Graphing numerical data

Put your name here

Put the date here

Introduction

In this module, we will use the `ggplot2` package for creating nicely formatted charts and graphs for numerical data.

Instructions

Presumably, you have already created a new project and downloaded this file into it. Please knit the document and work back and forth between this R Markdown file and the PDF output as you work through this module.

When you are finished with the assignment, knit to PDF one last time, proofread the PDF file **carefully**, export the PDF file to your computer, and then submit your assignment.

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]...
```

Be sure to remove the line `## Add code here to [do some task]...` when you have added your own code.

Sometimes you will be asked to type up your thoughts. That will appear in the document as follows:

Please write up your answer here.

Again, please be sure to remove the line "Please write up your answer here" when you have written up your answer. In these areas of the assignment, please use contextually meaningful full sentences/paragraphs (unless otherwise indicated) 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. If you need to use some R code as well, you can use inline R code inside the block between `\begin{answer}` and `\end{answer}`, or if you need an R code chunk, please go outside the `answer` block and start a new code chunk.

Load Packages

We load the `mosaic` package as well as the `MASS` package for working with data on risk factors associated with low birth weight. (Note that the `ggplot2` package we will use for graphing is automatically loaded alongside the `mosaic` package.)

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

ggplot

The `ggplot` command is an all-purpose graphing utility. It uses a graphing philosophy derived from a book called *The Grammar of Graphics* by Leland Wilkinson. The basic idea is that each variable you want to plot should correspond to some element or "aesthetic" component of the graph. The obvious places for data to go are along the x-axis or y-axis, but other aesthetics are important too; graphs often use color, shape, or size to illustrate different aspects of data. Once these aesthetics have been defined, we will add "layers" to the graph. These are objects like dots, boxes, lines, or bars that dictate the type of graph we want to see.

In an introductory course, we won't get too fancy with these graphs. But be aware that there's a whole field of data visualization that studies clear and interesting ways to understand data graphically.

It will be easier to explain the `ggplot` syntax in the context of specific graph types, so let's proceed to the next section and start looking at ways to graph numerical data.

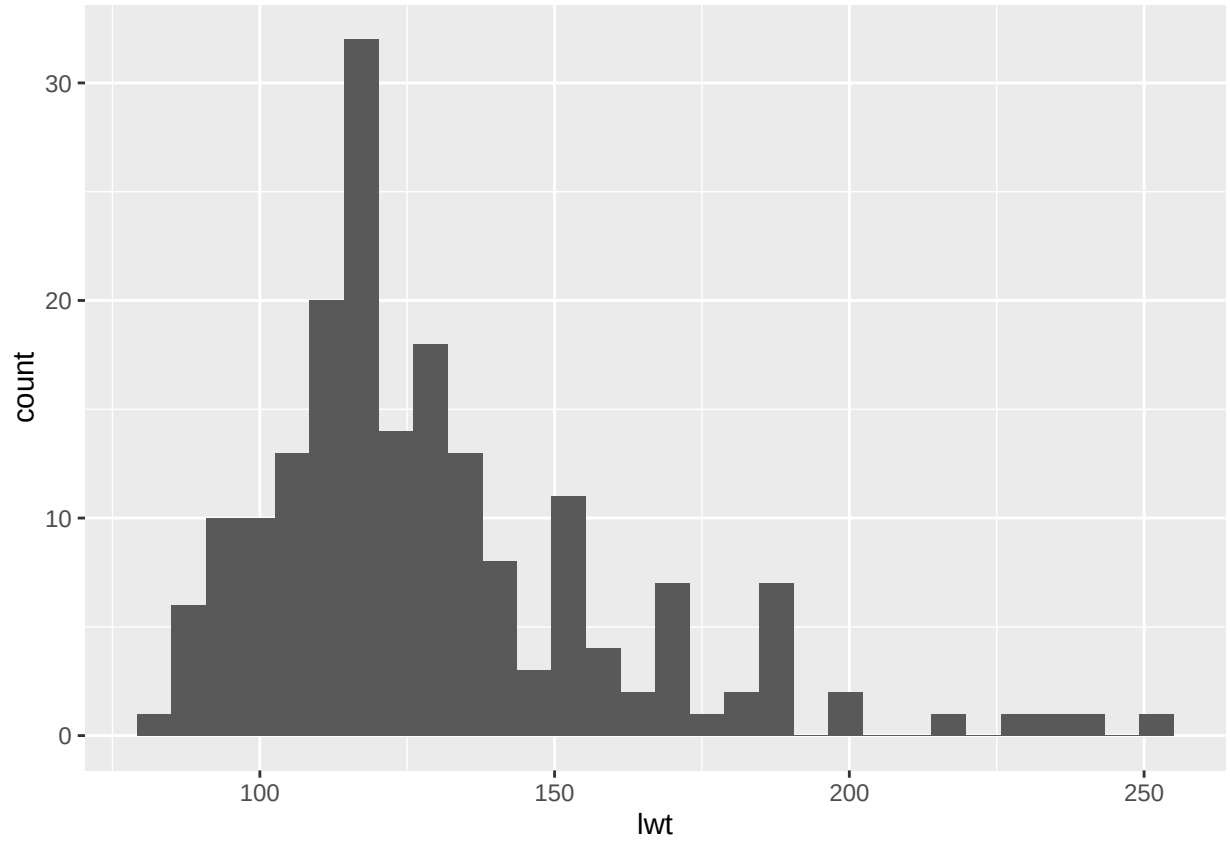
Graphing one numerical variable

From the birth weight data `birthwt`, let's consider the weight of the mother at her last menstrual period. This is clearly a numerical variable.

The single most useful display of a single numerical variable is a histogram. Here is the `ggplot` command to do that:

```
ggplot(birthwt, aes(x = lwt)) +
    geom_histogram()
```

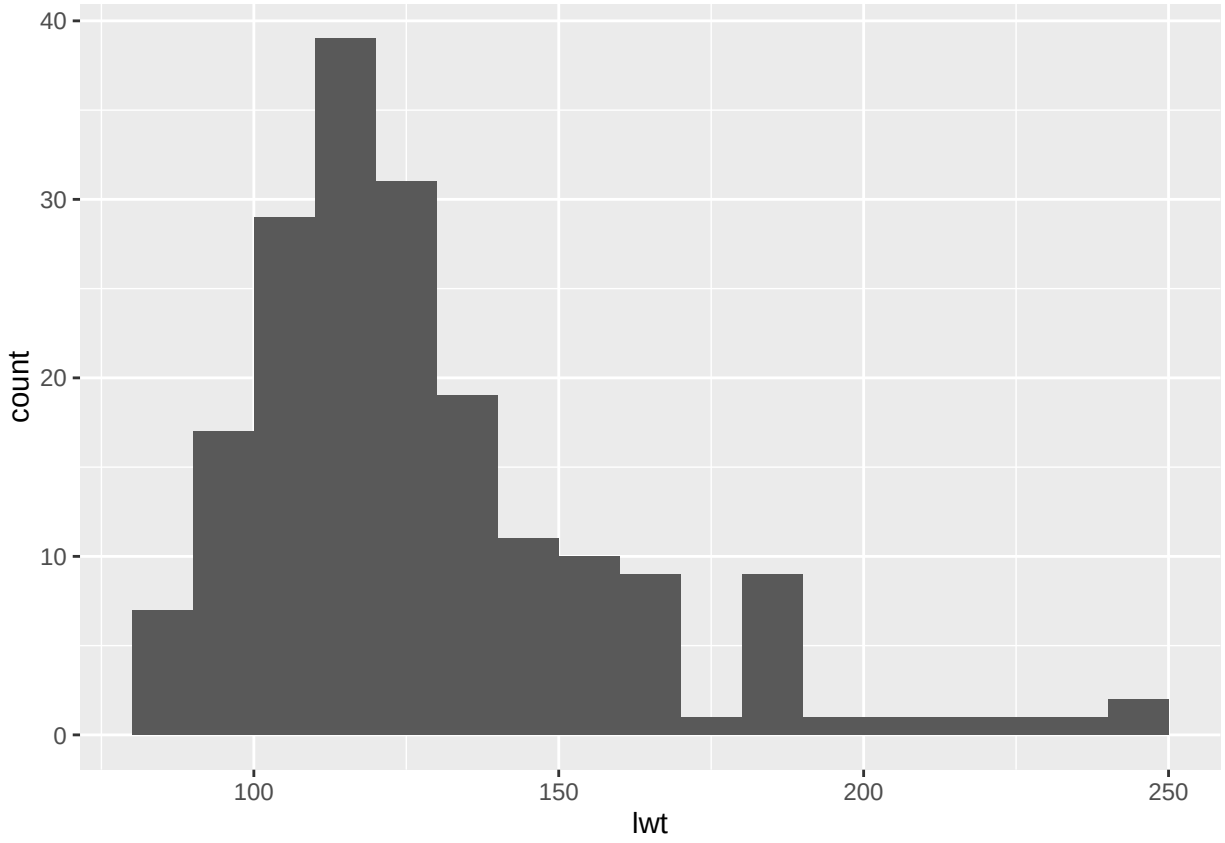
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let's walk through this syntax step by step. The first argument of the `ggplot` command is the name of the data frame, in this case, `birthwt`. Next we define the aesthetics using `aes` and parentheses. Inside the parentheses, we assign any variables we want to plot to aesthetics of the graph. For this analysis, we are only interested in the variable `lwt` and for a histogram, the numerical variable typically goes on the x-axis. That's why it says `x = lwt` inside the `aes` argument. Next, `ggplot` needs to know what kind of graph we want. Graph types are called "geoms" in the `ggplot` world, and `geom_histogram()` tells `ggplot` to add a histogram layer. (Adding a layer is accomplished by literally typing a plus sign.)

Generally, the default binning for `ggplot` histograms is not so great. In fact, if you knit this document and look at the output from the graphing command above, you can see that `ggplot` informs you that you can pick a better value. Let's try bins that are a little wider and that line up with numbers that are easy to see in the plot. Use `binwidth` to set the binwidth, and `boundary` to determine where you want one of the bin boundaries to fall. (The latter number is pretty arbitrary. We'd get the same graph if the boundary were set to 110 or 150, or any other multiple of 10.)

```
ggplot(birthwt, aes(x = lwt)) +
    geom_histogram(binwidth = 10,  boundary = 100)
```



Exercise

Write a paragraph or so describing the shape of the distribution of the `lwt` variable, focusing on the three key features (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.

Please write up your answer here.

Your turn

Create a histogram of the baby's birth weight (in grams). Adjust the binwidth if necessary to more clearly see the shape of the distribution. Then describe the shape of the distribution as you did for the `lwt` variable above.

```
## 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.
```

Please write up your answer here.

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 as they are just a messier version of histograms. I do like boxplots, but they are typically less informative than histograms. Boxplots are much better for comparing groups, so we'll see them in another module. Besides, there isn't an easy way to make a boxplot for a single numerical variable using `ggplot`.

Graphing two numerical variables

The proper graph for two numerical variables is a scatterplot.

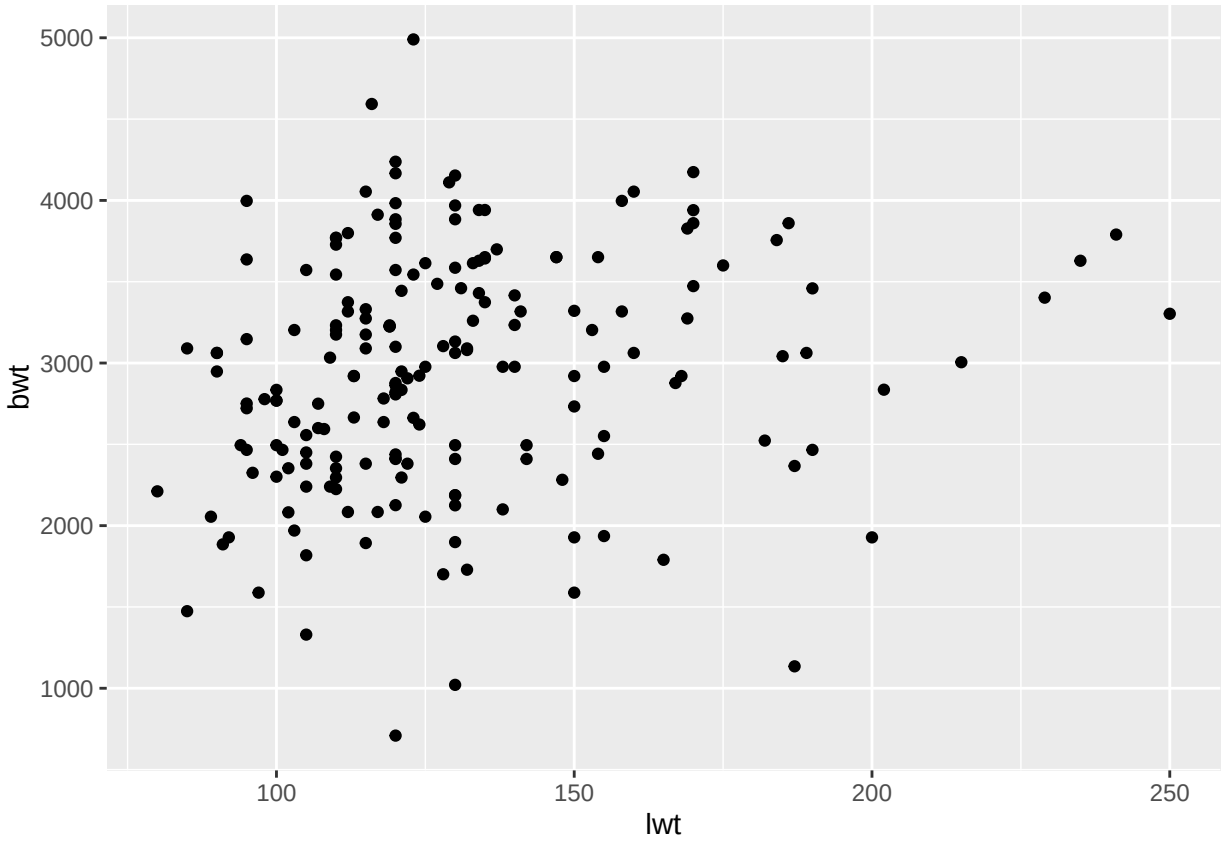
Exercise

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 not going to be proven from observational data. Instead, when there is an association, we say that the explanatory variable might be used to "predict" the value of the response variable.

Please write up your answer here.

Now we'll create a scatterplot of the weight of the mother at her last menstrual period and the birth weight of the baby. Since we are now plotting two variables, we have two aesthetics, one on the x-axis and one on the y-axis. Since scatterplots use points to plot each data value, the correct layer to add is `geom_point()`.

```
ggplot(birthwt, aes(x = lwt, y = bwt)) +
    geom_point()
```



Exercise

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.

Please write up your answer here.

Your turn

Consider the two variables `age` and `bwt`. Which variables would you consider as explanatory and response?

Please write up your answer here.

Now create a scatterplot to visualize the relationship between the mother's age and the birth weight of the baby.

```
## Add code here to create a scatterplot using age and bwt.
```

Now comment on the nature of the association.

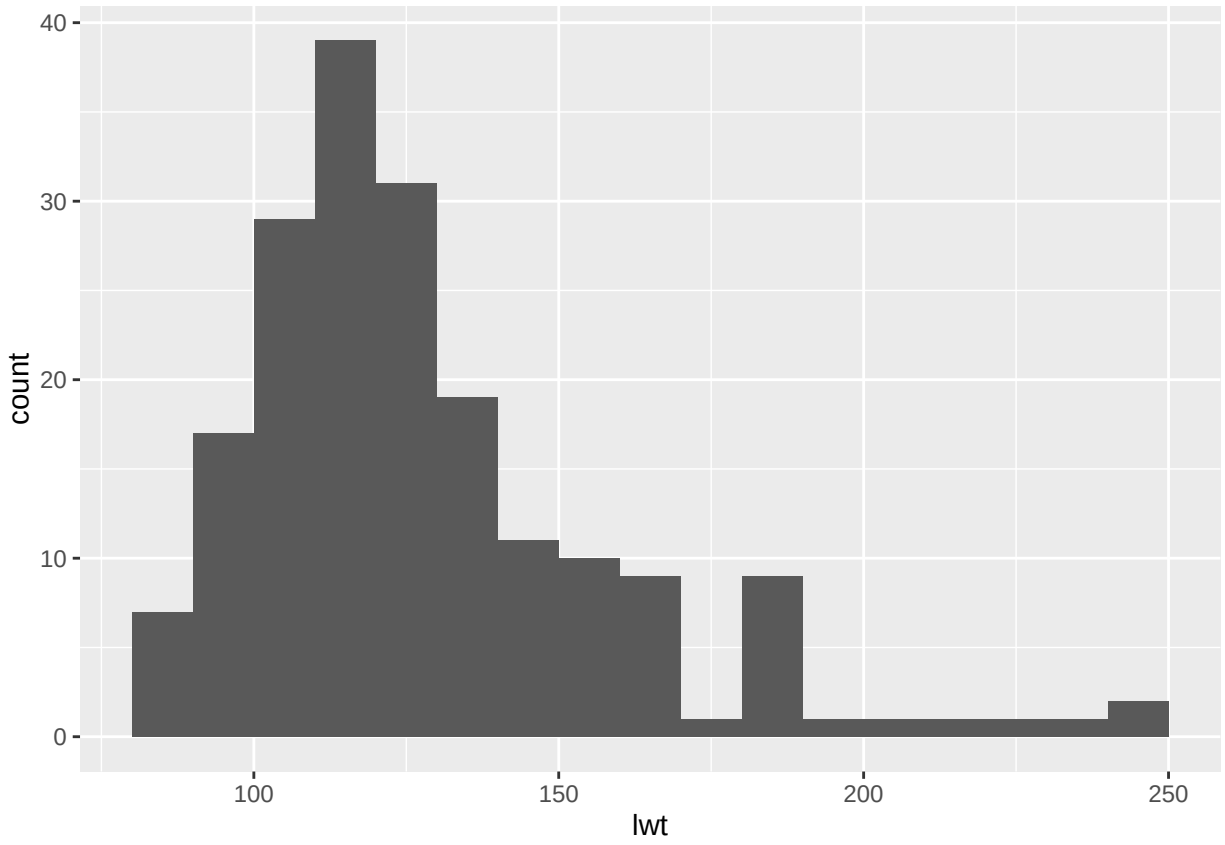
Please write up your answer here.

Publication-ready graphics

The great thing about `ggplot2` graphics is that they are already quite pretty. To take them from exploratory data analysis to the next level, there are a few things we can do to tidy them up.

Let's go back to the first histogram from this module.

```
ggplot(birthwt, aes(x = lwt)) +
    geom_histogram(binwidth = 10,  boundary = 100)
```



Note that the variable names of this data set are not terribly informative. In other words, if you were using this graph in a publication or presentation for an audience, they would have no idea what `lwt` was. Also note that this graph could use a title. We can do all this with `labs` (for labels). Observe:

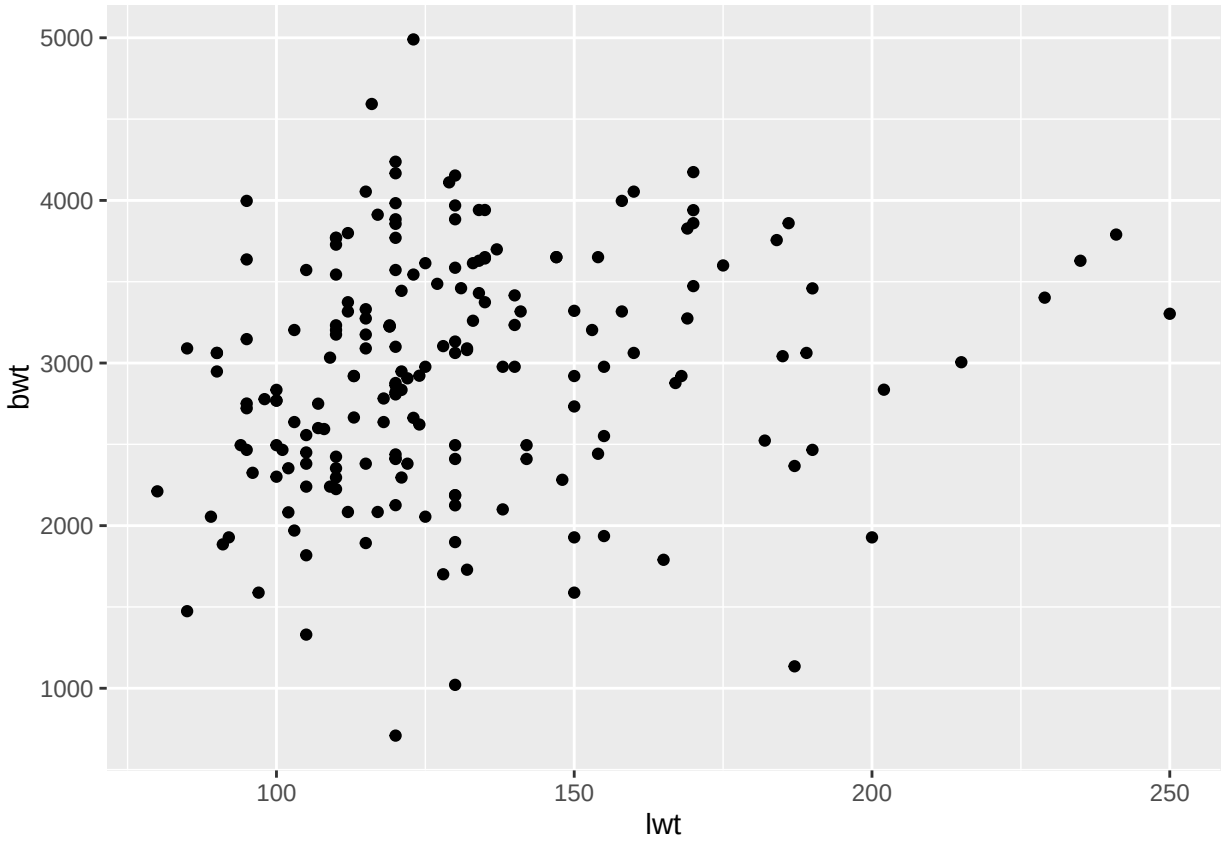
```
ggplot(birthwt, aes(x = lwt)) +
    geom_histogram(binwidth = 10,  boundary = 100) +
    labs(title = "Distribution of mother's weight at last menstual period",
         x = "Mother's weight (in pounds)")
```

You can also see that we took the opportunity to mention the units of measurement (pounds) for our variable in the x-axis label. This is good practice.

Exercise

Modify the following scatterplot by adding a title and labels for both the x-axis and y-axis.

```
## Modify the following scatterplot by adding a title and
## labels for both the x-axis and y-axis.
ggplot(birthwt, aes(x = lwt, y = bwt)) +
    geom_point()
```



Every part of the graph can be customized, from the color scheme to the tick marks on the axes, to the major and minor grid lines that appear on the background. We won't go into all that, but you can look at the ggplot2 documentation online and search Google for examples if you want to dig in and figure out how to do some of that stuff. However, the default options are often (but not always) the best, so be careful that your messing around doesn't inadvertently make the graph less clear or less appealing.

Conclusion

The `ggplot2` package with its `ggplot` command is a very versatile tool for creating nice graphs relatively easily. For a single numerical variable, the standard graph type is a histogram. For two numerical variables, use a scatterplot.