Time Series Analysis with R Part 1, Lecture 5

Michèle Fille

Table of contents

Lecture 5 – Time Series Fundamentals

5.1 Introduction

Exercise 5.1. Introduction

Exercise 5.1: Predictor Variables

Read the following case. List the possible predictor variables that might be useful, assuming that the relevant data are available.

Case: Car Fleet Company

A large car fleet company asks us to help them forecast vehicle resale values. They purchase new vehicles, lease them out for three years, and then sell them. Better forecasts of vehicle sales values would mean better control of profits; understanding what affects resale values may allow leasing and sales policies to be developed in order to maximize profits. The resale values are being forecast by a group of specialists. Unfortunately, they see any statistical model as a threat to their jobs, and are uncooperative in providing information. Nevertheless, the company provides a large amount of data on previous vehicles and their eventual resale values.

- *Model or make of the car*
- *Odometer reading (how many km)*
- *Conditions of the cars*
- *Leaser - Company the car was leased to*
- *Colour of car*
- *Date of sale/purchase*

Self-Study 5.1. Introduction

Self-Study 5.1: Steps of Forecasting

For the case described in Exercise 5.1, describe the five steps of forecasting in the context of this project.

1. Business Understanding and Problem Definition

- *The main stakeholders should be defined.*
- *Everyone has been questioned about which way he or she can benefit from the new system.*

In case of the fleet company probably the group of specialists was not recognized as stakeholders which led to complications in gathering relevant information and later in finding an appropriate statistical approach and deployment of the new forecasting method.

2. Data Understanding: Information Gathering and Exploratory Analysis.

- *Data set of past sales should be obtained, including surrounding information such as the way data were gathered, possible outliers and incorrect records, special values in the data.*

- *Expertise knowledge should be obtained from people responsible for the sales such as seasonal price fluctuations, if there is dependency of the price on the situation in economy, also finding other possible factors which can influence the price.*
- *Graphs which show dependency of the sale price on different predictor variables should be considered.*
- *Dependency of the sale price on month of the year should be plot.*

3. Data Preparation

- *Possible outliers and inconsistent information should be found (for example very small, zero or even negative prices).*

4. Choosing and fitting models

- *A model to start from (for example a linear model) and predictor variables which most likely affect the forecasts should be chosen.*

5. Evaluating a forecasting model

- *Predicting performance of the models must be evaluated.*
- *The model should be changed (for example by transforming parameters, adding or removing predictor variables) and it's performance evaluated.*

This should be done iteratively wit step 4 a few times until a satisfactory model is found.

6. Deploying a Forecasting Model

- *The appropriate software should be deployed to the company and relevant people should be educated how to use this software.*
- *Forecasting accuracy should be checked against new sales. If necessary the model should be updated and then the deployed software.*

5.2 Time Series in R

Exercise 5.2. Time Series in R

Exercise 5.2: Exploring tsibble Objects

1. Explore the following 3 time series. (The data sets are all contained in the package tsibbledata.)

- 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.

```
#install.packages("tsibbledata")

library(tsibbledata)
```

```
Warning: Paket 'tsibbledata' wurde unter R Version 4.3.3 erstellt
```

```
library(tsibble)
```

```
Attache Paket: 'tsibble'
```

```
Die folgenden Objekte sind maskiert von 'package:base':

    intersect, setdiff, union
```

```
library(ggplot2)
library(dplyr)
```

```
Attache Paket: 'dplyr'
```

```
Die folgenden Objekte sind maskiert von 'package:stats':

    filter, lag
```

```
Die folgenden Objekte sind maskiert von 'package:base':

    intersect, setdiff, setequal, union
```

```
library(fpp3) # needed for autoplot for tsibble
```

```
-- Attaching packages --------------------------------------------- fpp3 0.5 --
```

```
v tibble     3.2.1     v feasts     0.3.1
v tidyr      1.3.1     v fable      0.3.3
v lubridate  1.9.3     v fabletools 0.4.1

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

- Bricks from aus_production

```
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
```

```
help("aus_production")
```

```
starte den http Server für die Hilfe fertig
```

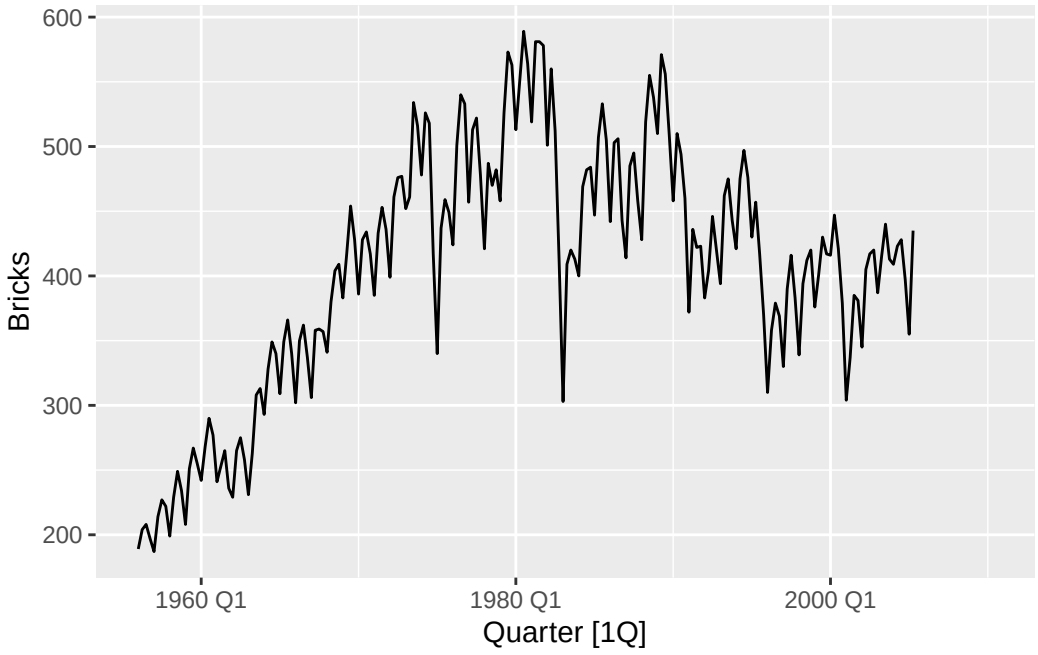
```
interval(aus_production)
```

```
Warning: tz(): Don't know how to compute timezone for object of class
tbl_ts/tbl_df/tbl/data.frame; returning "UTC".
```

```
<Interval[0]>
```

```
# The time interval is quarterly BUT help says half-hourly (?)

aus_production |> autoplot(Bricks)
```

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




```
# An upward trend is apparent until 1980, after which the number of bricks being produced
```

- Lynx from pelt

```
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
```

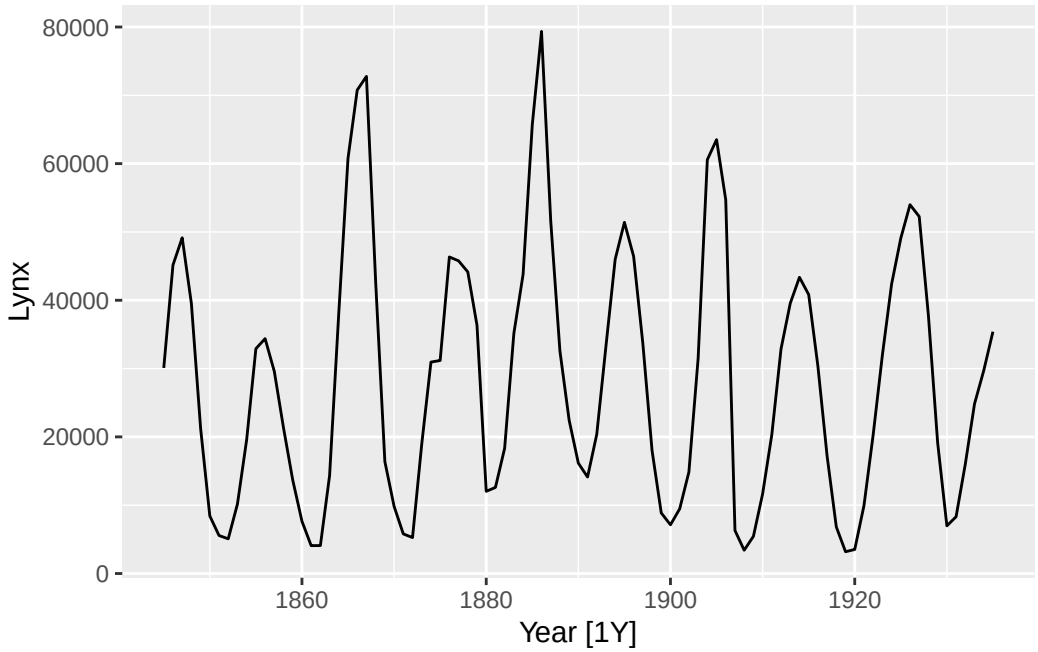
```
4  1848 11990 39520
5  1849 28040 21230
6  1850 58000  8420
```

```
help("pelt")

interval(pelt)
```

```
Warning: tz(): Don't know how to compute timezone for object of class
tbl_ts/tbl_df/tbl/data.frame; returning "UTC".
```

```
<Interval[0]>
```

```
# Interval is yearly. Looking closer at the data, we can see that the index is a Date vari

pelt |> autoplot(Lynx)
```




```
# Canadian lynx trappings are cyclic, as the extent of peak trappings
# is unpredictable, and the spacing between the peaks is irregular but approximately 10 ye
```

- Close from gafa_stock.

```
head(gafa_stock)
```

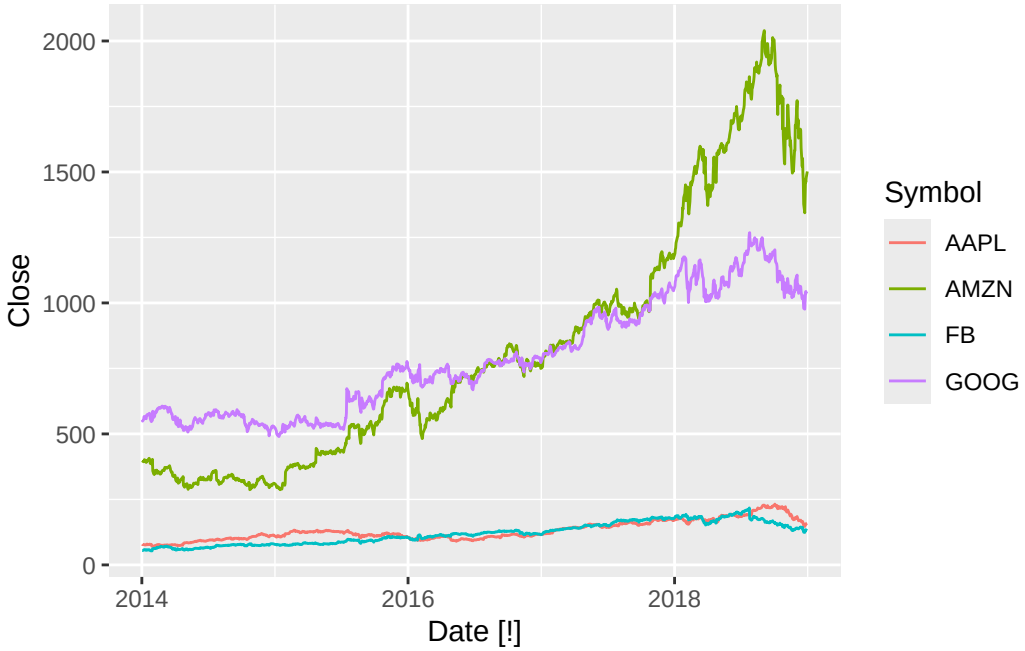
```
# A tsibble: 6 x 8 [!]
# Key:       Symbol [1]
  Symbol Date        Open  High   Low Close Adj_Close    Volume
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
1 AAPL   2014-01-02  79.4  79.6  78.9  79.0      67.0  58671200
2 AAPL   2014-01-03  79.0  79.1  77.2  77.3      65.5  98116900
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
```

```
help("gafa_stock")

interval(gafa_stock)
```

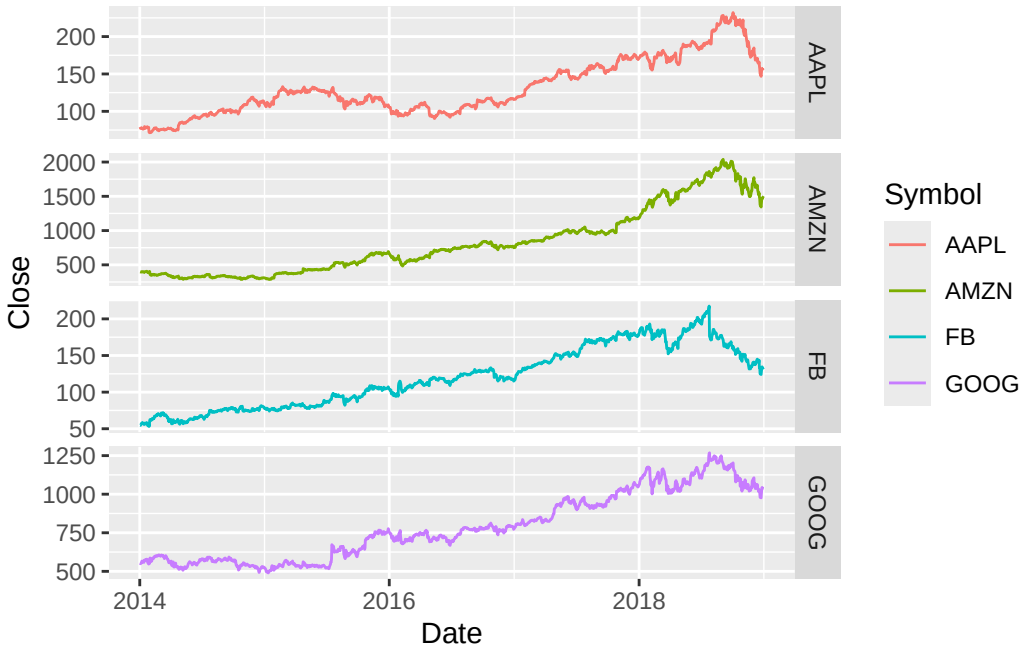
```
Warning: tz(): Don't know how to compute timezone for object of class
tbl_ts/tbl_df/tbl/data.frame; returning "UTC".
```

```
<Interval[0]>
```

```
# The time interval is daily (irregular)

gafa_stock |> autoplot(Close)
```




```
# Stock prices for these technology stocks have risen for most of the series, until mid-la
# The four stocks are on different scales, so they are not directly comparable. A plot wit

gafa_stock |>
  ggplot(aes(x=Date, y=Close, group=Symbol)) +
  geom_line(aes(col=Symbol)) +
  facet_grid(Symbol ~ ., scales='free')
```




```
# The downturn in the second half of 2018 is now very clear, with Facebook taking a big dr
```

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

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

```
# A tsibble: 4 x 3 [!]
# Key:       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.
```

Self-Study 5.2. Time Series in R

Self-Study 5.2: Creating tsibble Objects

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

- Install the USgas package.

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

```
Warning: Paket 'USgas' wurde unter R Version 4.3.3 erstellt
```

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

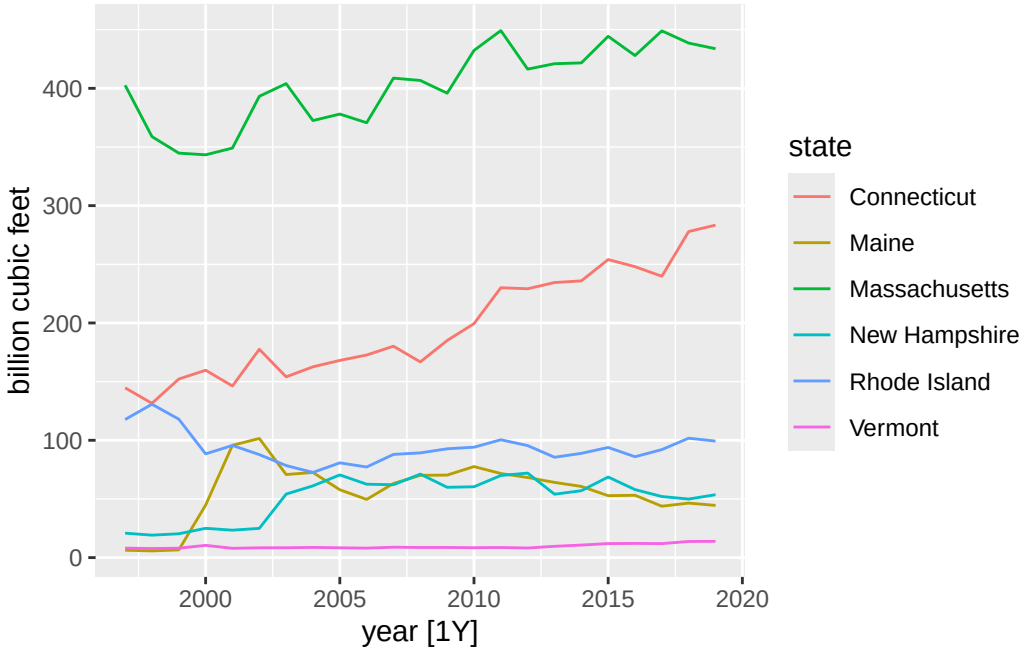
```
head(us_total)
```

```
  year   state      y
1 1997 Alabama 324158
2 1998 Alabama 329134
3 1999 Alabama 337270
4 2000 Alabama 353614
5 2001 Alabama 332693
6 2002 Alabama 379343
```

```
help(us_total) # y = yearly total natural gas consumption in a million cubic feet by state

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

- 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_tsibble %>%
  filter(state %in% c("Maine", "Vermont", "New Hampshire", "Massachusetts",
                      "Connecticut", "Rhode Island")) %>%
  autoplot(y/1e3) +
  labs(y = "billion cubic feet")
```



2. Download tourism.xlsx from GitHub and read it into R using readxl::read_excel().

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

```
library(readxl)
```

```
Warning: Paket 'readxl' wurde unter R Version 4.3.3 erstellt
```

```
my_tourism <- readxl::read_excel("tourism.xlsx") %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(
    index = Quarter,
    key = c(Region, State, Purpose, Trips)
  )
```

```
head(my_tourism)
```

```
# A tsibble: 6 x 5 [1Q]
# Key:       Region, State, Purpose, Trips [6]
  Quarter Region   State           Purpose  Trips
```

```
    <qtr> <chr>    <chr>           <chr>    <dbl>
1 2010 Q1 Adelaide South Australia Business  68.7
2 2005 Q2 Adelaide South Australia Business  73.3
3 2013 Q2 Adelaide South Australia Business 101.
4 2001 Q4 Adelaide South Australia Business 101.
5 2013 Q1 Adelaide South Australia Business 102.
6 2006 Q4 Adelaide South Australia Business 107.
```

```
head(tourism)
```

```
# A tsibble: 6 x 5 [1Q]
# Key:       Region, State, Purpose [1]
  Quarter Region   State           Purpose  Trips
    <qtr> <chr>    <chr>           <chr>    <dbl>
1 1998 Q1 Adelaide South Australia Business  135.
2 1998 Q2 Adelaide South Australia Business  110.
3 1998 Q3 Adelaide South Australia Business  166.
4 1998 Q4 Adelaide South Australia Business  127.
5 1999 Q1 Adelaide South Australia Business  137.
6 1999 Q2 Adelaide South Australia Business  200.
```

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

```
my_tourism %>%
  as_tibble() %>%
  summarise(Trips = mean(Trips), .by=c(Region, Purpose)) %>%
  filter(Trips == max(Trips))
```

```
# A tibble: 1 x 3
  Region Purpose  Trips
  <chr>  <chr>    <dbl>
1 Sydney Visiting  747.
```

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

???? who to combine Purpose and Region, when only the total trips by state

```r
state_tourism <- my_tourism %>%
  group_by(State) %>%
  summarise(Trips = sum(Trips)) %>%
  ungroup()

head(state_tourism)
```

```
# A tsibble: 6 x 3 [1Q]
# Key:       State [1]
  State Quarter 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.
```

5.3 Visualizing Time Series

Exercise 5.3. Visualizing Time Series

Exercise 5.3: Time Plots, Seasonal Plots and ACF Plots

1. The aus_arrivals data set comprises quarterly international arrivals to Australia from Japan, New Zealand, UK and the US.

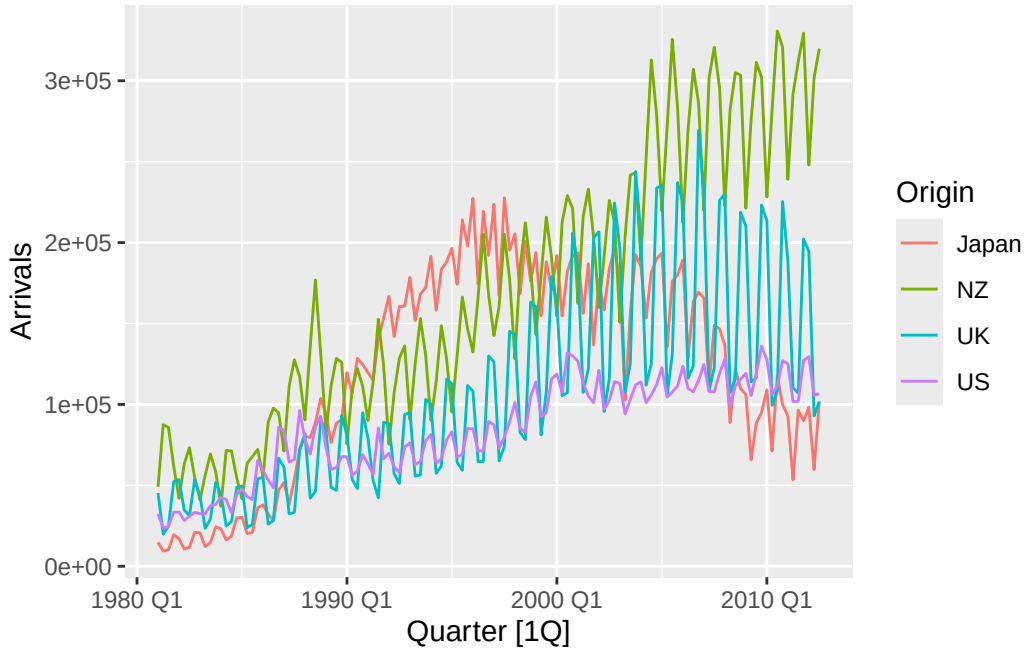
- Use autoplot(), gg_season() and gg_subseries() to compare the differences between the arrivals from these four countries.

```r
head(aus_arrivals)
```

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

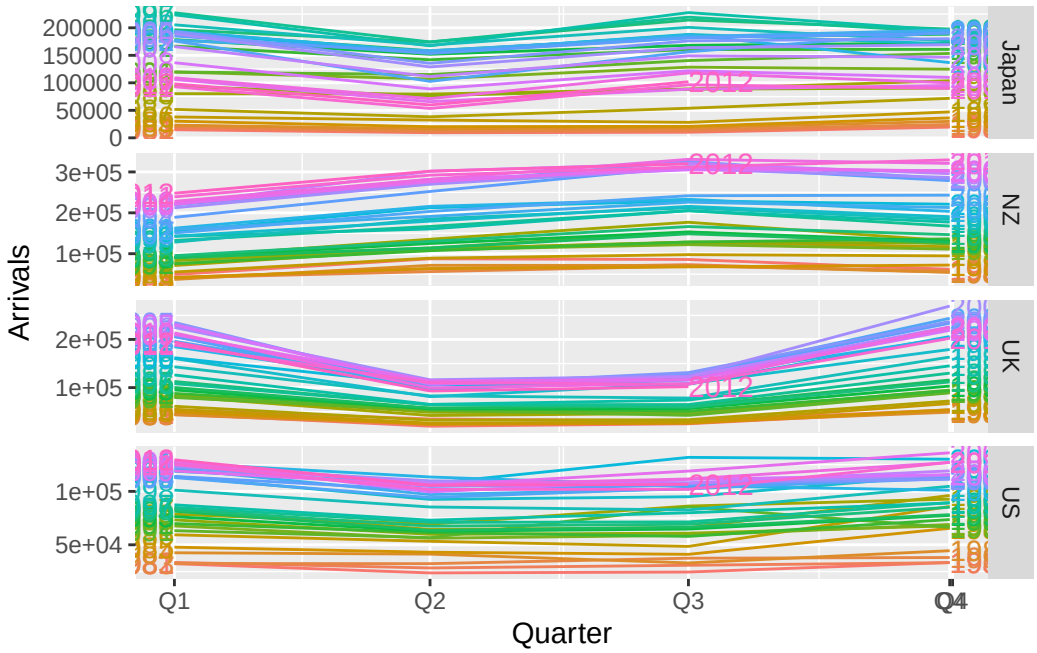
```
6 1982 Q2 Japan     10617
```

```
aus_arrivals %>%
  autoplot(Arrivals)
```



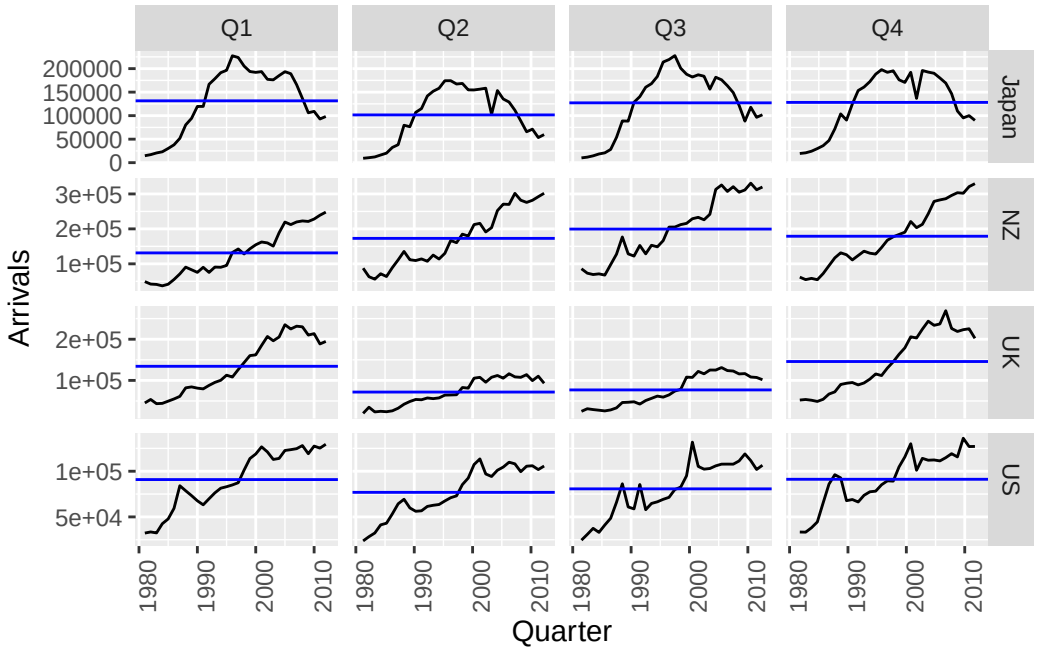

```
# Generally the number of arrivals to Australia is increasing over the entire series, with
```

```
aus_arrivals %>%
  gg_season(Arrivals, labels = "both")
```




```
# The seasonal pattern of arrivals appears to vary between each country. In particular, ar
```

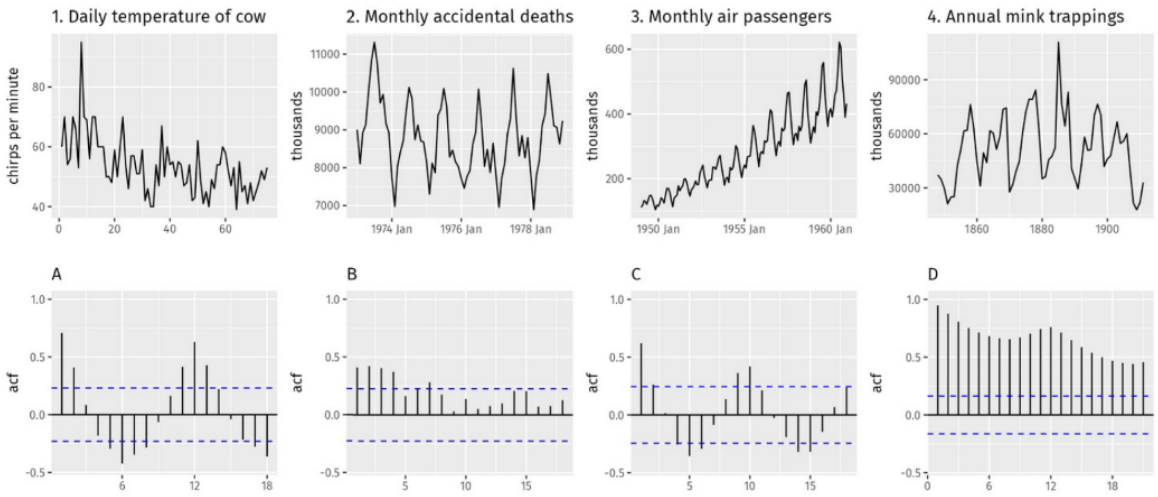
```
aus_arrivals %>%
  gg_subseries(Arrivals)
```




```
# The subseries plot reveals more interesting features. It is evident that whilst the UK a
```

- Can you identify any unusual observations?
  - *2000 Q3: Spikes from the US (Sydney Olympics arrivals)*
  - *2001 Q3-Q4 are unusual for US (9/11 effect)*
  - *1991 Q3 is unusual for the US (Gulf war effect?)*

2. The following time plots and ACF plots correspond to four different time series. Your task is to match each time plot in the first row with one of the ACF plots in the second row.



1-B, 2-A, 3-D, 4-C

Self-Study 5.3. Visualizing Time Series

Self-Study 5.3: Time Plots and White Noise

1. Download the file tute1.csv from GitHub, open it in Excel (or some other spreadsheet application), and review its contents. You will 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.

- Read the data into R and store it in the variable tute1.

```
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.
```

```
head(tute1)
```

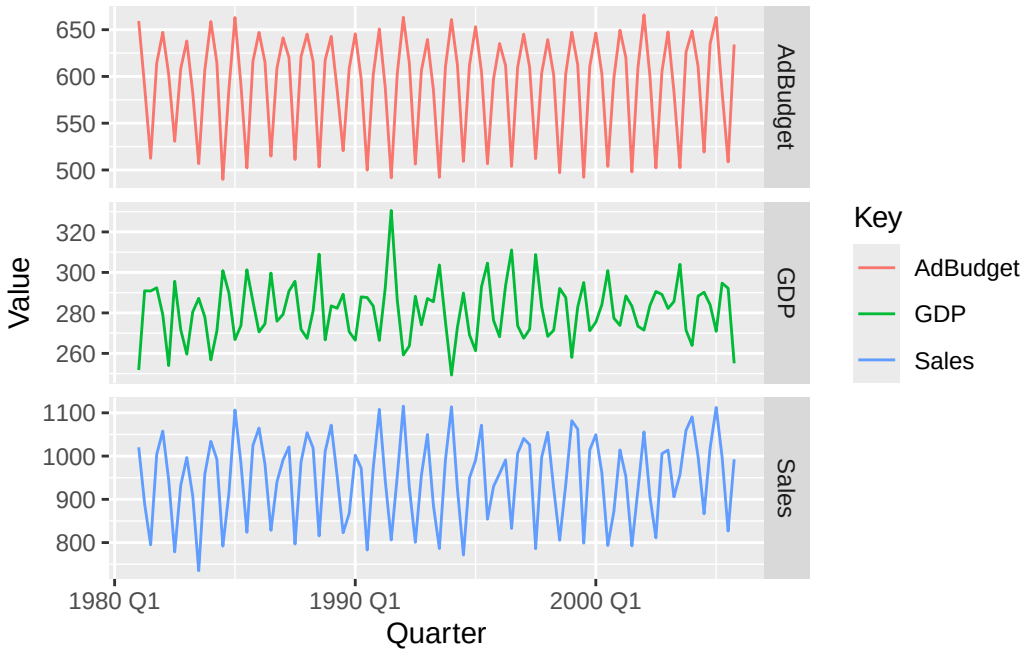
```
# A tibble: 6 x 4
  Quarter    Sales AdBudget   GDP
  <date>     <dbl>    <dbl> <dbl>
1 1981-03-01 1020.     659.  252.
2 1981-06-01  889.     589   291.
3 1981-09-01  795      512.  291.
4 1981-12-01 1004.     614.  292.
5 1982-03-01 1058.     647.  279.
6 1982-06-01  944.     602   254
```

- View the data set and convert it to time series.

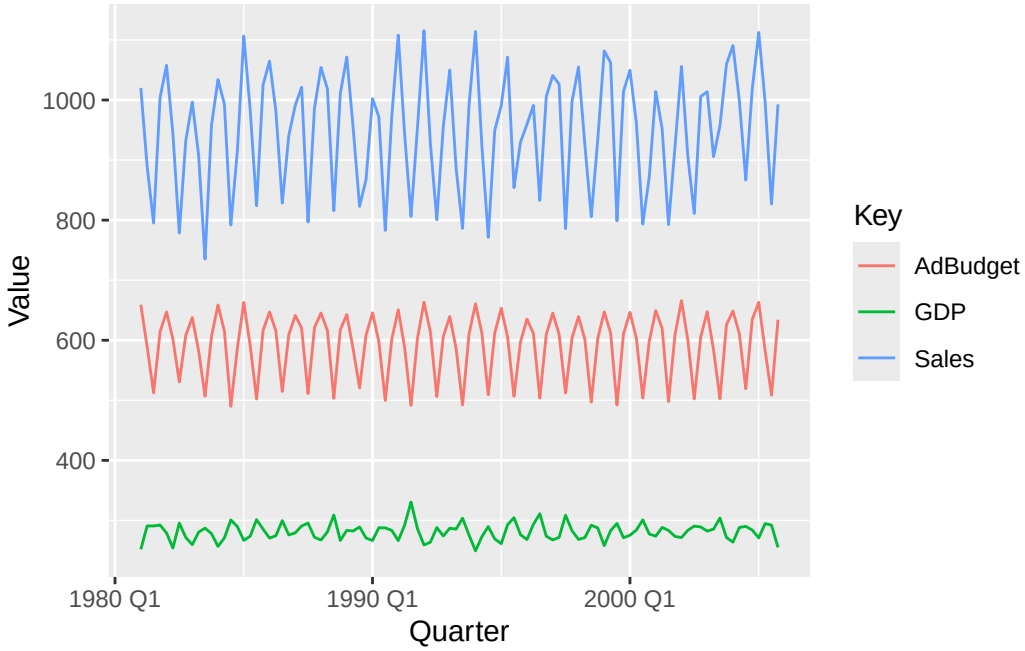
```
View(tute1)

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

- Construct time plots of each of the three series.
  - Hint: Consider using pivot_longer() and facet_grid().

```
mytimeseries %>%
  pivot_longer(-Quarter, names_to="Key", values_to="Value") %>%
  ggplot(aes(x = Quarter, y = Value, colour = Key)) +
  geom_line() +
  facet_grid(vars(Key), scales = "free_y")
```




```
# Without faceting:
mytimeseries %>%
  pivot_longer(-Quarter, names_to="Key", values_to="Value") %>%
  ggplot(aes(x = Quarter, y = Value, colour = Key)) +
  geom_line()
```



2. The aus_livestock data contains the monthly total number of pigs slaughtered in Victoria, Australia, from Jul 1972 to Dec 2018.

```
head(aus_livestock)
```

```
# A tsibble: 6 x 4 [1M]
# Key:       Animal, State [1]
     Month Animal                    State                           Count
     <mth> <fct>                     <fct>                           <dbl>
1 1976 Jul Bulls, bullocks and steers Australian Capital Territory  2300
2 1976 Aug Bulls, bullocks and steers Australian Capital Territory  2100
3 1976 Sep Bulls, bullocks and steers Australian Capital Territory  2100
4 1976 Okt Bulls, bullocks and steers Australian Capital Territory  1900
5 1976 Nov Bulls, bullocks and steers Australian Capital Territory  2100
6 1976 Dez Bulls, bullocks and steers Australian Capital Territory  1800
```

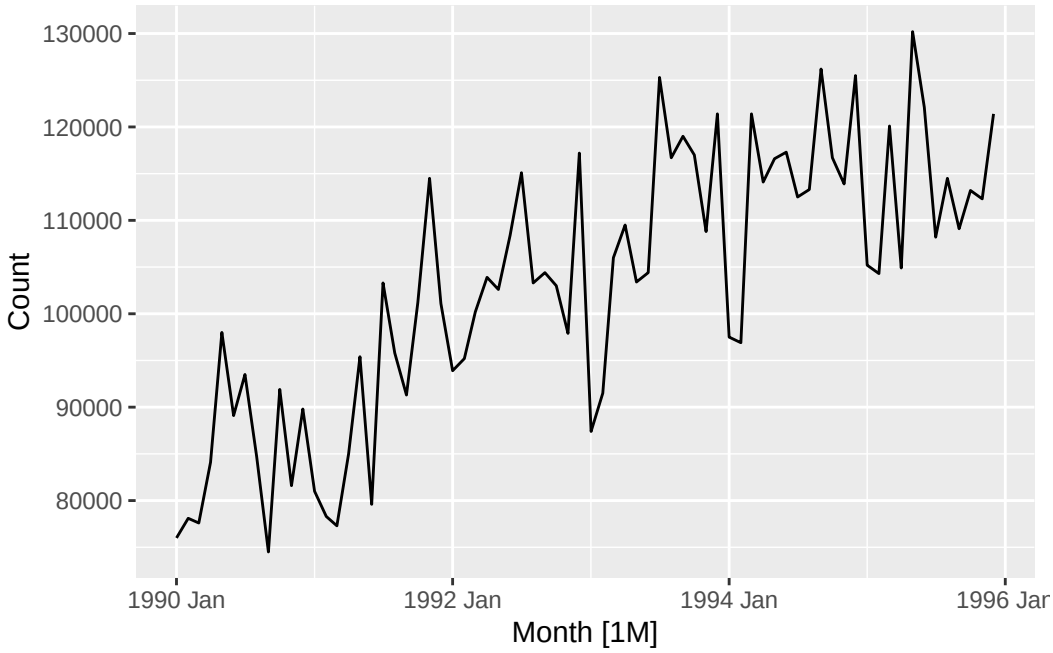
- Use filter() to extract pig slaughters in Victoria between 1990 and 1995.
  - Hint: Use the dplyr function between() to access the years between 1990 and 1995.
  - Hint: The column Month is a time object of class yearmonth. You can access the year in a yearmonth object x by year(x).

```r
vic_pigs <- aus_livestock %>%
  filter(Animal == "Pigs",
         State == "Victoria",
         between(year(Month), 1990, 1995))
head(vic_pigs)
```

```
# A tsibble: 6 x 4 [1M]
# Key:       Animal, State [1]
     Month Animal State    Count
     <mth> <fct>  <fct>    <dbl>
1 1990 Jan Pigs   Victoria 76000
2 1990 Feb Pigs   Victoria 78100
3 1990 Mär Pigs   Victoria 77600
4 1990 Apr Pigs   Victoria 84100
5 1990 Mai Pigs   Victoria 98000
6 1990 Jun Pigs   Victoria 89100
```

- Use autoplot() and ACF() for this data.

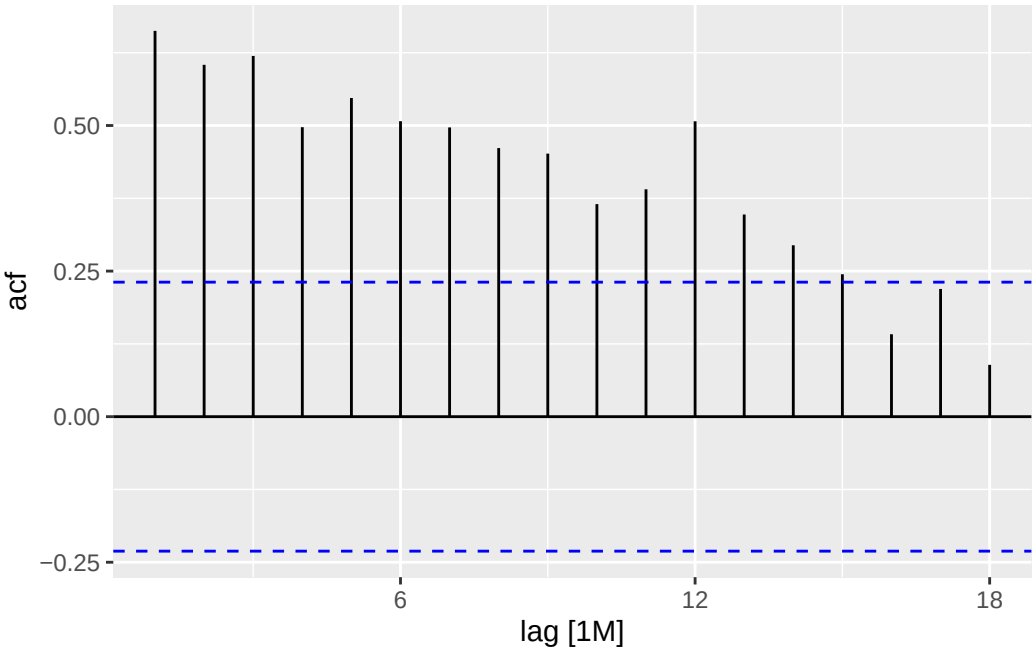
```r
vic_pigs %>%
  autoplot(Count)
```



```
# Although the values appear to vary erratically between months, a general upward trend is
```

- How do they differ from white noise?

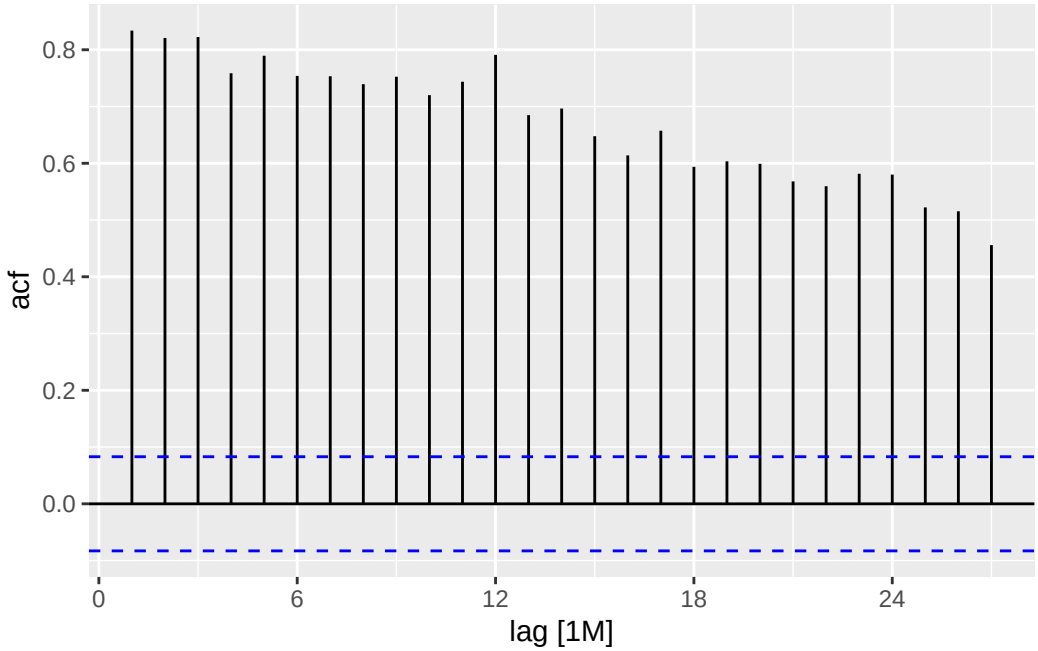
```
vic_pigs %>%
  ACF(Count) %>%
  autoplot()
```




```
# The first 14 lags are significant, as the ACF slowly decays. This suggests that the data
```

- If a longer period of data is used, what difference does it make to the ACF?

```
aus_livestock |>
  filter(Animal == "Pigs", State == "Victoria") |>
  ACF(Count) |>
  autoplot()
```




```
# The longer series has much larger autocorrelations, plus clear evidence of seasonality a
```