Assignment 1_data624

Heleine Fouda

2024-09-08

Loading the required libraries

Reference book:

Forecasting: Principles and Practices,3rd Edition by Rob J Hyndman and George Athanasopoulos

Exercise 1

Quarterly production of selected commodities in Australia (aus_production dataset): Time interval: A quarter

This dataset describes quarterly estimates of selected indicators of manufacturing production in Australia. **Format** Time series of class tsibble. Details 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.

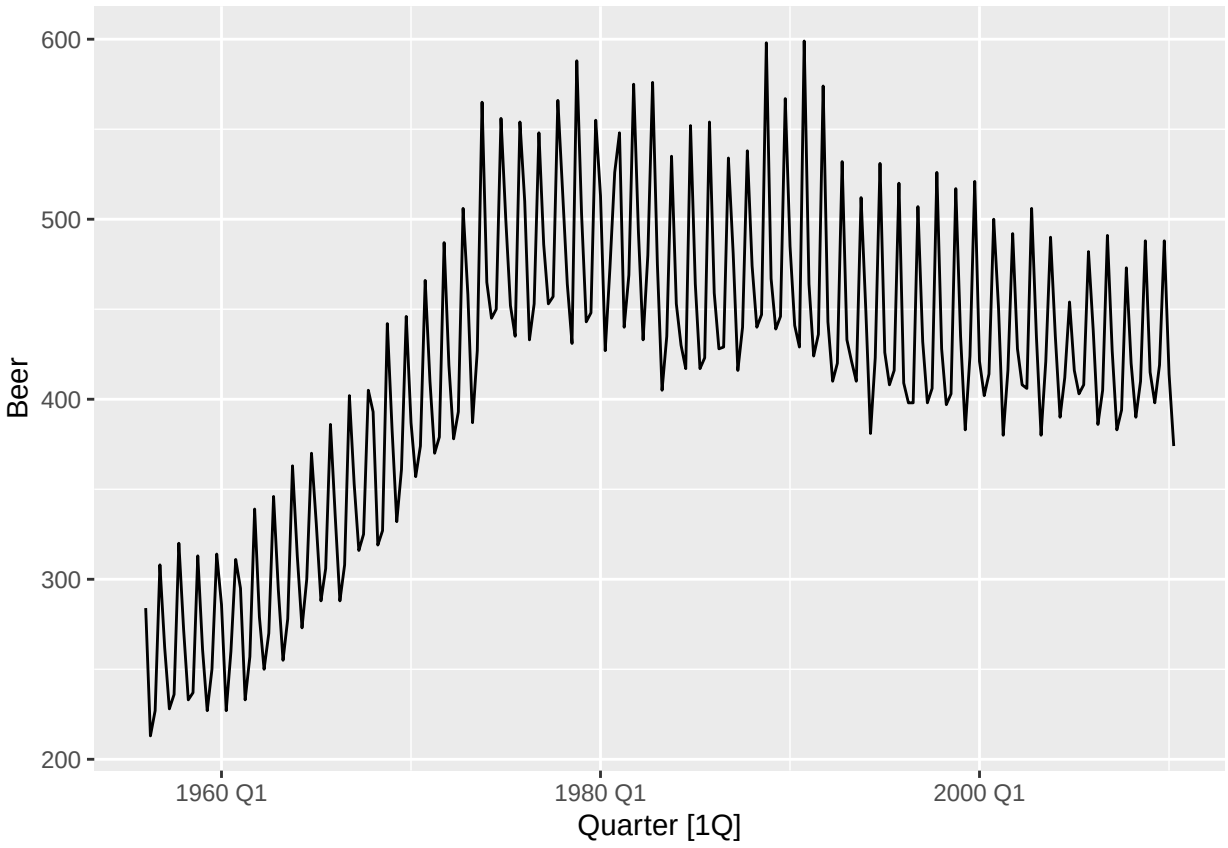
Source: Australian Bureau of Statistics, catalogue number 8301.0.55.001 table 1.

```
aus_production
```

```
## # A tsibble: 218 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
##  7 1957 Q3   236    5317    227    603        5259     7
##  8 1957 Q4   320    6152    222    582        4735     6
##  9 1958 Q1   272    5758    199    554        4608     5
## 10 1958 Q2   233    5641    229    620        5196     7
## # i 208 more rows
```

```
autoplot(aus_production )
```

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



Pelt trading records(pelt): Time interval: a year

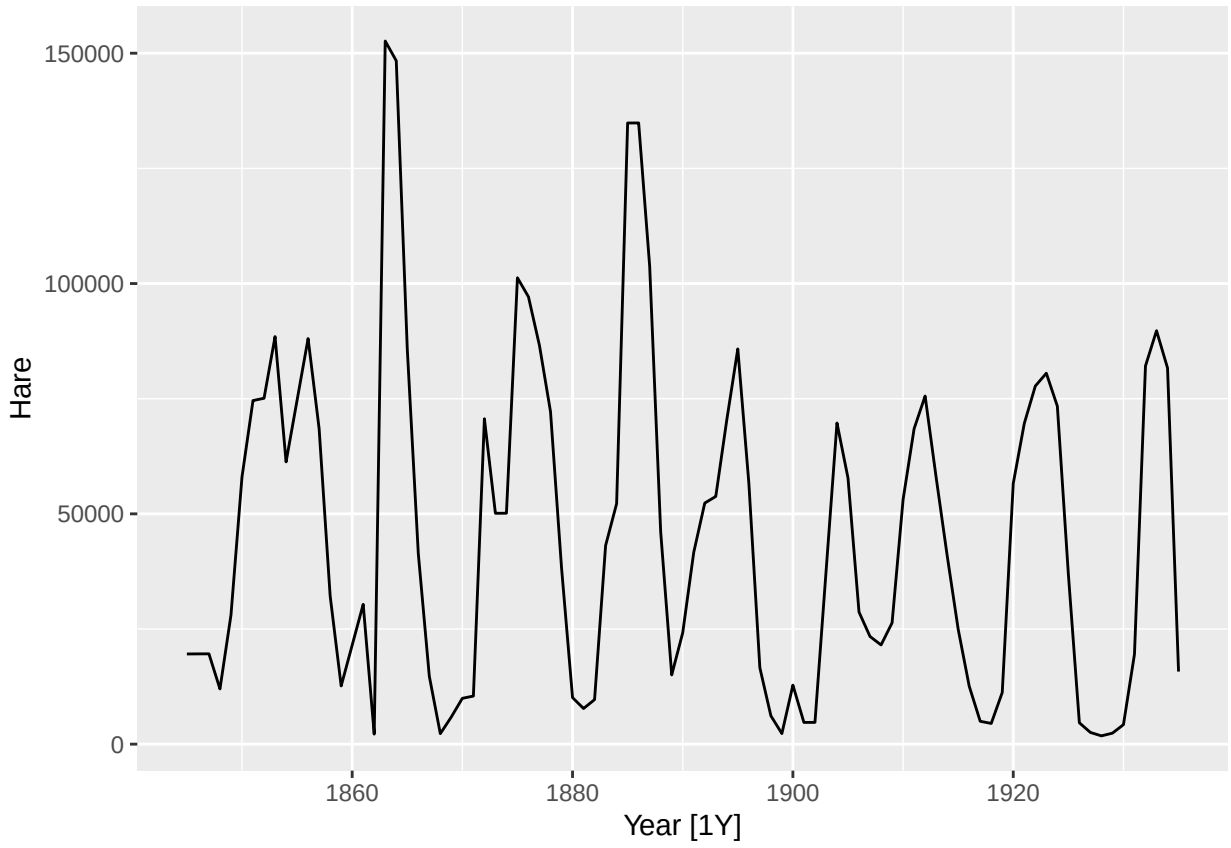
This dataset contains pelt trading records from Hudson Bay Company trading records for Snowshoe Hare and Canadian Lynx furs from 1845 to 1935. This data contains trade records for all areas of the company. **Format** Time series of class tsibble Details pelt is an annual tsibble with two values: Hare: The number of Snowshoe Hare pelts traded. Lynx: The number of Canadian Lynx pelts traded. **Source** Hudson Bay Company

```
pelt
```

```
## # A tsibble: 91 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
##  7  1851 74600  5560
##  8  1852 75090  5080
##  9  1853 88480 10170
## 10  1854 61280 19600
## # i 81 more rows
```

```
autoplot(pelt)
```

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



GAFA stock prices (gafa_stock): Time interval: a Day

This dataset contains Historical stock prices from 2014-2018 for Google, Amazon, Facebook and Apple. All prices are in $USD.

Format Time series of class tsibble

Details 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. **Source** Yahoo Finance historical data

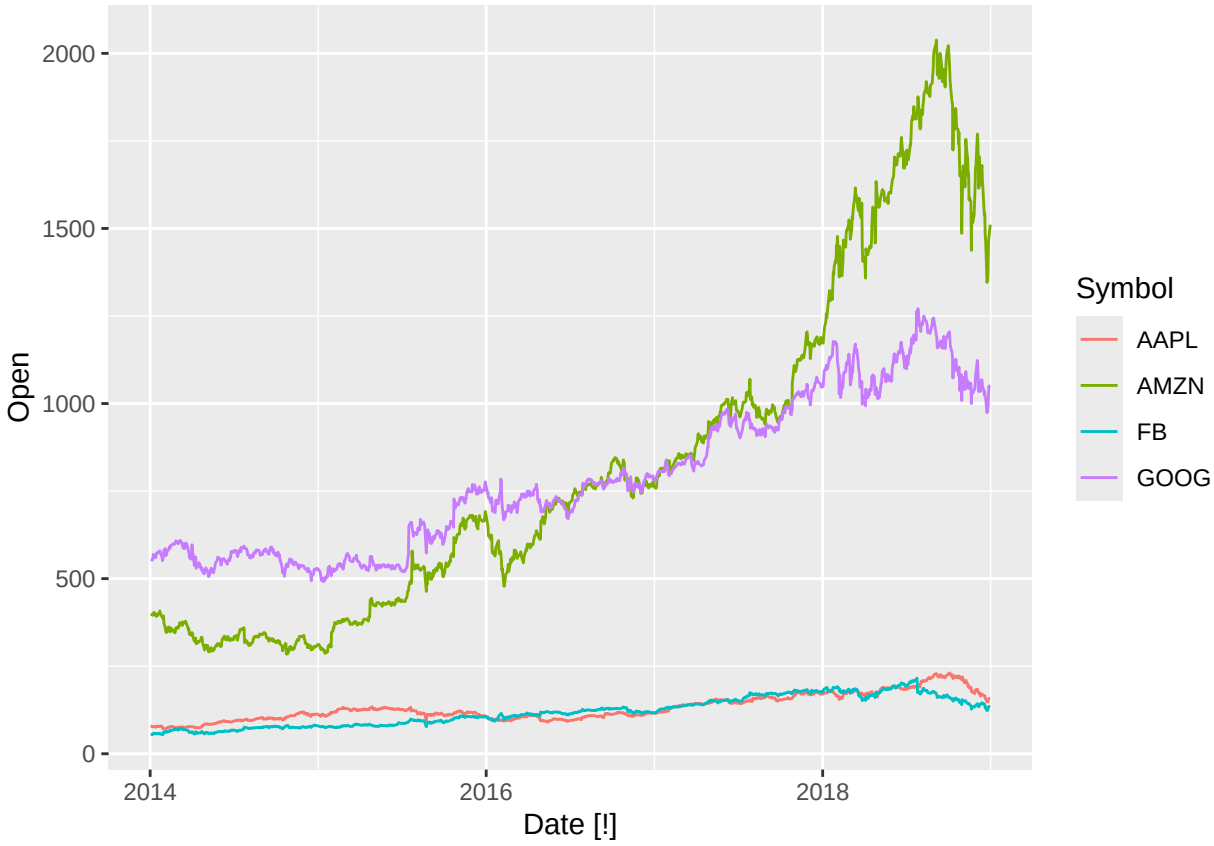
```
gafa_stock
```

```
## # A tsibble: 5,032 x 8 [!]
## # Key:       Symbol [4]
##    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
##  7 AAPL   2014-01-10  77.1  77.3  75.9  76.1      64.5  76244000
##  8 AAPL   2014-01-13  75.7  77.5  75.7  76.5      64.9  94623200
##  9 AAPL   2014-01-14  76.9  78.1  76.8  78.1      66.1  83140400
```

```
## 10 AAPL   2014-01-15  79.1  80.0  78.8  79.6      67.5  97909700
## # i 5,022 more rows
```

```
autoplot(gafa_stock)
```

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



Half-hourly electricity demand for Victoria, Australia (vic_elec): Time Interval: Half-hourly

Description 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. Format Time series of class tsibble.

Details 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).

Source Australian Energy Market Operato

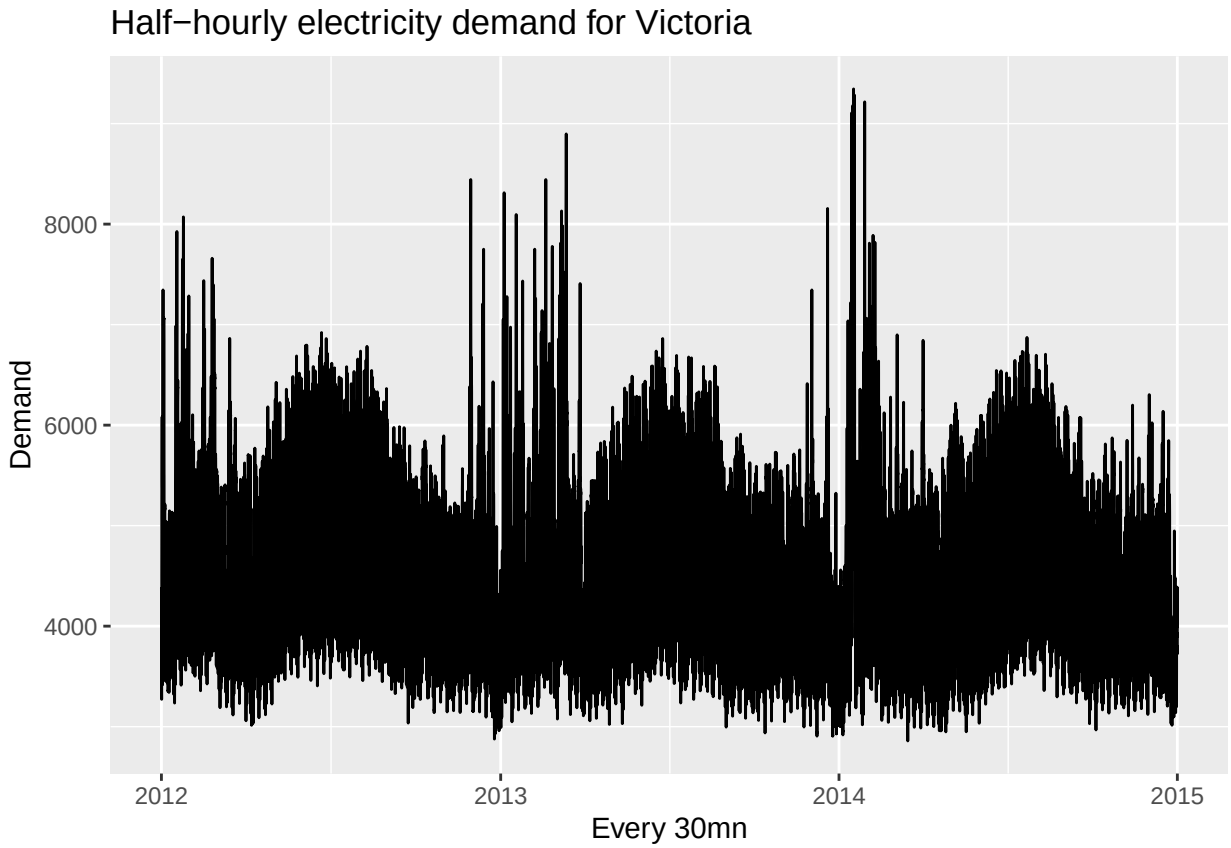
```
vic_elec
```

```
## # 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
```

```
autoplot(vic_elec) +
  labs (y= "Demand", x = "Every 30mn", title = "Half-hourly electricity demand for Victoria" )
```

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



Exercice 2: Using 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))|>
  ungroup() -> peak_closing_days
peak_closing_days
```

```
## # A tsibble: 4 x 8 [!]
## # Key:       Symbol [4]
```

```
##   Symbol Date        Open  High   Low Close Adj_Close   Volume
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl>
## 1 AAPL   2018-10-03  230.  233.  230.  232.      230. 28654800
## 2 AMZN   2018-09-04 2026. 2050. 2013  2040.     2040.  5721100
## 3 FB     2018-07-25  216.  219.  214.  218.      218. 58954200
## 4 GOOG   2018-07-26 1251  1270. 1249. 1268.     1268.  2405600
```

```
names(gafa_stock)
```

```
## [1] "Symbol"    "Date"      "Open"      "High"      "Low"       "Close"
## [7] "Adj_Close" "Volume"
```

Exercise 3

```
# reading the data into R
tute1 <- readr::read_csv("https://raw.githubusercontent.com/Heleinef/Data-Science-Master_Heleine/main/tu
```

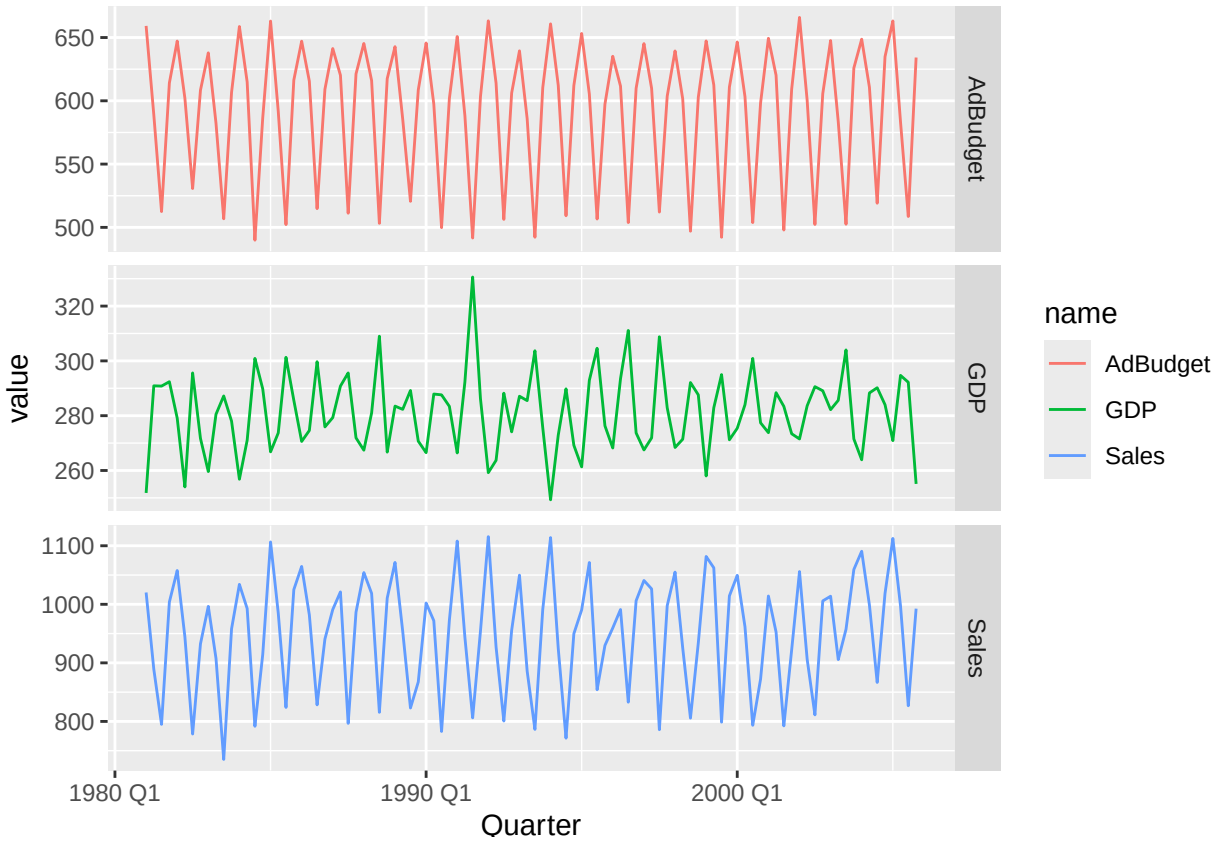
```
## Rows: 100 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (3): Sales, AdBudget, GDP
## date (1): Quarter
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(tute1)
```

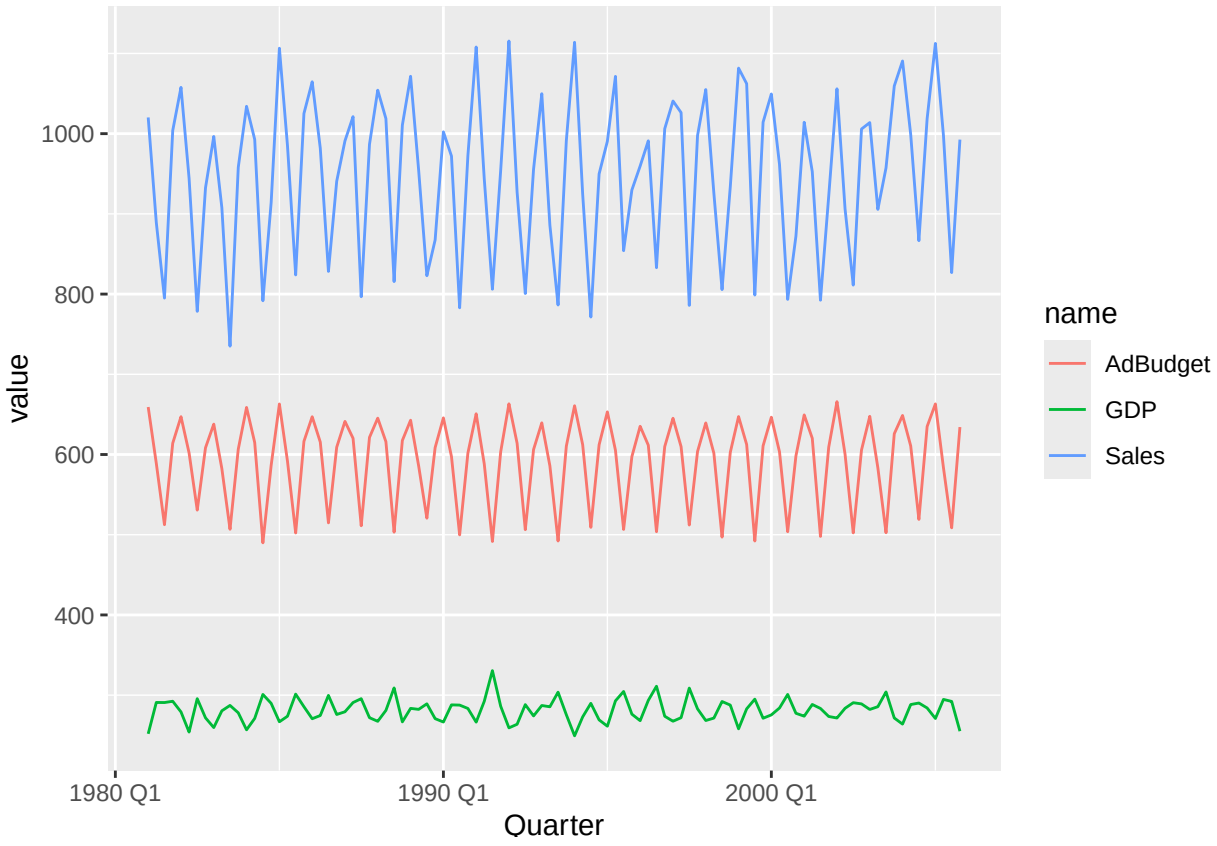
```
## # A tibble: 6 x 4
##   Quarter    Sales AdBudget   GDP
##   <date>     <dbl>    <dbl> <dbl>
## 1 1981-03-01 1020.     659.  252.
## 2 1981-06-01  889.     589   291.
## 3 1981-09-01  795      512.  291.
## 4 1981-12-01 1004.     614.  292.
## 5 1982-03-01 1058.     647.  279.
## 6 1982-06-01  944.     602   254
```

```
# Converting 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, colour = name)) +
  geom_line() +
  facet_grid(name ~ ., scales = "free_y")
```




```
# Result When facet_grid() is not included
mytimeseries |>
  pivot_longer(-Quarter) |>
  ggplot(aes(x = Quarter, y = value, colour = name)) +
  geom_line()
```



Exercise 4

```
# Install the USgas package.
data("us_total", package = "USgas")  # Load the data from USgas
glimpse(us_total)  # Take a look at the structure
```

```
## Rows: 1,266
## Columns: 3
## $ year  <int> 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007~
## $ state <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"~
## $ y     <int> 324158, 329134, 337270, 353614, 332693, 379343, 350345, 382367, ~
```

```
# Create a tsibble from us_total with year as the index and state as the key
us_total_tsibble <- us_total |>
  as_tsibble(index = year, key = state)
us_total_tsibble
```

```
## # A tsibble: 1,266 x 3 [1Y]
## # Key:       state [53]
##     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
```

```
##  7  2003 Alabama 350345
##  8  2004 Alabama 382367
##  9  2005 Alabama 353156
## 10  2006 Alabama 391093
## # i 1,256 more rows
```

```
# Plot the annual natural gas consumption by state for the New England area (comprising the states of M
new_england_states <- c("Maine", "Vermont", "New Hampshire", "Massachusetts", "Connecticut", "Rhode Isla
us_total_tsibble |>
  filter(state %in% new_england_states)->new_england_data
new_england_data
```

```
## # A tsibble: 138 x 3 [1Y]
## # Key:       state [6]
##     year state            y
##    <int> <chr>        <int>
##  1  1997 Connecticut 144708
##  2  1998 Connecticut 131497
##  3  1999 Connecticut 152237
##  4  2000 Connecticut 159712
##  5  2001 Connecticut 146278
##  6  2002 Connecticut 177587
##  7  2003 Connecticut 154075
##  8  2004 Connecticut 162642
##  9  2005 Connecticut 168067
## 10  2006 Connecticut 172682
## # i 128 more rows
```

Exercise 5

```
# Reading the tourism file into R
tourism_new <- readr::read_csv("https://raw.githubusercontent.com/Heleinef/Data-Science-Master_Heleine/m
```

```
## Rows: 24320 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Region, State, Purpose
## dbl  (1): Trips
## 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.
```

```
tourism_new
```

```
## # A tibble: 24,320 x 5
##    Quarter    Region   State           Purpose  Trips
##    <date>     <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
```

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

tourism_tsibble <- tourism_new |>
  as_tsibble(index = Quarter, key = c(Region, State, Purpose))
tourism_tsibble
```

```
## # A tsibble: 24,320 x 5 [1D]
## # Key:       Region, State, Purpose [304]
##    Quarter    Region   State           Purpose  Trips
##    <date>     <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
```

```
# First, calculate avaerage trips by region and purpose
tourism_avg <- tourism_tsibble |>
  group_by(Region, Purpose) |>
  summarise(avg_trips = mean(Trips, na.rm = TRUE)) |>
  ungroup()

tourism_avg
```

```
## # A tsibble: 24,320 x 4 [1D]
## # Key:       Region, Purpose [304]
##    Region   Purpose  Quarter    avg_trips
##    <chr>    <chr>    <date>         <dbl>
##  1 Adelaide Business 1998-01-01      135.
##  2 Adelaide Business 1998-04-01      110.
##  3 Adelaide Business 1998-07-01      166.
##  4 Adelaide Business 1998-10-01      127.
##  5 Adelaide Business 1999-01-01      137.
##  6 Adelaide Business 1999-04-01      200.
##  7 Adelaide Business 1999-07-01      169.
##  8 Adelaide Business 1999-10-01      134.
##  9 Adelaide Business 2000-01-01      154.
## 10 Adelaide Business 2000-04-01      169.
## # i 24,310 more rows
```

```
# Then, find what combination of Region and Purpose had the maximum number of overnight trips on averag
max_avg_trips <- tourism_avg |>
  filter(avg_trips == max(avg_trips))

max_avg_trips
```

```
## # A tsibble: 1 x 4 [1D]
## # Key:       Region, Purpose [1]
##   Region    Purpose  Quarter    avg_trips
##   <chr>     <chr>    <date>         <dbl>
## 1 Melbourne Visiting 2017-10-01      985.
```

```
# Create a new tsibble which combines the Purposes and Regions, and just has total trips by State.
new_tsibble <- tourism_tsibble |>
  group_by(State) |>
  summarise(Total_Trips = sum(Trips, na.rm = TRUE)) %>%  # Summarize total trips by state
  as_tsibble(key = State)     # Convert the result to a tsibble
new_tsibble
```

```
## # A tsibble: 640 x 3 [1D]
## # Key:       State [8]
##    State Quarter    Total_Trips
##    <chr> <date>           <dbl>
##  1 ACT   1998-01-01        551.
##  2 ACT   1998-04-01        416.
##  3 ACT   1998-07-01        436.
##  4 ACT   1998-10-01        450.
##  5 ACT   1999-01-01        379.
##  6 ACT   1999-04-01        558.
##  7 ACT   1999-07-01        449.
##  8 ACT   1999-10-01        595.
##  9 ACT   2000-01-01        600.
## 10 ACT   2000-04-01        557.
## # i 630 more rows
```

Exercise 6: On US Arrivals

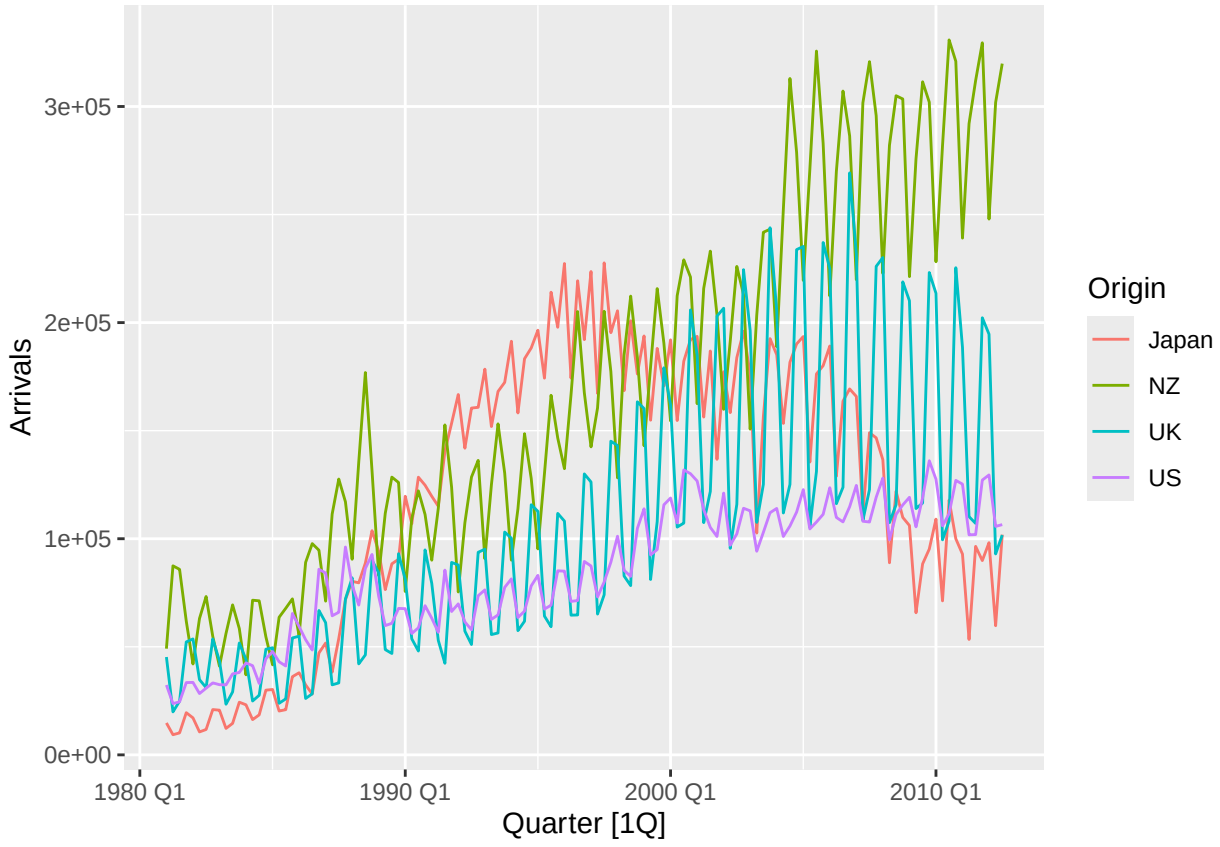
```
# The dataset
 aus_arrivals
```

```
## # A tsibble: 508 x 3 [1Q]
## # Key:       Origin [4]
##    Quarter Origin Arrivals
##      <qtr> <chr>     <int>
##  1 1981 Q1 Japan     14763
##  2 1981 Q2 Japan      9321
##  3 1981 Q3 Japan     10166
##  4 1981 Q4 Japan     19509
##  5 1982 Q1 Japan     17117
##  6 1982 Q2 Japan     10617
##  7 1982 Q3 Japan     11737
##  8 1982 Q4 Japan     20961
##  9 1983 Q1 Japan     20671
## 10 1983 Q2 Japan     12235
## # i 498 more rows
```

Use autoplot(), gg_season() and gg_subseries() to compare the differences between the arrivals from these four countries.

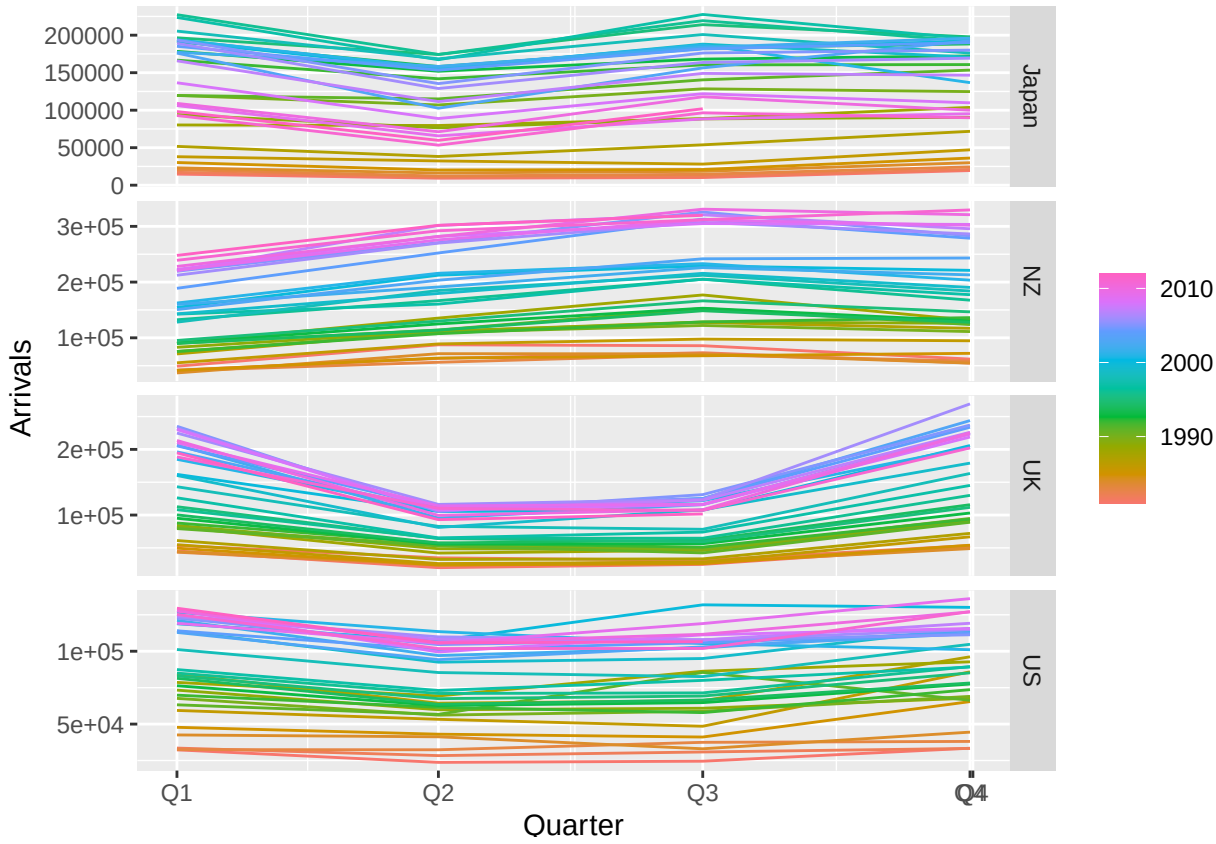
```
# Using the autoplot function
autoplot(aus_arrivals)
```

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



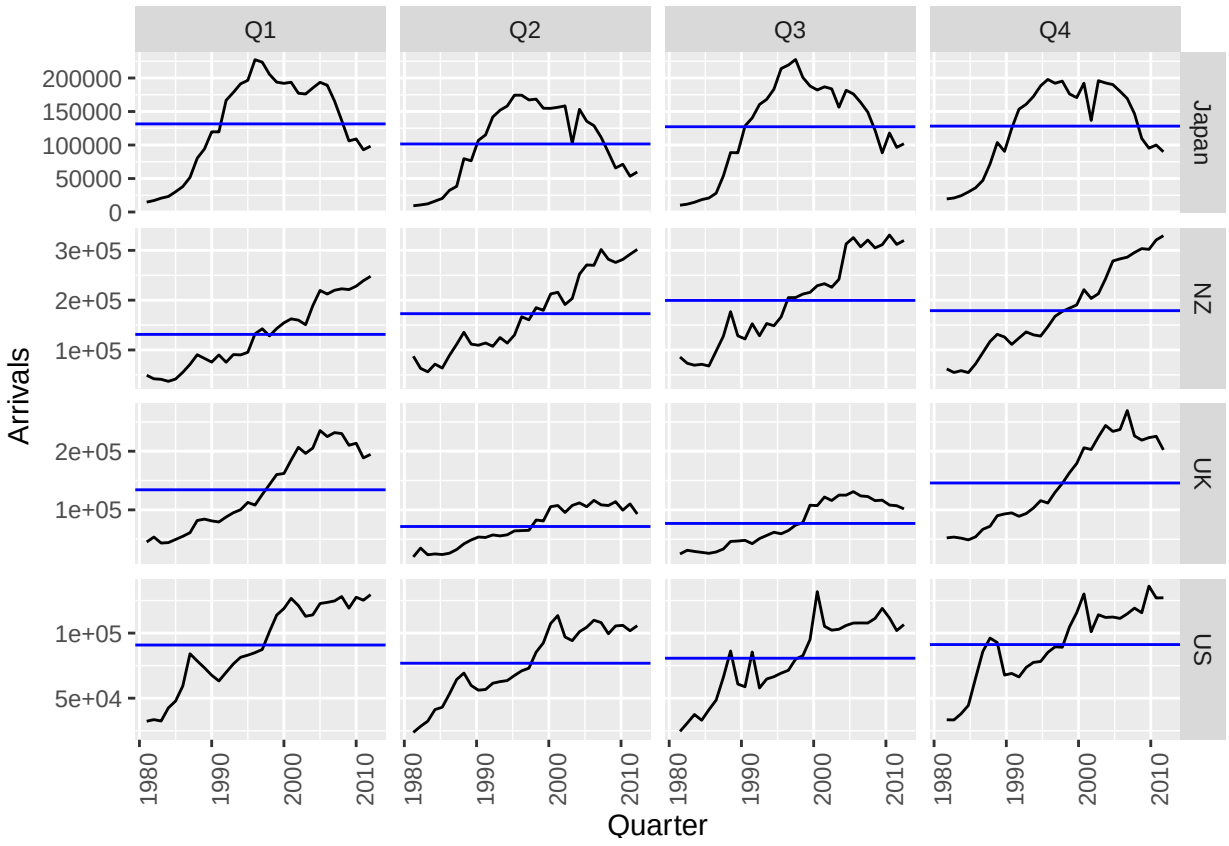

```
# Using the gg_season()
gg_season(aus_arrivals)
```

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




```
# Using the gg_subseries()
gg_subseries(aus_arrivals)
```

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



One notes that, compared to the other 3 countries, Japan's arrivals have been following a downward trend from the 1980s to 2010.

Exercise 7: Monthly Australian retail data

```
# The dataset

 aus_retail
```

```
## # A tsibble: 64,532 x 5 [1M]
## # Key:       State, Industry [152]
##    State                          Industry            `Series ID`    Month Turnover
##    <chr>                          <chr>               <chr>          <mth>    <dbl>
##  1 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Apr      4.4
##  2 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 May      3.4
##  3 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Jun      3.6
##  4 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Jul      4
##  5 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Aug      3.6
##  6 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Sep      4.2
##  7 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Oct      4.8
##  8 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Nov      5.4
##  9 Australian Capital Territory Cafes, restaurant~ A3349849A   1982 Dec      6.9
## 10 Australian Capital Territory Cafes, restaurant~ A3349849A   1983 Jan      3.8
## # i 64,522 more rows
```

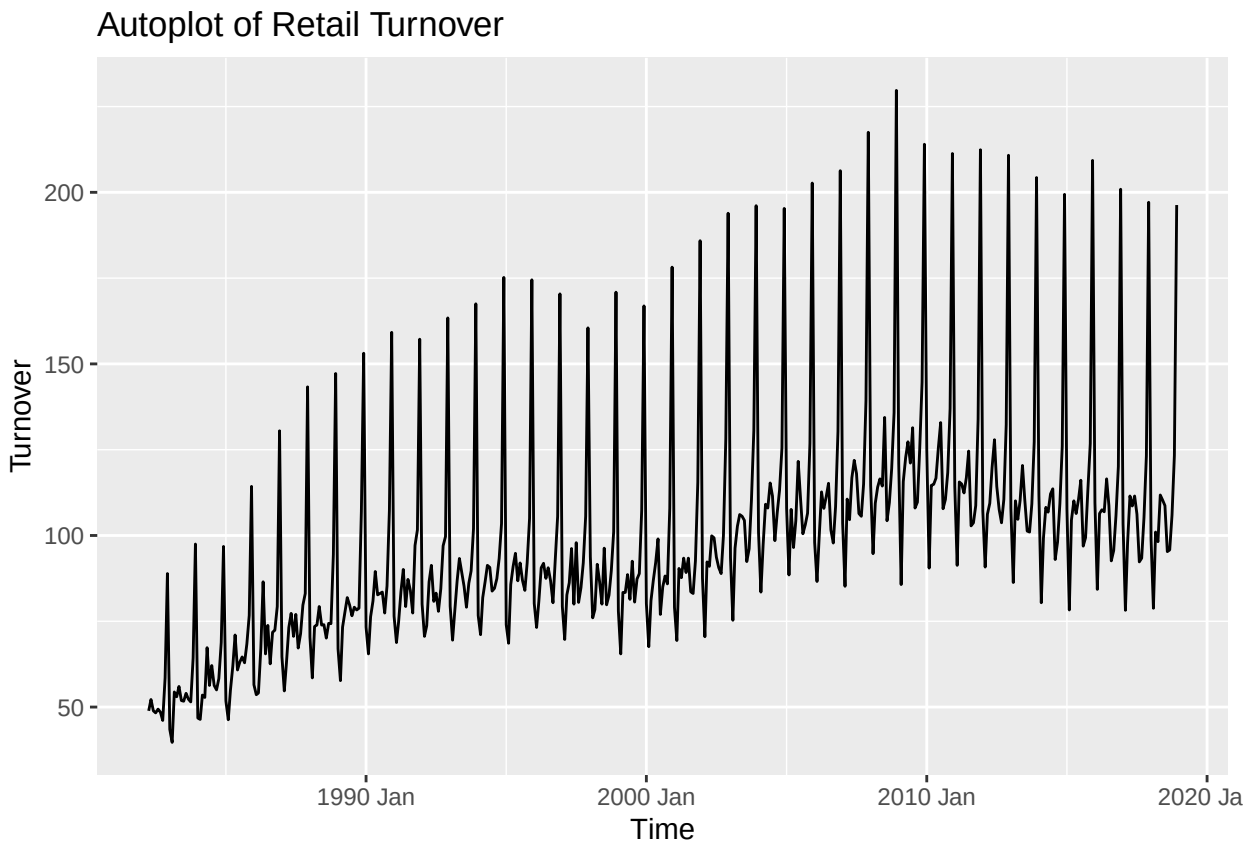
```
set.seed(13101917)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

```
myseries
```

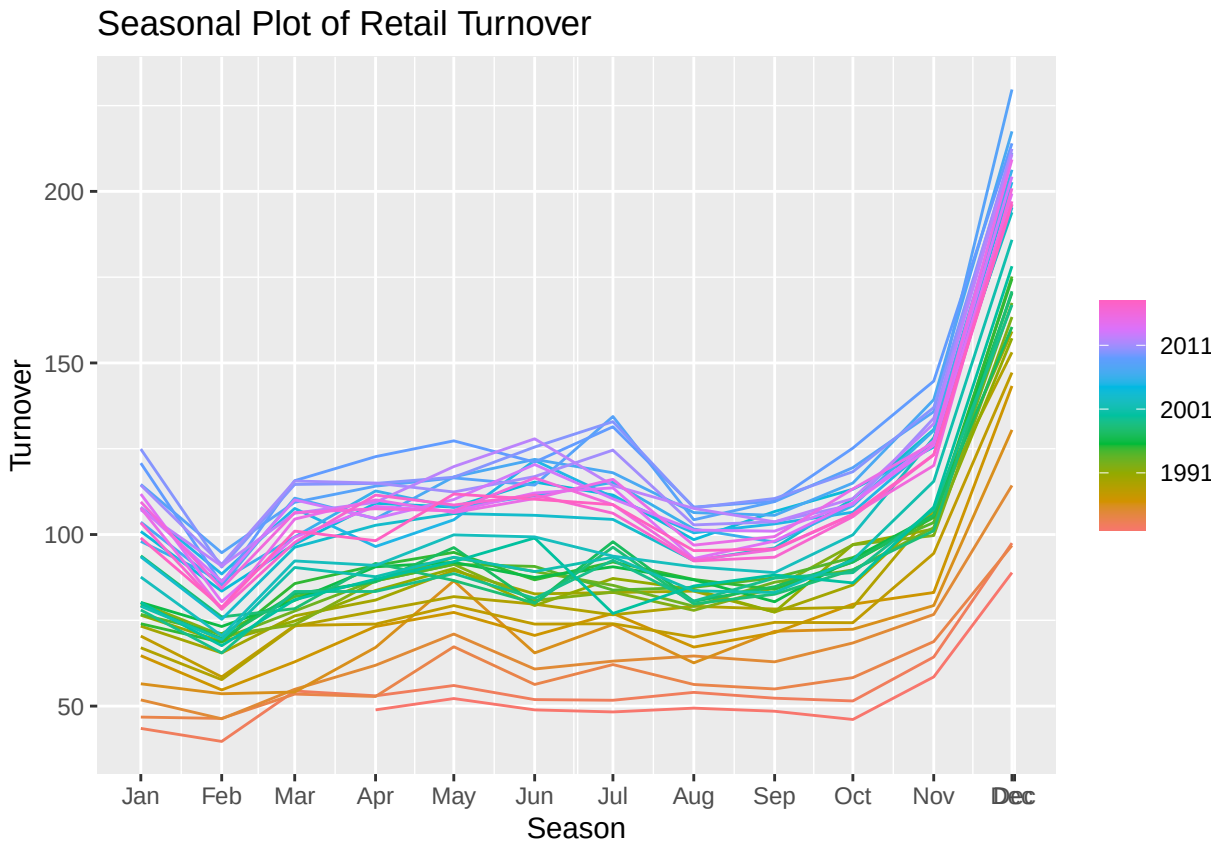
```
## # A tsibble: 441 x 5 [1M]
## # Key:       State, Industry [1]
##    State           Industry          `Series ID`    Month Turnover
##    <chr>           <chr>             <chr>          <mth>    <dbl>
##  1 South Australia Department stores A3349658K   1982 Apr     48.9
##  2 South Australia Department stores A3349658K   1982 May     52.2
##  3 South Australia Department stores A3349658K   1982 Jun     48.9
##  4 South Australia Department stores A3349658K   1982 Jul     48.3
##  5 South Australia Department stores A3349658K   1982 Aug     49.4
##  6 South Australia Department stores A3349658K   1982 Sep     48.5
##  7 South Australia Department stores A3349658K   1982 Oct     46.1
##  8 South Australia Department stores A3349658K   1982 Nov     58.5
##  9 South Australia Department stores A3349658K   1982 Dec     88.9
## 10 South Australia Department stores A3349658K   1983 Jan     43.5
## # i 431 more rows
```

Explore your chosen retail time series using the following functions:autoplot(), gg_season(), gg_subseries(), gg_lag(),ACF() |> autoplot() with visualizations

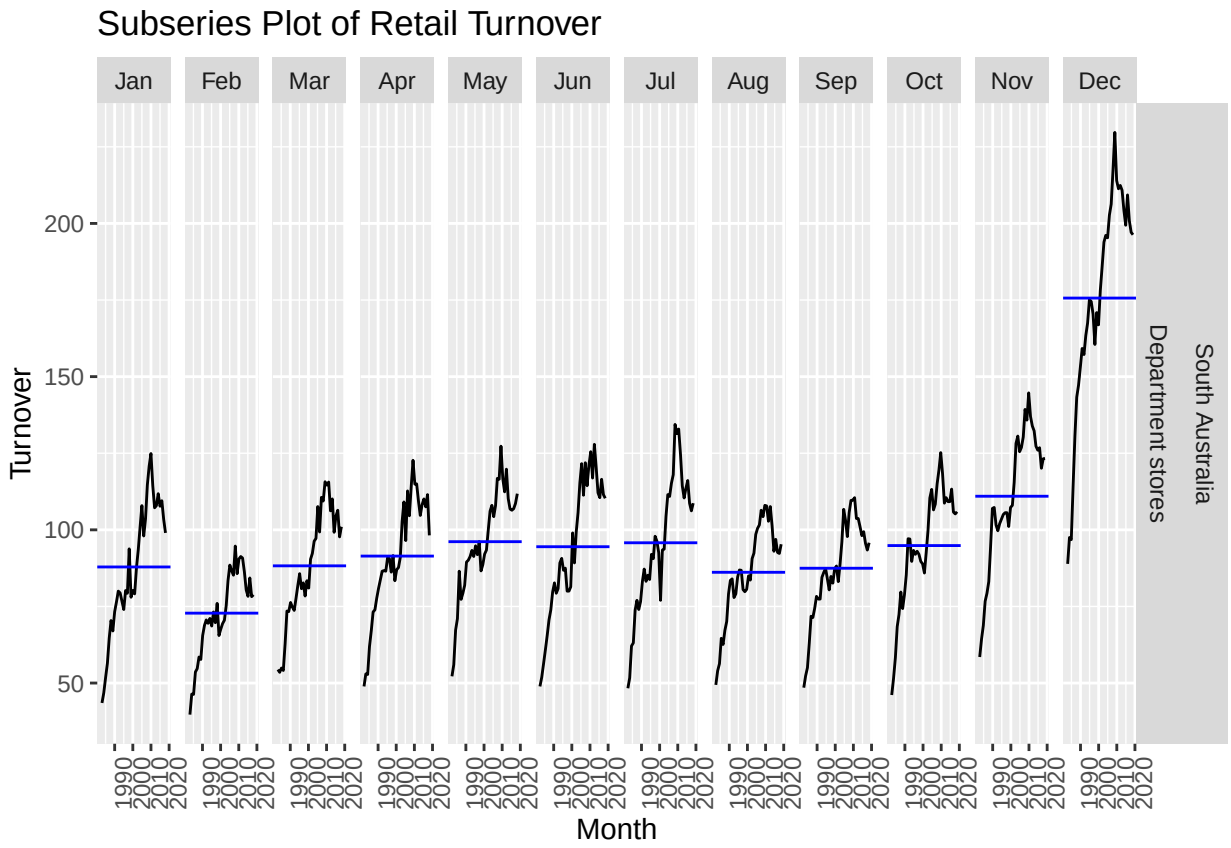
```
#  1. Autoplot of the selected series
autoplot(myseries, Turnover) +
  ggtitle("Autoplot of Retail Turnover") +
  ylab("Turnover") + xlab("Time")
```



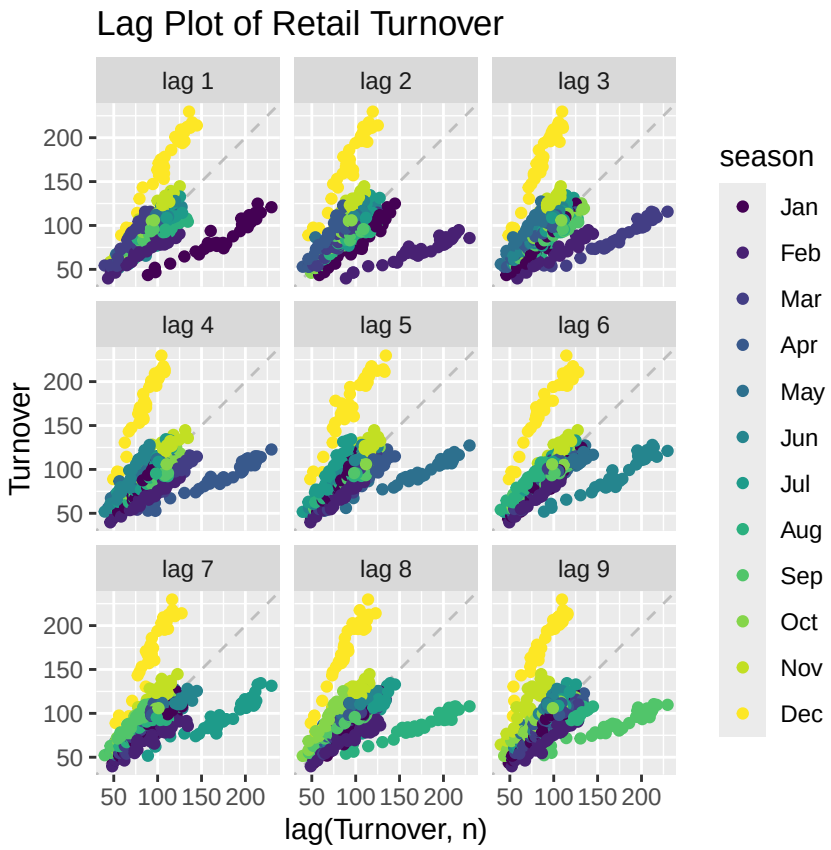
```r
# 2. Seasonal plot to visualize seasonality
gg_season(myseries, Turnover) +
  ggtitle("Seasonal Plot of Retail Turnover") +
  ylab("Turnover") + xlab("Season")
```



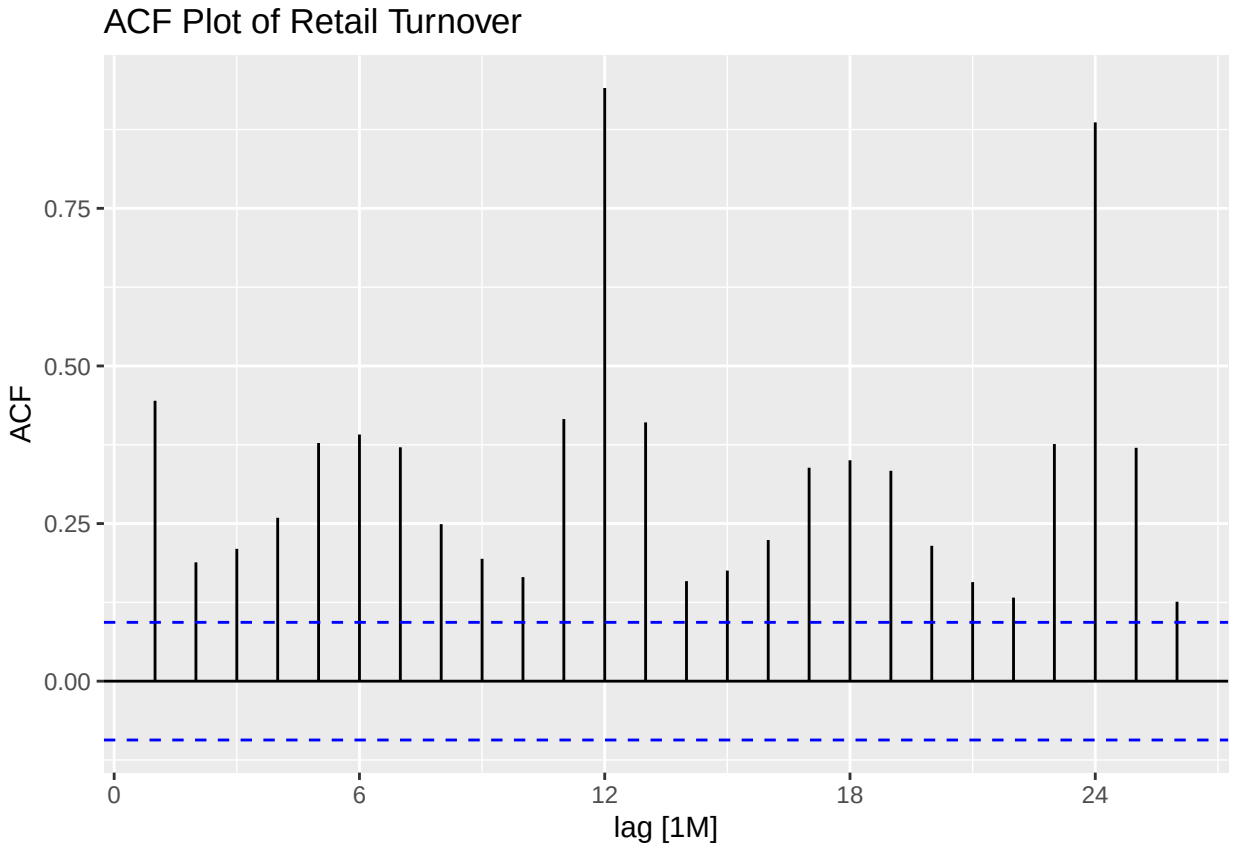

```r
# 3. Subseries plot for further seasonality exploration
gg_subseries(myseries, Turnover) +
  ggtitle("Subseries Plot of Retail Turnover") +
  ylab("Turnover") + xlab("Month")
```




```
# 4. Lag plot to examine correlation between lags
gg_lag(myseries, Turnover, geom = "point") +
  ggtitle("Lag Plot of Retail Turnover")
```




```
# 5. Autocorrelation plot to visualize cyclicity and correlation in the data
ACF(myseries, Turnover) %>%
  autoplot() +
  ggtitle("ACF Plot of Retail Turnover") +
  ylab("ACF")
```



Can you spot any seasonality, cyclicity and trend? What do you learn about the series? YES, I spot both trend, seaonality and cyclicity in plots.

Indeed, the various plots helped me visualize the behavior of the selected time series,and allowed me to detect seasonality, trends, or cyclic patterns in the retail turnover data, depending of the selected function.

autoplot(myseries, Turnover): provided an overview of the retail sales over time, allowing me to observe trends and potential patterns in the data.

gg_season(myseries, Turnover): Revealed the seasonal variation by decomposing the data into seasons (e.g., months). It helps identify repeating patterns.

gg_subseries(myseries, Turnover): This plot also helps in observing seasonal behavior by breaking down the data by individual months or seasons.It can highlight if a particular month/season has consistently high or low sales.

gg_lag(myseries, Turnover, geom = "point"): A lag plot helps visualize the correlation between observations at different time lags.Cyclic behavior: If there's a repeating cycle or seasonality, the lag plot might show a certain correlation structure.

ACF() |> autoplot(): The autocorrelation function (ACF) plot shows the correlation of the time series with its own lags. Peaks in the ACF at regular intervals can indicate seasonality, while slow decay might suggest a trend or cyclicity.

Exercise 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.

```
us_employment
```

```
## # A tsibble: 143,412 x 4 [1M]
## # Key:       Series_ID [148]
##       Month Series_ID     Title          Employed
##       <mth> <chr>         <chr>             <dbl>
##  1 1939 Jan CEU0500000001 Total Private    25338
##  2 1939 Feb CEU0500000001 Total Private    25447
##  3 1939 Mar CEU0500000001 Total Private    25833
##  4 1939 Apr CEU0500000001 Total Private    25801
##  5 1939 May CEU0500000001 Total Private    26113
##  6 1939 Jun CEU0500000001 Total Private    26485
##  7 1939 Jul CEU0500000001 Total Private    26481
##  8 1939 Aug CEU0500000001 Total Private    26848
##  9 1939 Sep CEU0500000001 Total Private    27468
## 10 1939 Oct CEU0500000001 Total Private    27830
## # i 143,402 more rows
```

```
colnames(us_employment)
```

```
## [1] "Month"     "Series_ID" "Title"     "Employed"
```

```
aus_production
```

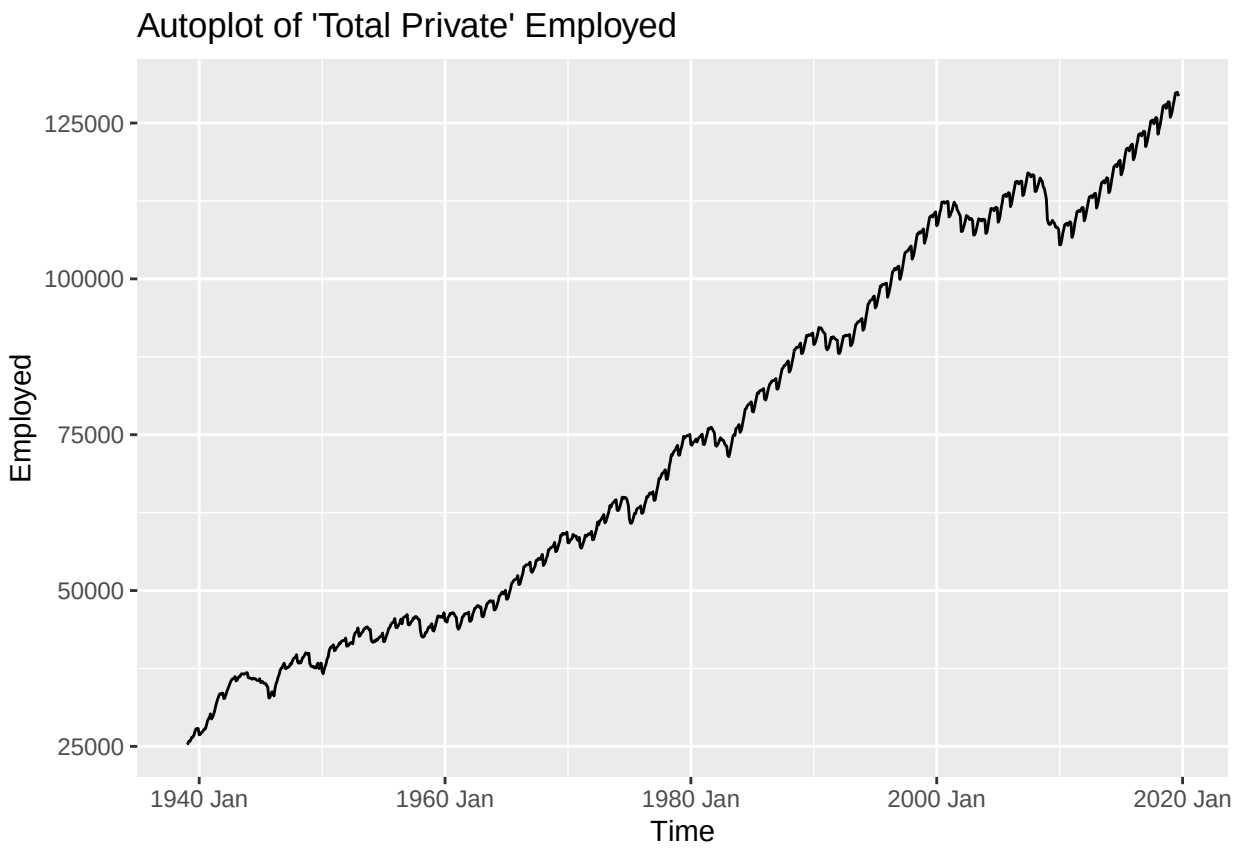
```
## # A tsibble: 218 x 7 [1Q]
##    Quarter  Beer Tobacco Bricks Cement Electricity   Gas
##      <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
##  1 1956 Q1   284    5225    189    465        3923     5
##  2 1956 Q2   213    5178    204    532        4436     6
##  3 1956 Q3   227    5297    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
##  7 1957 Q3   236    5317    227    603        5259     7
##  8 1957 Q4   320    6152    222    582        4735     6
##  9 1958 Q1   272    5758    199    554        4608     5
## 10 1958 Q2   233    5641    229    620        5196     7
## # i 208 more rows
```

```
PBS
```

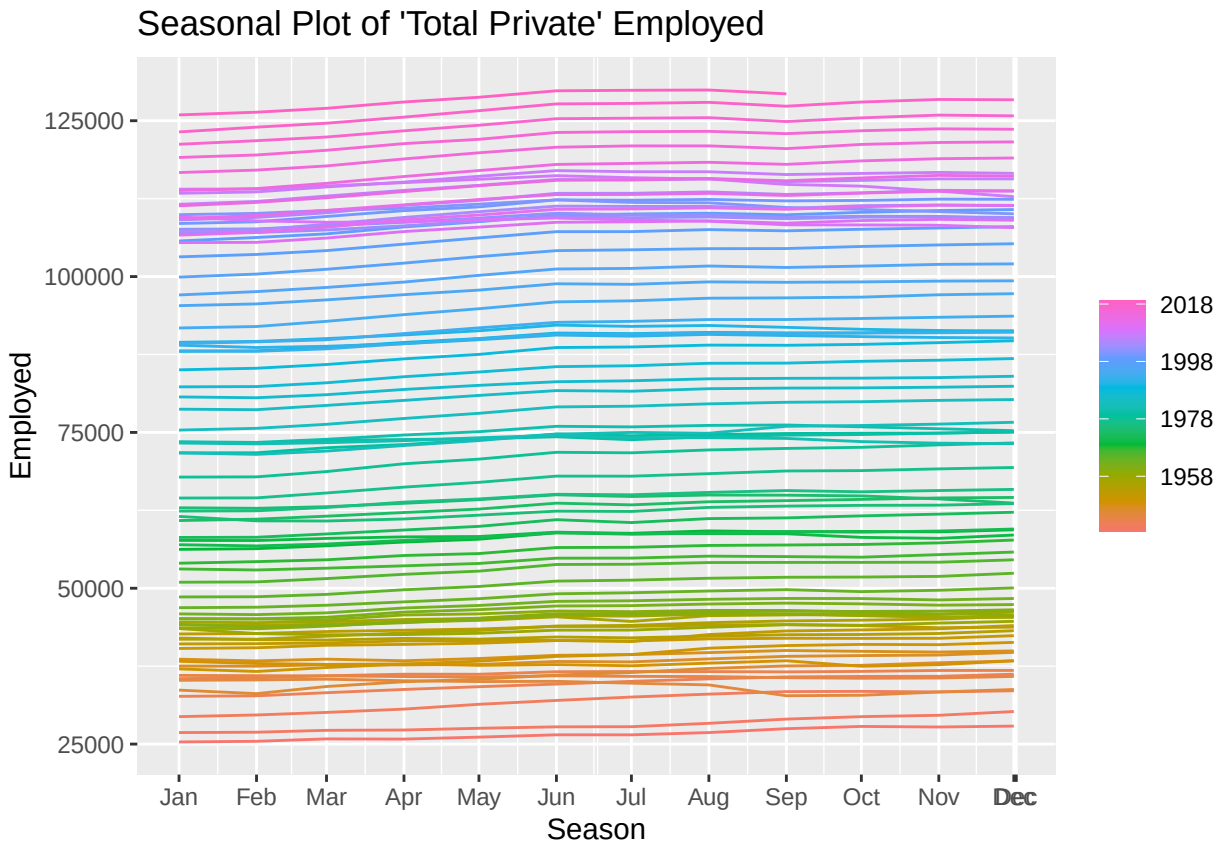
```
## # A tsibble: 67,596 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [336]
##       Month Concession   Type      ATC1  ATC1_desc ATC2  ATC2_desc Scripts  Cost
##       <mth> <chr>        <chr>     <chr> <chr>     <chr> <chr>       <dbl> <dbl>
##  1 1991 Jul Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   18228 67877
##  2 1991 Aug Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   15327 57011
##  3 1991 Sep Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   14775 55020
##  4 1991 Oct Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   15380 57222
##  5 1991 Nov Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   14371 52120
```

```
##  6 1991 Dec Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   15028 54299
##  7 1992 Jan Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   11040 39753
##  8 1992 Feb Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   15165 54405
##  9 1992 Mar Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   16898 61108
## 10 1992 Apr Concessional Co-payme~ A     Alimenta~ A01   STOMATOL~   18141 65356
## # i 67,586 more rows
```

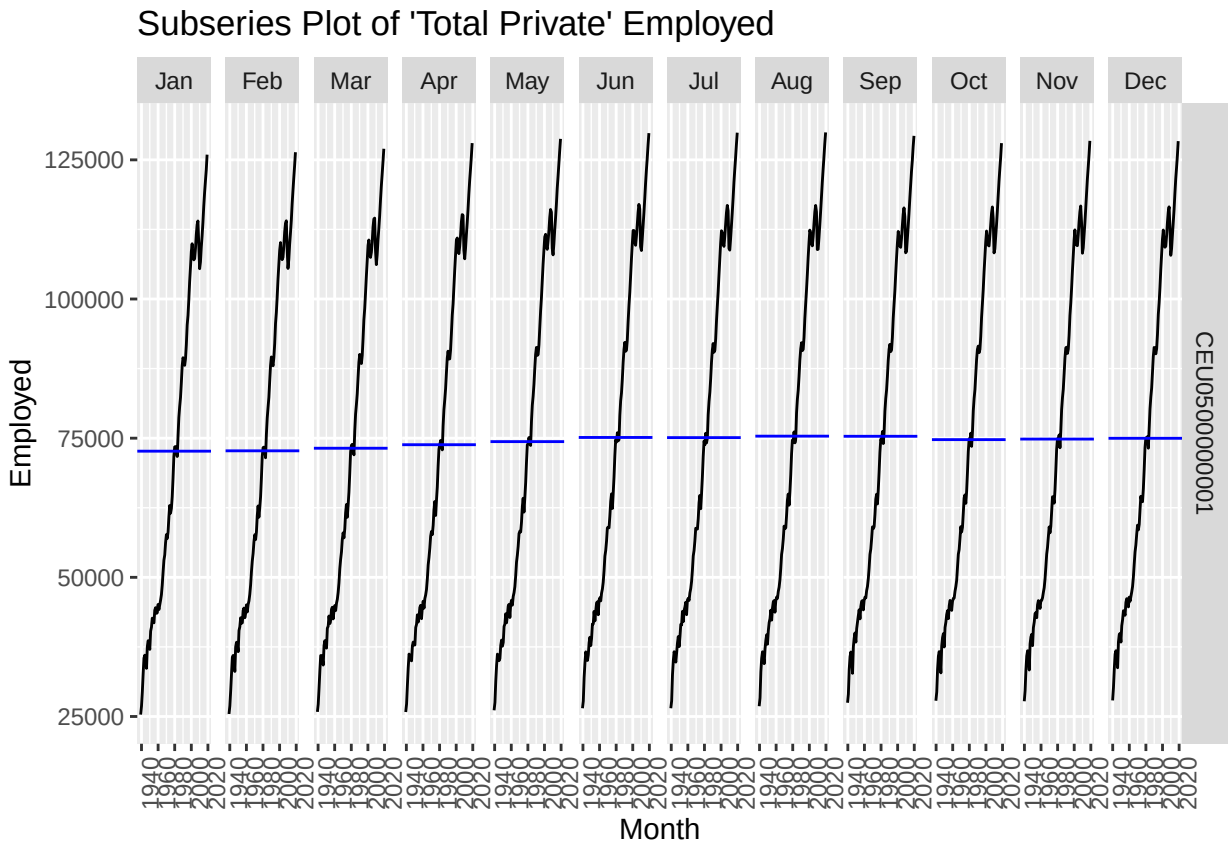
```r
# 1. Explore "Total Private" Employed from us_employment

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

# 2. Autoplot of the series
autoplot(total_private_employment, Employed) +
  ggtitle("Autoplot of 'Total Private' Employed") +
  ylab("Employed") + xlab("Time")
```



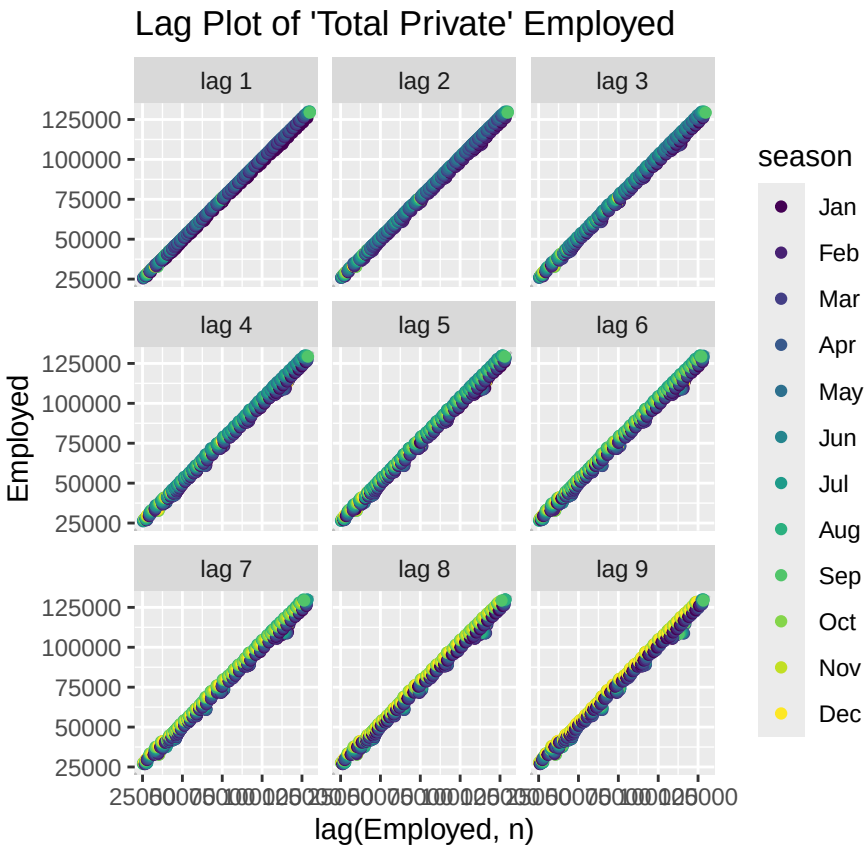

```r
# 3. Seasonal plot to visualize seasonality
gg_season(total_private_employment, Employed) +
  ggtitle("Seasonal Plot of 'Total Private' Employed") +
  ylab("Employed") + xlab("Season")
```



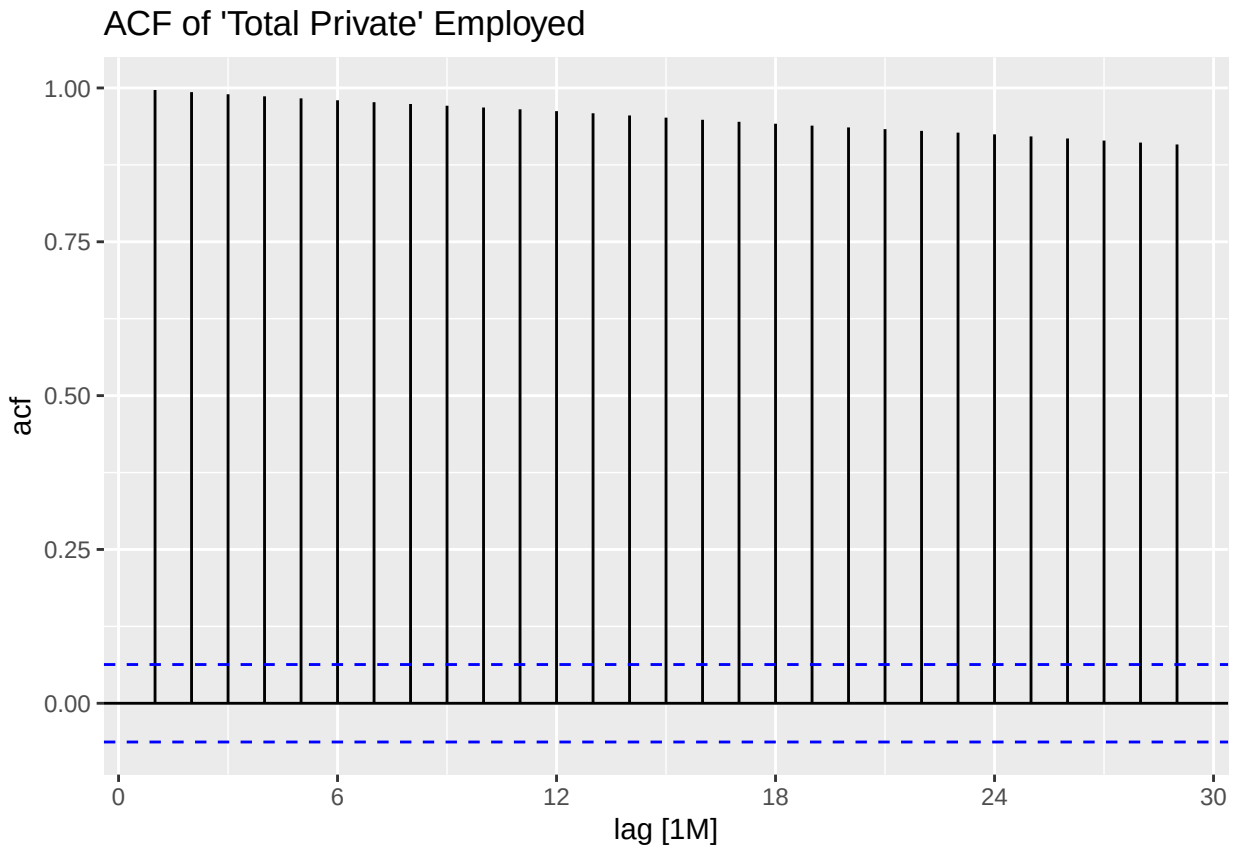

```
# 4. Subseries plot for further seasonality exploration
gg_subseries(total_private_employment, Employed) +
  ggtitle("Subseries Plot of 'Total Private' Employed") +
  ylab("Employed") + xlab("Month")
```




```
# 5. Lag plot to examine correlation between lags
gg_lag(total_private_employment, Employed, geom = "point") +
  ggtitle("Lag Plot of 'Total Private' Employed")
```

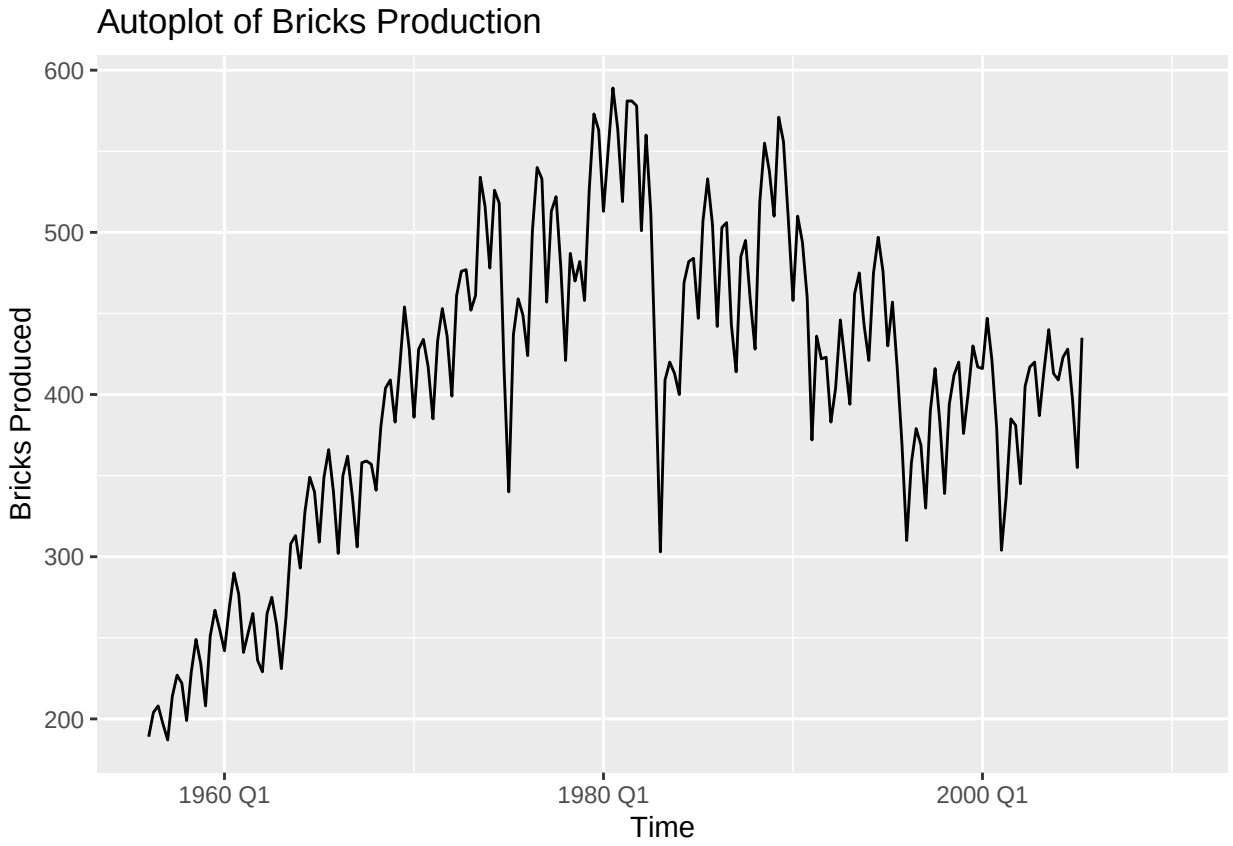



```
# 6. Autocorrelation plot to visualize cyclicity and correlation in the data
ACF(total_private_employment, Employed) %>%
  autoplot() +
  ggtitle("ACF of 'Total Private' Employed")
```



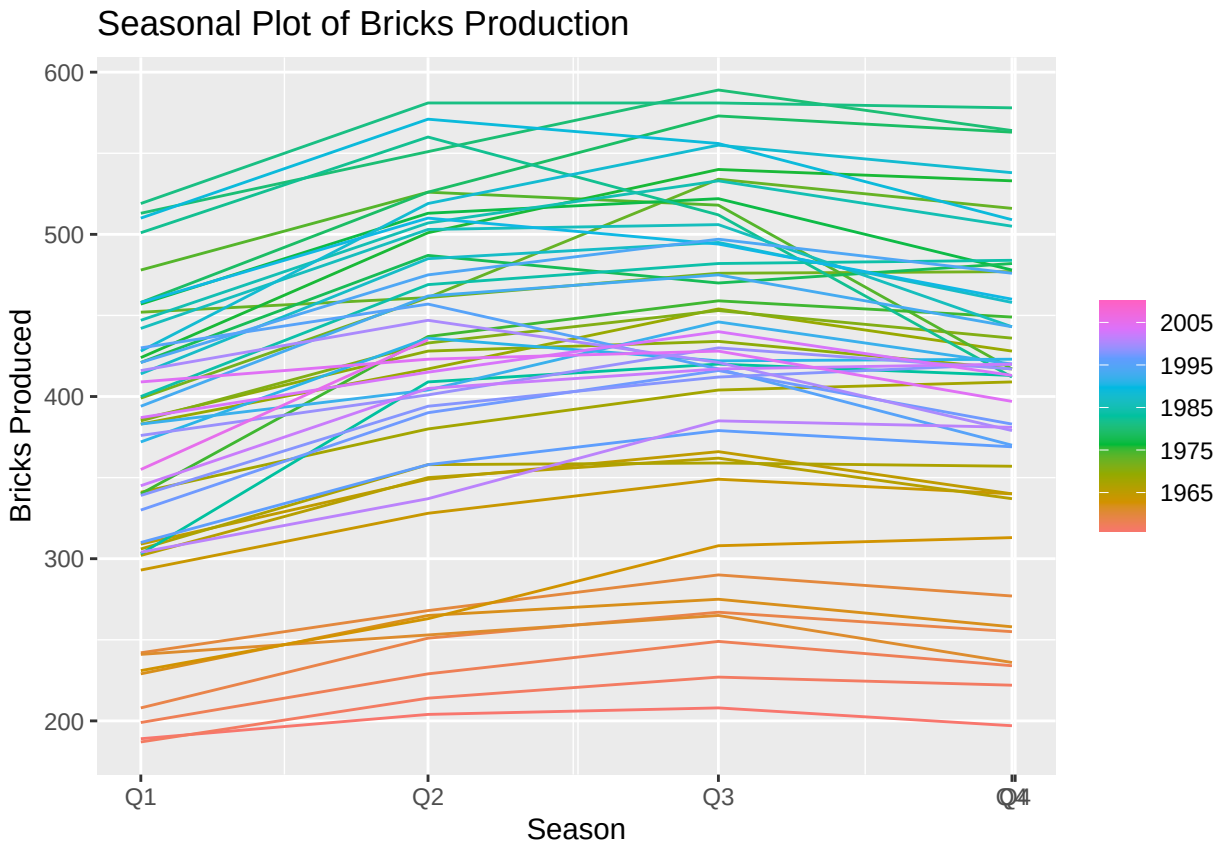

```
# 2. Explore Bricks from aus_production
autoplot(aus_production, Bricks) +
  ggtitle("Autoplot of Bricks Production") +
  ylab("Bricks Produced") + xlab("Time")
```

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



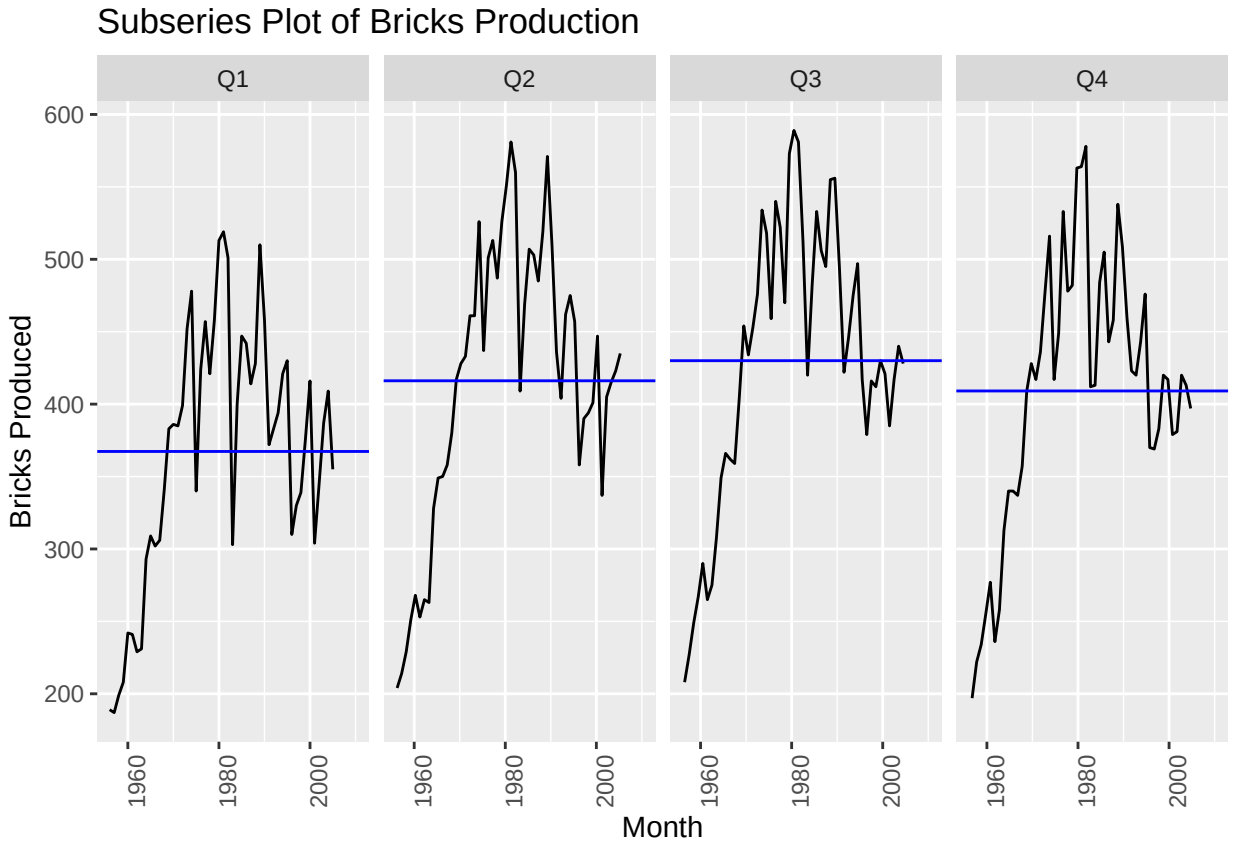

```
gg_season(aus_production, Bricks) +
  ggtitle("Seasonal Plot of Bricks Production") +
  ylab("Bricks Produced") + xlab("Season")
```

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



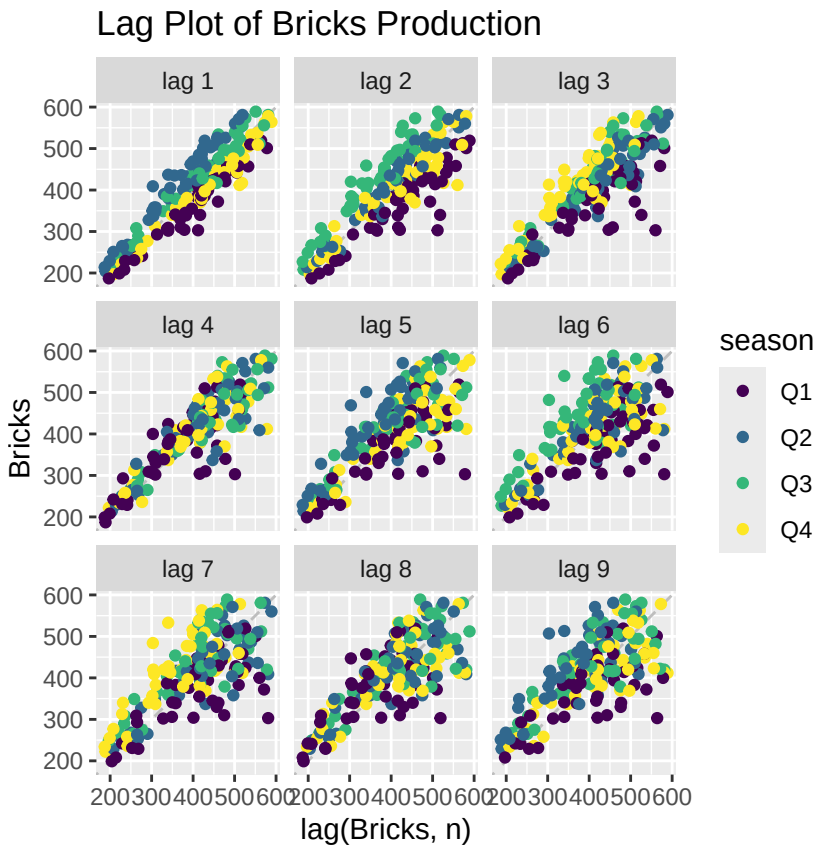

```
gg_subseries(aus_production, Bricks) +
  ggtitle("Subseries Plot of Bricks Production") +
  ylab("Bricks Produced") + xlab("Month")
```

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

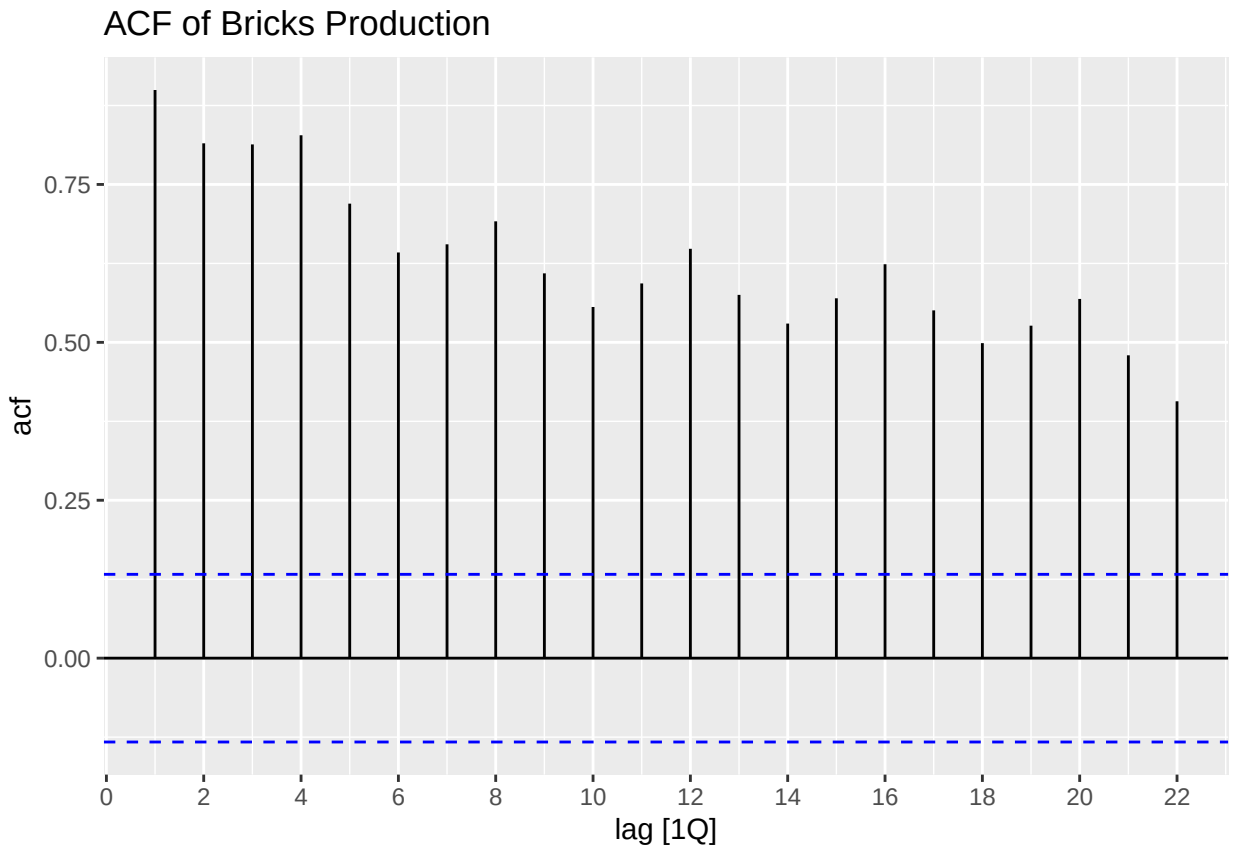



```
gg_lag(aus_production, Bricks, geom = "point") +
  ggtitle("Lag Plot of Bricks Production")
```

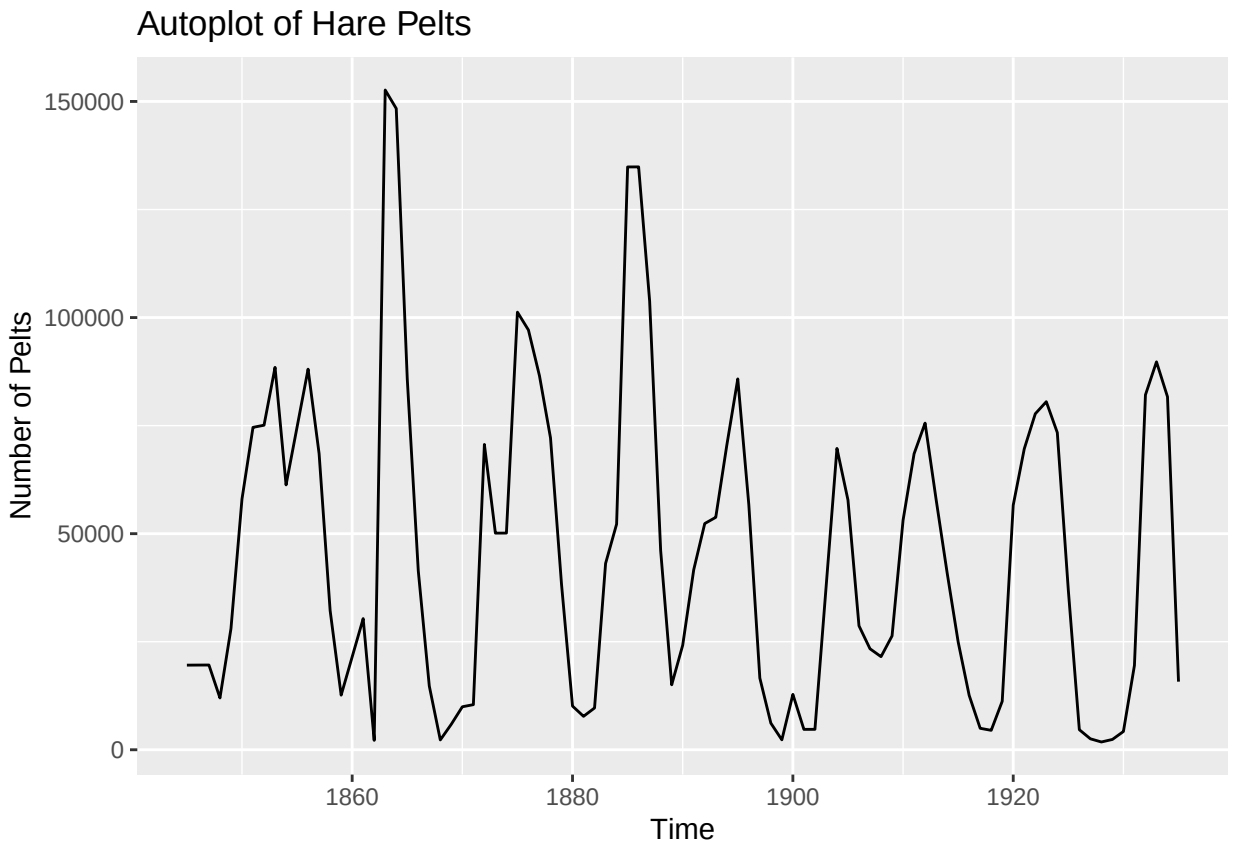
```
## Warning: Removed 20 rows containing missing values (gg_lag).
```



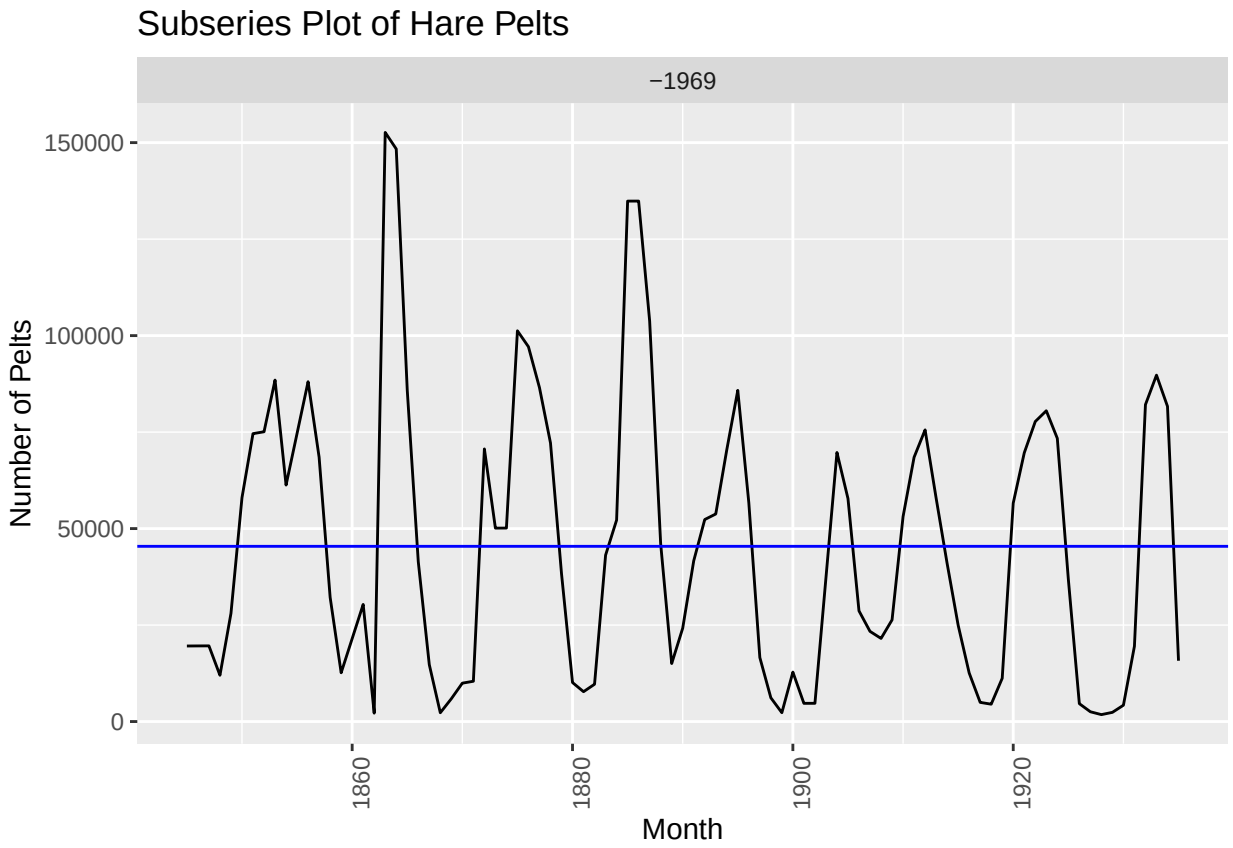

```
ACF(aus_production, Bricks) %>%
  autoplot() +
  ggtitle("ACF of Bricks Production")
```



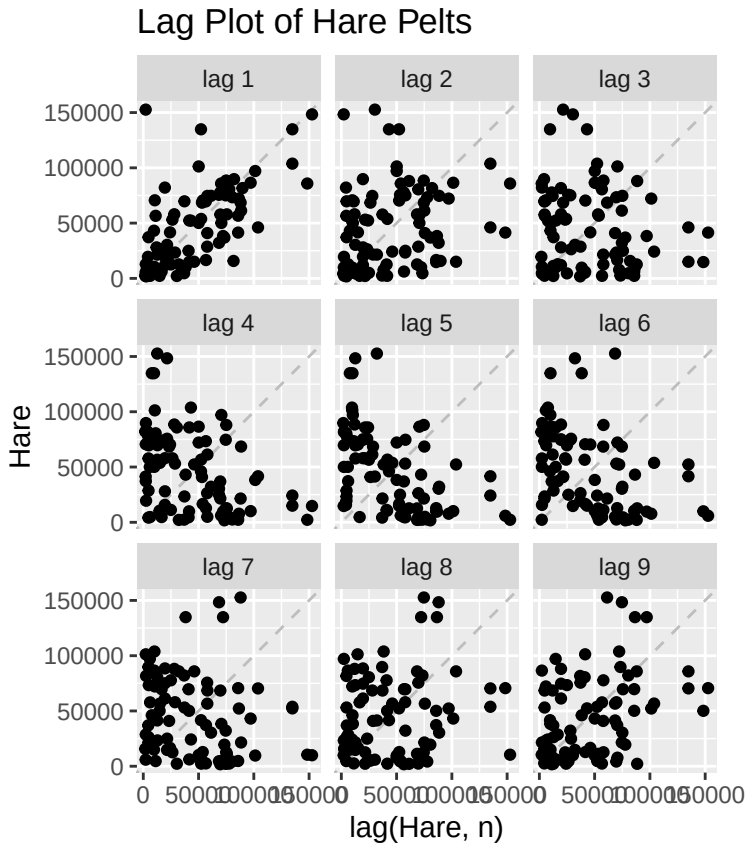

```
# 3. Explore Hare from pelt

# Convert to tsibble if necessary
pelt_tsibble <- pelt |>
  as_tsibble(index = Year)  # Assuming 'Year' is the index
#  Autoplot of Hare Pelts
autoplot(pelt_tsibble, Hare) +
  ggtitle("Autoplot of Hare Pelts") +
  ylab("Number of Pelts") + xlab("Time")
```



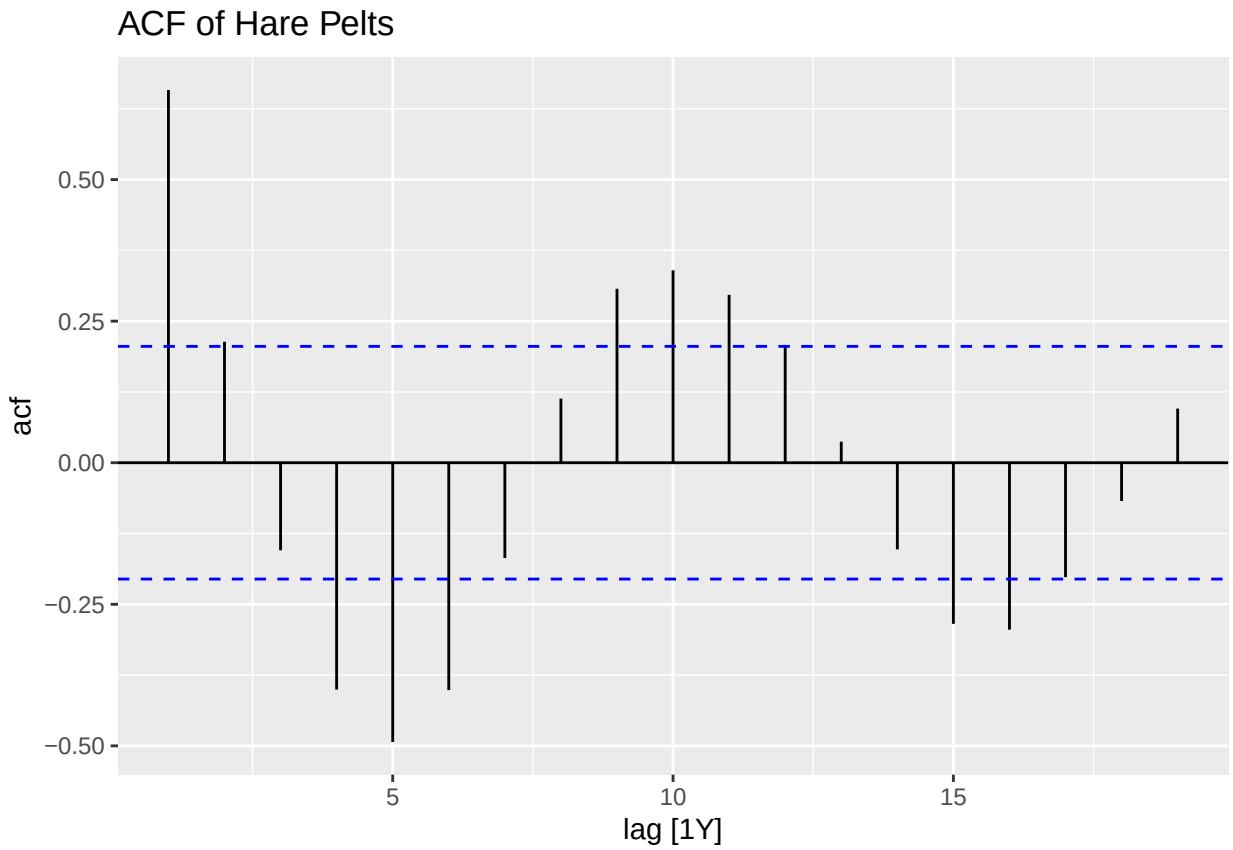

```
# Subseries plot for further seasonality exploration
gg_subseries(pelt_tsibble, Hare) +
  ggtitle("Subseries Plot of Hare Pelts") +
  ylab("Number of Pelts") + xlab("Month")
```




```
#  Lag plot to examine correlation between lags
gg_lag(pelt_tsibble, Hare, geom = "point") +
  ggtitle("Lag Plot of Hare Pelts")
```




```
#  Autocorrelation plot to visualize cyclicity and correlation in the data
ACF(pelt_tsibble, Hare) %>%
  autoplot() +
  ggtitle("ACF of Hare Pelts")
```



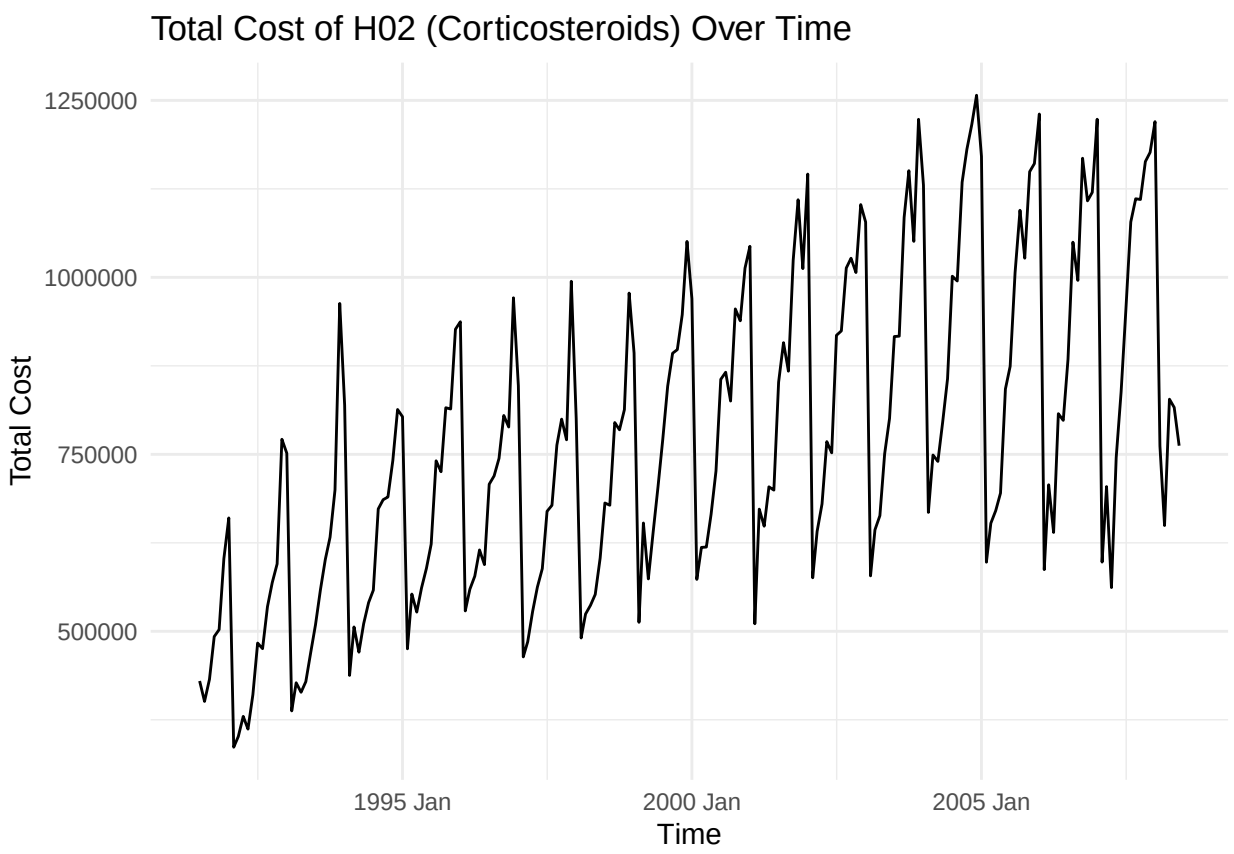
4. Explore "H02" Cost from PBS

```
PBS_H02 <- PBS %>%
  filter(ATC2 == "H02")

H02_summary <-PBS_H02 %>%
  index_by(Month) %>%  # Replace with the appropriate time index like Quarter or Year
  summarise(total_cost = sum(Cost, na.rm = TRUE))
H02_summary
```

```
## # A tsibble: 204 x 2 [1M]
##       Month total_cost
##       <mth>      <dbl>
##  1 1991 Jul     429795
##  2 1991 Aug     400906
##  3 1991 Sep     432159
##  4 1991 Oct     492543
##  5 1991 Nov     502369
##  6 1991 Dec     602652
##  7 1992 Jan     660119
##  8 1992 Feb     336220
##  9 1992 Mar     351348
## 10 1992 Apr     379808
## # i 194 more rows
```

```
ggplot(H02_summary, aes(x = Month, y = total_cost)) +  # Adjust `Month` to the appropriate time index
  geom_line() +
```

```
  labs(
    title = "Total Cost of H02 (Corticosteroids) Over Time",
    x = "Time",
    y = "Total Cost"
  ) +
  theme_minimal()
```

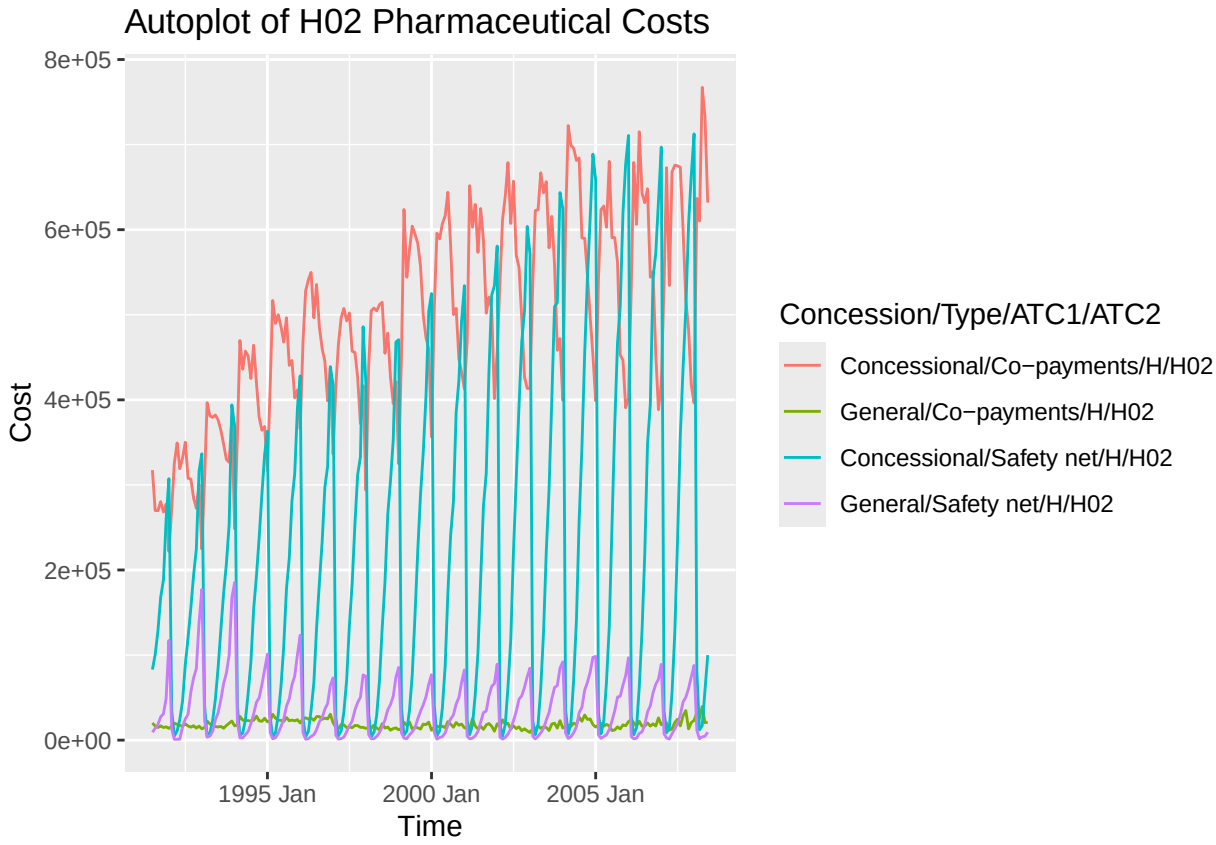



```
# Filter for "H02" category

PBS_H02 <- PBS %>%
  filter(ATC2 == "H02")
PBS_H02
```

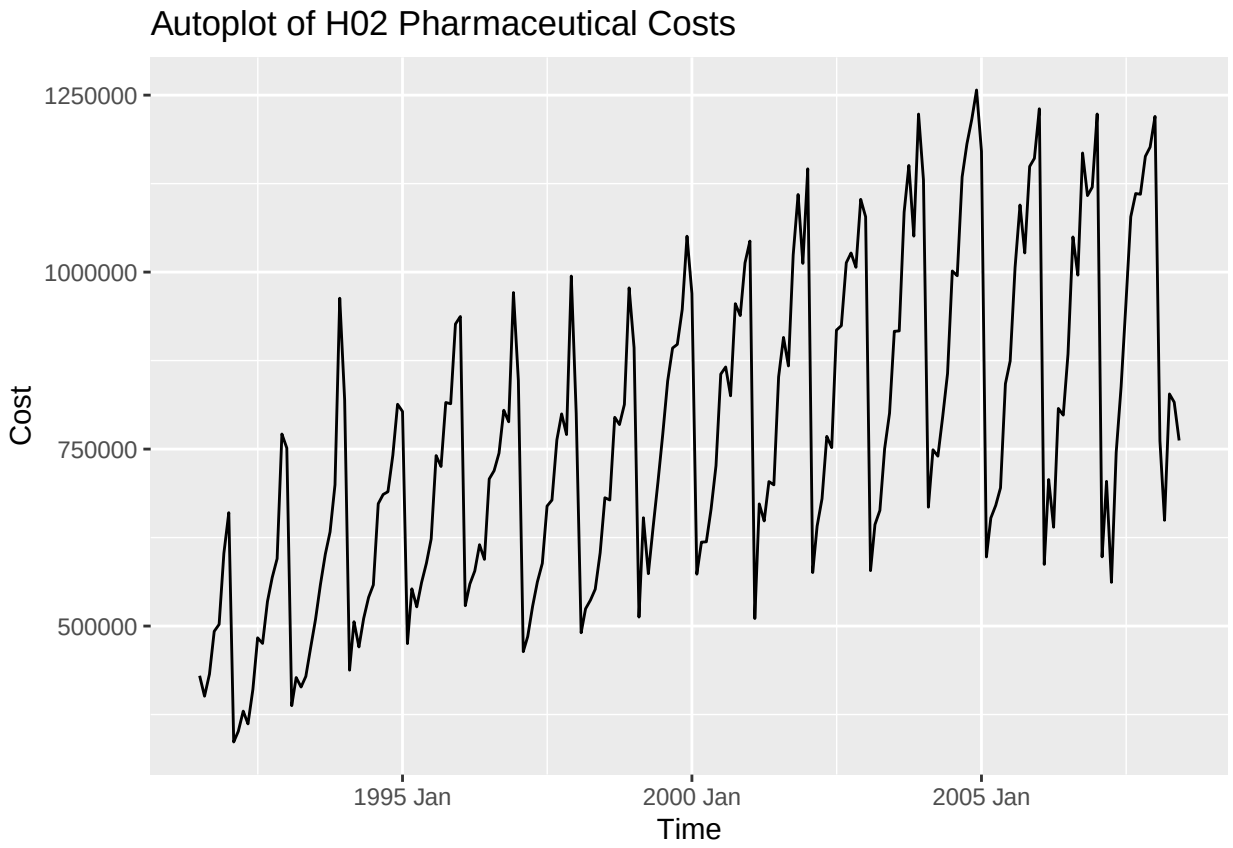
```
## # A tsibble: 816 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [4]
##       Month Concession   Type     ATC1  ATC1_desc ATC2  ATC2_desc Scripts   Cost
##       <mth> <chr>        <chr>    <chr> <chr>     <chr> <chr>       <dbl>  <dbl>
##  1 1991 Jul Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   63261 317384
##  2 1991 Aug Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   53528 269891
##  3 1991 Sep Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   52822 269703
##  4 1991 Oct Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   54016 280418
##  5 1991 Nov Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   49281 268070
##  6 1991 Dec Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   51798 277139
##  7 1992 Jan Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   42436 221772
##  8 1992 Feb Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   52913 272345
##  9 1992 Mar Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   62908 325700
## 10 1992 Apr Concessional Co-paym~ H     Systemic~ H02   CORTICOS~   68499 349271
```

```
## # i 806 more rows
```

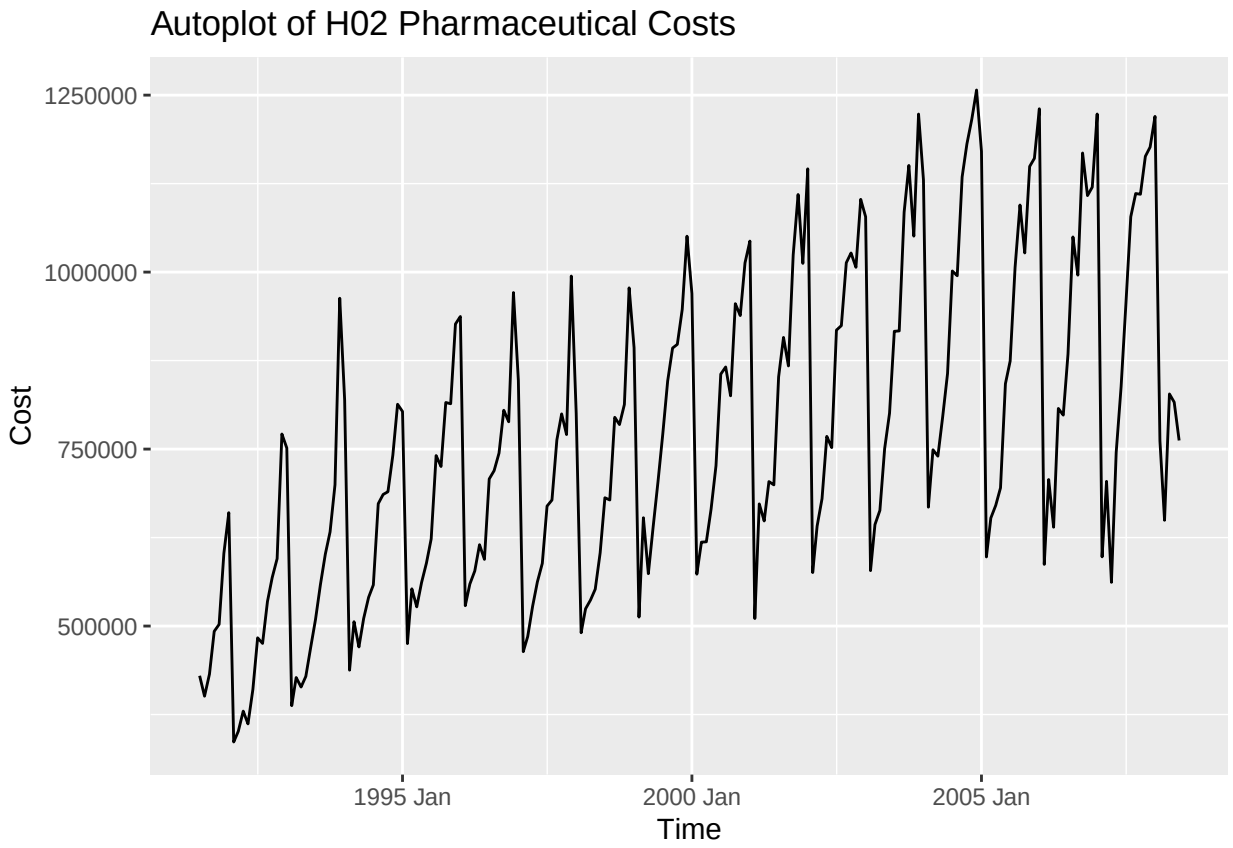
```
#  Autoplot of Cost for "H02"
autoplot(PBS_H02, Cost) +
  ggtitle("Autoplot of H02 Pharmaceutical Costs") +
  ylab("Cost") + xlab("Time")
```



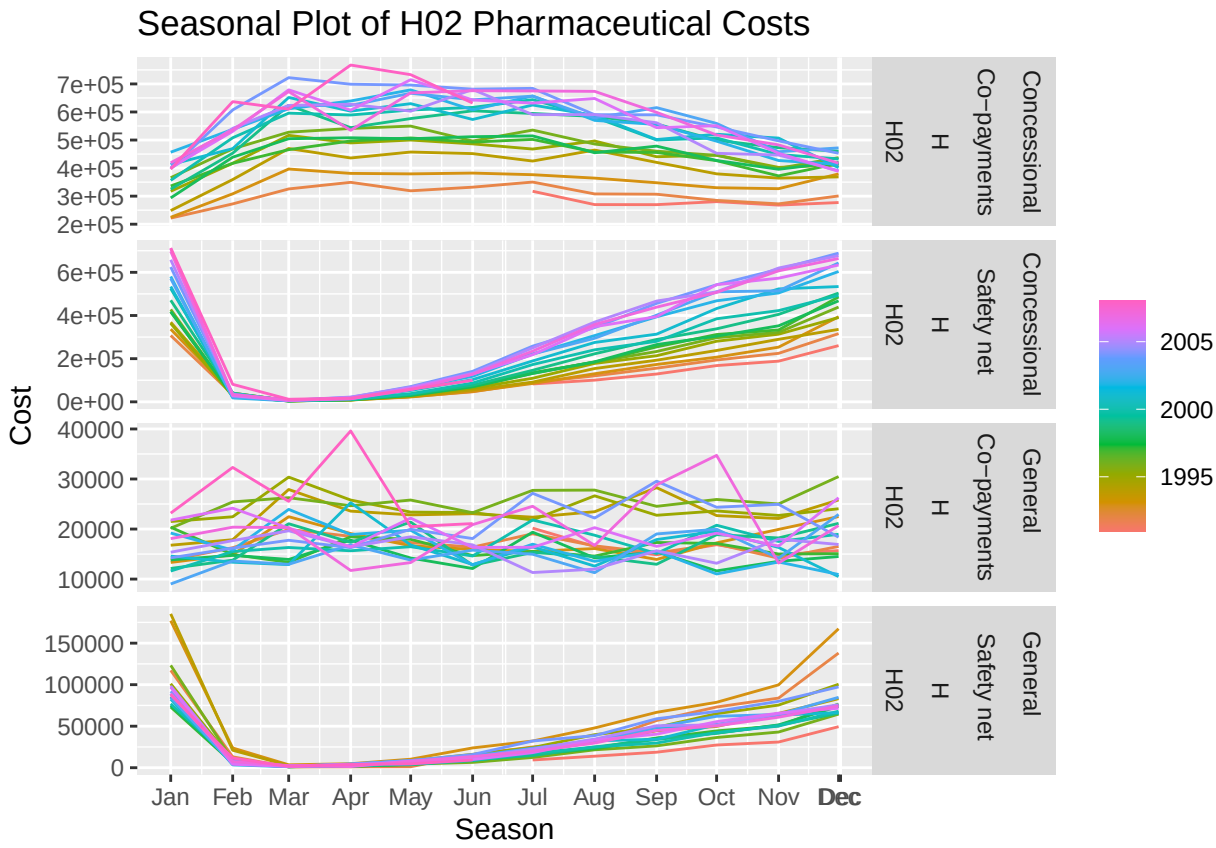

```
autoplot(H02_summary,  total_cost) +
  ggtitle("Autoplot of H02 Pharmaceutical Costs") +
  ylab("Cost") + xlab("Time")
```



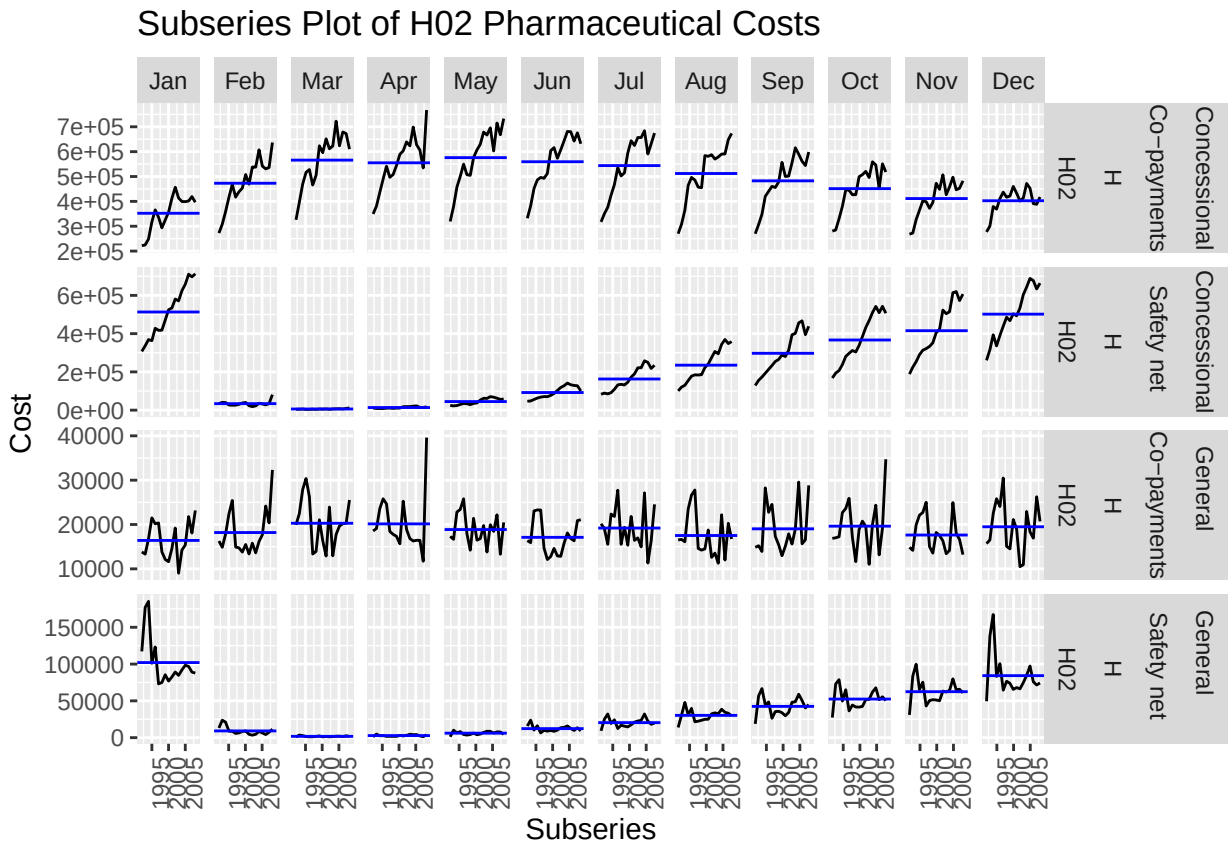

```
autoplot(H02_summary,  total_cost) +
  ggtitle("Autoplot of H02 Pharmaceutical Costs") +
  ylab("Cost") + xlab("Time")
```



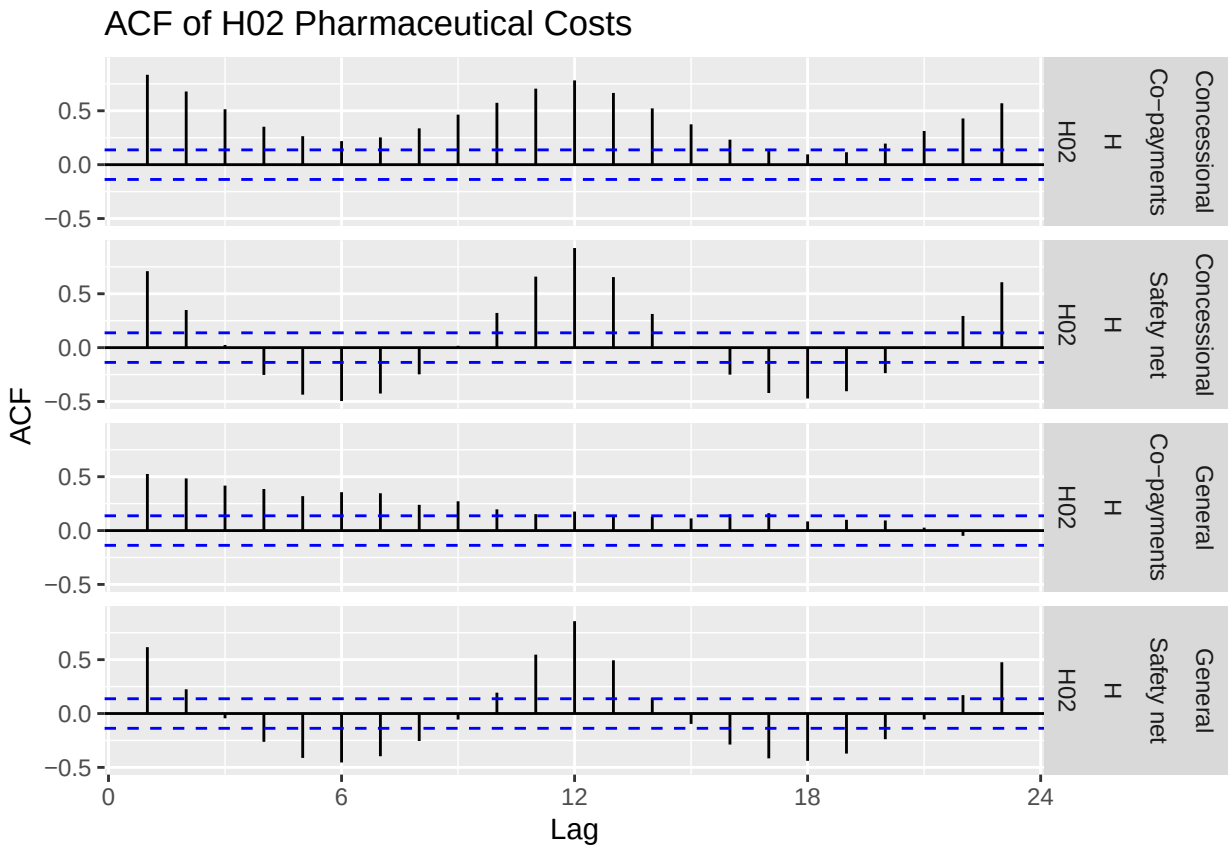

```
PBS_H02_filled <- PBS_H02 %>%
  fill_gaps(Cost = NA)

# 1. gg_season() - Seasonal plot
gg_season(PBS_H02_filled, Cost) +
  ggtitle("Seasonal Plot of H02 Pharmaceutical Costs") +
  ylab("Cost") + xlab("Season")
```



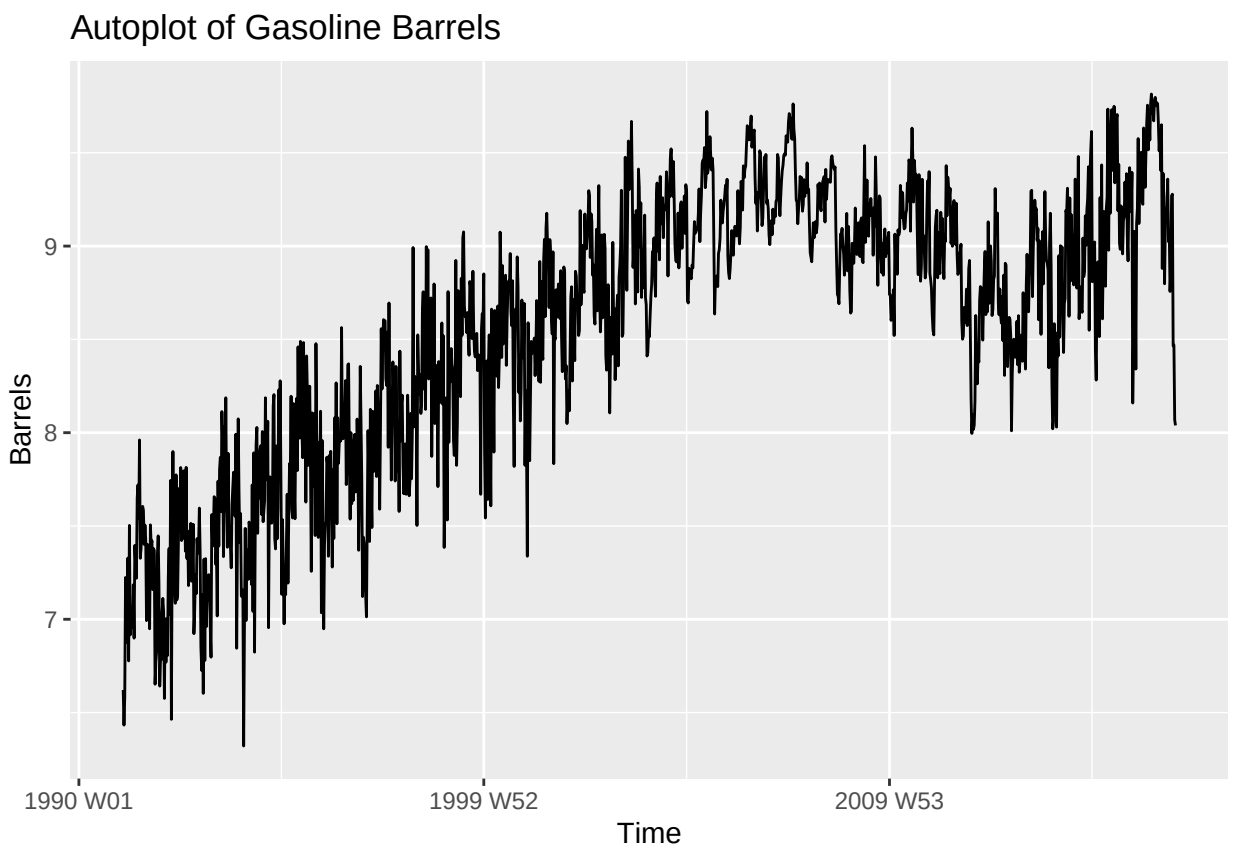

```
# 2. gg_subseries() - Subseries plot
gg_subseries(PBS_H02_filled, Cost) +
  ggtitle("Subseries Plot of H02 Pharmaceutical Costs") +
  ylab("Cost") + xlab("Subseries")
```



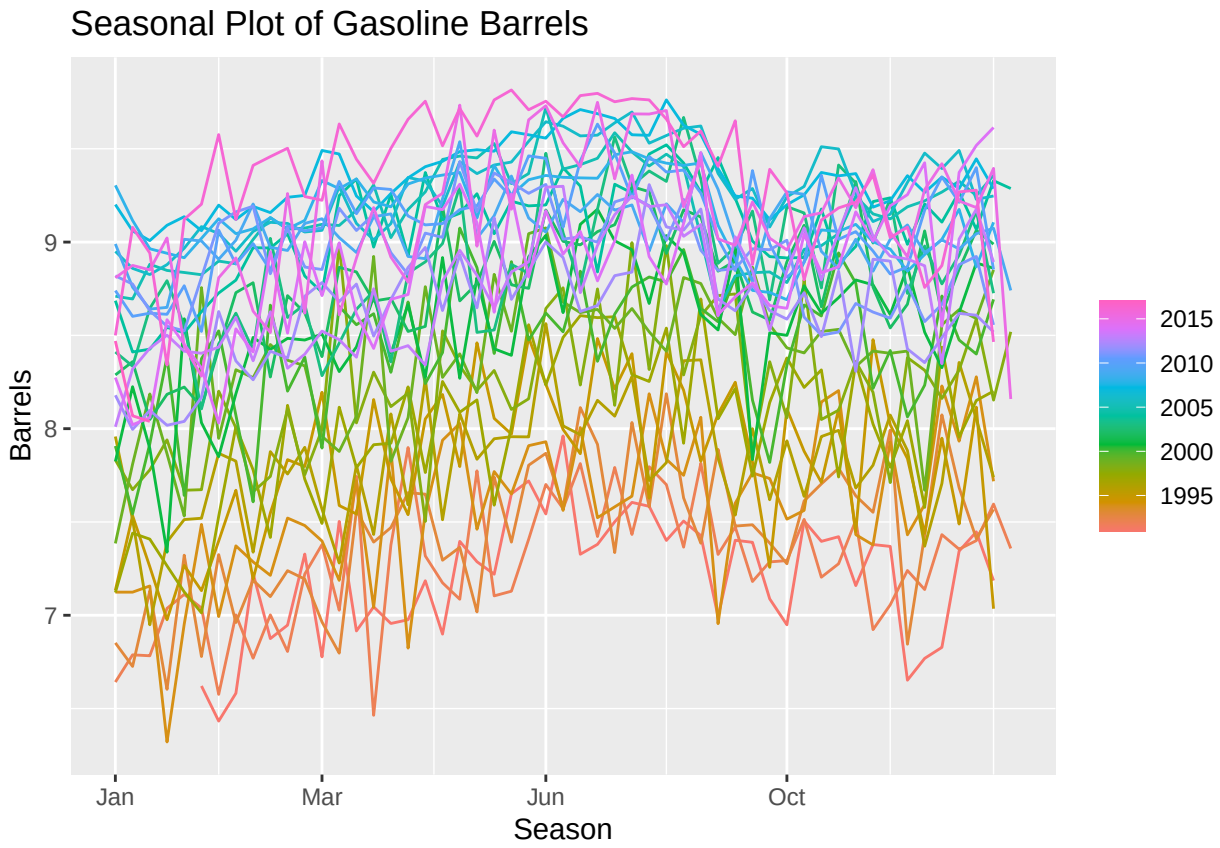

```
# 4. ACF() - Autocorrelation function
PBS_H02_filled %>%
  ACF(Cost) %>%
  autoplot() +
  ggtitle("ACF of H02 Pharmaceutical Costs") +
  ylab("ACF") + xlab("Lag")
```



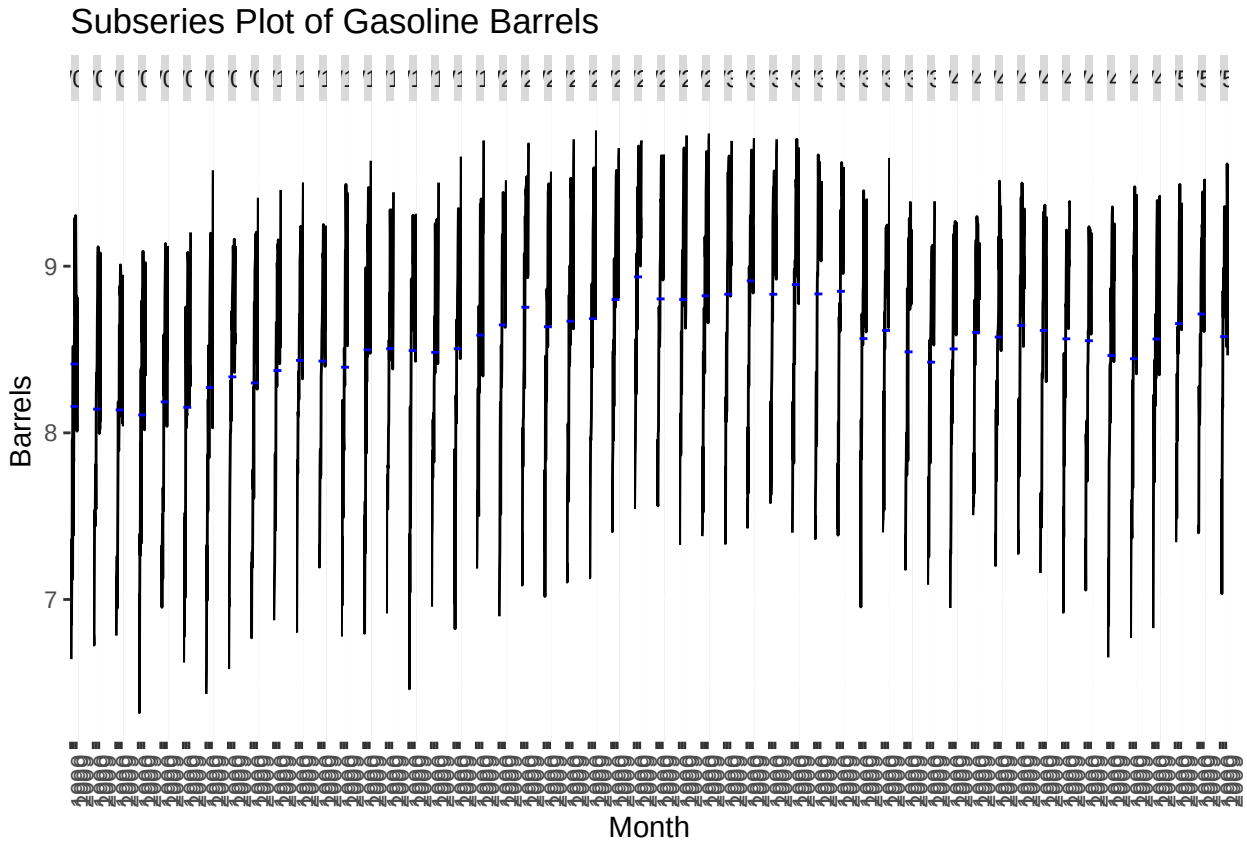

```
# 5. Explore Barrels from us_gasoline
autoplot(us_gasoline, Barrels) +
  ggtitle("Autoplot of Gasoline Barrels") +
  ylab("Barrels") + xlab("Time")
```



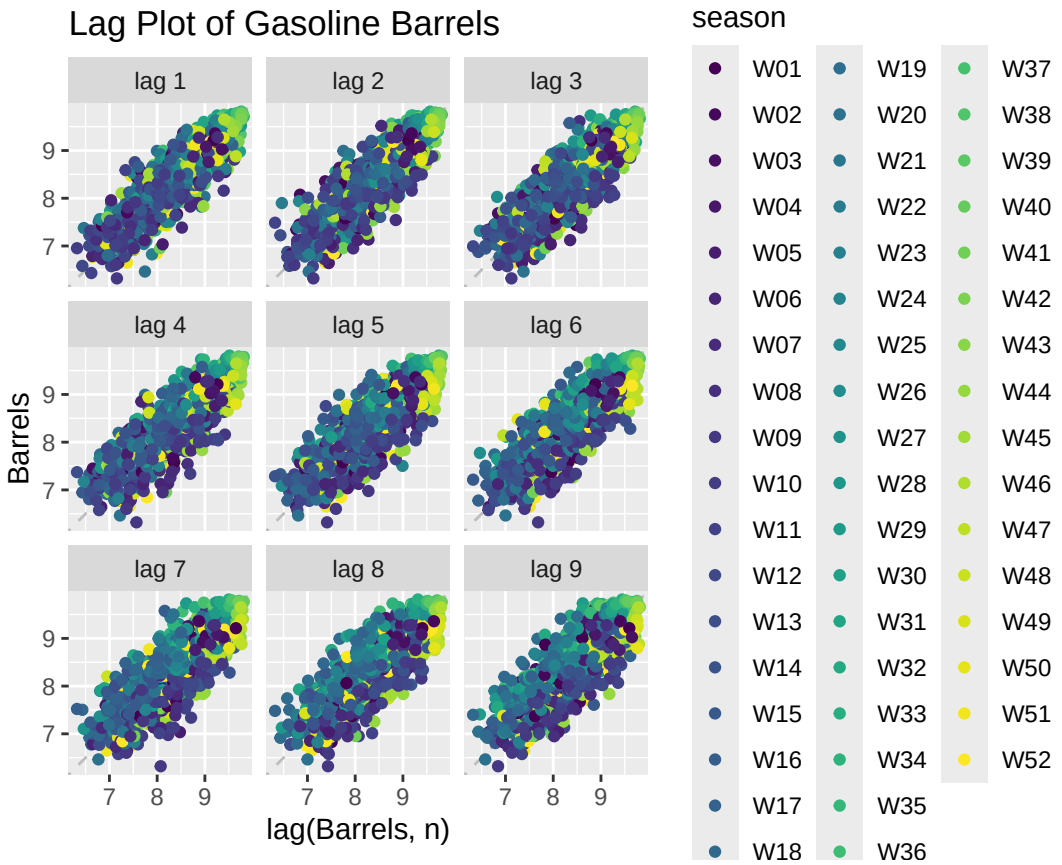

```
gg_season(us_gasoline, Barrels) +
  ggtitle("Seasonal Plot of Gasoline Barrels") +
  ylab("Barrels") + xlab("Season")
```



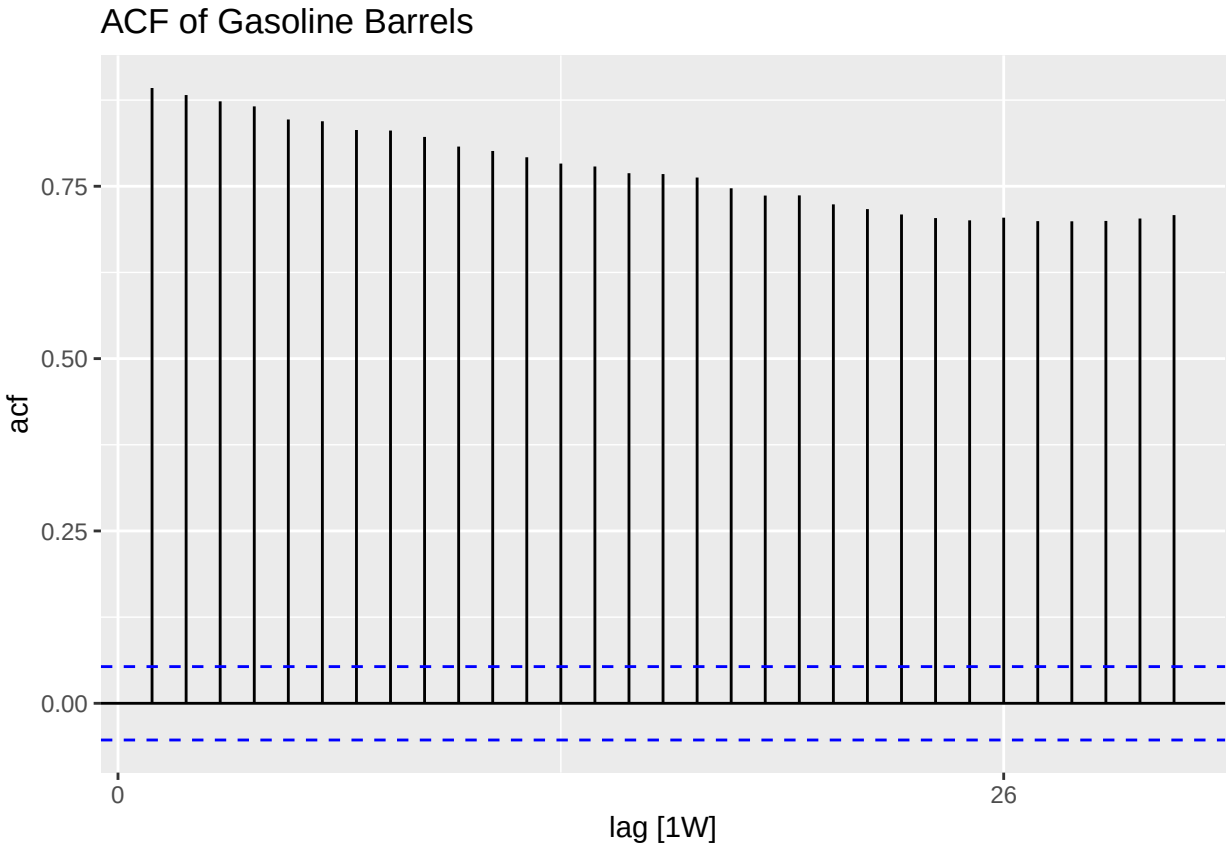

```
gg_subseries(us_gasoline, Barrels) +
  ggtitle("Subseries Plot of Gasoline Barrels") +
  ylab("Barrels") + xlab("Month")
```




```
gg_lag(us_gasoline, Barrels, geom = "point") +
  ggtitle("Lag Plot of Gasoline Barrels")
```




```
ACF(us_gasoline, Barrels) %>%
  autoplot() +
  ggtitle("ACF of Gasoline Barrels")
```


ACF of Gasoline Barrels
acf
0.75
0.50
0.25
0.00
0
26
lag [1W]