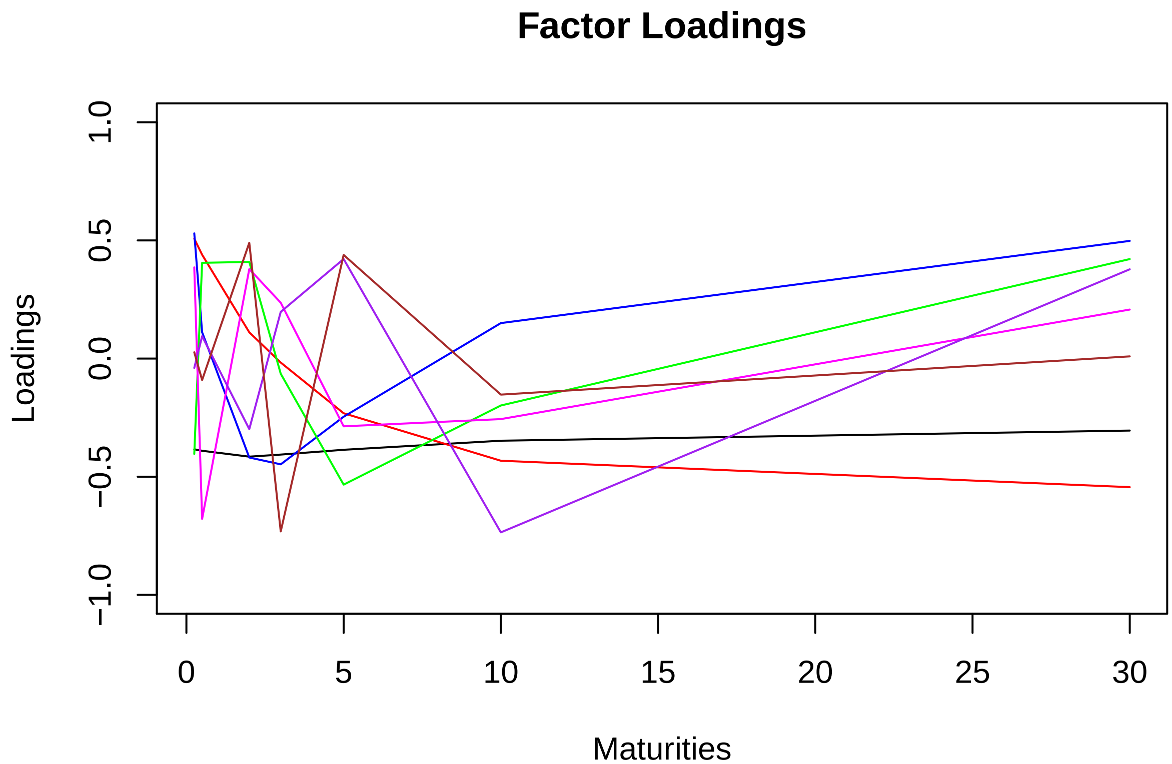
```r
#============================================#
# Michael Hyeong-Seok Beven                  #
# University of Chicago                      #
# Financial Mathematics                      #
# Homework Assignment on Statistical Models #
# 20160510                                   #
#============================================#

#########
# setup #
#########

rm(list=ls())
library(TTR) # rolling calculations
library(PerformanceAnalytics) # time series data

####################
# prepare the data #
####################

df <- as.data.frame(read.csv('assignments-Fixed Income Derivatives-Assignment Lecture 4- Statistical Mod
rownames(df) <- as.Date(df$Date,format='%m/%d/%Y')
df <- df[colnames(df)[2:8]]
maturities <- c(0.25,0.5,2,3,5,10,30) #maturities of instruments

##########################################
# estimate the 3-factor model using PCA #
##########################################

# define factor and factor loadings
df.cov <- cov(df) # covariance matrix
eigenvectors <- eigen(df.cov)$vectors
eigenvalues <- eigen(df.cov)$values
PC <- as.matrix(df) %*% eigenvectors # principal components

# calculate relative importance of factors
print(round(eigenvalues/sum(eigenvalues)*100,2))
```

```
## [1] 97.83  1.98  0.16  0.02  0.01  0.00  0.00
```

```r
print(round(cumsum(eigenvalues)/sum(eigenvalues)*100,2))
```

```
## [1]  97.83  99.81  99.97  99.98 100.00 100.00 100.00
```

```r
# plot and interpret the shapes of factor loadings
plot(maturities,eigenvectors[,1],type='l',ylim=c(-1,1),main='Factor Loadings',
     xlab='Maturities',ylab='Loadings')
lines(maturities,eigenvectors[,2],col='red')
lines(maturities,eigenvectors[,3],col='blue')
lines(maturities,eigenvectors[,4],col='green')
lines(maturities,eigenvectors[,5],col='magenta')
lines(maturities,eigenvectors[,6],col='purple')
lines(maturities,eigenvectors[,7],col='brown')
```



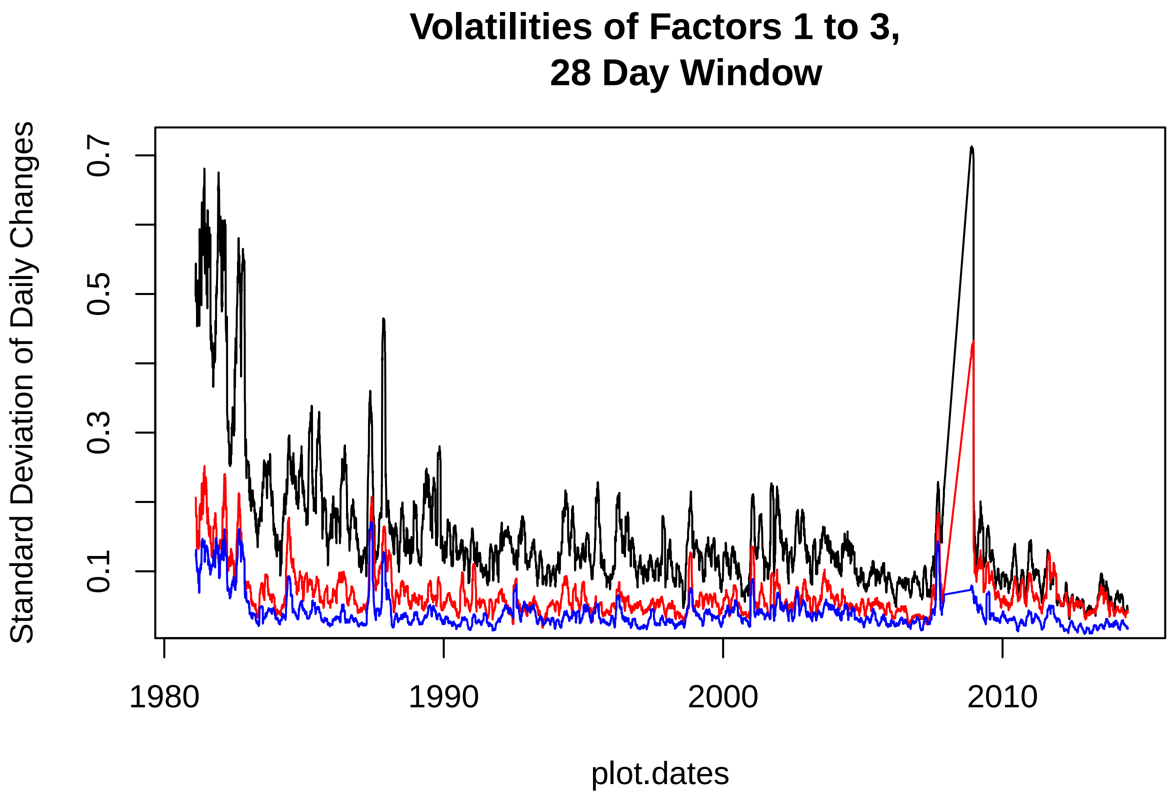

```
############################################################################
# calculate historical volatilities and correlation coefficients of factors #
############################################################################

# use the whole period of history to calculate var and cor
delta.fi <- diff(PC[,1:3])
print(diag(var(delta.fi)))
```

```
## [1] 0.034929992 0.006198080 0.002231987
```

```
print(cor(delta.fi))
```

```
##            [,1]       [,2]       [,3]
## [1,] 1.00000000 0.01394172 0.05079403
## [2,] 0.01394172 1.00000000 0.46744527
## [3,] 0.05079403 0.46744527 1.00000000
```

```
# calculate the same variables using a rolling window (approx 1 month)
plot.dates <- as.Date(rownames(runSD(delta.fi[,1])))
plot(plot.dates,runSD(delta.fi[,1],n=28),type='l',
     main='Volatilities of Factors 1 to 3,
     28 Day Window',ylab='Standard Deviation of Daily Changes')
lines(plot.dates,runSD(delta.fi[,2],n=28),col='red')
lines(plot.dates,runSD(delta.fi[,3],n=28),col='blue')
```




```
#################################################################
# find historical estimates of volatilities of the first 3 factors #
# corresponding to the last month of the observed period           #
#################################################################

PC.sub <- subset(PC[,1:3],rownames(PC)>='2014-06-01')
print(diag(var(PC.sub)))
```

```
## [1] 0.003411459 0.001527943 0.002283805
```

```
########################################################################
# calculate time series of each of the seven rates predicted by the model #
########################################################################

pred3M <- PC[,1:3] %*% eigenvectors[1,][1:3]
pred6M <- PC[,1:3] %*% eigenvectors[2,][1:3]
pred2Y <- PC[,1:3] %*% eigenvectors[3,][1:3]
pred3Y <- PC[,1:3] %*% eigenvectors[4,][1:3]
pred5Y <- PC[,1:3] %*% eigenvectors[5,][1:3]
pred10Y <- PC[,1:3] %*% eigenvectors[6,][1:3]
pred30Y <- PC[,1:3] %*% eigenvectors[7,][1:3]

################################################################
# fit parametric forms from slide 32 to each of the first three #
# vectors of factor loadings                                    #
################################################################

Loading.1 <- matrix(c(0.320,0.006,36.550,0.070,0.285,-0.292),nrow=3,ncol=2)
rownames(Loading.1) <- c(1,2,3)
colnames(Loading.1) <- c('a','b')
```

```r
L.bound.1 <- c(0,0,0,-Inf,-Inf,-Inf)

Loading.2 <- matrix(c(0.650,0.004,-1.130,0.539),nrow=2,ncol=2)
rownames(Loading.2) <- c(1,2)
colnames(Loading.2) <- c('a','b')
L.bound.2 <- c(0,0,-Inf,-Inf)

Loading.3 <- matrix(c(4.200e-01,5e-08,5e-01,2.876,-1.92,0.62,-0.41,3.035),nrow=4,ncol=2)
rownames(Loading.3) <- c(1,2,3,4)
colnames(Loading.3) <- c('a','b')
L.bound.3 <- c(0,0,0,0,-Inf,-Inf,-Inf)

fn <- function(mat) {
  mat <- matrix(mat,ncol=2)
  return(abs(pca.loading - sum(mat[,2] * (1-exp(-mat[,1]*tau))/(mat[,1]*tau))))}

tau <- 0.25
pca.loading <- eigenvectors[1,2]
optim(Loading.2,fn,method='L-BFGS-B',lower=L.bound.2)$par
```

```
##             a          b
## 1 0.719227281 -0.6280446
## 2 0.001979351  1.0825315
```

```r
##############################################################################
# calculate time series of instataneous forward rates with maturity 5 years #
# and discount bonds with maturity 4.5 year for whole period of observation #
##############################################################################
```