Stat 230 Lab 1- Review of SLR and R

P.B. Matheson grateful to and adapted from A.S. Wagaman

Instructions: This lab exercise is a warm-up designed to give you some familiarity working in R and RStudio. If you have seen R before, these problems should not be very hard. If you have not seen R before, these problems will give you some practice and help to develop your skills. The content will be a brief review of introductory statistics, so it should be familiar to all.

This assignment, like all assignments that you complete in this course, is prepared in R Markdown, a markup language that allows you to easily format your code for readability and reproducibility. For more information about formatting in R Markdown, click on the **?** button in the menu bar of your Markdown document in RStudio. The benefit of using R Markdown is that it enables you to seamlessly combine R commands, the output from those commands, and written analysis into one document. You should save the file after you open it somewhere you like and keep saving your work as you go!

Introductory Example - Sleep

Introduction NOTE: R is a programming language - there are often MULTIPLE ways to accomplish a given task. You can choose to work with any set of commands that works. Use your class notes and examples too! We are using ggformula for plots (you can use ggplot2 if you know that as well), so that may be different from what you have seen before.

We will almost always begin by loading the **mosaic** package. This is a useful add-on for R. Make sure to include the `include = FALSE` option in the chunk that loads **mosaic**. This will suppress unwanted messages and keep your Markdown document neat and clean. Above, you can see how this is done. I've also set some other useful settings and loaded another package that we may use.

Note that R is case-sensitive. That means capitalization and spelling count! The # in a line of code is a commenting convention. R ignores anything after the # as a comment, but it will still show up in the line of code. There are other ways to make code not show up that you can learn about.

The lines below under *calculations* make up a **code chunk**, that is, a block of R code embedded into the RMarkdown file. Once you click on `Knit PDF` (or a different Knit option) at the top, you'll see the generated file which not only contains code from your code chunk, but also the processed R output. However, you usually only want to knit when you are ready to get a full version of the document, not just as you are working.

You can get the output as you go in the console window (below) or in line (ask me if you want and can't get it) by running each chunk without knitting. Just click the run button while in a chunk or on a line, or use the green arrow or the keyboard shortcuts (Ctrl+Enter for a single line, etc. (check out the run menu)). For the sections below, try running with the green arrow in each chunk, or line by line.

Calculations - these can be done here in an R chunk or in the console;

R is first and foremost a (scientific) calculator. Let's try a few things:

```
3 + 5
```

```
## [1] 8
```

```
2 + 5*2
```

```
## [1] 12
```

```
(2+5)*2
```

```
## [1] 14
```

```
2^5
```

```
## [1] 32
```

Note that R doesn't understand multiplication without an asterisk. Typing 2(5) will give an error, not the value of 10.

Creating variables

We can store values (like our calculations). To do this, we use the <- operator for assignment. Think of it as "put things there".

```
x <- 3  # this creates a variable `x` with the value 3 stored
x  # if we just type `x` without an assignment, we can now see what's stored in `x`
```

```
## [1] 3
```

```
# y # would give an error because there is no "y" yet
```

It's often useful to create a variable that saves the output of a calculation:

```
y <- 2 + 5 * 2
```

Reading in and Working with Data

Let's start by loading some data. We can either load data that already exists in R using the data function or read in datasets that are provided elsewhere. To get us started, I have a dataset about sleep deprivation. It is saved on my web drive pmatheson.people.amherst.edu/stat230/ as sleep.csv A csv file is a file that can be opened in Excel but is saved in a special way (comma separated values). You will always have to tell R what kind of data you are asking for (a.k.a. "read.csv").

> Lingering effects of sleep deprivation? Researchers have established that sleep deprivation has a harmful effect on visual learning. But do these effects linger for several days, or can a person "make up" for sleep deprivation by getting a full night's sleep in subsequent nights? A recent study by Stickgold and colleagues investigated this question by randomly assigning 21 subjects (volunteers between the ages of 18 and 25) to one of two groups: one group was deprived of sleep

> on the night following training and pre-testing with a visual discrimination task, and the other group was permitted unrestricted sleep on that first night. Both groups were then allowed as much sleep as they wanted on the following two nights. All subjects were then re-tested on the third day. Subjects' performance on the test was recorded as the minimum time (in milliseconds) between stimuli appearing on a computer screen for which they could accurately report what they had seen on the screen. The sorted data presented here are the improvements in those reporting times between the pre-test and the post-test (a negative value indicates a decrease in performance).

Sleep deprivation (n=11): -14.7, -10.7, -10.7, 2.2, 2.4, 4.5, 7.2, 9.6, 10.0, 21.3, 21.8

Unrestricted sleep (n=10): -7.0, 11.6, 12.1, 12.6, 14.5, 18.6, 25.2, 30.5, 34.5, 45.6

Does it appear that subjects who get unrestricted sleep on the first night tended to have higher improvement scores than subjects who were sleep deprived on the first night? Use the CHOOSE-FIT-ASSESS-USE framework to asses that question and learn about R as you go! No need to enter the data in manually, it will be accessed online from the next R command. Remember, the code below will run if you click Run from your menu, or put your cursor in it and hit Ctrl+Alt+C or hit Run Current Chunk from the Chunks menu. Remember to run the chunks as you work, or no output will appear!

```
sleep <- read.csv("https://pmatheson.people.amherst.edu/stat230/sleep.csv")
```

sleep is now in our workspaces as a *data.frame*. There are other functions to read in data, but this works for our .csv file. This is the basic data structure that we will work with most often. There are several options for figuring out what is in the **sleep** data frame. The easiest way is to click on the name of it in the Environment tab (at right).

> Note that clicking on the name of the data set in the Environment tab will allow you to view it like a spreadsheet. This is fine for small data sets, but will not work for large ones! If the data set was provided with R, then typing in help(datasetname) in the console will open up documentation in the Help window. For data provided within R, you may need to run the View(Dataset) command before you can see it.

```
# Some useful R commands that work at the dataset level - Remember to RUN this
# What type of object is it?
class(sleep)
```

```
## [1] "data.frame"
```

```
# What are its dimensions?
dim(sleep)
```

```
## [1] 21  3
```

```
# How many rows and columns?
nrow(sleep)
```

```
## [1] 21
```

```
ncol(sleep)
```

```
## [1] 3
```

```
# What variables are present?
names(sleep)
```

```
## [1] "X"       "deprive" "discrim"
```

```
# A brief summary of each variable
summary(sleep)
```

```
##        X        deprive             discrim
##  Min.   : 1   Length:21          Min.   :-14.70
##  1st Qu.: 6   Class :character   1st Qu.:  2.40
##  Median :11   Mode  :character   Median : 11.60
##  Mean   :11                      Mean   : 11.48
##  3rd Qu.:16                      3rd Qu.: 21.30
##  Max.   :21                      Max.   : 45.60
```

```
# A glimpse of the data set; no summary info but has variable types
glimpse(sleep)
```

```
## Rows: 21
## Columns: 3
## $ X       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
## $ deprive <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",~
## $ discrim <dbl> -14.7, -10.7, -10.7, 2.2, 2.4, 4.5, 7.2, 9.6, 10.0, 21.3, 21.8~
```

```
# Similar to glimpse
str(sleep)
```

```
## 'data.frame':    21 obs. of  3 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ deprive: chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ discrim: num  -14.7 -10.7 -10.7 2.2 2.4 ...
```

```
# A sample of the first few rows
head(sleep)
```

```
##   X deprive discrim
## 1 1     Yes   -14.7
## 2 2     Yes   -10.7
## 3 3     Yes   -10.7
## 4 4     Yes     2.2
## 5 5     Yes     2.4
## 6 6     Yes     4.5
```

Univariate Analysis We will use the *formula* interface whenever possible. This means using the ~ (tilde) operator. To get a basic summary of a quantitative variable, use *favstats()*. If you get an error that R can't find favstats, it means the mosaic package wasn't loaded. This is a command that only works with mosaic. Look in the first R chunk above and you will see library(mosaic).

```
favstats(~ discrim, data = sleep) #summary of discrim variable in sleep data set
```

```
##    min  Q1 median   Q3  max     mean       sd  n missing
##  -14.7 2.4   11.6 21.3 45.6 11.48095 15.42827 21       0
```

To specifically retrieve the mean or standard deviation, use the respective commands.

```
mean(~ discrim, data = sleep)
```

```
## [1] 11.48095
```

```
sd(~ discrim, data = sleep)
```

```
## [1] 15.42827
```

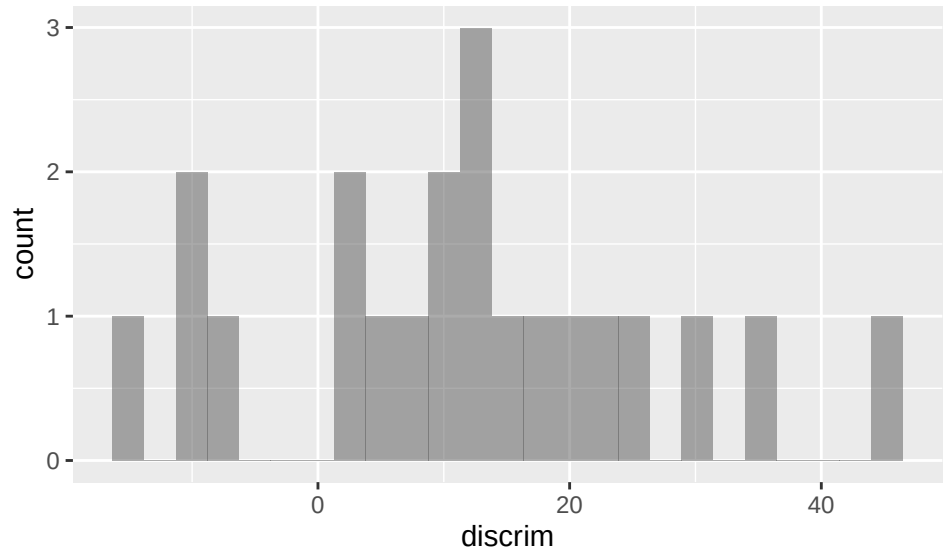
Note that you can store these values to new variables if you want to use them later on. When you execute the code chunk below, you will see "mu" pop up as a new value in the workspace/environment window. Unless you enter the second command "mu" R won't show you what mu is.

```
mu <- mean(~ discrim, data = sleep)
mu
```

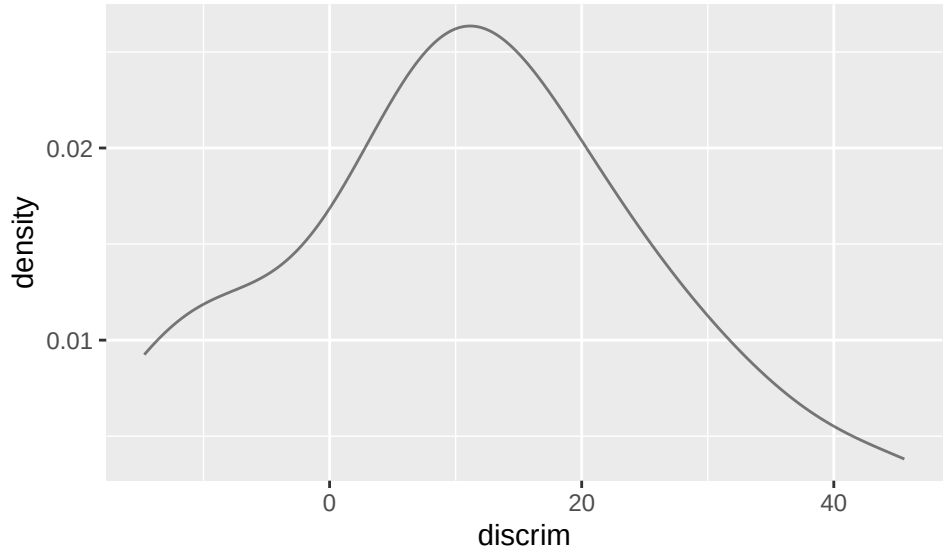
```
## [1] 11.48095
```

For a graphical description of the distribution of a single variable, we can create a histogram or a density plot. Once you generate plots, you can cycle through them using the arrow keys in the Plots window.

```
gf_histogram(~ discrim, data = sleep) #gf_dhistogram will give the density version of a histogram; defa
```

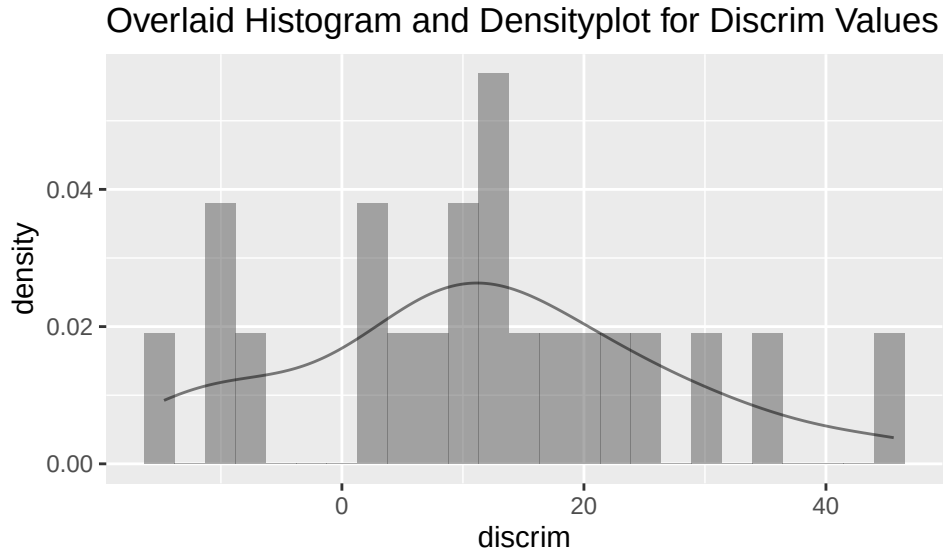


```
gf_dens(~ discrim, data = sleep)
```



The gf_ prefix is for our usage of ggformula. We can do lots of neat things with our plots, such as overlay them and give them titles. Here is an example. Note that this uses the piping operator %>%, which passes the output from one command into another.

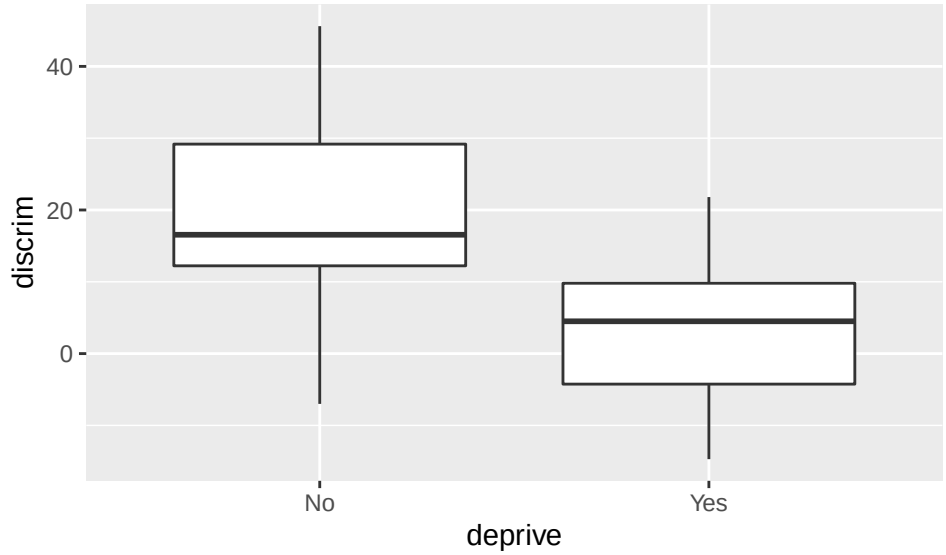
```
gf_dhistogram(~ discrim, data = sleep) %>% #note swap to gf_dhistogram for scaling
  gf_dens() %>%
  gf_labs(title = "Overlaid Histogram and Densityplot for Discrim Values")
```



No need to memorize these commands. You aren't coding from scratch in our class. Running the labs and working on the homework will add them to your library. If you go to your history tab in the upper right window you can search on commands. For now focus on learning what the sets of commands do, generally speaking, then when you want to do something similar (but say, on a different data set), copy and paste the code you have as an example and edit the variables and data set to your situation.

Bivariate Analysis For relating a quantitative variable to a categorical variable, we can draw a boxplot.

```
gf_boxplot(discrim ~ deprive, data = sleep)
```



Descriptive statistics by group can also be useful here.

```
favstats(discrim ~ deprive, data = sleep)
```

```
##   deprive   min     Q1 median     Q3  max  mean       sd  n missing
## 1      No  -7.0 12.225  16.55 29.175 45.6 19.82 14.72532 10       0
## 2     Yes -14.7 -4.250   4.50  9.800 21.8  3.90 12.17185 11       0
```

Finally, for counting the frequency of categorical variables, you can use *tally()*.

```
tally(~ deprive, data = sleep)
```

```
## deprive
##  No Yes
##  10  11
```

If you want a contingency table, it would be ~ firstvariable + secondvariable in the command above. We'll come back to bivariate analysis for scatterplots and regression in the next example. For now, consider what you've seen with the analysis so far in relation to the question:
1. "Does it appear that subjects who get unrestricted sleep on the first night tended to have higher improvement scores than subjects who were sleep deprived on the first night?" Is there an appropriate model we can fit (or test we can do)?

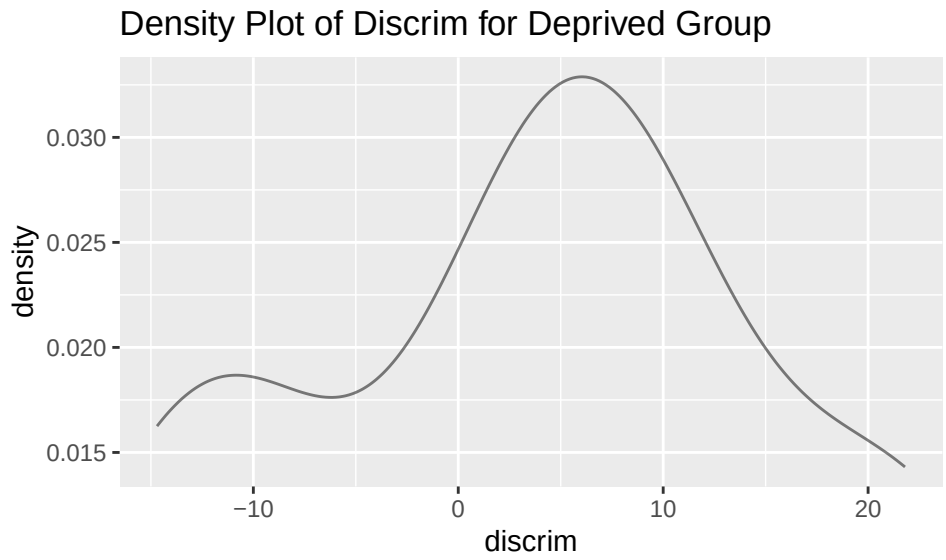
ANSWER:

PUT YOUR ANSWER HERE

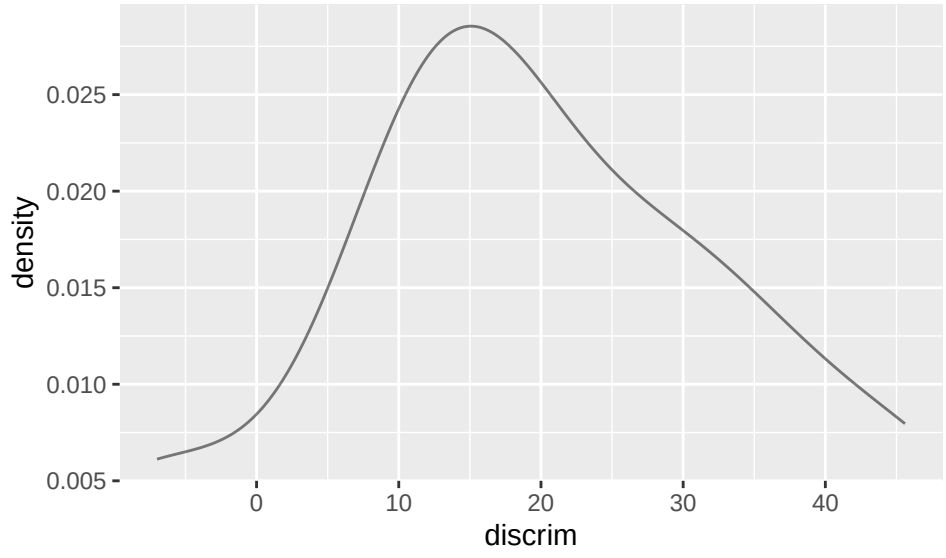
Based off the class and book examples, the answer should be yes! You should be thinking about applying a two-independent samples t-test. From a model standpoint, the response is simply a constant (mean) that differs between the two groups plus errors that are normally distributed with possibly different SDs for the two groups. You may not have seen it presented that way before, but that's the underlying model!

Be sure to return and ASSESS your conditions with appropriate plots before Using the model via the t-test. Try the plots below along with your boxplot.

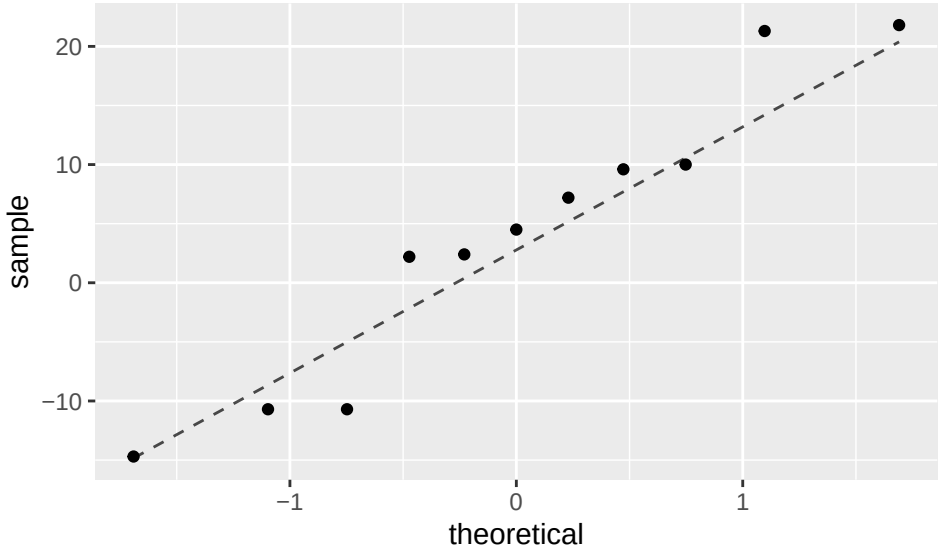
```
gf_dens(~ discrim, data = filter(sleep, deprive == "Yes")) %>%
  gf_labs(title = "Density Plot of Discrim for Deprived Group")
```



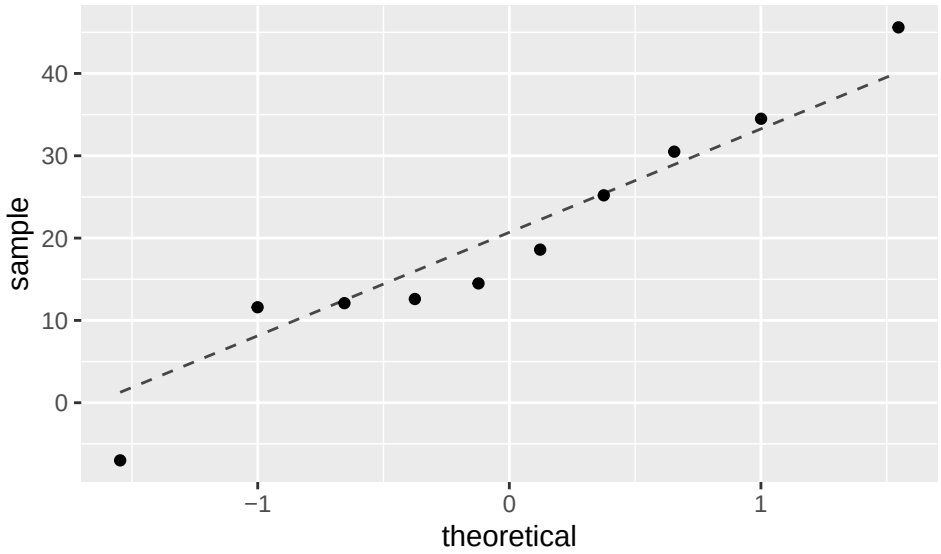

```
gf_dens(~ discrim, data = filter(sleep, deprive == "No"))
```



```
gf_qq(~ discrim, data = filter(sleep, deprive == "Yes")) %>% gf_qqline()
```




```
gf_qq(~ discrim, data = filter(sleep, deprive == "No")) %>% gf_qqline()
```



filter is a data wrangling verb (function) that helps us to subset the data set.

Sometimes when we use functions we will use *with* to pull from the correct data set. In these situations, this is done so that you don't have to type the name of the *data.frame* repeatedly. Again, there are LOTS of ways to accomplish what you want in R.

2. Do you have any concerns regarding the t-test conditions based on your plots?

   ANSWER:

PUT YOUR ANSWER HERE

Assuming you have no major concerns, we run the test using the t.test command.

```
t.test(discrim ~ deprive, data = sleep)
```

```
##
##  Welch Two Sample t-test
##
## data:  discrim by deprive
## t = 2.6851, df = 17.557, p-value = 0.01535
## alternative hypothesis: true difference in means between group No and group Yes is not equal to 0
## 95 percent confidence interval:
##   3.441222 28.398778
## sample estimates:
##  mean in group No mean in group Yes
##             19.82              3.90
```

3. What conclusion do you reach?

   ANSWER:

PUT YOUR ANSWER HERE

Regression Example - Galton Data

Our class will focus more on linear regression, so we should consider a linear regression example to get familiar with the relevant code.

The **Galton** data set is part of the **mosaic** package, and contains data on the height of many British adults, as well as the height of their mothers and fathers. To load it, we use the *data()* command, because we already loaded the mosaicData package.

```
data(Galton)
#help(Galton) #when uncommented, brings up the help menu on the Galton data set
#View(Galton) #when uncommented, allows you to view the data set instead of having to click
#Be sure the above two lines are commented off when knitting/compiling
```

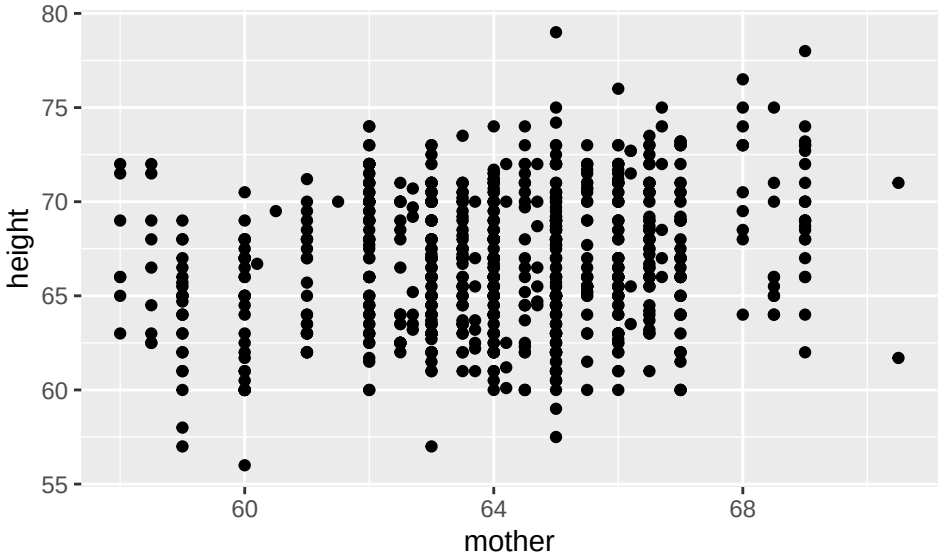
We want to understand the relationship between the height of an adult and their mother (two quantitative variables) and we want to see if the child's adult height can be predicted by the mother's height.

Bivariate Analysis

CHOOSE:

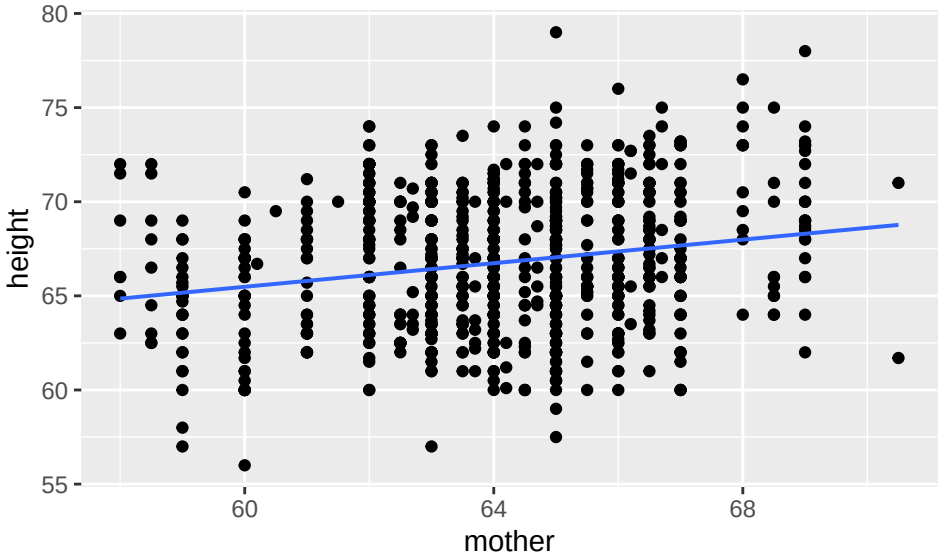
For assessing the relationship between two variables, we can draw a scatterplot.

```
gf_point(height ~ mother, data = Galton)
```



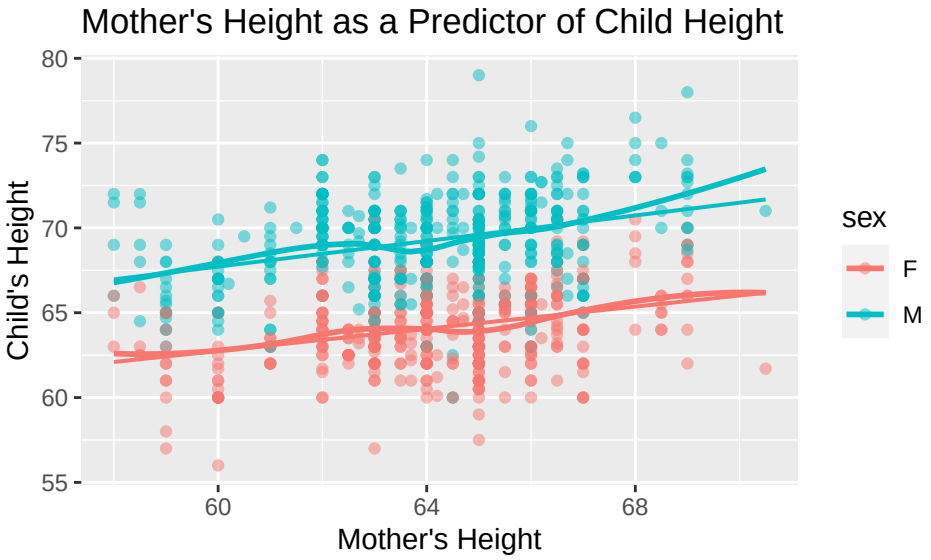
We can add the regression line easily to this plot.

```
gf_point(height ~ mother, data = Galton) %>%
  gf_lm()
```



Note that there are many graphical options that we can add to this plot. Here is the same basic plot, but jazzed up a bit. You should experiment with graphical options! If you don't know what is available, hit the tab key inside the function call when trying the command in the console.

```
gf_point(height ~ mother, data = Galton, color = ~ sex, alpha = 0.5, xlab = "Mother's Height", ylab = "
  gf_lm() %>% #adds linear regression line
  gf_smooth(method = "loess") %>% #adds a smoother; smoothed line
  gf_labs(title = "Mother's Height as a Predictor of Child Height")
```



Note that in this case we have conditioned on a third variable **sex**, via the color argument. (Bear in mind we are using this variable as provided - sex is not really binary but this data set presents it as such.)

For a numerical assessment of the linear relationship between two quantitative variables, we can compute their correlation coefficient.

```
cor(height ~ mother, data = Galton)
```

```
## [1] 0.2016549
```

Sometimes the *cor* function will not work due to missing data in the data frame. When that happens, you can still get the correlation between all pairs of observations without missing data like this:

```
cor(height ~ mother, data = Galton, use = "pairwise.complete.obs")
```

```
## [1] 0.2016549
```

Based on your work so far, a simple linear regression model is reasonable, but you may suspect it may not have a great fit. Let's find out!

FIT:

Now let's fit the model. This is probably the simplest part. Note that *fm* stands for fitted model, but you can use ANY name you want for the model. Common examples you will see in class are things like *mod*, *mod1*, *mymod*, etc. Just be sure to keep track of what name you are using for calling the summaries and other related commands.

```
fm <- lm(height ~ mother, data = Galton)
msummary(fm) #need the summary
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.69077    3.25874  14.328  < 2e-16 ***
## mother       0.31318    0.05082   6.163 1.08e-09 ***
```

```
##
## Residual standard error: 3.511 on 896 degrees of freedom
## Multiple R-squared:  0.04066,    Adjusted R-squared:  0.03959
## F-statistic: 37.98 on 1 and 896 DF,  p-value: 1.079e-09
```

5. What is the equation of the fitted regression line?

   ANSWER:

PUT YOUR ANSWER HERE

6. How would you interpret the slope?

   ANSWER:

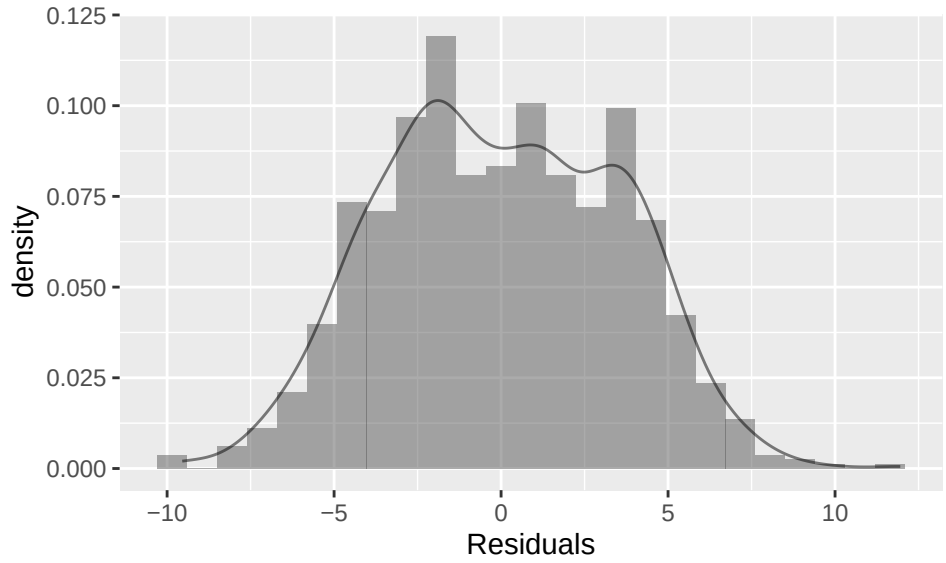
7. PUT YOUR ANSWER HERE

Before we can USE the model, we need to ASSESS it.

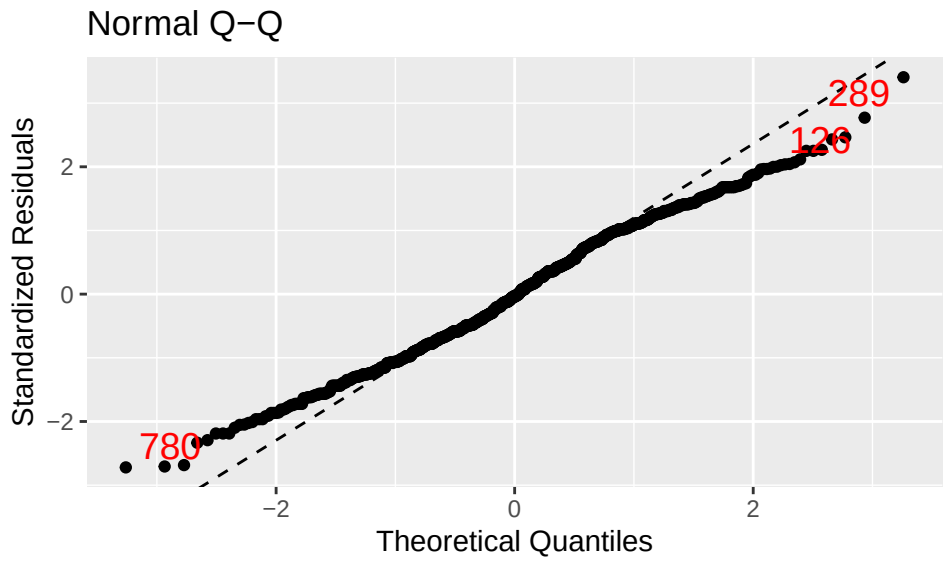
   ASSESS:

Normality of errors condition:

```
#generates a histogram with fitted density curve
gf_dhistogram(~ residuals(fm), xlab = "Residuals") %>%
  gf_dens()
```



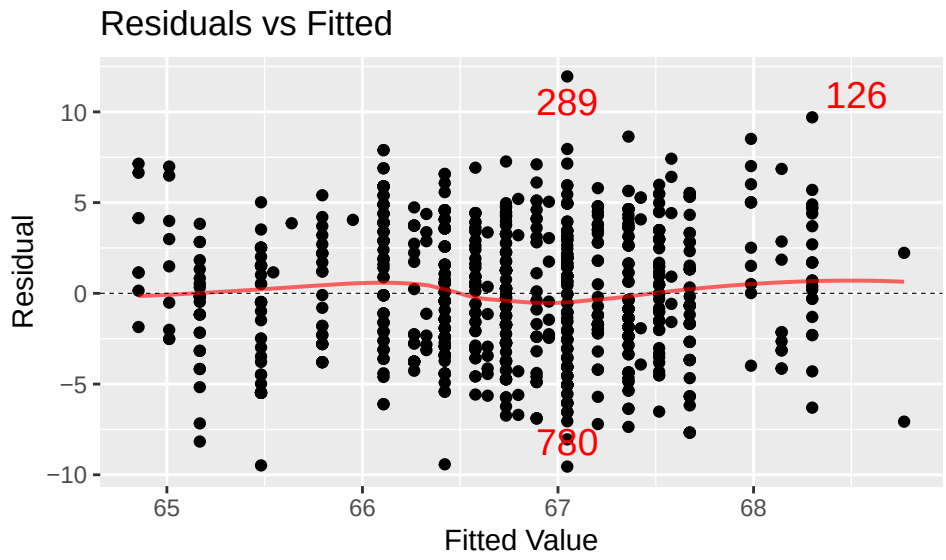

```
#generates the associated qqplot
mplot(fm, which = 2)
```



Constant Variance (and linearity) condition:

```
#Residuals vs. fitted plot
mplot(fm, which = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



8. What do you think about the conditions necessary for simple linear regression (SLR) to be appropriate?

   Remember: the required conditions are linearity of relationship, zero mean for errors (which we have because we are using least squares), constant variance of errors, and independence of errors.

   ANSWER:

PUT YOUR ANSWER HERE

9. How well satisfied are the additional conditions necessary for inference in this regression?

   Remember: the additional conditions are randomness of the data, and normality of the error terms.

   ANSWER:

PUT YOUR ANSWER HERE

Assuming no major issue with the conditions, USE the model to determine if adult height can be predicted by mother's height.

   USE:

10. You should already have the output needed for this. What values do you need to support your response?

    ANSWER:

PUT YOUR ANSWER HERE

11. Review: Do you think the predictions will be very accurate? There should be TWO values in the output that you are using as a gauge of this.

    ANSWER:

PUT YOUR ANSWER HERE

What if we wanted to make some predictions ourselves? There are multiple ways to do this, but the easiest way is to create a function that computes it. Using the Mosaic package, this is very easy. Suppose we had a mother's height of 64 and wanted a predicted child's height.

```
#creates a function that computes predicted values
fit.height <- makeFun(fm)
#obtain the predicted value of height when mother is 64 - USE
fit.height(mother = 64)
```

```
##        1
## 66.73426
```

The *augment* function in the broom package is another way we can get predicted values easily IF the value of interest for a variable is already in the data set. *augment* saves fitted values and residuals for the model and the data set used (which removes observations with missing values for the variables used) to a new data set that you can look at.

```
augmentGalton <- augment(fm)
head(augmentGalton)
```

```
## # A tibble: 6 x 8
##   height mother .fitted .resid    .hat .sigma  .cooksd .std.resid
##    <dbl>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>    <dbl>      <dbl>
## 1   73.2   67      67.7   5.53 0.00289   3.51 0.00361       1.58
## 2   69.2   67      67.7   1.53 0.00289   3.51 0.000275      0.435
## 3   69     67      67.7   1.33 0.00289   3.51 0.000208      0.378
## 4   69     67      67.7   1.33 0.00289   3.51 0.000208      0.378
## 5   73.5   66.5    67.5   5.98 0.00234   3.51 0.00341       1.71
## 6   72.5   66.5    67.5   4.98 0.00234   3.51 0.00236       1.42
```

Here, to find a prediction for when mother is 64, we just have to find an observation with mother of 64, which looks to happen for observation 9. (Found by just viewing the data set.)

```
augmentGalton[9, ] #says give row 9, but all columns
```

```
## # A tibble: 1 x 8
##   height mother .fitted .resid    .hat .sigma  .cooksd .std.resid
##    <dbl>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>    <dbl>      <dbl>
## 1     71     64    66.7   4.27 0.00112   3.51 0.000825       1.22
```

Later, we will learn to use the predict function (in case the value of interest is not already in the data set), which is a more flexible R function.

12. Obtain the predicted value for height when the value of mother takes it's minimum. (Hint: you have to find the minimum first!)

    ANSWER:

PUT bYOUR ANSWER HERE

Exercise For You To Do - KidsFeet

These problems use the **KidsFeet** data set, which is also part of **mosaic**.

```
data(KidsFeet)
#help(KidsFeet)
```

A correct answer to questions a-e consists of a(n) R command(s) that answers the question. For parts f and g, you will need to perform an appropriate analysis. Follow the CHOOSE-FIT-ASSESS-USE framework for part g (f is optional - it's review from Stat 135). Use the code above to help you out, and ask me if you run into any problems!

Problems

a. How many children are present in the data set?

b. What is the mean foot length?

c. For how many children is their left foot bigger than their right?

d. What percentage of the children were left-handed?

e. Which month contained the most births?

f. Were boys' feet longer, on average, than girls'? If so, by how much? Is the result statistically significant?

g. Draw a scatterplot of foot length as a function of foot width. Put the least squares regression line on the plot and provide meaningful axis labels. Is foot width a significant predictor of foot length?

Five Minute Warning: SAVE YOUR WORK - How this works

As you work through the lab, you should save the text you type in/commands, etc. That's as easy as clicking the usual save disc. If you want to see your final printed document, hit the "Knit" button in your menu. I recommend choosing the "Knit PDF" option. It will take a while to compile, but you'll end up with a document that you could then upload to Moodle or print, if you desired, after exporting it. You'll need to learn to upload the compiled .pdfs for homework. Proofread your document before submitting to Moodle!

Finally, for our class, do NOT submit the raw .Rmd files. Submit the compiled (knitted) .pdfs, with appropriate supporting work.

This lab (and most labs) are not collected.