Barfy's Finance Code Documentation Supplement

Daniel, John, Jose, Kay

We have provided in-code documentation. That is, for nearly every line of code within the source code, we have defined what it does and its purpose. We have also defined and explained what each function does. For completability, we have posted descriptions of the functions we used below.

Function Descriptions:

```
# getTicker(company)
# Parameters: company -- a string of a company name
# Scrapes a ticker website and searches for the ticker of the company
parameter
# Returns: company financial ticker | error if company doesn't exist
def getTicker(company):

# companyCheck(ticker)
# Parameters: ticker -- a string financial ticker
# Searches the same marketwatch website. If the inputted company /
ticker returns the string 'There were no matches found', returns
false, otherwise true
#We do this as another check because Marketwatch should be able to
return the same ticker, if we search for the ticker
# Returns: Boolean -- true if company exists, false otherwise
def companyCheck(ticker):

# calcs(balancesheet)
# Parameters: balancesheet from StockRow.com -- a pandas dataframe
# Searches the balancesheet dataframe for specific indexes, and will
use these to calculate ratios for each of the two returned years.
# Returns: Dictionary of calculated ratios
def calcs(balancesheet):

# useCSV(stock, sheet)
# Parameters: stock -- a string financial ticker | sheet -- a string
of which financial statement to use
# Downloads the respective sheet as an excel document from stockrow,
and outputs it into a pandas dataframe
# Returns: Corresponding 'sheet' -- a dataframe
def useCSV(stock,sheet):

# balsheet(stock)
# Parameters: stock -- a string financial ticker
# Duplicates useCSV for the balancesheet, but allows easier
calculation in the calcs(balancesheet) function
# Returns: Balance Sheet -- a pandas dataframe
def balsheet(stock):
```

```
# useWebscraping(url, sheet)
# Parameters: url -- a string financial ticker | sheet -- a string
corresponding to the financial statement to get
# Scrapes the Yahoo Finance website and places information for each
sheet into a dataframe
# Returns: Pandas Dataframe -- containing the information from
financial sheets
def useWebScraping(url,sheet):

# wsCalcs(bs)
# Parameters: bs -- a pandas dataframe containing the balancesheet
# Takes the bs dataframe and calculates the important financial
condition ratios, only for Yahoo Finance
# Returns: Dictionary of calculated financial ratios
def wsCalcs(bs):

# apiCalcs(balancesheet)
# Parameters: balancesheet -- a pandas dataframe
# Calculates the financial ratios from the API's balance sheet
dataframe object, only for API
# Returns: x -- Dictionary containing the important financial
condition ratios
def apiCals(balanceSheet):

# combine(data)
# Parameters: data -- a pandas dataframe
# combine the statements of 2019 and 2018 together. Put 2019 before
2018.
# Returns: out -- a pandas dataframe
def combine(data):

# apiPrint(companyName, sheet)
# Parameters: companyName -- a string financial ticker
#             sheet - specified which financial statement
# combine the statement of 2019 and 2018 together. Put 2019 before
2018.
# Returns: a dataframe with the specified financial statement
def apiPrint(companyName, sheet):

## THIS SECTION CONTAINS 3 FUNCTIONS USED TO EDIT BS, IS, AND SCF ##
# BSeditor(balancesheet)
# Parameters: balancesheet -- a pandas dataframe
# Rearranges, calculates ratios, and edits the pandas dataframe to
make it more similar to ones we see in class
# Returns: BS_edited -- a dataframe
def BSeditor(balancesheet):

# ISeditor(incstatement)
# Parameters: incstatement -- a pandas dataframe
# Rearranges, calculates ratios, and edits the pandas dataframe to
make it more similar to ones we see in class
# Returns: IS_edited -- a dataframe
```

```
def ISeditor(incstatement):

# SCFeditor(scf)
# Parameters: scf -- a pandas dataframe
# Rearranges, calculates ratios, and edits the pandas dataframe to
make it more similar to ones we see in class
# Returns: SCF_edited -- a dataframe
def SCFeditor(scf):
```