Problem Set 2: Basic Statistics

Introduction

In this Problem Set, we will practice doing basic statistical analyses and visualizations in R with a health dataset. Download the dataset associated with this assignment, "Session5PimaDiabetes.csv" from the Problem Set 2 folder on GitHub. Diabetes incidence and risk is extremely high among those with Pima Indian heritage from the Southwestern US. This dataset contains data for 768 women over the age of 21 with 8 continuous variables related to Diabetes risk along with a binary response variable called Diabetes. You will submit a short written report and accompanying R script file for this problem set. It is good practice to document your R code as you run it, so be sure to save code that you run into your R script file as you go along. Someone else running your R script file should be able to reproduce your results exactly.

Instructions

Follow the instructions below. Be sure to include the answers to any questions posed in your report. For the purpose of reproducibility, start your R analysis by making a new R script file to which we can save our code. It's recommended that you write commands into your script file and then run your script file, rather than typing commands into the R console and then copy/pasting them into your script, for the sake of reproducibility. When we evaluate and grade your R script, we should be able to run it and arrive at the same answers as you did, so make sure that you check that your R script is able to do so from a clean environment (i.e. starting from scratch, with no data loaded into your R environment).

Start your script file off by setting your seed to the number 1389 (or any other number) with the following command, so our analyses can be more easily reproduced:

```
set.seed(1389)
```

Then, we'll start, as usual, by loading the libraries that we'll want to use. Looking ahead in this assignment, we'll need at least the packages `e1071`, `MASS`, `car`, and `mvoutlier`. If you don't have these packages, you'll need to install them.

```
library("e1071")
library("MASS")
library("car")
library("mvoutlier")
```

If you want to use the `tidyverse` package or any other packages, load them here as well. We will not be using `tidyverse` in the following example code, but feel free to if you feel comfortable doing so.

Next, we'll need code that loads in our data. Remember that you can use the function `file.choose()` in place of a file location to pull up an interactive prompt to help you find your files. Here, we demonstrate this assuming that the data file is located in our current working directory—your data file is most likely not located in your current working directory unless you've moved it there or changed your working directory to the folding containing your data file.

```
# replace "Session5PimaDiabetes.csv", including the quotes, with file.choose() if desired.
dat <- read.csv("Session5PimaDiabetes.csv", header = TRUE)
head(dat) # look at the first few rows to check
```

Data Quality Control

First, assess the missing data for all samples and variables. If you don't remember how to do this, make sure you've seen the lecture material for week 5 and reviewed the accompanying R code. While you're reviewing the R code accompanying the lecture, we recommend that you run it side-by-side with the lecture material to get a better feel for it. You may scavenge bits of code from the lecture code to use in this assigment.

Discard (and document) any variables with greater than 40% missing data. Use multivariate imputation using the `mice` package to impute the remaining missing data points. **Answer this question and following questions posed in this assignment in a separate document. During imputation, do you include the response variable or not?**

With this new imputed data, assess each individual independent (predictor) variable for skewness and kurtosis of each independent variable using the `e1071` package. Note that the Kurtosis function in `e1017` has a normality expectation of 0 instead of 3. **Report any variable with a skewness less than -1 or greater than 1.**

For any variable with excess skewness (in this case any value between -1 & 1 is acceptable), anchor the variable to 1 and transform with the BoxCox. Use the `boxcox` function from the `MASS` packages and the `bcPower` function from the the `car` package. **Report the "optimal" lambda for each variable that needs transformation.**

Advanced (optional, extra credit)

Use the `uni.plot` function (with it's default setting) in the `mvoutlier` package to assess multivariate outliers. Report how many outliers it suggests there are.

Report which ones (use the row numbers as ID's for each sample). Report which sample is the most extreme outlier (hint: look at the mahalanobis distances in the output object)?

Data Source

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C. and Johannes, R. S. (1988) Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications in Medical Care (Washington, 1988), ed. R. A. Greenes, pp. 261–265. Los Alamitos, CA: IEEE Computer Society Press.

The report

Develop a report (I recommend a Word (or other text editor) document) for your problem set that includes answers to all of the questions posed above, showing plots where appropriate. If you're using RStudio, you can use the "Export" button to easily save plots as image files. If you're working in base R or prefer to turn your plots into image files using code, Datamentor has a short but helpful online guide on how to do so.

Save your report as a PDF file and submit your report through the course 2GW site. Clean up your code (your R script) and submit it as a supplementary file, along with your main report.

Due date

Day 7, Week 5