

STAT340 Lecture 05: Statistical Testing, Continued

Brian Powers

10/1/2024

Introduction

This week, we will continue our discussion of statistical hypothesis testing. We'll look at more complicated examples of statistical hypotheses, and the question of how we choose a test statistic. In the process, we will completely formalize the Neyman-Pearson hypothesis testing framework, and we will discuss the types of errors we can make. We'll see how these different kinds of errors let us compare different tests of a particular null hypothesis, helping us to choose a test statistic that is more "powerful".

Learning objectives

After this lesson, you will be able to

- ▶ Explain the meaning of Type 1 and Type 2 errors in statistical hypothesis testing.
- ▶ Explain the significance level of a test.
- ▶ Test a simple parametric null hypothesis

Testing and Types of Errors

When we perform a statistical hypothesis test and produce a p -value, this p -value denotes a probability that (if the null hypothesis were true) we would see results at least as extreme as the observed data. While it may be small, this p -value is usually not zero. Thus, there is **some** probability that, even if the null hypothesis is true, we observe very unlikely data and, as a result, reject the null hypothesis incorrectly. Similarly, even if the null hypothesis is false, the data that we observe may not be “weird” enough to constitute sufficient evidence against the null hypothesis, and we may conclude, incorrectly, that the null hypothesis is true.

Testing Error Table

The possible outcomes are summarized in the table below.

	H_0 true	H_0 false
Do not reject H_0	True negative ✓	False negative ×
Reject H_0	False positive ×	True positive ✓

The bottom-left and top-right entries of this table are the kinds of errors we can make. In statistics, we often call them “Type I” and “Type II” errors, respectively.

Said another way:

- ▶ A **Type I** error corresponds to rejecting the null hypothesis when it is in fact true (“false alarm”).
- ▶ A **Type II** error corresponds to accepting the null hypothesis when it is *not* true (“miss”).

Example: Coin flipping

Let's suppose that we are flipping a coin and we want to assess whether or not the coin is fair. We model coin flips as being drawn from a Bernoulli with success parameter p . A fair coin corresponds to $p = 1/2$, so our null hypothesis is

$$H_0 : p = 1/2,$$

where $p \in [0, 1]$ is the probability of a coin flip landing heads.

A Type I error would correspond to the case where $p = 1/2$ but we conclude (incorrectly) that the coin is *not* fair.

A Type II error would correspond to the case where $p \neq 1/2$, but we conclude (incorrectly!) that the coin is fair.

Example: lady tasting more tea

Recall our lady tasting tea example, where our null hypothesis was

$$H_0 : \text{Muriel Bristol is guessing at random}$$

A Type I error would correspond to the case where Bristol is guessing completely randomly (i.e., cannot tell the milk-first cups from the milk-second cups), but we conclude, incorrectly, that she can tell the difference.

A Type II error would correspond to the case where Bristol really can tell the difference, but we incorrectly conclude that she is guessing at random.

Balancing errors

Of course, there are trade-offs involved managing Type I and Type II errors.

One way to avoid committing a Type I error altogether is to just conduct a test wherein we *always accept* the null hypothesis. This is great, except that if the null hypothesis is *not* true, then we will commit a Type II error with probability 1.

In the other direction, we could design a test that always rejects the null hypothesis. Then, if the null hypothesis is true, our probability of a Type I error is 1, but if H_0 is *not true*, we will always be right!

This makes it clear that unless we are okay with a totally useless test, we need to balance these two types of errors against one another.

The level of the test

For a particular test, the conditional probability that we commit a Type I error is called the *level* or *size* of the test, and is denoted by α .

The standard hypothesis testing approach is to decide, ahead of time, how large we are willing to let α be, and then choose the “rejection threshold” for our test statistic accordingly. That is, we specify the probability of a Type I error that we are willing to tolerate, and then adjust our test accordingly so that we reject H_0 (when it is true) with probability α .

Testing at level α

The “standard” hypothesis testing framework says that we should set our acceptable Type I error probability α and then conduct our test in such a way that the probability of a Type I error is indeed α . Let’s use the standard $\alpha = 0.05$.

How do we ensure that our statistical test has level $\alpha = 0.05$?

Suppose that our data is D and our test statistic is $T(D)$, and “extreme” or “unusual” or “weird” outcomes (i.e., observed data) correspond to larger values of $T(D)$.

So our test will reject the null hypothesis for especially large values of $T(D)$.

Then our aim is to find a number t_α such that

$$\Pr[T(D) \geq t_\alpha; H_0] = \alpha,$$

Finding the critical value

How do we find t_α , aka the “critical value” or “rejection threshold”?

If we know the distribution of $T = T(D)$ under H_0 , then this is simple. If F_0 denotes the cumulative distribution function of T under the null, then by definition,

$$\Pr[T > t_\alpha; H_0] = 1 - \Pr[T \leq t_\alpha; H_0] = 1 - F_0(t_\alpha).$$

So we just need to choose the critical value t_α in such a way that

$$F_0(t_\alpha) = 1 - \alpha.$$

Coinflips revisited I

Returning to our coin flipping example, suppose that we observe a sample of 200 coin flips, which we model as 200 independent Bernoulli random variables with success probability p , and we want to test the null hypothesis

$$H_0 : p = 1/2.$$

The first thing we need to do is choose a measure of how “unusual” a particular observation is (a test statistic).

A natural choice of test statistic is $T = \#$ of heads . That is, we just count how many of our 200 coin flips landed heads.

Coinflips revisited II

Important point: This is a *parametric model* for our data, so the test that we are about to develop is a *parametric test* of our null hypothesis. We are going to make specific use of our model assumption that the coinflips are distributed as independent Bernoullis, and use the fact that the total number of heads from those coinflips will be distributed as a Binomial random variable.

Under our model, T is a sum of $n = 200$ independent Bernoulli random variables, which means that T is distributed according to a Binomial random variable with size parameter 200 and success probability p .

Coinflips revisited III

Let's assume that larger T corresponds to more “extreme” or “unusual” data. That is, small values of T are *not* considered evidence against the null hypothesis (we'll address any problems with this assumption later).

Following our discussion above, we need to find the value of t that solves

$$F_T(t) = 1 - \alpha,$$

and reject if $T \geq t$.

The qRV function in R, where RV is the name of a random variable, computes *quantiles* of a distribution. For $q \in [0, 1]$, the q -th quantile of a distribution is the value t such that $F(t) = q$.

Coinflips revisited IV

```
# size=200, prob=0.5 because those are the parameters of our model  
qbinom(0.95, size=200, prob=0.5)  
## [1] 112
```

Let's just verify that this makes sense.

```
pbinom( 112, size=200, prob=0.5)  
## [1] 0.9615812
```

This is not 0.95 because, the binomial distribution is discrete, so there may not be an exact solution to $F_T(t) = 1 - \alpha$.

Coinflips revisited V

Let's try one click smaller:

```
pbinom( 111, size=200, prob=0.5)
## [1] 0.9481805
```

That's a little smaller than we'd like ideally— we want this number to be exactly $0.95 = 1 - \alpha$, but it's good enough!

Let's use a critical value of $t = 111$ and reject if $T > 111$, bearing in mind that our test is only *approximately* level- α .

Coinflips revisited VI

We should reject H_0 any time that we see $T > 111$ heads.

Let's simulate data and count how often we *incorrectly* reject the null hypothesis. It should be close to $0.05 * 100 = 1/20 = 5\%$ of the time.

Coinflips revisited VII

```
run_coinflip_trial <- function(pheads=0.5) {  
  # run one iteration of our experiment.  
  # Return 0 if we accept the null hypothesis and return 1 if we reject  
  
  # Generate data under the null hypothesis.  
  number_of_heads <- rbinom(n=1, size=200, prob=pheads);  
  
  # number_of_heads is already equal to our test statistic T.  
  # Just need to check whether it is above our rejection threshold.  
  if( number_of_heads <= 111) {  
    return( 0 ); # 0 like H_0, i.e., accept the null.  
  } else { # test statistic is larger than rejection threshold  
    return( 1 );  
  }  
}
```

Coinflips revisited VIII

Now, let's run the experiment a bunch of times.

```
set.seed(1)
NMC <- 1e4; # Repeat our experiment 10K times.
nreject <- 0; # keep track of how often we reject
for( i in 1:NMC ) {
  # conveniently, run_coinflip_trial returns 0 if we accept
  # and 1 if we reject, so to count how many rejections we get,
  # we can just sum them up.
  nreject <- nreject + run_coinflip_trial()
}

# Now, compute our (empirical) probability of rejection
nreject/NMC
## [1] 0.0535
```

Coinflips revisited IX

Now, let's see what happens when the null isn't true. Suppose that in truth, the coin is bent, and $\Pr[\text{heads}] = 0.75$. How often do we (correctly) reject the null hypothesis?

```
set.seed(2); NMC <- 1e4; # Repeat our experiment 10K times.
nreject <- 0; # keep track of how often we reject
for( i in 1:NMC ) {
  # conveniently, run_coinflip_trial returns 0 if we accept
  # and 1 if we reject, so to count how many rejections we get,
  # we can just sum them up.
  nreject <- nreject + run_coinflip_trial(pheads=0.75);
}
# Now, compute our (empirical) probability of rejection
nreject/NMC
## [1] 1
```

Coinflips revisited X

When the true value of $p = \Pr[\text{heads}]$ is far from $1/2$, our test manages to detect this fact and correctly reject the null hypothesis.

Coinflips revisited XI

How about a different way that our null hypothesis could be incorrect: suppose that $p = 0.25$.

```
NMC <- 1e4; # Repeat our experiment 10K times.
nreject <- 0; # keep track of how often we reject
for( i in 1:NMC ) {
  # conveniently, run_coinflip_trial returns 0 if we accept
  # and 1 if we reject, so to count how many rejections we get,
  # we can just sum them up.
  nreject <- nreject + run_coinflip_trial(pheads=0.25);
}

# Now, compute our (empirical) probability of rejection
nreject/NMC
## [1] 0
```

Coinflips revisited XII

Our test *never* rejects the null hypothesis, even though $0.25 = 1/4$ is pretty far from $1/2$. Why?

Well, our test rejects when the number of heads is larger than 111. If $p = 0.25$, we expect that most of the time we'll see about $200p = 50$ heads, which is much smaller than 111. Thus, we should expect that our test will (almost) never reject the null hypothesis.

One-sided vs two-sided tests I

This problem arises because our test above is what is a *one-sided test*. Our test statistic is only good for detecting when the true value of p is *larger* than $1/2$.

Let's consider a *different* statistical test, still based on the number of heads T , but this time we'll devise a *two-sided* test, wherein we will reject if T is too large **or** too small.

We want to reject if T is smaller than t_1 or larger than $t_2 > t_1$.

And we still want to make sure that we have significance level α . That is, we want it to be the case that

$$\Pr[\text{reject}; H_0] = \Pr[\{T < t_{\alpha,1}\} \cup \{T > t_{\alpha,2}\}] = \alpha$$

One-sided vs two-sided tests II

There are a lot of ways we can do this, but the easiest is to choose $t_{\alpha,1}$ and $t_{\alpha,2}$ so that $\Pr[T < t_{\alpha,1}] = \Pr[T > t_{\alpha,2}] = \alpha/2$. Then, since the events $\{T < t_{\alpha,1}\}$ and $\{T > t_{\alpha,2}\}$ are disjoint,

$$\Pr[\{T < t_{\alpha,1}\} \cup \{T > t_{\alpha,2}\}] = \Pr[T < t_{\alpha,1}] + \Pr[T > t_{\alpha,2}] = \frac{\alpha}{2} + \frac{\alpha}{2} = \alpha.$$

The set

$$\{x : x < t_{\alpha,1} \text{ or } x > t_{\alpha,2}\}$$

Is called a *rejection region*, because it is the set of values for which we reject the null hypothesis.

One-sided vs two-sided tests III

Note: different books and articles will follow different conventions around whether the region should have $x \leq t_{\alpha,1}$ or $x < t_{\alpha,1}$, and similarly for the upper limit. Just something to be careful of.

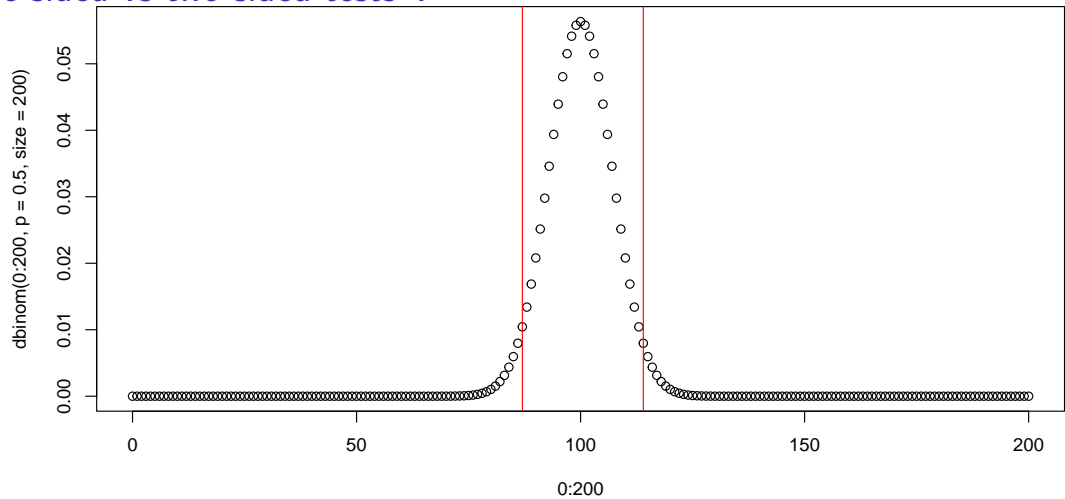
Which convention we use doesn't really matter– the important thing is that we are 1) consistent and 2) we make sure that we choose the rejection threshold so that our probability of rejecting under the null is α .

One-sided vs two-sided tests IV

Here's a picture of the situation:

```
# Draw the PMF of the Binomial  
plot(0:200, dbinom(0:200, p=0.5, size=200) )  
# Draw a line indicating a lower rejection threshold at 87  
# (we'll see below how we choose 86! For now, it's just for the sake of  
# choosing some lower limit to draw the picture!)  
abline(v=87, col='red')  
# and similarly draw a line for the right-hand rejection threshold.  
# again, how we choose  
abline(v=114, col='red')
```

One-sided vs two-sided tests V



One-sided vs two-sided tests VI

The dots indicate the distribution of our test statistic (i.e., the number of heads). Our test is going to reject the null hypothesis if the statistic T falls *outside* the two red lines.

The dots in the plot trace out the PMF of the Binomial, so the “area under the curve” is 1.

One-sided vs two-sided tests VII

To keep our test at level- α (i.e., to ensure that our probability of a Type I error is α , or, equivalently, that we incorrectly reject under the null with probability α), we need to choose these two lines so that the area under the curve *outside* the two lines is equal to α .

Equivalently, we need to choose the two red lines so that the area under the curve *between* the two lines is equal to $1 - \alpha$.

Choosing the rejection region I

How do we figure out $t_{\alpha,1}$ and $t_{\alpha,2}$?

Under H_0 T is a Binomial with size parameter 200 and success probability $p = .5$.

So, just as with t_α above, we can use `qbinom` to find $t_{\alpha,1}$ solving

$$F_T(t_{\alpha,1}) = \alpha/2.$$

Choosing the rejection region II

Okay, so let's do that.

```
# Reminder: we need the quantile of  $\alpha/2 = 0.025$ .  
qbinom(0.025, size=200, prob=0.5);  
## [1] 86
```

And let's just check that with pbinom. We want this to evaluate to (close to) 0.025.

```
pbinom(86, size=200, prob=0.5)  
## [1] 0.02798287
```

A bit high. Let's try one click smaller.

```
pbinom(85, size=200, prob=0.5)  
## [1] 0.0200186
```

Far too small, so let's stick with rejecting when $T \leq 86$, i.e., $T < 87$. So $t_{\alpha,1} = 87$.

Choosing the rejection region III

What about $t_{\alpha,2}$? We need to choose it so that $\Pr[T > t_{\alpha,2}; H_0] = 0.025 = \alpha/2$. Well,

$$\Pr[T > t_{\alpha,2}; H_0] = 1 - \Pr[T \leq t_{\alpha,2}; H_0].$$

So we just need to find $t_{\alpha,2}$ satisfying

$$\Pr[T \leq t_{\alpha,2}; H_0] = 1 - \alpha/2 = 1 - 0.025 = 0.975.$$

Once again, now that we have an event of the form $T \leq t_{\alpha,2}$, we have a plain old cumulative distribution function, and we can use R to solve this.

```
qbinom(0.975, size=200, prob=0.5)
## [1] 114
```

Choosing the rejection region IV

Double check with pbinom:

```
pbinom(114, size=200, prob=0.5)
## [1] 0.9799814
```

And checking the threshold a bit lower:

```
pbinom(113, size=200, prob=0.5)
## [1] 0.9720171
```

Okay, neither of these is great– we want exactly 0.975, remember. Let's stick with 114.

So we are going to use $t_{1,\alpha} = 87$ and $t_{2,\alpha} = 114$ as demarcating our rejection region.

Let's implement our two-sided hypothesis test.

Choosing the rejection region V

```
run_coinflip_trial2 <- function(pheads=0.5) {  
  # run one iteration of our experiment for our two-sided test.  
  # Return 0 if we accept the null hypothesis and return 1 if we reject  
  
  # Generate data under the null hypothesis.  
  number_of_heads <- rbinom(n=1, size=200, prob=pheads);  
  
  # number_of_heads is already equal to our test statistic T.  
  # Just need to check whether or not it is in the rejection region  
  if( number_of_heads < 87 | number_of_heads > 114 ) {  
    return(1); # Reject the null hypothesis.  
  } else {  
    return(0); # Accept H_0, hence returning 0.  
  }  
}
```

Choosing the rejection region VI

And once again, let's try running our experiment a large number of times where H_0 is true.

```
set.seed(1); NMC <- 1e4; # Repeat our experiment 10K times.
nreject <- 0; # keep track of how often we reject
for( i in 1:NMC ) {
  # conveniently, run_coinflip_trial returns 0 if we accept
  # and 1 if we reject, so to count how many rejections we get,
  # we can just sum them up.
  nreject <- nreject + run_coinflip_trial2(pheads=0.5);
}

# Now, compute our (empirical) probability of rejection
nreject/NMC
## [1] 0.0502
```

One-sided vs two-sided tests I

So, at this point, we've seen two different tests of our null hypothesis,

$$H_0 : p = \frac{1}{2}.$$

Both have the same level $\alpha \approx 0.05$, but they have different rejection regions.

Our first test was one-sided, and we only rejected for large values of $T = \text{number of heads}$.

Our second test was two-sided, and rejected for small *or* large values of T .

Notice, however, that the “large” threshold for these two different tests are different.
Compare

$$\{t : t > 114\} \text{ versus } \{t : t > 111\}.$$

One-sided vs two-sided tests II

Our two-sided test, which rejects values of T bigger than 114, requires “more extreme” values on the large side to reject the null hypothesis when compared with our one-sided test.

This means that there *should* be values of p for which our one-sided test rejects reasonably frequently while our two-sided test is less likely to reject. Let's see if we can find such a value.

First, let's write code to compare our two different tests.

One-sided vs two-sided tests III

```
compare_reject_rates <- function( p ) {  
  NMC <- 1e4; # Repeat our experiment 10K times.  
  
  # keep track of how often our two different tests reject.  
  nreject_twosided <- 0; nreject_onesided <- 0;  
  
  for( i in 1:NMC ) {  
    # conveniently, run_coinflip_trial returns 1 if we reject, so to  
    # count how many rejections we get, we can just sum them up.  
    nreject_onesided <- nreject_onesided+run_coinflip_trial(pheads=p)  
    nreject_twosided <- nreject_twosided+run_coinflip_trial2(pheads=p)  
  }  
  return( c( nreject_onesided/NMC, nreject_twosided/NMC ) )  
}
```

One-sided vs two-sided tests IV

Now, let's try some values of p .

```
pseq <- seq(0.5, 1, 0.01); #.5 to 1 by 0.01
```

```
# see ?mapply for details.
```

```
rej_rates <- mapply( compare_reject_rates, pseq )
```

```
# Take a careful look at how the output is shaped:
```

```
rej_rates[,1:9]
```

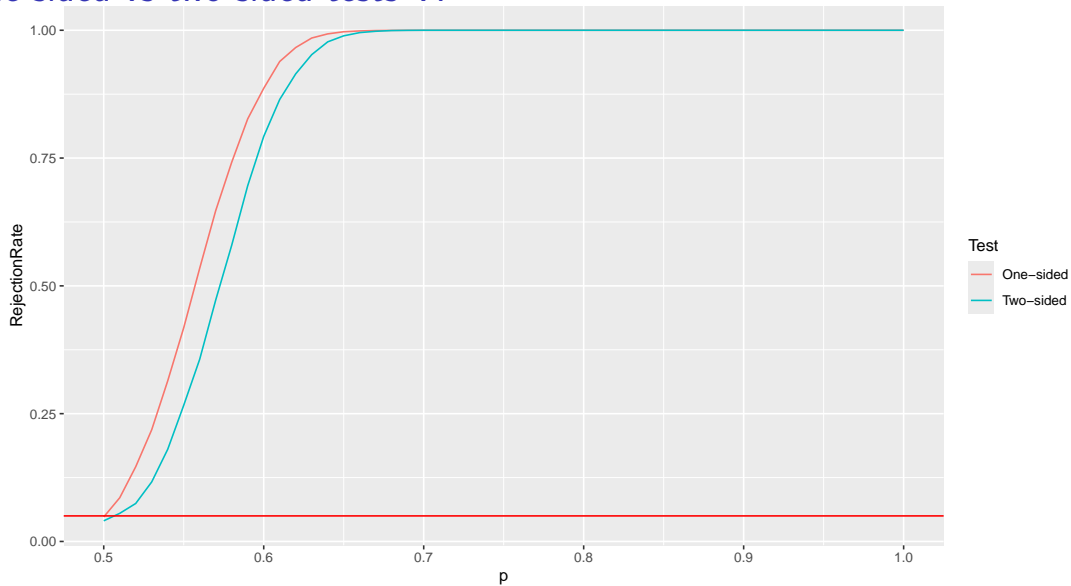
```
##           [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  
## [1,] 0.0479 0.0855 0.1462 0.2186 0.3142 0.4181 0.5345 0.6472 0.7418  
## [2,] 0.0404 0.0550 0.0743 0.1165 0.1803 0.2664 0.3565 0.4723 0.5795
```

Let's make a plot of the two different rejection rates as a function of p .

One-sided vs two-sided tests V

```
onesided <- rej_rates[1,]; twosided <- rej_rates[2,]  
# We're going to make a plot in ggplot2 because it's prettier.  
# ggplot2 wants a dataframe: We'll make a column of our probabilities,  
# a column of rejection rates, and a column of one- vs two-sided.  
# We need two copies of the pseq sequence, one for each of our tests,  
# hence the p=rep(pseq,2)  
rejrate <- c(onesided, twosided);  
sidedness <- c( rep('One-sided',length(onesided)),  
               rep('Two-sided',length(twosided) ) );  
df <- data.frame( p=rep(pseq,2), RejectionRate=rejrate, Test=sidedness );  
  
pp <- ggplot( df, aes(x=p, y=RejectionRate, color=Test) );  
pp <- pp + geom_line(aes(color=Test)) +  
  geom_hline(yintercept=0.05, color='red');
```

One-sided vs two-sided tests VI



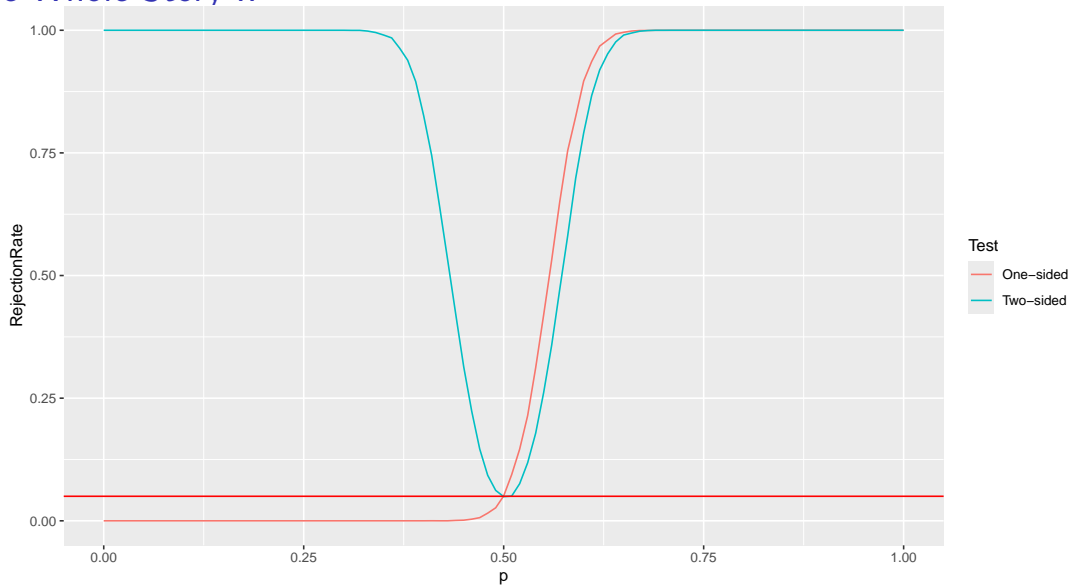
The Whole Story I

This plot doesn't tell the whole story– let's look at the whole range of $p \in [0, 1]$.

```
pseq <- seq(0, 1, 0.01);
rej_rates <- mapply( compare_reject_rates, pseq )
onesided <- rej_rates[1,]
twosided <- rej_rates[2,]
rejrate <- c(onesided,twosided);
sidedness <- c( rep('One-sided',length(onesided)),
               rep('Two-sided',length(twosided) ) );
df <- data.frame( p=rep(pseq,2), RejectionRate=rejrate, Test=sidedness );

pp <- ggplot( df, aes(x=p, y=RejectionRate, color=Test) );
pp <- pp + geom_line(aes(color=Test)) + geom_hline(yintercept=0.05, color=
```

The Whole Story II



An important point

Both of these test have the *same* level. That is, their probability of rejecting the null when H_0 is true (i.e., $p = 1/2$) is the same (up to the approximations that we had to make in dealing with the fact that the Binomial is a discrete distribution):

```
compare_reject_rates(0.5)
## [1] 0.0522 0.0485
```

Test statistics and where to find them

Crucial to our discussion above was that we have our function that measured how “unusual” or “extreme” our data is. This function is called a *test statistic* because it is a function of the data (hence a “statistic”) and because we use it in the test of our hypothesis.

Sometimes it’s really obvious what our test statistic should be— like counting how many cups our tea taster got right. Sometimes it’s less obvious. We’ll see examples like that later in the semester.

Let’s look at this in the context of our coinflipping example above. We’ll why the “right” test statistic isn’t always obvious.

Example: flipping a (possibly biased) coin, continued I

Reminder: our null hypothesis was that the coin is fair, which we would write as

$$H_0 : p = \frac{1}{2}.$$

Here is the result of flipping this coin 200 times:

```
flips <- paste0("HTTHTHHTTHTTTHTHHHTTHHTHHHHHTTTTHHHHHHHHTHHHTHHHTT",  
               "THTTTTHHHHTHHHHHHHTHTTHTTHTHTTTHHHHTHHHTTHTTTTTHHHHTTHTHT",  
               "HHTTTHHHTTTHTHHHTTHHTTTTTHTHHHTTTHTTHTTTHHTTTTTHTTTHTHHHHHT",  
               "TTTTHTTTTHHHHHHTHHHTHTHHHTTTTTTTHHHTTTHTTT")
```

How might we go about testing our null hypothesis that $p = 1/2$? Well, we have to start by asking what we expect the data to “look like” if the probability of heads is actually $1/2$, and then come up with a test statistic that captures that.

Example: flipping a (possibly biased) coin, continued II

Above, we chose the number of heads as our test statistic, which is certainly a reasonable choice, but there are other test statistics we could have chosen.

Let's implement that test statistic, this time in a way that it takes a string of coinflips like our data above.

```
count_heads <- function( coinflips ) {  
  # Just a reminder that in non-demo code, we usually want to include  
  # some error checking to make sure that our function's arguments are  
  # as we expect.  
  return( sum( coinflips==rep('H', length(coinflips)) ) );  
}
```


Example: flipping a (possibly biased) coin, continued III

Let's apply this statistic to our flips data.

```
# We need to turn our string of coin flips into a vector before  
# we pass it into `count_heads()`  
# strsplit( x, split=s) splits the string x on the separating character s,  
# and returns a list structure. See ?strsplit for more.  
# unlist( v ) turns a list structure into a flat vector that is easier  
# to work with. See ?unlist for details.  
count_heads( unlist( strsplit( flips, split=' ' ) ) );  
## [1] 99
```

We've chosen our test statistic, which measures how “unusual” or “extreme” our data is. But we don't know how unusual is unusual, or how extreme is extreme.

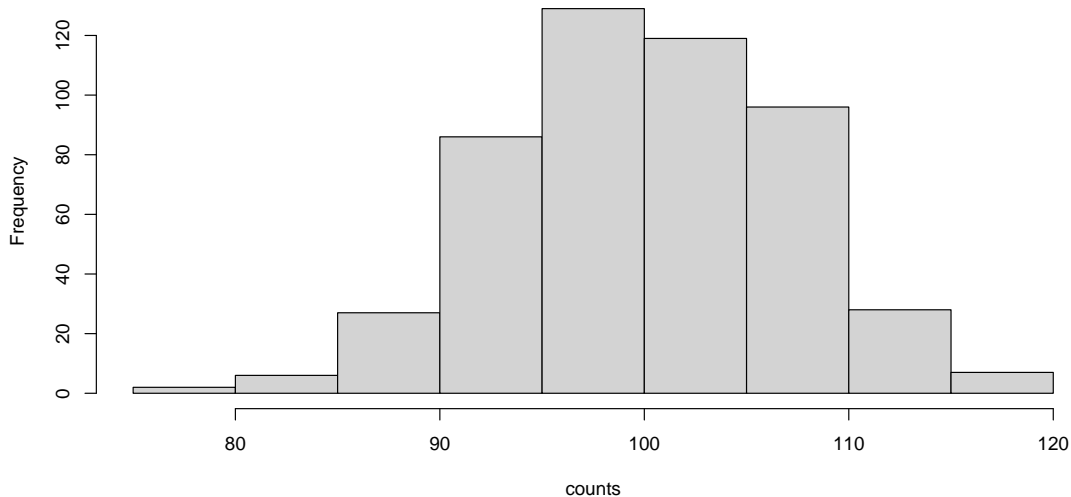
Let's simulate some data under the null hypothesis to get an idea.

Example: flipping a (possibly biased) coin, continued IV

```
simulate_coinflips <- function(n, p) {  
  flips <- sample( c('H','T'), size=n, replace=TRUE, prob=c(p, 1-p) );  
  # flips is now a vector of 'H' and 'T's, which is what count_heads  
  # expects, so let's return it.  
  return( flips );  
}  
  
# Now let's simulate a bunch of coinflips  
NMC <- 500;  
counts <- rep( 0, NMC );  
for( i in 1:NMC ) {  
  counts[i] <- count_heads( simulate_coinflips( 200, 0.5 ) );  
}  
  
hist( counts )
```

Example: flipping a (possibly biased) coin, continued V

Histogram of counts



Example: flipping a (possibly biased) coin, continued VI

The vast majority of the time, the number of heads in 200 fair coin flips is between 80 and 120. Once in a while, of course, it's more than that or less than that, but an “unusual” result would presumably correspond to a number of heads being much larger *or* much smaller than 100. This corresponds to our *two-sided test* discussed above— we are concerned about the true value of p being *either* higher or lower than our null value $p = 1/2$.

For now, let's just note that our data observed in `flips`, has a number of coin flips well within our “usual” range of 80 to 120.

```
count_heads( unlist( strsplit(flips, split='')) )  
## [1] 99
```

Example: flipping a (possibly biased) coin, continued VII

Now, here's a different sequence of coin flips, **100 heads** followed by **100 tails**.

```
more_flips <- paste(rep(c("H", "T"), c(100, 100)), collapse='')
```

```
count_heads( unlist( strsplit( more_flips, split='' )))  
## [1] 100
```

According to our test statistic, this is not an unusual outcome at all. Yet we can all agree that there's something unusual about the sequence of coin flips above.

So here we see that our test statistic doesn't capture all of the ways that our data could be “extreme” or “unusual”. This is precisely why devising a “good” test statistic can be hard!

Example: flipping a (possibly biased) coin, continued VIII

One example of a good statistic here may be to examine run lengths, e.g. using a function like this.

```
longestRun = function(x,target){  
  max(0,with(rle(x), lengths[values==target]))  
}
```

There are a lot of different ways that our data can be “weird”, and we need to be careful that we capture the right notion of weirdness!

Choosing α : trade-offs I

The standard choice in (most of) the sciences is to set $\alpha = 0.05$. You probably already remember this number from STAT240 or other courses.

This really is a pretty arbitrary choice, based largely on some writings by early statistics researchers, but for some reason it has stuck. You'll sometimes see researchers use $\alpha = 0.01$, also, but the “right” choice of α really depends on the nature of the research problem you are asking and on how “bad” it would be to commit a false positive.

Example

Consider a screening test for cancer; the null hypothesis is “the patient **does not** having cancer”. Type I error corresponds to mistakenly declaring that a test subject has cancer when they in fact do not. This false positive may cause worry and would trigger expensive follow-up tests to confirm a cancer diagnosis.

But, compare this risk against the risk of committing a Type II error. This would correspond to a patient who comes to us with cancer, but our test does not detect the cancer. Our patient goes home, mistakenly thinking they do not have cancer. This is certainly a much worse outcome than a Type I error, and we may want to spend more resources guarding against this, *even if* it means a higher chance of committing a Type I error.

Generally speaking, reducing α (i.e., reducing the probability of a Type I error) incurs an increase in the probability of a Type II error, and vice versa.

Review

In this lecture we covered:

- ▶ Type 1 and Type 2 Errors
- ▶ Balance between error types
- ▶ Significance Level α
- ▶ Rejection rules (critical value)
- ▶ One-sided vs two-sided tests
- ▶ Rejection region
- ▶ Power of a test statistic
- ▶ Power function (curve)
- ▶ Comparing test statistics
- ▶ Choosing α