MONASH University

MONASH BUSINESS SCHOOL

Tidy Time Series & Forecasting in R

tsibble feasts Fable

2. Time series graphics

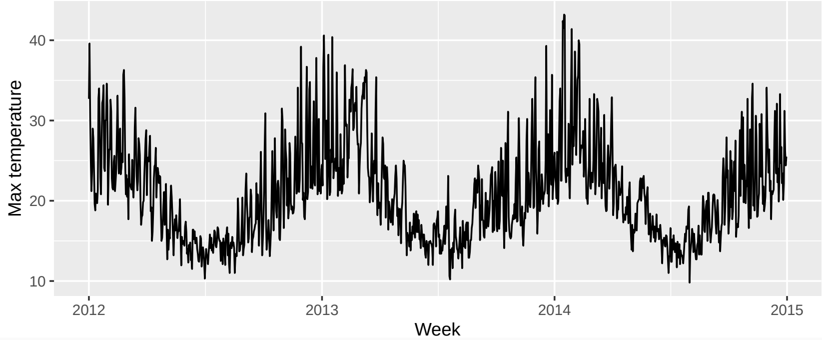
robjhyndman.com/workshop2020

Outline

Outline

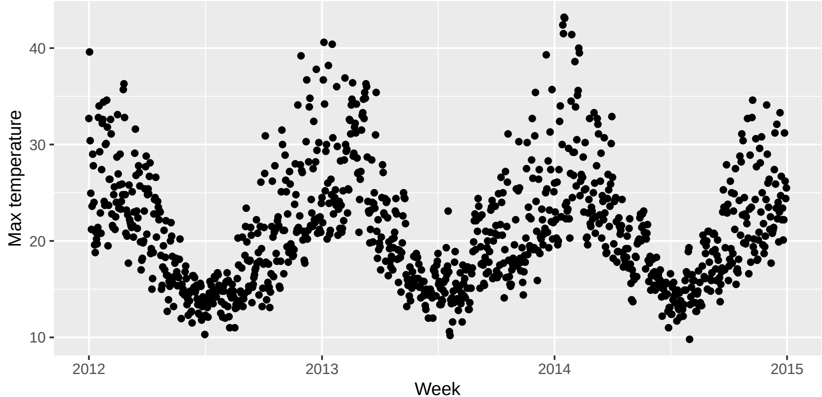
Are line plots best?

```
maxtemp <- vic_elec %>%
  index_by(Day = date(Time)) %>%
  summarise(Temperature = max(Temperature))
maxtemp %>%
  autoplot(Temperature) +
  xlab("Week") + ylab("Max temperature")
```



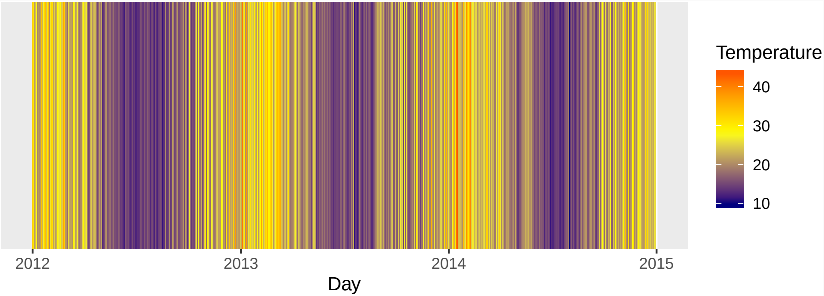
Are line plots best?

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

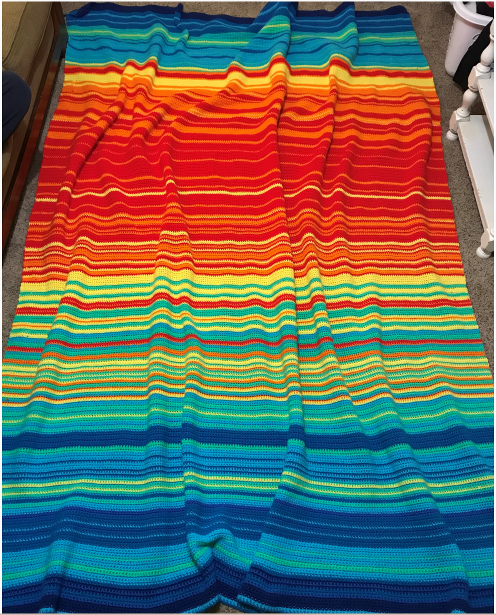


Are line plots best?

```r
maxtemp %>%
  ggplot(aes(x = Day, y = 1)) +
  geom_tile(aes(fill = Temperature)) +
  scale_fill_gradient2(low = "navy", mid = "yellow",
                       high = "red", midpoint=28) +
  ylab("") + scale_y_discrete(expand=c(0,0))
```



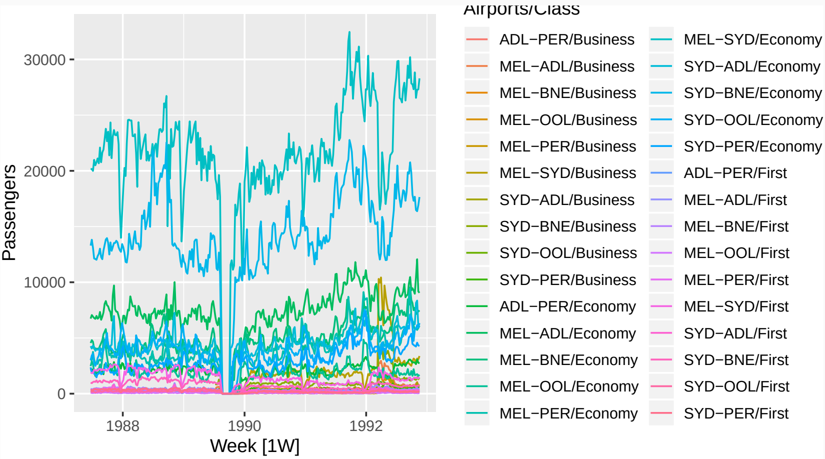
Are line plots best?



Ansett airlines

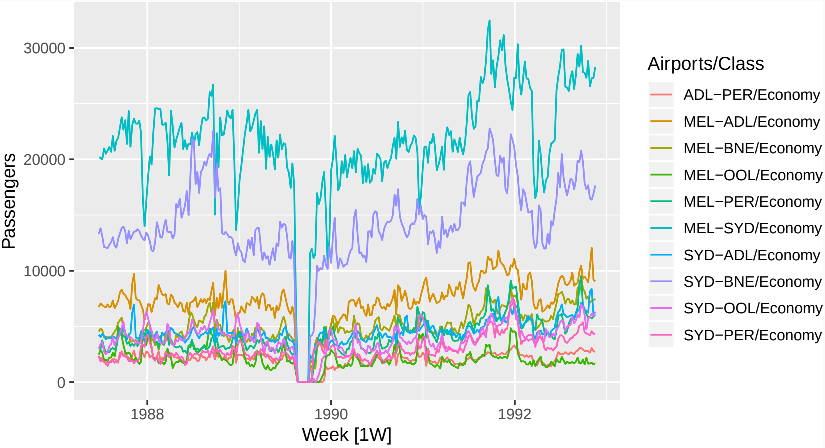
Ansett airlines

```r
ansett %>%
  autoplot(Passengers)
```



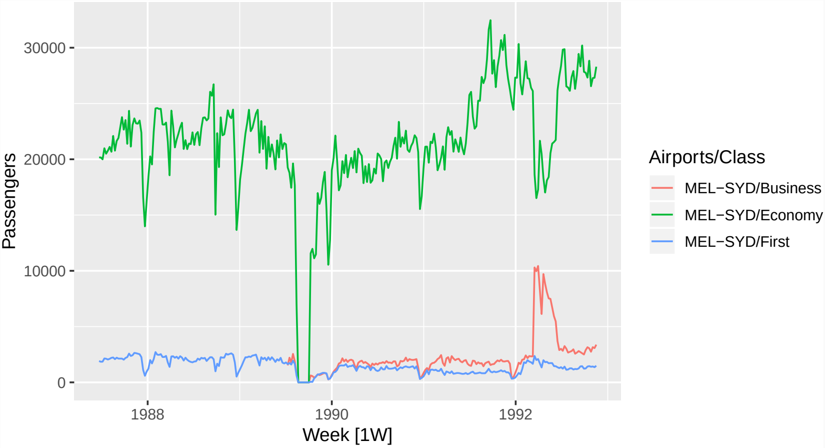
Ansett airlines

```r
ansett %>%
  filter(Class=="Economy") %>%
  autoplot(Passengers)
```



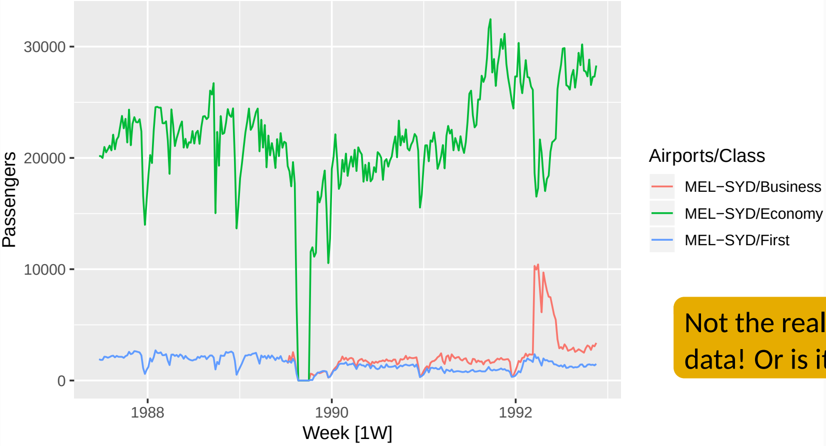
Ansett airlines

```r
ansett %>%
  filter(Airports=="MEL-SYD") %>%
  autoplot(Passengers)
```



Ansett airlines

```r
ansett %>%
  filter(Airports=="MEL-SYD") %>%
  autoplot(Passengers)
```



Not the real data! Or is it?

Outline

Lab Session 2

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

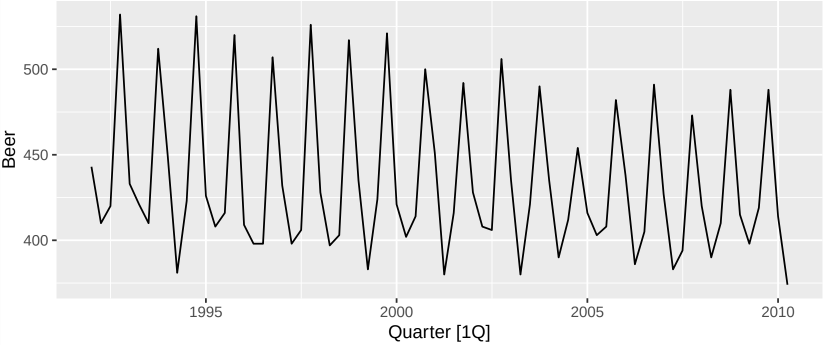
Outline

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()`

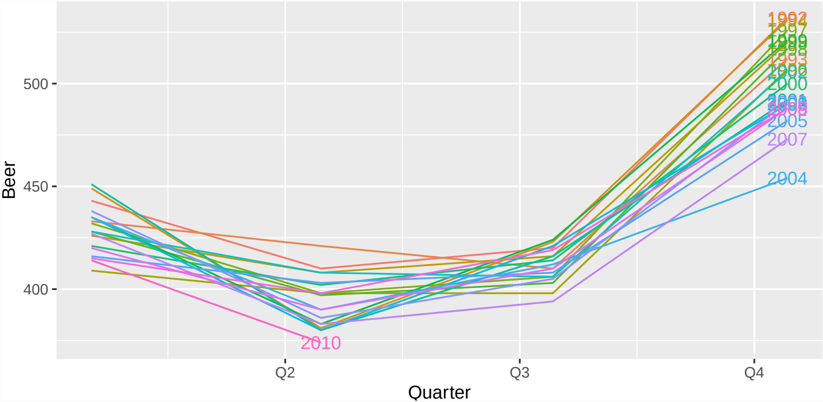
Quarterly Australian Beer Production

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



Quarterly Australian Beer Production

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



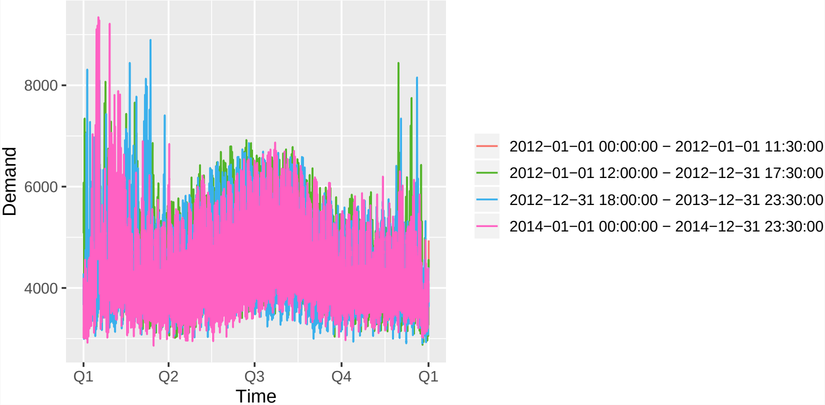
Multiple seasonal periods

```r
vic_elec
```

```
## # A tsibble: 52,608 x 5 [30m] <UTC>
##    Time                Demand Temperature Date       Holiday
##    <dttm>               <dbl>       <dbl> <date>     <lgl>
##  1 2012-01-01 00:00:00  4263.        21.0 2012-01-01 TRUE
##  2 2012-01-01 00:30:00  4049.        20.7 2012-01-01 TRUE
##  3 2012-01-01 01:00:00  3878.        20.6 2012-01-01 TRUE
##  4 2012-01-01 01:30:00  4036.        20.4 2012-01-01 TRUE
##  5 2012-01-01 02:00:00  3866.        20.2 2012-01-01 TRUE
##  6 2012-01-01 02:30:00  3694.        20.1 2012-01-01 TRUE
##  7 2012-01-01 03:00:00  3562.        19.6 2012-01-01 TRUE
##  8 2012-01-01 03:30:00  3433.        19.1 2012-01-01 TRUE
##  9 2012-01-01 04:00:00  3359.        19.0 2012-01-01 TRUE
## 10 2012-01-01 04:30:00  3331.        18.8 2012-01-01 TRUE
## # ... with 52,598 more rows
```

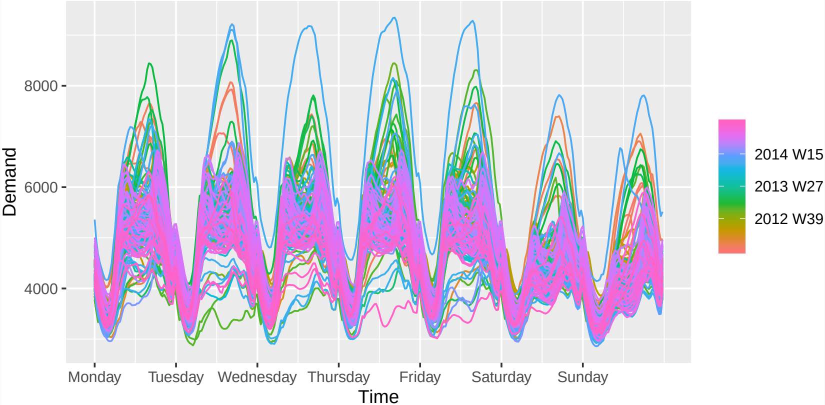
Multiple seasonal periods

```
vic_elec %>% gg_season(Demand)
```



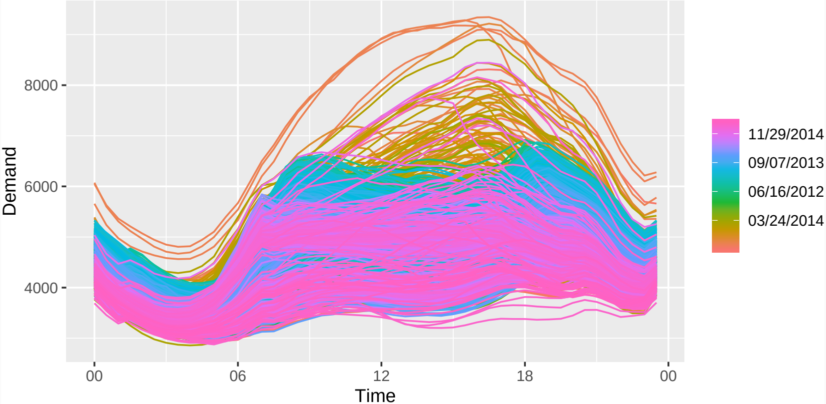
Multiple seasonal periods

```
vic_elec %>% gg_season(Demand, period="week")
```



Multiple seasonal periods

```r
vic_elec %>% gg_season(Demand, period="day")
```

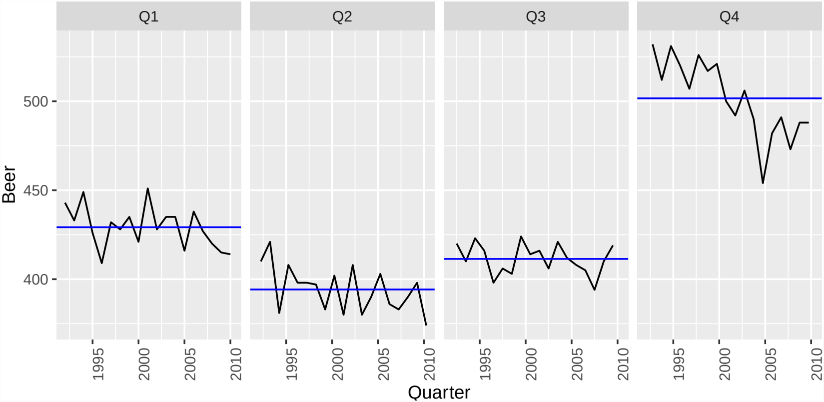


Seasonal subseries plots

- 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

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



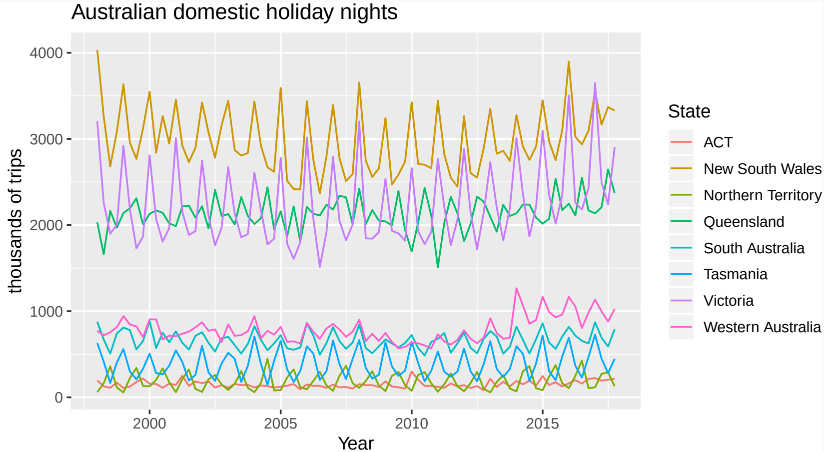
Australian holidays

```r
holidays <- tourism %>%
  filter(Purpose=="Holiday") %>%
  group_by(State) %>%
  summarise(Trips = sum(Trips))
```

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter Trips
##    <chr>   <qtr> <dbl>
##  1 ACT   1998 Q1  196.
##  2 ACT   1998 Q2  127.
##  3 ACT   1998 Q3  111.
##  4 ACT   1998 Q4  170.
##  5 ACT   1999 Q1  108.
##  6 ACT   1999 Q2  125.
##  7 ACT   1999 Q3  178.
##  8 ACT   1999 Q4  218.
##  9 ACT   2000 Q1  158.
## 10 ACT   2000 Q2  155.
```

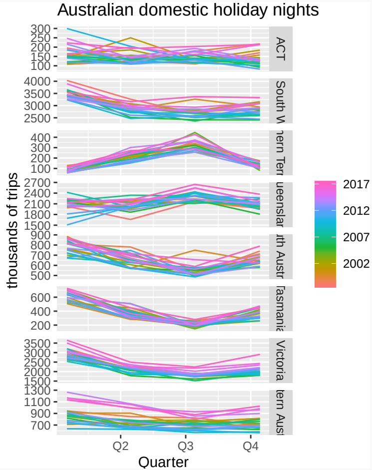
Australian holidays

```r
holidays %>% autoplot(Trips) +
  ylab("thousands of trips") + xlab("Year") +
  ggtitle("Australian domestic holiday nights")
```



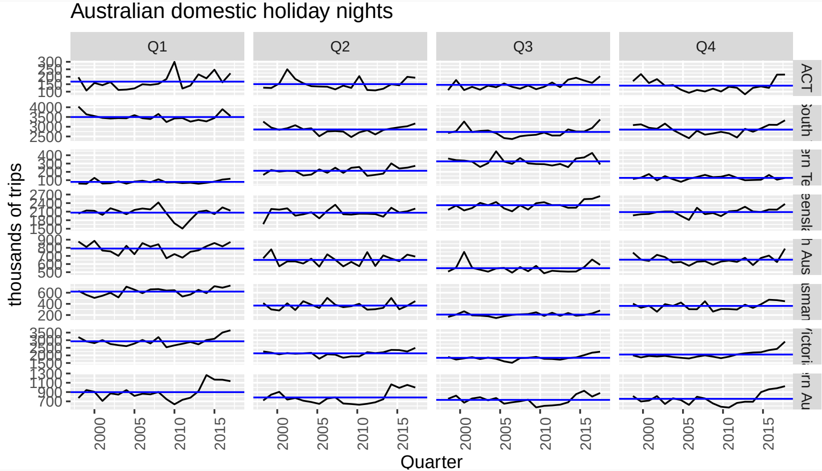
Seasonal plots

```r
holidays %>% gg_season(Trips) +
  ylab("thousands of trips") +
  ggtitle("Australian domestic holiday nights")
```



Seasonal subseries plots

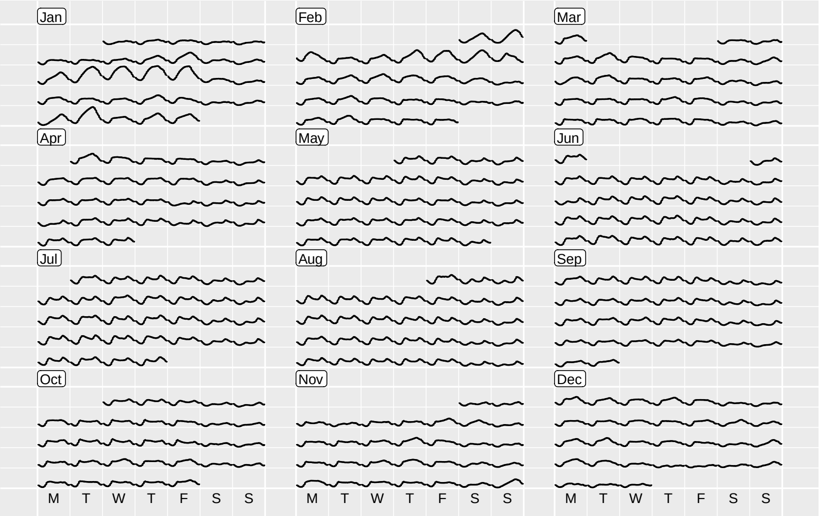
```r
holidays %>%
  gg_subseries(Trips) + ylab("thousands of trips") +
  ggtitle("Australian domestic holiday nights")
```



Calendar plots

```r
library(sugrrants)
vic_elec %>%
  filter(year(Date) == 2014) %>%
  mutate(Hour = hour(Time)) %>%
  frame_calendar(x = Hour, y = Demand, date = Date,
    nrow = 4) %>%
  ggplot(aes(x = .Hour, y = .Demand, group = Date)) +
  geom_line() -> p1
prettify(p1, size = 3,
  label.padding = unit(0.15, "lines"))
```

Calendar plots



Outline

Lab Session 3

1. Look at the quarterly tourism data for the Snowy Mountains

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

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

2. Produce a calendar plot for the `pedestrian` data from one location and one year.

Outline

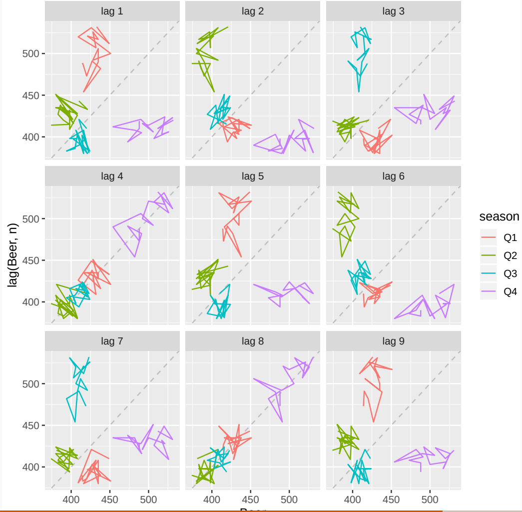
Example: Beer production

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

```
## # A tsibble: 74 x 7 [1Q]
##    Quarter  Beer Tobacco Bricks Cement Electricity   Gas
##      <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
##  1 1992 Q1   443    5777    383   1289       38332   117
##  2 1992 Q2   410    5853    404   1501       39774   151
##  3 1992 Q3   420    6416    446   1539       42246   175
##  4 1992 Q4   532    5825    420   1568       38498   129
##  5 1993 Q1   433    5724    394   1450       39460   116
##  6 1993 Q2   421    6036    462   1668       41356   149
##  7 1993 Q3   410    6570    475   1648       42949   163
##  8 1993 Q4   512    5675    443   1863       40974   138
##  9 1994 Q1   449    5311    421   1468       40162   127
```

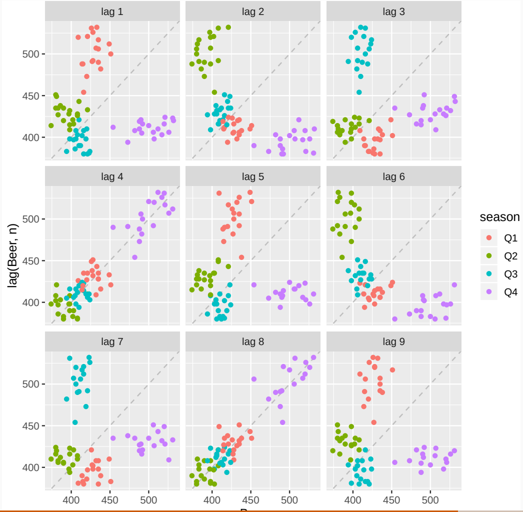
Example: Beer production

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



Example: Beer production

```r
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:

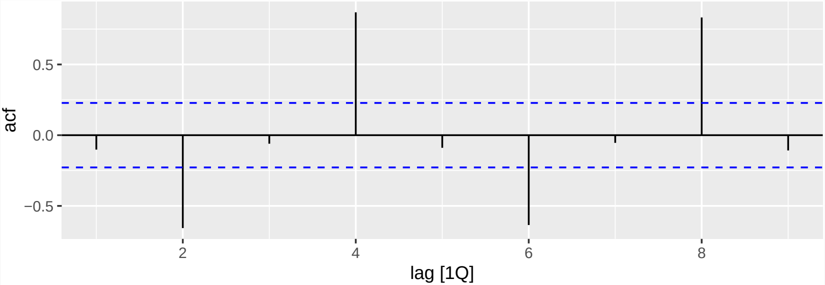
```r
new_production %>% ACF(Beer, lag_max = 9)
```

```
## # 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()
```

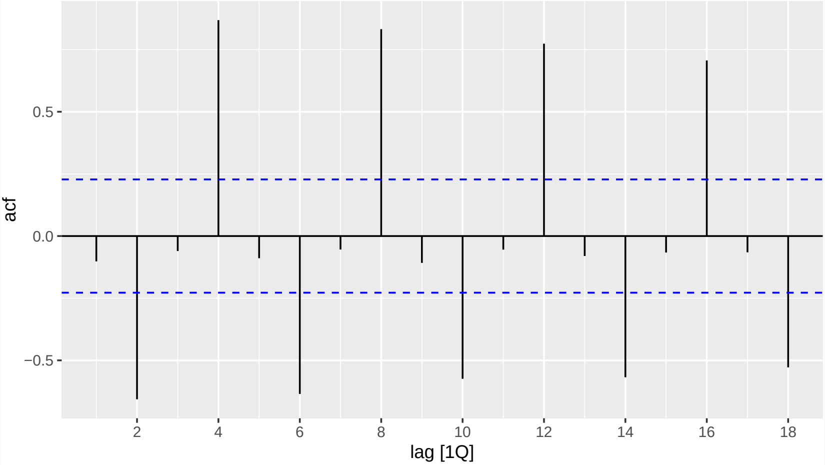


Autocorrelation

- 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, ..., make up the *autocorrelation* or ACF.
- The plot is known as a **correlogram**

ACF

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



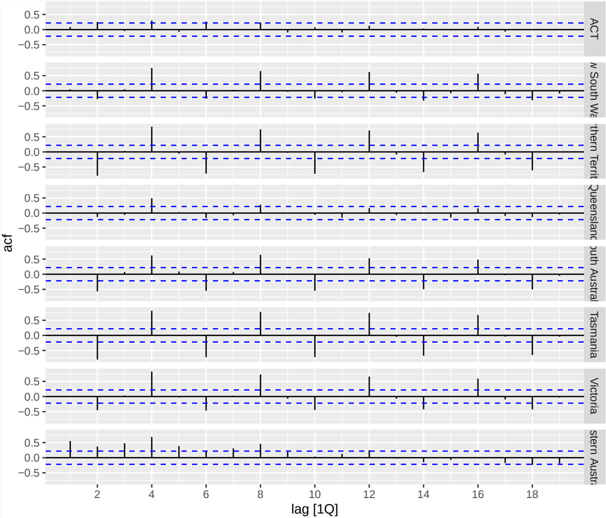
Australian holidays

```
holidays %>% ACF(Trips)
```

```
## # A tsibble: 152 x 3 [1Q]
## # Key:       State [8]
##    State   lag      acf
##    <chr> <lag>    <dbl>
##  1 ACT      1Q  0.0877
##  2 ACT      2Q  0.252
##  3 ACT      3Q -0.0496
##  4 ACT      4Q  0.300
##  5 ACT      5Q -0.0741
##  6 ACT      6Q  0.269
##  7 ACT      7Q -0.00504
##  8 ACT      8Q  0.236
##  9 ACT      9Q -0.0953
## 10 ACT     10Q  0.0750
## # ... with 142 more rows
```

Australian holidays

```r
holidays %>% ACF(Trips) %>% autoplot()
```

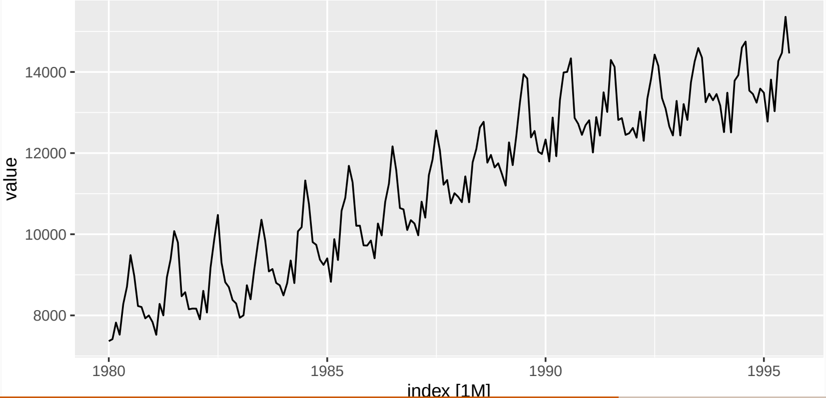


Trend and seasonality in ACF plots

- 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.

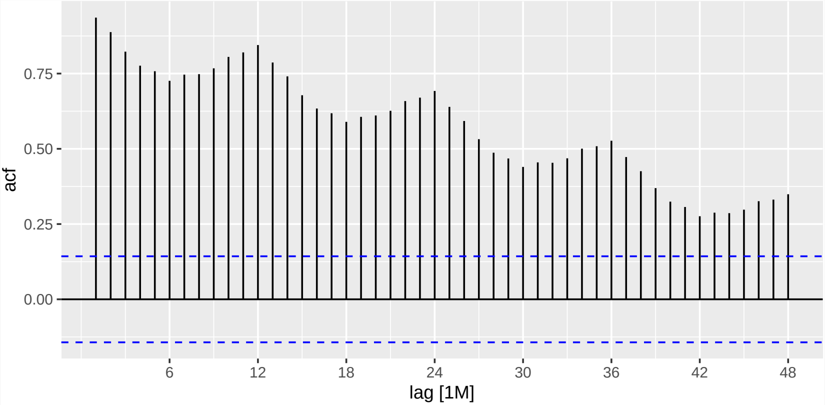
Aus monthly electricity production

```
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.

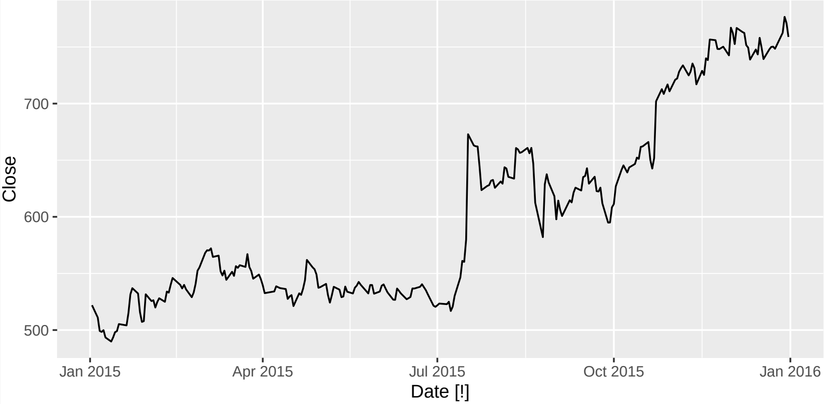
Google stock price

```
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.
```

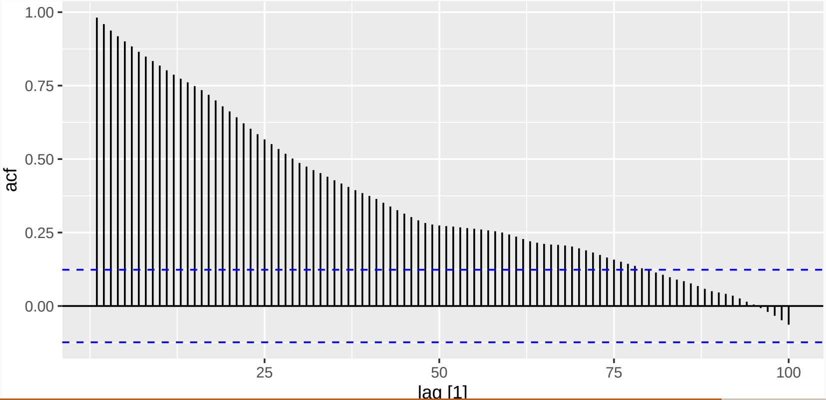
Google stock price

```
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()
```



Outline

Lab Session 4

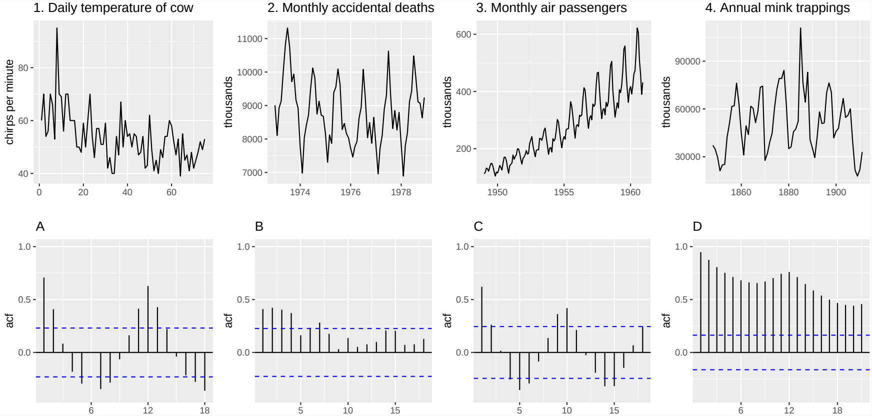
We have introduced the following functions:

- `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`

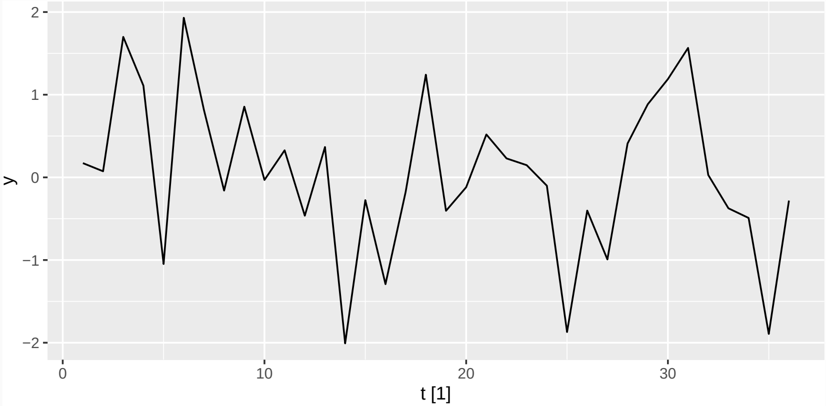
Which is which?



Outline

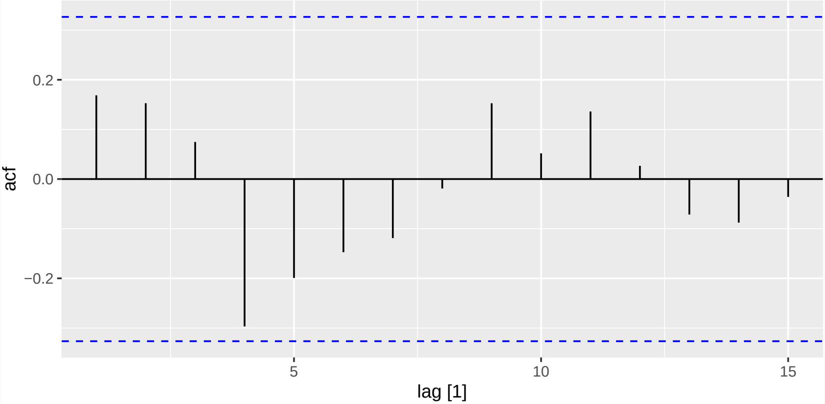
Example: White noise

```
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}
0.169	0.153	0.075	-0.297	-0.199	-0.147	-0.119	-0.019	0.153	0.052



Sampling distribution of autocorrelations

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

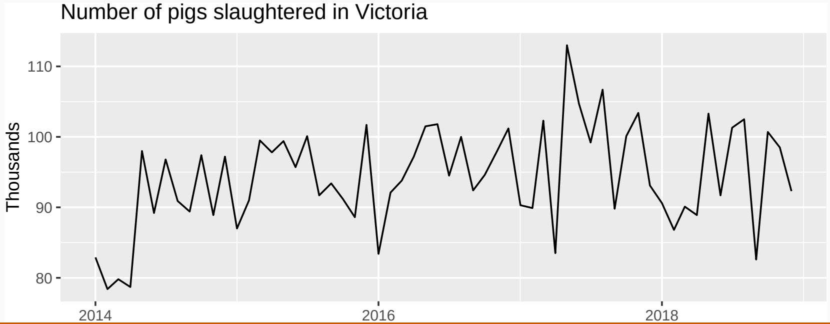
Sampling distribution of autocorrelations

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**.

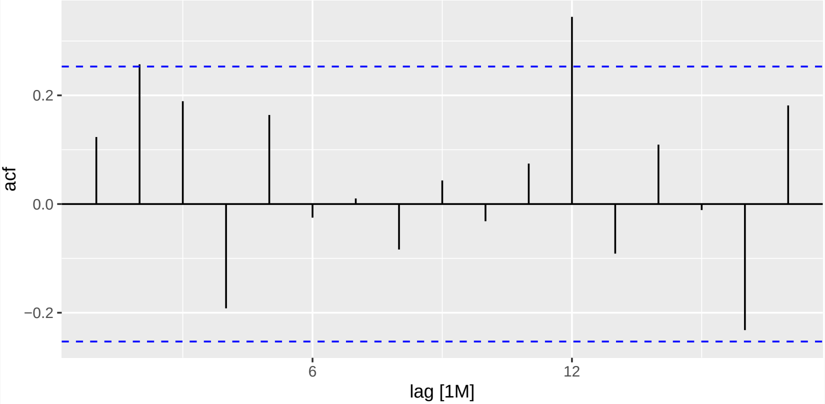
Example: Pigs slaughtered

```
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**.

Outline

Lab Session 5

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?