Data 624 Week2

Souleymane Doumbia

2024-09-18

Libraries

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.3.3
```

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

```
## -- Attaching packages --------------------------------------- fpp3 1.0.0 --
```

```
## v tibble      3.2.1     v tsibble     1.1.5
## v dplyr       1.1.3     v tsibbledata 0.4.1
## v tidyr       1.3.0     v feasts      0.3.2
## v lubridate   1.9.3     v fable       0.3.4
## v ggplot2     3.5.1     v fabletools  0.4.2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## Warning: package 'tsibble' was built under R version 4.3.3
```

```
## Warning: package 'feasts' was built under R version 4.3.2
```

```
## Warning: package 'fabletools' was built under R version 4.3.2
```

```
## Warning: package 'fable' was built under R version 4.3.2
```

```
## -- 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(USgas) # for Exercise 2.4
```

Exercise 2.1

Exploring the datasets using help()

```
help(aus_production)
head(aus_production)
```

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
```

```
help(pelt)
head(pelt)
```

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
```

```
help(gafa_stock)
head(gafa_stock)
```

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
```

```
help(vic_elec)
head(vic_elec)
```

vic_elec

```
## # A tsibble: 6 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
```

Inspecting the time interval of each series

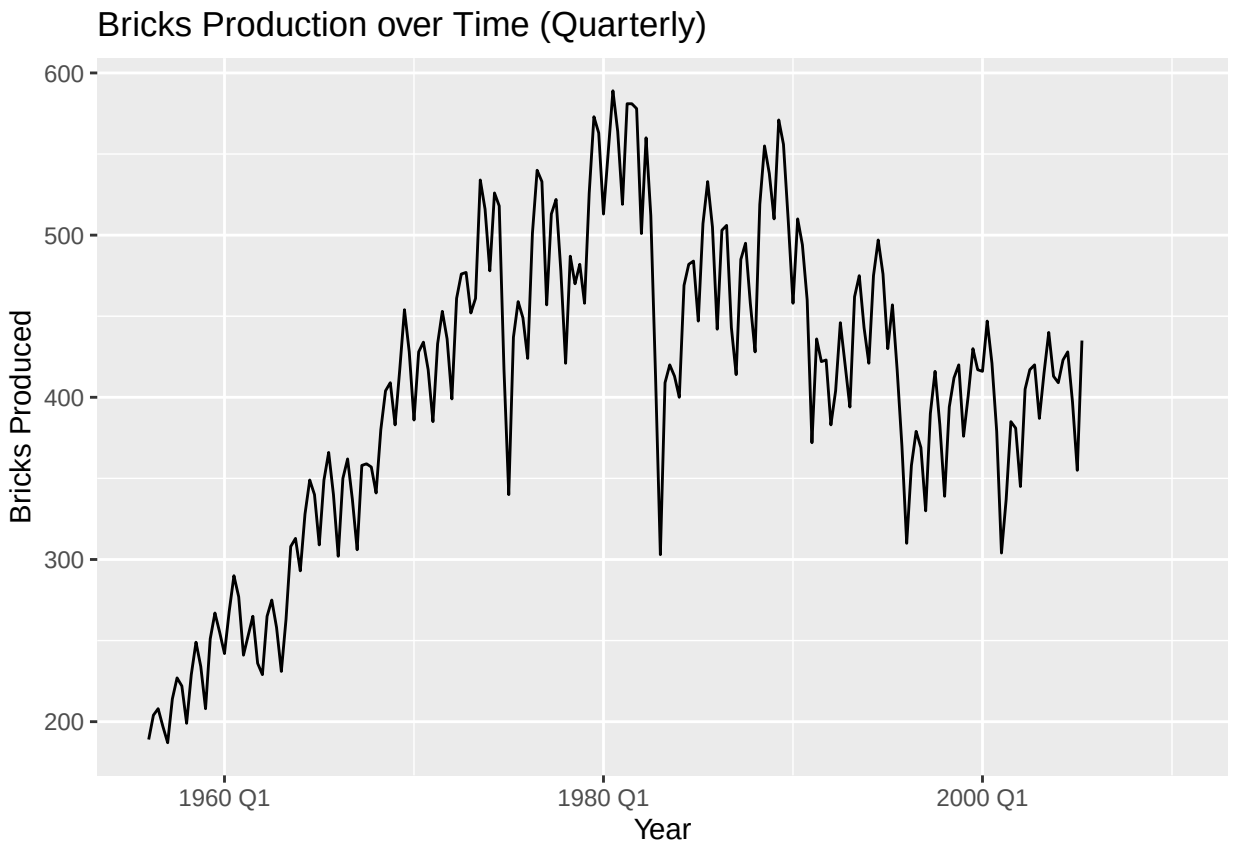
By inspecting the first rows of each dataset above, The Time Interval for each of them is:

- *Bricks from aus_production*: Quarter
- *Lynx from pelt*: Year
- *Close from gafa_stock*: Day
- *Demand from vic_elec*: Half hour (30 minutes)

Plotting time series using autoplot()

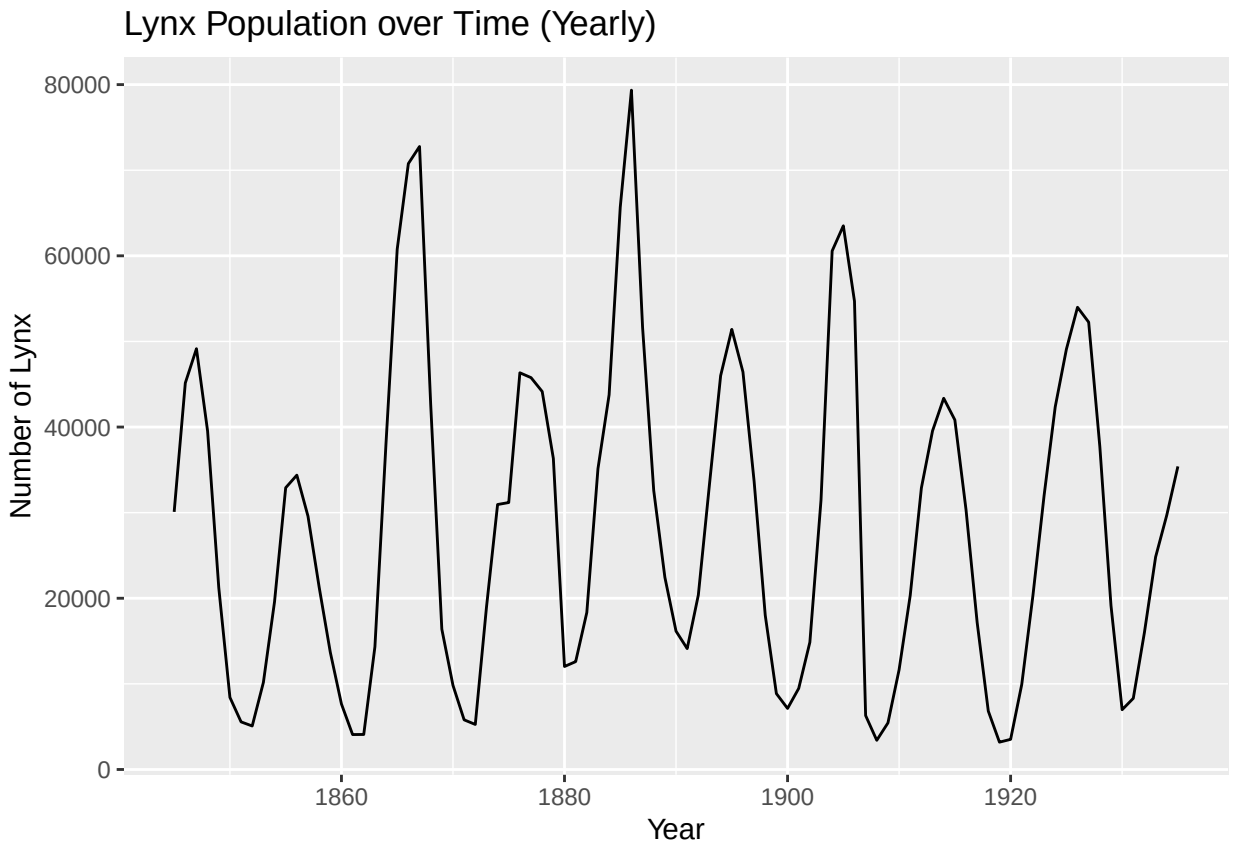
```
# Time plot for Bricks from aus_production
autoplot(aus_production, Bricks) +
  labs(title = "Bricks Production over Time (Quarterly)",
       y = "Bricks Produced",
       x = "Year")
```

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

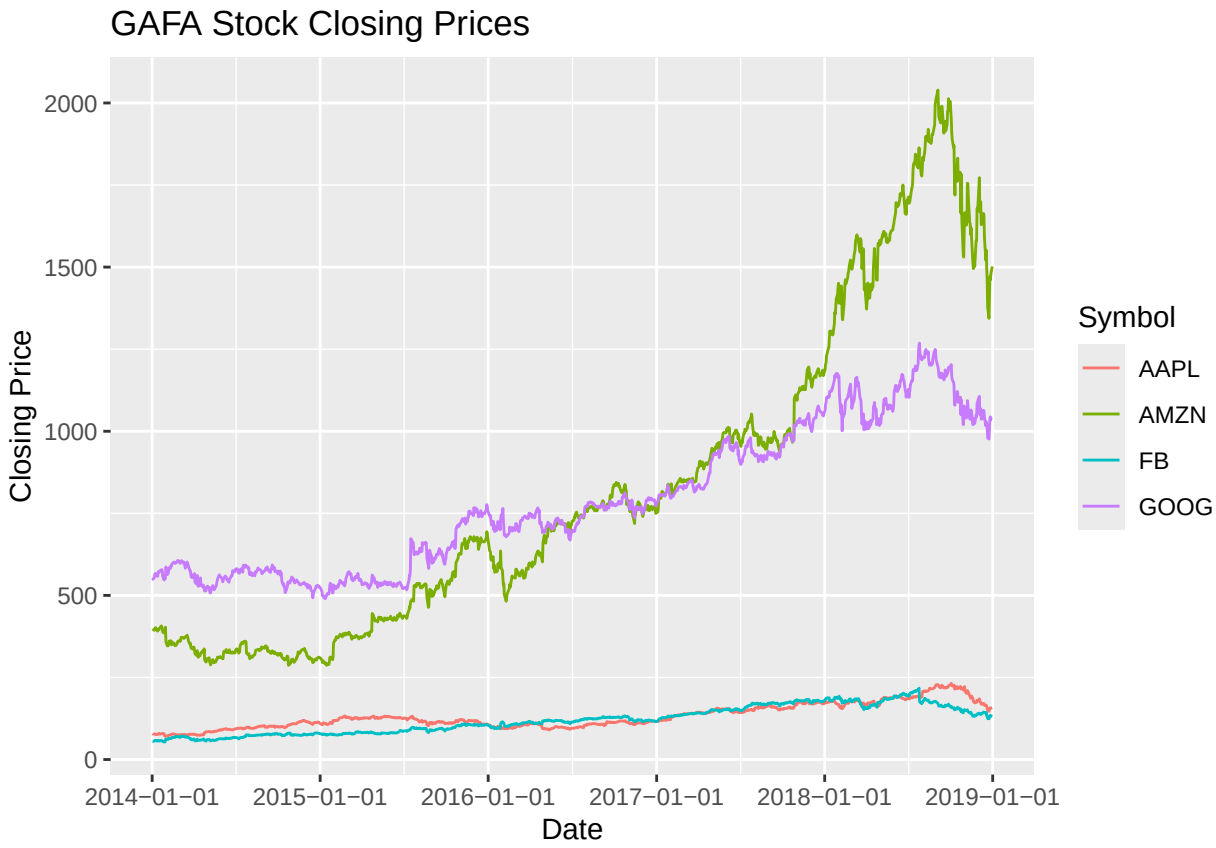



```
# Time plot for Lynx from pelt
autoplot(pelt, Lynx) +
  labs(title = "Lynx Population over Time (Yearly)",
```

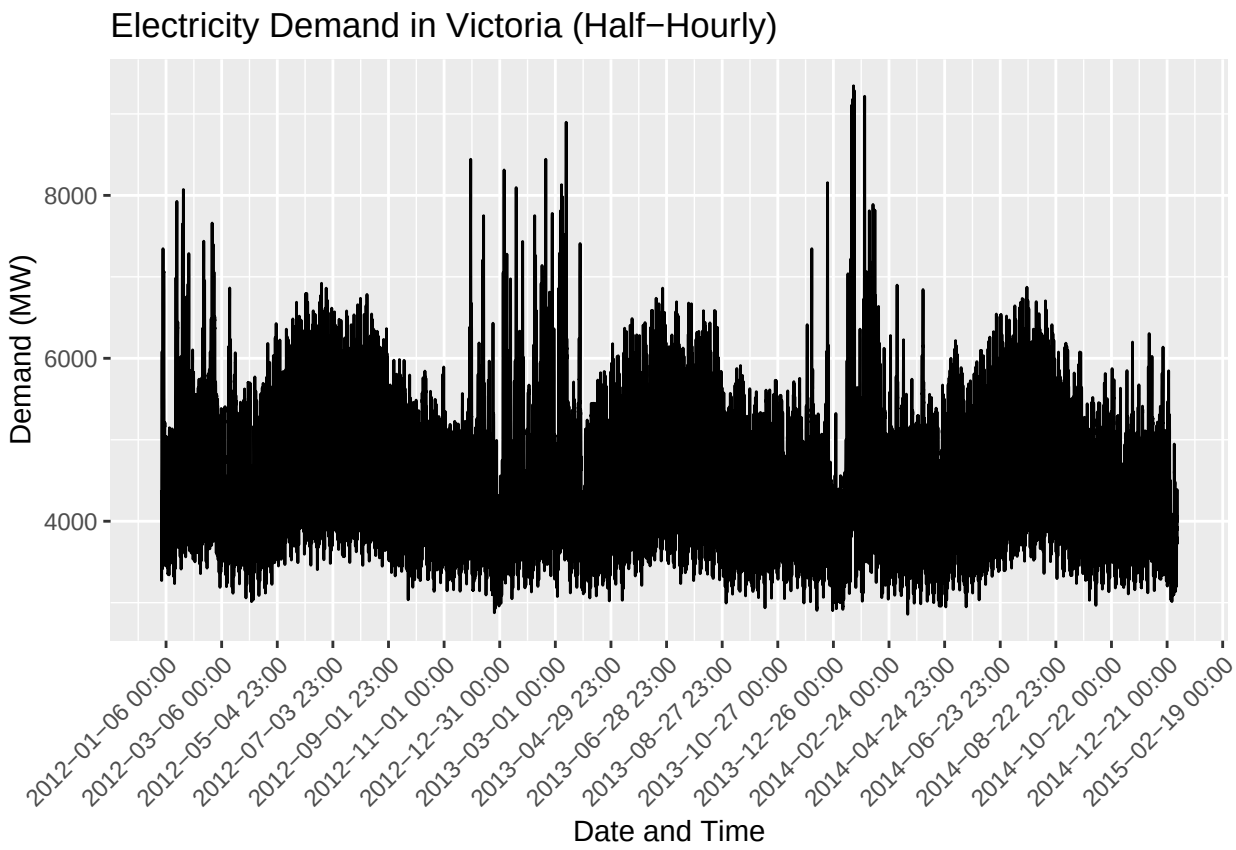
```
       y = "Number of Lynx",
       x = "Year")
```




```
# Time plot for Close from gafa_stock
autoplot(gafa_stock, Close) +
  labs(title = "GAFA Stock Closing Prices",
       y = "Closing Price",
       x = "Date") +
  scale_x_date(date_labels = "%Y-%m-%d",
               date_breaks = "1 year")
```




```
# Time plot for Demand from vic_elec
autoplot(vic_elec, Demand) +
  labs(title = "Electricity Demand in Victoria (Half-Hourly)",
       y = "Demand (MW)",
       x = "Date and Time") +
  scale_x_datetime(date_labels = "%Y-%m-%d %H:%M",
                   date_breaks = "60 day") +
  theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1))
```



Exercise 2.2

Identifying the peak closing prices for each stock

```
peak_closing_prices <- gafa_stock %>%
  as_tibble() %>%  # Convert the data to a regular tibble to avoid grouping issues
  group_by(Symbol) %>%
  summarise(Peak_Close = max(Close, na.rm = TRUE))

peak_closing_prices
```

```
## # A tibble: 4 x 2
##   Symbol Peak_Close
##   <chr>       <dbl>
## 1 AAPL         232.
## 2 AMZN        2040.
## 3 FB           218.
## 4 GOOG        1268.
```

Finding the day corresponding to the peak closing price for each stock

```
peak_days <- gafa_stock %>%
  as_tibble() %>%
  inner_join(peak_closing_prices, by = "Symbol") %>%
  filter(Close == Peak_Close)

peak_days %>%
```

```
  select(Date, Symbol, Close)
```

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

Exercise 2.3: File tute1.csv

Reading the data into R

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

```
## 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.
```

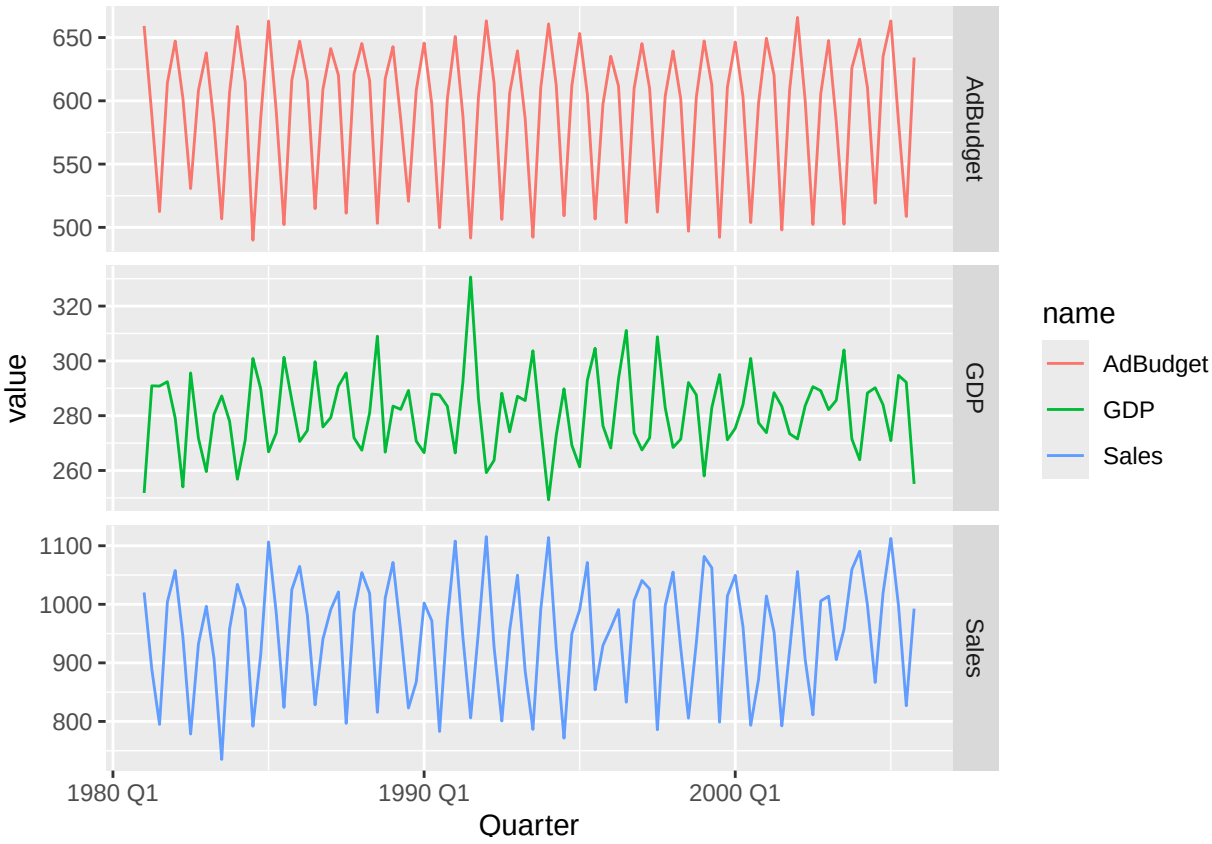
```
#View(tute1)
```

Converting the data to time series

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

Constructing 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")
```



Exercise 2.4:

Creating a tsibble from us_total

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

glimpse(us_total_tsibble)
```

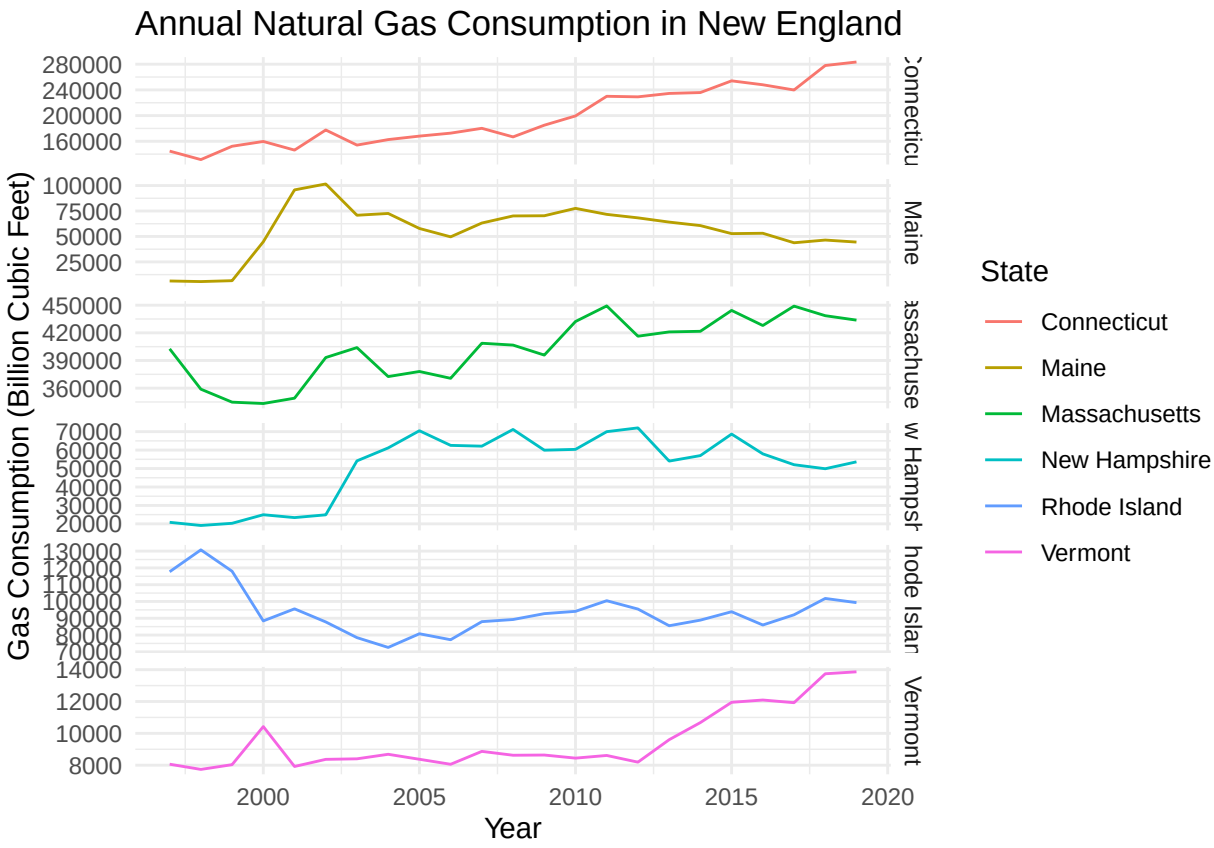
```
## Rows: 1,266
## Columns: 3
## Key: state [53]
## $ year  <int> 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007~
## $ state <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"~
## $ y     <int> 324158, 329134, 337270, 353614, 332693, 379343, 350345, 382367, ~
```

Filtering data for the New England states (Maine, Vermont, New Hampshire, Massachusetts, Connecticut and Rhode Island)

```
new_england_states <- c("Maine", "Vermont", "New Hampshire", "Massachusetts", "Connecticut", "Rhode Isla

new_england_data <- us_total_tsibble %>%
  filter(state %in% new_england_states)
```

Plotting the annual natural gas consumption by state (New England States)

```
ggplot(new_england_data, aes(x = year, y = y, colour = state)) +
  geom_line() +
  labs(title = "Annual Natural Gas Consumption in New England",
       x = "Year",
       y = "Gas Consumption (Billion Cubic Feet)",
       colour = "State") +
  theme_minimal() +
  facet_grid(state ~ ., scales = "free_y")
```



Exercise 2.5:

Reading the tourism.xlsx file

```
tourism_data <- readxl::read_excel("tourism.xlsx")

head(tourism_data)
```

```
## # A tibble: 6 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.
```

Creating a tsibble

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

glimpse(tourism_tsibble)
```

```
## Rows: 24,320
## Columns: 5
## Key: Region, State, Purpose [304]
## $ Quarter <qtr> 1998 Q1, 1998 Q2, 1998 Q3, 1998 Q4, 1999 Q1, 1999 Q2, 1999 Q3,~
## $ Region  <chr> "Adelaide", "Adelaide", "Adelaide", "Adelaide", "Adelaide", "A~
## $ State   <chr> "South Australia", "South Australia", "South Australia", "Sout~
## $ Purpose <chr> "Business", "Business", "Business", "Business", "Business", "B~
## $ Trips   <dbl> 135.0777, 109.9873, 166.0347, 127.1605, 137.4485, 199.9126, 16~
```

Find the combination of Region and Purpose with the maximum average number of overnight trips

```
max_avg_trips <- tourism_tsibble %>%
  group_by(Region, Purpose) %>%
  summarise(avg_trips = mean(Trips, na.rm = TRUE)) %>%
  arrange(desc(avg_trips)) %>%
  slice(1)  # Get the row with the maximum average trips
```

```
## Warning: Current temporal ordering may yield unexpected results.
## i Suggest to sort by `Region`, `Purpose`, `Quarter` first.
```

```
max_avg_trips
```

```
## # A tsibble: 76 x 4 [1Q]
## # Key:       Region, Purpose [76]
## # Groups:    Region [76]
##    Region                    Purpose  Quarter avg_trips
##    <chr>                     <chr>      <qtr>     <dbl>
##  1 Adelaide                  Visiting 2017 Q1     270.
##  2 Adelaide Hills            Visiting 2002 Q4      81.1
##  3 Alice Springs             Holiday  1998 Q3      76.5
##  4 Australia's Coral Coast   Holiday  2014 Q3     198.
##  5 Australia's Golden Outback Business 2017 Q3     174.
##  6 Australia's North West    Business 2016 Q3     297.
##  7 Australia's South West    Holiday  2016 Q1     612.
##  8 Ballarat                  Visiting 2004 Q1     103.
##  9 Barkly                    Holiday  1998 Q3      37.9
## 10 Barossa                   Holiday  2006 Q1      51.0
## # i 66 more rows
```

Creating a new tsibble with total trips by State

```
state_total_trips <- tourism_tsibble %>%
  group_by(State) %>%
  summarise(total_trips = sum(Trips, na.rm = TRUE)) %>%
```

```
  ungroup() %>%  # Removing grouping
  as_tsibble(index = Quarter, key = State)

state_total_trips
```

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter total_trips
##    <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
```

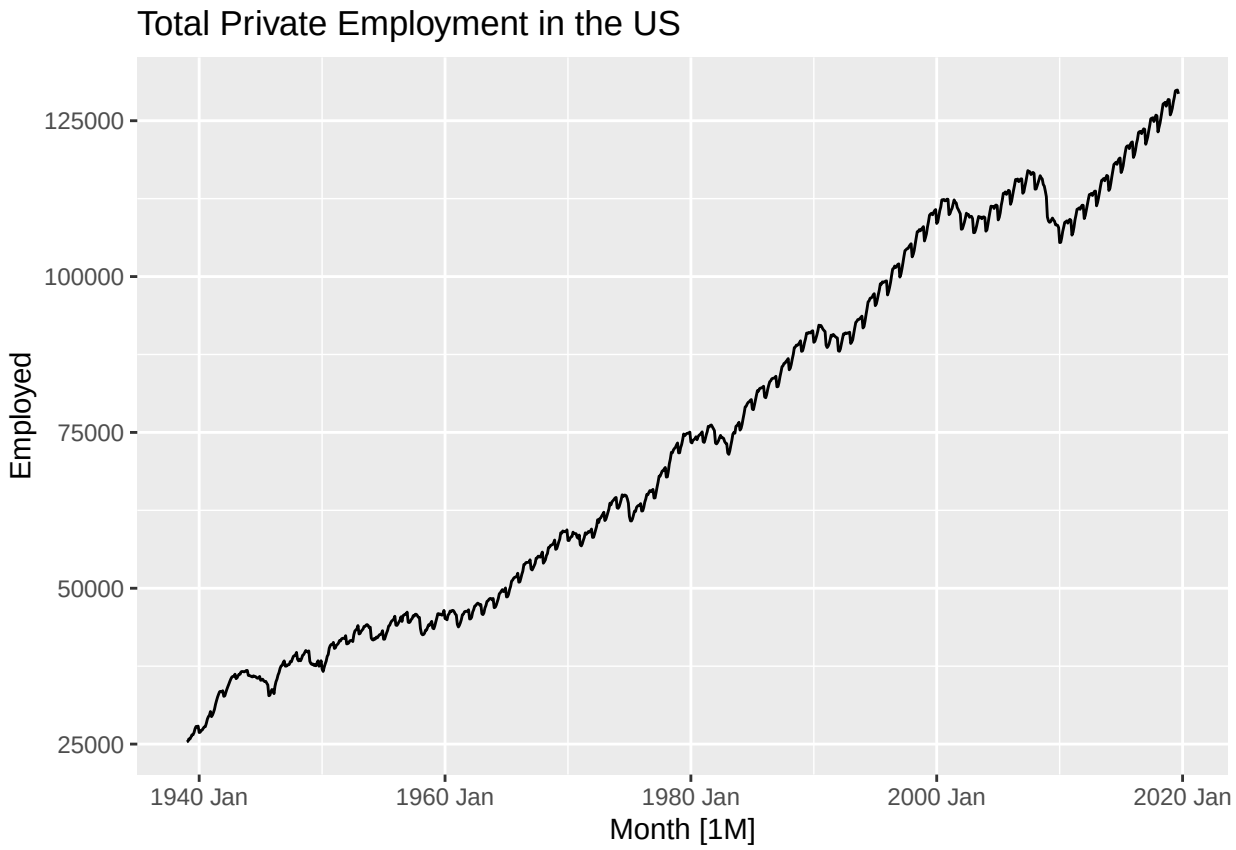
Exercise 2.8:

Loading and Subsetting the datasets

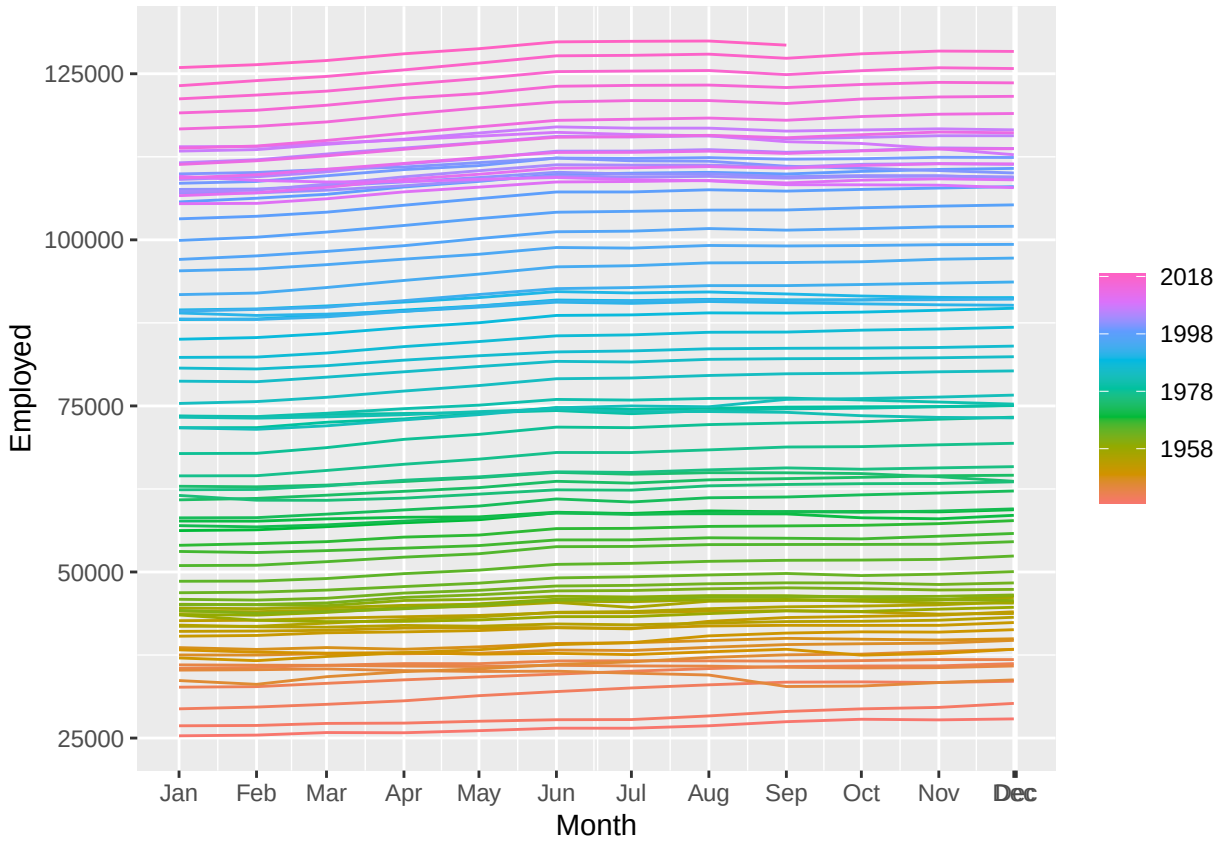
```
us_employment <- us_employment
aus_production <- aus_production
pelt <- pelt
PBS <- PBS
us_gasoline <- us_gasoline

total_private <- us_employment %>% filter(Title == "Total Private")
h02 <- PBS %>% filter(ATC2 == "H02")
barrels <- us_gasoline
```

Total Private Employed (us_employment):

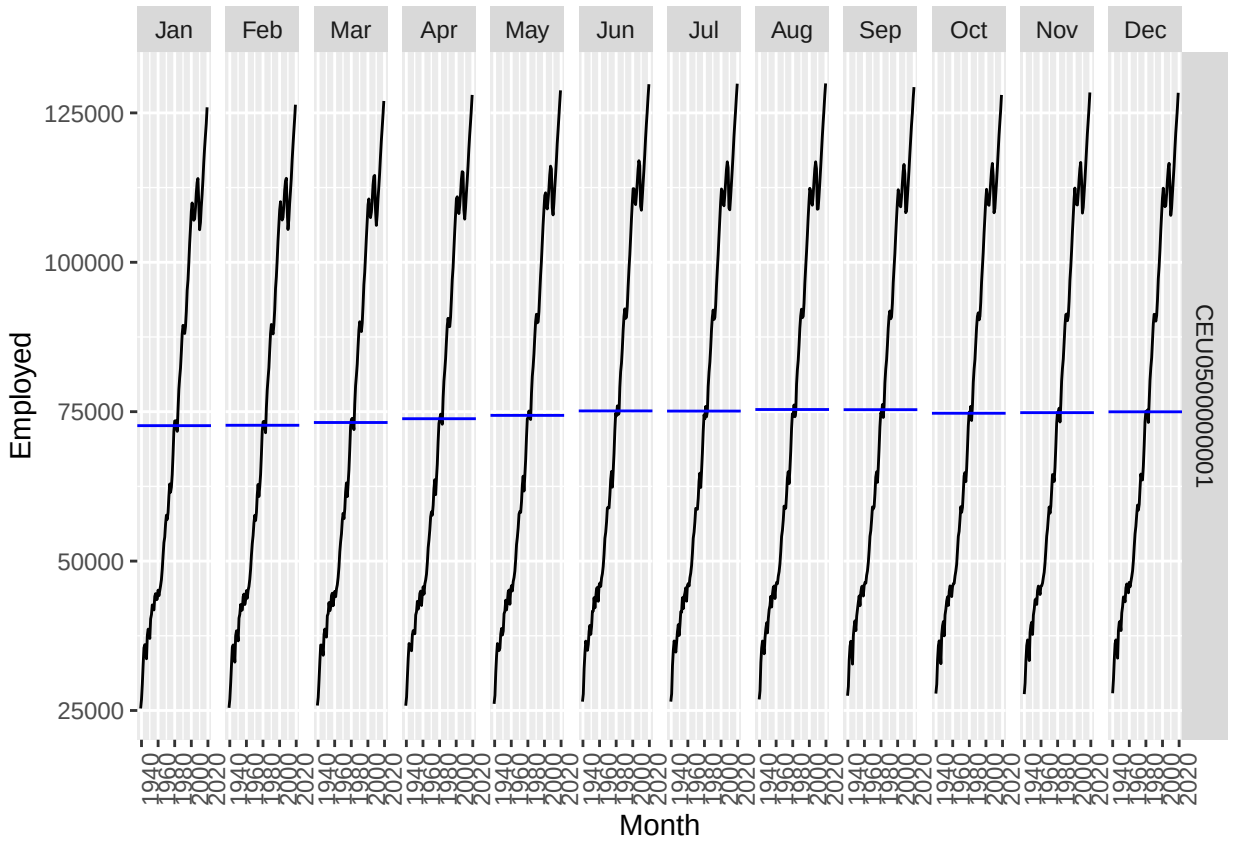
```
# Plot the series to spot trends, seasonality, and cyclicity
autoplot(total_private, Employed) +
  labs(title = "Total Private Employment in the US")
```



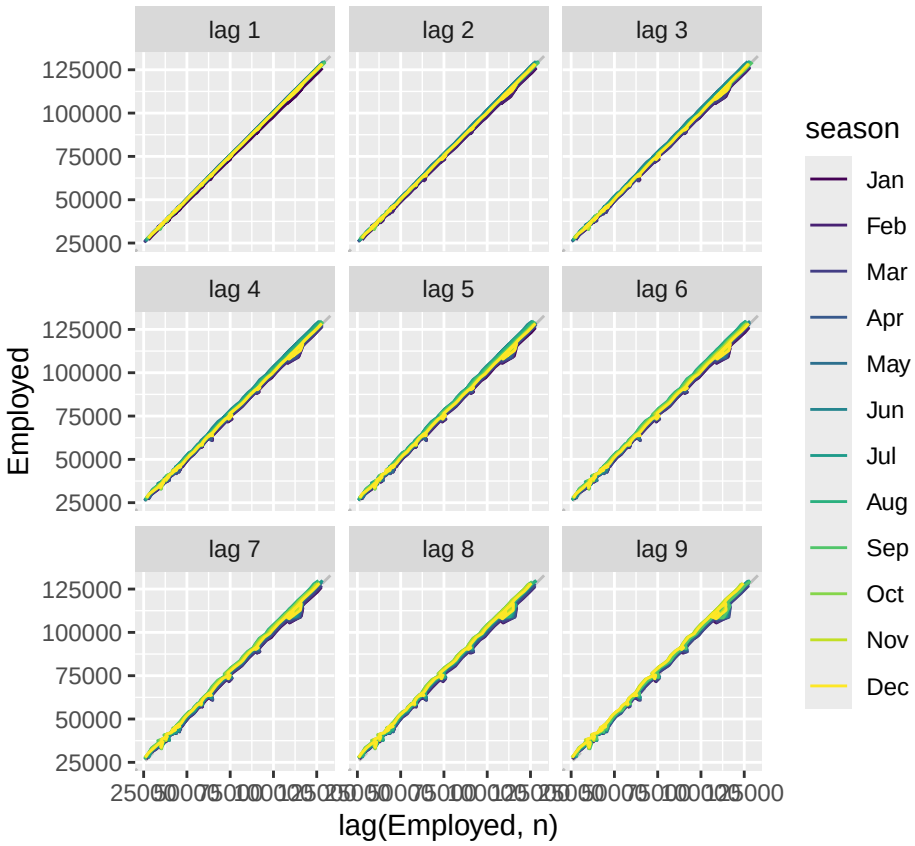

```
# Seasonality
gg_season(total_private, Employed)
```



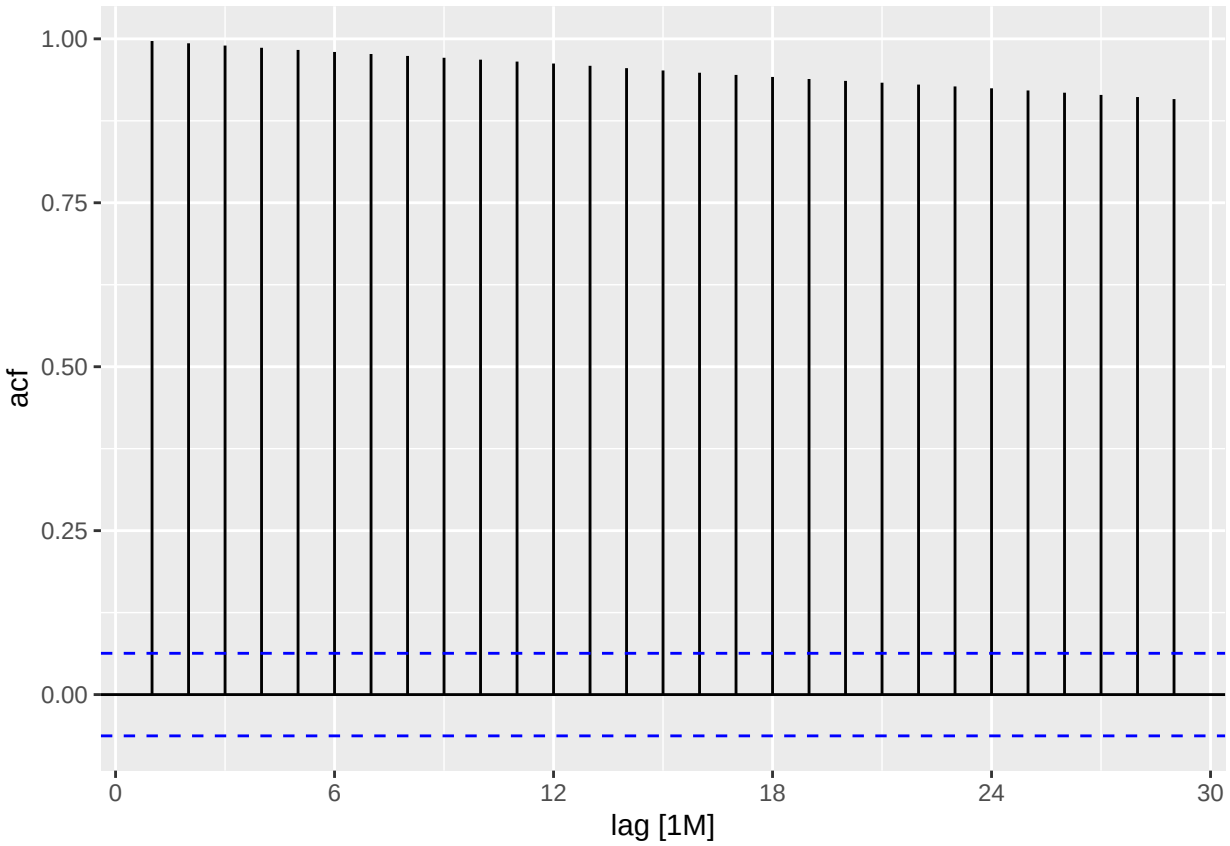

```
# Subseries plot for seasonal patterns
gg_subseries(total_private, Employed)
```




```
# Lag plot to check for autocorrelation
gg_lag(total_private, Employed)
```




```
# Autocorrelation function plot
ACF(total_private, Employed) %>% autoplot()
```



Seasonality, Cyclicity, and Trend:

- *Trend*: There is a clear upward trend in employment from 1940 to 2020.
- *Seasonality*: Strong seasonality is evident, with employment spiking in certain months each year.
- *Cyclicality*: Small dips occur during economic downturns (e.g., the 2008 recession).

What do you learn about the series? The series shows consistent growth in private employment over time, with predictable seasonal fluctuations.

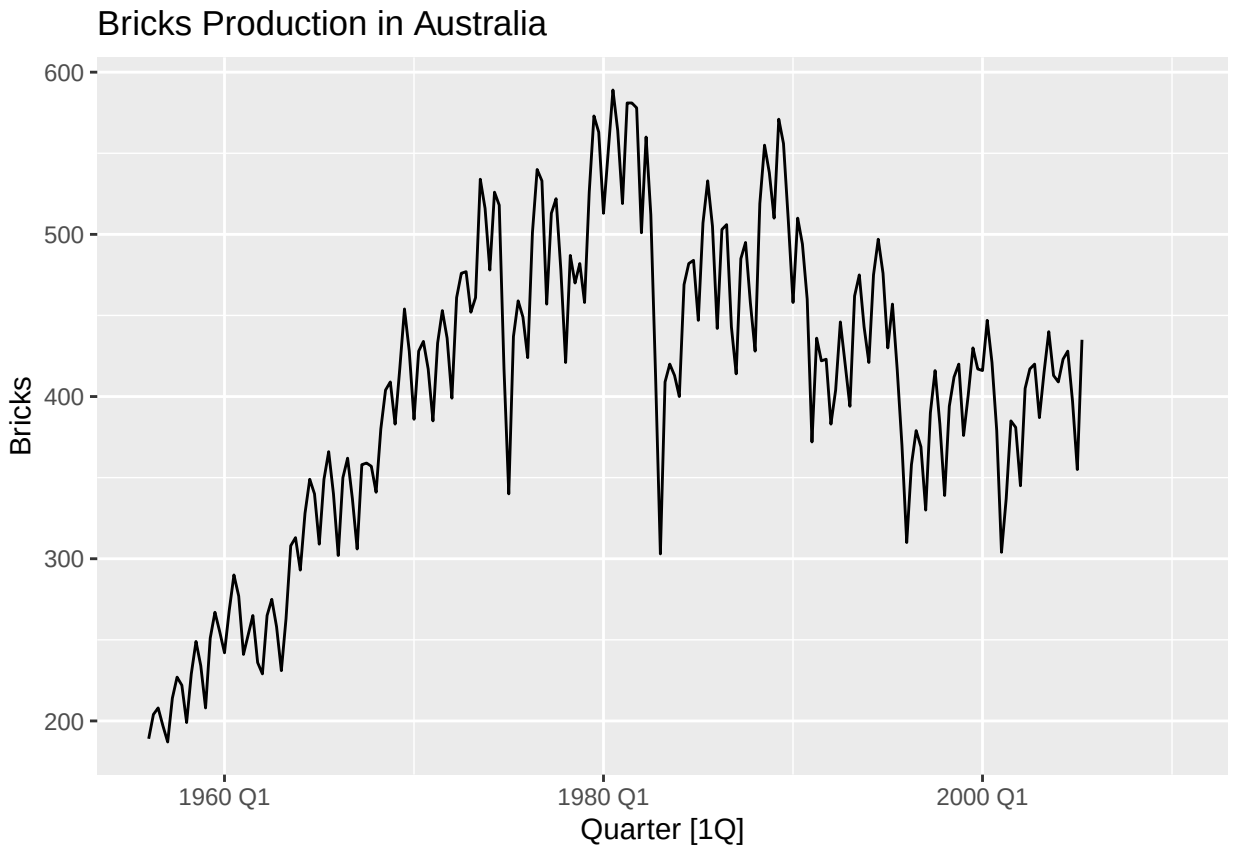
Seasonal Patterns: Employment increases during certain months (likely holiday seasons or business quarters), showing stable seasonal effects.

Unusual Years: The dip around 2008 (due to the financial crisis) stands out, but overall, the series shows steady growth.

Bricks (aus_production)

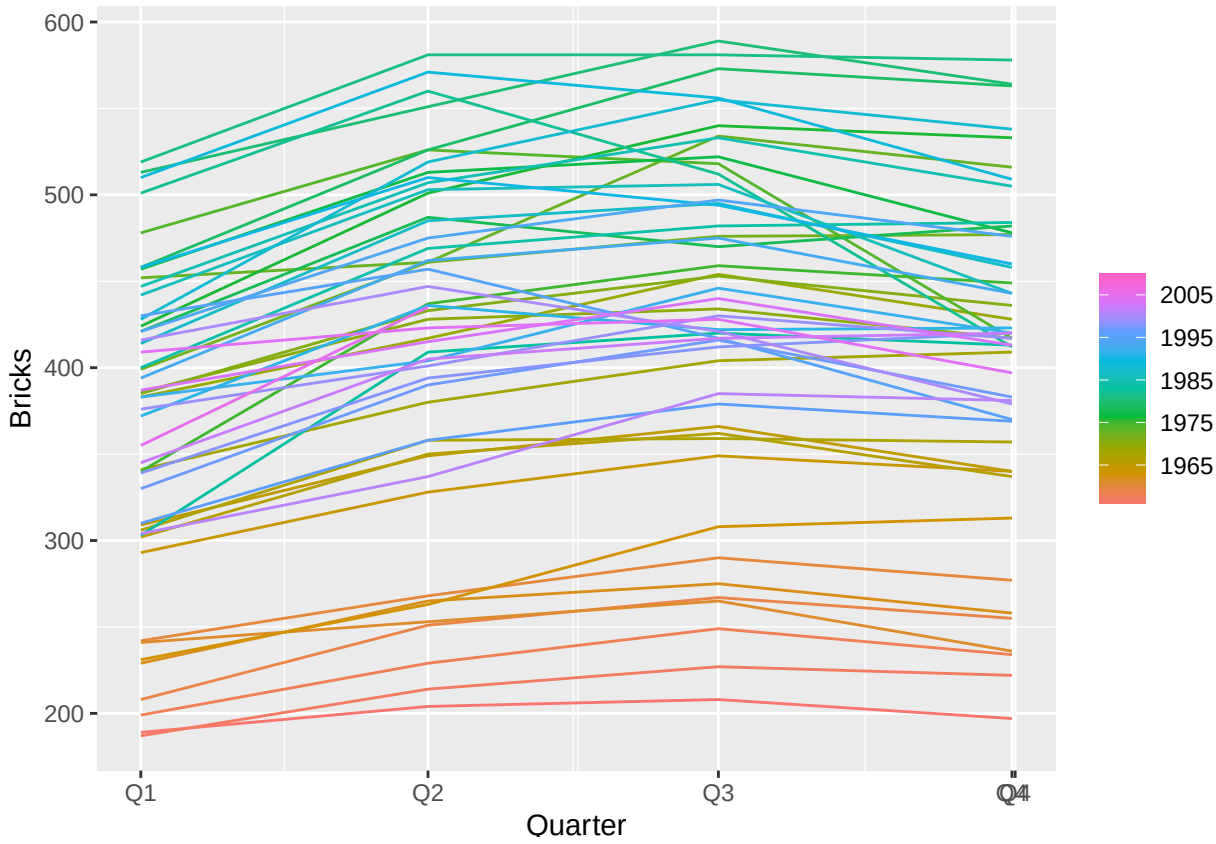
```
# Plot the Bricks production series
autoplot(aus_production, Bricks) +
  labs(title = "Bricks Production in Australia")
```

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



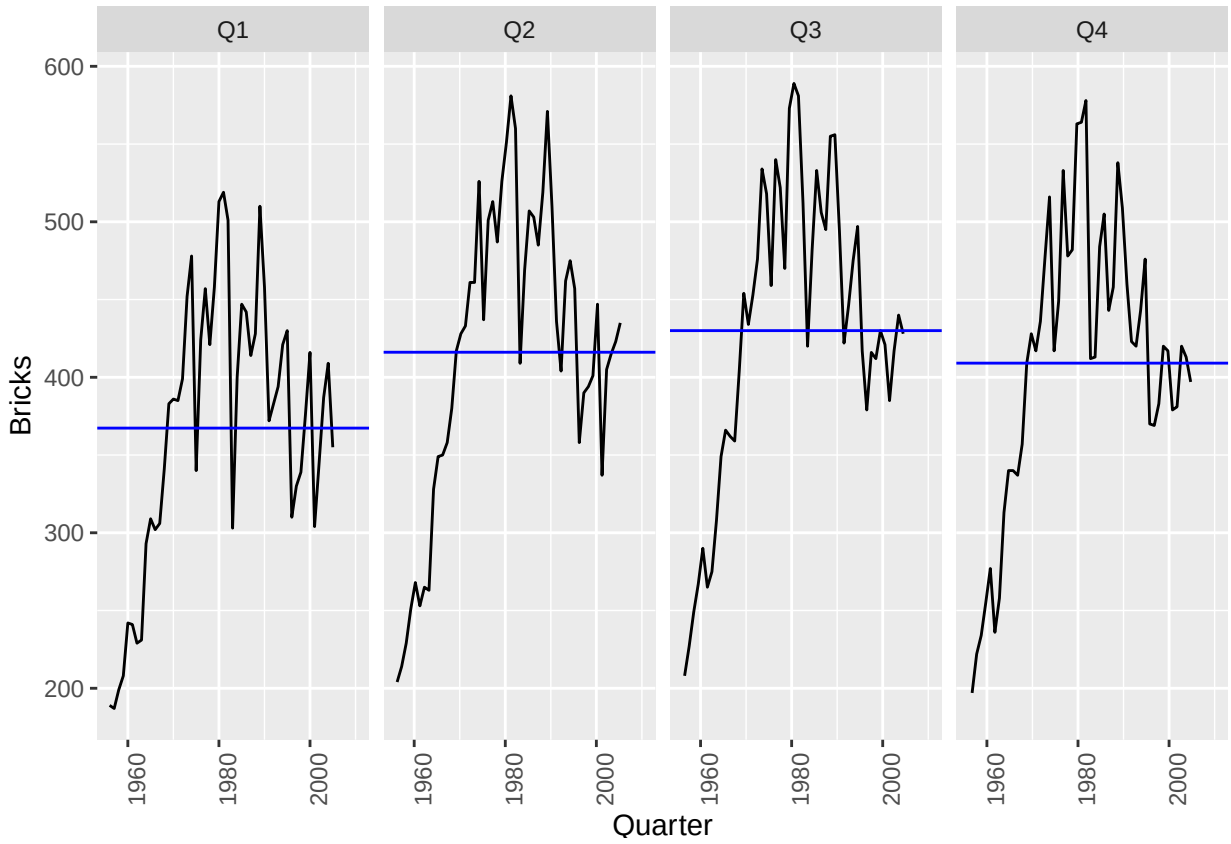

```
# Seasonality
gg_season(aus_production, Bricks)
```

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



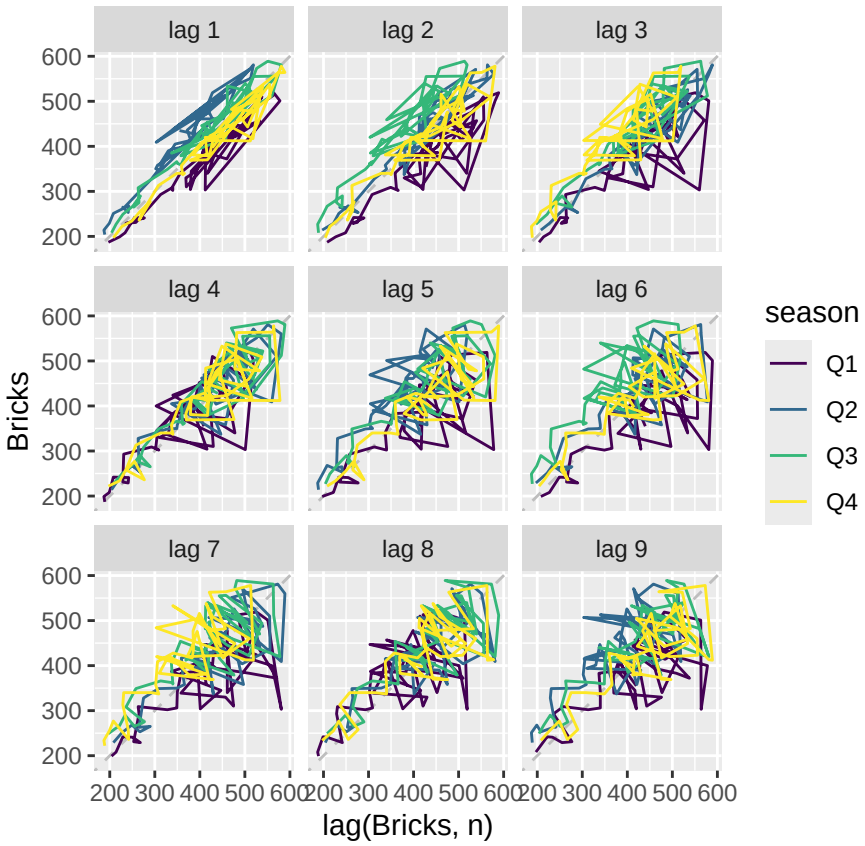

```
# Subseries plot for seasonal patterns
gg_subseries(aus_production, Bricks)
```

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

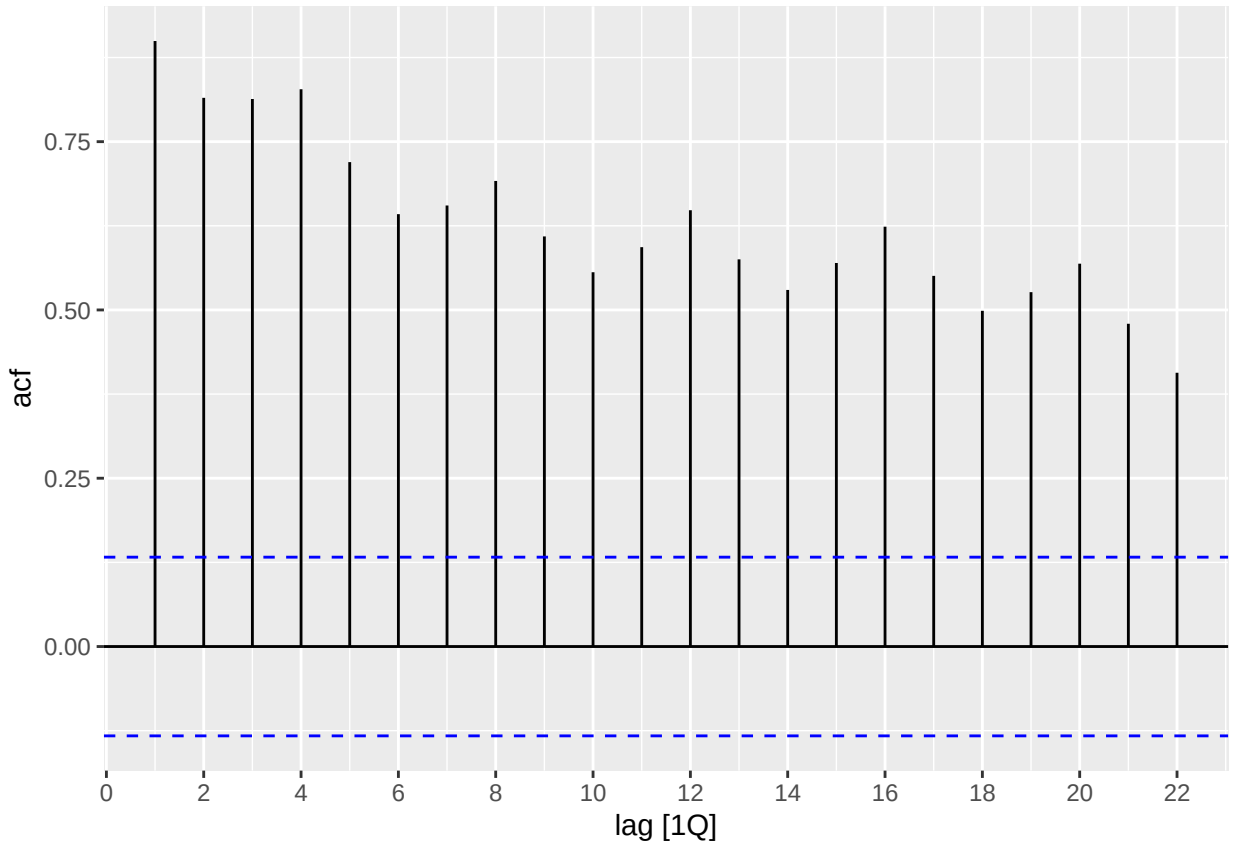



```
# Lag plot to explore autocorrelation
gg_lag(aus_production, Bricks)
```

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




```
# Autocorrelation function
ACF(aus_production, Bricks) %>% autoplot()
```



Seasonality, Cyclicity, and Trend:

- *Trend*: There is a clear upward trend in brick production from the 1960s to the 1980s, followed by a steady decline toward the 2000s.
- *Seasonality*: There is evidence of moderate seasonality, with regular fluctuations in brick production, especially in the subseries plots, showing consistent quarterly variations.
- *Cyclicality*: The series displays some cyclicality, particularly with the long-term rise and fall of brick production over multiple decades.

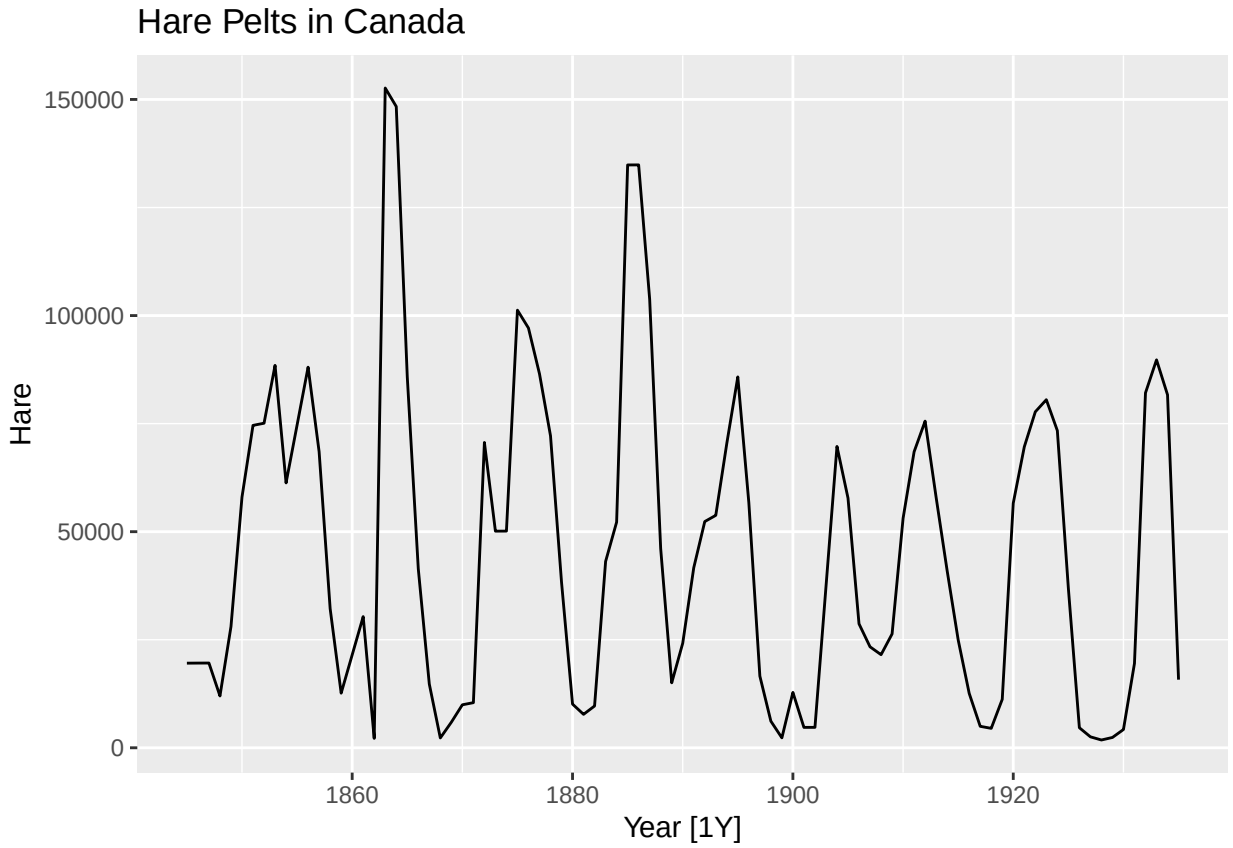
What do you learn about the series?: Brick production in Australia increased significantly from the 1960s to the early 1980s, likely reflecting construction booms during that time. After the 1980s, production started to decline, suggesting a downturn in demand or economic activity.

Seasonal Patterns: The subseries plots and lag plots show moderate seasonality, with slightly higher production in certain quarters. However, the seasonality is not as pronounced as in some other datasets.

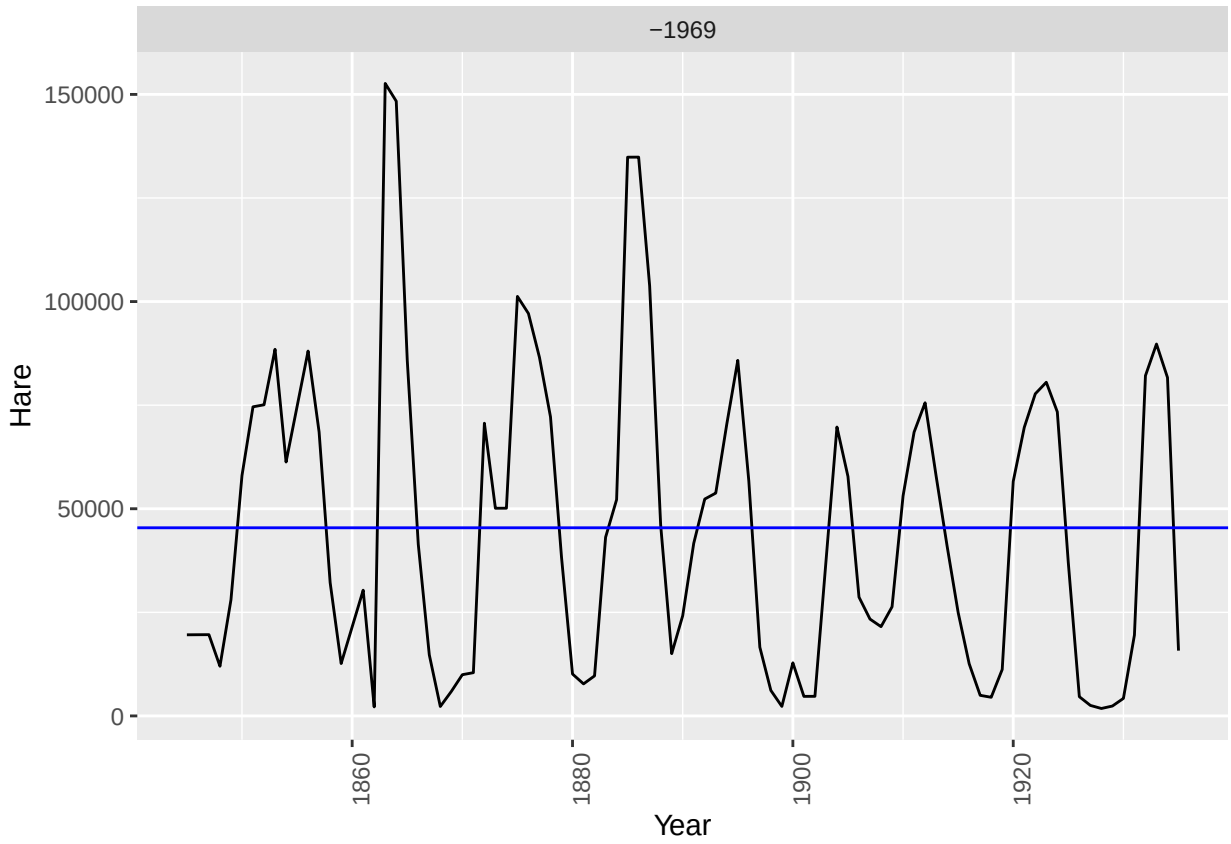
Unusual Years: The peak in production in the late 1970s and early 1980s stands out as an unusual period, followed by a significant decline afterward. This could indicate a construction boom followed by a market slowdown.

Hare Pelts (pelt)

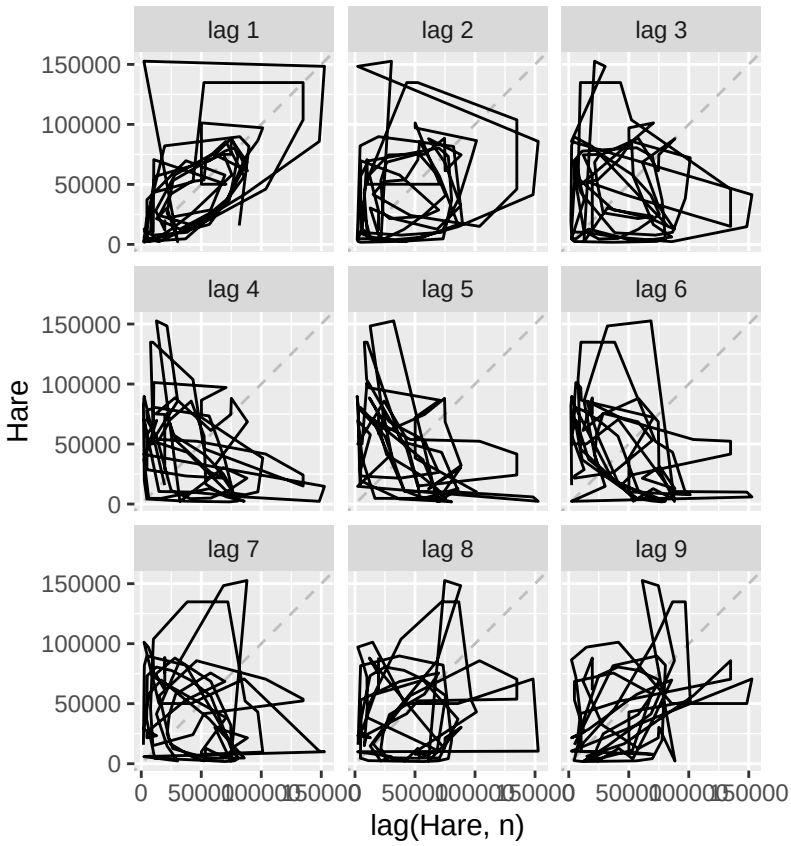
```
# Plot the Hare series
autoplot(pelt, Hare) +
  labs(title = "Hare Pelts in Canada")
```



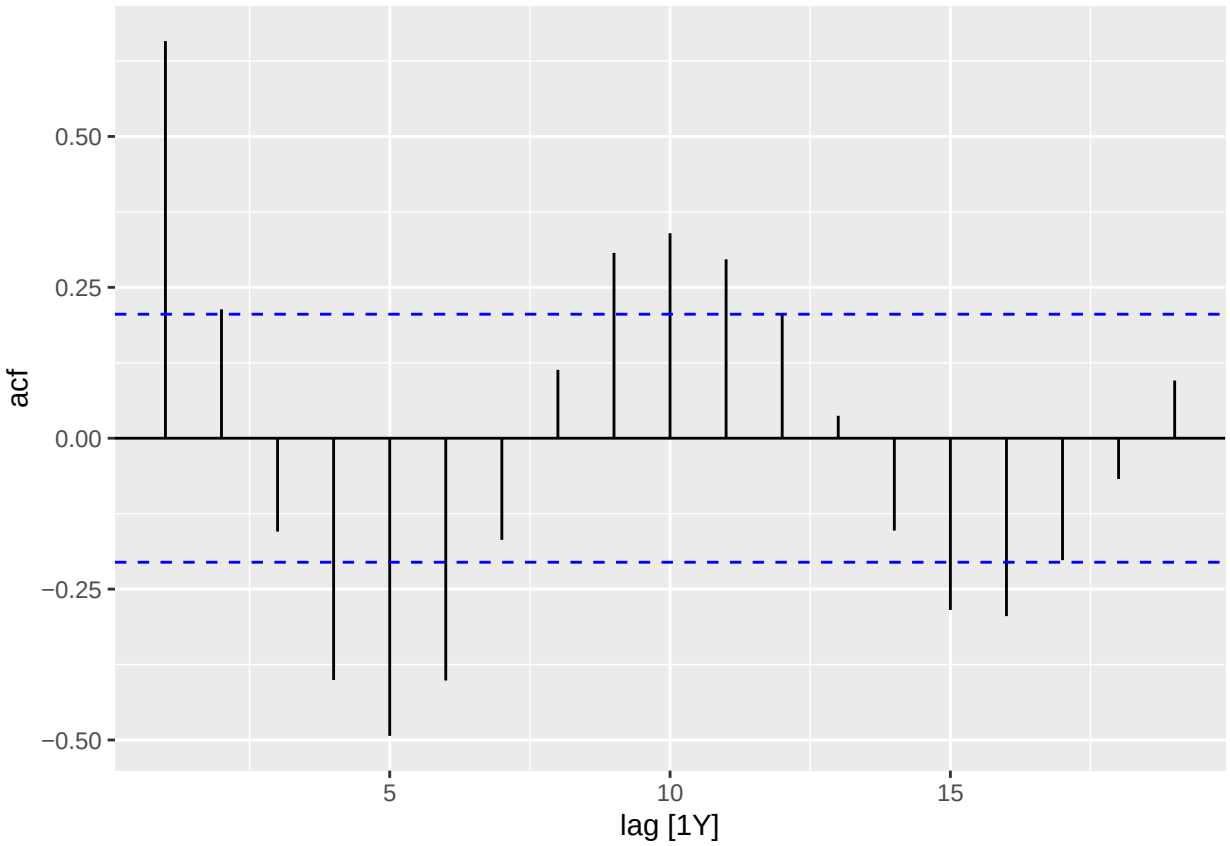

```
# Seasonality
#gg_season(pelt, Hare)

# Subseries plot for seasonal patterns
gg_subseries(pelt, Hare)
```




```
# Lag plot
gg_lag(pelt, Hare)
```




```
# Autocorrelation function
ACF(pelt, Hare) %>% autoplot()
```



Seasonality, Cyclicity, and Trend:

- *Trend*: The series shows a cyclical pattern with no clear long-term trend. There are significant rises and falls in the number of hare pelts.
- *Seasonality*: No strong seasonality is observed in the time series. The lag plots and autocorrelation function (ACF) suggest cyclical patterns but not regular seasonal spikes.
- *Cyclicality*: There are prominent cyclic patterns with large peaks and troughs recurring roughly every 10-15 years, indicating cycles in the hare population, likely influenced by ecological factors (e.g., predator-prey dynamics).

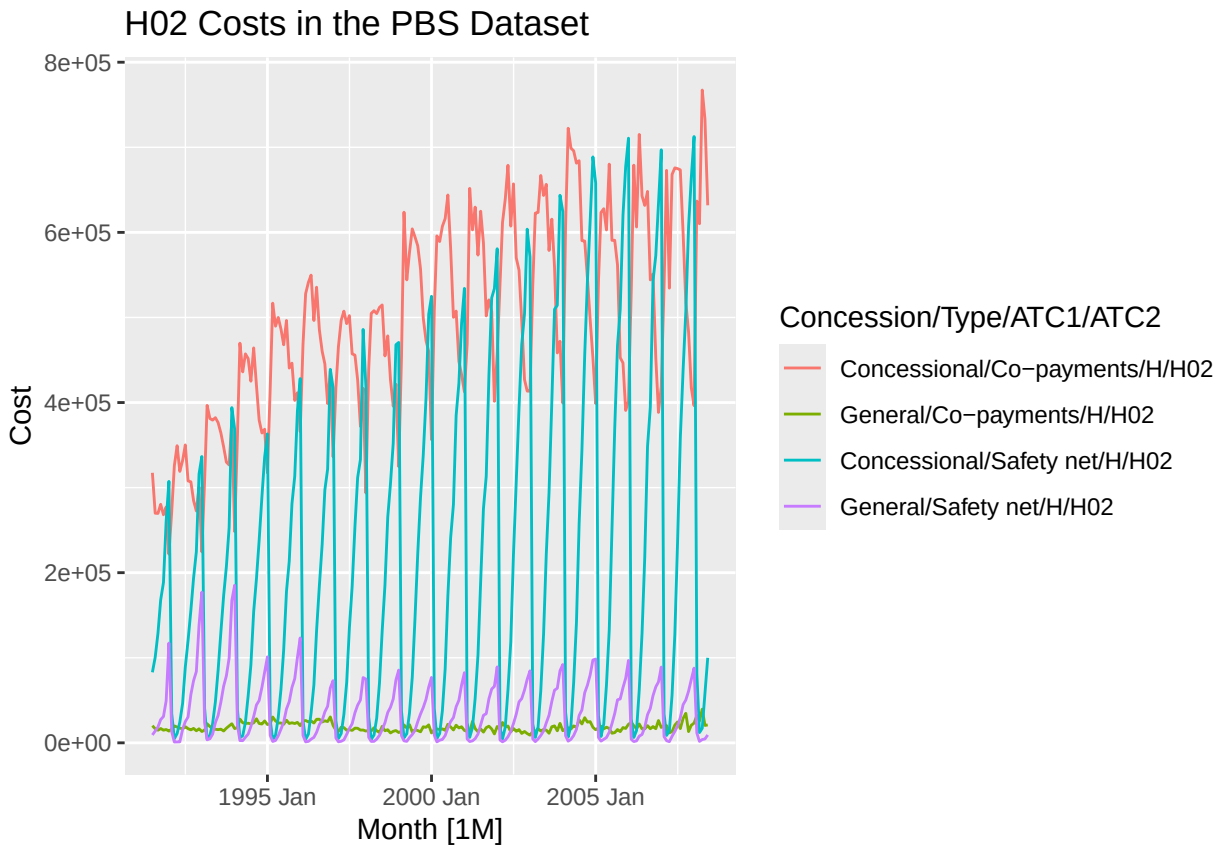
What do you learn about the series?: The hare pelt production follows a regular cyclical pattern, with sharp increases and decreases over the years. This could be due to natural population cycles driven by ecological factors.

Seasonal Patterns: The dataset does not exhibit strong seasonal behavior. The variation appears to be driven more by longer-term cyclical factors rather than predictable seasonal effects.

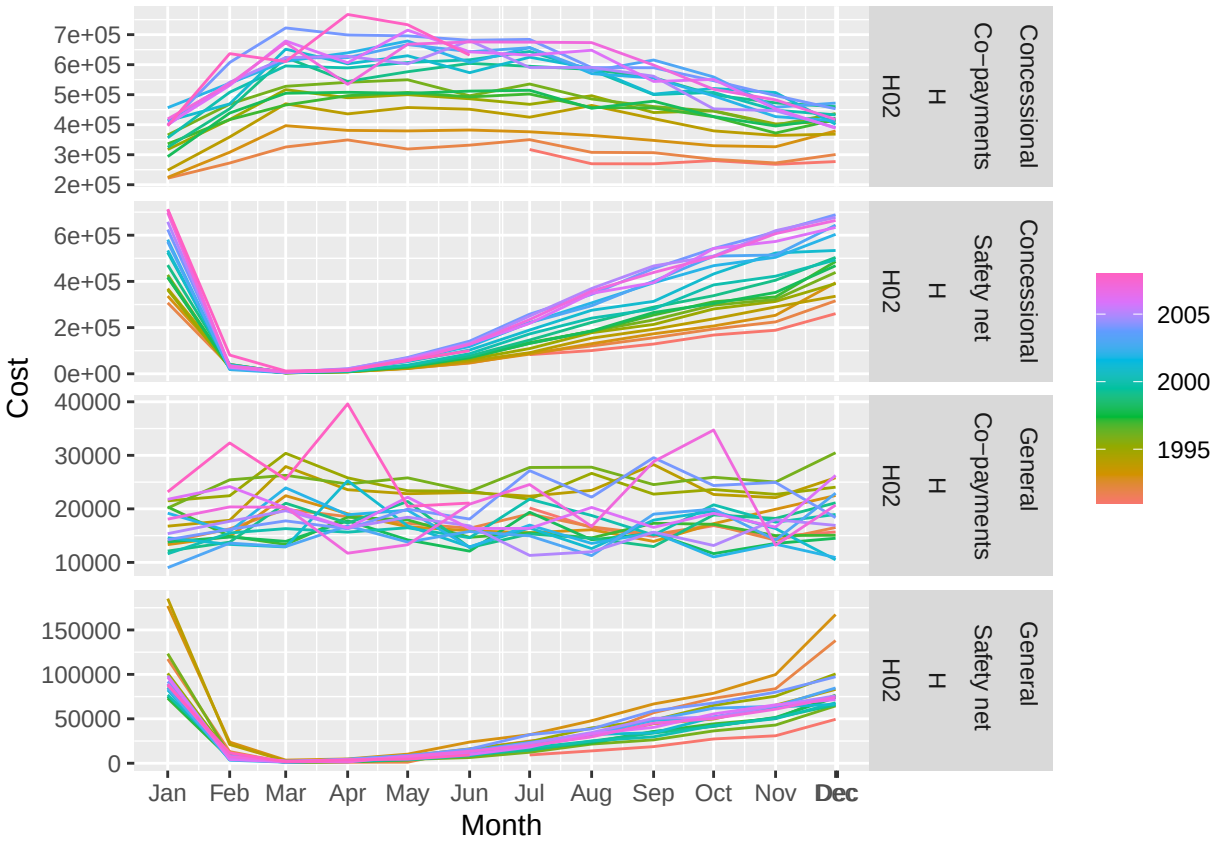
Unusual Years: Peaks around the late 1800s and early 1900s stand out as periods of unusually high hare pelt production, followed by sharp declines. These could be linked to specific environmental or economic factors affecting the hare population.

H02 Cost (PBS)

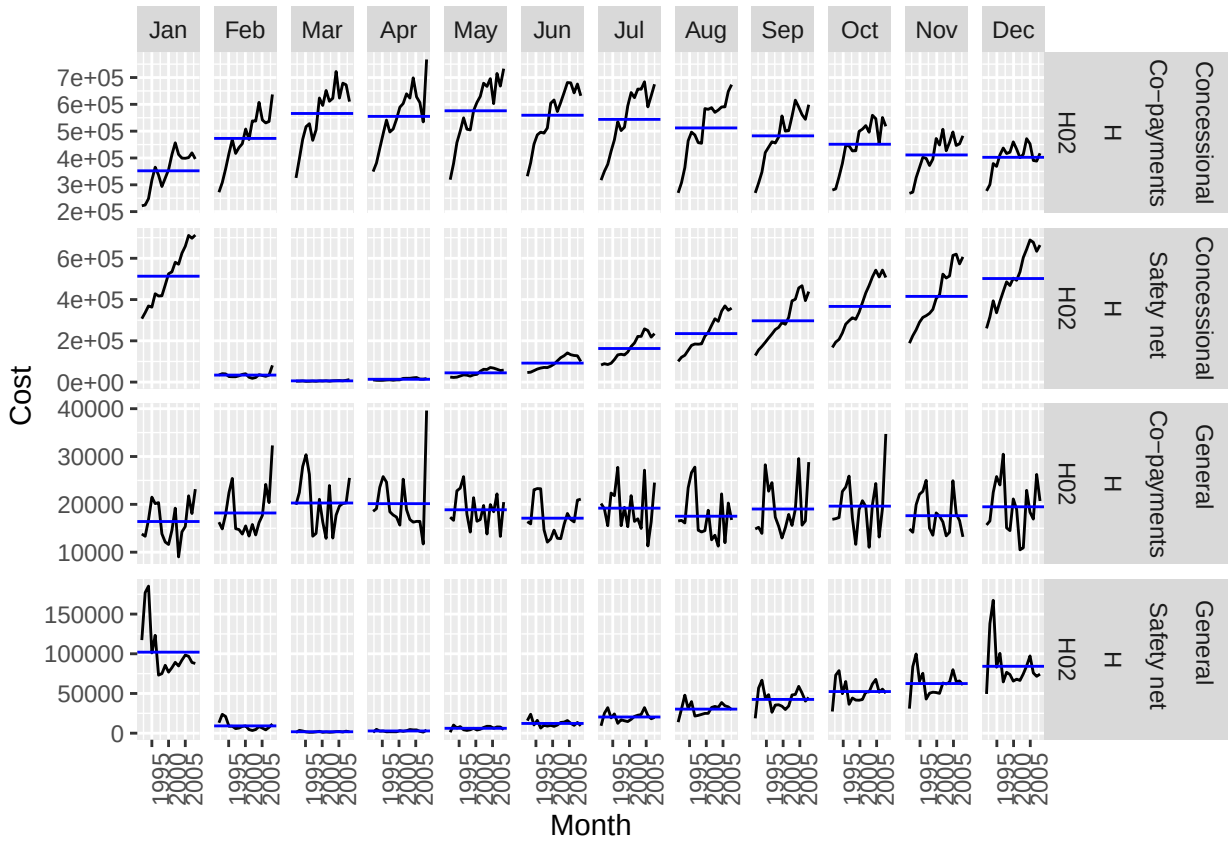
```r
# Plot the H02 Cost series
autoplot(h02, Cost) +
  labs(title = "H02 Costs in the PBS Dataset")
```



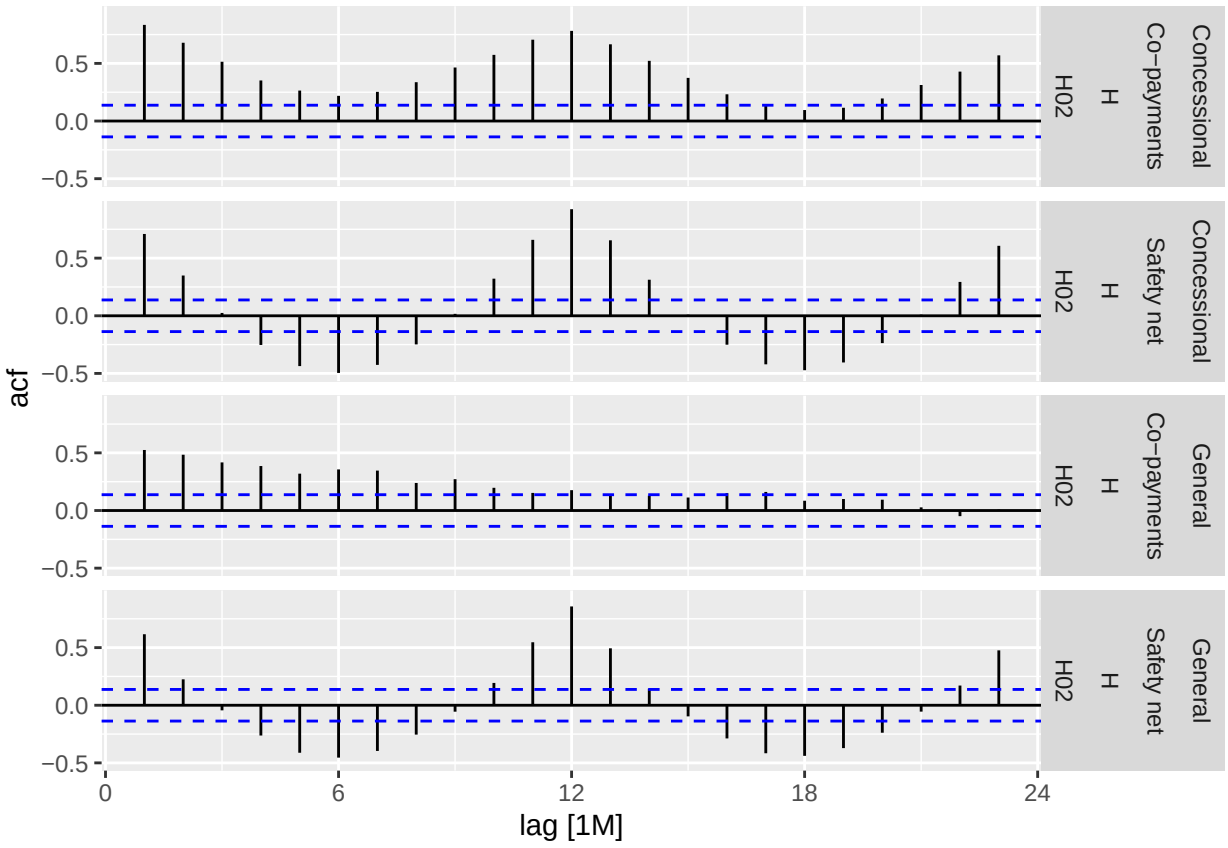

```r
# Seasonality
gg_season(h02, Cost)
```




```
# Subseries plot for seasonal patterns
gg_subseries(h02, Cost)
```

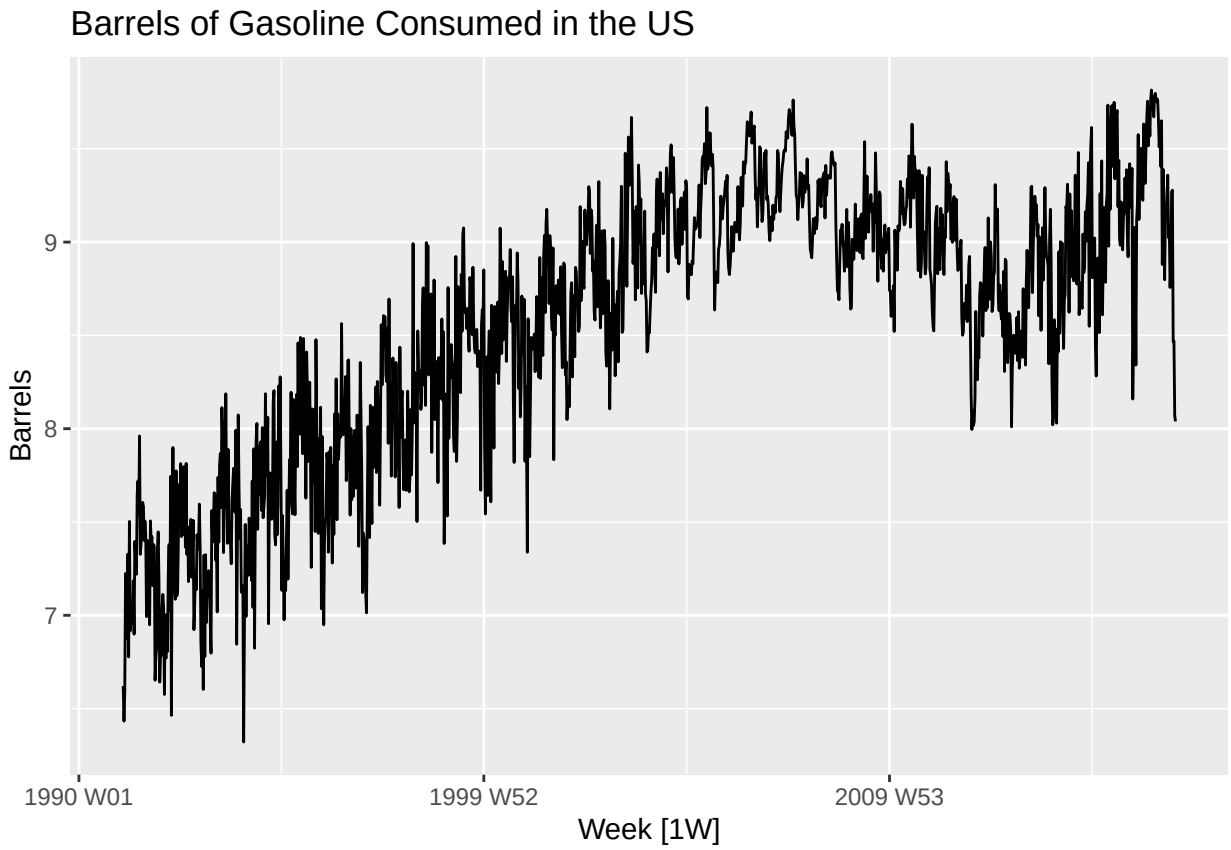



```
# Lag plot
#gg_lag(h02, Cost)

# Autocorrelation function
ACF(h02, Cost) %>% autoplot()
```

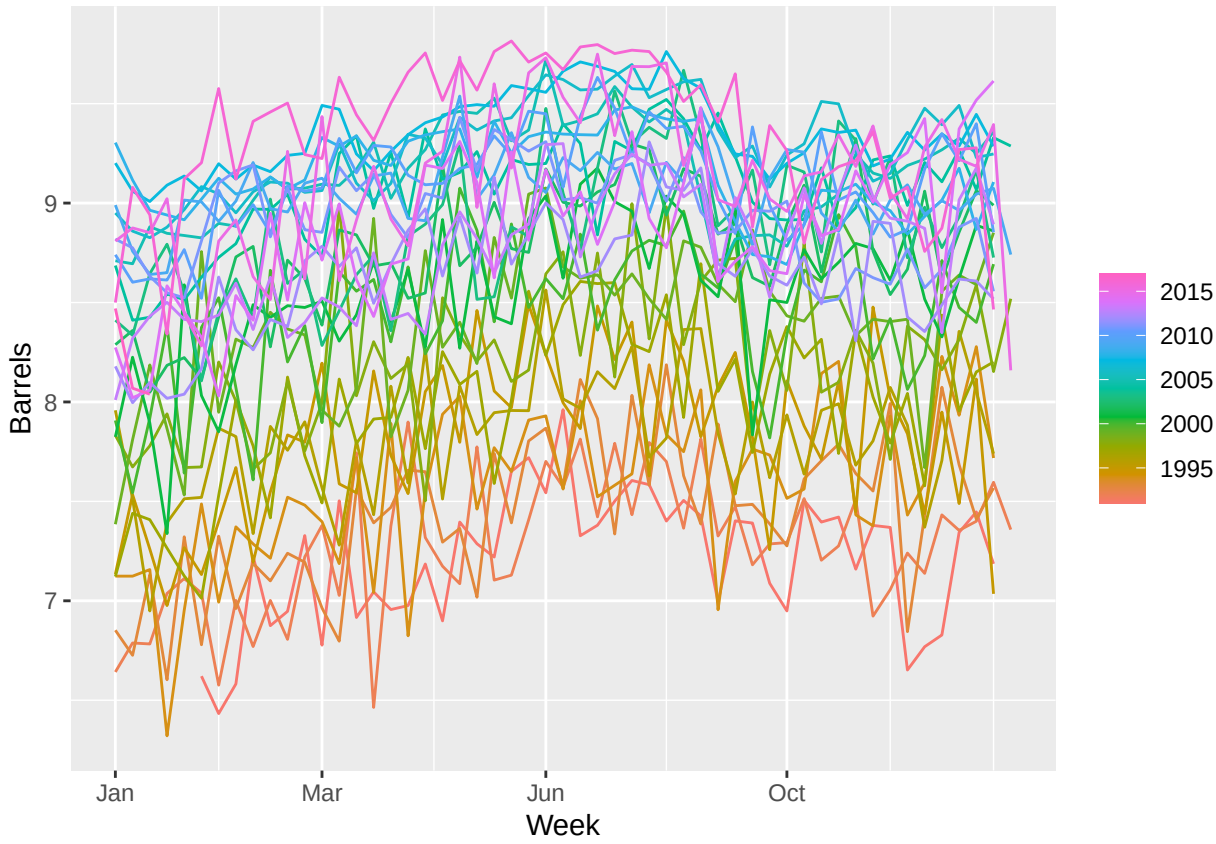


Barrels (us_gasoline)

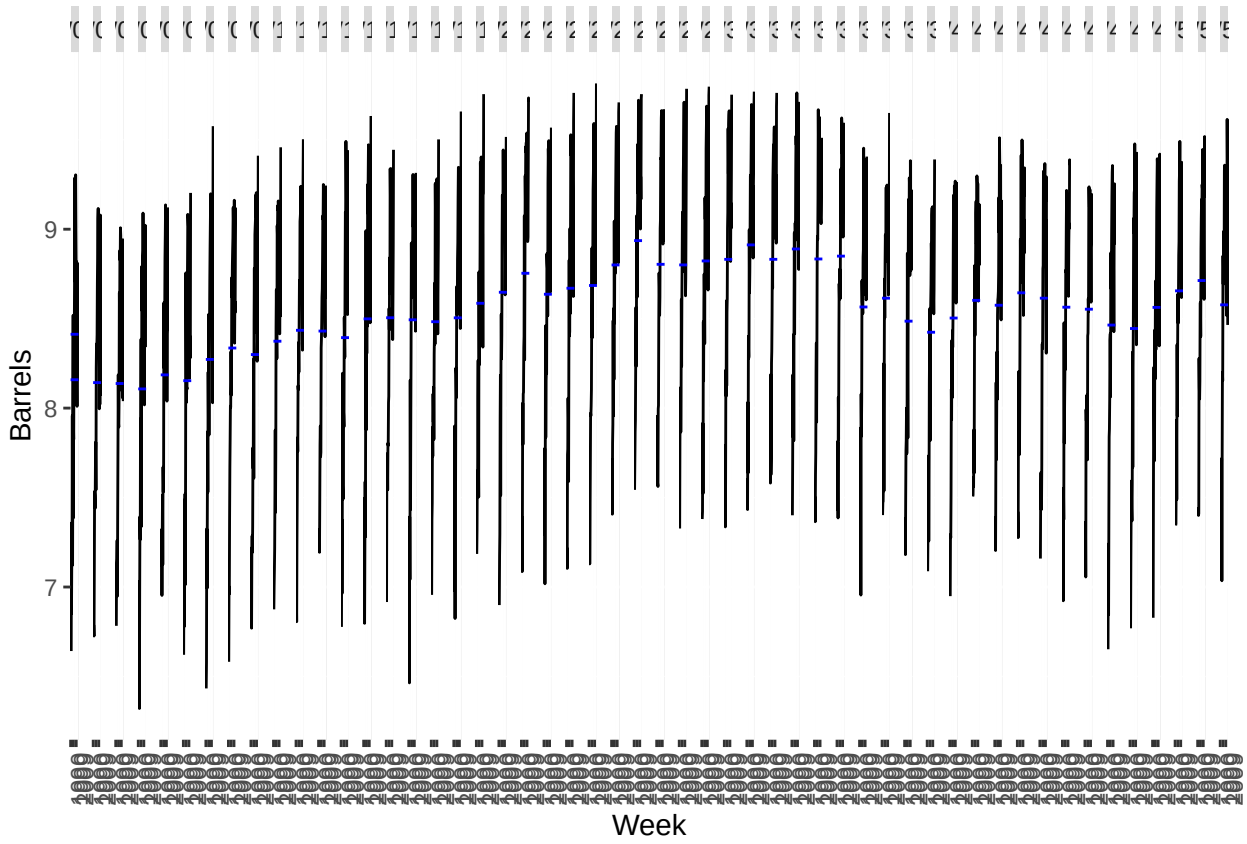
```
# Plot the Barrels series
autoplot(barrels, Barrels) +
  labs(title = "Barrels of Gasoline Consumed in the US")
```



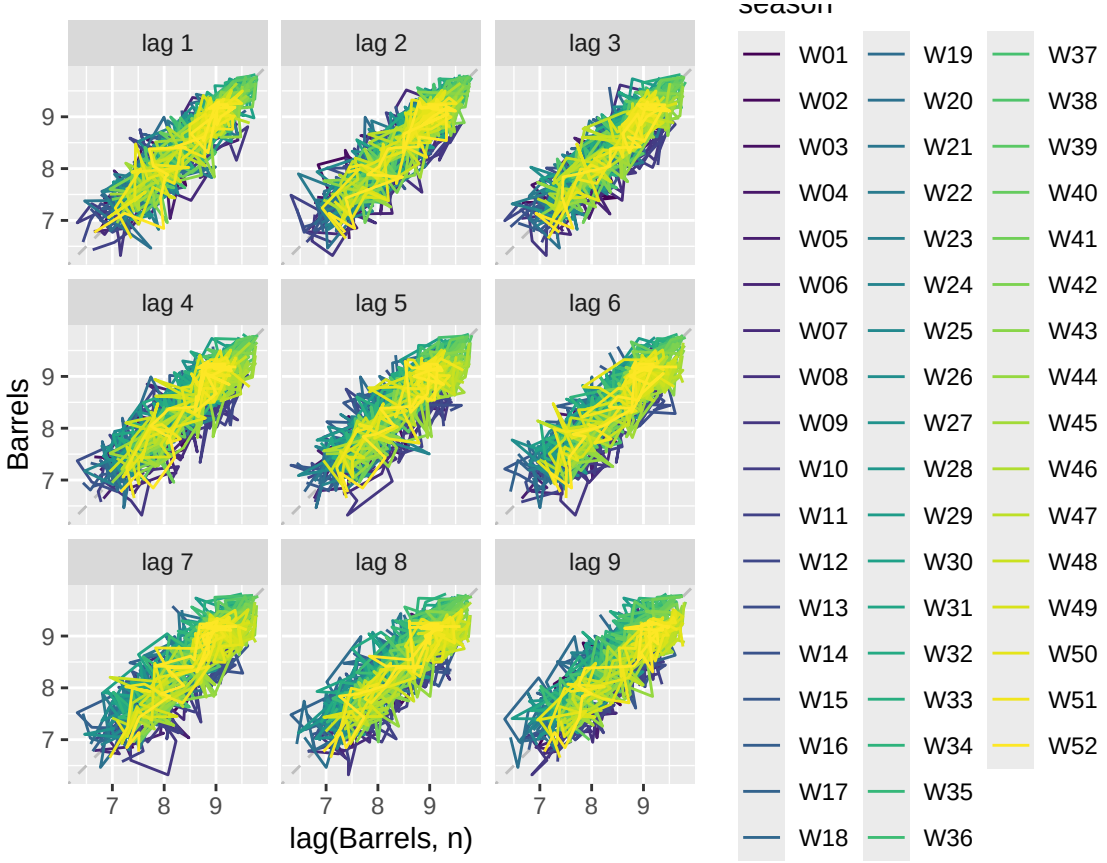

```
# Seasonality
gg_season(barrels, Barrels)
```



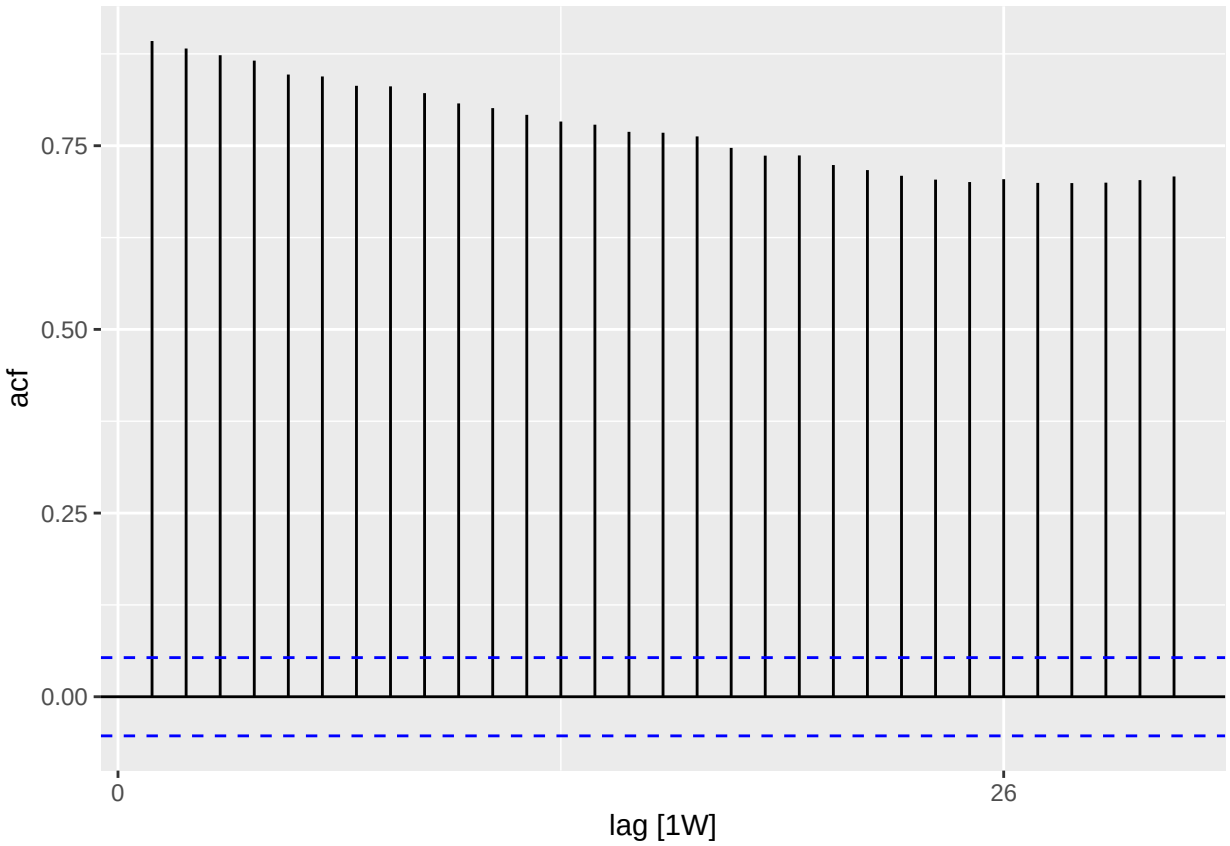

```
# Subseries plot for seasonal patterns
gg_subseries(barrels, Barrels)
```




```
# Lag plot
gg_lag(barrels, Barrels)
```




```
# Autocorrelation function
ACF(barrels, Barrels) %>% autoplot()
```


0.75
0.50
0.25
0.00
acf
0
26
lag [1W]