Homework 2

Statistical Computing, 36-350

Due Wednesday Sept 16, 2015

Your homework must be submitted in R Markdown format. We will not (indeed, cannot) grade homeworks in other formats. Your responses must be supported by both textual explanations and the code you generate to produce your result. (Just examining your various objects in the "Environment" section of R Studio is insufficient—you must use scripted commands.)

1. **Decathlon with Superheroes.** In your R console, run the following line: `install.packages('ade4')` in order to install the package `ade4` (you only have to do this once). Then, the following code:

```
library(ade4)
data(olympic)
```

will load an object called `olympic` into your current R workspace, containing data about records of 33 athletes in the 10 events of a decathlon: 100 meters (100), long jump (long), shotput (poid), high jump (haut), 400 meters (400), 110-meter hurdles (110), discus throw (disq), pole vault (perc), javelin (jave) and 1500 meters (1500).

   a. `olympic` is a list. How many objects does it hold? What are the types and names of these objects?
   b. Take the first object in `olympic` and copy it into a new object called `olympicmat`. Cast it into a matrix, then back to a data frame. Did anything change?
   c. Change the names of the `olympicmat` into something more human-readable (although in general, usage of succinct variable names is good practice): replace the column names with their longer versions shown above (e.g., `100` into `100 meters`). Show the first 10 rows after doing so.
   d. Suppose we have found out that the first three contestants are superheros. Replace *just* these names with ironman, wolverine and hulk. Again, show the first 10 rows after doing so.
   e. Now add a new datapoint to `olympicmat`, by appending the row shown below. Make sure to change the row name too. Show the last ten rows after doing so. Is thor an extraordinary discus thrower? Produce a plot of your choice to justify.

```
thor = c(8.52, 10.31, 16.28, 4.51, 30.12, 13.62, 50.5, 10.1, 100.24, 200.12)
```

   f. Add the above changes back to `olympic`, by assigning your `olypicmat` back into the first object of `olympic`. Make sure you refer to objects of a list by its name (e.g., `mylist[["mykey"]]`), and not the index (e.g., `mylist[[3]]`).
   g. Now we will add a few objects to the list `olympic`. Add `year` and `sporttype` to the list in that order, with those same names.

```
year = 1998
sporttype = "decathlon"
```

2. **Fun with Linear Regression.** We will get more practice with matrix and vector data types, and also take a glimpse at linear regression. When asked to plot, always label and title the plots clearly.

   a. First generate a matrix as follows (the seed is to ensure we all get the same `X1`).

```
onevec = rep(1,10)
set.seed(0)
X1 = rnorm(10,2.5,0.3)
X = cbind(X1, onevec)
```

b. The "hat" matrix for linear regression onto the matrix X is defined by $X(X^TX)^{-1}X^T$. Using functions for matrices like `%*%`, `t()` and `solve()`, calculate this matrix and store it as `P`.

c. Compute the eigendecomposition of `P` (see example in lecture notes). Write a one line command to produce how many eigenvalues are larger than 1e-3. (Some insight: this is one way to find the `rank` of the matrix X.)

d. Now right-multiply `P` by the vector `y` given below, and assign the resulting numeric vector to an object named `yhat`.

```
y = c( 9.01, 8.39, 8.86, 11.2,  9.2, 6.29, 8.15, 8.97, 9.07, 10.4)
```

e. Congratulations, you have computed your first linear regression! `yhat` is the best linear model prediction (of the form $y = aX_1 + b$) you could have made, using your *predictor matrix* `X`. Now, calculate $(X^TX)^{-1}X^Ty$, and cast it as a numeric vector called `lincoef`.

f. These are your regression coefficients (a, b)! A large positive value of `a` indicates a positive correlation between `X1` and `y` (You will learn more about this in later courses). You may also use it to make future predictions at unobserved, new values of X_1.

g. Create two scatter plots, one of `yhat` and `y`, and one of `y` and `X1`. Do entries in `yhat` and `y` closely follow? Do you see a linear trend in the latter plot?

h. Use `abline(lincoef[2], lincoef[1], col = 'red')` in the latter plot to see *your* regression model (and pat yourself on the back; a linear regression seems like an excellent choice).

i. Lastly, gather all this information (`P`,`X`,`y`,`lincoef`, and `yhat`) into a list object called `linregoutput`. Show this object.