Lab 7: Linear Modeling

Introduction

In this lab we will learn how to work with linear models in RStudio.

Overall Goals

In this lab we will learn how to:

- add a data file to a project by importing from Excel,
- customize scatterplots,
- fit, visualize, and analyze linear models,
- and interpret residual plots.

Make Your Own Project in RStudio

As we did in the previous lab, start a new project:

- In RStudio, go to File > New Project > New Directory > Empty Project
- Make sure the parent directory (second textbox) is the folder where you want to keep your projects for this class. Use the browse button if necessary.
- Enter a name for the new directory; it is good practice to avoid spaces in your file names by using underscores in place of spaces. For example, Lab_7 would be a good name.
- Click Create Project.
- In the Files pane, you should now see a file Lab_7.Rproj. This is the project configuration file, and you don't need to do anything with it.

Now you need to start a new RMarkdown report file.

- Go to File > New File > R Markdown....
- In the Document tab of the resulting dialog, give your report a title (Lab 7 Report) and put your name in the Author textbox. Keep HTML as the output format. Click OK.
- You should now see your new RMarkdown document at the upper left; **save** it. In the Save File dialog, enter the file name with no spaces: Lab7Report. (Do not use a filename extension.)
- Everything below the *first* provided code chunk is boilerplate and **should be removed**. Do so now.
- **Note**: The top-level section heading (one #) is already in use in this document for the report title. Use second-level headings (##) for the main sections of the report. (If you need subsections, use ###.)
- Use the Insert pulldown to add a new R code chunk. Add the command library(hanoverbase). Use the chunk options dialog to disable warnings, disable messages, and "show nothing (run code)".
- **Run the chunk** which you just created.

Import From Excel

In order to add a data file to the project itself, we start by uploading an Excel file:

- Download the Excel file that holds the relevant data: hanoverstatslabs.github.io/resources/datasets/guns.xlsx
- In the **Files** pane (lower right), click the upload button to start the Upload Files dialog.
- Click Choose File and navigate to your Downloads folder. Find the file **guns.xlsx** and click on it.
- Click OK to finish the upload. If this was successful, you should now see guns.xlsx in your Files pane.

Now we need to import the actual data from the Excel file into the RStudio project (into both the console and into the report):

- Click on the guns.xlsx file. Take the option Import Dataset. This should bring up the Import Excel Data dialog.
- If the preview looks reasonable, click the Import button to do the import. You should now see the guns data in the data viewer, and a couple of lines of import code in the console.
- Make a new R code chunk in your report (below the hanoverbase chunk).
- There are two lines of code to copy from the console and paste into the new chunk. The first looks like library(readxl) and the second begins guns <–. You do not need to run this chunk at this time, since we've already run those commands.

Explanation of the Variables in the Guns Dataset

We created the file guns.xlsx from data provided at www.openintro.org/stat/data. The four variables in the data are as follows:

- country: Name of the country.
- mort_rate: The number of gun-related deaths per 10,000 population.
- own_rate: The number of guns per 100 population. Note that this counts numbers of guns, not numbers of people. (Do you think it includes unregistered guns?)
- hdi: The country's numeric Human Development Index (a composite statistic of life expectancy, education, and per capita income indicators).

Statistical Investigations

You are now ready to start working on your report. The sections below give you questions to answer and commands to try. Here are a few reminders:

- **Important formatting note**: As you work through this lab assignment, answer the questions which are posed by typing into your R Markdown report, using formatting elements to make the report easy to read. See: R Markdown Syntax Sheet[1]
- **R Chunks**: As usual, create R Chunks (use the Insert pulldown) for your R commands. Use the R Cheatsheet for help as needed.
- **Knit Early and Often**

[1] ../rmarkdownBasics.html

Warmup

To get warmed up and familiarize ourselves with the variables, we will start with some one-variable investigations.

1. Make a favstats summary and a histogram for the own_rate variable; you should adjust the number of breaks in order to get a good view.

   Also make a labeled dotplot to show the countries in the dataset, sorted by gun ownership rate:

```
sum(~own_rate|country, data=guns) %>% sort() %>% dotplot()
```

   You should see two clusters in the dotplot, a couple of high outliers, and one extremely high outlier.

   Describe the distribution of own_rate. What are the outliers and what are their gun ownership rates (number of guns per 100 population)?

2. Do the same for mort_rate. Describe what you find.

3. Do the same for hdi. Describe what you find.

Mortality and Gun Ownership

We wonder if countries with high gun ownership rates also have high gun-related mortality rates. We can investigate this with a scatterplot.

4. Make a scatterplot for mort_rate (y) vs. own_rate (x). Describe the overall pattern (including direction, if any) of the data and identify any unusual points. Attach a name to your plot so you can recall that plot in the future:

```
g1 <- xyplot(mort_rate~own_rate, data=guns)
g1     # this is just to display the graph
```

5. First we'll do some modeling without removing any observations. Let's start with a smooth fit curve.

```
ladd(panel.loess(x, y, col="<your color here>", lwd=2), plot=g1)
```

   Does it look like a linear model is a good fit for these data? Explain.

6. Let's see what a linear model looks like on this plot:

```
ladd(panel.lmline(x, y, col="<your color here>"))
```

   In order to find the equation of our linear model, we need to calculate the slope and intercept. And to assess how well the model fits the data, we should calculate the square of the correlation (R-square linear).

```
fit1 <- lm(mort_rate~own_rate, data=guns)
coefficients(fit)
r1 =<- cor(mort_rate~own_rate, data=guns)
r1
r1^2
```

a. Write the equation of the linear model.
b. Assess how well the linear model fits the data. Explain.

7. In order to further assess the appropriateness of the linear model, we look at a residual plot, showing the **residual** (y minus fitted) vs. fitted. We add a horizontal line at 0 to help us judge the presence of a pattern:

```
xyplot(resid(fit)~fitted(fit))
ladd(panel.abline(h=0))
```

Remember that any remaining pattern in the residual plot indicates an incomplete model.

Do you see that there is a pattern in the residuals, or do they look "unpatterned" for the most part.

8. Correlation and regression are both susceptible to the effects of outliers and other influential points. Let's see what happens when we filter out the U.S. from the data.

```
gunsFiltered <- ....   # pipe the 'guns' data through a filter to remove U.S.
```

Repeat the code for #4 - #7 with the newly filtered dataset. Change the names g1, fit1 and r1 to g2, fit2 and r2. In the new scatterplot, for example, you should **no longer see** the point for the U.S. (own_rate > 80).

a. Has your perception of the association changed? Explain.
b. Is there still a positive association? A linear association?
c. Has the correlation changed? If so, how?

9. Now we will show both linear models on the original (unfiltered) scatterplot. This demonstrates the effect of single influential observation on the modeling process.

```
ladd(panel.abline(fit1, col="black", lwd=2), plot=g1)
ladd(panel.abline(fit2, col="magenta", lwd=2))
```

Explain why the second line (fit2) has a smaller slope.

Other Relationships

10. To make a correlation table, use the cor command in a new way:

```
cor(guns[,2:4])  # uses columns 2 - 4 in the data
```

Use a similar command to make a second table using the gunsFiltered data.

To see the correlation between two variables x and y, for example, find x in the columns and y in the rows. The number for that row and column gives the correlation for that pair.

a. Explain why the correlations on the "main diagonal" will always be 1.0.
b. With which of the two rates is the hdi most strongly correlated?
c. hdi tries to measure how "advanced" the country is in terms of education, health, etc. How does it make sense that hdi is positively correlated with gun ownership rate?

11. We want to investigate whether the relationship between mortality rate and ownership rate varies depending on the hdi of the various countries. In order to do that, we will split the countries into groups based on their hdi. To find a good split, draw a histogram for hdi using a suitable number of breaks. Where would you split the values into different categories?

12. Create a variable hdicat which splits the cases depending on whether the hdi is lower or higher than 0.85. We pipe the hdi values into the cut command to create the desired breaks.

```
guns$hdicat <- guns$hdi %>% cut(breaks=c(0.75, 0.85, 0.95))
```

    If this worked correctly, you should see the new variable hdicat in the data viewer.

    Now we make a paneled scatterplot for mort_rate versus own_rate, paneled by hdicat. We use ladd to add the linear models and cor to find the correlations:

```
xyplot(mort_rate~own_rate|hdicat, data=guns)
ladd(panel.lmline(x, y))
cor(mort_rate~own_rate, data=guns %>% filter(hdi <= 0.85))
cor(mort_rate~own_rate, data=guns %>% filter(hdi > 0.85))
```

    What do you see in the resulting plot and correlations? Explain as best you can what is going on here.

Submissions

- Make sure to knit one last time, then download the Lab7Report.Rmd file: in the **Files pane** (lower right), click the checkbox for the RMD file, then choose More > Export ... > Download. *Do the same for Lab7Report.html.*
- **Submit both files via Moodle**.