Confidence and Hypotheses

Author: Nicholas G Reich

This material is part of the ***statsTeachR*** *project*

A story: The Lady Tasting Tea

- In this famous statistical fable, an aristocratic British Lady claims she can tell whether milk has been poured into tea or vice versa. This story was first documented by Ronald Fisher in 1935. More details here.
- Question: How do we test this claim?
- One Possible Answer: Think about each guess about a cup of tea as a flip of a coin with a given probability (p) of being heads (or the guess being right).
- What would p=0.1 mean in the context of the tea-tasting lady? p=.5? p=.8?

Let's turn the Lady into a computer

We can use `rflip()` to simulate flipping coins (or tea-tasting ladies):

```
library(mosaic)
rflip()

##
## Flipping 1 coin [ Prob(Heads) = 0.5 ] ...
##
## H
##
## Number of Heads: 1 [Proportion Heads: 1]
```

Let's have the Lady try 10 cups of tea

Rather than flip each coin separately, we can flip multiple coins at once. `rflip(10)` simulates 1 lady tasting 10 cups of tea 1 time each time with a 50% chance of getting it 'right'.

```
rflip(10, prob=0.5)

##
## Flipping 10 coins [ Prob(Heads) = 0.5 ] ...
##
## H H T T H H H H T H
##
## Number of Heads: 7 [Proportion Heads: 0.7]
```

And then create a swarm of Ladies

We can do that 'many' times to see how multiple guessing ladies do:

```
do(2) * rflip(10, prob=0.5)

##    n heads tails prop
## 1 10     4     6  0.4
## 2 10     4     6  0.4
```

- 'do()' is a function within the 'mosaic' package that is clever about what it remembers (in many common situations).
- 2 isn't many Ladies – we'll do many in a minute – but it is a good idea to take a look at a small example before generating a lot of random data.
- What kind of R object does the command 'do(2) * rflip(10)' return?

Now let's simulate 5000 'imposter' tea-tasting ladies

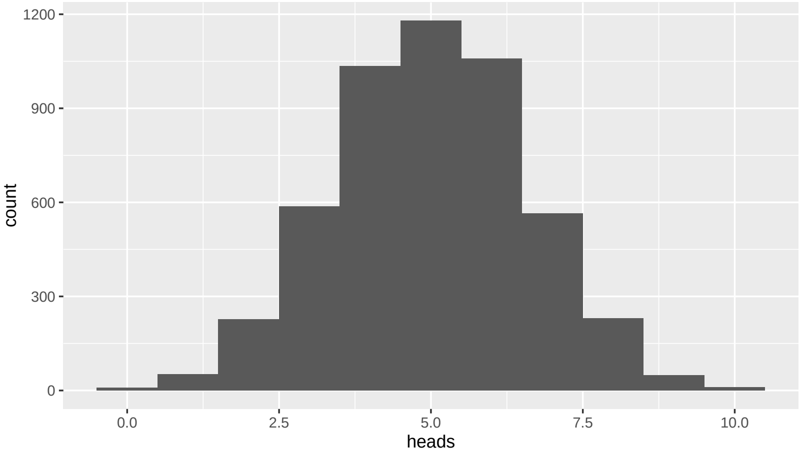
```
Ladies <- do(5000) * rflip(10)
head(Ladies, 8)

##    n heads tails prop
## 1 10     2     8  0.2
## 2 10     5     5  0.5
## 3 10     5     5  0.5
## 4 10     6     4  0.6
## 5 10     5     5  0.5
## 6 10     6     4  0.6
## 7 10     8     2  0.8
## 8 10     2     8  0.2
```

The results of our experiment...

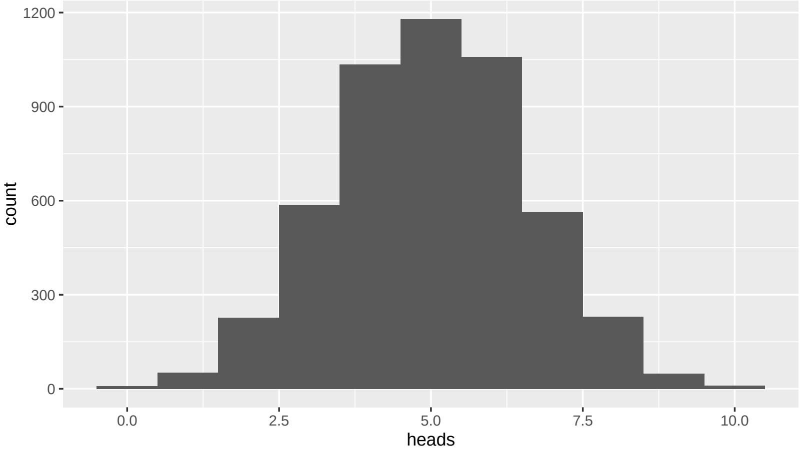
This is what natural randomness looks like if the ladies are guessing randomly, i.e. can't tell the difference between the cups of tea.

```
library(ggplot2)
qplot(heads, data=Ladies, binwidth=1)
```



The results of our experiment...

What would convince you that the original Lady can tell the difference between cups of tea?



Framing this in terms of hypotheses

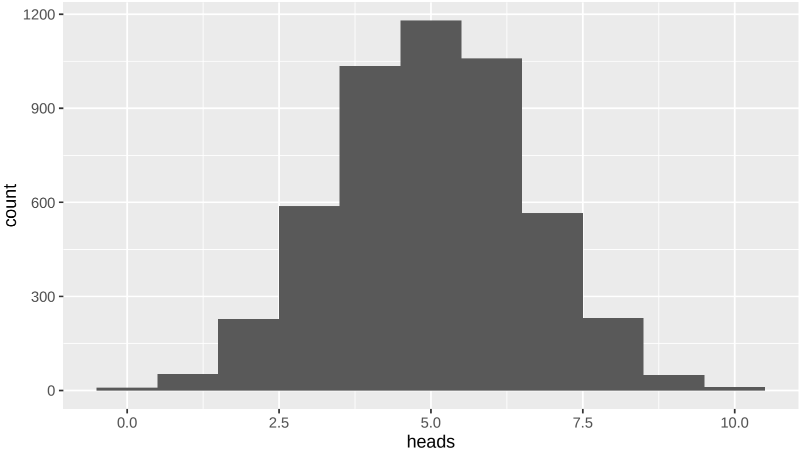
In statistical parlance, a hypothesis is a statement about the world that can be supported or refuted with data.

Classical hypothesis testing in statistics has a few ingredients

- A "Null Hypothesis" (H_0, or 'H-naught'): a proposition of "no difference" between groups or that deviations are due to sampling error. In our tea-tasting example, $H_0 : p = 0.5$.
- One or more "Alternative Hypotheses" (H_A): a proposition that there is a difference between groups, or a significant "effect".

Null distributions characterize expected variability

This is an example of a “null distribution”: it characterizes expected variability when H_0 is true. Values far from 0.5 indicate evidence against H_0.



Conclusions

- What proportion of your Ladies Tasting Tea guessed 9 or 10? (Note that this is the same as asking that, assuming we are flipping a fair coin, how often do we see 9 or 10 heads?)
- There are formal ties to the Binomial Distribution that we will talk about later...
- Rumor has it that the original Lady (described by Fisher) correctly guessed all 10 cups of tea.

Extension to election predictions

Binary outcome: the i^{th} guess about a cup of tea

$$X_i = \begin{cases} 1, & \text{if Lady guesses right about the } i^{th} \text{ cup of tea} \\ 0, & \text{if Lady guesses right about the } i^{th} \text{ cup of tea} \end{cases}$$

Binary outcome: the election outcome in the i^{th} state

$$X_i = \begin{cases} 1, & \text{if Clinton wins} \\ 0, & \text{if Clinton loses} \end{cases}$$

Code to simulate election results

The function `rbinom()` is similar to `rflip()` but it allows each state/cup of tea to have a different probability of 'success'.
Note: **all flips are independent!**

```
rbinom(n = 50, size = 1, prob = rep(.5, 52))

##  [1] 0 1 1 0 1 0 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 1
## [36] 1 0 0 1 1 0 1 0 1 0 0 1 0 1 0

do(2) * rbinom(n = 50, size = 1, prob = rep(.5, 52))

##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
## 1  1  0  1  0  0  1  1  1  1   0   0   0   1   0   1   1   1   1   1   0
## 2  1  1  0  1  0  1  0  0  0   1   0   1   0   1   1   0   1   0   1   1
##   V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38
## 1   1   0   0   1   0   1   1   1   1   0   1   1   0   0   1   1   1   0
## 2   1   1   0   0   1   1   0   1   0   0   0   1   1   0   1   0   0   1
##   V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50
## 1   1   0   0   0   0   1   1   0   1   1   1   0
## 2   0   0   0   0   0   0   1   1   0   0   1   1
```

Some data to help us simulate elections

```r
elect <- read.csv("https://goo.gl/HCHzui")
head(elect)

##         state electoral.votes HRC.win.prob
## 1    Alabama                9          0.5
## 2     Alaska                3          0.5
## 3    Arizona               11          0.5
## 4   Arkansas                6          0.5
## 5 California               55          0.5
## 6   Colorado                9          0.5
```

Some data to help us simulate elections

```
nsims <- 10
sims <- do(nsims) * rbinom(n = 50, size = 1,
                           prob = elect$HRC.win.prob)
tsims <- t(sims)
HRC_votes <- elect$electoral.votes*tsims
HRC_totals <- colSums(HRC_votes)
```

Plot the results!

An election simulation of your own

Assuming all states were an even coinflip (p=0.5) and independent, what is the probability that Clinton gets 270 or more electoral votes? Run the simulation a lot of times to find out!

Instead of assuming even probabilities, let's input probabilities from one of the online prediction models and then run it again. Remember, unless we do something more complicated, we are assuming that each state outcome is independent from all the others, which may not be a valid assumption.

Link to state-by-state win probabilites.