

UCLA College | Social Sciences Economics



Day 8: Pandas pandas

Paul Schumacher, MSc Quantitative Economics

Announcements



Attendance

Final August 4, 4-6
(format same as midterm)

HW2 August 2, 11pm

**Midterms & HW1 is
graded → waiting for the
upload/ publication on Canvas**

Student Fellow Panel

- **Today 6:30-7:30**
- **Q&A with PLFs**
- <https://ucla.zoom.us/j/96582171841?pwd=OG1DUmpPUIMzSUF6eWlyNlcwOFBUUT09>

HW2: due Wednesday August 2, 11pm

Data Collection:

1. Construct a portfolio with any 10 stocks of your choice (use daily adjusted closing prices)

Python Task

2. Compute the **max Sharpe ratio** and **optimal weights**.
3. You will also need to **plot the respective efficient frontier**.
4. As part of your stock selection, you will need to **look at the pairwise correlations** to gain insights about your selection,
5. and **look at their individual past performance**.

Explanation:

6. As part of your presentation, you will need to **!!!!TELL THE STORY!!!!** of how you decided on the stocks that you included,

Class Recap: Pandas

- “Better version” of excel:
Larger Data Sets, More Applications
- Most popular library for data
manipulation/ analysis



**Day 4: Classes, File Operations,
Modules & Methods**

Paul Schumacher, MSc Quantitative Economics

Pandas: Loading data into Pandas

```
pip install pandas
```

```
import pandas as pd

df = pd.read_csv('pokemon_data.csv')
print(df.head(5))

df_xlsx = pd.read_excel('pokemon_data.xlsx')
print(df_xlsx.head(3))

df = pd.read_csv('pokemon_data.txt', delimiter='\t')
print(df.head(5))
```

Data set:

#		Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk
0	1	Bulbasaur	Grass	Poison	45	49	49	65
1	2	Ivysaur	Grass	Poison	60	62	63	80
2	3	Venusaur	Grass	Poison	80	82	83	100
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122
4	4	Charmander	Fire	NaN	39	52	43	60

	Sp. Def	Speed	Generation	Legendary
0	65	45	1	False
1	80	60	1	False
2	100	80	1	False
3	120	80	1	False
4	50	65	1	False

Pandas : Reading Data in Pandas

```
#### Read Headers
```

```
df.columns
```

```
Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk',  
      'Sp. Def', 'Speed', 'Generation', 'Legendary'],  
      dtype='object')
```

```
## Read Each Row
```

```
print(df.iloc[3:8])
```

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122
4	Charmander	Fire	NaN	39	52	43	60
5	Charmeleon	Fire	NaN	58	64	58	80
6	Charizard	Fire	Flying	78	84	78	109
7	CharizardMega Charizard X	Fire	Dragon	78	130	111	130

Sp. Def	Speed	Generation	Legendary	
3	120	80	1	False
4	50	65	1	False
5	65	80	1	False
6	85	100	1	False
7	85	100	1	False

```
## Read each Column
```

```
print(df[['Name', 'Type 1', 'Attack']])
```

	Name	Type 1	Attack
0	Bulbasaur	Grass	49
1	Ivysaur	Grass	62
2	Venusaur	Grass	82
3	VenusaurMega Venusaur	Grass	100
4	Charmander	Fire	52
..
795	Diancie	Rock	100
796	DiancieMega Diancie	Rock	160
797	HoopaHoopa Confined	Psychic	110
798	HoopaHoopa Unbound	Psychic	160
799	Volcanion	Fire	110

```
[800 rows x 3 columns]
```

Pandas : Reading Data in Pandas

```
In [12]: #get index and specific column
for index, row in df.iterrows():
    print(index, row['Name'])
```

```
0 Bulbasaur
1 Ivysaur
2 Venusaur
3 VenusaurMega Venusaur
4 Charmander
5 Charmeleon
6 Charizard
7 CharizardMega Charizard X
```

```
## Read a specific location (Rows,Column)
print(df.iloc[2,1])
```

```
Venusaur
```

```
#finding data that is not based on index/ instead --> based on type/characteristic
grass_pokemon = df.loc[df['Type 1'] == "Grass"]
grass_pokemon
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
48	43	Oddish	Grass	Poison	45	50	55	75	65	30	1	False

Pandas: Sorting/Describing Data

```
#list elements by name in ascending alphabetical order
df.sort_values('Name', ascending = False)
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
794	718	Zygarde50% Forme	Dragon	Ground	108	100	121	81	95	95	6	True
695	634	Zweilous	Dark	Dragon	72	85	70	65	70	58	5	False
46	41	Zubat	Poison	Flying	40	45	35	30	40	55	1	False
631	570	Zorua	Dark	NaN	40	65	40	80	40	65	5	False
632	571	Zoroark	Dark	NaN	60	105	60	120	60	105	5	False
...

```
df.describe()
```

	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
count	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000
mean	362.813750	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500	3.32375
std	208.343798	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474	1.66129
min	1.000000	1.000000	5.000000	5.000000	10.000000	20.000000	5.000000	1.00000
25%	184.750000	50.000000	55.000000	50.000000	49.750000	50.000000	45.000000	2.00000
50%	364.500000	65.000000	75.000000	70.000000	65.000000	70.000000	65.000000	3.00000
75%	539.250000	80.000000	100.000000	90.000000	95.000000	90.000000	90.000000	5.00000
max	721.000000	255.000000	190.000000	230.000000	194.000000	230.000000	180.000000	6.00000

```
df.sort_values(['Type 1', 'HP'], ascending=[1,0]) #first one ascending, second one descending
df
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False

Pandas: Making changes to the data

```
#Create the column "Total"
df['Total'] = df['HP'] + df['Attack'] + df['Defense'] + df['Sp. Atk'] + df['Sp. Def'] + df['Speed']

#Delete the column "Total"
df = df.drop(columns=['Total'])

#Create the column "Total" with iloc
df['Total'] = df.iloc[:, 4:10].sum(axis=1)# sum up specific values in columns

#Change the position of column "Total"
cols = list(df.columns)
df = df[cols[0:4] + [cols[-1]]+cols[4:12]]

df.head(5)
```

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0 1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1 2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2 3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3 3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4 4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

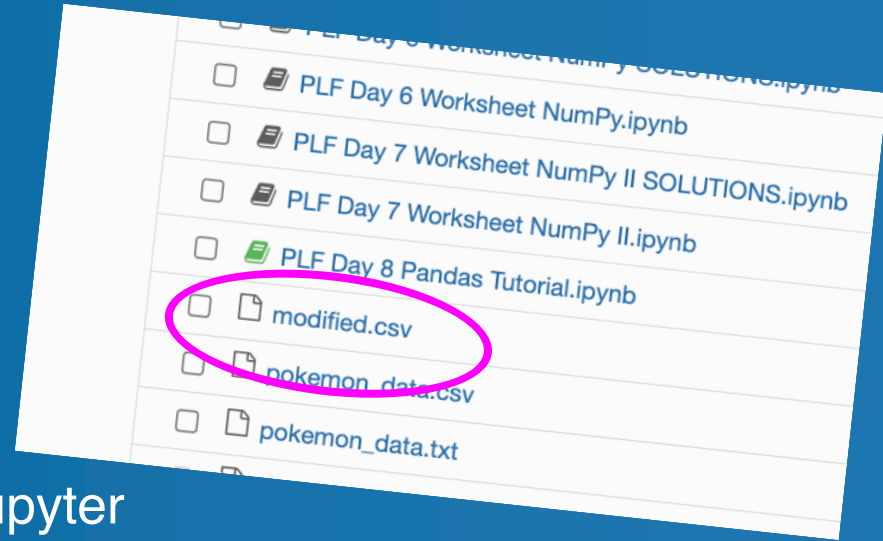
Always check
your code!!

45+49+49+65+65+45

318

Pandas: Saving our Data (Exporting into Desired Format)

```
df.to_csv('modified.csv', index=False)
df.to_excel('modified.xlsx', index=False)
df.to_csv('modified.txt', index=False, sep='\t')
```



The new file will be saved in Jupyter

Pandas: Filtering Data

```
#filter Pokemon that are type water OR poison
new_df = df.loc[(df['Type 1'] == 'Water') | (df['Type 2'] == 'Poison')]
new_df
```

#		Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
9	7	Squirtle	Water	NaN	44	48	65	50	64	43	1	False

```
#filter Pokemon that are type water AND poison
new_df = df.loc[(df['Type 1'] == 'Water') & (df['Type 2'] == 'Poison')]
new_df
```

#		Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
78	72	Tentacool	Water	Poison	40	40	35	50	100	70	1	False
79	73	Tentacruel	Water	Poison	80	70	65	80	120	100	1	False
227	211	Qwilfish	Water	Poison	65	95	75	55	55	85	2	False

Pandas: Conditional Changes

```
#if the pokemon has more then 500 in total, the value for "Generation" is changed  
#to "Test 1" and the value for "Legendary" is changed to "Hello"  
df.loc[df['Total'] > 500, ['Generation', 'Legendary']] = ['Test 1', 'hello']  
  
df  
  
#df = pd.read_csv('modified.csv')
```

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	Test 1	hello
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	Test 1	hello
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False
...

Think about how
many lines of code
we save here!

No "if statement"!

Pandas: Aggregate Statistics (Groupby)

```
df = pd.read_csv('modified.csv')

#add the column "count" and give every element the value "1"
df['count'] = 1

#group by "type 1" and count the total number of elements
df.groupby(['Type 1']).count()['count']

#group by "type 1" and "type 2" and count the respective total number of elements
df.groupby(['Type 1', 'Type 2']).count()['count']
```

Type 1	Type 2	count
Bug	Electric	2
	Fighting	2
	Fire	2
	Flying	14
	Ghost	1
Water	Ice	3
	Poison	3
	Psychic	5
	Rock	4
	Steel	1

Name: count, Length: 136, dtype: int64

Type 1	count
Bug	69
Dark	31
Dragon	32
Electric	44
Fairy	17
Fighting	27
Fire	52
Flying	4
Ghost	32
Grass	70
Ground	32
Ice	24
Normal	98
Poison	28
Psychic	57
Rock	44
Steel	27
Water	112

Name: count, dtype: int64

Class exercise: Pandas



- Data Set: “pokemon_data.csv”
- Recreate the EDA for yourself + Play around with the code
- Breakout rooms
- Time: 20 min



Any Questions?