Asset Pricing Models and Regression Analysis

Introduction

Here I look at different asset pricing models, including the Capital Asset Pricing Model (commonly known as CAPM) and also the Fama-French three factor model (FF). Both of these models are based on linear relationships, and so will be examined using a series of linear regressions.

First we import some packages that we'll need later:

```
require( dplyr )
require( lubridate )
require( quantreg )
require( stargazer )
```

Data

The analysis requires stock returns data for an individual security, as well as returns data for the three Fama-French factors. The FF returns data can be downloaded from the **French Data Library**.

```
# import the FF factor returns from a CSV file
FF.Data = read.csv( file = 'F-F_Research_Data_Factors_daily.CSV' , header = TRUE )
head( FF.Data )
```

```
##      Dates Mkt.RF   SMB   HML    RF
## 1 19260701   0.10 -0.24 -0.28 0.009
## 2 19260702   0.45 -0.32 -0.08 0.009
## 3 19260706   0.17  0.27 -0.35 0.009
## 4 19260707   0.09 -0.59  0.03 0.009
## 5 19260708   0.21 -0.36  0.15 0.009
## 6 19260709  -0.71  0.44  0.56 0.009
```

We similarly pull the stock returns data from a **posted Kaggle dataset**.

```
# import the FF factor returns from a CSV file
Stock.Price.Data = read.csv( file = 'apple.csv' , header = TRUE )
head( Stock.Price.Data )
```

```
##         Date   Open   High    Low  Close   Volume Ex.Dividend Split.Ratio
## 1 2010-01-04 213.43 214.50 212.38 214.01 17633200           0           1
## 2 2010-01-05 214.60 215.59 213.25 214.38 21496600           0           1
## 3 2010-01-06 214.38 215.23 210.75 210.97 19720000           0           1
## 4 2010-01-07 211.75 212.00 209.05 210.58 17040400           0           1
## 5 2010-01-08 210.30 212.00 209.06 211.98 15986100           0           1
## 6 2010-01-11 212.80 213.00 208.45 210.11 16508200           0           1
##   Adj..Open Adj..High Adj..Low Adj..Close Adj..Volume
## 1  27.42873  27.56624 27.29379   27.50327   123432400
## 2  27.57909  27.70632 27.40560   27.55082   150476200
## 3  27.55082  27.66005 27.08431   27.11259   138040000
## 4  27.21283  27.24495 26.86584   27.06246   119282800
## 5  27.02648  27.24495 26.86712   27.24238   111902700
## 6  27.34777  27.37347 26.78873   27.00206   115557400
```

Cleaning & Preprocessing

There are just a few things to clean up here before we can do our analysis. The stock data, for example, has come to us in price terms, and in order to perform our analysis we will actually need the returns data. And so in this next snippet we get the lagged adjusted closing prices and then compute the lognormal returns as a new column in our dataframe. Note that in the first mutate line we also just do some simple date converstion to tell R that this column is a Date.

```
Stock.Price.Manipulated =
  Stock.Price.Data %>%
  mutate( Format.Date = as.Date( Date ) ) %>%
  mutate( Last.Close.Price = lag( Adj..Close ) ) %>%
  mutate( Log.Returns = 100*log( Adj..Close / Last.Close.Price ) )
```

In order to choose our time period, let's look backwards two years from the last available pricing date in this dataset. Two years is usually a fairly standard time frame over which to look at an asset pricing model, although it's important to note that these relationships are dynamic through time:

```
Most.Recent.Stock.Date = max( Stock.Price.Manipulated$Format.Date )
Start.Date = seq( from = Most.Recent.Stock.Date , by = "-2 years" , length= 2 )[2]

Most.Recent.Stock.Date
```

```
## [1] "2016-12-30"
```

```
Start.Date
```

```
## [1] "2014-12-30"
```

Now let's use this information to subset our sample:

```
Stock.Price.Subset =
  Stock.Price.Manipulated %>%
  select( c( "Format.Date" , "Log.Returns" ) ) %>%
  filter( Format.Date >= Start.Date )
```

Now that we have the security data in a good place, let's go back to the FF returns data. We'll perform a few simple manipulations as outlined in the comments below:

```
# rename the first column of FF data
colnames( FF.Data )[1] = 'Date'

# convert text dates to R date objects
FF.Data =
  FF.Data %>%
  mutate( Date = as.character( Date ) ) %>%
  mutate( Format.Date = as.Date( Date , format = '%Y%m%d' ) )
```

Now comes the key step of joining the two datasets together, which R and specifically **dplyr** makes extremely easy for us:

```
Combined.Data =
  Stock.Price.Subset %>%
  left_join( FF.Data , by = c( "Format.Date" = "Format.Date" ) ) %>%
  mutate( Stock.RFR = Log.Returns - RF ) # this last row removes the RFR from the stock returns

head(Combined.Data)
```

```
##   Format.Date  Log.Returns     Date Mkt.RF   SMB   HML RF    Stock.RFR
## 1  2014-12-30 -1.227767929 20141230  -0.48  0.06  0.35  0 -1.227767929
```

```
## 2  2014-12-31 -1.920202561 20141231  -0.93  0.50 -0.39  0 -1.920202561
## 3  2015-01-02 -0.955812656 20150102  -0.11 -0.59  0.09  0 -0.955812656
## 4  2015-01-05 -2.857602364 20150105  -1.84  0.33 -0.63  0 -2.857602364
## 5  2015-01-06  0.009411322 20150106  -1.04 -0.78 -0.27  0  0.009411322
## 6  2015-01-07  1.392480796 20150107   1.19  0.17 -0.65  0  1.392480796
```

Note that in the last step there we also computed a new column, which is to subtract the risk free rate (RFR) from the daily security returns. This is because we want the returns in excess of (over and above) the risk free rate for our pricing models.

OLS Regression Analysis

Now that we have all of our data together in a single dataframe, we can begin our actual analysis. First, let's look at the CAPM:

```
CAPM_Reg = lm( Stock.RFR ~ Mkt.RF
               , data = Combined.Data )

stargazer( CAPM_Reg
           , summary = TRUE
           , title = 'CAPM Results'
           , type = 'text'
           , no.space = TRUE )
```

```
##
## CAPM Results
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                              Stock.RFR
## -----------------------------------------------
## Mkt.RF                       1.062***
##                               (0.061)
## Constant                     -0.017
##                               (0.055)
## -----------------------------------------------
## Observations                    506
## R2                             0.379
## Adjusted R2                    0.377
## Residual Std. Error      1.247 (df = 504)
## F Statistic          307.109*** (df = 1; 504)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

The result from this analysis is that our CAPM beta is 1.06. Also note that the R-squared of the model is about 38% – 38% of the movement of this security can be explained by the movement in the market as a whole.

Let's also plot these results just for visual effect:

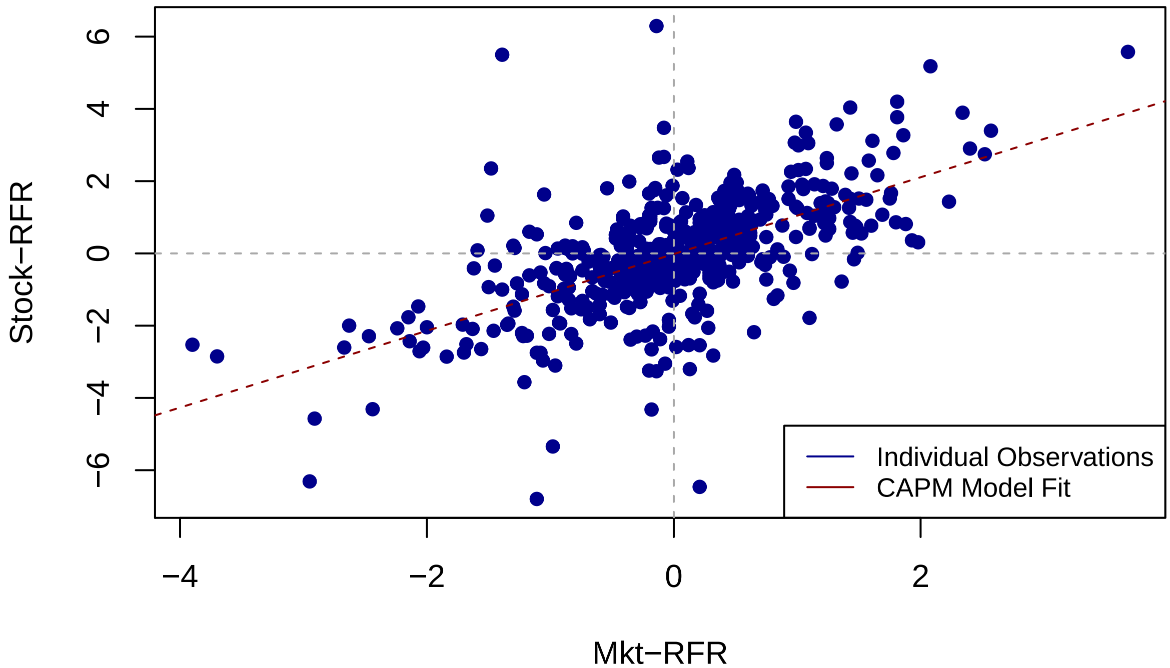
```
plot( x = Combined.Data$Mkt.RF
      , y = Combined.Data$Stock.RFR
      , pch = 16
      , col = 'darkblue'
      , xlab = 'Mkt-RFR'
```

```
       , ylab = 'Stock-RFR')

abline( h = 0 , lwd = 1 , lty = 2 , col = 'darkgrey' )
abline( v = 0 , lwd = 1 , lty = 2 , col = 'darkgrey' )

abline( coef( CAPM_Reg ) , lwd = 1 , lty = 2 , col = 'darkred' )

legend( 'bottomright'
         , legend = c( 'Individual Observations' , 'CAPM Model Fit' )
         , col = c( 'darkblue' , 'darkred' )
         , lwd = 1
         , lty  = 1
         , cex = 0.8 )
```



Now let's perform the same sort of analysis, but using the FF model:

```
FF_Reg = lm( Stock.RFR ~ Mkt.RF + SMB + HML
             , data = Combined.Data  )

stargazer( FF_Reg
           , summary = TRUE
           , title = 'FF Results'
           , type = 'text'
           , no.space = TRUE )
```

```
##
## FF Results
## ===============================================
##                          Dependent variable:
##                      ---------------------------
##                               Stock.RFR
## -----------------------------------------------
## Mkt.RF                        1.094***
```

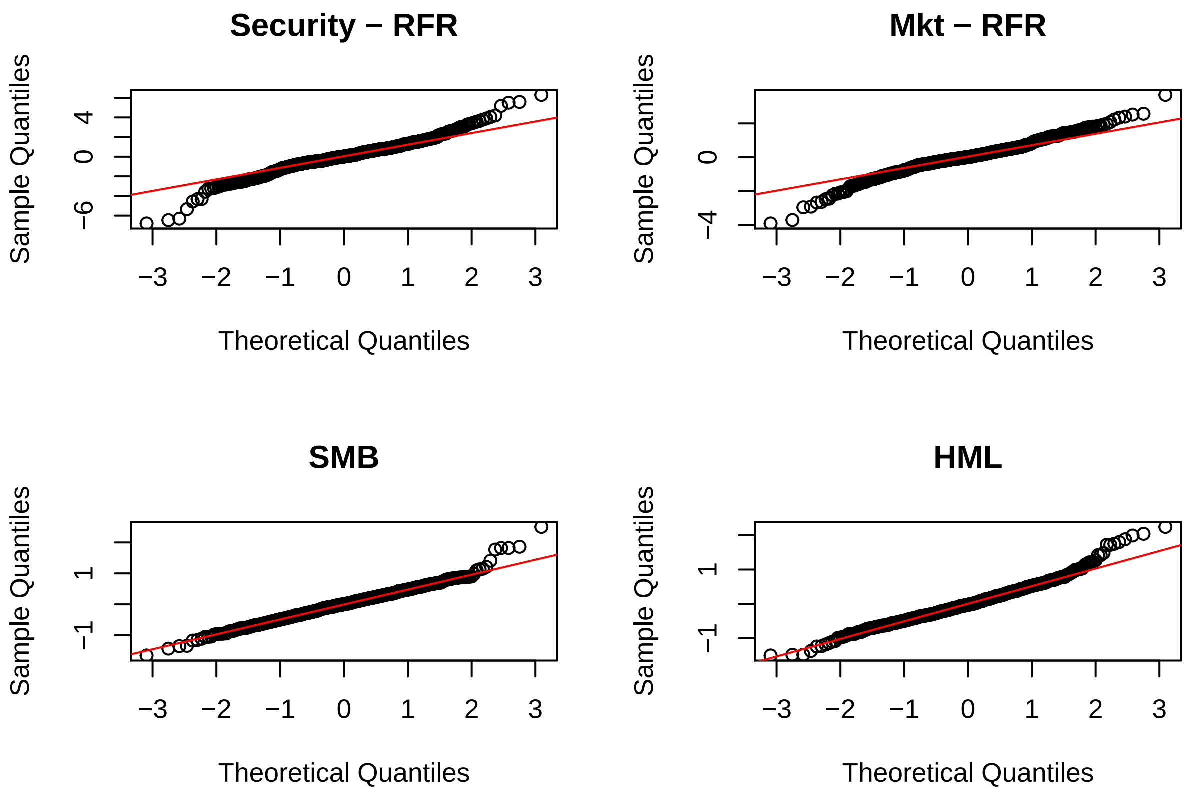
```
##                               (0.059)
## SMB                          -0.313***
##                               (0.107)
## HML                          -0.541***
##                               (0.099)
## Constant                      -0.006
##                               (0.054)
## ----------------------------------------------
## Observations                    506
## R2                             0.418
## Adjusted R2                    0.415
## Residual Std. Error      1.209 (df = 502)
## F Statistic          120.219*** (df = 3; 502)
## ==============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

From these results we can see that the security returns are significantly dependent on all three of the FF factors. The negative exposures to the SMB and HML factors are also worth considering further. The negative coefficient on the SMB factor, for example, tells us that this security likely belongs to a large company. The negative coefficient on the HML factor tells us that this security behaves more like a growth stock.

Quantile Regression Analysis

In the last section we used OLS in order to estimate our asset pricing models, but OLS assumes a normal distribution. Let's now use Q-Q plots to test if that was true:

```r
par( mfrow = c(2,2) )
# Security - RFR
qqnorm( Combined.Data$Stock.RFR , main = 'Security - RFR' )
qqline( Combined.Data$Stock.RFR , col = 2 )
# Security - RFR
qqnorm( Combined.Data$Mkt.RF , main = 'Mkt - RFR' )
qqline( Combined.Data$Mkt.RF , col = 2 )
# Security - RFR
qqnorm( Combined.Data$SMB , main = 'SMB' )
qqline( Combined.Data$SMB , col = 2 )
# Security - RFR
qqnorm( Combined.Data$HML , main = 'HML' )
qqline( Combined.Data$HML , col = 2 )
```



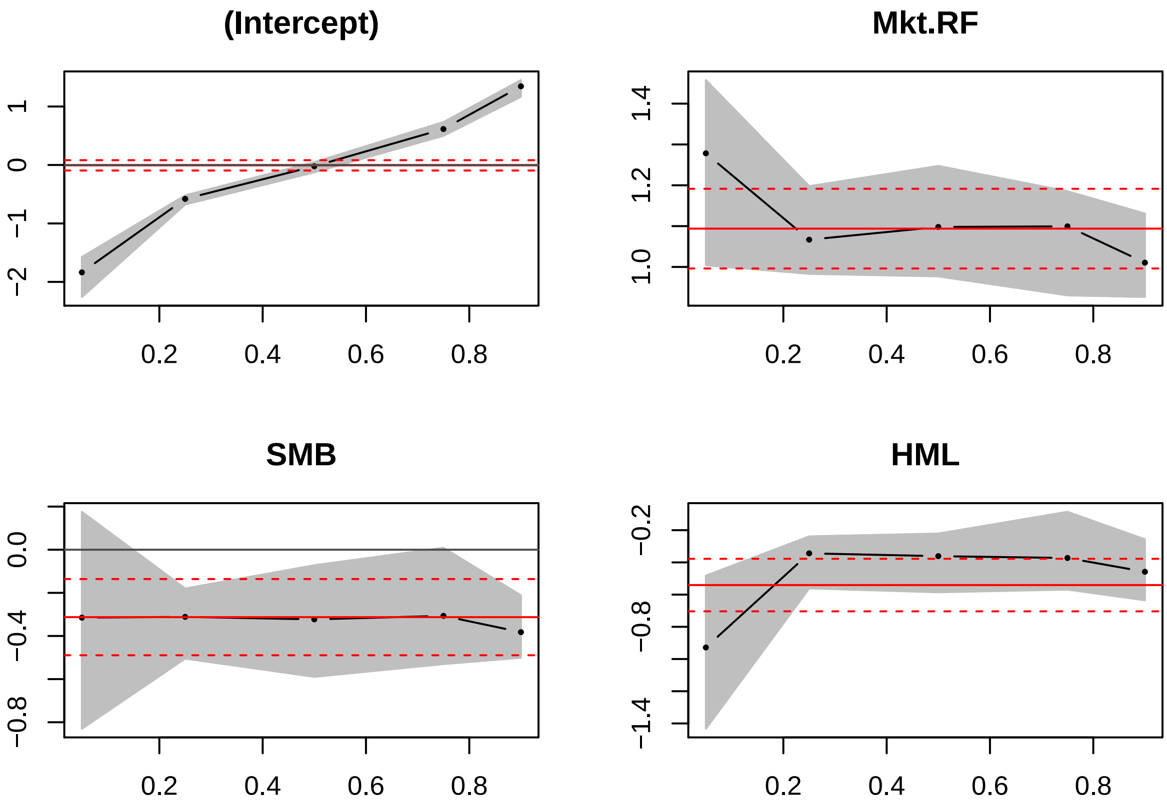
Any deviation of the circles from the red line indicates a lack of normality or some skew in the data. Therefore, OLS might not be the best way to measure these relationships. So, this part of the analysis looks at the relationship between the FF returns and the security returns over various quantiles.

```
# define percentiles that we want to examine
quantiles = c( .05 , .25 , .5 , .75 , .9 )

FF.Quantile.Regression =
  rq( Stock.RFR ~ Mkt.RF + SMB + HML
      , data = Combined.Data
      , tau = quantiles )
```

As before we could output the results by passing the regression object to the summary function, but in this case let's visualize the effects through plots:

```
plot( summary( FF.Quantile.Regression ) )
```



These summary plots show that the factor effects vary across the quantiles of the security return distribution. In other words, the return of this security have a different relationship with the various factor returns for lower and higher returns.

Summary

In this analysis we used OLS and quantile regressions in order to examine different asset pricing models, specifically the CAPM and the FF three factor model. The results of these asset pricing models are often used in cost of capital calculations, which are then often used to discount future cash flows and arrive at an asset valuation. Therefore, understanding these underlying models and principlies are important for asset valuation.