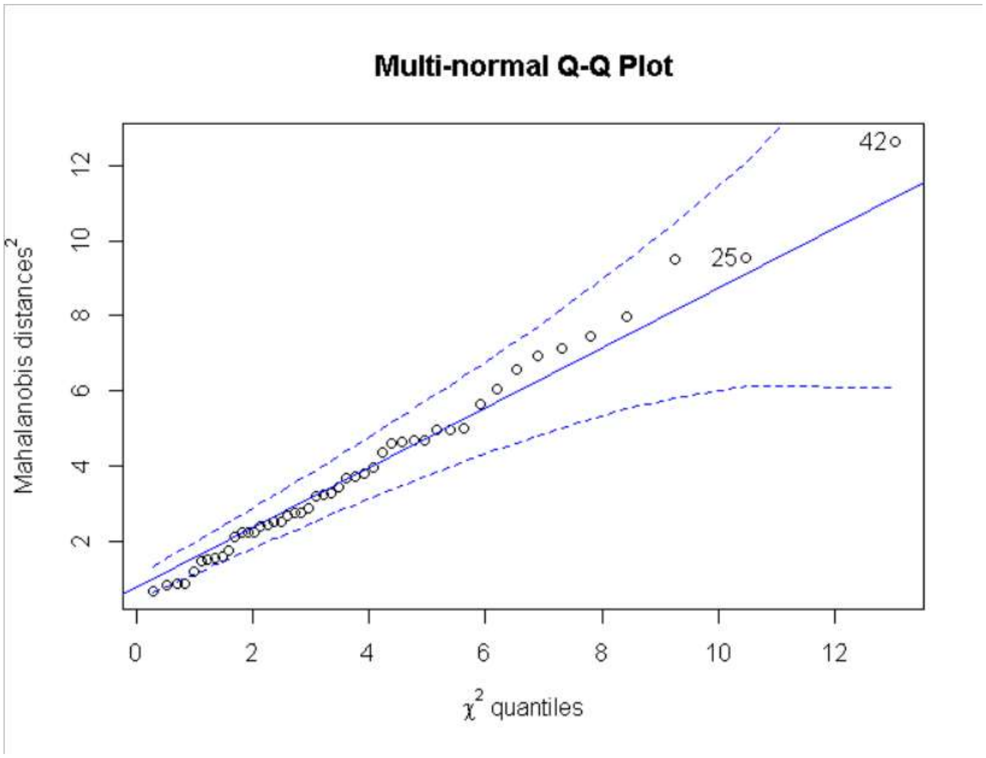
Test 2

Problem 1

For this dataset we did a MANOVA test to test if the mean vectors of all 4 variables were the same for the rootstocks. This yielded a Wilks lambda of 0.154 and a p-value of 7.7e-9 indicating that the means are not the same. For completeness I did a qq plot of the residuals from our model producing:

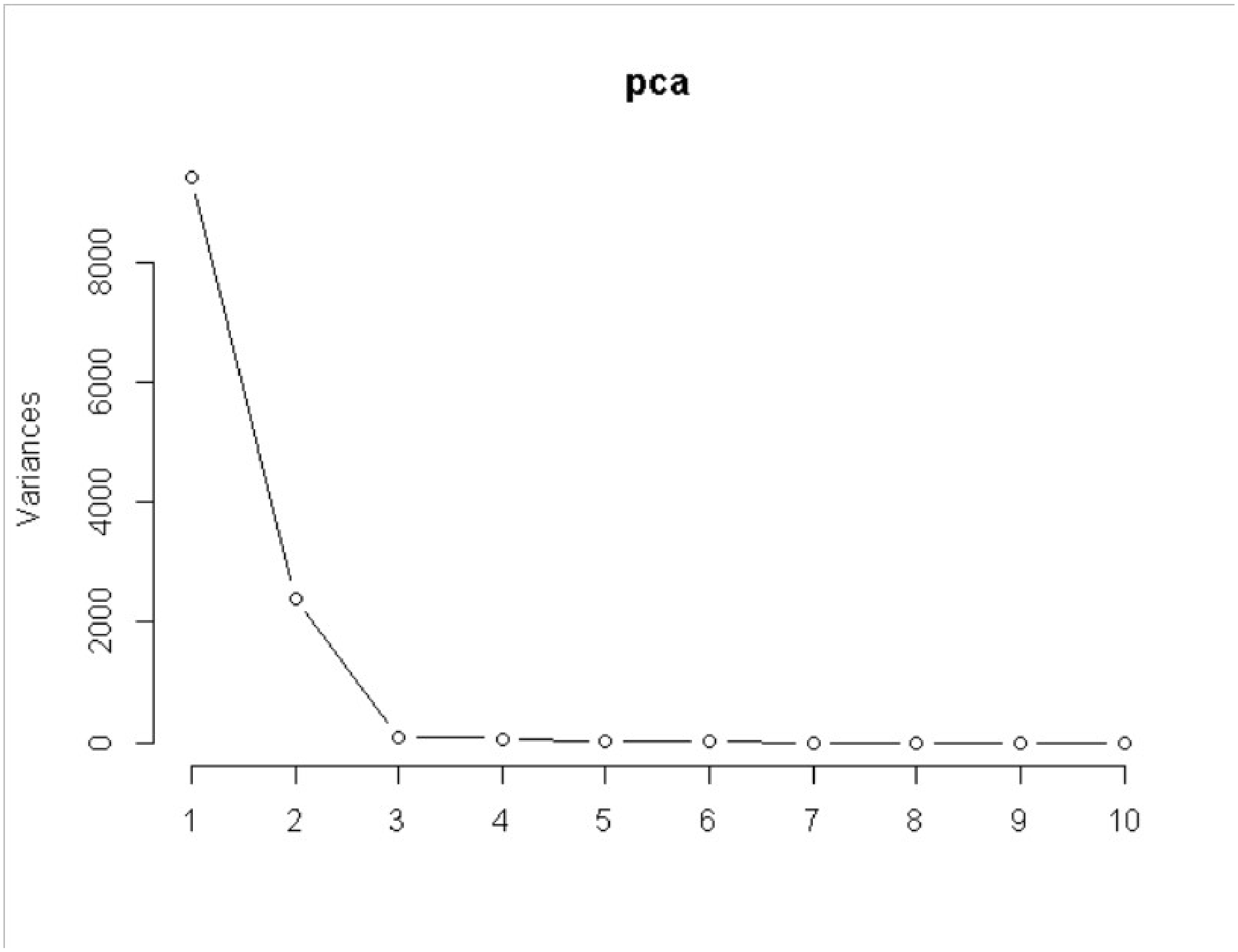


Given the small data size the dots to the right that do deviate aren't significant enough to discredit a multivariate normal distribution of the residuals. To further confirm that we can use the MANOVA we tested that the variances are all from the same distribution I ran an M-box test that resulted in a p-value of 0.711, meaning we do expect these to have the same variances.

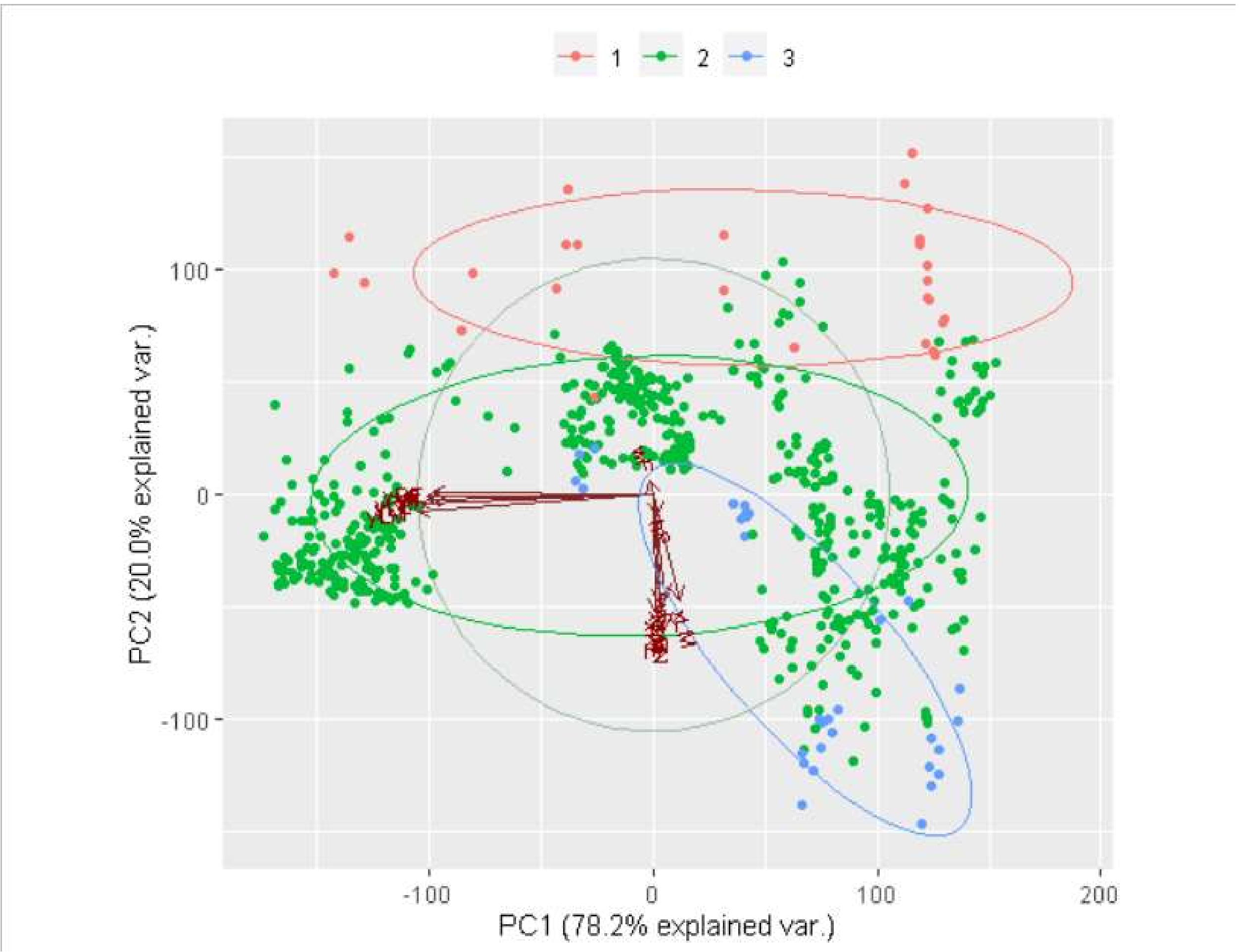
From here we ran a bootstrapping protocol in hopes of further supporting our previous tests. This provided a bootstrap lambda of 0, further supporting that these are not from the same mean distribution.

Problem 2

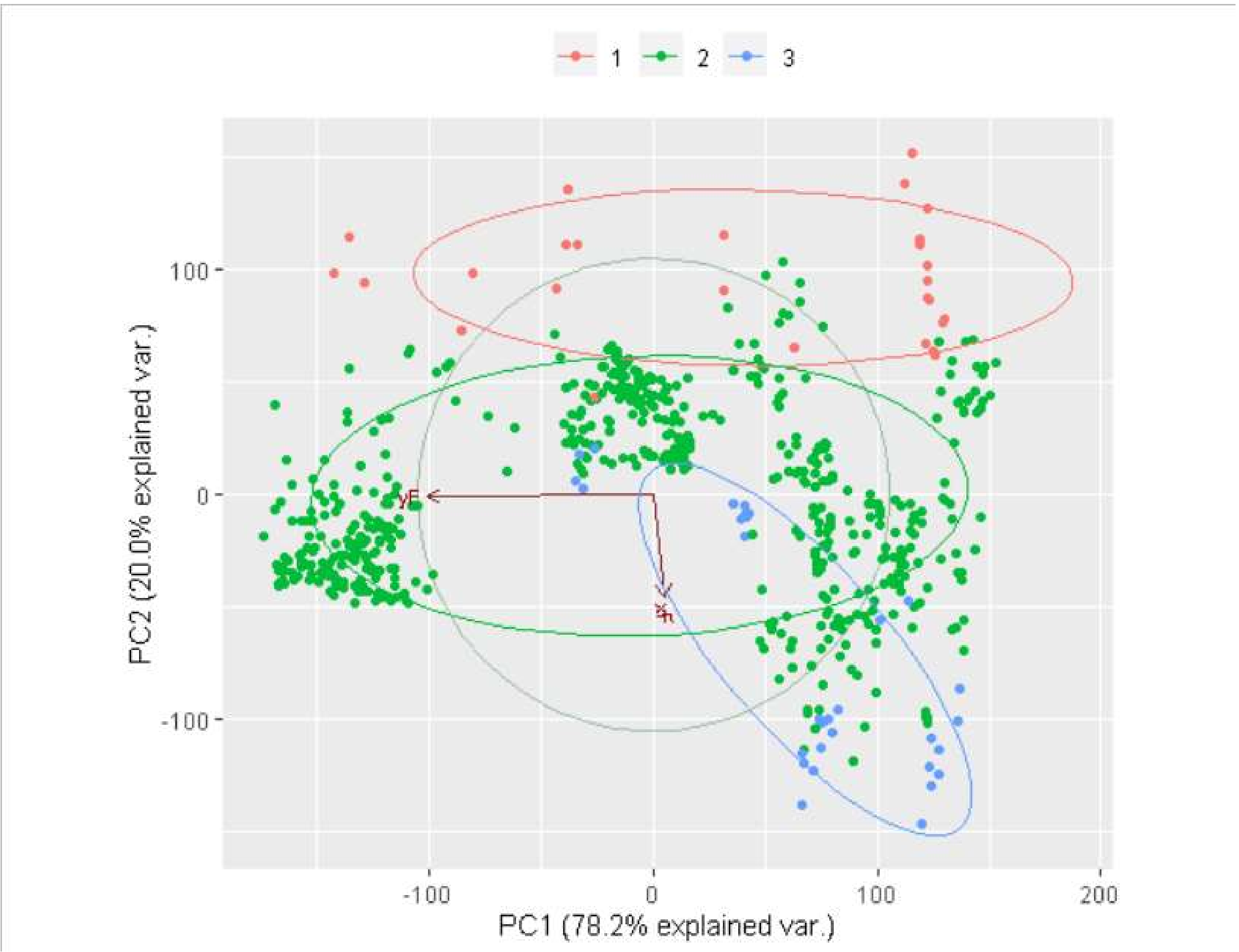
Using Principle Component Analysis (PCA) to predict variable 4 we get the following graph that shows how much more accurate we get with each additional PCA vector:



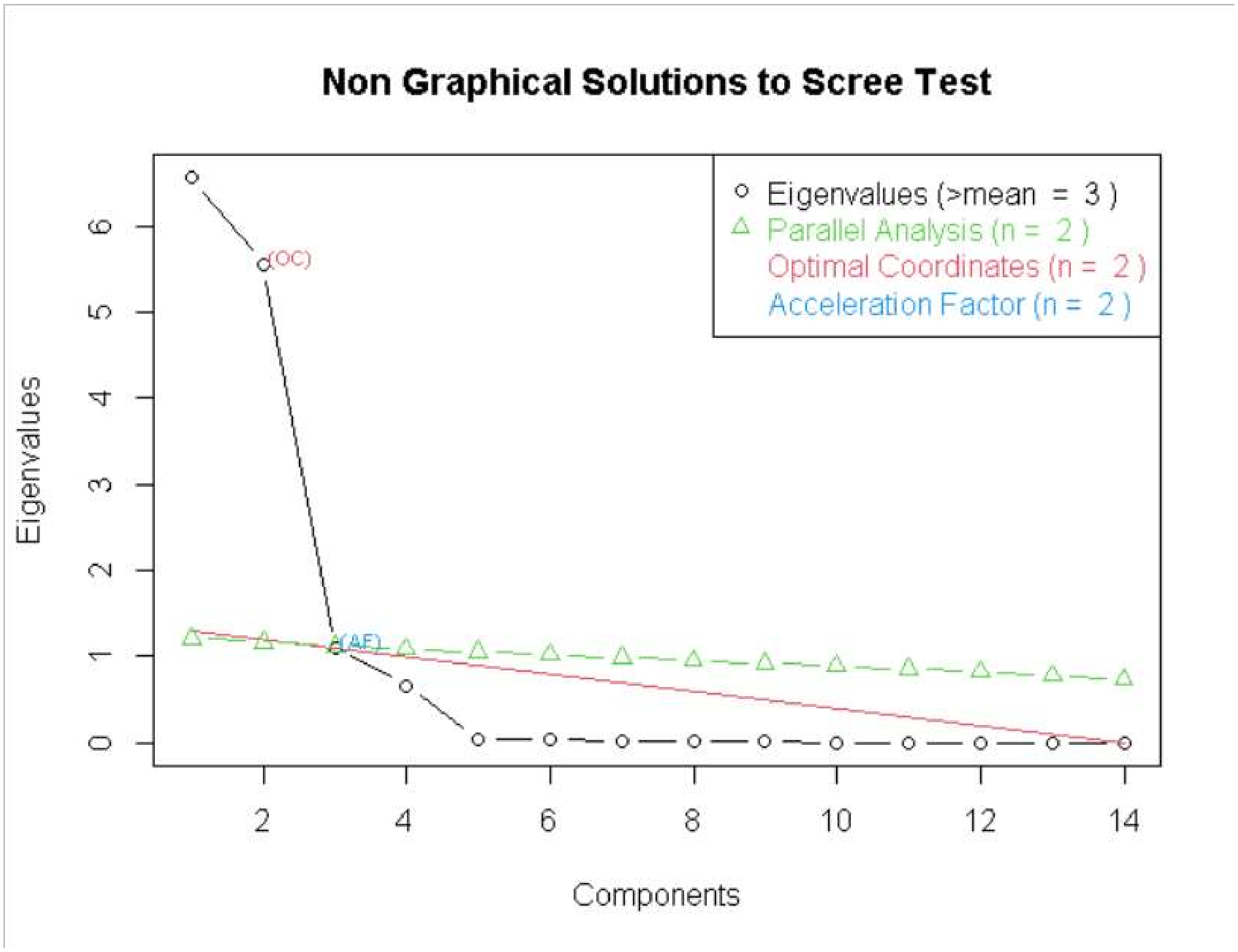
This suggest we should be using 2 vectors as the amount of information we get in additional vectors is near zero. Plotting every datapoint using the two principle components we get a graph where, by looking at the x and y axis labels, we can see 98.2% of the variance is explained with the two principle components:



In this graph each color corresponds to a value in variable 4, the ovals are where a majority of the points can be found for each value, and the dark red vectors are each principle component. Comparing the above graph to the below graph which has only the first two principle components graphed (the two we decided we needed) we see nearly every other principle component is mapped virtually on top of the first two:



While PCA can be used to simplify the model and make prediction as a result, Factor Analysis can show us hidden connections we don't realize are occurring. Thus, we analyzed the data using this method as well. Running the analysis with 9 factors, the max possible for the number of variables in the data set, we see that after factor 4 we do not have another factor with a value greater than 0.2. I re-ran the data with only 4 factors given the addition information I had gained running it with 9, this created an analysis where factor 1 and 2 were clearly very important; however, factor 3 and 4 both had at least one component at .842 indicated these were still important factors. Using the following graph we verified that 4 factors was indeed the best option.


Non Graphical Solutions to Scree Test
Eigenvalues (>mean = 3)
Parallel Analysis (n = 2)
Optimal Coordinates (n = 2)
Acceleration Factor (n = 2)
(OC)
(AF)
Eigenvalues
0
1
2
3
4
5
6
Components
2
4
6
8
10
12
14

Appendix: Code

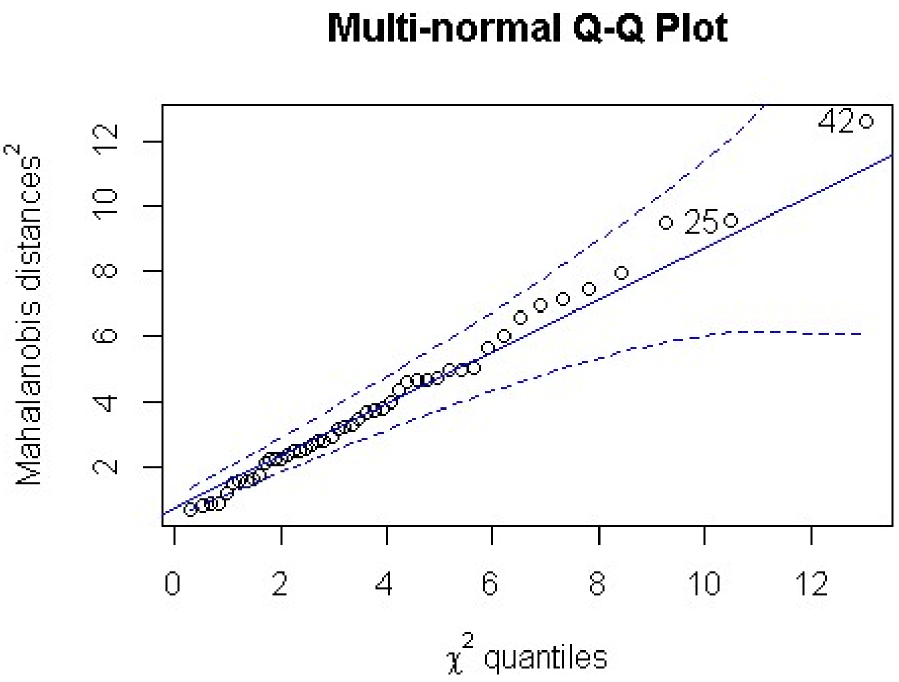
```
library(mvtnorm)
library(RVAideMemoire)

## *** Package RVAideMemoire v 0.9-77 ***

library(rstatix)

##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter

library(mnormt)
library(nFactors)

## Loading required package: lattice

##
## Attaching package: 'nFactors'

## The following object is masked from 'package:lattice':
##
##     parallel

setwd('C:\\Users\\User\\Desktop\\School\\Math_537\\Test2')
data=read.table('Apple.txt', header=T)

head(data)

##   Rootstock   y1    y2   y3    y4
## 1         1 1.11 2.569 3.58 0.760
## 2         1 1.19 2.928 3.75 0.821
## 3         1 1.09 2.865 3.93 0.928
## 4         1 1.25 3.844 3.94 1.009
## 5         1 1.11 3.027 3.60 0.766
## 6         1 1.08 2.336 3.51 0.726

######################################################
#1 a)

y=cbind(data$y1,data$y2,data$y3,data$y4)
model=manova(y~as.factor(Rootstock),data=data)
lam=summary.manova(model,test="Wilks")$stats[3]
lam

## [1] 0.1540077

######################################################
#1 b)
```

```
temp=summary.manova(model,test="Wilks")
temp$SS$Residuals
```

```
##           [,1]      [,2]      [,3]     [,4]
## [1,] 0.3199875  1.696564 0.5540875 0.217140
## [2,] 1.6965637 12.142790 4.3636125 2.110214
## [3,] 0.5540875  4.363612 4.2908125 2.481656
## [4,] 0.2171400  2.110214 2.4816562 1.722525
```

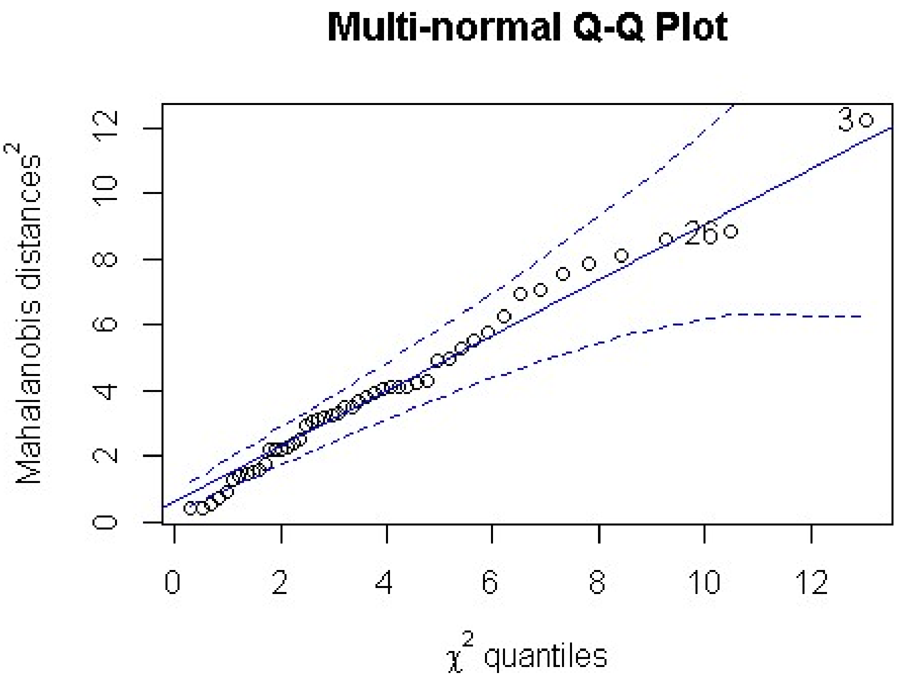
```
model$residuals
```

```
##        [,1]      [,2]     [,3]      [,4]
## 1  -0.02750 -0.408125 -0.15875 -0.111125
## 2   0.05250 -0.049125  0.01125 -0.050125
## 3  -0.04750 -0.112125  0.19125  0.056875
## 4   0.11250  0.866875  0.20125  0.137875
## 5  -0.02750  0.049875 -0.13875 -0.105125
## 6  -0.05750 -0.641125 -0.22875 -0.145125
## 7  -0.02750  0.233875  0.24125  0.337875
## 8   0.02250  0.059875 -0.11875 -0.121125
## 9  -0.10750 -1.035125 -0.42500 -0.244500
## 10  0.01250 -0.224125 -0.45500 -0.186500
## 11 -0.04750  0.268875  0.35500  0.354500
## 12  0.09250  0.796875  0.46500  0.236500
## 13  0.01250 -0.327125 -0.13500 -0.083500
## 14 -0.00750 -0.091125  0.13500 -0.036500
## 15  0.01250  0.273875  0.17500  0.214500
## 16  0.03250  0.337875 -0.11500 -0.254500
## 17 -0.03750 -0.310250 -0.69500 -0.479375
## 18 -0.11750 -0.500250 -0.01500  0.006625
## 19 -0.04750 -0.148250 -0.07500 -0.194375
## 20 -0.08750 -0.425250  0.21500  0.221625
## 21  0.04250  0.205750  0.02500  0.084625
## 22  0.09250  0.269750  0.32500  0.179625
## 23  0.09250  0.492750  0.11500  0.114625
## 24  0.06250  0.415750  0.10500  0.066625
## 25  0.12250 -0.041750 -0.01625 -0.095000
## 26 -0.06750 -0.528750  0.14375  0.202000
## 27  0.04250  0.121250  0.14375 -0.016000
## 28 -0.08750 -0.440750  0.01375  0.028000
## 29 -0.10750 -0.680750 -0.63625 -0.346000
## 30  0.01250  0.438250  0.04375  0.046000
## 31  0.10250  0.721250  0.36375  0.203000
## 32 -0.01750  0.411250 -0.05625 -0.022000
## 33 -0.17000 -1.025250 -0.27250 -0.097000
## 34  0.07000 -0.005250 -0.15250 -0.030000
## 35  0.06000  0.525750  0.47750  0.200000
## 36 -0.03000 -0.227250  0.10750  0.061000
## 37 -0.09000 -0.478250 -0.84250 -0.508000
## 38  0.14000  0.808750  0.09750 -0.044000
## 39 -0.03000 -0.141250  0.32750  0.274000
## 40  0.05000  0.542750  0.25750  0.144000
## 41  0.07375  0.598375  0.16375  0.065000
## 42 -0.28625 -1.374625 -0.45625 -0.129000
## 43  0.01375 -0.015625  0.15375  0.055000
## 44 -0.01625 -0.082625  0.39375  0.118000
## 45  0.01375 -0.265625 -0.25625 -0.125000
## 46  0.03375  0.036375 -0.38625 -0.173000
## 47  0.09375  0.849375  0.03375 -0.028000
## 48  0.07375  0.254375  0.35375  0.217000

mqqnorm(model$residuals)
```



```
## [1] 42 25

#fairly normal


box_m(y, data$Rootstock)

## # A tibble: 1 x 4
##   statistic p.value parameter method
##       <dbl>   <dbl>     <dbl> <chr>
## 1      44.0   0.711        50 Box's M-test for Homogeneity of Covariance M
atric~
```

```
# With a p-value of .711 this implies that each y_i came from similar varianc
es.


######################################################
#1 c)

n=10000
lam2=matrix(0,1,n)

data2=as.matrix(data)

ind=sample(1:48,replace=T)
final=cbind((data2[,1]),(data2[ind,2:5]))
results= final[,2:5]
y=cbind(final[,2],final[,3],final[,4],final[,5])
model=manova(y~as.factor(final[,1]),data=as.data.frame(final))
lam2[1]=summary.manova(model,test="Wilks")$stats[3]

for(i in 2:n)
{
  ind=sample(1:48,replace=T)
  final=cbind((data2[,1]),(data2[ind,2:5]))
  results=results+final[,2:5]
  y=cbind(final[,2],final[,3],final[,4],final[,5])
  model=manova(y~as.factor(final[,1]),data=as.data.frame(final))
  lam2[i]=summary.manova(model,test="Wilks")$stats[3]
}

sum(lam2<lam)/n

## [1] 0

results=results/n

results=cbind((data2[,1]),results)
y=cbind(results[,2],results[,3],results[,4],results[,5])
model=manova(y~as.factor(results[,1]),data=as.data.frame(results))
mqqnorm(model$residuals)
```

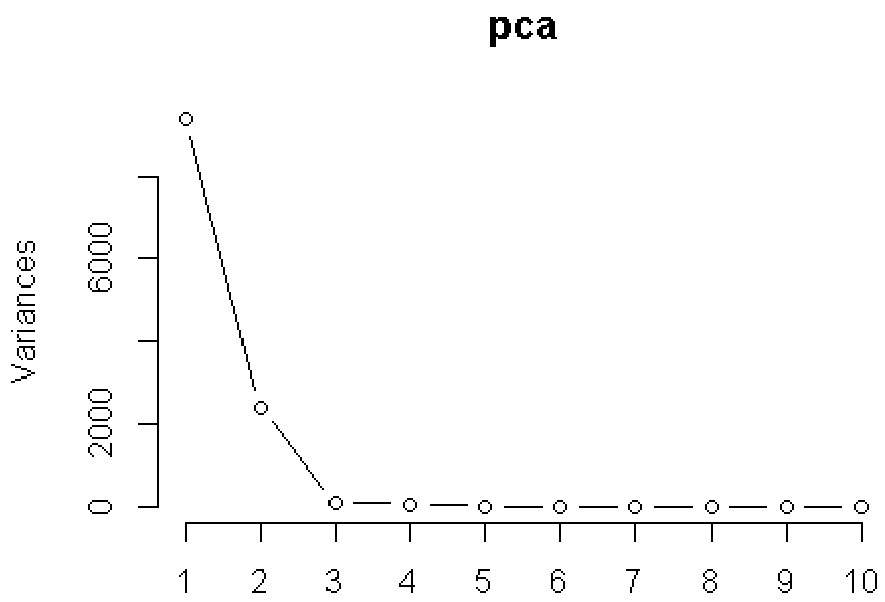


```
## [1]  3 26

#########################################################
#2 a)

data2=read.table('drivPoints.txt', sep=",", header=T)

data2F=scale(data2[,6:dim(data2)[2]],scale=F)

pca = prcomp(data2F)
pca

## Standard deviations (1, .., p=14):
##  [1] 96.989608 49.020727  9.624131  8.328001  4.206615  3.476437  2.617391
##  [8]  2.510391  1.937855  1.734361  1.621667  1.489140  1.224986  1.037106
##
## Rotation (n x k) = (14 x 14):
##              PC1          PC2         PC3          PC4          PC5
PC6
## xF   0.022139210 -0.364691878  0.25895493 -0.548645238  0.039618288 -0.167
28177
## yF  -0.405234308 -0.004778740 -0.26409687 -0.063828886 -0.024108922 -0.212
34173
## wF  -0.005415327  0.048131327  0.14303311  0.767759768  0.181515296 -0.236
89076
## hF   0.009165877 -0.073678717  0.77630615  0.104973303 -0.042171033  0.429
89565
```

```
## xRE  0.004989021 -0.416446184 -0.10826090  0.018492764  0.251632361  0.229
53706
## yRE -0.393713598  0.006608796 -0.03590325  0.008890949  0.038302284  0.065
96350
## xLE  0.014991964 -0.399378853  0.08544017  0.019941142  0.314556678 -0.195
00235
## yLE -0.415892117 -0.031293068 -0.13671720 -0.005511949  0.223947416  0.241
28765
## xN   0.018698874 -0.487562171 -0.30072123  0.229893445  0.057778273  0.342
10424
## yN  -0.405475548 -0.007788982  0.15285279 -0.019743770  0.362372110  0.049
41103
## xRM  0.011153666 -0.379847163 -0.12601406  0.164184853 -0.653772537  0.149
44976
## yRM -0.402354929 -0.021147988  0.18580243  0.053648042 -0.359204093 -0.169
31529
## xLM  0.048896923 -0.373292785  0.14299816  0.101111441 -0.001664671 -0.600
97517
## yLM -0.421652110 -0.058496574  0.12820531  0.017077591 -0.241118098 -0.032
64033
##             PC7         PC8         PC9        PC10        PC11        PC1
2
## xF   0.40491612 -0.33589079  0.33157585 -0.09590638  0.08240220 -0.2207800
9
## yF  -0.11704124 -0.05737643  0.32405676 -0.40915082 -0.53188154  0.3637055
2
## wF   0.31735222 -0.36542852  0.17719603 -0.13047353  0.06369219 -0.0803880
8
## hF  -0.21461040  0.13603592  0.13139530 -0.25335383 -0.19443019  0.0932116
1
## xRE  0.12138798 -0.10914983 -0.12095685 -0.17674548  0.36762451  0.6291271
2
## yRE  0.08383014  0.19473543  0.05547406 -0.26257109  0.14851806 -0.4127957
5
## xLE -0.40467679 -0.35098949 -0.37023166  0.15586806 -0.38526016 -0.2031211
2
## yLE -0.20689352 -0.07460613 -0.18571629 -0.32753504  0.27539435 -0.3237450
9
## xN  -0.08718797  0.20064332  0.52304943  0.35787832 -0.08996423 -0.1768641
2
## yN   0.51446644  0.32120677 -0.28163416  0.33918403 -0.28181877  0.1026247
4
## xRM  0.30047412 -0.03691901 -0.41203819 -0.19909829 -0.20384079 -0.1034874
6
## yRM -0.02933347 -0.03832200  0.14530317  0.29567071  0.06679623  0.0528724
1
## xLM -0.15730543  0.57588955 -0.04397738 -0.15662530  0.23206093  0.0190581
7
## yLM -0.24305524 -0.27447109 -0.03729036  0.34016760  0.31559282  0.1893999
5
```

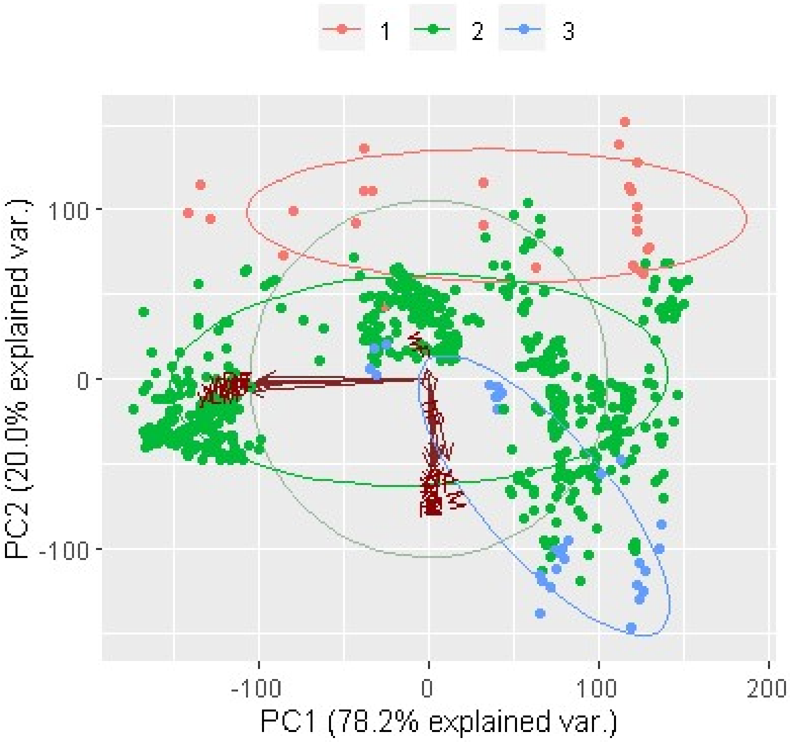
```
##             PC13         PC14
## xF  -0.10970164  0.073477543
## yF  -0.10803058 -0.003552877
## wF  -0.06914370  0.026523228
## hF  -0.04143233 -0.001925596
## xRE  0.29061840 -0.102728541
## yRE  0.28845793 -0.667175639
## xLE  0.19535238 -0.139401899
## yLE -0.14222970  0.554921054
## xN  -0.08702378  0.042554583
## yN  -0.14530951  0.068836782
## xRM -0.09530381  0.024390902
## yRM  0.64366421  0.331878469
## xLM -0.14985572  0.082970898
## yLM -0.51885202 -0.294385271
```

```
plot(pca,type="l")
```

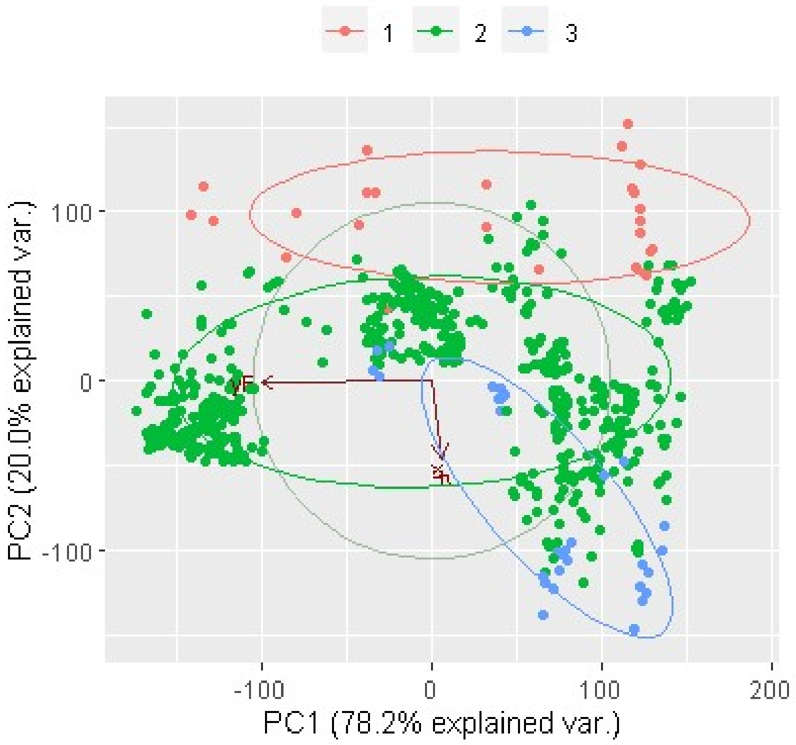


```
source("ggbiplot2.R")

componet=pca$rotation[,1:2]

#All principle components printed

g = ggbiplot(pca, obs.scale = 1, var.scale = 1,
             groups = as.factor(data2[,4]), ellipse = TRUE,
             circle = TRUE)
```

```
##
## Attaching package: 'plyr'

## The following objects are masked from 'package:rstatix':
##
##     desc, mutate

g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',
               legend.position = 'top')
g
```

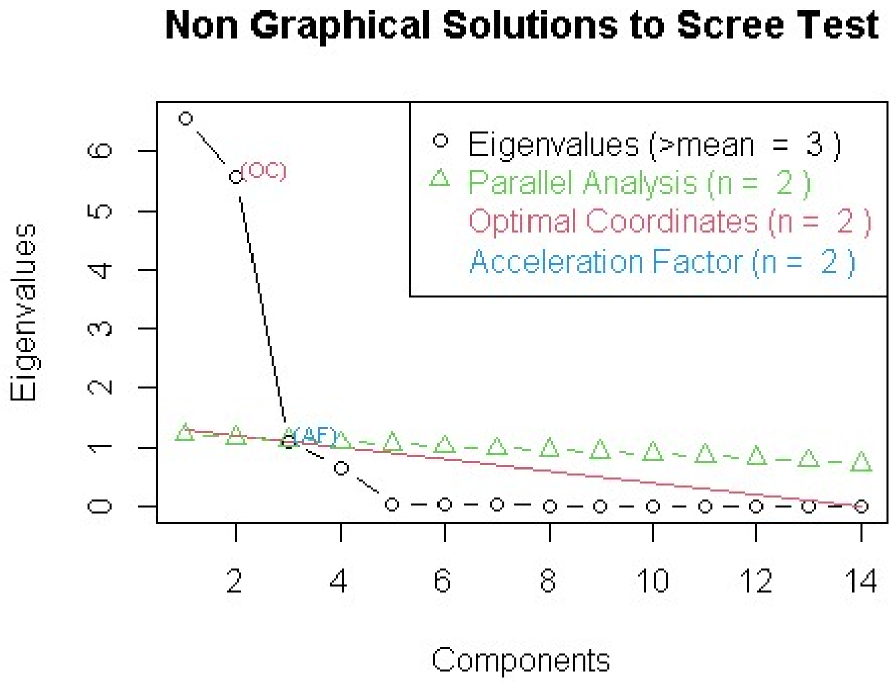


```
#Only the first two principle componenets printed (important ones)

g = ggbiplot(pca, obs.scale = 1, var.scale = 1,
             groups = as.factor(data2[,4]), ellipse = TRUE,
             circle = TRUE,numComp=2)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',
               legend.position = 'top')
g
```



```
head(data2)

##                  fileName subject imgNum label ang  xF  yF  wF  hF xRE yRE
xLE
## 1 20130529_01_Driv_001_f       1      1     2   0 292 209 100 112 323 232
367
## 2 20130529_01_Driv_002_f       1      2     2   0 286 200 109 128 324 235
366
## 3 20130529_01_Driv_003_f       1      3     2   0 290 204 105 121 325 240
367
## 4 20130529_01_Driv_004_f       1      4     2   0 287 202 112 118 325 230
369
## 5 20130529_01_Driv_005_f       1      5     2   0 290 193 104 119 325 224
366
## 6 20130529_01_Driv_006_f       1      6     2   0 290 204 105 118 324 231
368
##   yLE  xN  yN xRM yRM xLM yLM
## 1 231 353 254 332 278 361 278
## 2 235 353 258 333 281 361 281
## 3 239 351 260 334 282 362 282
## 4 230 353 253 335 274 362 275
## 5 225 353 244 333 268 363 268
## 6 232 351 253 335 277 362 276

######################################################
#2 b)
```

```
X=data2F

f1=factanal(data2F,factors=9,rotation="varimax")

f2=factanal(data2F,factors=4,rotation="varimax")

ev <- eigen(cor(X)) # get eigenvalues
ap <- parallel(subject=nrow(X),var=ncol(X),
               rep=100,cent=.05)
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)
plotnScree(nS)
```


Non Graphical Solutions to Scree Test
Eigenvalues (>mean = 3)
Parallel Analysis (n = 2)
Optimal Coordinates (n = 2)
Acceleration Factor (n = 2)
(OC)
(AF)
Eigenvalues
Components