

ΤΟΥΡΙΣΜΟΣ ΣΤΗΝ ΕΛΛΑΔΑ 2011-2014

(ΕΛΣΤΑΤ)

ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΣΤΗ ΓΛΩΣΣΑ PYTHON

Εικόνα που περιέχει ρολόι

Περιγραφή που δημιουργήθηκε αυτόματα

Βαγιανού Εμμανουέλα | ~~~CODE~~~ | [~~~email~~~](mailto:st1059607@ceid.upatras.gr) | 3o έτος

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

16/09/20

Εικόνα που περιέχει φαγητό, άτομα, υπογραφή

Περιγραφή που δημιουργήθηκε αυτόματα

Περιεχόμενα

[Εισαγωγή 2](#_Toc51128697)

[Ο Κώδικας (edit.py) 3](#_Toc51128698)

[Screenshots Παραδειγμάτων Εφαρμογής 11](#_Toc51128699)

[Τα Ζητούμενα Γραφήματα 16](#_Toc51128700)

[Επεξήγηση Της Υλοποίησης 18](#_Toc51128701)

[Δομή Του Κώδικα 18](#_Toc51128702)

[Διευκρινίσεις 18](#_Toc51128703)

# Εισαγωγή

Στην εργασία αυτή καλούμαστε να επεξεργαστούμε δεδομένα για το τουρισμό της Ελλάδας την τετραετία 2011-2014 που θα βρούμε στη σελίδα της [ΕΛΣΤΑΤ](http://www.statistics.gr).

Η υλοποίηση της εργασίας μου αποτελείται από 6 μέρη:

1. Το αρχείο <edit.py> το οποίο περιλαμβάνει τον κώδικα σε γλώσσα python\*
2. Τον φάκελο [downloads](downloads/) στον οποίο αποθηκεύονται τα απαραίτητα αρχεία excel από την σελίδα
3. Τον φάκελο [csv](csv/) στον οποίο αποθηκεύονται τα .csv αρχεία που δημιουργούνται
4. Τον φάκελο [plots](plots/) στον οποίο αποθηκεύονται σε εικόνες τύπου .png τα ζητούμενα γραφήματα της άσκησης
5. Τον φάκελο [screenshots](screenshots/) που περιλαμβάνει χρήσιμα στιγμιότυπα οθόνης που τραβήχτηκαν κατά την εκτέλεση του προγράμματος
6. Την παρούσα αναφορά (.doc + .pdf)

* Στις παρακάτω ενότητες, παρουσιάζονται και αναλύονται όλα τα μέρη της υλοποίησής μου.
* Στην ενότητα με τα στιγμιότυπα οθόνης, η ταξινόμησή τους είναι η σωστή, η σειρά δηλαδή με την οποία εμφανίζονται τα μηνύματα κατά την εκτέλεση του προγράμματος.

*\*Η υλοποίηση έγινε με ένα μόνο αρχείο κώδικα γλώσσας python.*

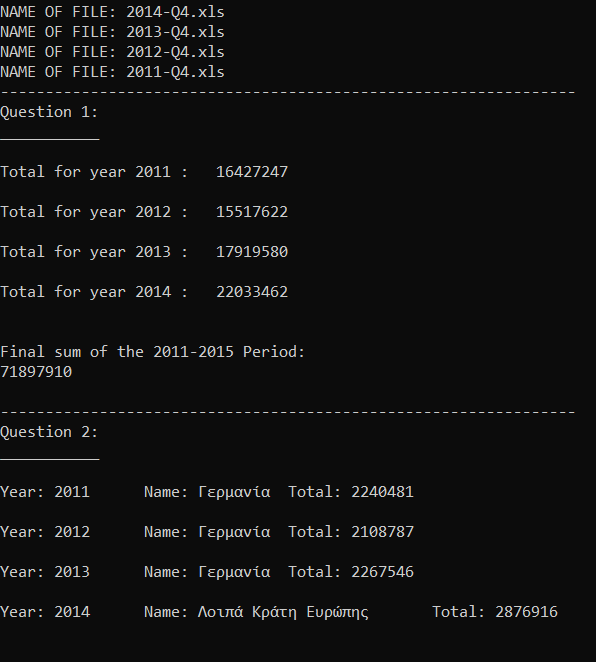
# Ο Κώδικας (edit.py)

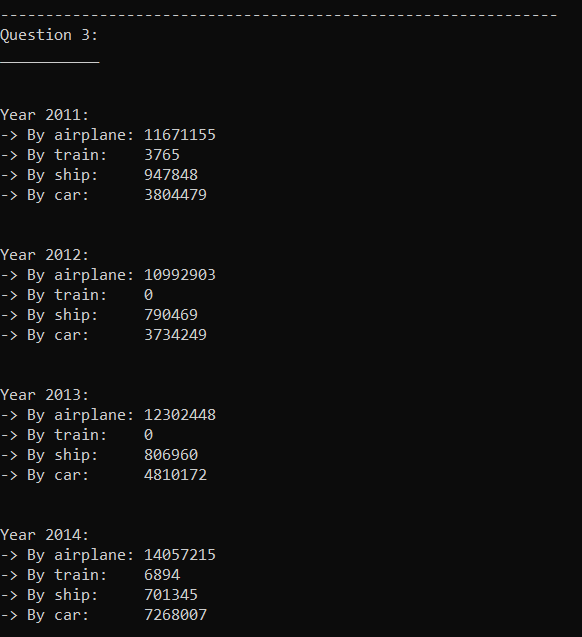
from bs4 import BeautifulSoup  
import requests  
from requests import get  
import xlrd  
import mysql.connector  
import pandas as pd  
import plotly.express as px  
import matplotlib.pyplot as plt  
import csv  
import numpy as np  
from translate import Translator  
  
  
# ----------------------------------------------------------------------------------------  
# DOWNLOAD AND USE NECESSARY FILES BELOW  
# ----------------------------------------------------------------------------------------  
  
# here i download the necessary excel files from the domain  
  
# hold domain name  
domain = "https://www.statistics.gr"  
# hold page to drag html from  
page = requests.get("https://www.statistics.gr/el/statistics/-/publication/STO04/")  
# hold page's html  
html0 = page.content  
# throw parser into html  
soup = BeautifulSoup(html0, 'html.parser')  
  
# find all links in the above html  
for link in soup.find\_all("a"):  
 # get the link from the link tag in html which are stored in the hypertext reference (href) attribute  
 url = link.get('href')  
 # if this part is contained in the href link:  
 # to only download the 4th quarter which holds the info for the whole year:  
 if (("/2014" in url) or ("/2013" in url) or ("/2012" in url) or ("/2011" in url)) and ("Q4" in url):  
 # create file named 2011-Q4.xls for example by using the link's name  
 file\_name = url.split('STO04/')[1]  
 # this way i grap the second part of the splited phrase and then (continue below)  
 # i add to this the .xls to turn it into an excel file name  
 file\_name = (file\_name + ".xls")  
 print("NAME OF FILE: " + file\_name)  
  
 # go to the wanted page  
 newpagename = domain + url   
 #i.e.: https://www.statistics.gr/link\_of\_file\_into\_html\_of\_domain\_page  
 pagaki = requests.get(newpagename) # now i go to the corresponding link to download it  
 # hold pagaki's html following the some procedure as above  
 htmlaki = pagaki.content  
 soupaki = BeautifulSoup(htmlaki, 'html.parser')  
  
 # find the wanted file  
 for coltitle in soupaki.find\_all('td', class\_="titleCol"):  
 # hold url in the same way as before  
 linkaki = coltitle.find('a')  
 urlaki = linkaki.get("href")  
  
 # get the wanted text's link  
 if "ανά χώρα προέλευσης και μέσο μεταφοράς" in linkaki.text:  
 # extract the wanted file into the corresponding file into a folder named downloads  
 with open("downloads/" + file\_name, "wb") as outfile:  
 newpage = urlaki  
 response = get(urlaki)  
 outfile.write(response.content)  
  
 else:  
 continue  
  
 else:  
 continue  
  
  
# --------------------------------------------------------------------------

# here i will read the necessary already downloaded excel files to collect all the info i want  
  
# here just open the 4 necessary files  
# lets try with pandas  
excel1 = 'downloads\\2011-Q4.xls'  
excel2 = 'downloads\\2012-Q4.xls'  
excel3 = 'downloads\\2013-Q4.xls'  
excel4 = 'downloads\\2014-Q4.xls'  
# initiate a dictionary named fck  
fck = {}  
# initiate a table with 0 value  
tbl = 0  
# basically shtnum variable is a counter for the sheets inside the downloaded excel file  
for shtnum in range(12):  
 # i only need the 3d month every time to collent the year's quarters.  
 # So i only need March June September December, so 2 5 8 11, starting from 0  
 if (shtnum == 2) or (shtnum == 5) or (shtnum == 8) or (shtnum == 11):  
 # every sheet has differences so i make sure that nothing is skipped or forgotten that shouldn't  
 if shtnum == 2:  
 skip2 = list(range(0, 71))  
 skip3 = list(range(0, 70))  
 skip4 = list(range(0, 72))  
 if shtnum == 5:  
 skip2 = list(range(0, 71))  
 skip3 = list(range(0, 70))  
 skip4 = list(range(0, 73))  
 if shtnum == 8:  
 skip2 = list(range(0, 71))  
 skip3 = list(range(0, 73))  
 skip4 = list(range(0, 71))  
 if shtnum == 11:  
 skip2 = list(range(0, 73))  
 skip3 = list(range(0, 73))  
 skip4 = list(range(0, 73))  
  
 # for tbl in range(1, 5):  
 # repeate the proccess for every wanted year from 2011 to 2014  
 tbl += 1  
 fck["s2011\_{0}".format(tbl)] = pd.read\_excel(excel1, sheet\_name=shtnum, header=1, index\_col=0, skiprows=range(0, 71))  
 fck["s2011\_{0}".format(tbl)] = fck["s2011\_{0}".format(tbl)].sort\_values(by=['ΣΥΝΟΛΟ'], ascending=False)  
 fck["s2012\_{0}".format(tbl)] = pd.read\_excel(excel2, sheet\_name=shtnum, header=1, index\_col=0, skiprows=skip2)  
 fck["s2012\_{0}".format(tbl)] = fck["s2012\_{0}".format(tbl)].sort\_values(by=['ΣΥΝΟΛΟ'], ascending=False)  
 fck["s2013\_{0}".format(tbl)] = pd.read\_excel(excel3, sheet\_name=shtnum, header=1, index\_col=0, skiprows=skip3)  
 fck["s2013\_{0}".format(tbl)] = fck["s2013\_{0}".format(tbl)].sort\_values(by=['ΣΥΝΟΛΟ'], ascending=False)  
 fck["s2014\_{0}".format(tbl)] = pd.read\_excel(excel4, sheet\_name=shtnum, header=1, index\_col=0, skiprows=skip4)  
 fck["s2014\_{0}".format(tbl)] = fck["s2014\_{0}".format(tbl)].sort\_values(by=['ΣΥΝΟΛΟ'], ascending=False)  
 else:  
 continue  
  
  
# ----------------------------------------------------------------------------------------  
# DATABASE CONNECTION, DATA STORING AND RESULT VERIFICATION BELOW  
# ----------------------------------------------------------------------------------------  
  
# create the database connection  
mydb = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="Manou",  
 db="statistics"  
)  
  
# print (mydb)  
  
# create database if it does not already exists  
mycursor = mydb.cursor()  
mycursor.execute("CREATE DATABASE IF NOT EXISTS statistics")  
  
# use the created database  
mycursor.execute("USE statistics")  
  
# drop tables to avoid dublicate inserts  
mycursor.execute("DROP TABLE IF EXISTS 1\_total\_arrivals")  
mycursor.execute("DROP TABLE IF EXISTS 2\_top\_country")  
mycursor.execute("DROP TABLE IF EXISTS 3\_transportation")  
mycursor.execute("DROP TABLE IF EXISTS 4\_quarter\_arrivals")  
  
# create tables to hold the info for each excel file  
# note: here, the "if not exists" can be emitted but why not?  
mycursor.execute("CREATE TABLE IF NOT EXISTS 1\_total\_arrivals(year INT, total INT)")  
mycursor.execute("CREATE TABLE IF NOT EXISTS 2\_top\_country(year INT, country VARCHAR(20), total INT)")  
mycursor.execute("CREATE TABLE IF NOT EXISTS 3\_transportation(year INT, air INT, train INT, ship INT, car INT)")  
mycursor.execute("CREATE TABLE IF NOT EXISTS 4\_quarter\_arrivals(year INT, q1 INT, q2 INT, q3 INT, q4 INT)")  
  
# ----------------------------------------------------------------------------------------  
# print erwtimata + insert into database  
  
sum = 0  
  
  
# erwthma 1:geniko etisio synolo  
print("----------------------------------------------------------------\nQuestion 1:\n\_\_\_\_\_\_\_\_\_\_\_\n")  
for year in range(1, 5):  
 total = fck["s201{0}\_4".format(year)]['ΣΥΝΟΛΟ'][0]  
 print("Total for year 201" + str(year) + " : \t" + str(round(total)))  
 print("")  
 sum += total  
  
 query1 = """INSERT INTO 1\_total\_arrivals (year, total) VALUES (%s, %s)"""  
 year1 = 2010 + year  
 total1 = int(round(total))  
 # assign value  
 values1 = (year1, total1)  
 # execute query  
 mycursor.execute(query1, values1)  
  
print("\nFinal sum of the 2011-2015 Period:")  
print(int(round(sum)))  
print("")  
  
  
# erwthma 2: xwres me perissoteres afikseis  
print("----------------------------------------------------------------\nQuestion 2:\n\_\_\_\_\_\_\_\_\_\_\_\n")  
for year in range(1, 5):  
 top5 = fck["s201{0}\_4".format(year)].iloc[3:9]  
 # print(top5)  
 # print("")  
# exit()  
 name = top5['Unnamed: 1'][0]  
 total = round(top5['ΣΥΝΟΛΟ'][0])  
 print("Year: 201" + str(year) + "\tName: " + name + "\tTotal: " + str(total) + "\n")  
  
 query2 = """INSERT INTO 2\_top\_country(year, country, total) VALUES (%s, %s, %s)"""  
 year2 = 2010 + year  
 country2 = name  
 total2 = total  
 # assign value  
 values2 = (year2, country2, total2)  
 # execute query  
 mycursor.execute(query2, values2)  
  
  
# erwthma 3: ana meso metaforas  
print("\n--------------------------------------------------------------\nQuestion 3:\n\_\_\_\_\_\_\_\_\_\_\_\n")  
for year in range(1, 5):  
 alltotals = fck["s201{0}\_4".format(year)].iloc[0]  
 ttlair = alltotals[1]  
 ttltrain = alltotals[2]  
 ttlship = alltotals[3]  
 ttlcar = alltotals[4]  
  
 print("\nYear 201" + str(year) + ":")  
 print("-> By airplane: " + str(round(ttlair)))  
 print("-> By train: \t" + str(round(ttltrain)))  
 print("-> By ship: \t" + str(round(ttlship)))  
 print("-> By car: \t" + str(round(ttlcar)))  
 print("")  
  
 query3 = """INSERT INTO 3\_transportation(year, air, train, ship, car) VALUES (%s, %s, %s, %s, %s)"""  
 year3 = 2010 + year  
 air3 = round(ttlair)  
 train3 = round(ttltrain)  
 ship3 = round(ttlship)  
 car3 = round(ttlcar)  
 # assign value  
 values3 = (year3, air3, train3, ship3, car3)  
 # execute query  
 mycursor.execute(query3, values3)  
  
  
# erwthma 4: ana triminia  
print("----------------------------------------------------------------\nQuestion 4:\n\_\_\_\_\_\_\_\_\_\_\_\n")  
for year in range (1, 5):  
 lasttotal = 0  
 print("Year 201" + str(year) + ":")  
 for quarter in range(1, 5):  
 qsum = fck["s201{0}\_{1}".format(year, quarter)]['ΣΥΝΟΛΟ'][0]  
 currenttotal = qsum - lasttotal  
 lasttotal += currenttotal  
 print("-> Total of Quarter number " + str(quarter) + ": \t" + str(round(currenttotal)))  
 if quarter == 1:  
 q1 = round(currenttotal)  
 if quarter == 2:  
 q2 = round(currenttotal)  
 if quarter == 3:  
 q3 = round(currenttotal)  
 if quarter == 4:  
 q4 = round(currenttotal)  
  
 print("")  
 query4 = """INSERT INTO 4\_quarter\_arrivals(year, q1, q2, q3, q4) VALUES (%s, %s, %s, %s, %s)"""  
 year4 = 2010 + year  
 values4 = (year4, q1, q2, q3, q4)  
 mycursor.execute(query4, values4)  
  
# ----------------------------------------------------------------------------------------  
  
# check if everything ok by printing the abore results from the database  
print("\n----------------------------------------------------------------\n--> T A B L E 1 :")  
sql1 = "SELECT \* FROM 1\_total\_arrivals"  
mycursor.execute(sql1)  
table1 = mycursor.fetchall()  
for x in table1:  
 print(x)

print("--> T A B L E 2 :")  
sql2 = "SELECT \* FROM 2\_top\_country"  
mycursor.execute(sql2)  
table2 = mycursor.fetchall()  
for x in table2:  
 print(x)  
  
print("--> T A B L E 3 :")  
sql3 = "SELECT \* FROM 3\_transportation"  
mycursor.execute(sql3)  
table3 = mycursor.fetchall()  
for x in table3:  
 print(x)  
  
print("--> T A B L E 4 :")  
sql4 = "SELECT \* FROM 4\_quarter\_arrivals"  
mycursor.execute(sql4)  
table4 = mycursor.fetchall()  
for x in table4:  
 print(x)  
  
print("\n----------------------------------------------------------------\n")  
# create the csv by using the data in database  
read1 = pd.read\_sql(sql1, mydb)  
df1 = read1.to\_csv('csv/Table1.csv', index=False, header=["YEAR", "TOTAL"])  
read2 = pd.read\_sql(sql2, mydb)  
df2 = read2.to\_csv('csv/Table2.csv', index=False, header=["YEAR", "COUNTRY", "ARRIVALS"])  
read3 = pd.read\_sql(sql3, mydb)  
df3 = read3.to\_csv('csv/Table3.csv', index=False, header=["YEAR", "BY\_AIRPLANE", "BY\_TRAIN", "BY\_SHIP", "BY\_CAR"])  
read4 = pd.read\_sql(sql4, mydb)  
df4 = read4.to\_csv('csv/Table4.csv', index=False, header=["YEAR", "first\_QUARTER", "second\_QUARTER", "third\_QUARTER", "forth\_QUARTER"])  
  
# print csvs to check if all ok  
  
  
# close connection with database  
mydb.close()  
  
  
# ----------------------------------------------------------------------------------------  
# CREATE THE PLOTS BELOW  
# ----------------------------------------------------------------------------------------  
  
# create plot for exercise 1 synola  
print("Figure 1 is getting ready. Please wait...")  
csv1 = pd.read\_csv("csv/Table1.csv")  
x = csv1.YEAR  
y = csv1.TOTAL  
  
X = np.arange(4)  
figure1 = plt.figure("Figure 1")  
ax = figure1.add\_axes([0.1, 0.1, 0.8, 0.8])  
ax.set\_ylabel('Total Arrivals')  
ax.set\_xlabel('Year')  
ax.set\_title('Arrivals for 2011-2015')  
ax.set\_xticks(np.arange(2010, 2015, 1))  
ax.set\_yticks(np.arange(0, 30000000, 1000000))  
ax.bar(x, y)  
for i in range(4):  
 plt.annotate(y[i], xy=(x[i], y[i]), xytext=(x[i]-0.25, y[i]+100000))  
 # 1o orisma: ti tha fainetai sto label / xy : poy deixnei to text / xytext: pou topotheteitai to text  
plt.savefig('plots/exercise1.png')  
plt.show()  
plt.close()  
print("\tFigure 1 COMPLETED\n")  
  
  
# create plot for exercise 2 ana prwth xwra  
print("Figure 2 is getting ready. Please wait...")  
translator = Translator(from\_lang="greek", to\_lang="english")  
csv2 = pd.read\_csv("csv/Table2.csv")  
x = csv2.COUNTRY  
y = csv2.ARRIVALS  
z = csv2.YEAR  
X = np.arange(4)  
figure2 = plt.figure("Figure 2")  
ax = figure2.add\_axes([0.1, 0.1, 0.8, 0.8])  
ax.set\_ylabel('Total Arrivals')  
ax.set\_xlabel('Year')  
ax.set\_title('Top Country for Arrivals in 2011-2015')  
ax.set\_xticks(np.arange(2010, 2015, 1))  
ax.set\_yticks(np.arange(0, 30000000, 100000))  
ax.bar(z, y, color='r')  
for i in range(4):  
 if i == 3:  
 plt.annotate(translator.translate(x[i]), xy=(z[i], y[i]), xytext=(z[i]-0.75, y[i]+10000))  
 else:  
 plt.annotate(translator.translate(x[i]), xy=(z[i], y[i]), xytext=(z[i]-0.25, y[i]+10000))  
plt.savefig('plots/exercise2.png')  
plt.show()  
plt.close()  
print("\tFigure 2 COMPLETED\n")  
  
  
# create plot for exercise 3 ana meso metaforas  
print("Figure 3 is getting ready. Please wait...")  
csv3 = pd.read\_csv("csv/Table3.csv")  
a = csv3.BY\_AIRPLANE  
t = csv3.BY\_TRAIN  
s = csv3.BY\_SHIP  
c = csv3.BY\_CAR  
y = csv3.YEAR  
X = np.arange(4)  
figure3 = plt.figure("Figure 3")  
ax = figure3.add\_axes([0.1, 0.1, 0.8, 0.8])  
ax.set\_ylabel('Total Arrivals')  
ax.set\_xlabel('Year')  
ax.set\_title('Arrivals in 2011-2015 by mean of transport')  
ax.set\_xticks(np.arange(2010, 2015, 1))  
ax.set\_yticks(np.arange(0, 30000000, 1000000))  
ax.bar(y + 0.00, a, color='r', width=0.25)  
ax.bar(y + 0.25, t, color='g', width=0.25)  
ax.bar(y + 0.50, s, color='b', width=0.25)  
ax.bar(y + 0.75, c, color='y', width=0.25)  
for i in range(4):  
 plt.annotate(a[i], xy=(y[i], a[i]), xytext=(y[i]-0.25, a[i]+100000))  
 plt.annotate(t[i], xy=(y[i], t[i]), xytext=(y[i]+0.105, t[i]+100000))  
 plt.annotate(s[i], xy=(y[i], s[i]), xytext=(y[i]+0.25, s[i]+100000))  
 plt.annotate(c[i], xy=(y[i], c[i]), xytext=(y[i]+0.5, c[i]+100000))  
ax.legend(labels=['Airplane', 'Train', 'Ship', 'Car'], loc='upper left', ncol=2)  
plt.savefig('plots/exercise3.png')  
plt.show()  
plt.close()  
print("\tFigure 3 COMPLETED\n")  
  
  
# create plot for exercise 4 ana trimino  
print("Figure 4 is getting ready. Please wait...")  
csv4 = pd.read\_csv("csv/Table4.csv")  
y = csv4.YEAR  
a = csv4.first\_QUARTER  
b = csv4.second\_QUARTER  
c = csv4.third\_QUARTER  
d = csv4.forth\_QUARTER  
X = np.arange(4)  
figure4 = plt.figure("Figure 4")  
ax = figure4.add\_axes([0.1, 0.1, 0.8, 0.8])  
ax.set\_ylabel('Total Arrivals')  
ax.set\_title('Arrivals in Greece during 2011-2015 by Quarters')  
ax.set\_xlabel('Year')  
ax.set\_xticks(np.arange(2010, 2015, 1))  
ax.set\_yticks(np.arange(0, 30000000, 1000000))  
ax.bar(y + 0.00, a, color = 'r', width=0.25)  
ax.bar(y + 0.25, b, color = 'g', width=0.25)  
ax.bar(y + 0.50, c, color = 'b', width=0.25)  
ax.bar(y + 0.75, d, color='y', width=0.25)  
for i in range(4):  
 plt.annotate(a[i], xy=(y[i], a[i]), xytext=(y[i]-0.25, a[i]+100000))  
 plt.annotate(b[i], xy=(y[i], b[i]), xytext=(y[i], b[i]+100000))  
 plt.annotate(c[i], xy=(y[i], c[i]), xytext=(y[i]+0.25, c[i]+100000))  
 plt.annotate(d[i], xy=(y[i], d[i]), xytext=(y[i]+0.5, d[i]+100000))  
ax.legend(labels=['First Q.', 'Second Q.', 'Third Q.', 'Forth Q.'], loc='upper left', ncol=2)  
plt.savefig('plots/exercise4.png')  
plt.show()  
plt.close()  
print("\tFigure 4 COMPLETED")  
  
# --------------------------------------------------------------------------------------------------------------------------  
#lets now print some THANKYOUBYE messages  
  
print("\n----------------------------------------------------------------\n")  
print("""Project completed! You can find everything as described below:  
 -the excel files in path: downloads/  
 -the csv files in path: csv/  
 -the figures in path: plots/  
 """)  
print("----------------------------------------------------------------\n")  
print(">>>>>> Thanks \'n\' byeeeeeeee :)) <<<<<<")

# Screenshots Παραδειγμάτων Εφαρμογής





Εικόνα που περιέχει καθιστός, πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει στιγμιότυπο οθόνης, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει στιγμιότυπο οθόνης

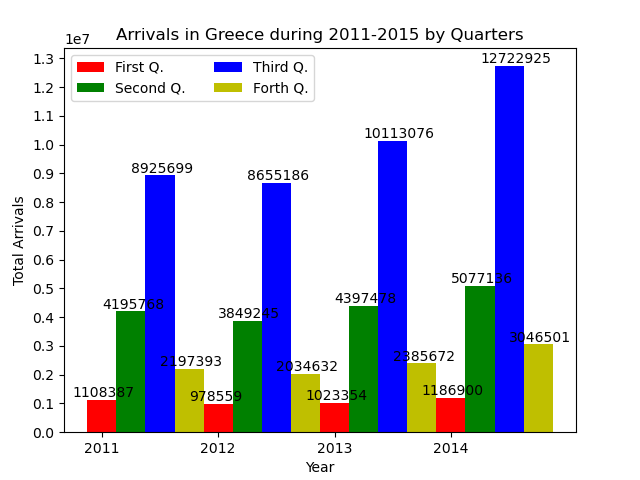
Περιγραφή που δημιουργήθηκε αυτόματα

# Τα Ζητούμενα Γραφήματα

Εικόνα που περιέχει στιγμιότυπο οθόνης, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει στιγμιότυπο οθόνης, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

# Επεξήγηση Της Υλοποίησης

## Δομή Του Κώδικα

Α/Α ΓΡΑΜΜΗ ΔΙΑΔΙΚΑΣΙΑ

1 [1-11] Εισαγωγή απαραίτητων βιβλιοθηκών

2 [20-70] Κατέβασμα των απαραίτητων αρχείων excel από τη σελίδα

3 [90-135] Συλλογή όλων των απαραίτητων στοιχείων από τα κατεβασμένα αρχεία

4 [142-170] Σύνδεση της βάσης δεδομένων και δημιουργία πινάκων, έναν για κάθε ερώτημα

5 [175-271] Αποθήκευση στη βάση δεδομένων τα στοιχεία στους αντίστοιχους πίνακες

6 [277-304] Εμφάνιση όλων των καταχωρήσεων στη βάση, ξεχωριστά για κάθε ερώτημα

7 [305-321] Δημιουργία των .csv με τα απαιτούμενα στοιχεία από τη βάση, ένα για κάθε ερώτημα

8 [328-440] Δημιουργία των τελικών γραφημάτων κάθε ερωτήματος με χρήση των .csv αρχείων

9 [445-452] Εκτύπωση τελευταίων αναφορικών μηνυμάτων

## Διευκρινίσεις

Νο. Δομής Σχόλιο

2 Με χρήση του source code (HTML) της σελίδας [ΕΛΣΤΑΤ](https://www.statistics.gr/el/statistics/-/publication/STO04/) γίνεται αναζήτηση των hypertext references (που περιέχουν τα link) των απαιτούμενων αρχείων. Επιλέγω για κάθε έτος της τετραετίας 2011-2014 να κατεβάσω το 4ο τρίμηνο, διότι περιέχει τα πλήρη στατιστικά του έτους στα διάφορα φύλλα που περιέχει. Από τις 3 διαθέσιμες επιλογές πινάκων που μου παρέχει η σελίδα, επιλέγω την 2η με όνομα «[02. Αφίξεις μη κατοίκων από το εξωτερικό ανά χώρα προέλευσης και μέσο μεταφοράς](https://www.statistics.gr/el/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=3&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_javax.faces.resource=document&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloadResources&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=210437&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el)», διότι περιλαμβάνει χώρα προέλευσης και μέσο μεταφοράς, στοιχεία δηλαδή που χρειάζομαι εξίσου για τη συνέχεια, με βάση τα ζητούμενα της άσκησης.

3 Για τη συλλογή των απαραίτητων στοιχείων, ανοίγω τα κατεβασμένα αρχεία, θέτοντας όμως μερικούς περιορισμούς οι οποίοι προκύπτουν συγκρίνοντας τα περιεχόμενα του κάθε φύλλου που περιέχουν τα αρχεία. Αρχικά, περιορίζω τα φύλλα που πρέπει να χρησιμοποιήσω μόνο σε αυτά που αντιστοιχούν στους τελευταίους μήνες κάθε τριμήνου, δηλαδή Μάρτιος, Ιούνιος, Σεπτέμβριος, Δεκέμβριος, ή αλλιώς, τα φύλλα με αριθμό 2, 5, 8, 11 αν θεωρήσουμε ότι η αρίθμηση ξεκινάει από το 0. Στη συνέχεια, για κάθε ένα από αυτά τα φύλλα, αγνοώ το πρώτο από τα δύο πινακάκια που περιέχονται σε αυτό, αφού δε με ενδιαφέρουν τα μηνιαία στοιχεία, αλλά αυτά του τριμήνου. Τέλος, αφού έχω το τελικό σύνολο στοιχείων που χρειάζομαι, καταχωρώ τα αποτελέσματα σε ένα λεξικό, ταξινομώντας τα ταυτόχρονα ως προς τη τελευταία στήλη, η οποία περιέχει τις συνολικές αφίξεις από κάθε χώρα. Θυμάμαι ότι η τελική μου επιλογή στοιχείων από κάθε ένα από τα φύλλα, περιέχει τις αφίξεις για το διάστημα Ιανουάριο μέχρι τον τρέχοντα μήνα στον οποίο αντιστοιχεί το φύλλο.

Νο. Δομής Σχόλιο

4 Έχω δημιουργήσει μια τοπική βάση δεδομένων στο Workbench στον υπολογιστή μου, την οποία και συνδέω με τα ανάλογα στοιχεία στο πρόγραμμά μου. Με χρήση της MySQL, δημιουργώ τη βάση, καθώς και 4 πίνακες, έναν για κάθε ζητούμενο.

5 Για το 1ο ζητούμενο, βλέπω το πρώτο στοιχείο του τελευταίου φύλλου κάθε έτους, αφού θυμάμαι ότι το φύλλο του Δεκεμβρίου περιέχει τα στοιχεία για το διάστημα Ιανουαρίου-Δεκεμβρίου, όλου δηλαδή του έτους, και ακόμη, ξέρω από το προηγούμενο στάδιο, ότι τα στοιχεία τα έχω ταξινομήσει με φθίνουσα σειρά αφίξεων. Άρα προφανώς το πρώτο στοιχείο θα είναι αυτό με τις περισσότερες αφίξεις. Φυσικά τα αποτελέσματα τα εκτυπώνω για να επιβεβαιώσω την επιλογή μου και τα αποθηκεύω στον αντίστοιχο πίνακα στη βάση.  
Αντίστοιχα για το 2ο ζητούμενο, κοιτάω το πρώτο στοιχείο μόνο που τώρα κρατάω το όνομα της Χώρας, ενώ πριν κρατούσα το πλήθος των αφίξεων.  
Για το 3ο ζητούμενο, με την ίδια λογική κοιτάω τη σειρά που περιέχει τα γενικά σύνολα ανά στήλη και παίρνω τα αντίστοιχα γενικά σύνολα για κάθε μέσο μεταφοράς, εισάγοντάς το και στη βάση.  
Τέλος, στο 4ο ζητούμενο, χρησιμοποιώ τα γενικά σύνολα αφίξεων των 4ων φύλλων που επεξεργάστηκα νωρίτερα από κάθε έτος, και κάθε φορά αφαιρώ από το παρόν σύνολο, αυτό που φύλλου με «την προηγούμενη 3μηνία». Έτσι παίρνω τις συνολικές αφίξεις για τις τριμηνίες όλων των ετών, και φυσικά τις αποθηκεύω στον αντίστοιχο πίνακα της βάσης μου.

6 Εδώ γίνεται μια απλή εκτύπωση τελικών πινάκων της βάσης, για να επιβεβαιώσω ότι τα αποτελέσματα που έχω ταυτίζονται με αυτά που θα έπρεπε.

7 Σε αυτό το κομμάτι, δημιουργώ ένα αρχείο .csv για κάθε ζητούμενο, στο οποίο τοποθετώ τα τελικά στοιχεία από τους αντίστοιχους πίνακες της βάσης, δίνοντας βεβαία τα επιθυμητά ονόματα στηλών.   
Ακόμη, τερματίζω τη σύνδεση με τη βάση μου μιας και δεν θα τη χρειαστώ άλλο, αφού έχω ήδη εξάγει όλες τις απαραίτητες πληροφορίες.

8 Τελευταίο κομμάτι της υλοποίησης είναι η δημιουργία των γραφημάτων. Κάθε γράφημα αντιστοιχεί και σε ένα ζητούμενο. Χρησιμοποιώ για κάθε γράφημα που θέλω να δημιουργήσω, το αντίστοιχο .csv που έχω δημιουργήσει νωρίτερα. Στη συνέχεια καθορίζω τους άξονες, τα ονόματά τους και τις τιμές που θα δείχνουν ή θα κυμαίνονται, αλλά και τον τίτλο του γραφήματος. Κάθε μπάρα αντιπροσωπεύει και ένα «αντικείμενο» του ζητήματος. Δηλαδή, για τα πρώτα 2, κάθε μπάρα αντιστοιχεί σε ένα έτος, ενώ στα επόμενα 2, σε κάθε μέσο μεταφοράς και τρίμηνο αντίστοιχα. Για τον ευκολότερο διαχωρισμό των μπαρών των 2 τελευταίων ζητούμενων έχω χρησιμοποιήσει διαφορετικά χρώματα τα οποία επισημαίνονται στο αντίστοιχο υπόμνημα. Τέλος, στη κορυφή κάθε μπάρας αναγράφεται το επιθυμητό αποτέλεσμα του ζητούμενου ώστε να είναι ξεκάθαρο. Δεν παραλείπω να εκτυπώνω τα κατάλληλα πάντα μηνύματα ώστε να παρακολουθώ την εξέλιξη του προγράμματός μου κάθε στιγμή.

9 Κλείνω τον κώδικά μου με μηνύματα, κάνοντας ξεκάθαρο στο χρήστη το που μπορεί να βρει όλα τα αρχεία, κάνοντας πλοήγηση στο φάκελο του προγράμματος.