

All I Do is Win Win Win!

Directions: Follow along with the slides and answer the questions in **BOLDED** font in your journal.

From Categories to Numbers

- In the last lab, we looked at how we can use simulations to estimate probabilities of compound events.
- We did this for a bag of different colored marbles.
- Many situations come up where we need to simulate probabilities of numbers instead.
- This lab will explore some of those situations and look at a few techniques to compute estimates.

The California Lottery

- The California lottery has a game called the *Daily 3*:
- 3 numbers between 0-9 are drawn daily.
- Numbers are drawn *with replacement*.
- Winners are usually awarded a couple hundred dollars.
- To win the maximum amount of money, players must correctly choose the numbers that are drawn, in order.
- **If you were going to play the *Daily 3* game, which numbers would you pick? Write these numbers down.**

Playing with the lotto

- Using the rules of the *Daily 3*, use the `sample` function and a `do-loop` to create 300 sample games.
- Assign the name `games` to your 300 simulations.
- **Write down the code you used to create your 300 sample games.**
- We can use these simulations to compute estimates for different events.

Simulations with Numbers

- We treat numbers a little differently than categories.
- Because of this, we use the `subset` command instead of `tally`.
- Type the following:

```
draws <- subset(games, V1 < 3 & V2 == 7)
```

- This code will look in our 300 simulated lotto `games`. Then,
- Find the `games` whose first draw was less than 3 AND whose second draw was exactly equal to 7.
- Assign only these simulated games to the `draws` data frame.

Counting (Ah-Ah-Ah..)

- After we've `subset` the `draws` we're interested in, we can use the `nrow()` function to count how many of our simulations had a first draw that was less than 3 and a second draw that was exactly 7:

```
nrow(draws)
```

- We use `nrow` to count the number of `rows` in our `draws` data frame.
- Then, we can divide this number by our total number of simulations (300) to give us an estimate for the probability that the first number drawn is less than 3 and the second number drawn is 7.

Give it a shot

- Go back to the 3 numbers you would use if you were playing the *Daily 3* lotto game.
- Use the `subset` function to find which of the simulated `games` you would have won.
- Use the `nrow` function to count how many games are in your subset.
- Divide by the number of simulations to compute your estimate of the probability of winning the *Daily 3*.
- **What was the estimated probability you came up with?**
- **Will you always end up with the exact same estimated probability for winning the lotto? Why or why not?**

Why isn't there a function to compute my estimate?

- As you've probably noticed in this lab, not everything you want to compute in programming comes with a function.
- Sometimes you have to be creative to calculate solutions.
- That's just how it goes for data scientists.
- **Write down the 3 steps involved in estimating the probability of winning the *Daily 3*.**
- **What role does each step play in the process and why is it important?**

On your own

- **Use the `games` you simulated to compute and write down estimates for the following probabilities:**
- The probability of correctly guessing the 1st and 3rd draws.
- The probability of correctly guessing the 1st and 2nd draws.
- **Should the actual probabilities of the above situations be the same? Why might their estimates be different?**
- **Does correctly guessing the 1st draw change the probability of correctly guessing the 2nd? Compute the actual probability of correctly guessing the 2nd draw and compare it to an appropriate estimate to find out.**