

Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών

Προαιρετική Εργαστηριακή Άσκηση 2020-2021

Γιαννοπούλου Αρχοντία AM 1070933 'Γ' ΕΤΟΣ

Είναι σημαντικό να αναφερθεί ότι για την υλοποίηση της άσκησης επιλέχθηκαν 2 αρχεία τα οποία περιείχαν αποτελέσματα για τους κατοίκους της κάθε χώρας, τους επισκέπτες και τις συνολικές τιμές και των 2. Κάθε ένα από αυτά τα αρχεία φιλτραρίστηκαν και πάρθηκαν σε ξεχωριστά dataframes .

Τα έτη που επιλέχθηκαν είναι : 2016 2017 2018 2019

Οι χώρες που επιλέχθηκαν είναι : Ελλάδα και Ισπανία

1. Κώδικας Python

Αρχικά θα γίνει μια σύντομη εξήγηση του κώδικα. Ο κώδικας εμφανίζεται παρακάτω καθώς και στο αρχείο main.py .

Ο κώδικας χωρίζεται σε 4 μέρη

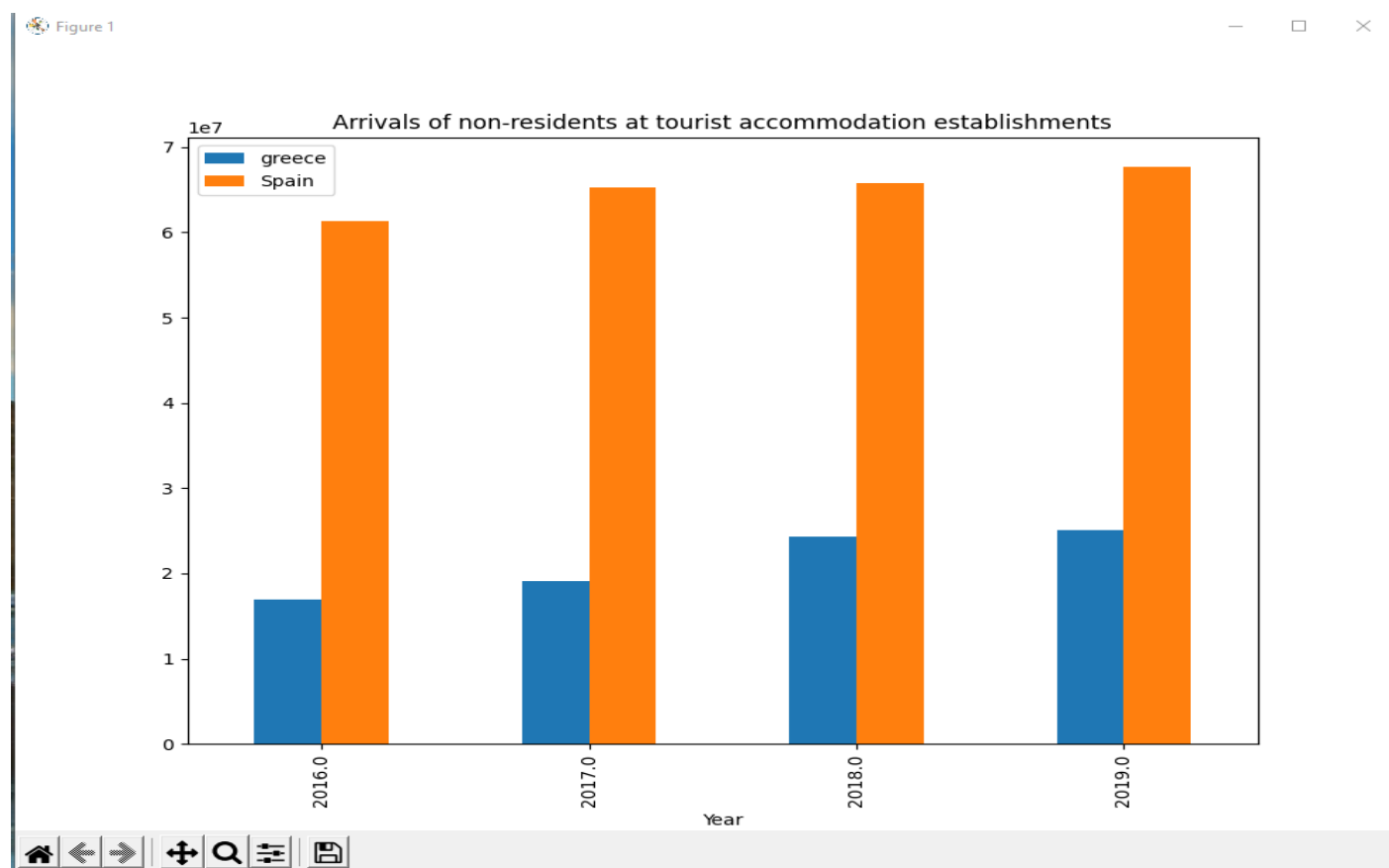
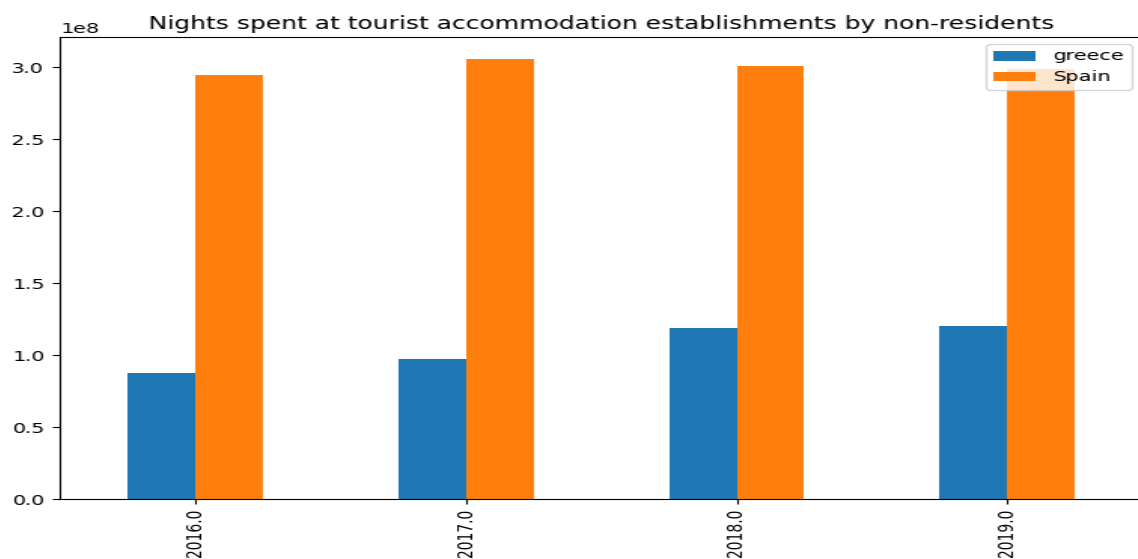
- 1^ο μέρος: Λαμβάνονται τα κατάλληλα δεδομένα από την ιστοσελίδα της Eurostat .Για το φιλτράρισμα των δεδομένων δημιουργήθηκαν 2 συναρτήσεις(ανάλογα με τα δεδομένα που θέλαμε να πάρουμε) οι οποίες υλοποιούν τα εξής : Αποθηκεύει προσωρινά τα αρχεία στην ρύθση και στη συνέχεια πραγματοποιεί αποσυμπίεση αυτών(Από tsv.gz σε txt). Γίνεται μετατροπή αυτών σε dataframes .Ύστερα φιλτράρονται κατάλληλα ορίζοντας τις σωστές τιμές για τις γραμμές(FOR,NR,I551-I553,EL TOTAL,NR,I551-I553,EL FOR,NR,I551-I553,ES TOTAL,NR,I551-I553,EL) και για τις στήλες(2016,2017,2018,2019)αντίστοιχα και για τα 2 αρχεία.
- 2^ο μέρος: Εξαγωγή γραφημάτων από τα φιλτραρισμένα δεδομένα. Αρχικά οι στήλες μετατράπηκαν σε γραμμές και οι γραμμές σε στήλες αντίστοιχα .Στη συνέχεια δημιουργήθηκε μια καινούργια στήλη που περιείχε όλα τα έτη και αφαιρέθηκε μία. Τέλος οι τιμές των στηλών "Greece","Spain","Year" λαμβάνονται ως είσοδοι για την δημιουργία των αντίστοιχων γραφημάτων στην matplotlib.
- 3^ο μέρος:Σύνδεση στην βάση δεδομένων και δημιουργία πινάκων. Για την σύνδεση σε μια βάση sql χρησιμοποιήθηκε η εφαρμογή χαμpp. Αρχικά <<καθαρίζει>> η βάση αν υπάρχει ήδη και στη συνέχεια δημιουργούνται 4 πίνακες για κάθε ένα από τα ζητούμενα. Αφού δημιουργηθούν οι πίνακες λαμβάνονται οι τιμές στους πίνακες από ολόκληρα τα dataframes. Η λήψη των τιμών έγινε μετά την επεξεργασία των dataframes και λίγο πριν την δημιουργία γραφημάτων.
- 4^ο μέρος: Μετατροπή των dataframes σε csv files. Στο τελευταίο κομμάτι κώδικα τα dataframes που δημιουργήθηκαν μετατρέπονται σε αρχεία csv.

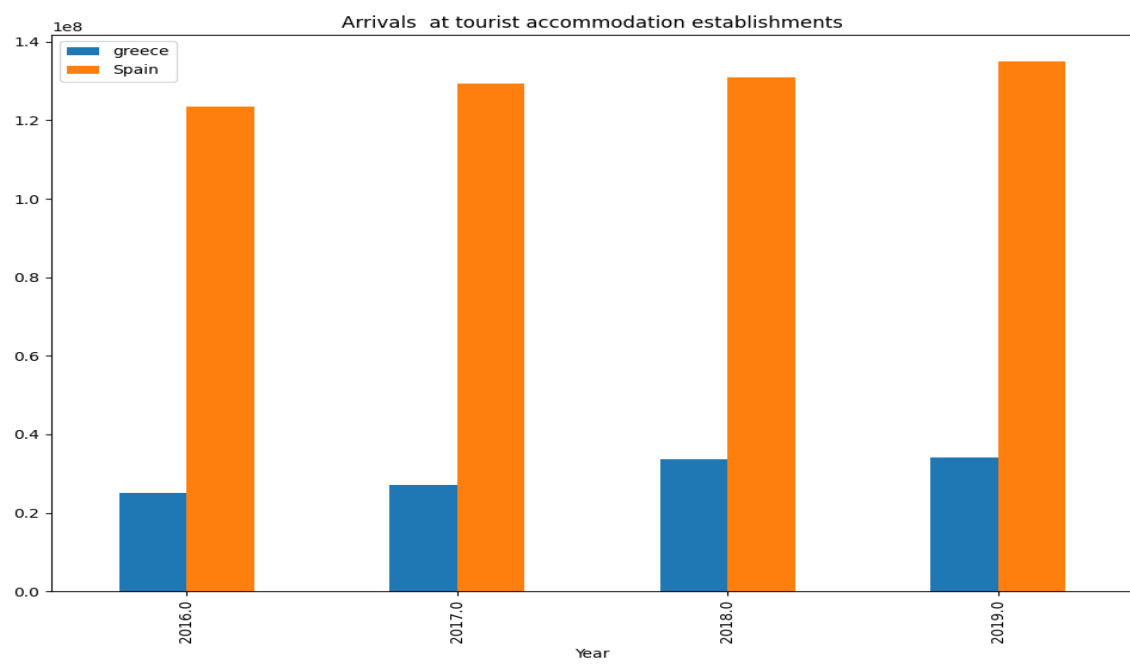
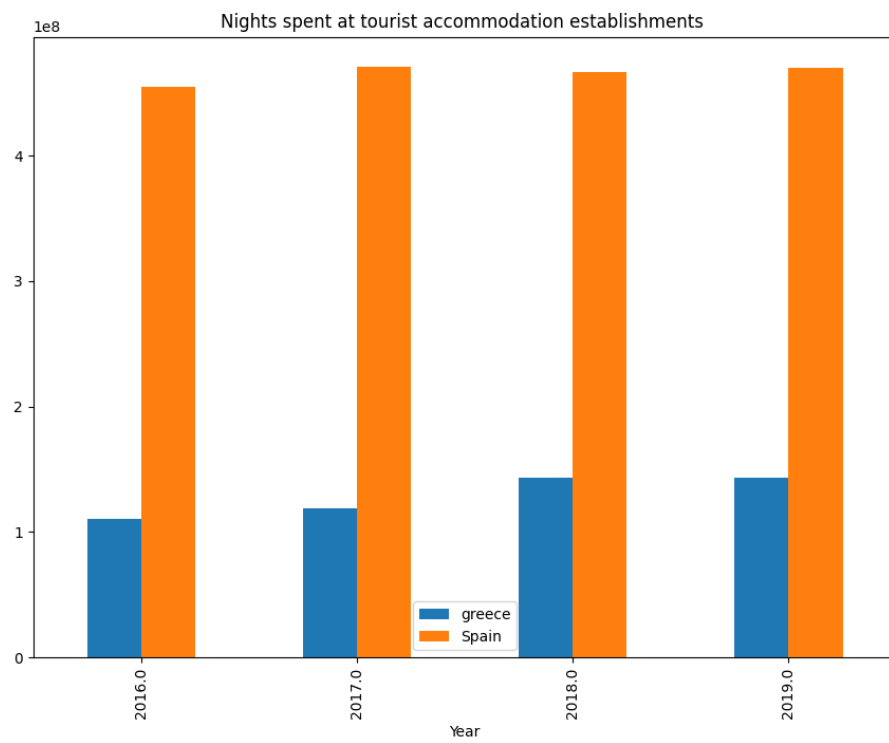
Τα παραπάνω χρησιμοποιήθηκαν για την υλοποίηση και των 4 ζητημάτων τα αποτελέσματα και οι τιμές των οποίων φαίνονται σε ξεχωριστά γραφήματα .

Σημείωση: Για την διαπίστωση και τεκμηρίωση όλων των παραπάνω στον κώδικα, χρησιμοποιήθηκαν συναρτήσεις print().

2. & .3 Screenshots από γραφήματα, αρχεία, υλοποιήσιμο κώδικα, csv files,sql

Γραφήματα





Υλοποίησης κώδικας :

```
main.py | Night2.txt | Arrivals.txt | Night2_Non.txt
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import gzip
4 import requests
5 import shutil
6 import mysql.connector
7 #connect and create database and tables for each link separate
8 db=mysql.connector.connect(user='root',password='',host='localhost')
9 cursor = db.cursor()
10 cursor.execute("DROP DATABASE IF EXISTS AGP")
11 cursor.execute("CREATE DATABASE IF NOT EXISTS AGP")
12 cursor.execute("USE AGP")
13 cursor.execute("CREATE TABLE IF NOT EXISTS NIGHT(year year(4) NOT NULL, greece FLOAT NOT NULL, spain BIGINT not null)")
14 cursor.execute("CREATE TABLE IF NOT EXISTS ARRIVALS(year year(4) NOT NULL, greece FLOAT NOT NULL, spain BIGINT not null)")
15 cursor.execute("CREATE TABLE IF NOT EXISTS NIGHT_NON(year year(4) NOT NULL, greece FLOAT NOT NULL, spain BIGINT not null)")
16 cursor.execute("CREATE TABLE IF NOT EXISTS ARRIVALS_NON(year year(4) NOT NULL, greece FLOAT NOT NULL, spain BIGINT not null)")
17 #visit each url that we are interested and parses into variables
18 url = 'https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tin00175.tsv.gz'
19 url1= 'https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tin00174.tsv.gz'
20 #function that download files from the web and filtering the data
21 def download_file(url, file1, file2):
22     r = requests.get(url, allow_redirects=True)
23     open(file1, 'wb').write(r.content)
24     with gzip.open(file1, 'rb') as f_in: #Decompress gzip File
25         with open(file2, 'wb') as f_out: #convert gz in txt
26             shutil.copyfileobj(f_in, f_out)
27
28     df = pd.read_csv(file2, sep='t', header=0) #read the file as csv keeps only the columns and rows that we want
29     df_all= df[df['c_resid,unit,page_r2,geo\\time'].str.contains("FOR,NR,I551-I553,EL|FOR,NR,I551-I553,ES",regex=True)]#filtering.Keep only the columns and rows tha we want
30     df_all= df_all[['2016 ', '2017 ', '2018 ', '2019 ']]
31
32     print(df_all)
33     return df_all
34 #execute function
35 df_night_NON= download_file(url, 'Night1.gz', 'Night2_Non.txt')
36 df_arrivals_NON= download_file(url1, 'Arrivals1.gz', 'Arrivals2_Non.txt')
37
38 def download_file1(url, file1, file2):
39     r = requests.get(url, allow_redirects=True)
40     open(file1, 'wb').write(r.content)
41     with gzip.open(file1, 'rb') as f_in: #Decompress gzip File
42         download_file()
```

```
Night2.txt | arrivals_non.csv | main.py | Arrivals.txt | Night2_Non.txt
40 open(file1, 'wb').write(r.content)
41 with gzip.open(file1, 'rb') as f_in: #Decompress gzip File
42     with open(file2, 'wb') as f_out: #convert gz in txt
43         shutil.copyfileobj(f_in, f_out)
44
45 df = pd.read_csv(file2, sep='t', header=0) #read the file as csv keeps only the columns and rows that we want
46 df_all= df[df['c_resid,unit,page_r2,geo\\time'].str.contains("TOTAL,NR,I551-I553,EL|TOTAL,NR,I551-I553,ES",regex=True)]#filtering.Keep only the columns and rows tha we want
47 df_all = df_all[['2016 ', '2017 ', '2018 ', '2019 ']]
48
49 print(df_all)
50 return df_all
51
52 #execute function
53 df_night= download_file1(url, 'Night1.gz', 'Night2.txt')
54 df_arrivals= download_file1(url1, 'Arrivals1.gz', 'Arrivals2.txt')
55
56
57 #ploting
58 print(df_night_NON.T)#changes columns to rows and rows to columns
59 df_night_NON.T = df_night_NON.T
60 df_night_NON.T["Year"] = df_night_NON.keys()#contains the 4 years of the dictionary as a list
61 print(df_night_NON.T)
62 df_night_NON.T=df_night_NON.T.rename(columns={10: 'greece',11: 'Spain'}, inplace_=False)#changes the name of the columns
63 print(df_night_NON.T)
64 df_night_NON.T= df_night_NON.T.reset_index()
65 print(df_night_NON.T)
66 df_night_NON.T = df_night_NON.T.drop(columns="index")#avoid the old index
67 #sql : insert the values of dataframe into table
68 sql = "INSERT INTO 'NIGHT_NON' ('year', 'greece', 'spain') VALUES (%s, %s, %s)"
69 for i,row in df_night_NON.T.iterrows():
70     cursor.execute(sql, (row["Year"], row["greece"], row["Spain"]))
71     print(row)
72 db.commit()
73 #To verify the results print the values of table
74 for i,row in df_night_NON.T.iterrows():
75     cursor.execute("SELECT * FROM NIGHT_NON")
76     print(cursor.fetchall())
77 #bar plotting data of link
78 df_night_NON.T = df_night_NON.T.astype(float)
79 ax = plt.gca()
80 df_night_NON.T.plot(kind='bar', y=['greece','Spain'],x='Year',ax=ax)
81 plt.title('Nights spent at tourist accommodation establishments by non-residents')
```

```
main.py | Night2.txt | Arrivals2.txt | Night2_Non.txt
78 df_night_NON_T = df_night_NON_T.astype(float)
79 ax = plt.gca()
80 df_night_NON_T.plot(kind='bar', y=['greece','Spain'],x='Year',ax=ax)
81 plt.title('Nights spent at tourist accommodation establishments by non-residents')
82 plt.show()
83
84
85
86
87 print(df_arrivals_NON_T)
88 df_arrivals_NON_T = df_arrivals_NON_T
89 df_arrivals_NON_T["Year"] = df_arrivals_NON_T.keys()
90 print(df_arrivals_NON_T)
91 df_arrivals_NON_T=df_arrivals_NON_T.rename(columns={'10: 'greece': 'Spain'}, inplace=False)
92 df_arrivals_NON_T = df_arrivals_NON_T.reset_index()
93 print(df_arrivals_NON_T)
94 df_arrivals_NON_T = df_arrivals_NON_T.drop(columns="index")
95 print(df_arrivals_NON_T)
96 #sql: insert the values of dataframe into table
97 sql = "INSERT INTO 'ARRIVALS_NON' ('year', 'greece', 'spain') VALUES (%s, %s, %s)"
98 for i,row in df_arrivals_NON_T.iterrows():
99     cursor.execute(sql, (row["Year"], row["greece"], row["Spain"]))
100     print(row)
101 db.commit()
102 #To verify the results print the values of table
103 for i,row in df_arrivals_NON_T.iterrows():
104     cursor.execute("SELECT * FROM ARRIVALS_NON ")
105     print(cursor.fetchall())
106 df_arrivals_T = df_arrivals_NON_T.astype(float)
107 #bar plotting data of link
108 ax = plt.gca()
109 df_arrivals_T.plot(kind='bar', y=['greece','Spain'],x='Year',ax=ax)
110 plt.title('Arrivals of non-residents at tourist accommodation establishments')
111 plt.show()
112
113 print(db)
114
115 #plotting
116 print(df_night_T)#changes columns to rows and rows to columns
117 df_night_T = df_night_T
118
```

```
main.py | Night2.txt | Arrivals2.txt | Night2_Non.txt
115 #plotting
116 print(df_night_T)#changes columns to rows and rows to columns
117 df_night_T = df_night_T
118 df_night_T["Year"] = df_night.keys()#contains the 4 years of the dictionary as a list
119 print(df_night_T)
120 df_night_T=df_night_T.rename(columns={'90: 'greece': 'Spain'}, inplace=False)#changes the name of the columns
121 print(df_night_T)
122 df_night_T = df_night_T.reset_index()
123 print(df_night_T)
124 df_night_T = df_night_T.drop(columns="index")#avoid the old index
125 #sql: insert the values of dataframe into table
126 sql = "INSERT INTO 'NIGHT' ('year', 'greece', 'spain') VALUES (%s, %s, %s)"
127 for i,row in df_night_T.iterrows():
128     cursor.execute(sql, (row["Year"], row["greece"], row["Spain"]))
129     print(row)
130 db.commit()
131 #To verify the results print the values of table
132 for i,row in df_night_T.iterrows():
133     cursor.execute("SELECT * FROM NIGHT")
134     print(cursor.fetchall())
135 #bar plotting data of link
136 df_night_T = df_night_T.astype(float)
137 ax = plt.gca()
138 df_night_T.plot(kind='bar', y=['greece','Spain'],x='Year',ax=ax)
139 plt.title('Nights spent at tourist accommodation establishments ')
140 plt.show()
141
142 print(df_arrivals_T)
143 df_arrivals_T = df_arrivals_T
144 df_arrivals_T["Year"] = df_arrivals.keys()
145 print(df_arrivals_T)
146 df_arrivals_T=df_arrivals_T.rename(columns={'90: 'greece': 'Spain'}, inplace=False)
147 df_arrivals_T = df_arrivals_T.reset_index()
148 print(df_arrivals_T)
149 df_arrivals_T = df_arrivals_T.drop(columns="index")
150 print(df_arrivals_T)
151 #sql: insert the values of dataframe into table
152 sql = "INSERT INTO 'ARRIVALS' ('year', 'greece', 'spain') VALUES (%s, %s, %s)"
153 for i,row in df_arrivals_T.iterrows():
154     cursor.execute(sql, (row["Year"], row["greece"], row["Spain"]))
155     print(row)
156 for i,row in df_night_T.iterrow...
```

```

154 cursor.execute(sql, (row["Year"], row["greece"], row["Spain"]))
155 print(row)
156 db.commit()
157 #To verify the results print the values of table
158 for i,row in df_arrivals.T.iterrows():
159     cursor.execute("SELECT * FROM ARRIVALS ")
160     print(cursor.fetchall())
161 df_arrivals_T = df_arrivals.T.astype(float)
162 #bar plotting data of link
163 ax = plt.gca()
164 df_arrivals_T.plot(kind='bar', y=['greece','Spain'],x='Year',ax=ax)
165 plt.title('Arrivals at tourist accommodation establishments')
166 plt.show()
167
168
169 #converting all the dataframes in csv files
170 df_night_NON.to_csv('nights_non.csv',index=False)
171
172 df_arrivals_NON.to_csv('arrivals_non.csv',index=False)
173
174 df_night.to_csv('nights_total.csv',index=False)
175
176 df_arrivals.to_csv('arrivals_total.csv',index=False)
177 #verification of results
178 df_final_nights_non = pd.read_csv('nights_non.csv', sep=',', header=0)
179
180 df_final_arrivals_non = pd.read_csv('arrivals_non.csv', sep=',', header=0)
181
182 df_final_nights_total = pd.read_csv('nights_total.csv', sep=',', header=0)
183
184 df_final_arrivals_total = pd.read_csv('arrivals_total.csv', sep=',', header=0)
185 #print
186 df_final_nights_non = pd.read_csv('nights_non.csv', sep=',', header=0)
187 print(df_final_nights_non)
188
189 df_final_arrivals_non = pd.read_csv('arrivals_non.csv', sep=',', header=0)
190 print(df_final_arrivals_non)
191
192 df_final_nights_total = pd.read_csv('nights_total.csv', sep=',', header=0)
193 print(df_final_nights_total)
194

```

```

194
195 df_final_arrivals_total = pd.read_csv('arrivals_total.csv', sep=',', header=0)
196 print(df_final_arrivals_total)
197
198
199

```

Οι τιμές των dataframes και τα αποτελέσματα από την επεξεργασία αυτών :

Nights spent by non-residents at tourist accommodation establishments:

		10	11	
2016	87912850	294556428		
2017	97034421	305907462		
2018	118876323	301022634		
2019	119971390	299091409		
	10	11	Year	
2016	87912850	294556428	2016	
2017	97034421	305907462	2017	
2018	118876323	301022634	2018	
2019	119971390	299091409	2019	
	greece	Spain	Year	
2016	87912850	294556428	2016	
2017	97034421	305907462	2017	
2018	118876323	301022634	2018	
2019	119971390	299091409	2019	
	index	greece	Spain	Year
0	2016	87912850	294556428	2016
1	2017	97034421	305907462	2017
2	2018	118876323	301022634	2018
3	2019	119971390	299091409	2019
greece		87912850		
Spain		294556428		
Year		2016		
Name: 0, dtype: object				
greece		97034421		
Spain		305907462		
Year		2017		
Name: 1, dtype: object				
greece		118876323		
Spain		301022634		
Year		2018		
Name: 2, dtype: object				
greece		119971390		
Spain		299091409		
Year		2019		

Arrivals of non-residents at tourist accommodation establishments:

	10	11		
2016	16915996	61341839		
2017	19068697	65233045		
2018	24320893	65771059		
2019	25038498	67728098		
	10	11	Year	
2016	16915996	61341839	2016	
2017	19068697	65233045	2017	
2018	24320893	65771059	2018	
2019	25038498	67728098	2019	
index	greece	Spain	Year	
0	2016	16915996	61341839	2016
1	2017	19068697	65233045	2017
2	2018	24320893	65771059	2018
3	2019	25038498	67728098	2019
	greece	Spain	Year	
0	16915996	61341839	2016	
1	19068697	65233045	2017	
2	24320893	65771059	2018	
3	25038498	67728098	2019	
greece	16915996			
Spain	61341839			
Year	2016			
Name: 0, dtype: object				
greece	19068697			
Spain	65233045			
Year	2017			
Name: 1, dtype: object				
greece	24320893			
Spain	65771059			
Year	2018			
Name: 2, dtype: object				
greece	25038498			
Spain	67728098			
Year	2019			
Name: 3, dtype: object				

Nights spent at tourist accommodation establishments:

		90	91	
2016	110020042	454957250		
2017	119009014	471199729		
2018	142940411	466940717		
2019	143594467	469813551		
	90	91	Year	
2016	110020042	454957250	2016	
2017	119009014	471199729	2017	
2018	142940411	466940717	2018	
2019	143594467	469813551	2019	
	greece	Spain	Year	
2016	110020042	454957250	2016	
2017	119009014	471199729	2017	
2018	142940411	466940717	2018	
2019	143594467	469813551	2019	
	index	greece	Spain	Year
0	2016	110020042	454957250	2016
1	2017	119009014	471199729	2017
2	2018	142940411	466940717	2018
3	2019	143594467	469813551	2019
greece	110020042			
Spain	454957250			
Year	2016			
Name: 0, dtype: object				
greece	119009014			
Spain	471199729			
Year	2017			
Name: 1, dtype: object				
greece	142940411			
Spain	466940717			
Year	2018			
Name: 2, dtype: object				
greece	143594467			
Spain	469813551			
Year	2019			
Name: 3, dtype: object				

```

90          91
2016  24996038  123541778
2017  27211268  129392382
2018  33585639  130803657
2019  34202053  135008823
      90          91      Year
2016  24996038  123541778  2016
2017  27211268  129392382  2017
2018  33585639  130803657  2018
2019  34202053  135008823  2019
      index      greece      Spain      Year
0  2016  24996038  123541778  2016
1  2017  27211268  129392382  2017
2  2018  33585639  130803657  2018
3  2019  34202053  135008823  2019
      greece      Spain      Year
0  24996038  123541778  2016
1  27211268  129392382  2017
2  33585639  130803657  2018
3  34202053  135008823  2019
greece  24996038 |
Spain   123541778
Year    2016
Name: 0, dtype: object
greece  27211268
Spain   129392382
Year    2017
Name: 1, dtype: object
greece  33585639
Spain   130803657
Year    2018
Name: 2, dtype: object
greece  34202053
Spain   135008823
Year    2019
Name: 3, dtype: object

```

Όνομα πίνακα : NIGHT_NON

Όνομα πίνακα : ARRIVALS_NON

```
greece    25038498
Spain     67728098
Year      2019
Name: 3, dtype: object
[(2016, 16916000.0, 61341839), (2017, 19068700.0, 65233045), (2018, 24320900.0, 65771059), (2019, 25038500.0, 67728098)]
[(2016, 16916000.0, 61341839), (2017, 19068700.0, 65233045), (2018, 24320900.0, 65771059), (2019, 25038500.0, 67728098)]
[(2016, 16916000.0, 61341839), (2017, 19068700.0, 65233045), (2018, 24320900.0, 65771059), (2019, 25038500.0, 67728098)]
[(2016, 16916000.0, 61341839), (2017, 19068700.0, 65233045), (2018, 24320900.0, 65771059), (2019, 25038500.0, 67728098)]
```


Όνομα πίνακα : NIGHT

```
Year      2019
Name: 3, dtype: object
[(2016, 110020000.0, 454957250), (2017, 119009000.0, 471199729), (2018, 142940000.0, 466940717), (2019, 143594000.0, 469813551)]
[(2016, 110020000.0, 454957250), (2017, 119009000.0, 471199729), (2018, 142940000.0, 466940717), (2019, 143594000.0, 469813551)]
[(2016, 110020000.0, 454957250), (2017, 119009000.0, 471199729), (2018, 142940000.0, 466940717), (2019, 143594000.0, 469813551)]
[(2016, 110020000.0, 454957250), (2017, 119009000.0, 471199729), (2018, 142940000.0, 466940717), (2019, 143594000.0, 469813551)]
90      91
```

Όνομα πίνακα : ARRIVALS

```
Spain      135008823
Year       2019
Name: 3, dtype: object
[(2016, 24996000.0, 123541778), (2017, 27211300.0, 129392382), (2018, 33585600.0, 130803657), (2019, 34202100.0, 135008823)]
[(2016, 24996000.0, 123541778), (2017, 27211300.0, 129392382), (2018, 33585600.0, 130803657), (2019, 34202100.0, 135008823)]
[(2016, 24996000.0, 123541778), (2017, 27211300.0, 129392382), (2018, 33585600.0, 130803657), (2019, 34202100.0, 135008823)]
[(2016, 24996000.0, 123541778), (2017, 27211300.0, 129392382), (2018, 33585600.0, 130803657), (2019, 34202100.0, 135008823)]
```

Το περιεχόμενο των csv αρχείων με την εξής σειρά :

Nights_non , Arrivals_non, Nights_total, Arrivals_total

```

    2016      2017      2018      2019
0  87912850  97034421  118876323  119971390
1  294556428  305907462  301022634  299091409

    2016      2017      2018      2019
0  16915996  19068697  24320893  25038498
1  61341839  65233045  65771059  67728098

    2016      2017      2018      2019
0  110020042  119009014  142940411  143594467
1  454957250  471199729  466940717  469813551

    2016      2017      2018      2019
0  24996038  27211268  33585639  34202053
1  123541778  129392382  130803657  135008823

Process finished with exit code 0

```

Επαλήθευση των δεδομένων (εμφάνιση πινάκων) και στην sql. Στο χαμrr εκτελέστηκαν τα εξής :

```
#mysql.exe -u root
```

```
SHOW DATABASES;
```

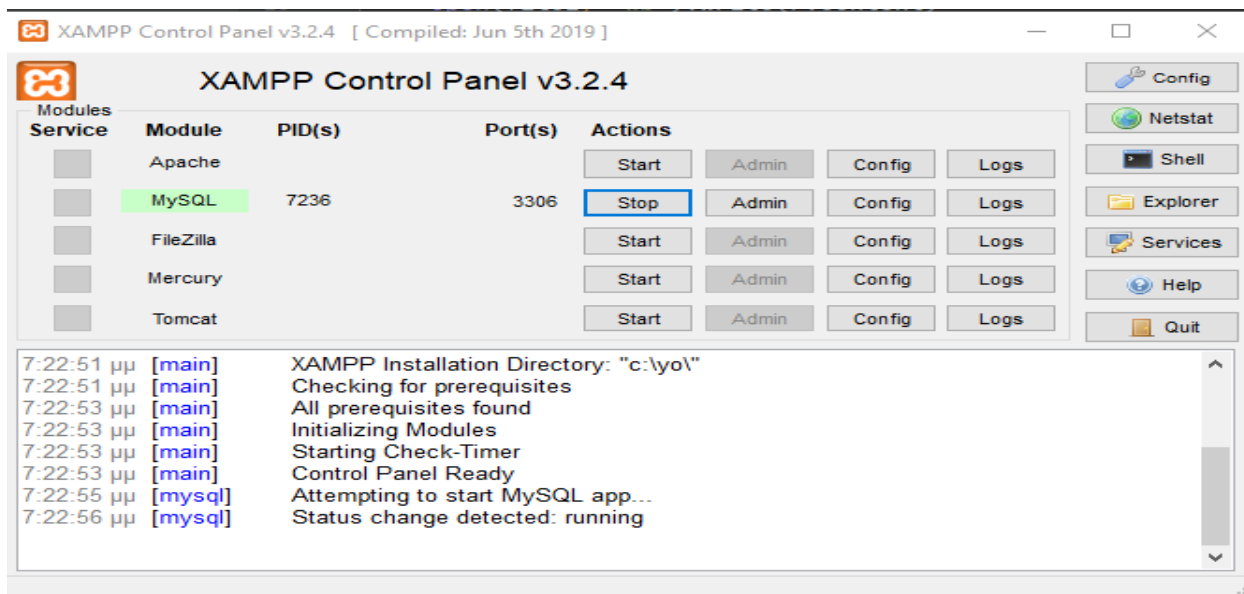
SHOW TABLES;

```
SELECT* FROM arrivals;
```

```
SELECT* FROM arrivals non;
```

```
SELECT* FROM night;
```

```
SELECT* FROM night non;
```



```

C:\> XAMPP for Windows - mysql.exe -u root

Tables_in_agp
+-----+
| arrivals |
| arrivals_non |
| night |
| night_non |
+-----+
4 rows in set (0.001 sec)

MariaDB [agp]> SELECT* FROM arrivals;
+-----+
| year | greece | spain |
+-----+
| 2016 | 24996000 | 123541778 |
| 2017 | 27211300 | 129392382 |
| 2018 | 33585600 | 130803657 |
| 2019 | 34202100 | 135008823 |
+-----+
4 rows in set (0.001 sec)

MariaDB [agp]> SELECT* FROM arrivals_non;
+-----+
| year | greece | spain |
+-----+
| 2016 | 16916000 | 61341839 |
| 2017 | 19068700 | 65233045 |
| 2018 | 24320900 | 65771059 |
| 2019 | 25038500 | 67728098 |
+-----+
4 rows in set (0.000 sec)

MariaDB [agp]> SELECT* FROM night_non;
+-----+
| year | greece | spain |
+-----+
| 2016 | 87912800 | 294556428 |
| 2017 | 97034400 | 305907462 |
| 2018 | 118876000 | 301022634 |
| 2019 | 119971000 | 299091409 |
+-----+
4 rows in set (0.000 sec)

MariaDB [agp]> SELECT* FROM night;
+-----+
| year | greece | spain |
+-----+
| 2016 | 110020000 | 454957250 |
| 2017 | 119009000 | 471199729 |
| 2018 | 142940000 | 466940717 |
| 2019 | 143594000 | 469813551 |
+-----+

```

Δημιουργία των csv files μετά την εκτέλεση του προγράμματος(nights_non, arrivals_non, nights_total, arrivals_total)

