## Owen Galvin

## HU Extension Assignment 01 E63 Big Data Analytics

### Handed out: 01/30/2016 Due by 11:59PM on Friday, 02/05/2016

**Download and install the latest version of R and R Studio.**

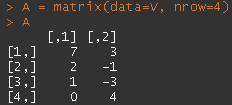
**Problem 1.** Create a vector V with 8 elements (7,2,1,0,3,-1,-3,4).

**Solution: Use c aka concatenate command**



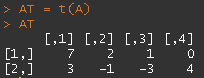
* Transform that vector into a rectangular matrix A of dimensions 4X2 (4- rows, 2-columns).

**S: Use matrix function to create a matrix out of vector V w/4 rows, which also has the result of 2 columns**



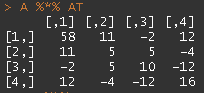
* Create a matrix transpose to the above matrix A. Call that matrix AT.

**S: Use t aka transpose function**



* Calculate matrix products: A\*AT and AT\*A. Present the results. What are the dimensions of those two product matrices.

**S: A\*AT leads to 4X4 (4 row, 4 column) matrix**



**AT\*A leads to 2X2 (2 row, 2 column) matrix**



* Square matrixes sometimes have an inverse matrix. Try calculating inverse matrices (or matrixes, if you prefer) of above matrices (matrixes) A\*AT and AT\*A.

**S: Use solve function to calculate inverse matrix for A\*AT, which fails**

****

**And then same for inverse matrix of AT\*A, which presents a solution**



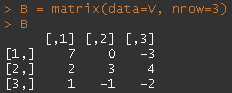
* Extend the above vector V with the ninth number of value -2. Do it elegantly by concatenating two vectors (☺).

**S: use c() to concatenate V and -2**



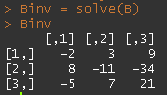
* Transform that extended vector into a 3X3 matrix B.

**S: create matrix based on V, use nrow=3, which will lead to a 3X3 matrix. Leaving default byrow=FALSE in place means the numbers go down 3 and repeat the same in 2nd & 3rd columns.**



* Calculate the inverse matrix of matrix B. Call it Binv. Demonstrate that the product of B and Binv is the same as the product of Binv and B and is equal to what?

**S: use solve function to calculate inverse**

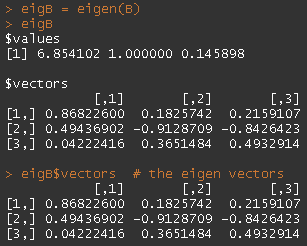


**Demonstrate equality of B \* Binv & Binv \* B, where snippet on the left makes it look like they are close to equal. Code on right shows they are “nearly equal” by R standards but not technically identical due to errors inherent in floating point calculations. Values in each matrix can be rounded, and even out to 10 places, those previous very small values evaluate to zero. The result is displayed on the right and in linear algebra a matrix with values like this (all zeros except for the diagonal ones) is known as the identity matrix.**

|  |  |
| --- | --- |
|  |  |

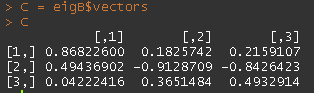
* Determine the eigenvectors of matrixes B.

**S: create variable named eigB to hold the results of calling eigen function on B. Access the eigenvectors via $vectors member of that variable.**

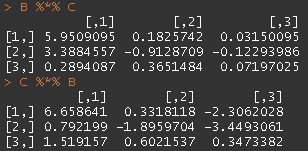


* Construct a new matrix C which is made by using each eigenvector of matrix B as a column. Calculate the product of matrix C and matrix B and the product of matrix B and C. Is there any significance to the elements of the product matrixes.

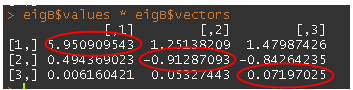
**S: Here is where it gets confusing. On the Discussion board, the professor notes that each eigen vector is a 1x3 matrix, but the R documentation for eigen function states that the $vectors holds “*a p \* p matrix whose columns contain the eigenvectors of x*”, implying that each is a 3x1 (3 rows, 1 column) matrix. I’m going to go with assumption both statements describe the same thing, where the $vectors holds 3 columnar eigen vectors and transposition is not necessary.**



**Simply do matrix multiplication on C & B and then again on B & C and display the output.**

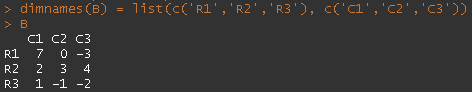


**I’ve spent quite a bit of time trying to find significance in the products (and re-running the calculations to make sure I didn’t miss anything) but haven’t found much. I did eventually realize the top-right to bottom left diagonal in B & C product is the same as the simple product of the eigen values and eigen vectors from the B matrix but I have little knowledge of linear algebra and don’t know if this is significant or a simple side effect of any calculations like these.**



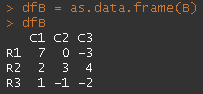
* Transform matrix B into a matrix with names columns and named rows.

**S: create very simple row and column names by assigning a list of character vectors to dimnames(B) function**



* Transformed that fully “named” matrix into a data.frame.

**S: use the as.data.frame function to coerce B into a data frame. Simply going by output of the new dfB, appears essentially unchanged.**



* Ask the object you just created what is its class().

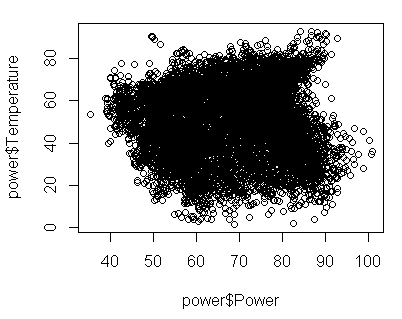
**S: calling class() on dfB reveals its class to be that of data.frame**



**Problem 2**. Consider file 2006Data.csv upload to the class site in Assignment 01 folder. File represents actual measurement of power consumption in a country somewhere in a California. Import data contained in that file into a data frame. You are expected to Google and find a function that will let you perform that import. Create a scatter plot of power consumption vs. temperature and power consumption vs. hour of the day. Subsequently create a boxplot with power on the vertical axis and hour of the day on the horizontal axis. The objective is to present the distribution (variation) of power consumption for every hour of the day.

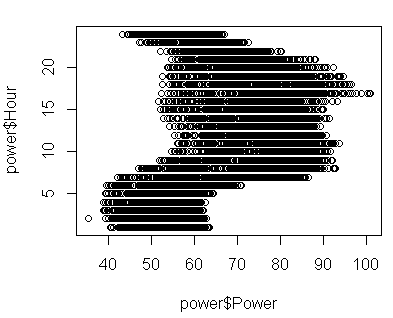
**S: Import the data into data.frame variable named power. Create a scatterplot of power consumption (presumably the Power variable) against temperature. Resulting plot is too dense to make much sense out of…don’t see any kind of linear relationship that is immediately visible.**





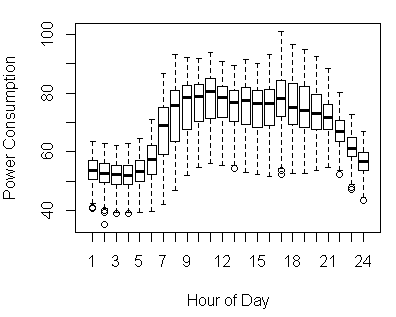
**Now plot the power consumption against hour of the day, which reveals a more information rich plot, showing that generally speaking more power is consumed around Hour 17, i.e. 5pm. Lowest levels of power consumption occur overnight.**





**Finally, boxplot with power on the vertical axis and hour of the day on the horizontal axis**





**Problem 3.** Separate temperature scale in a reasonable number of intervals: 50 or 100. Calculate average power consumption, minimum power consumption and maximum power consumptions for every interval. Present those three sets of values on a single scatter graph (perhaps in different colours). Calculate three covariance matrixes between temperature and each of those power indicators (min, average, max).

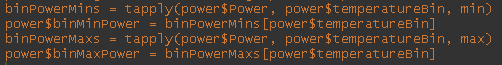
**S: Create a new variable called temperatureBin that will reference which one of the 50 intervals into which the temperature value falls:**



**Calculate the average power consumption in each of those 50 bins using tapply and assign the averages to each member of a given bin by matching the bin label, still in "(a,b]" format, to same label in the binPowerAvgs data frame. Basically a dictionary lookup to perform the assignations.**



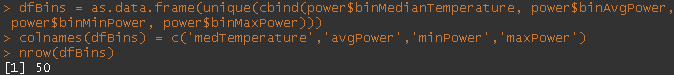
**Continue applying group aggregate values back to the original power data frame, as before. The next two steps involve following the same procedure but for min and max values within each of those bins:**



**And one more for the median temperature within each bin (my original approach used $temperatureBin for plotting, which created plot points just like the graph later on, but the median temp will come in handy for the covariance matrix).**



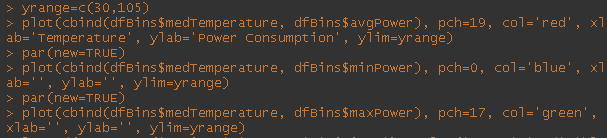
**Now it comes time to pull out the data for plotting and put it into a new dataframe named dfBins. The unique() function is used on the four columns we want and since each of these values repeat within each bin, we will wind up with 50 rows, matching the original 50 breaks. And now is also when I become fairly certain there was a more straightforward way of assembling the data present in the dfBins dataframe. After reviewing code & reviewing the data in Excel after exporting though, everything looks to be working fine. Also use colnames() function to add in description column names.**



**Plot each of the three sets (temperatureBin vs. binAvg/binMin/binMax) on the same scatter plot. First check the min and max values in Power variable.**



**From there determine that a set range between 30 and 105 would be appropriate for use with each of the three plots, create a yrange variable and assign to the ylim parameter during plot() so that avg/min/max values share the same numeric range on the y axis. Create each with a different color and symbol. Set par such that each subsequent graph is drawn upon the existing graph space. Use xlab/ylab params to add labels to x/y axes. I’m going to presume the plot presentation here can wait until everything has been drawn, as opposed to a new screenshot for each plot piece.**

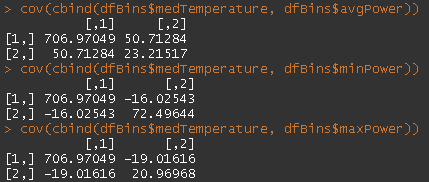


**Finally create a legend, using same colors and symbols as were used for each plot.**



|  |
| --- |
|  |

**Use cov() function on each combination of the median temperature and avg/min/max values in order to produce three covariance matrices.**



**SUBMISSION INSTRUCTIONS:**

Your main submission should be an MS Word document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you just copy your code and results from the R console and past them into the Word document. Start with this text of homework assignment as the template. Please add any other files that you might have used or generated.

Please read detailed submission and grading instructions on class site.