

# Econometrics Homework 1

b10901069 Chinying Lin

September 2024

## Environment Variables

```
Data <- read.csv("Equity_Premium.csv")
Matrix <- as.matrix(Data)
TimeMatrix <- Matrix[, 1, drop = FALSE]
YMatrix <- Matrix[, 2, drop = FALSE]
XMatrix <- Matrix[, -c(1, 2), drop = FALSE]
```

## Problem 1

### Time Series Function Setup

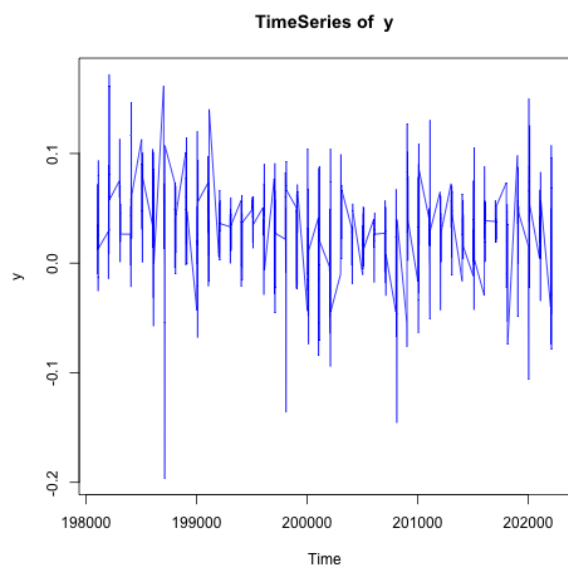
```
TimeSeries <- function(X, yLab){
  return(plot(TimeMatrix, X, type = "l", col = rgb(0, 0, 1), xlab = "Time",
             ylab = yLab, main = paste("TimeSeries of ", yLab)))
}
```

### Histogram Function Setup

```
Histogram <- function(X, FigureName){
  return(hist(X, probability = TRUE, main = FigureName, xlab = "Values",
             col = rgb(0.1, 0.3, 0.4, 0.5), border = "white", breaks = 20))
}
```

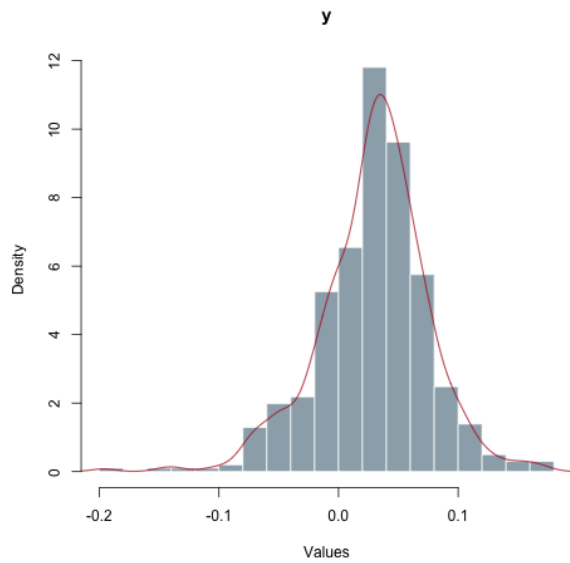
- Time Series Plot of y

```
png(paste(colnames(YMatrix), "_TimeSeriesPlot.png", sep = ""))
TimeSeries(YMatrix, colnames(YMatrix))
dev.off()
```



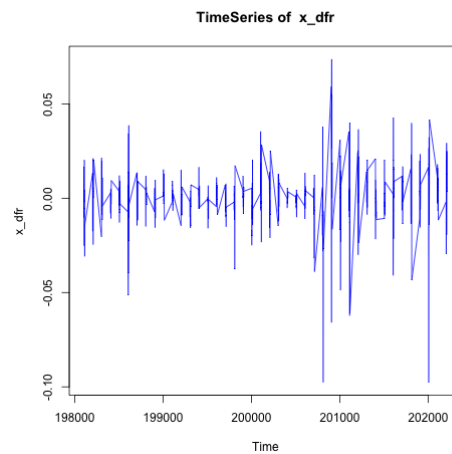
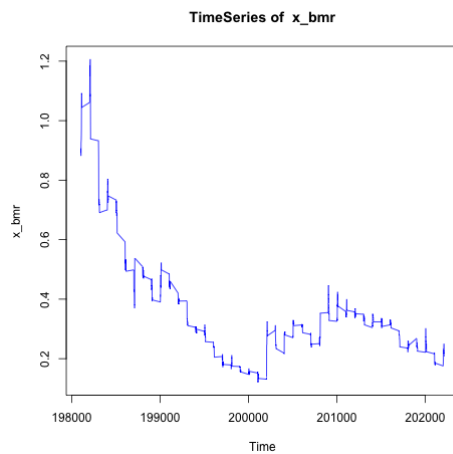
- Histogram of y

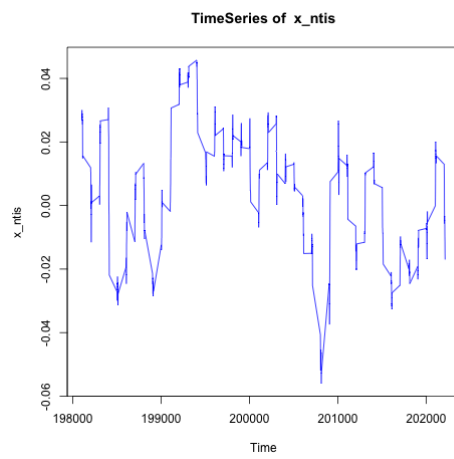
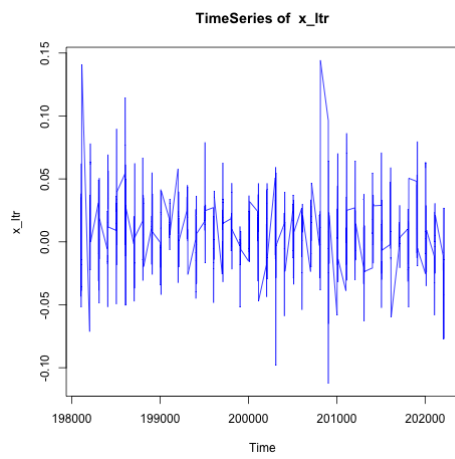
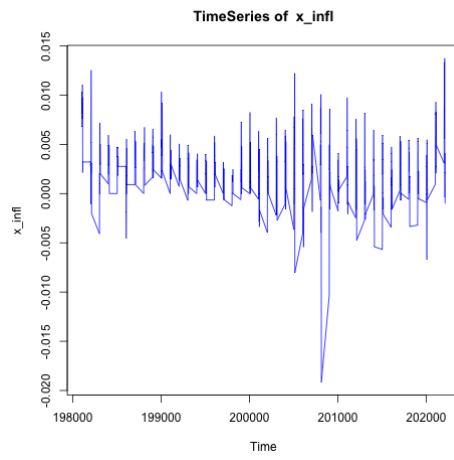
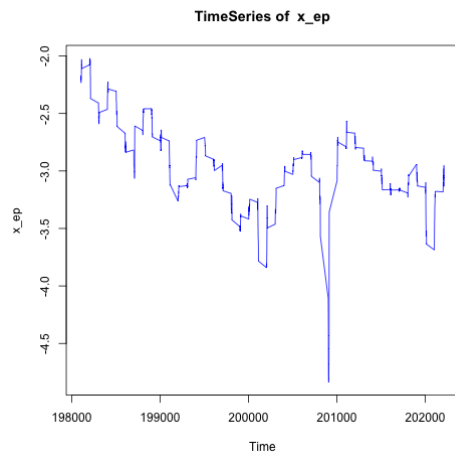
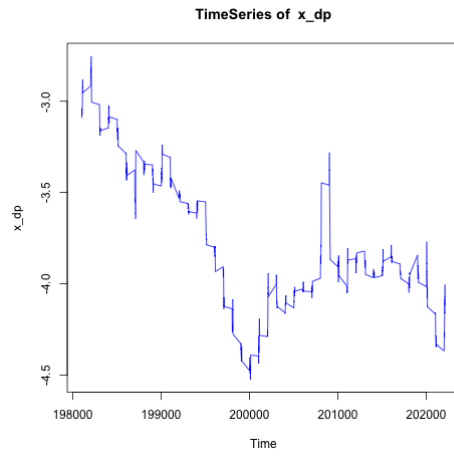
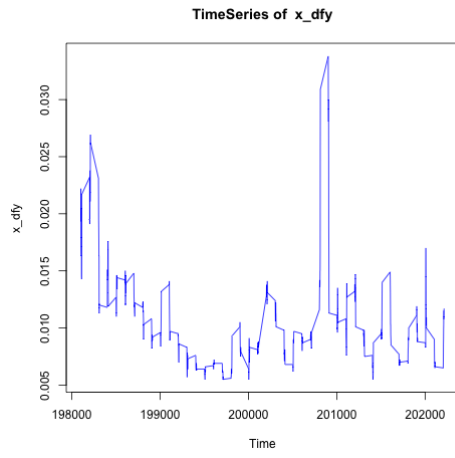
```
png(paste("Y_Histogram.png"))
Histogram(YMatrix, colnames(YMatrix))
lines(density(YMatrix), col = rgb(0.7, 0.1, 0.1, 0.7), lwd = 1.5)
dev.off()
```

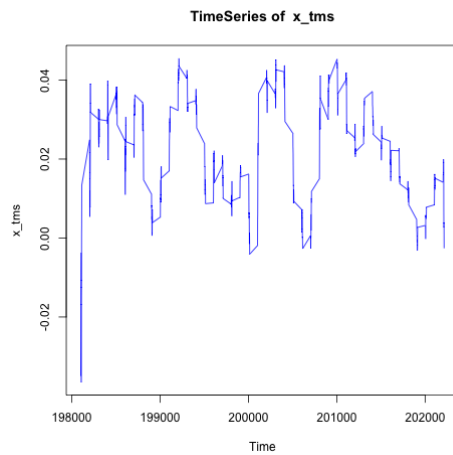
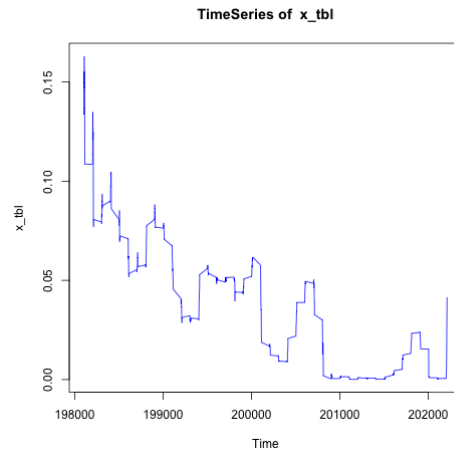
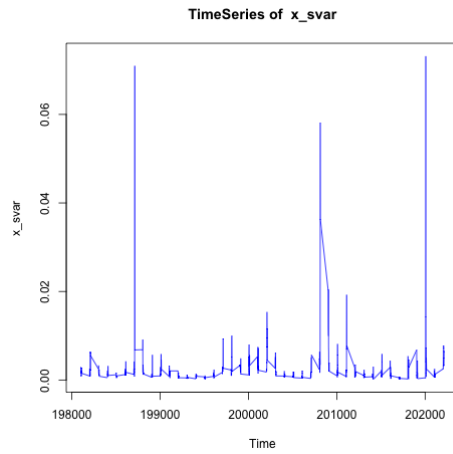


- Time Series Plot of X

```
for(i in 1:ncol(XMatrix))
{
  png(paste(colnames(XMatrix)[i], "_TimeSeriesPlot.png", sep = ""))
  TimeSeries(XMatrix[, i], colnames(XMatrix)[i])
  dev.off()
}
```

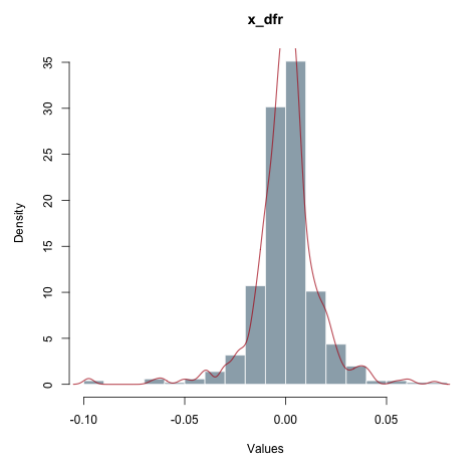
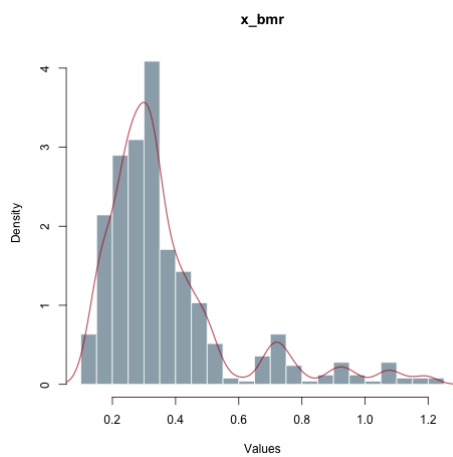


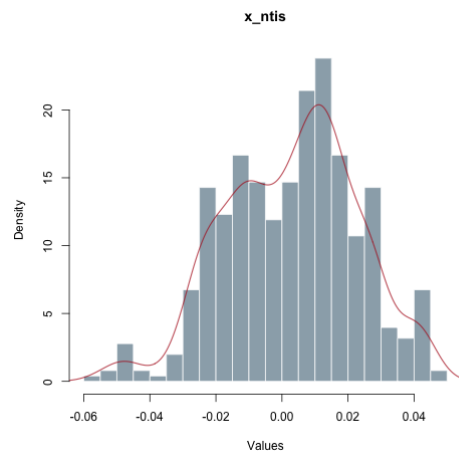
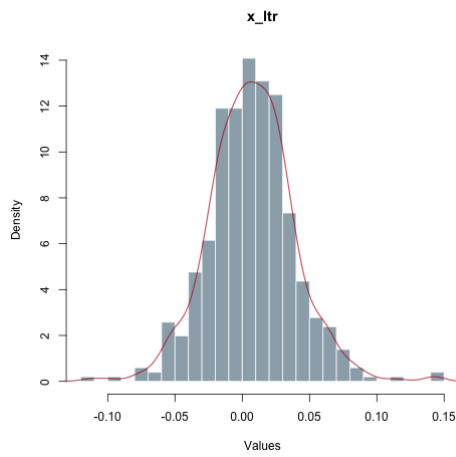
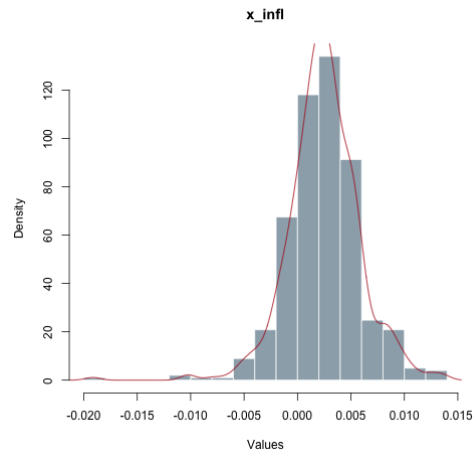
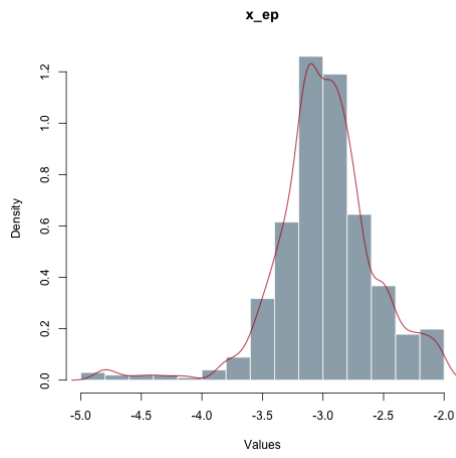
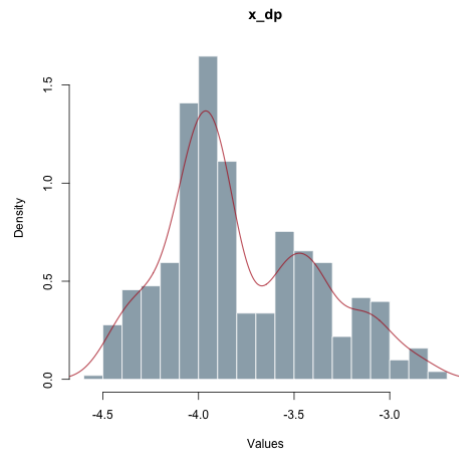
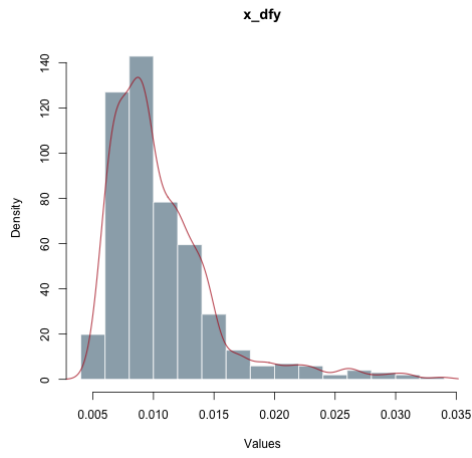


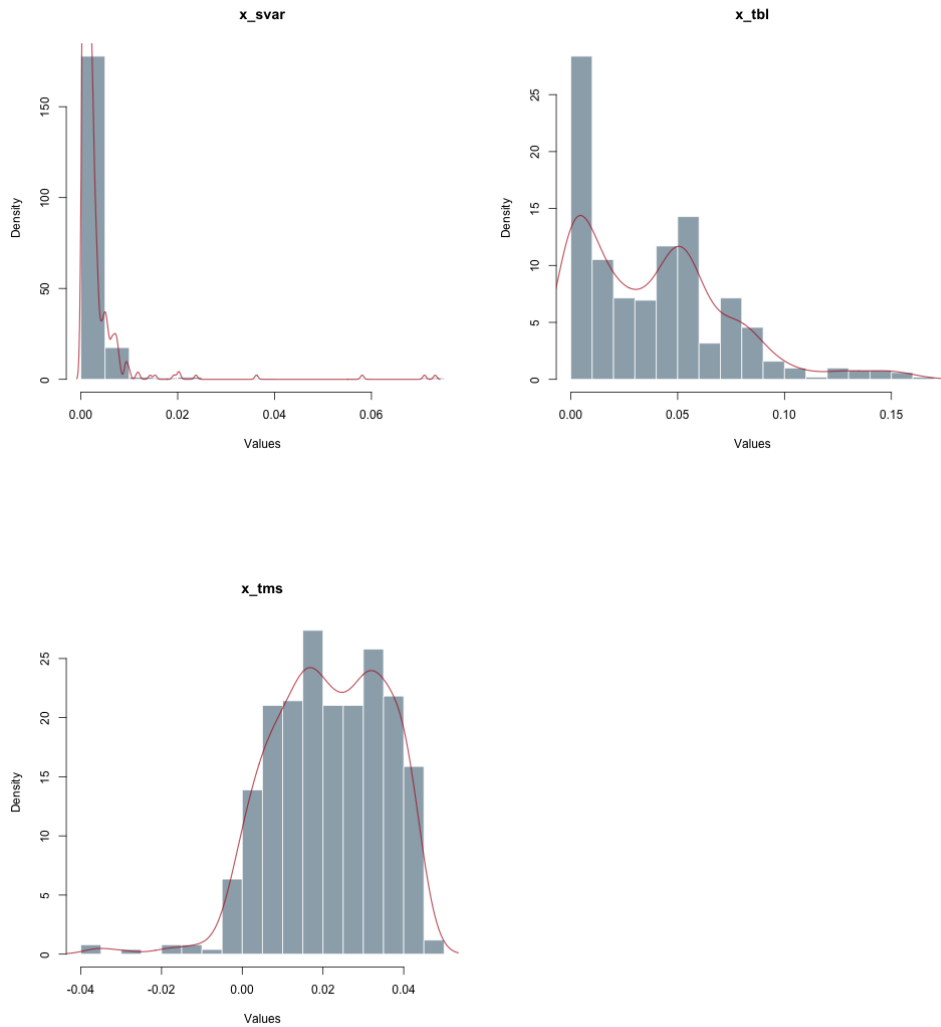


## • Histogram of X

```
for(i in 1:ncol(XMatrix))
{
  png(paste(colnames(XMatrix)[i], "_Histogram.png", sep = ""))
  Histogram(XMatrix[, i], colnames(XMatrix)[i])
  #lines(density(rnorm(504)), col = rgb(0.7, 0.1, 0.1, 0.7), lwd = 1.5)
  lines(density(XMatrix[, i]), col = rgb(0.7, 0.1, 0.1, 0.7), lwd = 1.5)
  dev.off()
}
```







## Problem 2

### 2.1

```
Matrix_2_1 <- XMatrix %*% solve(t(XMatrix) %*% XMatrix) %*% t(XMatrix)
TraceOfMatrix_2_1 <- sum(diag(Matrix_2_1))
print(TraceOfMatrix_2_1)
```

```
> print(TraceOfMatrix_2_1)
```

```
[1] 11
```

### 2.2

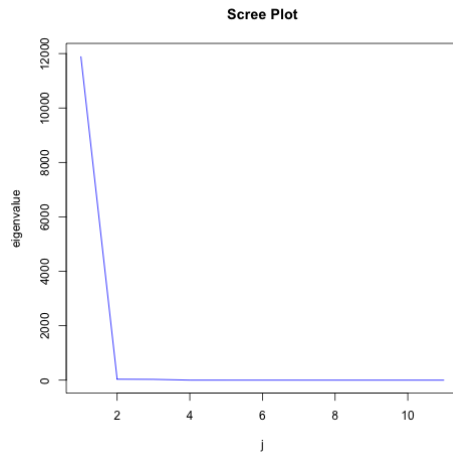
```
I_n <- diag(504)
Matrix_2_2 <- I_n - Matrix_2_1
TraceOfMatrix_2_2 <- sum(diag(Matrix_2_2))
print(TraceOfMatrix_2_2)
```

```
> print(TraceOfMatrix_2_2)
```

```
[1] 493
```

## Problem 3

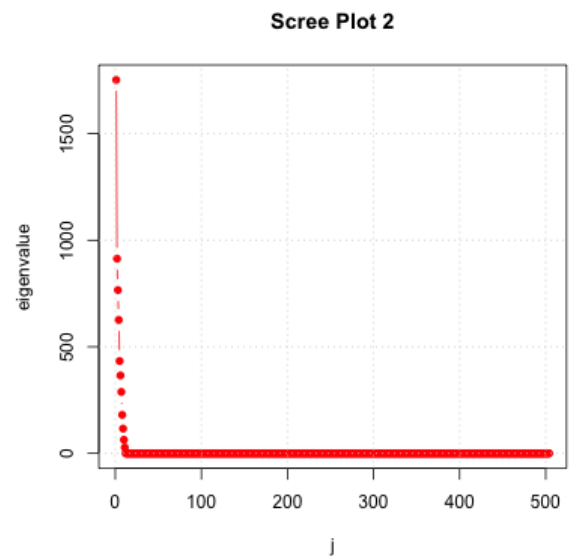
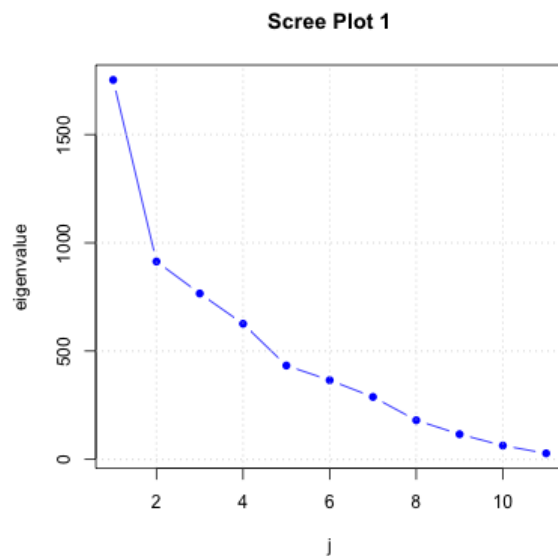
```
eigenX <- eigen(t(XMatrix) %*% XMatrix)
print(eigenX$values)
png(paste("ScreePlot_Problem3.png"))
plot(1:11, eigenX$values, type = "l", col = rgb(0, 0, 1), xlab = "j",
     ylab = "eigenvalue", main = "Scree Plot")
dev.off()
```



## Problem 4

```
Standardized_X <- scale(XMatrix)
eigen_4_1 <- eigen(t(Standardized_X) %*% Standardized_X)
eigen_4_2 <- eigen(Standardized_X %*% t(Standardized_X))

png(paste("ScreePlot_Problem4.png"), width = 800, height = 400)
layout(matrix(c(1, 2), 1, 2, byrow = TRUE), widths = c(1, 1))
plot(1:11, eigen_4_1$values, type = "b", col = rgb(0, 0, 1), xlab = "j",
     ylab = "eigenvalue", main = "Scree Plot 1", pch = 16)
grid()
plot(1:504, eigen_4_2$values, type = "b", col = rgb(1, 0, 0), xlab = "j",
     ylab = "eigenvalue", main = "Scree Plot 2", pch = 16)
grid()
dev.off()
```



## Problem 5

```
MatrixA <- solve(t(Standardized_X) %*% Standardized_X)
EigenA <- eigen(MatrixA)
P <- EigenA$vectors
D <- diag(EigenA$values)
A_Reconstructed <- P %*% D %*% solve(P)
all.equal(MatrixA, A_Reconstructed, tolerance = 1e-6, check.attributes = FALSE)
```

```
> all.equal(MatrixA, A_Reconstructed, tolerance = 1e-6, check.attributes = FALSE)
[1] TRUE
```

```
Matrix_5 <- A_Reconstructed %*% solve(A_Reconstructed)
all.equal(Matrix_5, diag(11), tolerance = 1e-6, check.attributes = FALSE)
```

```
> all.equal(Matrix_5, diag(11), tolerance = 1e-6, check.attributes = FALSE)
[1] TRUE
```

## Problem 6

```
MatrixB <- MatrixA %*% t(Standardized_X) %*% YMatrix
print(MatrixB)
```

```

              y
x_dfy -0.0051961288
x_infl -0.0012814334
x_svar -0.0005996872
x_tms -0.0045372084
x_tbl -0.0108741731
x_dfr 0.0044648300
x_dp 0.0177837102
x_ltr 0.0040115413
x_ep -0.0008651173
x_bmr 0.0060527912
x_ntis 0.0013990910
```

## GitHub Repo

Econometric-homework1-b10901069