Unit 1 Lab Solutions (Based on revised 2017 data)

James Molyneux[1] & Heidi Estevez[2][3]

[1] UCLA Department of Statistics
[2] UCLA Center X
[3] Los Angeles Unified School District

Updated: October 8, 2017

Lab 1A - Data, Code & RStudio

```
data(cdc)

View(cdc)
```

- **Describe the data that appeared after running `View(cdc)`:**
  - **Who is the information about?**
  - **What sorts of information about them was collected?**

The data that appears shows information from the CDC Youth Risk Behavior survey. Each row represents a single youth who participated in the survey. Each column describes one aspect about the youth's lives that the CDC measure. Information collected ranged from physical characteristics such as `height`, `weight`, `age`, `grade`, etc. and behavioral characteristics such as *how many hours do you sleep?* (`hours_sleep`), *have you felt depressed for two weeks in a row or more during the past 12 months* (`depressed`), etc.

To find out more information about the `cdc` data, type `?cdc` in the console.

- **How are observations represented in our data?**
- **What does the first column tell us about our observations?**
- **How often did our first observation wear a seatbelt while riding in a car?**

To answer these questions, we need to to run `View(cdc)` and look at our data. Each row represents a single student who participated in the survey. The first column describes the student's age. Finally, if we look along the first row and find the `seat_belt` column, we find that the first student listed in our data Always wore their seat belt.

- **How many students are in our `cdc` data set?**
- **How many variables were measured for each student?**

Our data measures 33 variables of information on 15624 students.

```
dim(cdc)
nrow(cdc)
ncol(cdc)
names(cdc)
```

- **Which of these functions tell us the number of observations in our data?**
- **Which of these functions tell us the number of variables?**

The `ncol` and `names` functions will tell us the number of variables. The `nrow` function tells us the number of observations. The `dim` function tells us both.

```
Names(cdc) # Gives an error
NAMES(cdc) # Gives an error
names(cdc) # Works
names(CDC) # Gives an error
```

The only line of code that works is the 3rd line. The others give errors because of various punctuation reasons.

- **Which of these plots would be useful for answer the question: *Is it unusual for students in the CDC dataset to be taller tha 1.8 meters?***

```
histogram(~height, data = cdc)
bargraph(~drive_text, data = cdc)
xyplot(weight ~ height, data = cdc)
```

- **Do you think it's unusual for students in the data to be taller than 1.8 meters? Why or why not?**

Since the question asks about the variable `height` and not the relationship between `height` and `weight`, the best plot to use would be the `histogram`.

Students being 1.8 meters or taller isn't so rare that we would call it *unusual* but we definitely wouldn't say students of that height happen very often.

On your own:

- **What is public health and do we collect data about it?**

Public health, as the name implies, studies the prevalence of health related topics in the public at large. There's a variety of ways to collect data about public health such as tabulating medical records from hospitals, but in this lab the data were the results of a survey.

- **How do you think our data was collected? Does it include every high school aged student in the US?**

Our data certainly doesn't collect information on every student in the US. Instead, the CDC chose schools and students at random and asked them to fill out a survey. **(*Technical side-note: The actual survey method is some sort of stratified sampling but that's really beyond the scope of this course.*)

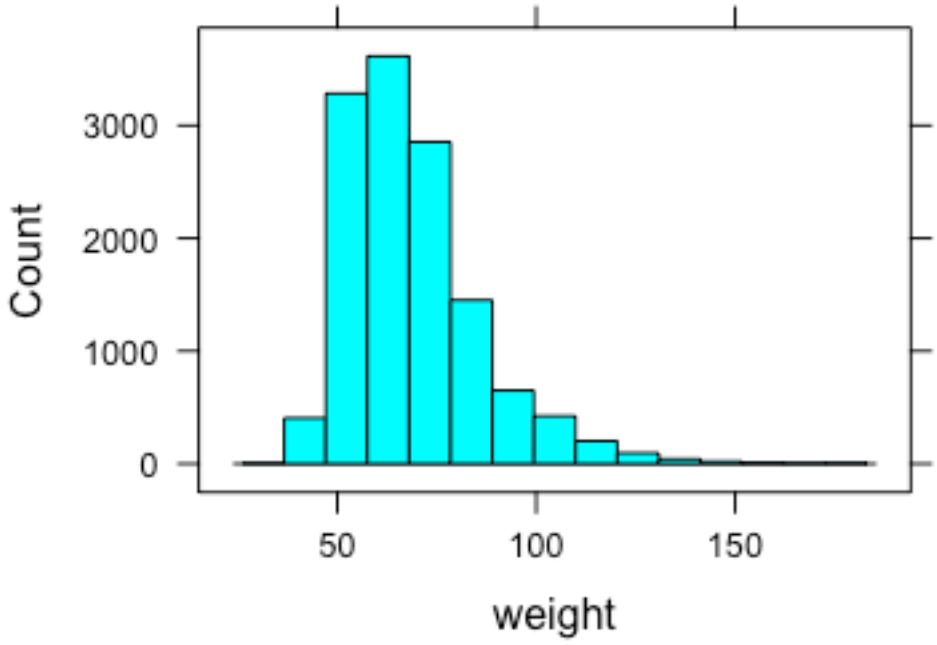
- **How might the CDC use this data? Who else could benefit from using this data?**

The CDC uses this data to get an overall picture of the health and well-being of young people in the US. One reason they might collect this info is to help them decide which risky behaviors are most prevalent and design campaigns to curtail them. The data can also be used with previous surveys to gauge the effectiveness of previous campaigns to encourage or discourage certain behaviors.

Others who benefit from this data might be sociologists studying how different behaviors span across different sexes or races. In this course, we use the data to help us teach a variety of statistical and computational methods to high school students.

- **Write the code to visualize the distribution of weights of the students in the CDC data with a histogram. What is the typical weight?**

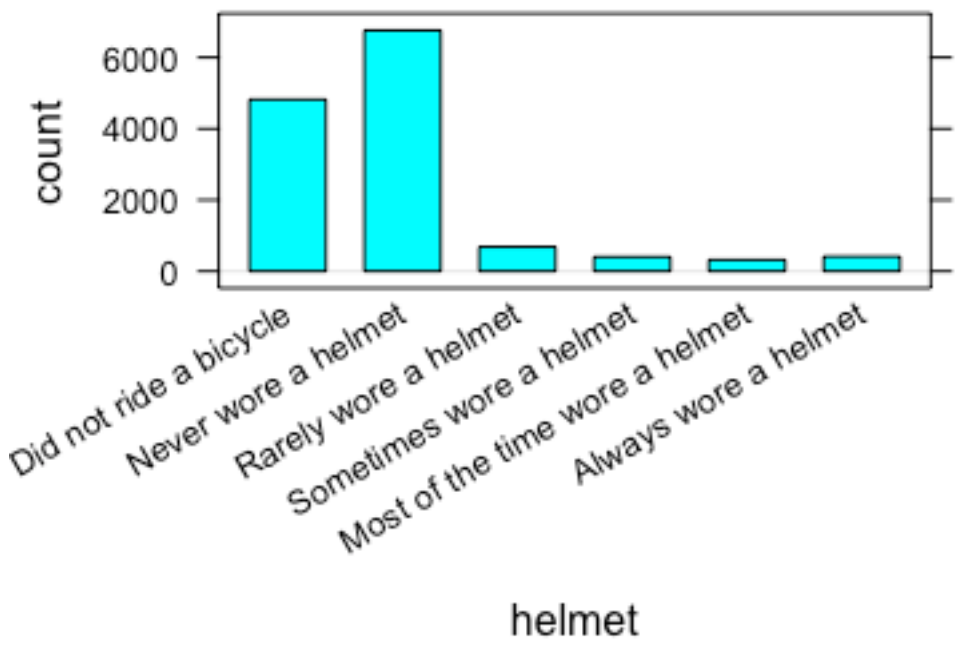
```
histogram(~weight, data = cdc)
```



The above code would be appropriate to visualize how weights are distributed. Based on the plot, we would say that a weight of around 75kg seems to be fairly typical.

- **Write the code to create a barplot to visualize the distribution of how often students wore a helmet while bike riding. About how many students never wore a helmet?**

```
bargraph(~helmet, data = cdc)
```



The above code could be used to create a bargraph of the `helmet` variable. It looks as though between 6500 and 7000 students reported never wearing a helmet.

Lab 1B - Get the Picture?

```
data(cdc)
```

- **Is height a numerical or categorical variable? Why?**
- **Is gender a numerical or categorical variable? Why?**
- **List either the different categories or what you think the measured units are for height and gender.**

The heights are measured in numerical units (meters) and so are numerical variables. Gender is measured qualitatively (Female, Male) and so is a categorical variable.

```
names(cdc)
```

```
##  [1] "age"               "gender"            "grade"
##  [4] "hisp_latino"       "race"              "height"
##  [7] "weight"            "helmet"            "seat_belt"
## [10] "drive_text"        "fights"            "bully_school"
## [13] "bully_electronic"  "depressed"         "days_smoking"
## [16] "days_vaping"       "sexuality"         "describe_weight"
## [19] "drink_juice"       "eat_fruit"         "eat_salad"
## [22] "drink_soda"        "drink_milk"        "eat_breakfast"
## [25] "days_exercise_60"  "hours_tv"          "hours_videogame"
```

```
## [28] "number_teams"      "asthma"            "hours_sleep"
## [31] "drink_sportsdrink" "drink_water"       "sunburns"
```

- **Write down 3 variables that you think are categorical variables and why.**
- **Write down 3 variables that you think are numerical variables and why.**

Based solely on the names of the variables, we'd imagine that it would make sense if `height`, `weight`, `age`, `grade`, etc. were numerical variables since we usually use numbers to describe these types of qualities.

Likewise, it'd make sense if `race`, `gender`, `asthma`, `seat_belt`, etc. were categorical since we normally associate each of these qualities as belonging to one group (category) or another.

```
str(cdc)
```

```
## 'data.frame':    15624 obs. of  33 variables:
##  $ age              : Factor w/ 7 levels "12 years old or younger",..
.: 5 5 6 6 5 7 7 6 6 6 ...
##  $ gender           : Factor w/ 2 levels "Female","Male": 2 1 2 2 1
2 2 2 1 1 ...
##  $ grade            : Factor w/ 5 levels "9th grade","10th grade",..
: 3 3 4 4 3 4 4 4 4 4 ...
##  $ hisp_latino      : Factor w/ 2 levels "Yes","No": 2 1 2 2 2 2 1 2
2 2 ...
##  $ race             : Factor w/ 8 levels "Am Indian / Alaska Native"
,..: 3 7 5 5 5 5 6 4 3 3 ...
##  $ height           : num  1.73 1.5 1.9 NA 1.63 1.7 1.73 1.75 1.5 1.
68 ...
##  $ weight           : num  54.4 51.3 66.7 NA 68.5 ...
##  $ helmet           : Factor w/ 6 levels "Did not ride a bicycle",..
: 2 2 1 2 1 2 2 NA 2 1 ...
##  $ seat_belt        : Factor w/ 5 levels "Never","Rarely",..: 5 5 5
5 4 5 5 5 4 5 ...
##  $ drive_text       : Factor w/ 8 levels "I did not drive the past 3
0 days",..: 1 1 1 4 1 1 8 NA 1 1 ...
##  $ fights           : Factor w/ 8 levels "0 times","1 time",..: 3 1
NA 2 1 1 1 1 1 1 ...
##  $ bully_school     : Factor w/ 2 levels "Yes","No": 1 2 1 2 1 1 2 2
2 2 ...
##  $ bully_electronic : Factor w/ 2 levels "Yes","No": 1 2 1 2 2 1 2 2
2 2 ...
##  $ depressed        : Factor w/ 2 levels "Yes","No": 2 2 1 1 1 1 1 2
1 2 ...
##  $ days_smoking     : Factor w/ 7 levels "0 days","1 or 2 days",..:
NA 1 NA 2 5 1 1 1 1 1 ...
##  $ days_vaping      : Factor w/ 7 levels "0 days","1 or 2 days",..:
3 1 NA 2 7 1 1 1 1 1 ...
##  $ sexuality        : Factor w/ 4 levels "Heterosexual (straight)",.
.: 1 1 2 1 1 1 1 1 1 1 ...
```

```
##  $ describe_weight  : Factor w/ 5 levels "Very underweight",..: 2 3
3 2 3 3 3 3 2 3 ...
##  $ drink_juice      : Factor w/ 7 levels "Did not drink fruit juice"
,..: 3 2 3 1 3 2 3 3 2 5 ...
##  $ eat_fruit        : Factor w/ 7 levels "Did not eat fruit",..: 4 4
4 2 3 2 7 2 2 6 ...
##  $ eat_salad        : Factor w/ 7 levels "Did not eat green salad",.
.: 1 2 1 1 1 2 2 2 2 2 ...
##  $ drink_soda       : Factor w/ 7 levels "Did not drink soda or pop"
,..: 3 3 4 4 2 7 3 1 2 5 ...
##  $ drink_milk       : Factor w/ 7 levels "Did not drink milk",..: 1
2 4 2 1 3 3 2 1 2 ...
##  $ eat_breakfast    : Factor w/ 8 levels "0 days","1 day",..: 4 8 6
3 1 2 8 8 3 8 ...
##  $ days_exercise_60 : Factor w/ 8 levels "0 days","1 day",..: 6 4 4
3 1 4 8 6 3 6 ...
##  $ hours_tv         : Factor w/ 7 levels "No TV on average school da
y",..: 5 NA 1 4 2 1 7 5 3 3 ...
##  $ hours_videogame  : Factor w/ 7 levels "No playing video/computer
game",..: 7 1 6 5 2 7 7 2 7 4 ...
##  $ number_teams     : Factor w/ 4 levels "0 teams","1 team",..: 1 1
1 2 1 2 2 3 1 2 ...
##  $ asthma           : Factor w/ 3 levels "Yes","No","Not sure": 2 2
NA 2 2 1 2 1 2 1 ...
##  $ hours_sleep      : Factor w/ 7 levels "4 or less hours",..: 4 5 5
3 1 2 3 4 2 1 ...
##  $ drink_sportsdrink: Factor w/ 7 levels "Did not drink sports drink
",..: 1 1 1 2 2 5 3 1 3 5 ...
##  $ drink_water      : Factor w/ 7 levels "Did not drink water",..: 4
5 7 4 7 2 6 7 2 5 ...
##  $ sunburns         : Factor w/ 6 levels "0 times","1 times",..: 1 1
3 1 2 3 1 2 1 1 ...
```

- **List all the types of info the str() function outputs**
- **Were you able to correctly guess which variables were categorical and numeric? Which ones did you mis-label?**

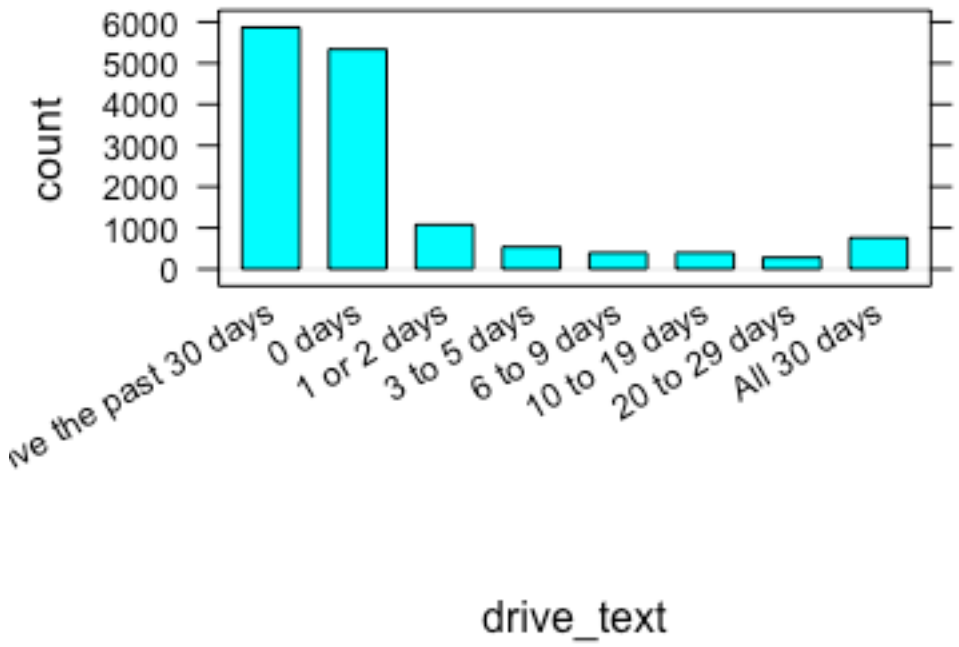
The `str` function lists the following types of information: - The type of object `cdc` is (data.frame) - The number of observations and variables - The variables in `cdc` along with what type of variable they are (Factor or Numerical), the number of categories for each categorical variable (levels), and the first few observations of each variable.

A few of the variables that we would normally think were numerical, such as `age` and `grade`, were actually categorical variables in this data.

- **Which function, either `bargraph` or `histogram` is better at visualizing categorical variables? Which is better at visualizing numerical variables?**

Bargraphs are better at visualizing categorical variables and `histograms` are better at visualizing numerical variables. The reason for this is because histograms have a continuous x-axis, just like the set of real numbers. Bargraphs have an x-axis that is broken up by categories and so are inappropriate for use with numerical variables.

```
bargraph(~drive_text, data = cdc)
```



- **What does the y-axis represent?**

The y-axis is the count, or frequency, each category occurred in our data.

- **What does the x-axis tell us?**

The x-axis is the different categories of the `drive_text` variable.

- **Would you say that most people never texted while driving? What does the word most mean?**
- **Approximately what percent of the people texted while driving for 20 or more days? (Hint: There's 15624 students in our data).**

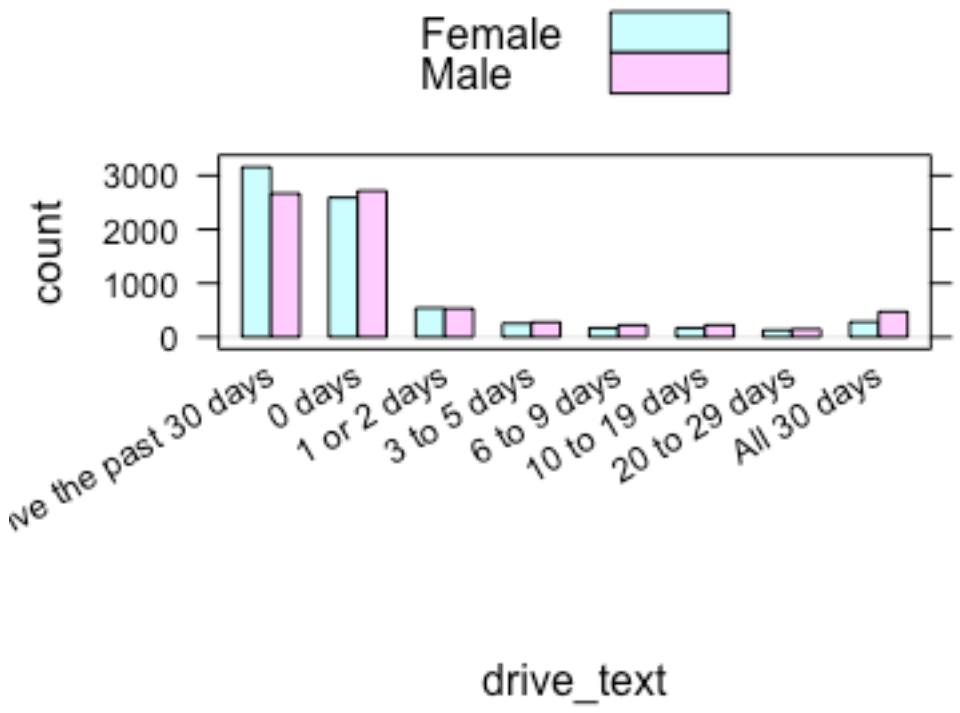
We would say that the most popular category of texting while driving was **"`0 days`"**. The reason for this is because around 5500 people texted 0 days while driving which is 35.2% of people (5500/15624). To calculate the percent of people who texted while driving, I'm going to subtract the people who did not drive in the past 30 days and those who texted 0 days while driving (15624 – 6000 – 5500 = 4124) to give us the remaining people who text and drove. This gives 26.4% of

people who texted and drove. Therefore, in this case we can say that most people never texted while driving.

The percent of people who texted while driving *for 20 or more days* seems to be around 1000 of the 15624 people or 6.4% (combine "20 to 29 days" and "All 30 days," which are around 200 and 800, respectfully).

***(Aside: At this point students are approximating and therefore do not need the exact values for each category. Had we used the exact values, we would need to account for the 967 people who did not answer this survey question.)*

```
bargraph(~drive_text, data = cdc, groups = gender)
```



- **Write a sentence explaining how boys and girls differ in their texting while driving.**
- **Would you say that most girls never text and drive? Would you say that most boys never text and drive?**
- **How did including the groups argument in your code change the graph?**

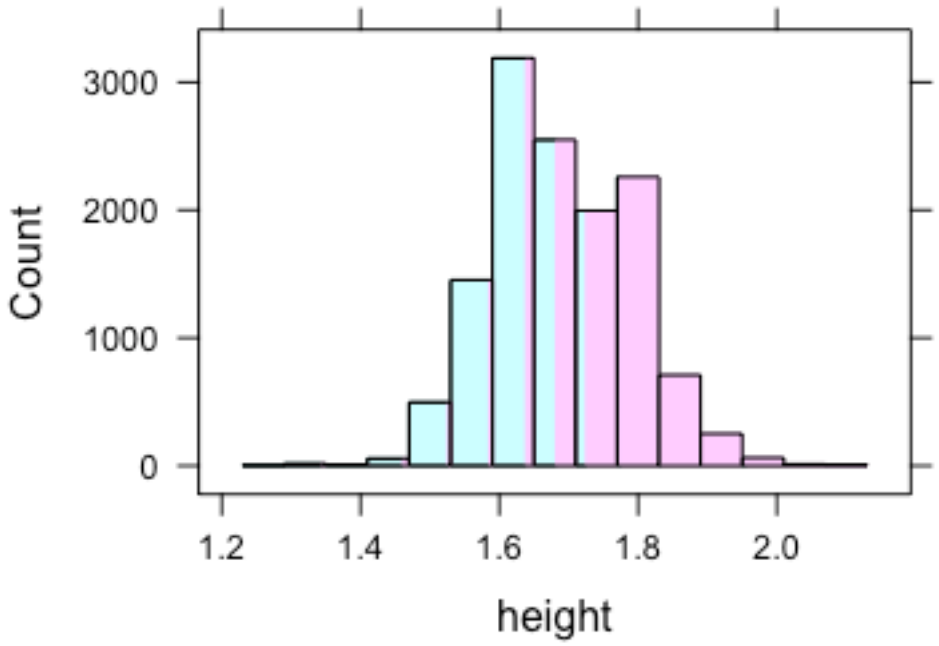
Females were more likely to report that they had not driven in the past 30 days and more males than females reported texting while at every other level, even not texting at all (0 days). Texting while driving 1 or 2 days appears equal between boys and girls.

If we only look at the drivers, we would claim that most females never text and drive since it appears that if we stacked the other bars on top of each other, the stacked

bars would be lower that the bar of females who never text and drive. For males this is more obvious.

The `groups` argument in the code split each bar based on the `gender` variable.

```
histogram(~height, data = cdc, groups = gender)
```



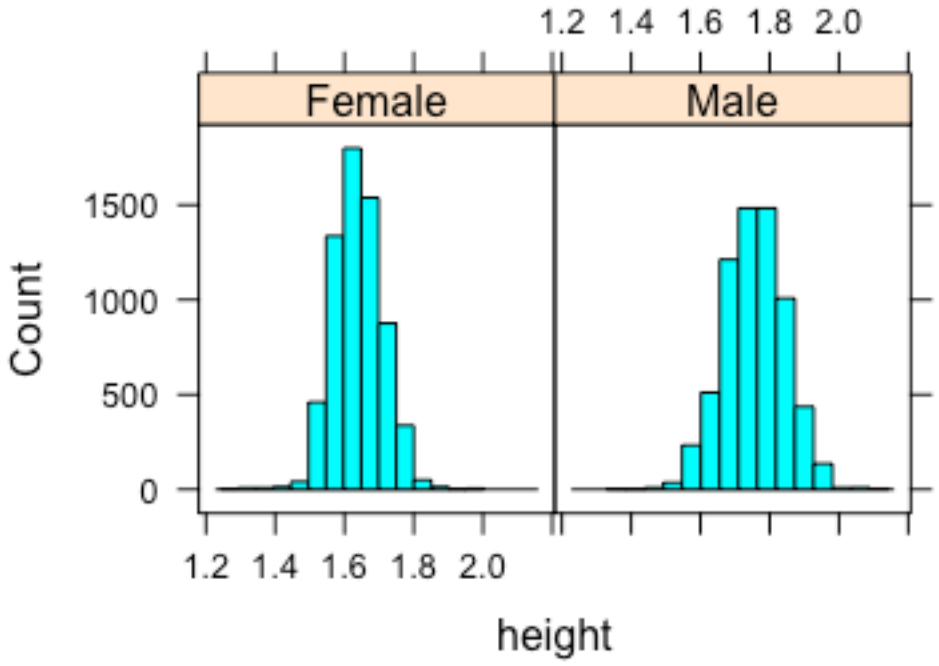
- **Can you use this graphic to answer the question at the top of the slide? Why or why not?**
- **Is grouping numeric values, such as heights, as helpful as grouping categorical variables, such as texting & driving?**

We cannot use this graphic to decide if males and females have similar heights. In this case, the graphic did not come out as helpful as it was when we used the `groups` option in the `bargraph` function.

- **Why does this work for bargraphs but not for for histograms?**

The `groups` argument attempts to split each group and place the bars next to each other on the x-axis. For histograms though, the x-axis is a continuous set of numbers. This means that each bar from the `groups` argument overlaps and as a result the plot comes out looking wonky.

```
histogram(~height | gender, data = cdc)
```



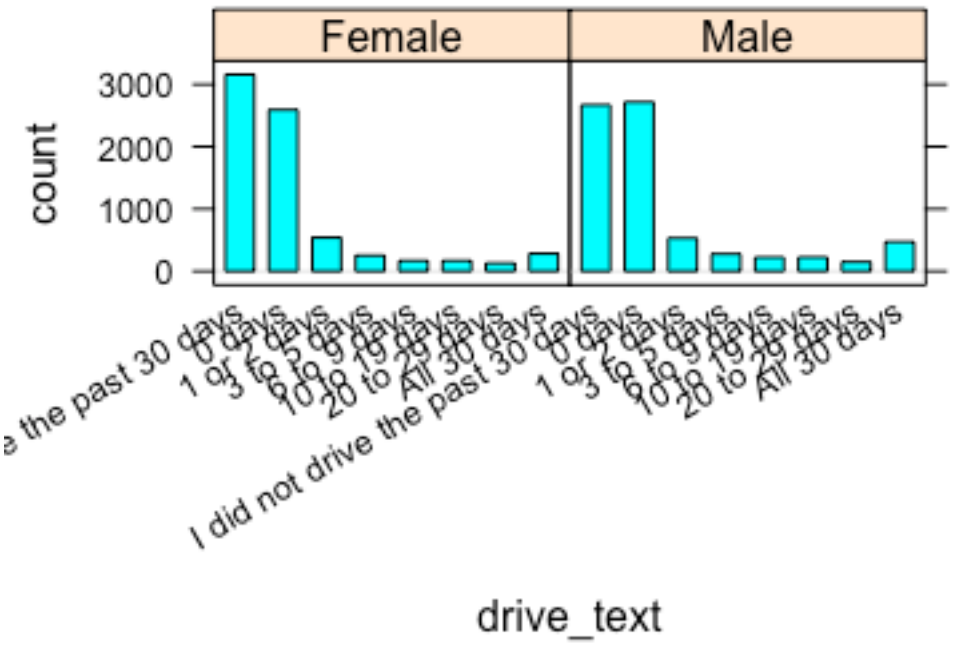
- **Do you think males & females have similar heights? Use the plot you create to justify your answer.**

It's hard to decide whether the heights are similar or not because it's hard to make horizontal comparisons. We guess if we were pressed for an answer, then sure, the distributions look pretty similar but their peaks occur at different numbers.

- **Just like we did for the histogram, is it possible to create a split bargraph? Try to create a bargraph of `drive_text` that's split by `gender` to find out.**

We can indeed create split bargraphs. In this case, instead of stacking the bars next to each other on the same x-axis, each bar is grouped with the other bars belonging to each category (Female, Male).

```
bargraph(~drive_text | gender, data = cdc)
```

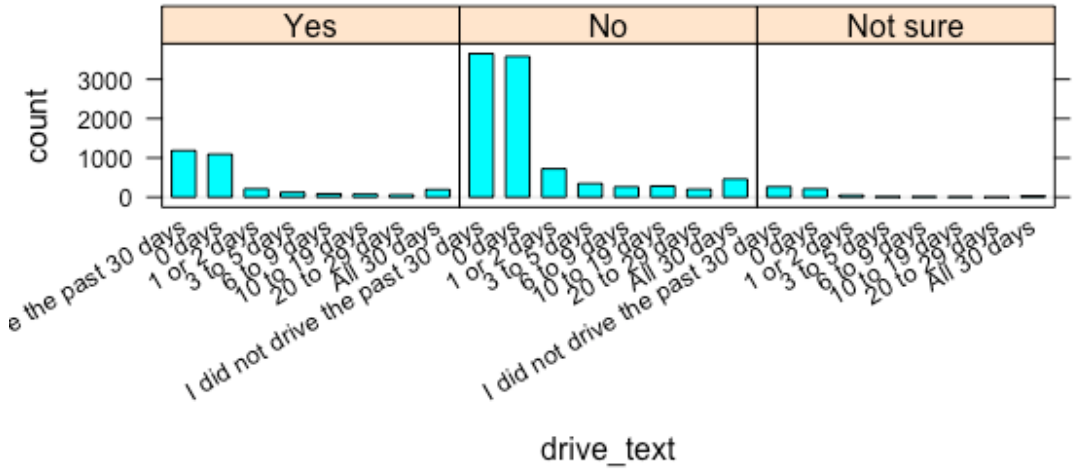


On your own:

- **What other factors do you think might affect how often people text and drive?**
- **Choose one variable from the `cdc` data, make a graph, and use the graph to describe how `drive_text` use differs with this variable.**

Factors that might affect people's texting while driving might be their age or who is messaging them. We'll investigate the effect of `asthma` on texting while driving.

```
bargraph(~drive_text | asthma, data = cdc)
```



Based on the plot, it appears that those who do not have asthma are more likely to not text and drive (about 3500 people versus 1000 with asthma who texted 0 days and 200 not sure if they have asthma who texted 0 days). However, it also appears that those without asthma had the highest number of non-drivers as well.

Lab 1C - Export, Upload, Import

(*James & Heidi: This lab is mostly about following the directions to export, upload and then import student's data. So we'll answer the very last question asked in the lab and supply a few things to watch out for.*)

- **View your data, select a variable and try to make an appropriate plot for that variable.**

When importing data, there's an option to name your data. If your students are having trouble making a plot, we imagine the most common reason for this is because they didn't give their data a name. Tell students to look at the *Environment* pane to see (1) if their data is listed and (2) what the name of their data is.

If the student's data isn't listed in the *Environment* pane then something went wrong and their data wasn't imported. To fix this, make sure their data was uploaded by asking them to find it in the *Files* pane and then have try importing again.

If the student's data is listed in the *Environment* pane, make sure whatever name they used when importing is the name they're using in the plot function, i.e. `data = _____` where the blank should be filled in with the exact name of their data.

Try to encourage students to not use spaces when importing their data.

Lab 1D - Zooming Through Data

(*James & Heidi: This lab uses the class participatory sensing data from the Food Habits campaign. As a result, many of the interpretations of the data will change depending on the data that is collected. we'll try our best to give general tips/solutions based on the code that should work, however;* ***we're going to assume that students named their data*** `food` ***when they imported the data****. If you find that the solutions aren't working,* ***make sure your data is named*** `food` ***OR replace the*** `data = ___` ***argument with the name you imported your data with****. For specific help or issues, send an email to support@mobilizingcs.org.*)

```
dotPlot(~sugar, data = food)
dotPlot(~sugar | salty_sweet, data = food)
```

- **Describe how R decides which observations go into the left or right plot.**
- **What does each dot in the plot represent?**

R will look at whether each observation is classified as `Salty` or `Sweet` and then create two plots, one for each category. Each dot in the plot then represents one survey response for each snack eaten.

```
dotPlot(~sugar | salty_sweet, data = food, layout = c(1, 2))
```

This line of code will change the `layout` to have `1` column and `2` rows of plots.

```
food_salty <- filter(food, salty_sweet == "Salty")
View(food_salty)
dotPlot(~sodium, data = food_salty)
```

- **View `food_salty` and write down the number of observations in it. Then use the subset data to make a dotPlot of the `sodium` in our `Salty` snacks.**

The number of observations can be found by using: `dim(food_salty)`, `nrow(food_salty)` or by looking in the *Environment* pane.

```
head(~salty_sweet == "Salty", data = food)
```

- **What do the values `TRUE` and `FALSE` tell us about how our rule applies to the first six snacks in our data? Which of the first six observations were `Salty`?**

If any of the first six observations were classified as being `Salty`, then those snacks will correspond to values of `TRUE`. Snacks classified as `Sweet` will correspond to values of `FALSE`.

- **About how much fat does the typical sweet snack have?**

```
food_sweet <- filter(food, salty_sweet == "Sweet")
dotPlot(~total_fat, data = food_sweet) # Option 1
dotPlot(~total_fat, data = food_sweet, nint = 10) # Option 2
```

Values for the typical amount of fat will vary, but students should be able to support their answer with well-reasoned explanations based on their `dotPlot`.

You'll notice in what we write as `# Option 2`, we've included the `nint` argument as an option. Changing the value of this option will change the number of bins used to create the `dotPlot`. Experiment with this option until you find a value that you think makes your `dotPlot` look nice.

- **How does the typical amount of fat compare when `healthy_level < 3` and when `healthy_level > 3`?**

```
food_healthy <- filter(food, healthy_level > 3)
food_unhealthy <- filter(food, healthy_level < 3)
dotPlot(~total_fat, data = food_healthy)
dotPlot(~total_fat, data = food_unhealthy)
```

Answers will vary but students should be able to write out a well-reasoned explanation about how they arrived at a *typical* value for each plot and then compare the two values.

Something interesting to look out for would be answers that find that both, what we refer to as healthy and unhealthy snacks, have similar typical values for fats or that healthy snacks have more/less fat.

An extension or talking point, depending on the answers, is how the class decides what snacks are healthy or unhealthy.

Lab 1E- What's the Relationship?

(*James & Heidi: This lab uses the class participatory sensing data from the Food Habits campaign. As a result, many of the interpretations of the data will change depending on the data that is collected. we'll try our best to give general tips/solutions based on the code that should work, however;* ***we're going to assume that students named their data*** `food` ***when they imported the data****. If you find that the solutions aren't working,* ***make sure your data is named*** `food` ***OR replace the*** `data = ___` ***argument with the name you imported your data with****. For specific help or issues, send an email to support@mobilizingcs.org.*)

- **How many variables were used to create this plot? Which variables were used and how were they used?**

Two variables were used to create the plots: `height` and `gender`. The code used to create them was

```
histogram(~height | gender, data = cdc)
```

In this code, we use `height` to create the histogram and `gender` to facet the plot.

- **Fill in the blanks to create a scatterplot with `sodium` on the y-axis and `sugar` on the x-axis.**

```
xyplot(sodium ~ sugar, data = food)
```

- **Do snacks that have more `calories` also have more `total_fat`? Why do you think that?**

To answer this question students should make the following plot.

```
xyplot(total_fat ~ calories, data = food)
```

Students should then look for a trend, or lack of a clear trend, in their data and justify why or why not the think a trend exists.

- **What happens if you swap the `calories` and `total_fat` variables in your code? Does the relationship between the variables change?**

Swapping the variables doesn't change the relationship between the variables. It only changes how we are visualizing them.

- **Does the relationship between `calories` and `total_fat` change when the snack is either `Salty` or `Sweet`? Write down the code you used to answer this question.**

```
xyplot(total_fat ~ calories | salty_sweet, data = food)
```

Starting with the student's belief about the existence of a relationship between `calories` and `total_fat`, students should next examine the individual faceted plots for `salty_sweet`. Does the relationship still exist, or not exist, in both salty and sweet snacks? Or does a new relationship appear? Answers will vary depending on the class' data.

- **Create a scatterplot that uses these 4 variables: sodium, sugar, healthy_level, salty_sweet.**

```
xyplot(sodium ~ sugar | salty_sweet, data = food, group = healthy_level
)
```

This is an example of a plot that would work. Students may also swap `sodium` and `sugar` or `salty_sweet` and `healthy_level`.

Something that would be considered *wrong* for this plot would be to swap the `sodium` or `sugar` with `healthy_level` from the code above. The `healthy_level` is one of the instances where a variable that appears to be numerical is actually a categorical. The reason for this is because the labels 1, 2, 3, 4, 5 aren't *measured* but instead are *assigned* a label.

- **How does the healthy_level of a Salty or Sweet snack impact the number of calories in the snack?**

```
dotPlot(~calories | healthy_level + salty_sweet, data = food)
```

The answers for this answer will vary, but the things to look for are where the typical value of `calories` lies in each of the faceted plots. Then, answering this

question boils down to finding out, under what levels of `healthy_level` and `salty_sweet`, the typical values appear to be systematically different.

On your own

- **Do healthier snacks cost more or less than less healthy snacks?**

```
histogram(~cost | healthy_level, data = food)
dotPlot(~cost | healthy_level, data = food)
```

This would be a statistically appropriate way to answer the question. Students should look to see how the typical values of `cost` changes as `healthy_level` increases from 1 to 5.

Using an `xyplot` would be inappropriate for this question because it's very likely that points will overlap one another and hide much of the variation in the data.

- **What other variables seem to be related to the `cost` of a snack? Describe their relationships.**

For this, students should make `xyplots` with `cost` on the y-axis and the other numerical variables on the x-axis. They should also make `histograms` or `dotPlots` of `cost` that are faceted by the categorical variables.

Based on the student's plots, they should then describe any relationships they find between `cost` and other variables.

Lab 1F - A Diamond in the Rough

```
data(atu_dirty)

View(atu_dirty)
```

- **Just by viewing the data, what parts of our ATU data do you think need cleaning?**

Some initial problems that students can see in the data:

- The variable names are not descriptive.
- The numbers in the data appear to be considered *characters*, that is, R treats them as words and not numbers.
- It's not obvious which level in the `gender` variable is for males and which is for females.

- **Use the example code and the variable information on the previous slide to rename the rest of the variables in atu_dirty**

```
atu_cleaner <- rename(atu_dirty, age = V1, gender = V2, employed = V3,
                      phys_diff = V4, sleep = V5, homework = V6,
                      social = V7)
```

Students might name their variables something else so if, later in the lab, students have problems one way to diagnose their issues would be to make sure they're using the variable names that they created in this step.

- **Write down the variables that should be numeric but are improperly coded as strings or characters.**

```
str(atu_cleaner)
```

```
## 'data.frame':    10493 obs. of  8 variables:
##  $ caseid   : Factor w/ 10493 levels "20160101160045",..: 1 2 3 4 5
6 7 8 9 10 ...
##  $ age      : chr  "62" "69" "24" "31" ...
##  $ gender   : Factor w/ 2 levels "01","02": 2 1 2 2 2 2 2 2 1 2 ...
##  $ employed : Factor w/ 3 levels "Full time","Part time",..: 3 3 3 2
2 3 3 3 3 1 ...
##  $ phys_diff: Factor w/ 2 levels "01","02": 1 2 1 1 1 1 1 1 1 1 ...
##  $ sleep    : chr  "690" "600" "940" "635" ...
##  $ homework : chr  "0" "0" "0" "0" ...
##  $ social   : chr  "465" "560" "20" "120" ...
```

Based on the `str` information (and what we named the variables for our solution), the variables that should be numeric are `age`, `sleep`, `homework` and `social`.

- **Look at the variables you thought should be numeric and select one. Then fill in the blanks below to see how we can correctly code it as a number:**
  - **Once you have this code working, use a similar line of code to correctly code the other numeric variables as numbers.**

```
atu_cleaner <- mutate(atu_cleaner, age = as.numeric(age),
                      sleep = as.numeric(sleep),
                      homework = as.numeric(homework),
                      social = as.numeric(social))
```

The above line fixes all of the numeric variables that were misclassified as strings.

- **To see the levels of gender type:**

```
tally(~gender, data = atu_cleaner)
```

- **Use similar code as we used above to write down the levels for the three factors in our data.**

```
tally(~gender, data = atu_cleaner)
```

```
## gender
##   01   02
## 4670 5823
```

```
tally(~employed, data = atu_cleaner)
```

```
## employed
## Full time Part time No answer
##      4979      1395      4119
```

```
tally(~phys_diff, data = atu_cleaner)

## phys_diff
##    01    02
##  9188  1305
```

- **Recode the categorical variable about whether the person surveyed had a phsical challenge or not.**

```
atu_cleaner <- mutate(atu_cleaner,
                      phys_diff = recode(phys_diff, "01"="No",  "02" =
"Yes"))
```

- **Create a script that:**
  - **(1) Loads the atu_dirty data set.**
  - **(2) Cleans the the data as we have in this lab.**
  - **(3) Saves a copy of the cleaned data (see next slide).**

Go to File -> New File -> R Script and write the script as follows:

```
# Load data
data(atu_dirty)

# Fix variable names
atu_cleaner <- rename(atu_dirty, age = V1, gender = V2, employed = V3,
                      phys_diff = V4, sleep = V5, homework = V6,
                      social = V7)

# Mutate numerical variables from stings back to numbers
atu_cleaner <- mutate(atu_cleaner, age = as.numeric(age),
                      sleep = as.numeric(sleep),
                      homework = as.numeric(homework),
                      social = as.numeric(social))

# Relabel categorical variable categories
atu_cleaner <- mutate(atu_cleaner,
                      gender = recode(gender, "01"="Male",  "02" = "Fem
ale"))
atu_cleaner <- mutate(atu_cleaner,
                      phys_diff = recode(phys_diff, "01"="No",  "02" =
"Yes"))

# Save a copy of the cleaned data
atu_clean <- atu_cleaner
save(atu_clean, file = "atu_clean.Rda")
```

Lab 1G - What's the FREQ

- **Use the data() function to load the atu_clean data file to use in this lab.**

```
data(atu_clean)
```

- **Fill in the blanks below to answer the following: How many more females than males are there in our ATU data??**

```
tally(~gender, data = atu_clean)
```

```
## gender
##   Male Female
##   4670   5823
```

There are 1153 more females than males in our data.

- **Use a line of code, that's similar to how we facet plots, to tally the number of people with phyiscal challanges by their genders.**

Beware: This question asks us to tally the people with physical challenges and *split them by gender*. The following code would be appropriate:

```
tally(~phys_challenge | gender, data = atu_clean)
```

```
##                  gender
## phys_challenge    Male Female
##   No difficulty   4140   5048
##   Has difficutlty  530    775
```

- **Does one gender seem to have a higher occurence of physical challenges than the other? If so, which one and explain your reasoning?**

At this point, it appears as though females have more physical challenges than males. It should be noted however that there are more females in the data set so the raw counts are misleading.

- **Include: format = "percent" as option to the code you used to make your 2-way frequency table. Then answer this question again:**

```
tally(~phys_challenge | gender, data = atu_clean, format= "percent", ma
rgins = TRUE)
```

```
##                 gender
## phys_challenge        Male    Female
##   No difficulty   88.65096  86.69071
##   Has difficulty  11.34904  13.30929
##   Total          100.00000 100.00000
```

- **Does one gender seem to have a higher occurence of physical challenges than the other? If so, which one and explain your reasoning?**
- **Did you answer change from before? Why?**

It now appears that, even when taking into account the difference in the numbers of males and females, that females tend to have more physical challenges in our data set.

Also note, we demonstrate the use of the `margins` option in the code but this step is optional for the students.

On your own

- **Describe what happens if you create a 2-way frequency table with a numerical variable and a categorical variable.**

To demonstrate this, we could run the following code:

```
tally(~gender | age, data = atu_clean)
```

What occurs is that every age is tallied, resulting in a table that is far too complex to be useful.

- **How are they types of statistical questions that 2-way frequency tables can answer different than 1-way frequency tables?**

For 1-way frequency tables, we can really only answer the very simplest of statistical questions. With 2-way frequency tables, we're able to compare the relationships between categorical variables which gives us a very good way to answer more complex statistical questions.

- **Which gender has a higher rate of part time employment**

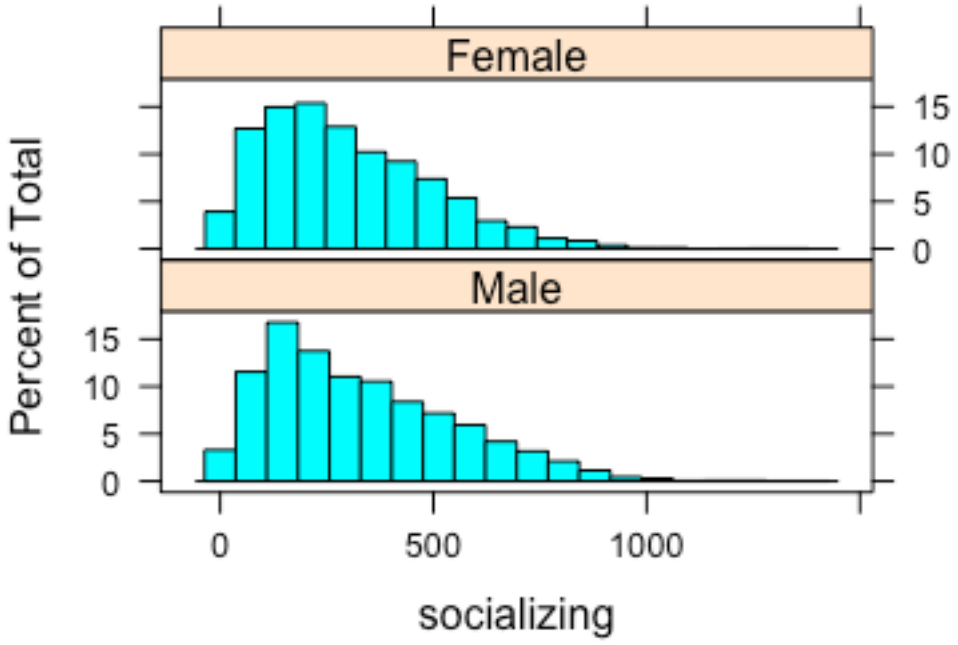
```
tally(~fulltime_emp | gender, data = atu_clean, format = "percent")
```

```
##              gender
## fulltime_emp      Male    Female
##    Full time 58.286938 38.760089
##    Part time  9.293362 16.503521
##    No answer 32.419700 44.736390
```

Females have a higher rate of part time employment.

- **Does one gender socialize more than the other? To answer this question first:**
- **Create a subset of the ATU data that includes only people who socialized more than 0 minutes.**
- **Include type = "percent" as an option in the histogram function.**

```
atu_social <- filter(atu_clean, socializing > 0)
histogram(~socializing | gender, data = atu_social, type = "percent",
          layout = c(1,2), nint = 20)
```



It's not obvious if one gender socializes more than the other. It appears as though time spent `socializing` isn't very different between the genders.

(*James & Heidi: Something you might notice is just how many options we've included in our histogram, each of which was introduced in previous labs. Encourage students, when making graphics, to try including options to see if what they make the plot easier to interpret*)

Lab 1H - Our Time

(*James & Heidi: This lab uses the class participatory sensing data from the Time Use campaign. As a result, many of the interpretations of the data will change depending on the data that is collected. we'll try our best to give general tips/solutions based on the code that should work, however;* ***we're going to assume that students named their data*** `timeuse_raw` ***when they imported the data****. If you find that the solutions aren't working,* ***make sure you imported your data and named it*** `timeuse_raw` ***OR replace the*** `data = ___` ***argument with the name you imported your data with****. For specific help or issues, send an email to support@mobilizingcs.org.*)

```
timeuse <- timeuse_format(timeuse_raw)
```

- **How many observations and variables are there?**
- **What are the names of the variables?**
- **Which row represents YOUR typical day?**

These questions can be answered by running the following:

```
dim(timeuse) # Number of observations and variables
names(timeuse) # Name of variables
```

The number of observations will vary by class but there are 16 variables. The names of the variables are `user.id`, `submissions`, `chores`, `friends`, `grooming`, `homework`, `meals`, `online`, `read`, `school`, `sleep`, `sports`, `television`, `travel`, `videogames`, and `work`. The row that represents YOUR typical day will vary by student.

To find an individual student's row, have them `View(timeuse)` and locate their row based on their unique `user.id`.

- **State and answer two statistical questions based on our research question.**
- **Also, state one way in which your personal data is typical and one way that it differs from the rest of the class.**
- **Justify your answers by using appropriate statistical graphics and summary tables.**
- **If you subset your data, explain why and how it benefited your analysis.**

The answers for this will vary by student. When evaluating the solutions to your student's analysis, look for the following:

1. Do the statistical questions address the research question? Are the questions interesting, appropriate and can they be answered with the data the students are given.
2. Are the values the students claim are *typical* explained and well-reasoned.
3. Are the plots created appropriate? And are the claims that are based on these plots appropriate? Are there other methods that the students could have used that would have been better?

(*James & Heidi: This guide to evaluating your students is definitely not definitive, so teachers will have to use their best judgement to adapt this guide to each student's analysis. We would encourage teachers to share interesting analyses with the other teachers in their home. The home groups would also make an ideal forum to ask questions as to the appropriateness of the analyses.*)