STAT 3014/ 3914

Semester 2	Applied Statistics	2012

Lab 1

Lab Exercise.

Download the file “comData.Rdata” from either Blackboard or Ed. Use the command

```
load("comData.Rdata")
```

to load the data into memory. The file contains 3 objects

```
ls()
```

```
[1] "Info" "X"    "Y"
```

The dataset was collected with the aim of determining what demographics (contained in the **X** matrix) lead to different broad categories of crime (contained in the **Y** matrix) based on a subset of U.S. cities.

Use the code and results below to construct a report analysing the data.

1. **Data cleaning.**

   (a) Summarise the basic characteristics of the **X** and **Y** matrices. Characterise the missing values in **X**.

   ```
   dim(X)
   dim(Y)
   colnames(X)
   colnames(Y)
   summary(X)
   ```

   (b) Plot the correlation matrix **X**. Use different methods “complete.obs”, “na.or.complete” and “pairwise.complete.obs” to handle the missing values.

   ```
   colRamp <- colorRampPalette(c("red", "green"))
   cols <- colRamp(50)

   R <- cor(X)
   R2 <- cor(X,use="complete.obs")
   R3 <- cor(X,use="na.or.complete")
   R4 <- cor(X,use="pairwise.complete.obs")

   par(mfrow=c(2,2))
   image(R,zlim=c(-1,1),col=cols)
   image(R2,zlim=c(-1,1),col=cols)
   image(R3,zlim=c(-1,1),col=cols)
   image(R4,zlim=c(-1,1),col=cols)
   ```

(c) Use the following function and the function `apply()` to find the number of missing entries in each row and each column.

```
sum.na <- function(x) {  sum(is.na(x)); }
```

To simplify analysis remove all columns from the matrix $\mathbf{X}$ which contains missing values and put the results in a matrix `X2`.

```
dim(X2)
```

```
[1] 2215  101
```

(d) The singular value decomposition (SVD) of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ (closely related to an eigenvalue decomposition) is a matrix factorization of the form $\mathbf{X} = \mathbf{U}\text{diag}(\mathbf{d})\mathbf{V}$ where $\mathbf{U} \in \mathbb{R}^{n \times p}$ is an orthonormal matrix, $\mathbf{d} \in \mathbb{R}^p$ is a vector of singular values (the square-root of the eigenvalues of $\mathbf{X}^T\mathbf{X}$) and $\mathbf{V} \in \mathbb{R}^{p \times p}$ is an orthonormal matrix. The SVD can be used to detect linear dependence in the covariates.

```
res.svd <- svd(X2)
```

```
[1] 1.508527e+07 1.166553e+07 1.334989e+06 1.147312e+06 8.985595e+05 6.630348
[7] 4.155246e+05 3.628540e+05 3.513764e+05 2.795465e+05 2.139332e+05 2.064011
[13] 1.531325e+05 1.342424e+05 1.216366e+05 8.815084e+04 7.291858e+04 3.78079
[19] 1.872939e+04 9.099117e+03 5.672523e+03 4.735951e+03 3.322255e+03 2.56002
[25] 1.296034e+03 1.069461e+03 8.886158e+02 8.732051e+02 7.624523e+02 7.34433
[31] 6.816092e+02 5.748951e+02 5.289567e+02 4.595183e+02 4.453243e+02 3.73862
[37] 3.442402e+02 3.243234e+02 3.015637e+02 2.762614e+02 2.600174e+02 2.44441
[43] 2.285669e+02 2.045496e+02 1.988509e+02 1.840301e+02 1.673946e+02 1.61113
[49] 1.541865e+02 1.484970e+02 1.418894e+02 1.395953e+02 1.342372e+02 1.28496
[55] 1.250864e+02 1.209288e+02 1.157892e+02 1.130087e+02 1.085100e+02 1.05886
[61] 1.016245e+02 9.561030e+01 9.167637e+01 8.708091e+01 8.346494e+01 8.12054
[67] 7.826118e+01 7.071691e+01 6.626852e+01 6.523241e+01 6.333076e+01 6.15261
[73] 5.733385e+01 5.371514e+01 5.132049e+01 4.807535e+01 4.525677e+01 4.30722
[79] 4.125820e+01 3.889626e+01 3.855151e+01 3.580030e+01 3.508800e+01 3.29151
[85] 2.819969e+01 2.384917e+01 1.687881e+01 1.462436e+01 1.426467e+01 1.34004
[91] 1.053330e+01 6.416995e+00 4.781234e+00 4.402443e+00 3.443868e+00 2.41561
[97] 2.123672e+00 1.157965e+00 7.427157e-01 8.344063e-09 1.164014e-09
```

There are two singular values which are close two zero. This indicates that there are at least two columns which can be written as linear combinations of the other variables. The following code tells us which ones:

```
colnames(X2)[ abs( res.svd$v[,100])>0.001 ]
colnames(X2)[ abs( res.svd$v[,101])>0.001 ]
```

Remove two columns from the matrix $\mathbf{X}$ so that the matrix $\mathbf{X}^T\mathbf{X}$ is non-singular and put the results in a matrix `X3`.

```
dim(X3)
```

```
[1] 2215   99
```

2. **Principal component analysis.** Perform a principal component analysis.

```
pca <- prcomp(X3,center = TRUE, scale. = TRUE)

Cols = function(vec) {
        cols = rainbow(length(unique(vec)))
        return(cols[as.numeric(as.factor(vec))])
}

cats <- as.numeric( Y[,1]>quantile(Y[,1],0.75,na.rm=TRUE) )
par(mfrow=c(1,2))
plot(pca$x[,1:2],col=Cols(cats),pch=19,xlab="Y1",ylab="Y2" ,main = colnames(Y)[1]
plot(pca$x[,c(1,3)],col=Cols(cats),pch=19,xlab="Y1",ylab="Y3",main = colnames(Y)[
```

   (a) What variables are the most related to the first two principal components?

   (b) What proportion of variance is explained (PVE) by the first 5 principal components?

   (c) Plot PVE against the number of principal components.

   (d) Plot cumulative PVE against the number of principal components.

   (e) The above code looks to see whether the principal components separate high values (the top quartile) of the first column of **Y** from low values. Modify the above code and state for which variables in **Y** are similarly separated by the first three principal components.

3. **Clustering**

```
# Categorizes each city by the maximum ``crime type''
Z <- colnames(Y)[apply(Y,1,which.max)]

X4 <- scale(X3)
km <- kmeans(X4,10,nstart=100)
cbind( table( km$cluster,Z ), table( km$cluster ))

res.hclust <- hclust( dist(X4), method="complete" )
res.cutree <- cutree( res.hclust, k = 10 )
cbind( table( res.cutree,Z ), table( res.cutree ))
```

The above code uses k-means and hierarchical clustering to create 10 clusters. Describe the relationships between the k-means clustering results and the categories defined by **Z**. Similarly describe the relationships between the hierarchical clustering results and the categories defined by **Z**. Compare the k-means and hierarchical clustering methods. Perform the same analysis with 20 clusters.