

CHAILLOU Quentin

GAUJARD Jules

ANNET Carla

Collective group Project Python : AnaFi330

Summary

I.	Context and objectives	4
II.	Technologies used	4
III.	Code structure	5
IV.	Key Features.....	5
V.	Technical choices and difficulties	10
VI.	Resultat.....	11
VII.	Limits and improvements.....	13
VIII.	Annexes	14
1)	GitHub.....	14
2)	Evaluation Grid	15

At first, we were three students passionate about finance, frustrated to see how the financial analysis remained obscure for most investors, both beginners and confirmed. We saw a recurring need: to have a simple, fast and reliable tool to assess the financial strength of listed companies.

This is how AnaFi360 was born.

Our ambition? Create a tool capable of condensing hours of analysis in a few clicks, based on proven criteria, inspired by the methods of one of the best managers in Europe, William Higgons. Thanks to a clear interface and impactful visualizations, AnaFi360 allows you to compare, understand and select the best performing companies according to rigorous criteria: growth, profitability, valuation, debt, and more.

This project, at the crossroads between financial expertise, technology and pedagogy, is aimed at all those who want to invest with method, without drowning in numbers.

AnaFi360 – your new smart investing reflex.

Available now at a price of 750€ HT.

I. Context and objectives

⇒ *What is the purpose of this code?*

This program aims to simplify and accelerate the financial analysis of listed companies. With just a few clicks, the user can access key information such as financial statements, performance indicators, market data and automatically calculated financial ratios. All this information is centralized on a single clear and intuitive interface, which represents a considerable time savings.

This code is aimed at a wide audience of investors, both novice and experienced. It provides beginners with an educational and structured tool to better understand the financial health of companies, while experienced investors can use it as a quick and effective decision support tool. By gathering reliable data and making it easier to read and interpret them, this program aims to support users in their investment choices, while providing them with a relevant and accessible market financial analysis

⇒ *What problem is it trying to solve?*

This program aims to respond to a common problem among investors: quickly identify listed companies that meet specific financial criteria. Indeed, given the massive amount of data available and the complexity of some analyses, it can be difficult for an investor—especially a beginner—to effectively select companies consistent with their investment strategy.

The tool allows companies to be filtered, analyzed and compared according to key financial ratios (profitability, solvency, liquidity, etc.), facilitating decision-making. It automates some of the analysis work, reducing the risk of manual errors while saving significant time.

II. Technologies used

⇒ *What programming languages did you use?*

We used a python language on the google collab interface.

⇒ *Did you use specific libraries or frameworks?*

Packages have been installed:

```
# Install the yfinance package to retrieve financial data from Yahoo Finance
!pip install yfinance

# Install forex-python for currency exchange rates and conversions
# (Note: yfinance is repeated here but not necessary since it was already installed above)
!pip install yfinance forex-python

# Install unicode to remove accents and normalize text (e.g., for searching company names)
!pip install unicode

# Install pytz for handling time zones (useful when working with financial data in different markets)
!pip install pytz

# Install tqdm for displaying progress bars in loops (e.g., when processing many stocks)
!pip install tqdm

# Upgrade yfinance to the latest version (helps fix issues like 'too many requests' from Yahoo Finance)
!pip install --upgrade yfinance
```

```
import requests
import pandas as pd
from unicode import unicode

from datetime import datetime, timedelta
import numpy as np
import pytz
```

```
import matplotlib.pyplot as plt
import numpy as np
```

⇒ *Is the code executed locally, online, embedded in an application?*

The code is executed online.

III. Code structure

⇒ *How is your code organized ?*

The code is organized in several steps, according to 3 functions:

- Installation of packages
- Rating of companies on an index on a scale from 1 to 10 (information from Yahoo Finance)
- Possibility of a zoom on company chosen by the user for further analysis

IV. Key Features

⇒ *What are the main functions that the code performs?*

Block 1: Installing packages with Pip Install

```
# Install the yfinance package to retrieve financial data from Yahoo Finance
!pip install yfinance

# Install forex-python for currency exchange rates and conversions
# (Note: yfinance is repeated here but not necessary since it was already installed above)
!pip install yfinance forex-python

# Install unicode to remove accents and normalize text (e.g., for searching company names)
!pip install unicode

# Install pytz for handling time zones (useful when working with financial data in different markets)
!pip install pytz

# Install tqdm for displaying progress bars in loops (e.g., when processing many stocks)
!pip install tqdm

# Upgrade yfinance to the latest version (helps fix issues like 'too many requests' from Yahoo Finance)
!pip install --upgrade yfinance
```

Block 2: Automatic fundamental analysis

1. Selection of the stock index

- Displays a list of major stock market indices.
- Prompts the user to choose one (e.g., "CAC 40").
- Uses the associated Wikipedia URL to retrieve the list of company tickers.

2. Ticker extraction

- Uses `pandas.read_html()` to read tables from Wikipedia.
- Detects the column containing stock symbols.
- Adjusts ticker formats according to the stock exchange (e.g., .PA for Paris, .DE for Germany, etc.).

3. Utility functions

- `normalizer()` : converts a real financial value into a score between 0 and 1, based on whether it meets a target.
- `try_get()` : searches for a financial value in a DataFrame.
- `growth_rate()` and `average_growth_per_year()` : calculate the average annual growth.
- `extract_series()` : extracts a series of financial data (revenue, earnings, etc.) over several years.

4. Fundamental Criteria and Target Values

The criteria used in our analysis are directly inspired by those employed by one of the companies in which we carry out our alternance, specialized in asset management.

They are also consistent with the investment principles of William Higgons, recognized as one of the best fund managers in Europe. At the head of Indépendance AM, he has achieved an exceptional performance of more than 4,000% since 1993.

Its three key criteria are: growing revenue, return on equity (ROE) above 10%, and a price/profit ratio (PE ratio) below 12.

```
# -----
# Define target objectives for each metric
# -----
valeurs_cibles = {
    "Croissance CA": 0.03,
    "Croissance RN": 0.10,
    "Croissance FCF": 0.00,
    "Croissance Dividende": 0.00,
    "Marge nette": 0.15,
    "ROE": 0.10,
    "ROCE": 0.12,
    "PER": 12,
    "Payout Ratio": (0.30, 0.60),
    "Dettes / CP": 0.80
}
```

Ten criteria are defined to assess the "financial quality" of a company:

- **Revenue Growth ($\geq 3\%$)**
A company that no longer grows risks losing market share or competitiveness. Moderate growth ($\geq 3\%$) indicates that the company is progressing and adapting to its environment.
- **Net Income Growth ($\geq 0\%$)**
An increase in profits reflects effective cost management, margins, or operational efficiency. It shows the company is successfully turning growth into profit.
- **Free Cash Flow Growth ($\geq 0\%$ to 5%)**
Free cash flow (FCF) is the money truly available after investments. A company may have solid accounting results but still burn cash. Maintaining or growing FCF is vital for financial autonomy (investments, dividends, share buybacks, etc.).
- **Dividend Growth ($\geq 0\%$)**
A stable or rising dividend demonstrates management's confidence in the company's future. It also appeals to long-term investors.

- **Net Margin ($\geq 15\%$)**

This is a direct indicator of profitability: the higher the margin, the more resilient the company is during downturns, and the more it can invest or distribute.

- **ROE – Return on Equity ($\geq 10\%$)**

ROE measures the return on shareholders' equity. A return of at least 10% means the company is creating value with its equity capital.

- **ROCE – Return on Capital Employed ($\geq 12\%$)**

ROCE includes both equity and debt, offering a broader view of operational profitability. It avoids misleading leverage effects on ROE.

- **PER – Price-to-Earnings Ratio (≤ 12 to 15)**

This valuation metric helps avoid overvalued companies. A PER below 15 indicates that the price paid for each euro of earnings is reasonable.

- **Payout Ratio (30–60%)**

- < 30%: too conservative, limited shareholder returns.
- 60%: too generous, potential financial strain if profits decline.
- Optimal range: 30–60%, striking a balance between shareholder payout and internal reinvestment.

- **Debt / Equity Ratio (≤ 0.8)**

This ratio reflects prudence in financing structure. High debt levels mean greater sensitivity to interest rate hikes or downturns. Avoid overly indebted companies.

5. Company Analysis

For each company in the selected index:

- Retrieves financial statements via **yfinance** (revenue, net income, FCF, margins, etc.).
- Calculates annual growth rates over 4 years for various metrics.
- Computes ratios such as ROE, ROCE, net margin, PER, etc.
- Compares each metric against target values.
- Assigns **1 green point** per criterion met (maximum = 10). If a criterion is not met, it receives an **orange point**.

6. Final Output

- Ranks companies by number of green points.
- Displays the ranking in the terminal.
- Saves the results in an Excel file: `classement_entreprises.xlsx`.

Block 3: Complete and Visual Fundamental Analysis of a Public Company

1. Downloading an Excel file from Dropbox

- Automatically downloads an Excel file from Dropbox.
- The file contains company names and their associated stock tickers

- The file is named `lib-ll-s.xlsx`.

2. Searching for a ticker from a company name

- The user inputs the name of a company (e.g., "L'Oréal").
- A `unicode()` function removes accents, converts to lowercase, and strips punctuation.
- The script performs a partial match search in the "name" column of the Excel sheet.
- If a match is found, the ticker is automatically retrieved; otherwise, the user must enter it manually.

3. Retrieving financial data

The script uses **yfinance** to collect:

- Financial statements (revenue, net income, free cash flow...)
- Balance sheet (total assets, debt...)
- Current stock price, dividends, sector, market capitalization...

It then calculates key financial ratios such as:

- Net margin = Net income / Revenue
- Debt / EBITDA
- ROE and ROCE
- PER, Payout ratio, Dividend yield

Example of some calculations:

```
debt_to_equity = info.get("debtToEquity")
debt_to_ebitda = total_debt/ebitda if total_debt and ebitda else None
net_margin = net_income/revenue if net_income and revenue else None

# Calculate ROCE if not a financial/insurance company
if info.get("sector", "").lower() in ["financial services", "insurance"]:
    roce = None
else:
    roce = ebit/(total_assets - current_liabilities) \
           if ebit and total_assets and current_liabilities else None
```

4. Clear display of financial fundamentals

Results are displayed in a clear and readable format:

- In millions of dollars (except for market cap in billions)
- Ratios are shown as percentages

The result is a concise and insightful summary of the company's financial health.

5. 4-Year Growth Analysis

The script retrieves annual data for:

- Revenue
- Net income

- Free cash flow
- Dividend per share
- Net margin (Net income / Revenue)

It displays:

- Values over the past 4 years
- Annual growth rates
- 3-year average growth rates

Example of some growth :

```
def growth_rate(start, end):
    """Compute simple growth rate between two values"""
    if start is None or end is None or start == 0:
        return None
    return (end - start) / abs(start)

def average_annual_growth(series):
    """Compute the average annual growth rate for a series"""
    rates = []
    for i in range(1, len(series)):
        r = growth_rate(series[i-1], series[i])
        if r is not None:
            rates.append(r)
    return sum(rates) / len(rates) if rates else None

def extract_series(df, label, years):
    """Extract a list of values for given years from a financial statement"""
    try:
        return [df.loc[label][yr] if label in df.index else None for yr in years]
    except:
        return [None] * len(years)

def display_variations(name, values, rates, labels, is_percent=True):
    """Print value and growth rate for each year"""
    print(f"\n{name}")
    for i, v in enumerate(values):
        disp = format_percent(v) if is_percent else format_millions(v)
        print(f" {labels[i]}: {disp}")
    for i in range(1, len(values)):
        print(
            f" Growth {labels[i-1]} → {labels[i]}: {format_percent(rates[i-1])}"
        )
```

6. Evaluation against financial targets

Each indicator is compared to an ideal target value:

- *For example: revenue growth $\geq 3\%$, ROE $\geq 10\%$, PER ≤ 12 , etc.*

Values are normalized between 0 and 1 based on performance.

This results in a multi-criteria assessment of the company's financial performance.

7. Interactive Radar Chart (Decagon)

Normalized scores are visualized in a radar chart:

- Each axis represents a fundamental financial criterion
- A green point indicates the target is met; otherwise, it's orange

- This allows for quick visualization of the company's financial strengths and weaknesses

8. Stock Price Trend Chart

The script plots the 5-year stock price trend:

- Includes a linear trendline (red line)
- Annotates the latest stock price and the average annual return (CAGR)

The chart is interactive, built with *Plotly*, clear, and includes helpful annotations.

V. Technical choices and difficulties

⇒ *What were the most important technical choices, and why?*

The first key technical choice was to define the correct financial evaluation criteria, in order to ensure that results are relevant for users. It was important that these criteria be objective, reliable and understandable.

Then, the use of adapted libraries was a central element: libraries such as *yfinance*, *pandas* or *plotly* were chosen for their robustness, their clear documentation and their compatibility with financial data processing.

Finally, special attention was paid to the visual output of the project. The objective was to produce readable and accessible graphical representations, even for non-financial users, in order to make the results easily usable.

⇒ *Did you encounter any particular problems? How did you solve them?*

Several difficulties were encountered during the project. The first concerned the precise understanding of users' needs, which is essential to adapt the final product.

Secondly, the selection of relevant financial criteria required a thorough analysis so as not to overload the results with unnecessary indicators. The technical development has also posed challenges, especially with regard to the structuring of the code and the overall workflow.

Programming errors, bugs and broken lines of code were identified at different stages. To solve them, a methodical debugging work has been put in place. In addition, it was necessary to avoid accidental changes to certain sensitive parts of the code.

Finally, resources such as official library documentation and ChatGPT assistance have been invaluable in overcoming various technical blockages.

⇒ *Did you optimize certain parts of the code (runtime, readability, etc.)?*

Yes, several optimizations have been implemented. To improve the readability of the results, visually explicit graphs were integrated: a radar chart was generated using the *matplotlib* library to compare several financial criteria, While the five-year price evolution was represented dynamically with *plotly*, which allows for better interaction with data.

Regarding the optimization of the execution time, a simple but effective strategy consisted in inserting delays (tempo) in the requests sent to Yahoo Finance, so as not to overload the service and avoid errors or blockages related to an excess of simultaneous requests.

VI. Resultat

This is the result of the second block with the analysis of the selected index with all the companies that are classified according to our criteria :

```

Available indices:
1. S&P 500
2. NASDAQ 100
3. DOW JONES
4. CAC 40
5. FTSE 100
6. DAX
7. HANG SENG
8. SMI
9. IBEX 35

Which index would you like to analyze? (copy-paste a name from the list above): CAC 40
Processing companies: 100% [██████████] 40/40 [01:03<00:00, 1.59s/company]

Company Green Points
TotalEnergies SE (TTE.PA) 9
Hermes International Societe en commandite par actions (RMS.PA) 9
L'Air Liquide S.A. (AI.PA) 8
Publicis Groupe S.A. (PUB.PA) 8
L'Oréal S.A. (OR.PA) 8
Legrand SA (LR.PA) 8
Schneider Electric S.E. (SU.PA) 8
Compagnie de Saint-Gobain S.A. (SGO.PA) 8
LVMH Moët Hennessy - Louis Vuitton, Societe Europeenne (MC.PA) 7
Teleperformance SE (TEP.PA) 7
Vinci SA (DG.PA) 7
Dassault Systemes SE (DSY.PA) 7
Capgemini SE (CAP.PA) 7
Airbus SE (AIR.PA) 7
Accor SA (AC.PA) 7
BNP Paribas SA (BNP.PA) 6
Thales S.A. (HO.PA) 6
Veolia Environnement SA (VIE.PA) 6
Safran SA (SAF.PA) 6
STMicroelectronics N.V. (STMPA.PA) 6
Engie SA (ENGI.PA) 6
Credit Agricole S.A. (ACA.PA) 6
Danone S.A. (BN.PA) 6
Edenred SE (EDEN.PA) 6
Compagnie Generale des Etablissements Michelin Societe en commandite par actions (ML.PA) 5
Societe Generale Societe anonyme (GLE.PA) 4
AXA SA (CS.PA) 4
Bouygues SA (EN.PA) 4
Eurofins Scientific SE (ERF.PA) 4
EssilorLuxottica Societe anonyme (EL.PA) 4
Orange S.A. (ORA.PA) 4
Pernod Ricard SA (RI.PA) 4
Sanofi (SAN.PA) 4
Renault SA (RNO.PA) 4
ArcelorMittal S.A. (MT.AS) 3
Carrefour SA (CA.PA) 3
Stellantis N.V. (STLAP.PA) 3
Unibail-Rodanco-Westfield SE (URW.PA) 2
Vivendi SE (VIV.PA) 2
Kering SA (KER.PA) 1

```

Here, are the results of the third block that zoom in on the chosen company :

```

Enter the stock name: total
The ticker for 'total' is: TTE.PA

--- Fundamental Analysis of TotalEnergies SE ---
Sector: Energy
Country: France
Market Capitalization: 111.9 B $

Revenue: 195 610 M $
Net Income: 15 758 M $
Free Cash Flow: 15 945 M $

P/E Ratio: 9.590909
Payout Ratio: 57.1 %
Net Margin: 8.1 %

ROE: 11.7 %
ROCE: 18.8 %

Debt/Equity: 49.73 %
Debt/EBITDA: 1.38

Dividend per Share: 3.4 $
Current Stock Price: 50.64 EUR
Last Dividend Paid: 01/10/2025
Next Dividend Expected: N/A
Dividend Yield: 666.00 %

Revenue
2021: 184 634 M $
2022: 263 310 M $
2023: 218 945 M $
2024: 195 610 M $
Growth 2021 → 2022: 42.6 %
Growth 2022 → 2023: -16.8 %
Growth 2023 → 2024: -10.7 %

Net Income
2021: 16 032 M $
2022: 20 526 M $
2023: 21 384 M $
2024: 15 758 M $
Growth 2021 → 2022: 28.0 %
Growth 2022 → 2023: 4.2 %
Growth 2023 → 2024: -26.3 %

```

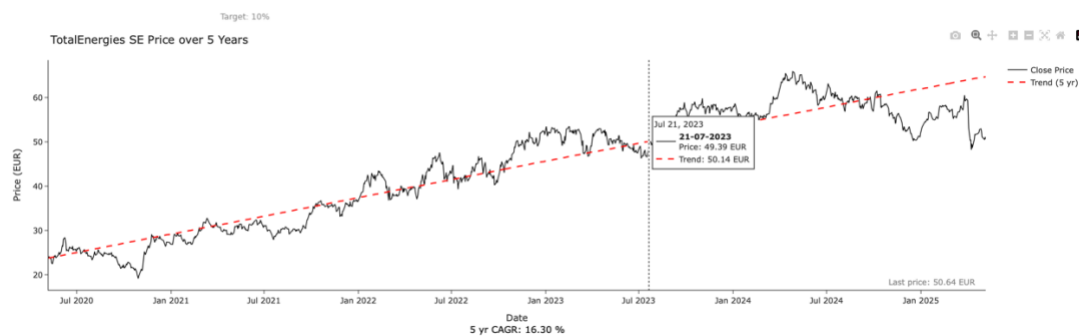
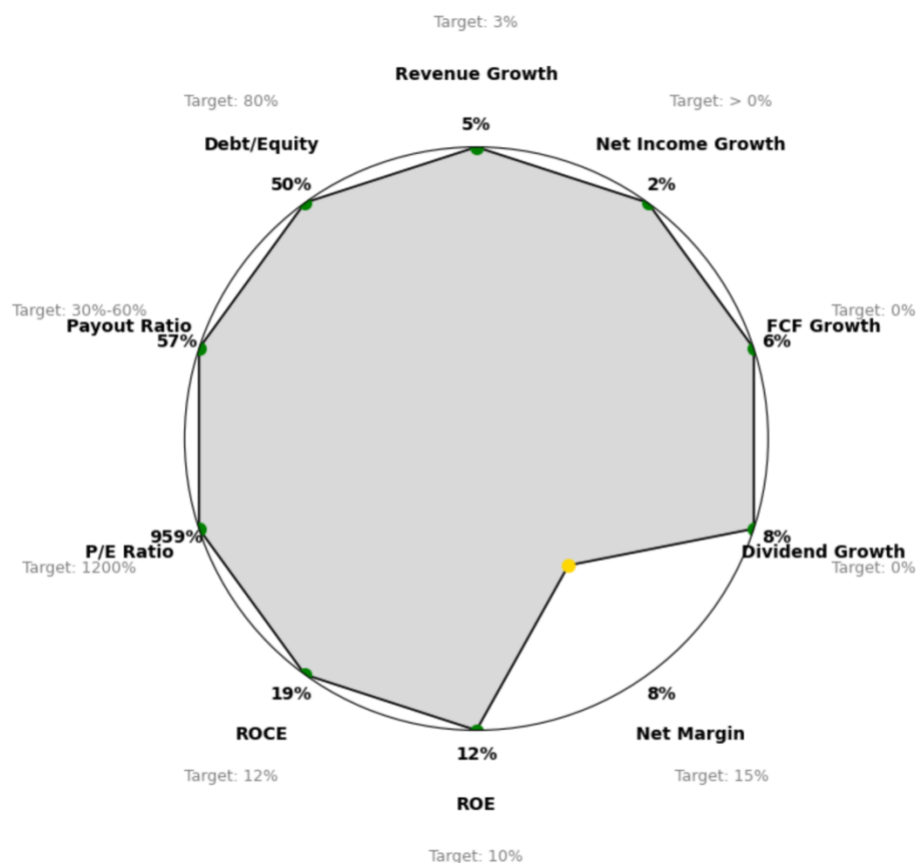
Net Margin
 2021: 8.7 %
 2022: 7.8 %
 2023: 9.8 %
 2024: 8.1 %
 Growth 2021 → 2022: -10.2 %
 Growth 2022 → 2023: 25.3 %
 Growth 2023 → 2024: -17.5 %

Free Cash Flow
 2021: 18 067 M \$
 2022: 31 677 M \$
 2023: 22 957 M \$
 2024: 15 945 M \$
 Growth 2021 → 2022: 75.3 %
 Growth 2022 → 2023: -27.5 %
 Growth 2023 → 2024: -30.5 %

Dividend per Share
 2021: 2.64 €
 2022: 3.67 €
 2023: 2.86 €
 2024: 3.06 €
 Growth 2021 → 2022: 39.0 %
 Growth 2022 → 2023: -22.1 %
 Growth 2023 → 2024: 7.0 %

--- Average Annual Growth Over 3 Years ---

Revenue: 5.0 %
 Net Income: 2.0 %
 Net Margin: -0.8 %
 Free Cash Flow: 5.8 %
 Dividend per Share: 8.0 %



VII. Limits and improvements

⇒ *Are there any known limitations to this code?*

Yes, there are several limitations to be considered when using this code. First, calculated financial targets – such as the price/profit ratio (PE ratio) – must be interpreted with caution as their relevance varies greatly depending on the business sector of the company being analyzed. For example, a high PE may be normal for a fast-growing technology company but worrisome for an industrial company.

Second, the analysis is based solely on the last five years of historical data. This choice allows a relatively recent and relevant reading, but may obscure longer dynamics or broader economic cycles.

Another technical limitation concerns the reliance on Yahoo Finance via the yfinance library. This source, although accessible and rich, can sometimes present data failures or inconsistencies. It is also common that errors occur if the packages used are not up to date, which requires regular maintenance of the code.

Thus, even if the program is generally reliable, it must be monitored and updated to ensure its long-term operation.

⇒ *What could you improve if you had more time?*

With more time, several areas of improvement could be implemented to enhance the quality and scope of the project. First, a more thorough benchmarking phase would allow the performance of the company under analysis to be compared with that of direct competitors or with the average in the sector. This would provide a more relevant reference framework and help to better interpret financial results.

Then, the development of a dedicated application, with a clear and interactive interface, would allow non-specialist users to generate their own financial analyses in just a few clicks. This application could integrate custom filters, alerts or exportable reports.

Finally, in a commercial exploitation logic, a communication and dissemination strategy of the product could be put in place. This would involve identifying target customers, promoting the product and, above all, taking user feedback into account to evolve the tool. The use of a business profile could also help to structure the sales process and broaden the scope of the project beyond the academic framework.

VIII. Annexes

1) GitHub

⇒ The project justification is posted on GitHub:

The screenshot displays the GitHub interface for the repository 'AnaFi360' owned by 'julesgaujard'. The repository is private and has 1 branch and 0 tags. The main content area shows the 'README' file with the title 'AnaFi360'. The right sidebar provides information about the repository, including 'About' (no description), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (100.0% Jupyter Notebook). The bottom section, 'Manage access', shows a list of collaborators: 'agoumar', 'carla.annet@essca.eu', and 'quentin.chaillou@essca.eu', all with 'Pending Invite' status. A sidebar on the left contains navigation links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings.

2) Evaluation Grid

Evaluation Grid

Skills	COLAB or GIT LINK:	
Subject and data source linked to finance / specialization		/2
Database import		/2
Database processing		/2
Code structure		/2
Packages		/2
Syntax		/2
Clarity and brevity of the scripts		/2
Runs (upload your script on Moodle + explanation document)		/2
Originality		/2
Use of colab / github / spyder / ...		/2
Overall Grade		/20