Multi-Asset Principal Component Regression using RcppParallel

Jason Foster
CRAN - Package ‘roll’

May 21, 2016
Functions use RcppParallel to parallelize rolling statistics of time-series data. Install the latest version from CRAN:

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>roll_mean</td>
<td>Rolling means</td>
</tr>
<tr>
<td>roll_var</td>
<td>Rolling variances</td>
</tr>
<tr>
<td>roll_sd</td>
<td>Rolling standard deviations</td>
</tr>
<tr>
<td>roll_cov</td>
<td>Rolling covariance matrices</td>
</tr>
<tr>
<td>roll_cor</td>
<td>Rolling correlation matrices</td>
</tr>
<tr>
<td>roll_lm</td>
<td>Rolling linear models</td>
</tr>
<tr>
<td>roll_eigen</td>
<td>Rolling eigenvalues and eigenvectors</td>
</tr>
<tr>
<td>roll_pcr</td>
<td>Rolling principal component regressions</td>
</tr>
<tr>
<td>roll_vif</td>
<td>Rolling variance inflation factors</td>
</tr>
</tbody>
</table>
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!

From 14 hours to 4 mins!

Tested with 4 cores (8 threads) on a 2.5GHz laptop
Download factor returns using Quandl:

```r
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:

```r
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")
```
Repeat the exercise and take advantage of parallel processing in C++ by using the `roll_lm` function in the `roll` package:

```r
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:

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

```
## [1] TRUE
```
Crowded positions

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.*

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:

<table>
<thead>
<tr>
<th>Stocks</th>
<th>Bonds &amp; Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Large Cap Value</td>
<td>US Gov 1-10Y</td>
</tr>
<tr>
<td>US Large Cap Growth</td>
<td>US Gov 10Y+</td>
</tr>
<tr>
<td>US Mid Cap</td>
<td>Non-US Gov</td>
</tr>
<tr>
<td>US Small Cap</td>
<td>IG Corp</td>
</tr>
<tr>
<td>Europe</td>
<td>HY Corp</td>
</tr>
<tr>
<td>Japan</td>
<td>MBS</td>
</tr>
<tr>
<td>Pacific ex Japan</td>
<td>Commodities</td>
</tr>
<tr>
<td>Emerging Markets</td>
<td></td>
</tr>
</tbody>
</table>

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:

351x over R version!

Multicollinearity

Purple areas are stocks and green areas are bonds & other

Variance inflation factors

Number of asset-classes

Seconds

R's lm
roll's roll_vif

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:

**Eigenvalues**

- PC1: Variance explained (%)
- PC2: Variance explained (%)

**First eigenvector**

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

214x over R version!

Speed test uses univariate betas
The R implementation of principal component regression by rolling the `pcr` function is given below:

```r
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:

```r
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:

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

## [1] TRUE
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")
```