Lab 5: Customizing Graphs in R

Introduction

TODO

Introduce mosaic plots.

Start a New Project in RStudio

We start similar to the last lab. We will start a new project based on a prepared repository.

- In RStudio, go to File > New Project > Version Control > Git.
- Paste this URL in: https://github.com/HanoverStatsLabs/Lab-Customize-Graphs.git
- Press Tab to go to the next field — it should auto-complete the directory name.
- Make sure the parent directory is the folder where you want to keep your projects for this class.
- Click Create Project.
- Your Files pane should now show the provided files, in particular a file titled Lab5Report.Rmd. Click the file name to edit the file, and then run all the R chunks.
- Recall that we will mostly be editing this report file, rather than working directly in the console. You may want to make the report window larger than the console window.

Import and View Dataset

- Use the data command to load the dataset named brfss. Include this in an R chunk in your report.
- Use the View command *in the console* to see the data in the Preview window. **Note**: Do not put the View command in an R chunk, it will prevent your report from knitting/compiling. Similarly, the ? command should not be put into your report.
- You should see that the data has 23 variables and over 1.3 million rows.
- Use the ? command *in the console* to view information about the brfss data file (the "codebook"): ?brfss

General Health (`genhealth`)

1. Make a tally and a barchart (as you did for WeekDays in a previous lab) for the genhealth variable. Which category of the variable is most frequent?

2. It would be useful to have **percentages** rather than raw counts in our tally. Add a *command option* by inserting the code , format="percent" before the closing parentheses of the tally command, like this:

```
tally(~genhealth, data=brfss, format="percent", useNA="no")
```

What percent of respondents report being in either "Very good" or "Excellent" health? What percent of respondents did not answer the question (NA)?

3. In the barchart, we like to have control of the **bar colors**. This is controlled with the col= option. For example, add the option , col="green" to redraw the genhealth barchart with green bars.

   To see all the named colors, use the command colors(). (Do this in the console, or hide the output if you put it into a code chunk – it's a *long* list.) Experiment with some of the named colors to find one you like, and include that plotting command in your report.

   To make gradations or use colors from other sorts of palettes, you can use the RColorBrewer package. To learn more details of working with color palettes, and colors in general, see the Colors section of the commands documentation[1]. For instance, try the following:

   ```
   tally(~genhealth, data=brfss) %>% barchart(col=brewer.pal(5, "YlGnBu"))
   ```

4. Up to this point, we have been uncritical of the default **chart labeling**. Let's see how to take control of this in R. In particular, for our barchart, we should improve the x-axis label and add an overall title.

   A correct label for the x-axis is "Percent of Respondents". To add an x-axis label, use the 'xlab= option, as follows:

   ```
   tally(~genhealth, data=brfss, format="percent", useNA="no") %>% barchart(col=brewer.pal(
   ```

   To add a main title, like "Distribution of General Health", use the 'main= option:

   ```
   tally(~genhealth, data=brfss, format="percent", useNA="no") %>%
       barchart(col=brewer.pal(5, "YlGnBu"), xlab="Percent of Respondents",
           main="Distribution of General Health")
   ```

5. **Pie charts** are popular, but also tend to be over-used. Once in a while it's nice to have one, as long as "proportion of the whole" is the item of interest. To show the genhealth responses in a pie chart, try the following:

   ```
   tally(~genhealth, data=brfss, format="percent", useNA="no") %>%
       pie(col=brewer.pal(5, "YlGnBu"))
   ```

General Health vs. Exercise

Stacked bar graphs are nice for visualizing the relationship between two categorical variables. Unfortunately, this is one of the tricker plots to make in R. Suppose we're interested in the relationship between genhealth and exerciseany. We expect to see that those respondents who never exercise have worse general health.

[1] ../commands.html

6. Make a tally of percentages for genhealth against exerciseany, and store the result:

```
healthVsExercise <- tally(~genhealth|exerciseany, data=brfss,
    format="percent", useNA="no")
healthVsExercise
```

You should see a table with column-wise percentages. Does it look like the "exercisers" tend to have better general health than the "non-exercisers"? Explain.

7. Because of the limitations of the barchart command, we have to transpose the table (swap the rows and columns) before piping into the barchart. We use the t() command to transpose. auto.key is used to make a legend for the barchart. The whole thing looks like this (try the various versions below):

```
healthVsExercise %>% t() %>% barchart(auto.key=list(columns=3))
healthVsExercise %>% t() %>% barchart(auto.key=list(space="top"))
healthVsExercise %>% t() %>% barchart(auto.key=list(space="right"),
    main="General Health vs. Does the Respondent Ever Exercise")
```

8. If you want to specify a color palette, you really have to dig deep. Let's give it a try. The brewer palette which ranges from red to purple will make an attractive spectrum. Let's store it with 5 color values:

   ```
   myColors <- Brewer.pal(5, RdPu)
   ```

   Store the levels of the genhealth variable:

   ```
   healthLevels <- list(levels(brfss$genhealth))
   ```

   Build the legend myKey using the levels for genhealth and the colors in myColors:

   ```
   myKey <- list(text=healthLevels, rectangles=list(col=myColors))
   ```

   Draw the stacked barchart with the new palette in both the bars and the legend, and give it a main title:

   ```
   healthVsExercise %>% t() %>% barchart(col=myColors, key=myKey,
       main="General Health vs. Does the Respondent Ever Exercise")
   ```

   Does the relationship between genhealth and exerciseany prove that if a person exercises they will improve their health? Explain.

General Health vs. Income and Age Group

Now let's see how income level and age group might be related to general health for this group of respondents.

9. **Before looking at any relevant graphs**, make some predictions. Do you think there is any relationship between income level and health? What about the relationship between age and health? Explain.

10. Using what you learned previously in this lab, make a nice stacked bar graph for general health vs. income level (income). What do you learn from the graph? Compare with your prediction above.

11. Also make a stacked bar graph for general health vs. age. What do you learn from the graph? Compare with your prediction above.

Let's investigate the 3-way relationship among general health, income, and age. In particular, is the relationship between general health and income constant across all ages, or does it vary with each age group? This calls for a 3-way graph, plotting general health against income and using age as a "panel" variable. The formula syntax for this is ~genhealth|income+age. Store the tally and take a look at it:

```
healthVsIncomeAndAge <- tally(~genhealth|income+age, data=brfss,
    format="percent", useNA="no")
healthVsIncomeAndAge
```

This makes a rather overwhelming table of numbers! To help the average human make sense of this, we will make a paneled stacked barchart. There is a "magic" bit of code here, aperm(c(2,3,1)), taking the place of t() for 3-way graphs.

```
healthVsIncomeAndAge %>% aperm(c(2,3,1)) %>%
    barchart(col=myColors, key=myKey)
```

12. For each age group, is it the case that those with higher income level report better general health? Explain how you see this in the graph.

13. Is the nature of the relationship between general health and income exactly the same across all ages, or do you notice it being weaker for some age groups? Explain what you see and how that makes sense.