FERM-504-Project_II.R

selim

2021-01-07

```r
# I have tried doing homework completely on R.
# Necessary libraries for code
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(ggplot2)
library(readxl)

# Set working directory
setwd("~/Documents/Ozu/FERM 504/HW")

# Load the ntickers of stocks that comprise BIST30
tickers<- as.data.frame(read_excel("BIST30.xlsx"))

# OYAKC stock values do not exist in at Yahoo so I downloaded
# OYAKC separately
oyakc <- as.data.frame(read_excel("OYAKC.xlsx"))

# Load risk free returns and BIST30 index returns
market<- as.data.frame(read_excel("IndexReturns.xlsx"))

# Add ".IS" to the end of tickers for download
tickers<- tickers[,2]
for (k in 1:29)
  tickers[k]<- paste0(tickers[k],".IS")

# Set start and end date
from.dat <- as.Date("11/01/15", format="%m/%d/%y")
```

```r
to.dat <- as.Date("11/02/20", format="%m/%d/%y")

# Create empty vector, download stocks
# and bind adjusted values together
data <-c()
for (x in tickers){
  getSymbols(x,periodicity = "monthly", src="yahoo", from = from.dat, to = to.dat)
  data<-cbind(data,Ad(get(x)))
}
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```r
# Compute returns on adjusted values for 29 stocks, except OYAKC
for (x in 1:29){
  vec=as.vector(data[,x])
  diff_vec=diff(vec)
  diff_vec <- diff_vec[!is.na(diff_vec)]
  vec=diff_vec/vec[1:length(diff_vec)]
  data[2:61,x]=vec
}
data<-data[2:61,]

# Compute OYAKC returns and combine with data
# dates reversed for alignment of data
Re.oyakc=(rev(oyakc[,9])/100)[13:72]
data <- cbind(data,Re.oyakc)

# Combine BIST30 index returns and risk free rates
# Put stock names as column names of data
data=cbind(data, rev(market[,2]),rev(market[,3]))
colnames(data)<-c(tickers,"OYAKC.IS","XU100","R_free")

# Deduct risk free rate from all returns to compute risk premium
for (x in 1:31){
  vec=as.vector(data[,x]-data[,32])
  data[,x]=vec
}


# Make regression analysis for each stock against BIST30 returns
# Compute alpha, beta, R_2 squared and standard deviation of
# regression of each stock
alpha<-c()
beta <-c()
sigma<-c()
r_2<-c()
for (x in 1:30){
```

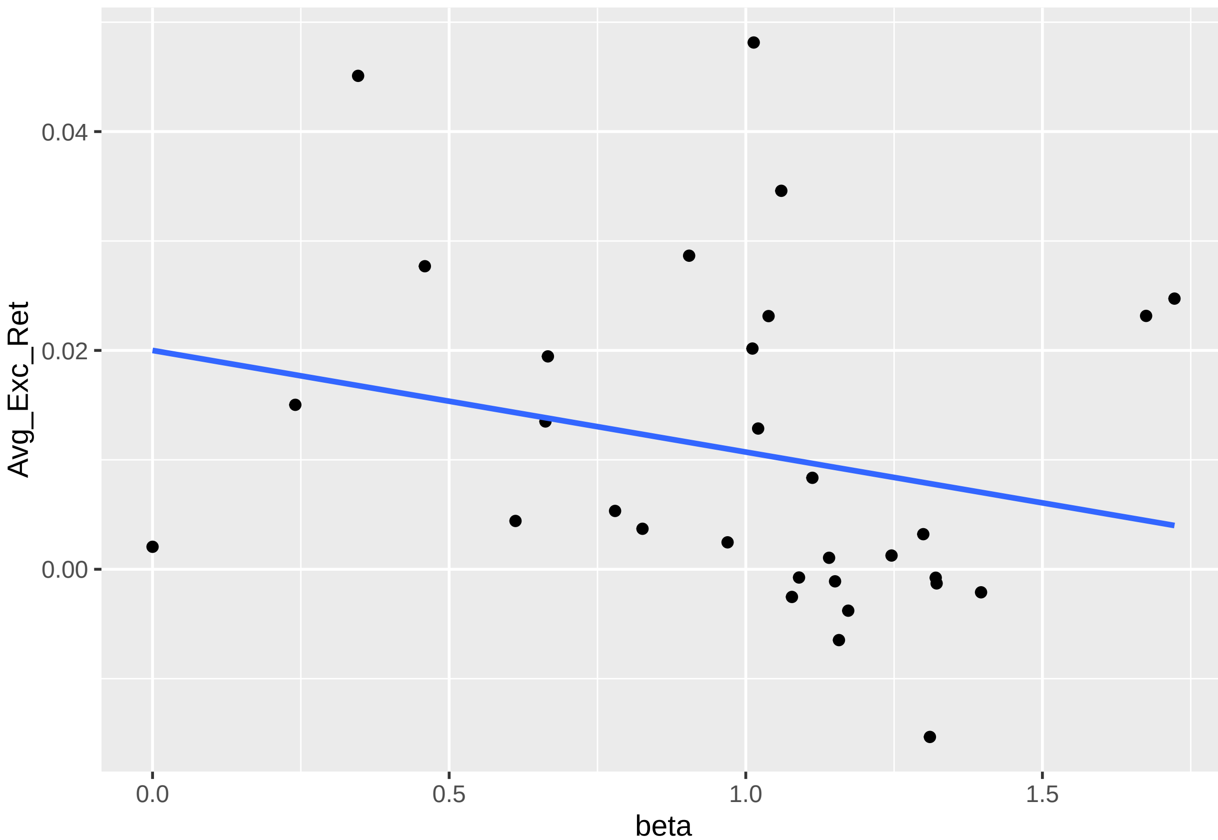
```r
  formula=paste0(colnames(data)[x],"~","XU100")
  regstats <- lm(formula=formula, data=data)
  alpha<-c(alpha,regstats$coefficients[1])
  beta<-c(beta,regstats$coefficients[2])
  sigma <-c(sigma,as.numeric(summary(regstats)[6]))
  r_2<-c(r_2,as.numeric(summary(regstats)[8] ))
}

# Create a summary data-frame for regression statistics
results_df <- data.frame(matrix(NA, nrow=30, ncol=5))
results_df[,1]=alpha
results_df[,2]=beta
results_df[,3]=sigma
results_df[,4]=r_2

# Assign row names as stock codes
rownames(results_df)<-c(tickers,"OYAKC.IS")

# Assign column names as regression statistics
colnames(results_df)<-c("alpha","beta","sigma","r2","Avg_Exc_Ret")

# Compute average excess returns of each stock and
for (x in 1:30){
  results_df[x,5]=mean(data[,x])
}

# Compute last line for BIST30 average excess return
results_df<-rbind(results_df,c(0,0,0,0,mean(data[,31])))

# Add XU100 to the names of rows
rownames(results_df)<-c(rownames(results_df)[1:30],"XU100")

# Printout of Result Data-Frame
results_df
```

```
##                   alpha      beta      sigma          r2   Avg_Exc_Ret
## AKBNK.IS -0.0061925182 1.1726770 0.07251077 0.549410623 -0.0037833017
## ARCLK.IS  0.0037291540 0.7796468 0.08968040 0.260542336  0.0053309062
## ASELS.IS  0.0443836623 0.3464981 0.27979491 0.007098922  0.0450955284
## BIMAS.IS  0.0145349292 0.2406306 0.13940637 0.013699668  0.0150292949
## DOHOL.IS  0.0267532441 0.4590126 0.15158827 0.040992490  0.0276962665
## EKGYO.IS -0.0088503854 1.1570071 0.08486172 0.464263159 -0.0064733619
## EREGL.IS  0.0210042470 1.0383925 0.08994104 0.383249188  0.0231375815
## GARAN.IS -0.0040070019 1.3216767 0.06024451 0.691717448 -0.0012916716
## GUBRF.IS  0.0324113400 1.0598552 0.14968838 0.189438115  0.0345887688
## HALKB.IS -0.0180161695 1.3103439 0.06631298 0.645425040 -0.0153241219
## ISCTR.IS -0.0013038932 1.2456861 0.06000946 0.667644355  0.0012553175
## KCHOL.IS -0.0012952980 1.1403805 0.04898285 0.716459127  0.0010475667
## KOZAA.IS  0.0460591664 1.0133757 0.18936467 0.117783045  0.0481411050
## KOZAL.IS  0.0267988754 0.9045040 0.12814172 0.188493424  0.0286571416
## KRDMD.IS  0.0211979384 1.7226348 0.10154694 0.572938202  0.0247370206
## MGROS.IS  0.0060727964 1.1122618 0.07886807 0.481113899  0.0083578923
## PETKM.IS  0.0121457363 0.6622781 0.12521387 0.115372102  0.0135063594
## PGSUS.IS  0.0197147180 1.6747766 0.14007800 0.399905752  0.0231554774
```

```
## SAHOL.IS -0.0047433172 1.0777665 0.04197653 0.754496986 -0.0025290903
## SISE.IS   0.0107652445 1.0208287 0.07495975 0.463691138  0.0128624949
## TAVHL.IS -0.0034664324 1.1504455 0.09147351 0.424428759 -0.0011028894
## TCELL.IS  0.0004698936 0.9693168 0.05543403 0.587700287  0.0024613151
## THYAO.IS  0.0005374066 1.2991754 0.09398882 0.471104539  0.0032065089
## TKFEN.IS  0.0180950628 1.0112030 0.09301666 0.355237740  0.0201725377
## TSKB.IS   0.0180873431 0.6664247 0.20542055 0.046770920  0.0194564851
## TTKOM.IS -0.0029918081 1.0897439 0.06565745 0.562217087 -0.0007529742
## TUPRS.IS  0.0020050326 0.8258806 0.07888393 0.338187046  0.0037017703
## VAKBN.IS -0.0049740803 1.3965663 0.06390638 0.690054940 -0.0021048926
## YKBNK.IS -0.0034852853 1.3201374 0.06403671 0.664572966 -0.0007731174
## OYAKC.IS  0.0031587025 0.6117911 0.11374973 0.118830832  0.0044156021
## XU100     0.0000000000 0.0000000 0.00000000 0.000000000  0.0020544587
```

```r
# Plot Average Excess Returns against Betas
# Add a trend-line
ggplot(data=results_df,aes(x=beta,y=Avg_Exc_Ret))+
geom_point() +
  geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```r
# Create a cross regression of average excess returns for BIST30 stocks
# on their beta and risk estimates (sigmas in previous calculation).
regstats <- lm(Avg_Exc_Ret ~ beta+sigma, data=results_df[1:30,])
summary(regstats)
```

```
##
## Call:
```

```
## lm(formula = Avg_Exc_Ret ~ beta + sigma, data = results_df[1:30,
##     ])
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0185294 -0.0063761 -0.0005612  0.0067718  0.0154836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.022352   0.008928  -2.503   0.0186 *
## beta         0.006304   0.005844   1.079   0.2902
## sigma        0.260840   0.038428   6.788 2.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00923 on 27 degrees of freedom
## Multiple R-squared:  0.6697, Adjusted R-squared:  0.6452
## F-statistic: 27.37 on 2 and 27 DF,  p-value: 3.201e-07
```

```
# Analysis of Regression
# R_2 squared is quite high (67%), therefore we can say
# our model is a good model for explanation of deviation

# We have e negative intercept amount (-2.24%) which is statistically significant
# therefore we can reject null hypothesis intercept is zero
# and say confidently that intercept is not zero with 95% confidence level.

# Slope on beta is (0.63%) with standard deviation of (0.58%).
# As slope=0 value is inside confidence interval (around 2.05 times  of standard deviation)
# given as(-0.57%, 1.82%), we cannot reject null hypothesis
# that slope is not zero at 95% confidence level. Slope may be zero.

# Market premium is (0.21%).
# As market premium value is inside confidence interval (around 2.05 times  of standard deviation)
# given as(-0.57%, 1.82%), we cannot reject null hypothesis
# that slope is not market premium at 95% confidence level. Slope may be equal to market premium.

# We can reject null hypothesis that slope on sigma is zero as
# coefficient of sigma is statistically significant and
# confidence interval of slope of coefficient
# is computed as (18%, 34%) and we can reject null hypothesis
# that slope is zero at 95% confidence level.

# Since intercept and slope of sigma are significantly
# different from zero we cannot say CAPM holds for Turkish market.
```