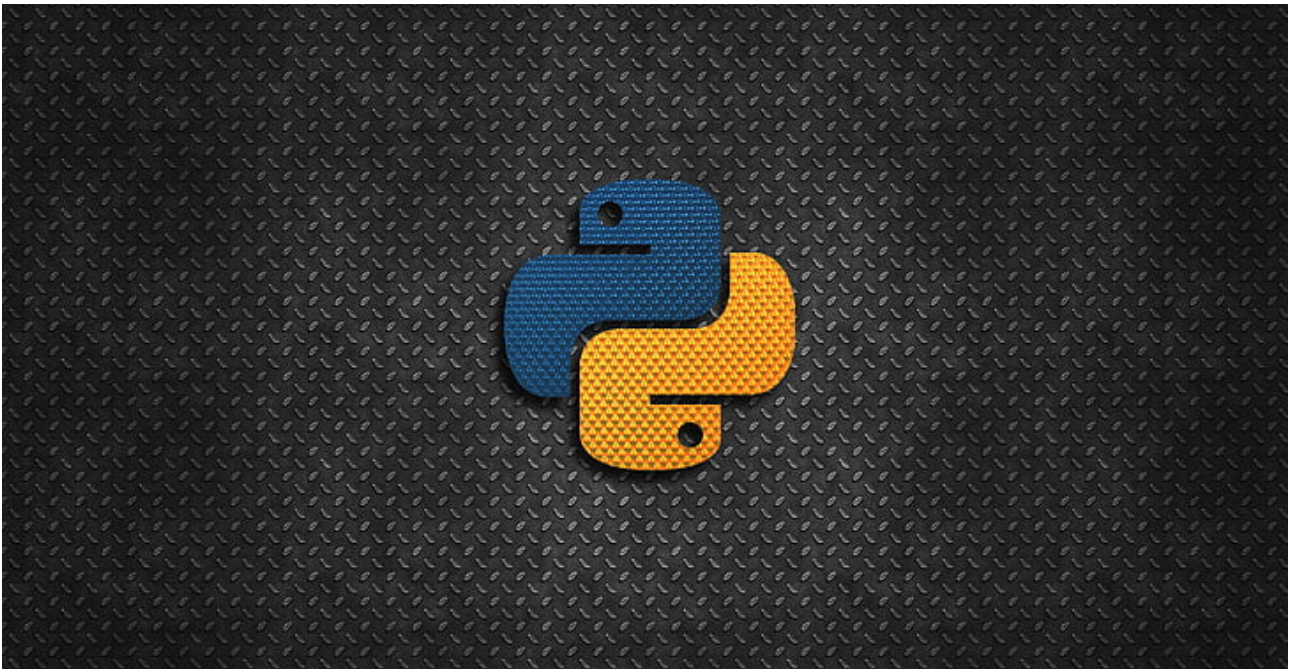


**Πανεπιστήμιο Πατρών**  
**Μηχανικών Η/Υ και Πληροφορικής**

**Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών**

Προαιρετική Εργαστηριακή Άσκηση  
Εαρινό Εξάμηνο 2021



Τσαπάροβ Καλογιάν  
ΑΜ (παλιό): 235909  
ΑΜ: 1041620  
Έτος: 8ο

## Εισαγωγή

Η γλώσσα προγραμματισμού Python, μαζί με τα εξωτερικά πακέτα και βιβλιοθήκες λογισμικού, αποτελούν ένα ισχυρό εργαλείο ανοικτού κώδικα. Έχει μεγάλη χρησιμότητα όχι μόνο στην αυστηρά οριοθετημένη περιοχή της επιστήμης των υπολογιστών αλλά και στην έρευνα κάθε ειδικότητας αναφορικά με την ανάλυση και την επεξεργασία των δεδομένων.

Στην εργαστηριακή άσκηση θα χρησιμοποιηθούν δεδομένα σχετικά με τον τουρισμό στην **Ελλάδα** και την **Βουλγαρία** για την τετραετία **2003-2006** από την ιστοσελίδα της Eurostat.

## Python Script

Θεώρησα ότι να επιλέξω τις χώρες και τις ημερομηνίες για τις οποίες με ενδιαφέρουν τα δεδομένα στο site της eurostat και μετά να πάρω το link για το αρχείο βρίσκοντάς το κάνοντας inspect τον κώδικά της σελίδας, για τις κατηγορίες:

- Nights spent at tourist accommodation establishments (nsa)  
<https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/cf0502da-22f7-4f02-b97b-ecbbae007595>
- Nights spent by non-residents at tourist accommodation establishments (nsb)  
<https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/7e8af94e-dcbb-49af-9714-396458d35a41>
- Arrivals at tourist accommodation establishments (aat)  
<https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/19c12315-0b09-4092-b407-ed0aebb990cb>
- Arrivals of non-residents at tourist accommodation establishments (aon)  
<https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/b8c9ea31-63e0-4474-b2fc-f2306b66f3f4>

Επέλεξα να κατεβάσω τα στοιχεία σε μορφή αρχείου .xlsx για να είναι πιο εύκολη η κατανόηση των περιεχομένων του. Η εργασία δεν αναφέρει σε τι μορφή θα έπρεπε να τα κατεβάζει το script.

```
Cancel</button>
▼<a ng-href="https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/cf0502da-22f7-4f02-b97b-ecbbae007595" class="btn btn-primary pull-right" ng-click="close()" data-qe-id="data-ready-download-start-button" download>
  <span translate>Download</span>
</a> == $0
::after
```

*Παράδειγμα το inspect του κώδικα για το πρώτο link*

## Python script code

```
import requests as req

#NSA 2003-2006 // GR & BG
with req.get('https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/cf0502da-22f7-4f02-b97b-ecbbae007595')
as rq:
    with open('nsa.xlsx','wb') as file:
        file.write(rq.content)

#NSB 2003-2006 // GR & BG
with req.get('https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/7e8af94e-dcbb-49af-9714-396458d35a41')
as rq:
    with open('nsb.xlsx','wb') as file:
        file.write(rq.content)

#AAT2003-2006 // GR & BG
with req.get('https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/19c12315-0b09-4092-b407-ed0aebb990cb')
as rq:
    with open('aat.xlsx','wb') as file:
        file.write(rq.content)

#AON 2003-2006 // GR & BG
with req.get('https://ec.europa.eu/eurostat/databrowser-backend/api/query/1.0/LIVE/xlsx/en/download/b8c9ea31-63e0-4474-b2fc-f2306b66f3f4')
as rq:
    with open('aon.xlsx','wb') as file:
        file.write(rq.content)
```

Αφου έτρεξα το script.py, έλεγξα αν τα αρχεία που κατέβασε ήταν σωστά.

## Python program

Αρχικά επειδή τα αρχεία που το script κατέβασε είναι τύπου .xlsx, η βιβλιοθήκη xlrd δεν ήταν κατάλληλη καθώς πλέον διαχειρίζεται μόνο αρχεία xls. Για αυτό προτίμησα την βιβλιοθήκη openpyxl.

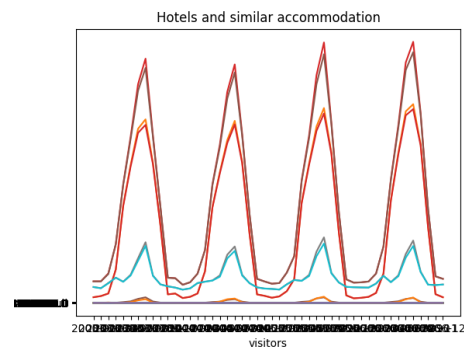
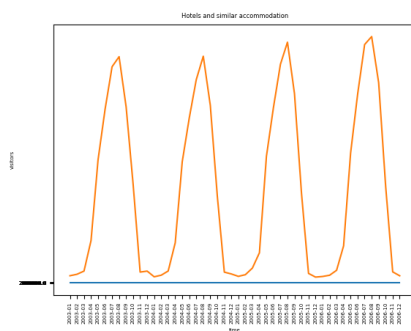
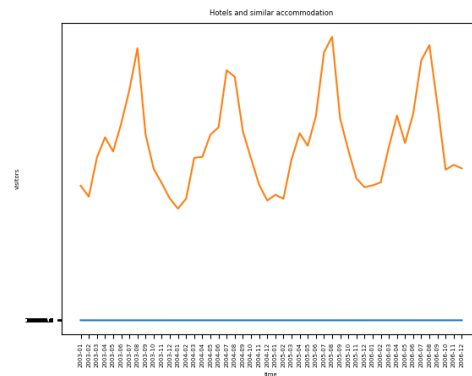
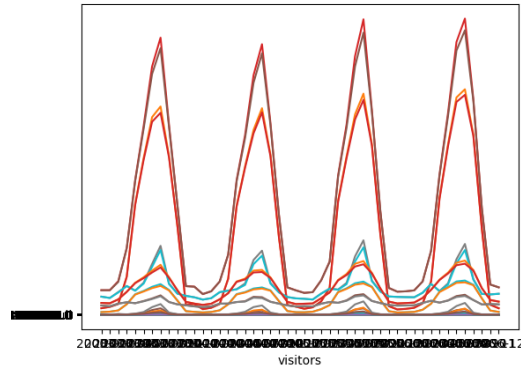
Παρατηρώντας τα αρχεία είδα ότι έχουν αρκετά παρομοιότυπη δομή, εάν όχι ίδια. Κάθε αρχείο έχει στην αρχή 2 sheets με γενικές πληροφορίες που δεν μας ενδιαφέρουν. Όσα sheet μας ενδιαφέρουν έχουν όνομα “Sheet x”. Παρατηρώντας το πρώτο αρχείο (το έχω συμβολίσει nsa) παρατηρώ ότι δεν χρειαζόμαστε όλες τις καρτέλες αλλά συγκεκριμένες μόνο, καθώς σε άλλες δείχνει τα δεδομένα με νούμερα και σε άλλες με ποσοστά.

Οπότε, σε κάθε sheet που με ενδιαφέρει, βλέπουμε το όνομα του πίνακα στο C8, και τα δεδομένα

που χρειαζόμαστε είναι στις γραμμές 10,12,13. Με τον κώδικά μου φτιάχνω έναν πίνακα που έχει σε μια γραμμή το time, το Bg και το GR. Στην συνέχεια τα περνάω στην βάση δεδομένων αλλά και τα εξάγω σαν .csv, τον κάθε ένα πίνακα που δημιουργώ για κάθε sheet.

Ύστερα από διάφορες αποτυχημένες προσπάθειες, όσο ανακάλυπτα τις δυνατότητες της matplotlib, κατάλαβα ότι λόγω της διαφοράς των τιμών μεταξύ των Ελλάδα, Βουλγαρία, τα διαγράμματα θα είναι πιο αποτελεσματικά στο να μεταβιβάσουν την πληροφορία οπτικά αν γίνουν ξεχωριστά για κάθε μία χώρα.

her short-stay accommodation; camping grounds, recreational vehicle parks



Μερικές αποτυχημένες προσπάθειες χρήσης της matplotlib

Προσπάθησα να βρώ τρόπο να ξεφορτωθώ σε κάποια διαγράμματα το 1E6 που εμφανίζεται πάνω αριστερά αλλά δεν τα κατάφερα οπότε εν τέλει αφαίρεσα σχετικά πραγματάκια (import ScalarFormatter).

Ένα άλλο σημείο που με έκανα να κολλήσω ήταν στην εισαγωγή των δεδομένων στην βάση δεδομένων. Συγκεκριμένα στην αρχή χρησιμοποίησα το:

```
selectquery="INSERT INTO "+name+" (TIME, BG, GR) VALUES ('+time+', '+bg+', '+gr+');"

```

όμως η μεταβλητή time είχε τιμή πχ. 2003-12 με αποτέλεσμα στην βάση να εισάγεται 1991. Όταν το συνειδητοποίησα, δηλαδή αφού πρώτα τσέκαρα ότι όντως τα αρχεία δεν είχαν τέτοιες τιμές, απλά έβαλα το time μέσα σε εισαγωγικά.

```
time=" '"+time+"'"

```

Ένα άλλο πράγμα που με δυσκόλεψε ήταν ότι δεν είχα ελέξει τι τιμές μπορεί να συναντήσω στα αρχεία nsb και aon, με αποτέλεσμα η στήλη gr να παίρνει την τιμή : που προκαλούσε θέμα στο sql query. Για αυτό και υλοποίησα το πρόγραμμα σε δύο φάσεις, πρώτα τα nsb,aat και μετά τα nsb,aon.

## Κώδικας Προγράμματος

```
from openpyxl import Workbook, load_workbook
#import pandas as pd
from openpyxl.utils import get_column_letter
import numpy as np
from matplotlib import pyplot as plt
import mysql.connector
import csv
from matplotlib.ticker import ScalarFormatter

print("starts here")
# P GIA TA ARXEIA NSA K AAT
p=["nsa","aat"]
for a in p:
    path=str(a)+".xlsx"
    wb=load_workbook(path)
    conn = mysql.connector.connect(host="localhost", port="3306",user="root",password="root",database="test")
    cursor = conn.cursor()
    print("i will ignore the warning yolo \n")

    if a=="nsa":
        sheet=[1,2,3,7,8,9,13,14,15]

    if a=="aat":
        sheet=[1,2,3,7,8,9,13,14,15]

    sheet=[1,5,9]
    for s in sheet:

        ws = wb.get_sheet_by_name("Sheet "+str(s))
        print(ws)
        table_name=ws['C8'].value
        print(table_name)
        name=str(a)+"pinakas"+str(s)
        print('-----')
        selectquery="CREATE TABLE "+name+" (Counter int AUTO_INCREMENT,TIME varchar(255)
),BG varchar(255) NULL,GR varchar(255) NULL, PRIMARY KEY (Counter));"
        print(selectquery)
        cursor.execute(selectquery)
        table = np.array(['TIME','BG','GR'])
        timecol=[]
        bgcol=[]
        grcol=[]
        for col in range (2,97,2):
            char=get_column_letter(col)
```

```

time=ws[char+'10'].value
bg=ws[char+'12'].value

if bg==":" :
    bg="0"

gr = ws[char+'13'].value

timecol.append(time)
bgcol.append(bg)
grcol.append(gr)

new_row = [ time,bg ,gr ]
table = np.vstack([table,new_row])
time=""+time+" "
bg=str(bg)
gr=str(gr)

selectquery="INSERT INTO "+name+" (TIME, BG, GR) VALUES (" +time+", "+bg+", "+gr
+");"
cursor.execute(selectquery)
print(table)

plt.plot(timecol,grcol)
plt.title(table_name, fontsize=6)
plt.xlabel("time",fontsize=5)
plt.ylabel("visitors",fontsize=5,rotation=90)
plt.xticks(fontsize=5, rotation=90)
plt.yticks(fontsize=5)

file=name+"a.png"
plt.savefig(file)
plt.clf()

plt.plot(timecol,bgcol)
plt.title(table_name, fontsize=6)
plt.xlabel("time",fontsize=5)
plt.ylabel("visitors",fontsize=5,rotation=90)
plt.xticks(fontsize=5, rotation=90)
plt.yticks(fontsize=5)

file=name+"b.png"
plt.savefig(file)
plt.clf()

with open(name,'w') as f:
    write = csv.writer(f)

```

```

write.writerows(table)

#NSB KAI AON
p=["nsb","aon"]
for a in p:
    path=str(a)+".xlsx"
    wb=load_workbook(path)

    print("i will ignore the warning yolo \n")

    if a=="nsb":
        sheet=[1]

    if a=="aon":
        sheet=[1,5,9]

    for s in sheet:

        ws = wb.get_sheet_by_name("Sheet "+str(s))
        print(ws)
        table_name=ws['C7'].value
        print(table_name)
        name=str(a)+"pinakas"+str(s)
        print('-----')
        selectquery="CREATE TABLE "+name+" (Counter int AUTO_INCREMENT,TIME varchar(255
),BG varchar(255) NULL,GR varchar(255) NULL, PRIMARY KEY (Counter));"
        print(selectquery)
        cursor.execute(selectquery)
        table = np.array(['TIME','BG','GR'])
        timecol=[]
        bgcol=[]
        grcol=[]
        for col in range (2,97,2):
            char=get_column_letter(col)

            time=ws[char+'10'].value
            bg=ws[char+'12'].value

            if bg=="":
                bg="0"

            gr = ws[char+'13'].value
            if gr=="":
                gr="0"
            timecol.append(time)
            bgcol.append(bg)
            grcol.append(gr)

```

```

new_row = [ time,bg ,gr ]
table = np.vstack([table,new_row])
time=""+time+" "
bg=str(bg)
gr=str(gr)

selectquery="""INSERT INTO """+name+""" (TIME, BG, GR) VALUES (""+time+""","
""+bg+""","""+gr+""");"""
cursor.execute(selectquery)
print(table)

plt.plot(timecol,grcol)
plt.title(table_name, fontsize=6)
plt.xlabel("time",fontsize=5)
plt.ylabel("visitors",fontsize=5,rotation=90)
plt.xticks(fontsize=5, rotation=90)
plt.yticks(fontsize=5)

file=name+"a.png"
plt.savefig(file)
plt.clf()

plt.plot(timecol,bgcol)
plt.title(table_name, fontsize=6)
plt.xlabel("time",fontsize=5)
plt.ylabel("visitors",fontsize=5,rotation=90)
plt.xticks(fontsize=5, rotation=90)
plt.yticks(fontsize=5)

file=name+"b.png"
plt.savefig(file)
plt.clf()

with open(name,'w') as f:
    write = csv.writer(f)
    write.writerows(table)

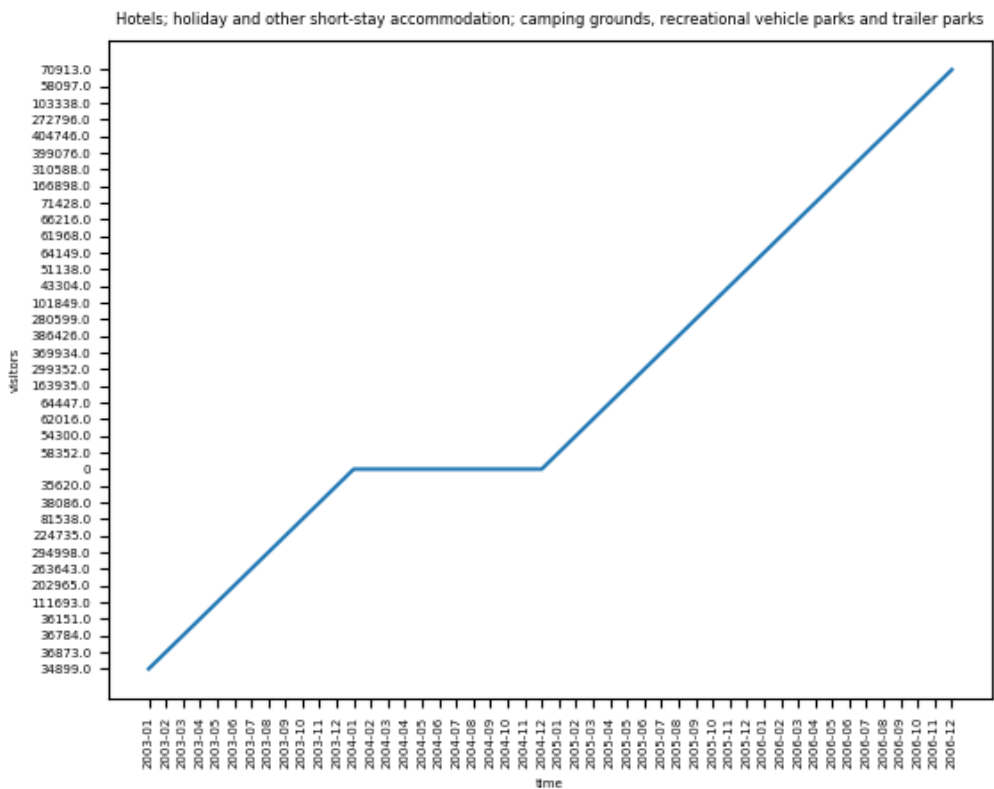
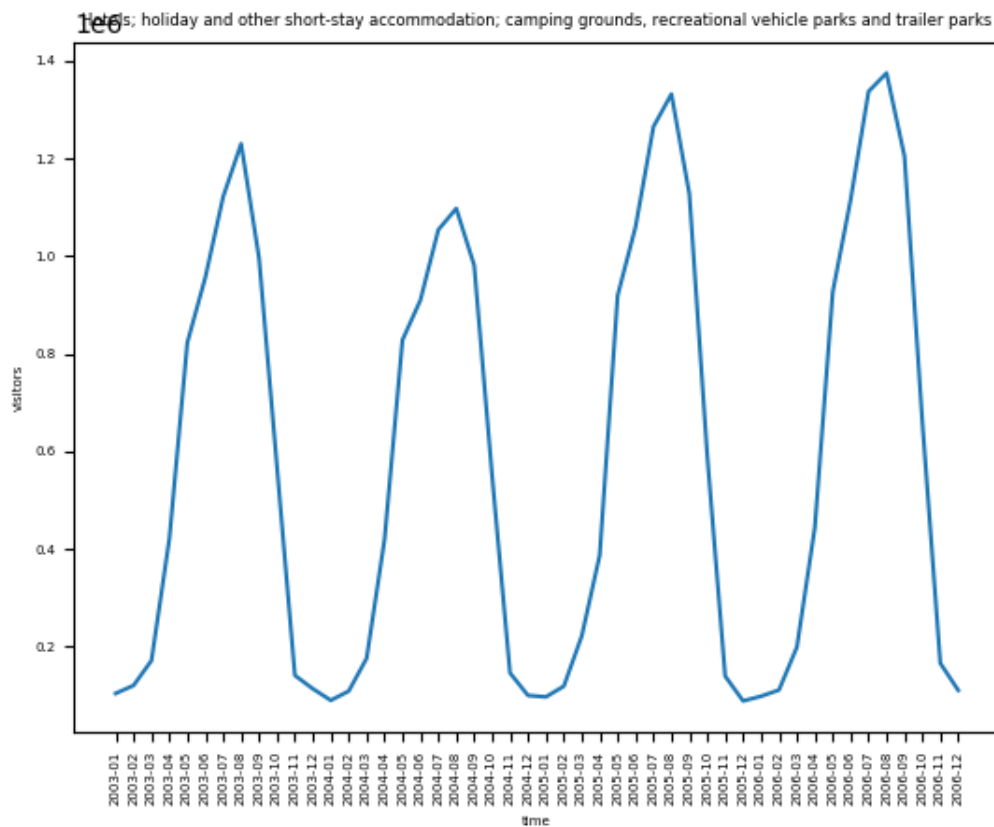
conn.commit()

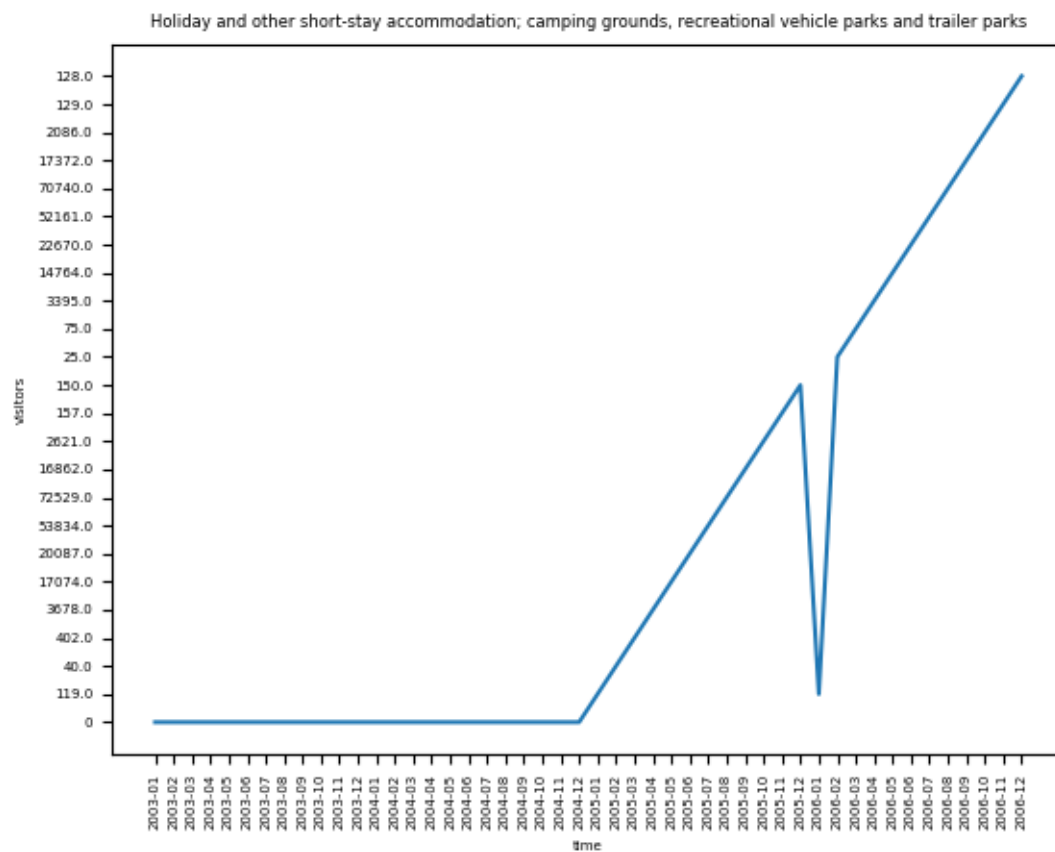
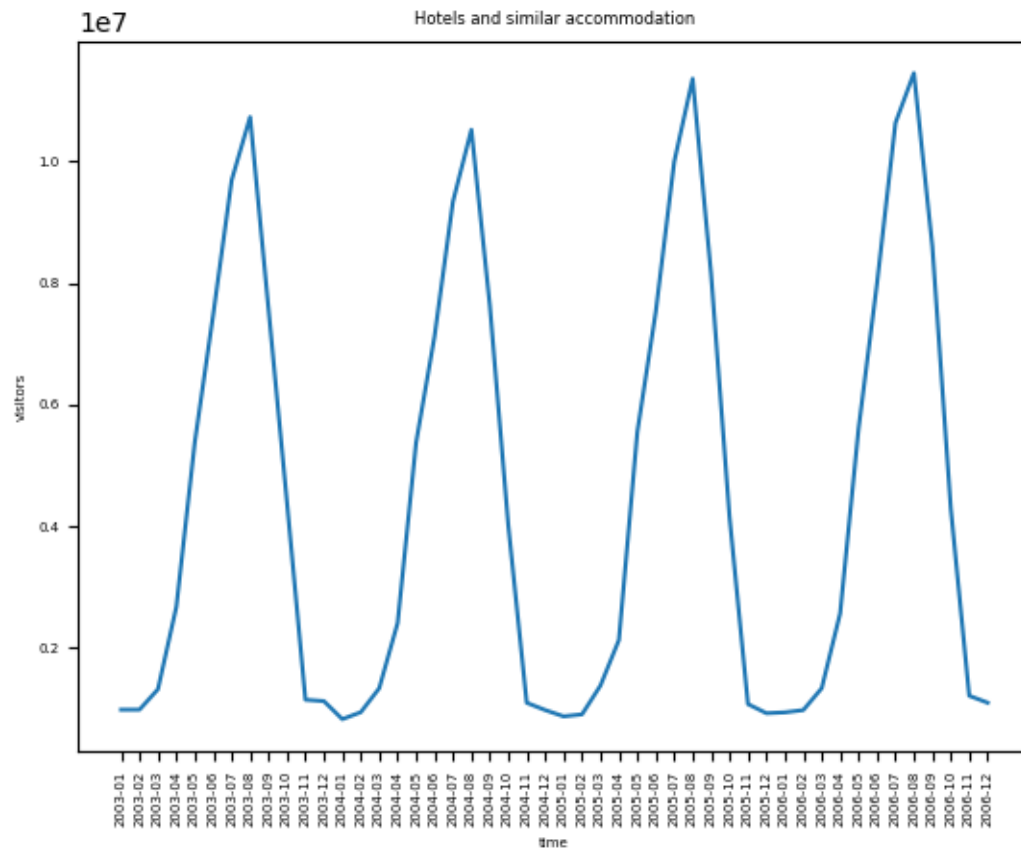
```

# Οι γραμμές ενδέχεται να έχουν μετατοπιστεί κατά την διάρκεια του copy / paste στην αναφορά



Ακολουθούν μερικά διαγράμματα από το τελικά αποτέλεσμα. Υπάρχουν όλα στα συνημμένα αρχεία της εργασίας.

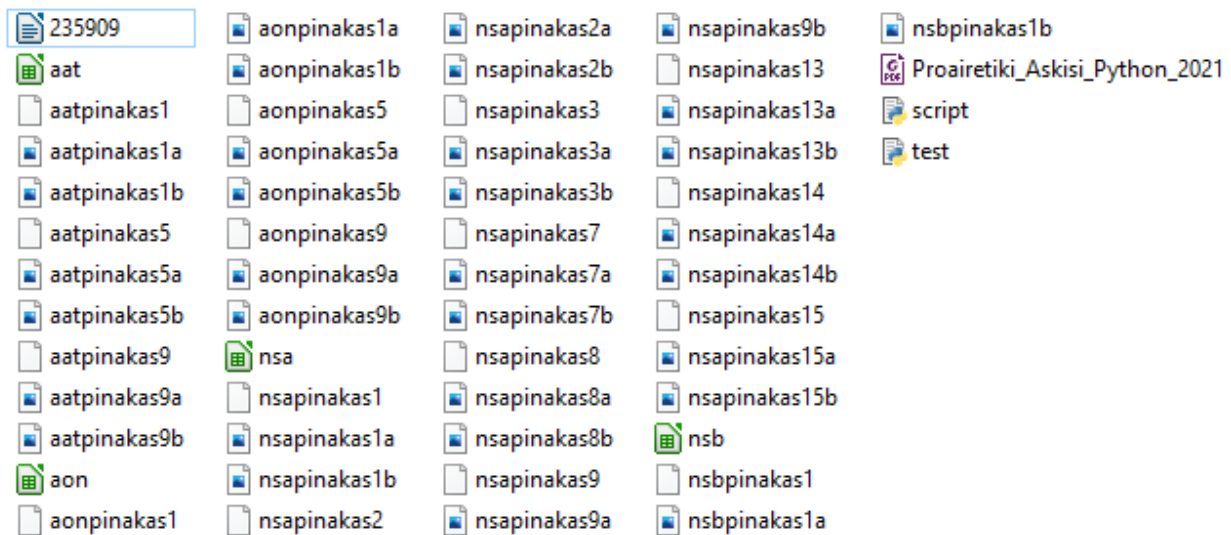




Το αποτέλεσμα στην βάση δεδομένων μετά το τρέξιμο του προγράμματος. Απλά τα αποθήκευσα μεσά στην βάση. Δεν υπάρχει καμία σύνδεση μεταξύ τιμών, πινάκων.

Filters		Containing the word: <input type="text"/>						
Table	Action	Rows	Type	Collation	Size	Overhead		
<input type="checkbox"/> aatpinakas1	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> aatpinakas5	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> aatpinakas9	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> aonpinakas1	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> aonpinakas5	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> aonpinakas9	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas1	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas2	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas3	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas7	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas8	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas9	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas13	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas14	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsapinakas15	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-		
<input type="checkbox"/> nsbpinakas1	★ Browse Structure Search Insert Empty Drop	48	InnoDB	latin1_swedish_ci	16.0 KiB	-		
16 tables	Sum	720	InnoDB	latin1_swedish_ci	256.0 KiB	0 B		

Τα περιεχόμενα του directory μετά την εκτέλεση του script και του προγράμματος



\*To test.py είναι το πρόγραμμα.

Από την στιγμή που δεν μας ενδιαφέρει όντως να δούμε τα διαγράμματα και να αντλήσουμε πληροφορίες από εκεί, αλλά μας ενδιαφέρει ότι έχουμε κατανοήσει το πως μέσω της python μπορούμε να κατεβάσουμε αρχεία, τα οποία να διαχειριστούμε, να τα φορτώσουμε σε βάση δεδομένων ή να τα εξάγουμε σε αρχεία και τέλος να φτιάξουμε διαγράμματα στις κατηγορίες aon, nsb, επιλέχτηκαν τυχαία 3-4 sheets.

*Παράδειγμα ενός csv αρχείου*

CSV Viewer

	TIME	BG	GR
7	2003-07	33.30	-0.90
8	2003-08	43.20	3.60
9	2003-09	39.90	-6.40
10	2003-10	29.20	-0.50
11	2003-11	37.60	-4.20
12	2003-12	16.90	-1.50
13	2004-01	0.00	-13.40
14	2004-02	0.00	-9.90
15	2004-03	0.00	3.30
16	2004-04	0.00	-0.70
17	2004-05	0.00	0.50
18	2004-06	0.00	-4.40
19	2004-07	0.00	-4.90
20	2004-08	0.00	-9.20
21	2004-09	0.00	-1.50
22	2004-10	0.00	-3.30
23	2004-11	0.00	3.50
24	2004-12	0.00	-11.90
25	2005-01	0.00	7.95
26	2005-02	0.00	9.60
27	2005-03	0.00	25.93
28	2005-04	0.00	-7.82
29	2005-05	0.00	10.79
30	2005-06	0.00	16.36
31	2005-07	0.00	18.75
32	2005-08	0.00	19.33
33	2005-09	0.00	15.25
34	2005-10	0.00	7.84
35	2005-11	0.00	-4.48

