T-tests and Power Analysis

Nicholas Kortessis

2025-03-19

T-tests: Tests about the mean of normally distributed populations

You've heard about t-tests. t-tests are tests about the **mean** of a population, μ. Moreover, they are tests about the mean of the population assuming the character you measure is **normally distributed.**

Just as we learned that t-distributions are useful for estimating the mean, they are also interested in estimating how different the mean is compared to a hypothesized value, or testing differences between means.

T-tests build on the basic idea that the sample mean of a normally distributed population is also normally distributed. That is, if the individual character, X_i, can be modeled as

$$X_i \sim \text{Normal}(\mu, \sigma^2)$$

then the sample mean, $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ can be modeled with the following distribution

$$\frac{\bar{X} - \mu}{SE_{\bar{X}}} \sim t_{df=n-1}.$$

This means that the standardized sample mean follows a t-distribution with $n-1$ degrees of freedom, assuming the population mean is μ. This is the central idea used to estimate means and create confidence intervals. It applies here as well.

Uses of the t-test

t-tests come in three flavors:

1. **One sample test**: a one-sample test evaluates evidence consistent with the hypothesis that the population mean is μ_0. Practically, we are asking, is the mean of the population μ_0 and we can calculate a p-value associated with that question.

2. **Two sample test**: a two-sample evaluates evidence consistent with the hypothesis that the difference in population means of two groups (i.e., samples) is a certain value, $\Delta\mu_0$.

   The triangle symbol here is the greek letter "Capital Delta" and is typically used to represent a difference. If we hypothesise that the population mean of one group is μ_{01} and the mean of the other group is μ_{02}, then their difference is $\Delta\mu_0 = \mu_{01} - \mu_{02}$. In nearly every circumstance, the null hypothesis here is $\Delta\mu_0 = 0$, meaning that the two groups have the same mean.

3. **Paired sample test:** Paired tests evaluate evidence consistent with the hypothesis that the population average difference in repeated measures of individuals is μ_0. Paired tests are useful when two measurements of statistical individuals are very closely related to one another. For example, you might measure a disease biomarker (e.g., blood pressure) on a single person both before and after a treatment. The difference in the measurements before and after constitute a paired difference. The same argument can be made for an experiment done in the greenhouse or in the field where experimental treatments

(e.g., treatment versus control) are closely positioned in space. Because they are so close in space, they share many similarities not shared by areas further apart in space. This is an argument for a paired test.

In each of these cases, t-tests use identical machinery to compare a sample mean to an expected value.

- In the case of a one-sample test, it compares a sample mean ($\bar{X}$) to a specific, hypothesized value, μ_0.
- In the case of a two-sample test, it compares the difference in sample means ($\Delta\bar{X} = \bar{X}_1 - \bar{X}_2$) to a hypothesized value ($\Delta\mu_0 = \mu_{01} - \mu_{02}$).
- And in the case of a paired sample test, it compared the mean of the differences of a paired values ($\overline{\Delta X}$; the long bar over the Delta X means to take an average of a difference in paired X's) with a hypothesized value.

Every single one of these can be described by a t-distribution. Let's go through these cases in order.

One sample t-test: by hand

First, let's start with an example where we do the entire t-test 'by hand' (with the computational help of a computer, of course). This will help you visualize what is going on, how the t-distribution is used, and how the p-value is calculated.

Question: Are A-list movie stars taller or shorter than the average person in the USA?

I'm curious about this question. I could see it two ways. On the one hand, I could argue that people tend to view celebrities as more attractive, and height is often associated with attractiveness in media, so movie stars should be taller than average. On the other hand, movies are a place where height can easily be masked using optical tricks and having huge disparities in heights of cast members might look odd. Based on those grounds, maybe movie stars are the same height as everyone else.

If we are going to answer this question, we need a reference against which to compare the heights of movie stars. Wikipedia says the average height of a male in the USA is 175.3 cm (5' 9") and the average height of a female in the USA is 163.1 cm (5' 3.5"). We will separate males from females because it is well know that they differ in height, so we'll use the gender-specific averages as null hypotheses.

Our null hypotheses are thus

$$H_{0,\text{Females}} : \mu_0 = 163.1cm$$

$$H_{0,\text{Males}} : \mu_0 = 175.3cm$$

What we need to do now is to think about the sampling design. Ideally, we would randomly sample from our population "A-list movie stars". I don't know how to do that well, so I asked ChatGPT to help in finding 30 random individuals from the population. Here is its response.

Query: what are the heights of 30 random a-list movie stars in cm?

ChatGPT Response: Here are the heights of 30 A-list movie stars, converted to centimeters:

1. **Robert Downey Jr.**: 173 cm (5'8") YouTube+3South China Morning Post+3Telegrafi+3
2. **Rami Malek**: 171 cm (5'7½") South China Morning Post
3. **Jeremy Allen White**: 170 cm (5'7") Telegrafi+1South China Morning Post+1
4. **Tom Cruise**: 170 cm (5'7")
5. **Kit Harington**: 168 cm (5'6") Celeb Heights

6. **Zac Efron**: 168 cm (5'6") British Vogue
7. **Daniel Radcliffe**: 165 cm (5'5") British Vogue
8. **Kevin Hart**: 158 cm (5'2¼") Telegrafi
9. **Kenny Baker**: 112 cm (3'8") Celeb Heights+4IMDb+4Telegrafi+4
10. **Peter Dinklage**: 132 cm (4'4") IMDb
11. **Danny DeVito**: 147 cm (4'10") IMDb
12. **Ben Affleck**: 193 cm (6'4")
13. **Jacob Elordi**: 196 cm (6'5") Reddit
14. **Alfred Enoch**: 193 cm (6'4") BuzzFeed
15. **Bill Skarsgård**: 193 cm (6'4") BuzzFeed
16. **Alexander Skarsgård**: 193 cm (6'4") BuzzFeed
17. **Winston Duke**: 196 cm (6'5") IMDb+8Reddit+8BuzzFeed+8
18. **Josh Duhamel**: 193 cm (6'4") South China Morning Post+2Telegrafi+2BuzzFeed+2
19. **Dwayne "The Rock" Johnson**: 196 cm (6'5")
20. **Armie Hammer**: 195 cm (6'4.75") Reddit
21. **Joe Manganiello**: 194 cm (6'4.25")
22. **Jared Padalecki**: 193 cm (6'4")
23. **Jason Momoa**: 192 cm (6'3.5")
24. **Liam Hemsworth**: 192 cm (6'3.5")
25. **Chris Hemsworth**: 191 cm (6'3") Reddit
26. **Ben Affleck**: 191 cm (6'3")
27. **Zachary Levi**: 191 cm (6'3")
28. **Tom Welling**: 189 cm (6'2.25") Reddit
29. **Ashton Kutcher**: 188 cm (6'2")
30. **Terry Crews**: 187 cm (6'1.5") Reddit

Please note that these heights are approximate and may vary slightly depending on the source.

Query: Great. Now give me 30 random male a-list actors and their heights and 30 random female a-list actresses and their heights, and compile them into a csv file for me.

> ChatGPT Response: Here's a CSV file containing the heights of 30 random male and 30 random female A-list actors and actresses. I'll generate and provide the file for you.
>
> I have compiled the data into a CSV file. You can download it using the link below:
>
> Download the file

Turns out you don't need to download the file. It's given in the assignment page. Load it into R

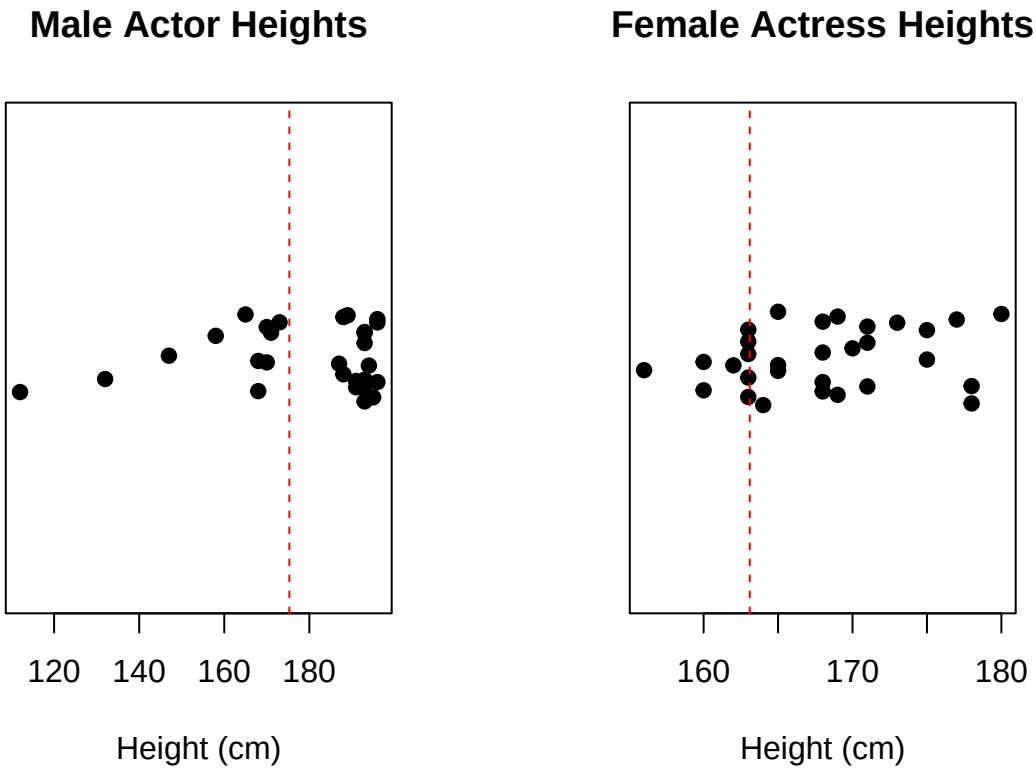
```
setwd("/Users/nicholaskortessis/Library/CloudStorage/GoogleDrive-kortessn@wfu.edu/My Drive/Import/Wake
heights.df <- read.csv(file = 'a_list_actors_heights.csv')
```

```
str(heights.df)
```

```
## 'data.frame':    60 obs. of  3 variables:
##  $ Name     : chr  "Robert Downey Jr." "Rami Malek" "Jeremy Allen White" "Tom Cruise" ...
##  $ Gender   : chr  "Male" "Male" "Male" "Male" ...
##  $ Height_cm: int  173 171 170 170 168 168 165 158 112 132 ...
```

Checkpoint 1: What are the statistical individuals in this data frame and what are the characteristics of each?

Checkpoint 2: Write code to show a 2-panel plot. One panel should show the distribution of heights in this sample for female actresses and one should show the distribution of heights in this sample for male actresses. You can use a stripplot, a histogram, a boxplot, or a density plot, whichever you prefer. In addition, show the US average height for each gender on each panel. Here is what my figures look like.



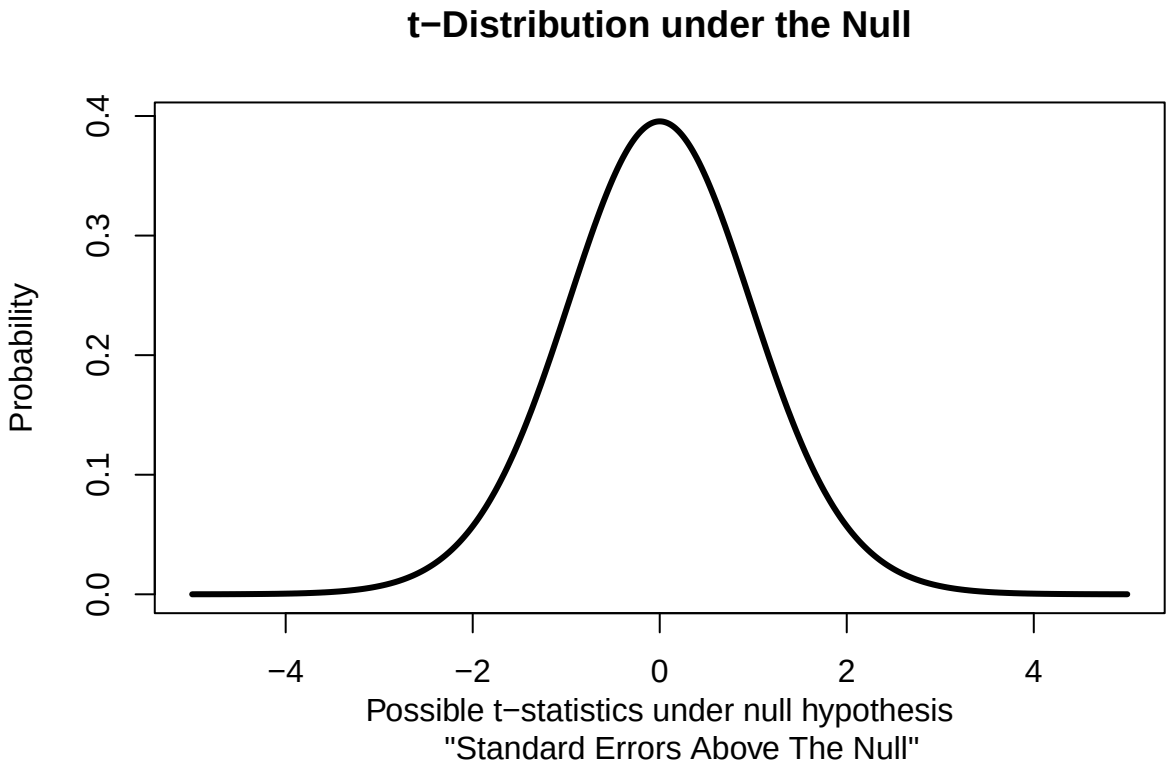
Now that we have looked at the data, let's think about to construct a test. To construct a t-test, we have to use a t-distribution under the null hypothesis, which says that distribution of the sample mean's difference from the null hypothesis, standardized by standard error, follows a t-distribution. Let's write this first for females.

$$\frac{\bar{X}_{\text{females}} - \mu_{0\text{females}}}{SE_{\bar{X}_{\text{females}}}} = \frac{\bar{X}_{\text{females}} - 163.1cm}{SE_{\bar{X}_{\text{females}}}} \sim t_{n-1}.$$

At this point, we could look at this distribution. All we need to do so is figure out how many degrees of freedom it has, as this is the only parameter that matters for a t-distribution. For this test, the degrees of freedom are $n-1$. Let's set that for the females in the data set and make the null sampling distribution.

```
n.females <- length(heights.df$Height_cm[heights.df$Gender == 'Female'])
t.values <- seq(from = -5, to = 5, length = 1000)
t.dist <- dt(t.values, df = n.females -1)
plot(t.values, t.dist,
```

```
     xlab = 'Possible t-statistics under null hypothesis
     "Standard Errors Above The Null"',
     ylab = 'Probability',
     main = 't-Distribution under the Null',
     typ = 'l', lwd = 3)
```

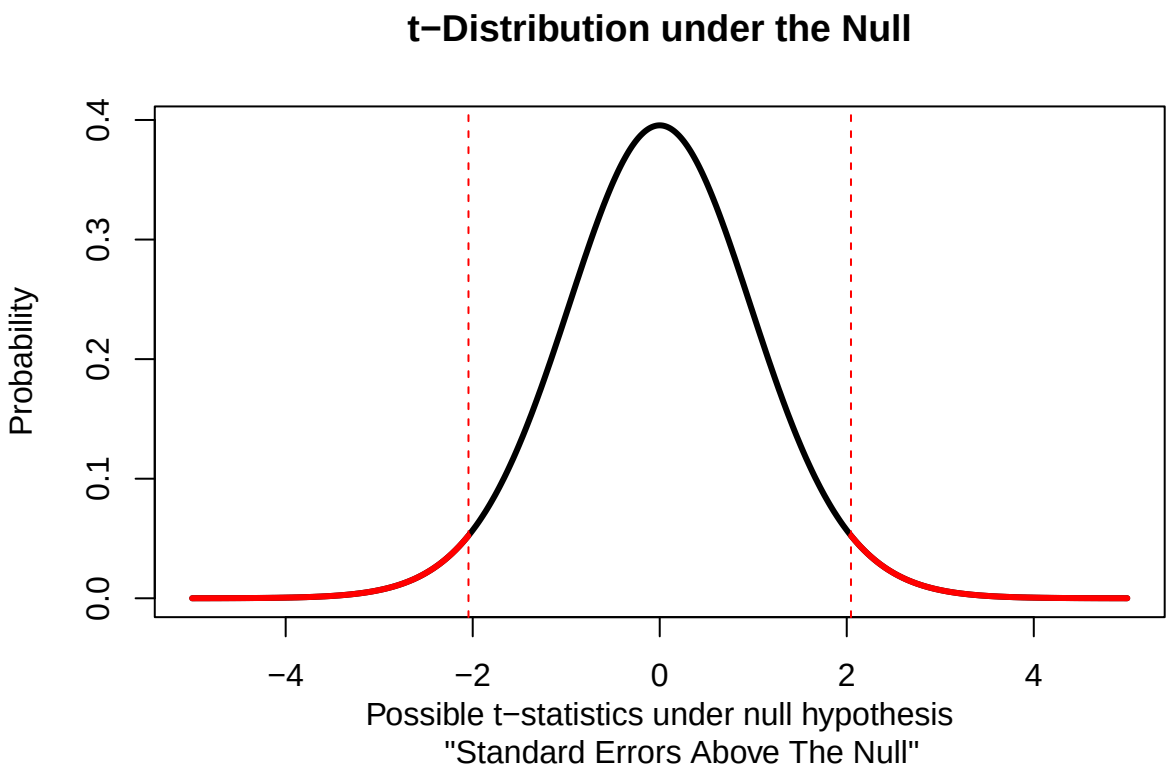


Given this, we can make a rejection region based on an α value. Let's choose $\alpha = 0.05$, which means we would reject the null hypothesis in 5% of samples taken from a distribution where the null hypothesis is true. With $\alpha = 0.05$, we can generate a rejection region that encompasses the 5% of least likely outcomes. Let's do that using quantiles and plot them on the same figure.

```
alpha <- 0.05
(crit.values <- qt(c(alpha/2, 1 - alpha/2), df = n.females - 1))
```

```
## [1] -2.04523  2.04523
```

```
plot(t.values, t.dist,
     xlab = 'Possible t-statistics under null hypothesis
     "Standard Errors Above The Null"',
     ylab = 'Probability',
     main = 't-Distribution under the Null',
     typ = 'l', lwd = 3)
abline(v = crit.values, lty = 2, col = 'red')
lines(t.values[t.values < crit.values[1]],
      t.dist[t.values < crit.values[1]],
      col = 'red', lwd = 3)
lines(t.values[t.values > crit.values[2]],
      t.dist[t.values > crit.values[2]],
      col = 'red', lwd = 3)
```



Our critical values that define the rejection region (the red areas) is as follows.

- If sample t-value $<$ -2.045, reject H_0.
- If sample -2.045 $<$ t-value $<$ 2.045, fail to reject H_0.
- If sample t-value $>$ 2.045, reject H_0.

The decision rule above corresponds to rejecting H_0 if the t-value falls in the red region. Another way to state it is that 5% of the least likely outcomes are more than 2.045 standard errors away from the null hypothesis of $\mu_{0\text{female}} = 163.1cm$.

All there is to do now is calculate the t-value for our data and see if it fall in the rejection region. To calculate our t-value ($t = (\bar{X} - \mu_0)/SE_{\bar{X}}$), we need simply by calculating the mean of female heights and the standard error of female heights and figure out how many standard errors our sample mean is from the hypothesis.

```
females <- subset(heights.df, subset = Gender == 'Female')
(Xbar.female <- mean(females$Height_cm))
```

```
## [1] 168.0333
```

```
(s.female <- sd(females$Height_cm))
```

```
## [1] 6.037146
```

```
(n.female <- length(females$Height_cm))
```

```
## [1] 30
```

```
(SE.Xbar.female <- s.female/sqrt(n.female))
```

```
## [1] 1.102227
```

Now the t-value is

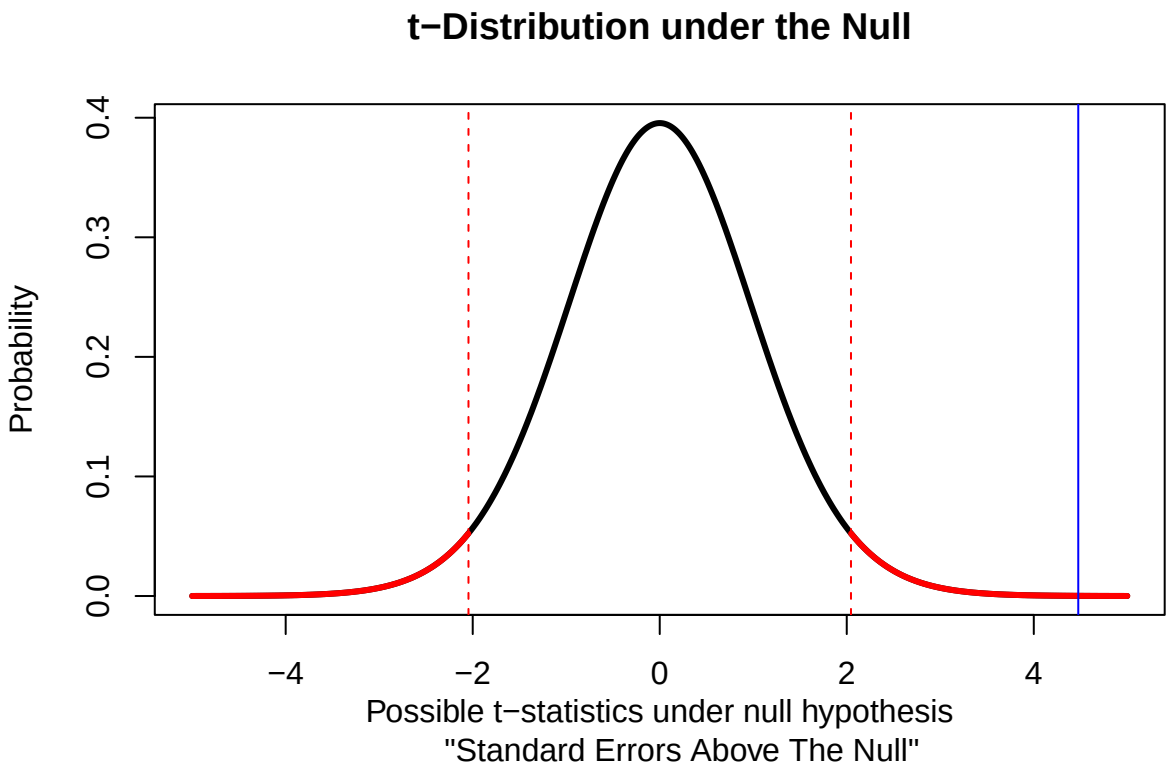
```
(t.female <- (Xbar.female - mu_0female)/SE.Xbar.female)
```

```
## [1] 4.475787
```

This says the average height of the A-list actresses in our sample is almost 4.5 standard errors higher than the US population average female value of $\mu_{0\text{female}} = 163.1cm$. This test says that it is highly unlikely that these women are a random sample from the female population with respect to height.

We can see how unlikely this is by plotting the mean of the actresses on the null distribution.

```
plot(t.values, t.dist,
     xlab = 'Possible t-statistics under null hypothesis
     "Standard Errors Above The Null"',
     ylab = 'Probability',
     main = 't-Distribution under the Null',
     typ = 'l', lwd = 3)
abline(v = crit.values, lty = 2, col = 'red')
lines(t.values[t.values < crit.values[1]],
      t.dist[t.values < crit.values[1]],
      col = 'red', lwd = 3)
lines(t.values[t.values > crit.values[2]],
      t.dist[t.values > crit.values[2]],
      col = 'red', lwd = 3)
abline(v = t.female, col = 'blue')
```



Visually, it is easy to see that this is an extremely unlikely occurrence if in fact the women were a random sample of height from the female population.

The last thing to do is to evaluate what the chances are of a more extreme effect size that this. The effect size here is best understood in terms of the t-value. We have a t-value of 4.476 so a more extreme outcome is t > 4.476. An equally extreme value is t < -4.476. We can calculate these probabilities with cumulative density functions using the R function `pt()`. Here is the math

$\Pr(t > 4.476) = 1 - \Pr(t < 4.476)$

`1 - pt(4.476, df = n - 1)`

$\Pr(t < 4.476)$

`pt(-4.476, df = n-1)`

The p-value is the sum of these two values. Let's calculate them.

```
p.value.female <- 1 - pt(t.female, df = n.female - 1) +
                  pt(-t.female, df = n.female - 1)
print(p.value.female)
```

```
## [1] 0.0001085592
```

This says that the p-value is 0.0001, which means that, if the null hypothesis is true (these actresses' heights are a random sample from the heights of all US females), then there is only a 0.01% chance of seeing heights this different from that expected by the null hypothesis.

Typically, this is particularly strong evidence against the null hypothesis. So what is the average height of the population of A-list actresses? We could estimate that number and quantify our uncertainty with confidence intervals. Let's generate a 99% confidence interval.

```
alpha <- 0.01
ci.t.values <- qt(c(alpha/2, 1-alpha/2), df = n.female - 1)
ci.Xbar.99 <- Xbar.female + ci.t.values*SE.Xbar.female
ci.Xbar.99
```

```
## [1] 164.9952 171.0715
```

Checkpoint 3: Now it's your turn. Make a sampling distribution, find the rejection region for $\alpha = 0.05$ for males, run a test to see if A-list actors have heights consistent with the broader US population, calculate a p-value, and estimate the 95% confidence interval for the average height of A-list actors.

One sample t-test: Using R

Here is the part where I tell you that we did this the hard way. R does all these things for you. But this has been instructive because everything you just calculated shows up in the test results. Let's do it again, easier this time.

All we really need to do for one sample test is to put the height data into the function `t.test()`. The default is assume $\mu_0 = 0$, but we can change that easily. Let's do this with the females.

```
t.test(females$Height_cm, # Sample data
       mu = mu_0female, # Null hypothesis
       conf.level = 0.99) # Level to make the confidence interval
```

```
##
##  One Sample t-test
##
## data:  females$Height_cm
## t = 4.4758, df = 29, p-value = 0.0001086
## alternative hypothesis: true mean is not equal to 163.1
## 99 percent confidence interval:
##  164.9952 171.0715
## sample estimates:
## mean of x
##  168.0333
```

Checkpoint 4: That was much better. Now write the code to let R run the t-test for you for males. If things are different than what you have in Checkpoint 3, double check your code for checkpoint 3.

Two-Sample t-test: Evaluating Whether Means of Two Groups are Different

Let's step away from heights of movie stars and think about more classical biological problems. One problem is whether vaccines provide similar protection as natural infections. In theory, vaccines work by exposing immune systems to signatures of an infection, and the goal is to produce a vaccine provides an immune signature of prior infection without the harmful effects of an actual infection.

In mid-2020, the first human trials of mRNA vaccines were running and in July 2020, the first reports of the immune responses of humans in mRNA vaccine trials were made public. We will use data from (Jackson et al. 2020) to illustrate t-tests.

The data we use here is not *exactly* the data from the preliminary report. However, it has the exact same statistical properties. The exact data isn't readily avaialble, but I have created synthetic data with exactly the same sample sizes, and distributions of outcomes. For all intents and purposes, you can think of this as the real study data.

This study had groups 15 participants who each received a single dose of the Moderna mRNA vaccine. The full study had 3 different groups that recieved different vaccine doeses. We will focus on the moderate does level, 100 μg. At different time points after each participant received the vaccine, a blood sample is taken and the blood is exposed to viral antigens (the virus parts that antibodies recognize). The study then measured **antibody titres**, which are a measure of immune activity in response to a particular antigen. The higher the antibody titre, the greater the presumed immune response.

We have data on the vaccinated group at two time points: 15 days and 57 days after vaccination. A note from the paper is that at 15 days, all individuals had evidence of *seroconversion*, which means the immune system recognized the vaccine as a foreign actor and had produced antibody memory consistent with the vaccine. In short, the vaccine had enough time to be working by day 15.

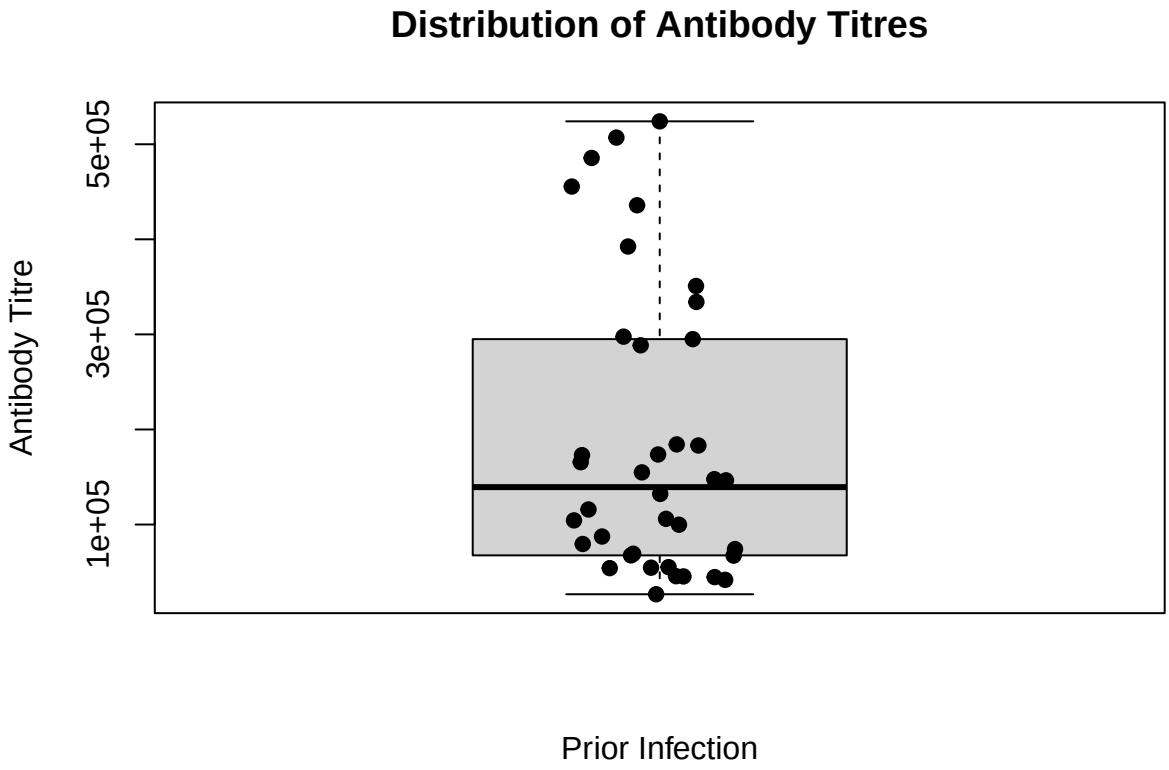
To compare the vaccine with typical immune responses after infection, the study also took blood from 38 individuals with known prior infection to SARS-CoV-2 and measured their antibody titres as well. Download the data from the file "COVID_mRNA.csv".

```
mrna.df <- read.csv(file = 'COVID_mRNA.csv')
str(mrna.df)
```

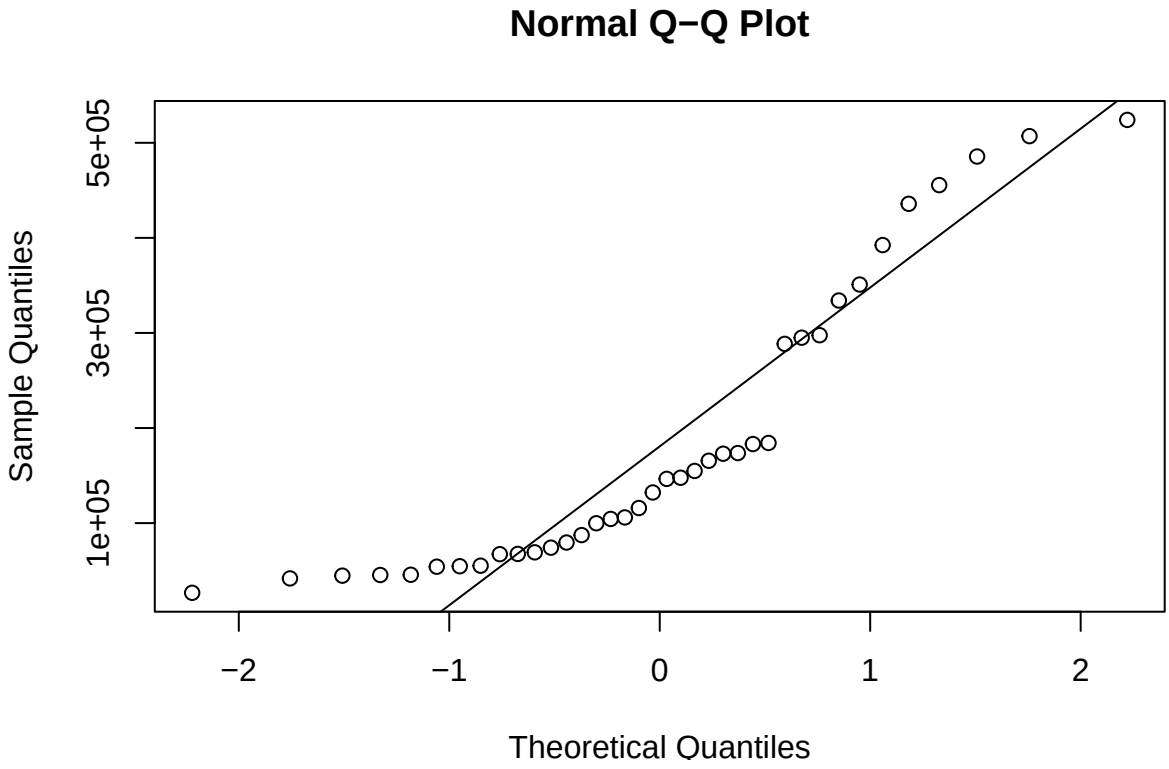
```
## 'data.frame':    68 obs. of  4 variables:
##  $ Individual.id : int  1865 1817 1566 1812 1348 1770 1011 1476 1363 1507 ...
##  $ Group         : chr  "Vaccine" "Vaccine" "Vaccine" "Vaccine" ...
##  $ Timepoint     : int  15 15 15 15 15 15 15 15 15 15 ...
##  $ Antibody.Titre: num  53342 99358 45427 293715 111133 ...
```

Antibody titres are well-known for NOT being normally distributed. Let's take a look at the distribution of each group.

```
mrna.df$Timepoint <- as.factor(mrna.df$Timepoint)
nat.inf.titre <- mrna.df$Antibody.Titre[mrna.df$Group == 'Natural Infection']
boxplot(nat.inf.titre,
        main = 'Distribution of Antibody Titres',
        xlab = 'Prior Infection',
        ylab = 'Antibody Titre')
stripchart(nat.inf.titre, add = T,
           vertical = T, pch = 19, method = 'jitter')
```

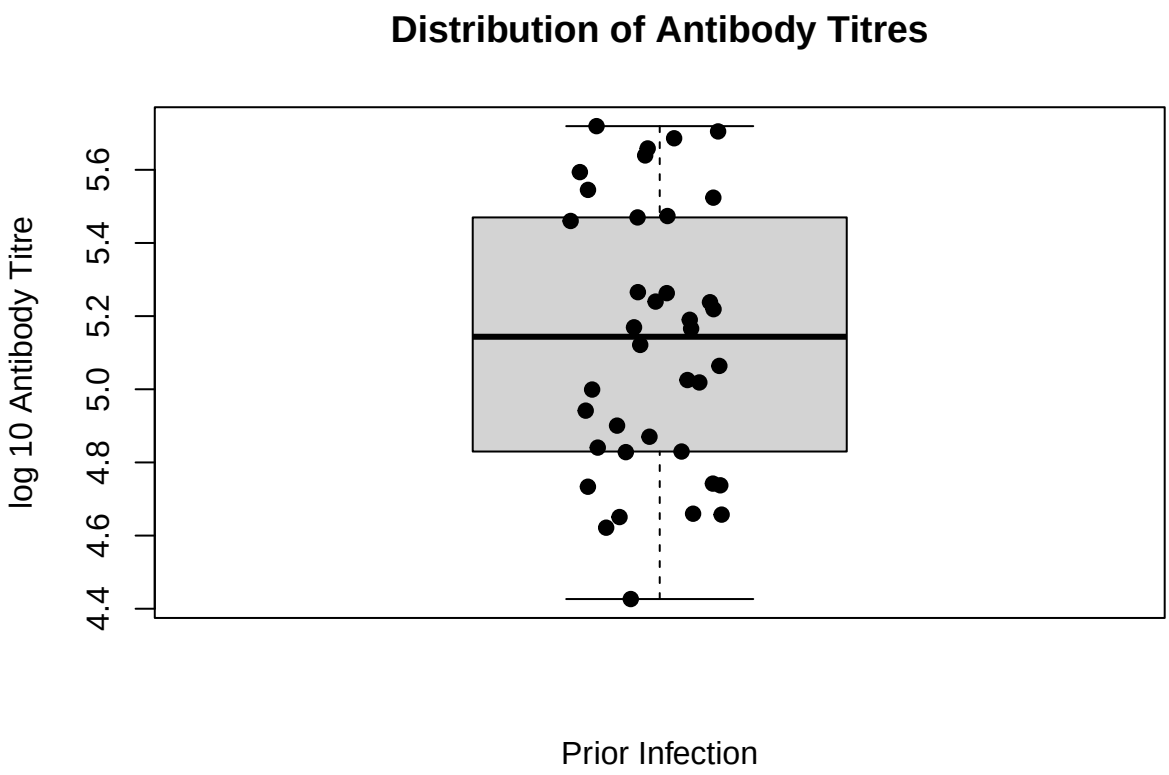



```
qqnorm(nat.inf.titre)
qqline(nat.inf.titre)
```

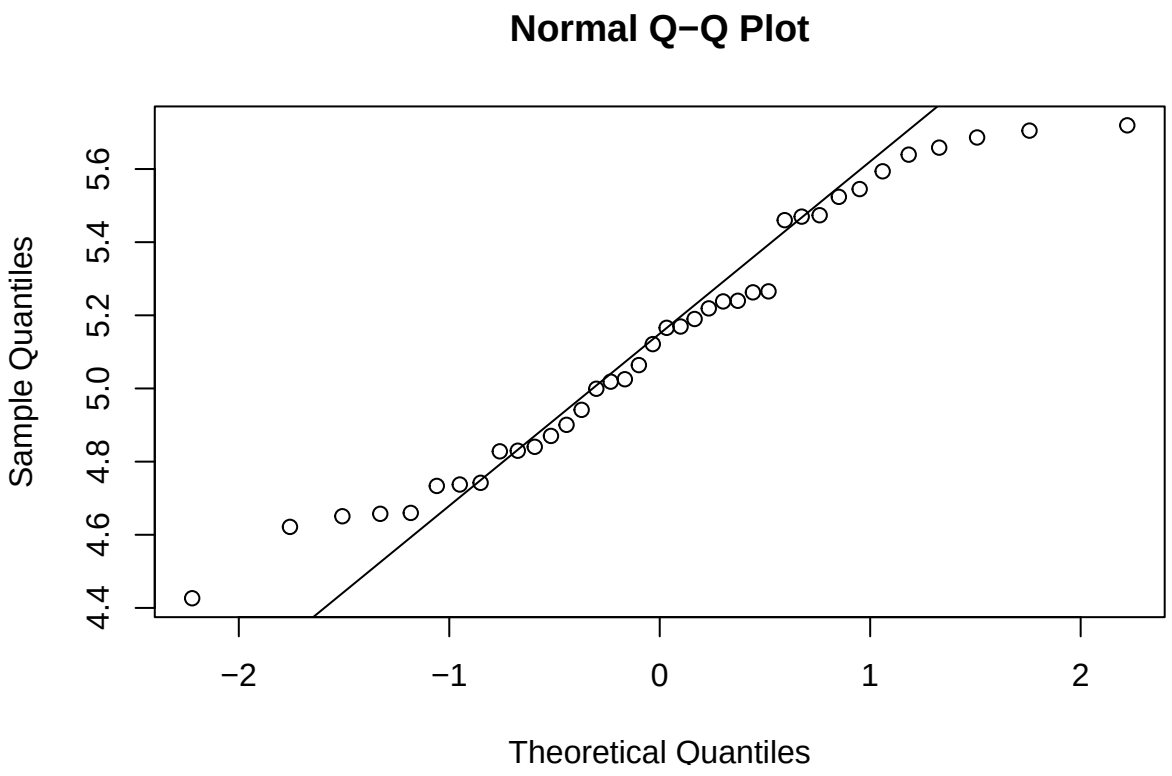


A way to resolve this is to take the logarithm of these values. You can take a natural log, a base 10 log, or a base 2 log. They all work. Which one you choose depends a bit on conventions in your field. I personally like the natural log, but let's stick with base 10 because it is easier to interpret.

```r
boxplot(log10(nat.inf.titre),
        main = 'Distribution of Antibody Titres',
        xlab = 'Prior Infection',
        ylab = 'log 10 Antibody Titre')
stripchart(log10(nat.inf.titre),
           add = T,
           vertical = T, pch = 19, method = 'jitter')
```

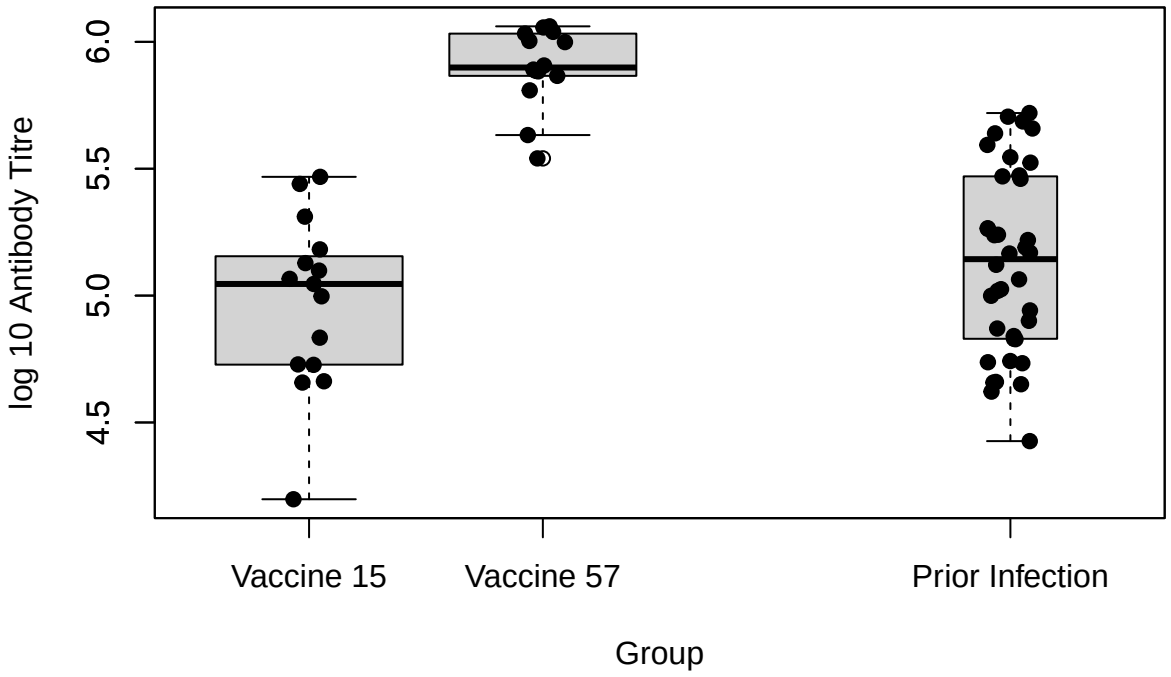



```r
qqnorm(log10(nat.inf.titre))
qqline(log10(nat.inf.titre))
```



It's not perfect, but much better. One of the difficulties of data analysis is knowing the art of when a model is good enough, even if not perfect!

```
boxplot(log10(Antibody.Titre) ~ Group + Timepoint, # Make the boxplot by group and time
        data = mrna.df, # Use this data.frame
        at = 1:2, # put the boxes at x-points 1 and 2.
        xlim = c(0.5, 4.5), # make axes big enough to put prior infection on
        xaxt = 'n', # Remove x-ticks. We will put them on manually
        xlab = 'Group',
        main = 'Antibody Titre Responses in Moderna Vaccine Trial',
        ylab = 'log 10 Antibody Titre')

stripchart(log10(Antibody.Titre) ~ Group + Timepoint, # Do the same with stripchart
           data = mrna.df,
           at = 1:2, vertical = T, pch = 19,
           add = T, method = 'jitter')
boxplot(log10(nat.inf.titre),
        at = 4,
        vertical = T, add = T)
stripchart(log10(nat.inf.titre),
        at = 4,
        vertical = T,
        pch = 19,
        add = T,
        method = 'jitter')

axis(1, at = c(1,2,4),
     labels = c("Vaccine 15", "Vaccine 57", "Prior Infection"))
```



So that is a nice visual of our data. Let's answer the question: **Does this vaccine mount a similar immune response to natural infection?**

To answer this question, we can pose it as whether the *average immune response is the vaccinated individuals and individuals with prior infection.* That is, we want to make this comparison,

$$\mu_{\text{Vaccinated}} - \mu_{\text{Prior Infection}}$$

Let's set our null hypothesis that the immune responses are identical. All that is needed now is to pick two groups and throw them into the `t.test` function in R. Let's use the 15 day marker for the vaccinated group, since this is the first time the study detects the presence of immune memory.

```
vaccine.15 <- subset(mrna.df, subset = (Group == 'Vaccine' & Timepoint == "15"))
t.test(log10(vaccine.15$Antibody.Titre), # log titres of vaccine group
       log10(nat.inf.titre), # log titres of natural infection group
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  log10(vaccine.15$Antibody.Titre) and log10(nat.inf.titre)
## t = -1.5095, df = 27.294, p-value = 0.1427
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.37544337  0.05708363
## sample estimates:
## mean of x mean of y
##  4.969592  5.128772
```

Checkpoint 4: Using the model output above, answer the original question of whether there is evidence that vaccinated antibody responses are different from natural infection at 15 days after infection. Use test outputs to justify your answer.

Paired-Sample t-test

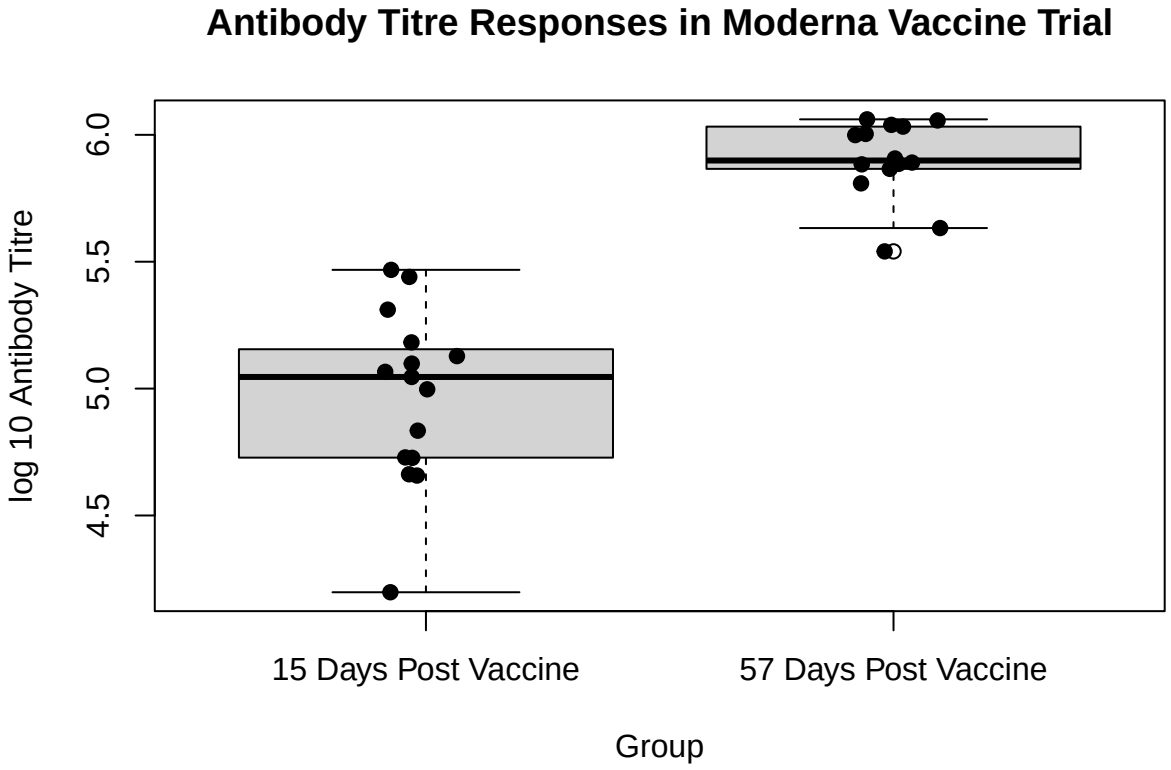
The last kind of t-test is a paired sample t-test. The mRNA vaccine study has some groups that are well suited for a paired sample t-test because some individuals in the study had their blood drawn multiple times at different time points. As such, the antibody titres at two time points are paired together since they come from the same individual.

The reason for pairing up the individuals is that it helps identify how statistical individuals change either over time or space (or some other feature). And it is the change that we are interested in.

A natural question to ask with this data is: **Do individuals mount stronger immune responses with more time since vaccination?**

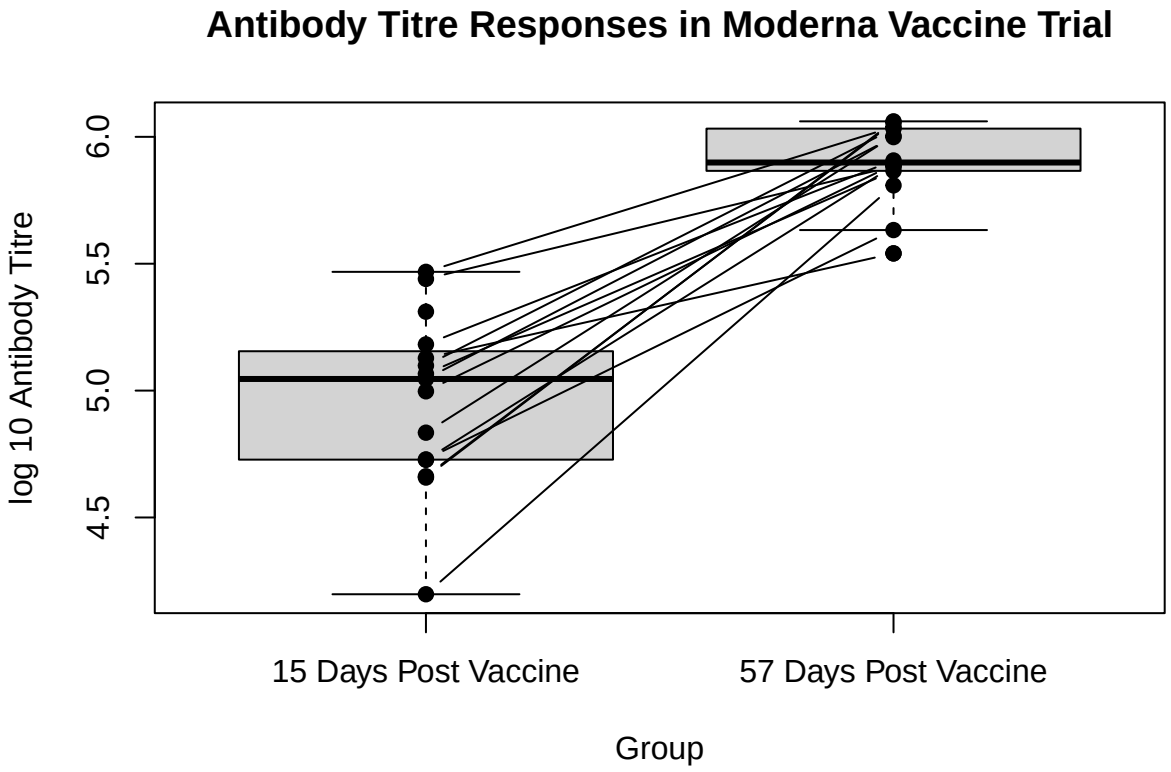
To answer this question, we need to figure out *how much an individual's antibody response changed over time.* To do this, we will look at time points 15 and 57 in the vaccinated group. Here is the figure we made before.

```
boxplot(log10(Antibody.Titre) ~ Group + Timepoint, # Make the boxplot by group and time
        data = mrna.df, # Use this data.frame
        at = 1:2, # put the boxes at x-points 1 and 2.
        xaxt = 'n', # Remove x-ticks. We will put them on manually
        xlab = 'Group',
        main = 'Antibody Titre Responses in Moderna Vaccine Trial',
        ylab = 'log 10 Antibody Titre')

stripchart(log10(Antibody.Titre) ~ Group + Timepoint, # Do the same with stripchart
           data = mrna.df,
           at = 1:2, vertical = T, pch = 19,
           add = T, method = 'jitter')
axis(1, at = c(1,2),
     labels = c("15 Days Post Vaccine", "57 Days Post Vaccine"))
```



This picture makes it look like yes, they mount a stronger response 57 days after the vaccine rather than 15 days. But maybe some individuals had much stronger responses than others. To see that, we can pair up the point as follows.

```
boxplot(log10(Antibody.Titre) ~ Group + Timepoint, # Make the boxplot by group and time
        data = mrna.df, # Use this data.frame
        at = 1:2, # put the boxes at x-points 1 and 2.
        xaxt = 'n', # Remove x-ticks. We will put them on manually
        xlab = 'Group',
        main = 'Antibody Titre Responses in Moderna Vaccine Trial',
        ylab = 'log 10 Antibody Titre')

axis(1, at = c(1,2),
     labels = c("15 Days Post Vaccine", "57 Days Post Vaccine"))

# Identify every individual
ids <- unique(mrna.df$Individual.id)

# Loop over every individual
for (i in 1:length(ids)){
  # Extract data from focal individual
  ind.df <- subset(mrna.df, subset = Individual.id == ids[i])

  # Plot points for each time interval on the log scale
  points(c(1,2),
       log10(c(ind.df$Antibody.Titre[ind.df$Timepoint == '15'],
               ind.df$Antibody.Titre[ind.df$Timepoint == '57'])),
       pch = 19, typ = 'b') # Use 'b' to plot both points and lines
} # This signals the end of loop, i.e., do it again for the next individual in the list.
```



Checkpoint 5: What does this plot show you that the other plot without the lines does not? In a paired t-test, it is these lines that are what we are studying. Each line represents a *difference within an individual over time.* When we do a paired t-test, we are asking whether the **distribution of individuals differences is, on average, some amount given by the null hypotheses (typically zero).**

Let's calculate these differences. First, we need to break up each group and make sure the values are paired up by individual. To do that, we will separate the groups and order them by the ids. When we take the difference, this will then allow us to make sure we are comparing the same individuals.

```
# Separate the groups
day.15 <- subset(mrna.df, subset = Timepoint == '15')
day.57 <- subset(mrna.df, subset = Timepoint == '57')

# Order by id
day.15[order(day.15$Individual.id),]
```

```
##    Individual.id   Group Timepoint Antibody.Titre
## 7           1011 Vaccine        15      125459.34
## 11          1192 Vaccine        15      275474.98
## 12          1293 Vaccine        15      116402.45
## 5           1348 Vaccine        15      111132.93
## 9           1363 Vaccine        15      134265.62
## 8           1476 Vaccine        15      152113.00
## 10          1507 Vaccine        15       68254.47
## 3           1566 Vaccine        15       45427.40
## 14          1623 Vaccine        15       15756.31
## 6           1770 Vaccine        15       45959.27
## 4           1812 Vaccine        15      293714.80
## 2           1817 Vaccine        15       99358.03
## 1           1865 Vaccine        15       53341.96
```

```
## 15          1956 Vaccine        15      204683.84
## 13          1989 Vaccine        15       53555.91
```

```
day.57[order(day.57$Individual.id),]
```

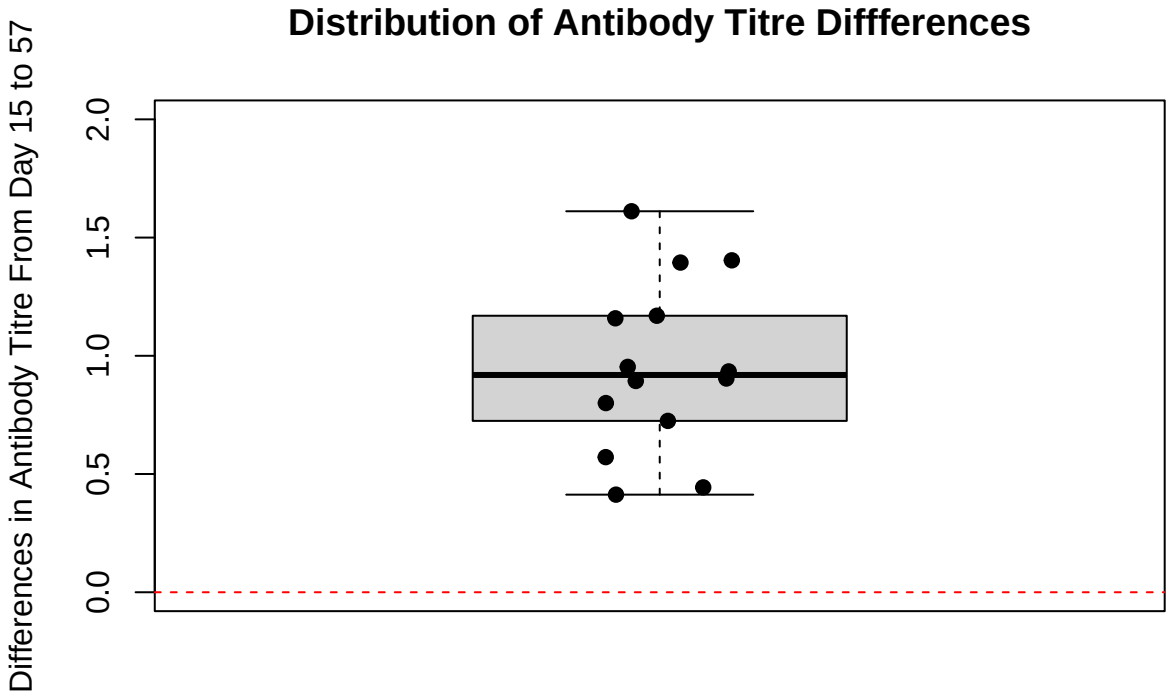
```
##    Individual.id   Group Timepoint Antibody.Titre
## 22          1011 Vaccine        57      1077380.0
## 26          1192 Vaccine        57       764975.9
## 27          1293 Vaccine        57       734455.3
## 20          1348 Vaccine        57       997638.4
## 24          1363 Vaccine        57       347267.5
## 23          1476 Vaccine        57       806928.1
## 25          1507 Vaccine        57      1008260.0
## 18          1566 Vaccine        57      1150951.4
## 29          1623 Vaccine        57       643835.1
## 21          1770 Vaccine        57      1139322.4
## 19          1812 Vaccine        57      1094720.4
## 17          1817 Vaccine        57       777559.2
## 16          1865 Vaccine        57       768482.5
## 30          1956 Vaccine        57             NA
## 28          1989 Vaccine        57       429211.8
```

```
# Now that they are ordered, we can take their differences
differences <- log10(day.57$Antibody.Titre) - log10(day.15$Antibody.Titre)
# And let's add their individual ids to make it clearer who had what repsonse.
ind.diff.df <- data.frame(day.15$Individual.id[order(day.15$Individual.id)], differences)
ind.diff.df
```

```
##    day.15.Individual.id.order.day.15.Individual.id.. differences
## 1                                               1011   1.1585650
## 2                                               1192   0.8935305
## 3                                               1293   1.4037392
## 4                                               1348   0.5713774
## 5                                               1363   0.9531304
## 6                                               1476   1.3942735
## 7                                               1507   0.9338659
## 8                                               1566   0.7246685
## 9                                               1623   0.4126994
## 10                                              1770   1.1694415
## 11                                              1812   0.4435656
## 12                                              1817   0.8000032
## 13                                              1865   0.9038643
## 14                                              1956   1.6113202
## 15                                              1989          NA
```

Now that we have the table of differences, let's see what the distribution of differences looks like.

```
boxplot(differences, ylab = 'Differences in Antibody Titre From Day 15 to 57',
        main = 'Distribution of Antibody Titre Diffferences', vertical = T,
        ylim = c(0,2))
stripchart(differences, vertical = T, pch = 19, method = 'jitter', add = T)
abline(h = 0, col = 'red', lty = 2)
```



I put the null hypothesis that there are zero differences in log-titre over time as the horizontal dashed red line. This histogram and stripchart shows **the distribution of the slopes of each of the lines in the figure with both groups.**

To do a paired test, we simply need to put in both groups and specify that the two groups are paired, and not a two-sample test. Like this,

```
t.test(log(day.57$Antibody.Titre), log(day.15$Antibody.Titre),
       paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(day.57$Antibody.Titre) and log(day.15$Antibody.Titre)
## t = 9.8865, df = 13, p-value = 2.054e-07
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  1.718976 2.680292
## sample estimates:
## mean difference
##        2.199634
```

Or we could throw the differences into a one-sample test, like this

```
t.test(ind.diff.df$differences)
```

```
##
##  One Sample t-test
##
## data:  ind.diff.df$differences
## t = 9.8865, df = 13, p-value = 2.054e-07
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.7465417 1.1640361
```

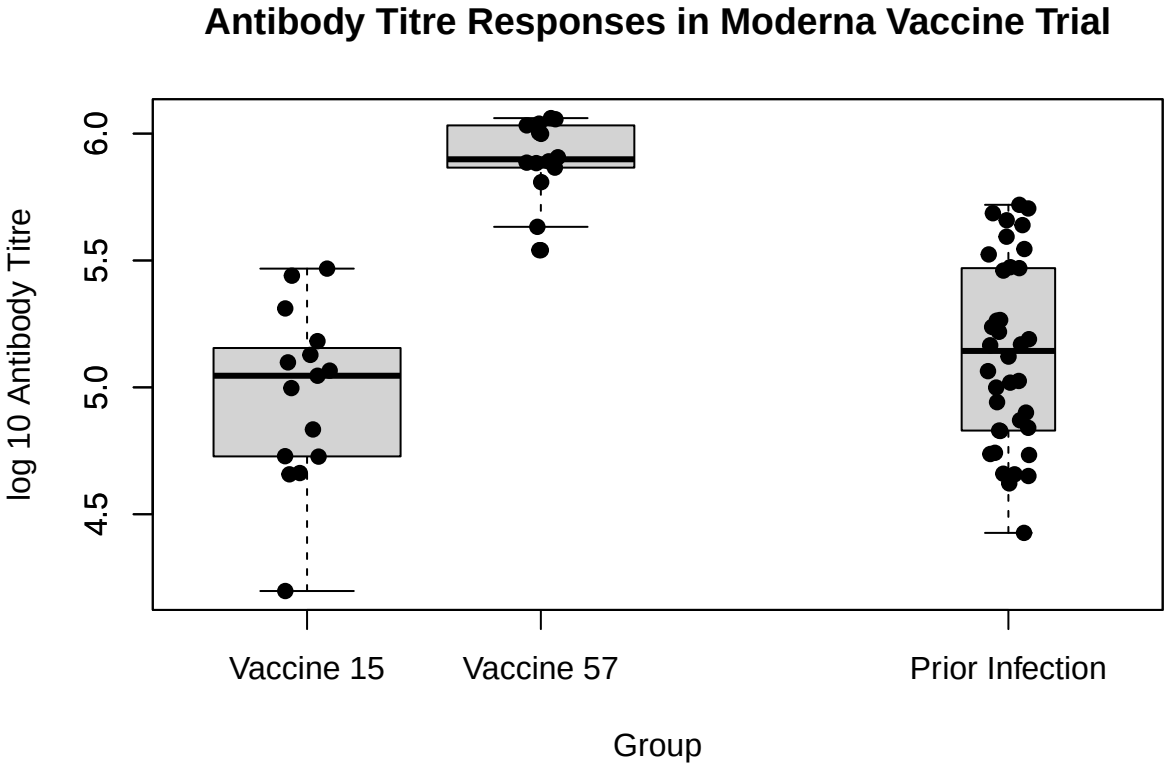
```
## sample estimates:
## mean of x
## 0.9552889
```

Note how the t-values, degrees of freedom, and p-values are all the same in either method.

Checkpoint 6: What does this test tell you about the question of whether individuals mount a stronger immune response longer after vaccination? If it is stronger, how much stronger? Us pieces of evidence from the test output to justify your answer.

Power Analysis

Let's do a quick power analysis on the two-sample t-test. Here was our plot and test output.




```
##
##  Welch Two Sample t-test
##
## data:  log10(vaccine.15$Antibody.Titre) and log10(nat.inf.titre)
## t = -1.5095, df = 27.294, p-value = 0.1427
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.37544337  0.05708363
## sample estimates:
## mean of x mean of y
##  4.969592  5.128772
```

This test didn't have enough evidence at the $\alpha = 0.05$ to reject the null hypothesis that these two groups come from populations with different means. However, the **samples** have different means. That difference is called an **effect size**. Here is the effect size:

```r
two.sample.mdl <- t.test(log10(vaccine.15$Antibody.Titre), # log titres of vaccine group
       log10(nat.inf.titre), # log titres of natural infection group
       conf.level = 0.95)
two.sample.mdl$estimate
```

```
## mean of x mean of y
##  4.969592  5.128772
```

```r
(effect.size <- two.sample.mdl$estimate[2] - two.sample.mdl$estimate[1])
```

```
## mean of y
## 0.1591799
```

This means the means of the samples differed by 0.16 titres on the log10 scale. Alternatively, one could say that prior infected individuals have, on average, $10^{0.16} = 1.45$ times the titres as the vaccinated group.

A power analysis answers the question: **How large would our sample need to be to reject the null hypothesis if the difference in the populations is 0.16?**

To do this, let's just mimic the sampling process for different sample sizes. To do this, we need some populations to represent the alternative, we need a sampling protocol, and we need the test.

The test is just the two-sample t-test under the null.

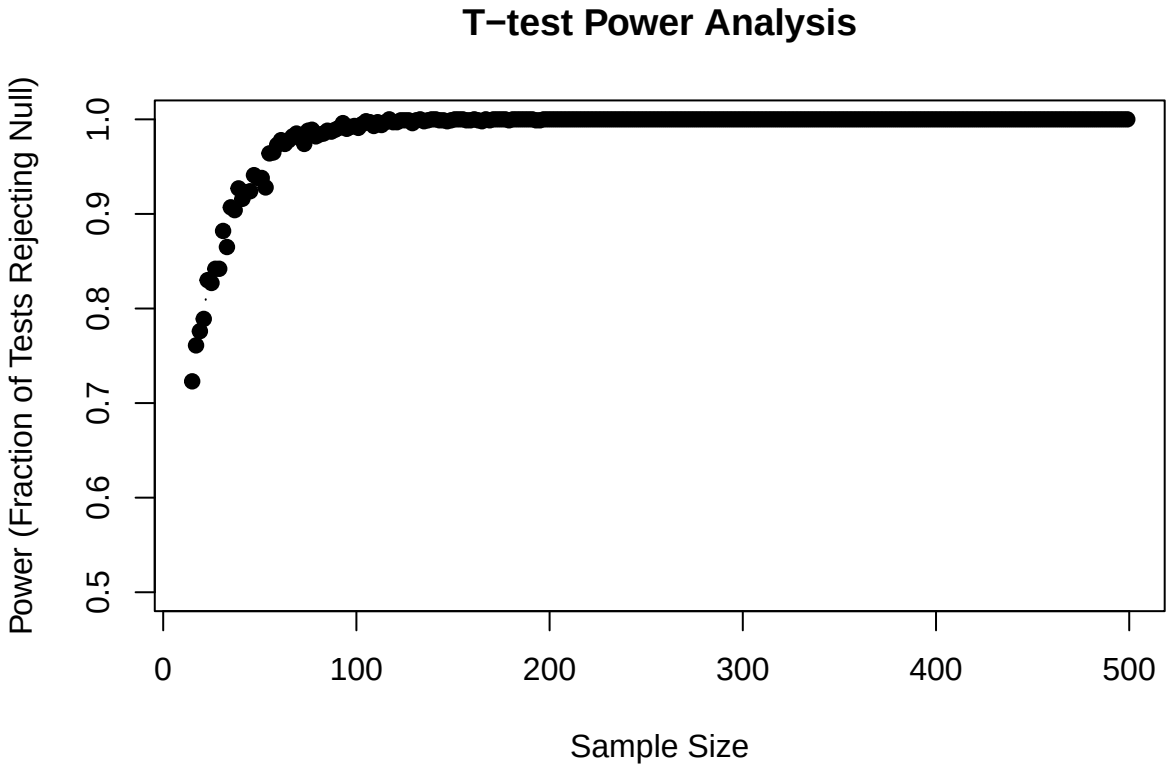
The sampling alternative is to take 15 random individuals from the

```r
sd.vaccine <- sd(log10(vaccine.15$Antibody.Titre))
sd.prior <- sd(log10(nat.inf.titre))
```

The sampling protocol we will pick is random sampling with some specified sampling size, n. Let's go ahead and make the populations. We need two, one with mean of group A and one with mean of group B

```r
# Make Vaccine Population
mu.vaccine <- two.sample.mdl$estimate[1]
sigma.vaccine <- sd.vaccine

# Make Prior Infection Population
mu.prior <- two.sample.mdl$estimate[2]
sigma.prior <- sd.prior

# Set our Type I error probability
alpha <- 0.5

# Here are the sample sizes to test. We will assume
# we have the same sample size in each group.
sample.sizes <- seq(from = 15, to = 500, by = 2)

# Here is the number of times we repeat sampling the
# population and running a test. Each time we can ask
# whether we reject the test. Remember, this is just
# like building a sampling distribution under the null
# but this time we are doing it under an alterantive
# where the populations actually differ by 0.16.
tests <- 1000

# Here we record, for a given sample size, how many
# of the tests we reject.
fraction.tests.rejected <- rep(NA, length(sample.sizes))
```

```r
# 'Loop' over different sample sizes
for (i in 1:length(sample.sizes)){

  # Create a place to store different test p-values
  test.p <- rep(NA, tests)
  # Set the sample size for this part of the loop
  n <- sample.sizes[i]

  # 'Loop' over repeated samples and tests (like bootstrapping)!
  for (j in 1:tests){
    # Make vaccine sample
    vaccine.sample <- rnorm(n, mu.vaccine, sigma.vaccine)
    # Make prior infection sample
    prior.sample <- rnorm(n, mu.prior, sigma.prior)

    # Run the test on this data
    boot.mdl <- t.test(vaccine.sample, prior.sample)

    # Store p-value
    test.p[j] <- boot.mdl$p.value
  }
  # Determine what fraction of tests reject null by counting
  # the fraction that have a p-value below alpha.
  fraction.tests.rejected[i] <- sum(test.p < alpha)/tests
}

# Now plot the results
plot(sample.sizes, fraction.tests.rejected, pch = 19,
     typ = 'b', xlab = 'Sample Size',
     ylab = 'Power (Fraction of Tests Rejecting Null)',
     main = 'T-test Power Analysis',
     ylim = c(0.5,1))
```



Here is our guess of the power analysis. We had about a 75% chance of detecting an effect as large as we saw in the population. But we didn't, which is interesting. If you want to make the claim that there are actually differences between vaccinated and prior infected individuals but that you need more individuals to tell the difference, I would respond that you had pretty high power in the first place!

Checkpoint 7: Now it's your turn. Run a power analysis for this study with an effect size of 0.0016, 0.016, and 1.6 log antibody titre differences. What are the sample sizes needed to detect a difference with 90% chance in each of these cases?

References

Jackson, Lisa A., Evan J. Anderson, Nadine G. Rouphael, Paul C. Roberts, Mamodikoe Makhene, Rhea N. Coler, Michele P. McCullough, et al. 2020. "An mRNA Vaccine Against SARS-CoV-2 — Preliminary Report." *New England Journal of Medicine* 383 (20): 1920–31. https://doi.org/10.1056/NEJMoa2022483.