Time Series HW

Nick Climaco

2024-02-04

Contents


```
library(fpp3)
```

HA HW1

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?

```
bricks <- aus_production |>
  select(Bricks)

lynx <- pelt |>
  select(Lynx)

close <- gafa_stock |>
  select(Close)

demand <- vic_elec |>
  select(Demand)
```

The time intervals of each series:

- Bricks is quarterly

```
head(bricks)
```

```
## # A tsibble: 6 x 2 [1Q]
##   Bricks Quarter
##    <dbl>   <qtr>
## 1    189 1956 Q1
```

```
## 2    204 1956 Q2
## 3    208 1956 Q3
## 4    197 1956 Q4
## 5    187 1957 Q1
## 6    214 1957 Q2
```

- Lynx is yearly

```r
head(lynx)
```

```
## # A tsibble: 6 x 2 [1Y]
##    Lynx  Year
##   <dbl> <dbl>
## 1 30090  1845
## 2 45150  1846
## 3 49150  1847
## 4 39520  1848
## 5 21230  1849
## 6  8420  1850
```

- Close is daily

```r
close <- close |> mutate(Date = as_date(Date)) |>
  as_tsibble(index = Date)

head(close)
```

```
## # A tsibble: 6 x 3 [1D]
## # Key:       Symbol [1]
##   Close Date       Symbol
##   <dbl> <date>     <chr>
## 1  79.0 2014-01-02 AAPL
## 2  77.3 2014-01-03 AAPL
## 3  77.7 2014-01-06 AAPL
## 4  77.1 2014-01-07 AAPL
## 5  77.6 2014-01-08 AAPL
## 6  76.6 2014-01-09 AAPL
```

- Demand is every 30 mins

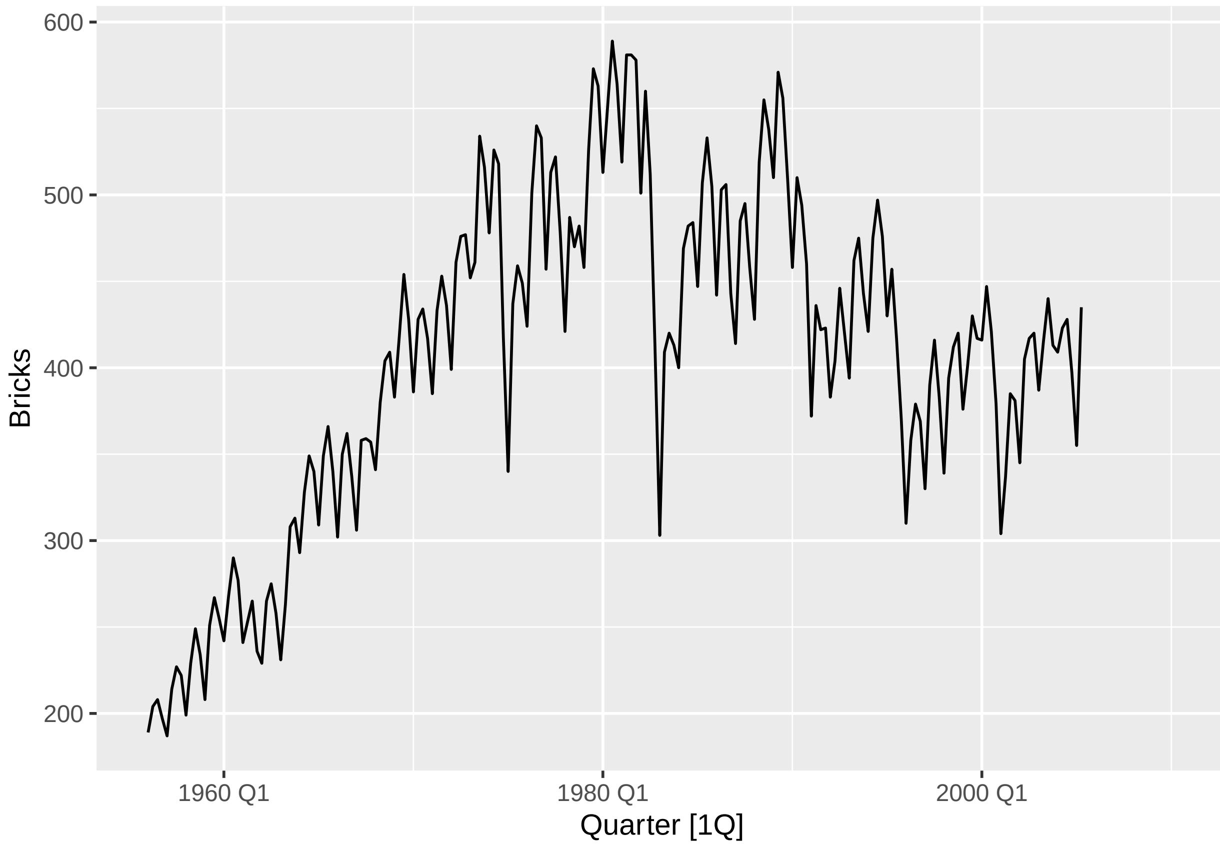
```r
head(demand)
```

```
## # A tsibble: 6 x 2 [30m] <Australia/Melbourne>
##   Demand Time
##    <dbl> <dttm>
## 1  4383. 2012-01-01 00:00:00
## 2  4263. 2012-01-01 00:30:00
## 3  4049. 2012-01-01 01:00:00
## 4  3878. 2012-01-01 01:30:00
## 5  4036. 2012-01-01 02:00:00
## 6  3866. 2012-01-01 02:30:00
```

- Use autoplot() to produce a time plot of each series.
- For the last plot, modify the axis labels and title.

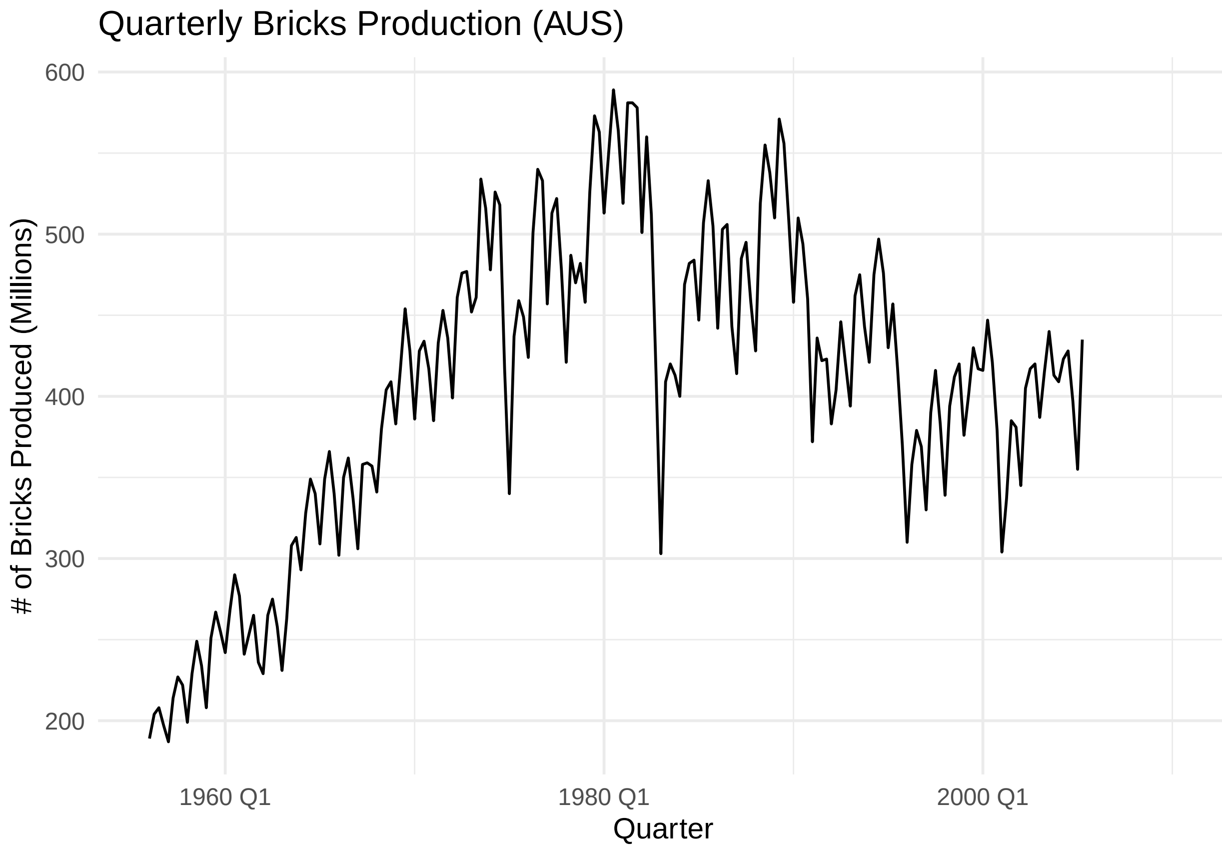
Bricks, Lynx, Close, and Demand

```r
autoplot(bricks, Bricks)
```

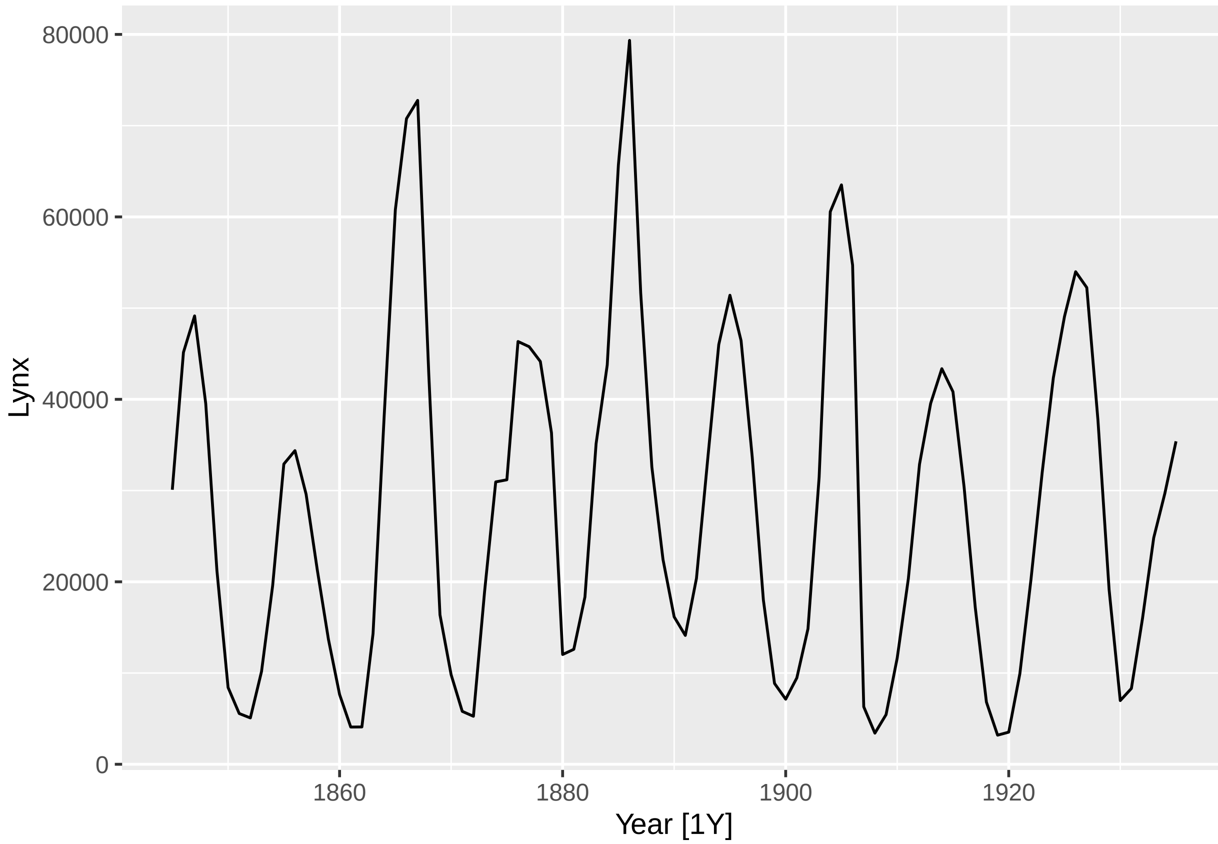


Bricks

```r
autoplot(bricks, Bricks) +
  labs(title="Quarterly Bricks Production (AUS)",
       x= "Quarter",
       y= "# of Bricks Produced (Millions)") +
  theme_minimal()
```

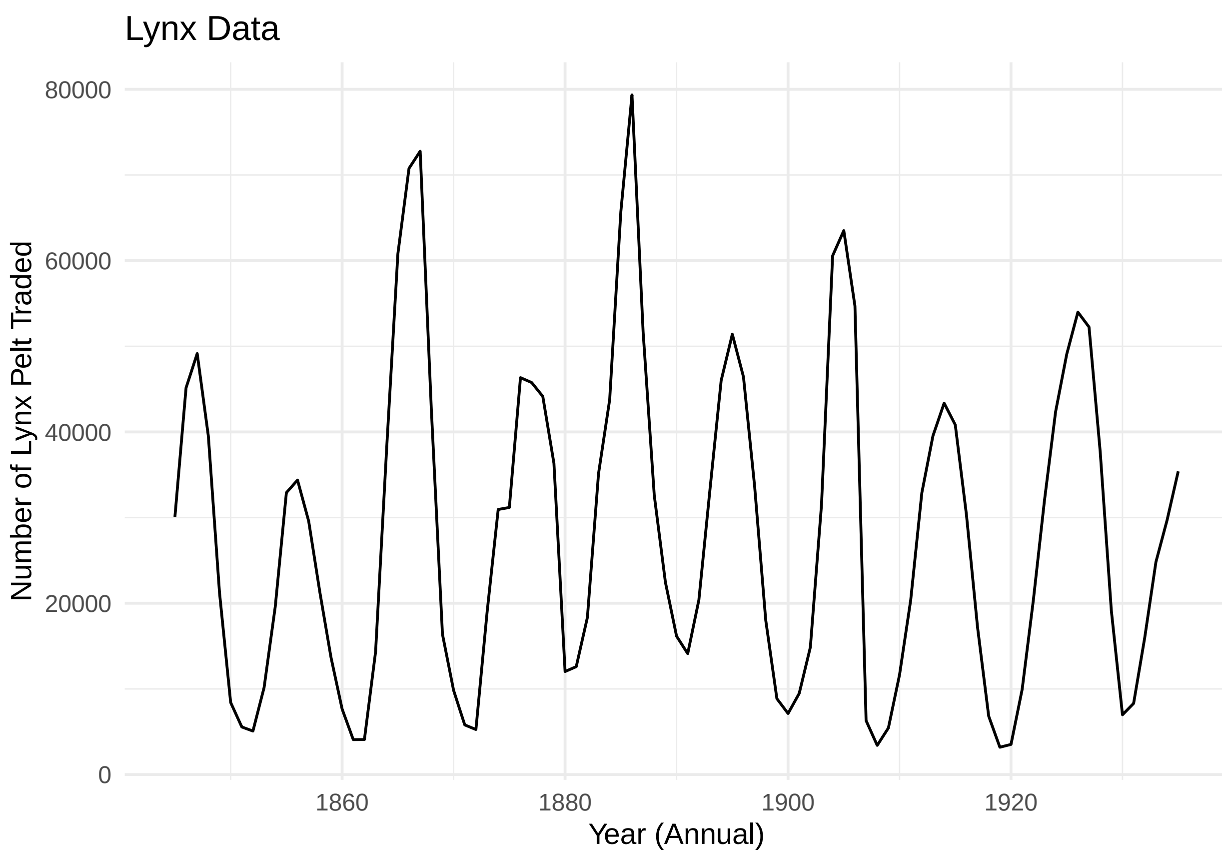



```
autoplot(lynx, Lynx)
```

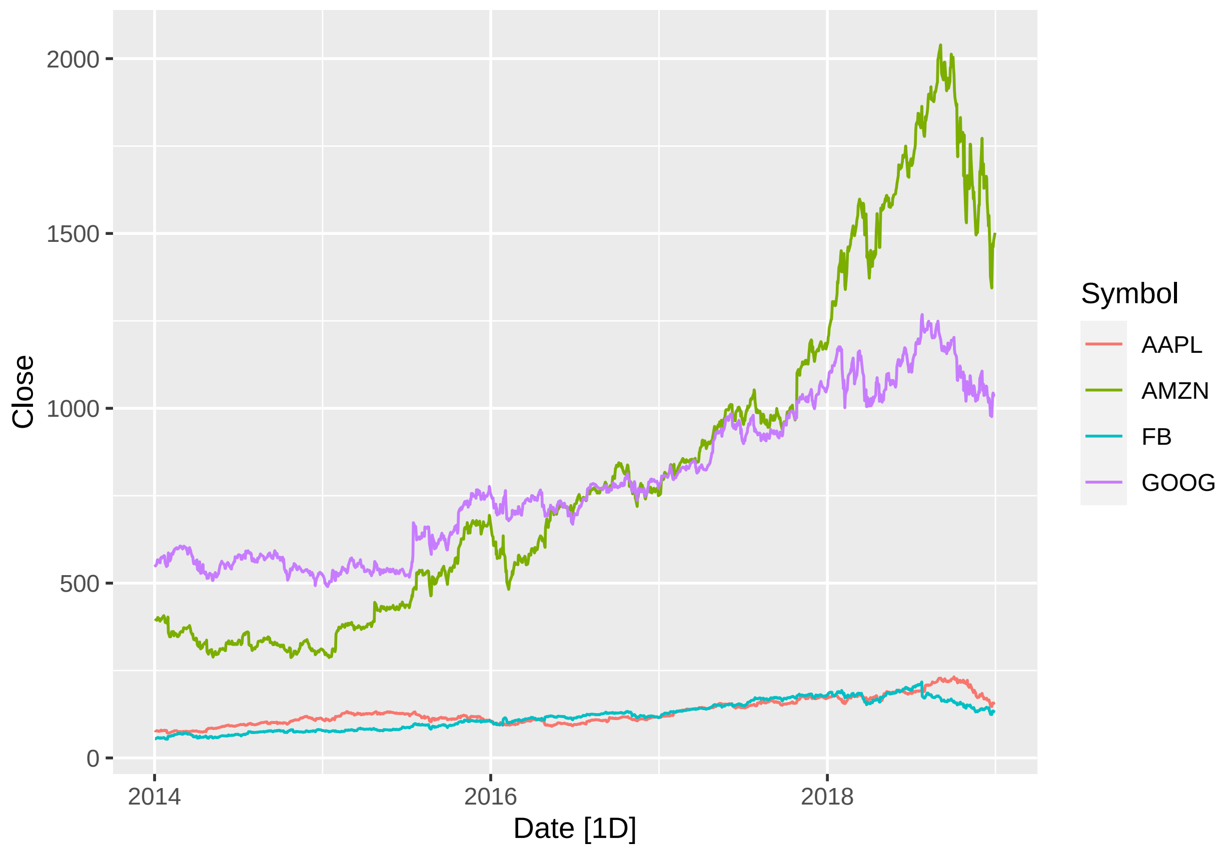


Lynx

```r
autoplot(lynx, Lynx) +
  labs(title = "Lynx Data",
       x= "Year (Annual)",
       y = "Number of Lynx Pelt Traded") +
  theme_minimal()
```

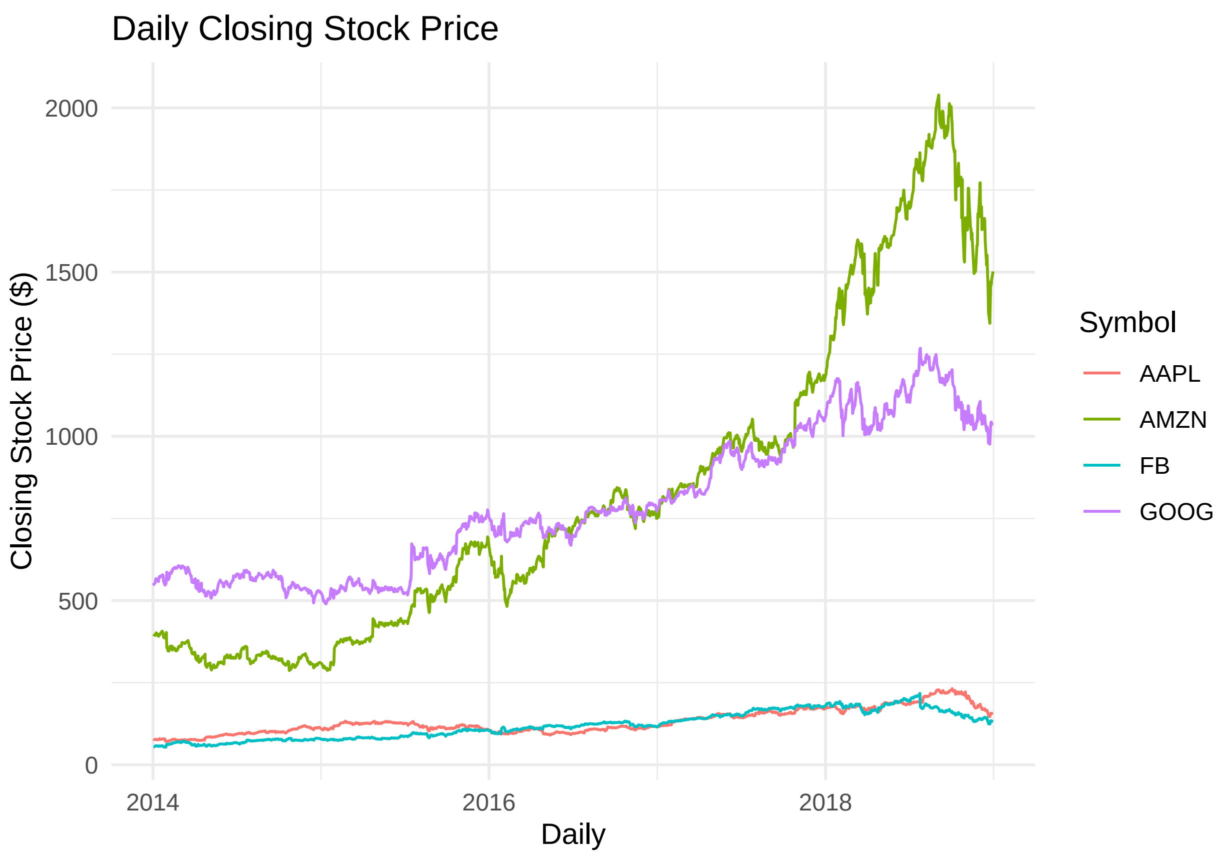



```r
autoplot(close, Close)
```

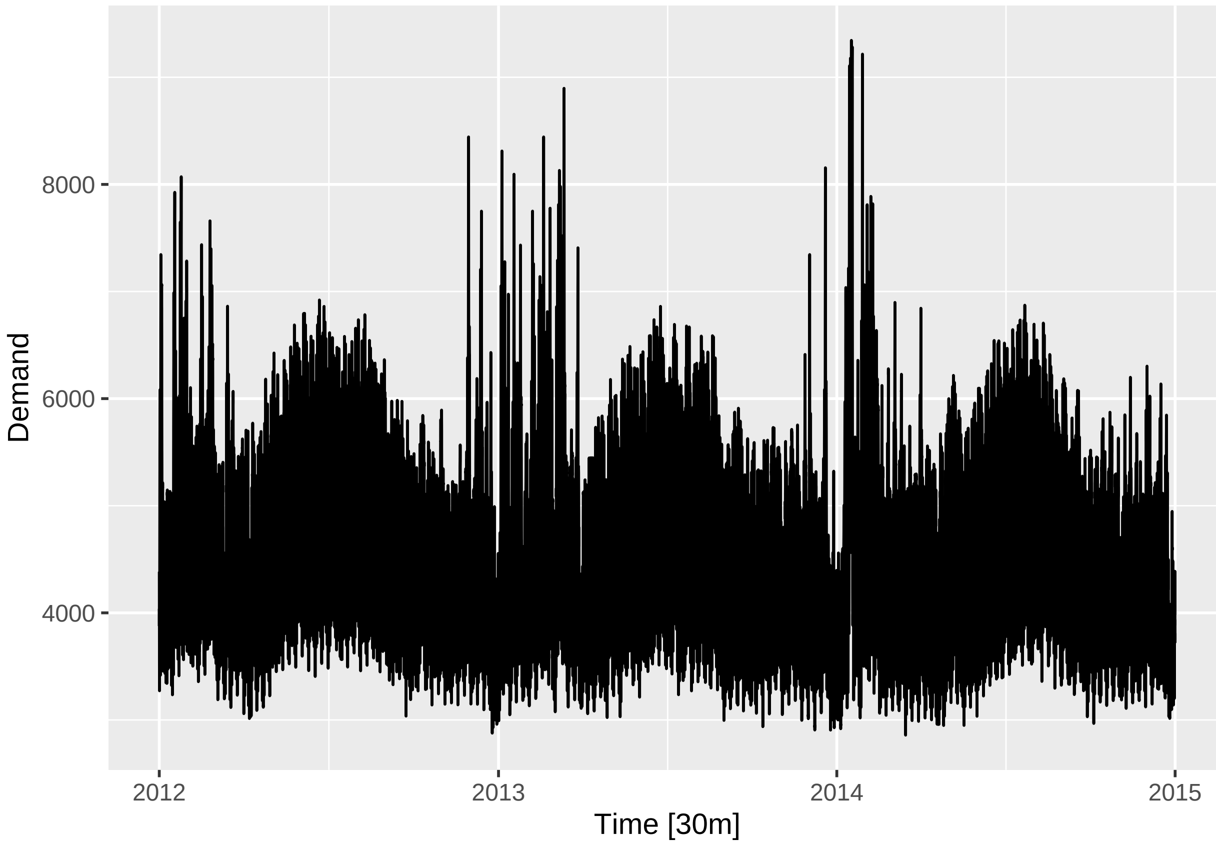


Close

```r
autoplot(close, Close) +
  labs(title = "Daily Closing Stock Price",
       x="Daily",
       y = "Closing Stock Price ($)") +
  theme_minimal()
```

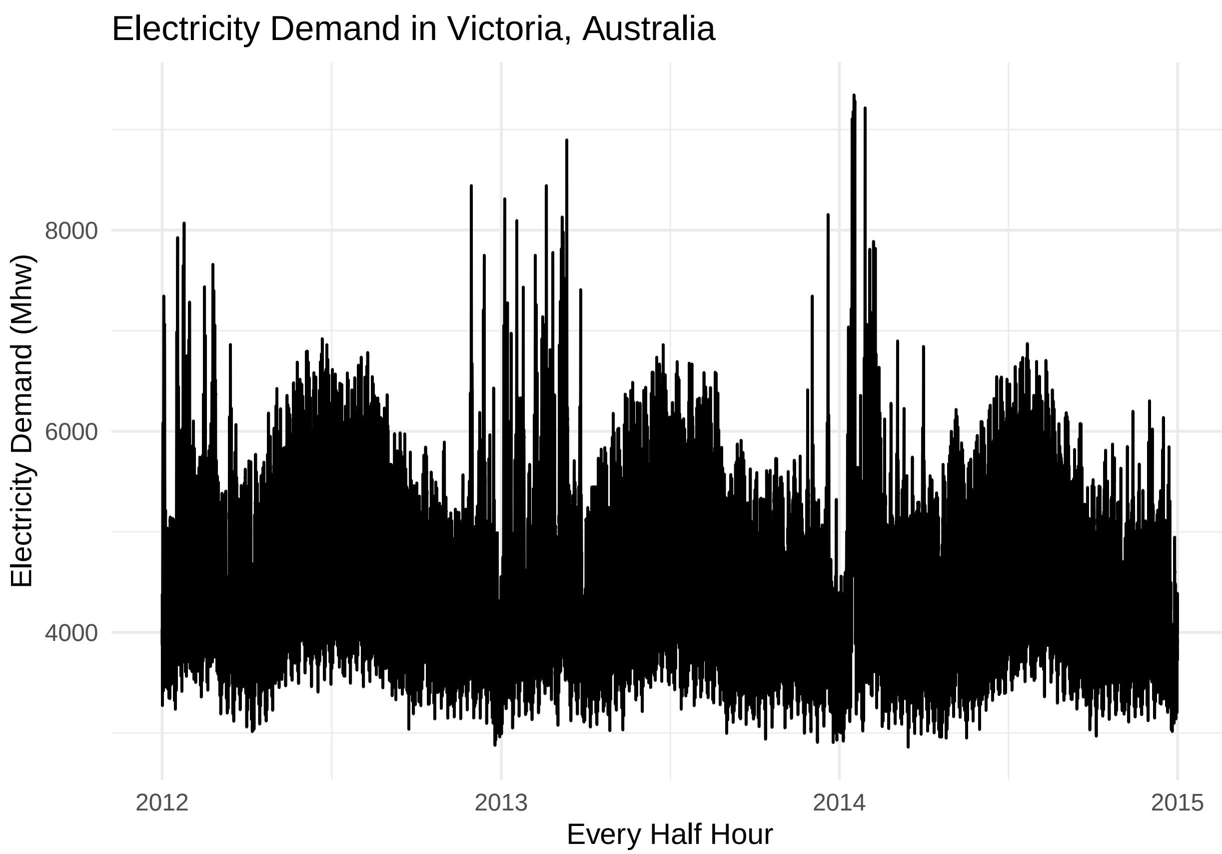



```
autoplot(demand, Demand)
```



Demand

```
autoplot(demand,Demand) +
  labs(title = "Electricity Demand in Victoria, Australia ",
       x = "Every Half Hour",
       y = "Electricity Demand (Mhw)") +
  theme_minimal()
```



Exercise 2

First, we need to group the data by `Symbol` so then we can filter `Close` to have the maximum or peak closing price for each stock. We then select the columns we want to see which are `Date`, `Symbol`, and `Close`.

```
close |>
  group_by(Symbol) |>
  filter(Close == max(Close)) |>
  select(Date, Symbol, Close)
```

```
## # A tsibble: 4 x 3 [1D]
## # Key:       Symbol [4]
## # Groups:    Symbol [4]
##   Date       Symbol Close
##   <date>     <chr>  <dbl>
## 1 2018-10-03 AAPL    232.
## 2 2018-09-04 AMZN   2040.
## 3 2018-07-25 FB      218.
## 4 2018-07-26 GOOG   1268.
```

```
# fidn the days where the closing price was at its peak i.e. max
```

Exercise 3

Code provided by textbook. We downloaded the tute1.csv file and used view() to examine the data. Using the as_tsibble() function, we convert the data to a time series where the time interval is quarterly setting the index to `Quarter`. Plotting the timeseries a line plot using ggplot() and geom_line() and utilized facet_grid()

to create a subgrid of the timeseries.

Parts

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

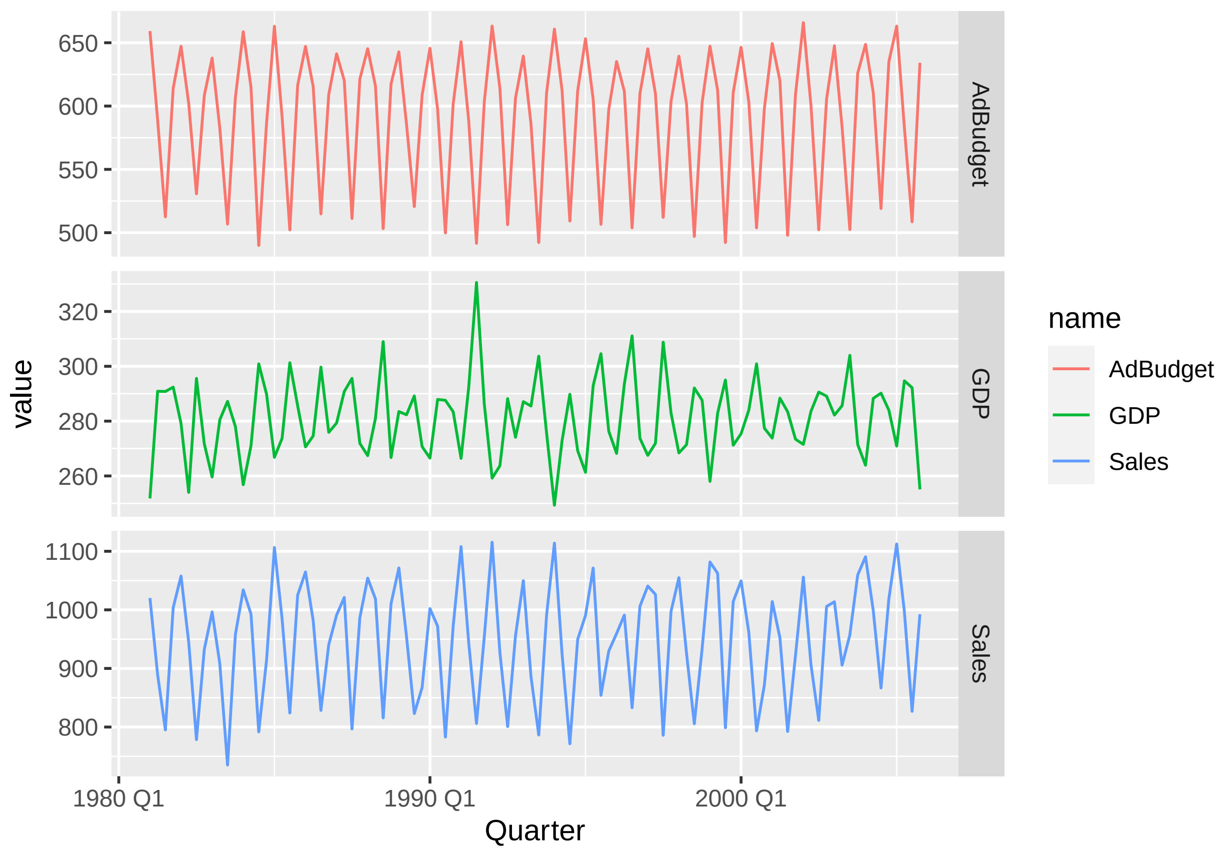
```
tute1 <- readr::read_csv("tute1.csv")
#View(tute1)
```

b. Convert the data to time series

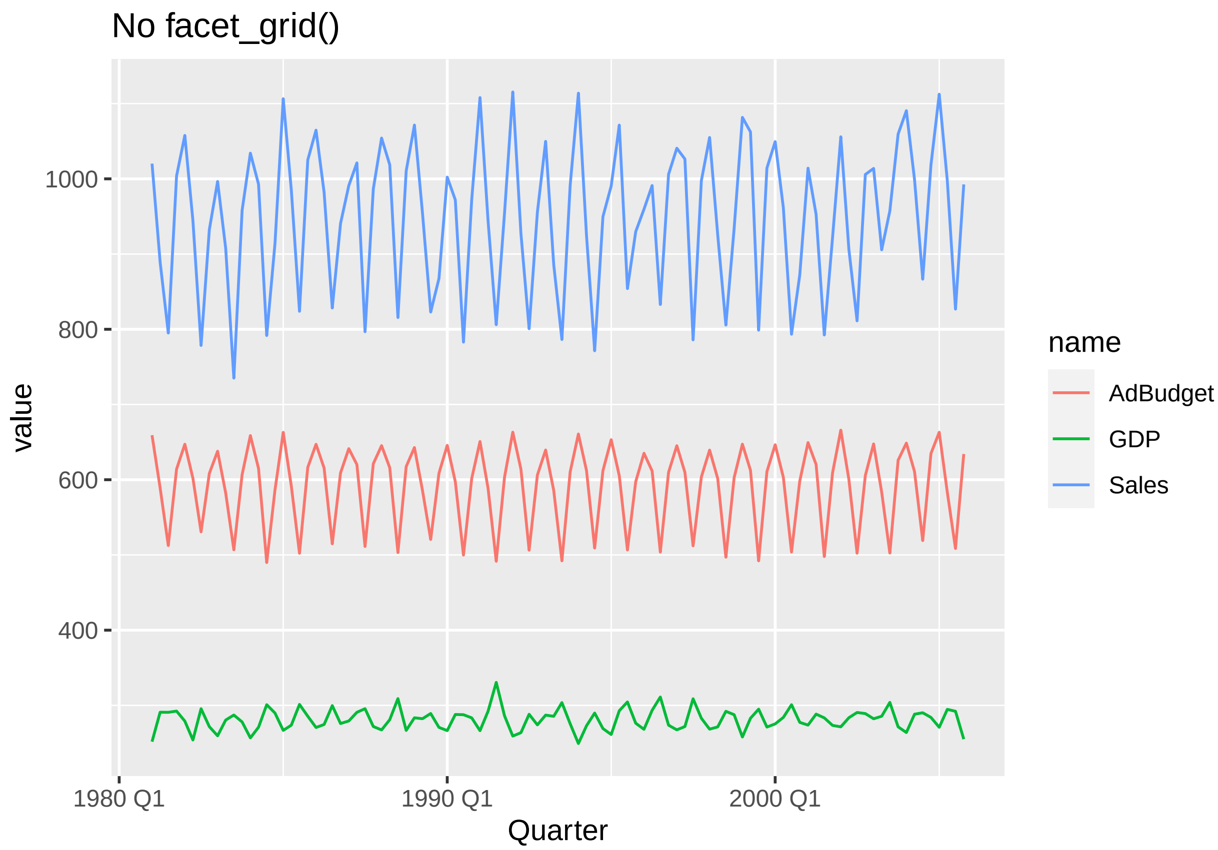
```
mytimeseries <- tute1 |>
  mutate(Quarter = yearquarter(Quarter)) |>
  as_tsibble(index = Quarter)
```

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")
```




```
mytimeseries |>
  pivot_longer(-Quarter) |>
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  labs(title = "No facet_grid()") +
  geom_line()
```



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

Exercise 4

First, we install the package USgas and converted Usgas to a timeseries using as_tsibble() and setting the index to `year` and key to `state`. Afterwards, we wanted to see the gas consumption for the following states; Maine, Vermont, New Hampshire, Massachussetts, Connecticut and Rhode Island. To do that using the filter() function and used autoplot() to visualize the timeseries.

```
#install.packages("USgas")
library(USgas)
```

Install the USgas package.

```
head(us_total)
```

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

```
##   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
```

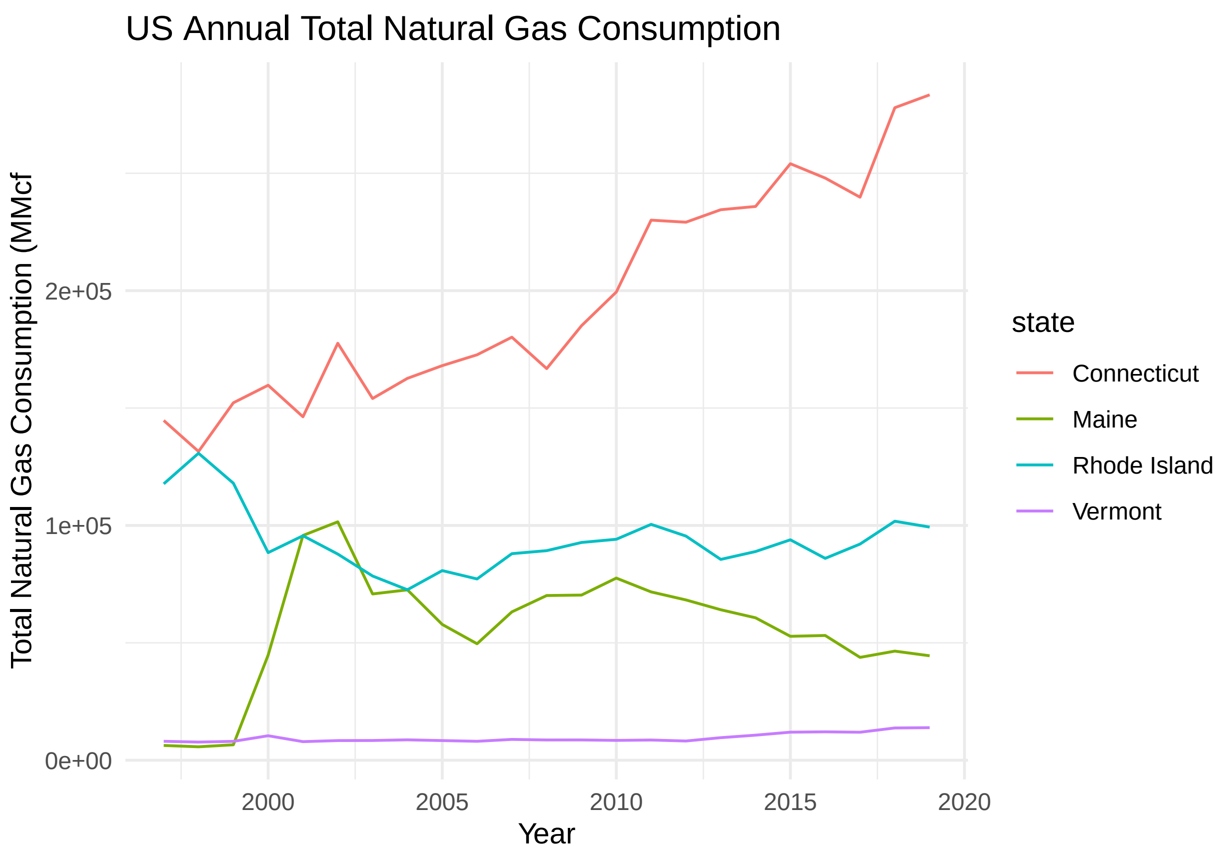
```r
us_total <- us_total |>
  as_tsibble(index = year, key = state)
head(us_total)
```

```
## # 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
```

```r
filtered_us_total <- us_total |>
  filter(state %in% c("Maine", "Vermont", "New Hamposhire", "Massachusetss", "Connecticut", "Rhode Islan

autoplot(filtered_us_total, y ) +
  labs(title='US Annual Total Natural Gas Consumption',
       x = "Year",
       y = "Total Natural Gas Consumption (MMcf") +
  theme_minimal()
```

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).



Exercise 5

Using read_excel() from readxl package to load in the data, in order to create a tsibble object that is identical to the tourism tsibble from the tsibble package. Created the tourism tsibble object with setting the index to Quarter, a timeseries with quarterly time interval, and setting the key to `Region`, `State`, and `Purpose` to mimic the original tourism tsibble. Then, we wanted find which combination of `Region` and `Purpose` had the highest average of overnight trips. We did that by first grouping the timeseries by the `Region` and `Purpose` then using summarize() to calculate the mean() of `Trips`. Next, filtered the timeseries for the max overnight trips of each possible combination. FInally, arranged() the output in descending order to show the the combination with the highest average of overnight trips.

```r
library(readxl)
data <- read_excel('tourism.xlsx')
```

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

```r
head(tourism) # tourism from the tsibble package
```

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
tourism_2 <- data |>
  mutate(Quarter = yearquarter(Quarter)) |>
  as_tsibble(index = Quarter, key = c(Region, State, Purpose))

head(tourism_2)
```

```
## # 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
tourism_2 |>
  group_by(Region, Purpose) |>
  summarize(AverageTrips = mean(Trips, na.rm = TRUE)) |>
  filter(AverageTrips == max(AverageTrips)) |>
  arrange(desc(AverageTrips)) |> head()
```

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

```
## # A tsibble: 6 x 4 [1Q]
## # Key:       Region, Purpose [6]
## # Groups:    Region [6]
##   Region          Purpose  Quarter AverageTrips
##   <chr>           <chr>      <qtr>        <dbl>
## 1 Melbourne       Visiting 2017 Q4         985.
## 2 Sydney          Business 2001 Q4         948.
## 3 South Coast     Holiday  1998 Q1         915.
## 4 North Coast NSW Holiday  2016 Q1         906.
## 5 Brisbane        Visiting 2016 Q4         796.
## 6 Gold Coast      Holiday  2002 Q1         711.
```

```r
tourism_2 |>
  group_by(State) |>
  summarize(TotalTrips = sum(Trips)) |>
  as_tsibble(index = Quarter) |>
  head()
```

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

```
## # A tsibble: 6 x 3 [1Q]
```

```
## # Key:       State [1]
##   State Quarter TotalTrips
##   <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.
```

Exercise 8

Created a function that will generate the following plots each timeseries input, autoplot(), gg_season(), gg_subseries(), gg_lag(), and ACF(), since we will be performing similar operation on multiple timeseries. While answering the following question:

- Can you spot any seasonality, cyclicity and trend?
- What do you learn about the series?
- What can you say about the seasonal patterns?
- Can you identify any unusual years?

```r
plot_time_series <- function(data) {

  p1 <- autoplot(data) + theme_minimal()

  p2 <- gg_season(data) + theme_minimal()

  p3 <- gg_subseries(data) + theme_minimal()

  p4 <- gg_lag(data, geom = "point") + theme_minimal()

  p5 <- ACF(data) |>
    autoplot() + theme_minimal()

  print(p1)
  print(p2)
  print(p3)
  print(p4)
  print(p5)
}
```

```r
plot_time_series_without_gg_season <- function(data) {

  p1 <- autoplot(data) + theme_minimal()

  p3 <- gg_subseries(data) + theme_minimal()

  p4 <- gg_lag(data, geom = "point") + theme_minimal()

  p5 <- ACF(data) |>
    autoplot() + theme_minimal()

  print(p1)
  print(p3)
```

```
  print(p4)
  print(p5)
}
```

Time Series :

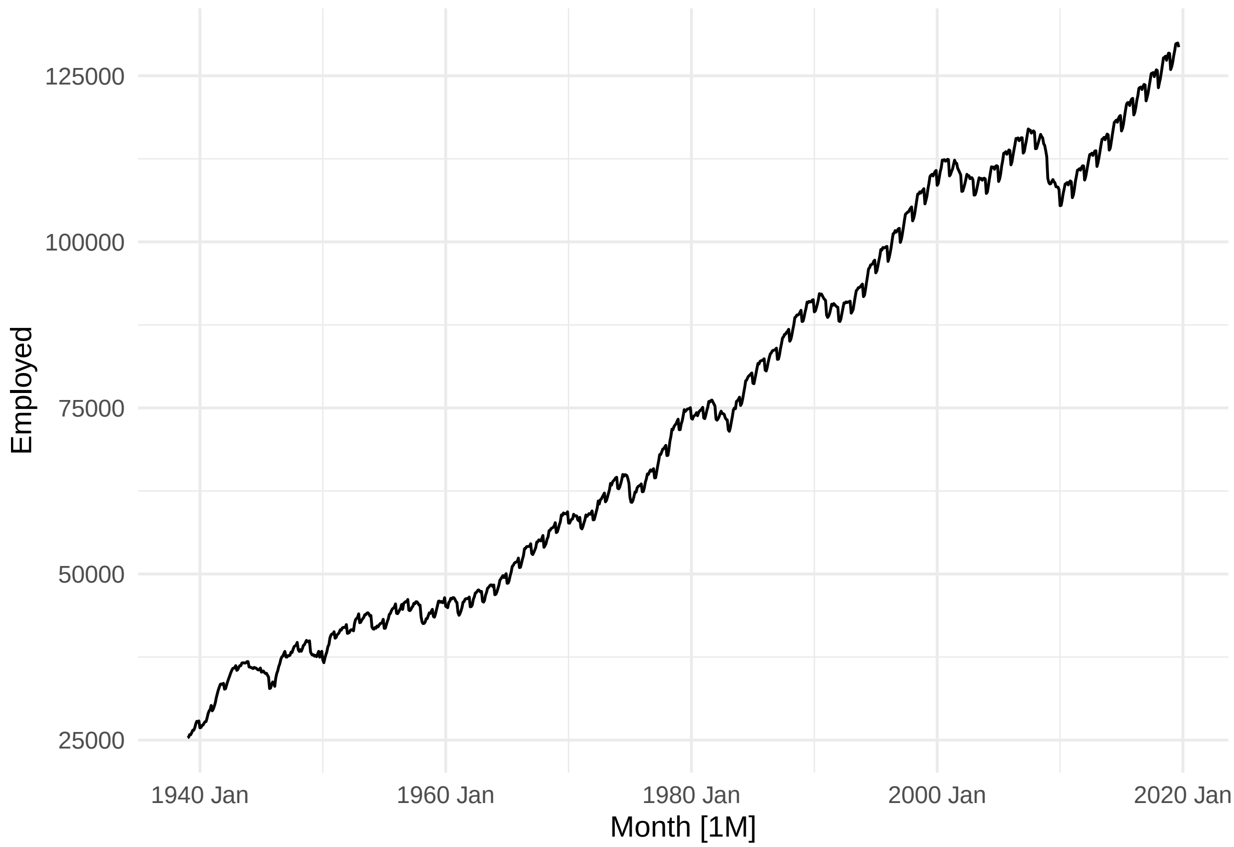
us_employment

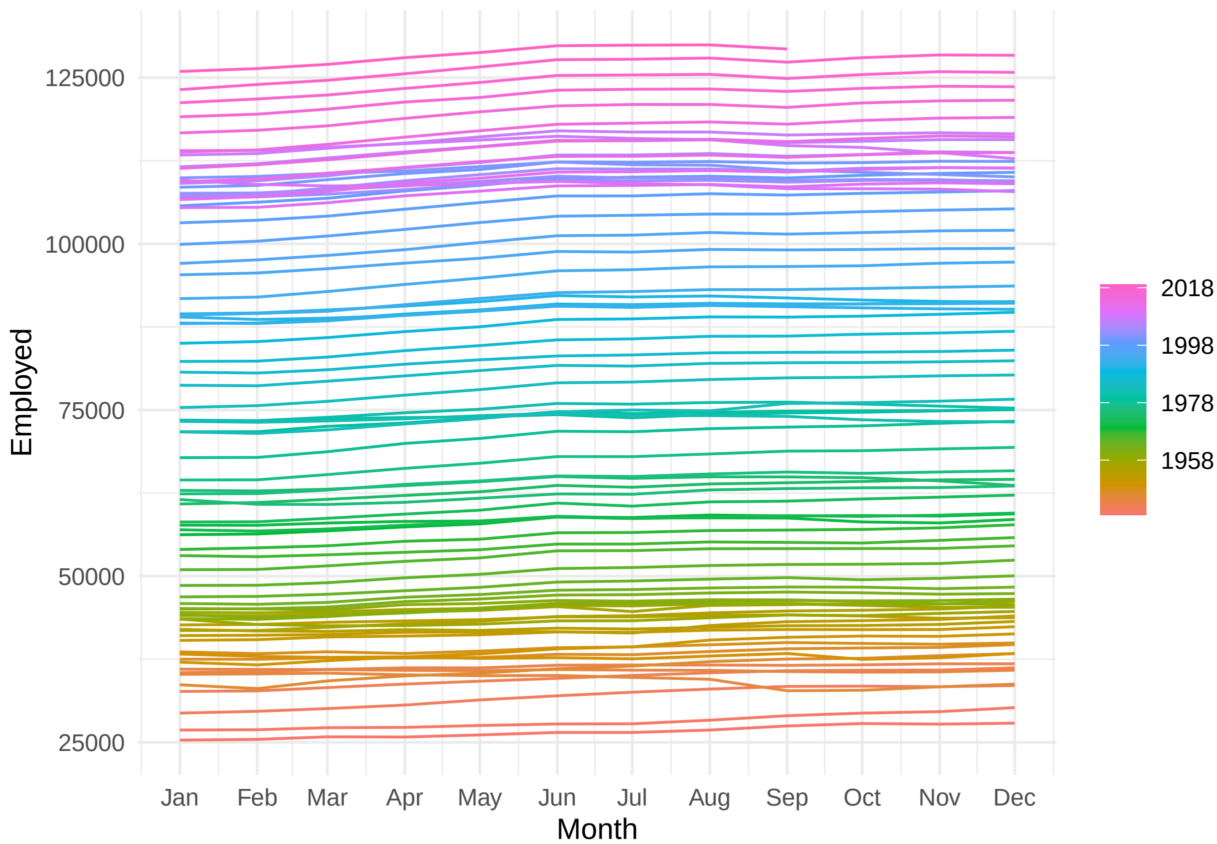
- Can you spot any seasonality, cyclicity and trend?

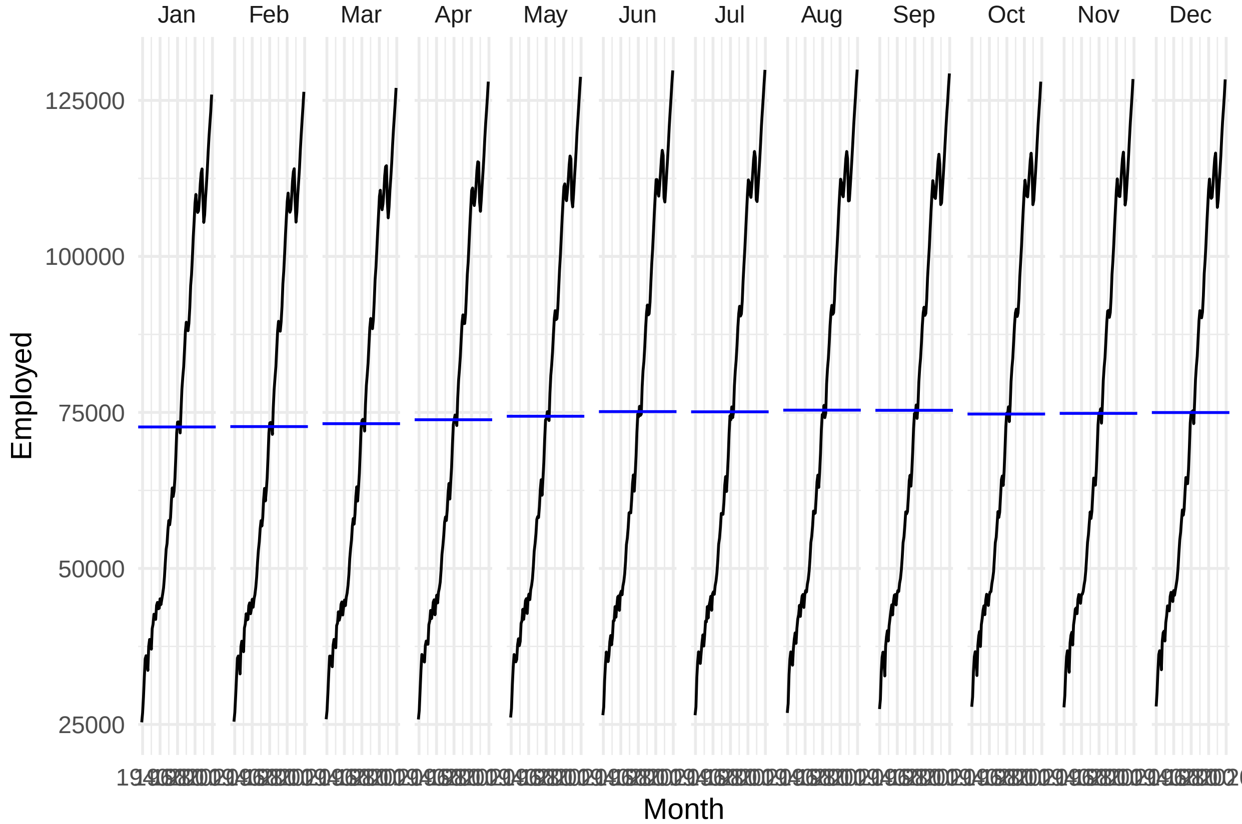
Overall, this time series has only linearly increasing trend. Some may argue that there may be a seasonal aspect to it when taking a look at the peaks and valleys generated from autoplot() but looking at gg_season() plot it is almost perfect parallel line indicating little to no seasonality. Look the plots, we can say that job growth in private sector in the US has been linearly increasing over the years. It seems that the years 2001 and 2008 had impacts on job growth where we a decrease in the number of employed in the private sector. Moreover, there is a strongt positive correlation between lag as n is from 1 to 9 due the general increasing trend of the timeseries.

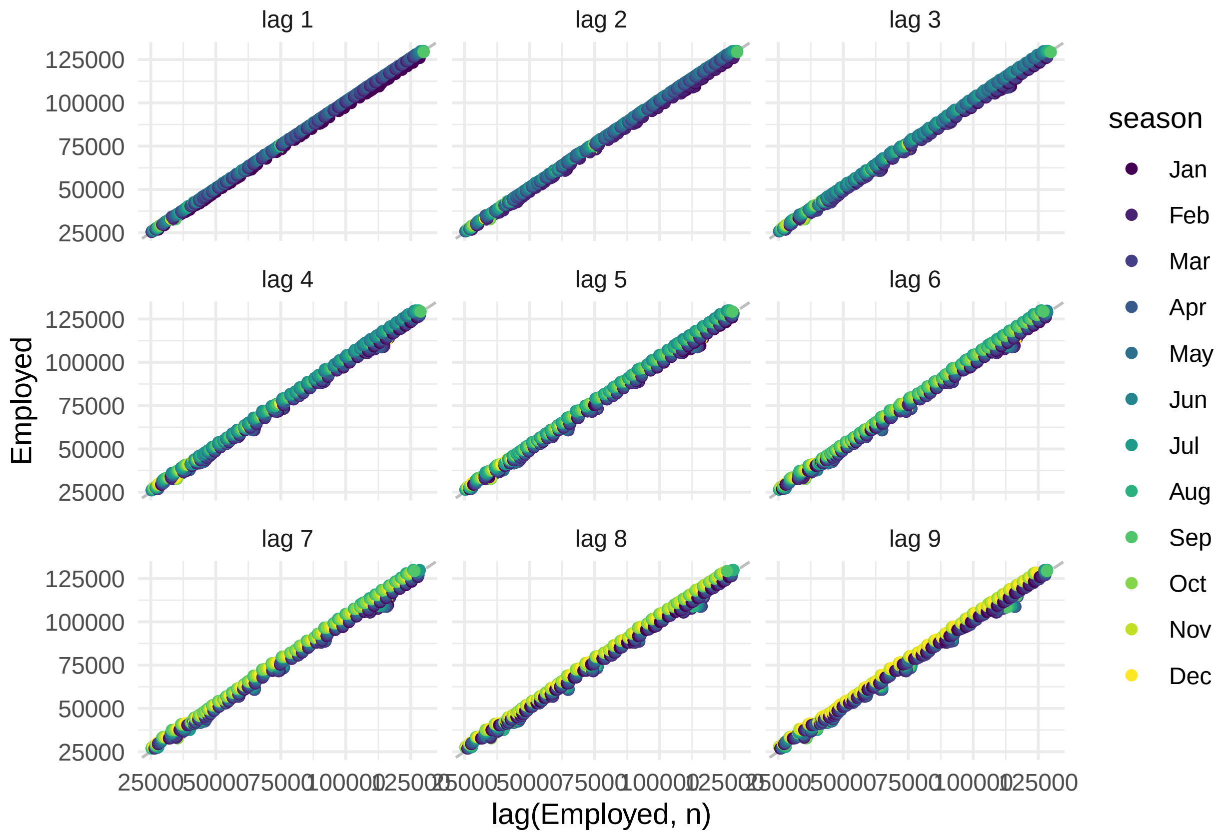
```
total_private <- us_employment |>
  filter(Title == "Total Private") |> select(Month, Employed)
```

```
print(plot_time_series(total_private))
```

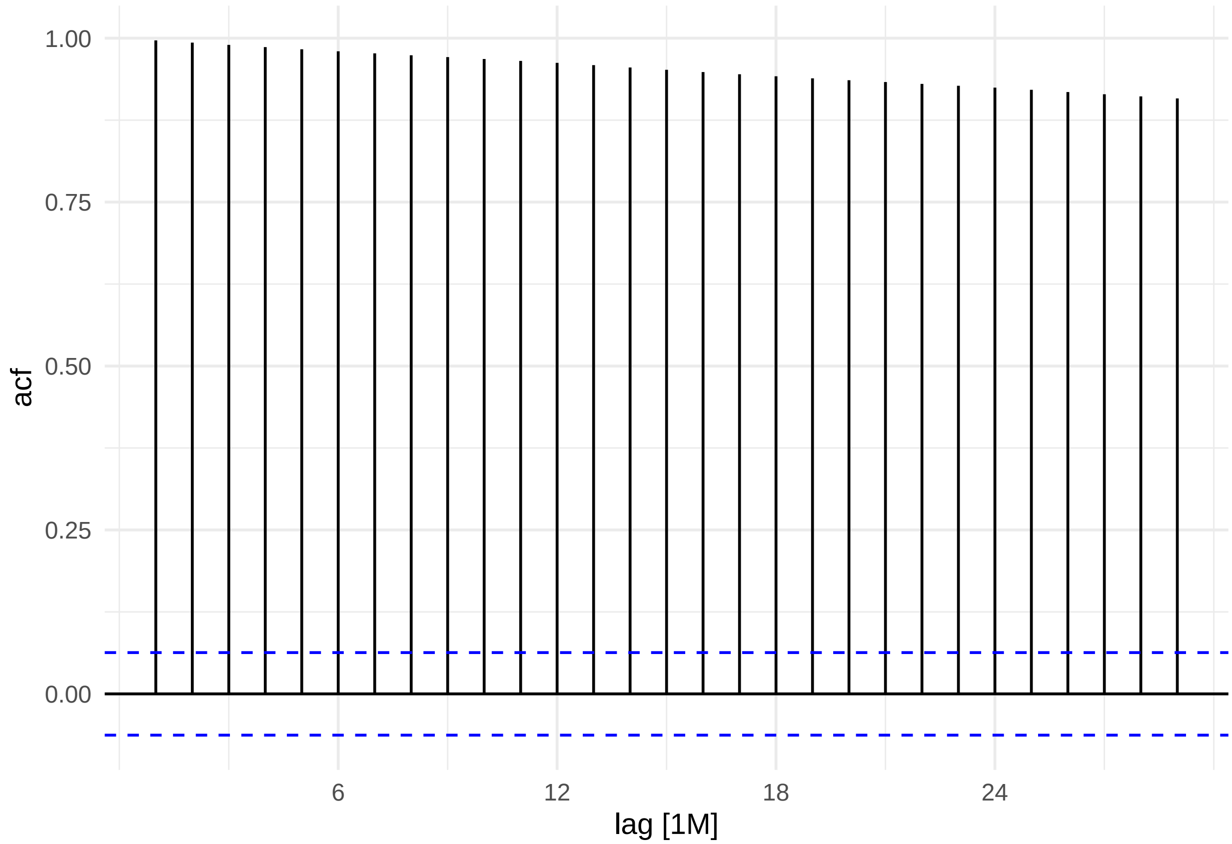


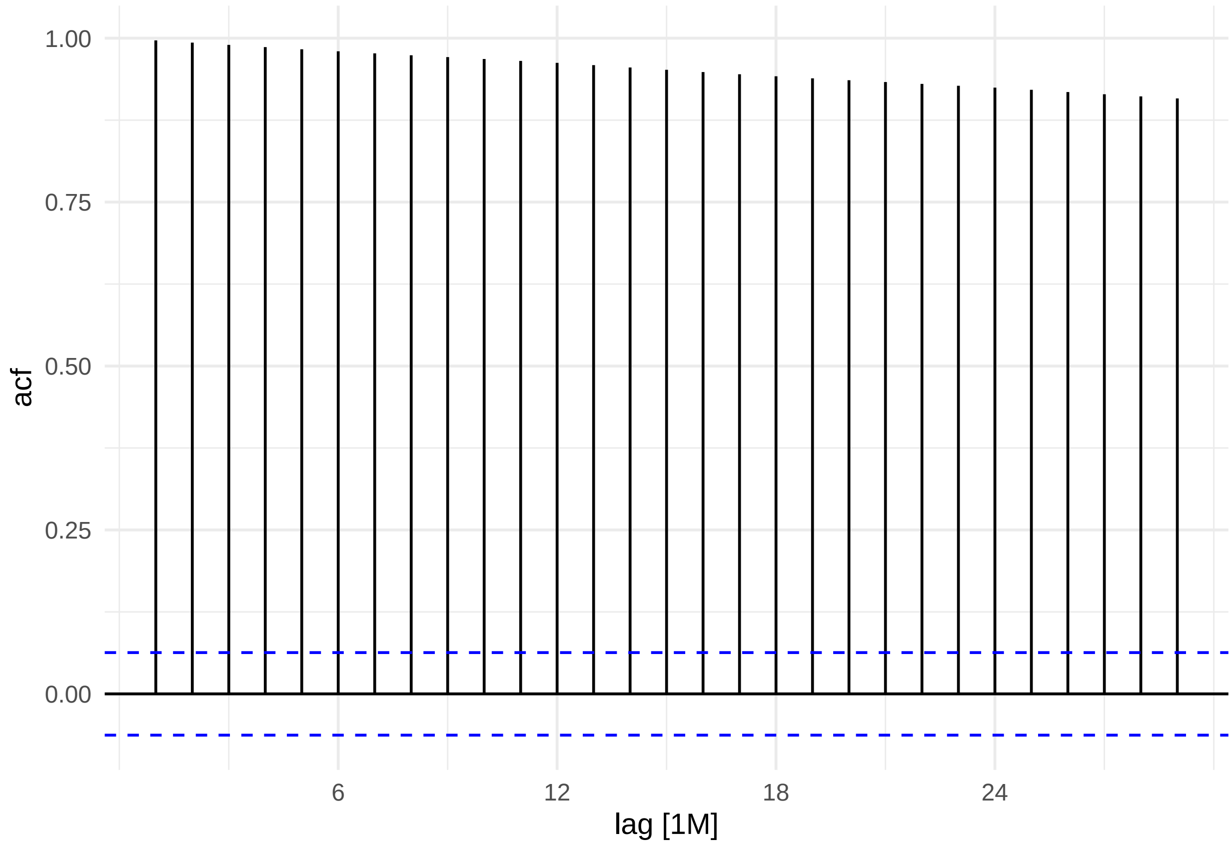

125000
100000
75000
50000
25000
Employed
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
2018
1998
1978
1958


Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
125000
100000
75000
50000
25000
Employed
Month



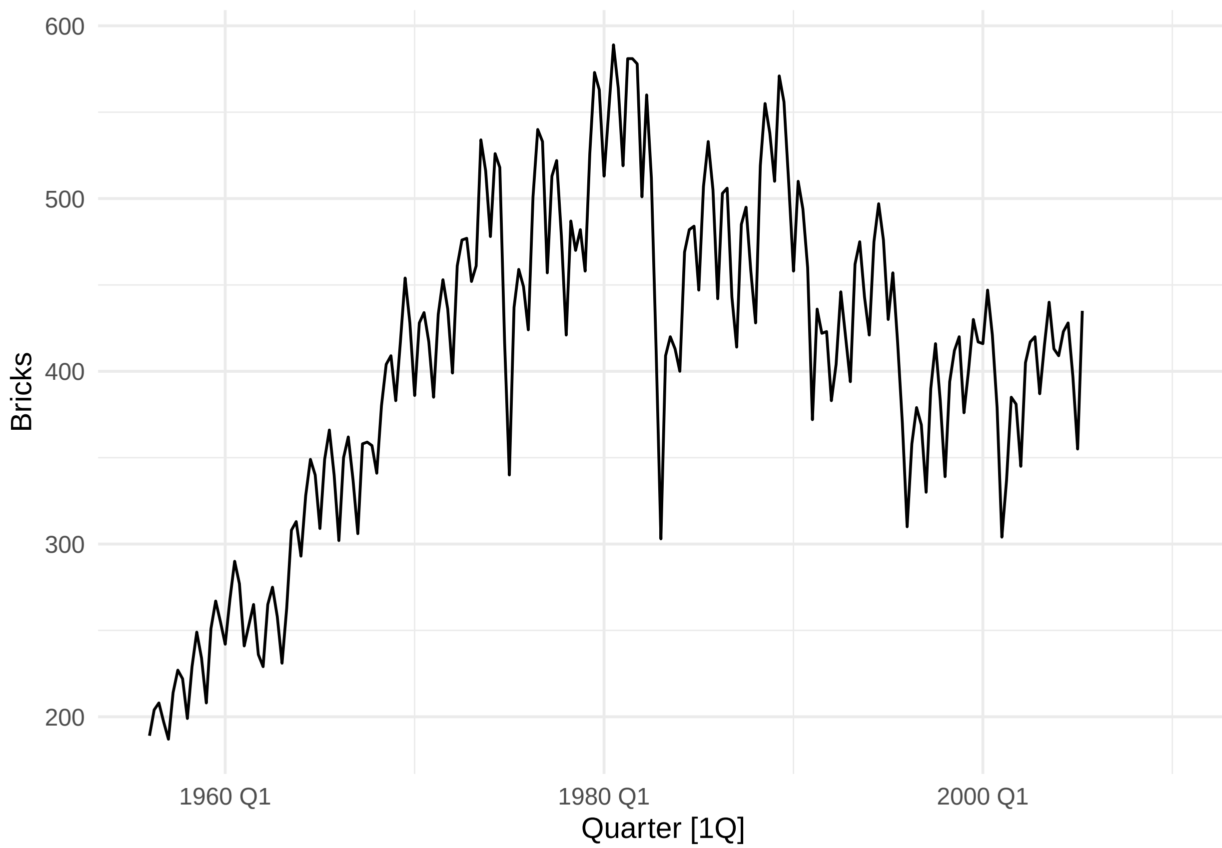
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
Employed
lag(Employed, n)
125000
100000
75000
50000
25000
season
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

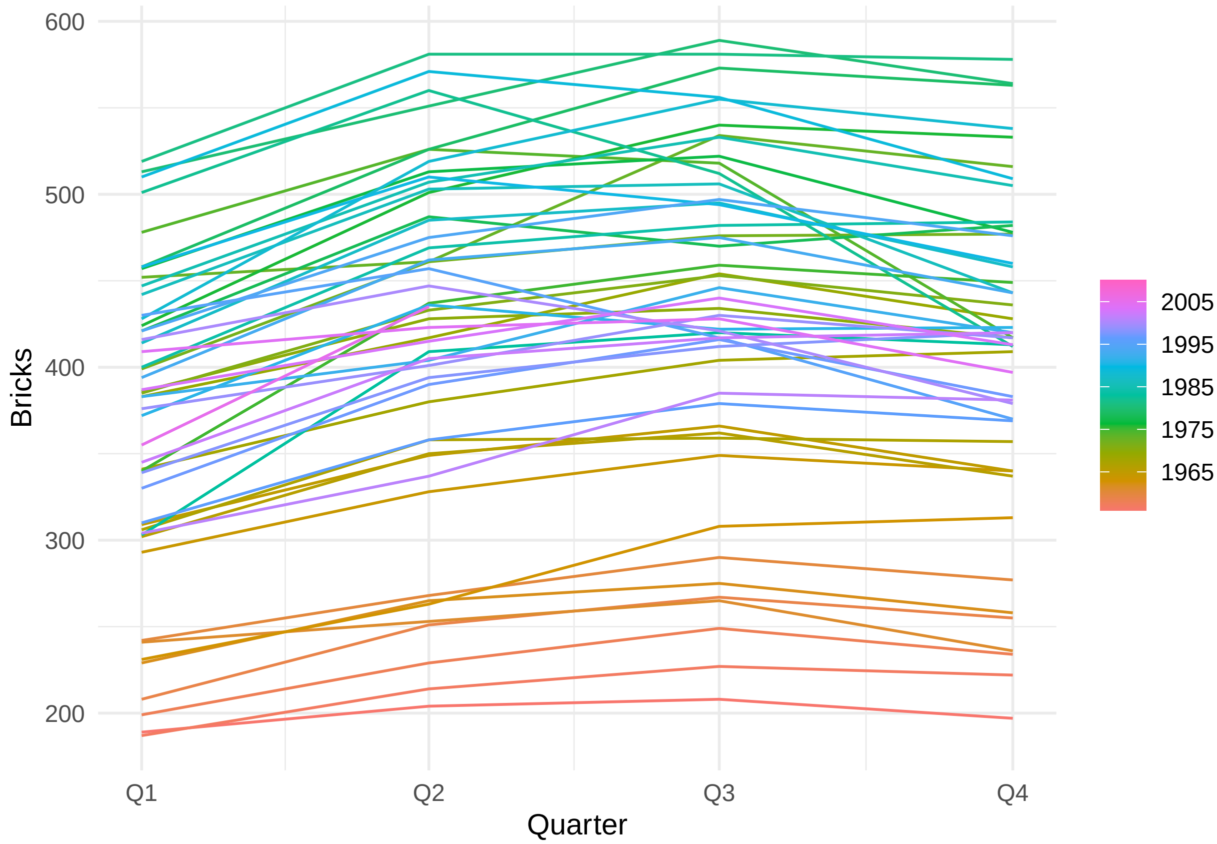

1.00
0.75
0.50
0.25
0.00
acf
6
12
18
24
lag [1M]

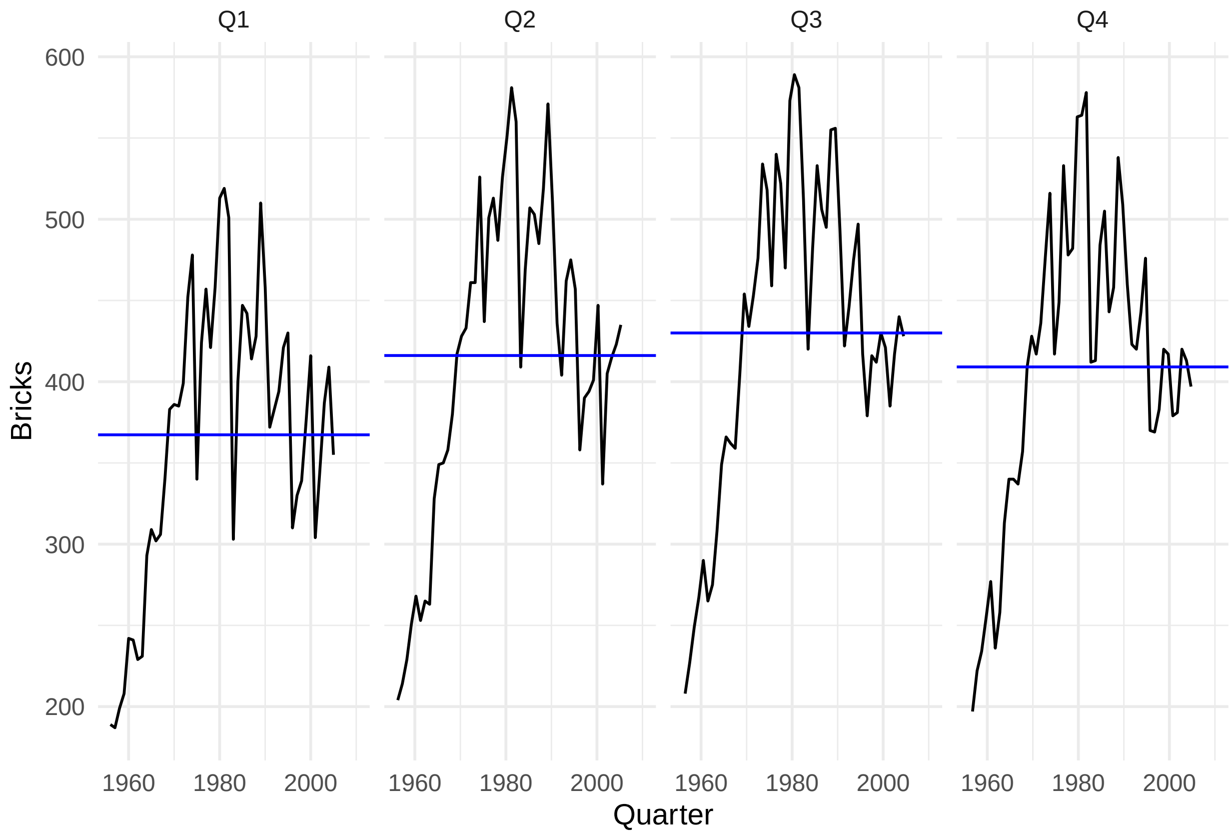


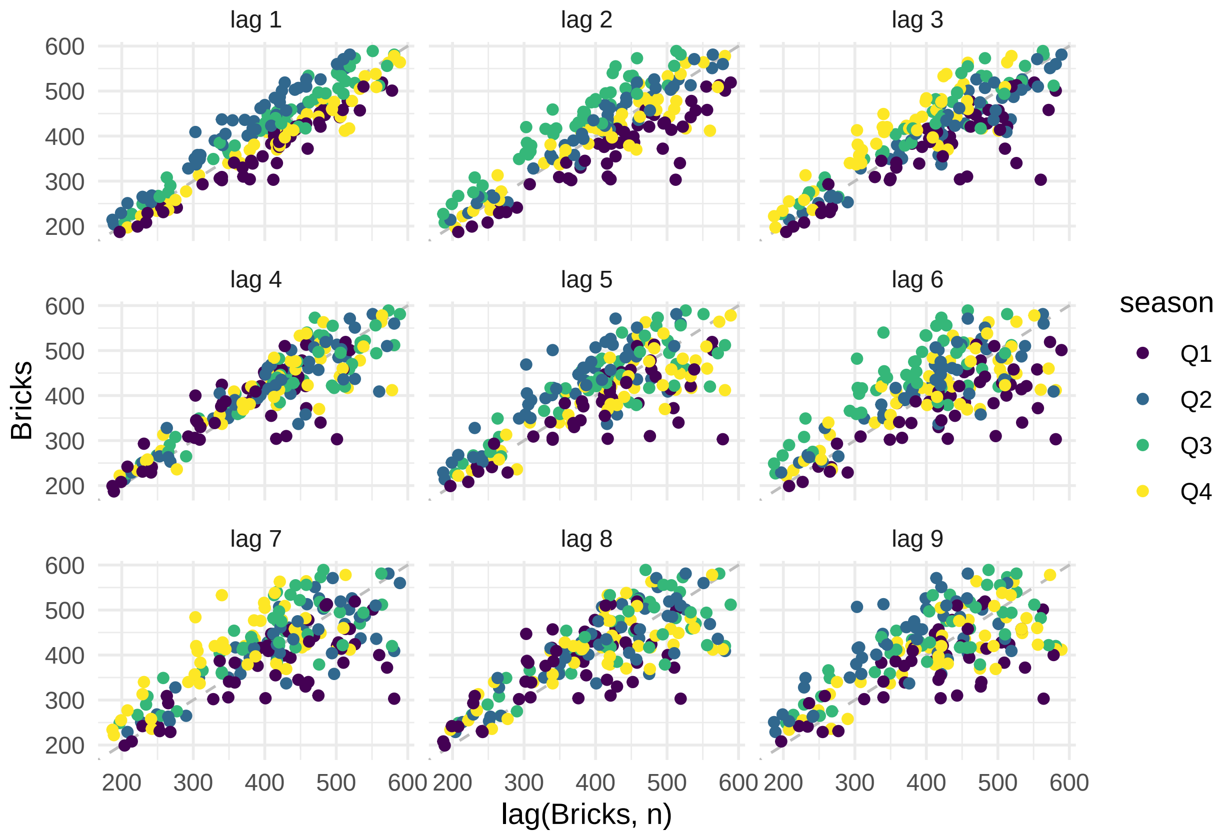
aus_production From 1960 - 1970s, there is an increased trend of bricks production in Australia. Then, that trend starts decreasing in the 1980s. The plots do not suggest any cyclic behavior but I suspect there exists some seasonality within each quarter. The plots supports my suspicion of a strong seasonality in the time series with its strong positive correlation from lag 1 to 9.

```
print(plot_time_series(bricks))
```

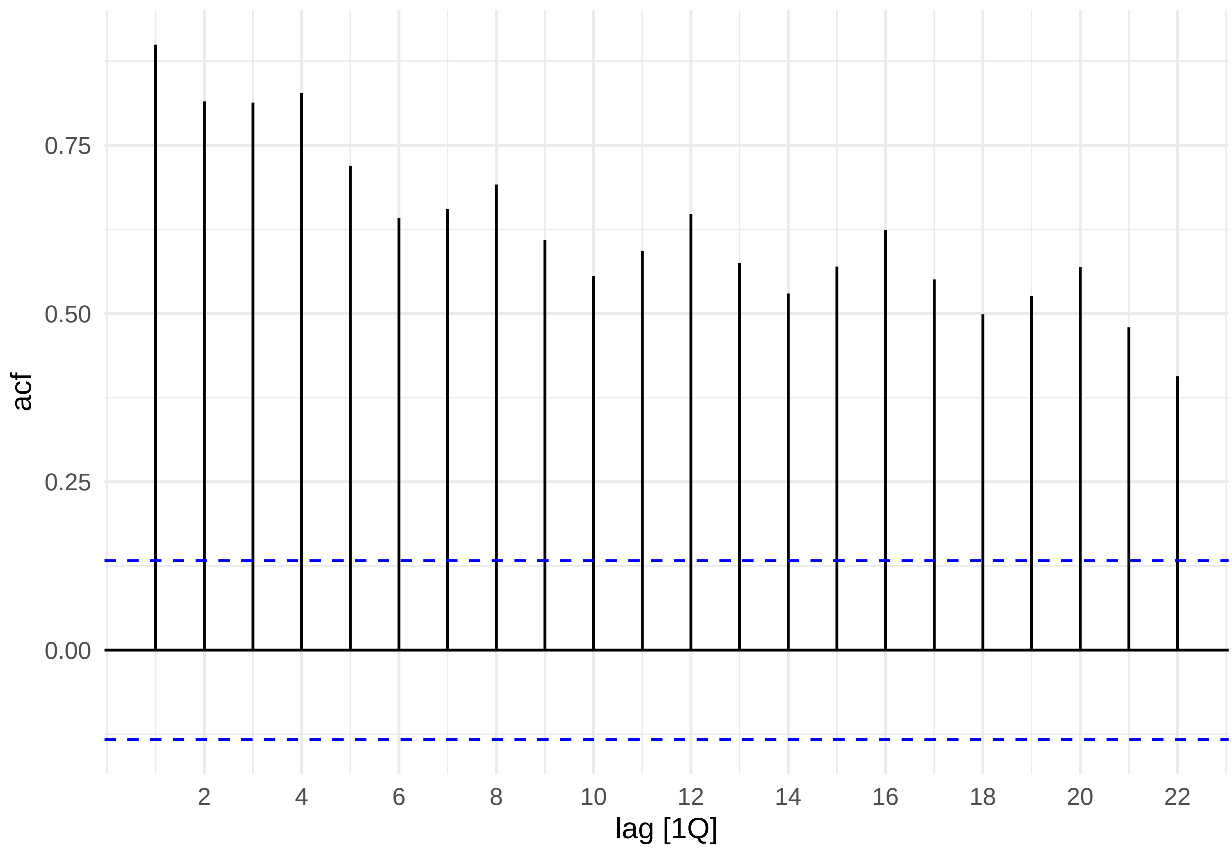

600
500
400
300
200
Bricks
1960 Q1
1980 Q1
2000 Q1
Quarter [1Q]


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

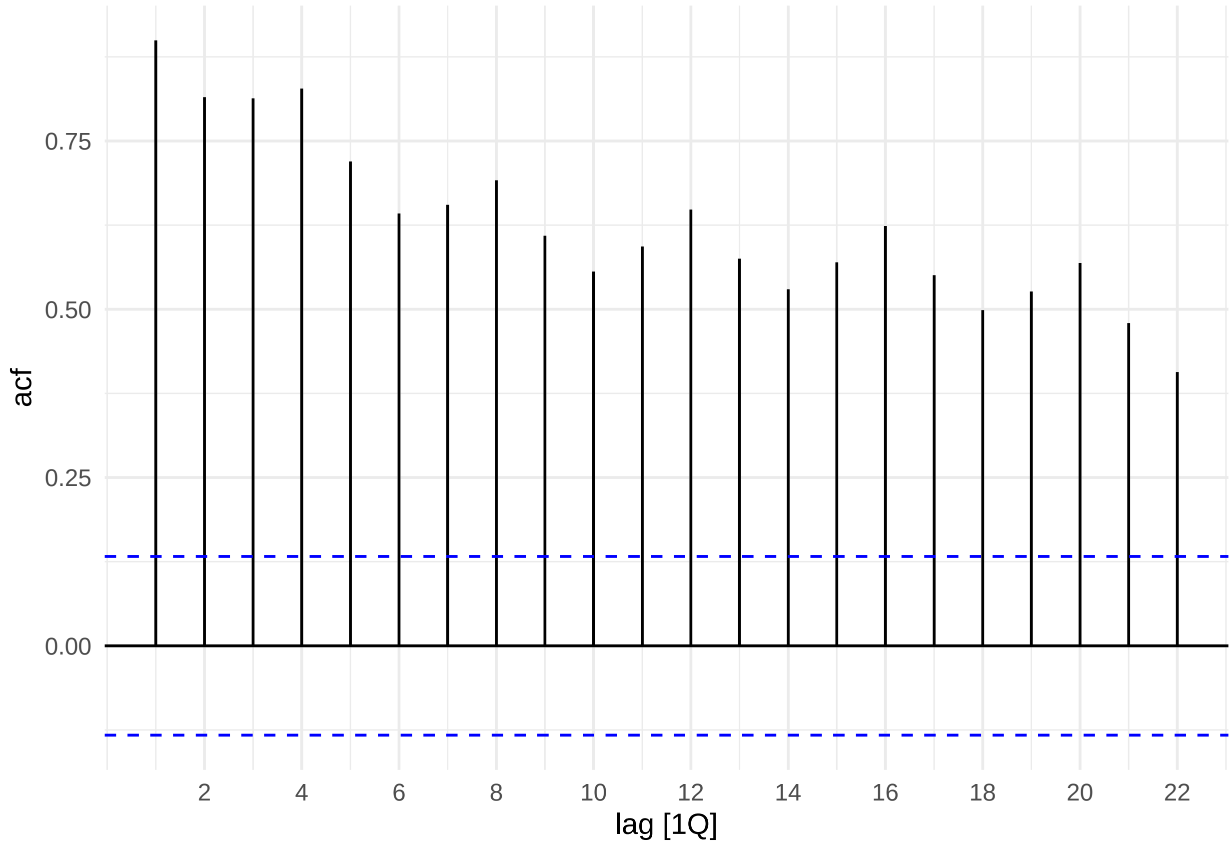

Q1
Q2
Q3
Q4
600
500
400
300
200
Bricks
1960
1980
2000
Quarter



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



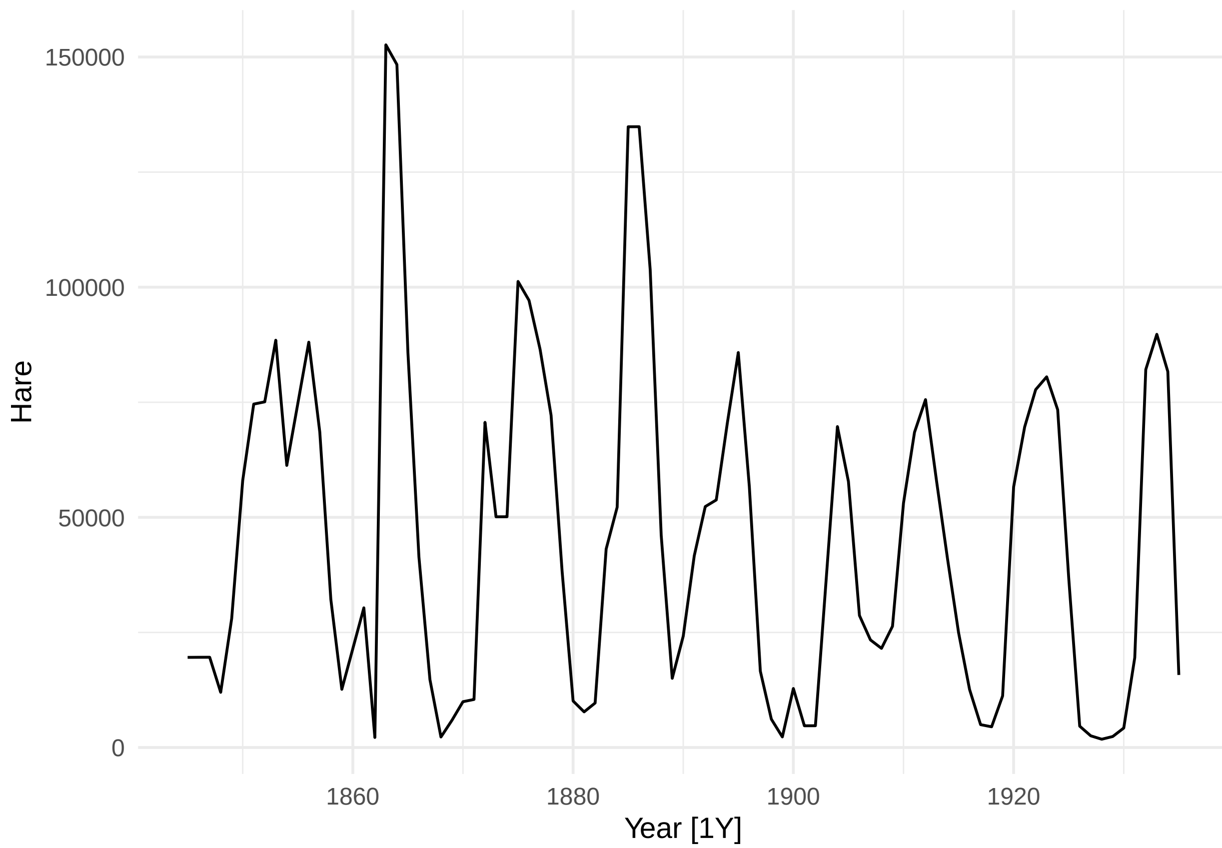
acf
0.75
0.50
0.25
0.00
2
4
6
8
10
12
14
16
18
20
22
lag [1Q]

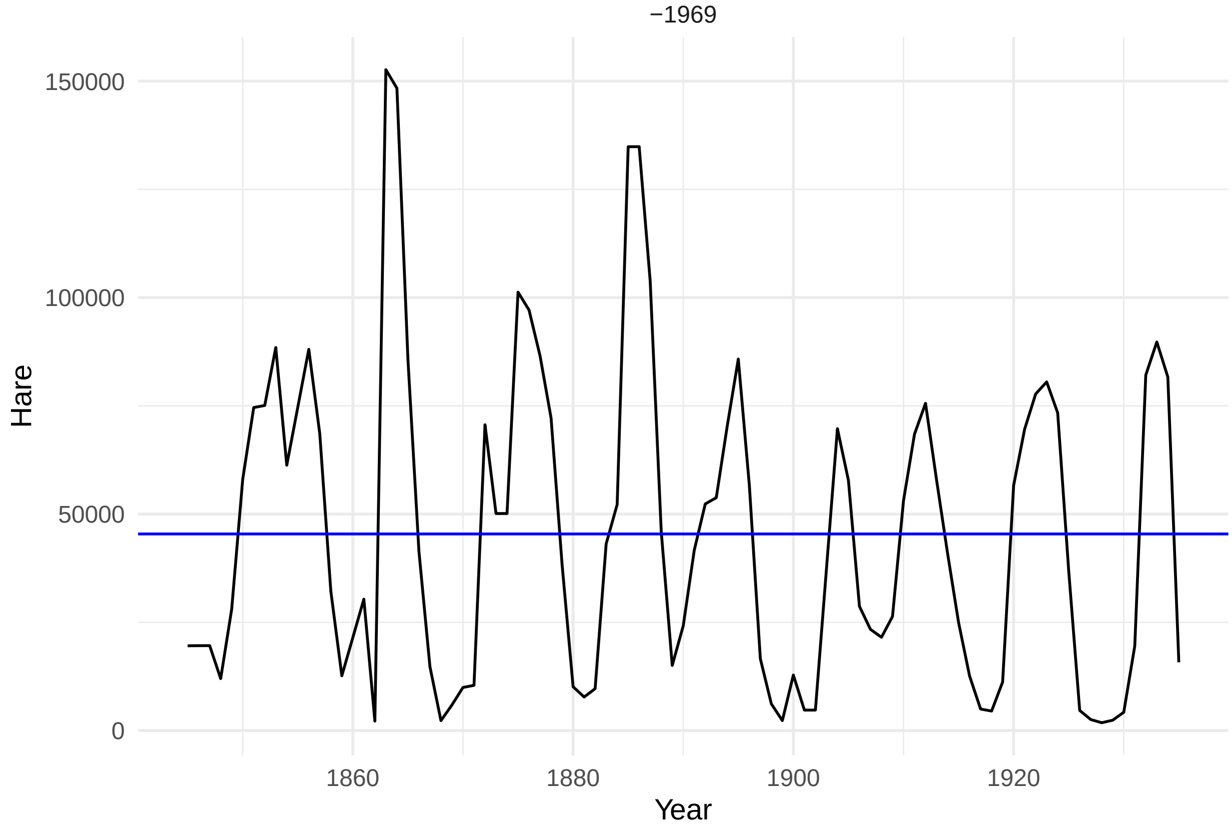


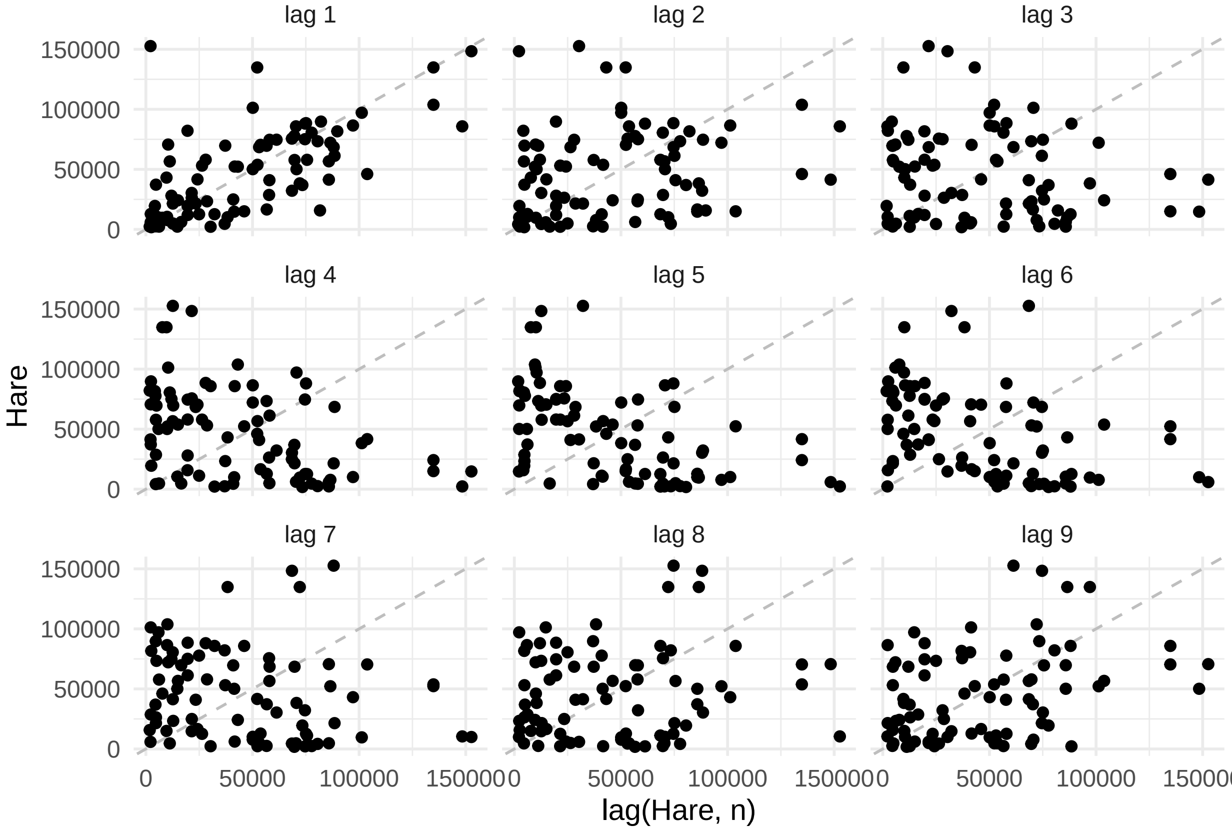
pelt This timeseries demonstrates neither an increasing nor decreasing trend but rather a cyclic pattern. Now, taking a look the lag plots there is no obvious correlation from prior time intervals. Moreover, the ACF() plots supports the observation that this time series exhibits cyclical behavior.

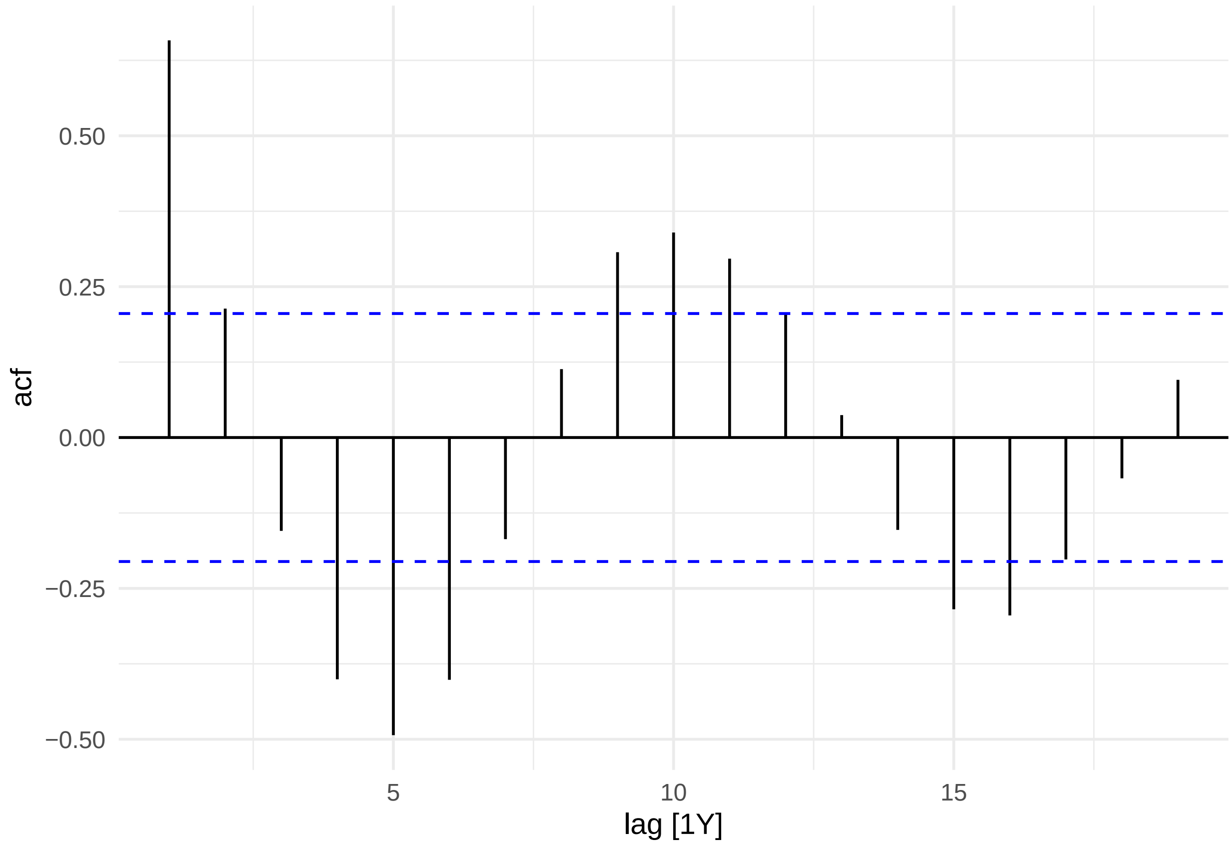
```
hare <- pelt |> select(-Lynx) |> as_tsibble(index = Year)
```

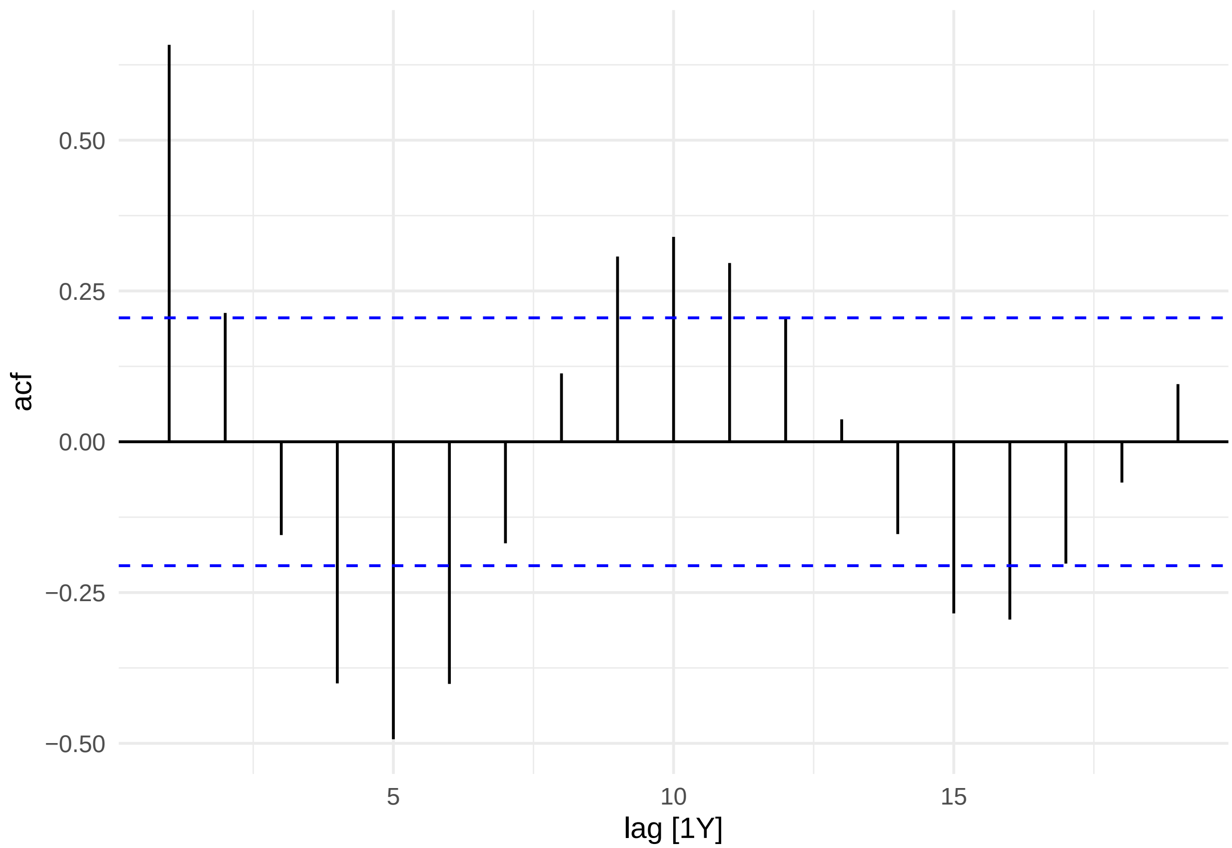
```
print(plot_time_series_without_gg_season(hare))
```


150000
100000
50000
0
Hare
1860
1880
1900
1920
Year [1Y]


−1969
Hare
150000
100000
50000
0
1860
1880
1900
1920
Year


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


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

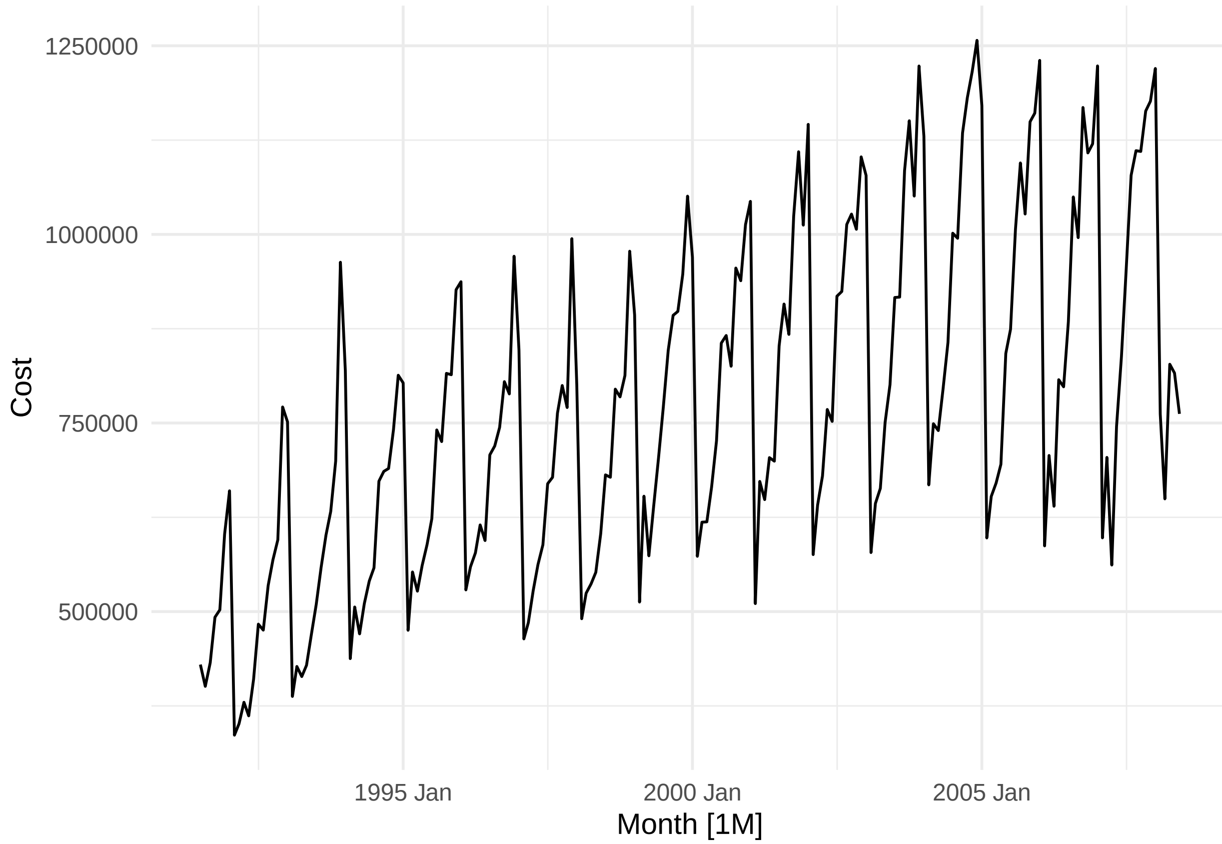


gg_season() doesnt work on this particular time series since a season suggest sub-yearly data.

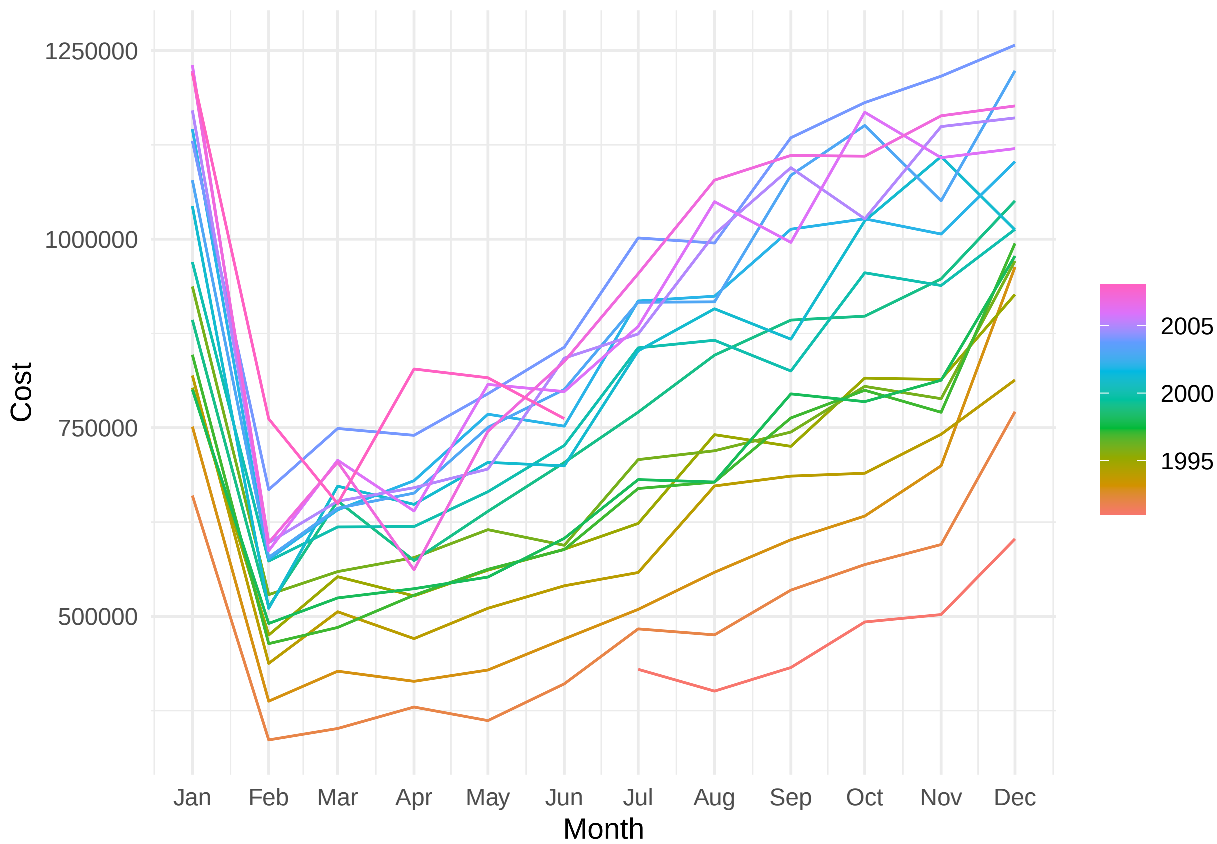
PBS PBS time series exhibits a somewhat increasing trend along with a seasonal patterns where cost is usually at its peak during December. From this time series, we learned that Australian presciption cost is at its lowest in the month of February, repeating for each consecutive years.

```
h02 <- PBS |>
  filter(ATC2 == "H02") |>
  as_tibble() |>
  select(Month, Cost) |>
  group_by(Month) |>
  summarize(Cost = sum(Cost)) |>
  as_tsibble(index = Month)
```

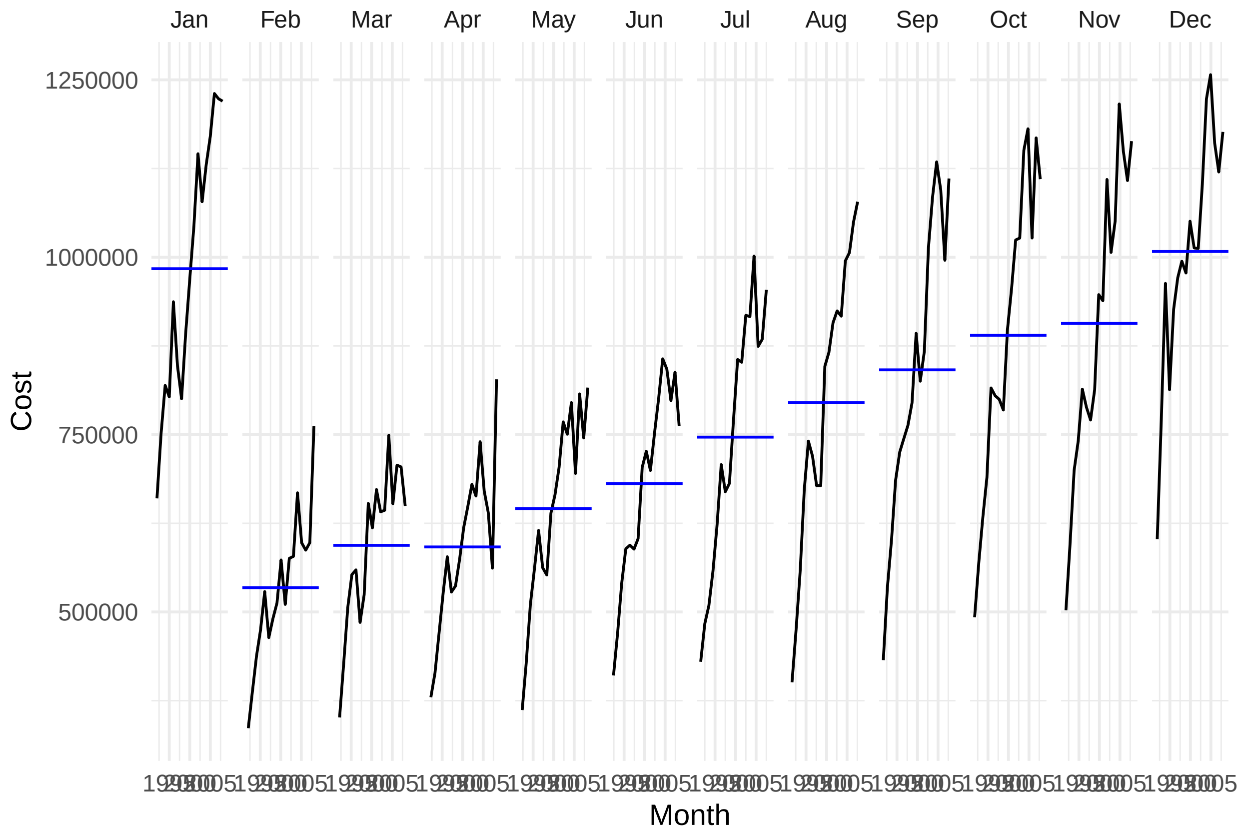
```
print(plot_time_series(h02))
```



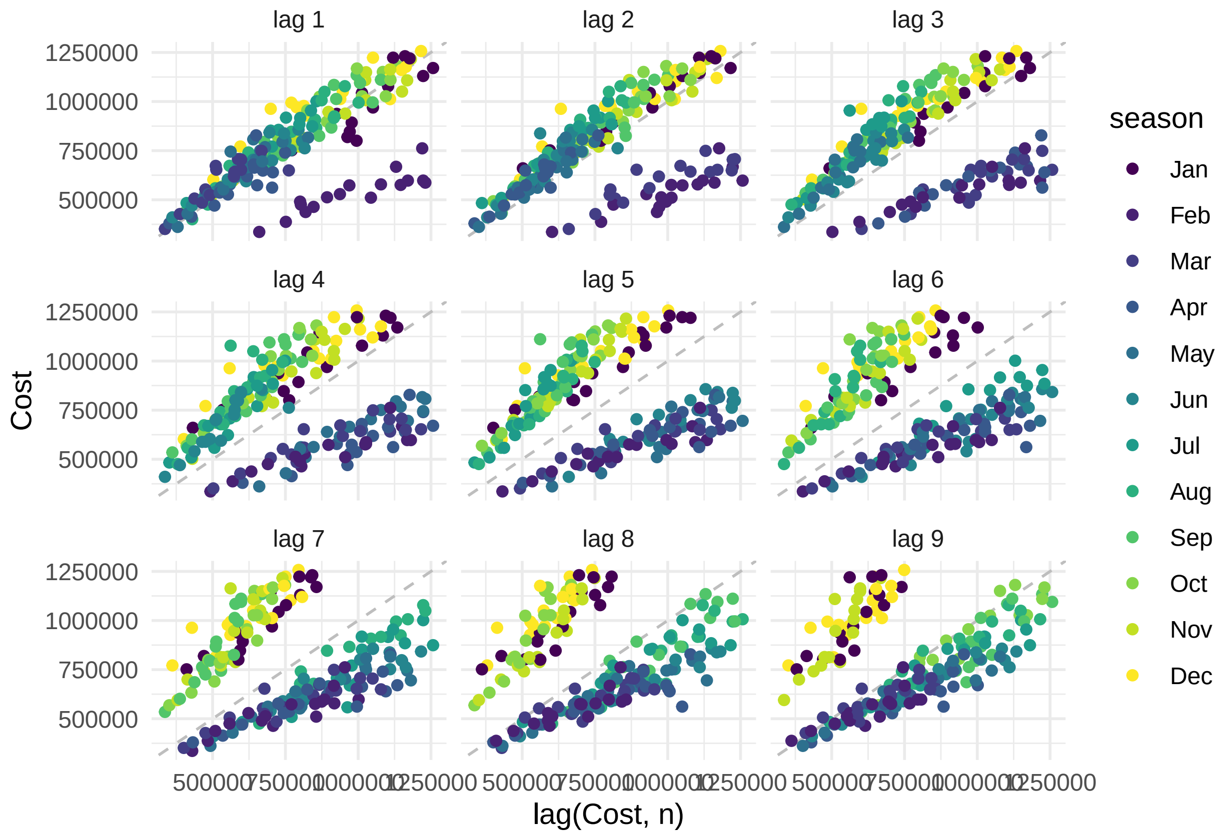
1250000
1000000
750000
500000
Cost
1995 Jan
2000 Jan
2005 Jan
Month [1M]



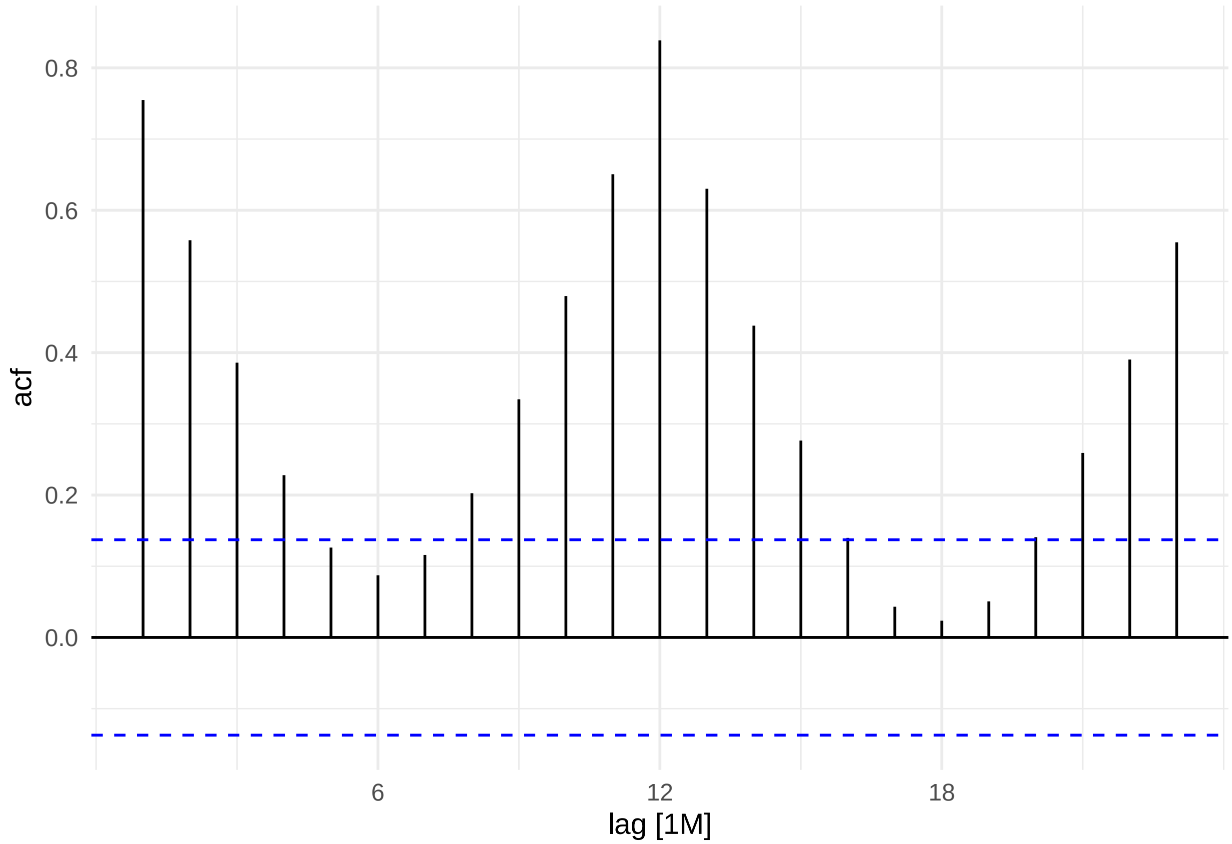
1250000
1000000
750000
500000
Cost
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
2005
2000
1995



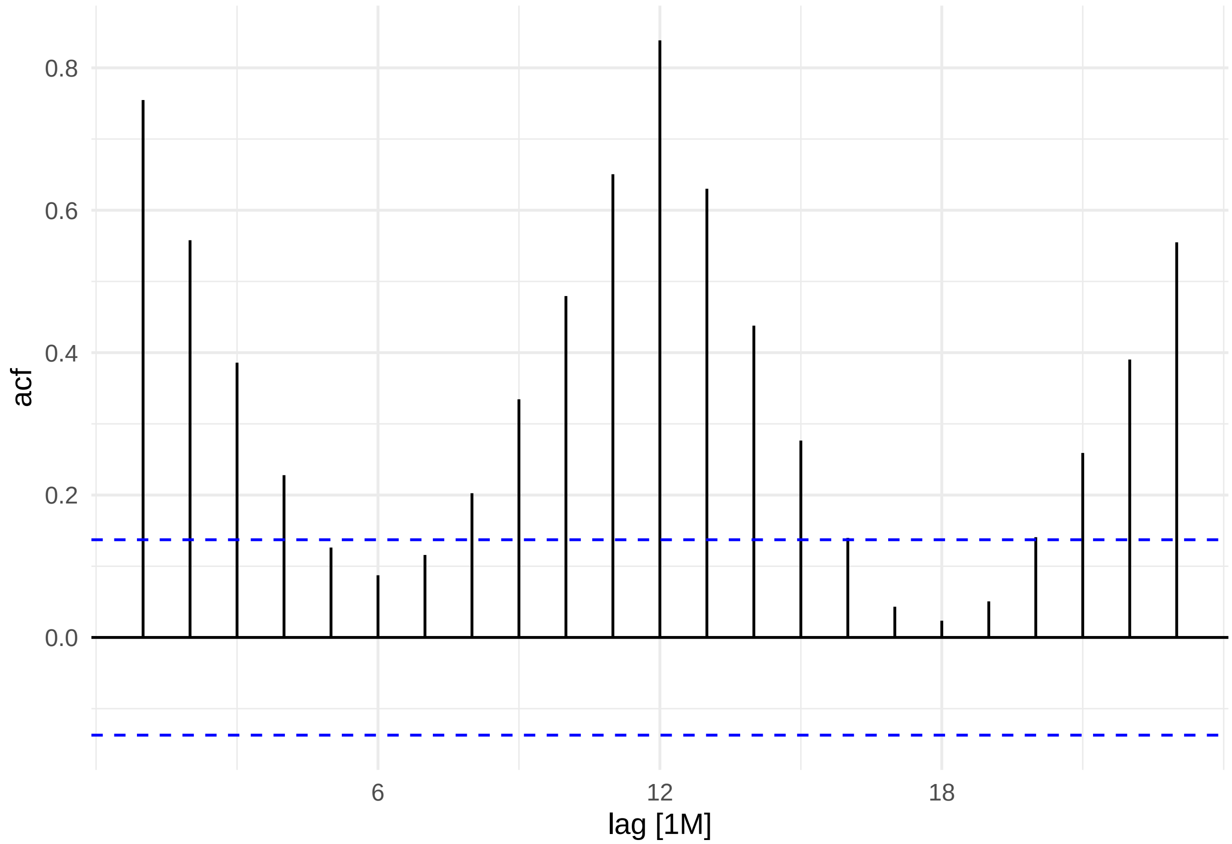
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1250000
1000000
750000
500000
Cost
Month



lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
1250000
1000000
750000
500000
Cost
500000
750000
1000000
1250000
lag(Cost, n)
season
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

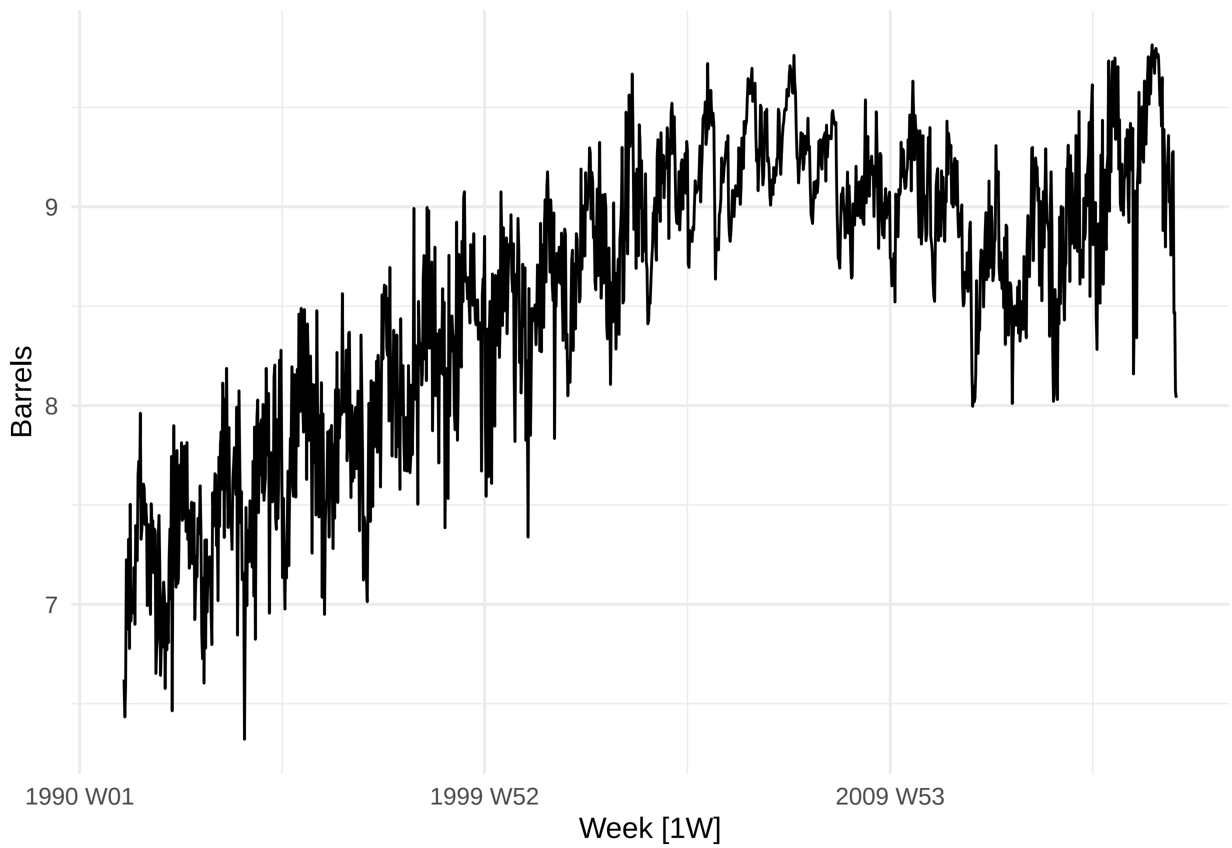


acf
0.8
0.6
0.4
0.2
0.0
6
12
18
lag [1M]

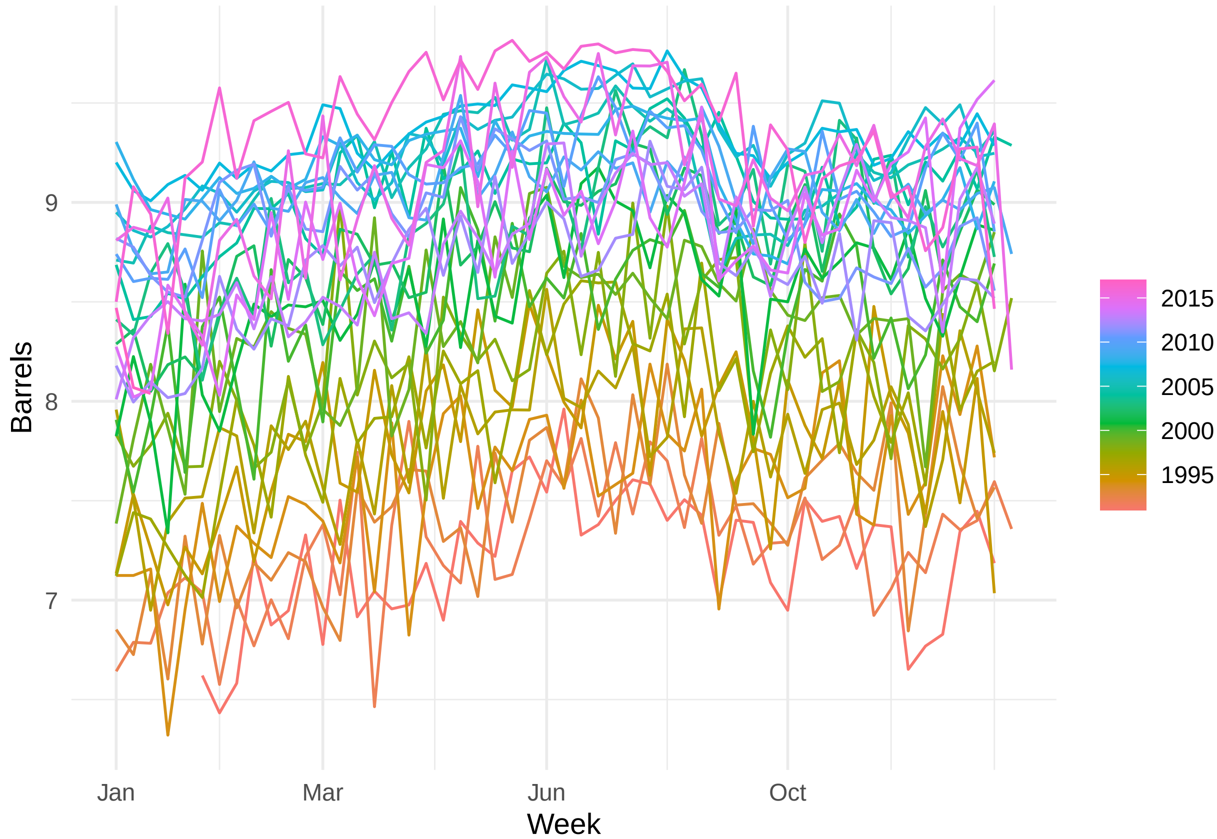


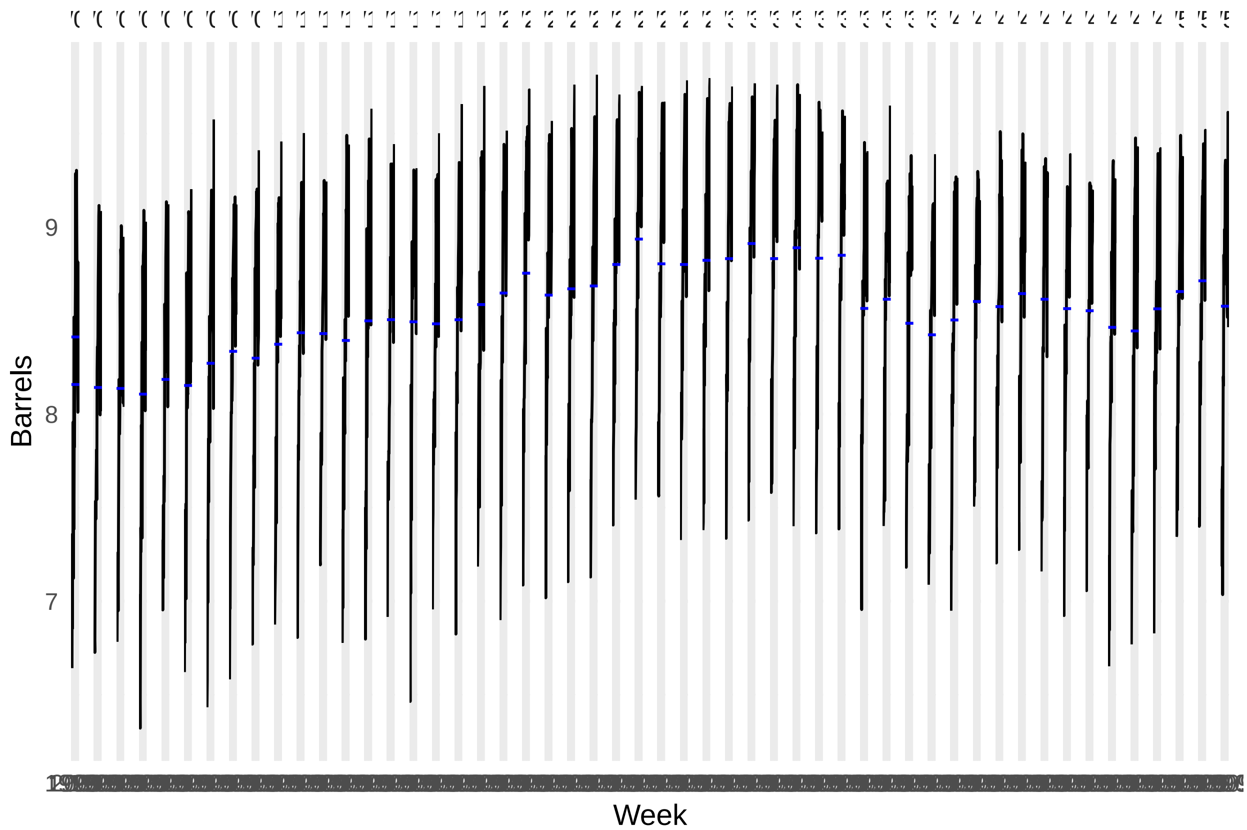
us_gasoline Us_gasoline time series increasing trend with a seasonal pattern. When inspecting the lag plot, all 52 weeks are closely grouped together along the positive correlation line indicating that us gas price generally increase every year. Then, this is supported by AFC() plot where the values are all above 0.50 correlation.

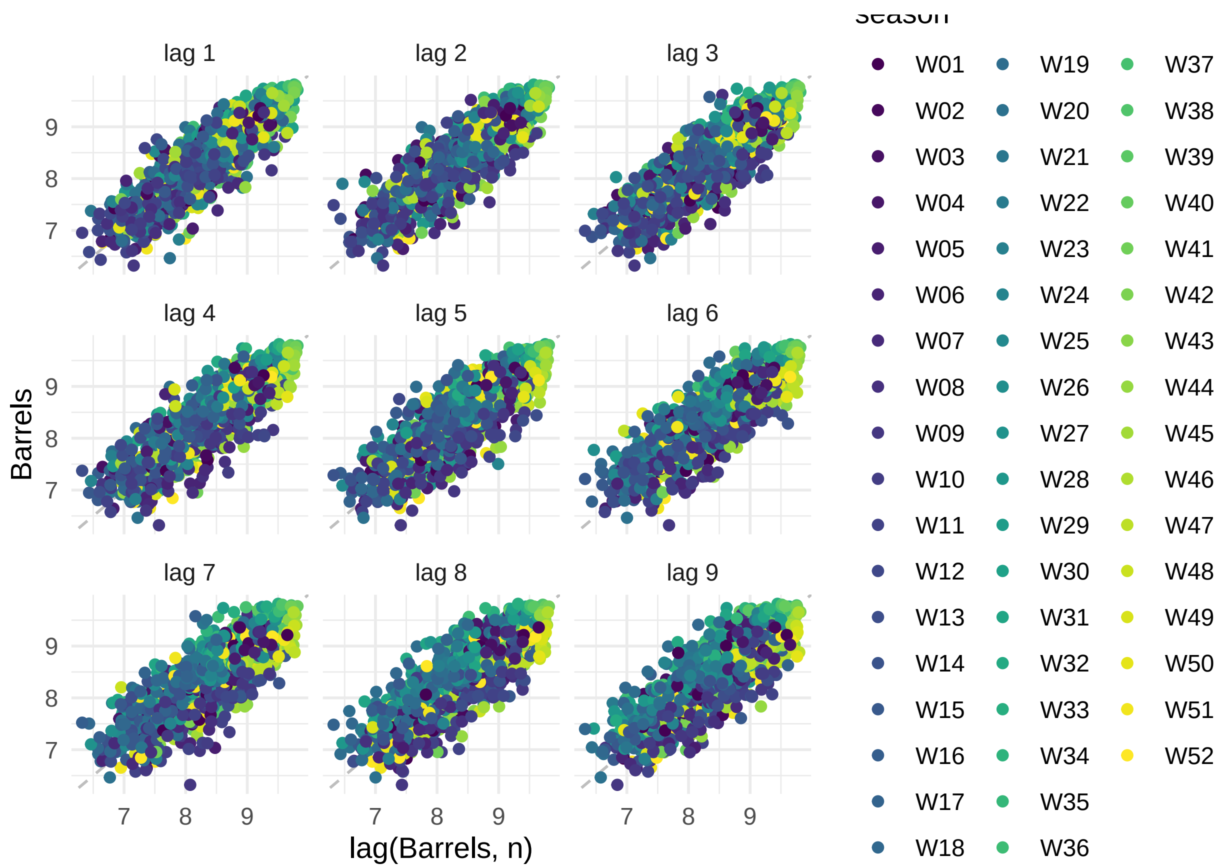
```
print(plot_time_series(us_gasoline))
```

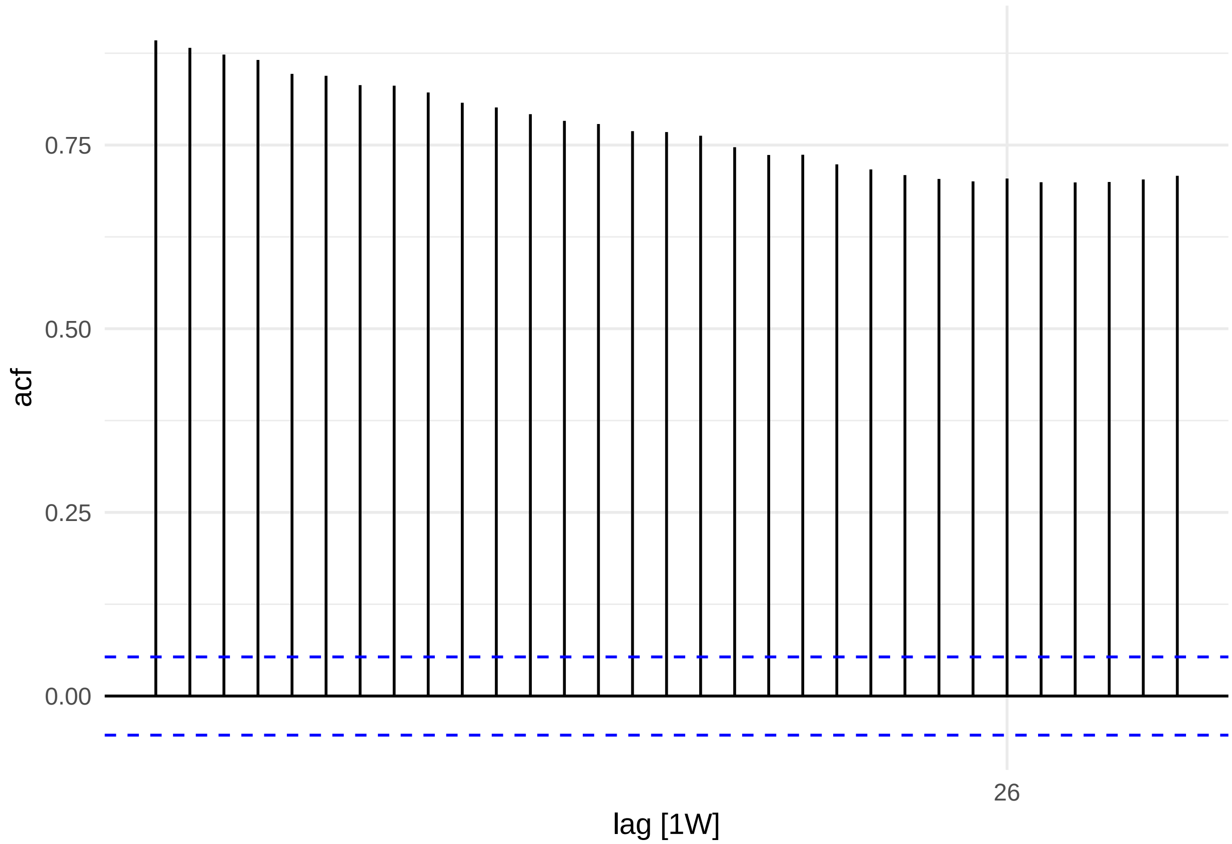


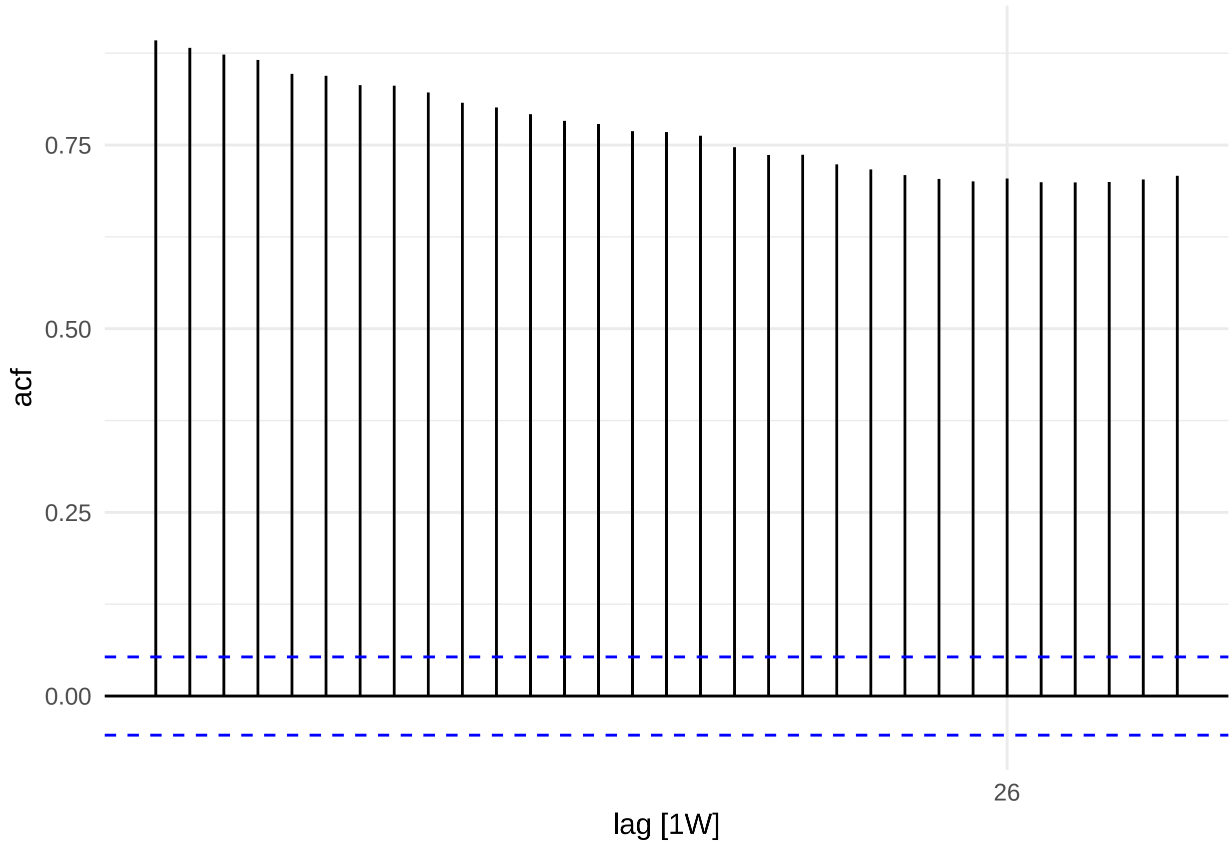
Barrels
9
8
7
1990 W01
1999 W52
2009 W53
Week [1W]


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


Barrels
Week
9
8
7


season
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9
Barrels
9
8
7
lag(Barrels, n)
7 8 9
W01 W02 W03 W04 W05 W06 W07 W08 W09 W10 W11 W12 W13 W14 W15 W16 W17 W18
W19 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 W30 W31 W32 W33 W34 W35 W36
W37 W38 W39 W40 W41 W42 W43 W44 W45 W46 W47 W48 W49 W50 W51 W52


0.75
0.50
0.25
0.00
acf
26
lag [1W]



0.75
0.50
0.25
0.00
acf
26
lag [1W]