

Data 624 Homework 1 Chapter 2.10

Enid Roman

2024-09-06

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

```
# Load required packages
```

```
library(fpp3)
library(tsibble)
library(ggplot2)
```

Quarterly production of selected commodities in Australia.

Quarterly estimates of selected indicators of manufacturing production in Australia. `aus_production` is a half-hourly tsibble with six values:

Beer: Beer production in megalitres. Tobacco: Tobacco and cigarette production in tonnes. Bricks: Clay brick production in millions of bricks. Cement: Portland cement production in thousands of tonnes. Electricity: Electricity production in gigawatt hours. Gas: Gas production in petajoules.

```
# View the first few rows of the aus_production dataset
head(aus_production)
```

```
## # A tsibble: 6 x 7 [1Q]
##   Quarter Beer Tobacco Bricks Cement Electricity Gas
##   <qtr> <dbl>   <dbl>   <dbl>   <dbl>       <dbl> <dbl>
## 1 1956 Q1   284     5225    189     465       3923    5
## 2 1956 Q2   213     5178    204     532       4436    6
## 3 1956 Q3   227     5297    208     561       4806    7
## 4 1956 Q4   308     5681    197     570       4418    6
## 5 1957 Q1   262     5577    187     529       4339    5
## 6 1957 Q2   228     5651    214     604       4811    7
```

```
# Get the list of dates from the 'Quarter' column.
start_date <- min(aus_production$Quarter)
end_date <- max(aus_production$Quarter)
start_date
```

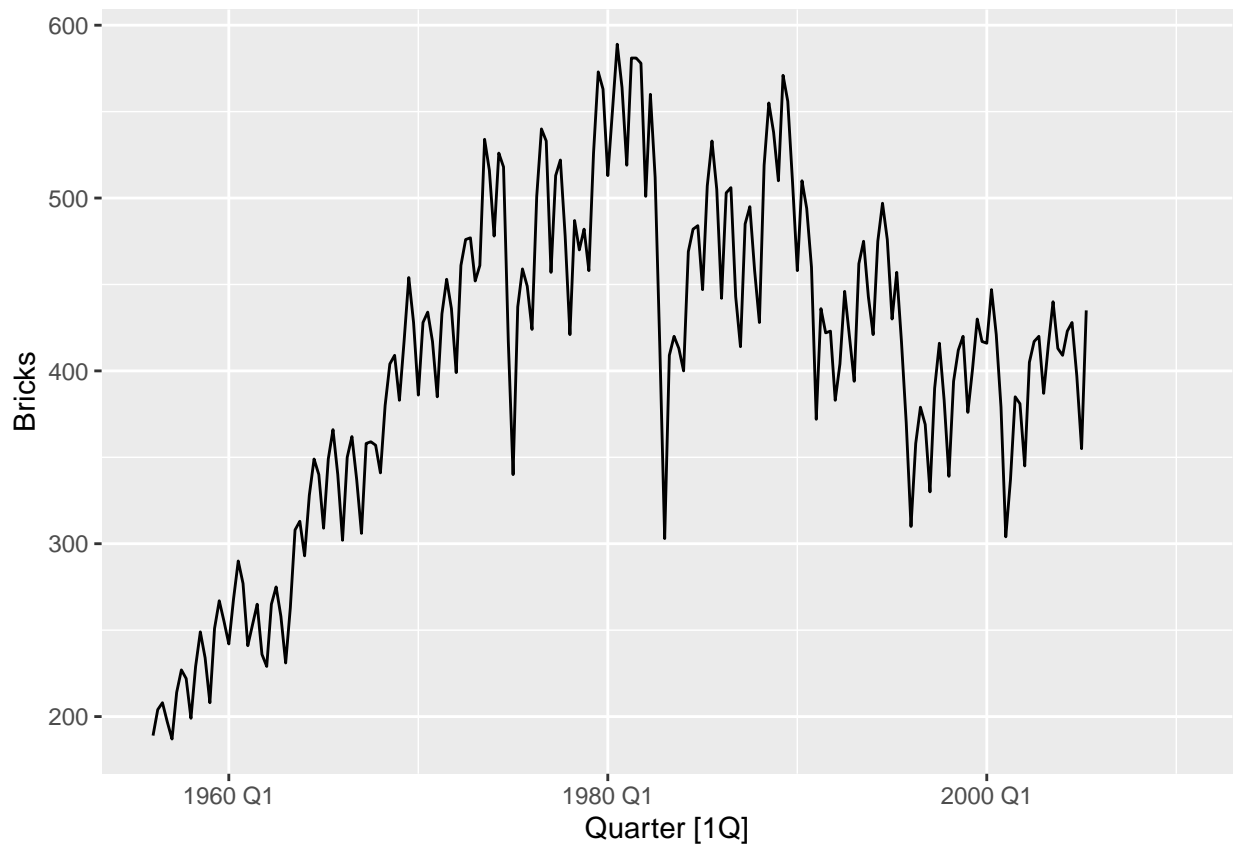
```
## <yearquarter[1]>
## [1] "1956 Q1"
## # Year starts on: January
```

```
end_date
```

```
## <yearquarter[1]>  
## [1] "2010 Q2"  
## # Year starts on: January
```

The time interval for Aus_production is Quarterly and it extends from 1956 to 2010.

```
# Create a time series plot of Bricks production  
aus_production |>  
autoplot(Bricks)
```



In aus_production time series the first half from mid 1900 to about 1975 there was a strong increasing trend. Lots of bricks were being produced and trend is going upward. Then we have a big decrease in production during mid 70s and mid 80s during Australia's recession. From then on there was increase and decrease in production. This is very cyclical, the data exhibit rises and falls that are not of a fixed frequency.

Pelt trading records.

Hudson Bay Company trading records for Snowshoe Hare and Canadian Lynx furs from 1845 to 1935. This data contains trade records for all areas of the company. Pelt is an annual tsibble with two values:

Hare: The number of Snowshoe Hare pelts traded. Lynx: The number of Canadian Lynx pelts traded.

```
# View the first few rows of the Pelt dataset  
head(pelt)
```

```
## # A tsibble: 6 x 3 [1Y]  
##   Year  Hare  Lynx  
##   <dbl> <dbl> <dbl>  
## 1  1845 19580 30090  
## 2  1846 19600 45150  
## 3  1847 19610 49150  
## 4  1848 11990 39520  
## 5  1849 28040 21230  
## 6  1850 58000  8420
```

```
# Get the list of dates from the 'Year' column.  
start_date <- min(pelt$Year)  
end_date <- max(pelt$Year)  
start_date
```

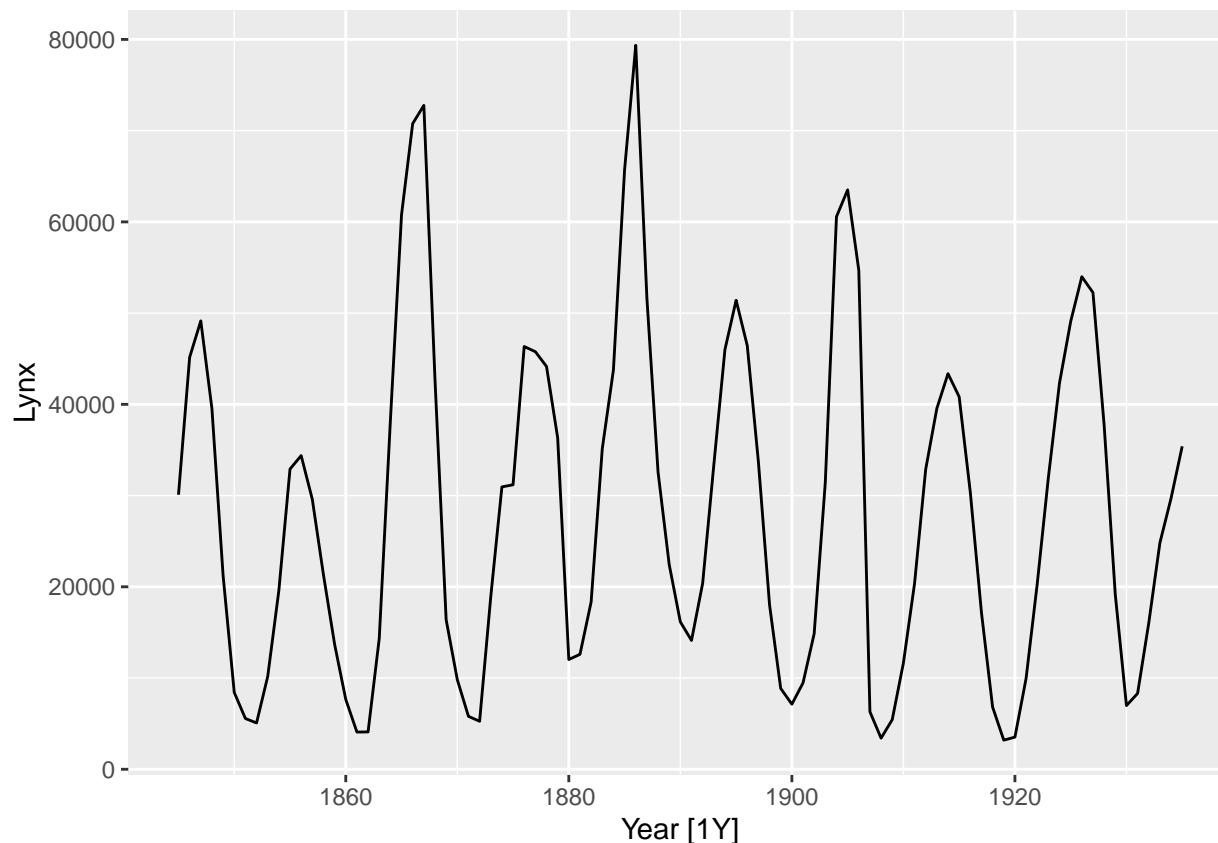
```
## [1] 1845
```

```
end_date
```

```
## [1] 1935
```

The time interval for Pelt is Yearly and it extends from 1845 to 1935.

```
# Create a time series plot of Lynx  
pelt |>  
  autoplot(Lynx)
```



In Pelt time series the pattern seems to be seasonal but is not seasonal cause annual data does not have seasonality. This time series has a smooth pattern so this makes it a cycle and a life cycle of the lynx

Gafa stock prices.

Historical stock prices from 2014-2018 for Google, Amazon, Facebook and Apple. All prices are in \$USD. gafa_stock is a tsibble containing data on irregular trading days:

Open: The opening price for the stock. High: The stock's highest trading price. Low: The stock's lowest trading price. Close: The closing price for the stock. Adj_Close: The adjusted closing price for the stock. Volume: The amount of stock traded. Each stock is uniquely identified by one key:

Symbol: The ticker symbol for the stock.

```
# View the first few rows of the gafa_stock dataset
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
```

```
# Get the list of dates from the 'Year' column.
start_date <- min(gafa_stock$Date)
end_date <- max(gafa_stock$Date)
start_date
```

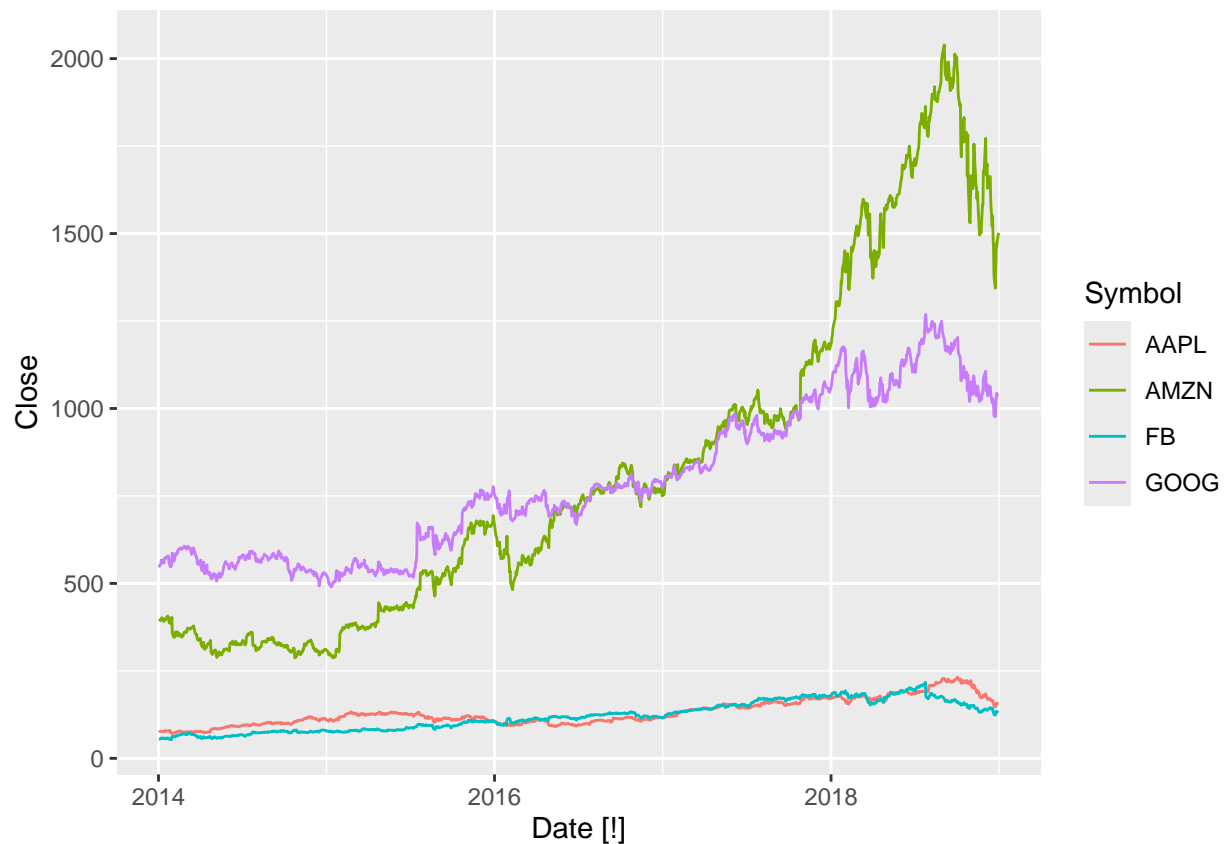
```
## [1] "2014-01-02"
```

```
end_date
```

```
## [1] "2018-12-31"
```

The time interval for gafa_stock is Yearly and it extends from 1845 to 1935.

```
# Plot the closing prices from the gafa_stock dataset
gafa_stock |>
  autoplot(Close)
```



Half-hourly electricity demand for Victoria, Australia.

vic_elec is a half-hourly tsibble with three values:

Demand: Total electricity demand in MWh. Temperature: Temperature of Melbourne (BOM site 086071).
 Holiday: Indicator for if that day is a public holiday.

This data is for operational demand, which is the demand met by local scheduled generating units, semi-scheduled generating units, and non-scheduled intermittent generating units of aggregate capacity larger than 30 MWh, and by generation imports to the region. The operational demand excludes the demand met by non-scheduled non-intermittent generating units, non-scheduled intermittent generating units of aggregate capacity smaller than 30 MWh, exempt generation (e.g. rooftop solar, gas tri-generation, very small wind farms, etc), and demand of local scheduled loads. It also excludes some very large industrial users (such as mines or smelters).

```
# View the first few rows of the vic_elec dataset
head(vic_elec)
```

```
## # A tibble: 6 x 5 [30m] <Australia/Melbourne>
##   Time                Demand Temperature Date      Holiday
##   <dtm>              <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
```

```
# Get the list of dates from the 'Year' column.
start_date <- min(vic_elec$Date)
end_date <- max(vic_elec$Date)
start_date
```

```
## [1] "2012-01-01"
```

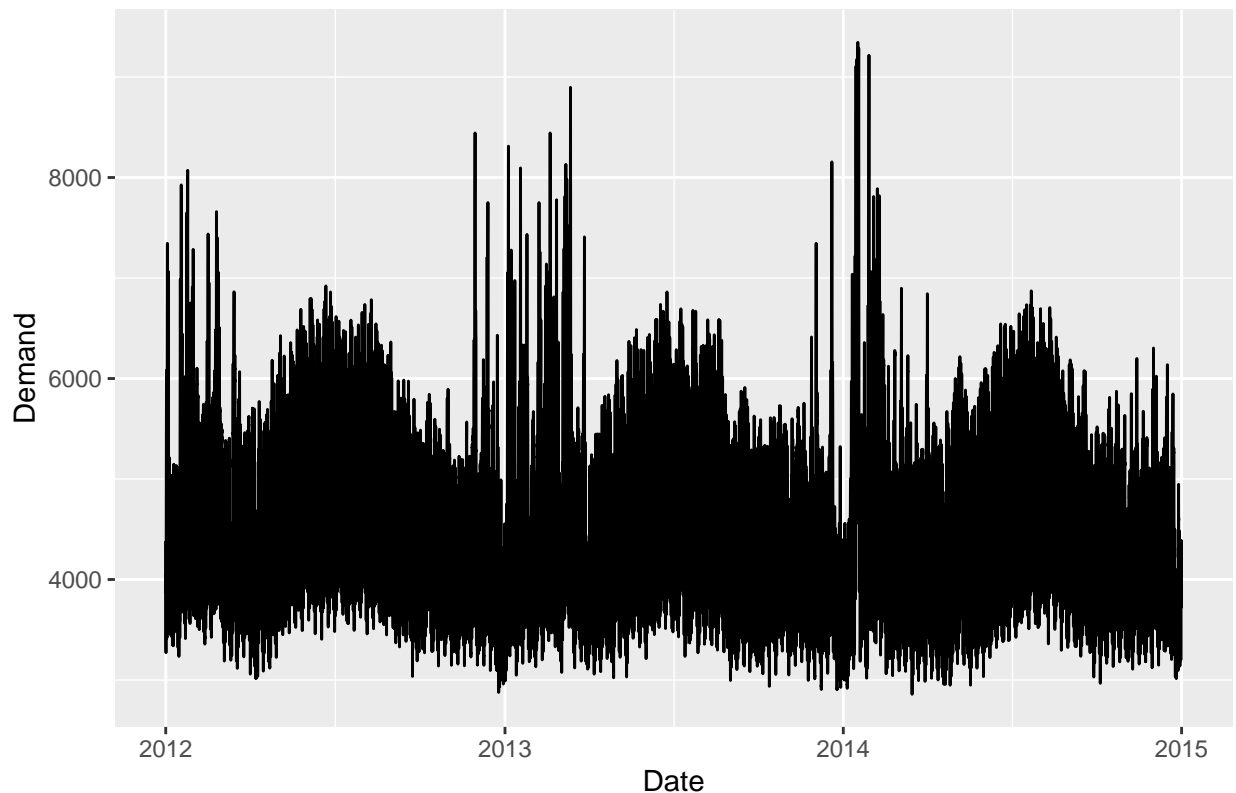
```
end_date
```

```
## [1] "2014-12-31"
```

The time interval for vic_elec is half-hourly and it extends from 1/1/2012 to 12/31/2014.

```
# Modifying Axis and Title
vic_elec %>%
  autoplot(Demand) +
  labs(x = "Date", y = "Demand") +
  ggtitle("Half-hourly Electricity Demand for Victoria, Australia")
```

Half-hourly Electricity Demand for Victoria, Australia



In `vic_elec` time series is multiple seasonality. Dec., Jan., and Feb., summer month in Australia we see a lot of variations and spikes in demand on very hot days, and a lot of air conditioner happening. This is why we see the increase variation in the summer. Then as we move along the x axis to autumn we have a lower demand than summer and higher in winter months, June, July, and August. Then we have spring which is fairly stable season in Victoria.

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

Peak Closing Price

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

```
## # A tibble: 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.
```

Apple had a Max at 232.07 on 10/03/2018. Amazon had a max of 2039.51 on 9/4/2018. Facebook had a max of 217.50 on 7/25/2018. Google had a max of 1268.33 on 7/26/2018.

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.

Tutel

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

```
# Upload csv file and view the first few rows of the aus_production dataset
tute1 <- read.csv("https://raw.githubusercontent.com/enidroman/Data-624-Predictive-Analytics/main/tute1.csv")
head(tute1)
```

```
##      Quarter  Sales AdBudget   GDP
## 1  3/1/1981 1020.2    659.2 251.8
## 2  6/1/1981  889.2    589.0 290.9
## 3  9/1/1981  795.0    512.5 290.8
## 4 12/1/1981 1003.9    614.1 292.4
## 5  3/1/1982 1057.7    647.2 279.1
## 6  6/1/1982  944.4    602.0 254.0
```

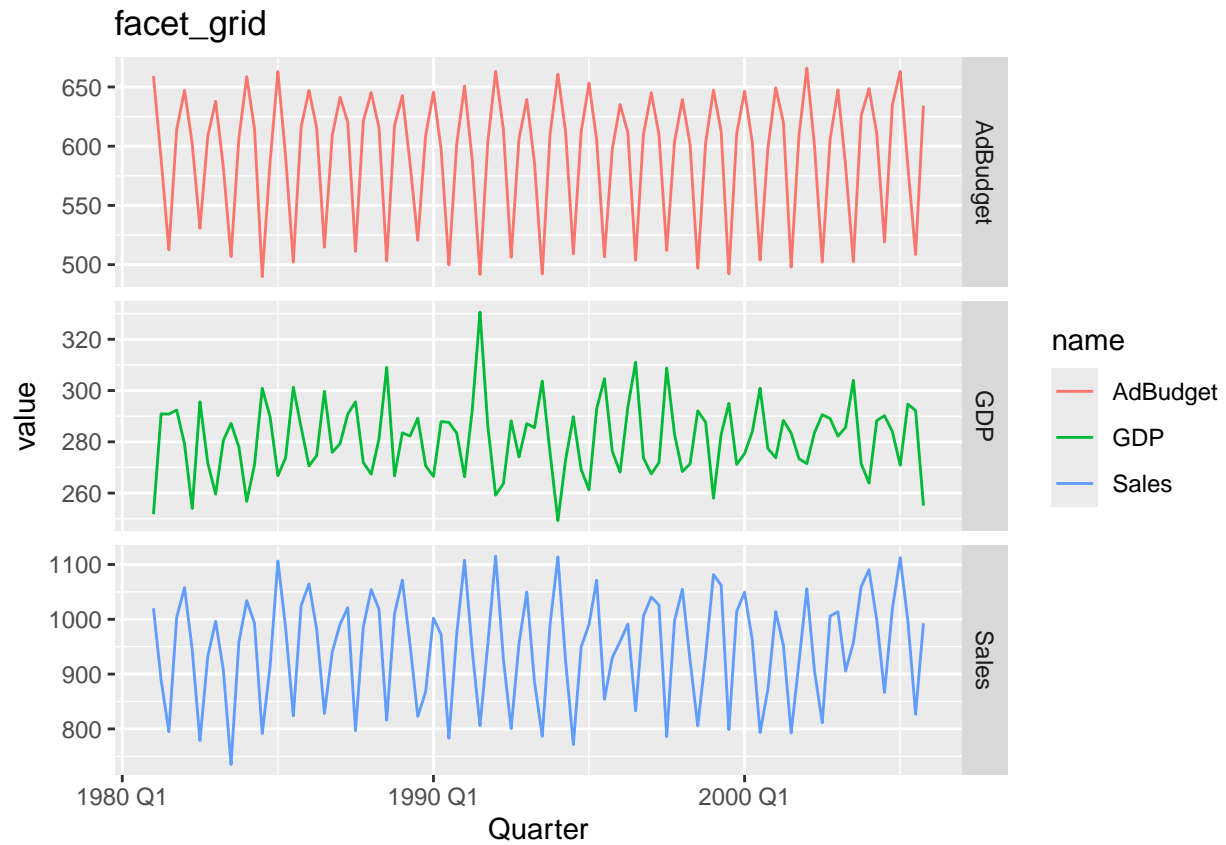
- b. Convert the data to time series.

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

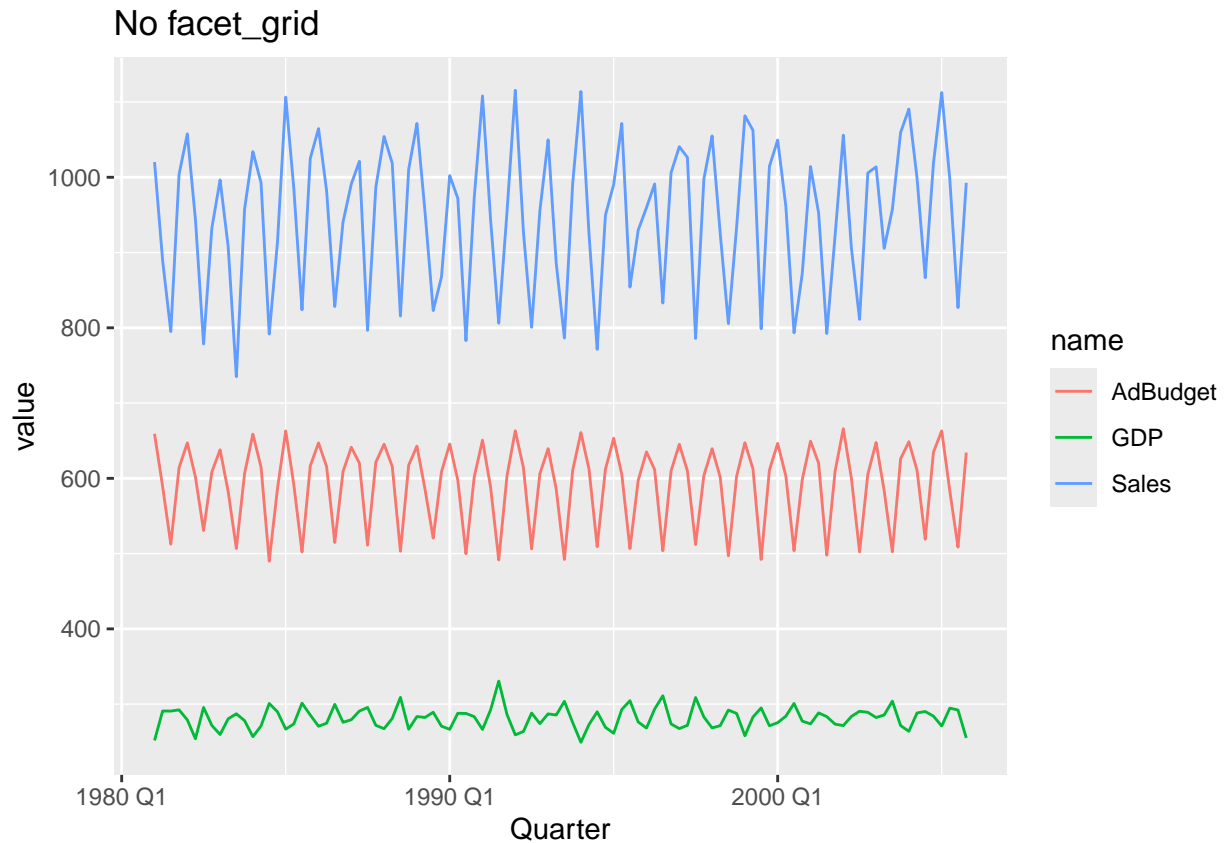
- c. Construct time series plots of each of the three series.

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

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

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



When you don't include `facet_grid()`, the plot will not be divided into separate panels; instead, you will get a single plot that displays all the data points together.

When you include `facet_grid()`, the plot will be split into multiple panels based on the levels of the categorical variable, making it easier to compare groups visually.

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

USgas

- a. Install the USgas package.

```
#install.packages('USgas')
library(USgas)
data("us_total")
str(us_total)
```

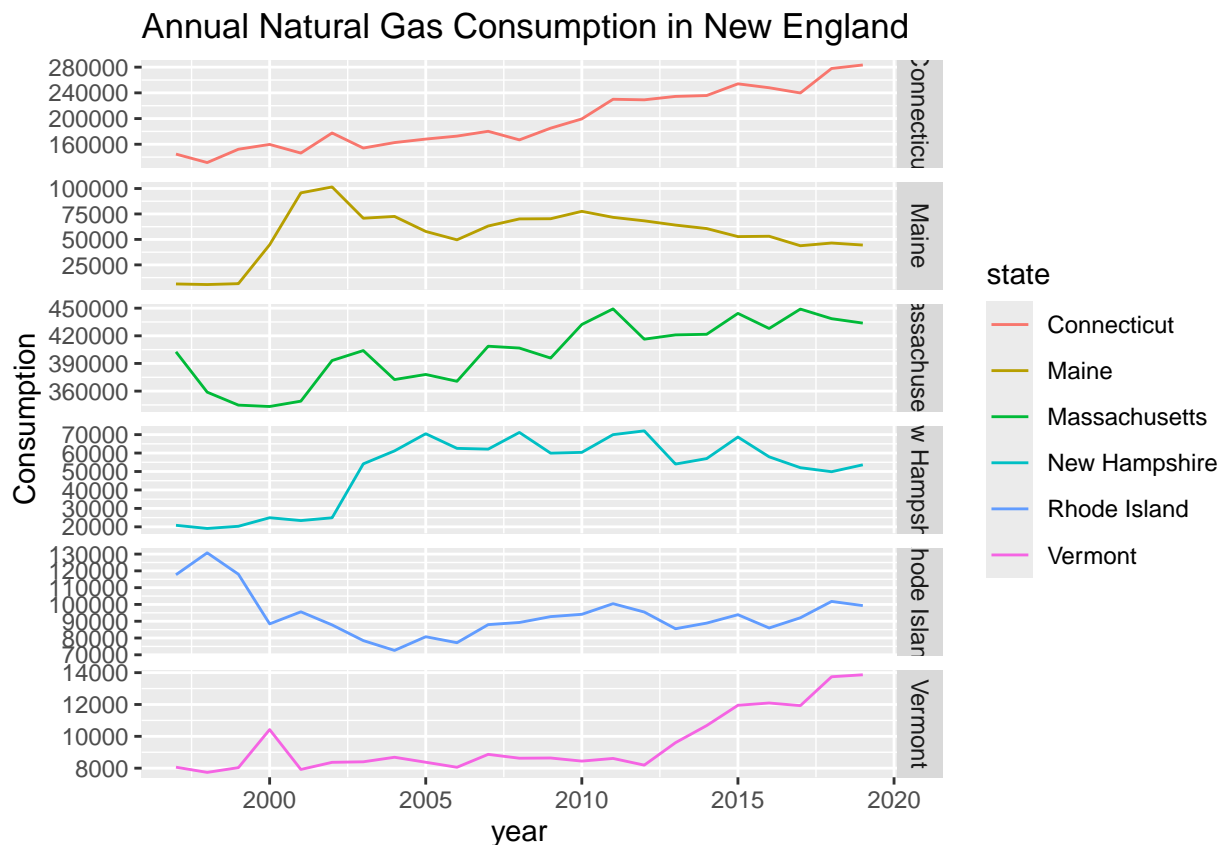
```
## 'data.frame':   1266 obs. of  3 variables:
##  $ year : int   1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 ...
##  $ state: chr   "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ y    : int  324158 329134 337270 353614 332693 379343 350345 382367 353156 391093 ...
```

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

```
us_total <- us_total %>%
  as_tibble(key = state,
            index = year)
```

- 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 Island')) +
  ggplot(aes(x = year, y = y, colour = state)) +
  geom_line() +
  facet_grid(state ~ ., scales = "free_y") +
  labs(title = "Annual Natural Gas Consumption in New England",
       y = "Consumption")
```



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

Tourism

```
tourism <- read.csv("https://raw.githubusercontent.com/enidroman/Data-624-Predictive-Analytics/main/tourism")
head(tourism)
```

```
##      Quarter Region      State Purpose   Trips
## 1 1998-01-01 Adelaide South Australia Business 135.0777
## 2 1998-04-01 Adelaide South Australia Business 109.9873
## 3 1998-07-01 Adelaide South Australia Business 166.0347
## 4 1998-10-01 Adelaide South Australia Business 127.1605
## 5 1999-01-01 Adelaide South Australia Business 137.4485
## 6 1999-04-01 Adelaide South Australia Business 199.9126
```

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

```
tourism_ts <- tourism %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(key = c(Region, State, Purpose),
            index = Quarter)
```

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

```
tourism_ts %>%
  group_by(Region, Purpose) %>%
  mutate(Avg_Trips = mean(Trips)) %>%
  ungroup() %>%
  filter(Avg_Trips == max(Avg_Trips)) %>%
  distinct(Region, Purpose)
```

```
## # A tibble: 1 x 2
##   Region Purpose
##   <chr>   <chr>
## 1 Sydney Visiting
```

Sydney, Australia has the maximum number of overnight trips on average for Visiting.

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

```
# Tsibble for Total Trips by State
total_trips_by_state <- tourism %>%
  group_by(State) %>%
  summarise(total_trips = sum(Trips)) %>%
  arrange(desc(total_trips))

head(total_trips_by_state)
```

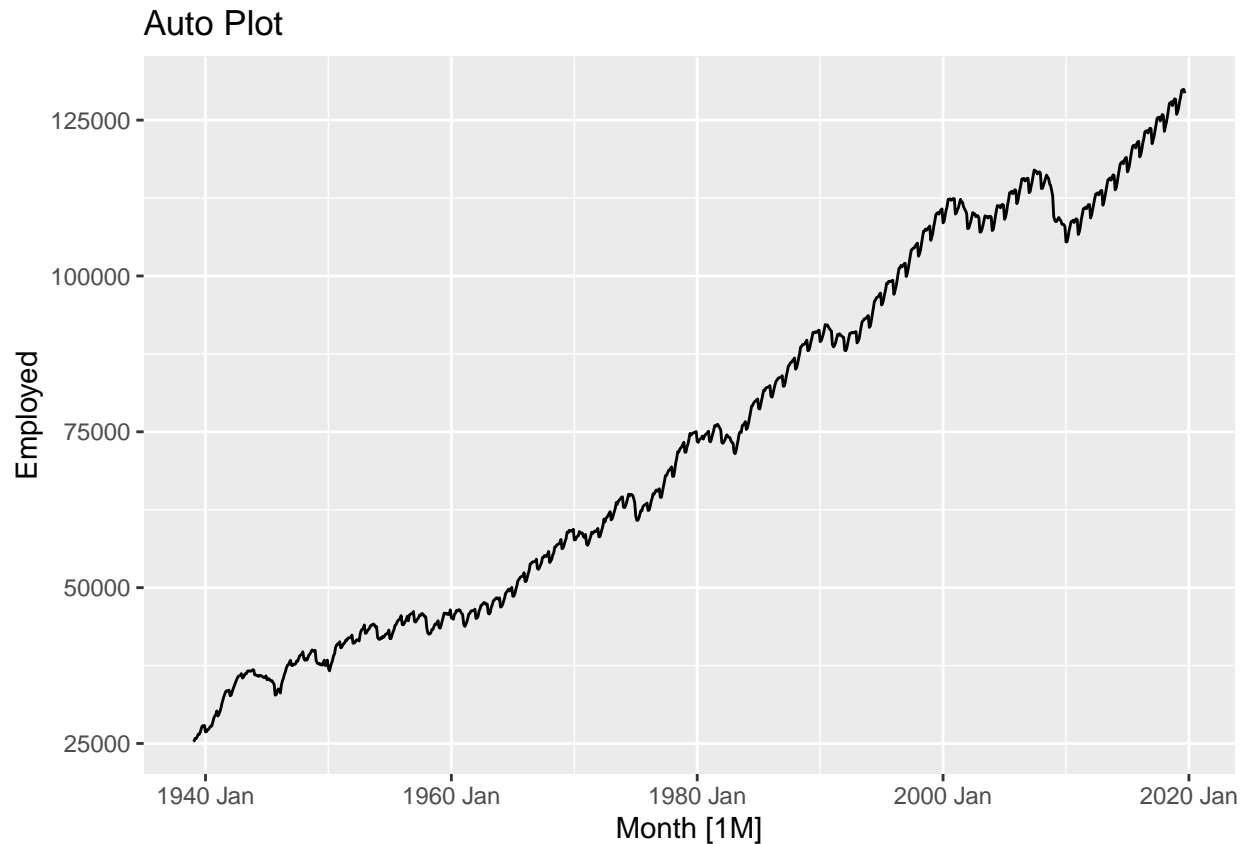
```
## # A tibble: 6 x 2
##   State      total_trips
##   <chr>         <dbl>
## 1 New South Wales 557367.
## 2 Victoria      390463.
## 3 Queensland     386643.
## 4 Western Australia 147820.
## 5 South Australia 118151.
## 6 Tasmania       54137.
```

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`.

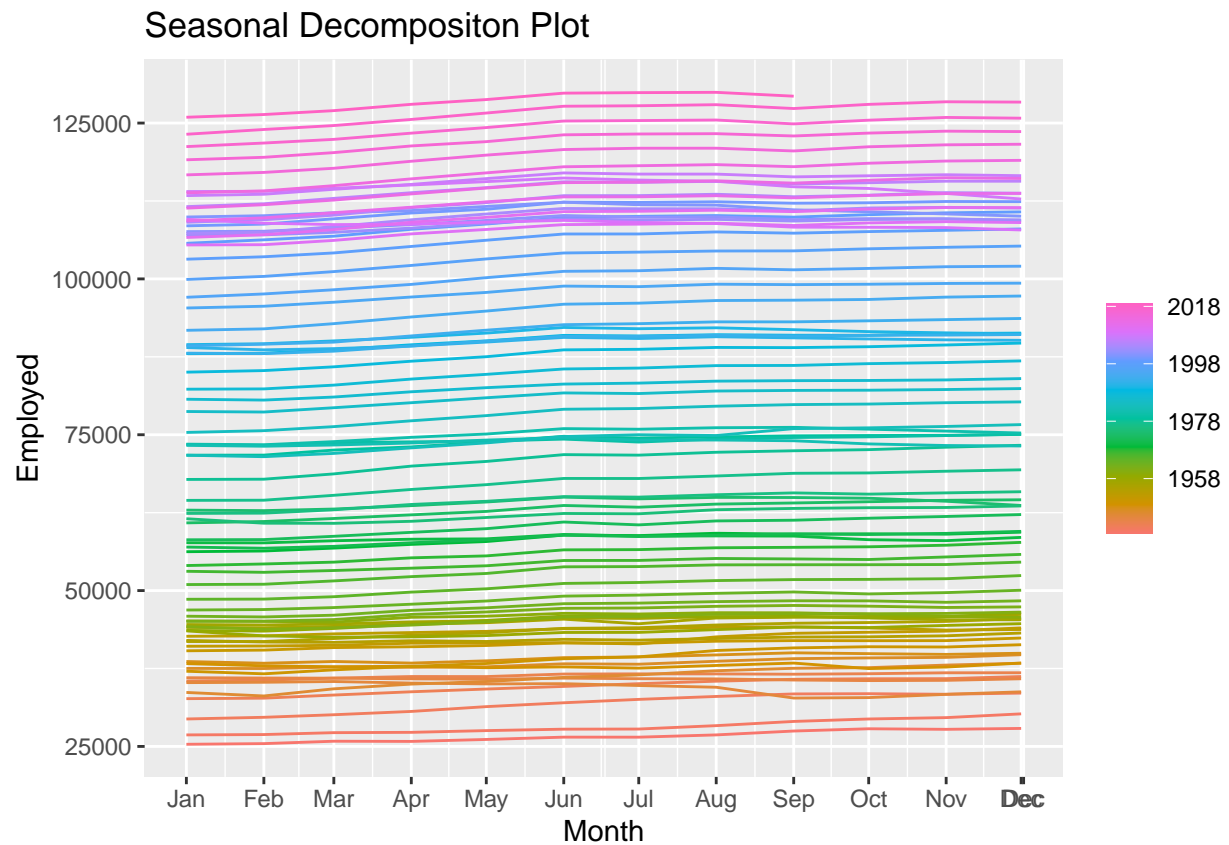
Can you spot any seasonality, cyclicity and trend? What do you learn about the series? What can you say about the seasonal patterns? Can you identify any unusual years?

“Total Private” Employed from `us_employment`

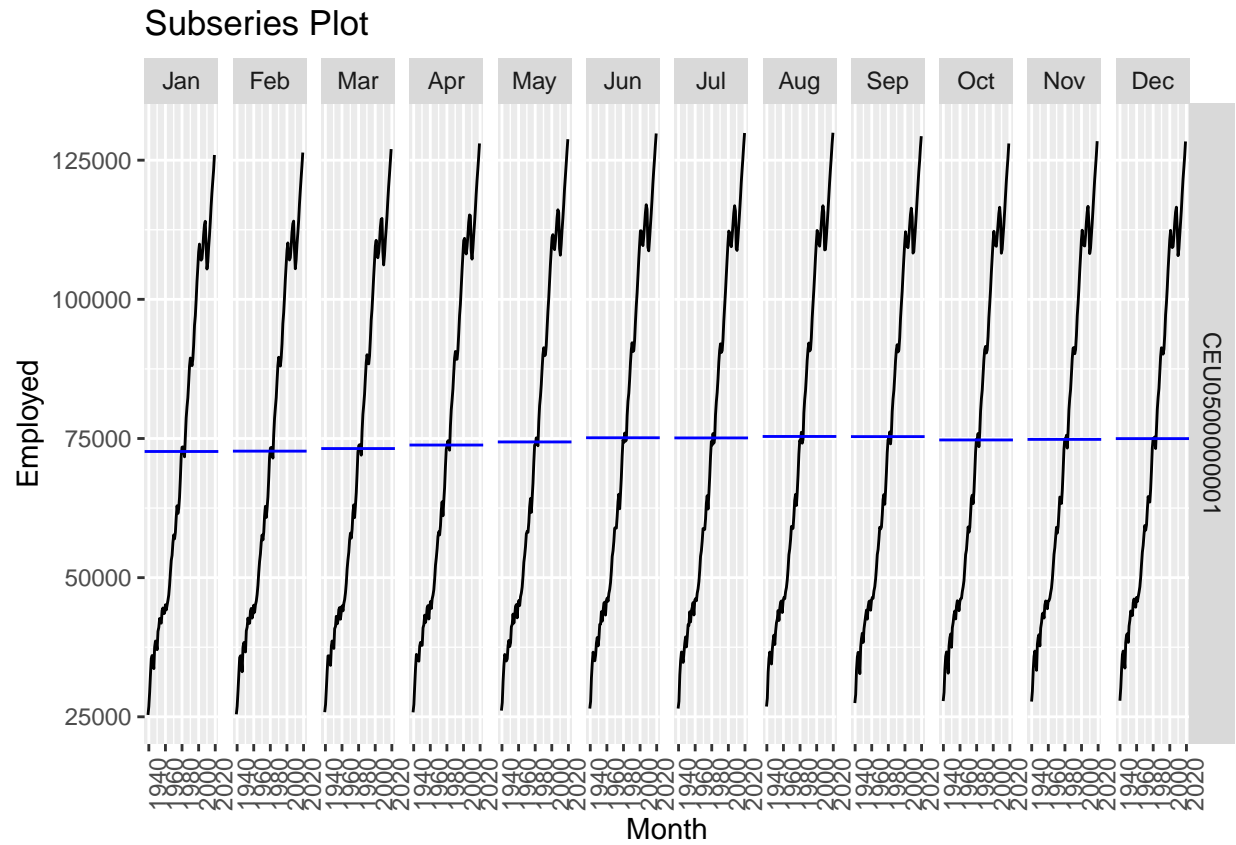
```
# This will produce a time series plot showing the number of people employed in the "Total Private" sec  
data("us_employment")  
us_employment %>%  
  filter(Title == "Total Private") %>%  
  autoplot(Employed) +  
  ggtitle("Auto Plot")
```



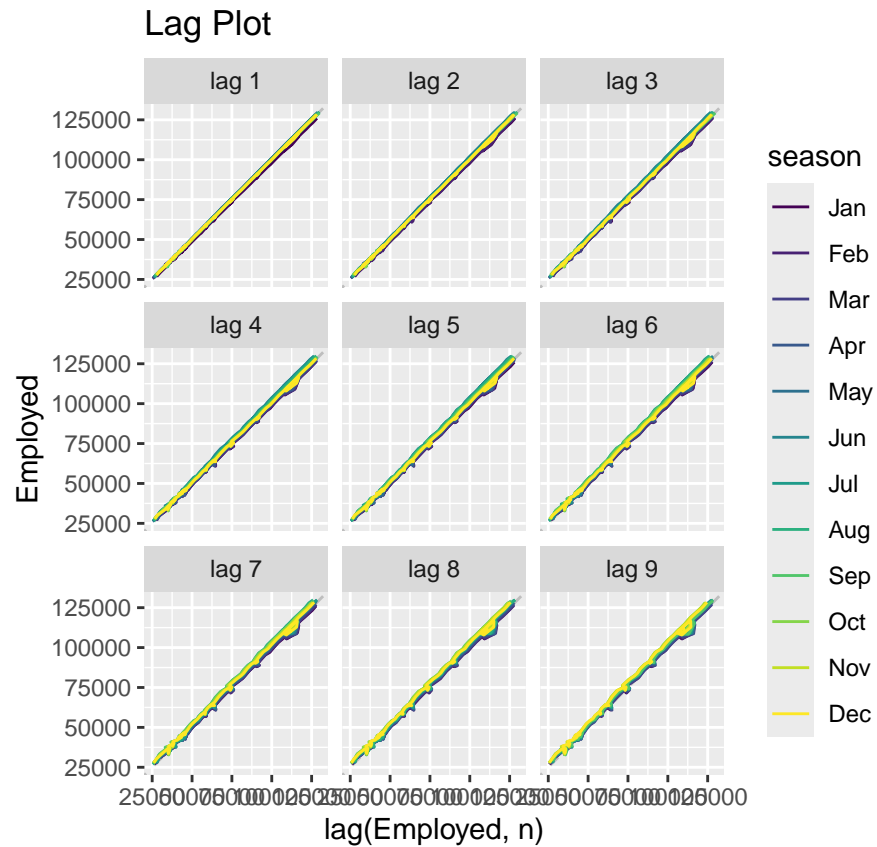
```
# This generates a seasonal decomposition plot for the Employed variable in the us_employment dataset,  
us_employment %>% filter(Title == "Total Private") %>% gg_season(Employed) +  
  ggtitle("Seasonal Decompositon Plot")
```



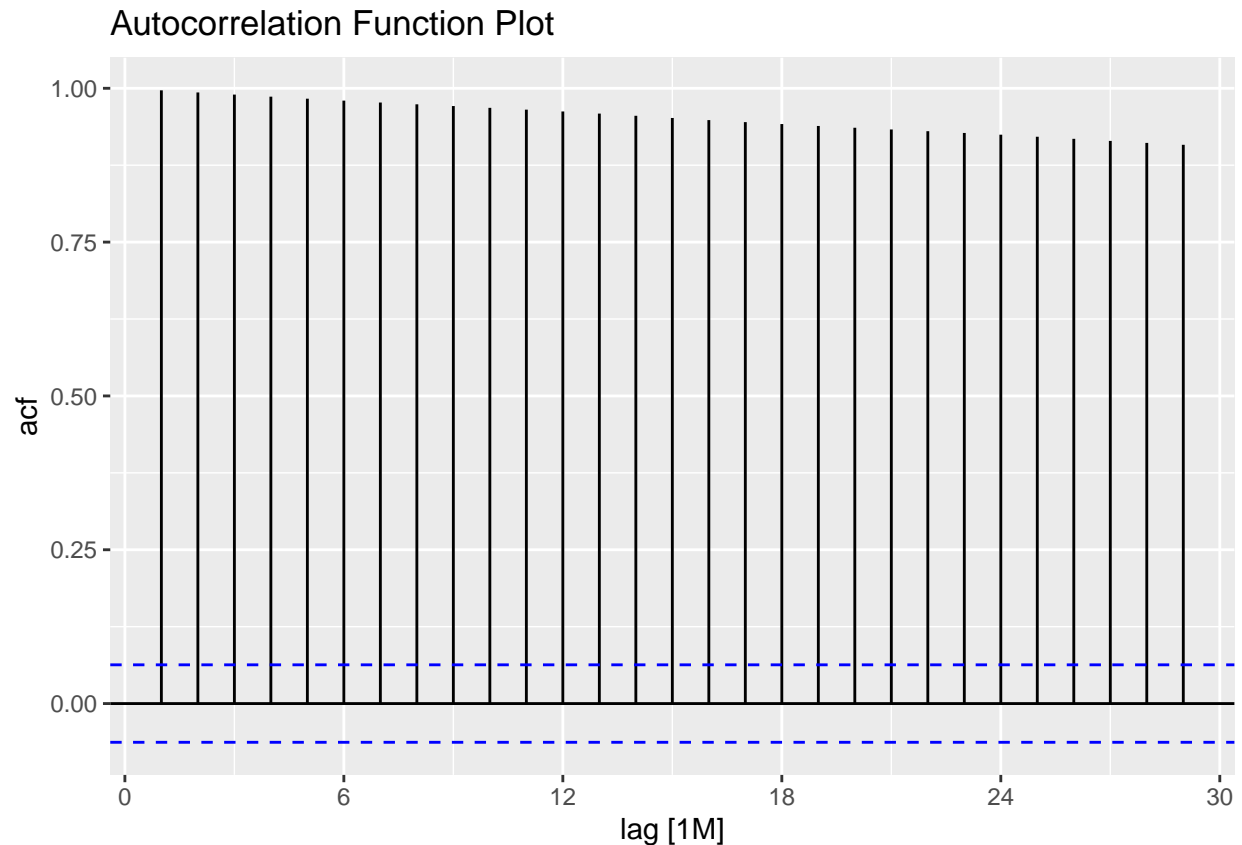
```
# This generates a subseries plot for the Employed variable in the us_employment dataset, filtered to i
us_employment %>%
  filter(Title == "Total Private") %>%
  gg_subseries(Employed) +
  ggtitle("Subseries Plot")
```



```
# This is to generate a lag plot for the "Total Private" employment data from the us_employment dataset
us_employment %>%
  filter(Title == "Total Private") %>%
  gg_lag(Employed) +
  ggtitle("Lag Plot")
```



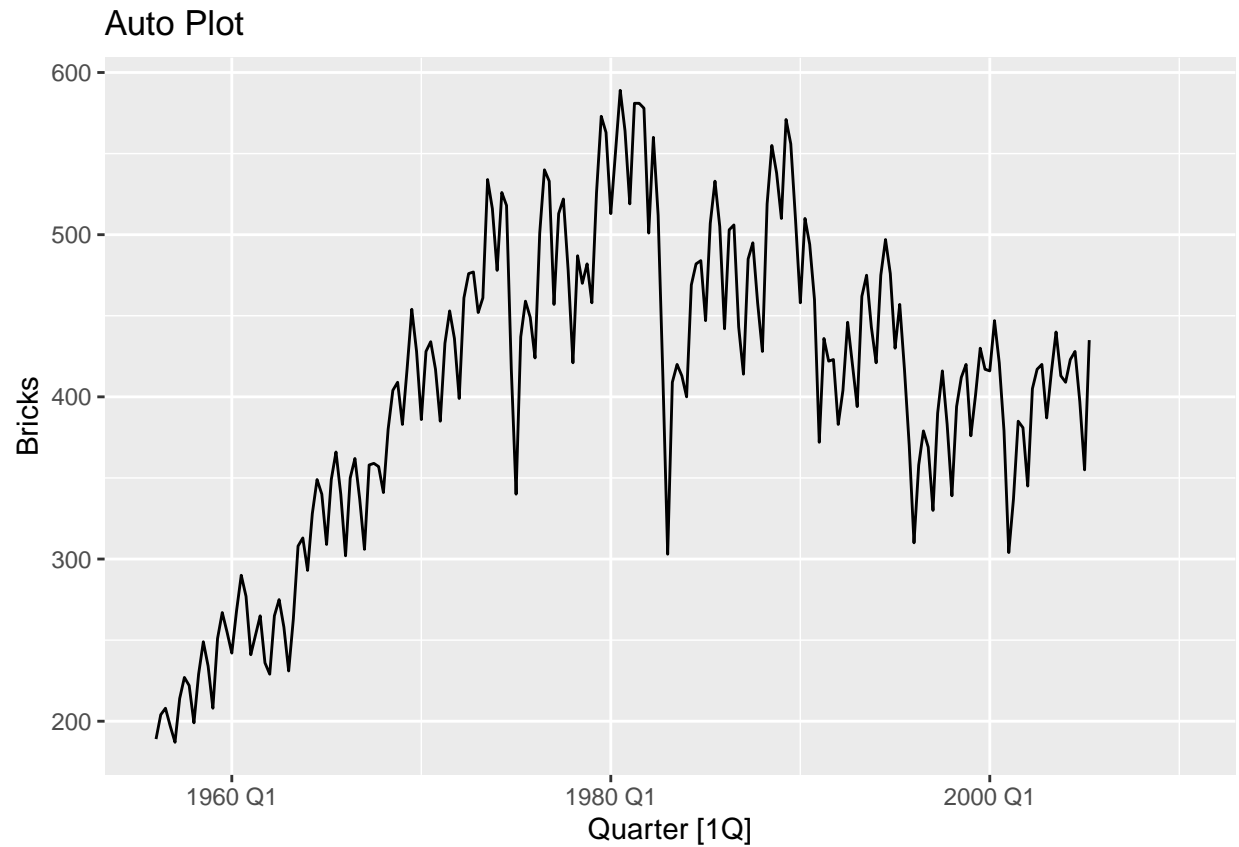
```
# This generates an Autocorrelation Function (ACF) plot for private-sector employment data using the us_employment %>%
  filter(Title == "Total Private") %>%
  ACF(Employed) %>%
  autoplot() +
  ggtitle("Autocorrelation Function Plot")
```

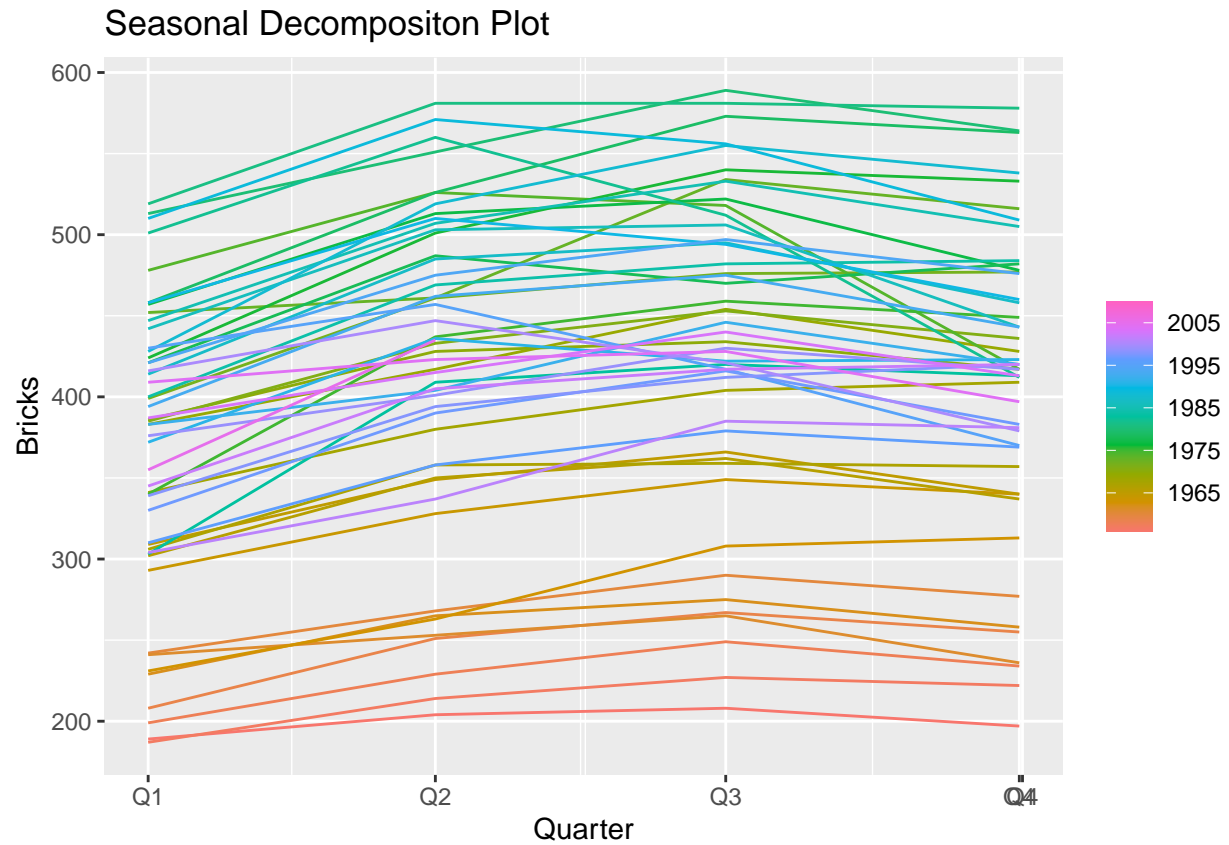
In regards to “Total Private” Employed from `us_employment`, the data shows a strong upward trend with clear seasonal fluctuations. Meaning both seasonality and trend are present. There is no evidence of cyclical patterns in the series. Employment levels steadily increase over time, with stable seasonal patterns throughout the year. However, a slight rise occurs in June, which tapers off after December. Anomalies were observed in the years 2008/2010.

Bricks from `aus_production`

```
# This will produce a time series plot showing the number of people employed in the "Total Private" sec
data("us_employment")
data("aus_production")
aus_production %>%
  autoplot(Bricks) +
  ggtitle("Auto Plot")
```

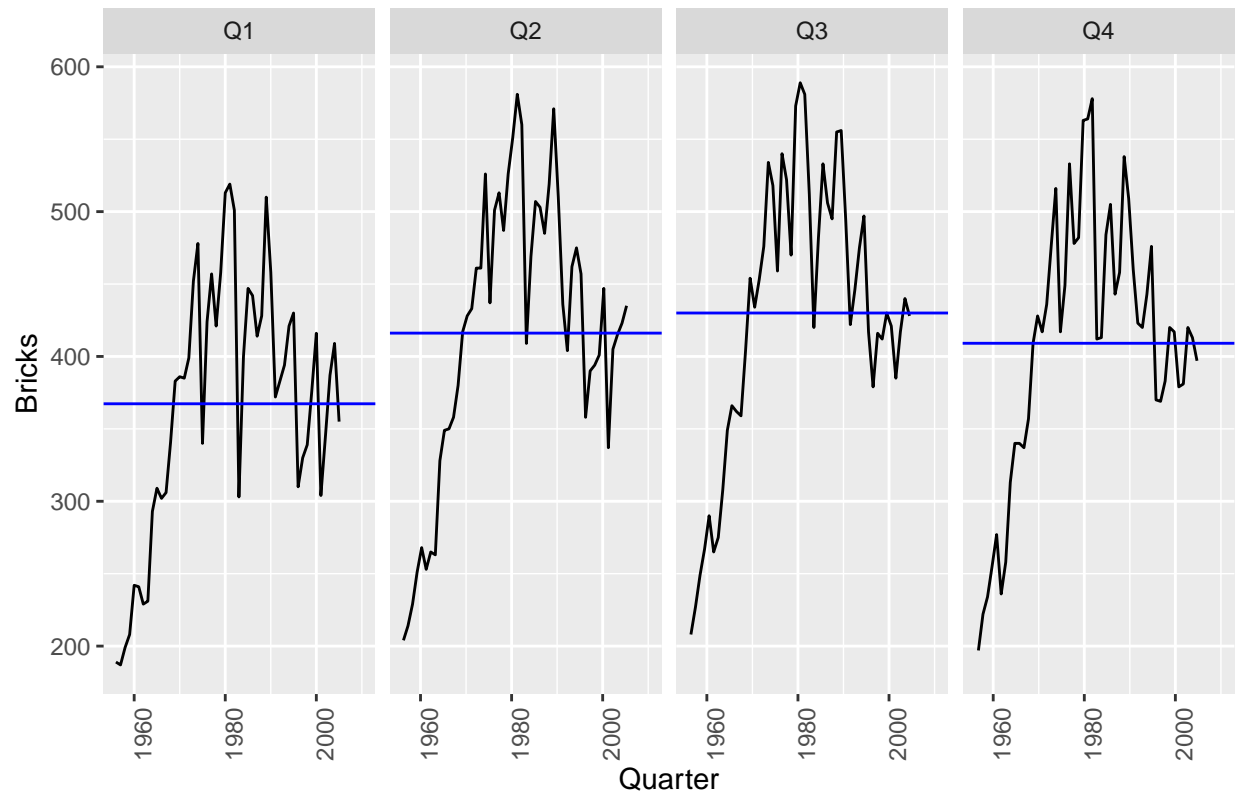


```
# This generates a seasonal decomposition plot for the Bricks variable from the aus_production dataset.  
aus_production %>%  
  gg_season(Bricks) +  
  ggtitle("Seasonal Decompositon Plot")
```

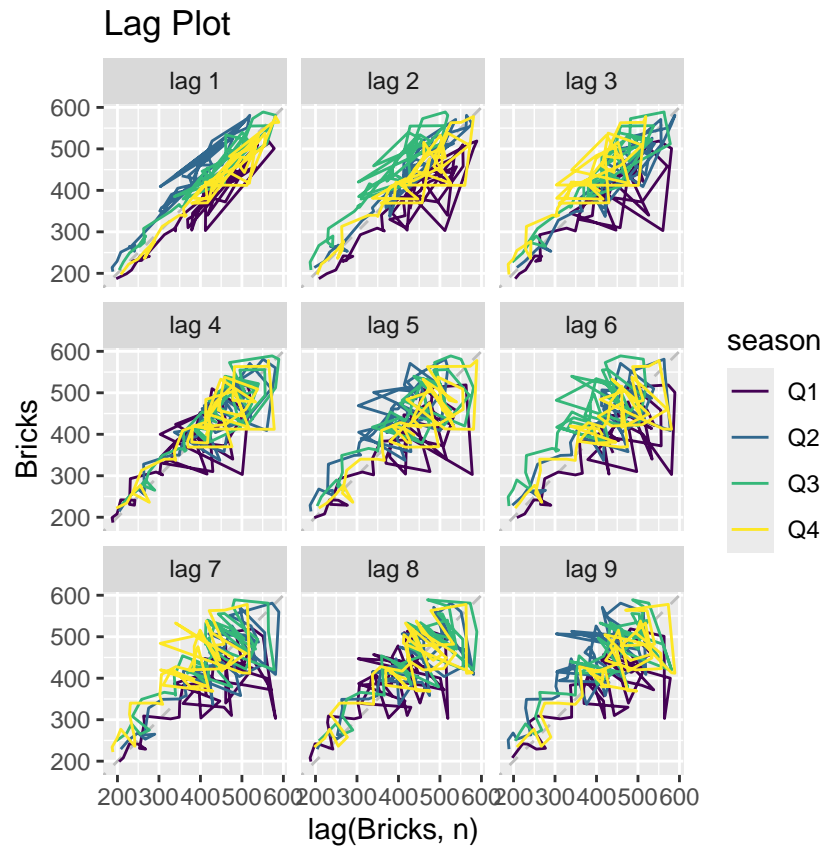


```
# This generates a subseries plot for the Bricks variable from the aus_production dataset.
aus_production %>%
  gg_subseries(Bricks) +
  ggtitle("Subseries Plot")
```

Subseries Plot

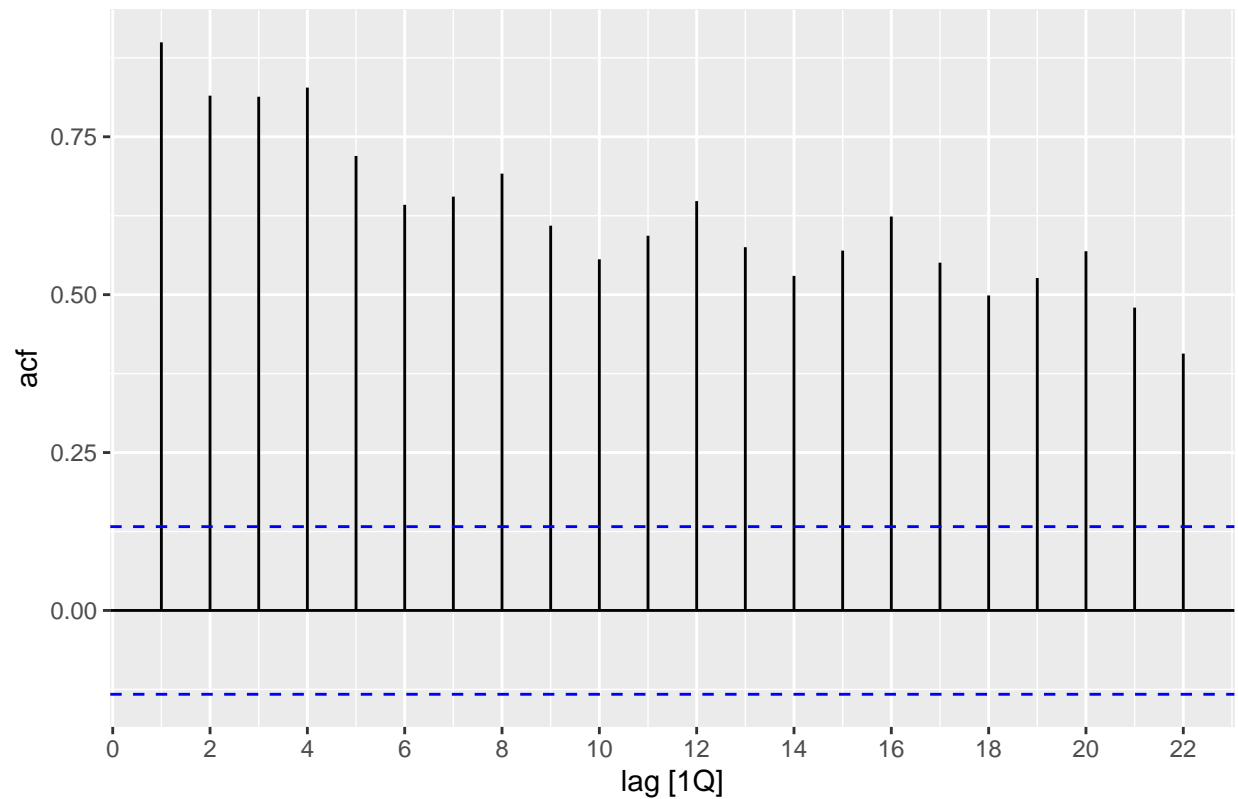


```
# This generates a lag plot for the Bricks variable from the aus_production dataset.
aus_production %>%
  gg_lag(Bricks) +
  ggtitle("Lag Plot")
```



```
# This generates an Autocorrelation Function (ACF) plot for the Bricks variable from the aus_production
aus_production %>%
  ACF(Bricks) %>%
  autoplot() +
  ggtitle("Autocorrelation Function Plot")
```

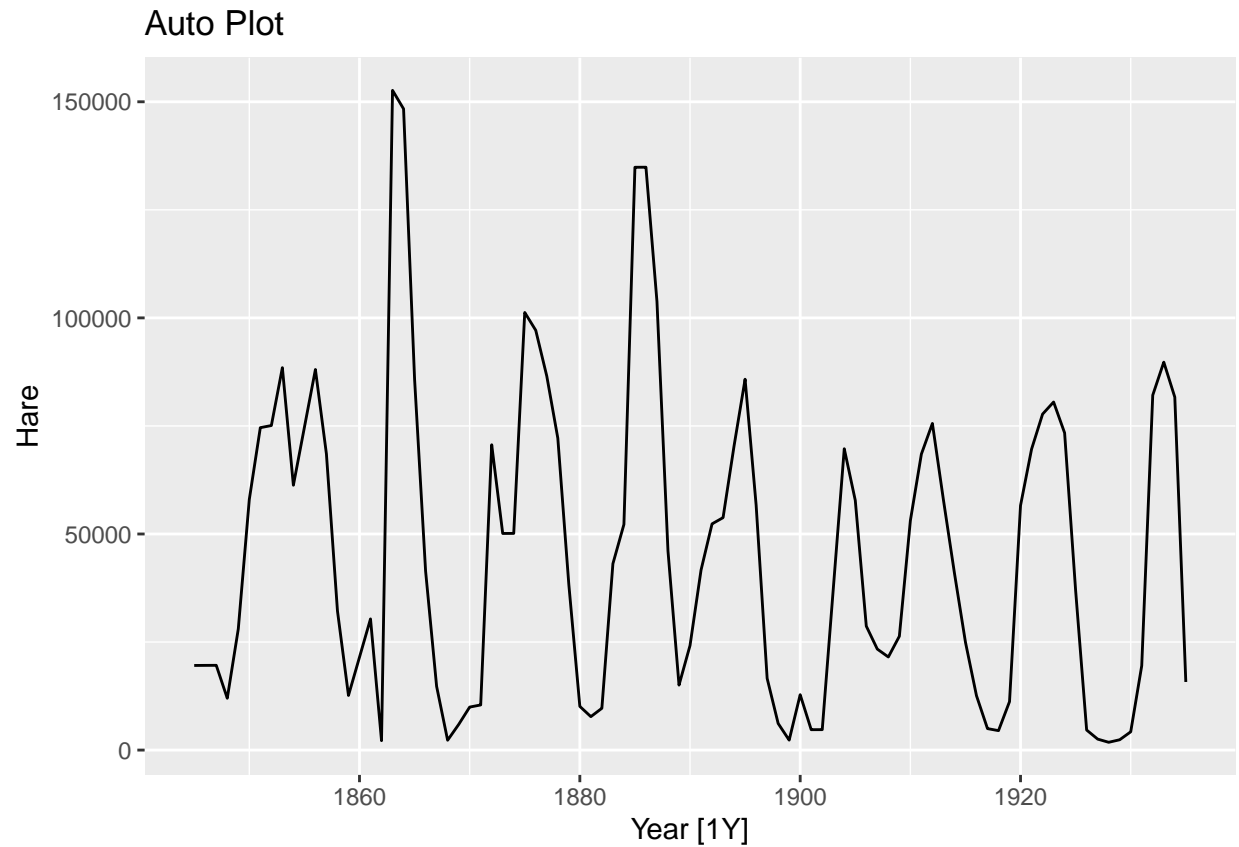
Autocorrelation Function Plot



In regards to Bricks from `aus_production`, the data reveals significant seasonality each year and shows clear cyclic behavior with a 40-year period. No noticeable trend is present over this timeframe. Brick production rises from the first to the third quarter, then declines in the fourth quarter. A marked drop in brick production is evident around 1983.

Hare from Pelt

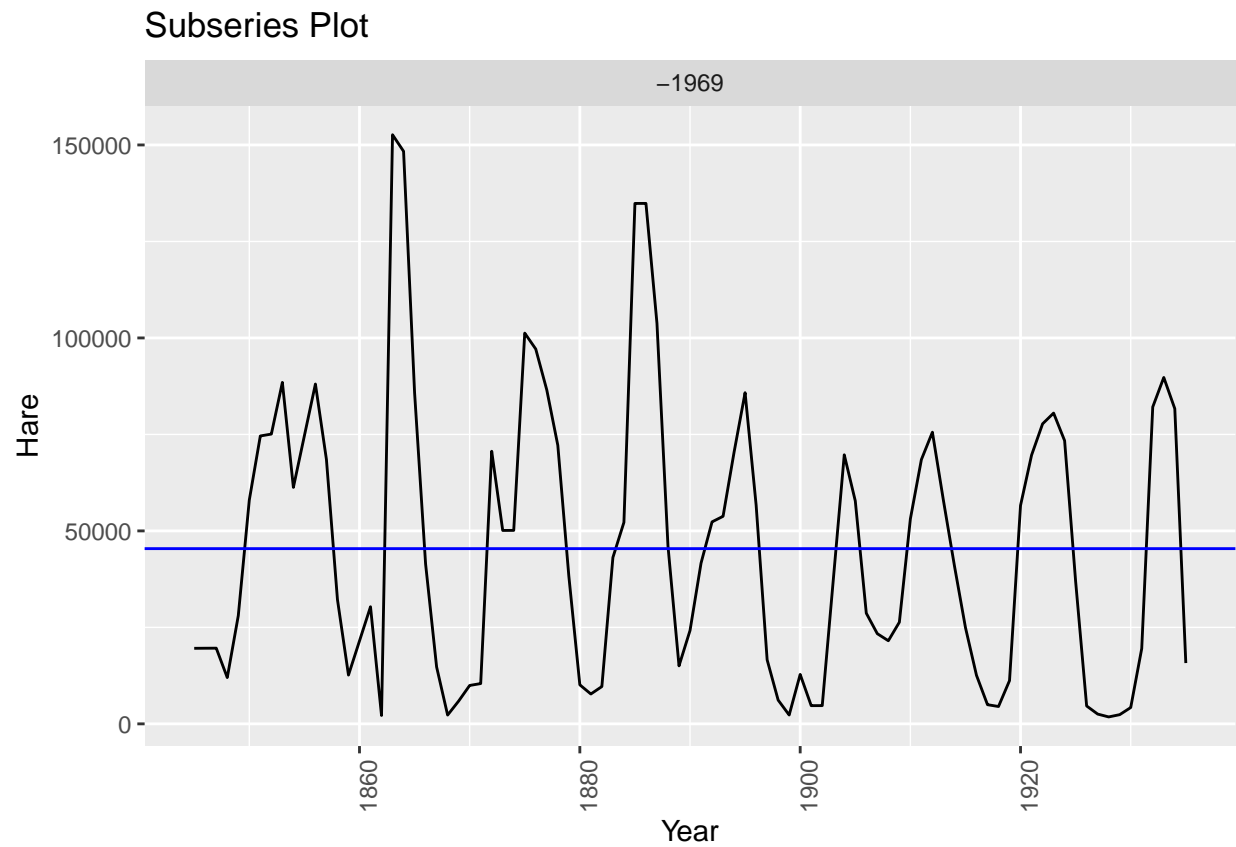
```
# This generates a time series plot for the Hare variable from the pelt dataset.
data("pelt")
pelt %>%
  autoplot(Hare) +
  ggtitle("Auto Plot")
```



For `gg_season`: The data must contain at least one observation per seasonal period.

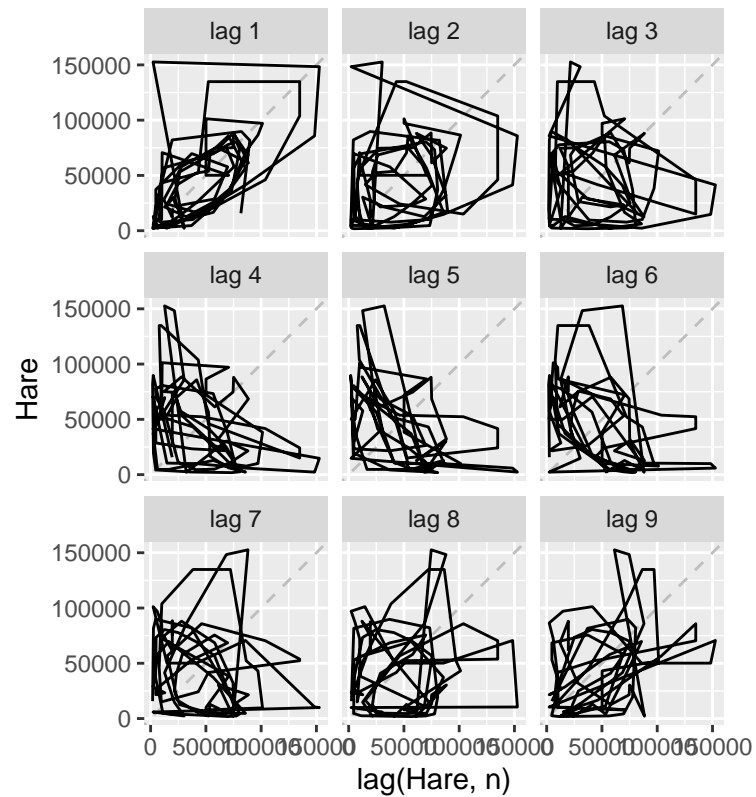
```
#pelt %>%  
  #gg_season(Hare)  
  #ggtitle("Seasonal Decomposition")
```

```
# This generates a subseries plot for the Hare variable from the pelt dataset.  
pelt %>%  
  gg_subseries(Hare)+  
  ggtitle("Subseries Plot")
```



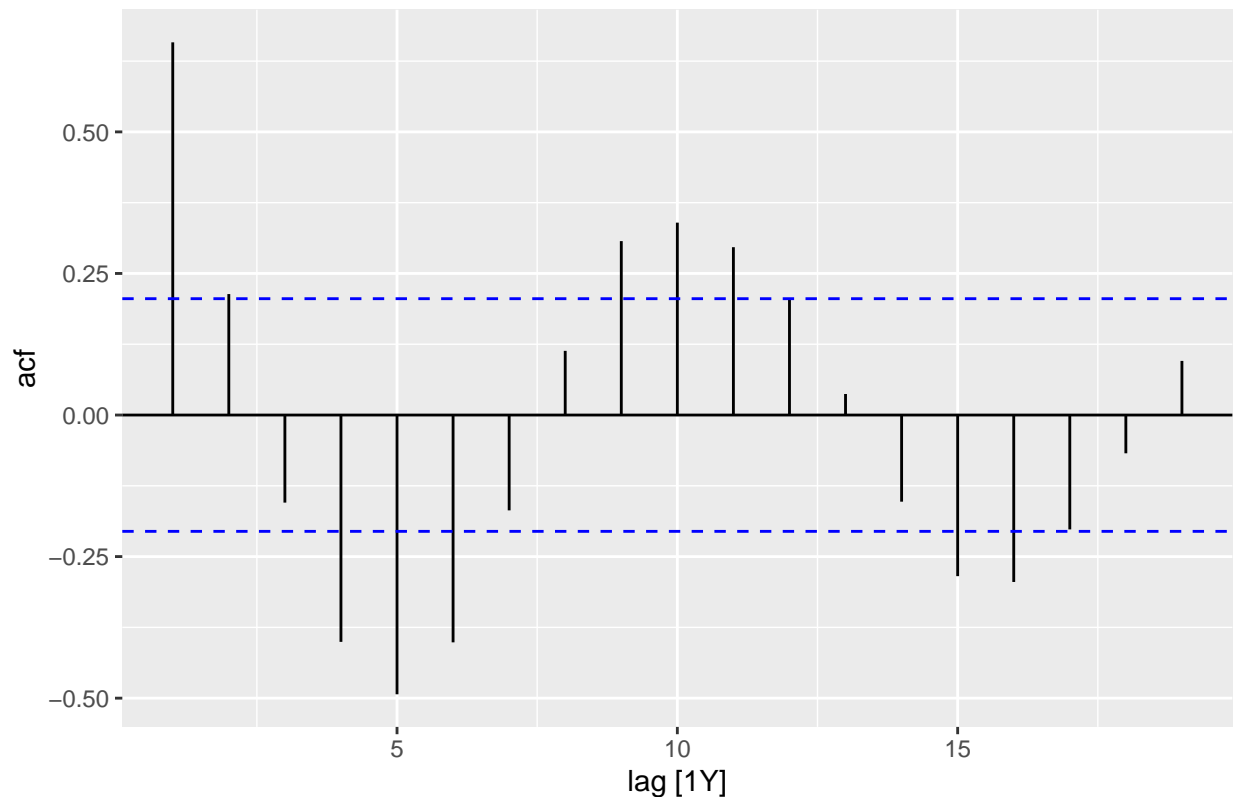
```
# This generates a lag plot for the Hare variable from the pelt dataset.  
pelt %>%  
  gg_lag(Hare) +  
  ggtitle("Lag Plot")
```


Lag Plot



```
# This generates an Autocorrelation Function (ACF) plot for the Hare variable from the pelt dataset.
pelt %>%
  ACF(Hare) %>%
  autoplot() +
  ggtitle("Autocorrelation Function Plot")
```

Autocorrelation Function Plot

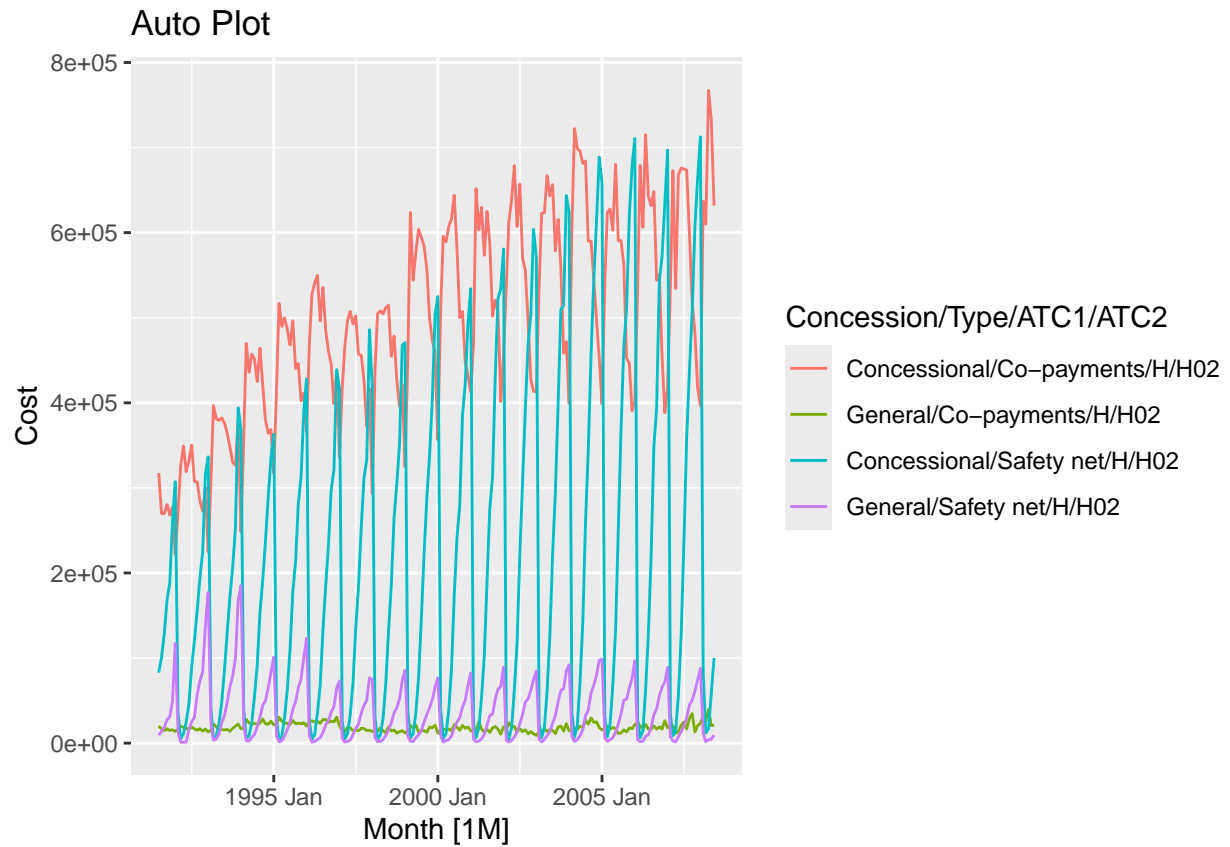


In regards to Hare from pelt, the data demonstrates cyclical behavior with no apparent trend or clear seasonality. The ACF plot indicates a recurring cycle approximately every ten years. The trend shows notable fluctuations over time. There is a significant drop before the 1860s. A marked increase occurs after 1860. There is another drop after the mid-1860s. An increase is observed before 1870. These fluctuations appear to follow a pattern of varying significantly approximately every 5 years. The significant drops and increases at these points, especially around the 1860s, can be considered unusual or noteworthy.

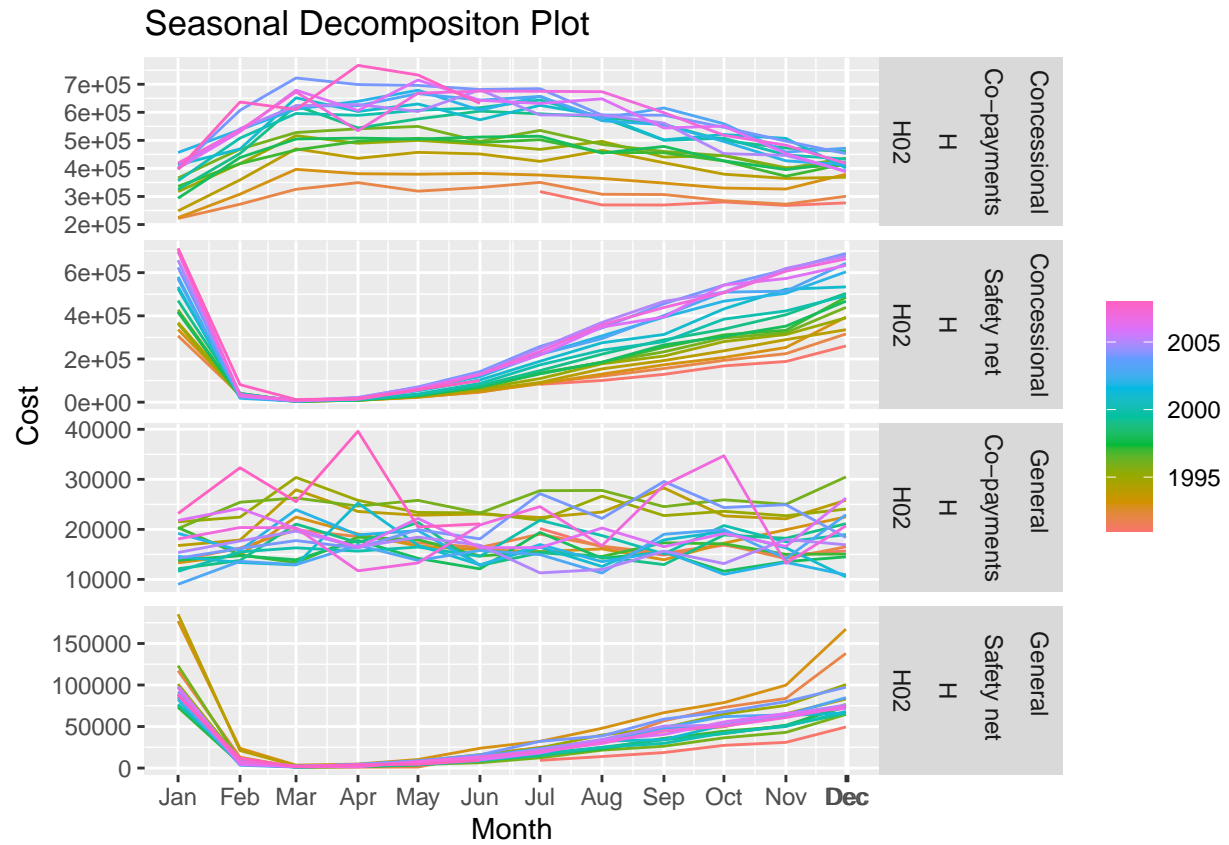
Please note for Seasonal Decomposition Plot, the data must contain at least one observation per seasonal period. In which it lacks. For a Seasonal Decomposition Plot to be effective, the dataset needs to include at least one data point for each season or period within the cycle. However, the current dataset lacks this requirement, meaning there are not enough observations to properly capture or analyze the seasonal patterns. Without sufficient data for each season, the decomposition may not yield accurate or meaningful results.

“H02” Cost from PBS

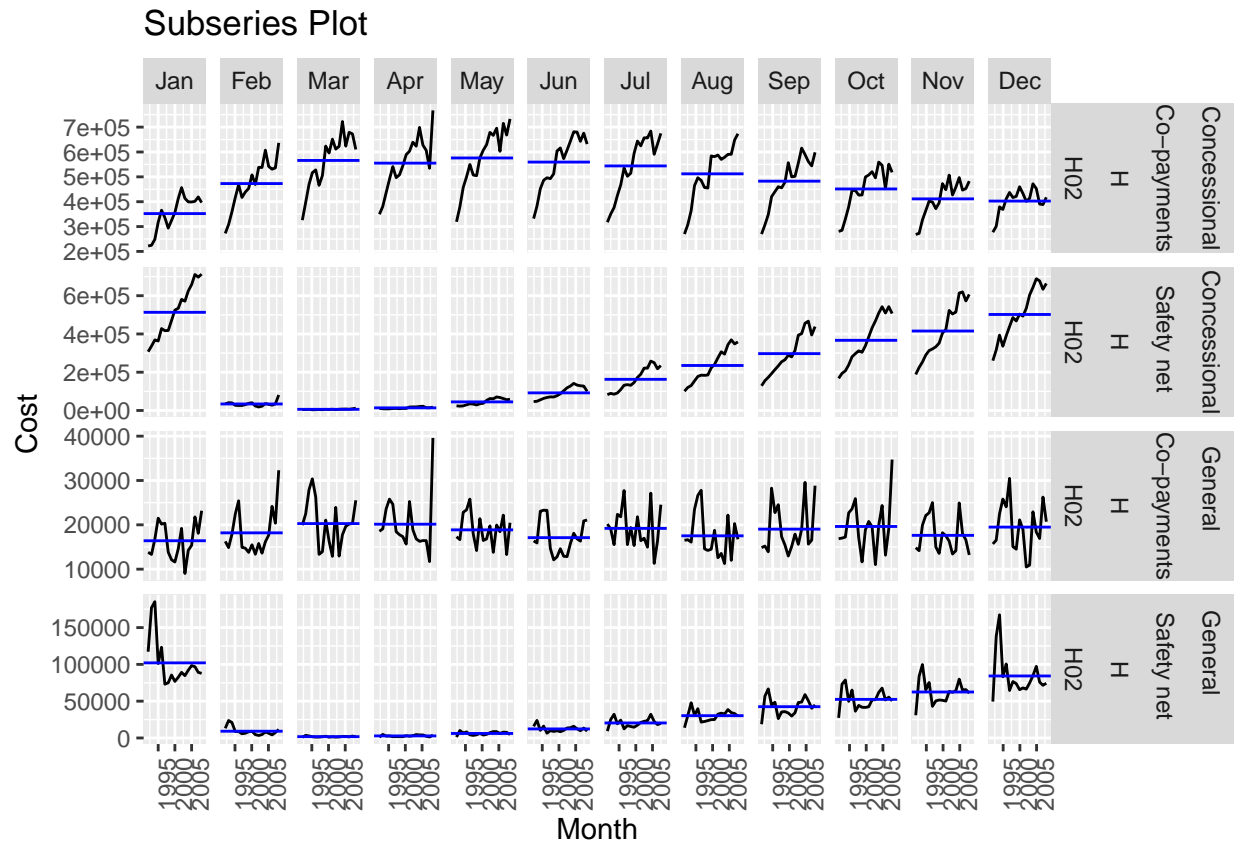
```
# This generates a time series plot for the Cost variable within the H02 category from the PBS dataset.
h_02 <- PBS %>% filter(ATC2 == "H02")
h_02 %>% autoplot(Cost) +
  ggtitle("Auto Plot")
```



```
# This generates a seasonal decomposition plot for the Cost variable within the H02 category from the h_02
h_02 %>% gg_season(Cost) +
  ggtitle("Seasonal Decompositon Plot")
```



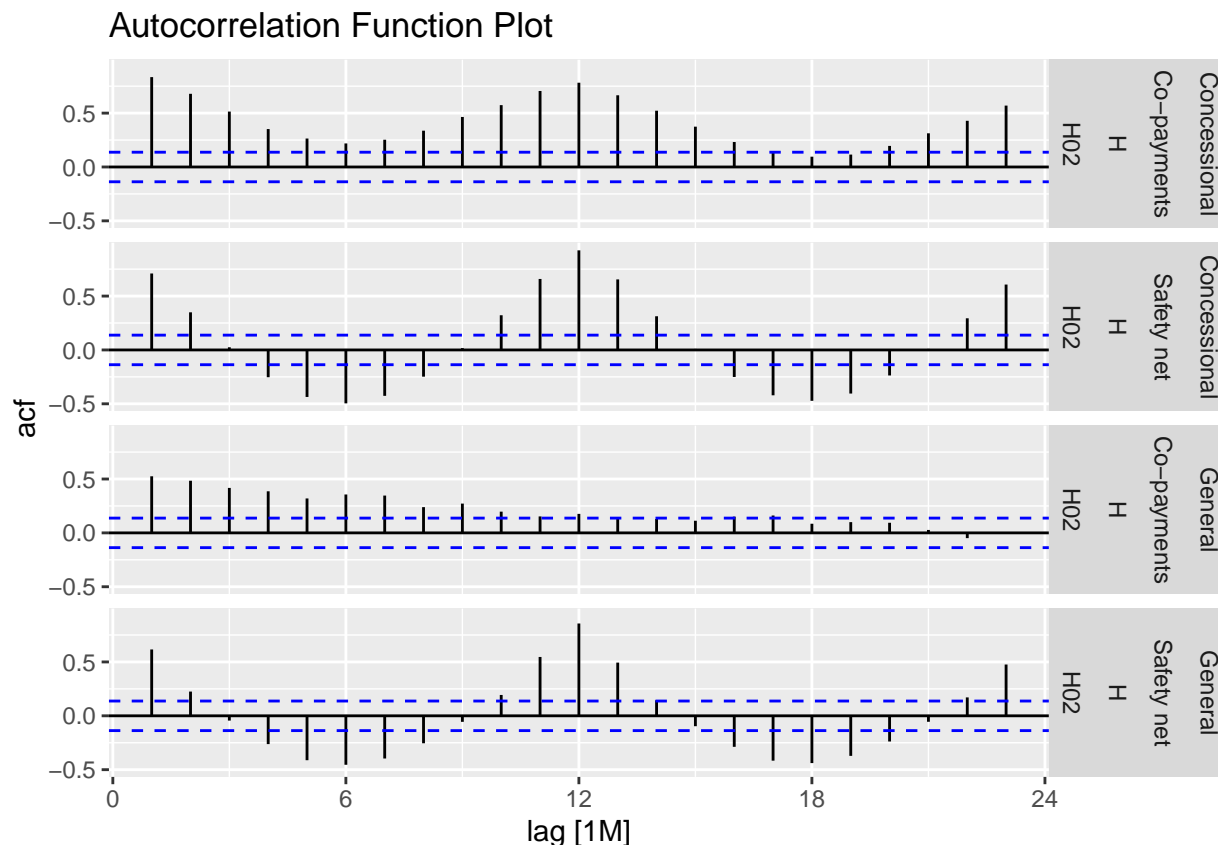
```
# This generates a subseries plot for the Cost variable within the H02 category from the h_02 dataset.
h_02 %>% gg_subseries(Cost) +
  ggtitle("Subseries Plot")
```



The data provided to contains more than one time series. Please filter a single time series to use 'gg_lag()

```
#h_02 %>% gg_lag(Cost) +  
  #ggtitle("Lag Plot")
```

```
# This generates an Autocorrelation Function (ACF) plot for the Cost variable within the H02 category f  
h_02 %>%  
  ACF(Cost) %>%  
  autoplot() +  
  ggtitle("Autocorrelation Function Plot")
```



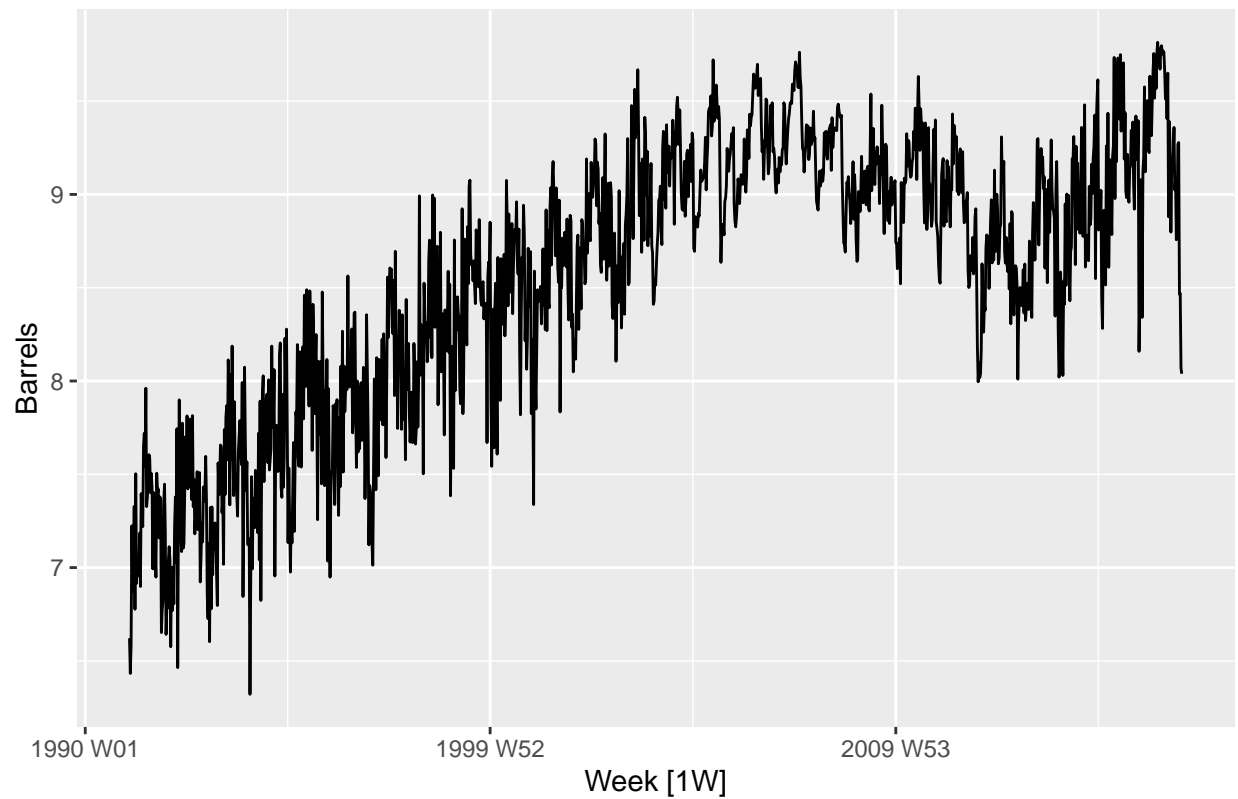
In regards to “H02” Cost from PBS, The data displays strong annual seasonality and distinct cyclic behavior, but no noticeable trend. From 1995 to 2005, each month shows an upward fluctuation for Concessional Co-Payments, General Co-Payments, and Concessional/Safety Co-Payments. However, General Safety Co-Payments experienced a significant drop in the first few months of 1995, followed by fluctuations until 2005. This period, especially the drop in early 1995 and the fluctuations throughout the rest of the decade, can be considered unusual or noteworthy.

Please note in regards to Lag Plot, the data provided needs to contains more than one time series. Please filter a single time series to use ‘gg_lag()’. In which this data lacks. A Lag Plot is typically used to identify patterns or relationships within a single time series by plotting values at different time lags. However, the provided data contains more than one time series, which is not suitable for this type of analysis. To use the gg_lag() function, a single time series needs to be selected or filtered from the dataset. Since the current data lacks a singular time series, it cannot be used directly for creating a lag plot until a single time series is isolated.

Barrels from us_gasoline

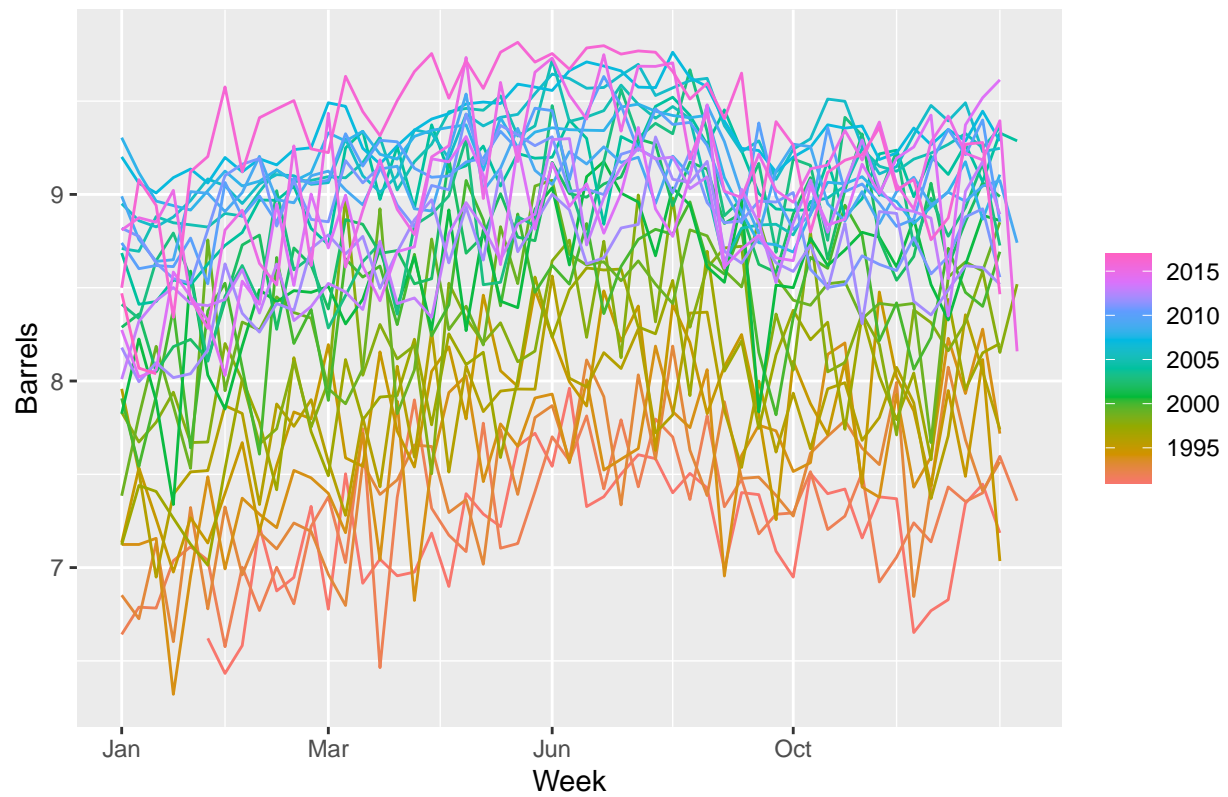
```
# This generates a time series plot for the us_gasoline dataset.
data("us_gasoline")
us_gasoline %>%
  autoplot() +
  ggtitle("Autoplot")
```

Autoplot



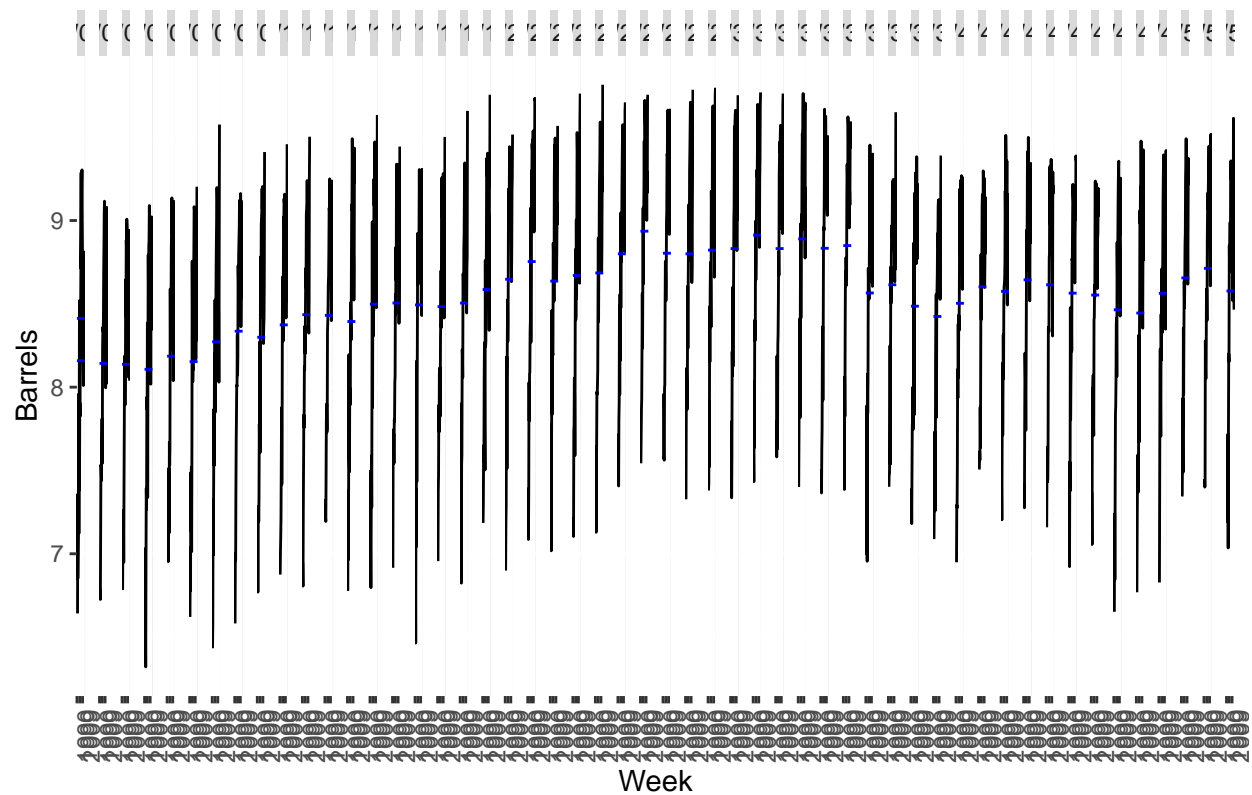
```
# This generates a seasonal decomposition plot for the us_gasoline dataset.  
us_gasoline %>%  
  gg_season() +  
  ggtitle("Seasonal Decomposition Plot")
```

Seasonal Decomposition Plot

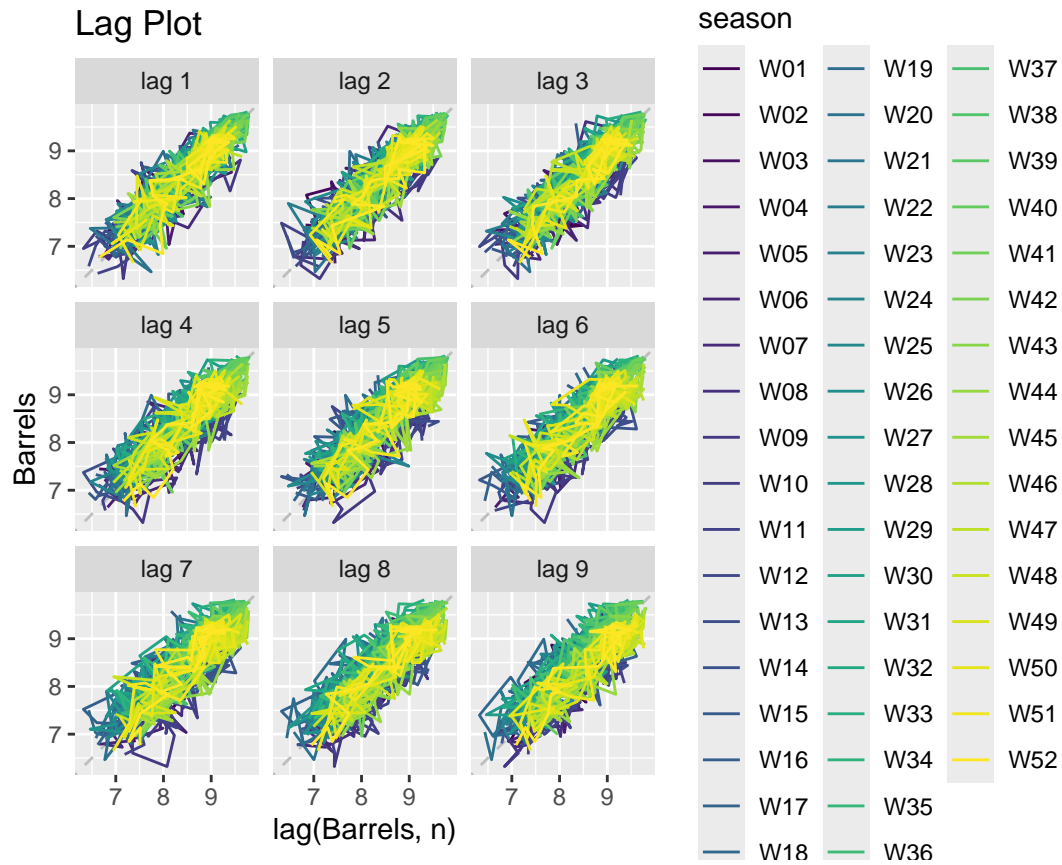


```
# This generates a subseries plot for the us_gasoline dataset.  
us_gasoline %>%  
  gg_subseries()+  
  ggtitle("Subseries Plot")
```


Subseries Plot

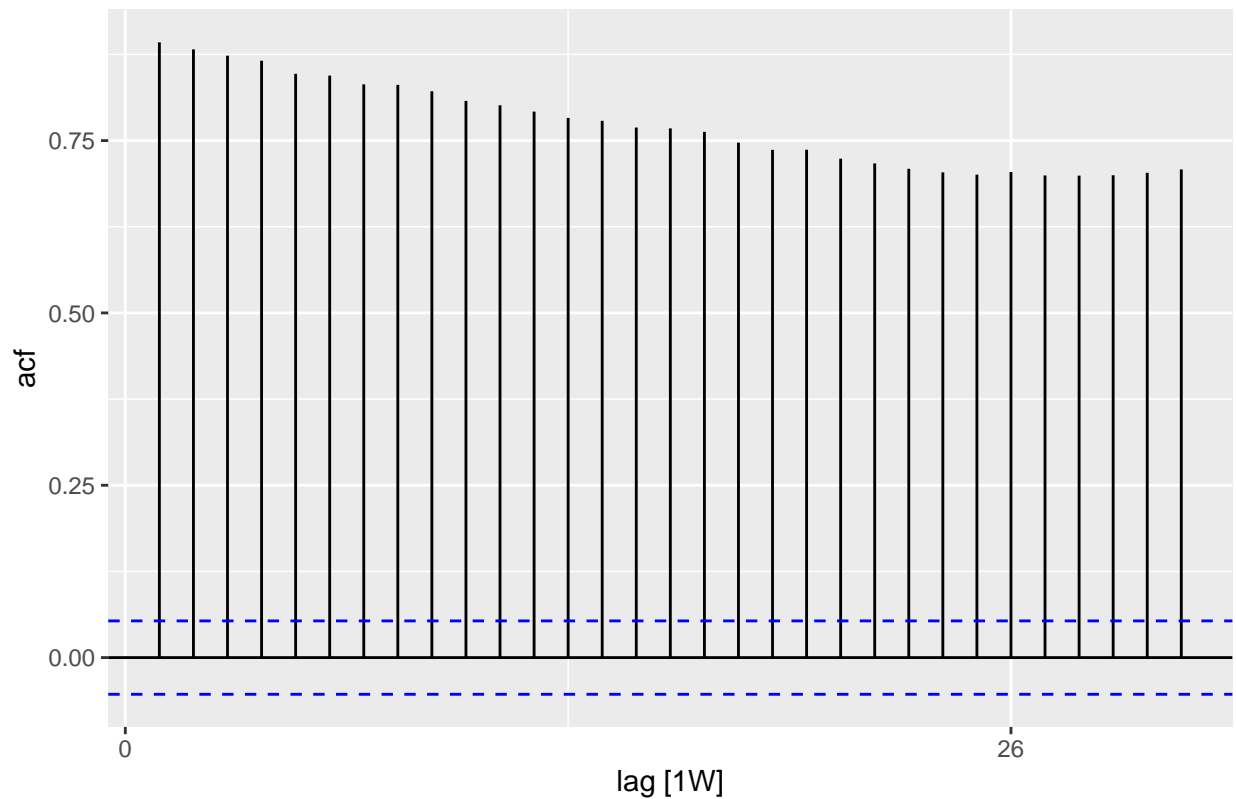


```
# This generates a lag plot for the us_gasoline dataset.
us_gasoline %>%
  gg_lag() +
  ggtitle("Lag Plot")
```



```
# This generates a lag plot for the us_gasoline dataset.
us_gasoline %>%
  ACF() %>%
  autoplot() +
  ggtitle("Autocorrelation Function Plot")
```

Autocorrelation Function Plot



In regards to Barrels from us_gasoline, There is no trend, seasonality, or cyclic behavior in the data. It shows random fluctuations that seem unpredictable, with no clear patterns to support the creation of a forecasting model. From 1995 to 2015, the data shows an increase in fluctuation over time. There is a slight downward trend from 2008 to after 2009, followed by an increase again. These periods of increased fluctuation and the specific downturn from 2008 to after 2009 could be considered unusual or noteworthy years.