Financial Data Analytics
Final Project Assignment

Markus Pelger

Due date: September 30th, 2017

Instructions:

- **Please submit your final project report by email. In your final report you should answer briefly and concisely each question. Please include any relevant R output and plot with your solution. Rstudio allows you to export and save any plot that you have created. For most questions you simply need to copy the relevant R output and write one short sentence.**
- **The data used in the project is included in the project folder. In this folder I am also providing R code which you can use for answering the questions. Please make sure that you understand the commands.**
- **There are two questions with 45 points in total. In addition there are five bonus questions that give you additional 115 bonus points. These questions are optional. You can get the full score without answering the bonus questions.**

Good luck!

1. **Exploratory data analysis I: 15 points**
   Consider the daily simple returns of Starbucks (SBUX) stock, CRSP value-weighted index (VW), CRSP equal-weighted index (EW), and the S&P composite index (SP) from January 3, 2007 to December 31, 2015. Returns of the three indexes include dividends. The data are in the file `d-sbux3dx-0715.txt` and the columns show permno of SBUX, date, SBUX, vwretd, ewretd, and sprtrn, respectively, with the last four columns showing the simple returns.

   (a) Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of each simple return series.

   (b) Obtain the empirical density function of the simple returns of Starbucks stock. Are the daily simple returns normally distributed? Perform a normality test to justify your answer.

   (c) Transform the simple returns to log returns. Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of each log return series.

   (d) Test the null hypothesis that the mean of the log returns of Starbucks stock is zero. Do the same test for S&P composite index.

(e) Obtain the empirical density plot of the daily log returns of Starbucks stock and the value-weighted index index.

2. **Factor models: 30 points**
Import the following four data sets

- `IndustryPortfolios.csv`
- `FamaFrenchFactors.csv`
- `RiskFreeRate.csv`
- `SizeValuePortfolios.csv`.

The data is taken from Kenneth French's website:
`http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research`,
which provides further information about how the portfolios are formed. The industry portfolios are 30 portfolios based on industry classification codes. The three Fama-French factors are the market return minus the risk-free rate, a size and a value factor. The 25 size and value sorted portfolios are the intersections of 5 portfolios formed on size (market equity, ME) and 5 portfolios formed on the ratio of book equity to market equity (BE/ME).

(a) Apply Principal Component Analysis to the 30 industry portfolios. How many statistical factors do you need to capture the variance. Plot the eigenvalues. (Hint: Use the command `pca = prcomp()`)

(b) Use the commands `pca$x[,1]`, `pca$x[,2]` and `pca$x[,3]` to extract the first 3 statistical factors. Run a regression of the steel industry and food industry on the three PCA factors. Then use the market portfolio included in the Fama-French factors and run again a linear regression on the steel and food industry. Comment on your results. Plot the returns for the steel and food industry against their predicted values based on the various regressions.

(c) Take the second and 15th size-value sorted portfolio. Explain how it is constructed.

(d) Calculate the excess returns on the size and value sorted portfolios.

(e) Fit the CAPM model to the excess returns for the 2nd and 15th size-value sorted portfolio. Can you accept the null hypothesis that the intercepts in each regression are zero? Why or why not? Include the p-values with your work.

(f) Next, you will fit the Fama-French three-factor model. Run two linear regressions of the excess returns for the 2nd and 15th size-value sorted portfolio on the excess market return and the size and value factor. Can you accept the null hypothesis that the intercepts in each regression are zero? Why or why not? Include the p-values with your work.

(g) Use robust standard errors (Newey-West) for the previous four linear regressions. Do your results change?

(h) Apply Principal Component Analysis to the size-value-sorted portfolios. How many statistical factors do you need to capture the variance. Plot the eigenvalues. (Hint: Use the command `pca = prcomp()`)

(i) Use the commands `pca$x[,1]`, `pca$x[,2]` and `pca$x[,3]` to extract the first 3 statistical factors. Run three regressions in which your try to explain the market, value

and size factors with the three PCA factors. Interpret your results. Also include a plot of the fitted values versus the realized values. Comment on the statement: "The Fama-French factors are simply the first three principal components based on the size-value sorted portfolios".

Bonus Questions: Note these questions are optional.

3. **Exploratory data analysis II: 15 points**
   Daily foreign exchange rates (spot rates) can be obtained from the Federal Reserve Bank in St Louis (FRED). The data are the noon buying rates in New York City certified by the Federal Reserve Bank of New York. Consider the exchange rates between the U.S. dollar and the Euro from January 3, 2005 to March 18, 2016. See the file d-exuseu-0516.txt. The file has four columns, namely year, month, day, and euro, respectively, where euro denotes the US dollars of one Euro. Answer the following questions:

   (a) Compute the daily log return of the exchange rate.
   (b) Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of the log returns of the exchange rate.
   (c) Obtain a density plot of the daily log returns of Dollar-Euro exchange rate.
   (d) Test $H_0 : \mu = 0$ versus H_1 $\mu \neq 0$, where μ denotes the mean of the daily log return of Dollar-Euro exchange rate.
   (e) Are the log returns of the exchange rate normally distributed? Why? (Hint: Use a Shapiro Normality Test and also include a quantile-quantile plot.)

4. **Time-series model: 20 points**
   Consider the monthly crude oil prices from January 1986 to February 2016. The original data are from FRED and also available in m-COILWTICO.txt.

   (a) Obtain the time plot of the oil prices and its first differenced series.
   (b) Based on the plots, is the first differenced series weakly stationary? Why?
   (c) Use the ADF test to test for non-stationarity in the oil prices and its first differenced series. Report your results.
   (d) Plot the ACF and PACF functions for the oil price and the differenced series. Draw your conclusion.
   (e) Let r_t be the first differenced price series. Test $H_0 : \rho_1 = \cdots = \rho_{12} = 0$ versus H_1: $\rho_i \neq 0$ for some $1 \leq i \leq 12$. Draw your conclusion.
   (f) Build an AR model for r_t, including model checking. (Hint: Use the PACF to determine the order of the AR(p) process. Use the command `ts.diag` to test if there is autocorrelation left in the residuals.) Refine the model by excluding all estimates with t-ratio less than 1.645. (Hint: the t-ratio is the ratio of the estimate divided by the standard error.) Write down the fitted model.
   (g) Build an ARIMA model for r_t, including model checking. Write down the fitted model. (Hint: an ARMA(1,6) might be a good choice.)
   (h) Use the fitted ARMA(1,6) model to compute 1-step to 4-step ahead forecasts of r_t at the forecast origin February, 2016. Also, compute the corresponding 95% interval forecasts. Report the numbers and also include a plot for your forecast.

5. **GARCH: 20 points**
   Consider the daily returns of Amazon (amzn) stock from January 2, 2009 to December 31, 2014. The simple returns are available from CRSP and in the file `d-amzn3dx0914.txt` (the column with heading `amzn`). Transform the simple returns to log returns. Multiple

the log returns by 100 to obtain the percentage returns. Let r_t be the percentage log returns.

(a) Is the expected value of r_t zero? Why?

(b) Are there any serial correlations in r_t? Why?

(c) Fit a Gaussian ARMA-GARCH model to the r_t series. Obtain the normal QQ- plot of the standardized residuals, and write down the fitted model. Plot the ACF of the standardized and squared standardized residuals. Is the model adequate?

(d) Build an ARMA-GARCH model with Student-t innovations for the $r - t$ series. Perform model checking, including the QQ-plot. Plot the ACF of the standardized and squared standardized residuals. Is the model adequate?

(e) Write down the fitted model.

(f) Obtain 1-step to 5-step ahead mean and volatility forecasts using the fitted ARMA-GARCH model with Student-t innovations. Report the numerical values and the plots for your forecast. Comment on the statement: "Returns are not predictable (for short horizons)".

6. **Time-series model II: 30 bonus points**

Climate change is a big concern for many. In this problem, we analyze the monthly global temperature anomalies from January 1880 to December 2015 for the Northern Hemisphere. The data consist of measurements from land and sea surface and are in 0.01C. The original data are from Goddard Institute for Space Studies. See `http://data.giss.nasa.gov/gistemp`. The data are available in the file `m-globaltemp.txt`.

(a) Plot both the monthly temperature and its first differenced series on the same page. [You can use the command `par(mfcol=c(2,1))` in R.]

(b) Is there a unit root in the temperature series? Why?

(c) Let $z_t = x_t - x_{t-1}$ with x_t being the global temperature. Test $H_0 : E(z_t) = 0$ versus $H_1 : E(z_t) \neq 0$. Draw your conclusion.

(d) Compute ACF and PACF of the z_t series. Plot them on the same page.

(e) Consider the z_t series. Test $H_0 : \rho_1 = \cdots = \rho_{12} = 0$ versus $H_1 : \rho_i \neq 0$ for some $1 \leq i \leq 12$. Draw the conclusion.

(f) Use the command `auto.arima` to identify an AR model for the z_t series. Fit the specified AR model, perform model checking, and write down the fitted model. [You may include the subcommand `include.mean=F` to remove the mean in light of the test result in part (c).

(g) Next, return to the global temperature series x_t. Based on the AR model in part (f), fit a AR model for x_t. Use the fitted model to compute 1-step to 12-step ahead point forecasts of the global temperature at the forecast origin December 2015. For your information, the actual data for January and February 2016 are 153 and 190, respectively.

(h) Compute the 1-step to 2-step ahead 95% interval forecasts for x_t. Are the actual values in these intervals?

(i) Use the command auto.arima in the package forecast to identify an ARIMA model for x_t.

(j) Is the model adequate? Why? Which residual ACF are significantly different from zero, if any?

(k) A refined model can be obtained. But one needs to use the first-differenced series z_t.

```
arima(zt,order=c(1,0,2),seasonal=list(order=c(1,0,0),period=12))
```

Perform model checking. Is the model adequate? Why?.

(l) Compare the seasonal model with the AR model built in (f) for z_t. In terms of in-sample fitting, which model is preferred? Why? Which model is preferred with respect to the AIC criterion? (Hint: The model which has a smaller AIC criterion, is considered to be better.)

(m) Use the seasonal model for 1-step to 12-step ahead point forecasts of the global temperature at the forecast origin December 2015. Are your results different from (g) and (h)?

7. **GARCH II: 30 bonus points**

Consider the monthly returns of the value-weighted index, including dividends from 1966 to 20014. The simple returns are in the file `m-mcd3dx6614.txt` (column with heading `vwretd`). Transform the simple returns to log returns.

(a) Find an adequate model for the monthly log return series. Perform model checking to justify your model.

(b) Obtain 1-step to 5-step ahead predictions of the log return and its volatility at the forecast origin December 2015.

(c) Fit an APARCH model to the monthly log return series. Write down the model. Is the leverage effect statistically significant? Why?

You can use the following R code for answering the questions:

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
#This command sets the working directory to the directory of the R script
#If you use a windows computer please click on "Session" and "Set Working Directory"
#"To Source File Location"

### Problem 1

da=read.table("d-sbux3dx-0715.txt",header=T)
head(da)

rtn=da[,3:6]
attach(rtn)
require(fBasics)

# (a)
basicStats(SBUX)
basicStats(vwretd)
basicStats(ewretd)
basicStats(sprtrn)

# (b)
densitySBUX=density(rtn$SBUX)
plot(densitySBUX)
# This is the same density plot but with labels
plot(densitySBUX,xlab='return',ylab='SBUX',main='Density')

normalTest(SBUX)

# (c)
lrtn=log(rtn+1)  ### log returns
basicStats(lrtn$SBUX)
basicStats(lrtn$vwretd)
basicStats(lrtn$ewretd)
basicStats(lrtn$sprtrn)

# (d)
t.test(lrtn$SBUX)
t.test(lrtn$sprtrn)

# (e)
densitylogSBUX=density(lrtn$SBUX)
densitylogvwretd=density(lrtn$vwretd)
par(mfcol=c(1,2))  ## Put two plots in a frame (left and right)
plot(densitylogSBUX,xlab='log-return',ylab='SBUX',main='Density')
plot(densitylogSBUX,xlab='log-return',ylab='vw index',main='Density')
par(mfcol=c(1,1))



#Problem 5
```

```
FFfactors <- read.csv("FamaFrenchFactors.csv",header=T) # Load csv data with names.
head(FFfactors)  # See the first 6 rows
attach(FFfactors) #Now you can refer directly to the names of the variables.
#Note that you might receive a warning: "The following objects are masked from FFfactors ..."
# This just means that you are overwriting variables with these names that you have used before.
# For our purposes you can simply ignore this message.
names(FFfactors)

rf <- read.csv("RiskFreeRate.csv",header=T) # Load csv data with names.
head(rf)  # See the first 6 rows

sv <- read.csv("SizeValuePortfolios.csv",header=T) # Load csv data with names.
head(sv)  # See the first 6 rows
names(sv)

Industry=read.csv("IndustryPortfolios.csv",header=T) # Load csv data with names.
head(Industry)  # See the first 6 rows
names(Industry)

# (a)
Industry=Industry[,2:31]
attach(Industry) #Now you can refer directly to the names of the variables.
#Note that you might receive a warning: "The following objects are masked from..."
# This just means that you are overwriting variables with these names that you have used before.
# For our purposes you can simply ignore this message.
industrypca=prcomp(Industry)
summary(industrypca)
plot(industrypca)

# (b)
fitSteel=lm(Steel~industrypca$x[,1:3])
summary(fitSteel)
plot(Steel,fitSteel$fitted.values)
abline(c(0,1))

fitSteel2=lm(Steel~Mkt)
summary(fitSteel2)
plot(Steel,fitSteel2$fitted.values)
abline(c(0,1))

fitFood=lm(Food~industrypca$x[,1:3])
summary(fitFood)
plot(Food,fitFood$fitted.values)
abline(c(0,1))

fitFood2=lm(Steel~Mkt)
summary(fitFood2)
plot(Food,fitFood2$fitted.values)
abline(c(0,1))

# (d)
```

```
svrf=sv[,2:26]-rf$RF

# (e)
fit1=lm(svrf[[2]]~Mkt)
summary(fit1)

fit3=lm(svrf[[15]]~Mkt)
summary(fit3)

# (f)
fit2=lm(svrf[[2]]~Mkt+SMB+HML)
summary(fit2)

fit4=lm(svrf[[15]]~Mkt+SMB+HML)
summary(fit4)

# (g)
library(lmtest)
library(sandwich)

coeftest(fit1, vcov = vcov(fit1))
coeftest(fit1, vcov = NeweyWest(fit1))

coeftest(fit2, vcov = vcov(fit2))
coeftest(fit2, vcov = NeweyWest(fit2))

coeftest(fit3, vcov = vcov(fit3))
coeftest(fit3, vcov = NeweyWest(fit3))

coeftest(fit4, vcov = vcov(fit4))
coeftest(fit4, vcov = NeweyWest(fit4))

# (h)
pcasv=prcomp(svrf)
summary(pcasv)
plot(pcasv)

# (i)
fit5=lm(Mkt~pcasv$x[,(1:3)])
summary(fit5)
plot(Mkt,fit5$fitted.values,ylab="fitted values")
abline(c(0,1))

fit6=lm(SMB~pcasv$x[,(1:3)])
summary(fit6)
plot(SMB,fit6$fitted.values,ylab="fitted values")
abline(c(0,1))

fit7=lm(HML~pcasv$x[,(1:3)])
summary(fit7)
plot(HML,fit7$fitted.values,ylab="fitted values")
abline(c(0,1))
```

```
## Problem 3
da <- read.table("d-exuseu-0516.txt",header=T)
head(da)

# (a)
rtn=diff(log(da$euro))  ## Compute log return

# (b)
basicStats(rtn)

# (c)
densityEX=density(rtn)
plot(densityEX,xlab="ln-rtn",ylab='euro',main='Density EX')

# (d)
t.test(rtn)

# (e)
normalTest(rtn)

qqnorm(rtn) # Quantile-quantile plot
qqline(rtn) # Include a line in the quantile-quantile plot.
# If empirical values are normally distributed
# then the quantiles should all be on the line.


### Problem 4
da=read.table("m-COILWTICO.txt",header=T)
head(da)

# (a)
oil=da$VALUE
doil=diff(oil)
dim(da)

# (b)
tdx <- c(1:362)/12+1986
plot(tdx,oil,xlab='year',ylab='coil',type='l')
plot(tdx[-1],doil,xlab='year',ylab='diff(oil)',type='l')

# (c)
# If you have not installed the package fUnitRoots please use the following command:
#install.packages("fUnitRoots")
require(fUnitRoots)
adfTest(oil,lags=11,type="c")
adfTest(doil,lags=11,type="c")

# (d)
acf(oil)
```

```
pacf(oil)

acf(doil)
pacf(doil)

# (e)
Box.test(doil,lag=12,type="Ljung")

# (f)
library("forecast")
ARoil=arima(doil,c(6,0,0))
ARoil
tsdiag(ARoil)

# You can also use the auto.arima command to fit an AR model
# ARoil2=auto.arima(doil,max.p = 20, max.q = 0,  d = 0)
# ARoil2
# tsdiag(ARoil2)

# (f)
#You can use the auto.arima command to fit an ARMA model
# ARIMAoil=auto.arima(doil)
# ARIMAoil
# tsdiag(ARIMAoil)

ARIMAoil2=arima(doil,order=c(1,0,6))
ARIMAoil2
tsdiag(ARIMAoil2,gof=24)

# (g)
Oilpredict=predict(ARIMAoil2,4)
Oilpredict

lcl=Oilpredict$pred-1.96*Oilpredict$se
ucl=Oilpredict$pred+1.96*Oilpredict$se
cf=cbind(lcl,ucl)
cf

Oilforecast=forecast(ARIMAoil2,4)
Oilforecast
plot(Oilforecast)
# The next plot includes only the last 24 observations and labels
plot(Oilforecast,include=24,xlab="time",ylab="oil return",main="Prediction")


#Problem 5
require(forecast)

da=read.table("d-amzn3dx0914.txt",header=T)
head(da)
rt=log(da$amzn+1)*100
```

```
# (a)
t.test(rt)

# (b)
acf(rt)
Box.test(rt,lag=10,type='Ljung')

# (c)
library(rugarch)

garch.norm = ugarchspec(mean.model=list(armaOrder=c(0,0)),
                        variance.model=list(garchOrder=c(1,1)))
amazonGarch = ugarchfit(data=rt, spec=garch.norm)
show(amazonGarch)
plot(amazonGarch, which="all")
plot(amazonGarch, which=9)
plot(amazonGarch, which=10)
plot(amazonGarch, which=11)

# (d)
arma.garch.t = ugarchspec(mean.model=list(armaOrder=c(0,0)),
                          variance.model=list(garchOrder=c(1,1)),
                          distribution.model = "std")
amazonGarch.t = ugarchfit(data=rt, spec=arma.garch.t)
show(amazonGarch.t)
plot(amazonGarch.t, which="all")
plot(amazonGarch.t, which=9)
plot(amazonGarch.t, which=10)
plot(amazonGarch.t, which=11)

# (f)
amazonforecast=ugarchforecast(amazonGarch.t, data = rt, n.ahead = 5)
show(amazonforecast)
plot(amazonforecast,which=1)
plot(amazonforecast,which=3)




#### Problem 6
require(forecast)

da <- read.table("m-globaltemp.txt",header=T)
dd <- da[,2:13]
xt <- c(t(dd))
zt <- diff(xt)
length(xt)
tdx <- c(1:1632)/12+1880

# (a)
par(mfcol=c(2,1))
```

```
plot(tdx,xt,xlab='year',ylab='temp',type='l')
plot(tdx[-1],zt,xlab='year',ylab='diff(temp)',type='l')
par(mfcol=c(1,1))

# (b)
require(fUnitRoots)
adfTest(xt,lags=11,type="c")
adfTest(zt,lags=11,type="c")

# (c)
t.test(zt)

# (d)
par(mfcol=c(2,1))
acf(zt)
pacf(zt)
par(mfcol=c(1,1))

# (e)
Box.test(zt,lag=12,type='Ljung')

# (f)
ARtemp=auto.arima(zt,max.p = 20, max.q = 0,  d = 0)
ARtemp
tsdiag(ARtemp)

ARtemp2=arima(zt,order=c(11,0,0),include.mean=F)
ARtemp2
tsdiag(ARtemp2,gof=24)

# (g)
ARtemp3=arima(xt,order=c(11,1,0))
ARtemp3

PredictTemp <- predict(ARtemp3,12)
PredictTemp

# (h)
lcl <- PredictTemp$pred-1.96*PredictTemp$se
ucl <- PredictTemp$pred+1.96*PredictTemp$se
cf <- cbind(lcl,ucl)
cf[1:2,]
cf[1:12,]

# (i)
require(forecast)

ARIMAtemp=auto.arima(xt)
ARIMAtemp
tsdiag(ARIMAtemp,gof=24)
```

```
ARIMAtemp2=arima(zt,order=c(1,0,2),seasonal=list(order=c(1,0,0),period=12))
ARIMAtemp2
tsdiag(ARIMAtemp2,gof=24)

# (j)
ARIMAtemp3=arima(xt,order=c(1,1,2),seasonal=list(order=c(1,0,0),period=12))
ARIMAtemp3
tsdiag(ARIMAtemp3,gof=24)

PredictTemp=predict(ARIMAtemp3,12)
PredictTemp

lcl <- PredictTemp$pred-1.96*PredictTemp$se
ucl <- PredictTemp$pred+1.96*PredictTemp$se
cf <- cbind(lcl,ucl)
cf[1:2,]
cf[1:12,]

Tempforecast=forecast(ARIMAtemp3,12)
Tempforecast
plot(Tempforecast)
plot(Tempforecast,include=24,xlab="time",ylab="temperature",main="Prediction")


#Problem 7
da=read.table("d-sbux3dx-0715.txt",header=T)
head(da)

rtn=da[,3:6]
attach(rtn)

vw=log(da$vwretd+1)

# (a)
t.test(vw)
Box.test(vw,lag=12,type='Ljung')
acf(vw)
acf(vw^2)

garch.norm = ugarchspec(mean.model=list(armaOrder=c(0,0),include.mean = FALSE),
                        variance.model=list(garchOrder=c(1,1)))
vwGarch = ugarchfit(data=vw, spec=garch.norm)
show(vwGarch)
plot(vwGarch, which="all")
plot(vwGarch, which=9)

arma.garch.t = ugarchspec(mean.model=list(armaOrder=c(0,0),include.mean = FALSE),
                          variance.model=list(garchOrder=c(1,1)),
                          distribution.model = "std")
vwGarch.t = ugarchfit(data=vw, spec=arma.garch.t)
show(vwGarch.t)
plot(vwGarch.t, which="all")
```

```
plot(vwGarch.t, which=9)

# (b)
vwforecast=ugarchforecast(vwGarch.t, data = vw, n.ahead = 5)
show(vwforecast)
plot(vwforecast,which=1)
plot(vwforecast,which=3)

# (c)
arma.aparch.t = ugarchspec(mean.model=list(armaOrder=c(0,0),include.mean = FALSE),
                           variance.model=list(model="apARCH",
                                               garchOrder=c(1,1)),
                           distribution.model = "std")
vwGarch.a = ugarchfit(data=vw, spec=arma.aparch.t)
show(vwGarch.a)
```