Chapter 4: Data Visualization

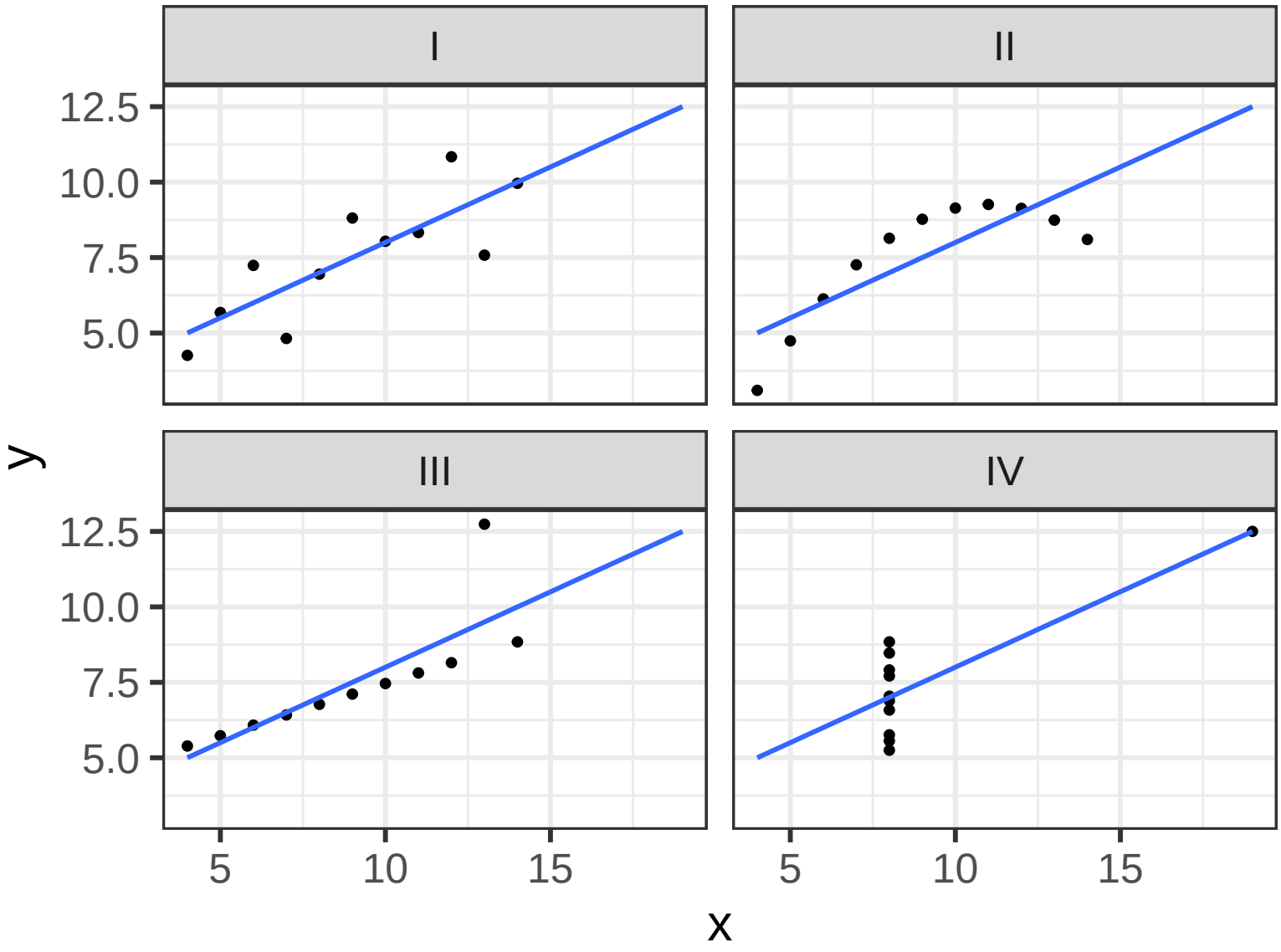
Mark Andrews
Psychology Department, Nottingham Trent University

mark.andrews@ntu.ac.uk

The aim of data visualization

- Rather than being a means to add some eye-candy or ornamentation to otherwise dull reports or slides, the prupose of visualization is to allow us explore data and find patterns that would easily be missed were we to rely only on numerical summary statistics.
- A classic illustration of this *Anscombe's quartet* (Anscombe 1973):

set	mean(x)	mean(y)	sd(x)	sd(y)	cor(x, y)
I	9	7.5	3.32	2.03	0.82
II	9	7.5	3.32	2.03	0.82
III	9	7.5	3.32	2.03	0.82
IV	9	7.5	3.32	2.03	0.82


I
II
III
IV
12.5
10.0
7.5
5.0
5
10
15
y
x

The aim of data visualization

- A key characteristic of data visualization, therefore, is that "it forces us to notice what we never expected to see" (Tukey 1977).
- In other words, data visualization is not simply a means to graphically illustrate what we already know, but to reveal patterns and structures in the data.
- Hartwig and Dearing (1979) state we that we should be guided by principles of *scepticism* and *openness*; we ought to be sceptical to the possibility that any visualization may obscure or misrepresent our data, and we should be open to the possibility of patterns and structures that we were not expecting.

Some guiding principles for visualization

Some guiding principles for visualization mentioned by Edward R. Tufte in his *Visual Display of Quantitative Information* (Tufte 1983) are the following.

- *Above all else show the data*
- *Avoid distorting what the data have to say*
- *Present many numbers in a small space*
- *Encourage the eye to compare different pieces of data*
- *Reveal the data at several levels of detail, from a broad overview to the fine structure*

Some major visualization tools

- *Histograms*, *density plots*, *bar plots*: These are used to display the distribution of values of continuous and discrete variables.
- *Boxplots*: Like histograms and density plots, boxplots (or box-and-whisker plots) display the distribution of values of continuous variables. However, they are more closely tied to robust statistical descriptions and so deserve to be treated as a class onto themselves.
- *Scatterplots*: Scatterplots and their variants such as *bubbleplots* are used to display bivariate data, or the relationships between two variables. Usually, scatterplots are used in cases where both variables are continuous, but may also be used, though perhaps with additional modification, when one variable is discrete.

Histograms

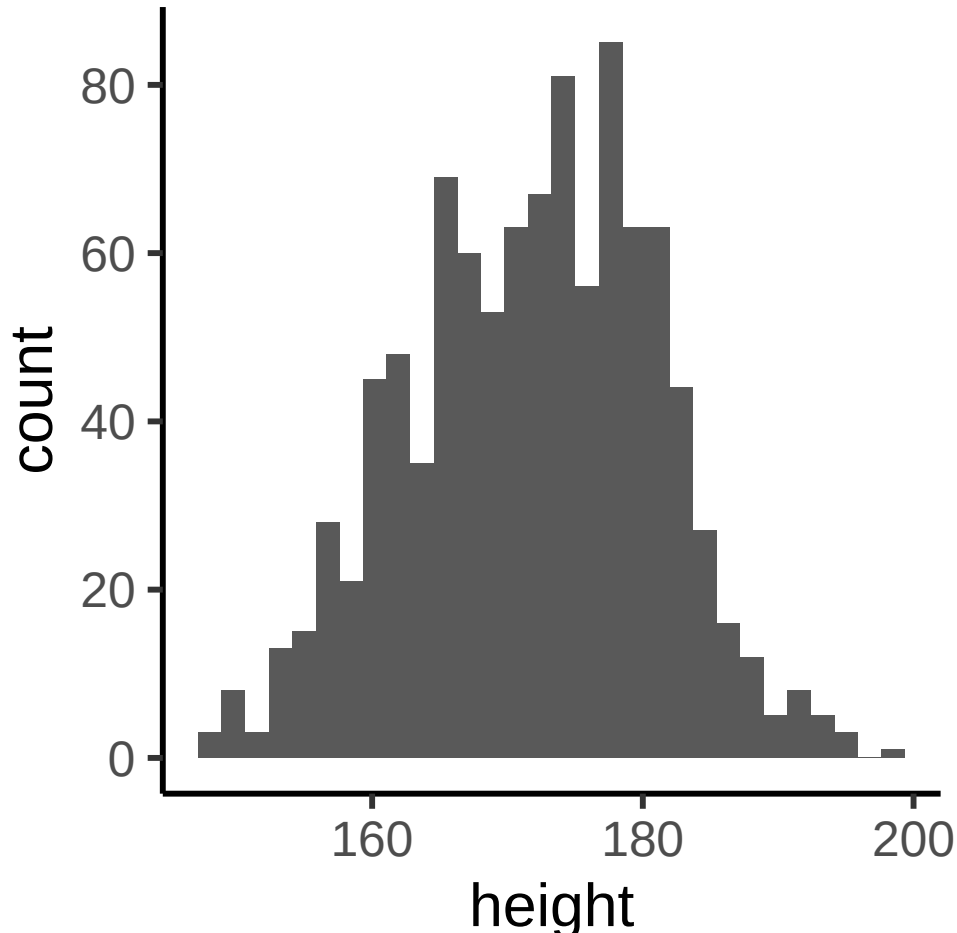
Histograms are one of the simplest and generally useful ways of visualizing distributions of the values of individual variables. To illustrate them, we'll use the `weight` data frame, from which will be downsample to 1000 points.

```
down_sample <- 1000
weight_df <- read_csv("data/weight.csv") %>%
  sample_n(down_sample)
```

Histograms

If we want to display the distribution of the `height` variable, we would proceed as follows.

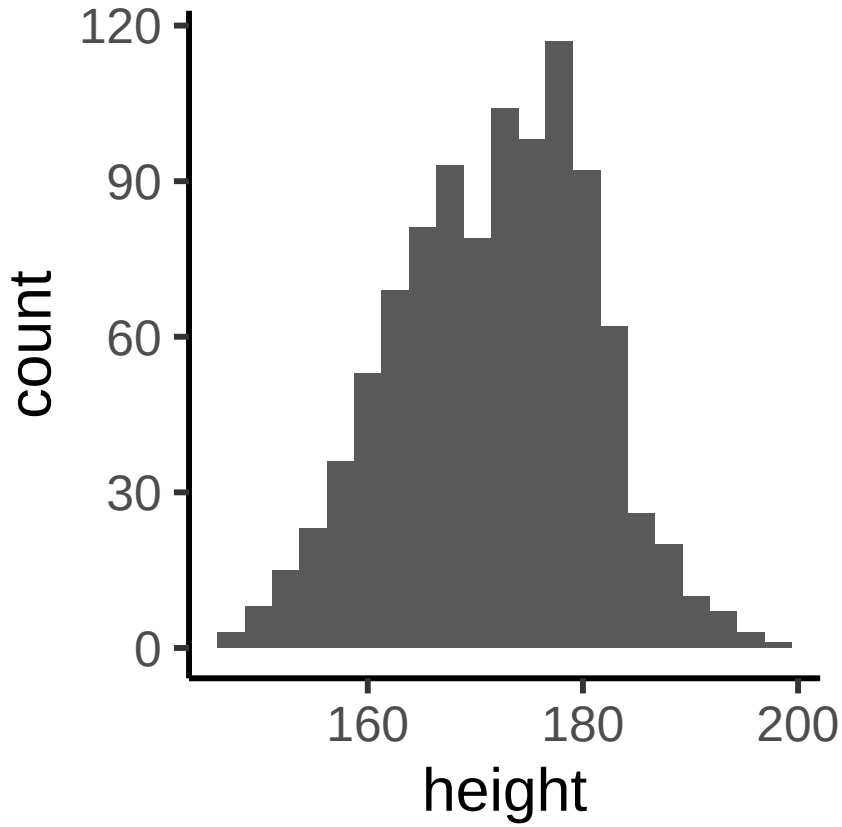
```
ggplot(weight_df,
       mapping = aes(x = height)
) + geom_histogram()
```



Histograms

By default, the histogram will have 30 bins. It is usually good to override this either by specifying another value for `bins`, or by specifying the `binwidth`.

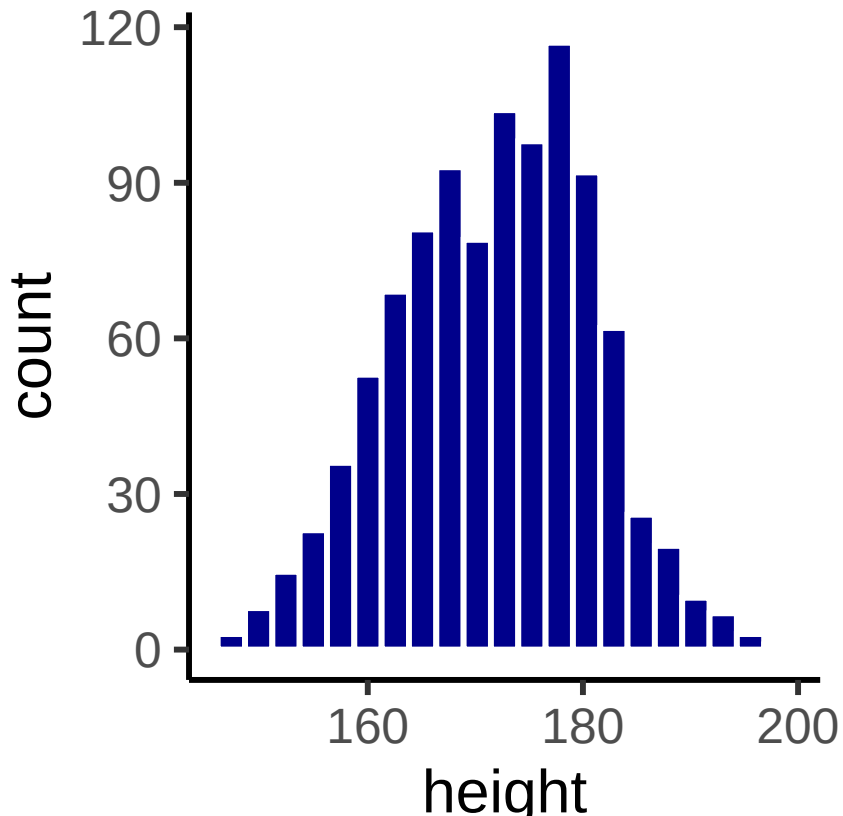
```
ggplot(weight_df,
       mapping = aes(x = height)
) + geom_histogram(binwidth = 2.54)
```



Histograms

Any histogram consists of a set of bars, and each bar has a colour for its interior and another for its border. The interior colour is its `fill` colour, while `colour` specifies the colour of its border.

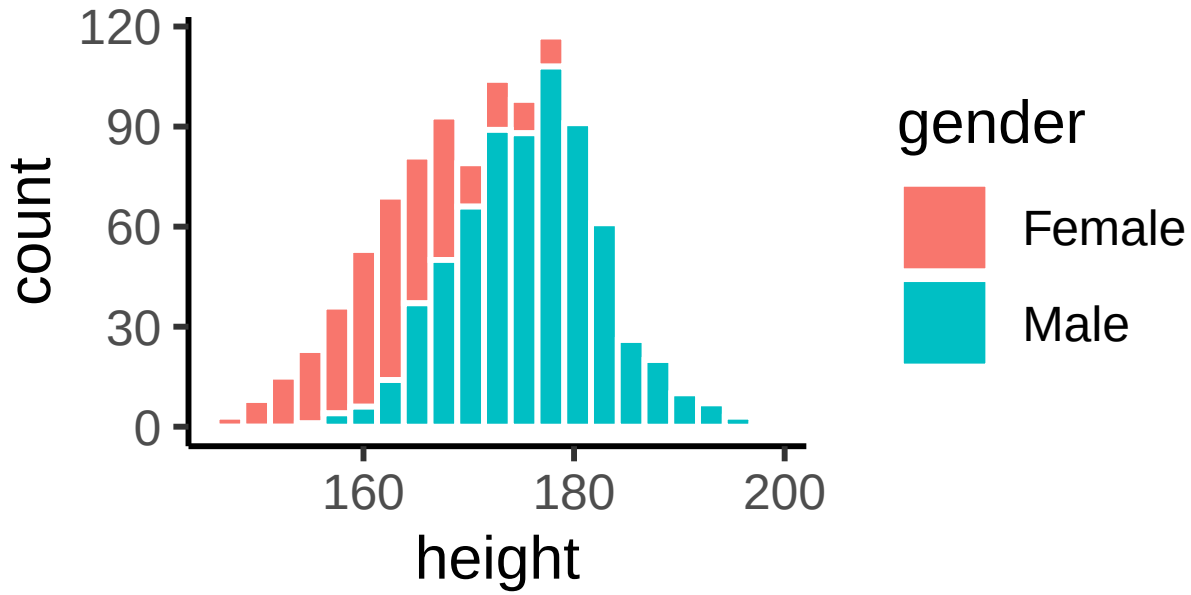
```
ggplot(weight_df,
       mapping = aes(x = height)
) + geom_histogram(binwidth = 2.54, colour = 'white',
                   fill = 'darkblue')
```



Histograms

If, in the `aes` mapping, we specify that either `colour` or `fill`, or both, should be mapped some another variable with discrete values, we obtain a *stacked* histogram. In following example, we set the `fill` values to vary by the `gender` variable.

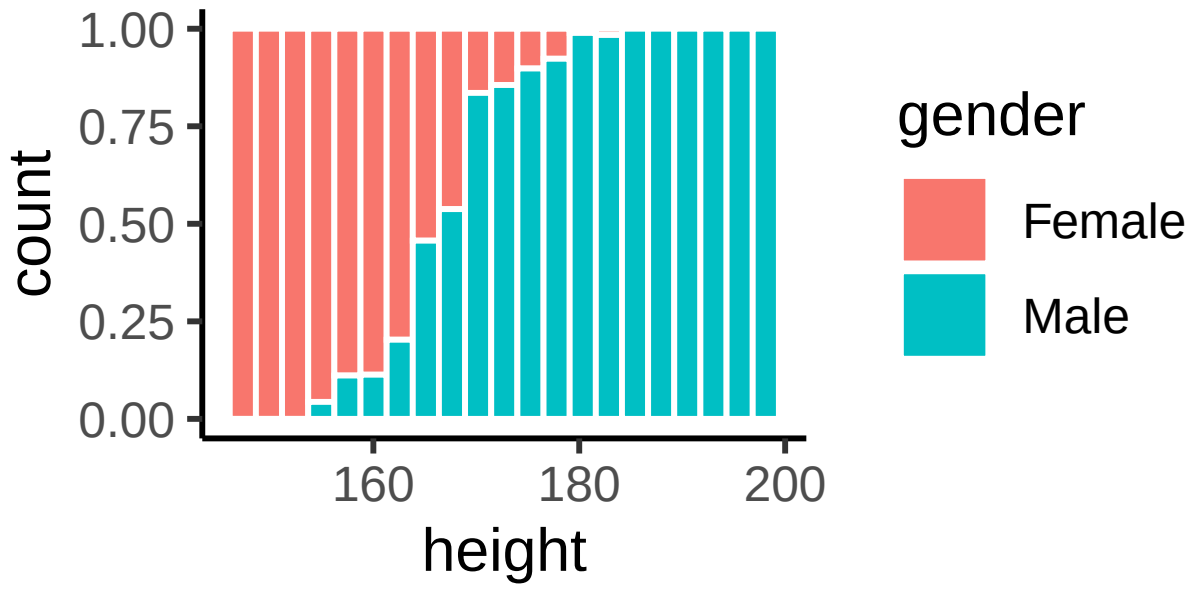
```
ggplot(weight_df,
       mapping = aes(x = height, fill = gender)
) + geom_histogram(binwidth = 2.54, colour = 'white')
```



Histograms

A variant of the stacked histogram above is where each bar occupies 100% of the plot's height so that what is shown is the proportion of the bin's value corresponding to each value of the grouping variable.

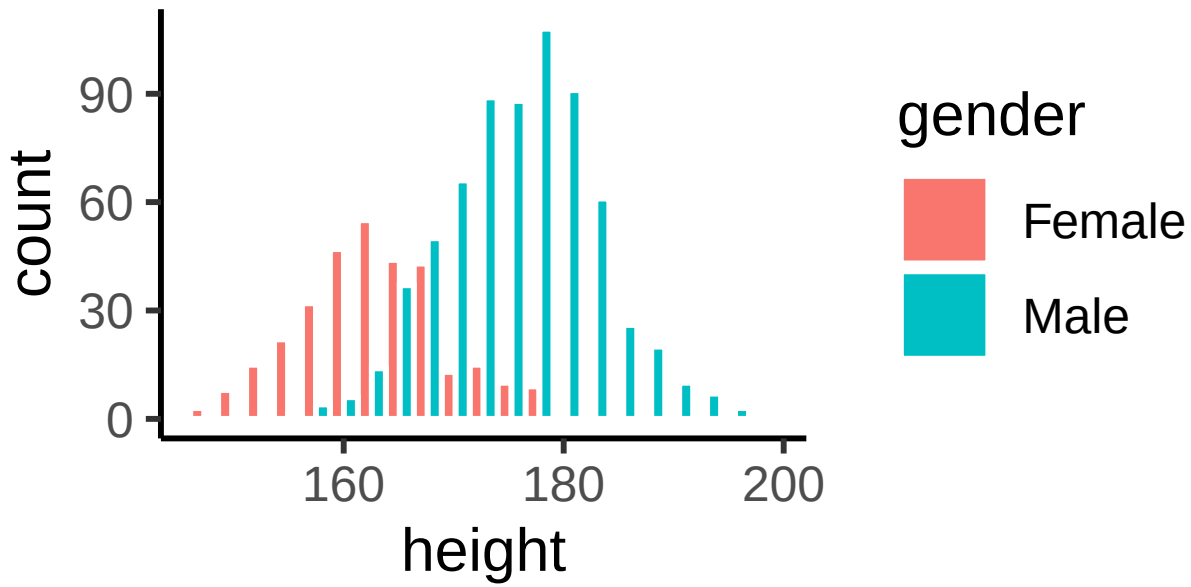
```
ggplot(weight_df,
       mapping = aes(x = height, fill = gender)
) + geom_histogram(binwidth = 2.54, colour = 'white',
                   position = 'fill')
```



Histograms

If we want two separate histograms, one for males and another for females, we can use other options. One option is to specify `position = 'dodge'` within `geom_histogram` as follows.

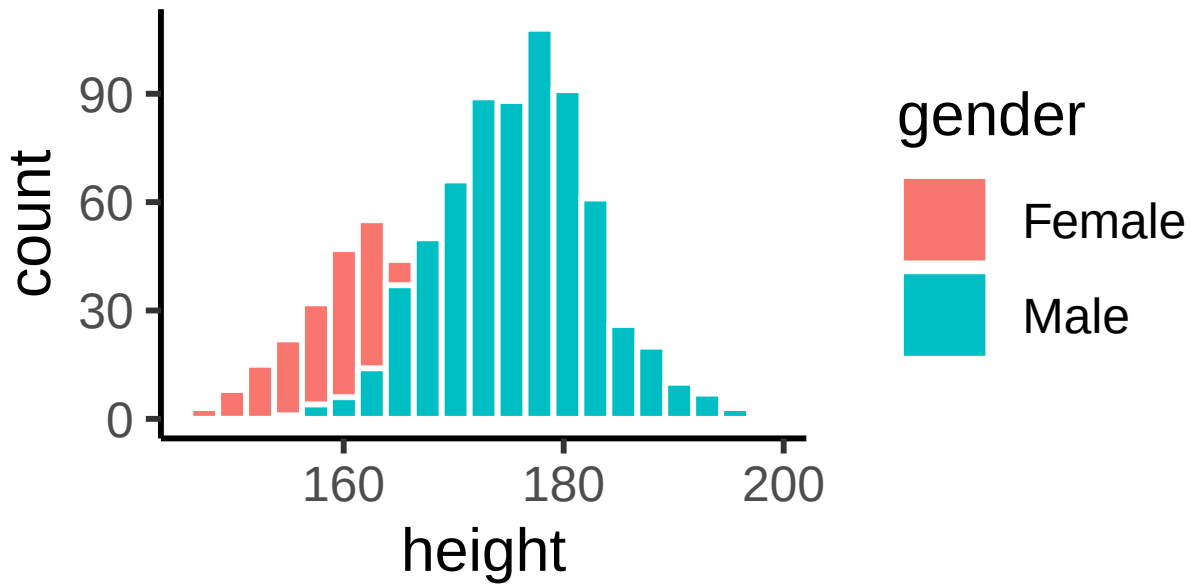
```
ggplot(weight_df,
       mapping = aes(x = height, fill = gender)
) + geom_histogram(binwidth = 2.54, colour = 'white',
                   position = 'dodge')
```



Histograms

An alternative option is to place the bars corresponding to males and females at the exact same location by using `position = 'identity'` within `geom_histogram` as follows.

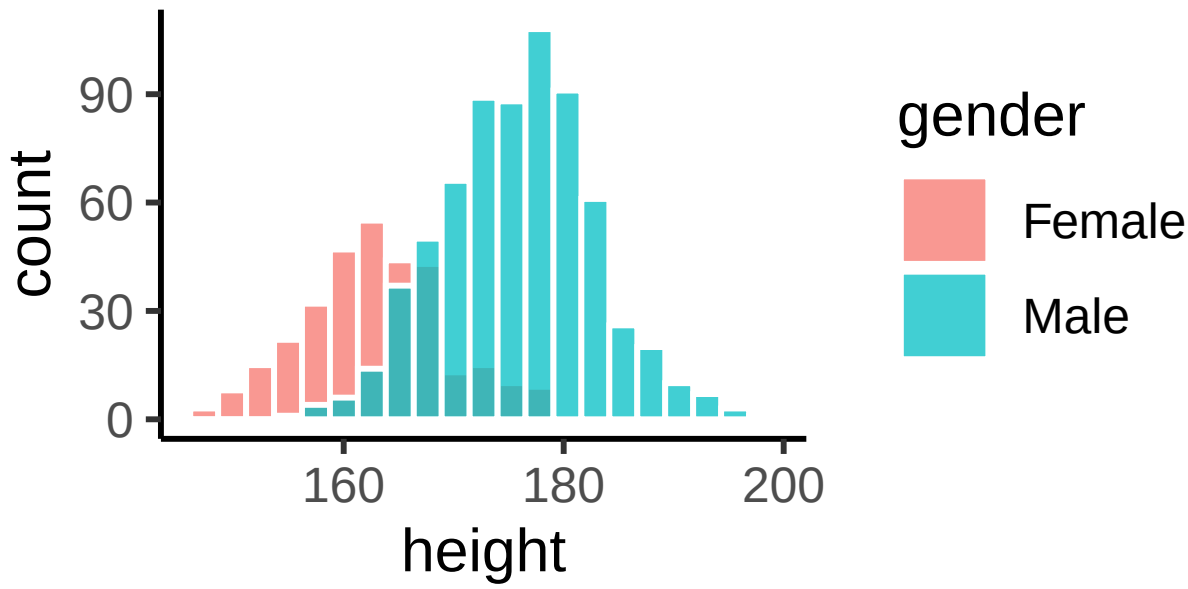
```
ggplot(weight_df,
       mapping = aes(x = height, fill = gender)
) + geom_histogram(binwidth = 2.54, colour = 'white',
                   position = 'identity')
```



Histograms

We can avoid complete occlusion by setting the **alpha**, or opacity, level of the bars to be a value less than 1.0 as in the following example.

```
ggplot(weight_df,
       mapping = aes(x = height, fill = gender)
) + geom_histogram(binwidth = 2.54, colour = 'white',
                   position = 'identity', alpha = 0.75)
```



Tukey boxplots

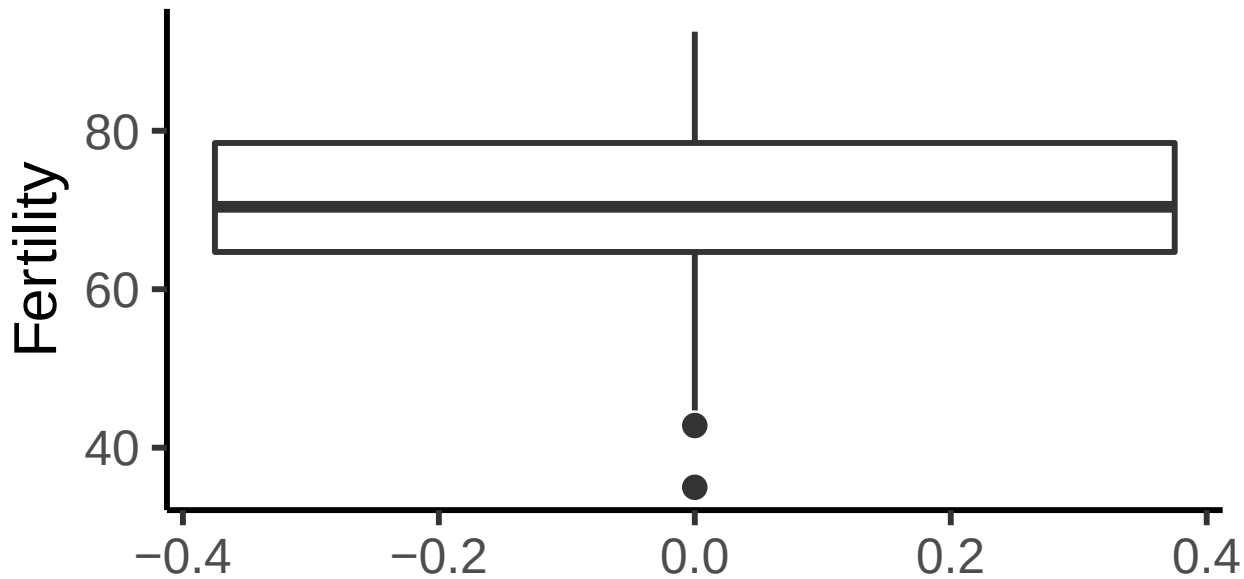
Boxplots, also known as box and whisker plots, are used to display the distribution of values of a variable. One subtype of boxplot is the *Tukey boxplot* (Tukey 1977). These are in fact most common subtype and are the default type implemented in `ggplot2` using the `geom_boxplot` function.

For some of following examples, we'll use the R built-in `swiss` data set used that provides data on fertility rates in 47 Swiss provinces in 1888.

Tukey boxplots

In the following plot, we use a Tukey boxplot to display the distribution of the `Fertility` variable in the `swiss` data set.

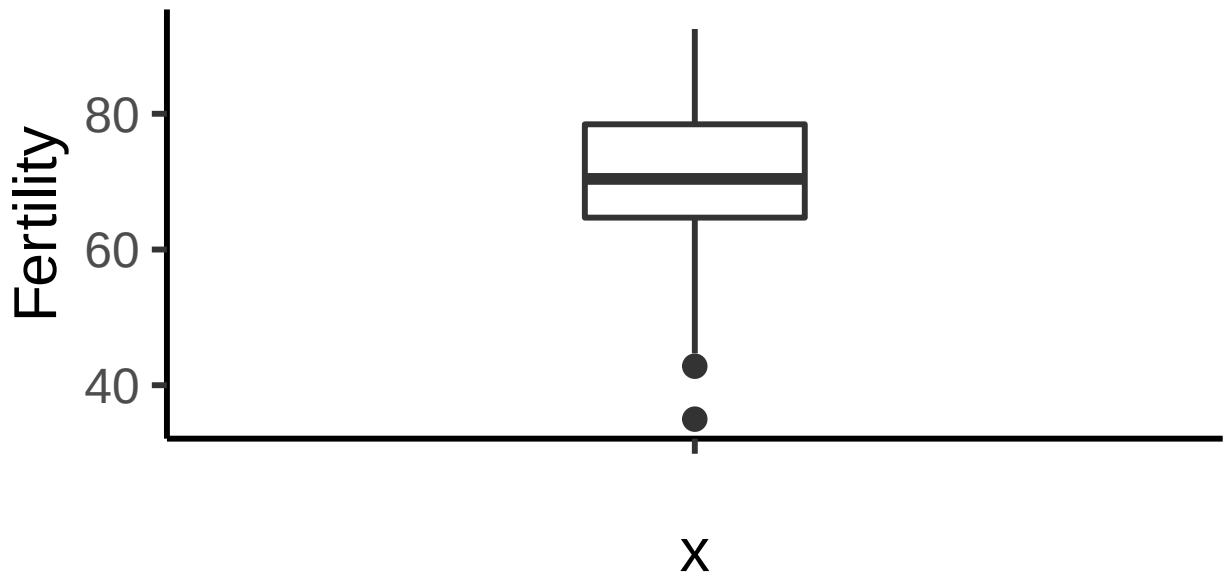
```
ggplot(swiss_df,
       mapping = aes(y = Fertility)
) + geom_boxplot()
```



Tukey boxplots

The default style for a single boxplot can be improved by indicating that the x axis variable is discrete by setting `x = ''` within the `aes` mapping, and then changing the width of the boxplot.

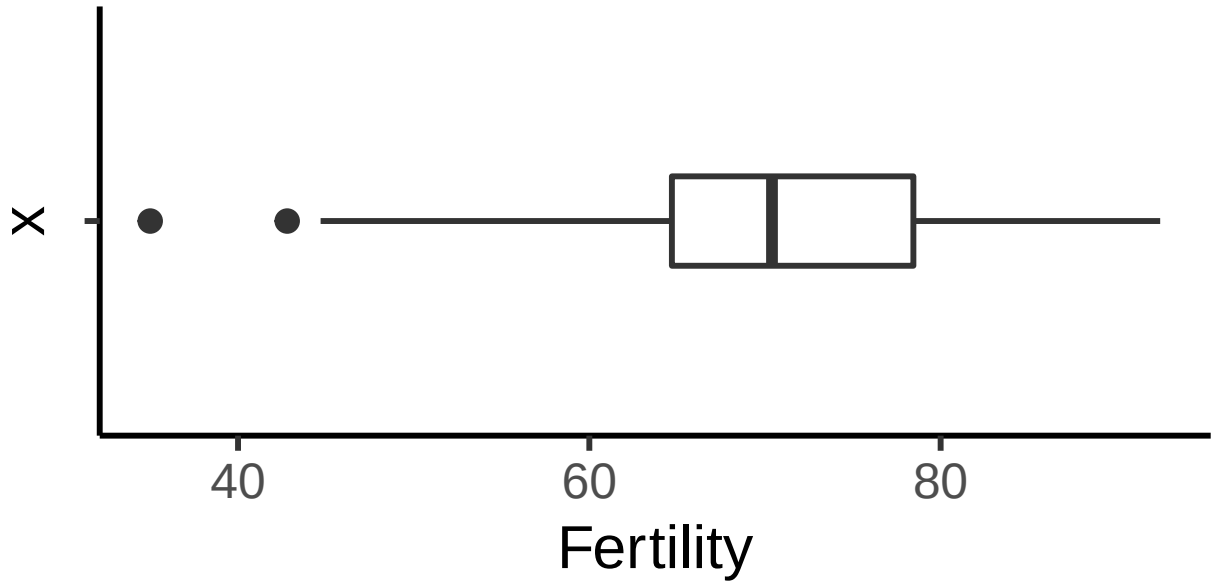
```
ggplot(swiss_df,
       mapping = aes(x = '', y = Fertility)
) + geom_boxplot(width = 0.25)
```



Tukey boxplots

We may convert this vertically extended boxplot to a horizontal one by a `coord_flip()`.

```
ggplot(swiss_df,
       mapping = aes(x = '', y = Fertility)
) + geom_boxplot(width = 0.25) +
  coord_flip()
```



Tukey boxplots

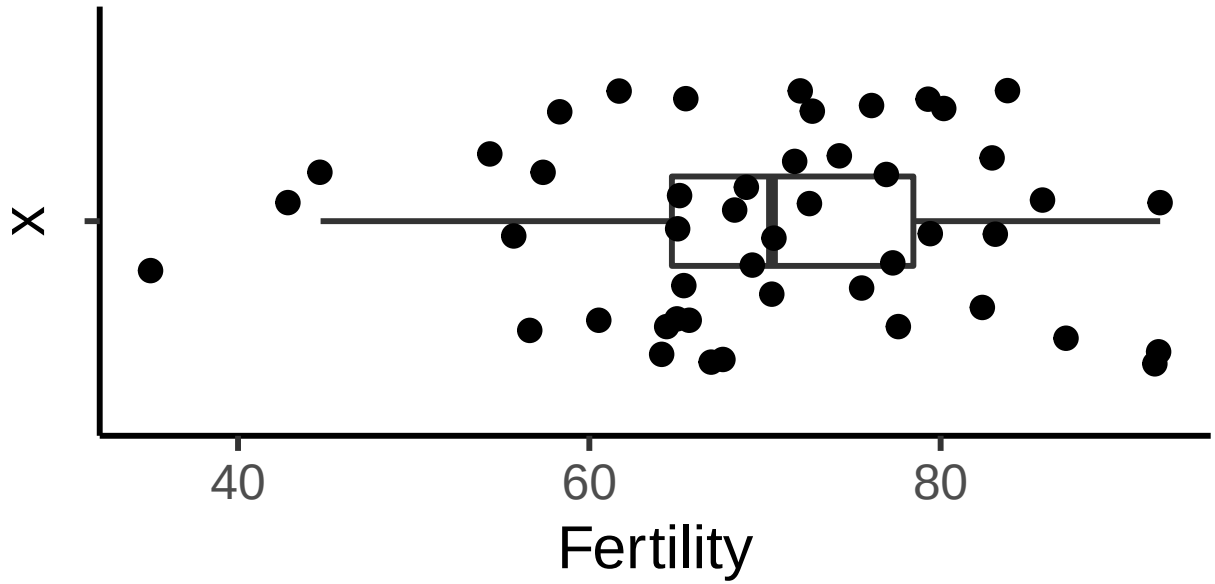
Tukey boxplots are defined as follows:

1. The *box* extends from the 25th to the 75th percentile.
2. The line or band within the box is the median value, which is also the 50th percentile.
3. The *whiskers* extend to the furthest points above the 75th percentile, or below the 25th percentile, that are within 1.5 times the inter-quartile range (the range from the 25th to the 75th percentile).
4. Any points beyond 1.5 times the inter-quartile range above the 75th percentile or below the 25th percentile is represented by a point and is classed as an *outlier*.

Tukey boxplots

It is generally a good idea, therefore, to supplement the boxplot with visualizations of the individual data points. One option for displaying all the data is to provide a *jitter* plot as follows.

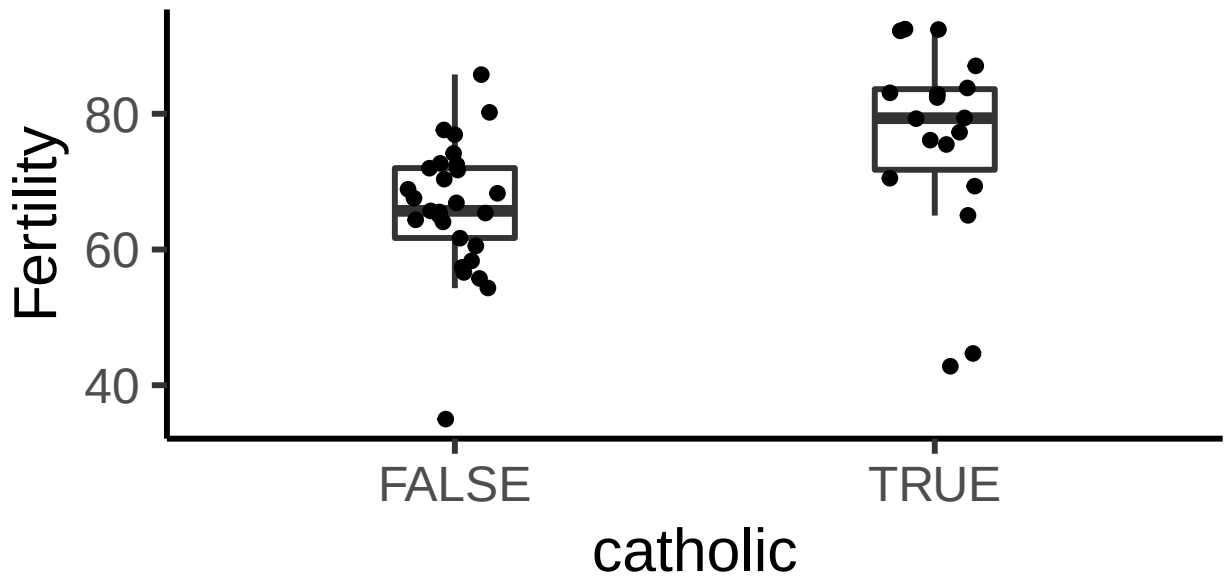
```
ggplot(swiss_df,
       mapping = aes(x = '', y = Fertility)
) + geom_boxplot(width = 0.25, outlier.shape = NA) +
  geom_jitter() +
  coord_flip()
```



Tukey boxplots

By mapping the `x` property to a third variable, we may display multiple box plots side by side.

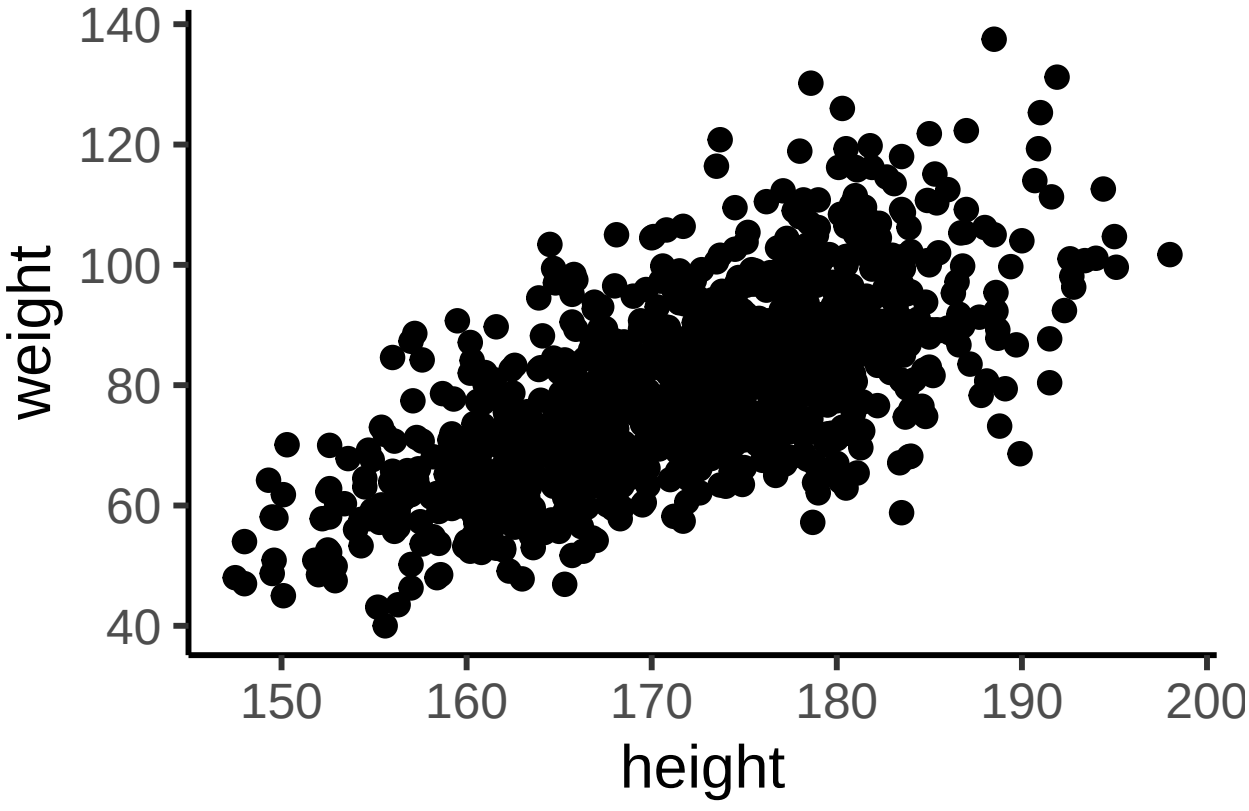
```
ggplot(swiss_df,
       mapping = aes(x = catholic, y = Fertility)
) + geom_boxplot(width = 0.25, outlier.shape = NA) +
  geom_jitter(width = 0.1, size = 0.75)
```



Scatterplots

The following code will display a scatterplot of `weight` as a function of `height`.

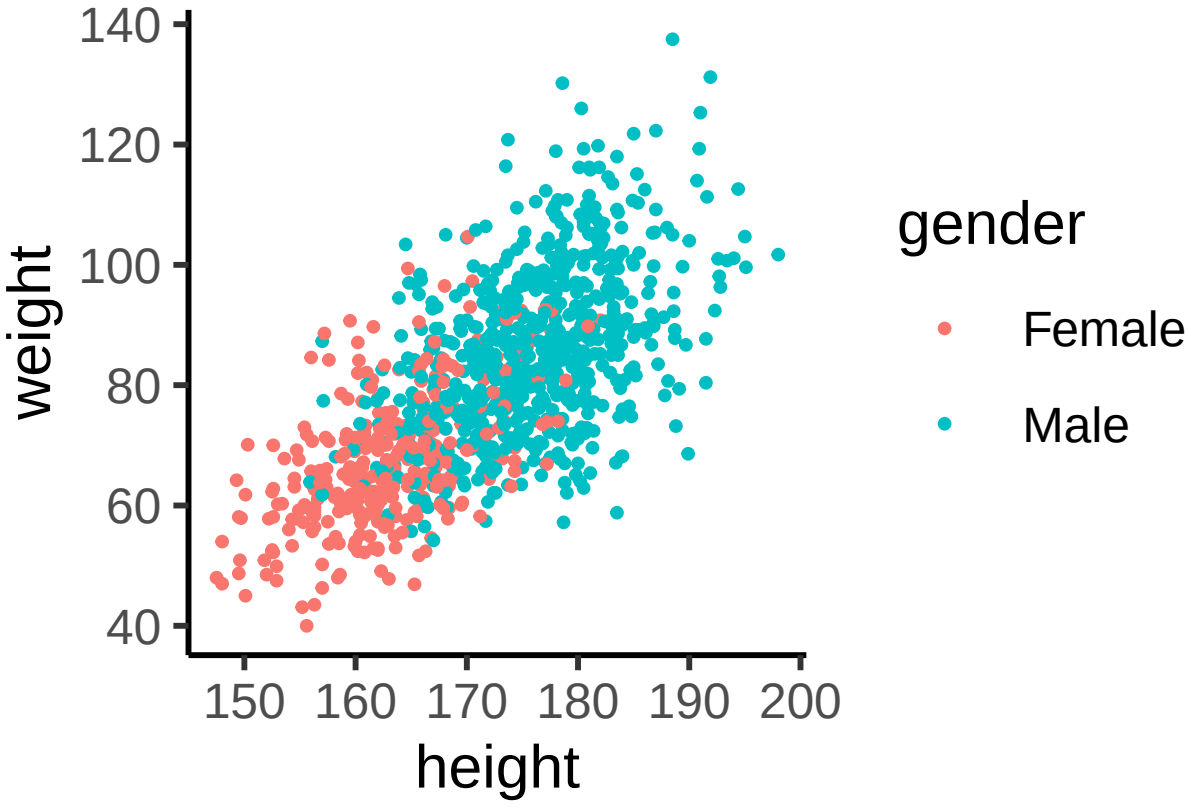
```
ggplot(weight_df,
       mapping = aes(x=height, y=weight)
) + geom_point()
```



Scatterplots

In the following example, we colour code the points according to whether the observation corresponds to a male or a female.

```
ggplot(weight_df,
       mapping = aes(x=height, y=weight, colour = gender)
) + geom_point(size = 0.5)
```



Facet plots

Facet plots allow us produce multiple related subplots, where each subplot displays some subset of the data. For example, in the following plot, we produce one scatterplot with line of bestfit for each one of 18 subjects in an experiment. This data, available in `sleepstudy.csv`, was originally derived from a data set in the package `lme4`.

```
sleepstudy_df <- read_csv("data/sleepstudy.csv")

ggplot(sleepstudy_df,
       mapping = aes(x = Days, y = Reaction, colour = Subject)
) + geom_point() +
  geom_smooth(method = 'lm', se = F) +
  facet_wrap(~Subject) +
  theme_minimal() +
  theme(legend.position = 'none')
```

Facet plots


s308
s309
s310
s330
s331
s332
s333
s334
s335
s337
s349
s350
s351
s352
s369
s370
s371
s372
Reaction
Days
200
300
400
0.0
2.5
5.0
7.5

References

Anscombe, Francis J. 1973. "Graphs in Statistical Analysis." *The American Statistician* 27 (1): 17–21.

Hartwig, Frederick, and Brian E Dearing. 1979. *Exploratory Data Analysis*. Sage.

Tufte, E. R. 1983. *The Visual Display of Quantitative Information*. Graphics Press.

Tukey, J. W. 1977. *Exploratory Data Analysis*. Addison-Wesley Publishing Company.