Master Thesis

Jonas Fremming & Mathias Vollan Pettersen

2022-07-01

Loading required packages:

library(tidyverse)

library(readxl) library(readr)

library(lubridate)

library(stargazer)

library(plm)

library(dplyr)

library(tidyr)

library(broom)

library(glue)

library(ggplot2)

library(car)

library(ggpubr)

library(quantmod)

library(tidyquant)

library(timeDate)

library(data.table)

library(timeSeries)

library(plotly)

library(riingo)

library(roll)

library(tictoc)

library(Hmisc)

```
require(reshape2)
require(foreign)
require(MASS)
require(lmtest)
require(DescTools)
require(mfx)
require(plyr)
require(MatchIt)
require(sandwich)
library(SciViews)
library(knitr)
library(rmarkdown)
library(car)
library(lmtest)
library(sandwich)
{r setup, include=TRUE}
knitr::opts_chunk$set(echo = TRUE)
```

Master Thesis Programming Section

Authors: Jonas Fremming & Mathias Vollan Pettersen

1.0 List of content

- 1 List of content
- 2 Introduction
- 3 Entire period regression illustrations for portfolios
  - 3.1 CAPM
  - 3.2 Fama-French three-factor models
  - 3.3 Fama-French five-factor models
  - 3.4 Fama-French five-factor models with momentum extension
  - 3.5 Comments

- 4 Entire period difference portfolio illustrations
    - 4.1 CAPM
    - 4.2 Fama-French three-factor models
    - 4.3 Fama-French five-factor models
    - 4.4 Fama-French five-factor models with momentum extension
- 5 Model testing
    - 5.1 Heteroskedasticity
    - 5.2 Autocorrelation
    - 5.3 Normality

2.0 Introduction

In this part of the master thesis, we go through the programming related to the data presented in the main paper. Our main data is mostly from Refinitiv Datastream, whereas we have used excel to format the data before importing it into R. Furthermore, for the descriptive data, we have only used excel. In this session, we present our programming for illustrational purposes. We have done considerable regressions for our results, as one can imagine, given the high amounts of data and results presented in our main thesis. Thus, for simplicity, we go through each new part of the code, meaning that we will not present the same regressions too often. All our regression outputs are the same as presented in our main thesis; thus, there is no point in illustrating the regression output for each regression we have done. We also present our model testing in the same way.

3 Entire period regression illustrations for portfolios

This chapter shows the regressions for various regression models. We only illustrate the total sin portfolio as we have done a high number of regressions the same way, suggesting that every method is the same. Thus, we only illustrate some of the regressions to simplify our thesis for the reader. However, all the outputs from the regressions are closely commented on and illustrated in the main thesis paper.

We use the value-weighted total sin portfolio from 2000-to 2021 for all regressions. We focus on the total sin portfolio and the difference portfolio in chapter 4. The remaining main regressions are the same, while the main thesis illustrates the output.

3.1 CAPM

We will now show the code and CAPM regression output for the total period total sin portfolio.

```
capm_sin_data <- read.table("CAPM_total_sin.csv",header=TRUE,sep=",")

MktRF <- capm_sin_data[,2]
```

```
RF <- capm_sin_data[,3]


sin_capm <- capm_sin_data[,4]


Excess_returns_sincapm <- sin_capm - RF



capm_sin_regression <- lm(Excess_returns_sincapm ~ MktRF)



print <- print(summary(capm_sin_regression))

##
## Call:
## lm(formula = Excess_returns_sincapm ~ MktRF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2940  -1.7473  -0.0145   1.7905   9.6691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.99503    0.17794   5.592 5.64e-08 ***
## MktRF        0.64623    0.03925  16.463  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.873 on 262 degrees of freedom
## Multiple R-squared:  0.5085, Adjusted R-squared:  0.5066
## F-statistic:   271 on 1 and 262 DF,  p-value: < 2.2e-16
```

3.2 Fama-French three-factor model

We continue the programming illustration with the three-factor regression model for the total sin portfolio.

```
ff3sin_data <- read.table("FF3_valueweighted_totalsin.csv",header=TRUE,sep=",
")

MktRF <- ff3sin_data[,2]


SMB <- ff3sin_data[,3]
```

```
HML <- ff3sin_data[,4]

RF <- ff3sin_data[,5]

Sin <- ff3sin_data[,6]

Excess_returns <- Sin - RF

ff3sin_regression <- lm(Excess_returns ~ MktRF + SMB + HML)

print <- print(summary(ff3sin_regression))

##
## Call:
## lm(formula = Excess_returns ~ MktRF + SMB + HML)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0963  -1.9106   0.1306   1.8490   8.1271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.91351    0.17315   5.276 2.78e-07 ***
## MktRF        0.66257    0.03824  17.327  < 2e-16 ***
## SMB         -0.04980    0.08907  -0.559    0.577
## HML          0.28294    0.06556   4.316 2.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.776 on 260 degrees of freedom
## Multiple R-squared:  0.5445, Adjusted R-squared:  0.5392
## F-statistic: 103.6 on 3 and 260 DF,  p-value: < 2.2e-16
```

3.3 Fama-French five-factor model

We continue with the five-factor regression model for the total sin portfolio.

```
ff5sin_data <- read.table("FF5_valueweighted_sin.csv", header = TRUE,sep=",")

MktRF <- ff5sin_data[,2]
SMB <- ff5sin_data[,3]
HML <- ff5sin_data[,4]
RMW <- ff5sin_data[,5]
CMA <- ff5sin_data[,6]
RF <- ff5sin_data[,7]
Sin <- ff5sin_data[,8]


Excess_returns_sin <- Sin - RF
```

```
ff5sin_regression <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)

print <- print(summary(ff5sin_regression))

##
## Call:
## lm(formula = Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7240 -1.7584 -0.1239  1.5948  7.7583
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.50139    0.17112   2.930  0.00369 **
## MktRF        0.79070    0.04225  18.713  < 2e-16 ***
## SMB          0.18471    0.08883   2.080  0.03856 *
## HML          0.09727    0.09277   1.048  0.29542
## RMW          0.74014    0.11321   6.538 3.32e-10 ***
## CMA          0.29549    0.14003   2.110  0.03580 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.562 on 258 degrees of freedom
## Multiple R-squared:  0.615,  Adjusted R-squared:  0.6075
## F-statistic: 82.41 on 5 and 258 DF,  p-value: < 2.2e-16
```

3.4 Fama-French five-factor model with a momentum extension

```
ff5momsin_data <- read.table("FF5MOM_valueweighted_sin.csv",header=TRUE,sep="
,")
MktRF <- ff5momsin_data[,2]
SMB <- ff5momsin_data[,3]
HML <- ff5momsin_data[,4]
RMW <- ff5momsin_data[,5]
CMA <- ff5momsin_data[,6]
RF <- ff5momsin_data[,7]
WML <- ff5momsin_data[,8]
Sin <- ff5momsin_data[,9]


Excess_returns_sin <- Sin - RF


ff5momsin_regression <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA
+ WML)


print <- print(summary(ff5momsin_regression))
```

```
##
## Call:
## lm(formula = Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA +
##     WML)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6777 -1.7860 -0.1756  1.6176  7.9505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.49683    0.17170   2.894  0.00414 **
## MktRF        0.79440    0.04316  18.404  < 2e-16 ***
## SMB          0.17135    0.09409   1.821  0.06976 .
## HML          0.11505    0.10146   1.134  0.25790
## RMW          0.73254    0.11472   6.386 7.94e-10 ***
## CMA          0.27911    0.14519   1.922  0.05566 .
## WML          0.02011    0.04611   0.436  0.66301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.566 on 257 degrees of freedom
## Multiple R-squared:  0.6152, Adjusted R-squared:  0.6063
## F-statistic: 68.49 on 6 and 257 DF,  p-value: < 2.2e-16
```

3.5 Comments

We have now illustrated all our main regressions for the total sin portfolio. We have done the same regressions as illustrated for various period-dived portfolios, equally-weighted portfolios, industry-divided portfolios, and crisis period-divided portfolios. The regressions are the same except for different data inputs, including various return data and Fama French factors.

4 Entire period difference portfolio illustrations

In this section, we provide the regressions for the traditional sin difference portfolio, i.e., the long sin portfolio and short comparable portfolio. The regressions are the same as in chapter 3, except for the "excess returns" for the sin portfolio being netted against the comparable portfolio in the csv file, i.e., before importing the data. The same regressions are done for various periods- and industry portfolios as well, whereas all the output is illustrated in the main thesis document.

4.1 CAPM

```
capm_tot_data <- read.table("CAPM_diff.csv",header=TRUE,sep=",")
MktRF <- capm_tot_data[,2]
RF <- capm_tot_data[,3]
totcapm <- capm_tot_data[,4]
```

```
Excess_returns_totcapm <- totcapm


capm_totcapm_regression <- lm(Excess_returns_totcapm ~ MktRF)


print <- print(summary(capm_totcapm_regression))

## 
## Call:
## lm(formula = Excess_returns_totcapm ~ MktRF)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.5612 -1.4154 -0.1624  1.4813  7.3287 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.37852    0.15781   2.399   0.0172 *  
## MktRF        0.14393    0.03481   4.134  4.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.548 on 262 degrees of freedom
## Multiple R-squared:  0.06125,    Adjusted R-squared:  0.05766 
## F-statistic: 17.09 on 1 and 262 DF,  p-value: 4.795e-05
```

4.2 Fama-French three-factor

```
ff3tot_data <- read.table("3factor_diff.csv",header=TRUE,sep=",")
MktRF <- ff3tot_data[,2]
SMB <- ff3tot_data[,3]
HML <- ff3tot_data[,4]
RF <- ff3tot_data[,5]
Tot <- ff3tot_data[,6]


Excess_returns <- Tot


ff3tot_regression <- lm(Excess_returns ~ MktRF + SMB + HML)


print <- print(summary(ff3tot_regression))

## 
## Call:
## lm(formula = Excess_returns ~ MktRF + SMB + HML)
## 
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -8.6183 -1.4582 -0.1797  1.3443  7.8214
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33297    0.15572   2.138  0.03343 *
## MktRF        0.13749    0.03439   3.998 8.34e-05 ***
## SMB          0.26129    0.08011   3.262  0.00126 **
## HML          0.12051    0.05896   2.044  0.04198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.497 on 260 degrees of freedom
## Multiple R-squared:  0.1052, Adjusted R-squared:  0.09491
## F-statistic: 10.19 on 3 and 260 DF,  p-value: 2.277e-06
```

4.3 Fama-French five-factor

```
ff5tot_data <- read.table("5factor_diff.csv",header=TRUE,sep=",")
MktRF <- ff5tot_data[,2]
SMB <- ff5tot_data[,3]
HML <- ff5tot_data[,4]
RMW <- ff5tot_data[,5]
CMA <- ff5tot_data[,6]
RF <- ff5tot_data[,7]
Totff5 <- ff5tot_data[,8]


Excess_returns_totff5 <- Totff5


ff5tot_regression <- lm(Excess_returns_totff5 ~ MktRF + SMB + HML + RMW + CMA
)


print <- print(summary(ff5tot_regression))

##
## Call:
## lm(formula = Excess_returns_totff5 ~ MktRF + SMB + HML + RMW +
##     CMA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.4092 -1.4676 -0.1105  1.2809  7.6877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.24248    0.16456   1.474  0.14182
## MktRF        0.14426    0.04065   3.549  0.00046 ***
## SMB          0.27336    0.08625   3.169  0.00171 **
```

```
## HML          0.18711    0.08929   2.095  0.03711 *
## RMW          0.21321    0.10884   1.959  0.05120 .
## CMA         -0.22232    0.13455  -1.652  0.09970 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.458 on 257 degrees of freedom
## Multiple R-squared:  0.1285, Adjusted R-squared:  0.1116
## F-statistic: 7.581 on 5 and 257 DF,  p-value: 1.162e-06
```

4.4 Fama-French five-factor with momentum extension

```
ff5momtot_data <- read.table("5factor_diff_mom.csv",header=TRUE,sep=",")
MktRF <- ff5momtot_data[,2]
SMB <- ff5momtot_data[,3]
HML <- ff5momtot_data[,4]
RMW <- ff5momtot_data[,5]
CMA <- ff5momtot_data[,6]
RF <- ff5momtot_data[,7]
WML <- ff5momtot_data[,8]
Totff5mom <- ff5momtot_data[,9]




Excess_returns_momtot <- Totff5mom


ff5momtot_regression <- lm(Excess_returns_momtot ~ MktRF + SMB + HML + RMW +
CMA + WML)


print <- print(summary(ff5momtot_regression))

##
## Call:
## lm(formula = Excess_returns_momtot ~ MktRF + SMB + HML + RMW +
##     CMA + WML)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3362 -1.5508 -0.2018  1.3938  8.5757
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.25143    0.16511   1.523 0.129034
## MktRF        0.15189    0.04151   3.659 0.000307 ***
## SMB          0.25071    0.09048   2.771 0.005999 **
## HML          0.24173    0.09757   2.478 0.013872 *
## RMW          0.16537    0.11031   1.499 0.135060
## CMA         -0.26986    0.13961  -1.933 0.054336 .
```

```
## WML          0.08118    0.04433   1.831 0.068232 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.468 on 257 degrees of freedom
## Multiple R-squared:  0.1361, Adjusted R-squared:  0.1159
## F-statistic: 6.748 on 6 and 257 DF,  p-value: 1.182e-06
```

5 Model testing

In this chapter, we test various assumptions of the classical linear regression model. We only need to test for homoscedasticity, autocorrelation, and normality, as assumption one regarding standard errors with zero mean holds because of the constant term, while assumption four always holds for the Fama French regression models, according to Cahart (1997), as explained in chapter 3.2.5 in the thesis document.

We illustrate our code and output for the total sin portfolio. The same code is used for the tests for the other portfolios and Fama French factors as well.

5.1 Heteroscedasticity

We use a Breusch-Pagan test to test for heteroscedasticity.

5.1.1 Fama-French three-factor model for total sin portfolio

```
ff3sin_data <- read.table("FF3_valueweighted_totalsin.csv",header=TRUE,sep=",
")
MktRF <- ff3sin_data[,2]
SMB <- ff3sin_data[,3]
HML <- ff3sin_data[,4]
RF <- ff3sin_data[,5]
Sin <- ff3sin_data[,6]

Excess_returns_sin <- Sin-RF

ff3sin_regressiont <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

bptest(ff3sin_regressiont)

##
##  studentized Breusch-Pagan test
##
```

```
## data:  ff3sin_regressiont
## BP = 7.2838, df = 5, p-value = 0.2004
```

5.1.2 Fama-French five-factor model for total sin portfolio

```
ff5sin_data <- read.table("FF5_valueweighted_sin.csv",header=TRUE,sep=",")
MktRF <- ff5sin_data[,2]
SMB <- ff5sin_data[,3]
HML <- ff5sin_data[,4]
RMW <- ff5sin_data[,5]
CMA <- ff5sin_data[,6]
RF <- ff5sin_data[,7]
Sin <- ff5sin_data[,8]

Excess_returns_sin <- Sin-RF



ff5sin_regressiont <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)

bptest(ff5sin_regressiont)

##
##  studentized Breusch-Pagan test
##
## data:  ff5sin_regressiont
## BP = 6.9481, df = 5, p-value = 0.2245
```

5.2 Autocorrelation

We illustrate our tests for autocorrelation through the test on the FF3 and FF5 regressions on the total sin portfolio.

5.2.1 Fama-French three-factor model for total sin portfolio

```
ff3sin_data <- read.table("FF3_valueweighted_totalsin.csv",header=TRUE,sep=",
")
MktRF <- ff3sin_data[,2]
SMB <- ff3sin_data[,3]
HML <- ff3sin_data[,4]
RF <- ff3sin_data[,5]
Sin <- ff3sin_data[,6]

#Calculating excess returns for sin returns
Excess_returns_sin <- Sin-RF

#Running FF regression on total sin portfolio
ff3sin_regression <- lm(Excess_returns_sin ~ MktRF + SMB + HML)

T <- nrow(ff3sin_regression$model)
print(T)

## [1] 264
```

```r
ff3sin_data$u <-summary(ff3sin_regression)$residuals

## Breush-Gofrey test with lags

#lagged estimated residuals

ff3sin_data$u1 <- c(0, ff3sin_data$u[1:T-1])
ff3sin_data$u2 <- c(0, ff3sin_data$u1[1:T-1])
ff3sin_data$u3 <- c(0, ff3sin_data$u2[1:T-1])
ff3sin_data$u4 <- c(0, ff3sin_data$u3[1:T-1])
ff3sin_data$u5 <- c(0, ff3sin_data$u4[1:T-1])
ff3sin_data$u6 <- c(0, ff3sin_data$u5[1:T-1])
ff3sin_data$u7 <- c(0, ff3sin_data$u6[1:T-1])
ff3sin_data$u8 <- c(0, ff3sin_data$u7[1:T-1])
ff3sin_data$u9 <- c(0, ff3sin_data$u8[1:T-1])
ff3sin_data$u10 <- c(0, ff3sin_data$u9[1:T-1])

# auxiliary regression

aux_regr <- lm(u ~ MktRF + SMB + HML + u1 + u2 + u3 + u4 + u5 + u6 + u7 + u8
+ u9 + u10, data=ff3sin_data)
summary(aux_regr)

##
## Call:
## lm(formula = u ~ MktRF + SMB + HML + u1 + u2 + u3 + u4 + u5 +
##     u6 + u7 + u8 + u9 + u10, data = ff3sin_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7795 -1.8060  0.1058  1.8423  7.9645
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0011989  0.1730836   0.007   0.9945
## MktRF        0.0006552  0.0390852   0.017   0.9866
## SMB         -0.0070792  0.0903545  -0.078   0.9376
## HML         -0.0247597  0.0705497  -0.351   0.7259
## u1          -0.0322666  0.0637616  -0.506   0.6133
## u2          -0.0410841  0.0634721  -0.647   0.5180
## u3           0.0296339  0.0645894   0.459   0.6468
## u4           0.0404331  0.0636101   0.636   0.5256
## u5           0.1153478  0.0639375   1.804   0.0724 .
## u6           0.0421607  0.0648240   0.650   0.5160
## u7          -0.0868950  0.0640885  -1.356   0.1764
## u8           0.0983799  0.0641986   1.532   0.1267
## u9           0.0194020  0.0642907   0.302   0.7631
## u10         -0.0085699  0.0651901  -0.131   0.8955
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.775 on 250 degrees of freedom
## Multiple R-squared:  0.03948,    Adjusted R-squared:  -0.01046
## F-statistic: 0.7905 on 13 and 250 DF,  p-value: 0.67

R2_aux <- summary(aux_regr)$r.squared
print(R2_aux)

## [1] 0.03948283

# testing

alpha = 0.05

r <- 10

hyp1.test_stat <- R2_aux * (T-r)
print(hyp1.test_stat)

## [1] 10.02864

hyp1.crit_val <- qchisq(1-alpha, r)
print(hyp1.crit_val)

## [1] 18.30704

if (hyp1.test_stat>hyp1.crit_val) {hyp1.reject <- "reject H0"} else {hyp1.rej
ect <- "do not reject H0"}

print(hyp1.reject)

## [1] "do not reject H0"

hyp1.p <- 1-pchisq(hyp1.test_stat,r)
print(hyp1.p)

## [1] 0.4379843

print(alpha)

## [1] 0.05
```

5.2.2 Fama-French five-factor model for total sin portfolio

```
ff5sin_data <- read.table("FF5_valueweighted_sin.csv",header=TRUE,sep=",")
MktRF <- ff5sin_data[,2]
SMB <- ff5sin_data[,3]
HML <- ff5sin_data[,4]
RMW <- ff5sin_data[,5]
CMA <- ff5sin_data[,6]
RF <- ff5sin_data[,7]
Sin <- ff5sin_data[,8]
```

```r
#Calculating excess returns for sin returns
Excess_returns_sin <- Sin-RF

#Running FF regression on total sin portfolio
ff5sin_regression <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)

T <- nrow(ff5sin_regression$model)
print(T)

## [1] 264

ff5sin_data$u <-summary(ff5sin_regression)$residuals

# Breush-Gofrey test with lags

#lagged estimated residuals

ff5sin_data$u1 <- c(0, ff5sin_data$u[1:T-1])
ff5sin_data$u2 <- c(0, ff5sin_data$u1[1:T-1])
ff5sin_data$u3 <- c(0, ff5sin_data$u2[1:T-1])
ff5sin_data$u4 <- c(0, ff5sin_data$u3[1:T-1])
ff5sin_data$u5 <- c(0, ff5sin_data$u4[1:T-1])
ff5sin_data$u6 <- c(0, ff5sin_data$u5[1:T-1])
ff5sin_data$u7 <- c(0, ff5sin_data$u6[1:T-1])
ff5sin_data$u8 <- c(0, ff5sin_data$u7[1:T-1])
ff5sin_data$u9 <- c(0, ff5sin_data$u8[1:T-1])
ff5sin_data$u10 <- c(0, ff5sin_data$u9[1:T-1])

# auxiliary regression

aux_regr <- lm(u ~ MktRF + SMB + HML + RMW + CMA + u1 + u2 + u3 + u4 + u5 + u
6 + u7 + u8 + u9 + u10, data=ff5sin_data)
summary(aux_regr)

##
## Call:
## lm(formula = u ~ MktRF + SMB + HML + RMW + CMA + u1 + u2 + u3 +
##     u4 + u5 + u6 + u7 + u8 + u9 + u10, data = ff5sin_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6895 -1.5734 -0.0204  1.6360  7.3695
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.013034   0.170065   0.077   0.9390
## MktRF        0.003314   0.042806   0.077   0.9384
## SMB         -0.011665   0.089702  -0.130   0.8966
## HML         -0.071248   0.097737  -0.729   0.4667
## RMW         -0.052420   0.114684  -0.457   0.6480
```

```
## CMA          0.061922   0.142358   0.435   0.6640
## u1          -0.094198   0.064141  -1.469   0.1432
## u2          -0.049534   0.064117  -0.773   0.4405
## u3           0.065073   0.065751   0.990   0.3233
## u4           0.110260   0.064515   1.709   0.0887 .
## u5           0.152625   0.065747   2.321   0.0211 *
## u6           0.053186   0.067035   0.793   0.4283
## u7          -0.085050   0.066140  -1.286   0.1997
## u8           0.021181   0.066220   0.320   0.7494
## u9           0.050740   0.066234   0.766   0.4444
## u10         -0.007209   0.066598  -0.108   0.9139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.541 on 248 degrees of freedom
## Multiple R-squared:  0.05444,    Adjusted R-squared:  -0.00275
## F-statistic: 0.9519 on 15 and 248 DF,  p-value: 0.5072
```

```
R2_aux <- summary(aux_regr)$r.squared
print(R2_aux)
```

```
## [1] 0.05444127
```

```
# testing

alpha = 0.05

r <- 10

hyp1.test_stat <- R2_aux * (T-r)
print(hyp1.test_stat)
```

```
## [1] 13.82808
```

```
hyp1.crit_val <- qchisq(1-alpha, r)
print(hyp1.crit_val)
```

```
## [1] 18.30704
```

```
if (hyp1.test_stat>hyp1.crit_val) {hyp1.reject <- "reject H0"} else {hyp1.rej
ect <- "do not reject H0"}

print(hyp1.reject)
```

```
## [1] "do not reject H0"
```

```
hyp1.p <- 1-pchisq(hyp1.test_stat,r)
print(hyp1.p)
```

```
## [1] 0.1809787
```

```
print(alpha)
```

```
## [1] 0.05
```

5.3 Normality

We use the total sin portfolio as illustration, as previously explained.

```
ff5sin_data <- read.table("FF5_valueweighted_sin.csv",header=TRUE,sep=",")
MktRF <- ff5sin_data[,2]
SMB <- ff5sin_data[,3]
HML <- ff5sin_data[,4]
RMW <- ff5sin_data[,5]
CMA <- ff5sin_data[,6]
RF <- ff5sin_data[,7]
Sin <- ff5sin_data[,8]

#Calculating excess returns for sin returns
Excess_returns_sin <- Sin-RF

#Running FF regression on total sin portfolio
ff5sin_regression <- lm(Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)

T <- nrow(ff5sin_regression$model)
print(T)

## [1] 264

ff5sin_data$u <-summary(ff5sin_regression)$residuals

summary(ff5sin_regression)

##
## Call:
## lm(formula = Excess_returns_sin ~ MktRF + SMB + HML + RMW + CMA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7240 -1.7584 -0.1239  1.5948  7.7583
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.50139    0.17112   2.930  0.00369 **
## MktRF        0.79070    0.04225  18.713  < 2e-16 ***
## SMB          0.18471    0.08883   2.080  0.03856 *
## HML          0.09727    0.09277   1.048  0.29542
## RMW          0.74014    0.11321   6.538 3.32e-10 ***
## CMA          0.29549    0.14003   2.110  0.03580 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.562 on 258 degrees of freedom
```

```
## Multiple R-squared:  0.615,  Adjusted R-squared:  0.6075
## F-statistic: 82.41 on 5 and 258 DF,  p-value: < 2.2e-16

## Jarque-Bera test

alpha<-0.05

b1 <- mean(ff5sin_data$u^3)/mean(ff5sin_data$u^2)^(3/2)
print(b1)

## [1] 0.03293987

b2 <- mean(ff5sin_data$u^4)/mean(ff5sin_data$u^2)^2
print(b2)

## [1] 3.36515

hyp1.test_stat <- T*(b1^2/6+(b2-3)^2)^2
print(hyp1.test_stat)

## [1] 4.706135

if (hyp1.test_stat>hyp1.crit_val) {hyp1.reject <- "Reject H0"} else {hyp1.rej
ect <- "Do not reject H0"}
print(hyp1.reject)

## [1] "Do not reject H0"

hyp1.p <- 1-pchisq(hyp1.test_stat,2)
print(hyp1.p)

## [1] 0.09507705

print(alpha)

## [1] 0.05

if (hyp1.p<alpha) {hyp1.reject <- "reject Ho"} else {hyp1.reject <- "do not r
eject H0"}
print(hyp1.reject)

## [1] "do not reject H0"
```