Graphs, Tables, and Statistics

Sean Raleigh

Now that you know your way around R a little bit, let's start learning how to summarize data using graphs, tables, and summary statistics.

Instructions

Remember that this assignment (and all future assignments) will require you to modify this R Markdown file. You may want to knit to HTML while you're actively working on the assignment. You'll need to knit to PDF and export the PDF to your computer in order to submit the final draft in PDF form to Canvas.

Sometimes you will be asked to add your own R code. That will appear in this document as a code chunk with a request for you to add your own code, like so:

```
## Add code here to [do some task]...
```

When you see that in a code chunk, you need to type some R code to complete a task.

Sometimes you will be asked to type up your thoughts. Instructions to do that will be labeled as follows. If you are currently reading this in the knit output, please look back at the R Markdown file to see the following text:

(When you knit the document, you can't see the text from the line above. That's because the crazy notation surrounding that text is an HTML comment, and therefore doesn't appear in the output.) In these areas of the assignment, please use full sentences and proper spelling, grammar, punctuation, etc. This is not R code, but rather a free response section where you talk about your analysis and conclusions.

Getting started

Make sure you're in a project

If you're looking at this document, you should have already created a project and uploaded this R Markdown file to that project folder.

Save your file!

The first thing we **always** do is save our file. You'll probably want to save this under a new name. Go to the "File" menu and then "Save As". Once you've saved the file with the new name, from them on it's easier to just hit Ctrl-S (or Cmd-S on a Mac) to keep saving it periodically.

Remember that file names should not have any spaces in them. (In fact, you should avoid other kinds of special characters as well, like periods, commas, number signs, etc. Stick to letters and numerals and you should be just fine.) If you want a multiword file name, I recommend using underscores like this: `this_filename_has_spaces_in_it`.

Load Packages

We load the standard `mosaic` package as well as the `MASS` package so that we can continue to work with the birth weight data. At the same time, we'll also load the `gmodels` package to make nice contingency tables.

```
library(mosaic)
library(MASS)
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 3.2.2
```

Numerical data

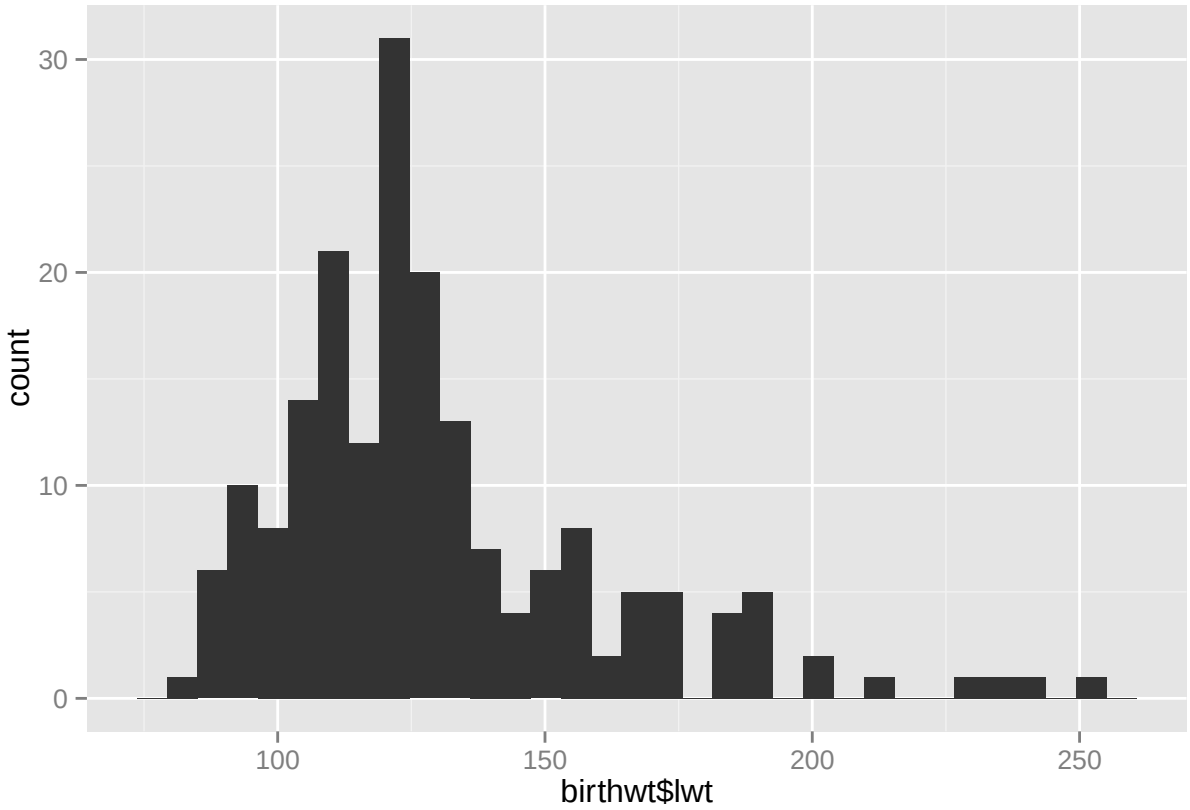
Graphing one numerical variable

To make plots in R, we will almost always use the `qplot` command. (This command is part of the `ggplot2` package which is one of the packages loaded when you load the `mosaic` package.)

The most useful display of a single numerical variable is a histogram. If you use the `qplot` command with a single numerical variable, the default graph type is already a histogram.

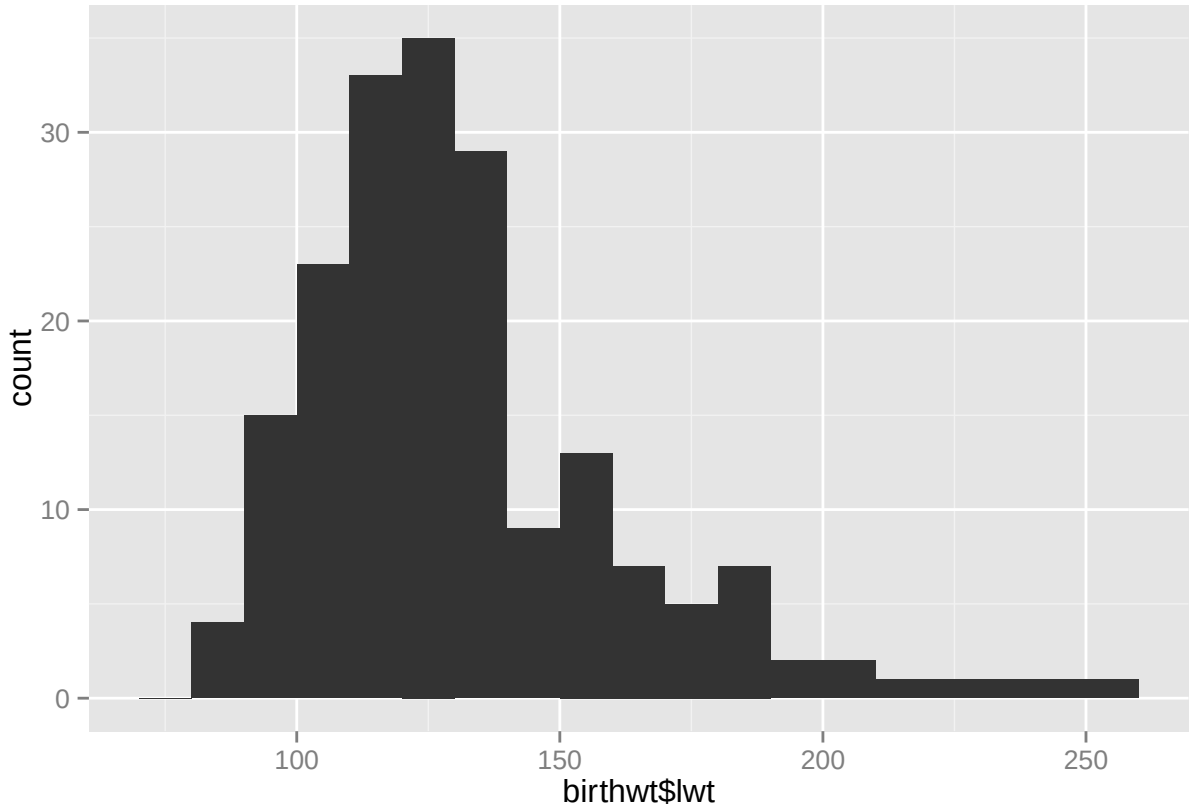
```
qplot(birthwt$lwt)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Question: What does the variable `lwt` represent? Generally, the default binning for `qplot` histograms is not so great. Let's try bins that are a little wider and that line up with numbers that are easy to see in the plot.

```
qplot(birthwt$lwt, binwidth = 10)
```



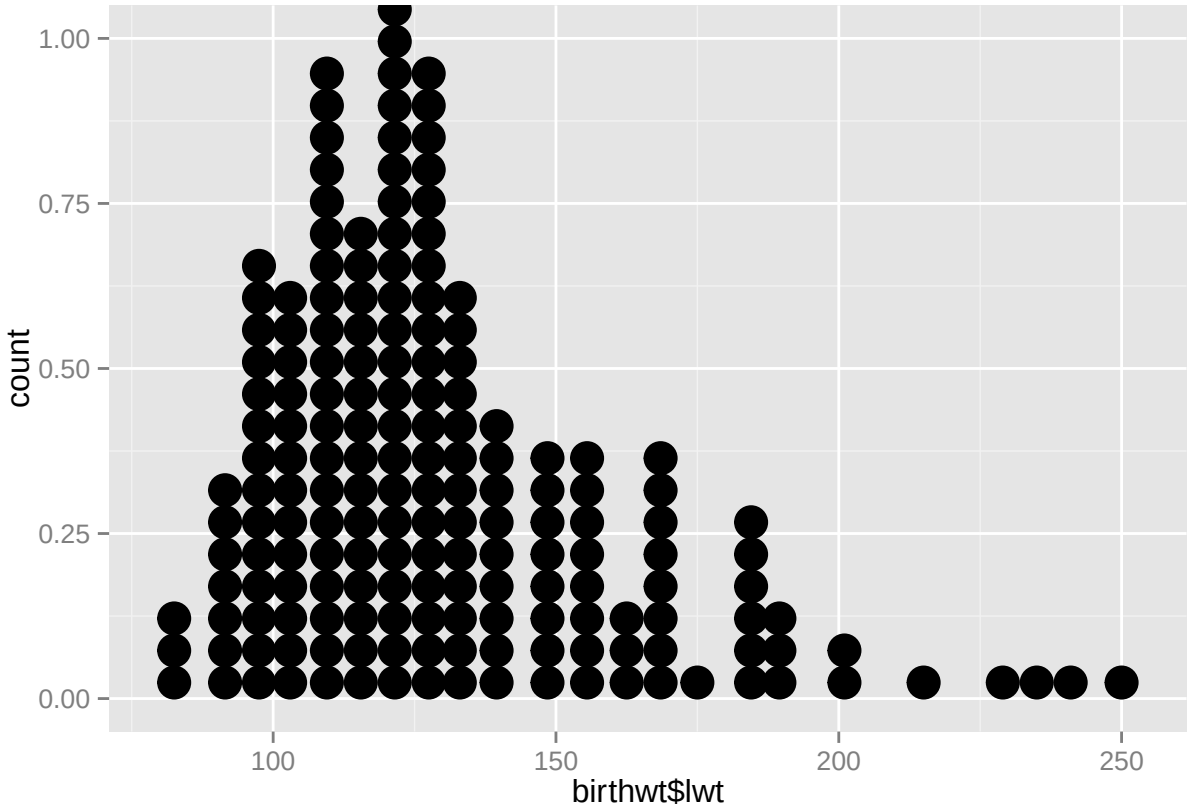
Write a paragraph or so describing the shape of the distribution of the `lwt` variable, focusing on the three key features we discussed in class (modes, symmetry, and outliers). Be sure to speak about these in the context of the data; in other words, your answer should refer to women and their weight, and not just abstract numbers and stats words.

```
## Add code here to create a histogram for the distribution of
## the baby's birth weight (in grams).
## Don't forget to adjust the bins if necessary.
```

Your turn! Create a histogram of the baby's birth weight (in grams). Then describe the shape of the distribution as you did for the `lwt` variable above. There are two other graph types that one might see for a single numerical variable: dotplots and boxplots. I'm not a big fan of dotplots. For the sake of completeness, here is the R code for making a dotplot:

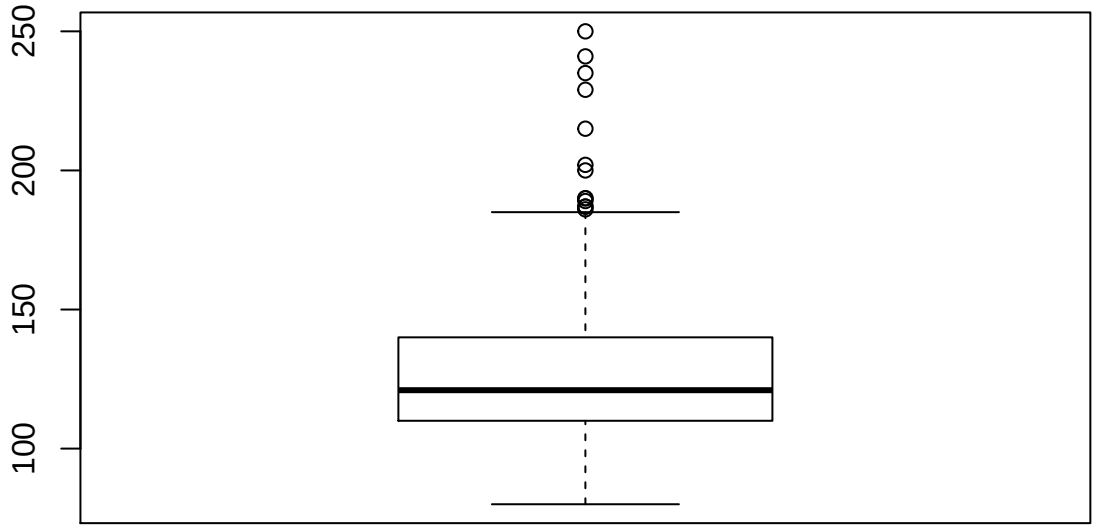
```
qplot(birthwt$lwt, geom = "dotplot")
```

```
## stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



I do like boxplots, but they are typically less informative than histograms. Boxplots are much better for comparing groups, so we'll see them again later. Besides, there isn't an easy way to make a boxplot for a single numerical variable using `qplot`. Here is an ugly alternative using base R graphics:

```
boxplot(birthwt$lwt)
```



Compare the histogram of the `lwt` variable to the boxplot generated above. Do you think all those dots at the top of the boxplot are outliers? Explain what's really going on.

Graphing two numerical variables

The proper graph for two numerical variables is a scatterplot. Again, `qplot` will default to a scatterplot if you include two numerical variables.

If you were interested in exploring a possible association between the weight of the mother at her last menstrual period and the birth weight of the baby, which variable would you consider to be the explanatory variable and which would be the response variable? Explain your reasoning. (Be careful not to use language that suggests cause or effect. There may, in fact, be a causal relationship, but that is never going to be proven from observational data.)

```
## Add code here to create a scatterplot of the
## weight of the mother at her last menstrual
## period and the birth weight of the baby.
```

Now create a scatterplot of the weight of the mother at her last menstrual period and the birth weight of the baby. (Your x-axis and y-axis should agree with the choice of explanatory and response variables from the last question.)

Comment on the nature of the association. (Is is positive/negative, or are these two variables independent?) As always, be sure to word your answer in the context of the data.

Summary statistics

In a previous assignment, you have already seen how to summarize a numerical variable. The `summary` commands gives you the five-number summary, and throws the mean in for good measure.

```
summary(birthwt$lwt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   110.0   121.0   129.8   140.0   250.0
```

If you want to isolate the various pieces of this, you can:

```
min(birthwt$lwt)
```

```
## [1] 80
```

```
max(birthwt$lwt)
```

```
## [1] 250
```

```
median(birthwt$lwt)
```

```
## [1] 121
```

```
mean(birthwt$lwt)
```

```
## [1] 129.8148
```

```
quantile(birthwt$lwt)
```

```
##   0%  25%  50%  75% 100%
##   80  110  121  140  250
```

You can also calculate the IQR.

```
IQR(birthwt$lwt)
```

```
## [1] 30
```

The `mosaic` package also has a summary command called `favstats` that has a little more information, including the standard deviation, the sample size, and a count of any cases that are missing data.

```
favstats(birthwt$lwt)
```

```
##  min  Q1 median  Q3 max     mean       sd   n missing
##   80 110    121 140 250 129.8148 30.57938 189       0
```

Again, you can isolate these:

```r
sd(birthwt$lwt)
```

```
## [1] 30.57938
```

```r
nrow(birthwt)
```

```
## [1] 189
```

Also, don't forget about the trick for using R commands inline. If you need to mention a statistic in the middle of a sentence, there is no need to break the sentence and display a code chunk. Be sure you're looking at the R Markdown document to note that the numbers in the next sentence are not manually entered, but are calculated on the fly:

There are 189 births represented in this data and the median weight of the women as of their last menstrual period is 121 pounds.

Your turn! Type a full sentence using inline R code (as above) summarizing the minimum and maximum baby weights (in grams) in our data set.

Categorical data

Working with factor variables

R uses the term "factor variable" to refer to a categorical variable. Your data set may already come with its variables coded correctly as factor variables, but often they are not. For example, our birth weight data has several categorical variables, but they are all coded numerically.

The code here is somewhat involved and technical. After the code chunk, I'll explain what each piece does.

```r
race <- factor(birthwt$race, levels = c(1, 2, 3), labels = c("White", "Black", "Other"))
```

First of all, because `birthwt` is a dataset defined in the `MASS` package, we cannot modify it. Therefore, if we want to change something, we have to assign a new name to the resulting operation. That is why we have `race <-` at the beginning of the code line. The symbol `<-` is taking the result of the command on the right (in this case, the `factor` command) and giving it a new name. From now on, when we want to analyze `birthwt$race`, we will just type `race` instead.

The `factor` command is converting `birthwt$race` into a factor variable. The `levels` of the variable are the pre-existing numerical values. The `labels` are the names we actually want to appear in our output.

The only weird thing left is the presence of the letter `c` in `c(1, 2, 3)` and `c("White", "Black", "Other")`. This letter `c` is necessary whenever we want to combine more than one thing into a single expression. (In technical terms, the "c" stands for "combine" or "concatenate" and creates a "vector". Don't worry too much about it now.)

Whenever you need to create a factor variable, I recommend that you just copy and paste the syntax from the above chunk and make the necessary changes.

Graphing one categorical variable

When asked, "What type of graph should I use when graphing a single categorical variable?" the simple answer is "None." If you do need to summarize a categorical variable, a frequency table usually suffices.

```
table(race)
```

```
## race
## White Black Other
##    96    26    67
```

If you want percentages, put the above command inside the `prop.table` command:

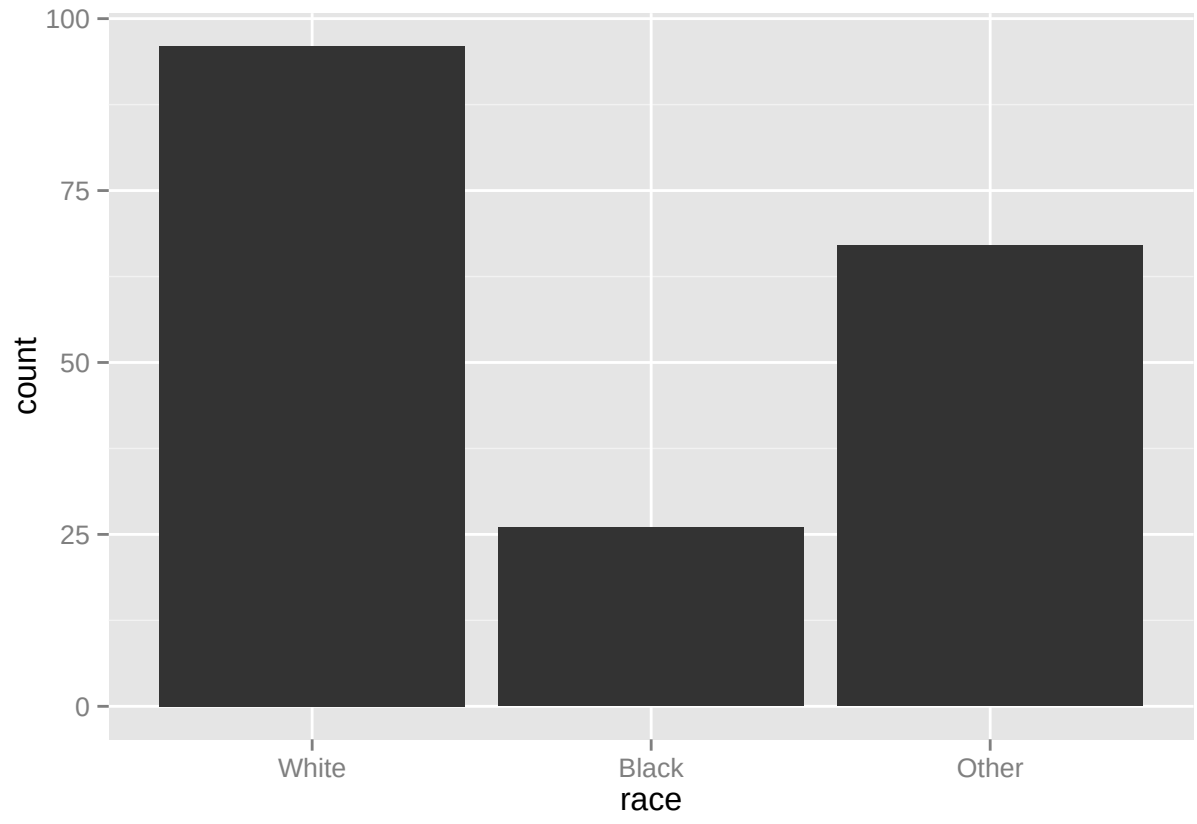
```
prop.table(table(race))
```

```
## race
##     White     Black     Other
## 0.5079365 0.1375661 0.3544974
```

```
## Add code here to generate a frequency table like above,
## but this time use the original variable birthwt$race.
## You should be able to see the advantage of creating a
## factor variable with meaningful labels.
```

If you really want a bar chart, `qplot` will give you one by default when you use a factor variable.

```
qplot(race)
```



```
## Add code here to generate a bar chart like above,
## but this time use the original variable birthwt$race.
## You should be able to see the advantage of creating a
## factor variable with meaningful labels.
```

What about pie charts? Just. Don't.

Seriously. Pie charts suck.

Graphing two categorical variables

Two categorical variables can be shown in table form or chart form.

The table form is called a contingency table (or pivot chart, or cross-tabulation, or probably several other terms as well). There are multiple ways of getting contingency tables out of R, but the most flexible is the `CrossTable` command from the `gmodels` package.

First things first, though. We need to create one more factor variable. We'll use the `smoke` variable about whether the mothers smoked during pregnancy.

```
smoke <- factor(birthwt$smoke, levels = c(0, 1), labels = c("No", "Yes"))
```

And now for the contingency table. The first variable will be your row variable and the second, your column variable. There's no right or wrong way to do this, but I prefer to put my explanatory variable as the row and the response variable as the column. I might be interested in knowing if a woman's race is associated with how likely she might have been to smoke. Therefore, `race` will be my row variable and `smoke` will be my column variable. For now, ignore all the extra options on the second line of the code chunk below.

```
CrossTable(race, smoke,
           prop.r = FALSE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  189
##
##
##              | smoke
##         race |        No |       Yes | Row Total |
## -------------|-----------|-----------|-----------|
##        White |        44 |        52 |        96 |
## -------------|-----------|-----------|-----------|
##        Black |        16 |        10 |        26 |
## -------------|-----------|-----------|-----------|
##        Other |        55 |        12 |        67 |
## -------------|-----------|-----------|-----------|
## Column Total |       115 |        74 |       189 |
## -------------|-----------|-----------|-----------|
##
##
```

As we talked about in class, this is highly misleading. For example, one cannot compare the 10 black women who smoked to the 12 "other" women who smoked. The 10 are out of 26, but the 12 are out of 67. That's why we need percentages. As the explanatory variable is in the rows, we turn on row percentages using the `prop.r` option of `CrossTable`.

```
CrossTable(race, smoke,
           prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  189
##
##
##              | smoke
##         race |        No |       Yes | Row Total |
## -------------|-----------|-----------|-----------|
##        White |        44 |        52 |        96 |
##              |     0.458 |     0.542 |     0.508 |
## -------------|-----------|-----------|-----------|
##        Black |        16 |        10 |        26 |
##              |     0.615 |     0.385 |     0.138 |
## -------------|-----------|-----------|-----------|
##        Other |        55 |        12 |        67 |
##              |     0.821 |     0.179 |     0.354 |
## -------------|-----------|-----------|-----------|
## Column Total |       115 |        74 |       189 |
## -------------|-----------|-----------|-----------|
##
##
```

Question: What percentage of black women smoked during pregnancy? What percentage of "other" women smoked during pregnancy?

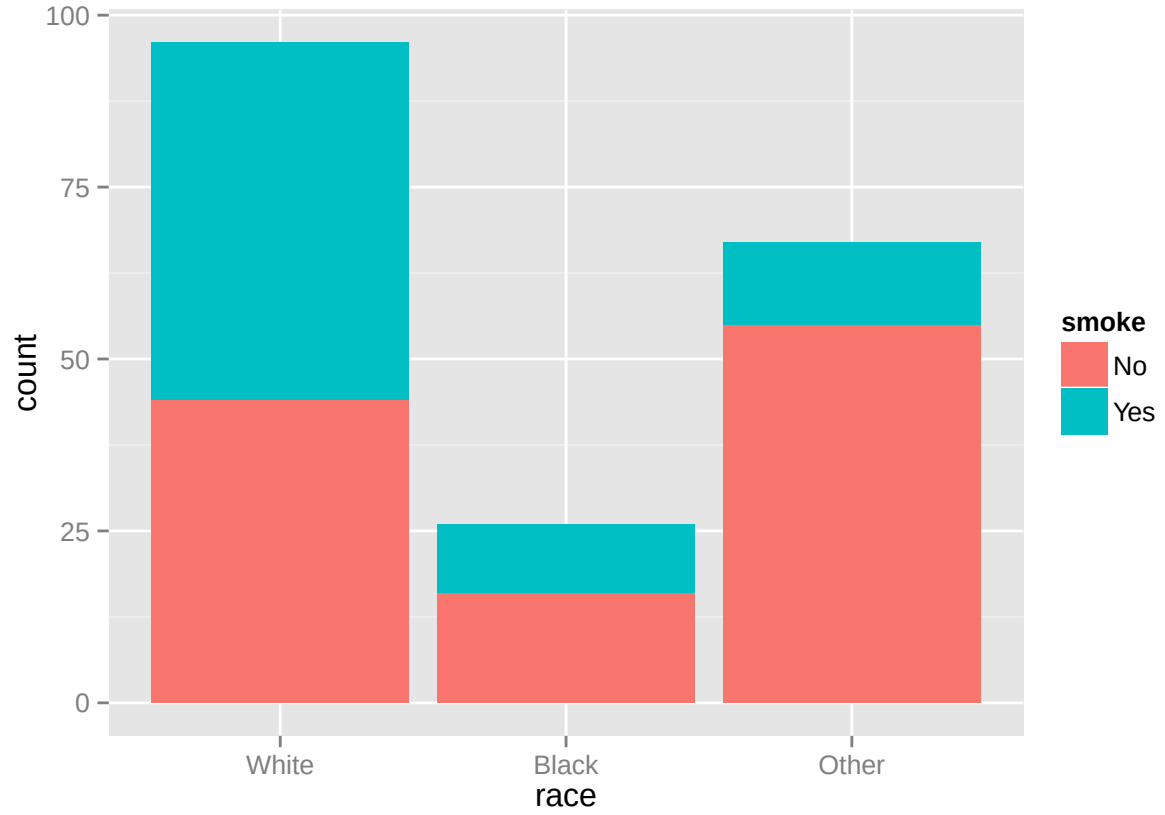
Question: Does race appear to be associated with the likelihood of smoking during pregancy for the women in this data set? Or are these variables independent?

```
## Add code here to convert one or more variables to factor variables.
```

```
## Add code here to create a contingency table with row percentages.
```

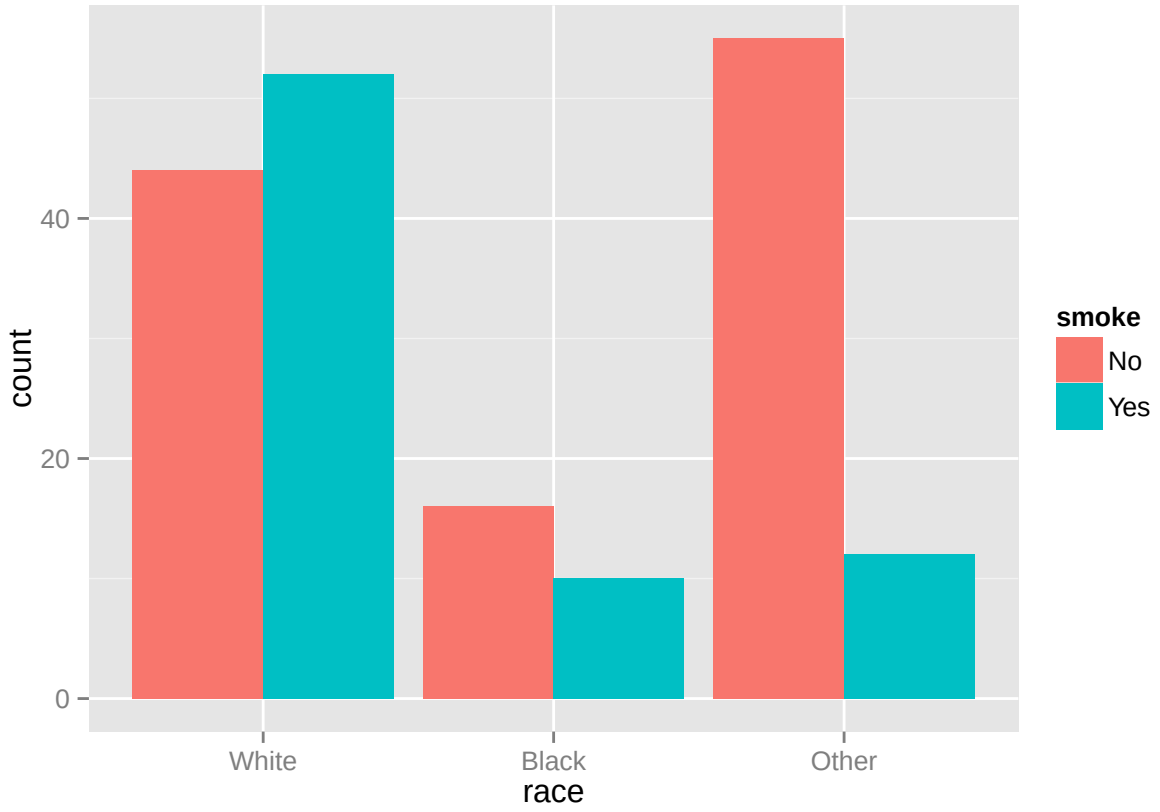
Your turn. Choose two categorical variables of interest from the `birthwt` data set. (Choose at least one variable other than `race` or `smoke`.) Turn them into factor variables with meaningful levels. Identify one as explanatory and one as response. Then create a contingency table with row percentages. Comment on the association (or independence) of the two variables. Now let's look at the graphical analogue of the contingency table, the stacked bar chart:

```
qplot(race, fill = smoke)
```



Or, with a minor change in syntax, one can create a side-by-side bar chart (which I prefer):

```
qplot(race, fill = smoke, position = "dodge")
```



A better option here would be to use relative frequencies (i.e., percentages within each group) instead of counts on the y-axis for the same reason that row percentages were better in a contingency table. Unfortunately, it is nearly impossible to do this with `qplot`, so we will say no more about the subject.

```
## Add code here to make your own side-by-side bar chart
## for the combination of variables for which you created
## the contingency table earlier.
```

Your turn! Make a side-by-side bar chart for the combination of variables for which you created the contingency table earlier. Be sure that your explanatory variable ends up on the x-axis and your response variable is represented as the different colors within each cluster of bars. Discuss the resulting graph. Comment on the association (or independence) of the two variables.

Numerical data by groups

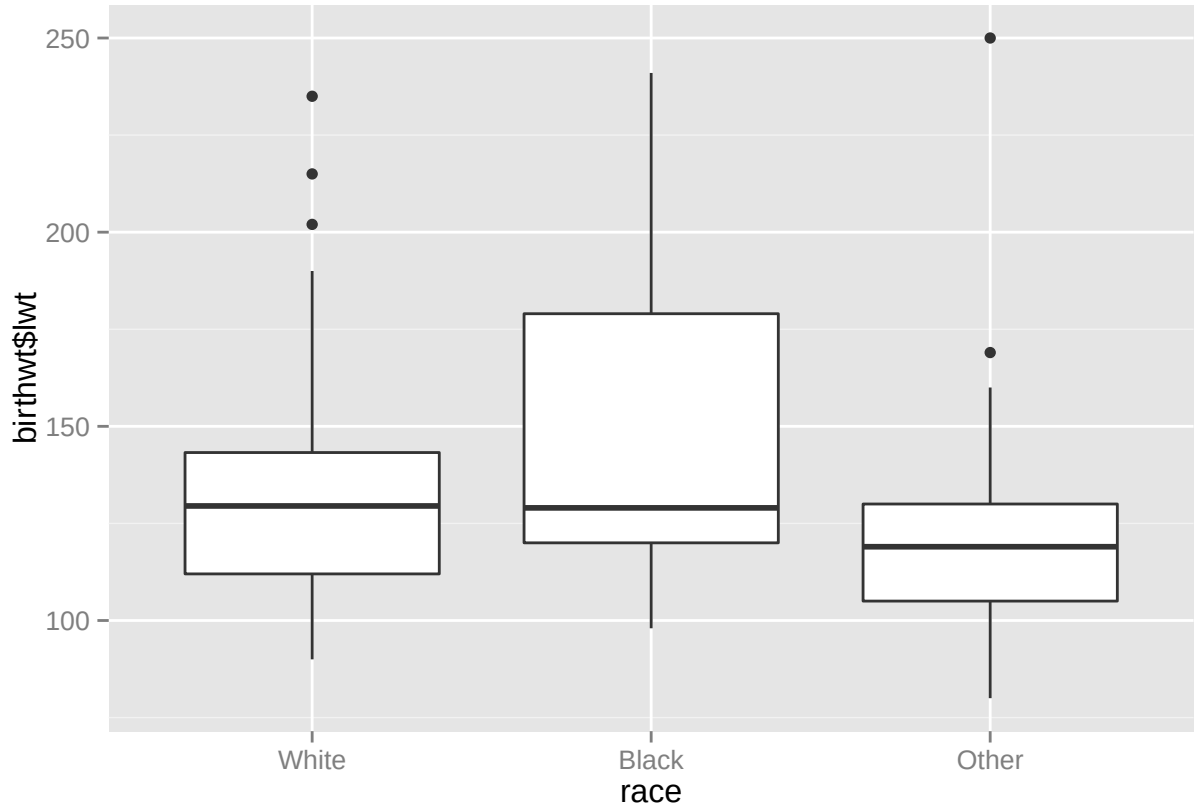
Graphing numerical data by groups

We have already learned how to analyze one or two numerical variables and one or two categorical variables. The only combination we have not considered yet is when you have one numerical variable and one categorical variable. The idea here is that the categorical variable divides up the data into groups and you are interested in understanding the numerical variable for each group separately. Another way to say this is that your

categorical variable is explanatory and your numerical variable is response. For an example, let's consider the mother's weight by race.

Graphically, there are two good options here. The first is a side-by-side boxplot.

```
qplot(race, birthwt$lwt, geom = "boxplot")
```

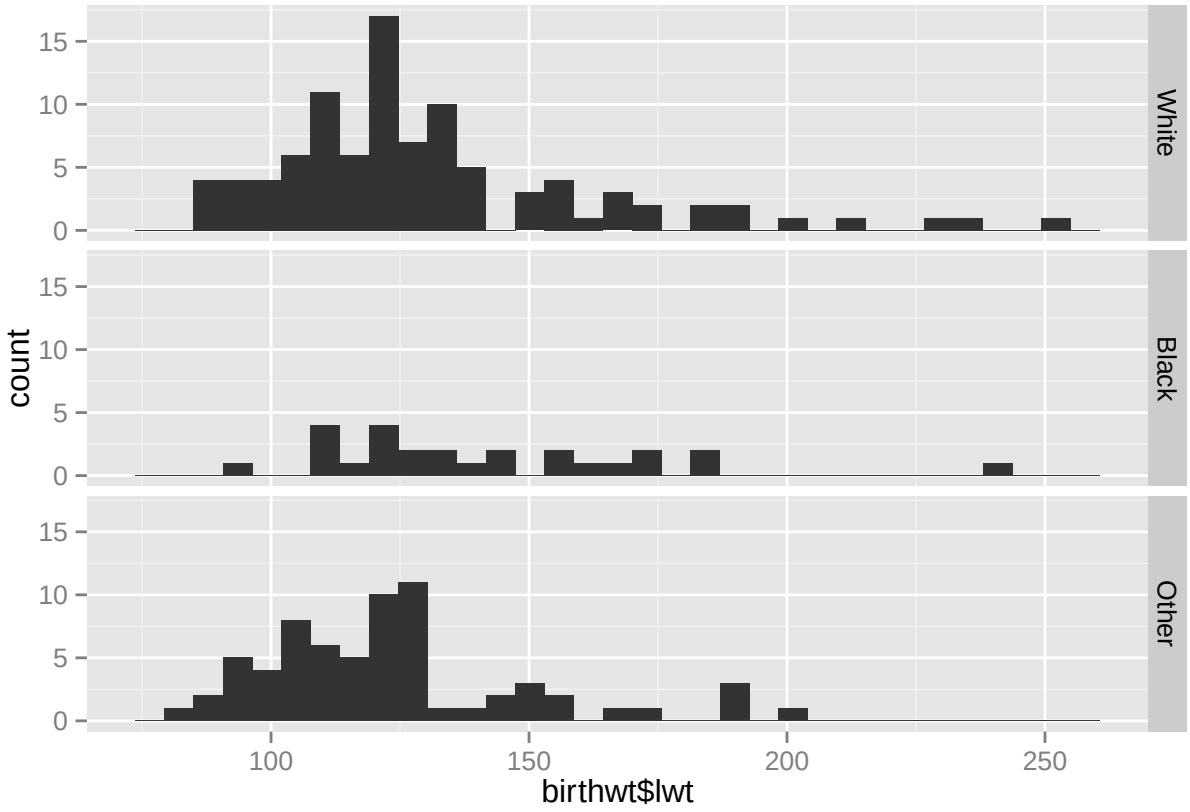


Notice the order of the variables. The x-axis variable (in this case, `race`) goes first, followed by the y-axis variable.

The other possible graph is a stacked histogram. This uses a feature called "faceting" that creates a different plot for each group. The syntax is a little unusual.

```
qplot(birthwt$lwt, facets = race ~ .)
```

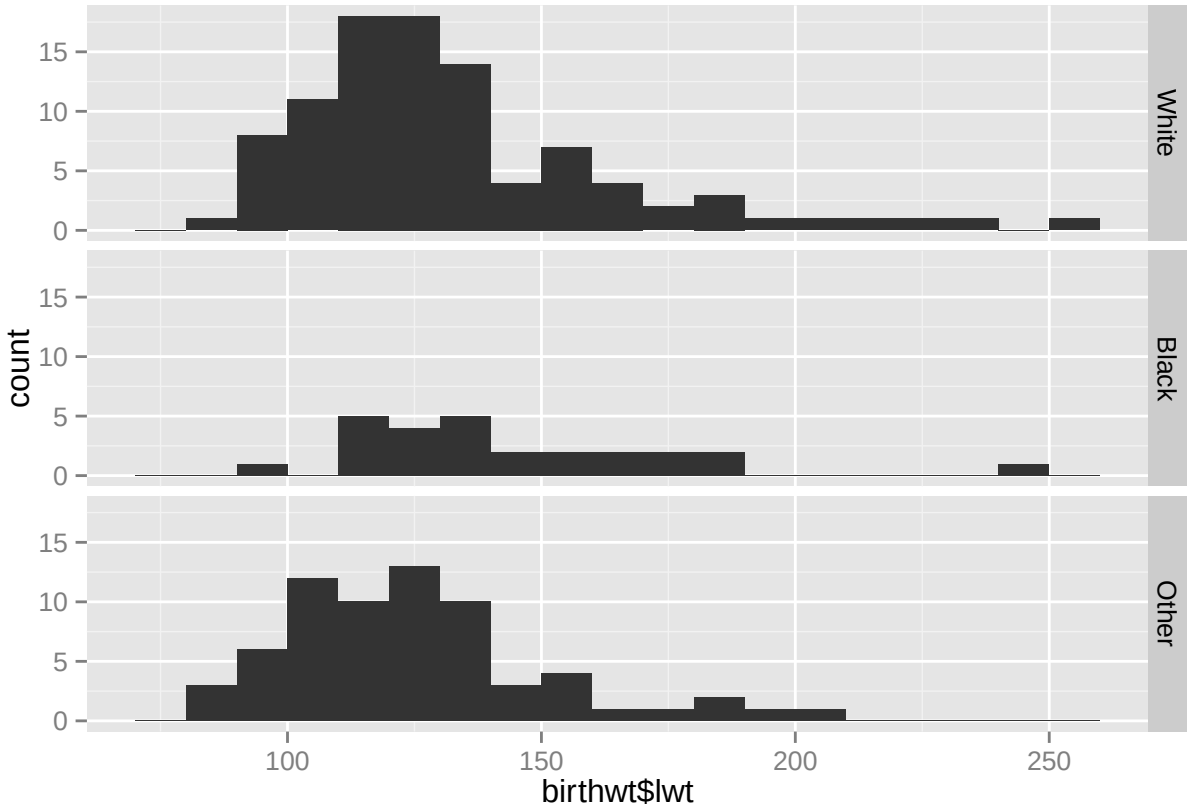
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



The option `facets = race ~ .` means, "Put each race on a different row." We'll talk more about this notation a little later.

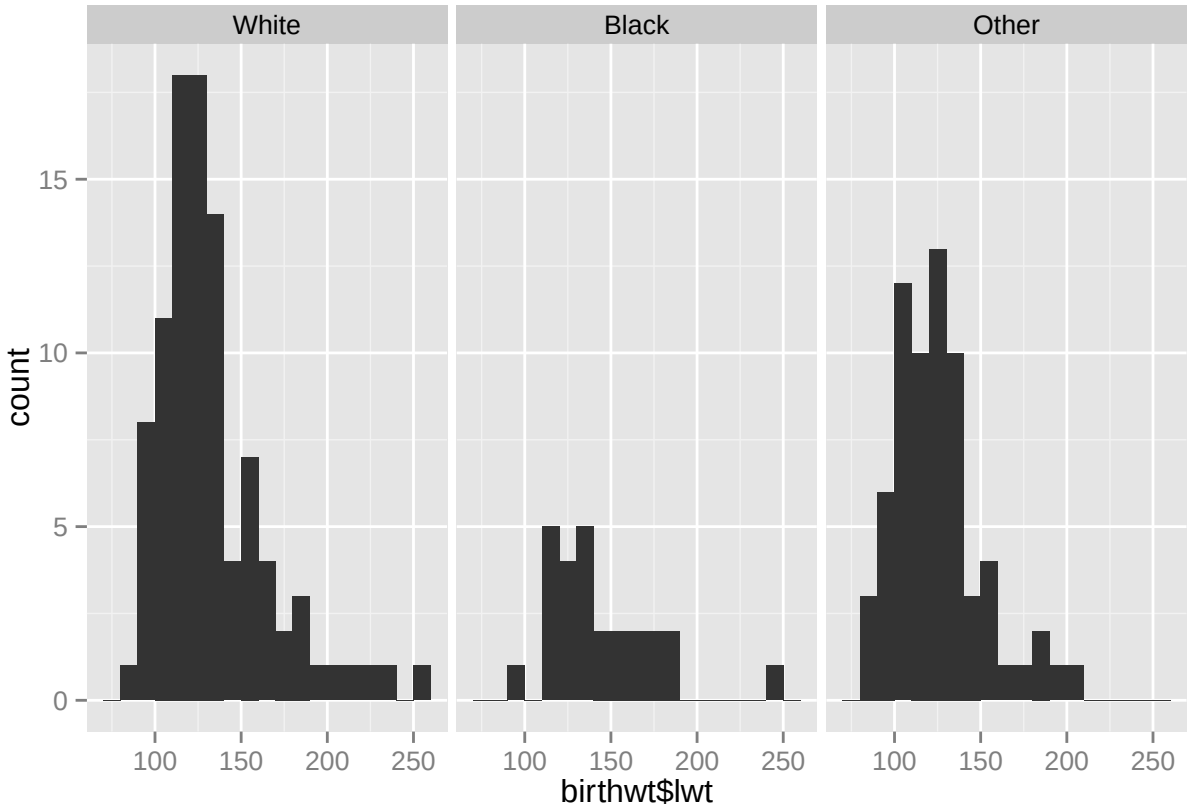
As always, the default bins suck, so let's change them.

```
qplot(birthwt$lwt, facets = race ~ ., binwidth = 10)
```



Consider the following subtle change in notation:

```
qplot(birthwt$lwt, facets = . ~ race, binwidth = 10)
```



Explain why that last graph (which might be called a side-by-side histogram) is less effective than the earlier stacked histogram. Can you figure out what's going on with the weird syntax of `race ~ .` vs `. ~ race`? The other thing that kind of sucks is the fact that the y-axis is showing counts. That makes it hard to see, for example, the black women, as there are fewer of them in the data set. It would be nice to scale these using percentages, but `qplot` doesn't play nice here either.

```
## Add code here to create both a side-by-side boxplot
## and a stacked histogram.
```

Your turn! Choose an interesting numerical variable and an interesting categorical variable. (If you pick a categorical variable that you haven't used before, be sure to convert it to a factor variable.) Create both a side-by-side boxplot and a stacked histogram. Discuss the resulting graphs. Comment on the association (or independence) of the two variables.

Summary statistics

Using base R, it's not so easy to get summary statistics for each group separately. Fortunately, the `mosaic` package comes to the rescue. When you load `mosaic`, it redefines many of the basic statistical commands to allow for more flexibility. For example:

```
favstats( ~ birthwt$lwt | race)
```

```
##    race min  Q1 median     Q3 max     mean       sd  n missing
## 1 White  90 112  129.5 143.25 235 132.0521 29.09381 96       0
## 2 Black  98 120  129.0 179.00 241 146.8077 39.63939 26       0
## 3 Other  80 105  119.0 130.00 250 120.0149 25.13026 67       0
```

The notation is, again, a little weird. Don't worry about the tilde for now. Just learn that it needs to be there. The important part is the `birthwt$lwt | race`. This says, "Look at the numerical variable `birthwt$lwt` broken down by `race`." Indeed, the output has three lines, one for each race in the data. This works for lots of commands, like `mean`, `sd` `median`, `IQR`, `quantile`, etc.

Your turn! Using the combination of one numerical and one categorical variable that you chose above, find the five-number summary of your numerical variable grouped by your categorical variable.

Conclusion

Whew! That was a lot of material covered in one assignment. As you come across new data in the future, use this assignment as a resource. Remember that you can't just make any type of graph for any type of data. Each combination of numerical and categorical variables will only have a few types of graphs that are appropriate. Be sure that you know each graph type well enough to use the correct one!