HW9

Market Model

4/4/2021

pacman::p\_load(pacman, tidyverse, tseries, psych, knitr, here)  
knitr::opts\_chunk$set(message = FALSE, tidy = TRUE)

## Question 8.3

x = get.hist.quote(instrument = "^GSPC",  
 start = "2010-12-01",  
 end = "2015-12-31",  
 quote = "AdjClose",  
 compression = "m")

## time series ends 2015-12-01

## time series ends 2015-12-01  
sp500 = as.vector(x)  
n = length(sp500)  
  
# Net returns  
sp500\_ret = (sp500[-1] - sp500[-n])/sp500[-n]  
  
  
  
data1 = read.table(file.choose(),header = T,sep = ",")  
data1

## TimePeriod Yield  
## 1 2011-01 0.15  
## 2 2011-02 0.13  
## 3 2011-03 0.10  
## 4 2011-04 0.06  
## 5 2011-05 0.04  
## 6 2011-06 0.04  
## 7 2011-07 0.04  
## 8 2011-08 0.02  
## 9 2011-09 0.01  
## 10 2011-10 0.02  
## 11 2011-11 0.01  
## 12 2011-12 0.01  
## 13 2012-01 0.03  
## 14 2012-02 0.09  
## 15 2012-03 0.08  
## 16 2012-04 0.08  
## 17 2012-05 0.09  
## 18 2012-06 0.09  
## 19 2012-07 0.10  
## 20 2012-08 0.10  
## 21 2012-09 0.11  
## 22 2012-10 0.10  
## 23 2012-11 0.09  
## 24 2012-12 0.07  
## 25 2013-01 0.07  
## 26 2013-02 0.10  
## 27 2013-03 0.09  
## 28 2013-04 0.06  
## 29 2013-05 0.04  
## 30 2013-06 0.05  
## 31 2013-07 0.04  
## 32 2013-08 0.04  
## 33 2013-09 0.02  
## 34 2013-10 0.05  
## 35 2013-11 0.07  
## 36 2013-12 0.07  
## 37 2014-01 0.04  
## 38 2014-02 0.05  
## 39 2014-03 0.05  
## 40 2014-04 0.03  
## 41 2014-05 0.03  
## 42 2014-06 0.04  
## 43 2014-07 0.03  
## 44 2014-08 0.03  
## 45 2014-09 0.02  
## 46 2014-10 0.02  
## 47 2014-11 0.02  
## 48 2014-12 0.03  
## 49 2015-01 0.03  
## 50 2015-02 0.02  
## 51 2015-03 0.03  
## 52 2015-04 0.02  
## 53 2015-05 0.02  
## 54 2015-06 0.02  
## 55 2015-07 0.03  
## 56 2015-08 0.07  
## 57 2015-09 0.02  
## 58 2015-10 0.02  
## 59 2015-11 0.11  
## 60 2015-12 0.22

rff1 = data1$Yield  
table(is.na(rff1))

##   
## FALSE   
## 60

rff1

## [1] 0.15 0.13 0.10 0.06 0.04 0.04 0.04 0.02 0.01 0.02 0.01 0.01 0.03 0.09 0.08  
## [16] 0.08 0.09 0.09 0.10 0.10 0.11 0.10 0.09 0.07 0.07 0.10 0.09 0.06 0.04 0.05  
## [31] 0.04 0.04 0.02 0.05 0.07 0.07 0.04 0.05 0.05 0.03 0.03 0.04 0.03 0.03 0.02  
## [46] 0.02 0.02 0.03 0.03 0.02 0.03 0.02 0.02 0.02 0.03 0.07 0.02 0.02 0.11 0.22

rfree1 = (1 + rff1/100)^(1/12) - 1  
head(rfree1)

## [1] 1.249141e-04 1.082688e-04 8.329516e-05 4.998626e-05 3.332722e-05  
## [6] 3.332722e-05

d = tibble(rfree1, sp500\_ret)  
d = d %>%  
 mutate(SP500 = sp500\_ret - rfree1)  
d

## # A tibble: 60 x 3  
## rfree1 sp500\_ret SP500  
## <dbl> <dbl> <dbl>  
## 1 0.000125 0.0226 0.0225   
## 2 0.000108 0.0320 0.0318   
## 3 0.0000833 -0.00105 -0.00113  
## 4 0.0000500 0.0285 0.0284   
## 5 0.0000333 -0.0135 -0.0135   
## 6 0.0000333 -0.0183 -0.0183   
## 7 0.0000333 -0.0215 -0.0215   
## 8 0.0000167 -0.0568 -0.0568   
## 9 0.00000833 -0.0718 -0.0718   
## 10 0.0000167 0.108 0.108   
## # ... with 50 more rows

indices = d %>%  
 transmute(SP500)  
indices

## # A tibble: 60 x 1  
## SP500  
## <dbl>  
## 1 0.0225   
## 2 0.0318   
## 3 -0.00113  
## 4 0.0284   
## 5 -0.0135   
## 6 -0.0183   
## 7 -0.0215   
## 8 -0.0568   
## 9 -0.0718   
## 10 0.108   
## # ... with 50 more rows

#PZZA  
x1 = get.hist.quote(instrument = "PZZA", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

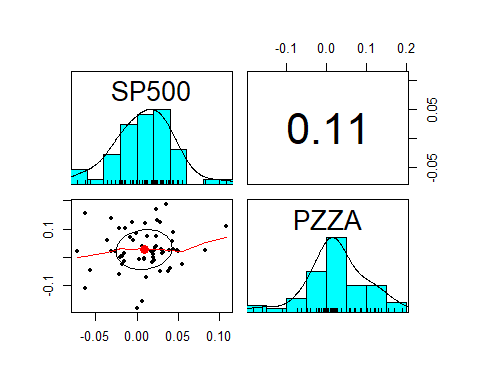
## time series ends 2015-12-01  
pzza\_m = as.vector(x1)  
n = length(pzza\_m)  
  
pzza\_m\_ret = (pzza\_m[-1] - pzza\_m[-n])/pzza\_m[-n]  
  
d1 = tibble(pzza\_m\_ret, rfree1)  
d1 = d1 %>% mutate(PZZA = pzza\_m\_ret - rfree1)  
d1

## # A tibble: 60 x 3  
## pzza\_m\_ret rfree1 PZZA  
## <dbl> <dbl> <dbl>  
## 1 0.0361 0.000125 0.0360   
## 2 0.0167 0.000108 0.0166   
## 3 0.0853 0.0000833 0.0852   
## 4 -0.0508 0.0000500 -0.0509   
## 5 0.102 0.0000333 0.102   
## 6 0.00392 0.0000333 0.00389  
## 7 -0.0616 0.0000333 -0.0617   
## 8 -0.0468 0.0000167 -0.0468   
## 9 0.0218 0.00000833 0.0218   
## 10 0.111 0.0000167 0.111   
## # ... with 50 more rows

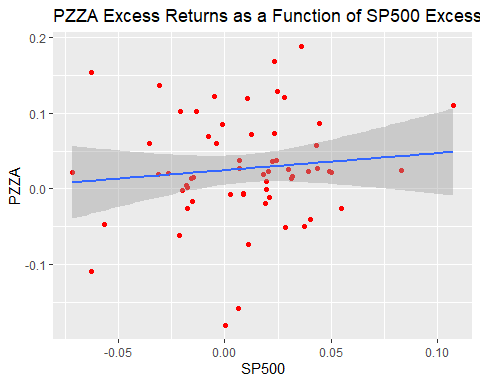
d6 = tibble(SP500 = indices$SP500, PZZA = d1$PZZA)  
d6

## # A tibble: 60 x 2  
## SP500 PZZA  
## <dbl> <dbl>  
## 1 0.0225 0.0360   
## 2 0.0318 0.0166   
## 3 -0.00113 0.0852   
## 4 0.0284 -0.0509   
## 5 -0.0135 0.102   
## 6 -0.0183 0.00389  
## 7 -0.0215 -0.0617   
## 8 -0.0568 -0.0468   
## 9 -0.0718 0.0218   
## 10 0.108 0.111   
## # ... with 50 more rows

pairs.panels(d6)



d6 %>%  
 ggplot(aes(x = SP500, y = PZZA )) +  
 geom\_point(color="red") +  
 geom\_smooth(method="lm") +  
 ggtitle("PZZA Excess Returns as a Function of SP500 Excess Returns")



lm(PZZA ~ SP500, data=d6) #Estimate the parameters

##   
## Call:  
## lm(formula = PZZA ~ SP500, data = d6)  
##   
## Coefficients:  
## (Intercept) SP500   
## 0.02451 0.22442

#Save the result and summarize them  
pzza.mm = lm(PZZA ~ SP500, data=d6)  
summary(pzza.mm)

##   
## Call:  
## lm(formula = PZZA ~ SP500, data = d6)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.205542 -0.032781 -0.007223 0.043913 0.155971   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.024511 0.009555 2.565 0.0129 \*  
## SP500 0.224422 0.276283 0.812 0.4199   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07167 on 58 degrees of freedom  
## Multiple R-squared: 0.01125, Adjusted R-squared: -0.005799   
## F-statistic: 0.6598 on 1 and 58 DF, p-value: 0.4199

#Coefficients summary  
summary(pzza.mm)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.0245114 0.009554868 2.5653308 0.01291594  
## SP500 0.2244225 0.276282791 0.8122926 0.41994530

#Isolate beta  
summary(pzza.mm)$coefficients[2,1]

## [1] 0.2244225

#Isolate sigma  
summary(pzza.mm)$sigma

## [1] 0.07166625

#Isolate r.squared  
summary(pzza.mm)$r.squared

## [1] 0.01124823

confint(pzza.mm) # 95% interval

## 2.5 % 97.5 %  
## (Intercept) 0.005385251 0.04363754  
## SP500 -0.328617630 0.77746256

confint(pzza.mm,level = 0.9) # 90% interval

## 5 % 95 %  
## (Intercept) 0.008539932 0.04048286  
## SP500 -0.237398799 0.68624373

#BBBY  
x2 = get.hist.quote(instrument = "BBBY", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

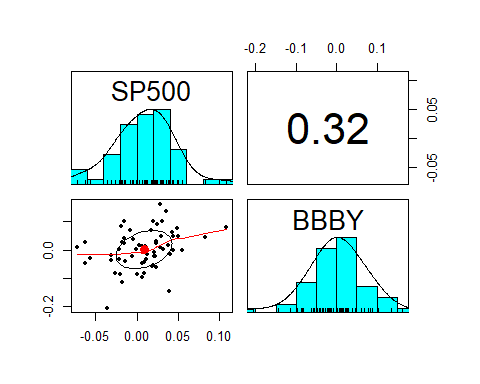
## time series ends 2015-12-01  
bbby\_m = as.vector(x2)  
n = length(bbby\_m)  
  
bbby\_m\_ret = (bbby\_m[-1] - bbby\_m[-n])/bbby\_m[-n]  
  
d2 = tibble(bbby\_m\_ret, rfree1)  
d2 = d2 %>% mutate(BBBY = bbby\_m\_ret - rfree1)  
d2

## # A tibble: 60 x 3  
## bbby\_m\_ret rfree1 BBBY  
## <dbl> <dbl> <dbl>  
## 1 -0.0234 0.000125 -0.0235   
## 2 0.00313 0.000108 0.00302  
## 3 0.00249 0.0000833 0.00241  
## 4 0.163 0.0000500 0.163   
## 5 -0.0399 0.0000333 -0.0399   
## 6 0.0831 0.0000333 0.0831   
## 7 0.00206 0.0000333 0.00202  
## 8 -0.0279 0.0000167 -0.0279   
## 9 0.00791 0.00000833 0.00791  
## 10 0.0790 0.0000167 0.0790   
## # ... with 50 more rows

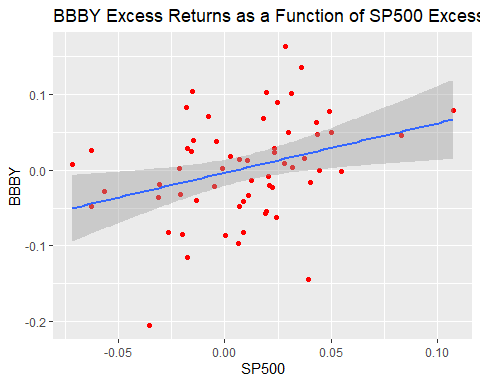
d7 = tibble(SP500 = indices$SP500, BBBY = d2$BBBY)  
d7

## # A tibble: 60 x 2  
## SP500 BBBY  
## <dbl> <dbl>  
## 1 0.0225 -0.0235   
## 2 0.0318 0.00302  
## 3 -0.00113 0.00241  
## 4 0.0284 0.163   
## 5 -0.0135 -0.0399   
## 6 -0.0183 0.0831   
## 7 -0.0215 0.00202  
## 8 -0.0568 -0.0279   
## 9 -0.0718 0.00791  
## 10 0.108 0.0790   
## # ... with 50 more rows

pairs.panels(d7)



d7 %>%  
 ggplot(aes(x = SP500, y = BBBY )) +  
 geom\_point(color="red") +  
 geom\_smooth(method="lm") +  
 ggtitle("BBBY Excess Returns as a Function of SP500 Excess Returns")



lm(BBBY ~ SP500, data=d7)

##   
## Call:  
## lm(formula = BBBY ~ SP500, data = d7)  
##   
## Coefficients:  
## (Intercept) SP500   
## -0.003675 0.650968

#Save the result and summarize them  
bbby.mm = lm(BBBY ~ SP500, data=d7)  
summary(bbby.mm)

##   
## Call:  
## lm(formula = BBBY ~ SP500, data = d7)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.178030 -0.035074 0.001028 0.039497 0.147943   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.003675 0.008625 -0.426 0.6716   
## SP500 0.650968 0.249383 2.610 0.0115 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06469 on 58 degrees of freedom  
## Multiple R-squared: 0.1051, Adjusted R-squared: 0.0897   
## F-statistic: 6.814 on 1 and 58 DF, p-value: 0.01149

#Coefficients summary  
summary(bbby.mm)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -0.00367542 0.008624591 -0.4261558 0.67157174  
## SP500 0.65096765 0.249383459 2.6103081 0.01149257

#Isolate beta  
summary(bbby.mm)$coefficients[2,1]

## [1] 0.6509677

#Isolate sigma  
summary(bbby.mm)$sigma

## [1] 0.06468871

#Isolate r.squared  
summary(bbby.mm)$r.squared

## [1] 0.1051276

confint(bbby.mm) # 95% interval

## 2.5 % 97.5 %  
## (Intercept) -0.02093941 0.01358857  
## SP500 0.15177242 1.15016288

confint(bbby.mm,level = 0.9) # 90% interval

## 5 % 95 %  
## (Intercept) -0.01809188 0.01074104  
## SP500 0.23411005 1.06782526

#NFLX  
  
x3 = get.hist.quote(instrument = "NFLX", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

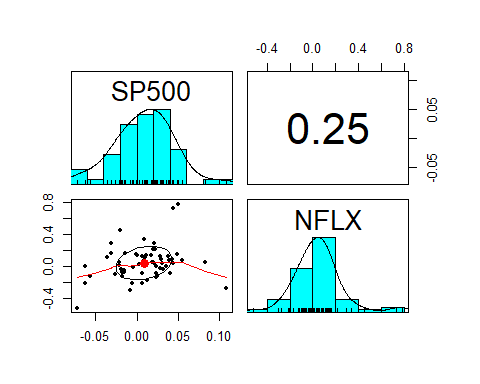
## time series ends 2015-12-01  
nflx\_m = as.vector(x3)  
n = length(nflx\_m)  
  
nflx\_m\_ret = (nflx\_m[-1] - nflx\_m[-n])/nflx\_m[-n]  
  
d3 = tibble(nflx\_m\_ret, rfree1)  
d3 = d3 %>% mutate(NFLX = nflx\_m\_ret - rfree1)  
d3

## # A tibble: 60 x 3  
## nflx\_m\_ret rfree1 NFLX  
## <dbl> <dbl> <dbl>  
## 1 0.218 0.000125 0.218   
## 2 -0.0346 0.000108 -0.0347  
## 3 0.151 0.0000833 0.150   
## 4 -0.0215 0.0000500 -0.0215  
## 5 0.164 0.0000333 0.164   
## 6 -0.0299 0.0000333 -0.0300  
## 7 0.0126 0.0000333 0.0125  
## 8 -0.116 0.0000167 -0.116   
## 9 -0.518 0.00000833 -0.518   
## 10 -0.275 0.0000167 -0.275   
## # ... with 50 more rows

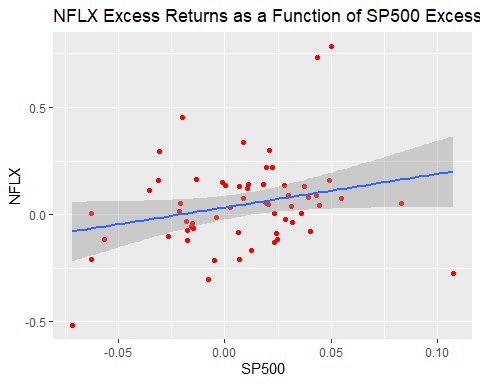
d8 = tibble(SP500 = indices$SP500, NFLX = d3$NFLX)  
d8

## # A tibble: 60 x 2  
## SP500 NFLX  
## <dbl> <dbl>  
## 1 0.0225 0.218   
## 2 0.0318 -0.0347  
## 3 -0.00113 0.150   
## 4 0.0284 -0.0215  
## 5 -0.0135 0.164   
## 6 -0.0183 -0.0300  
## 7 -0.0215 0.0125  
## 8 -0.0568 -0.116   
## 9 -0.0718 -0.518   
## 10 0.108 -0.275   
## # ... with 50 more rows

pairs.panels(d8)



d8 %>%  
 ggplot(aes(x = SP500, y = NFLX )) +  
 geom\_point(color="red") +  
 geom\_smooth(method="lm") +  
 ggtitle("NFLX Excess Returns as a Function of SP500 Excess Returns")



lm(NFLX ~ SP500, data=d8)

##   
## Call:  
## lm(formula = NFLX ~ SP500, data = d8)  
##   
## Coefficients:  
## (Intercept) SP500   
## 0.03225 1.56629

#Save the result and summarize them  
nflx.mm = lm(NFLX ~ SP500, data=d8)  
summary(nflx.mm)

##   
## Call:  
## lm(formula = NFLX ~ SP500, data = d8)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.47632 -0.10191 -0.01609 0.08292 0.67344   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.03225 0.02740 1.177 0.2441   
## SP500 1.56629 0.79236 1.977 0.0528 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2055 on 58 degrees of freedom  
## Multiple R-squared: 0.06312, Adjusted R-squared: 0.04697   
## F-statistic: 3.908 on 1 and 58 DF, p-value: 0.05283

#Coefficients summary  
summary(nflx.mm)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.03224645 0.02740266 1.176764 0.2440968  
## SP500 1.56629425 0.79235866 1.976749 0.0528310

#Isolate beta  
summary(nflx.mm)$coefficients[2,1]

## [1] 1.566294

#Isolate sigma  
summary(nflx.mm)$sigma

## [1] 0.2055335

#Isolate r.squared  
summary(nflx.mm)$r.squared

## [1] 0.06311892

confint(nflx.mm) # 95% interval

## 2.5 % 97.5 %  
## (Intercept) -0.02260593 0.08709883  
## SP500 -0.01978394 3.15237243

confint(nflx.mm,level = 0.9) # 90% interval

## 5 % 95 %  
## (Intercept) -0.01355854 0.07805143  
## SP500 0.24182494 2.89076356

#T  
  
X4 = get.hist.quote(instrument = "T", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

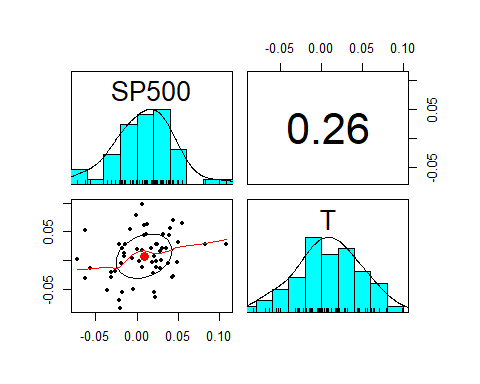
## time series ends 2015-12-01  
t\_m = as.vector(X4)  
n = length(t\_m)  
  
t\_m\_ret = (t\_m[-1] - t\_m[-n])/t\_m[-n]  
  
d4 = tibble(t\_m\_ret, rfree1)  
d4 = d4 %>% mutate(T = t\_m\_ret - rfree1)  
d4

## # A tibble: 60 x 3  
## t\_m\_ret rfree1 T  
## <dbl> <dbl> <dbl>  
## 1 -0.0633 0.000125 -0.0634   
## 2 0.0463 0.000108 0.0461   
## 3 0.0786 0.0000833 0.0785   
## 4 0.0167 0.0000500 0.0166   
## 5 0.0285 0.0000333 0.0285   
## 6 -0.00475 0.0000333 -0.00479  
## 7 -0.0684 0.0000333 -0.0685   
## 8 -0.0132 0.0000167 -0.0133   
## 9 0.00140 0.00000833 0.00140  
## 10 0.0277 0.0000167 0.0277   
## # ... with 50 more rows

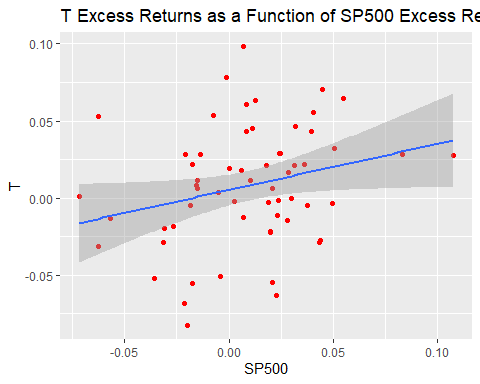
d9 = tibble(SP500 = indices$SP500, T = d4$T)  
d9

## # A tibble: 60 x 2  
## SP500 T  
## <dbl> <dbl>  
## 1 0.0225 -0.0634   
## 2 0.0318 0.0461   
## 3 -0.00113 0.0785   
## 4 0.0284 0.0166   
## 5 -0.0135 0.0285   
## 6 -0.0183 -0.00479  
## 7 -0.0215 -0.0685   
## 8 -0.0568 -0.0133   
## 9 -0.0718 0.00140  
## 10 0.108 0.0277   
## # ... with 50 more rows

pairs.panels(d9)



d9 %>%  
 ggplot(aes(x = SP500, y = T )) +  
 geom\_point(color="red") +  
 geom\_smooth(method="lm") +  
 ggtitle("T Excess Returns as a Function of SP500 Excess Returns")



lm(T ~ SP500, data=d9)

##   
## Call:  
## lm(formula = T ~ SP500, data = d9)  
##   
## Coefficients:  
## (Intercept) SP500   
## 0.005237 0.299116

#Save the result and summarize them  
t.mm = lm(T ~ SP500, data=d9)  
summary(t.mm)

##   
## Call:  
## lm(formula = T ~ SP500, data = d9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.081869 -0.022019 0.001564 0.023014 0.091001   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.005237 0.005000 1.047 0.299   
## SP500 0.299116 0.144570 2.069 0.043 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0375 on 58 degrees of freedom  
## Multiple R-squared: 0.06873, Adjusted R-squared: 0.05268   
## F-statistic: 4.281 on 1 and 58 DF, p-value: 0.04301

#Coefficients summary  
summary(t.mm)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.005236594 0.004999747 1.047372 0.29927334  
## SP500 0.299116395 0.144569670 2.069012 0.04301121

#Isolate beta  
summary(t.mm)$coefficients[2,1]

## [1] 0.2991164

#Isolate sigma  
summary(t.mm)$sigma

## [1] 0.03750058

#Isolate r.squared  
summary(t.mm)$r.squared

## [1] 0.06873403

confint(t.mm) # 95% interval

## 2.5 % 97.5 %  
## (Intercept) -0.004771488 0.01524468  
## SP500 0.009728759 0.58850403

confint(t.mm,level = 0.9) # 90% interval

## 5 % 95 %  
## (Intercept) -0.003120748 0.01359394  
## SP500 0.057460564 0.54077223

#VZ  
  
X5 = get.hist.quote(instrument = "VZ", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

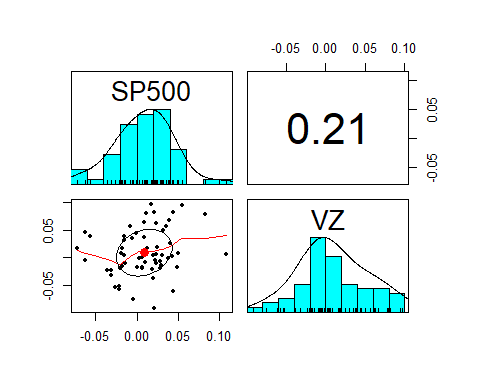
## time series ends 2015-12-01  
vz\_m = as.vector(X5)  
n <- length(vz\_m)  
  
vz\_m\_ret = (vz\_m[-1] - vz\_m[-n])/vz\_m[-n]  
  
d5 = tibble(vz\_m\_ret, rfree1)  
d5 = d5 %>% mutate(VZ = vz\_m\_ret - rfree1)  
d5

## # A tibble: 60 x 3  
## vz\_m\_ret rfree1 VZ  
## <dbl> <dbl> <dbl>  
## 1 -0.00447 0.000125 -0.00460  
## 2 0.0501 0.000108 0.0500   
## 3 0.0439 0.0000833 0.0438   
## 4 -0.0197 0.0000500 -0.0198   
## 5 -0.00991 0.0000333 -0.00994  
## 6 0.00812 0.0000333 0.00809  
## 7 -0.0521 0.0000333 -0.0521   
## 8 0.0383 0.0000167 0.0383   
## 9 0.0174 0.00000833 0.0174   
## 10 0.00489 0.0000167 0.00487  
## # ... with 50 more rows

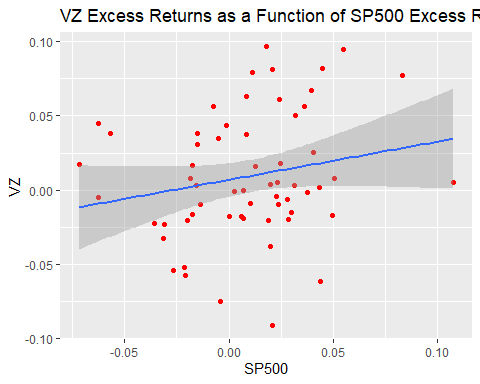
d10 = tibble(SP500 = indices$SP500, VZ = d5$VZ)  
d10

## # A tibble: 60 x 2  
## SP500 VZ  
## <dbl> <dbl>  
## 1 0.0225 -0.00460  
## 2 0.0318 0.0500   
## 3 -0.00113 0.0438   
## 4 0.0284 -0.0198   
## 5 -0.0135 -0.00994  
## 6 -0.0183 0.00809  
## 7 -0.0215 -0.0521   
## 8 -0.0568 0.0383   
## 9 -0.0718 0.0174   
## 10 0.108 0.00487  
## # ... with 50 more rows

pairs.panels(d10)



d10 %>%  
 ggplot(aes(x = SP500, y = VZ )) +  
 geom\_point(color="red") +  
 geom\_smooth(method="lm") +  
 ggtitle("VZ Excess Returns as a Function of SP500 Excess Returns")



lm(VZ ~ SP500, data=d10)

##   
## Call:  
## lm(formula = VZ ~ SP500, data = d10)  
##   
## Coefficients:  
## (Intercept) SP500   
## 0.006832 0.257519

#Save the result and summarize them  
vz.mm = lm(VZ ~ SP500, data=d10)  
summary(vz.mm)

##   
## Call:  
## lm(formula = VZ ~ SP500, data = d10)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.103484 -0.025231 -0.008619 0.035060 0.085320   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.006832 0.005582 1.224 0.226  
## SP500 0.257519 0.161416 1.595 0.116  
##   
## Residual standard error: 0.04187 on 58 degrees of freedom  
## Multiple R-squared: 0.04204, Adjusted R-squared: 0.02552   
## F-statistic: 2.545 on 1 and 58 DF, p-value: 0.1161

#Coefficients summary  
summary(vz.mm)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.006831672 0.00558237 1.223794 0.2259782  
## SP500 0.257519052 0.16141643 1.595371 0.1160645

#Isolate beta  
summary(vz.mm)$coefficients[2,1]

## [1] 0.2575191

#Isolate sigma  
summary(vz.mm)$sigma

## [1] 0.04187054

#Isolate r.squared  
summary(vz.mm)$r.squared

## [1] 0.04203814

confint(vz.mm) # 95% interval

## 2.5 % 97.5 %  
## (Intercept) -0.004342655 0.0180060  
## SP500 -0.065591042 0.5806291

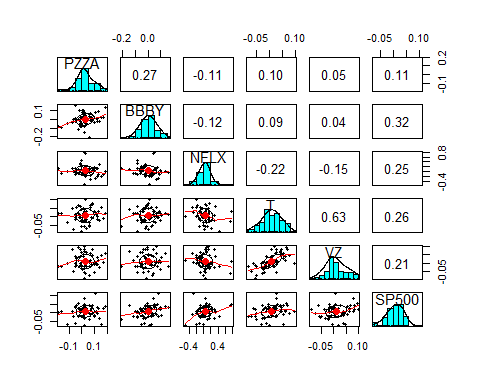
confint(vz.mm,level = 0.9) # 90% interval

## 5 % 95 %  
## (Intercept) -0.002499554 0.0161629  
## SP500 -0.012297031 0.5273351

#BIG5  
big5 = tibble(  
 PZZA = d1$PZZA,  
 BBBY = d2$BBBY,  
 NFLX = d3$NFLX,  
 T = d4$T,  
 VZ = d5$VZ  
)  
big5

## # A tibble: 60 x 5  
## PZZA BBBY NFLX T VZ  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0360 -0.0235 0.218 -0.0634 -0.00460  
## 2 0.0166 0.00302 -0.0347 0.0461 0.0500   
## 3 0.0852 0.00241 0.150 0.0785 0.0438   
## 4 -0.0509 0.163 -0.0215 0.0166 -0.0198   
## 5 0.102 -0.0399 0.164 0.0285 -0.00994  
## 6 0.00389 0.0831 -0.0300 -0.00479 0.00809  
## 7 -0.0617 0.00202 0.0125 -0.0685 -0.0521   
## 8 -0.0468 -0.0279 -0.116 -0.0133 0.0383   
## 9 0.0218 0.00791 -0.518 0.00140 0.0174   
## 10 0.111 0.0790 -0.275 0.0277 0.00487  
## # ... with 50 more rows

#Examine relationship between the big8 and the SP500 index.  
pairs.panels(big5 %>% mutate(SP500=d$SP500))



big5\_mat = as.matrix(big5) # convert to matrix  
sp500 = indices$SP500 # extract from indices  
  
big5.mm = lm(big5\_mat ~ sp500)  
big5.mm

##   
## Call:  
## lm(formula = big5\_mat ~ sp500)  
##   
## Coefficients:  
## PZZA BBBY NFLX T VZ   
## (Intercept) 0.024511 -0.003675 0.032246 0.005237 0.006832  
## sp500 0.224422 0.650968 1.566294 0.299116 0.257519

# Papa John’s International stock is least sensitive  
# Netflix stock is least sensitive

## Question 8.6

x = get.hist.quote(instrument = "^GSPC",  
 start = "2010-12-01",  
 end = "2015-12-31",  
 quote = "AdjClose",  
 compression = "m")

## time series ends 2015-12-01

## time series ends 2015-12-01  
sp500 = as.vector(x)  
n = length(sp500)  
  
# Net returns  
sp500\_ret = (sp500[-1] - sp500[-n])/sp500[-n]  
  
data1 = read.table(file.choose(),header = T,sep = ",")  
data1

## TimePeriod Yield  
## 1 2011-01 0.15  
## 2 2011-02 0.13  
## 3 2011-03 0.10  
## 4 2011-04 0.06  
## 5 2011-05 0.04  
## 6 2011-06 0.04  
## 7 2011-07 0.04  
## 8 2011-08 0.02  
## 9 2011-09 0.01  
## 10 2011-10 0.02  
## 11 2011-11 0.01  
## 12 2011-12 0.01  
## 13 2012-01 0.03  
## 14 2012-02 0.09  
## 15 2012-03 0.08  
## 16 2012-04 0.08  
## 17 2012-05 0.09  
## 18 2012-06 0.09  
## 19 2012-07 0.10  
## 20 2012-08 0.10  
## 21 2012-09 0.11  
## 22 2012-10 0.10  
## 23 2012-11 0.09  
## 24 2012-12 0.07  
## 25 2013-01 0.07  
## 26 2013-02 0.10  
## 27 2013-03 0.09  
## 28 2013-04 0.06  
## 29 2013-05 0.04  
## 30 2013-06 0.05  
## 31 2013-07 0.04  
## 32 2013-08 0.04  
## 33 2013-09 0.02  
## 34 2013-10 0.05  
## 35 2013-11 0.07  
## 36 2013-12 0.07  
## 37 2014-01 0.04  
## 38 2014-02 0.05  
## 39 2014-03 0.05  
## 40 2014-04 0.03  
## 41 2014-05 0.03  
## 42 2014-06 0.04  
## 43 2014-07 0.03  
## 44 2014-08 0.03  
## 45 2014-09 0.02  
## 46 2014-10 0.02  
## 47 2014-11 0.02  
## 48 2014-12 0.03  
## 49 2015-01 0.03  
## 50 2015-02 0.02  
## 51 2015-03 0.03  
## 52 2015-04 0.02  
## 53 2015-05 0.02  
## 54 2015-06 0.02  
## 55 2015-07 0.03  
## 56 2015-08 0.07  
## 57 2015-09 0.02  
## 58 2015-10 0.02  
## 59 2015-11 0.11  
## 60 2015-12 0.22

rff1 = data1$Yield  
table(is.na(rff1))

##   
## FALSE   
## 60

rff1

## [1] 0.15 0.13 0.10 0.06 0.04 0.04 0.04 0.02 0.01 0.02 0.01 0.01 0.03 0.09 0.08  
## [16] 0.08 0.09 0.09 0.10 0.10 0.11 0.10 0.09 0.07 0.07 0.10 0.09 0.06 0.04 0.05  
## [31] 0.04 0.04 0.02 0.05 0.07 0.07 0.04 0.05 0.05 0.03 0.03 0.04 0.03 0.03 0.02  
## [46] 0.02 0.02 0.03 0.03 0.02 0.03 0.02 0.02 0.02 0.03 0.07 0.02 0.02 0.11 0.22

rfree1 = (1 + rff1/100)^(1/12) - 1  
head(rfree1)

## [1] 1.249141e-04 1.082688e-04 8.329516e-05 4.998626e-05 3.332722e-05  
## [6] 3.332722e-05

#PZZA  
x1 = get.hist.quote(instrument = "PZZA", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
pzza\_m = as.vector(x1)  
n = length(pzza\_m)  
  
pzza\_m\_ret = (pzza\_m[-1] - pzza\_m[-n])/pzza\_m[-n]  
  
d1 = tibble(pzza\_m\_ret, rfree1)  
d1 = d1 %>% mutate(PZZA = pzza\_m\_ret - rfree1)  
d1

## # A tibble: 60 x 3  
## pzza\_m\_ret rfree1 PZZA  
## <dbl> <dbl> <dbl>  
## 1 0.0361 0.000125 0.0360   
## 2 0.0167 0.000108 0.0166   
## 3 0.0853 0.0000833 0.0852   
## 4 -0.0508 0.0000500 -0.0509   
## 5 0.102 0.0000333 0.102   
## 6 0.00392 0.0000333 0.00389  
## 7 -0.0616 0.0000333 -0.0617   
## 8 -0.0468 0.0000167 -0.0468   
## 9 0.0218 0.00000833 0.0218   
## 10 0.111 0.0000167 0.111   
## # ... with 50 more rows

#BBBY  
x2 = get.hist.quote(instrument = "BBBY", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
bbby\_m = as.vector(x2)  
n = length(bbby\_m)  
  
bbby\_m\_ret = (bbby\_m[-1] - bbby\_m[-n])/bbby\_m[-n]  
  
d2 = tibble(bbby\_m\_ret, rfree1)  
d2 = d2 %>% mutate(BBBY = bbby\_m\_ret - rfree1)  
d2

## # A tibble: 60 x 3  
## bbby\_m\_ret rfree1 BBBY  
## <dbl> <dbl> <dbl>  
## 1 -0.0234 0.000125 -0.0235   
## 2 0.00313 0.000108 0.00302  
## 3 0.00249 0.0000833 0.00241  
## 4 0.163 0.0000500 0.163   
## 5 -0.0399 0.0000333 -0.0399   
## 6 0.0831 0.0000333 0.0831   
## 7 0.00206 0.0000333 0.00202  
## 8 -0.0279 0.0000167 -0.0279   
## 9 0.00791 0.00000833 0.00791  
## 10 0.0790 0.0000167 0.0790   
## # ... with 50 more rows

#NFLX  
  
x3 = get.hist.quote(instrument = "NFLX", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
nflx\_m = as.vector(x3)  
n = length(nflx\_m)  
  
nflx\_m\_ret = (nflx\_m[-1] - nflx\_m[-n])/nflx\_m[-n]  
  
d3 = tibble(nflx\_m\_ret, rfree1)  
d3 = d3 %>% mutate(NFLX = nflx\_m\_ret - rfree1)  
d3

## # A tibble: 60 x 3  
## nflx\_m\_ret rfree1 NFLX  
## <dbl> <dbl> <dbl>  
## 1 0.218 0.000125 0.218   
## 2 -0.0346 0.000108 -0.0347  
## 3 0.151 0.0000833 0.150   
## 4 -0.0215 0.0000500 -0.0215  
## 5 0.164 0.0000333 0.164   
## 6 -0.0299 0.0000333 -0.0300  
## 7 0.0126 0.0000333 0.0125  
## 8 -0.116 0.0000167 -0.116   
## 9 -0.518 0.00000833 -0.518   
## 10 -0.275 0.0000167 -0.275   
## # ... with 50 more rows

#T  
  
X4 = get.hist.quote(instrument = "T", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
t\_m = as.vector(X4)  
n = length(t\_m)  
  
t\_m\_ret = (t\_m[-1] - t\_m[-n])/t\_m[-n]  
  
d4 = tibble(t\_m\_ret, rfree1)  
d4 = d4 %>% mutate(T = t\_m\_ret - rfree1)  
d4

## # A tibble: 60 x 3  
## t\_m\_ret rfree1 T  
## <dbl> <dbl> <dbl>  
## 1 -0.0633 0.000125 -0.0634   
## 2 0.0463 0.000108 0.0461   
## 3 0.0786 0.0000833 0.0785   
## 4 0.0167 0.0000500 0.0166   
## 5 0.0285 0.0000333 0.0285   
## 6 -0.00475 0.0000333 -0.00479  
## 7 -0.0684 0.0000333 -0.0685   
## 8 -0.0132 0.0000167 -0.0133   
## 9 0.00140 0.00000833 0.00140  
## 10 0.0277 0.0000167 0.0277   
## # ... with 50 more rows

#VZ  
  
X5 = get.hist.quote(instrument = "VZ", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

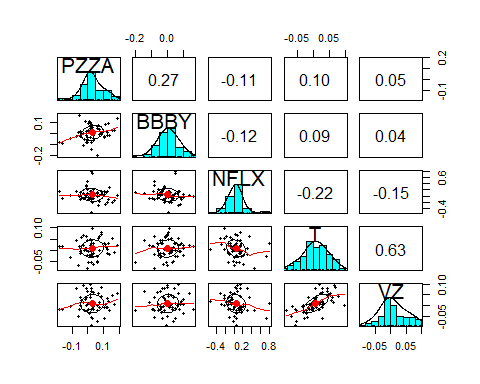
## time series ends 2015-12-01  
vz\_m = as.vector(X5)  
n = length(vz\_m)  
  
vz\_m\_ret = (vz\_m[-1] - vz\_m[-n])/vz\_m[-n]  
  
d5 = tibble(vz\_m\_ret, rfree1)  
d5 = d5 %>% mutate(VZ = vz\_m\_ret - rfree1)  
d5

## # A tibble: 60 x 3  
## vz\_m\_ret rfree1 VZ  
## <dbl> <dbl> <dbl>  
## 1 -0.00447 0.000125 -0.00460  
## 2 0.0501 0.000108 0.0500   
## 3 0.0439 0.0000833 0.0438   
## 4 -0.0197 0.0000500 -0.0198   
## 5 -0.00991 0.0000333 -0.00994  
## 6 0.00812 0.0000333 0.00809  
## 7 -0.0521 0.0000333 -0.0521   
## 8 0.0383 0.0000167 0.0383   
## 9 0.0174 0.00000833 0.0174   
## 10 0.00489 0.0000167 0.00487  
## # ... with 50 more rows

#BIG5  
big5 = tibble(PZZA = d1$PZZA, BBBY = d2$BBBY, NFLX = d3$NFLX, T = d4$T, VZ = d5$VZ)  
  
big5

## # A tibble: 60 x 5  
## PZZA BBBY NFLX T VZ  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0360 -0.0235 0.218 -0.0634 -0.00460  
## 2 0.0166 0.00302 -0.0347 0.0461 0.0500   
## 3 0.0852 0.00241 0.150 0.0785 0.0438   
## 4 -0.0509 0.163 -0.0215 0.0166 -0.0198   
## 5 0.102 -0.0399 0.164 0.0285 -0.00994  
## 6 0.00389 0.0831 -0.0300 -0.00479 0.00809  
## 7 -0.0617 0.00202 0.0125 -0.0685 -0.0521   
## 8 -0.0468 -0.0279 -0.116 -0.0133 0.0383   
## 9 0.0218 0.00791 -0.518 0.00140 0.0174   
## 10 0.111 0.0790 -0.275 0.0277 0.00487  
## # ... with 50 more rows

#Exploratory data analysis  
pairs.panels(big5)



#Estimate betas of big  
big = as.matrix(big5) # convert to matrix  
big.mm = lm(big ~ sp500\_ret )  
big.beta = big.mm$coefficients[2,]  
big.beta

## PZZA BBBY NFLX T VZ   
## 0.2241574 0.6506543 1.5665804 0.2992419 0.2576918

beta.bar = mean(big.beta)  
tausq.beta = mean( (big.beta - beta.bar)^2 )  
  
tibble(beta.bar,tausq.beta) %>% kable()

|  |  |
| --- | --- |
| beta.bar | tausq.beta |
| 0.5996651 | 0.2571462 |

f.betase = function(y){  
 summary(lm(y ~ sp500\_ret))$coefficients[2,2]  
}  
#Compute standard errors of beta\_hat  
big.betase = apply(big,2,f.betase)  
big.betase

## PZZA BBBY NFLX T VZ   
## 0.2762837 0.2493947 0.7923404 0.1445636 0.1614100

sesq.bar = mean(big.betase^2)  
tibble(sesq.bar) %>% kable()

|  |
| --- |
| sesq.bar |
| 0.1626571 |

big.psi = sesq.bar/(sesq.bar + tausq.beta)  
tibble(big.psi) %>% kable()

|  |
| --- |
| big.psi |
| 0.3874603 |

big.psi \* beta.bar + (1 - big.psi)\*big.beta

## PZZA BBBY NFLX T VZ   
## 0.3696517 0.6308980 1.1919392 0.4156440 0.3901929

pzza = big5$PZZA  
summary(lm(pzza ~ sp500\_ret,))

##   
## Call:  
## lm(formula = pzza ~ sp500\_ret)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.205554 -0.032785 -0.007216 0.043905 0.155972   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.024503 0.009558 2.564 0.013 \*  
## sp500\_ret 0.224157 0.276284 0.811 0.420   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07167 on 58 degrees of freedom  
## Multiple R-squared: 0.01122, Adjusted R-squared: -0.005826   
## F-statistic: 0.6583 on 1 and 58 DF, p-value: 0.4205

psi.big = (big.betase^2) / (tausq.beta + big.betase^2)  
psi.big

## PZZA BBBY NFLX T VZ   
## 0.22889807 0.19476724 0.70942272 0.07516284 0.09199589

psi.big\*beta.bar + (1-psi.big)\*big.beta

## PZZA BBBY NFLX T VZ   
## 0.3101104 0.6407233 0.8806288 0.3218226 0.2891519

#Recall that the shrinkage estimates based on a global value for ψ are given by  
big.psi \* beta.bar + (1 - big.psi)\*big.beta

## PZZA BBBY NFLX T VZ   
## 0.3696517 0.6308980 1.1919392 0.4156440 0.3901929

#both of the method are appropiate as at the end we get to same result

## Question 8.10

#COPLX  
x1 = get.hist.quote(instrument = "COPLX", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
coplx\_m = as.vector(x1)  
n = length(coplx\_m)  
  
coplx\_m\_ret = (coplx\_m[-1] - coplx\_m[-n])/coplx\_m[-n]  
  
d1 = tibble(coplx\_m\_ret, rfree1)  
d1 = d1 %>% mutate(COPLX = coplx\_m\_ret - rfree1)  
d1

## # A tibble: 60 x 3  
## coplx\_m\_ret rfree1 COPLX  
## <dbl> <dbl> <dbl>  
## 1 0.0159 0.000125 0.0158   
## 2 0.0223 0.000108 0.0222   
## 3 0.00995 0.0000833 0.00986  
## 4 0.0191 0.0000500 0.0190   
## 5 0.00252 0.0000333 0.00249  
## 6 -0.00671 0.0000333 -0.00674  
## 7 -0.0129 0.0000333 -0.0129   
## 8 0.00321 0.0000167 0.00319  
## 9 -0.0181 0.00000833 -0.0181   
## 10 0.0466 0.0000167 0.0466   
## # ... with 50 more rows

#FCNTX  
x2 = get.hist.quote(instrument = "FCNTX", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

## time series ends 2015-12-01  
fcntx\_m = as.vector(x2)  
n = length(fcntx\_m)  
  
fcntx\_m\_ret = (fcntx\_m[-1] - fcntx\_m[-n])/fcntx\_m[-n]  
  
d2 = tibble(fcntx\_m\_ret, rfree1)  
d2 = d2 %>% mutate(FCNTX = fcntx\_m\_ret - rfree1)  
d2

## # A tibble: 60 x 3  
## fcntx\_m\_ret rfree1 FCNTX  
## <dbl> <dbl> <dbl>  
## 1 0.00591 0.000125 0.00578  
## 2 0.0399 0.000108 0.0398   
## 3 0.00198 0.0000833 0.00189  
## 4 0.0270 0.0000500 0.0270   
## 5 -0.0160 0.0000333 -0.0161   
## 6 -0.0158 0.0000333 -0.0158   
## 7 0.00382 0.0000333 0.00379  
## 8 -0.0491 0.0000167 -0.0491   
## 9 -0.0743 0.00000833 -0.0743   
## 10 0.102 0.0000167 0.102   
## # ... with 50 more rows

#EGFFX  
  
x3 = get.hist.quote(instrument = "EGFFX", start = "2010-12-01", end = "2015-12-31",   
 quote = "AdjClose", compression = "m") # monthly

## time series ends 2015-12-01

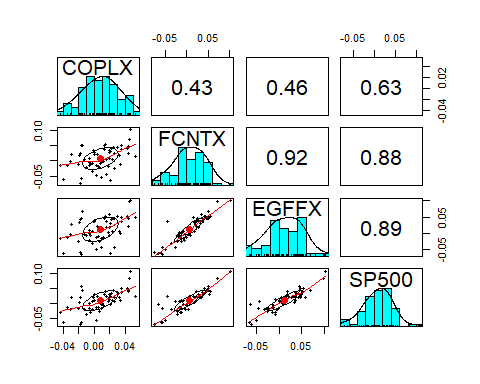
## time series ends 2015-12-01  
egffx\_m = as.vector(x3)  
n = length(egffx\_m)  
  
egffx\_m\_ret = (egffx\_m[-1] - egffx\_m[-n])/egffx\_m[-n]  
  
d3 = tibble(egffx\_m\_ret, rfree1)  
d3 = d3 %>% mutate(EGFFX = egffx\_m\_ret - rfree1)  
d3

## # A tibble: 60 x 3  
## egffx\_m\_ret rfree1 EGFFX  
## <dbl> <dbl> <dbl>  
## 1 0.0172 0.000125 0.0171   
## 2 0.0436 0.000108 0.0435   
## 3 0.00768 0.0000833 0.00760   
## 4 0.0186 0.0000500 0.0186   
## 5 -0.0125 0.0000333 -0.0125   
## 6 -0.000842 0.0000333 -0.000875  
## 7 -0.00421 0.0000333 -0.00425   
## 8 -0.0499 0.0000167 -0.0499   
## 9 -0.0614 0.00000833 -0.0615   
## 10 0.102 0.0000167 0.102   
## # ... with 50 more rows

funds3 = tibble(  
 COPLX = d1$COPLX,  
 FCNTX = d2$FCNTX,  
 EGFFX = d3$EGFFX,  
)  
funds3

## # A tibble: 60 x 3  
## COPLX FCNTX EGFFX  
## <dbl> <dbl> <dbl>  
## 1 0.0158 0.00578 0.0171   
## 2 0.0222 0.0398 0.0435   
## 3 0.00986 0.00189 0.00760   
## 4 0.0190 0.0270 0.0186   
## 5 0.00249 -0.0161 -0.0125   
## 6 -0.00674 -0.0158 -0.000875  
## 7 -0.0129 0.00379 -0.00425   
## 8 0.00319 -0.0491 -0.0499   
## 9 -0.0181 -0.0743 -0.0615   
## 10 0.0466 0.102 0.102   
## # ... with 50 more rows

pairs.panels(funds3 %>% mutate(SP500 = d$SP500))



funds = as.matrix(funds3) # convert to matrix  
funds3.mm = lm(funds ~ sp500\_ret )  
#means  
funds.means = apply(funds3, 2, mean)  
funds.means

## COPLX FCNTX EGFFX   
## 0.007719863 0.007041563 0.011694584

funds.sd = apply(funds3, 2, sd)  
funds.sd

## COPLX FCNTX EGFFX   
## 0.02290522 0.03593476 0.03841071

funds3.port = apply(funds3, 1, mean)  
apply(funds3, 2, sd)

## COPLX FCNTX EGFFX   
## 0.02290522 0.03593476 0.03841071

summary(lm(funds~sp500\_ret))$sigma

## NULL

f.sighat = function(y){summary(lm(y~sp500\_ret))$sigma}  
f.sighat(funds)

## NULL

apply(funds3, 2, f.sighat)

## COPLX FCNTX EGFFX   
## 0.01793400 0.01726126 0.01771230

sd(funds3.port)

## [1] 0.02858367

f.sighat(funds3.port)

## [1] 0.01020645

mean(apply(funds3, 2, sd))

## [1] 0.0324169

mean(apply(funds3, 2, f.sighat))

## [1] 0.01763585

funds3.mm$coefficients[2,]\*sd(sp500)

## COPLX FCNTX EGFFX   
## 134.2356 293.7634 317.5958

f.rsq = function(y){summary(lm(y ~ sp500\_ret))$r.squared}  
apply(funds3, 2, f.rsq)

## COPLX FCNTX EGFFX   
## 0.3973548 0.7731748 0.7909641

f.rsq(funds3.port)

## [1] 0.87466

#Edgewood Growth Retail is most diversified

## Question 8.14

funds = as.matrix(funds3) # convert to matrix  
funds.mm = lm(funds ~ sp500\_ret)  
  
funds.s = apply(funds, 2, f.sighat)  
funds.s

## COPLX FCNTX EGFFX   
## 0.01793400 0.01726126 0.01771230

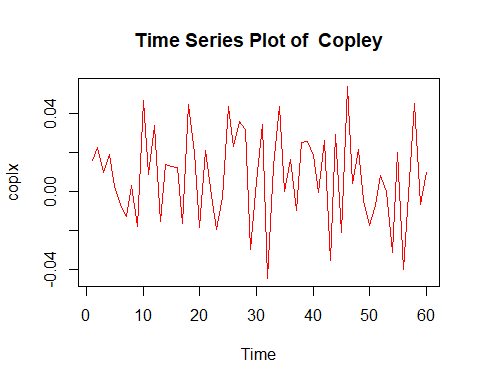
funds.alpha = funds.mm$coefficients[1,]  
funds.appraisal = funds.alpha/funds.s  
funds.appraisal

## COPLX FCNTX EGFFX   
## 0.22343291 -0.06277792 0.16430482

#Copley has the largest estimated appraisel  
# Fidelity Contrafund has the lowest estmated appraisel   
# Edgewood Growth Retail has the 2nd largest estimated appraisel

## Question 8.15

Treynor = function(rmat, ind){  
 ret = rmat[ind,1]  
 mkt = rmat[ind,2]   
 beta = lm(ret ~ mkt)$coefficient[2]  
 mean(ret)/beta  
}  
# function  
Sharpe <- function(x, ind){  
 mean(x[ind])/sd(x[ind])  
}  
  
#COPLX  
coplx = funds3$COPLX  
ts.plot(coplx, col="red")  
title("Time Series Plot of Copley")



coplx5 = coplx[1:5]  
coplx5

## [1] 0.015819461 0.022217062 0.009862629 0.019003729 0.002487744

tibble(mean = mean(coplx5),   
 se = sd(coplx5)/5^0.5) %>% kable()

|  |  |
| --- | --- |
| mean | se |
| 0.0138781 | 0.0035032 |

Treynor(cbind(coplx,sp500\_ret), 1:60)

## mkt   
## 0.01805615

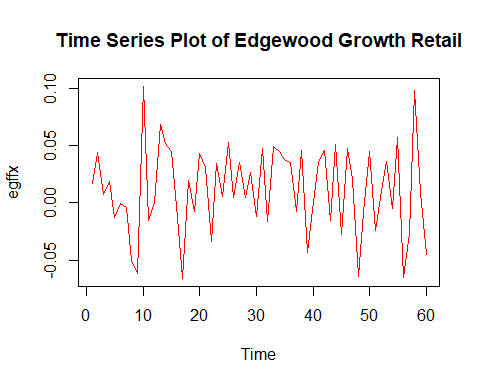
library(boot)  
  
boot(cbind(coplx, sp500\_ret), Treynor, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = cbind(coplx, sp500\_ret), statistic = Treynor, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.01805615 0.000191093 0.007333202

boot(coplx, Sharpe, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = coplx, statistic = Sharpe, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.3370351 0.007802504 0.139025

#EGFFX  
egffx = funds3$EGFFX  
ts.plot(egffx, col="red")  
title("Time Series Plot of Edgewood Growth Retail")



egffx5 = egffx[1:5]  
egffx5

## [1] 0.017085231 0.043524857 0.007595886 0.018578295 -0.012502155

tibble(mean = mean(egffx5),   
 se = sd(egffx5)/5^0.5) %>% kable()

|  |  |
| --- | --- |
| mean | se |
| 0.0148564 | 0.0090606 |

Treynor(cbind(egffx,sp500\_ret), 1:60)

## mkt   
## 0.01156094

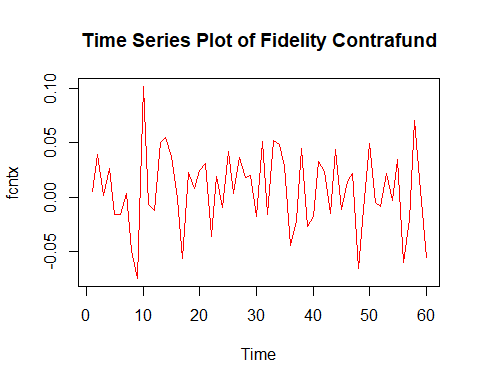
boot(cbind(egffx, sp500\_ret), Treynor, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = cbind(egffx, sp500\_ret), statistic = Treynor, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.01156094 -8.426429e-05 0.004862182

boot(egffx, Sharpe, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = egffx, statistic = Sharpe, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.3044615 0.006045261 0.137671

#FCNTX  
fcntx = funds3$FCNTX  
ts.plot(fcntx, col="red")  
title("Time Series Plot of Fidelity Contrafund")



fcntx5 = fcntx[1:5]  
fcntx5

## [1] 0.005780999 0.039815198 0.001892955 0.026995951 -0.016080527

tibble(mean = mean(fcntx5),   
 se = sd(fcntx5)/5^0.5) %>% kable()

|  |  |
| --- | --- |
| mean | se |
| 0.0116809 | 0.0098168 |

Treynor(cbind(fcntx,sp500\_ret), 1:60)

## mkt   
## 0.007525834

boot(cbind(fcntx, sp500\_ret), Treynor, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = cbind(fcntx, sp500\_ret), statistic = Treynor, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.007525834 -9.610788e-05 0.004909032

boot(fcntx, Sharpe, 10000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = fcntx, statistic = Sharpe, R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 0.1959541 0.006017382 0.1339527