# Inference for one mean

*Put your name here*

*Put the date here*

## Introduction

In this module, we'll learn about the Student t distribution and use it to perform a t-test for a single mean.

## Instructions

Presumably, you have already created a new project and downloaded this file into it. Please knit the document and work back and forth between this R Markdown file and the PDF output as you work through this module.

When you are finished with the assignment, knit to PDF one last time, proofread the PDF file **carefully**, export the PDF file to your computer, and then submit your assignment.

Sometimes you will be asked to add your own R code. That will appear in this document as a code chunk with a request for you to add your own code, like so:

```
## Add code here to [do some task]...
```

Be sure to remove the line `## Add code here to [do some task]...` when you have added your own code.

Sometimes you will be asked to type up your thoughts. That will appear in the document as follows:

Please write up your answer here.

Again, please be sure to remove the line "Please write up your answer here" when you have written up your answer. In these areas of the assignment, please use contextually meaningful full sentences/paragraphs (unless otherwise indicated) and proper spelling, grammar, punctuation, etc. This is not R code, but rather a free response section where you talk about your analysis and conclusions. If you need to use some R code as well, you can use inline R code inside the block between `\begin{answer}` and `\end{answer}`, or if you need an R code chunk, please go outside the `answer` block and start a new code chunk.

## Load Packages

We load the standard `mosaic` package as well as the `OIdata` package to get the `teacher` data and the `openintro` package for the `hsb2` data. The `broom` package will give us tidy output.

```r
library(OIdata)
data(teacher)
library(openintro)
library(broom)
library(mosaic)
```

We'll set the seed for our simulations.

```
set.seed(5151977)
```

When the number of digits gets large enough, R has an annoying habit of trying to report these numbers in scientific notation. The worst part is that it actually breaks the process of knitting to PDF. The following command will turn off this tendency.

```
options(scipen = 999)
```

## Simulating means

Systolic blood pressure (SBP) for women in the U.S. and Canada follows a normal distribution with a mean of 114 and a standard deviation of 14.

Suppose we gather a random sample of 16 women and measure their SBP. We can simulate doing that with the `rnorm` command:

```
SBP_sample <- rnorm(16, mean = 114, sd = 14)
SBP_sample
```

```
##  [1]  99.75130 126.47739  99.53632 115.05247 137.72850 132.87008 104.97316
##  [8] 129.09372 104.25388  87.01786 128.84187 102.09662 142.75688 102.53217
## [15]  93.13302 105.39891
```

We summarize our sample by taking the mean and standard deviation:

```
mean(SBP_sample)
```

```
## [1] 113.2196
```

```
sd(SBP_sample)
```

```
## [1] 17.20576
```

The sample mean $\bar{y} = 113.2196343$ is pretty close to the true population mean $\mu = 114$ and the sample standard deviation $s = 17.205759$ is somewhat close to the true population standard deviation $\sigma = 14$. ($\mu$ is the Greek letter "mu" and $\sigma$ is the Greek letter "sigma".)

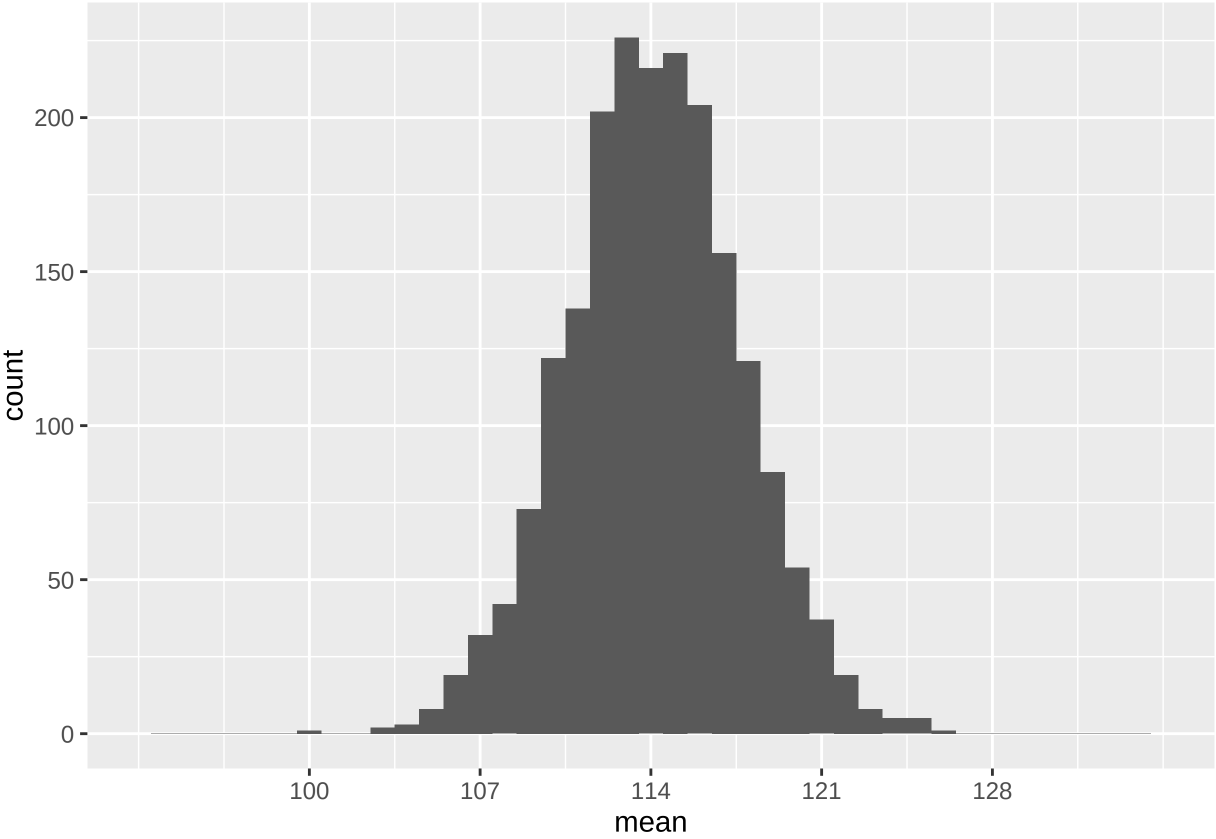
Let's simulate lots of samples of size 16. For each sample, we calculate the sample mean.

```
sims <- do(2000) * mean(rnorm(16, mean = 114, sd = 14))
head(sims)
```

```
##       mean
## 1 113.4971
## 2 116.1031
## 3 121.7838
## 4 115.8919
## 5 115.0994
## 6 116.9770
```

Again, we see that the sample means are close to 114, but there is some variability. Naturally, not every sample is going to have an average of exactly 114. So how much variability do we expect? Let's graph and find out. I'm going to set the x-axis manually so that we can do some comparisons later.

```
ggplot(sims, aes(x = mean)) +
    geom_histogram(binwidth = 1) +
    scale_x_continuous(limits = c(93, 135),
                       breaks = c(100, 107, 114, 121, 128))
```



Most sample means are around 114, but there is a good range of possibilities from about 100 to 128. The population standard deviation $\sigma$ is 14, but the standard deviation in this graph is clearly much smaller than that. (Pretty much all the samples are within 14 of the mean!)

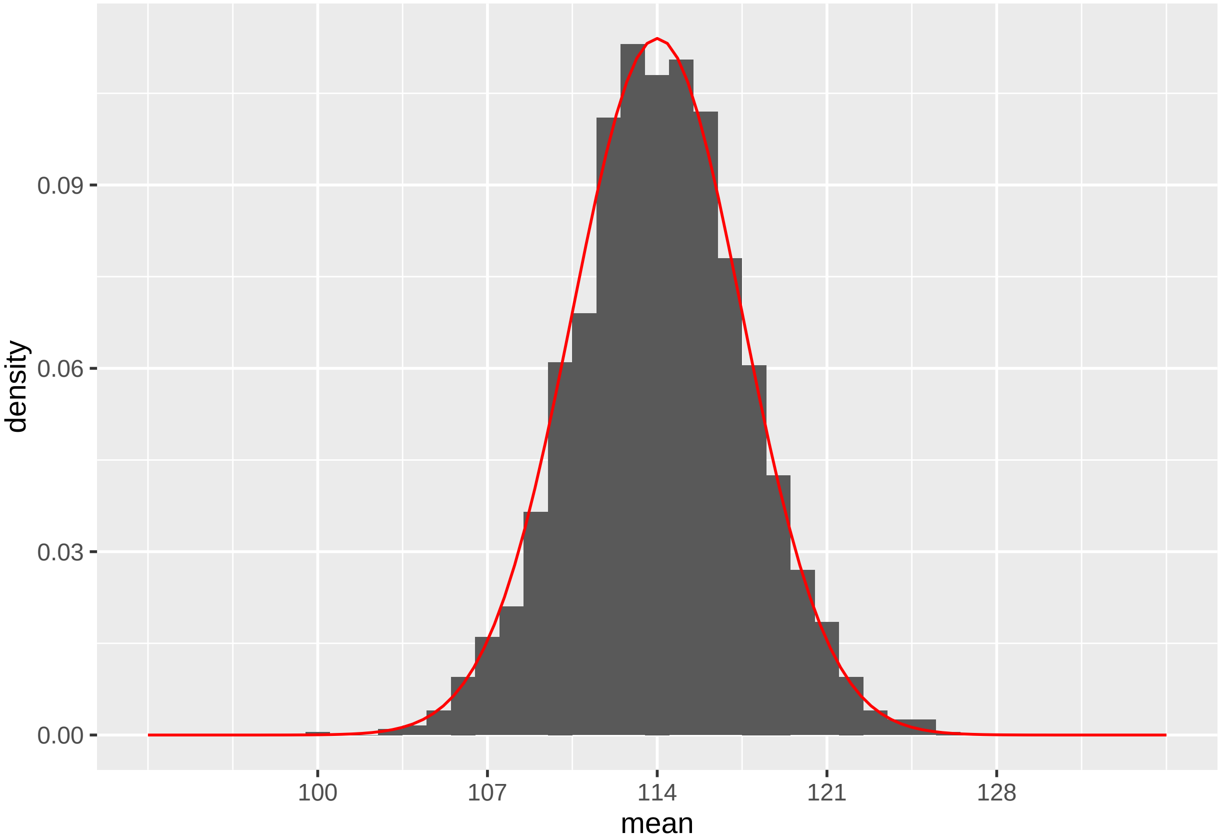
With some fancy mathematics, one can show that the standard deviation of this sampling distribution is not $\sigma$, but rather $\sigma/\sqrt{n}$. In other words, this sampling distribution of the mean has a standard error of

$$\frac{\sigma}{\sqrt{n}} = \frac{14}{\sqrt{16}} = 3.5.$$

This makes sense: as the sample size increases, we expect the sample mean to be more and more accurate, so the standard error should shrink with large sample sizes.

Let's re-scale the y-axis to use percentages instead of counts. Then we should be able to superimpose the normal model $N(114, 3.5)$ to check visually that it's the right fit.

```
ggplot(sims, aes(x = mean)) +
    geom_histogram(aes(y = ..density..), binwidth = 1) +
    scale_x_continuous(limits = c(93, 135),
                       breaks = c(100, 107, 114, 121, 128)) +
    stat_function(fun = dnorm, args = list(mean = 114, sd = 3.5),
                  color = "red")
```



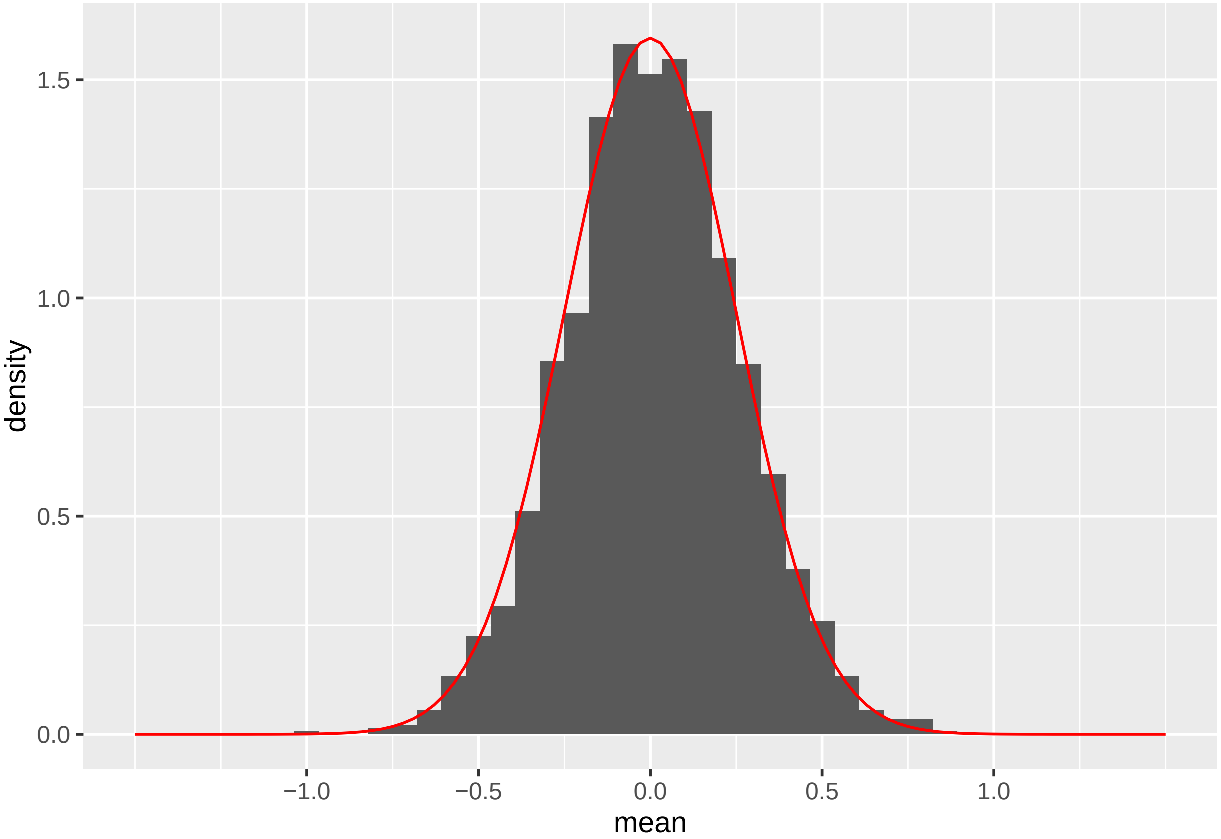
Looks good!

All we do now is convert everything to z-scores. In other words, suppose we sample 16 individuals from a population distributed according to the normal model $N(0,1)$. Now the standard error of the sampling distribution is

$$\frac{\sigma}{\sqrt{n}} = \frac{1}{\sqrt{16}} = 0.25.$$

The following code will accomplish all of this. (Don't worry about the messy syntax. All I'm doing here is making sure that this graph looks exactly the same as the previous graph, except now centered at $\mu = 0$ instead of $\mu = 114$.)

```
sims_z <- data.frame(mean = scale(sims$mean, center = 114, scale = 14))
ggplot(sims_z, aes(x = mean)) +
    geom_histogram(aes(y = ..density..), binwidth = 1/14) +
    scale_x_continuous(limits = c(-1.5, 1.5),
                       breaks = c(-1, -0.5, 0, 0.5, 1)) +
    stat_function(fun = dnorm, args = list(mean = 0, sd = 0.25),
                  color = "red")
```



## Unknown standard errors

If we want to run a hypothesis test, we will have a null hypothesis about the true value of the population mean $\mu$. For example,

$$H_0 : \mu = 114$$

Now we gather a sample and compute the sample mean, say 113.2196343. We would like to be able to compare the sample mean $\bar{y}$ to the hypothesized value 114 using a z-score:

$$z = \frac{(\bar{y} - \mu)}{\sigma/\sqrt{n}} = \frac{(113.22 - 114)}{\sigma/\sqrt{16}}.$$

However, we have a problem: we don't know the true value of $\sigma$. (In our SBP example, we do happen to know it's 14, but we won't know this for a general research question.)

The best we can do with a sample is calculate this z-score replacing the unknown $\sigma$ with the sample standard deviation $s$, 17.205759. We'll call this a "t-score" instead of a "z-score":

$$t = \frac{(\bar{y} - \mu)}{s/\sqrt{n}} = \frac{(113.22 - 114)}{17.2/\sqrt{16}} = -0.18.$$

The problem is that $s$ is not a perfect estimate of $\sigma$. We saw earlier that $s$ is usually close to $\sigma$, but $s$ has its own sampling variability. That means that our earlier simulation in which we assumed that $\sigma$ was known and equal to 14 was wrong for the type of situation that will arise when we run a hypothesis test. How wrong was it?

## Simulating t-scores

Let's run the simulation again, but this time with the added uncertainty of using $s$ to estimate $\sigma$.

The first step is to write a little function of our own to compute simulated t-scores. This function will take a sample of size $n$ from the true population $N(\mu, \sigma)$, calculate the sample mean and sample standard deviation, then compute the t-score.

```
sim_t <- function(n, mu, sigma) {
    sample_values <- rnorm(n, mean = mu, sd = sigma)
    y_bar <- mean(sample_values)
    s <- sd(sample_values)
    t <- (y_bar - mu)/(s / sqrt(n))
}
```

Now we can simulate doing this 5000 times.

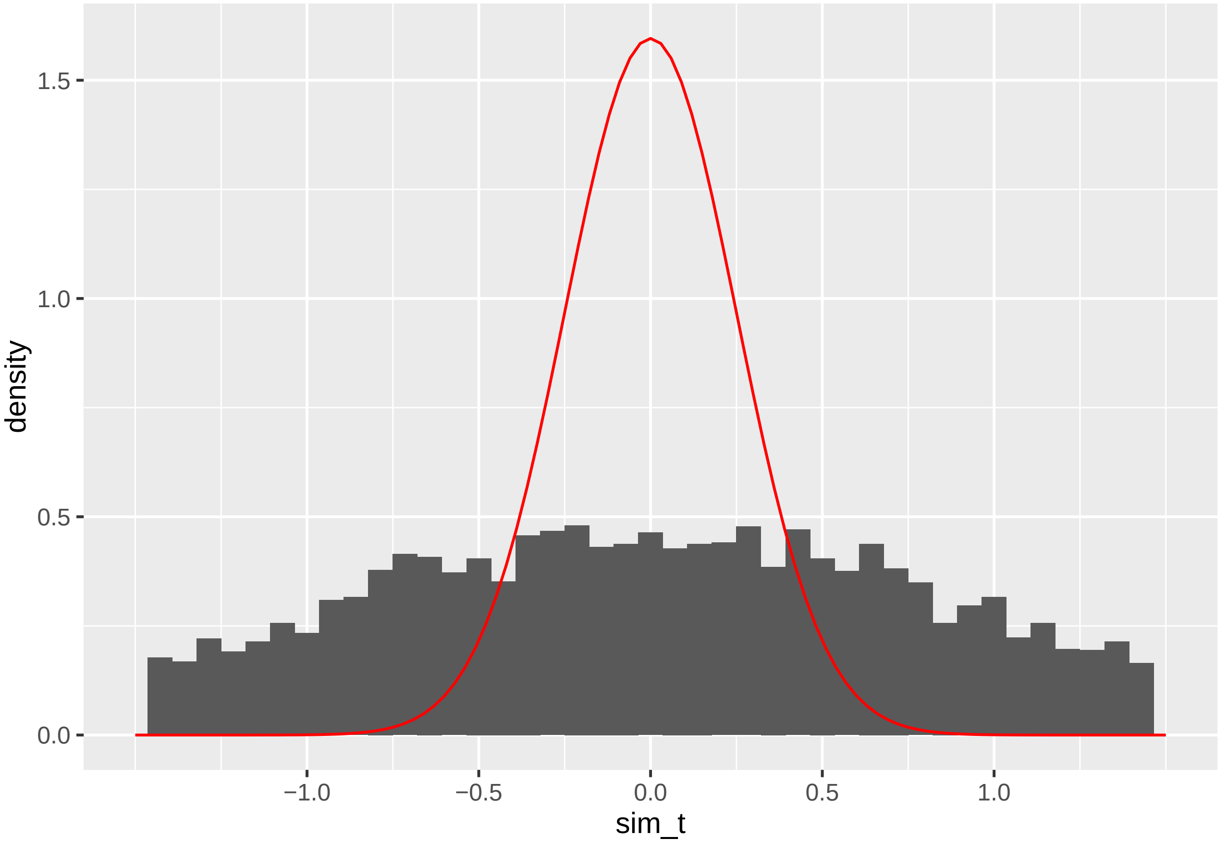
```
sims_t <- do(5000) * sim_t(16, mu = 114, sigma = 14)
head(sims_t)
```

```
##         sim_t
## 1 -0.18257964
## 2  0.01200787
## 3 -0.13536246
## 4  1.46131619
## 5  0.94319727
## 6  0.05878609
```

Let's plot our simulated t-scores alongside the normal sampling distribution $N(0, 0.25)$ from before (also using the same x-axis scale from before).

```
ggplot(sims_t, aes(x = sim_t)) +
    geom_histogram(aes(y = ..density..), binwidth = 1/14) +
    scale_x_continuous(limits = c(-1.5, 1.5),
                       breaks = c(-1, -0.5, 0, 0.5, 1)) +
    stat_function(fun = dnorm,  args = list(mean = 0, sd = 0.25),
                  color = "red")
```

```
## Warning: Removed 747 rows containing non-finite values (stat_bin).
```



Ruh roh! These t-scores are not even close to the normal model we had when we knew $\sigma$.

William Gosset figured this all out in the early 20th century. He found a new function that is similar to a normal distribution, but is more spread out. This new function accounts for the extra variability one gets when using the sample standard deviation $s$ as an estimate for the true population standard deviation $\sigma$.

Gosset published his findings under the pseudonym "Student" and this new function became known as the *Student t distribution*.

Gosset also realized that the spread of the t distribution depends on the sample size. This makes sense: the accuracy of $s$ will be greater when we have a larger sample. In fact, for large enough samples, the t-distribution is very close to a normal model.

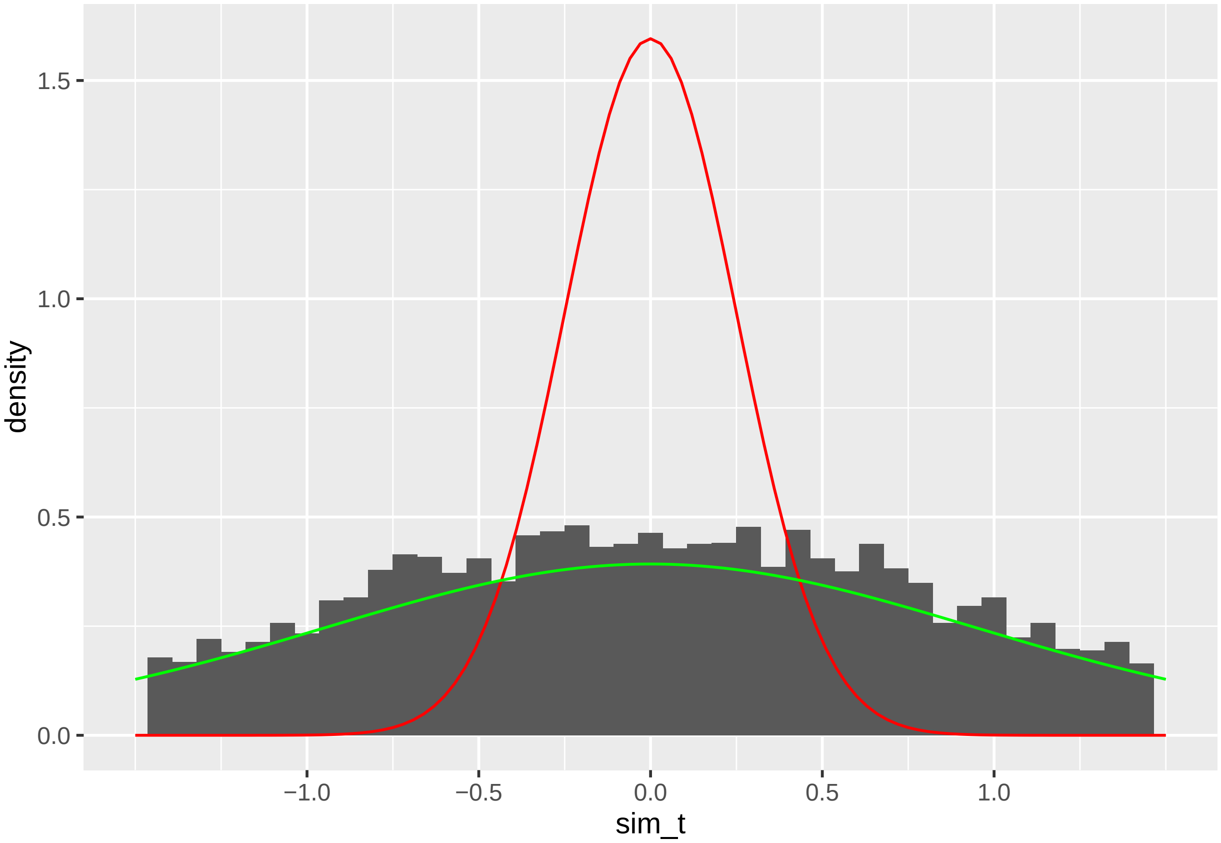
Gosset used the term *degrees of freedom* to describe how the sample size influences the spread of the t distribution. It's somewhat mathematical and technical, so suffice it to say here that the number of degrees of freedom is simply the sample size minus 1:

$$df = n - 1.$$

So is the t model right for our simulated t-scores? Our sample size was 16, so we should use a t model with 15 degrees of freedom. Let's plot it in green on top of our previous graph and see:

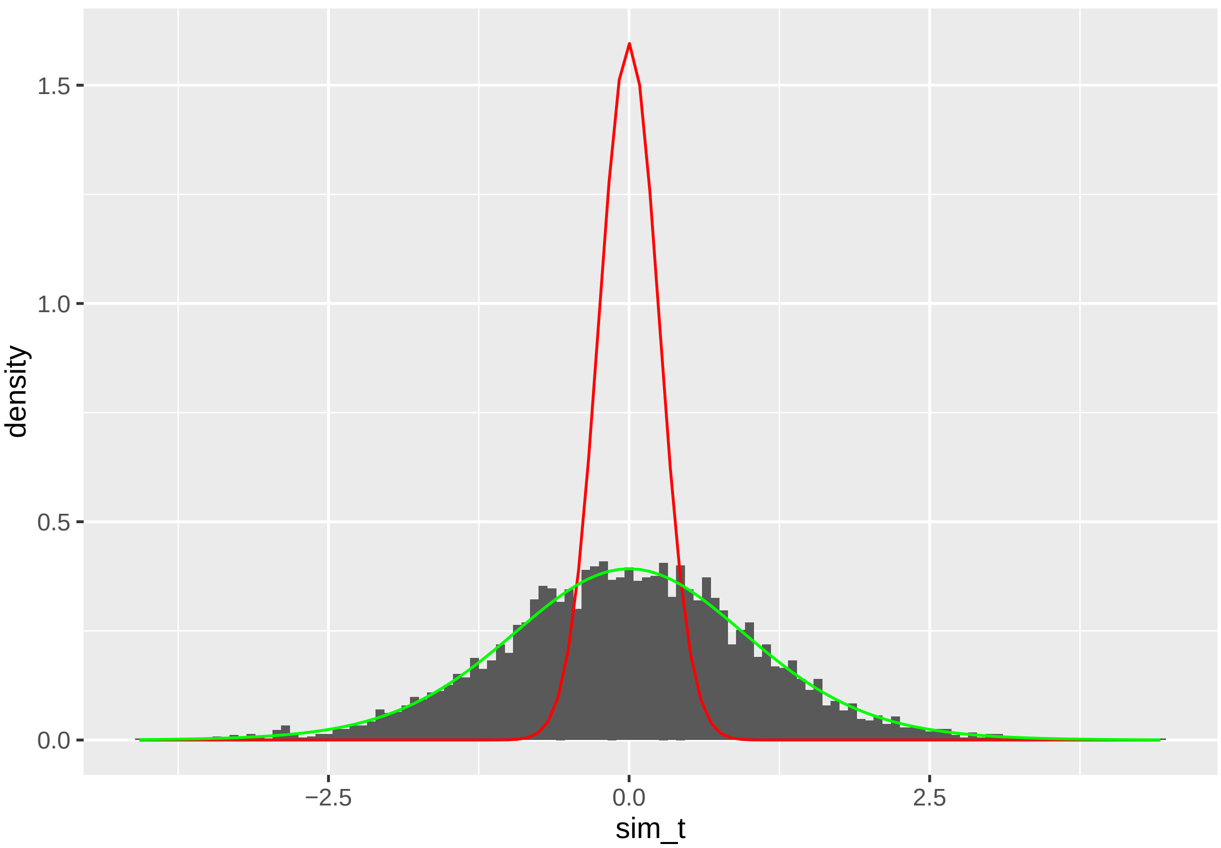
```r
ggplot(sims_t, aes(x = sim_t)) +
    geom_histogram(aes(y = ..density..), binwidth = 1/14) +
    scale_x_continuous(limits = c(-1.5, 1.5),
                       breaks = c(-1, -0.5, 0, 0.5, 1)) +
    stat_function(fun = dnorm,  args = list(mean = 0, sd = 0.25),
                  color = "red") +
    stat_function(fun = dt,  args = list(df = 15),
                  color = "green")
```

```
## Warning: Removed 747 rows containing non-finite values (stat_bin).
```



The green curve seems to fit the simulated values much better. Let's zoom out and see more of the picture.

```
ggplot(sims_t, aes(x = sim_t)) +
    geom_histogram(aes(y = ..density..), binwidth = 1/14) +
    stat_function(fun = dnorm,  args = list(mean = 0, sd = 0.25),
                  color = "red") +
    stat_function(fun = dt,  args = list(df = 15),
                  color = "green")
```



The t model with $df = 15$ fits great! Clearly, the normal model does not.

## Inference for one mean

When we have a single numerical variable, we can ask if the sample mean is consistent or not with a null hypothesis. We will use a t model for our sampling distribution model as long as certain conditions are met.

One of the assumptions we made in the simulation above was that the true population was normally distributed. In general, we have no way of knowing if this is true. So instead we check the *nearly normal* condition: if a histogram or QQ plot of our data shows that the data is nearly normal, then there is a reasonable assumption that the whole population is shaped the same way.

If our sample size is large enough, the central limit theorem tells us that the sampling distribution gets closer and closer to a normal model. Therefore, we'll use a rule of thumb that says that if the sample size is greater than 30, we won't worry about any deviations from normality in the data.

The number 30 is somewhat arbitrary. If the sample size is 25 and a histogram shows only a little skewness, we're probably okay. But if the sample size is 10, we need for the data to be very normal to justify using the t model. The irony, of course, is that small sample sizes are the hardest to check for normality. We'll have to use our best judgment.

## Outliers

We also need to be on the lookout for outliers. We've seen before that outliers can have a huge effect on means and standard deviations, especially when sample sizes are small. Whenever we find an outlier, we need to investigate.

Some outliers are mistakes. Perhaps someone entered data incorrectly into the computer. When it's clear that outliers are data entry errors, we are free to either correct them (if we know what error was made) or delete them from our data completely.

Some outliers are not necessarily mistakes, but should be excluded for other reasons. For example, if we are studying the weight of birds and we have sampled a bunch of hummingbirds and one emu, the emu's weight will appear as an outlier. It's not that its weight is "wrong", but it clearly doesn't belong in the analysis.

In general, though, outliers are "real" data that just happen to be unusual. It's not ethical just to throw away such data points because they are inconvenient. (We only do so in very narrow and well-justified circumstances like the emu.) The best policy to follow when faced with such outliers is to run inference twice—once with the outlier included, and once with the outlier excluded. If when running a hypothesis test the conclusion is the same either way, then the outlier wasn't all that influential, so we leave it in. If when computing a confidence interval the endpoints don't change a lot either way, then we leave the outlier in. However, when conclusions or intervals are dramatically different depending on whether the outlier was in or out, then we have no choice but to state that honestly.

## Research question

The `teacher` data from the `OIdata` package contains information on 71 teachers employed by the St. Louis Public School in Michigan. According to Google, the average teacher salary in Michigan was $63,024 in 2010. So does this data suggest that the teachers in this region are paid differently than teachers in other parts of Michigan?

Let's walk through the rubric.

## Exploratory data analysis

**Use data documentaton (help files, code books, Google, etc.), the str command, and other summary functions to understand the data.**

[Type `library(OIdata)` then `?teacher` to read the help file.]

```
str(teacher)
```

```
## 'data.frame':    71 obs. of  8 variables:
##  $ id        : Factor w/ 71 levels "01","02","03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ degree    : Factor w/ 2 levels "BA","MA": 1 2 2 1 1 1 1 2 1 1 ...
##  $ fte       : Factor w/ 2 levels "0.5","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ years     : num  5 15 16 10 26 28.5 12 32 25 12 ...
##  $ base      : int  45388 60649 60649 54466 65360 65360 58097 68230 65360 58097 ...
##  $ fica      : num  3472 4640 4640 4167 5000 ...
##  $ retirement: num  7689 10274 10274 9227 11072 ...
##  $ total     : num  56549 75563 75563 67859 81432 ...
```

Since `total` is a numerical variable, we can also summarize it using `favstats`:

```
favstats(teacher$total)
```

```
##       min       Q1   median       Q3      max     mean       sd  n missing
##  24793.41 63757.69 74646.85 81432.02 85007.76 70288.64 12439.34 71       0
```

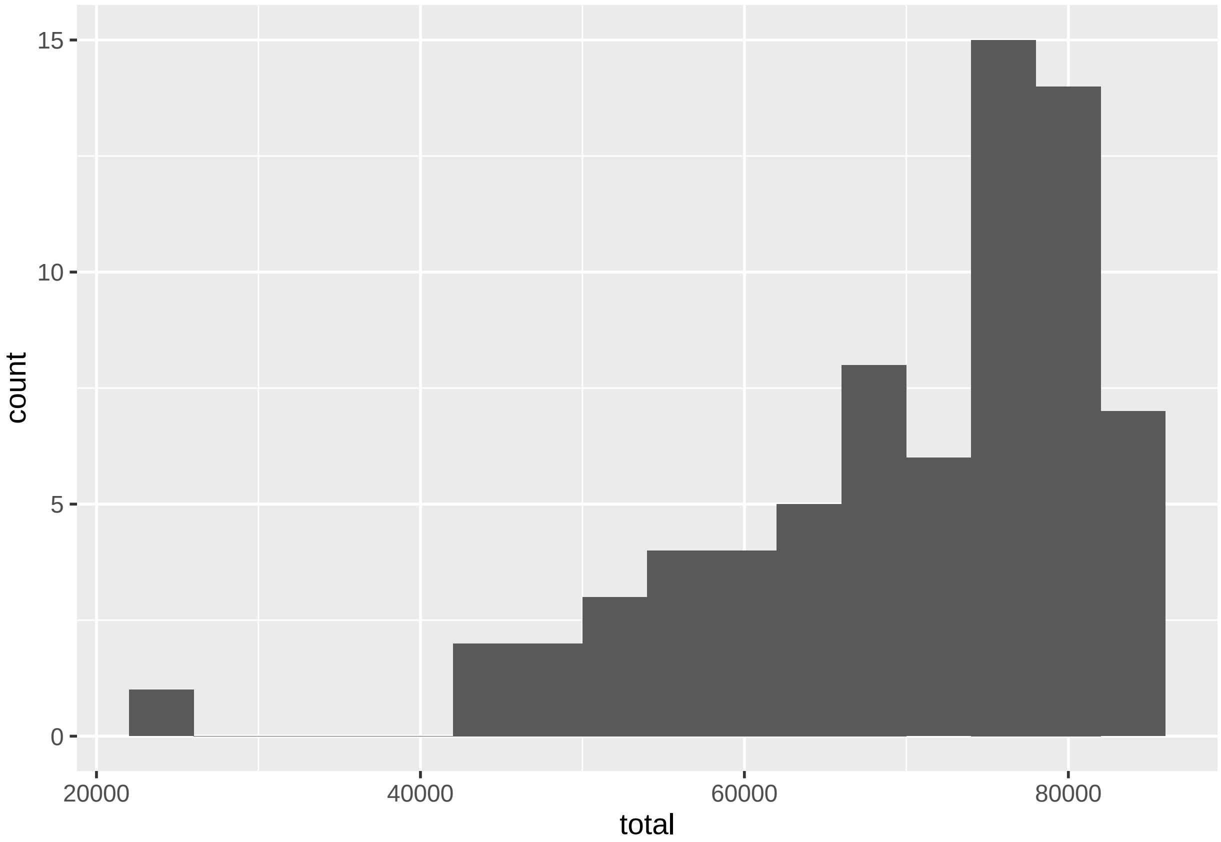
**Prepare the data for analysis.**

Not necessary here.

**Make tables or plots to explore the data visually.**

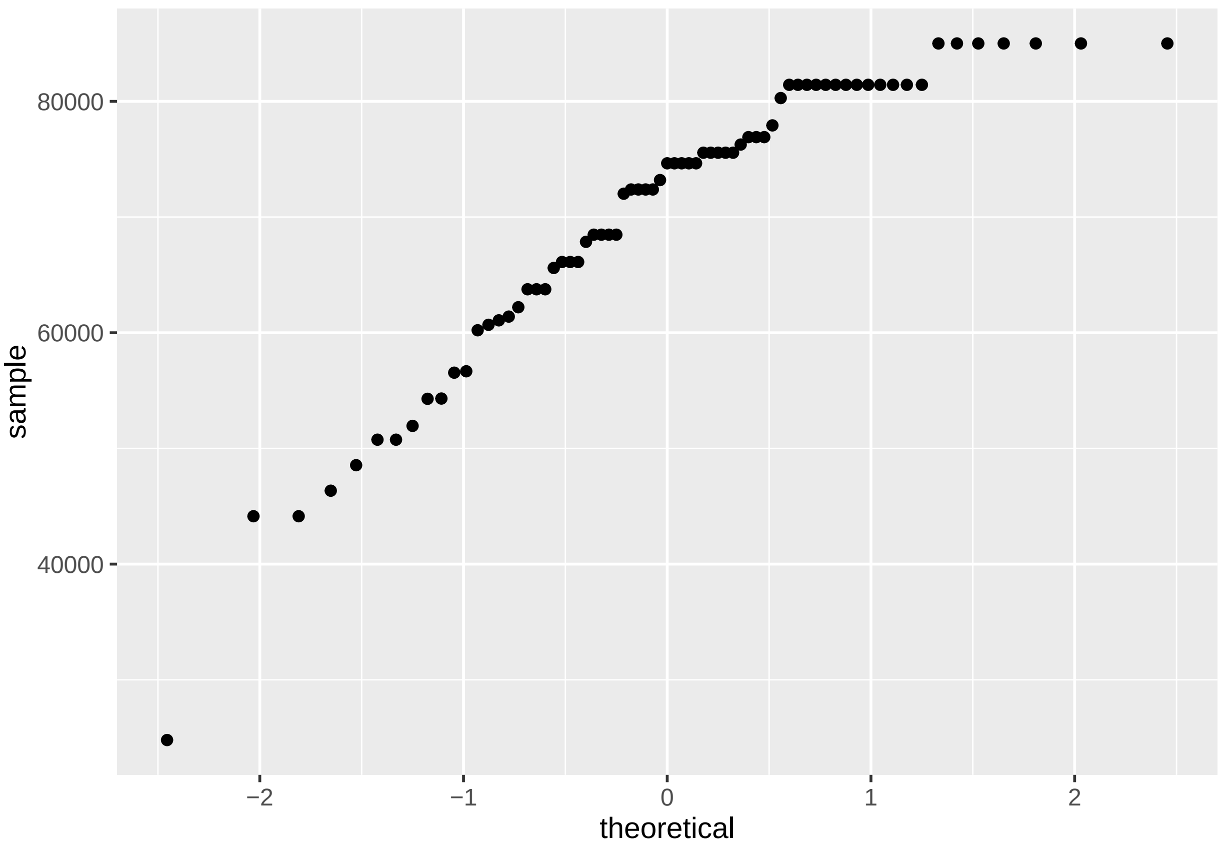
Here is a histogram.

```
ggplot(teacher, aes(x = total)) +
    geom_histogram(binwidth = 4000)
```



And here is a QQ plot.

```
ggplot(teacher, aes(sample = total)) +
    geom_qq()
```



This distribution is quite skewed to the left. Of even more concern is the extreme outlier on the left.

With any outlier, we need to investigate. Go to the Console and type

```
View(teacher)
```

(Be sure to use a capital "V" in `View`).

**Exercise**

In the spreadsheet view, you can use the arrows in the column headers to sort the data. Sort by `total` (ascending) and see if you can figure out how the person with the lowest total salary is different from all the other teachers. (Hint: you may need to Google "fte".)

Please write up your answer here.

---

Based on your answer to the above exercise, hopefully it's clear that this is an outlier for which we can easily justify exclusion. We can use the `filter` command to get only the rows we want. There are lots of ways to do this, but it's easy enough to grab only salaries above $40,000. (There's only one salary below $40,000, so that outlier will be excluded.)

```
teacher_2 <- filter(teacher, total > 40000)
```

Check to make sure this had the desired effect:

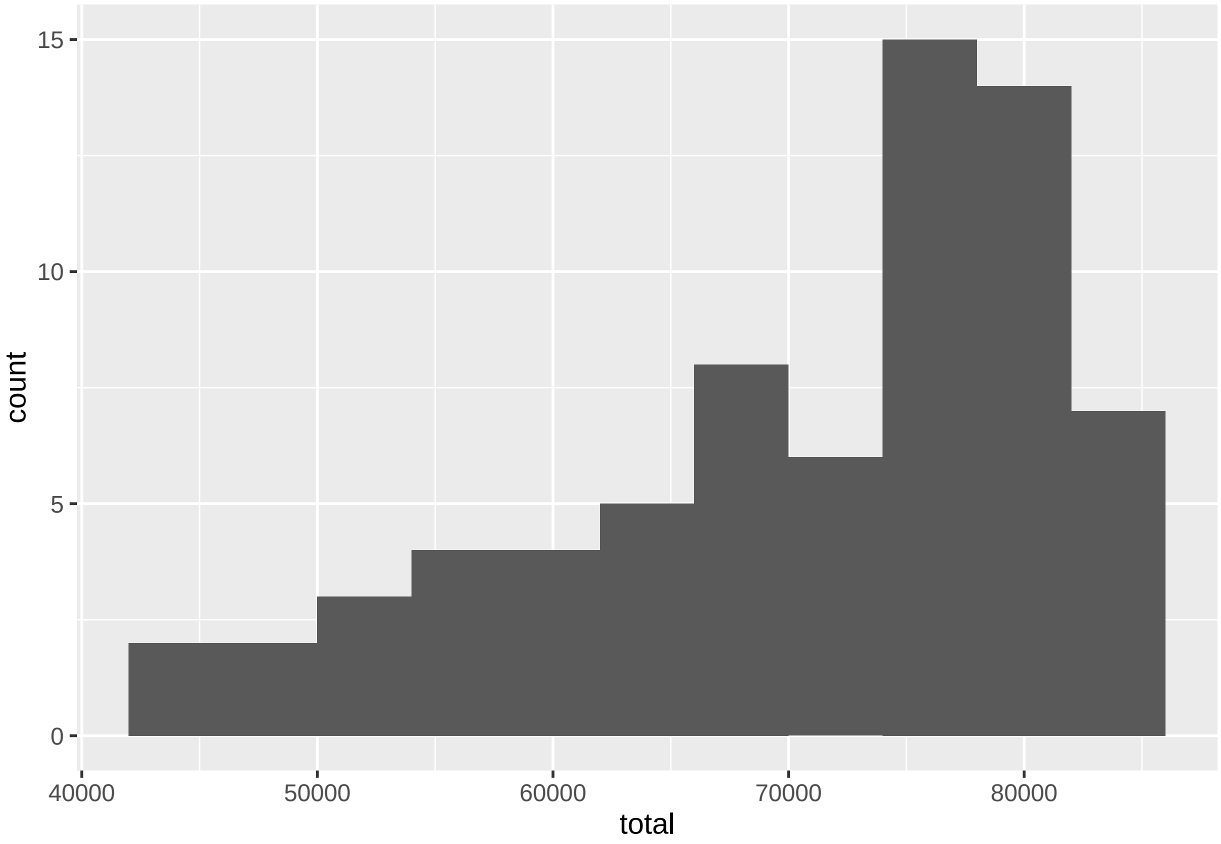
```
favstats(teacher_2$total)
```

```
##      min       Q1   median       Q3      max     mean       sd  n missing
##  44138.5 63757.69 74646.85 81432.02 85007.76 70938.57 11249.61 70       0
```

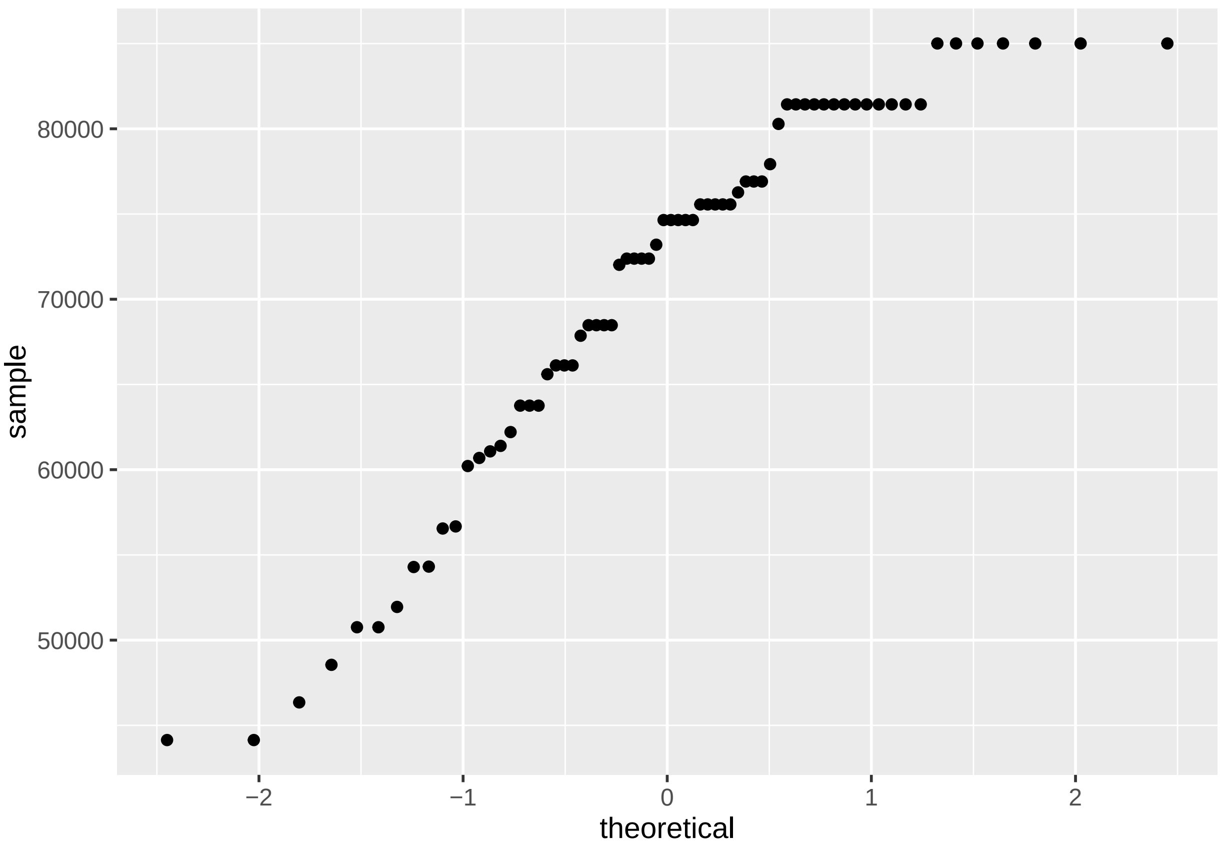
Notice how the min is no longer $24,793.41 and the sample size is 70 instead of 71.

Here are the new plots:

```
ggplot(teacher_2, aes(x = total)) +
    geom_histogram(binwidth = 4000)
```



```
ggplot(teacher_2, aes(sample = total)) +
    geom_qq()
```



The left skew is still present, but we have removed the outlier.

## Hypotheses

**Identify the sample (or samples) and a reasonable population (or populations) of interest.**

The sample consists of 70 teachers employed by the St. Louis Public School in Michigan. We are using these 70 teachers as a hopefully representative sample of all teachers in that region of Michigan.

**Express the null and alternative hypotheses as contextually meaningful full sentences.**

$H_0$: Teachers in the St. Louis region earn \$63,024 on average. (In other words, these teachers are the same as the teachers anywhere else in Michigan.)

$H_A$: Teachers in the St. Louis region do not earn \$63,024 on average. (In other words, these teachers are *not* the same as the teachers anywhere else in Michigan.)

**Express the null and alternative hypotheses in symbols.**

$H_0 : \mu = 63024$

$H_A : \mu \neq 63024$

## Model

### Identify the sampling distribution model.

We will use a t model with 69 degrees of freedom.

### Check the relevant conditions to ensure that model assumptions are met.

- Random
    - We know this isn't a random sample. We're not sure if this school is representative of other schools in the region, so we'll proceed with caution.
- 10%
    - This is also suspect, as it's not clear that there are 700 teachers in the region. One way to look at it is this: if there are 10 or more schools in the region, and all the school are about the size of the St. Louis Public School under consideration, then we should be okay.
- Nearly Normal
    - For this, we note that the sample size is much larger than 30, so we should be okay, even with the skewness in the data.

## Mechanics

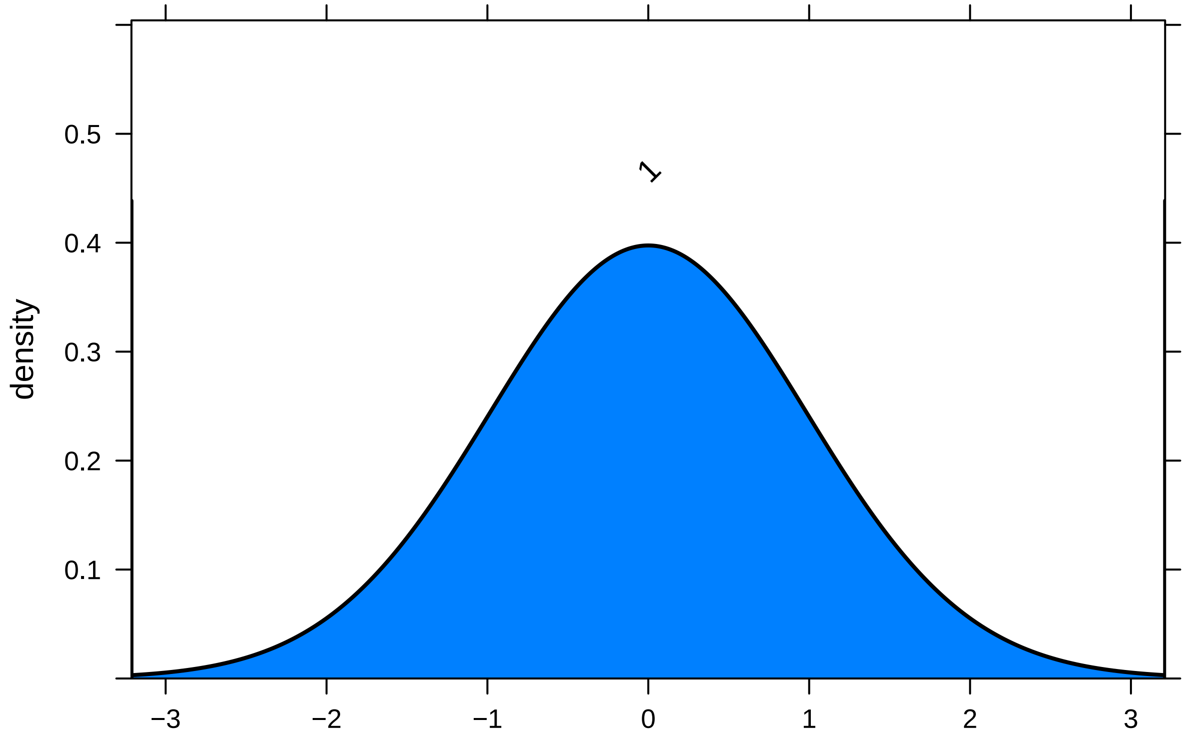
### Compute the test statistic.

```
total_test <- tidy(t.test(teacher_2$total,  mu = 63024))
t <- total_test$statistic
```

The t-score is 5.886253.

### Plot the null distribution.

```
pdist("t", df = total_test$parameter, q = c(-t, t))
```



```
## [1] 0.00000006454346 0.99999993545654
```

Commentary: The `pdist` command is the same as always except now we use a `"t"` model instead of `"norm"`. We know there are 69 degrees of freedom, but we might as well use the value stored in the `t.test` output to make our code reusable. Finally, note that `q = c(-t, t)` will plot in both tails of the distribution since we're running a two-sided test. Of course, since the t-score is crazy huge, it doesn't actually appear in the resulting graph.

**Calculate the P-value.**

```
total_test$p.value
```

```
## [1] 0.0000001290869
```

When the P-value is this small, it is traditional to report simply $P < 0.001$.

## Conclusion

**State the statistical conclusion.**

We reject the null hypothesis.

**State (but do not overstate) a contextually meaningful conclusion.**

There is sufficient evidence that teachers in the St. Louis region do not earn $63,024 on average.

**Identify the possibility of either a Type I or Type II error and state what making such an error means in the context of the hypotheses.**

If we've made a Type I error, then the truth is that teachers in this region do make around $63,024 on average, but our sample was way off.

## Confidence interval

### Conditions

All the conditions have been checked already.

### Calculation

```
total_test$conf.low
```

```
## [1] 68256.2
```

```
total_test$conf.high
```

```
## [1] 73620.95
```

### Conclusion

We are 95% confident that the true mean salary for teachers in the St. Louis region is captured in the interval ($68256.2, $73620.95).

Commentary: As these are dollar amounts, it makes sense to round them to two decimal places. Even then, R is finicky and sometimes it will not respect your wishes.)

## Your turn

In the High School and Beyond survey (the `hsb2` data set from the `openintro` package), among the many scores that are recorded are standardized math scores. Suppose that these scores are normalized so that a score of 50 represents some kind of international average. (This is not really true. I had to make something up here to give you a baseline number with which to work.) The question is, then, are American students different from this international baseline?

Follow the rubric, copying and pasting thoughtfully from the example above, to answer this question.

**Exercise**

After running inference above, answer the following questions:

1. Even though the result was *statistically* significant, do you think the result is *practically* significant? By this, I mean, are scores for American students so vastly different than 50? Do we have a lot of reason to brag about American scores based on your analysis?

Please write up your answer here.

2. What makes it possible for a small effect like this to be statistically significant even if it's not practically very different from 50? In other words, what has to be true of data to detect small but statistically significant effects?

Please write up your answer here.

---