hw2

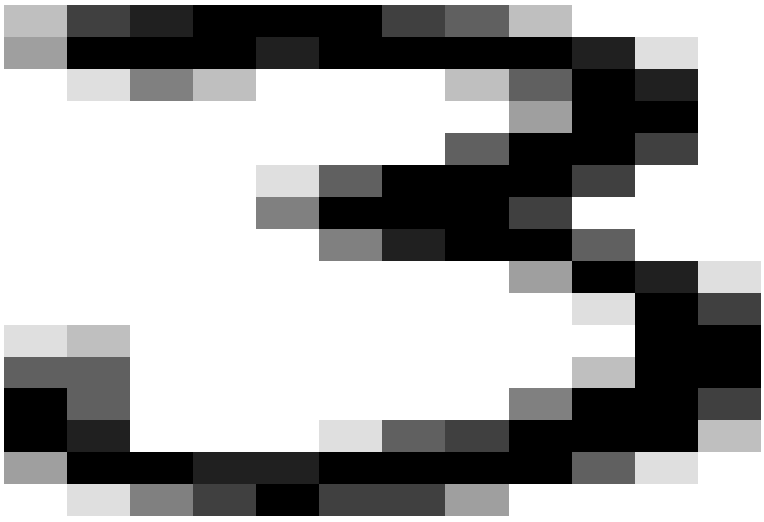
jing wen-jw4059

2/8/2021

```
# Define output.image function (Lab 2)
output.image<-function(vector) {
  digit<-matrix(vector, nrow=16, ncol=16)
  index= seq(from=16, to =1, by=-1)
  sym_digit = digit[,index]
  image(sym_digit, col= gray((8:0)/8), axes=FALSE)
}

# Read in digits features for 3,5,8
digit_3 <- read.table("train_3.txt",header = FALSE,sep=',')
digit_5 <- read.table("train_5.txt",header = FALSE,sep=',')
digit_8 <- read.table("train_8.txt",header = FALSE,sep=',')

# Define temporary data matrix
X_temp <- rbind(digit_3,digit_5,digit_8)
dim(X_temp)
```

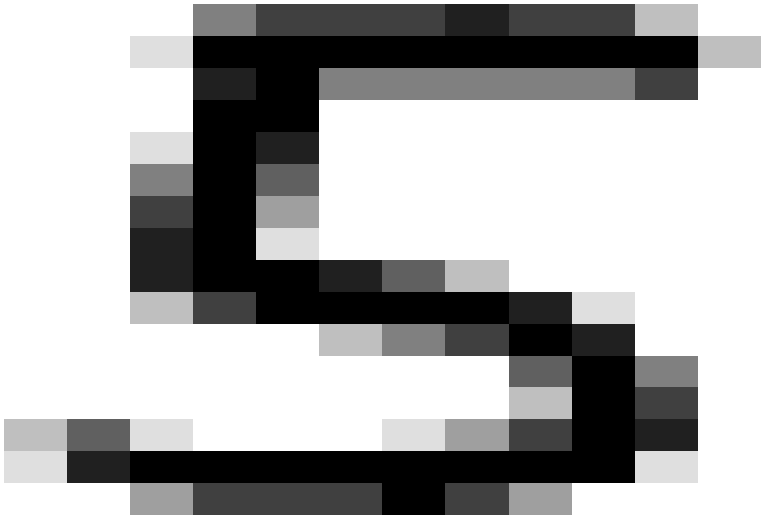
```
## [1] 1756  256
```

```
# Extract "test" cases
# Test case 1
ConstructCase_1 <- X_temp[20,]
output.image(as.matrix(ConstructCase_1))
```

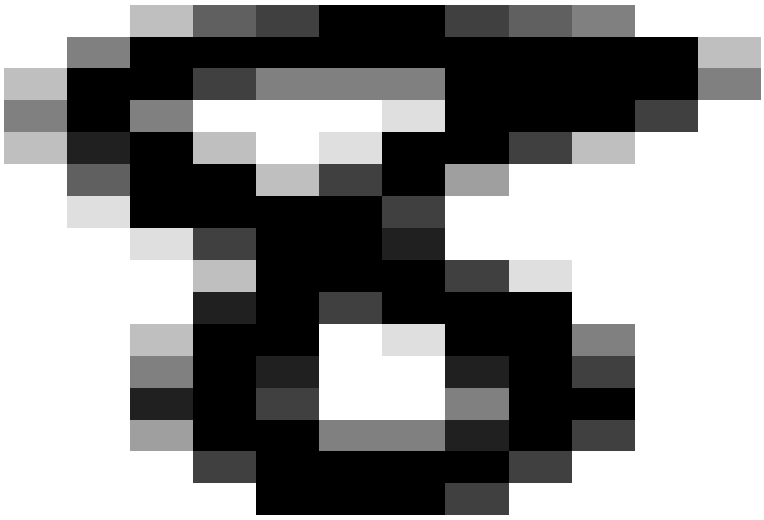


```
ConstructCase_1 <- unlist(ConstructCase_1) # Not needed but might be helpful

# Test case 2
```

```
ConstructCase_2 <- X_temp[735,]
output.image(as.matrix(ConstructCase_2))
```



```
ConstructCase_2 <- unlist(ConstructCase_2) # Not needed but might be helpful

# Test case 3
ConstructCase_3 <- X_temp[1260,]
output.image(as.matrix(ConstructCase_3))
```



```
ConstructCase_3 <- unlist(ConstructCase_3) # Not needed but might be helpful

# Remove cases 20, 735, 1260 from original dataframe
X <- X_temp[-c(20,735,1260),]
dim(X)
```

```
## [1] 1753  256
```

Q4

```
#1
zip=read.table("zip_test.txt",header =FALSE,sep='')
zip.test=zip[zip$V1=="3"|zip$V1=="5"|zip$V1=="8",]
```

```
digit_3$response="3"
digit_5$response="5"
digit_8$response="8"
X_temp.2 <- rbind(digit_3,digit_5,digit_8)
response=X_temp.2$response
list2=colnames(zip.test[,1:256])
zip.t=zip.test[,2:ncol(zip.test)]
names(zip.t)=list2
library(MASS)
model <- lda(X_temp,grouping=response)
model.pre1=predict(model, zip.t)
lda.class1=model.pre1$class
mean(lda.class1 != zip.test$V1) #test error
```

```
## [1] 0.08739837
```

```
model.pre2=predict(model, X_temp)
lda.class2=model.pre2$class
mean(lda.class2 != response) #train error
```

```
## [1] 0.01594533
```

```
#2
pca49<-prcomp(X_temp, rank. = 49)
response=as.numeric(response)
x.49<-cbind(response, pca49$x)
x.49<-data.frame(x.49)
t.49<-cbind(response=zip.test[,1], predict(pca49, zip.t))
t.49<-data.frame(t.49)
model2 <- lda(x.49[,-1],grouping=response)
model.pre3=predict(model2, t.49[,-1])
lda.class3=model.pre3$class
mean(lda.class3 != t.49$response) #test error
```

```
## [1] 0.08536585
```

```
model.pre4=predict(model2, x.49[,-1])
lda.class4=model.pre4$class
mean(lda.class4 != x.49$response) #train error
```

```
## [1] 0.04384966
```

```
#3
library(nnet)
mlr=multinom(response~.,data=x.49)
```

```
## # weights:  153 (100 variable)
## initial  value 1929.163179
## iter  10 value 399.115486
## iter  20 value 258.843521
## iter  30 value 168.405454
## iter  40 value 143.539876
## iter  50 value 138.085749
## iter  60 value 136.949732
## iter  70 value 136.548804
## iter  80 value 135.270883
## iter  90 value 133.736946
```

```
## iter 100 value 133.514892
## final  value 133.514892
## stopped after 100 iterations
```

```
model.pre5=predict(mlr, t.49[,-1])
mean(model.pre5 !=  t.49[,1]) #test error
```

```
## [1] 0.08536585
```

```
model.pre6=predict(mlr, x.49[,-1])
mean(model.pre6 != x.49[,1]) #train error
```

```
## [1] 0.02676538
```

q5

```
library(plyr)
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

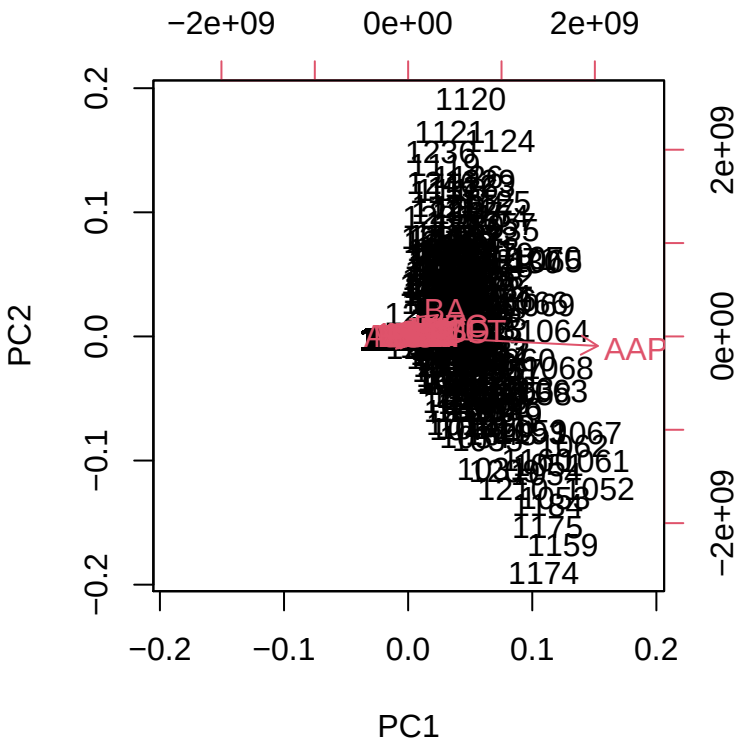
```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

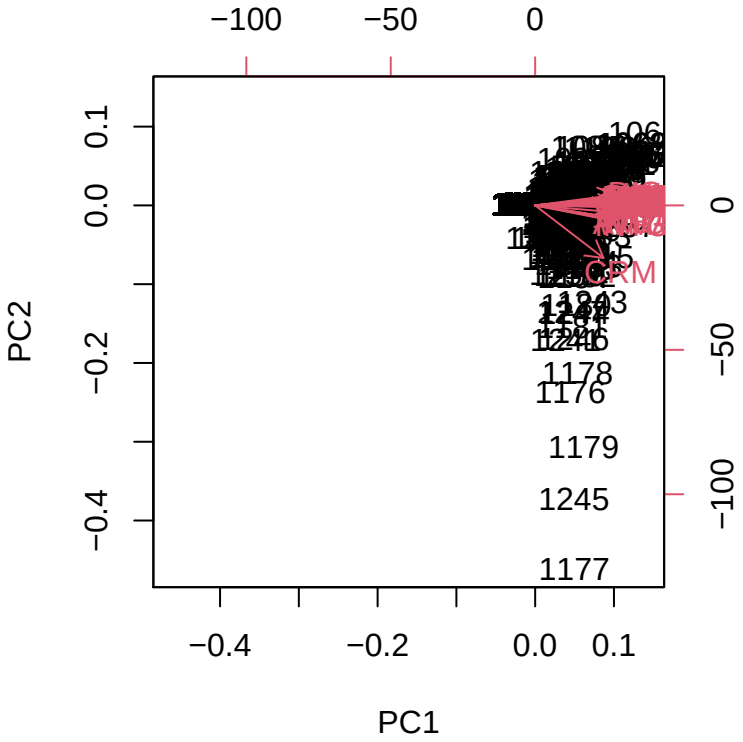
```
stock=function(x){
 data=getSymbols(x, auto.assign = FALSE, from ="2020-01-01", to = "2021-01-01")
 return(data)
}
list=c("AXP","AMGN","AAPL","BA","CAT","CVX","CSCO","KO","DIS","DOW","GS","HD","HON","INTC","JNJ","JPM",
dataf=llply(list,stock)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
dataf=as.data.frame(matrix(unlist(dataf), nrow=length(unlist(dataf[1]))))
colnames(dataf)=list
#2
pca=prcomp(dataf)
biplot(pca)
```

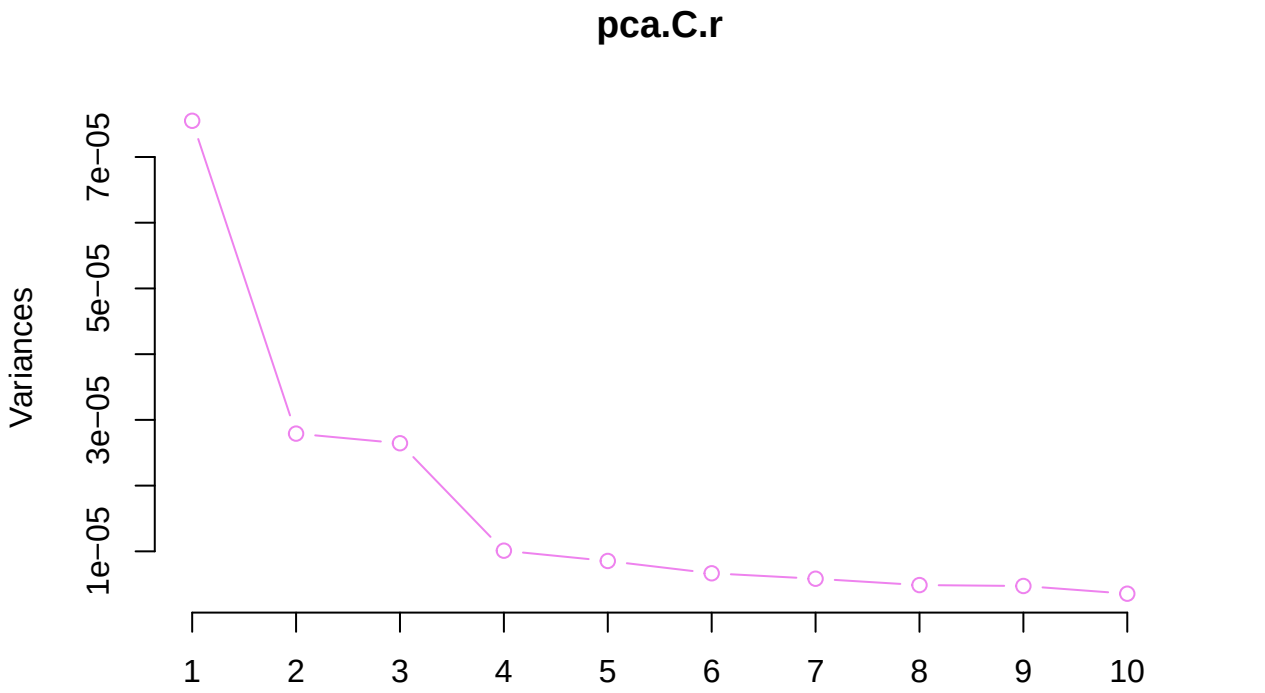



```
#3
dataf.C <- scale(dataf)
pca.C <- prcomp(dataf.C)
biplot(pca.C)
```

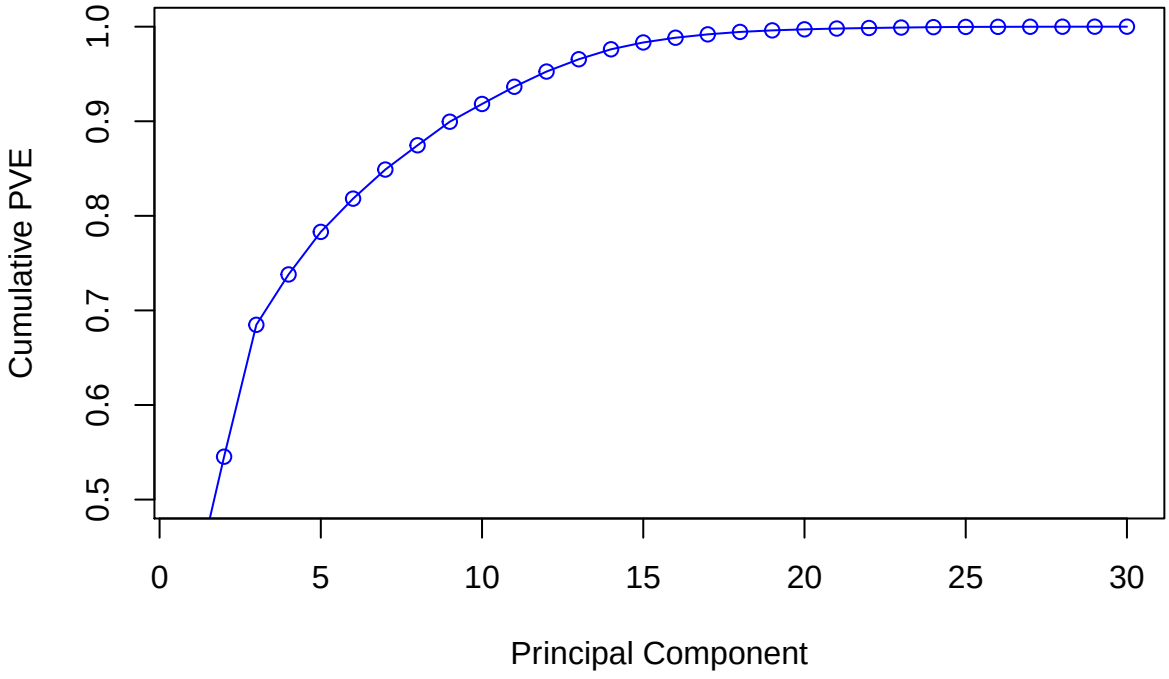



```
#4
ret=c()
```

```r
ret.r=function(x){
  ret=diff(x)/x[1:(length(x)-1)]
  return(ret)
}
dataf.r=lapply(dataf,ret.r)
dataf.r=as.data.frame(matrix(unlist(dataf.r), nrow=length(unlist(dataf.r[1]))))
colnames(dataf.r)=list
dataf.C.r <- scale(dataf.r)
pca.C.r <- prcomp(t(dataf.C.r))
screeplot(pca.C.r,type="lines",col="violet")
```



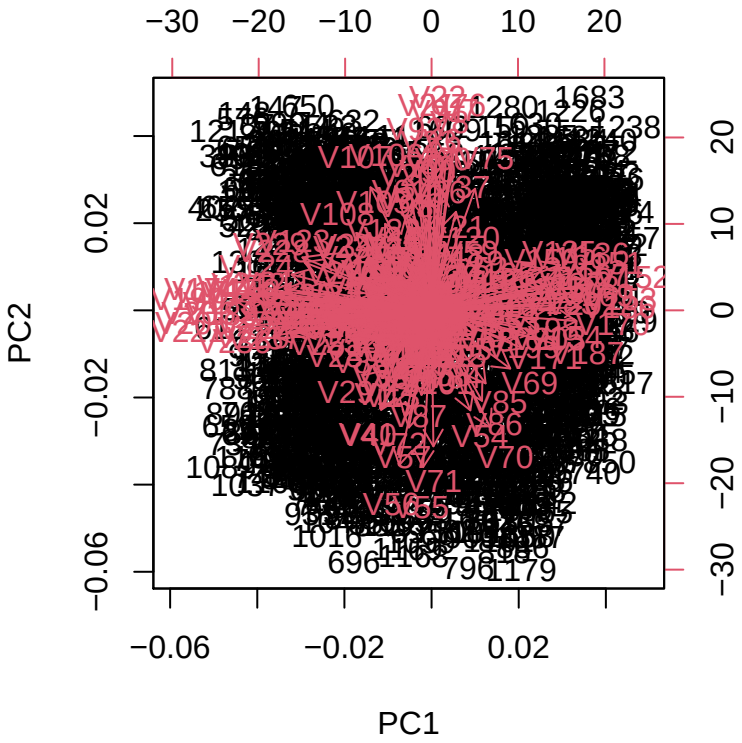

```r
CPVE <- (cumsum((pca.C.r$sdev)^2)/sum((pca.C.r$sdev)^2))
plot(1:30,CPVE,type="l",col="blue",
     ylim=c(.5,1),xlab="Principal Component",ylab="Cumulative PVE")
points(1:30,CPVE,col="blue")
```



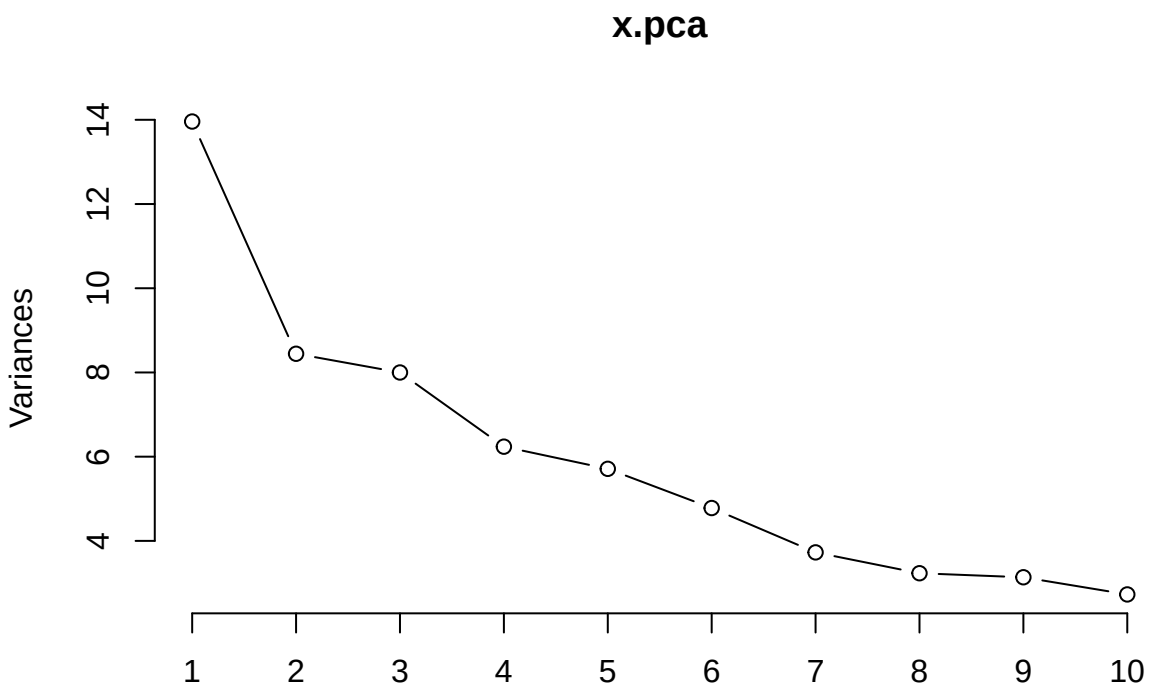
#2

from the biplot, stock AAF seems to have a larger eigen value than the other stocks.AAF is the only that fully show up other than the other stocks in the plot. #3, after scale the variables, CRM turns out to be important from the biplot #4 from screeplot, it seems PC1 caputres the most variation in PCA.for returns, all the stocks have the smiliar direction.If each stock were fluctuating up and down randomly and independent of all the other stocks,i would expect scree plot to be a sraight line. q6

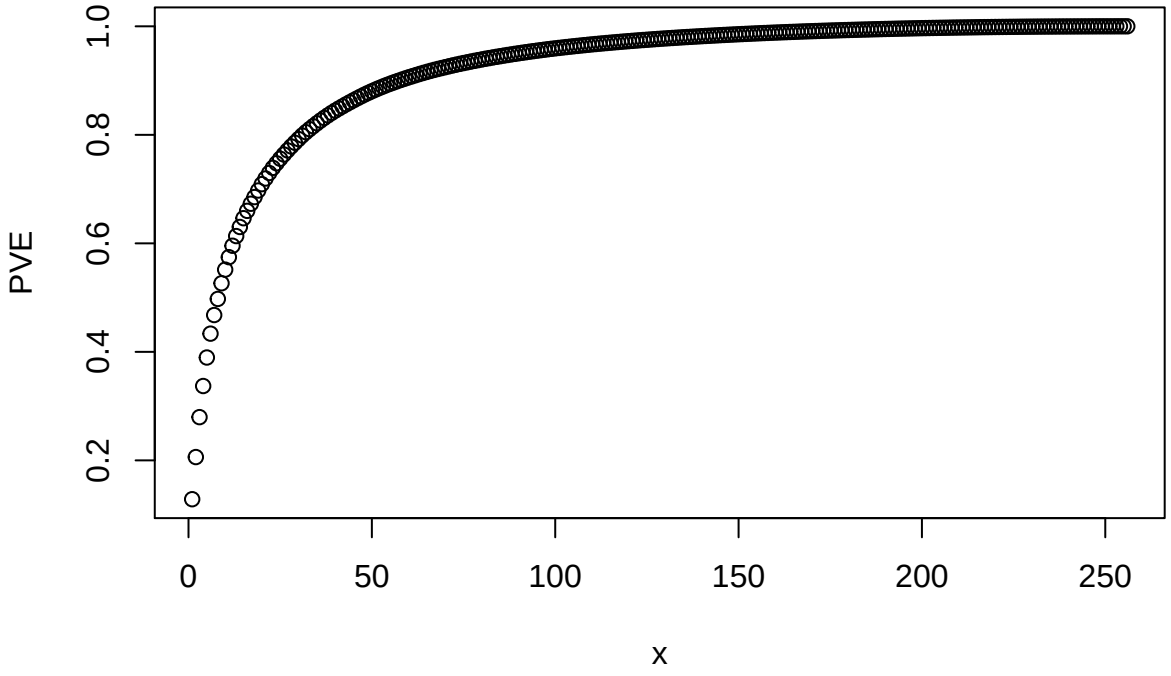
```
x.c <- scale(X,center=TRUE,scale=FALSE)
x.pca=prcomp(x.c)
biplot(x.pca)
```



```r
screeplot(x.pca,type="l")
```



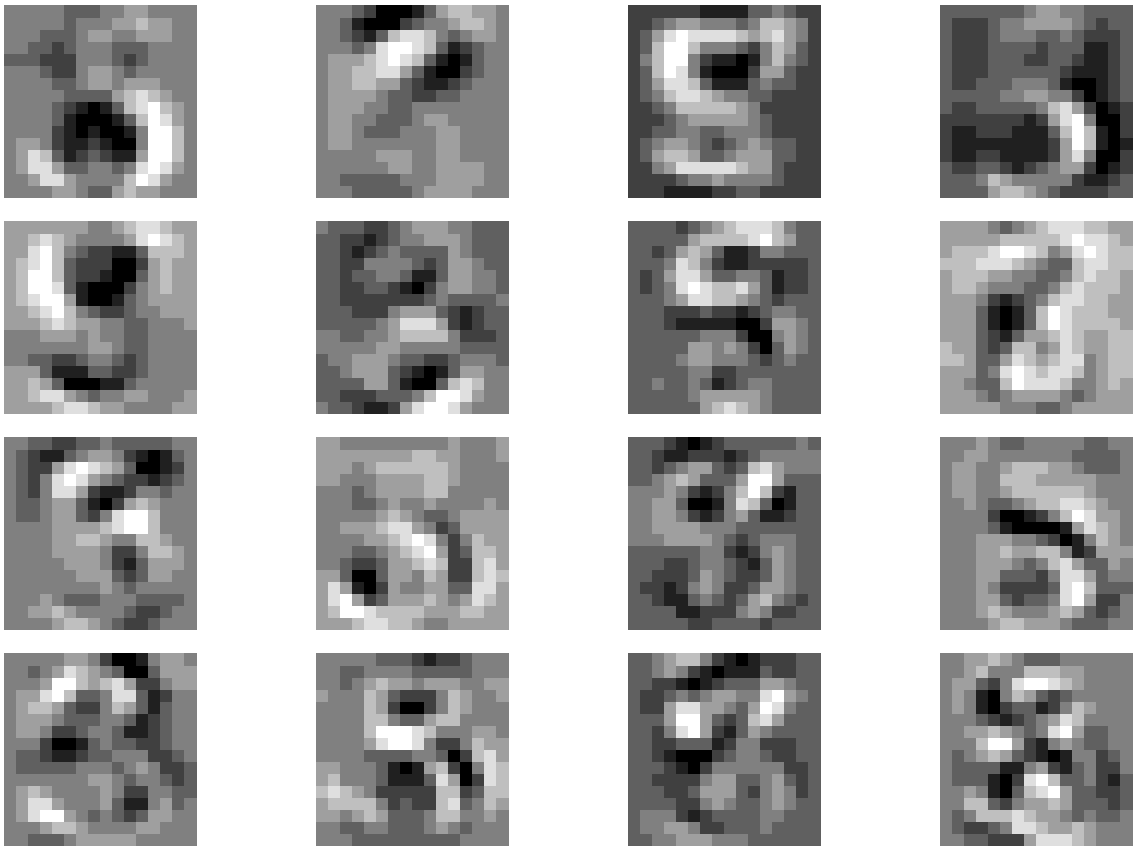

```r
#2
PVE <- (cumsum((x.pca$sdev)^2)/sum((x.pca$sdev)^2))
x=c(1:256)
plot(x,PVE)
```




```r
sum(PVE<0.9)
```

```
## [1] 57
```

```r
#3
xr=x.pca$rotation
par(mfrow=c(4,4),pin = c(1,1))
for (i in 1:16){
  output.image(as.matrix(xr[,i]))
}
```



```r
#4
list=c(3,58,256)
construct1 <- rep(0,3)
construct2 <- rep(0,58)
construct3 <- rep(0,256)
for (j in list){
   for (i in 1:j) {
  construct1 <- construct1 + sum(x.pca$rotation[,i]*ConstructCase_1)*x.pca$rotation[,i]
   }
   output.image(construct1)
}
```

```
## Warning in construct1 + sum(x.pca$rotation[, i] * ConstructCase_1) *
## x.pca$rotation[, : longer object length is not a multiple of shorter object
## length
```

```r
for (j in list){
   for (i in 1:j) {
  construct2 <- construct2 + sum(x.pca$rotation[,i]*ConstructCase_2)*x.pca$rotation[,i]
   }
   output.image(construct2)
```

```
}
```

```
## Warning in construct2 + sum(x.pca$rotation[, i] * ConstructCase_2) *
## x.pca$rotation[, : longer object length is not a multiple of shorter object
## length
```

```
for (j in list){
   for (i in 1:j) {
  construct3 <- construct3 + sum(x.pca$rotation[,i]*ConstructCase_3)*x.pca$rotation[,i]
   }
   output.image(construct3)
}
```