# 08 Visualizing data distributions

Soyoung Park

Pusan National University
Department of Statistics

# Abstract

In this chapter, we first discuss properties of a variety of distributions and how to visualize distributions using a motivating example of student heights. We then discuss the ggplot2 geometries for these visualizations.

# Variable types

We will be working with two types of variables: categorical and numeric. Each can be divided into two other groups: categorical can be ordinal or not, whereas numerical variables can be discrete or continuous.

# Case study: describing student heights

Here we introduce a new motivating problem. It is an artificial one, but it will help us illustrate the concepts needed to understand distributions.

```r
library(tidyverse)
library(dslabs)
data(heights)
```

We first focus on male heights. We examine the female height data later.

# Distribution function

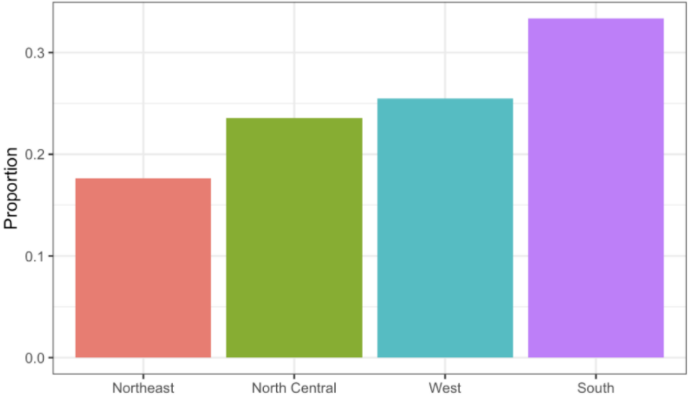
The most basic statistical summary of a list of objects or numbers is its distribution. For example, with categorical data, the distribution simply describes the proportion of each unique category. The sex represented in the heights dataset is:

```
## # A tibble: 2 x 2
##   sex      MEAN
##   <fct>  <dbl>
## 1 Female 0.227
## 2 Male   0.773
```

This two-category frequency table is the simplest form of a distribution. We don't really need to visualize it since one number describes everything we need to know: 23% are females and the rest are males.

# Distribution function

When there are more categories, then a simple barplot describes the distribution. Here is an example with US state regions:

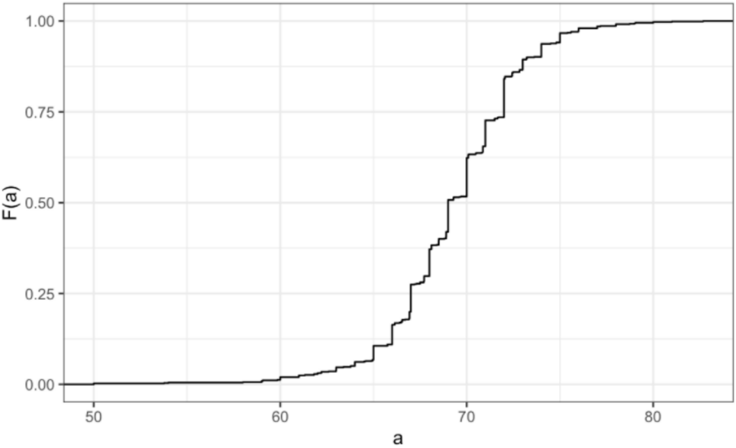


# Distribution function

This particular plot simply shows us four numbers, one for each category. We usually use barplots to display a few numbers. Although this particular plot does not provide much more insight than a frequency table itself, it is a first example of how we convert a vector into a plot.

# Cumulative distribution functions

Numerical data that are not categorical also have distributions. Here is a plot of CDF for the male height data:



# Cumulative distribution functions

Similar to what the frequency table does for categorical data, the CDF defines the distribution for numerical data. In this case, a picture is as informative as 812 numbers.

Because CDFs can be defined mathematically, so the word empirical is added to make the distinction when data is used. We therefore use the term empirical CDF (eCDF).

# Histograms

Although the CDF concept is widely discussed in statistics textbooks, the plot is actually not very popular in practice. The main reason is that it does not easily convey characteristics of interest such as:

1) what value is the distribution centered?

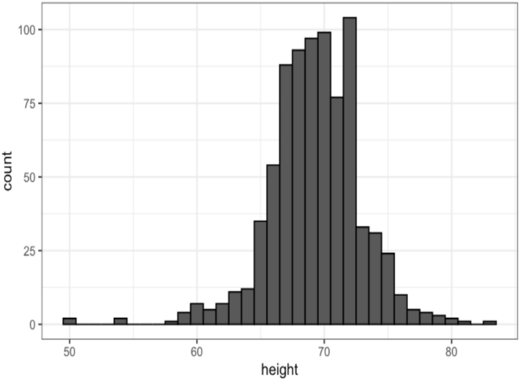
2) Is the distribution symmetric?

3) What ranges contain 95% of the values?

# Histograms

Histograms are much preferred because they greatly facilitate answering such questions. Here is the histogram for the height data splitting the range of values into one inch intervals:

$(49.5, 50.5], (50.5, 51.5], (51.5, 52.5], (52.5, 53.5], ..., (82.5, 83.5]$

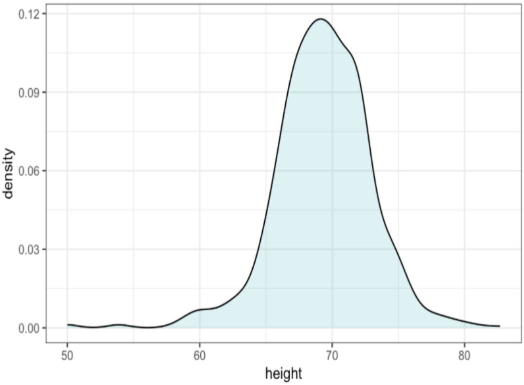


# Histograms

As you can see in the figure above, a histogram is similar to a barplot, but it differs in that the x-axis is numerical, not categorical. The histogram above is not only easy to interpret, but also provides almost all the information contained in the raw list of 812 heights with about 30 bin counts.

# Smoothed density

Smooth density plots are aesthetically more appealing than histograms. Here is what a smooth density plot looks like for our heights data:

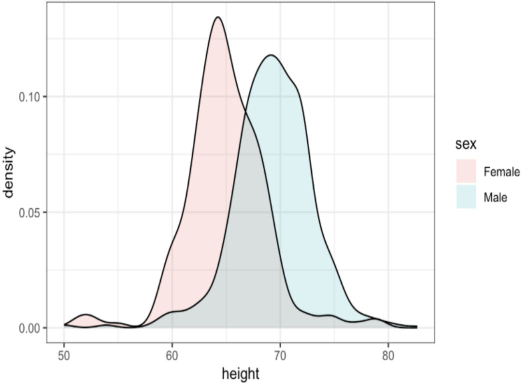


# Smoothed density

In this plot, we no longer have sharp edges at the interval boundaries and many of the local peaks have been removed. Also, the scale of the y-axis changed from counts to density. This plot have an advantage of smooth densities over histograms for visualization. Densities make it easier to compare two distributions.

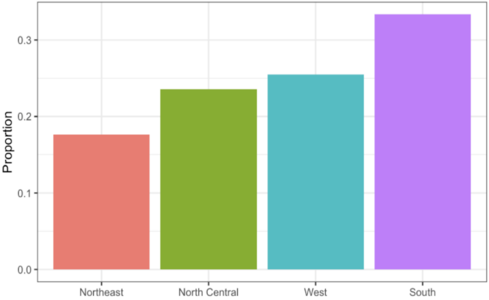
# Smoothed density

Here is an example comparing male and female heights:



# Exercises

1. In the murders dataset, the region is a categorical variable and the following is its distribution:



To the closest 5%, what proportion of the states are in the North Central region?

# Exercises

2. Which of the following is true:

a) The graph above is a histogram.

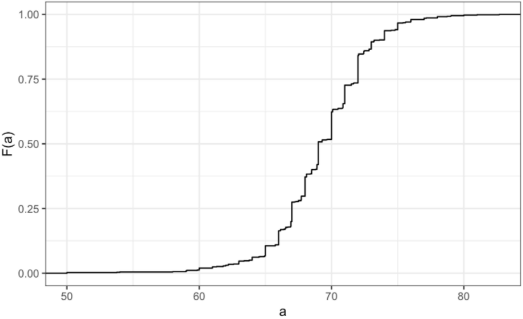
b) The graph above shows only four numbers with a bar plot.

c) Categories are not numbers, so it does not make sense to graph the distribution.

d) The colors, not the height of the bars, describe the distribution.

# Exercises

3. The plot below shows the eCDF for male heights:



Based on the plot, what percentage of males are shorter than 75 inches?

# Exercises

4. To the closest inch, what height `m` has the property that $1/2$ of the male students are taller than `m` and $1/2$ are shorter?

a) 61 inches

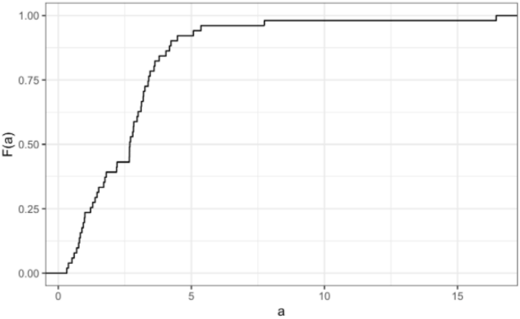
b) 64 inches

c) 69 inches

d) 74 inches

# Exercises

5. Here is an eCDF of the murder rates across states:



Knowing that there are 51 states (counting DC) and based on this plot, how many states have murder rates larger than 10 per 100,000 people?

# Exercises

6. Based on the eCDF above, which of the following statements are true:

a) About half the states have murder rates above 7 per 100,000 and the other half below.

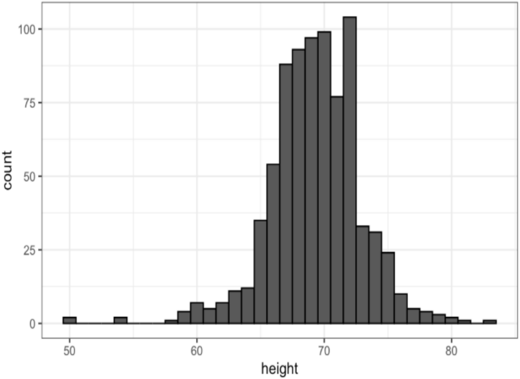
b) Most states have murder rates below 2 per 100,000.

c) All the states have murder rates above 2 per 100,000.

d) With the exception of 4 states, the murder rates are below 5 per 100,000.

# Exercises

7. Below is a histogram of male heights in our `heights` dataset:



Based on this plot, how many males are between 63.5 and 65.5?

# Exercises

8. About what percentage are shorter than 60 inches?

a) 1%

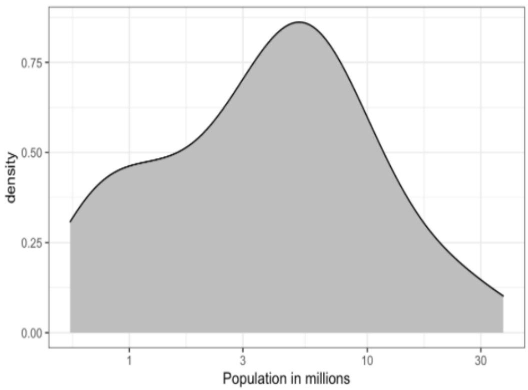
b) 10%

c) 25%

d) 50

# Exercises

9. Based on the density plot below, about what proportion of US states have populations larger than 10 million?



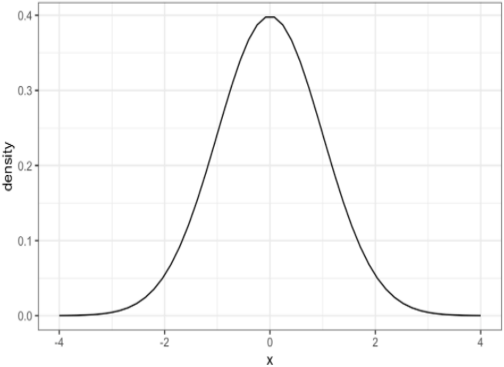
# The normal distribution

The normal distribution, also known as the bell curve and as the Gaussian distribution, is one of the most famous mathematical concepts in history. A reason for this is that approximately normal distributions occur in many situations, including gambling winnings, heights, weights, blood pressure, standardized test scores, and experimental measurement errors.

The distribution is symmetric, centered at the average, and most values (about 95%) are within 2 SDs from the average.

# The normal distribution

Here is what the normal distribution looks like when the average is 0 and the SD is 1:



# The normal distribution

The fact that the distribution is defined by just two parameters implies that if a dataset is approximated by a normal distribution, all the information needed to describe the distribution can be encoded in just two numbers: the average and the standard deviation. We now define these values for an arbitrary list of numbers.

For a list of numbers contained in a vector x, the average is defined as:

```
mu <- sum(x) / length(x)
```

and the SD is defined as:

```
s <- sqrt(sum((x-mu)^2) / length(x))
```

# The normal distribution

Let's compute the values for the height for males which we will store in the object $x$:

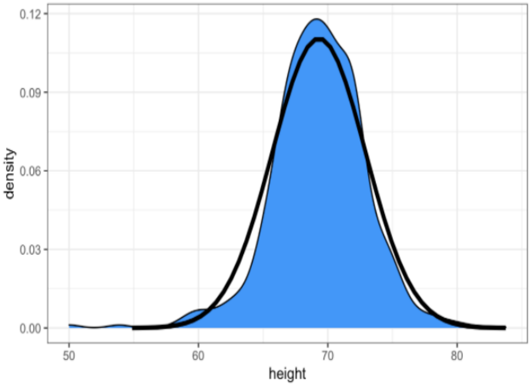
```
index <- heights$sex == "Male"
x <- heights$height[index]
```

The pre-built functions mean and sd can be used here:

```
m <- mean(x)
s <- sd(x)
c(average = m, sd = s)
```

# The normal distribution

Here is a plot of the smooth density and the normal distribution with mean = 69.3 and SD = 3.6 plotted as a black line with our student height smooth density in blue:



# The normal distribution

The normal distribution does appear to be quite a good approximation here. We now will see how well this approximation works at predicting the proportion of values within intervals.

# Standard units

The standard unit of a value tells us how many standard deviations away from the average it is. Specifically, for a value `x` from a vector `X`, we define the value of `x` in standard units as `z = (x - m)/s` with `m` and `s` the average and standard deviation of `X`, respectively.

# Quantile-quantile plots

A systematic way to assess how well the normal distribution fits the data is to check if the observed and predicted proportions match. In general, this is the approach of the quantile-quantile plot (QQ-plot).

# Quantile-quantile plots

First let's define the theoretical quantiles for the normal distribution. In R, we can evaluate Φ using the `pnorm` function:

```
pnorm(-1.96)
```

```
## [1] 0.0249979
```

In R, we can evaluate the inverse of Φ using the qnorm function.

```
qnorm(0.975)
```

```
## [1] 1.959964
```

# Quantile-quantile plots

Note that these calculations are for the standard normal distribution by default, but we can also define these for any normal distribution.

```r
qnorm(0.975, mean = 5, sd = 2)
```

```
## [1] 8.919928
```

# Quantile-quantile plots

For the normal distribution, all the calculations related to quantiles are done without data, thus the name *theoretical quantiles*. But quantiles can be defined for any distribution, including an empirical one.

Using R code, we can define `q` as the value for which `mean(x <= q) = p`. Notice that not all $p$ have a $q$ for which the proportion is exactly $p$. There are several ways of defining the best $q$ as discussed in the help for the `quantile` function.

# Quantile-quantile plots

To give a quick example, for the male heights data, we have that:

```
mean(x <= 69.5)
```

```
## [1] 0.5147783
```

So about 50% are shorter or equal to 69 inches.

# Quantile-quantile plots

The idea of a QQ-plot is that if your data is well approximated by normal distribution then the quantiles of your data should be similar to the quantiles of a normal distribution.

To construct a QQ-plot, we do the following:

1. Define a vector of $m$ proportions $p_1, p_2, \cdots, p_m$

2. Define a vector of quantiles $q_1, \cdots, q_m$ for your data for the proportions $p_1, \cdots, p_m$. We refer to these as the *sample quantiles*.

3. Define a vector of theoretical quantiles for the proportions $p_1, \cdots, p_m$ for a normal distribution with the same average and standard deviation as the data.

4. Plot the sample quantiles versus the theoretical quantiles.

# Quantile-quantile plots

Let's construct a QQ-plot using R code. Start by defining the vector of proportions.

```r
p <- seq(0.05, 0.95, 0.05)
```

To obtain the quantiles from the data, we can use the `quantile` function like this:

```r
sample_quantiles <- quantile(x, p)
```

To obtain the theoretical normal distribution quantiles with the corresponding average and SD, we use the qnorm function:

```r
theoretical_quantiles <- qnorm(p, mean = mean(x), sd = sd(x))
```

# Quantile-quantile plots

To see if they match or not, we plot them against each other and draw the identity line:

```
qplot(theoretical_quantiles, sample_quantiles) + geom_abline()
```

# Quantile-quantile plots

Notice that this code becomes much cleaner if we use standard units:

```
z <- scale(x)
sample_quantiles <- quantile(z, p)
theoretical_quantiles <- qnorm(p)
qplot(theoretical_quantiles, sample_quantiles) + geom_abline()
```

# Boxplots

The boxplot Provide a five-number summary composed of the range along with the quartiles (the 25th, 50th, and 75th percentiles).

```
boxplot(x)
```

with the box defined by the 25% and 75% percentile and showing the range. The distance between these two is called the interquartile range and the median is shown with a horizontal line.

# Stratification

In data analysis we often divide observations into groups based on the values of one or more variables associated with those observations. For example in the next section we divide the height values into groups based on a sex variable. We call this procedure *stratification* and refer to the resulting groups as *strata*.

Stratification is common in data visualization because we are often interested in how the distribution of variables differs across different subgroups.

# Case study: describing student heights (continued)

Boxplots are useful when we want to quickly compare two or more distributions. Here are the heights for men and women:

```r
heights %>%
  group_by(sex) %>%
  ggplot(aes(x=sex, y=height, fill=sex))+geom_boxplot()
```

The plot immediately reveals that males are, on average, taller than females. The standard deviations appear to be similar.

# Exercises

1. Define variables containing the heights of males and females like this:

```r
library(dslabs)
data(heights)
male <- heights$height[heights$sex == "Male"]
female <- heights$height[heights$sex == "Female"]
```

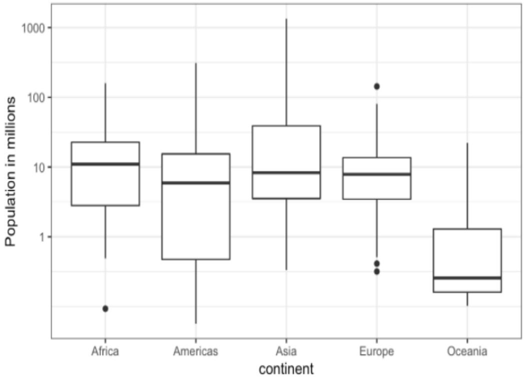
How many measurements do we have for each?

# Exercises

2. Suppose we can't make a plot and want to compare the distributions side by side. We can't just list all the numbers. Instead, we will look at the percentiles. Create a five row table showing `female_percentiles` and `male_percentiles` with the 10th, 30th, 50th, 70th, & 90th percentiles for each sex. Then create a data frame with these two as columns.

# Exercises

3. Study the following boxplots showing population sizes by country:



Which continent has the country with the biggest population size?

# Exercises

4. What continent has the largest median population size?

5. What is median population size for Africa to the nearest million?

6. What proportion of countries in Europe have populations below 14 million?

# Exercises

7. If we use a log transformation, which continent shown above has the largest interquartile range?

8. Load the height data set and create a vector x with just the male heights:

```
library(dslabs)
data(heights)
x <- heights$height[heights$sex=="Male"]
```

What proportion of the data is between 69 and 72 inches (taller than 69, but shorter or equal to 72)? Hint: use a logical operator and mean.

# Exercises

9. Suppose all you know about the data is the average and the standard deviation. Use the normal approximation to estimate the proportion you just calculated. Hint: start by computing the average and standard deviation. Then use the `pnorm` function to predict the proportions.

10. Notice that the approximation calculated in question nine is very close to the exact calculation in the first question. Now perform the same task for more extreme values. Compare the exact calculation and the normal approximation for the interval (79,81]. How many times bigger is the actual proportion than the approximation?

# Exercises

11. Approximate the distribution of adult men in the world as normally distributed with an average of 69 inches and a standard deviation of 3 inches. Using this approximation, estimate the proportion of adult men that are 7 feet tall or taller, referred to as *seven footers*. Hint: use the `pnorm` function.

12. There are about 1 billion men between the ages of 18 and 40 in the world. Use your answer to the previous question to estimate how many of these men (18-40 year olds) are seven feet tall or taller in the world?

# Exercises

13. There are about 10 National Basketball Association (NBA) players that are 7 feet tall or higher. Using the answer to the previous two questions, what proportion of the world's 18-to-40-year-old seven footers are in the NBA?

14. Repeat the calculations performed in the previous question for Lebron James' height: 6 feet 8 inches. There are about 150 players that are at least that tall.

# Exercises

15. In answering the previous questions, we found that it is not at all rare for a seven footer to become an NBA player. What would be a fair critique of our calculations:

a. Practice and talent are what make a great basketball player, not height.

b. The normal approximation is not appropriate for heights.

c. As seen in question 10, the normal approximation tends to underestimate the extreme values. It's possible that there are more seven footers than we predicted.

d. As seen in question 10, the normal approximation tends to overestimate the extreme values. It's possible that there are fewer seven footers than we predicted.

# ggplot2 geometries

Here we demonstrate how to generate plots related to distributions, specifically the plots shown earlier in this chapter.

# Barplots

To generate a barplot we can use the `geom_bar` geometry. The default is to count the number of each category and draw a bar. Here is the plot for the regions of the US.

```
murders %>% ggplot(aes(region)) + geom_bar()
```

We often already have a table with a distribution that we want to present as a barplot. Here is an example of such a table:

```
data(murders)
tab <- murders %>%
  count(region) %>%
  mutate(proportion = n/sum(n))
tab
```

# Barplots

We no longer want `geom_bar` to count, but rather just plot a bar to the height provided by the `proportion` variable. For this we need to provide `x` (the categories) and `y` (the values) and use the `stat="identity"` option.

```r
tab %>% ggplot(aes(region, proportion)) + geom_bar(stat = "ide
```

# Histograms

To generate histograms we use geom_histogram. The code looks like this:

```
heights %>%
  filter(sex == "Female") %>%
  ggplot(aes(height)) +
  geom_histogram()
```

If we run the code above, it gives us a message. We previously used a bin size of 1 inch, so the code looks like this:

```
heights %>%
  filter(sex == "Female") %>%
  ggplot(aes(height)) +
  geom_histogram(binwidth = 1)
```

# Histograms

Finally, if for aesthetic reasons we want to add color, we use the arguments described in the help file. We also add labels and a title:

```
heights %>%
  filter(sex == "Female") %>%
  ggplot(aes(height)) +
  geom_histogram(binwidth = 1, fill = "blue", col = "black") +
  xlab("Female heights in inches") +
  ggtitle("Histogram")
```

# Density plots

To create a smooth density, we use the `geom_density`. To make a smooth density plot with the data previously shown as a histogram we can use this code:

```
heights %>%
  filter(sex == "Female") %>%
  ggplot(aes(height)) +
  geom_density()
```

To fill in with color, we can use the `fill` argument.

```
heights %>%
  filter(sex == "Female") %>%
  ggplot(aes(height)) +
  geom_density(fill="blue")
```

# Density plots

To change the smoothness of the density, we use the adjust argument to multiply the default value by that adjust. For example, if we want the bandwidth to be twice as big we use:

```
heights %>%
  filter(sex == "Female") +
  geom_density(fill="blue", adjust = 2)
```

# Boxplots

The geometry for boxplot is `geom_boxplot`. As discussed, boxplots are useful for comparing distributions.

```
heights %>%
  group_by(sex) %>%
  ggplot(aes(x=sex, y=height))+geom_boxplot()
```

# QQ-plots

For qq-plots we use the geom_qq geometry. From the help file, we learn that we need to specify the sample. Here is the qqplot for men heights.

```r
heights %>% filter(sex=="Male") %>%
  ggplot(aes(sample = height)) +
  geom_qq()
```

By default, the sample variable is compared to a normal distribution with average 0 and standard deviation 1.

# QQ-plots

To change this, we use the dparams arguments based on the help file. Adding an identity line is as simple as assigning another layer. For straight lines, we use the geom_abline function. The default line is the identity line (slope = 1, intercept = 0).

```r
params <- heights %>% filter(sex=="Male") %>%
  summarize(mean = mean(height), sd = sd(height))

heights %>% filter(sex=="Male") %>%
  ggplot(aes(sample = height)) +
  geom_qq(dparams = params) +
  geom_abline()
```

# QQ-plots

Another option here is to scale the data first and then make a qqplot against the standard normal.

```r
heights %>%
  filter(sex=="Male") %>%
  ggplot(aes(sample = scale(height))) +
  geom_qq() +
  geom_abline()
```

# Quick plots

We can also use `qplot` to make histograms, density plots, boxplot, qqplots and more. Although it does not provide the level of control of `ggplot`, `qplot` is definitely useful as it permits us to make a plot with a short snippet of code.

Suppose we have the female heights in an object `x`:

```
x <- heights %>%
  filter(sex=="Male") %>%
  pull(height)
```

To make a quick histogram we can use:

```
qplot(x)
```

# Quick plots

To make a quick qqplot you have to use the `sample` argument. Note that we can add layers just as we do with `ggplot`.

```
qplot(sample = scale(x)) + geom_abline()
```

If we supply a factor and a numeric vector, we obtain a plot like the one below.

# Quick plots

Note that in the code below we are using the `data` argument. Because the data frame is not the first argument in `qplot`, we have to use the dot operator.

```
heights %>% qplot(sex, height, data = .)
```

We can also select a specific geometry by using the `geom` argument. So to convert the plot above to a boxplot, we use the following code:

```
heights %>% qplot(sex, height, data = ., geom = "boxplot")
```

# Quick plots

We can also use the geom argument to generate a density plot instead of a histogram:

```r
qplot(x, geom = "density")
```

Although not as much as with ggplot, we do have some flexibility to improve the results of qplot. Here is an example:

```r
qplot(x, bins=15, color = I("black"), xlab = "Population")
```

# Quick plots

Technical note: The reason we use `I("black")` is because we want qplot to treat `"black"` as a character rather than convert it to a factor, which is the default behavior within `aes`, which is internally called here. In general, the function `I` is used in R to say “keep it as it is.”

# Exercises

1. Now we are going to use the `geom_histogram` function to make a histogram of the heights in the `height` data frame. Make a histogram of all the plots. What is the variable containing the heights?

a) `sex`

b) `heights`

c) `height`

d) `heights$height`

# Exercises

2. Now create a ggplot object using the pipe to assign the heights data to a ggplot object. Assign `height` to the `x` values through the `aes` function.

3. Now we are ready to add a layer to actually make the histogram. Use the object created in the previous exercise and the `geom_histogram` function to make the histogram.

# Exercises

4. Note that when we run the code in the previous exercise we get the warning: `stat_bin()` using `bins = 30`. Pick better value with `binwidth`. Use the `binwidth` argument to change the histogram made in the previous exercise to use bins of size 1 inch.

5. Instead of a histogram, we are going to make a smooth density plot. In this case we will not make an object, but instead render the plot with one line of code. Change the geometry in the code previously used to make a smooth density instead of a histogram.

# Exercises

6. Now we are going to make a density plot for males and females separately. We can do this using the `group` argument. We assign groups via the aesthetic mapping as each point needs to a group before making the calculations needed to estimate a density.

7. We can also assign groups through the `color` argument. This has the added benefit that it uses color to distinguish the groups. Change the code above to use color.

# Exercises

8. We can also assign groups through the `fill` argument. This has the added benefit that it uses colors to distinguish the groups, like this:

```
heights %>%
  ggplot(aes(height, fill = sex)) +
  geom_density()
```

However, here the second density is drawn over the other. We can make the curves more visible by using alpha blending to add transparency. Set the alpha parameter to 0.2 in the `geom_density` function to make this change.