

ISE 5103 Intelligent Data Analytics

Homework #1

Instructor: Charles Nicholson

See course website for due date

Learning objective: Learn the basics of R programming.

Submission notes:

1. Clearly identify each problem (e.g. Problem 1a, Problem 2b, etc.)
2. All *relevant* computer output should be provided unless noted otherwise.
3. The R code itself is part of your solution – make sure to *provide comments* on what your code is doing. Keep it clean and clear!
4. You will submit your complete R script. Note: include `library` commands to load *all* packages that are used in the completion of the assignment. Place these statements at the top of your script.
5. Use “R Markdown” to produce a Word document or PDF as your submission.
6. Do not zip your files for submission. Submit exactly two files. Name the files “LastName-HW1” with the appropriate file extension (that is, .R, .pdf, .docx, or .html)

Note: it is very helpful to create a “New Project” using RStudio for each homework assignment. This allows for easier management of script files and data.

1 Using R: Vectors

- (a) Create a vector with 10 numbers (3, 12, 6, -5, 0, 8, 15, 1, -10, 7) and assign it to `x`.
- (b) Using the commands `seq`, `min`, and `max` with one line of code create a new vector `y` with 10 elements ranging from the minimum value of `x` to the maximum value of `x`.
- (c) Compute the sum, mean, standard deviation, variance, mean absolute deviation, quartiles, and quintiles for `x` and `y`.
- (d) Create a new 7 element vector `z` by using R to randomly sample from `x` *with* replacement.
- (e) Find a package (or packages) that provide the statistical measures skewness and kurtosis. Use the appropriate functions from the package to calculate the skewness and kurtosis of `x`.
- (f) Use `t.test()` to compute a statistical test for differences in means between the vectors `x` and `y`. Are the differences in means significant?
- (g) Sort the vector `x` and re-run the t-test as a paired t-test.
- (h) Create a logical vector that identifies which numbers in `x` are negative.
- (i) Use this logical vector to remove all entries with negative numbers from `x`. (Make sure to overwrite the vector `x` so that the new vector `x` has 8 elements!)

2 Using R: Introductory data exploration

This exercise relates to the College data set, which can be found in the file “College.csv” in D2L. The file contains a number of variables for 777 different universities and colleges in the US. The variables are

- **Private** : Public/private indicator
- **Apps** : Number of applications received
- **Accept** : Number of applicants accepted
- **Enroll** : Number of new students enrolled
- **Top10perc** : New students from top 10
- **Top25perc** : New students from top 25
- **F.Undergrad** : Number of full-time undergraduates
- **P.Undergrad** : Number of part-time undergraduates
- **Outstate** : Out-of-state tuition
- **Room.Board** : Room and board costs
- **Books** : Estimated book costs
- **Personal** : Estimated personal spending
- **PhD** : Percent of faculty with Ph.D.s
- **Terminal** : Percent of faculty with terminal degree
- **S.F.Ratio** : Student/faculty ratio
- **perc.alumni** : Percent of alumni who donate
- **Expend** : Instructional expenditure per student
- **Grad.Rate** : Graduation rate

Before reading the data into R, it can be viewed in Excel or a text editor.

- (a) Use the `read.csv()` function to read the data into a data frame in R. Call the data frame `college`. Make sure that you have the directory set to the correct location for the data (or that the data is in the same directory as the RStudio project).
- (b) Look at the data using RStudio. You should notice that the first column is just the name of each university. We don't really want R to treat this as data. However, it may be handy to have these names for later. Try the following commands:

```
rownames (college) <- college [,1]
View (college )
```

You should see that there is now a `rownames` column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored. Try

```
college <- college [,-1]
```

and then view the data (either with the `View` command or clicking on the `college` data frame in the RStudio workspace window) Now you should see that the first data column is **Private**.

- (c)
 - i. Use the `summary()` function to produce a numerical summary of the variables in the data set.
 - ii. Access help for the `pairs` function and then use `pairs` to produce a scatterplot matrix of the *first ten columns*. Recall that you can reference the first ten columns of a matrix `A` using `A[,1:10]`.
 - iii. Use the `plot()` function to produce side-by-side boxplots of Outstate versus Private. Label the axes and main title appropriately.

- iv. Using the following bit of code you will create a new qualitative variable, called `Elite` by binning the `Top10perc` variable. That is, `Elite` will classify the universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%.

Add comments to each line below explaining what the corresponding code is doing and then run the code.

```
Elite <- rep ("No", nrow(college ))
Elite [college$Top10perc >50] <- "Yes"
Elite <- as.factor (Elite)
college <- data.frame(college ,Elite)
```

- v. Use the `summary()` function to see how many elite universities there are.
- vi. Now use the `plot()` function to produce side-by-side boxplots of `Outstate` versus `Elite`. Label the axes and main title appropriately.
- vii. Use the `hist()` function to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the command `par(mfrow=c(2,2))` useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

3 Using R: Manipulating data in data frames

- (a) Load the data frame `baseball` in the `plyr` package. Use `?baseball` to get information about the data set and definitions for the variables.
- (b) You will calculate the *on base percentage* for each player, but first clean up the data:
- Before 1954, sacrifice flies were counted as part of sacrifice hits, so for players before 1954, sacrifice flies (i.e. the variable `sf`) should be set to 0.
 - *Hit by pitch* (the variable `hbp`) is often missing – set these missings to 0.
 - Exclude all player records with fewer than 50 *at bats* (the variable `ab`).
- (c) Compute *on base percentage* in the variable `obp` according to the formula:

$$\text{obp} = \frac{\text{h} + \text{bb} + \text{hbp}}{\text{ab} + \text{bb} + \text{hbp} + \text{sf}}$$

- (d) Sort the data based on the computed `obp` and print the year, player name, and on base percentage for the top five records based on this value.

4 Using R: aggregate() function

The `aggregate` function is very useful method in R and allows you to easily compute statistics (such as the mean) for different groupings, e.g. if you have a set of data for students which contains both demographic and grade information; to compute the mean class grade *by* gender, you could use the `aggregate` command.

To complete this problem, you will need to look up information on how to use `aggregate`. You can use the built-in R documentation, look for help online, or both.

- (a) Load the `quakes` data from the `datasets` package.
- (b) Plot the recorded earthquake magnitude against the earthquake depth using the `plot` command.
- (c) Use `aggregate` to compute the average earthquake depth for each magnitude level. Store these results in a new data frame named `quakeAvgDepth`.
- (d) Rename the variables in `quakeAvgDepth` to something meaningful.
- (e) Plot the magnitude vs. the average depth.
- (f) From the two plots, do you think there is a relationship between earthquake depth and magnitude?