# Name:

# Classwork 20: **Fandango**

Let’s start with a data set called **fandango** from [FiveThirtyEight](https://fivethirtyeight.com/features/higher-rates-of-hate-crimes-are-tied-to-income-inequality/). If you want to know more information about these variables at any time, google: **fandango r documentation fivethirtyeight**.

1. Go ahead and use R to look at the first six lines of this data frame. What are the cases in this data frame?
2. What movie rating system do you trust the most: Rotten Tomatoes (out of 100), Metacritic, IMDB, Fandango? Why?
3. What movie rating system do you trust the least: Rotten Tomatoes (out of 100), Metacritic, IMDB, Fandango? Why?
4. My husband thinks Metacritic is most trustworthy. And Fandango is the worst. He argues Fandango is just about selling tickets so you can’t trust a movie that gets a good rating on Fandango. But I have a sneaking suspicion that a good movie will basically get rated as a good movie on all of these rating systems. My hypothesis is that if I know a movie’s Fandango score, I will be able to make a better prediction about their Metacritic rating.

My husband argues that Fandango is trash and I’d be better off knowing a movie’s Rotten Tomatoes score. Then I’ll be able to make a better prediction about its Metacritic rating.

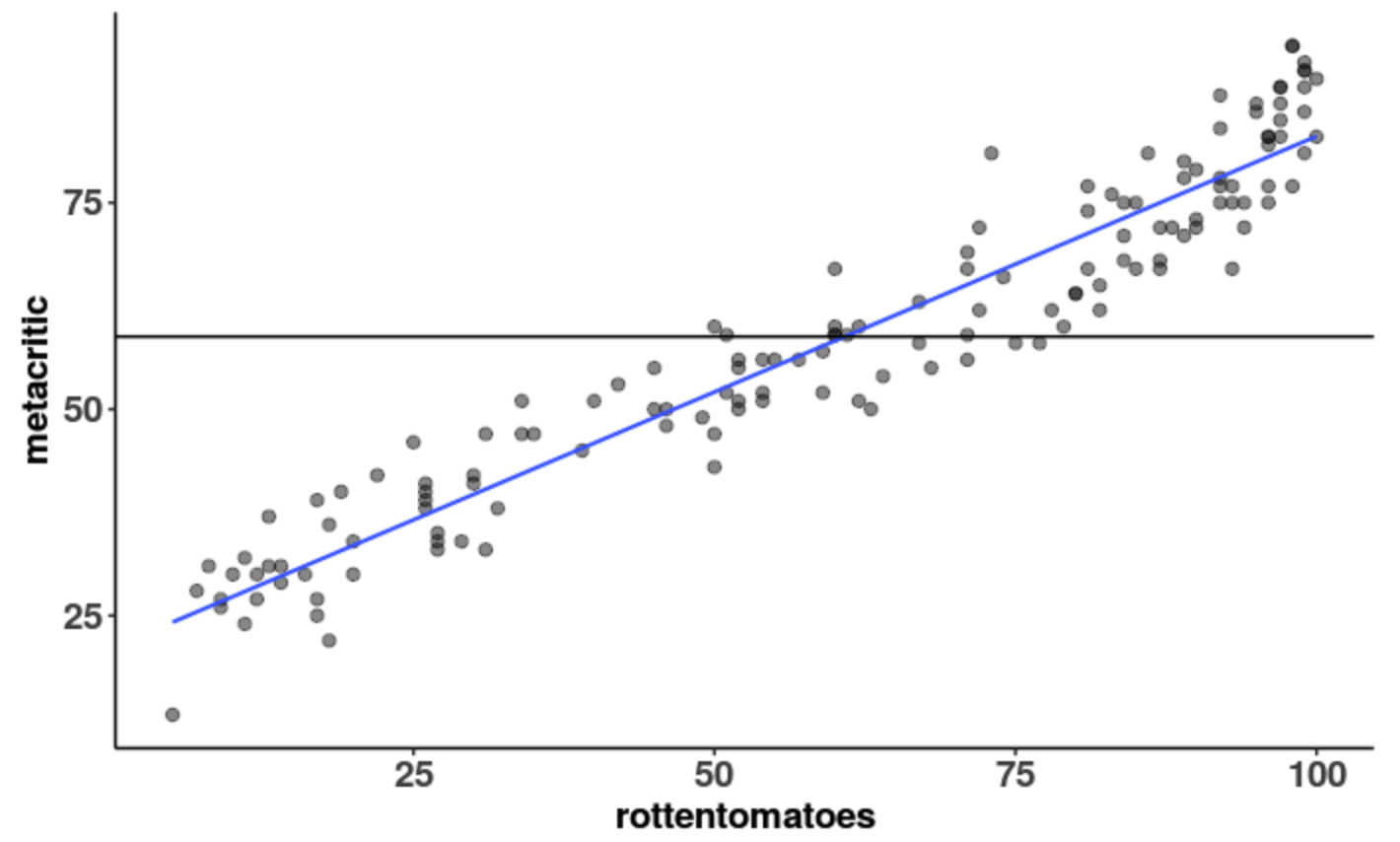
How would you settle this argument?

1. Part of a good marriage is the humility of just being wrong plenty of times. He’s probably right… **rottentomatoes** at least looks like a much better explanatory variable.

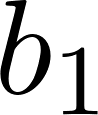
Let’s run through all our skills with this hypothesis!

|  |  |
| --- | --- |
| Word equation |  |
| Making visualizations with R code: |  |
| Specifying the empty model in GLM (no numbers) versus the **rottentomatoes** model (no numbers) |  |
| Finding best fitting parameter estimates with R code (both models) |  |
| Writing best fitting GLM model (both models) |  |
| How much variation have we explained with the **rottentomatoes** model? |  |

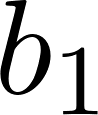
1. Highlight each of these models (empty versus **rottentomatoes**) in the visualization. Draw a few residuals from the empty model and a few residuals from the **rottentomatoes** model.



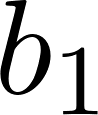
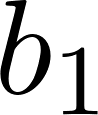
1. Is it possible to have gotten this pattern of data even if there was no relationship between metacritic and rottentomatoes in the DGP? Is it likely?
2. Is it possible to have gotten this pattern of data if the relationship between metacritic and rottentomatoes in the DGP was basically like our sample? Is it likely?
3. We have been exploring two random processes of generating data in our class: shuffle and resampling. Which of these processes is like #7? Which one is like #8?
4. Which of these processes is like the “empty model”?
5. If the true DGP was basically like the empty model, what would be the true value of ?
6. Can we ever know the true value of ?
7. The first scatterplot is the real data. Are the rest shuffled or resampled data?

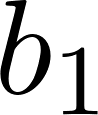
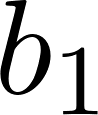
In each box, try to estimate what the [](https://www.codecogs.com/eqnedit.php?latex=b_1%250) is.

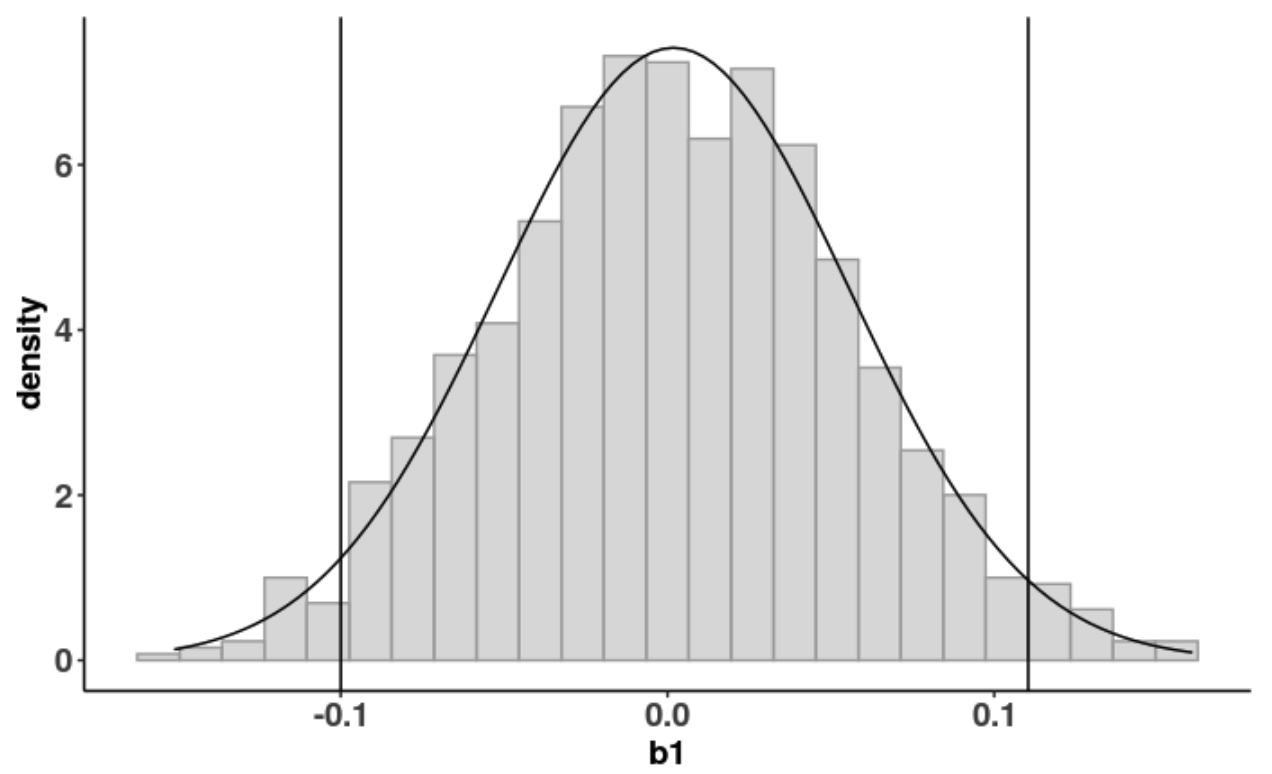
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

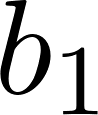
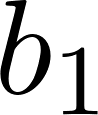
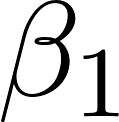
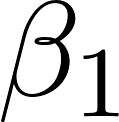
1. Shuffling is as if we live in a world where the **rottentomatoes** can’t help us predict **metacritic**. How would you use the functions **b1()** and **shuffle()** to find the [](https://www.codecogs.com/eqnedit.php?latex=b_1%250) of shuffled data? Write the code.

1. Let’s use the **do()** function to run that code 1000 times. What does the **do()** function do? What does this function give you? A single number? A variable? A data frame? What kind of distribution do we get: a sample, the DGP, or sampling distribution?

1. What do you notice about these [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s? How much do you think the [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s vary?

1. If we visualized this distribution, what do you think the shape, center, and spread of this distribution would be? (This should roughly correspond with your answer to a prior question!)
2. Create a distribution of 1000 shuffled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s and save them in a data frame called **shuff.sdob1**. Write the code here.
3. Here we visualized the 1000 shuffled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s in a histogram. Draw (and label with numbers) the mean and standard error of this distribution.

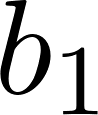
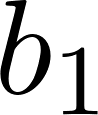


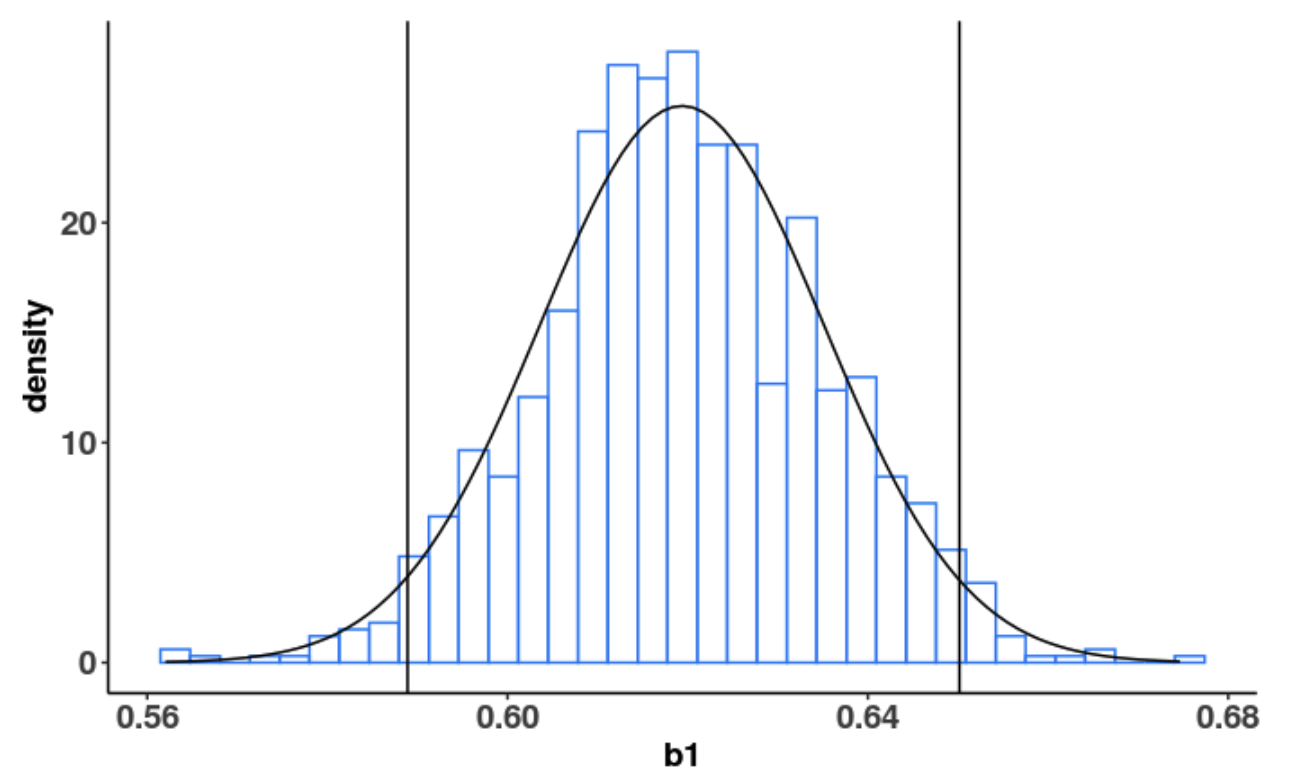
1. Compare what you got with the mean and standard error someone else in the class got. Although you may have gotten very similar numbers, they will be ever so slightly different. Why is this the case?
2. Why are the shuffled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s clustered around 0?
3. If the sampling distribution of shuffled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s are clustered around 0, then what must have been the assumed [](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta_1%250) of our DGP? Does this indicate that 0 is the true [](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta_1%250) of our DGP?

**Shuffle versus Resampling**

1. The first scatterplot is the real data. The rest are resampled data. What is different about the resampled data versus the shuffled data in the previous grid?

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. Create a distribution of 1000 resampled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s and save them in a data frame called **resample.sdob1**. Write the code here.
2. Here we visualized the 1000 resampled [](https://www.codecogs.com/eqnedit.php?latex=b_1%250)s in a histogram. Draw (and label) the mean and standard error of this distribution.



1. If we used the Central Limit Theorem to estimate standard error (instead of actual making a sampling distribution and looking at it ourselves), how would we have calculated standard error?
2. Which sampling distribution has a standard error closer to the CLT’s approximation?
3. Which sampling distribution has a more conservative (aka worst case scenario) standard error?
4. Which technique (resampling versus shuffle) will lead to a confidence interval that is similar to the output of **confint()**?
5. When should we use which one?