Airline Data Challenge

Data Visualization

Graphs Generated in Python

Visuals.py

Divya Jayaprakash Koneri

File: Visuals.py

Note: Below listed graphs are in the sequence as in the Visual.py file

Graph [1]:

Title: High Profit To Cost Ratio Graph

Objective: Identify the route with the highest return on investment(cost)

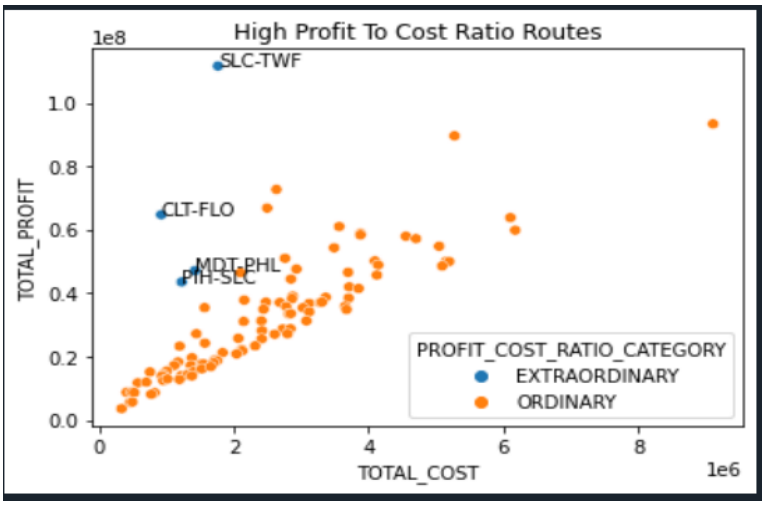
Corresponds To: Data Challenge question#3 - The 5 round trip routes that you recommend to invest in based on any factors that you choose.

Result: CLT-FLO is the route with high return on cost with a profit to cost ratio of 70.23. This is one of the 5 round-trip routes that I recommend for the airline venture

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk


print('Question#3 - Profit to Cost Ratio Graph')
#Draw the graph

routes_profit_cost_ratio = pd.read_excel("C://Users//...//Capital One Analytics//data4//Profitable Roundtrip Routes.xlsx")
routes_profit_cost_ratio = routes_profit_cost_ratio.sort_values(['PROFIT_COST_RATIO'], ascending=[False])
routes_profit_cost_ratio = routes_profit_cost_ratio.head(100)
value=(routes_profit_cost_ratio['PROFIT_COST_RATIO']>30)
routes_profit_cost_ratio['PROFIT_COST_RATIO_CATEGORY']= np.where( value==True , "EXTRAORDINARY", "ORDINARY")

routes = routes_profit_cost_ratio['ROUTE'].to_list()
total_cost = routes_profit_cost_ratio['TOTAL_COST'].to_list()
total_profit = routes_profit_cost_ratio['TOTAL_PROFIT'].to_list()
profit_cost_ratio_category = routes_profit_cost_ratio['PROFIT_COST_RATIO_CATEGORY'].to_list()
list_route_category = (routes_profit_cost_ratio['ROUTE']+"||"+routes_profit_cost_ratio['PROFIT_COST_RATIO_CATEGORY']).to_list()
# Plot
ax = sns.scatterplot(data=routes_profit_cost_ratio, x="TOTAL_COST", y="TOTAL_PROFIT", hue="PROFIT_COST_RATIO_CATEGORY")
for i, route_category in enumerate(list_route_category):
    if ("EXTRAORDINARY" in route_category):
       ax.text(total_cost[i], total_profit[i], route_category[0:7])

plt.title("High Profit To Cost Ratio Routes")
plt.show()

#*****************************************************************************************************************************************
```

Graph Output: High profit to cost ratio route is CLT-FLO



Graph [2]:

Title: Short Distance Route with High Occupancy, High Profit To Cost Ratio and Quick Breakeven Graph.

Objective: Identify the best route in terms of multiple factors such as the distance, occupancy, profit to cost ratio and breakeven

Corresponds To: Data Challenge question#3 - The 5 round trip routes that you recommend to invest in based on any factors that you choose.

Result: MDT-PHL is a short distance route with high occupancy, high profit to cost ratio and quick breakeven. This is one of the 5 round-trip routes that I recommend for the airline venture

Code:

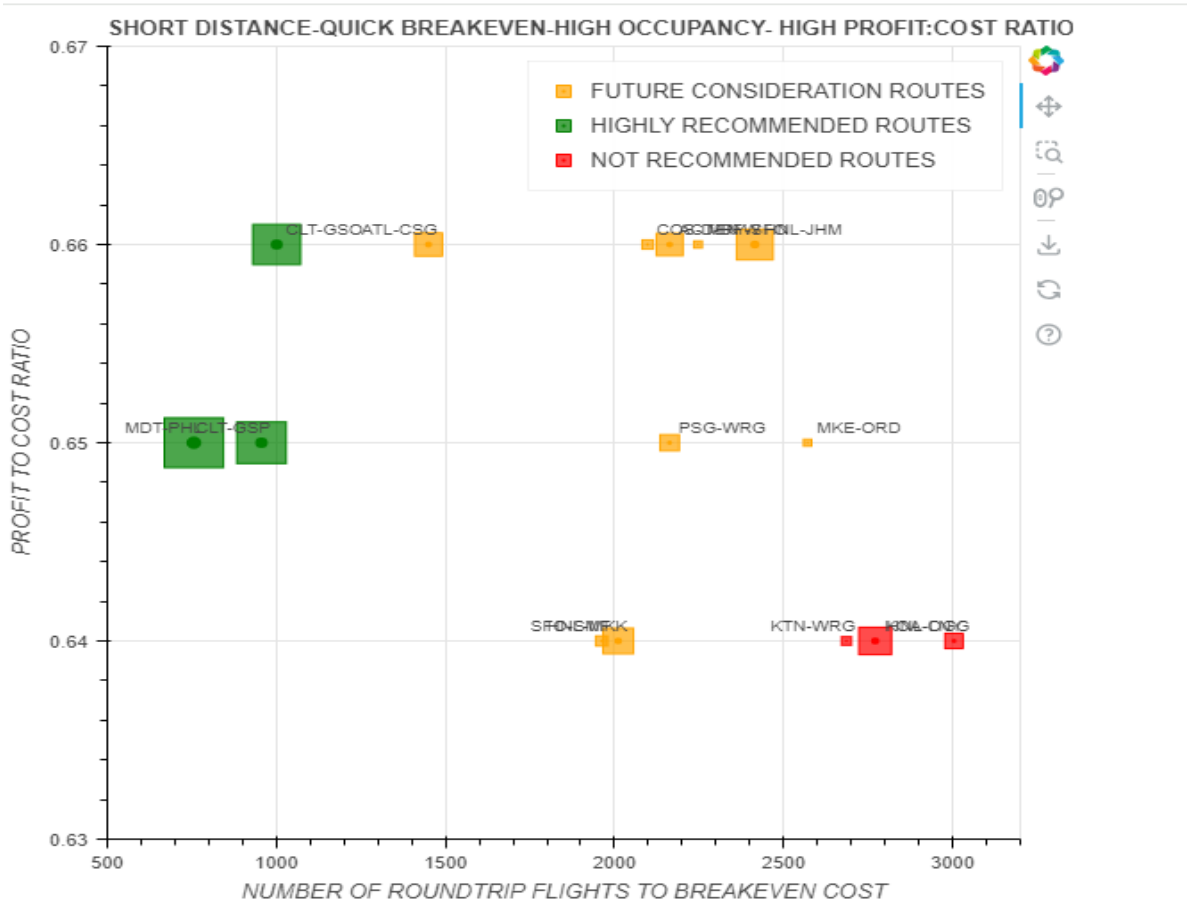
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns

print('Question#3 - SHORT DISTANCE FLIGHT WITH HIGH OCCUPANCY,HIGH PROFIT TO COST RATIO AND QUICK BREAKEVEN Graph')
#Draw the graph

from bokeh.plotting import figure, output_file, show
from bokeh.models import Range1d, ColumnDataSource, LabelSet

routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Profitable Roundtrip Routes.xlsx")
routes_data = routes_data.sort_values(['ROUTE_DISTANCE'], ascending=[True])
routes_data = routes_data.head(15)
routes = routes_data['ROUTE'].to_list()
breakeven_flights = routes_data['BREAKEVEN_FLIGHTS'].to_list()
profit_cost_ratio = routes_data['PROFIT_COST_RATIO'].to_list()
avg_occupancy =round((routes_data['AVG_OCCUPANCY_RATE_A_TO_B']+routes_data['AVG_OCCUPANCY_RATE_B_TO_A'])/2,2).to_list()
value=(round((routes_data['AVG_OCCUPANCY_RATE_A_TO_B']+routes_data['AVG_OCCUPANCY_RATE_B_TO_A'])/2,2)>=0.65)&(routes_data['BREAKEVEN_FLIGHTS']<1200)
routes_data['COLOR_CATEGORY']= np.where( value==True , "green", "orange")
value=(round((routes_data['AVG_OCCUPANCY_RATE_A_TO_B']+routes_data['AVG_OCCUPANCY_RATE_B_TO_A'])/2,2)<=0.64)&(routes_data['BREAKEVEN_FLIGHTS']>2600)
routes_data['COLOR_CATEGORY']= np.where( value==True , "red", routes_data['COLOR_CATEGORY'])
color_category = routes_data['COLOR_CATEGORY'].to_list()

'''
<READ_ME>
def get_short_distance_route_category(COLOR_CATEGORY):
    if COLOR_CATEGORY == 'green':
        return 'HIGHLY RECOMMENDED ROUTES: Less Breakeven Flights, High Occupancy, High Profit:Cost Ratio'
    elif COLOR_CATEGORY == 'orange':
        return 'FUTURE CONSIDERATION ROUTES: Medium Breakeven Flights, Medium Occupancy, Medium Profit:Cost Ratio'
    else:
        return 'NOT RECOMMENDED ROUTES: High Breakeven Flights, Low Occupancy, Low Profit:Cost Ratio'
'''
```

```
def get_short_distance_route_category(COLOR_CATEGORY):
    if COLOR_CATEGORY == 'green':
        return 'HIGHLY RECOMMENDED ROUTES'
    elif COLOR_CATEGORY == 'orange':
        return 'FUTURE CONSIDERATION ROUTES'
    else:
        return 'NOT RECOMMENDED ROUTES'

routes_data['SHORT_DISTANCE_ROUTE_CATEGORY'] = routes_data['COLOR_CATEGORY'].apply(get_short_distance_route_category)
short_distance_route_category = routes_data['SHORT_DISTANCE_ROUTE_CATEGORY'].to_list()
routes_data['FILL_ALPHA']= 0.70
fill_alpha = routes_data['FILL_ALPHA'].to_list()
routes_data['LINE_ALPHA']= 1
line_alpha = routes_data['LINE_ALPHA'].to_list()
routes_data = routes_data.sort_values(['BREAKEVEN_FLIGHTS'], ascending=[True])
routes_data['TEXT_ALIGN'] = ['left','right','left','right','left','right','left','right','left','right','left','right','left','right','left']
text_align = routes_data['TEXT_ALIGN'].to_list()
routes_data = routes_data.sort_values(['ROUTE_DISTANCE'], ascending=[True])

source = ColumnDataSource(data=dict(r = routes,bf = breakeven_flights,pcr = profit_cost_ratio, ao = avg_occupancy, cc = color_category,
                                    sdrc = short_distance_route_category, fa = fill_alpha, la = line_alpha, ta = text_align))

graph = figure(title = "SHORT DISTANCE-QUICK BREAKEVEN-HIGH OCCUPANCY- HIGH PROFIT:COST RATIO ROUTE")
graph.xaxis.axis_label = "NUMBER OF ROUNDTRIP FLIGHTS TO BREAKEVEN COST"
graph.yaxis.axis_label = "PROFIT TO COST RATIO"
graph.y_range = Range1d(0.63, 0.67)
graph.x_range = Range1d(500, 3200)
```

```
graph = figure(title = "SHORT DISTANCE-QUICK BREAKEVEN-HIGH OCCUPANCY- HIGH PROFIT:COST RATIO ROUTE")
graph.xaxis.axis_label = "NUMBER OF ROUNDTRIP FLIGHTS TO BREAKEVEN COST"
graph.yaxis.axis_label = "PROFIT TO COST RATIO"
graph.y_range = Range1d(0.63, 0.67)
graph.x_range = Range1d(500, 3200)

# plotting the graph
graph.square_dot(x = 'bf',
                 y = 'ao',
                 size = 'pcr',
                 color = 'cc',
                 fill_alpha = 'fa',
                 line_alpha = 'la',
                 legend_group = 'sdrc',
                 source = source)

labels = LabelSet(x='bf', y='ao', text='r', x_offset=5, y_offset=5, source=source, text_font_size = "10px", text_align = 'ta')
graph.add_layout(labels)
# file to save the model
output_file("bokeh.html")

# displaying the model
show(graph)

#*****************************************************************************************************************************
```

Graph Output:

Short Distance Route with High Occupancy, High Profit To Cost Ratio and Quick Breakeven **is MDT-PHL**



Graph [3]:

Title: Top 10 Profitable Routes Graph

Objective: Identify the top 10 routes with highest profit in the quarter

Corresponds To: Data Challenge question#2 - The 10 most profitable round trip routes (without considering the upfront airplane cost) in the quarter. Along with the profit, show total revenue, total cost, summary values of other key components and total round trip flights in the quarter for the top 10 most profitable routes. Exclude canceled flights from these calculations. And,

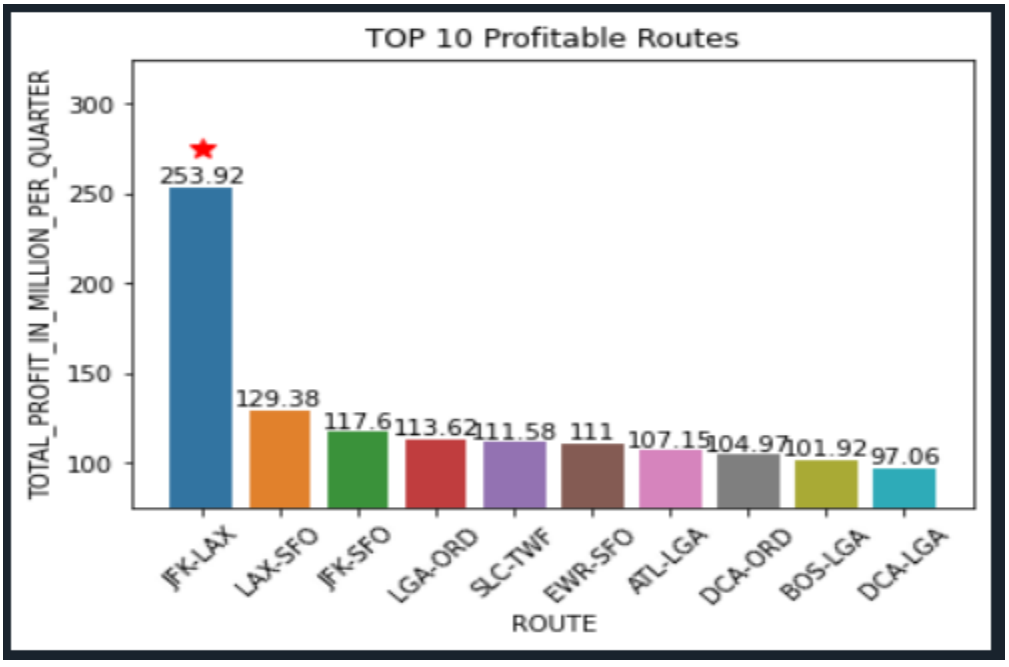
Data Challenge question#3 - The 5 round trip routes that you recommend to invest in based on any factors that you choose

Result: JFK-LAX is the route with the highest profit of 253.92 M in the quarter. This is one of the 5 round-trip routes that I recommend for the airline venture. The top 10 high profit routes: JFK-LAX, LAX-SFO, JFK-SFO, LGA-ORD, SLC-TWF, EWR-SFO, ATL-LGA, DCA-ORD, BOS-LGA & DCA-LGA

Code:

```
#*************************************************************************************************************************
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk

print('Question#2 - TOP 10 Profitable Routes Graph')
#Draw the graph

routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Profitable Roundtrip Routes.xlsx")
routes_data = routes_data.sort_values(['TOTAL_PROFIT'], ascending=[False])
routes_data = routes_data.head(10)
routes_data['TOTAL_PROFIT_IN_MILLION_PER_QUARTER'] = round((routes_data['TOTAL_PROFIT']/1000000),2)
ax = sns.barplot(routes_data, x="ROUTE", y="TOTAL_PROFIT_IN_MILLION_PER_QUARTER")
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
ax.bar_label(ax.containers[0], fontsize=10)
ax.set_ylim(75, 325)
ax.plot(0.01, 275, "*", markersize=10, color="r")
plt.title("TOP 10 Profitable Routes")
plt.show()
```

Graph Output: HIGH PROFIT PER QUARTER ROUTE is JFK-LAX



Graph [4]:

Title: Top 10 Busiest Routes Graph

Objective: Identify the top 10 busiest routes in terms of number of roundtrip flights without cancellations

Corresponds To: Data Challenge question#1 - The 10 busiest round-trip routes in terms of number of round trip flights in the quarter. Exclude canceled flights when performing the calculation.

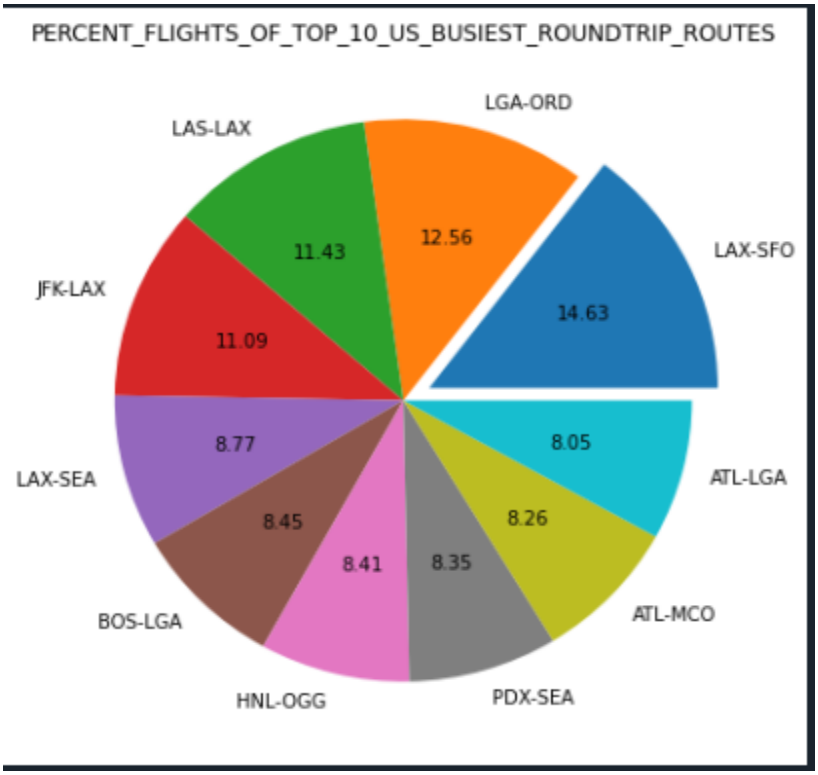
And Data Challenge question#3 - The 5 round trip routes that you recommend to invest in based on any factors that you choose

Result: LAX-SFO is the busiest route with 14.63% of the operated flights of the top 10 busiest routes. This is one of the 5 round-trip routes that I recommend for the airline venture. The top 10 busiest routes are LAX-SFO, LGA-ORD, LAS-LAX, JFK-LAX, LAX-SEA, BOS-LGA, HNL-OGG, PDX-SEA, ATL-MCO and ATL-LGA

Code:

```
#*************************************************************************************************************************
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk
print('Question#1 - TOP 10 Busiest Routes')

#Draw the graph
routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Profitable Roundtrip Routes.xlsx")
routes_data = routes_data.sort_values(['ROUNDTRIP_FLIGHTS'], ascending=[False])
routes_data = routes_data.head(10)
top_10_US_Roundtrips = routes_data['ROUNDTRIP_FLIGHTS'].sum()
routes_data['PERCENT_FLIGHTS_OF_TOP_10_US_ROUNDTRIPS'] = round((routes_data['ROUNDTRIP_FLIGHTS']/top_10_US_Roundtrips)*100,2)
routes_data = routes_data[['ROUTE','PERCENT_FLIGHTS_OF_TOP_10_US_ROUNDTRIPS']]

# Creating plot
routes = routes_data['ROUTE'].to_list()
percent_flights_of_top_10_US_roundtrips = routes_data['PERCENT_FLIGHTS_OF_TOP_10_US_ROUNDTRIPS'].to_list()
fig = plt.figure(figsize=(10, 7))
explode = (0.1, 0, 0, 0, 0, 0,0,0,0,0)
def absolute_value(val):
    return round(val,2)

plt.pie(percent_flights_of_top_10_US_roundtrips, labels=routes, explode=explode, autopct= absolute_value)
plt.title("PERCENT_FLIGHTS_OF_TOP_10_US_BUSIEST_ROUNDTRIP_ROUTES")
# show plot
plt.show()
```

Graph Output: BUSIEST ROUTE is LAX-SFO



Graph [5]:

Title: Avg Flight Delay (In Mins) For The Five New Roundtrip Routes (Q1)

Objective: One of the key performance metrics is to measure the avg delay per route (arrival delay + departure delay)

Corresponds To:

Data Challenge question#5 - Key Performance Indicators (KPI's) that you recommend tracking in the future to measure the success of the round-trip routes that you recommend.

Result:

LAX-SFO is the route with the highest average delay.

Avg delay leg LAX to SFO: 38 mins

Avg delay leg SFO to LAX: 32 mins

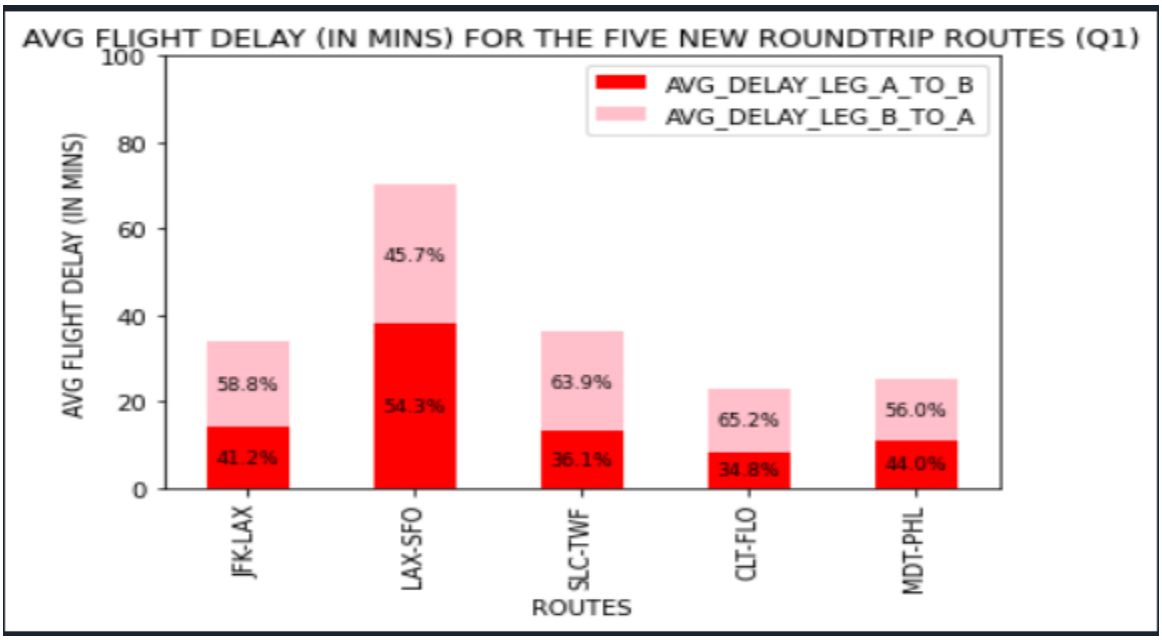
Total Avg leg delay for the route is 70mins

Code:

```
#*************************************************************************************************************************************
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk
print('Question#5 - AVG FLIGHT DELAY (IN MINS) FOR THE FIVE NEW ROUNDTRIP ROUTES (Q1)')
#Draw the graph
routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Key Performance Indicators.xlsx")
routes_data = routes_data[['ROUTE','AVG_DELAY_LEG_A_TO_B','AVG_DELAY_LEG_B_TO_A']]
ax = routes_data.plot(x = 'ROUTE', kind='bar', stacked=True, color=['red', 'pink'])

# Add Title and Labels
plt.title('AVG FLIGHT DELAY (IN MINS) FOR THE FIVE NEW ROUNDTRIP ROUTES (Q1)')
plt.xlabel('ROUTES')
plt.ylabel('AVG FLIGHT DELAY (IN MINS)')
ax.set_ylim(0, 100)
print(routes_data)

routes_data_total = routes_data["AVG_DELAY_LEG_A_TO_B"] + routes_data["AVG_DELAY_LEG_B_TO_A"]
routes_data_rel = routes_data[routes_data.columns[1:]].div(routes_data_total, 0) * 100
print(routes_data_total)
print(routes_data_rel)

for n in routes_data_rel:
    print("Iteration begin : " + n)
    print(routes_data.iloc[:, 1:].cumsum(1)[n])
    print(routes_data[n])
    print(routes_data_rel[n])
    print(zip(routes_data.iloc[:, 1:].cumsum(1)[n],routes_data[n], routes_data_rel[n]))
    print(enumerate(zip(routes_data.iloc[:, 1:].cumsum(1)[n],routes_data[n], routes_data_rel[n])))
    print("Iteration end : " + n)
    for i, (cs, ab, pc) in enumerate(zip(routes_data.iloc[:, 1:].cumsum(1)[n],
```

```
    for i, (cs, ab, pc) in enumerate(zip(routes_data.iloc[:, 1:].cumsum(1)[n],
                                         routes_data[n], routes_data_rel[n])):
        print("**********Iteration begin : " )
        print(i)
        print(cs)
        print(ab)
        print(pc)
        print(cs - ab / 2)
        print(str(np.round(pc, 1)) + '%')
        print("**********Iteration end : ")

        plt.text(i,cs - ab / 2, str(np.round(pc, 1)) + '%',
                 va = 'center', ha = 'center', fontsize = 8)

#*************************************************************************************************************************************
```

Graph Output: Route Delay Monitoring



Graph [6]:

Title: Operated VS. Cancelled flights For The Five New Roundtrip Routes (Q1)

Objective: one of the key performance metrics is to measure the flight cancellations

Corresponds To:

Data Challenge question#5 - Key Performance Indicators (KPI's) that you recommend tracking in the future to measure the success of the round-trip routes that you recommend.

Result: LAX-SFO is the route with the highest cancellations in the quarter

Operated flights LAX-SFO route: 4164 flights

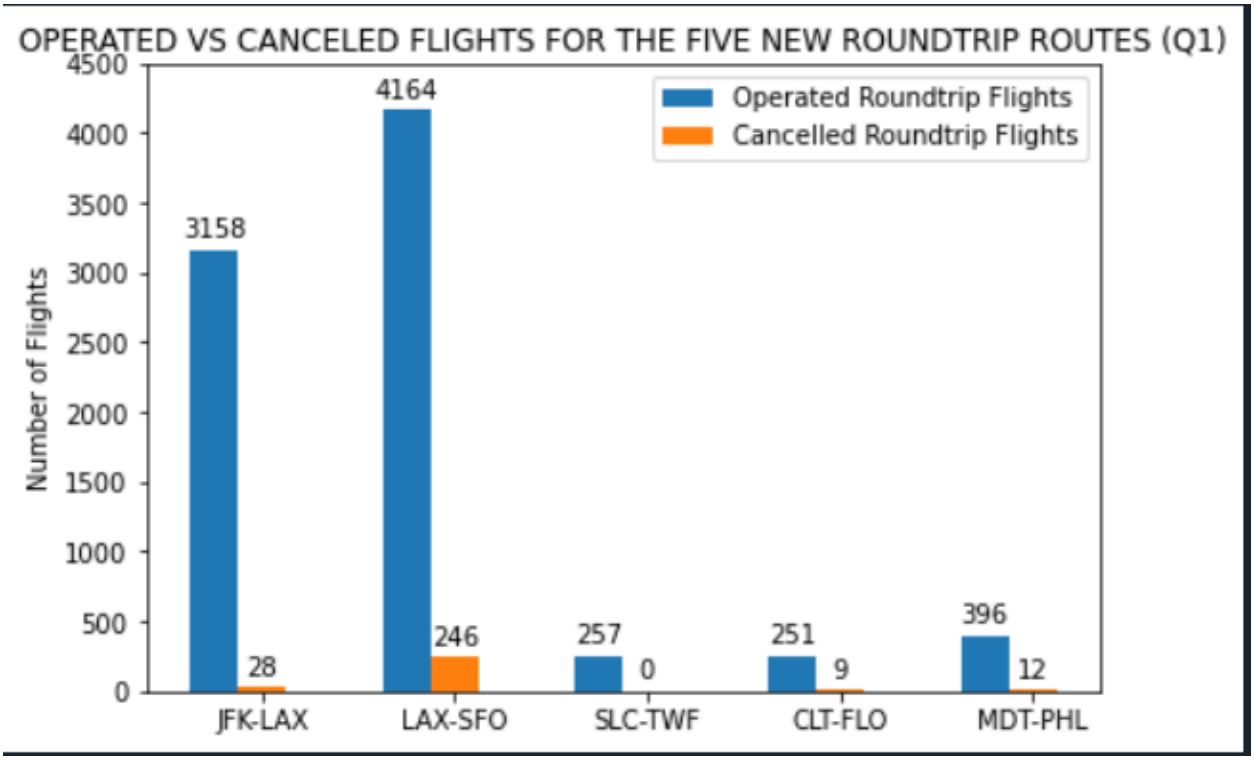
Cancelled flights LAX-SFO route: 246 flights

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk
print('Question#5 - OPERATED VS CANCELED FLIGHTS FOR THE FIVE NEW ROUNDTRIP ROUTES (Q1)')
#Draw the graph

routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Key Performance Indicators.xlsx")
routes_data = routes_data[['ROUTE','ROUNDTRIP_FLIGHTS','ROUNDTRIP_CAN_FLIGHTS']]
print("Grouped Bar chart")

routes = routes_data["ROUTE"].to_list()
route_means = {
    'Operated Roundtrip Flights': routes_data["ROUNDTRIP_FLIGHTS"].to_list(),
    'Cancelled Roundtrip Flights': routes_data["ROUNDTRIP_CAN_FLIGHTS"].to_list()
}
x = np.arange(len(routes))  # the label locations
width = 0.25  # the width of the bars
multiplier = 0
fig, ax = plt.subplots(layout='constrained')

for attribute, measurement in route_means.items():
    offset = width * multiplier
    rects = ax.bar(x + offset, measurement, width, label=attribute)
    ax.bar_label(rects, padding=3)
    multiplier += 1
# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Number of Flights')
ax.set_title('OPERATED VS CANCELED FLIGHTS FOR THE FIVE NEW ROUNDTRIP ROUTES (Q1)')
ax.set_xticks(x + width, routes)
ax.legend(loc='upper right', ncols=1)
ax.set_ylim(0, 4500)
plt.show()
```

Graph Output:

Flight Cancellation Monitoring:



Graph [7]:

Title: Breakeven Analysis for the Route SLC-TWF

Objective: Identify the number of flights to breakeven on the operating cost and the upfront airplane cost for the route SLC-TWF. The breakeven graphs for the other routes can be generated in a similar fashion

Corresponds To:

Data Challenge question#4 - The number of round trip flights it will take to breakeven on the upfront airplane cost for each of the 5 round trip routes that you recommend. Print key summary components for these routes.

And

Data Challenge question#3 - The 5 round trip routes that you recommend to invest in based on any factors that you choose

Result: SLC-TWF is the route with the lowest number of flights to quickly breakeven on the upfront airline cost and the operating costs. It takes only 207 flights to breakeven. This is one of the 5 round-trip routes that I recommend for the airline venture

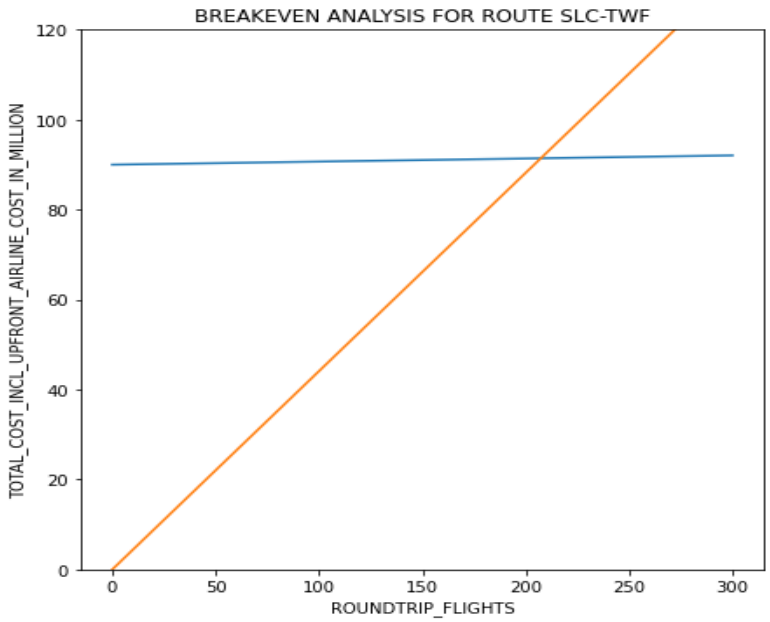
Code:

```
#*************************************************************************************************************************************
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
#import pyplot as plt
#import Tickets as tk
print('Question#5 - BREAKEVEN ANALYSIS FOR ROUTE SLC-TWF')
#Draw the graph
routes_data = pd.read_excel("C://Users//...//Capital One Analytics//data4//Key Performance Indicators.xlsx")
routes_data = routes_data[['ROUTE','ROUNDTRIP_FLIGHTS','TOTAL_REVENUE','TOTAL_COST']]
routes_data = routes_data[routes_data['ROUTE']=='SLC-TWF']
'''
<READ_ME>
for SLC-TWF ROUTE
    ROUNDTRIP_FLIGHTS    TOTAL_REVENUE       TOTAL_COST
    257                  113340150           1763742
    X                    (X/257)*113340150   (X/257)*1763742

'''
routes_data = pd.DataFrame({"ROUTE": ['SLC-TWF','SLC-TWF','SLC-TWF','SLC-TWF','SLC-TWF','SLC-TWF','SLC-TWF'],
                        "ROUNDTRIP_FLIGHTS":[0, 50, 100, 150, 200, 250 ,300],
                        "TOTAL_REVENUE_IN_MILLION":[0, 0, 0, 0, 0, 0, 0],
                        "TOTAL_COST_IN_MILLION":[0, 0, 0, 0, 0, 0, 0],
                        "TOTAL_COST_INCL_UPFRONT_AIRLINE_COST_IN_MILLION":[0, 0, 0, 0, 0, 0, 0]
                        })
def get_total_revenue(roundtrip_flights):
        return round(int((roundtrip_flights/257)*113340150)/1000000,2)

def get_total_cost(roundtrip_flights):
    return round(int((roundtrip_flights/257)*1763742)/1000000,2)

routes_data['TOTAL_REVENUE_IN_MILLION'] = routes_data['ROUNDTRIP_FLIGHTS'].apply(get_total_revenue)
routes_data['TOTAL_COST_IN_MILLION'] = routes_data['ROUNDTRIP_FLIGHTS'].apply(get_total_cost)
routes_data['TOTAL_COST_INCL_UPFRONT_AIRLINE_COST_IN_MILLION'] = routes_data['TOTAL_COST_IN_MILLION'] + 90;
```

```
revenue = routes_data['TOTAL_REVENUE_IN_MILLION'].to_list()
cost = routes_data['TOTAL_COST_INCL_UPFRONT_AIRLINE_COST_IN_MILLION'].to_list()
number_of_flights = routes_data['ROUNDTRIP_FLIGHTS'].to_list()

ax = plt.subplots(figsize=(7,7))
ax = sns.lineplot(x = "ROUNDTRIP_FLIGHTS", y = "TOTAL_COST_INCL_UPFRONT_AIRLINE_COST_IN_MILLION", data = routes_data, label='Costs')
ax = sns.lineplot(x = "ROUNDTRIP_FLIGHTS", y = "TOTAL_REVENUE_IN_MILLION", data = routes_data, label='Revenue')
plt.ylim(0, 120)
plt.title("BREAKEVEN ANALYSIS FOR ROUTE SLC-TWF")
plt.show()
```

Graph Output:

QUICK BREAKEVEN ROUTE is SLC-TWF with 207 flights to breakeven on the upfront airline cost



Final recommendation:

The origination airport and destination airport for each of the five round trip routes that are recommended are the below:

- High Profit Per Quarter Route: JFK-LAX
- Busiest Route: LAX-SFO
- Quick Break-even Route: SLC-TWF
- High Profit to Cost Ratio Route: CLT-FLO
- Short Distance route with High Occupancy, High Profit to Cost Ratio and Quick Breakeven: MDT-PHL

The recommended key performance indicators whose trend needs to be tracked over time (preferably week over week) to track to measure success are:

<u>KPIs Per route (week over week)</u>

- Total Profit
- Total Revenue
- Total cost
- Total profit to cost ratio
- Avg Delay
- Number of roundtrip flights
- Total flight cancellations
- Total roundtrip passengers
- Avg route occupancy rate
- Ticket Revenue and Baggage Revenue
- Number of flights pending to breakeven