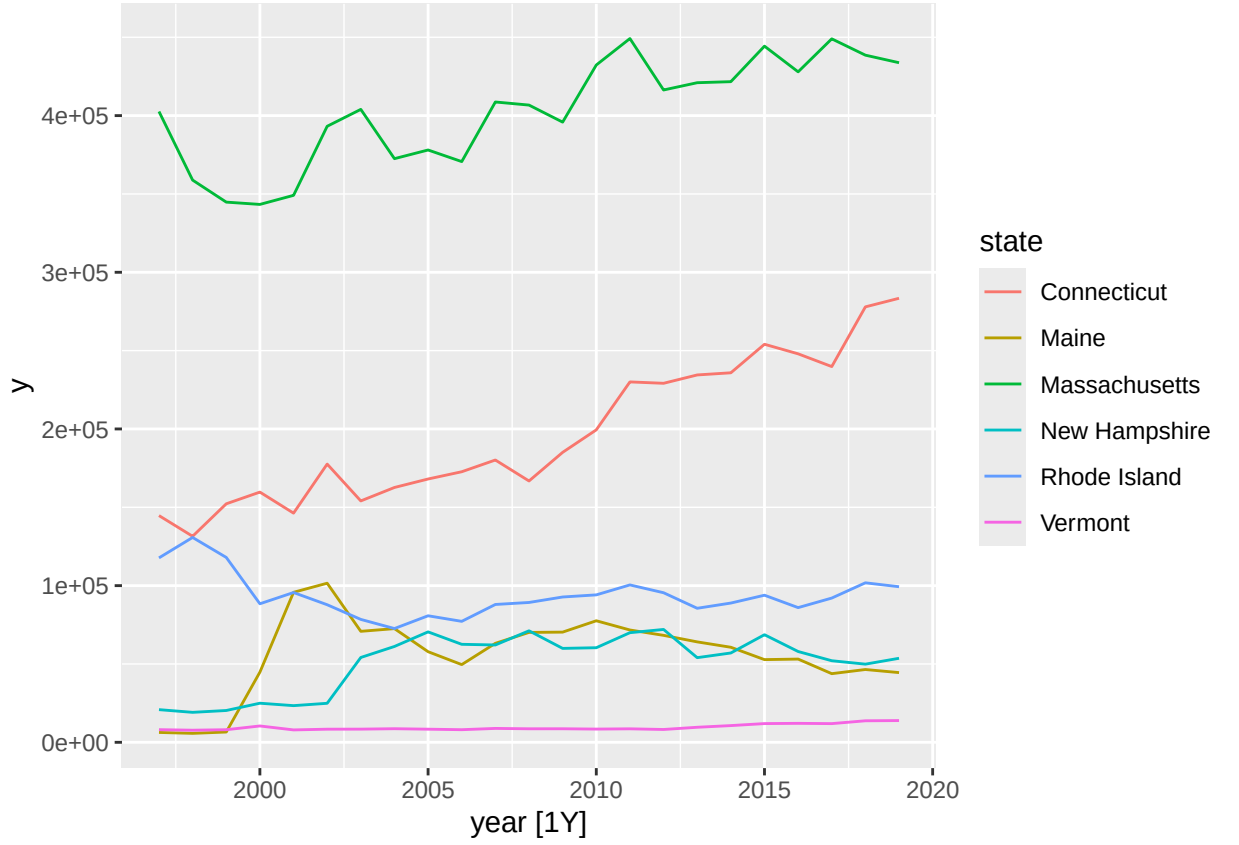
```
#USgas package was installed and loaded in a previous section
?us_total
glimpse(us_total)
```

```
## Rows: 1,266
## Columns: 3
## $ year  <int> 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007~
## $ state <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"~
## $ y     <int> 324158, 329134, 337270, 353614, 332693, 379343, 350345, 382367, ~
```

```r
us_total_ts <- us_total %>%
  as_tsibble(index = year, key = state)

new_england <- us_total_ts %>%
  filter(state == "Maine" |
           state == "Vermont" |
           state == "New Hampshire" |
           state == "Massachusetts" |
           state == "Connecticut" |
           state == "Rhode Island")

autoplot(new_england, y)
```



Exercise 5

Download `tourism.xlsx` from the book website and read it into R using `readxl::read_excel()`. Create a tsibble which is identical to the `tourism` tsibble from the `tsibble` package. Find what combination of `Region` and `Purpose` had the maximum number of overnight trips on average. Create a new tsibble which combines the Purposes and Regions, and just has total trips by State.

```r
url1 <- "https://raw.githubusercontent.com/Stevee-G/Data624/refs/heads/main/tourism.csv"
tourism1 <- readr::read_csv(url1) #Had to resort to csv due to an issue with OneDrive making the excel
View(tourism1)
```