IS605 Final

Max Wagner

May 22, 2016

Part I.

(1) What is the rank of the following matrix?

$$\begin{bmatrix} 1 & -1 & 3 & -5 \\ 2 & 1 & 5 & -9 \\ 6 & -1 & -2 & 4 \end{bmatrix}$$

3x4 matrix, so rank = 3

(2) What is the reduced row-echelon form of the above matrix?

With row reduction we can get:

```
m <- matrix(c(1,0,0,2/55,0,1,0,-14/55,0,0,1,-97/55), byrow = TRUE, ncol = 4)
m
```

```
##      [,1] [,2] [,3]        [,4]
## [1,]    1    0    0  0.03636364
## [2,]    0    1    0 -0.25454545
## [3,]    0    0    1 -1.76363636
```

(3) Define orthonormal basis vectors. Please write down at least one orthonormal basis for the 4-dimensional vector space R^4.

The basic definition is that they are unit vectors that are orthogonal. They must also be non-zero. An example could be:

```
m <- matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1), byrow = TRUE, ncol = 4)
m
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

(4) Given the following matrix, what is its characteristic polynomial?

$$A = \begin{bmatrix} 6 & 1 & 1 \\ 0 & 7 & -1 \\ -1 & 3 & 0 \end{bmatrix}$$

First we solve for the determinant:

$$det(\begin{bmatrix} 6 & 1 & 1 \\ 0 & 7 & -1 \\ -1 & 3 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}) = 0$$

Which will give a characteristic polynomial of:

$$p(\lambda) = -\lambda^3 + 13\lambda^2 - 46\lambda + 26$$

(5) What are its eigenvectors and eigenvalues? Please note that it is possible to get complex vectors as eigenvectors.

I can get eigenvalues from the process from above and plugging back into the original matrix, or from using R, let's use R.

```
m <- matrix(c(6,1,1,0,7,-1,-1,3,0), byrow = TRUE, ncol = 3)
print(eigen(m))
```

```
## $values
## [1] 6.7856670 5.5202305 0.6941025
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.8339613  0.9451655  0.2108156
## [2,] 0.5395686 -0.1828644 -0.1531045
## [3,] 0.1156473 -0.2705972 -0.9654614
```

(6) Given a column stochastic matrix of links between URLs, what can you say about the PageRank of this set of URLs? How is it related to its eigendecomposition?

Given that the column is stochastic (randomly determined), and given that PageRank assumes that more popular websites will receive more links, we can then assume that the PageRank for this set of URLs will be fairly random as well. Generally PageRank iterates until a stable result is found, in the case of random input there shouldn't be a stable result, so decomposition will continue to increase powers nearly infinitely, but ultimately the rankings will still be random.

(7) Assuming that we are repeatedly sampling sets of numbers (each set is of size n) from an unknown probability density function. What can we say about the mean value of each set?

If sampling from a random PDF, the mean values will lead to a normal distribution.

(8) What is the derivative of $e^x \sin^2(x)$?

$$e^x sin(x)(sin(x) + 2cos(x))$$

(9) What is the derivative of $\log(x^3 + \sin(x))$?

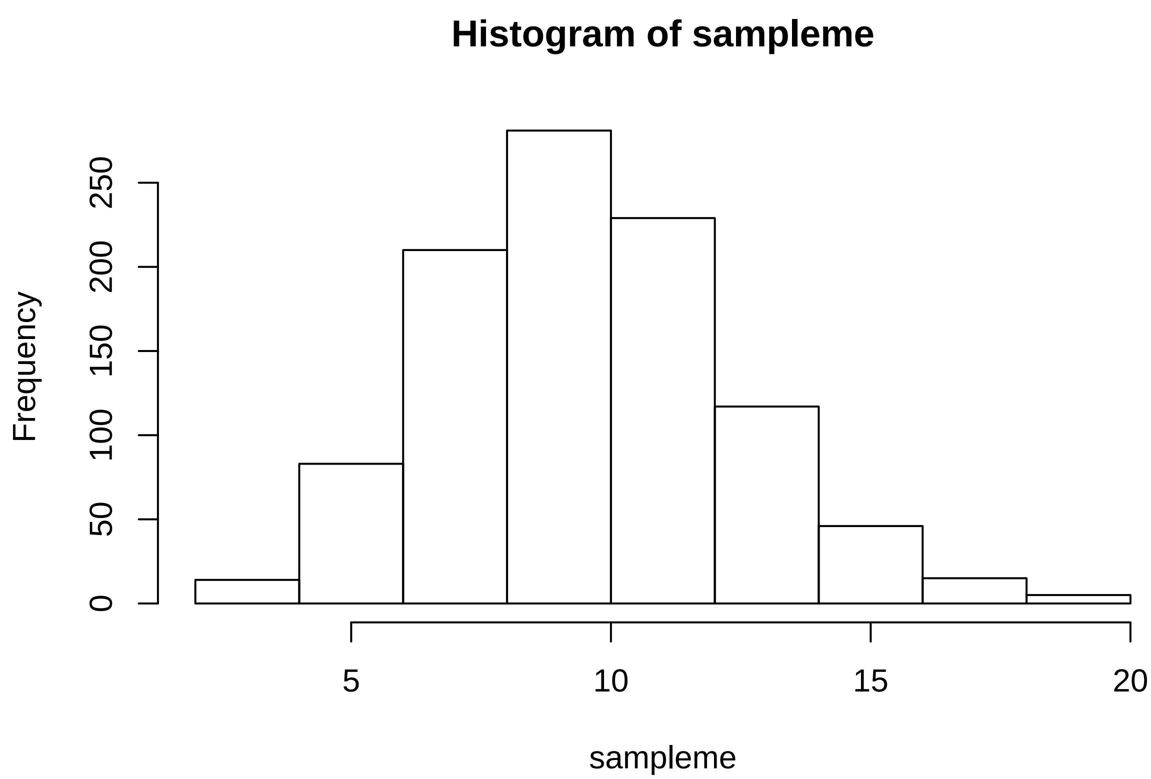
$$\frac{cos(x) + 3x^2}{sin(x) + x^3}$$

(10) What is $\int e^x \cos(x) + \sin(x)dx$? Don't forget the constant of integration.

$$\frac{e^x sin(x) + (e^x - 2)cos(x)}{2} + C$$

2.1. **Sampling from function.** Assume that you have a function that generates integers between 0 and 50 with the following probability distribution: $P(x == k) = \binom{50}{k} p^k q^{50-k}$ where $p = 0.2$ and $q = 1 - p = 0.8$ and $x \in [0, 50]$. This is also known as a Binomial Distribution. Write a function to sample from this distribution. After that, generate 1000 samples from this distribution and plot the histogram of the sample. Please note that the Binomial distribution is a discrete distribution and takes values only at integer values of x between $x \in [0, 50]$. Sampling from a discrete distribution with finite values is very simple but it is not the same as sampling from a continuous distribution.

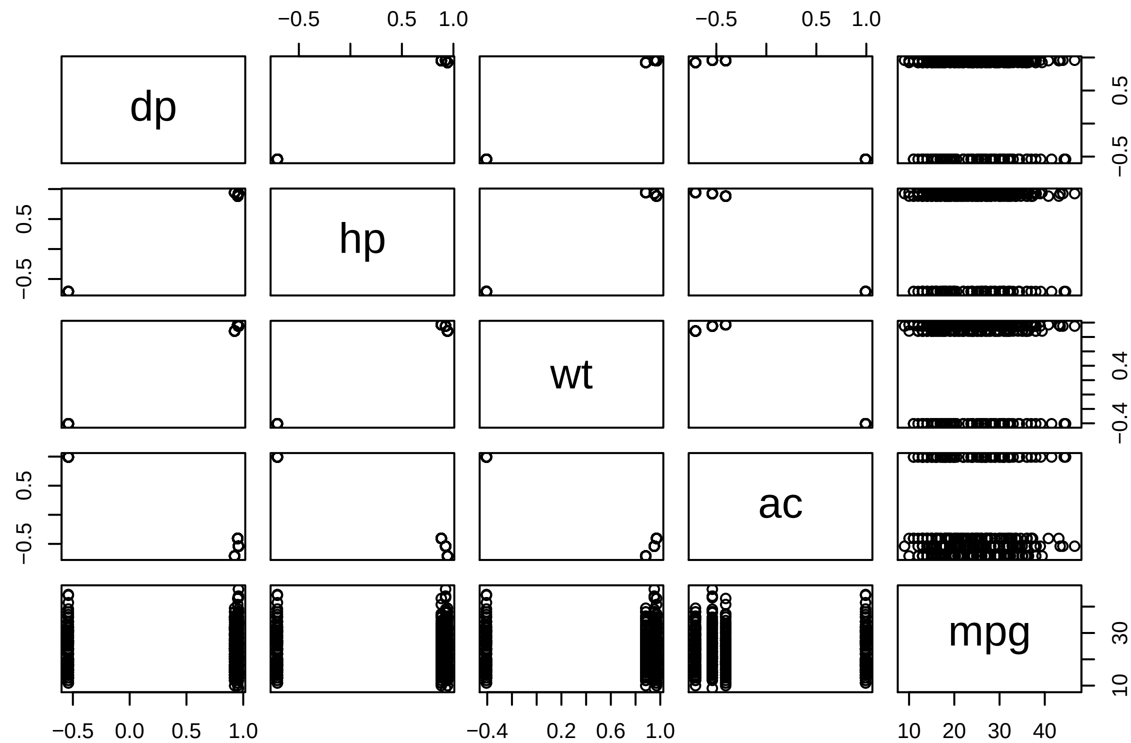
```
boink <- function(v) {
  return (dbinom(v, 50, .2))
}
v <- seq(0,50,1)
x <- boink(v)
sampleme <- sample(v, 1000, TRUE, x)
hist(sampleme)
```



It's normal, hooray!

2.2. **Principal Components Analysis.** For the auto data set attached with the final exam, please perform a Principal Components Analysis by performing an SVD on the 4 independent variables (with mpg as the dependent variable) and select the top 2 directions. Please scatter plot the data set after it has been projected to these two dimensions. Your code should print out the two orthogonal vectors and also perform the scatter plot of the data after it has been projected to these two dimensions.

```
auto <- read.table("auto-mpg.data", col.names = c("dp", "hp", "wt", "ac", "mpg"))
svd <- prcomp(scale(auto[1:4])) # scale for svd, and do a principal comp analysis
svd$sdev <- svd$sdev^2

# 2 dimensions
d <- diag(svd$sdev)[1:2,1:2]
u <- svd$rotation[,1:2]
v <- svd$rotation[,1:2]
auto_a <- u %*% d %*% t(v)
pairs(~.,cbind(auto_a,auto[5])) # looks a bit off
```



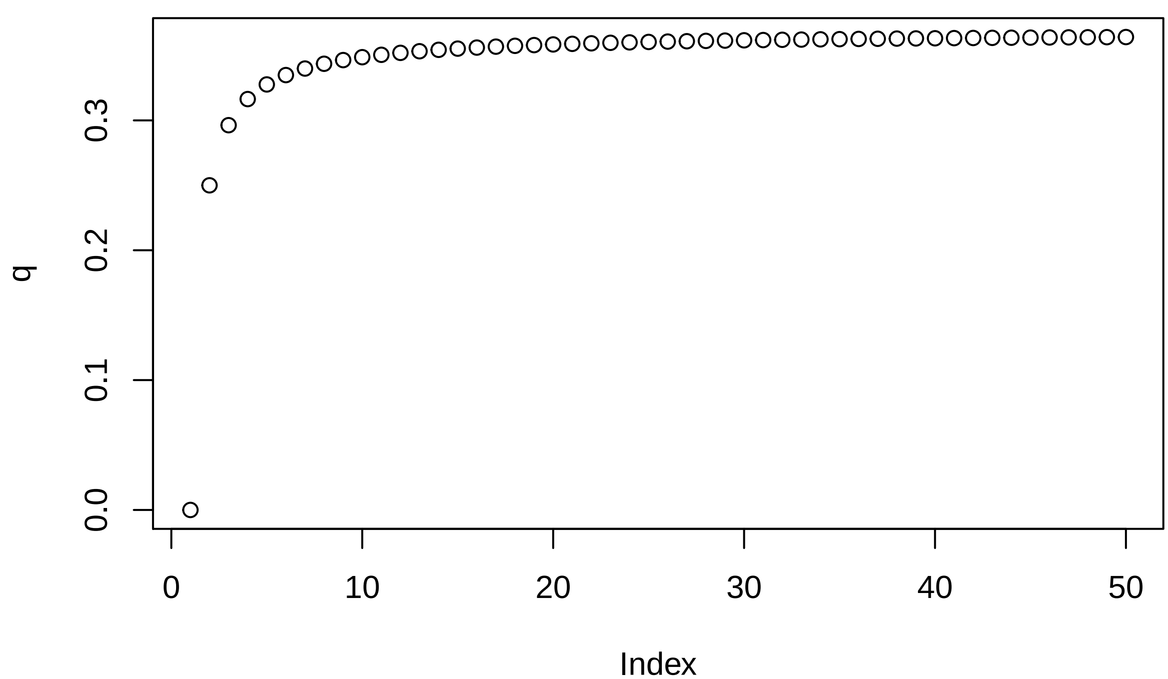
2.3. **Sampling in Bootstrapping.** As we discussed in class, in bootstrapping we start with n data points and repeatedly sample many times with replacement. Each time, we generate a candidate data set of size n from the original data set. All parameter estimations are performed on these candidate data sets. It can be easily shown that any particular data set generated by sampling n points from an original set of size n covers roughly 63.2% of the original data set. Using probability theory and limits, please prove that this is true. After that, write a program to perform this sampling and show that the empirical observation also agrees this.

$$p = 1/n$$
$$q = 1 - p$$
$$q_n = (1 - p)^n$$

```
n <- c(1:50)
q <-(1-1/n)^n
answer <- 1-max(q);answer
```

```
## [1] 0.6358303
```

```
plot(q)
```

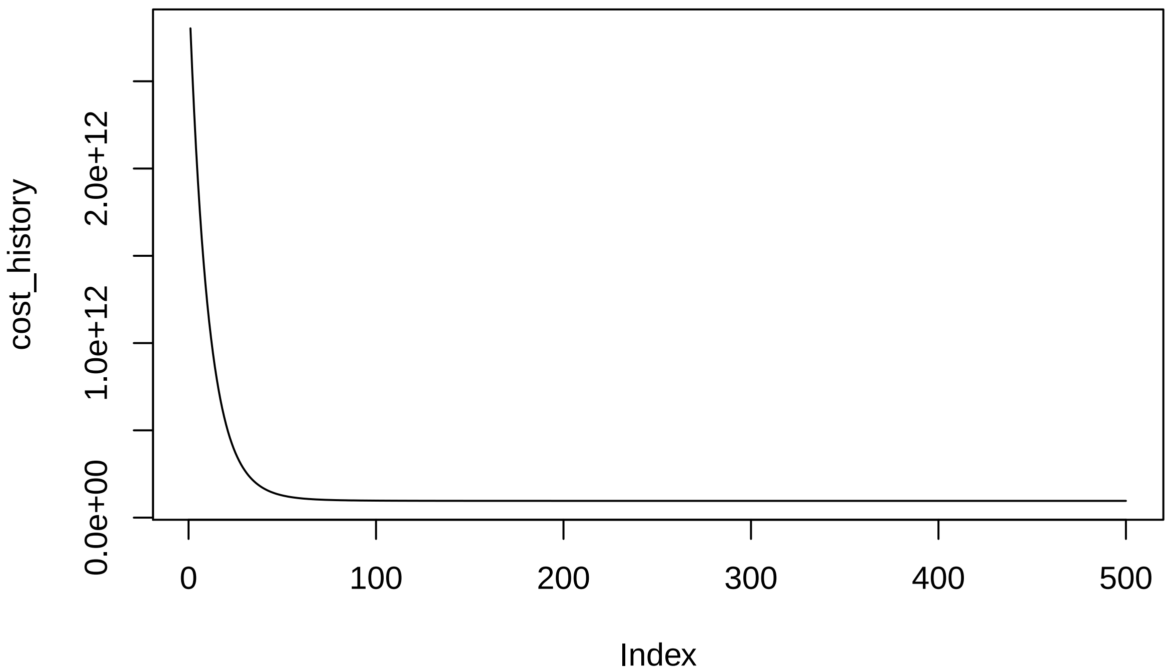


It checks out!

3. Mini-project - 20 points

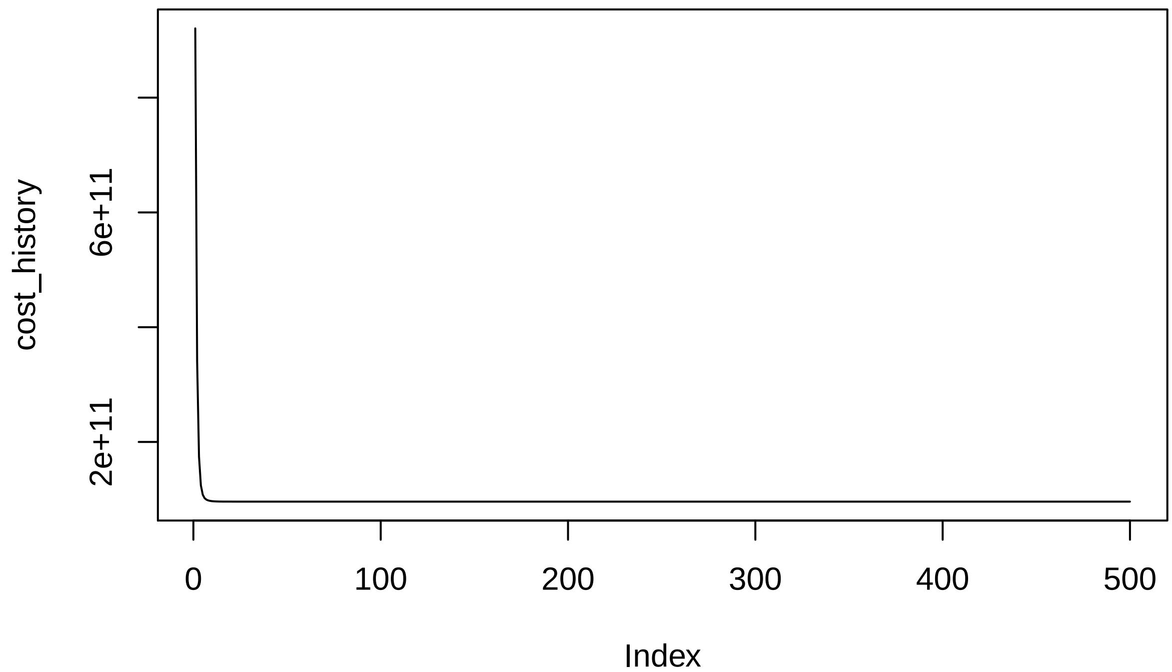
```r
# load in
ex3x <- read.table("mini-project-data/ex3x.dat", col.names = c("sqft", "beds"))
ex3y <- read.table("mini-project-data/ex3y.dat", col.names = c("price"))

# standardize with scale again, same as doing something like
# (ex3x[,1] - mean(ex3x[,1])) / sd(ex3x[,1])
ex3x <- scale(ex3x)

# let's use the Andrew Ng example
cost <- function(x,y,theta) {
  return (sum((x %*% theta - y)^2) / (2*length(y)))
}

# initialize some things
alpha <- c(.001,.01,.1,1)
iters <- 500
x <- as.matrix(cbind(1, ex3x))
y <- ex3y
theta <- matrix(c(0,0,0,0,0,0,0,0,0), nrow = 3) # work around for a matrix argument thing
cost_history <- c()

#gradient time
regress <- function(x,y,alpha,theta,iters) {
  for (i in 1:iters) {
    error <- as.matrix(x %*% theta - cbind(y,y,y))
    delta <- t(x) %*% error / length(y)
    theta <- theta - alpha * delta
    cost_history[i] <- cost(x,y,theta[,1])
  }
  print(theta[,1])
  plot(cost_history,type = "l")
}

# plug in
regress(x,y,alpha[1],theta,iters)
```

```
##                  sqft       beds
## 340412.660 110628.924  -6647.348
```



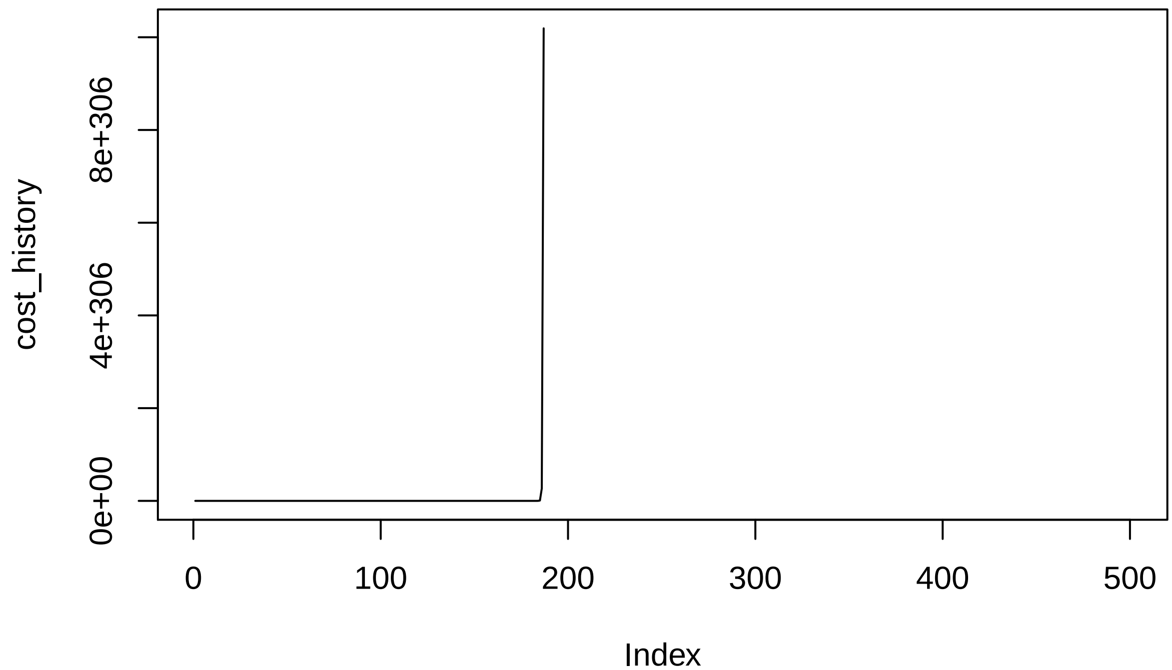

```r
regress(x,y,alpha[2],theta,iters)
```

```
##                    sqft       beds
## 340412.660 110631.050  -6649.474
```



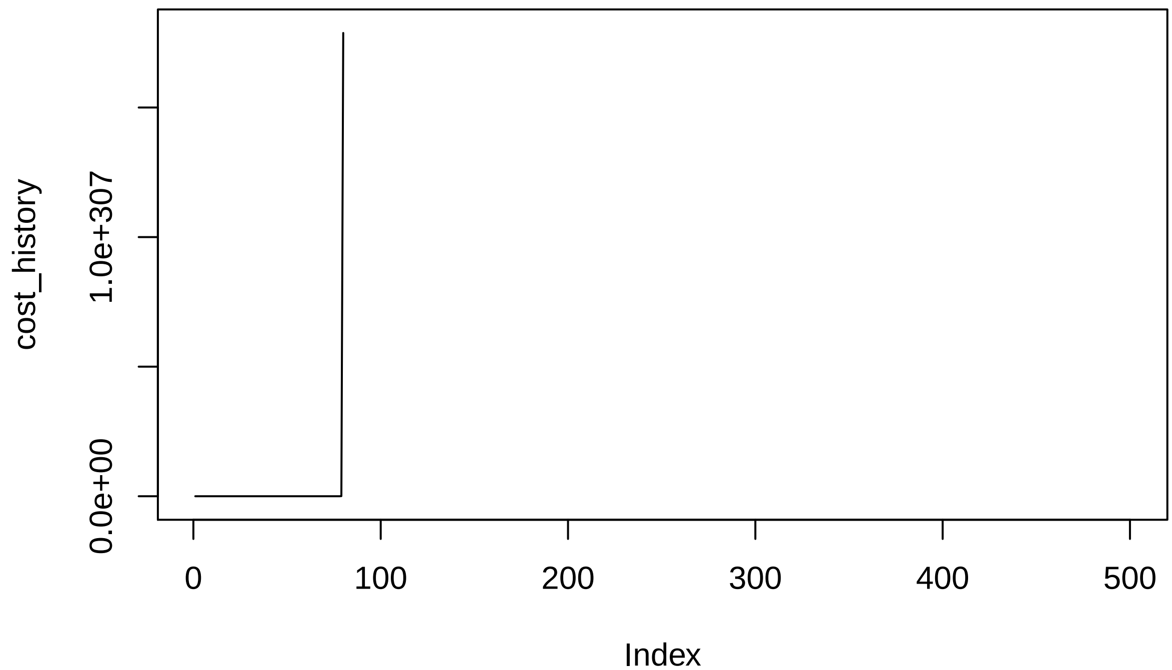

```r
regress(x,y,alpha[3],theta,iters)
```

```
##      sqft beds
##  NaN  NaN  NaN
```




```r
regress(x,y,alpha[4],theta,iters)
```

```
##       sqft beds
##  NaN  NaN  NaN
```



As we can see, alpha values of .1 and 1 are giving NaN currently. However, the first two graphs appear to be what was to be expected. We can try it with the built in linear regression to see what we get.

```
ex3x <- read.table("mini-project-data/ex3x.dat", col.names = c("sqft", "beds"))
ex3y <- read.table("mini-project-data/ex3y.dat", col.names = c("price"))
ex3x <- scale(ex3x)
houses <- cbind(ex3x,ex3y)
glm(price ~ sqft + beds, data = houses)
```

```
##
## Call:  glm(formula = price ~ sqft + beds, data = houses)
##
## Coefficients:
## (Intercept)         sqft         beds
##      340413       110631        -6649
##
## Degrees of Freedom: 46 Total (i.e. Null);  44 Residual
## Null Deviance:       7.192e+11
## Residual Deviance: 1.921e+11     AIC: 1182
```

It appears to be almost identical to the answers from the SGD at .001 and .01 alpha levels. I would assume this means at least part of the above is correct.