Meta-Data for Users Defined Variables

1) In Functions:

1.1) data_wrangling() function: This function is used to clean all the data's (flights, tickets, airports). I have created local variables to take in the data and to process it.
   a) data: This is used to take in the dataframe in which the data is stored.
   b) cols_to_clean: This is a list which takes in the columns names from the data which are needed to be cleaned. (Ex: DISTANCE and ITIN_FARE).
   c) colstype_to_change: This is a list which takes in the column names from the data where the datatype of the column needs to be changed to assess further.
   d) dtypes: This is a list where the datatypes are passed to be change the datatype of the columns passed above. They should be passed respectively.
   e) dropna: This is a variable where default is set to "0" where the user can explicitly change the value while calling the function. This is used to drop the records containing null values in the data.

1.2) sorted_routes() function: This function is used to combine similar routes from origin and destination.
   a) data: This is used to take in the dataframe in which the data is stored and continue with the process.

1.3) revenue() function: This function is used to calculate the revenue generated through different ways.
   a) baggage_tickets_revenue_data: This is used to take in the dataframe in which the data is stored and continue with the process. This data is the information containing all the flights data along with their airport types for both on origin and destination field.
   b) avg_ticket_price_data: This is used to take in the dataframe in which the data is stored and continue with the process. This data is the information containing all the ITIN_FARE's for origin to destination.
   c) seating_capacity: This variable is to represent the capacity it can accommodate per flight. And it set to a default value "200" where this can be passed explicitly by user to change it in the future.
   d) bag_price: This variable is used to represent the price charged for a single check in bag. It is set to default value of "35" where this can be passed explicitly by user to change it in the future.

1.4) costs() function: This function is used to calculate the costs incurred over all the trips in a particular route.
   a) data: This is used to take in the dataframe in which the data is stored and continue with the process. This data is the information containing the flights data along with their airport types for both on origin and destination field.
   b) Fuel_Oil_Main_Crew_Costs: This variable is used to calculate Fuel, Oil, Maintenance, Crew costs incurred on the flight trip. The cost is calculated per mile of the total distance the trip is. It is set to a default value of "8$" can be changed explicitly by user in case the charge changes in the future.
   c) Depri_Insu_Oth_Costs: This variable is used to calculate the depreciation, Insurance and Other costs incurred on the flight trip. The cost is calculated per mile of the total distance the trip is. It is set to a default value of "1.18$" can be changed explicitly by user in case the charge changes in the future.
   d) large_airport_cost: This variable represents the cost charged when a flight either lands or takes off from a large airport. The default value is set to "10000" can be changed explicitly by user in case the charge changes in the future.

1.5) medium_airport_cost: This variable represents the cost charged when a flight either lands or takes off from a medium airport. The default value is set to "5000" can be changed explicitly by user in case the charge changes in the future.

1.6) profit(): The function is used to calculate the profits from revenue data and cost data, it also performs merge operation to give out all the key metrics along with the profit.
   a. revenue_data: This is a function parameter to take in revenue data.
   b. costs_data: This is a function parameter to take in cost data.
   c. competitors_data: This is a function parameter to take in competitors data.
   d. flights_data: This is a function parameter to take in flights data.
   e. profits: A local variable in which the operations are stored and is returned in the end.

1.7) bep(): This function is used to calculate the break even analysis on the profits data.
   a) profit_data: This is a function parameter which takes in the profits data.
   b) break_even_analysis: This is a function variable which is used to store all the calculations and is returned in the last.

1.8) data_munging(): This is a function used to join (or) merge 2 datasets by checking multiple conditions.
   a) left_df: This is a function parameter which takes in the left dataframe in the merge operation.
   b) right_df: This is a function parameter which takes in the right dataframe in the merge operation.
   c) join: The type of join (inner, left, etc.) it corresponds to 'how' in merge function. Has a default value of "inner" join.
   d) on_key: This is the column that represents "on" parameter in merge function. Can be used if dataframes are joined on same column name.
   e) left_key: This is the column names to join on left dataframe, represents the "left_on" parameter in merge function.
   f) right_key: This is the column names to join on right dataframe, represents the "right_on" parameter in merge function.

1.8) competitors(): This function is used to calculate the number of unique operating carriers in each route.

   a) flights_data: This is a function parameter where this take in the flights data.
   b) competition: This is a variable on which all the operations are performed and is returned.

2) While Processing The Data:
   a) flights: This variable is used to store in the data in a form of a dataframe. It consists of flights data.
   b) airports: This variable is used to store in the data in a form of a dataframe. It consists of airports data.
   c) tickets: This variable is used to store in the data in a form of a dataframe. It consists of tickets data.
   d) flights_airports_origin_destination: This variable is used to store in the data in a form of a dataframe. It consists of flights data joined with the airport types of origin and destination from the airports data.
   e) busy_routes: This variable has the data regarding the number of round-trip flights in all the routes in quarter Q1.
   f) x: This variable is just a temporary variable I have used from time to time and in this case I have used it to create a subset of just the top 10 busiest round trip routes for the above mentioned variable(busy_routes).
   g) baggage_revenue: This variable has the data regarding revenue produced through baggage alone.
   h) avg_ticket_price: This variable has the data regarding the average ticket price in a particular round trip route.

i) tickets_revenue: This variable has the information about the number of passengers that have travelled in the round-trip route, average round trip fare, total tickets revenue generated throughout the round trip.
j) revenues: This has the data from the 3 variables created above (baggage_revenue , avg_ticket_price, tickets_revenue) and the calculated total revenue(baggae_revenue + tickets_revenue).
k) avg_delays: This variable has the information about average departure and average arrival delays for each round-trip route.
l) cost: This variable has the information about average delays, costs incurred due to delays, Fuel_Oil_Maintanance_Crew, Depriciation_Insurance_Other, Airport Operational costs and total cost incurred due to all the individual costs.
m) competition: This variable has the information about the number of unique carriers operating in each route.
n) profits: This variable has the information about the number of competitors, number of flights round trip flights, revenue, costs and the profit acquired for each route.
o) break even analysis: This variable has the information about the break-even analysis. i.e: number of round trips flights required in order to break even the upfront airplane cost(90,000,000$).

Data Munging:

In total throughout my work, I have merged dataframes 8 times.

1. I have joined the "flights" data with the "airports" data using "inner" join with "ORIGIN" on flights and "IATA_CODE" form airports to get the airport type of the ORIGIN of these matched records.

2. I have joined the "flights" data with the "airports" data using "inner" join with "DESTINATION" on flights and "IATA_CODE" form airports to get the airport type of the DESTINATION of these matched records.

3. I have joined the "passengers" data with the "avg_ticket_price" data using "inner" join on "Route" where it is used to compute the ticket revenue.

4. I have joined the "tickets_revenue" data with the "baggage_revenue" data using "inner" join on "Route" where it is used to compute the total revenue.

5. I have joined the "avg_delays" data with the "cost" data using "inner" join on "Route" where it is used to add a key metric average departure delay and average arrival delay for each round-trip route in the final data of costs which can be used to see why delay costs are incurred.

6. I have joined the "revenue" data with the "costs" data using "inner" join on "Route" to calculate profits.

7. I have joined the "profits" data with the "competitors" data using "inner" join on "Route" to see the number of unique competitors in each round-trip route along with profit metrics.

8. I have joined the "profits" data with the "flights" data using "inner" join on "Route" to see the number of round-trip flights in each round-trip route along with profit metrics.

In all the cases I have used "inner" join on data as the 'Route' column has unique records of the round-trip routes so I have used inner join to combine 2 dataframes.