DATA 624 Homework 1

Kevin Havis

```r
library(tidyverse)
library(fpp3)
library(tsibble)
library(ggplot2)
library(tsibbledata)
library(knitr)
```

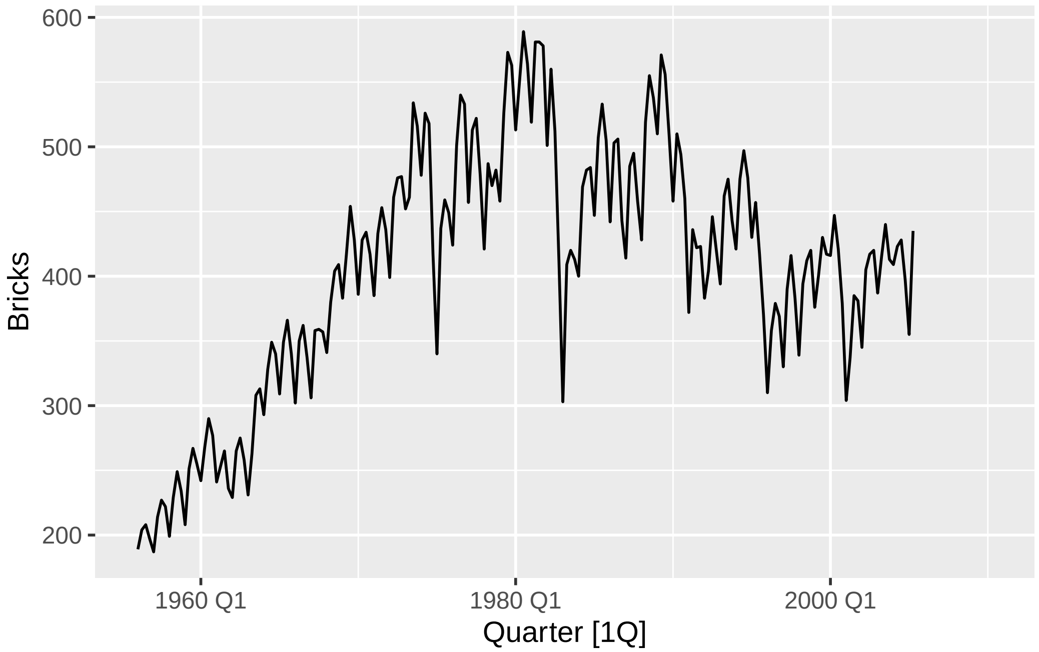
Homework 1

The document contains selected exercises from Hyndman, Athanasopoulos "Forecasting: Principles and Practice" 3rd Edition, Chapter 2

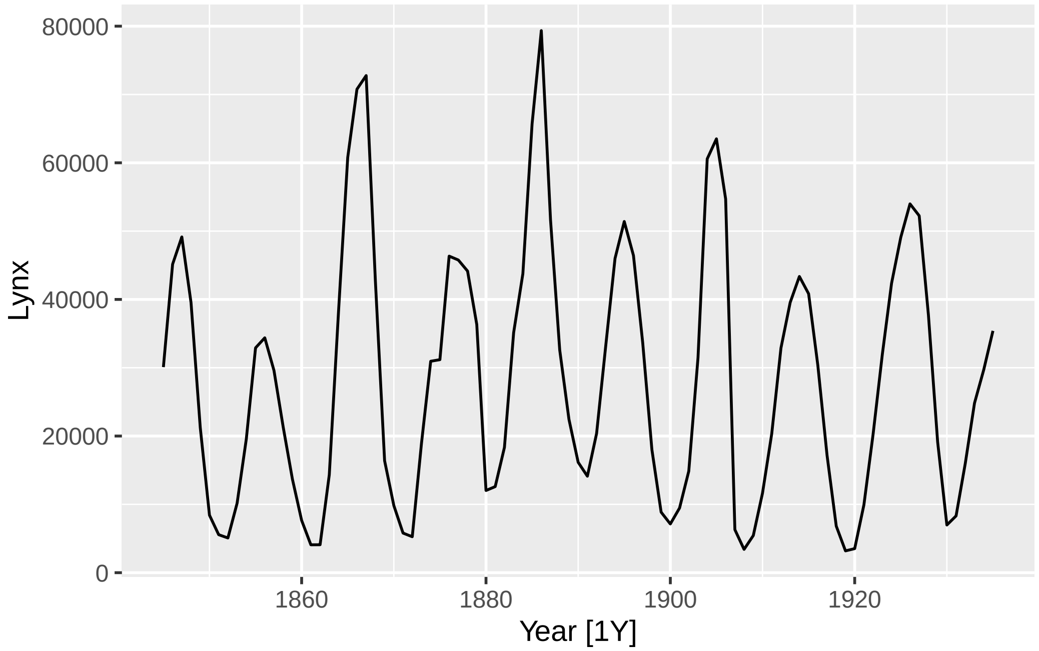
Exercise 2.1

```r
# https://otexts.com/fpp3/graphics-exercises.html

?aus_production # Half hourly
?lynx # Annual
?gafa_stock # trading days
?vic_elec # Half hourly
```

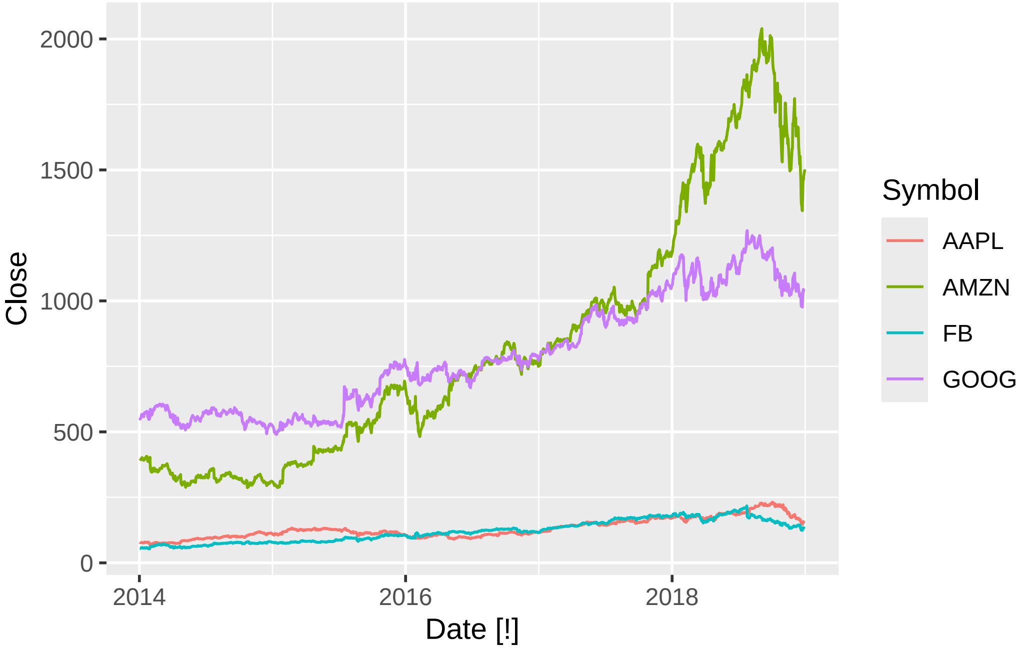
```r
# Autoplot each data by the specified feature
autoplot(aus_production, Bricks)
```



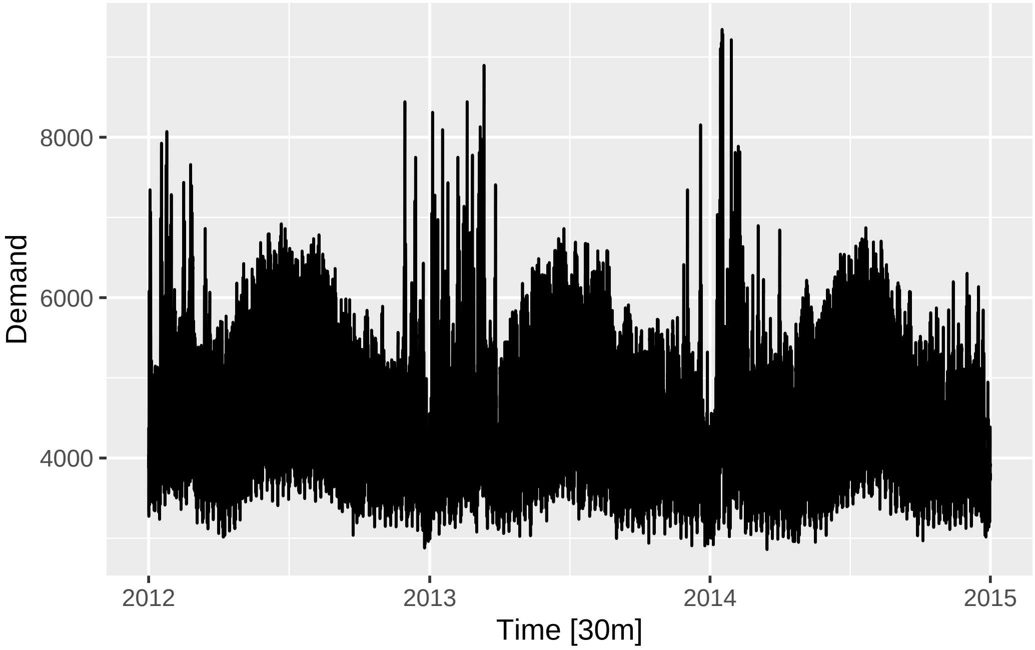
```r
autoplot(pelt, Lynx)
```



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



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



Exercise 2.2

```
# I can groupby and take the max to find top close days for each stock
top_stock_days <- gafa_stock |>
  group_by(Symbol) |>
  slice_max(Close, n=1) # Gets the max for each element I've grouped (Symbol)

kable(head(top_stock_days))
```

Symbol	Date	Open	High	Low	Close	Adj_Close	Volume
AAPL	2018-10-03	230.05	233.470	229.78	232.07	230.2755	28654800
AMZN	2018-09-04	2026.50	2050.500	2013.00	2039.51	2039.5100	5721100
FB	2018-07-25	215.72	218.620	214.27	217.50	217.5000	58954200
GOOG	2018-07-26	1251.00	1269.771	1249.02	1268.33	1268.3300	2405600

Exercise 2.3

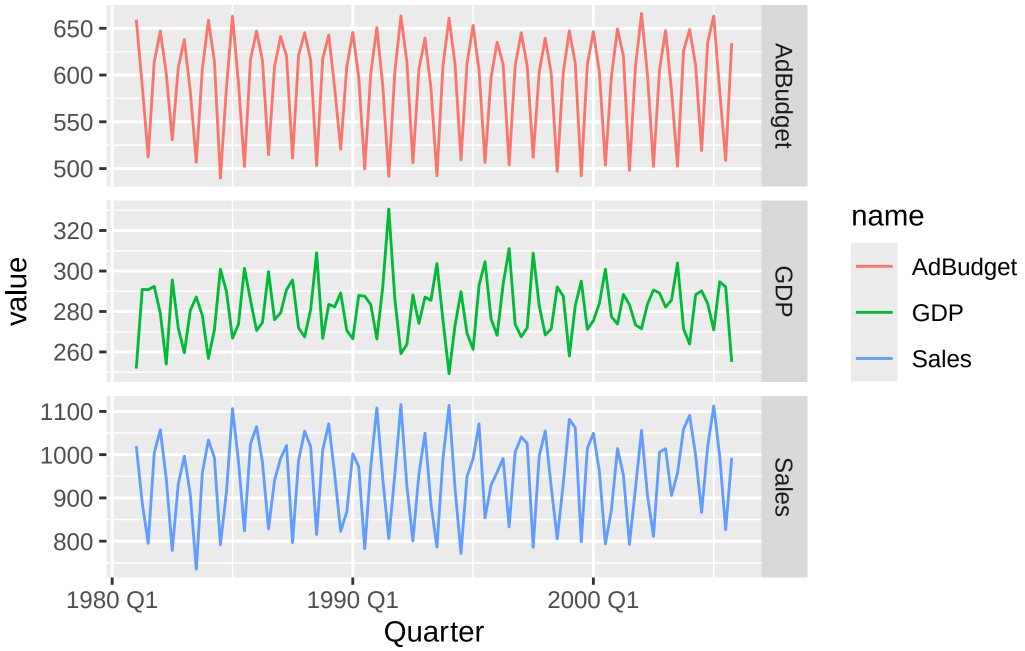
```
# Read data from URL
tute1 <- read_csv("https://otexts.com/fpp3/extrafiles/tute1.csv")
kable(head(tute1))
```

Quarter	Sales	AdBudget	GDP
1981-03-01	1020.2	659.2	251.8
1981-06-01	889.2	589.0	290.9
1981-09-01	795.0	512.5	290.8
1981-12-01	1003.9	614.1	292.4
1982-03-01	1057.7	647.2	279.1
1982-06-01	944.4	602.0	254.0

```
# Convert to tsibble
mytimeseries <- tute1 |>
  mutate(Quarter = yearquarter(Quarter)) |> # Formats date to yearquarter type
  as_tsibble(index = Quarter) # Converts to tsibble
kable(head(mytimeseries))
```

Quarter	Sales	AdBudget	GDP
1981 Q1	1020.2	659.2	251.8
1981 Q2	889.2	589.0	290.9
1981 Q3	795.0	512.5	290.8
1981 Q4	1003.9	614.1	292.4
1982 Q1	1057.7	647.2	279.1
1982 Q2	944.4	602.0	254.0

```
# Plot the timeseries
mytimeseries |>
  pivot_longer(-Quarter) |> # Viz friendly format
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line() +
  facet_grid(name ~ ., scales = "free_y") # Sets a facet for each column
```



Exercise 2.4

```
# Get the US Gas dataset
install.packages("USgas")
```

```
The downloaded binary packages are in
    /var/folders/7_/3jtyrt0n1w9_jssp00807w2h0000gn/T//RtmphrWZcJ/downloaded_packages
```

```r
library(USgas)
```

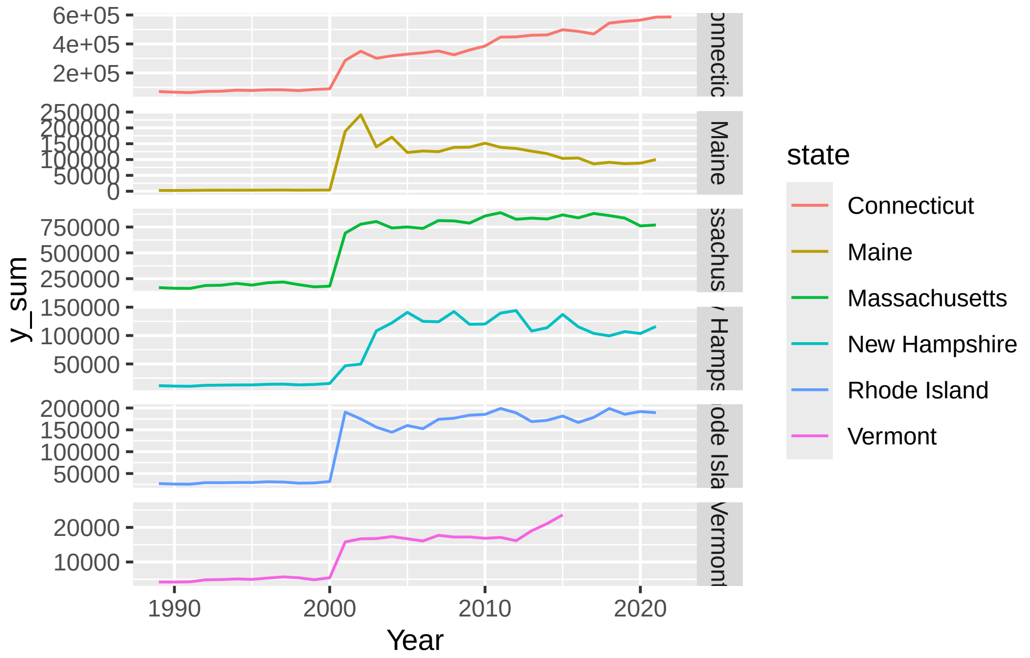
```r
head(usgas)
```

```
        date                 process state state_abb      y
1 1973-01-01  Commercial Consumption  U.S.      U.S. 392315
2 1973-01-01 Residential Consumption  U.S.      U.S. 843900
3 1973-02-01  Commercial Consumption  U.S.      U.S. 394281
4 1973-02-01 Residential Consumption  U.S.      U.S. 747331
5 1973-03-01  Commercial Consumption  U.S.      U.S. 310799
6 1973-03-01 Residential Consumption  U.S.      U.S. 648504
```

```r
# Plot the New England states
ne_gas <- usgas |>
  mutate(Year = year(date)) |>
  group_by(Year, state) |>
  summarize(y_sum = sum(y)) |>
  filter(state %in% c("Maine", "Vermont", "New Hampshire", "Massachusetts", "Connecticut","Rh
  as_tsibble(index=Year, key=state)

kable(head(ne_gas))
```

Year	state	y_sum
1989	Connecticut	71468
1990	Connecticut	66856
1991	Connecticut	64020
1992	Connecticut	72234
1993	Connecticut	73641
1994	Connecticut	80683

```r
# Plot the data
ne_gas |>
  ggplot(aes(x = Year, y = y_sum, colour = state)) +
  geom_line() +
  facet_grid(state ~ ., scales = "free_y")
```



Exercise 2.5

```
# Load the tourism data from the package
# Our reference point
kable(head(tsibble::tourism))
```

Quarter	Region	State	Purpose	Trips
1998 Q1	Adelaide	South Australia	Business	135.0777
1998 Q2	Adelaide	South Australia	Business	109.9873
1998 Q3	Adelaide	South Australia	Business	166.0347
1998 Q4	Adelaide	South Australia	Business	127.1605
1999 Q1	Adelaide	South Australia	Business	137.4485
1999 Q2	Adelaide	South Australia	Business	199.9126

```
# Download the example tourism file
library(readxl)
my_tourism <- read_excel("~/projects/data_624/assignments/tourism.xlsx")
kable(head(my_tourism))
```

Quarter	Region	State	Purpose	Trips
1998-01-01	Adelaide	South Australia	Business	135.0777
1998-04-01	Adelaide	South Australia	Business	109.9873
1998-07-01	Adelaide	South Australia	Business	166.0347
1998-10-01	Adelaide	South Australia	Business	127.1605
1999-01-01	Adelaide	South Australia	Business	137.4485
1999-04-01	Adelaide	South Australia	Business	199.9126

```
# Match downloaded tourism data to package data
# Just need to convert the char Quarter to yearquarter
my_tourism <- my_tourism |>
  mutate(Quarter = yearquarter(Quarter))

kable(head(my_tourism))
```

Quarter	Region	State	Purpose	Trips
1998 Q1	Adelaide	South Australia	Business	135.0777
1998 Q2	Adelaide	South Australia	Business	109.9873
1998 Q3	Adelaide	South Australia	Business	166.0347
1998 Q4	Adelaide	South Australia	Business	127.1605
1999 Q1	Adelaide	South Australia	Business	137.4485
1999 Q2	Adelaide	South Australia	Business	199.9126

```
# Find the most amount of flights by Purpose and Region
my_tourism |>
  group_by(Region, Purpose) |>
  summarize(total_trips = sum(Trips)) |>
  arrange(desc(total_trips))
```

```
# A tibble: 304 x 3
# Groups:   Region [76]
   Region          Purpose  total_trips
   <chr>           <chr>          <dbl>
 1 Sydney          Visiting      59782.
 2 Melbourne       Visiting      49512.
 3 Sydney          Business      48164.
 4 North Coast NSW Holiday       47032.
 5 Sydney          Holiday       44026.
 6 Gold Coast      Holiday       42267.
 7 Melbourne       Holiday       40583.
 8 South Coast     Holiday       39605.
 9 Brisbane        Visiting      39424.
10 Melbourne       Business      38238.
# i 294 more rows
```

```
# It's Sydney!
```

```
# Group by just state and trips
my_tourism |>
  group_by(State) |>
  summarize(total_trips = sum(Trips)) |>
  arrange(desc(total_trips))
```

```
# A tibble: 8 x 2
  State              total_trips
  <chr>                    <dbl>
1 New South Wales        557367.
2 Victoria               390463.
3 Queensland             386643.
4 Western Australia      147820.
5 South Australia        118151.
6 Tasmania                54137.
7 ACT                     41007.
8 Northern Territory      28614.
```

Exercise 2.6

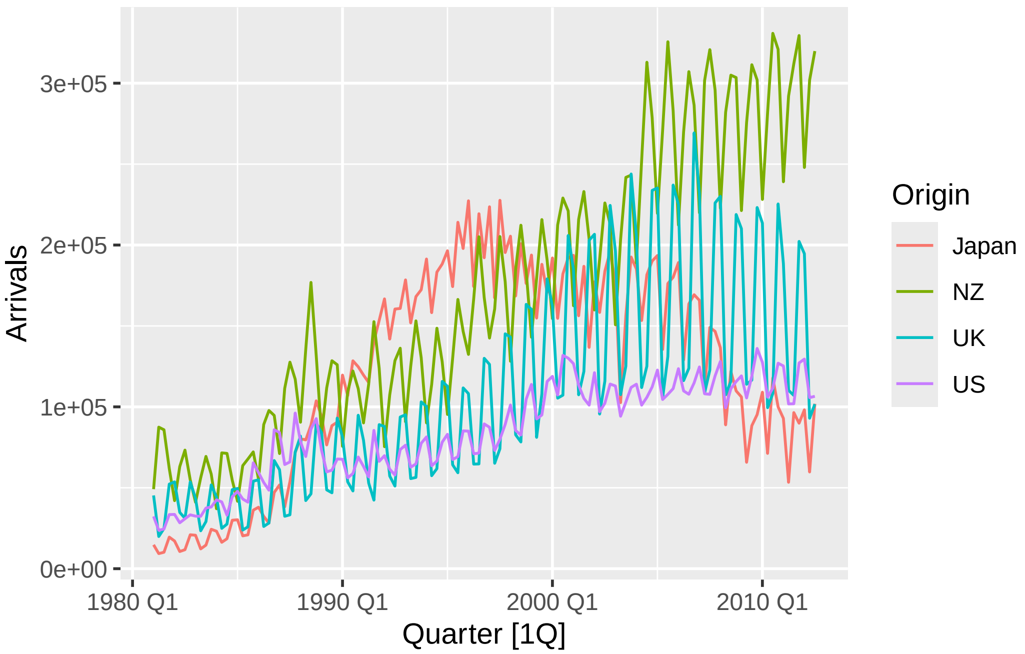
Observing the various plots below of our `aus_arrivals` data, there are a few interesting observations we can make.

1. *Seasonality*: We can see decrease in travel to Australia from the US and UK during the middle months, the inverse for New Zealand
2. *Trend*: Flights to Australia have generally increased overtime, with the exception of those from Japan which sharply falls after early 1990s

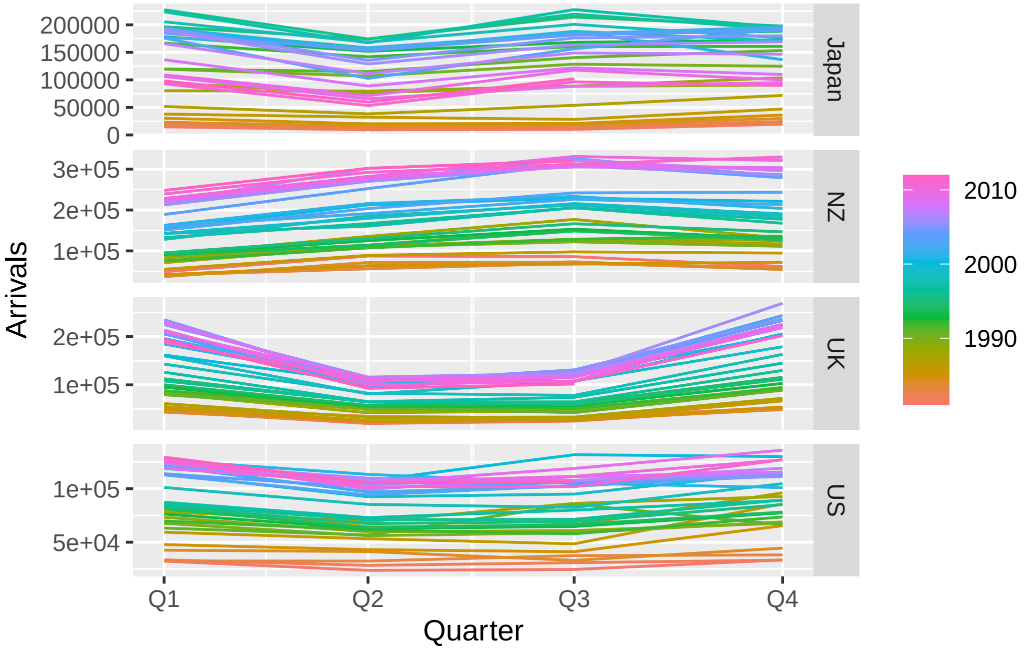
For seasonality plots, the `gg_season` plot is well suited for general observations, but I prefer the detail and addition of the mean we get "for free" from the `gg_subseries` plot.

```
# Get the aus_arrival data from the package
arrival_data <- fpp3::aus_arrivals
```

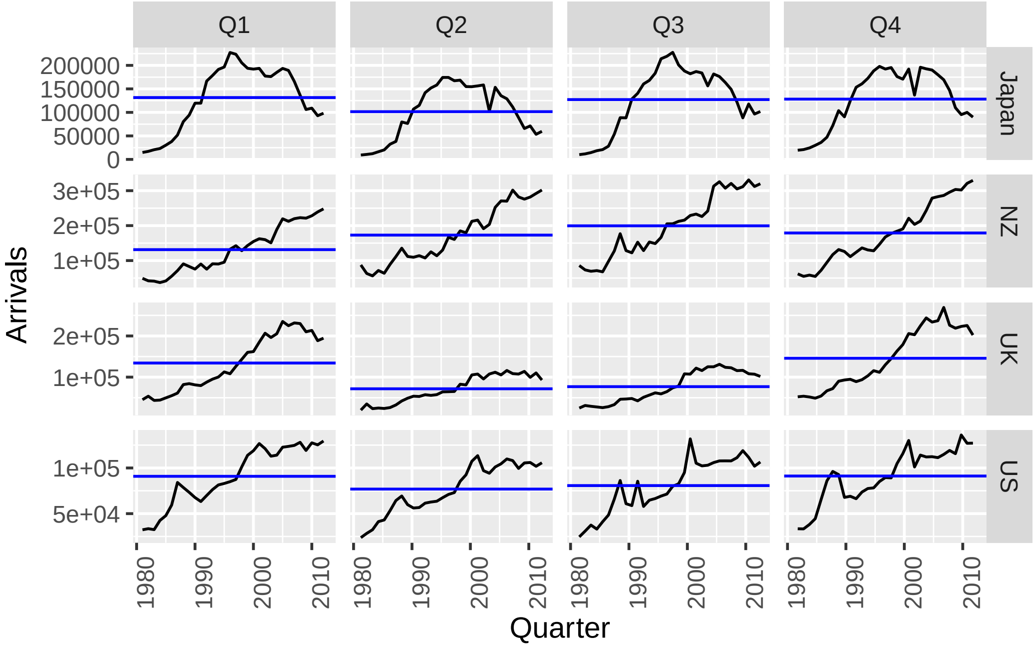
```
# Try autoplot
autoplot(arrival_data)
```



```
# Try seasons
library(feasts)
gg_season(arrival_data)
```



```r
# Try subseries
gg_subseries(arrival_data)
```



Exercise 2.7

Considering the Australian retail dataset, the visualizations help us easily identify patterns of trend, seasonality, and cycles.

We can see that the autocorrelation rate is quite high (and positive) for Turnover in all the the graphs. This tells us that the Turnover of the prior interval is a good indicator of future intervals, which we would describe as having high autocorrelation.

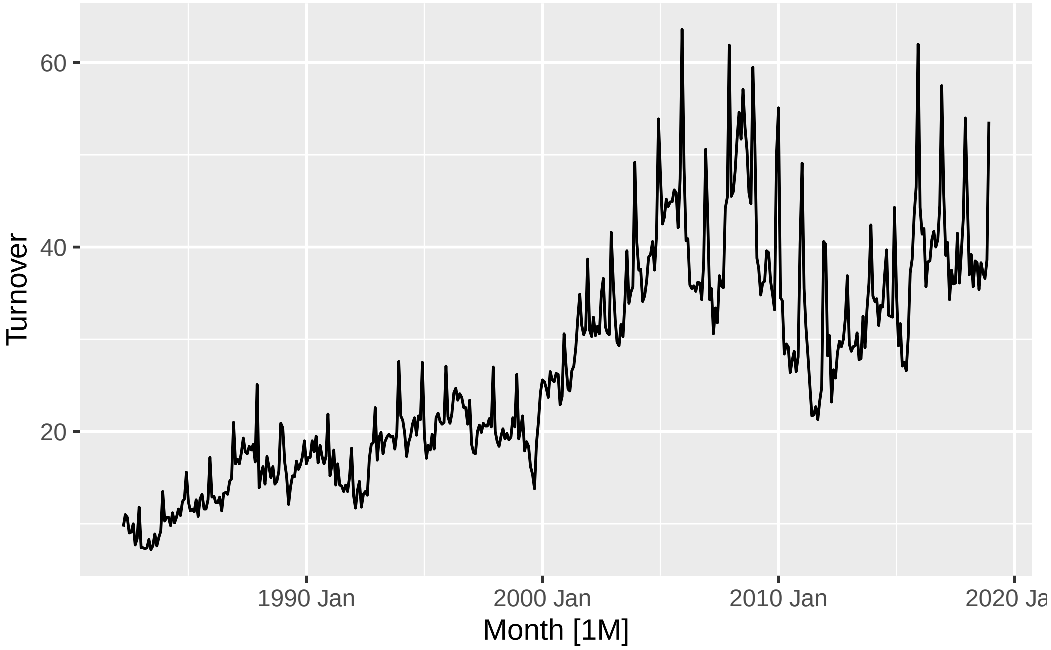
In addition, we can see the seasonality, especially in the `gg_subseries` plot, where the mean is indicated by the blue line. We can see Turnover remains relatively flat through the second and third quarter, with higher Turnover rates near the beginning and end of years.

The trend, or the long-term direction of the series over time, is positive, as we can see in the rising Turnover rates, especially from 2000 and onward.

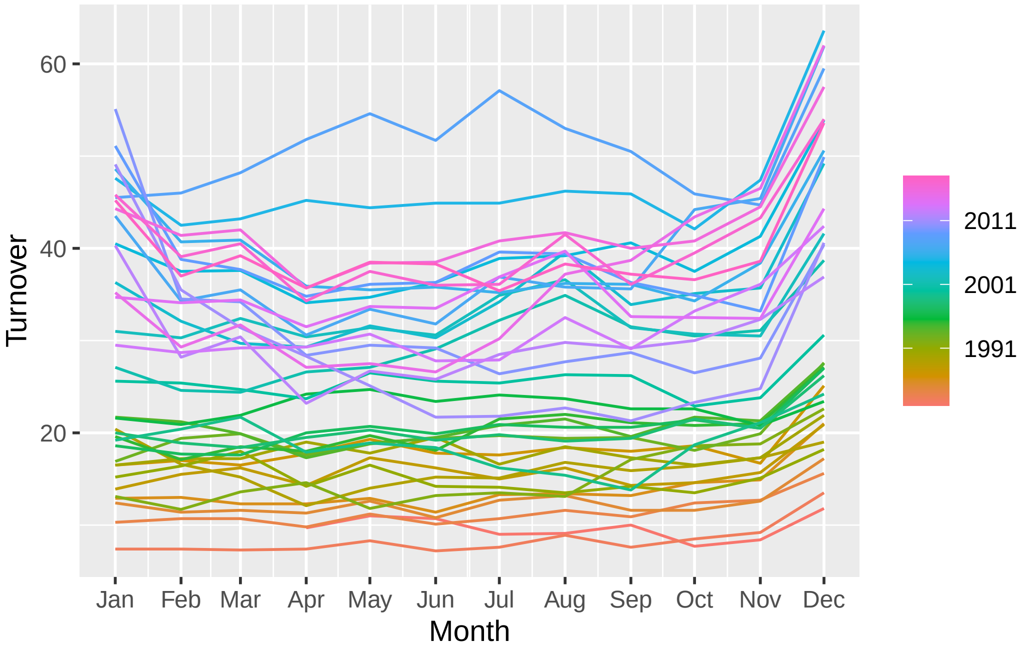
As far as cycles, there may be a subtle 5-7 cycle of low to high turnover that is becoming more exaggerated in recent years, showing between '86-'88, '93-'00, and '06-'12, but would need to be confirmed analytically.

```r
# Pick a series to plot
set.seed(42)

myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

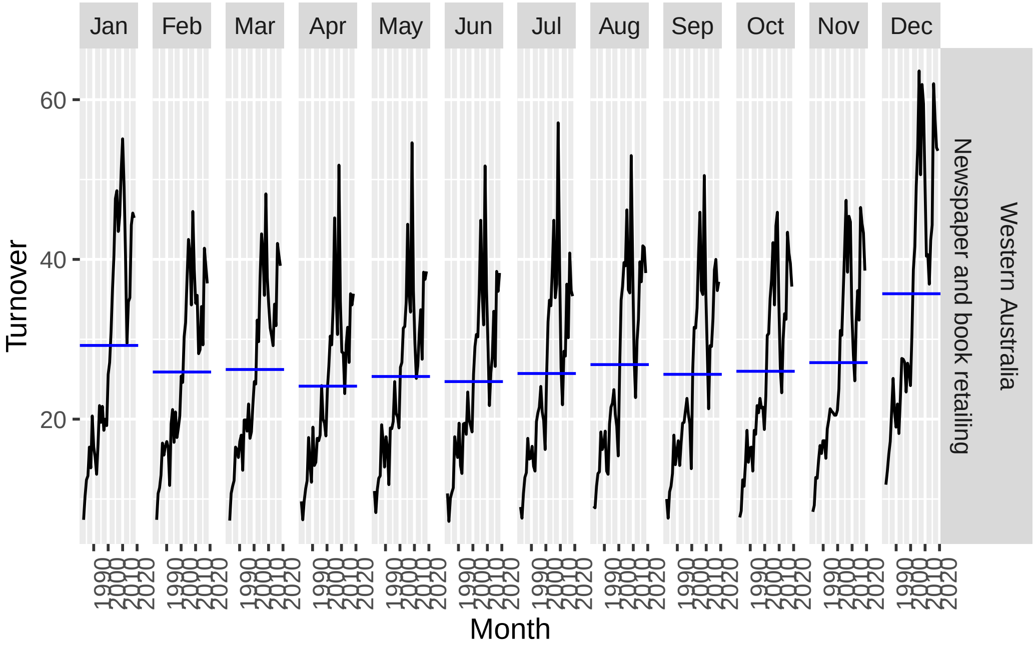
```r
autoplot(myseries)
```



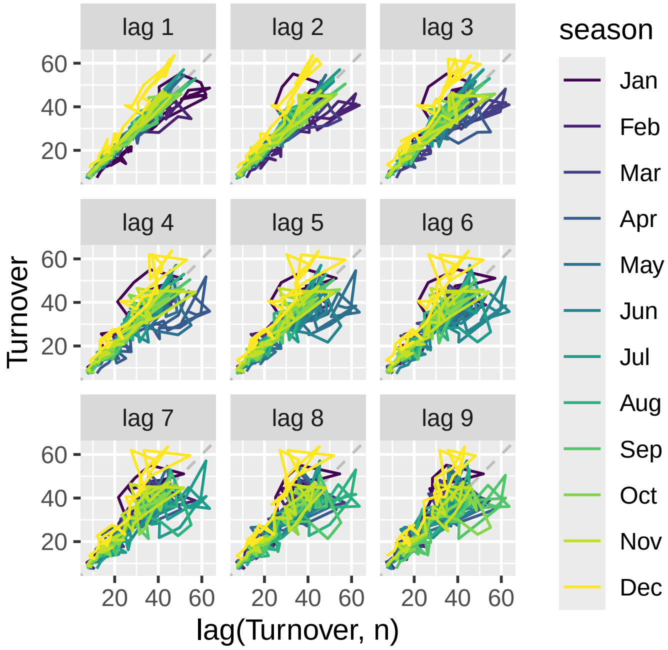
```
gg_season(myseries)
```



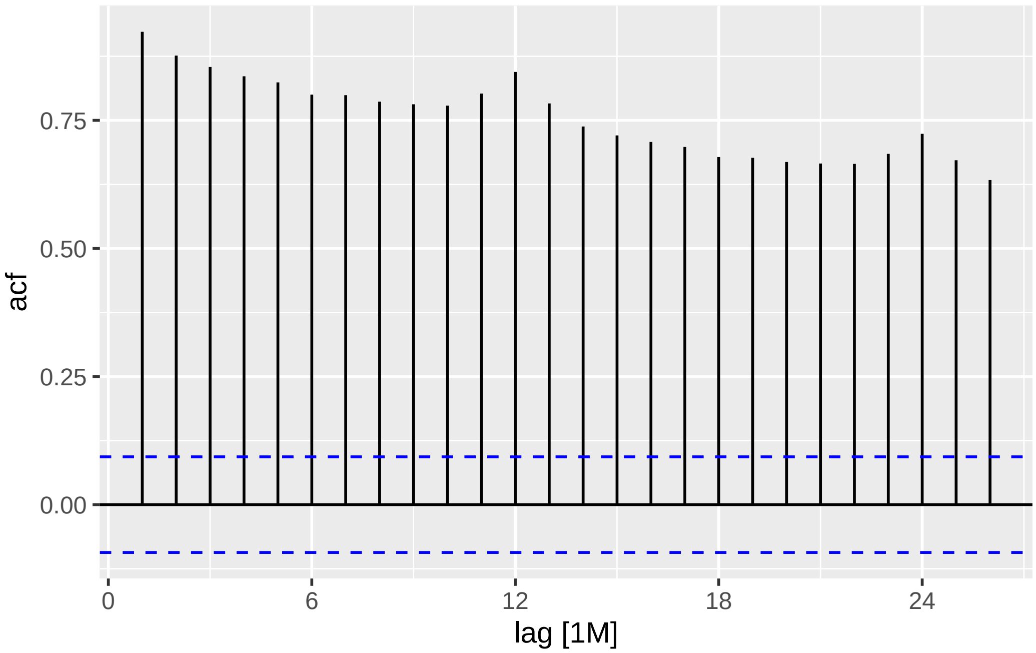
```
gg_subseries(myseries)
```



```
gg_lag(myseries)
```



```
myseries |> ACF(Turnover) |> autoplot()
```



Exercise 2.8

```
# Generate plots of all the data
# We create a function to do this repeatedly
plot_series <- function(data, value_col, period = NULL) {
  print(data |> autoplot(.vars = {{ value_col }}) + labs(title = "Time Series Plot"))

  tryCatch({
    if (is.null(period)) {
      print(data |> gg_season({{ value_col }}) + labs(title = "Seasonal Plot"))
    } else {
      print(data |> gg_season({{ value_col }}, period = period) + labs(title = "Seasonal Plo
    }
  }, error = function(e) {
    cat("Skipping seasonal plot:", e$message, "\n")
  })

  tryCatch({
    print(data |> gg_subseries({{ value_col }}) + labs(title = "Seasonal Subseries Plot"))
  }, error = function(e) {
    cat("Skipping subseries plot:", e$message, "\n")
  })

  tryCatch({
  print(data |> gg_lag({{ value_col }}) + labs(title = "Lag Plot"))
  }, error = function(e) {
    cat("Skipipng gglag plot:", e$message, "\n")
  })

  print(data |> ACF({{ value_col }}) |> autoplot() + labs(title = "Autocorrelation Function"
}
```

United States Employement Series

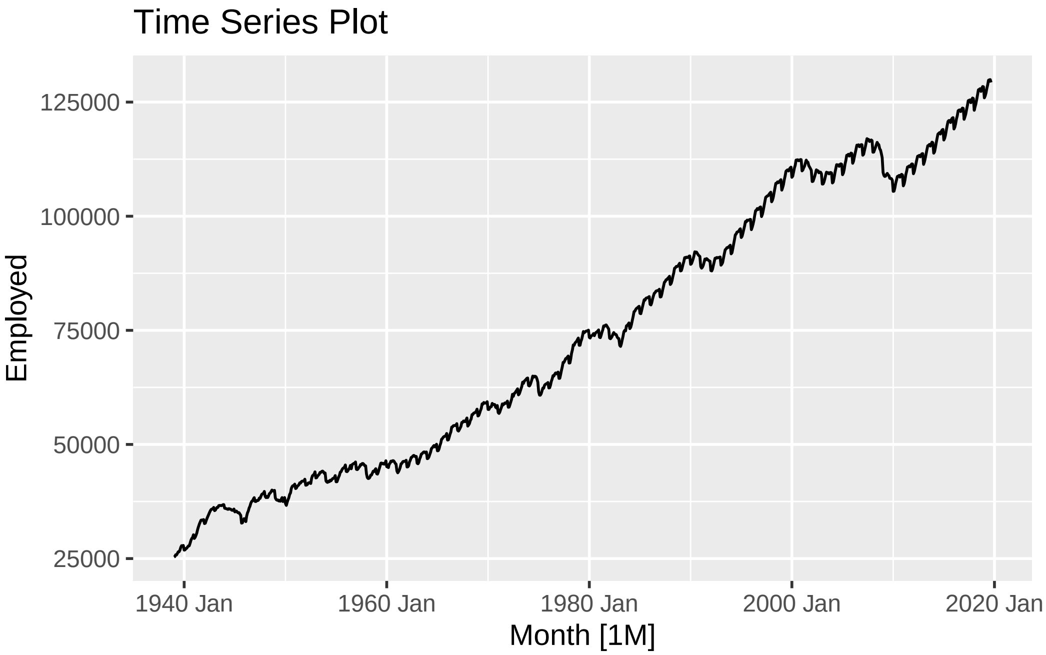
US private sector employment has been trending up through all of dates available in the data. We can see slightly upticks in employment during periods of peak seasonal work through the summer months. The data is very highly autocorrelated, averaging nearly 90% across all observed years.

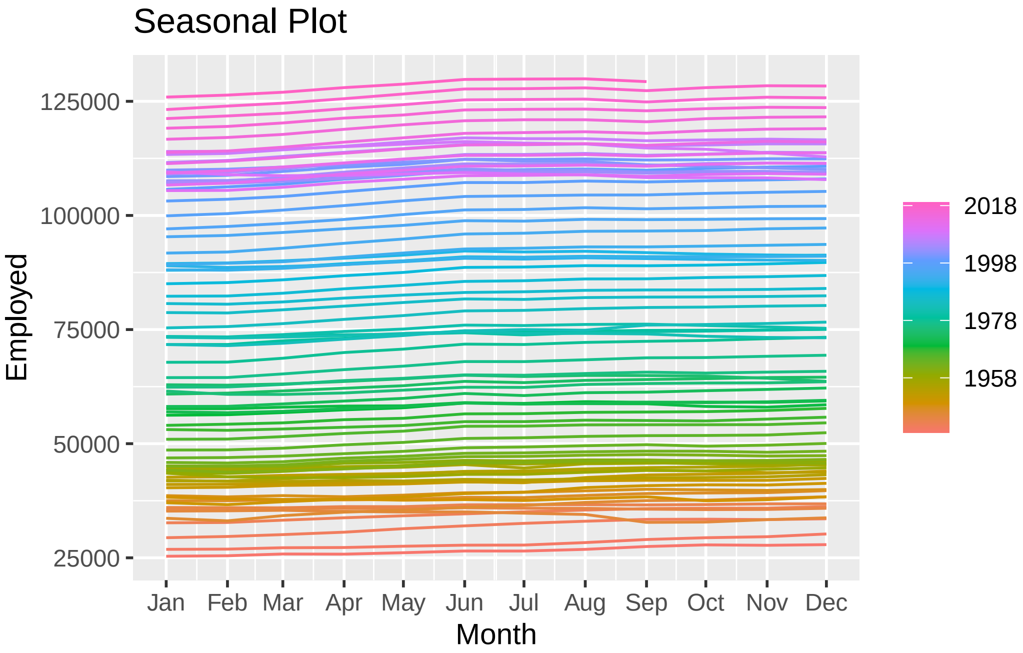
We can see brief periods of downwards employment trends periodically through the larger timeseries (besides the seasonality); we would describe this as cycles of unemployment that likely follow recessions typical of most healthy economies.

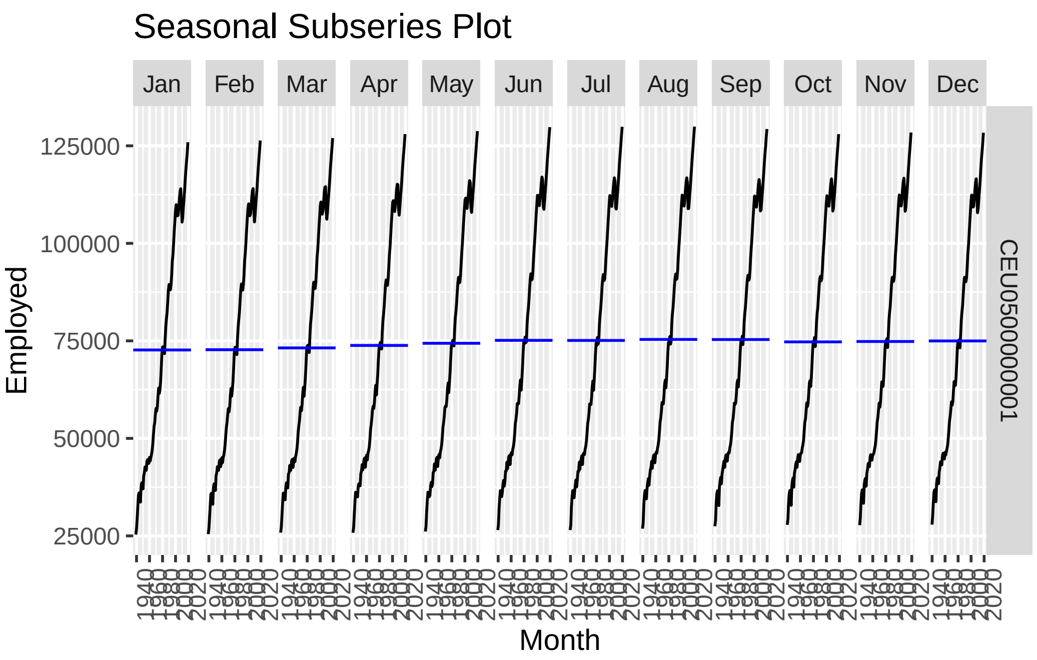
One such downturn aligns with the recession and housing market crash of 2008, which is clear in the data.

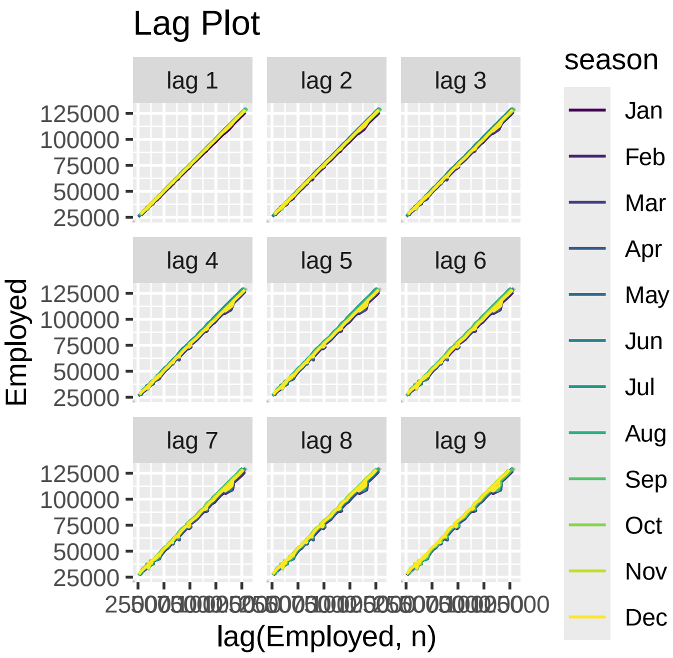
Overall, the graphs show a positive and relatively stable growth of employment in the United States private sector.

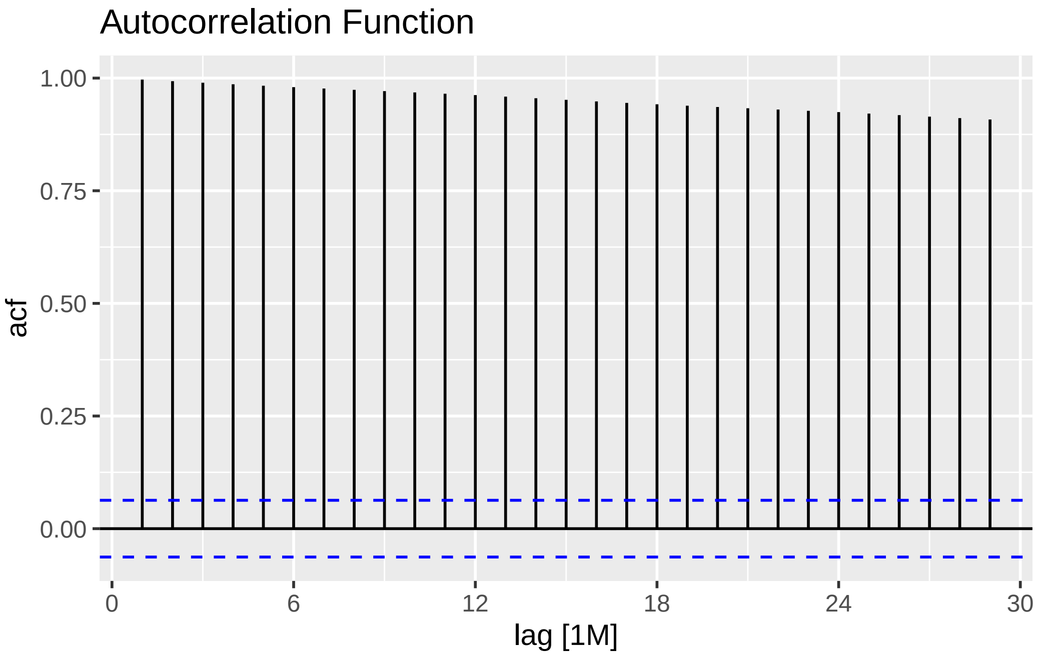
```
# US Employement

set.seed(42)
us_employment <- fpp3::us_employment |>
  filter(Title == "Total Private") |>
  drop_na(Employed)

# Then plot with ACF column specified
plot_series(us_employment, Employed)
```


Time Series Plot
Employed
125000
100000
75000
50000
25000
1940 Jan
1960 Jan
1980 Jan
2000 Jan
2020 Jan
Month [1M]


Seasonal Plot
Employed
125000
100000
75000
50000
25000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
2018
1998
1978
1958


Seasonal Subseries Plot
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
125000
100000
75000
50000
25000
Employed
CEU0500000001
1940
1960
1980
2000
2020
Month


Lag Plot
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
125000
100000
75000
50000
25000
Employed
25000
50000
75000
100000
125000
lag(Employed, n)
season
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec



Australian Bricks Production Series

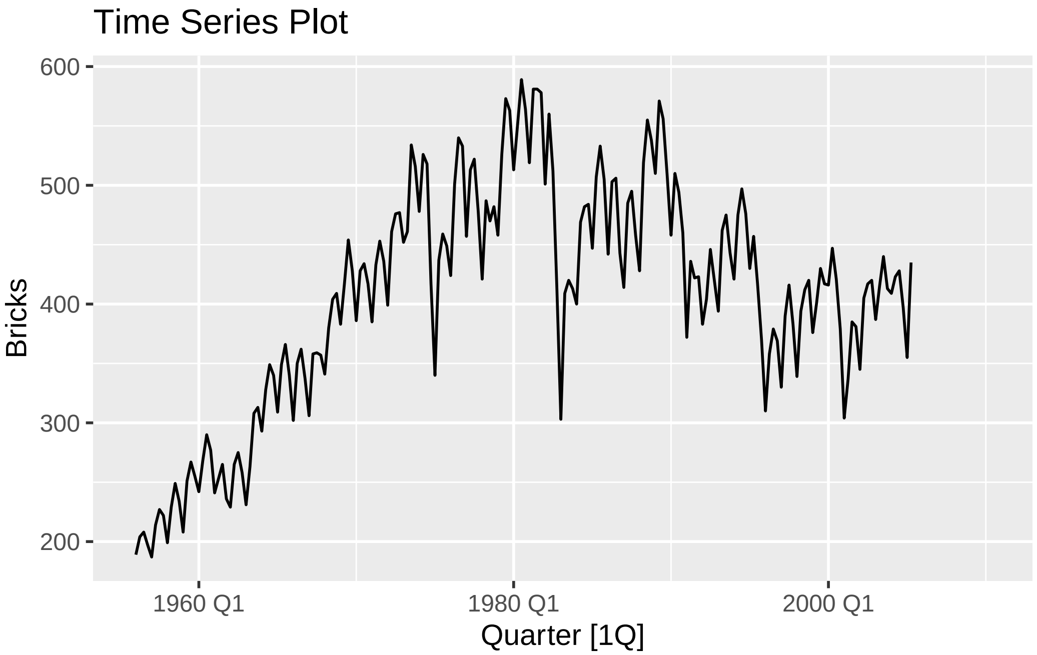
Production of bricks in the Australian retail market is varied by both cycles and seasons. We can see from the timeseries line graph that there's no unidirectional trend; the market appears to have peaked in the early 1980s before sharply crashing, initiating a downward trend that continues to the edges of our data.

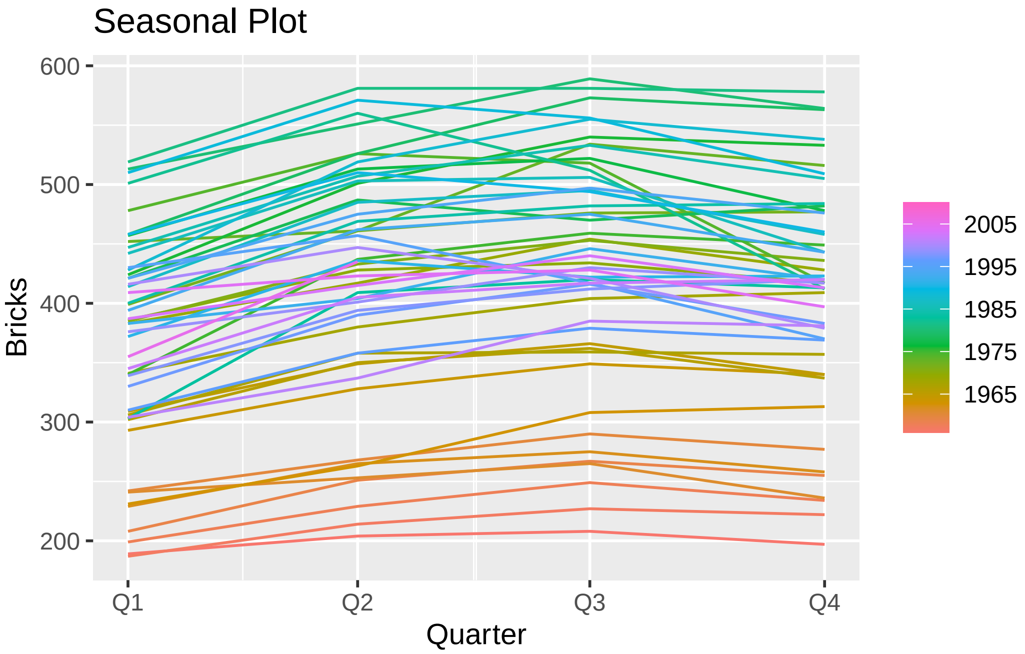
Australia's spring season occurs in Q3, which is where we see an uptick in brick production; this is intuitive as we would expect peak construction season to be in the summer months, and bricks would be a natural direct material.

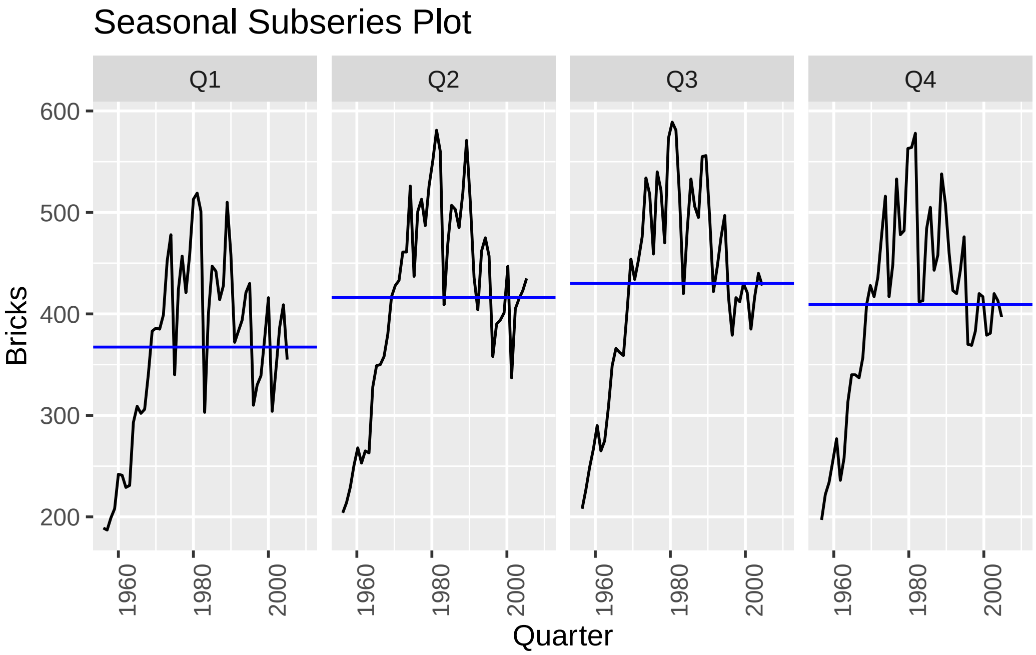
Brick production is still quite positively autocorrelated, but more noise is introduced when increasing the lag parameter.

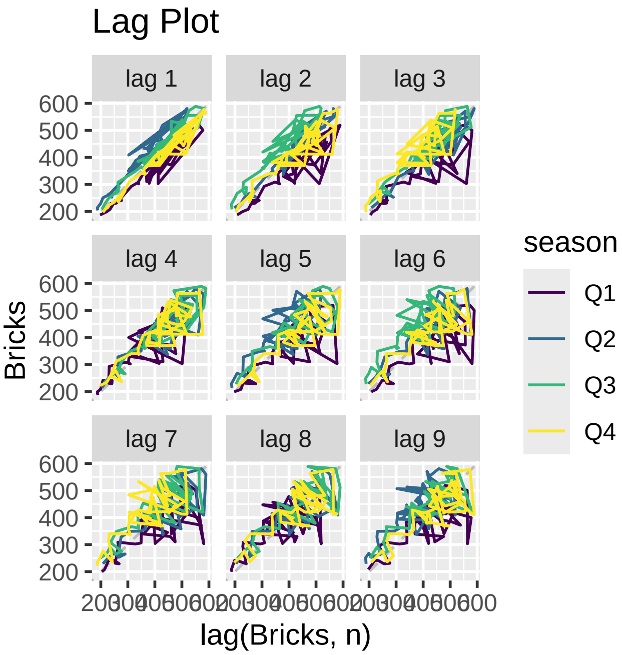
Overall, the graphs indicate to me that the brick production market is relatively saturated at this point, especially after a boom in the 1980s, and is currently trending towards a lower mean level of production.

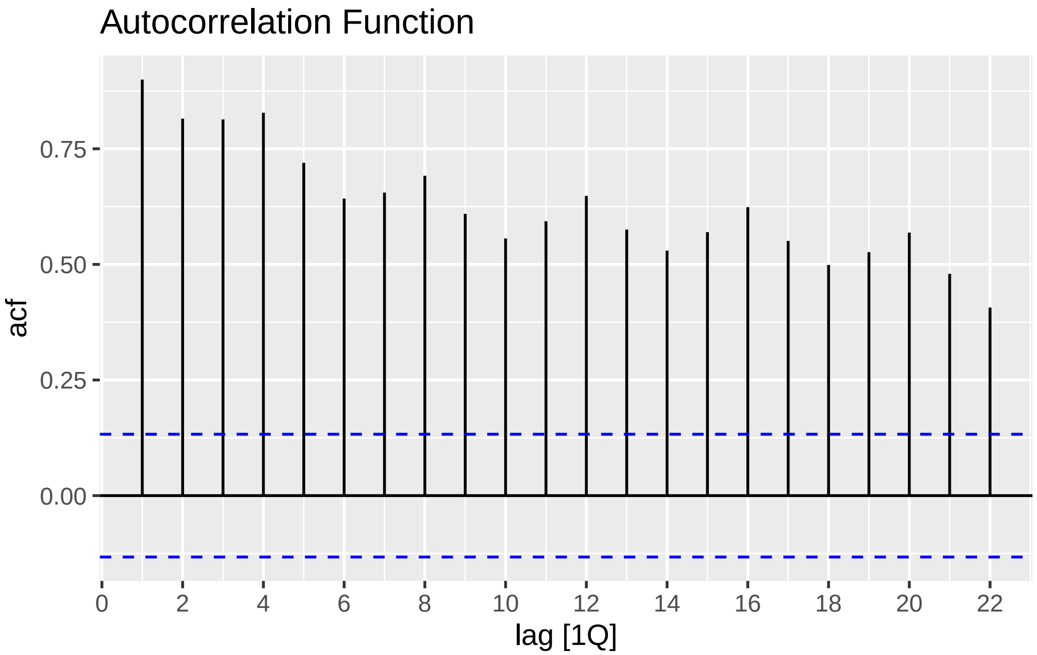
```
plot_series(aus_production, Bricks)
```


Time Series Plot
Bricks
600
500
400
300
200
1960 Q1
1980 Q1
2000 Q1
Quarter [1Q]


Seasonal Plot
Bricks
600
500
400
300
200
Q1
Q2
Q3
Q4
Quarter
2005
1995
1985
1975
1965


Seasonal Subseries Plot
Q1
Q2
Q3
Q4
Bricks
600
500
400
300
200
1960
1980
2000
Quarter


Lag Plot
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
Bricks
600
500
400
300
200
200
300
400
500
600
lag(Bricks, n)
season
Q1
Q2
Q3
Q4



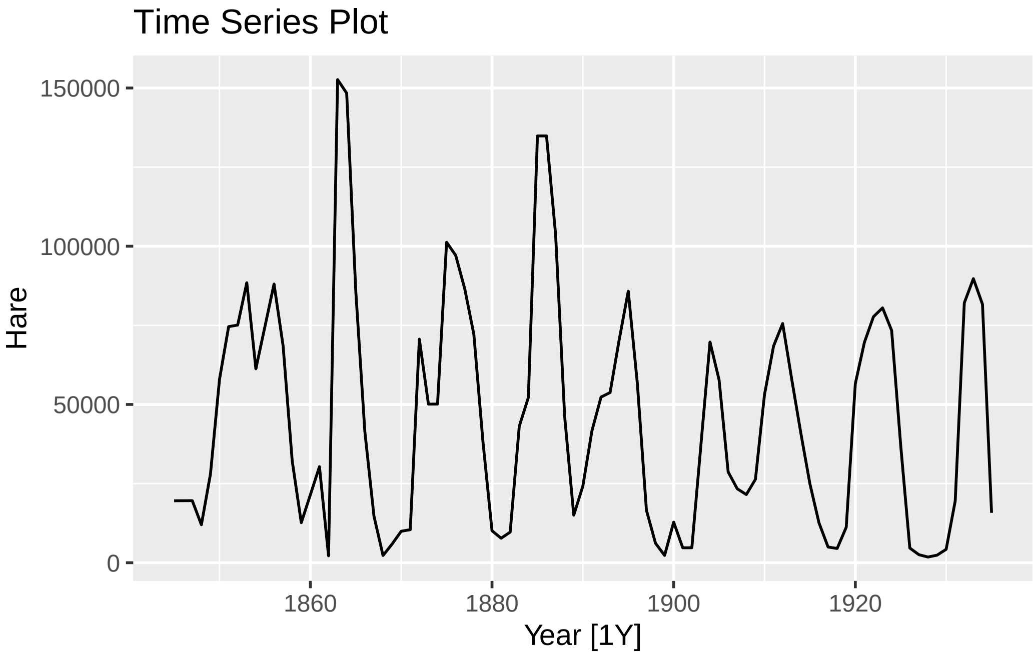
Hare Pelt Harvesting Series

The pelt trade was naturally seasonal, due to the nature of the animals from which they were harvested from and their lifecycle. This is supported emperically by the data.

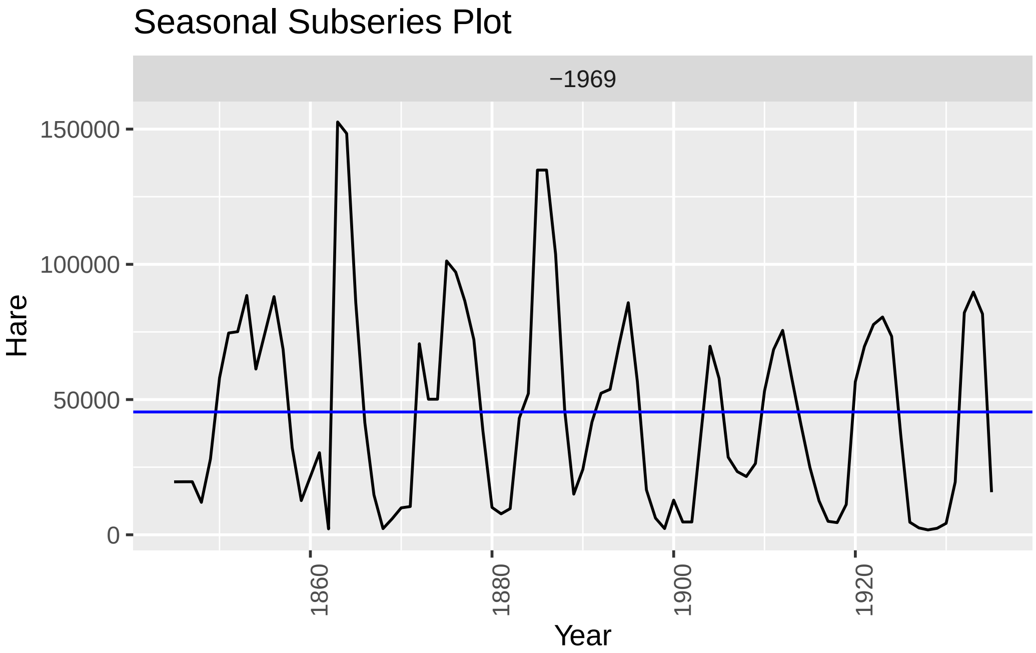
From the data available we see a very flat trend, neither upwards or downwards, punctuated with a high degree of seasonality. Of especial interest is the autocorrolation function, which highlights the autocorrelation of the series dropping off as the peak hunting seasons end.

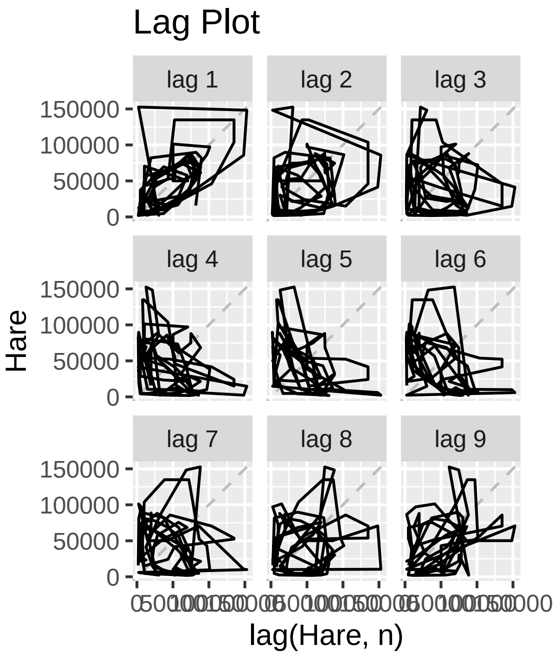
We can see one peak in particular around 1862, tripling the mean of pelts across the timeseries. This may have been a year of particularly beneficial weather or high hare population, or perhaps simply a temporal high demand.

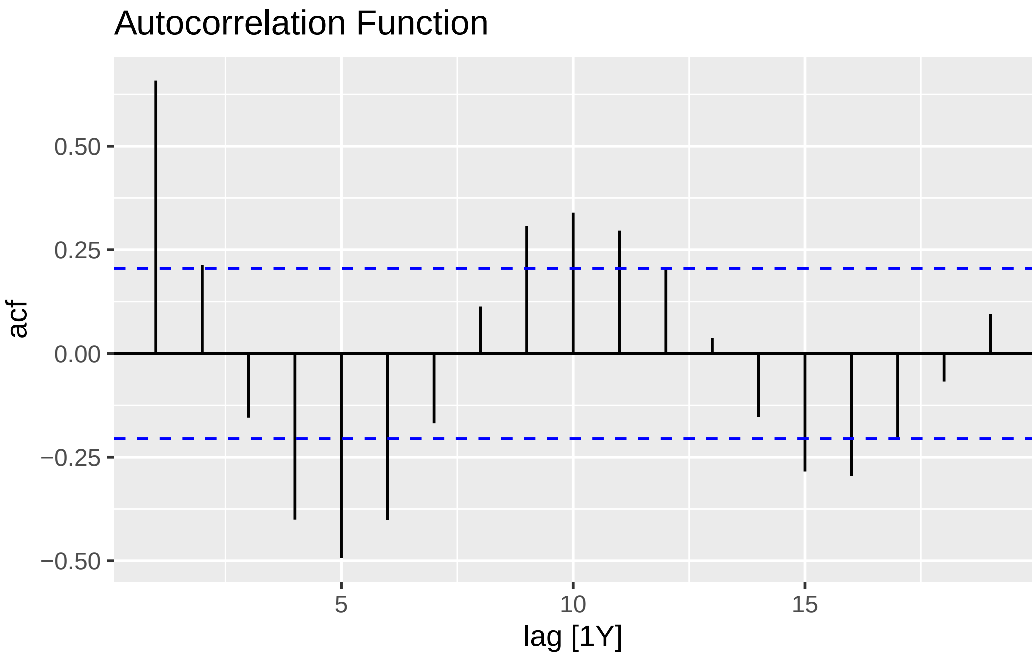
```
plot_series(tsibbledata::pelt, Hare, period = '1y')
```



```
Skipping seasonal plot: The data must contain at least one observation per seasonal period.
```




Lag Plot
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
150000
100000
50000
0
Hare
lag(Hare, n)
0
50000
100000
150000


Autocorrelation Function
0.50
0.25
0.00
−0.25
−0.50
acf
5
10
15
lag [1Y]

PBS Cost of Co-payments Series

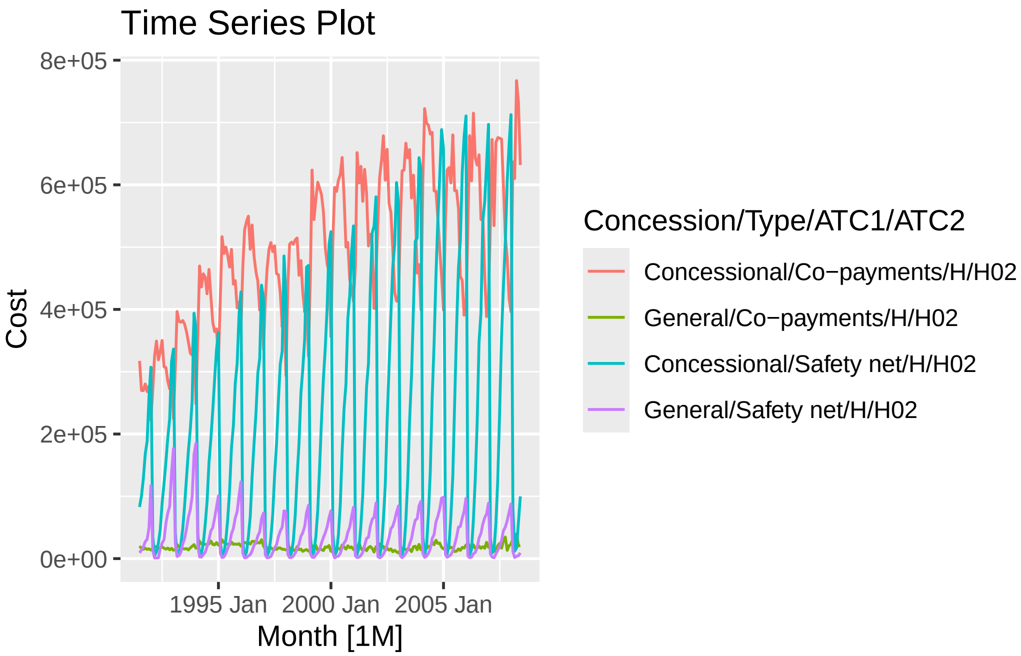
The PBS data is a bit complex. It covers prescriptions costs across multiple categories which can broadly be broken into Co-payments, where an individual pays for a script out of pocket, and Saftey Net, the additional cost beyond a specific threshold which is subsidized.

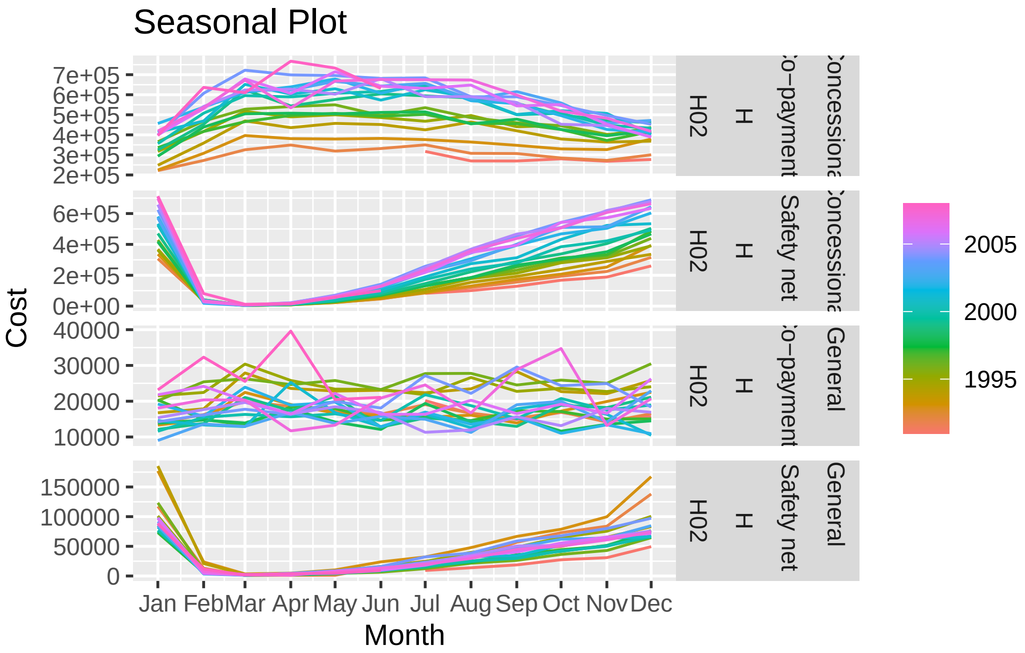
The main story in this data is the increasing costs of subsidized Safety nets. The trend is growing positively and inversely to co-payments, indicating that more and more health costs are being subsidized.

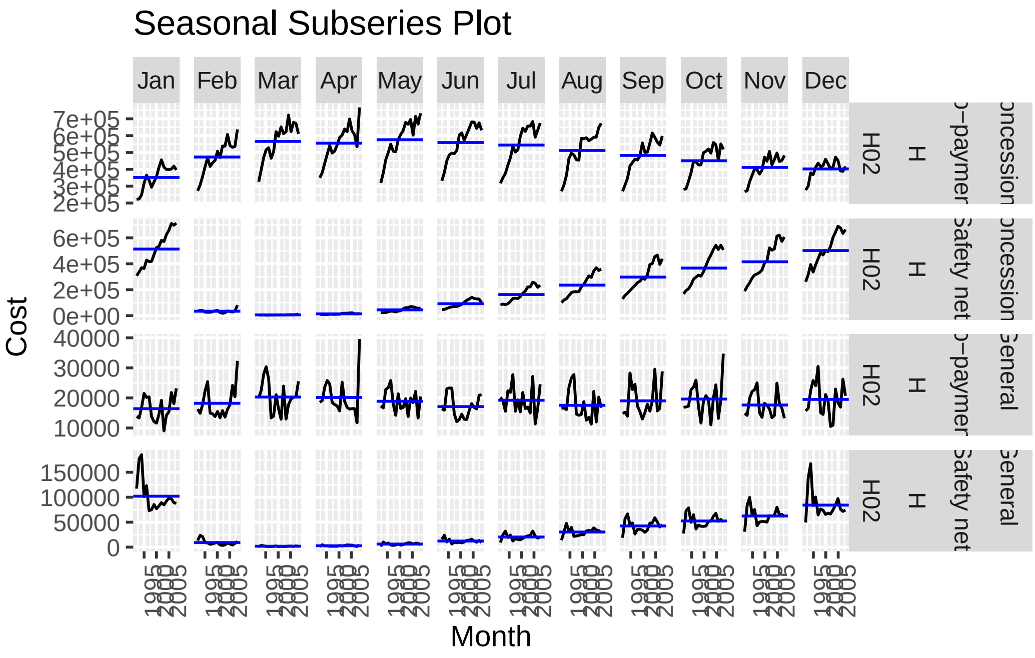
From a seasonality perspective, we can observe saftey net (and conversely the lowest co-payments) script costs increase throughout the year, peaking in January before dramatically falling off.

The autocorrelation plot supports this with a cosine shaped distribution, showing the highest correlation around December and January.

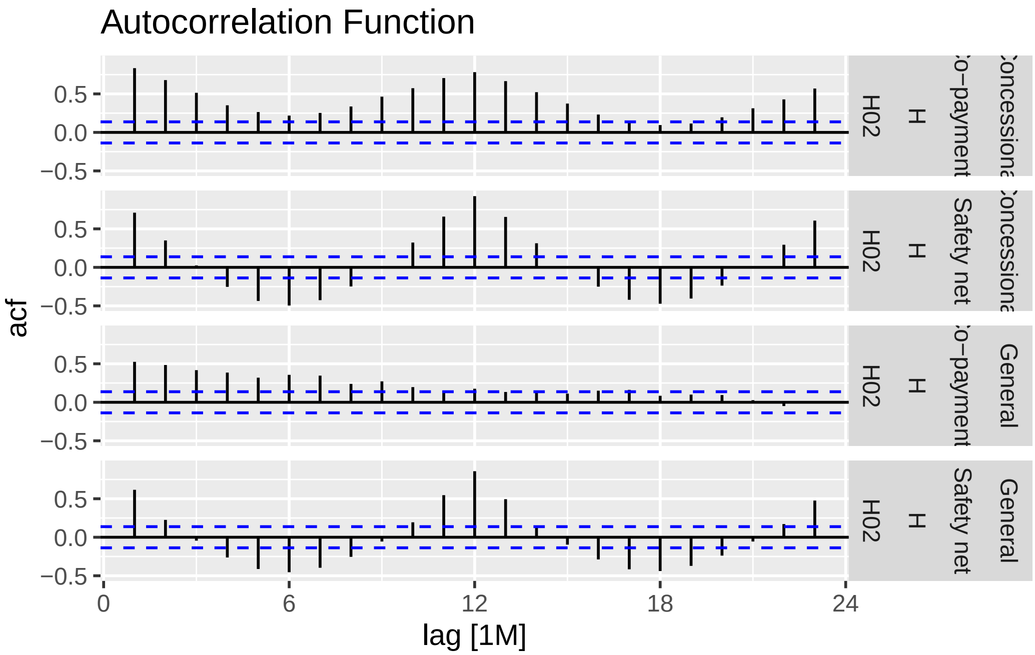
```
# PBS
pbs <- tsibbledata::PBS |>
  filter(ATC2=="H02")

plot_series(data=pbs, Cost)
```







```
Skipipng gglag plot: The data provided to contains more than one time series. Please filter a
```

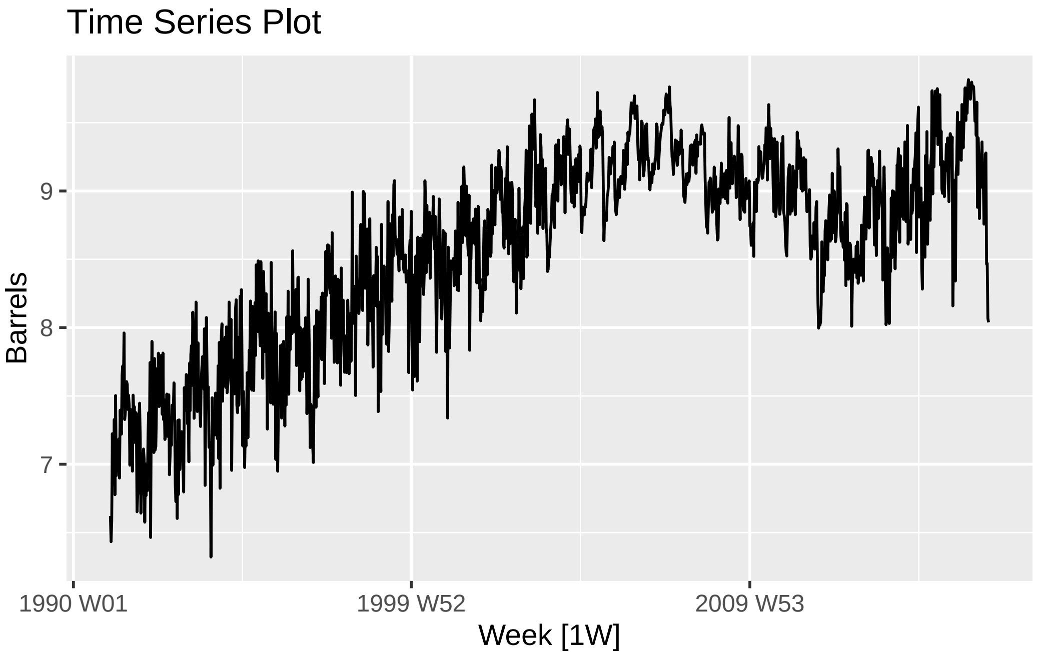


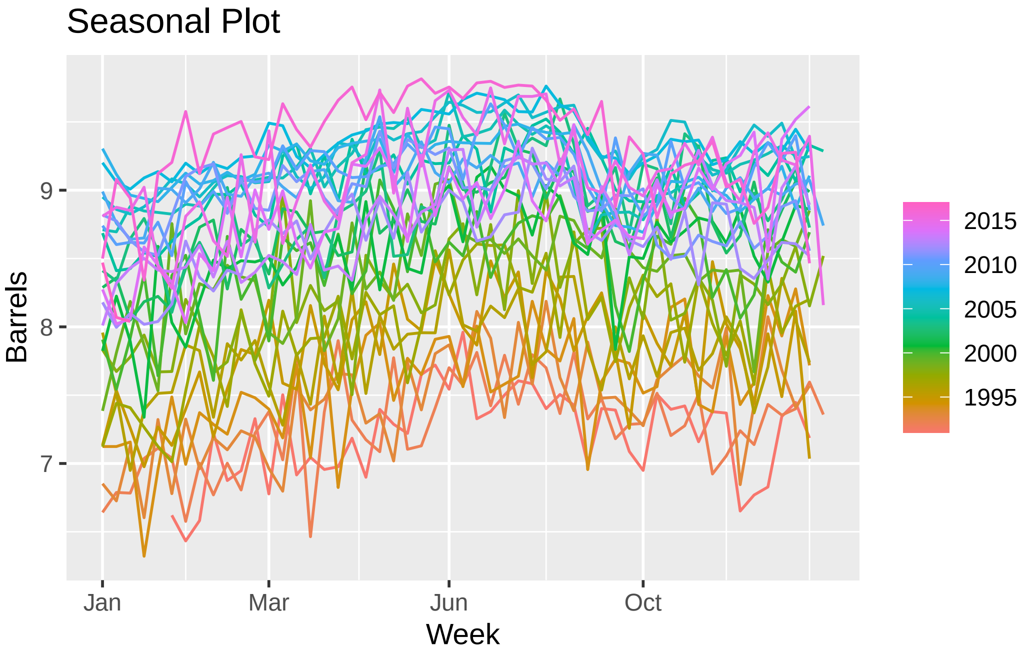
United States Gasoline Production Series

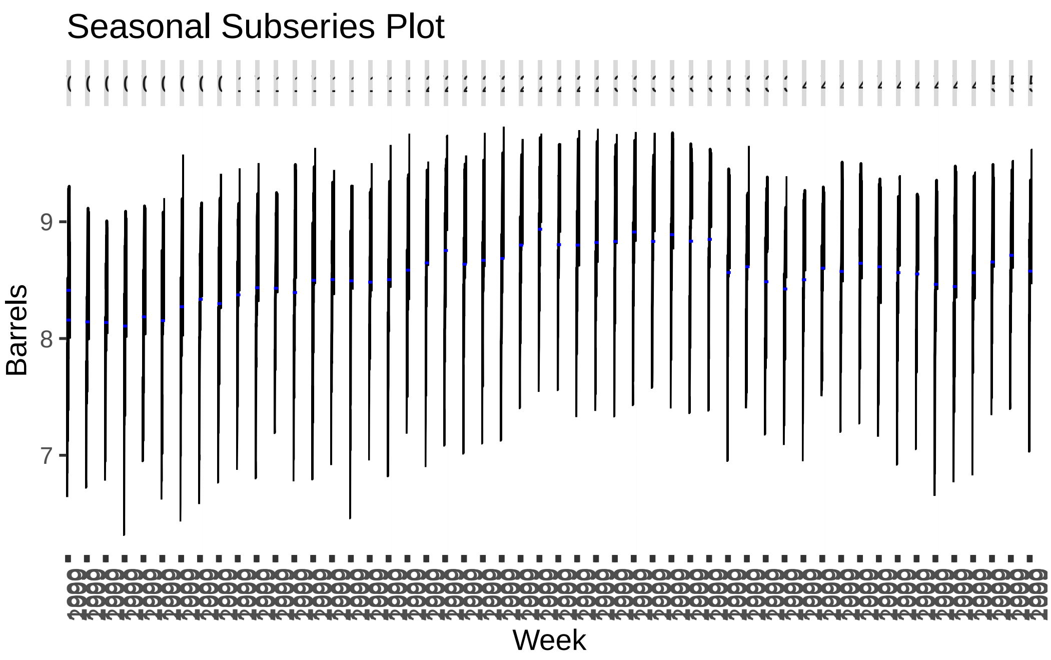
U.S. gasoline production is displayed in these graphs. Another interesting trend, where we see an increase up to the early 2000's before starting to dip and rise again with greater variability. The series is still highly autocorrolated.

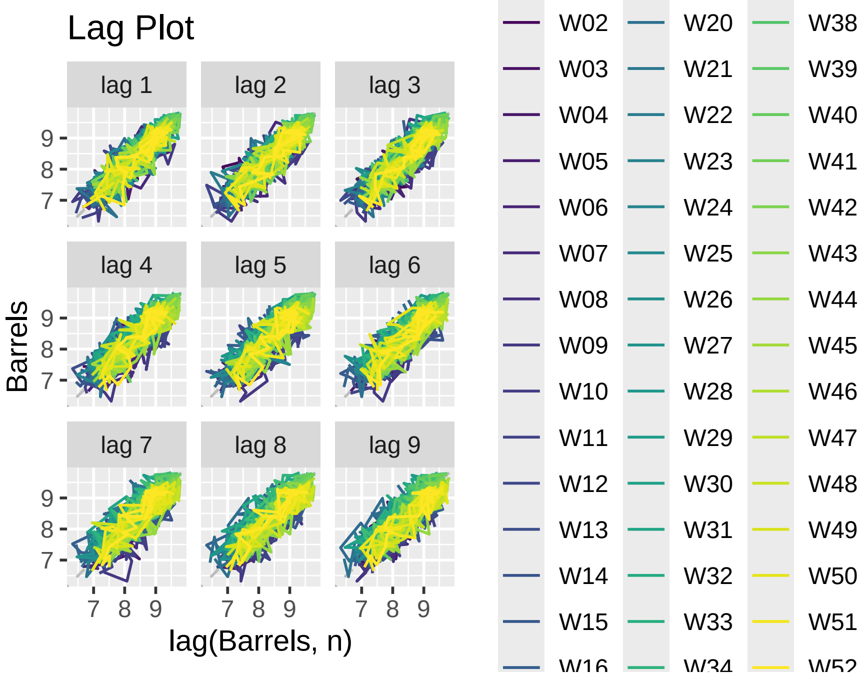
As expected, there's seasonality in summer months, however a surprising increase around major U.S. Holidays, specifically Thanksgiving and Christmas. Given this is related to U.S. production, I would not expect high travel to impact these numbers. More domain knowledge would perhaps explain this data.

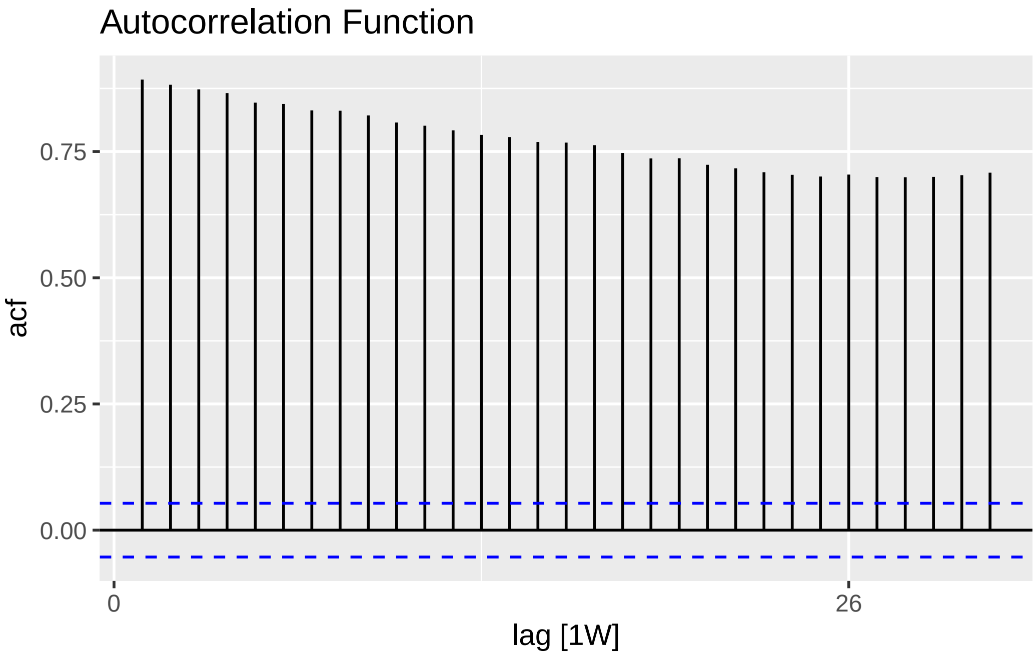
```
usgas <- fpp3::us_gasoline
plot_series(usgas, Barrels)
```


Time Series Plot
Barrels
9
8
7
1990 W01
1999 W52
2009 W53
Week [1W]


Seasonal Plot
Barrels
9
8
7
Jan
Mar
Jun
Oct
Week
2015
2010
2005
2000
1995


Seasonal Subseries Plot
Barrels
9
8
7
Week


Lag Plot
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
Barrels
9
8
7
7 8 9
lag(Barrels, n)
W02
W03
W04
W05
W06
W07
W08
W09
W10
W11
W12
W13
W14
W15
W16
W20
W21
W22
W23
W24
W25
W26
W27
W28
W29
W30
W31
W32
W33
W34
W38
W39
W40
W41
W42
W43
W44
W45
W46
W47
W48
W49
W50
W51
W52



Conclusion

Visualizations condense much information into a singular picture that takes advantage of the way our brains work to quickly understand the data we're working with. Combining the visuals with time series data allows us to observe the cadences of the data easily, and is a great starting point for working with time series data.