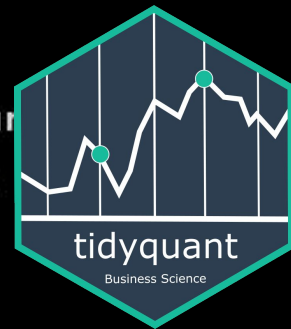


```
3 library(tidyquant)
4
5 start <- "2014-01-01" %>% ymd()
6 end   <- start + years(5) - days(1)
7
8 Ra <- c("AAPL", "GOOG", "NFLX") %>%
9   tq_get(get = "stock.prices",
10         from = start,
```



# Finance with

**Performance Analysis and  
Portfolio Optimization** using **tidyquant**

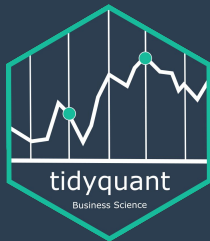


Matt Dancho  
*R/Finance 2019*

# About Me

- **Founder**  
(Business Science)

- **Creator**  
(tidyquant)



- **Educator**  
(I teach DS4B)



**Matt Dancho**

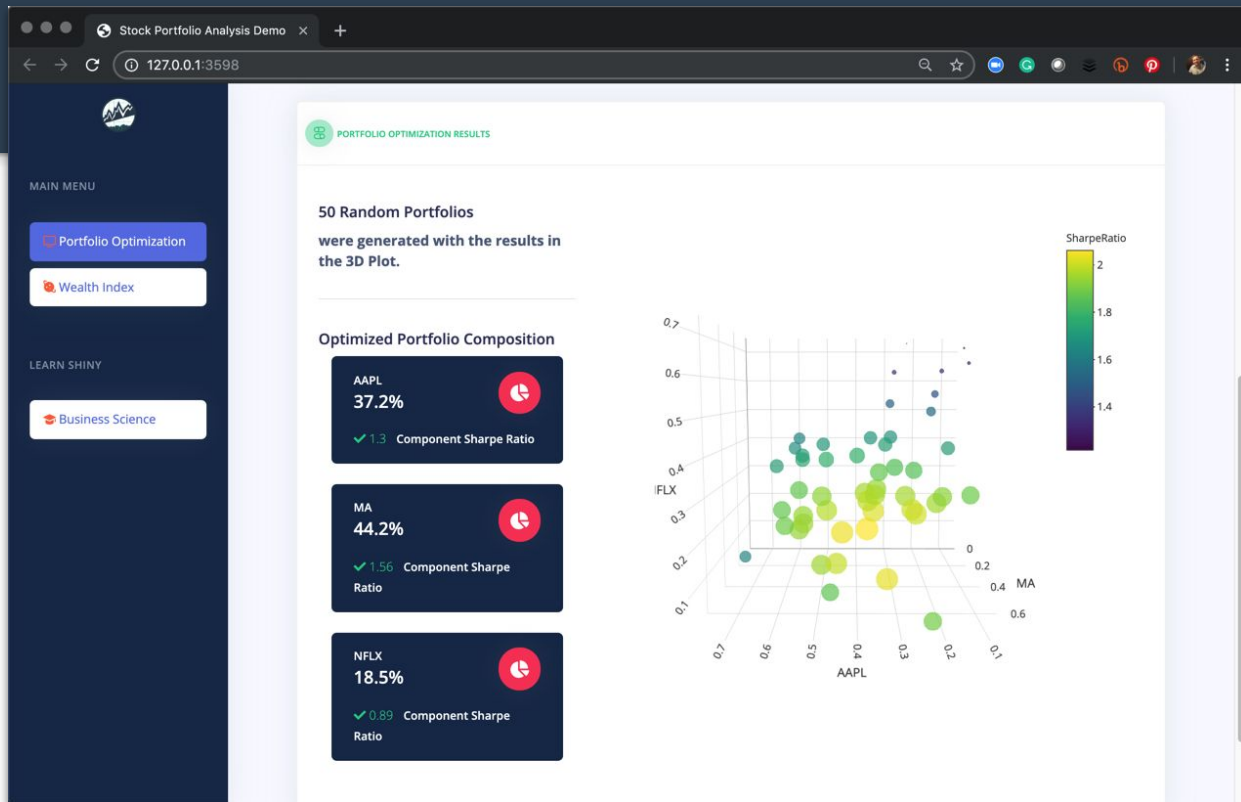
Founder of Business Science, Matt designs and executes educational courses and workshops that deliver immediate value to organizations. His passion is **up-leveling future data scientists** coming from **untraditional backgrounds**.



# Demo

## Portfolio Optimization App

- Shiny
- tidyquant
- argonDash



# What is tidyquant?

And how do we use it?



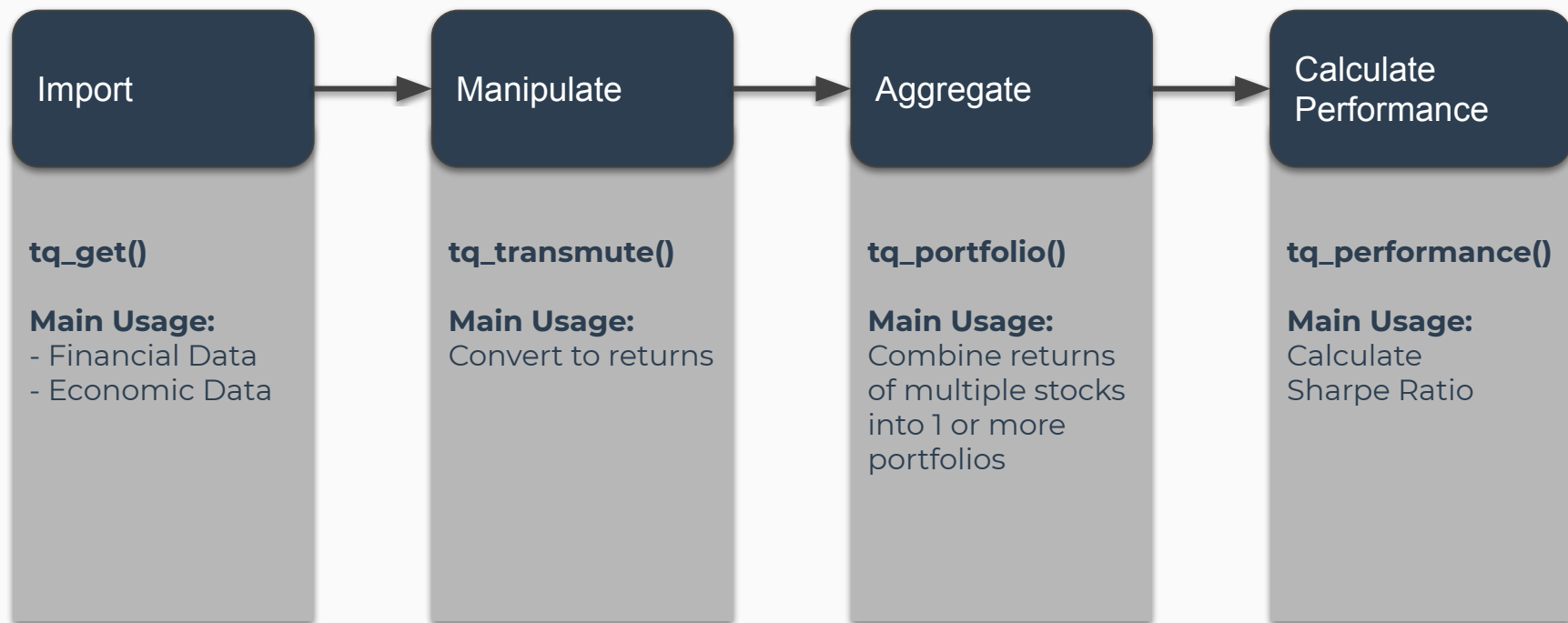
- Financial Software
- 113,000 Downloads (9,462/mo)
- Used by Many Financial Companies
- It's a Workflow

The screenshot shows the Tidyquant website documentation page. The header includes the 'tidyquant' logo, version '0.5.6', and navigation links: Home, Function Reference, Vignettes, and News. The main content area features the title 'tidyquant' with a subtitle 'Bringing financial analysis to the tidyverse'. Below this is a paragraph describing the package's integration with tidyverse and financial data sources. A section titled '2-Minutes To Tidyquant' includes a link to a YouTube video. A diagram illustrates the workflow between 'quantmod/xts packages' (listing quantmod, xts, zoo, PerformanceAnalytics, and TTR) and 'tidyverse packages' (listing dplyr, tidyr, ggplot2, purrr, and more), with 'tidyquant' acting as the bridge. The right sidebar contains links for downloading from CRAN, browsing source code on GitHub, reporting bugs, and viewing the license. The footer mentions a 'Software Intro Series' on YouTube.

Documentation: [business-science.github.io/tidyquant](https://business-science.github.io/tidyquant/)



## API that manages the process of **systematically** evaluating stocks & portfolios





Import

**tq\_get()**

**Main Usage:**

- Financial Data
- Economic Data

Manipulate

**tq\_transmute()**

**Main Usage:**

Convert to returns

Aggregate

**tq\_portfolio()**

**Main Usage:**

Combine returns  
of multiple stocks  
into 1 or more  
portfolios

Calculate  
Performance

**tq\_performance()**

**Main Usage:**

Calculate  
Sharpe Ratio



## Get Data

```
1 library(tidyquant)
2
3 # Stock Prices
4 tq_get("AAPL", from = "2018-01-01", to = "2018-12-31")
5
6 # FRED Economic Data
7 tibble(symbol = c("MKTGDP CNA646NWDB"),
8         name   = c("GDP China")) %>%
9   tq_get(get = "economic.data", from = "1960-01-01")
10
```

Connects To  
**Yahoo - Bloomberg - FRED - Quandl - more!**





## Stock Data

```
> tq_get("AAPL", from = "2018-01-01", to = "2018-12-31")
# A tibble: 250 x 7
```

	date	open	high	low	close	volume	adjusted
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2018-01-02	170.	172.	169.	172.	25555900	169.
2	2018-01-03	173.	175.	172.	172.	29517900	169.
3	2018-01-04	173.	173.	172.	173.	22434600	170.
4	2018-01-05	173.	175.	173.	175.	23660000	172.
5	2018-01-08	174.	176.	174.	174.	20567800	171.
6	2018-01-09	175.	175.	173.	174.	21584000	171.
7	2018-01-10	173.	174.	173.	174.	23959900	171.
8	2018-01-11	175.	175.	174.	175.	18667700	172.
9	2018-01-12	176.	177.	176.	177.	25226000	174.
10	2018-01-16	178.	179.	176.	176.	29565900	173.

```
# ... with 240 more rows
```

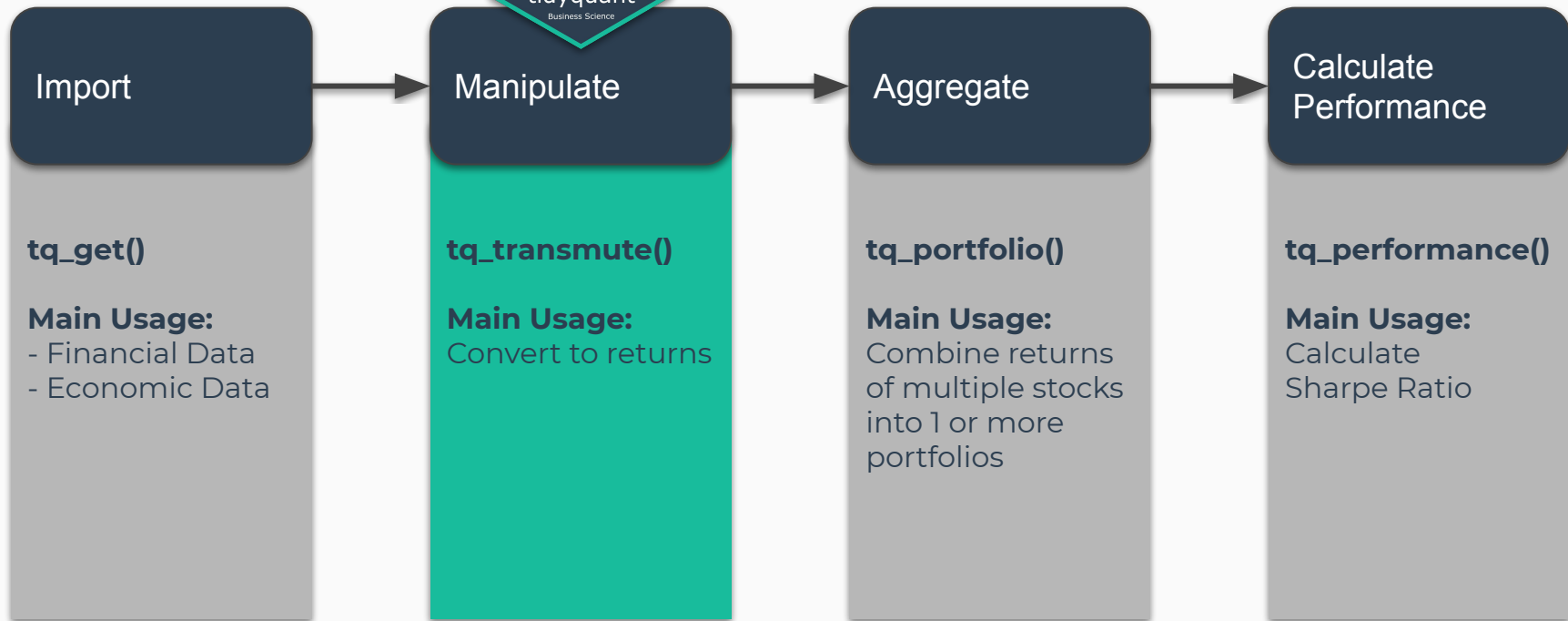
## Economic Data

```
> tibble(symbol = c("MKTGDP CNA646NWDB"),
+         name   = c("GDP China")) %>%
+   tq_get(get = "economic.data", from = "1960-01-01")
# A tibble: 58 x 4
```

	symbol	name	date	price
	<chr>	<chr>	<date>	<dbl>
1	MKTGDP CNA646NWDB	GDP China	1960-01-01	59716467625.
2	MKTGDP CNA646NWDB	GDP China	1961-01-01	50056868958.
3	MKTGDP CNA646NWDB	GDP China	1962-01-01	47209359006.
4	MKTGDP CNA646NWDB	GDP China	1963-01-01	50706799903.
5	MKTGDP CNA646NWDB	GDP China	1964-01-01	59708343489.
6	MKTGDP CNA646NWDB	GDP China	1965-01-01	70436266147.
7	MKTGDP CNA646NWDB	GDP China	1966-01-01	76720285970.
8	MKTGDP CNA646NWDB	GDP China	1967-01-01	72881631327.
9	MKTGDP CNA646NWDB	GDP China	1968-01-01	70846535056.
10	MKTGDP CNA646NWDB	GDP China	1969-01-01	79705906247.

```
# ... with 48 more rows
```

Connects To  
**Yahoo - Bloomberg - FRED - Quandl - more!**

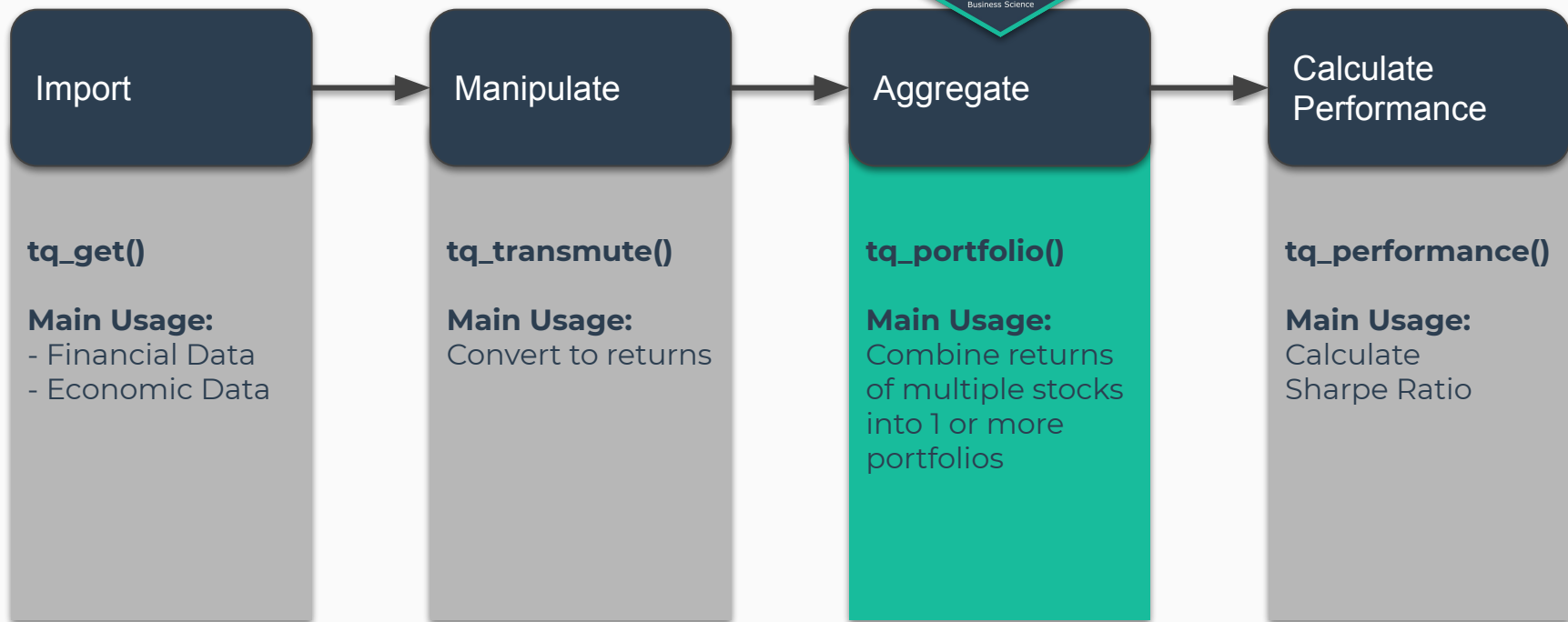




## Transform Financial Data to Returns

```
12 library(tidyquant)
13
14 end   <- "2018-12-31" %>% ymd()
15 start <- end - years(5) + days(1)
16
17 returns_m_components_tbl <- c("AAPL", "GOOG", "NFLX") %>%
18   tq_get(get = "stock.prices",
19         from = start,
20         to   = end) %>%
21   group_by(symbol) %>%
22   tq_transmute(select      = adjusted,
23               mutate_fun = periodReturn,
24               period      = "monthly") %>%
25   ungroup()|
26
27 returns_m_components_tbl
28
```

```
> Ra
# A tibble: 180 x 3
# Groups:   symbol [3]
  symbol date      Ra
  <chr>  <date>    <dbl>
1 AAPL   2014-01-31 -0.0950
2 AAPL   2014-02-28  0.0969
3 AAPL   2014-03-31  0.0200
4 AAPL   2014-04-30  0.0994
5 AAPL   2014-05-30  0.116
6 AAPL   2014-06-30  0.0277
7 AAPL   2014-07-31  0.0287
8 AAPL   2014-08-29  0.0775
9 AAPL   2014-09-30 -0.0171
10 AAPL  2014-10-31  0.0720
# ... with 170 more rows
```





## Aggregate to Portfolio

```
28 wts_tbl <- returns_m_components_tbl %>%  
29   distinct(symbol) %>%  
30   mutate(weights = c(0.25, 0.25, 0.5))  
31  
32 wts_tbl  
33  
34 returns_m_portfolio_tbl <- returns_m_components_tbl %>%  
35   tq_portfolio(symbol, monthly.returns,  
36               weights = wts_tbl,  
37               rebalance_on = "quarters"  
38 )
```

```
> wts_tbl  
# A tibble: 3 x 2  
  symbol weights  
  <chr>    <dbl>  
1 AAPL     0.25  
2 GOOG     0.25  
3 NFLX     0.5
```

```
> returns_m_portfolio_tbl  
# A tibble: 60 x 2  
  date           portfolio.returns  
  <date>              <dbl>  
1 2014-01-31         0.0479  
2 2014-02-28         0.0656  
3 2014-03-31        -0.130  
4 2014-04-30        -0.0314  
5 2014-05-30         0.176  
6 2014-06-30         0.0416  
7 2014-07-31        -0.0147  
8 2014-08-29         0.0821  
9 2014-09-30        -0.0302  
10 2014-10-31        -0.0547  
# ... with 50 more rows
```



## Aggregate to Portfolio

```
28 wts_tbl <- returns_m_components_tbl %>%  
29   distinct(symbol) %>%  
30   mutate(weights = c(0.25, 0.25, 0.5))  
31  
32 wts_tbl  
33  
34 returns_m_portfolio_tbl <- returns_m_components_tbl %>%  
35   tq_portfolio(symbol, monthly.returns,  
36               weights = wts_tbl,  
37               rebalance_on = "quarters"  
38 )
```

```
> wts_tbl  
# A tibble: 3 x 2  
  symbol weights  
  <chr>    <dbl>  
1 AAPL     0.25  
2 GOOG     0.25  
3 NFLX     0.5
```

```
> returns_m_portfolio_tbl  
# A tibble: 60 x 2  
  date           portfolio.returns  
  <date>              <dbl>  
1 2014-01-31         0.0479  
2 2014-02-28         0.0656  
3 2014-03-31        -0.130  
4 2014-04-30        -0.0314  
5 2014-05-30         0.176  
6 2014-06-30         0.0416  
7 2014-07-31        -0.0147  
8 2014-08-29         0.0821  
9 2014-09-30        -0.0302  
10 2014-10-31        -0.0547  
# ... with 50 more rows
```



## Aggregate to Portfolio

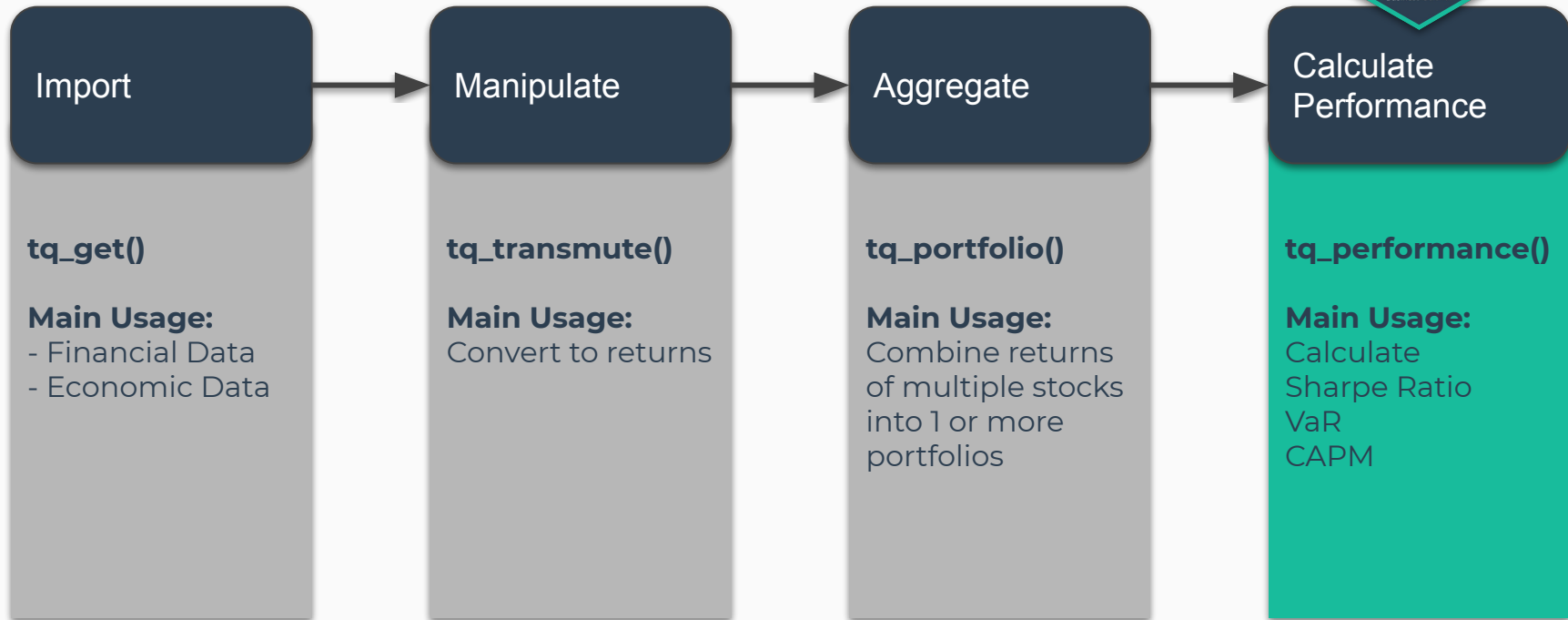
```
28 wts_tbl <- returns_m_components_tbl %>%  
29   distinct(symbol) %>%  
30   mutate(weights = c(0.25, 0.25, 0.5))
```

```
31  
32 wts_tbl
```

```
33  
34 returns_m_portfolio_tbl <- returns_m_components_tbl %>%  
35   tq_portfolio(symbol, monthly.returns,  
36     weights = wts_tbl,  
37     rebalance_on = "quarters"  
38   )
```

```
> wts_tbl  
# A tibble: 3 x 2  
  symbol weights  
  <chr>    <dbl>  
1 AAPL     0.25  
2 GOOG     0.25  
3 NFLX     0.5
```

```
> returns_m_portfolio_tbl  
# A tibble: 60 x 2  
  date           portfolio.returns  
  <date>              <dbl>  
1 2014-01-31         0.0479  
2 2014-02-28         0.0656  
3 2014-03-31        -0.130  
4 2014-04-30        -0.0314  
5 2014-05-30         0.176  
6 2014-06-30         0.0416  
7 2014-07-31        -0.0147  
8 2014-08-29         0.0821  
9 2014-09-30        -0.0302  
10 2014-10-31        -0.0547  
# ... with 50 more rows
```







## Measure Sharpe Ratio

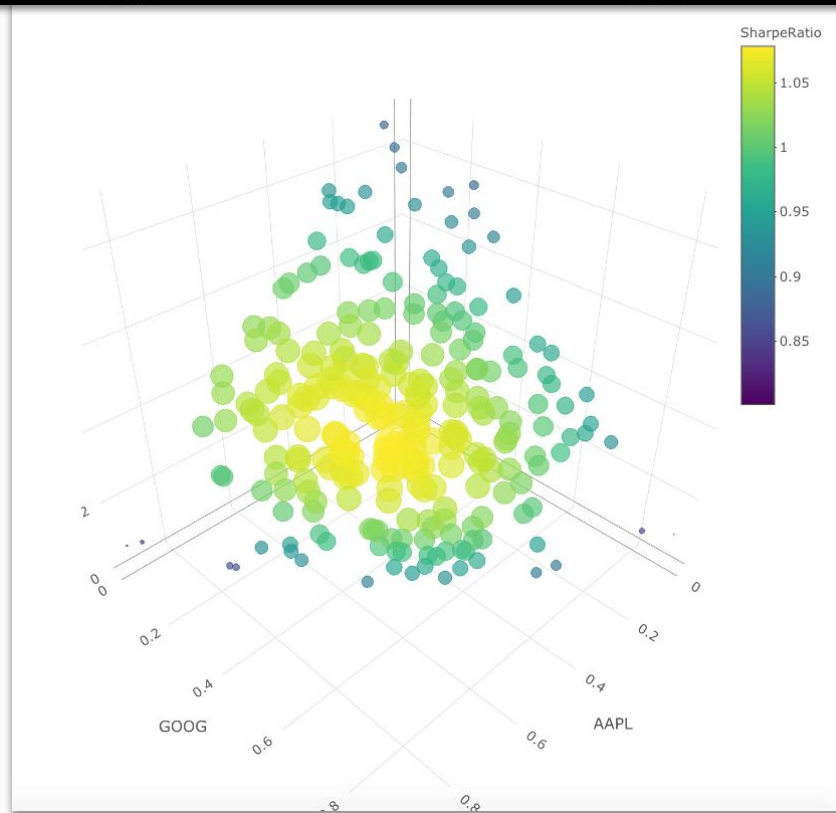
```
93 returns_m_portfolio_merged_m_tbl <- returns_m_portfolio_tbl %>%
94   add_column(symbol = "Portfolio", .before = 1) %>%
95   bind_rows(returns_m_benchmark_tbl)
96
97 returns_m_portfolio_merged_m_tbl %>%
98   group_by(symbol) %>%
99   tq_performance(Ra = monthly.returns,
100                 performance_fun = SharpeRatio.annualized,
101                 scale = 12)
102
```

```
# A tibble: 2 x 2
# Groups:   symbol [2]
  symbol `AnnualizedSharpeRatio(Rf=0%)`
  <chr>      <dbl>
1 Portfolio  1.02
2 XLK       0.984
```

# Optimize Performance

**Iterate 50X+**  
**Measure Sharpe**  
**Ratio**  
**Pick Best**  
**Portfolio**

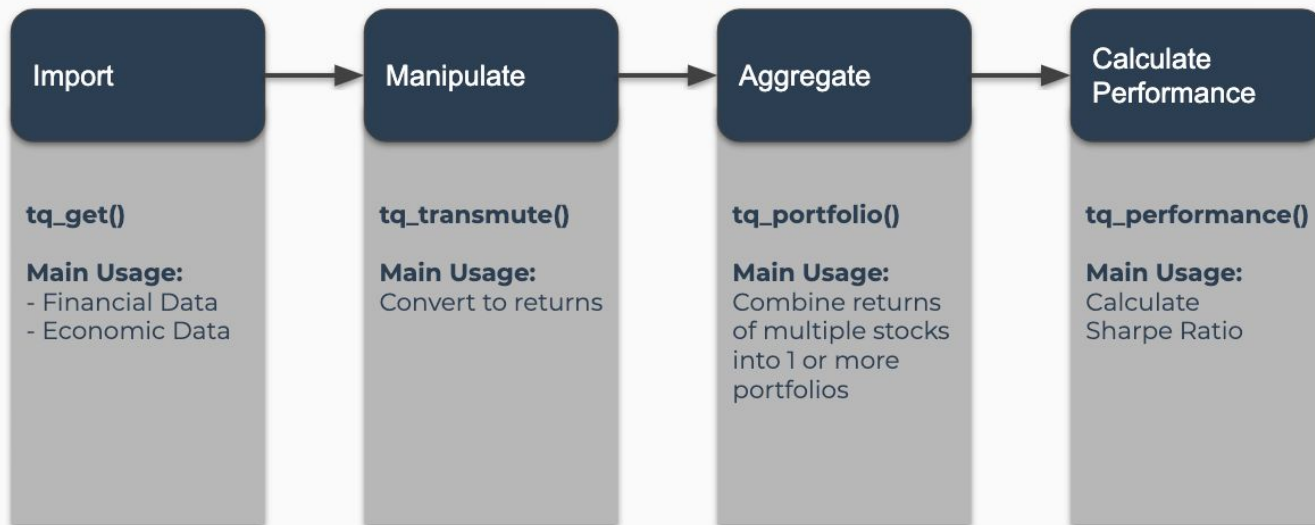
```
> best_portfolio_tbl %>%  
+   select(portfolio_id, `AnnualizedSharpeRatio(Rf=0%)`)  
# A tibble: 1 x 2  
  portfolio_id `AnnualizedSharpeRatio(Rf=0%)`  
    <int>          <dbl>  
1       124          1.08
```



# 30-Second Recap



API that manages the process of **systematically**  
evaluating stocks & portfolios



Slides Available Here:  
[bit.ly/rfinance-slides-tidyquant](https://bit.ly/rfinance-slides-tidyquant)

# Get To Know Business Science

Join Us for a LIVE Learning Lab

Take Machine Learning Models  
into Production

Learn how to create API's

Plumber R Package

## Live Event - Learning Lab (It's Free)



**Building API's in R**  
Using the plumber package & postman

Matt Dancho & David Curry  
Business Science Learning Lab

**Register Here:**  
**[bit.ly/learning-lab-plumber](https://bit.ly/learning-lab-plumber)**