FDA_HW1

Wenjuan Bian

2023-07-01

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
suppressWarnings(library(fda))
```

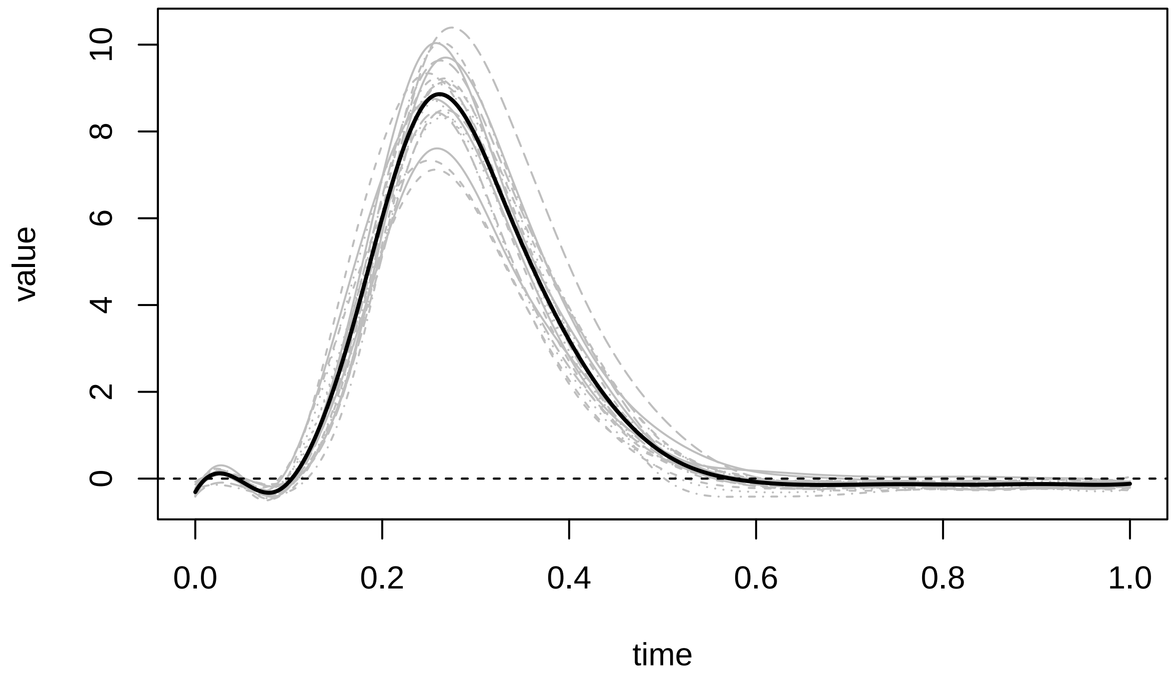
```
## Loading required package: splines

## Loading required package: fds

## Loading required package: rainbow

## Loading required package: MASS

## Loading required package: pcaPP

## Loading required package: RCurl

## Loading required package: deSolve

##
## Attaching package: 'fda'

## The following object is masked from 'package:graphics':
##
##     matplot
```

```
  library("grDevices")
#####
## 1.1 The pinch is a dataset included in the fda package. It consists of 151 measurements of pinch for
## (a) Convert the pinch data to functional objects using 15 B-splines of order four (cubic splines) an
## one graph.
```

```r
my_basis1 <- create.bspline.basis(c(0,1), nbasis = 15, norder = 4)

# Define argvals
pts1 <- seq(0,1, length.out = nrow(pinch))  # note the change from ncol to nrow

# Convert data to functional data
# Note: transpose is not required here because each column represents a distinct curve.
Pinch.F <- Data2fd(pts1, y = pinch, my_basis1)
my_basis<-create.bspline.basis(c(0,1),nbasis=20)


# Plot the functional data
plot(Pinch.F, col="gray")
```

```
## [1] "done"
```

```r
## (b) Calculate the pointwise mean and SD and add them to the plot.
Pinch.F.mean<-mean.fd(Pinch.F)
plot(Pinch.F.mean,add=TRUE,lwd=2)
```



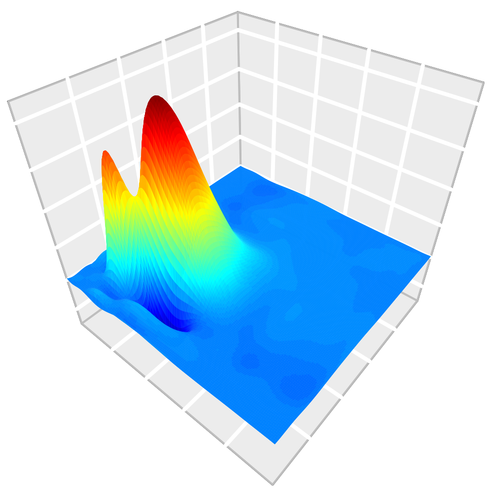

```
## [1] "done"
```

```r
## (c) Graph the perspective and contour plots of the sample covariance function ^c(t, s) of the pinch 
if (!require(plot3D)) {
  install.packages("plot3D")
}
```

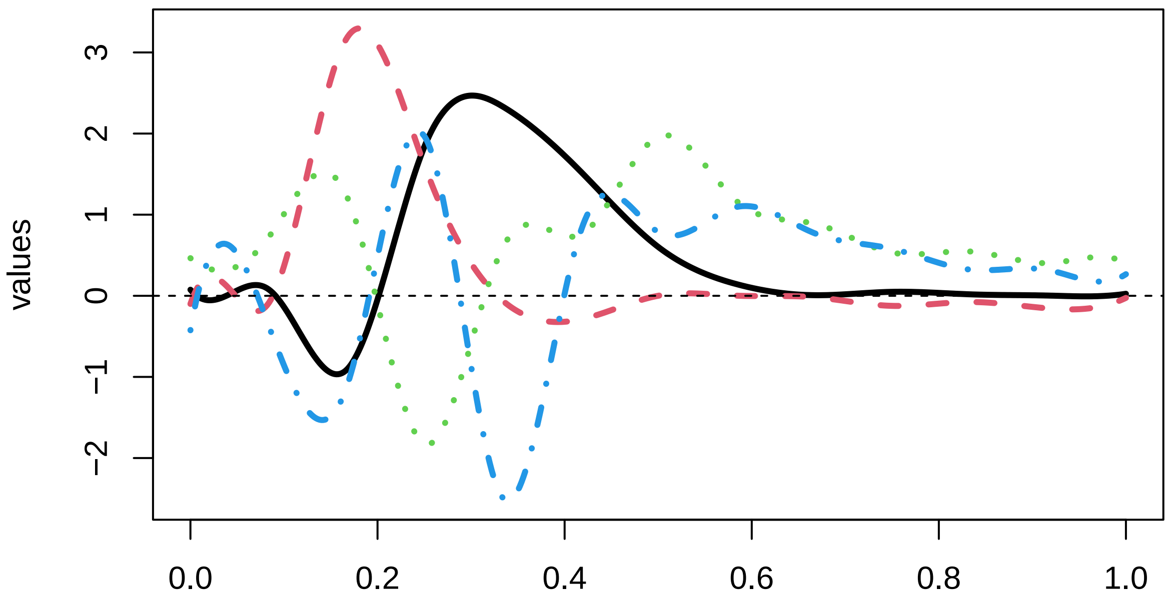
```
## Loading required package: plot3D
```

```
## Warning: package 'plot3D' was built under R version 4.2.3
```

```r
# Load the package
library(plot3D)
Cov_pinch<-var.fd(Pinch.F)
Cov_pinch_mat<-eval.bifd(pts1,pts1,Cov_pinch)
persp3D(pts1,pts1,Cov_pinch_mat,axes=FALSE,colkey=FALSE,bty="g")
```



```r
## (d) Graph the first four EFPC's of the pinch data. How many components do you need to explain 90% of

pinch.pca <- pca.fd(Pinch.F, nharm = 4)
plot(pinch.pca$harmonics, lwd=3)
```




```
## [1] "done"
```

```r
pinch.pca$varprop
```

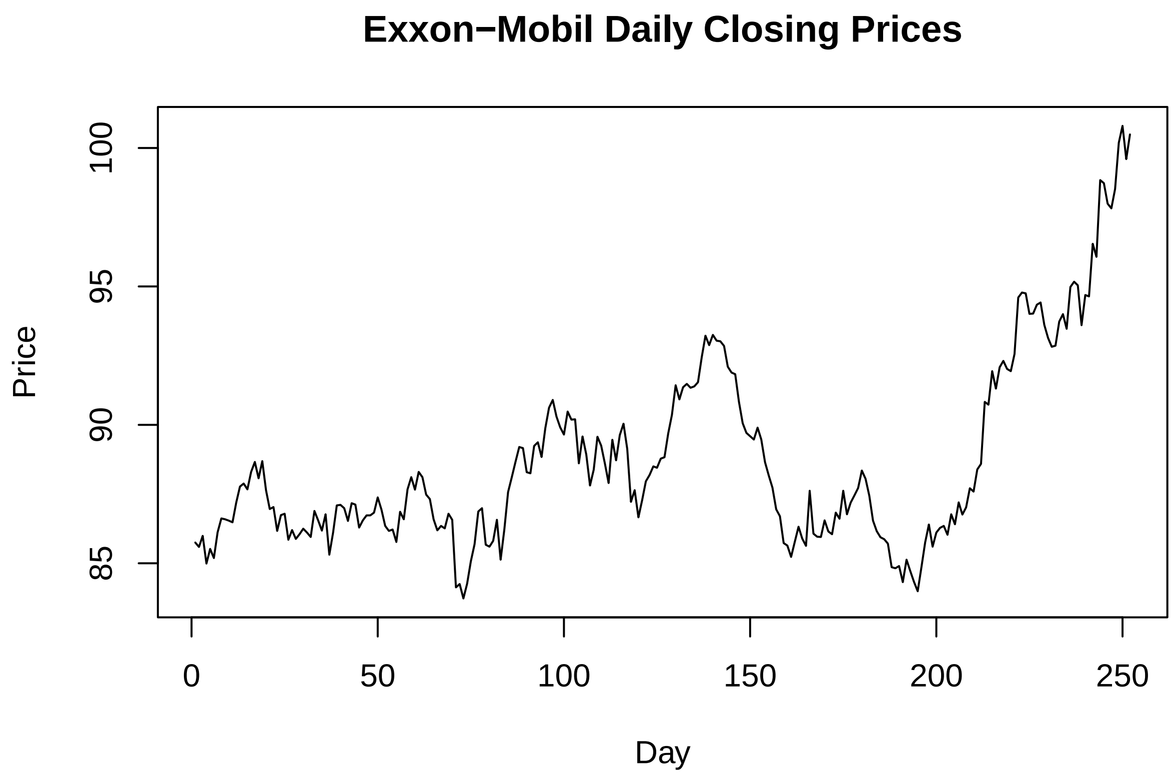
```
## [1] 0.67227408 0.24844734 0.04603413 0.01933435
```

```r
var.explained <- cumsum(pinch.pca$varprop)
n.components <- which(var.explained >= 0.90)[1]

print(paste("Number of components needed to explain 90% of variation: ", n.components))
```

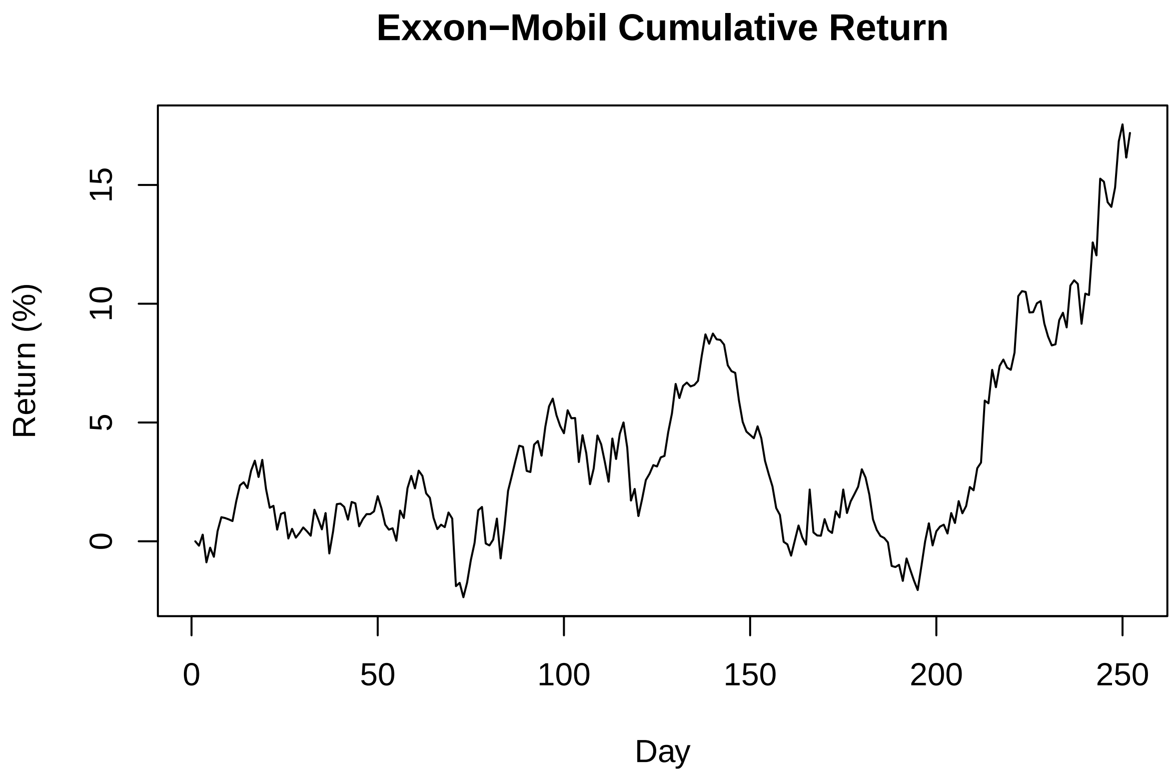
```
## [1] "Number of components needed to explain 90% of variation:  2"
```

```r
#####
## 1.4 This problem illustrates the concept of a functional boxplot using the
## curves of cumulative returns of 30 companies comprising the Dow Jones Industrial Average (DJIA) stoc
## (a) Download the data:
## library("fda")
## d = read.csv("Dow_companies_data.csv")
## data = d[,2:31]
## The first column of the object d is the date; there were 252 trading days in
## 2013. The 252 by 30 matrix data contains the daily closing prices of the 30
## stocks.
## The first column of the object d is the date; there were 252 trading days in 2013. The 252 by 30 mati
```

```r
## closing prices of the 30 stocks.
## Plot the stock value of Exxon-Mobil, ticker symbol XOM. For the same
## stock, calculate and plot the cumulative return function defined as
## Plot the stock value of Exxon-Mobil, ticker symbol XOM. For the same stock, calculate and plot the
## cumulative return function defined as Rn=100*(Pn-P1)/P1, 1 < n < 152.
## where Pn is the closing price on day n, starting from the beginning of the
## year. By how much, in percent, did the Exxon-Mobil stock price increase in 2013?

d = read.csv("C:/Users/fwjbi/OneDrive - Bowling Green State University/Summer 2023/6820-Function/Chapte
data1 <- d[,2:31]

xom_prices <- data1$XOM
plot(xom_prices, type = "l", main = "Exxon-Mobil Daily Closing Prices", xlab = "Day", ylab = "Price")
```



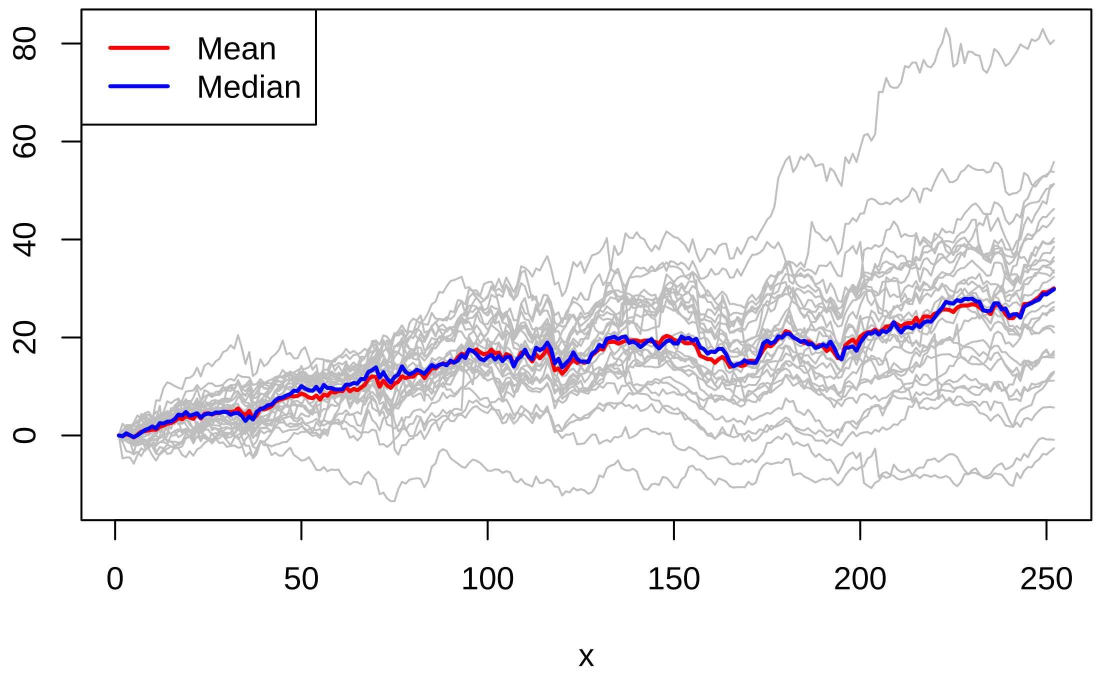

```r
xom_returns <- 100 * (xom_prices - xom_prices[1]) / xom_prices[1]
plot(xom_returns, type = "l", main = "Exxon-Mobil Cumulative Return", xlab = "Day", ylab = "Return (%)"
```



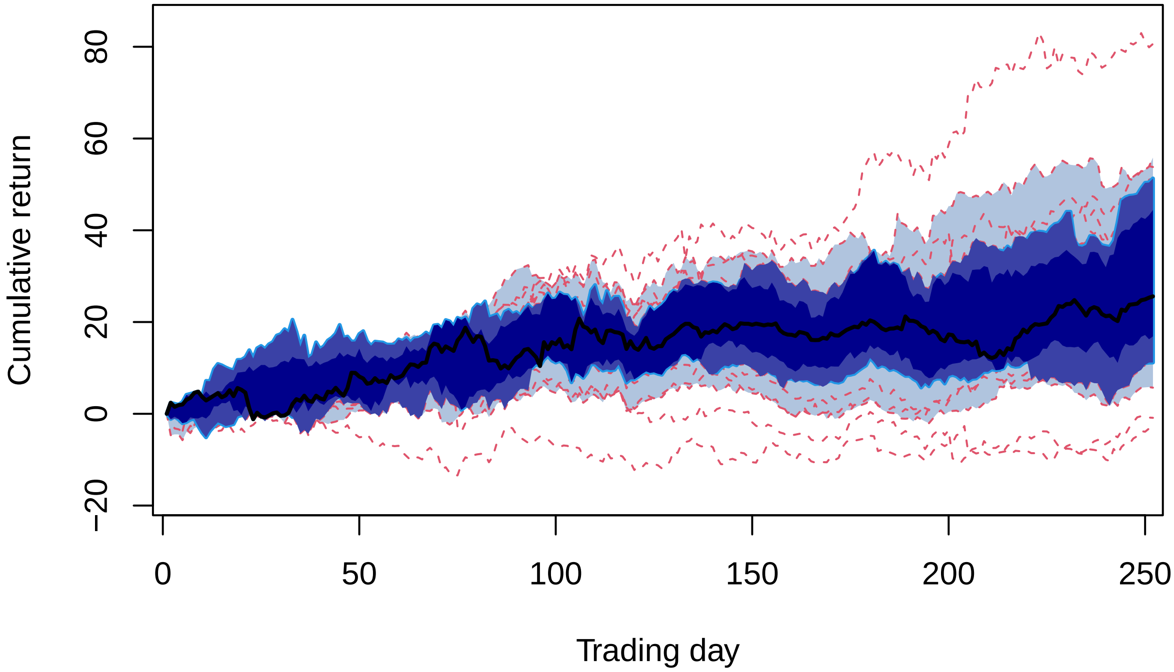

```r
percent_increase <- tail(xom_returns, n = 1) # Get the last value of the return
print(paste("Exxon-Mobil stock price increased by", round(percent_increase, 2), "% in 2013."))
```

```
## [1] "Exxon-Mobil stock price increased by 17.19 % in 2013."
```

```r
#####
## (b) Create a 252 by 30 matrix cr that contains the cumulative returns on
## all stocks in DJIA. Plot all the cumulative return functions in one plot. Add
## the mean and median functions in color.
#####

cr <- matrix(, nrow = 252, ncol = 30)
for (i in 1:30) {
  cr[,i] <- 100 * (data1[,i] - data1[1,i]) / data1[1,i]
}
matplot(cr, type = "l", lty = 1, col = "gray")
mean_cr <- apply(cr, 1, mean)
median_cr <- apply(cr, 1, median)

lines(mean_cr, col = "red", lwd = 2)
lines(median_cr, col = "blue", lwd = 2)
legend("topleft", legend = c("Mean", "Median"), col = c("red", "blue"), lwd = 2)
```




```r
#####
## c) A useful way to visualize the behavior of cumulatve returns is to use
## a functional boxplot, which can be obtained using the function fbplot in
## the fda package. Functional boxplots are analogous to standard boxplots;
## however, they treat each function as a single data point. Use the following
## code to display a functional boxplot for the cumulative return data:
## fbplot(fit = cr, ylim = c(-20,90), xlab="Trading day",
## ylab = "Cumulative return")
## Functional boxplots are not obtained from pointwise boxplots for each day;
## a measure of centrality for each function compared to the other functions is
## used, see Sun and Genton (2011). In the boxplot produced by the above code,
## the black line is the median curve. The magenta region represents the middle
## 50% of curves, and the dashed red curves are the outlying functions. Note that
## functional boxplots produce outlying curves rather than individual outlying
## returns for each day.
## After consulting the help file, produce a functional boxplot which shows
## three regions corresponding to probability levels of 30, 60 and 90 percent,
## using a dark color for the central 30 percent and fading colors for the 60 and
## 90 percent of the curves
#####

suppressWarnings(
  fbplot(fit = cr,xlim = c(7, 245),  ylim = c(-18, 85), xlab="Trading day",
         ylab = "Cumulative return",  prob=c(0.90,0.50,0.60, 0.30),
         color = colorRampPalette(c("lightsteelblue","darkblue"))(4)))
```




```
## $depth
##  [1] 0.4152915 0.3129926 0.2976852 0.1558498 0.1287926 0.4631477 0.4010787
##  [8] 0.3526204 0.3906062 0.2521096 0.4612320 0.4046730 0.4735837 0.3979771
## [15] 0.2310368 0.3479132 0.4568167 0.4265850 0.4872856 0.3478768 0.3127919
## [22] 0.4678366 0.4624726 0.4250160 0.3453772 0.4552842 0.4319308 0.4645525
## [29] 0.4375319 0.3458881
##
## $outpoint
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30
##
## $medcurve
## [1] 19
```