More On Hypothesis Testing

```
library(fastR2)
library(tidyr)
library(resampledata)
```

Introduction

In a previous notebook, we introduced the idea and terminology of statistical hypothesis testing which is an important theme in inferrential statistics. In particular, we considered permutation tests and applied this technique to test a difference of means in the beer and hotwings data set. In this notebook, we will expand on our permutation testing experience and consider some additional facets of hypothesis tests.

One-sided and Two-sided Tests

For the beer wings data, we tested the null hypothesis

$H_0 : \mu_M - \mu_F = 0$

versus the alternative hypothesis

$H_A : \mu_F - \mu_F > 0$

This is an example of a one-sided test. Another option would be to conduct a two-sided test which takes the form

$H_0 : \mu_M - \mu_F = 0$

versus

$H_A : \mu_F - \mu_F \neq 0$

Sometimes this might be written as

$H_0 : \mu_M = \mu_F$

versus

$H_A : \mu_F \neq \mu_F$

In the context of the beer and hotwings data, this tests the hypothesis that there **is no** difference between the average number of hotwings consumed by students recorded as male and the average number of hotwings consumed by students recorded as female; versus there **is a** difference between the average number of hotwings consumed by students recorded as male and the average number of hotwings consumed by students recorded as female.

How do we conduct a two-sided test? We compute the test statistic for permutated data and repeat some large number of times just as we did with the one-sided test. However, for a two-sided test, we calculate both one-sided p-values, multiply the smaller by two, and finally (if necessary) round down to 1.0 (because probabilities can never be larger than 1.0). The reason why we multiply by two will become clear after an example.

Two-sided Test on Beerwings Example

Recall the Beerwings dataset

```
head(Beerwings)
```

```
##   ID Hotwings Beer Gender
## 1  1        4   24      F
## 2  2        5    0      F
## 3  3        5   12      F
## 4  4        6   12      F
## 5  5        7   12      F
## 6  6        7   12      F
```

Here is complete R code for conducting a two-sided permutation test on the difference in means for the number of hotwings consumed by students that are recorded as either female or male.

```
# get observed test statistic value
mean_f_obs <- Beerwings %>%
  filter(Gender == "F") %>%
  summarise(mean=mean(Hotwings)) %>% unlist()
mean_m_obs <- Beerwings %>%
  filter(Gender == "M") %>%
  summarise(mean=mean(Hotwings)) %>% unlist()
test_sts_obs <- mean_m_obs - mean_f_obs  # observed test stat value
# number of data points recorded as male
num_m <- Beerwings %>% filter(Gender == "M") %>% nrow()
# number of data  points recorded as female
num_f <- Beerwings %>% filter(Gender == "F") %>% nrow()
# total number of rows
data_size <- nrow(Beerwings)
# vector of values recorded for number of hotwings consumed
wings_consumed <- Beerwings$Hotwings %>% unlist()
# number of times to repeat permutations
N <- 10^5 - 1
perm_diff_means <- numeric(N)
for (i in 1:N){
  # permute data
  perm_ind <- sample(data_size,num_f,replace = F)
  # compute permuted group means
  perm_f_mean <- mean(wings_consumed[perm_ind])
  perm_m_mean <- mean(wings_consumed[-perm_ind])
  perm_diff_means[i] <- perm_m_mean - perm_f_mean
}

p_less <- (sum(perm_diff_means <= test_sts_obs) + 1)/(N+1)
p_greater <- (sum(perm_diff_means >= test_sts_obs) + 1)/(N+1)
(p_val <- min(1,2*min(p_less,p_greater)))
```

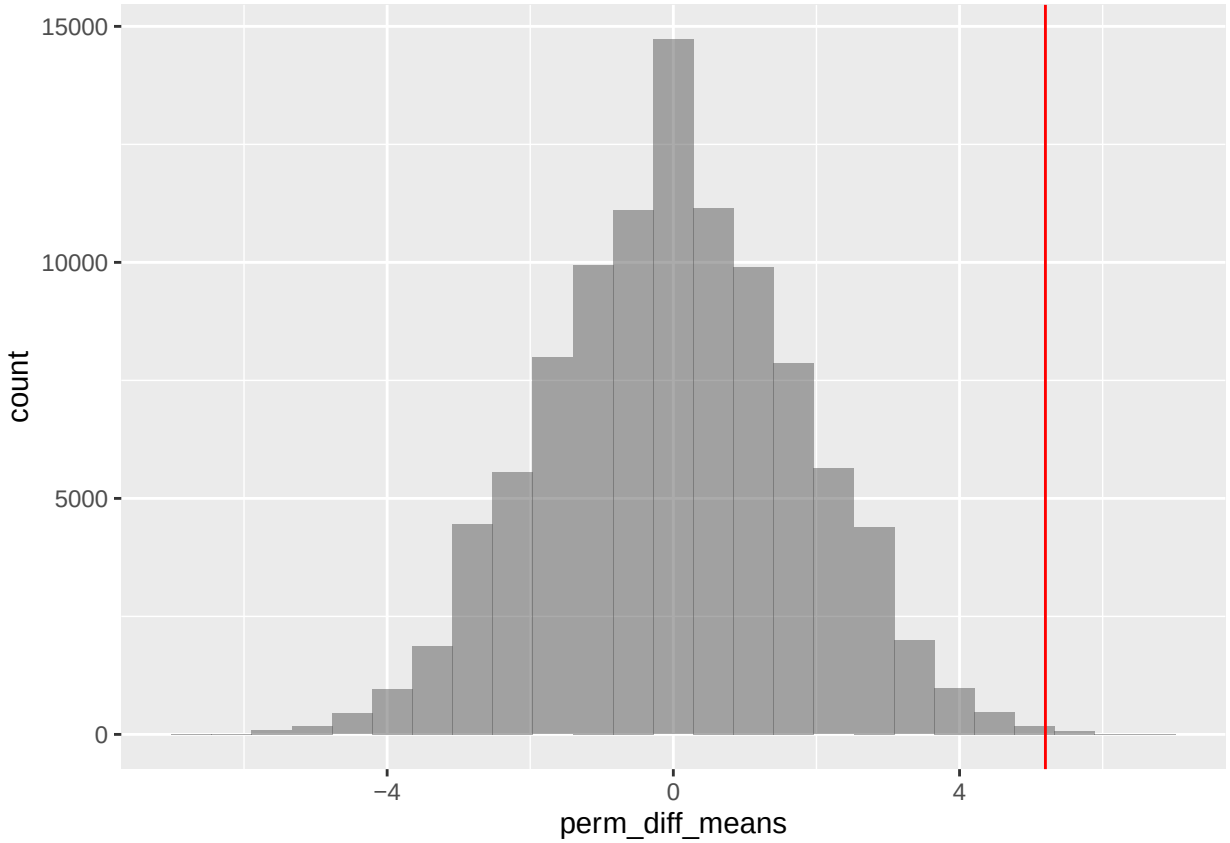
```
## [1] 0.00244
```

This p-value is quite small so we conclude that the data provides sufficient evidence to reject the null-hypothesis that there is no difference between the average number of hotwings consumed by students recorded as male and the average number of hotwings consumed by students recorded as female.

Let's examine a plot to get a better sense of what is going on:

```
gf_histogram(~perm_diff_means) %>% gf_vline(xintercept = test_sts_obs,color="red")
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```



When should you use a two-sided test?

Two-sided p-values are the default in statistical practice. You should perform a two-sided tests **unless** there is a very clear reason to pick a one-sided alternatice hypothesis. It is not fair to look at the data before deciding to use a one-sided hypothesis. Notice that a one-sided p-value will always be smaller that a two-sided p-value.

Choice of Test Statistic

In our examples so far, we have used the difference in means as our test statistic. This is a choice and not a necessity. When conducting a permutation test there are often many equally valid options for a test statistic. In fact, this can be formalized as a theorem.

Theorem: In permutation testing, if two test statistics T_1 and T_2 are related by a strictly increasing function $T_1(X^*) = f(T_2(X^*))$ where X^* is any permutation resample of the original data x, then they yield exactly the same p-values.

The point here is that when conducting a permutation test, you are free to use a test statistic that is well-suited to your particular problem regardless as to whether you know the sampling distribution for that particular test statistic or not.

Matched Pairs

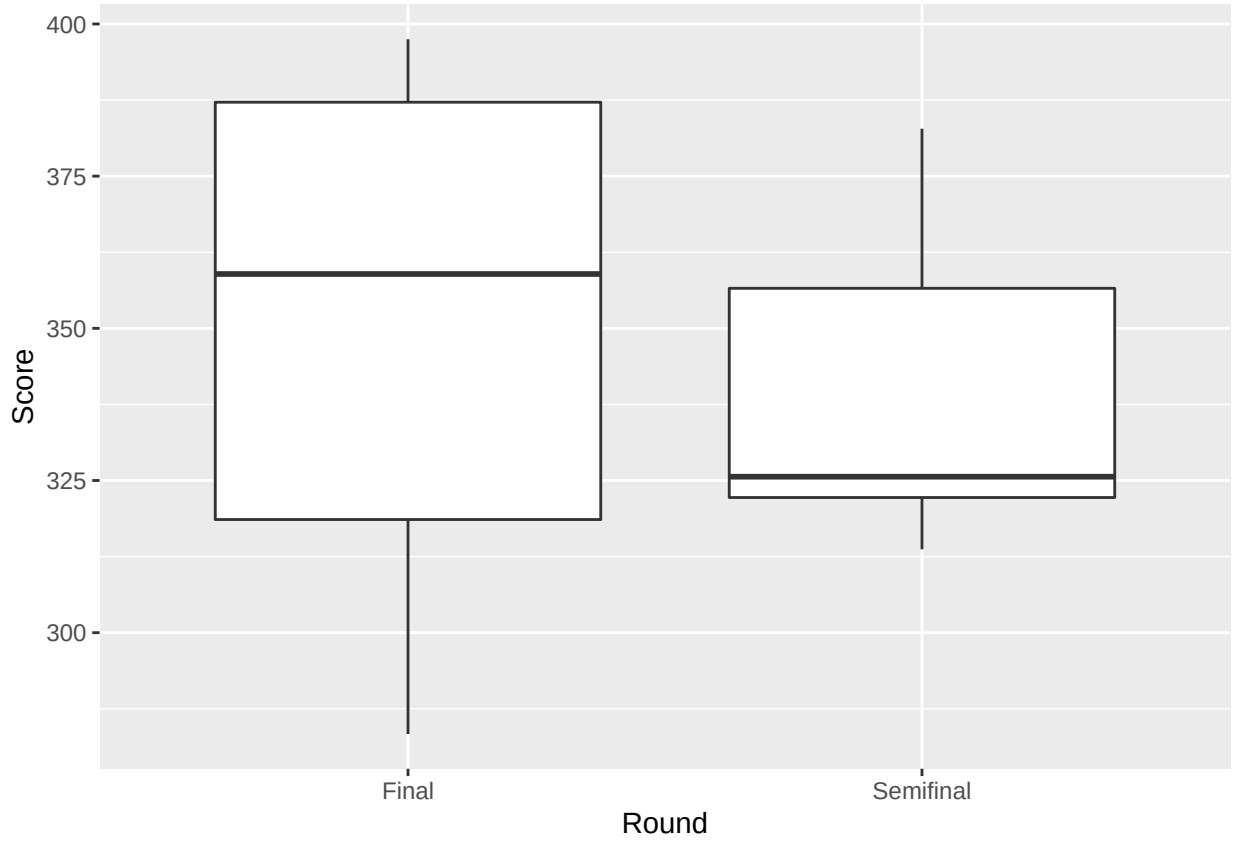
Consider the following data:

```
head(Diving2017)
```

```
##                Name     Country Semifinal  Final
## 1 CHEONG Jun Hoong    Malaysia    325.50 397.50
## 2          SI Yajie       China    382.80 396.00
## 3          REN Qian       China    367.50 391.95
## 4       KIM Mi Rae North Korea    346.00 385.55
## 5        WU Melissa   Australia    318.70 370.20
## 6    KIM Kuk Hyang North Korea    360.85 360.00
```

This data records the semifinal and final round scores for a number of competitive divers. A question we may be interested in is: is there a significant difference in a divers score between the semifinal and final rounds? To investigate this, let's begin with a plot:

```
Diving2017 %>%
  pivot_longer(c(Semifinal,Final),names_to = "Round",values_to = "Score") %>%
  gf_boxplot(Score ~ Round)
```



Let's also plot the differences between the semifinal round score and final round scores for each of the divers:

```
Diving2017 %>% mutate(ScoreDiff = Semifinal - Final) %>%
  gf_boxplot(~ScoreDiff)
```

We can easily compute the mean of the difference in scores between the semifinal and final rounds:

```
(obs_test_stat <- mean(Diving2017$Semifinal - Diving2017$Final))
```

```
## [1] -11.975
```

Is this difference likely due to random chance or is there perhaps some real effect that is the cause?

Question: How is this problem different than the one we considered for the Beerwings data? They are both **two-sample** problems. In the Beerwings dataset, the same measurement (number of hotwings consumed) was collected on two different independent groups, students recorded as female and students recorded as male. However, in the diving data, we are collecting two different measurements on the same group, that it, the same divers. The big difference here is that in the Beerwings data, the two samples are *independent* but they are *dependent* in the divers data. In fact, the divers data corresponds to something like a "before and after" scenario. This type of situation is referred to as a **matched pairs** sample.

How should we conduct a test on a matched pairs sample? For a permutation test of for mathced pairs, we randomly select some of the divers and transpose their two scores, leaving the other divers the same. This is essentially the same as randomly changing the sign from positive to negative (or vice versa) from some random subset of the difference in scores between the two rounds. Let's implement this:

```
score_diffs <- Diving2017 %>% mutate(ScoreDiff = Semifinal - Final) %>%
  select(ScoreDiff) %>% unlist()
data_length <- length(score_diffs)
N <- 10^5 - 1
mean_diffs_perm <- numeric(N)
for (i in 1:N){
  rand_signs <- sample(c(-1,1),data_length,replace = T)
  mean_diffs_perm[i] <- mean(rand_signs*score_diffs)
```

```
}

p_less <- (sum(mean_diffs_perm <= obs_test_stat) + 1)/(N+1)
p_greater <- (sum(mean_diffs_perm >= obs_test_stat) + 1)/(N+1)
(p_val <- min(1,2*min(p_less,p_greater)))
```

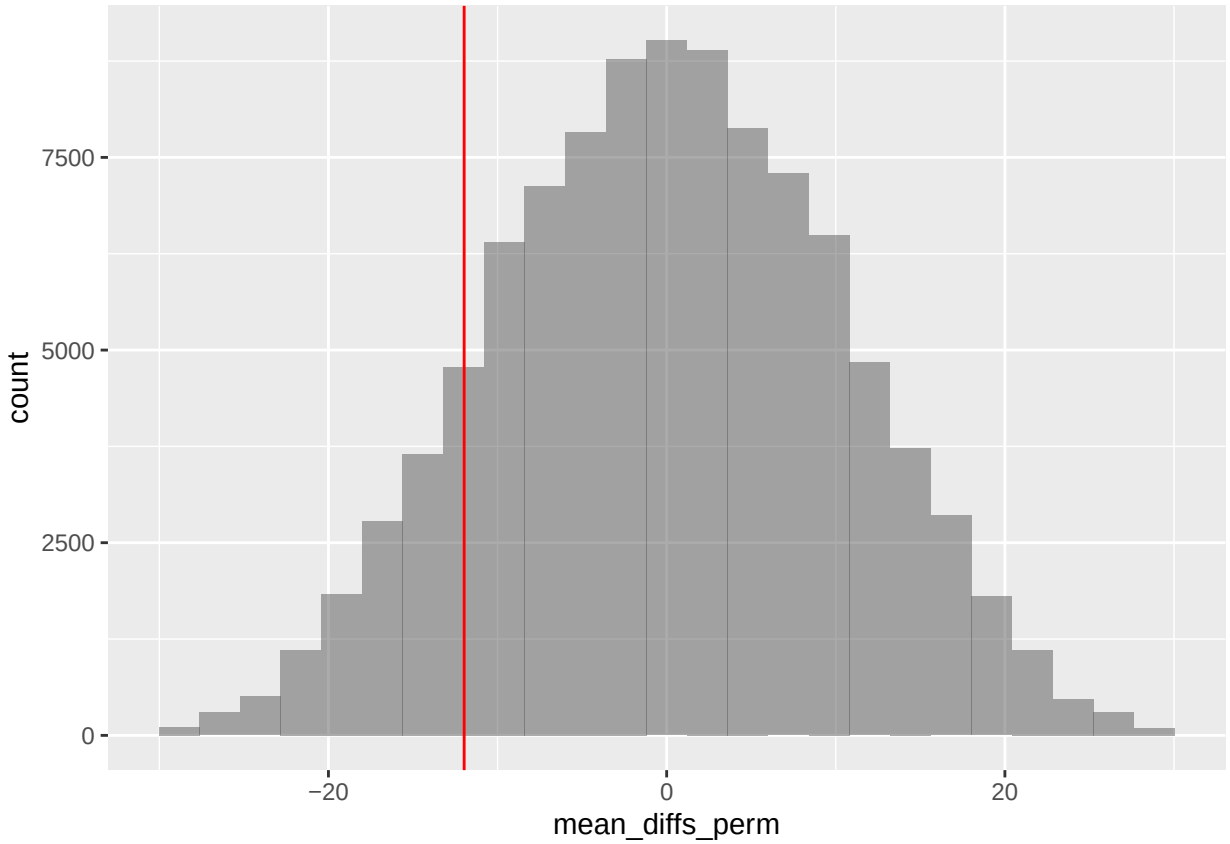
```
## [1] 0.2571
```

This p-value is not particularly small so we can not conclude that the data provides sufficient evidence to reject the null hypothesis.

Again, let's examine a plot:

```
gf_histogram(~mean_diffs_perm) %>%
  gf_vline(xintercept = obs_test_stat,color="red")
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```



Traditional Hypothesis Testing in R

Before the advent of sufficiently powerful and broadly accessible computing power, it was not feasible to use permutation tests in most practical applications. Thus, historically we have relied on formula based hypothesis tests that employ a specific probability distribution for one of a small number of common test statistics in order to compute p-values. For example, we know, by the central limit theorem, that the sampling distribution for the sample mean is normal. From this, we can derive that the sampling distribution for the so-called t statistic (which is often used to test hypotheses about the sample mean) is a t distribution and either the pdf or cdf for a t distribution can be used to compute the probability values necessary to

find a p-value. This leads to what is commonly called a t-test. We emphasize that the use of t-tests relies strongly on the fact that the data has been sampled from a population that follows a normal distribution. Permutation tests on the other hand do not rely on this assumption.

Here is an example application of a traditional t-test. Suppose that a company claims that their coffee vending machine dispenses 7 ounces of coffee per cup. It is unlikely that the machine is so perfect as to fill each cup with exactly 7 ounces each time so there is some natural variation to be expected. When an experiment is performed, we find that out of some number n cups vended randomly selected for measurement, we find the following data:

```
coffee <- c(6.4,6.5,7.1,6.5,7,6.5,7,6.6,7,6.6,6.7,6.8,6.5,7,6.8,6.6,6.8,6.6)
```

Thus, the average amount of coffee per cup is about 6.72 ounces. Is the company being honest about their claim? In order to determine this, we can test

$H_0 : \mu = 7$

versus

$H_A : \mu \neq 7$

In R, this can be done with a t-test as follows:

```
t.test(coffee,mu=7)
```

```
##
##  One Sample t-test
##
## data:  coffee
## t = -5.33, df = 17, p-value = 5.527e-05
## alternative hypothesis: true mean is not equal to 7
## 95 percent confidence interval:
##  6.612268 6.832177
## sample estimates:
## mean of x
##  6.722222
```

There is a lot of information in this output. What we are most intersted in at the moment is the p-value. This can be extracted as follows:

```
t.test(coffee,mu=7)$p.value
```

```
## [1] 5.526501e-05
```

This p-value is pretty small so we conclude that the data provides sufficient evidence to reject the null hypothesis that the true mean is 7 ounces of coffee per cup. Thus, we should be suspicious of the companies claim.

You may wonder how is this p-value being computed by R, it is obtained by using a t distribution as mentioned previously. Here is how this works:

```
# compute observed t statistic value
(obs_t <- (mean(coffee) - 7)/sqrt(var(coffee)/length(coffee)))
```

```
## [1] -5.330018
```

```
(t_p_val <- 2*pt(obs_t,df=(length(coffee)-1)))
```

```
## [1] 5.526501e-05
```

The first number is the observed value of a so-called t test statistic, and the second number is the probability of obtaining a value that is as or more extreme than the observed t statistic value under the null distribution

(which for reasons that we don't explain is known to be a t probability distribution).

What if we only wanted to test a one-sided hypothesis, for example:

$H_0 : \mu = 7$

versus

$H_A : \mu < 7$

Then we simply do this:

```
t.test(coffee,mu=7,alternative = "less")$p.value
```

```
## [1] 2.76325e-05
```

Two-sample t-tests

These t tests can be extended to two-sample situations like our hotwings and diving examples.

For example, here is a t-test applied to the two-sample data of the hotwings problem.

```
t.test(Hotwings~Gender,data=Beerwings)
```

```
##
##  Welch Two Sample t-test
##
## data:  Hotwings by Gender
## t = -3.5094, df = 26.584, p-value = 0.001619
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.242507 -2.157493
## sample estimates:
## mean in group F mean in group M
##        9.333333       14.533333
```

Notice that the p-value we obtain is very close to what we got using a permutation test:

```
t.test(Hotwings~Gender,data=Beerwings)$p.value
```

```
## [1] 0.001619451
```

compared with about 0.002 obtained from a permutation test.

Here is a t-test applied to the two-sample matched pairs data of the diving problem.

```
t.test(Diving2017$Semifinal,Diving2017$Final,paired = T)
```

```
##
##  Paired t-test
##
## data:  Diving2017$Semifinal and Diving2017$Final
## t = -1.1903, df = 11, p-value = 0.259
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -34.11726  10.16726
## sample estimates:
## mean of the differences
##                  -11.975
```

Notice that the p-value we obtain is very close to what we got using a permutation test:

```
t.test(Diving2017$Semifinal,Diving2017$Final,paired = T)$p.value
```

```
## [1] 0.2589684
```

compared with about 0.25774 obtained from a permutation test.

A One-sample Permutation Test

So far, we have only seen two-sample permutation tests. How can we conduct a one-sided permutation test as we did with the coffee data? Let's think about this. Subtract 7 (the hypothesized mean value) from each of the data points in the coffee data:

```
coffee_minus_seven <- coffee - 7
```

Under the null hypothesis that 7 is the true mean, we should expect to get a roughly equal number of positive and negative values upon repeated resampling.

Let's determine how often (out of 18 total data values) the observed data point is less than the hypothesized mean value of 7:

```
(obs_neg <- sum(coffee_minus_seven < 0))
```

```
## [1] 13
```

We see that 13 out of 18 of data points are less than 7.

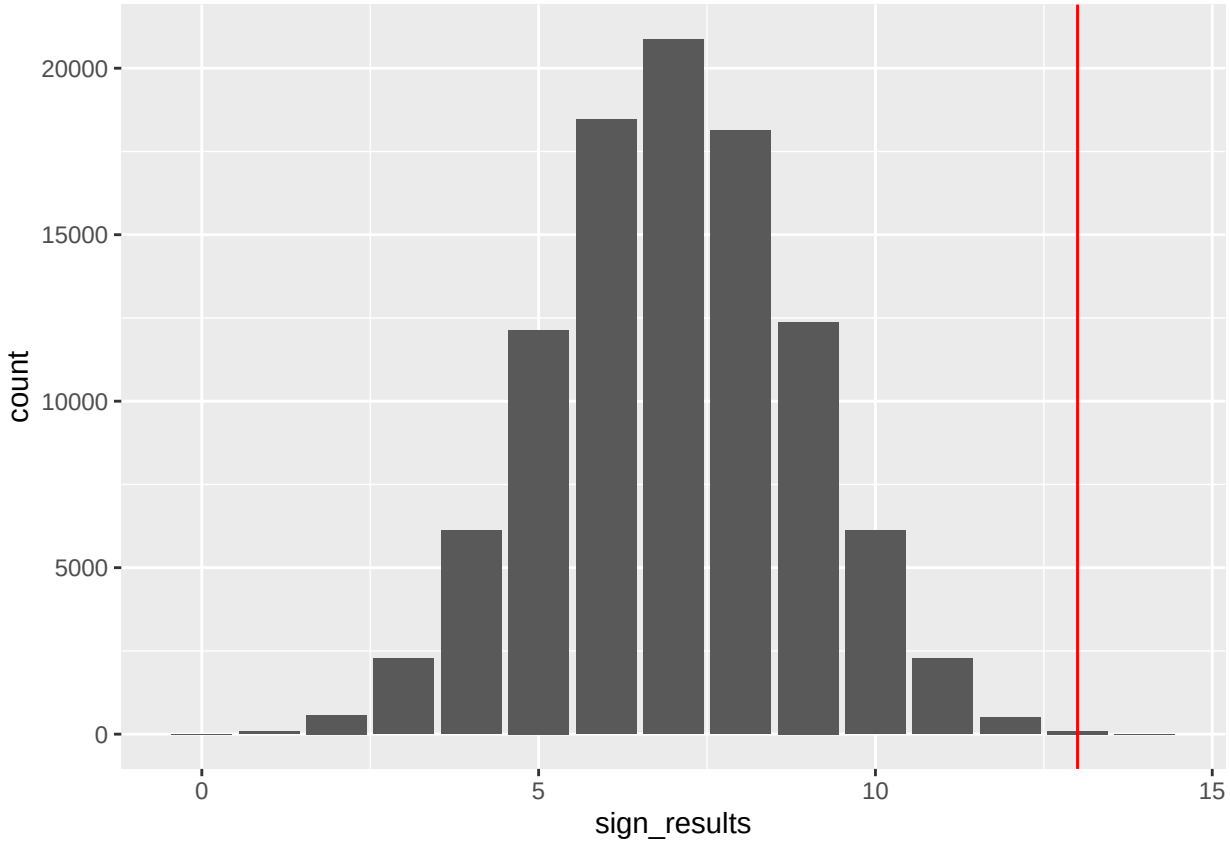
Now what we can do is to randomly shuffle the sign of the data values minus 7 and find the proportion of times that we get a result as or more extreme that what we observe in the collected data:

```
N <- 10^5 - 1
sign_results <- numeric(N)
for (i in 1:N){
  sign_vals <- sample(c(-1,1),length(coffee),replace = T)
  sign_results[i] <- sum(sign_vals*coffee_minus_seven < 0)
}
```

Let's look at the distribution of the number of negative values we get if we randomly assign a sign to 18 values:

```
gf_bar(~sign_results) %>% gf_vline(xintercept = obs_neg,color="red")
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```



The red vertical line indicates our observed value of 13 negative values. We see that under the null hypothesis, we expect that most often we will get about the same number of positive values as negative. Let's compute the proportion of times that we get as many or more than 13 negative values:

```
(sum(sign_results >= obs_neg) + 1)/(N+1)
```

```
## [1] 0.00104
```

Again, we obtain a very small p-value so we can feel reasonably confident that our data provides strong evidence for rejecting the null hypothesis. There is one subtle issue here however. The p-value in our permutation test is a fair amount larger than the p-value obtained using a t-test. The reason for this is most likely due to the fact that for a reliable t-test on a relatively small sample size, we are relying heavily on an assumption that the data comes from a population with a normal distribution. It is not necessarily (in fact unlikely) the case that our coffee data is sampled from a normal distribution. In this case, the permutation test is more reliable because it does not depend on an assumption about the distribution from which the data is sampled. This is not to imply that things can never go wrong with a permutation test. For example, regardless of what type of test you use, a small sample size can lead to unreliable results. How small is too small? The answer to that question can be complicated and will be taken up later.