TEST FAMA-FRENCH THREE FACTOR MODEL USING FAMA-MACBETH APPROACH

RAN ZHAO

ABSTRACT. In this paper, we use the Fama-MacBeth [2] approach to estimate and evaluate Fama-French three factor model [3]. The market factor and SMB factors have positive and statistically significant coefficients, and HMB has negative but non-significant coefficient. The pricing errors terms seem not to be 0 under Fama-French three-factor model framework.

1. INTRODUCTION

Fama and French (1993) identify three common stock market risk factors: overall market factor (RMF), firm size (SMB) and book-to-market (HML), as shown in estimation equation 1. Their testing results show that these three factors explain average returns on stocks.

$$r_i - r_f = \beta_0 + \beta_1(r_m - r_f) + \beta_2 SMB + \beta_3 HML + \epsilon_i \tag{1}$$

where r_i is the rate of return for security i, r_f represents risk-free rate and ϵ is the pricing error. The risk factors are market $(r_m - r_f)$, small market capitalization minus big (SMB) and high book-to-market ratio minus low.

The motivation of constructing three-factor model is based on the stock market observations, where two classes stocks tend to have better than market returns: small-cap stocks and low PB ratio stocks. The observation seems to violate traditional capital asset pricing model (CAPM) model, as in the model the only risk factor is overall market. Fama and French show that the extended model improves the explanatory power of the factor model to average stock market returns.

In following section, I present the coefficient estimation and model validation using Fama-MacBeth approach.

2. ESTIMATION AND EVALUATION

2.1. **Fama-Macbeth Approach.** Fama and MacBeth [2] suggest a two-stage procedure for estimating cross-sectional regressions. The first pass is to run time-series regression. Fama and MacBeth use rolling five years regressions and I use the technique with full-sample betas, just for faster computation. The estimation model is selected as in Equation 2.

$$r_{i,t} - r_{f,t} = \beta_{0,i} + \beta_{1,i}(r_{m,t} - r_{f,t}) + \beta_{2,i} SMB_t + \beta_{3,i} HML_t + \epsilon_i \tag{2}$$

Note that the regression is with respect to each security i. The Fama-French three-factor risk factors (in terms of returns) are available on French's public website [1]. In this research, I use the month frequency for both risk factors and individual stock returns. The individual stock returns are from CRSP [2]. The data starts from July, 1926 and ends with September, 2016. There are 30,297 individual stocks in the time series regression.

The second pass is to run cross-sectional regression at each time period, i.e.,

[1] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_portfolios.html

[2] When processing the data, I exclude stocks with less than 10 observations during research period

Table 1 The mean, standard error, t-statistics and p-value from cross-sectional regression.

	constant	rm-rf	HML	SMB
mean	-0.2721	0.3363	-0.04807	0.1940
s.e.	0.4432	0.0474	0.0352	0.0299
t-stats	-0.6139	7.0948	-1.3626	6.4831
p-value	0.5394	0.0000***	0.1733	0.0000***

$$r^e_{i,t} = \lambda_{0,t} + \lambda_{1,t}\beta_{RMF,i} + \lambda_{2,t}\beta_{SMB,i} + \lambda_{3,t}\beta_{HML,i} + \alpha_{i,t} \tag{3}$$

Then we estimate λ and the pricing error as the average of the cross-sectional regression estimates,

$$\hat{\lambda} = \frac{1}{T}\sum_{t=1}^{T}\lambda_t, \qquad \hat{\alpha}_i = \frac{1}{T}\sum_{t=1}^{T}\alpha_{i,t}$$

The standard deviation of the cross-sectional regression estimates to generate the sampling errors from,

$$\sigma^2(\hat{\lambda}) = \frac{1}{T^2}\sum_{t=1}^{T}(\hat{\lambda}_t - \hat{\lambda})^2, \qquad \sigma^2(\hat{\alpha}_i) = \frac{1}{T^2}\sum_{t=1}^{T}(\hat{\alpha}_{i,t} - \hat{\alpha}_i)^2$$

Cochrane [1] proposes the test on whether all the pricing errors are jointly zero. As the pricing error denoted by α shown in Equation 3, the covariance matrix of the sample pricing error is

$$\begin{aligned} \hat{\alpha} &= \frac{1}{T}\sum_{t=1}^{T}\hat{\alpha}_t \\ cov(\hat{\alpha}) &= \frac{1}{T^2}\sum_{t=1}^{T}(\hat{\alpha}_t - \hat{\alpha})(\hat{\alpha}_t - \hat{\alpha})' \end{aligned}$$

and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ where N is the total number of securities. Finally we should test

$$\hat{\alpha}' cov(\hat{\alpha}^{-1}\hat{\alpha} \sim \chi^2_{N-1}$$

2.2. **Empirical Results.** The mean and standard errors of the cross-sectional regressions are reported in Table 1. The market factor and SMB factor are positive and statistically significant. The HML factor, however, has negative mean and is not significant at 10% level. The constant is expected to be non-significant.

The chi-squared statistic is 1.4670, and the null hypothesis is rejected that all pricing errors are zero.

REFERENCES

[1] Cochrane, H. John. Asset pricing. *Princeton university press*, 2009.

[2] Fama, F. Eugene and James D. MacBeth. Risk, return, and equilibrium: Empirical tests. *Journal of Financial Economics*, 81:607–636, 1973.

[3] Fama, F. Eugene and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56, 1993.

APPENDIX: CODE

```
setwd('C:\\Users\\ranzhao\\Documents\\Empirical-Asset-Pricing\\Assignment 8')
setwd('D:\\PhD FE\\Empirical-Asset-Pricing\\Assignment 8')
setwd('D:\\Empirical-Asset-Pricing\\Assignment 8')

require("data.table")

ff.factors = fread('F-F_Research_Data_Factors.CSV', data.table=FALSE)
colnames(ff.factors)[1] = 'date'

stock.data = fread('individual_stocks.csv', header=TRUE, data.table=FALSE, select=c('
    PERMNO','date','RET'))
stock.data = stock.data[stock.data[,3] != '' & stock.data[,3] != 'C' & stock.data[,3]
    != 'B',]
stock.data[,3] = as.numeric(stock.data[,3])
stock.data[,2] = substr(stock.data[,2], 1, 6)
ff.start = min(ff.factors$date)
ff.end = max(ff.factors$date)
stock.data = stock.data[stock.data$date >= ff.start & stock.data$date <= ff.end, ]

asset.set = unique(stock.data$PERMNO)

# exclude stock with less than 10 ovservations in research period
exclude.stock = c()
for (i in 1:length(asset.set)){
  permno = asset.set[i]
  stock = stock.data[stock.data[,1]==permno, c(2,3)]
  if (nrow(stock) < 10){
    exclude.stock = c(exclude.stock, permno)
    stock.data = stock.data[stock.data$PERMNO != permno, ]
  }

  if (i %% 1000 == 0){
    cat("iteration = ", i, "\n")
  }
}

length(exclude.stock)

# cleaned data
asset.set = unique(stock.data$PERMNO)
first.pass.betas = matrix(0, nrow=length(asset.set), ncol=4,
                          dimnames=list(sort(asset.set), c('constant', 'rm_rf' ,'HML',
                              'SMB')))
first.pass.betas = as.data.frame(first.pass.betas)

# first pass regression: time series on each asset
for (i in 1:length(asset.set)){
  permno = asset.set[i]
  stock = stock.data[stock.data[,1]==permno, c(2,3)]
  merged.data = merge(stock, ff.factors, by.x = "date", by.y = "date")
  if (nrow(merged.data) < 10){  # exclude data with less than 10 time series
      observations
    exclude.stock = c(exclude.stock, permno)
  }
  else{
    reg.model = lm(RET-RF~`Mkt-RF`+SMB+HML, data=merged.data)
```

```
    first.pass.betas[row.names(first.pass.betas) == permno,] = as.numeric(reg.model$
        coefficients)
  }

  if (i %% 1000 == 0){
    cat("iteration = ", i, "\n")
  }
}

# second pass regression: cross sectional over time
research.period = unique(ff.factors$date)
second.pass.lambdas = matrix(0, nrow=length(research.period), ncol=4,
                             dimnames=list(sort(research.period), c('constant', 'rm_rf
                                 ','HML', 'SMB')))
second.pass.lambdas = as.data.frame(second.pass.lambdas)
second.pass.errors = matrix(0, nrow=length(asset.set), ncol=length(research.period),
                            dimnames=list(sort(asset.set), sort(research.period)))
first.pass.betas$perm = row.names(first.pass.betas)

for (j in 1:length(research.period)){
  date = research.period[j]
  stock = stock.data[stock.data$date == date, ]
  merged.data = merge(stock, first.pass.betas, by.x = "PERMNO", by.y = "perm")
  rf = ff.factors$RF[ff.factors$date == date]
  reg.model = lm(RET-rf~rm_rf+SMB+HML,data=merged.data)
  # lambdas
  second.pass.lambdas[row.names(second.pass.lambdas) == date, ] = as.numeric(reg.model
      $coefficients)
  # mean errors
  second.pass.errors[row.names(second.pass.errors) %in% merged.data$PERMNO,colnames(
      second.pass.errors) == date] = as.numeric(reg.model$residuals)
}

T = nrow(second.pass.lambdas)
lambda.est = colMeans(second.pass.lambdas)
lambda.sd = sqrt(1/T^2 * colMeans((second.pass.lambdas - colMeans(second.pass.lambdas)
    )^2))

mean.errors = matrix(0, nrow=length(asset.set), ncol=1)
for (i in 1:length(asset.set)){
  permno = asset.set[i]
  mean.errors.i = second.pass.errors[row.names(second.pass.errors) == permno,
                                     second.pass.errors[row.names(second.pass.errors)
                                         == permno, ] != 0]
  mean.errors[i] = mean(as.numeric(mean.errors.i))

  if (i %% 1000 == 0){
    cat("iteration = ", i, "\n")
  }
}

second.pass.errors = as.matrix(second.pass.errors)
chi.mean.error = 0
for (i in 1:length(asset.set)){
#foreach(i=1:length(asset.set)) %dopar% {
  permno = asset.set[i]
#  mean.errors.i = second.pass.errors[row.names(second.pass.errors) == permno,
```

```
#                                         second.pass.errors[row.names(second.pass.errors)
    == permno, ] != 0]
#  dates.i = colnames(mean.errors.i)
  data.i=second.pass.errors[i,]
  i.start = min(which(data.i!=0))
  i.end = max(which(data.i!=0))
  for (j in i:length(asset.set)){
#  foreach(j=1:length(asset.set)) %dopar% {
    if (i == j){
      chi.mean.error = chi.mean.error + mean.errors[i]^2*as.numeric(1/length(data.i[i.
          start:i.end])^2 * sum((data.i[i.start:i.end]-mean(data.i[i.start:i.end]))^2))
    }
    else {
      permno.j = asset.set[j]
#      mean.errors.j = second.pass.errors[row.names(second.pass.errors) == permno.j,
#                                         second.pass.errors[row.names(second.pass.
    errors) == permno.j, ] != 0]
      data.j = second.pass.errors[j, ]
      j.start = min(which(data.j!=0))
      j.end = max(which(data.j!=0))
      start.date = max(i.start, j.start)
      end.date = min(i.end, j.end)
      #dates.j = colnames(mean.errors.j)
      #inter.dates = intersect(dates.i, dates.j)
      if (end.date > start.date){
        data.i.use = data.i[start.date:end.date]
        data.j.use = data.j[start.date:end.date]
        chi.mean.error = chi.mean.error + mean.errors[i]*mean.errors[j]*as.numeric(2*1
            /(end.date-start.date)^2 * sum((data.i.use-mean(data.i.use)) * (data.j.use-
            mean(data.j.use))))
      }
    }
  }

  if (i %% 100 == 0){
    cat("Main (i) iteration = ", i, "\n")
  }
}
```

assignment8.R