Lab 4: Exploring and Graphing Birth Data using Bar Plots

Graphing Birth Data

This is a lab that builds on the options you learned in Labs 1 and 2. We are using the same data that you used in Lab 2 in order to learn how to graph it.

This lab reviews the following topics from the first 3 labs:

1. How to load packages into R
2. How to use comparisons and `filter()` to select a subset of data
3. How to generate summary statistics using the `aggregate()` command

When you complete this lab, you should know the following:

1. How to use ggplot to generate a histogram
2. How to use ggplot to generate a bar graph with means (by combining it with the aggregate command)
3. How to use theme elements to change the axis labels and title of the plot, as well as the appearance of the bars.

Before you get started, create a new folder and a new R notebook

The first thing you will have to do is load the `fivethirtyeight` package and the `tidyverse` package that you installed in the last lab. If you don't remember how to do this, check in lab 3.

Then you should load the information from US_births_2000_2014 into the dataframe `birth` in the same way you did in lab 3.

Once we install packages, we have to load them. You only have to install packages once but every time you reopen R, you have to load the correct packages. We can load them by using the library() command.

Type `View(birth)` to make sure your data loaded correctly. As a reminder, here are what the columns are

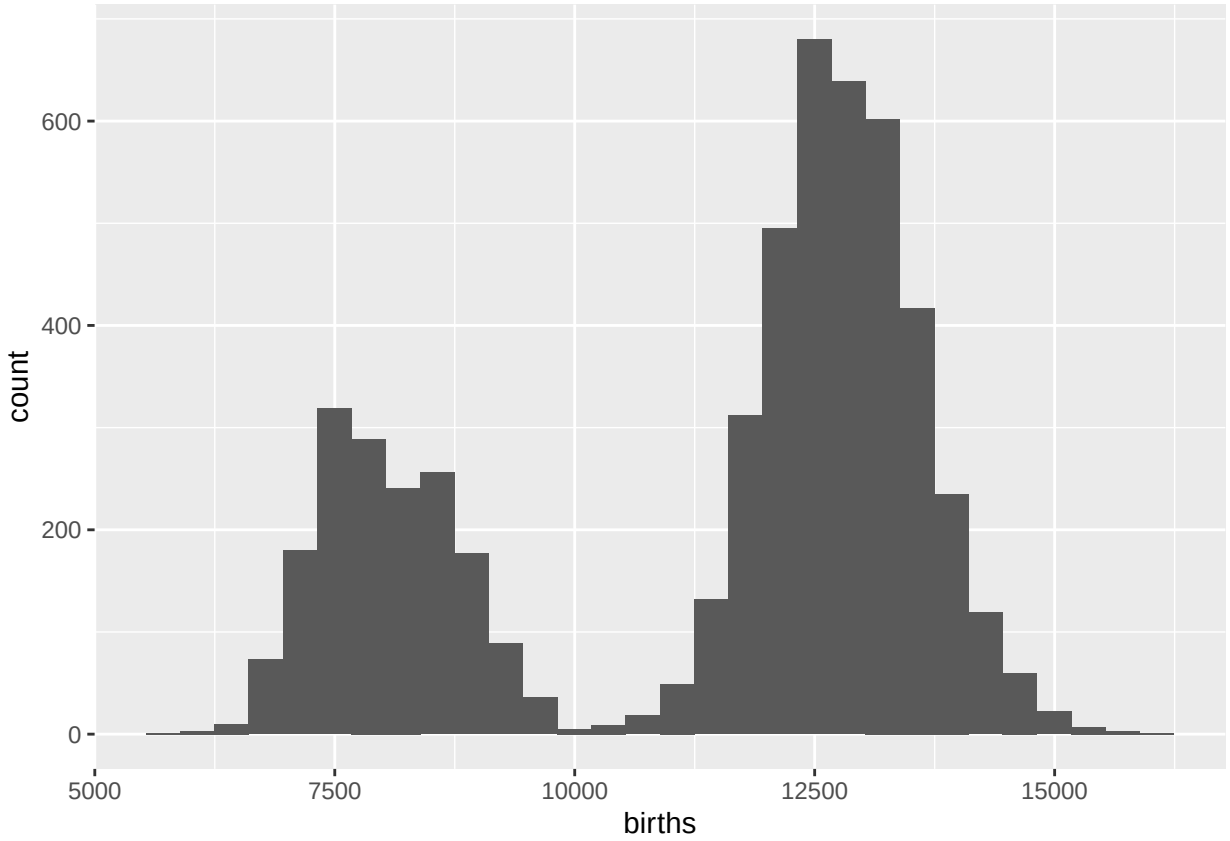
- year: Year
- month: Month
- date_of_month: Day
- day_of_week: Abbreviation of day of week
- births: Number of births

Making histograms

R has a built-in plotting functions, like the `hist()` command to make histograms but ggplot allows for more elegant histograms. Here is how we would make a histogram to see all the elements for births:

```
ggplot(data = birth) + geom_histogram(aes(x = births))
```

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

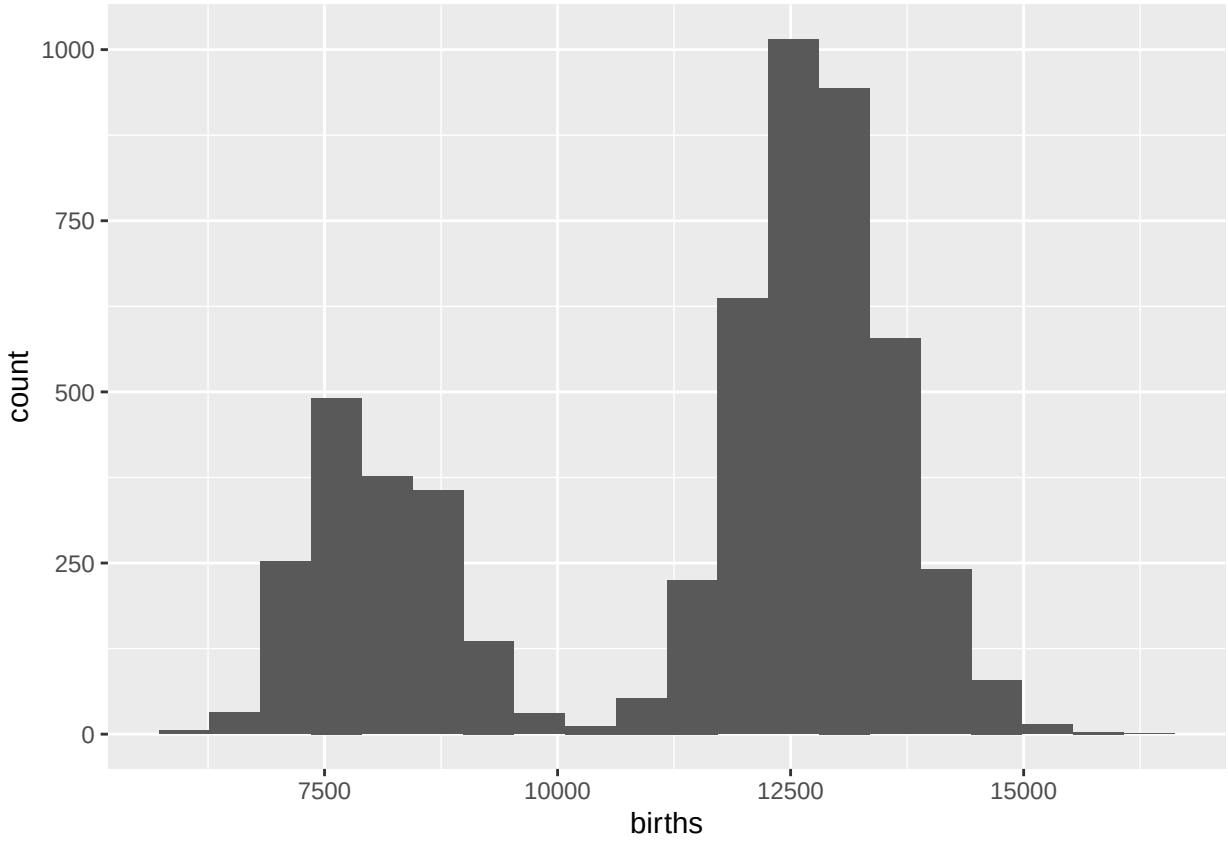


Type this command into R. You should get a histogram that looks like the one pictured here. You may also get a warning, like the one above that says that ggplot chose to use 30 bins.

The ggplot command has a few parts. The first command, the `ggplot(data = birth)` is where you tell ggplot which data frame you are analyzing. The second part is the `geom_histogram()` which is where you tell ggplot that you want a histogram. To tell ggplot what data to graph, you have to add the `aes(x = birth)` to say to plot the birth column.

The next thing we might want to do is to change the number of bins. Try typing the following to use 20 bins instead of 30:

```
ggplot(data = birth) + geom_histogram(aes(x = births), bins = 20)
```



1. What would you type into R to plot 40 bins?

Changing the appearance of the bars

The bars might be hard to see because they are gray blobs. We can change the appearance of the bars by adding extra options to the `geom_histogram()` part. Here are a couple extra options that might be useful:

- fill: changes the color of the bars themselves
- color: changes the color of the outline that surrounds the bars
- size: changest the size of the outline that surrounds the bars. This is a number with 1 being the default. To make it thicker, use a number like 2 and thinner would use a decimal less than 1 (like 0.5).

If you want to find out all the possible colors in R, you can type `colors()`

If I want to change the histogram for each bar to have a black outline and the bars to be blue, I would type the following:

```
ggplot(data = birth) + geom_histogram(aes(x = births), bins = 20, color = "black", fill = "blue")
```

And if I want to create a histogram with pink bars with dark green outlines that are thicker, I would do the following:

```
ggplot(data = birth) + geom_histogram(aes(x = births), bins = 20, color = "darkgreen", fill = "pink", si
```

Try typing these examples.

2. Imagine the plot below. Describe to me what this plot should look like. What color will the bars be, and the outlines. Write an annotation about what you expect. Then plot it and see if it matches up. Did it match what you expected?

```
ggplot(data = birth) + geom_histogram(aes(x = births), bins = 40, fill = "yellow", color = "black" , si
```

3. Create a code chunk containing a plot that has thick brown lines and has a orange fill of each bar. What did you type?

Using `filter()` to make separate histograms for weekday and weekends

In this section, I want you to combine what you learned in Lab 2 and so far in lab 3 to make two histograms, one histogram for weekdays and one for weekends. Your histogram should have the following:

- The weekday one should have black lines and "tan" fill, with 20 bins. The weekend one should have "steelblue" lines and "plum" fill, with 30 bins.

To do this, you will have to do the following. I would advise you place all of this in one code chunk.

1. Use filter to create a temporary data frame called `weekday` and `weekend`.
2. Use the weekday data frame as the input to ggplot when you create the histogram for weekday.
3. Use the weekend data frame as the input to ggplot when you create the histogram for weekend.
4. Create the histograms for weekday and then for weekend to answer the question.

Plotting counts of variables

The next type of plot we are going to do is look at plotting how many people were born on each day of the week. In Lab 2 and Lab 3, we discussed using the `aggregate()` function to take a dependent variable and calculate a summary statistic of that variable, based on different levels of a different variable. For example, if we wanted to calculate the median number of births for each day of the week, we would type:

```
aggregate(births~day_of_week, birth, median)
```

To plot this data, we have to do two steps. The first step will be to use the aggregate function to get the data for each day of the week. But instead of outputting that to the screen, we will put it in a temporary dataframe that we will then use in ggplot to plot. The first step I will do is create a temporary dataframe named `d` which holds the data from the last step.

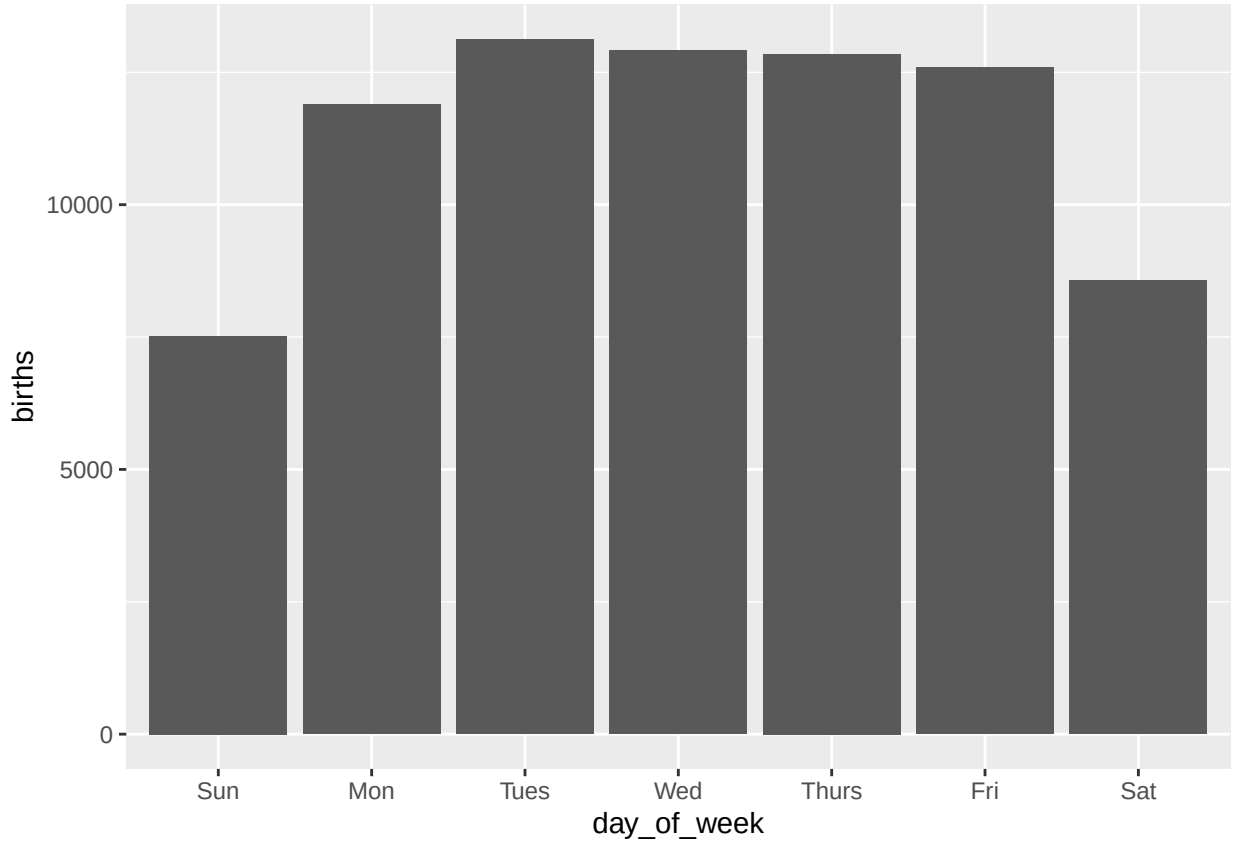
```
d = aggregate(births~day_of_week, birth, mean)
```

The next step will be to plot this data. The command has the same options as before, except in this case, we will have to use the `geom_bar()` geom, instead of the `geom_histogram()` option.

If we look at the temporary dataframe `d` that we made, there are two variables. The first variable is `day_of_week` and the second variable is `births`. We want `births` to be the y-variable since it will be plotted on the y-axis, and so the `day_of_week` is the x-variable. As a rule of thumb, the y-variable will always be numeric.

The only thing else to notice here is that I added the `stat = "identity"` part. This tells ggplot to plot the numbers themselves, rather than to do some sort of transformation to them.

```
ggplot(d) + geom_bar(aes(x = day_of_week, y = births), stat = "identity")
```



I advise doing this in one code chunk that contains both the aggregate and the ggplot command. with both commands, so you know they are working together. Now answer the questions below, with one code chunk per question, and an annotation telling me what question you are answering.

4: How would you change what you wrote above to plot means instead of medians?

5. How would you change the appearance to have black lines around the gray bars.

Another thing we may be interested in is plotting how many people were born on each date of the month (the first, second, etc), we may want to look at plots of subsets. What if we wanted to see the pattern of births for December? We could combine all the things in this lecture and do that. To do this, first we need to make a new data frame that is the subset of the birth data for just December using filter, then aggregate it for each day, and then plot it. Here is the code:

```
dec = filter(birth, month == 12)
d = aggregate(births~date_of_month, dec, mean)
ggplot(d, aes(x = date_of_month, y = births)) + geom_bar(stat = "identity")
```

Question 6: Now plot a graph like the one above, which shows the mean number of poeple born on each day of the month in January. Are there any dates of those months that are lower than any others?

Question 7: Use the filter command to select only the month of March. Then use the aggregate command to calculate the mean number of people who were born on each day of the week. Plot this data.

Changing the axis titles by using filter elements

The plots you've made so far look good, but you may want to change the labels of each axis. `day_of_week` may not be the best looking x-axis for instance. If we want to change these options, we can add the `labs()` command. The `labs()` command has three options:

- title: changes the title
- x: changes the x-axis label
- y: changes the y-axis label

For example, if we want to change the labels for the graph plotting the number of people born in December, we would add the following.

```
dec = filter(birth, month == 12)
d = aggregate(births~date_of_month, dec, mean)
ggplot(d, aes(x = date_of_month, y = births)) +
  geom_bar(stat = "identity") +
  labs(title = "Mean number of births in December", x = "Date of month", y = "Number of Births")
```

Question 8: Create a plot that has the mean number of births for each day in July. Use what you've learned about formatting to make the graph look pretty and add appropriate axis labels. Make sure you have annotations which indicate what you are doing.

Question 8: Create a plot that gives the mean number of births for each month. Your x-axis should have the month of the year and your y-axis should have the mean number of births. Give the plot appropriate axis labels and make it look nice. What did you type?

Summary

Now you are done, so go ahead and knit the notebook and submit.

When you're done with this lab you should know how to do the following and what the following commands do. It may also be a good idea to make sure you save

- How to create a histogram using `ggplot()`, `geom_histogram()` and the `aes()` function
- How to change the color and size of the bars and the outline of the bars
- How to use `filter()` in the tidyverse package to create a subset of data for a histogram
- How to plot means using `aggregate()` to create a subset and the `geom_bar(stat = "identity"` options
- How to use the `labs()` command to change the plot's labels.