

Programming with R Test #1

Instructions

R markdown is a plain-text file format for integrating text and R code, and creating transparent, reproducible and interactive reports. An R markdown file (.Rmd) contains metadata, markdown and R code “chunks”, and can be “knit” into numerous output types. Answer the test questions by adding R code to the fenced code areas below each item. Once completed, you will “knit” and submit the resulting .html file, as well the .Rmd file. The .html will include your R code *and* the output. The .html file will be graded and returned with comments as a .pdf document.

Before proceeding, look to the top of the .Rmd for the (YAML) metadata block, where the *title* and *output* are given. Please change *title* from ‘Programming with R Test #1’ to your name, with the format ‘lastName_firstName.’

If you encounter issues with knitting the .html, please send an email via Canvas to your TA.

Each code chunk is delineated by six (6) backticks; three (3) at the start and three (3) at the end. After the opening ticks, arguments are passed to the code chunk and in curly brackets. **Please do not add or remove backticks, or modify the arguments or values inside the curly brackets.** An example code chunk is included here:

```
# Comments are included in each code chunk, simply as prompts

...R code placed here

...R code placed here
```

You need only enter text inside the code chunks for each test item.

Depending on the problem, grading will be based on: 1) the correct result, 2) coding efficiency and 3) graphical presentation features (labeling, colors, size, legibility, etc). I will be looking for well-rendered displays. Do not print and display the contents of vectors or data frames unless requested by the problem. You should be able to display each solution in fewer than ten lines of code.

Submit both the .Rmd and .html files for grading

Example Problem with Solution: Use `rbinom()` to generate two random samples of size 10,000 from the binomial distribution. For the first sample, use $p = 0.45$ and $n = 10$. For the second sample, use $p = 0.55$ and $n = 10$.

- (a) Convert the sample frequencies to sample proportions and compute the mean number of successes for each sample. Present these statistics.

```
set.seed(123)
sample.one <- table(rbinom(10000, 10, 0.45)) / 10000
sample.two <- table(rbinom(10000, 10, 0.55)) / 10000

successes <- seq(0, 10)

sum(sample.one * successes) # [1] 4.4827

## [1] 4.4827

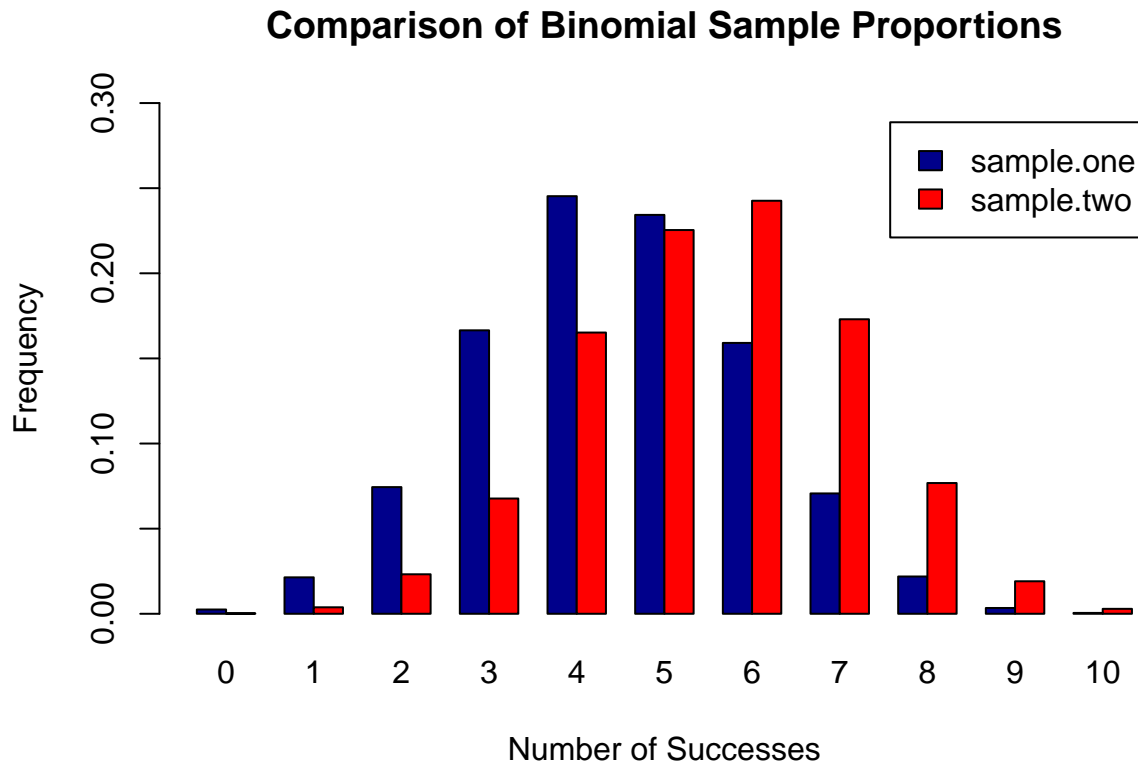
sum(sample.two * successes) # [1] 5.523
```

```
## [1] 5.523
```

(b) Present the proportions in a vertical, side-by-side barplot color coding the two samples.

```
counts <- rbind(sample.one, sample.two)

barplot(counts, main = "Comparison of Binomial Sample Proportions",
        ylab = "Frequency", ylim = c(0,0.3), xlab = "Number of Successes",
        beside = TRUE, col = c("darkblue", "red"), legend = rownames(counts),
        names.arg = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10"))
```



Please delete the Examples shown above prior to submitting your .Rmd and .html files.

Test Items

(1) (4 points) Create a vector that contains the following, in this order, and print the contents. Do not round off any values unless requested.

- A sequence of integers from 5 to 10, inclusive.
- A two repetitions of the vector $c(2, -5.1, -23)$.
- The value of the sum of $7/42$ and 3

(1)(a) Extract the first and last elements of the vector you have created to form another vector using the extracted elements. Print this vector.

(1)(b) Form a third vector from the elements not extracted in (a). Print this vector.

(1)(c) Use the vectors from (a) and (b) to reconstruct the original vector. Print this vector. Sum the elements of this vector, and round the sum to one decimal place. Print the result.

(2) (4 points) The expression $y = a + bx + cx^2$ is a quadratic function.

(2)(a) Write a function in R that accepts values for a, b, c and x and returns a value for y.

(2)(b) Create a vector, x, of 101 equally-spaced values from -2 to 2, inclusive. **Do not print x.** Pass the following values - a = -1, b = 0 and c = 1 - to your function in (a) and, using the vector x, calculate values for a vector, y. **Do not print y.**

(2)(c) Plot y versus x in color, with x on the horizontal axis. Add a title and other features such as a legend to this plot.

(2)(d) Roots are sometimes referred to as the “zeros” of a function. Confirm the roots of the quadratic function in (b) are -1 and 1 by using coding methods shown in the *Quick Start Guide for R*. This can be done with one line of code. Do not use the traditional formula for finding roots of a quadratic equation, or functions such as *polyroot()*. A graphical display does not have the precision needed for this confirmation and is not sufficient by itself.

(3) (10 points) Use the “trees” dataset for the following items. This dataset has three variables (Girth, Height, Volume) on 31 trees.

(3)(a) Use *data(trees)* to load the file. Check the structure with *str()*. Use *apply()* to return the median values for the three variables in “trees.” Using R and logicals, give the row number and print the three measurements - Girth, Height and Volume - of the tree with Volume equal to the median of Volume.

(3)(b) Girth is defined as the diameter of a tree taken at 4 feet 6 inches from the ground. Convert each diameter to a radius, r. Calculate the cross-sectional area of each tree using pi times the squared radius. Present stem-and-leaf plots of the radius and area.

(3)(c) Use *par(mfrow = c(1, 4))* and present colored boxplots of the radii and areas calculated in (b) along with Height and Volume. Label each accordingly.

(3)(d) Demonstrate that the outlier revealed in the boxplot of Volume is not an extreme outlier. This can be done with one line of code using *boxplot.stats* or logicals.

(3)(e) Plot Volume versus Area in color with Area on the horizontal axis. Add a title. Use *abline()* to present a horizontal line and a vertical line corresponding to the location of the median for Volume and Area respectively.

(4) (3 points) Use matrix operations shown in the “Quick Start Guide” to solve the following system of linear equations. Display the R script and the numerical solutions for x, y and z. Demonstrate your solution is correct by using matrix operations to reproduce the values 1, 1, 3. This can be accomplished with matrix multiplication in one line of code.

$$x - y + z = 1$$

$$x + y - z = 1$$

$$x + y + z = 3$$

(5) (4 points) Use `set.seed(123)` and `rnorm()` with `mean = 0`. Generate two different random samples, each of size `n = 1000`. Designate the first as `x` and use a standard deviation of 2. Designate the second as `y` and use a standard deviation of 1. Do not print `x` and `y`.

(5)(a) Generate a new object using `cbind(x, y)`. Do not print this object. Use `apply()` with this object to compute the sample standard deviation for each column: `x` and `y`. Round to two decimal places. Present this result.

(5)(b) Use `par(mfrow = c(1, 2))` and present two histograms in color with titles, one for `x` and the second for `y`. Maintain comparability of the x-axes with `xlim = c(-6, 6)`.

(5)(c) The two vectors, `x` and `y`, are not correlated. Use the vectors `x` and `y` without sorting or any other manipulation and plot `y` versus `x` to produce a colored, titled scatterplot of their values with `x` on the x-axis. **This is not a multi-colored plot.** Use `cex = 0.5` to control the size of the plotted data points.

Position a legend in the lower left corner which indicates what the rounded sample standard deviations from (a) are for each variable. Check the .html document to make sure the legend does not overlap any data points. It may be necessary to adjust the limits on the `x` and `y` axes to avoid this.