Tutorial-04

Table of contents

Visualisation in R

Overview

Today you will be learning how to create visualisations in R. This will involve adding layers to a plot using the grammar of graphics.

Learning Objectives

- Develop the skills to create plots in R
- Develop your understanding of the grammar of graphics
- Learn about different plot geometries
- Learn about different ways to encode colour
- Practice changing the plot appearance (e.g. theme)

Preparation

- Ensure you installed R and RStudio
- Ensure you have installed the ggplot2 package. You can install this package directly, but this package is also part of the tidyverse package.
- Set yourself an R project for this tutorial.

If you are confused about any of the above, let us know! It's important you don't get left behind here and we want to help you. You may also like to refer back to the Lecture 1 and Tutorial 1 material.

Tasks

Exercise 1

Today you will be creating visualisations in ggplot2 to analyse sporting statistics from the Boston Celtics NBA basketball team.

1. Download the dataset from Moodle, boston_celtics.csv, and place it in a folder called data within the R Project you set up. Then read your data set into R.

We recommend installing the package here to help keep things organised when referencing files.

```r
if(!require(here)){
  install.packages("here")
}
library(here)
```

```r
library(tidyverse)
boston_celtics <- read_csv(here("data", "boston_celtics.csv"))
```

We also recommend following the Reading Data Checklist if you have any problems (see lecture).

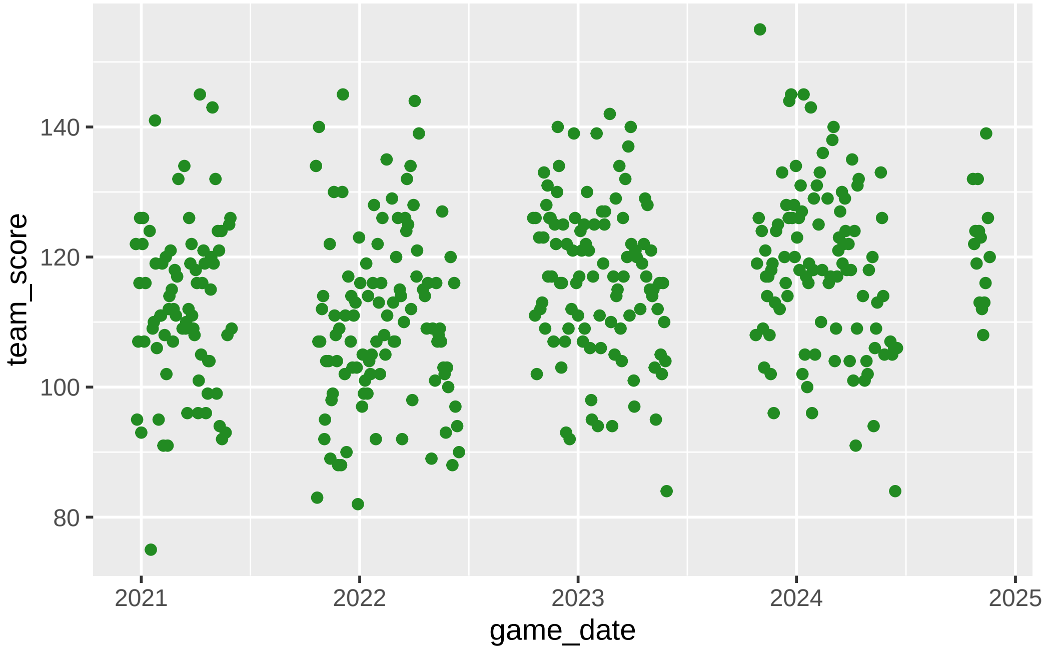
```r
# Check your working directory
getwd()

# Check your data file is where you think it is
file.exists(here("data","boston_celtics.csv"))
```

```r
# Read in your data
library(tidyverse)
boston_celtics <- read_csv(here("data", "boston_celtics.csv"))

# Look at your data
View(boston_celtics)

# Look at a summary
summary(boston_celtics)
```

2. Create a scatter plot that shows the `game_date` and `team_scores`. Make the colour of the points `forestgreen`. Focus on the geometry and aesthetic layer for now.

```r
  ggplot(data = ???) +
    geom_???(aes(x = ???, y = ???), col = ???)
```

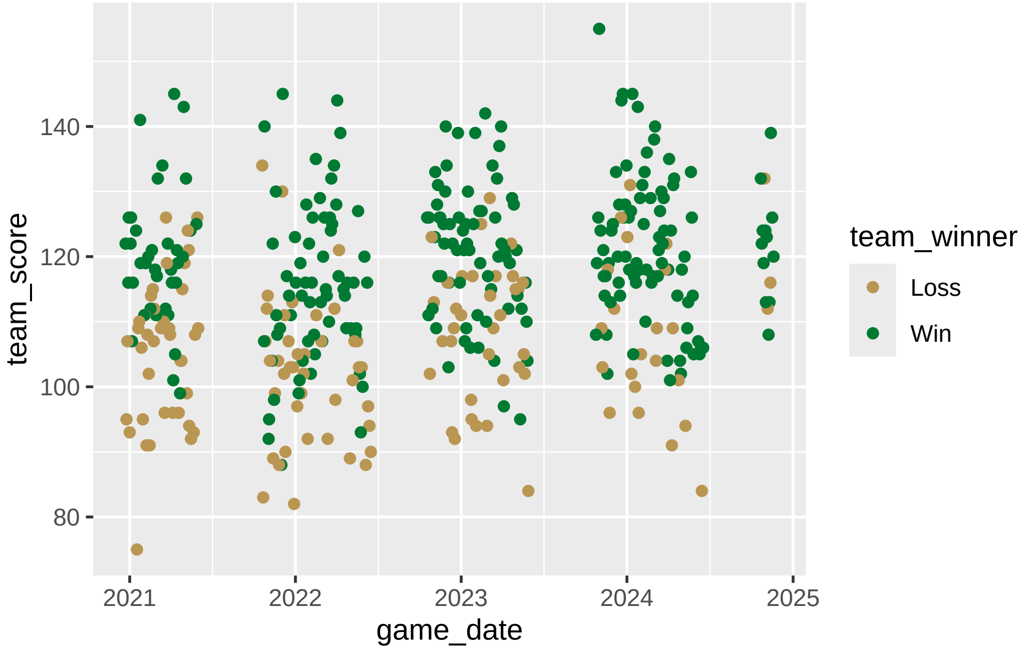
```r
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score), col = "forestgreen")
```



3. Colour the points according to if the team won or lost. Use the hex colour codes from the Boston Celtics team to set the colour scheme. *Notice that the colour command now moves inside the bracket specify aesthetic mappings - Make sure you understand why this is.*

```
    ggplot(data = ???) +
      geom_???(aes(x = ???, y = ???, col = ???))
```

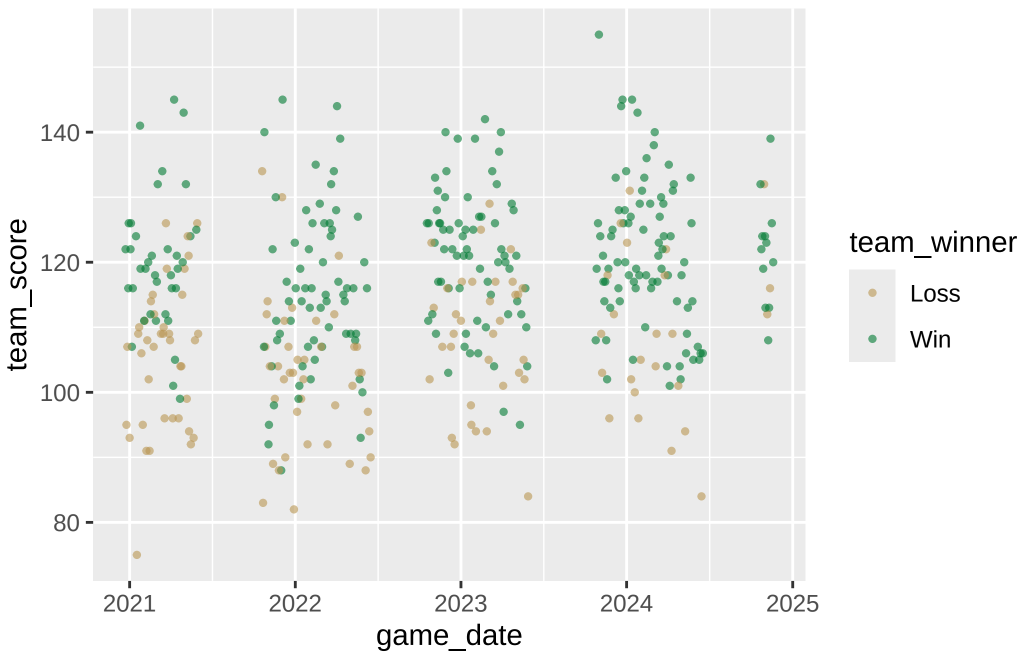
```
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score, col = team_winner)) +
  scale_colour_manual(label = c("Loss", "Win"), values = c("TRUE" = "#007A33", "FALSE" = "#BA
```



4. As there are a lot of points, try changing the size and the transparency to see if that improves the plot.

```
ggplot(data = ???) +
  geom_???(
    aes(x = ???, y = ???, col = ???),
    size = ???, alpha = ???
    )
```

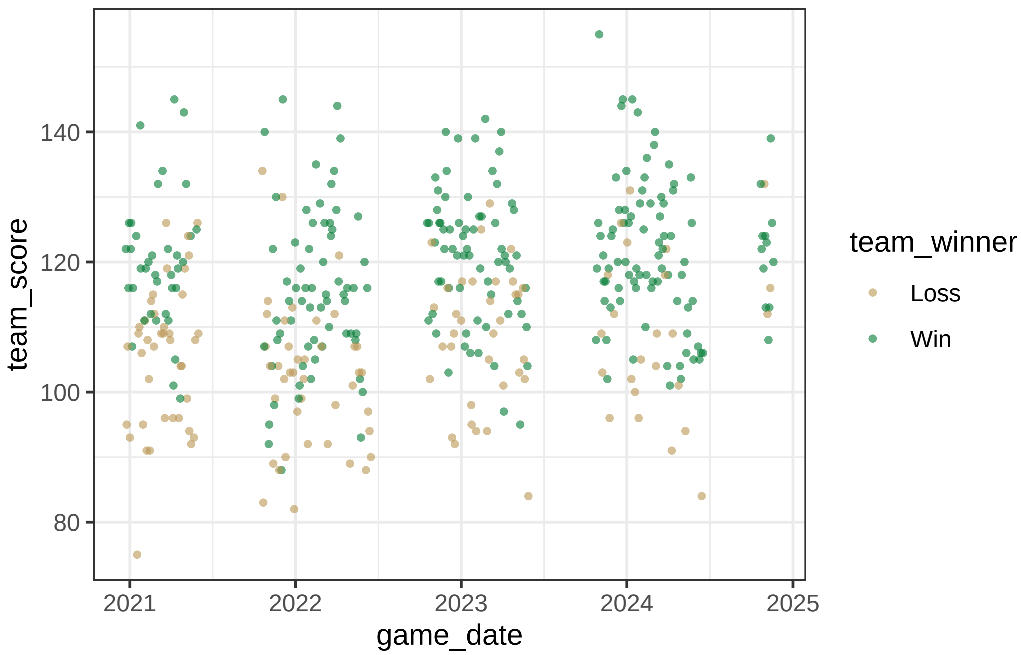
```
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score, col = team_winner), size = 0.8, alpha = 0.6)
  scale_colour_manual(label = c("Loss", "Win"), values = c("TRUE" = "#007A33", "FALSE" = "#BA
```



5. Look at the theme options and try using a few different ones, before deciding on the theme for your plot.

```
ggplot(data = ???) +
  geom_???(
    aes(x = ???, y = ???, col = ???),
    size = ???, alpha = ???
    ) +
  theme_???()
```

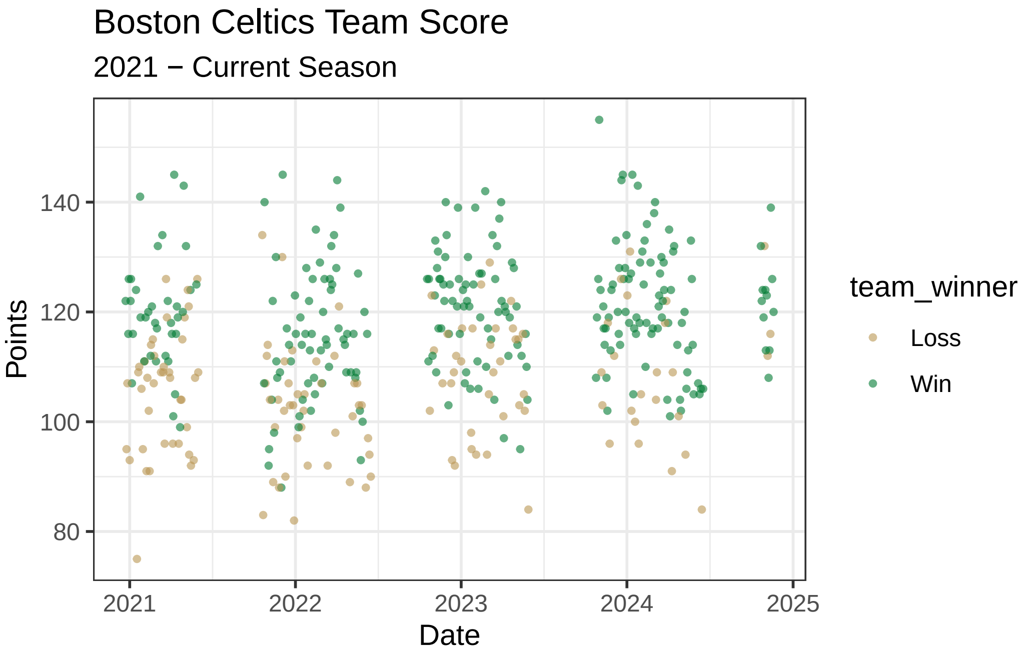
```
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score, col = team_winner), size = 0.8, alpha = 0.6)
  scale_colour_manual(label = c("Loss", "Win"), values = c("TRUE" = "#007A33", "FALSE" = "#BA
  theme_bw()
```



5. Add appropriate labels to your plot.

```
ggplot(data = ???) +
  geom_???(
    aes(x = ???, y = ???, col = ???),
    size = ???, alpha = ???
    ) +
  theme_???() +
  labs(title = ???, subtitle = ???, x = ???, y = ???)
```

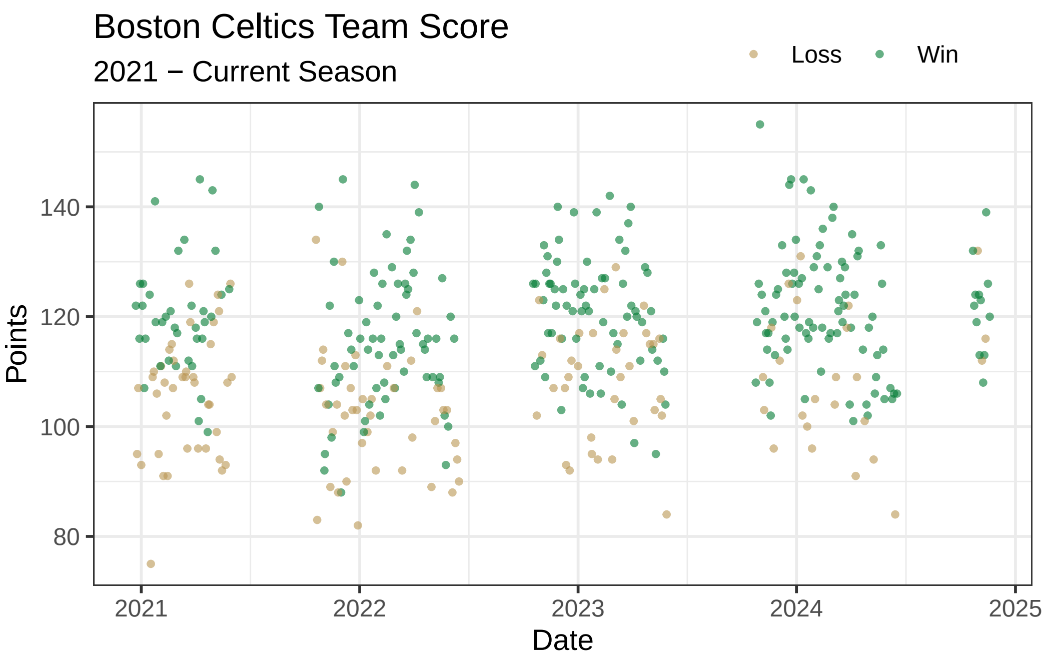
```
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score, col = team_winner), size = 0.8, alpha = 0.6)
  scale_colour_manual(label = c("Loss", "Win"), values = c("TRUE" = "#007A33", "FALSE" = "#B
  theme_bw() +
  labs(title = "Boston Celtics Team Score",
       subtitle = "2021 - Current Season",
       x = "Date",
       y = "Points")
```



6. Move the legend on the plot to increase the data-density. *(Note the solutions show a more advanced way to move the legend than the lecture notes.)*

```
ggplot(data = ???) +
  geom_???(
    aes(x = ???, y = ???, col = ???),
    size = ???, alpha = ???
    ) +
  theme_???() +
  labs(title = ???, subtitle = ???, x = ???, y = ???)
  theme(legend.position = ???)
```

```
ggplot(data = boston_celtics) +
  geom_point(aes(x = game_date, y = team_score, col = team_winner), size = 0.8, alpha = 0.6)
  scale_colour_manual(label = c("Loss", "Win"), values = c("TRUE" = "#007A33", "FALSE" = "#BA
  theme_bw() +
  labs(title = "Boston Celtics Team Score",
       subtitle = "2021 - Current Season",
       x = "Date",
       y = "Points") +
  theme(legend.position = c(0.8, 1.1), legend.title = element_blank(), legend.direction = "ho
```

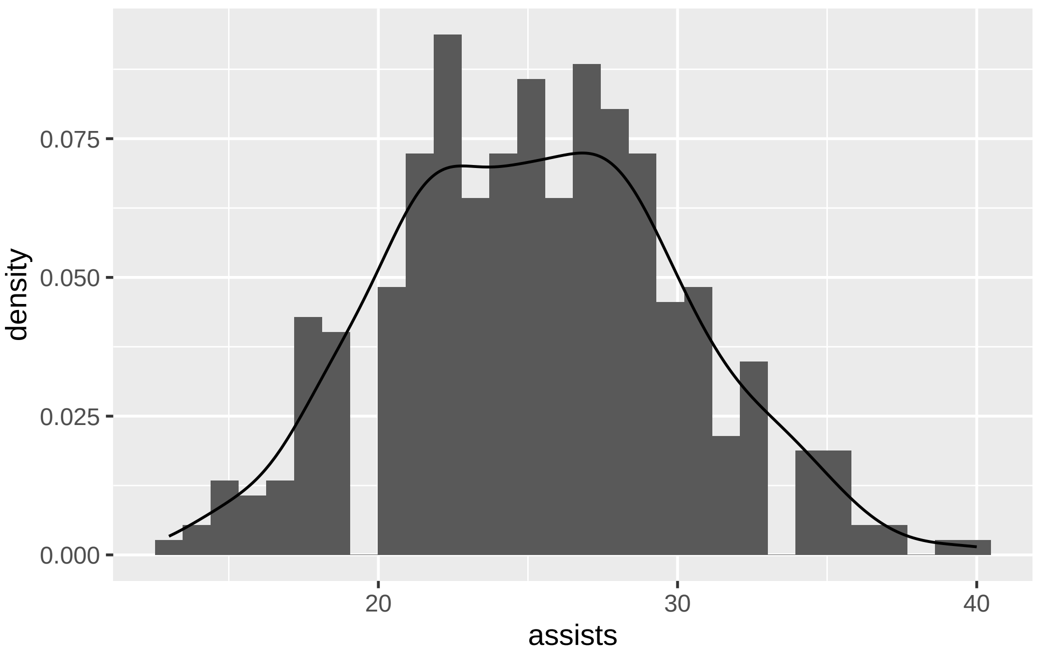


Exercise 2

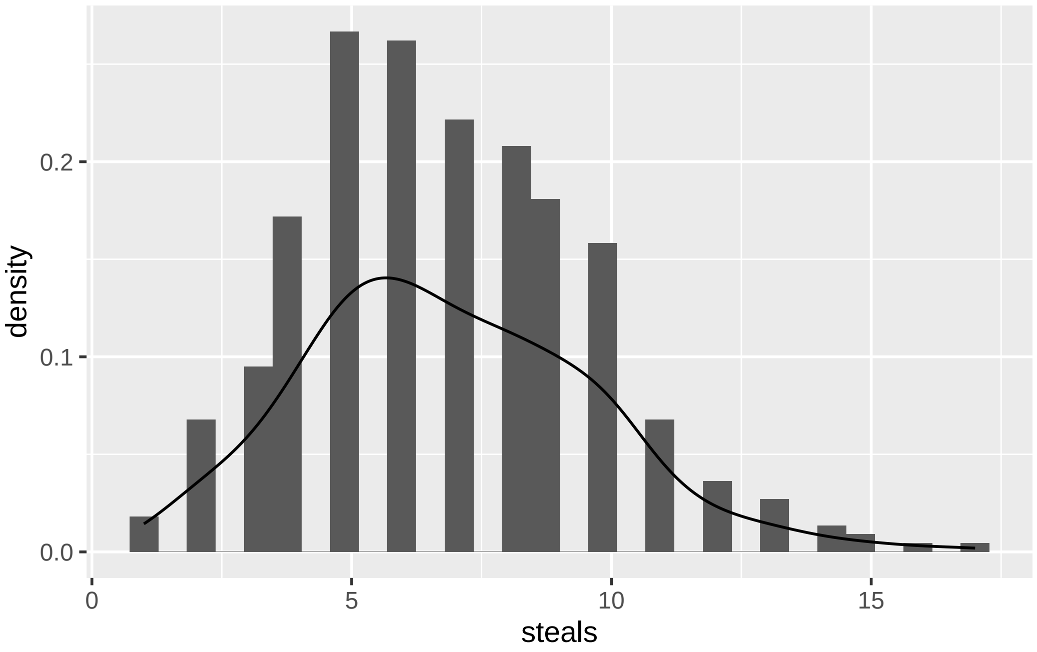
Now we've mastered the basic layers, take some time to creating some other visualisations that are interesting to you!

- You may like to produce plots looking at the distribution of some of the key variables. For example, `assists`, `rebounds`, `steals` etc.

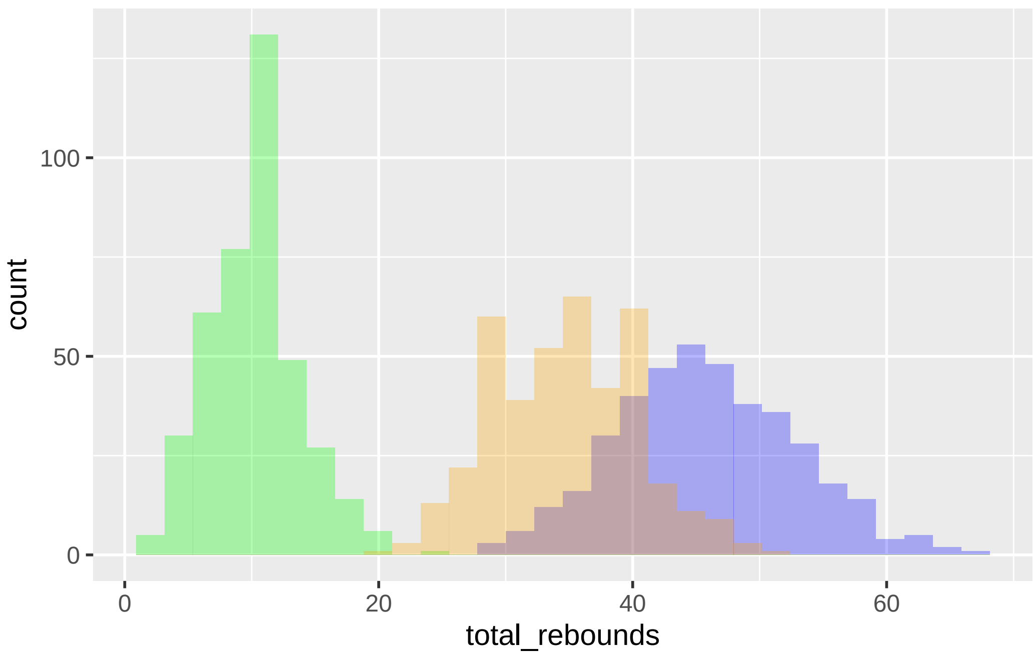
```
boston_celtics |>
  ggplot(aes(x = assists)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density()
```



```
boston_celtics |>
  ggplot(aes(x = steals)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density()
```

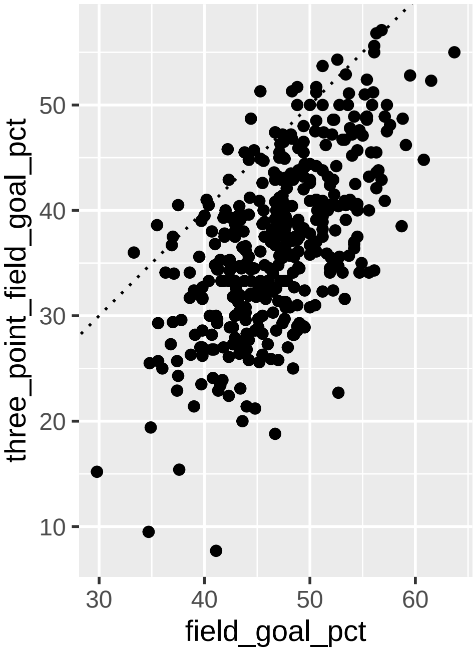


```
boston_celtics |>
  ggplot() +
  geom_histogram(aes(x = total_rebounds), fill = "blue", alpha = 0.3) +
  geom_histogram(aes(x = offensive_rebounds), fill = "green", alpha = 0.3) +
  geom_histogram(aes(x = defensive_rebounds), fill = "orange", alpha = 0.3)
```



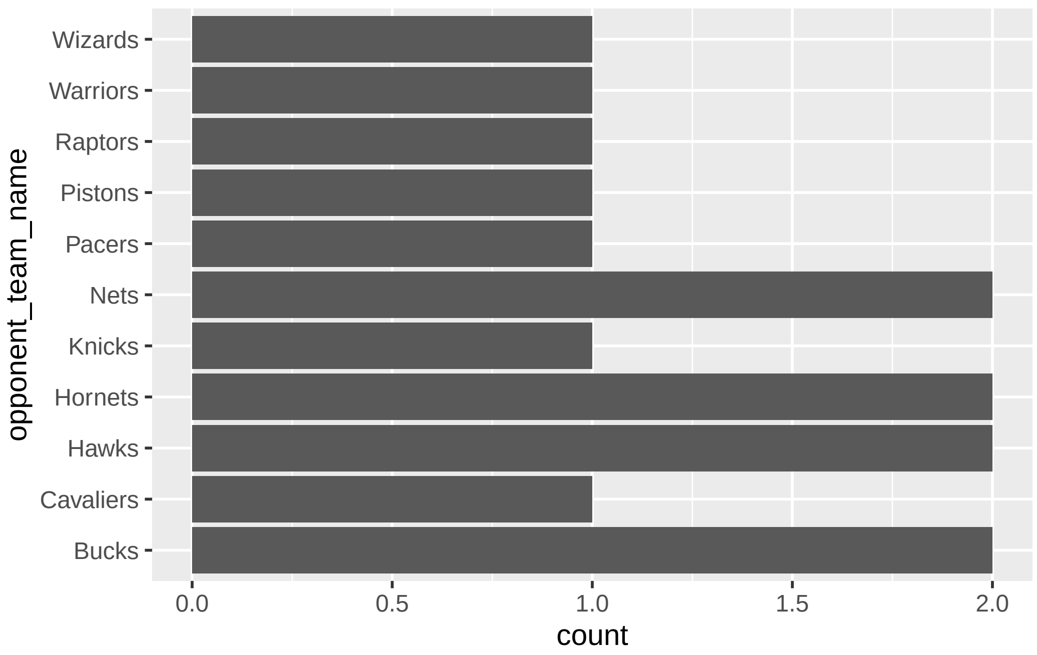
- Maybe you are interested in how the shot percentage varies between `field_goal_pct` (2 pts) and `three_point_field_goal_pct` (3 pts).

```
boston_celtics |>
  # filter(season == 2025) |>
  ggplot() +
  geom_point(aes(x = field_goal_pct, y = three_point_field_goal_pct)) +
  geom_abline(slope = 1, linetype = "dotted") +
  coord_fixed()
```



- Perhaps you want to create a bar chart (`geom_bar`) to see how many times they've played opposing teams this season.

```
boston_celtics |>
  filter(season == 2025) |>
  ggplot() +
  geom_bar(aes(y = opponent_team_name))
```



Wrap Up

By the end of this tutorial you should feel comfortable reading in a data set into R and creating a visualisation by adding layers in `ggplot2`.

Important: We will continue to build on this knowledge in coming weeks, so if you had any difficulties come along to consult to get some 1-on-1 help. We can slowly walk you through any parts you are struggling with so it makes sense.

In Your Own Time

Go through the code in exercise 2 line by line to check you understanding. You may like to also add labels and themes to these plots. Have a think about what else you might do to improve them.