Lab 3 - Is It Normal?

# Introduction

This lab will introduce you to your first of many statistical tests, the **Shapiro-Wilk** test for normality. All this test asks is the likelihood of something coming from a population with a normal distribution or, if you prefer, “is it normal?” First this lab will walk you through how to make test data sets so that you can see what happens when you run a Shapiro-Wilk test on data that you know is normal. Then, you will use real salary data from the University of Arizona, determining its summary statistics and whether it is normally distributed.

# Learning Outcomes

By the end of today’s lab you should be able to:

* Identify an obviously non-normal distribution using a histogram
* Use rnorm() to make up fake distribution data, both normal and bimodal
* Use c() to bind together vectors
* Use shapiro.test() to evaluate the likelihood a sample came from a normal distribution
* Use mean(), sd(), median(), max() and min() to summarize data
* Describe what it says about a distribution when the mean and median are very different
* Use sample() to take a sub-sample of a large data set

# Part 1: Making Randomized Data

Sometimes it’s easier to understand statistics when you practice them using made up data sets where you know what the answer should be - sort of like using a walkthrough with an answer key in front of you. Making randomized data is like making your own key, because you know what the answer *should* be. Today we will be using a test to determine if a sample came from a normal distribution, so our first step is to make up data that we know is normally distributed and see how the tests performs on that data.

## Part 1.1 Make the Normal Data

There are many functions in R that make different data distributions. The rnorm() function makes normal data, and takes the arguments of sample size (**n**), mean (**mean**), and standard deviation (**sd**).

If we wanted a sample (called **normaldata**) of 150 values from a normal distribution with a mean of 25 and a standard deviation of 2, we’d write:

normaldata <- rnorm(n = 150, mean = 25, sd = 2)

## Part 1.2 Double Check the Normal Data

First, you’ll want to make sure it looks right, by visualizing the data with a histogram. This R object isn’t a data frame like we’ve used in previous labs. Instead, it is a **vector**. If we want to see how the code worked using a histogram we don’t need to use $ to pull out a column. Instead, enter the object itself, like so:

hist(normaldata)

We can see it looks pretty normal but not perfect, which makes sense - it is, after all, a sample.

## Part 1.3 Make the Non-Normal Data

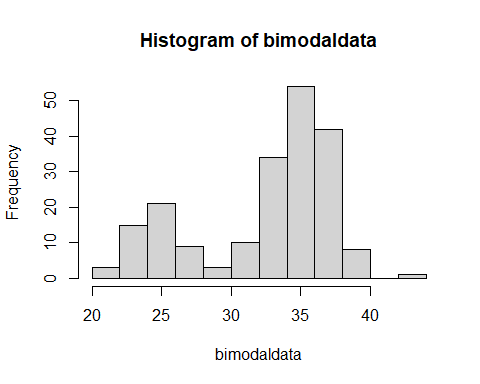
Before we go into testing, you’ll also want to make sure that you have an example of a data set that isn’t actually normal. There are many different types of distributions, but probably the easiest way to make a non-normal data set is to make a **bimodal** dataset.

You can do this pretty easily by making two different normal distributions of random data, with slightly different means. While you can do completely equal modes, for this exercise we will have uneven numbers - **mode1** will have fewer entries than **mode2**.

mode1 <- rnorm(n = 50, mean = 25, sd = 2)  
mode2 <- rnorm(n = 150, mean = 35, sd = 2)

Then, to smoosh them together into a data frame that has both (called **bimodaldata**), you use the c() function, which puts them together nicely.

bimodaldata <- c(mode1, mode2)

Use your hist() code to double check that the bimodal dataset looks right! That should look something like this (but not exactly): 

# Part 2: Summary Statistics on Randomized Data

Before we get into the Shapiro-Wilk test, let’s use some basic R functions on this data to pull out important characteristics.

## Part 2.1 Central Tendency

The mean and median are very easy to find in R - their functions are mean() and median() and you run them like so.

median(normaldata)  
mean(normaldata)  
median(bimodaldata)  
mean(bimodaldata)

### QUESTION 1: Why is it that the mean and median are close together for the normal data, but further apart for the bimodal data?

## Part 2.2 Dispersion

Some simple ways of calculating dispersion in R are to calculate the maximum (max()), the minimum (min()), or the standard deviation (sd()). You run them just like you did above, by placing the object inside the parentheses, i.e.:

max(normaldata)  
min(bimodaldata)  
sd(bimodaldata)

But though all three measure dispersion, if you look at the output there is a big difference between the maximum and the standard deviation. Think about why that is, then try to answer the next question.

### QUESTION 2: Which of the following chunks of code should, statistically, yield something that is most similar to the maximum value?

A. 2\*sd(normaldata)  
B. mean(normaldata)  
C. mean(normaldata) + 2\*sd(normaldata)  
D. median(normaldata)

# Part 3: Shapiro-Wilk Test on Randomized Data

In this class you will learn many different **statistical tests**, each of which asks a different question. This particular test, the **Shapiro-Wilk Normality Test** asks “how likely is it that this sample came from a normally distributed population?”

## Part 3.1 Normal Data

To run it is pretty simple - you use shapiro.test() and put your R object in the parentheses:

shapiro.test(normaldata)

What this does is return a few values:

##   
## Shapiro-Wilk normality test  
##   
## data: normaldata  
## W = 0.98607, p-value = 0.1369

Mine are a little different than yours because this is a random sample, but the pieces are the same:

1. **data**: the name of your data, in this case **normaldata**
2. **W**: the calculated **test statistic**
3. **p-value**: the probability of your results

W is calculated by the Shapiro-Wilk test to measure how normal or not normal your distribution is. Then, the computer takes that W test statistic and says “what’s the probability you would get a W like this from a normal distribution?”

That’s what the **p-value** is: a measure of how frequently the conditions of the test are true.

In my case, I got a *p*-value of 0.14, which is essentially a 14% probability this came from a normal distribution (but remember yours will be slightly different because normaldata is randomized!)

## Part 3.2 Bimodal Data

While different people use different cut-offs, a general rule is that if the *p*-value is less than .05 (or 5% chance of happening randomly), it’s weird. So if you run a Shapiro-Wilk test and get *p* < 0.05, it’s probably (but not necessarily) not from a normal distribution.

And you can test that! You have a dataset that you know does not follow a normal distribution (**bimodaldata**). Use the steps from part 3.1 to find out whether the Shapiro-Wilk test can determine that this data is not from a normally distributed population.

### QUESTION 3: Does the Shapiro-Wilk test tell you that the bimodal dataset is probably from a normal distribution, or not? Use your p-value to support your answer.

# Part 4: Analyses of Real Data

The data you will use for today’s lab comes from <https://www.wildcat.arizona.edu/page/university-of-arizona-salary-database-fy2020>, which has compiled the salaries of all of the staff at University of Arizona for fiscal year 2020. I have done some data cleaning and added a few extra columns, and uploaded this data set as a csv file to D2L.

## Part 4.1 Load the Data

Download the dataset Faculty Salaries FY 2020.csv from D2L. Import your dataset into R using the steps we learned in Labs 1 and 2, and remember to call it something short and easy to spell like “salary” or “fs.” In my case, I’ve called it fs.

fs <- read.csv("Your File Path Here/Faculty Salaries FY 2020.csv")

## Part 4.2 Look at the Data

Use the hist() function to evaluate the data (Lab 2 can help you with directions). You are specifically looking at the column called Annual.at.Full.FTE. You’ll notice it doesn’t really look much like a normal distribution. You might want to use the xlim argument to look in finer detail (see Lab 2).

### QUESTION 4: What makes this data look like it is not likely to be a normal distribution? Use hist() and the xlim or breaks arguments to evaluate each of the answers below and select what is the most accurate description of this data:

1. Most of the data is below 150,000 but there are a few individuals that make more, up to several million a year
2. Most of the data is below 500,000 but there are a few individuals that make more, up to several million a year
3. Almost everyone is making 250,000 dollars a year but there are a few individuals that make more, up to several million a year

## Part 4.3 Summarize the Data

Use the mean(), median(), sd(), max() and min() functions to pull out summary statistics for the annual salary at **full** FTE. Remember that this is a **data frame** and not a **vector** so you need to tell R what column to look at using the $ indicator.

### QUESTION 5: Enter the code you used to find the standard deviation of this dataset.

### QUESTION 6: Enter the code you used to find the mean of this dataset.

### QUESTION 7: The mean and the median of this data are pretty different. That is a good indication that:

1. The data is normally distributed
2. The data is not normally distributed
3. The mean is a better measure of central tendency for this data
4. Something was wrong with your math

## Part 4.4 Shapiro-Wilk Test

So if you try to run a Shapiro-Wilk test on this data, you will run into an error. Try it, using the following code:

shapiro.test(fs$Annual.at.Full.FTE)

Like many tests, the Shapiro-Wilk test has certain restrictions. You can’t run it on a sample size that is too small, and you can’t run it on a sample size that is too large. In this case, you need between 3 and 5000 samples.

One way of getting around this is to take a smaller sample of your much larger dataset. You can do this using the sample() function. sample() takes three separate arguments: what you want to take a sample of (**x**), how big that sample should be (**size**), and whether you can take the same observation twice (**replace**). We do not want to take replacements, so we will set replace to be *FALSE*:

n <- sample(x=fs$Annual.at.Full.FTE, size=4999, replace=FALSE)

Now I have a random sample of 4999 values of the Full FTE salary. Because it is random, your samples will be slightly different from mine, and if you re-run the code you will get different samples over and over and over again. But now that I have my random smaller sub-sample, I can run a Shapiro-Wilk test:

shapiro.test(n)

**NOTE**: Because n is a value string, you don’t need to use the $ in this code! Just enter the r object.

### QUESTION 8: Take a random sample of 4999 values and run a Shapiro-Wilk test on it. Is your random sample of salary data normally distributed?

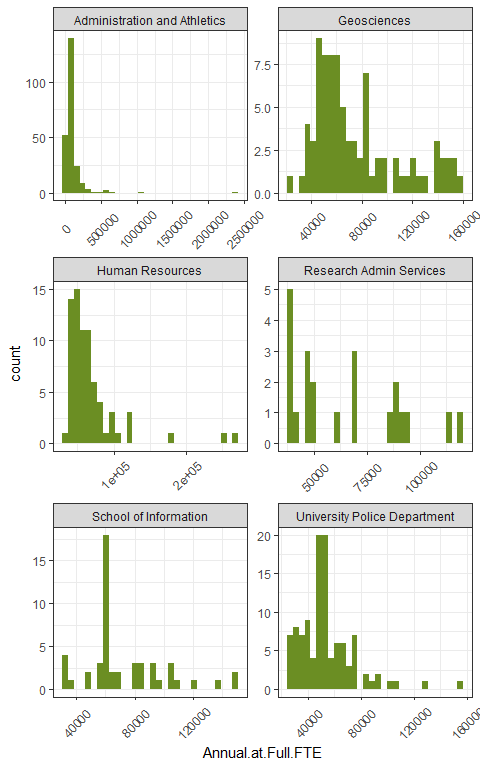
1. Yes, my p value is less than .05 meaning the data is likely from a normally distributed population
2. No, my p value is less than .05 meaning there is a less than 5% probability of this sample coming from a normally distributed population

### QUESTION 9: Is there a chance that someone in this class will get a p value of more than .05 for their subsample in this class?

1. Yes, but it is unlikely because the sample size is so small compared to the original highly skewed dataset
2. Yes, but it is unlikely because of how large the sample size is, which captures much of the original highly skewed dataset
3. No, there is no chance because the data is so skewed

## Part 4.5 Understanding Data

Salary analyses are an important part of many companies decision-making processes. Salary analyses can pinpoint problems like racial or gender inequity, show when company resources are going to the wrong area, and can help inform whether certain groups of people need to have a raise or not. *Almost all salary studies in America show a non-normal distribution.* In fact, if you look at individual departments almost all of them will be non-normally distributed too. Here are 6 example departments at UA:



There are some common explanations for salary data’s non-normal distributions. They almost always have the bulk of the salaries in lower-paid brackets, and then a few longer tails. Evaluating the difference between the uppermost and lowermost values can tell you a lot about the extent and shape of salary distributions.

Remember that you can use R like a calculator to add values together. For example, if you wanted to find out what the difference was between the maximum and minimum of the bimodal dataset, you could use the / (forward slash) to divide them, or - (minus sign) to subtract them:

max(bimodaldata) / min(bimodaldata)  
max(bimodaldata) - min(bimodaldata)

### QUESTION 10: Use the max() and min() functions on Annual.at.Full.FTE to find out how much the lowest and the highest paid employees at UA make, then fill in the blank: The highest paid person at UA makes \_\_\_ times as much money as the lowest paid person at UA.

However, there are actually two columns of salary data here: the “full FTE”, and “actual FTE”. This is because some people do not work full-time. So the amount of money that people take home is not always as much as their full FTE.

### QUESTION 11: As in question 10, look at the maximums and minimums. This time, look at the Annual.at.Actual.FTE column and fill in the answer: When considering full time versus part time pay, the highest paid person at UA takes home \_\_\_ times as much money as the lowest paid person at UA.

In 2020, a furlough was broadly applied to employees of the university to help offset anticipated losses from COVID-19. The furlough amounts ranged between 5% and 20%, depending on the faculty salary. This is represented in the **NewSalary** column. This amount was applied to their actual pay, not the full-time salary expectation. Employees that made less than a certain amount each year did not have a salary cut, as it would have made their salaries unlivable.

### QUESTION 12: As in question 10, look at the maximums and minimums of the NewSalary column to fill in the answer: after the salary adjustments for the furlough, the highest paid person at UA makes \_\_\_ times as much money as the lowest paid person at UA.

Large-scale salary differences are expensive to correct. One could raise minimum salaries, but as that’s usually where most workers are it becomes very quickly expensive to do that. One could also lower or cap maximum salaries, but then risk losing higher-paid staff to other universities without caps. But these aren’t the only options!

### QUESTION 13: Without increasing or decreasing anyone’s salaries, what is one way that UA could shrink the difference between maximum and minimum salaries at the university?

That’s it! Upload your answers to D2l.