Today, we will explore the rolling Fama French model and the explanatory power of the 3 factors in different time periods. In the financial world, we often look at rolling means, standard deviations and models to make sure we haven’t missed anything unusual, risky, or concerning during different market or economic regimes. Our portfolio returns history is for the years 2013 through 2017, which is rather a short history, but there still might a be a 24-month period where the Fama French factors were particularly strong, weak, or meaningless. We would like to unearth and hypothesize about what explains them or their future likelihood.

We will be working with our usual portfolio consisting of:

+ SPY (S&P500 fund) weighted 25%

+ EFA (a non-US equities fund) weighted 25%

+ IJS (a small-cap value fund) weighted 20%

+ EEM (an emerging-mkts fund) weighted 20%

+ AGG (a bond fund) weighted 10%

Before we can run a Fama French model for that portfolio, we need to find portfolio monthly returns, which was covered in [this post](https://rviews.rstudio.com/2017/10/11/from-asset-to-portfolio-returns/). I won’t go through the logic again but the code is here:

library(tidyquant)

library(tidyverse)

library(timetk)

symbols <- c("SPY","EFA", "IJS", "EEM","AGG")

prices <-

getSymbols(symbols, src = 'yahoo',

from = "2012-12-31",

to = "2017-12-31",

auto.assign = TRUE, warnings = FALSE) %>%

map(~Ad(get(.))) %>%

reduce(merge) %>%

`colnames<-`(symbols)

w <- c(0.25, 0.25, 0.20, 0.20, 0.10)

asset\_returns\_long <-

prices %>%

to.monthly(indexAt = "lastof", OHLC = FALSE) %>%

tk\_tbl(preserve\_index = TRUE, rename\_index = "date") %>%

gather(asset, returns, -date) %>%

group\_by(asset) %>%

mutate(returns = (log(returns) - log(lag(returns)))) %>%

na.omit()

portfolio\_returns\_tq\_rebalanced\_monthly <-

asset\_returns\_long %>%

tq\_portfolio(assets\_col = asset,

returns\_col = returns,

weights = w,

col\_rename = "returns",

rebalance\_on = "months")

We also need to import the Fama French factors and combine them into one object with our portfolio returns. We painstakingly covered this in the [previous post](https://rviews.rstudio.com/2018/04/11/introduction-to-fama-french/) and the code for doing so is here:

temp <- tempfile()

base <-

"http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/"

factor <-

"Global\_3\_Factors"

format<-

"\_CSV.zip"

full\_url <-

glue(base,

factor,

format,

sep ="")

download.file(

full\_url,

temp,

quiet = TRUE)

Global\_3\_Factors <-

read\_csv(unz(temp, "Global\_3\_Factors.csv"),

skip = 6) %>%

rename(date = X1) %>%

mutate\_at(vars(-date), as.numeric) %>%

mutate(date =

rollback(ymd(parse\_date\_time(date, "%Y%m") + months(1)))) %>%

filter(date >=

first(portfolio\_returns\_tq\_rebalanced\_monthly$date) & date <=

last(portfolio\_returns\_tq\_rebalanced\_monthly$date))

ff\_portfolio\_returns <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

left\_join(Global\_3\_Factors, by = "date") %>%

mutate(MKT\_RF = Global\_3\_Factors$`Mkt-RF`/100,

SMB = Global\_3\_Factors$SMB/100,

HML = Global\_3\_Factors$HML/100,

RF = Global\_3\_Factors$RF/100,

R\_excess = round(returns - RF, 4))

We now have one data frame ff\_portfolio\_returns that holds our Fama French factors and portfolio returns. Let’s get to the rolling analysis.

We first define a rolling model with the rollify() function from tibbletime. However, instead of wrapping an existing function, such as kurtosis() or skewness(), we will pass in our linear Fama French model.

# Choose a 24-month rolling window

window <- 24

library(tibbletime)

# define a rolling ff model with tibbletime

rolling\_lm <-

rollify(.f = function(R\_excess, MKT\_RF, SMB, HML) {

lm(R\_excess ~ MKT\_RF + SMB + HML)

}, window = window, unlist = FALSE)

Next, we pass columns from ff\_portfolio\_returns to the rolling function model.

rolling\_ff\_betas <-

ff\_portfolio\_returns %>%

mutate(rolling\_ff =

rolling\_lm(R\_excess,

MKT\_RF,

SMB,

HML)) %>%

slice(-1:-23) %>%

select(date, rolling\_ff)

head(rolling\_ff\_betas, 3)

# A tibble: 3 x 2

date rolling\_ff

1 2014-12-31

2 2015-01-31

3 2015-02-28

We now have a new data frame called rolling\_ff\_betas, in which the column rolling\_ff holds an S3 object of our model results. We can tidy() that column with map(rolling\_ff, tidy) and then unnest() the results, very similar to [our CAPM](https://rviews.rstudio.com/2018/02/08/capm-beta/) work, except we have more than one independent variable.

rolling\_ff\_betas <-

ff\_portfolio\_returns %>%

mutate(rolling\_ff =

rolling\_lm(R\_excess,

MKT\_RF,

SMB,

HML)) %>%

mutate(tidied = map(rolling\_ff,

tidy,

conf.int = T)) %>%

unnest(tidied) %>%

slice(-1:-23) %>%

select(date, term, estimate, conf.low, conf.high) %>%

filter(term != "(Intercept)") %>%

rename(beta = estimate, factor = term) %>%

group\_by(factor)

head(rolling\_ff\_betas, 3)

# A tibble: 3 x 5

# Groups: factor [3]

date factor beta conf.low conf.high

1 2014-12-31 MKT\_RF 0.931 0.784 1.08

2 2014-12-31 SMB -0.0130 -0.278 0.252

3 2014-12-31 HML -0.160 -0.459 0.139

We now have rolling betas and confidence intervals for each of our 3 factors. Let’s apply the same code logic and extract the rolling R-squared for our model. The only difference is we call glance() instead of tidy().

rolling\_ff\_rsquared <-

ff\_portfolio\_returns %>%

mutate(rolling\_ff =

rolling\_lm(R\_excess,

MKT\_RF,

SMB,

HML)) %>%

slice(-1:-23) %>%

mutate(glanced = map(rolling\_ff,

glance)) %>%

unnest(glanced) %>%

select(date, r.squared, adj.r.squared, p.value)

head(rolling\_ff\_rsquared, 3)

# A tibble: 3 x 4

date r.squared adj.r.squared p.value

1 2014-12-31 0.898 0.883 4.22e-10

2 2015-01-31 0.914 0.901 8.22e-11

3 2015-02-28 0.919 0.907 4.19e-11

We have extracted rolling factor betas and the rolling model R-squared, now let’s visualize.

**Visualizing Rolling Fama French**

We start by charting the rolling factor betas with ggplot(). This gives us an view into how the explanatory power of each factor has changed over time.

rolling\_ff\_betas %>%

ggplot(aes(x = date,

y = beta,

color = factor)) +

geom\_line() +

labs(title= "24-Month Rolling FF Factor Betas",

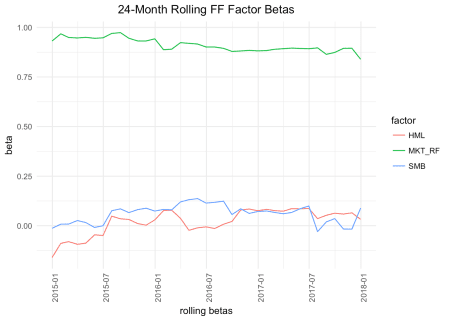
x = "rolling betas") +

scale\_x\_date(breaks = scales::pretty\_breaks(n = 10)) +

theme\_minimal() +

theme(plot.title = element\_text(hjust = 0.5),

axis.text.x = element\_text(angle = 90))



The rolling factor beta chart reveals some interesting trends. Both SMB and HML have hovered around zero, while the MKT factor has hovered around 1. That’s consistent with our plot of betas with confidence intervals from last time.

Next, let’s visualize the rolling R-squared with highcharter.

We first convert rolling\_ff\_rsquared to xts, using the tk\_xts() function.

rolling\_ff\_rsquared\_xts <-

rolling\_ff\_rsquared %>%

tk\_xts(date\_var = date, silent = TRUE)

Then pass the xts object to a highchart(type = "stock") code flow, adding the rolling R-squared time series with hc\_add\_series(rolling\_ff\_rsquared\_xts$r.squared...).

highchart(type = "stock") %>%

hc\_add\_series(rolling\_ff\_rsquared\_xts$r.squared,

color = "cornflowerblue",

name = "r-squared") %>%

hc\_title(text = "Rolling FF 3-Factor R-Squared") %>%

hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE) %>%

hc\_exporting(enabled = TRUE)

That chart looks choppy, but the R-squared never really left the range between .9 and .95. We can tweak the minimum and maximum y-axis values for some perspective.

highchart(type = "stock") %>%

hc\_add\_series(rolling\_ff\_rsquared\_xts$r.squared,

color = "cornflowerblue",

name = "r-squared") %>%

hc\_title(text = "Rolling FF 3-Factor R-Squared") %>%

hc\_yAxis( max = 2, min = 0) %>%

hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE) %>%

hc\_exporting(enabled = TRUE)

Ah, when the y-axis is zoomed out a bit, our R-squared looks consistently near 1 for the life of the portfolio.

That’s all for today. Thanks and see you next time!