

yfscreen: Yahoo Finance Screener in R and Python

Open Source Quantitative Finance 2025

Jason Foster

April 12, 2025

Open-source landscape

- Predefined screeners using yahooquery:

```
from yahooquery import Screener
s = Screener()
s.get_screeners(["most_actives", "day_gainers"])
```

- Equity screeners using openbb (*predefined screeners also supported*):

```
from openbb import obb
obb.equity.screener(provider = "yfinance", country = "us")
```

- Equity and fund screeners using yfinance (*support for predefined screeners also available*):

```
import yfinance as yf
from yfinance import EquityQuery, FundQuery
q = EquityQuery("eq", ["region", "us"])
yf.screen(q)
```

Overview

yfscreen is a package that provides simple and efficient access to Yahoo Finance's screener functionality for querying and retrieval of financial data:

Abstracts API complexities:

- **Session management**
- **Crumb and cookie handling**
- **Query construction**
- **Pagination**
- **JSON payload generation**

Coverage across security types:

- **Equities**
- **Mutual funds**
- **ETFs**
- **Indices**
- **Futures**

Available in both R and Python!

Installation in R

- Install the released version from CRAN:

```
install.packages("yfscreen")
```

- Or the development version from GitHub:

```
# install.packages("devtools")  
devtools::install_github("jasonjfoster/screen/r")
```

- Then load the package:

```
library(yfscreen)
```

Installation in Python

- Install the released version from PyPI:

```
pip install yfscreen
```

- Or the development version from GitHub:

```
pip install \
git+https://github.com/jasonjfoster/\
screen.git@main#subdirectory=python
```

- Then import the package:

```
import yfscreen as yfs
```

Available filters

Load the package and explore the available filter options using `yfscreen::data_filters` in R or `yfs.data_filters` in Python:

Security types

- equity
- mutualfund
- etf
- index
- future

Data types

- Market Data
- Ratios
- Performance
- :
- Technicals

Field names

- eodprice
- exchange
- intradayprice
- :
- ticker

Create query

Build a query using filters with a comparison operator ("gt", "lt", "eq", or "btwn"), a field name, and value(s); "btwn" requires two values:

```
most_actives <- list(  
  list("eq", list("region", "us")),  
  list("btwn", list("intradaymarketcap", 2e9, 1e10)),  
  list("gt", list("dayvolume", 5e6))  
)  
query <- yfscreen::create_query(most_actives)
```

```
most_actives = [  
  ["eq", ["region", "us"]],  
  ["btwn", ["intradaymarketcap", 2e9, 1e10]],  
  ["gt", ["dayvolume", 5e6]]  
]  
query = yfs.create_query(most_actives)
```

Get data

Retrieve data from the Yahoo Finance API using a payload created with `create_query()` and `create_payload()`:

R

```
payload <- yfscreen::create_payload("equity", query)
data <- yfscreen::get_data(payload)
```

Python

```
payload = yfs.create_payload("equity", query)
data = yfs.get_data(payload)
```

View data

R

```
head(data[, c("symbol", "regularMarketPrice.raw",  
              "regularMarketChangePercent.raw", ...)])
```

Python

```
data[["symbol", "regularMarketPrice.raw",  
      "regularMarketChangePercent.raw", ...]].head()
```

symbol	price	chg (%)	vol (m)	mkt cap (b)	...
GME	22.09	-22.11%	95.80	9.88	...
IPG	26.40	-0.34%	6.44	9.84	...
WBA	11.16	0.54%	9.86	9.65	...
SARO	28.04	1.08%	8.73	9.38	...
RKLB	18.42	-5.25%	10.11	8.35	...

More filters in R

- Equities with earnings growth above 25% and relatively low P/E and PEG ratios:

```
sec_type <- "equity"
undervalued_growth_stocks <- list(
  list("btwn", list("peratio.lasttwelvemonths", 0, 20)),
  list("lt", list("pegratio_5y", 1)),
  list("gt", list("epsgrowth.lasttwelvemonths", 25))
)
```

- Mutual funds with a performance rating of 4 or 5:

```
sec_type <- "mutualfund"
top_mutual_funds <- list(
  list("gt", list("intradayprice", 15)),
  list("eq", list("performanceratingoverall", 4)),
  list("eq", list("performanceratingoverall", 5))
)
```

More filters in Python

- Equities with the largest gains during the trading day:

```
sec_type = "equity"  
day_gainers = [  
    ["gt", ["percentchange", 3]],  
    ["gt", ["intradaymarketcap", 2e9]],  
    ["gt", ["intradayprice", 5]],  
    ["gt", ["dayvolume", 1.5e3]]  
]
```

- Equities with the largest losses during the trading day:

```
day_losers = [  
    ["lt", ["percentchange", -2.5]],  
    ["gt", ["intradaymarketcap", 2e9]],  
    ["gt", ["intradayprice", 5]],  
    ["gt", ["dayvolume", 2e3]]  
]
```

Performance and positioning

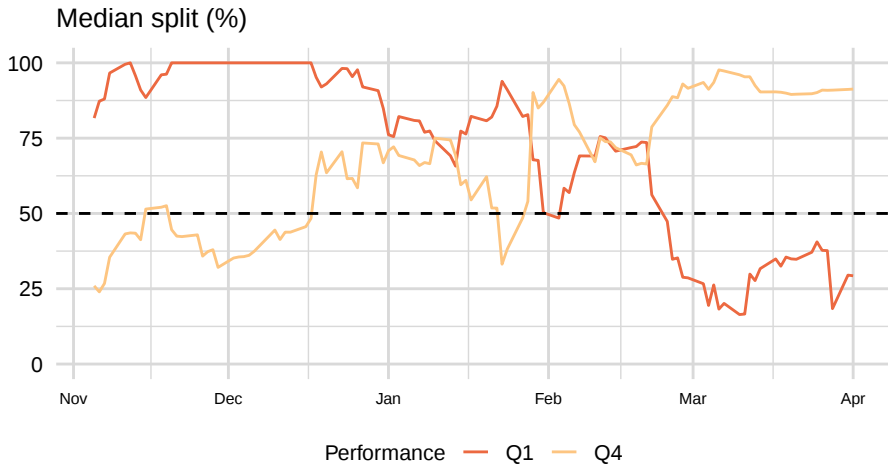
Estimate asset allocations of "Tactical Allocation" mutual funds via "categoryname" using constrained least squares:

$$\begin{array}{ll}\min & \|y - X\beta\|^2 \\ \text{s.t.} & \sum \beta_i = 1 \\ & 0 \leq \beta_i \leq 1\end{array}$$

Then separate top and bottom performing managers to analyze how positioning contributed to recent performance

Tactical allocation

Chart shows regression-based market and cash split for top (Q1) and bottom (Q4) performing managers since the U.S. election in Nov 2024:



Thank you!

Contact me at [jason.j.foster at gmail.com](mailto:jason.j.foster@gmail.com) and visit the following resources for more information:

- GitHub: <https://github.com/jasonjfoster>
- GitHub Pages: <https://jasonjfoster.github.io>
- r-universe: <https://jasonjfoster.r-universe.dev>