Homework 1

STAT6306; Due: 09/05/2017

Problem 0

R is a standard software interface for computing and graphics and Rstudio is an integrated development environment (IDE) for R. Install both on your computer.

- R: http://lib.stat.cmu.edu/R/CRAN/
- Rstudio: https://www.rstudio.com/products/rstudio/#Desktop

Problem 1

Suppose we have the following matrix:

```
set.seed(1)
A = matrix(rnorm(4*3),nrow=4,ncol=3)
```

We want to get the column mean for each column of the matrix A. Do this using each of the following techniques:

Part a

Hard coding (that is, write $(A[1,1] + A[2,1] + ...)/4, ...$)

```
#SOLUTION
```

Part b

For loop(s)

```
#SOLUTION
```

Part c

The apply (or related) function

```
#SOLUTION
```

Problem 2

Many statistical methods can be computed/analyzed using the SVD[1]. Let's look at solving least squares problems as they are fundamental to modern data analysis.

[1] For this question, I'm using common linear algebra notation of A, x, and b. This x is not to be confused with a feature

Part a

```
set.seed(10)
A = matrix(rnorm(24),nrow=6,ncol=4)
A[,1] = 1
```

Write $A = UDV^\top$ (that is, form svd.out = svd(A)).

Suppose we wish to solve for $\hat{x} = arg\min_x ||Ax - b||_2^2 = (A^\top A)^{-1} A^\top b$ for $b = (1, 2, 3, 4, 5, 6)^\top$. As an aside, to show this, note that[2]

$$
\begin{aligned}
||Ax - b||_2^2 &= x^\top A^\top Ax + b^\top b - 2x^\top A^\top b && (1)\\
&\Rightarrow \nabla_x = 2A^\top A\hat{x} - 2A^\top b \stackrel{\text{set}}{=} 0 && (2)\\
&\Rightarrow \hat{x} = (A^\top A)^{-1} A^\top b && (3)
\end{aligned}
$$

How can I solve this using the SVD? Here, let's follow the steps:

1. Form $U^\top b$
2. Solve $Dw = U^\top b$
3. Form $\hat{x} = Vw$

Produce this $\hat{x}$ in R via this method. Note that in this particular case, all the singular values in D are nonzero and hence $\hat{x} = VD^{-1}U^\top b$.

```
#SOLUTION
```

Part b

Suppose instead we have observations under the model $Y = \mathbb{X}\beta + \epsilon$, where $Y = b$ and $\mathbb{X} = A$. Using the R function lm and predict, what is the least squares solution $\hat{\beta}$ and the fitted values $\hat{Y}$ for Y using the least squares solution?[3]

How does the produced coefficient vector $\hat{\beta}$ compare the $\hat{x}$?

```
#SOLUTION
```

Problem 3

Now, let's look at a new A

```
set.seed(100)
A = matrix(rnorm(4*3),ncol=4,nrow=3)
A[,1] = 1
```

and $b = (1, 2, 3)^\top$. This is an example of an *underdetermined* system.

[2] ∇_x indicates gradient with respect to x

[3] Remember to not have R add an intercept as there is already a column of ones

Part a

What do(es) the corresponding $\hat{x}$ look like using the SVD? What do(es) the $\hat{\beta}$ look like using lm?

```
#SOLUTION
```

Part b

What do(es) the corresponding $A\hat{x}$ look like using the SVD? What do(es) the $\hat{Y} = \mathbb{X}\hat{\beta}$ look like using predict?

```
#SOLUTION
```

NOTE: it is worth considering why the two objects have different formatting.

Part c

Though this is just one simulated example and not a proof, your findings generalize to all situations when $p > n$. Summarize in words what these findings are.

SOLUTION:

Problem 4

```
set.seed(1)
n = 2000
p = 500
X = matrix(rnorm(n*p),nrow=n,ncol=p)
X[,1] = 1
format(object.size(X),units='auto')#memory used by X
```

```
## [1] "7.6 Mb"
```

```
b = rep(0,p)
b[1:5] = 25
b_0 = 0
Xdf = data.frame(X)
Y = b_0 + X %*% b + rnorm(n)
hatBeta = coef(lm(Y~X-1)) #Here, the [-1] ignores the intercept

#Using out-of-core technique
write.table(X[1:500,],file='Xchunk1.txt',sep=',',row.names=F,col.names=names(Xdf))
write.table(X[501:1000,],file='Xchunk2.txt',sep=',',row.names=F,col.names=names(Xdf))
write.table(X[1001:1500,],file='Xchunk3.txt',sep=',',row.names=F,col.names=names(Xdf))
write.table(X[1501:2000,],file='Xchunk4.txt',sep=',',row.names=F,col.names=names(Xdf))
write.table(Y[1:500],file='Ychunk1.txt',sep=',',row.names=F,col.names=F)
write.table(Y[501:1000],file='Ychunk2.txt',sep=',',row.names=F,col.names=F)
write.table(Y[1001:1500],file='Ychunk3.txt',sep=',',row.names=F,col.names=F)
write.table(Y[1501:2000],file='Ychunk4.txt',sep=',',row.names=F,col.names=F)
```

Part a

Report the first 5 entries in $\hat{\beta}$ (that is, hatBeta in the above code) using lm on all the data simultaneously

```r
#SOLUTION
```

Part b

Alternatively, we can read in each chunk and update the solution using biglm. Here is the first part. Complete the procedure in the natural way on the remaining chunks. Compare the first 5 entries in $\hat{\beta}$ formed by this method with the entries in (a)

```r
if(!require(biglm,quietly=TRUE)){
  install.packages('biglm',repos='http://cran.us.r-project.org');require(biglm)
}

# Chunk 1
Xchunk = read.table(file='Xchunk1.txt',sep=',',header=T)
Ychunk = scan(file='Ychunk1.txt',sep=',')
form = as.formula(paste('Ychunk ~ -1 + ',paste(names(Xchunk),collapse=' + '),collapse=''))
out.biglm = biglm(formula = form,data=Xchunk)
hatBeta[1:5]
```

```
##       X1       X2       X3       X4       X5
## 24.99160 24.98724 24.96734 25.05268 25.04096
```

```r
coef(out.biglm)[1:5]
```

```
##       X1       X2       X3       X4       X5
## 25.08815 24.48229 26.09057 26.84305 24.64633
```

```r
# Chunk 2
Xchunk = read.table(file='Xchunk2.txt',sep=',',header=T)
Ychunk = scan(file='Ychunk2.txt',sep=',')
out.biglm = update(out.biglm,moredata=Xchunk)
hatBeta[1:5]
```

```
##       X1       X2       X3       X4       X5
## 24.99160 24.98724 24.96734 25.05268 25.04096
```

```r
coef(out.biglm)[1:5]
```

```
##       X1       X2       X3       X4       X5
## 24.96665 25.00382 24.97887 25.01876 25.08741
```

```r
# Chunk 3
Xchunk = read.table(file='Xchunk3.txt',sep=',',header=T)
Ychunk = scan(file='Ychunk3.txt',sep=',')
out.biglm = update(out.biglm,moredata=Xchunk)
hatBeta[1:5]
```

```
##       X1       X2       X3       X4       X5
## 24.99160 24.98724 24.96734 25.05268 25.04096
```

```r
coef(out.biglm)[1:5]
```

```
##       X1       X2       X3       X4       X5
## 24.98077 25.00492 24.97527 25.02329 25.08050
```

```r
## Can you figure out the final steps? Have we updated on all of the chunks?
```

```r
print(hatBeta[1:5])
```

```
##       X1       X2       X3       X4       X5
## 24.99160 24.98724 24.96734 25.05268 25.04096
```

```r
print(coef(out.biglm)[1:5])
```

```
##       X1       X2       X3       X4       X5
## 24.98077 25.00492 24.97527 25.02329 25.08050
```

Problem 5

Forward selection.

Part a

Using the $\mathbb{X}$ and Y generated in the previous problem, use forward selection and AIC to estimate b.

```r
if(!require(leaps)){install.packages('leaps',repos='http://cran.us.r-project.org');require(leaps)}
```

```
## Loading required package: leaps
## Warning: package 'leaps' was built under R version 3.3.2
```

Part b (optional)

Save the $\mathbb{X}$ generated in the previous problem to a .csv file. Using forward selection and AIC, estimate b without having $\mathbb{X}$ stored in memory. Verify that your answer matches (a)

```r
#Solution
```

Problem 6 (optional)

On the first set of lecture notes, we covered an example for predicting punctuation given a male user has entered the phrase "thank you". We computed the loss for two different procedures $\hat{f}_1$ and $\hat{f}_2$. Now, we want to compute the risk, which is the expected value of the loss.

As a review, suppose a random variable Z takes a value 1 with probability π and 0 with probability $1-\pi$, where $0 \leq \pi \leq 1$. Then

$$\mathbb{E}Z = 1 * \pi + 0 * (1-\pi) = \pi.$$

Compute the following risks.

Part a

$$R(\hat{f}_1) = \mathbb{E}\ell(\hat{f}_1(X), Y) = \ldots$$

Part b

$$R(\hat{f}_2) = \mathbb{E}\ell(\hat{f}_2(X), Y) = \dots$$