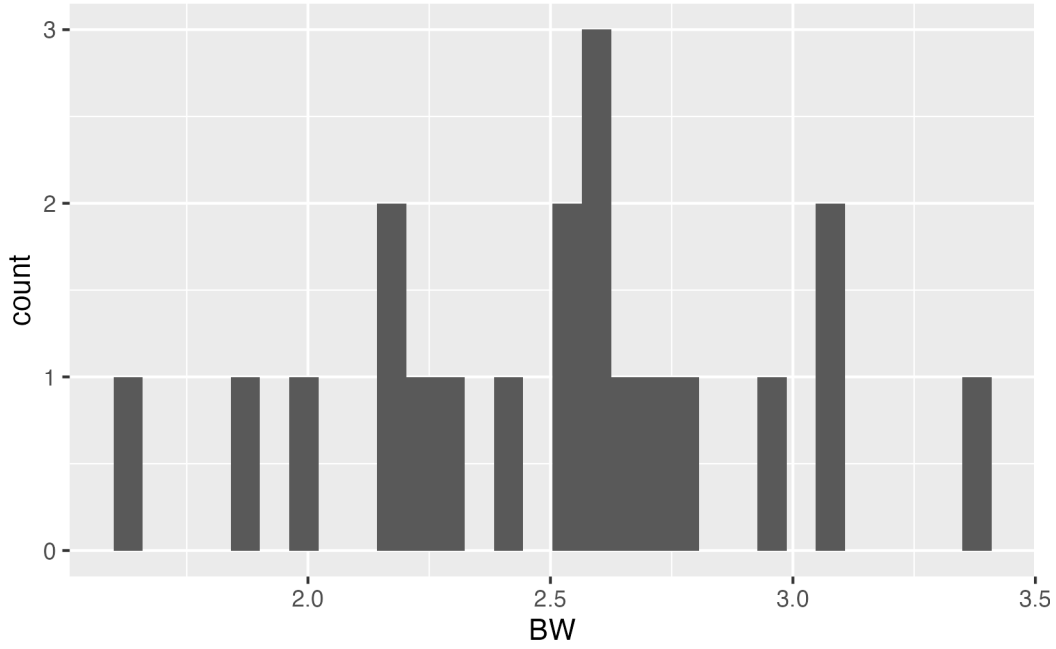
# Histograms in R with `ggplot2`

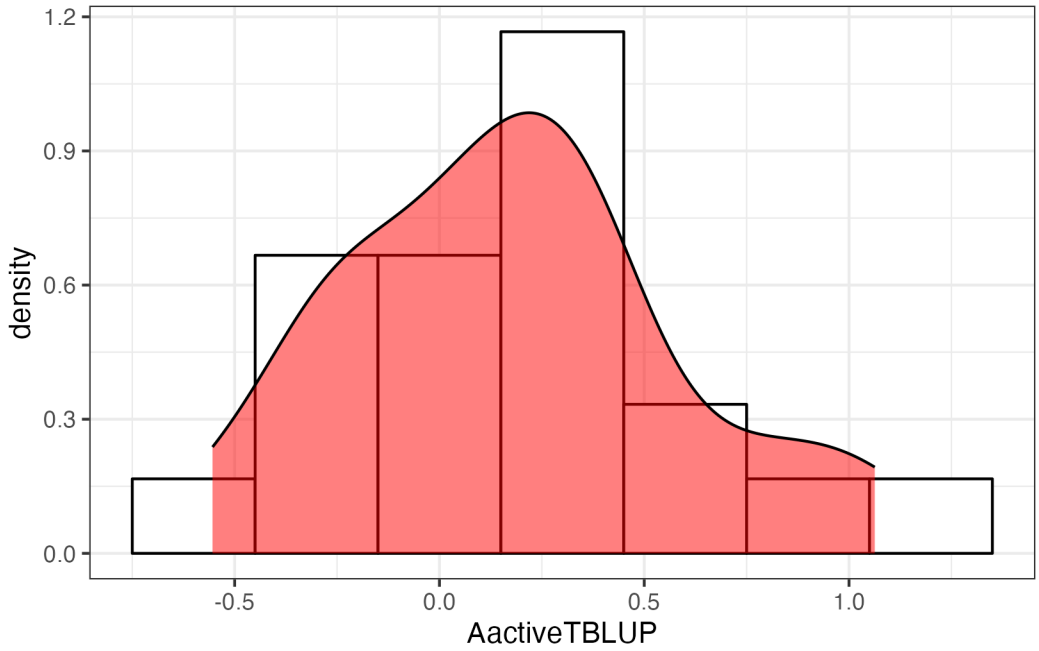
**Cheatsheet**

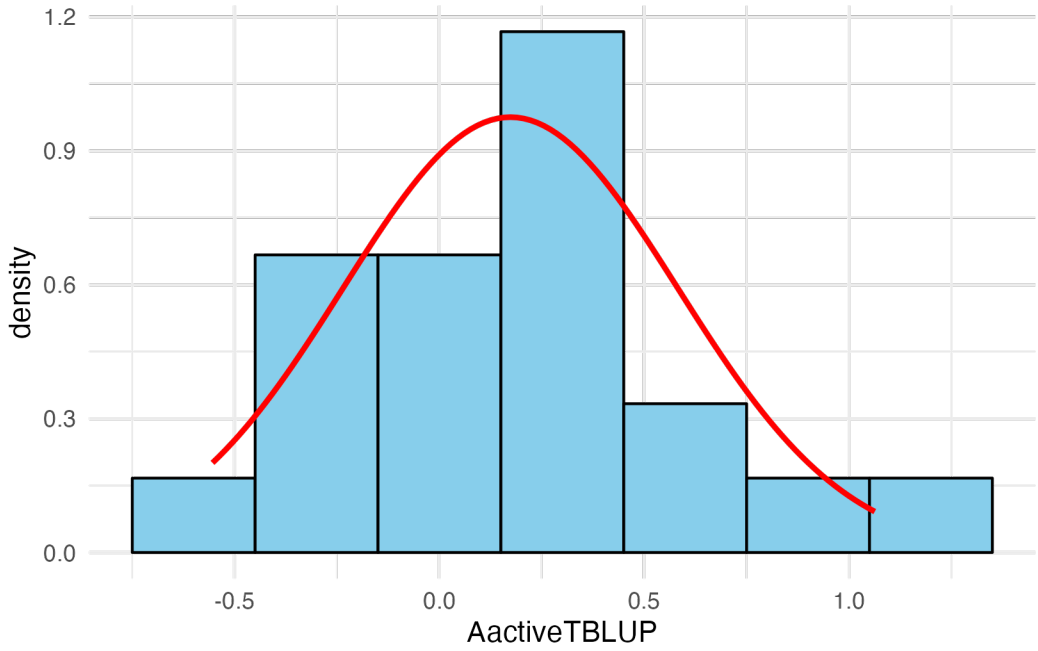
2024-07-27



---




1.2
0.9
0.6
0.3
0.0
density
-0.5
0.0
0.5
1.0
AactiveTBLUP


1.2
0.9
0.6
0.3
0.0
density
-0.5
0.0
0.5
1.0
AactiveTBLUP

## About

The histogram is a bar plot that shows the frequency of (often) continuous values in a dataset. It helps identify patterns, outliers, and the shape of the distribution (skewness, kurtosis).

**Assumed knowledge**

- You know how to install and load packages in R.
- You know how to import data into R.
- You recognise data frames and vectors.

**Data structure**

The data should be in a **long format** (also known as tidy data), where each row is an observation and each column is a variable (Figure 1). If your data is not already structured this way, reshape it manually in a spreadsheet program or in R using the `pivot_longer()` function from the `tidyr` package.

| Sex | BW |
|---|---|
| F | 2.15 |
| M | 2.55 |
| F | 2.95 |
| F | 2.70 |
| M | 2.20 |
| F | 1.85 |
| M | 2.55 |
| M | 2.60 |

| F | M |
|---|---|
| 2.15 | 2.55 |
| 2.95 | 2.20 |
| 2.70 | 2.55 |
| 1.85 | 2.60 |

Figure 1: Data should be in long format (left) where each row is an observation and each column is a variable. This is the preferred format for most statistical software. Wide format (right) is also common, but may require additional steps to analyse or visualise in some instances.

## Data

For this cheatsheet we will use the entire possums dataset used in BIOL2022 labs.

### Import data

```
library(readxl)
possums <- read_excel("possums.xlsx", sheet = 2)
```

### Plot

Use the different plots below to explore the use of histograms in R. Note that histograms only need one variable to be plotted, therefore we pick any one of the several continuous variables in the dataset in the `aes()` function.

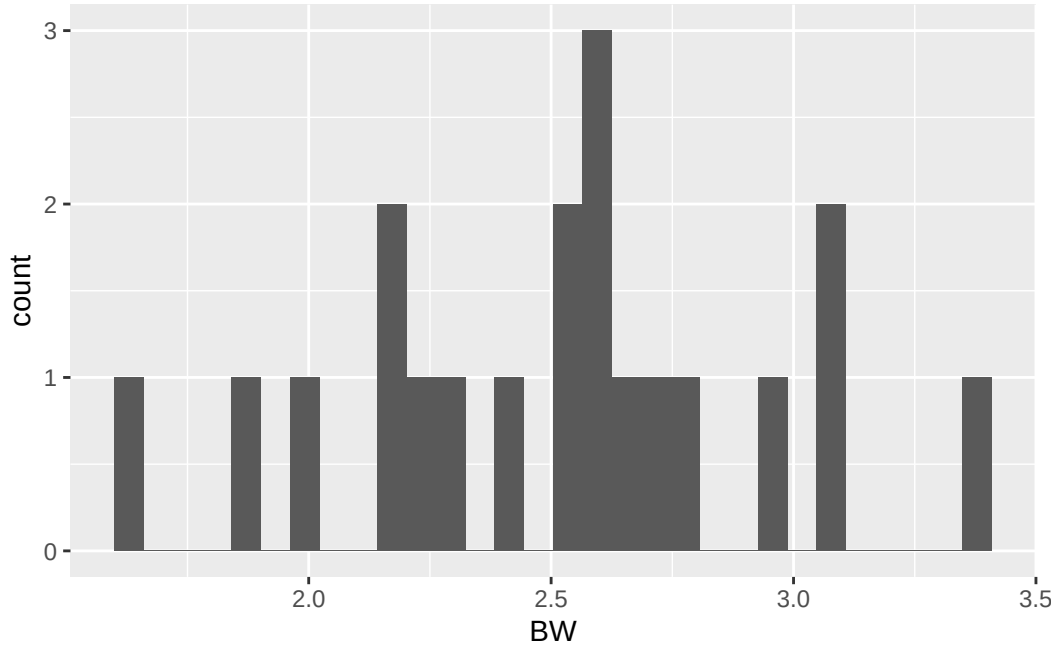
### Version 1

There bare minimum code to create a histogram in R.

```
library(ggplot2)                                    ①
ggplot(possums, aes(x = BW)) +                      ②
  geom_histogram()                                  ③
```

① Load the `ggplot2` package with `library(ggplot2)`.
② Create a canvas using `ggplot()` and specify the variable to be plotted on the x-axis with `aes(x = BW)`.
③ Add a histogram layer using `geom_histogram()`.

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

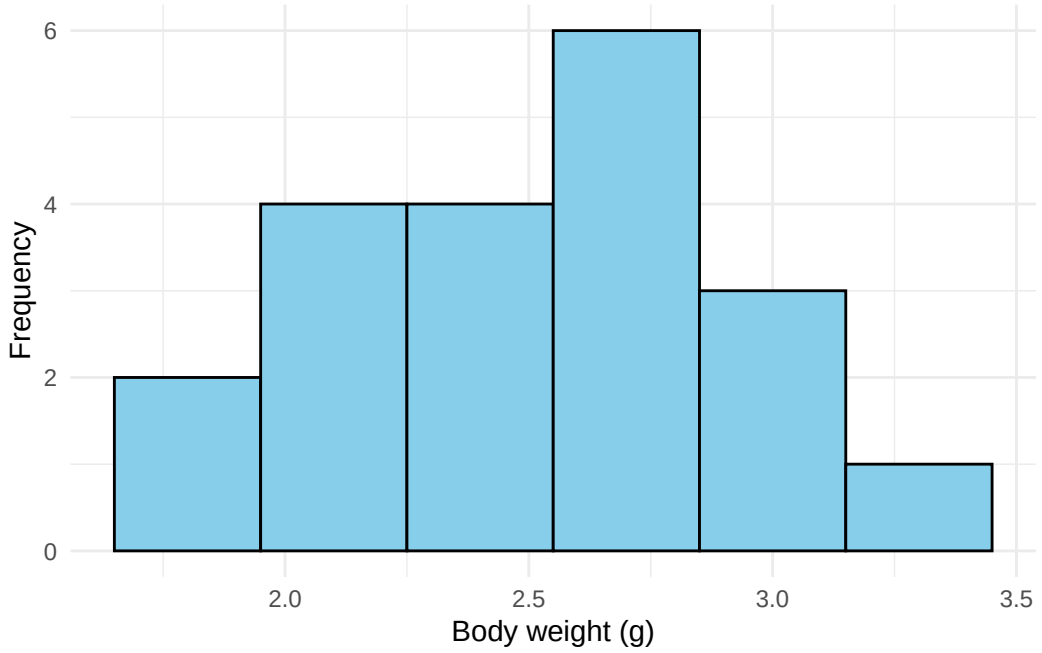



### Version 2

Add colours, labels, adjust bin width, and change the theme.

```
library(ggplot2)
ggplot(possums, aes(x = BW)) +
  geom_histogram(fill = "skyblue", color = "black",                    ①
                 binwidth = .3) +                                      ②
  labs(x = "Body weight (g)",                                          ③
       y = "Frequency") +
  theme_minimal()                                                      ④
```

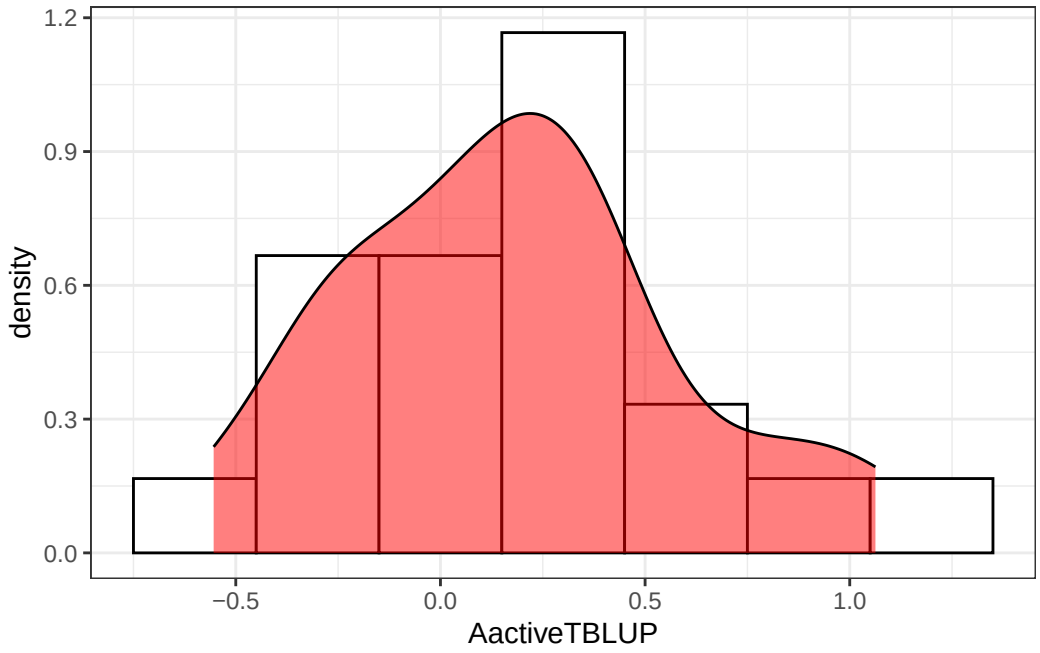
① Use `colour` and `fill` arguments to change bar colors.
② Adjust bin width with `binwidth` to control the detail of the histogram. A smaller value will increase the number of bars based on the range of the data.
③ Add axis labels using `labs()`.
④ Use `theme_minimal()` for a standardized appearance.




### Version 3

Plot both a histogram and a density plot at the same time.

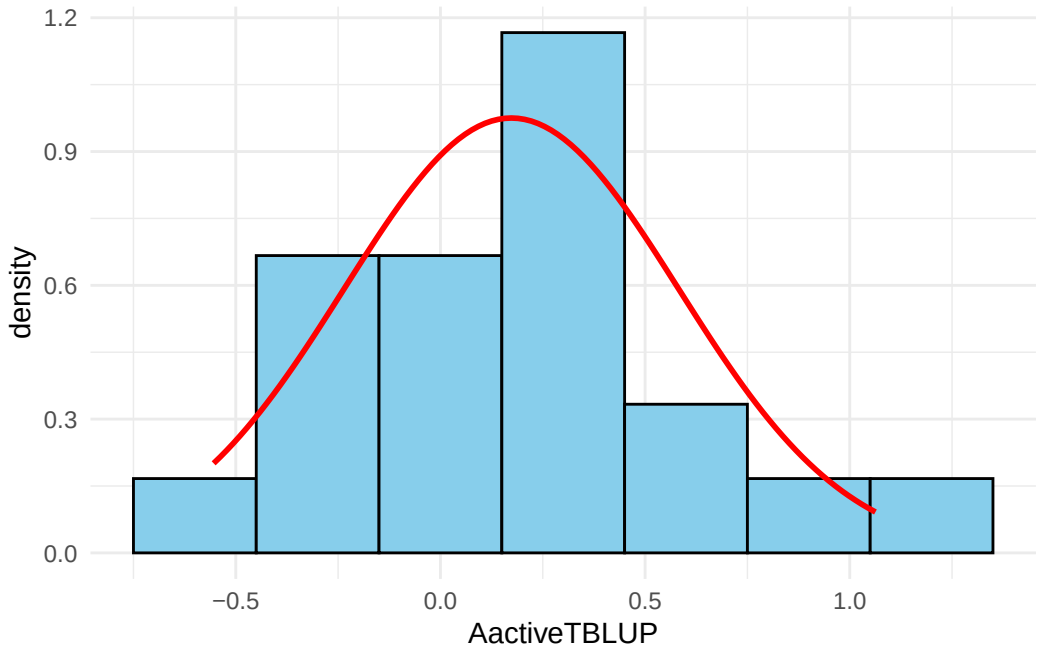
```
library(ggplot2)
p3 <- ggplot(possums, aes(x = AactiveTBLUP,
                     y = after_stat(density))) +
  geom_histogram(
    fill = "white",
    color = "black",
    position = "identity",
    binwidth = 0.3) +
  geom_density(position = "identity", fill = "red", alpha = 0.5) +
  theme_bw()

p3
```




### Version 4

Compare a histogram with a standardised normal distribution.

```
library(ggplot2)
ggplot(possums, aes(x = AactiveTBLUP)) +
  geom_histogram(aes(y = after_stat(density)),
                 fill = "skyblue",
                 color = "black",
                 binwidth = 0.3) +
  stat_function(fun = dnorm,
                args = list(mean = mean(possums$AactiveTBLUP),
                            sd = sd(possums$AactiveTBLUP)),
                color = "red", linewidth = 1) +
  theme_minimal()
```




## Export

Use the `ggsave()` function to save the plot as an image file.

- The `filename` argument specifies the name of the file.
- The `plot` argument specifies the plot to be saved. In this case, the plot is stored in the object `p3` (from Version 3).
- The `width` and `height` arguments specify the dimensions of the plot in inches.

```
ggsave(filename = "histogram.pdf", plot = p3, width = 7, height = 5)
```

The plot will be saved in the working directory, unless you specify a different path in the `filename` argument.