

Implementation of Value Strategies using R

Dirk Hugen¹

¹Department of Statistics and Actuarial Science
University of Iowa

May 19, 2016

- The SAS software package has historically been the main tool for academic and industrial analysis of financial data.
- In recent years R and Postgres have developed functionality that allows the researcher to replicate much of functionality of SAS.
- In this talk I will discuss how I have created and populated a Postgres using R as well as implement some value investment strategies using R, PostgreSQL, the RPostgreSQL package and the PL/R extension to PostgreSQL.

Having installed R, PostgreSQL, and the RPostgreSQL package, we load RPostgreSQL and connect to the wrds database on the local host. Define the PROD function to multiply across monthly returns to produce a cumulative return.

```
> library(RPostgreSQL)
> con<-dbConnect(PostgreSQL(),
+               dbname="wrds",
+               host="localhost")
> query<-"
+ CREATE AGGREGATE PROD(double precision)
+ ( SFUNC = float8mul, STYPE = double precision) "
> res<-dbSendQuery(con,query)
> res
<PostgreSQLResult: (589, 0, 1)>
```

Greenblatt Value Formula

In *The Little Book that Beats the Market*, Joel Greenblatt outlines a two factor value strategy in which the best stocks in terms of earnings yield and operating cashflow yield are selected. In the following query we select those firms with earnings yield at least 10% and operating cashflow yield of at least 20%.

Greenblatt Magic Formula

```
> query <- "  
+ SELECT tsymbol, PROD(1+ret)  
+ FROM msf  
+ WHERE date BETWEEN '2011-01-01' AND '2013-12-31'  
+ AND tsymbol IN (  
+   SELECT tic  
+   FROM funda  
+   WHERE fyear = 2010  
+   AND (sich < 6000 OR sich > 6999)  
+   AND seq > 1000  
+   AND ni/(prcc_f*csho) > .1  
+   AND ni/(prcc_f*csho) IS NOT NULL  
+   AND oancf/(csho*prcc_f) > 0.2  
+   AND oancf/(csho*prcc_f) IS NOT NULL  
+   AND fic = 'USA')  
+ GROUP BY tsymbol"
```

Greenblatt Value Formula

```
> dbGetQuery(con, query)
  tsymbol      prod
1      MU 2.7119699
2      GCI 2.1660894
3      WDC 2.5471961
4      EIX 1.3135508
5      UFS 1.3237531
6      STR 1.4569846
7     SKYW 0.9848063
8      TER 1.2549858
9      CHK 1.0924886
10     ALK 2.6042898
11     OSK 1.4340242
12     VSH 0.9032697
13     ETR 1.0377337
14      T 1.4051001
15     TSN 2.0049275
```

O'Schaughnessy PSR strategy

In *What Works on Wall Street*, James O'Schaughnessy outlines a price to sales (PSR) strategy in which during a given year the strategy picks the 50 stocks with the lowest price to sales ratio and market cap with at least 200 million.

O'Schaughnessy PSR strategy

```
> query<-"
+ SELECT tsymbol, PROD(1+ret)
+ FROM msf
+ WHERE date BETWEEN '2012-01-01' AND '2012-12-31'
+ AND tsymbol IN
+ (SELECT tic FROM
+ (SELECT gvkey, tic, sale/(prcc_f*csho) AS psr
+ FROM funda
+ WHERE fyear = 2012
+ AND (sich < 6000 OR sich > 6999)
+ AND fic = 'USA'
+ AND seq > 200
+ AND sale IS NOT NULL
+ AND prcc_f*csho IS NOT NULL
+ ORDER BY psr LIMIT 50) f2)
+ GROUP BY tsymbol"
```


This query returns 50 firms, so we cannot list them all on one slide, but we can calculate the mean and show a histogram of returns.

```
> res<-dbGetQuery(con, query)
> mean(res$prod)
[1] 1.379376
```

- Many times instead of querying for a enterprise yield of over a set cut-off, we wish query for firms with enterprise yield over a percentile. For example, we may wish to query for all firms that have a debt-equity ratio of no greater than the 20th percentile.
- Postgres does not (yet) include such functions natively, so we must find a way to include outside functions in the `SELECT` query.
- The PL/R extension to PostgreSQL, developed by Joe Conway, provides this bridge. See www.joeconway.com for more detail.



PL/R gives the framework to declare Postgres stored procedures containing R code. Working within the `wrds` database on the local host, with the PL/R package installed, we declare the `plr` extension, embed the R function `quantile()` with the `prob` argument hard wired to 0.2 and pass to it an array.

```
wrds=# CREATE EXTENSION plr;
wrds=# CREATE FUNCTION r_quantile(float8[])
wrds-# RETURNS float8 AS '
wrds'# quantile(arg1,probs=0.2,names=FALSE)
wrds'# ' LANGUAGE 'plr';
CREATE FUNCTION
```

Let us now query for the 20th percentile of revenue in fiscal year 2010. We see that the firm at the 20th percentile had total revenue (sales) of 4.91 million dollars and observe that a full fifth of all listed firms are quite small.

```
> query <- "SELECT quantile(sale) FROM funda WHERE  
> res<-dbGetQuery(con,query)  
> res  
  quantile  
1    4.9186
```

References

-  Bennett, M. and D. Hugen, 2016. *Financial Analytics in R: Building a Laptop Laboratory for Data Science*. Cambridge University Press.
-  Conway, Joe. *R Functionality within PostgreSQL*
www.joeconway.com
-  Edelbuettel, D. et al. *Package RPostgreSQL*
www.r-project.org
-  Greenblatt, J. 2004. *The Little Book That Beats the Market*. John Wiley and Sons.
-  O'Schaughnessy, J. 1998. *What Works on Wall Street*. McGraw Hill.