Installation

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 outperform the market, while "junk" companies are likely to underperform relative to the market.

Here we use the equations and methods described in the paper, coupled with data taken 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(financials) #Stores financial documents for the given list of companies.
> data(prices) # Stores price returns and closing stock prices for the past two
> data(qualityscores) #Stores the quality scores and the scores of its component
```

If you want to measure quality call:

```
> #market_data(companies, financials, prices)
```

If you're only interested in accessing certain quality factors, such as profitability, as well as what makes it up (such as gross profits over assets (GPOA), or cash flow over assets (CFOA)) you're able to call:

```
> #market_profitability(companies, financials)
```

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 can generate financial statements for a data frame of companies as follows:

```
> #companies #Your custom data frame of company names and tickers. The column na
> #rawdata <- getinfo(companies) #Retrieves raw financial statements from google
> #financials <- tidyinfo(rawdata) #Renders raw data in a usable format.
```

These functions will automatically retrieve relevant financial data from the web for your data frame of companies, and then allows you to pass in your new financials variable for the various quality functions.

Updating Prices

Updating prices is a separate, lengthy process, and for that reason is separated from the other functions that automatically collect financial statements. To update prices, which is necessary for calculating safety measurements, call:

```
> # rawprices <- get_prices(companies)
> # prices <- tidy_prices(rawprices)
```

The get_prices function is able to save its progress as it temporarily saves its download data to the extdata folder in the package's folder. Data is retrieved from Google finance, with the S&P 500 being taken from Yahoo finance.