
26. ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ PANDAS III

26.0.1 Λύσεις προηγούμενων ασκήσεων

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 την παρακάτω γραμμή στο index[2] του dataframe

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

Λύση

```

1.
# importing pandas as pd
import pandas as pd
# Ας δημιουργήσουμε το dataframe
df = pd.DataFrame({'Date':['10/2/2011', '12/2/2011',
'13/2/2011', '14/2/2011'], 'Event':['Music', 'Poetry',
'Theatre', 'Comedy'], 'Cost':[10000, 5000, 15000,
2000]})
# Εκτύπωση του dataframe
print(df)

```

Στην έξοδο έχουμε:

```

>>>
=== RESTART: C:\Users\NK\AppData
      Date      Event      Cost
0  10/2/2011    Music   10000
1  12/2/2011   Poetry    5000
2  13/2/2011  Theatre   15000
3  14/2/2011   Comedy    2000
>>>

```

2.

Τώρα θα πρέπει να προσθέσουμε στο index 2 τη γραμμή που μας δίνεται. Αν όμως το κάνουμε αυτό δίνοντας αυτό το index, τότε η γραμμή 2 θα διαγραφεί και από πάνω της θα γραφτεί αυτή που θα δώσουμε.

Έτσι, δίνουμε μια τιμή πολύ κοντινή σ' αυτή που θέλουμε να είναι το index και η Pandas τοποθετεί τη γραμμή ακριβώς εκεί που θέλουμε.

```
import pandas as pd
# Ας δημιουργήσουμε το dataframe
df = pd.DataFrame({'Date': ['10/2/2011', '12/2/2011', '13/2/2011', '14/2/2011'], 'Event': ['Music', 'Poetry', 'Theatre', 'Comedy'], 'Cost': [10000, 5000, 15000, 2000]})
# Τοποθετούμε τη νέα γραμμή "ανάμεσα" στα index 1 και 2
df.loc[1.5] = ['11/2/2011', 'Wrestling', 12000]
df = df.sort_index().reset_index(drop=True)
```

Έτσι, έχουμε το νέο dataframe

```
>>>
=== RESTART: C:\Users\NK\AppData\Local\Microsoft\Windows\Apps\PythonSoftwareFoundation.Python.3.9.0.0_qbz5n2kz8v0win\python.exe
      Date      Event  Cost
0  10/2/2011    Music  10000
1  12/2/2011    Poetry   5000
2  11/2/2011  Wrestling  12000
3  13/2/2011   Theatre  15000
4  14/2/2011    Comedy   2000
>>>
```

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"

```
import pandas as pd

# Η λίστα μας είναι
lst = [['tom', 25], ['krish', 30],
       ['nick', 26], ['juli', 22]]

df = pd.DataFrame(lst, columns=['Name', 'Age'])
print(df)
```

Και η έξοδος είναι:

```
>>>
--- RESTART: C:
      Name  Age
0     tom   25
1  krish   30
2   nick   26
3   juli   22
>>> |
```

26.1.0 Project με αληθινά στοιχεία

Θα χρησιμοποιήσουμε δεδομένα του πραγματικού κόσμου και θα δημιουργήσουμε ερωτήματα όπως θα κάναμε ίσως για μια εταιρία ή για κάποιον οργανισμό.

Χρησιμοποιούμε ένα σύνολο [δεδομένων](#) (από το Kaggle) με δασικές πυρκαγιές στη Βραζιλία, η οποία όπως γνωρίζουμε φιλοξενεί το μεγαλύτερο τροπικό δάσος στη Γη - τον Αμαζόνιο. Ακόμα καλύτερα είναι να κατεβάσουμε το αρχείο, έτοιμο, από [δω](#).

26.1.1 Project – Οι πυρκαγιές στον Αμαζόνιο

Κάποιες επεξηγήσεις σχετικά με τα δεδομένα:

`Year` είναι η χρονιά που έγινε η δασική πυρκαγιά.

`state` είναι το κράτος της Βραζιλίας.

`month` είναι ο μήνας που έγινε η δασική πυρκαγιά.

`number` είναι ο αριθμός των δασικών πυρκαγιών που αναφέρθηκαν.

`date` είναι η ημερομηνία που αναφέρθηκε η δασική πυρκαγιά

- Βλέποντας το αρχείο csv (`amazon.csv`), διαπιστώνουμε ότι κάποιοι αριθμοί στη στήλη `numbers` είναι δεκαδικοί. Για παράδειγμα, ο αριθμός 2.588 πυρκαγιές δεν έχει νόημα. Αυτό συμβαίνει επειδή με την τελεία μορφοποιούνται οι χιλιάδες. Άρα, 2.588 σημαίνει 2588 δασικές πυρκαγιές. Αυτό το λαμβάνουμε υπόψη κατά την ανάγνωση του αρχείου csv.
- Παρατηρούμε επίσης ότι η στήλη `month` είναι στα πορτογαλικά. Θα δώσουμε μια λύση και για αυτό.
- Όταν εισάγουμε το αρχείο για πρώτη φορά μετά τη λήψη του, λαμβάνουμε ένα ή και περισσότερα σφάλματα: `UnicodeDecodeError: 'utf-8' codec can't decode byte in position : invalid continuation byte`. με μορφή csv και Encoding -> UTF
Για να το διορθώσουμε, ανοίγουμε το csv στο Notepad και το αποθηκεύουμε -8. Ξαναπροσπαθώντας να

εισάγουμε και να διαβάσουμε το αρχείο (με το pandas), παρατηρούμε ότι όλα έχουν διορθωθεί και μπορούμε να παίρνουμε πλέον τις εκτυπώσεις που ζητάμε, χωρίς σφάλματα.

26.1.2 Εισαγωγή στη χρήση του αρχείου

Χρησιμοποιούμε τρεις μεγάλες βιβλιοθήκες τις οποίες και κάνουμε εγκατάσταση, αν δεν υπάρχουν ήδη: `pandas`, `matplotlib` και `googletrans`. Όπως ήδη γνωρίζουμε, μπορούμε να τις εγκαταστήσουμε χρησιμοποιώντας την εντολή `pip install` στο command line μας και τις εισάγουμε στον κώδικά μας.

```
import pandas as pd
import googletrans
import matplotlib.pyplot as plt; plt.rcParamsDefaults()
```

26.1.3 Ανάγνωση του αρχείου

Επιβεβαιώνουμε ότι το αρχείο `amazon.csv` βρίσκεται στον κατάλογο εργασίας μας και ανοίγουμε ένα νέο αρχείο όπου και θα γράφουμε τον κώδικά μας.

```
data = pd.read_csv("amazon.csv", thousands = '.')
```

Η παράμετρος `thousands = "."` αναπληρώνει τη δεκαδική μορφοποίηση.

26.1.4 Δομή του αρχείου μας

Δίνοντας την εντολή `print(data.shape)` παίρνουμε τη δομή του αρχείου μας:

```
6454, 5
```

Δηλαδή, έχουμε ένα αρχείο με 6454 γραμμές και 5 στήλες

26.1.5 Προεπεξεργασία κι ενημέρωση

Δίνοντας τώρα :

```
print(data.head())
```

βλέπουμε τι είδους δεδομένα περιέχονται στο αρχείο και παίρνουμε τις πρώτες γραμμές, καθώς και τις επικεφαλίδες των στηλών:

```
>>> |   year state   month  number      date
    0  1998   Acre  Janeiro        0  1998-01-01
    1  1999   Acre  Janeiro        0  1999-01-01
    2  2000   Acre  Janeiro        0  2000-01-01
    3  2001   Acre  Janeiro        0  2001-01-01
    4  2002   Acre  Janeiro        0  2002-01-01
```

Θα μπορούσαμε να δώσουμε ένα όρισμα στην παραπάνω συνάρτηση για να πάρουμε συγκεκριμένο αριθμό γραμμών. Με κενό, παίρνουμε τις επικεφαλίδες και τις επόμενες 5 γραμμές.

Συνεχίζουμε τη μελέτη του εγγράφου μας, δίνοντας την παρακάτω εντολή:

```
print(data.describe(include= "all"))
```

η οποία μας δίνει μια ωραία περίληψη των δεδομένων. Όπως, το πλήθος όλων των στηλών, την υψηλότερη τιμή που εμφανίζεται σε κάθε στήλη (αν υπάρχει) και τη συχνότητά της.

	year	state	month	number	date
count	6454.000000	6454	6454	6454.000000	6454
unique	NaN	23	12	NaN	20
top	NaN	Rio	Janeiro	NaN	1998-01-01
freq	NaN	717	541	NaN	324
mean	2007.461729	NaN	NaN	522.696312	NaN
std	5.746654	NaN	NaN	1554.846486	NaN
min	1998.000000	NaN	NaN	0.000000	NaN
25%	2002.000000	NaN	NaN	9.000000	NaN
50%	2007.000000	NaN	NaN	54.000000	NaN
75%	2012.000000	NaN	NaN	269.000000	NaN
max	2017.000000	NaN	NaN	25963.000000	NaN

Πριν προχωρήσουμε παρακάτω, καλό είναι να ρίξουμε μια ματιά για να δούμε αν κάποιες τιμές λείπουν:

```
print(data.isna().sum())
```

```
year      0
state     0
month     0
number    0
date      0
dtype: int64
```

Διαπιστώνουμε έτσι, ότι δεν μας λείπει τιμή από καμία στήλη.

26.1.6 Περαιτέρω μελέτη των στοιχείων

Μια σκέψη είναι να οπτικοποιήσουμε τον αριθμό των δασικών πυρκαγιών κατά τη διάρκεια των ετών, και των μηνών ώστε να είμαστε σε θέση να δούμε πιο αναλυτικά αυτή τη μεγάλη και ακαθόριστη προς το παρόν εικόνα που έχουμε σε σχέση με τις πυρκαγιές.

Αυτό θα αποτελέσει και τον τελικό μας στόχο μ' αυτό το project.

26.2.0 Αρχή επεξεργασίας και καθαρισμός δεδομένων

Μπορούμε να διαγράψουμε τις γραμμές του dataset που δεν συμβάλλουν στον αριθμό των δασικών πυρκαγιών. Επομένως, οποιαδήποτε γραμμή με τιμή στη στήλη `numbers` ίση με «0», μπορεί να απορριφθεί. Πρώτα μετατρέπουμε τα 0 σε `NaN` και μετά διαγράφουμε τις γραμμές με `NaN` στη συγκεκριμένη στήλη `numbers`.

Επίσης, εδώ θα χρειαστεί να χρησιμοποιήσουμε και μια άλλη γνωστή και μεγάλη βιβλιοθήκη της Python, την `numpy`, η οποία μπορεί να διαχειριστεί πολύ καλά, πολύπλοκα μαθηματικά προβλήματα και πράξεις.

Αφού λοιπόν εισάγουμε και την παραπάνω βιβλιοθήκη, δίνουμε τις εντολές:

```
data = data.replace(0, np.nan)
data2 = data.dropna(subset=["number"])
```

και ζητάμε και πάλι την εκτύπωση:

```
print(data2.describe(include= "all"))
```

όπου παρατηρούμε πως πλέον το σύνολο των δεδομένων μας έχει πλέον μειωθεί από 6454 σε 5837 γραμμές.

	year	state	month	number	date
count	5837.000000	5837	5837	5837.000000	5837
unique	NaN	23	12	NaN	20
top	NaN	Rio	Outubro	NaN	2016-01-01
freq	NaN	661	534	NaN	317
mean	2007.834847	NaN	NaN	577.947918	NaN
std	5.649076	NaN	NaN	1625.176973	NaN
min	1998.000000	NaN	NaN	1.000000	NaN
25%	2003.000000	NaN	NaN	16.000000	NaN
50%	2008.000000	NaN	NaN	72.000000	NaN
75%	2013.000000	NaN	NaN	334.000000	NaN
max	2017.000000	NaN	NaN	25963.000000	NaN

26.2.1 Δημιουργία υποσυνόλου δεδομένων

Ας δώσουμε μια εντολή ώστε να ομαδοποιήσουμε τα δεδομένα ανά μήνα και να αθροίσουμε τους αριθμούς των πυρκαγιών ανά μήνα:

```
forest_fire_per_month =
data.groupby('month')['number'].sum():
```

και ταυτόχρονα, ζητάμε την εκτύπωση

`print(forest_fire_per_month)`: παρατηρούμε ότι το αποτέλεσμα είναι με αλφαβητική σειρά.

```
month
Abril      28364
Agosto     740841
Dezembro   152596
Fevereiro  30952
Janeiro     52587
Julho      217620
Junho      111405
Maio       46083
Março      35118
Novembro   312326
Outubro    629665
Setembro   1015925
Name: number, dtype: int64
```

Για να το επαναφέρουμε σε σειρά μηνών, χρησιμοποιούμε την `reindex`:

```
forest_fire_per_month =
forest_fire_per_month.reindex(months_unique, axis=0)
```

όπου,

```
months_unique = list(data.month.unique())
```

Είναι οι μοναδικοί μήνες (όπως ήταν αναμενόμενο άλλωστε). Στη συνέχεια μετατρέπουμε τη σειρά σε dataframe:

```
forest_fire_per_month = forest_fire_per_month.to_frame()
```

και ζητάμε την εκτύπωση:

```
forest_fire_per_month.head()
```

Ο κώδικας για την εκτύπωσή μας είναι:

```
import pandas as pd
import googletrans
import matplotlib.pyplot as plt; plt.rcParams()
import numpy as np

data = pd.read_csv("amazon.csv", thousands = '.')
forest_fire_per_month =
data.groupby('month')['number'].sum()
months_unique = list(data.month.unique())
forest_fire_per_month =
forest_fire_per_month.reindex(months_unique, axis=0)
forest_fire_per_month = forest_fire_per_month.to_frame()
print(forest_fire_per_month.head())
```

και η έξοδος:

month	number
Janeiro	52587
Fevereiro	30952
Março	35118
Abril	28364
Maio	46083

26.2.2 Δημιουργία αναγνώσιμων και χρήσιμων δεδομένων

Αν και παίρνουμε μια ικανοποιητική εκτύπωση, κάτι φαίνεται πως δεν είναι σωστό. Η στήλη `month` θεωρείται ως το ευρετήριο του `dataframe`.

Για να ορίσουμε ένα προεπιλεγμένο ευρετήριο, γράφουμε:

```
forest_fire_per_month.reset_index(level=0, inplace=True)
```

και ξαναζητάμε την εκτύπωση

```
print(forest_fire_per_month.head())
```

	month	number
0	Janeiro	52587
1	Fevereiro	30952
2	Março	35118
3	Abril	28364
4	Maio	46083

Δοκιμάζουμε τώρα να εισάγουμε τη βιβλιοθήκη `googletrans` η οποία, όπως φαίνεται και από την ονομασία της, είναι μια βιβλιοθήκη python που υλοποιεί το Google Translate API.

Τη δοκιμάζουμε στη μεταβλητή `months_unique` που

χρησιμοποιήσαμε παραπάνω και μπορούμε επίσης να ζητήσουμε αναλυτική και επεξηγηματική εκτύπωση.

Ο κώδικάς μας είναι:

```
import pandas as pd
from googletrans import Translator
import matplotlib.pyplot as plt; plt.rcParamsDefaults()
import numpy as np

data = pd.read_csv("amazon.csv", thousands = '.')
forest_fire_per_month =
data.groupby('month')['number'].sum()
months_unique = list(data.month.unique())
forest_fire_per_month =
forest_fire_per_month.reindex(months_unique, axis=0)
forest_fire_per_month = forest_fire_per_month.to_frame()
forest_fire_per_month.reset_index(level=0, inplace=True)

translator = Translator() #Δημιουργία αντικειμένου
translator
for month in months_unique:
    detected = translator.detect(month)
    translated = translator.translate(month)
    print(detected)
    print(translated)
    print("...")
```

όμως παίρνουμε τα παρακάτω errors από τον IDLE:

```
= RESTART:
C:/Users/micha/AppData/Local/Programs/Python/Python311/amazon_fires.py
Traceback (most recent call last):
  File
"C:/Users/micha/AppData/Local/Programs/Python/Python311/amazon_fires.py", line 15, in <module>
    detected = translator.detect(month)
```

```
File
"C:\Users\micha\AppData\Local\Programs\Python\Python311\Lib\
site-packages\googletrans\client.py", line 255, in detect
    data = self._translate(text, 'en', 'auto', kwargs)
File
"C:\Users\micha\AppData\Local\Programs\Python\Python311\Lib\
site-packages\googletrans\client.py", line 78, in _translate
    token = self.token_acquirer.do(text)
File
"C:\Users\micha\AppData\Local\Programs\Python\Python311\Lib\
site-packages\googletrans\gtoken.py", line 194, in do
    self._update()
File
"C:\Users\micha\AppData\Local\Programs\Python\Python311\Lib\
site-packages\googletrans\gtoken.py", line 62, in _update
    code = self.RE_TKK.search(r.text).group(1).replace('var
', '')
AttributeError: 'NoneType' object has no attribute 'group'
```

Τα οποία οφείλονται στη βιβλιοθήκη της Google.

26.2.3 Μετάφραση επιλεγμένων δεδομένων

Ας προσπαθήσουμε να τα διορθώσουμε, εγκαθιστώντας την τελευταία, σταθερή της έκδοση:

```
pip install --upgrade googletrans==4.0.0-rc1
```

Χρησιμοποιώντας και πρότερες γνώσεις, μιας και ίσως λάβουμε και πάλι λάθη, μπορούμε να χρησιμοποιήσουμε μια δομή try, except για χειρισμό των πιθανών λαθών:

```
import pandas as pd
from googletrans import Translator
import matplotlib.pyplot as plt
import numpy as np
```



```

data = pd.read_csv("amazon.csv", thousands='.')
forest_fire_per_month =
data.groupby('month')['number'].sum()
months_unique = list(data.month.unique())
forest_fire_per_month =
forest_fire_per_month.reindex(months_unique, axis=0)
forest_fire_per_month = forest_fire_per_month.to_frame()
forest_fire_per_month.reset_index(level=0, inplace=True)

translator = Translator() # Δημιουργία αντικειμένου
for month in months_unique:
    try:
        detected = translator.detect(month)
        translated = translator.translate(month)
        print(detected)
        print(translated)
        print("...")
    except Exception as e:
        print(f"Λάθος μετάφρασης {month}: {str(e)}")

```

και πλέον, μπορούμε να λάβουμε μια εκτύπωση χωρίς λάθη:

```

Translated(src=pt, dest=en, text=March, pronunciation=None, extra_data="('confid
en...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=April, pronunciation=None, extra_data="('confid
en...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=May, pronunciation=None, extra_data="('confiden
...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=June, pronunciation=None, extra_data="('confide
n...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=July, pronunciation=None, extra_data="('confide
n...")
...
Detected(lang=it, confidence=None)
Translated(src=it, dest=en, text=August, pronunciation=None, extra_data="('confi
den...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=September, pronunciation=None, extra_data="('co
nfiden...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=October, pronunciation=None, extra_data="('conf
iden...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=November, pronunciation=None, extra_data="('con
fiden...")
...
Detected(lang=pt, confidence=None)
Translated(src=pt, dest=en, text=December, pronunciation=None, extra_data="('con
fiden...")
...

```

Περιγραφή μεταβλητών του επιστρεφόμενου `Translator()` αντικειμένου:

- `src` — γλώσσα πηγής (προεπιλογή: αυτόματη)
- `dest` — γλώσσα προορισμού (προεπιλογή: en)
- `text` — μεταφρασμένο κείμενο
- `pronunciation` — προφορά

Αυτές οι μεταβλητές μπορούν να προσπελαστούν και μεμονωμένα χρησιμοποιώντας το dot notation, δηλ. την τελεία «.».

Παράδειγμα: Δίνοντας `translated.text` θα παράγουμε το μεταφρασμένο αγγλικό κείμενο.

Δημιουργούμε ένα στιγμιότυπο του `Translator` για να μεταφράσουμε τη στήλη `month` στα `forest_fire_per_month` στα αγγλικά.

```
translator2 = Translator() #Δημιουργία νέου αντικειμένου
Translator.
# Δεν μπορούμε να χρησιμοποιήσουμε το προηγούμενο, καθώς
# περιέχει
# την προηγούμενη υπολογισμένη τιμή των μηνών
for i, m in enumerate(forest_fire_per_month['month']):
    translated = translator2.translate(m)
    month1 = translated.text
    forest_fire_per_month.at[i, 'month'] = month1
```

26.2.4 Συνάρτηση `enumerate()`

Η συνάρτηση `enumerate()` εκχωρεί ένα ευρετήριο σε κάθε στοιχείο σε ένα επαναληπτικό αντικείμενο (iterable) που μπορεί να χρησιμοποιηθεί για αναφορά στο στοιχείο αργότερα.

Η `enumerate()` χρησιμοποιείται συνήθως αντί του βρόχου `for`. Τη χρειαζόμαστε επειδή η `enumerate()` μπορεί να επαναλάβει το ευρετήριο ενός στοιχείου, καθώς και το ίδιο το στοιχείο.

Δίνοντας τώρα την παρακάτω εντολή,

```
print(forest_fire_per_month)
```

παίρνουμε το επιθυμητό αποτέλεσμα:

```
      month  number
0  January    52587
1  February   30952
2   March    35118
3  April     28364
4   May     46083
5   June    111405
6   July    217620
7  August    740841
8  September 1015925
9  October    629665
10 November    312326
11 December    152596
```

26.2.5 Οπτικοποίηση δεδομένων

Διαμορφώνουμε τον κώδικά μας όπως παρακάτω και ζητάμε την εκτύπωση του διαγράμματος (ραβδόγραμμα).

```
import pandas as pd
from googletrans import Translator
import matplotlib.pyplot as plt
import numpy as np

data = pd.read_csv("amazon.csv", thousands='.')
forest_fire_per_month =
data.groupby('month')['number'].sum()
months_unique = list(data.month.unique())
forest_fire_per_month =
forest_fire_per_month.reindex(months_unique, axis=0)
forest_fire_per_month = forest_fire_per_month.to_frame()
forest_fire_per_month.reset_index(level=0, inplace=True)

translator2 = Translator() #Δημιουργία νέου αντικειμένου
Translator.
# Δεν μπορούμε να χρησιμοποιήσουμε το προηγούμενο, καθώς
περιέχει
# την προηγούμενη υπολογισμένη τιμή των μηνών
for i, m in enumerate(forest_fire_per_month['month']):
    translated = translator2.translate(m)
    month1 = translated.text
    forest_fire_per_month.at[i, 'month'] = month1

plt.figure(figsize=(25, 15)) #καθορισμός πλάτους και ύψους
σε ίντσες (!)
#plt.bar(x-values, y-values)
plt.bar(
forest_fire_per_month['month'],
forest_fire_per_month['number'],
color = (0.5,0.1,0.5,0.6)
)
#Χρήση .suptitle για τον τίτλο και .title για τον υπότιτλο
plt.suptitle('Πυρκαγιές στον Αμαζόνιο ανά μήνα',
fontsize=20)
plt.title('Στοιχεία ετών 1998 - 2017', fontsize=20)
```

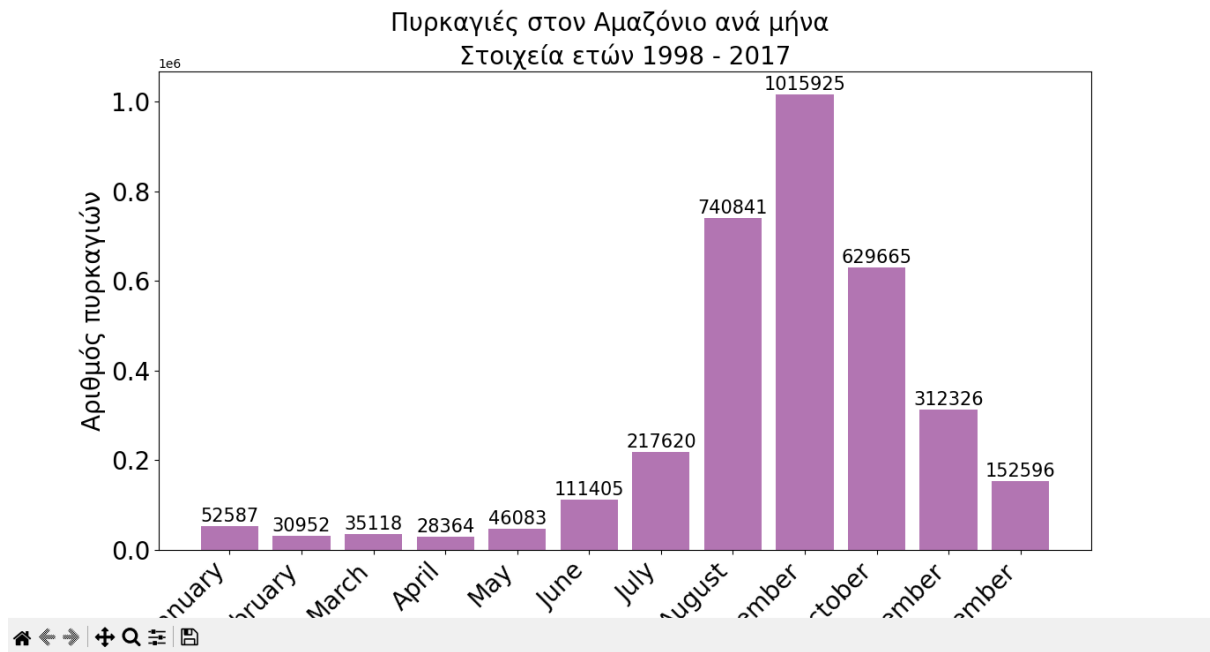
```

plt.xlabel('Μήνας', fontsize=20)
plt.ylabel('Αριθμός πυρκαγιών', fontsize=20)
#plt.text(x-coordinate, y-coordinate, valueOfText,
alignment, fontsize)
#θέτουμε τον κείμενο με την αξία της κάθε μπάρας στην κορυφή
της
for i, num in enumerate(forest_fire_per_month['number']):
    plt.text(
        i,
        num + 10000,
        num,
        ha='center',
        fontsize=15)
# Το plt.setp θέτει μια ιδιότητα σε ένα αντικείμενο.
#Το plt.gca() αποτυπώνει το στιγμιότυπο του κάθε άξονα (gca)
στον συγκεκριμένο αριθμό
#σύμφωνα με τα ορίσματα.
#xticklabels και yticklabels είναι οι τιμές των ετικετών
στους άξονες x και y
# Παρακάτω, θέτουμε το μέγεθος της γραμματοσειράς και τη
στοίχιση των ετικετών
#στους άξονες x και y
plt.setp(plt.gca().get_xticklabels(),
        rotation=45,
        horizontalalignment='right',
        fontsize=20)
plt.setp(plt.gca().get_yticklabels(), fontsize=20)

plt.savefig('forest_fires_plot.png')
plt.show()

```

Η έξοδος:



Εδώ ολοκληρώνουμε αυτό το project και σας παροτρύνουμε να μελετήσετε κάποιες πηγές πόρων που χρησιμοποιήθηκαν για το [googletranslate](#) και την [enumerate\(\)](#), οι οποίες μπορεί να μας φανούν χρήσιμες και σε άλλα projects.

26.3.0 Ασκήσεις

1. Γράψτε ένα πρόγραμμα το οποίο να προσθέτει, αφαιρεί, πολλαπλασιάζει και διαιρεί 2 pandas series (γραμμές)
Χρησιμοποιήστε τις σειρές: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9]

2. Γράψτε ένα πρόγραμμα το οποίο να συγκρίνει τις παρακάτω σειρές: [2, 4, 6, 8, 10], [1, 3, 5, 7, 10]. (ποιοι αριθμοί είναι ίσοι, μεγαλύτεροι ή μικρότεροι; Χρησιμοποιήστε τους τελεστές «==», «>» και «<»).

3. Γράψτε ένα πρόγραμμα το οποίο να τυπώνει τις πρώτες 3 σειρές του παρακάτω DataFrame:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',  
                      'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
             'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
             'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
             'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Μπορείτε να χρησιμοποιήσετε την `iloc` και αυτά που μάθαμε για τα `slices` ώστε να πάρετε μόνο τα 3 στοιχεία.

ΚΑΛΗ ΜΕΛΕΤΗ!
