



ΤΟΥΡΙΣΜΟΣ ΣΤΗΝ ΕΛΛΑΔΑ 2011-2014 (ΕΛΣΤΑΤ)

ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ
ΣΤΗ ΓΛΩΣΣΑ ΡΥΘΟΝ



Βαγιανού Εμμανουέλα | ~~~CODE~~~ | ~~~email~~~ | 3ο έτος
Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών
16/09/20



Περιεχόμενα

Εισαγωγή.....	2
Ο Κώδικας (edit.py).....	3
Screenshots Παραδειγμάτων Εφαρμογής	11
Τα Ζητούμενα Γραφήματα	16
Επεξήγηση Της Υλοποίησης.....	18
Δομή Του Κώδικα.....	18
Διευκρινίσεις	18

Εισαγωγή

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

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](#) στον οποίο αποθηκεύονται τα απαραίτητα αρχεία excel από την σελίδα
 3. Τον φάκελο [csv](#) στον οποίο αποθηκεύονται τα .csv αρχεία που δημιουργούνται
 4. Τον φάκελο [plots](#) στον οποίο αποθηκεύονται σε εικόνες τύπου .png τα ζητούμενα γραφήματα της άσκησης
 5. Τον φάκελο [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/ST004/")
# 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('ST004/')[1]
        # this way i grap the second part of the splited phrase and then (continue below)
        # i add to this the .xls toturn 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):
# repeat the process 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 duplicate 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 above 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, 1000000))
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]+100000))
    else:
        plt.annotate(translator.translate(x[i]), xy=(z[i], y[i]), xytext=(z[i]-0.25,
y[i]+100000))
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

```
NAME OF FILE: 2014-Q4.xls
NAME OF FILE: 2013-Q4.xls
NAME OF FILE: 2012-Q4.xls
NAME OF FILE: 2011-Q4.xls
-----
Question 1:
-----

Total for year 2011 :    16427247

Total for year 2012 :    15517622

Total for year 2013 :    17919580

Total for year 2014 :    22033462

Final sum of the 2011-2015 Period:
71897910
-----
Question 2:
-----

Year: 2011      Name: Γερμανία  Total: 2240481

Year: 2012      Name: Γερμανία  Total: 2108787

Year: 2013      Name: Γερμανία  Total: 2267546

Year: 2014      Name: Λοιπά Κράτη Ευρώπης      Total: 2876916
```

```
-----
Question 3:
-----

Year 2011:
-> By airplane: 11671155
-> By train:    3765
-> By ship:     947848
-> By car:      3804479

Year 2012:
-> By airplane: 10992903
-> By train:    0
-> By ship:     790469
-> By car:      3734249

Year 2013:
-> By airplane: 12302448
-> By train:    0
-> By ship:     806960
-> By car:      4810172

Year 2014:
-> By airplane: 14057215
-> By train:    6894
-> By ship:     701345
-> By car:      7268007
```

Question 4:

Year 2011:

-> Total of Quarter number 1: 1108387
-> Total of Quarter number 2: 4195768
-> Total of Quarter number 3: 8925699
-> Total of Quarter number 4: 2197393

Year 2012:

-> Total of Quarter number 1: 978559
-> Total of Quarter number 2: 3849245
-> Total of Quarter number 3: 8655186
-> Total of Quarter number 4: 2034632

Year 2013:

-> Total of Quarter number 1: 1023354
-> Total of Quarter number 2: 4397478
-> Total of Quarter number 3: 10113076
-> Total of Quarter number 4: 2385672

Year 2014:

-> Total of Quarter number 1: 1186900
-> Total of Quarter number 2: 5077136
-> Total of Quarter number 3: 12722925
-> Total of Quarter number 4: 3046501

--> T A B L E 1 :

(2011, 16427247)
(2012, 15517622)
(2013, 17919580)
(2014, 22033462)

--> T A B L E 2 :

(2011, 'Γερμανία', 2240481)
(2012, 'Γερμανία', 2108787)
(2013, 'Γερμανία', 2267546)
(2014, 'Λοιπά Κράτη Ευρώπης', 2876916)

--> T A B L E 3 :

(2011, 11671155, 3765, 947848, 3804479)
(2012, 10992903, 0, 790469, 3734249)
(2013, 12302448, 0, 806960, 4810172)
(2014, 14057215, 6894, 701345, 7268007)

--> T A B L E 4 :

(2011, 1108387, 4195768, 8925699, 2197393)
(2012, 978559, 3849245, 8655186, 2034632)
(2013, 1023354, 4397478, 10113076, 2385672)
(2014, 1186900, 5077136, 12722925, 3046501)

Figure 1 is getting ready. Please wait...

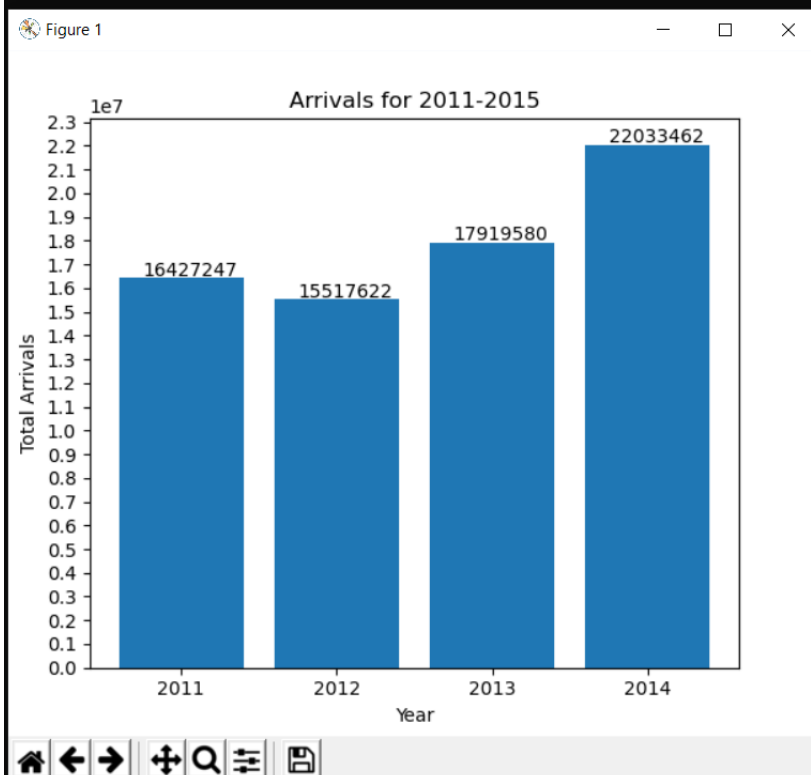


Figure 1 is getting ready. Please wait...

Figure 1 COMPLETED

Figure 2 is getting ready. Please wait...

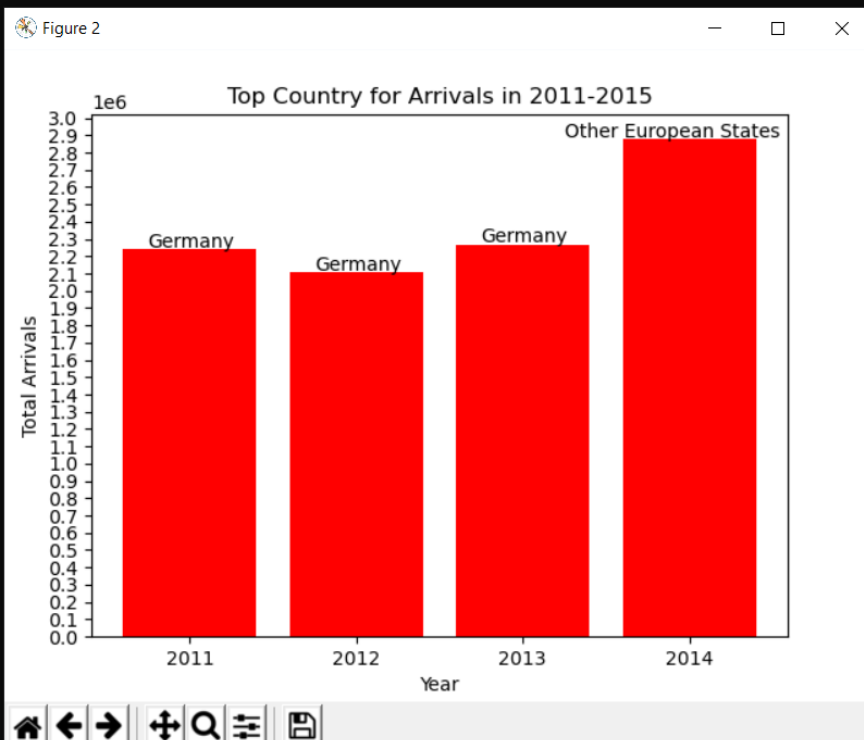


Figure 1 is getting ready. Please wait...

Figure 1 COMPLETED

Figure 2 is getting ready. Please wait...

Figure 2 COMPLETED

Figure 3 is getting ready. Please wait...

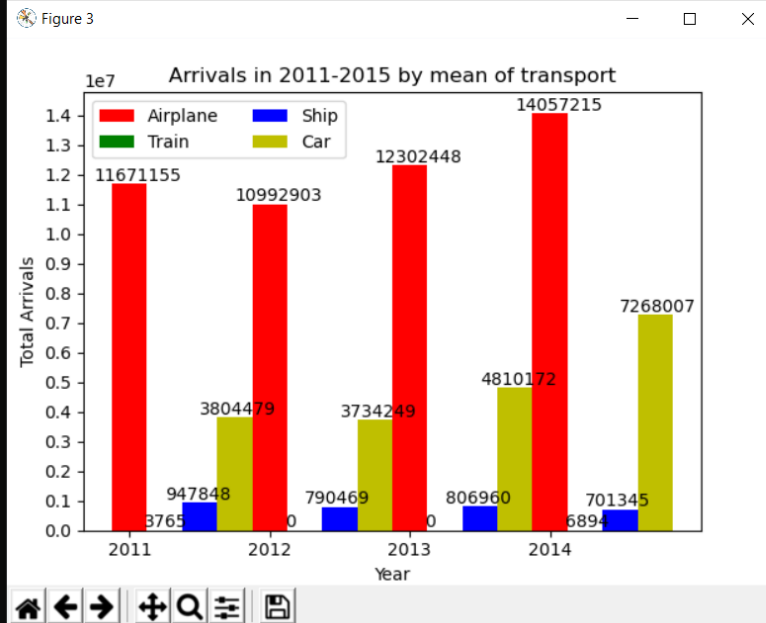


Figure 1 is getting ready. Please wait...

Figure 1 COMPLETED

Figure 2 is getting ready. Please wait...

Figure 2 COMPLETED

Figure 3 is getting ready. Please wait...

Figure 3 COMPLETED

Figure 4 is getting ready. Please wait...

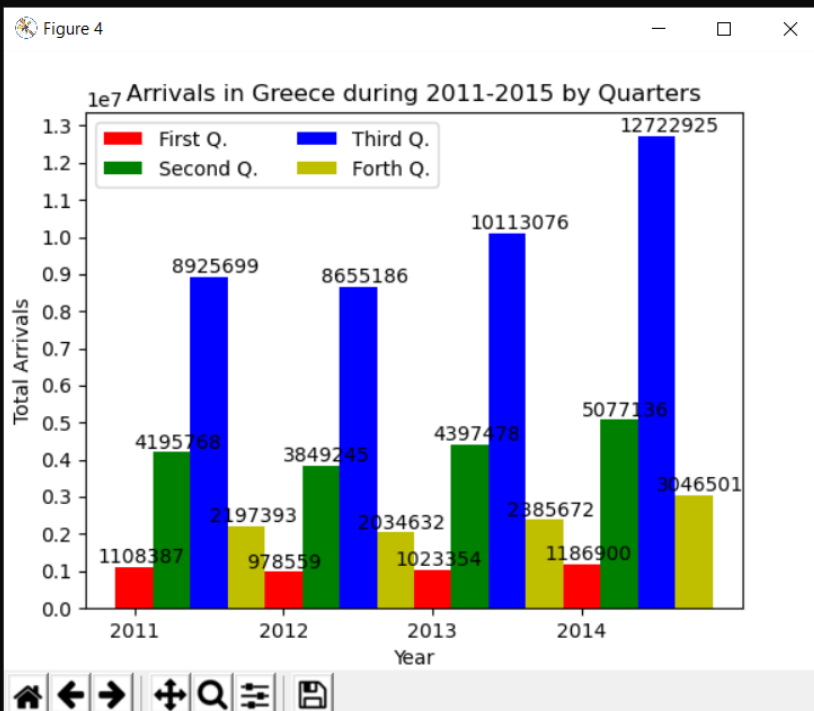




Figure 1 is getting ready. Please wait...
Figure 1 COMPLETED

Figure 2 is getting ready. Please wait...
Figure 2 COMPLETED

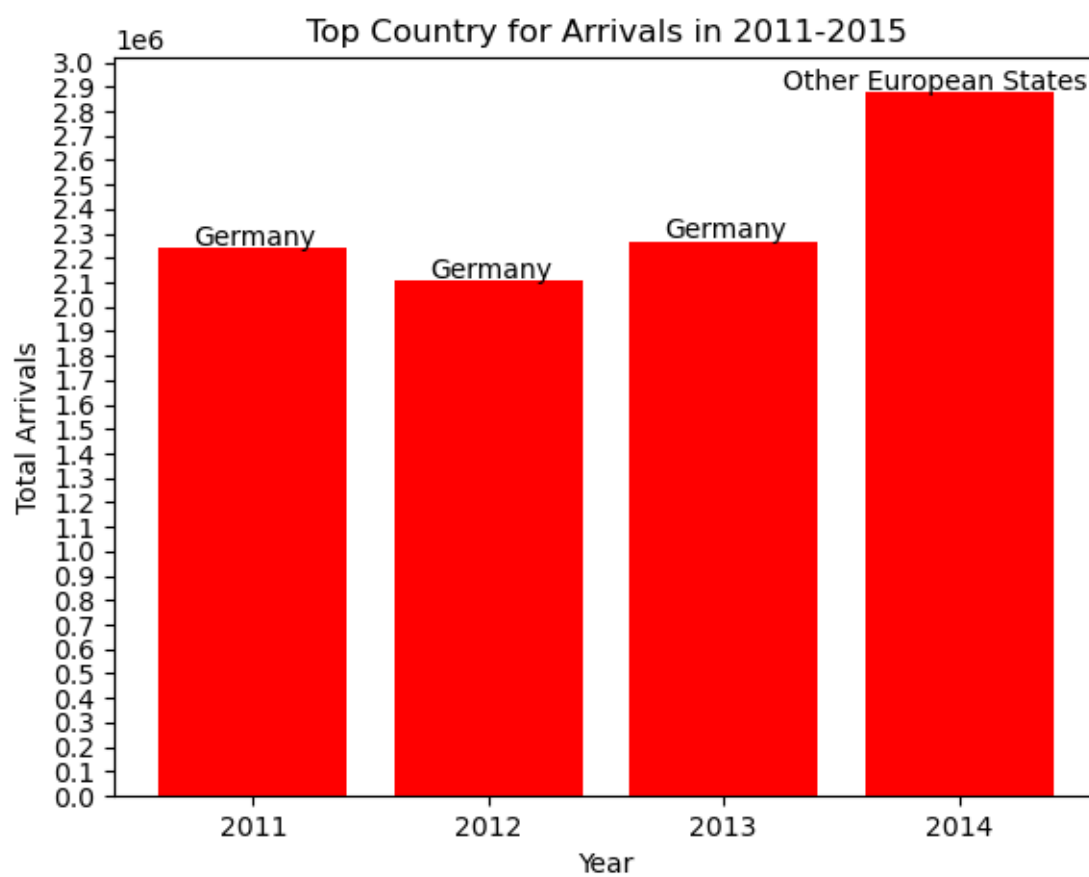
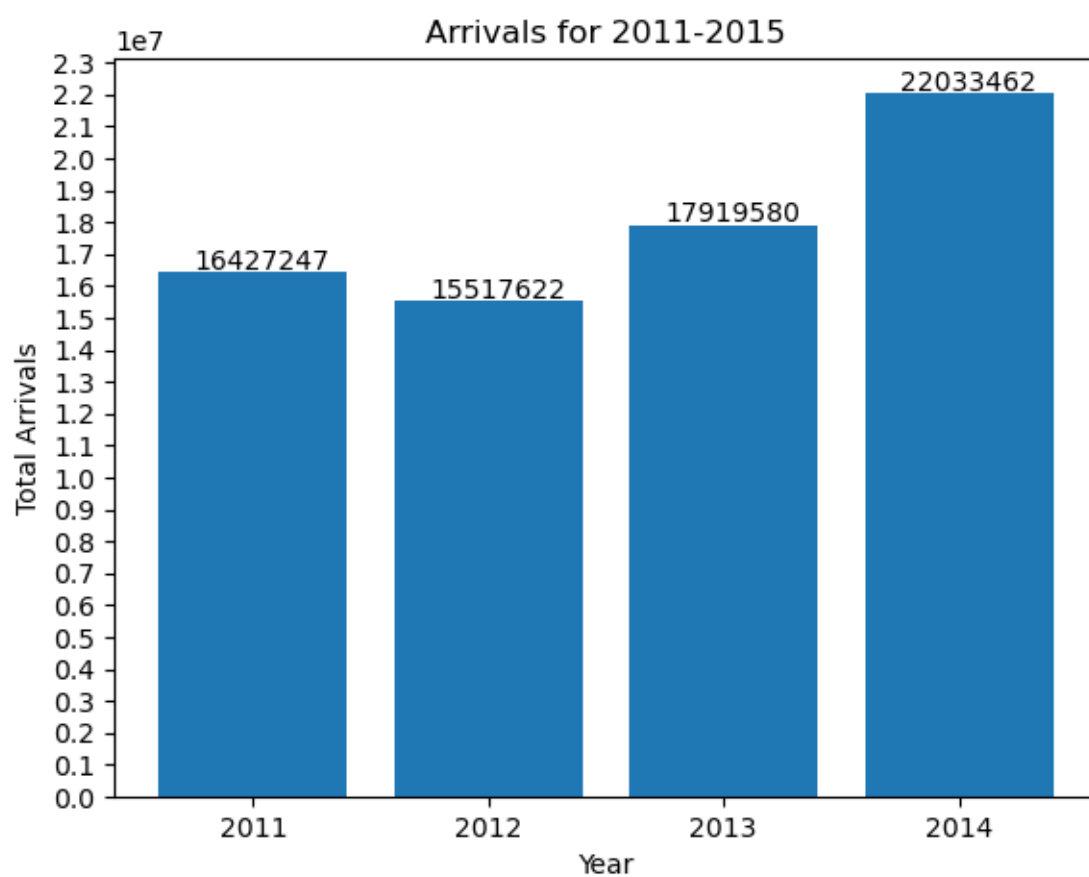
Figure 3 is getting ready. Please wait...
Figure 3 COMPLETED

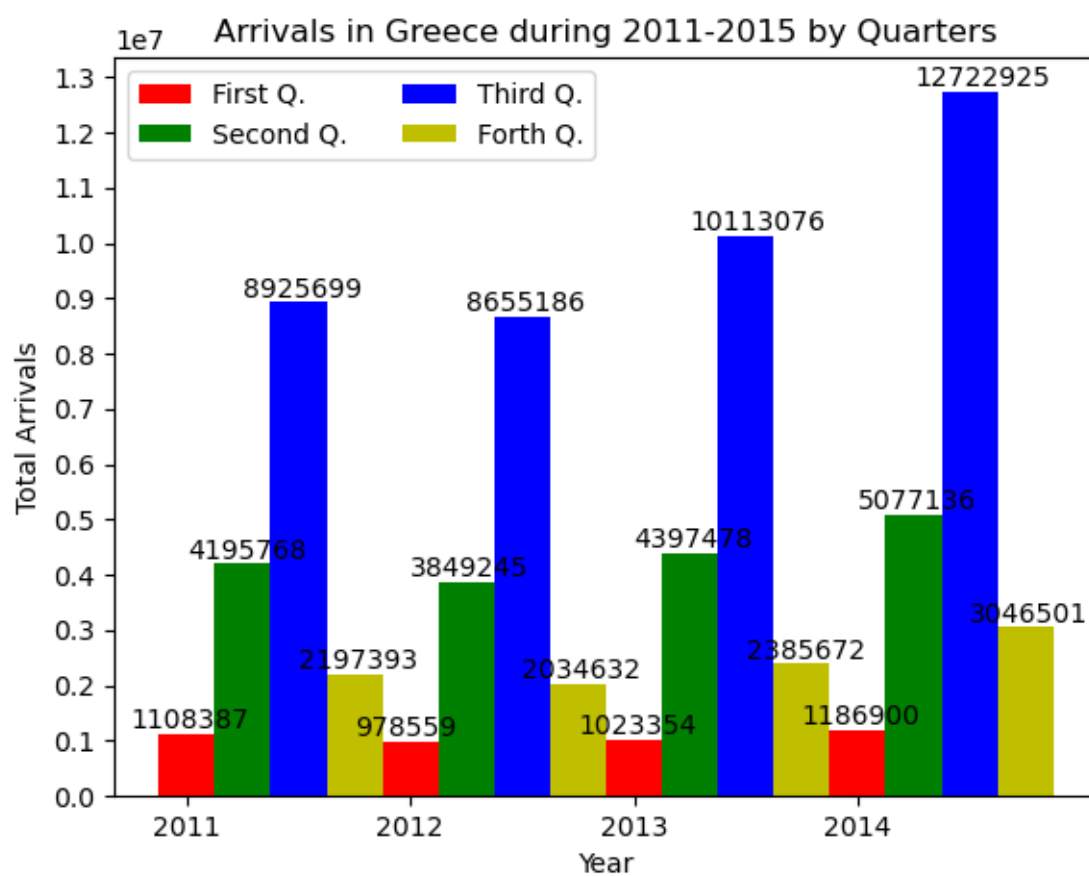
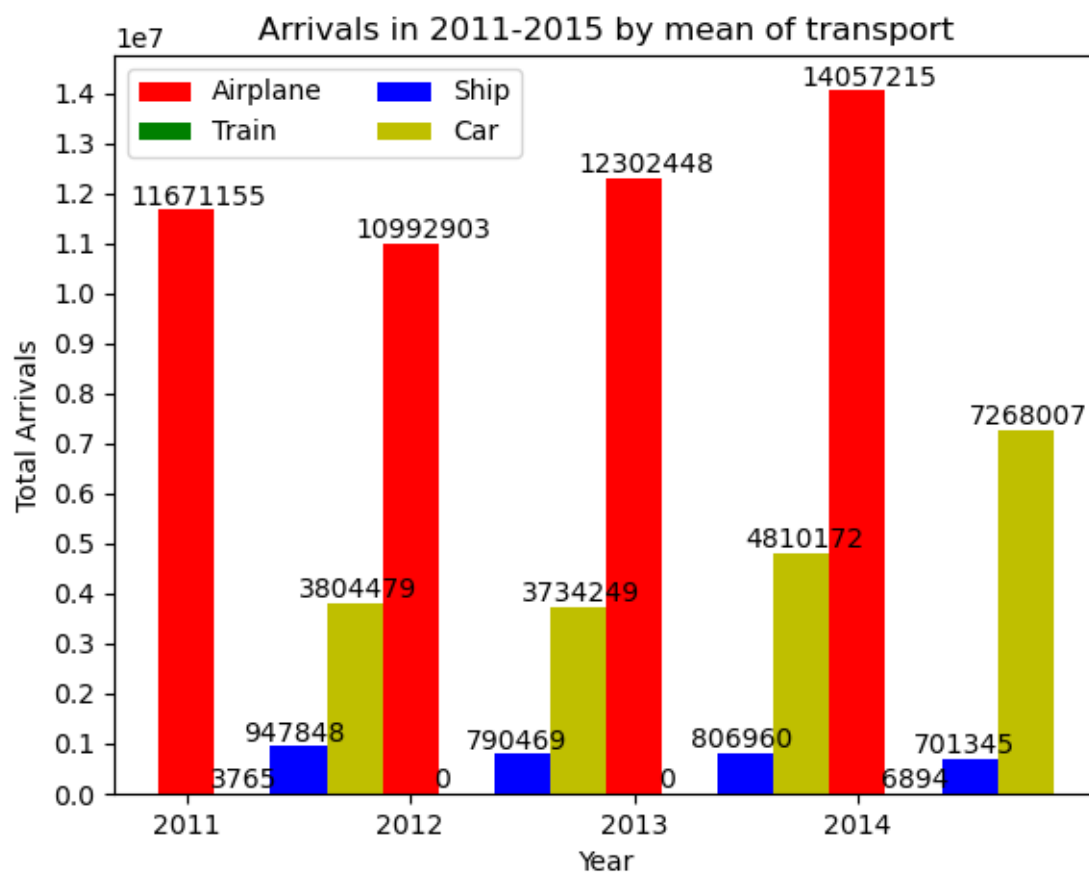
Figure 4 is getting ready. Please wait...
Figure 4 COMPLETED

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/

>>>>> Thanks 'n' byeeeeeeee :)) <<<<<

Τα Ζητούμενα Γραφήματα





Επεξήγηση Της Υλοποίησης

Δομή Του Κώδικα

A/A	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

No. Δομής	Σχόλιο
2	Με χρήση του source code (HTML) της σελίδας ΕΛΣΤΑΤ 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 2 ^η με όνομα « 02. Αφίξεις μη κατοίκων από το εξωτερικό ανά χώρα προέλευσης και μέσο μεταφοράς », διότι περιλαμβάνει χώρα προέλευσης και μέσο μεταφοράς, στοιχεία δηλαδή που χρειαζόμαι 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 χρειαζόμαι, 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.

- 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 1^ο 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 2^ο ζητούμενο, κοιτάω το πρώτο στοιχείο μόνο που τώρα κρατάω το όνομα της Χώρας, ενώ πριν κρατούσα το πλήθος των αφίξεων.
Για το 3^ο 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 4^ο ζητούμενο, χρησιμοποιώ τα γενικά σύνολα αφίξεων των 4^{ων} 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.