ISTE 780 HOMEWORK INSTRUCTIONS
MICK MCQUAID
SUMMER 2019

INTRO

Part of your homework grade will depend on following these instructions carefully. These instructions serve two purposes. First, they make it easier for the instructor to concentrate on what you did right or wrong by removing variables like formatting. Second, they give you practice using a simple format for R reporting.

INSTRUCTIONS

Text Files. For each homework assignment, you will turn in one or more files. You will always turn in a text file and sometimes you will turn in one or more graphics files. Here's how to format the text files.

0. Do not zip or otherwise compress any files you turn in.

1. Use a full-featured text editor that saves your files as UTF-8.

2. Save the files with the extension `.r` and make the file name match the roman numeral of the assignment but in lower case. For example, the first homework assignment will be turned in as a file called `i.r`. The second assignment will be called `ii.r` and so on. Do not name them 'II.R' or similar variations or I will subtract points from your score.

3. There are three kinds of information in each file: homework instructions, your R code, and your comments. These are to be marked up as follows. Put one # symbol followed by a space in front of every line of homework instructions, two # symbols followed by a space in front of every line of

your comments, and make every line of your R code flush left with no # symbol.

4. The first four lines of the file contain the following info: the homework number from the textbook, your name, and the semester. That covers the first three lines. The fourth line should be blank. For example, for the first homework, my file would begin

```
## Exercise 2.4.8
## Mick McQuaid
## Summer 2016
```

Notice that each line begins with two # symbols followed by a space.

5. Cut and paste the homework specification from the textbook into the file. For the first homework assignment, I would cut and paste the following text from pages 54 and 55.

```
# 8. This exercise relates to the College data set,
#    which can be found in the file College.csv. It
#    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 % of high school
# • Top25perc : New students from top 25 % of high school
# • 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
#     R. Call the loaded data college. Make sure that you
#     have the directory set to the correct location for
#     data.
# (b) Look at the data using the fix() function. You
#     should notice that the first column is just the nam
#     each university. We don't really want R to treat th
#     as data. However, it may be handy to have these nam
#     for later. Try the following commands:
# (c)
#     > rownames(college)=college[,1]
#     > fix(college)
#     You should see that there is now a row.names column
#     with the name of each university recorded. This mea
#     that R has given each row a name corresponding to t
#     appropriate university. R will not try to perform
#     calculations on the row names. However, we still ne
#     to eliminate the first column in the data where the
#     names are stored. Try
#     > college=college[,-1]
#     > fix(college)
#     Now you should see that the first data column is
#     Private. Note that another column labeled row.names
#     appears before the Private column. However, this is
```

```
#     a data column but rather the name that R is giving
#     each row.
# i. Use the summary() function to produce a numerical
#    summary of the variables in the data set.
# ii. Use the pairs() function to produce a scatterplot
#     matrix of the first ten columns or variables of the
#     data. 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.
# iv. Create a new qualitative variable, called Elite, by
#     binning the Top10perc variable. We are going to div
#     universities into two groups based on whether or no
#     the proportion of students coming from the top 10%
#     their high school classes exceeds 50 %.
#     > Elite=rep("No",nrow(college))
#     > Elite[college$Top10perc >50]="Yes"
#     > Elite=as.factor(Elite)
#     > college=data.frame(college ,Elite)
#     Use the summary() function to see how many elite
#     univer- sities there are. Now use the plot() functi
#     to produce side-by-side boxplots of Outstate versus
#     Elite.
# v. Use the hist() function to produce some histograms
#    with differing numbers of bins for a few of the
#    quantitative vari- ables. 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 m
#    simultaneously. Modifying the arguments to this
#    function will divide the screen in other ways.
# vi. Continue exploring the data, and provide a brief
#     summary of what you discover.
```

Obviously, this is a lot of text for some of the problems but it allows you to see immediately what questions you are answering.

Notice that I have added a # symbol in front of every row. Any full-featured text editor should allow you to do this with a single command, rather than having to manually enter these in front of every line. For example, with Vim, the text editor I am using, I said `:35,113s/^/# /` to substitute # (a hash followed by a space) for the beginning of the line (symbolized by ^) on lines 35 through 113. The letter `s` is an abbreviation for *substitute*.

6. Interleave your answers with the problem specification, leaving a blank line between your text and the problem specification text. Leave a blank line between each paragraph of your writing.

7. Whenever your answer includes R code, write the code flush left without any preceding # symbol. For example

```
pairs(Auto)
plot(mpg,cylinders)
sapply(Auto[,3:7],mean)
```

8. If you want to include mathematical expressions, you may write them in LaTeX and surround them with dollar signs. This will allow you to say things like $\widehat{\mu}_0$. If you cannot understand LaTeX (it will be demonstrated for you and examples will be given), you can write the expressions out phonetically. For example, the above expression is pronounced *mu hat nought* and is written in LaTeX as `$\widehat\mu_0`

9. Feel free to use markdown characters within your comments. If you do not know markdown, don't worry about this.

10. For the most part, you do not need to include the output of your R code. An R file can be processed to automatically run all R expressions in it so I can tell easily whether your code produces correct results.

11. One exception to the above rule is that the homework specification often calls upon you to comment on your findings or justify your findings. You will frequently cite statistics from your output.

Graphics files. The homework specification often calls upon you to produce graphics. I may even add more graphics requirements to some of the assignments. For really basic graphics such as those in the first assignment, I will be content with the R code that produces the graphics. In more advanced cases, I will require you to submit graphics files with your output.

1. Each graphic should be saved in a pdf file produced by R.

2. Each file should be named with the basename of the homework followed by a lowercase Latin letter, a–z. (You'll never have 26 graphics in this course!) For example, if homework IV required three graphics, they would be named `iva.pdf`, `ivb.pdf`, and `ivc.pdf`. They do not need to be identified with your name because myCourses automatically adds your name, student id, and a bunch of other characters to every filename you submit. You won't see all these characters but I have a different view of the dropboxes than you have.

3. You may refer to the graphics inside your `.r` file using the following convention. This will allow post-processing software to add the graphics to your homework. The following must appear after a blank line. Post-processing software will add the graphic in `iva.pdf` to an output file and label the graphic with `description`. Obviously, you should not use the literal word *description* but instead a short description of the contents of the file. Here is how to name the graphic in your `.r` file:

```
## ![description](iva.pdf)
```

HW X. DISCUSSION OF A SIGKDD ARTICLE

The tenth homework assignment differs from all the others. For this assignment, visit the Discussions section of MyCourses and provide a reflection on the SigKDD article assigned in the syllabus. This reflection should include a summary of the paper and your perspective on the strong and weak points of the paper. You should also read the reflections of other students but yours should be independent of theirs. In other words, write as if yours was the only one. Don't make the mistake of saying "Others have already summarized the article, so I'll just briefly add …". That would be harmful to your grade.

PROJECT

Project Overview. For the project, you will work in teams of two students on a problem of your choosing that is interesting, significant, and relevant to developing data analytics algorithms for knowledge extraction from non-trivial dataset with a reasonable size. You will have great latitude in what you choose to work on, so take advantage of this opportunity to make a big impact! This may form the basis for your capstone, so plan accordingly.

The primary requirements of the project are:

- Your project must use some non-trivial data that your team collects. Here are some widely known data repositories:
  - Open Government Data: https://www.data.gov, https://www.data.gov.uk, https://www.data.gouv.fr
  - Kaggle: https://www.kaggle.com
  - KDD Nuggets: https://www.kdnuggets.com/datasets/
  - UCI Machine Learning Repository: https://archive.ics.uci.edu/ml
  - StatDat: https://lib.stat.cmu.edu/StatDat

  - TwitteR: https://cran.r-project.org/web/packages/twitteR/index.html
  - rfigshare: https://figshare.com, https://cran.r-project.org/web/packages/rfigshare/index.html

- Your project must perform statistical learning algorithms to analyze the data.
- Choose one algorithm as your core algorithm and include four other algorithms for performance comparison. The core algorithm must be studied more closely and fine-tuned for better performance.
- Include at least one algorithm that has NOT been covered in the class. The new algorithm can be used as either the core algorithms or the comparing algorithms.

Grading Criteria. The course project counts for twenty-five percent of your final grade. Your project will be graded based on five key milestones, each weighted as five percent of your final course grade.

Milestone 1: Project proposal. 1–2 page PDF

Each group should post a 1-2 page project proposal as `m1.pdf` to Assignments on myCourses. Provide a brief, descriptive name of your project. Your name should be something memorable! In the proposal, you should address the following issues:

- Description of the problem.
- What is exactly the function of your system? That is, what will it do? (Supervised or Unsupervised? Regression or classification?)
- Why would we need such a tool and who would you expect to use it and benefit from it?
- You should mention the data and the candidate core algorithms that you will use.

Milestone 2: Data summary/visualization. 3–5 page PDF

For the second project milestone, you must have collected your dataset that your project will ultimately use. You should summarize the major characteristics of the data and use different ways to visualize the data. Data summary and visualizations should be presented in captioned tables and figures. Post your milestone 2 report as `m2.pdf` to Assignments on myCourses.

Milestone 3: Algorithm Testing. 2 page PDF

You should test at least three candidate algorithms and report the result along with your detailed analysis. Post your milestone 3 report as `m3.pdf` to Assignments on myCourses.

Milestone 4: Core Algorithm Fine Tuning. 2 page PDF

For the fourth project milestone, you must have chosen your core algorithm and included all four other algorithms for comparison. The core algorithm should be fine tuned to improve its performance. It is important to provide a detailed analysis about the performance of the core algorithm, especially what fine-tuning steps are conducted and whether/how they improve the core algorithm. Detailed comparison with other algorithms should also be provided. Post your milestone 4 report as `m4.pdf` to Assignments on myCourses.

Milestone 5: Final Project Presentation. online

Project presentation online using 20 to 25 slides.

- Each project presentation should be on Google Slides or an equivalent platform as a self-running slideshow
- Post the URL in Assignments on myCourses
- Presentations on projects should include motivation, problem definition, key issues and alternative ways of resolving those, related work and their limitations, your approach, validation, conclusions (key contributions), and future work (assumptions and potential extensions).
- You are expected to visit office hours of the instructor and meet with your peers to insure timely completion of each step.