25. Επιστημη Δεδομενων και Pandas II

25.0.1 Λύσεις των προηγούμενων ασκήσεων

Άσκηση 1:

Αντικαταστήστε όλες τις κενές τιμές στη στήλη 'Calories' με την τιμή 0.

Λύση

```
import pandas as pd

# Ανάγνωση του dataset
df = pd.read_csv('data.csv')

# Αντικατάσταση των κενών τιμών στη στήλη 'Calories'
με το 0
df['Calories'].fillna(0, inplace=True)

# Εκτύπωση του νέου DataFrame
print(df.to_string())
```

Ένα τμήμα της εκτύπωσης που παίρνουμε είναι:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Progr
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       0.0
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
21         45    100       119     282.0
22         60    130       101     300.0
23         45    105       132     246.0
24         60    102       126     334.5
25         60    100       120     250.0
26         60     92       118     241.0
27         60    103       132       0.0
28         60    100       132     280.0
29         60    102       129     380.3
30         60     92       115     243.0
31         45     90       112     180.1
```

Άσκηση 2:

Αντικαταστήστε τις κενές τιμές στις στήλες 'Maxpulse' και'Pulse' με την τιμή mean κάθε στήλης

Λύση:

```
import pandas as pd

# Ανάγνωση του dataset
df = pd.read_csv('data.csv')

# Υπολογισμός της τιμής mean για την  'Maxpulse' και την 'Pulse'
mean_maxpulse = df['Maxpulse'].mean()
mean_pulse = df['Pulse'].mean()

# Αντικατάσταση των κενών τιμών στη 'Maxpulse' και την 'Pulse' με τις τιμές  means
df['Maxpulse'].fillna(mean_maxpulse, inplace=True)
df['Pulse'].fillna(mean_pulse, inplace=True)

# Εκτύπωση του νέου DataFrame
print(df.to_string())
```

Μέρος της εκτύπωσης που θα πάρουμε είναι:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Progr
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112       NaN
18        60    103       123     323.0
19        45     97       125     243.0
20        60    108       131     364.2
21        45    100       119     282.0
22        60    130       101     300.0
23        45    105       132     246.0
24        60    102       126     334.5
25        60    100       120     250.0
26        60     92       118     241.0
27        60    103       132       NaN
28        60    100       132     280.0
29        60    102       129     380.3
30        60     92       115     243.0
```

Άσκηση 3:

Αντικαταστήστε τις κενές τιμές στη στήλη 'Duration' με τη mode (πιο συχνή τιμή) της στήλης.

Λύση:

```
import pandas as pd

# Ανάγνωση του dataset
df = pd.read_csv('data.csv')

# Υπολογισμός της πιο συχνής τιμής για τη στήλη
'Duration'
mode_duration = df['Duration'].mode()[0]

# Αντικατάσταση των κενών τιμών με την τιμή mode, στη
στήλη 'Duration'
df['Duration'].fillna(mode_duration, inplace=True)

# Εκτύπωση του νέου DataFrame
print(df.to_string())
```

Μέρος της εξόδου είναι:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Progr
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
21         45    100       119     282.0
22         60    130       101     300.0
23         45    105       132     246.0
24         60    102       126     334.5
25         60    100       120     250.0
26         60     92       118     241.0
27         60    103       132       NaN
28         60    100       132     280.0
29         60    102       129     380.3
```

25.1.0 Pandas - Καθαρισμός δεδομένων λανθασμένης μορφής

Συνεχίζουμε την ενημέρωση σε σχέση με τις δυνατότητες της βιβλιοθήκης Pandas.

Τα κελιά με δεδομένα λανθασμένης μορφής μπορεί να δυσκολέψουν, ή ακόμα και αδύνατη, την ανάλυση δεδομένων.

Για να το διορθώσουμε, έχουμε δύο επιλογές:. Είτε να αφαιρέσουμε τις σειρές είτε να μετατρέψουμε όλα τα κελιά των στηλών στην ίδια μορφή.

Πάμε να δούμε μια-μια τις λύσεις

25.1.1 Μετατροπή σε σωστή μορφή

Στο πλαίσιο δεδομένων μας, έχουμε δύο κελιά με λάθος μορφή. Αν δούμε τις σειρές 22 και 26 στη στήλη "Date", πρέπει να έχουμε συμβολοσειρές που αντιπροσωπεύουν ημερομηνίες,

Πρώτα τρέχουμε τον κώδικα:

```
import pandas as pd

df =
pd.read_csv('D:\\Cookoo_Home\\Documents\\Ευδόκιμος\\Παρουσιάσεις Μαθήματος Python\\25\\data.csv')

df['Date'] = pd.to_datetime(df['Date'])

print(df.to_string())
```

...όμως δεν τρέχει, καθότι δεν δέχεται τη μορφή της ημερομηνίας στην οποία θέλουμε να τροποποιήσουμε και δοκιμάζουμε με την παράμετρο format='mixed', η οποία και μας προτείνεται στο σφάλμα του IDLE.

Δηλ, συμπληρώνουμε στη γραμμή

```
df['Date'] = pd.to_datetime(df['Date'],
format='mixed')
```

Έτσι τώρα, έχουμε στην εκτύπωσή μας:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Programs\Pyth
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4         45 2020-12-05    117       148     406.0
5         60 2020-12-06    102       127     300.0
6         60 2020-12-07    110       136     374.0
7        450 2020-12-08    104       134     253.3
8         30 2020-12-09    109       133     195.1
9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
22        45        NaT    100       119     282.0
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
25        60 2020-12-25    102       126     334.5
26        60 2020-12-26    100       120     250.0
27        60 2020-12-27     92       118     241.0
28        60 2020-12-28    103       132       NaN
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
>>>
```

Τώρα όμως παρατηρούμε ότι στη γραμμή 22 έχουμε και πάλι μια κενή τιμή (NaT = Not a Time) και η λύση που μας μένει είναι να αφαιρέσουμε όλη τη γραμμή.

Διαμορφώνουμε λοιπόν τον κώδικά μας ως εξής:

import pandas as pd

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

df['Date'] = pd.to_datetime(df['Date'])

df.dropna(subset=['Date'], inplace = True)

print(df.to_string())
```

κα τώρα έχουμε:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Programs\Pyt
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4         45 2020-12-05    117       148     406.0
5         60 2020-12-06    102       127     300.0
6         60 2020-12-07    110       136     374.0
7        450 2020-12-08    104       134     253.3
8         30 2020-12-09    109       133     195.1
9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
25        60 2020-12-25    102       126     334.5
26        60 2020-12-26    100       120     250.0
27        60 2020-12-27     92       118     241.0
28        60 2020-12-28    103       132       NaN
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
>>>
```

Ας προσπαθήσουμε τώρα να μετατρέψουμε όλα τα κελιά της στήλης ‘Date’ σε ημερομηνίες. Υπάρχει μια ειδική μέθοδος γι’ αυτό, η to_datetime():

```
import pandas as pd

df = pd.read_csv('data.csv')

df['Date'] = pd.to_datetime(df['Date'])

print(df.to_string())
```

και πάλι πρέπει να κάνουμε την παραπάνω διόρθωση (format – ‘mixed’), και παίρνουμε:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Programs\Pyth
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4         45 2020-12-05    117       148     406.0
5         60 2020-12-06    102       127     300.0
6         60 2020-12-07    110       136     374.0
7        450 2020-12-08    104       134     253.3
8         30 2020-12-09    109       133     195.1
9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
22        45        NaT    100       119     282.0
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
25        60 2020-12-25    102       126     334.5
26        60 2020-12-26    100       120     250.0
27        60 2020-12-27     92       118     241.0
28        60 2020-12-28    103       132       NaN
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
>>>
```

Παρατηρούμε ότι διορθώθηκε η ημερομηνία στη γραμμή 25, αλλά η κενή ημ/νία στη γραμμή 22 μας έδωσε ένα αποτέλεσμα NaT (Not a Time), μ’ άλλα λόγια μια κενή τιμή και όπως ήδη έχουμε πει, ένας τρόπος να αντιμετωπίζουμε τις κενές τιμές, είναι να διαγράφουμε όλη τη γραμμή.

25.1.2 Διαγραφή γραμμών

Πάμε να το δούμε. Ας αφαιρέσουμε τη γραμμή με την κενή τιμή:

```
import pandas as pd

df =
pd.read_csv('D:\\Cookoo_Home\\Documents\\Ευδόκιμος\\Παρουσιάσεις Μαθήματος Python\\25\\data.csv')

df['Date'] = pd.to_datetime(df['Date'], format =
'mixed')

df.dropna(subset=['Date'], inplace = True)

print(df.to_string())
```

κι έτσι τώρα έχουμε:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Programs\Pyth
    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4         45 2020-12-05    117       148     406.0
5         60 2020-12-06    102       127     300.0
6         60 2020-12-07    110       136     374.0
7        450 2020-12-08    104       134     253.3
8         30 2020-12-09    109       133     195.1
9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
25        60 2020-12-25    102       126     334.5
26        60 2020-12-26    100       120     250.0
27        60 2020-12-27     92       118     241.0
28        60 2020-12-28    103       132       NaN
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
>>>
```

25.1.3 Διόρθωση λανθασμένων δεδομένων

Τα λανθασμένα δεδομένα δεν είναι απαραίτητο να είναι κενές τιμές ή λάθος τύπος δεδομένων, μπορεί απλά να είναι λανθασμένα.

Κάποιες φορές, μπορούμε να διακρίνουμε αυτά τα δεδομένα απλώς κοιτάζοντας το data set, καθότι αντιλαμβανόμαστε τι μορφής πρέπει να είναι τα δεδομένα.

Αν δούμε τα προηγούμενα μόλις δεδομένα, θα δούμε ότι στη γραμμή 7, η διάρκεια (Duration) είναι 450, ενώ όλες οι άλλες τιμές είναι μεταξύ του 30 και του 60.

Δεν είναι απαραίτητο να είναι λάθος, αλλά έχοντας για παράδειγμα τη γνώση ότι αυτό είναι ένα data set για την ώρα εκγύμνασης κάποιου, συμπεραίνουμε πως η τιμή είναι λάθος, καθότι είναι σχετικά απρόσμενο το άτομο να γυμνάστηκε για 450 λεπτά μόνο για μια μέρα.

Πώς μπορούμε να διορθώσουμε τέτοιου είδους λανθασμένα δεδομένα;

25.1.4 Αντικατάσταση τιμών

Στο παράδειγμά μας, κατά πάσα πιθανότητα πρόκειται για ένα λάθος πληκτρολόγησης, όπου αντί για «45», πληκτρολογήθηκε «450».

Οπότε, μπορούμε να πάρουμε την απόφαση να αντικαταστήσουμε τη συγκεκριμένη τιμή στη γραμμή 7, με την τιμή «45».

Σε κώδικα έχουμε: `df.loc[7, 'Duration'] = 45`

Δηλ.:

```
import pandas as pd

df = pd.read_csv('data.csv')

df.loc[7,'Duration'] = 45

print(df.to_string())
```

Σε μικρά data sets μπορούμε να αντικαταστήσουμε τα αλνθασμένα δεδομένα, ένα προς ένα. Φυσικά αυτό είναι αδύνατο σε μεγαλύτερα data sets.

Σε τέτοιες περιπτώσεις μπορούμε να δημιουργήσουμε κάποιους κανόνες, για παράδειγμα να βάλουμε κάποια όρια για τις αποδεκτές τιμές και να αντικαταστήσουμε οποιεσδήποτε τιμές είναι έξω απ' αυτά.

Στο παρακάτω παράδειγμα, εφαρμόζουμε ένα loop στις τιμές της στήλης "Duration" και αν η τιμή είναι μεγαλύτερη από 120, την ορίζουμε σε «120»:

```
import pandas as pd

df = pd.read_csv('data.csv')

for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.loc[x, "Duration"] = 120

print(df.to_string())
```

25.1.5 Αφαίρεση γραμμών

Όπως ειπώθηκε και προηγουμένως, ένας άλλος τρόπος να αντιμετωπίσουμε τα λανθασμένα δεδομένα, είναι να διαγράψουμε τις γραμμές. Μ' αυτό τον τρόπο, δεν χρειάζεται να βρούμε μια τιμή αντικατάστασης και το επιλέγουμε αυτό, όταν κρίνουμε πως η συγκεκριμένη διαγραφή δεν θα επηρεάσει το αποτέλεσμα της ανάλυσής μας.

Έτσι, μπορούμε να εφαρμόσουμε το παραπάνω loop, ως εξής:

import pandas as pd

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

for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.drop(x, inplace = True)

# Πρέπει να θυμόμαστε να συμπεριλαμβάνουμε τη δήλωση
'inplace = True' ώστε οι αλλαγές να γίνονται στο
αρχικό DataFrame αντί να παίρνουμε ένα αντίγραφό του

print(df.to_string())
```

25.1.6 Αφαίρεση Διπλοτύπων

Πρόκειται για τις γραμμές οι οποίες έχουν καταχωρηθεί πάνω από μια φορά. Κοιτάζοντας το δοκιμαστικό μας dataset, μπορούμε να δούμε ότι οι γραμμές 11 και 12 είναι διπλότυπα.

Για να τα ανακαλύψουμε, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση duplicated(), η οποία επιστρέφει τιμές Boolean για κάθε γραμμή, δηλαδή, True για κάθε γραμμή που είναι διπλή, αλλιώς False:

```
import pandas as pd

df = pd.read_csv('data.csv')

print(df.duplicated())
```

Ένα δείγμα της εκτύπωσης του παραπάνω κώδικα είναι:

```
>>>
=== RESTART:
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12     True
13    False
14    False
```

Μπορούμε επίσης να απομακρύνουμε τελείως τις διπλότυπες εγγραφές:

```
import pandas as pd

df = pd.read_csv('data.csv')

df.drop_duplicates(inplace = True)

print(df.to_string())
```

Εύρεση σχέσεων – Data Correlations

Μια γνωστή μέθοδος της βιβλιοθήκης Pandas είναι η corr(). Αυτή η μέθοδος υπολογίζει τη σχέση κάθε στήλης με την άλλη.

Κατεβάζουμε ένα νέο αρχείο data.csv από δω.

Ας την τρέξουμε για να σχολιάσουμε την εκτύπωση:

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Programs\Py
             Duration     Pulse  Maxpulse  Calories
Duration     1.000000 -0.155408  0.009403  0.922717
Pulse       -0.155408  1.000000  0.786535  0.025121
Maxpulse     0.009403  0.786535  1.000000  0.203813
Calories     0.922717  0.025121  0.203813  1.000000

>>>
```

ΣΗΜΕΙΩΣΗ: Η μέθοδος corr() αγνοεί τις μη αριθμητικές στήλες.

25.1.7 Επεξήγηση αποτελέσματος

Το αποτέλεσμα της corr()μεθόδου είναι ένας πίνακας με πολλούς αριθμούς που αντιπροσωπεύει πόσο καλή είναι η σχέση μεταξύ δύο στηλών.

Κάθε αριθμός ποικίλλει από το -1 έως το 1.

Το 1 σημαίνει ότι υπάρχει μια σχέση 1 προς 1 (μια τέλεια συσχέτιση) και για αυτό το data set, κάθε φορά που μια τιμή ανέβαινε στην πρώτη στήλη, ανέβαινε και η τιμή στην άλλη.

Το 0,9 είναι επίσης μια καλή σχέση, και αν αυξήσουμε τη μία τιμή, πιθανότατα θα αυξηθεί και η άλλη.

Το -0,9 θα ήταν εξίσου καλή σχέση με το 0,9, αλλά αν αυξήσετε τη μία τιμή, η άλλη πιθανότατα θα μειωθεί.

Το 0,2 σημαίνει ΟΧΙ καλή σχέση, που σημαίνει ότι αν η μία τιμή ανεβαίνει δεν σημαίνει ότι και η άλλη θα αυξηθεί.

Τι είναι ένας καλός συσχετισμός; Εξαρτάται από τη χρήση, αλλά είναι ασφαλές να πούμε ότι πρέπει να έχουμε τουλάχιστον 0.6(ή -0.6) για να έχουμε έναν καλό συσχετισμό.

25.1.8 Τέλεια συσχέτιση

Συνεχίζοντας τον σχολιασμό μας, παρατηρούμε ότι η στήλες «Duration” και “Duration” έχουν την τέλεια συσχέτιση (1). Αυτό είναι φυσιολογικό για κάθε στήλη, καθότι η καθεμιά συσχετίζεται με τον εαυτό της.

25.1.9 Καλή συσχέτιση

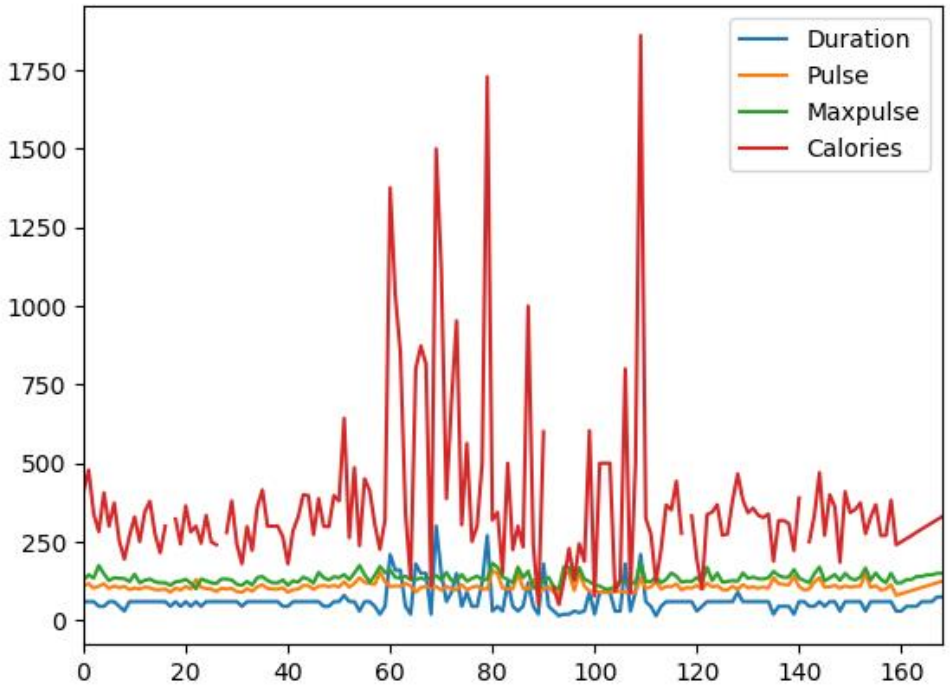
Η στήλη "Duration" και "Calories" έχουν μια συσχέτιση 0.922721, η οποία είναι ένας πολύ καλός συσχετισμός, και μπορούμε να προβλέψουμε ότι όσο περισσότερο γυμνάζεται κάποιος, τόσο περισσότερες θερμίδες καίει και το αντίστροφο: αν κάψουμε πολλές θερμίδες, μάλλον εξασκηθήκαμε για πολλή ώρα.

25.1.10 Κακή συσχέτιση

Οι στήλες "Duration" και "Maxpulse" έχουν μια συσχέτιση 0.009403, η οποία είναι πολύ κακή συσχέτιση, που σημαίνει ότι δεν μπορούμε να προβλέψουμε τον μέγιστο παλμό κοιτάζοντας απλώς τη διάρκεια της εκγύμνασης και το αντίστροφο.

25.1.11 Pandas plotting

Ας δούμε το παρακάτω διάγραμμα:



25.1.12 Κατασκευή διαγράμματος

Το Pandas χρησιμοποιεί τη plot()μέθοδο για να δημιουργεί διαγράμματα.

Μπορούμε να χρησιμοποιήσουμε το Pyplot, μια υπομονάδα της βιβλιοθήκης Matplotlib για να απεικονίσουμε το διάγραμμα στην οθόνη.

Όμως, θα πρέπει πρώτα να εγκαταστήσουμε και κατόπιν να εισάγουμε τη βιβλιοθήκη:

```
pip install matplotlib
```

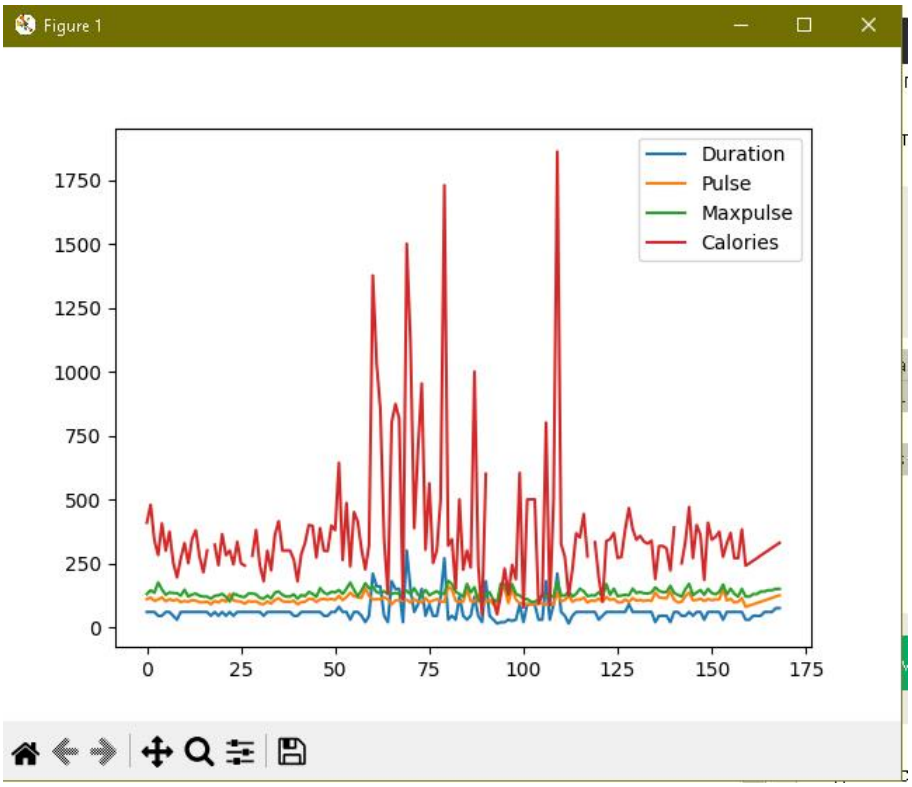
Εισαγάγετε pyplot από το Matplotlib και οπτικοποιήστε το DataFrame μας:

```
import pandas as pd import matplotlib.pyplot as plt

df = pd.read_csv('data.csv')

df.plot()

plt.show()
```



25.1.13 Διάγραμμα διασποράς

Για να ορίσουμε ένα διάγραμμα διασποράς χρησιμοποιούμε την έκφραση:

kind = 'scatter'

Ένα διάγραμμα διασποράς χρειάζεται άξονες x και y.

Στο παρακάτω παράδειγμα θα χρησιμοποιήσουμε τη στήλη "Duration" για τον άξονα x και "Calories" για τον άξονα y.

Ορίζουμε τα ορίσματα x και y ως εξής:

x = 'Duration', y = 'Calories'

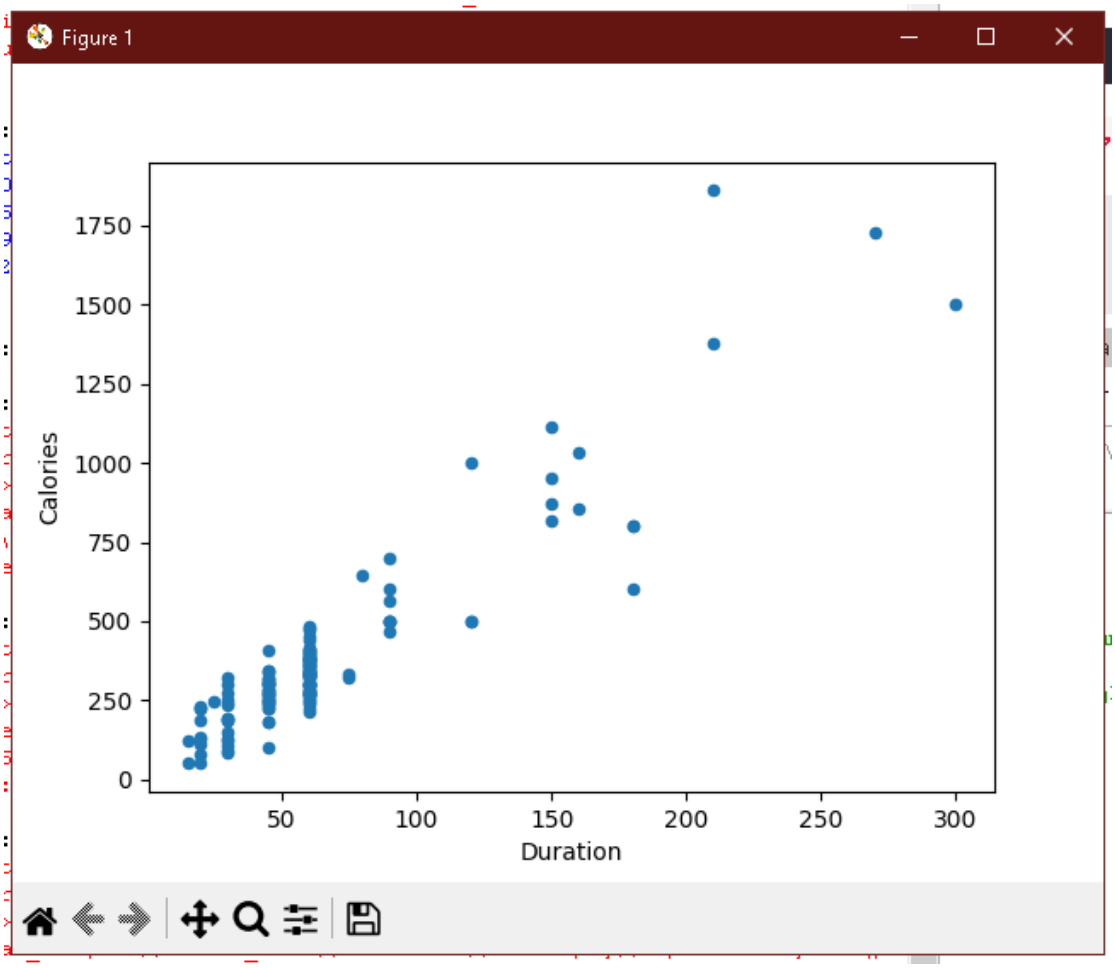
```
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_csv('D:\\Cookoo_Home\\Documents\\Ευδόκιμος\\Παρ
ουσιάσεις Μαθήματος Python\\25\\data.csv')

df.plot(kind = 'scatter', x = 'Duration', y =
'Calories')

plt.show()
```

και το διάγραμμα που παίρνουμε είναι:



ΣΗΜΕΙΩΣΗ: Στο προηγούμενο παράδειγμα, είδαμε ότι η συσχέτιση μεταξύ "Duration" και "Calories" ήταν 0.922721, και καταλήξαμε στο γεγονός ότι μεγαλύτερη διάρκεια σημαίνει περισσότερες θερμίδες που καίγονται. Βλέποντας το scatterplot, μπορούμε αμέσως να έχουμε αυτή την εικόνα.

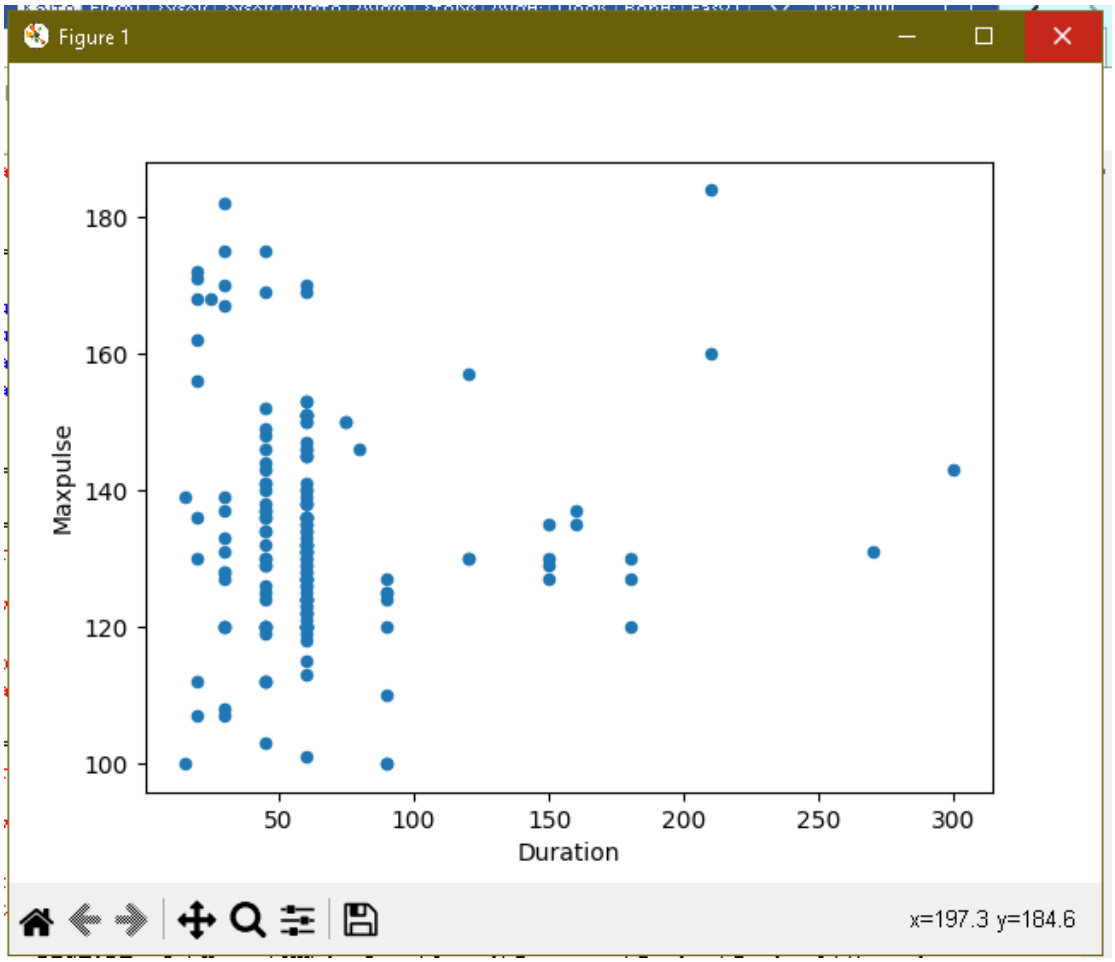
Ας δημιουργήσουμε ένα άλλο scatterplot, όπου υπάρχει κακή σχέση μεταξύ των στηλών, όπως , μεταξύ των στηλών"Duration" και "Maxpulse", με τη συσχέτιση 0.009403:

```
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_csv('D:\\Cookoo_Home\\Documents\\Ευδόκιμος\\Παρ
ουσιάσεις Μαθήματος Python\\25\\data.csv')

df.plot(kind = 'scatter', x = 'Duration', y =
'Maxpulse')

plt.show()
```



25.1.14 Δημιουργία ιστογράμματος

Θα χρησιμοποιήσουμε τώρα το όρισμα kind για να δηλώσουμε ότι θέλουμε ένα ιστόγραμμα:

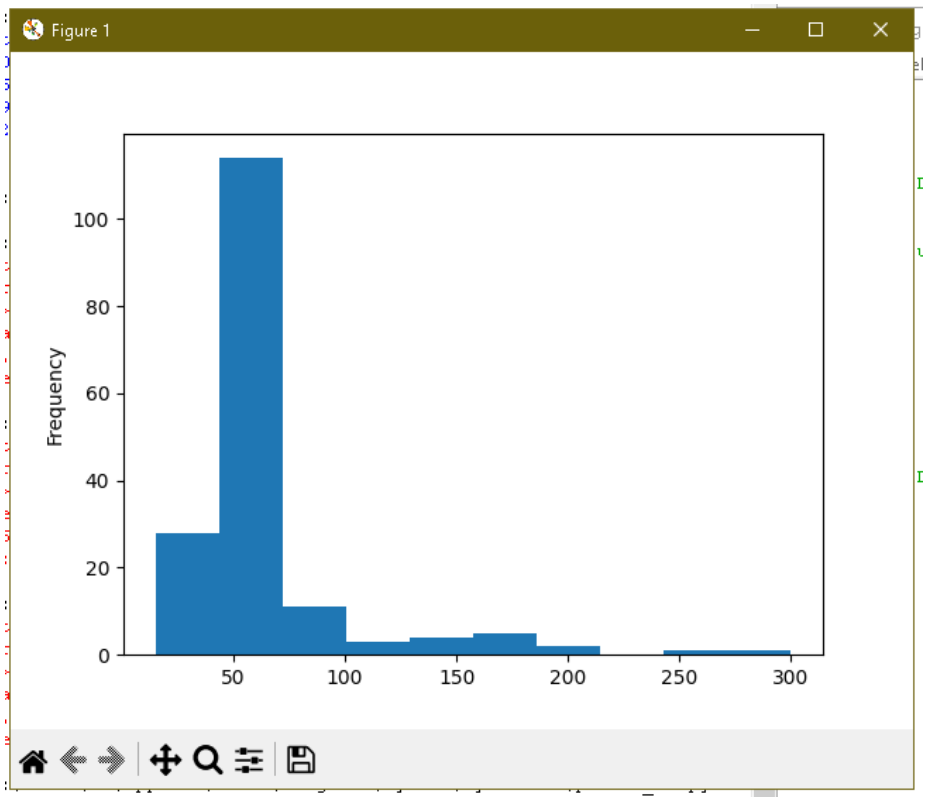
kind = 'hist'

Ένα ιστόγραμμα χρειάζεται μόνο μία στήλη και μας δείχνει τη συχνότητα κάθε διαστήματος, π.χ. πόσες προπονήσεις διήρκεσαν μεταξύ 50 και 60 λεπτών;

Στο παρακάτω παράδειγμα θα χρησιμοποιήσουμε τη στήλη "Duration" για να δημιουργήσουμε το ιστόγραμμα:

Παράδειγμα

```
df["Duration"].plot(kind = 'hist')
```



25.1.15 Ασκήσεις

Άσκηση 1

Δίνεται το λεξικό theater_event = {

```
    'Date': ['10/2/2011', '12/2/2011', '13/2/2011',
'14/2/2011'],
    'Event': ['Music', 'Poetry', 'Theatre', 'Comedy'],
    'Cost': [10000, 5000, 15000, 2000]
```

1. Δημιουργήστε ένα dataframe από το λεξικό.

2. Εισάγετε τη γραμμή στο index[2] του dataframe

```
row_value = ['11/2/2011', 'Wrestling', 12000]
```

Άσκηση 2

Δημιουργήστε ένα dataframe χρησιμοποιώντας την παρακάτω πολυδιάστατη λίστα;

```
lst = [['tom', 25], ['krish', 30],
       ['nick', 26], ['juli', 22]]
```

Το dataframe πρέπει να έχει δύο στήλες. Τις "name" και "age"

ΚΑΛΗ ΜΕΛΕΤΗ!
