Math 207
Introduction to Statistics

Introductory R Markdown Exercises

We have seen how to use R in interactive mode using RStudio. Typically we will use a tool called R Markdown to generate reports (web pages, pdfs, etc.) from RStudio.

1. **First R Markdown Test**. Open RStudio.

a) Select File → New File → R Markdown from the File menu. In the popup window, select HTML output. Also in this window, enter your name and call the file RMarkdown1. Click OK to create the file.

b) Now click the Knit button in R Studio to generate a web page.

2. **Add Your Own R Markdown Content**. Now delete everything in the RMarkdown1 file and replace it with the following. Recompile with Knit when you are done.

````
---
title: "RMarkdown1"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Here is my first experiment with R Markdown.
```{r}
2 + 3
```
````

3. **Inline R Code**. Add the following to the end of RMarkdown1 then recompile.

````
## Inline R Code
We can add some code inside text:  2 + 3 = `r 2 + 3`.
````

4. **Statistical Sampling**. Add the following to the end of RMarkdown1 then recompile.

````
## Statistical Sampling
Rolling a six-sided die is like randomly sampling numbered tickets
from the box containing `r 1:6`. Let's roll 10 dice.
```{r}
sample(1:6, 10, replace = TRUE)
sample(1:6, 10, replace = TRUE)
sample(1:6, 10, replace = TRUE)
```
````

5. **Law of Large Numbers I**. Add the following to the end of RMarkdown1 then recompile.

````
## Law of Large Nubmers I
Let's roll 100 dice and make a histogram of the results.
```{r}
x <- sample(1:6, 100, replace=TRUE)
x
table(x)
barplot(table(x)/100, xlab="Outcome", ylab="Frequency")
```
````

Why was table(x) divided by 100?

6. **Law of Large Numbers II**. Repeat #5 using 1000 dice then using 10,000 dice.

7. **Working With a Data Frame**. A data frame is similar to a spreadsheet. It is divided into rows and columns. Each column stores data of a particular type (dates, strings, etc.). Each row is, therefore, a list of values of the types stored in the rows.

a) Add the following to the end of RMarkdown1 then recompile.

````
## Working With a Data Frame
Let's look at the mtcars data set.
```{r}
class(mtcars)
head(mtcars)
```
````

b) We can compute statistics for numeric columns (either discrete or continuous). Add the following to the end of RMarkdown1 then recompile.

````
Now compute statistics for some of the columns.
```{r}
mean(mtcars$mpg)
median(mtcars$mpg)
sd(mtcars$mpg)
var(mtcars$mpg)
range(mtcars$mpg)
```
````

c) Add code to RMarkdown1 to compute the statistics from b) for engine horsepower (column `hp`).

d) Find the car with the highest mileage (column `mpg`).

````
Find the car with the highest mileage.
```{r}
mtcars[mtcars$mpg == max(mtcars$mpg, ]
```
````

e) Find the car(s) with the lowest mileage.

f) Add code to RMarkdown1 to explore the relationship between mileage and horsepower.

````
```{r}
plot(mtcars$hp, mtcars$mpg, xlab="horsepower", ylab="mileage")
cor(mtcars$hp, mtcars$mpg)
```
````

Are higher horsepower cars more or less fuel efficient than lower horsepower cars?

g) Explore the relationship between mileage and `qsec`. What does the variabe `qsec` mean? Use `?mpg` in `R` to get help.

8. **Generating a pdf**. At the top of your RMarkdown file, change `html_document` to `pdf_document` then recompile. Note that you will need to have TEX installed. See http://www.tug.org/mactex.