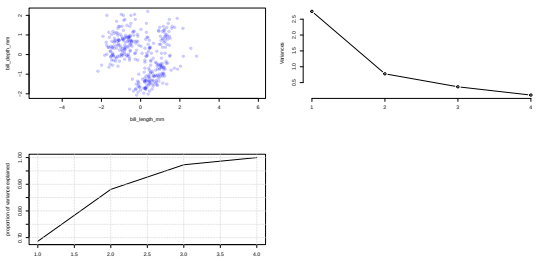
STAT312 Assignment 1 Name: Joshua Buchanan

Q1. (a)

```
> penguins = penguins[complete.cases(penguins),]
> X = scale(penguins[,3:6], scale=TRUE)
> #n = 333, p = 4, r = 4
> plot(X,asp=1,pch=19,col=rgb(0,0,1,alpha=0.2))
> pc = prcomp(X)
> plot(pc,type="l",main="")
> plot(1:4,cumsum(pc$sdev^2)/sum(pc$sdev^2),type="l",xlim=c(1,4),xlab="",
 ylab="proportion of variance explained")
> grid()
>
> #Z matrix
> pc$x
             PC1           PC2           PC3           PC4
1   -1.85080775 -0.032021188  0.2345486909  0.5276026405
2   -1.31427621  0.442860308  0.0274287986  0.4011229839
3   -1.37453656  0.160988208 -0.1894042308 -0.5278675182
5   -1.88245548  0.012332676  0.6279277238 -0.4721826194
6   -1.91709572 -0.816369578  0.6999979673 -0.1961213213
7   -1.77035612  0.365672659 -0.0284176935  0.5046091971
8   -0.81726635 -0.500489901  1.3329977168  0.3477362505
  ------------------------------------------------------
259  1.01750383  1.197651475  0.2508191726 -0.0067575085
 [ reached getOption("max.print") -- omitted 83 rows ]
>
> #A matrix
> t(pc$rotation)
                          PC1         PC2        PC3         PC4
bill_length_mm      0.4537532 -0.60019490 -0.6424951  0.1451695
bill_depth_mm      -0.3990472 -0.79616951  0.4258004 -0.1599044
flipper_length_mm   0.5768250 -0.00578817  0.2360952 -0.7819837
body_mass_g         0.5496747 -0.07646366  0.5917374  0.5846861
```



(b)

```
> A = t(pc$rotation)
> Z = X %*% t(A)
```

```
>
> #picks up on inconsistencies
> #in rounding so return FALSE
> identical(Z, pc$x)
[1] FALSE
> all(Z == pc$x)
[1] FALSE
>
> #returns TRUE which shows X %*% t(A) is
> #equal to Z accounting for rounding errors
> all(Z - pc$x < 1.0 * 10^-14)
[1] TRUE
```

(c)
```
> covmat = 1/(333-1) * t(X) %*% X
>
> A1 = princomp(covmat=covmat)
> A1$loadings

Loadings:
                  Comp.1 Comp.2 Comp.3 Comp.4
bill_length_mm     0.454  0.600  0.642  0.145
bill_depth_mm     -0.399  0.796 -0.426 -0.160
flipper_length_mm  0.577         -0.236 -0.782
body_mass_g        0.550         -0.592  0.585
```

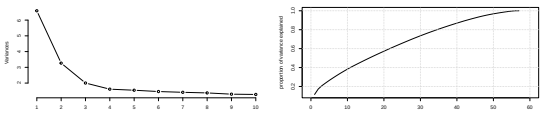
Some of the signs are changed and the two smallest values from the original A matrix are not included.

Q2. (a) The spam data set has 57 numerical variables and one factor variable (type, the variable of interest). 4601 observations for each variable. dim(spam) = 4601, 58.

(b)
```
> library(kernlab)
> data(spam)
>
> Xspam = scale(spam[,1:57])
>
> pcspam = prcomp(Xspam)
>
> plot(pcspam, type="l",main="")
> plot(1:57,cumsum(pcspam$sdev^2)/sum(pcspam$sdev^2),type="l",xlim=c(0,60),x
> grid()
>
> #Z matrix
> pcspam$x
                PC1           PC2           PC3           PC4
1    -7.316702e-01 -4.302089e-02 -5.805991e-01 -2.511157e-01
2    -1.184956e+00  2.067625e+00  3.603072e-02  4.304122e-01
```

```
3    -1.467435e+00  5.023208e+00  3.277235e+00  9.949572e-01
4    -8.052598e-01  4.274566e-01 -5.830434e-01 -8.241236e-03
5    -8.061914e-01  4.267008e-01 -5.850911e-01 -8.166074e-03
6    -4.926199e-01 -4.895102e-01 -3.733637e-01 -4.433782e-01
7    -1.025721e+00  1.020638e+00 -1.737424e+00  1.495643e-02
8    -5.043950e-01 -4.874826e-01 -4.184803e-01 -4.322688e-01
9    -1.265801e+00  3.640849e+00  1.491548e+00  1.192208e-01
10   -8.411384e-01  4.004673e-01  2.986462e-02 -2.959398e-01
11   -1.830553e-01  1.702605e+00 -1.677627e+00  6.198921e-01
12   -6.391914e-01 -4.641919e-01 -4.725005e-01 -4.448842e-01
13   -6.937291e-01 -5.755557e-02 -5.769809e-01 -2.330295e-01
14   -1.052856e+00  1.032881e+00 -1.491406e+00  2.712803e-01
15   -1.281213e+00  1.661085e+00 -1.723951e+00 -1.129273e-01
16   -3.520490e-01  2.804161e-01 -1.580456e-01  6.501212e-01
17   -6.997410e-01 -3.691750e-01 -8.935168e-01 -4.505809e-01
-----------------------------------------------------
              PC57
1    -2.703945e-03
2    -2.051520e-03
3    -6.793555e-04
4    -3.688456e-04
5    -3.743049e-04
6    -3.192206e-04
7    -1.323788e-03
8    -2.590903e-04
9     6.249279e-03
10   -1.640090e-03
11    1.277468e-02
12   -2.371054e-03
13   -2.608710e-03
14    1.204930e-03
15    2.495726e-03
16    1.641009e-03
17   -1.676886e-03
 [ reached getOption("max.print") -- omitted 4584 rows ]
>
> #A matrix
> pcspam$rotation
                            PC1          PC2           PC3
make              -0.043672283  0.169619189 -0.0636415262
address           -0.011108508 -0.016823000 -0.0095522991
all               -0.047165679  0.165185650 -0.0206819240
num3d             -0.006222519  0.011017233  0.0126783323
our               -0.036747189  0.121653457 -0.1368425459
over              -0.045816921  0.167773971  0.0069760760
```

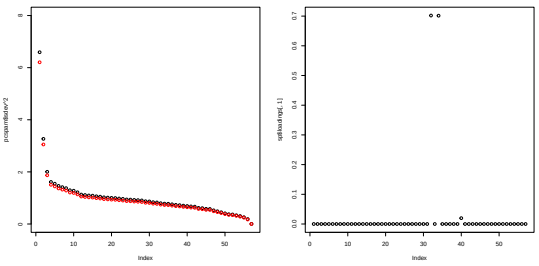
```
remove            -0.046221024  0.144368111 -0.1292322337
internet          -0.033913841  0.132235561 -0.0470827299
order             -0.045550187  0.235003660  0.1315532647
mail              -0.020205702  0.153591425  0.0604323825
receive           -0.050354981  0.218739738 -0.1039657916
will              -0.022181403  0.076685909 -0.0922257849
people            -0.037573990  0.125069915 -0.0112944912
report            -0.017977580  0.068652666  0.0614676249
addresses         -0.030883904  0.217401181  0.1487721204
free              -0.042761454  0.101695607 -0.1269380855
business          -0.045146718  0.199460832 -0.1109913158
-----------------------------------------------
                            PC55          PC56          PC57
make              -0.0109306292  0.0196971556 -2.948893e-03
address            0.0444816175  0.0211655803  1.064680e-03
all               -0.0230950529  0.0062538090  5.969686e-04
num3d              0.0163920212  0.0046016125  1.283731e-04
our                0.0085329086  0.0181205866  5.649072e-04
over               0.0156472696  0.0190133051  4.184348e-04
remove             0.0338644604  0.0059177013  6.562763e-04
internet           0.0021116871 -0.0060350128  2.651995e-04
order             -0.0104797132  0.0133788233  8.668275e-04
mail              -0.0080349287 -0.0038143891 -2.175567e-04
receive            0.0006487631  0.0152026963  6.816880e-05
will               0.0096333233 -0.0179523195  5.539508e-04
people             0.0169831890 -0.0024541630  6.469989e-04
report            -0.0129813970  0.0049193659  3.785802e-04
addresses          0.1432587085 -0.0554517253 -5.671017e-04
free               0.0183727671  0.0011251802 -4.970406e-05
business          -0.0040448243 -0.0453310458 -4.784601e-04
 [ reached getOption("max.print") -- omitted 40 rows ]
>
```



Because the curve on the second graph above is almost linear this is evidence that there is not low-dimensional representation of the data. With each principal component added, the variance explained goes up a similar amount. However, the first graph shows huge variance in

the first principal component which is indicative of low-dimensional representation.

(c)
```
> library(sparsepca)
> sp = spca(Xspam,k=57, alpha = 1/100, beta=1e-4, verbose=FALSE)
> plot(pcspam$sdev^2,ylim=c(0,8))
> points(sp$sdev^2,col="red")
> plot(sp$loadings[,1])
> which(sp$loadings[,1] != 0)
[1] 32 34 40
```



Variables 32, 34, and 40 remain when extreme sparsity is enforced. With variation explained not changing much at all this is indicative that there is a sparse representation of the data. As alpha is decreased, there is more noise and more variables are included in the sparse representation. As beta is increased, there is more noise also. When there is some more noise, the variation explained actually changes even more which is an argument against sparse representation of the data. The close variation explained with the alpha and beta I chose could be chance.

(d)
```
> set.seed(238)
>   train = sample(nrow(spam),0.7*nrow(spam))
>   test = -train
>   Xtrain = scale(spam[train,-58])
>   Xtest = scale(spam[test,-58],center=attr(Xtrain,"scaled:center"),
+                   scale=attr(Xtrain,"scaled:scale"))
>     pc = prcomp(Xtrain)
>     Atr = pc$rotation
>
>   Ztrain = Xtrain %*% Atr[,1:10]
>   Ztrain = data.frame(Ztrain, spam[train,58])
>
>   Ztest = Xtest %*% Atr[,1:10]
>   Ztest = data.frame(Ztest, spam[test,58])
>
>   mdl = glm(spam.test..58.~., data=Ztest, family="binomial")
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

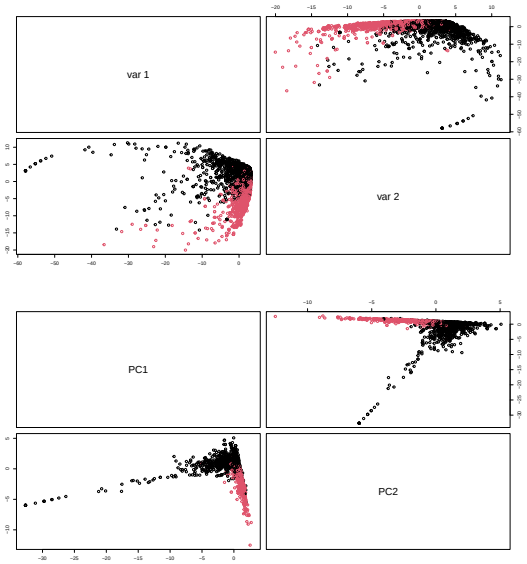
```
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

       nonspam spam
  FALSE     784   89
  TRUE       51  457
>
>   mdl = glm(spam.test..58.~.-PC10, data=Ztest, family="binomial")
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

       nonspam spam
  FALSE     784   89
  TRUE       51  457
>
>   mdl = glm(spam.test..58.~.-PC10-PC9, data=Ztest, family="binomial")
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

       nonspam spam
  FALSE     774   94
  TRUE       61  452
>
>   mdl = glm(spam.test..58.~.-PC10-PC9-PC8, data=Ztest, family="binomial")
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

       nonspam spam
  FALSE     770  101
  TRUE       65  445
>
>   mdl = glm(spam.test..58.~.-PC10-PC9-PC8-PC7, data=Ztest, family="binomia
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

       nonspam spam
  FALSE     770  102
  TRUE       65  444
>
>   mdl = glm(spam.test..58.~.-PC10-PC9-PC8-PC7-PC6, data=Ztest, family="bin
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
```

```
>   table(pred>0.5, Ztest$spam.test..58.)

        nonspam spam
  FALSE     773  100
  TRUE       62  446
>
>   mdl = glm(spam.test..58.~.-PC10-PC9-PC8-PC7-PC6-PC5, data=Ztest, family=
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

        nonspam spam
  FALSE     778  100
  TRUE       57  446
>
>     mdl = glm(spam.test..58.~.-PC10-PC9-PC8-PC7-PC6-PC5-PC4, data=Ztest, fa
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

        nonspam spam
  FALSE     774  107
  TRUE       61  439
>
>     mdl = glm(spam.test..58.~.-PC10-PC9-PC8-PC7-PC6-PC5-PC4-PC3, data=Ztest
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
>   pred = predict(mdl, type="response")
>   table(pred>0.5, Ztest$spam.test..58.)

        nonspam spam
  FALSE     773  107
  TRUE       62  439
>
```

With the first two PCs the misclassification error is 169/1381. With three, it is 168/1381. With four it is 157/1381. With five it is 162/1381. With six it is 167/1381. With seven, 166/1381. With eight, 155/1381. With nine, 140/1381. And with ten, 140/1381. The misclassification error goes down as more PCs are added but not by much, showing that the data can be compressed without much loss down to two principle components.

(e)
```
> library(kernlab)
> kpc = kpca(Xtrain, kernel = "rbfdot",kpar=list(sigma=0.001),features=10)
> eig(kpc)
     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
```

```
0.010567350 0.006456637 0.003923801 0.003062551 0.002603799
     Comp.6      Comp.7      Comp.8      Comp.9     Comp.10
0.002375856 0.002200311 0.002172574 0.002121338 0.001894428
>
> Z = rotated(kpc)
> pairs(Z,col=as.numeric(Ztrain$spam.train..58.))
> pairs(Ztrain[,1:2],col=as.numeric(Ztrain$spam.train..58.))
```



Plot using kernel PCA is much more rounded whereas the plot from the data in d) is more obviously different between groups. Misclassification error is similar.