MONASH University

MONASH BUSINESS SCHOOL

ETC3550 Applied forecasting for business and economics

Ch2. Time series graphics

OTexts.org/fpp3/

Outline

Outline

Class packages

```r
# Data manipulation and plotting functions
library(tidyverse)
# Time series manipulation
library(tsibble)
# Forecasting functions
library(fable)
# Time series graphics and statistics
library(feasts)
# Tidy time series data
library(tsibbledata)
```

`tsibble` objects

A `tsibble` allows storage and manipulation of time series in R.

It contains:

- Measured variable(s): numbers of interest
- Key variable(s): unique identifiers for each series
- An index: time information about the observation

tsibble objects

Example

```r
library(tsibble)
y <- tsibble(year = 2012:2016,
  y = c(123,39,78,52,110), index = year)
y
```

```
## # A tsibble: 5 x 2 [1Y]
##    year     y
##   <int> <dbl>
## 1  2012   123
## 2  2013    39
## 3  2014    78
## 4  2015    52
```

The `tsibble` index

Common time index variables can be created with these functions:

Frequency	Function
Annual	start:end
Quarterly	yearquarter()
Monthly	yearmonth()
Weekly	yearweek()
Daily	as_date(), ymd()
Sub-daily	as_datetime()

The key to many time series

Year	Length	Sex	Time
1896	100m	men	12.0
1928	100m	women	12.2
1900	200m	men	22.2
1948	200m	women	24.4
1896	400m	men	54.2
1964	400m	women	52.0

The key to many time series

```
olympic_running %>% as_tsibble(
  key = id(Length, Sex), index = Year)
```

```
## Warning: id() is deprecated for creating ke
## Please use key = c(Length, Sex).
```

```
## # A tsibble: 312 x 4 [4Y]
## # Key:         Length, Sex [14]
##      Year Length Sex    Time
##     <dbl> <fct>  <chr> <dbl>
##   1  1896 100m   men      12
##   2  1900 100m   men      11
```

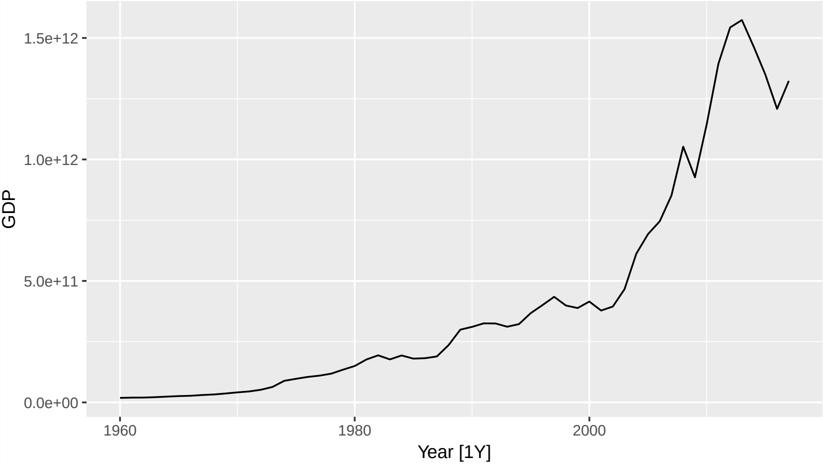
Australian GDP

```r
aus_economy <- global_economy %>%
  filter(Code == "AUS")
```

```
## # A tsibble: 58 x 9 [1Y]
## # Key:       Country [1]
##    Country Code   Year     GDP Growth   CPI
##    <fct>   <fct> <dbl>   <dbl>  <dbl> <dbl>
##  1 Austra~ AUS    1960 1.86e10  NA     7.96
##  2 Austra~ AUS    1961 1.96e10   2.49  8.14
##  3 Austra~ AUS    1962 1.99e10   1.30  8.12
##  4 Austra~ AUS    1963 2.15e10   6.21  8.17
##  5 Austra~ AUS    1964 2.38e10   6.98  8.40
##  6 Austra~ AUS    1965 2.59e10   5.98  8.69
##  7 Austra~ AUS    1966 2.73e10   2.38  8.98
```

Australian GDP

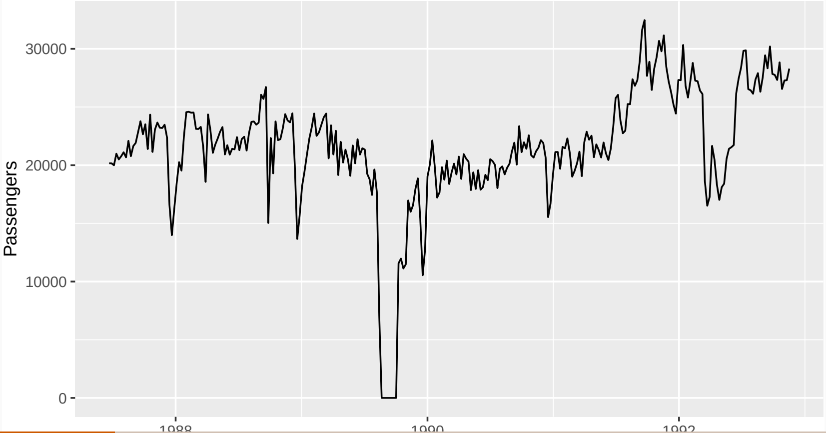
```r
aus_economy %>% autoplot(GDP)
```



Outline

Time plots

```
ansett %>%
  filter(Airports=="MEL-SYD", Class=="Economy") %>%
  autoplot(Passengers)
```



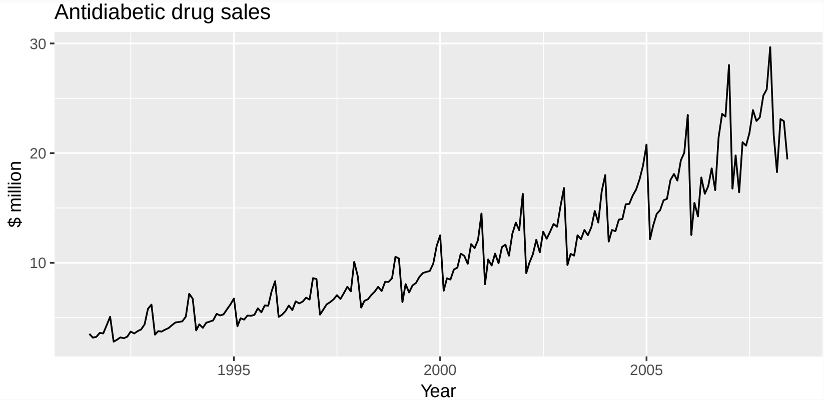
Time plots

```
a10 <- PBS %>%
  filter(ATC2 == "A10") %>%
  summarise(Cost = sum(Cost)/1e6)
```

```
## # A tsibble: 204 x 2 [1M]
##        Month  Cost
##        <mth> <dbl>
##  1 1991 Jul  3.53
##  2 1991 Aug  3.18
##  3 1991 Sep  3.25
##  4 1991 Oct  3.61
```

Time plots

```
a10 %>% autoplot(Cost) +
  ylab("$ million") + xlab("Year") +
  ggtitle("Antidiabetic drug sales")
```

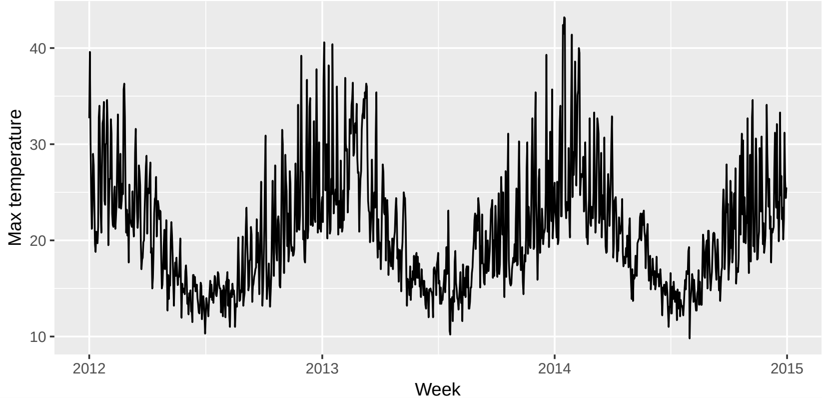


Your turn

- Create plots of the following time series: Bricks from `aus_production`, Lynx from `pelt`, Google from `gafa_stock`
- Use `help()` to find out about the data in each series.
- For the last plot, modify the axis labels and title.

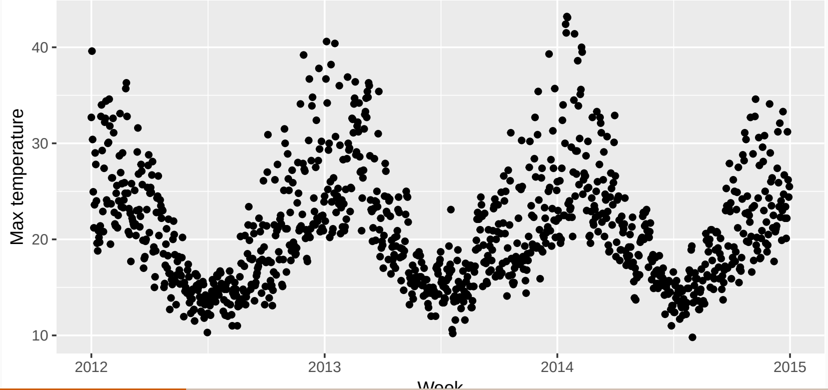
Are time plots best?

```
maxtemp %>%
  autoplot(Temperature) +
  xlab("Week") + ylab("Max temperature")
```

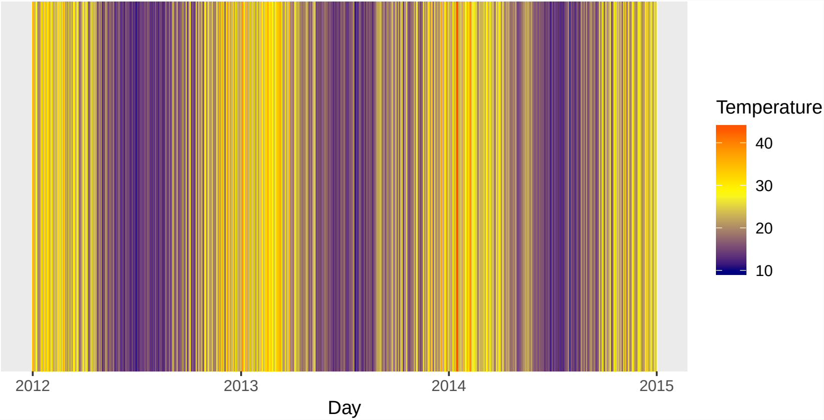


Are time plots best?

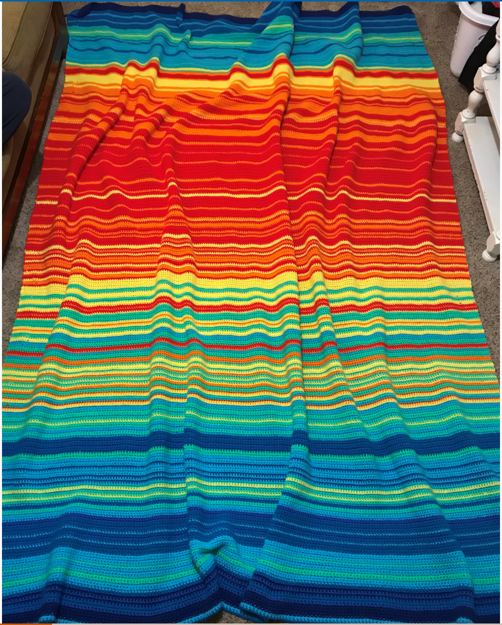
```
maxtemp %>%
  ggplot(aes(x = Day, y = Temperature)) +
  geom_point() +
  xlab("Week") + ylab("Max temperature")
```



Are time plots best?



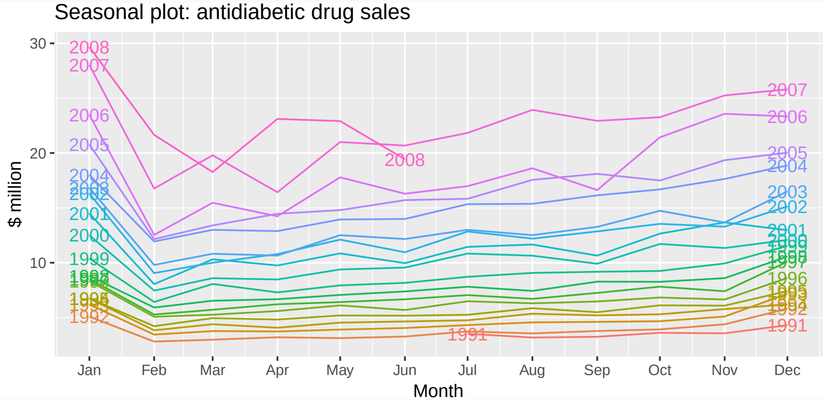
Are time plots best?



Outline

Seasonal plots

```r
a10 %>% gg_season(Cost, labels = "both") +
  ylab("$ million") +
  ggtitle("Seasonal plot: antidiabetic drug sales")
```

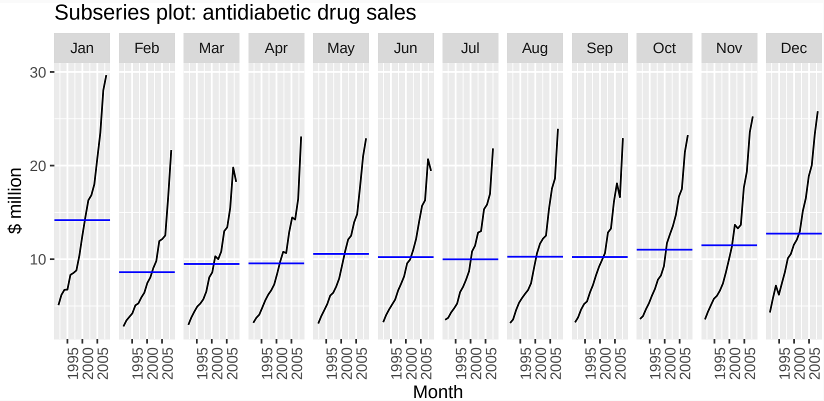


Seasonal plots

- Data plotted against the individual “seasons” in which the data were observed. (In this case a “season” is a month.)
- Something like a time plot except that the data from each season are overlapped.
- Enables the underlying seasonal pattern to be seen more clearly, and also allows any substantial departures from the seasonal pattern to be easily identified.
- In R: `gg_season()`

Seasonal subseries plots

```r
a10 %>%
  gg_subseries(Cost) + ylab("$ million") +
  ggtitle("Subseries plot: antidiabetic drug sales")
```

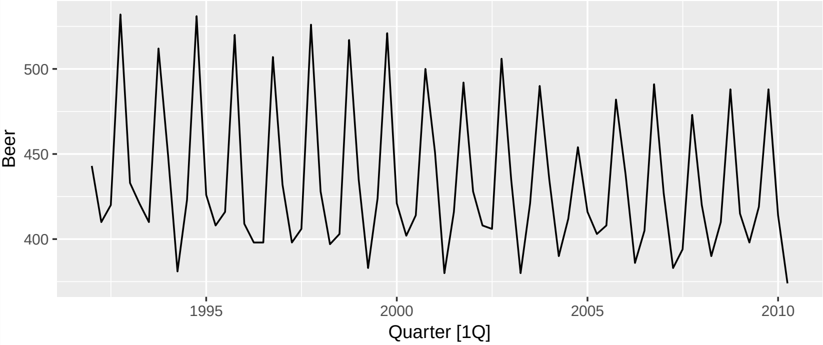


Seasonal subseries plots

- Data for each season collected together in time plot as separate time series.
- Enables the underlying seasonal pattern to be seen clearly, and changes in seasonality over time to be visualized.
- In R: `gg_subseries()`

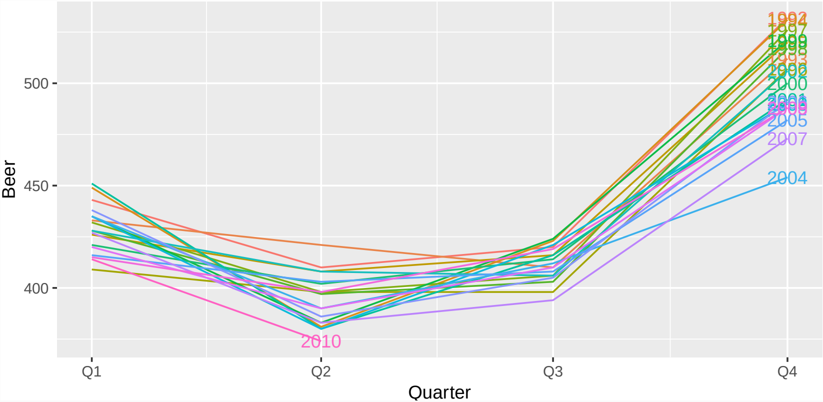
Quarterly Australian Beer Production

```r
beer <- aus_production %>%
  select(Quarter, Beer) %>%
  filter(year(Quarter) >= 1992)
beer %>% autoplot(Beer)
```



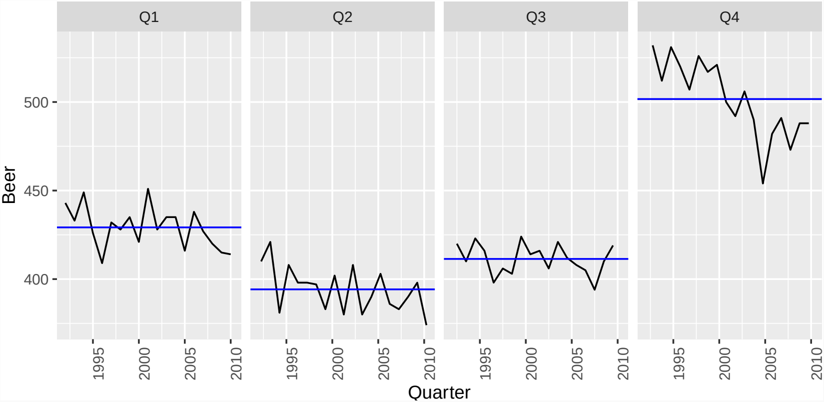
Quarterly Australian Beer Production

```r
beer %>% gg_season(Beer, labels="right")
```



Quarterly Australian Beer Production

```
beer %>% gg_subseries(Beer)
```



Your turn

Look at the quarterly tourism data for the Snowy Mountains

```
snowy <- filter(tourism,
  Region == "Snowy Mountains",
  Purpose == "Holiday")
```

- Use `autoplot()`, `gg_season()` and `gg_subseries()` to explore the data.
- What do you learn?

Outline

Time series patterns

Trend pattern exists when there is a long-term increase or decrease in the data.

Seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

Time series components

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

Time series patterns

```r
as_tsibble(fma::elec) %>%
  filter(index >= 1980) %>%
  autoplot(value) + xlab("Year") + ylab("GWh") +
  ggtitle("Australian electricity production")
```

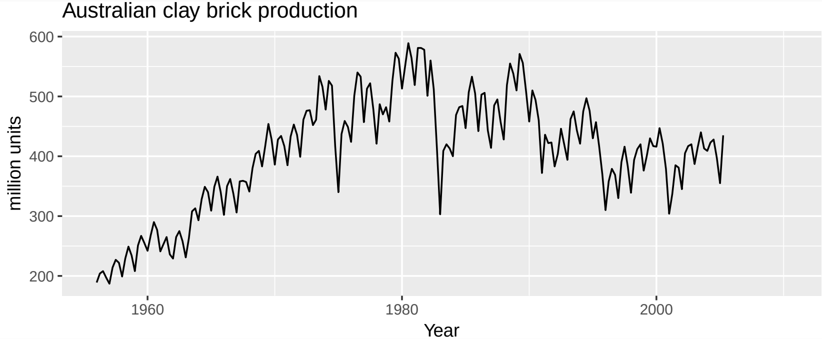
```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
```

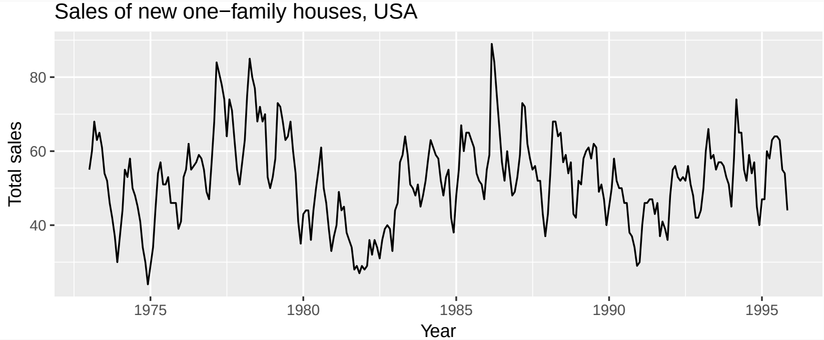
Time series patterns

```
aus_production %>%
  autoplot(Bricks) +
  ggtitle("Australian clay brick production") +
  xlab("Year") + ylab("million units")
```



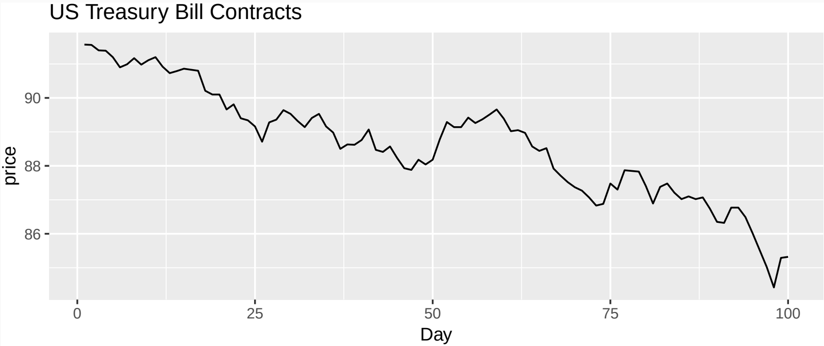
Time series patterns

```r
as_tsibble(fma::hsales) %>%
  autoplot(value) +
  ggtitle("Sales of new one-family houses, USA") +
  xlab("Year") + ylab("Total sales")
```



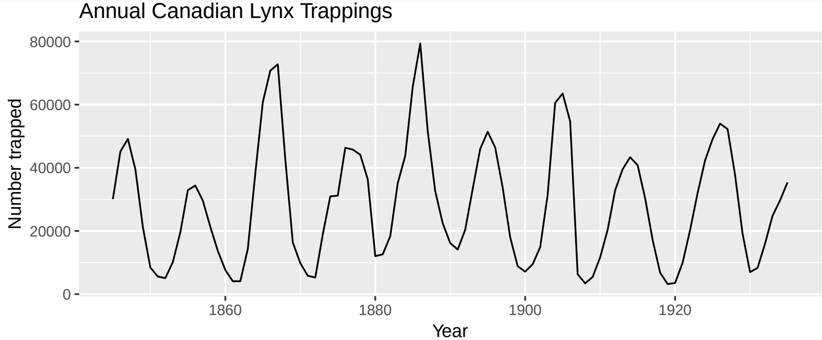
Time series patterns

```r
as_tsibble(fma::ustreas) %>%
  autoplot(value) +
  ggtitle("US Treasury Bill Contracts") +
  xlab("Day") + ylab("price")
```



Time series patterns

```r
pelt %>%
  autoplot(Lynx) +
  ggtitle("Annual Canadian Lynx Trappings") +
  xlab("Year") + ylab("Number trapped")
```



Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

The timing of peaks and troughs is predictable with seasonal data, but unpredictable in the long term with cyclic data.

Outline

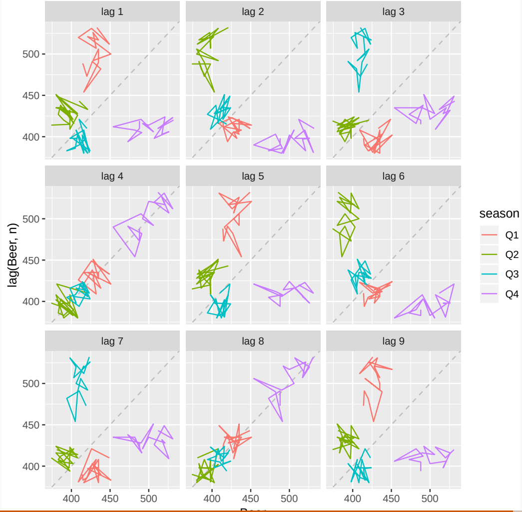
Example: Beer production

```
new_production <- aus_production %>%
  filter(year(Quarter) >= 1992)
new_production
```

```
## # A tsibble: 74 x 7 [1Q]
##        Quarter  Beer Tobacco Bricks Cement
##          <qtr> <dbl>   <dbl>  <dbl>  <dbl>
##   1    1992 Q1   443    5777    383   1289
##   2    1992 Q2   410    5853    404   1501
##   3    1992 Q3   420    6416    446   1539
##   4    1992 Q4   532    5825    420   1568
##   5    1993 Q1   433    5724    394   1450
##   6    1993 Q2   421    6036    462   1668
```

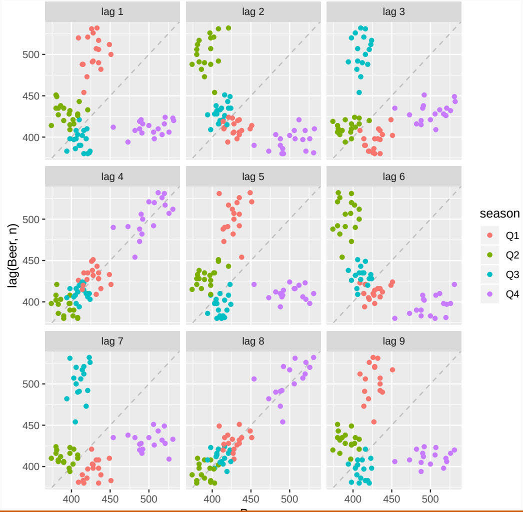
Example: Beer production

```
new_production %>% gg_lag(Beer)
```



Example: Beer production

```
new_production %>% gg_lag(Beer, geom='point')
```



Lagged scatterplots

- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

We measure the relationship between:

- y_t and y_{t-1}
- y_t and y_{t-2}
- y_t and y_{t-3}
- etc.

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k. Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})$$

and $$r_k = c_k / c_0$$

Autocorrelation

We denote the sample autocovariance at lag k by c_k and the sample autocorrelation at lag k by r_k. Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})$$

$$\text{and} \qquad r_k = c_k / c_0$$

- r_1 indicates how successive values of y relate to each other
- r_2 indicates how y values two periods apart relate to each other
- r_k is *almost* the same as the sample correlation between y_t and y_{t-k}.

Autocorrelation

Results for first 9 lags for beer data:

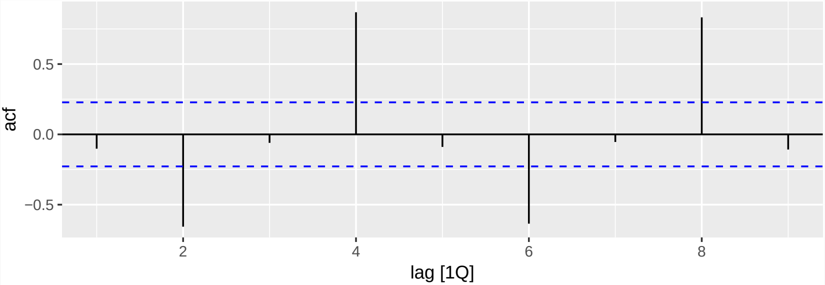
```
new_production %>% ACF(Beer, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1Q]
##      lag     acf
##    <lag>   <dbl>
## 1     1Q -0.102
## 2     2Q -0.657
## 3     3Q -0.0603
## 4     4Q  0.869
## 5     5Q -0.0892
## 6     6Q -0.635
## 7     7Q -0.0542
## 8     8Q  0.832
```

Autocorrelation

Results for first 9 lags for beer data:

```
new_production %>% ACF(Beer, lag_max = 9) %>% autoplot()
```

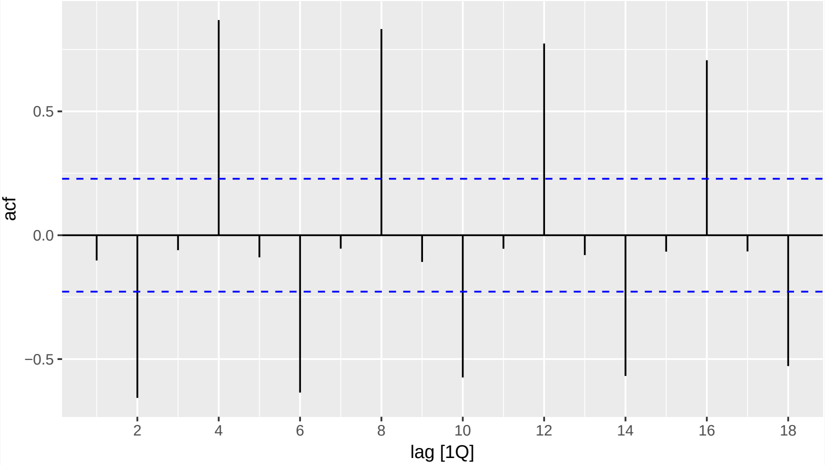


Autocorrelation

- r_4 higher than for the other lags. This is due to **the seasonal pattern in the data**: the peaks tend to be **4 quarters** apart and the troughs tend to be **2 quarters** apart.
- r_2 is more negative than for the other lags because troughs tend to be 2 quarters behind peaks.
- Together, the autocorrelations at lags 1, 2, $\dots$, make up the *autocorrelation* or ACF.
- The plot is known as a **correlogram**

ACF

```
new_production %>% ACF(Beer) %>% autoplot()
```

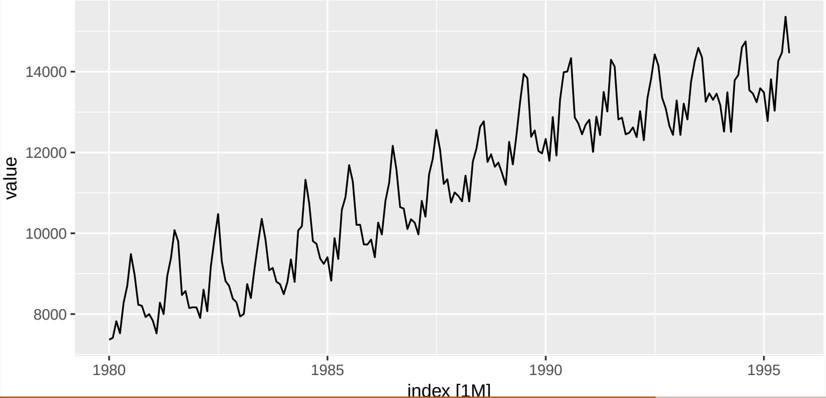


Trend and seasonality in ACF plots

- When data have a trend, the autocorrelations for small lags tend to be large and positive.
- When data are seasonal, the autocorrelations will be larger at the seasonal lags (i.e., at multiples of the seasonal frequency)
- When data are trended and seasonal, you see a combination of these effects.

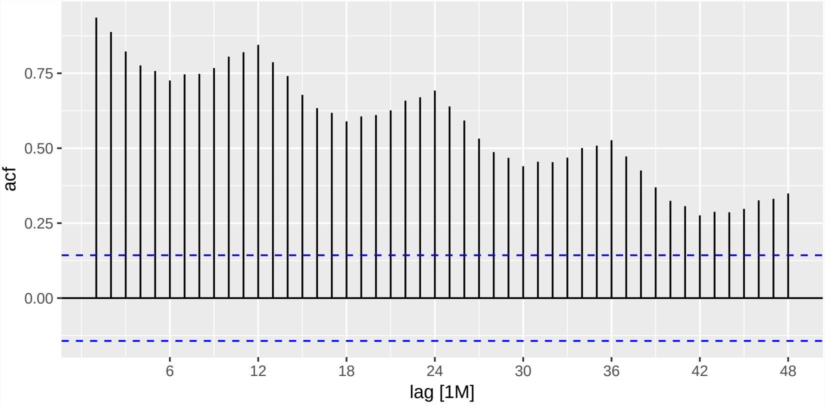
Aus monthly electricity production

```r
elec2 <- as_tsibble(fma::elec) %>%
  filter(year(index) >= 1980)
elec2 %>% autoplot(value)
```



Aus monthly electricity production

```
elec2 %>% ACF(value, lag_max=48) %>%
  autoplot()
```



Aus monthly electricity production

Time plot shows clear trend and seasonality.

The same features are reflected in the ACF.

- The slowly decaying ACF indicates trend.
- The ACF peaks at lags 12, 24, 36, . . . , indicate seasonality of length 12.

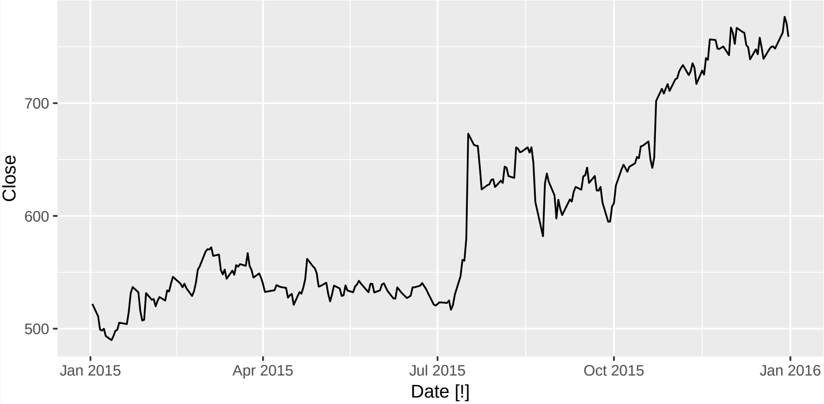
Google stock price

```
google_2015 <- gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) == 2015) %>%
  select(Date, Close)
google_2015
```

```
## # A tsibble: 252 x 2 [!]
##    Date       Close
##    <date>     <dbl>
##  1 2015-01-02  522.
##  2 2015-01-05  511.
##  3 2015-01-06  499.
##  4 2015-01-07  498.
##  5 2015-01-08  500.
##  6 2015-01-09  493.
```

Google stock price

```
google_2015 %>% autoplot(Close)
```



Google stock price

```
google_2015 %>%
  ACF(Close, lag_max=100)
# Error: Can't handle tsibble of irregular interval.
```

Google stock price

```r
google_2015 %>%
  ACF(Close, lag_max=100)
# Error: Can't handle tsibble of irregular interval.
```

```r
google_2015
```

```
## # A tsibble: 252 x 2 [!]
##    Date       Close
##    <date>     <dbl>
##  1 2015-01-02  522.
##  2 2015-01-05  511.
##  3 2015-01-06  499.
```

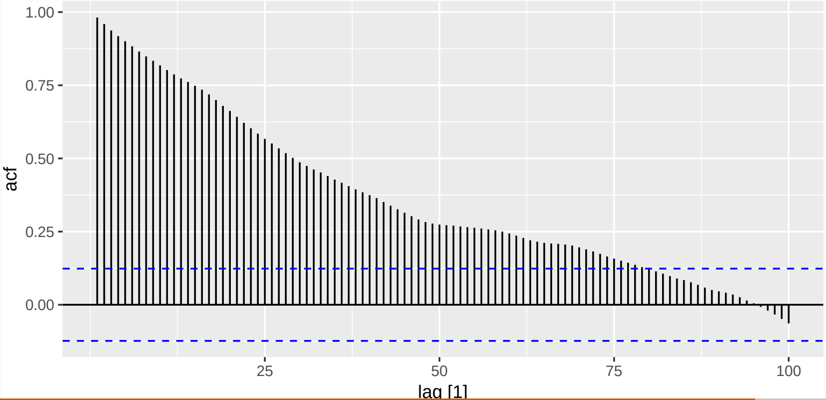
Google stock price

```
google_2015 <- google_2015 %>%
  mutate(trading_day = row_number()) %>%
  update_tsibble(index=trading_day, regular=TRUE)
google_2015
```

```
## # A tsibble: 252 x 3 [1]
##    Date       Close trading_day
##    <date>     <dbl>       <int>
##  1 2015-01-02  522.           1
##  2 2015-01-05  511.           2
##  3 2015-01-06  499.           3
##  4 2015-01-07  498.           4
##  5 2015-01-08  500.           5
##  6 2015-01-09  493.           6
```

Google stock price

```
google_2015 %>%
  ACF(Close, lag_max=100) %>%
  autoplot()
```



Your turn

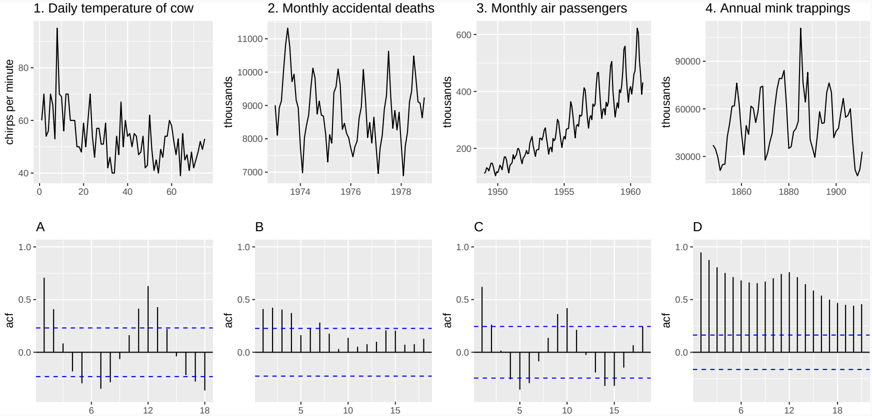
We have introduced the following functions:

- `gg_lag`
- `ACF`

Explore the following time series using these functions. Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

- Bricks from `aus_production`
- Lynx from `pelt`
- Victorian Electricity Demand from `aus_elec`

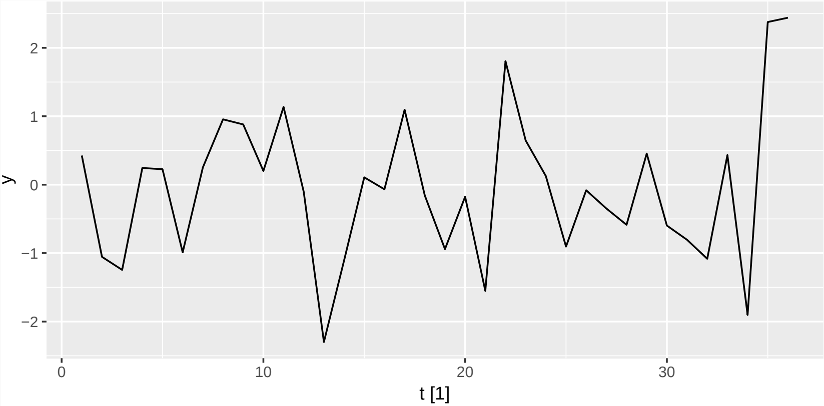
Which is which?



Outline

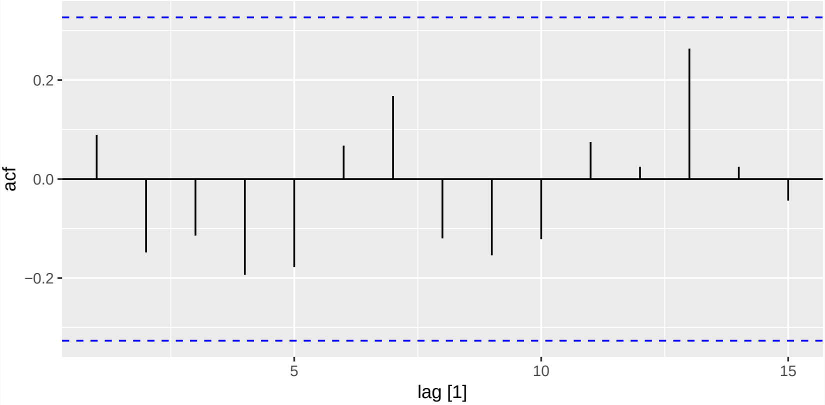
Example: White noise

```r
wn <- tsibble(t = seq_len(36), y = rnorm(36),
              index = t)
wn %>% autoplot(y)
```



Example: White noise

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
0.089	-0.148	-0.114	-0.193	-0.178	0.067	0.168	-0.120	-0.154	-0.121



Sampling distribution of autocorrelations

Sampling distribution of r_k for white noise data is asymptotically $\text{N}(0,1/T)$.

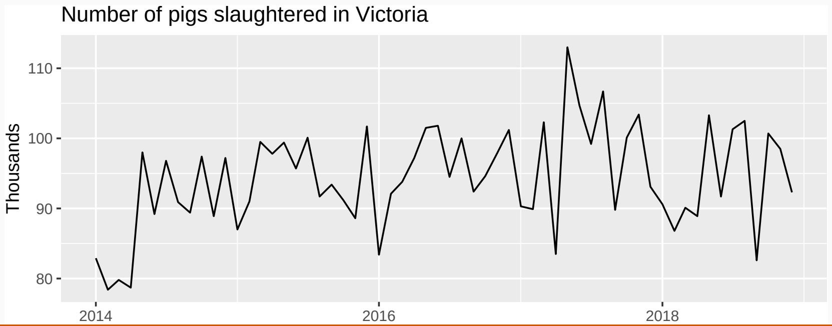
Sampling distribution of autocorrelations

Sampling distribution of r_k for white noise data is asymptotically N(0,1/T).

- 95% of all r_k for white noise must lie within $\pm 1.96/\sqrt{T}$.
- If this is not the case, the series is probably not WN.
- Common to plot lines at $\pm 1.96/\sqrt{T}$ when plotting ACF. These are the **critical values**.

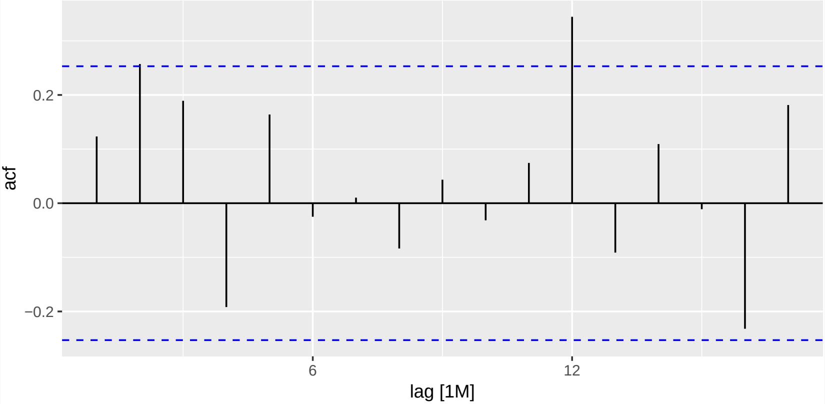
Example: Pigs slaughtered

```
pigs <- aus_livestock %>%
  filter(State == "Victoria", Animal == "Pigs",
         year(Month) >= 2014)
pigs %>% autoplot(Count/1e3) +
  xlab("Year") + ylab("Thousands") +
  ggtitle("Number of pigs slaughtered in Victoria")
```



Example: Pigs slaughtered

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



Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows significant autocorrelation for lag 2 and 12.
- Indicate some slight seasonality.

Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 2014 through December 2018 (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows significant autocorrelation for lag 2 and 12.
- Indicate some slight seasonality.

These show the series is **not a white noise series**.

Your turn

You can compute the daily changes in the Google stock price in 2018 using

```
dgoog <- gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) >= 2018) %>%
  mutate(trading_day = row_number()) %>%
  update_tsibble(index=trading_day, regular=TRUE) %>%
  mutate(diff = difference(Close))
```

Does `diff` look like white noise?