Assignment_9

Hemal Agarwal

December 5, 2016

1 Data

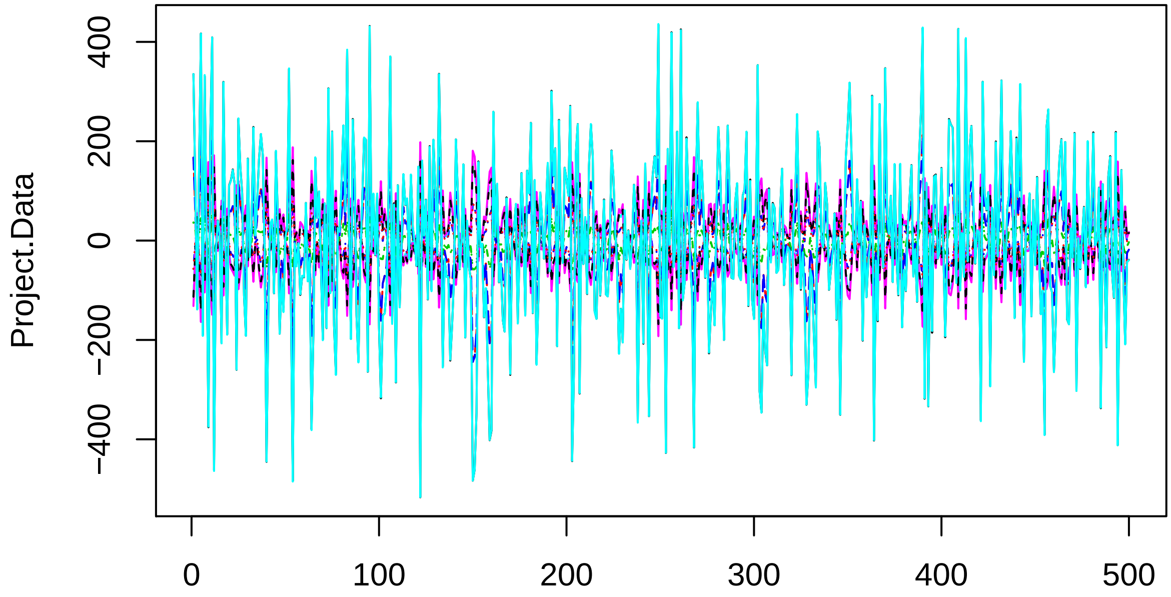
```
#setwd("C:/Users/Adroit/Desktop/CourseProject")
Project.Data <- read.csv("PCA_ExampleData.csv",header=TRUE,sep=",")
Project.Data[1:10,]
```

```
##            V1         V2        V3         V4         V5         V6
## 1   -10.00000   -8.00000  -6.00000  -4.000000   -2.00000    0.00000
## 2   335.26043  167.30996  36.42212 -57.023949 -112.88559 -131.66066
## 3   106.04431   61.94413  22.68583  -4.491896  -16.86543  -23.94030
## 4  -138.25926  -69.13931 -14.48162  24.403158   47.02190   55.09567
## 5   139.51863   66.76133  11.15244 -28.738979  -53.45132  -61.10555
## 6   416.99904  207.59255  45.27888 -70.756695 -140.82071 -163.84332
## 7  -190.35076 -104.67105 -34.61287  14.833058   41.78893   52.80835
## 8   331.03604  177.06454  52.22067 -36.066936  -85.00319 -104.34305
## 9    18.52366   16.95822  13.74827  11.802604   12.21568   11.16780
## 10 -375.72268 -182.05494 -34.09990  72.048443  137.85975  158.20484
##            V7         V8        V9        V10        V11
## 1     2.00000   4.000000   6.00000    8.00000   10.00000
## 2  -113.07746 -56.864257  36.48108  167.10119  335.37523
## 3   -20.52873  -1.442949  23.81151   57.95828  108.23602
## 4    47.68518  23.851115 -14.68544  -68.41763 -138.65609
## 5   -52.72716 -29.341695  10.92991   67.54925  139.08538
## 6  -140.40840 -71.099858  45.15218  208.04116  416.75236
## 7    44.31460  12.730956 -35.38898 -101.92300 -191.86184
## 8   -88.76262 -32.937981  53.37590  172.97410  333.28527
## 9    10.74362  13.027791  14.20061   15.35655   19.40438
## 10  135.88303  73.693656 -33.49248 -184.20571 -374.54004
```

```
Data.Levels<-as.numeric(Project.Data[1,])
Project.Data<-Project.Data[-1,]
head(Project.Data)
```

```
##          V1         V2        V3         V4         V5         V6
## 2  335.2604  167.30996  36.42212 -57.023949 -112.88559 -131.66066
## 3  106.0443   61.94413  22.68583  -4.491896  -16.86543  -23.94030
## 4 -138.2593  -69.13931 -14.48162  24.403158   47.02190   55.09567
## 5  139.5186   66.76133  11.15244 -28.738979  -53.45132  -61.10555
## 6  416.9990  207.59255  45.27888 -70.756695 -140.82071 -163.84332
## 7 -190.3508 -104.67105 -34.61287  14.833058   41.78893   52.80835
##           V7         V8        V9       V10       V11
## 2 -113.07746 -56.864257  36.48108 167.10119  335.3752
## 3  -20.52873  -1.442949  23.81151  57.95828  108.2360
## 4   47.68518  23.851115 -14.68544 -68.41763 -138.6561
## 5  -52.72716 -29.341695  10.92991   67.54925  139.0854
## 6 -140.40840 -71.099858  45.15218  208.04116  416.7524
## 7   44.31460  12.730956 -35.38898 -101.92300 -191.8618
```

```
matplot(Project.Data,type="l")
```



# 2 PCA

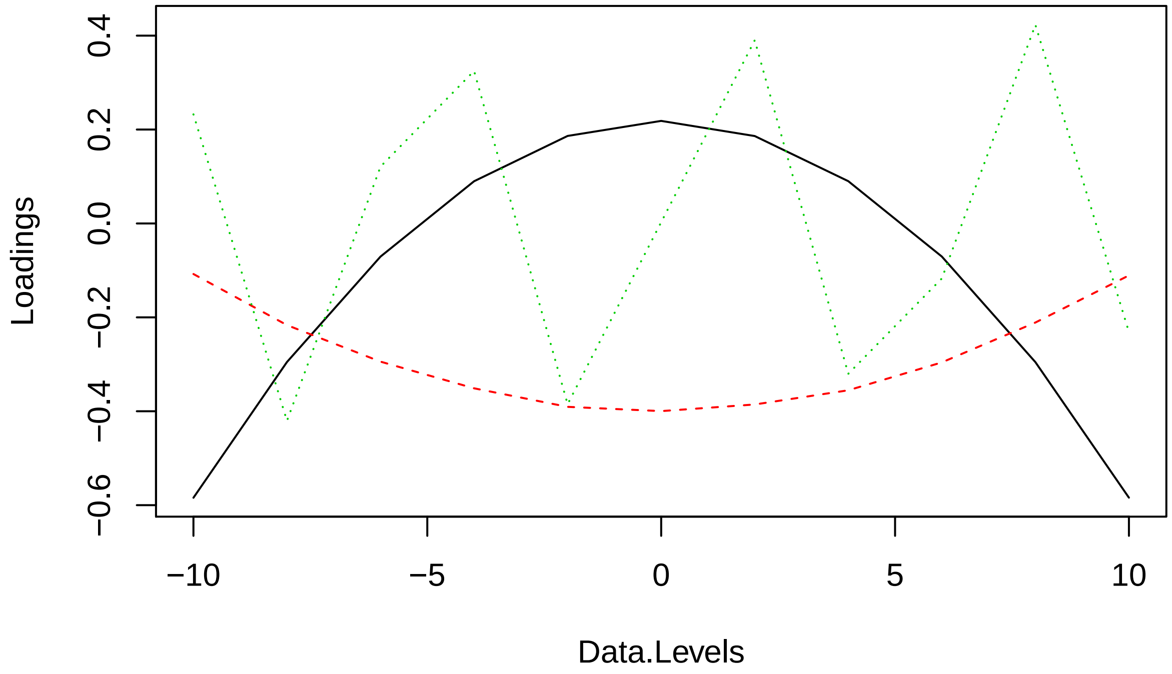
```
Project.Data.PCA <- princomp(Project.Data)
names(Project.Data.PCA)
```

```
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"
## [7] "call"
```

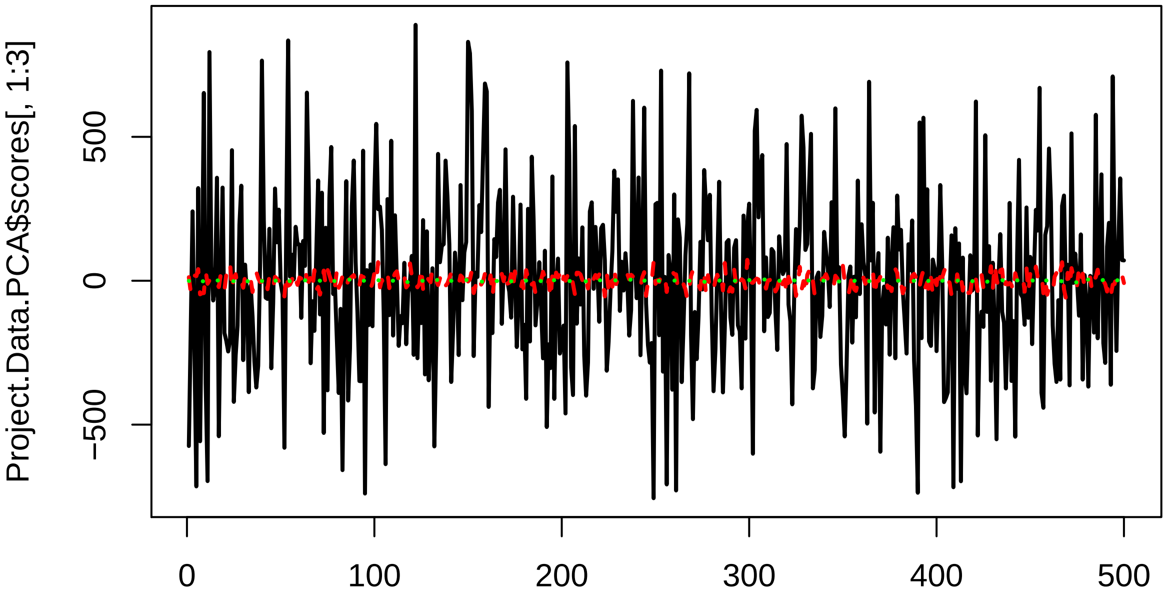
```
Project.Data.PCA$loadings
```

```
##
## Loadings:
##     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## V1  -0.584 -0.108  0.232  0.725                                    -0.261
## V2  -0.295 -0.217 -0.420        -0.450 -0.103        -0.198  0.402  0.229
## V3         -0.294  0.121         0.293  0.635  0.329 -0.397 -0.147  0.133
## V4         -0.351  0.325 -0.102  0.188 -0.562  0.545         0.286
## V5   0.186 -0.391 -0.385        -0.168                0.160 -0.477 -0.576
## V6   0.218 -0.400         0.157 -0.470  0.207  0.213 -0.120  0.152  0.116
## V7   0.186 -0.386  0.390  0.134 -0.126 -0.167 -0.429        -0.392  0.462
## V8         -0.355 -0.320         0.565 -0.176 -0.423 -0.360  0.178
## V9         -0.296 -0.117         0.238  0.330         0.785  0.305  0.120
## V10 -0.295 -0.211  0.423 -0.558 -0.164        -0.359         0.129 -0.425
## V11 -0.584 -0.111 -0.231 -0.298        -0.218  0.189        -0.436  0.319
##     Comp.11
## V1
## V2   0.460
## V3   0.302
## V4   0.139
## V5   0.202
## V6  -0.639
## V7   0.210
## V8  -0.241
## V9
## V10 -0.106
## V11 -0.340
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
## Cumulative Var  0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727
##                Comp.9 Comp.10 Comp.11
## SS loadings     1.000   1.000   1.000
## Proportion Var  0.091   0.091   0.091
## Cumulative Var  0.818   0.909   1.000
```

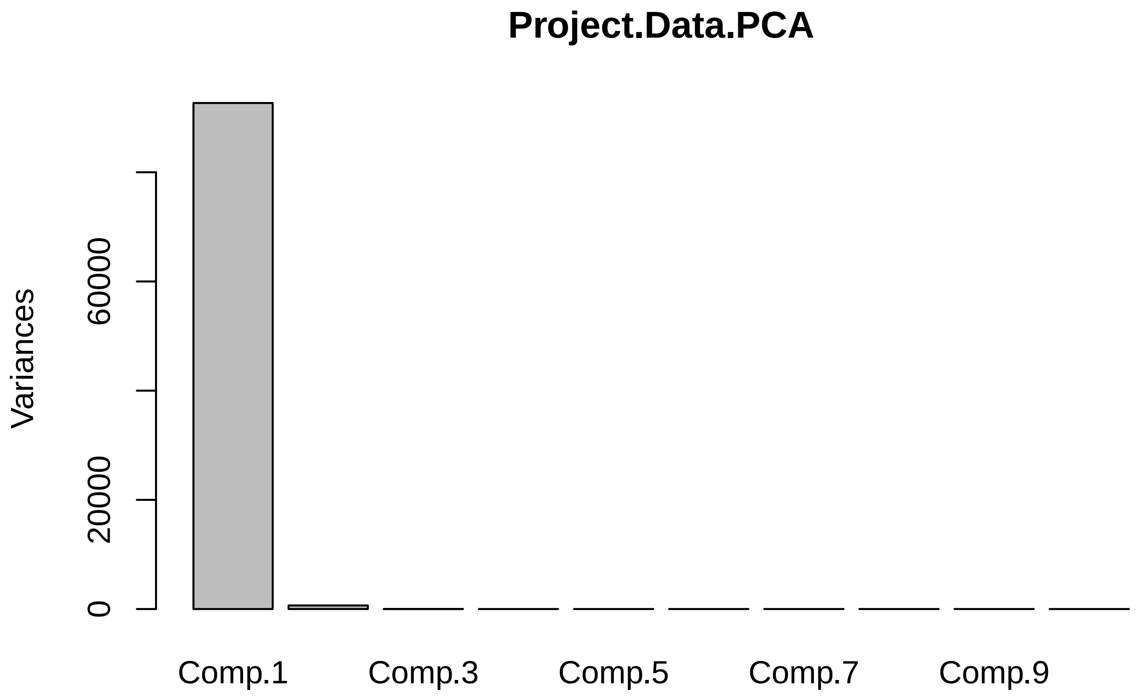
```r
matplot(Data.Levels,Project.Data.PCA$loadings[,1:3],type = "l",ylab = "Loadings")
```



```
matplot(Project.Data.PCA$scores[,1:3],type="l",lty=c(1,2,3),lwd=2,col=c("black","red","green"))
```



```
plot(Project.Data.PCA)
```



```
(Project.Data.PCA$sdev)^2
```

```
##       Comp.1       Comp.2       Comp.3       Comp.4       Comp.5
## 9.268609e+04 6.489602e+02 5.576366e+00 3.053585e-11 3.062401e-12
##       Comp.6       Comp.7       Comp.8       Comp.9      Comp.10
## 1.785265e-12 1.371684e-12 0.000000e+00 0.000000e+00 0.000000e+00
##      Comp.11
## 0.000000e+00
```

Estimate PCA using manual calculation with eigen(). For this recall the steps on slide 16 of the lecture notes.

```
setwd("C:/Users/Adroit/Desktop/CourseProject")
Project.Data <- read.csv("PCA_ExampleData.csv",header=TRUE,sep=",")
Project.Data<-Project.Data[-1,]
mean_vector <- as.vector(apply(as.matrix(Project.Data),2, mean))
mean_matrix <- matrix(mean_vector, length(Project.Data$V1),length(mean_vector), byrow= TRUE)
centered_matrix <- as.matrix(Project.Data) - as.matrix(mean_matrix)
head(centered_matrix)
```

```
##          V1        V2        V3         V4         V5         V6
## 2  333.9805 166.99331  36.83981 -56.079121 -111.61265 -130.28723
## 3  104.7643  61.62748  23.10352  -3.547069  -15.59249  -22.56687
## 4 -139.5392 -69.45596 -14.06393  25.347986   48.29483   56.46910
## 5  138.2387   66.44468  11.57013 -27.794151  -52.17839  -59.73211
## 6  415.7191  207.27590  45.69657 -69.811867 -139.54778 -162.46989
## 7 -191.6307 -104.98770 -34.19517  15.777886   43.06186   54.18178
##           V7         V8        V9        V10       V11
## 2 -111.81554 -55.910258  36.90216  166.77256  334.1018
## 3  -19.26681  -0.488951  24.23260   57.62964  106.9626
## 4   48.94710  24.805113 -14.26436  -68.74627 -139.9295
## 5  -51.46524 -28.387697  11.35099   67.22061  137.8120
## 6 -139.14649 -70.145860  45.57326  207.71253  415.4790
## 7   45.57652  13.684954 -34.96789 -102.25163 -193.1352
```

```
covariance_matrix <-cov(centered_matrix)
eigen_values<-eigen(covariance_matrix)
eigen_values$vectors
```

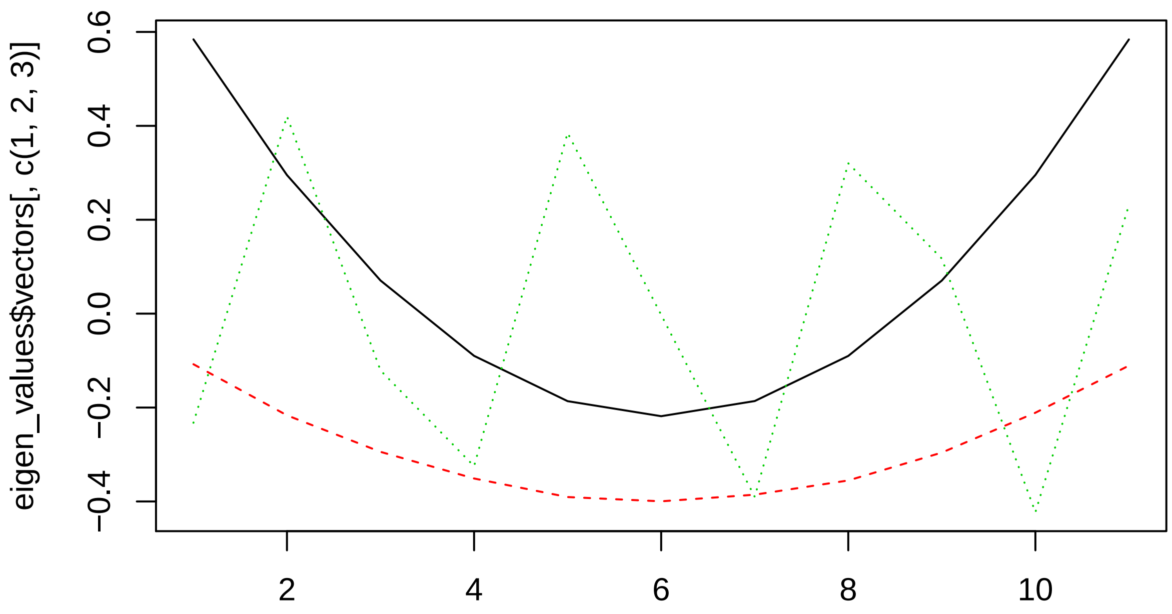
```
##              [,1]       [,2]         [,3]         [,4]        [,5]
##  [1,]  0.58420418 -0.1077103 -0.232409041  0.704088457  0.00000000
##  [2,]  0.29510562 -0.2165408  0.420358898 -0.445851077 -0.04958303
##  [3,]  0.07055080 -0.2943480 -0.120991690 -0.101718860  0.39148860
##  [4,] -0.08989770 -0.3511162 -0.324880404 -0.119855757 -0.22566137
##  [5,] -0.18640449 -0.3906019  0.384963011  0.263811539  0.45870310
##  [6,] -0.21839512 -0.3996945 -0.002667934  0.100673838 -0.40721865
##  [7,] -0.18618311 -0.3855492 -0.390129058 -0.105466307  0.15420914
##  [8,] -0.09008195 -0.3553215  0.320223821  0.221787141 -0.50799816
##  [9,]  0.07048278 -0.2959006  0.117184214  0.001162004  0.34780785
## [10,]  0.29534648 -0.2110431 -0.422977624 -0.315124579 -0.05611926
## [11,]  0.58407174 -0.1107333  0.231318707 -0.203506399 -0.10562822
##              [,6]        [,7]        [,8]         [,9]       [,10]
##  [1,]  0.00000000  0.00000000  0.00000000  0.000000000  0.00000000
##  [2,] -0.02976470  0.02421440  0.02556953  0.009529595  0.08423477
##  [3,] -0.03071383  0.09868706 -0.76589307  0.266898146  0.22963443
##  [4,]  0.70711677  0.38064516  0.01132780 -0.042577037 -0.21269880
##  [5,] -0.15010781  0.49940024  0.29068194 -0.089986835 -0.10509697
##  [6,] -0.36508083 -0.14412998 -0.04445893  0.521624807 -0.43209635
##  [7,] -0.27458962 -0.33382232 -0.02329083 -0.633460487 -0.11996743
##  [8,]  0.06242348 -0.08920801 -0.11772099 -0.253343219  0.56506975
##  [9,]  0.41412266 -0.63922546  0.30828191  0.269378476  0.05220206
## [10,] -0.30080502  0.21554266  0.44666628  0.207805404  0.39296306
## [11,] -0.03260110 -0.01210376 -0.13116364 -0.255868849 -0.45424453
##             [,11]
##  [1,] -0.31200873
##  [2,] -0.69193000
##  [3,]  0.09429502
##  [4,] -0.07558668
##  [5,]  0.09439269
##  [6,] -0.04177062
##  [7,] -0.16291027
##  [8,]  0.21595673
##  [9,]  0.14945530
## [10,]  0.22980901
## [11,]  0.50029755
```

```r
Project.Data.PCA$loadings
```

```
##
## Loadings:
##     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
## V1  -0.584 -0.108  0.232  0.725                                    -0.261
## V2  -0.295 -0.217 -0.420        -0.450 -0.103        -0.198  0.402  0.229
## V3         -0.294  0.121         0.293  0.635  0.329 -0.397 -0.147  0.133
## V4         -0.351  0.325 -0.102  0.188 -0.562  0.545         0.286
## V5   0.186 -0.391 -0.385        -0.168                0.160 -0.477 -0.576
## V6   0.218 -0.400         0.157 -0.470  0.207  0.213 -0.120  0.152  0.116
## V7   0.186 -0.386  0.390  0.134 -0.126 -0.167 -0.429        -0.392  0.462
## V8         -0.355 -0.320         0.565 -0.176 -0.423 -0.360  0.178
## V9         -0.296 -0.117         0.238  0.330         0.785  0.305  0.120
## V10 -0.295 -0.211  0.423 -0.558 -0.164        -0.359         0.129 -0.425
## V11 -0.584 -0.111 -0.231 -0.298        -0.218  0.189        -0.436  0.319
##     Comp.11
## V1
## V2   0.460
## V3   0.302
## V4   0.139
## V5   0.202
## V6  -0.639
## V7   0.210
## V8  -0.241
## V9
## V10 -0.106
## V11 -0.340
##
##                 Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings      1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var   0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
## Cumulative Var   0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727
##                 Comp.9 Comp.10 Comp.11
## SS loadings      1.000   1.000   1.000
## Proportion Var   0.091   0.091   0.091
## Cumulative Var   0.818   0.909   1.000
```

Calculate 3 factor loadings using PCA and using manual method based on eigen-decomposition. Combine them in one matrix Project.Data.PCA.by.eigen.Loadings and compare

```r
matplot(eigen_values$vectors[,c(1,2,3)],type = "l")
```



```
Project.Data.PCA.Eigen.Loadings= cbind(L1.eigen = eigen_values$vectors[,1],L2.eigen = eigen_values$vect
Project.Data.PCA.Eigen.Loadings
```

```
##        L1.eigen   L2.eigen     L3.eigen      L1.PCA     L2.PCA
## V1   0.58420418 -0.1077103 -0.232409041 -0.58420418 -0.1077103
## V2   0.29510562 -0.2165408  0.420358898 -0.29510562 -0.2165408
## V3   0.07055080 -0.2943480 -0.120991690 -0.07055080 -0.2943480
## V4  -0.08989770 -0.3511162 -0.324880404  0.08989770 -0.3511162
## V5  -0.18640449 -0.3906019  0.384963011  0.18640449 -0.3906019
## V6  -0.21839512 -0.3996945 -0.002667934  0.21839512 -0.3996945
## V7  -0.18618311 -0.3855492 -0.390129058  0.18618311 -0.3855492
## V8  -0.09008195 -0.3553215  0.320223821  0.09008195 -0.3553215
## V9   0.07048278 -0.2959006  0.117184214 -0.07048278 -0.2959006
## V10  0.29534648 -0.2110431 -0.422977624 -0.29534648 -0.2110431
## V11  0.58407174 -0.1107333  0.231318707 -0.58407174 -0.1107333
##           L3.PCA
## V1   0.232409041
## V2  -0.420358898
## V3   0.120991690
## V4   0.324880404
## V5  -0.384963011
## V6   0.002667934
## V7   0.390129058
## V8  -0.320223821
## V9  -0.117184214
## V10  0.422977624
## V11 -0.231318707
```