Homework 4

Nikhil Guruji, Jialiang Le, Bingjie Hu

February 5, 2018

Q1-

```
#reading the data
library(httr)
library(data.table)
url1="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/F-F_Research_Data_5_Factors_2x3.CSV"
url2="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_BE-ME.CSV"
url3="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_BETA.csv"
url4="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_INV.CSV"
url5="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_ME.CSV"
url6="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_OP.CSV"


GET(url1, write_disk(tf1 <- tempfile(fileext = ".csv")))
```

```
GET(url2, write_disk(tf2 <- tempfile(fileext = ".csv")))
```

```
GET(url3, write_disk(tf3 <- tempfile(fileext = ".csv")))
```

```
GET(url4, write_disk(tf4 <- tempfile(fileext = ".csv")))
```

```
GET(url5, write_disk(tf5 <- tempfile(fileext = ".csv")))
```

```
GET(url6, write_disk(tf6 <- tempfile(fileext = ".csv")))
```

```
five_factor_data=read.csv(tf1,skip=3)
btm_data=read.csv(tf2,skip=23)
beta_data=read.csv(tf3,skip=15)
inv_data=read.csv(tf4,skip=17)
me_data=read.csv(tf5,skip=12)
op_data=read.csv(tf6,skip=18)

#using missing values for -99.99
five_factor_data[five_factor_data==-99.99] <- NA
btm_data[btm_data==-99.99] <- NA
beta_data[beta_data[,]==-99.99] <- NA
inv_data[inv_data==-99.99] <- NA
me_data[me_data==-99.99] <- NA
op_data[op_data==-99.99] <- NA

#calculating excess returns on quintile portfolios
btm_excess_q=btm_data[,-1] - five_factor_data$RF
btm_excess_q=cbind(btm_data[,1],btm_excess_q[,(5:9)]) #taking only the quintiles

beta_excess_q=beta_data[,-1] - five_factor_data$RF
beta_excess_q=cbind(beta_data[,1],beta_excess_q[,(1:5)])
beta_excess_q=beta_excess_q[1:630,] #taking data only from July 1963 to December 2015

inv_excess_q=inv_data[,-1] - five_factor_data$RF
```

```r
inv_excess_q=cbind(inv_data[,1],inv_excess_q[,(4:8)])
inv_excess_q=inv_excess_q[1:630,]

me_excess_q=me_data[,-1] - five_factor_data$RF
me_excess_q=cbind(me_data[,1],me_excess_q[,(5:9)])
me_excess_q=me_excess_q[1:630,]

op_excess_q=op_data[,-1] - five_factor_data$RF
op_excess_q=cbind(op_data[,1],op_excess_q[,(4:8)])
op_excess_q=op_excess_q[1:630,]



#running multiple regressions for every portfolio for each sort on all five factors and
#storing the alpha, beta, residual details

outbtm_multi=list()
btm_alphas=matrix(nrow=5,ncol=4)
btm_res=matrix(ncol=5,nrow=630)
btm_betas=matrix(ncol=5,nrow=5)
colnames(btm_betas)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:6){
  outbtm_multi[[i-1]]=lm(btm_excess_q[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
       + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  btm_res[,(i-1)] = summary(outbtm_multi[[i-1]])$residuals
  btm_alphas[(i-1),]=summary(outbtm_multi[[i-1]])$coefficients[1,]
  btm_betas[(i-1),]=summary(outbtm_multi[[i-1]])$coefficients[(2:6),1]
}
colnames(btm_alphas)=c("Estimate","Std. Error","t value","Pr (>|t|)")

outbeta_multi=list()
beta_alphas=matrix(nrow=5,ncol=4)
beta_res=matrix(ncol=5,nrow=630)
beta_betas=matrix(ncol=5,nrow=5)
colnames(beta_betas)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:6){
  outbeta_multi[[i-1]]=lm(beta_excess_q[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
                        + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  beta_res[,(i-1)] = summary(outbeta_multi[[i-1]])$residuals
  beta_alphas[(i-1),]=summary(outbeta_multi[[i-1]])$coefficients[1,]
  beta_betas[(i-1),]=summary(outbeta_multi[[i-1]])$coefficients[(2:6),1]
}
colnames(beta_alphas)=c("Estimate","Std. Error","t value","Pr (>|t|)")

outinv_multi=list()
inv_alphas=matrix(nrow=5,ncol=4)
inv_res=matrix(ncol=5,nrow=630)
inv_betas=matrix(ncol=5,nrow=5)
colnames(inv_betas)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:6){
  outinv_multi[[i-1]]=lm(inv_excess_q[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
                        + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  inv_res[,(i-1)] = summary(outinv_multi[[i-1]])$residuals
```

```r
  inv_alphas[(i-1),]=summary(outinv_multi[[i-1]])$coefficients[1,]
  inv_betas[(i-1),]=summary(outinv_multi[[i-1]])$coefficients[(2:6),1]
}
colnames(inv_alphas)=c("Estimate","Std. Error","t value","Pr (>|t|)")

outme_multi=list()
me_alphas=matrix(nrow=5,ncol=4)
me_res=matrix(ncol=5,nrow=630)
me_betas=matrix(ncol=5,nrow=5)
colnames(me_betas)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:6){
  outme_multi[[i-1]]=lm(me_excess_q[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
                        + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  me_res[,(i-1)] = summary(outme_multi[[i-1]])$residuals
  me_alphas[(i-1),]=summary(outme_multi[[i-1]])$coefficients[1,]
  me_betas[(i-1),]=summary(outme_multi[[i-1]])$coefficients[(2:6),1]
}
colnames(me_alphas)=c("Estimate","Std. Error","t value","Pr (>|t|)")

outop_multi=list()
op_alphas=matrix(nrow=5,ncol=4)
op_res=matrix(ncol=5,nrow=630)
op_betas=matrix(ncol=5,nrow=5)
colnames(op_betas)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:6){
  outop_multi[[i-1]]=lm(op_excess_q[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
                        + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  op_res[,(i-1)] = summary(outop_multi[[i-1]])$residuals
  op_alphas[(i-1),]=summary(outop_multi[[i-1]])$coefficients[1,]
  op_betas[(i-1),]=summary(outop_multi[[i-1]])$coefficients[(2:6),1]
}
colnames(op_alphas)=c("Estimate","Std. Error","t value","Pr (>|t|)")

res <- cbind(op_res,inv_res,btm_res,me_res,beta_res)
alphas <- rbind(op_alphas,inv_alphas,btm_alphas,me_alphas,beta_alphas)
alphas_est=alphas[,1]
covariance <- cov(res)

T_1 <- 630
N <- 25
K <- 5
f_avg=colMeans(five_factor_data[,-c(1,7)])
f_avg
```

```
##    Mkt.RF       SMB       HML       RMW       CMA
## 0.4992857 0.2534603 0.3458095 0.2442222 0.2998889
```

```r
sigma_f=cov(five_factor_data[,-c(1,7)])
Ftest <- (T_1-N-K)/N*((1+ t(f_avg)%*%chol2inv(chol(sigma_f))%*%f_avg)**-1)*
  t(alphas_est)%*%chol2inv(chol(covariance))%*%alphas_est
Ftest
```

```
##          [,1]
## [1,] 1.835267
```

```r
qf(p=0.95,df2=T_1-N-K,df1=N)
```

```
## [1] 1.52448
```

```r
# the null is rejected at 95% level of confidence because Ftest>Fcritical
chisq_test <- t(alphas_est)%*%chol2inv(chol(covariance))%*%alphas_est
chisq_test
```

```
##             [,1]
## [1,] 0.08399441
```

```r
qchisq(p=0.95,df=N)
```

```
## [1] 37.65248
```

```r
#cannot reject the null at 95% level of confidence because Xtest<Xcritical
```

The null is rejected (at 95% level of confidence) using the F test because Ftest>Fcritical but we cannot reject the null (at 95% level of confidence) using the Chi squared test because Xtest<Xcritical.

Q2-

```r
#2
Erie=c(colMeans(op_excess_q[,-1]),colMeans(inv_excess_q[,-1])
           ,colMeans(btm_excess_q[,-1]),colMeans(me_excess_q[,-1])
           ,colMeans(beta_excess_q[,-1]))
betas=rbind(op_betas,inv_betas,btm_betas,me_betas,beta_betas)
crsreg=lm(Erie~betas) #regressing excess returns on Betas
summary(crsreg)
```

```
##
## Call:
## lm(formula = Erie ~ betas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06745 -0.03500 -0.01274  0.03271  0.10279
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.22726    0.14570   1.560  0.13530
## betasMkt.RF  0.27330    0.14169   1.929  0.06882 .
## betasSMB     0.25105    0.03616   6.942 1.29e-06 ***
## betasHML     0.24697    0.05013   4.927 9.36e-05 ***
## betasRMW     0.20579    0.05431   3.789  0.00124 **
## betasCMA     0.16507    0.04809   3.432  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0518 on 19 degrees of freedom
## Multiple R-squared:  0.8226, Adjusted R-squared:  0.7759
## F-statistic: 17.62 on 5 and 19 DF,  p-value: 1.495e-06
```

Here we can see the lambdas and their standard errors using OLS (without Shanken Correction)

```r
all_the_data <- cbind(op_excess_q[,-1],inv_excess_q[,-1],btm_excess_q[,-1],me_excess_q[,-1]
                      ,beta_excess_q[,-1])
all_the_data <- as.matrix(all_the_data)
```

```r
lambdas_fama_mb=matrix(nrow=nrow(all_the_data),ncol=6)
lambdas_std_err_fmb=matrix(nrow=nrow(all_the_data),ncol=6)
for(i in 1:nrow(all_the_data)){
  lambdas_fama_mb[i,]=summary(lm(all_the_data[i,]~betas))$coefficients[1:6]
  lambdas_std_err_fmb[i,]=summary(lm(all_the_data[i,]~betas))$coef[,2]
}
lambdas <- colMeans(lambdas_fama_mb)
lambdas_std_err=colMeans(lambdas_std_err_fmb)
lambdas_comparison=cbind(lambdas,summary(crsreg)$coefficients[1:6])
colnames(lambdas_comparison)=c("By Fama Macbeth","By OLS")
lambdas_comparison
```

```
##      By Fama Macbeth    By OLS
## [1,]       0.2272600 0.2272600
## [2,]       0.2732982 0.2732982
## [3,]       0.2510527 0.2510527
## [4,]       0.2469731 0.2469731
## [5,]       0.2057888 0.2057888
## [6,]       0.1650697 0.1650697
```

```r
lambdas_se_comparison=cbind(lambdas_std_err,summary(crsreg)$coef[,2])
colnames(lambdas_se_comparison)=c("By Fama Macbeth","By OLS")

lambdas_se_comparison #standard errors of lambdas by Fama Macbeth vs OLS (without Shanken Correction)
```

```
##             By Fama Macbeth     By OLS
## (Intercept)       2.6714006 0.14569549
## betasMkt.RF       2.5979065 0.14168720
## betasSMB          0.6630482 0.03616198
## betasHML          0.9190926 0.05012638
## betasRMW          0.9958653 0.05431349
## betasCMA          0.8818345 0.04809436
```

```r
#including Shanken correction
lam=lambdas[-1]
shanken=(1/T_1)*(chol2inv(chol(crossprod(betas)))%*%t(betas)%*%covariance%*%
                   betas%*%(chol2inv(chol(crossprod(betas))))*as.integer(1+(t(lam)%*%sigma_f
                                                                            %*%lam)) +
                   sigma_f)
sigma_lambda <- diag(shanken) #OLS errors with Shanken correction
sigma_lambda
```

```
##     Mkt.RF        SMB        HML        RMW        CMA
## 0.03196553 0.01602150 0.01667703 0.01302494 0.01298983
```

```r
a <- summary(crsreg)
cs_rsquare <- a$r.squared
cs_rsquare #cross sectional R squared
```

```
## [1] 0.8225883
```

Q3-

```r
#3
#a

#reading the data
```

```r
url7="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_NI.csv"
url8="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/Portfolios_Formed_on_RESVAR.csv"
url9="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/10_Industry_Portfolios.CSV"
urlx="https://github.com/nikhilg12/nikhil/raw/master/empiricalhw4/10_Portfolios_Prior_12_2.CSV"

GET(url7, write_disk(tf7 <- tempfile(fileext = ".csv")))
```

```r
GET(url8, write_disk(tf8 <- tempfile(fileext = ".csv")))
```

```r
GET(url9, write_disk(tf9 <- tempfile(fileext = ".csv")))
```

```r
GET(urlx, write_disk(tfx <- tempfile(fileext = ".csv")))
```

```r
ni_data=read.csv(tf7,skip=16)
resvar_data=read.csv(tf8,skip=16)
ten_industry_data=read.csv(tf9,skip=11)
mom_data=read.csv(tfx,skip=10)

#using missing values for -99.99
ni_data[ni_data==-99.99] <- NA
resvar_data[resvar_data==-99.99] <- NA
ten_industry_data[ten_industry_data==-99.99] <- NA
mom_data[mom_data==-99.99] <- NA

#taking the excess returns and respective portfolios
ni_d=ni_data[(1:630),-1] - five_factor_data$RF
ni_d=cbind(ni_data[(1:630),1],ni_d[,(3:7)]) #taking only the quintiles

resvar_d=resvar_data[(1:630),-1] - five_factor_data$RF  #taking data only from July 1963 to December 20
resvar_d=cbind(resvar_data[(1:630),1],resvar_d[,(1:5)])

ten_industry_data=ten_industry_data[(1:630),-1] - five_factor_data$RF
ten_industry_data=cbind(resvar_d[(1:630),1],ten_industry_data)

mom_d=mom_data[(1:630),-1] - five_factor_data$RF
mom_d=cbind(resvar_d[(1:630),1],mom_d)

#combining all the 30 portfolios
all_portfolios=cbind(mom_d,ni_d[,-1],resvar_d[,-1],ten_industry_data[,-1])


#running 30 regressions on each factor
out_multi=list()
alphas3=matrix(nrow=30,ncol=4)
res3=matrix(ncol=30,nrow=630)
betas3=matrix(ncol=5,nrow=30)
colnames(betas3)=colnames(five_factor_data[,-c(1,7)])
for(i in 2:ncol(all_portfolios)){ #skipping the date column
  out_multi[[i-1]]=lm(all_portfolios[,i]~five_factor_data$Mkt.RF + five_factor_data$SMB
                        + five_factor_data$HML + five_factor_data$RMW + five_factor_data$CMA)
  res3[,(i-1)] = summary(out_multi[[i-1]])$residuals
  alphas3[(i-1),]=summary(out_multi[[i-1]])$coefficients[1,]
  betas3[(i-1),]=summary(out_multi[[i-1]])$coefficients[(2:6),1]
}
```

```r
all_portfolios_no_date=all_portfolios[,-1] #not considering the date
all_portfolios_no_date=as.matrix(all_portfolios_no_date)

lambdas_fama_mb_3=matrix(nrow=nrow(all_portfolios_no_date),ncol=6)
lambdas_std_err_fmb_3=matrix(nrow=nrow(all_portfolios_no_date),ncol=6)
for(i in 1:nrow(all_portfolios_no_date)){
  lambdas_fama_mb_3[i,]=summary(lm(all_portfolios_no_date[i,]~betas3))$coefficients[1:6]
  lambdas_std_err_fmb_3[i,]=summary(lm(all_portfolios_no_date[i,]~betas3))$coef[,2]
}
lambdas <- colMeans(lambdas_fama_mb_3)
lambdas_std_err=colMeans(lambdas_std_err_fmb_3)

Erie_3=c(colMeans(mom_d[,-1]),colMeans(ni_d[,-1]),colMeans(resvar_d[,-1])
         ,colMeans(ten_industry_data[,-1]))

crsreg_3=lm(Erie_3~betas3)
a_3 <- summary(crsreg_3)
a_3 # coefficients and standard errors for Q3
```

```
##
## Call:
## lm(formula = Erie_3 ~ betas3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24984 -0.06908  0.00584  0.08167  0.35465
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.1309     0.3832   0.342  0.73568
## betas3Mkt.RF   0.4120     0.3904   1.055  0.30173
## betas3SMB      0.1287     0.2068   0.622  0.53959
## betas3HML     -0.5857     0.1295  -4.523  0.00014 ***
## betas3RMW      0.1626     0.1904   0.854  0.40154
## betas3CMA      0.3881     0.2521   1.540  0.13674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1391 on 24 degrees of freedom
## Multiple R-squared:  0.6869, Adjusted R-squared:  0.6217
## F-statistic: 10.53 on 5 and 24 DF,  p-value: 1.927e-05
```

```r
summary(crsreg) #coefficients and standard errors for Q2
```

```
##
## Call:
## lm(formula = Erie ~ betas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06745 -0.03500 -0.01274  0.03271  0.10279
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.22726    0.14570   1.560  0.13530
## betasMkt.RF  0.27330    0.14169   1.929  0.06882 .
## betasSMB     0.25105    0.03616   6.942 1.29e-06 ***
## betasHML     0.24697    0.05013   4.927 9.36e-05 ***
## betasRMW     0.20579    0.05431   3.789  0.00124 **
## betasCMA     0.16507    0.04809   3.432  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0518 on 19 degrees of freedom
## Multiple R-squared:  0.8226, Adjusted R-squared:  0.7759
## F-statistic: 17.62 on 5 and 19 DF,  p-value: 1.495e-06
```

```r
a_3$r.squared #cross sectional R squared
```

```
## [1] 0.6869437
```

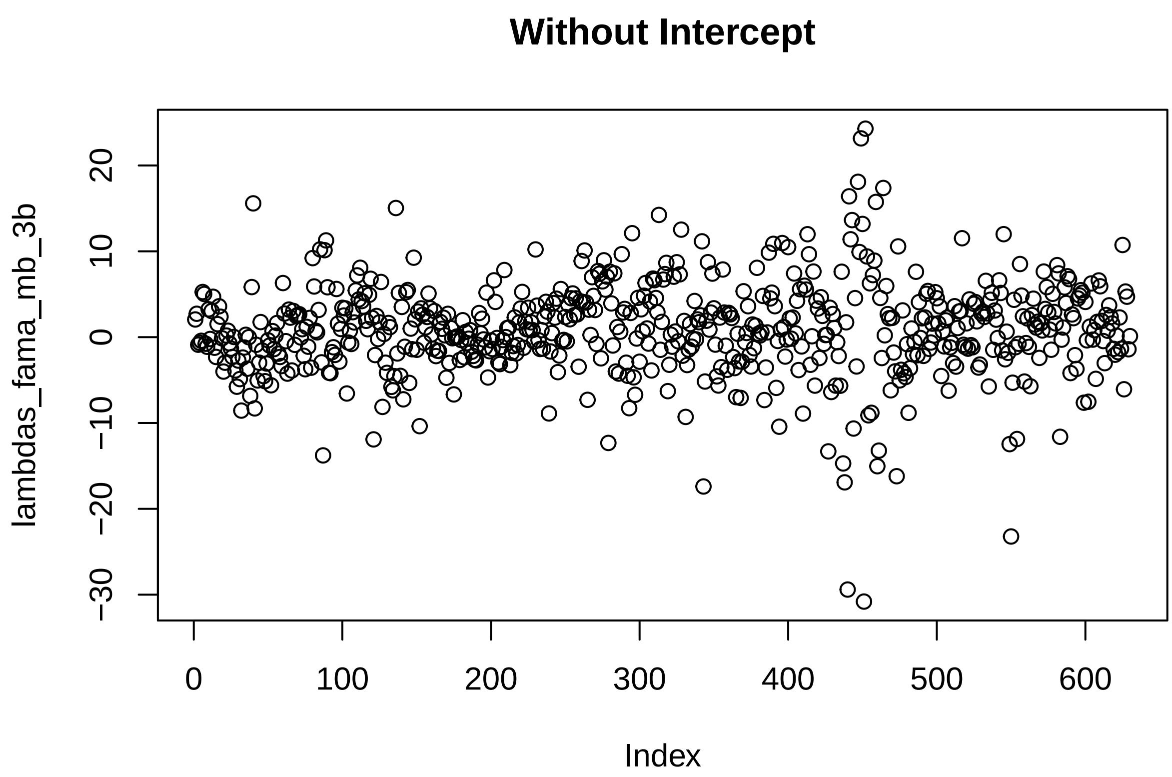
```r
#b

#first, running the regressions without the intercept

#combining all the 30 portfolios
all_portfolios=cbind(mom_d,ni_d[,-1],resvar_d[,-1],ten_industry_data[,-1])

out_multib=list()
res3b=matrix(ncol=30,nrow=630)
betas3b=matrix(ncol=1,nrow=30)
colnames(betas3b)=colnames(five_factor_data[,2])
for(i in 2:ncol(all_portfolios)){ #skipping the date column
  out_multib[[i-1]]=lm(all_portfolios[,i]~0+five_factor_data$Mkt.RF)
  res3b[,(i-1)] = summary(out_multib[[i-1]])$residuals
  betas3b[(i-1),]=summary(out_multib[[i-1]])$coefficients[1,1]
}

# running the regressions with the intercept

out_multib2=list()
alphas3b2=matrix(nrow=30,ncol=4)
res3b2=matrix(ncol=30,nrow=630)
betas3b2=matrix(ncol=1,nrow=30)
colnames(betas3b2)=colnames(five_factor_data[,2])
for(i in 2:ncol(all_portfolios)){ #skipping the date column
  out_multib2[[i-1]]=lm(all_portfolios[,i]~five_factor_data$Mkt.RF)
  res3b2[,(i-1)] = summary(out_multib2[[i-1]])$residuals
  alphas3b2[(i-1),]=summary(out_multib2[[i-1]])$coefficients[1,1]
  betas3b2[(i-1),]=summary(out_multib2[[i-1]])$coefficients[2,1]
}



betas_comp_3b=cbind(betas3b,betas3b2)
colnames(betas_comp_3b)=c("Without Intercept","With Intercept")
betas_comp_3b # comparing betas with and without the intercept
```

```
##       Without Intercept With Intercept
```

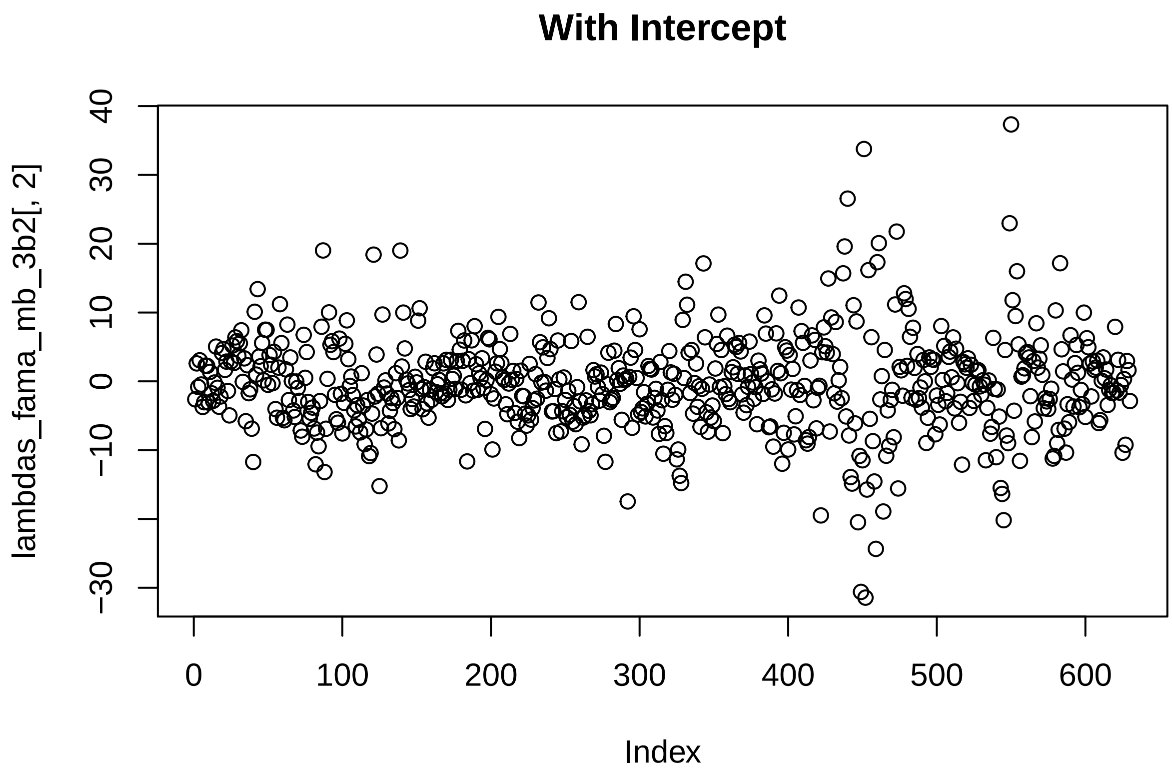
```
##  [1,]        1.4139550     1.4382058
##  [2,]        1.1732472     1.1812952
##  [3,]        1.0184663     1.0203053
##  [4,]        0.9609371     0.9608765
##  [5,]        0.9192574     0.9202454
##  [6,]        0.9372513     0.9365034
##  [7,]        0.9072693     0.9055038
##  [8,]        0.9272822     0.9217820
##  [9,]        0.9894927     0.9833083
## [10,]        1.1914840     1.1784775
## [11,]        0.9048071     0.9034076
## [12,]        0.9988379     0.9981862
## [13,]        1.0610621     1.0573769
## [14,]        1.1370316     1.1397281
## [15,]        1.1263088     1.1358710
## [16,]        0.7802338     0.7767642
## [17,]        0.9781090     0.9762954
## [18,]        1.1281364     1.1251207
## [19,]        1.3129355     1.3118574
## [20,]        1.5361481     1.5497192
## [21,]        0.8083825     0.8012517
## [22,]        1.1384873     1.1414747
## [23,]        1.0424373     1.0414983
## [24,]        0.8082177     0.8032773
## [25,]        1.2527688     1.2537739
## [26,]        0.7854209     0.7836000
## [27,]        1.0066304     1.0031738
## [28,]        0.8428667     0.8359987
## [29,]        0.5314984     0.5276382
## [30,]        1.1044212     1.1050299
```

```r
lambdas_fama_mb_3b=matrix(nrow=nrow(all_portfolios_no_date),ncol=1)
lambdas_fama_mb_3b2=matrix(nrow=nrow(all_portfolios_no_date),ncol=2)

lambdas_std_err_fmb_3b=matrix(nrow=nrow(all_portfolios_no_date),ncol=1)
lambdas_std_err_fmb_3b2=matrix(nrow=nrow(all_portfolios_no_date),ncol=2)

for(i in 1:nrow(all_portfolios_no_date)){
  lambdas_fama_mb_3b[i,]=summary(lm(all_portfolios_no_date[i,]~betas3b))$coefficients[1]
  lambdas_fama_mb_3b2[i,]=summary(lm(all_portfolios_no_date[i,]~betas3b2))$coefficients[1:2]

  lambdas_std_err_fmb_3b[i,]=summary(lm(all_portfolios_no_date[i,]~betas3b))$coef[1,2]
  lambdas_std_err_fmb_3b2[i,]=summary(lm(all_portfolios_no_date[i,]~betas3b2))$coef[,2]

}
lambdas_3b <- colMeans(lambdas_fama_mb_3b)
lambdas_3b2 <- colMeans(lambdas_fama_mb_3b2)

lambdas_std_err_3b=colMeans(lambdas_std_err_fmb_3b)

Erie_3b=cbind(rowMeans(mom_d[,-1]),rowMeans(ni_d[,-1]),rowMeans(resvar_d[,-1])
             ,rowMeans(ten_industry_data[,-1]))
colnames(Erie_3b)=c("Momentum Sorted Portfolio","Net Income Portfolio","Residual Variance Portfolio",
                   "10 Industry Portfolios")
```

```r
plot(lambdas_fama_mb_3b,main="Without Intercept")
```

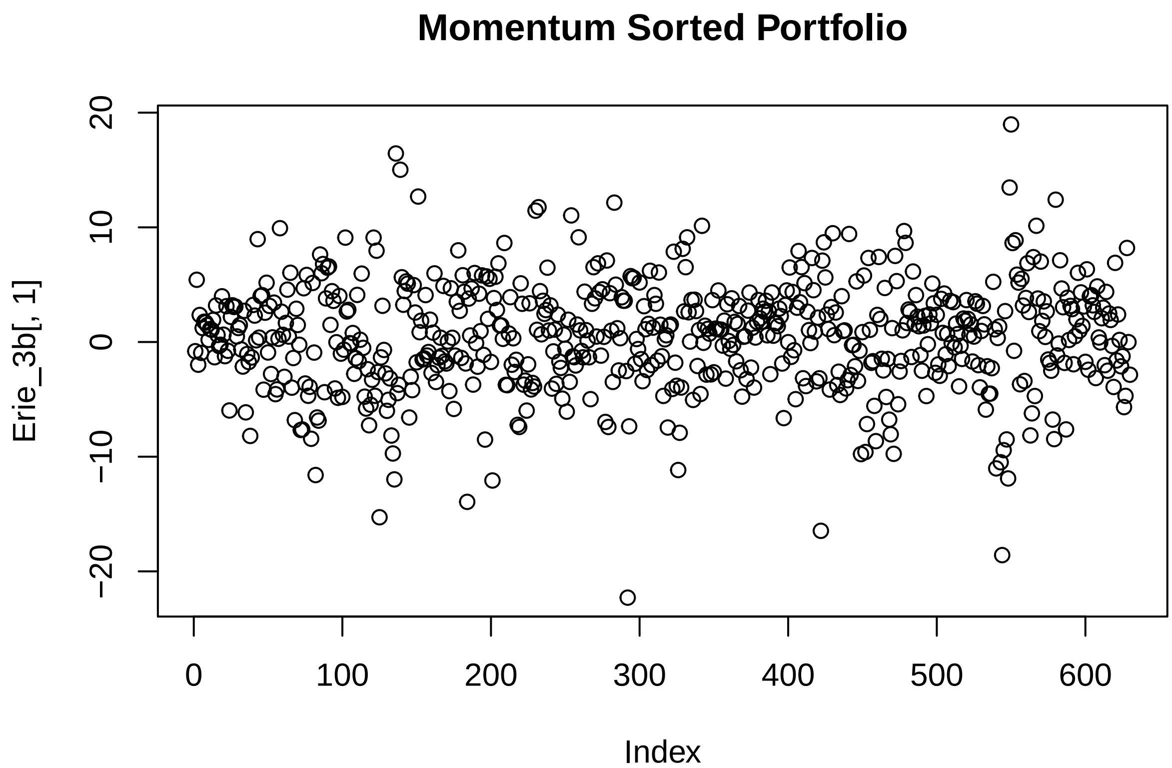



```r
plot(lambdas_fama_mb_3b2[,2],main="With Intercept")
```

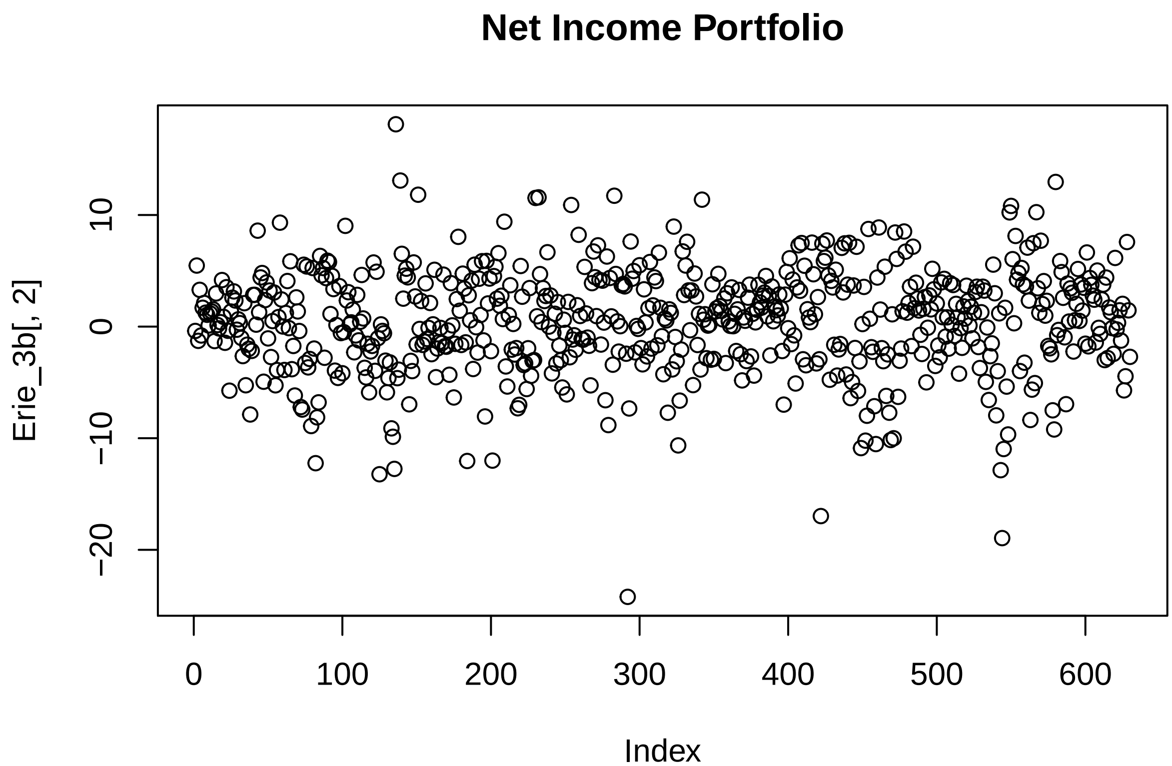


It is observed from the plots above that the λ_t's are almost the same with or without the intercept. So, the results are not sensitive

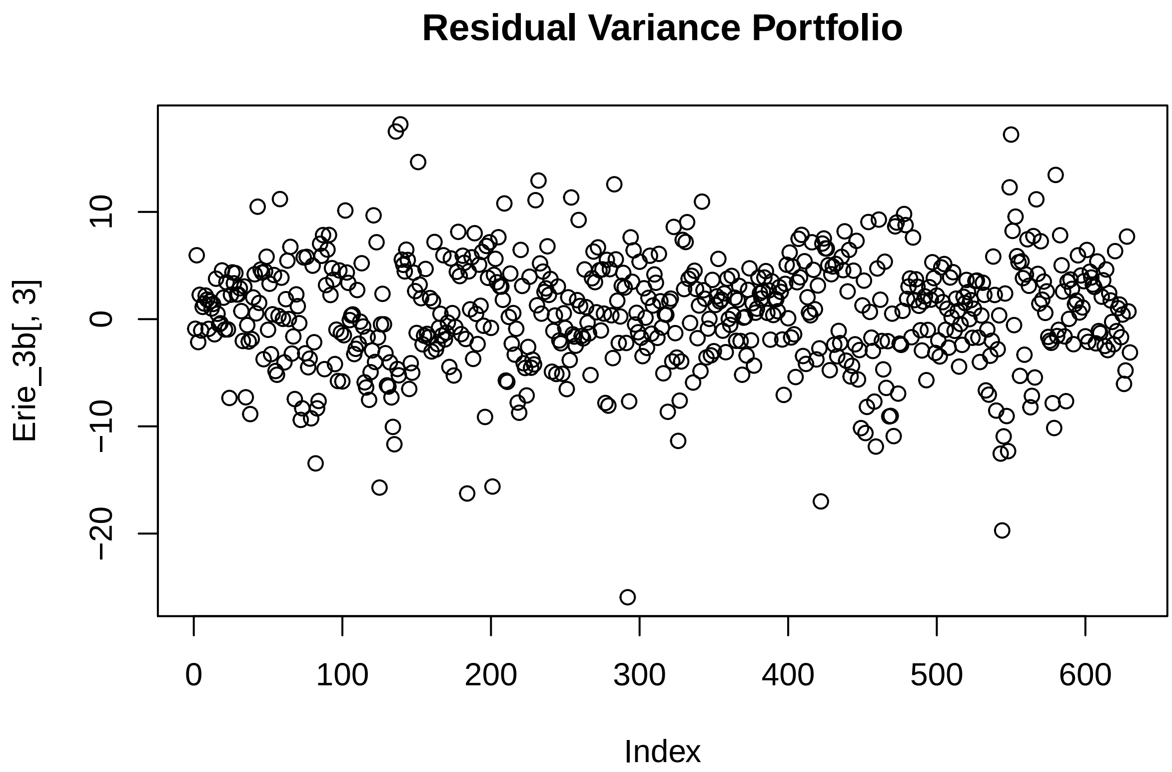
```r
#excess market returns
plot(Erie_3b[,1],main="Momentum Sorted Portfolio")
```



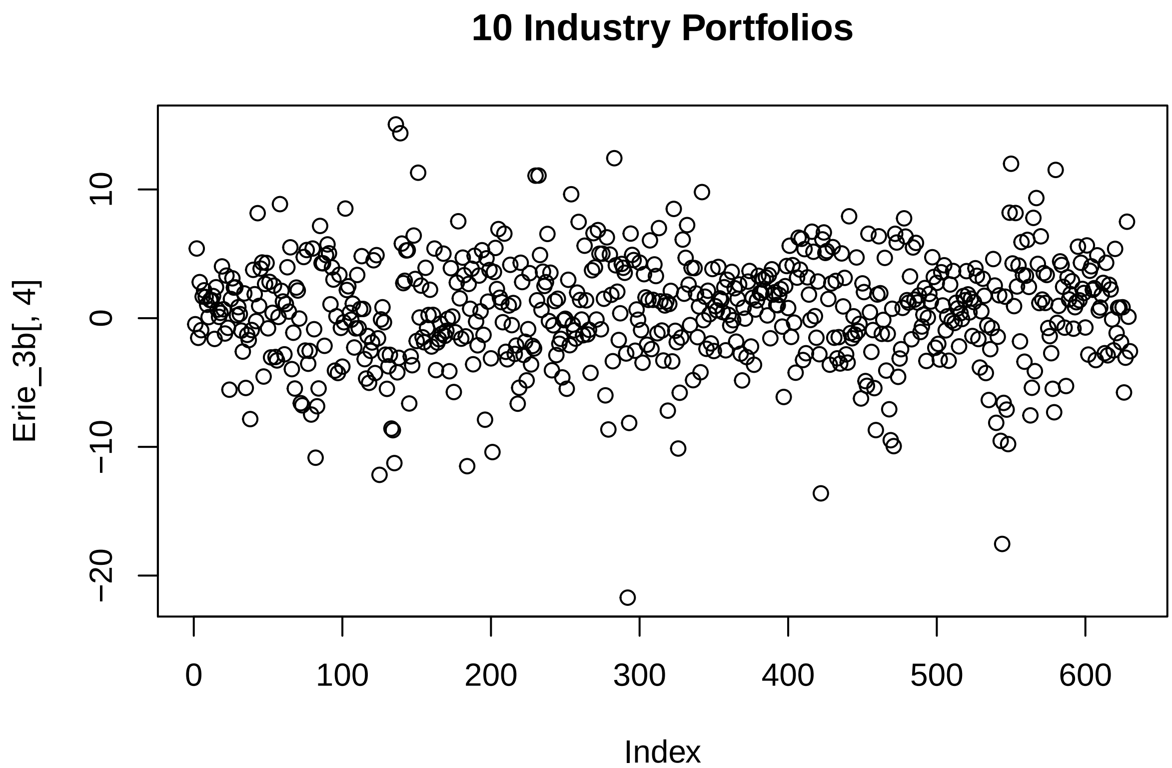

```
plot(Erie_3b[,2],main="Net Income Portfolio")
```




```r
plot(Erie_3b[,3],main="Residual Variance Portfolio")
```




```r
plot(Erie_3b[,4],main="10 Industry Portfolios")
```




```r
cor_matrix=cor(Erie_3b)
cor_matrix
```

```
##                              Momentum Sorted Portfolio Net Income Portfolio
## Momentum Sorted Portfolio                    1.0000000            0.9739504
## Net Income Portfolio                         0.9739504            1.0000000
## Residual Variance Portfolio                  0.9816406            0.9845423
## 10 Industry Portfolios                       0.9804335            0.9782617
##                              Residual Variance Portfolio
## Momentum Sorted Portfolio                      0.9816406
## Net Income Portfolio                           0.9845423
## Residual Variance Portfolio                    1.0000000
## 10 Industry Portfolios                         0.9700136
##                              10 Industry Portfolios
## Momentum Sorted Portfolio                 0.9804335
## Net Income Portfolio                      0.9782617
## Residual Variance Portfolio               0.9700136
## 10 Industry Portfolios                    1.0000000
```