

# Workshop I (100 pts)

---

## Task I - Install the Python IDE we will be using during the quarter. (10 pts)

I will be using Pycharm Edu during the class (however many other IDEs will work Anaconda,...).

The purpose of an IDE (Integrated Development Environment) is to help you to create your code infrastructure. It is the equivalent of R-Studio for R. It will help you by giving auto code completion, “compiling”, interpreting (in our case), installing libraries.

Take time to get familiar with the IDEs of your choice (we will also work with python 3.5 (or higher)).

Install Pycharm Edu for Windows/macOS/Linux  
<https://www.jetbrains.com/pycharm/download/>

To show that it works, just create your first code in python.

Example:

```
print ('that works')
```

When you run your code, you should display the following string on the stdout:

that works

## Task II – Refresh your R knowledge using Python (90pts)

### Part 1: Using modules (10 pts)

You will create a Python file by importing the following modules:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pylab
#Simple Linear Regression
import statsmodels.formula.api as smf
from pandas_datareader import data
```

## Part 2: Loading data (10pts)

You will load the following stocks between June 1<sup>st</sup>, 2014 and June 13<sup>th</sup>, 2016:

#1 American Airlines Group Inc. *AAL*

#2 Alaska Air Group, Inc. *ALK*

#3 Crude Oil WTI

You can use the following code:

```
all_data = {}
```

```
for ticker in ['AAL', 'ALK', 'WTI']:
```

```
    all_data[ticker] = data.DataReader(ticker, 'yahoo', '2014-06-01', '2016-06-13')
```

```
all_data['WTI'].head()
```

Date	Open	High	Low	Close	Volume	Adj Close
2014-06-02	14.67	14.79	14.32	14.33	562100	14.073248
2014-06-03	14.27	14.28	14.06	14.21	791900	13.955398
2014-06-04	14.22	14.25	14.00	14.12	601700	13.867010
2014-06-05	14.22	14.31	14.03	14.26	1205200	14.004502
2014-06-06	14.45	14.50	14.26	14.48	752600	14.220560

```
all_data['AAL'].head()
```

Date	Open	High	Low	Close	Volume	Adj Close
2014-06-02	40.000000	41.250000	40.000000	41.220001	9902100	40.213663
2014-06-03	41.130001	42.090000	41.110001	41.439999	9456200	40.428290
2014-06-04	41.360001	42.900002	41.200001	42.820000	11931500	41.774600
2014-06-05	43.000000	43.490002	42.270000	42.410000	8539600	41.374610
2014-06-06	42.450001	43.919998	42.419998	43.880001	13615400	42.808722

```
all_data['ALK'].head()
```

Date	Open	High	Low	Close	Volume	Adj Close
2014-06-02	98.720001	100.860001	98.089996	100.070000	1227600	48.447837
2014-06-03	100.099998	100.980003	99.500000	99.889999	1636600	48.360692
2014-06-04	99.739998	99.970001	98.919998	99.230003	1901000	48.041161
2014-06-05	99.889999	100.709999	98.559998	98.900002	1707600	47.881395
2014-06-06	99.349998	99.919998	98.750000	99.910004	1617200	48.370376

## Part 3: Using dataframe (20pts)

# Store the adjusted close prices into a DataFrame

```
price = pd.DataFrame({tic: data['Adj Close'] for tic, data in all_data.items()})
```

```
price.head(5)
```

Date	AAL	ALK	WTI
2014-06-02	40.213663	48.447837	14.073248
2014-06-03	40.428290	48.360692	13.955398
2014-06-04	41.774600	48.041161	13.867010
2014-06-05	41.374610	47.881395	14.004502
2014-06-06	42.808722	48.370376	14.220560

#### Part 4: Calculate Return (10pts)

Calculate the daily return using the adjusted close column (previously defined during the part 3) and the function `pct_change()`.

```
daily_return.head(5)
```

	AAL	ALK	WTI
Date			
2014-06-02	NaN	NaN	NaN
2014-06-03	0.005337	-0.001799	-0.008374
2014-06-04	0.033301	-0.006607	-0.006334
2014-06-05	-0.009575	-0.003326	0.009915
2014-06-06	0.034662	0.010212	0.015428

Tip: You just need to suffix price using `pct_change()`.

#### Part 5: Scatter plot (10pts)

Draw the scatter plot between the daily return of AAL and the daily return of WTI.

Tip: use the function:

```
plt.scatter
```

#### Part 6: Scatter plot (10pts)

Draw the scatter plot between the daily return of ALK and the daily return of WTI.

Tip: use the function:

```
plt.scatter
```

#### Part 7: Using linear regression (20pts)

With pandas, you can do the following code to do a linear regression:

```
df = pd.DataFrame({"AAL": [1,2,3,4,5], "ALK": [20, 30, 10, 40, 50],
...: "WTI": [32, 234, 23, 23, 42523]})
result = sm.ols(formula="WTI ~ AAL", data=df).fit()
print (result.params)
```

A- Calculate the Intercept and the coefficient for the linear regression between AAL and WTI:

```
print (result.params)
Intercept    6.576217
AAL          -0.010857
dtype: float64
```

B- Calculate the Intercept and the coefficient for the linear regression between ALK and WTI:

```
print (result.params)
Intercept    6.576217
AAL          -0.010857
dtype: float64
```