

16/06/2024

Προαιρετική Εργαστηριακή Άσκηση Python 2024

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ	ΕΤΟΣ	EMAIL
ΔΗΜΗΤΡΙΟΣ ΤΣΑΜΠΡΑΣ	1072467	5 ^ο	up1072467@ac.upatras.gr

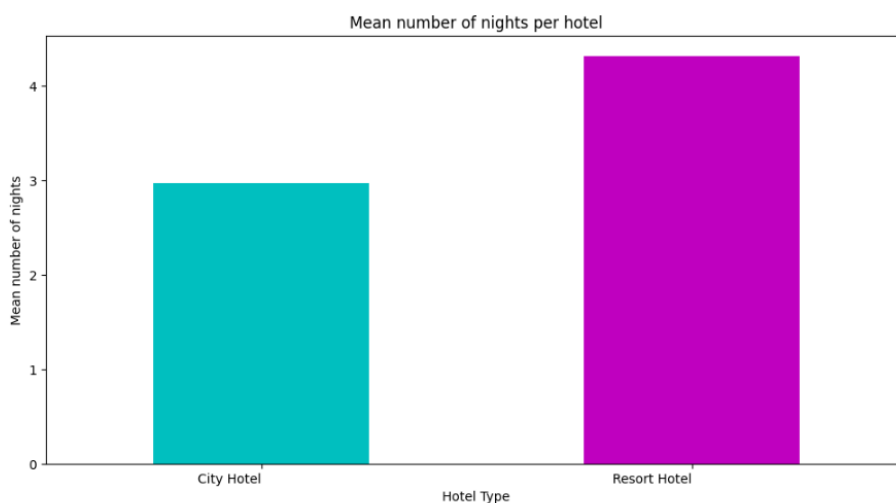
Πίνακας περιεχομένων

Γραφήματα Εργασίας	2
G.U.I	5
Βάση Δεδομένων.....	9
Σχόλια-Παραδοχές.....	12
Κώδικας.....	12

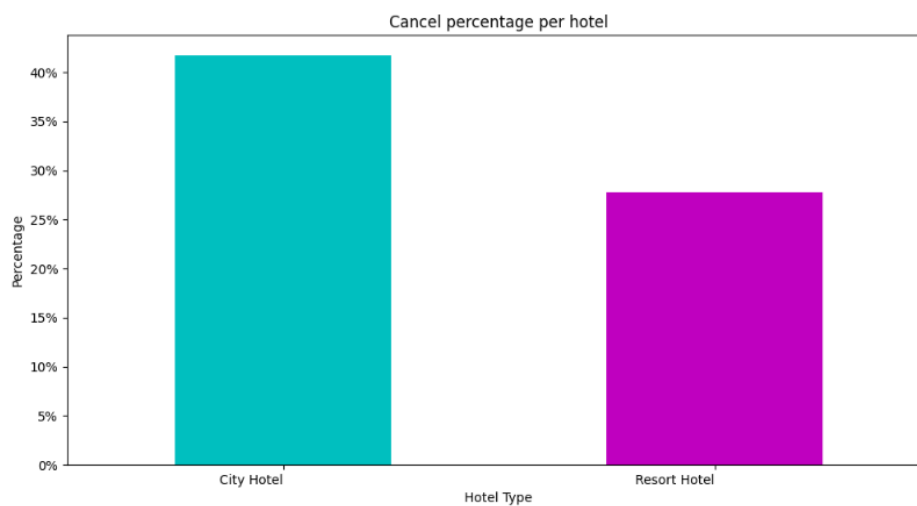
Γραφήματα Εργασίας

Τα ζητούμενα γραφήματα παρουσιάζονται παρακάτω μαζί με κάθε ερώτημα της εργασίας:

1. Ποια είναι τα βασικά στατιστικά στοιχεία για τις κρατήσεις σε κάθε ξενοδοχείο (μέσος όρος διανυκτερεύσεων, ποσοστό ακυρώσεων)?

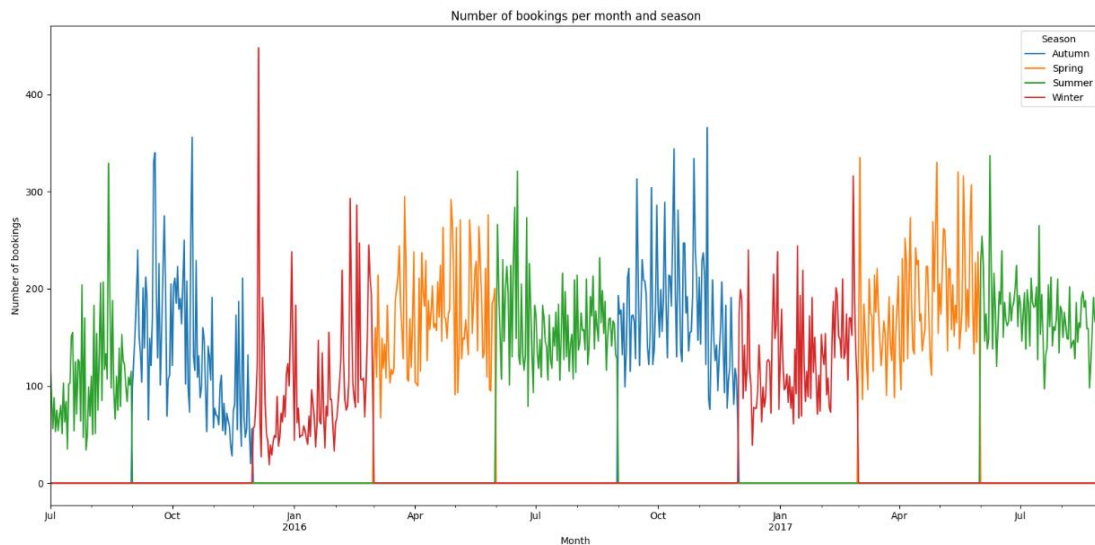


Με βάση αυτό το γράφημα, παρατηρούμε ότι για το City Hotel ο μέσος όρος διανυκτερεύσεων κυμαίνεται περίπου στο 3.0, ενώ για το Resort Hotel περίπου στο 4.32.



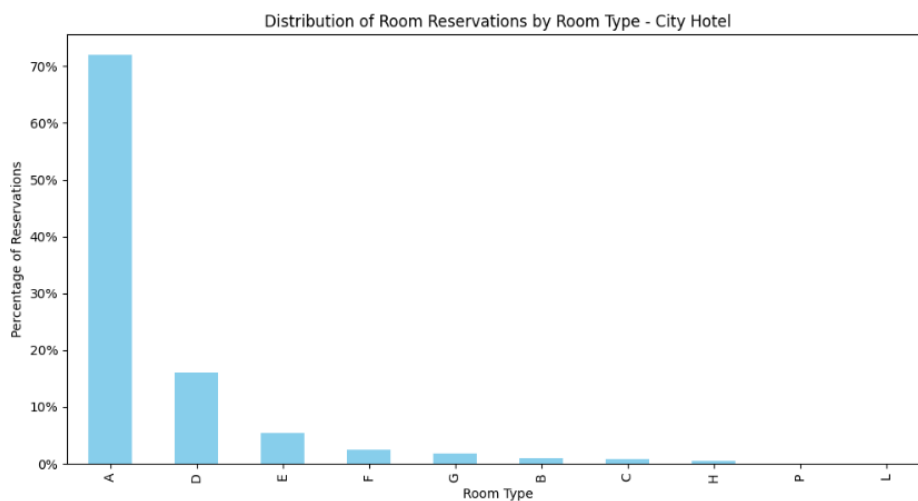
Με βάση αυτό το γράφημα, παρατηρούμε ότι για το City Hotel το ποσοστό ακυρώσεων κυμαίνεται περίπου στο 42%, ενώ για το Resort Hotel περίπου στο 28%.

2. Ποια είναι η κατανομή των κρατήσεων ανά μήνα και ανά εποχή;



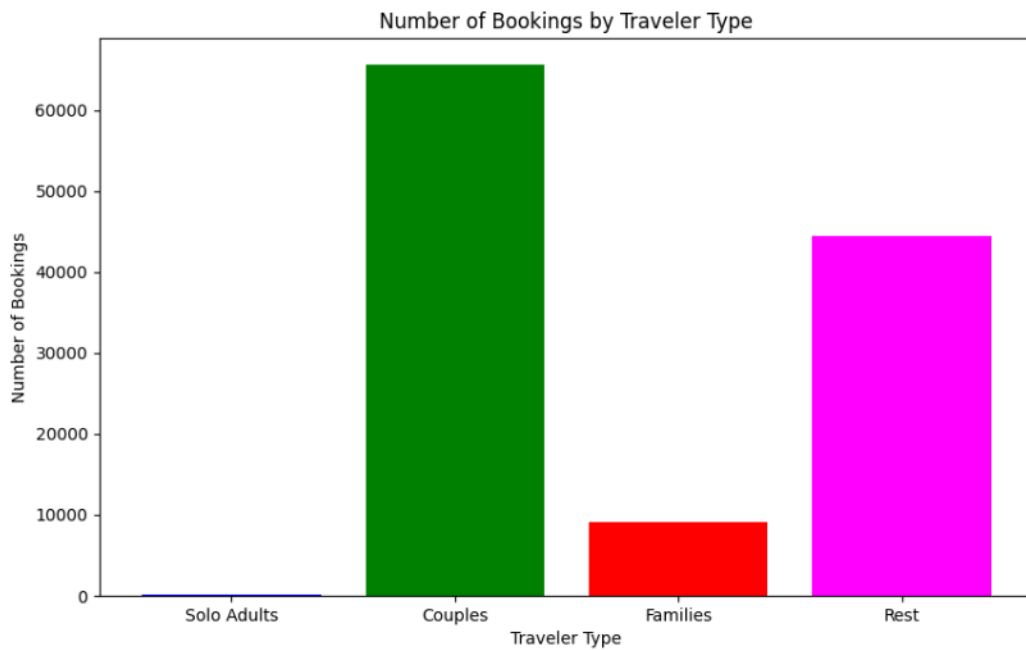
Με βάση αυτό το γράφημα, παρατηρούμε ότι τον χειμώνα του 2016, οι κρατήσεις είναι περισσότερες σε σχέση με τις άλλες εποχές.

3. Ποια είναι η κατανομή των κρατήσεων ανά τύπο δωματίου;



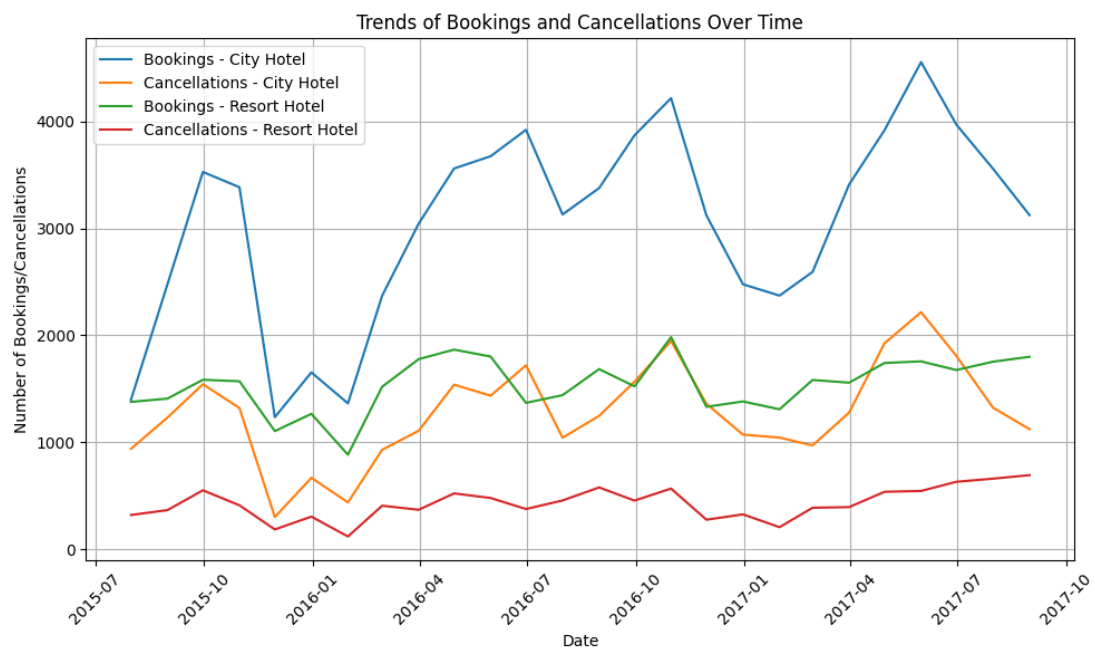
Με βάση αυτό το γράφημα, παρατηρούμε ότι ο επικρατέστερος τύπος δωματίου είναι ο 'Α'.

4. Πόσες κρατήσεις αφορούν οικογένειες, ζευγάρια ή μεμονωμένους ταξιδιώτες;



Με βάση αυτό το γράφημα, η πλειοψηφία των κρατήσεων αφορά τα ζευγάρια.

5. Ποιες είναι οι τάσεις των κρατήσεων με την πάροδο του χρόνου;



Με βάση το παραπάνω γράφημα, φαίνονται οι τάσεις των κρατήσεων με την πάροδο του χρόνου.

6. Υπάρχει εποχικότητα στις κρατήσεις ή στις ακυρώσεις σε κάθε ξενοδοχείο;

Από το γράφημα στο ερώτημα 5, μπορεί να απαντηθεί το συγκεκριμένο ερώτημα: Οι περισσότερες κρατήσεις έγιναν για το City Hotel, αλλά παράλληλα έγιναν και οι περισσότερες ακυρώσεις. Στο Resort Hotel οι κρατήσεις ήταν επίσης περισσότερες από τις ακυρώσεις.

G.U.I

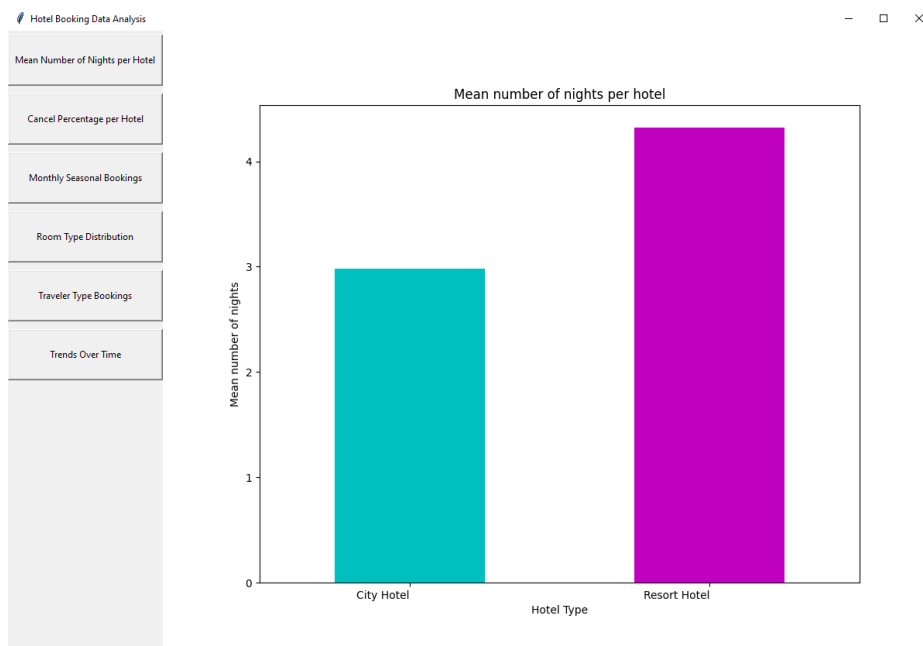
Τα παραπάνω διαγράμματα, που αποτελούν τις απαντήσεις κάθε ερώτησης, έχουν εισαχθεί μέσα στο G.U.I. Παρακάτω υπάρχουν τα screenshots από το γραφικό περιβάλλον:

- Όταν εκτελείται ο κώδικας, εμφανίζεται η αρχική οθόνη του γραφικού περιβάλλοντος:

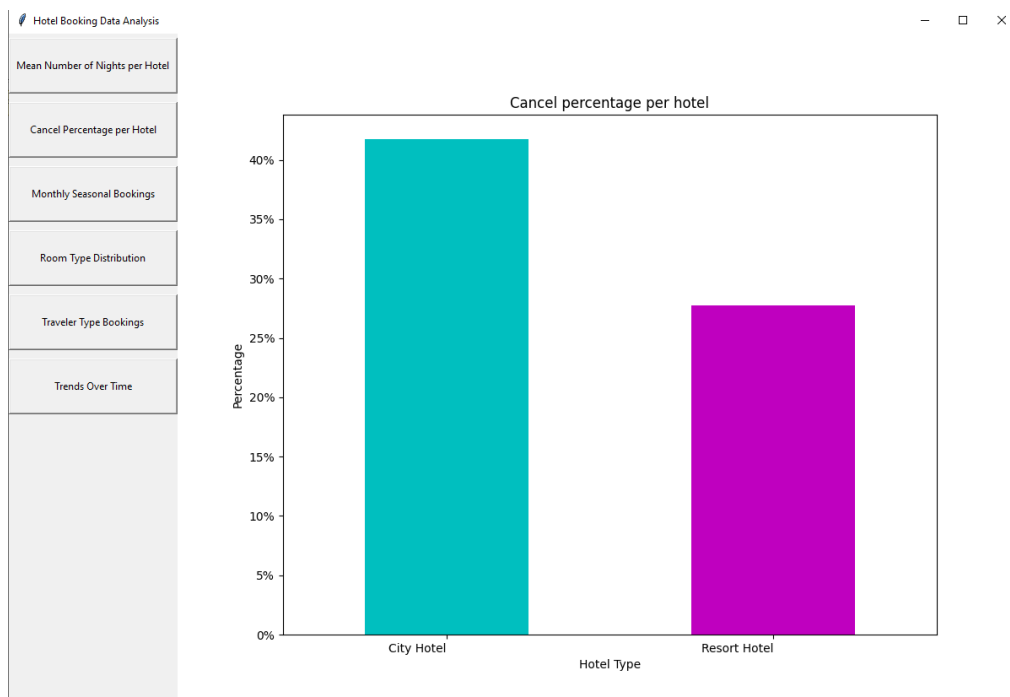


Figure 1: Αρχική Οθόνη

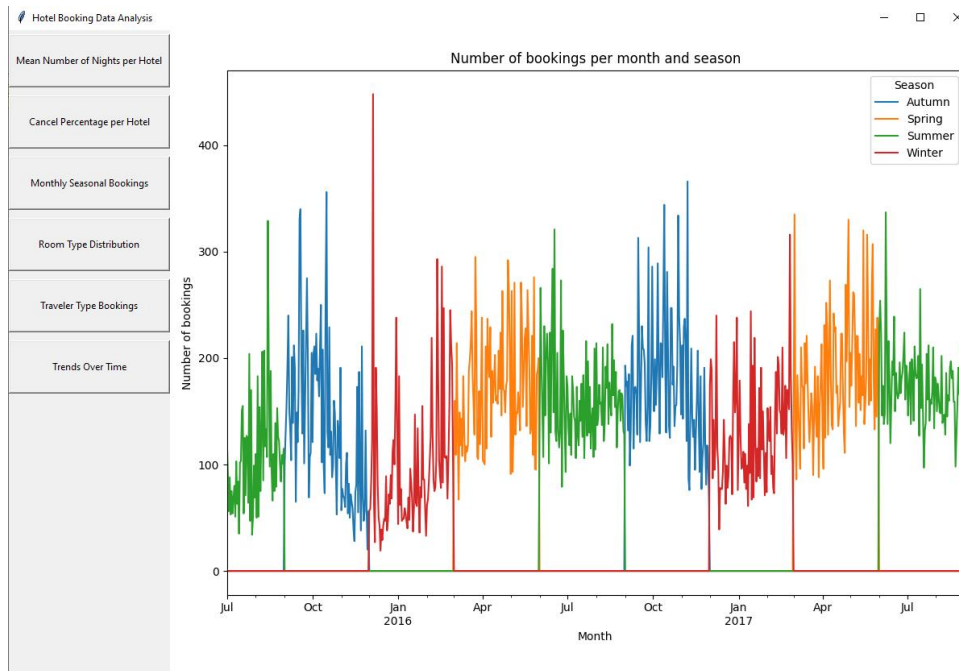
- Πατάμε το κουμπί **Mean Number of Nights per Hotel** και εμφανίζεται το αντίστοιχο γράφημα:



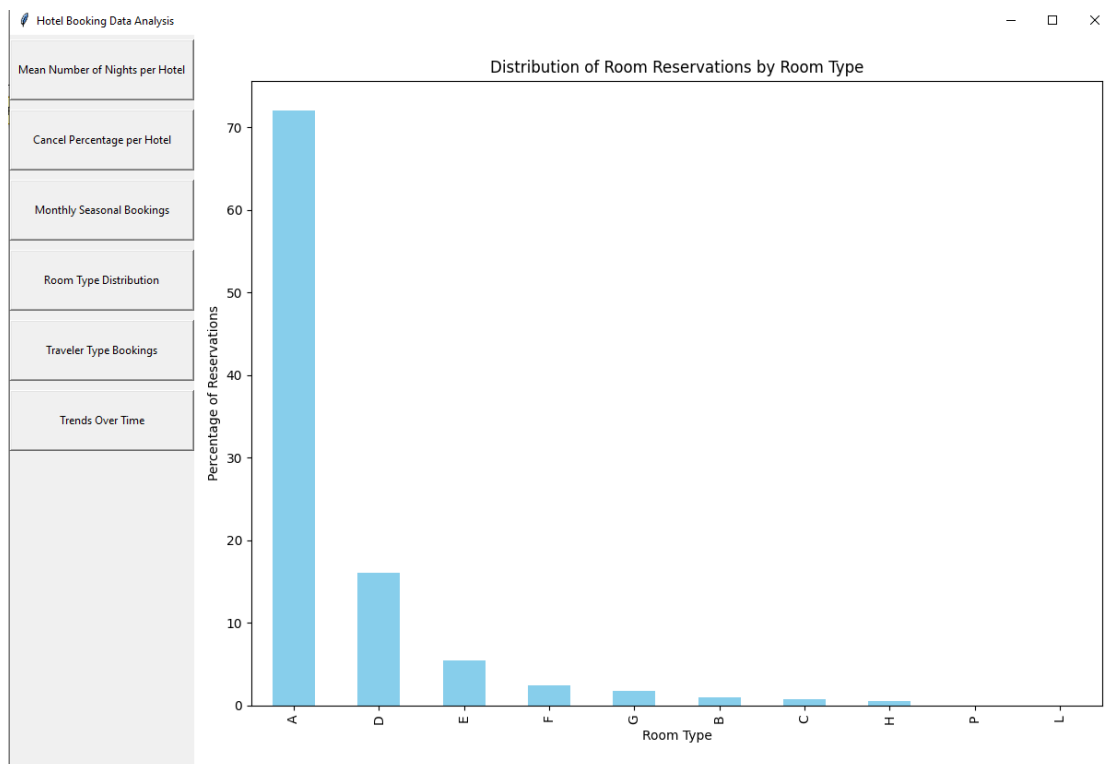
- Πατάμε το κουμπί **Cancel Percentage per Hotel** και εμφανίζεται το αντίστοιχο γράφημα:



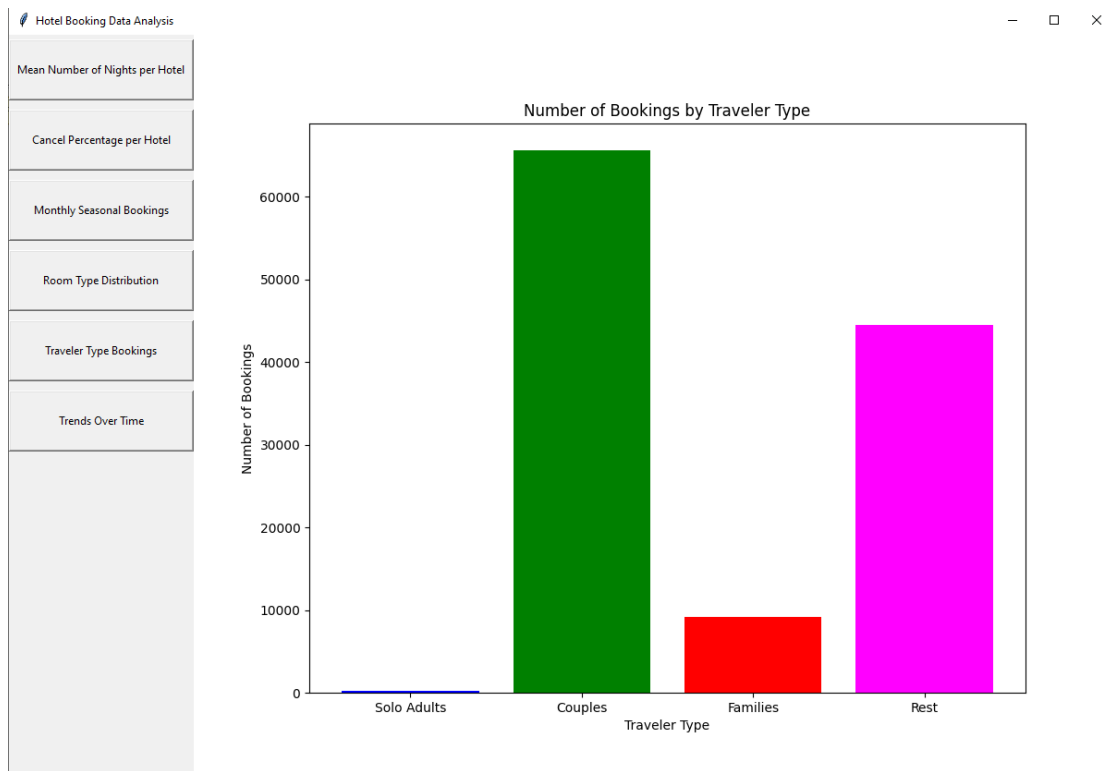
- Πατάμε το κουμπί **Monthly Seasonal Bookings** και εμφανίζεται το αντίστοιχο γράφημα:



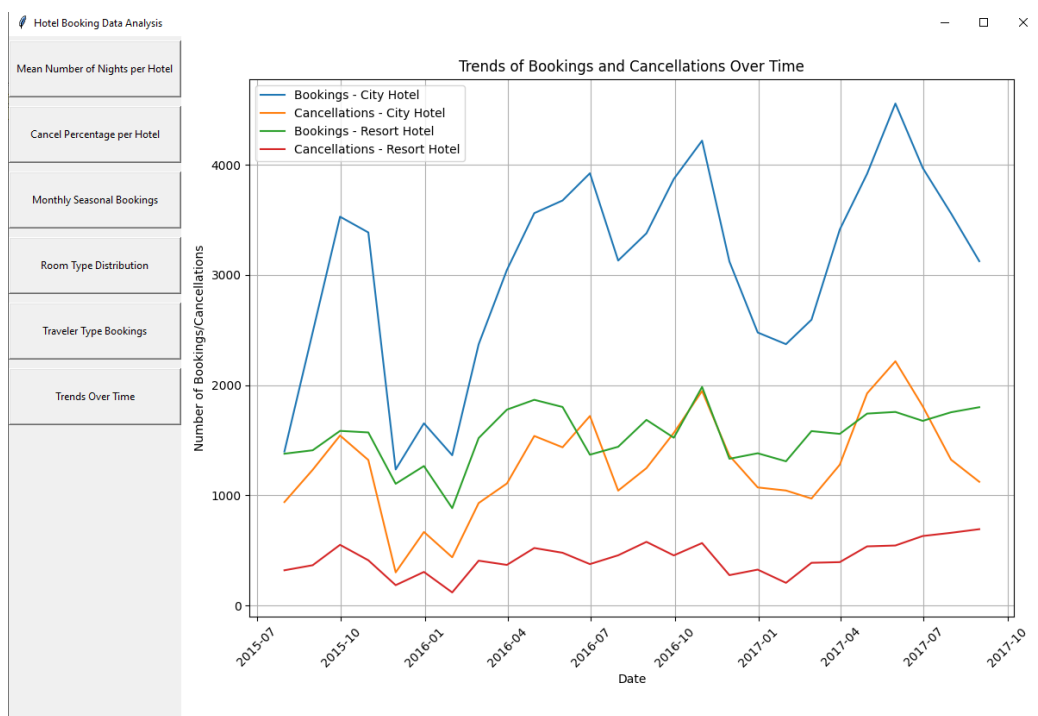
- Πατάμε το κουμπί **Room Type Distribution** και εμφανίζεται το αντίστοιχο γράφημα:



- Πατάμε το κουμπί **Traveler Type Bookings** και εμφανίζεται το αντίστοιχο γράφημα:



- Πατάμε το κουμπί **Trends Over Time** και εμφανίζεται το αντίστοιχο γράφημα:



Βάση Δεδομένων

- Με την εκτέλεση του παρακάτω κώδικα , δημιουργείται η Βάση Δεδομένων με όνομα **hotel_booking**:

```
# Create the Database
mydb = mysql.connector.connect (
    host="localhost",
    user="root",
    passwd="",
)

mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE hotel_booking")
```



Database
hotel_booking
information_schema
mysql
performance_schema
trip_project2023

- Με την εκτέλεση του παρακάτω κώδικα , δημιουργούνται οι πίνακες για κάθε αποτέλεσμα μαζί με τα πεδία κάθε πίνακα:

```
# Create the Database
mydb = mysql.connector.connect (
    host="localhost",
    user="root",
    passwd="",
    database="hotel_booking"
)

# Create the tables
mycursor = mydb.cursor()
mycursor.execute("CREATE TABLE mean_nights_per_hotel (hotel VARCHAR(255), mean_nights DECIMAL(10, 2))")
mycursor.execute("CREATE TABLE cancel_percentage_per_hotel (hotel VARCHAR(255), cancel_percentage DECIMAL(10, 2))")
mycursor.execute("CREATE TABLE monthly_seasonal_bookings (month VARCHAR(255), season VARCHAR(255), bookings INT)")
mycursor.execute("CREATE TABLE room_type_distribution (room_type VARCHAR(255), count INT)")
mycursor.execute("CREATE TABLE traveler_type_bookings (traveler_type VARCHAR(255), count INT)")
mycursor.execute("CREATE TABLE trends_over_time (hotel VARCHAR(255), date DATE, bookings INT, cancellations INT)")
```

Tables_in_hotel_booking
cancel_percentage_per_hotel
mean_nights_per_hotel
monthly_seasonal_bookings
room_type_distribution
traveler_type_bookings
trends_over_time

- Με την εκτέλεση του παρακάτω κώδικα , εισάγονται τα αποτελέσματα ως εγγραφές σε κάθε πίνακα και εμφανίζονται όλες οι εγγραφές για κάθε πίνακα στο terminal της python:

```
# Insert data into mean nights per hotel table
for hotel, mean_nights in mean_nights_per_hotel.items():
    mycursor.execute("INSERT INTO mean_nights_per_hotel (hotel,
mean_nights) VALUES (%s, %s)", (hotel, mean_nights))

# Insert data into cancel_percentage_per_hotel table
for hotel, cancel_percentage in cancel_percentage_per_hotel.items():
    mycursor.execute("INSERT INTO cancel_percentage_per_hotel (hotel,
cancel_percentage) VALUES (%s, %s)", (hotel, cancel_percentage))

# Insert data into monthly_seasonal_bookings table
for (month, season), bookings in
monthly_seasonal_bookings.stack().items():
    mycursor.execute("INSERT INTO monthly_seasonal_bookings (month,
season, bookings) VALUES (%s, %s, %s)", (month.strftime('%Y-%m-%d'),
season, bookings))

# Insert data into room_type_distribution table
for room_type, count in room_type_distribution.items():
    mycursor.execute("INSERT INTO room_type_distribution (room_type,
count) VALUES (%s, %s)", (room_type, count))

# Insert data into traveler_type_bookings table
for traveler_type, count in zip(categories, counts):
    mycursor.execute("INSERT INTO traveler_type_bookings
(traveler_type, count) VALUES (%s, %s)", (traveler_type, count))

# select * from mean_nights_per_hotel
mycursor = mydb.cursor()
mycursor.execute("SELECT hotel, mean_nights FROM
mean_nights_per_hotel")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from cancel_percentage_per_hotel
mycursor = mydb.cursor()
mycursor.execute("SELECT hotel, cancel_percentage FROM
cancel_percentage_per_hotel")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from monthly_seasonal_bookings
mycursor = mydb.cursor()
```

```

mycursor.execute("SELECT month, season, bookings FROM
monthly_seasonal_bookings")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from room_type_distribution
mycursor = mydb.cursor()
mycursor.execute("SELECT room_type, count FROM
room_type_distribution")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from traveler_type_bookings
mycursor = mydb.cursor()
mycursor.execute("SELECT traveler_type, count FROM
traveler_type_bookings")
myresult = mycursor.fetchall()
for x in myresult: print(x)

```

hotel	mean_nights
City Hotel	2.98
Resort Hotel	4.32

Figure 2: Μέσος όρος διανυκτερεύσεων

hotel	cancel_percentage
City Hotel	41.73
Resort Hotel	27.76

Figure 3: Ποσοστό ακυρώσεων

month	season	bookings
2015-07-01	Autumn	0
2015-07-01	Spring	0
2015-07-01	Summer	122
2015-07-01	Winter	0
2015-07-02	Autumn	0
2015-07-02	Spring	0
2015-07-02	Summer	93
2015-07-02	Winter	0
2015-07-03	Autumn	0
2015-07-03	Spring	0
2015-07-03	Summer	56
2015-07-03	Winter	0
2015-07-04	Autumn	0
2015-07-04	Spring	0

Figure 4: Κατανομή κρατήσεων ανά μήνα και εποχή

room_type	count
A	85994
D	19201
E	6535
F	2897
G	2094
B	1118
C	932
H	601
P	12
L	6

Figure 5: : Κατανομή κρατήσεων ανά τύπο δωματίου

traveler_type	count
Solo Adults	192
Couples	65600
Families	9117
Rest	44481

Figure 6: Κρατήσεις βάσει το είδος ταξιδιώτη

Σχόλια-Παραδοχές

- Η υλοποίηση του προγράμματος έγινε με το εργαλείο PyCharm.
- Ο πίνακας **trends_over_time** στην MySQL δεν περιέχει εγγραφές, αλλά έχει εξαχθεί ως κενός πίνακας.
- Η επεξεργασία της Βάσης Δεδομένων έγινε αποκλειστικά με την Python.
- Το dataset “Hotel Booking” διαβάζεται ως κατεβασμένο αρχείο του υπολογιστή και δεν διαβάζεται από την ιστοσελίδα.

Κώδικας

Παρακάτω βρίσκεται ο ζητούμενος τελικός κώδικας:

```
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import ticker as mtick
```

```

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import tkinter as tk
from tkinter import Button
import mysql.connector

# csv file root
file_root = "E:/Users/Δημήτρης
Τσάμπρας/Desktop/Projects_Αρχές_Γλωσσών/Python
Project/hotel_booking.csv"

# read the csv
data = pd.read_csv(file_root)

# create column total number of nights
data['total_nights'] = data['stays_in_weekend_nights'] +
data['stays_in_week_nights']

# mean number of nights per hotel
mean_nights_per_hotel = data.groupby('hotel')['total_nights'].mean()

# mean cancel percentage per hotel
cancel_percentage_per_hotel =
(data.groupby('hotel')['is_canceled'].mean()) * 100

# Construct a new column 'arrival_date' combining year, month, and
day columns with - between
data['arrival_date'] = pd.to_datetime(
    data['arrival_date_year'].astype(str) + '-' +
data['arrival_date_month'].astype(str) + '-' + data[
    'arrival_date_day_of_month'].astype(str))

# Define the seasons
seasons = {
    'Spring': [3, 4, 5],
    'Summer': [6, 7, 8],
    'Autumn': [9, 10, 11],
    'Winter': [12, 1, 2]
}

# Construct a new column for season based on arrival month
data['season'] = data['arrival_date'].dt.month.map(
    lambda month: next(season for season, months in seasons.items()
if month in months)) # check the number and
# define its season

# Group by month and season and count the bookings
monthly_seasonal_bookings = data.groupby(['arrival_date',
'season']).size().unstack(fill_value=0)

# load the distribution
room_type_distribution = data['reserved_room_type'].value_counts()

# Calculate total number of reservations
total_reservations = room_type_distribution.sum()

# Calculate the percentage of the room distribution
room_type_percentage = (room_type_distribution / total_reservations)
* 100

# define the kinds of customers
couple = (data['adults'] == 2).sum() & (data['children'] == 0).sum()

```

```

& (data['babies'] == 0).sum()
solo_adult = (data['adults'] == 1).sum() & (data['children'] ==
0).sum() & (data['babies'] == 0).sum()
families = (data['adults'] >= 2).sum() & (data['children'] >=
1).sum() | (data['babies'] >= 1).sum()
rest = data.shape[0] - (couple + solo_adult + families)

# Categories
categories = ['Solo Adults', 'Couples', 'Families', 'Rest']

# put the kinds in a list
counts = [solo_adult, couple, families, rest]

# Group data by hotel
hotels_grouped = data.groupby('hotel')

#----- Functions for G.U.I-----

def plot_mean_nights():
    fig, axes = plt.subplots(figsize=(10, 6))
    mean_nights_per_hotel.plot(kind='bar', color=['c', 'm'], ax=axes)
    axes.set_title("Mean number of nights per hotel")
    axes.set_ylabel("Mean number of nights")
    axes.set_xlabel('Hotel Type')
    plt.setp(axes.get_xmajorticklabels(), rotation=0, ha='right')
    show_plot(fig)

def plot_cancel_percentage():
    fig, axes = plt.subplots(figsize=(10, 6))
    cancel_percentage_per_hotel.plot(kind='bar', color=['c', 'm'],
ax=axes)
    axes.set_title("Cancel percentage per hotel")
    axes.set_ylabel("Percentage")
    axes.set_xlabel('Hotel Type')
    plt.setp(axes.get_xmajorticklabels(), rotation=0, ha='right')

    # put the percentage in the y label
    percent_formatter = mtick.FuncFormatter(lambda y, _:
'{:.0f}%'.format(y))
    axes.yaxis.set_major_formatter(percent_formatter)
    show_plot(fig)

def plot_monthly_seasonal_bookings():
    fig, ax = plt.subplots(figsize=(10, 6))
    monthly_seasonal_bookings.plot(kind='line', ax=ax)
    ax.set_title('Number of bookings per month and season')
    ax.set_xlabel('Month')
    ax.set_ylabel('Number of bookings')
    ax.legend(title='Season')
    plt.tight_layout()
    show_plot(fig)

def plot_room_type_percentage():
    fig, ax = plt.subplots(figsize=(10, 6))
    room_type_percentage.plot(kind='bar', color='skyblue', ax=ax)
    ax.set_title('Distribution of Room Reservations by Room Type')
    ax.set_xlabel('Room Type')

```

```

ax.set_ylabel('Percentage of Reservations')
plt.tight_layout()
show_plot(fig)

def plot_traveler_type():
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.bar(categories, counts, color=['blue', 'green', 'red',
'magenta'])
    ax.set_title('Number of Bookings by Traveler Type')
    ax.set_xlabel('Traveler Type')
    ax.set_ylabel('Number of Bookings')
    show_plot(fig)

def plot_trends_over_time():
    fig, ax = plt.subplots(figsize=(10, 6))
    for hotel, hotel_data in hotels_grouped:
        monthly_bookings =
hotel_data.groupby(pd.Grouper(key='arrival_date',
freq='M'))['hotel'].count()
        monthly_cancellations = \
            hotel_data[hotel_data['is_canceled'] ==
1].groupby(pd.Grouper(key='arrival_date', freq='M'))[
                'hotel'].count()

        ax.plot(monthly_bookings.index, monthly_bookings.values,
label=f'Bookings - {hotel}')
        ax.plot(monthly_cancellations.index,
monthly_cancellations.values, label=f'Cancellations - {hotel}')

        ax.set_title('Trends of Bookings and Cancellations Over Time')
        ax.set_xlabel('Date')
        ax.set_ylabel('Number of Bookings/Cancellations')
        ax.legend()
        ax.grid(True)
        plt.xticks(rotation=45)
        plt.tight_layout()
        show_plot(fig)

def show_plot(fig):
    for widget in plot_container.winfo_children():
        widget.destroy()

    canvas = FigureCanvasTkAgg(fig, master=plot_container)
    canvas.draw()
    canvas.get_tk_widget().pack(fill=tk.BOTH, expand=True)

# function to export the results in the csv
def export_tables_to_csv(table_names):
    for table in table_names:
        query = f"SELECT * FROM {table}"
        mycursor.execute(query)
        result = mycursor.fetchall()
        columns = [i[0] for i in mycursor.description]

        df = pd.DataFrame(result, columns=columns)
        df.to_csv(f"{table}.csv", index=False)
        print(f"Table {table} exported to {table}.csv")

```



```

# Create the GUI
root = tk.Tk()
root.title("Hotel Booking Data Analysis")
root.geometry("1200x800") # Increase the size of the main window

plot_frame = tk.Frame(root)
plot_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

plot_container = tk.Frame(plot_frame) # Create a separate frame for
the plot
plot_container.pack(fill=tk.BOTH, expand=True)

button_frame = tk.Frame(root)
button_frame.pack(side=tk.LEFT, fill=tk.Y)

buttons = [
    ("Mean Number of Nights per Hotel", plot_mean_nights),
    ("Cancel Percentage per Hotel", plot_cancel_percentage),
    ("Monthly Seasonal Bookings", plot_monthly_seasonal_bookings),
    ("Room Type Distribution", plot_room_type_percentage),
    ("Traveler Type Bookings", plot_traveler_type),
    ("Trends Over Time", plot_trends_over_time)
]

border_colors = ['red', 'green', 'blue', 'yellow', 'magenta', 'cyan']

# Parameters for the buttons
for i, (text, func) in enumerate(buttons):
    button = Button(button_frame, text=text, command=func, height=3,
width=25,
                    padx=10, pady=5,
highlightbackground=border_colors[i], highlightthickness=2)
    button.pack(fill='x', pady=5)

# Start the GUI
root.mainloop()

# MySQL parameters
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database=""
)

mycursor = mydb.cursor()

# Create the database
mycursor.execute("CREATE DATABASE Hotel_booking")

# Create the tables
mycursor.execute("CREATE TABLE mean_nights_per_hotel (hotel
VARCHAR(255), mean_nights DECIMAL(10, 2))")
mycursor.execute("CREATE TABLE cancel_percentage_per_hotel (hotel
VARCHAR(255), cancel_percentage DECIMAL(10, 2))")
mycursor.execute("CREATE TABLE monthly_seasonal_bookings (month
VARCHAR(255), season VARCHAR(255), bookings INT)")
mycursor.execute("CREATE TABLE room_type_distribution (room_type
VARCHAR(255), count INT)")

```

```

mycursor.execute("CREATE TABLE traveler_type_bookings (traveler_type
VARCHAR(255), count INT)")
mycursor.execute("CREATE TABLE trends_over_time (hotel
VARCHAR(255), date DATE, bookings INT, cancellations INT)")

# Insert data into mean_nights_per_hotel table
for hotel, mean_nights in mean_nights_per_hotel.items():
    mycursor.execute("INSERT INTO mean_nights_per_hotel (hotel,
mean_nights) VALUES (%s, %s)", (hotel, mean_nights))

# Insert data into cancel_percentage_per_hotel table
for hotel, cancel_percentage in cancel_percentage_per_hotel.items():
    mycursor.execute("INSERT INTO cancel_percentage_per_hotel (hotel,
cancel_percentage) VALUES (%s, %s)",
                    (hotel, cancel_percentage))

# Insert data into monthly_seasonal_bookings table
for (month, season), bookings in
monthly_seasonal_bookings.stack().items():
    mycursor.execute("INSERT INTO monthly_seasonal_bookings (month,
season, bookings) VALUES (%s, %s, %s)",
                    (month.strftime('%Y-%m-%d'), season, bookings))

# Insert data into room_type_distribution table
for room_type, count in room_type_distribution.items():
    mycursor.execute("INSERT INTO room_type_distribution (room_type,
count) VALUES (%s, %s)", (room_type, count))

# Insert data into traveler_type_bookings table
for traveler_type, count in zip(categories, counts):
    mycursor.execute("INSERT INTO traveler_type_bookings
(traveler_type, count) VALUES (%s, %s)",
                    (traveler_type, int(count)))

# select * from mean_nights_per_hotel
mycursor = mydb.cursor()
mycursor.execute("SELECT hotel, mean_nights FROM
mean_nights_per_hotel")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from cancel_percentage_per_hotel
mycursor = mydb.cursor()
mycursor.execute("SELECT hotel, cancel_percentage FROM
cancel_percentage_per_hotel")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from monthly_seasonal_bookings
mycursor = mydb.cursor()
mycursor.execute("SELECT month, season, bookings FROM
monthly_seasonal_bookings")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# select * from room_type_distribution
mycursor = mydb.cursor()
mycursor.execute("SELECT room_type, count FROM
room_type_distribution")
myresult = mycursor.fetchall()
for x in myresult: print(x)

```

```

# select * from traveler_type_bookings
mycursor = mydb.cursor()
mycursor.execute("SELECT traveler_type, count FROM
traveler_type_bookings")
myresult = mycursor.fetchall()
for x in myresult: print(x)

# Define tables to export
tables_to_export = ["mean_nights_per_hotel",
"cancel_percentage_per_hotel", "monthly_seasonal_bookings",
"room_type_distribution",
"traveler_type_bookings", "trends_over_time"]

# Export tables to CSV
export_tables_to_csv(tables_to_export)

# Commit changes and close the database
mydb.commit()
mycursor.close()
mydb.close()

```