

## Wk-05-Lab-Steps

Linking directory (on unix like systems)

```
ln -s <SOURCE> <LINK>
```

```
ln -s dataZ data
```

```
ln -s dataT data
```

Linking directory (on window like systems)

```
mklink /D <LINK> <SOURCE>
```

```
mklink /D data dataZ
```

```
mklink /D data dataT
```

to remove link use: **rmdir** data (be careful)

—> **do not use del**

---

You are supplied a working module that plots 2 scatterplot involving the symbols.

```
symbols = ['SPY', 'XOM', 'GLD'].
```

One scatterplot plots XOM against the benchmark SPY and the other GLD vs. SPY.

The data is from directory **dataT**. The model reads from a directory data, make

sure you link that directory (dataT) to data/ before running the program.

---

— For your LAB:

- You will evaluate symbols from directory **dataZ/** and evaluate (and rank) their:
  - alpha, beta and correlation coefficients against SPY (as X).
- You need to evaluate the symbols listed below against the benchmark SPY
  - symbols = ['SPY', 'TLT', 'GLD', 'AAPL', 'XOM', 'JPM', 'IBM', 'PYPL']
- Use the data range: **'2021-01-01', '2021-12-31'**

- Use the data **dataZ**

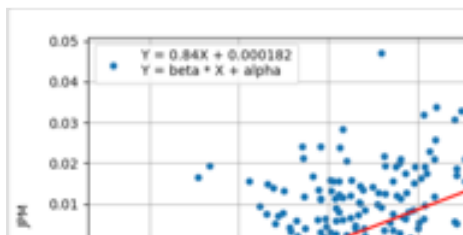
## Step 1 - Step 3 - Clean up the Code

---

**Step 1:** Add a legend that lists the regression equation on the plot. use label in .plot(), and use f-string. See Example 3 in below link.

<https://www.geeksforgeeks.org/matplotlib-pyplot-legend-in-python/>

You resulting legend should look like the below:



**Step 2:** Have `plot_scatter` return **beta, alpha**

**Step 3:** Create a frame **df**. that is indexed by 'symbols' and contains column names:

**alpha & beta**

**Step 4:** Create a **for** loop that calls `plot_scatter` iterative through the list of symbols, and fills in the values for alpha & beta each symbol.

**Step 4:** For the correlation matrix, only print the first row, that corresponds to the correlations with SPY. Add this column to df the 0-th position using insert. Call the column n.

```
df.insert(loc, [new column name], correlation_matrix.iloc[0],
allow_duplicates=False)
```

**Step 5:** convert each column to floats. Example for the alpha

column.

```
df.alpha = df.alpha.astype(float)
```

**Step 6:** Sort according each column, for each column display 2 top and 2 bottom ranked value.

Example: `df = df.sort_values(by=['alpha'], ascending=True)`

Example: `print(f"\nfinal_df = {df.head(2)}\n\n{df.tail(2)}")`

Columns: r, alpha, beta.

Example for r output should be:

===

```
df r =      r  alpha  beta
TLT -0.140287 0.000014 -0.150733
GLD 0.192728 -0.000470 0.201348
```

```
      r  alpha  beta
GOOG 0.688814 0.000808 1.261902
SPY 1.000000 0.000000 1.000000
```

===

Analyze - each heuristics according to the guide - fill in the below.

**top 2 - bottom 2.**

Symbol (Low to High)	<b>Correlation</b> Value (r)	Heuristics See r Guide
1.		
2.		
7.		
8.		

Symbol (Low to High)	$\beta$ - <b>Beta</b>	Heuristics See $\beta$ Guide
1.		
2.		
7.		
8.		
Symbol (Low to High)	$\alpha$ - <b>Alpha</b>	Heuristics See $\alpha$ Guide
1.		
2.		
7.		
8.		