Homework 2 EEB590C

Devin Molnau, Holly Loper, Elizabeth McMurchie

April 10, 2021

Assignment:

Select one of the two datasets (HW2.dat1.csv or HW2.dat2.csv found in the Data Repository). Each contains a multivariate dataset and several independent (X) variables. Using the methods learned in weeks 6-10, examine patterns in the dataset. You may use one or more (or all) of the X-variables, and a variety of methods to describe the patterns.

You must use at least one method from the material learned in: Weeks 6-7, Week 8, Week 9, and Week 10

```
## Loading required package: rgl
```

```
## Loading required package: Matrix
```

Read in data1

```
#READ in both csv datasets
dat1<-read.csv("HW2.dat1.csv", header= TRUE)
# dat2<-read.csv("HW2.dat2.csv", header = TRUE)
```

WEEK 6 MATERIAL

We selected dataset 1 to analyse.

```
dat1.dat<-log(as.matrix(dat1[,(4:9)]))
mydat<-rrpp.data.frame("Y"=dat1.dat,"X1"= as.factor(dat1$X1),"X2"= as.factor(dat1$X2),"X3"= dat1$X3)
```

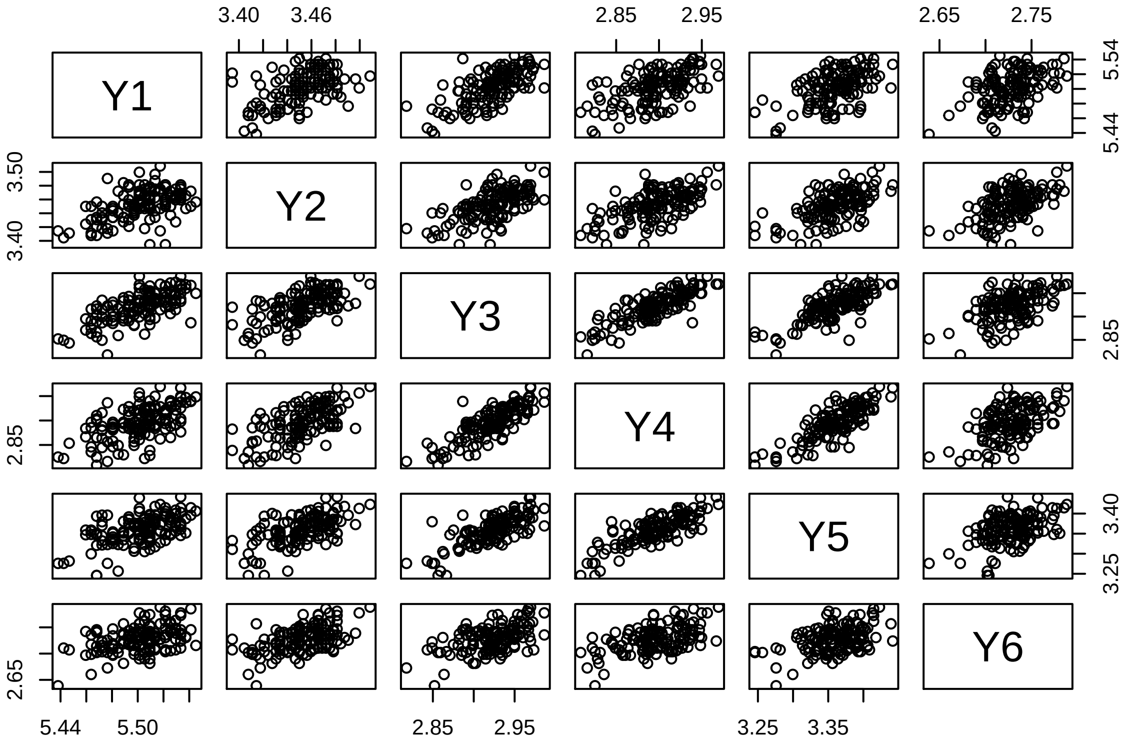
A brief look at the dataset. Looking at the correlation of the Y values.

```
cor(dat1.dat)
```

```
##           Y1        Y2        Y3        Y4        Y5        Y6
## Y1 1.0000000 0.5035656 0.6794433 0.5778571 0.5336809 0.4364807
## Y2 0.5035656 1.0000000 0.6234217 0.6163699 0.5853149 0.5347691
## Y3 0.6794433 0.6234217 1.0000000 0.8215338 0.7486924 0.5121106
## Y4 0.5778571 0.6163699 0.8215338 1.0000000 0.8107807 0.5229477
## Y5 0.5336809 0.5853149 0.7486924 0.8107807 1.0000000 0.4609773
## Y6 0.4364807 0.5347691 0.5121106 0.5229477 0.4609773 1.0000000
```

Pairs gives us a metric of scatterplots of the Y data just so we can visualize our multivariate data.

```
pairs(dat1.dat)
```



To look at the variances of the Y values we use the var() function.

```
var(dat1.dat)
```

```
##              Y1           Y2           Y3           Y4           Y5
## Y1 0.0005119657 0.0002543216 0.0004889894 0.0004448629 0.0004388798
## Y2 0.0002543216 0.0004982111 0.0004426031 0.0004680943 0.0004748318
## Y3 0.0004889894 0.0004426031 0.0010116995 0.0008890710 0.0008655112
## Y4 0.0004448629 0.0004680943 0.0008890710 0.0011576319 0.0010026103
## Y5 0.0004388798 0.0004748318 0.0008655112 0.0010026103 0.0013209518
## Y6 0.0002462486 0.0002976194 0.0004061418 0.0004436410 0.0004177450
##              Y6
## Y1 0.0002462486
## Y2 0.0002976194
## Y3 0.0004061418
## Y4 0.0004436410
## Y5 0.0004177450
## Y6 0.0006216936
```

To scale that variance we can use scale() which will make the diagonal equal to 1.000

```
var(scale(dat1.dat))
```

```
##           Y1        Y2        Y3        Y4        Y5        Y6
## Y1 1.0000000 0.5035656 0.6794433 0.5778571 0.5336809 0.4364807
## Y2 0.5035656 1.0000000 0.6234217 0.6163699 0.5853149 0.5347691
## Y3 0.6794433 0.6234217 1.0000000 0.8215338 0.7486924 0.5121106
## Y4 0.5778571 0.6163699 0.8215338 1.0000000 0.8107807 0.5229477
## Y5 0.5336809 0.5853149 0.7486924 0.8107807 1.0000000 0.4609773
## Y6 0.4364807 0.5347691 0.5121106 0.5229477 0.4609773 1.0000000
```

```
#dist(dat1.dat, method= "euclidean")
```

Single factor MANOVA

To do a multivariate Y data against the X1 column we first ran a linear model lm() and then looked at the results using summary(manova()) Pillai's and Wilk's. We then do the test again using MANOVA from the RRPP package.

```
#single factor MANOVA
x1<-as.factor(dat1$X1)
model1 <- lm(dat1.dat~x1)
summary(model1) #yields a set of univariate analyses
```

```
## Response Y1 :
##
## Call:
## lm(formula = Y1 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.047855 -0.013544 -0.001324  0.015324  0.043495
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  5.510712   0.002068 2664.377  < 2e-16 ***
## x11         -0.024778   0.003446   -7.191 4.09e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01929 on 134 degrees of freedom
## Multiple R-squared:  0.2784, Adjusted R-squared:  0.2731
## F-statistic: 51.71 on 1 and 134 DF,  p-value: 4.089e-11
##
##
## Response Y2 :
##
## Call:
## lm(formula = Y2 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.059680 -0.013926  0.001607  0.015340  0.060176
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  3.454188   0.002383 1449.525   <2e-16 ***
## x11         -0.005808   0.003970   -1.463    0.146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02223 on 134 degrees of freedom
## Multiple R-squared:  0.01572,    Adjusted R-squared:  0.008377
## F-statistic:  2.14 on 1 and 134 DF,  p-value: 0.1458
```

```
##
##
## Response Y3 :
##
## Call:
## lm(formula = Y3 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.108253 -0.018507  0.003394  0.020429  0.070796
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.925970   0.003379 865.827   <2e-16 ***
## x11         -0.010478   0.005630  -1.861   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03152 on 134 degrees of freedom
## Multiple R-squared:  0.0252, Adjusted R-squared:  0.01792
## F-statistic: 3.464 on 1 and 134 DF,  p-value: 0.06492
##
##
## Response Y4 :
##
## Call:
## lm(formula = Y4 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.088257 -0.015131 -0.000352  0.027031  0.075104
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.896828   0.003659 791.670   <2e-16 ***
## x11         -0.002451   0.006096  -0.402    0.688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03413 on 134 degrees of freedom
## Multiple R-squared:  0.001205,   Adjusted R-squared:  -0.006248
## F-statistic: 0.1617 on 1 and 134 DF,  p-value: 0.6882
##
##
## Response Y5 :
##
## Call:
## lm(formula = Y5 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.112349 -0.018086  0.000696  0.024309  0.083726
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.358038   0.003906  859.78   <2e-16 ***
## x11         0.003967   0.006507    0.61    0.543
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03643 on 134 degrees of freedom
## Multiple R-squared:  0.002767,   Adjusted R-squared:  -0.004675
## F-statistic: 0.3718 on 1 and 134 DF,  p-value: 0.5431
##
##
## Response Y6 :
##
## Call:
## lm(formula = Y6 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.086408 -0.018190 -0.000752  0.015316  0.063325
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  2.729201   0.002675 1020.320   <2e-16 ***
## x11         -0.004065   0.004456   -0.912    0.363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02495 on 134 degrees of freedom
## Multiple R-squared:  0.00617,    Adjusted R-squared:  -0.001246
## F-statistic: 0.8319 on 1 and 134 DF,  p-value: 0.3634
```

```r
summary(manova(model1)) #does multivariate test (using Pillai's)
```

```
##            Df  Pillai approx F num Df den Df    Pr(>F)
## x1          1 0.44162   17.005      6    129 2.098e-14 ***
## Residuals 134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

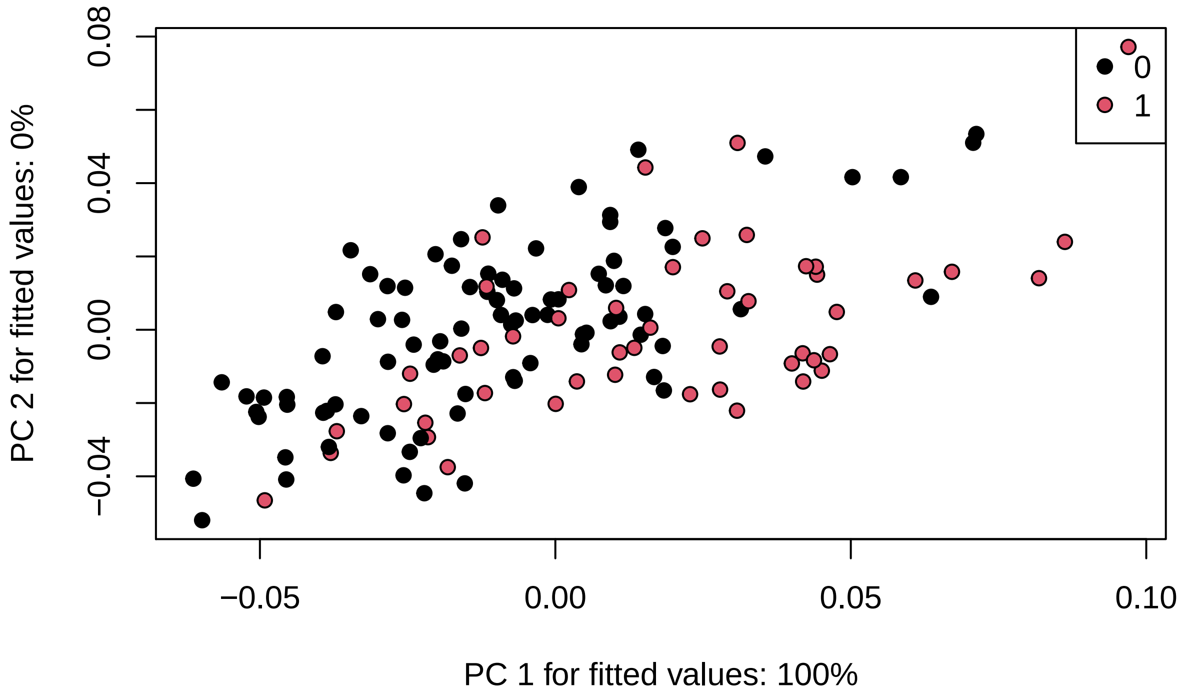
```r
summary(manova(model1),test="Wilks")    #does multivariate test (using Wilks)
```

```
##            Df   Wilks approx F num Df den Df    Pr(>F)
## x1          1 0.55838   17.005      6    129 2.098e-14 ***
## Residuals 134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
##### MANOVA via RRPP
model.rrpp <- lm.rrpp(dat1.dat~x1,data = mydat, print.progress = FALSE)
anova(model.rrpp)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type I
## Effect sizes (Z) based on F distributions
##
##            Df      SS        MS     Rsq      F      Z Pr(>F)
## x1          1 0.02494 0.0249429 0.03607 5.0144 2.1329  0.018 *
## Residuals 134 0.66655 0.0049742 0.96393
## Total     135 0.69149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = dat1.dat ~ x1, data = mydat, print.progress = FALSE)
```

```
plot(model.rrpp, type = "PC", pch=21, bg = x1)  #PC PLOT!
legend("topright", levels(x1), pch = 21, pt.bg = 1:4)
```



Factorial MANOVA

To assess the multivariate Y data against both X1 and X2 we use a factorial manova using the base R and then again in RRPP. There is a significance correlation of F=1.777e-14 between our Y data and X1.

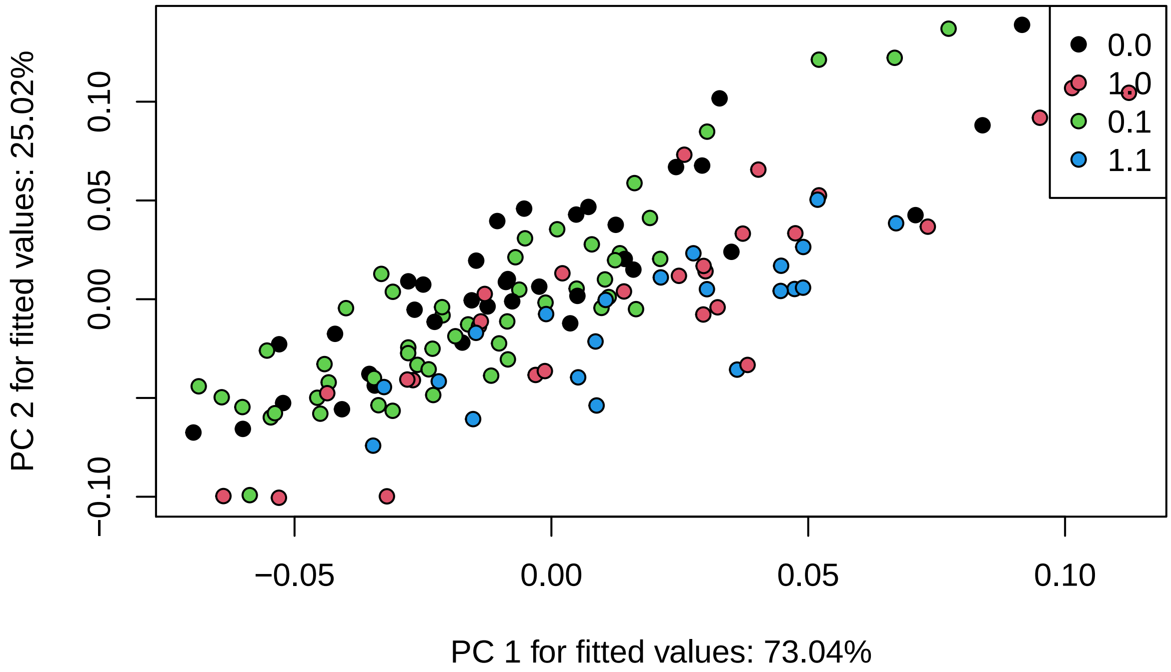
```
#Factorial MANOVA
model2<-lm(mydat$Y~mydat$X1*mydat$X2)
summary(manova(model2))
```

```
##                     Df  Pillai approx F num Df den Df    Pr(>F)
## mydat$X1            1 0.44815  17.1889      6    127 1.778e-14 ***
## mydat$X2            1 0.06874   1.5625      6    127    0.1634
## mydat$X1:mydat$X2   1 0.02965   0.6468      6    127    0.6926
## Residuals         132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Factorial MANOVA via RRPP
model2.rrpp <- lm.rrpp(mydat$Y~mydat$X1*mydat$X2,data = mydat, print.progress = FALSE)
anova(model2.rrpp)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type I
## Effect sizes (Z) based on F distributions
##
##                    Df      SS        MS     Rsq      F        Z Pr(>F)
## mydat$X1            1 0.02494 0.0249429 0.03607 5.0236  2.13464  0.016 *
## mydat$X2            1 0.00955 0.0095488 0.01381 1.9232  1.10454  0.141
## mydat$X1:mydat$X2   1 0.00160 0.0015977 0.00231 0.3218 -0.68919  0.749
## Residuals         132 0.65540 0.0049652 0.94781
## Total             135 0.69149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = mydat$Y ~ mydat$X1 * mydat$X2, data = mydat, print.progress = FALSE)
```

```
groups <- interaction(mydat$X1,mydat$X2)
plot(model2.rrpp, type = "PC", pch=21, bg = groups)
legend("topright", levels(groups), pch = 21, pt.bg = 1:4)
```



Multivariate Regression

To assess the multivariate Y data against X3 we use a multivariate regression using the base R and then again in RRPP. There is a significance correlation between our Y data and X3.

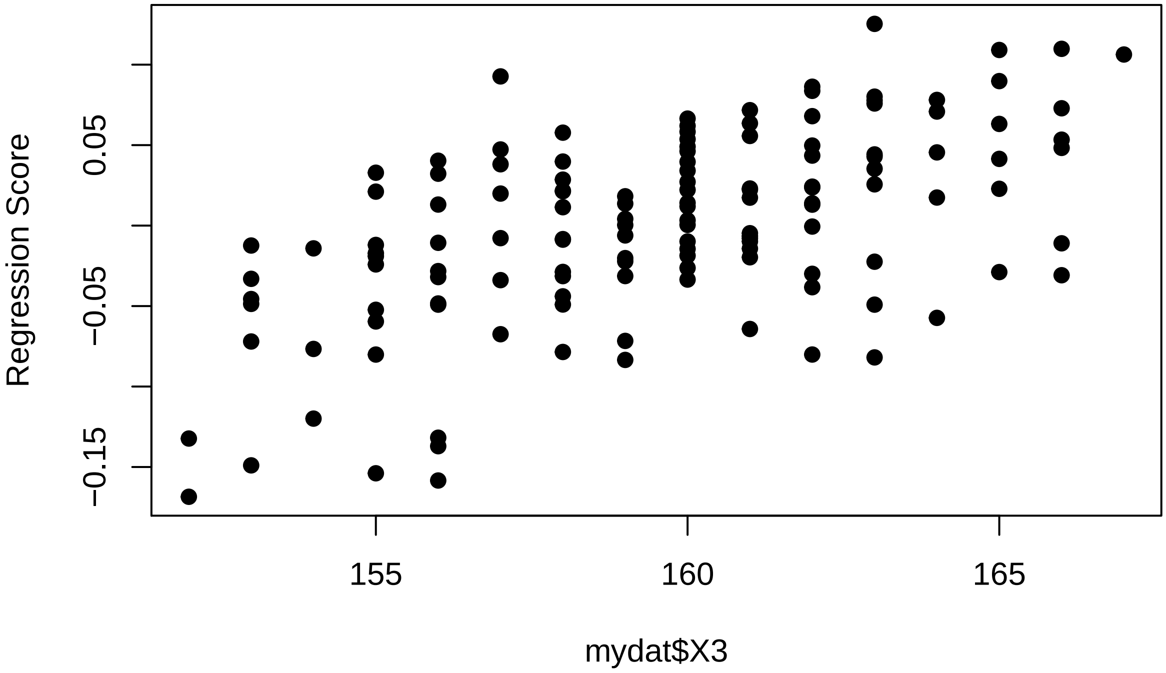
```
#______________________________________________________________________#
### Multivariate Regression
summary(manova(lm(mydat$Y~mydat$X3)))
```

```
##              Df  Pillai approx F num Df den Df    Pr(>F)    
## mydat$X3      1 0.51567   22.892      6    129 < 2.2e-16 ***
## Residuals 134                                               
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.reg <- lm.rrpp(mydat$Y~mydat$X3, data = mydat, print.progress = FALSE)
anova(model.reg)
```

```
## 
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals 
## Number of permutations: 1000 
## Estimation method: Ordinary Least Squares 
## Sums of Squares and Cross-products: Type I 
## Effect sizes (Z) based on F distributions
## 
##              Df      SS       MS     Rsq      F      Z Pr(>F)   
## mydat$X3      1 0.15206 0.152065 0.21991 37.775 4.6113  0.001 **
## Residuals 134 0.53943 0.004026 0.78009
## Total     135 0.69149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = mydat$Y ~ mydat$X3, data = mydat, print.progress = FALSE)
```

```
### Visualizing multivariate regression
plot(model.reg, type = "regression", reg.type = "RegScore",
     predictor = mydat$X3, pch=19)
```



MANCOVA

To assess the multivariate Y data against all three X columns (X1,X2, X3) we use a MANOVA (Multivariate Analysis of Variance) using the base R and MANCOVA (Multivariate Covariance Analysis of Variance) then again in RRPP. There is a significance correlation between our Y data and X1 and the Y data and X3.

```
summary(manova(lm(mydat$Y~ mydat$X1*mydat$X2*mydat$X3))) #no iteraction significant, just X1 and X3 sig
```

```
##                             Df  Pillai approx F num Df den Df Pr(>F)
## mydat$X1                     1 0.53847  23.9171      6    123 <2e-16 ***
## mydat$X2                     1 0.06933   1.5271      6    123 0.1747
## mydat$X3                     1 0.52166  22.3566      6    123 <2e-16 ***
## mydat$X1:mydat$X2            1 0.07557   1.6758      6    123 0.1324
## mydat$X1:mydat$X3            1 0.07297   1.6137      6    123 0.1488
## mydat$X2:mydat$X3            1 0.03431   0.7282      6    123 0.6277
## mydat$X1:mydat$X2:mydat$X3   1 0.01441   0.2997      6    123 0.9360
## Residuals                  128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

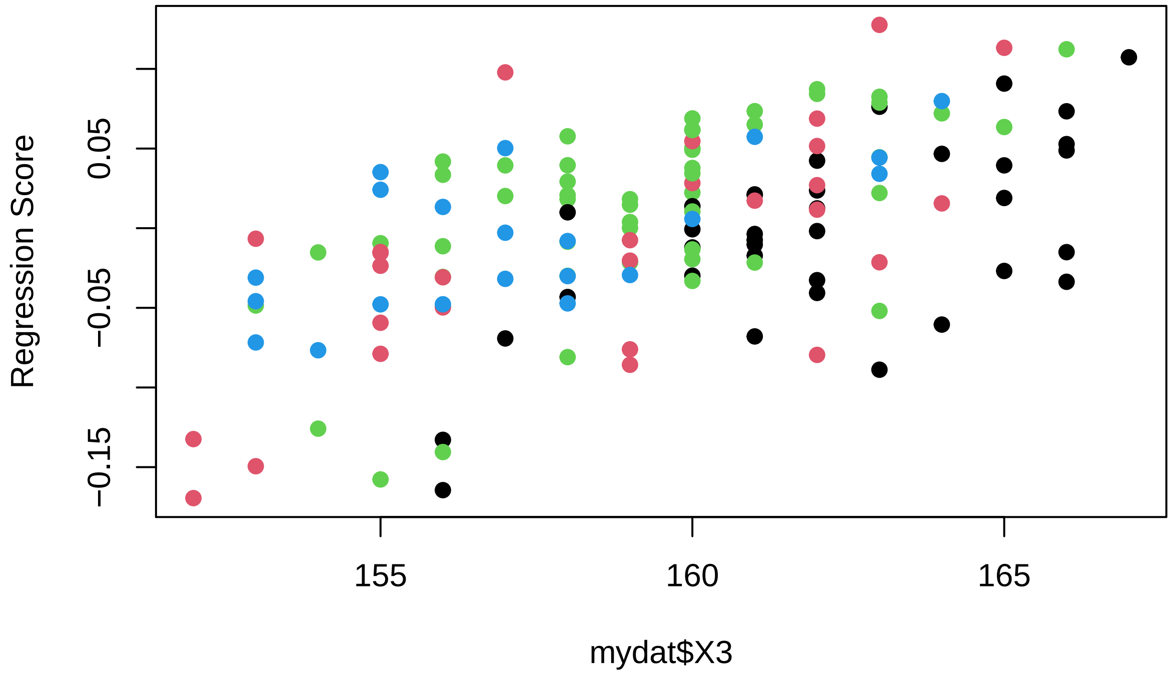
```
summary(manova(lm(mydat$Y~ mydat$X1+mydat$X3))) # FIT COMMON SLOPE
```

```
##             Df  Pillai approx F num Df den Df    Pr(>F)    
## mydat$X1    1 0.51929   23.045      6    128 < 2.2e-16 ***
## mydat$X3    1 0.45726   17.973      6    128 4.804e-15 ***
## Residuals 133                                              
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

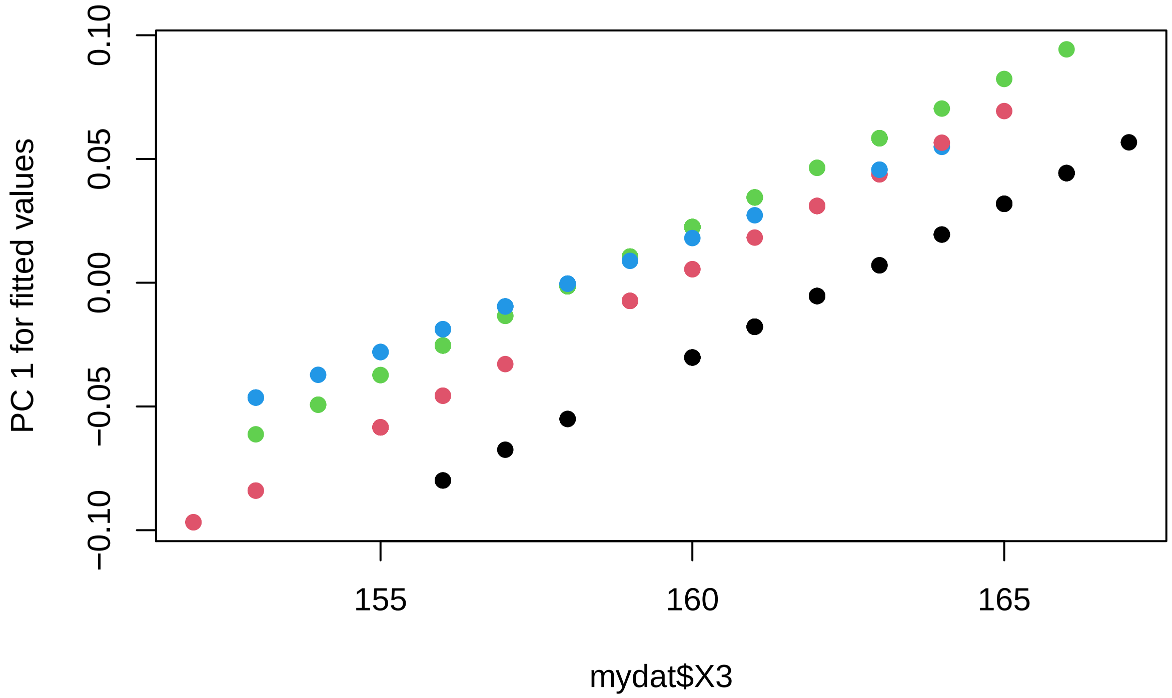
```
#MANCOVA via RRPP
model.mancova <- lm.rrpp(mydat$Y~ mydat$X1*mydat$X2*mydat$X3, data =mydat, print.progress = FALSE)
anova(model.mancova)
```

```
## 
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals 
## Number of permutations: 1000 
## Estimation method: Ordinary Least Squares 
## Sums of Squares and Cross-products: Type I 
## Effect sizes (Z) based on F distributions
## 
##                             Df      SS       MS     Rsq       F       Z Pr(>F)
## mydat$X1                     1 0.02494 0.024943 0.03607  6.9316  2.4835  0.005
## mydat$X2                     1 0.00955 0.009549 0.01381  2.6536  1.4403  0.086
## mydat$X3                     1 0.18310 0.183096 0.26478 50.8822  4.7686  0.001
## mydat$X1:mydat$X2            1 0.00782 0.007822 0.01131  2.1738  1.4143  0.088
## mydat$X1:mydat$X3            1 0.00250 0.002501 0.00362  0.6951 -0.0430  0.508
## mydat$X2:mydat$X3            1 0.00155 0.001550 0.00224  0.4307 -0.5844  0.722
## mydat$X1:mydat$X2:mydat$X3   1 0.00143 0.001429 0.00207  0.3972 -0.6541  0.746
## Residuals                  128 0.46060 0.003598 0.66610                       
## Total                      135 0.69149                                        
##                              
## mydat$X1                   **
## mydat$X2                   . 
## mydat$X3                   **
## mydat$X1:mydat$X2          . 
## mydat$X1:mydat$X3            
## mydat$X2:mydat$X3            
## mydat$X1:mydat$X2:mydat$X3   
## Residuals                    
## Total                        
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Call: lm.rrpp(f1 = mydat$Y ~ mydat$X1 * mydat$X2 * mydat$X3, data = mydat, 
##     print.progress = FALSE)
```

```
### Visualizing MANCOVA
plot(model.mancova, type = "regression", reg.type = "RegScore",
     predictor = mydat$X3, pch=19, col = as.numeric(groups))
```



```
plot(model.mancova, type = "regression", reg.type = "PredLine",
     predictor = mydat$X3, pch=19,
     col = as.numeric(groups))
```



WEEK 8 MATERIAL

Principal components analysis (PCA) of the Y data

```
Y <- scale(mydat$Y, scale = FALSE) #center data
pca.dat1<-prcomp(Y) #told nothing about groups
summary(pca.dat1)
```

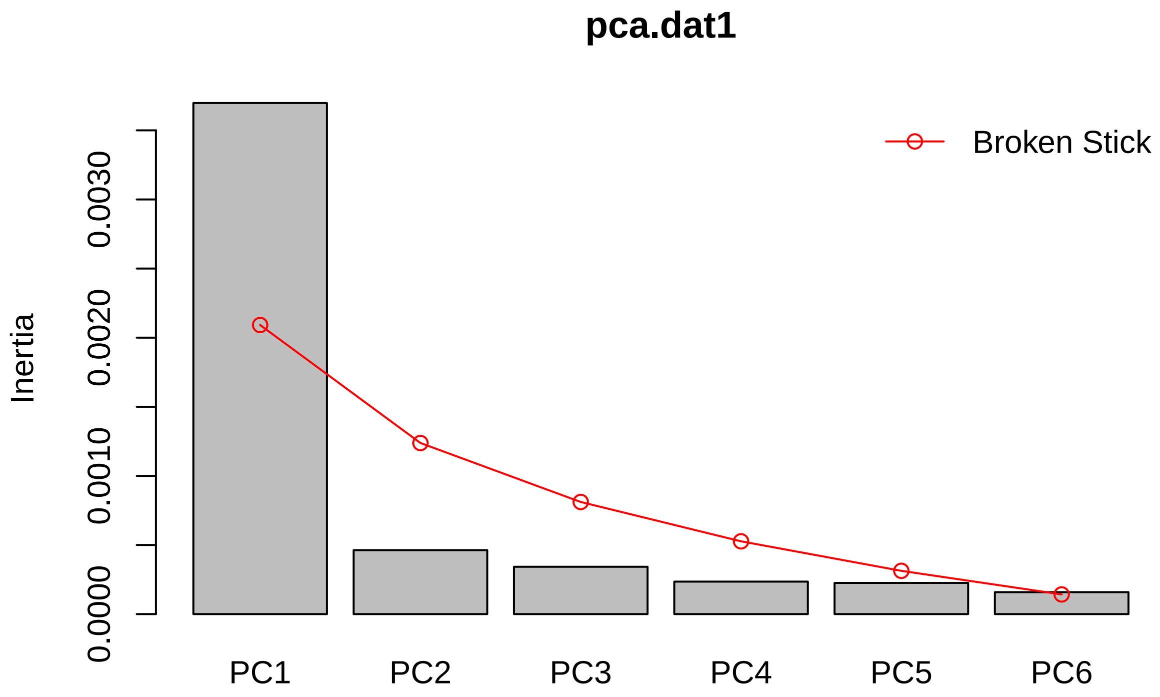
```
## Importance of components:
##                            PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     0.06081 0.02151 0.01850 0.01533 0.01502 0.01260
## Proportion of Variance 0.72190 0.09036 0.06683 0.04587 0.04403 0.03102
## Cumulative Proportion  0.72190 0.81225 0.87908 0.92495 0.96898 1.00000
```

Plot the PCA of the Y data using the vegan package. The broken stick plot shows us that PC1 accounts for the majority of the variance in our data.

```
library(vegan)
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-7
```

```
screeplot(pca.dat1,bstick = TRUE)
```



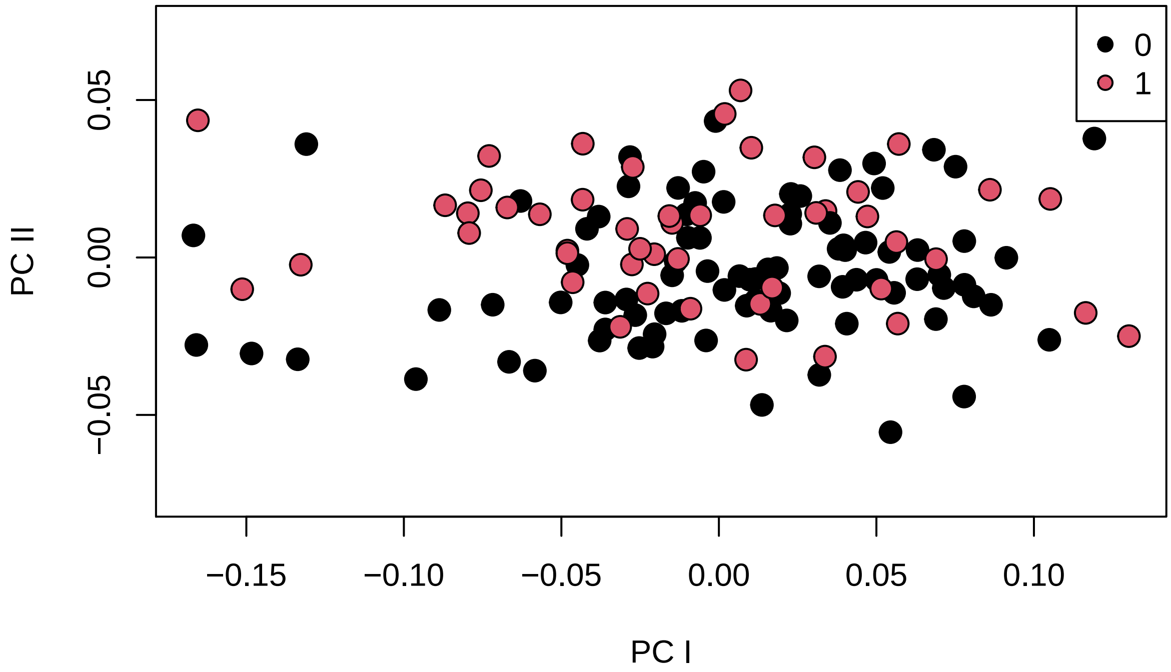
To look at what compromises the PC1 variance we look at the rotation. The values that are farther away form 0 are more important for the PC.

```
pca.dat1$rotation[,1] #PC1
```

```
##        Y1        Y2        Y3        Y4        Y5        Y6
## 0.2620346 0.2675598 0.4778146 0.5212309 0.5411779 0.2586263
```

Plot PC.scores and color code based on the "levels of X1"

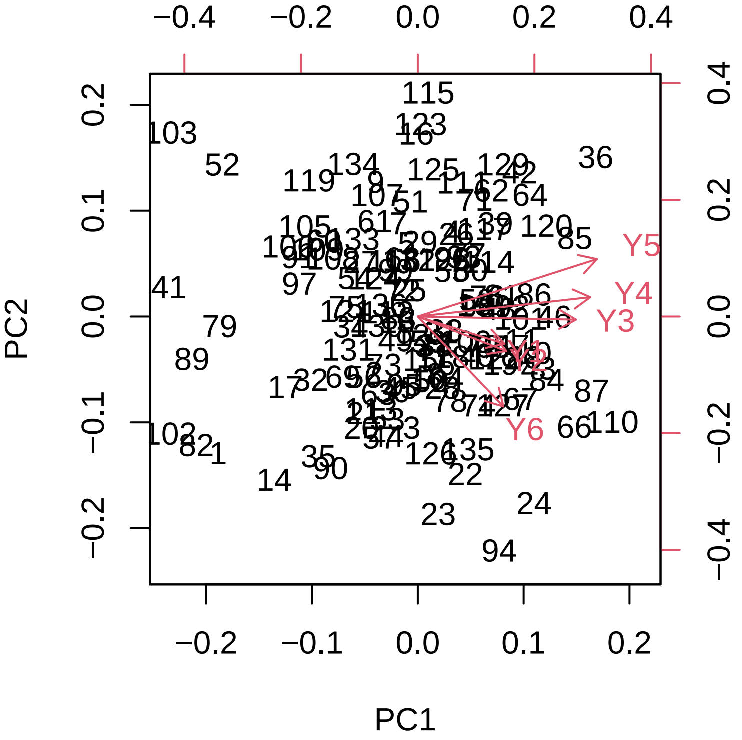
```
PC.scores<-pca.dat1$x

plot(PC.scores,xlab="PC I", ylab="PC II",asp=1,pch=21,bg=mydat$X1,cex = 1.5)
legend("topright", levels(mydat$X1), pch = 21,pt.bg=1:2)
```



Biplot

To superimpose vectors for the variables in the PCA space we do a biplot of pca.dat1.The length of the vectors indicate the importance of variance from that Y. The Y3, Y4 and Y5 appear to be marginally longer although, all Y's appear to be approximately the same length and therefore importance to teh variance.

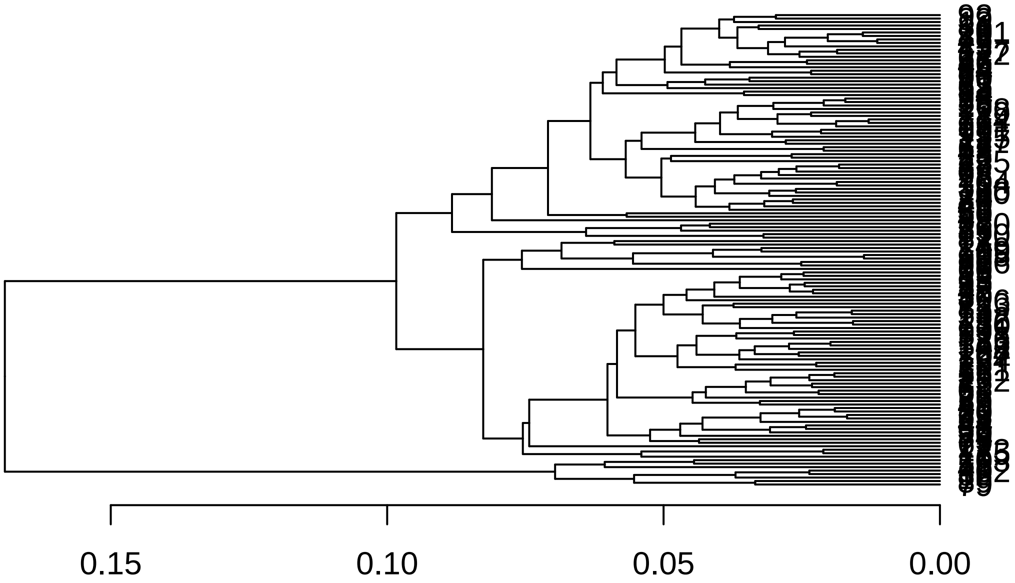
```r
#Biplot of dat1
biplot(pca.dat1)
```



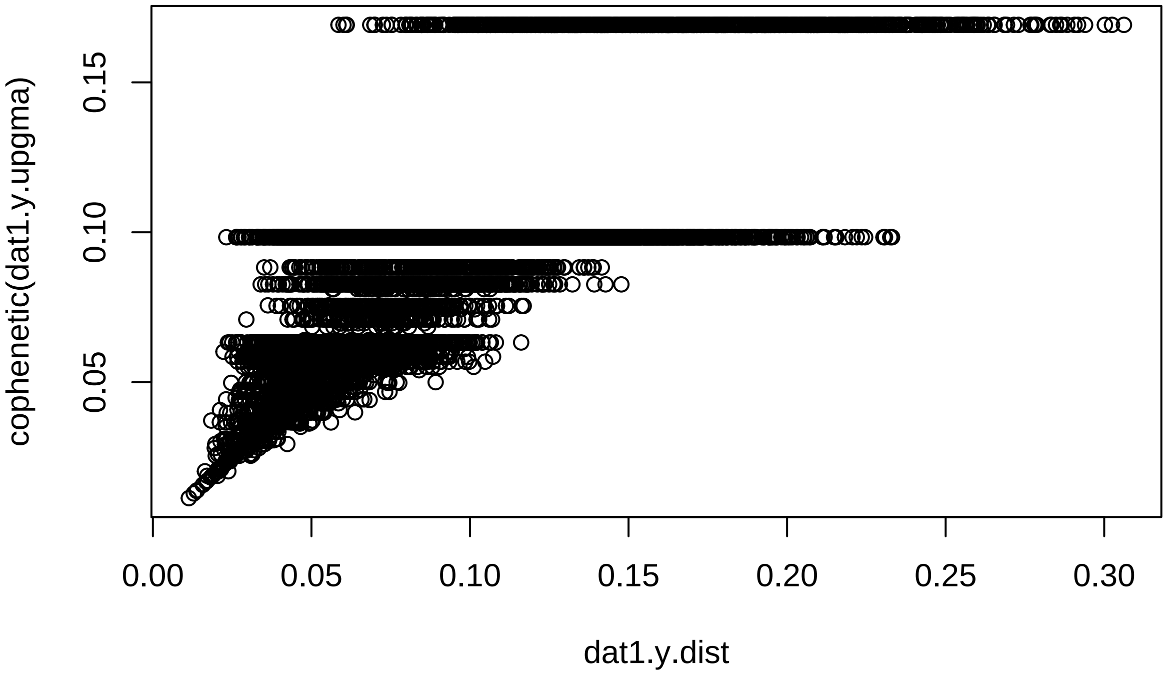
WEEK 9 MATERIAL

To cluster using UPGMA we first make a distance matrix (default is euclidean) of the PC scores and then perform a hierarchical cluster analysis using hclust()

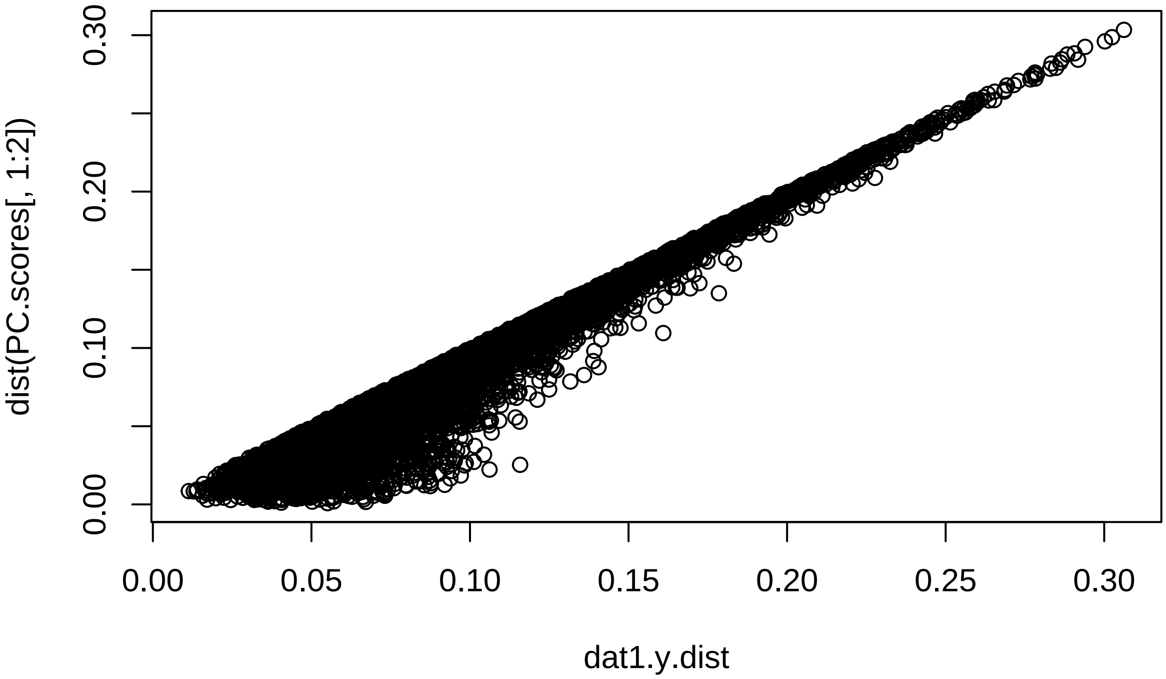
```
##UPGMA
dat1.y.dist<-dist(PC.scores)
dat1.y.upgma<-hclust(dat1.y.dist,method="average")
plot(as.dendrogram(dat1.y.upgma),horiz=TRUE,lwd=4)   #UPGMA
```



```
#PLOT of actual vs. UPGMA distances
plot(dat1.y.dist,cophenetic(dat1.y.upgma))
```



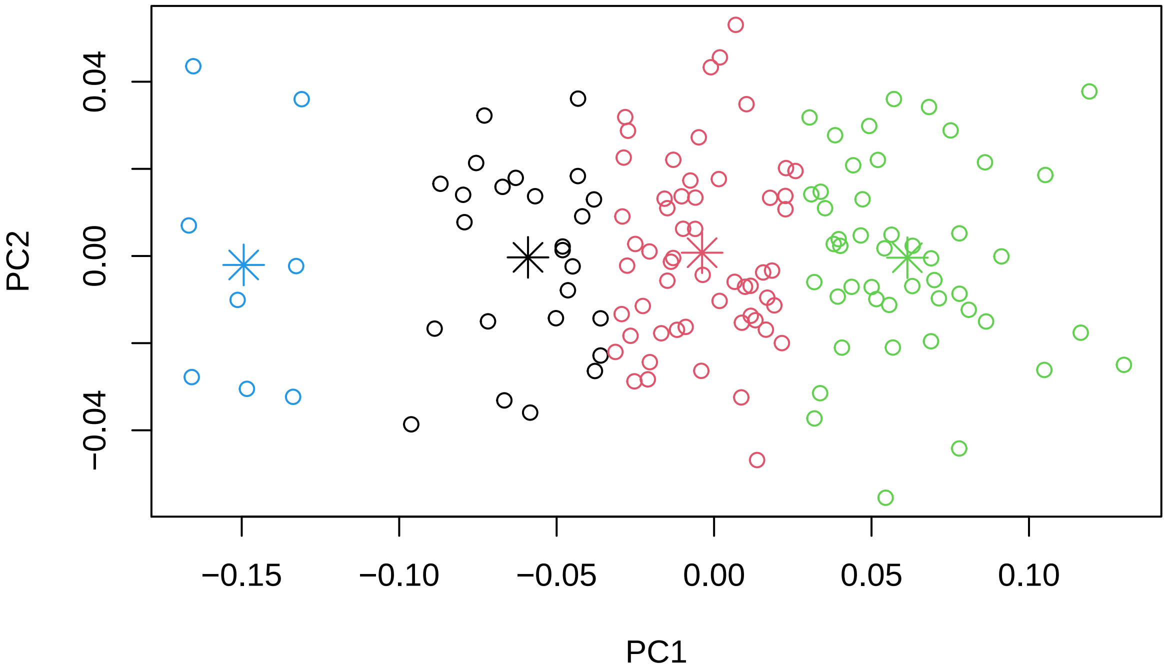
```
# SAME from PC
plot(dat1.y.dist,dist(PC.scores[,1:2]))
```



K-MEANS CLUSTERING METHODS

To cluster using non-hierarchical method of identifying groups #### Clustering by 4 Clustering under the assumption of 4 groups (k=4).

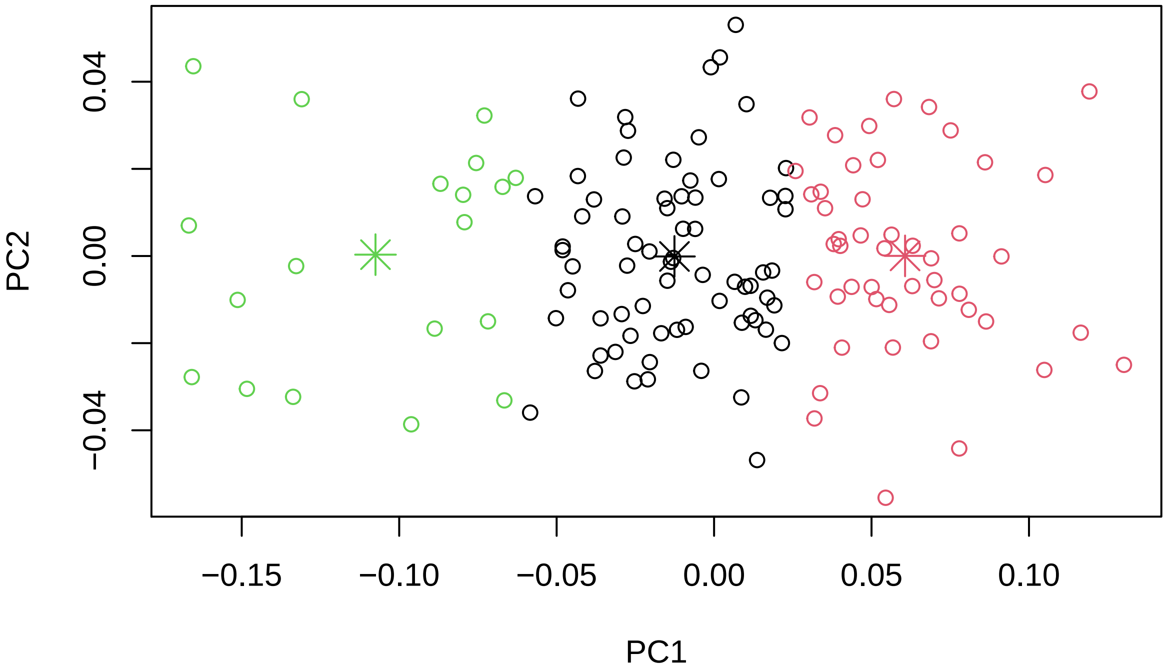
```
#K-means = 4
kclusters4<-kmeans(PC.scores,4)
plot(PC.scores[,1:2],col=kclusters4$cluster)
points(kclusters4$centers, col = 1:4, pch = 8, cex=2)
```



Clustering by 3

Clustering under the assumption of 3 groups (k=3).

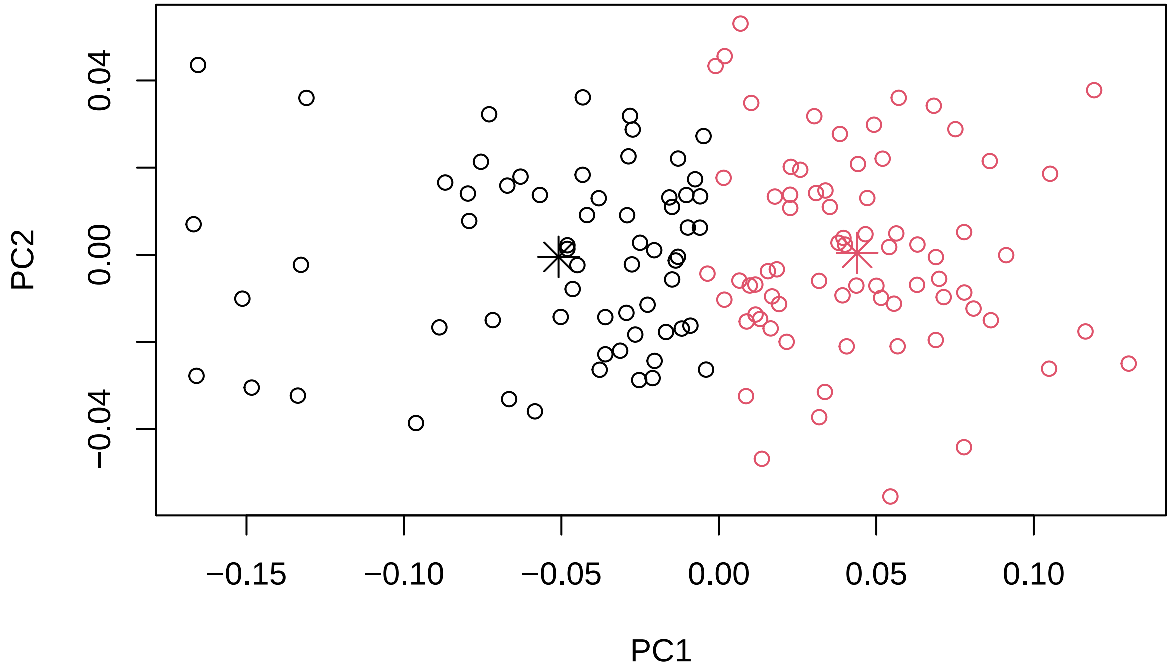
```
#K-means = 3
kclusters3<-kmeans(PC.scores,3)
plot(PC.scores[,1:2],col=kclusters3$cluster)
points(kclusters3$centers, col = 1:3, pch = 8, cex=2)
```



Clustering by 2

Clustering under the assumption of 2 groups (k=2).

```r
#K-means = 2
kclusters2<-kmeans(PC.scores,2)
plot(PC.scores[,1:2],col=kclusters2$cluster)
points(kclusters2$centers, col = 1:2, pch = 8, cex=2)
```

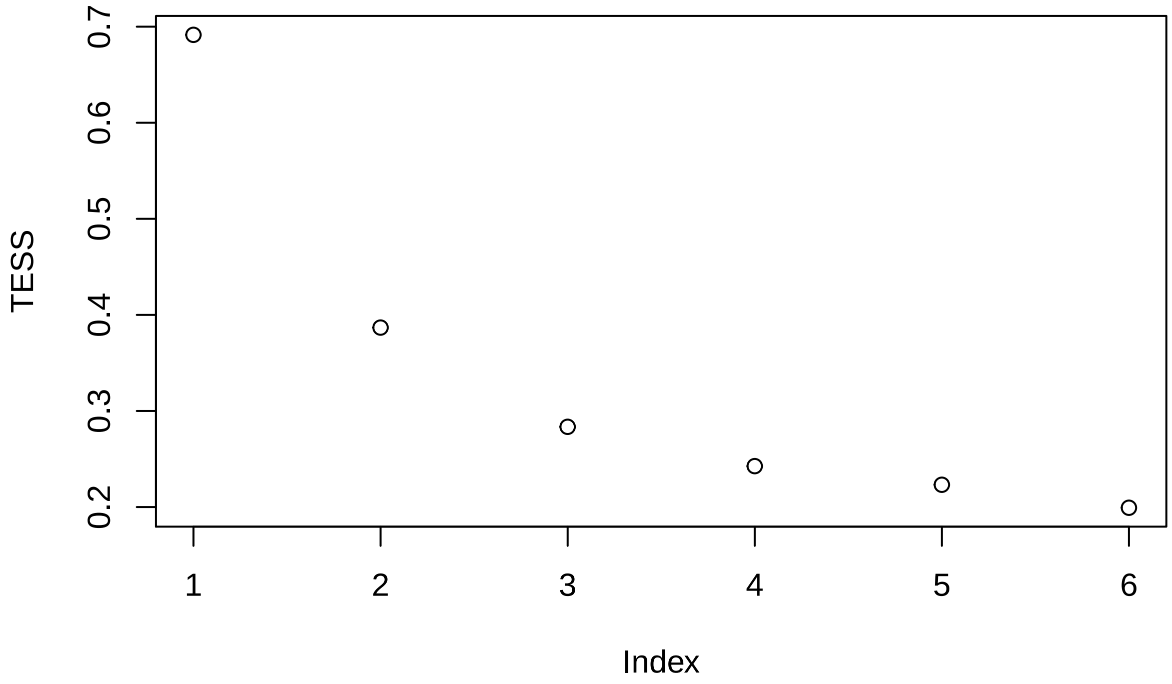


```
#NOTE: repeating k-means at a given level can lead to differing results
```

TESS: total error sums-of-squares

Compare the total error sums-of-squares to see which grouping results in a leveling off of the kmeans of PC scores.

```
#compare TESS
TESS<-array(NA,6)
for (i in 1:6){
  TESS[i]<-kmeans(PC.scores,i)$tot.withinss
}
plot( TESS)  #seems to bottom out at 3 groups
```



Based on the TESS results, it appears that the mean PC.scores level off at about a k grouping of 3 or 4, but this appears to be very gradual and the groups are not clearly defined. We would argue that there is only really one group (k=1).

WEEK 10 MATERIAL

Partial Least Squares (PLS)

PLS is to summarize covariation between columns. We chose to look at the covariation of Y4 and Y5 because the PC1 choses these as teh columns with the most variation. Comparing the covarion of the two columns we get a pvalue of 0.001.

```
pls.res<-two.b.pls(mydat$Y[,4], mydat$Y[,5],print.progress = FALSE)
```

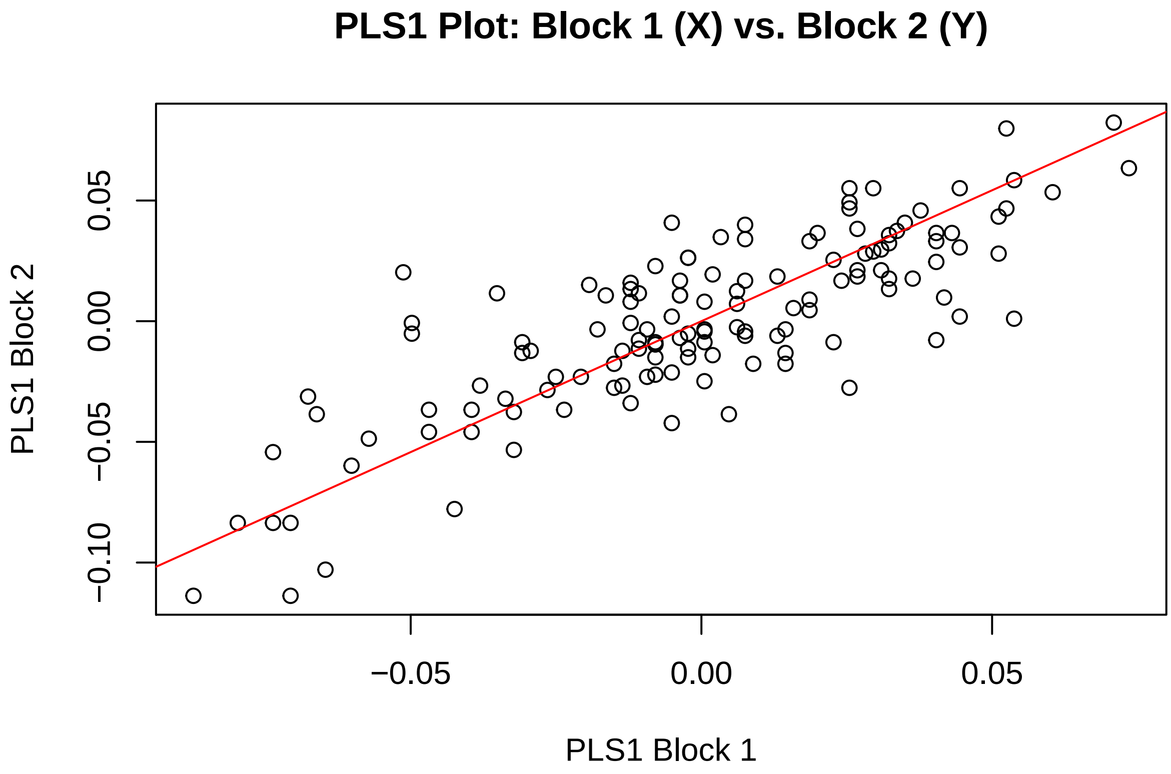
```
## Data in either A1 or A2 do not have names.  It is assumed data in both A1 and A2 are ordered the sam
```

```
summary(pls.res)
```

```
##
## Call:
## two.b.pls(A1 = mydat$Y[, 4], A2 = mydat$Y[, 5], print.progress = FALSE)
##
##
##
## r-PLS: 0.811
##
## Effect Size (Z): 8.9198
##
```

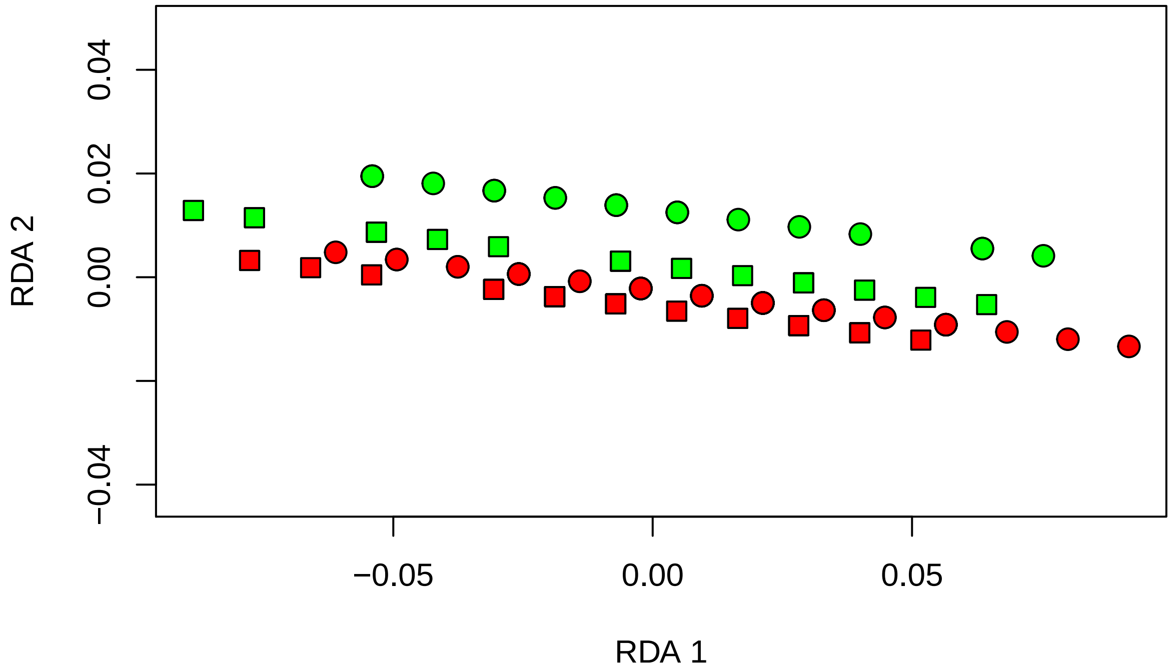
```
## P-value: 0.001
##
## Based on 1000 random permutations
```

```
plot(pls.res)
```



Redundancy Analysis

```
Y<-pca.dat1$x
col.gp<-rep("green",nrow(Y));   col.gp[which(mydat$X1== '0')]<-"red"
shape.gp<-rep(21,nrow(Y));   shape.gp[which(mydat$X2== '0')]<-22
rda.dat1<-rda(Y~mydat$X1+mydat$X2+mydat$X3+mydat$X1*mydat$X2)
rda.scores<-predict(rda.dat1)
plot(rda.scores,pch=shape.gp,bg=col.gp,asp=1,cex=1.5,xlab="RDA 1", ylab="RDA 2")
```


RDA 2
0.04
0.02
0.00
−0.04
−0.05
0.00
0.05
RDA 1