624 Week 1

Chi Hang(Philip) Cheung

2025-02-01

```
library(fpp3)
```

```
## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr

## -- Attaching packages --------------------------------------- fpp3 1.0.1 --

## v tibble      3.2.1     v tsibble     1.1.6
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.1     v feasts      0.4.1
## v lubridate   1.9.4     v fable       0.4.1
## v ggplot2     3.5.1

## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(dplyr)
```

Exercise 2.1

- What is the time interval of each series?

Ans: Bricks dataset is from 1956 to 2010

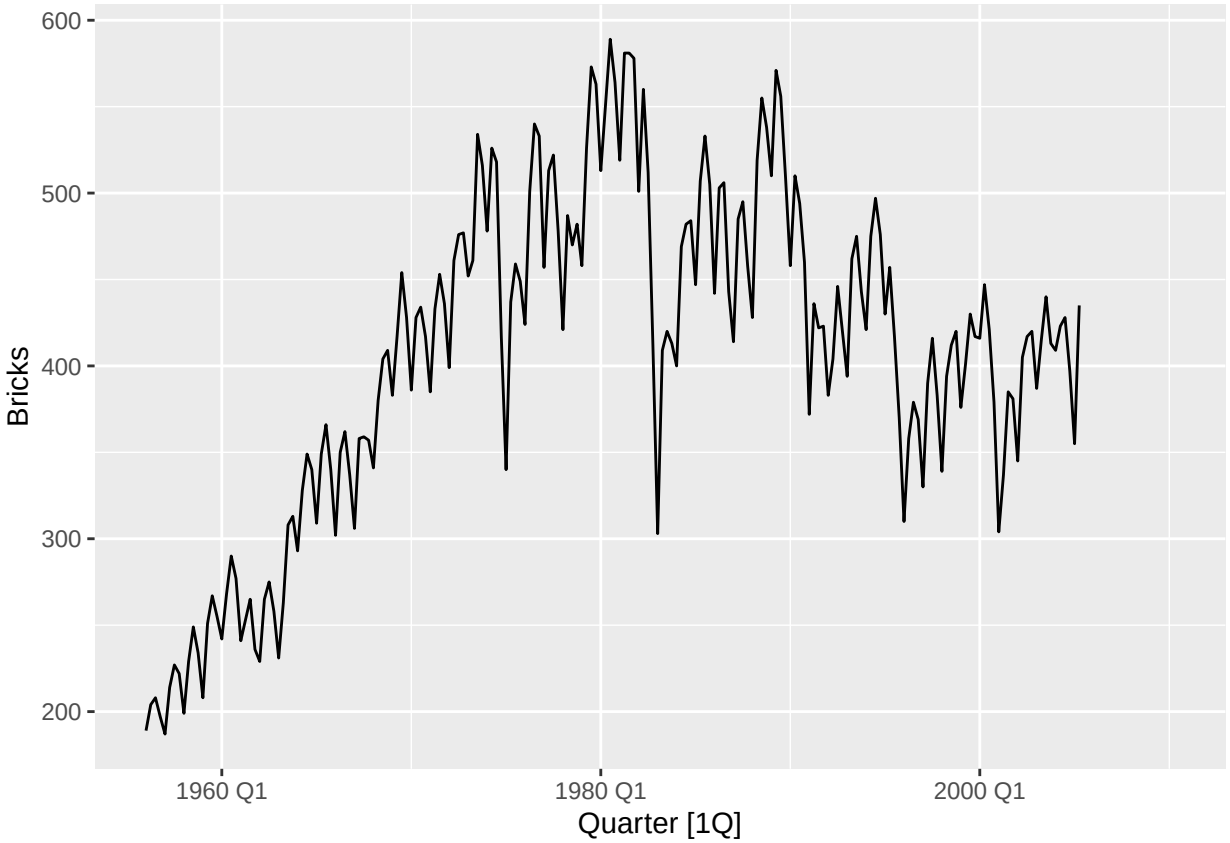
- Use `autoplot()` to produce a time plot of each series.

Brick production shows an upward trend until 1980 Q1 then the production started to trend downward.

```
bricks<- aus_production %>%
  select(Quarter, Bricks)

autoplot(bricks)
```

```
## Plot variable not specified, automatically selected `.vars = Bricks`
```

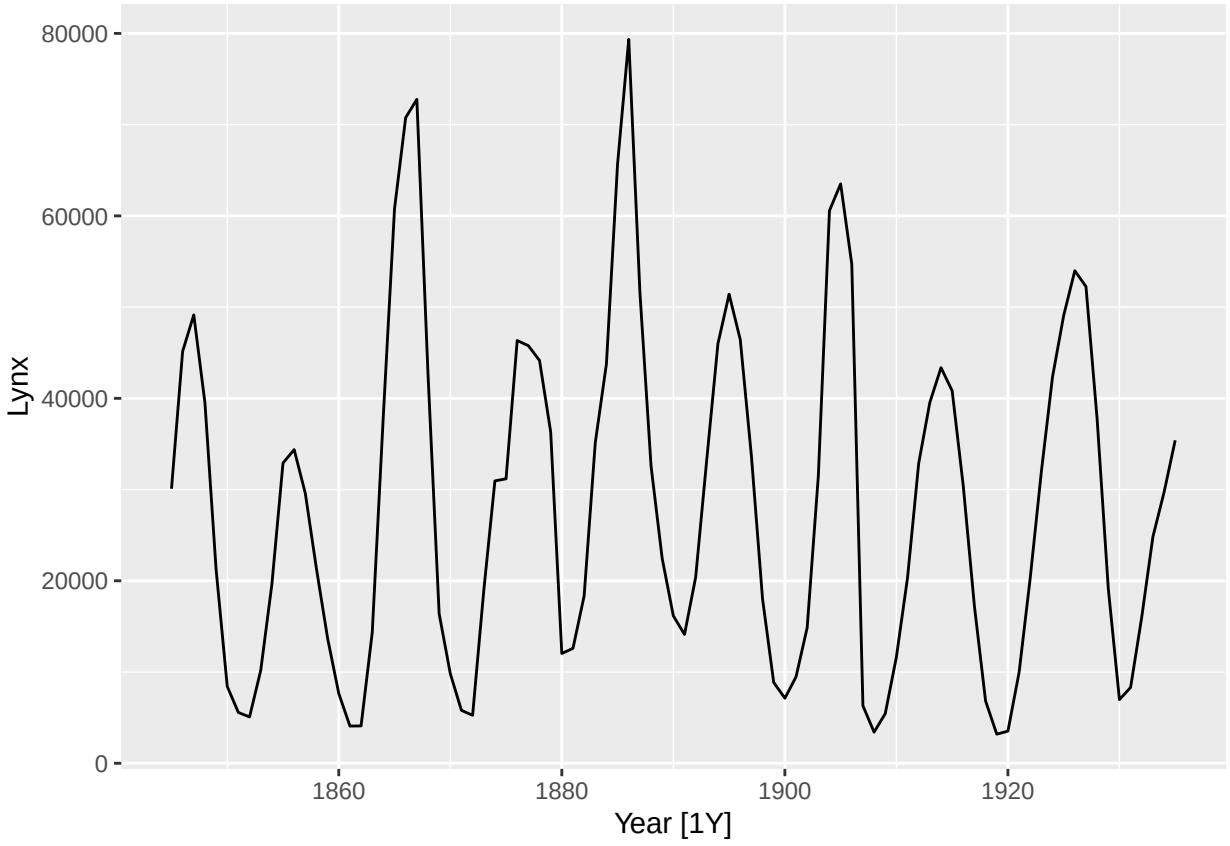
```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Pelt: from 1845 to 1935 Lynx pelt trading record has a seasonal pattern.

```
lynxx<- pelt %>%
  select(Year, Lynx)

autoplot(lynxx)
```

```
## Plot variable not specified, automatically selected `.vars = Lynx`
```



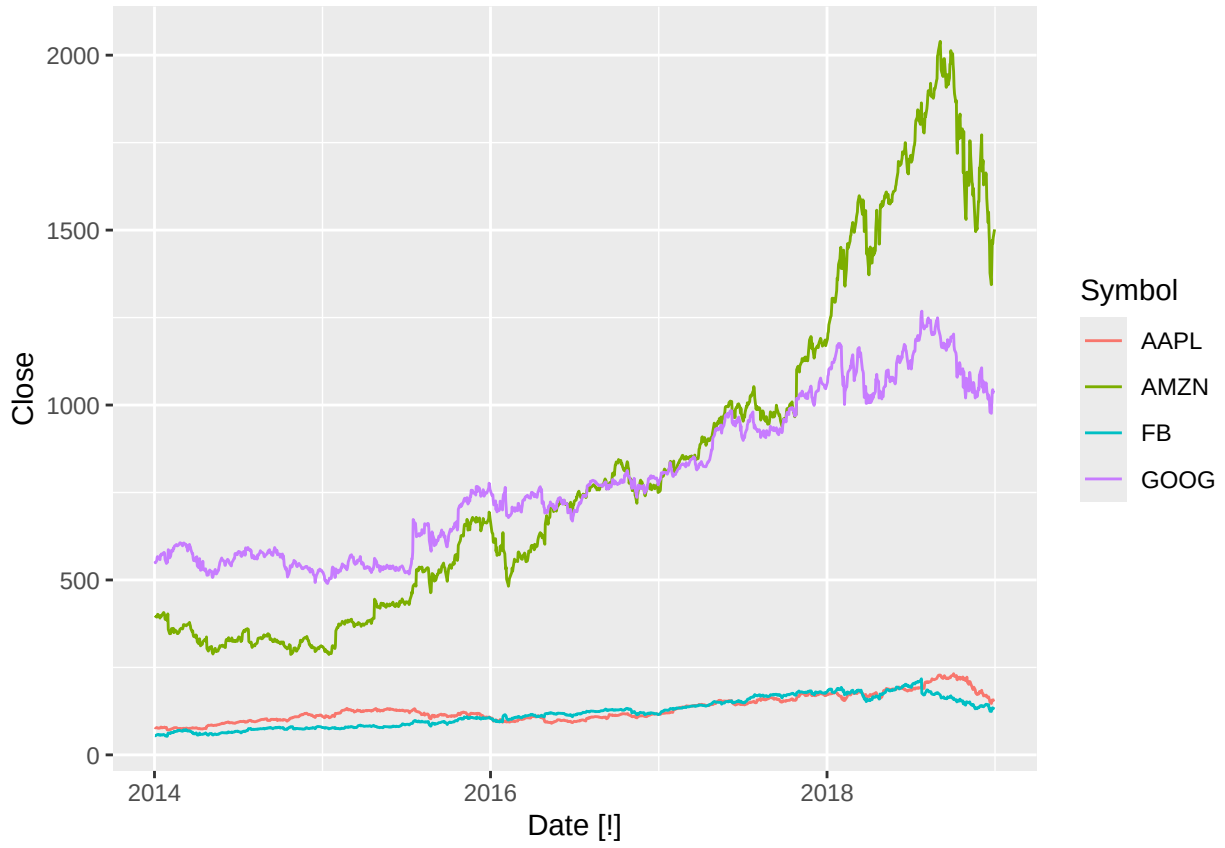
GAFA Stock

Period: From 2014 to 2018

Amazon and Google both have an upward trend. Facebook and AAPL are relatively stable.

```
close<- gafa_stock %>%
  select(Date, Symbol, Close)
autoplot(close)
```

```
## Plot variable not specified, automatically selected `.vars = Close`
```



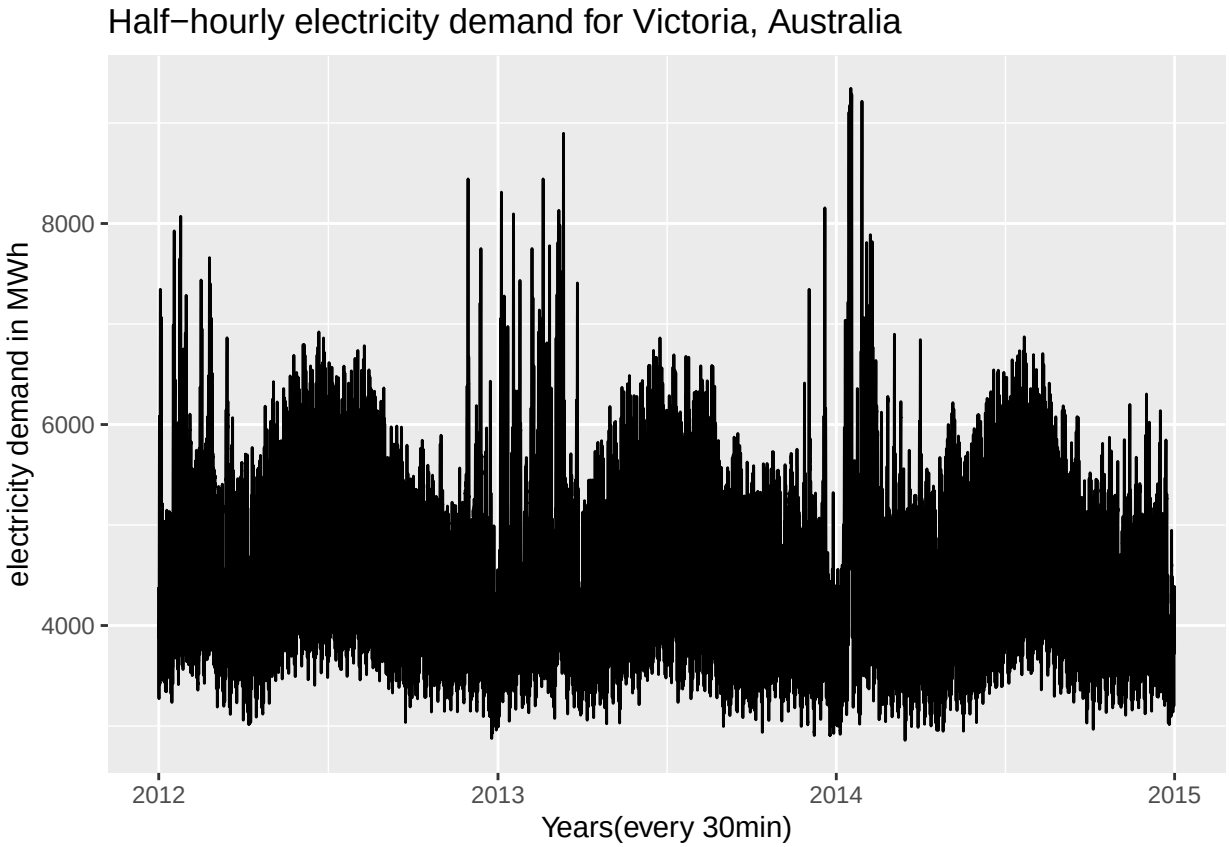
vic_elec From 2012 to 2015 This shows a seasonal pattern

```
vic_elec
```

```
## # 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
```

```
demand<- vic_elec %>%
  select(Date, Demand)
autoplot(demand) + labs(y='electricity demand in MWh',
                        x='Years(every 30min)',
                        title = "Half-hourly electricity demand for Victoria, Australia")
```

```
## Plot variable not specified, automatically selected `.vars = Demand`
```



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

```
gafa_stock %>%
  group_by(Symbol) %>%
  filter(Close == max(Close)) %>%
  select(Symbol, Date, 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.
```

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

```
url<- 'https://raw.githubusercontent.com/stormwhale/data-mines/refs/heads/main/tute1%20(2).csv'
tute1<- readr::read_csv(url)
```

```
## 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.
```

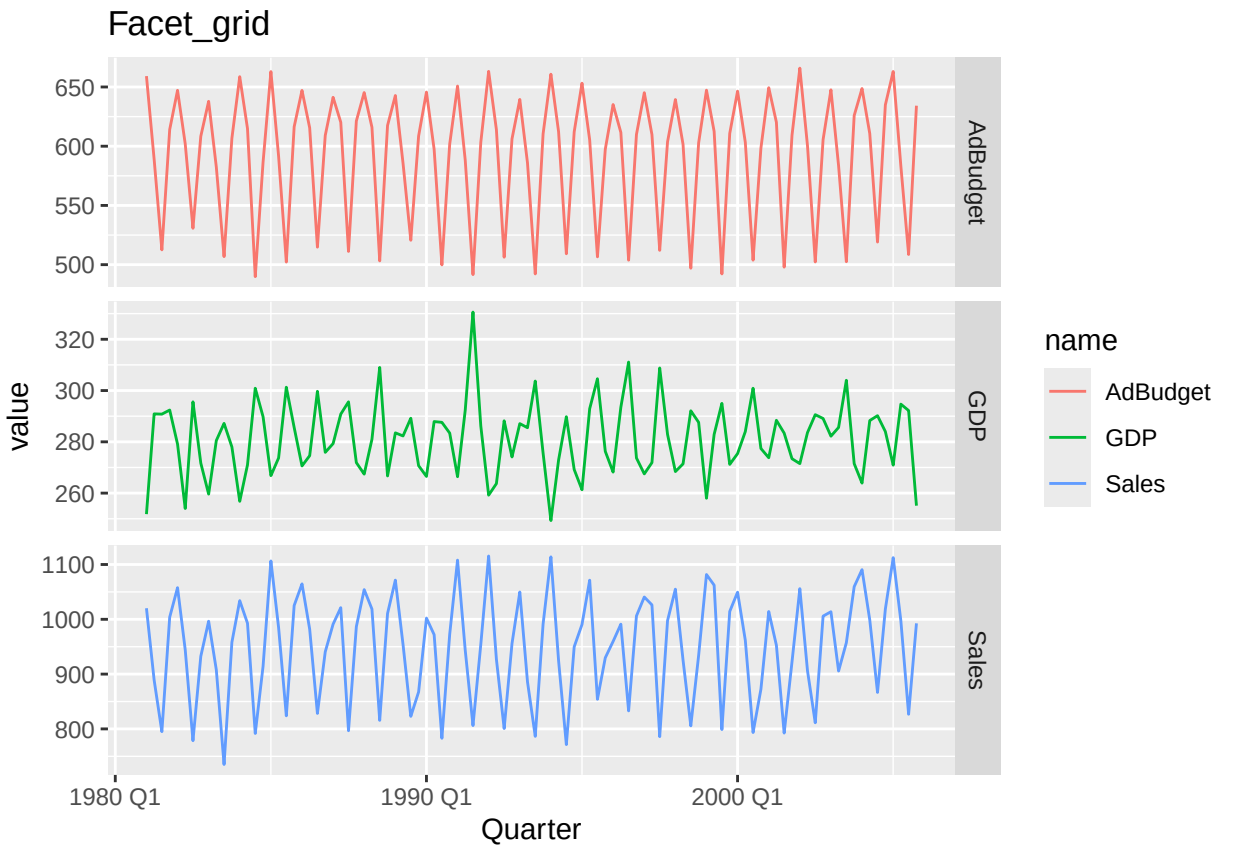
```
view(tute1)
```

Convert the data to time series

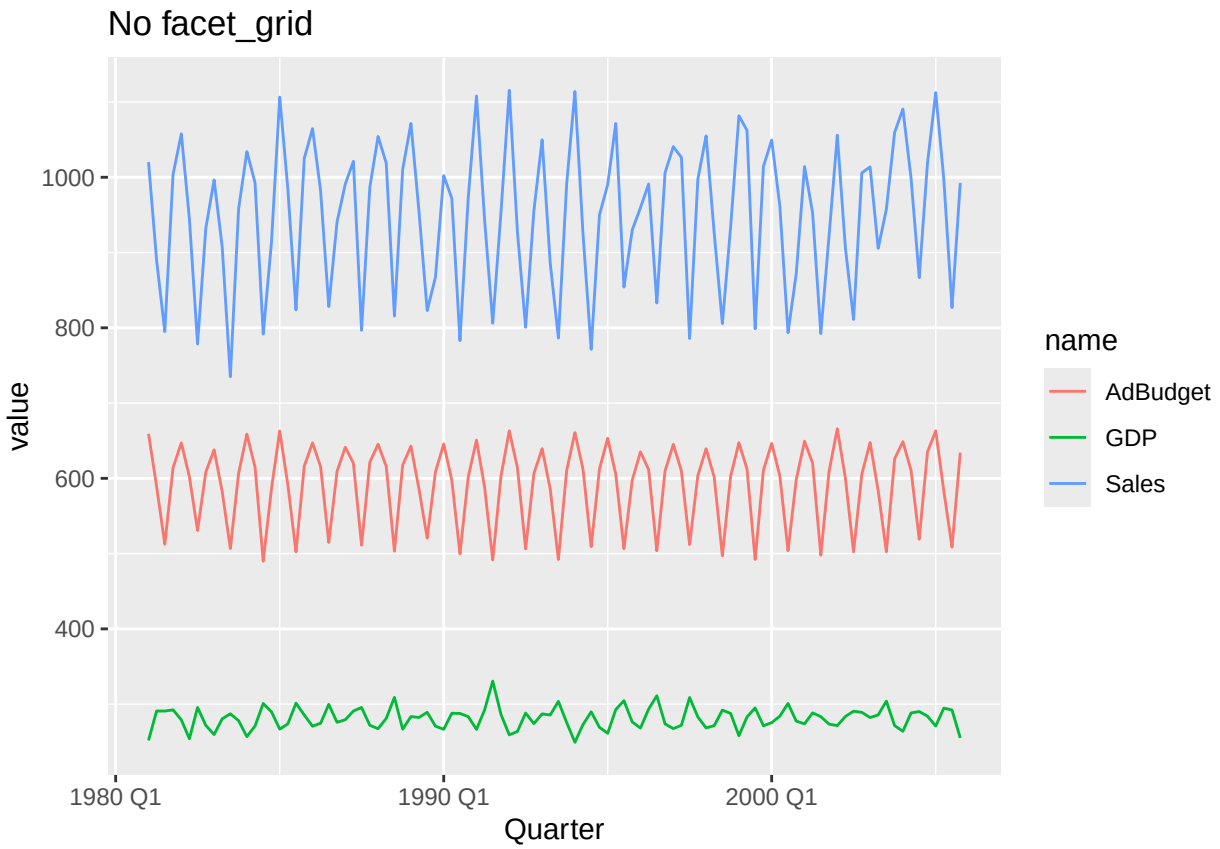
```
mytimeseries<- tute1 %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(index=Quarter)
```

Construct time series plots of each of the three series

```
mytimeseries %>%
  pivot_longer(-Quarter) %>%
  ggplot(aes(x=Quarter, y=value, color = name))+
  geom_line() +
  facet_grid(name ~ .,scales ='free_y')+
  labs(title='Facet_grid')
```




```
mytimeseries %>%
  pivot_longer(-Quarter) %>%
  ggplot(aes(x=Quarter, y=value, color = name))+
  geom_line() +
  labs(title = 'No facet_grid')
```



Question: Check what happens when you don't include `facet_grid()`?

Ans: Without the facet_grid, three plots are combined together in a single graph with the same y-axis.

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

a) Install the USgas package.

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

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

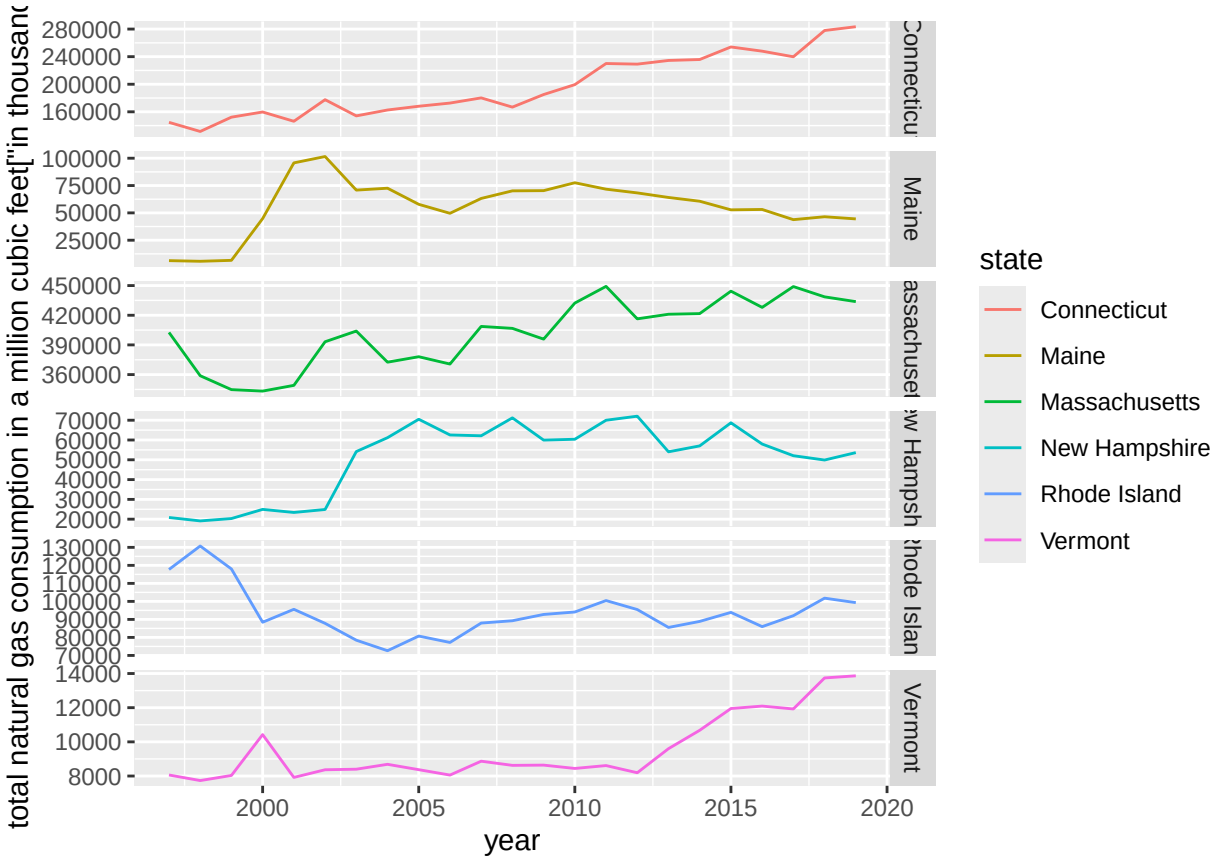
```
us_total <- USgas::us_total %>%
  as_tsibble(key = state,
             index = year)
head(us_total)
```

```
## # 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
```

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).

```
us_total %>%
  filter(state %in% c('Maine', 'Vermont', 'New Hampshire', 'Massachusetts', 'Connecticut', 'Rhode Islan
  ggplot(aes(x=year, y=y, color=state)) +
  geom_line()+
  facet_grid(state ~., scale='free_y')+
  labs(y='total natural gas consumption in a million cubic feet["in thousand"]')
```



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

```
url2<- 'C:\\Users\\godly\\Downloads\\tourism.xlsx'
tour2<- readxl::read_excel(url2)
tour2
```

```
## # A tibble: 24,320 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.
##  7 1999-07-01 Adelaide South Australia Business  169.
##  8 1999-10-01 Adelaide South Australia Business  134.
##  9 2000-01-01 Adelaide South Australia Business  154.
## 10 2000-04-01 Adelaide South Australia Business  169.
## # i 24,310 more rows
```

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

```
tourr2<- tour2 %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(index=Quarter, key=c('Region', 'State','Purpose','Trips'))
```

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

Ans: Sydney and visiting are the combination that has the max number of trips on average.

```
tour2 %>%
  group_by(Region, Purpose) %>%
  summarise(avg_trip = mean(Trips)) %>%
  slice_max(avg_trip, n=1) %>%
  arrange(desc(avg_trip))
```

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

```
## # A tibble: 76 x 3
## # Groups:   Region [76]
##    Region                 Purpose  avg_trip
##    <chr>                  <chr>       <dbl>
##  1 Sydney                 Visiting     747.
##  2 Melbourne              Visiting     619.
##  3 North Coast NSW        Holiday      588.
##  4 Gold Coast             Holiday      528.
##  5 South Coast            Holiday      495.
##  6 Brisbane               Visiting     493.
##  7 Sunshine Coast         Holiday      436.
##  8 Hunter                 Holiday      319.
##  9 Australia's South West Holiday      309.
## 10 Experience Perth       Visiting     291.
## # i 66 more rows
```

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

```
tourr2 %>%
  group_by(State) %>%
  summarise(total_trip=sum(Trips)) %>%
  index_by(Quarter)
```

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
## # Groups:    @ Quarter [80]
##    State Quarter total_trip
##    <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
```

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.

Data 1 US_employment:

Questions:

1) Can you spot any seasonality, cyclicity and trend?

   Ans: The US employment data has an obvious upward trend rather than any seasonality and cyclicity. The autoplot shows almost a straight upward straight line.

2) What do you learn about the series?

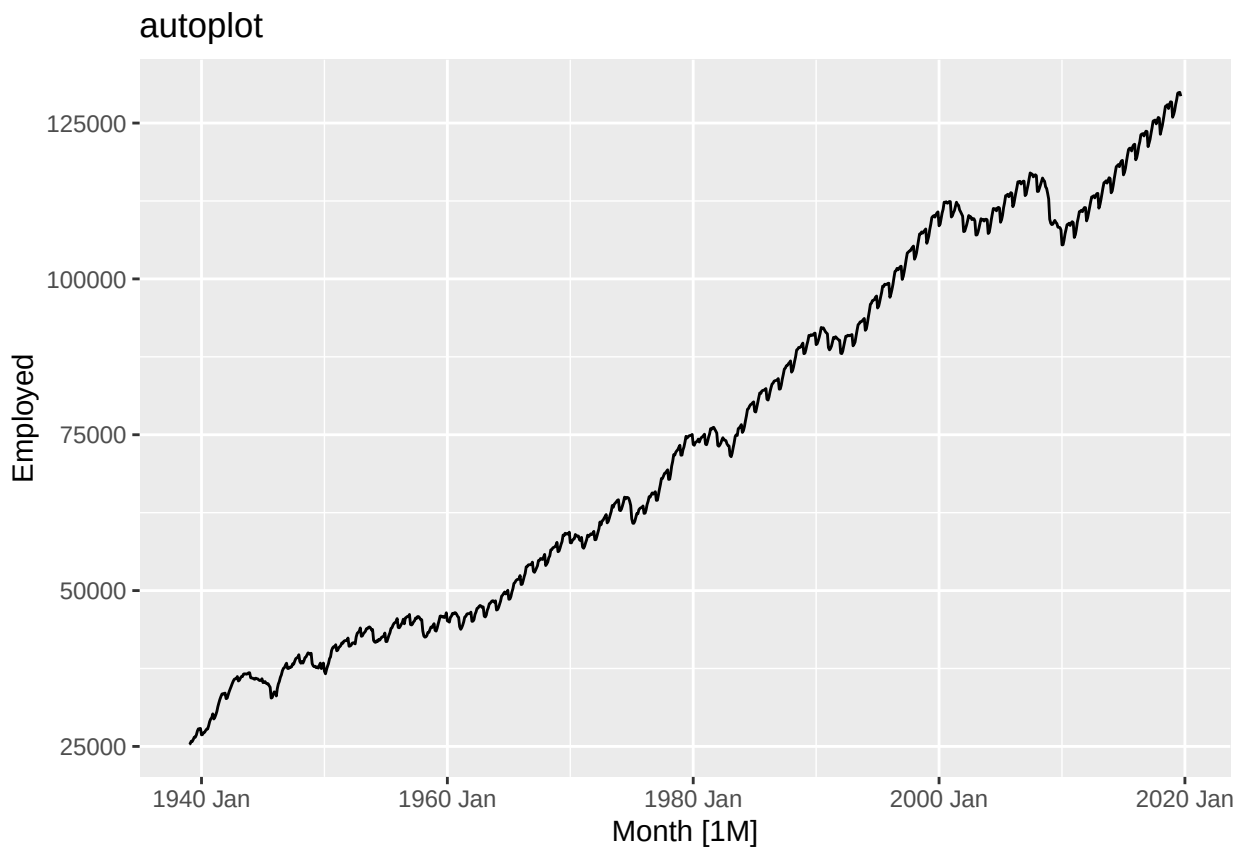
   Ans: The gg_subseries shows a very obvious pattern that throughout different years and acroos all the months, the employment rate is constantly going up.

3) What can you say about the seasonal patterns? Can you identify any unusual years?

   Ans: The only slight unusual year in around 2000, where the employment rate showed a slight drop for the while year. Besides that the trend is just upward.

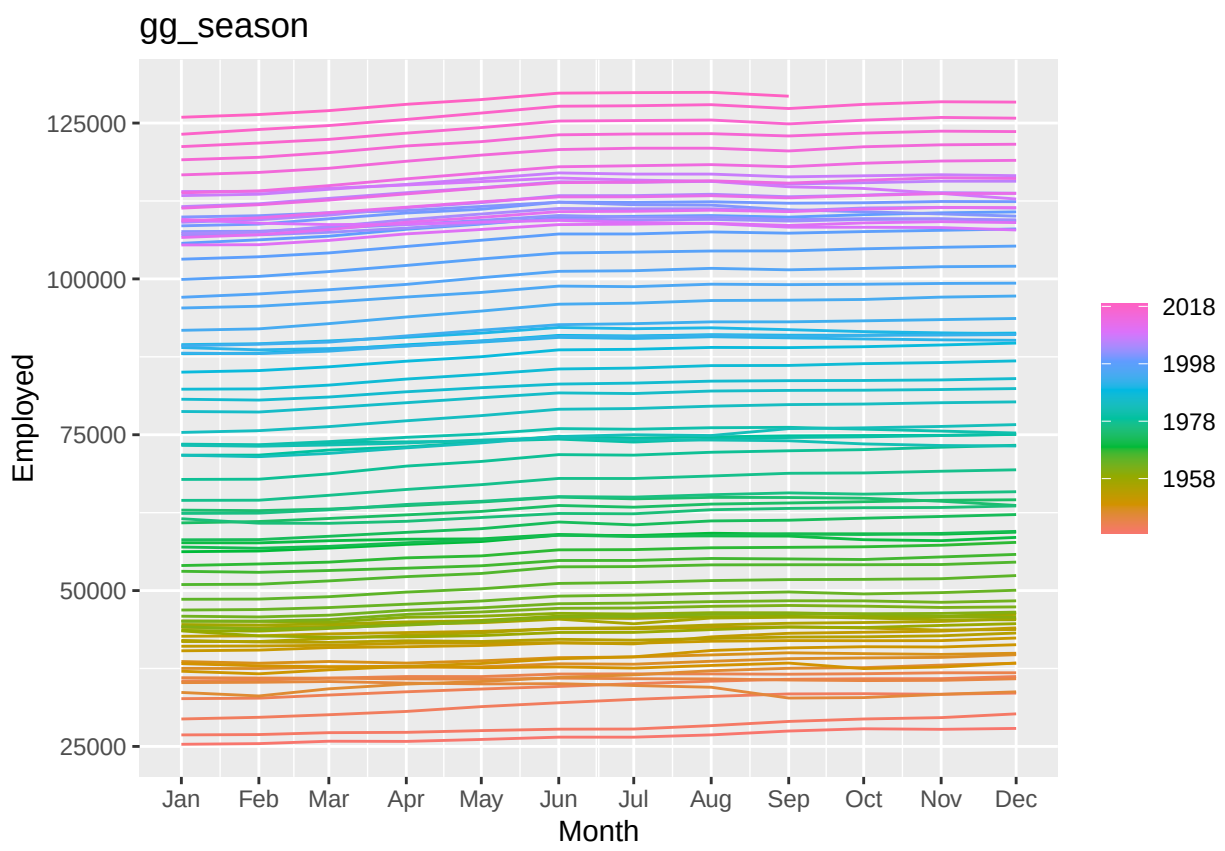
```
total_private<- us_employment %>%
  mutate(Month = yearmonth(Month)) %>%
  filter(Title=='Total Private') %>%
  as_tsibble(index=Month)

autoplot(total_private) +  labs(title = 'autoplot')
```

```
## Plot variable not specified, automatically selected `.vars = Employed`
```



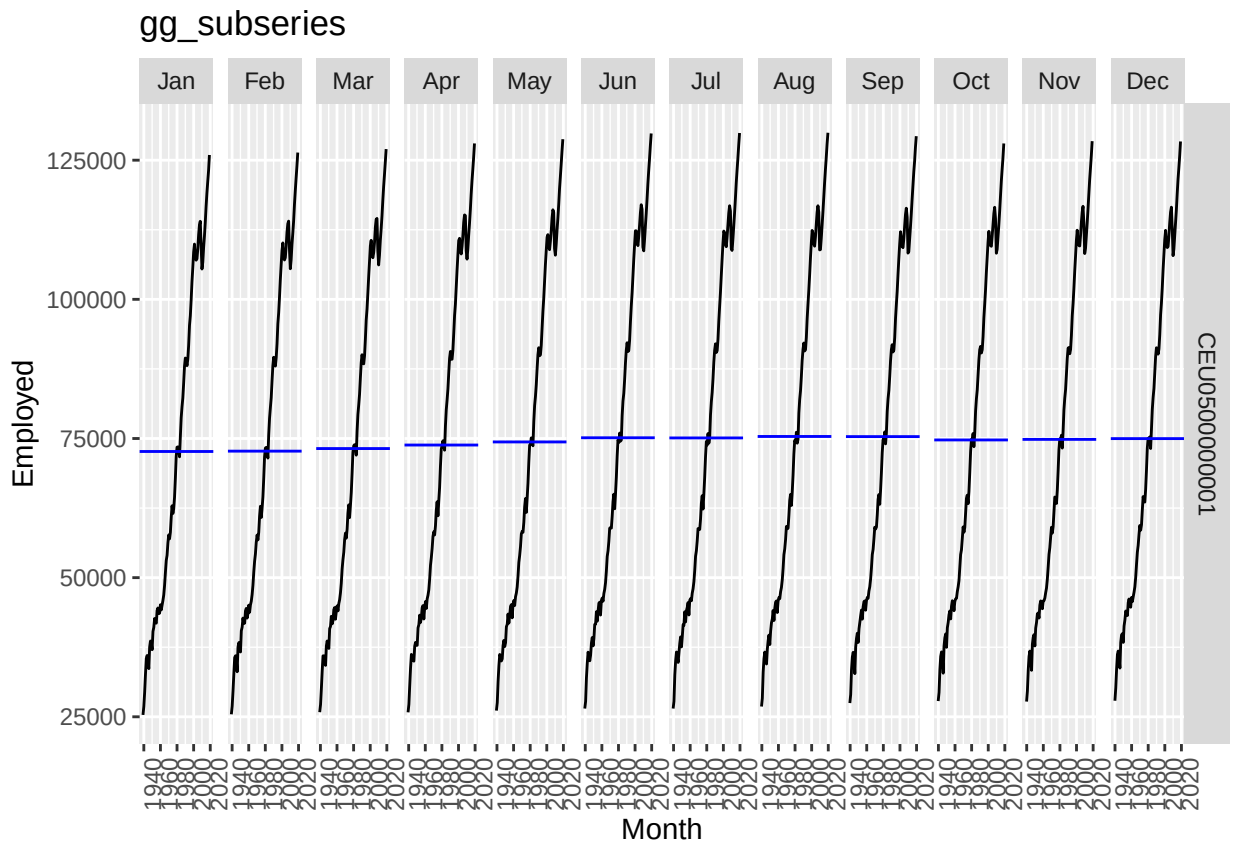

```
gg_season(total_private)+  labs(title = 'gg_season')
```

```
## Plot variable not specified, automatically selected ‘y = Employed‘
```



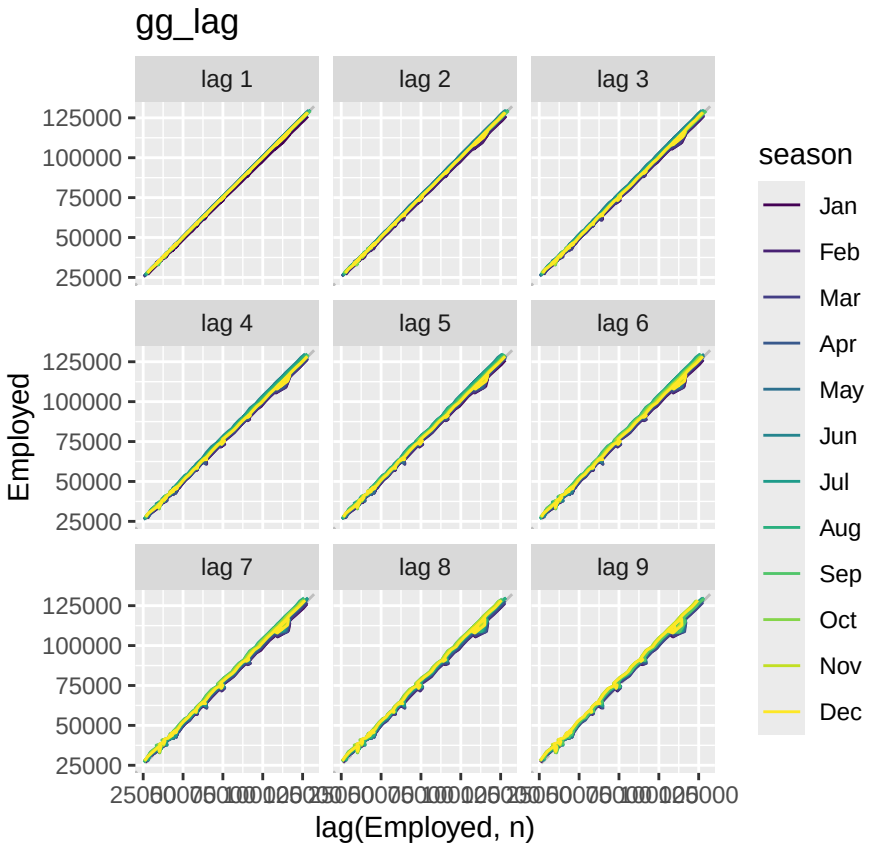

```
gg_subseries(total_private)+  labs(title = 'gg_subseries')
```

```
## Plot variable not specified, automatically selected ‘y = Employed‘
```

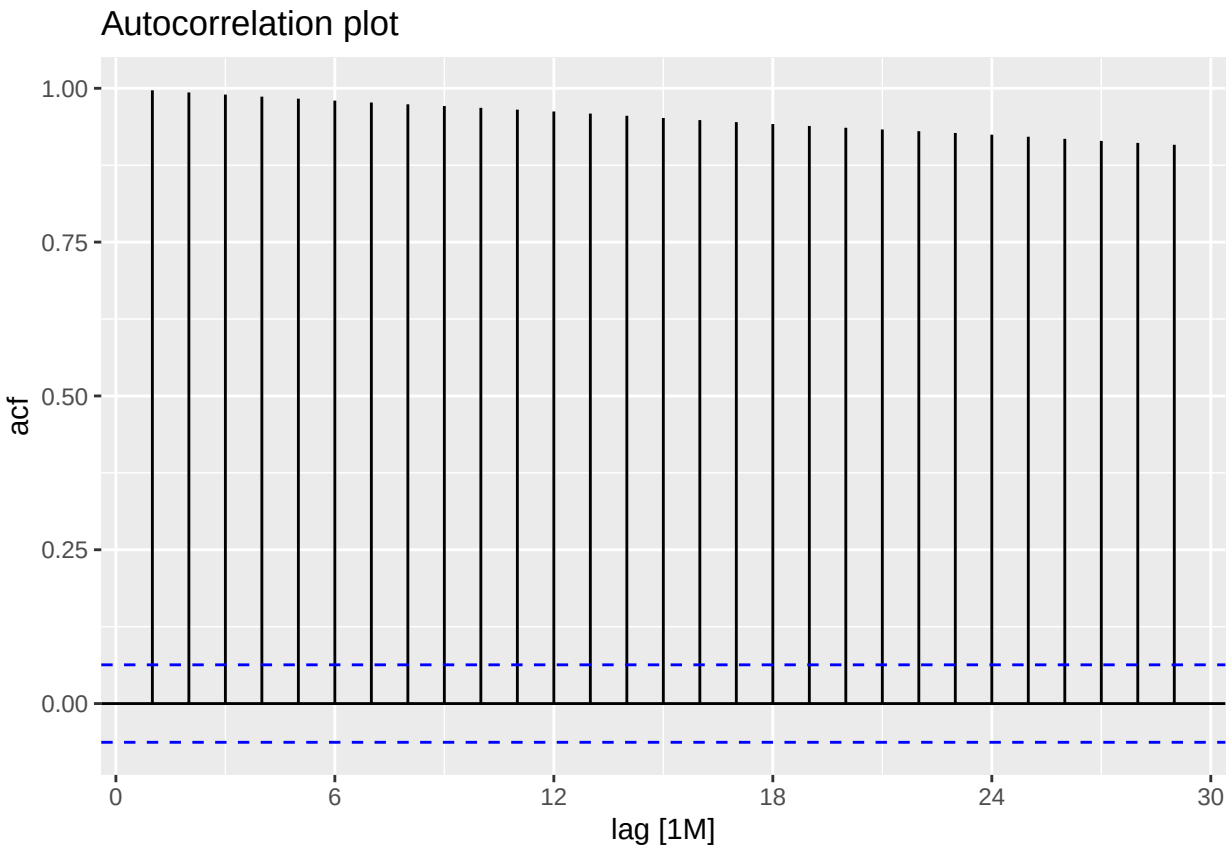



```
gg_lag(total_private)+  labs(title = 'gg_lag')
```

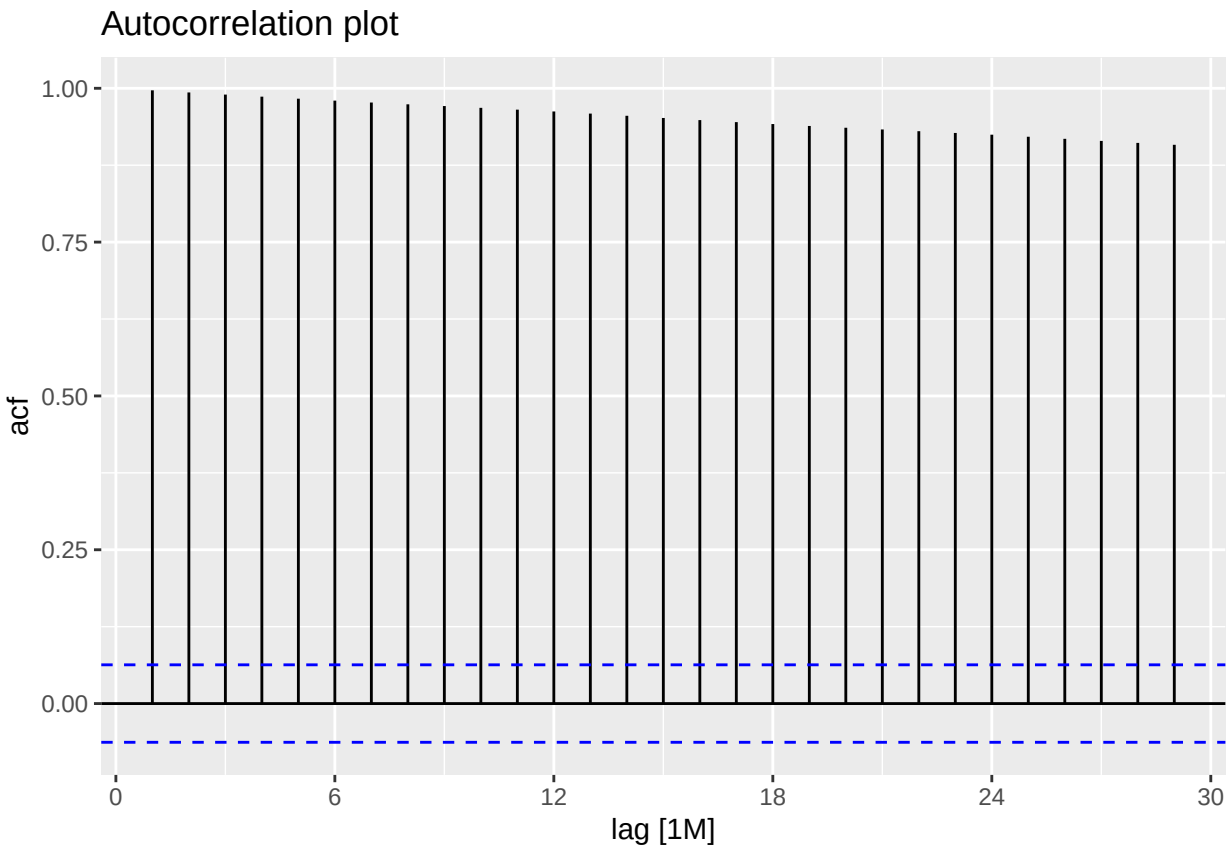
```
## Plot variable not specified, automatically selected ‘y = Employed‘
```




```
total_private %>%
  ACF(total_private$Employed) %>%
  autoplot() + labs(title = 'Autocorrelation plot')
```




```
total_private %>%
  ACF(total_private$Employed) %>%
  autoplot() + labs(title = 'Autocorrelation plot')
```



Us employment has an upward trend. It has been increasing over the years.

Data 2 aus_production:

Questions:

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

   Ans: There seems to be a seasonality in the data. The Bricks production usually is low at the beginning of a year's quarter and peaks at Q3.

2. What do you learn about the series?

   Ans: The gg_season and gg_lag plots do not offer much insight due to their structures. The line plot and the gg_subseries provide a lot of great insight into the seasonality patterns.

3. What can you say about the seasonal patterns?

   Ans: Yearly speaking, Q1 is usually the trough and Q3 is the peak of brick production.

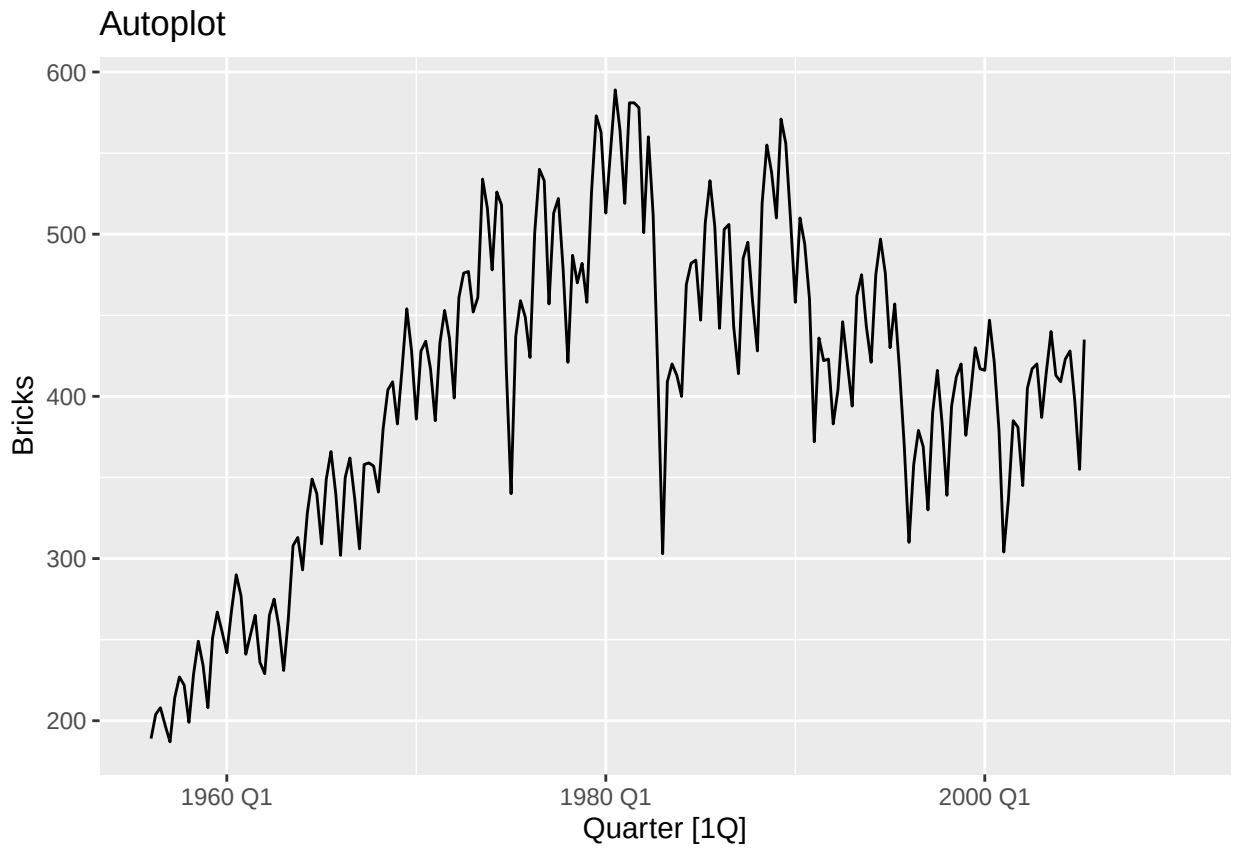
4. Can you identify any unusual years?

   Ans: 1980 had the brick production at all time high followed by the biggest drop in around 1981 Q1.

```
brick<- aus_production %>%
  select(Quarter, Bricks)
autoplot(brick)+ labs(title = 'Autoplot')
```

```
## Plot variable not specified, automatically selected `.vars = Bricks`
```

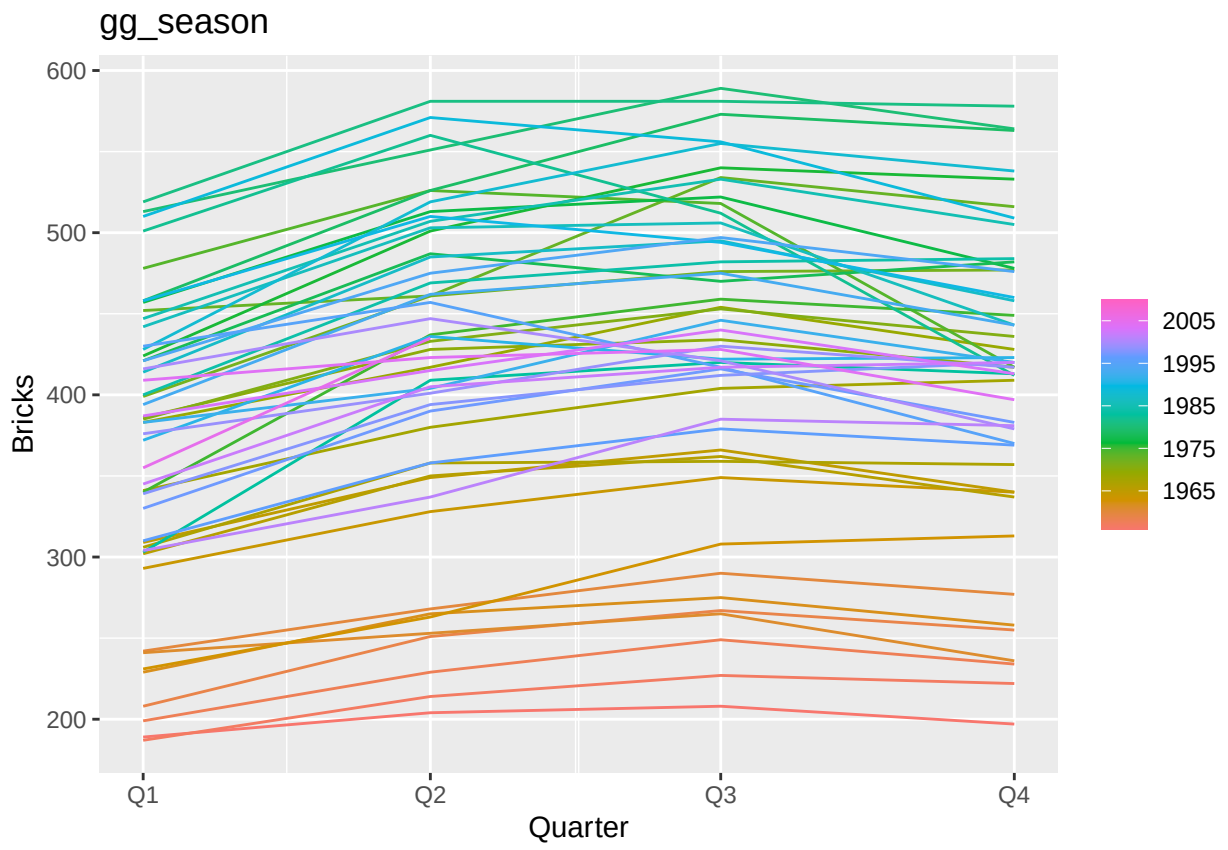
```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
gg_season(brick)+ labs(title = 'gg_season')
```

```
## Plot variable not specified, automatically selected `y = Bricks`
```

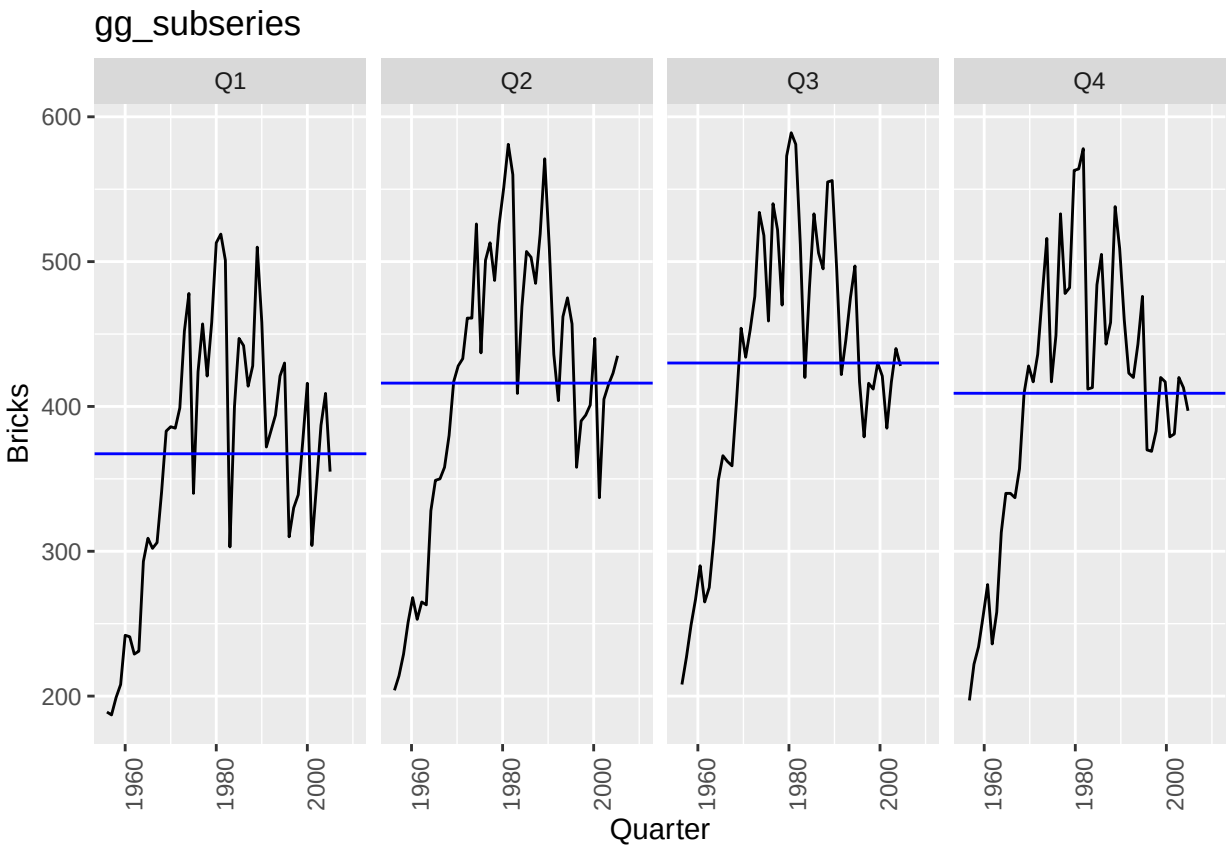
```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
gg_subseries(brick)+ labs(title = 'gg_subseries')
```

```
## Plot variable not specified, automatically selected `y = Bricks`
```

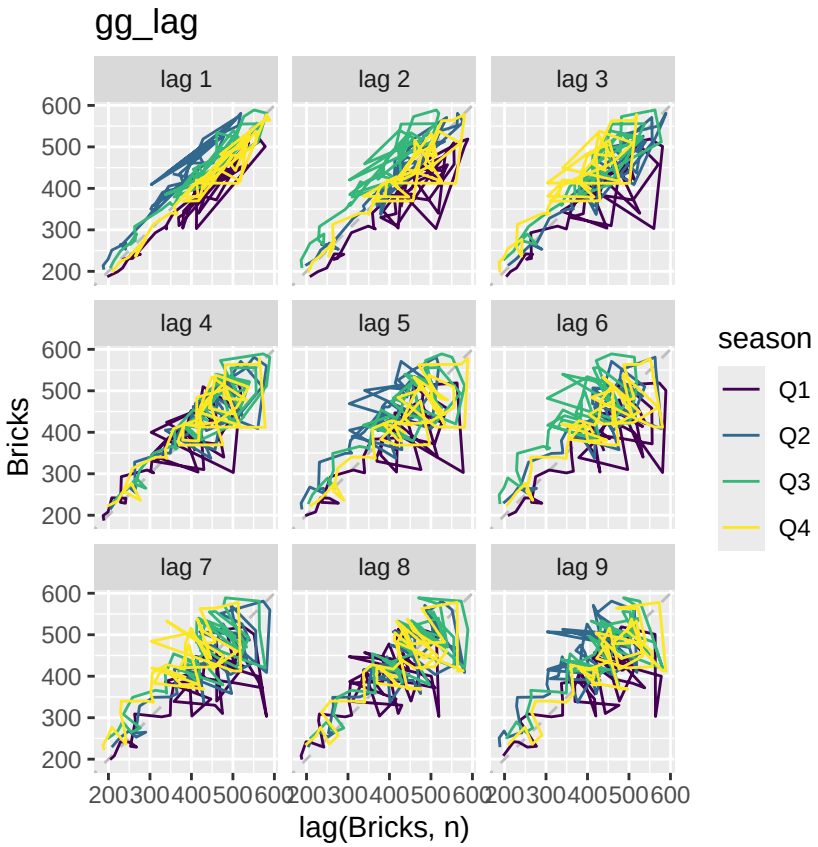
```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
gg_lag(brick)+ labs(title = 'gg_lag')
```

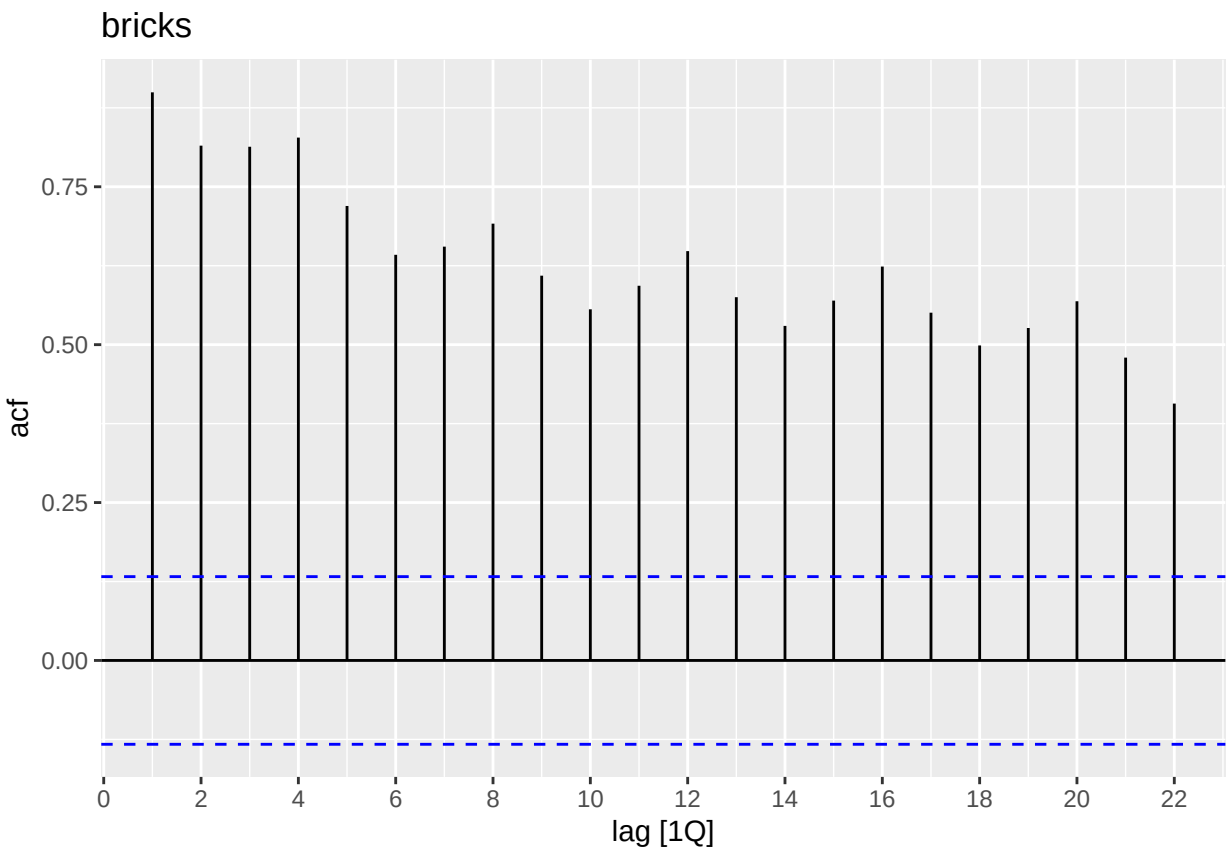
```
## Plot variable not specified, automatically selected ‘y = Bricks‘
```

```
## Warning: Removed 20 rows containing missing values (gg_lag).
```




```
ACF(brick) %>% autoplot() + labs(title='bricks')
```

```
## Response variable not specified, automatically selected ‘var = Bricks‘
```



Data 3 Hare from pelt

Questions:

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

   Ans: There is a seasonality in the hare pelt trade. The ups and downs are relatively constant throughout different year, hinting a feature of seasonality.

2. What do you learn about the series?

   Ans: The gg_season(unable to generate due to data type), gg_subseries, and gg_lag plots are not useful since there is only one variable for the observation. The ACF plot, however, reveals a seasonality pattern that for a negative correlation in trading hare pelt is observed every 10 years. There is a reversal of trading correlation from negative to positive and vice versa.

3. What can you say about the seasonal patterns?

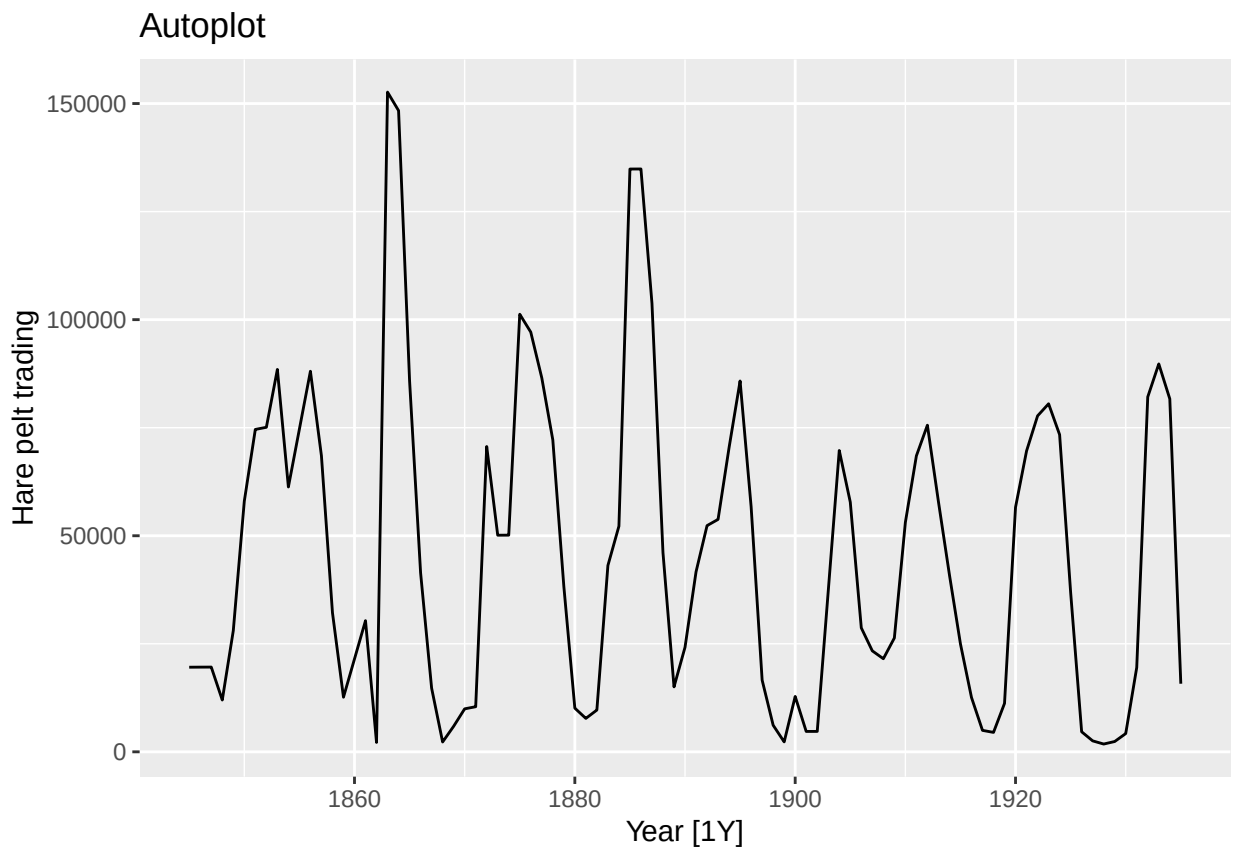
   Ans: The seasonal pattern shows a periodic reversal of correlation in pelt trading, roughly every 5 years.

4. Can you identify any unusual years?

   Ans: There is an unusual high trading record for hare pelt in around 1865 followed by a steep drop in 1870.

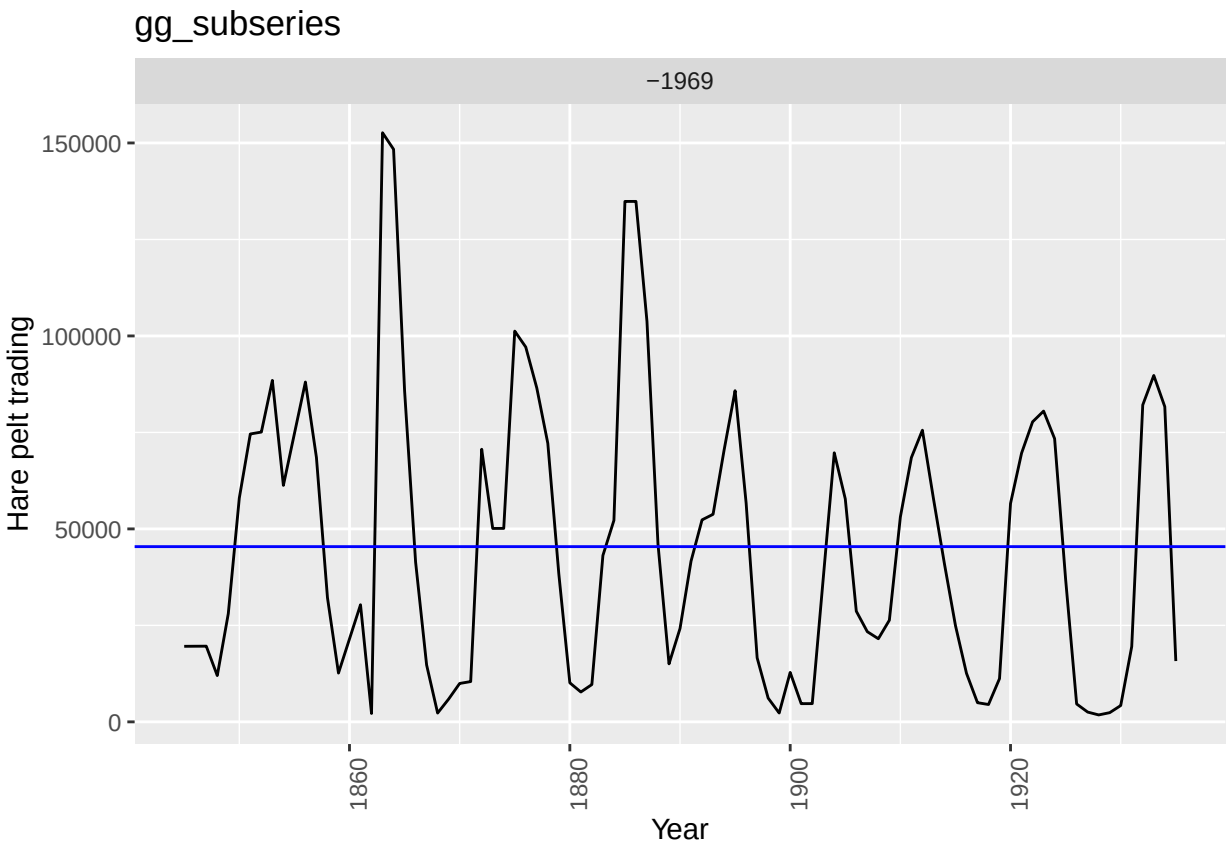
```
hare <- pelt %>%
  select(Year, Hare)

autoplot(hare) + labs(y='Hare pelt trading', title = 'Autoplot')
```

```
## Plot variable not specified, automatically selected `.vars = Hare`
```



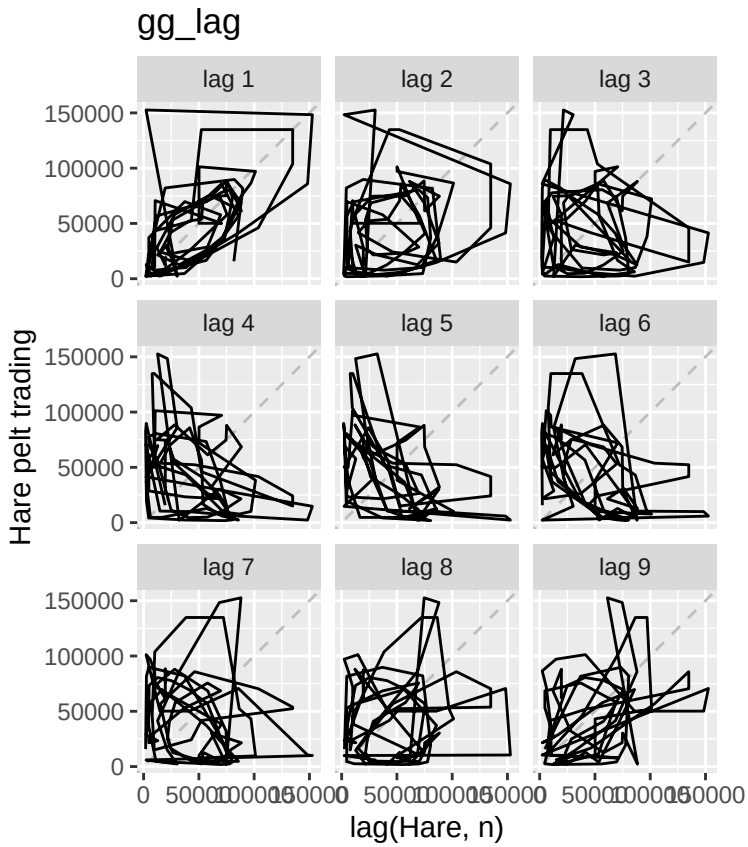

```
gg_subseries(hare)+labs(y='Hare pelt trading', title = 'gg_subseries')
```

```
## Plot variable not specified, automatically selected `y = Hare`
```



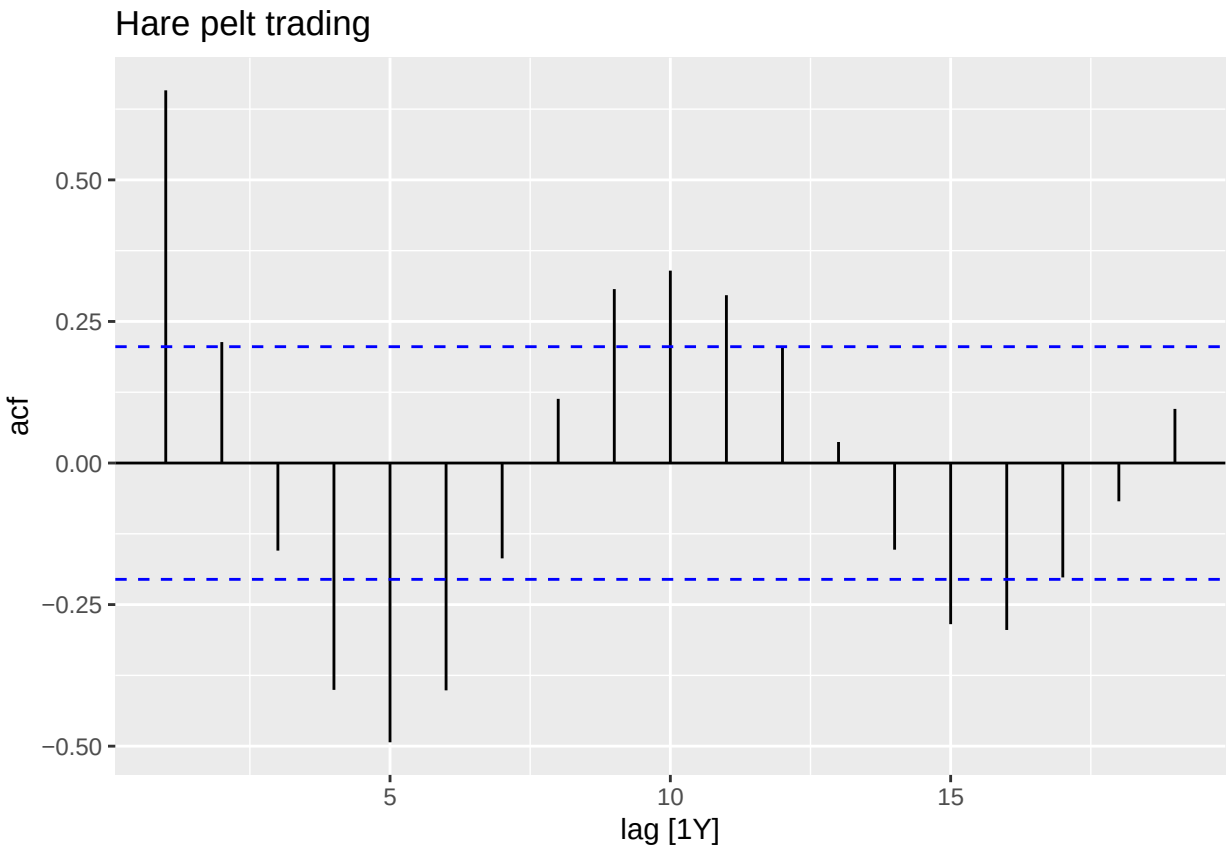

```
gg_lag(hare)+labs(y='Hare pelt trading', title = 'gg_lag')
```

```
## Plot variable not specified, automatically selected ‘y = Hare‘
```




```
ACF(hare) %>% autoplot() +labs(title = 'Hare pelt trading')
```

```
## Response variable not specified, automatically selected ‘var = Hare‘
```



Data 4 "H02" Cost from PBS

Question:

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

   Ans: Seasonality and upward trend is observed. The seasonality pattern is seen repeating in about 6 month time frame.

2. What do you learn about the series?

   Ans: The autoplot shows that there is a general upward trend for the cost of Medicare prescription in AUS. The gg_season is not particular revealing due to how massive the data series is. The gg_subseries shows that in the month of January, the cost of medicare is generally higher than other months of the year. gg_lag is not particular revealing. Finally, gg_lag shows a seasonality correlation that reverses in a period of about 6 months.

3. What can you say about the seasonal patterns?

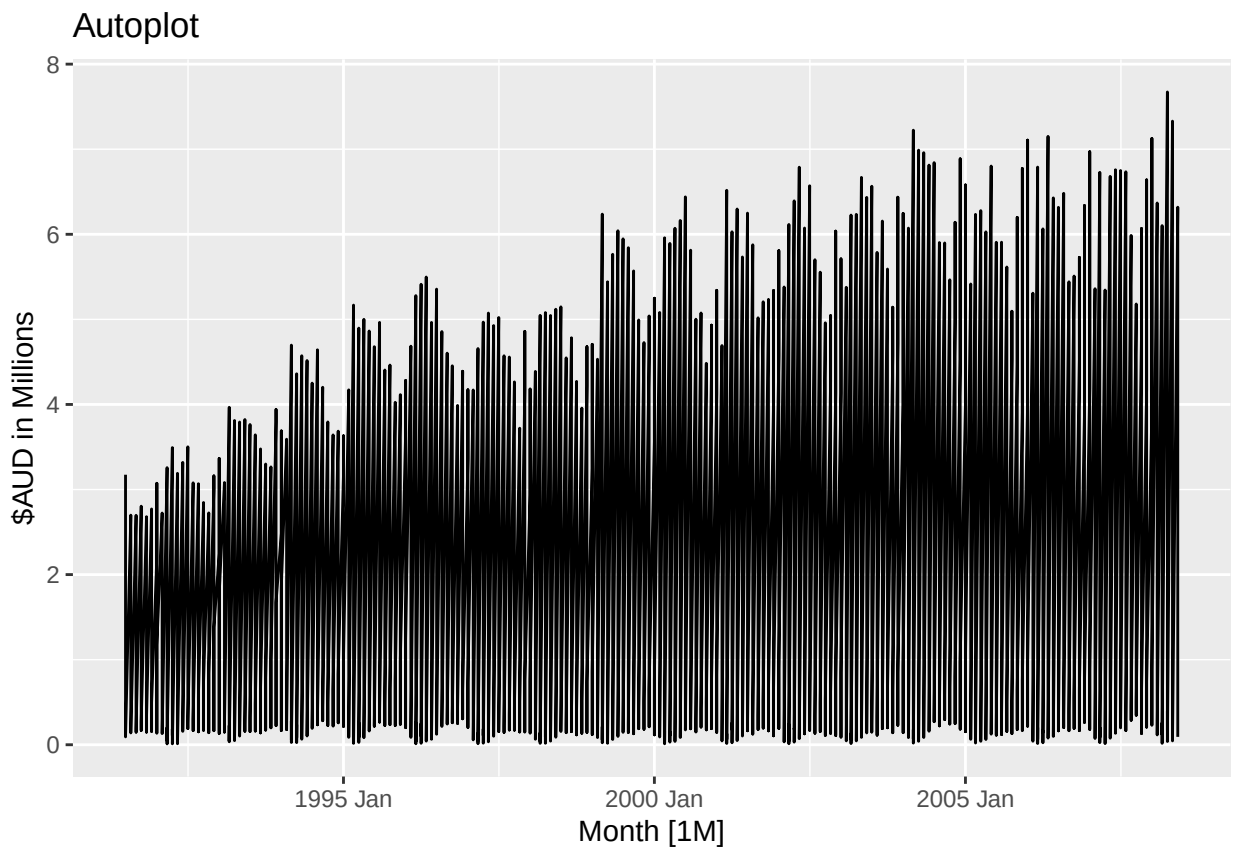
   Ans: The cost of medicare fluctuates in a 6 months period that transitions from positively correlated to negatively correlated and it repeats.

4. Can you identify any unusual years?

   Ans: There isn't a particular year that stands out.

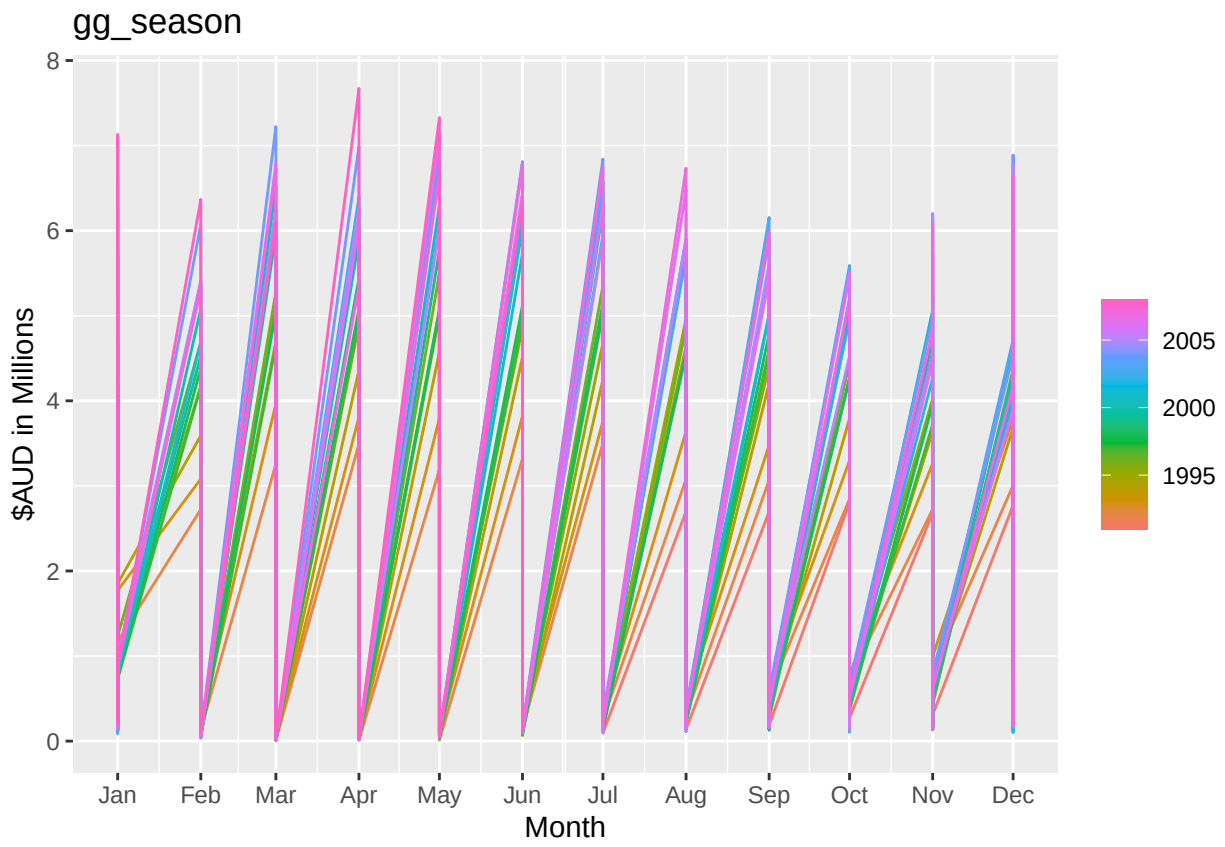
```
H02<- PBS %>%
  subset(ATC2=='H02', select = c(Month, Cost)) %>%
  mutate(Cost = Cost/100000)

autoplot(H02) + labs(y='$AUD in Millions', title = 'Autoplot')
```

```
## Plot variable not specified, automatically selected `.vars = Cost`
```



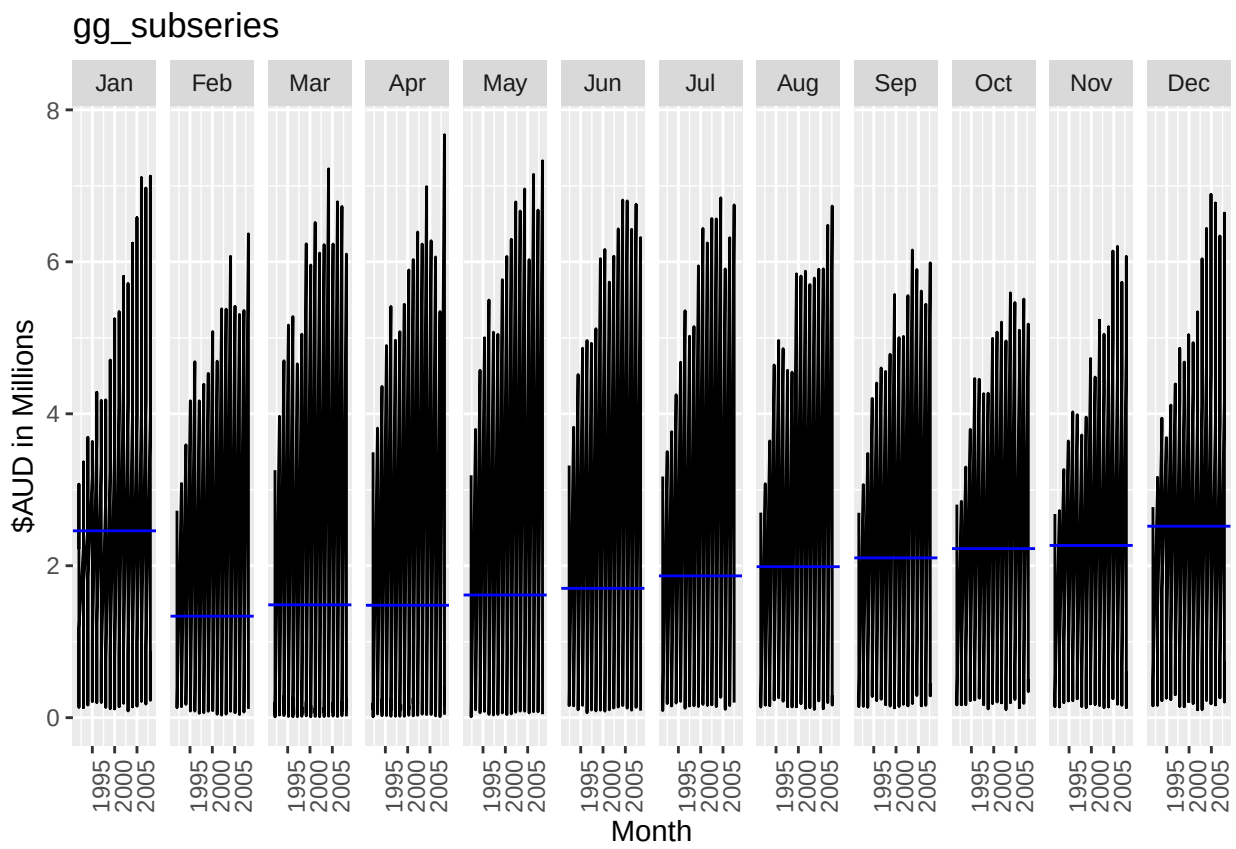

```
gg_season(H02) + labs(y='$AUD in Millions', title = 'gg_season')
```

```
## Plot variable not specified, automatically selected `y = Cost`
```



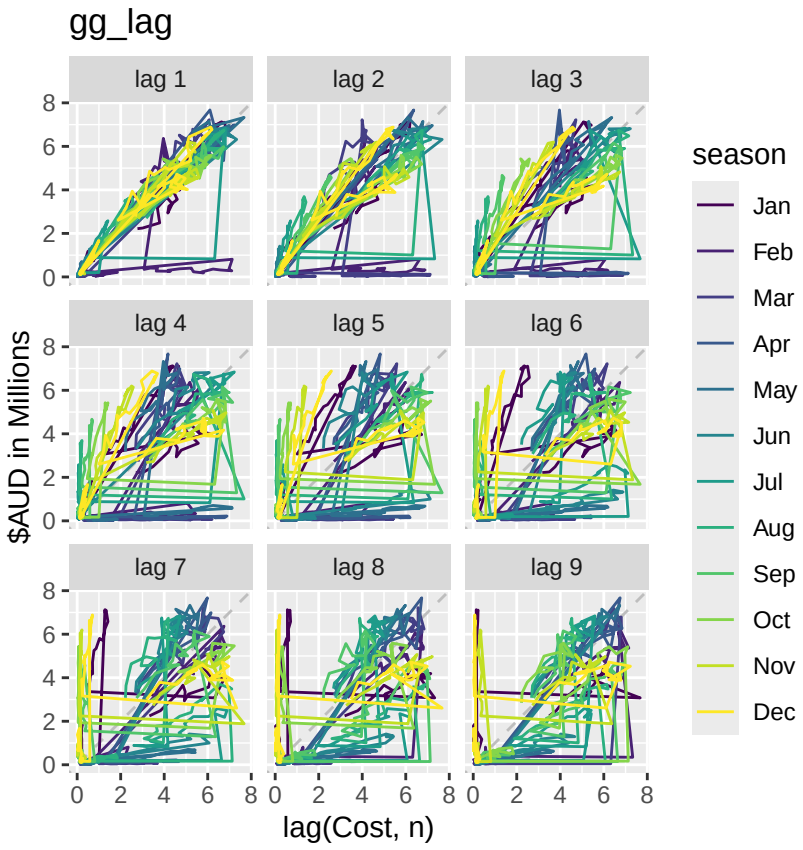

```
gg_subseries(H02)+labs(y='$AUD in Millions', title = 'gg_subseries')
```

```
## Plot variable not specified, automatically selected `y = Cost`
```



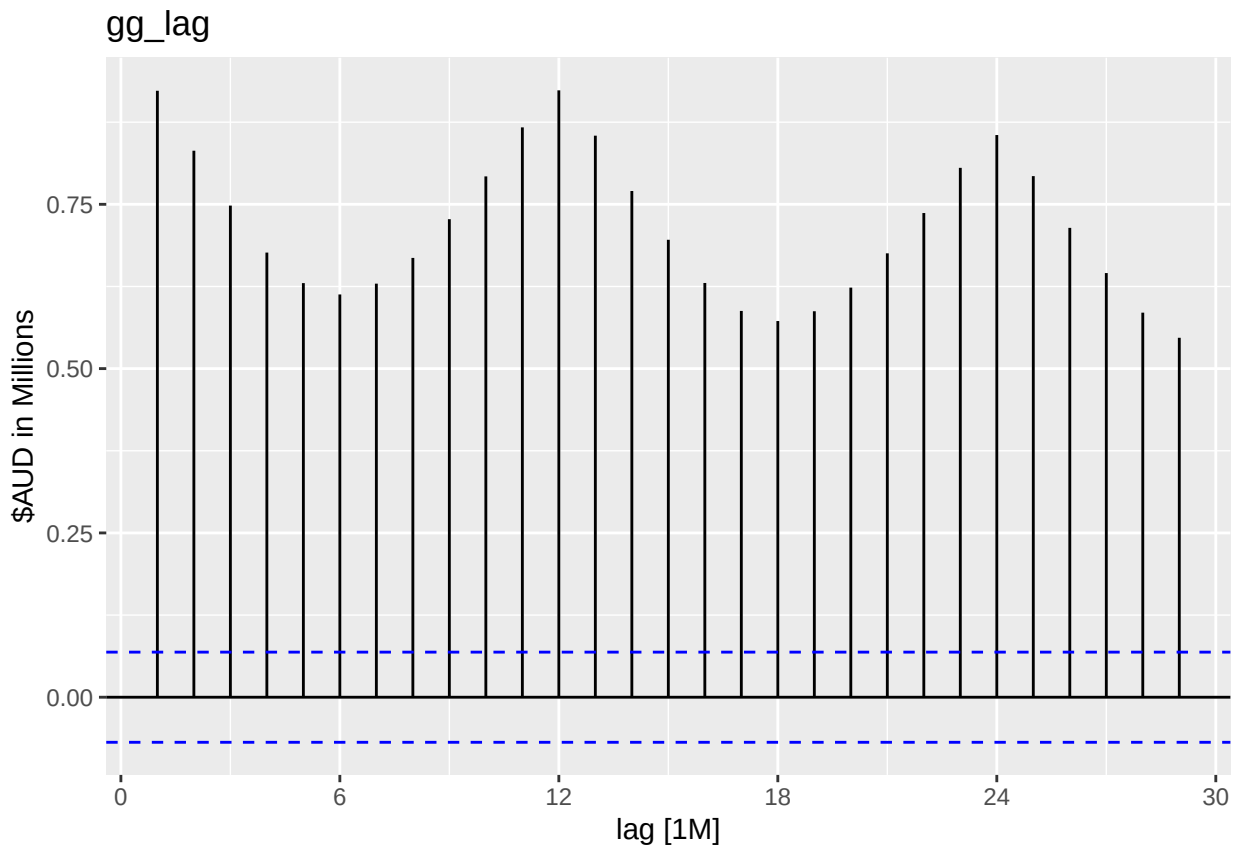

```
gg_lag(H02) + labs(y='$AUD in Millions', title = 'gg_lag')
```

```
## Plot variable not specified, automatically selected ‘y = Cost‘
```




```
ACF(H02) %>% autoplot()+labs(y='$AUD in Millions', title = 'gg_lag')
```

```
## Response variable not specified, automatically selected ‘var = Cost‘
```



Data 5 Barrels from us_gasoline

Questions:

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

   Ans: There is an upward trend of gasoline barrel production since 1991 to 2017. No distinct seasonality or cyclicity is observed.

2. What do you learn about the series?

   Ans: The line plot shows an upward trend for the gasoline production through the years. gg_season also confirms the upward trend for placing the later years higher in the graph and gg_subseries shows similar pattern as the autoplot line plot. ACF plot also shows that there is a general positive correlation between gasoline production and years (in weeks).

3. What can you say about the seasonal patterns?

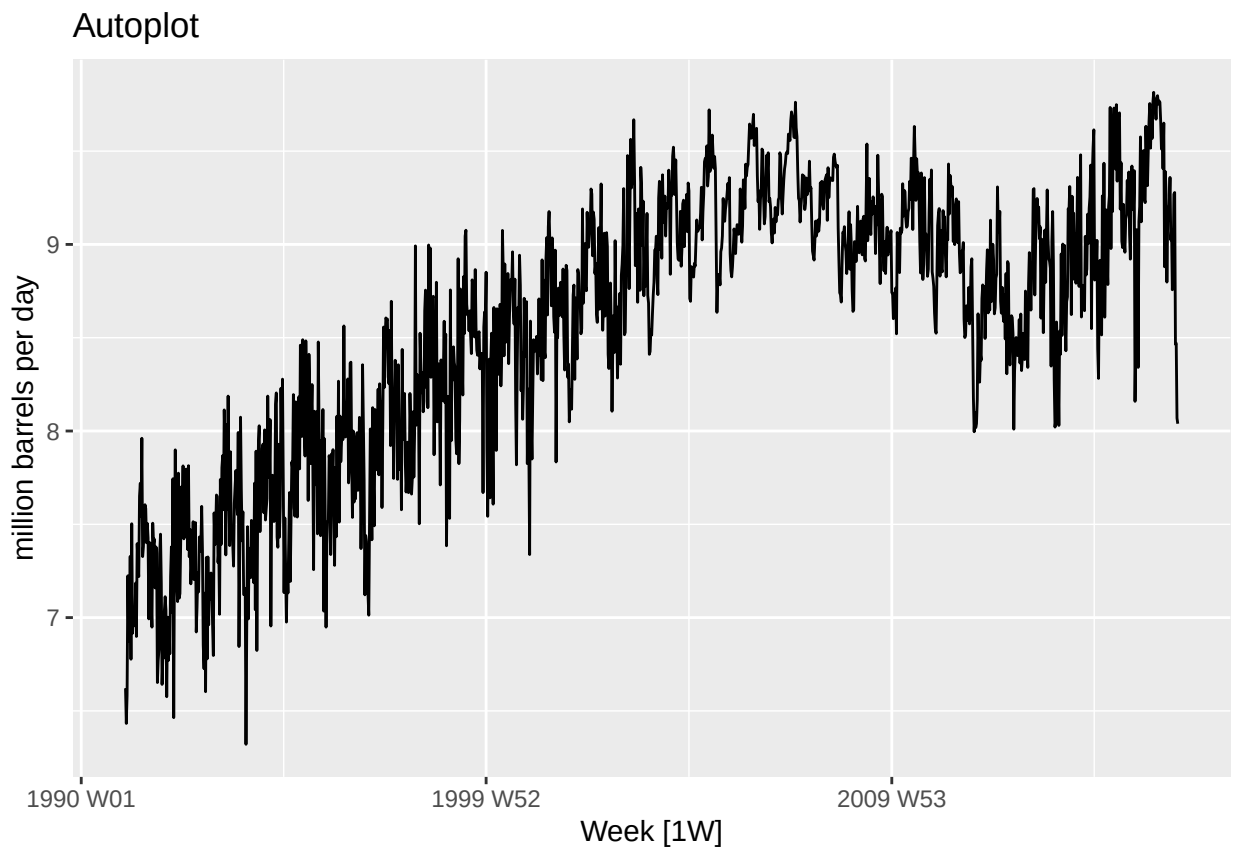
   Ans: There isn't a distinct seasonal pattern.

4. Can you identify any unusual years?

   Ans: There is a relatively big drop in gasoline production after 2009 week 53. However, the gasoline production picked up soon after.

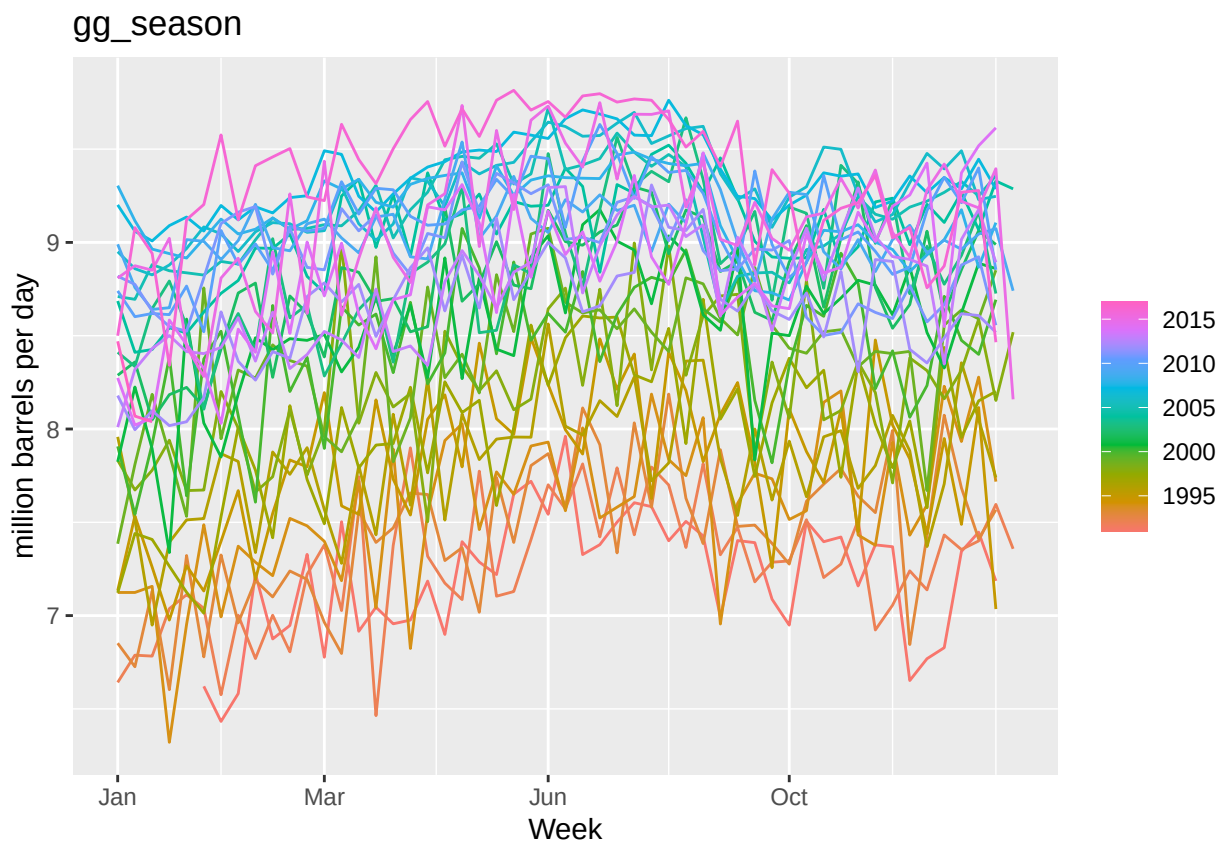
```
autoplot(us_gasoline) + labs(y='million barrels per day', title = 'Autoplot')
```

```
## Plot variable not specified, automatically selected `.vars = Barrels`
```



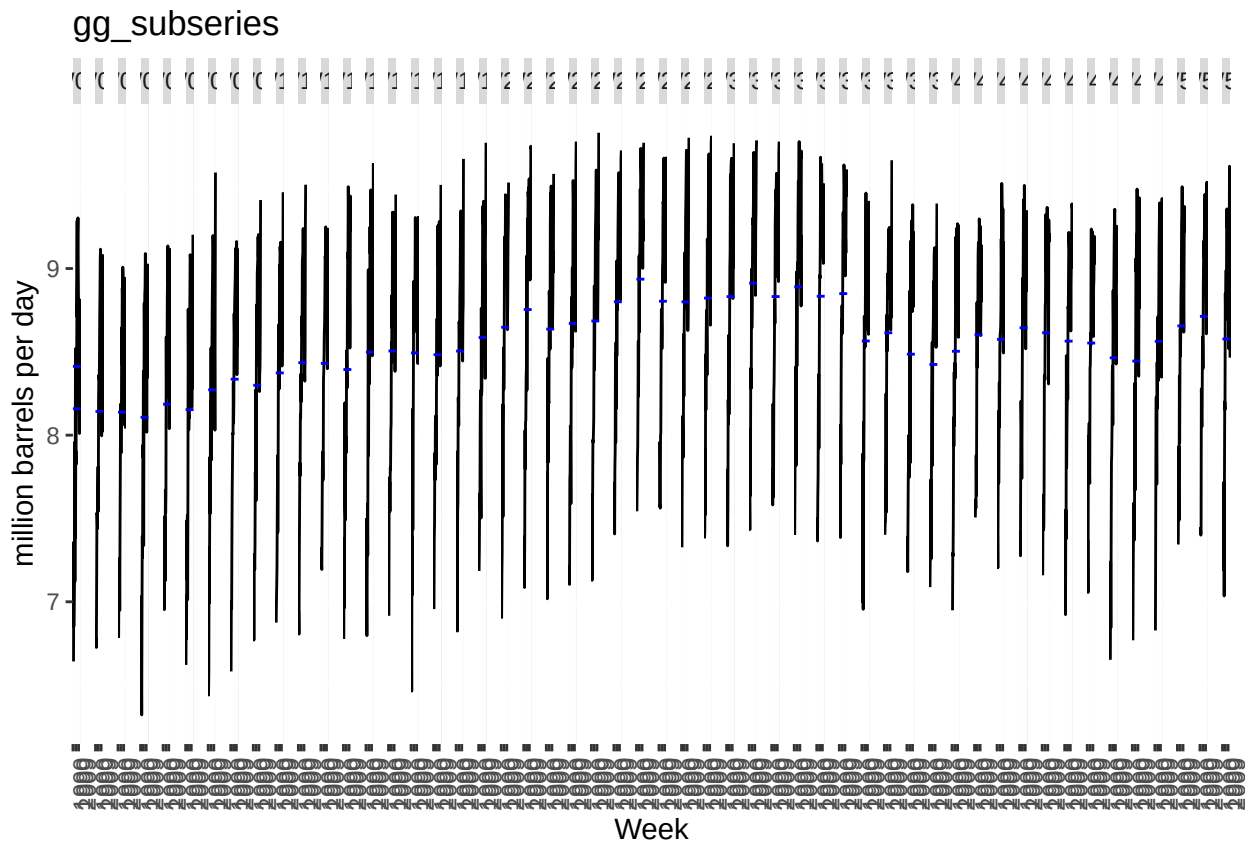

```
gg_season(us_gasoline) + labs(y='million barrels per day', title = 'gg_season')
```

```
## Plot variable not specified, automatically selected ‘y = Barrels‘
```



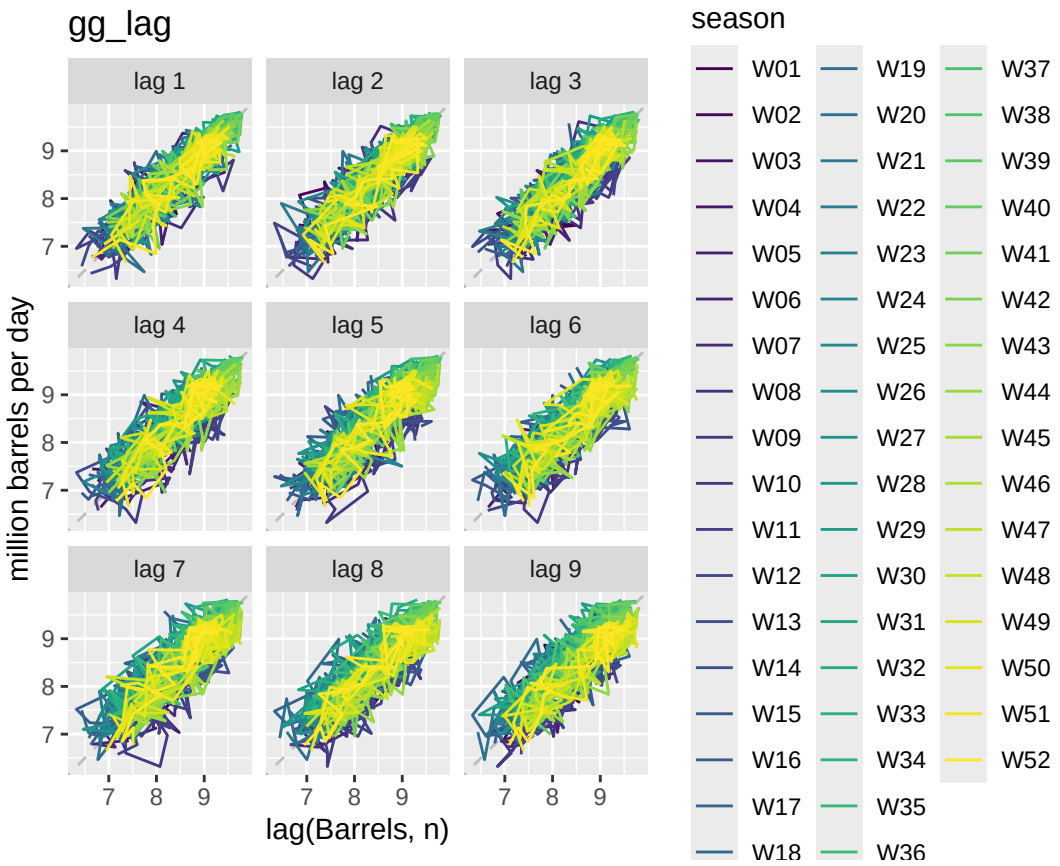

```
gg_subseries(us_gasoline)+labs(y='million barrels per day', title = 'gg_subseries')
```

```
## Plot variable not specified, automatically selected `y = Barrels`
```



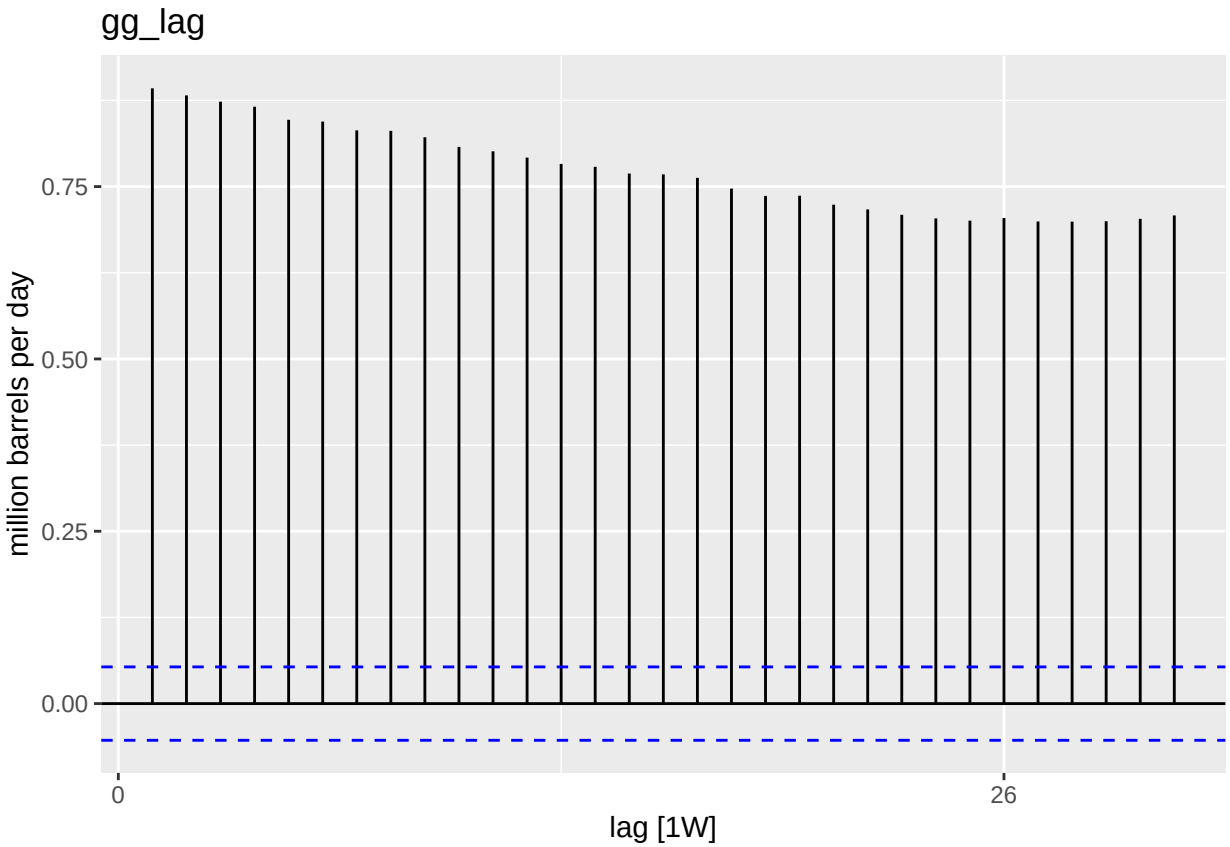

```
gg_lag(us_gasoline) + labs(y='million barrels per day', title = 'gg_lag')
```

```
## Plot variable not specified, automatically selected ‘y = Barrels‘
```




```
ACF(us_gasoline) %>% autoplot()+labs(y='million barrels per day', title = 'gg_lag')
```

```
## Response variable not specified, automatically selected ‘var = Barrels‘
```


gg_lag
million barrels per day
0.75
0.50
0.25
0.00
0
26
lag [1W]