Time Series Analysis & Forecasting Using R

Time series patterns and graphics

Bahman Rostami-Tabar

Outline

Outline

Learning outcome

You should be able to:

1. Create time series graphics
2. Identify key patterns in time series data

Outline

Key components of time series

- Level
- Trend
- Seasonal/cycle pattern
- Autocorrelation
- Unpredictable patterns/Noise

Time series patterns

Level The *level* of a time series describes the center of the series at any point.

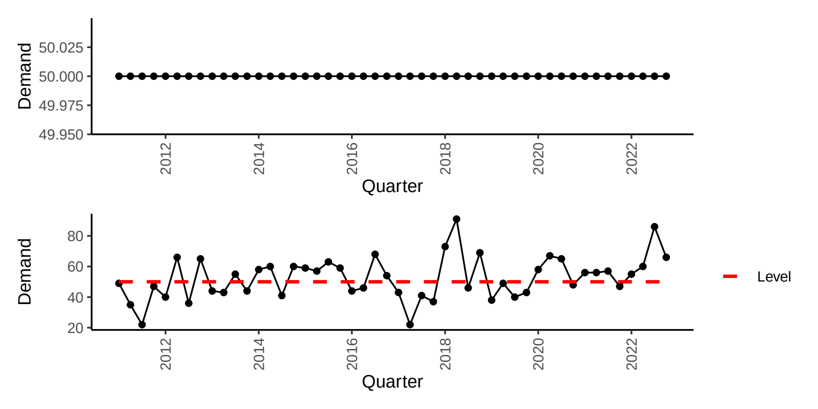
Trend pattern exists when there is a increase or decrease in the data.

Seasonal A consistent pattern over a fixed period of time (e.g., every 24 hours you have the same shape (daily seasonal pattern), every seven days you have the same shape (weekly seasonal pattern), etc).

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

Outline

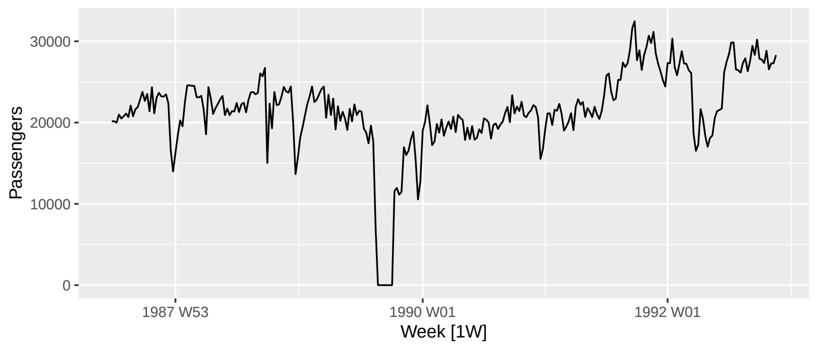
Time plots



Demand
50.025
50.000
49.975
49.950
2012
2014
2016
2018
2020
2022
Quarter
Demand
80
60
40
20
2012
2014
2016
2018
2020
2022
Quarter
Level

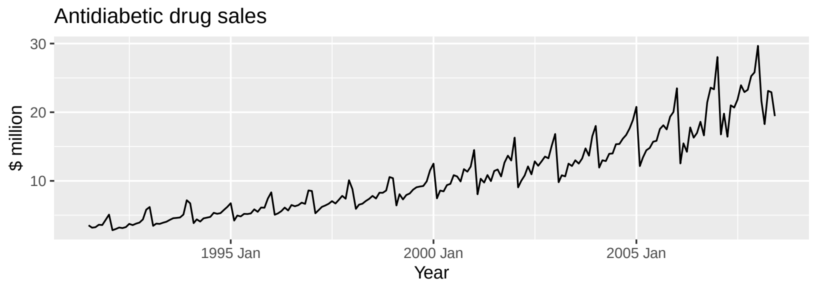
Time plots

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

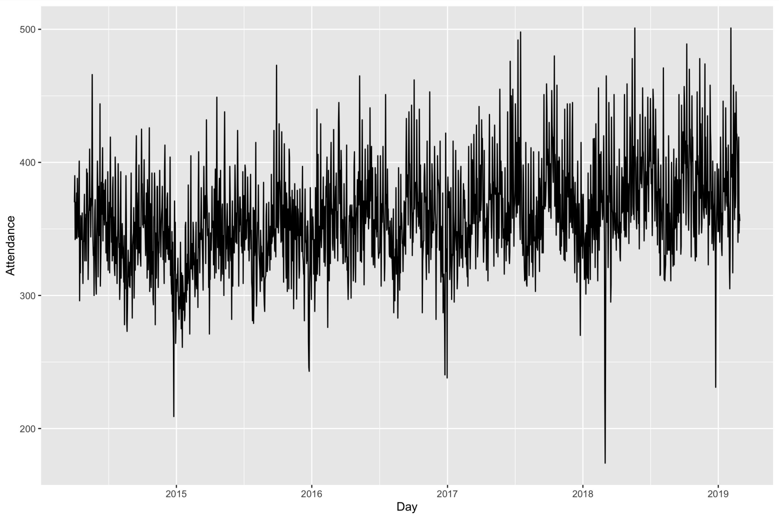


Time plots

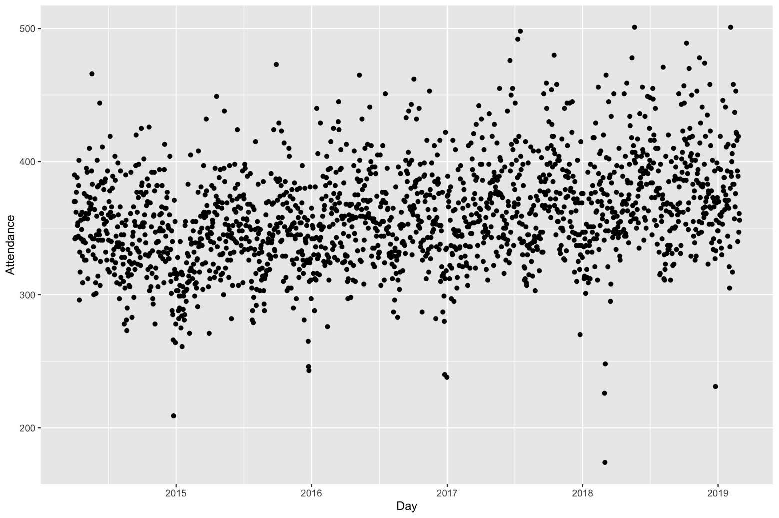
```r
PBS %>% filter(ATC2 == "A10") %>%
  summarise(Cost = sum(Cost)/1e6) %>% autoplot(Cost) +
  ylab("$ million") + xlab("Year") +
  ggtitle("Antidiabetic drug sales")
```



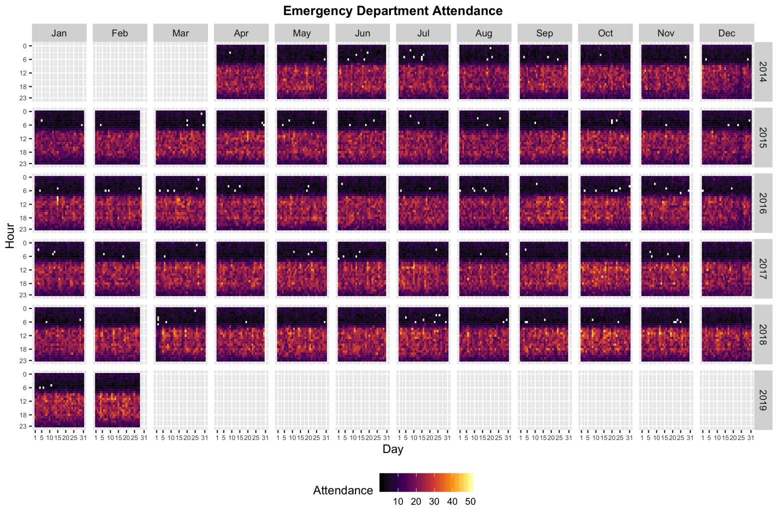
Are time plots best?



Are time plots best?



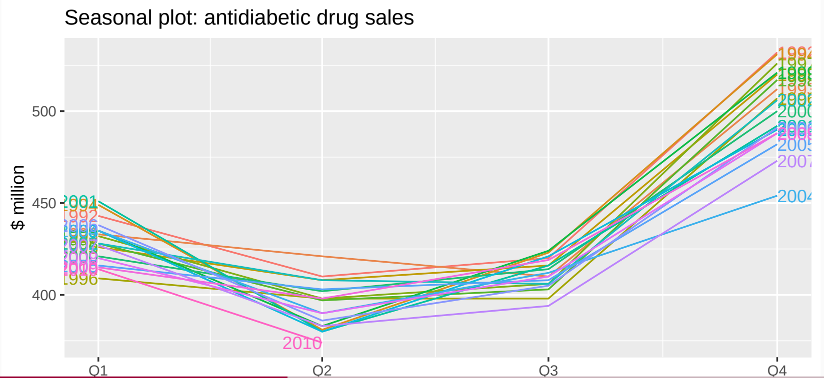
Are time plots best?



Outline

Seasonal plots

```r
new_production <- aus_production %>%
  filter(year(Quarter) >= 1992)
new_production %>% gg_season(Beer, labels = "both")+
  ylab("$ million") +
  ggtitle("Seasonal plot: antidiabetic drug sales")
```

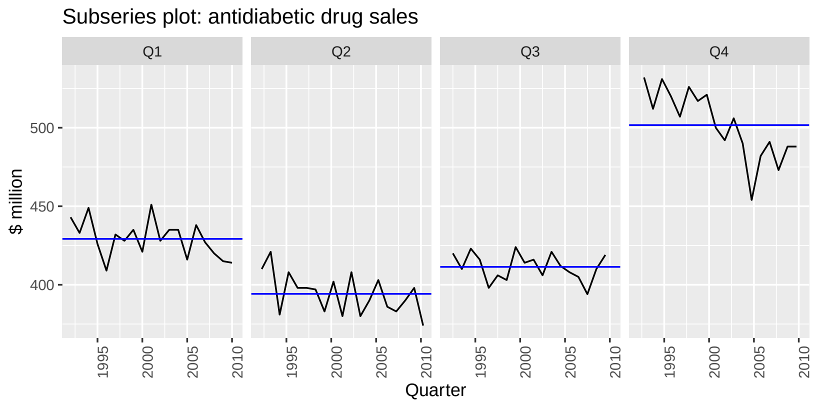


Seasonal plots

- 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

```
new_production %>% gg_subseries(Beer) + ylab("$ million
  ggtitle("Subseries plot: antidiabetic drug sales")
```

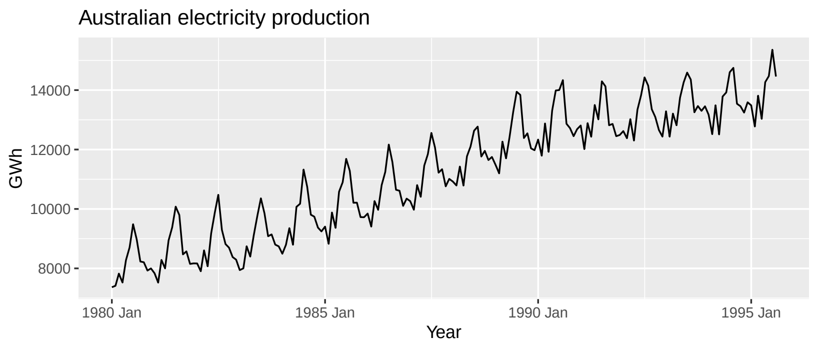


Seasonal subseries plots

- 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()`

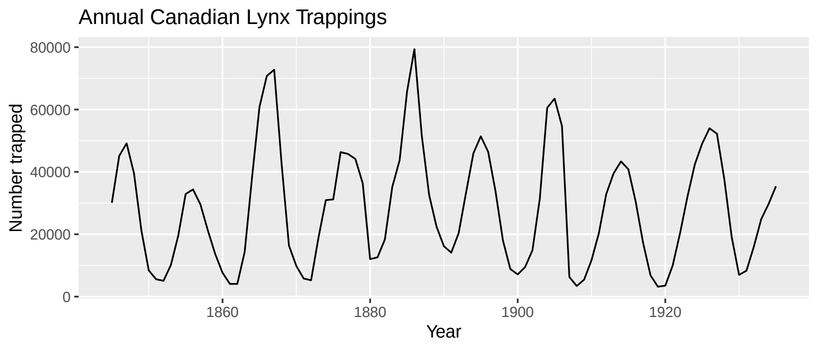
Time series patterns

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



Time series patterns

```
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

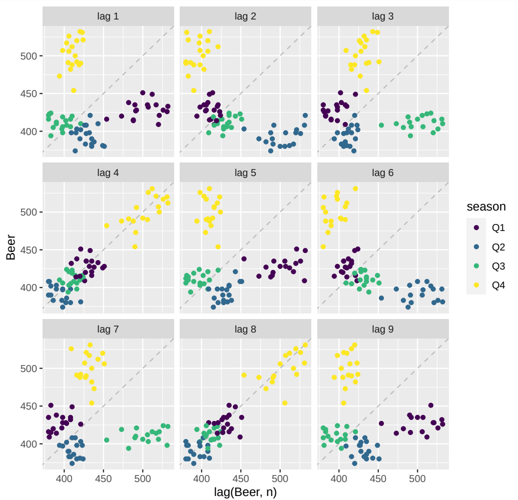
Example: Beer production

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

```
## # A tsibble: 74 x 7 [1Q]
##    Quarter  Beer Tobacco Bricks Cement Electrici
##      <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <db
##  1 1992 Q1   443    5777    383   1289       383
##  2 1992 Q2   410    5853    404   1501       397
##  3 1992 Q3   420    6416    446   1539       422
##  4 1992 Q4   532    5825    420   1568       384
##  5 1993 Q1   433    5724    394   1450       394
##  6 1993 Q2   421    6036    462   1668       413
```

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

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

and $$r_k = c_k/c_0$$

- 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:

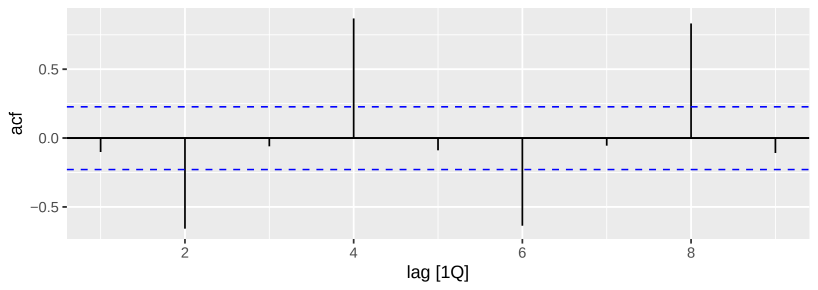
```
new_production %>% ACF(Beer, lag_max = 9)
```

```
## # A tsibble: 9 x 2 [1Q]
##          lag     acf
##    <cf_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

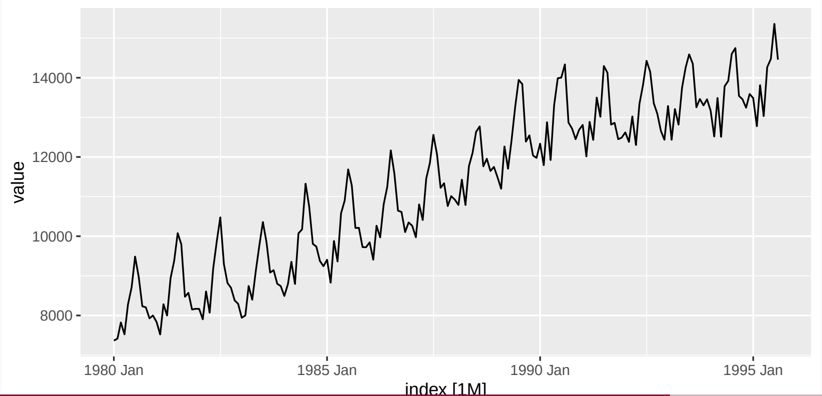
- r_4 higher than for the other lags. This is due to **the seasonal pattern in the data**: the peaks tend to be **4 quarters** apart and the troughs tend to be **2 quarters** apart.
- r_2 is more negative than for the other lags because troughs tend to be 2 quarters behind peaks.
- Together, the autocorrelations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

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.

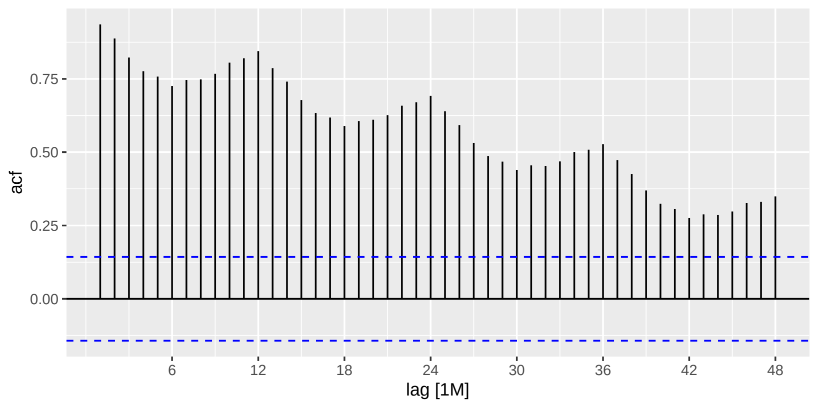
Aus monthly electricity production

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



Aus monthly electricity production

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



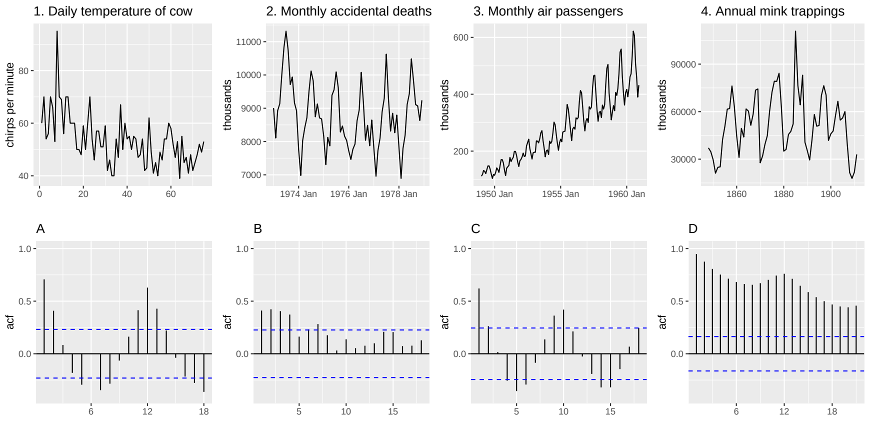
Aus monthly electricity production

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.

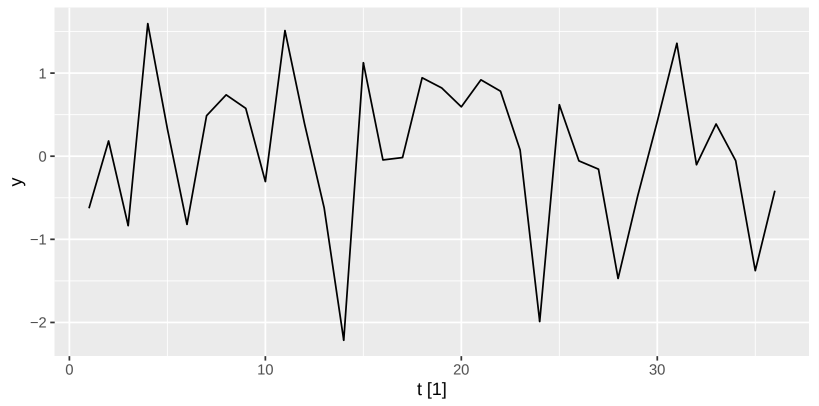
Which is which?



Outline

Example: White noise

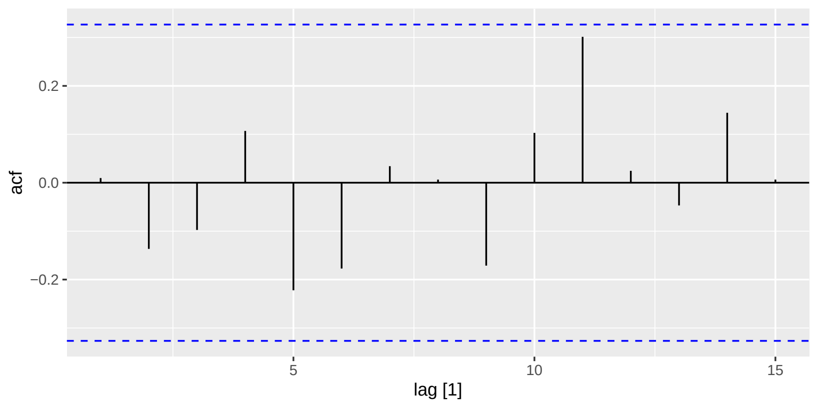
```
set.seed(1)
wn <- tsibble(t = seq_len(36), y = rnorm(36),
              index = t)
wn %>% autoplot(y)
```



Example: White noise

lag	acf
1	0.010
2	-0.137
3	-0.098
4	0.107
5	-0.222
6	-0.177
7	0.034
8	0.006
9	-0.171
10	0.103

Example: White noise



Sampling distribution of autocorrelations

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

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**.

Outline

Lab Session 2