

assignment_1

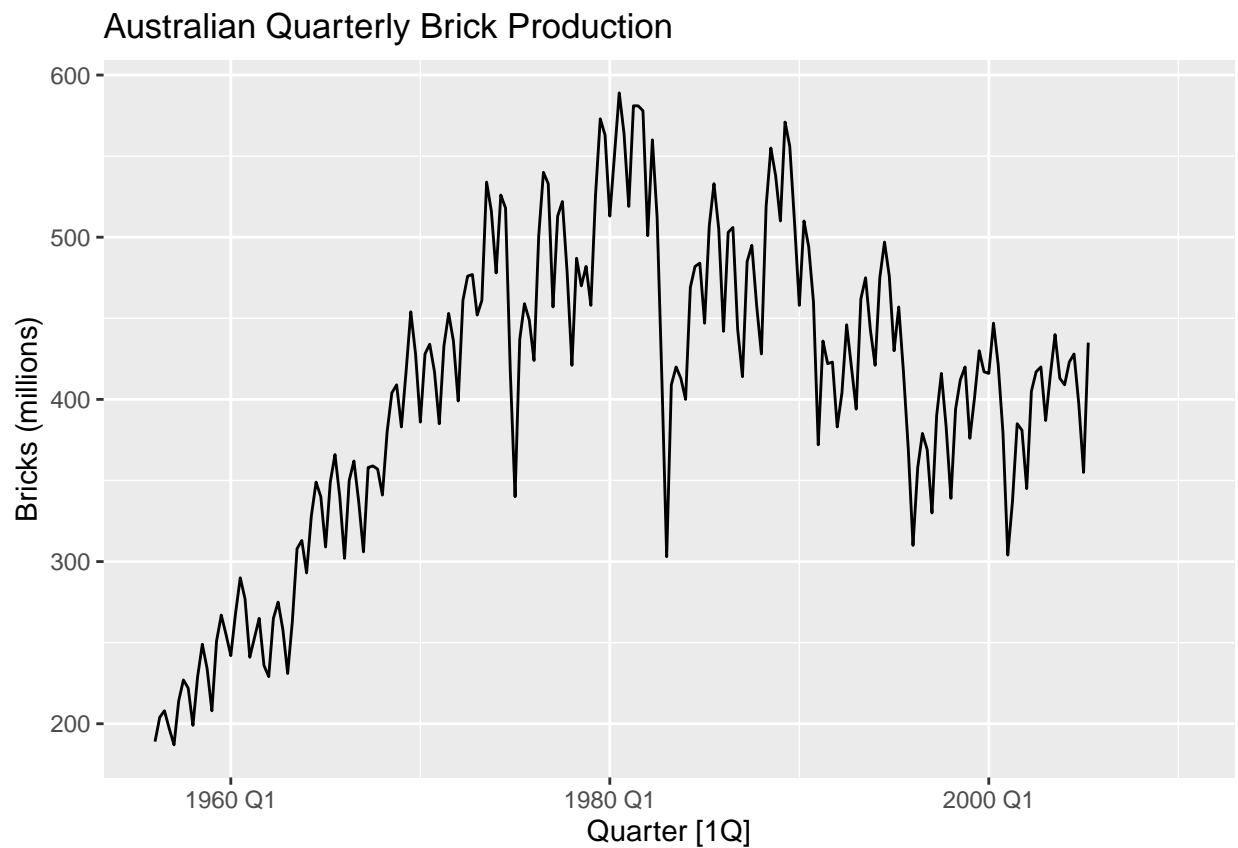
Matthew Tillmawitz

2024-09-07

2.1

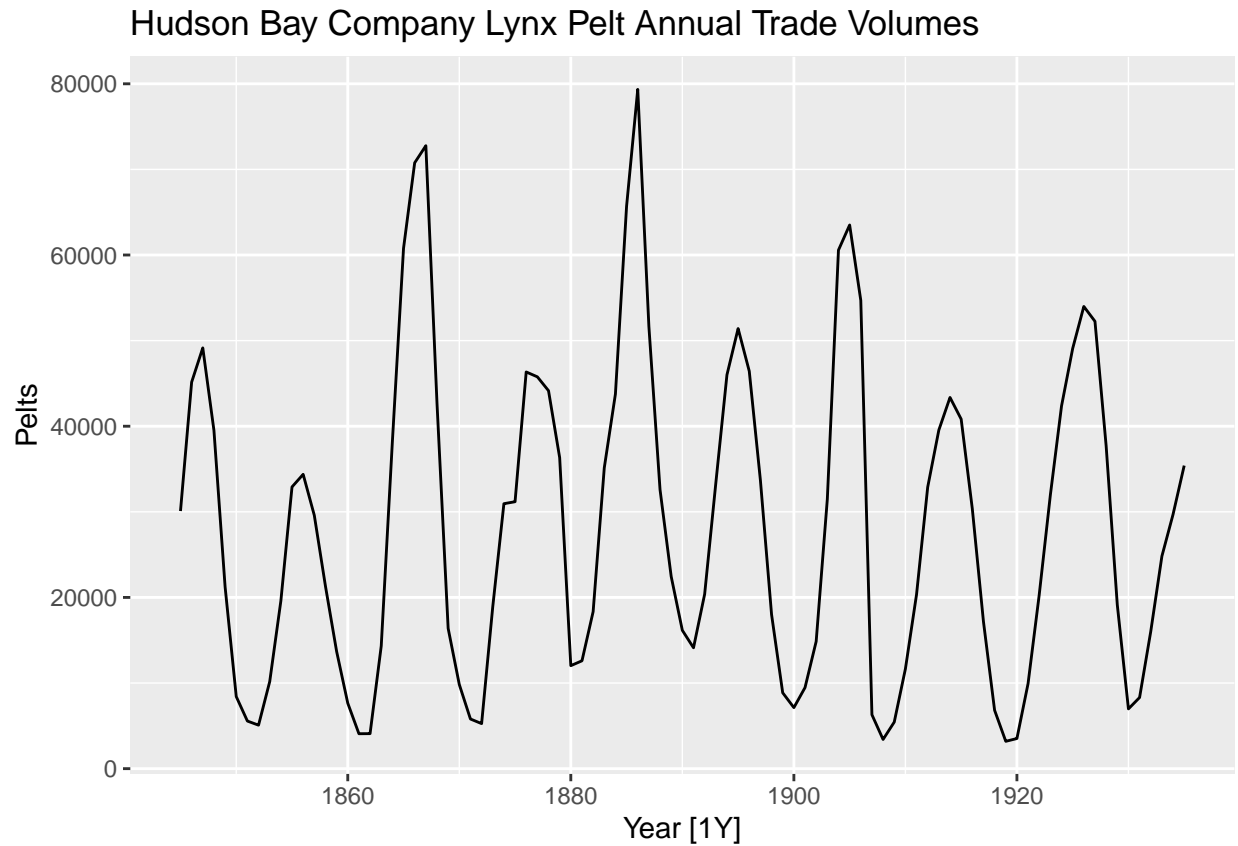
The time interval for the Bricks series is quarterly estimate of certain manufacturing production. The Brick output is measured in millions of bricks.

```
aus_production |>
  select(Quarter, Bricks) |>
  autoplot() +
  labs(y = "Bricks (millions)",
       title = "Australian Quarterly Brick Production")
```



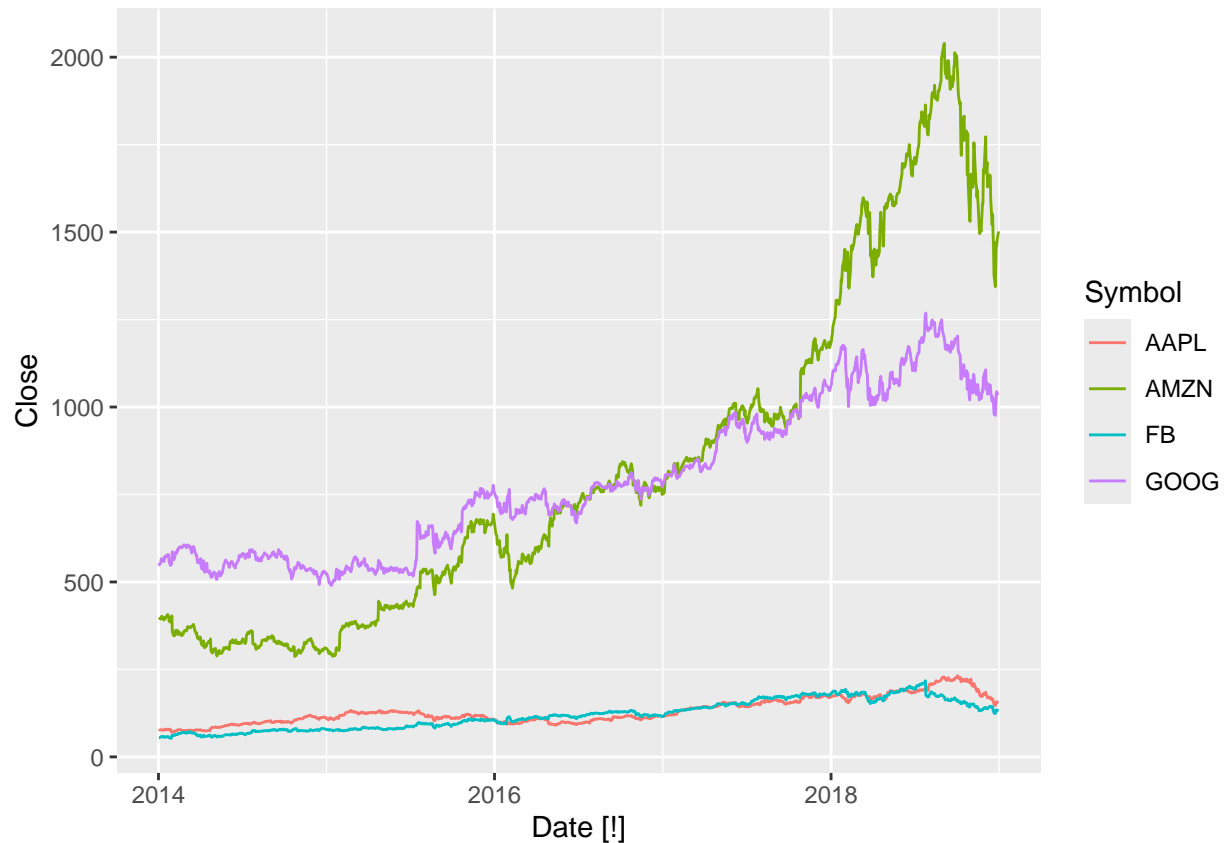
The pelt dataset contains the yearly trading records Lynx and Hare pelts by the Hudson Bay Company from 1845 to 1935. The pelt numbers are raw volume.

```
pelt |>
  select(Year, Lynx) |>
  autoplot() +
  labs(y = "Pelts",
       title = "Hudson Bay Company Lynx Pelt Annual Trade Volumes")
```



The `gafa_stock` dataset consists of stock data for Google, Amazon, Facebook, and Apple from 2014-2018. The prices are in USD and the time interval is daily and includes the opening, high, low, and close prices in addition to the adjusted close and total trade volume for the day.

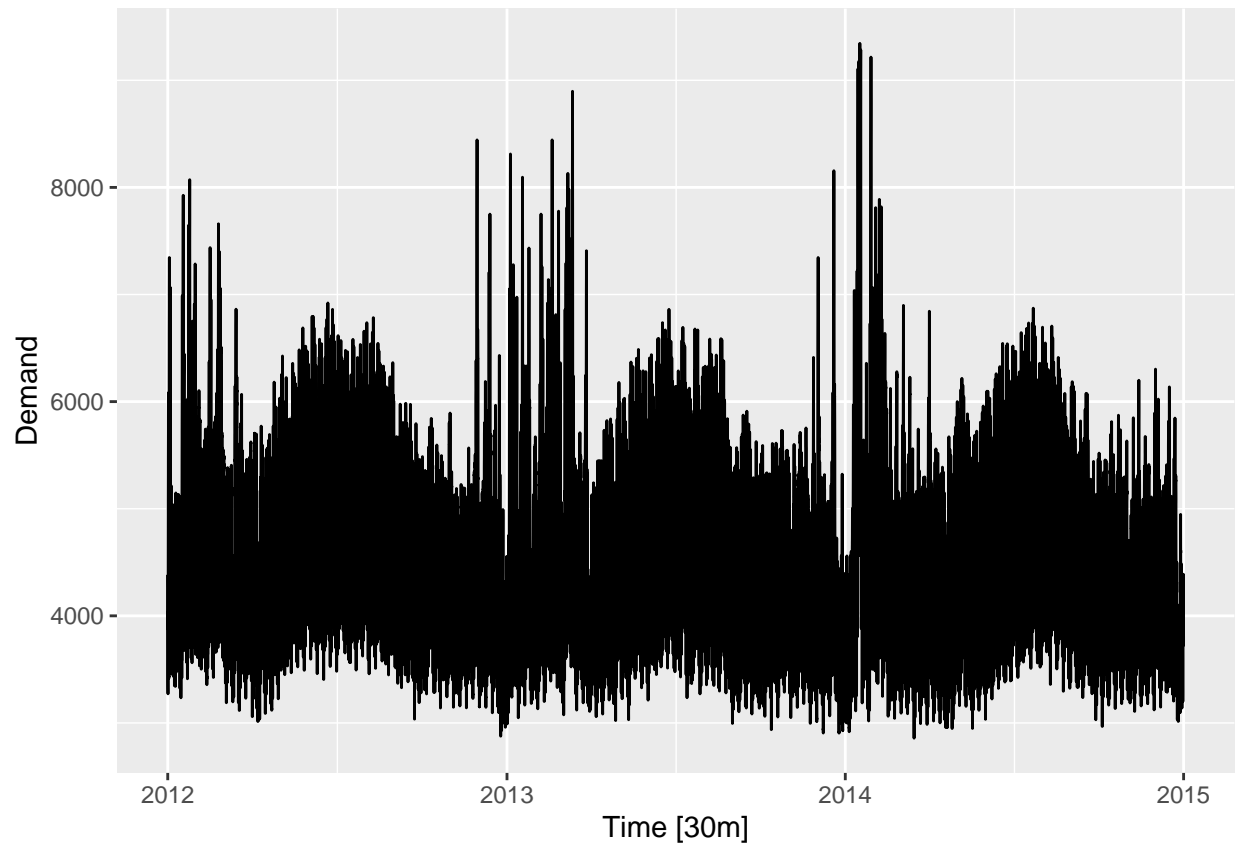
```
gafa_stock |>
  select(Symbol, Date, Close) |>
  autoplot()
```



The `vic_elec` dataset consists of the half-hourly operational demand for electricity in the Australian state of Victoria. In addition to the electricity demand in MWh it also tracks the temperature at a location in Melbourne and whether the day was a holiday. I did not rename the x-axis as it seemed appropriately titled already.

```
vic_elec |>
  select(Time, Demand) |>
  autoplot() |>
  labs(y = "Demand (MWh)",
       title = "Electricity Demand in Victoria, Australia")
```

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



```
##
## $y
## [1] "Demand (MWh)"
##
## $title
## [1] "Electricity Demand in Victoria, Australia"
##
## attr(,"class")
## [1] "labels"
```

2.2

The max Apple close price over the period is \$232.07.

```
aapl_max <- gafa_stock |>
  filter(Symbol == "AAPL") |>
  reframe(max_close = max(Close))
aapl_max
```

```
## # A tibble: 1 x 1
##   max_close
##   <dbl>
## 1      232.
```

The max Amazon close price over the period is \$2039.51

```
amzn_max <- gafa_stock |>
  filter(Symbol == "AMZN") |>
  reframe(max_close = max(Close))
amzn_max
```

```
## # A tibble: 1 x 1
##   max_close
##       <dbl>
## 1      2040.
```

The max Facebook close price over the period is \$217.50

```
fb_max <- gafa_stock |>
  filter(Symbol == "FB") |>
  reframe(max_close = max(Close))
fb_max
```

```
## # A tibble: 1 x 1
##   max_close
##       <dbl>
## 1       218.
```

The max Google close price over the period is \$1268.33

```
goog_max <- gafa_stock |>
  filter(Symbol == "GOOG") |>
  reframe(max_close = max(Close))
goog_max
```

```
## # A tibble: 1 x 1
##   max_close
##       <dbl>
## 1      1268.
```

A note that this question seems a bit dated as there is a much simpler way of getting these values with less code in a single tibble:

```
gafa_stock |>
  group_by(Symbol) |>
  reframe(max_close = max(Close))
```

```
## # A tibble: 4 x 2
##   Symbol max_close
## * <chr>       <dbl>
## 1 AAPL        232.
## 2 AMZN        2040.
## 3 FB          218.
## 4 GOOG        1268.
```

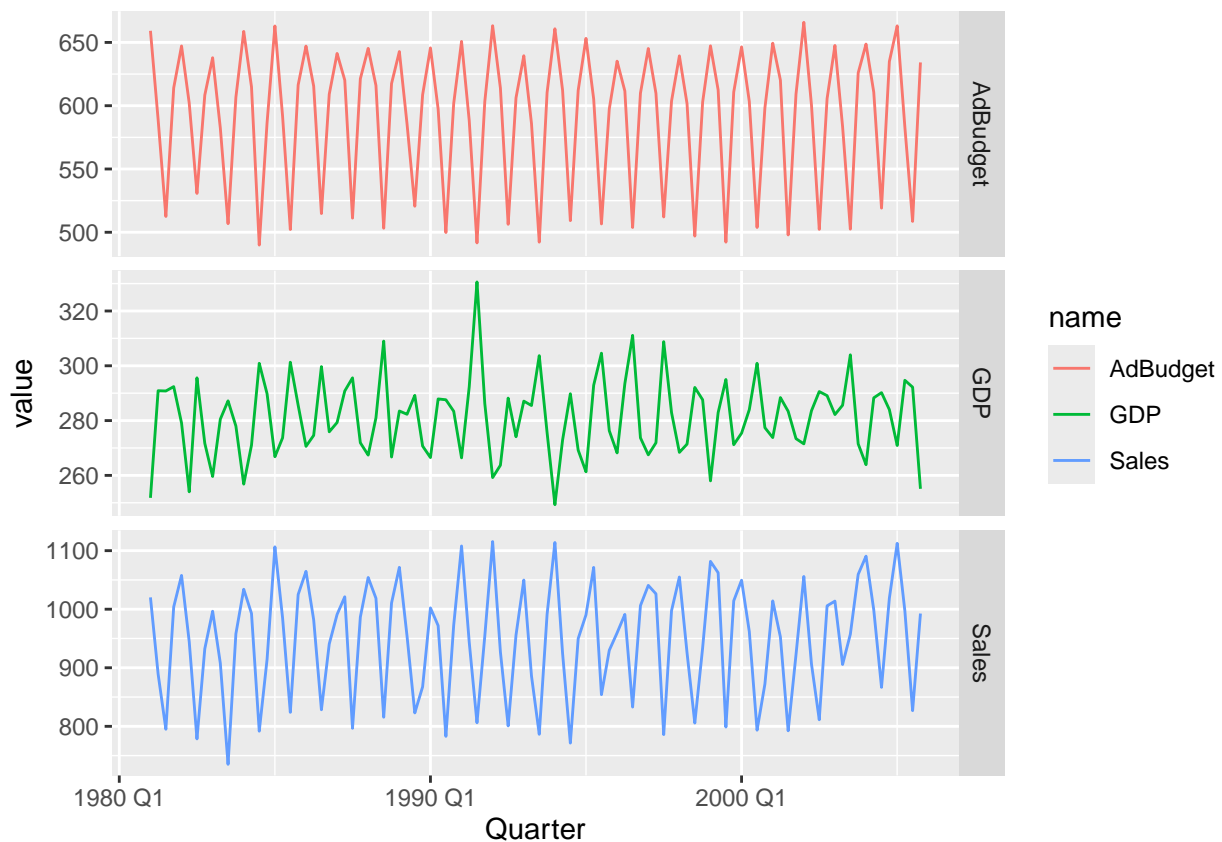
2.3

Below is code copied from the problem for comparison.

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

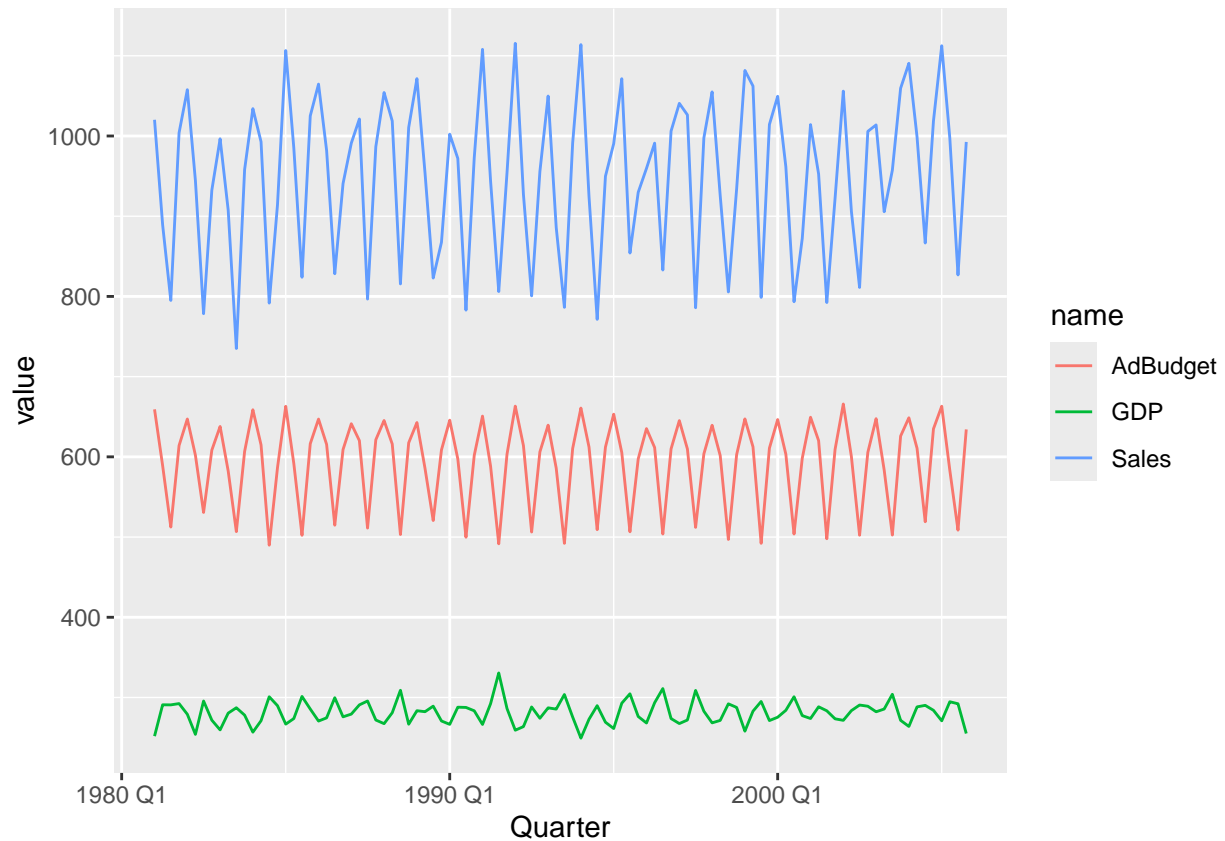
mytimeseries <- tute1 |>
  mutate(Quarter = yearquarter(Quarter)) |>
  as_tsibble(index = Quarter)

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



Without the `face_grid()` function the data is graphed together on the same plot with a single scale on the y axis instead of three separate plots each with their own relevant y axis scale.

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



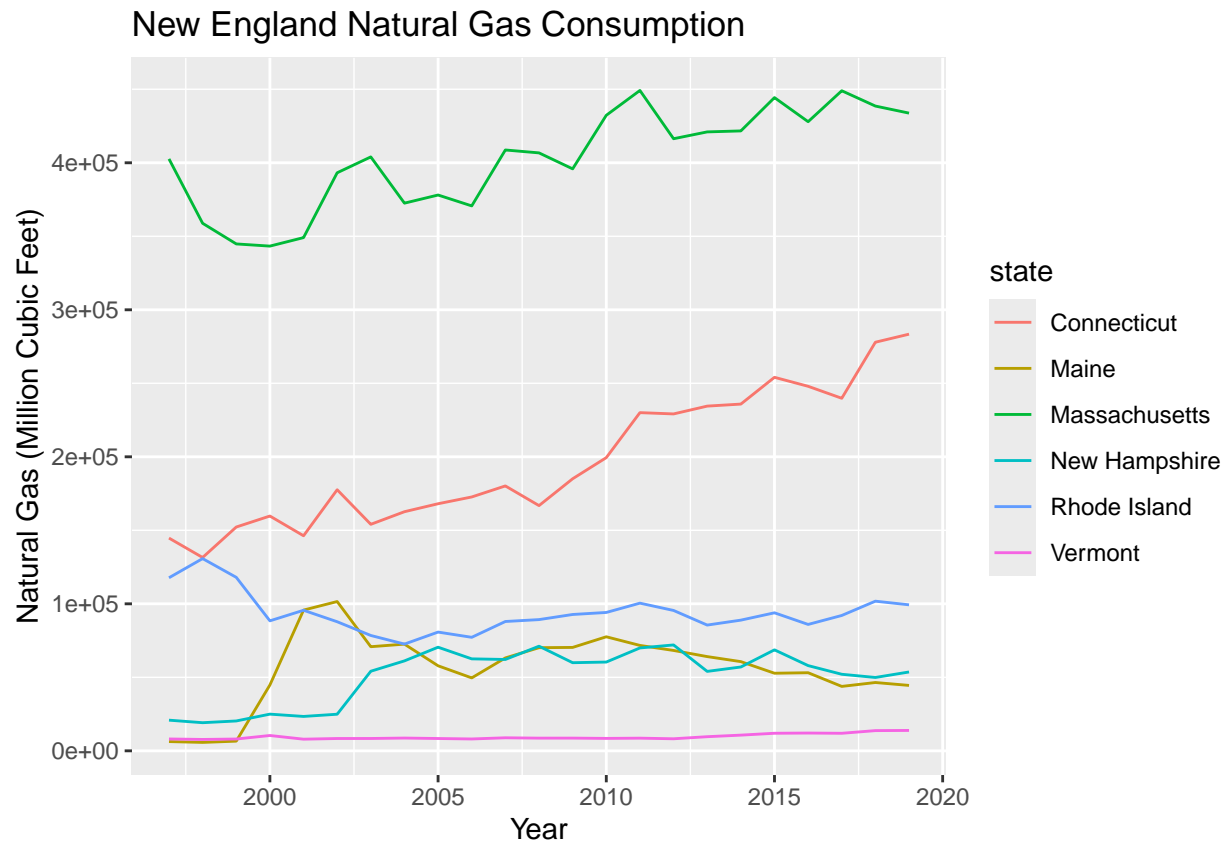
2.4

```
library(USgas)
us_tsibble <- us_total |>
  as_tsibble(key = state, index = year)
head(us_tsibble)
```

```
## # A tsibble: 6 x 3 [1Y]
## # Key:      state [1]
##   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
```

```
us_tsibble |>
  filter(state %in% c("Maine", "Vermont", "New Hampshire", "Massachusetts", "Connecticut", "Rhode Island"))
ggplot(aes(x = year, y = y, color = state)) +
  geom_line() +
  labs(title = "New England Natural Gas Consumption",
```

```
x = "Year",
y = "Natural Gas (Million Cubic Feet)"
```



2.5

Using the `all.equal()` function we can see that the generated `aus_tourism` tsibble is functionally identical to the `tourism` tsibble.

```
aus_tourism <- readxl::read_excel("tourism.xlsx")
aus_tourism <- aus_tourism |>
  mutate(Quarter = yearquarter(as.Date(Quarter))) |>
  as_tsibble(key = c(Region, State, Purpose), index = Quarter)
all.equal(tourism, aus_tourism)
```

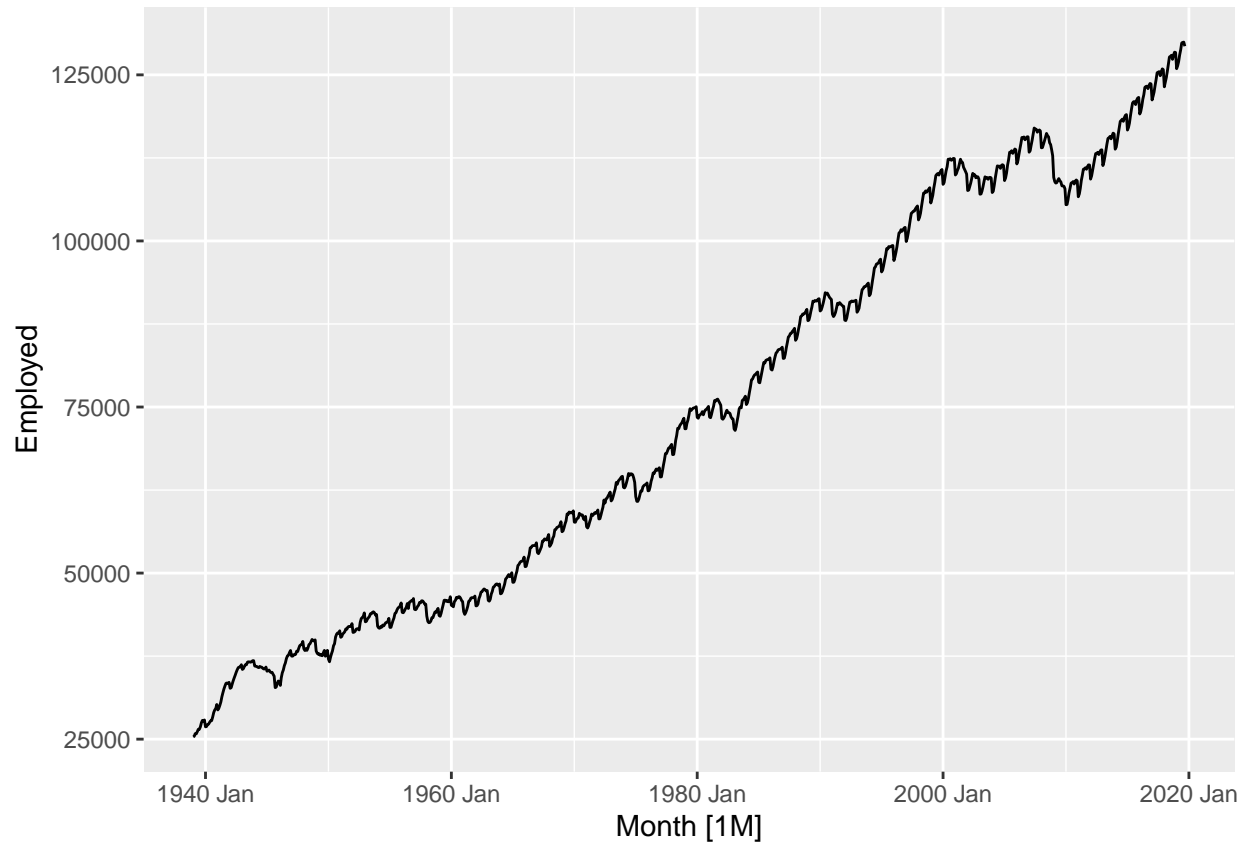
```
## [1] TRUE
```

2.8

Looking at the autoplot for US Private Employment we can clearly see a strong positive trend with some obvious seasonality. There are a few anomalous years that correspond to economic recessions and depressions as one would expect, such as the 2001 dot com bust and the 2008 financial crisis.

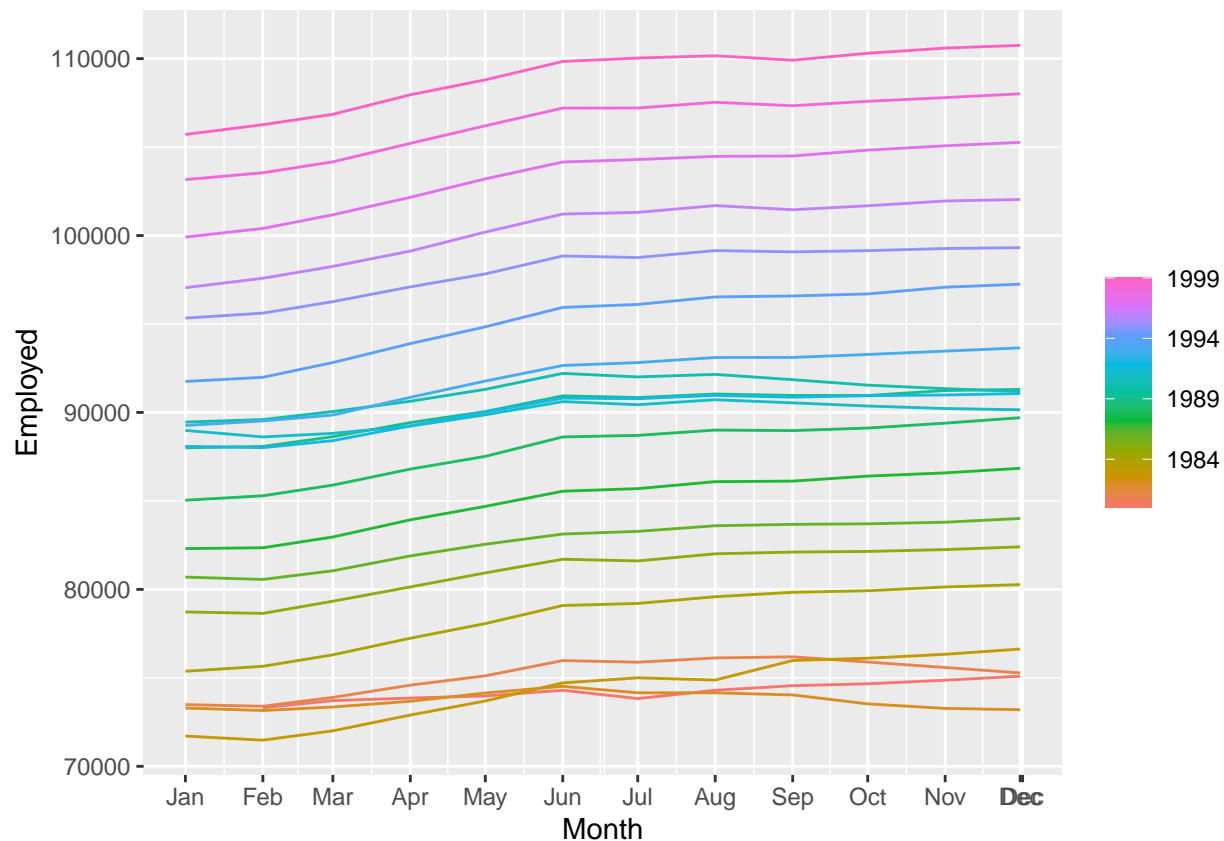

```
total_private_employment <- us_employment |>
  filter(Title == "Total Private") |>
  select(!Title)

total_private_employment |>
  autoplot()
```



Digging into the seasonality we can see a general trend of employment dropping at the start of each year from the previous year as the holiday season ends. Employment then tends to pick up until June when it will largely plateau before seeing another smaller uptick from September through the end of the year. This pattern is more easily seen by picking a shorter time window to allow for a smaller range of employment values as demonstrated below.

```
total_private_employment |>
  filter(Month > make_yearmonth(year = 1980L, month = 1L) & Month < make_yearmonth(year = 2000L, month = 1L)) |>
  gg_season(period = "year")
```

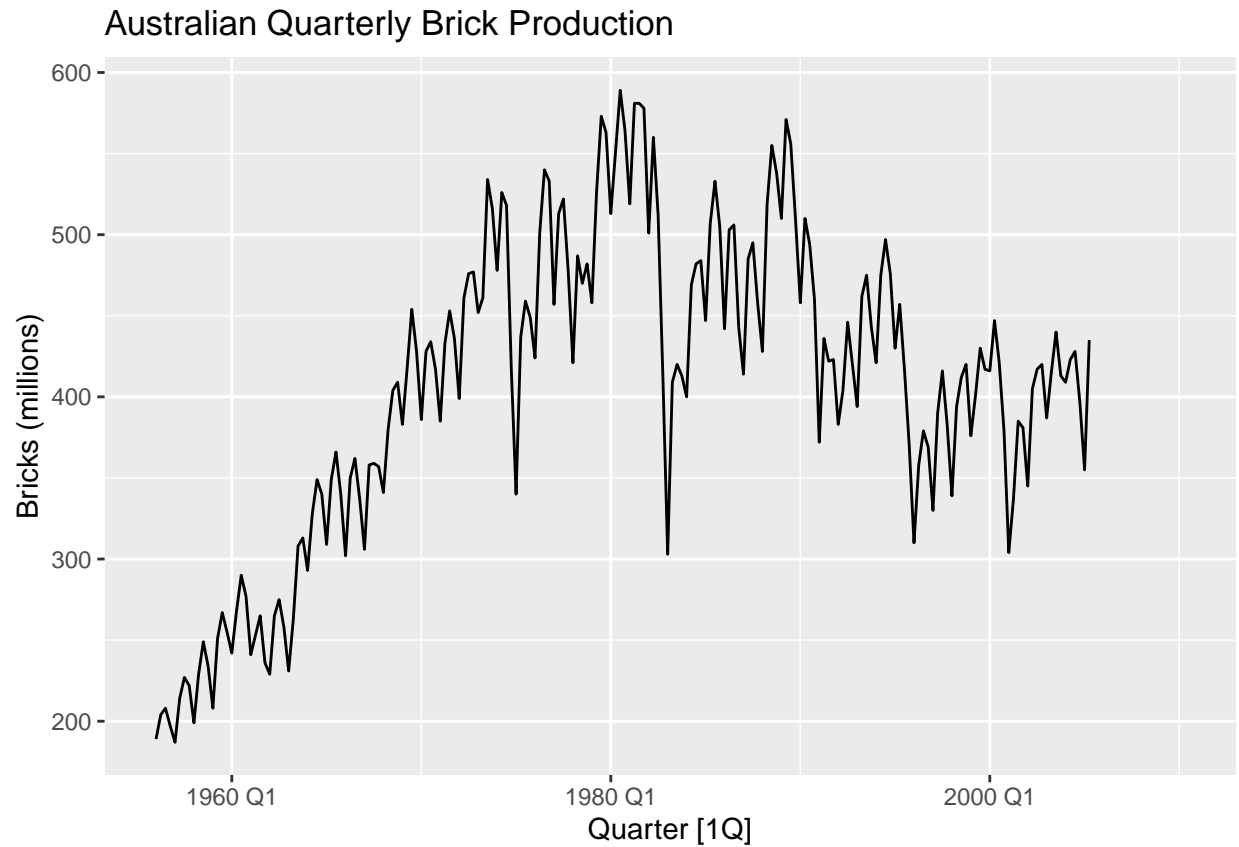


The lag, subseries, and ACF are largely uninteresting for this dataset given the strong positive trend of the data and do not tell us much more than the previous graphs.

Looking at the Australian Brick production from earlier, we can see a general positive trend until 1980 followed by a more gradual decline in production.

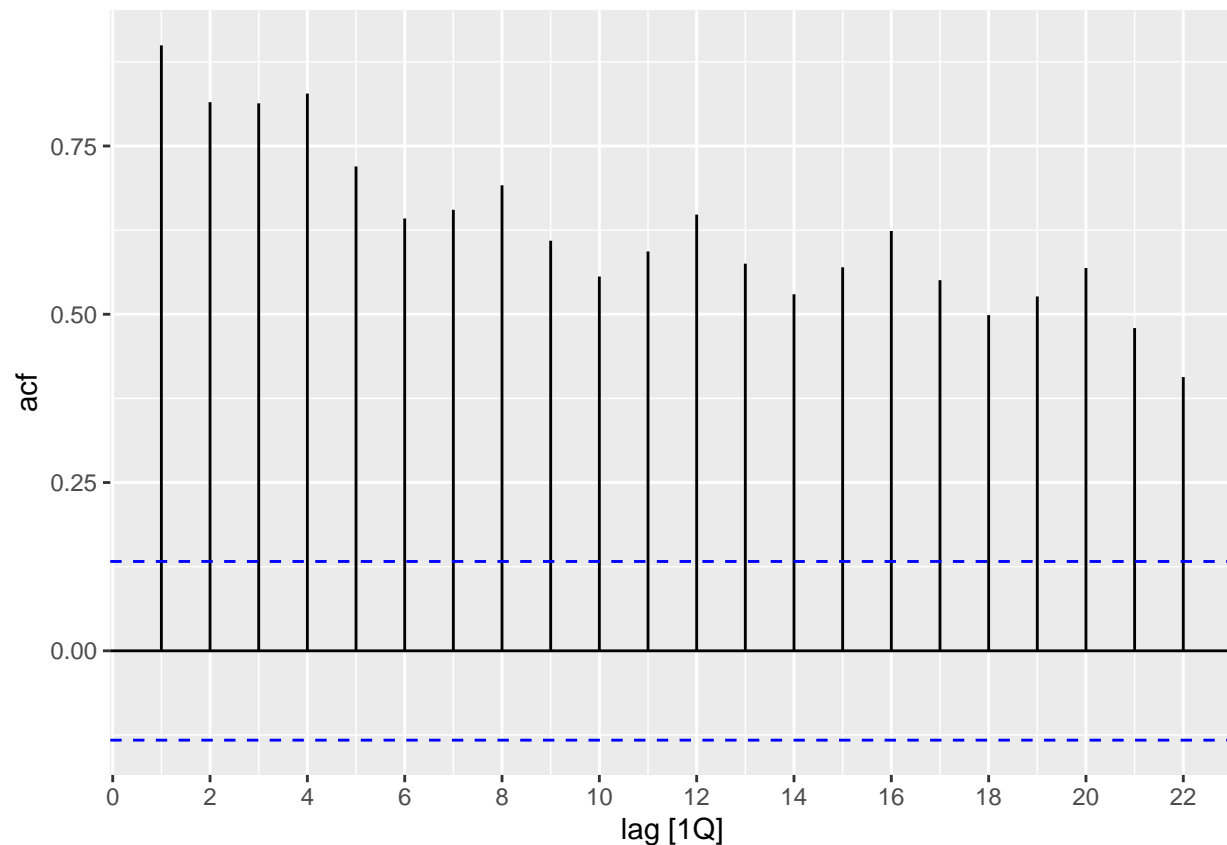
```
aus_bricks <- aus_production |>
  select(Quarter, Bricks)

aus_bricks |>
  autoplot() +
  labs(y = "Bricks (millions)",
       title = "Australian Quarterly Brick Production")
```



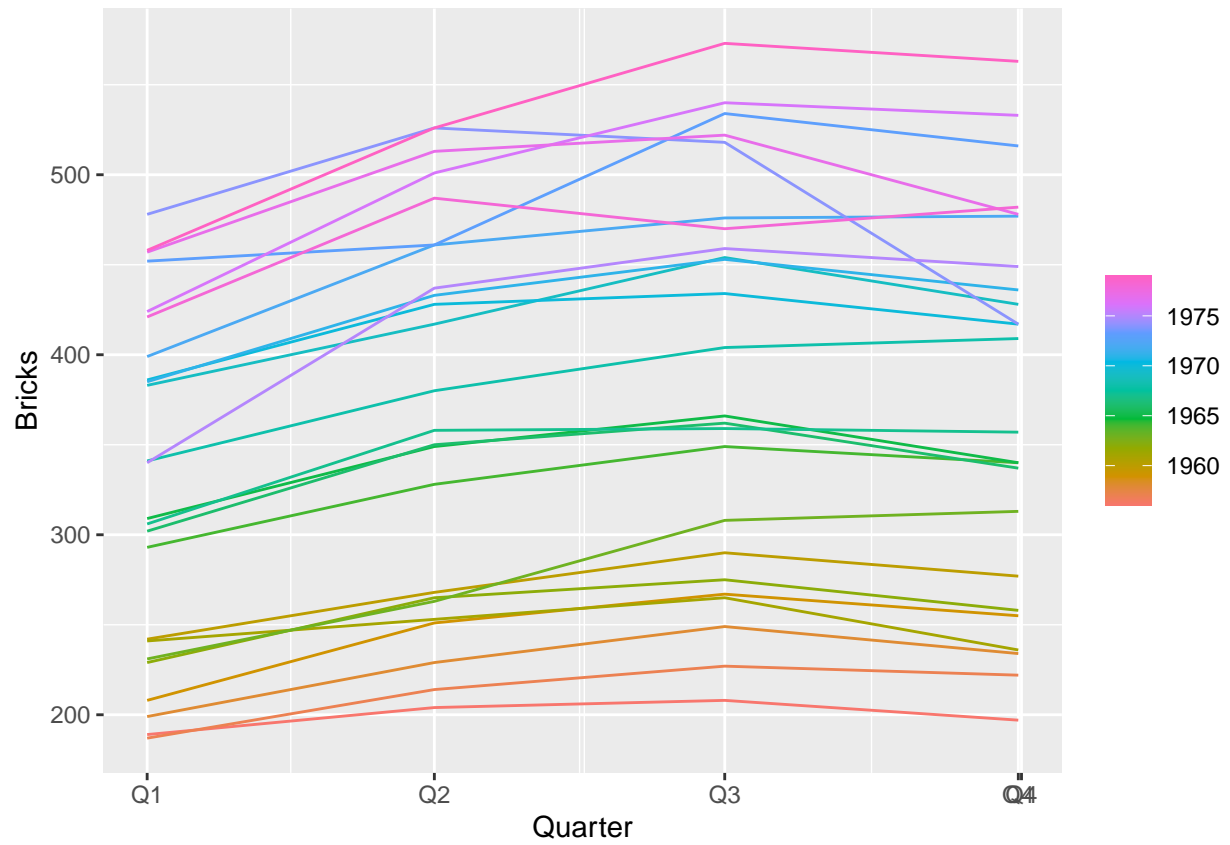
We can see the strong seasonality reflected in the plot of the autocorrelation with the highest correlation on the annual lag and the lowest correlation on the half year lag.

```
aus_bricks |>  
  ACF() |>  
  autoplot()
```

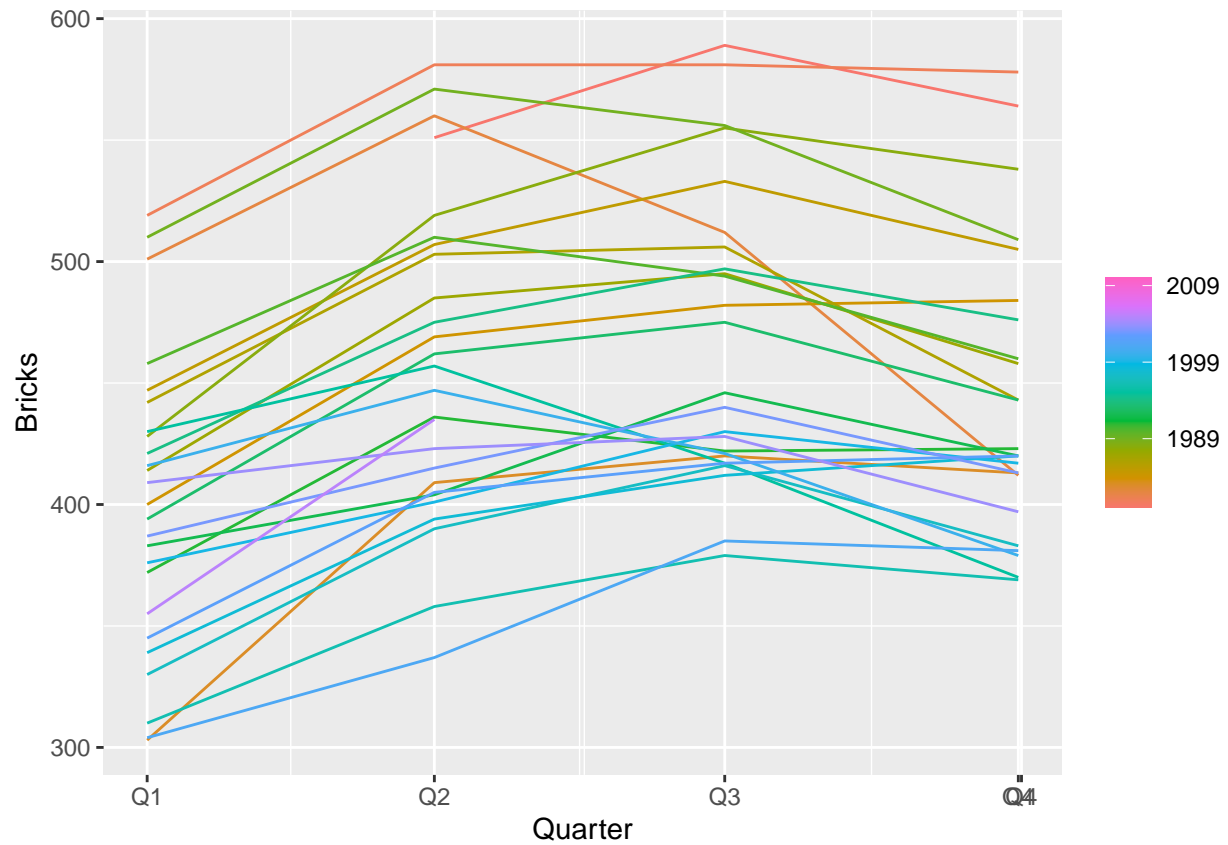


The following two graphs show an interesting phenomenon with regards to the seasonal peaks. We can see that during the period of increasing production up to about 1980 production tended to peak in Q3, whereas in the period of decline after 1980 the peak tends to occur more frequently in Q2.

```
aus_bricks |>
  filter(Quarter < make_yearquarter(year = 1980L, quarter = 1L)) |>
  gg_season()
```



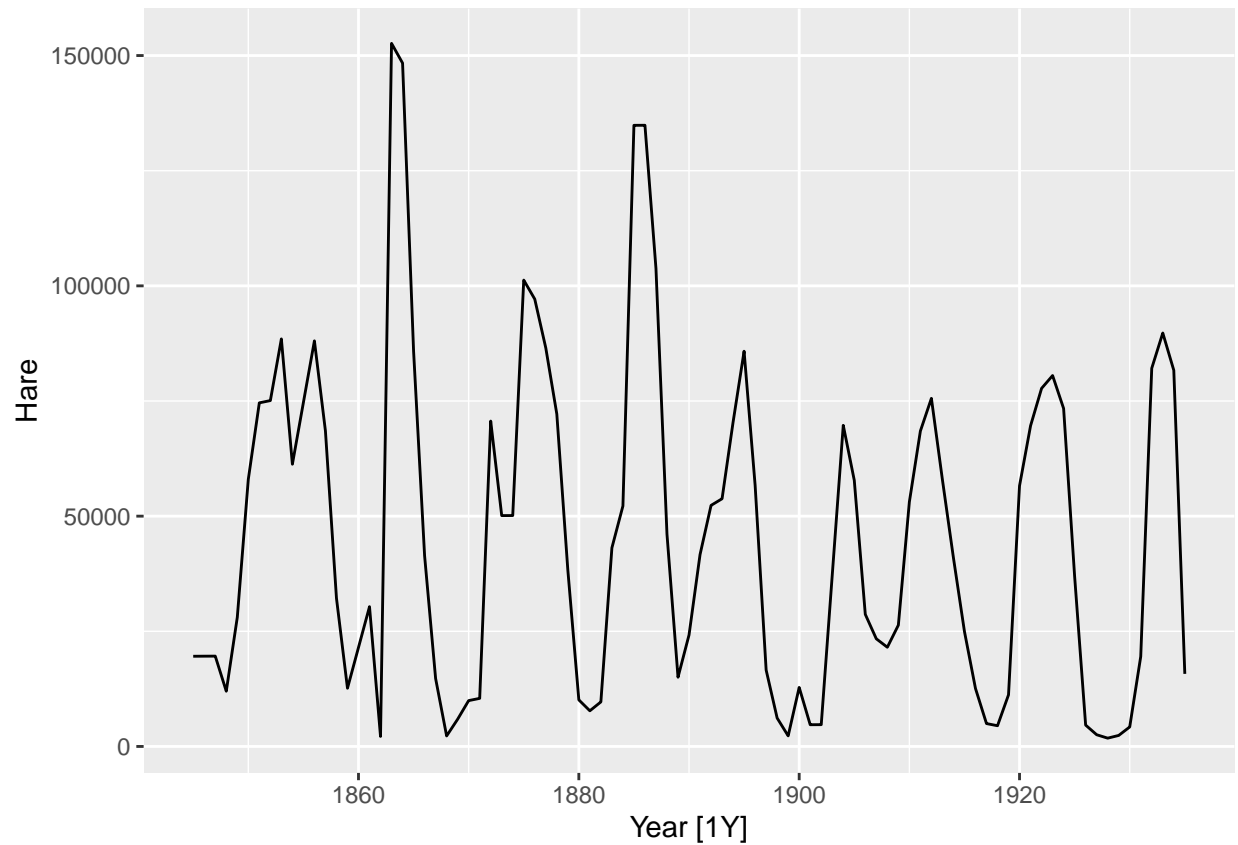
```
aus_bricks |>
  filter(Quarter > make_yearquarter(year = 1980L, quarter = 1L)) |>
  gg_season()
```



The Hare portion of the pelts data shows a flat trend with strong seasonal behavior.

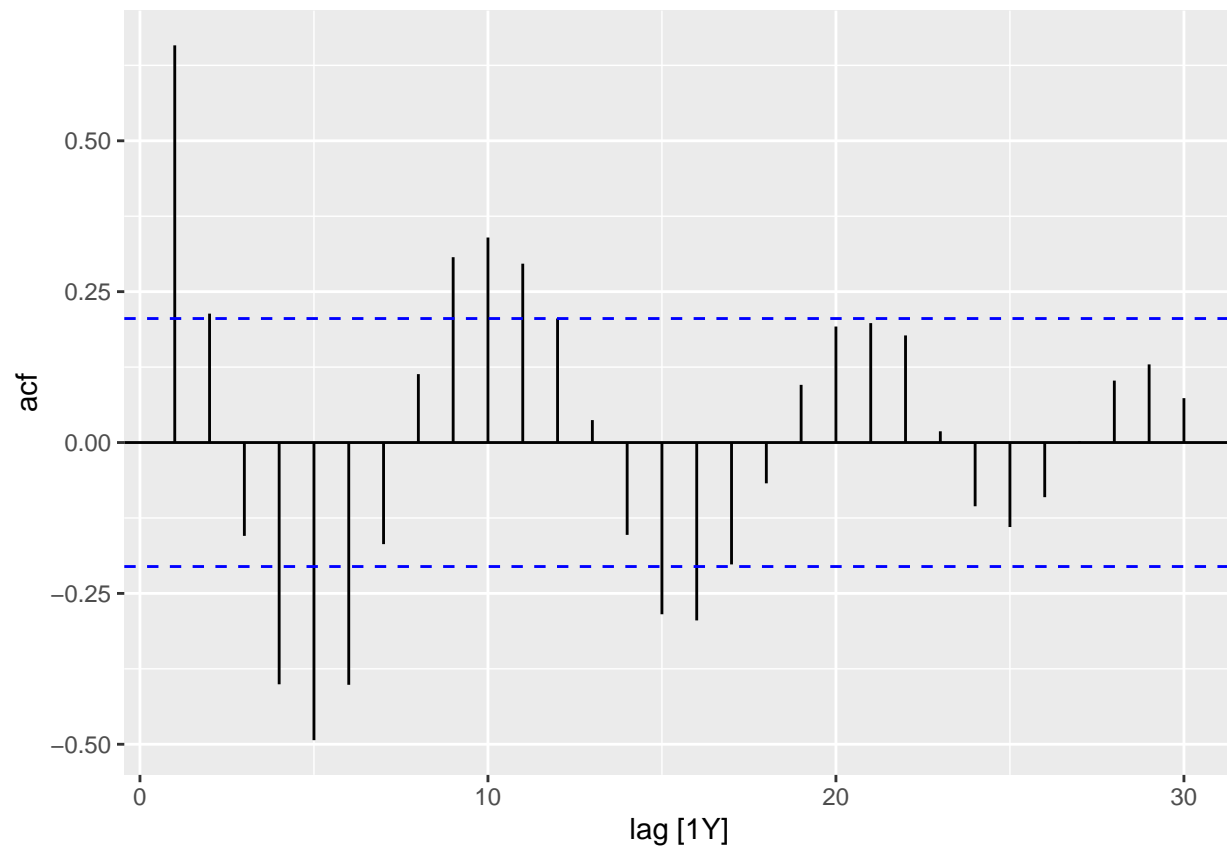
```
hare_pelts <- pelt |>
  select(Year, Hare)

hare_pelts |>
  autoplot()
```



The ACF function reflects this strong seasonality, and we can see that peaks and troughs occur about every 10 years and are offset by 5 years.

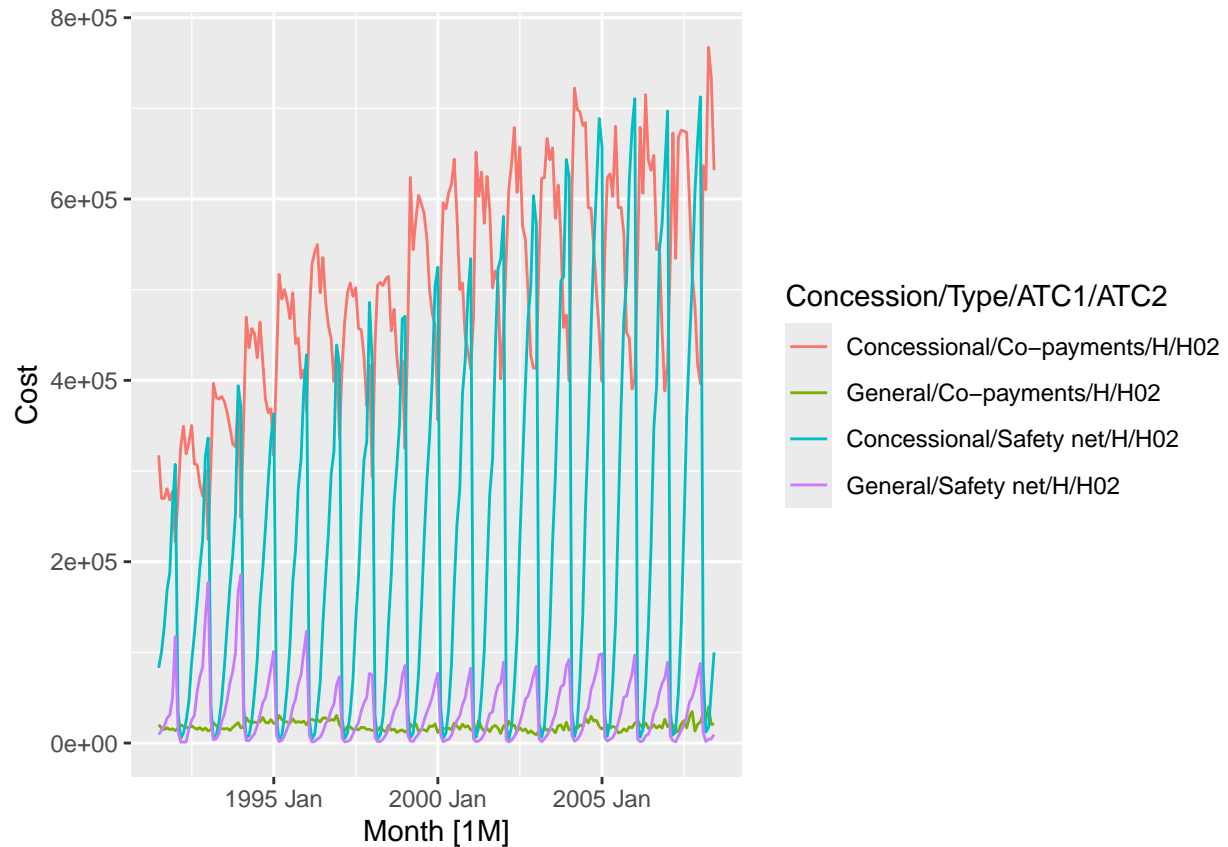
```
hare_pelts |>  
  ACF(lag_max = 30) |>  
  autoplot()
```



Looking at the “H02” costs from the PBS data, we once again see strong seasonality in three of the four graphs. The General/Safety net (G/S) and Concessional/Safety net (C/S) costs share seasonal patterns and are inversely related to the Concessional/Co-payments (C/C).

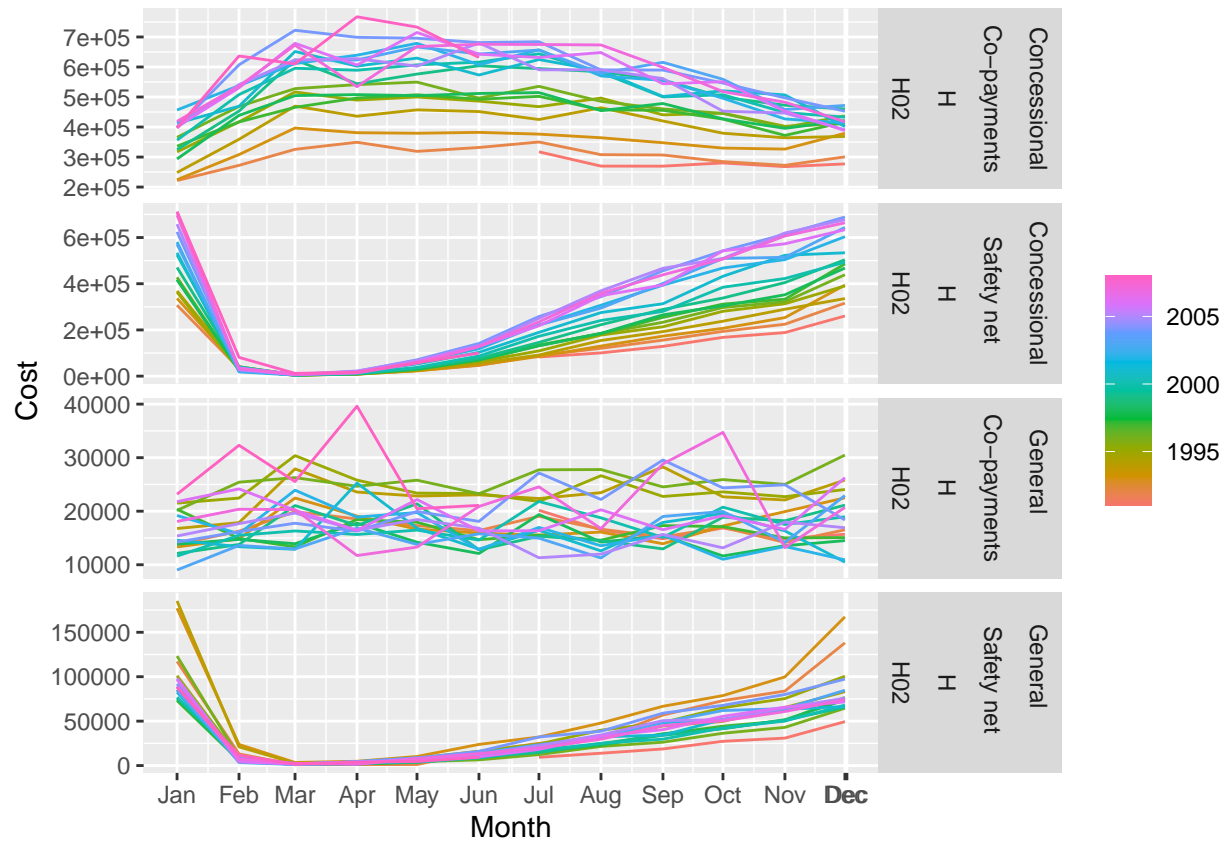
```
ho2_cost <- PBS |>
  filter(ATC2 == "H02") |>
  select(Month, Cost)

ho2_cost |>
  autoplot()
```

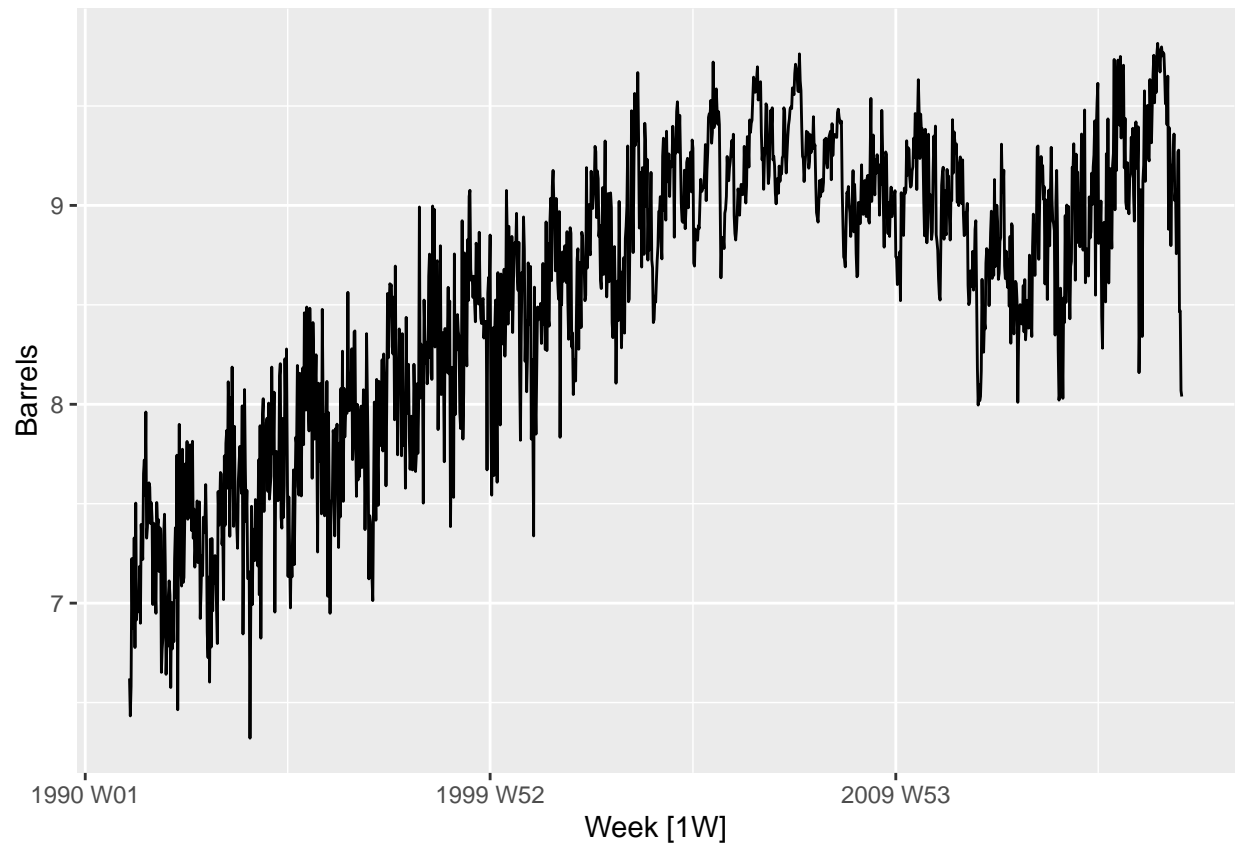
We can validate these initial observations by taking a closer look at the seasonal plots of each cost type, as well as validate the General/Co-payments (G/C) do not have seasonality hidden by the scale of the other categories. The C/S and G/S costs appear to spike significantly at the end of each year before falling to their lowest point in Spring. The C/C costs show an inverse relationship as mentioned earlier.

```
ho2_cost |>
  gg_season()
```



The us_gasoline dataset shows an overall positive trend with signs of seasonality that bear further examination.

```
us_gasoline |>
  autoplot()
```



The autocorrelation plot confirms this trend, showing some scalloping implying seasonality. There appears to be main cycle about 26 weeks in length, with a smaller lasting about 10 weeks.

```
us_gasoline |>  
  ACF() |>  
  autoplot()
```

