CA2

Chris Eason, Dongook Kim, Rongyu Zhang

The file CA2.csv contains 100 observations on 12 unknown variables. Consider this as some data matrix X. Using Singular Value Decomposition, find lower rank approximations of X for all ranks from 1 – 12

```
X <- read.csv("CA2.csv")
```

```
s <- svd(X)
svd_approx <- function(rank){
  app <- s$u[,1:rank] %*% as.matrix((diag(s$d))[1:rank, 1:rank]) %*% t(s$v[, 1:rank])
  colnames(app) <- c("V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8", "V9", "V10", "V11", "V12
  return(app)
}

X_k <- list()
for (i in 1:12){
  X_k[[i]] <- svd_approx(i)
}

# For each Approximation of X in Rank K
delta <- list()
for(i in 1:12){
  delta[[i]] <- X - X_k[[i]]
}
```

Question 1

For each approximation $\tilde{X}_k$ of rank k, calculate the error, $\Delta_k = X - \tilde{X}_k$

```
mean_delta <- colMeans(delta[[4]])
as.matrix(mean_delta)
```

```
            [,1]
V1   0.012350793
V2  -0.011464781
V3   0.132007862
V4   0.015116997
V5  -0.041999346
V6   0.012416893
V7   0.009807781
V8  -0.020731376
V9   0.060532151
V10 -0.043384888
V11  0.138223796
V12 -0.037550849
```

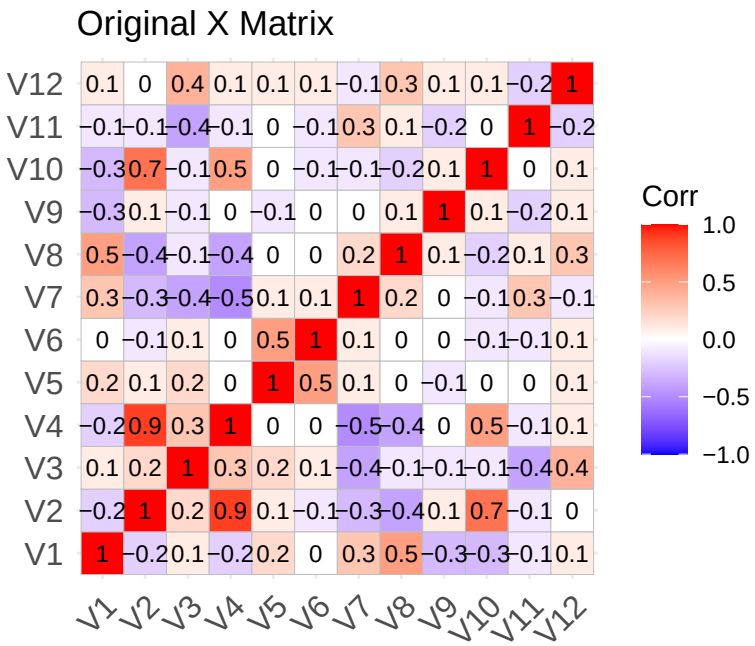
Question 2

Compare the correlation matrix of X with that of $\tilde{X}_2$and briefly interpret.

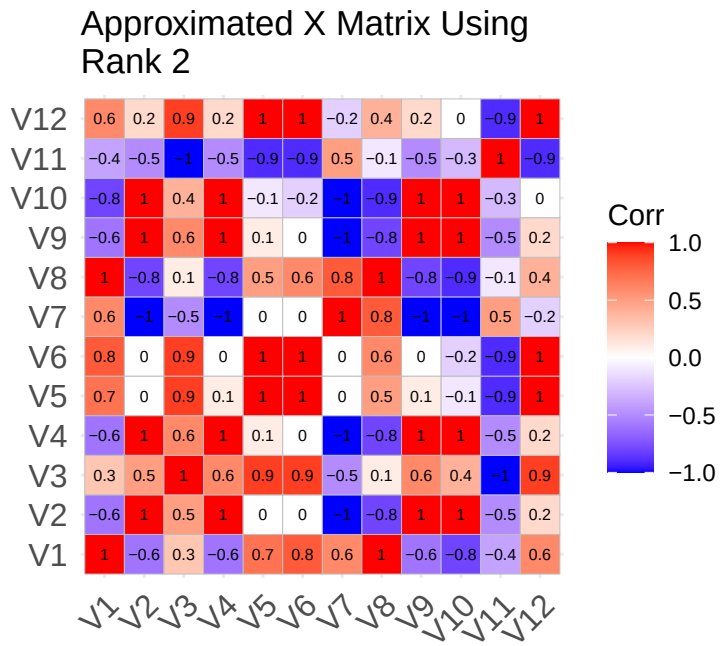
```
library(ggcorrplot)
```

```
Loading required package: ggplot2
```

```
library(patchwork)

x_cor <- round(cor(X), 1)
p1 <- ggcorrplot(x_cor, "square", lab = TRUE, title = "Original X Matrix", lab_size = 3)

x_2_cor <- round(cor(X_k[[2]]), 1)
p2 <- ggcorrplot(x_2_cor, "square", lab =TRUE, title = "Approximated X Matrix Using \nRank 2

p1
```




```
p2
```



Interpretation:

1. The original data shows that most of the variables are uncorrelated, the rank 2 approximation shows that they are corerlated. For example with V6 and V8 in the original matrix are uncorrelated while the rank 2 approximation correlation coefficient is 0.6.
2. The original data shows slight correlations between variables with each other while the rank 2 approximation of X exagerates these correaltions. For example with V12 and V11 in the original matrix is -0.2 while the rank 2 approximation is -0.9.
3. The rank 2 captures the 2 largest singular values and therefore attempts to capture a majority of the variation in the original X matrix but discards the other higher order variations. Therefore the rank 2 approximation has altered some of the structures in the data and has affected the relationships between variables, changing their correlation coefficients.

Question 3

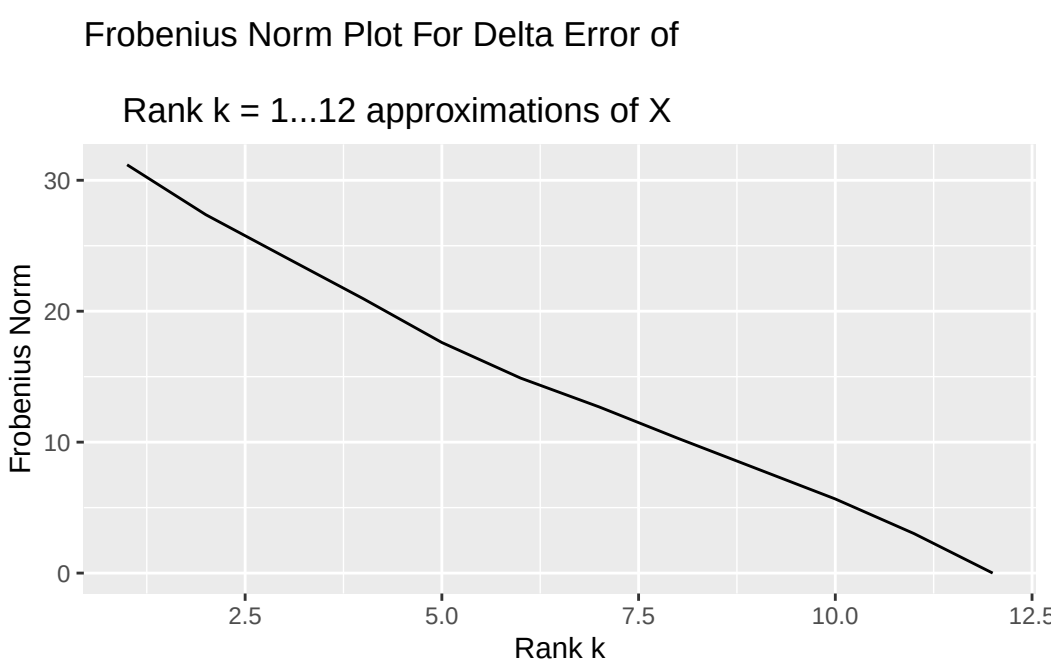
Calculate the Frobenius norm, defined as

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}|aij|^2}$$

for Δ_k, k =1,2....12. Plot the Frobenius norm as a function of k and briefly describe your findings.

```
FN <- c()
Frobenius_Norm <- function(rank){
   FN[rank]<- sqrt(sum(delta[[rank]]^2))
}

for(i in 1:12){
  FN[i] <- Frobenius_Norm(i)
}

df <- data.frame(x = 1:12, y = FN)
library(ggplot2)
ggplot(df, aes(x=x, y=y))+
  geom_line() +
  labs (
    title = "Frobenius Norm Plot For Delta Error of \n
    Rank k = 1...12 approximations of X",
```

```
    x = "Rank k",
    y = "Frobenius Norm"
  )
```



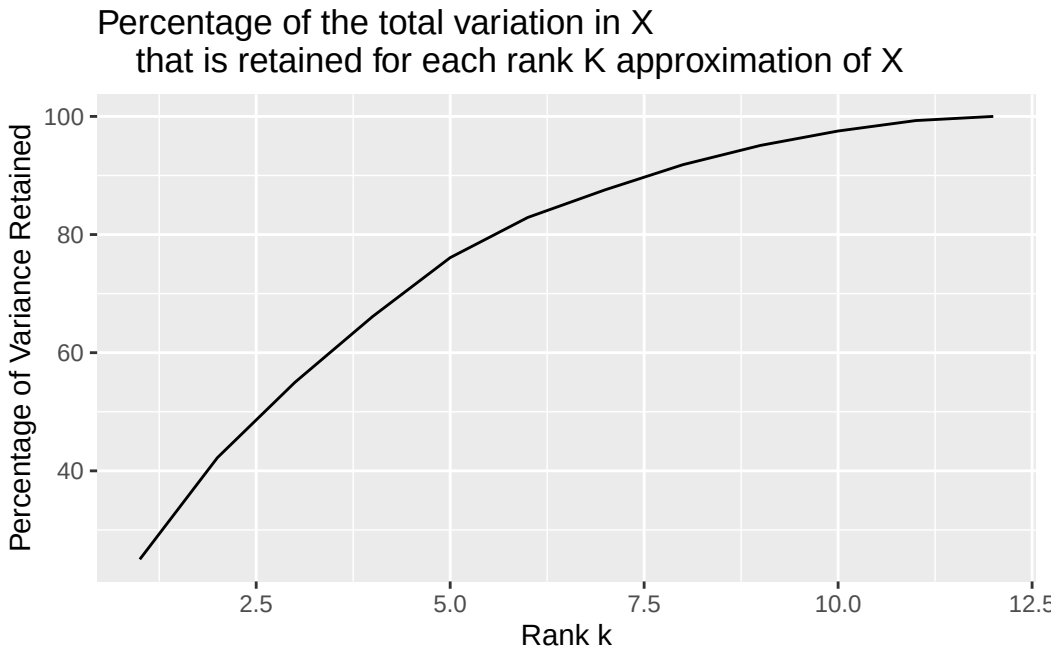
Interpretation:

1. Decreasing (almost linear) trend of the Frobenius Norm as the the approximation of rank K increases.
2. In other words, the delta error between the original matrix and the approximated rank K matrix decreases as the approximation gets closer to rank 12, where the original data set and the approximation are equal and the error is zero.

Question 4

Plot the percentage of the total variation in X retained in $\tilde{X}_k$ for K =1,…12. Again, briefly interpret.

```r
total_var <- sum(s$d^2)
retained_var <- c()

for (i in 1:12){
  retained_var[i] <- ((sum(s$d[1:i]^2)) / total_var) * 100
}

df <- data.frame(x = 1:12, y = retained_var)
library(ggplot2)
ggplot(df, aes(x=x, y=y))+
  geom_line() +
  labs (
    title = "Percentage of the total variation in X
    that is retained for each rank K approximation of X",
    x = "Rank k",
    y = "Percentage of Variance Retained"
  )
```




```r
print(paste("Percentage of variance retained rank 7:", round(retained_var[8],2)))
```

```
[1] "Percentage of variance retained rank 7: 91.83"
```

Interpretation

1. The amount of variation in the original matrix retained increasing at a decreasing rate by the rank k approximation as the the approximation of rank K increases.
2. This makes sense as we use more eigenvalues to approximate our original matrix, we keep more and more of the variation in the original matrix.
3. Shows that by the time the rank k = rank 8, the approximation basically retains 92% of the variation of in the original matrix. Therefore using rank 8 approximation captures most of the data structures in the original matrix while being in lower dimensions and removing some of the linear dependencies between variables.