Principal Components Analysis

Using Treasury Yields from 2020 for PCA.

```
library(readxl)
library(MASS)

# Load Data
setwd("C:/Users/uditg/Documents/R scripts/")
Rates_raw <- read_excel("PCA_TsyRates.xlsx",
                           col_types = c("date", "numeric", "numeric",
                                         "numeric", "numeric", "numeric",
                                         "numeric", "numeric", "numeric",
                                         "numeric", "numeric", "numeric",
                                         "numeric"))
# Droping the Date column
Rates_raw = Rates_raw[,2:ncol(Rates_raw)]
Rates = Rates_raw

m = apply(Rates, 2, mean)
print(m)
```

```
##        1m        2m        3m        6m        1y        2y        3y        5y
## 0.3527092 0.3605578 0.3602789 0.3709562 0.3701195 0.3886454 0.4209960 0.5330677
##        7y       10y       20y       30y
## 0.7218327 0.8892032 1.3482072 1.5561355
```

```
s = apply(Rates, 2, sd)
print(s)
```

```
##        1m        2m        3m        6m        1y        2y        3y        5y
## 0.5445084 0.5402545 0.5301739 0.5187407 0.4902901 0.4681570 0.4513688 0.4170081
##        7y       10y       20y       30y
## 0.3839977 0.3532242 0.3039098 0.2840447
```

- Scaling the Rates before running PCA

```
Rates = scale(Rates)
Rates[1:5,]
```

```
##            1m       2m       3m       6m       1y       2y       3y       5y
## [1,] 2.162117 2.201633 2.225159 2.311451 2.426891 2.544776 2.589909 2.726403
## [2,] 2.143752 2.201633 2.187435 2.272896 2.406495 2.437974 2.479134 2.534561
## [3,] 2.180482 2.183124 2.262882 2.292174 2.386098 2.459334 2.523444 2.582521
## [4,] 2.143752 2.164614 2.225159 2.292174 2.365702 2.459334 2.501289 2.606502
## [5,] 2.107021 2.164614 2.225159 2.292174 2.406495 2.544776 2.634218 2.726403
##            7y       10y       20y       30y
## [1,] 2.781702 2.805008 2.769877 2.724446
## [2,] 2.573368 2.578523 2.506641 2.478006
## [3,] 2.599410 2.606834 2.572450 2.548417
## [4,] 2.651493 2.663455 2.671164 2.654035
## [5,] 2.755661 2.776697 2.835686 2.794858
```

- Run PCA

```
# Run PCA
pca = prcomp(Rates, scale=FALSE)
Loading = pca$rotation
print(Loading)
```

```
##             PC1         PC2         PC3          PC4          PC5          PC6
## 1m  -0.2891222 -0.25978939  0.34828578  0.567768167 -0.224111631 -0.005264558
## 2m  -0.2902946 -0.25046358  0.31221090  0.275338944 -0.004520183  0.048099791
## 3m  -0.2914535 -0.22731113  0.28496457 -0.125994018  0.234787993 -0.097731829
## 6m  -0.2922516 -0.21318049  0.17958408 -0.415984263  0.218390267 -0.179708137
## 1y  -0.2933205 -0.18874576  0.01045044 -0.485098860  0.119566295  0.118272525
## 2y  -0.2940592 -0.14228752 -0.26374670 -0.151667436 -0.337381357  0.313523691
## 3y  -0.2938734 -0.11655838 -0.37063977 -0.044281401 -0.383831797  0.057542591
## 5y  -0.2947993 -0.01735932 -0.38921716  0.086023301 -0.059324699 -0.138645419
## 7y  -0.2935889  0.11033186 -0.40217113  0.269362095  0.186184970 -0.305100420
## 10y -0.2894402  0.28602430 -0.15138125  0.204093856  0.628512991  0.072306764
## 20y -0.2728428  0.52689546  0.20966872 -0.008565264 -0.033780650  0.655949630
## 30y -0.2675667  0.57558792  0.28392231 -0.175903395 -0.370311668 -0.540748112
##             PC7         PC8         PC9         PC10        PC11        PC12
## 1m   0.19386425 -0.31473262  0.04433356 -0.313870697  0.29711088 -0.17206180
## 2m  -0.08368685  0.13223101  0.10811646  0.392094410 -0.63360919  0.29325670
## 3m  -0.19478635  0.46713844 -0.25356661  0.320576188  0.49921149 -0.17085380
## 6m  -0.24453405 -0.05635004 -0.09969101 -0.662438853 -0.23928389  0.12193378
## 1y   0.30499426 -0.39586091  0.49517461  0.292928652  0.07692996 -0.16725089
## 2y   0.44126889  0.28489036 -0.19706841 -0.116215747  0.14356755  0.48965332
## 3y  -0.11744314  0.11239289 -0.19338117  0.005604189 -0.31422657 -0.66894603
## 5y  -0.48914687 -0.49804425 -0.19667306  0.234369931  0.20252390  0.33048723
## 7y  -0.04172234  0.37862805  0.59742195 -0.170088239  0.08446730  0.03880829
## 10y  0.39270377 -0.13520882 -0.40114706  0.016533281 -0.15444797 -0.10974013
## 20y -0.35314872  0.04059125  0.17010844 -0.105893255  0.07246063 -0.02702039
## 30y  0.19027937 -0.01509289 -0.07233104  0.107061323 -0.03566583  0.04016785
```

```
Scores = pca$x
print(Scores[1:5,])
```

```
##             PC1       PC2        PC3         PC4         PC5          PC6
## [1,] -8.726584 0.8557619 -0.3663589 -0.03269365 -0.01305345  0.015352007
## [2,] -8.308035 0.5443033 -0.2538381 -0.06457221 -0.01970636  0.018169833
## [3,] -8.420998 0.5991240 -0.2437692 -0.05767781 -0.04108339  0.005627315
## [4,] -8.475459 0.7626381 -0.2532596 -0.05977617 -0.03414411 -0.002552605
## [5,] -8.722269 0.9463735 -0.3686554 -0.08395894 -0.07513415  0.028453990
##               PC7         PC8          PC9         PC10          PC11
## [1,] -0.01270483 -0.03738700  0.013167100  0.007037492  0.0008092568
## [2,]  0.01983048 -0.04153733  0.035298227  0.020284716 -0.0281026353
## [3,] -0.01617989 -0.01849348 -0.008134559  0.012555030  0.0249614761
## [4,] -0.02445367 -0.01888472  0.004824729 -0.002813962  0.0162338713
## [5,] -0.04687208 -0.01522514 -0.008123899  0.021390011  0.0014462745
##               PC12
## [1,]  0.0187070354
## [2,] -0.0005900514
## [3,] -0.0238439155
## [4,]  0.0070326968
## [5,] -0.0081031836
```

```
# Reconciling: Rates * Loading = Scores
PCA_scores = Rates %*% Loading
sum(round(PCA_scores - Scores,4))
```

```
## [1] 0
```

- Reconciling PCA output with EIGEN Vectors/ Values

```
# Using Eigen function - Eigen vector same as loading
eigen.vector = eigen(t(Rates)%*%Rates)$vectors
print(eigen.vector)
```

```
##              [,1]        [,2]        [,3]         [,4]         [,5]         [,6]
##  [1,] -0.2891222 -0.25978939  0.34828578  0.567768167 -0.224111631  0.005264558
##  [2,] -0.2902946 -0.25046358  0.31221090  0.275338944 -0.004520183 -0.048099791
##  [3,] -0.2914535 -0.22731113  0.28496457 -0.125994018  0.234787993  0.097731829
##  [4,] -0.2922516 -0.21318049  0.17958408 -0.415984263  0.218390267  0.179708137
##  [5,] -0.2933205 -0.18874576  0.01045044 -0.485098860  0.119566295 -0.118272525
##  [6,] -0.2940592 -0.14228752 -0.26374670 -0.151667436 -0.337381357 -0.313523691
##  [7,] -0.2938734 -0.11655838 -0.37063977 -0.044281401 -0.383831797 -0.057542591
##  [8,] -0.2947993 -0.01735932 -0.38921716  0.086023301 -0.059324699  0.138645419
##  [9,] -0.2935889  0.11033186 -0.40217113  0.269362095  0.186184970  0.305100420
## [10,] -0.2894402  0.28602430 -0.15138125  0.204093856  0.628512991 -0.072306764
## [11,] -0.2728428  0.52689546  0.20966872 -0.008565264 -0.033780650 -0.655949630
## [12,] -0.2675667  0.57558792  0.28392231 -0.175903395 -0.370311668  0.540748112
##               [,7]        [,8]        [,9]        [,10]       [,11]       [,12]
##  [1,]  0.19386425 -0.31473262  0.04433356  0.313870697 -0.29711088  0.17206180
##  [2,] -0.08368685  0.13223101  0.10811646 -0.392094411  0.63360919 -0.29325670
##  [3,] -0.19478635  0.46713844 -0.25356661 -0.320576188 -0.49921149  0.17085380
##  [4,] -0.24453405 -0.05635004 -0.09969101  0.662438853  0.23928389 -0.12193378
##  [5,]  0.30499426 -0.39586091  0.49517461 -0.292928652 -0.07692996  0.16725089
##  [6,]  0.44126889  0.28489036 -0.19706841  0.116215747 -0.14356755 -0.48965332
##  [7,] -0.11744314  0.11239289 -0.19338117 -0.005604189  0.31422657  0.66894603
##  [8,] -0.48914687 -0.49804425 -0.19667306 -0.234369931 -0.20252390 -0.33048723
##  [9,] -0.04172234  0.37862805  0.59742195  0.170088239 -0.08446730 -0.03880829
## [10,]  0.39270377 -0.13520882 -0.40114706 -0.016533281  0.15444797  0.10974013
## [11,] -0.35314872  0.04059125  0.17010844  0.105893255 -0.07246063  0.02702039
## [12,]  0.19027937 -0.01509289 -0.07233104 -0.107061323  0.03566583 -0.04016785
```

```
round(Loading - eigen.vector,4)
```

```
##     PC1 PC2 PC3 PC4 PC5     PC6 PC7 PC8 PC9    PC10    PC11    PC12
## 1m    0   0   0   0   0 -0.0105   0   0   0 -0.6277  0.5942 -0.3441
## 2m    0   0   0   0   0  0.0962   0   0   0  0.7842 -1.2672  0.5865
## 3m    0   0   0   0   0 -0.1955   0   0   0  0.6412  0.9984 -0.3417
## 6m    0   0   0   0   0 -0.3594   0   0   0 -1.3249 -0.4786  0.2439
## 1y    0   0   0   0   0  0.2365   0   0   0  0.5859  0.1539 -0.3345
## 2y    0   0   0   0   0  0.6270   0   0   0 -0.2324  0.2871  0.9793
## 3y    0   0   0   0   0  0.1151   0   0   0  0.0112 -0.6285 -1.3379
## 5y    0   0   0   0   0 -0.2773   0   0   0  0.4687  0.4050  0.6610
## 7y    0   0   0   0   0 -0.6102   0   0   0 -0.3402  0.1689  0.0776
## 10y   0   0   0   0   0  0.1446   0   0   0  0.0331 -0.3089 -0.2195
## 20y   0   0   0   0   0  1.3119   0   0   0 -0.2118  0.1449 -0.0540
## 30y   0   0   0   0   0 -1.0815   0   0   0  0.2141 -0.0713  0.0803
```

```
eigen.vector[,6] = -eigen.vector[,6]      # some vectors have opposite sign
eigen.vector[,10] = -eigen.vector[,10]
eigen.vector[,11] = -eigen.vector[,11]
eigen.vector[,12] = -eigen.vector[,12]

sum(round(Loading - eigen.vector,4))
```

```
## [1] 0
```

```
# Using Eigen function - Eigen values explain the %age of variance captured
eigen.values = eigen(t(Rates)%*%Rates)$values

round(eigen.values/sum(eigen.values)*100,2)
```

```
##  [1] 94.67  4.50  0.66  0.10  0.03  0.02  0.01  0.01  0.00  0.00  0.00  0.00
```

```
round(pca$sdev^2/sum(pca$sdev^2)*100,2)
```

```
##  [1] 94.67  4.50  0.66  0.10  0.03  0.02  0.01  0.01  0.00  0.00  0.00  0.00
```

- Reconstructing Data from all Principal Components

```
# Reconstructing Data from PCA scores
round(ginv(Loading) %*% Loading,4)
```

```
##       PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12
##  [1,]   1   0   0   0   0   0   0   0   0    0    0    0
##  [2,]   0   1   0   0   0   0   0   0   0    0    0    0
##  [3,]   0   0   1   0   0   0   0   0   0    0    0    0
##  [4,]   0   0   0   1   0   0   0   0   0    0    0    0
##  [5,]   0   0   0   0   1   0   0   0   0    0    0    0
##  [6,]   0   0   0   0   0   1   0   0   0    0    0    0
##  [7,]   0   0   0   0   0   0   1   0   0    0    0    0
##  [8,]   0   0   0   0   0   0   0   1   0    0    0    0
##  [9,]    0    0    0    0    0    0    0    0    1     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     1     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     1     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     1
```

```
Loading_inv = ginv(Loading)

# Reconciling reconstructed data
round((Scores %*% Loading_inv)[1:5,] - Rates[1:5,],4)
```

```
##      1m 2m 3m 6m 1y 2y 3y 5y 7y 10y 20y 30y
## [1,]  0  0  0  0  0  0  0  0  0   0   0   0
## [2,]  0  0  0  0  0  0  0  0  0   0   0   0
## [3,]  0  0  0  0  0  0  0  0  0   0   0   0
## [4,]  0  0  0  0  0  0  0  0  0   0   0   0
## [5,]  0  0  0  0  0  0  0  0  0   0   0   0
```

- Reconstructing Data from only 3 Principal Components

```
# Reconstructing Data using only 3 PCs
Scores_3 = Scores[,1:3]
Loading_inv_3 = Loading_inv[1:3,]

# Reconstructed Data
data = round((Scores_3 %*% Loading_inv_3)[1:5,],4)

# Descaling Data
data_reconstruct = matrix(data = NA, nrow = nrow(data), ncol = ncol(data))

for(i in 1:nrow(data)){
data_reconstruct[i,] = data[i,]*s+m
}

# Error between Reconstructed Data and Actual Data
Rates_raw[1:5,]
```

```
## # A tibble: 5 x 12
##    ‘1m‘  ‘2m‘  ‘3m‘  ‘6m‘  ‘1y‘  ‘2y‘  ‘3y‘  ‘5y‘  ‘7y‘ ‘10y‘ ‘20y‘ ‘30y‘
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1.53  1.55  1.54  1.57  1.56  1.58  1.59  1.67  1.79  1.88  2.19  2.33
## 2  1.52  1.55  1.52  1.55  1.55  1.53  1.54  1.59  1.71  1.8   2.11  2.26
## 3  1.54  1.54  1.56  1.56  1.54  1.54  1.56  1.61  1.72  1.81  2.13  2.28
## 4  1.52  1.53  1.54  1.56  1.53  1.54  1.55  1.62  1.74  1.83  2.16  2.31
## 5  1.5   1.53  1.54  1.56  1.55  1.58  1.61  1.67  1.78  1.87  2.21  2.35
```

```
round(data_reconstruct,2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 1.54 1.55 1.55 1.57 1.54 1.58 1.59 1.66 1.80  1.89  2.19  2.33
## [2,] 1.54 1.55 1.54 1.55 1.51 1.53 1.54 1.59 1.72  1.81  2.11  2.26
## [3,] 1.55 1.56 1.55 1.56 1.52 1.54 1.55 1.60 1.73  1.82  2.13  2.27
## [4,] 1.53 1.54 1.54 1.55 1.52 1.54 1.55 1.61 1.75  1.85  2.16  2.30
## [5,] 1.52 1.54 1.54 1.55 1.53 1.57 1.59 1.66 1.80  1.90  2.20  2.34
```

```
round(data_reconstruct - Rates_raw[1:5,],4)
```

```
##      1m      2m      3m      6m      1y      2y      3y      5y     7y    10y
## 1 0.0060  0.0016  0.0103 -0.0049 -0.0160 -0.0018  0.0048 -0.0109 0.0085 0.0075
## 2 0.0155 -0.0029  0.0201 -0.0034 -0.0368 -0.0025 -0.0031  0.0017 0.0107 0.0072
## 3 0.0075  0.0191 -0.0075 -0.0013 -0.0155 -0.0019 -0.0127 -0.0065 0.0142 0.0137
## 4 0.0111  0.0139 -0.0002 -0.0121 -0.0129 -0.0041 -0.0025 -0.0094 0.0088 0.0163
## 5 0.0220  0.0083 -0.0017 -0.0057 -0.0150 -0.0081 -0.0201 -0.0117 0.0222 0.0263
##       20y     30y
## 1 -0.0045 -0.0003
## 2 -0.0019 -0.0039
## 3 -0.0031 -0.0056
## 4 -0.0030 -0.0055
## 5 -0.0105 -0.0060
```