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 years.
> data(quality) #Stores the quality scores and the scores of
> #its components.
> #And more detailed data sets into what makes up quality
> data(profitability)
> data(growth)
> data(payouts)
> data(safety)
```

Getting a quality data frame and a holistic summary of all its components can be done by calling

```
> #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)) call

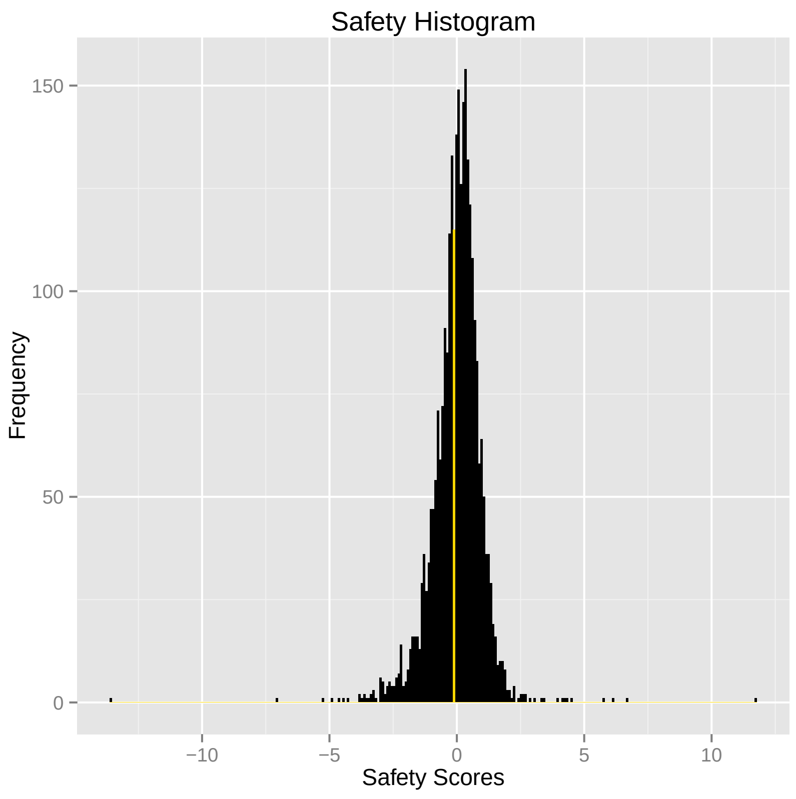
```
> #market_profitability(companies, financials)
```

Analyzing your Data

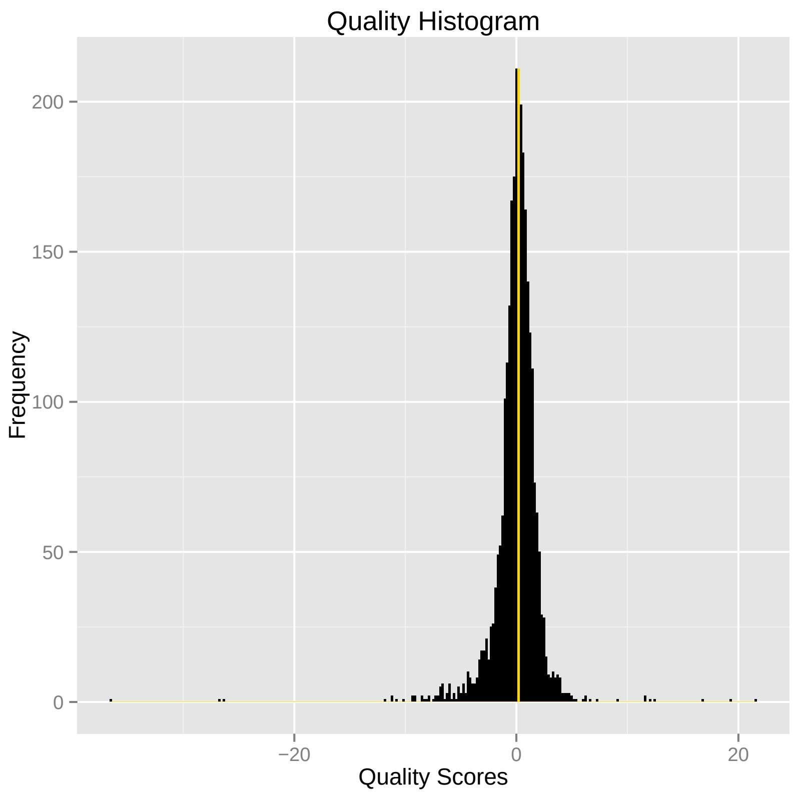
The qmj package has stored a large number of qmj objects, which store significant amounts of information about a single company and which allows more in-depth analysis of that company. Some examples of analysis follow:

```
> data(qmjs)
> first_qmj <- qmjs[[1]]
> summarize(first_qmj) # Displays key information about this qmj object.

Information for:  FLWS
---------------------------------------
Quality Score:  0.3588747
---------------------------------------
  profitability     GPOA         ROE       ROA      CFOA      GMAR       ACC
1     0.2911018 0.262272 -0.01867136 0.2019969 0.1198031 0.2308763 0.1606091


       growth        GPOA ROE        ROA        CFOA        GMAR        ACC
1 -0.03357491 -0.03355643   0 0.02050147 -0.01102495 -0.05582183 0.01561731
```

```
       safety      BAB       IVOL          LEV OhlsonOScore AltmanZScore
1 -0.08842572 0.142523 -0.2184943 -0.009880442            0            0


    payouts      EISS       DISS       NPOP
1 0.1897736 0.1998357 0.09870005 0.01514234

---------------------------------------
> #We're clearly missing some interesting data, but we can still
> #perform some analysis.
> data(safety)

> plot_safety(first_qmj, safety)

[1] "Selected object is in the yellow bin."
```



```
> #Now let's look at a graph for quality.
> data(quality)
> second_qmj <- qmjs[[2]]

> plot_quality(second_qmj, quality)

[1] "Selected object is in the yellow bin."
```



```
> #What if I'm only interested in looking closely at a few companies?
> #Well, voila.
> desired_companies <- c("GOOG", "IBM", "FLWS")
> #Returns a list containing the given qmj objects in order.
> desired_qmjs <- get_qmjs(desired_companies, qmjs)
> summarize(desired_qmjs[[1]])
```

```
Information for:  GOOG
---------------------------------------
Quality Score:  0.4324132
---------------------------------------
  profitability     GPOA        ROE        ROA       CFOA      GMAR        ACC
1     0.1768459 2.118874 -0.2618723 -0.8683505 -0.5174924 0.4885829 -0.3784276


       growth        GPOA ROE        ROA        CFOA        GMAR        ACC
1 -0.03648541 -0.02796106   0 0.01365657 -0.01396494 -0.05372761 0.01214001


      safety        BAB      IVOL       LEV OhlsonOScore AltmanZScore
1 -0.2362488 -0.3126085 0.7353582 0.3042633   -0.6317125   -0.6706373


    payouts      EISS      DISS      NPOP
1 0.5283014 0.1308389 0.7281532 0.0142409

---------------------------------------
```

But the package also provides some tools for better examining your data en masse, as opposed to individual companies.

```
> #Let's look at the head of our quality data frame.
> data(quality)
> head(quality)

                     name ticker profitability      growth      safety
1         ANGIES LIST INC   ANGI    -0.1575365 24.55764246 -0.89076389
2 SEACOAST BANKING CORP F   SBCF    -0.9552288 21.17749195  0.23623565
3                  AMERCO   UHAL    -0.3350943 19.25596295 -0.16064717
4   GUIDANCE SOFTWARE INC   GUID     0.2245725 13.61798234  0.09569998
5       BROWN & BROWN INC    BRO     0.1484205 -0.04063592 11.76587071
6 CAPITOL FEDERAL FINL IN   CFFN     5.7540587 -0.04904148  5.73770806
     payouts  quality
1 -1.8699982 21.63934
2 -1.0820805 19.37642
```

```
3 -1.9855567 16.77466
4 -1.5591484 12.37911
5  0.2254432 12.09910
6  0.2292110 11.67194

> #Angies has an abnormally high growth score, which is very suspicious.
> #Companies that are primarily driven by a single component score
> #are suspect, so let's filter out companies that are driven by growth.
> sans_growth <- filter_companies(quality, filter="growth")
> head(sans_growth)

                      name ticker profitability      growth     safety
5        BROWN & BROWN INC    BRO     0.1484205 -0.04063592 11.7658707
6  CAPITOL FEDERAL FINL IN   CFFN     5.7540587 -0.04904148  5.7377081
8      CENTURY ALUMINUM CO   CENX    -0.1912805  3.43982232  6.1767505
9  CORRECTIONS CORP OF AME    CXW    -0.3283571  3.22020489  4.1009744
10    ROUSE PROPERTIES INC    RSE     3.6209141  0.04701919  1.8876675
11       PATTERSON COS INC   PDCO    -0.7411908 -0.05296278  0.1659304
      payouts   quality
5   0.2254432 12.099098
6   0.2292110 11.671936
8  -0.1956603  9.229632
9   0.1910389  7.183861
10  0.9855644  6.541165
11  6.7805708  6.152348

> #On the other hand, if we're interested in only companies that are
> #driven by growth, we can do the following:
> driven_by_growth <- filter_companies(quality, filter="growth", remove=FALSE, i
> head(driven_by_growth)

                      name ticker profitability   growth       safety    payouts
1          ANGIES LIST INC   ANGI   -0.15753650 24.55764 -0.89076389 -1.8699982
2  SEACOAST BANKING CORP F   SBCF   -0.95522880 21.17749  0.23623565 -1.0820805
3                   AMERCO   UHAL   -0.33509433 19.25596 -0.16064717 -1.9855567
4    GUIDANCE SOFTWARE INC   GUID    0.22457246 13.61798  0.09569998 -1.5591484
7       UTAH MED PRODS INC   UTMD   -3.30709253 16.48412 -1.72466975  0.1363612
38       NEWBRIDGE BANCORP   NBBC   -0.05392051  3.83622 -0.17048483  0.2178660
    quality
```

```
1  21.63934
2  19.37642
3  16.77466
4  12.37911
7  11.58872
38  3.82968
```

```
> #We can also remove all companies with quality scores which are
> #primarily driven by any component.
> #Notice that the remove parameter is by default TRUE, and
> #isolate is by default FALSE
> liberal_arts_companies <- filter_companies(quality, filter="all")
> head(liberal_arts_companies)
```

```
                      name ticker profitability      growth    safety   payouts
6  CAPITOL FEDERAL FINL IN   CFFN     5.7540587 -0.04904148 5.7377081 0.2292110
21 HANNON ARMSTRONG SUSTAI   HASI     1.2166640 -0.02772366 1.7356800 1.7275803
22  ADAMAS PHARMACEUTICALS   ADMS     1.6916207  0.50007697 2.2264045 0.2107355
24   OPLINK COMMUNICATIONS   OPLK     0.9992939  1.89816751 1.2105671 0.2178525
33              WATSCO INC    WSO     1.9397013 -0.01815565 1.8758314 0.1712876
34          P C CONNECTION   PCCC     1.4991871 -0.01996304 0.5011669 1.9790272
     quality
6  11.671936
21  4.652201
22  4.628838
24  4.325881
33  3.968665
34  3.959418
```

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 name for tickers must be "ticker"
```

```
> #rawdata <- get_info(companies) #Retrieves raw financial
> #statements from google finance through the quantmod package.
> #financials <- tidyinfo(rawdata) #Renders raw data in a format
> #usable by other functions in this package.
```

get_info temporarily saves your progress to the extdata folder at all stages of its process, allowing you to resume your downloading if the process is interrupted for any reason.

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) #Retrieves stock price
> #data from Google Finance for listed companies for the past
> #two years. Also saves data from the S&P 500, retrieved from
> #Yahoo Finance.

> # prices <- tidy_prices(rawprices) #Renders the raw data into
> #a form usable by other functions in this package.
```

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.