20598 – Finance with Big Data

PC Lab #2: Applying the CAPM (Week 3)

Clément Mazet-Sonilhac
clement.mazetsonilhac@unibocconi.it

Finance Department, Bocconi University

PC Labs Grading

- PC Labs solutions are submitted as Jupyter Notebooks, via email
  - Email title : PCLab#2 - Group X - Name1 Name2 Name3
  - Your Jupyter Notebook starts in the same way (same .ipynb name)
  - Tell me how long did it take and the difficulty (from 1 to 10)

PC Labs Grading

- PC Labs solutions are submitted as Jupyter Notebooks, via email
  - Email title : PCLab#2 - Group X - Name1 Name2 Name3
  - Your Jupyter Notebook starts in the same way (same .ipynb name)
  - Tell me how long did it take and the difficulty (from 1 to 10)

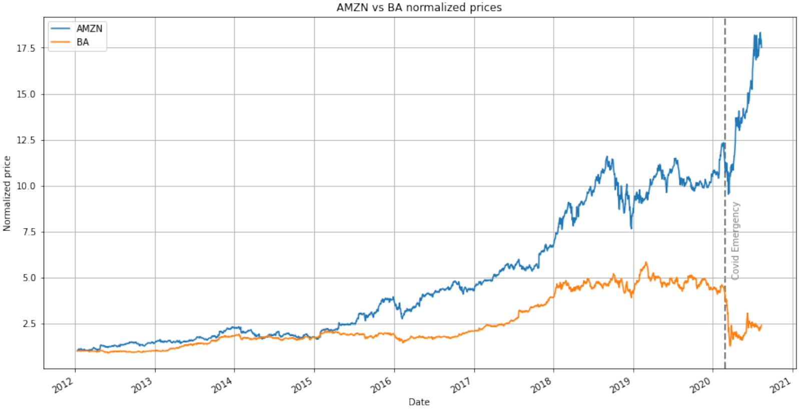
- PC Labs grade will depend on:
  - Your ability to submit it before the deadline **(Friday, midnight)**
  - The quality of your code (comments, readability, use of functions, etc.)
  - The structure of the Jupyter Notebook: well organized, explain what you are doing and why
  - Your ability to complete the tasks and innovate

Do’s and don’ts

Great stuff !

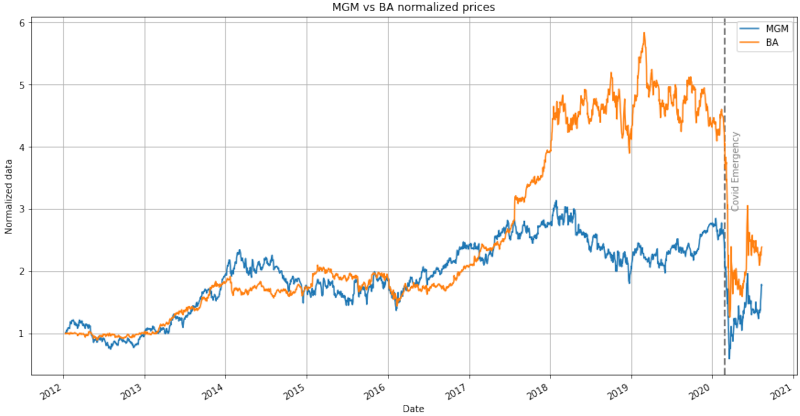
Do's and don'ts

Great stuff!



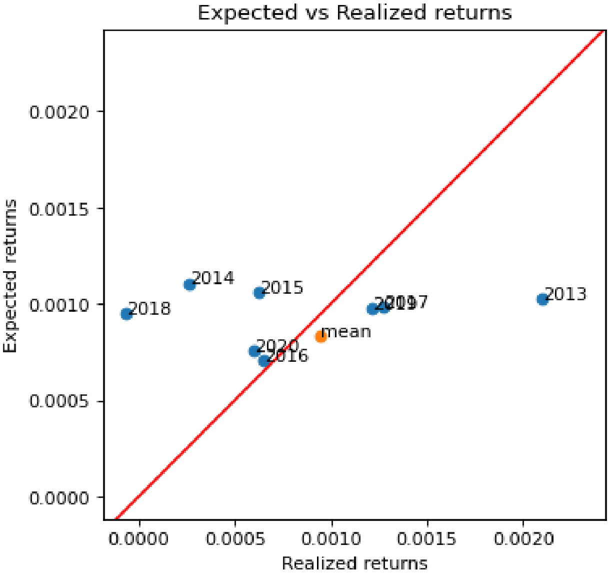
Do’s and don’ts

Great stuff !



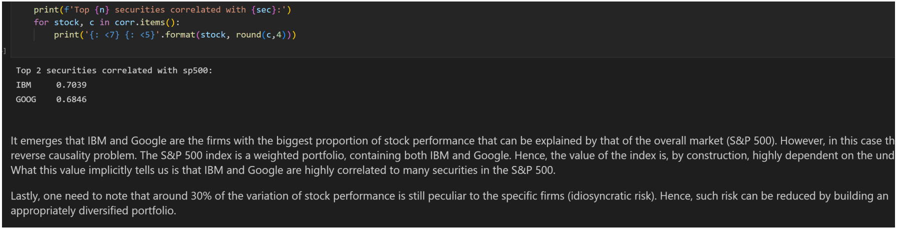
Do's and don'ts

Great stuff!



Do's and don'ts

Great stuff !



```
    print(f'Top {n} securities correlated with {sec}:')
    for stock, c in corr.items():
        print('{: <7} {: <5}'.format(stock, round(c,4)))
```

```
Top 2 securities correlated with sp500:
IBM     0.7039
GOOG    0.6846
```

It emerges that IBM and Google are the firms with the biggest proportion of stock performance that can be explained by that of the overall market (S&P 500). However, in this case th
reverse causality problem. The S&P 500 index is a weighted portfolio, containing both IBM and Google. Hence, the value of the index is, by construction, highly dependent on the und
What this value implicitly tells us is that IBM and Google are highly correlated to many securities in the S&P 500.

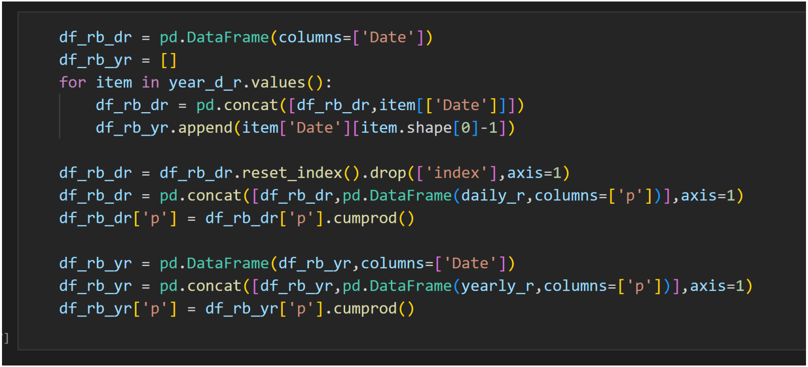
Lastly, one need to note that around 30% of the variation of stock performance is still peculiar to the specific firms (idiosyncratic risk). Hence, such risk can be reduced by building an appropriately diversified portfolio.

Do's and don'ts

Not so great...

Do’s and don’ts

Not so great...



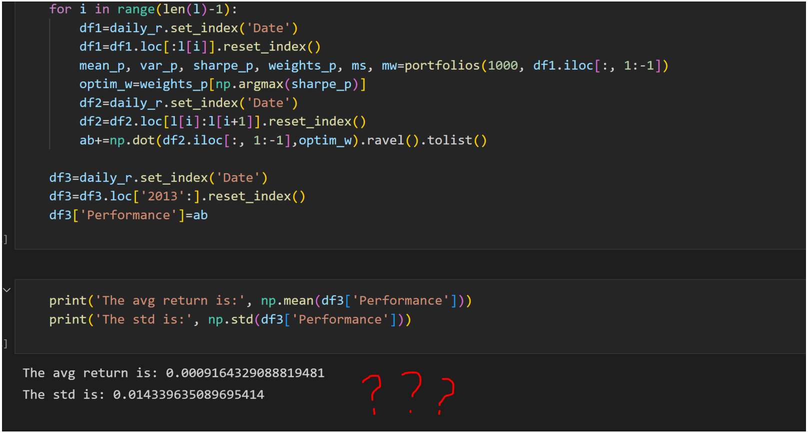
```
df_rb_dr = pd.DataFrame(columns=['Date'])
df_rb_yr = []
for item in year_d_r.values():
    df_rb_dr = pd.concat([df_rb_dr,item[['Date']]])
    df_rb_yr.append(item['Date'][item.shape[0]-1])

df_rb_dr = df_rb_dr.reset_index().drop(['index'],axis=1)
df_rb_dr = pd.concat([df_rb_dr,pd.DataFrame(daily_r,columns=['p'])],axis=1)
df_rb_dr['p'] = df_rb_dr['p'].cumprod()

df_rb_yr = pd.DataFrame(df_rb_yr,columns=['Date'])
df_rb_yr = pd.concat([df_rb_yr,pd.DataFrame(yearly_r,columns=['p'])],axis=1)
df_rb_yr['p'] = df_rb_yr['p'].cumprod()
```

Do’s and don’ts

Not so great...



```
    for i in range(len(l)-1):
        df1=daily_r.set_index('Date')
        df1=df1.loc[:l[i]].reset_index()
        mean_p, var_p, sharpe_p, weights_p, ms, mw=portfolios(1000, df1.iloc[:, 1:-1])
        optim_w=weights_p[np.argmax(sharpe_p)]
        df2=daily_r.set_index('Date')
        df2=df2.loc[l[i]:l[i+1]].reset_index()
        ab+=np.dot(df2.iloc[:, 1:-1],optim_w).ravel().tolist()

    df3=daily_r.set_index('Date')
    df3=df3.loc['2013':].reset_index()
    df3['Performance']=ab
```

```
    print('The avg return is:', np.mean(df3['Performance']))
    print('The std is:', np.std(df3['Performance']))
```

```
The avg return is: 0.0009164329088819481
The std is: 0.014339635089695414
```

? ? ?

Applied CAPM Theory

Goals

- Manipulate and visualize stock market data (S&P 500)
- Web-scrapping market data
- Compute CAPM beta and alpha
- Test the theory: does the CAPM predict explain and expected return ?

Big picture context

- You've just been hired by a small unsophisticated hedge-fund
- Your money comes from the return more than the size → you're not a passive investor anymore

Big picture context

- You've just been hired by a small unsophisticated hedge-fund

- Your money comes from the return more than the size → you're not a passive investor anymore

- The hedge-fund manager asks you to figure out the systemic risk (beta) of 8 stocks, for which your broker offers no transaction fees.

- The (same) data is on Bboard or on my website

Task #1 : Visualization and 1st step toward the CAPM

- Use the `PCLab_Data.xlsx` to produce scatter plots of each stocks' daily returns against the market daily returns

- Comment: is there an apparent correlation ? Strong ? For which stock ?

Task #2 : Compute alpha and beta

- Apply the CAPM : run an OLS regression of stock i returns on market returns, over the whole period. Assume $r_f = 0$. Are the estimates significant ?

- Plot the beta and alpha for the 8 stocks. Which stock has the highest (lowest) beta and alpha ? Comment !

- Use the observed returns and the predicted ones to compute and plot the histogram of error terms ε_i for each stock. Comment !

- Your boss wants to take a lot of risk to deliver high return. He asks you to (i) select the 4 riskier assets (over the full period), (ii) form an equally weighted portfolio, and (iii) to estimate the portfolio return based on parameters (β and r_m) estimated above.

Task #3 : Testing the CAPM theory

- The goal of this last part is to test the CAPM model predictions

1. For each year : compute β_i^{y-1} over 252 business days at the end of each year $y-1$ (OLS regression)
2. At the end or year y, compare the average return of the stock $\bar{r}_i^y$ (annualized) to the one predicted by the CAPM model $\hat{r}_M^y$ (using β_i^{y-1} measured at $y-1$ but the average market return $\bar{r}_M^y$ measured at y).
3. Generate the scatter plot of the realized returns against the beta or find innovative ways to plot you results.
4. Alternatively, you may use a β computed over longer (shorter) periods of time.
5. Comment your results in the light of yesterday's lecture : try to give as much economic interpretation as you can !

Optional Task : Obtain data from the web

- Web-scrapping : get the list of S&P 500 tickers from Wikipedia
- Use the list of tickers as input in the `yfinance` package (an API that gives access to Yahoo Finance data)
  - You need to pip install both `bs4` and `yfinance`
  - The API works like this : `data = yf.download(tickers, start=date1, end=date2)`
  - You could alternatively to pip install both `bs4` and `yfinance`
- Describe and comment the data. Why do you have more variables than in the `PCLab_Data.xlsx` file ?