



Αρχές Γλωσσών και Μεταφραστών

Προαιρετική Εργαστηριακή Άσκηση Python 2021

Αβραμόπουλος Μιχαήλ
ΑΜ:1067451

Έτος: 3ο

Email: up1067451@upnet.gr

Περιεχόμενα

Εισαγωγή.....	3
Βάση Δεδομένων MySQL.....	3
Ανάλυση Κώδικα.....	4
Βιβλιοθήκες.....	4
Κατέβασμα αρχείων.....	4
Διάβασμα και Επεξεργασία Δεδομένων.....	5
Προσπέλαση στη Βάση Δεδομένων.....	8
Ανάκτηση δεδομένων από την Βάση Δεδομένων και δημιουργία .csv αρχείων.....	10
Δημιουργία Γραφημάτων.....	12
Γραφήματα.....	13
Κώδικες.....	15
Κώδικας MySQL.....	15
Κώδικας Python.....	17

Εισαγωγή

Για την συγκεκριμένη εργασία επιλέχτηκε σαν δεύτερη χώρα η Ισπανία. Η 4ετία που χρησιμοποιώ είναι 2008 – 2011. Μαζί με την αναφορά στο αρχείο περιέχονται: ο κώδικας Python, ο κώδικας SQL, το σχήμα της βάσης δεδομένων και τα αρχεία .csv.

Βάση Δεδομένων MySQL

Για την βάση δεδομένων χρησιμοποιήθηκε η MySQL (ο κώδικας θα βρίσκεται στο παράρτημα).

```
python_data nsatae_gr
date : date
# number : int(10)
```

```
python_data nsatae_es
date : date
# number : int(10)
```

```
python_data nsbnratae_gr
date : date
# number : int(10)
```

```
python_data nsbnratae_es
date : date
# number : int(10)
```

```
python_data aatae_gr
date : date
# number : int(10)
```

```
python_data aatae_es
date : date
# number : int(10)
```

```
python_data aonratae_gr
date : date
# number : int(10)
```

```
python_data aonratae_es
date : date
# number : int(10)
```

Η βάση αποτελείται από 4 πίνακες για κάθε αρχείο που περιέχει τα Ελληνικά δεδομένα και αντίστοιχα για την Ισπανία (σύνολο 8). Περιέχει 2 μεταβλητές, μία τύπου **date** για τις ημερομηνίες και μία τύπου **int** των 10 χαρακτήρων για την αποθήκευση των ποσών.

Ανάλυση Κώδικα

Βιβλιοθήκες

```
import requests
import gzip
import mysql.connector
import csv
import matplotlib.pyplot as plt
```

import requests: Για το κατέβασμα των αρχείων

import gzip: Για το extract bytes απο τα gz τύπου αρχεία

import mysql.connector: Για την σύνδεση με τη βάση δεδομένων MySQL

import csv: Για την δημιουργία των csv αρχείων

import matplotlib.pyplot as plt: Για την δημιουργία των γραφημάτων

Κατέβασμα αρχείων

```
url = ['https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tour_occ_nim.tsv.gz',
      'https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tour_occ_ninrmw.tsv.gz',
      'https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tour_occ_arm.tsv.gz',
      'https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tour_occ_arinrmw.tsv.gz']
```

Ο **url** είναι ένας πίνακας που περιέχει σε μορφή Strings τα URLs των αρχείων που θέλουμε να κατεβάσουμε για να επεξεργαστούμε.

```
i = 0
for u in url:
    r = requests.get(u, allow_redirects=True) # κατεβάζω το αρχείο
    filename[i] = u.rsplit('/', 1)[-1] # ονομάζω το αρχείο σύμφωνα με το link
    open(filename[i], 'wb').write(r.content) # αποθηκεύω το αρχείο
    i = i + 1
```

Μετά για κάθε στοιχείο στον πίνακα **url**, κάνουμε **request.get** για να πάρουμε το αρχείο του link. Έπειτα για την ονομασία του αρχείου παίρνουμε ουσιαστικά τους χαρακτήρες μετά το τελευταίο '/', αυτό το κάνουμε κάνοντας **split** το String για κάθε χαρακτήρα '/' και επιλέγοντας το τελευταίο στον πίνακα που δημιουργείται από τον διαχωρισμό ([-1] για να παίρνουμε το τελευταίο στοιχείο ανεξαρτήτως μήκος). Αποθηκεύουμε το όνομα στον πίνακα **filename** και ανοίγουμε την εγγραφή αρχείου (**open(...).write**) με περιεχόμενο τα δεδομένα του κατεβασμένου αρχείου.

Διάβασμα και Επεξεργασία Δεδομένων

```
start_gr = ['NAT,NR,I551,EL ',
            'NR,I551-I553,EU27_2007,EL ',
            'NAT,NR,I551-I553,EL ',
            'NR,I551-I553,EU27_2007,EL ']

end_gr = ['NAT,NR,I551,ES ',
          'NR,I551-I553,EU27_2007,ES ',
          'NAT,NR,I551-I553,ES ',
          'NR,I551-I553,EU27_2007,ES ']

end_es = ['NAT,NR,I551,EU27_2007 ',
          'NR,I551-I553,EU27_2007,EU25 ',
          'NAT,NR,I551-I553,EU27_2007 ',
          'NR,I551-I553,EU27_2007,FI ']

# Οι παραπάνω πίνακες δείχνουν τα σημεία που αρχίζουν και τελειώνουν τα δεδομένα των χωρών
```

Οι παραπάνω πίνακες δείχνουν τα σημεία που ξεκινούν και σταματούν τα δεδομένα που χρειαζόμαστε για τις δύο χώρες (Ελλάδα, Ισπανία) και για τα τέσσερα αρχεία.

```

for a in filename:
    f = gzip.open(a, 'rb') # ανοίγω το gz αρχείο
    file_content = f.read() # διαβάζω τα περιεχόμενα του και τα αποθηκεύω σε μορφή bytes
    f.close()

    file_content_str[i] = file_content.decode() # μετατρέπω τα bytes σε string
    i = i + 1

```

Στη συνέχεια διαβάζω τα δεδομένα των αρχείων σε μια for loop σε μορφή bytes και τα αποθηκεύω προσωρινά στην μεταβλητή **file_content**. Μετά μετατρέπω τα περιεχόμενα από bytes σε strings με την μέθοδο **decode()** και τα αποθηκεύω στο array **file_content_str** που έχει μέγεθος 4 (για τα τέσσερα αρχεία μας)

```

date_final = '9999M99'
i = 0
for b in file_content_str:
    date = b.split("\t", 1)[1]
    date = date.split("\t", 1)[0]
    if date_final > date:
        date_final = date # βρίσκω τη μικρότερη ημερομηνία των 2 μηνών

```

Έπειτα βρίσκω την μικρότερη ημερομηνία που ξεκινούν τα δεδομένα μου, κάποια αρχεία αρχίζουν στις 02/2021 ενώ κάποια άλλα στις 12/2011.

```

data_gr = ['', '', '', '']
data_es = ['', '', '', '']
for b in file_content_str:
    str1 = b.split(start_gr[i], 1)[1] # βρίσκω που ξεκινάνε τα δεδομένα της Ελλάδας
    data_gr = str1.split(end_gr[i], 1)[0] # βρίσκω που τελειώνουν τα δεδομένα της Ελλάδας
    str2 = b.split(end_gr[i], 1)[1] # βρίσκω που ξεκινάνε τα δεδομένα της Ισπανίας
    data_es = str2.split(end_es[i], 1)[0] # βρίσκω που τελειώνουν τα δεδομένα της Ισπανίας
    res_gr = [int(s) for s in data_gr.split() if s.isdigit()] # αποθηκεύω τα δεδομένα σε έναν int πίνακα
    # χρησιμοποιώ list comprehension
    # μετά περνάω τα δεδομένα σε ξεχωριστούς πίνακες για να τα χρησιμοποιήσω παρακάτω
    if i == 0:
        nsatae_gr = res_gr[109:] # κόβω τα δεδομένα για να βρω αντιστοιχία στους μήνες (2011/12 - 2008/01)
    elif i == 1:
        nsbnratae_gr = res_gr
    elif i == 2:
        aatae_gr = res_gr[106:]
    else:
        aonratae_gr = res_gr

    res_es = [int(s) for s in data_es.split() if s.isdigit()]

    if i == 0:
        nsatae_es = res_es[110:]
    elif i == 1:
        nsbnratae_es = res_es
    elif i == 2:
        aatae_es = res_es[110:]
    else:
        aonratae_es = res_es

    i = i + 1

```

Τελευταίο βήμα είναι η επεξεργασία των δεδομένων, αρχικά φτιάχνω 2 πίνακες μεγέθους 4 (έναν για κάθε χώρα με μέγεθος τον αριθμό των αρχείων). Μετά ξεκινώ μια for loop για τα στοιχεία της **file_content_str** όπου κόβω τα strings με τα δεδομένα σε κομμάτια (οι **str1** και **str2** είναι προσωρινές μεταβλητές), αφού περάσω το κατάλληλο string μετά ακολουθεί μια list comprehension for loop με if που ουσιαστικά λέω για κάθε στοιχείο στα δεδομένα μου (δηλαδή όπου δεν υπάρχουν κενά) αν είναι αριθμός (**isdigit()**) τότε αποθήκευσε το στον πίνακα **res_gr**. Τέλος περνάω τα αποτελέσματα στον κατάλληλο πίνακα για να προσπελαστούν στην βάση δεδομένων μου. Αυτό επιτυγχάνεται με την βοηθητική μεταβλητή **i** που μετράει σε ποια επανάληψη είμαι (αν είμαι στην 1η είμαι στο 1ο αρχείο, στη 2η στο 2ο κτλ). Όμως όπως ανέφερα πριν επειδή υπάρχουν κάποιες κενές τιμές για να ξεκινούν σωστά οι πίνακες έπρεπε υπολογίσω που ξεκινάνε και να διαγράψω τα εξτρά στοιχεία (πχ στη περίπτωση που κάποια αρχεία έχουν αρχή το 2021 οπώς το 1ο έχω βάλει **nsatae_gr = res_gr[109:]**).

Προσπέλαση στη Βάση Δεδομένων

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="",  
    database="python_data"  
) # δεδομένα για να συνδεθώ με την βάση MySQL
```

Αυτά είναι τα απαραίτητα στοιχεία για να συνδεθώ στην βάση μου.

```
mycursor = mydb.cursor()  
  
tables = ['nsatae_gr',  
          'nsbnratae_gr',  
          'aatae_gr',  
          'aonratae_gr',  
          'nsatae_es',  
          'nsbnratae_es',  
          'aatae_es',  
          'aonratae_es'] # ονόματα πινάκων στην βάση  
  
j = 0  
  
csv_names = ['Nights spent at tourist accommodation establishments of Greece.csv',  
             'Nights spent by non-residents at tourist accommodation establishments of Greece.csv',  
             'Arrivals at tourist accommodation establishments of Greece.csv',  
             'Arrivals of non-residents at tourist accommodation establishments of Greece.csv',  
             'Nights spent at tourist accommodation establishments of Spain.csv',  
             'Nights spent by non-residents at tourist accommodation establishments of Spain.csv',  
             'Arrivals at tourist accommodation establishments of Spain.csv',  
             'Arrivals of non-residents at tourist accommodation establishments of Spain.csv',  
             ] # ονόματα αρχείων csv που θα δημιουργηθούν
```

Μετά ορίζω τον cursor μου για να προσπελάσω την βάση με εγγραφές. Επίσης δημιουργώ 2 πίνακες, ο πίνακας **tables** περιέχει τα ονόματα των πινάκων της βάσης ενώ ο πίνακας **csv_names** τα ονόματα των αρχείων .csv που θα δημιουργηθούν.


```

for t in tables:
    sql_str = "INSERT IGNORE INTO {} (date, number) VALUES(%s, %s)".format(t) # εντολή εισαγωγής SQL
    # περνάω τα δεδομένα μου σε κάθε κύκλο στον πίνακα res
    if j == 0:
        res = nsatae_gr
    elif j == 1:
        res = nsbnratae_gr
    elif j == 2:
        res = aatae_gr
    elif j == 3:
        res = aonratae_gr
    elif j == 4:
        res = nsatae_es
    elif j == 5:
        res = nsbnratae_es
    elif j == 6:
        res = aatae_es
    else:
        res = aonratae_es
    month_str = date_final.split("M", 1)[1]
    year_str = date_final.split("M", 1)[0]
    month = int(month_str)
    year = int(year_str)
    # κάνω την ημερομηνία απο String τύπου 2011M02 σε ξεχωριστούς ακέραιους

```

Μετά ξεκινάω την προσπάθεια στους 8 πίνακες, οπότε κάνω μια for loop με όσα στοιχεία έχει ο πίνακας **tables**. Για την εντολή εισαγωγής: κάνω **INSERT** στον πίνακα **x**, αγνοώντας αν υπάρχουν ήδη οι ίδιες εγγραφές, 2 τιμές που θα ορίσω αργότερα. Ο πίνακας ορίζεται με την μέθοδο **format()**. Έπειτα χρησιμοποιώντας την βοηθητική μεταβλητή **j** αποθηκεύω τα δεδομένα από τους πίνακες **nsatae_gr**, **nsbnratae_gr** κτλ στην μεταβλητή **res**. Μετά αποθηκεύω την μικρότερη ημερομηνία που βρήκα νωρίτερα σε μορφή **int** (βγάζω το M που είναι ενδιάμεσα και κάνω typecasting).

```

month = int(month_str)
year = int(year_str)
# κάνω την ημερομηνία απο String τύπου 2011M02 σε ξεχωριστούς ακέραιους
for i in range(48):
    if month == 0: # αν ο μετρητής του μήνα μηδενιστεί
        month = 12 # τότε ξεκίνα πάλι από το 12 μήνα
        year = year - 1 # και άλλαξε χρονιά
    date_str = "{}-{}-01".format(year, month) # φτιάχνω το format DATE της MySQL
    val = (date_str, res[i]) # περνάω τα δεδομένα προς ανέβασμα
    mycursor.execute(sql_str, val) # εκτελώ την εντολή εισαγωγής
    mydb.commit() # ανανεώνω την βάση
    month = month - 1

```

Τώρα ξεκινάω μία for loop για τους 48 μήνες (δηλαδή την 4ετία) όπου παράλληλα μετράω αντίστροφα τους μήνες (αφού τα δεδομένα έχουν

αποθηκευτεί με σειρά πιο πρόσφατου) και αν η μεταβλητή **month** μηδενιστεί τότε πάει στην τιμή 12 για να αρχίσει η νέα χρονιά και μειώνω την μεταβλητή **year**. Παράλληλα αποθηκεύω την ημερομηνία στη μορφή που θέλει η SQL (YYYY-MM-DD) και για μέρα βάζω τη 1η κάθε μήνα (είναι αδιάφορο στοιχείο). Μετά περνάω στην μεταβλητή **val** ένα ένα τα στοιχεία του πίνακα **res** και την ημερομηνία. Τέλος κάνω **execute** την εντολή **INSERT** που είχα αναφέρει πιο πάνω και χρησιμοποιώ την μέθοδο **commit()** για να ανανεωθούν οι πίνακες της βάσης.

Ανάκτηση δεδομένων από την Βάση Δεδομένων και δημιουργία .csv αρχείων

```
data_sql = "SELECT YEAR(date), MONTH(date), number FROM {} ORDER BY date".format(t)
# με την παραπάνω εντολή πέρνω τα στοιχεία που με ενδιαφέρουν
mycursor.execute(data_sql)
results = mycursor.fetchall() # τα αποθηκεύω όλα στον πίνακα results
with open(csv_names[j], 'w', newline='', encoding='UTF8') as c_file: # ανοίγουμε το αρχείο
    c = csv.writer(c_file, dialect='excel-tab') # ορίζουμε τον writer
    # dialect='excel-tab' για να ανοίγει και με εξελ
    c.writerow(["Date", "Number"])
    w = 0
    date_res = [0 for j in range(48)]
    num_res = [0 for j in range(48)]
    for x in results:
        date_res[w] = "{}-{}".format(x[0], x[1])
        num_res[w] = x[2]
        c.writerow([date_res[w], num_res[w]])
        w = w + 1
    if j == 0:
        nsatae_gr = num_res
    elif j == 1:
        nsbnratae_gr = num_res
    elif j == 2:
        aatae_gr = num_res
    elif j == 3:
        aonratae_gr = num_res
    elif j == 4:
        nsatae_es = num_res
    elif j == 5:
        nsbnratae_es = num_res
    elif j == 6:
        aatae_es = num_res
    else:
        aonratae_es = num_res
    j = j + 1
```

Αρχικά εκτελώ την εντολή **SELECT** της SQL και παίρνω την χρονιά και τον μήνα από την μεταβλητή **date** της βάσης και τους αριθμούς από τον x πίνακα (χρησιμοποιώ πάλι την μέθοδο **format()**) και τους βάζω σε σειρά αύξουσα με την εντολή **ORDER BY** σύμφωνα με την ημερομηνία.

Αφού κάνω execute την εντολή αποθηκεύω τα δεδομένα που πάρθηκαν στον πίνακα **results** με την μέθοδο **fetchall()**. Έπειτα δημιουργώ και ανοίγω τα αρχεία .csv ένα ένα και με την εντολή **with** όσο έχω ανοιχτό κάποιο αρχείο το επεξεργάζομαι. Ορίζω κάποια βασικά πράγματα (όπως το encoding), ορίζω τον **writer** και γράφω αρχικά στην πρώτη γραμμή Date, Number (2 στήλες δηλαδή). Μετά χρησιμοποιώντας τη βοηθητική μεταβλητή **w** για περάσω τους πίνακες **date_res** και **num_res** τα δεδομένα από τη βάση και να τα περάσω στο αρχείο csv μέσω της μεθόδου **writerow()**. Τέλος ξαναπαιρνάω τα νούμερα στους πίνακες **nsatae_gr**, **nsbnratae_gr** κτλ για να τα χρησιμοποιήσω στα γραφήματα τώρα που είναι στη σειρά που επιθυμώ (2008 → 2011).

Δημιουργία Γραφημάτων

```
f1 = plt.figure(1)
plt.plot(date_res, nsatae_gr, label="Greece")
plt.plot(date_res, nsatae_es, label="Spain")
plt.title("Nights spent at tourist accommodation establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Nights")
plt.legend()

f2 = plt.figure(2)
plt.plot(date_res, nsbnratae_gr, label="Greece")
plt.plot(date_res, nsbnratae_es, label="Spain")
plt.title("Nights spent by non-residents at tourist accommodation establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Nights")
plt.legend()

f3 = plt.figure(3)
plt.plot(date_res, aatae_gr, label="Greece")
plt.plot(date_res, aatae_es, label="Spain")
plt.title("Arrivals at tourist accommodation establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Arrivals")
plt.legend()

f4 = plt.figure(4)
plt.plot(date_res, aonratae_gr, label="Greece")
plt.plot(date_res, aonratae_es, label="Spain")
plt.title("Arrivals of non-residents at tourist accommodation establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Arrivals")
plt.legend()

plt.show()
```

Δημιουργώ 4 γραφήματα, ένα για κάθε αρχείο. Στον άξονα x βάζω τις ημερομηνίες και στον άξονα y βάζω τους πίνακες: **nsatae_gr**, **nsbnratae_gr** κτλ για την εκάστοτε χώρα. Ονομάζω την κάθε γραμμή σύμφωνα με τη χώρα, ορίζω σαν τίτλο το όνομα που βρίσκεται στην ιστοσελίδα της Eurostat, ονομάζω τους άξονες και δημιουργώ και το υπόμνημα με την μέθοδο **legend()**. Τέλος εμφανίζω τα γραφήματα με την μέθοδο **show()**.

Γραφήματα

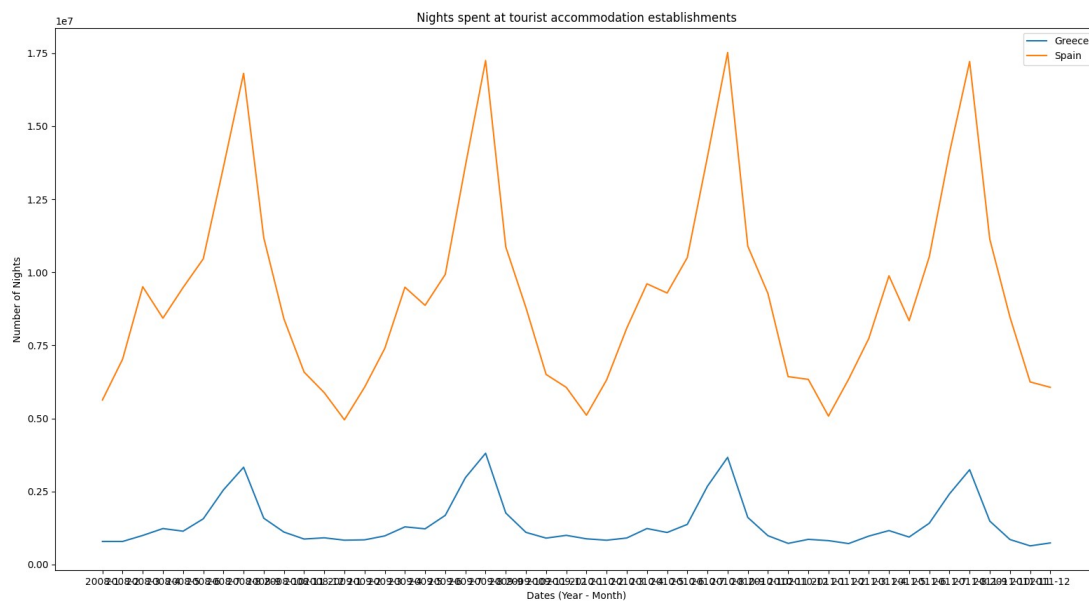


Figure 1: Nights spent at tourist accommodation establishments

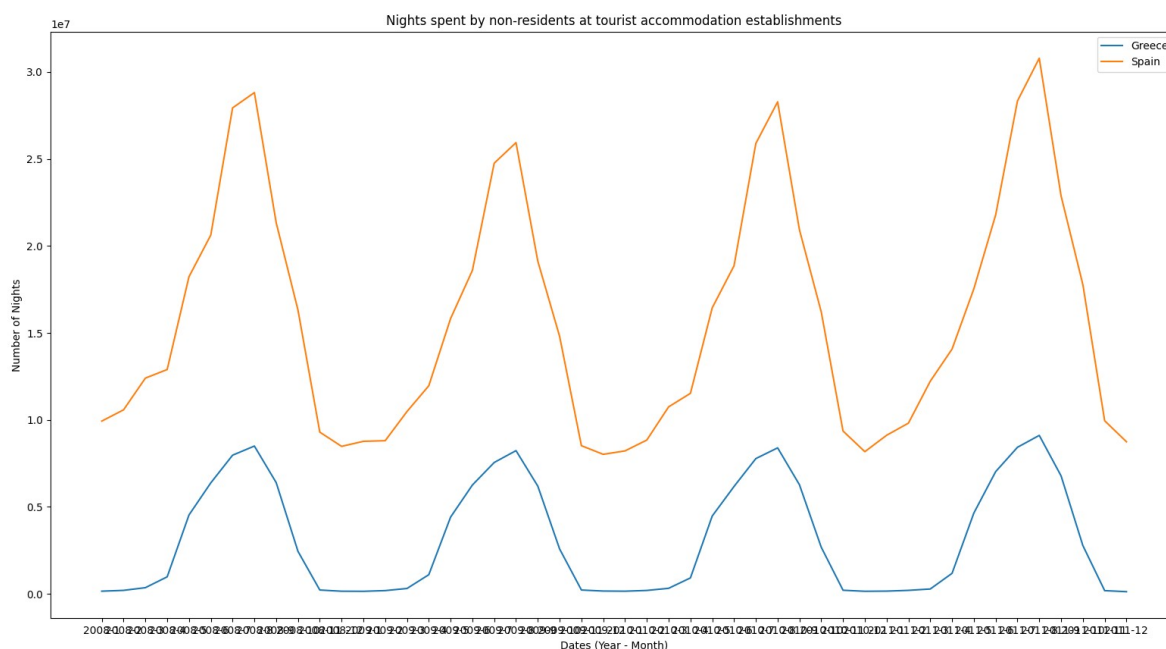


Figure 2: Nights spent by non-residents at tourist accommodation establishments

Αβραμόπουλος Μιχαήλ
1067451

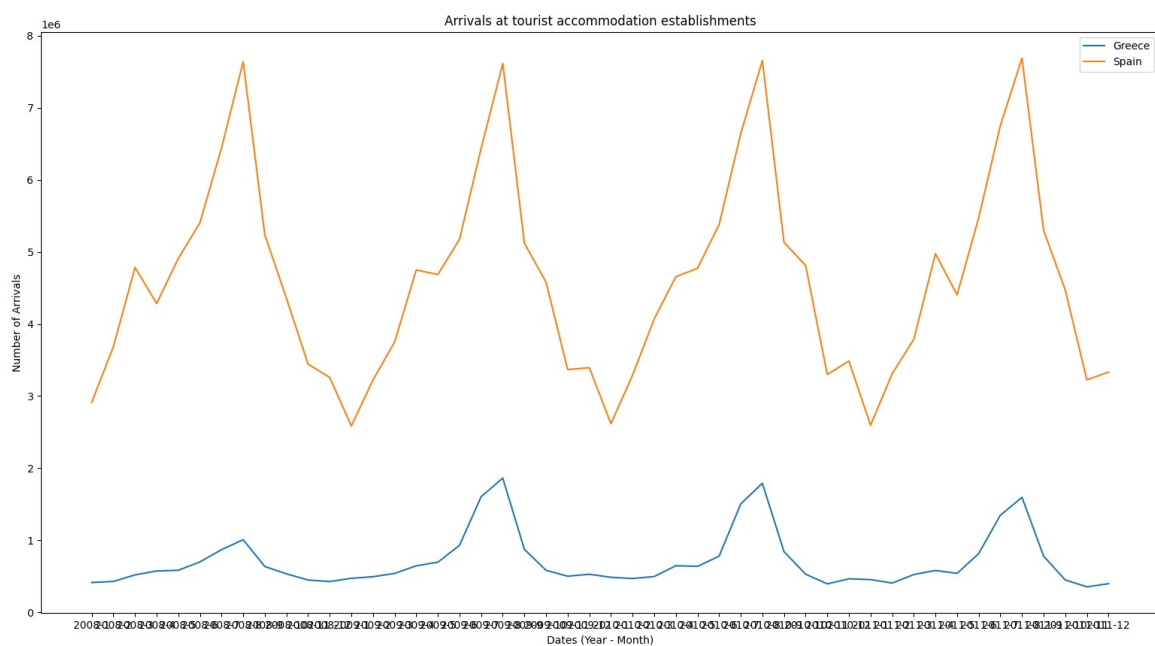


Figure 3: Arrivals at tourist accommodation establishments

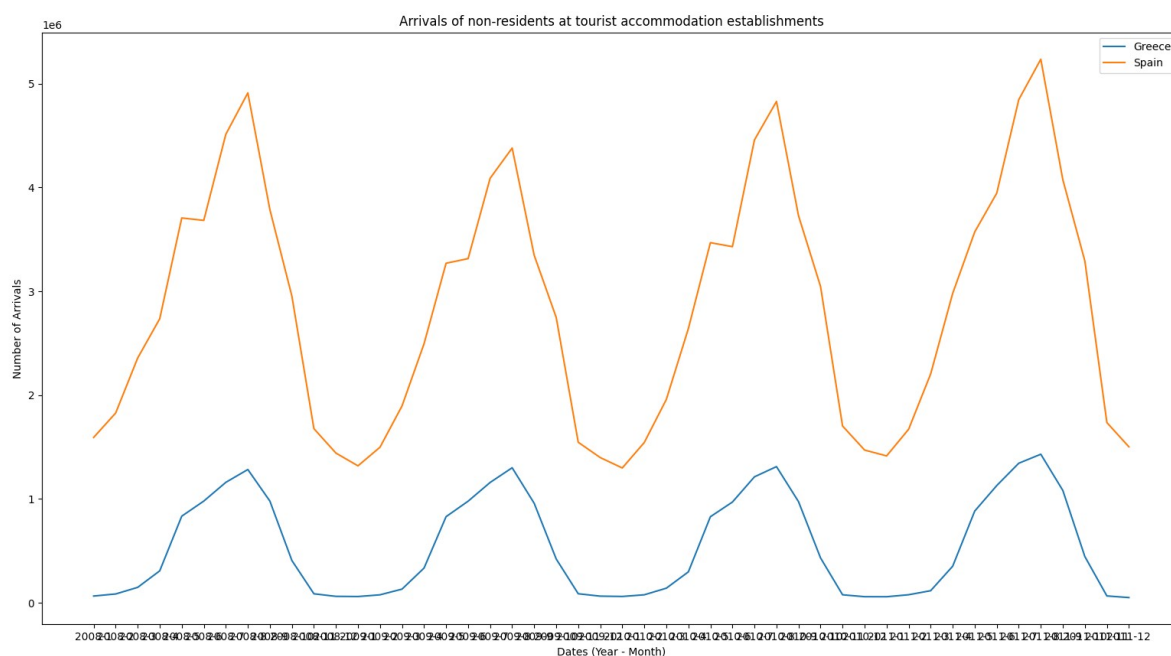


Figure 4: Arrivals of non-residents at tourist accommodation establishments

Κώδικες

Κώδικας MySQL

```
DROP DATABASE IF EXISTS python_data;
```

```
CREATE DATABASE python_data;
```

```
USE python_data;
```

```
CREATE TABLE nsatae_gr(  
date DATE,  
number INT(10),  
PRIMARY KEY(date)  
); -- table for Nights spent at tourist accommodation  
establishments of Greece
```

```
CREATE TABLE nsbnratae_gr(  
date DATE,  
number INT(10),  
PRIMARY KEY(date)  
); -- table for Nights spent by non-residents at tourist  
accommodation establishments of Greece
```

```
CREATE TABLE aatae_gr(  
date DATE,  
number INT(10),
```

```
PRIMARY KEY(date)
); -- table for Arrivals at tourist accommodation
establishments of Greece

CREATE TABLE aonratae_gr(
date DATE,
number INT(10),
PRIMARY KEY(date)
); -- table for Arrivals of non-residents at tourist
accommodation establishments of Greece

CREATE TABLE nsatae_es(
date DATE,
number INT(10),
PRIMARY KEY(date)
); -- table for Nights spent at tourist accommodation
establishments of Spain

CREATE TABLE nsbnratae_es(
date DATE,
number INT(10),
PRIMARY KEY(date)
```



```
); -- table for Nights spent by non-residents at tourist  
accommodation establishments of Spain
```

```
CREATE TABLE aatae_es(  
date DATE,  
number INT(10),  
PRIMARY KEY(date)
```

```
); -- table for Arrivals at tourist accommodation  
establishments of Spain
```

```
CREATE TABLE aonratae_es(  
date DATE,  
number INT(10),  
PRIMARY KEY(date)
```

```
); -- table for Arrivals of non-residents at tourist  
accommodation establishments of Spain
```

Κώδικας Python

```
import requests  
import gzip  
import mysql.connector  
import csv  
import matplotlib.pyplot as plt
```

```

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="python_data"
) # δεδομένα για να συνδεθώ με την βάση MySQL

url = ['https://ec.europa.eu/eurostat/estat-navtree-
portlet-prod/BulkDownloadListing?file=data/
tour_occ_nim.tsv.gz',
       'https://ec.europa.eu/eurostat/estat-navtree-
portlet-prod/BulkDownloadListing?file=data/
tour_occ_ninrmw.tsv.gz',
       'https://ec.europa.eu/eurostat/estat-navtree-
portlet-prod/BulkDownloadListing?file=data/
tour_occ_arm.tsv.gz',
       'https://ec.europa.eu/eurostat/estat-navtree-
portlet-prod/BulkDownloadListing?file=data/
tour_occ_arnrmw.tsv.gz']

# URLs των αρχείων

# 1ο αρχείο: Ελλάδα(NAT,NR,I551,EL)
Ισπανία(NAT,NR,I551,ES) 2021M02

# 2ο αρχείο: Ελλάδα(NR,I551-I553,EU27_2007,EL)
Ισπανία(NR,I551-I553,EU27_2007,ES) 2011M12

```

```

# 3ο αρχείο: Ελλάδα(NAT,NR,I551-I553,EL)
Ισπανία(NAT,NR,I551-I553,ES) 2021M02

# 4ο αρχείο: Ελλάδα(NR,I551-I553,EU27_2007,EL)
Ισπανία(NR,I551-I553,EU27_2007,ES) 2011M12

start_gr = ['NAT,NR,I551,EL',
            'NR,I551-I553,EU27_2007,EL',
            'NAT,NR,I551-I553,EL',
            'NR,I551-I553,EU27_2007,EL']

end_gr = ['NAT,NR,I551,ES',
          'NR,I551-I553,EU27_2007,ES',
          'NAT,NR,I551-I553,ES',
          'NR,I551-I553,EU27_2007,ES']

end_es = ['NAT,NR,I551,EU27_2007',
          'NR,I551-I553,EU27_2007,EU25',
          'NAT,NR,I551-I553,EU27_2007',
          'NR,I551-I553,EU27_2007,FI']

# Οι παραπάνω πίνακες δείχνουν τα σημεία που αρχίζουν
και τελειώνουν τα δεδομένα των χωρών

filename = ['', '', '', '']

i = 0

for u in url:
    r = requests.get(u, allow_redirects=True) #
κατεβάζω το αρχείο

```

```

    filename[i] = u.rspllit('/', 1)[-1] # ονομάζω το
αρχείο σύμφωνα με το link
    open(filename[i], 'wb').write(r.content) #
αποθηκεύω το αρχείο
    i = i + 1
i = 0
file_content_str = ['', '', '', '']
for a in filename:
    f = gzip.open(a, 'rb') # ανοίγω το gz αρχείο
    file_content = f.read() # διαβάζω τα περιεχόμενα
του και τα αποθηκεύω σε μορφή bytes
    f.close()

    file_content_str[i] = file_content.decode() #
μετατρέπω τα bytes σε string
    i = i + 1
date_final = '9999M99'
i = 0
for b in file_content_str:
    date = b.split("\t", 1)[1]
    date = date.split("\t", 1)[0]
    if date_final > date:
        date_final = date # βρίσκω τη μικρότερη
ημερομηνία των 2 μηνών

```

```

i = 0
data_gr = ['', '', '', '']
data_es = ['', '', '', '']
for b in file_content_str:
    str1 = b.split(start_gr[i], 1)[1] # βρίσκω που
    ξεκινάνε τα δεδομένα της Ελλάδας
    data_gr = str1.split(end_gr[i], 1)[0] # βρίσκω που
    τελειώνουν τα δεδομένα της Ελλάδας
    str2 = b.split(end_gr[i], 1)[1] # βρίσκω που
    ξεκινάνε τα δεδομένα της Ισπανίας
    data_es = str2.split(end_es[i], 1)[0] # βρίσκω που
    τελειώνουν τα δεδομένα της Ισπανίας
    res_gr = [int(s) for s in data_gr.split() if
    s.isdigit()] # αποθηκεύω τα δεδομένα σε έναν int πίνακα
    # χρησιμοποιώ list comprehension
    # μετά περνάω τα δεδομένα σε ξεχωριστούς πίνακες για
    να τα χρησιμοποιήσω παρακάτω
    if i == 0:
        nsatae_gr = res_gr[109:] # κόβω τα δεδομένα για
        να βρω αντιστοιχία στους μήνες (2011/12 - 2008/01)
    elif i == 1:
        nsbnratae_gr = res_gr
    elif i == 2:
        aatae_gr = res_gr[106:]
    else:

```

```

        aonratae_gr = res_gr

    res_es = [int(s) for s in data_es.split() if
s.isdigit()]

    if i == 0:
        nsatae_es = res_es[110:]
    elif i == 1:
        nsbnratae_es = res_es
    elif i == 2:
        aatae_es = res_es[110:]
    else:
        aonratae_es = res_es

    i = i + 1

mycursor = mydb.cursor()

tables = ['nsatae_gr',
        'nsbnratae_gr',
        'aatae_gr',
        'aonratae_gr',
        'nsatae_es',
        'nsbnratae_es',

```

```

        'aatae_es',
        'aonratae_es']] # ονόματα πινάκων στην βάση
j = 0

csv_names = ['Nights spent at tourist accommodation
establishments of Greece.csv',
             'Nights spent by non-residents at tourist
accommodation establishments of Greece.csv',
             'Arrivals at tourist accommodation
establishments of Greece.csv',
             'Arrivals of non-residents at tourist
accommodation establishments of Greece.csv',
             'Nights spent at tourist accommodation
establishments of Spain.csv',
             'Nights spent by non-residents at tourist
accommodation establishments of Spain.csv',
             'Arrivals at tourist accommodation
establishments of Spain.csv',
             'Arrivals of non-residents at tourist
accommodation establishments of Spain.csv',
             ] # ονόματα αρχείων csv που θα
δημιουργηθούν

for t in tables:
    sql_str = "INSERT IGNORE INTO {} (date, number)
VALUES(%s, %s)".format(t) # εντολή εισαγωγής SQL

```

```

# περνάω τα δεδομένα μου σε κάθε κύκλο στον πίνακα
res

if j == 0:
    res = nsatae_gr
elif j == 1:
    res = nsbnratae_gr
elif j == 2:
    res = aatae_gr
elif j == 3:
    res = aonratae_gr
elif j == 4:
    res = nsatae_es
elif j == 5:
    res = nsbnratae_es
elif j == 6:
    res = aatae_es
else:
    res = aonratae_es

month_str = date_final.split("M", 1)[1]
year_str = date_final.split("M", 1)[0]
month = int(month_str)
year = int(year_str)

# κάνω την ημερομηνία απο String τύπου 2011M02 σε
ξεχωριστούς ακέραιους

```



```

for i in range(48):
    if month == 0: # αν ο μετρητής του μήνα
μηδενιστεί
        month = 12 # τότε ξεκίνα πάλι από το 12
μήνα
        year = year - 1 # και άλλαξε χρονιά
        date_str = "{}-{}-01".format(year, month) #
φτιάχνω το format DATE της MySQL
        val = (date_str, res[i]) # περνάω τα δεδομένα
προς ανέβασμα
        mycursor.execute(sql_str, val) # εκτελώ την
εντολή εισαγωγής
        mydb.commit() # ανανεώνω την βάση
        month = month - 1

data_sql = "SELECT YEAR(date), MONTH(date), number
FROM {} ORDER BY date".format(t)

# με την παραπάνω εντολή πέρνω τα στοιχεία που με
ενδιαφέρουν
mycursor.execute(data_sql)
results = mycursor.fetchall() # τα αποθηκεύω όλα
στον πίνακα results

with open(csv_names[j], 'w', newline='',
encoding='UTF8') as c_file: # ανοίγουμε το αρχείο
    c = csv.writer(c_file, dialect='excel-tab') #
ορίζουμε τον writer

```

```
# dialect='excel-tab' για να ανοίγει και με
εξελ

c.writerow(["Date", "Number"])

w = 0

date_res = [0 for j in range(48)]
num_res = [0 for j in range(48)]

for x in results:

    date_res[w] = "{}-{}".format(x[0], x[1])
    num_res[w] = x[2]

    c.writerow([date_res[w], num_res[w]])

    w = w + 1

if j == 0:
    nsatae_gr = num_res
elif j == 1:
    nsbnratae_gr = num_res
elif j == 2:
    aatae_gr = num_res
elif j == 3:
    aonratae_gr = num_res
elif j == 4:
    nsatae_es = num_res
elif j == 5:
    nsbnratae_es = num_res
```

```
elif j == 6:  
    aatae_es = num_res  
else:  
    aonratae_es = num_res  
j = j + 1
```

```
f1 = plt.figure(1)  
plt.plot(date_res, nsatae_gr, label="Greece")  
plt.plot(date_res, nsatae_es, label="Spain")  
plt.title("Nights spent at tourist accommodation  
establishments")  
plt.xlabel("Dates (Year - Month)")  
plt.ylabel("Number of Nights")  
plt.legend()  
  
f2 = plt.figure(2)  
plt.plot(date_res, nsbnratae_gr, label="Greece")  
plt.plot(date_res, nsbnratae_es, label="Spain")  
plt.title("Nights spent by non-residents at tourist  
accommodation establishments")  
plt.xlabel("Dates (Year - Month)")  
plt.ylabel("Number of Nights")  
plt.legend()
```

```
f3 = plt.figure(3)
plt.plot(date_res, aatae_gr, label="Greece")
plt.plot(date_res, aatae_es, label="Spain")
plt.title("Arrivals at tourist accommodation
establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Arrivals")
plt.legend()

f4 = plt.figure(4)
plt.plot(date_res, aonratae_gr, label="Greece")
plt.plot(date_res, aonratae_es, label="Spain")
plt.title("Arrivals of non-residents at tourist
accommodation establishments")
plt.xlabel("Dates (Year - Month)")
plt.ylabel("Number of Arrivals")
plt.legend()

plt.show()
```