In [1]:
```
#INTRO TO ML Assignment-01 Name: Varshini Narayana SUID: 693330560 Course: MS in Computer Engineering

# Generic inputs for most ML tasks
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
pd.options.display.float_format = '{:,.2f}'.format
# setup interactive notebook mode
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

from IPython.display import display, HTML
```

In [2]:
```
# Fetching invistico Airline data
invistico_Airline_data = pd.read_csv('invistico_Airline.csv')
invistico_Airline_data.head()
```

Out[2]:

	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Check servic
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	...	2	3	3	0	3	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	...	2	3	4	4	4	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	...	2	2	3	3	4	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	...	3	1	1	0	1	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	...	4	2	2	0	2	

5 rows × 23 columns

In [3]:
```
#Checking for NaN values
invistico_Airline_data.isna().sum()
```

Out[3]:
```
satisfaction                          0
Gender                                0
Customer Type                         0
Age                                   0
Type of Travel                        0
Class                                 0
Flight Distance                       0
Seat comfort                          0
Departure/Arrival time convenient     0
Food and drink                        0
Gate location                         0
Inflight wifi service                 0
Inflight entertainment                0
Online support                        0
Ease of Online booking                0
On-board service                      0
Leg room service                      0
Baggage handling                      0
Checkin service                       0
Cleanliness                           0
Online boarding                       0
Departure Delay in Minutes            0
Arrival Delay in Minutes            393
dtype: int64
```

In [4]:
```
#Dropping NaN values
invistico_Airline_data = invistico_Airline_data.dropna()
print(invistico_Airline_data.isnull().sum())
```

```
satisfaction                         0
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Class                                0
Flight Distance                      0
Seat comfort                         0
Departure/Arrival time convenient    0
Food and drink                       0
Gate location                        0
Inflight wifi service                0
Inflight entertainment               0
Online support                       0
Ease of Online booking               0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Cleanliness                          0
Online boarding                      0
Departure Delay in Minutes           0
Arrival Delay in Minutes             0
dtype: int64
```

In [5]:
```
#Checking if the NaN values are dropped
invistico_Airline_data.isna().sum()
```

Out[5]:
```
satisfaction                         0
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Class                                0
Flight Distance                      0
Seat comfort                         0
Departure/Arrival time convenient    0
Food and drink                       0
Gate location                        0
Inflight wifi service                0
Inflight entertainment               0
Online support                       0
Ease of Online booking               0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Cleanliness                          0
Online boarding                      0
Departure Delay in Minutes           0
Arrival Delay in Minutes             0
dtype: int64
```

In [6]:
```
#Checking the data types of columns
df = pd.DataFrame(invistico_Airline_data)

print("DF:")
print(df)

# apply the dtype attribute
result = df.dtypes

print("Output:")
print(result)
```

```
DF:
        satisfaction  Gender      Customer Type  Age   Type of Travel  \
0          satisfied  Female     Loyal Customer   65  Personal Travel
1          satisfied    Male     Loyal Customer   47  Personal Travel
2          satisfied  Female     Loyal Customer   15  Personal Travel
3          satisfied  Female     Loyal Customer   60  Personal Travel
4          satisfied  Female     Loyal Customer   70  Personal Travel
...              ...     ...                ...  ...              ...
129875     satisfied  Female  disloyal Customer   29  Personal Travel
129876  dissatisfied    Male  disloyal Customer   63  Personal Travel
129877  dissatisfied    Male  disloyal Customer   69  Personal Travel
129878  dissatisfied    Male  disloyal Customer   66  Personal Travel
129879  dissatisfied  Female  disloyal Customer   38  Personal Travel

           Class  Flight Distance  Seat comfort  \
0            Eco              265             0
1       Business             2464             0
2            Eco             2138             0
3            Eco              623             0
4            Eco              354             0
...          ...              ...           ...
129875       Eco             1731             5
129876  Business             2087             2
129877       Eco             2320             3
129878       Eco             2450             3
129879       Eco             4307             3

        Departure/Arrival time convenient  Food and drink  ...  \
0                                       0               0  ...
1                                       0               0  ...
2                                       0               0  ...
3                                       0               0  ...
4                                       0               0  ...
...                                   ...             ...  ...
129875                                  5               5  ...
129876                                  3               2  ...
129877                                  0               3  ...
129878                                  2               3  ...
129879                                  4               3  ...

        Online support  Ease of Online booking  On-board service  \
0                    2                       3                 3
1                    2                       3                 4
2                    2                       2                 3
3                    3                       1                 1
4                    4                       2                 2
...                ...                     ...               ...
129875               2                       2                 3
129876               1                       3                 2
129877               2                       4                 4
129878               2                       3                 3
129879               3                       4                 5

        Leg room service  Baggage handling  Checkin service  Cleanliness  \
0                      0                 3                5            3
1                      4                 4                2            3
2                      3                 4                4            4
3                      0                 1                4            1
4                      0                 2                4            2
...                  ...               ...              ...          ...
129875                 3                 4                4            4
129876                 3                 3                1            2
129877                 3                 4                2            3
129878                 2                 3                2            1
129879                 5                 5                3            3

        Online boarding  Departure Delay in Minutes  Arrival Delay in Minutes
0                     2                           0                      0.00
1                     2                         310                    305.00
2                     2                           0                      0.00
3                     3                           0                      0.00
4                     5                           0                      0.00
...                 ...                         ...                       ...
129875                2                           0                      0.00
129876                1                         174                    172.00
129877                2                         155                    163.00
129878                2                         193                    205.00
129879                3                         185                    186.00

[129487 rows x 23 columns]
Output:
satisfaction                      object
Gender                            object
Customer Type                     object
Age                                int64
Type of Travel                    object
```

```
Class                                object
Flight Distance                       int64
Seat comfort                          int64
Departure/Arrival time convenient     int64
Food and drink                        int64
Gate location                         int64
Inflight wifi service                 int64
Inflight entertainment                int64
Online support                        int64
Ease of Online booking                int64
On-board service                      int64
Leg room service                      int64
Baggage handling                      int64
Checkin service                       int64
Cleanliness                           int64
Online boarding                       int64
Departure Delay in Minutes            int64
Arrival Delay in Minutes            float64
dtype: object
```

In [7]:
```
# Checking for number of rows (n) in the single dataframe without counting the column name.
n = invistico_Airline_data.shape[0]
print("Number of rows:", n)
```

```
Number of rows: 129487
```

In [8]:
```
#Finding mean and standard deviation values
invistico_Airline_data.describe()
```

Out[8]:

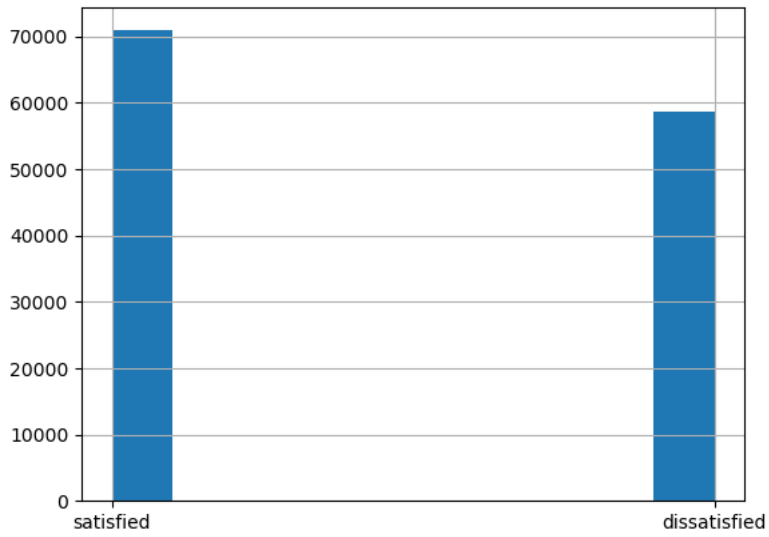
	Age	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	Inflight entertainment	Online support	Ease of Online booking	On-board service	Leg room service	Ba ha
count	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129,487.00	129
mean	39.43	1,981.01	2.84	2.99	2.85	2.99	3.25	3.38	3.52	3.47	3.47	3.49	
std	15.12	1,026.88	1.39	1.53	1.44	1.31	1.32	1.35	1.31	1.31	1.27	1.29	
min	7.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
25%	27.00	1,359.00	2.00	2.00	2.00	2.00	2.00	2.00	3.00	2.00	3.00	2.00	
50%	40.00	1,924.00	3.00	3.00	3.00	3.00	3.00	4.00	4.00	4.00	4.00	4.00	
75%	51.00	2,543.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	5.00	4.00	5.00	
max	85.00	6,951.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	

In [9]:
```
#Obtaining the histograms of the columns with numerical data
cols = invistico_Airline_data.columns
print(cols)
for col in cols:
    invistico_Airline_data[col].hist()
    print(col)
    plt.show()
```

```
Index(['satisfaction', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Seat comfort',
       'Departure/Arrival time convenient', 'Food and drink', 'Gate location',
       'Inflight wifi service', 'Inflight entertainment', 'Online support',
       'Ease of Online booking', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Cleanliness', 'Online boarding',
       'Departure Delay in Minutes', 'Arrival Delay in Minutes'],
      dtype='object')
```

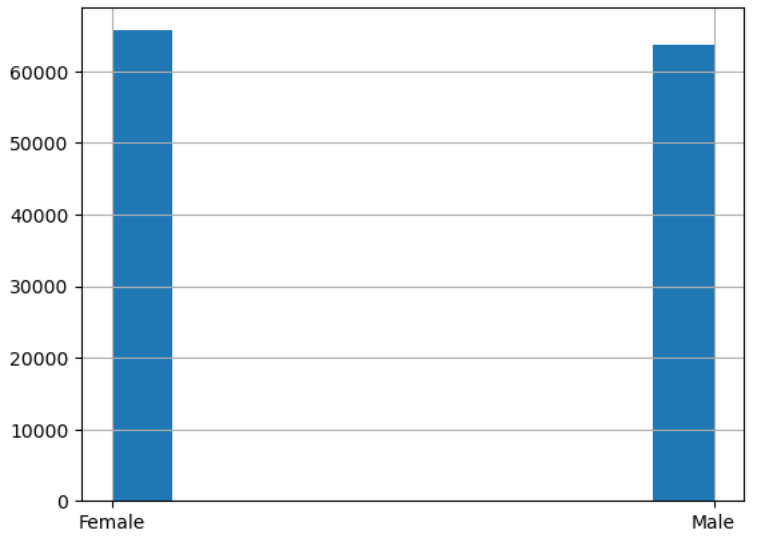
Out[9]: <AxesSubplot:>

satisfaction



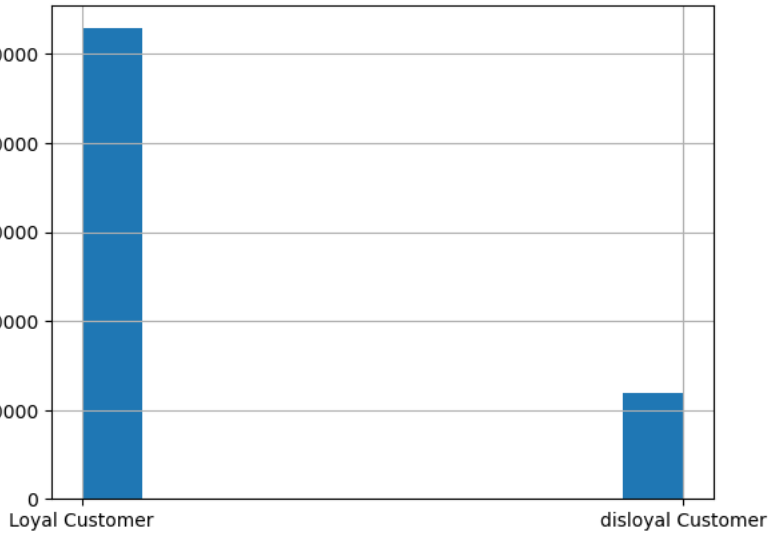
Out[9]: <AxesSubplot:>

Gender



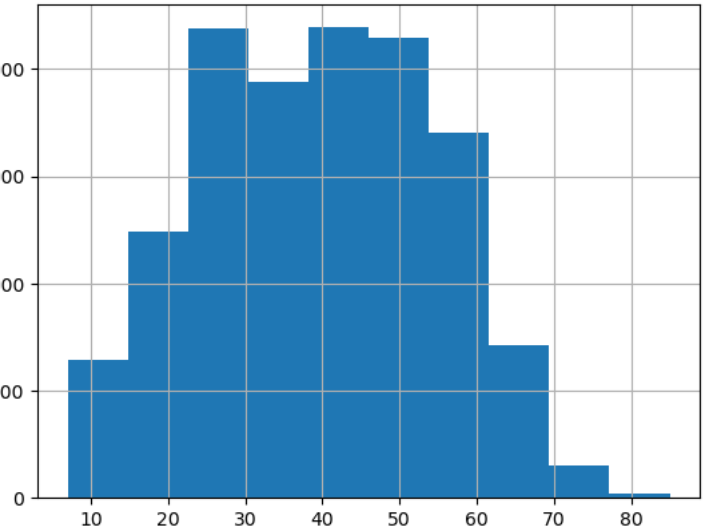
Out[9]: <AxesSubplot:>

Customer Type



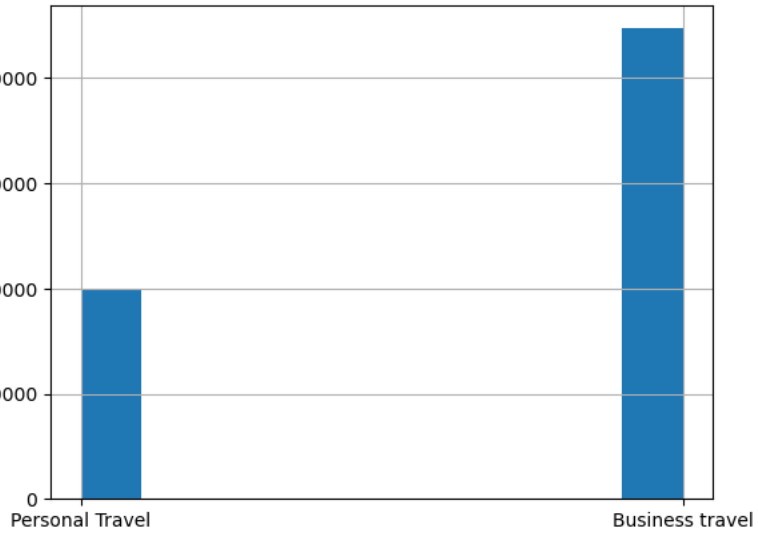
Out[9]: <AxesSubplot:>

Age



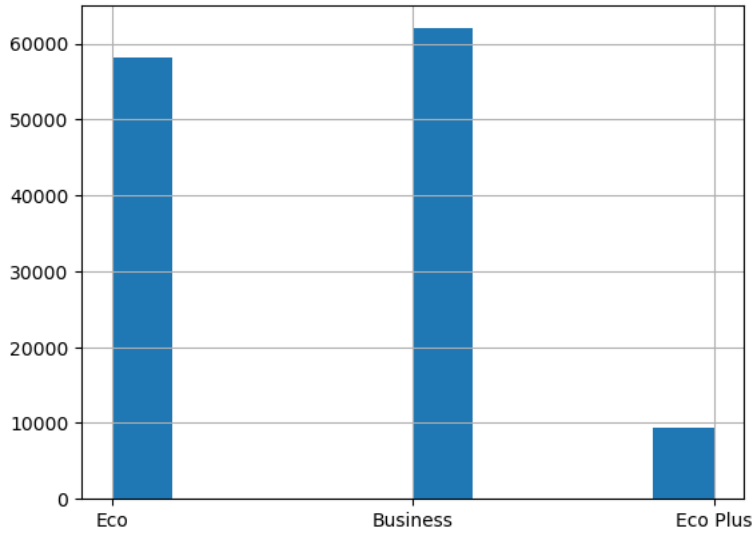
Out[9]: <AxesSubplot:>

Type of Travel



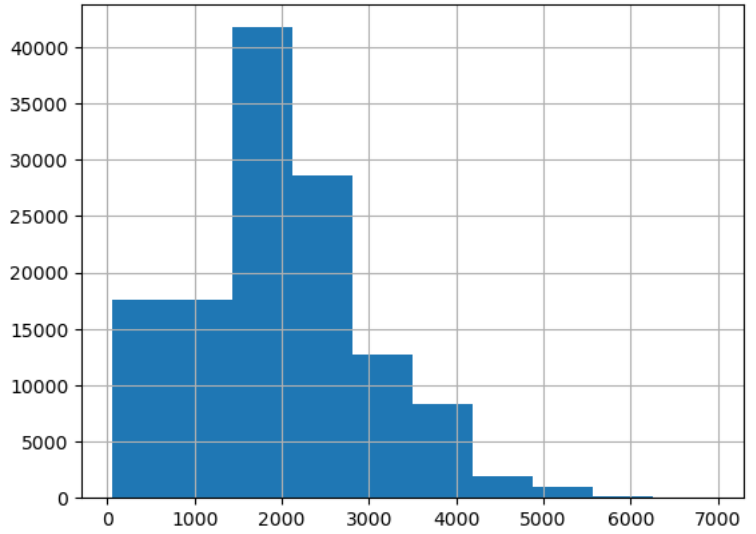
Out[9]: <AxesSubplot:>

Class



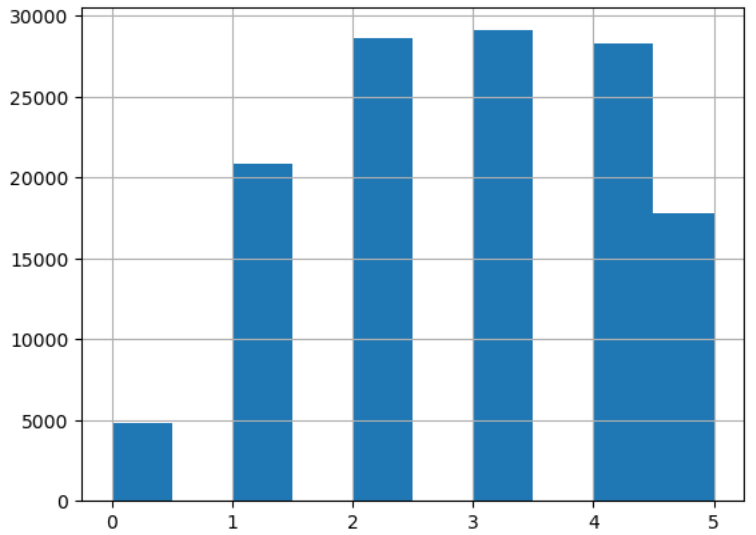
Out[9]: <AxesSubplot:>

```
Flight Distance
```



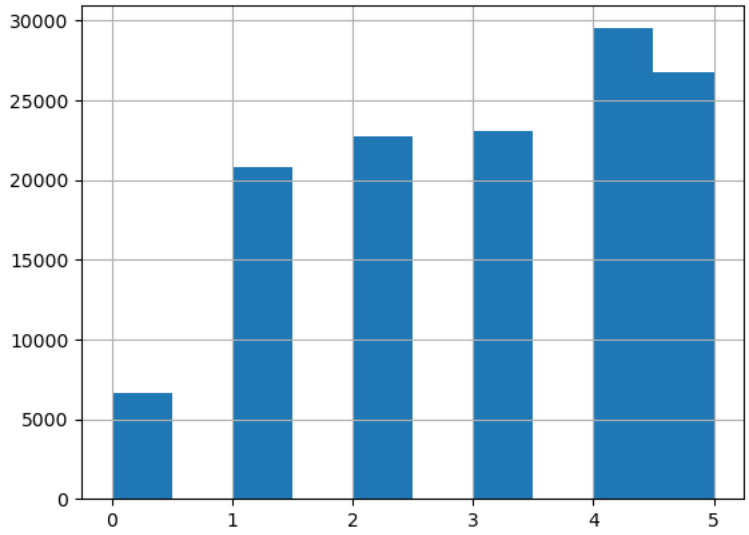
Out[9]: <AxesSubplot:>

```
Seat comfort
```



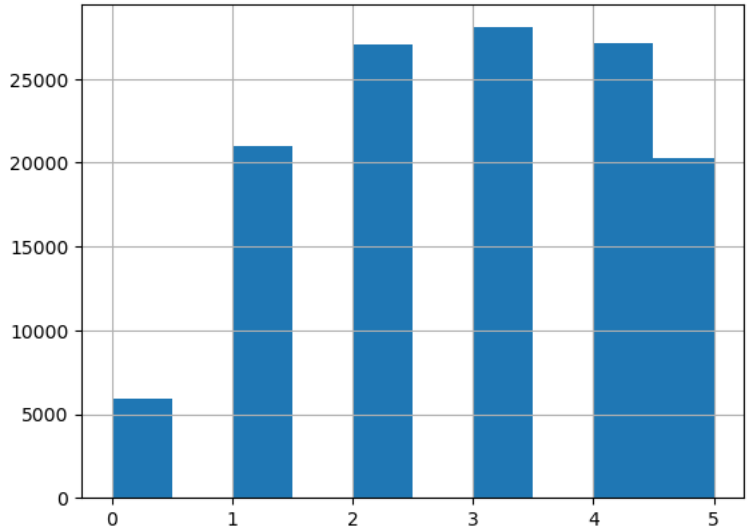
Out[9]: <AxesSubplot:>

```
Departure/Arrival time convenient
```



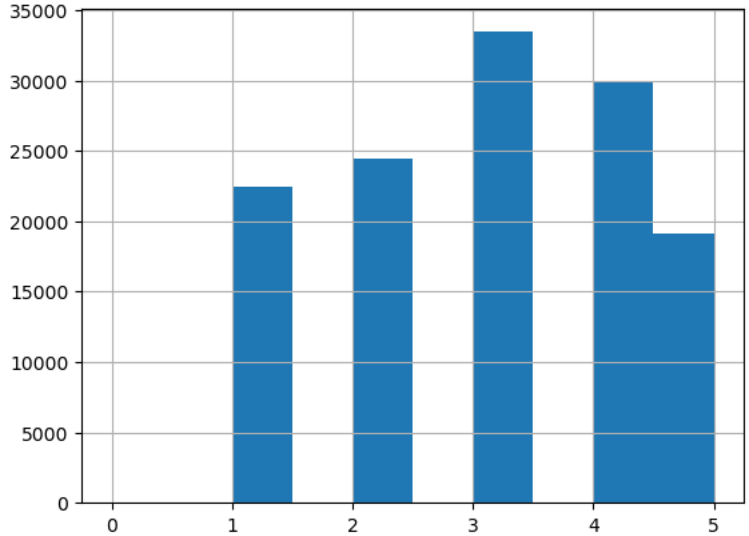
Out[9]: <AxesSubplot:>

Food and drink



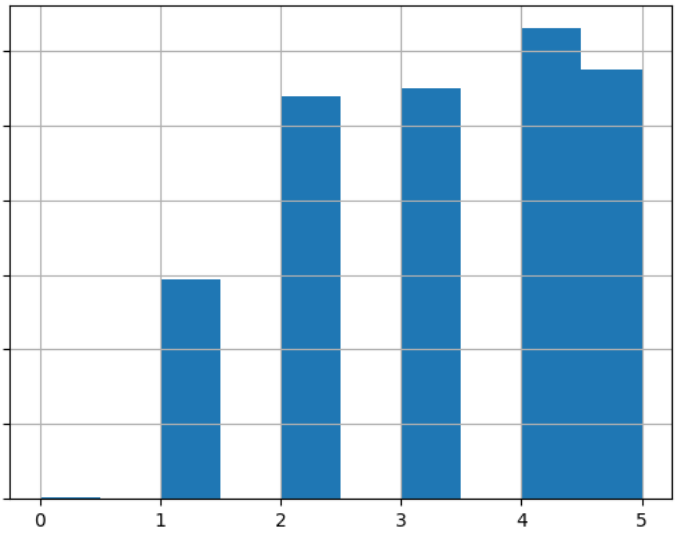
Out[9]: <AxesSubplot:>

Gate location



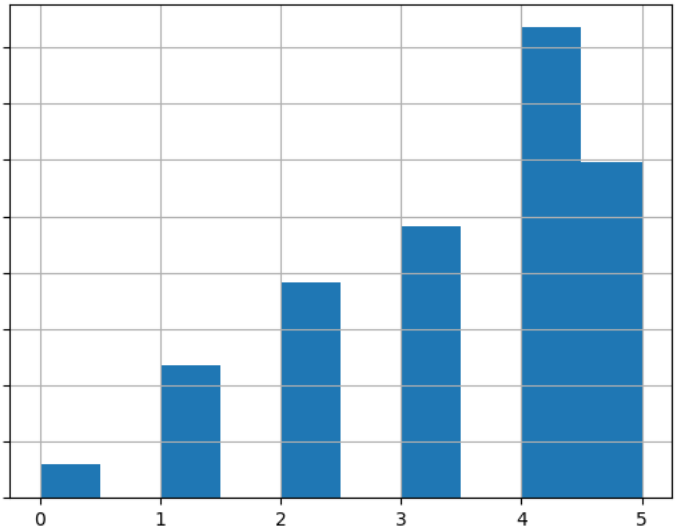
Out[9]: <AxesSubplot:>

Inflight wifi service



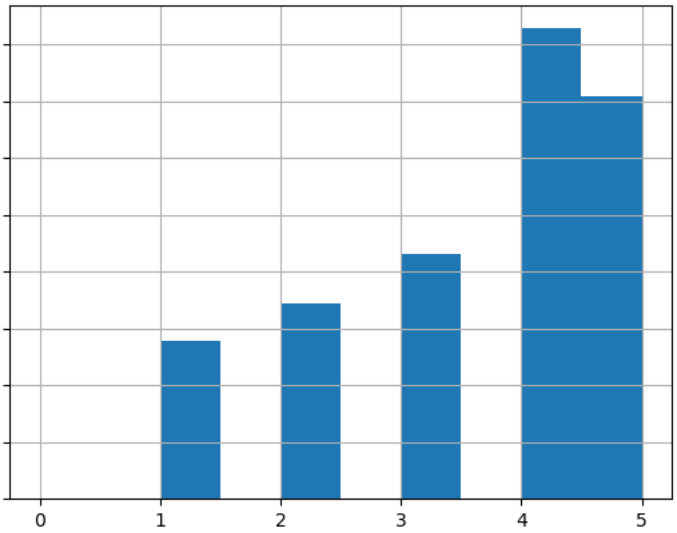
Out[9]: <AxesSubplot:>

```
Inflight entertainment
```



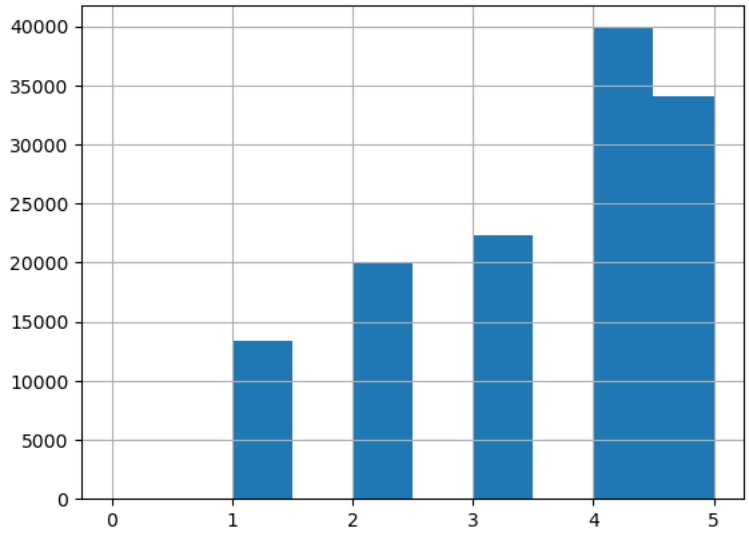
Out[9]: <AxesSubplot:>

```
Online support
```



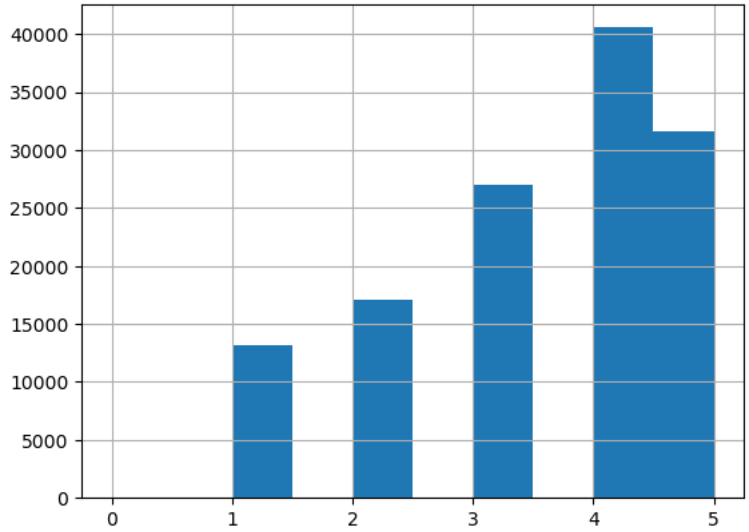
Out[9]: <AxesSubplot:>

```
Ease of Online booking
```



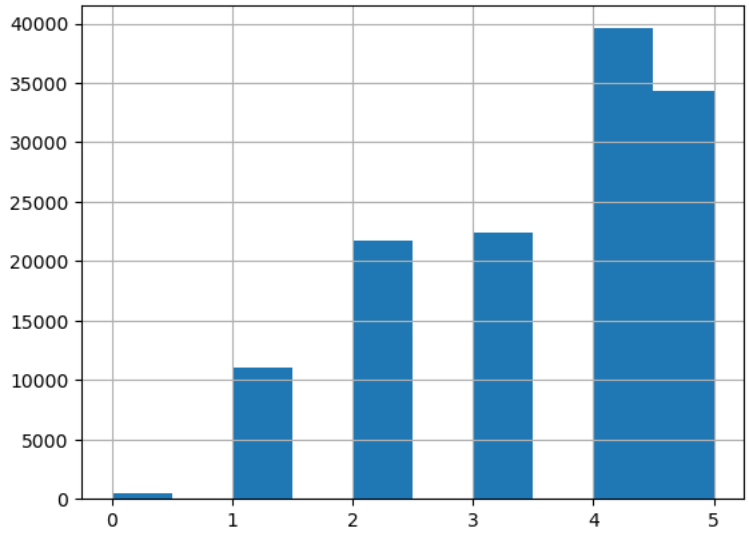
Out[9]: <AxesSubplot:>

On-board service



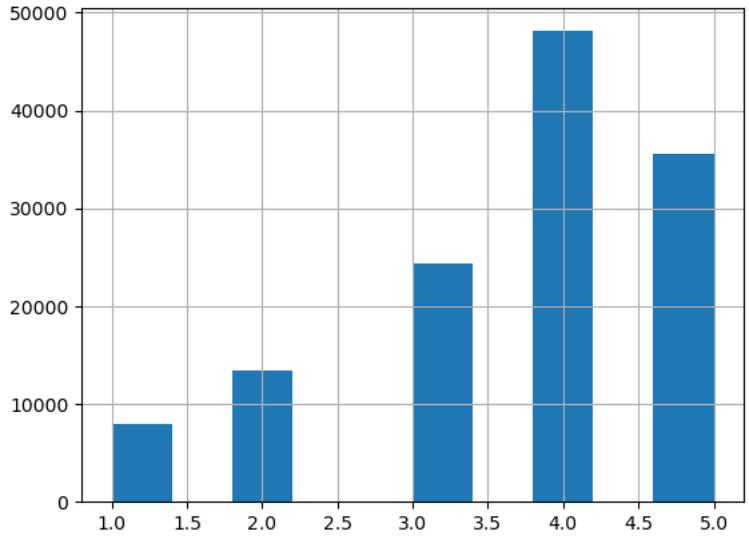
Out[9]: <AxesSubplot:>

Leg room service



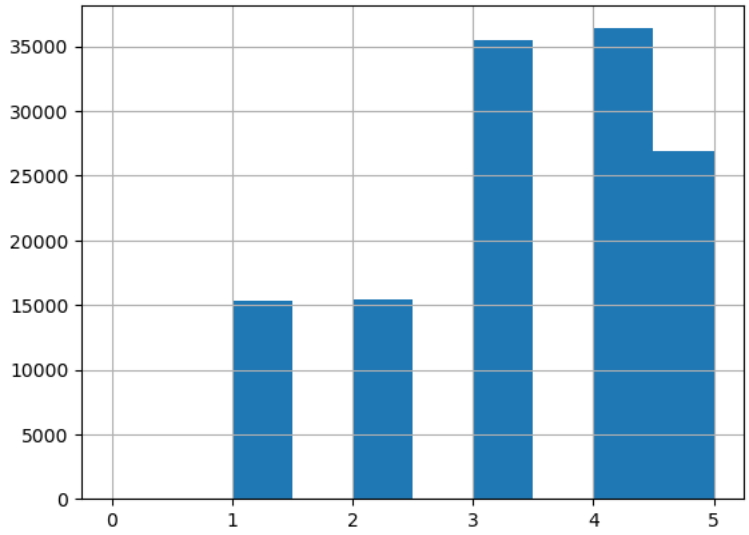
Out[9]: <AxesSubplot:>

Baggage handling



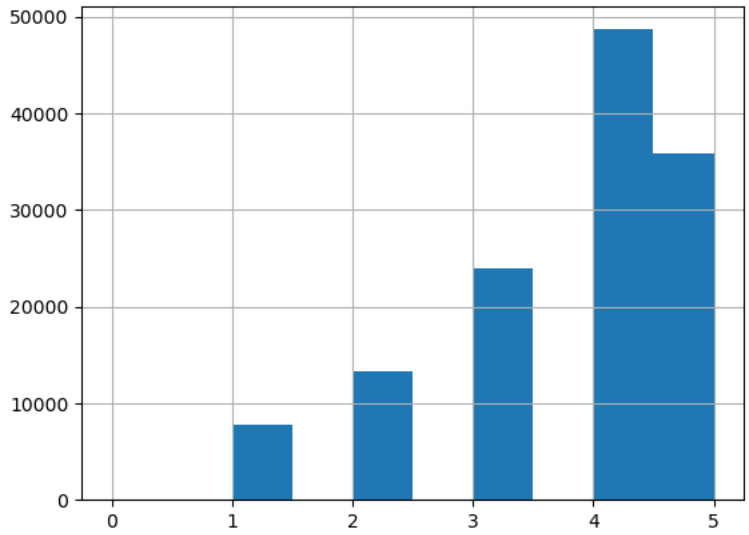
Out[9]: <AxesSubplot:>

Checkin service



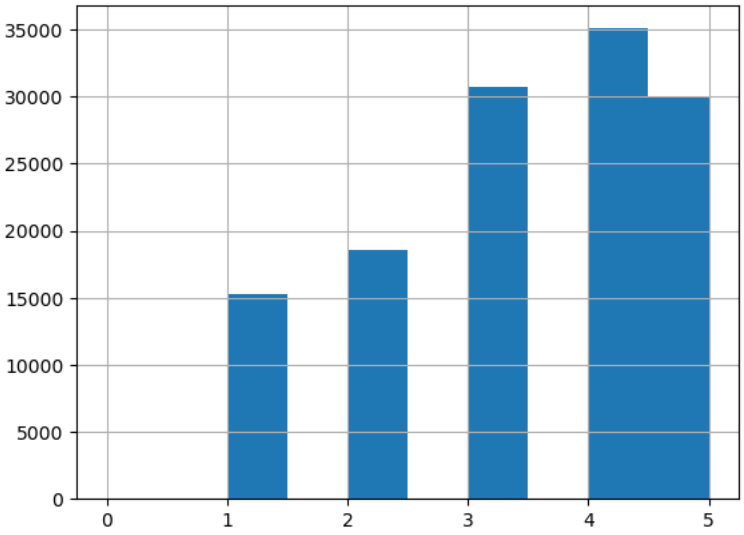
Out[9]: <AxesSubplot:>

Cleanliness



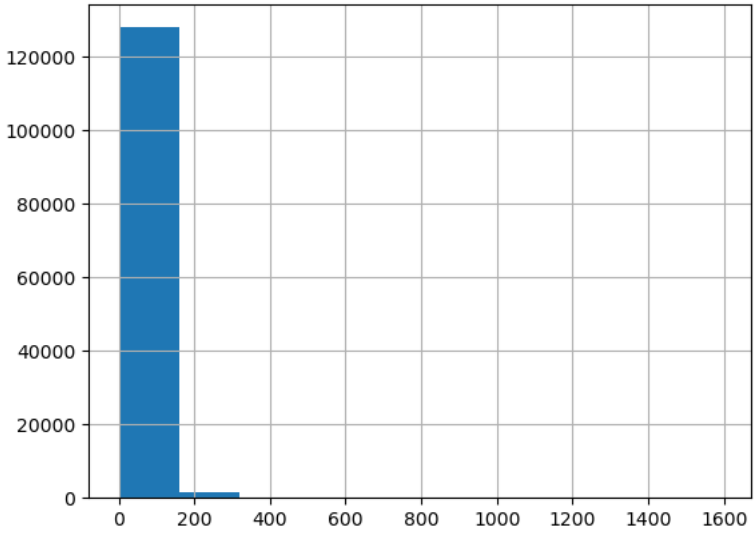
Out[9]: <AxesSubplot:>

Online boarding



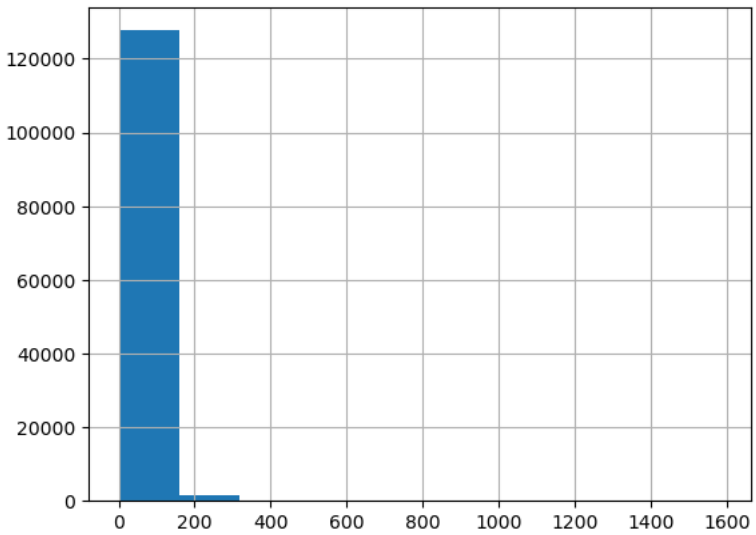
Out[9]: <AxesSubplot:>

```
Departure Delay in Minutes
```



Out[9]: <AxesSubplot:>

```
Arrival Delay in Minutes
```



In [10]:
```
# define function to import viz libraries
import plotly
plotly.offline.init_notebook_mode(connected=True)
from plotly.graph_objs import *
from plotly import tools
import plotly.graph_objects as go
import seaborn as sns
```

In [11]:
```
# Function to visualize by features
def visualize_by_features(list_of_features, y_col):
    for x_col in list_of_features:
        plot_data = []
        df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
        plot_data.append(go.Scatter(x= df[x_col], y= df[y_col],
                        name = 'feature = {}'.format(x_col), mode = 'markers' ))

        layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                           title = 'Data for {}'.format(y_col))
        fig = go.Figure(data= plot_data, layout=layout)
        plotly.offline.iplot(fig)
```

In [12]:
```
invistico_Airline_data.columns
```

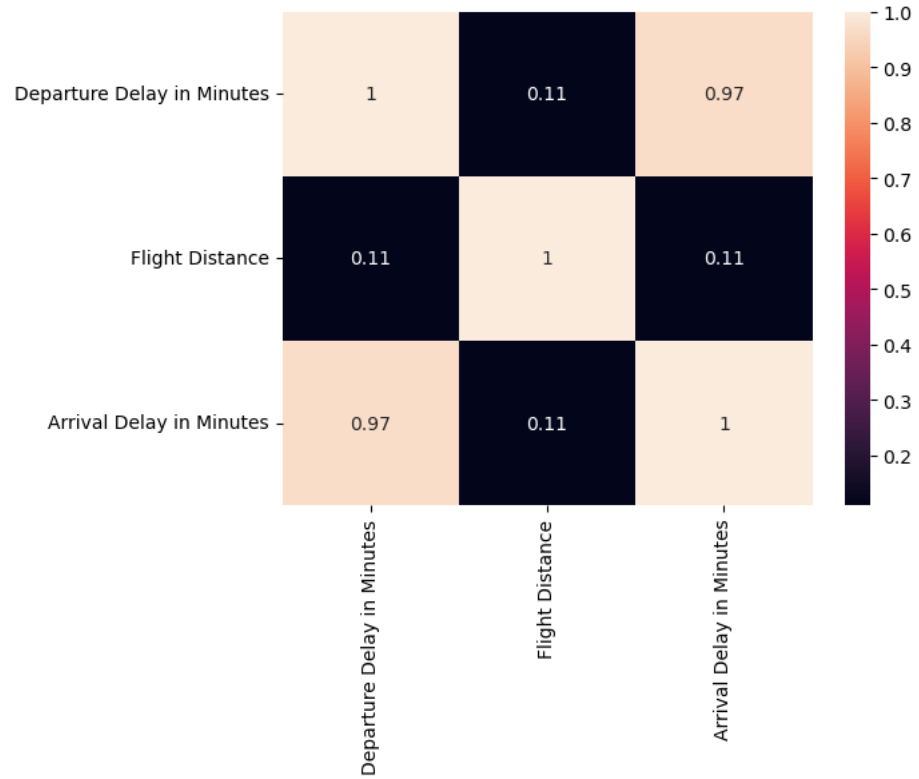
Out[12]:
```
Index(['satisfaction', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Seat comfort',
       'Departure/Arrival time convenient', 'Food and drink', 'Gate location',
       'Inflight wifi service', 'Inflight entertainment', 'Online support',
       'Ease of Online booking', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Cleanliness', 'Online boarding',
       'Departure Delay in Minutes', 'Arrival Delay in Minutes'],
      dtype='object')
```

In [13]:
```
#Arrival delay in minutes is more correlated with Departure delay in minutes than with Flight distance. - True
df_1var = invistico_Airline_data[['Departure Delay in Minutes', 'Flight Distance', 'Arrival Delay in Minutes']]
```

In [14]:
```
corr = df_1var.corr()
```

In [15]:
```
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True)
```

Out[15]:
```
<AxesSubplot:>
```

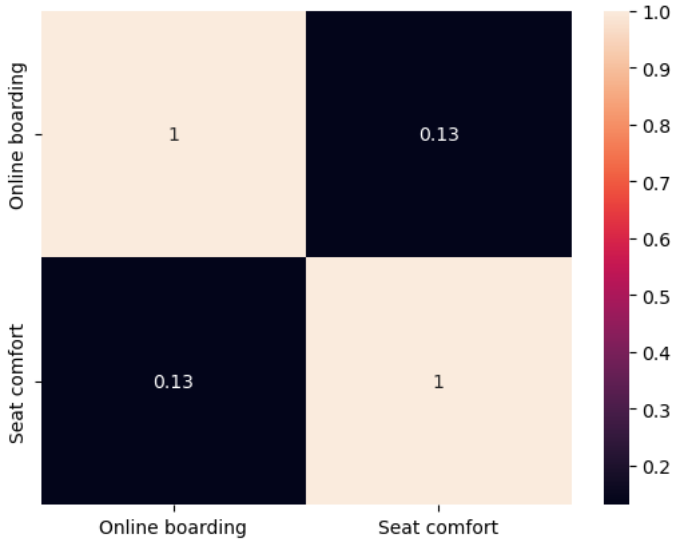


In [16]:
```
# The correlation value between online boarding and Seat comfort is lower than -0.3. - False
df_2var = invistico_Airline_data[['Online boarding','Seat comfort']]
```

In [17]:
```
corr = df_2var.corr()
```

In [18]:
```
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True)
```

Out[18]: <AxesSubplot:>

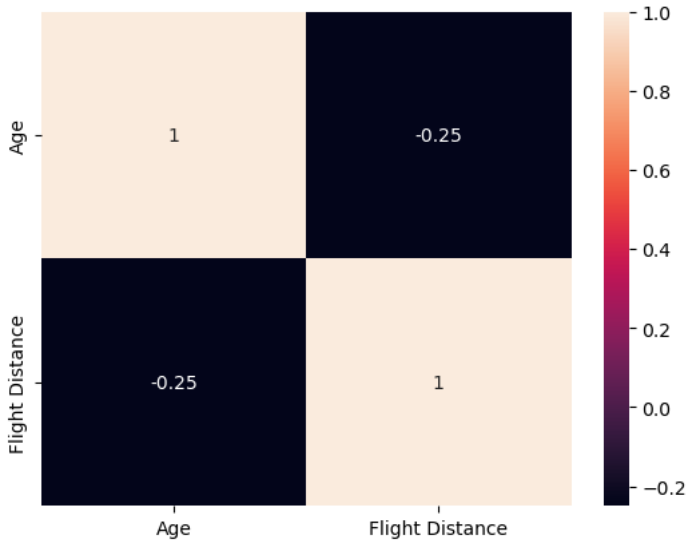


In [19]:
```
#Age and Flight distance show a good positive correlation. - False
df_3var = invistico_Airline_data[['Age', 'Flight Distance']]
```

In [20]:
```
corr = df_3var.corr()
```

In [21]:
```
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True)
```

Out[21]: <AxesSubplot:>

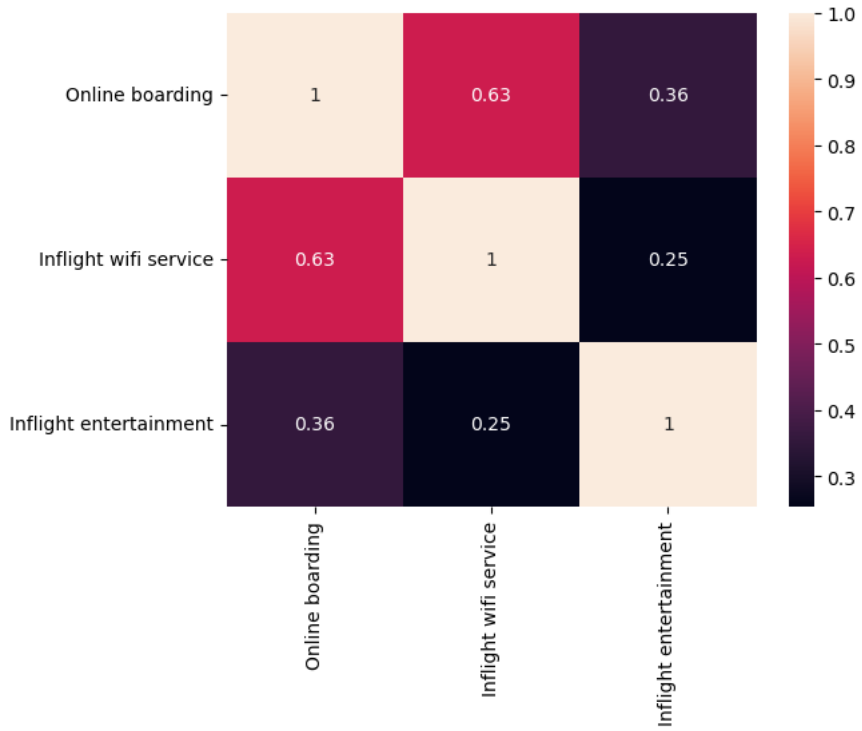


In [22]:
```
#Online boarding is more correlated with Inflight entertainment than with Inflight wifi service. - False
df_4var = invistico_Airline_data[['Online boarding', 'Inflight wifi service','Inflight entertainment']]
```

In [23]:
```
corr = df_4var.corr()
```

In [25]:
```
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True)
```

Out[25]: <AxesSubplot:>

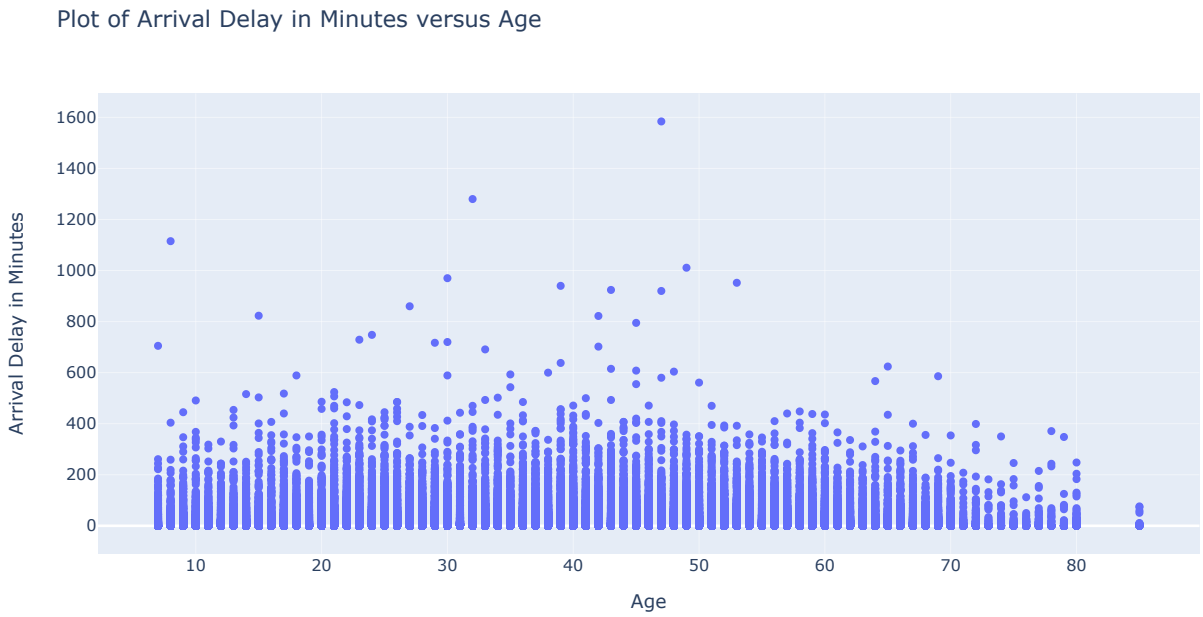


In [26]:
```
he data by removing any row where the 'Departure Delay in Minutes' and the 'Arrival Delay in Minutes' are both zero and
 as pd
sv("invistico_Airline.csv")
Departure Delay in Minutes'] > 0) & (df['Arrival Delay in Minutes'] > 0)]
0]
 of rows:", n)
```

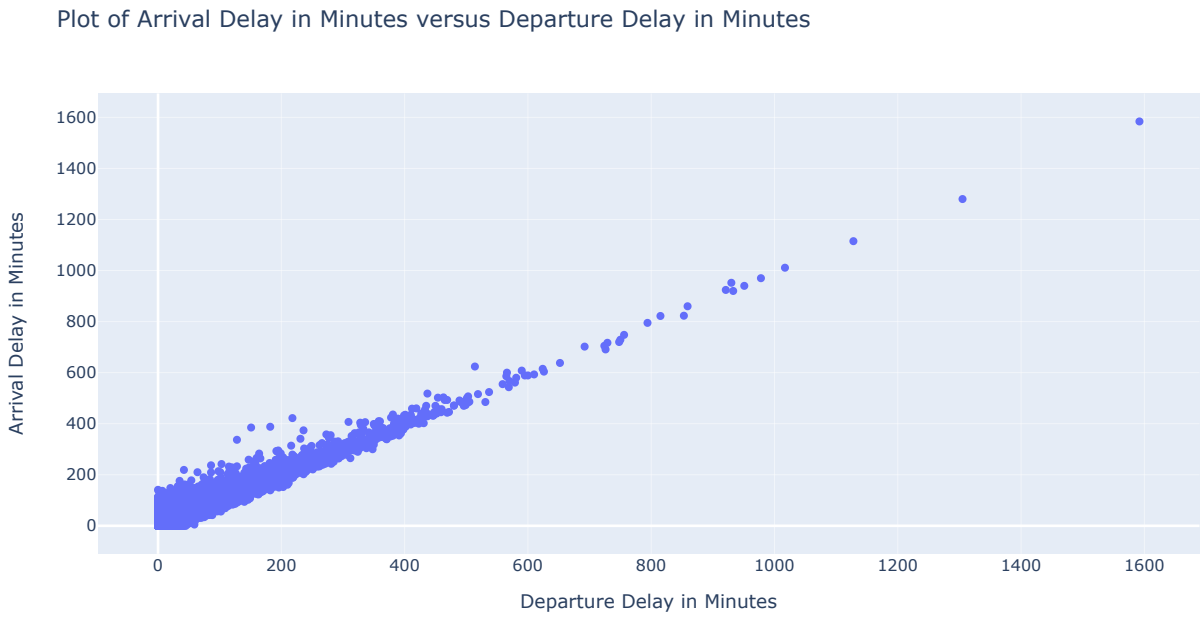
```
Number of rows: 42876
```

In [27]:
```
import plotly
plotly.offline.init_notebook_mode(connected=True)
from plotly.graph_objs import *
from plotly import tools
import plotly.graph_objects as go
import seaborn as sns
```

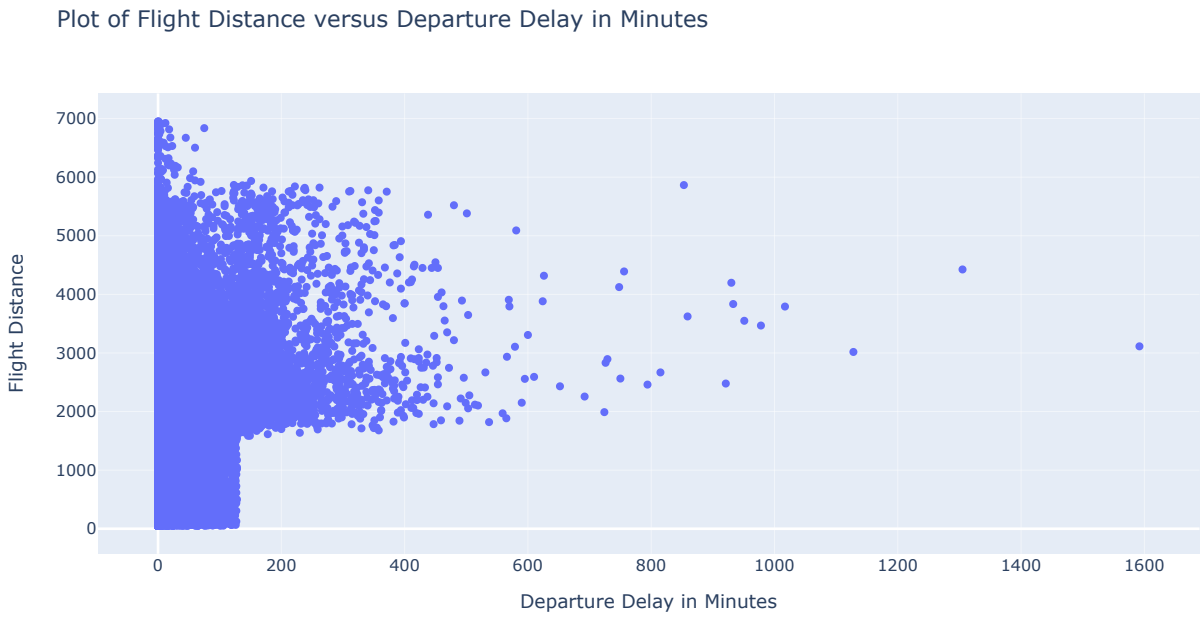
In [28]:
```
#There are no arrival dalay longer than 800 minutes among passengers older than 60.- True
if True:
    y_col = 'Arrival Delay in Minutes'
    x_col = 'Age'

    df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
    plot_data = []
    plot_data.append(go.Scatter(x= df[x_col], y= df[y_col] , mode = 'markers'))

    layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                       title = 'Plot of {} versus {}'.format(y_col, x_col))
    fig = go.Figure(data= plot_data, layout=layout)
    plotly.offline.iplot(fig)
```



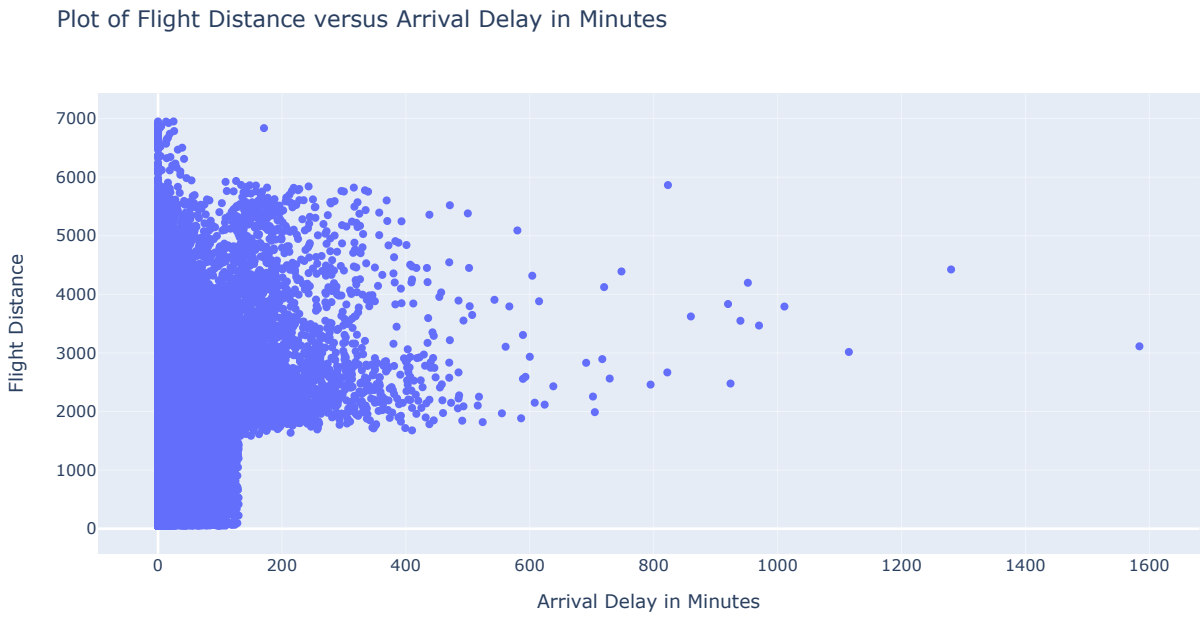
In [29]:
```
#All the flights with departure delays longer than 1000 minutes have arrival delays longer than 800 minutes. - True
if True:
    y_col = 'Arrival Delay in Minutes'
    x_col = 'Departure Delay in Minutes'

    df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
    plot_data = []
    plot_data.append(go.Scatter(x= df[x_col], y= df[y_col] , mode = 'markers'))

    layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                       title = 'Plot of {} versus {}'.format(y_col, x_col))
    fig = go.Figure(data= plot_data, layout=layout)
    plotly.offline.iplot(fig)
```



In [30]:
```
#Flights with short distance (within 1000) do not have delay more than 200 minutes. - True
if True:
    y_col = 'Flight Distance'
    x_col = 'Departure Delay in Minutes'

    df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
    plot_data = []
    plot_data.append(go.Scatter(x= df[x_col], y= df[y_col] , mode = 'markers'))

    layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                       title = 'Plot of {} versus {}'.format(y_col, x_col))
    fig = go.Figure(data= plot_data, layout=layout)
    plotly.offline.iplot(fig)
```



In [31]:
```
#Flights with short distance (within 1000) do not have delay more than 200 minutes. - True
if True:
    y_col = 'Flight Distance'
    x_col = 'Arrival Delay in Minutes'

    df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
    plot_data = []
    plot_data.append(go.Scatter(x= df[x_col], y= df[y_col] , mode = 'markers'))

    layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                       title = 'Plot of {} versus {}'.format(y_col, x_col))
    fig = go.Figure(data= plot_data, layout=layout)
    plotly.offline.iplot(fig)
```



In [32]:
```
#In general, the longer the flight distance is, the longer arrival delay is. - False
if True:
    y_col = 'Flight Distance'
    x_col = 'Arrival Delay in Minutes'

    df = invistico_Airline_data[[x_col, y_col]].sort_values(by=x_col)
    plot_data = []
    plot_data.append(go.Scatter(x= df[x_col], y= df[y_col] , mode = 'markers'))

    layout = go.Layout(xaxis = dict(title=x_col), yaxis = dict(title= y_col),
                       title = 'Plot of {} versus {}'.format(y_col, x_col))
    fig = go.Figure(data= plot_data, layout=layout)
    plotly.offline.iplot(fig)
```