ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ενότητα 2 –Python για Μηχανική Μάθηση

Μέθοδοι Μηχανικής Μάθησης στα Χρηματοοικονομικά

Αθανάσιος Σάκκας, Επ. Καθηγητής, ΟΠΑ

Ι.Εισαγωγή στο Pandas

- Το Pandas είναι μια βιβλιοθήκη ανοιχτού κώδικα που παρέχει υψηλής απόδοσης, εύχρηστες δομές δεδομένων και εργαλεία ανάλυσης δεδομένων για τη γλώσσα προγραμματισμού Python. Βασίζεται στην έννοια του πλαισίου δεδομένων (data frame).
- Το πλαίσιο δεδομένων είναι ένα κρίσιμο συστατικό των Pandas.
- Ο ακόλουθος κώδικας φορτώνει το σύνολο δεδομένων σε ένα πλαίσιο δεδομένων:

Αρχείο 2.1 Introduction to Pandas

In [2]:
```
#Simple dataframe
import os
import pandas as pd

raw = pd.read_csv("https://raw.githubusercontent.com/thanossakkas/data/main/Demographic_S
# check the raw data
print("Size of the dataset (row, col): ", raw.shape)
print("\nFirst 5 rows\n", raw.head(n=5))
print("\nFirst 5 rows and 5 columns\n",raw .iloc[:5 , :5])
```

```
Size of the dataset (row, col):  (236, 46)

First 5 rows
    JURISDICTION NAME  COUNT PARTICIPANTS  COUNT FEMALE  PERCENT FEMALE  \
0              10001                  44            22            0.50
1              10002                  35            19            0.54
2              10003                   1             1            1.00
3              10004                   0             0            0.00
4              10005                   2             2            1.00

   COUNT MALE  PERCENT MALE  COUNT GENDER UNKNOWN  PERCENT GENDER UNKNOWN  \
0          22          0.50                     0                       0
1          16          0.46                     0                       0
2           0          0.00                     0                       0
3           0          0.00                     0                       0
4           0          0.00                     0                       0

   COUNT GENDER TOTAL  PERCENT GENDER TOTAL  ...  COUNT CITIZEN STATUS TOTAL  \
0                  44                   100  ...                          44
1                  35                   100  ...                          35
2                   1                   100  ...                           1
3                   0                     0  ...                           0
4                   2                   100  ...                           2

   PERCENT CITIZEN STATUS TOTAL  COUNT RECEIVES PUBLIC ASSISTANCE  \
0                           100                               20
1                           100                                2
2                           100                                0
```

```
First 5 rows and 5 columns
    JURISDICTION NAME  COUNT PARTICIPANTS  COUNT FEMALE  PERCENT FEMALE  \
0               10001                  44            22            0.50
1               10002                  35            19            0.54
2               10003                   1             1            1.00
3               10004                   0             0            0.00
4               10005                   2             2            1.00

   COUNT MALE
0          22
1          16
2           0
3           0
4           0
```

- Η λειτουργία **display** παρέχει «καθαρότερη» εικόνα από την απλή εκτύπωση του πλαισίου δεδομένων. Ο καθορισμός των μέγιστων γραμμών και στηλών σας επιτρέπει να επιτύχετε μεγαλύτερο έλεγχο στην οθόνη.

```
In [3]: pd.set_option('display.max_columns', 5)
        pd.set_option('display.max_rows', 5)
        display(raw)
```

	JURISDICTION NAME	COUNT PARTICIPANTS	...	COUNT PUBLIC ASSISTANCE TOTAL	PERCENT PUBLIC ASSISTANCE TOTAL
0	10001	44	...	44	100
1	10002	35	...	35	100
...	...	...	...	...	...
234	16091	0	...	0	0
235	20459	0	...	0	0

236 rows × 46 columns

- Υπολογισμός των Summary Statistics και Correlation matrix

```
In [4]: # print summary statistics
        print("\nSummary statistics\n", raw.describe())
        print("\nCorrelation matrix\n", raw.corr())
```

```
Summary statistics
        JURISDICTION NAME  COUNT PARTICIPANTS  ...  \
count         236.000000          236.000000  ...
mean        11127.173729           17.661017  ...
...                  ...                 ...  ...
75%         11422.250000           13.000000  ...
max         20459.000000          272.000000  ...

       COUNT PUBLIC ASSISTANCE TOTAL  PERCENT PUBLIC ASSISTANCE TOTAL
count                     236.000000                       236.000000
mean                       17.661017                        44.491525
...                              ...                              ...
75%                        13.000000                       100.000000
max                       272.000000                       100.000000

[8 rows x 46 columns]

Correlation matrix
```

- Είναι δυνατή η δημιουργία ενός δεύτερου πλαισίου δεδομένων για την εμφάνιση στατιστικών πληροφοριών σχετικά με το πρώτο πλαίσιο δεδομένων.

In [5]:
```
# Strip non-numerics
raw = raw.select_dtypes(include=['int', 'float'])

headers = list(raw.columns.values)
fields = []

for field in headers:
    fields.append({
        'name' : field,
        'mean': raw[field].mean(),
        'var': raw[field].var(),
        'sdev': raw[field].std()
    })

for field in fields:
    print(field)
```

```
{'name': 'JURISDICTION NAME', 'mean': 11127.17372881356, 'var': 1107612.671817526, 'sdev': 1052.431789627017}
{'name': 'COUNT PARTICIPANTS', 'mean': 17.661016949152543, 'var': 1873.135665344391, 'sdev': 43.279737352996854}
{'name': 'COUNT FEMALE', 'mean': 10.296610169491526, 'var': 794.6265416516422, 'sdev': 28.189120980471213}
{'name': 'PERCENT FEMALE', 'mean': 0.24398305084745767, 'var': 0.11134066354129064, 'sdev': 0.3336774843187515}
{'name': 'COUNT MALE', 'mean': 7.364406779661017, 'var': 356.48791922106005, 'sdev': 18.8808767036815}
{'name': 'PERCENT MALE', 'mean': 0.20101694915254234, 'var': 0.08745683375405687, 'sdev': 0.29573101588108214}
{'name': 'COUNT GENDER UNKNOWN', 'mean': 0.0, 'var': 0.0, 'sdev': 0.0}
{'name': 'PERCENT GENDER UNKNOWN', 'mean': 0.0, 'var': 0.0, 'sdev': 0.0}
{'name': 'COUNT GENDER TOTAL', 'mean': 17.661016949152543, 'var': 1873.135665344391, 'sdev': 43.279737352996854}
{'name': 'PERCENT GENDER TOTAL', 'mean': 44.49152542372882, 'var': 2480.1658853227614, 'sdev': 49.801263892824664}
{'name': 'COUNT PACIFIC ISLANDER', 'mean': 0.025423728813559324, 'var': 0.04190407500901555, 'sdev': 0.20470484852346696}
{'name': 'PERCENT PACIFIC ISLANDER', 'mean': 0.0002966101694915254, 'var': 5.443562928236557e-06, 'sdev': 0.00233144429356347
5}
{'name': 'COUNT HISPANIC LATINO', 'mean': 1.8559322033898304, 'var': 36.030219978362815, 'sdev': 6.002517803252466}
{'name': 'PERCENT HISPANIC LATINO', 'mean': 0.07983050847457628, 'var': 0.033005928597186936, 'sdev': 0.18167533843972036}
{'name': 'COUNT AMERICAN INDIAN', 'mean': 0.0211864406779661, 'var': 0.02933645870897932, 'sdev': 0.17127889160366294}
{'name': 'PERCENT AMERICAN INDIAN', 'mean': 0.001016949152542373, 'var': 0.0001717273710782533, 'sdev': 0.013104479046427343}
{'name': 'COUNT ASIAN NON HISPANIC', 'mean': 0.5254237288135594, 'var': 4.931265777136714, 'sdev': 2.2206453514995848}
{'name': 'PERCENT ASIAN NON HISPANIC', 'mean': 0.056567796610169474, 'var': 0.04006348900108193, 'sdev': 0.2001586595705565}
{'name': 'COUNT WHITE NON HISPANIC', 'mean': 12.190677966101696, 'var': 1593.9336999639365, 'sdev': 39.92409923797826}
{'name': 'PERCENT WHITE NON HISPANIC', 'mean': 0.17775423728813558, 'var': 0.12585408402452222, 'sdev': 0.3547591915997699}
```

- Αυτός ο κώδικας εξάγει μια λίστα λεξικών που περιέχουν αυτές τις στατιστικές πληροφορίες. Αυτές οι πληροφορίες μοιάζουν με τον κώδικα JSON.
- Το πρόγραμμα Python μπορεί να μετατρέψει αυτές τις πληροφορίες τύπου JSON σε πλαίσιο δεδομένων για καλύτερη εμφάνιση.

```
In [6]: pd.set_option('display.max_columns', 0)
        pd.set_option('display.max_rows', 0)
        raw2 = pd.DataFrame(fields)
        display(raw2)
```

	name	mean	var	sdev
0	JURISDICTION NAME	11127.173729	1.107613e+06	1052.431790
1	COUNT PARTICIPANTS	17.661017	1.873136e+03	43.279737
2	COUNT FEMALE	10.296610	7.946265e+02	28.189121
3	PERCENT FEMALE	0.243983	1.113407e-01	0.333677
4	COUNT MALE	7.364407	3.564879e+02	18.880888
5	PERCENT MALE	0.201017	8.745683e-02	0.295731
6	COUNT GENDER UNKNOWN	0.000000	0.000000e+00	0.000000
7	PERCENT GENDER UNKNOWN	0.000000	0.000000e+00	0.000000
8	COUNT GENDER TOTAL	17.661017	1.873136e+03	43.279737
9	PERCENT GENDER TOTAL	44.491525	2.480166e+03	49.801264
10	COUNT PACIFIC ISLANDER	0.025424	4.190408e-02	0.204705
11	PERCENT PACIFIC ISLANDER	0.000297	5.443563e-06	0.002333
...	...	...	...	...
34	COUNT CITIZEN STATUS UNKNOWN	0.000000	0.000000e+00	0.000000
35	PERCENT CITIZEN STATUS UNKNOWN	0.000000	0.000000e+00	0.000000
36	COUNT CITIZEN STATUS TOTAL	17.661017	1.873136e+03	43.279737
37	PERCENT CITIZEN STATUS TOTAL	44.487288	2.479698e+03	49.796563
38	COUNT RECEIVES PUBLIC ASSISTANCE	5.974576	2.823653e+02	16.803729
39	PERCENT RECEIVES PUBLIC ASSISTANCE	0.139195	5.197424e-02	0.227979
40	COUNT NRECEIVES PUBLIC ASSISTANCE	11.686441	8.625651e+02	29.369458
41	PERCENT NRECEIVES PUBLIC ASSISTANCE	0.305805	1.448576e-01	0.380602
42	COUNT PUBLIC ASSISTANCE UNKNOWN	0.000000	0.000000e+00	0.000000

Check for missing values

- Ο ευκολότερος τρόπος για να ελέγξετε για τιμές που λείπουν σε ένα πλαίσιο δεδομένων Pandas είναι μέσω της συνάρτησης isna(). Η συνάρτηση isna() επιστρέφει μια δυαδική τιμή (True ή False) εάν λείπει η τιμή της στήλης Pandas, επομένως εάν εκτελέσετε τη raw.isna() θα λάβετε πίσω ένα πλαίσιο δεδομένων που θα σας δείχνει ένα φόρτο δυαδικών τιμών.

```
In [9]: raw.isna().head()
```

Out[9]:

	JURISDICTION NAME	COUNT PARTICIPANTS	COUNT FEMALE	PERCENT FEMALE	COUNT MALE	PERCENT MALE	COUNT GENDER UNKNOWN	PERCENT GENDER UNKNOWN	COUNT GENDER TOTAL	PERCENT GENDER TOTAL	COUNT PACIFIC ISLANDER	PERCENT PACIFIC ISLANDER	CO HISP LA
0	False	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	False	

- Αυτό συνήθως δεν είναι πολύ χρήσιμο, επομένως θα υπολογίσουμε το sum() των τιμών που λείπουν εκτελώντας το raw.isna().sum(). Αυτό επιστρέφει τις στήλες στο πλαίσιο δεδομένων Pandas μαζί με τον αριθμό των τιμών που λείπουν που εντοπίστηκαν σε καθεμία, επομένως το 0 σημαίνει ότι δεν λείπουν τιμές και το 1 σημαίνει ότι λείπει μία τιμή.

```
In [17]: check = raw.isna().sum()
         check

Out[17]: JURISDICTION NAME                         0
         COUNT PARTICIPANTS                        0
         COUNT FEMALE                              0
         PERCENT FEMALE                            0
         COUNT MALE                                0
         PERCENT MALE                              0
         COUNT GENDER UNKNOWN                      0
         PERCENT GENDER UNKNOWN                    0
         COUNT GENDER TOTAL                        0
         PERCENT GENDER TOTAL                      0
         COUNT PACIFIC ISLANDER                    0
         PERCENT PACIFIC ISLANDER                  0
         COUNT HISPANIC LATINO                     0
         PERCENT HISPANIC LATINO                   0
         COUNT AMERICAN INDIAN                     0
                                                  ..
         PERCENT US CITIZEN                        0
         COUNT OTHER CITIZEN STATUS                0
         PERCENT OTHER CITIZEN STATUS              0
         COUNT CITIZEN STATUS UNKNOWN              0
         PERCENT CITIZEN STATUS UNKNOWN            0
         COUNT CITIZEN STATUS TOTAL                0
         PERCENT CITIZEN STATUS TOTAL              0
         COUNT RECEIVES PUBLIC ASSISTANCE          0
         PERCENT RECEIVES PUBLIC ASSISTANCE        0
         COUNT NRECEIVES PUBLIC ASSISTANCE         0
         PERCENT NRECEIVES PUBLIC ASSISTANCE       0
         COUNT PUBLIC ASSISTANCE UNKNOWN           0
         PERCENT PUBLIC ASSISTANCE UNKNOWN         0
         COUNT PUBLIC ASSISTANCE TOTAL             0
         PERCENT PUBLIC ASSISTANCE TOTAL           0
         Length: 46, dtype: int64
```

- Ένας γρήγορος τρόπος για να το αποθηκεύσετε το Dataframe σε CSV είναι ο ακόλουθος

```
In [18]: check.to_csv('checkformissing.csv')
```

Missing Values

- Τα missing values που λείπουν είναι μια πραγματικότητα. Στην ιδανική περίπτωση, κάθε σειρά δεδομένων θα έχει τιμές για όλες τις στήλες. Ωστόσο, αυτό συμβαίνει σπάνια.
- Μια πρακτική είναι κάνεις drop με την εντολή dropna(). Δείτε το *2.2 dropnanvalues.ipynb*

```
In [1]: #Simple dataframe
import os
import pandas as pd
raw = pd.read_csv("https://raw.githubusercontent.com/thanossakkas/data/main/GDP_EU.csv")
# check the raw data
print("Size of the dataset (row, col): ", raw.shape)
print("\nFirst 5 rows\n", raw.head(n=5))
print("\nFirst 5 rows and 5 columns\n",raw .iloc[:5 , :5])

Size of the dataset (row, col):  (52, 4)

First 5 rows
    TIME          AUT          BEL  CZE
0  1970  3829.731721  3876.367950  NaN
1  1971  4210.441192  4210.433244  NaN
2  1972  4638.534351  4606.148069  NaN
3  1973  5103.430018  5140.661848  NaN
4  1974  5772.158418  5820.638690  NaN

First 5 rows and 5 columns
    TIME          AUT          BEL  CZE
0  1970  3829.731721  3876.367950  NaN
1  1971  4210.441192  4210.433244  NaN
2  1972  4638.534351  4606.148069  NaN
3  1973  5103.430018  5140.661848  NaN
4  1974  5772.158418  5820.638690  NaN

In [3]: rawwithoutnan= raw.dropna()

In [6]: rawwithoutnan.to_csv('rawwithoutnan.csv')
```

- Μια άλλη πρακτική είναι η αντικατάσταση των τιμών που λείπουν με τη διάμεσο για αυτήν τη στήλη. Ο ακόλουθος κώδικας αντικαθιστά τυχόν τιμές nan με τη διάμεσο:
- Δείτε το *2.3 mediannanvalues.ipynb*

```
In [5]: #Simple dataframe
        import os
        import pandas as pd
        raw = pd.read_csv("https://raw.githubusercontent.com/thanossakkas/data/main/GDP_EU_missing.csv")
        raw
```

Με NaN

	TIME	AUT	BEL	CZE
0	1990	19473.50451	18687.89145	12689.40152
1	1991	20618.03694	19599.30247	11655.62003
2	1992	21293.74118	20269.73153	11850.36547
3	1993	21733.35372	20471.13838	12123.72874
4	1994	22643.34432	21518.98195	12736.02198
5	1995	23698.62927	22447.55031	13855.63267
6	1996	24561.16050	22744.39956	14689.90856
7	1997	25427.22674	23733.31705	14825.65829
8	1998	26676.24650	24370.04646	14977.57594
9	1999	27606.48424	25441.89139	15397.92282
10	2000	29380.03111	27789.05351	16210.13770
11	2001	29707.46226	28791.40584	17610.74958
12	2002	31178.05144	30281.66799	18245.66901
13	2003	NaN	30934.59859	19524.25779
14	2004	33784.43265	32063.67380	20912.04591
15	2005	35024.55748	33176.68088	22045.99888
16	2006	37659.84067	35253.92329	23855.58792
17	2007	39436.42013	36794.23420	NaN
18	2008	41316.02264	37883.23342	27853.54990
19	2009	40929.33675	37753.27627	27637.15866
20	2010	42020.55064	39837.99795	27768.00435
21	2011	44469.20964	40943.34348	28999.75476
22	2012	46477.65508	42290.47767	29258.90485
23	2013	47936.67796	43672.71229	30828.52641
24	2014	48813.53441	44929.93333	NaN
25	2015	49942.05629	46201.68589	33909.30924
26	2016	52665.08746	48599.20268	36101.28560

Αντικαθιστώντας τα NaN με τη διάμεσο της κάθε στήλης

	TIME	AUT	BEL	CZE
0	1990	19473.50451	18687.89145	12689.401520
1	1991	20618.03694	19599.30247	11655.620030
2	1992	21293.74118	20269.73153	11850.365470
3	1993	21733.35372	20471.13838	12123.728740
4	1994	22643.34432	21518.98195	12736.021980
5	1995	23698.62927	22447.55031	13855.632670
6	1996	24561.16050	22744.39956	14689.908560
7	1997	25427.22674	23733.31705	14825.658290
8	1998	26676.24650	24370.04646	14977.575940
9	1999	27606.48424	25441.89139	15397.922820
10	2000	29380.03111	27789.05351	16210.137700
11	2001	29707.46226	28791.40584	17610.749580
12	2002	31178.05144	30281.66799	18245.669010
13	2003	37659.84067	30934.59859	19524.257790
14	2004	33784.43265	32063.67380	20912.045910
15	2005	35024.55748	33176.68088	22045.998880
16	2006	37659.84067	35253.92329	23855.587920
17	2007	39436.42013	36794.23420	21479.022395
18	2008	41316.02264	37883.23342	27853.549900
19	2009	40929.33675	37753.27627	27637.158660
20	2010	42020.55064	39837.99795	27768.004350
21	2011	44469.20964	40943.34348	28999.754760
22	2012	46477.65508	42290.47767	29258.904850
23	2013	47936.67796	43672.71229	30828.526410
24	2014	48813.53441	44929.93333	21479.022395
25	2015	49942.05629	46201.68589	33909.309240
26	2016	52665.08746	48599.20268	36101.285600

Dealing with Outliers

- Οι ακραίες τιμές (outliers) είναι τιμές που είναι ασυνήθιστα υψηλές ή χαμηλές. Μερικές φορές οι ακραίες τιμές είναι απλώς σφάλματα, αυτό είναι αποτέλεσμα λάθους παρατήρησης. Οι ακραίες τιμές μπορεί επίσης να είναι πραγματικά μεγάλες ή μικρές τιμές που μπορεί να είναι δύσκολο να αντιμετωπιστούν.
- Επιλογές για την αντιμετώπισή τους:¶

α. Διορθώστε τα δεδομένα: Δείτε τα δεδομένα και διορθώστε τα. Μπορεί να είναι δαπανηρή ή αδύνατη αυτή η αντιμετώπιση.

β. Trimming: Διαγραφή παρατηρήσεων που είναι ακραίες.

γ.Winsorization: Αλλάξτε την τιμή έτσι ώστε να είναι πιο κοντά στην υπόλοιπη κατανομή

- Παράδειγμα: Οποιαδήποτε τιμή πάνω από το 99ο εκατοστημόριο για μια μεταβλητή αλλάζει ώστε να ισούται με το 99ο εκατοστημόριο.
- Αυτή είναι μια συνηθισμένη και χωρίς κόστος (ad-hoc) διόρθωση που υποβαθμίζει το βάρος της ακραίας τιμής στην ανάλυσή σας επειδή οι τιμές μειώνονται, χωρίς να απορρίπτεται εντελώς η παρατήρηση.
- Δύσκολη ερώτηση που εξαρτάται από τα δεδομένα/εφαρμογή: Ποιο είναι το «σωστό» ποσό της διανομής για winsorize.

Δείτε το αρχείο *2.4 removeoutliers.ipynb*

Removing duplicates

- **1η μέθοδος: Χρησιμοποιώντας *set()**

Καταργεί πρώτα τα διπλότυπα και επιστρέφει ένα λεξικό που πρέπει να μετατραπεί σε λίστα.

- **2η μέθοδος: Χρησιμοποιώντας collections.OrderedDict.fromkeys()**

Καταργεί πρώτα τα διπλότυπα και επιστρέφει ένα λεξικό που πρέπει να μετατραπεί σε λίστα. Αυτό λειτουργεί καλά και στην περίπτωση των strings .

- **3η μέθοδος: Χρησιμοποιώντας numpy unique method**

Note: Install numpy module using command “pip install numpy”

Αυτή η μέθοδος χρησιμοποιείται όταν η λίστα περιέχει στοιχεία του ίδιου τύπου και χρησιμοποιείται για την κατάργηση διπλότυπων από τη λίστα. Πρώτα μετατρέπει τη λίστα σε έναν πίνακα numpy και στη συνέχεια χρησιμοποιεί τη μέθοδο numpy unique() για να αφαιρέσει όλα τα διπλότυπα στοιχεία από τη λίστα.

Δείτε το αρχείο *2.5 removedupl.ipynb*

Μετατρέποντας ένα Dataframe σε Matrix

- Το πρόγραμμα χρησιμοποιεί την ιδιότητα values για να μετατρέψει τα δεδομένα σε μήτρα (matrix).
- Δείτε το αρχείο *2.6 dataframe2numeric.ipynb*

```
In [1]: #Simple dataframe
        import os
        import pandas as pd
        raw = pd.read_csv("https://raw.githubusercontent.com/thanossakkas/data/main/
```

```
In [2]: raw.values
```

```
Out[2]: array([[ 1990.     , 19473.50451, 18687.89145, 12689.40152],
               [ 1991.     , 20618.03694, 19599.30247, 11655.62003],
               [ 1992.     , 21293.74118, 20269.73153, 11850.36547],
               [ 1993.     , 21733.35372, 20471.13838, 12123.72874],
               [ 1994.     , 22643.34432, 21518.98195, 12736.02198],
               [ 1995.     , 23698.62927, 22447.55031, 13855.63267],
               [ 1996.     , 24561.1605 , 22744.39956, 14689.90856],
               [ 1997.     , 25427.22674, 23733.31705, 14825.65829],
               [ 1998.     , 26676.2465 , 24370.04646, 14977.57594],
               [ 1999.     , 27606.48424, 25441.89139, 15397.92282],
               [ 2000.     , 29380.03111, 27789.05351, 16210.1377 ],
               [ 2001.     , 29707.46226, 28791.40584, 17610.74958],
               [ 2002.     , 31178.05144, 30281.66799, 18245.66901],
               [ 2003.     ,         nan, 30934.59859, 19524.25779],
               [ 2004.     , 33784.43265, 32063.6738 , 20912.04591],
               [ 2005.     , 35024.55748, 33176.68088, 22045.99888],
               [ 2006.     , 37659.84067, 35253.92329, 23855.58792],
               [ 2007.     , 39436.42013, 36794.2342 ,         nan],
               [ 2008.     , 41316.02264, 37883.23342, 27853.5499 ],
               [ 2009.     , 40929.33675, 37753.27627, 27637.15866],
               [ 2010.     , 42020.55064, 39837.99795, 27768.00435],
               [ 2011.     , 44469.20964, 40943.34348, 28999.75476],
               [ 2012.     , 46477.65508, 42290.47767, 29258.90485],
               [ 2013.     , 47936.67796, 43672.71229, 30828.52641],
               [ 2014.     , 48813.53441, 44929.93333,         nan],
               [ 2015.     , 49942.05629, 46201.68589, 33909.30924],
               [ 2016.     , 52665.08746, 48599.20268, 36101.2856 ],
               [ 2017.     , 54188.36067, 50442.94752, 38842.89627],
               [ 2018.     , 56956.11056,         nan, 41157.37041],
               [ 2019.     , 59719.33165, 55800.82599, 44223.08816],
               [ 2020.     , 57253.30056, 54539.03253, 42813.74134],
               [ 2021.     , 59976.26467, 58806.11925, 44801.66688]])
```

Πηγές δεδομένων

- http://www.bls.gov - Bureau of Labor Statistics
- http://www.federalreserve.gov - Federal Reserve Board
- http://research.stlouisfed.org/fred2 - Federal Reserve Bank of St. Louis
- http://www.nationwide.co.uk/hpi/datadownload/data_download.htm - Nationwide
- http://www.oanda.com/convert/fxhistory - Oanda
- http://finance.yahoo.com - Yahoo! Finance
- http://www.dallasfed.org/ - Federal Reserve of Bank of Dallas
- http://www.bankofengland.co.uk/Pages/home.aspx - Bank of England
- https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html - Kenneth R. French - Data Library

- Βάσεις Δεδομένων με Συνδρομή: Thomson Reuters Eikon, Bloomberg, etc.

«Κατεβάζοντας» δεδομένα από το Yahoo Finance

Ανοίξτε το αρχείο *2.7 retrieve_yahoodata.ipynb*

In [29]:
```
from pandas_datareader import data as pdr
import yfinance as yf

yf.pdr_override()

stocks = ['msft', 'aapl', 'twtr', 'intc', 'tsm', 'goog', 'amzn', 'nvda']
start = datetime.datetime(2012,5,31)
end = datetime.datetime(2023,1,1)

yahoodata_specificperiod = pdr.get_data_yahoo(stocks, start=start, end=end)
yahoodata_maxperiod = pdr.get_data_yahoo(stocks,period="max")
```

```
[**********100%**********]  8 of 8 completed
[**********100%**********]  8 of 8 completed
```

In [30]:
```
yahoodata_specificperiod.to_csv('yahoodata_specificperiod.csv')
yahoodata_maxperiod.to_csv('yahoodata_fullperiod.csv')
```

«Κατεβάζοντας» δεδομένα από το Kenneth R. French - Data Library

- Δείτε το αρχείο *2.8 FF_factors.ipynb*

In [9]:
```
import urllib.request
import zipfile
ff_url = "http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/Europe_5_Factors_D
# Download the file and save it
# We will name it fama_french.zip file
urllib.request.urlretrieve(ff_url,'Europe_5_Factors_Daily_CSV.zip')
zip_file = zipfile.ZipFile('Europe_5_Factors_Daily_CSV.zip', 'r')
# Next we extact the file data
# We will call it ff_factors.csv
zip_file.extractall()
# Make sure you close the file after extraction
zip_file.close()
import pandas as pd
ff_factors = pd.read_csv('Europe_5_Factors_Daily.csv', skiprows = 3)
print(ff_factors.head())
```

```
   Unnamed: 0  Mkt-RF   SMB   HML   RMW   CMA    RF
0    19900702    0.99  0.00 -0.56  0.43 -0.25  0.03
1    19900703    0.33 -0.09  0.00  0.02  0.28  0.03
2    19900704    0.24  0.03 -0.19 -0.09  0.23  0.03
3    19900705   -0.64  0.22  0.08 -0.36  0.07  0.03
4    19900706    0.07 -0.24  0.13  0.02  0.11  0.03
```

In [10]:
```
print(ff_factors.iloc[1112:1120],)
```

```
      Unnamed: 0  Mkt-RF   SMB   HML   RMW   CMA    RF
1112    19941005   -1.32  0.83 -0.16  0.24 -0.09  0.02
1113    19941006    0.39 -0.57 -0.17  0.14  0.05  0.02
1114    19941007   -0.18 -0.26 -0.10  0.26 -0.05  0.02
1115    19941010    1.38 -1.24  0.40 -0.09 -0.01  0.02
1116    19941011    0.69 -0.74 -0.15  0.16 -0.09  0.02
1117    19941012    0.59 -0.05 -0.23  0.05 -0.10  0.02
1118    19941013    1.59 -0.91  0.20 -0.08 -0.05  0.02
1119    19941014    0.08  0.53  0.05 -0.17  0.01  0.02
```

In [11]:
```
ff_factors.to_csv('ff_factors.csv')
```

II. Categorical και Continuous Values

- Προτού εξετάσουμε συγκεκριμένους τρόπους επεξεργασίας δεδομένων, είναι σημαντικό να εξετάσουμε τέσσερις βασικούς τύπους δεδομένων:
- Character Data (strings)
  - **Nominal** - Individual discrete items, no order. Για παράδειγμα, color, zip code, shape.
  - **Ordinal** - Individual distinct items have an implied order. Για παράδειγμα grade level, job title, Starbucks coffee size (tall, vente, grande)
- Numeric Data
  - **Interval** - Numeric values, no defined start. Για παράδειγμα, temperature. Δε θα πείτε ποτέ, "yesterday was twice as hot as today."
  - **Ratio** - Numeric values, clearly defined start. Για παράδειγμα, speed. Θα πείτε "The first car is going twice as fast as the second."

- **Encoding Continuous Values**

Ένας κοινός μετασχηματισμός είναι η κανονικοποίηση normalization των εισόδων. Μερικές φορές είναι πολύτιμο για την κανονικοποίηση των αριθμητικών εισόδων να τίθενται σε τυπική μορφή, έτσι ώστε το πρόγραμμα να μπορεί εύκολα να συγκρίνει αυτές τις δύο τιμές. Σκεφτείτε αν κάποιος φίλος σας είπε ότι έλαβε έκπτωση 10 δολαρίων. Είναι αυτή μια καλή συμφωνία; Μπορεί. Αλλά το κόστος δεν κανονικοποιείται. Εάν ο φίλος σας αγόρασε ένα αυτοκίνητο, τότε η έκπτωση δεν είναι τόσο καλή. Εάν ο φίλος σας αγόρασε δείπνο, αυτή είναι μια εξαιρετική έκπτωση!

Τα ποσοστά είναι μια διαδεδομένη μορφή κανονικοποίησης. Εάν ο φίλος σας σας πει ότι έχει έκπτωση 10%, ξέρουμε ότι αυτή είναι μια καλύτερη έκπτωση από το 5%. Δεν έχει σημασία πόσο ήταν η τιμή αγοράς.

Μια ευρέως διαδεδομένη κανονικοποίηση μηχανικής μάθησης είναι το Z-Score:

$$\text{Z-score}: \quad \text{Value} \rightarrow \frac{\text{Value}-\text{Mean}}{\text{SD}}$$

- **Encoding Categorical Values as Dummies**

Το παραδοσιακό μέσο για την κωδικοποίηση κατηγορικών τιμών είναι η δημιουργία τους σε dummy variables.

- **Encoding Categorical Values as Ordinal**

Συνήθως οι κατηγορίες θα κωδικοποιούνται ως dummy variables. Ωστόσο, ενδέχεται να υπάρχουν άλλες τεχνικές για τη μετατροπή κατηγοριών σε αριθμητικές. Κάθε φορά που υπάρχει ένα order στις κατηγορίες, θα πρέπει να χρησιμοποιείται ένας αριθμός. Σκεφτείτε εάν είχατε μια κατηγορία που περιέγραφε το τρέχον επίπεδο εκπαίδευσης ενός ατόμου.

Kindergarten (0), First Grade (1), Second Grade (2), Third Grade (3), Fourth Grade (4), Fifth Grade (5), Sixth Grade (6), Seventh Grade (7), Eighth Grade (8), High School Freshman (9), High School Sophomore (10), High School Junior (11), High School Senior (12), College Freshman (13), College Sophomore (14), College Junior (15), College Senior (16), Graduate Student (17), PhD Candidate (18), Doctorate (19), Post Doctorate (20)

Η παραπάνω λίστα έχει 21 επίπεδα. Αυτό θα χρειαζόταν 21 dummy μεταβλητές. Ωστόσο, η απλή κωδικοποίηση αυτού σε dummy θα χάσει τις πληροφορίες του ordering. Ίσως η πιο εύκολη προσέγγιση θα ήταν να τους εκχωρήσετε απλώς τον αριθμό και να εκχωρήσετε στην κατηγορία έναν μοναδικό αριθμό που είναι ίσος με την τιμή στην παραπάνω παρένθεση. Ωστόσο, ίσως μπορέσουμε να τα πάμε ακόμα καλύτερα. Ο προπτυχιακός φοιτητής είναι πιθανό να υπερβαίνει το ένα έτος, επομένως μπορεί να αυξήσετε περισσότερο την τιμή που του αναλογεί.

III. Grouping, Sorting, and Shuffling

- Η **ταξινόμηση** του συνόλου δεδομένων σάς επιτρέπει να ταξινομήσετε τις σειρές με αύξουσα ή φθίνουσα σειρά για μία ή περισσότερες στήλες.

Δείτε το αρχείο *2.9 sorting.ipynb*

- Η **ομαδοποίηση** είναι μια τυπική λειτουργία σε σύνολα δεδομένων.

Δείτε το αρχείο *2.10 grouping.ipynb*

- **Ανακάτεμα** ενός συνόλου δεδομένων

Δείτε το αρχείο *2.11 shuffling.ipynb*