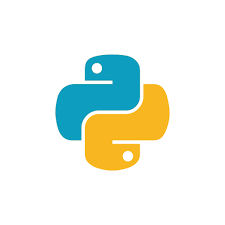
**Προαιρετική Εργαστηριακή Άσκηση Python**

**Έτος 2019-2020**

****

**Κονταρίνης Απόστολος**

**ΑΜ: 1059565**

**Έτος: 3ο**

**st1059565@ceid.upatras.gr**

**up1059565@upnet.gr**

****

**Περιεχόμενα:**

Ζητούμενα

Παραδοτέα

Τελικά γραφήματα

Screenshots λειτουργίας

Κώδικες που υλοποιήθηκαν (χωρίς τα σχόλια τους για λόγους εμφάνισης)

**Ζητούμενα:**

Στα πλαίσια της εργαστηριακής άσκησης ζητήθηκε:

Η λήψη αρχείων τύπου xls από την ιστοσελίδα της ΕΛΣΤΑΤ από την ενότητα του Τουρισμού με στοιχεία για της αφίξεις τουριστών τα έτη 2011, 2012, 2013, 2014

Επεξεργασία των αρχείων αυτών και εξαγωγή των χρήσιμων δεδομένων. Αποθήκευση των δεδομένων σε μία βάση δεδομένων sqlite3 καθώς και εγγραφή αυτών σε αρχεία τύπου csv.

Τέλος, ζητήθηκε η σχεδίαση γραφημάτων από τα στοιχεία στα αρχεία csv.

1ο γράφημα: Συνολικές αφίξεις τουριστών στην Ελλάδα για την τετραετία 2011-2014

2ο γράφημα: Χώρες καταγωγής με το μεγαλύτερο μερίδιο στις αφίξεις τουριστών στην Ελλάδα για την τετραετία 2011-2014

3ο γράφημα: Αφίξεις τουριστών στην Ελλάδα ανά μέσο μεταφοράς για την τετραετία 2011-2014

4ο γράφημα: Αφίξεις τουριστών στην Ελλάδα ανά τρίμηνο για την τετραετία 2011-2014

**Παραδοτέα**

Αντίστοιχα με τα ζητούμενα της άσκησης υλοποιήθηκαν 3 script Python 3 με ονόματα:

xls\_download.py

κατέβασμα αρχείων xls

process\_xls.py

διάβασμα xls αρχείων, δημιουργία και προσθήκες σε βάση sqlite3, δημιουργία αρχείων csv και εγγραφή δεδομένων σε αυτά

matplot.py

διάβασμα csv αρχείων και δημιουργία – παρουσίαση γραφημάτων

Σε αυτό το σημείο ενημερώνεται ο αναγνώστης πως έχει υπάρξει μέριμνα ο κώδικας να είναι ευανάγνωστος και πλήρως σχολιασμένος ώστε να μπορεί κάποιος να τον κατανοήσει και χωρίς την αναφορά.

Παρακάτω παρατίθεται αναφορά της εργασία εις βάθος:

**xls\_download.py**

Το script αποτελείται από 3 SECTIONS όπως αναγράφεται και μέσα στον κώδικα.

Το SECTION 1 εξυπηρετεί το να μπορεί ο κώδικα να τρέχει σε οποιονδήποτε Η/Υ και να αποφευκτούν πιθανά προβλήματα με τα paths των αρχείων, έχει υλοποιηθεί ένα κομμάτι κώδικα για την εύρεση του path του ιδίου του script αρχείου, με την βοήθεια συναρτήσεων της βιβλιοθήκης os όπως η os.path.dirname(os.path.realpath(\_\_file\_\_))και την μετατροπής του από μορφή πχ C:\Users\my\_user\... σε C:/User/my\_user/… δηλαδή αλλαγή του «\» με «/» για λόγους συμβατότητας με κάποιες εντολές. Επιπλέον κατασκευάζει το path αποθήκευσης (SECTION 1 – END SECTION 1 όπως αναγράφεται μέσα στον κώδικα)

Το SECTION 2 ορίζει τις μεταβλητές που περιέχουν τα links για τα ζητούμενα xls αρχεία (SECTION 2 – END SECTION 2 όπως αναγράφεται μέσα στον κώδικα).

Το SECTION 3 κάνει χρήση της συνάρτησης urllib.request.ulrretrieve() της βιβλιοθήκης urllib.request, με όρισμα τα links του SELECTION 2 και το path αποθήκευσης για να κατεβάσει τα αρχεία xls (SECTION 3 – END SECTION 3 όπως αναγράφεται μέσα στον κώδικα).

**process\_xls.py**

Το script αποτελείται από 3 SECTIONS όπως αναγράφεται και μέσα στον κώδικα.

Το SECTION 1 εξυπηρετεί το να μπορεί ο κώδικα να τρέχει σε οποιονδήποτε Η/Υ και να αποφευκτούν πιθανά προβλήματα με τα paths των αρχείων, έχει υλοποιηθεί ένα κομμάτι κώδικα για την εύρεση του path του ιδίου του script αρχείου, με την βοήθεια συναρτήσεων της βιβλιοθήκης os όπως η os.path.dirname(os.path.realpath(\_\_file\_\_))και την μετατροπής του από μορφή πχ C:\Users\my\_user\... σε C:/User/my\_user/… δηλαδή αλλαγή του «\» με «/» για λόγους συμβατότητας με κάποιες εντολές. Επιπλέον, κατασκευάζει το path δημιουργίας και αποθήκευσης της βάση δεδομένων, αρχικοποιεί με 0 τους create\_e\_counter (create error counter για τα errors στις εντολές CREATE TABLE που ακολουθούν) και ins\_e\_counter (insert error counter για τα errors στις εντολές insert που ακολουθούν και αρχικοποιεί το csv header που θα χρησιμοποιηθεί (SECTION 1 – END SECTION 1 όπως αναγράφεται μέσα στον κώδικα).

Το SECTION 2 δημιουργεί της βάση δεμομένων και τον connector της με την χρήση της sqlite3.connect() της βιβλιοθήκης sqlite3 και όρισμα το path της βάσης δεδομένων που κατασκευάστηκε στο προηγούμενο SECTION. Στην συνέχεια αρχικοποιείται cursor για την βάση με την συνάρτηση “connector\_name”.cursor() της βιβλιοθήκης sqlite3 και δημιουργείται μία λίστα με «ανοιγμένα» αρχεία, που κατέβηκαν στο προηγούμενο script (xls\_download.py), μέσω της χρήσης της συνάρτησης xlrd.open\_workbook() της βιβλιοθήκης xlrd με όρισμα το path του csv αρχείου που συντάσσεται επί τόπου κάνοντας χρήση του path που επιστράφηκε από το SECTION 1 (SECTION 2 – END SECTION 2 όπως αναγράφεται μέσα στον κώδικα).

Στο SECTION 3 γίνεται η επεξεργασία των xls αρχείων. Αρχικά αρχικοποιήται μία μεταβλητή date = 2011 η οποία χρησιμοποιείται για την κατασκευή ονομάτων αρχείων csv και πινάκων βάσης. Για κάθε xls αρχείο (της λίστας των xls αρχείων που δημιουργήθηκε στο προηγούμενο SECTION), λαμβάνεται ο αριθμός των Φύλλων του (που λόγο της επιλογής που έγινε κατά το κατέβασμα είναι 12, ένα για κάθε μήνα), για κάθε μήνα κατασκευάζεται ένα όνομα πίνακα με την αριθμητική του αντιστοίχιση και την μεταβλητή date και γίνεται try except για την δημιουργία πίνακα στην βάση δεδομένων . Αυτό πραγματοποιείται με χρήση της “cursor\_name”.execute() και όρισμα την εντολή CREATE για τον ανάλογο πίνακα της sqlite3 και της “connector\_name”.commit(). Αν η δημιουργία πίνακα είναι επιτυχείς εμφανίζεται κατάλληλο μήνυμα, ενώ αν είναι ανεπιτυχής αυξάνεται ο μετρήτης σφαλμάτων create\_e\_counter και τυπώνονται ανάλογα μηνύματα. Στην συνέχεια δημιουργείται μία λίστα λιστών στην οποία θα αποθηκεύονται λίστες που περιέχουν γραμμές του προς ανάγνωση xls αρχείου, ενώ παράλληλα λαμβάνονται οι αριθμοί index για το sheet που γίνεται ανάγνωση καθώς και οι αριθμοί πλήθους στηλών και γραμμών αυτού. Μετά από αυτά σαρώνεται κατά γραμμές το φύλλο με κριτήριο αν το πρώτο κελί της γραμμή ανήκει σε μία κανονική έκφραση που ξεχωρίζει στα συγκεκριμένα αρχεία τα χρήσιμα δεδομένα. Συγκεκριμένα παρατηρήθηκε ότι τα δεδομένα που ήταν χρήσιμα στο πρώτο κελί της γραμμή είχα έναν αριθμό και μία τελεία δηλαδή κανονική έκφραση

Η έκφραση των κανονικών εκφράσεων έγινε με την βιβλιοθήκη re. Με βάση αυτό, για κάθε γραμμή που πληρεί της προδιαγραφή δημιουργείται μία λίστα στην οποία εισάγονται τα στοιχεία της. Αυτές οι λίστες μετά εισάγονται στην μεγαλύτερη λίστα. Σε αυτό το σημείο η μεγαλύτερη λίστα περιέχει όλες τις λίστες εκείνες που έχουν τα χρήσιμα στοιχεία του xls αρχείου. Γίνεται σύνταξη του insert query της sqlite3 και με try-except δοκιμάζεται με την η εισαγωγή. Η εισαγωγή γίνεται με την εντολή “cursor\_name”.execute() και ορίσματα το insert query και data values variable. Η επιτυχία της εισόδου έχει αντίστοιχο μήνυμα, ενώ για την αποτυχία μαζί με αντίστοιχο μήνυμα περιλαμβάνεται και η αύξηση και τύπωση του ins\_e\_counter που μετράει τις αποτυχημένες εισόδους. Μαζί με την εισαγωγή στην βάση δημιουργούνται τα csv αρχεία (με header αυτό του SECTION 1) ( αντίστοιχα με τους πίνακες της βάσης) και τα λίστες με τα στοιχεία των επιθυμητών γραμμών γράφονται επίσης εκεί. Γίνεται χρήση εντολών όπως open(), writerow() της βιβλιοθήκης csv. Στο τέλος της επανάληψης για το κάθε αρχείο xls καθαρίζονται οι λίστες και η μεταβλητή date αυξάνεται, ενώ εκτός όλων των επαναλήψεων και ο connector της βάσης με την εντολή “connector\_name”.close() της βιβλιοθήκης sqlite3 (SECTION 3 – END SECTION 3 όπως αναγράφεται μέσα στον κώδικα).

**matplot.py**

Το script αποτελείται από 4 SECTIONS όπως αναγράφεται και μέσα στον κώδικα.

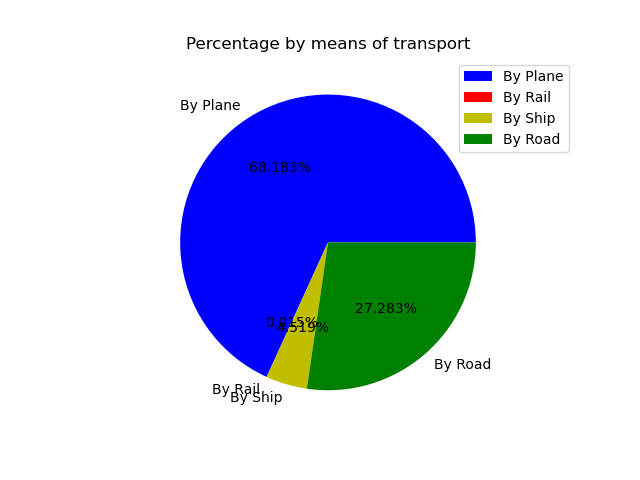
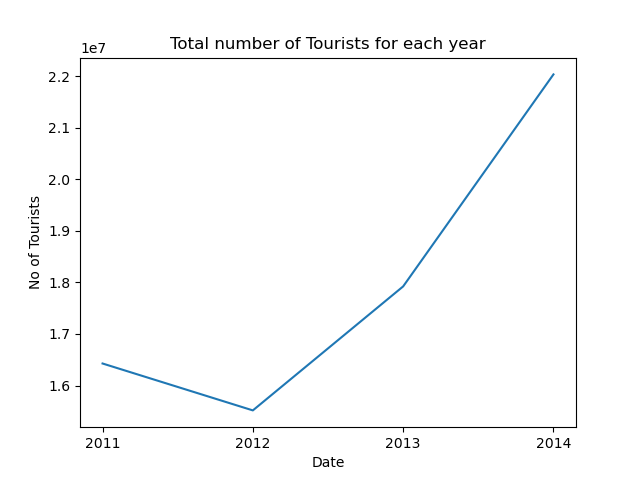
Το SECTION 1 εξυπηρετεί το να μπορεί ο κώδικα να τρέχει σε οποιονδήποτε Η/Υ και να αποφευκτούν πιθανά προβλήματα με τα paths των αρχείων, έχει υλοποιηθεί ένα κομμάτι κώδικα για την εύρεση του path του ιδίου του script αρχείου, με την βοήθεια συναρτήσεων της βιβλιοθήκης os όπως η os.path.dirname(os.path.realpath(\_\_file\_\_))και την μετατροπής του από μορφή πχ C:\Users\my\_user\... σε C:/User/my\_user/… δηλαδή αλλαγή του «\» με «/» για λόγους συμβατότητας με κάποιες εντολές. Επιπλέον γίνεται αρχικοποίηση μεταβλητών και λιστών που χρησιμοποιούνται στο πρόγραμμα (SECTION 1 – END SECTION 1 όπως αναγράφεται μέσα στον κώδικα).

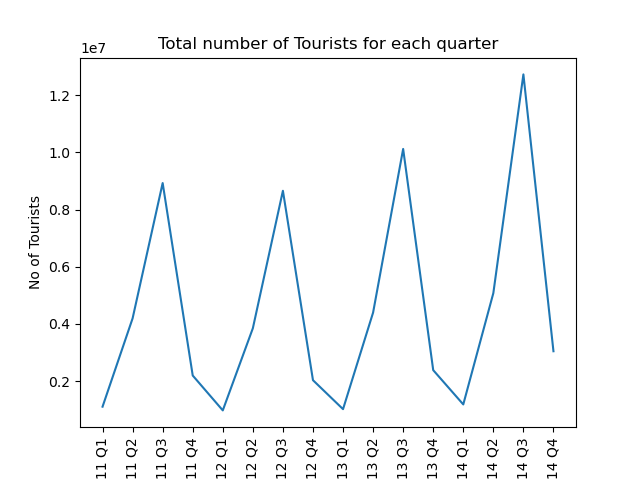
Στο SECTION 2 ανατρέχονται τα csv αρχείαμε εντολές της βιβλιοθήκης csv και καταμετρούνται οι συνολικοί τουρίστες ανά έτος, οι τουρίστες ανά μέσο, οι τουρίστες ανά τρίμηνο και με χρήση dictionaries καταμετρούνται οι τουρίστες ανά χώρα για τα τέσσερα έτη. Αντιμετωπίστηκαν κάποια προβλήματα με κενές θέσεις στα csv αρχεία τα οποία λύθηκαν με αντικατάσταση των κενών από «0» (SECTION 2 – END SECTION 2 όπως αναγράφεται μέσα στον κώδικα).

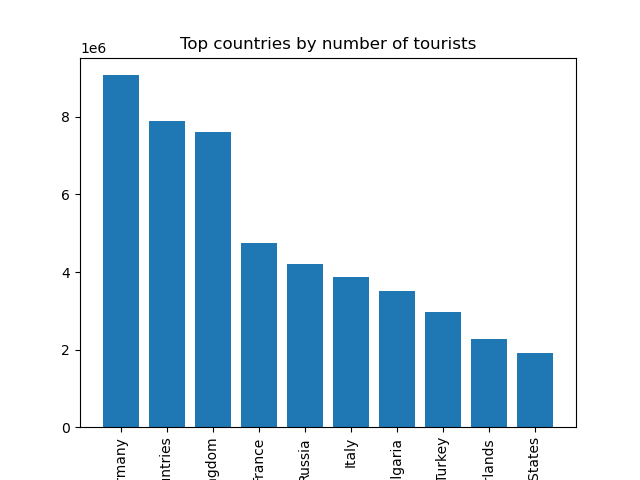
Στο SECTION 3 βρίσκονται τα διαγράμματα. Ως plt1 από την βιβλιοθήκη matplotlib.pyplot έχει οριστεί το διάγραμμα τουριστών ανά έτους, το οποίο μπορεί να ανακτηθεί με την συνάρτηση plot\_total\_tourists(). Ως plt2 έχει οριστεί το διάγραμμα πίτα που αναπαριστά την χρήση μέσων μεταφορά από τους τουρίστες για να μεταβούν στην Ελλάδα, το οποίο μπορεί να ανακτηθεί με την συνάρτηση plot\_means\_of\_transport(). Ως plt3 έχει οριστεί το διάγραμμα τουριστών ανά εξάμηνο, το οποίο μπορεί να ανακτηθεί με την συνάρτηση plot\_tourists\_by\_quarter() και ως plt4 το ραβδόγραμμα με τις πρώτες 10 χώρες σε αφίξεις, το οποίο μπορεί να ανακτηθεί με την συνάρτηση plot\_top\_countries(). Για να επιτευχθεί αυτό το γράφημα έγινε χρήση της βιβλιοθήκης operator για να ταξινομηθεί με φθίνουσα σειρά με βάση τα keys το dictionary που περιλαμβάνει τις χώρες με τις αφίξεις τους. Στην συνέχεια το ταξινομημένο dictionary γίνεται reverse και οι πρώτες 10 τιμές του αποθηκεύονται σε 2 λίστες για την σχεδίαση του ραβδογράμματος (SECTION 3 – END SECTION 3 όπως αναγράφεται μέσα στον κώδικα).

Στο SECTION 4 κατασκευάζεται το GUI του προγράμματος. Αυτό γίνεται με χρήση της βιβλιοθήκη tkinter. Αρχικά αρχικοποιείται το παράθυρο, ο τίτλος του, το μέγεθος του καθώς και η εικόνα background. Επιπλέον δημιουργούνται 4 κουμπιά τα οποία καλούν τις συναρτήσεις του SECTION 3 και με κλικ πάνω τους εμφανίζονται τα αντίστοιχα (με το όνομα του κουμπιού) διαγράμματα (SECTION 4 – END SECTION 4 όπως αναγράφεται μέσα στον κώδικα).

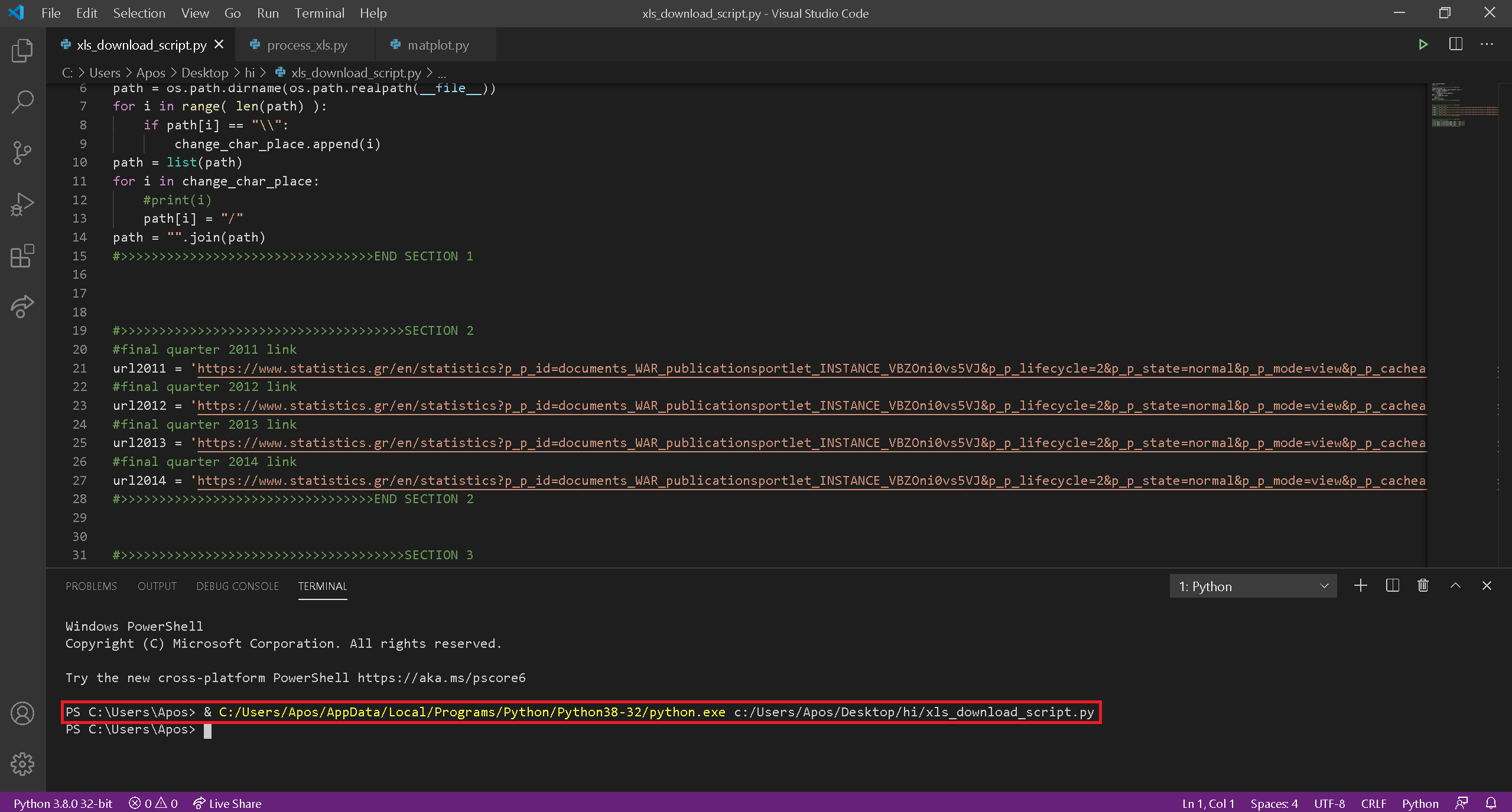
**Τελικά Γραφήματα**

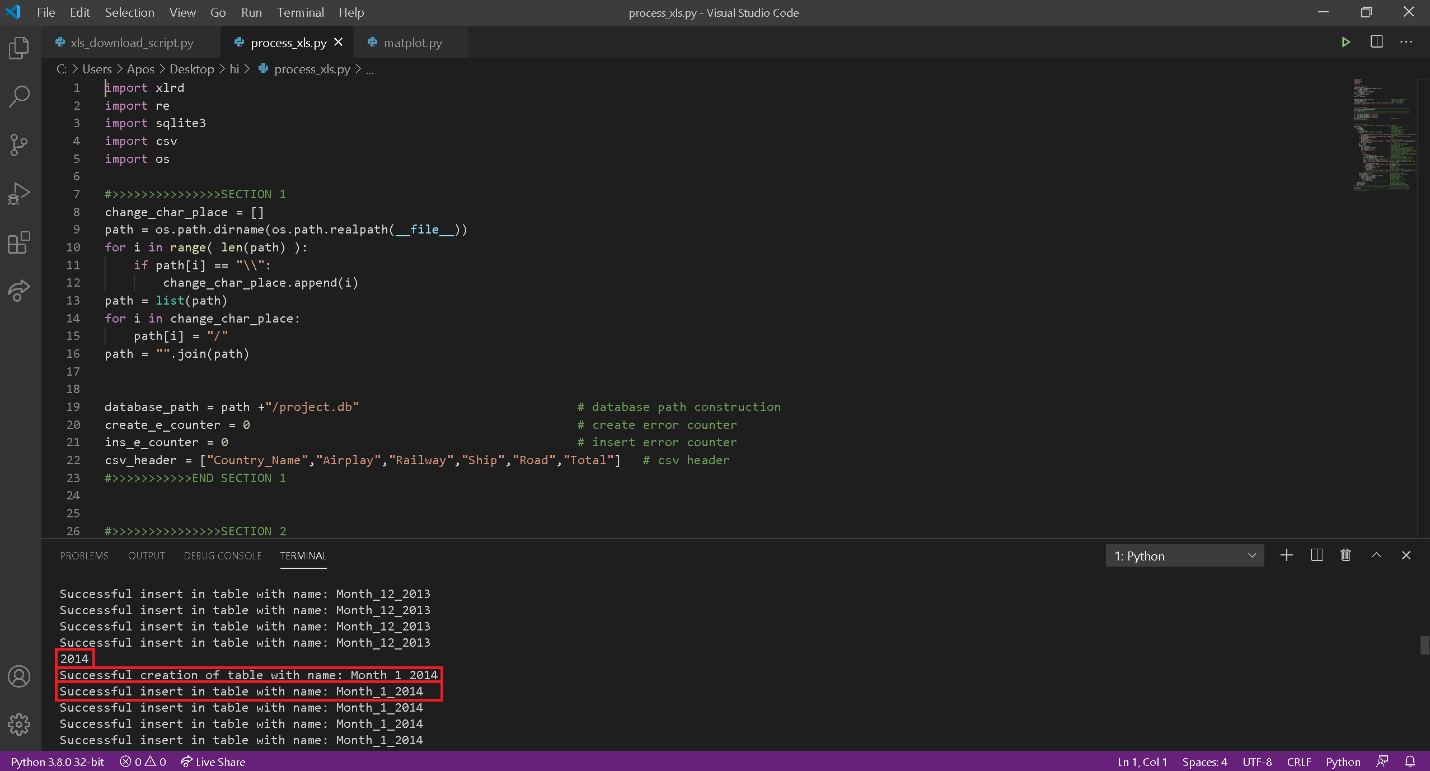


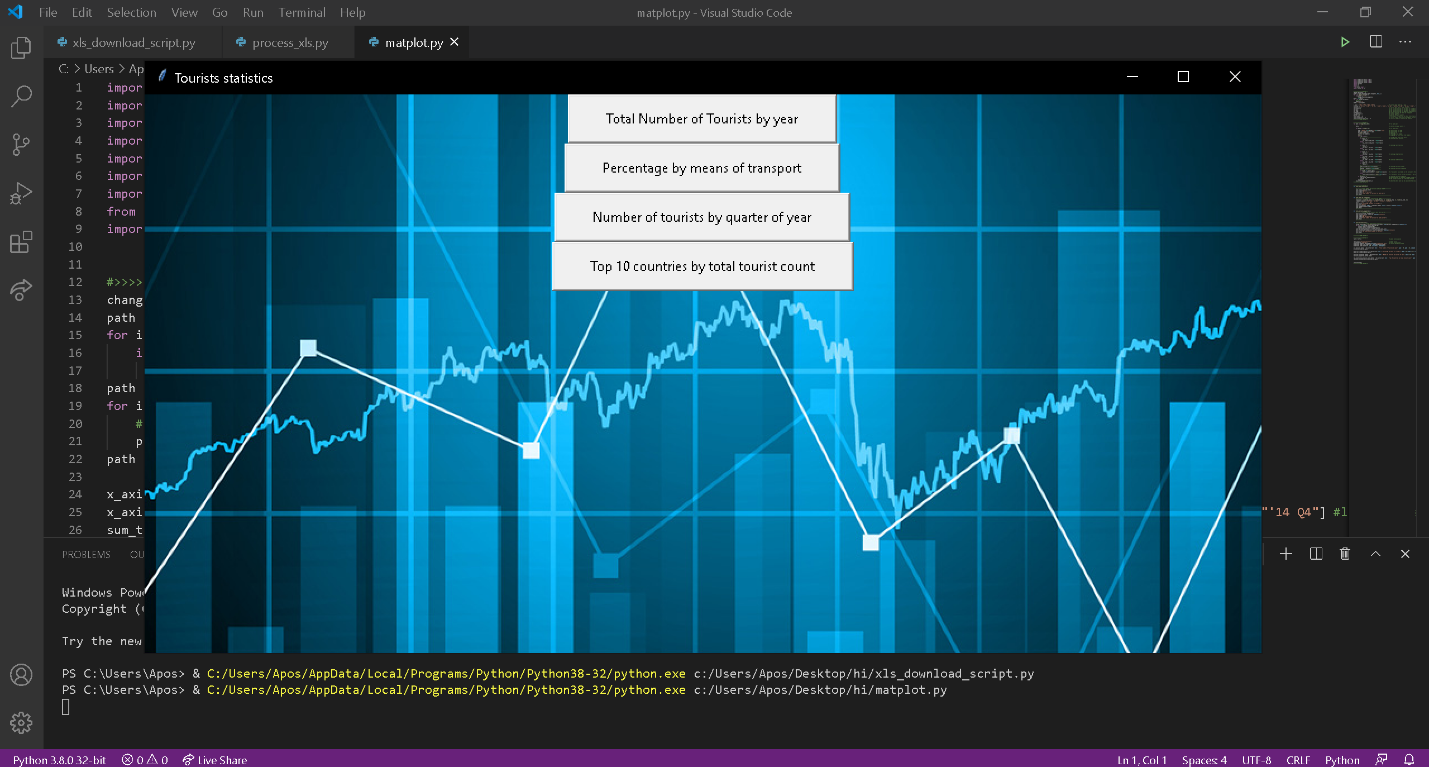


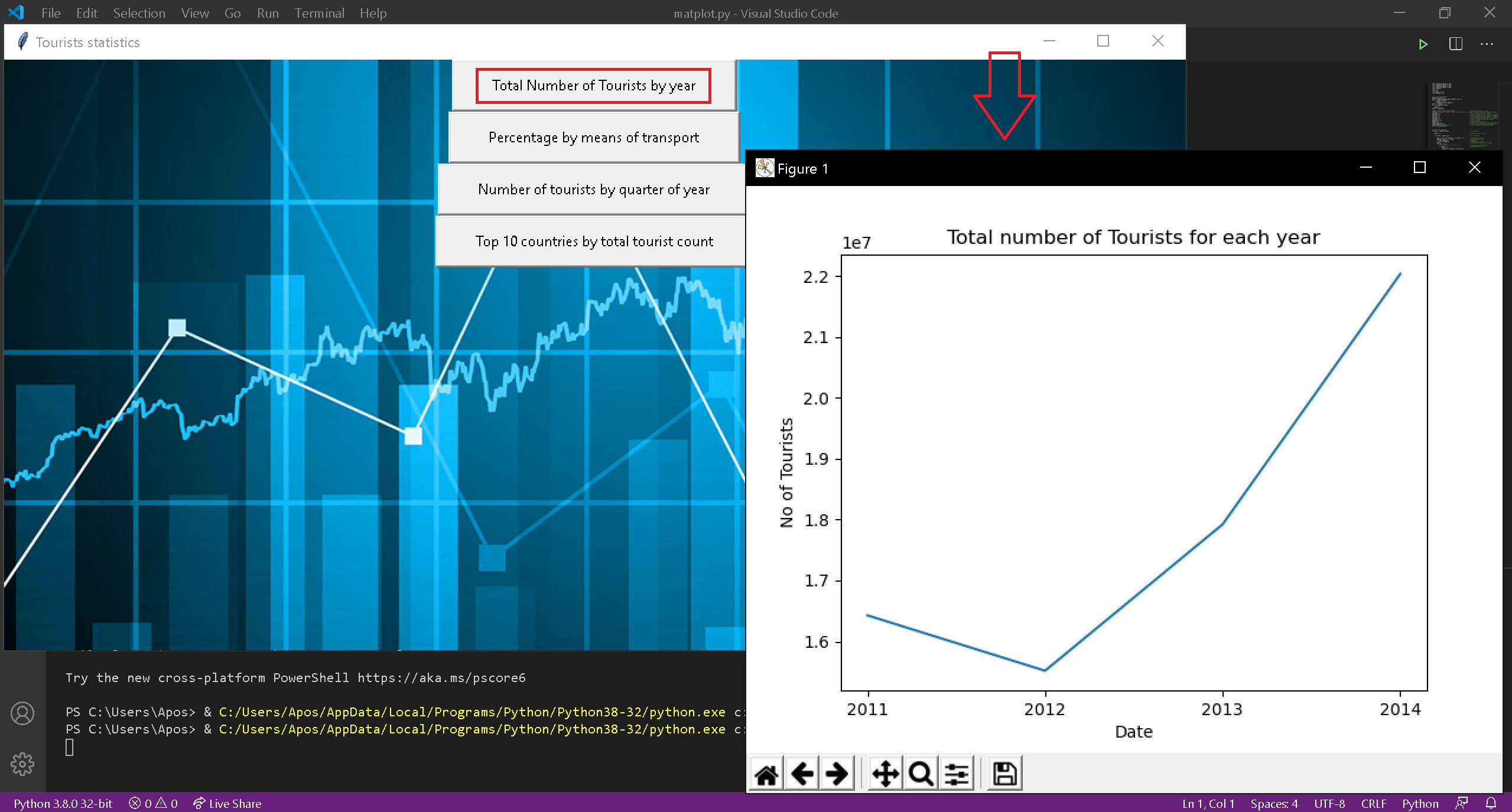
****

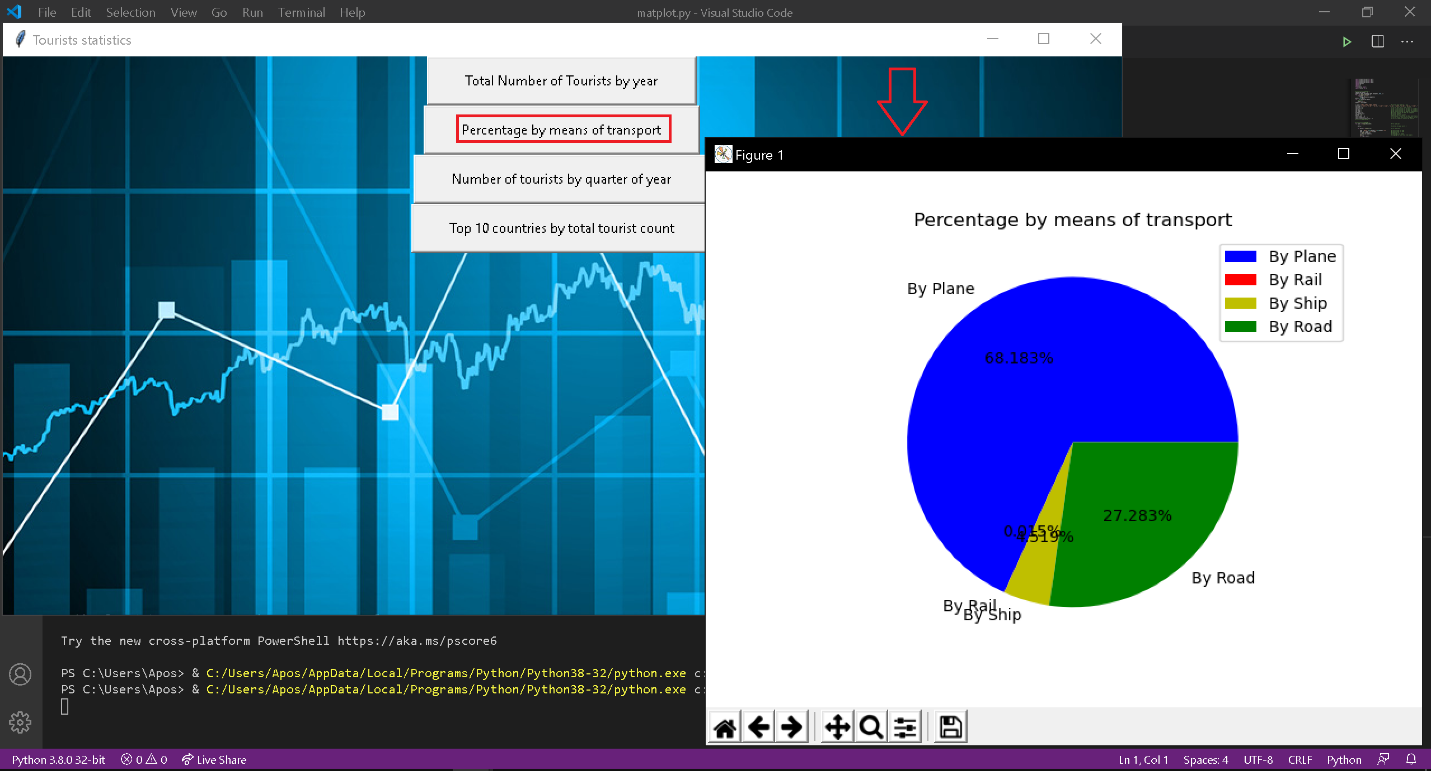
**Screenshot λειτουργίας:**

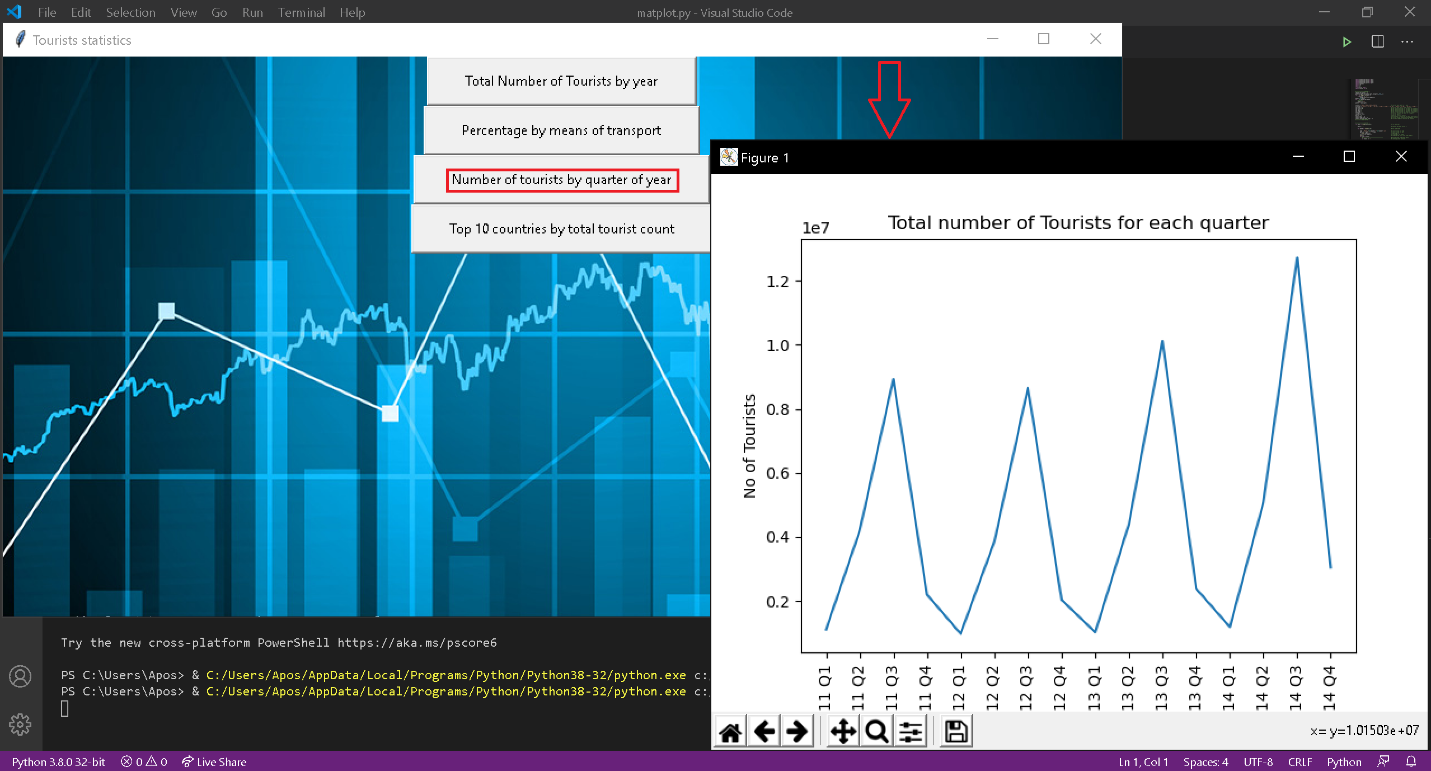


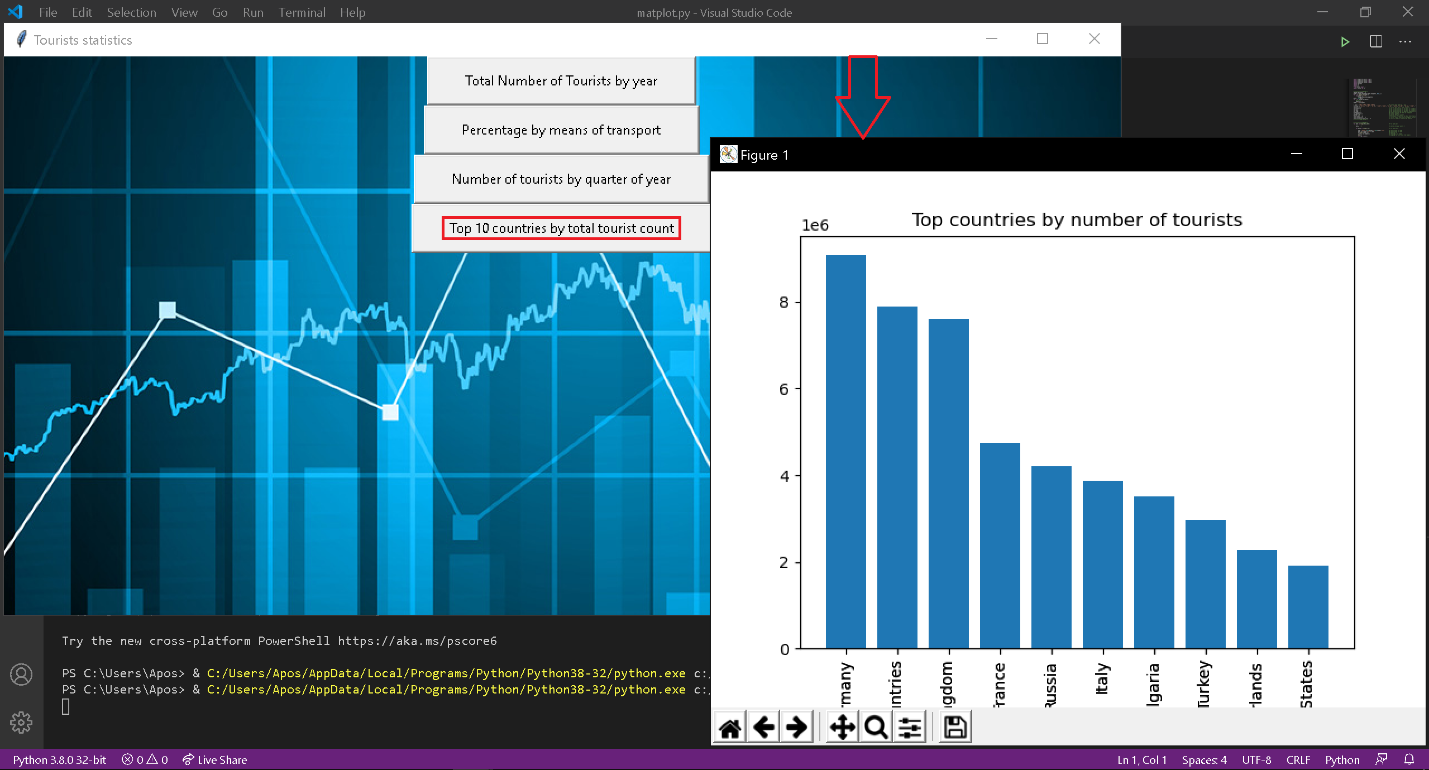












**Κώδικας αρχείου xls\_download.py:**

import urllib.request

import os

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>SECTION 1

change\_char\_place = []

path = os.path.dirname(os.path.realpath(\_\_file\_\_))

for i in range( len(path) ):

    if path[i] == "\\":

        change\_char\_place.append(i)

path = list(path)

for i in change\_char\_place:

    #print(i)

    path[i] = "/"

path = "".join(path)

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>END SECTION 1

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>SECTION 2

#final quarter 2011 link

url2011 = 'https://www.statistics.gr/en/statistics?p\_p\_id=documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ&p\_p\_lifecycle=2&p\_p\_state=normal&p\_p\_mode=view&p\_p\_cacheability=cacheLevelPage&p\_p\_col\_id=column-2&p\_p\_col\_count=4&p\_p\_col\_pos=2&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_javax.faces.resource=document&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_ln=downloadResources&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_documentID=113865&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_locale=en'

#final quarter 2012 link

url2012 = 'https://www.statistics.gr/en/statistics?p\_p\_id=documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ&p\_p\_lifecycle=2&p\_p\_state=normal&p\_p\_mode=view&p\_p\_cacheability=cacheLevelPage&p\_p\_col\_id=column-2&p\_p\_col\_count=4&p\_p\_col\_pos=2&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_javax.faces.resource=document&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_ln=downloadResources&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_documentID=113886&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_locale=en'

#final quarter 2013 link

url2013 = 'https://www.statistics.gr/en/statistics?p\_p\_id=documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ&p\_p\_lifecycle=2&p\_p\_state=normal&p\_p\_mode=view&p\_p\_cacheability=cacheLevelPage&p\_p\_col\_id=column-2&p\_p\_col\_count=4&p\_p\_col\_pos=2&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_javax.faces.resource=document&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_ln=downloadResources&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_documentID=113905&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_locale=en'

#final quarter 2014 link

url2014 = 'https://www.statistics.gr/en/statistics?p\_p\_id=documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ&p\_p\_lifecycle=2&p\_p\_state=normal&p\_p\_mode=view&p\_p\_cacheability=cacheLevelPage&p\_p\_col\_id=column-2&p\_p\_col\_count=4&p\_p\_col\_pos=2&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_javax.faces.resource=document&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_ln=downloadResources&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_documentID=113925&\_documents\_WAR\_publicationsportlet\_INSTANCE\_VBZOni0vs5VJ\_locale=en'

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>END SECTION 2

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>SECTION 3

#download of above xls fils

urllib.request.urlretrieve(url2011, path + "/2011.xls")

urllib.request.urlretrieve(url2012, path + "/2012.xls")

urllib.request.urlretrieve(url2013, path + "/2013.xls")

urllib.request.urlretrieve(url2014, path + "/2014.xls")

#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>END SECTION 3

**Κώδικας αρχείου process\_xls.py:**

import xlrd

import re

import sqlite3

import csv

import os

#>>>>>>>>>>>>>>>SECTION 1

change\_char\_place = []

path = os.path.dirname(os.path.realpath(\_\_file\_\_))

for i in range( len(path) ):

    if path[i] == "\\":

        change\_char\_place.append(i)

path = list(path)

for i in change\_char\_place:

    path[i] = "/"

path = "".join(path)

database\_path = path +"/project.db"

create\_e\_counter = 0

ins\_e\_counter = 0

csv\_header = ["Country\_Name","Airplay","Railway","Ship","Road","Total"]

#>>>>>>>>>>>END SECTION 1

#>>>>>>>>>>>>>>>SECTION 2

#---------------Connector initialization-----------------------------------------------------------

conn = sqlite3.connect(database\_path)

cur = conn.cursor()

#--------------------------------------------------------------------------------------------------

#--------------------------------------------------------------------------------------------------

a = xlrd.open\_workbook(path + "/2011.xls")

b = xlrd.open\_workbook(path + "/2012.xls")

c = xlrd.open\_workbook(path + "/2013.xls")

d = xlrd.open\_workbook(path + "/2014.xls")

docs = [a, b, c, d]

#>>>>>>>>>>>END SECTION 2

#>>>>>>>>>>>>>>>SECTION 3

#---------------XLS READING PROCESS - DB CREATE INSERT- CSV CREATE WRITE---------------------------

date = 2011

for m in docs:

    print(date)

    n = m.nsheets

    for i in range (n):

        table\_n = "Month\_"+str(i+1)+"\_"+str(date)

        try:

            cur.execute("CREATE TABLE IF NOT EXISTS "+table\_n+" (Contry\_Name text PRIMARY KEY, Airplane FLOAT, Railway FLOAT, Ship FLOAT, Road FLOAT, Total FLOAT)")

            conn.commit()

            print("Successful creation of table with name: "+table\_n)

        except sqlite3.Error as error:

            print(error)

            create\_e\_counter +=1

            print("Couldn't create table with name: " + table\_n + "Could not create "+ str(create\_e\_counter) + "tables.")

        s = m.sheet\_by\_index(i)

        s1 = s.nrows

        s2 = s.ncols

        table\_rec = []

        for j in range (s1):

            c = str(s.cell(j,0).value)

            p = re.match(r'\d.',c)

            w = re.search("Source",c)

            if w:

                break

            if p:

                record = []

                for k in range (1, s2):

                    record.append(s.cell(j,k).value)

                table\_rec.append(record)

#---------Importing data to SQLite3-----------------------------------------------

                ins\_q ="INSERT INTO "+table\_n+" VALUES(?, ?, ?, ?, ?, ?)"

                data = (str(record[0]), record[1], record[2], record[3], record[4], record[5])

                try:

                    cur.execute(ins\_q,data)

                    print("Successful insert in table with name: "+table\_n)

                    conn.commit()

                except sqlite3.Error as error:

                    ins\_e\_counter+=1

                    print(error)

                    print("Insertion failed. Number of failed insertion attemts: " + str(ins\_e\_counter))

#---------------------------------------------------------------------------------

#---------Importing data to CSV---------------------------------------------------

        csv\_n = table\_n+".csv"

        csv\_p = path + "/" + csv\_n

        with open (csv\_p, 'w', newline='') as file:

            writer = csv.writer(file)

            writer.writerow(csv\_header)

            for r in table\_rec:

                writer.writerow(r)

#---------------------------------------------------------------------------------

    record.clear()

    table\_rec.clear()

    date = date + 1

#--------------------------------------------------------------------------------------------------

#----------------------------------Connection closing----------------------------------------------

conn.close()

#--------------------------------------------------------------------------------------------------

#>>>>>>>>>>>END SECTION 3

**Κώδικας αρχείου matplot.py:**

import matplotlib.pyplot as plt1

import matplotlib.pyplot as plt2

import matplotlib.pyplot as plt3

import matplotlib.pyplot as plt4

import operator

import csv

import os

from tkinter import \*

import tkinter as tk

#>>>>>>>>>>>>>>>SECTION 1

change\_char\_place = []

path = os.path.dirname(os.path.realpath(\_\_file\_\_))

for i in range( len(path) ):

    if path[i] == "\\":

        change\_char\_place.append(i)

path = list(path)

for i in change\_char\_place:

    path[i] = "/"

path = "".join(path)

x\_axis = ["2011","2012","2013","2014"]

x\_axis\_q =["'11 Q1", "'11 Q2", "'11 Q3", "'11 Q4","'12 Q1", "'12 Q2", "'12 Q3", "'12 Q4", "'13 Q1", "'13 Q2", "'13 Q3", "'13 Q4","'14 Q1", "'14 Q2", "'14 Q3", "'14 Q4"]

sum\_total = []

sum\_plane = 0

sum\_rail = 0

sum\_ship = 0

sum\_road = 0

transport = []

sum\_quarter = []

quarter = 0

country\_dict = {}

country\_name\_list = []

country\_tourist\_count\_list = []

#>>>>>>>>>>>END SECTION 1

#>>>>>>>>>>>>>>>SECTION 2

for date in range(2011,2015):

    year\_s = 0

    for month in range(1,13):

        name = "Month\_"+str(month)+"\_"+str(date)+".csv"

        path\_csv = path + "/" + name

        f = open (path\_csv)

        csv\_f = csv.reader(f)

        next(f)

        for row in csv\_f:

            #------------------------------------

            if row[1] == '':

                row[1] = 0

                sum\_plane = sum\_plane + float(row[1])

            else:

                sum\_plane = sum\_plane + float(row[1])

            if row[2] == '':

                row[2] = 0

                sum\_rail = sum\_rail + float(row[2])

            else:

                sum\_rail = sum\_rail + float(row[2])

            if row[3] == '':

                row[3] = 0

                sum\_ship = sum\_ship + float(row[3])

            else:

                sum\_ship = sum\_ship + float(row[3])

            if row[4] == '':

                row[4] = 0

                sum\_road = sum\_road + float(row[4])

            else:

                sum\_road = sum\_road + float(row[4])

            if row[5] == '':

                row[5] = 0

            #------------------------------------

            year\_s = year\_s + float(row[5])

            #------------------------------------

            quarter = quarter + float(row[5])

            #------------------------------------

            if row[0] in country\_dict:

                country\_dict[row[0]] += round(float(row[5]))

            else:

                country\_dict[row[0]] = round(float(row[5]))

            #------------------------------------

        if month%3 == 0:

            sum\_quarter.append(quarter)

            quarter = 0

        f.close()

    sum\_total.append(round(year\_s))

#>>>>>>>>>>>END SECTION 2

#>>>>>>>>>>>>>>>SECTION 3

def plot\_total\_tourists():

    #-----------Total Number of Tourists each year CHART-----------

    plt1.plot(x\_axis,sum\_total)

    plt1.xlabel("Date")

    plt1.ylabel("No of Tourists")

    plt1.title("Total number of Tourists for each year")

    plt1.show()

    #--------------------------------------------------------------

def plot\_means\_of\_transport():

    #-----------------Means of transport PIE CHART-----------------

    transport = [round(sum\_plane, 2), round(sum\_rail, 2), round(sum\_ship, 2), round(sum\_road, 2)]

    transport\_labels = ["By Plane", "By Rail", "By Ship", "By Road"]

    colors = ['b', 'r', 'y', 'g']

    plt2.title("Percentage by means of transport")

    plt2.axis("equal")

    plt2.pie(transport, labels = transport\_labels, colors = colors, autopct='%1.3f%%')

    plt2.legend(transport\_labels)

    plt2.show()

    #--------------------------------------------------------------

def plot\_tourists\_by\_quarter():

    #----------Total Number of Tourists each year Quarter----------

    plt3.plot(x\_axis\_q, sum\_quarter)

    plt3.xticks(x\_axis\_q, x\_axis\_q, rotation='vertical')

    plt3.xlabel("Date")

    plt3.ylabel("No of Tourists")

    plt3.title("Total number of Tourists for each quarter")

    plt3.show()

    #--------------------------------------------------------------

def plot\_top\_countries():

    #-----------------Top 10 countries by tourists-----------------

    sorted\_country\_dict = sorted(country\_dict.items(), key=operator.itemgetter(1),reverse=True)

    for x in list(sorted\_country\_dict)[0:10]:

        country\_name\_list.append(x[0])

        country\_tourist\_count\_list.append(x[1])

    plt4.bar(country\_name\_list, country\_tourist\_count\_list)

    plt4.xticks(country\_name\_list, country\_name\_list, rotation='vertical')

    plt4.title("Top countries by number of tourists")

    plt4.show()

    #--------------------------------------------------------------

#>>>>>>>>>>>END SECTION 3

#>>>>>>>>>>>>>>>SECTION 4

root = tk.Tk()

root.title("Tourists statistics")

root.geometry("1000x500")

background\_image=tk.PhotoImage(file=path+"/assets/background.png")

background\_label = tk.Label(root, image=background\_image)

background\_label.place(x=0, y=0, relwidth=1, relheight=1)

all\_tourists\_button = tk.Button(root, text = "Total Number of Tourists by year", padx = 30, pady = 10, command = plot\_total\_tourists)

all\_tourists\_button.pack()

means\_of\_transport\_button = tk.Button(root, text = "Percentage by means of transport", padx = 30, pady = 10, command = plot\_means\_of\_transport)

means\_of\_transport\_button.pack()

tourists\_by\_quarter\_button = tk.Button(root, text = "Number of tourists by quarter of year", padx = 30, pady = 10, command = plot\_tourists\_by\_quarter)

tourists\_by\_quarter\_button.pack()

top\_countries\_by\_tourist\_count\_button = tk.Button(root, text = "Top 10 countries by total tourist count", padx = 30, pady = 10, command = plot\_top\_countries)

top\_countries\_by\_tourist\_count\_button.pack()

root.mainloop()

#>>>>>>>>>>>END SECTION 4

**Τέλος**