

To install the package:

```
> #library(devtools)
> #install_github("rynkwn/qmj")
```

Background

qmj implements the results and methodology of the paper *Quality Minus Junk* by Clifford Asness, Andrea Frazzini, Lasse Pedersen. In their paper, they use several measures to calculate the relative profitability, growth, safety, and payouts of a company, which they use to provide an overall quality score for a company.

This quality score is used to recommend which companies to buy and which to sell, by reasoning that quality companies are likely to increase in price in the future, while “junk” companies are likely to fall in price.

Here we use the equations and methods described in the paper, coupled with data scraped from reputable online sources, in order to produce quality measurements for companies listed in the Russell 3000 Index.

Getting Started

In order to start you off, qmj comes equipped with several data sets, including company information, financial statements, and daily stock data. To access them, call:

```
> library(qmj)
> data(companies) #Stores company names and tickers from the Russell 3000 index
> data(tidydaily) #stores daily stock data for the past 5 years, if possible.
> data(tidybalance) #Stores balance sheets.
> data(tidycash) #Stores cash flow statements.
> data(tidyincome) #Stores income statements.
```

qmj currently works based on the data saved in the package. Consequently, if you want a quality measurement of the companies currently stored in the package, call:

```
> #collectmarketdata()
```

If you're only interested in accessing certain quality factors, such as profitability, you're able to call:

```
> #collectmarketprofitability(companies, tidybalance, tidycash, tidyincome)
```

This will return a numeric vector containing profitability z-scores for the given companies, where the nth number corresponds to the nth company.

Updating your Data

If you're interested in inputting your own data, you must follow the organization of the tidy data sets, though these can be generated from a user-created company data file as follows:

```
> #companies #Your custom data frame of company names and tickers.  
> #save_companies(companies)  
> #getinfo()  
> #tidyinfo()
```

These commands will automatically retrieve relevant financial data from the web for your data frame of companies, store that data in the extdata folder of qmj, and then produce tidy versions of that data to be stored in the data folder.

Updating Daily Data

Updating daily data is a slightly more protracted process, and for that reason is separated from the other functions that automatically collect financial statements. To update daily data, which is necessary for calculating safety measurements, call:

```
> #update_dailydata(companies)
```

The update dailydata function deliberately requires a parameter as opposed to the financial statement functions due to the amount of time necessary to read in and compile relevant stock data from our sources. (Google finance, with the S&P 500 being taken from Yahoo finance)

Updating Extra Financials

Updating extra financials can be used to grab the most recent betas and earnings before interest, taxes, depreciation, and amortization (*EBITDA*). However, the current data in the package is quite recent, betas do not tend to fluctuate much, and *EBITDA* are updated on an annual basis, so updating extra financials, a time-consuming process, should rarely be necessary. To update extra financials, call:

```
> #get_extrafin()
```

This function does not have parameters because when updating extra financials, it is vital that every company be updated since beta is relative, and thus modifying the beta of one company will affect the betas of other companies.