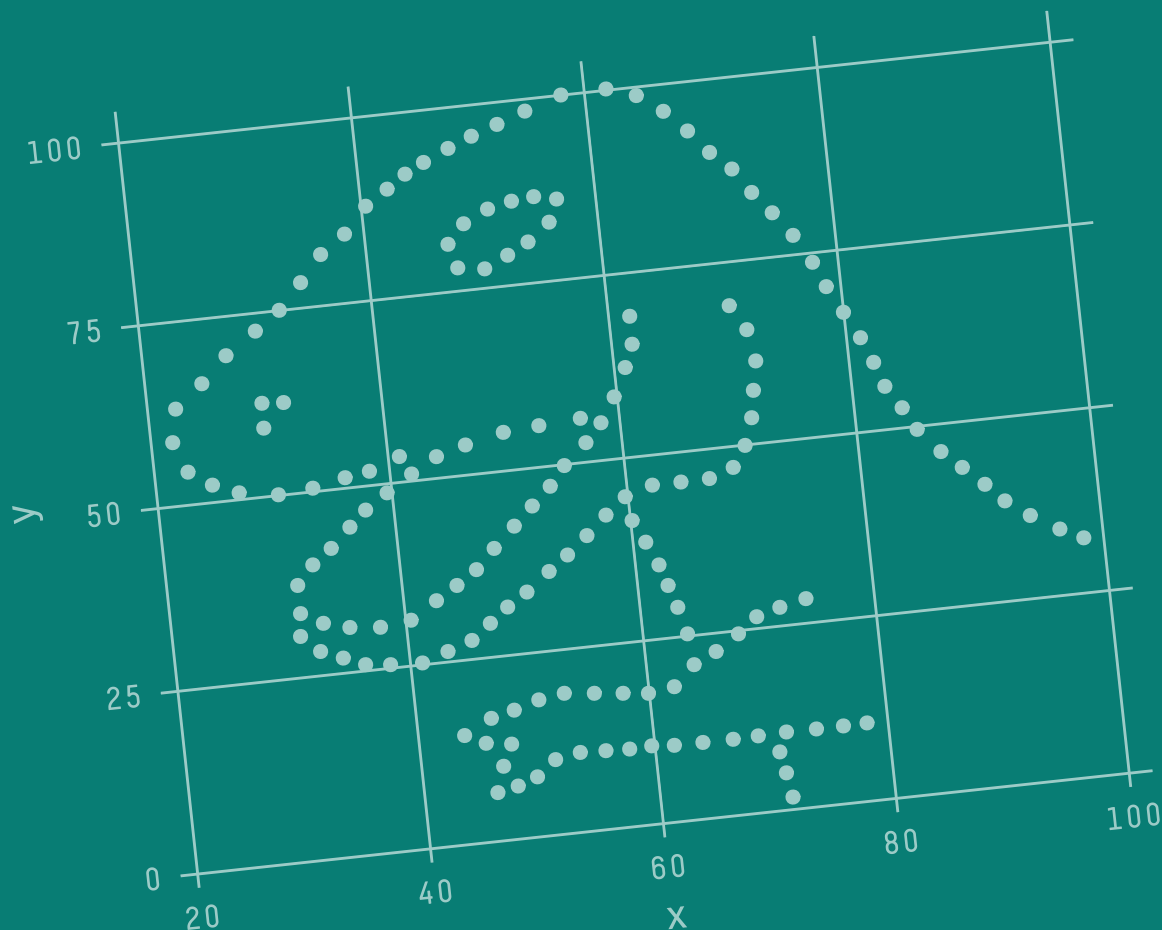


THE HITCHHIKER'S GUIDE TO GGPILOT2

Don't panic and create beautiful plots with R



CHAPTER 7

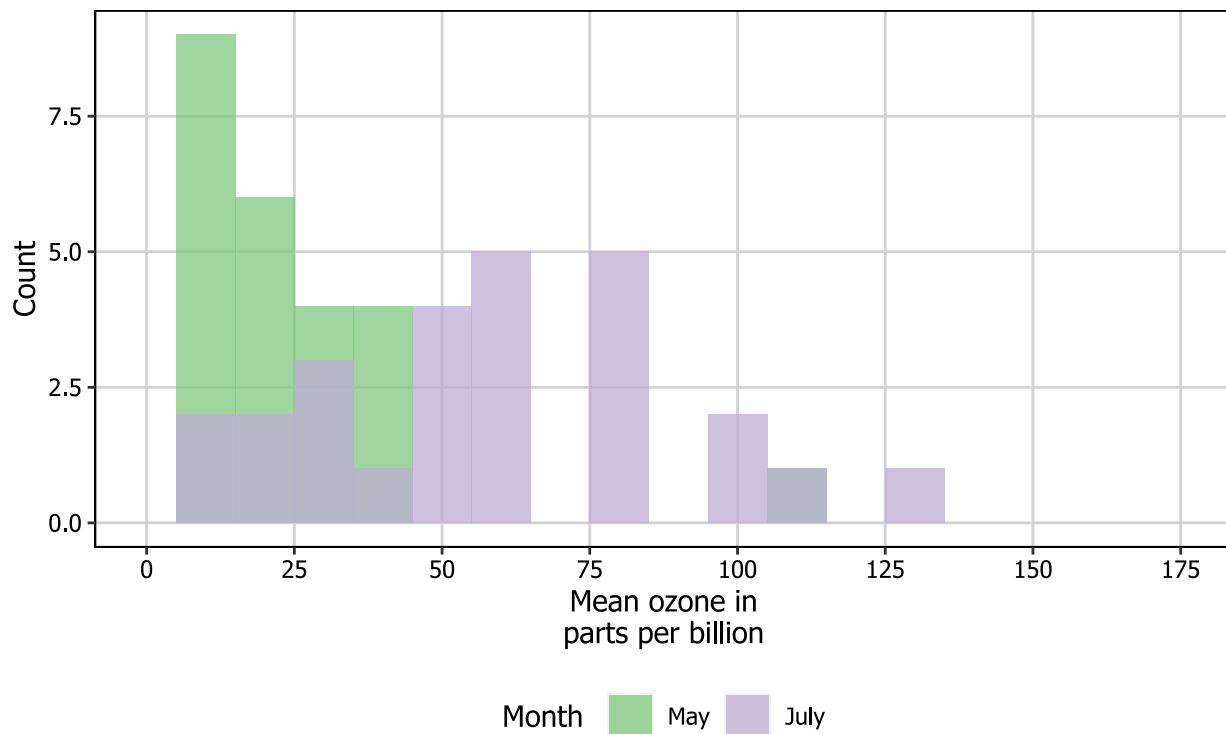
Histograms

7.1. Introduction

In this chapter, we will work towards creating the histogram below. We will take you from a basic histogram and explain all the customisations we add to the code step-by-step.

Frequency histogram of mean ozone

Source: New York State Department of Conservation



The first thing to do is load in the data and the libraries, as below:

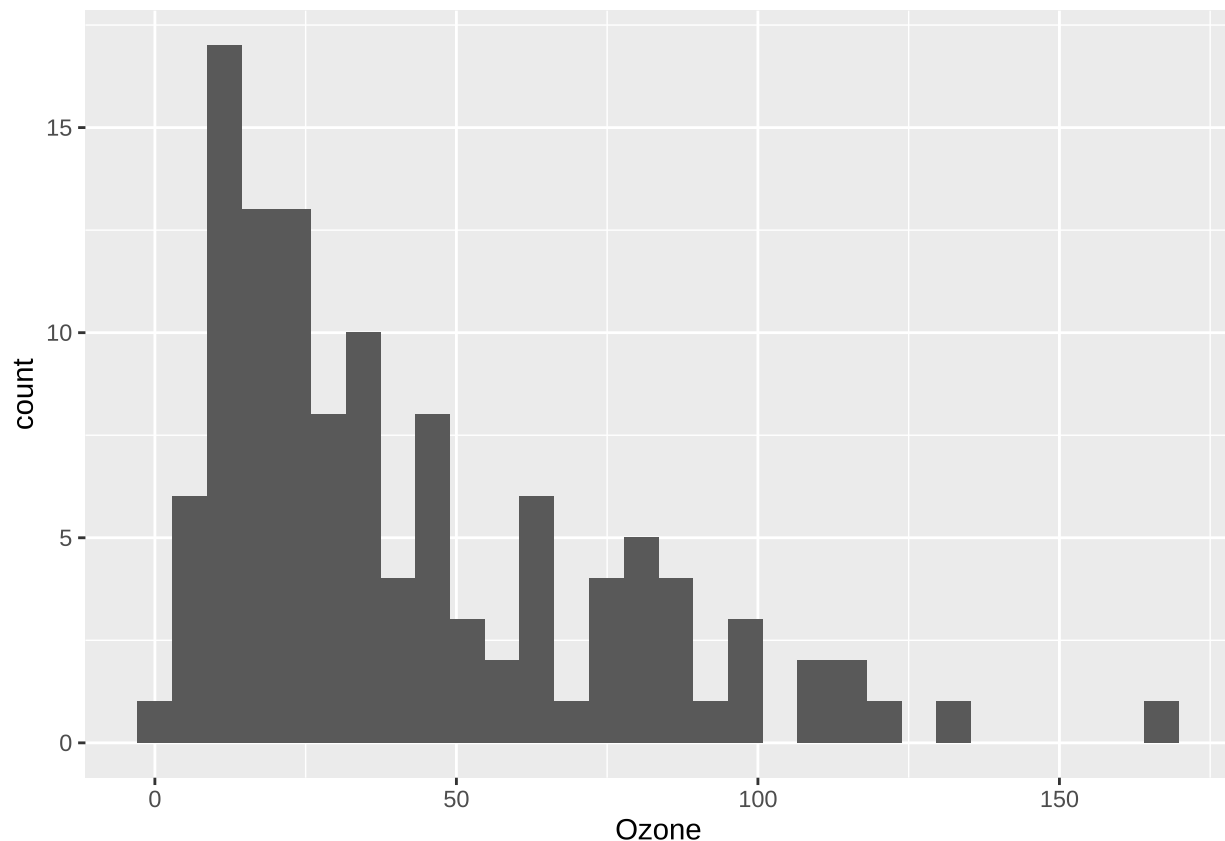
```
library(datasets)
library(dplyr)
library(ggplot2)
library(ggthemes)
library(grid)
library(RColorBrewer)

data(airquality)
```

7.2. Basic histogram

In order to initialise a plot we tell ggplot that `airquality` is our data, and specify that our x axis plots the `Ozone` variable. We then instruct ggplot to render this as a histogram by adding the `geom_histogram()` option.

```
p7 <- ggplot(airquality, aes(x = Ozone)) +
  geom_histogram()
p7
```

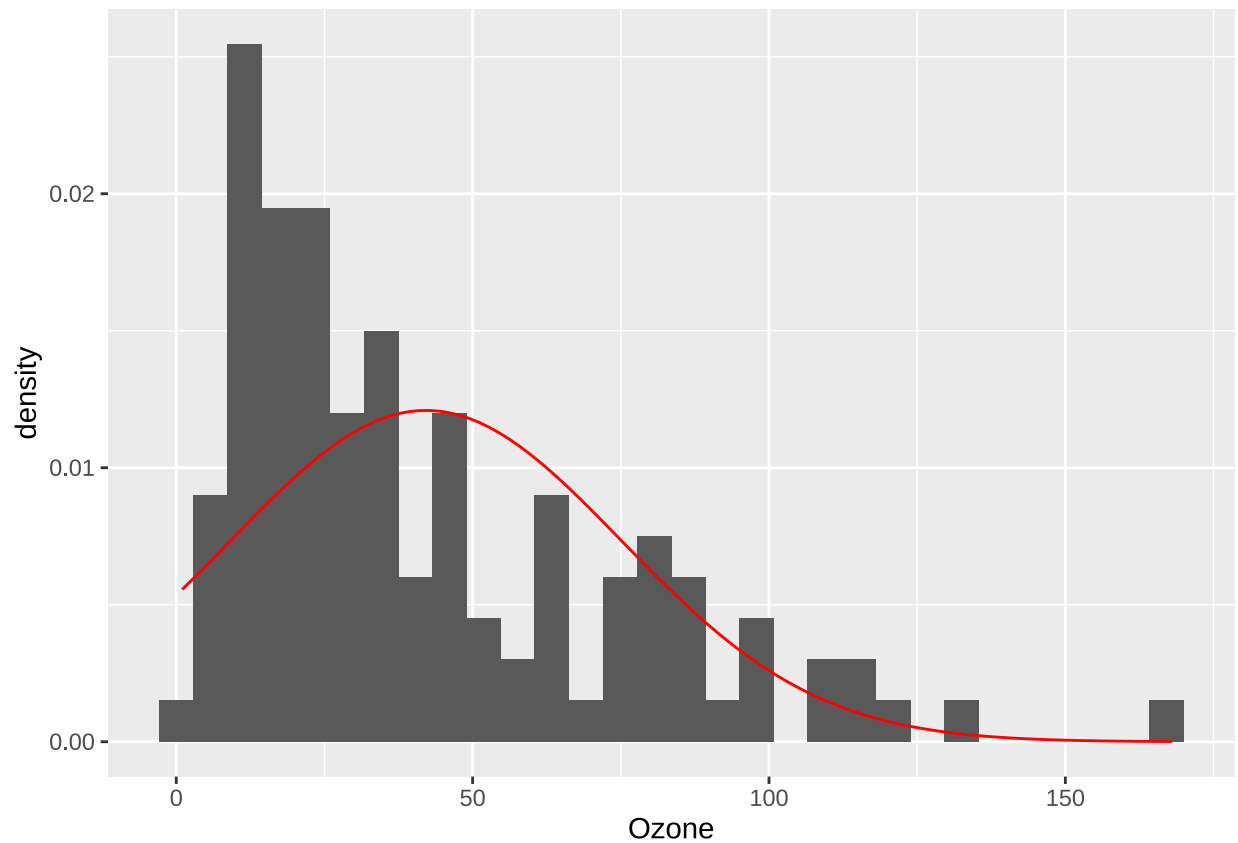


7.3. Adding a normal density curve

We can overlay a normal density function curve on top of our histogram to see how closely (or not) it fits a normal distribution. In this case, we can see it deviates from a normal distribution, showing marked positive skew. In order to overlay the function curve, we add the option `stat_function(fun = dnorm)`, and specify the shape using the `mean = mean(airquality$Ozone)` and `sd = sd(airquality$Ozone)` arguments. If you have missing data like we did, make sure you pass the `na.rm = TRUE` argument to the `mean` and `sd` parameters. Finally, you can change the colour using the `colour = "red"` argument. We will discuss how to customise colours further below.

One further change we must make to display the normal curve correctly is adding `aes(y = after_stat(density))` to the `geom_histogram` option. Note that the normal density curve will not work if you are using the frequency rather than the density, which we are changing in our next step.

```
p7 <- ggplot(airquality, aes(x = Ozone)) +  
  geom_histogram(aes(y = after_stat(density))) +  
  stat_function(  
    fun = dnorm, colour = "red",  
    args = list(  
      mean = mean(airquality$Ozone, na.rm = TRUE),  
      sd = sd(airquality$Ozone, na.rm = TRUE)  
    )  
  )  
p7
```



7.4. Changing from density to frequency

Let's go back to the basic plot and lose the function curve. To change the y-axis from density to frequency, we add the `aes(y = after_stat(count))` option to `geom_histogram`.

```
p7 <- ggplot(airquality, aes(x = Ozone)) +  
  geom_histogram(aes(y = after_stat(count)))  
p7
```