

Lecture 1.2 - Characteristics of distributions activity

DKU Stats 101 Spring 2025

Professor MacDonald

2025-10-28

Lecture 1.2 in-class activity - exploring distributions

How to Make High-Quality Histograms

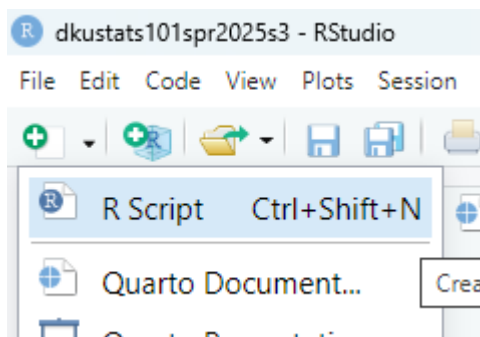
Taken from [STHDA guide](#) Additional resources: [geom_histogram documentation](#)

Part 1: Setup

First, we need to install a good-quality graphical display library:

1. If you have not installed the `tidyverse` package, you can do so by typing `install.packages("tidyverse")` into the Console at the bottom of your RStudio terminal. Answer yes when prompted.
2. Type `library(tidyverse)` into the Console after the installation has finished - this loads the package into memory

Second, we need to open a basic R Script page to help you run your code



Part 2: Basic Histograms

`ggplot` is the basic graphing function that you will use over and over again.

We can create a basic histogram of the miles per gallon (a measure of fuel efficiency in automobiles) with:

```
ggplot(mtcars, aes(x=mpg)) +  
  geom_histogram()
```

The `ggplot` function has three components:

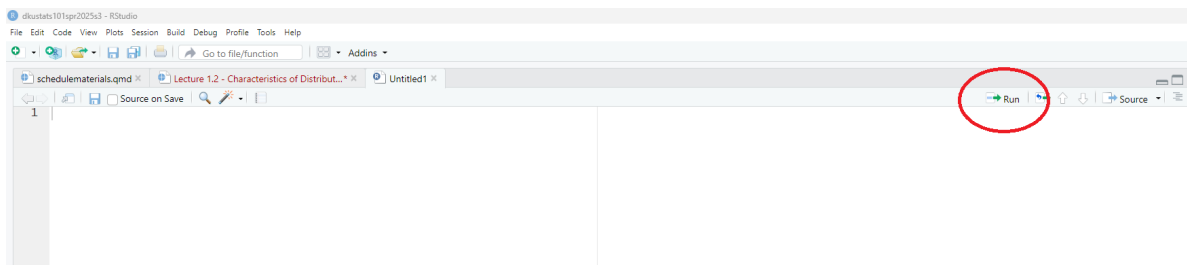
1. Dataset being used (in this case, the built-in test dataset for R called `mtcars`)
2. The data you intend to display (named inside of the `aes()` part)
3. The type of graphical display you wish to make (appended after the `ggplot()` function with a `+` sign)

However, the default `ggplot` settings often are not so great. In particular, the binwidth for this histogram is not that well chosen.

To alter the binwidth, we can set each bin to equal a set number of mpg (miles per gallon):

```
ggplot(mtcars, aes(x=mpg)) +  
  geom_histogram(binwidth=1)
```

Copy the above code to your R Script document. You can highlight the code and click on the Run button on the upper right to run the code. For the rest of the examples in the document, add them as lines to your R script.



Play around with the binwidth, try to find something that you feel accurately captures the distribution of the data.

1. What is the shape, center, and spread of this data? Are there any outliers?
2. How does changing the binwidth affect your answers to these questions?

To properly label the graph, we add the following amendments:

```
ggplot(mtcars, aes(x=mpg)) +
  geom_histogram() +
  labs(x="MPG", y="Count", title="MPG of the cars in this dataset")
```

You can see here that additional layers on top of the graph are created with each + sign. So the first part of the `ggplot()` function specifies the data to be used and the additional + each create a layer on top of the graph.

Part 3: Adding layers to the histogram

Now maybe we want to add a mean line. We can do so with the following command:

```
p <- ggplot(mtcars, aes(x=mpg)) +
  geom_histogram() +
  labs(x="MPG", y="Count", title="MPG of the cars in this dataset")
```

In this case, we are saving the output of the graph to a new variable, called `p`

You can display your graph by typing `p` in the Console

Now we can add a vertical line at the mean of `mpg` with the following command:

```
p + geom_vline(aes(xintercept=mean(mpg)), color="red", linetype="dashed", linewidth=1)
```

Or we can change the histogram type to be a density plot instead of a count plot:

```
p <- ggplot(mtcars, aes(x=mpg)) +
  geom_histogram(aes(y=after_stat(density))) +
  labs(x="MPG", y="Density", title="MPG of the cars in this dataset")
```

Note that this overwrites the previous value of `p`

Think about the difference between a density plot and the previous plots we have produced - note the change in the *y* axis

You can then overlay a density layer to get another view of the distribution of the data:

```
p + geom_density(alpha=.2, fill="#FF6666")
```

Part 4: Dividing the histogram by a categorical variable

We can also differentiate `mpg` by a categorical variable (number of cylinders in a car) via the following steps:

1. First, we need to let R know that `cyl` is a categorical variable

```
mtcars <- mtcars %>%  
  mutate(cyl = factor(cyl))
```

R generally does not know if a variable is a categorical one or not unless you tell it directly. `factor()` is the method of converting a quantitative variable to a categorical variable.

2. Then we can create histograms that vary by cylinder:

```
ggplot(mtcars, aes(x=mpg, color=cyl)) +  
  geom_histogram(fill="white") +  
  labs(x="MPG", y="Count", title="MPG of cars in the MPG Dataset")
```

Here the `color` aspect of `aes()` indicates that the color should be changed according to each group of the variable `cylinder`

Play around with the `binwidth` and add a density layer. How would you interpret this new information?

3. Next we want to create group means for each cylinder (i.e. what is the mean weight of each category of cylinders). To do so, we need the following commands (don't worry about understanding it, we'll learn more about this later):

```
cyl.means <- mtcars %>%  
  group_by(cyl) %>%  
  summarize(grp.mean = mean(mpg))
```

When you type `cyl.means` in the console, you should get the following output:

```
  cyl grp.mean  
1   4 26.66364  
2   6 19.74286  
3   8 15.10000
```

Now we can add a mean line to each of the categories:

```
ggplot(mtcars, aes(x=mpg, color=cyl)) +
  geom_histogram(fill="white") +
  labs(x="MPG", y="Count", title="MPG of cars in the MPG Dataset") +
  geom_vline(data=cyl.means, aes(xintercept=grp.mean, color=cyl), linetype="dashed")
```

Thinking about distributions of variables in this dataset

To learn more about each of the variables in the dataset, you can examine the dataset (and any function documentation in R) by entering `?mtcars`

Estimating the basic characteristics of the distribution

For the variables `hp` and `wt`, estimate or guess the shape, center and spread of the variables before examining your data.

- Shape
- Center
- Spread

Further refining your mental model of the distributions

Using the commands listed below, develop a set of measures of the distribution of the two variables. Based on your results that you generate, more fully describe and estimate what you expect the shape of the distribution to be.

- Shape
 - Skew - can estimate by comparing mean and median
 - Modes
- Center
 - Mean
 - Median
- Spread
 - Range
 - IQR
 - Standard deviation

To find these characteristics, you can use the following commands:

```
mt.cars.summary <- mtcars %>%  
  summarize(mean.mpg = mean(mpg),  
            median.mpg = median(mpg),  
            min.mpg = min(mpg),  
            max.mpg = max(mpg),  
            IQR.mpg = IQR(mpg),  
            sd.mpg = sd(mpg))
```

To find the mode, you can type:

```
mt.cars.counts <- mtcars %>%  
  count(mpg) %>%  
  arrange(desc(n))
```

Checking your work

Next, create some histograms to assess whether you are correct or not.

Also think about:

1. Why did your estimates vary from what you see in the histogram? What features of the distribution caused this discrepancy?
2. Do you think either of these variables meaningfully vary in their distribution by one of the categorical variables? Why or why not?
3. Do you see any outliers? Would either of these variables benefit from reexpressions?