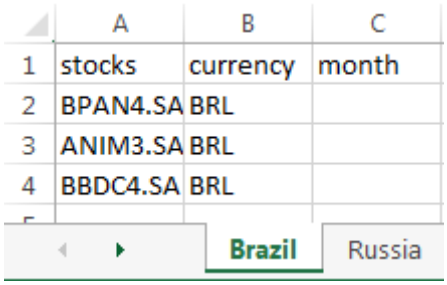
Capstone Solution design

Description

The program calculates Piotroski's F-Score and checks if the price prediction is confirmed across a dataset of non-US companies' financial data, presented in a form of a spreadsheet.

Input

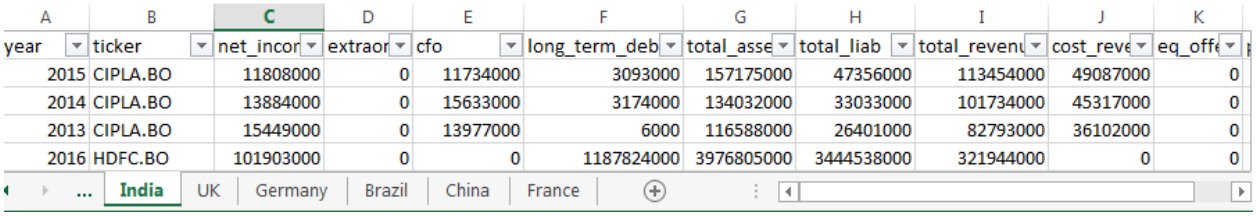
1. The spreadsheet with the list of stocks' tickers and currency of the stock for each tested country.



	A	B	C
1	stocks	currency	month
2	BPAN4.SA	BRL	
3	ANIM3.SA	BRL	
4	BBDC4.SA	BRL	

Brazil | Russia

2. The spreadsheet with financial data for each year and country:



year	ticker	net_incor	extraor	cfo	long_term_deb	total_asse	total_liab	total_reven	cost_reve	eq_offe
2015	CIPLA.BO	11808000	0	11734000	3093000	157175000	47356000	113454000	49087000	0
2014	CIPLA.BO	13884000	0	15633000	3174000	134032000	33033000	101734000	45317000	0
2013	CIPLA.BO	15449000	0	13977000	6000	116588000	26401000	82793000	36102000	0
2016	HDFC.BO	101903000	0	0	1187824000	3976805000	3444538000	321944000	0	0

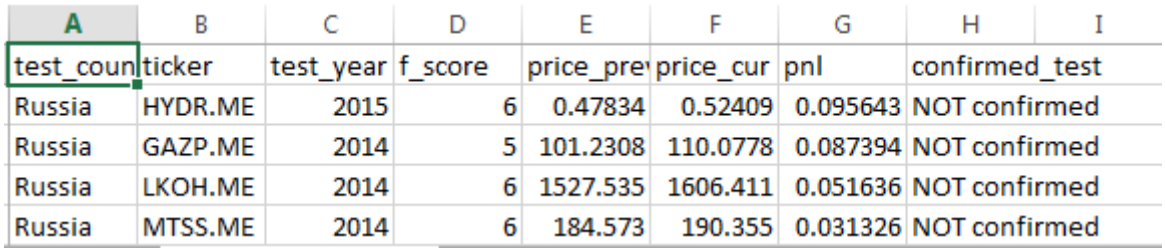
... | India | UK | Germany | Brazil | China | France

Fields:

Year, Ticker, Net Income, Extraordinary items, Cash Flow from operations, Long term debt, Total Assets, Total liabilities, Total Revenue, Cost of Revenue, Equity Offerings. All values are in thousands and in local currency.

Output

The csv-file with F-Score calculated for each year and ticker, where it is possible.



test_coun	ticker	test_year	f_score	price_pre	price_cur	pnl	confirmed_test
Russia	HYDR.ME	2015	6	0.47834	0.52409	0.095643	NOT confirmed
Russia	GAZP.ME	2014	5	101.2308	110.0778	0.087394	NOT confirmed
Russia	LKOH.ME	2014	6	1527.535	1606.411	0.051636	NOT confirmed
Russia	MTSS.ME	2014	6	184.573	190.355	0.031326	NOT confirmed

Fields:

Country, Ticker, Year, F-Score, stock price for the previous year, stock price for the testing year, pnl, status of the test: confirmed or not.

Algorithm

1. Define constants for test years and countries:
    a. year_list = [2016, 2015, 2014, 2013, 2012, 2011]
    b. countries_list = ['Russia', 'India', 'Brazil', 'China', 'UK', 'Germany', 'France']
2. Iterate years and countries
    a. Loads stocks list with **load_list_stocks(country)**
    b. Load financial data for the company with **get_source_data(country, ticker, year)**
    c. Get price for stock for testing year and the year before with **get_price_special(ticker, year)**
    d. Calculate F-Score for the test year and stock with **calc_f_score()**, providing all loaded data.
    e. Make necessary validations for missing stock price data of invalid F-Score.
    f. Calculate pnl for stocks.
    g. Calculates status 'Confirmed' if F-Score is high (7 or greater) and pnl is positive, or F-Score is low and pnl is negative; 'NOT confirmed' if otherwise.
    h. Store current result into a dataset (python list)
3. Output number of tests and save the dataset into csv-file.

A note for calc_f_score() function

The main function calculates the F-Score. It can handle all missing and invalid data. In addition, it is suitable for financials in every currency, because it only subtract the values disregard the currency or multiplier. The names of the parameters speak for themselves.

```
def calc_f_score(net_income, extraord_items, cfo, net_income_prev,
                 extraord_items_prev, long_term_debt,
                 long_term_debt_prev,
                 total_assets,
                 total_liab,
                 total_assets_prev, total_liab_prev,
                 total_revenue, cost_revenue,
                 total_revenue_prev, cost_revenue_prev,
                 eq_offer=False):
```

Conclusion

The provided solution may be applied to a wide range of tests of Piotroski's F-Score. The output of the program can be easily used for the live trading systems as the ground for creation of long or short position on a particular stock (based on the current F-Score for a selected stock).

Strengths: missing data is handled well, any country and any stock can be tested.

Weaknesses: the program requires a lot of data as input: stock list, financial data. Currently, it can't access the financial data directly.

Threats: The program relies on available Yahoo.Finance data. If at the moment of the test the data is unavailable, the test may give incorrect results.

Opportunities: With a proper financial data-scrapper or access to the financial data store, the tests may increase reliability.