# ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ PYTHON

Αθηνά Φουσέκη, 3ο έτος
1059623
st1059623@ceid.upatras.gr

## 1. Κώδικας script για downloading αρχείων excel

```
import urllib.request
import urllib.error
import re
import wget

my_UA = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101
Firefox/74.0'


#basic loop for getting years and trimesters
for i in range(2011,2015):
    print("\n")

    url = "https://www.statistics.gr/el/statistics/-/publication/STO04/{}-
Q4".format(i) #using only the 4th trimester because the file has all the needed
data
    print(url) #printing for personal checking

    try:
        headers={}
        headers['User_Agent']= my_UA #using a user agent to guarantee our access
to the data
        req=urllib.request.Request(url, headers=headers)
        print(type(req))
        with urllib.request.urlopen(url) as response:
            print(type(response))
            char_set=response.headers.get_content_charset()
            #html=response.read().decode(char_set)       revisit this one
            html=response.read().decode("utf-8")
            #print(html)

    except urllib.error.HTTPError as e: #handling errors
        print("HTTP Error:", e.code)
    except urllib.error.URLError as e:
        print("Failed connecting to the server.")
        print("Cause: ", e.reason)

    #with this regex I'm capturing the urls I'm going to use to download the
excel files
    li=re.findall('(https://www.statistics.gr:443/el/statistics\?
p_p_id=documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ&amp;p_p_lifecycle
=2&amp;p_p_state=normal&amp;p_p_mode=view&amp;p_p_cacheability=cacheLevelPage&am
p;p_p_col_id=column-
2&amp;p_p_col_count=4&amp;p_p_col_pos=3&amp;_documents_WAR_publicationsportlet_I
NSTANCE_VBZOni0vs5VJ_javax\.faces\.resource=document&amp;_documents_WAR_publicat
ionsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&amp;_documents_WAR_public
ationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=[0-9]
{6}&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el)"
target="_blank">Αφίξεις μη κατοίκων από το εξωτερικό ανά χώρα προέλευσης ' ,
html)
    print(li)
    print("I'm here")
    for index, item in enumerate(li):
        item=item.replace("amp;", '') #replacing certain parts of the initial
url, so that I can download the files
```

```
        item=item.replace(":443",'')  #figured out the strings I wanted to
replace by comparing the url in the html file and the url I would get if I tried
downloading the file by hand
        wget.download(item,'C:/Users/athin/OneDrive/Documents/CeiD/excells3/
{}_tri4_f{}.xls'.format(i,index))

        #Have downloaded the excell files
        #name layout: year_trimester_fileNumber.xls
```

## 2. Κώδικας python για data extraction, matplot, sqlite και αρχεία csv

```
import csv
import os
import fnmatch
import xlrd
import re
import prettytable
import matplotlib
from matplotlib import pyplot as plt
import numpy as np
import sqlite3 as lite

# ----------------------------------
# take the second element for sort
def take_second(elem):
    return elem[1]


# ----------------------------------
#--------------------------------------------------------------------------------
def autolabel(rects):
    #Attach a text label above each bar in *rects*, displaying its height
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{:,d}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')



#--------------------------------------------------------------------------------
#
--------------------------------------------------------------------------------
"""
ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΓΡΑΦΗΜΑ 1
"""
def by_country(wbook):
#With this function I gather the data from the wbook that is passed as an
argument.
    sheet = wbook.sheet_by_index(11) #με αυτό επιλέγω τα sheets μέσα στο αρχείο
#I have found where in the excel file the wanted values are.
    # print(sheet.cell_value(0,0))
#I want the last sheet.
```

```
    # print("Έτος: {}".format(int(sheet.cell_value(3,3))))
#In the cell (2,C) there is the year.
    try:
#In either the cell (138,C) or (139,C) there is the number for the total of
incomings
        year = int(round(sheet.cell_value(3, 3)))
#of the year. I have used a try-except block to ensure that I'll gather the
correct data.
        total = int(round(sheet.cell_value(138, 3)))
#If the data in the  first cell is incorrect, it's going to be a string, which
cannot be
    except:
#rounded, so an exception will occur and we will try to find the data in the
other cell.
        total = int(round(sheet.cell_value(137, 3)))
#Finally we return a tuple containing the year and the total of incomings.

    return (year, total)


#
--------------------------------------------------------------------------------
# ----------------------------------------------------------------------------
"""
ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΓΡΑΦΗΜΑ 2
"""
def for_top_10(wbook):
#In this function I gather the data from the wbook that is passed as an
argument.
    top_li = []
#I have studied the structure of the excel file and parsed the lines 78 to 137.
    sheet = wbook.sheet_by_index(11)
#I used regex to ensure that the cells that I'm taking the data from contain the
    for i in range(78, 137):
#incomings of countries.
        cell = sheet.cell_value(i, 0)
#If the cells in column was a number followed by a dot, the data was correct.
        if re.search("[0-9]+\.", cell):
#I rounded and converted the collected values, because they had errors and gave
me
            # print(cell,sheet.cell_value(i,1))
            incomings = int(round(sheet.cell_value(i, 3)))
#wrong results. Also, I removed any spaces before of after the string that was
the name
            country = re.findall(".+[^0-9\(\)]", sheet.cell_value(i, 1))
#of the country.Finally, I passed the data into a list of tuples.
            country = country[0].strip()
#top_li=[(incomings1,country1), (incomings2, country2), ...]
            top_li.append((incomings, country))

    return top_li


# --------------------------------------------------------------------------------
# ----------------------------------------------
"""
ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΓΡΑΦΗΜΑ 4: Προσπάθεια για πρόσθεση αφίξεων ανά τρίμηνο
```

```
"""
def trimesters(wbook):
#In this function I'm trying to collect the data for the incomings for each

#year per trimester. I use a for loop to choose the right sheet and collect
    trims = {}
#the data for each trimester.In each loop, I collect the incomings for each

#month and I add it in the "in_total". At the end of the loop I save the data
    for i in range(0, 10, 3):
#in a dictionary. The key is the number of the trimester and the value is the
in_total.
        in_total = 0
#After that I set the "in_total" to zero.Again, I use the try-except block to
make sure
        sheet = wbook.sheet_by_index(i)
#that I collect the correct values, since there were inconsistencies refering to
the cell
        try:
#number with the wanted values.
            in_total += int(round(sheet.cell_value(66, 3)))
        except:
            in_total += int(round(sheet.cell_value(65, 3)))
        sheet = wbook.sheet_by_index(i + 1)
        try:
            in_total += int(round(sheet.cell_value(66, 3)))
        except:
            in_total += int(round(sheet.cell_value(65, 3)))
        sheet = wbook.sheet_by_index(i + 2)
        try:
            in_total += int(round(sheet.cell_value(66, 3)))
        except:
            in_total += int(round(sheet.cell_value(65, 3)))
        trims[i] = in_total

    return trims


# ------------------------------------------------------------
# ---------------------------------------------
"""
ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΓΡΑΦΗΜΑ 3
"""
def by_transportation(wbook):
#In this function I collect the incomings for each year per form of
transportation.
    sheet = wbook.sheet_by_index(11)
#I start parsing the file from line 134, and based on what I get from the cell
(line, 1)
    line = 134
#I go to the next line for the wanted data or not.
    tots = sheet.cell_value(line, 1)
#After I have found the line with the useful data, I pass the values (rounded
and converted
    while tots != "ΓΕΝΙΚΟ ΣΥΝΟΛΟ":
#to integers) in seperate variables, that I later put in a tuple for the
function to return.
```

```
        line += 1
        tots = sheet.cell_value(line, 1)
    year = (sheet.cell_value(1, 1))
    year = year[-4:]
    air = int(sheet.cell_value(line, 2))
    train = int(sheet.cell_value(line, 3))
    sea = int(sheet.cell_value(line, 4))
    car = int(sheet.cell_value(line, 5))

    return (year, air, train, sea, car)


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


"""
Πρόγραμμα που είναι συνδυασμός των από πάνω και ανάλογα με τον τίτλο του αρχείου
εκτυπώνει κατάλληλο μήνυμα.
"""
d = {} #ititializing the dictionaries I'm gonna use
d_country = {}
d_trans = {}
d_trim = {}
for_table = {}

table = prettytable.PrettyTable(['Έτος', 'Σύνολο']) #creating the tables for
printing the data
table_tri = prettytable.PrettyTable(["Έτος", "Τρι1", "Τρι2", "Τρι3", "Τρι4"])
table_trns = prettytable.PrettyTable(["Έτος", "Αερ/κώς", "Σιδ/κώς", "Θαλασσίως",
"Οδικώς"])
table_top = prettytable.PrettyTable(["Χώρες", "Αφίξεις"])

dir_path = 'C:/Users/athin/OneDrive/Documents/CeiD/excells3' #the path for the
directory with the excel files

with os.scandir(dir_path) as entries:
    for entry in entries:
        if entry.is_file() and fnmatch.fnmatch(entry, '*_tri4_*.xls'): #checking
the form of the files name
            #print(entry.name)
            dir_path2 =
('C:/Users/athin/OneDrive/Documents/CeiD/excells2/{}'.format(entry.name)) #path
to the excel file
            wbook = xlrd.open_workbook(dir_path2) #opening the file as a
workbook
            sheet = wbook.sheet_by_index(0) #selecting the sheet

            #print(sheet.cell_value(0, 0))
            title = sheet.cell_value(0, 0)

            if re.search("[.]*ΑΝΑ ΧΩΡΑ ΠΡΟΕΛΕΥΣΗΣ  ΚΑΙ ΜΕΣΟ ΜΕΤΑΦΟΡΑΣ[.]*",
title): #for files about incomings per way of transportation
                #print("Αρχείο για: Μέσο μεταφοράς")
                trnsprt = by_transportation(wbook) #calling the function to
collect the data for transportation graph
                table_trns.add_row([trnsprt[0], "{:,d}".format(trnsprt[1]),
"{:,d}".format(trnsprt[2]),
"{:,d}".format(trnsprt[3]),"{:,d}".format(trnsprt[4])]) #passing the data into a
printable table in which the numbers are formatted with ','
                d_trans[trnsprt[0]] = [trnsprt[1], trnsprt[2],trnsprt[3],
trnsprt[4]]  #passing the data into a dictionary || year is the key and the
incomings in a list the value

            else:#for files about yearly incomings
                #print("Αρχείο για: Ανά χώρα προέλευσης")
                tri = trimesters(wbook)  # calling the function
                vals = list(tri.values())  # saving the returned values in a
list
                table_tri.add_row([int(sheet.cell_value(3, 3)),
"{:,d}".format(vals[0]), "{:,d}".format(vals[1]),
                                   "{:,d}".format(vals[2]),
"{:,d}".format(vals[3])])  # adding the values in a table
                d_trim[int(sheet.cell_value(3, 3))] = [vals[0], vals[1],
                                                       vals[2], vals[3]]
#passing the data into a dictionary || year is the key and the incomings in a
list the value
                # --------------------------------------------------------------
                tot = by_country(wbook)  # calling the function
                table.add_row([tot[0], "{:,d}".format(tot[1])])  #passing the
data into a printable table in which the numbers are formatted with ','
                d_country[tot[0]] = tot[1] #pasing the data into a dictionary ||
key = year, value = incomings
                # --------------------------------------------------------------
                li = for_top_10(wbook)  # calling the function
                for item in li:
                    try:
                        d[item[1]] += item[0]  # adding the incomings of each
year
                    except:
                        d[item[1]] = item[0]  # passing the contents of the list
into a dictionary


d = sorted(d.items()) # turning the dict into a list of tuples
d = sorted(d, key=take_second) #sorted list by the incomings
d.reverse() #reversed sorted list | most visiting => least visiting

c = 0
# in d we have the countries sorted by most visiting to least visiting
for item in d:
    if c == 10: break #because I want the top 10 countries, I use this counter
    table_top.add_row([item[0], "{:,d}".format(item[1])])  #passing the data
into a printable table in which the numbers are formatted with ','
    c += 1

"""-------------------------------M A T P L O T L I
B---------------------------------------------"""
x=[]
y=[]
# in d_country I have the data for the incomings for each year
for item in d_country: #passing the data in lists to make the bars in the graph
    x.append(str(item))
    y.append(d_country[item])
```

```
fig, ax = plt.subplots()
plt.bar(x,y, color=['lightskyblue']) #setting a custom colour
ax.set_ylabel('Αφίξεις') #labeling the y axis
ax.set_title('Αφίξεις μη κατοίκων κατά την τετραετία 2011-2015 ανά έτος')
#giving a title
ax.get_yaxis().set_major_formatter(
    matplotlib.ticker.FuncFormatter(lambda x, p: format(int(x), ',')))
#formating the numbers

fig.tight_layout()
plt.show() #showing the graph
#-----------------transport----------------------------------------------------
labels=[]
air=[]
train=[]
sea=[]
car=[]
for item in d_trans: # passing the data into lists to use for the bar graph
    labels.append(item)
    air.append(d_trans[item][0])
    train.append(d_trans[item][1])
    sea.append(d_trans[item][2])
    car.append(d_trans[item][3])

x = np.arange(len(labels))  # the label locations
width = 0.50 # the width of the bars

fig, ax = plt.subplots()

#setting the characteristics for each bar
rects1 = ax.bar(x - 3*width/8, air, width/4, label='Air', color='salmon' )
rects2 = ax.bar(x - width/8, train, width/4, label='Train', color='gold')
rects3 = ax.bar(x +width/8, sea, width/4, label='Sea', color='yellowgreen')
rects4 = ax.bar(x + 3*width/8, car, width/4, label='Car', color= 'lightskyblue')

# Adding some text for labels, title and custom x-axis tick labels
ax.set_ylabel('Αφίξεις')
ax.set_title('Αφίξεις ανά μέσο μεταφοράς κάθε χρόνο')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()
ax.get_yaxis().set_major_formatter(
    matplotlib.ticker.FuncFormatter(lambda x, p: format(int(x), ',')))

#calling the function for the labeling of each bar
autolabel(rects1)
autolabel(rects2)
autolabel(rects3)
autolabel(rects4)


fig.tight_layout()

plt.show()

#--------------trims---------------------------------------------
year=[]
```

```
first=[]
sec=[]
third=[]
fourth=[]

for item in d_trim: # passing the data into lists to use for the bar graph
    year.append(item)
    first.append(d_trim[item][0])
    sec.append(d_trim[item][1])
    third.append(d_trim[item][2])
    fourth.append(d_trim[item][3])

x2 = np.arange(len(year))  # the label locations
width = 0.50 # the width of the bars

fig2, ax = plt.subplots()
trim1 = ax.bar(x - 3*width/8, first, width/4, label='First', color='salmon')
trim2 = ax.bar(x - width/8, sec, width/4, label='Second', color='gold')
trim3 = ax.bar(x +width/8, third, width/4, label='Third', color='yellowgreen')
trim4 = ax.bar(x + 3*width/8, fourth, width/4, label='Fourth',
color='lightskyblue')

# adding some text for the labels, the title and the custom x-axis tick labels
ax.set_ylabel('Αφίξεις')
ax.set_title('Αφίξεις ανά τρίμηνο για την τετραετία 2011-2015')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()
ax.get_yaxis().set_major_formatter(
    matplotlib.ticker.FuncFormatter(lambda x, p: format(int(x), ',')))

autolabel(trim1)
autolabel(trim2)
autolabel(trim3)
autolabel(trim4)

fig2.tight_layout()

plt.show()

#-------------------------
top10-----------------------------------------------------------------
labels = []
names=[]
c=0
for item in d:
    if c == 10: break
    labels.append(item[0]) #take the labels for the graph
    names.append(item) #save the top 10 countries
    c+=1


def find_percentage(li): #function to convert the incomings into percentages to
use for the pie
    d_top = []

    for item in li:  # calculate total
```

```
        try:
            total += item[1]
        except:
            total = item[1]

    for item in li:  # find the percentages
        d_top.append(item[1] * 100 / total)

    return d_top


per = find_percentage(names) #call the function for the top 10 countries
cols=['salmon', 'gold', 'yellowgreen', 'lightseagreen', 'thistle',
'mediumaquamarine', 'sandybrown', 'lightpink', 'khaki', 'lightskyblue']

fig1, ax1 = plt.subplots()
ax1.pie(per, labels=labels, colors=cols, radius=0.6, autopct='%1.1f%
%',shadow=False, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
ax1.set_title('Χώρες καταγωγής με το μεγαλύτερο μερίδιο στις αφίξεις τουριστών
')
plt.show()

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



"""# ----------------------------C S V  F I L E
S-----------------------------------------------"""



# incomings per year in csv
fields = ['έτος', 'αφίξεις']
with open('yearly.csv', 'w') as f:
    writer = csv.writer(f) #creating a writer to put the fields at the first
line of the file
    writer.writerow(fields)
    for item in d_country: #writing the data
        f.write("%s,%s\n" % (item, d_country[item]))

# incoming per way of transportation
fields1 = ['έτος', 'αεροπορικώς', 'σιδηροδρομικώς', 'θαλασσίως', 'οδικώς']
with open('transportation.csv', 'w') as f1:
    writer = csv.writer(f1)
    writer.writerow(fields1)
    for item in d_trans:
        f1.write("%s,%s,%s,%s,%s\n" % (
        item, d_trans[item][0], d_trans[item][1], d_trans[item][2],
        d_trans[item][3]))

# incomings per trimester csv
fields2 = ['έτος', '1ο', '2ο', '3ο', '4ο']
with open('trimesters.csv', 'w') as f2:
    writer = csv.writer(f2)
    writer.writerow(fields2)
    for item in d_trim:
        f2.write("%s,%s,%s,%s,%s\n" % (
        item, d_trim[item][0], d_trim[item][1], d_trim[item][2],
        d_trim[item][3]))

# top 10 countries
fields3 = ['έτος', 'χώρες']
c = 0
with open('top10.csv', 'w') as f3:
    writer = csv.writer(f3)
    writer.writerow(fields3)
    for item in d:
        if c == 10: break
        f3.write("%s,%s\n" % (item[0], "{}".format(item[1])))
        c += 1

"""-------------------------------------S Q L i t
e---------------------------------------------------------"""

c=0
try:
    conn=lite.connect('tourism_database.db') #we're creating a connection object
that represents the database by calling the connect() function
    with conn:
        curs=conn.cursor() #we're creating a cursor object by calling the
cursor() function, to be able to call its execute() method
        sql='create table top_countries(Country text primary key, Visitors
int);' #table for top 10
        curs.execute(sql)
        for item in d: #with this loop we pass the top 10 countries
            if c == 10: break #in the SQLite database
            sql1="insert into top_countries values('{}','{}');"
            curs.execute(sql1.format(item[0],"{:,d}".format(item[1])))
            c+=1

        sql2='create table in_each_year(Year int primary key, Visitors int);'
#table for the incomings for each year
        curs.execute(sql2)
        for item in d_country:
            sql3="insert into in_each_year values('{}','{}');"
            curs.execute(sql3.format(item,"{:,d}".format(d_country[item])))


        sql4='create table by_transportation(Year int primary key, by_air int,
by_sea int, by_train int, by_car int);' #table for the incomings for each year
        curs.execute(sql4)
#per form of transportation
        for item in d_trans:
            sql5="insert into by_transportation values('{}','{:,d}', '{:,d}',
'{:,d}', '{:,d}');"
            curs.execute(sql5.format(item,d_trans[item][0],d_trans[item]
[1],d_trans[item][2],d_trans[item][3]))


        sql6='create table by_trimester(Year int primary key, First_trim int,
Second_trim int, Third_trim int, Fourth_trim int);' #table for the incomings for
each
```

```
        curs.execute(sql6)
#year per trimester
        for item in d_trim:
            sql7="insert into by_trimester values('{}','{:,d}', '{:,d}',
'{:,d}', '{:,d}');"
            curs.execute(sql7.format(item,d_trim[item][0],d_trim[item]
[1],d_trim[item][2], d_trim[item][3]))


except lite.Error as er:
    print(er)
conn.close()

"""--------------------------------------------------------------------------------
-----------------------------------"""


print(table)  # printing the table for incomings for each year
print(table_tri)  # printing the table for trimesters
print(table_trns)  # printing the table for transportation
print("~~~~~~~~~~~~~~~TOP 10 ΧΩΡΕΣ~~~~~~~~~~~~~~")
print(table_top)  # printing the table for the top 10 most visiting countries
```

Σημειώσεις:
-Ο κώδικας είναι πολύ πιο ευανάγνωστος στα αρχεία κώδικα python, ανοιγμένα σε κάποιο IDE (χρησιμοποίησα το IDLE) ή σε κάποιο notepad (χρησιμοποίησα το Notepad++).

-Κατέβασα τα αρχεία Excel μόνο των 4ων τριμήνων, καθώς περιείχαν τα απαραίτητα δεδομένα για τη δημιουργία των γραφημάτων. Στην πρώτη μου προσπάθεια είχα κατεβάσει όλων των τριμήνων.

-Αντιμετώπισα πρόβλημα με το extraction των αριθμών από τα αρχεία Excel, αλλά το διόρθωσα με τη συνάρτηση round(). Επίσης άλλαξα το format των αριθμών έτσι ώστε να είναι πιο ευανάγνωστα, χωρίζοντάς τα με "," ανά τρία ψηφία.

-Για το πρώτο γράφημα έχω πάρει τα δεδομένα για το σύνολο των επισκεπτών στην Ελλάδα ανά έτος.

- Για το τέταρτο γράφημα έχω μετρήσει τις κορυφαίες 10 χώρες που έχουν επισκεφτεί την Ελλάδα συνολικά την τετραετία 2011-2014.

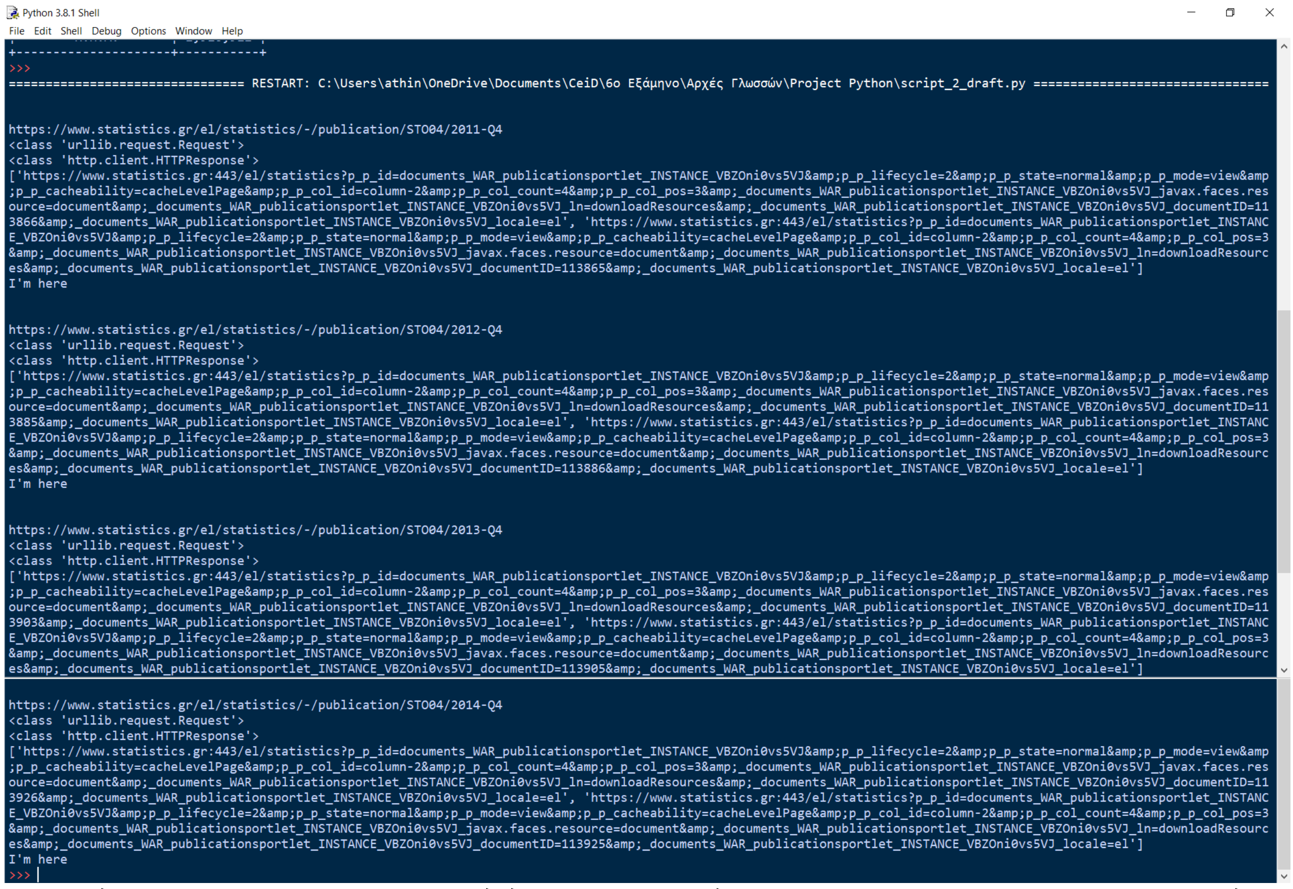
-Για τα διάφορα γραφήματα προσπάθησα να χρησιμοποιήσω διαφορετικούς τύπους, και για την καλύτερη και πιο ξεκάθαρη αναπαράσταση των δεδομένων, αλλά και για να δοκιμάσω ξεχωριστούς τύπους γραφημάτων.

-Οι χρωματικές παλέτες για τα γραφήματα είναι custom-made. Προσπάθησα να επιλέξω χρώματα που δεν είναι κουραστικά για το μάτι.

- Για να δω τη βάση δεδομένων που δημιούργησα χρησιμοποίησα το extension του Firefox:  SQLite Manager.

## 3. Screenshots παραδειγμάτων της εφαρμογής

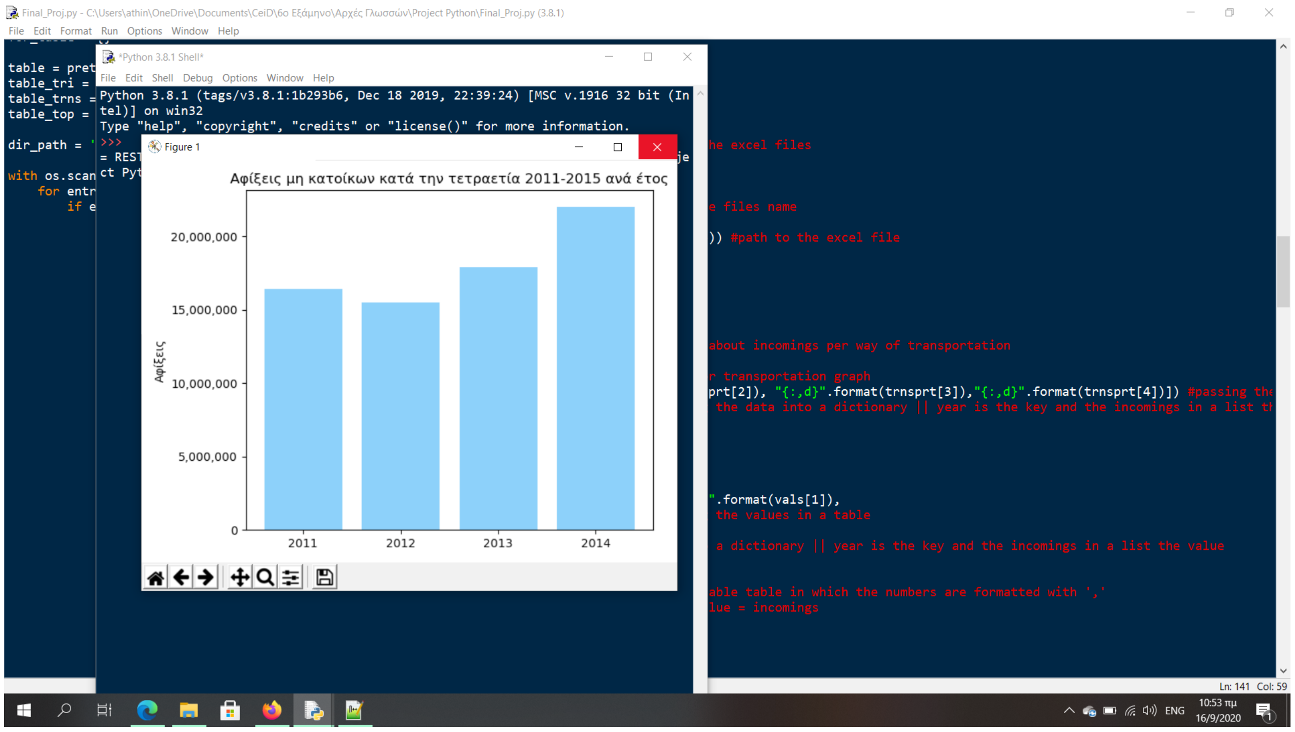
Όταν τρέξω το script για το downloading των αρχείων:



```
+--------------------+-----------+
>>>
============================== RESTART: C:\Users\athin\OneDrive\Documents\CeiD\6o Εξάμηνο\Αρχές Γλωσσών\Project Python\script_2_draft.py ==============================


https://www.statistics.gr/el/statistics/-/publication/STO04/2011-Q4
<class 'urllib.request.Request'>
<class 'http.client.HTTPResponse'>
['https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp
;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.res
ource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=11
3866&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el', 'https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANC
E_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3
&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.resource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResourc
es&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113865&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el']
I'm here


https://www.statistics.gr/el/statistics/-/publication/STO04/2012-Q4
<class 'urllib.request.Request'>
<class 'http.client.HTTPResponse'>
['https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp
;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.res
ource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=11
3885&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el', 'https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANC
E_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3
&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.resource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResourc
es&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113886&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el']
I'm here


https://www.statistics.gr/el/statistics/-/publication/STO04/2013-Q4
<class 'urllib.request.Request'>
<class 'http.client.HTTPResponse'>
['https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp
;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.res
ource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=11
3903&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el', 'https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANC
E_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3
&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.resource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResourc
es&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113905&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el']

https://www.statistics.gr/el/statistics/-/publication/STO04/2014-Q4
<class 'urllib.request.Request'>
<class 'http.client.HTTPResponse'>
['https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp
;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.res
ource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=11
3926&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el', 'https://www.statistics.gr:443/el/statistics?p_p_id=documents_WAR_publicationsportlet_INSTANC
E_VBZOni0vs5VJ&amp;p_p_lifecycle=2&amp;p_p_state=normal&amp;p_p_mode=view&amp;p_p_cacheability=cacheLevelPage&amp;p_p_col_id=column-2&amp;p_p_col_count=4&amp;p_p_col_pos=3
&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.resource=document&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResourc
es&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113925&amp;_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el']
I'm here
>>> |
```

Έχω κάποια prints για προσωπικό έλεγχο και εκτύπωση των download links των αρχείων.

Για το κύριο πρόγραμμα έχω:

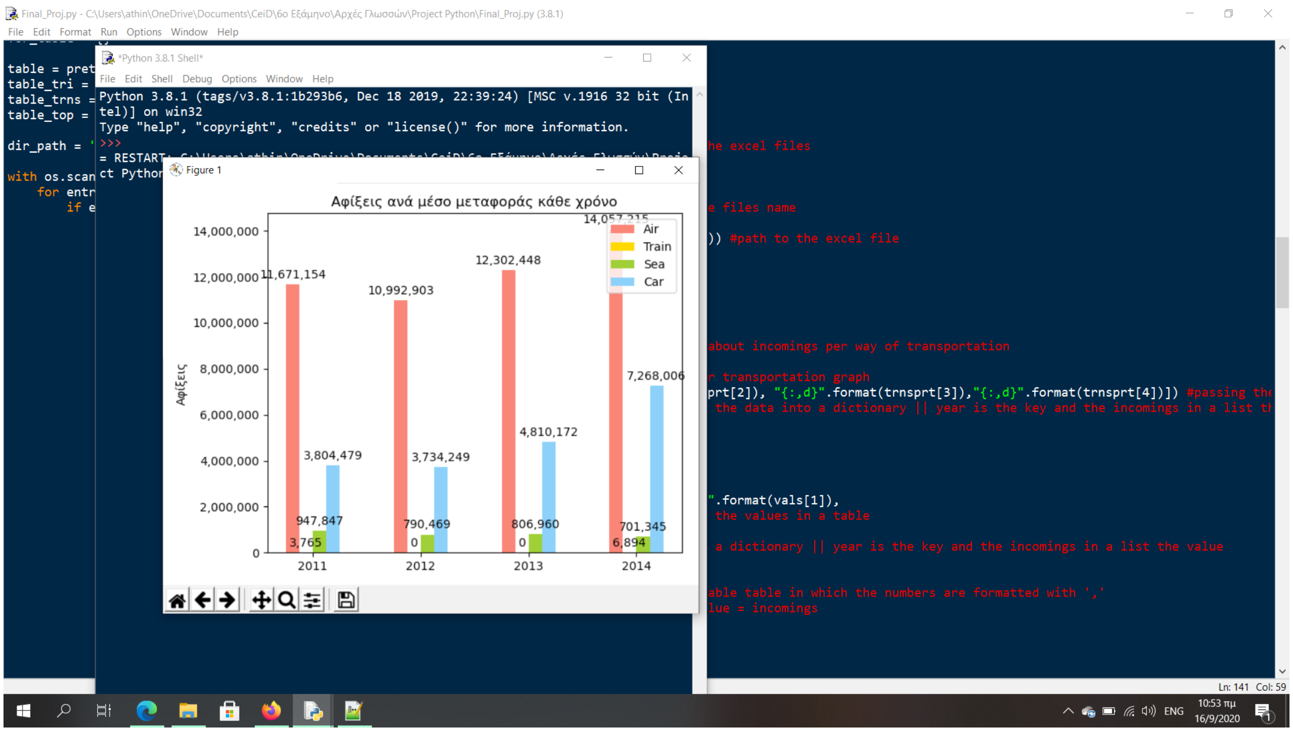


```
table = pret
table_tri =
table_trns =
table_top =

dir_path = '

with os.scan
    for entr
        if e
```

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= REST                                                                      je
ct Pyt
```

```
he excel files

e files name

)) #path to the excel file

about incomings per way of transportation

r transportation graph
prt[2]), "{:,d}".format(trnsprt[3]),"{:,d}".format(trnsprt[4])]) #passing the
 the data into a dictionary || year is the key and the incomings in a list th

".format(vals[1]),
 the values in a table

 a dictionary || year is the key and the incomings in a list the value

able table in which the numbers are formatted with ','
lue = incomings
```

Final_Proj.py - C:\Users\athin\OneDrive\Documents\CeiD\6o Εξάμηνο\Αρχές Γλωσσών\Project Python\Final_Proj.py (3.8.1)
File Edit Format Run Options Window Help
*Python 3.8.1 Shell*
File Edit Shell Debug Options Window Help
Figure 1
Αφίξεις μη κατοίκων κατά την τετραετία 2011-2015 ανά έτος
Αφίξεις
20,000,000
15,000,000
10,000,000
5,000,000
0
2011
2012
2013
2014
Ln: 141 Col: 59
ENG
10:53 πμ
16/9/2020



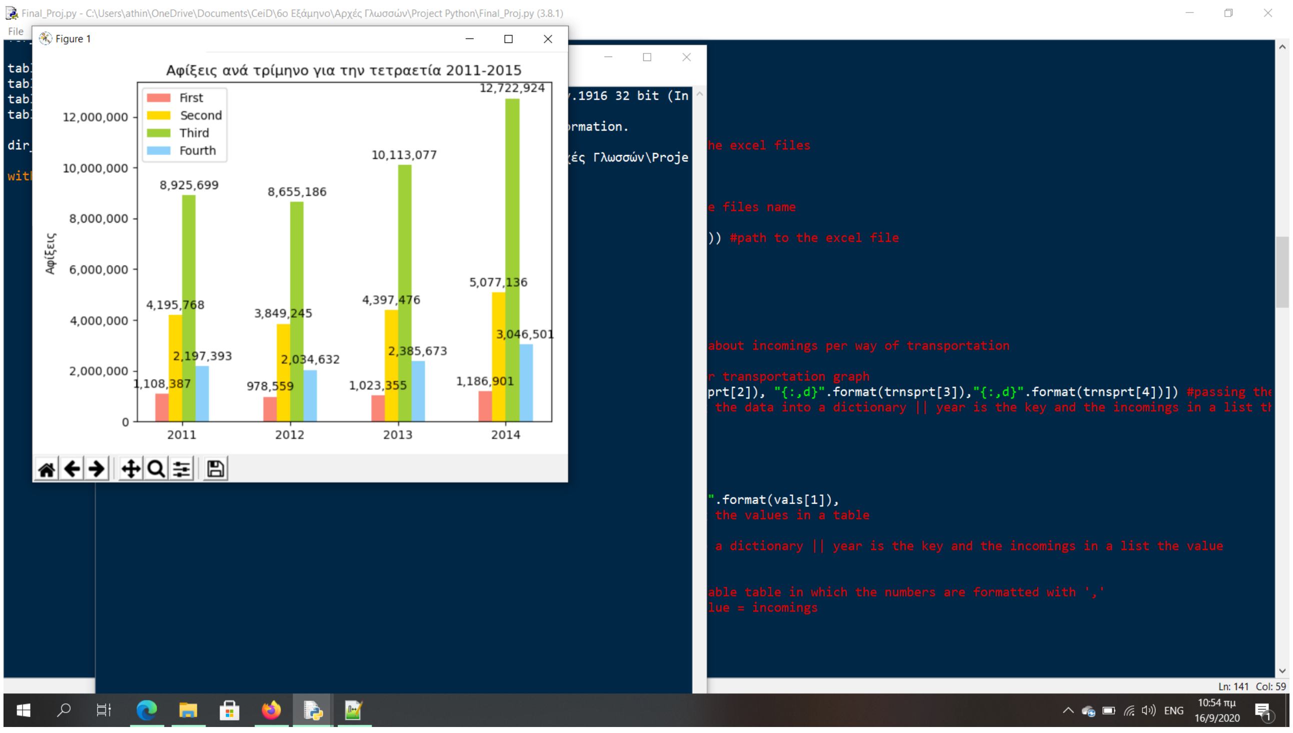
Final_Proj.py - C:\Users\athin\OneDrive\Documents\CeiD\6ο Εξάμηνο\Αρχές Γλωσσών\Project Python\Final_Proj.py (3.8.1)

```
table = pret
table_tri =
table_trns =
table_top =

dir_path = '

with os.scan
    for entr
        if e
```

*Python 3.8.1 Shell*

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\athin\OneDrive\Documents\CeiD\6ο Εξάμηνο\Αρχές Γλωσσών\Proje
ct Python
```

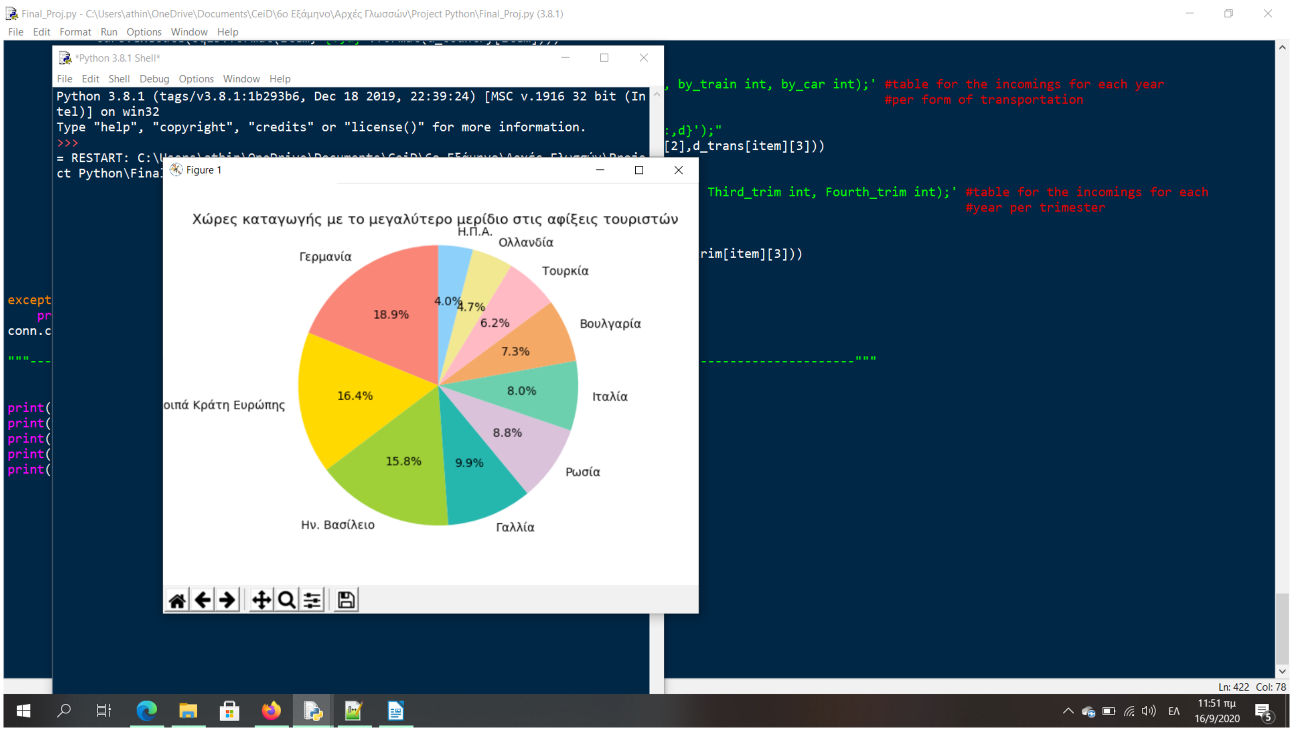
```
he excel files

e files name

)) #path to the excel file

about incomings per way of transportation

r transportation graph
prt[2]), "{:,d}".format(trnsprt[3]),"{:,d}".format(trnsprt[4])]) #passing the
 the data into a dictionary || year is the key and the incomings in a list th

".format(vals[1]),
 the values in a table

 a dictionary || year is the key and the incomings in a list the value

able table in which the numbers are formatted with ','
lue = incomings
```



Final_Proj.py - C:\Users\athin\OneDrive\Documents\CeiD\6ο Εξάμηνο\Αρχές Γλωσσών\Project Python\Final_Proj.py (3.8.1)

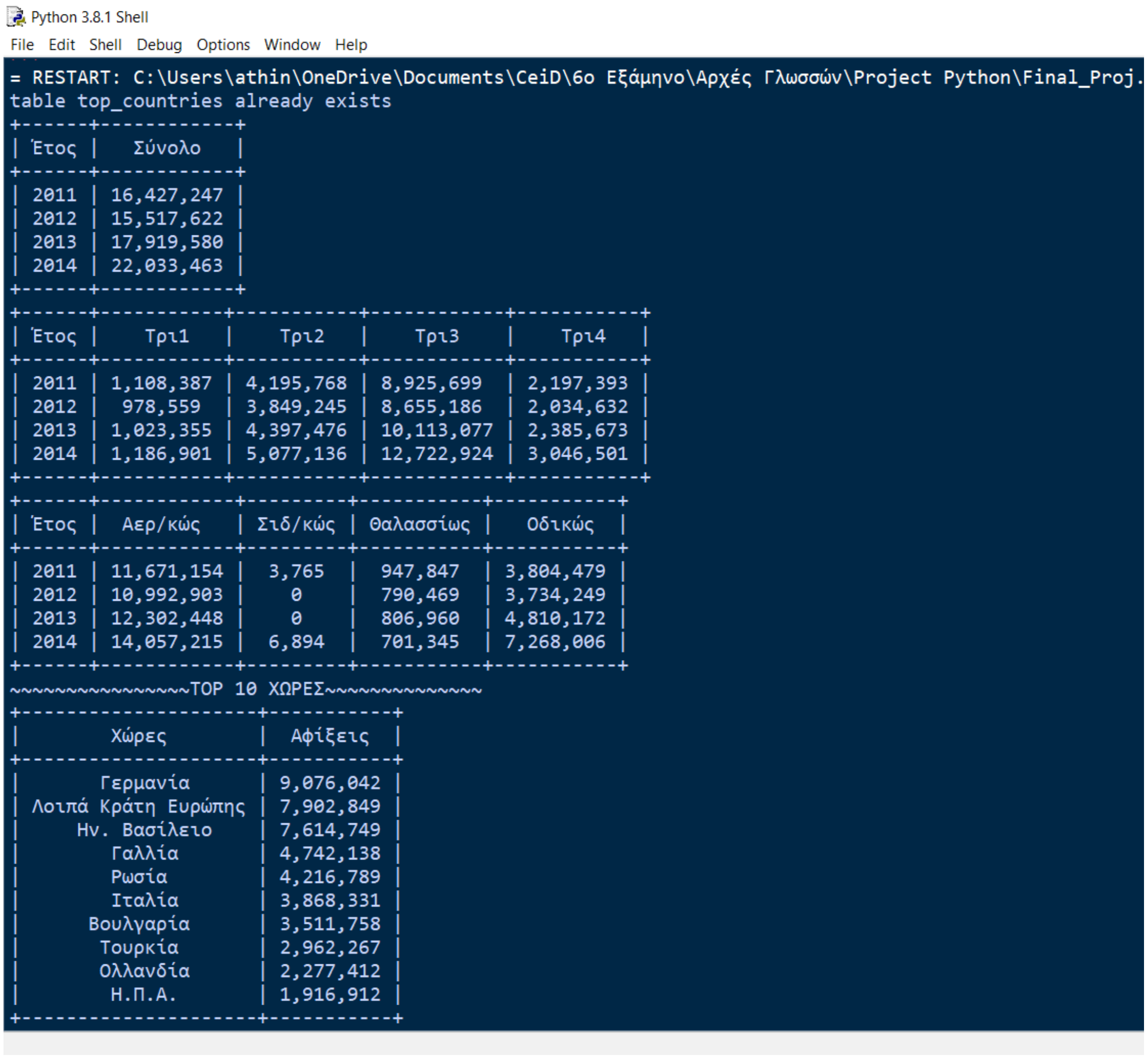
```
.1916 32 bit (In
rmation.
ές Γλωσσών\Proje
```

```
he excel files

e files name

)) #path to the excel file

about incomings per way of transportation

r transportation graph
prt[2]), "{:,d}".format(trnsprt[3]),"{:,d}".format(trnsprt[4])]) #passing the
 the data into a dictionary || year is the key and the incomings in a list th

".format(vals[1]),
 the values in a table

 a dictionary || year is the key and the incomings in a list the value

able table in which the numbers are formatted with ','
lue = incomings
```



```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\athin\OneDrive\Documents\CeiD\6o Εξάμηνο\Αρχές Γλωσσών\Proje
ct Python\Fina
```

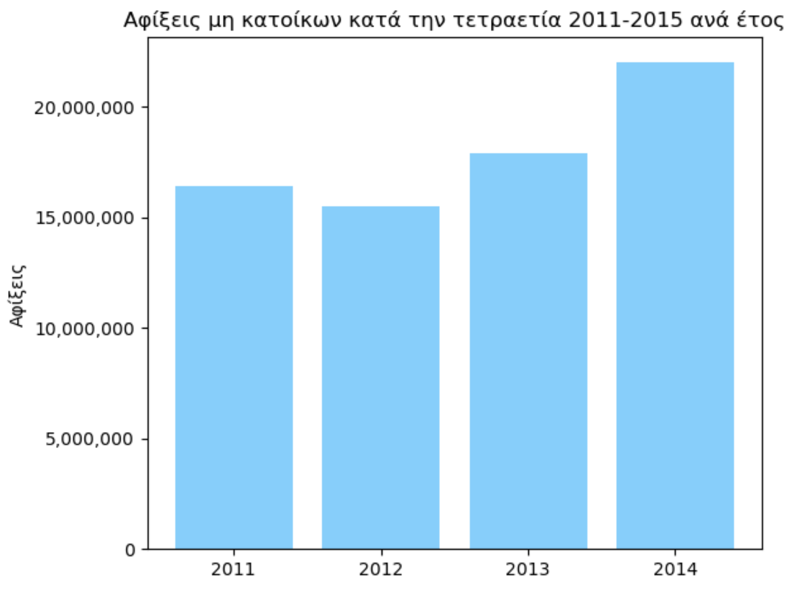
```
, by_train int, by_car int);' #table for the incomings for each year
                               #per form of transportation
:,d}');"
[2],d_trans[item][3]))
 Third_trim int, Fourth_trim int);' #table for the incomings for each
                                    #year per trimester
rim[item][3]))
except
    pr
conn.c
"""---                                    ---------------------"""
print(
print(
print(
print(
print(
```

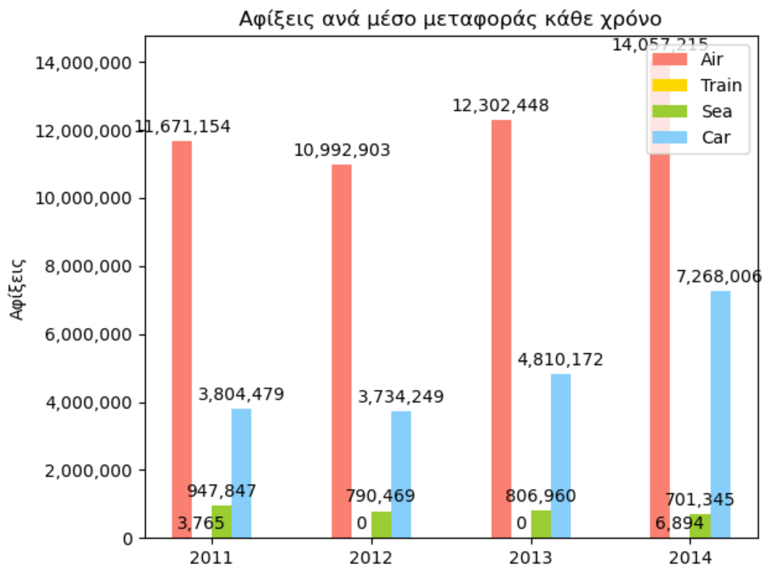


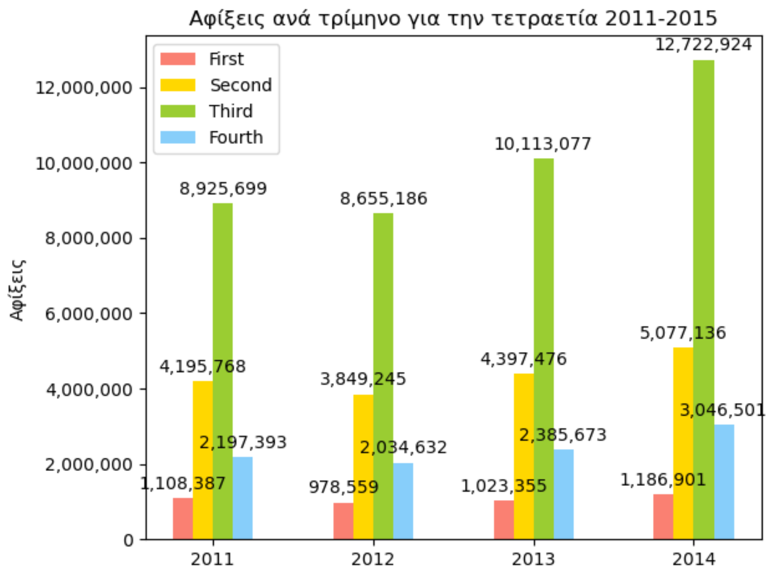
```
= RESTART: C:\Users\athin\OneDrive\Documents\CeiD\6o Εξάμηνο\Αρχές Γλωσσών\Project Python\Final_Proj.
table top_countries already exists
+------+------------+
| Έτος |   Σύνολο   |
+------+------------+
| 2011 | 16,427,247 |
| 2012 | 15,517,622 |
| 2013 | 17,919,580 |
| 2014 | 22,033,463 |
+------+------------+
+------+-----------+-----------+------------+-----------+
| Έτος |    Τρι1   |    Τρι2   |    Τρι3    |    Τρι4   |
+------+-----------+-----------+------------+-----------+
| 2011 | 1,108,387 | 4,195,768 | 8,925,699  | 2,197,393 |
| 2012 |  978,559  | 3,849,245 | 8,655,186  | 2,034,632 |
| 2013 | 1,023,355 | 4,397,476 | 10,113,077 | 2,385,673 |
| 2014 | 1,186,901 | 5,077,136 | 12,722,924 | 3,046,501 |
+------+-----------+-----------+------------+-----------+
+------+------------+---------+------------+-----------+
| Έτος |  Αερ/κώς   | Σιδ/κώς | Θαλασσίως  |   Οδικώς  |
+------+------------+---------+------------+-----------+
| 2011 | 11,671,154 |  3,765  |  947,847   | 3,804,479 |
| 2012 | 10,992,903 |    0    |  790,469   | 3,734,249 |
| 2013 | 12,302,448 |    0    |  806,960   | 4,810,172 |
| 2014 | 14,057,215 |  6,894  |  701,345   | 7,268,006 |
+------+------------+---------+------------+-----------+
~~~~~~~~~~~~~~~~TOP 10 ΧΩΡΕΣ~~~~~~~~~~~~~~
+---------------------+-----------+
|        Χώρες        |  Αφίξεις  |
+---------------------+-----------+
|       Γερμανία      | 9,076,042 |
| Λοιπά Κράτη Ευρώπης | 7,902,849 |
|     Ην. Βασίλειο    | 7,614,749 |
|        Γαλλία       | 4,742,138 |
|        Ρωσία        | 4,216,789 |
|        Ιταλία       | 3,868,331 |
|      Βουλγαρία      | 3,511,758 |
|       Τουρκία       | 2,962,267 |
|       Ολλανδία      | 2,277,412 |
|        Η.Π.Α.       | 1,916,912 |
+---------------------+-----------+
```

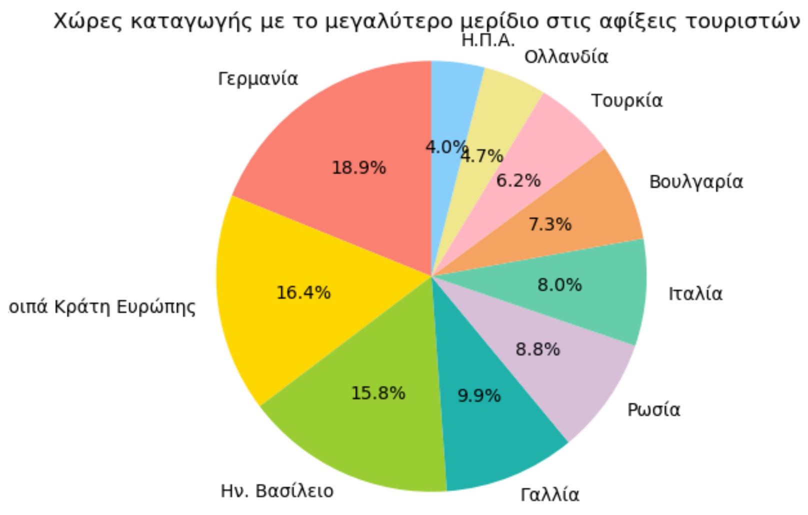
Τέλος έχω κάποιες εκτυπώσεις πινάκων με τα δεδομένα, χρησιμοποιώντας το module prettytable.

## 4. Τα γραφήματα

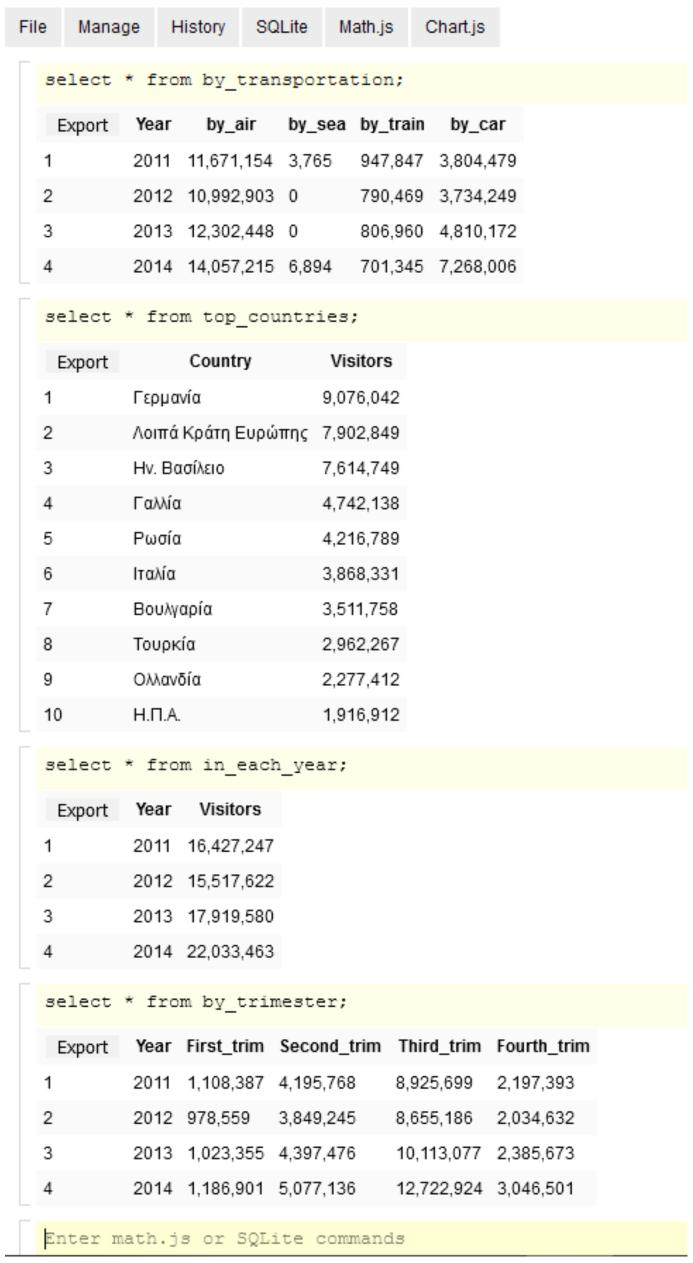








Και για τη βάση δεδομένων (με το SQLite Manager):



```
select * from by_transportation;
```

| Export | Year | by_air | by_sea | by_train | by_car |
|---|---|---|---|---|---|
| 1 | 2011 | 11,671,154 | 3,765 | 947,847 | 3,804,479 |
| 2 | 2012 | 10,992,903 | 0 | 790,469 | 3,734,249 |
| 3 | 2013 | 12,302,448 | 0 | 806,960 | 4,810,172 |
| 4 | 2014 | 14,057,215 | 6,894 | 701,345 | 7,268,006 |

```
select * from top_countries;
```

| Export | Country | Visitors |
|---|---|---|
| 1 | Γερμανία | 9,076,042 |
| 2 | Λοιπά Κράτη Ευρώπης | 7,902,849 |
| 3 | Ην. Βασίλειο | 7,614,749 |
| 4 | Γαλλία | 4,742,138 |
| 5 | Ρωσία | 4,216,789 |
| 6 | Ιταλία | 3,868,331 |
| 7 | Βουλγαρία | 3,511,758 |
| 8 | Τουρκία | 2,962,267 |
| 9 | Ολλανδία | 2,277,412 |
| 10 | Η.Π.Α. | 1,916,912 |

```
select * from in_each_year;
```

| Export | Year | Visitors |
|---|---|---|
| 1 | 2011 | 16,427,247 |
| 2 | 2012 | 15,517,622 |
| 3 | 2013 | 17,919,580 |
| 4 | 2014 | 22,033,463 |

```
select * from by_trimester;
```

| Export | Year | First_trim | Second_trim | Third_trim | Fourth_trim |
|---|---|---|---|---|---|
| 1 | 2011 | 1,108,387 | 4,195,768 | 8,925,699 | 2,197,393 |
| 2 | 2012 | 978,559 | 3,849,245 | 8,655,186 | 2,034,632 |
| 3 | 2013 | 1,023,355 | 4,397,476 | 10,113,077 | 2,385,673 |
| 4 | 2014 | 1,186,901 | 5,077,136 | 12,722,924 | 3,046,501 |

```
Enter math.js or SQLite commands
```