Replication of the Fama French 3 Factor Model and Relevance Today

Xin Tan
Daria Skidnova
Jeff Giddens

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center for Applied Statistics
and Economics
Humboldt–UniversitÃt zu Berlin
http://lvb.wiwi.hu-berlin.de
http://case.hu-berlin.de

Structure

What is the Fama French model?

The Fama French model is a model for explaining stock returns. It extends the classical Capital Asset Pricing Model (CAPM) by having additional factors.

$$R_i - R_F = \beta \cdot (R_M - R_F)$$

Fama and French 1993 introduces SMB (Small market cap Minus Big / Size) and HML (High book-to-market Minus Low / Value) to capture the observation that small capitalization and high book value to market value ("value" in contrast to "growth") stocks tend to outperform the market.

$$R_i - R_F = \beta_M \cdot (R_M - R_F) + \beta_S \cdot SMB + \beta_V \cdot HML$$

Replicating the 3-Factor Model

To check that we have implemented the Fama French model correctly, we try to replicate the results of table 6 of @Fama1993 which involves monthly return data of 25 value-weighted portfolios from July 1963 to December 1991.
The data set structures the data as 1 column of months (YYYYDD format) plus 25 columns of portfolio monthly returns. The first return column is SMALL (market cap) LoBM (low book-to-market / "growth"). The first 5 return columns are all small cap but with increasing book-to-market ratios. The last 5 return columns are all large cap with the last column being BIG (market cap) HiBM (high book-to-market / "value").
In reporting, results are structured in a matrix with rows representing market cap and columns for book to market ratios.

Fama and French 2015 adds RMW (Robust operating profit Minus Weak / Profitability) and CMA (Conservative investment strategy Minus Aggressive / Investment).

$$R_i - R_F = \beta_M \cdot (R_M - R_F) + \beta_S \cdot SMB + \beta_V \cdot HML + \beta_P \cdot RMW + \beta_I \cdot CMA$$

Fama French factors are calculated as return spreads between two portfolios, e.g. SMB is the difference between the return of a small cap portfolio and that of a large cap portfolio.
We choose the Fama French model due to the high quality data available at Kenneth R. French's data library
Refer to Wikepedia for more information.

Fama French Data

French's data library contains data for the factors, corresponding market returns and risk free rates, as well as the portfolios returns featured in the papers:

- 3 Factors 1926.07.01 to 2018.03.29 as daily / weekly / monthly data
- 5 Factors 1963.07.01 to 2018.03.29 as daily / monthly / yearly data
- 25 Portfolios (5x5) formed on Size and Book-to-Market 1926.07 to 2018.03 corresponding to the @Fama1993 3-factor setup (P24 Table 6).

The downloaded CSV data contains headers and footers that need to be removed before input to R.

S&P500 Stock Data

The BatchGetSymbols library has a function BatchGetSymbols()for downloading SP500 stock prices and volumes from a cached repository, thus avoiding problems when downloading large amount of data directly from Yahoo or Google (e.g. the getSymbols function from the quantmod library)

```
library(BatchGetSymbols)


\# Get company information incl. tickers for SP500 stocks
Companies <- GetSP500Stocks()


\# Batch download data from Yahoo Finance
Stocks<-BatchGetSymbols(tickers = Companies$tickers,
                        first.date = "2017-01-01",
                        last.date = "2017-12-31")
```

The downloaded list contains 2 dataframes:

- df.control contains descriptive information like whether the download for the ticker is successful.
- df.tickers contains the downloaded price data. Each row is the price data for one ticker at one date, hence we need to process the data into a format easier to work with.

(Use kable() function in Knitr library to format table output in PDF.) kable(head(Stocks$df.control, n = 3$))

ticker	src	download.status	total.obs	perc.benchmark.dates	t
MMM	yahoo	OK	251	1	
ABT	yahoo	OK	251	1	
ABBV	yahoo	OK	251	1	

kable(head(Stocks$df.tickers[, 1 : 5], n = 3$))

price.open	price.high	price.low	price.close	volume
178.83	180.00	177.22	178.05	2509300
178.03	178.90	177.61	178.32	1542000
178.26	179.14	176.89	177.71	1447800

kable(head(Stocks$df.tickers[, 6 : 10], n = 3$))

price.adjusted	ref.date	ticker	ret.adjusted.prices	ret.closing.pri
171.77	2017-01-03	MMM	NA	NA
172.03	2017-01-04	MMM	0.0015164	0.0015165
171.44	2017-01-05	MMM	-0.0034209	-0.0034208

Below code selects the downloaded tickers (marked by df.control$threshold.decision=="KEEP") and use the dates from 3M as the date column for dataframe SP500.data.

It reads stocks ticker by ticker and matches previous price series by date. The unmatched dates will have NAs. The new stock price series is merged into the dataframe as a new column with the ticker symbol as the column name.

We write the processed data to CSVs.

```
good.tickers <- Stocks\$df.control\$
        ticker[Stocks\$df.control\$threshold.decision=="KEEP"]


\# Fill dates as the first stock "MMM" happens to have complete dates
\# (column name = "date")
SP500.data<-data.frame(date = Stocks\$
                         df.tickers\$
                         ref.date[1:max(Stocks\$df.control\$total.obs)])


for(i in 1:length(good.tickers))
\{
  \# X is a temp dataframe that has 2 columns,
  \# 1st is date (for matching), 2nd is the actual data (e.g. closing price)
```

```
  \# Choose relevant data by matching tickers
  X <- data.frame(date =
        Stocks\$df.tickers\$ref.date[Stocks\$df.tickers\$ticker==good.tickers[i]],
        Stocks\$df.tickers\$price.adjusted[Stocks\$df.tickers\$ticker==good.tickers[i]])

  \# change the column name of X to be the ticker of the stock
  \# colnames(X)[2] = good.tickers[i] # this one don't work
  colnames(X)[2] <- Stocks$df.tickers$
        ticker[Stocks$df.tickers$ticker==good.tickers[i]]

  \# merge X as a new column into SP500.data by matching date
  \# missing dates will have NA by default
  SP500.data <- merge.data.frame(SP500.data, X, by = "date", all.x = TRUE)
\}
```

Simple Regression

readxl library for reading Excel data. The imported data would be stored as data.frame and must be unlist() into vectors for regression.
(data.frame is also a list in R)

```
library(readxl)
FF3<- read_excel("Data/FF3_196307-199112.xlsx")
# unlist: convert the data into vector format
rmrf<-unlist(FF3[,2])
```

OLS regression can be performed with two lines of code:

```
y <- lm(rirf ~ rmrf + smb + hml);
summary(y)
```

Call:

lm(formula = rirf   rmrf + smb + hml)

Residuals:

Min	$1Q$	*Median*	$3Q$	*Max*
-5.7407	-1.2491	-0.0457	1.2168	7.9848

Coefficients:

	Estimate	*Std.Error*	*tvalue*	$Pr(>\lvert t \rvert)$	
(*Intercept*)	-0.38175	0.10752	-3.550	0.000439	$***$
rmrf	1.03489	0.02638	39.226	$<2e^{-16}$	$***$
smb	1.39851	0.03928	35.607	$<2e^{-16}$	$***$
hml	-0.29792	0.04402	-6.768	$5.79e^{-11}$	$***$

$---$

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ‘ 1

Residual standard error: 1.94 on 338 degrees of freedom

Multiple R-squared: 0.9385, Adjusted R-squared: 0.9379

F-statistic: 1718 on 3 and 338 DF, p-value: $<2.2e^{-}16$

summary(y) contains the regression results and specific results could be obtained, e.g., via:
summary(y)\$coefficients which returns the regression betas and their standard errors, t-values and p-values in a matrix.
kable(summary(y)\$coefficients, digits=2)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept	-0.38	0.11	-3.55	0
rmrf	1.03	0.03	39.23	0
smb	1.40	0.04	35.61	0
hml	-0.30	0.04	-6.77	0

With the OLS regression code working, below code runs regression on each portfolio and saves the results in a list results.

```r
# Store summaries into a results list
results <- list()
# The first column of P25 is dates, not data
for(i in 1:(ncol(P25)-1))
{
  rirf<-unlist(P25[,i+1])-rf # Data starts from the 2nd col of P25
  y<-lm(rirf~rmrf+smb+hml)
  results[[i]]<-summary(y)
}
```

We then read out the results, stack them into corresponding vectors, then reshape them into the 5×5 format as in the paper for ease of comparison. Results are highly similar and we have not yet identify why they do not match exactly, perhaps due to rounding errors.

```
betas <- vector()
std.errors <- vector()
t.values <- vector()
R.squareds <- vector()
# save all betas
for(i in 1:(ncol(P25)-1))
{
  betas <- cbind(betas,results[[i]]$coefficients
    [,1])
  std.errors <- cbind(std.errors,results[[i]]$sigma)
  t.values <- cbind(t.values, results[[i]]
    $coefficients[,3])
  R.squareds <- cbind(R.squareds, results[[i]]$adj.r
    .squared)
}
```

```
# resize the output to 5x5 format like Fama French paper
resize <- function(x)
{
  df = data.frame(matrix(x, nrow=5, byrow = TRUE))
  colnames(df) = c("Low", "2", "3", "4", "High")
  rownames(df) = c("Small", "2", "3", "4", "Big")
  return(df)
}
# resize alpha
alpha <- resize(betas[1,])
kable(alpha, digits=2)
```

	Low	2	3	4	High
Small	-0.38	-0.10	-0.07	0.08	0.06
2	-0.13	-0.02	0.14	0.15	0.06
3	-0.04	0.11	-0.02	0.14	0.05
4	0.11	-0.16	0.01	0.08	0.04
Big	0.21	-0.02	-0.06	-0.06	-0.18

```
# resize beta
market.beta <- resize(betas[2,])
SMB.beta <- resize(betas[3,])
HML.beta <- resize(betas[4,])

# display beta below

kable(market.beta, digits=2)
```

	Low	2	3	4	High
Small	1.03	0.97	0.94	0.89	0.95
2	1.10	1.02	0.96	0.97	1.07
3	1.10	1.02	0.97	0.97	1.06
4	1.06	1.07	1.04	1.03	1.15
Big	0.96	1.02	0.96	1.01	1.03

kable(SMB.beta, digits=2)

	Low	2	3	4	High
Small	1.4	1.27	1.16	1.10	1.19
2	1.0	0.94	0.83	0.71	0.85
3	0.7	0.63	0.54	0.45	0.65
4	0.3	0.27	0.25	0.22	0.36
Big	-0.2	-0.19	-0.27	-0.19	-0.04

kable(HML.beta, digits=2)

	Low	2	3	4	High
Small	-0.30	0.08	0.27	0.38	0.62
2	-0.48	0.03	0.23	0.47	0.70
3	-0.43	0.04	0.31	0.50	0.71
4	-0.44	0.03	0.30	0.56	0.74
Big	-0.44	-0.02	0.20	0.56	0.76

Table 6

Regressions of excess stock and bond returns (in percent) on the excess market return ($RM-RF$) and the mimicking returns for the size (SMB) and book-to-market equity (HML) factors: July 1963 to December 1991, 342 months.[a]

$$R(t) - RF(t) = a + b[RM(t) - RF(t)] + sSMB(t) + hHML(t) + e(t)$$

Dependent variable: Excess returns on 25 stock portfolios formed on size and book-to-market equity

Size quintile	Book-to-market equity (BE/ME) quintiles									
	Low	2	3	4	High	Low	2	3	4	High
			b					$t(b)$		
Small	1.04	1.02	0.95	0.91	0.96	39.37	51.80	60.44	59.73	57.89
2	1.11	1.06	1.00	0.97	1.09	52.49	61.18	55.88	61.54	65.52
3	1.12	1.02	0.98	0.97	1.09	56.88	53.17	50.78	54.38	52.52
4	1.07	1.08	1.04	1.05	1.18	53.94	53.51	51.21	47.09	46.10
Big	0.96	1.02	0.98	0.99	1.06	60.93	56.76	46.57	53.87	38.61
			s					$t(s)$		
Small	1.46	1.26	1.19	1.17	1.23	37.92	44.11	52.03	52.85	50.97
2	1.00	0.98	0.88	0.73	0.89	32.73	38.79	34.03	31.66	36.78
3	0.76	0.65	0.60	0.48	0.66	26.40	23.39	21.23	18.62	21.91
4	0.37	0.33	0.29	0.24	0.41	12.73	11.11	9.81	7.38	11.01
Big	−0.17	−0.12	−0.23	−0.17	−0.05	−7.18	−4.51	−7.58	−6.27	−1.18
			h					$t(h)$		
Small	−0.29	0.08	0.26	0.40	0.62	−6.47	2.35	9.66	15.53	22.24
2	−0.52	0.01	0.26	0.46	0.70	−14.57	0.41	8.56	17.24	24.80
3	−0.38	−0.00	0.32	0.51	0.68	−11.26	−0.05	9.75	16.88	19.39
4	−0.42	0.04	0.30	0.56	0.74	−12.51	1.04	8.83	14.84	17.09
Big	−0.46	0.00	0.21	0.57	0.76	−17.03	0.09	5.80	18.34	16.24

Similarly for t-statistics and R^2:

```
# resize t-stats
market.t <-resize(t.values[2,])
SMB.t <- resize(t.values[3,])
HML.t <- resize(t.values[4,])

kable(market.t, digits=2)
```

	Low	2	3	4	High
Small	39.23	50.60	58.42	57.99	57.76
2	53.20	58.56	59.98	62.77	63.25
3	59.68	56.81	53.35	58.93	51.14
4	57.16	52.61	50.34	51.30	46.30
Big	57.20	56.98	42.80	55.04	37.70

kable(SMB.t, digits=2)

	Low	2	3	4	High
Small	35.61	44.82	48.65	48.10	48.63
2	32.62	36.36	34.80	30.78	33.82