Multi-Asset Principal Component Regression using RcppParallel

Jason Foster
CRAN - Package 'roll'

May 21, 2016

CRAN - Package 'roll'

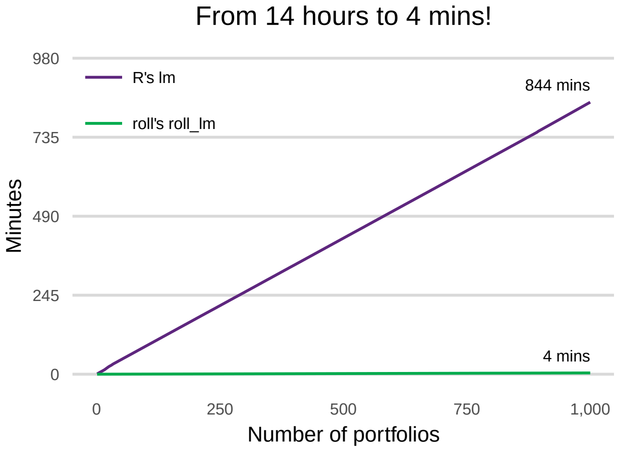
Functions use RcppParallel to parallelize rolling statistics of time-series data. Install the latest version from CRAN:

```
install.packages("roll")
```

Function	Description
roll_mean	Rolling means
roll_var	Rolling variances
roll_sd	Rolling standard deviations
roll_cov	Rolling covariance matrices
roll_cor	Rolling correlation matrices
roll_lm	Rolling linear models
roll_eigen	Rolling eigenvalues and eigenvectors
roll_pcr	Rolling principal component regressions
roll_vif	Rolling variance inflation factors

Roll the Fama-French model for 1,000 portfolios!

Full data set includes daily factor returns from Jul-1926 to Mar-2016. A daily rolling window requires over 70 million calculations for 1,000 portfolios!



Tested with 4 cores (8 threads) on a 2.5GHz laptop

R version

Download factor returns using Quandl:

```
library(Quandl)
factors <- Quandl("KFRENCH/FACTORS_D")
```

Start with plain R and estimate a portfolio's sensitivity to each factor by rolling the `lm` function:

```
lm_coef <- function(returns) {
  return(lm(formula = Portfolio ~., data = returns)$coef)
}

r_lm_coef <- rollapply(data = returns, width = 252,
                       FUN = lm_coef, by.column = FALSE,
                       align = "right")
```

RcppParallel version

Repeat the exercise and take advantage of parallel processing in C++ by using the **roll_lm** function in the **roll** package:

```
library(roll)
roll_lm_coef <- roll_lm(x = returns[ , factors],
                        y = returns[ , portfolio],
                        width = 252)$coefficients
```

And test that the coefficients from the **lm** and **roll_lm** functions are equal:

```
all.equal(r_lm_coef, roll_lm_coef)
```

```
## [1] TRUE
```

Here's an excerpt from BlackRock's Global Investment Outlook (Apr-2016):

Theme 3: Volatility and Dispersion

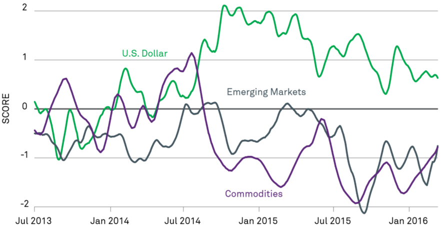
Markets today are characterized by a lot of "me-too" trades. Many investors have piled into similar strategies. Trends have been persistent — and counting on yesterday's winners rising (or falling) further has often paid off. Popular trades have included overweighting the U.S. dollar and underweighting EM and commodity assets. **See the *Copycats* chart.**

We see two problems with this picture. First, many of these trades are highly correlated. This means portfolios may be riskier than they appear. Second, monetary policy normalization is likely to increase volatility, we believe. This raises the risk of rapid momentum reversals and shifts in market leadership. Positioning in popular trades has moderated from recent peaks. This has coincided with a slowing of the U.S. dollar's rise and signs of stabilization in EM economies.

Gold, inflation-linked bonds, government debt and currency exposures can be useful portfolio hedges for volatility spikes.

COPYCATS

Crowded Positions, 2013–2016



Source: BlackRock Investment Institute, March 2016. Notes: Data are based on BlackRock analysis of portfolio flows, reported positions by fund managers and price momentum. A positive score means investors are overweight the asset class; a negative score indicates the reverse. The emerging markets line is based on an average of emerging market currency and equities positioning. Commodities are based on an average of energy and industrial metals.

Full report: https://www.blackrock.com/investing/insights/blackrock-investment-institute/outlook

Style analysis of multi-asset portfolios

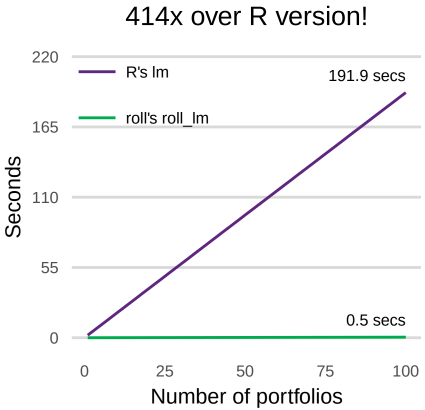
Download daily returns for 100 portfolios in the Tactical Allocation category and apply a 15 asset-class factor model on a daily rolling window:

Stocks	Bonds & Other
US Large Cap Value	US Gov 1-10Y
US Large Cap Growth	US Gov 10Y+
US Mid Cap	Non-US Gov
US Small Cap	IG Corp
Europe	HY Corp
Japan	MBS
Pacific ex Japan	Commodities
Emerging Markets	

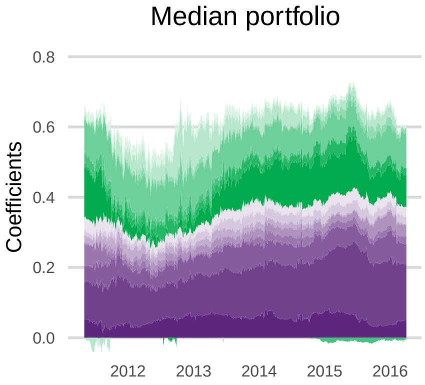
Sharpe, William F. 1988. "Determining a Fund's Effective Asset Mix." *Investment Management Review*.

Identify trends in the markets

Daily returns are readily available and help make inferences about style:



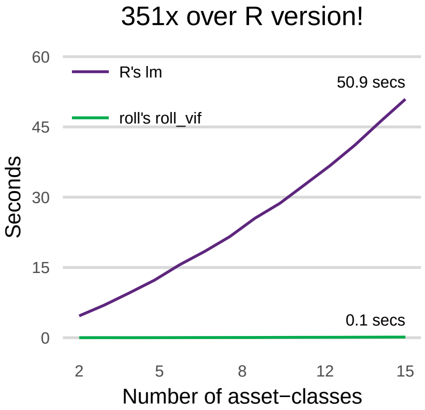
Speed test uses univariate betas

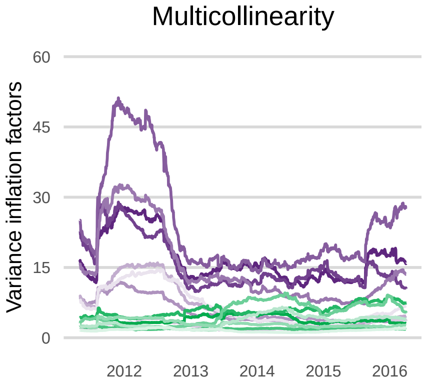


Purple *areas are stocks and* ***green*** *areas are bonds & other*

What about market structure?

Asset classes can be strongly correlated to one another:

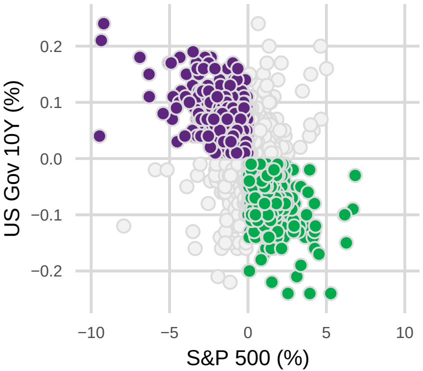




Purple *areas are stocks and* ***green*** *areas are bonds & other*

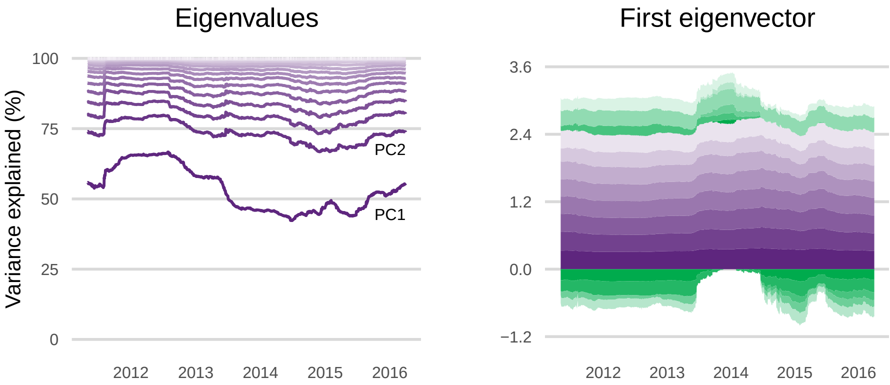
Risk-on, risk-off

- A market environment that is driven by changes in investor risk tolerance is often described as "**risk-on**, **risk-off**"
- That is, during periods of low (high) risk appetite, investors gravitate towards higher (lower) risk investments



Principal component analysis

The **`roll_eigen`** function shows a change in direction during the Taper Tantrum, a scenario where both equities and bonds sold-off:



Purple *areas are stocks and* ***green*** *areas are bonds & other*

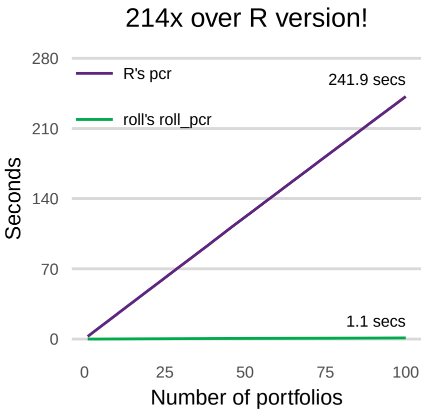
Principal component regression

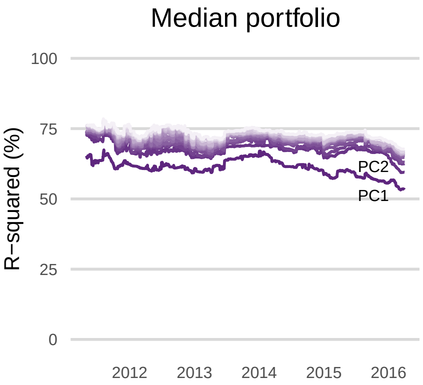
Principal component regression is a regression technique that uses principal component analysis:

1. Perform principal component analysis on the correlation matrix of asset-class returns
2. Select a subset of principal components and regress against a multi-asset portfolio's returns
3. Transform coefficients back to get the dimension equal to the number of asset-classes

Explanatory power by principal component

Explain the variance in each multi-asset portfolio's returns using principal component regression:





Speed test uses univariate betas

R version

The R implementation of principal component regression by rolling the **pcr** function is given below:

```
library(pls)
pcr_coef <- function(returns) {
  return(coef(pcr(formula = Portfolio ~., data = returns,
                  ncomp = 1, scale = TRUE),
              intercept = TRUE))
}

r_pcr_coef <- rollapply(data = returns, width = 252,
                        FUN = pcr_coef, by.column = FALSE,
                        align = "right")
```

RcppParallel version

Repeat the exercise and take advantage of parallel processing in C++ by using the **roll_pcr** function in the **roll** package:

```
library(roll)
roll_pcr_coef <- roll_pcr(x = returns[ , factors],
                          y = returns[ , portfolio],
                          width = 252, comps = 1,
                          scale_x = TRUE)$coefficients
```

And test that the coefficients from the **pcr** and **roll_pcr** functions are equal:

```
all.equal(r_pcr_coef, roll_pcr_coef)
```

```
## [1] TRUE
```

CRAN - Package ‘roll’

Get the released version from CRAN:

```r
install.packages("roll")
```

Or the development version from GitHub:

```r
# install.packages("devtools")
devtools::install_github("jjf234/roll")
```