Quiz 9

Chaeeun Shin

4/16/2024

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(neuralnet)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(rpart)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(MASS)
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
```

```
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::combine()       masks randomForest::combine()
## x dplyr::compute()       masks neuralnet::compute()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::lift()          masks caret::lift()
## x randomForest::margin() masks ggplot2::margin()
## x dplyr::select()        masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## The following object is masked from 'package:bitops':
##
##     %&%
##
## Loaded glmnet 4.1-8
```

```
library(leaps)
library(ggplot2)
```

1.

(a)

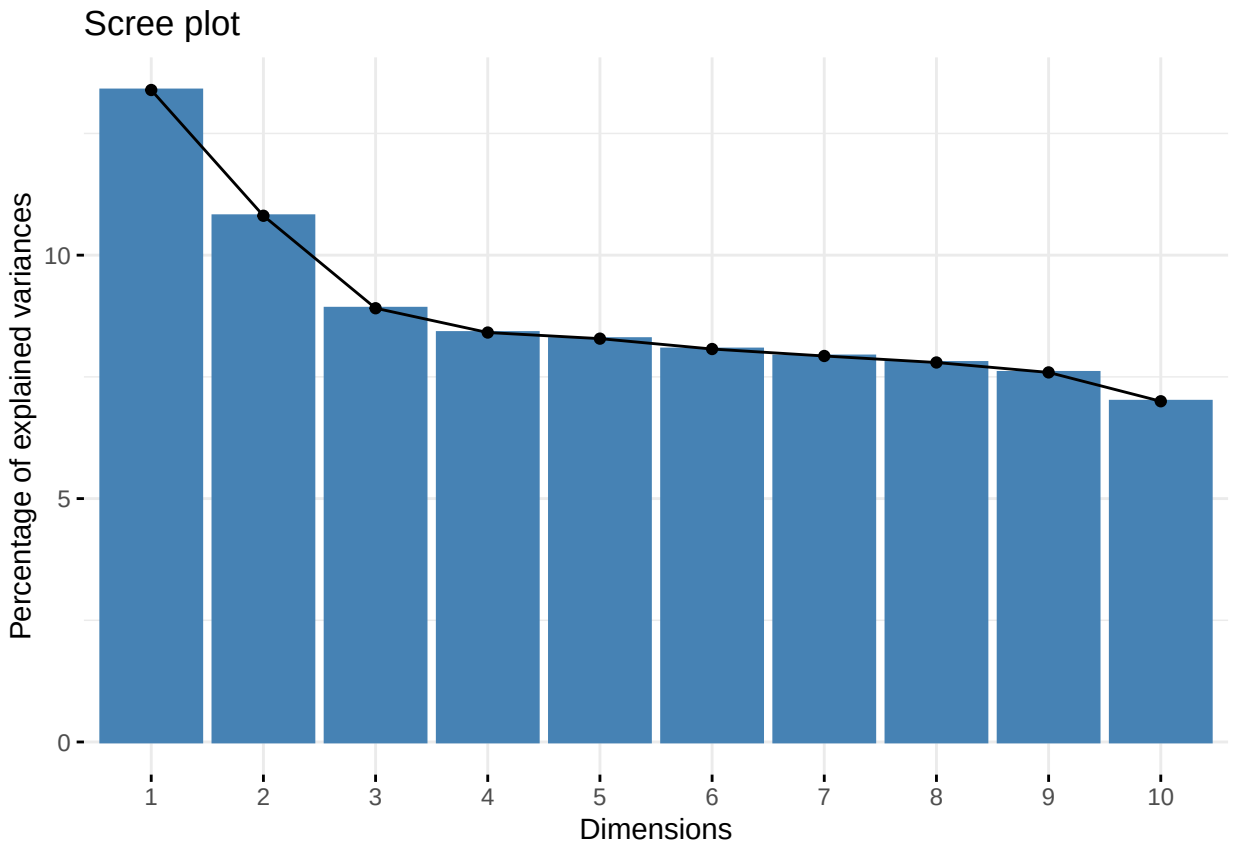
```
GreatUnknown <- read.csv("GreatUnknown.csv")
GreatUnknown <- na.omit(GreatUnknown)
data <- scale(GreatUnknown[-13])
pc <- princomp(data,cor=TRUE)
```

(b)

```
library(factoextra)
```

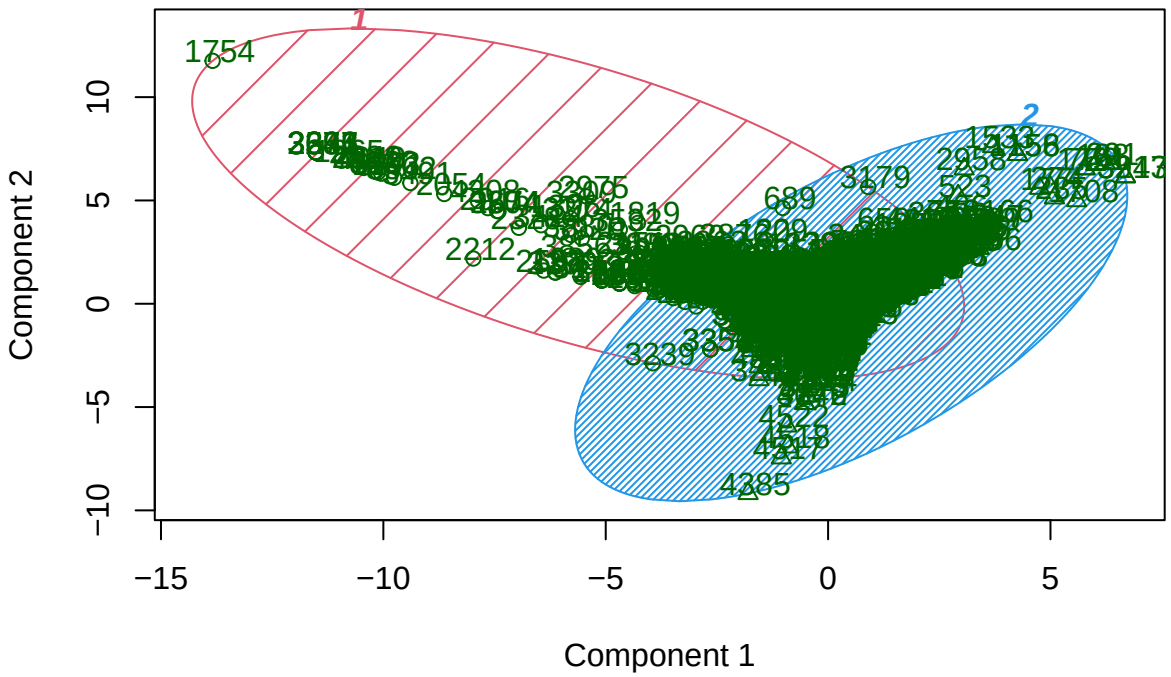
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(pc)
```



K-means fit

```
k.means.fit <- kmeans(data,2)
library(cluster)
clusplot(data,k.means.fit$cluster,main="2D representation of the Cluster solution",color=TRUE,shade=TRU
```



These two components explain 24.2 % of the point variability.

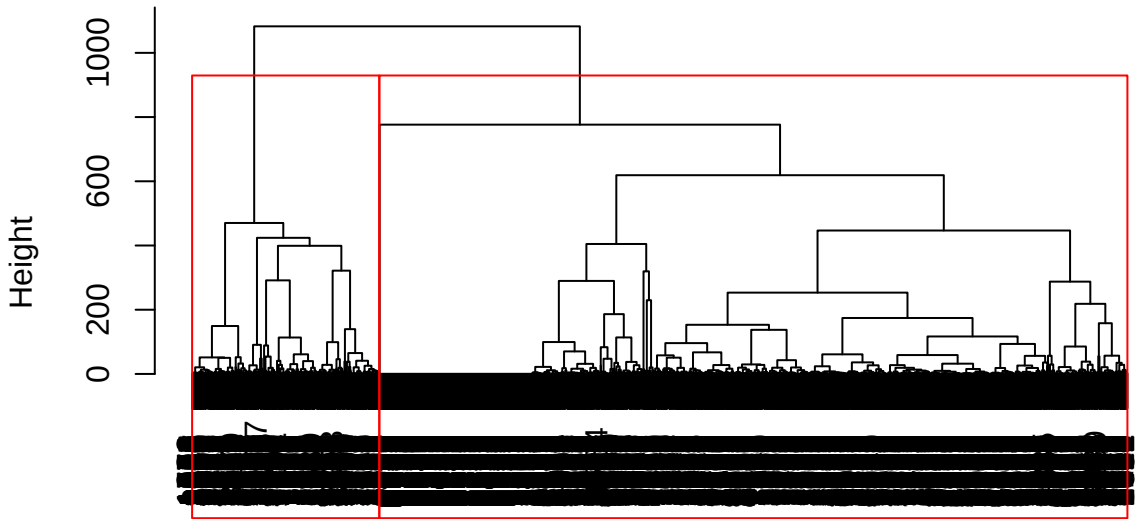
```
table(k.means.fit$cluster, GreatUnknown$y)
```

```
##
##        0    1
##   1  177    6
##   2 2611 1807
```

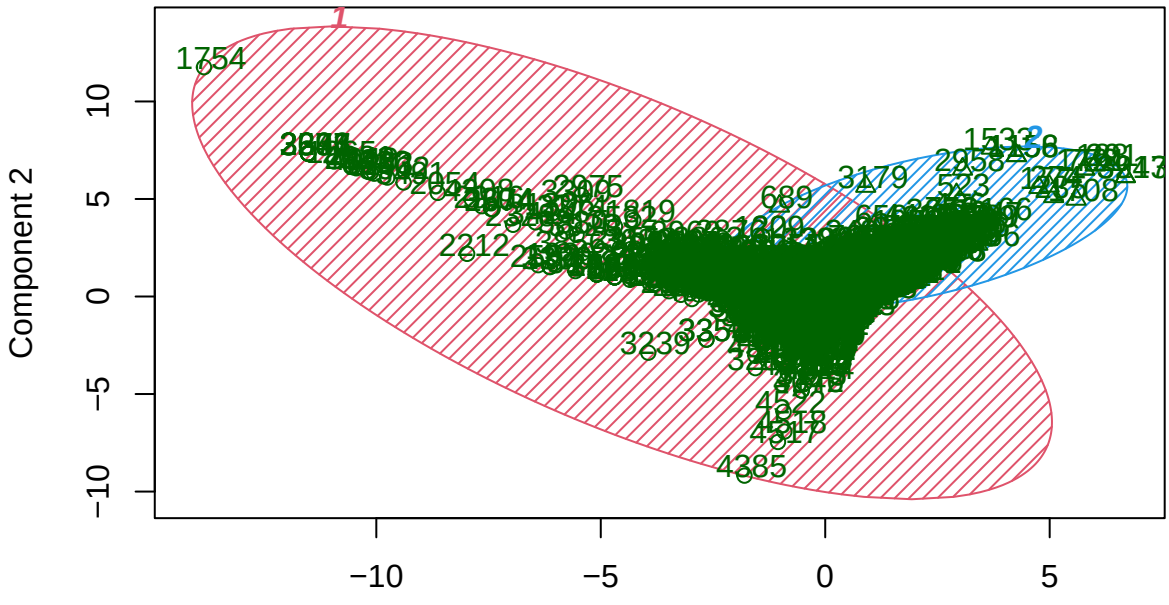
```
# accuracy = (2643+913)/4601
```

H.ward

```
d <- dist(data,method="euclidean")
H.fit <- hclust(d,method="ward.D")
plot(H.fit)
rect.hclust(H.fit,k=2,border="red")
```




```
groups <- cutree(H.fit,k=2)
clusters <- factor(groups,levels=1:2,labels=c(0,1))
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE,shade=TRUE,labels=2,li
```



These two components explain 24.2 % of the point variability.

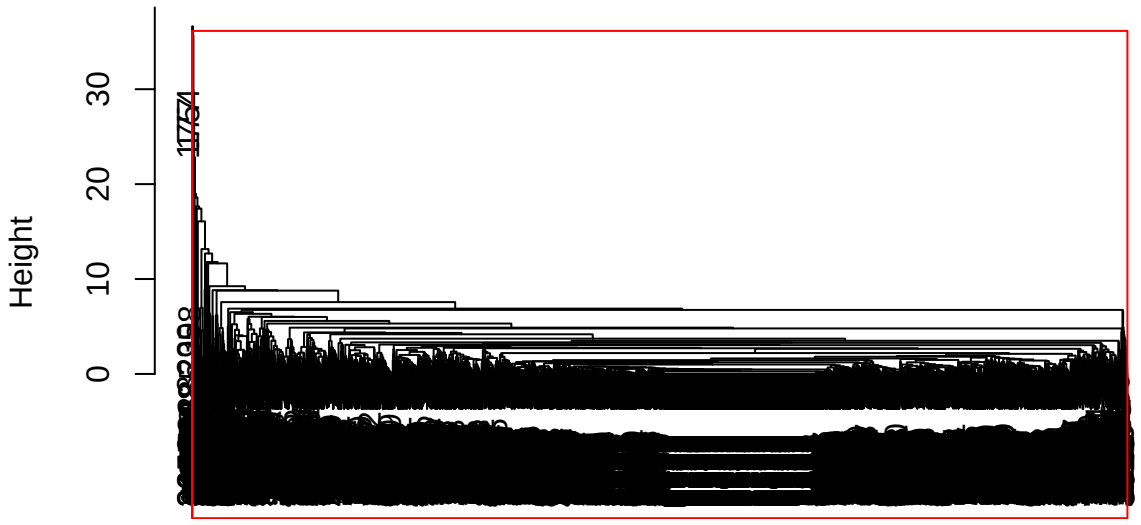
```
table(GreatUnknown$y,clusters)
```

```
##    clusters
##        0    1
##   0 2621  167
##   1 1059  754
```

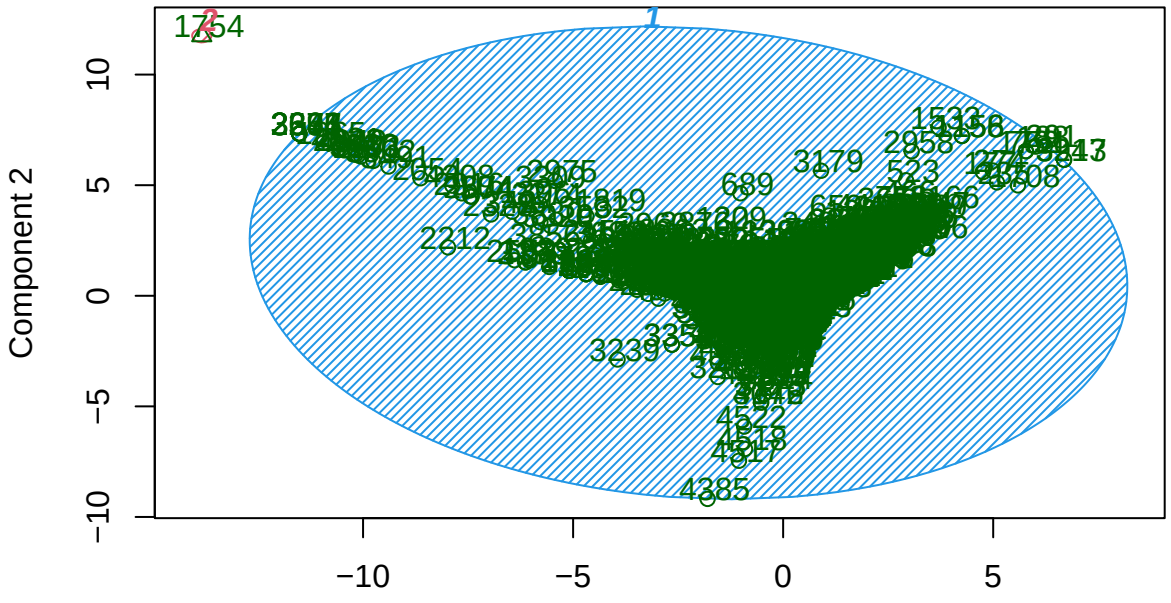
```
# accuracy = (2621+754)/4601
```

H.Average

```
H.fit <- hclust(d,method="average")
plot(H.fit)
rect.hclust(H.fit,k=2,border="red")
```




```
groups <- cutree(H.fit,k=2)
clusters <- factor(groups,levels=1:2,labels=c(0,1))
clusplot(data,groups,main="2D representation of the Cluster solution",color=TRUE,shade=TRUE,labels=2,li
```



These two components explain 24.2 % of the point variability.

```
table(GreatUnknown$y,clusters)
```

```
##    clusters
##        0    1
##   0 2788    0
##   1 1812    1
```

```
# accuracy = (2788+1)/4601
```

Comparison: K-means>H.ward>H.average Based on their accuracy on confusion matrix.

2.

```
library(devtools)
```

```
## Loading required package: usethis
```

```
## Warning: package 'usethis' was built under R version 4.2.3
```

```
library(ggbiplot)
```

```
## Warning: package 'ggbiplot' was built under R version 4.2.3
```

```
##
## Attaching package: 'ggbiplot'
```

```
## The following object is masked from 'package:rattle':
##
##     wine
```

(a)

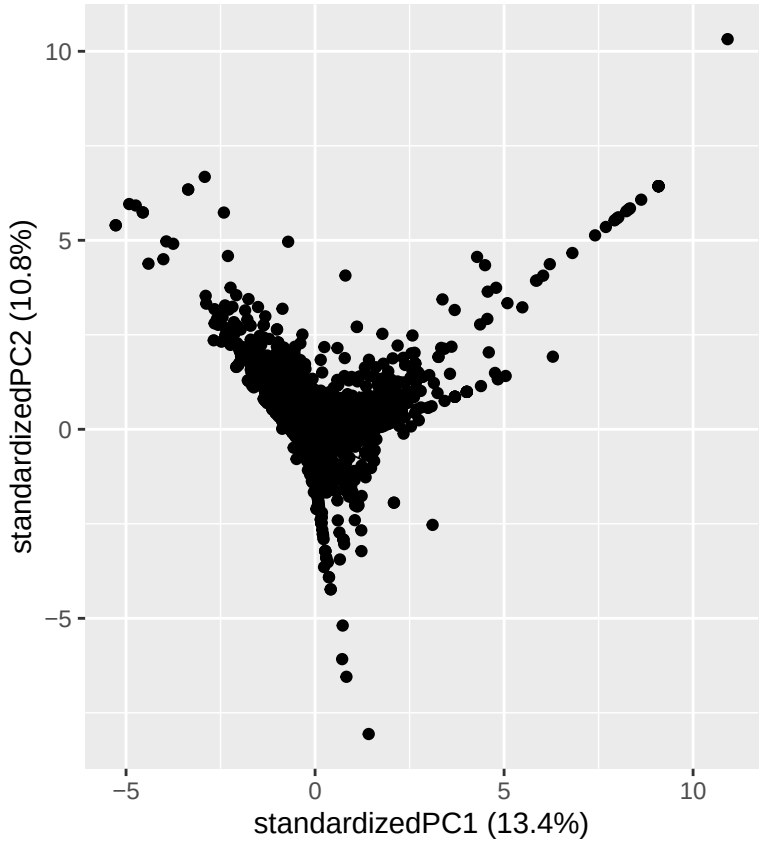
```
GreatUnknown.pca <- prcomp(data,center=TRUE,scale.=TRUE)
summary(GreatUnknown.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.2677 1.1389 1.0340 1.00462 0.99711 0.98418 0.97543
## Proportion of Variance 0.1339 0.1081 0.0891 0.08411 0.08285 0.08072 0.07929
## Cumulative Proportion  0.1339 0.2420 0.3311 0.41522 0.49807 0.57879 0.65807
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.96718 0.95444 0.91643 0.89253 0.78756
## Proportion of Variance 0.07795 0.07591 0.06999 0.06638 0.05169
## Cumulative Proportion  0.73603 0.81194 0.88193 0.94831 1.00000
```

By the row "Proportion of Variance", we can find what percentage of the variations each PC would explain.

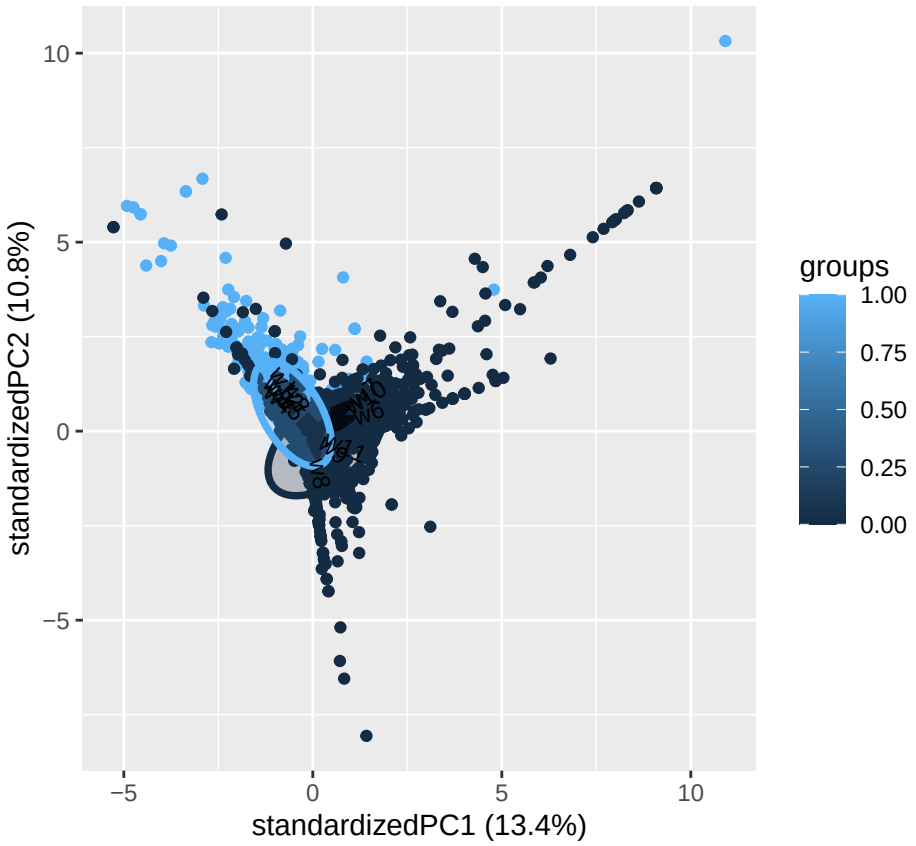
(b)

```
ggbiplot(GreatUnknown.pca)
```



(c)

```
ggbiplot(GreatUnknown.pca, ellipse=TRUE, groups=GreatUnknown$y)
```



(d)

```
GreatUnknown.pca$rotation[,1]
```

```
##          w1          w2          w3          w4          w5          w6
## -0.26312837 -0.30497476 -0.10446981 -0.27139508 -0.13025875  0.52197111
##          w7          w8          w9         w10         w11         w12
##  0.49664611  0.05810733  0.08953016  0.40916977  0.10868236 -0.16273359
```

PC1 contains parts of each w1~w12 that accounts for the greatestt possible variance.