DATA 624 - Homework 1

Richie Rivera

Question 2.1

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

```
# Loading the datasets
data(aus_production)
data(pelt)
data(gafa_stock)
data(vic_elec)
```

```
?aus_production
?pelt
?gafa_stock
?vic_elec
```

A: Use ? (or help()) to find out about the data in each series.

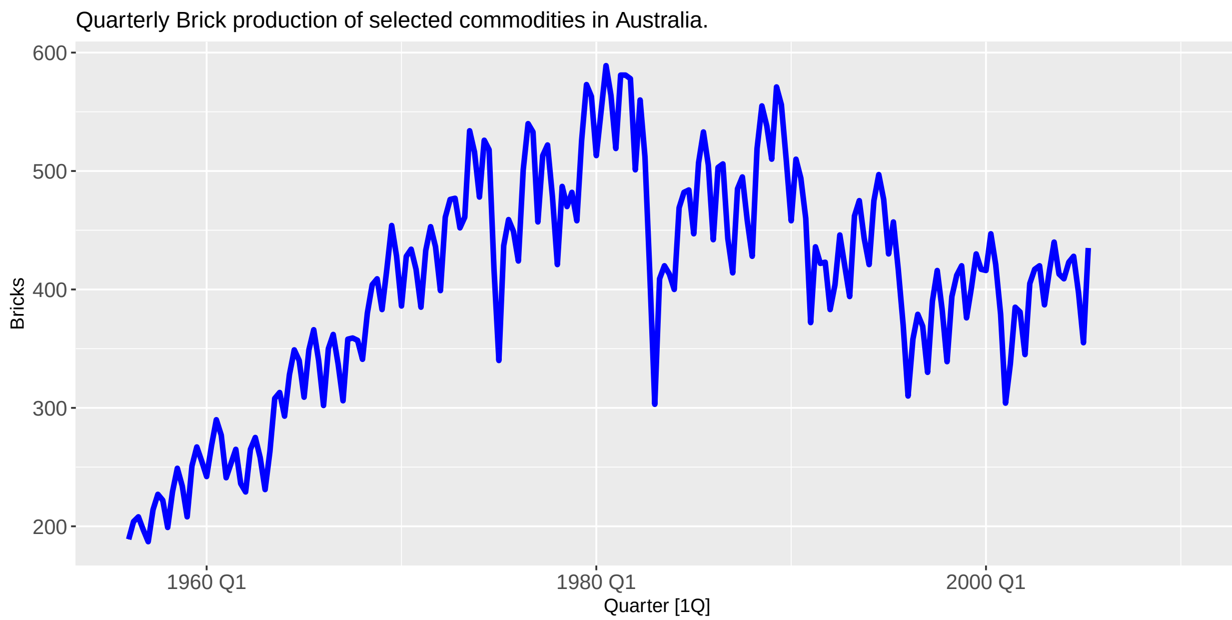
B: What is the time interval of each series?

Series	Time Interval	Description
aus_production	Quarterly	Quarterly production of selected commodities in Australia.
pelt	Annual	Pelt trading records
gafa_stock	Daily	GAFA stock prices
vic_elec	Half-Hourly	Half-hourly electricity demand for Victoria, Australia

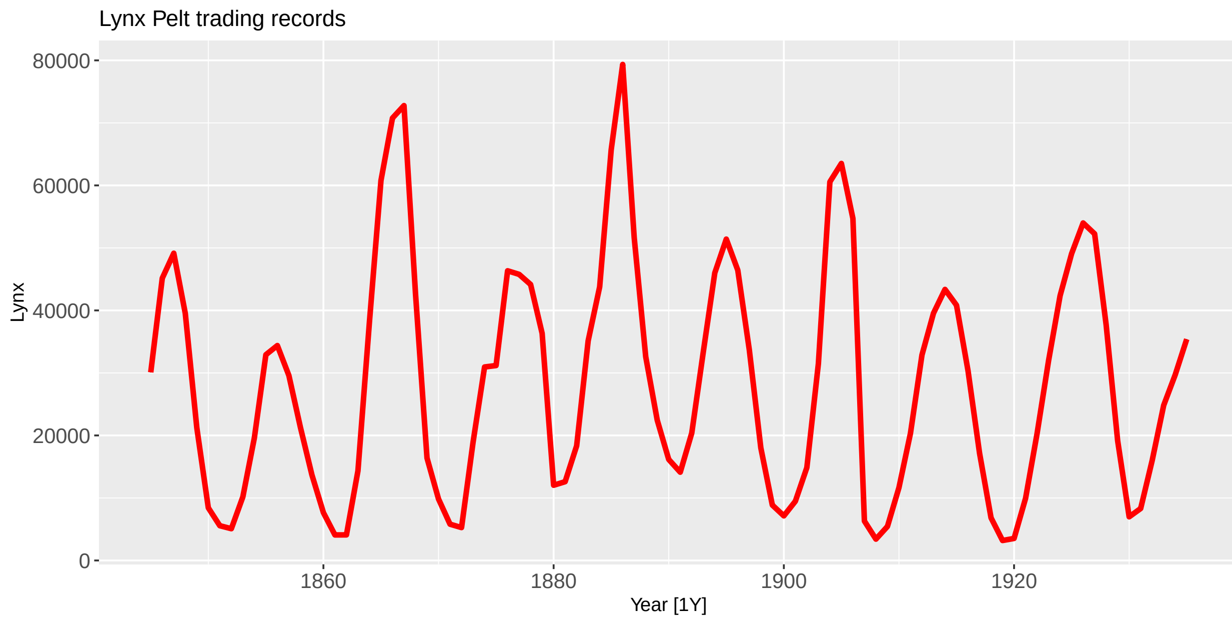
```
# Using autoplot to plot charts
autoplot(aus_production, Bricks) + ggtitle(
  "Quarterly Brick production of selected commodities in Australia."
) + geom_line(color = "blue", size = 1.5) +
  theme(axis.text = element_text(size = 12))
```

C: Use autoplot() to produce a time plot of each series.

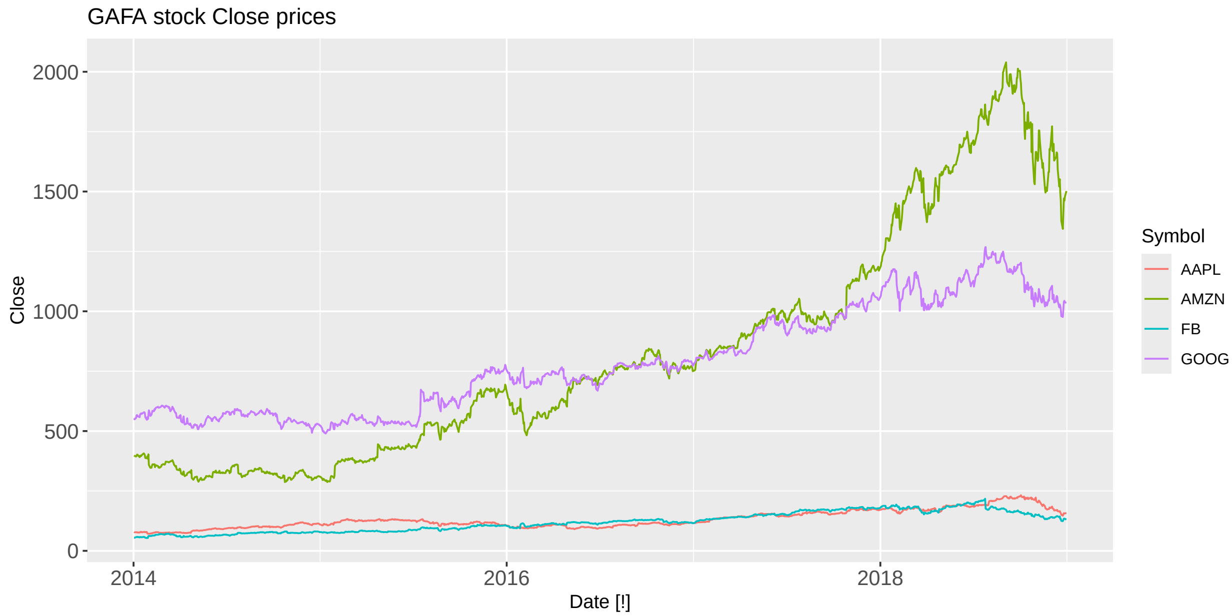
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: Removed 20 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 20 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



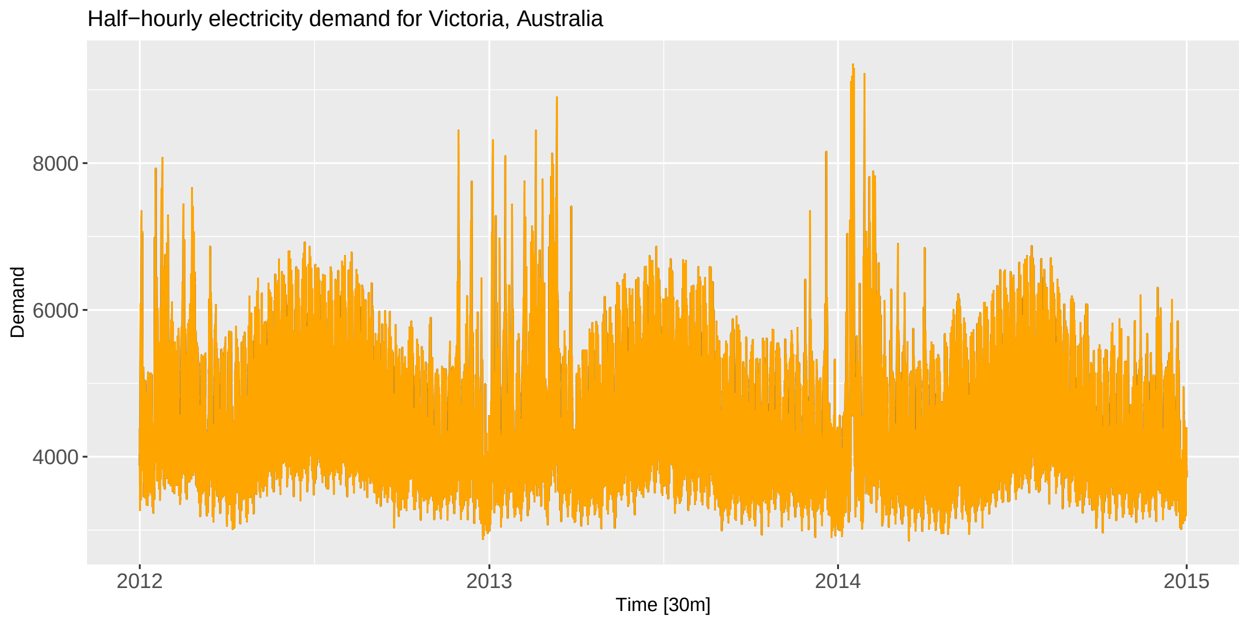

```r
autoplot(pelt, Lynx) + ggtitle(
  "Lynx Pelt trading records"
) + geom_line(color = "red", size = 1.5) +
  theme(axis.text = element_text(size = 12))
```



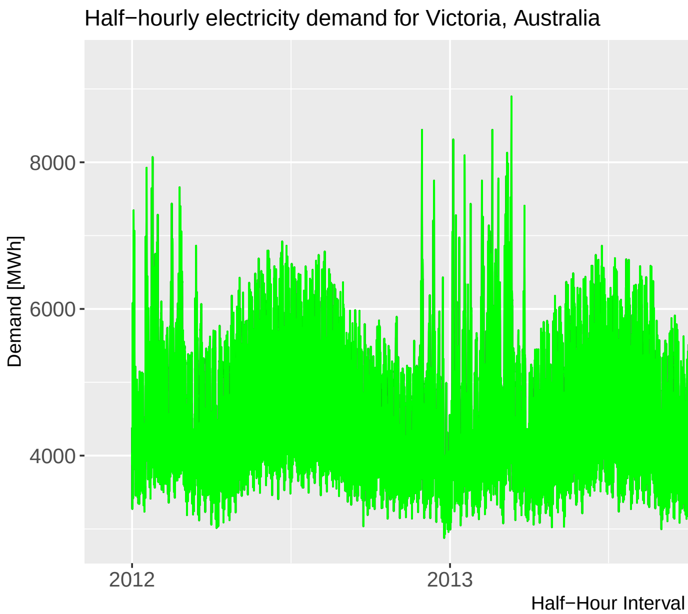

```r
autoplot(gafa_stock, Close) + ggtitle(
  "GAFA stock Close prices"
) + theme(axis.text = element_text(size = 12))
```




```r
autoplot(vic_elec, Demand) + ggtitle(
  "Half-hourly electricity demand for Victoria, Australia"
) + geom_line(color = "orange", size = 0.5) +
  theme(axis.text = element_text(size = 12))
```




```r
# Modifying chart legends and axis
autoplot(vic_elec, Demand) + ggtitle(
  "Half-hourly electricity demand for Victoria, Australia"
) + geom_line(color = "green", size = 0.5) +
  theme(axis.text = element_text(size = 12), aspect.ratio = 0.5) +
  xlab("Half-Hour Interval") +
  ylab("Demand [MWh]")
```



D: For the last plot, modify the axis labels and title.

Question 2.2

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

```
# Importing dplyr
library(dplyr)

# inspecting the first few rows of the data
head(gafa_stock)
```

```
## # A tsibble: 6 x 8 [!]
## # Key:       Symbol [1]
##   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
```

```
# Filtering the data
gafa_stock |>
  select(
    Symbol,
    Date,
    Close
  ) |>
  group_by(Symbol) |>
  filter(
    Close == max(Close)
  )
```

```
## # A tsibble: 4 x 3 [!]
## # Key:       Symbol [4]
## # Groups:    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.
```

Question 2.3

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.

```r
# importing readr
library(readr)

# Reading and viewing the csv
tute1 <- read_csv("https://raw.githubusercontent.com/riverar9/cuny-msds/main/data624-predictive-analyti
```

A. You can read the data into R with the following script

```
## 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.
```

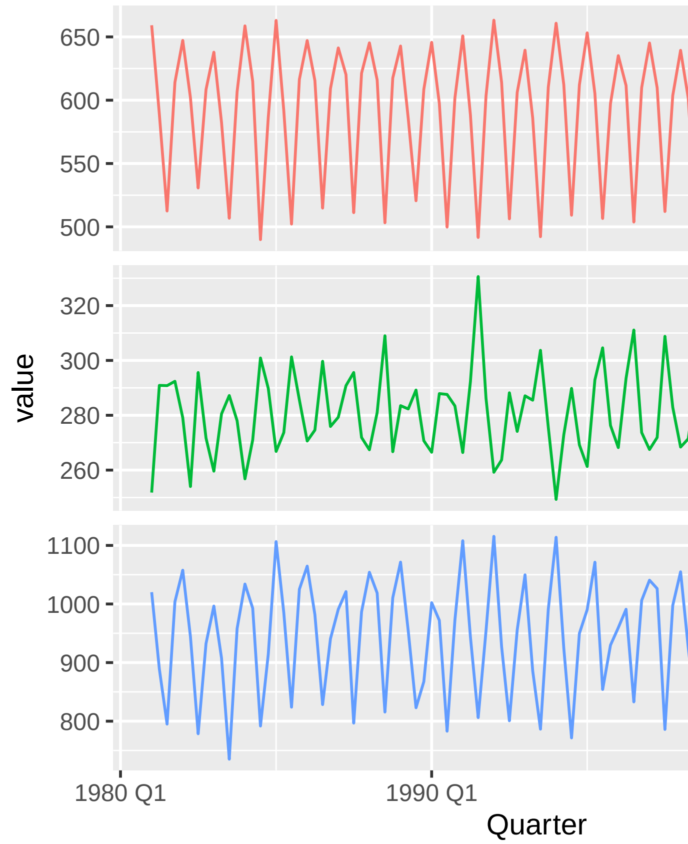
```r
View(tute1)
```

```r
# Converting the data into a timeseries
mytimeseries <- tute1 |>
  mutate(Quarter = yearquarter(Quarter)) |>
  as_tsibble(index = Quarter)

head(mytimeseries)
```

B. Convert the data to time series

```
## # A tsibble: 6 x 4 [1Q]
##   Quarter Sales AdBudget   GDP
##     <qtr> <dbl>    <dbl> <dbl>
## 1 1981 Q1 1020.     659.  252.
## 2 1981 Q2  889.     589   291.
## 3 1981 Q3  795      512.  291.
## 4 1981 Q4 1004.     614.  292.
## 5 1982 Q1 1058.     647.  279.
## 6 1982 Q2  944.     602   254
```

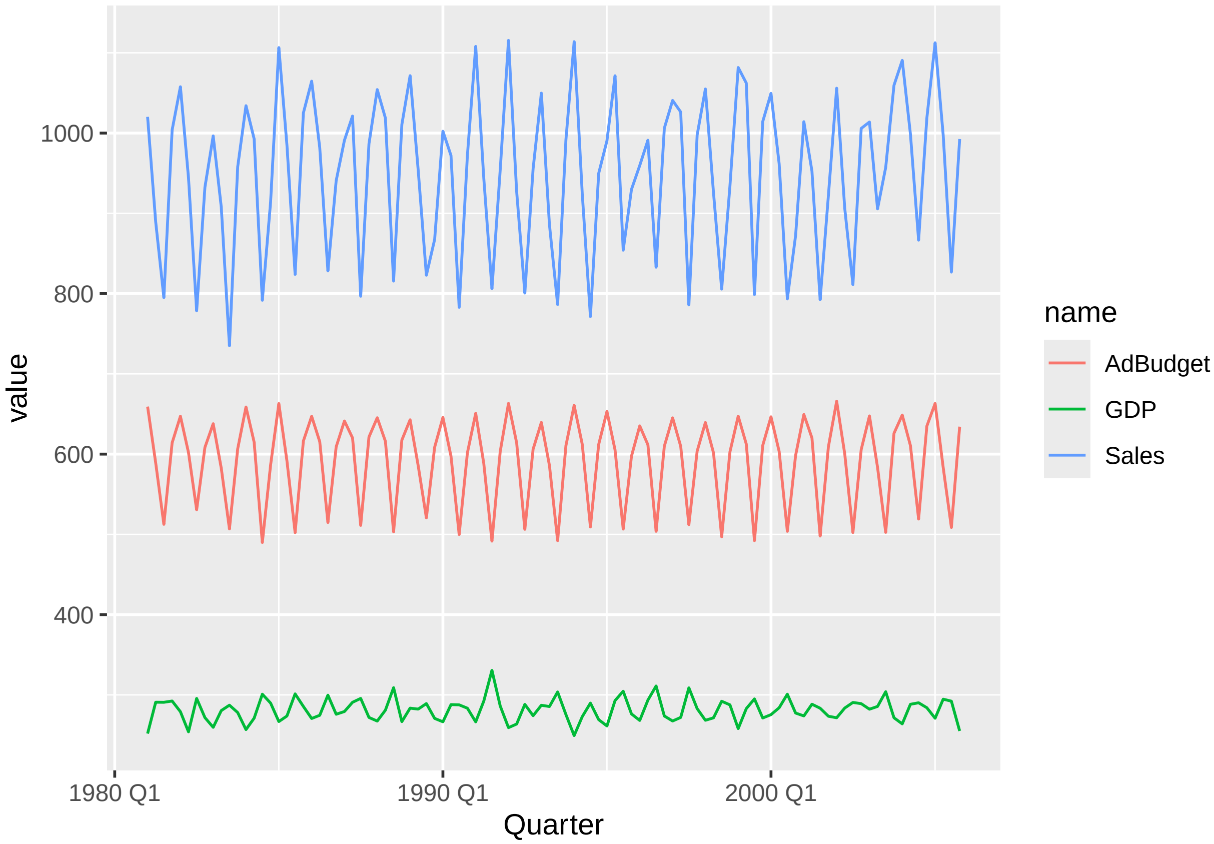
```r
# Create a series of plots using facet_grid
mytimeseries |>
  pivot_longer(-Quarter) |>
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line() +
  facet_grid(name ~ ., scales = "free_y")
```



C. Construct time series plots of each of the three series

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

```
# Removing facet wrap
mytimeseries |>
  pivot_longer(-Quarter) |>
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line()
```



Without facet_grid, the plots are all on the same chart. In my opinion this is more helpful as it provides immediate insight into the relative value of these timeseries against eachother.

Question 2.4

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

```
# Installing the package.
#install.packages("USgas")
```

A. Install the USgas package.

```
# Improting USgas and tsibble
library(USgas)
```

B. Create a tsibble from us_total with year as the index and state as the key.

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

```r
library(tsibble)
library(tibble)

# Loading us_total and displaying the first few records
?us_total
```

```
## starting httpd help server ... done
```

```r
data(us_total)
head(us_total)
```

```
##   year   state      y
## 1 1997 Alabama 324158
## 2 1998 Alabama 329134
## 3 1999 Alabama 337270
## 4 2000 Alabama 353614
## 5 2001 Alabama 332693
## 6 2002 Alabama 379343
```

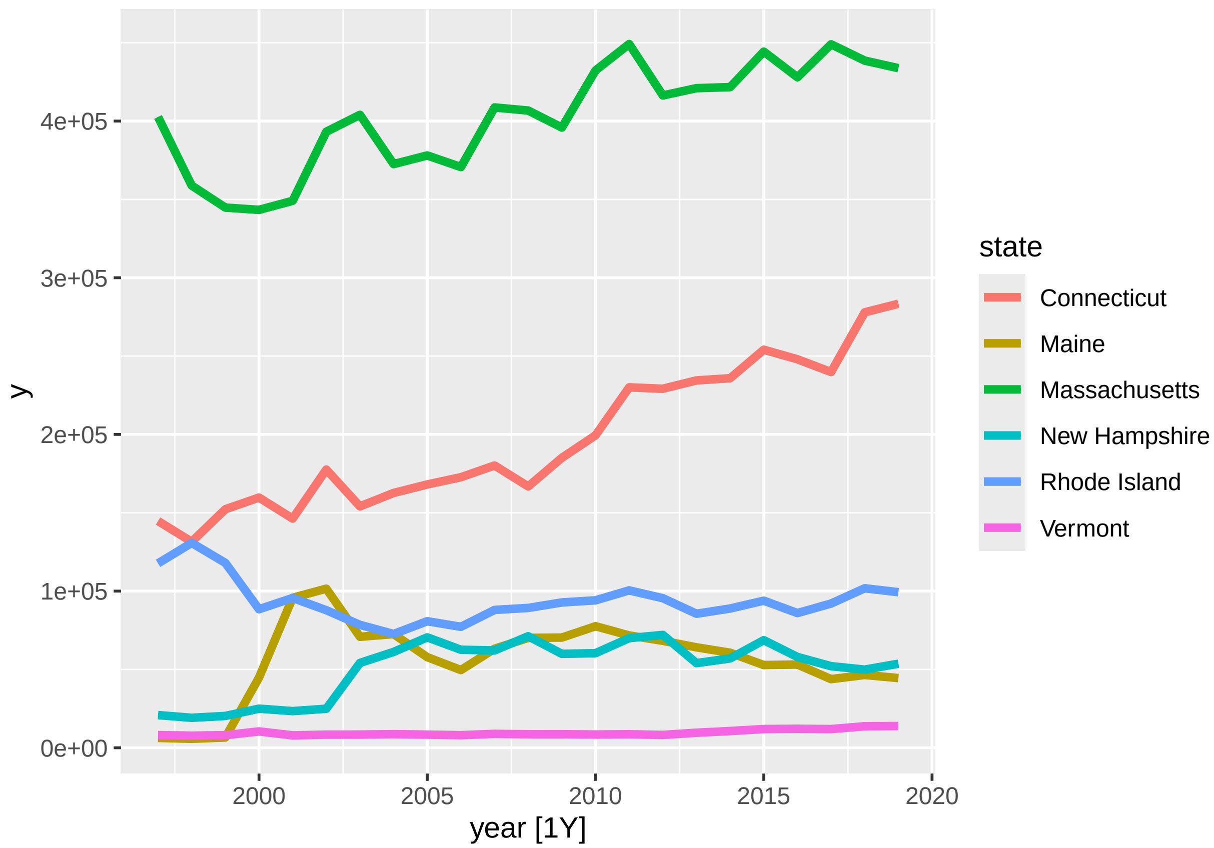
```r
# creating the tsibble
us_total_tsibble <- us_total |>
  as_tsibble(
    key = state,
    index = year
  )

us_total_tsibble
```

```
## # A tsibble: 1,266 x 3 [1Y]
## # Key:       state [53]
##     year state        y
##    <int> <chr>    <int>
##  1  1997 Alabama 324158
##  2  1998 Alabama 329134
##  3  1999 Alabama 337270
##  4  2000 Alabama 353614
##  5  2001 Alabama 332693
##  6  2002 Alabama 379343
##  7  2003 Alabama 350345
##  8  2004 Alabama 382367
##  9  2005 Alabama 353156
## 10  2006 Alabama 391093
## # i 1,256 more rows
```

```r
# Create a variable with just the states of interest
filtered_states <- us_total_tsibble |>
  filter(
    state %in% c(
      "Maine",
```

```
      "Vermont",
      "New Hampshire",
      "Massachusetts",
      "Connecticut",
      "Rhode Island"
    )
  )

# Plot the annual consumption by state
autoplot(
  filtered_states,
  y
) + geom_line(
  size = 1.5
)
```

C. 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).



Question 2.5

```
# Import the readxl and httr libraries
library(readxl)
```

```r
library(httr)

# Specify the file URL
file_url <- "https://github.com/riverar9/cuny-msds/raw/main/data624-predictive-analytics/homework/homew

# Download the file to the local repository
GET(
  file_url,
  write_disk(
    temp_file <- tempfile(
      fileext = ".xlsx"
    )
  )
)
```

A. Download tourism.xlsx from the book website and read it into R using readxl::read_excel().

```
## Response [https://raw.githubusercontent.com/riverar9/cuny-msds/main/data624-predictive-analytics/home
##   Date: 2024-09-08 23:59
##   Status: 200
##   Content-Type: application/octet-stream
##   Size: 679 kB
## <ON DISK>  C:\Users\Richie\AppData\Local\Temp\RtmpkvgrSa\file2b704b6270a.xlsx
```

```r
# Read in the file
tourism <- read_excel(temp_file)

# Delete the temp file
file.remove(temp_file)
```

```
## [1] TRUE
```

```r
# Display part of the file
head(tourism)
```

```
## # A tibble: 6 x 5
##   Quarter    Region   State           Purpose  Trips
##   <chr>      <chr>    <chr>           <chr>    <dbl>
## 1 1998-01-01 Adelaide South Australia Business  135.
## 2 1998-04-01 Adelaide South Australia Business  110.
## 3 1998-07-01 Adelaide South Australia Business  166.
## 4 1998-10-01 Adelaide South Australia Business  127.
## 5 1999-01-01 Adelaide South Australia Business  137.
## 6 1999-04-01 Adelaide South Australia Business  200.
```

```r
# Converting tourism into a tsibble
tourism_ts <- tourism |>
  mutate(
    Quarter = yearquarter(Quarter)
```

```r
  ) |>
  as_tsibble(
    key = c(
      Region,
      State,
      Purpose
    ),
    index = Quarter
  )

head(tourism_ts)
```

B. Create a tsibble which is identical to the tourism tsibble from the tsibble package.

```
## # A tsibble: 6 x 5 [1Q]
## # Key:       Region, State, Purpose [1]
##   Quarter Region   State           Purpose  Trips
##     <qtr> <chr>    <chr>           <chr>    <dbl>
## 1 1998 Q1 Adelaide South Australia Business  135.
## 2 1998 Q2 Adelaide South Australia Business  110.
## 3 1998 Q3 Adelaide South Australia Business  166.
## 4 1998 Q4 Adelaide South Australia Business  127.
## 5 1999 Q1 Adelaide South Australia Business  137.
## 6 1999 Q2 Adelaide South Australia Business  200.
```

```r
key(tourism_ts)
```

```
## [[1]]
## Region
##
## [[2]]
## State
##
## [[3]]
## Purpose
```

```r
index(tourism_ts)
```

```
## Quarter
```

```r
# Using the tibble, we'll:
#   1. group by region and purpose
#   2. calculate the average trip by the group
#   3. Ungroup the data to remove the grouping structure
#   4. filter to display the entry that has the maximum value of trip_avg
tourism |>
  group_by(
    Region,
    Purpose
```

```r
  ) |>
  summarize(
    trip_avg = mean(Trips)
  ) |>
  ungroup() |>
  filter(
    trip_avg == max(trip_avg)
  )
```

C. Find what combination of Region and Purpose had the maximum number of overnight trips on average.

```
## `summarise()` has grouped output by 'Region'. You can override using the
## `.groups` argument.

## # A tibble: 1 x 3
##   Region Purpose  trip_avg
##   <chr>  <chr>       <dbl>
## 1 Sydney Visiting     747.
```

```r
# Using the tourism tibble, we'll:
#   1. group by state
#   2. summarize to create a total_trips feature
tourism_ts |>
  group_by(
    State
  ) |>
  summarize(
    total_trips = sum(Trips)
  )
```

D. Create a new tsibble which combines the Purposes and Regions, and just has total trips by State.

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter total_trips
##    <chr>   <qtr>       <dbl>
##  1 ACT   1998 Q1        551.
##  2 ACT   1998 Q2        416.
##  3 ACT   1998 Q3        436.
##  4 ACT   1998 Q4        450.
##  5 ACT   1999 Q1        379.
##  6 ACT   1999 Q2        558.
##  7 ACT   1999 Q3        449.
##  8 ACT   1999 Q4        595.
##  9 ACT   2000 Q1        600.
## 10 ACT   2000 Q2        557.
## # i 630 more rows
```

Question 2.8

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 Barrels from us_gasoline.

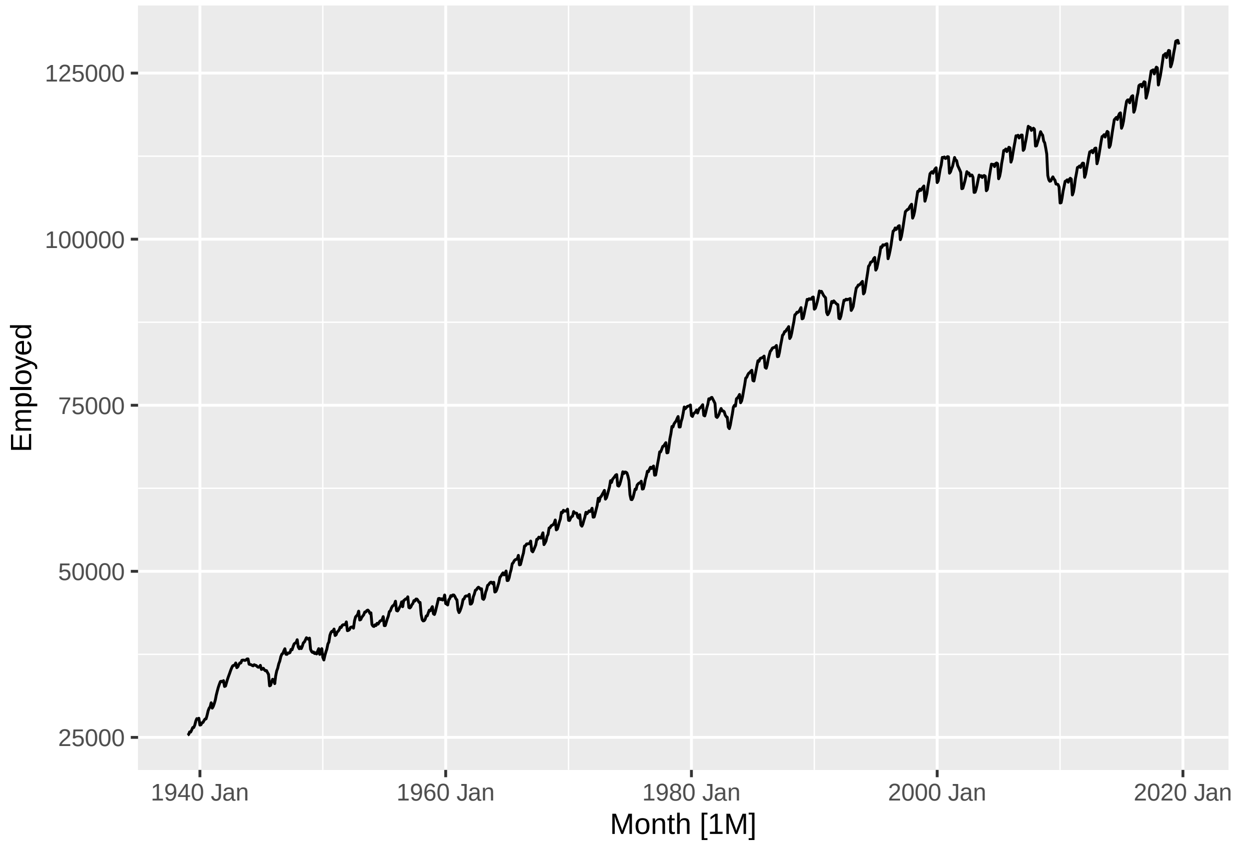
A. Can you spot any seasonality, cyclicity and trend?

B. What do you learn about the series?

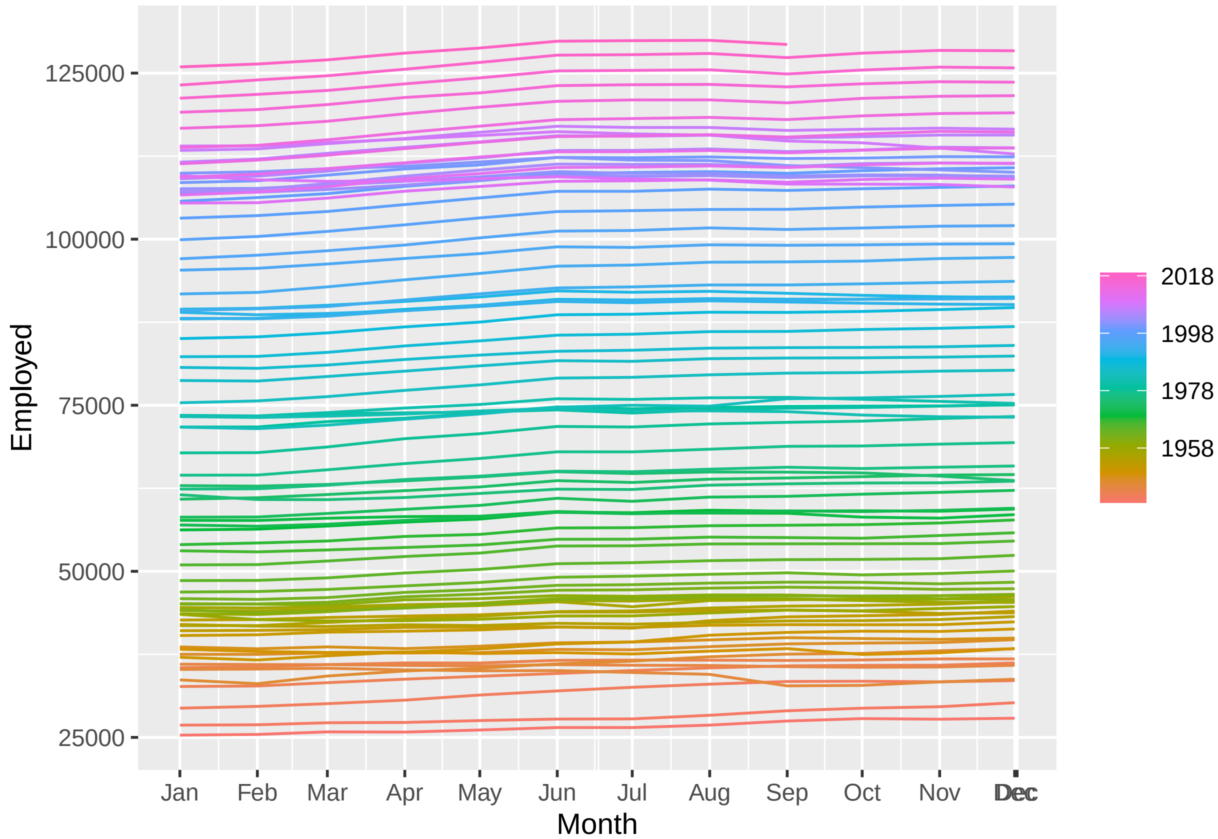
C. What can you say about the seasonal patterns?

D. Can you identify any unusual years? *All of these are answered in their respective cells below*

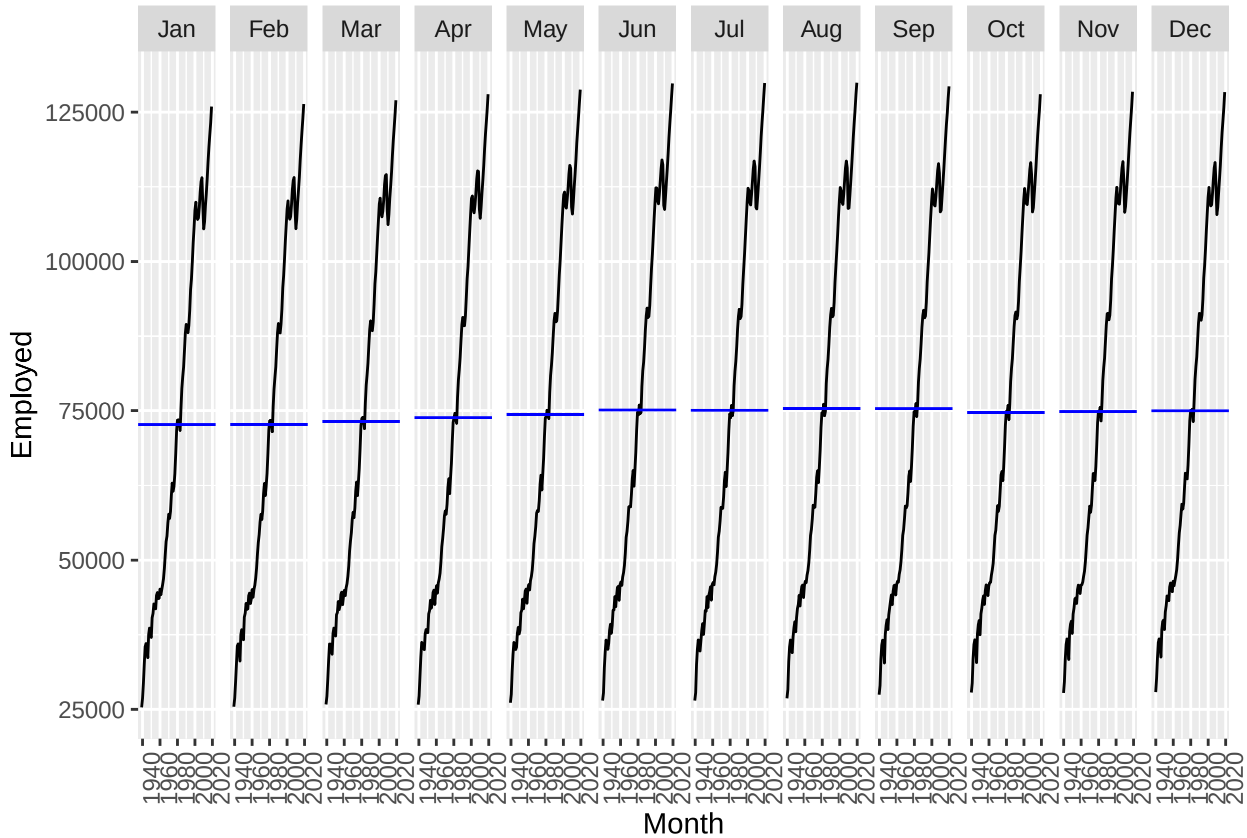
```
# loading our datasets
data(us_employment)
data(aus_production)
data(pelt)
data(PBS)
data(us_gasoline)

# Inspect our datasets
View(us_employment)
View(aus_production)
View(pelt)
View(PBS)
View(us_gasoline)

# us_employment: Check for seasonality, cyclicality and trend.
autoplot(
  us_employment |>
    filter(
      Title == "Total Private"
    ) |>
    select(
      Month,
      Employed
    ),
  Employed
)
```




```r
gg_season(
  us_employment |>
    filter(
      Title == "Total Private"
    ) |>
    select(
      Month,
      Employed
    ),
  Employed
)
```



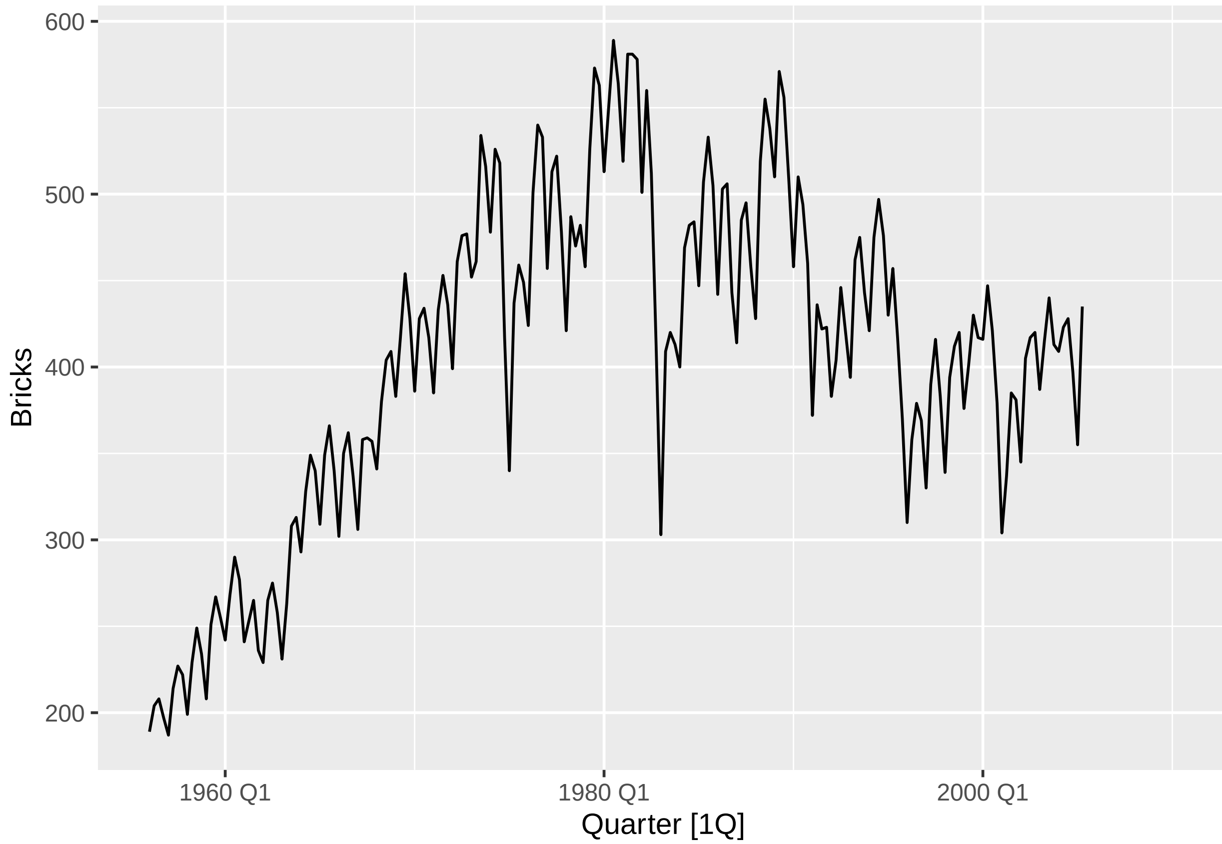

```r
gg_subseries(
  us_employment |>
    filter(
      Title == "Total Private"
    ) |>
    select(
      Month,
      Employed
    ),
  Employed
)
```



From the 1st plot above, we can see that the value of "Employed" increases as time goes on. In the seasonal plot, we can see that the rate of that growth seems to slow or even decrease in the summer months (June onward). This is especially true in the more recent years (values along the top of the plot).

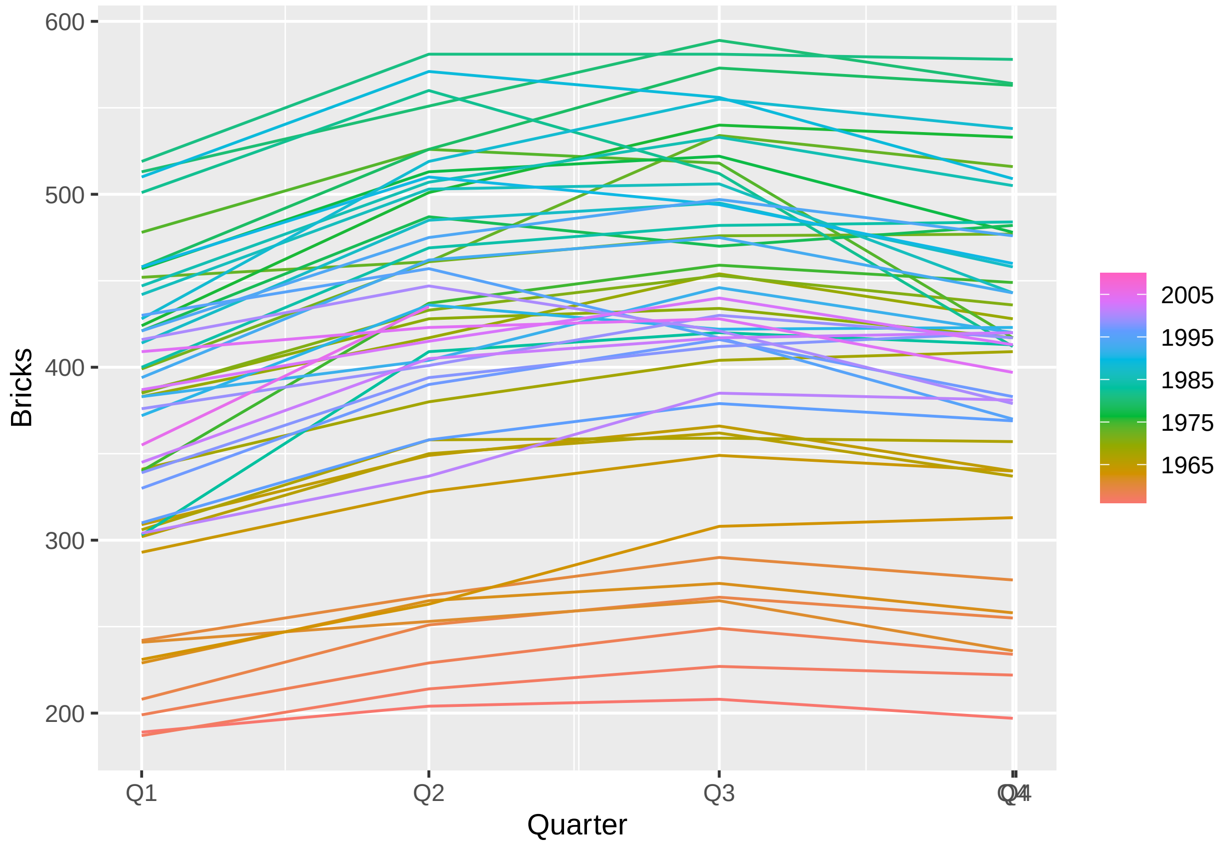
```
autoplot(
  aus_production |>
    select(
      Quarter,
      Bricks
    ),
  Bricks
)
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



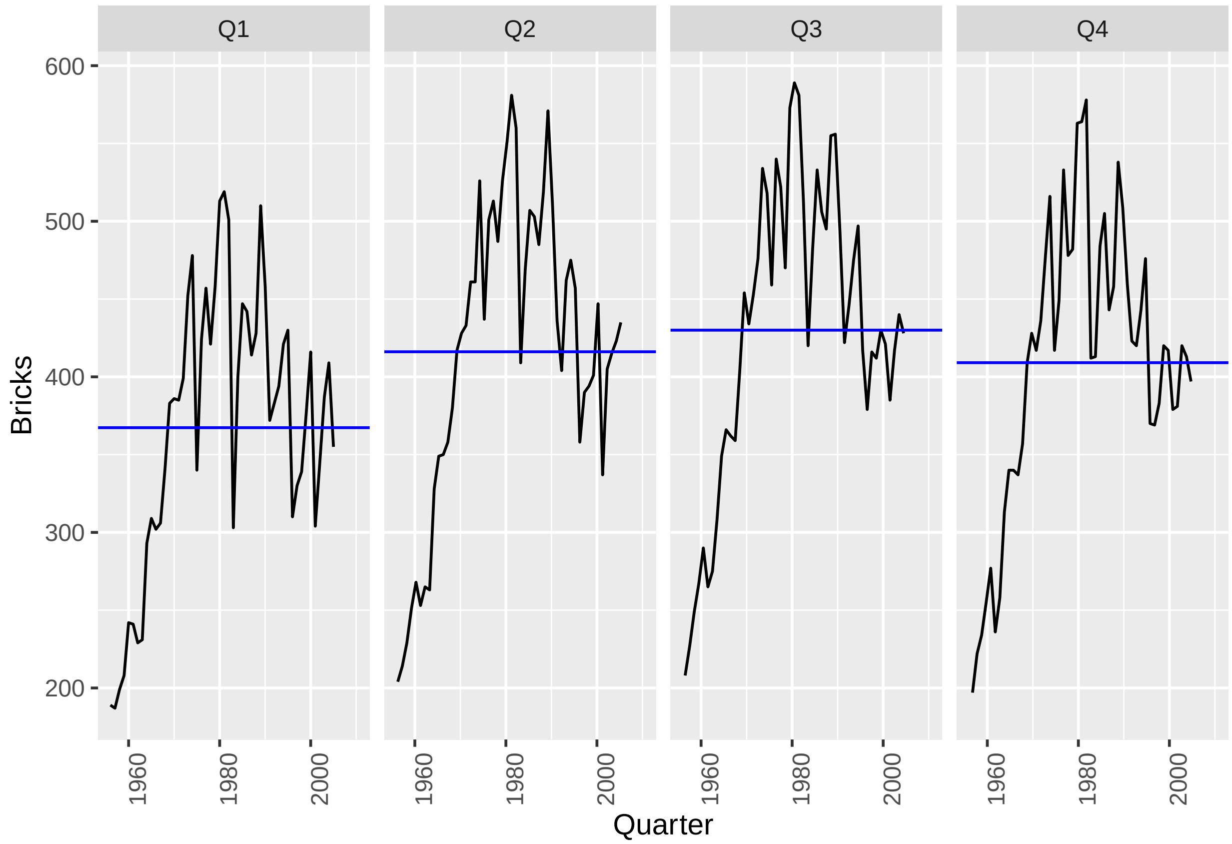

```r
gg_season(
  aus_production |>
    select(
      Quarter,
      Bricks
    ),
  Bricks
)
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



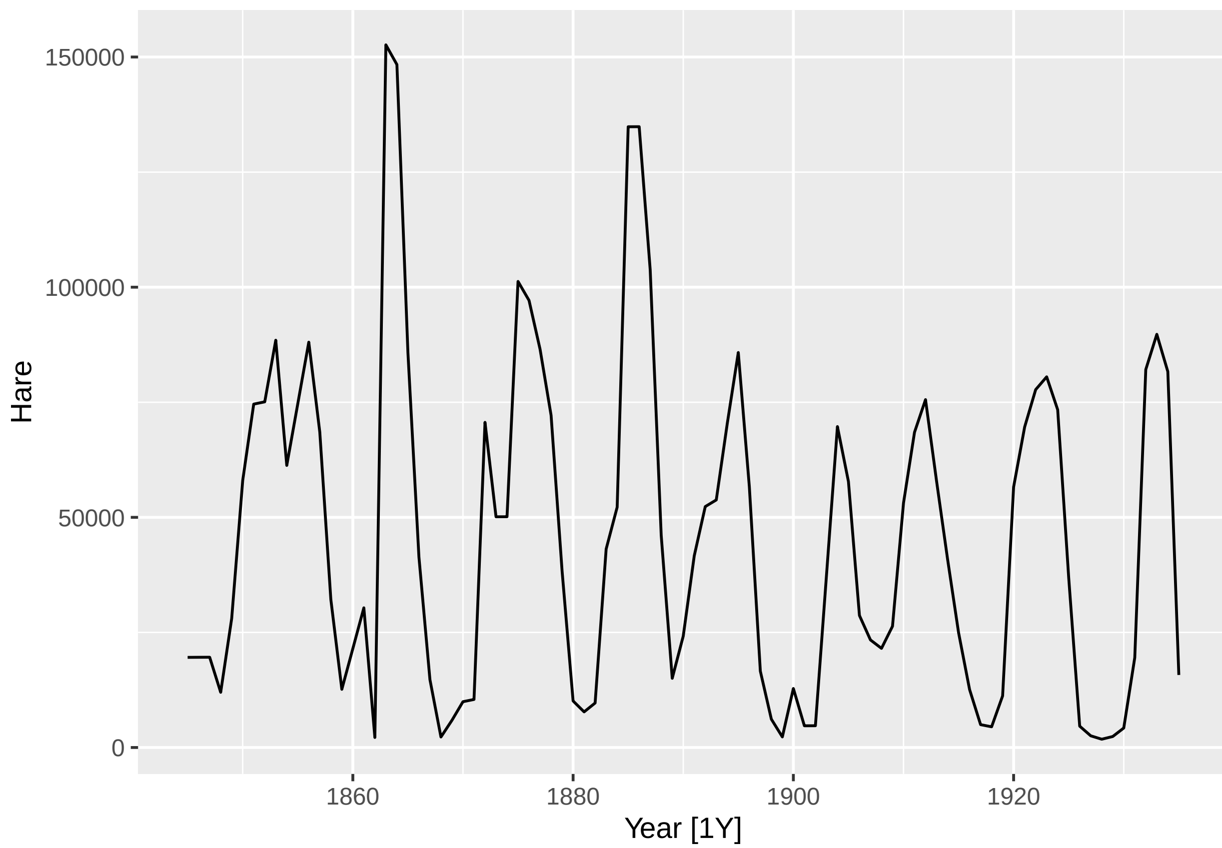

```r
gg_subseries(
  aus_production |>
    select(
      Quarter,
      Bricks
    ),
  Bricks
)
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```

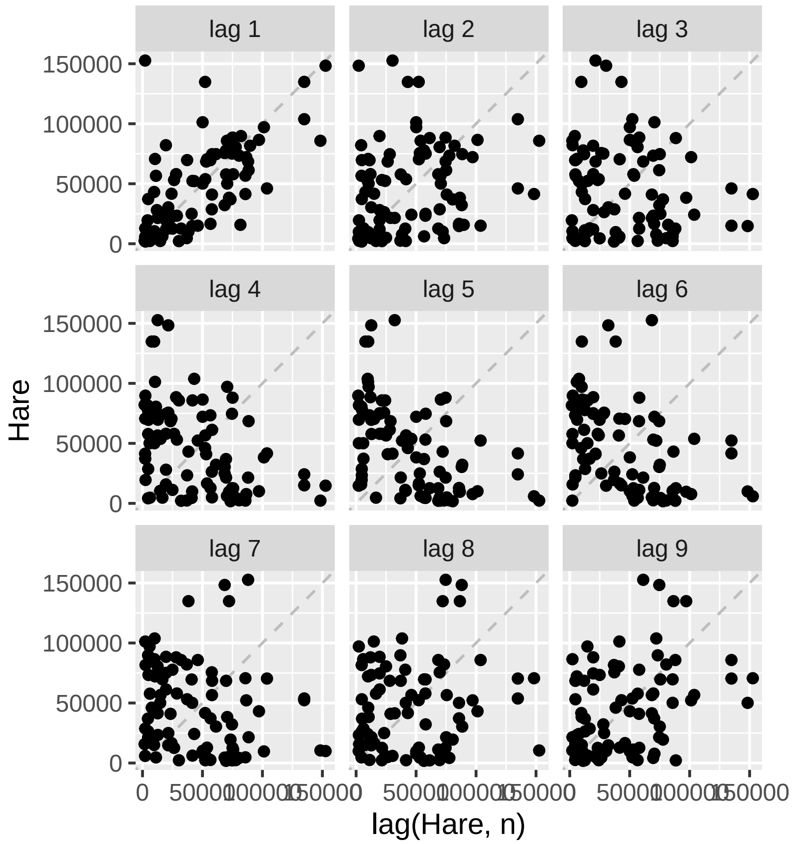


There seems to have been great growth from 1960 to 1980 and since then there seems to be a stagnation and a decrease that in 1990. In the season plot, we can see that Q1 often sees the lowest values and Q3 sees the highest. We can also see in the subseries plot where the mean is notably higher than the rest of the quarters. appears to begin

```
autoplot(
  pelt |>
    select(
      Year,
      Hare
    ),
  Hare
)
```




```
gg_lag(
  pelt |>
    select(
      Year,
      Hare
    ),
  Hare,
  geom = "point"
)
```




```
ACF(
  pelt |>
    select(
      Year,
      Hare
    ),
  Hare,
  geom = "point"
)
```

```
## Warning: The `...` argument of `PACF()` is deprecated as of feasts 0.2.2.
## i ACF variables should be passed to the `y` argument. If multiple variables are
##   to be used, specify them using `vars(...)`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

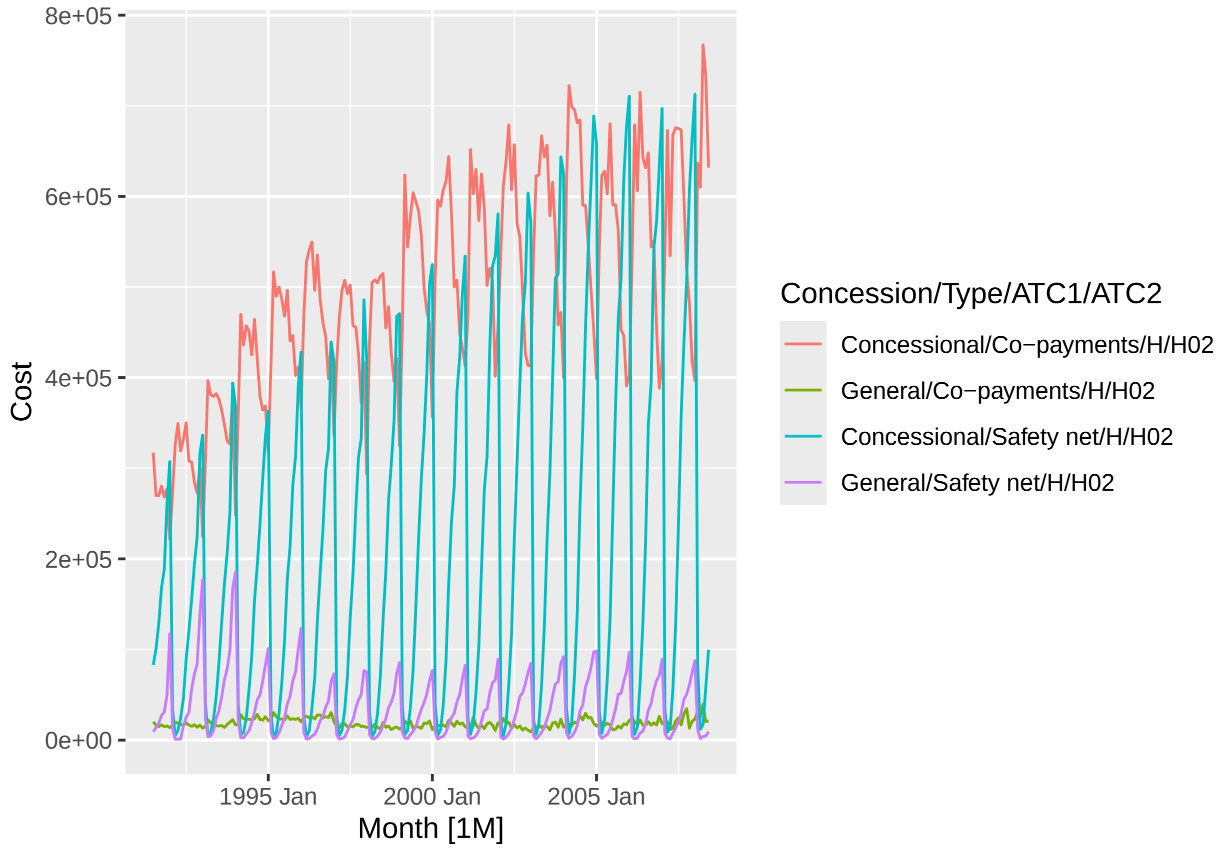
```
## Warning: ACF currently only supports one column, `Hare` will be used.
```

```
## # A tsibble: 19 x 2 [1Y]
##          lag     acf
##    <cf_lag>   <dbl>
##  1        1Y  0.658
##  2        2Y  0.214
##  3        3Y -0.155
```

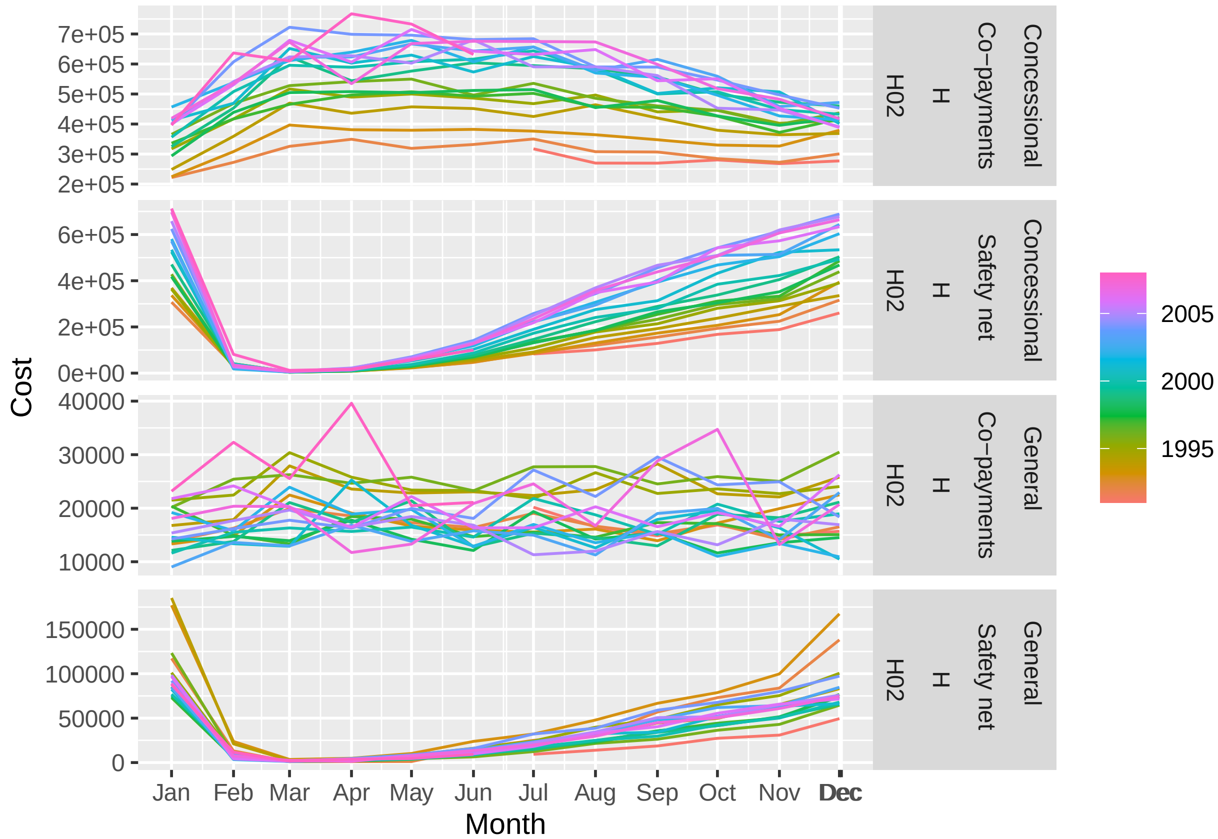
```
## 4        4Y -0.401
## 5        5Y -0.493
## 6        6Y -0.401
## 7        7Y -0.168
## 8        8Y  0.113
## 9        9Y  0.307
## 10      10Y  0.340
## 11      11Y  0.296
## 12      12Y  0.206
## 13      13Y  0.0372
## 14      14Y -0.153
## 15      15Y -0.285
## 16      16Y -0.295
## 17      17Y -0.202
## 18      18Y -0.0676
## 19      19Y  0.0956
```

The hare pelts dataset seems to have a series of peaks and valleys. From the granularity of the data, we won't be able to see any information on annual trends, but we can see that over the years their periods of strong year over year growth followed by periods of sharp declines. Looking at the lag plot, we can see that there seems to be a correlation with the data and lag 1, indicating that there may be a relationship between one year's trades and the next. Looking at the results of ACF, we see that lag1 is a bit of an exception and the correlation is not very good (65.8%).

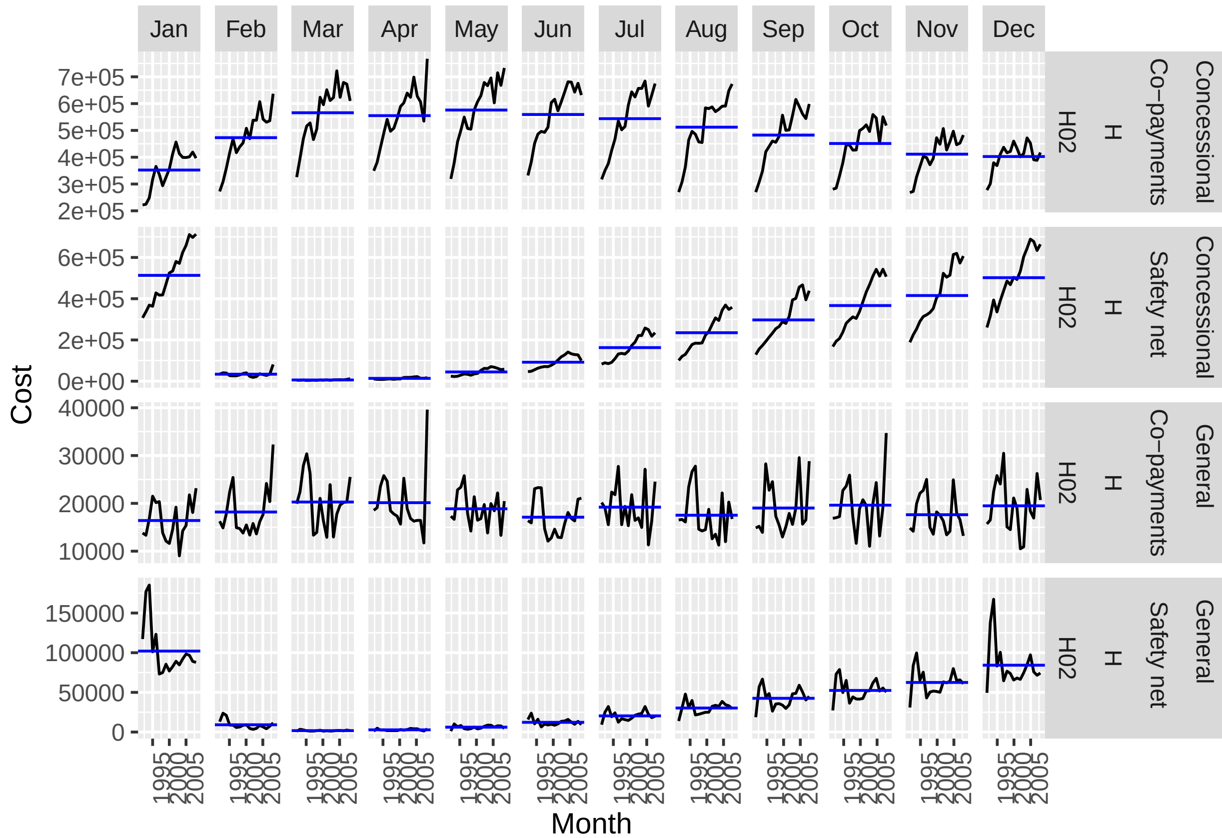
```r
autoplot(
  PBS |>
    filter(
      ATC2 == "H02"
    ) |>
    select(
      Month,
      Cost
    ),
  Cost
)
```




```
gg_season(
  PBS |>
    filter(
      ATC2 == "H02"
    ) |>
    select(
      Month,
      Cost
    ),
  Cost
)
```



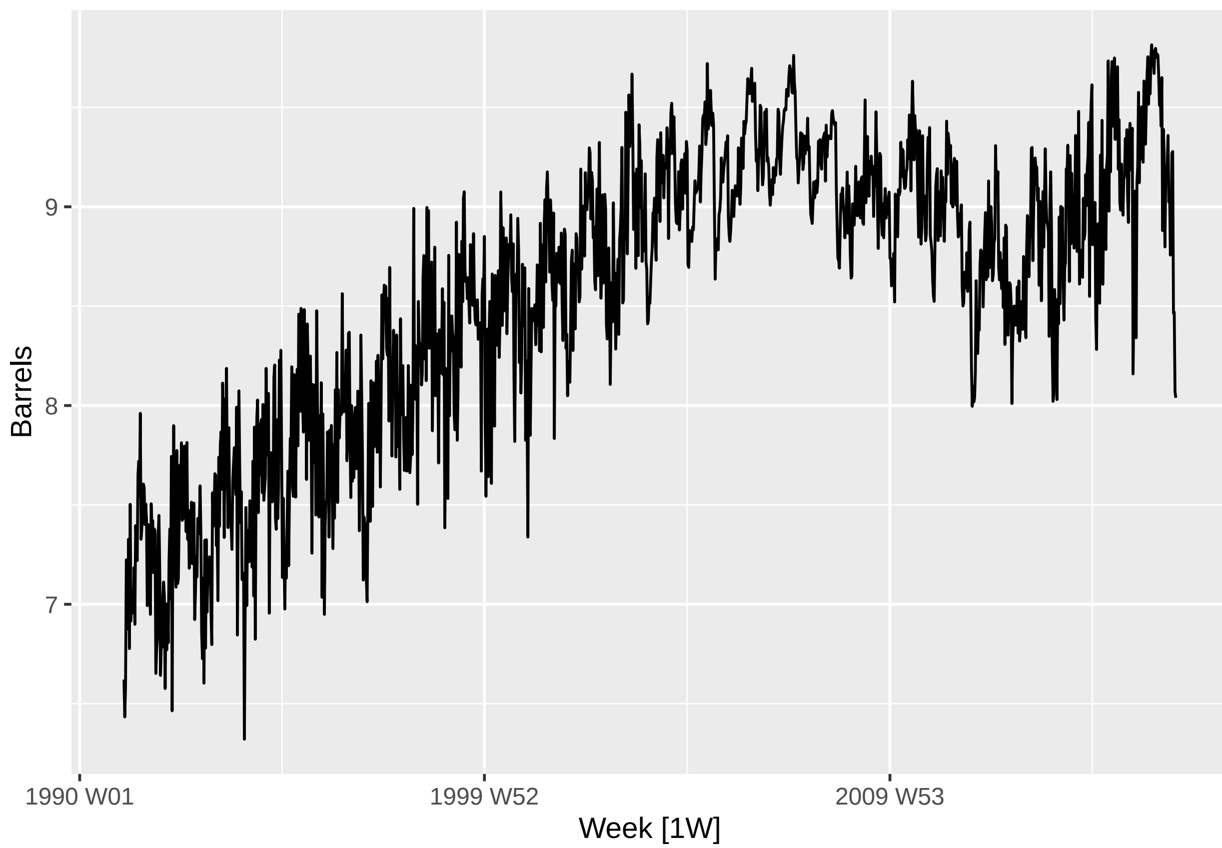

```r
gg_subseries(
  PBS |>
    filter(
      ATC2 == "H02"
    ) |>
    select(
      Month,
      Cost
    ),
  Cost
)
```



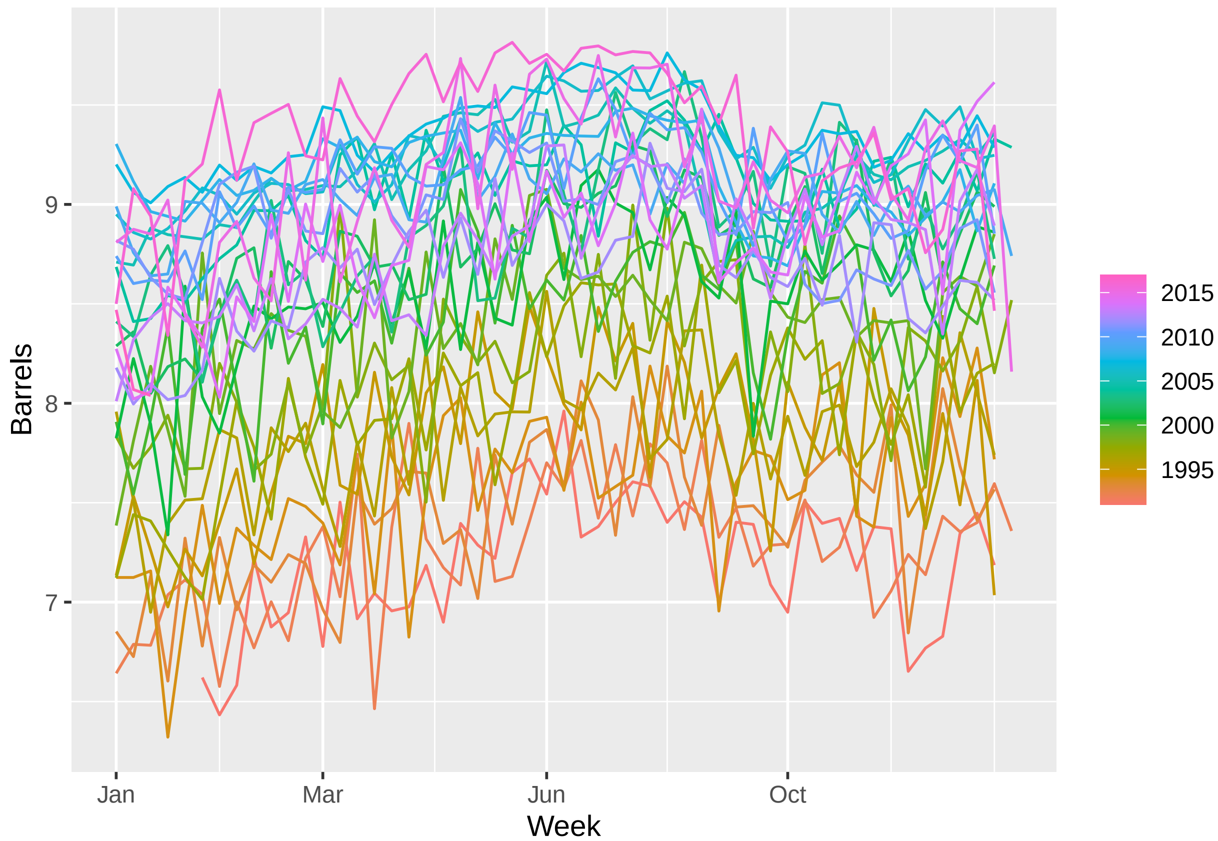
From the first plot we can see that there is some cyclicality to each of these metrics and that some of them remain relatively constant while others increase over time. This cyclicality can be observed better using the season plot where we see that the concessional and general safety nets have a sharp decline from January to February and then slowly ramps up for the remainder of the year. A new insight that we can see here is that the concessional co-payments seem to show an inverse relationship as the safety net metrics.

Lastly, from the sub-series we can see more evidence for the seasonality and trends we've noticed in the other plots.

```
autoplot(
  us_gasoline,
  Barrels
)
```



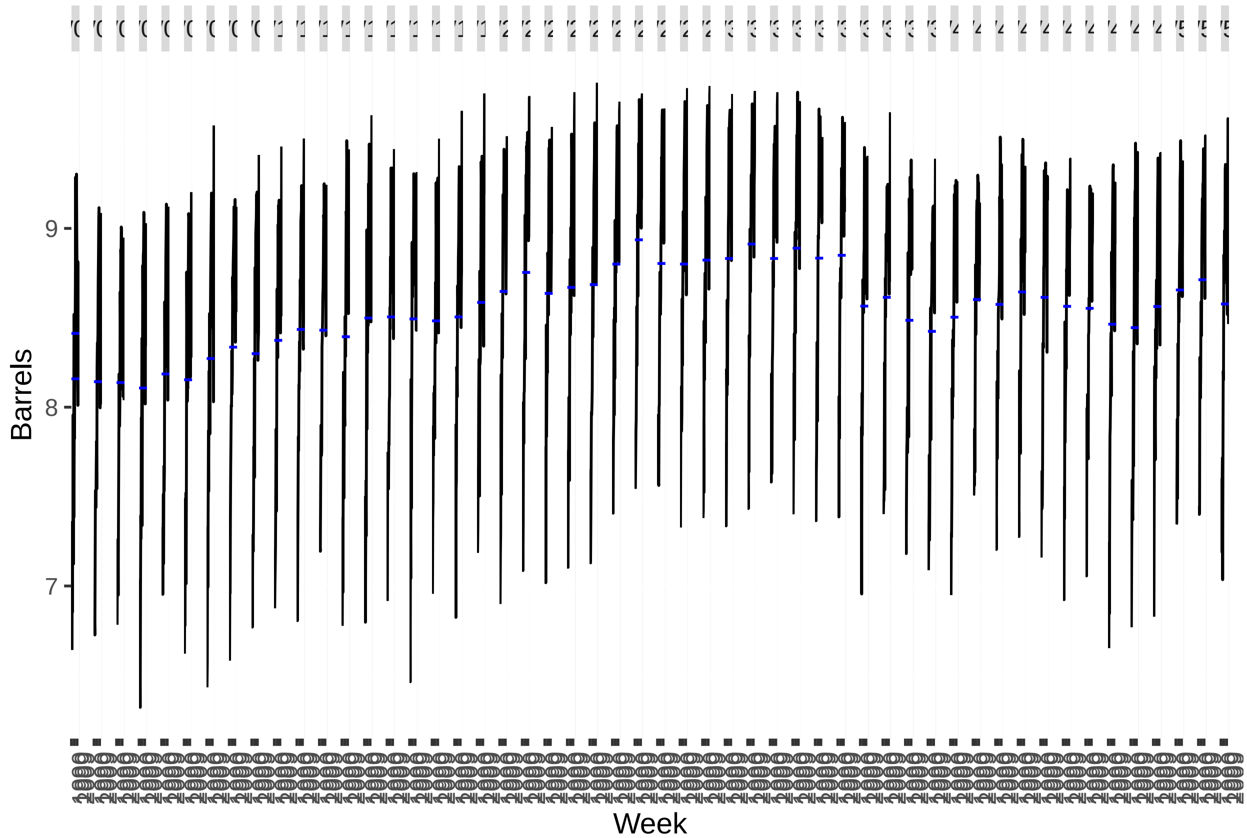

```
gg_season(
  us_gasoline,
  Barrels
)
```



From the first plot, we can see that the number of barrels per day increases until around 2006 where the trend reverses and it seems to decrease a bit and then remain steady.

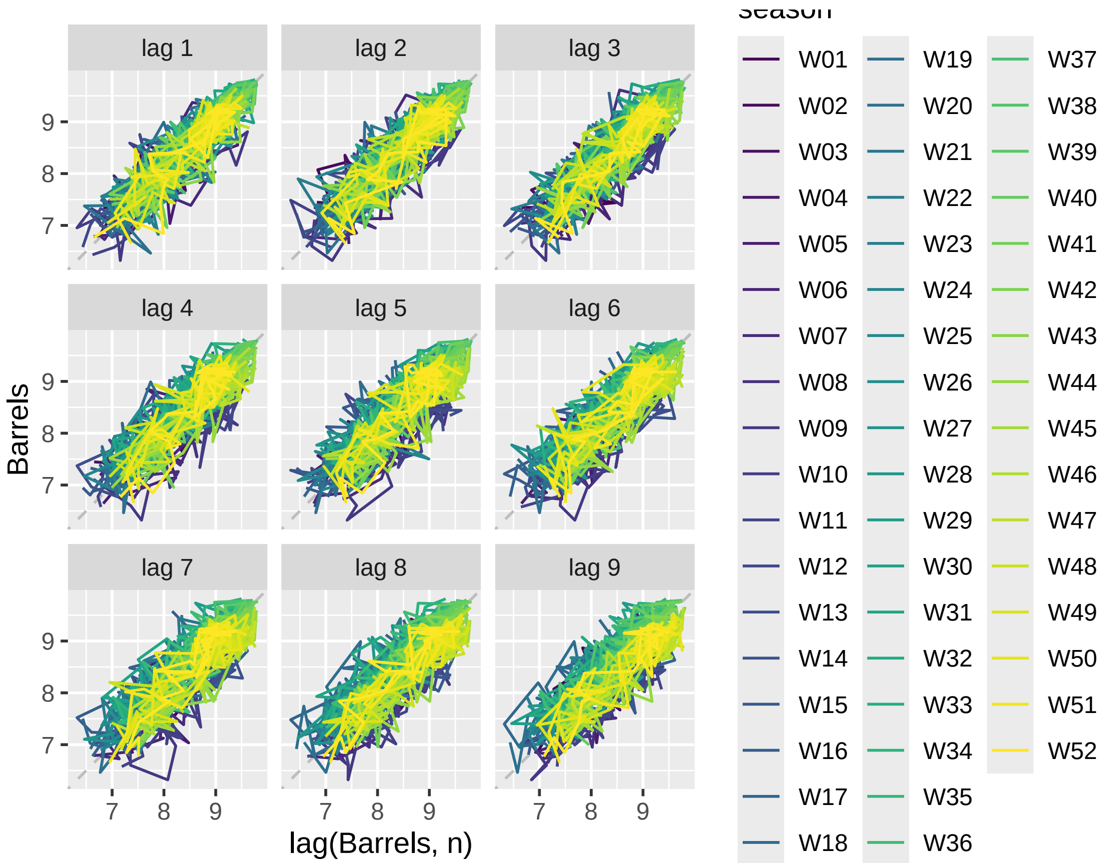
The values on a weekly basis jump greatly, making it difficult to see if there is any seasonality.

```
gg_subseries(
  us_gasoline,
  Barrels
)
```



By looking at the mean value (blue line) in the subseries plot, we can see that the number of barrels a day increases as the weeks go on until around week 35 which is where the mean begins to decrease.

```
gg_lag(
  us_gasoline,
  Barrels
)
```




```
print(
  ACF(
    us_gasoline,
    Barrels
  ),
  n = 100
)
```

```
## # A tsibble: 31 x 2 [1W]
##          lag   acf
##    <cf_lag> <dbl>
##  1        1W 0.893
##  2        2W 0.882
##  3        3W 0.873
##  4        4W 0.866
##  5        5W 0.847
##  6        6W 0.844
##  7        7W 0.832
##  8        8W 0.831
##  9        9W 0.822
## 10       10W 0.808
## 11       11W 0.801
## 12       12W 0.792
## 13       13W 0.783
## 14       14W 0.779
## 15       15W 0.769
```

```
## 16       16W 0.768
## 17       17W 0.763
## 18       18W 0.747
## 19       19W 0.736
## 20       20W 0.737
## 21       21W 0.724
## 22       22W 0.717
## 23       23W 0.709
## 24       24W 0.704
## 25       25W 0.701
## 26       26W 0.704
## 27       27W 0.699
## 28       28W 0.699
## 29       29W 0.700
## 30       30W 0.703
## 31       31W 0.708
```

From the lag plot and the ACF results, we can see that the correlation is consistently strong.