

# ΑΝΑΦΟΡΑ

Προαιρετική Εργαστηριακή Άσκηση 2020

Ονοματεπώνυμο: Αριάδνη Μαχιά

ΑΜ: 1059556

Έτος Σπουδών: 3<sup>ο</sup>

Email: st1059556@ceid.upatras.gr



Τμήμα Μηχανικών Η/Υ & Πληροφορικής (CEID)  
Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών

## 1. Κώδικας σε γλώσσα Python εμπλουτισμένος με σχόλια

Σημείωση: Δεν ήμουν σίγουρη σε μερικά υποερωτήματα, εάν η εκφώνηση εννοούσε τις συνολικές αφίξεις και για τα τέσσερα χρόνια συνολικά ή τις συνολικές αφίξεις ξεχωριστά για κάθε χρονιά μέσα στην τετραετία, οπότε έγραψα κώδικα και για τις δύο περιπτώσεις.

Επίσης, η αποθήκευση σε βάση πραγματοποιήθηκε με SQLite.

```
import xlrd
import matplotlib.pyplot as plt
from urllib.request import urlretrieve as retrieve
import numpy as np
import csv
import sqlite3

# === BEGINS DOWNLOAD EXCELS
=====
# Downloads excel file from www.statistics.gr

url = []
url_name = ['statistics_2011.xls', 'statistics_2012.xls', 'statistics_2013.xls',
'statistics_2014.xls']
print("Starting to download")
# 2011
url.append("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=cacheLe
velPage&p_p_col_id=column-
2&p_p_col_count=4&p_p_col_pos=3&documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_java
x.faces.resource=document&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloa
dResources&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113865&_docume
nts_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el")
retrieve(url[0], url_name[0])
# 2012
url.append("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=cacheLe
velPage&p_p_col_id=column-
2&p_p_col_count=4&p_p_col_pos=3&documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_java
x.faces.resource=document&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloa
dResources&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113886&_docume
nts_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el")
retrieve(url[1], url_name[1])
# 2013
url.append("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=cacheLe
velPage&p_p_col_id=column-
2&p_p_col_count=4&p_p_col_pos=3&documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_java
x.faces.resource=document&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloa
dResources&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113905&_docume
nts_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el")
retrieve(url[2], url_name[2])
# 2014
url.append("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=cacheLe
velPage&p_p_col_id=column-
2&p_p_col_count=4&p_p_col_pos=3&documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_java
x.faces.resource=document&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_ln=downloa
```

```

dResources&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_documentID=113925&_documents_WAR_publicationsportlet_INSTANCE_VBZOni0vs5VJ_locale=el")
retrieve(url[3], url_name[3])

print("Complete. Download was Successful")
# === ENDS DOWNLOAD EXCELS
=====

# === BEGINS TOTAL ARRIVALS
=====
# Total tourist for each year in list: total_tourists[0] ->2011, ... , [3]-> 2014
total_tourists = []
# Reading the 4 files
for i in range(0, 4):
    file_location = url_name[i]
    workbook = xlrd.open_workbook(file_location)
    sheet = workbook.sheet_by_index(11) # sheet of December has a second array with total
data of the whole year
    for row in range(134, 137):
        check = sheet.cell_value(row, 1) # if check is True then I found the row where the
data are
        if check == "ΤΕΝΙΚΟ ΣΥΝΟΛΟ": # total sums are in this row
            total_tourists.append(round(sheet.cell_value(row, 6)))
            break
# PRINTS
print("-----TOTAL ARRIVALS-----")
for c in range(0, 4):
    print("Year", c+2011, ":", total_tourists[c])
# === ENDS TOTAL ARRIVALS
=====
=

# === BEGINS TOP 10 COUNTRIES OF ORIGIN
=====
# Countries of origin with the largest share in tourist arrivals
# Top 10 countries for 4-year period (in total) in list: top_countries[0] ->2011, ... , [3]->
2014
# Top 10 countries for each year in list: top_for_each_year[0] ->2011, ... , [3]-> 2014
top_countries = [] # top 10 (arrivals,countries) in 4 years total
top_for_each_year = [] # top 10 (arrivals,countries) for each year
total_countries = [] # total arrivals and country for all countries
country = [] # temp list for taking the arrivals and countries of each excel
for i in range(0, 4):
    file_location = url_name[i]
    workbook = xlrd.open_workbook(file_location)
    sheet = workbook.sheet_by_index(11)
    country.clear()
    c = 0
    j = 0
    for row in range(76, 137): # between these lines is "ΤΕΝΙΚΟ ΣΥΝΟΛΟ" for the whole year
for all of the 4 excels
        if sheet.cell_value(row, 1) == "ΤΕΝΙΚΟ ΣΥΝΟΛΟ":
            break
        if sheet.cell_value(row, 1) != '' and sheet.cell_value(row, 1) != "από τις οποίες:":
# ignores the lines we don't need
            country.append((round(sheet.cell_value(row, 6)), sheet.cell_value(row, 1)))
            if i == 0: # first repetition, we want all the data
                total_countries.append((round(sheet.cell_value(row, 6)),

```

```

sheet.cell_value(row, 1))
    else:
        if country[c][1] == "Κροατία (2)": # Croatia is special case !! it has a
number (for some reason --)
            total_countries.append((round(sheet.cell_value(row, 6)), "Κροατία")) #
gets arrivals and renames the country section
            c = c + 1
            temp = len(total_countries) - 1
            elif country[c][1] == "Κροατία (1)": # handling this: Croatia (1) == Croatia
== Croatia (2)
                if total_countries[temp][1] == "Κροατία":
                    total_countries[temp] = (total_countries[temp][0] + country[c][0],
total_countries[temp][1])
                    c = c + 1
                elif total_countries[j][1] == country[c][1] or country[c][1] == "Σερβία ":
                    total_countries[j] = (total_countries[j][0] + country[c][0],
total_countries[j][1])
                    j = j + 1
                    c = c + 1
                else:
                    total_countries.append((round(sheet.cell_value(row, 6)),
sheet.cell_value(row, 1))
                    c = c + 1
                    j = j + 1
            country.sort(reverse=True) # sorting list 'country' in descending order
            top_for_each_year.append((country[0:10])) # top_for_each_year appends the top 10 country
of i year
# PRINTS
# Prints Lists: total_countries & top_countries & top_for_each_year
print("-----TOTAL COUNTRIES-----")
for c in total_countries:
    print(c)
total_countries.sort(reverse=True)
print("-----TOP 10 COUNTRIES-----")
for c in range(0, 10):
    top_countries.append(total_countries[c])
for c in range(0, 10):
    print(c+1, top_countries[c])
print("-----TOP 10 COUNTRIES EACH YEAR -----")
for year in range(0, 4):
    print("_____ Year ", year+11, "_____")
    for c in range(0, 10):
        print(c+1, top_for_each_year[year][c])

# === ENDS TOP 10 COUNTRIES OF ORIGIN
=====

# === BEGINS MEANS OF TRANSPORT
=====

# Total arrivals for each transportation in list: transportation:
#
transportation[0]-> by plane, [1]-> by rail,
[2]-> by sea, [3]-> by road
transportation = [] # total arrivals for each transportation for 4-year period
transportation_each_year = [] # total arrivals for each year and each transportation
# transportation_each_year -> (arrivals, mean_of_transport, year)
for i in range(0, 4):
    file_location = url_name[i]
    workbook = xlrd.open_workbook(file_location)

```

```

sheet = workbook.sheet_by_index(11)
for row in range(134, 137):
    check = sheet.cell_value(row, 1)
    if check == "ΤΕΝΙΚΟ ΣΥΝΟΛΟ": # between these lines is "ΤΕΝΙΚΟ ΣΥΝΟΛΟ" for the whole
year for all of the 4 excels
        transportation_each_year.append((round(sheet.cell_value(row, 2)), "by plane", i +
2011))
        transportation_each_year.append((round(sheet.cell_value(row, 3)), "by rail", i +
2011))
        transportation_each_year.append((round(sheet.cell_value(row, 4)), "by sea", i +
2011))
        transportation_each_year.append((round(sheet.cell_value(row, 5)), "by road", i +
2011))

    if i == 0: # first repetition, we want the intact data of arrivals
        transportation.append(round(sheet.cell_value(row, 2)))
        transportation.append(round(sheet.cell_value(row, 3)))
        transportation.append(round(sheet.cell_value(row, 4)))
        transportation.append(round(sheet.cell_value(row, 5)))
    else: # rest repetition, we need to add the old arrivals with the new for each
transport
        transportation[0] = transportation[0] + round(sheet.cell_value(row, 2)) # by
plane
        transportation[1] = transportation[1] + round(sheet.cell_value(row, 3)) # by
rail
        transportation[2] = transportation[2] + round(sheet.cell_value(row, 4)) # by
sea
        transportation[3] = transportation[3] + round(sheet.cell_value(row, 5)) # by
road

    break

# PRINTS
print("-----ARRIVALS PER MEAN OF TRANSPORT-----")
for i in range(0, 16, 4):
    print("_____ YEAR ", transportation_each_year[i][2], "_____")
    for x in range(i, i+4):
        print("Transportation: ", transportation_each_year[x][1], "\t| Total arrivals: ",
transportation_each_year[x][0])
print("_____ TOTAL ARRIVALS PER MEAN OF TRANSPORT _____")
for i in range(0, 4):
    if i == 0:
        print("Transportation by plane :", transportation[i])
    elif i == 1:
        print("Transportation by rail :", transportation[i])
    elif i == 2:
        print("Transportation by sea :", transportation[i])
    else:
        print("Transportation by road :", transportation[i])

# === ENDS MEANS OF TRANSPORT
=====

# === BEGINS ARRIVALS PER QUARTER
=====

quarter = [] # will be storing the arrivals for each quarter of each year
# quarter -> ( arrivals, months_of_quarter, year )
k = 0
for i in range(0, 4): # year
    for j in range(2, 12, 3): # quarter of each year (more specific: the last months of each
quarter)
        file_location = url_name[i]

```

```

workbook = xlrd.open_workbook(file_location)
sheet = workbook.sheet_by_index(j)
for row in range(132, 137):
    check = sheet.cell_value(row, 1)
    if check == "ΤΕΝΙΚΟ ΣΥΝΟΛΟ": # between these lines is "ΤΕΝΙΚΟ ΣΥΝΟΛΟ" for the
whole year for all of the 4 excels
        # the last months of each quarter has total data from Jan until this month
        # so in the 2nd, 3rd and 4th quarter we need to subtract the previous quarter
in order get the correct total of each quarter
        if j == 2: # first quarter
            quarter.append((round(sheet.cell_value(row, 6)), "January-February-
March", i+2011))
        elif j == 5: # second quarter
            quarter.append((round(sheet.cell_value(row, 6)) - quarter[k-1][0],
"April-May-June", i + 2011))
        elif j == 8: # third quarter
            quarter.append((round(sheet.cell_value(row, 6)) - quarter[k-1][0] -
quarter[k-2][0], "July-August-September", i + 2011))
        else: # j == 11 # forth quarter
            quarter.append((round(sheet.cell_value(row, 6)) - quarter[k-1][0] -
quarter[k-2][0] - quarter[k-3][0], "October-November-December", i + 2011))
            k = k + 1 # index for list 'quarter' in order to do the subtraction
            break

# PRINTS
print("-----ARRIVALS PER QUARTER-----")
for i in range(0, 16, 4):
    print("_____ YEAR ", quarter[i][2], "_____")
    for x in range(i, i+4):
        print("Total arrivals: ", quarter[x][0], "\t| Quarter: ", quarter[x][1], )
# === ENDS ARRIVALS PER QUARTER
=====

#
~~~~~
~~~~~
#
~~~~~
~~~~~
#
~~~~~
~~~~~

# === BEGINS CHARTS
=====
=====
print("Starting the creation of charts")
# === BEGINS CHARTS FOR TOTAL ARRIVALS
=====

years = ['2011', '2012', '2013', '2014']
arrivals = [] # will contain the number of arrivals
countries = [] # will contain the name of the countries

plt.figure("Total Arrivals")
plt.bar(years, total_tourists, width=0.4) # bar(x, y, width)
plt.xticks(years) # ->in order for the x axis to have the years
plt.xlabel('Έτος', fontsize=12) # label for x axis
plt.ylabel('Αφίξεις Τουριστών', fontsize=12) # label for y axis

```

```

plt.title('Συνολικές Αφίξεις Τουριστών στην Ελλάδα για την τετραετία 2011-2015', fontsize=18)
#chart title
plt.ticklabel_format(style='plain', axis='y') # -> in order to show the numbers (and not
default format, it was not easy on the eye)

# === ENDS CHARTS FOR TOTAL ARRIVALS
=====

# === BEGINS CHARTS FOR TOP 10 COUNTRIES
=====

plt.figure("Top 10 Countries With The Largest Share In Tourist Arrivals In Greece")
for i in range(0, 10):
    arrivals.append(top_countries[i][0])
    countries.append(top_countries[i][1])
plt.barh(countries, arrivals) # horizontal bar in order for the countries to be readable
plt.yticks(countries) # ->in order for the y axis to have the countries
plt.ylabel('Χώρα', fontsize=12)
plt.xlabel('Αφίξεις Τουριστών', fontsize=12)
plt.title('Συνολικές Αφίξεις Τουριστών για την τετραετία 2011-2015\nΑνά τις 10 Χώρες
Καταγωγής με το Μεγαλύτερο Μεριδίο στις Αφίξεις Τουριστών στην Ελλάδα ', fontsize=18)
plt.ticklabel_format(style='plain', axis='x')

# === ENDS CHARTS FOR TOP 10 COUNTRIES
=====

# === BEGINS TOP 10 COUNTRIES FOR EACH YEAR
=====

# these 4 lists will be used also in the SQL section
# These 4 lists will contain the arrivals for each year
arrivals2011 = []
arrivals2012 = []
arrivals2013 = []
arrivals2014 = []
countries.clear()
# the top 10 countries are the same for all the years so I just append for the year 2011
for i in range(0, 10):
    countries.append(top_for_each_year[0][i][1])
# list: top_for_each_year for year 2012-2014 is NOT in the same order as 2011. The countries
are in different order
# That's ^ why I will check to put the numbers in list arrivals2012-14 in the order by the
list: countries
# Otherwise the number will NOT match the correct countries in the graph
# Serbia-Maurobounio ~ Serbia. Must make separate if_statement for this case
for i in range(0, 10):
    for c in range(0, 10):
        if countries[i] == top_for_each_year[0][c][1]:
            arrivals2011.append(top_for_each_year[0][c][0])
            break
    for c in range(0, 10):
        if countries[i] == top_for_each_year[1][c][1]:
            arrivals2012.append(top_for_each_year[1][c][0])
            break
    for c in range(0, 10):
        if i == 6: # i==6 -> Σερβία-Μαυροβούνια ~ Σερβία c=7
            arrivals2013.append(top_for_each_year[1][7][0])
            break
        elif countries[i] == top_for_each_year[2][c][1]:

```

```

        arrivals2013.append(top_for_each_year[2][c][0])
        break
    for c in range(0, 10):
        if i == 6: # i==6 -> Σερβία-Μαυροβούνια ~ Σερβία c=7
            arrivals2014.append(top_for_each_year[1][7][0])
            break
        elif countries[i] == top_for_each_year[3][c][1]:
            arrivals2014.append(top_for_each_year[3][c][0])
            break

# Making one chart with the data of all 10 countries of each year
fig1, ax1 = plt.subplots()
ypos = np.arange(10) # ->needed in order to arrange the position of the bars for each
country
ax1.set_yticks(np.arange(len(countries)))
ax1.set_yticklabels(countries) # ->in order for the y axis to have the countries
ax1.set_yticklabels(countries, fontsize=12)
ax1.ticklabel_format(style='plain', axis='x')
fig1.canvas.set_window_title('Total Arrival Per Country and Per Year')
ax1.set_xlabel('Αφίξεις Τουριστών', fontsize=12)
ax1.set_ylabel('Χώρα', fontsize=12)
plt.title('Συνολικές Αφίξεις Τουριστών\nΑνά Έτος & Ανά τις 10 Χώρες Καταγωγής με το
Μεγαλύτερο Μεριδίο στις Αφίξεις Τουριστών στην Ελλάδα', fontsize=18)
ax1.barh(ypos+0.00, arrivals2011, height=0.1, label='2011') # (position from beginning of
column, data, height, label)
ax1.barh(ypos+0.25, arrivals2012, height=0.1, label='2012')
ax1.barh(ypos+0.50, arrivals2013, height=0.1, label='2013')
ax1.barh(ypos+0.75, arrivals2014, height=0.1, label='2014')
ax1.legend() # in order to show labels of each bar colour

# === ENDS TOP 10 COUNTRIES FOR EACH YEAR
=====

# === BEGINS CHARTS FOR TRANSPORTATION
=====
# same logic as the countries chart
# !!! BY RAIL IS TOO SMALL IN COMPARISON TO THE OTHERS. IT EXIST BUT YOU NEED TO ZOOM IN
!!!!!!!!!
plt.figure("Total Arrival For Each Mean Of Transport")
plt.bar(["Αεροπορικός", "Σιδη/κός", "Θαλασσίως", "Οδικώς"], transportation, width=0.4)
plt.xticks(["Αεροπορικός", "Σιδη/κός", "Θαλασσίως", "Οδικώς"])
plt.xlabel('Μέσο Μεταφοράς', fontsize=12)
plt.ylabel('Αφίξεις Τουριστών', fontsize=12)
plt.title('Για την τετραετία 2011-2015\nΑνά Μέσο Μεταφοράς οι Συνολικές Αφίξεις Τουριστών
στην Ελλάδα', fontsize=18)
plt.ticklabel_format(style='plain', axis='y')

# === ENDS CHARTS FOR TRANSPORTATION
=====

# === BEGINS CHARTS FOR TRANSPORTATION
=====
# same logic as the countries chart
# !!! BY RAIL IS TOO SMALL IN COMPARISON TO THE OTHERS. IT EXIST BUT YOU NEED TO ZOOM IN
!!!!!!!!!
fig2, ax2 = plt.subplots()
xpos = np.arange(4)

```



```

ax2.set_xticks(np.arange(len(years))
ax2.set_xticklabels(years)
ax2.set_xticklabels(years, fontsize=12)
ax2.ticklabel_format(style='plain', axis='y')
ax2.set_xlabel('Έτος', fontsize=12)
ax2.set_ylabel('Αφίξεις Τουριστών', fontsize=12)
plt.title('Συνολικές Αφίξεις Τουριστών στην Ελλάδα\nΑνά Μέσο Μεταφοράς και Ανά Έτος',
          fontsize=18)
fig2.canvas.set_window_title('Total Arrival For Each Mean Of Transport Per Year')
ax2.bar(xpos+0.00, transportation_each_year[0][0], width=0.15, label='Αεροπορικώς')
ax2.bar(xpos+0.25, transportation_each_year[1][0], width=0.15, label='Σιδηρώς')
ax2.bar(xpos+0.50, transportation_each_year[2][0], width=0.15, label='Θαλασσίνως')
ax2.bar(xpos+0.75, transportation_each_year[3][0], width=0.15, label='Οδικώς')
ax2.legend()

# === ENDS CHARTS FOR TRANSPORTATION
=====

# === BEGINS CHARTS FOR QUARTERS
=====

# same logic as the countries chart
# these 4 lists will be used also in the SQL section
# These 4 lists will contain the arrivals of all(4) quarter for each year
q2011 = []
q2012 = []
q2013 = []
q2014 = []
for i in range(0, 4):
    q2011.append(quarter[i][0])
for i in range(4, 8):
    q2012.append(quarter[i][0])
for i in range(8, 12):
    q2013.append(quarter[i][0])
for i in range(12, 16):
    q2014.append(quarter[i][0])

quarter_name = ['Ιαν-Φεβ-Μαρτ', 'Απρ-Μαι-Ιουν', 'Ιουλ-Αυγ-Σεπτ', 'Οκτ-Νοεμ-Δεκ']
fig, ax = plt.subplots()
xpos = np.arange(4)
ax.set_xticks(np.arange(len(quarter_name)))
ax.set_xticklabels(quarter_name)
ax.set_xticklabels(quarter_name, fontsize=12)
ax.ticklabel_format(style='plain', axis='y')
ax.set_xlabel('Τρίμηνο', fontsize=12)
ax.set_ylabel('Αφίξεις Τουριστών', fontsize=12)
plt.title('Για την τετραετία 2011-2015\nΑνά Τρίμηνο οι Συνολικές Αφίξεις Τουριστών στην Ελλάδα',
          fontsize=18)
fig.canvas.set_window_title('Total Arrival For Each Quarter Per Year')
ax.bar(xpos+0.00, q2011, width=0.15, label='2011')
ax.bar(xpos+0.25, q2012, width=0.15, label='2012')
ax.bar(xpos+0.50, q2013, width=0.15, label='2013')
ax.bar(xpos+0.75, q2014, width=0.15, label='2014')
ax.legend()
# === ENDS CHARTS FOR QUARTERS
=====

print("Showing all the charts")
plt.show() # Shows all the charts
# === ENDS CHARTS

```

```

=====
=====

#
~~~~~
~~~~~
#
~~~~~
~~~~~
#
~~~~~
~~~~~

# === BEGINS - CSV -
=====
=====
print("Starting the creation of .csv file")

# === BEGINS - CSV - TOTAL ARRIVALS
=====
year = ['2011', '2012', '2013', '2014']
with open('total_tourists.csv', 'w', newline='') as f: # opens (and creates) .csv file
    writer = csv.writer(f)
    writer.writerow(['Συνολικές Αφίξεις τουριστών στην Ελλάδα για την τετραετία 2011-2015'])
# row with the title
    writer.writerow(year) # column headers
    writer.writerow(total_tourists) # row data
    f.close()

# === BEGINS - CSV - Top 10 Countries 2011-2014
=====
with open('top_10_countries.csv', 'w', newline='') as f2: # opens (and creates) .csv file
    writer = csv.writer(f2)
    arr = [] # will contain arrivals
    coun = [] # will countain countries
    for i in range(0, 10):
        arr.append(top_countries[i][0]) # use list 'top_countries' from the first section of
the code
        coun.append(top_countries[i][1])
    writer.writerow(['10 χώρες καταγωγής με το μεγαλύτερο μερίδιο στις αφίξεις τουριστών στην
Ελλάδα για την τετραετία 2011-2015'])
    writer.writerow(coun)
    writer.writerow(arr)
    f2.close()

# === BEGINS - CSV - Top 10 Countries EACH YEAR
=====
with open('top_10_countries_each_year.csv', 'w', newline='') as f3: # opens (and creates)
.csv file
    writer = csv.writer(f3)
# === 2011 ===
    arr2011 = [] # will contain arrivals for 2011
    coun = [] # will contain the correct top 10 countries minus the added 'Ετος' from
previous section of the code
    coun.append('Ετος')
    for i in range(0, 10):
        coun.append(countries[i]) # use list 'countries' from chart | contains the correct

```

```

top 10 countries
arr2011.append(2011)
for i in range(0, 10):
    arr2011.append(arrivals2011[i])
writer.writerow(['Ανά Έτος οι 10 χώρες καταγωγής με το μεγαλύτερο μερίδιο στις αφίξεις
τουριστών στην Ελλάδα (σε φθίνουσα σειρά)'])
writer.writerow(coun) # column headers
writer.writerow(arr2011) # add row data
# === 2012 ===
arr2012 = [] # will contain arrivals for 2012
arr2012.append(2012)
for i in range(0, 10):
    arr2012.append(arrivals2012[i])
writer.writerow(arr2012) # add row data
# === 2013 ===
arr2013 = [] # will contain arrivals for 2013
arr2013.append(2013)
for i in range(0, 10):
    arr2013.append(arrivals2013[i])
writer.writerow(arr2013) # add row data
# === 2014 ===
arr2014 = [] # will contain arrivals for 2014
arr2014.append(2014)
for i in range(0, 10):
    arr2014.append(arrivals2014[i])
writer.writerow(arr2014) # add row data
f3.close()

# === BEGINS - CSV - MEAN OF TRANSPORT
=====
with open('means_of_transport.csv', 'w', newline='') as f4: # opens (and creates) .csv file
    writer = csv.writer(f4)
    writer.writerow(['Αφίξεις τουριστών στην Ελλάδα Ανά Μέσο Μεταφοράς για την τετραετία
2011-2015']) # Title
    writer.writerow(['Αεροπορικώς', 'Σιδηρώς', 'Θαλασσίως', 'Οδικώς']) # column headers
    writer.writerow(transportation) # add row data
    f4.close()

# === BEGINS - CSV - MEAN OF TRANSPORT EACH YEAR
=====
with open('means_of_transport_each_year.csv', 'w', newline='') as f5: # opens (and creates)
.csv file
    writer = csv.writer(f5)
# === 2011 ===
    arr2011_tr = []
    arr2011_tr.append(2011)
    for i in range(0, 4):
        arr2011_tr.append(transportation_each_year[i][0]) # for i item get [0]-columns which
is arrivals
    writer.writerow(['Αφίξεις τουριστών στην Ελλάδα Ανά Μέσο Μεταφοράς και Ανά Έτος']) #
Title
    writer.writerow(['Έτος', 'Αεροπορικώς', 'Σιδηρώς', 'Θαλασσίως', 'Οδικώς']) # column
headers
    writer.writerow(arr2011_tr) # add row data
# === 2012 ===
    arr2012_tr = []
    arr2012_tr.append(2012)
    for i in range(4, 8):

```

```

        arr2012_tr.append(transportation_each_year[i][0])
    writer.writerow(arr2012_tr) # add row data
# === 2013 ===
    arr2013_tr = []
    arr2013_tr.append(2013)
    for i in range(8, 12):
        arr2013_tr.append(transportation_each_year[i][0])
    writer.writerow(arr2013_tr) # add row data
# === 2014 ===
    arr2014_tr = []
    arr2014_tr.append(2014)
    for i in range(12, 16):
        arr2014_tr.append(transportation_each_year[i][0])
    writer.writerow(arr2014_tr) # add row data
f5.close()

# === BEGINS - CSV - ARRIVALS PER QUARTER
=====
with open('arrivals_per_quarter.csv', 'w', newline='') as f6: # opens (and creates) .csv
file
    writer = csv.writer(f6)
# === 2011 ===
    arr2011_q = []
    arr2011_q.append(2011)
    for i in range(0, 4):
        arr2011_q.append(q2011[i])
    writer.writerow(['Αφίξεις τουριστών στην Ελλάδα Ανά Τρίμηνο για την τετραετία 2011-
2015'])
    writer.writerow(['Έτος', 'Ιαν-Φεβ-Μαρτ', 'Απρ-Μαι-Ιουν', 'Ιουλ-Αυγ-Σεπτ', 'Οκτ-Νοεμ-
Δεκ'])
    writer.writerow(arr2011_q)
# === 2012 ===
    arr2012_q = []
    arr2012_q.append(2012)
    for i in range(0, 4):
        arr2012_q.append(q2012[i])
    writer.writerow(arr2012_q)
# === 2013 ===
    arr2013_q = []
    arr2013_q.append(2013)
    for i in range(0, 4):
        arr2013_q.append(q2013[i])
    writer.writerow(arr2013_q)
# === 2014 ===
    arr2014_q = []
    arr2014_q.append(2014)
    for i in range(0, 4):
        arr2014_q.append(q2014[i])
    writer.writerow(arr2014_q)
f6.close()

print("Complete. The creation of all .csv files was sucessful")
# === ENDS - CVS -
=====
#

```

```

~~~~~
#
~~~~~
#
~~~~~

# === BEGINS - SQL -
=====

print("Starting the storage of data in a SQLite database")
# === BEGINS - SQL - TOTAL ARRIVALS
=====
conn = sqlite3.connect('statistics.db') # connects (and creates) database
c = conn.cursor() # open cursor
c.execute("DROP TABLE IF EXISTS total_tourists")
c.execute("CREATE TABLE total_tourists(year integer PRIMARY KEY, arrivals integer)")
for i in range(0, 4):
    c.execute("INSERT INTO total_tourists VALUES(?,?)", (i+2011, total_tourists[i]))
c.execute("SELECT * FROM total_tourists")
print("SQL - Total Tourist -\n", c.fetchall())

# === BEGINS - SQL - Top 10 Countries 2011-2014
=====
coun.remove('Eroç')
c.execute("DROP TABLE IF EXISTS top_countries")
c.execute("CREATE TABLE top_countries(country text PRIMARY KEY, arrivals integer)")
for i in range(0, 10):
    c.execute("INSERT INTO top_countries VALUES(?,?)", (coun[i], arr[i]))
c.execute("SELECT * FROM top_countries")
print("SQL - Top 10 Countries 2011-2014 -\n", c.fetchall())

# === BEGINS - SQL - Top 10 Countries EACH YEAR
=====
c.execute("DROP TABLE IF EXISTS top_countries_each_year")
c.execute("CREATE TABLE top_countries_each_year(year integer, country text, arrivals integer,
PRIMARY KEY (year, country))")
# using lists arr2011-14 from CSV section ( we don't want the year so we remove it)
arr2011.remove(2011)
arr2012.remove(2012)
arr2013.remove(2013)
arr2014.remove(2014)
for i in range(0, 10): # inserts to database
    c.execute("INSERT INTO top_countries_each_year VALUES(?,?,?)", (2011, coun[i],
arr2011[i]))
    c.execute("INSERT INTO top_countries_each_year VALUES(?,?,?)", (2012, coun[i],
arr2012[i]))
    c.execute("INSERT INTO top_countries_each_year VALUES(?,?,?)", (2013, coun[i],
arr2013[i]))
    c.execute("INSERT INTO top_countries_each_year VALUES(?,?,?)", (2014, coun[i],
arr2014[i]))
c.execute("SELECT * FROM top_countries_each_year") # prints table
print("SQL - Top 10 Countries For Each Year -\n", c.fetchall())

# === BEGINS - SQL - MEAN OF TRANSPORT

```

```

=====
transport = ["Αεροπορικώς", "Σιδηρώς", "Θαλασσίως", "Οδικώς"]
c.execute("DROP TABLE IF EXISTS transportation")
c.execute("CREATE TABLE transportation(transport text PRIMARY KEY, arrivals integer)")
for i in range(0, 4):
    c.execute("INSERT INTO transportation VALUES(?,?)", (transport[i], transportation[i]))
c.execute("SELECT * FROM transportation")
print("SQL - MEAN OF TRANSPORT (IN TOTAL) 2011-2015 -\n", c.fetchall())

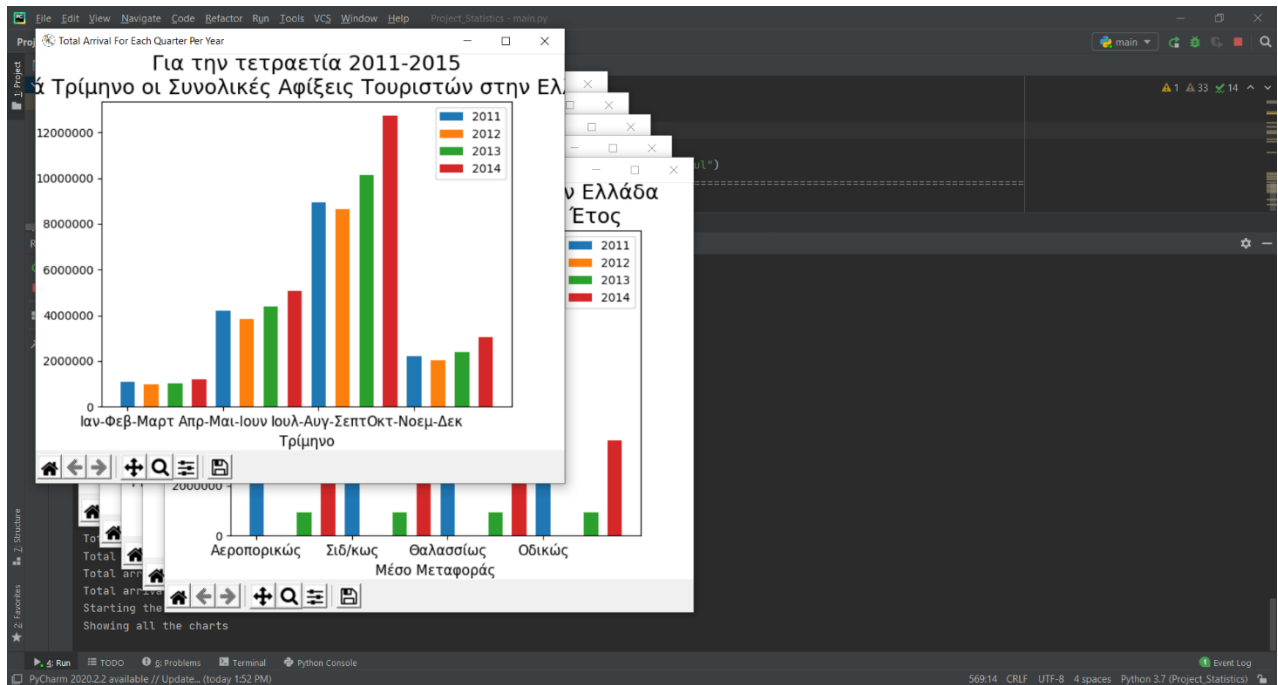
# === BEGINS - SQL - MEAN OF TRANSPORT EACH YEAR
=====
c.execute("DROP TABLE IF EXISTS transportation_each_year")
c.execute("CREATE TABLE transportation_each_year(year integer, transport text, arrivals
integer, PRIMARY KEY (year, transport))")
# using lists arr2011-14_tr from CSV section ( we don't want the year so we remove it)
arr2011_tr.remove(2011)
arr2012_tr.remove(2012)
arr2013_tr.remove(2013)
arr2014_tr.remove(2014)
for i in range(0, 4):
    c.execute("INSERT INTO transportation_each_year VALUES(?,?,?)", (2011, transport[i],
arr2011_tr[i]))
    c.execute("INSERT INTO transportation_each_year VALUES(?,?,?)", (2012, transport[i],
arr2012_tr[i]))
    c.execute("INSERT INTO transportation_each_year VALUES(?,?,?)", (2013, transport[i],
arr2013_tr[i]))
    c.execute("INSERT INTO transportation_each_year VALUES(?,?,?)", (2014, transport[i],
arr2014_tr[i]))
c.execute("SELECT * FROM transportation_each_year")
print("SQL - MEAN OF TRANSPORT EACH YEAR -\n", c.fetchall())

# === BEGINS - SQL - ARRIVALS PER QUARTER
=====
# from before: quarter_name = ['Ιαν-Φεβ-Μαρτ', 'Απρ-Μαι-Ιουν', 'Ιουλ-Αυγ-Σεπτ', 'Οκτ-Νοεμ-
Δεκ']
c.execute("DROP TABLE IF EXISTS quarter")
c.execute("CREATE TABLE quarter(year integer, quarter_name text, arrivals integer, PRIMARY
KEY (year, quarter_name))")
# using lists arr2011-14_q from CSV section (we don't want the year so we remove it)
arr2011_q.remove(2011)
arr2012_q.remove(2012)
arr2013_q.remove(2013)
arr2014_q.remove(2014)
for i in range(0, 4):
    c.execute("INSERT INTO quarter VALUES(?,?,?)", (2011, quarter_name[i], arr2011_q[i]))
    c.execute("INSERT INTO quarter VALUES(?,?,?)", (2012, quarter_name[i], arr2012_q[i]))
    c.execute("INSERT INTO quarter VALUES(?,?,?)", (2013, quarter_name[i], arr2013_q[i]))
    c.execute("INSERT INTO quarter VALUES(?,?,?)", (2014, quarter_name[i], arr2014_q[i]))
c.execute("SELECT * FROM quarter")
print("SQL - ARRIVALS PER QUARTER -\n", c.fetchall())

c.close() # close cursor
conn.commit()
conn.close() # close connection
print("Complete. The storage of data in a SQLite database was successful")
# === ENDS - SQL -
=====
=====

```

## 2. Screenshots παραδειγμάτων εφαρμογής



```
Project Statistics > main.py
c.close() # close cursor
conn.commit()
```

```
Run: main
Starting to download
Complete. Download was Successful
-----TOTAL ARRIVALS-----
Year 2011 : 16427247
Year 2012 : 15517622
Year 2013 : 17919580
Year 2014 : 22033462
-----TOTAL COUNTRIES-----
(1068383, 'Αυστρία')
(1513313, 'Βέλγιο')
(3511757, 'Βουλγαρία')
(4742139, 'Γαλλία')
(9076042, 'Γερμανία')
(893076, 'Δανία')
(53264, 'Εσθονία')
(7614749, 'Ην. Βασίλειο')
(203403, 'Ιρλανδία')
(538716, 'Ισπανία')
(3868332, 'Ιταλία')
(1711934, 'Κύπρος')
(74514, 'Λεττονία')
(125031, 'Λιθουανία')
(82614, 'Λουξεμβούργο')
(8769, 'Μάλτα')
(2277412, 'Ολλανδία')
(309231, 'Ουγγαρία')
(1679486, 'Πολωνία')
(82634, 'Πορτογαλία')
```

```
Project Statistics > main.py
c.close() # close cursor
conn.commit()
```

```
Run: main
(70430, 'Λοιπά κράτη Ωκεανίας')
(0, 'Μη προσδιορίσιμες χώρες ταξιδιωτών')
(26270, 'Κροατία')
-----TOP 10 COUNTRIES-----
1 (9076042, 'Γερμανία')
2 (7902849, 'Λοιπά κράτη Ευρώπης')
3 (7614749, 'Ην. Βασίλειο')
4 (4742139, 'Γαλλία')
5 (4216789, 'Ρωσία')
6 (3868332, 'Ιταλία')
7 (3511757, 'Βουλγαρία')
8 (3076935, 'Σερβία - Μαυροβούνιο')
9 (2962267, 'Τουρκία')
10 (2277412, 'Ολλανδία')
-----TOP 10 COUNTRIES EACH YEAR -----
Year 11 -----
1 (2240481, 'Γερμανία')
2 (1758093, 'Ην. Βασίλειο')
3 (1523236, 'Λοιπά Κράτη Ευρώπης')
4 (1149388, 'Γαλλία')
5 (938232, 'Ιταλία')
6 (738927, 'Ρωσία')
7 (692059, 'Σερβία - Μαυροβούνιο')
8 (686209, 'Βουλγαρία')
9 (560723, 'Ολλανδία')
10 (552090, 'Τουρκία')
Year 12 -----
1 (2108787, 'Γερμανία')
2 (1920794, 'Ην. Βασίλειο')
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Project_Stat
Project_Statistics main.py
Project_Statistics C:\Users\venv\library root
main.py
c.close() # close cursor
conn.commit()

Run: main
9 (976758, 'ΠΟΥΡΚΙ(Α')
10 (657339, 'ΟΛΛΑΝΔ(Α')
----ARRIVALS PER MEAN OF TRANSPORT----
YEAR 2011
Transportation: by plane | Total arrivals: 11671155
Transportation: by rail | Total arrivals: 3765
Transportation: by sea | Total arrivals: 947848
Transportation: by road | Total arrivals: 3804479
YEAR 2012
Transportation: by plane | Total arrivals: 10992903
Transportation: by rail | Total arrivals: 0
Transportation: by sea | Total arrivals: 790469
Transportation: by road | Total arrivals: 3734249
YEAR 2013
Transportation: by plane | Total arrivals: 12302448
Transportation: by rail | Total arrivals: 0
Transportation: by sea | Total arrivals: 806960
Transportation: by road | Total arrivals: 4810172
YEAR 2014
Transportation: by plane | Total arrivals: 14057215
Transportation: by rail | Total arrivals: 6894
Transportation: by sea | Total arrivals: 701345
Transportation: by road | Total arrivals: 7268007
TOTAL ARRIVALS PER MEAN OF TRANSPORT
Transportation by plane : 49023721
Transportation by rail : 10659
Transportation by sea : 3246622
Transportation by road : 19616907
----ARRIVALS PER QUARTER----
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Project_Stat
Project_Statistics main.py
Project_Statistics C:\Users\venv\library root
main.py
c.close() # close cursor
conn.commit()

Run: main
9 (976758, 'ΠΟΥΡΚΙ(Α')
10 (657339, 'ΟΛΛΑΝΔ(Α')
----ARRIVALS PER MEAN OF TRANSPORT----
YEAR 2011
Transportation: by plane | Total arrivals: 11671155
Transportation: by rail | Total arrivals: 3765
Transportation: by sea | Total arrivals: 947848
Transportation: by road | Total arrivals: 3804479
YEAR 2012
Transportation: by plane | Total arrivals: 10992903
Transportation: by rail | Total arrivals: 0
Transportation: by sea | Total arrivals: 790469
Transportation: by road | Total arrivals: 3734249
YEAR 2013
Transportation: by plane | Total arrivals: 12302448
Transportation: by rail | Total arrivals: 0
Transportation: by sea | Total arrivals: 806960
Transportation: by road | Total arrivals: 4810172
YEAR 2014
Transportation: by plane | Total arrivals: 14057215
Transportation: by rail | Total arrivals: 6894
Transportation: by sea | Total arrivals: 701345
Transportation: by road | Total arrivals: 7268007
TOTAL ARRIVALS PER MEAN OF TRANSPORT
Transportation by plane : 49023721
Transportation by rail : 10659
Transportation by sea : 3246622
Transportation by road : 19616907
----ARRIVALS PER QUARTER----
YEAR 2011
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Project_Statistics - main.py
Project_Statistics main.py
Project_Statistics C:\Users\venv\library root
main.py
c.close() # close cursor
conn.commit()

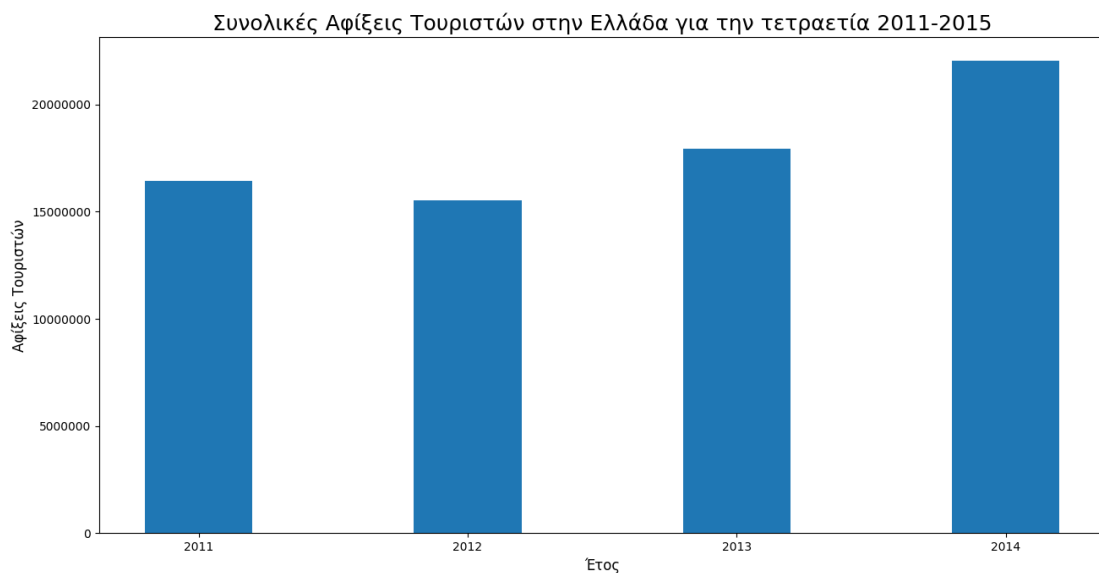
Run: main
Transportation by plane : 49023721
Transportation by rail : 10659
Transportation by sea : 3246622
Transportation by road : 19616907
----ARRIVALS PER QUARTER----
YEAR 2011
Total arrivals: 1108387 | Quarter: January-February-March
Total arrivals: 4195768 | Quarter: April-May-June
Total arrivals: 8925699 | Quarter: July-August-September
Total arrivals: 2197393 | Quarter: October-November-December
YEAR 2012
Total arrivals: 978559 | Quarter: January-February-March
Total arrivals: 3849244 | Quarter: April-May-June
Total arrivals: 8655186 | Quarter: July-August-September
Total arrivals: 2834633 | Quarter: October-November-December
YEAR 2013
Total arrivals: 1023354 | Quarter: January-February-March
Total arrivals: 4397477 | Quarter: April-May-June
Total arrivals: 10113076 | Quarter: July-August-September
Total arrivals: 2385673 | Quarter: October-November-December
YEAR 2014
Total arrivals: 1186900 | Quarter: January-February-March
Total arrivals: 5077137 | Quarter: April-May-June
Total arrivals: 12722924 | Quarter: July-August-September
Total arrivals: 3046501 | Quarter: October-November-December
Starting the creation of charts
Showing all the charts
Starting the creation of .csv file
Complete. The creation of .csv file was successful
Starting the storage of data in a SQLite database
```



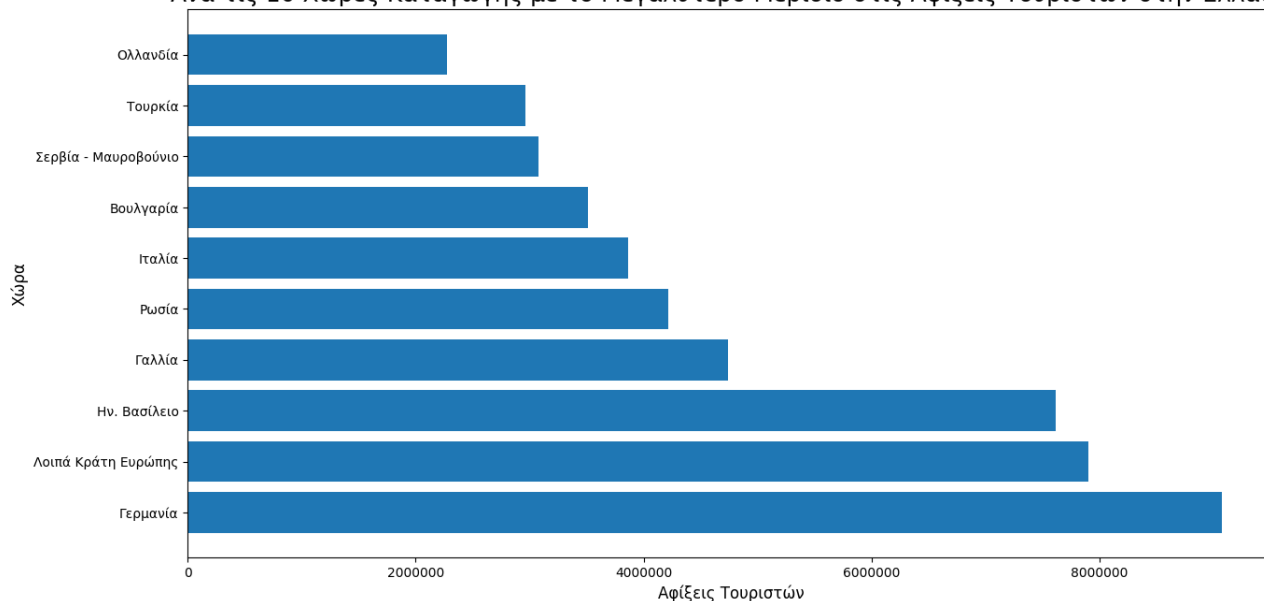
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ProjectStatistics - main.py
Project Statistics - main.py
main.py
Run
main
Total arrivals: 1623334 | Quarter: January-February-March
Total arrivals: 4397477 | Quarter: April-May-June
Total arrivals: 18113876 | Quarter: July-August-September
Total arrivals: 2385673 | Quarter: October-November-December
----- YEAR 2014 -----
Total arrivals: 1186988 | Quarter: January-February-March
Total arrivals: 5877137 | Quarter: April-May-June
Total arrivals: 12722924 | Quarter: July-August-September
Total arrivals: 3846581 | Quarter: October-November-December
Starting the creation of charts
Showing all the charts
Starting the creation of .csv file
Complete. The creation of .csv file was successful
Starting the storage of data in a SQLite database
SQL - Total Tourist -
[(2011, 16427247), (2012, 15517622), (2013, 17919588), (2014, 22833462)]
SQL - Top 10 Countries 2011-2014 -
[('Γερμανία', 9876842), ('Ην. Βασίλειο', 7982849), ('Λοιπά Κράτη Ευρώπης', 7614749), ('Γαλλία', 4742139), ('Ιταλία', 4216789), ('Ρωσία', 3868332), ('Σερβία - Μαυροβούνιο', 3511757), ('Αεροπορικές', 34923721), ('Σιδηρόδρομος', 19616987)]
SQL - Top 10 Countries For Each Year -
[(2011, 'Γερμανία', 2240481), (2012, 'Γερμανία', 2188787), (2013, 'Γερμανία', 2267546), (2014, 'Γερμανία', 2459228), (2011, 'Ην. Βασίλειο', 1758893), (2012, 'Ην. Βασίλειο', 192879), (2011, 'Αεροπορικές', 11671155), (2012, 'Αεροπορικές', 10992983), (2013, 'Αεροπορικές', 12382448), (2014, 'Αεροπορικές', 14857215), (2011, 'Σιδηρόδρομος', 3765), (2012, 'Σιδηρόδρομος', 0), (2011, 'Ιαν-Φεβ-Μαρτ', 1108387), (2012, 'Ιαν-Φεβ-Μαρτ', 978559), (2013, 'Ιαν-Φεβ-Μαρτ', 1023354), (2014, 'Ιαν-Φεβ-Μαρτ', 1186988), (2011, 'Απρ-Μαι-Ιουν', 4195768), (2012, 'Απρ-Μαι-Ιουν', 4195768)]
Complete. The storage of data in a SQLite database was successful
Process finished with exit code 0
```

### 3. Γραφήματα

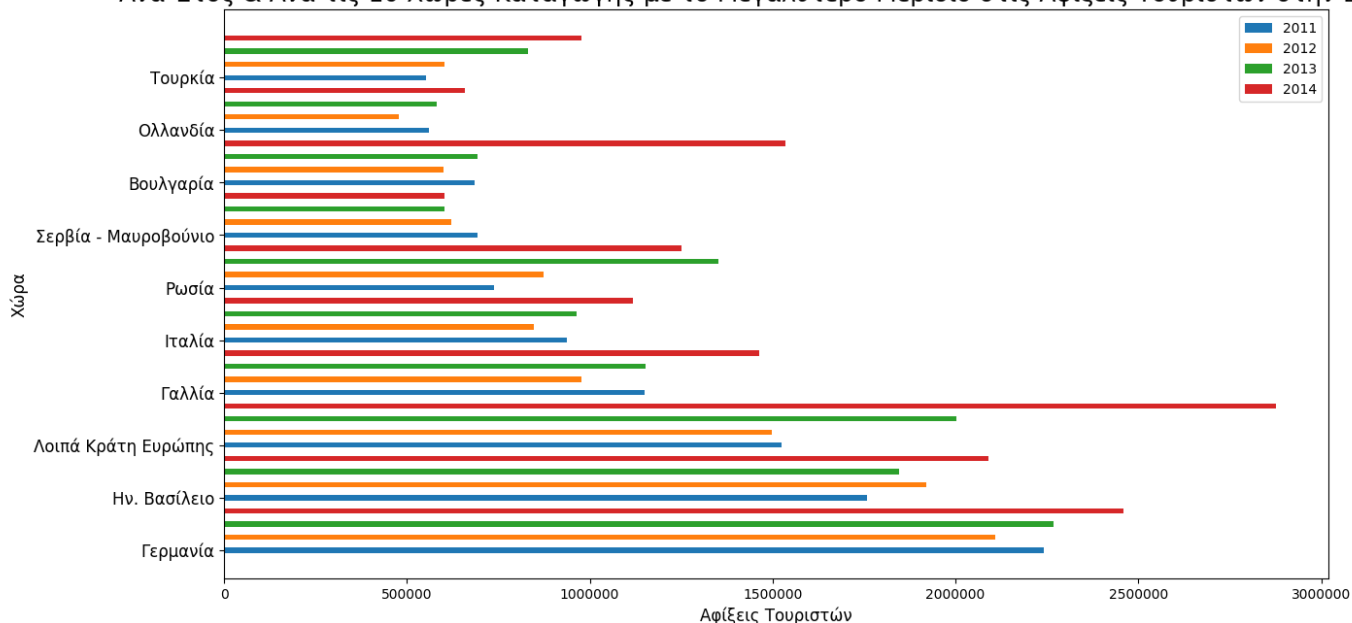
Σημείωση: Δεν ήμουν σίγουρη σε μερικά υποερωτήματα, εάν η εκφώνηση εννοούσε τις συνολικές αφίξεις και για τα τέσσερα χρόνια συνολικά ή τις συνολικές αφίξεις ξεχωριστά για κάθε χρονιά ξεχωριστά μέσα στην τετραετία, οπότε έκανα γραφήματα και για τις δύο περιπτώσεις.



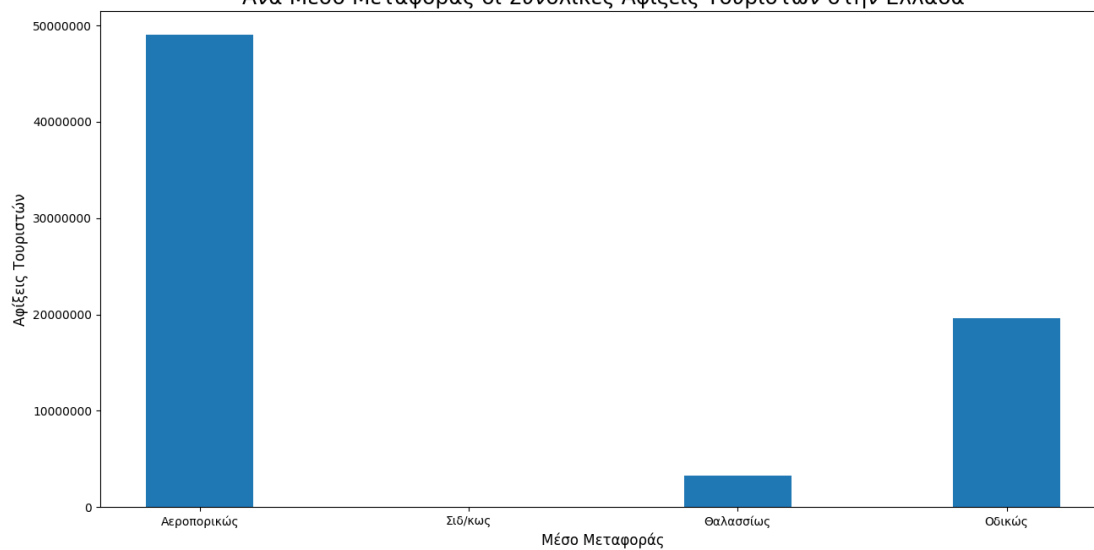
Συνολικές Αφίξεις Τουριστών για την τετραετία 2011-2015  
Ανά τις 10 Χώρες Καταγωγής με το Μεγαλύτερο Μερίδιο στις Αφίξεις Τουριστών στην Ελλάδα



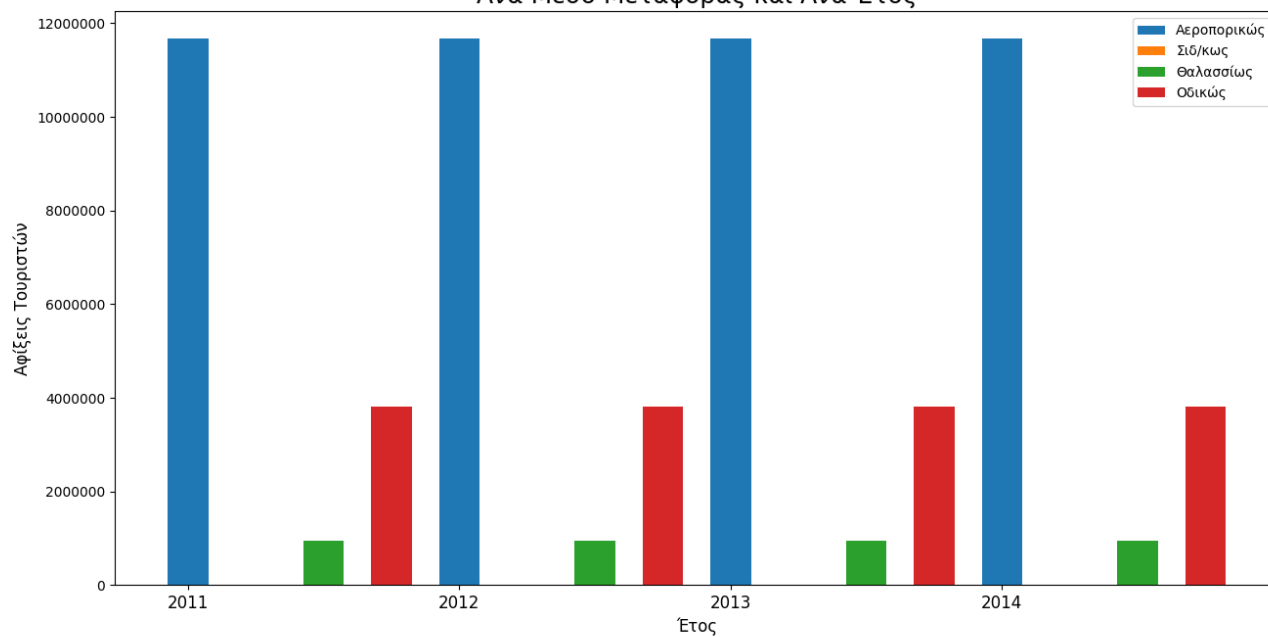
Συνολικές Αφίξεις Τουριστών  
Ανά Έτος & Ανά τις 10 Χώρες Καταγωγής με το Μεγαλύτερο Μερίδιο στις Αφίξεις Τουριστών στην Ελλάδα



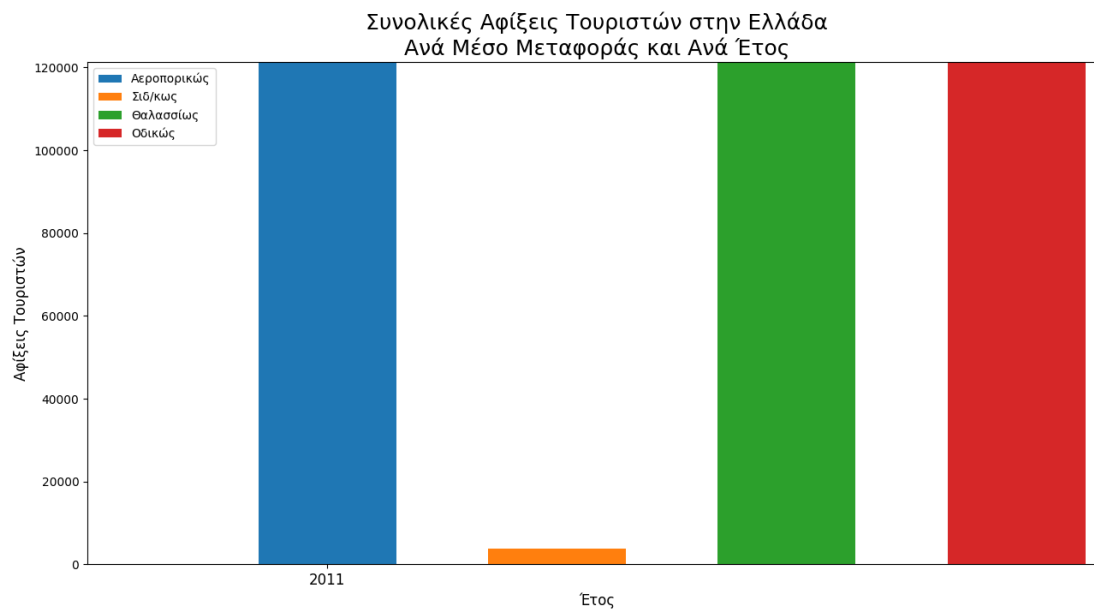
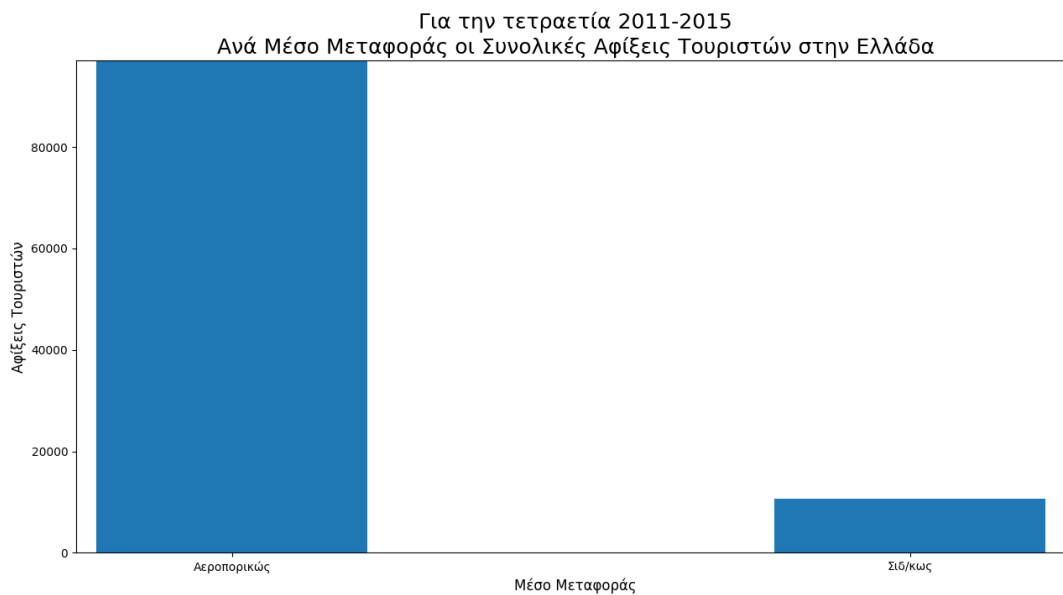
Για την τετραετία 2011-2015  
Ανά Μέσο Μεταφοράς οι Συνολικές Αφίξεις Τουριστών στην Ελλάδα



Συνολικές Αφίξεις Τουριστών στην Ελλάδα  
Ανά Μέσο Μεταφοράς και Ανά Έτος



Σημείωση: Στα δύο παραπάνω γραφήματα το «Σιδ/κως» υπάρχει, απλά είναι πολύ μικρό σε σύγκριση με τα άλλα τρία. Φαίνεται καλύτερα στις δύο παρακάτω εικόνες που έχει γίνει zoom in.



Για την τετραετία 2011-2015  
Ανά Τρίμηνο οι Συνολικές Αφίξεις Τουριστών στην Ελλάδα

