Project 2B Epsilon

Nishanth Gandhidoss, Raghavendar Shankar, Mitul Shah

15 February 2017

Continuation of Project 2

Question 7

College Data

(70 points) Consider methods to groups colleges based on several different factors. The set of colleges considered are the 21 top colleges based on earning potential of undergraduates with computer science degrees: college.csv (Sources: Payscale.com and the National Center for Education Statistics, NCES1) After loading in the data, you will want to ensure the data is scaled in order calculate the dissimilarity matrix (scale).

In R, you will use the clustering package cluster, consider using the methods diana and agnes. In Matlab, use the linkage, pdist, and cluster functions.

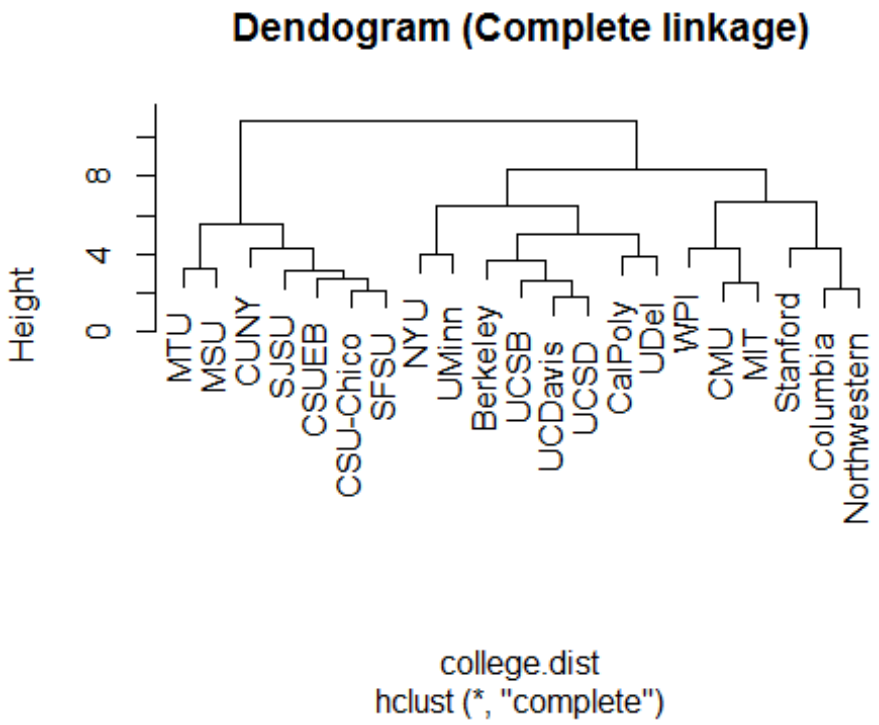
```
# Loading the data
college <- read.csv("data/college_data.csv", header = T)
```

Question 7a

(8 points) First, perform agglomerative hierarchical clustering on the college data with complete linkage. Report the dendrogram.

```
# Subsetting numeric variables
college.use <- college[, -c(1:3)]

# Computing mean for each column
means <- apply(college.use, 2, mean)

# Computing sd for each column
sd <- apply(college.use, 2, sd)

# Standardization (z-score)
college.use <- scale(college.use, center=means, scale=sd)

# Distance Matrix
college.dist <- dist(college.use, method = "euclidean", diag = FALSE, upper =
FALSE, p = 2)

# Hierarchical Clustering by complete linkage
college.hclust <- hclust(college.dist, method = "complete")
```

```r
# Plotting the Dendogram
plot(college.hclust, labels = college$ShortHandName, main='Dendogram
(Complete linkage)')
```

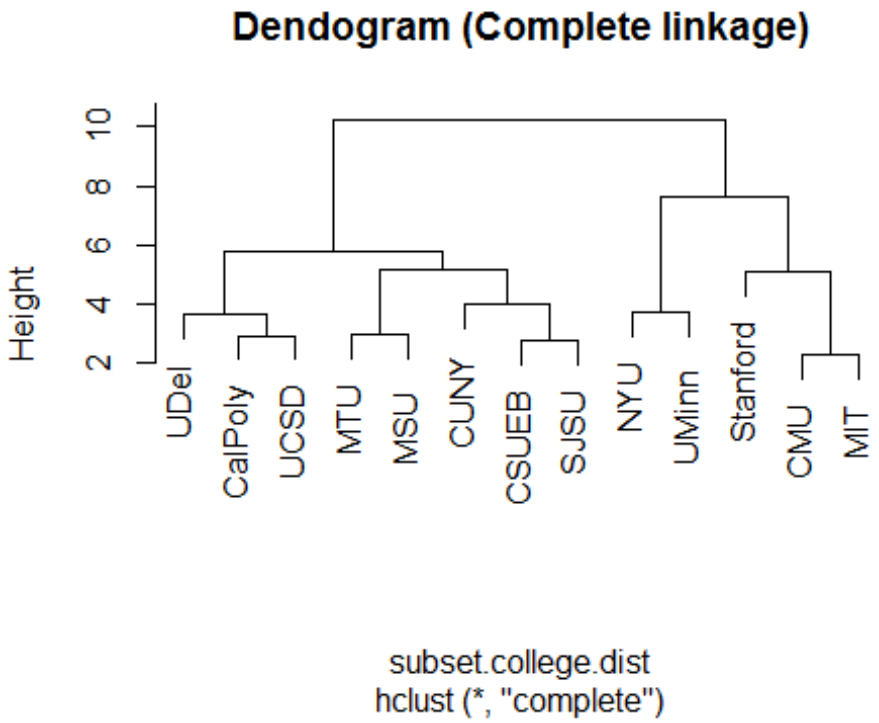


Question 7b

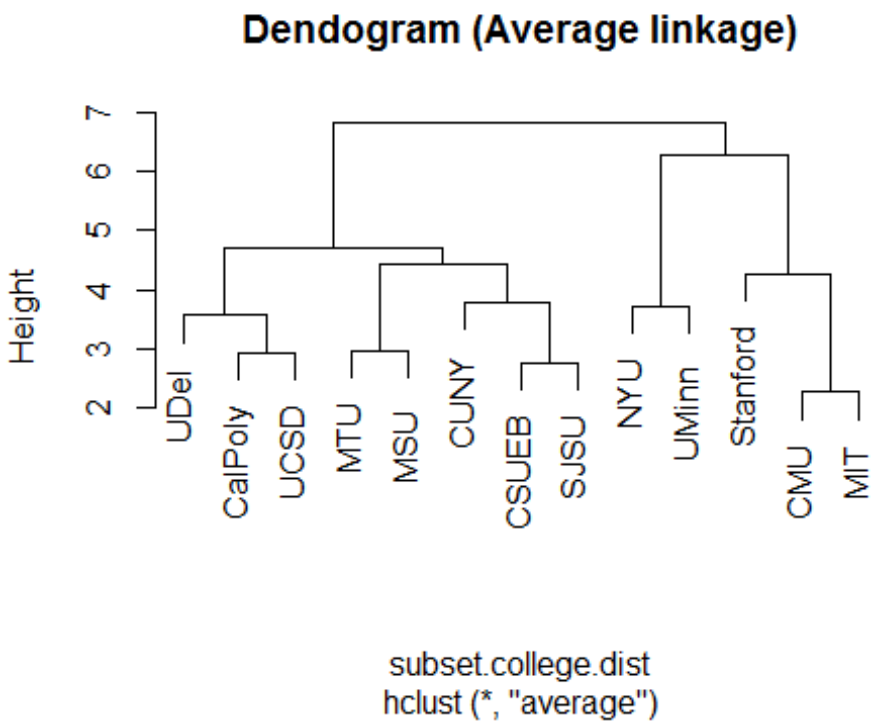
(16 points) Remove from consideration the following schools: CSU-Chico, Columbia, Northwestern, SFSU, Berkeley, UCDavis, UCSB, and WPI. Perform and report the dendrogram for agglomerative hierarchical clustering with complete, and average linkage.

```r
# Removing the colleges according to the question
subset.college <- college[-c(2, 5, 11, 12, 15, 16, 18, 21), ]

# Subsetting numeric variables
subset.college.use <- subset.college[, -c(1:3)]

# Computing mean for each column
means <- apply(subset.college.use, 2, mean)

# Computing sd for each column
sd <- apply(subset.college.use, 2, sd)

# Standardization (z-score)
subset.college.use <- scale(subset.college.use, center=means, scale=sd)

# Distance Matrix
```

```r
subset.college.dist <- dist(subset.college.use, method = "euclidean", diag =
FALSE, upper = FALSE, p = 2)

# Hierarchical Clustering by complete linkage
complete.college.hclust <- hclust(subset.college.dist, method = "complete")

# Plotting the Dendogram
plot(complete.college.hclust, labels = subset.college$ShortHandName,
main='Dendogram (Complete linkage)')
```



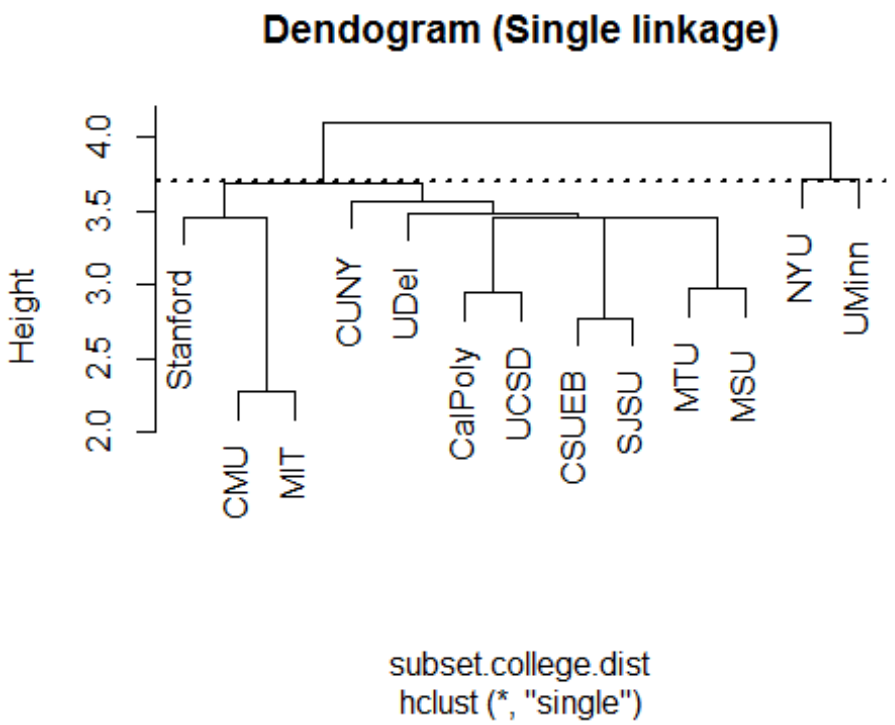

```r
# Hierarchical Clustering by average linkage
average.college.hclust <- hclust(subset.college.dist, method = "average")

# Plotting the Dendogram
plot(average.college.hclust, labels = subset.college$ShortHandName,
main='Dendogram (Average linkage)')
```



Question 7c

(12 points) Run the agglomerative hierarchical clustering with single linkage for the same data as in (b). Report the k = 3 cluster assignments.

```
# Hierarchical Clustering by single linkage
single.college.hclust <- hclust(subset.college.dist, method = "single")

# Plotting the Dendogram
plot(single.college.hclust, labels = subset.college$ShortHandName,
main='Dendogram (Single linkage)')

# Dendogram showing 3 clusters
abline(h=3.7, lty=3, lwd=2)
```




```
# Viewing the number of colleges in different clusters
groups.3 <- cutree(single.college.hclust,3)
table(groups.3)

## groups.3
##  1  2  3
## 11  1  1
```

We see that there are 11 colleges in the 1st cluster while the clusters 2 and 3 have only one college.

Clusters:

Cluster 1: Stanford, CMU, MIT, CUNY, UDel, CalPoly, UCSD, CSUEB, SJSU, MTU, MSU

Cluster 2: NYU

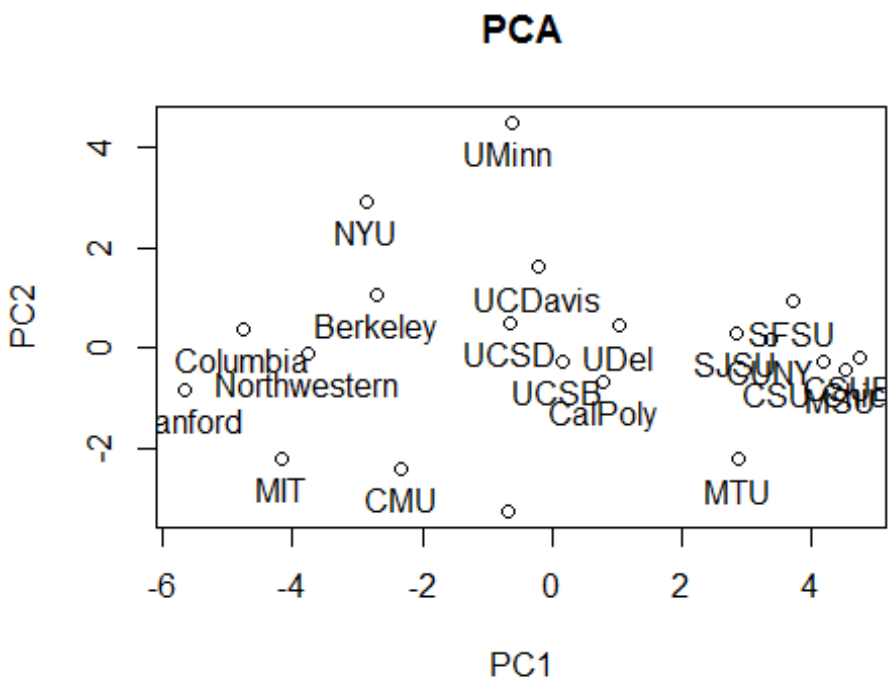
Cluster 3: UMinn

Question 7d

(10 points) Perform principal component analysis on the college data of (a). Plot the first two principal component scores (labeling each point with the school it represents).

```
## Perform PCA
pc <- prcomp(college.use, scale = T)

## Plotting the first 2 Principal components
```

```
plot(pc$x[, 1], pc$x[, 2], main = "PCA", xlab = "PC1", ylab = "PC2")

## Labelling by the school
text(pc$x[,1] , pc$x[,2], labels = college$ShortHandName, pos = 1)
```



Question 7e

(12 points) Run kmeans clustering on the data of (b) and report the assignments with k=3 and compare to (c)

```
# Setting the seed to get the same results as the initial cluster assignments are random
set.seed(10)

# k-Means Clustering
kmeans.college.cluster <- kmeans(subset.college.use, 3, nstart = 10)

# Looking at clusters formed by k-Means Clustering
kmeans.college.cluster$cluster
```

```
##  1  3  4  6  7  8  9 10 13 14 17 19 20
##  1  1  3  1  3  1  1  2  1  3  2  1  2
```

Let's compare these clustering results with single linkage hierarchical clustering.

kmeans Clustering results:

Cluster 1: CalPoly, CSUEB, CUNY, MTU, MSU, SJSU, UDel

Cluster 2: NYU, UCSD, UMinn

Cluster 3: CMU, MIT, Stanford

Single Linkage Hierarchical Clustering results:

Cluster 1: Stanford, CMU, MIT, CUNY, UDel, CalPoly, UCSD, CSUEB, SJSU, MTU, MSU

Cluster 2: NYU

Cluster 3: UMinn

The results of the kMeans Clustering seem to be better here as the Single linkage Hierarchical Clustering only shows 1 college in 2 of the clusters. Also, the Universities CMU, MIT and Stanford in 1 cluster in KMeans results makes more sense.

Question 7f

(12 points) Run kmediods clustering on the data of (b) and report the assignments with k=3 and compare to (e)

```
# Loading the library cluster
library(cluster)

# PAM (Partitioning around Mediods)
subset.college.pam <- pam(subset.college.dist, 3)

# Looking at the results of KMediods Clustering
subset.college.pam$clustering
```

```
##  1  3  4  6  7  8  9 10 13 14 17 19 20
##  1  2  3  2  3  2  2  1  2  3  1  1  1
```

kMediods Clustering Results:

Cluster 1: CalPoly, NYU, UCSD, UDel, UMinn

Cluster 2: CSUEB, CUNY, MTU, MSU, SJSU

Cluster 3: CMU, MIT, Stanford

kmeans Clustering results:

Cluster 1: CalPoly, CSUEB, CUNY, MTU, MSU, SJSU, UDel

Cluster 2: NYU, UCSD, UMinn

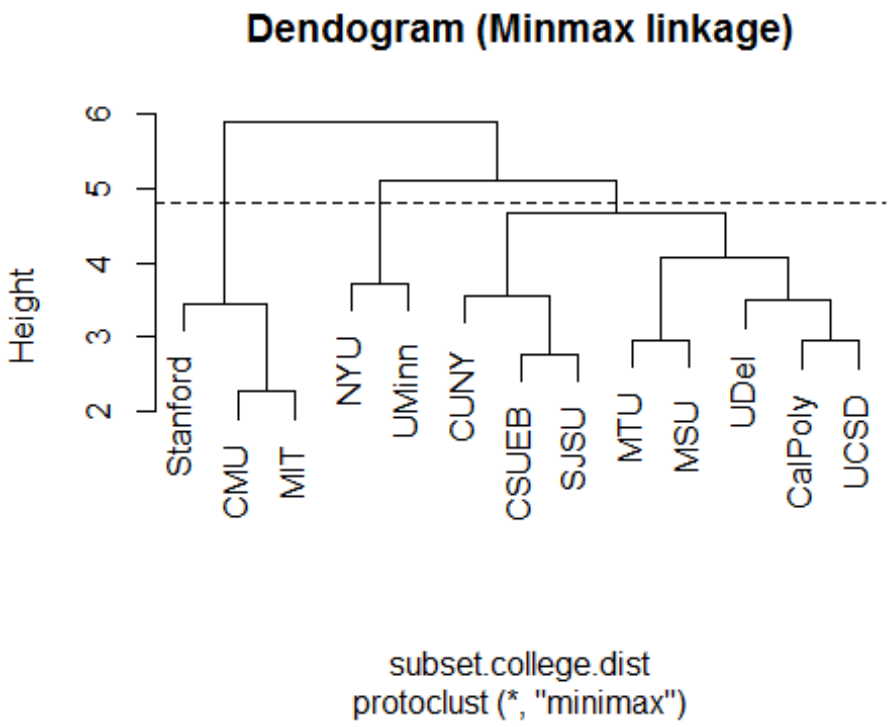
Cluster 3: CMU, MIT, Stanford

If we look at the results carefully, only two colleges, i.e. UDel and CalPoly changed their clusters. If we want our clusters to have more equal number of colleges in each cluster, then we might prefer kMediods Clustering here.

Question 7g

(4 points (bonus)) Perform agglomerative hierarchical clustering with minimax linkage. Report the dendrogram.

```
# Min-max Linkage
minmax.college.hclust <- protoclust(subset.college.dist)

# Choosing the number of clusters to be 3
k <- 3

# Dendogram for Minmax Linkage
plot(minmax.college.hclust, labels = subset.college$ShortHandName, main =
"Dendogram (Minmax linkage)")

# Showing 3 clusters by Dendogram
abline(h=4.8, lty=2)
```




```
# Looking at results of Minmax Linkage
cut <- protocut(minmax.college.hclust, k=k)
cut$cl
```

```
##  1  3  4  6  7  8  9 10 13 14 17 19 20
##  1  1  2  1  2  1  1  3  1  2  1  1  3
```

Question 8

Music Data

For this problem you will consider several properties that have been measured from music recordings.

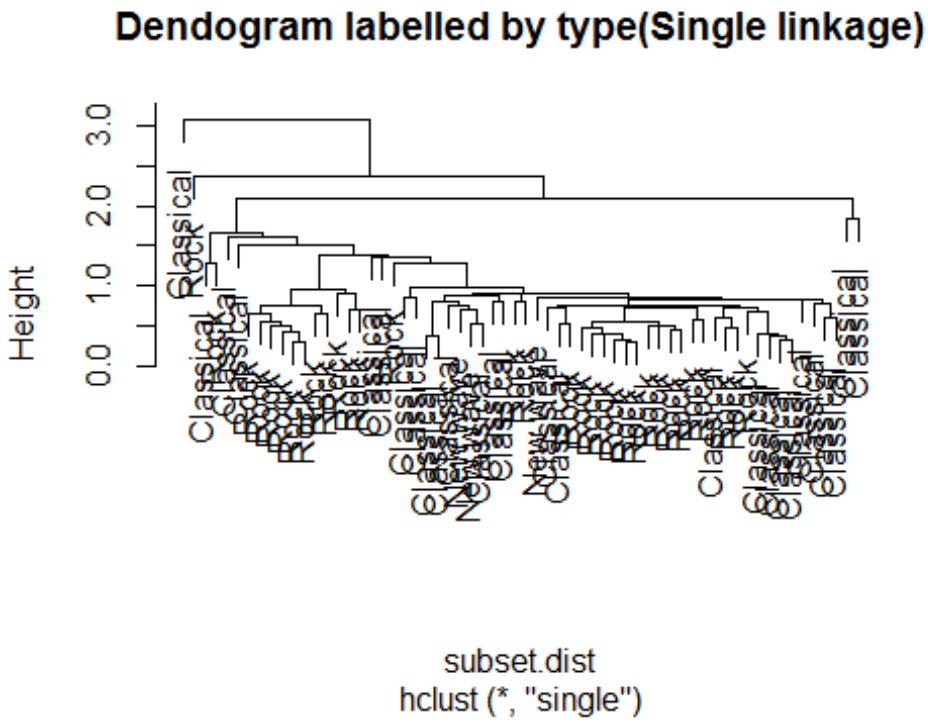
Consider only the numeric variables from the data: music.csv.

First, standardize the variables.

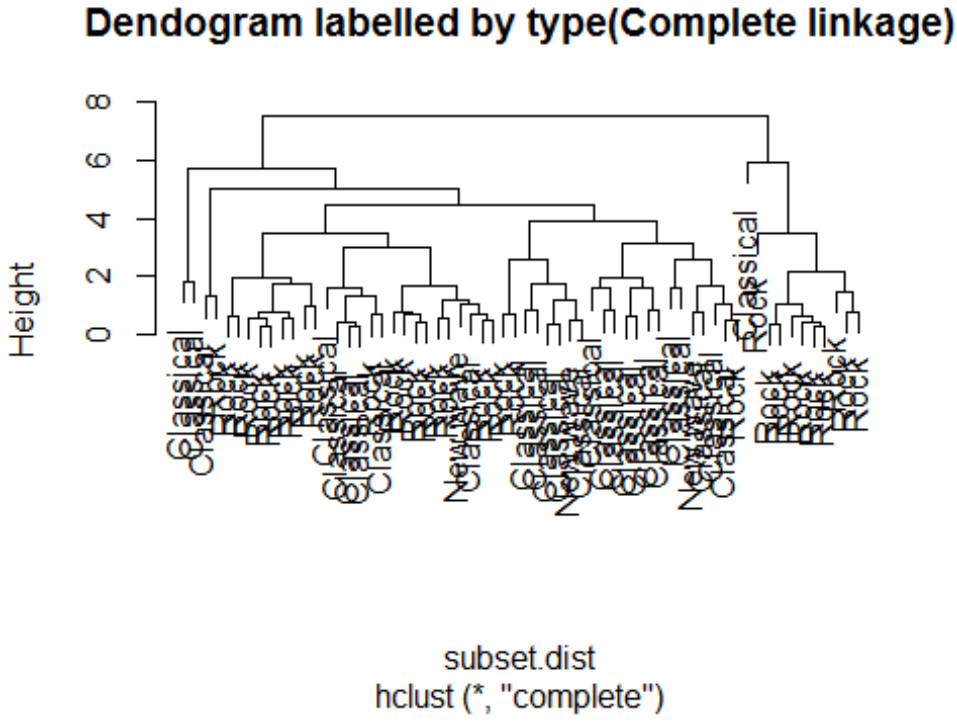
Then, perform hierarchical clustering two times, with single and complete linkage. Label the clusters by the `Type' of music.

Repeat the analysis as above, but label the samples by the musical `Artist'. Which method seems best?

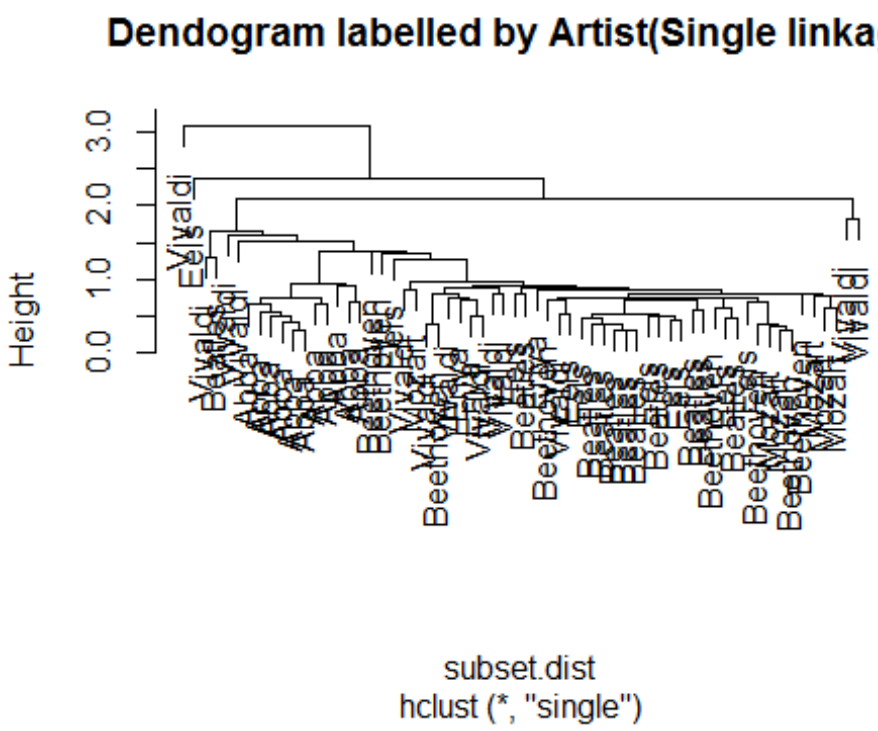
```
# Reading the data
music <- read.csv("data/music2.csv", header = T)

# Subset the numeric variables (Looked at the Structure of the dataframe)
subset.music <- music[, -c(1:3)]

## Computing mean for each column
means <- apply(subset.music, 2, mean)

## Computing sd for each column
sd <- apply(subset.music, 2, sd)

## Standardization (z-score)
subset.music <- scale(subset.music, center=means, scale=sd)

## Distance Matrix
subset.dist <- dist(subset.music, method = "euclidean", diag = FALSE, upper =
FALSE, p = 2)

## Hierarchical Clustering by single linkage
music.hclust.single <- hclust(subset.dist, method = "single")

## Hierarchical Clustering by complete linkage
music.hclust.complete = hclust(subset.dist, method = "complete")

# Labeling by type
# Plotting the Dendogram by single linkage
plot(music.hclust.single, labels = music$Type, main='Dendogram labelled by
type(Single linkage)')
```



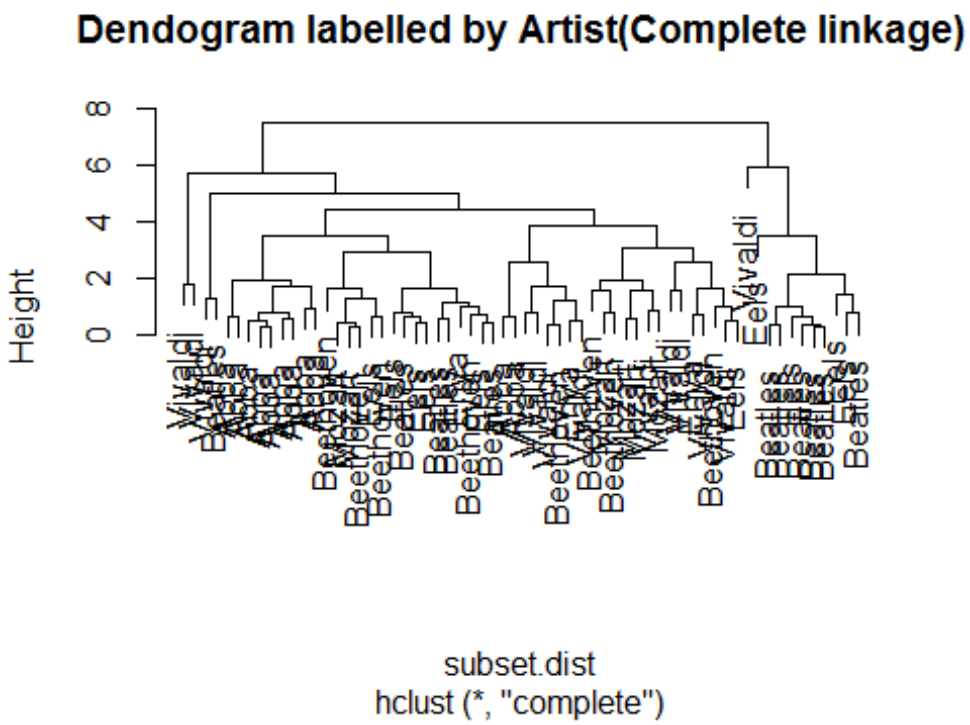

```
# Plotting the Dendogram by complete linkage
plot(music.hclust.complete, labels = music$Type, main='Dendogram labelled by
type(Complete linkage)')
```



```r
#Labeling by artist
# Plotting the Dendogram by single linkage
plot(music.hclust.single, labels = music$Artist, main='Dendogram labelled by
Artist(Single linkage)')
```




```r
# Plotting the Dendogram by complete linkage
plot(music.hclust.complete, labels = music$Artist, main='Dendogram labelled
by Artist(Complete linkage)')
```



Now, let's try to compare all the clustering methods.

Here, we are asked to label by Type and Artist. But, we know that if we consider grouping by type, we have 3 clusters and if we consider grouping by Artist, we have 7 clusters, as the labels are already given in the dataset (Note that in an unsupervised learning problem like Clustering, these labels are unknown and we wouldn't know how many groups exist in the data). So let's try to form 3 clusters and 7 clusters by using both single linkage and complete linkage and try to see whether we can actually separate these groups.

```
## Number of unique types in the data
length(unique(music$Type))

## [1] 3

## Number of unique artists in the data
length(unique(music$Artist))

## [1] 7

#### Comparing all the clusters

## Using 3 clusters for complete linkage
complete.linkage.clusterCut1 <- cutree(music.hclust.complete, 3)

## Looking at the results labelling by type to see whether we can actually
```

```
separate all 3 types using complete linkage
table(complete.linkage.clusterCut1, music$Type)

##
## complete.linkage.clusterCut1 Classical New wave Rock
##                            1        26        3   22
##                            2         1        0    0
##                            3         0        0   10

## Using 7 clusters for complete linkage
complete.linkage.clusterCut2 <- cutree(music.hclust.complete, 7)

## Looking at the results labelling by artist to see whether we can actually
separate all 7 artists using complete linkage
table(complete.linkage.clusterCut2, music$Artist)

##
## complete.linkage.clusterCut2 Abba Beatles Beethoven Eels Enya Mozart
##                            1    9       4         4    5    1      2
##                            2    2       0         1    0    1      0
##                            3    0       0         3    1    1      4
##                            4    0       0         0    0    0      0
##                            5    0       1         0    0    0      0
##                            6    0       0         0    0    0      0
##                            7    0       6         0    4    0      0
##
## complete.linkage.clusterCut2 Vivaldi
##                            1       0
##                            2       4
##                            3       5
##                            4       2
##                            5       1
##                            6       1
##                            7       0

## Using 3 clusters for single linkage
single.linkage.clusterCut1 <- cutree(music.hclust.single, 3)

## Looking at the results labelling by type to see whether we can actually
separate all 3 types using single linkage
table(single.linkage.clusterCut1, music$Type)

##
## single.linkage.clusterCut1 Classical New wave Rock
##                          1        26        3   31
##                          2         1        0    0
##                          3         0        0    1

## Using 7 clusters for single linkage
single.linkage.clusterCut2 <- cutree(music.hclust.single, 7)
```

```
## Looking at the results labelling by artist to see whether we can actually
separate all 7 artists using single linkage
table(single.linkage.clusterCut2, music$Artist)

##
## single.linkage.clusterCut2 Abba Beatles Beethoven Eels Enya Mozart Vivaldi
##                          1   11      10         8    9    3      6       8
##                          2    0       0         0    0    0      0       1
##                          3    0       0         0    0    0      0       1
##                          4    0       1         0    0    0      0       1
##                          5    0       0         0    0    0      0       1
##                          6    0       0         0    0    0      0       1
##                          7    0       0         0    1    0      0       0
```

If we look at the clustering results of all the 4 methods above, we see that no method was able to separate all the groups. Neither were we able to separate all three Types into 3 clusters using complete/single linkage, nor were we able to separate seven artists into 7 clusters. But, looking at the dendogram, the complete linkage seems to be giving better results as the clusters seem to have equal number of points when we try to form less number of clusters. Here, no method seems to separate the groups well as we see that the first cluster in all the 4 results is very vague and contains everything. But, if we still have to choose one of these methods, we would go with separating type by complete linkage as this method atleast puts all the Classical Music into 1 cluster. But again that cluster also contains 22 Rock types. We might be able to separate all the 3 types using some other more complicated clustering methods like Chameleon, Birch, etc.

Question 9

Voting Data

For this question you will consider methods to group members of Congress based on their voting records. The voting records from congress are available at Oce of the Clerk, US House of Representatives, but not in a form that is easily digestible for analysis.

In fact it was only in 2016, that Congress agreed to make legislative data available themselves. Govtrack.us has links to primary data sources and api's projects that collect and release the data in easier digestible forms:

- Govtrack.us - https://www.govtrack.us/developers
- Github Congress project - https://github.com/unitedstates/congress
- ProPublica's Congress API - https://propublica.github.io/congress-api-docs/

A long-standing project to document congressional roll call votes is available at the Inter-university Consortium for Political and Social Research (ICPSR). This data includes roll call votes from 1789 - 1990. The ICPSR formatting for storing this data has been used on other sites which are keeping up with the creating a record, UCLA. For more recent years of Congress,data is collected at the following sites: http://adric.sscnet.ucla.edu/rollcall/3.

For instance, to explore the 114th Congress, there are two files available:

- H114.ord - roll call file detailing votes of each member of US House of Representatives in the 114th Congress.
- H114desc.csv - file detailing the votes

The basic format of the data consist of many fixed width items listed for each Congressperson (including their votes for each roll call).

114107132313MICHIGA 10001CONYERS 7966161161661616111 ... 114130351313NEW YOR 10001RANGEL 9900000161661616111 ... 1142114123 1MICHIGA 20001BENISHEK 7911611111116161116 ...

The main structure of the data is:

- 3 digits - Congress number
- 5 digits - ICPSR number (unique number for each member of Congress/Senate)
- 2 digits - ICPSR State code
- 2 digits - District number
- 8 characters - State name
- 3 digits - Party code: (100 Dem., 200 Rep.)
- 2 digits - ICPSR Occupancy and Once code
- 11 characters - Member's name
- remaining digits - Votes

Votes codes: 0 - Not in Legislature, f1; 2; 3g - Yea, f4; 5; 6g - Nay, f7; 8; 9g - Abstentions

The first 5 votes of the 114th Congress have to do with electing the Speaker of the House and rules votes; ignore these first 5 votes. Randomly select 20 votes from the remaining roll calls of the 114th Congress. Perform principal component analysis on the voting record and plot the first two principal components.

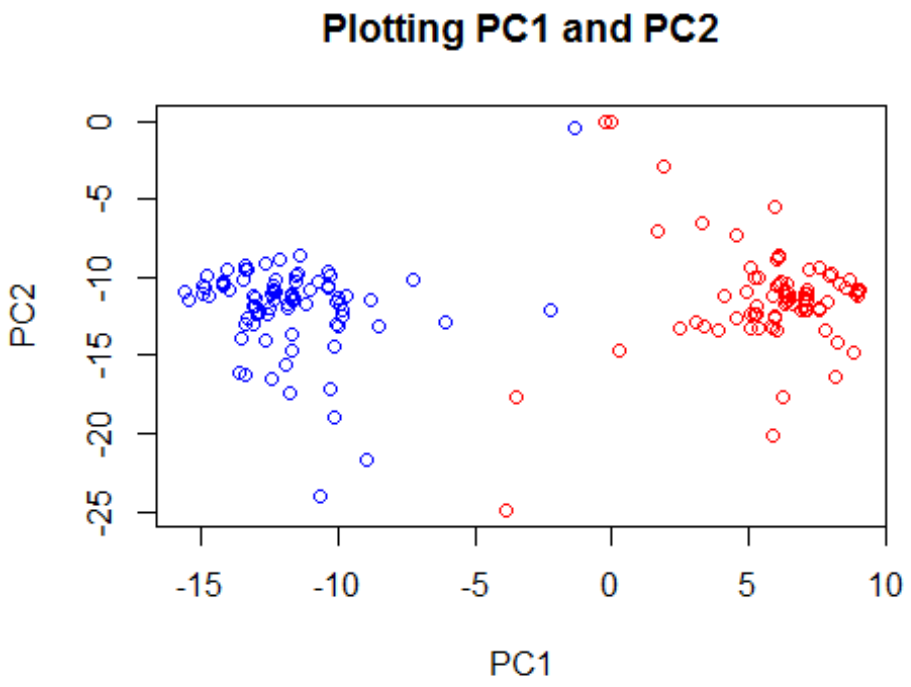
Note, this data is in a sense supervised, we know that party affiliation of each member of the House of Representative. Therefore, color the plot of the first two principal components based on party (red = Republican, blue = Democrat). Comment on the results of this analysis.

Repeat this analysis with a random sample of 100 votes from the 114th Congress. Also, show the amount of variance explained by the first 10 principal components. Comment on whether you should be able to predict a Congress person's party affiliation from their voting record.

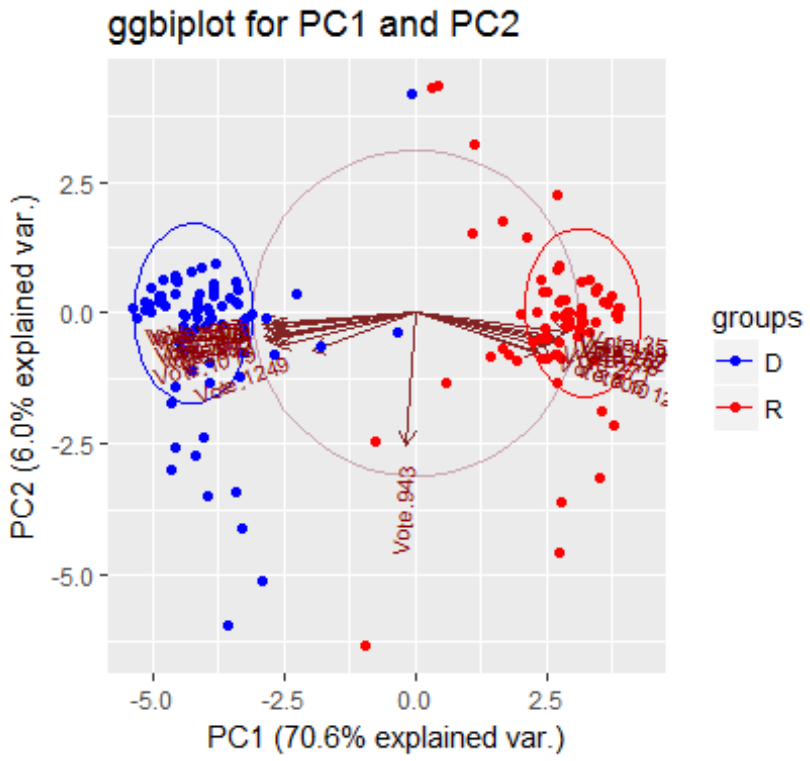
```
# Importing the data from ord file
H114 <- readKH("data/H114.ord")

## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 441 legislators and 1327 roll calls
## Frequency counts for vote types:
```

```
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  11936 323831 226660   1313  21467      0

summary(H114)

## Source:       data/H114.ord
##
## Number of Legislators:        441
## Number of Roll Call Votes:    1327
##
##
## Using the following codes to represent roll call votes:
## Yea:      1 2 3
## Nay:      4 5 6
## Abstentions:  7 8 9
## Not In Legislature:   0
##
## Party Composition:
##    D    R <NA>
##  190  251    0
##
## Vote Summary:
##                  Count Percent
## 0 (notInLegis)  11936     2.0
## 1 (yea)        323831    55.3
## 6 (nay)        226660    38.7
## 7 (missing)      1313     0.2
## 9 (missing)     21467     3.7
##
## Use summary(H114,verbose=TRUE) for more detailed information.

data <- data.frame(H114$legis.data,H114$votes)

# Ignoring the first five votes representing the U.S House of Representatives
data <- data[-c(7,8,9,10,11)]

# Sampling and scaling 20 votes from the dataset
set.seed(1)
data.vote.20 <- sample(x = data[7:1328],size = 20)
data.vote.20 <- data.frame(cbind(data$party,data.vote.20))

# Feature Scaling for 20 votes
data.vote.20[,2:21] <- scale(data.vote.20[,2:21],center = TRUE,scale = TRUE)

# Perform Principal Component Analysis for 20 votes
# Applying PCA for 20 votes

pca.vote.20 <- princomp(x = data.vote.20[-1],cor = TRUE,scores = TRUE)
pca.predict.20 <- data.frame(predict(object = pca.vote.20,data[7:1328]))
```

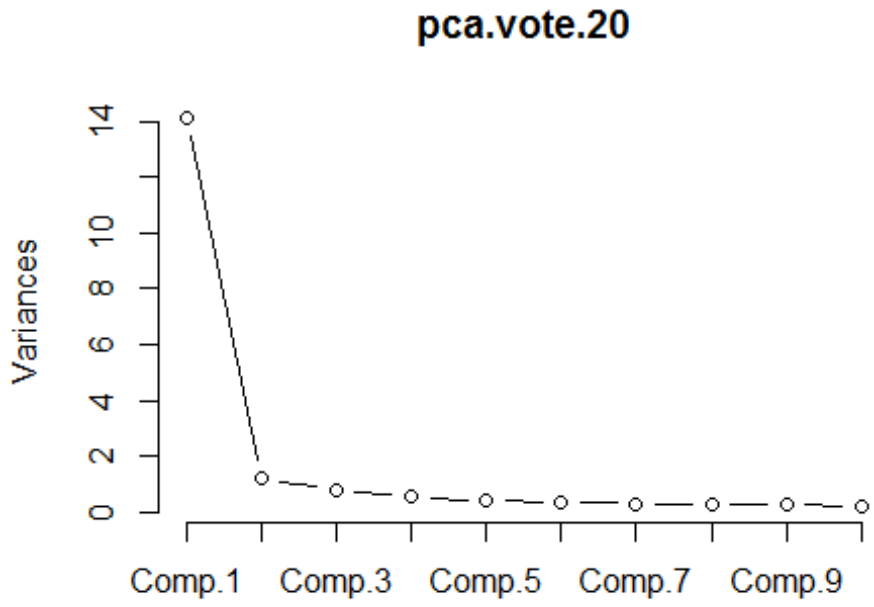
```r
pca.predict.20 <- mutate(.data = pca.predict.20,Data.Party =
data.vote.20$data.party)

# Plotting the extracted features and variences across the components
plot(x = pca.predict.20$Comp.1, y = pca.predict.20$Comp.2,xlab = "PC1",ylab =
"PC2",main = "Plotting PC1 and PC2")
points(x = pca.predict.20,pch = 21,col = ifelse(pca.predict.20$Data.Party ==
'D','blue','red'))
```




```r
# Using ggbiplot for PCA model1
ggbiplot(pcobj = pca.vote.20,obs.scale = 1,var.scale = 1,groups =
data.vote.20$data.party,ellipse = TRUE,circle =
TRUE)+scale_color_manual(values = c("D" = "blue","R" = "red"))+ggtitle(label
= "ggbiplot for PC1 and PC2")
```



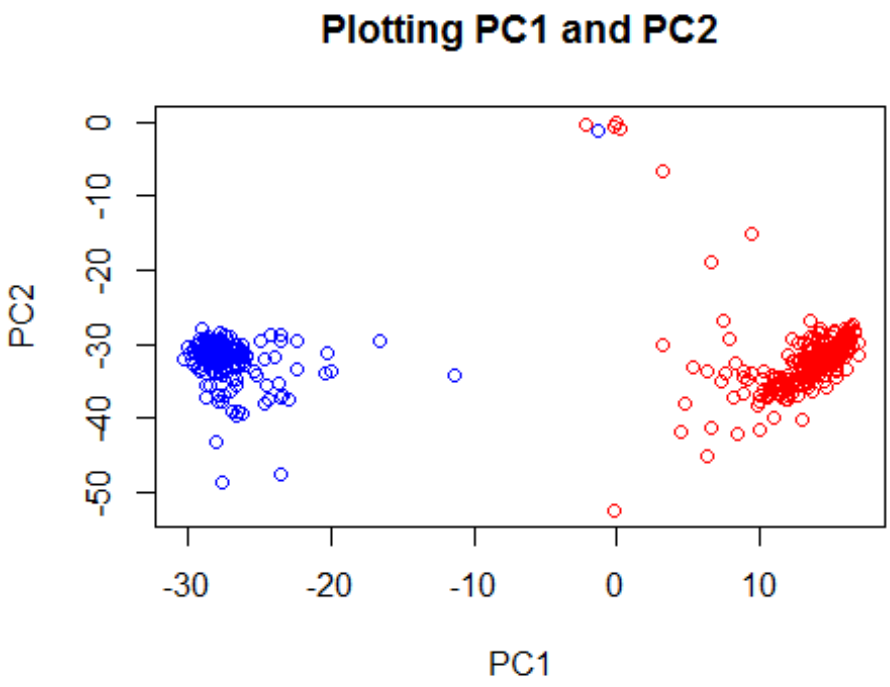

```
# Varience across the features for 20 votes
plot(pca.vote.20,type = 'l')
```



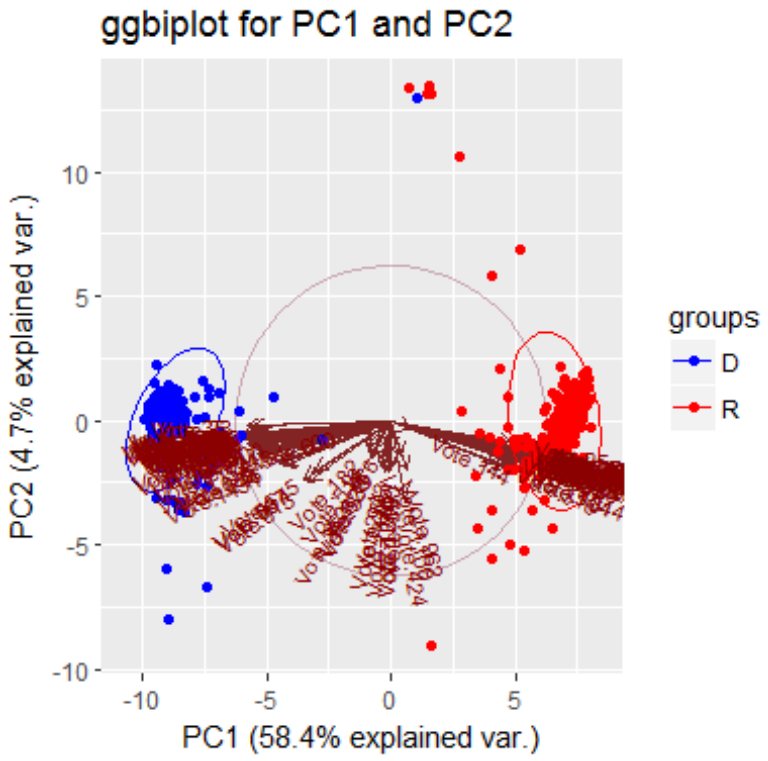
From the scatterplot,ggbiplot and varience, it is seen that PC1 accounts to 70.6% variance on the data with PC2 as 6.0% when analysing 20 sampled votes. Here, there is a definite segregation between the two parties based on their voting records. Hinojosa is a staunch democrat based on the component PC1 having high negative values and Hice is a staunch republican based on high positive values. Jones, Gibson and Davidson, being republicans are closer to the centre and hence neutral in voting records.

Repeating Principal Component Analysis for 100 votes

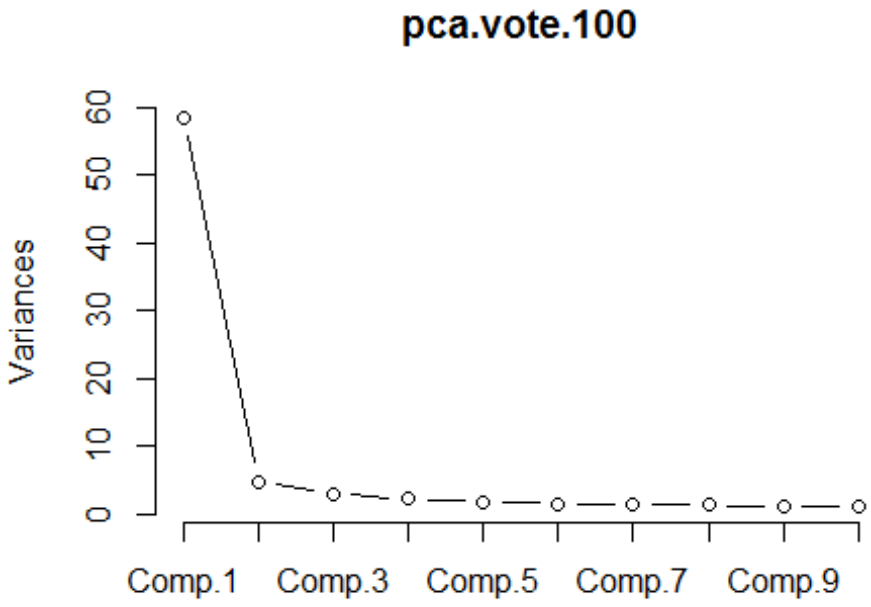
```
# Data sampling and scaling for 100 votes
set.seed(1)
data.vote.100 <- sample(x = data[7:1328],size = 100)
data.vote.100 <- data.frame(cbind(data$party,data.vote.100))
data.vote.100[,2:101] <- scale(data.vote.100[,2:101],center = TRUE,scale =
TRUE)

# Data Preprocessing using PCA analysis for 100 votes
pca.vote.100 <- princomp(x = data.vote.100[-1],cor = TRUE,scores = TRUE)
pca.predict.100 <- data.frame(predict(object = pca.vote.100,data[7:1328]))
pca.predict.100 <- mutate(.data = pca.predict.100,Data.Party =
data.vote.100$data.party)

# Plotting the extracted features and variences across the components
# of the two extracted features for 100 votes
plot(x = pca.predict.100$Comp.1, y = pca.predict.100$Comp.2,xlab = "PC1",ylab
= "PC2",main = "Plotting PC1 and PC2")
points(x = pca.predict.100,pch = 21,col = ifelse(pca.predict.100$Data.Party
== 'D','blue','red'))
```




```
# Using ggbiplot for PCA model2}
ggbiplot(pcobj = pca.vote.100,obs.scale = 1,var.scale = 1,groups =
data.vote.100$data.party,ellipse = TRUE,circle =
TRUE)+scale_color_manual(values = c("D" = "blue","R" = "red"))+ggtitle(label
= "ggbiplot for PC1 and PC2")
```




```
# Plotting the results
plot(pca.vote.100,type = 'l')
```



From the scatterplot,ggbiplot and varience, it is seen that PC1 accounts to 58.4% variance on the data with PC2 as 4.7% when analysing 100 sampled votes. Here Hanabusa and Evans, though democratic candidates has voting records similar to that of republicans. These data has contradictions between their party affiliations and voting records. Frankel is a staunch democrat based on the components PC1 and PC2 having high negative values. Peterson, being a democrat and Nunnelee, being a republican are closer to the centre and hence neutral in voting records.

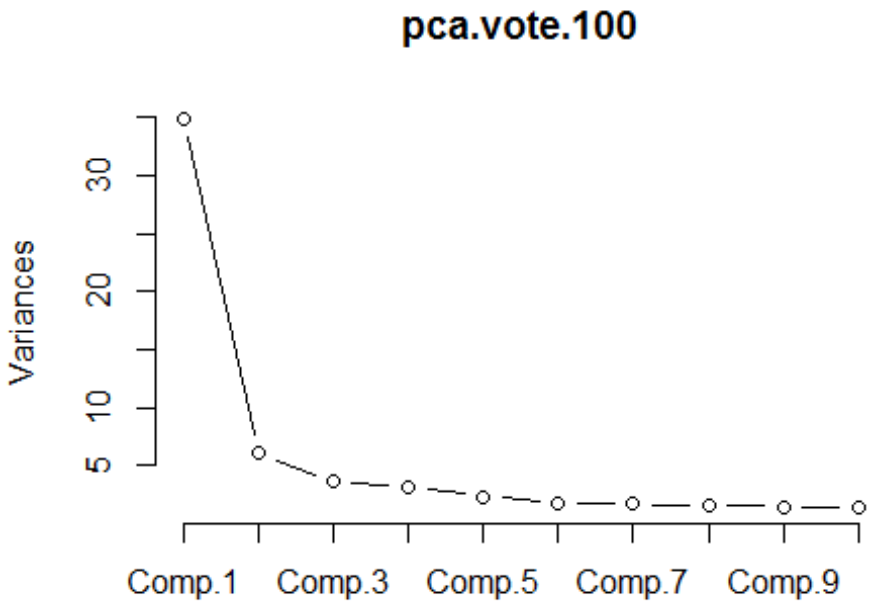
Question 10

(10 points (bonus)) Consider the Voting Data question above. Have the voting records of Congress always been so divided between party affiliation? Create a small multiples plot showing the results of PCA (colored by party) for the last 20 years. You will need to download prior Congress's roll calls from the sites indicated above.

```
# H.ord files function
House.func <- function(x){
    # Importing the data from ord file
    H.ord <- readKH(paste0("data/congress/",x))
    summary(H.ord)
    data.ord <- data.frame(H.ord$legis.data,H.ord$votes)

    # Ignoring the first five votes representing the U.S House of
Representatives
    data.ord <- data.ord[-c(7,8,9,10,11)]
    col <- ncol(data.ord)

    # Sampling and scaling 100 votes from the dataset
    set.seed(1)
    data.vote.100 <- sample(x = data.ord[7:col],size = 100)
    data.vote.100$party <- data.ord$party

    data.vote.100[,1:100] <- scale(data.vote.100[,1:100],center = TRUE,scale
= TRUE)

    # Data Preprocessing using PCA analysis for 100 votes
    pca.vote.100 <- princomp(x = data.vote.100[-101],cor = TRUE,scores =
TRUE)

    # Using ggbiplot for PCA model2}
    ggbiplot(pcobj = pca.vote.100,obs.scale = 1,var.scale = 1,groups =
data.vote.100$party,ellipse = TRUE,circle = TRUE)+scale_color_manual(values =
c("D" = "blue","R" = "red","Indep" = "black"))+ggtitle(label = "ggbiplot for
PC1 and PC2")

    # Plotting the varience
    plot(pca.vote.100,type = 'l')
}
```

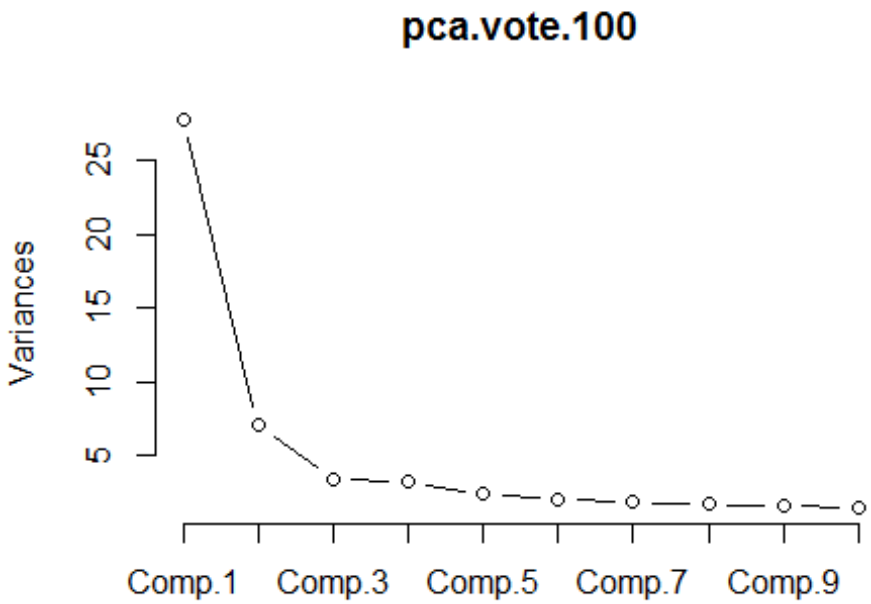
```
# Calling the functions
House.func("house_104.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 446 legislators and 1321 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  15358 320138 231414    389  21867      0
```



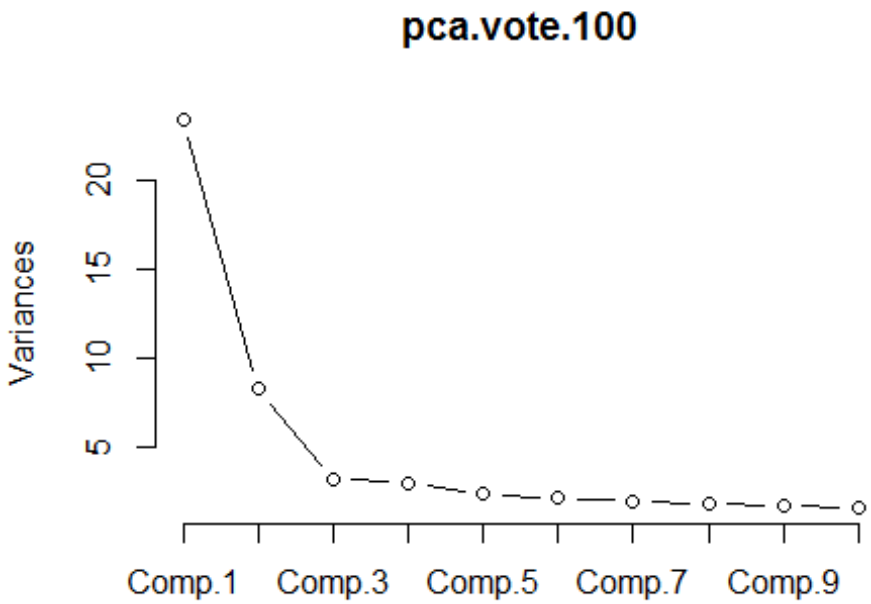

```
House.func("house_105.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 445 legislators and 1166 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  12681 302229 182871    456  20633      0
```



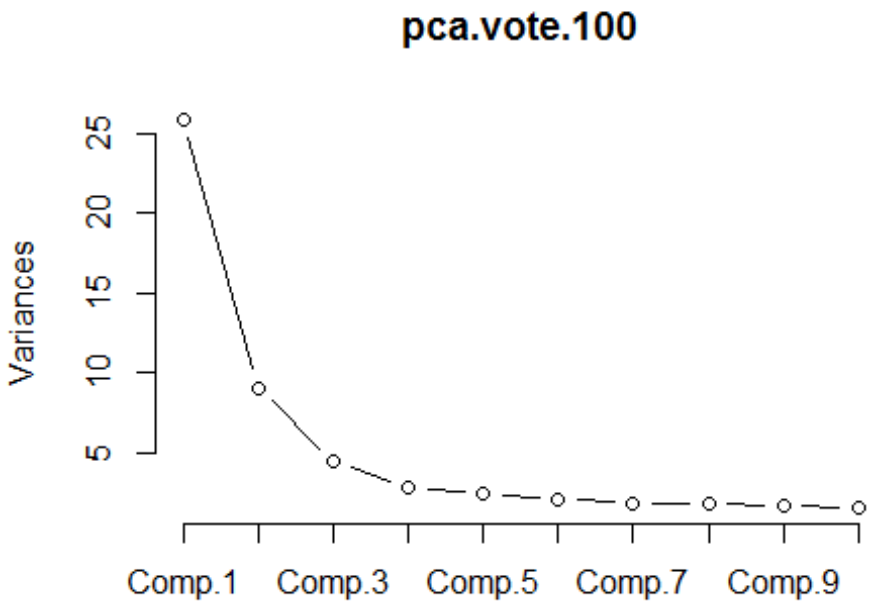

```
House.func("house_106.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 441 legislators and 1209 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##   6803 339494 160096    480  26296      0
```



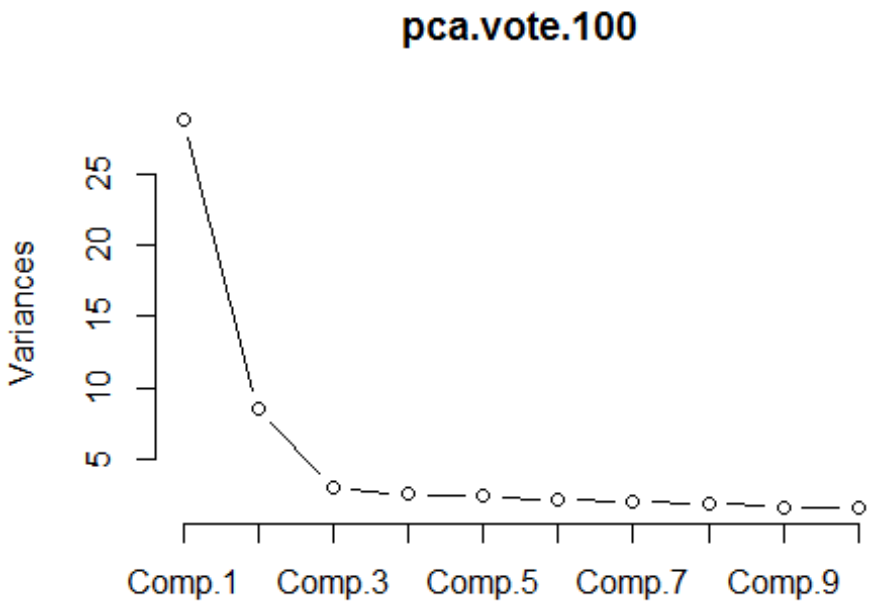

```
House.func("house_107.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 445 legislators and 990 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  10225 289457 120276    279  20313      0
```



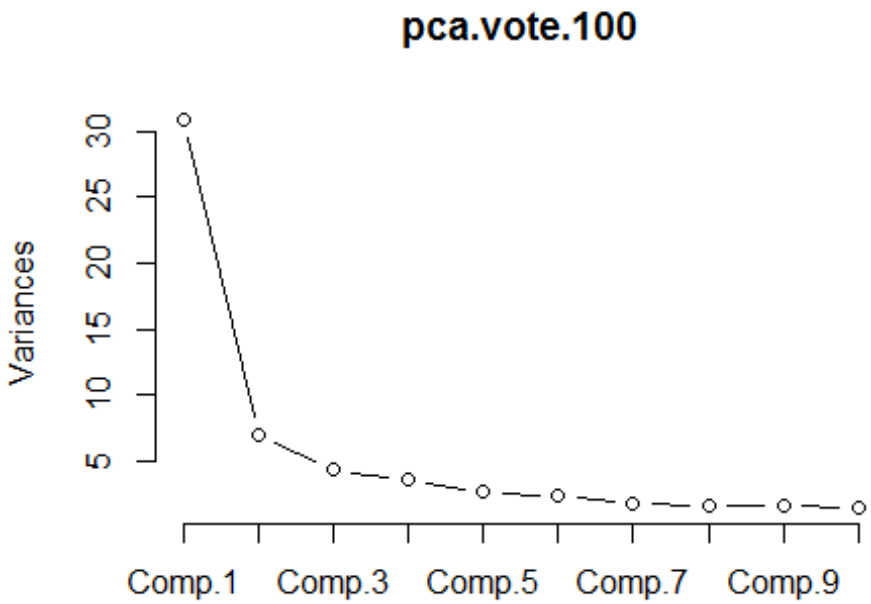

```
House.func("house_108.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 441 legislators and 1218 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##   6859 342940 159559    160  27620      0
```



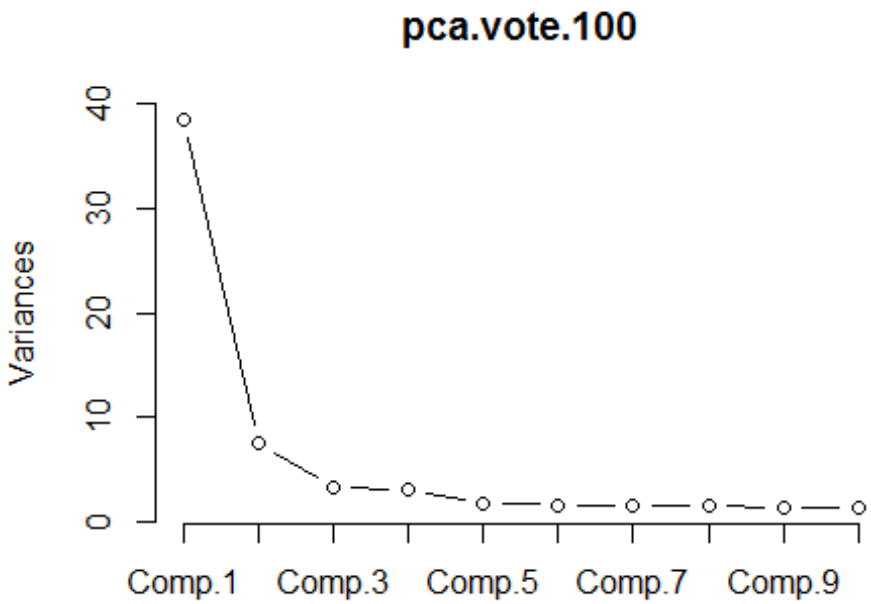

```
House.func("house_109.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 441 legislators and 1210 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##   8849 329763 172597    334  22067      0
```



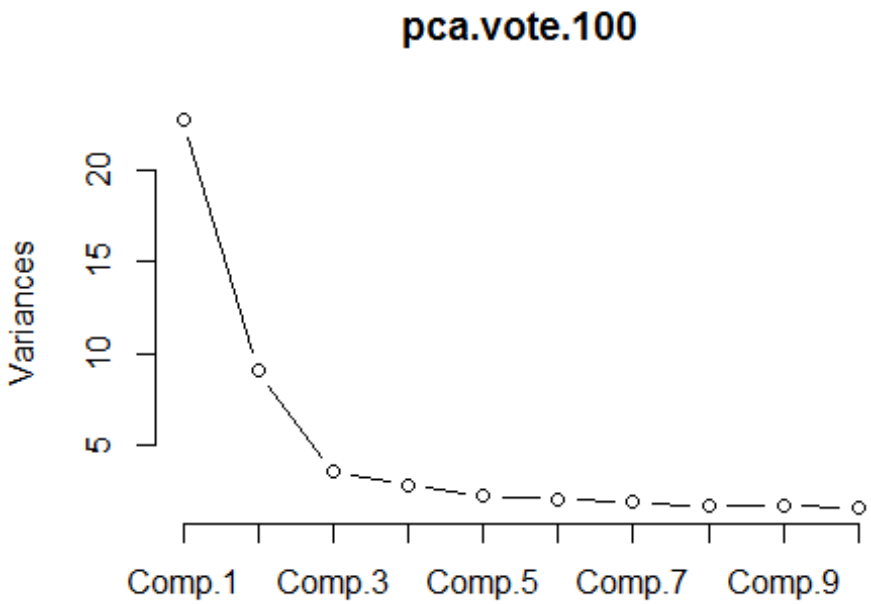

```
House.func("house_110.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 449 legislators and 1865 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  29521 502417 265710    684  39053      0
```



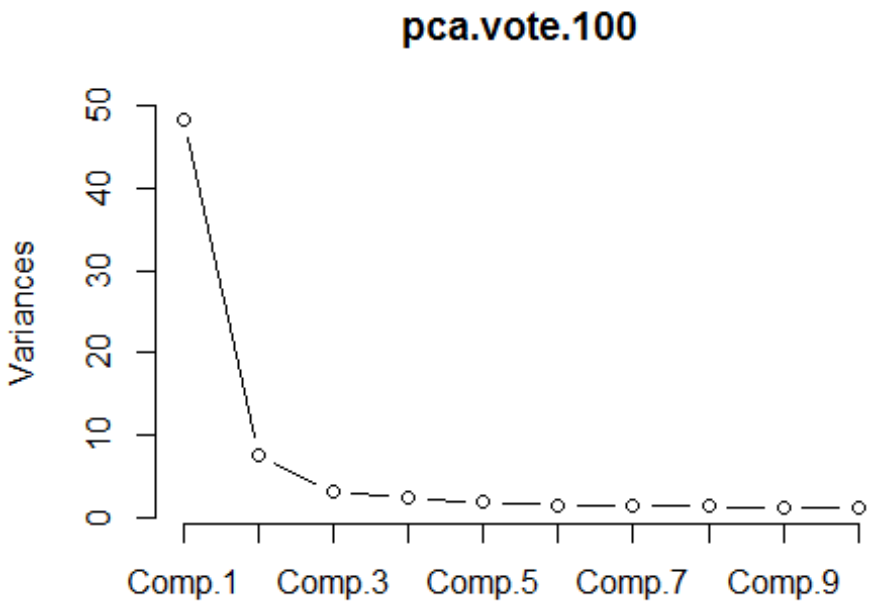

```
House.func("house_111.ord")
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 447 legislators and 1647 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  22750 501200 178634    777  32848      0
```

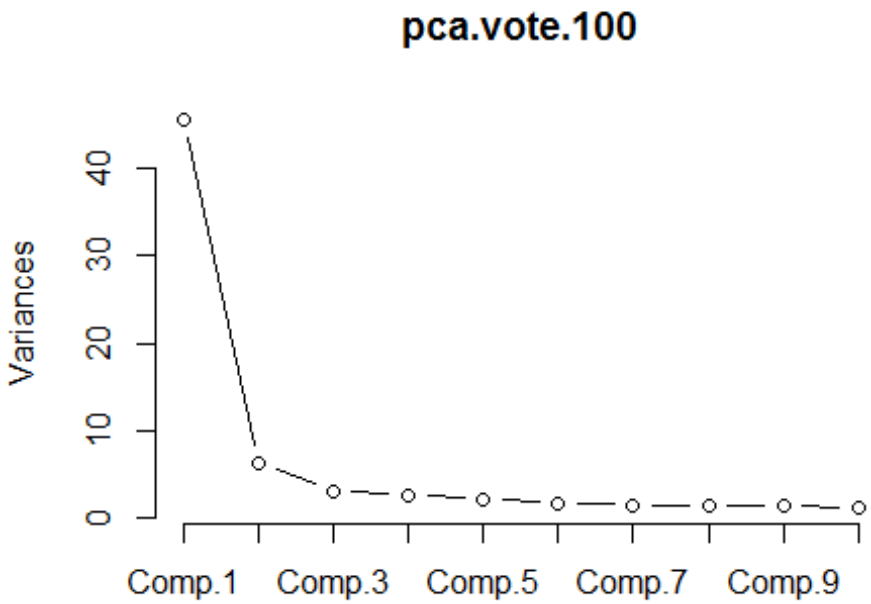



```
House.func("house_112.ord")
```

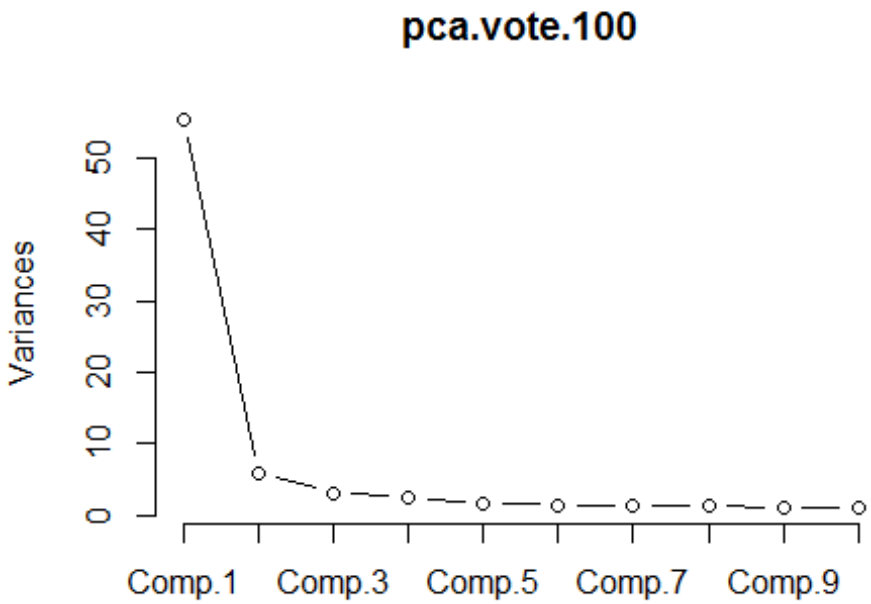
```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 446 legislators and 1602 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  21163 366339 299856    397  26737      0
```




```
House.func("house_113.ord")

## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 445 legislators and 1202 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  14576 295753 202943    290  21328      0
```




```
House.func("house_114.ord")

## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 442 legislators and 1322 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9   <NA>
##  11034 324283 227048     94  21865      0
```



From the ggbiplots, it is seen that the voting records is not divided between the party's affiliation for all the congress in House for the past 20 years as some democrats have similar voting records to that of the republicans as shown from the plot

End of assignemnt