In [76]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
import numpy as np
plt.style.use('ggplot')
```

In [2]:
```
#LOAD THE DATASET
dataset=pd.read_excel("Flyzy Flight Cancellation.xlsx")
```

In [3]:
```
#PRINTING THE SHAPE OF THE DATASET TO FIND OUT HOW MANY ROWNS AND COLUMNS ARE THERE
dataset.shape
```

Out[3]:
```
(3000, 14)
```

In [4]:
```
#DISPLAY THE FIRST FEW ROWS OF THE DATASET.
dataset.head()
```

Out[4]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	Airline D	475	Airport 3	Airport 2	4
1	4791965	Airline E	538	Airport 5	Airport 4	12
2	2991718	Airline C	565	Airport 1	Airport 2	17
3	4220106	Airline E	658	Airport 5	Airport 3	1
4	2263008	Airline E	566	Airport 2	Airport 2	19

In [5]:
```
#DISPLAY THE FIRST FEW ROWS OF THE DATASET.
dataset.head()
```

Out[5]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	Airline D	475	Airport 3	Airport 2	4
1	4791965	Airline E	538	Airport 5	Airport 4	12
2	2991718	Airline C	565	Airport 1	Airport 2	17
3	4220106	Airline E	658	Airport 5	Airport 3	1
4	2263008	Airline E	566	Airport 2	Airport 2	19

In [6]:
```
dataset.describe()
```

Out[6]:

	Flight ID	Flight_Distance	Scheduled_Departure_Time	Day_of_Week	Month	Weat
count	3.000000e+03	3000.000000	3000.000000	3000.000000	3000.000000	30
mean	4.997429e+06	498.909333	11.435000	3.963000	6.381000	
std	2.868139e+06	98.892266	6.899298	2.016346	3.473979	
min	3.681000e+03	138.000000	0.000000	1.000000	1.000000	
25%	2.520313e+06	431.000000	6.000000	2.000000	3.000000	
50%	5.073096e+06	497.000000	12.000000	4.000000	6.000000	
75%	7.462026e+06	566.000000	17.000000	6.000000	9.000000	
max	9.999011e+06	864.000000	23.000000	7.000000	12.000000	

In [7]:
```
#DISPLAY THE SUMMARY OF THE DATASET.
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Flight ID                     3000 non-null   int64
 1   Airline                       3000 non-null   object
 2   Flight_Distance               3000 non-null   int64
 3   Origin_Airport                3000 non-null   object
 4   Destination_Airport           3000 non-null   object
 5   Scheduled_Departure_Time      3000 non-null   int64
 6   Day_of_Week                   3000 non-null   int64
 7   Month                         3000 non-null   int64
 8   Airplane_Type                 3000 non-null   object
 9   Weather_Score                 3000 non-null   float64
 10  Previous_Flight_Delay_Minutes 3000 non-null   float64
 11  Airline_Rating                3000 non-null   float64
 12  Passenger_Load                3000 non-null   float64
 13  Flight_Cancelled              3000 non-null   int64
dtypes: float64(4), int64(6), object(4)
memory usage: 328.3+ KB
```

In [8]:
```
#CHECKING THE TYPE OF DATASET.
type(dataset)
```

Out[8]:
```
pandas.core.frame.DataFrame
```

In [9]:
```
#DISPLAYING THE DATA TYPES OF EACH COLUMN.
dataset.dtypes
```

Out[9]:
```
Flight ID                          int64
Airline                           object
Flight_Distance                    int64
Origin_Airport                    object
Destination_Airport               object
Scheduled_Departure_Time            int64
Day_of_Week                        int64
Month                              int64
Airplane_Type                     object
Weather_Score                    float64
Previous_Flight_Delay_Minutes    float64
Airline_Rating                   float64
Passenger_Load                   float64
Flight_Cancelled                   int64
dtype: object
```

In [10]:
```
#CHANGING ALL CATEGORICAL COLUMNS TO NUMERIC COLUMNS FOR COMPATABILITY WITH MACHINE
from sklearn.preprocessing import LabelEncoder
label_encoder=LabelEncoder()
```

In [11]:
```
dataset['Airline']=label_encoder.fit_transform(dataset['Airline'])
dataset['Origin_Airport']=label_encoder.fit_transform(dataset['Origin_Airport'])
dataset['Destination_Airport']=label_encoder.fit_transform(dataset['Destination_Air
dataset['Airplane_Type']=label_encoder.fit_transform(dataset['Airplane_Type'])
```

In [12]:
```
#CHECKING IF ALL COLUMNS ARE NOW NUMERIC.
dataset.dtypes
```

Out[12]:
```
Flight ID                          int64
Airline                            int32
Flight_Distance                    int64
Origin_Airport                     int32
Destination_Airport                int32
Scheduled_Departure_Time            int64
Day_of_Week                        int64
Month                              int64
Airplane_Type                      int32
Weather_Score                    float64
Previous_Flight_Delay_Minutes    float64
Airline_Rating                   float64
Passenger_Load                   float64
Flight_Cancelled                   int64
dtype: object
```

In [13]:
```
#CHECKING FOR MISSING VALUES.
dataset.isnull().sum()
```

Out[13]:
```
Flight ID                        0
Airline                          0
Flight_Distance                  0
Origin_Airport                   0
Destination_Airport              0
Scheduled_Departure_Time         0
Day_of_Week                      0
Month                            0
Airplane_Type                    0
Weather_Score                    0
Previous_Flight_Delay_Minutes    0
Airline_Rating                   0
Passenger_Load                   0
Flight_Cancelled                 0
dtype: int64
```

In [14]:
```
#CHECKING FOR ANY DUPLICATED ROWS IN EVERY COLUMN AND DROPPING THEM.
dataset.drop_duplicates(subset='Flight ID', keep='first')
```

Out[14]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

3000 rows × 14 columns

In [15]:
```
dataset.drop_duplicates(subset='Airline', keep='first')
```

Out[15]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
2	2991718	2	565	0	0	1
5	9450813	1	446	3	3	
12	6765292	0	371	0	0	1

In [16]:
```
dataset.drop_duplicates(subset='Flight_Distance', keep='first')
```

Out[16]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2885	8862534	4	285	3	0	
2892	4858025	0	210	1	2	
2919	3395017	2	306	2	0	
2990	3647668	4	207	0	1	
2993	5680807	0	829	0	3	

470 rows × 14 columns

In [17]:
```
dataset.drop_duplicates(subset='Origin_Airport', keep='first')
```

Out[17]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
2	2991718	2	565	0	0	17
4	2263008	4	566	1	0	19
5	9450813	1	446	3	3	3

In [18]:
```
dataset.drop_duplicates(subset='Destination_Airport', keep='first')
```

Out[18]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
3	4220106	4	658	4	1	1
5	9450813	1	446	3	3	3

In [19]:
```
dataset.drop_duplicates(subset='Scheduled_Departure_Time', keep='first')
```

Out[19]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Ti
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
5	9450813	1	446	3	3	
7	5641104	2	472	1	2	
8	5559237	2	591	2	3	
9	6633214	4	442	2	3	
10	9421450	3	422	3	3	
13	3779836	1	452	2	0	
16	3553627	3	322	0	3	
18	3344010	0	499	3	1	
20	1149015	1	450	1	0	
30	8069823	2	638	1	3	
33	7813168	0	451	3	3	
39	2314231	0	595	0	0	
43	6409414	1	446	3	0	
45	4855327	0	396	3	0	
49	9791618	3	389	4	0	
75	3110200	1	578	1	0	
104	1962565	0	435	3	0	
120	8581059	0	417	0	0	
137	4298874	1	437	1	0	

In [20]:
```
dataset.drop_duplicates(subset='Day_of_Week', keep='first')
```

Out[20]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
2	2991718	2	565	0	0	17
4	2263008	4	566	1	0	19
5	9450813	1	446	3	3	3
6	3515052	2	688	3	0	12
8	5559237	2	591	2	3	15

In [21]:
```
dataset.drop_duplicates(subset='Month', keep='first')
```

Out[21]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
2	2991718	2	565	0	0	1
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	1
5	9450813	1	446	3	3	
10	9421450	3	422	3	3	
15	517306	1	526	2	3	
17	9978315	4	583	3	2	1
24	5247488	0	329	4	0	
37	2643614	4	733	2	0	1
40	4987114	3	610	4	0	

In [22]:
```
dataset.drop_duplicates(subset='Airplane_Type', keep='first')
```

Out[22]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
4	2263008	4	566	1	0	1
5	9450813	1	446	3	3	
11	2858209	2	364	3	0	1

In [23]:
```
dataset.drop_duplicates(subset='Weather_Score', keep='first')
```

Out[23]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2999 rows × 14 columns

In [24]:
```
dataset.drop_duplicates(subset='Previous_Flight_Delay_Minutes', keep='first')
```

Out[24]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2704	8378528	0	468	0	0	
2720	7464476	0	530	0	0	
2839	9897841	0	490	0	0	
2886	6758398	4	590	4	0	
2979	630473	0	484	0	0	

304 rows × 14 columns

In [25]:
```
dataset.drop_duplicates(subset='Airline_Rating', keep='first')
```

Out[25]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2999 rows × 14 columns

In [26]:
```
dataset.drop_duplicates(subset='Passenger_Load', keep='first')
```

Out[26]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2995 rows × 14 columns

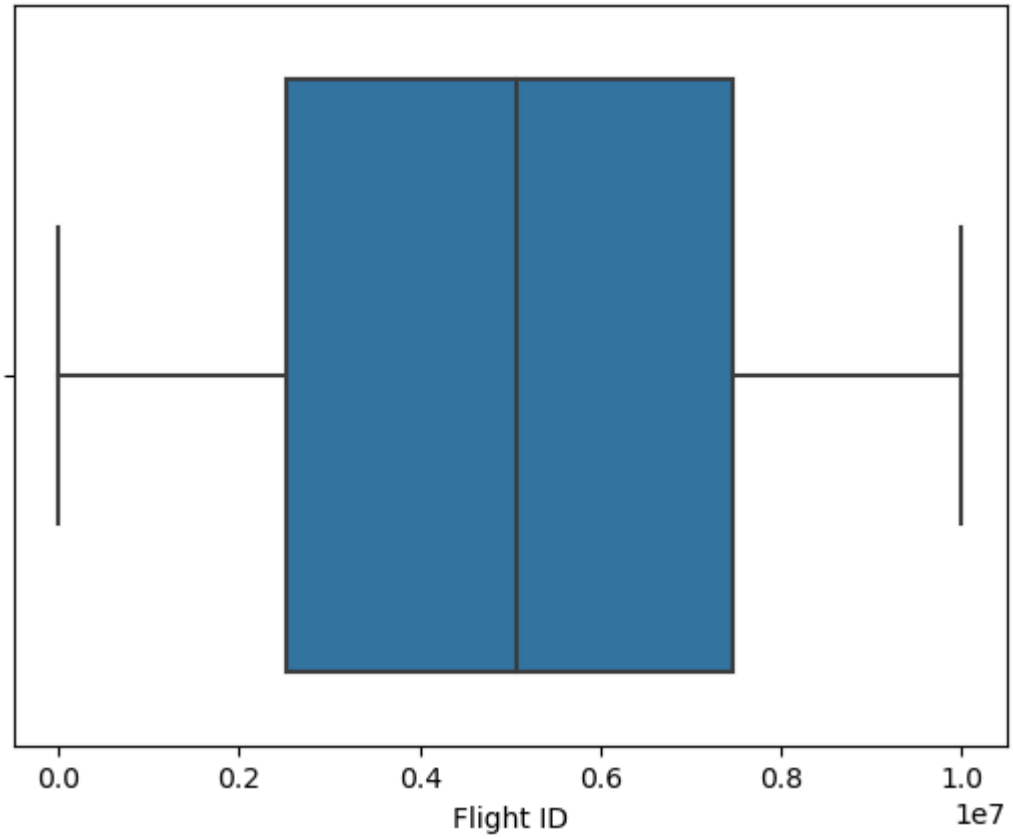
In [27]:
```
dataset.drop_duplicates(subset='Flight_Cancelled', keep='first')
```

Out[27]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12

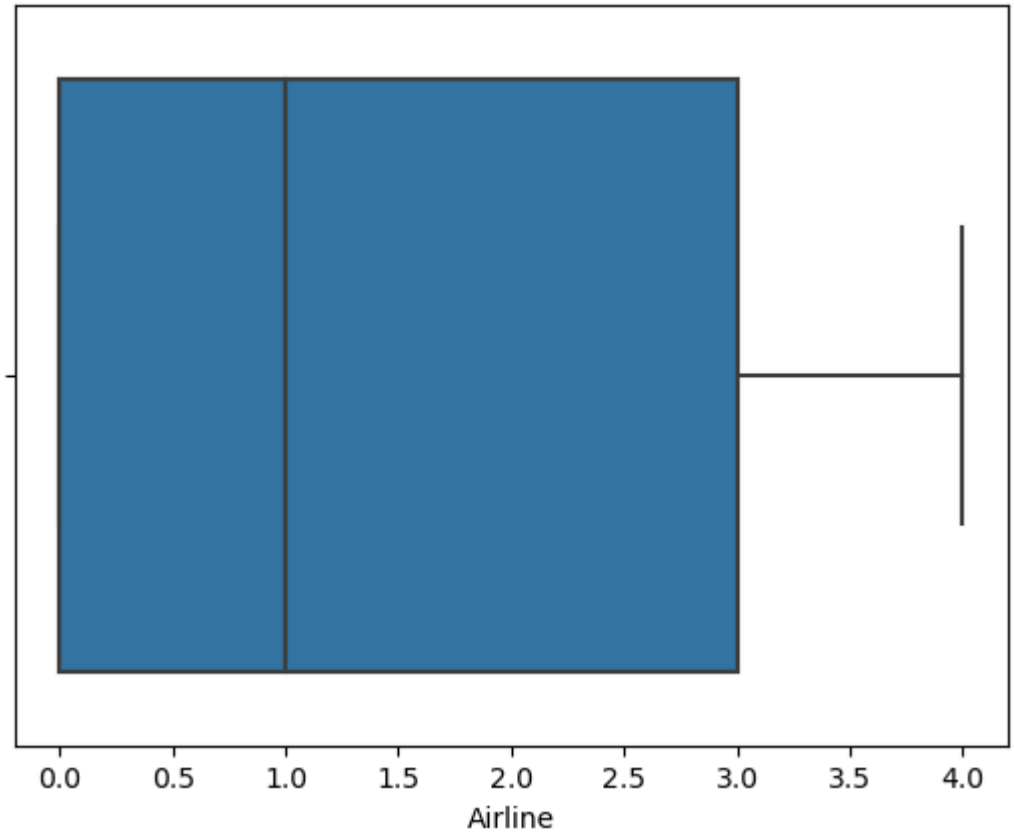
In [28]:
```
#CHECKING FOR OUTLIERS USING A BOXPLOT.
sns.boxplot(x=dataset['Flight ID'])
```

Out[28]:
```
<Axes: xlabel='Flight ID'>
```



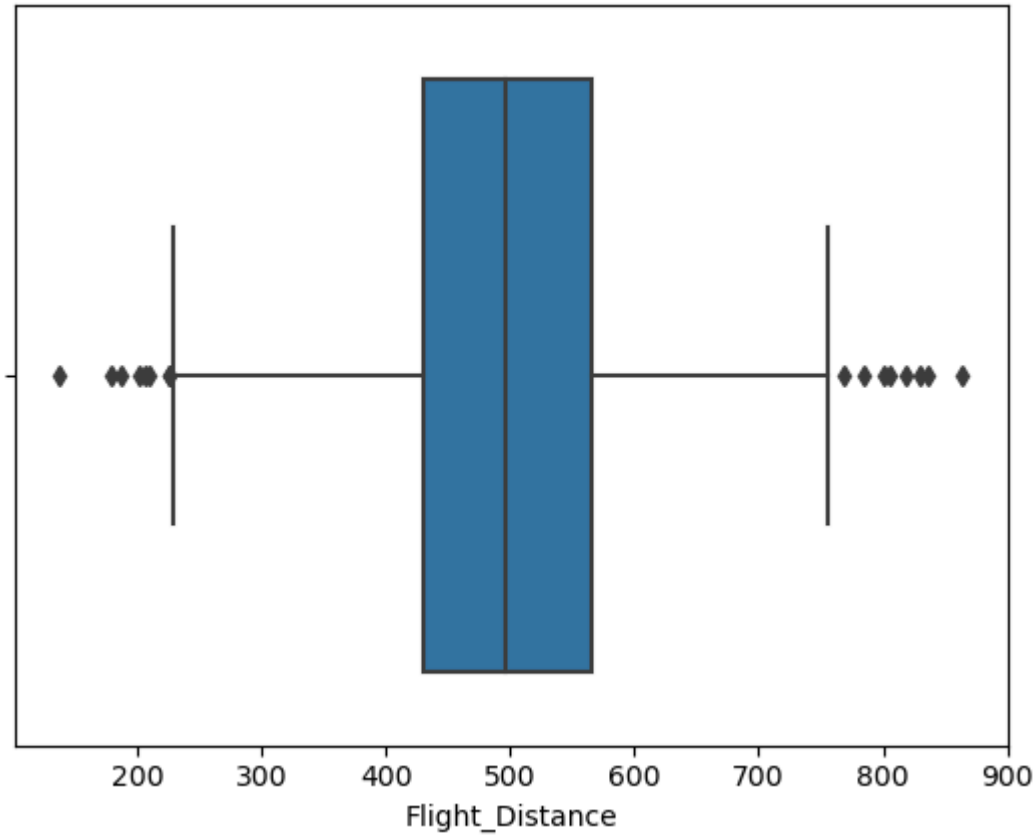
In [29]:
```
sns.boxplot(x=dataset['Airline'])
```

Out[29]:
```
<Axes: xlabel='Airline'>
```



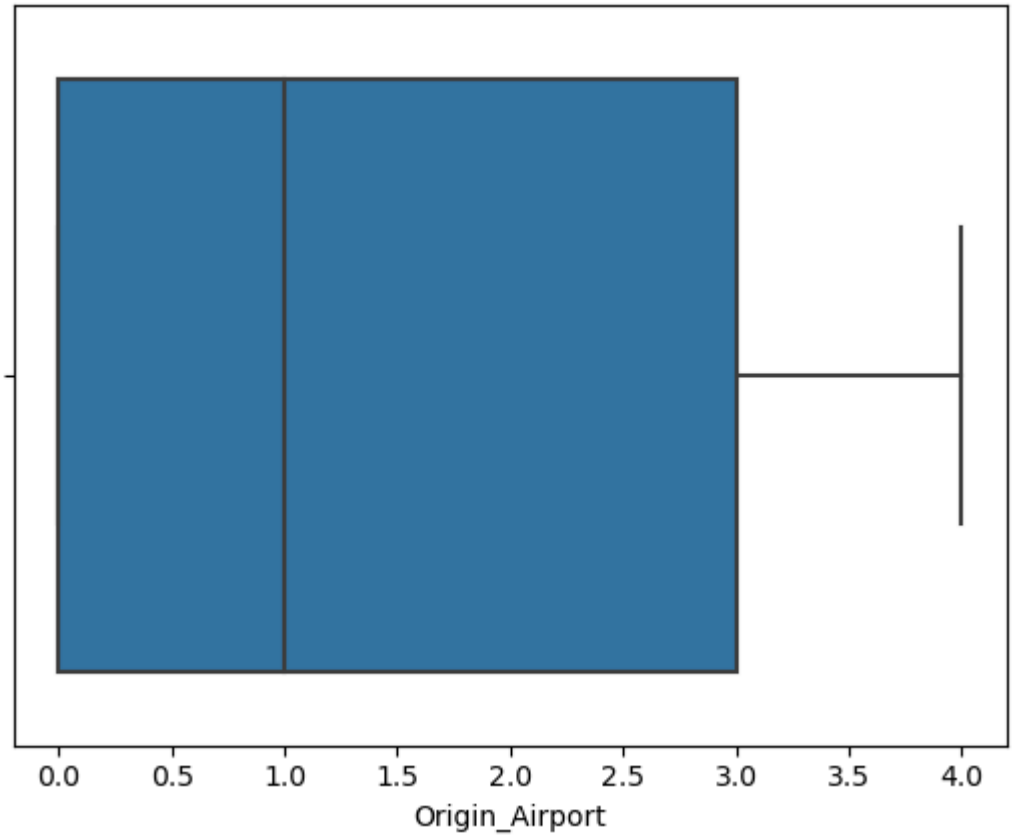
In [30]:
```
sns.boxplot(x=dataset['Flight_Distance'])
```

Out[30]:
```
<Axes: xlabel='Flight_Distance'>
```



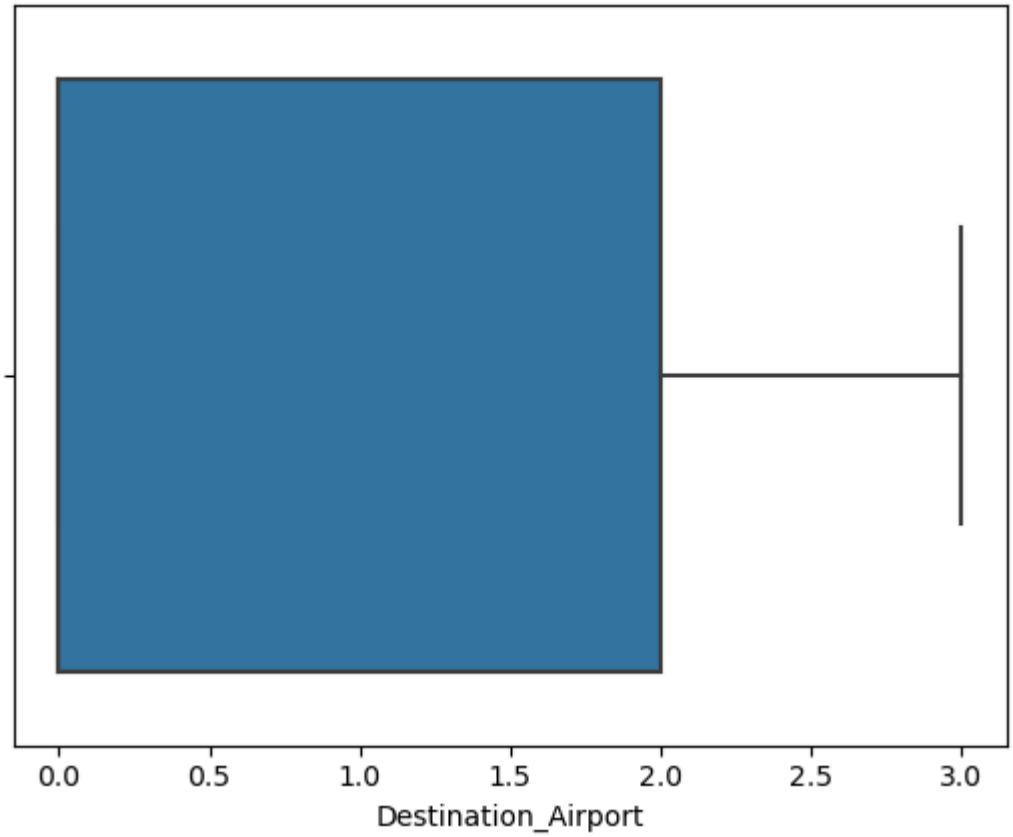
In [31]:
```
sns.boxplot(x=dataset['Origin_Airport'])
```

Out[31]:
```
<Axes: xlabel='Origin_Airport'>
```



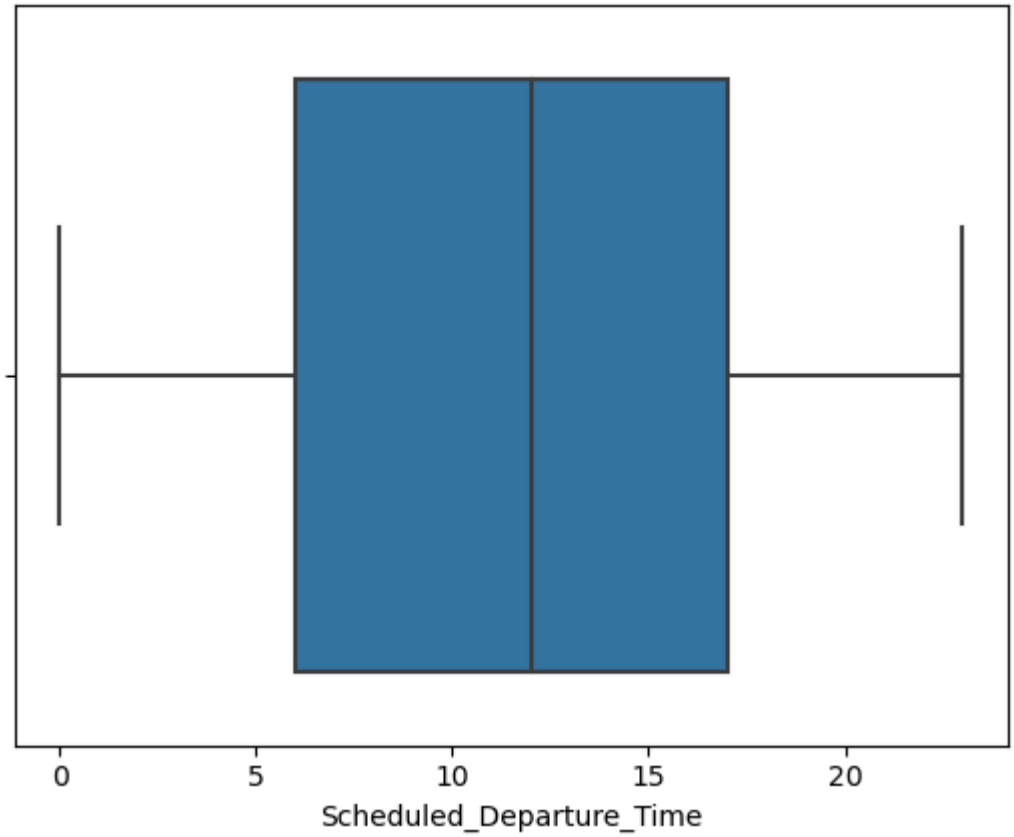
In [32]:
```
sns.boxplot(x=dataset['Destination_Airport'])
```

Out[32]:
```
<Axes: xlabel='Destination_Airport'>
```



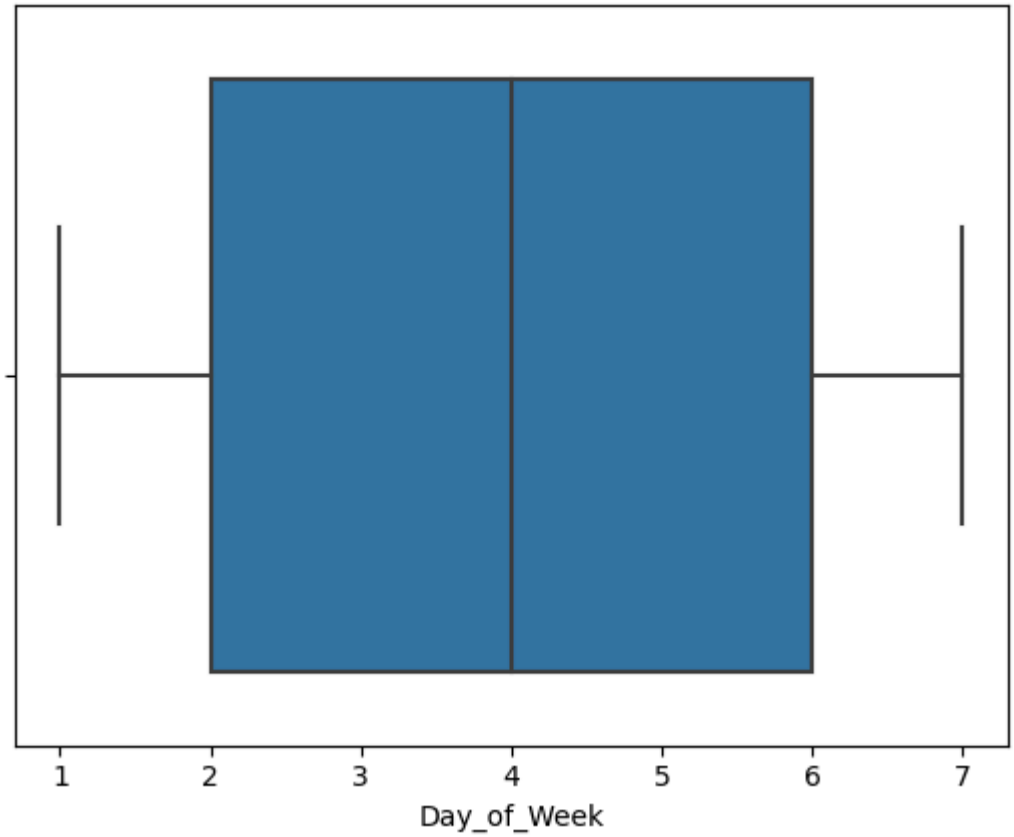
In [33]:
```
sns.boxplot(x=dataset['Scheduled_Departure_Time'])
```

Out[33]:
```
<Axes: xlabel='Scheduled_Departure_Time'>
```



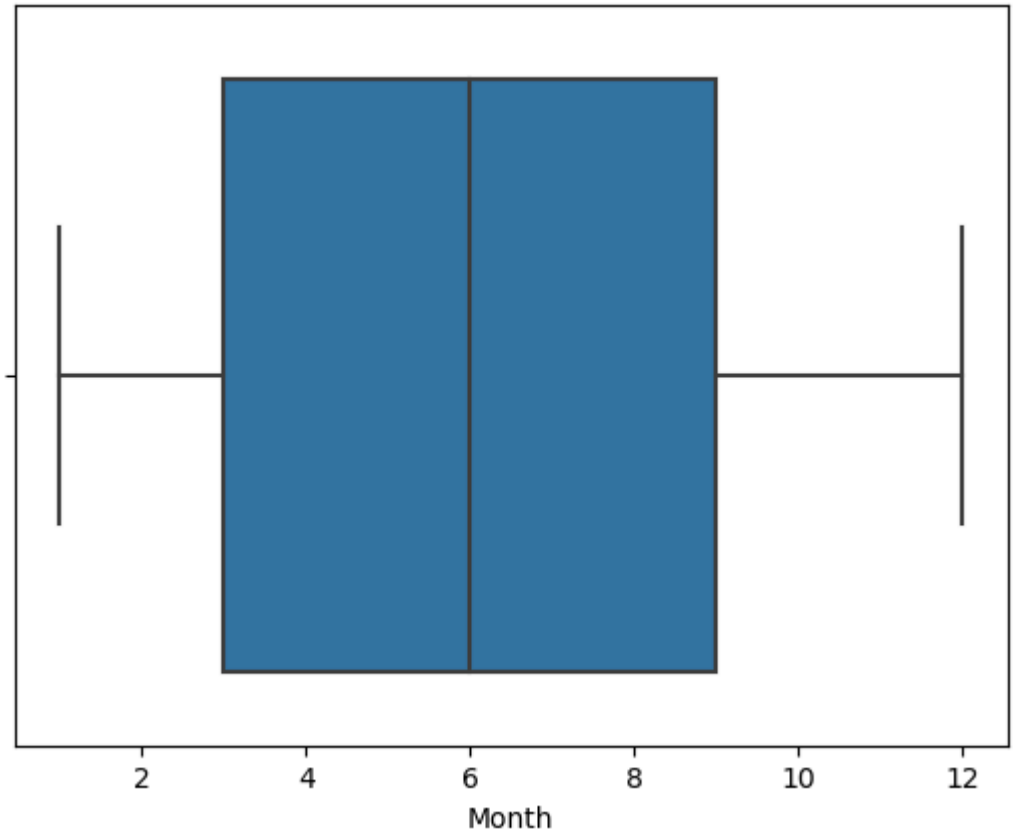
In [34]:
```
sns.boxplot(x=dataset['Day_of_Week'])
```

Out[34]:
```
<Axes: xlabel='Day_of_Week'>
```



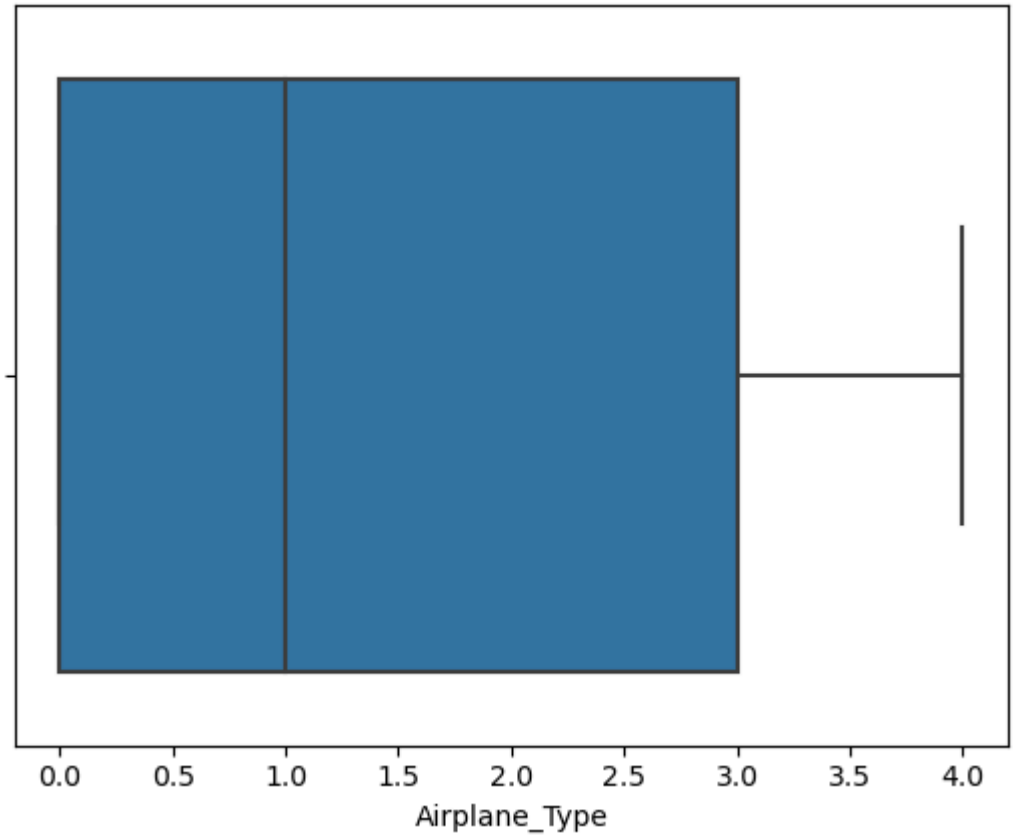
In [35]:
```
sns.boxplot(x=dataset['Month'])
```

Out[35]:
```
<Axes: xlabel='Month'>
```



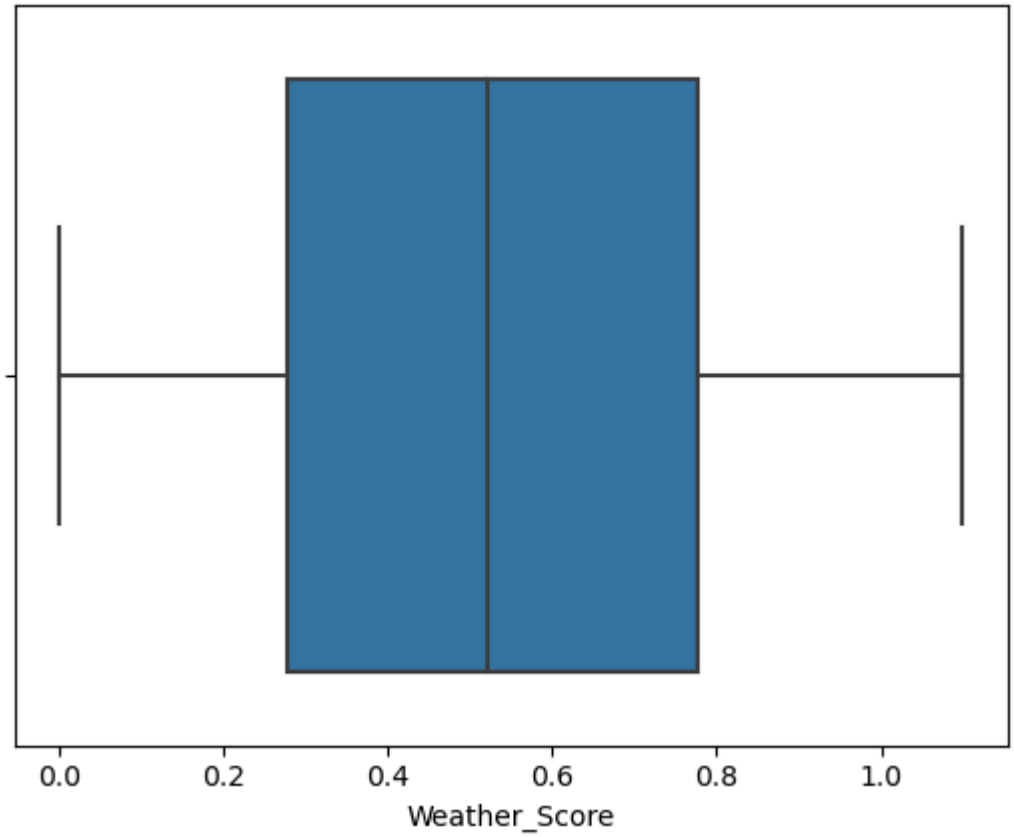
In [36]:
```
sns.boxplot(x=dataset['Airplane_Type'])
```

Out[36]:
```
<Axes: xlabel='Airplane_Type'>
```



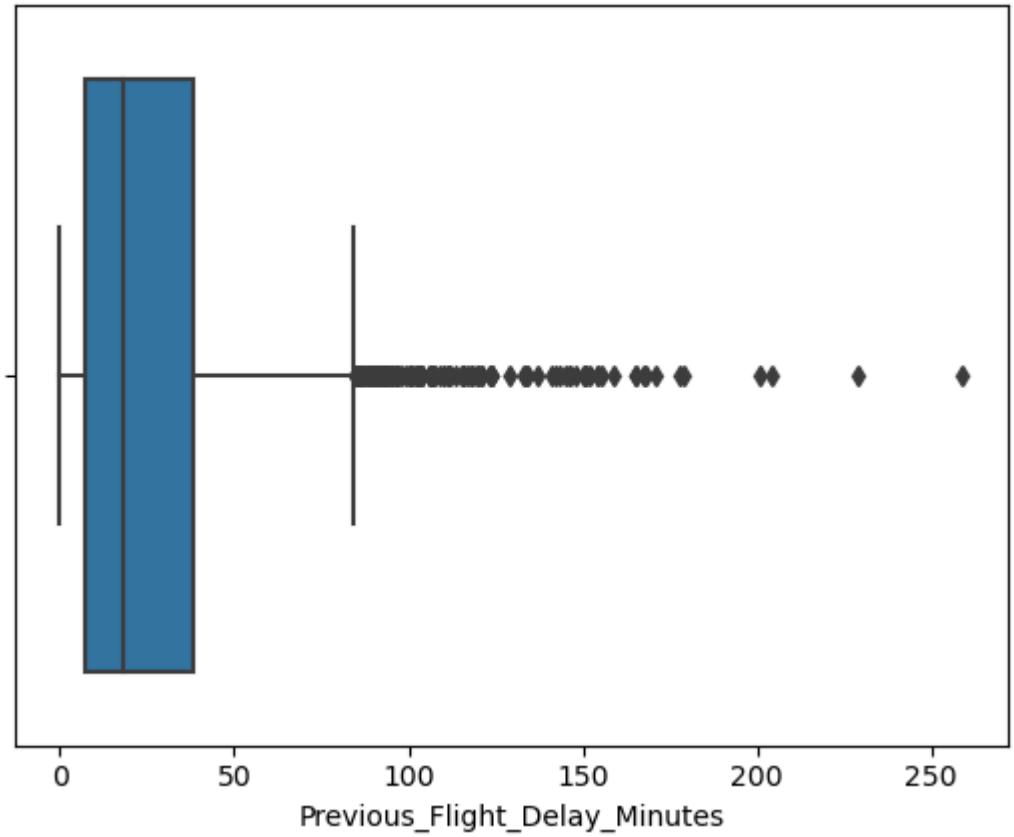
In [37]:
```
sns.boxplot(x=dataset['Weather_Score'])
```

Out[37]:
```
<Axes: xlabel='Weather_Score'>
```



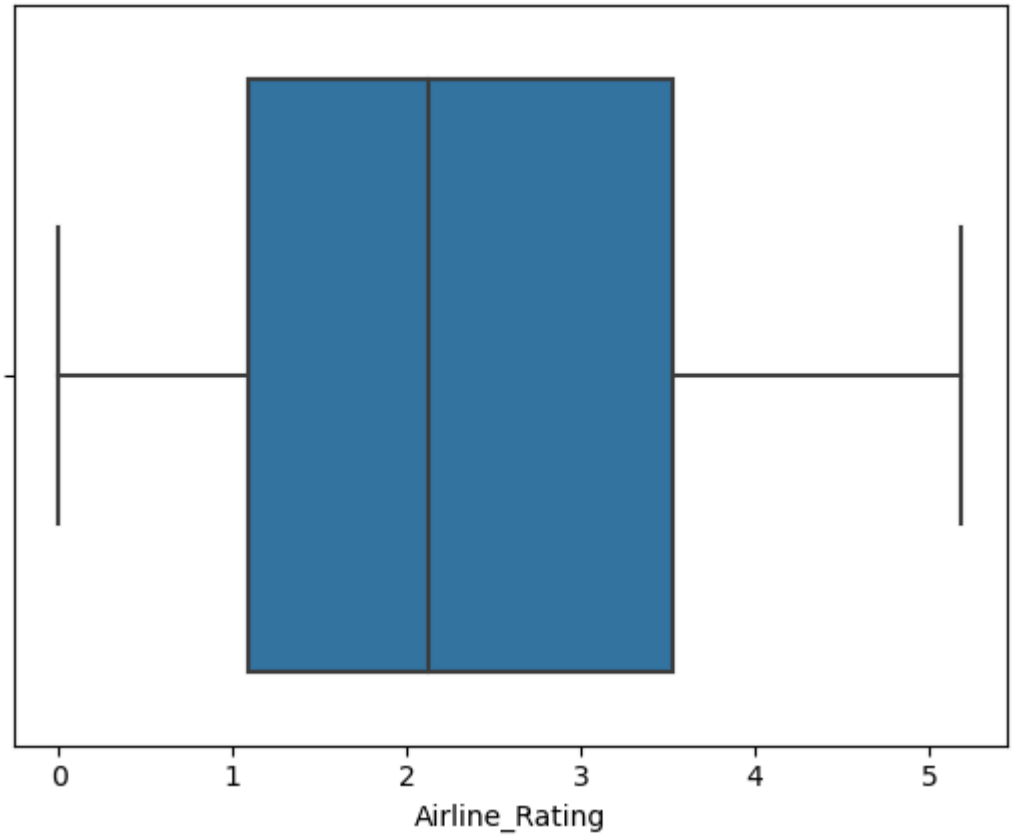
In [38]:
```
sns.boxplot(x=dataset['Previous_Flight_Delay_Minutes'])
```

Out[38]:
```
<Axes: xlabel='Previous_Flight_Delay_Minutes'>
```



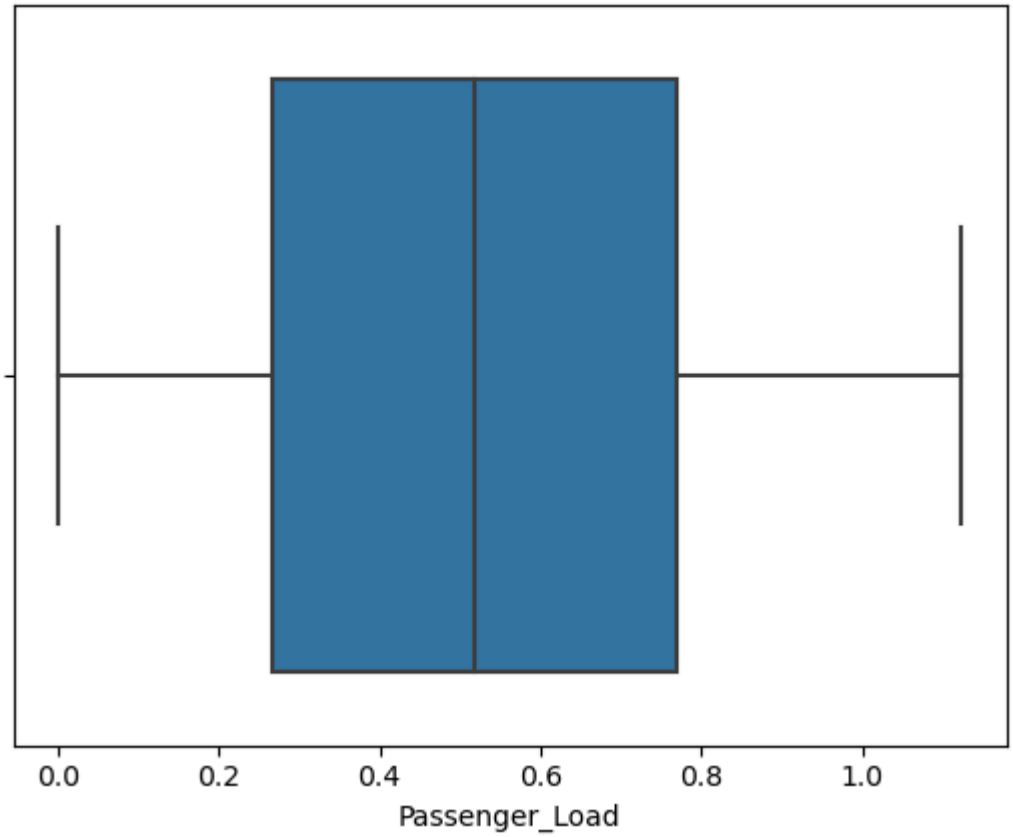
In [39]:
```
sns.boxplot(x=dataset['Airline_Rating'])
```

Out[39]:
```
<Axes: xlabel='Airline_Rating'>
```



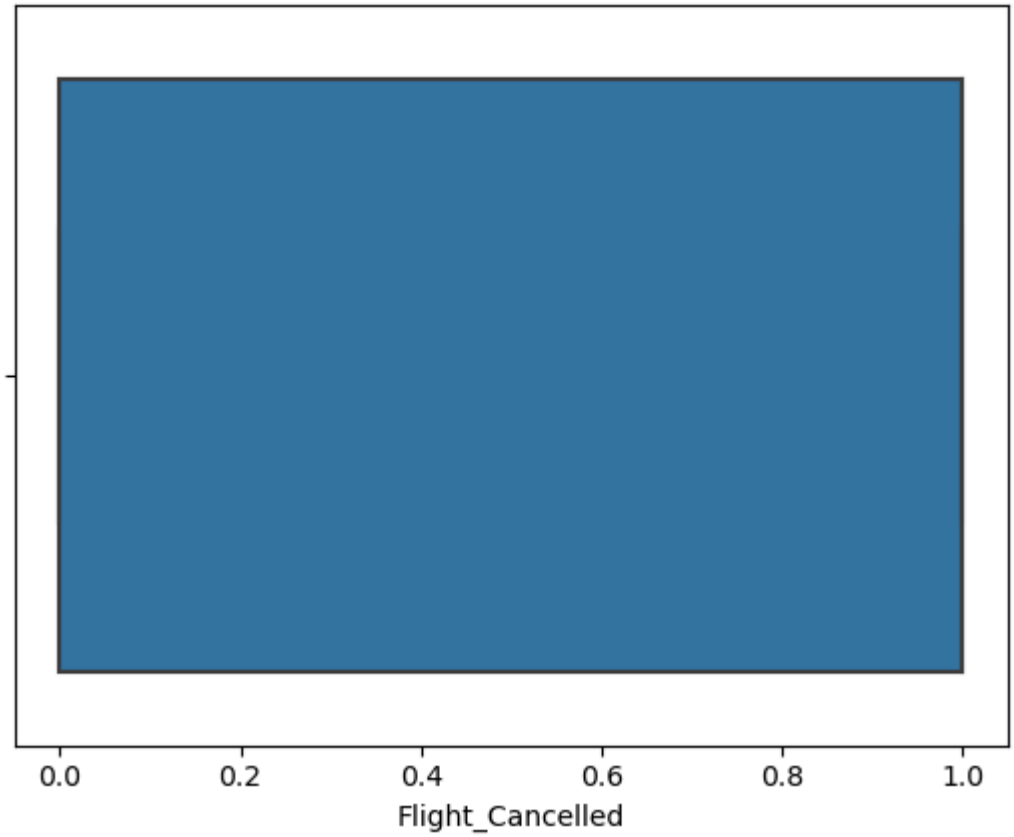
In [40]:
```
sns.boxplot(x=dataset['Passenger_Load'])
```

Out[40]:
```
<Axes: xlabel='Passenger_Load'>
```



In [41]:
```
sns.boxplot(x=dataset['Flight_Cancelled'])
```

Out[41]:
```
<Axes: xlabel='Flight_Cancelled'>
```



In [49]:
```
#DROPPING OUTLIERS.
from scipy.stats import zscore
```

In [52]:
```
z_scores=np.abs(zscore(dataset))
```

In [53]:
```
threshold=3
```

In [54]:
```
dataset_clean=dataset[(z_scores < threshold).all(axis=1)]
```

In [55]:
```
print(dataset_clean)
```

```
      Flight ID  Airline  Flight_Distance  Origin_Airport  \
0       7319483        3              475               2
1       4791965        4              538               4
2       2991718        2              565               0
3       4220106        4              658               4
4       2263008        4              566               1
...         ...      ...              ...             ...
2995    1265781        3              395               1
2996    5440150        4              547               0
2997     779080        2              461               0
2998    4044431        1              464               2
2999    2806578        0              369               0

      Destination_Airport  Scheduled_Departure_Time  Day_of_Week  Month  \
0                       0                         4            6      1
1                       2                        12            1      6
2                       0                        17            3      9
3                       1                         1            1      8
4                       0                        19            7     12
...                   ...                       ...          ...    ...
2995                    1                         0            6      1
2996                    2                        22            4      7
2997                    1                         8            3      1
2998                    1                         5            5      3
2999                    0                         1            1     10

      Airplane_Type  Weather_Score  Previous_Flight_Delay_Minutes  \
0                 2       0.225122                        5.00000
1                 1       0.060346                       68.00000
2                 2       0.093920                       18.00000
3                 1       0.656750                       13.00000
4                 4       0.505211                        4.00000
...             ...            ...                            ...
2995              1       0.190018                        1.00000
2996              4       0.719271                       91.00000
2997              1       0.458724                        3.00000
2998              4       0.443373                       46.00000
2999              0       0.704563                       18.66667

      Airline_Rating  Passenger_Load  Flight_Cancelled
0           2.151974        0.477202                 0
1           1.600779        0.159718                 1
2           4.406848        0.256803                 0
3           0.998757        0.504077                 1
4           3.806206        0.019638                 0
...              ...             ...               ...
2995        2.451216        0.283440                 1
2996        0.027039        0.665294                 1
2997        1.131315        0.991307                 0
2998        0.968651        0.254808                 1
2999        1.879411        0.532486                 1

[2939 rows x 14 columns]
```

In [56]:
```
#ENSURE APPROPRIATE DATA TYPES.
print(dataset.dtypes)
```

```
Flight ID                          int64
Airline                            int32
Flight_Distance                    int64
Origin_Airport                     int32
Destination_Airport                int32
Scheduled_Departure_Time           int64
Day_of_Week                        int64
Month                              int64
Airplane_Type                      int32
Weather_Score                    float64
Previous_Flight_Delay_Minutes    float64
Airline_Rating                   float64
Passenger_Load                   float64
Flight_Cancelled                   int64
dtype: object
```

In [57]:
```
#CHECKING FOR DUPLICATED ROWS IN THE DATASET AND DROPPING THEM.
dataset.drop_duplicates(subset='Flight ID', keep='first')
```

Out[57]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

3000 rows × 14 columns

In [58]:
```
dataset.drop_duplicates(subset='Airline', keep='first')
```

Out[58]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
2	2991718	2	565	0	0	1
5	9450813	1	446	3	3	
12	6765292	0	371	0	0	1

In [59]:
```
dataset.drop_duplicates(subset='Flight_Distance', keep='first')
```

Out[59]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2885	8862534	4	285	3	0	
2892	4858025	0	210	1	2	
2919	3395017	2	306	2	0	
2990	3647668	4	207	0	1	
2993	5680807	0	829	0	3	

470 rows × 14 columns

In [60]:
```
dataset.drop_duplicates(subset='Origin_Airport', keep='first')
```

Out[60]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
2	2991718	2	565	0	0	17
4	2263008	4	566	1	0	19
5	9450813	1	446	3	3	3

In [61]:
```
dataset.drop_duplicates(subset='Destination_Airport', keep='first')
```

Out[61]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
3	4220106	4	658	4	1	1
5	9450813	1	446	3	3	3

In [62]:
```
dataset.drop_duplicates(subset='Scheduled_Departure_Time', keep='first')
```

Out[62]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Ti
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
5	9450813	1	446	3	3	
7	5641104	2	472	1	2	
8	5559237	2	591	2	3	
9	6633214	4	442	2	3	
10	9421450	3	422	3	3	
13	3779836	1	452	2	0	
16	3553627	3	322	0	3	
18	3344010	0	499	3	1	
20	1149015	1	450	1	0	
30	8069823	2	638	1	3	
33	7813168	0	451	3	3	
39	2314231	0	595	0	0	
43	6409414	1	446	3	0	
45	4855327	0	396	3	0	
49	9791618	3	389	4	0	
75	3110200	1	578	1	0	
104	1962565	0	435	3	0	
120	8581059	0	417	0	0	
137	4298874	1	437	1	0	

In [63]:
```
dataset.drop_duplicates(subset='Day_of_Week', keep='first')
```

Out[63]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12
2	2991718	2	565	0	0	17
4	2263008	4	566	1	0	19
5	9450813	1	446	3	3	3
6	3515052	2	688	3	0	12
8	5559237	2	591	2	3	15

In [64]:
```
dataset.drop_duplicates(subset='Month', keep='first')
```

Out[64]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
2	2991718	2	565	0	0	1
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	1
5	9450813	1	446	3	3	
10	9421450	3	422	3	3	
15	517306	1	526	2	3	
17	9978315	4	583	3	2	1
24	5247488	0	329	4	0	
37	2643614	4	733	2	0	1
40	4987114	3	610	4	0	

In [65]:
```
dataset.drop_duplicates(subset='Airplane_Type', keep='first')
```

Out[65]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Tim
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	1
4	2263008	4	566	1	0	1
5	9450813	1	446	3	3	
11	2858209	2	364	3	0	1

In [66]:
```
dataset.drop_duplicates(subset='Weather_Score', keep='first')
```

Out[66]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2999 rows × 14 columns

In [67]:
```
dataset.drop_duplicates(subset='Previous_Flight_Delay_Minutes', keep='first')
```

Out[67]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2704	8378528	0	468	0	0	
2720	7464476	0	530	0	0	
2839	9897841	0	490	0	0	
2886	6758398	4	590	4	0	
2979	630473	0	484	0	0	

304 rows × 14 columns

In [68]:
```
dataset.drop_duplicates(subset='Airline_Rating', keep='first')
```

Out[68]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2999 rows × 14 columns

In [69]:
```
dataset.drop_duplicates(subset='Passenger_Load', keep='first')
```

Out[69]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_T
0	7319483	3	475	2	0	
1	4791965	4	538	4	2	
2	2991718	2	565	0	0	
3	4220106	4	658	4	1	
4	2263008	4	566	1	0	
...	...	...	...	...	...	
2995	1265781	3	395	1	1	
2996	5440150	4	547	0	2	
2997	779080	2	461	0	1	
2998	4044431	1	464	2	1	
2999	2806578	0	369	0	0	

2995 rows × 14 columns

In [70]:
```
dataset.drop_duplicates(subset='Flight_Cancelled', keep='first')
```

Out[70]:

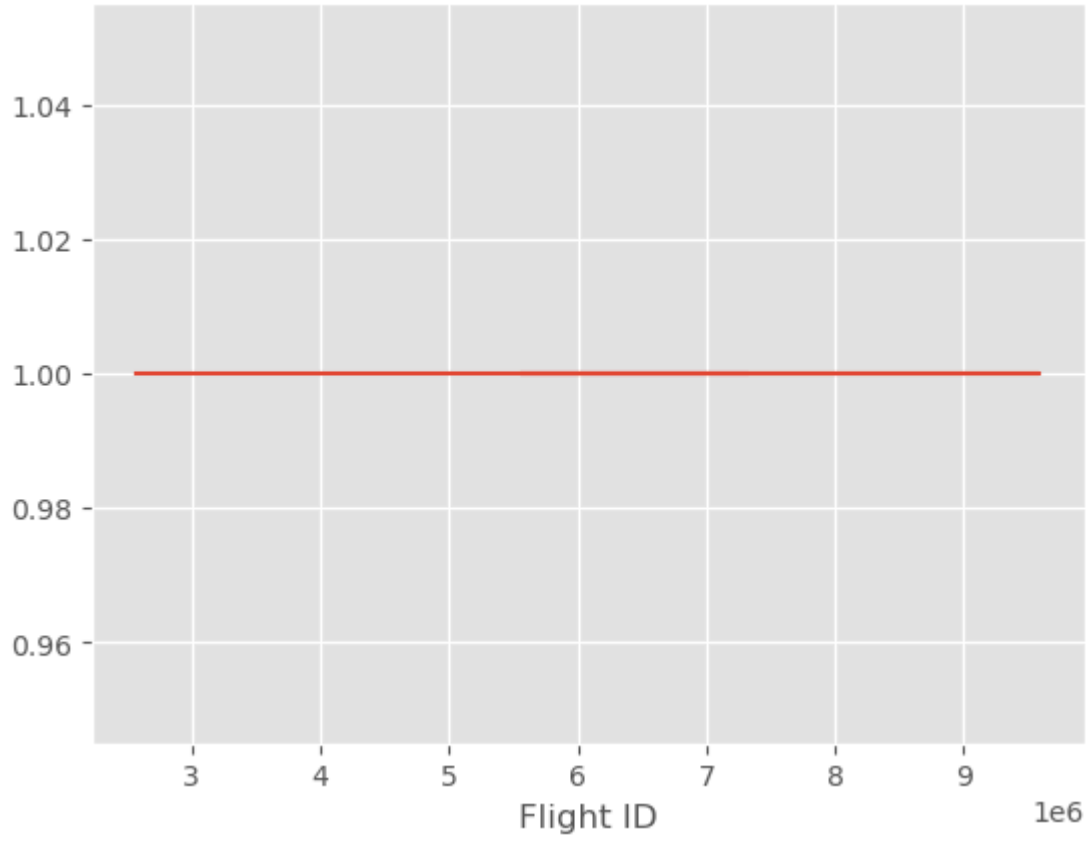
	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	3	475	2	0	4
1	4791965	4	538	4	2	12

In [71]:
```
#CHECKING FOR ANY MISSING VALUES.
dataset.isnull().sum()
```

Out[71]:
```
Flight ID                       0
Airline                         0
Flight_Distance                 0
Origin_Airport                  0
Destination_Airport             0
Scheduled_Departure_Time        0
Day_of_Week                     0
Month                           0
Airplane_Type                   0
Weather_Score                   0
Previous_Flight_Delay_Minutes   0
Airline_Rating                  0
Passenger_Load                  0
Flight_Cancelled                0
dtype: int64
```

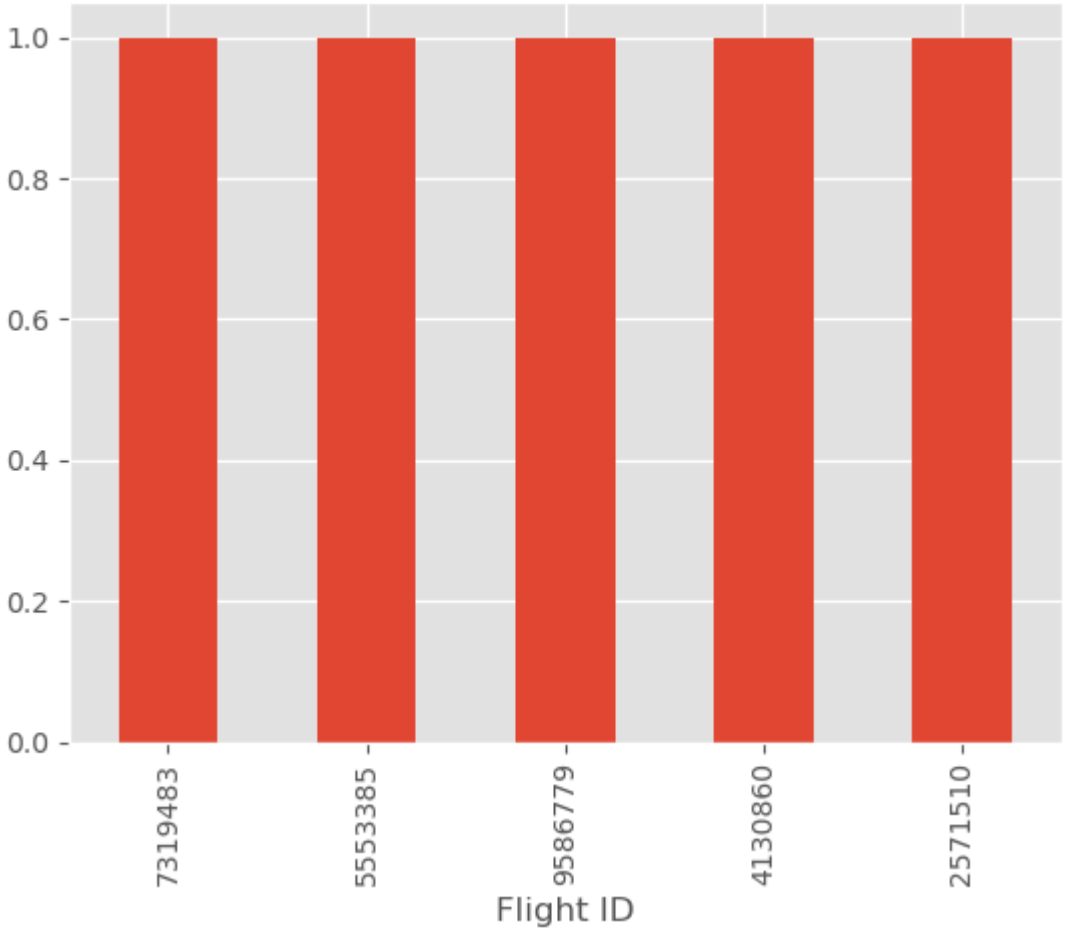
In [77]:
```
#DISPLAYING THE COUNTS OF THE FIVE MOST COMMON VALUES IN THE SPECIFIED COLUMN OF TH
dataset['Flight ID'].value_counts().head().plot()
```

Out[77]:
```
<Axes: xlabel='Flight ID'>
```



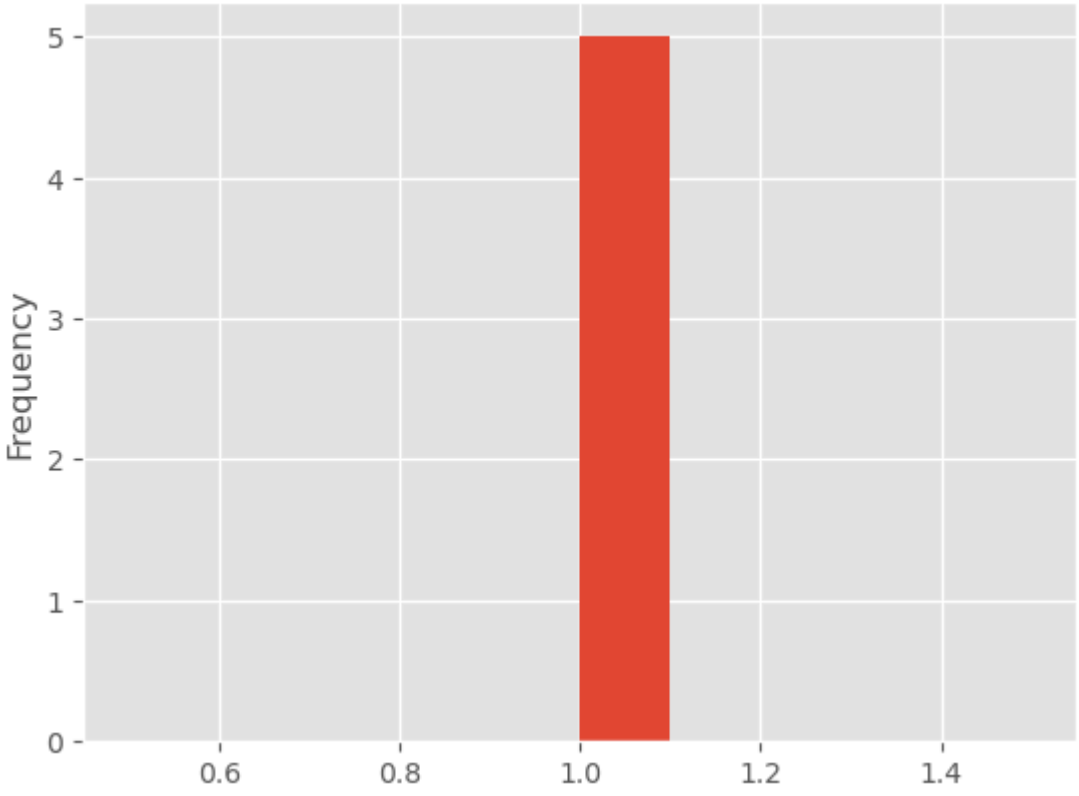
In [78]:
```
dataset['Flight ID'].value_counts().head().plot(kind='bar')
```

Out[78]:
```
<Axes: xlabel='Flight ID'>
```



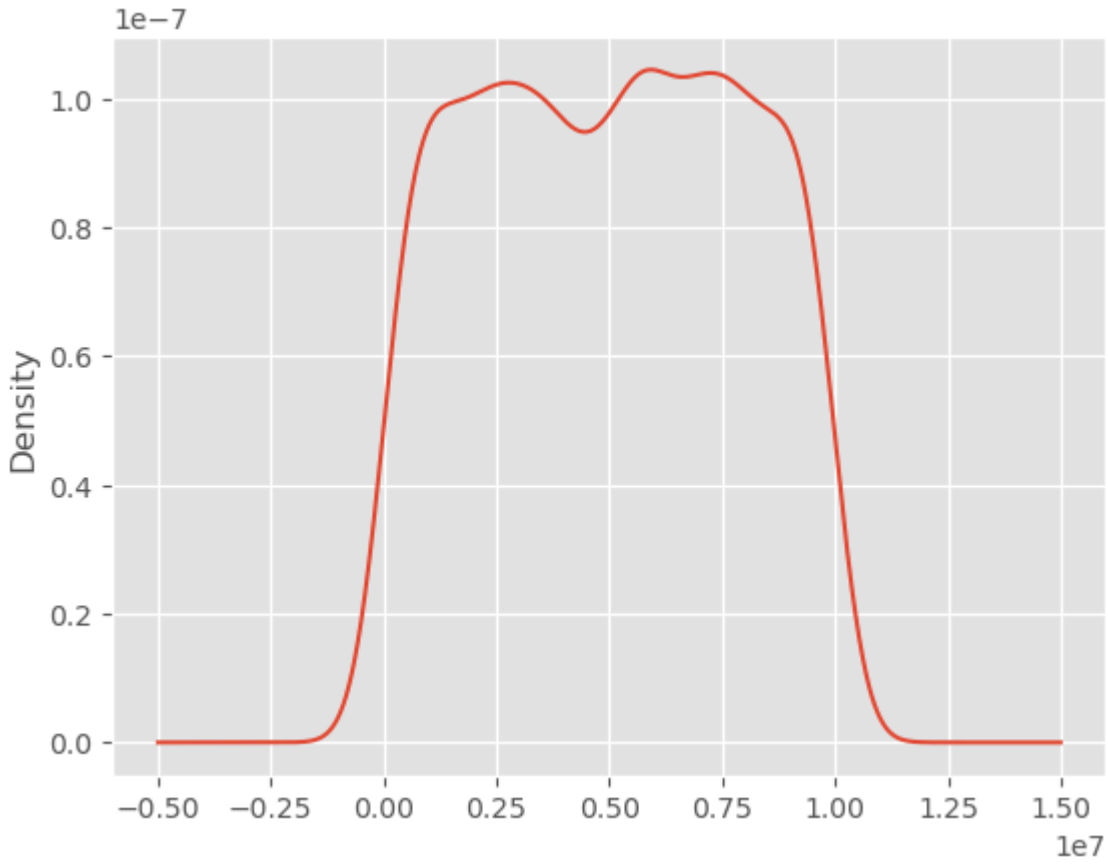
In [79]:
```
dataset['Flight ID'].value_counts().head().plot(kind='hist')
```

Out[79]:
```
<Axes: ylabel='Frequency'>
```



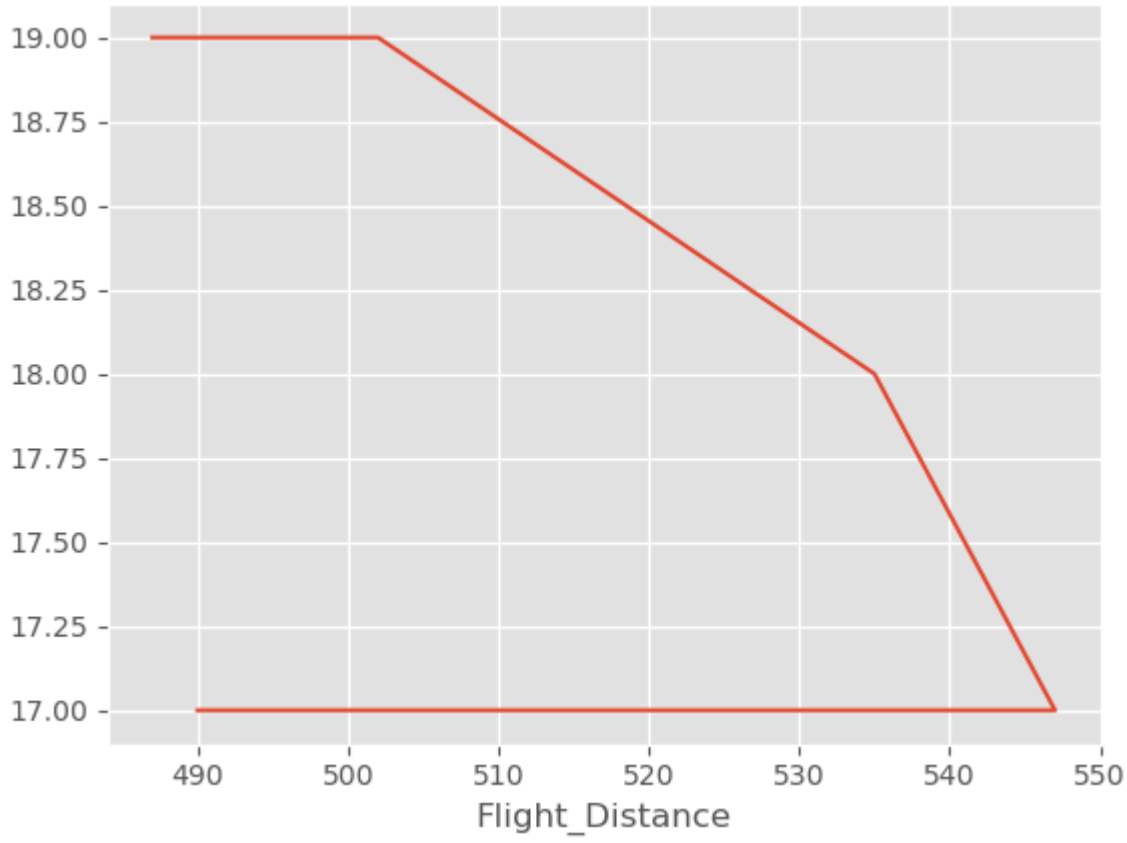
In [80]:
```
#GENERATING A KDE PLOT TO VISUALIZE THE DISTRIBUTION OF THE DATA IN THE SPECIFIED C
dataset['Flight ID'].plot(kind='kde')
```

Out[80]: <Axes: ylabel='Density'>



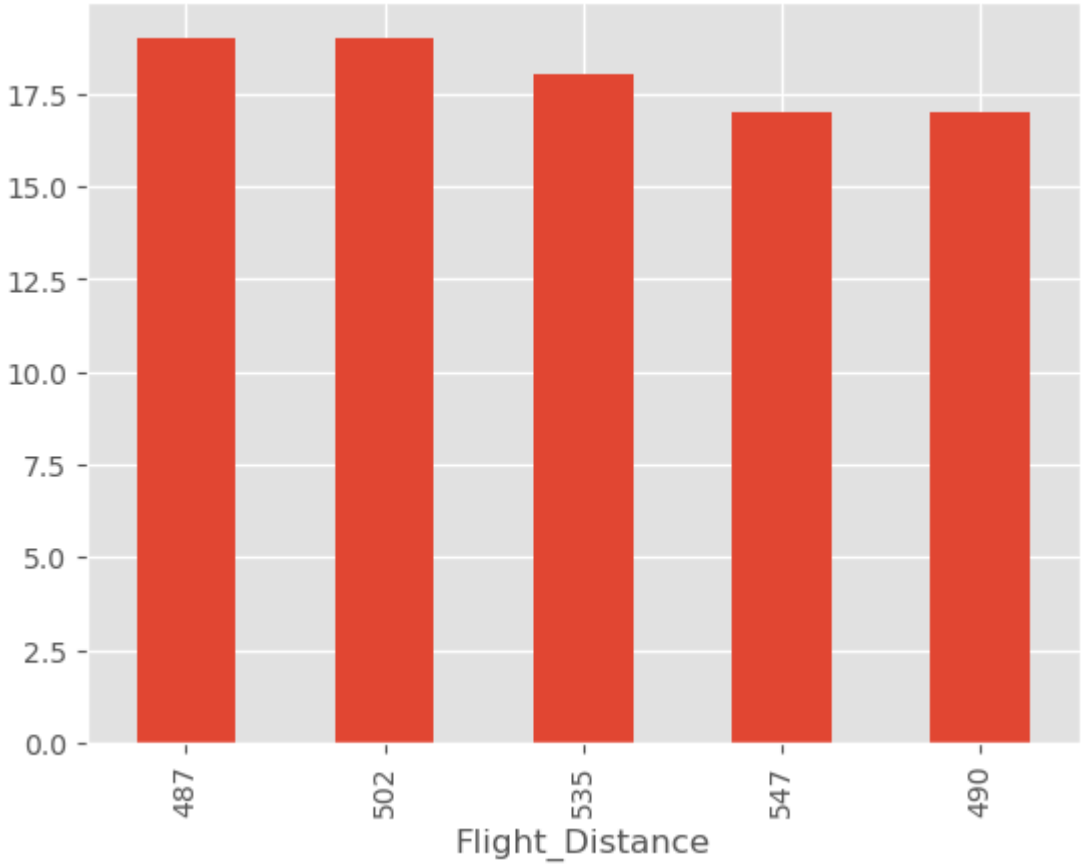
In [81]:
```
dataset['Flight_Distance'].value_counts().head().plot()
```

Out[81]: <Axes: xlabel='Flight_Distance'>



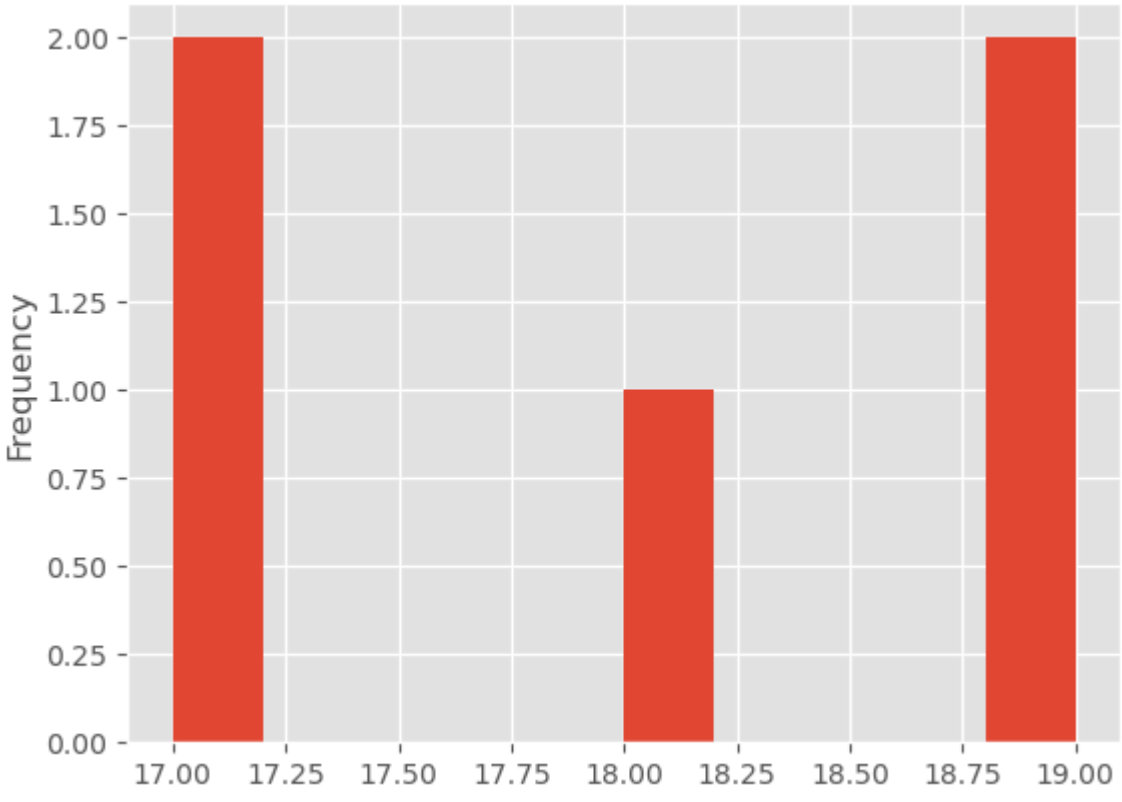
In [84]:
```
dataset['Flight_Distance'].value_counts().head().plot(kind='bar')
```

Out[84]: <Axes: xlabel='Flight_Distance'>



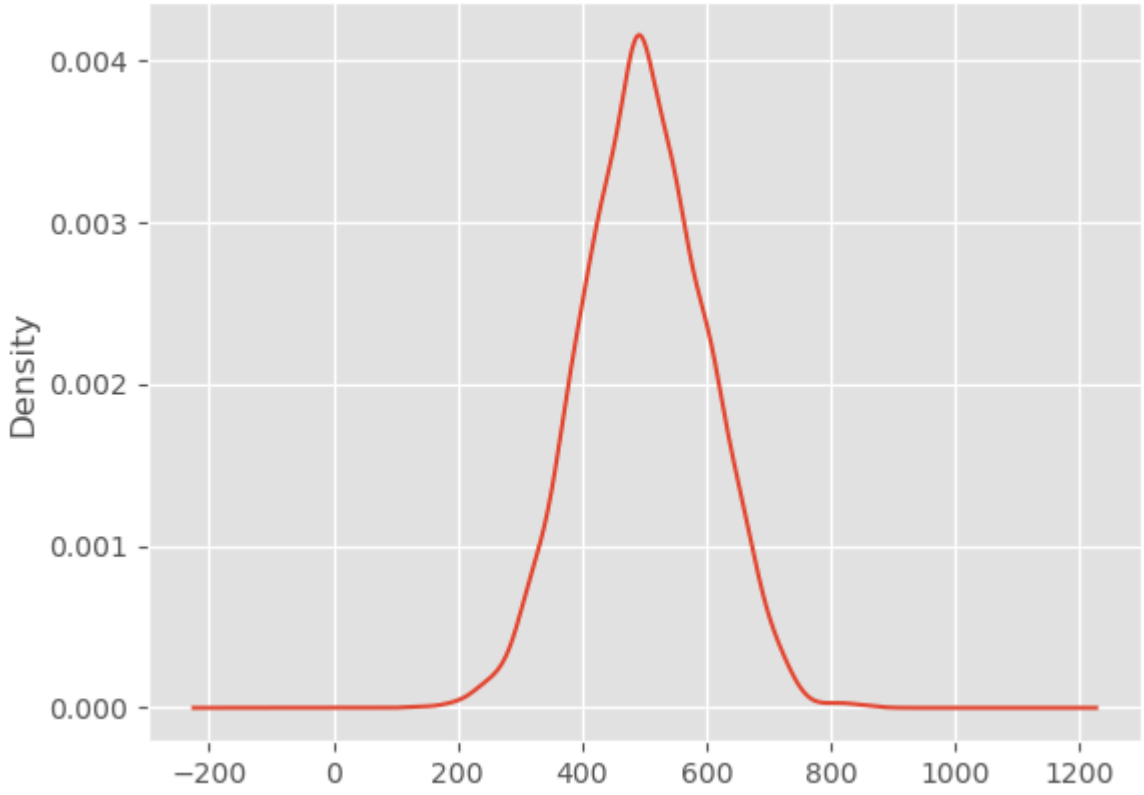
In [85]:
```
dataset['Flight_Distance'].value_counts().head().plot(kind='hist')
```

Out[85]: <Axes: ylabel='Frequency'>



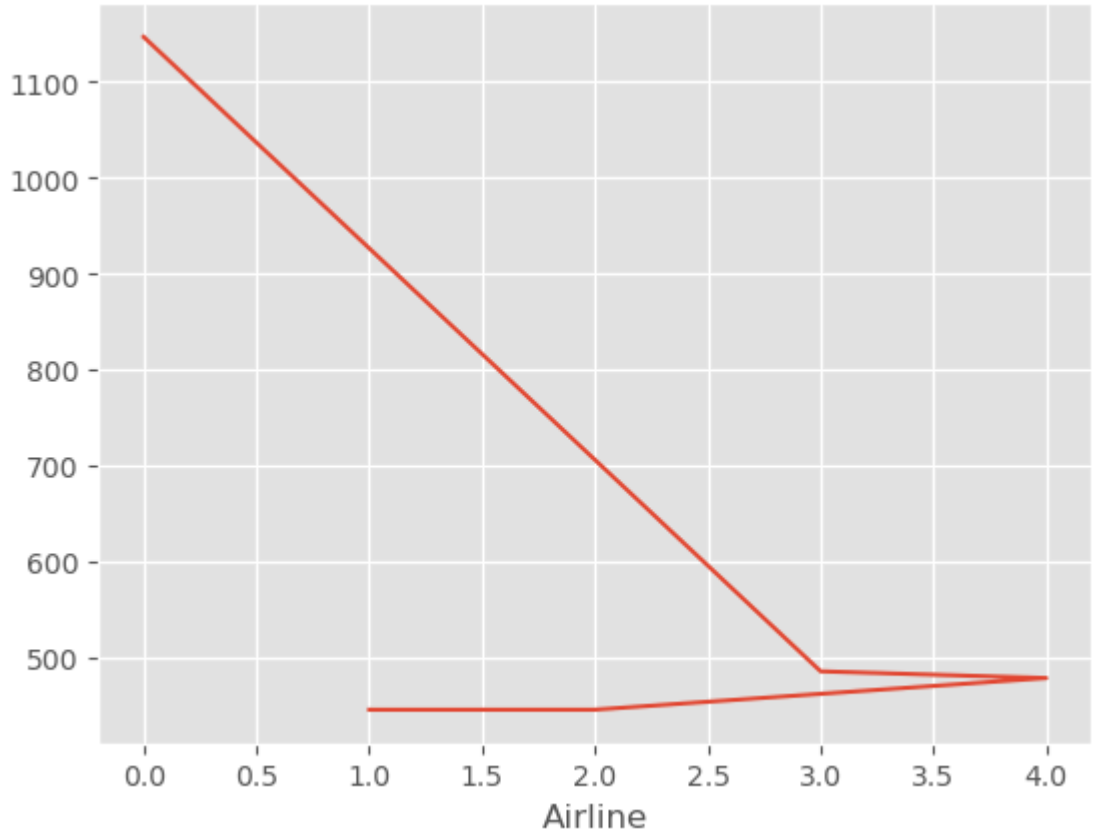
In [86]:
```
dataset['Flight_Distance'].plot(kind='kde')
```

Out[86]: <Axes: ylabel='Density'>



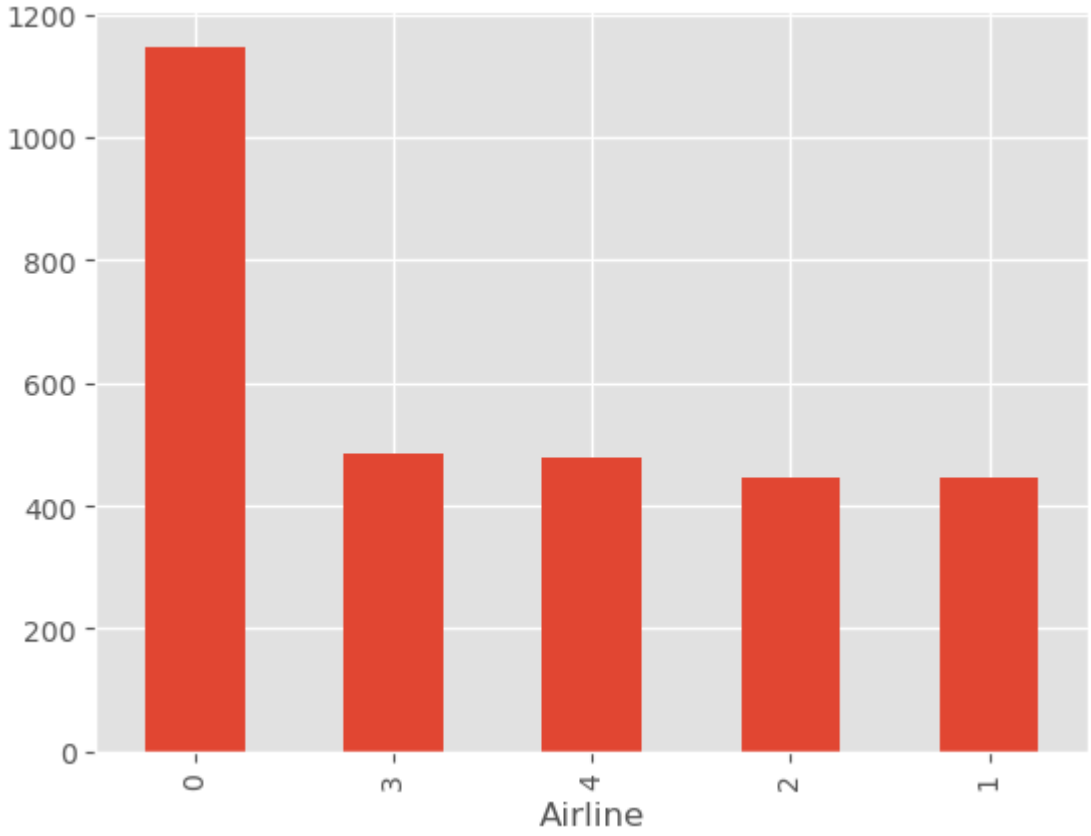
In [87]:
```
dataset['Airline'].value_counts().head().plot()
```

Out[87]:
```
<Axes: xlabel='Airline'>
```



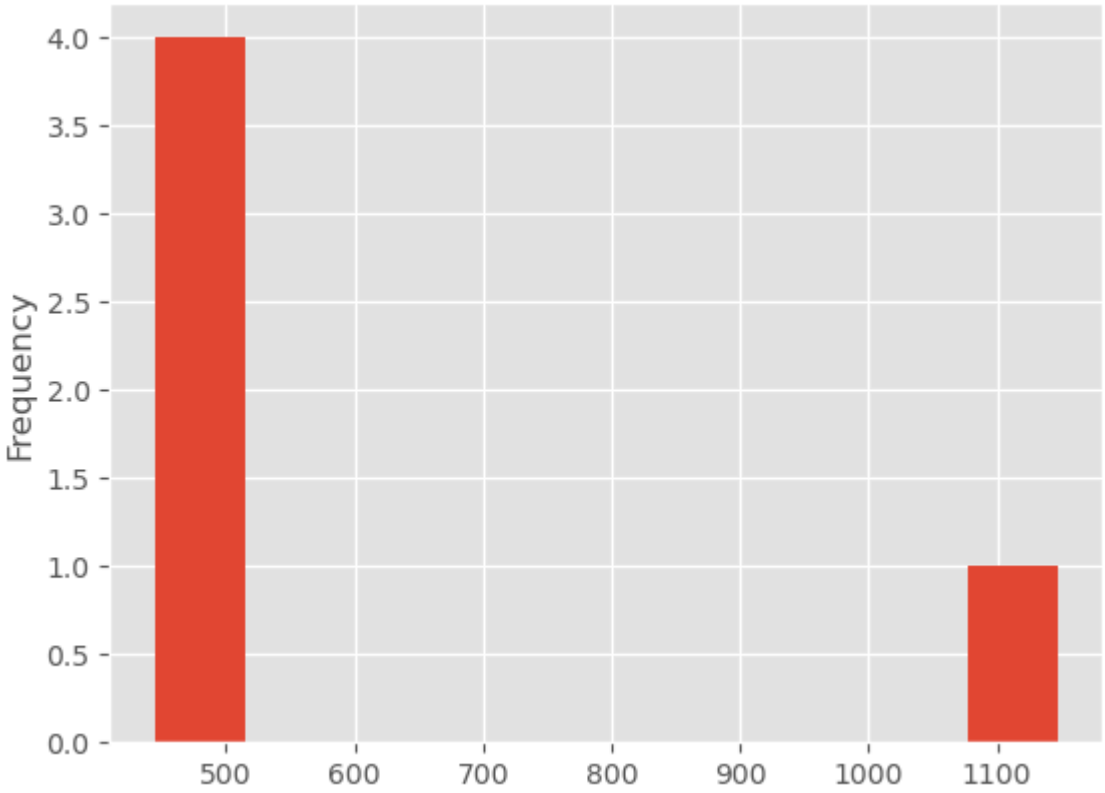
In [88]:
```
dataset['Airline'].value_counts().head().plot(kind='bar')
```

Out[88]:
```
<Axes: xlabel='Airline'>
```



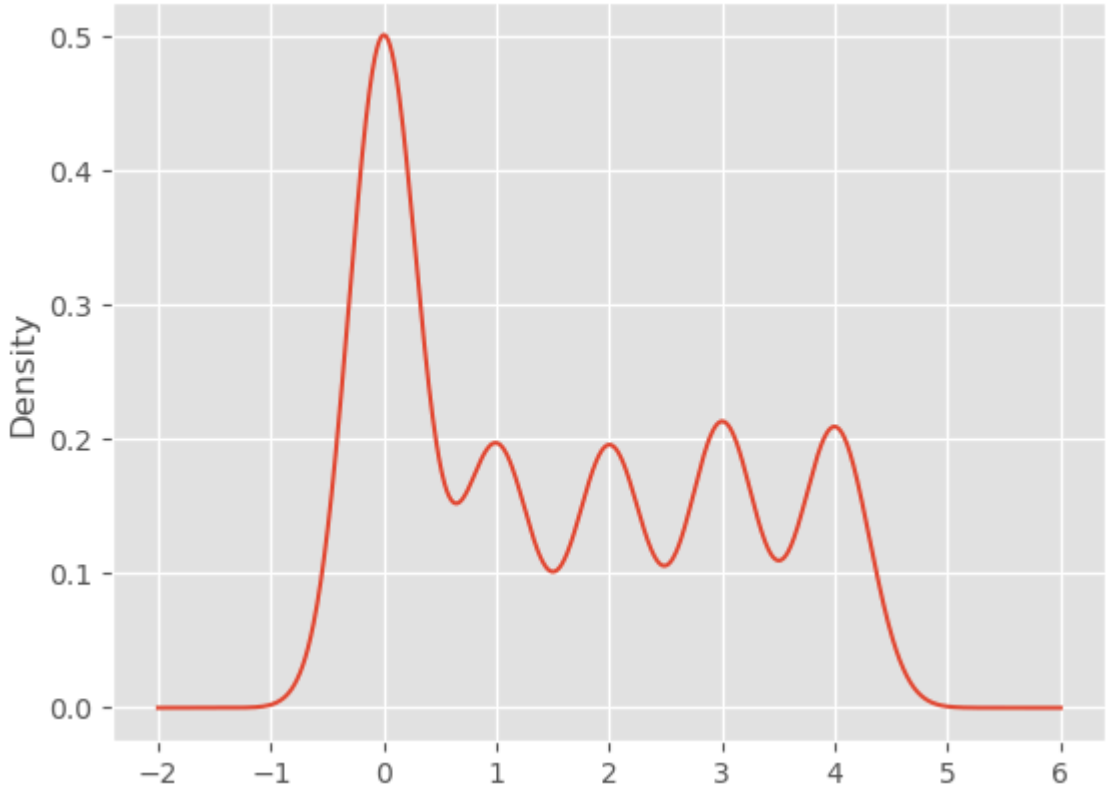
In [89]:
```
dataset['Airline'].value_counts().head().plot(kind='hist')
```

Out[89]:
```
<Axes: ylabel='Frequency'>
```



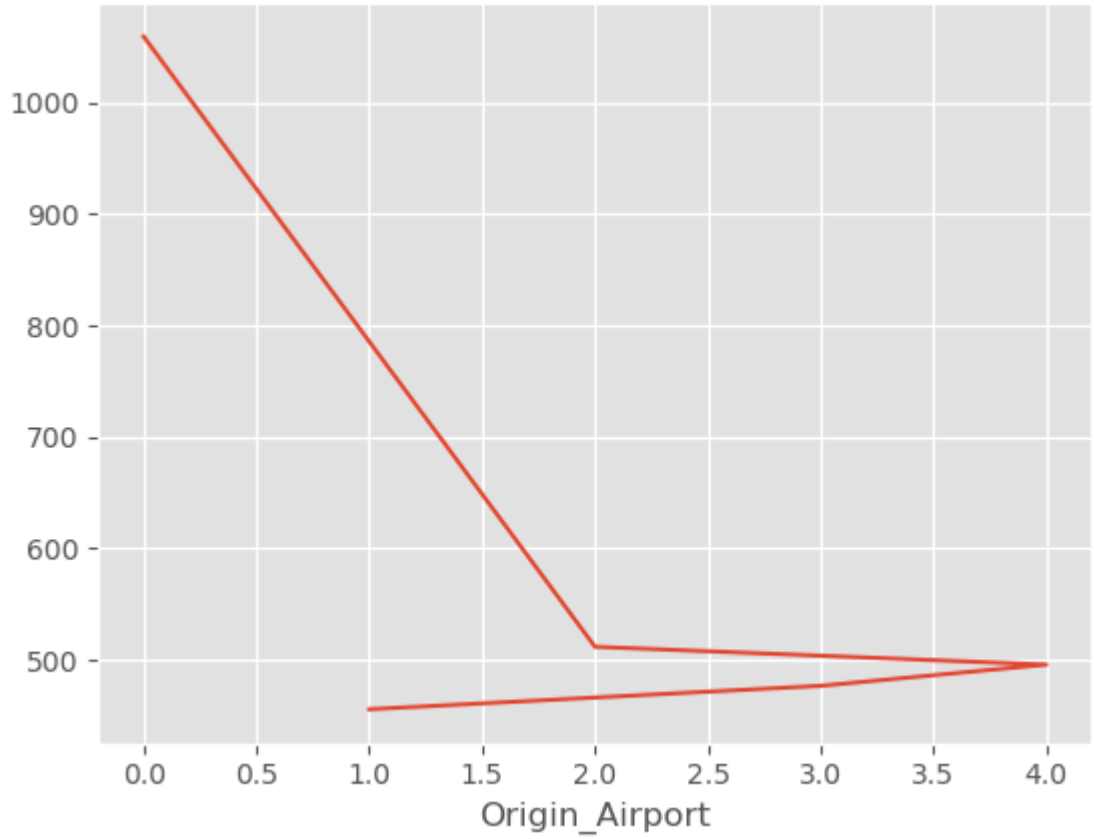
In [90]:
```
dataset['Airline'].plot(kind='kde')
```

Out[90]:
```
<Axes: ylabel='Density'>
```



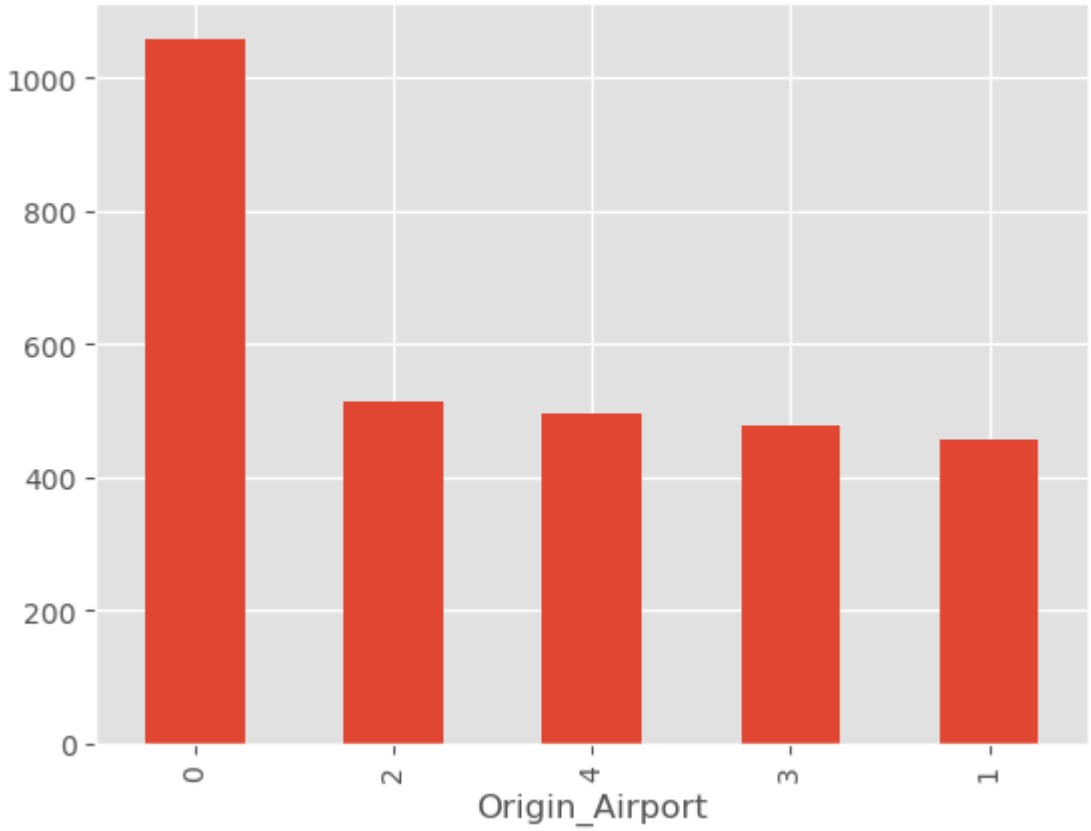
In [91]:
```
dataset['Origin_Airport'].value_counts().head().plot()
```

Out[91]: <Axes: xlabel='Origin_Airport'>



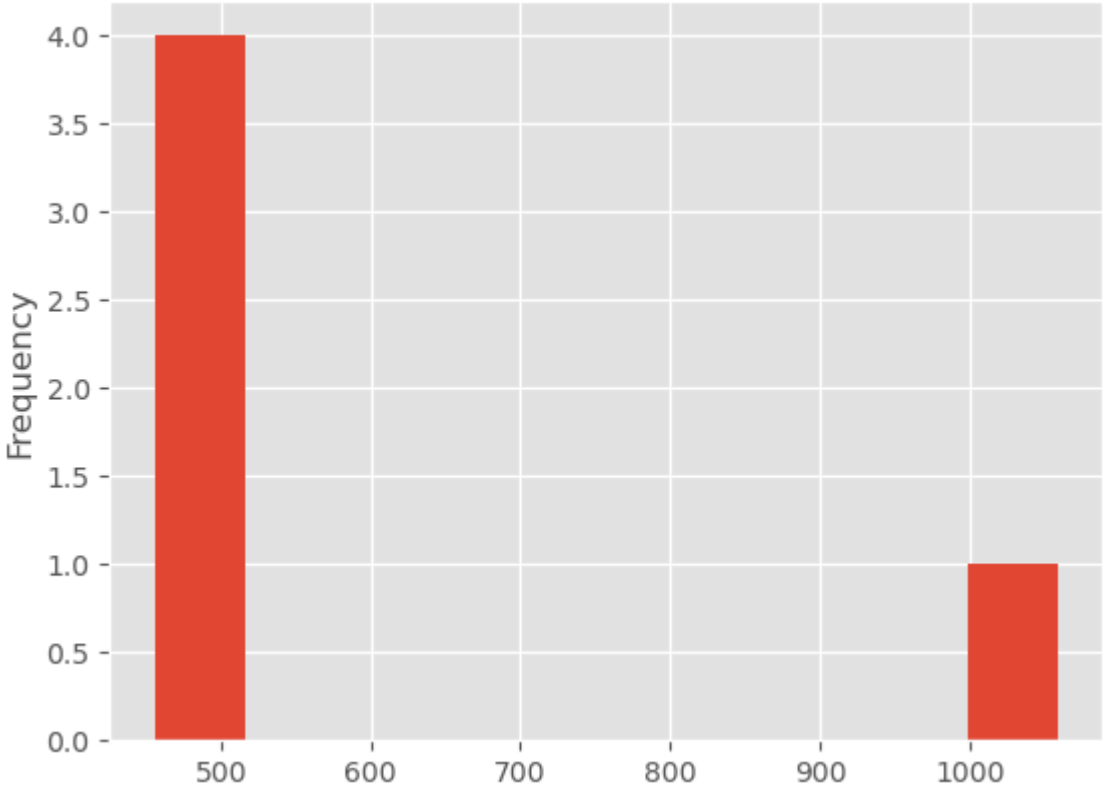
In [92]:
```
dataset['Origin_Airport'].value_counts().head().plot(kind='bar')
```

Out[92]: <Axes: xlabel='Origin_Airport'>



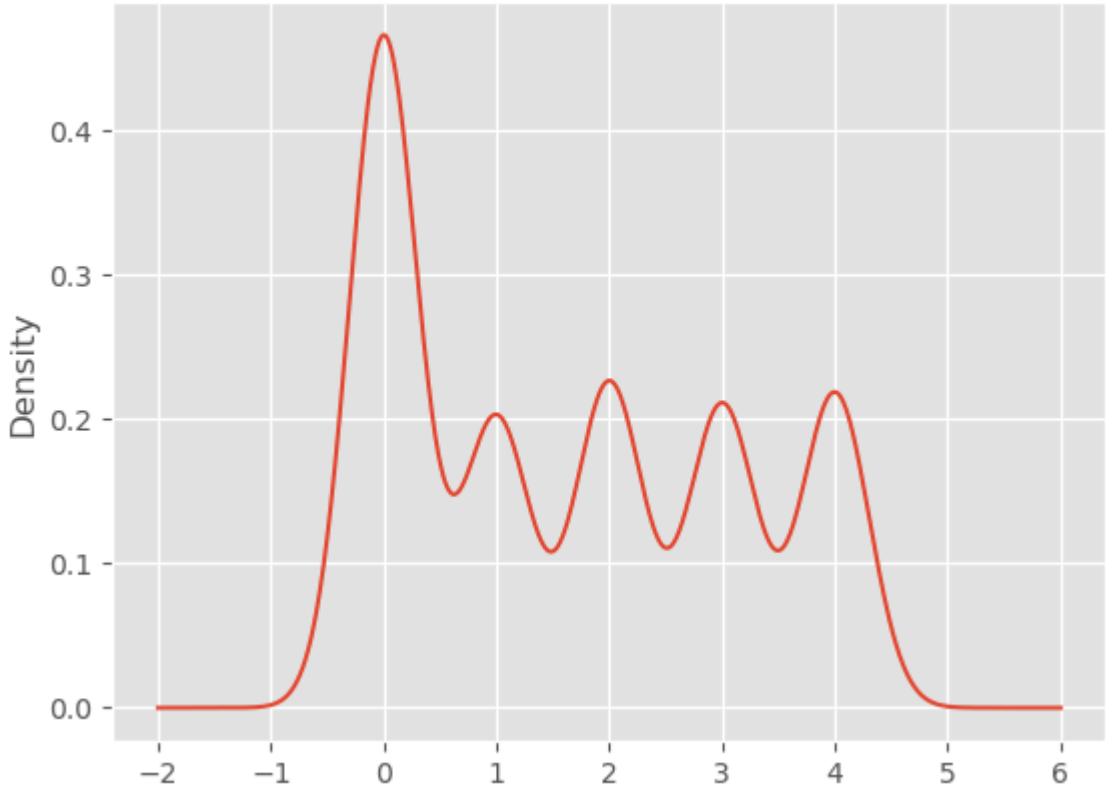
In [93]:
```
dataset['Origin_Airport'].value_counts().head().plot(kind='hist')
```

Out[93]:
```
<Axes: ylabel='Frequency'>
```



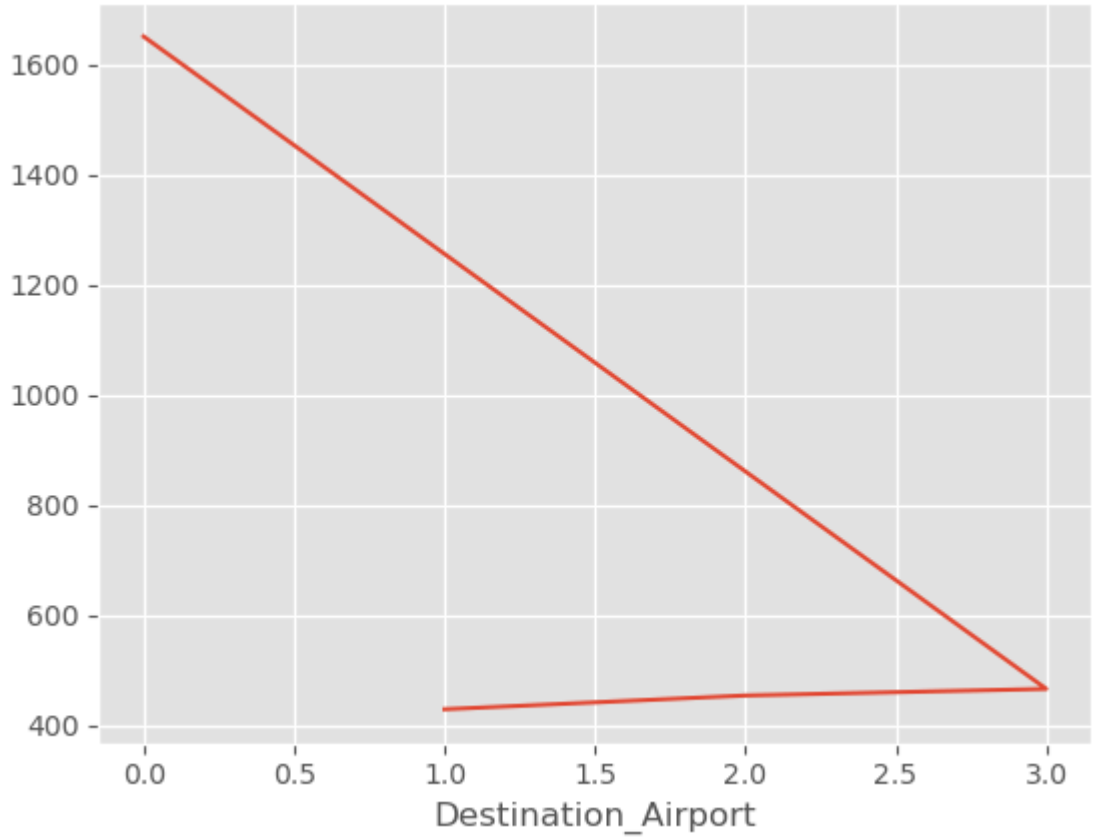
In [94]:
```
dataset['Origin_Airport'].plot(kind='kde')
```

Out[94]:
```
<Axes: ylabel='Density'>
```



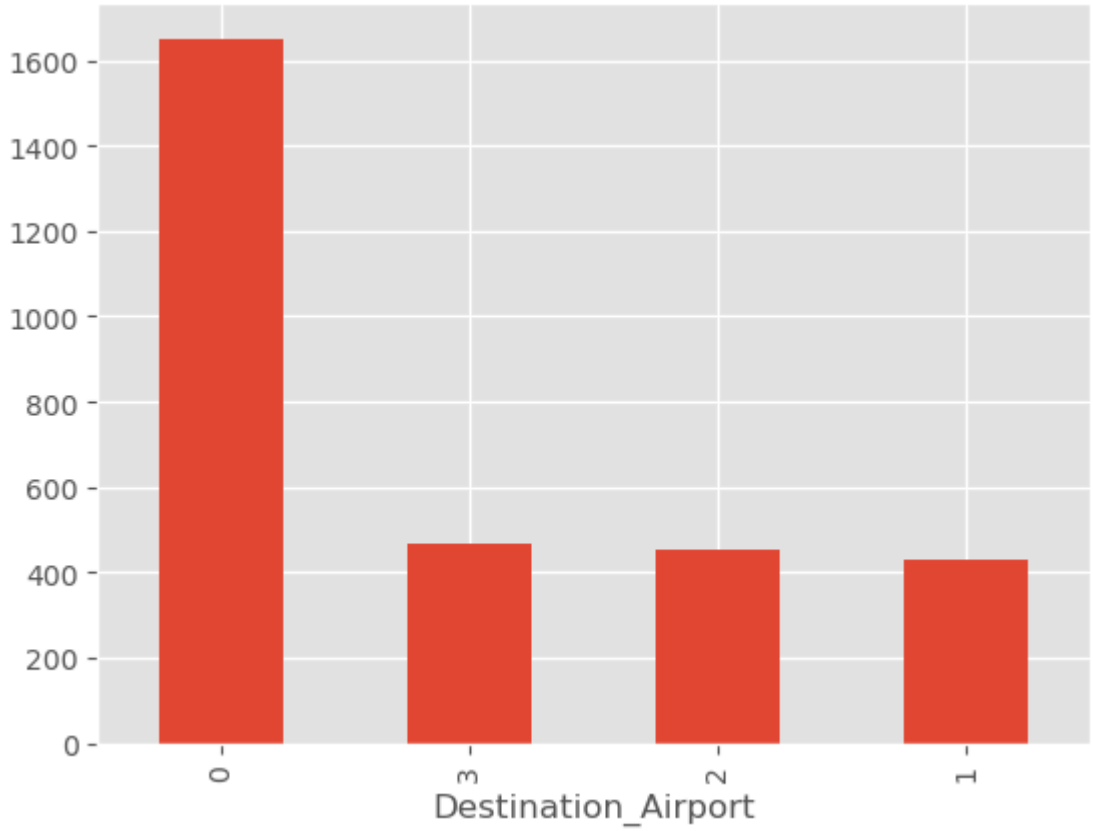
In [95]:
```
dataset['Destination_Airport'].value_counts().head().plot()
```

Out[95]:
```
<Axes: xlabel='Destination_Airport'>
```



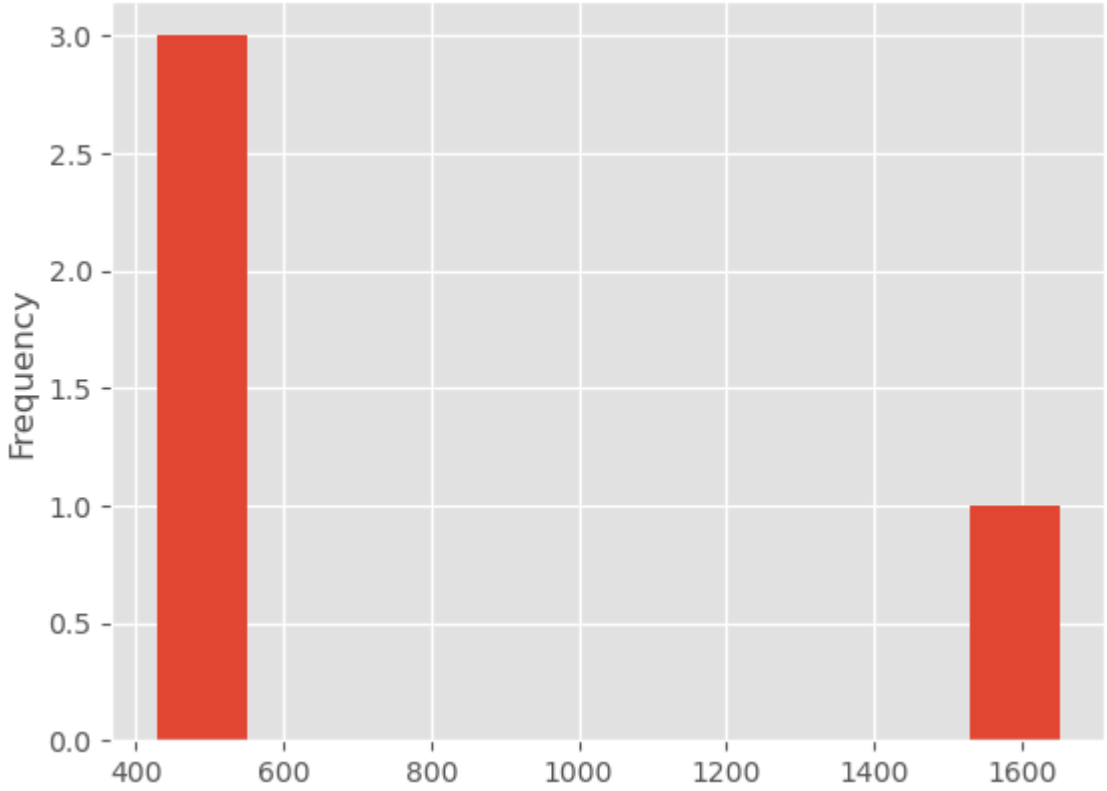
In [96]:
```
dataset['Destination_Airport'].value_counts().head().plot(kind='bar')
```

Out[96]:
```
<Axes: xlabel='Destination_Airport'>
```



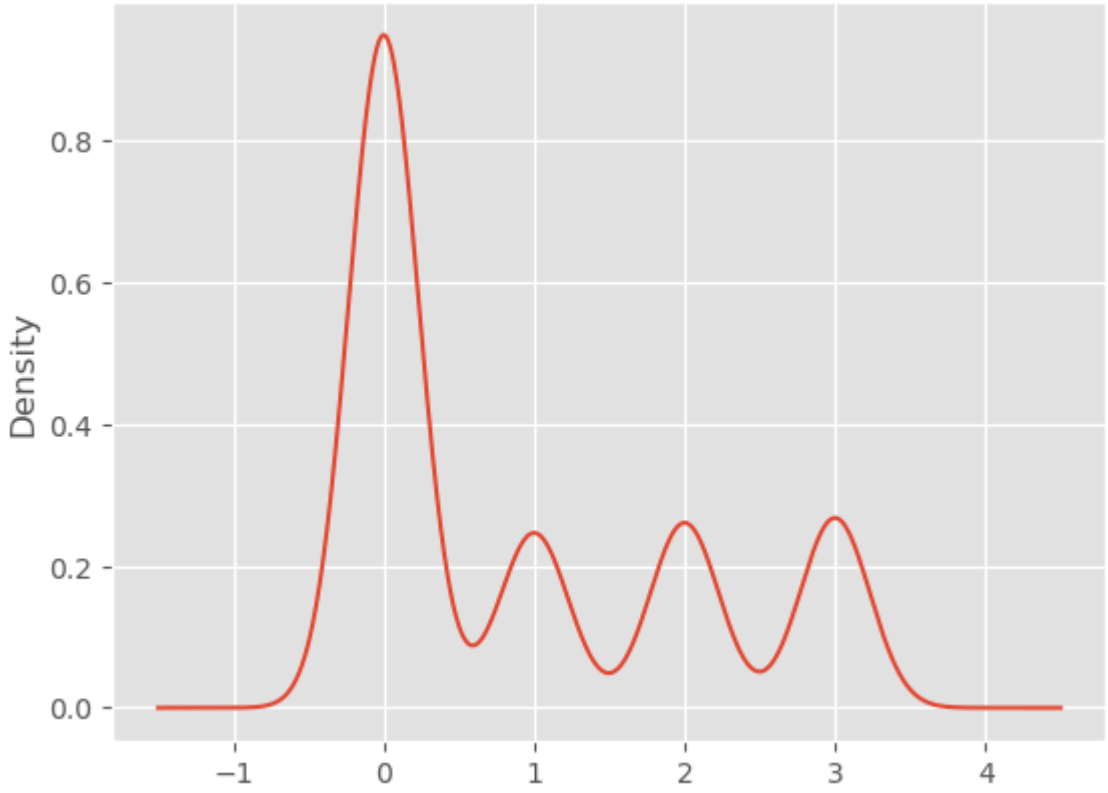
In [97]:
```
dataset['Destination_Airport'].value_counts().head().plot(kind='hist')
```

Out[97]:
```
<Axes: ylabel='Frequency'>
```



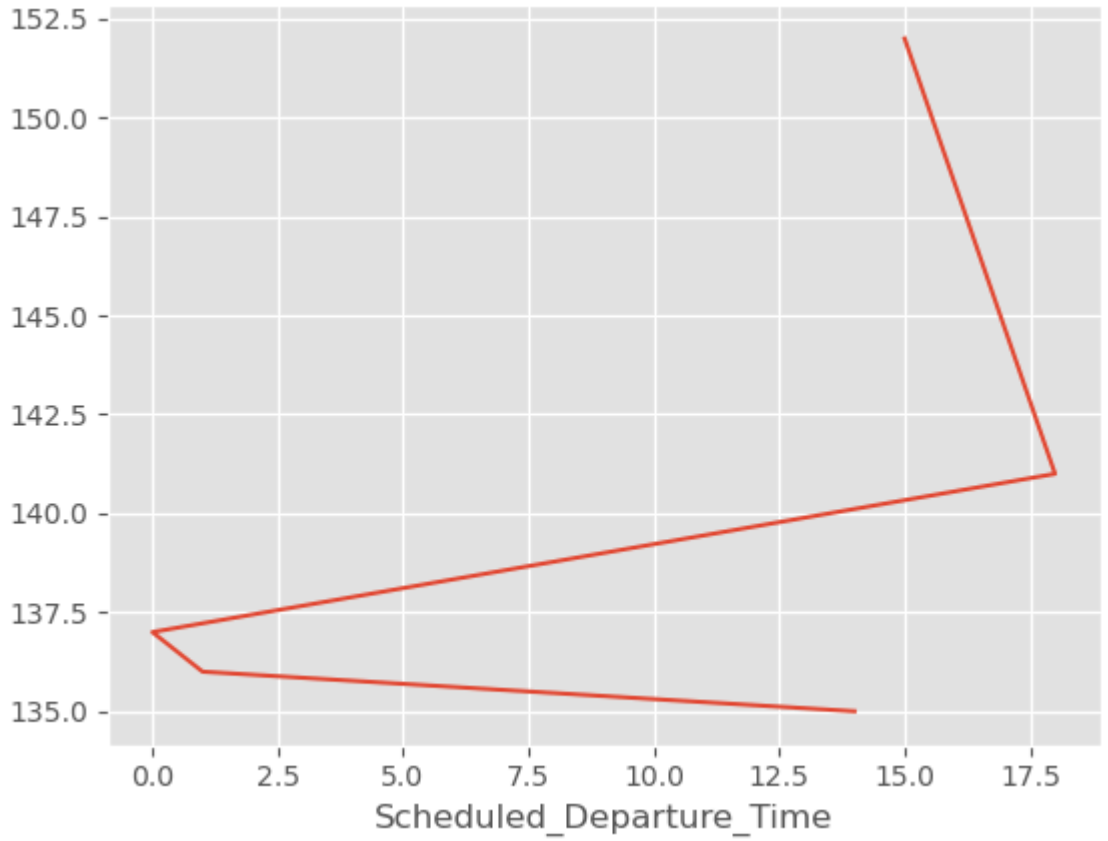
In [98]:
```
dataset['Destination_Airport'].plot(kind='kde')
```

Out[98]:
```
<Axes: ylabel='Density'>
```



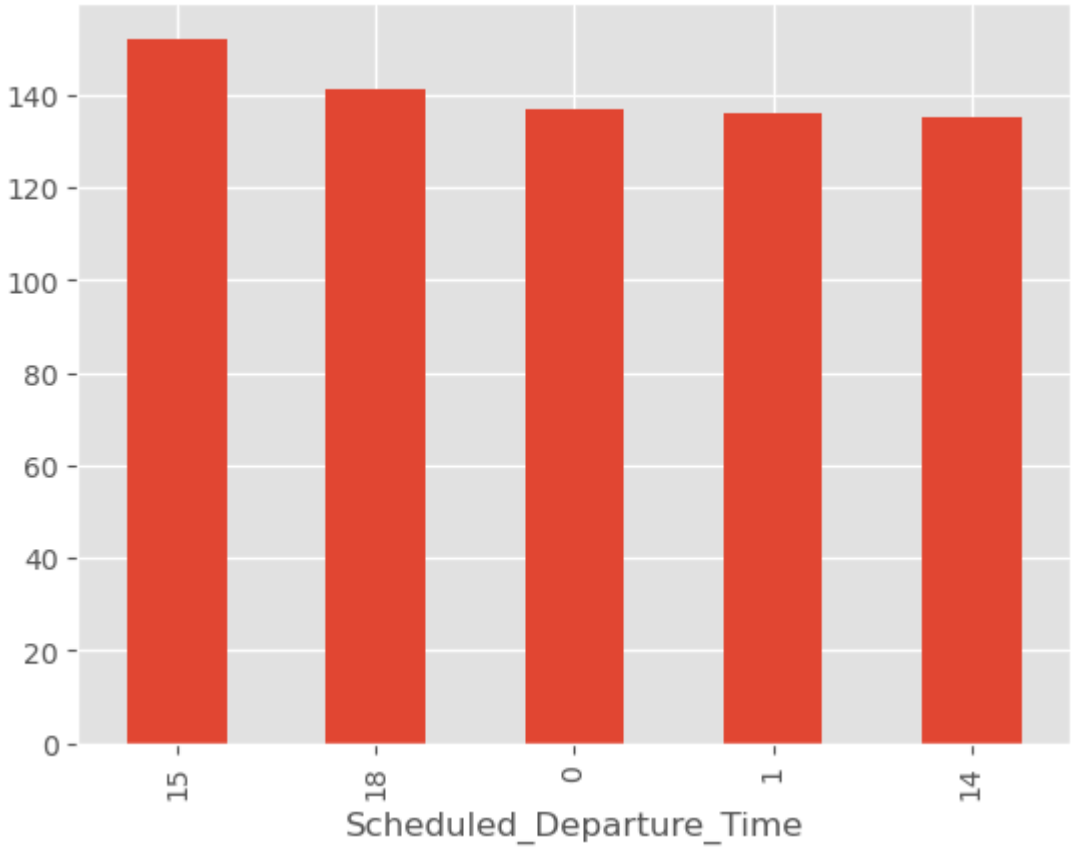
In [99]:
```
dataset['Scheduled_Departure_Time'].value_counts().head().plot()
```

Out[99]:
```
<Axes: xlabel='Scheduled_Departure_Time'>
```



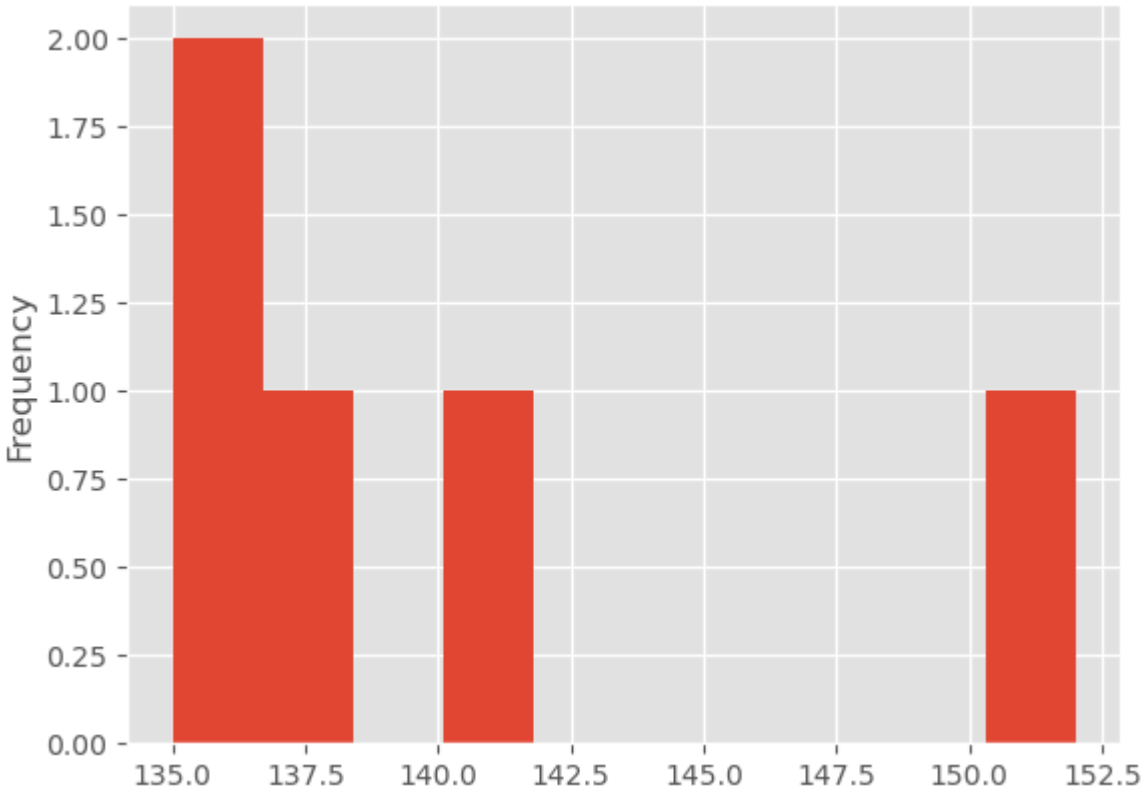
In [100…
```
dataset['Scheduled_Departure_Time'].value_counts().head().plot(kind='bar')
```

Out[100]:
```
<Axes: xlabel='Scheduled_Departure_Time'>
```



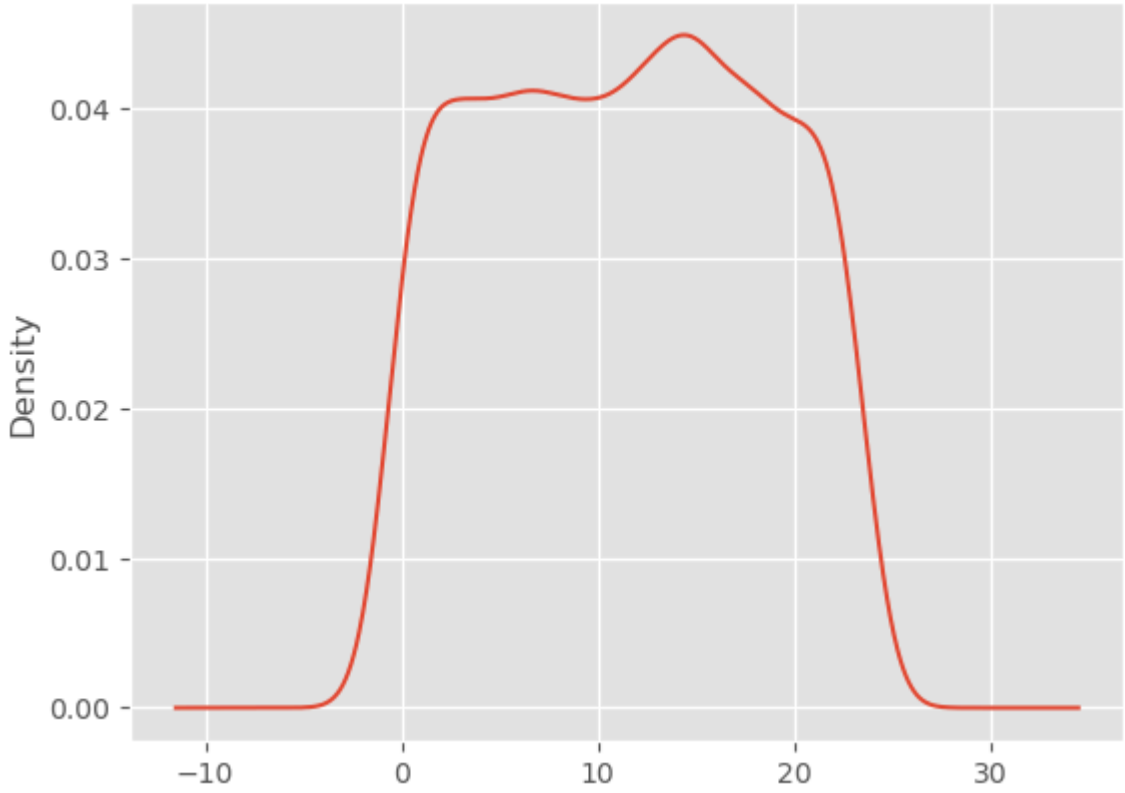
In [101…
```
dataset['Scheduled_Departure_Time'].value_counts().head().plot(kind='hist')
```

Out[101]:
```
<Axes: ylabel='Frequency'>
```



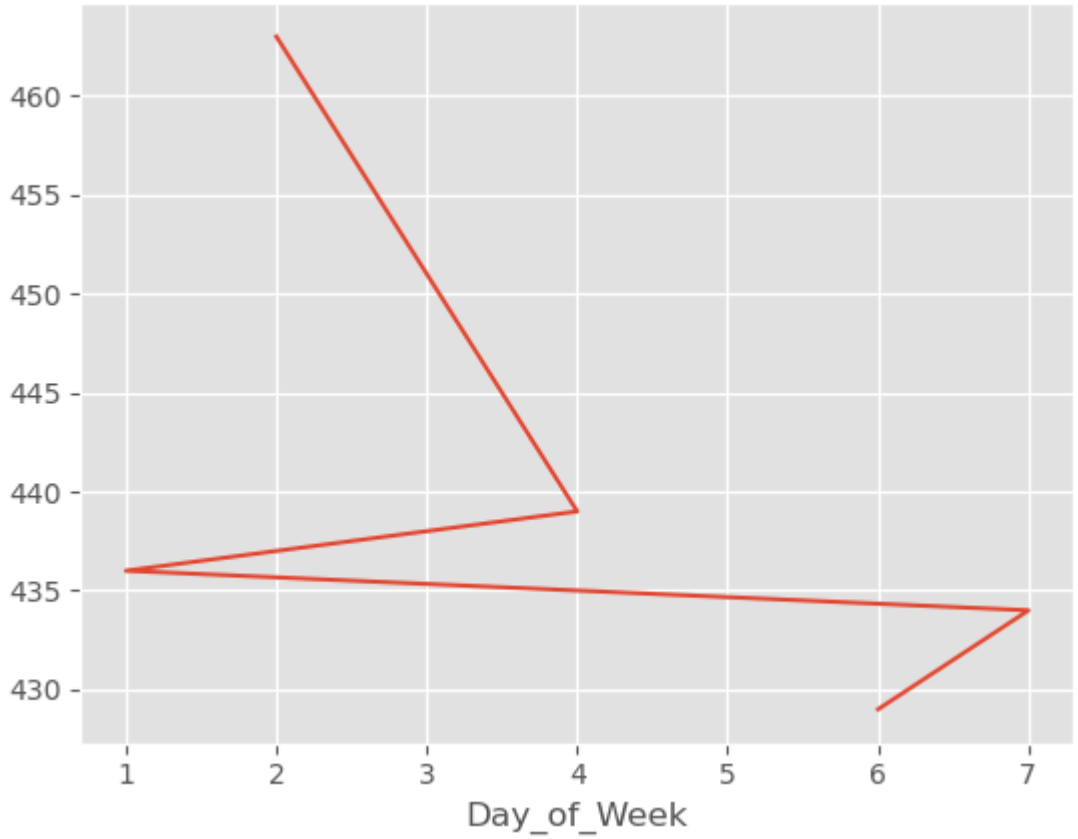
In [102…
```
dataset['Scheduled_Departure_Time'].plot(kind='kde')
```

Out[102]:
```
<Axes: ylabel='Density'>
```



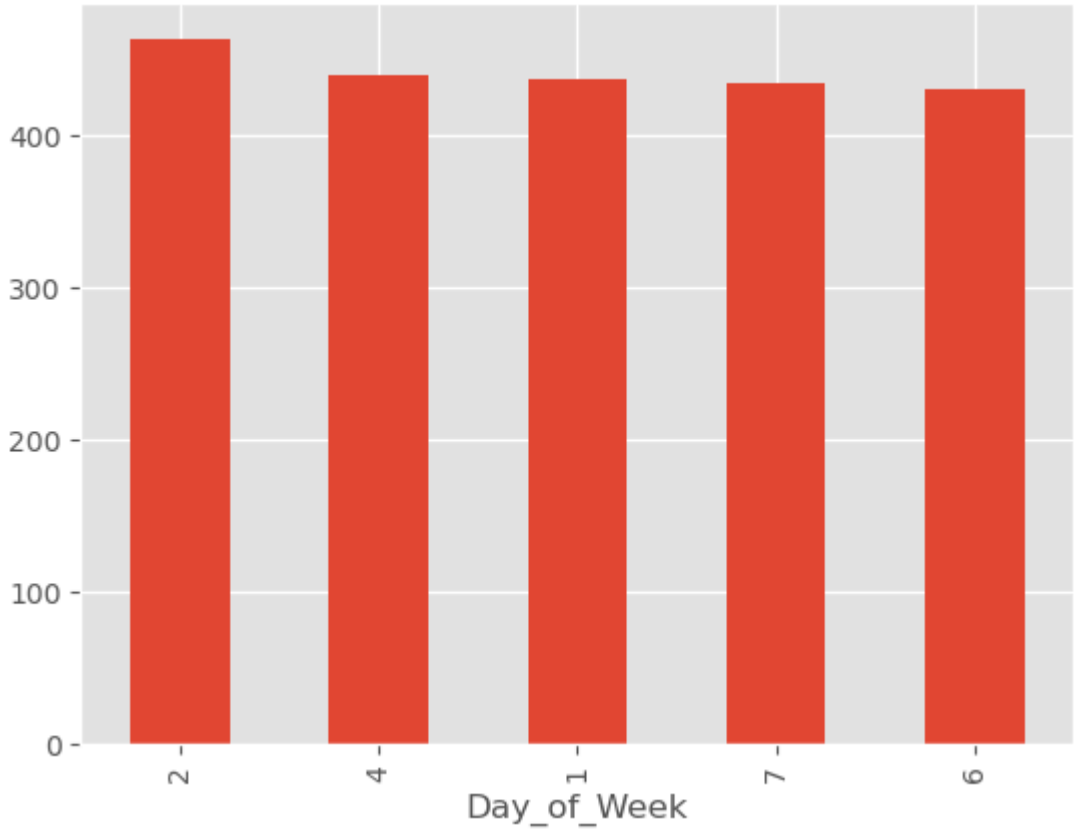
In [103…
```
dataset['Day_of_Week'].value_counts().head().plot()
```

Out[103]:
```
<Axes: xlabel='Day_of_Week'>
```



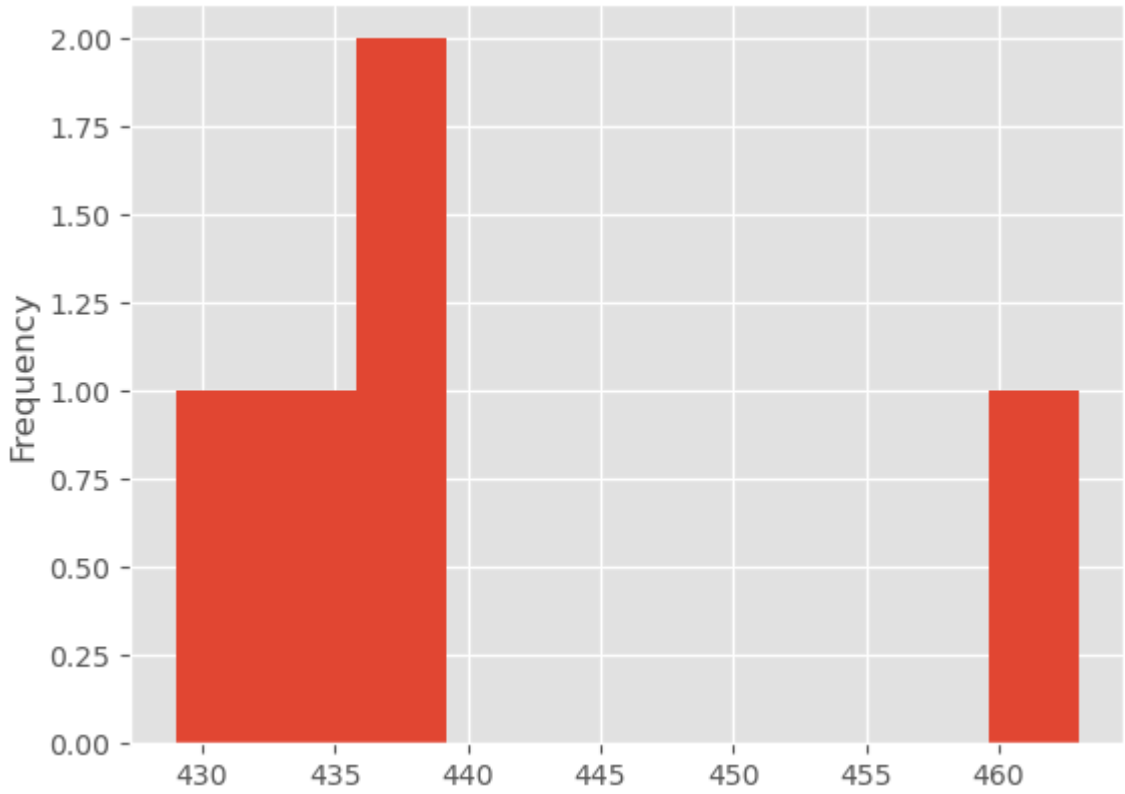
In [104…
```
dataset['Day_of_Week'].value_counts().head().plot(kind='bar')
```

Out[104]:
```
<Axes: xlabel='Day_of_Week'>
```



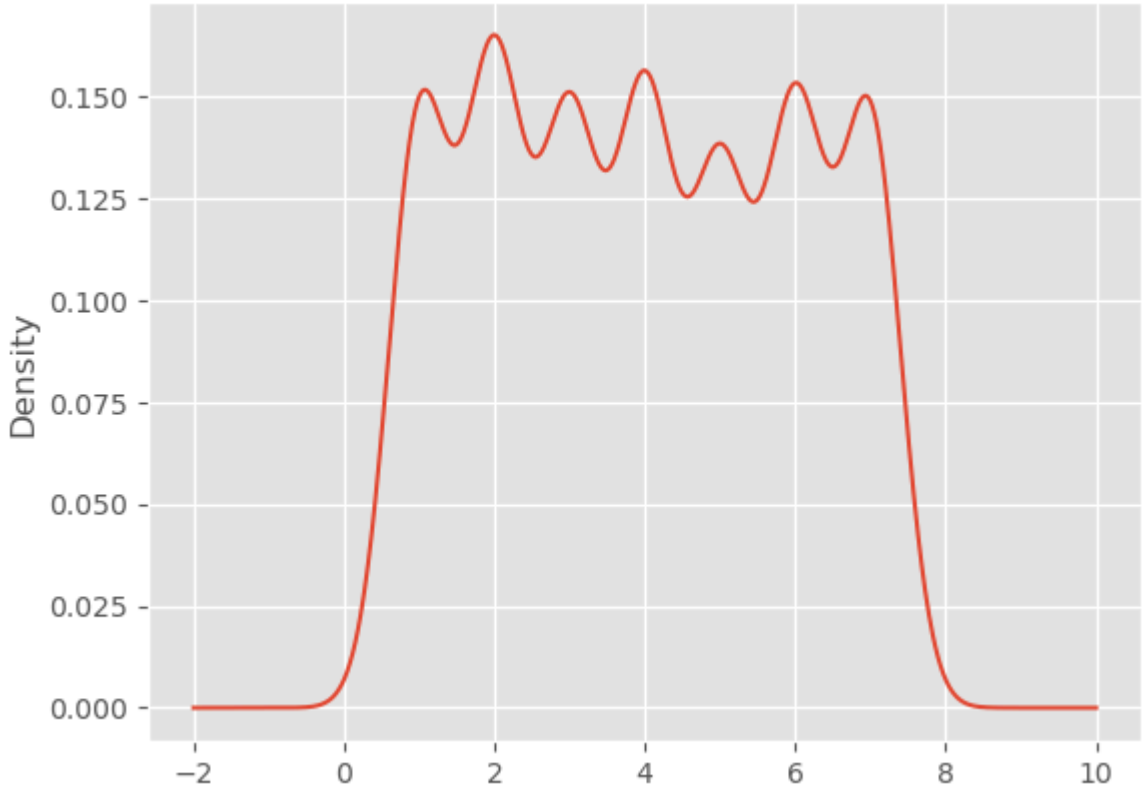
In [105…
```
dataset['Day_of_Week'].value_counts().head().plot(kind='hist')
```

Out[105]:
```
<Axes: ylabel='Frequency'>
```



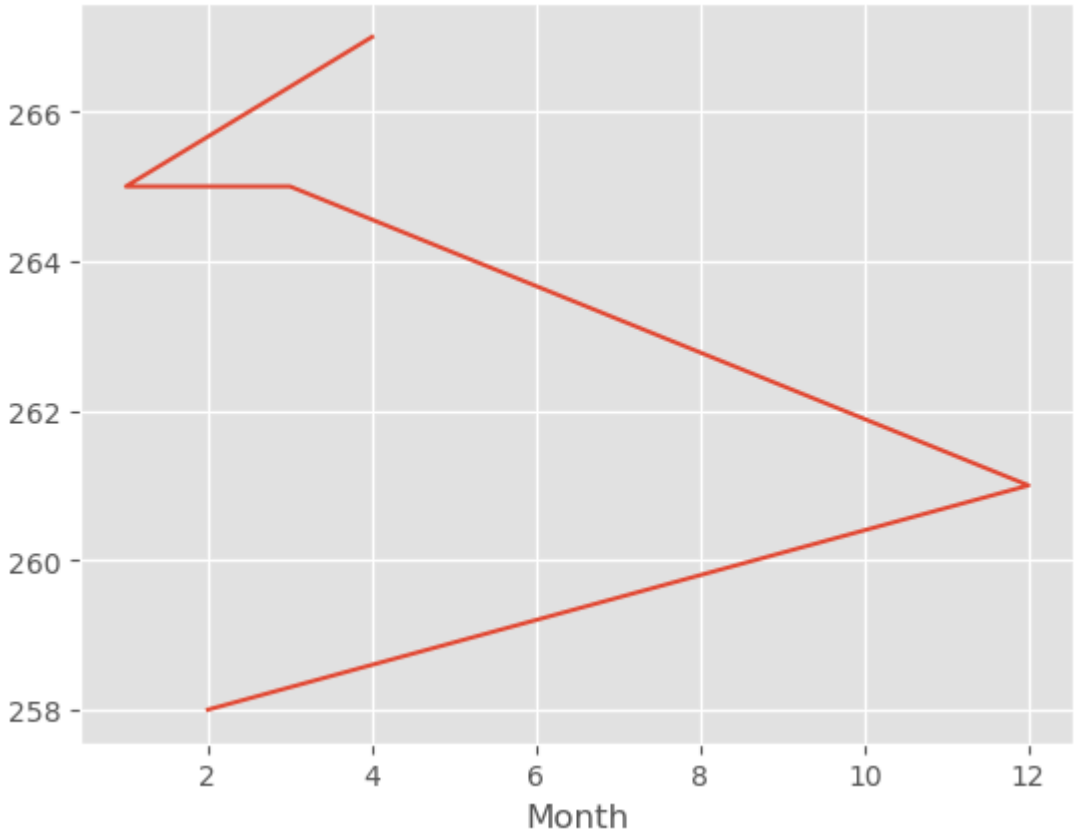
In [106…
```
dataset['Day_of_Week'].plot(kind='kde')
```

Out[106]:
```
<Axes: ylabel='Density'>
```



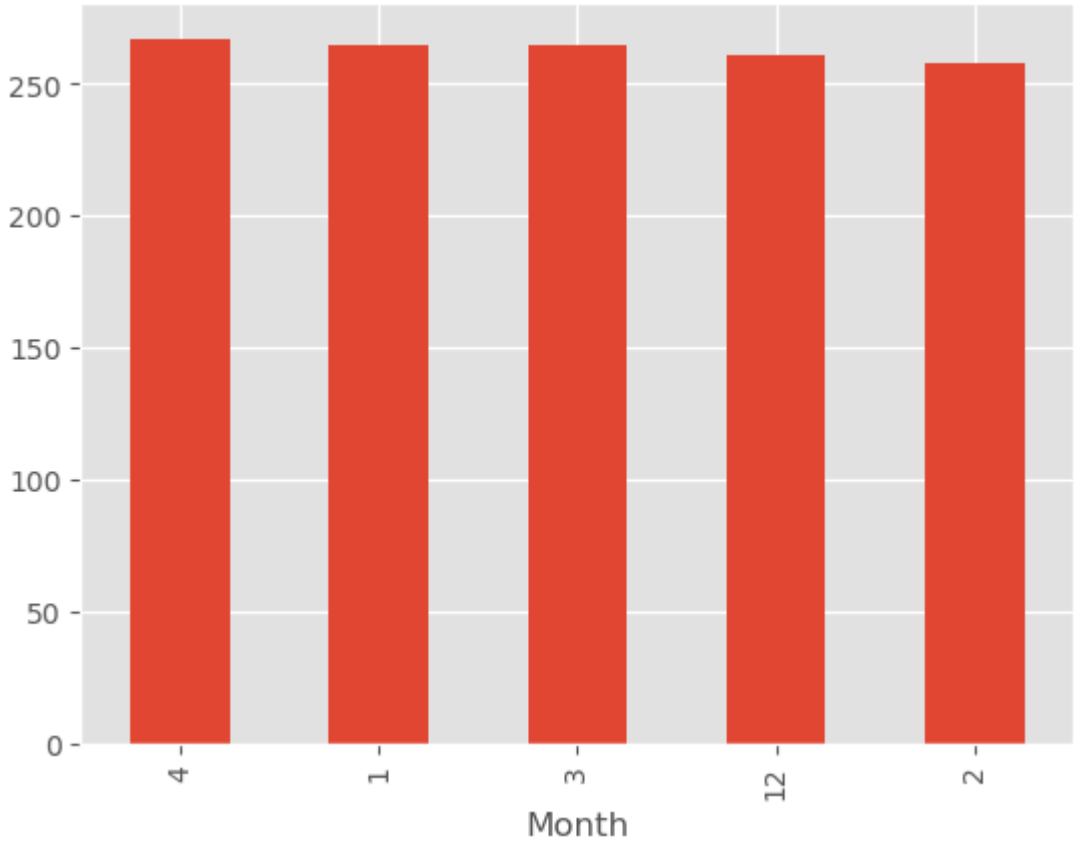
In [107…
```
dataset['Month'].value_counts().head().plot()
```

Out[107]:
```
<Axes: xlabel='Month'>
```



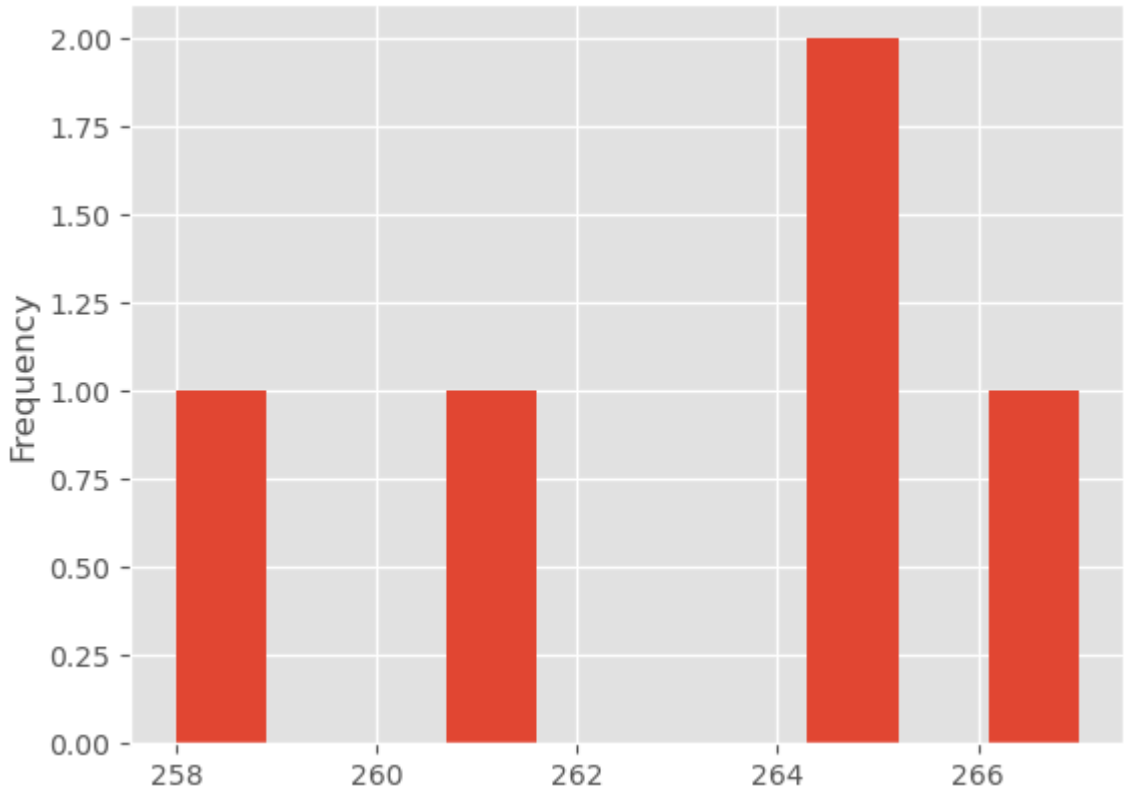
In [108…
```
dataset['Month'].value_counts().head().plot(kind='bar')
```

Out[108]:
```
<Axes: xlabel='Month'>
```



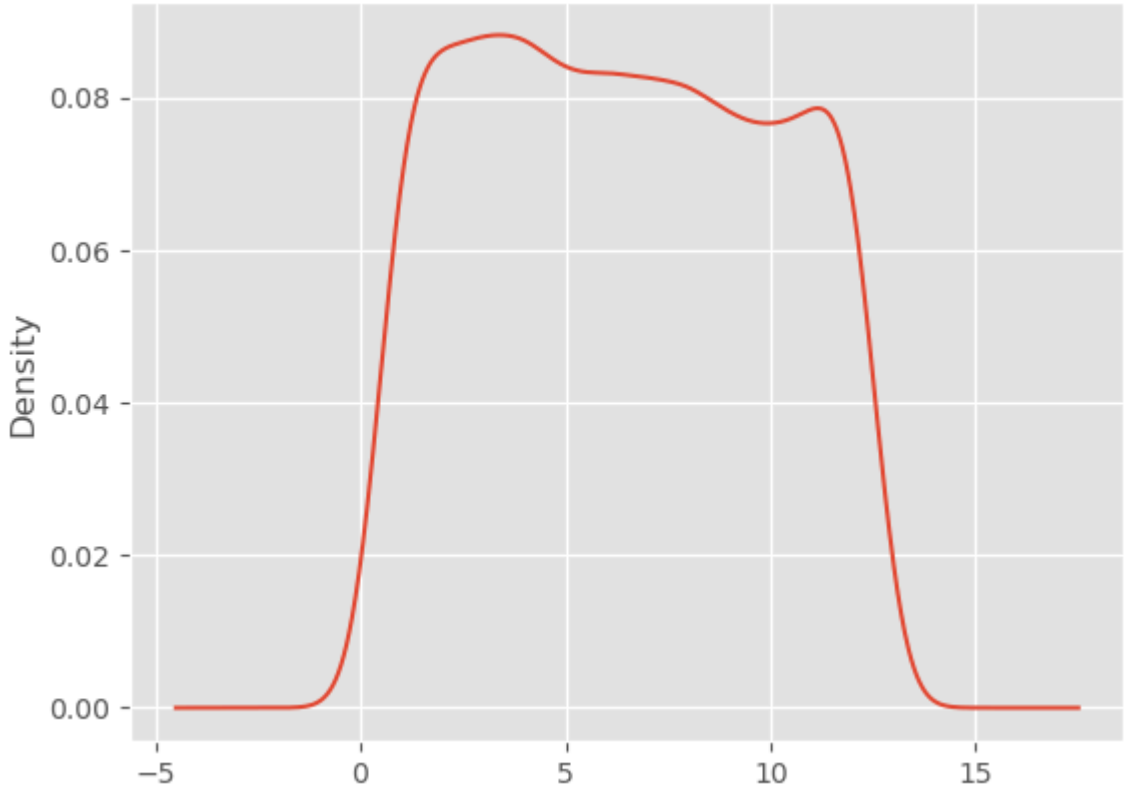
In [109…
```
dataset['Month'].value_counts().head().plot(kind='hist')
```

Out[109]:
```
<Axes: ylabel='Frequency'>
```



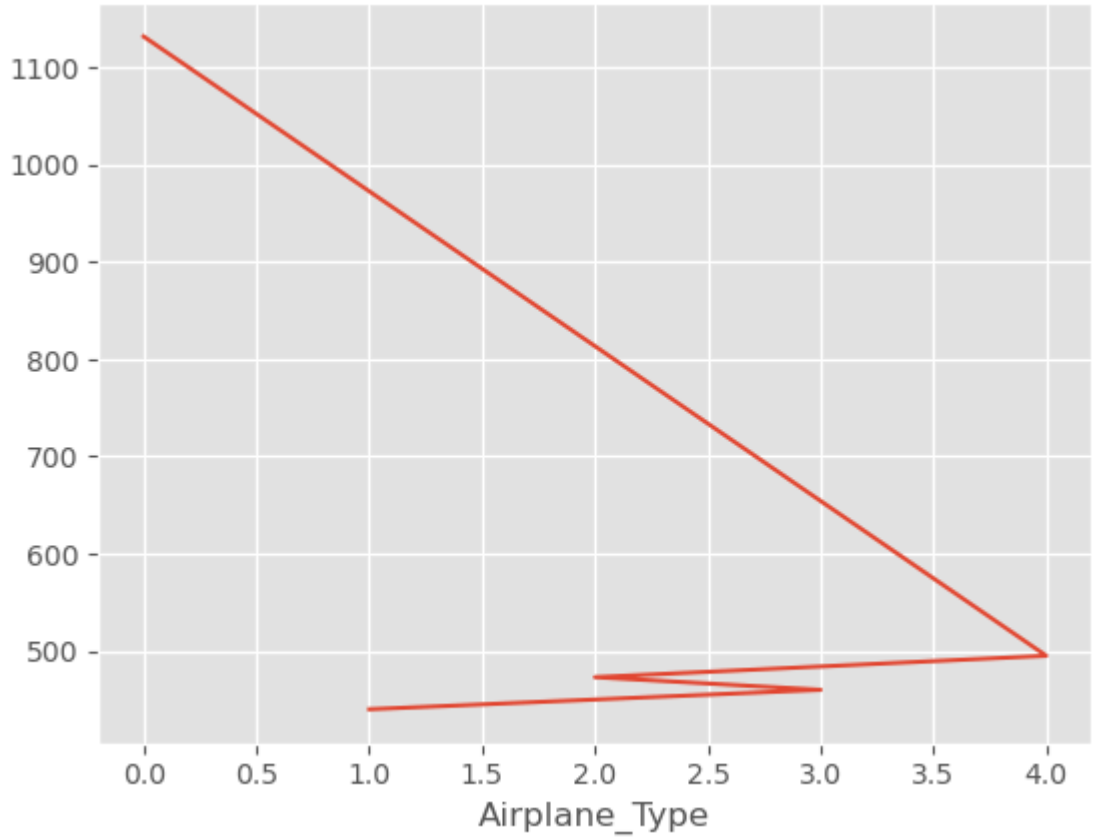
In [110…
```
dataset['Month'].plot(kind='kde')
```

Out[110]:
```
<Axes: ylabel='Density'>
```



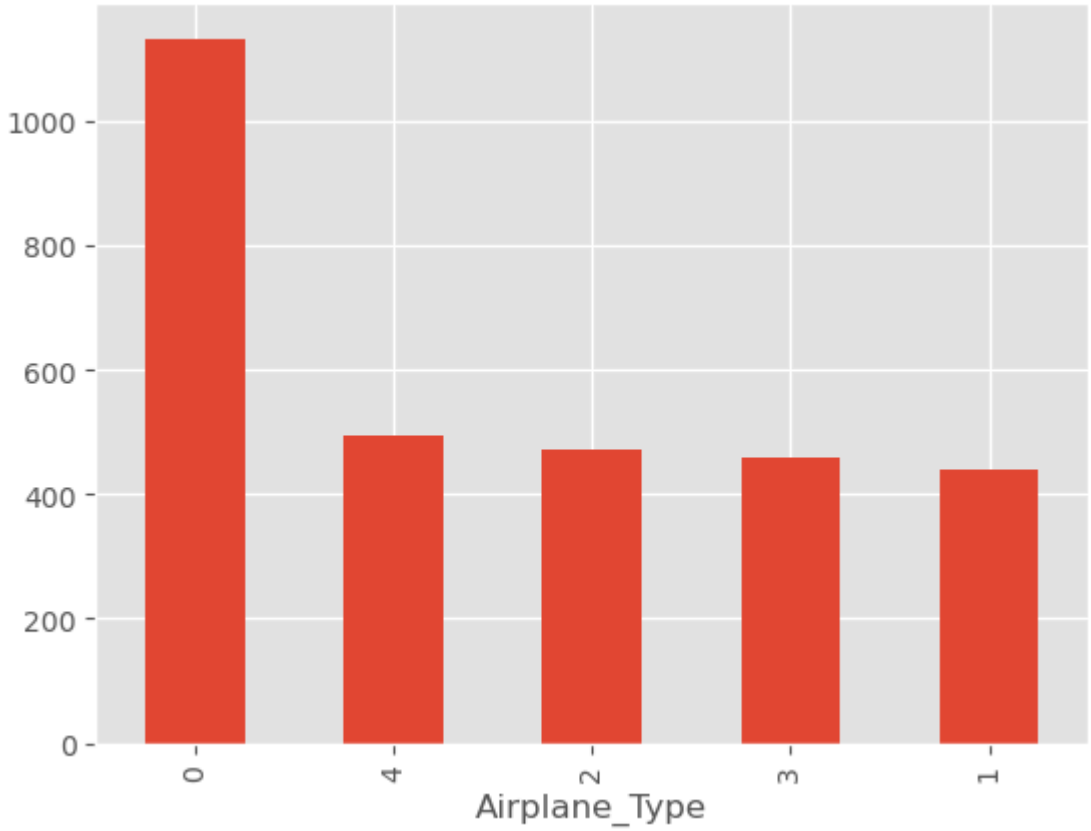
In [111…
```
dataset['Airplane_Type'].value_counts().head().plot()
```

Out[111]:
```
<Axes: xlabel='Airplane_Type'>
```



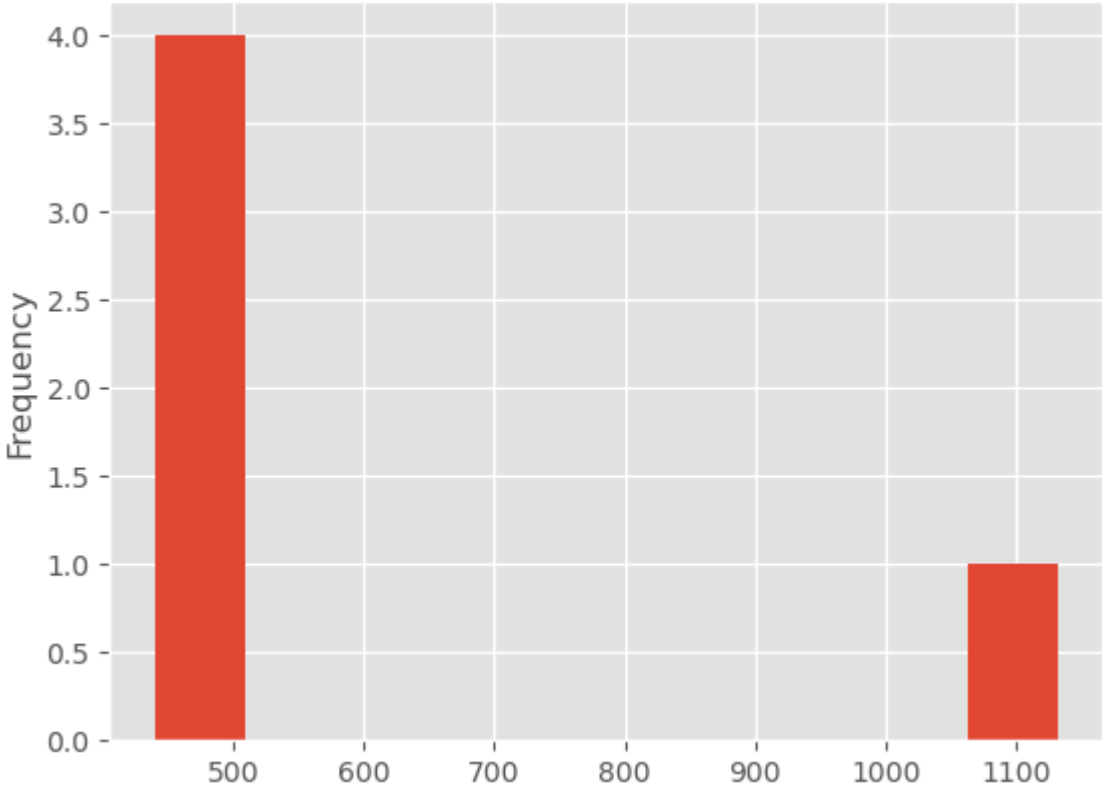
In [112…
```
dataset['Airplane_Type'].value_counts().head().plot(kind='bar')
```

Out[112]:
```
<Axes: xlabel='Airplane_Type'>
```



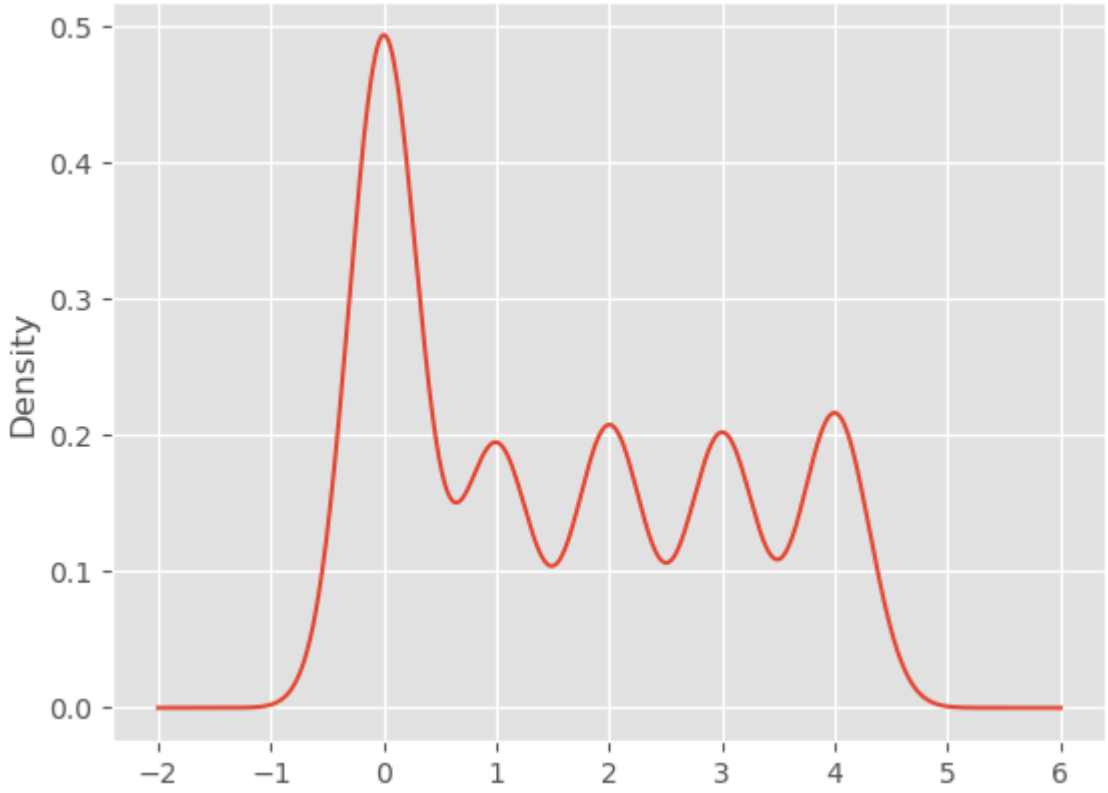
In [113…
```
dataset['Airplane_Type'].value_counts().head().plot(kind='hist')
```

Out[113]:
```
<Axes: ylabel='Frequency'>
```



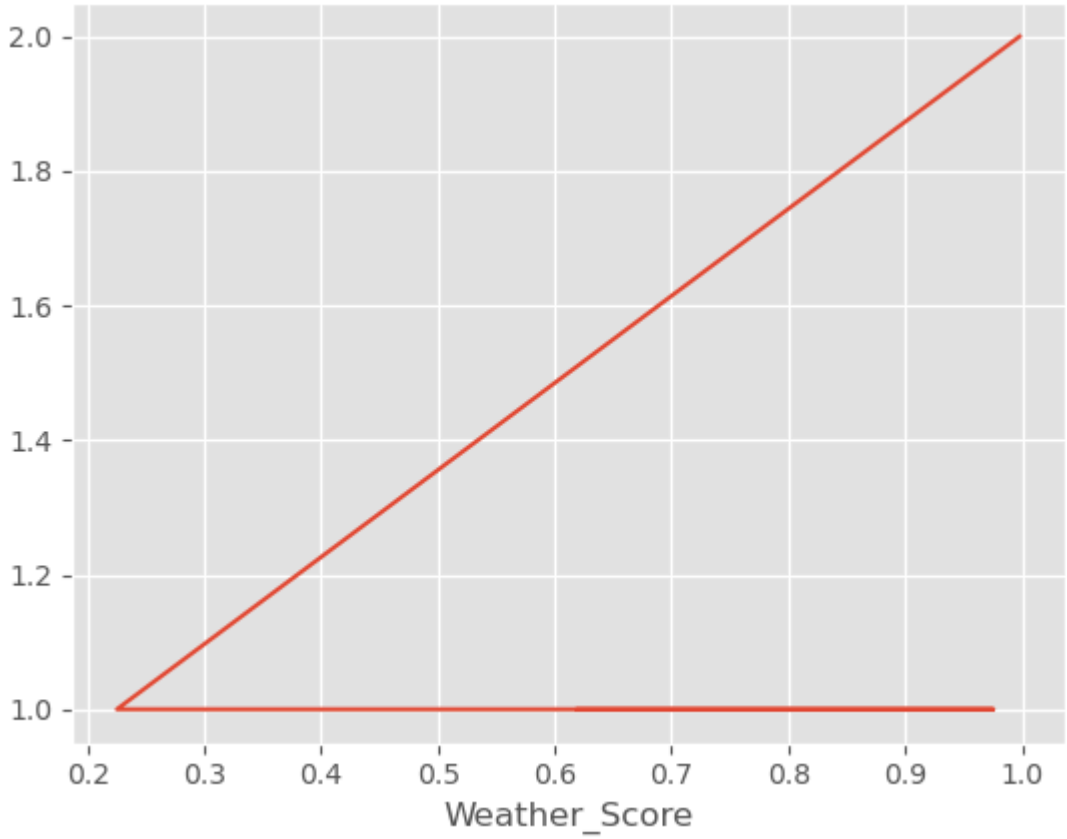
In [114…
```
dataset['Airplane_Type'].plot(kind='kde')
```

Out[114]:
```
<Axes: ylabel='Density'>
```



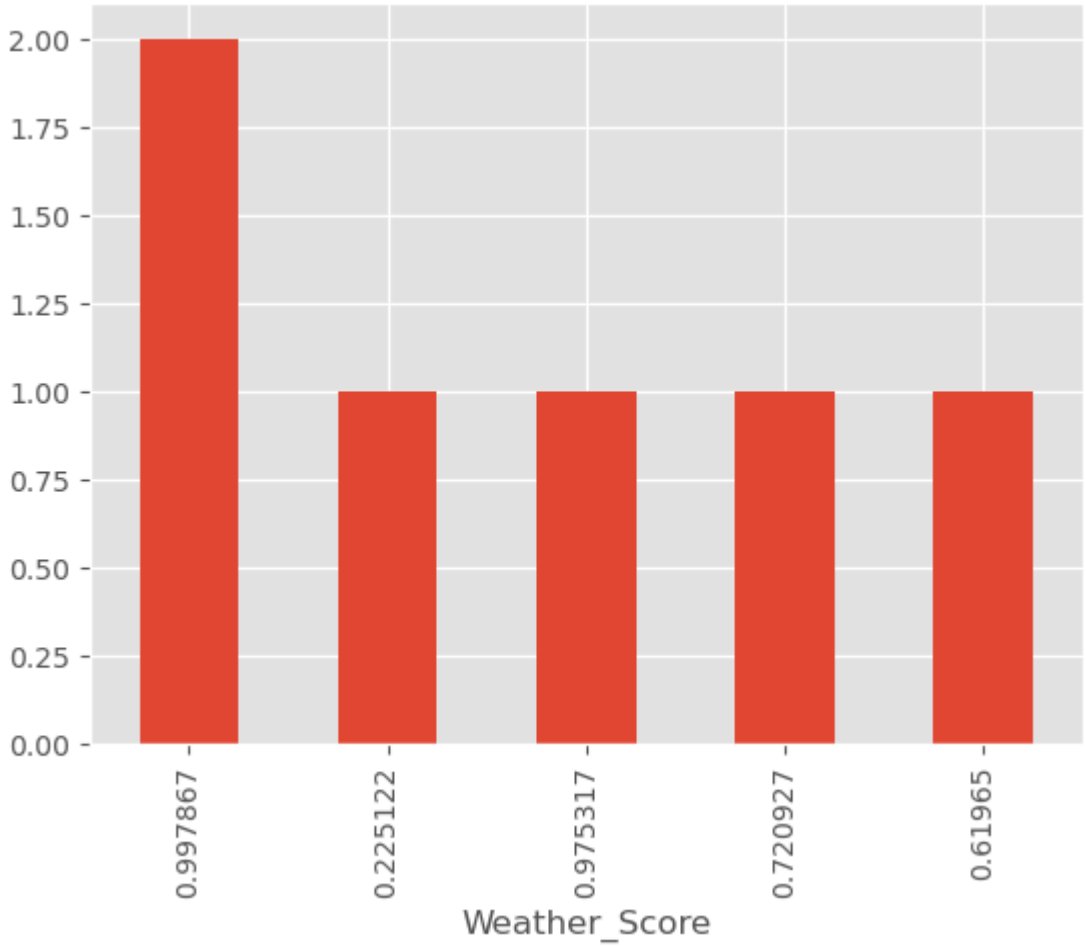
In [115…
```
dataset['Weather_Score'].value_counts().head().plot()
```

Out[115]:
```
<Axes: xlabel='Weather_Score'>
```



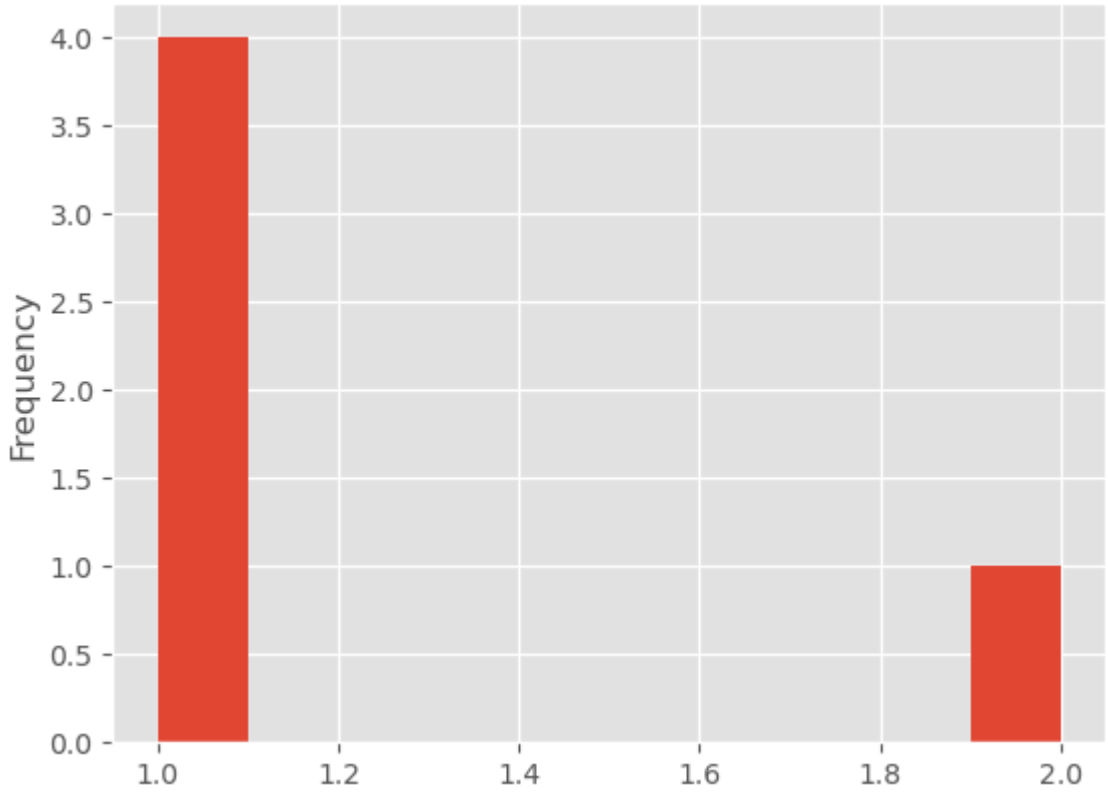
In [116…
```
dataset['Weather_Score'].value_counts().head().plot(kind='bar')
```

Out[116]:
```
<Axes: xlabel='Weather_Score'>
```



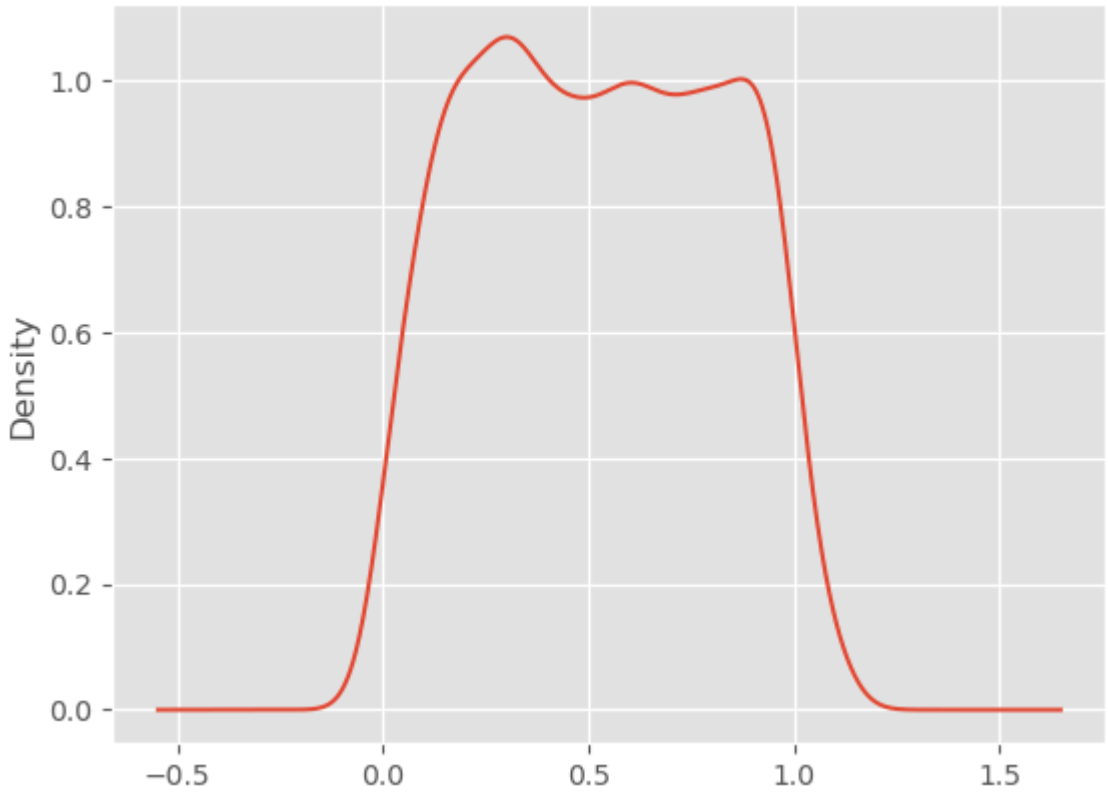
In [117…
```
dataset['Weather_Score'].value_counts().head().plot(kind='hist')
```

Out[117]:
```
<Axes: ylabel='Frequency'>
```



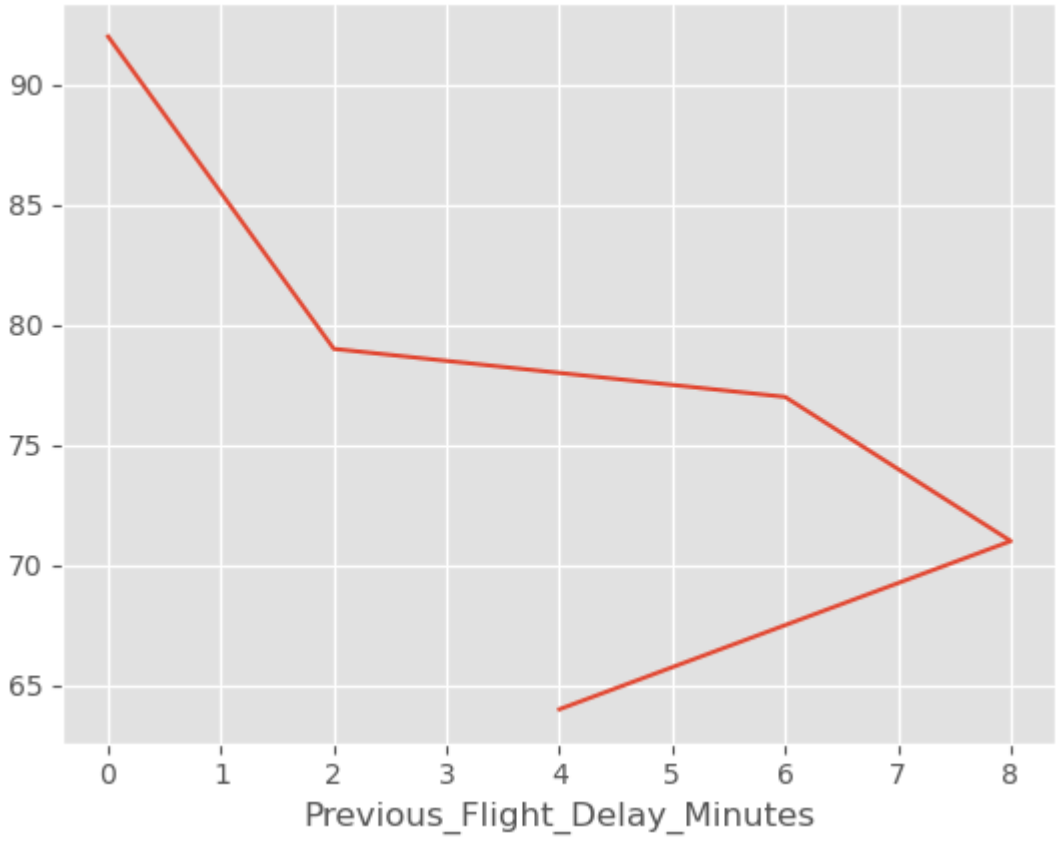
In [118…
```
dataset['Weather_Score'].plot(kind='kde')
```

Out[118]: <Axes: ylabel='Density'>



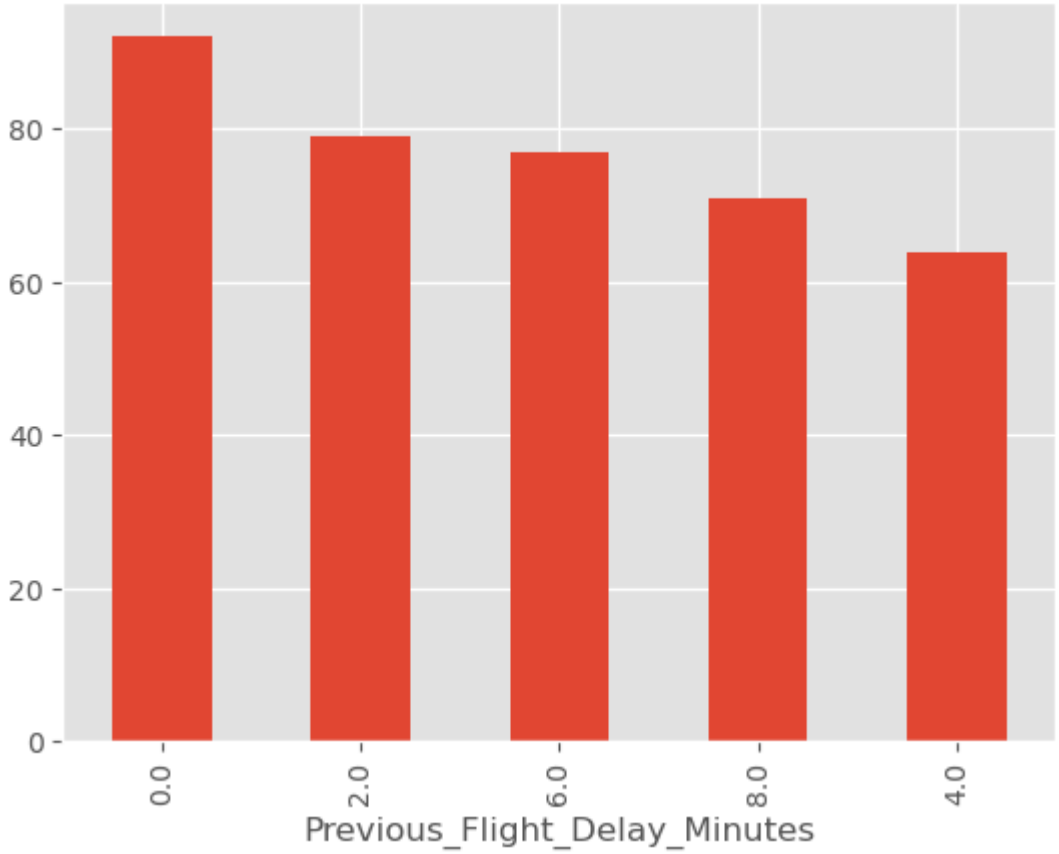
In [119…
```
dataset['Previous_Flight_Delay_Minutes'].value_counts().head().plot()
```

Out[119]: <Axes: xlabel='Previous_Flight_Delay_Minutes'>



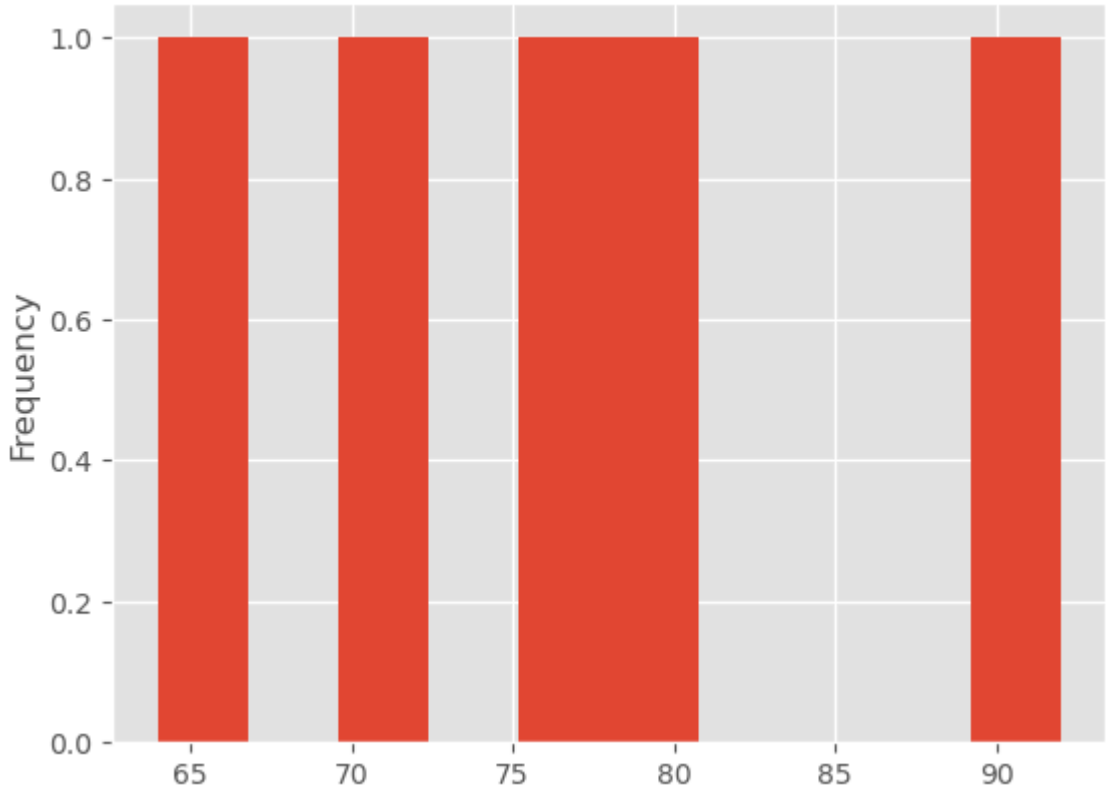
In [120…
```
dataset['Previous_Flight_Delay_Minutes'].value_counts().head().plot(kind='bar')
```

Out[120]: <Axes: xlabel='Previous_Flight_Delay_Minutes'>



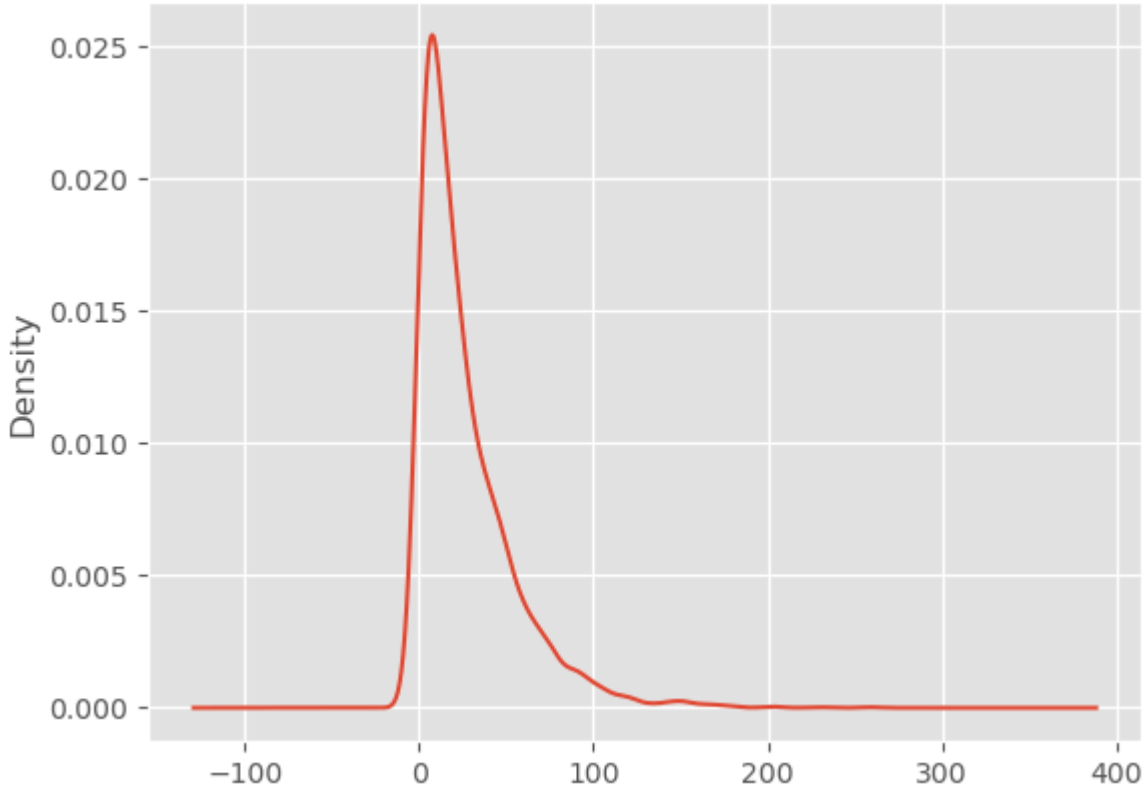
In [121…
```
dataset['Previous_Flight_Delay_Minutes'].value_counts().head().plot(kind='hist')
```

Out[121]:
```
<Axes: ylabel='Frequency'>
```



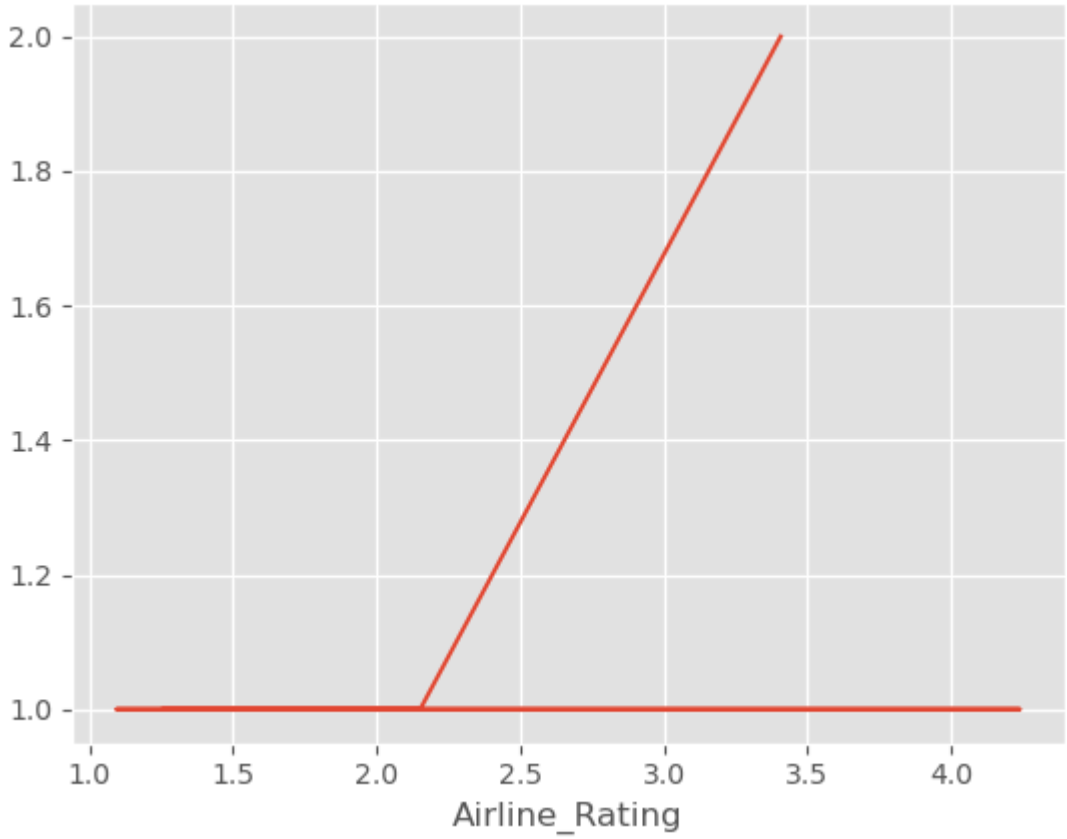
In [122…
```
dataset['Previous_Flight_Delay_Minutes'].plot(kind='kde')
```

Out[122]:
```
<Axes: ylabel='Density'>
```



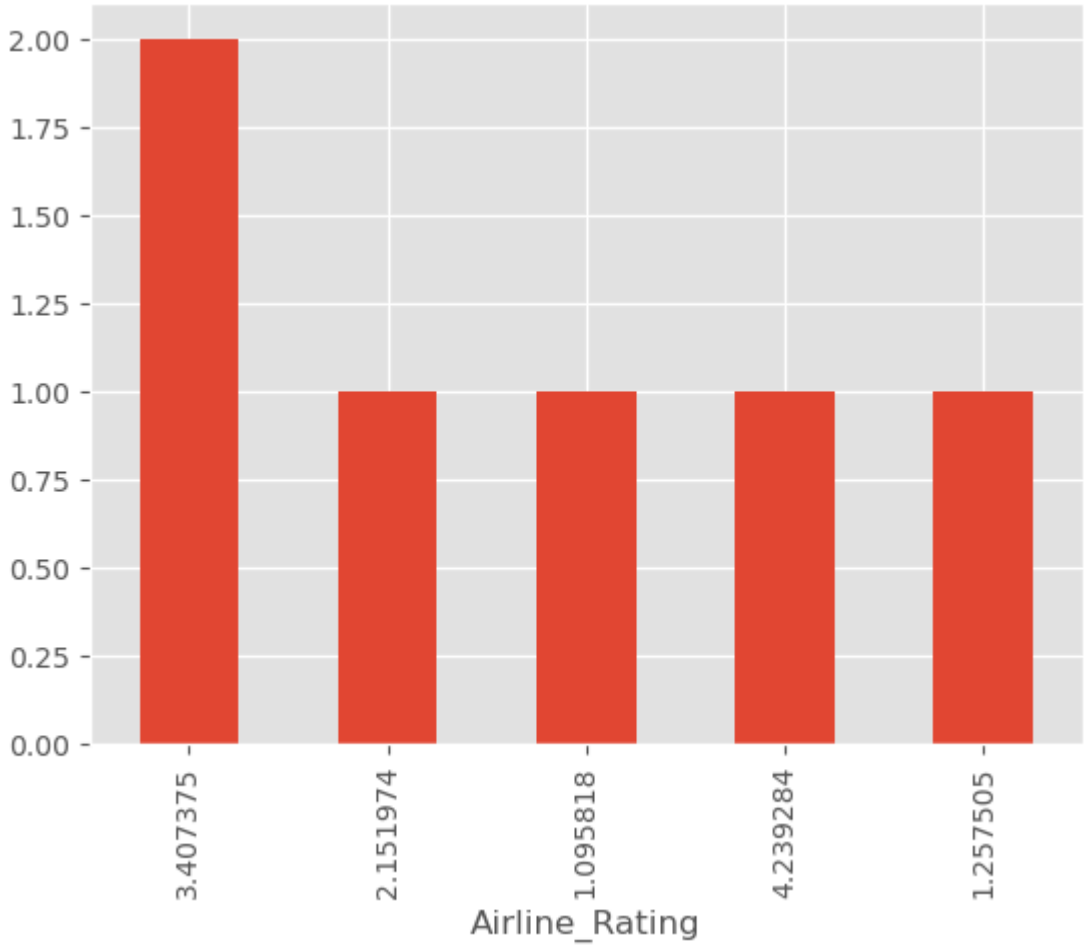
In [123…
```
dataset['Airline_Rating'].value_counts().head().plot()
```

Out[123]:
```
<Axes: xlabel='Airline_Rating'>
```



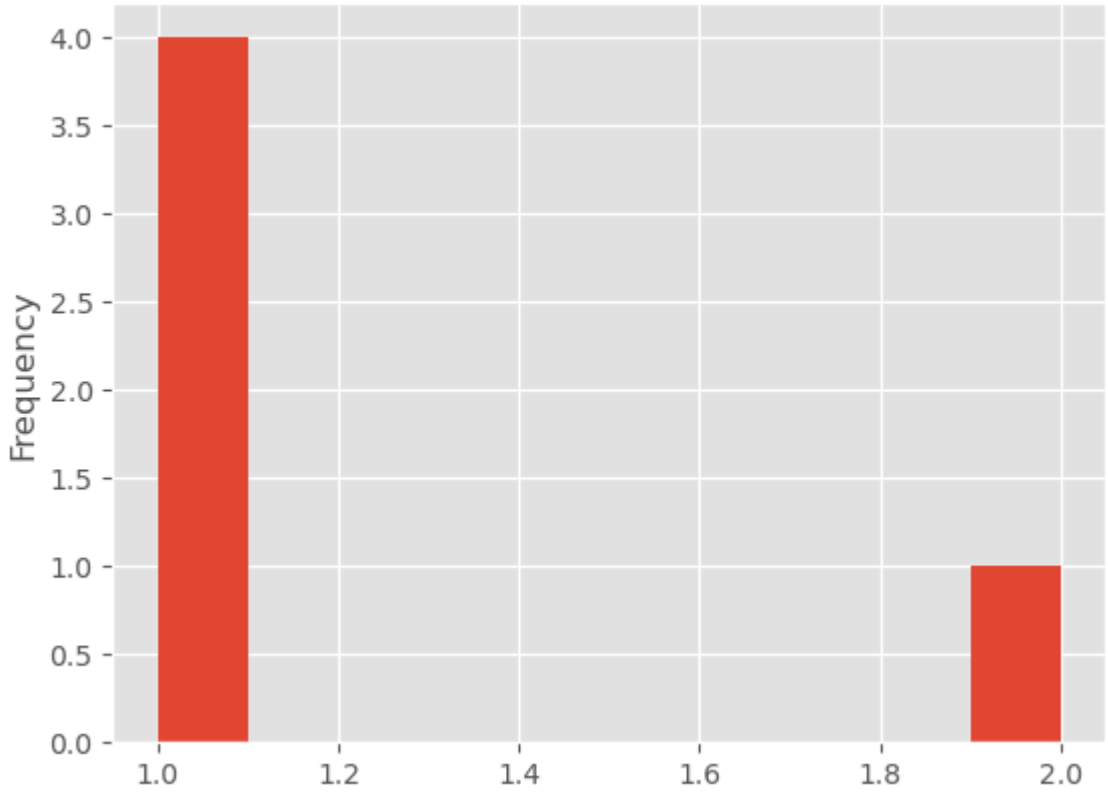
In [124…
```
dataset['Airline_Rating'].value_counts().head().plot(kind='bar')
```

Out[124]:
```
<Axes: xlabel='Airline_Rating'>
```



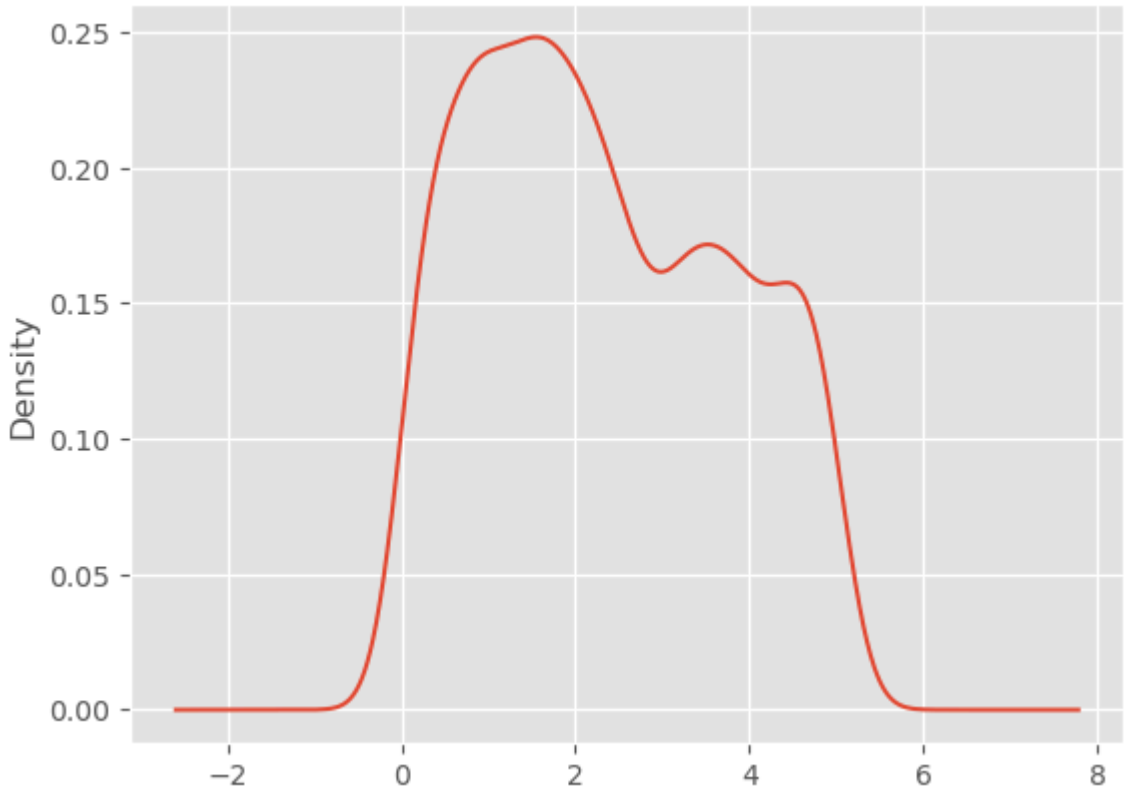
In [125…
```
dataset['Airline_Rating'].value_counts().head().plot(kind='hist')
```

Out[125]:
```
<Axes: ylabel='Frequency'>
```



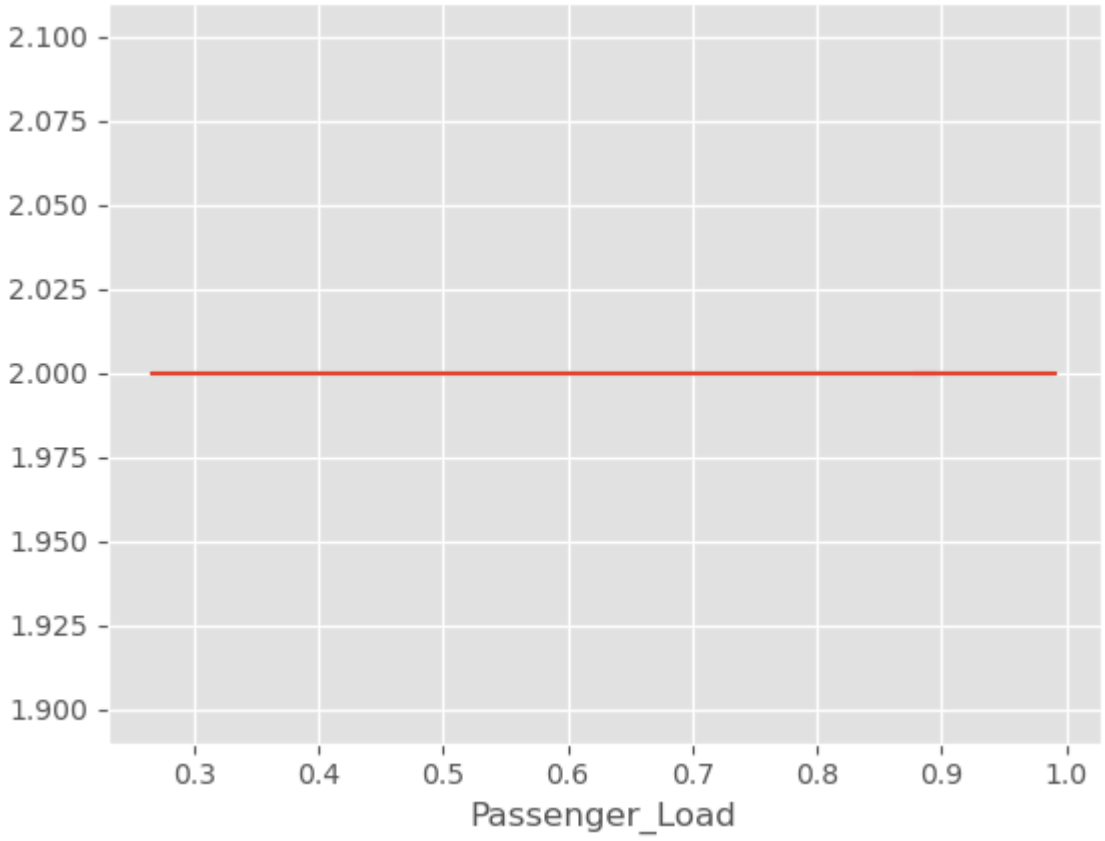
In [126…
```
dataset['Airline_Rating'].plot(kind='kde')
```

Out[126]: <Axes: ylabel='Density'>



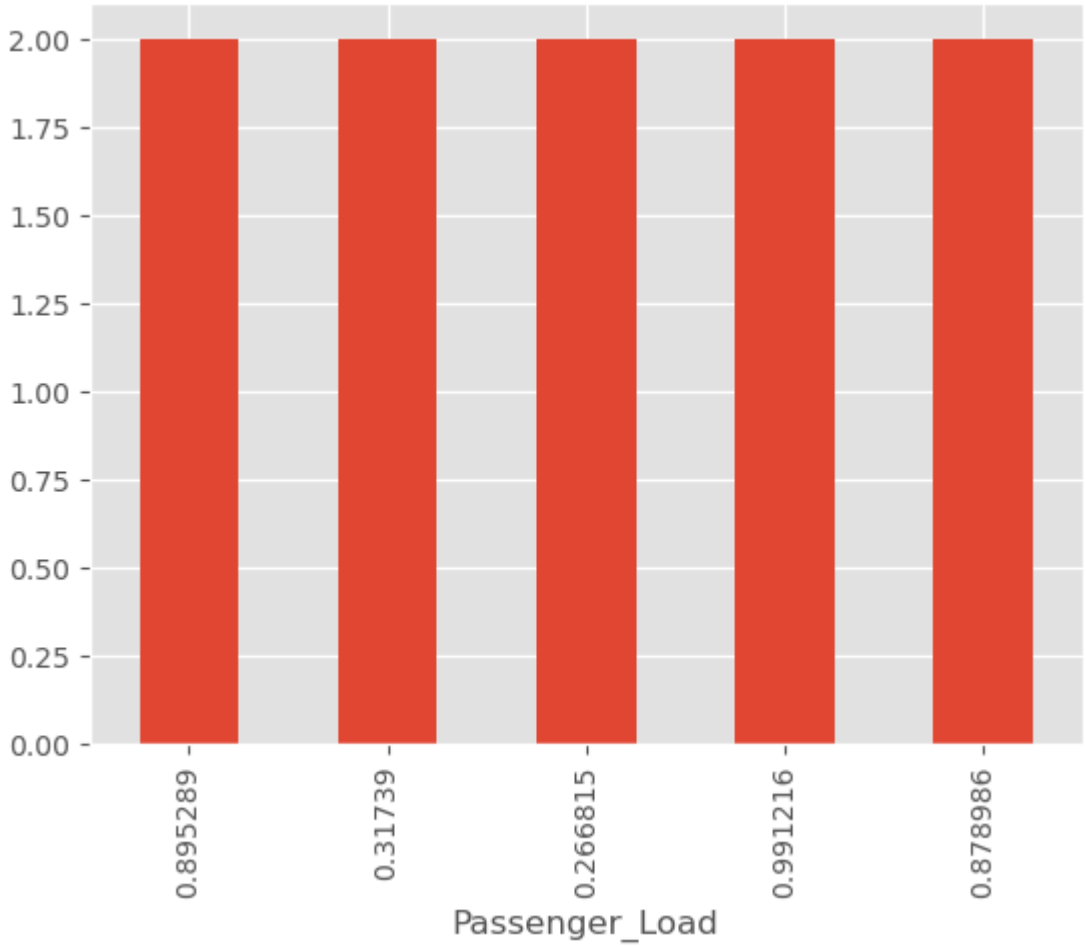
In [127…
```
dataset['Passenger_Load'].value_counts().head().plot()
```

Out[127]: <Axes: xlabel='Passenger_Load'>



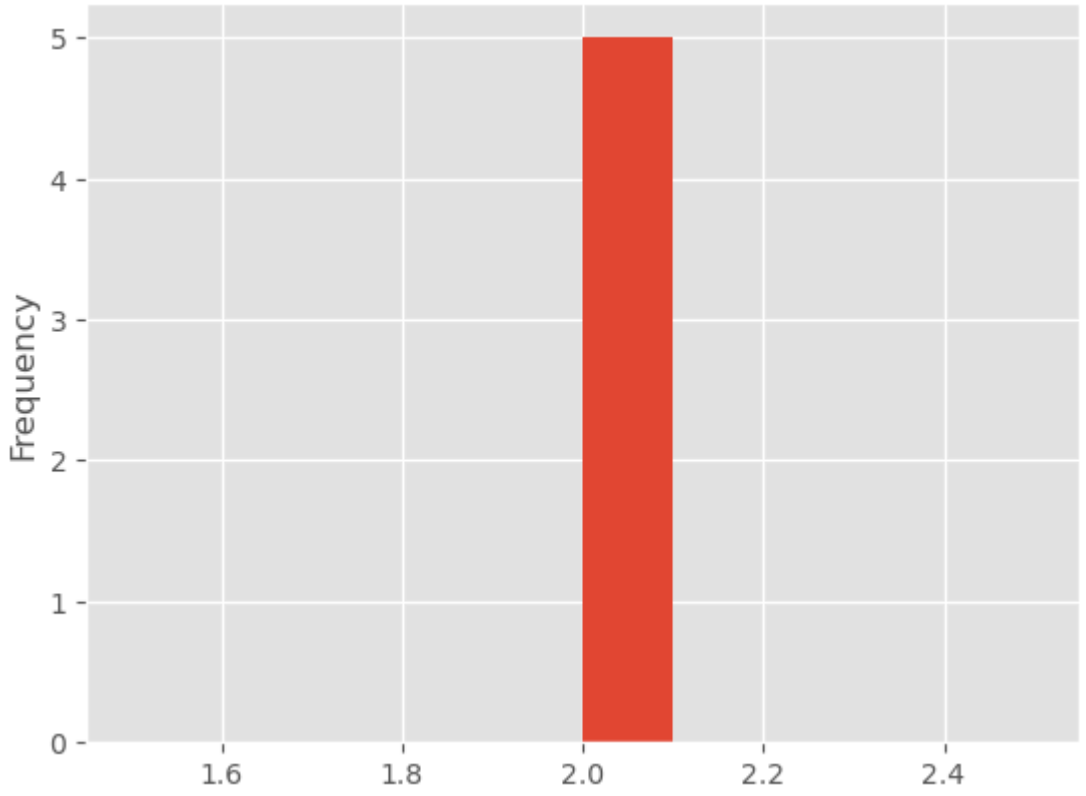
In [128…
```
dataset['Passenger_Load'].value_counts().head().plot(kind='bar')
```

Out[128]: <Axes: xlabel='Passenger_Load'>



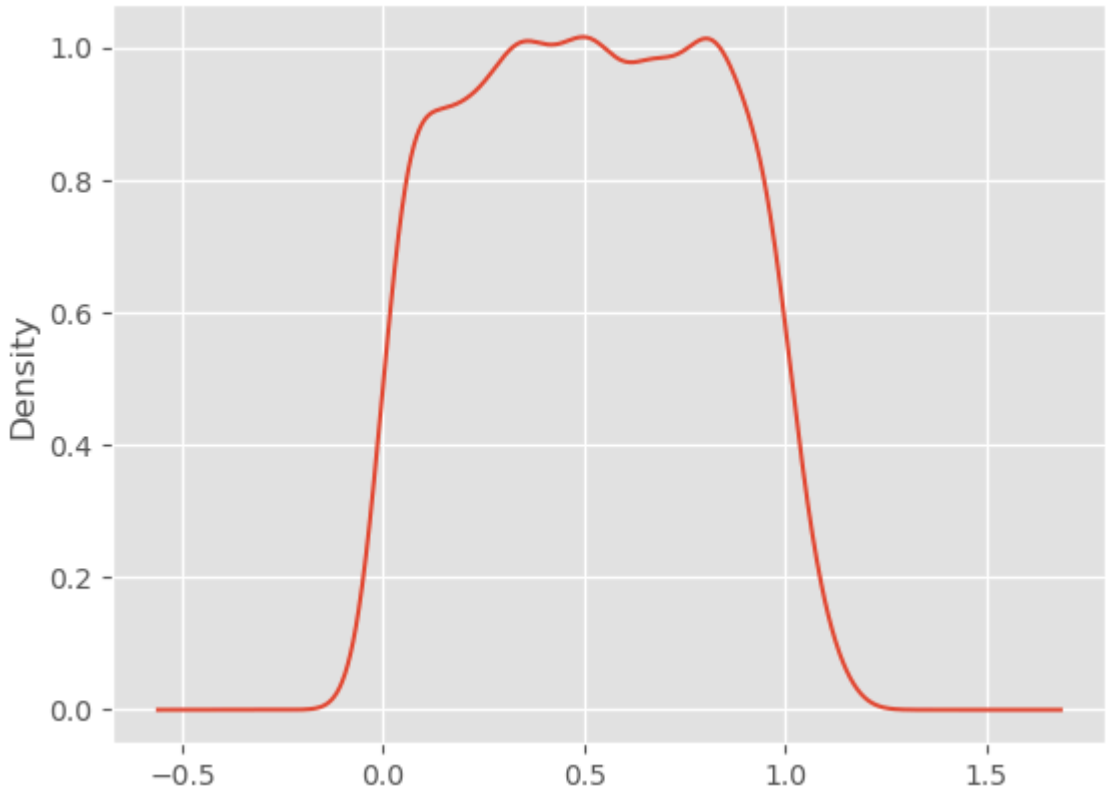
In [129…
```
dataset['Passenger_Load'].value_counts().head().plot(kind='hist')
```

Out[129]:
```
<Axes: ylabel='Frequency'>
```



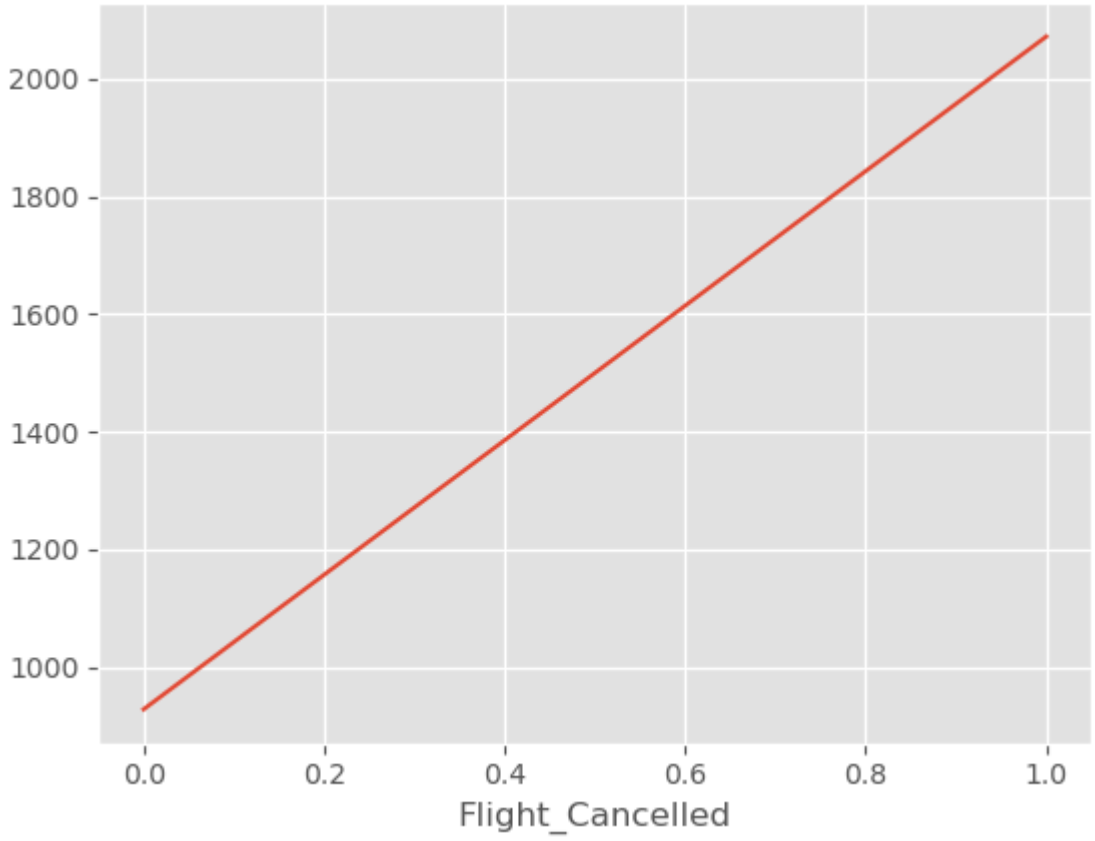
In [130…
```
dataset['Passenger_Load'].plot(kind='kde')
```

Out[130]: <Axes: ylabel='Density'>



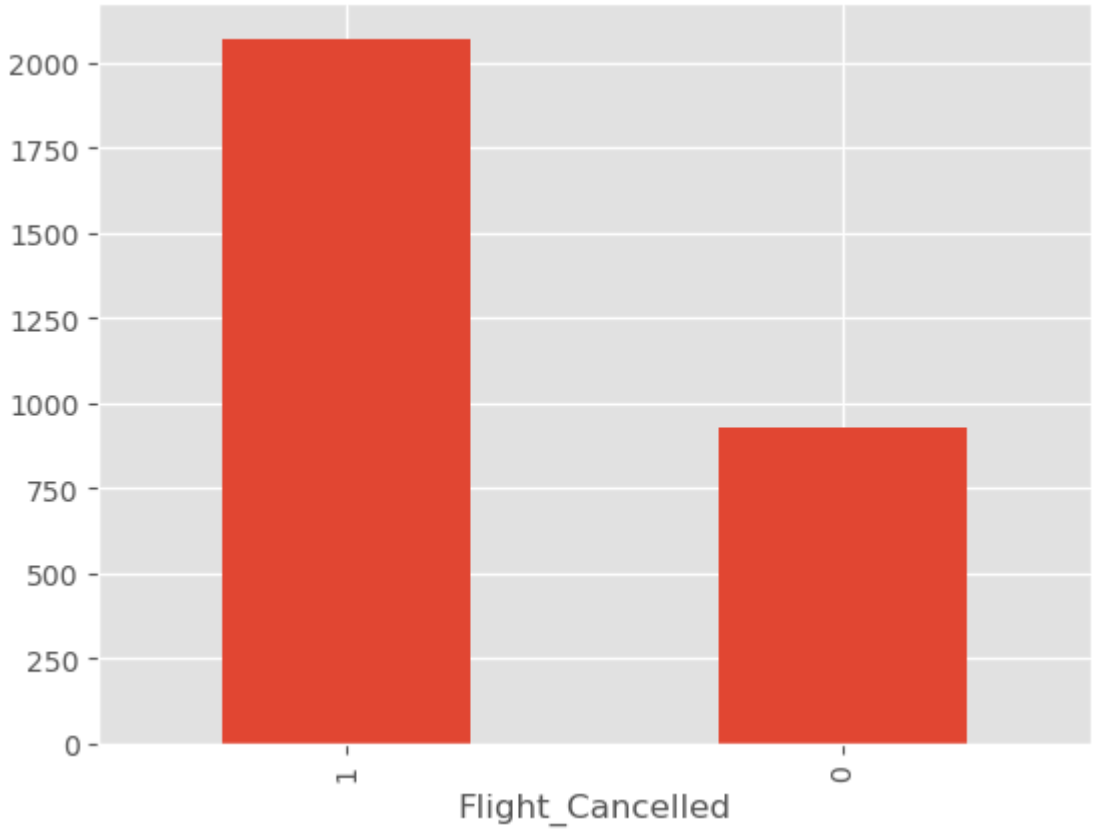
In [131…
```
dataset['Flight_Cancelled'].value_counts().head().plot()
```

Out[131]: <Axes: xlabel='Flight_Cancelled'>



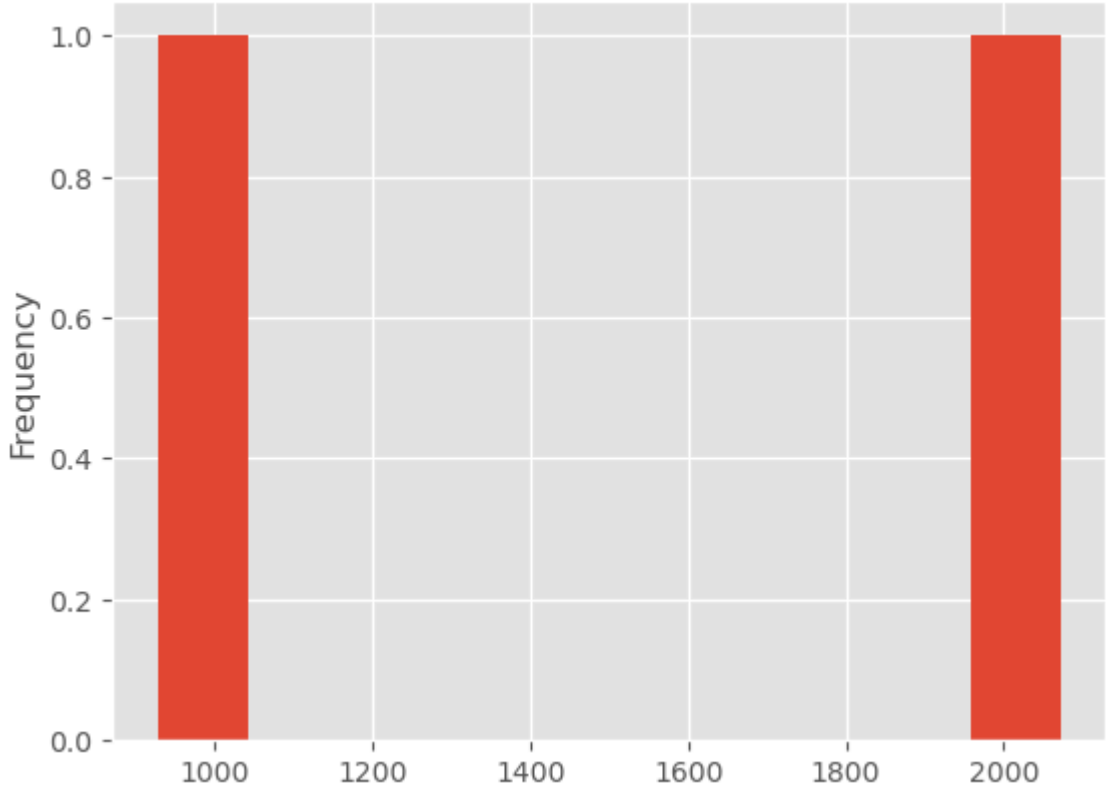
In [132…
```
dataset['Flight_Cancelled'].value_counts().head().plot(kind='bar')
```

Out[132]: <Axes: xlabel='Flight_Cancelled'>



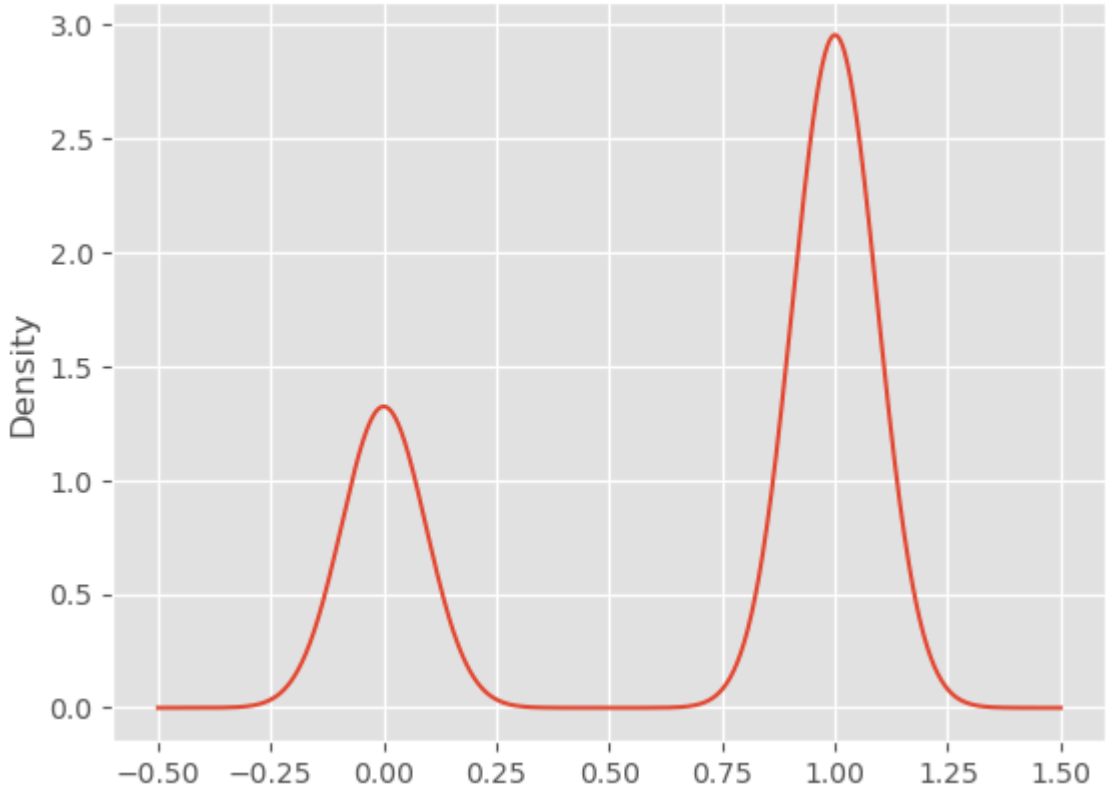
In [133…
```
dataset['Flight_Cancelled'].value_counts().head().plot(kind='hist')
```

Out[133]:
```
<Axes: ylabel='Frequency'>
```



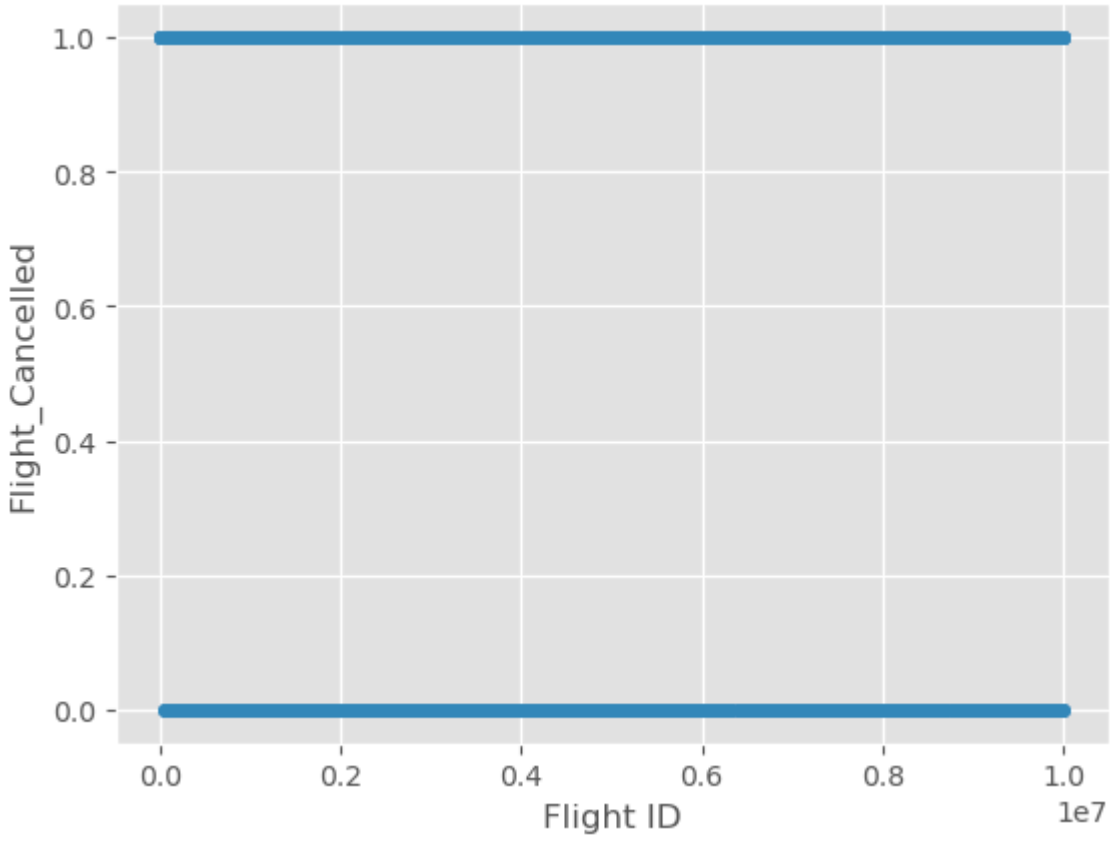
In [134…
```
dataset['Flight_Cancelled'].plot(kind='kde')
```

Out[134]:
```
<Axes: ylabel='Density'>
```



In [135…
```
#USING SCATTER PLOTS TO VISUALIZE THE TREND OF THE NUMBER OF FLIGHTS CANCELLED
dataset.plot(kind='scatter',
             x='Flight ID',
             y='Flight_Cancelled')
```

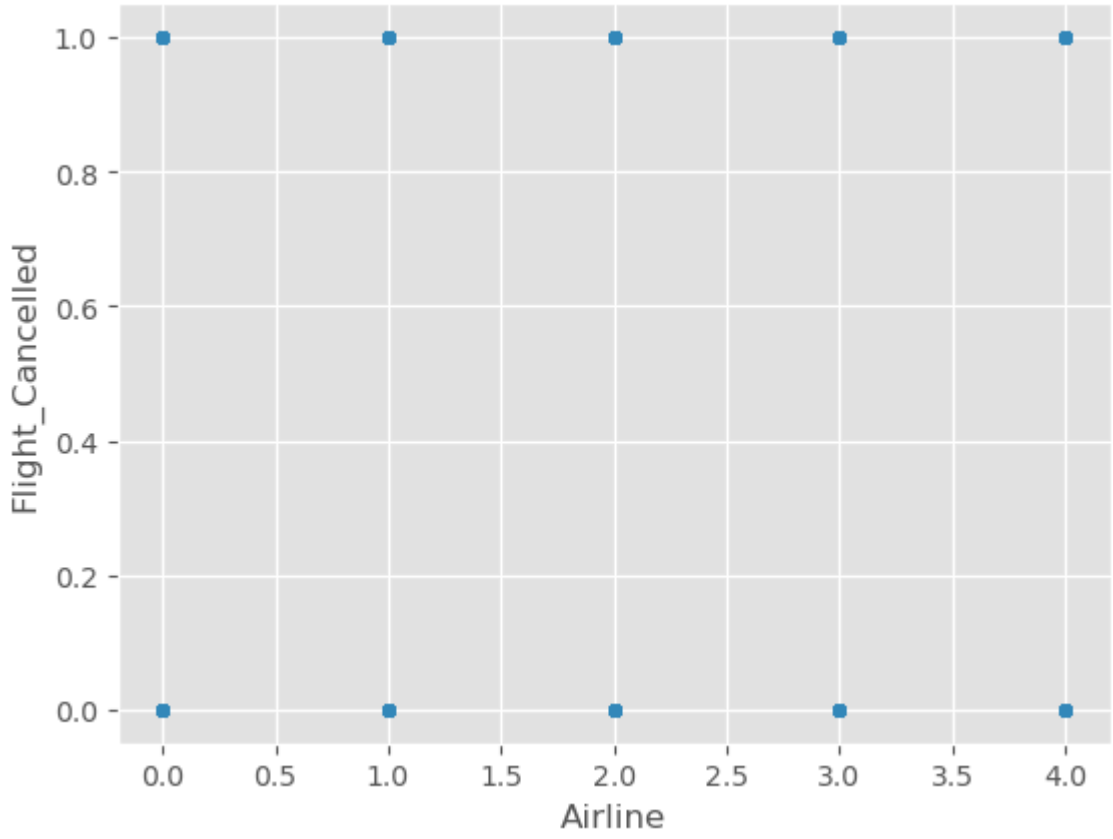
Out[135]:
```
<Axes: xlabel='Flight ID', ylabel='Flight_Cancelled'>
```



In [136…
```
#USING SCATTER PLOTS TO COMPARE THE NUMBER OF CANCELLATIONS ACROSS DIFFERENT AIRLIN
dataset.plot(kind='scatter',
```

```
            x='Airline',
            y='Flight_Cancelled')
```

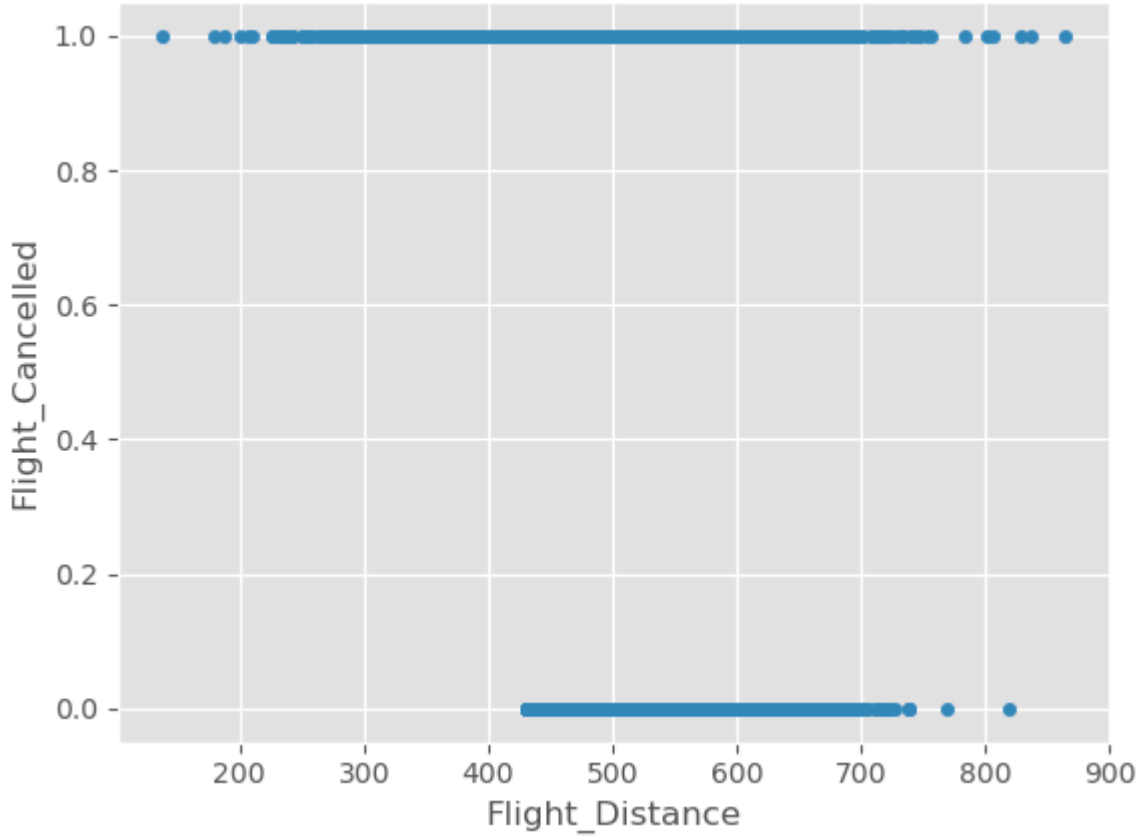
Out[136]: <Axes: xlabel='Airline', ylabel='Flight_Cancelled'>



In [137…

```
#USING SCATTER PLOT TO ANALYZE IF LONGER OR SHORTER FLIGHTS HAVE DIFFERENT CANCELLA
dataset.plot(kind='scatter',
            x='Flight_Distance',
            y='Flight_Cancelled')
```

Out[137]: <Axes: xlabel='Flight_Distance', ylabel='Flight_Cancelled'>

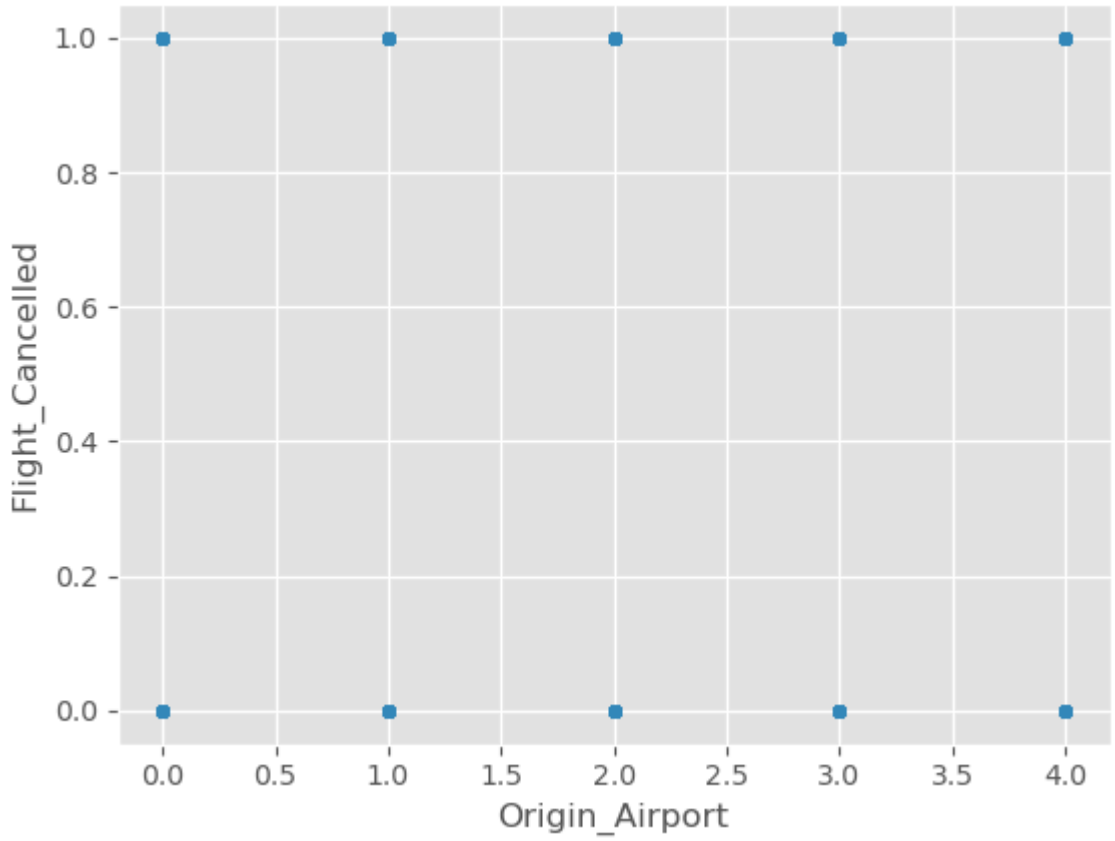


In [138…

```
##USING SCATTER PLOT TO ANALYZE AND COMPARE THE NUMBER OF FLIGHT CANCELLATIONS ACRO
dataset.plot(kind='scatter',
             x='Origin_Airport',
             y='Flight_Cancelled')
```

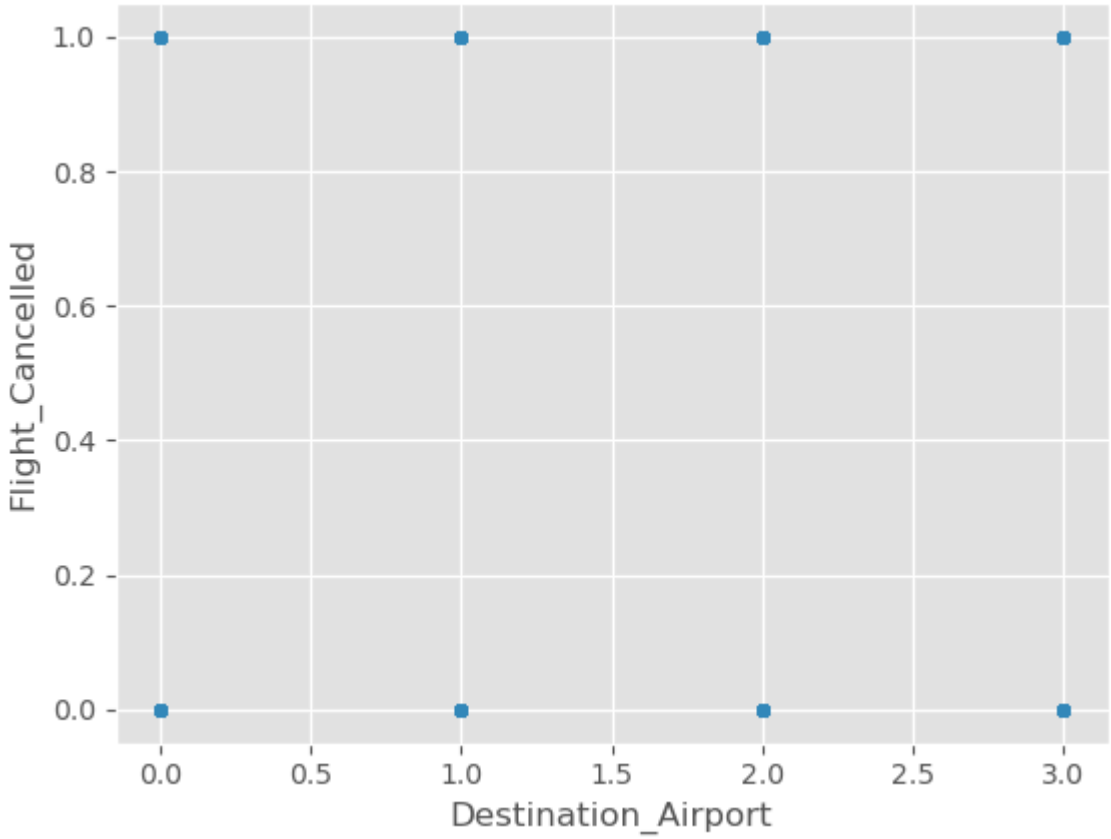
Out[138]:

```
<Axes: xlabel='Origin_Airport', ylabel='Flight_Cancelled'>
```



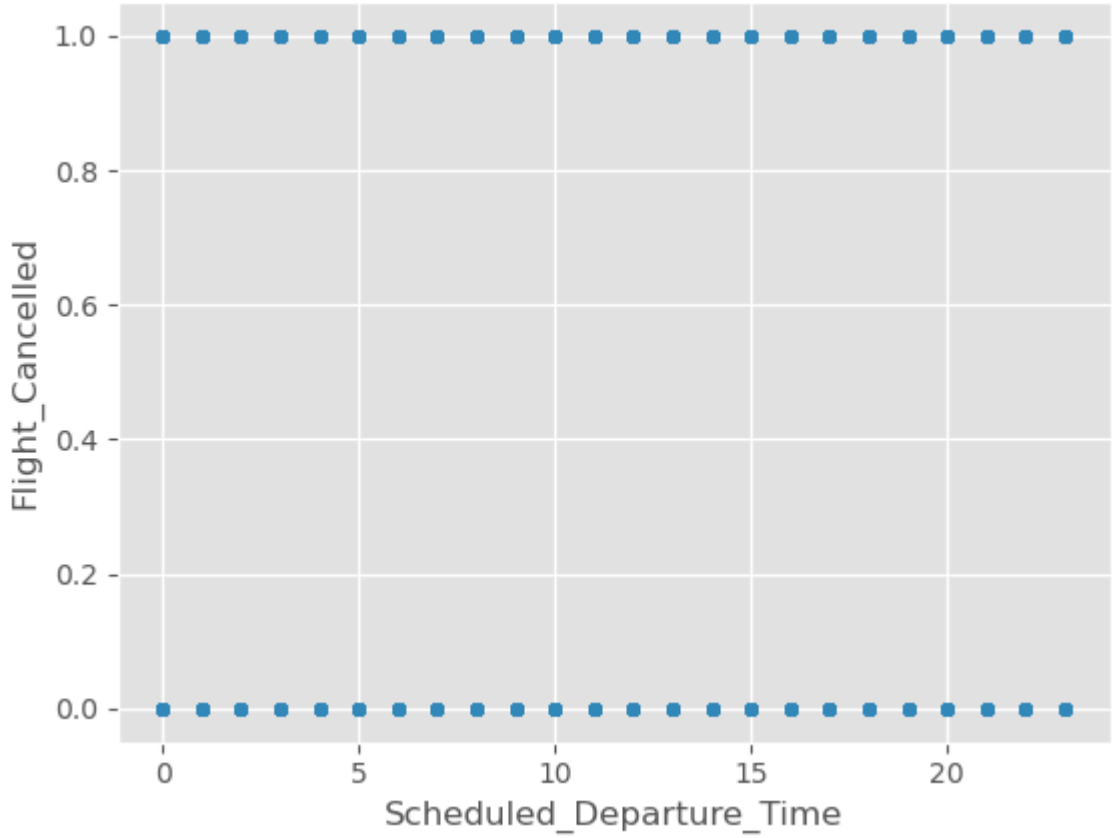
In [139…
```
#USING SCATTER PLOT TO ANALYZE AND COMPARE THE NUMBER OF FLIGHT CANCELLATIONS ACROS
dataset.plot(kind='scatter',
             x='Destination_Airport',
             y='Flight_Cancelled')
```

Out[139]:
```
<Axes: xlabel='Destination_Airport', ylabel='Flight_Cancelled'>
```



In [140…
```
#USING SCATTER PLOT TO SEE IF THERE ARE SPECIFIC TIMES OF THE DAY WHEN CANCELLATION
dataset.plot(kind='scatter',
             x='Scheduled_Departure_Time',
             y='Flight_Cancelled')
```

Out[140]:
```
<Axes: xlabel='Scheduled_Departure_Time', ylabel='Flight_Cancelled'>
```

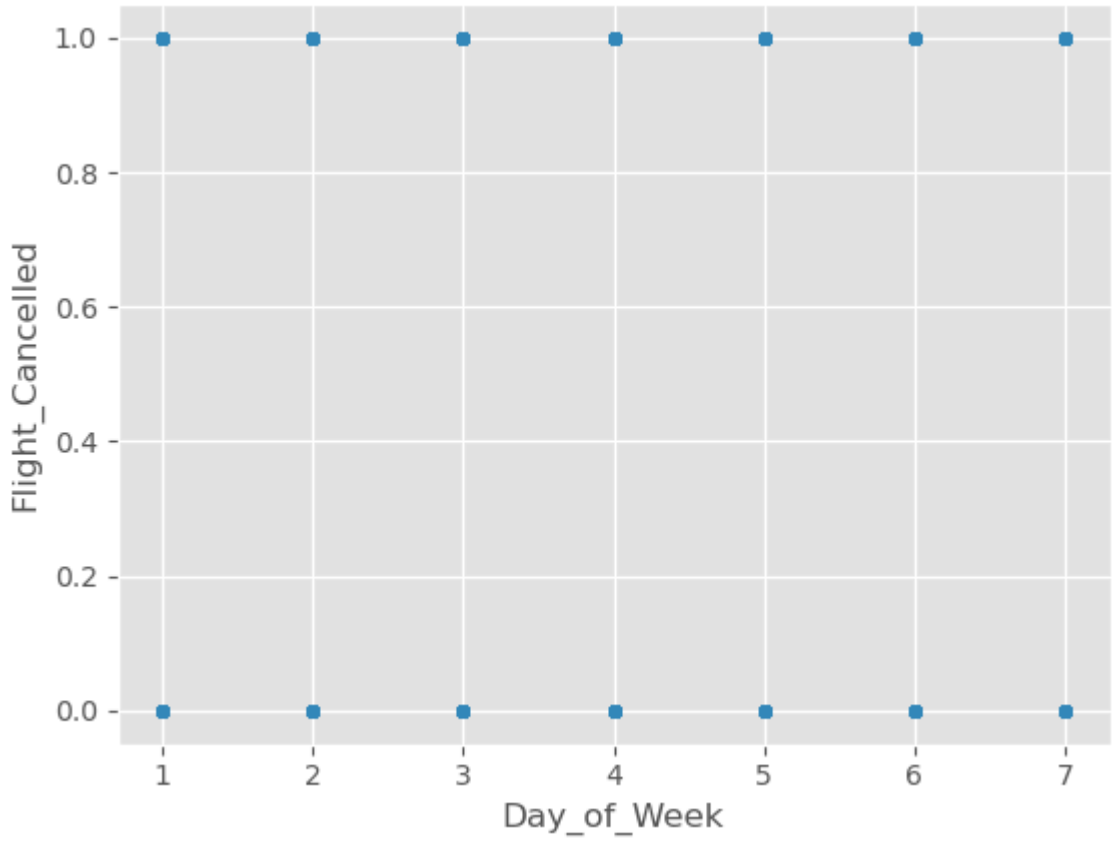


In [141…

```
#USING SCATTER PLOT TO DETERMINE IF CANCELLATIONS ARE MORE COMMON ON CERTAIN DAYS
dataset.plot(kind='scatter',
             x='Day_of_Week',
             y='Flight_Cancelled')
```

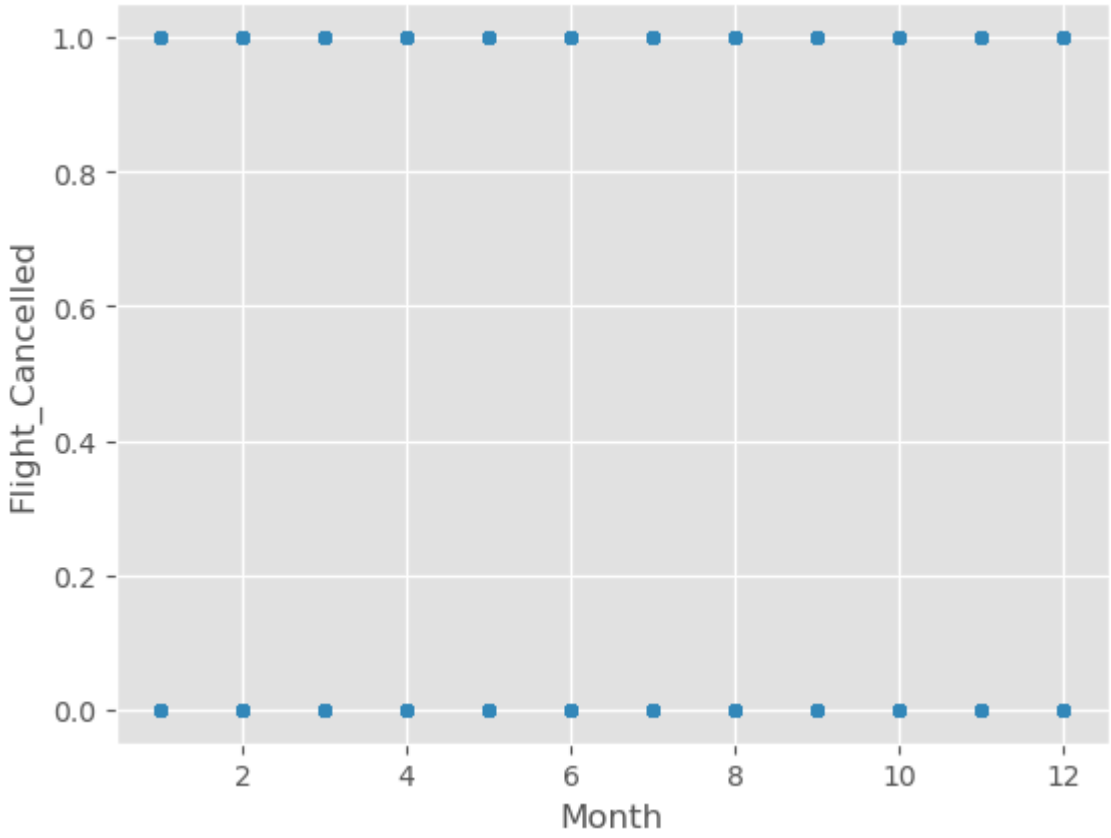
Out[141]:

```
<Axes: xlabel='Day_of_Week', ylabel='Flight_Cancelled'>
```



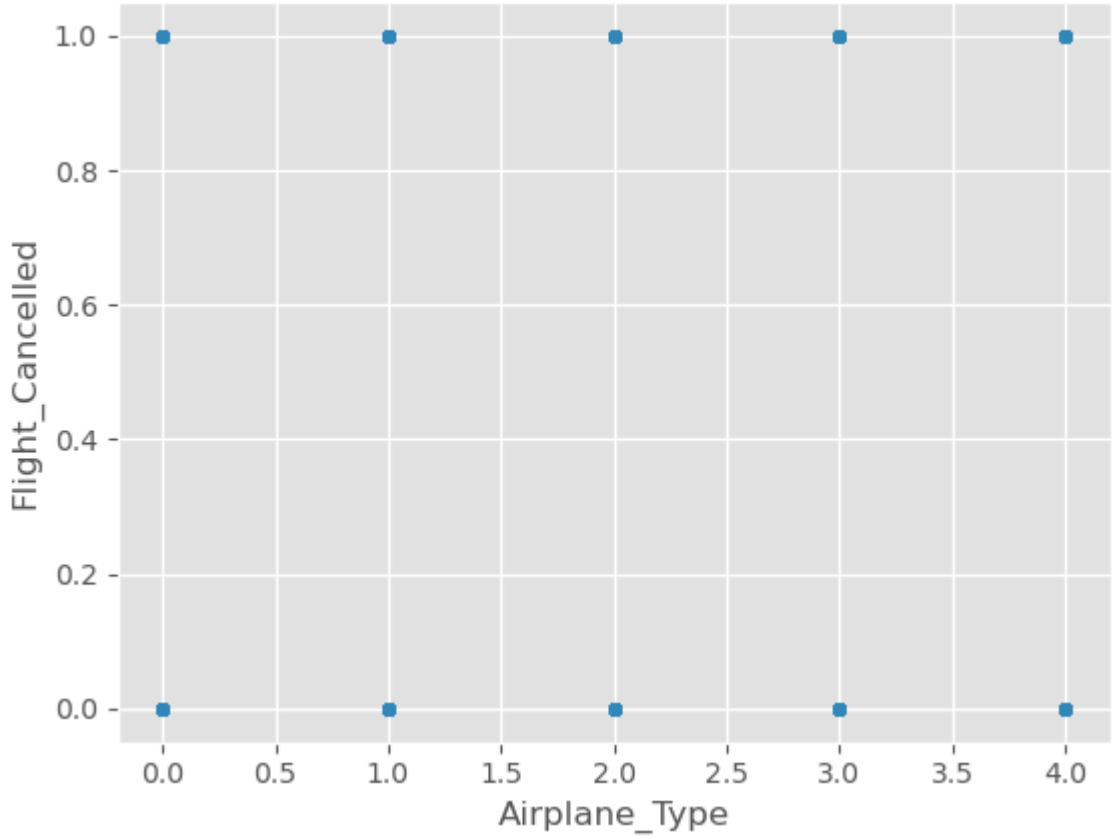
In [142…
```
#USING SCATTER PLOT TO COMPARE THE NUMBER OF FLIGHT CANCELLATIONS ACROSS DIFFERENT
dataset.plot(kind='scatter',
             x='Month',
             y='Flight_Cancelled')
```

Out[142]:
```
<Axes: xlabel='Month', ylabel='Flight_Cancelled'>
```



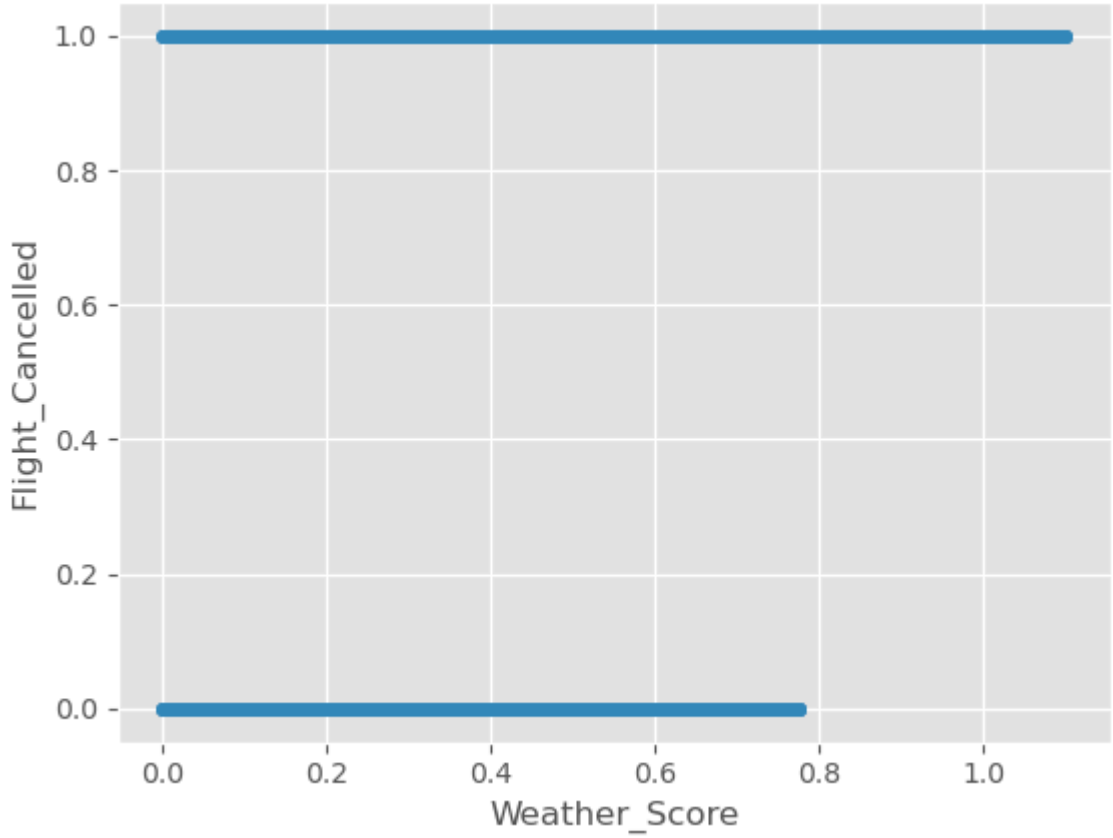
In [143…
```
#USING SCATTER PLOT TO COMPARE THE NUMBER OF FLIGHT CANCELLATIONS ACROSS DIFFERENT
#THIS WILL HELP IDENTIFY IF CERTAIN TYPES OF AIRPLANES HAVE HIGHER CANCELLATION RAT
dataset.plot(kind='scatter',
             x='Airplane_Type',
             y='Flight_Cancelled')
```

Out[143]:
```
<Axes: xlabel='Airplane_Type', ylabel='Flight_Cancelled'>
```



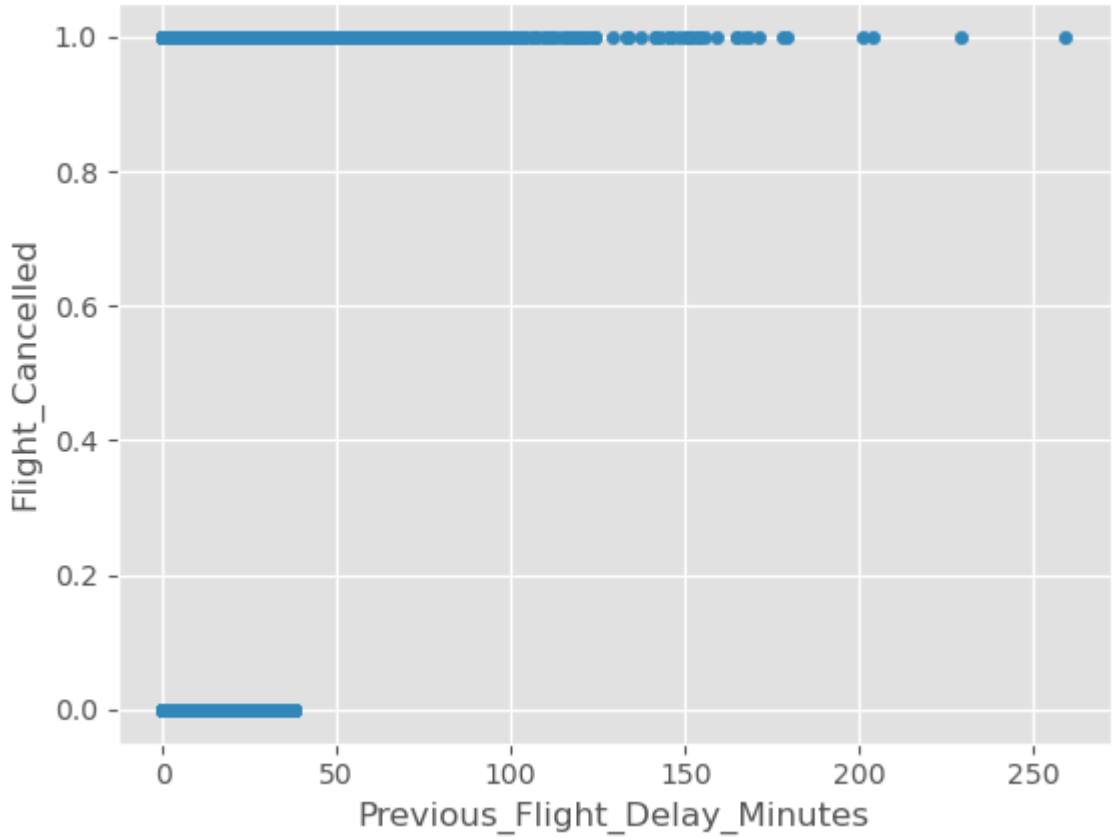
In [144…
```
#USING SCATTER PLOT TO ANALYZE THE RELATIONSHIP BETWEEN WEATHER CONDITIONS AND THE
#THIS WILL HELP IDENTIFY HOW DIFFERENT WEATHER CONDITIONS AFFECT FLIGHT CANCELLATIC
dataset.plot(kind='scatter',
             x='Weather_Score',
             y='Flight_Cancelled')
```

Out[144]:
```
<Axes: xlabel='Weather_Score', ylabel='Flight_Cancelled'>
```



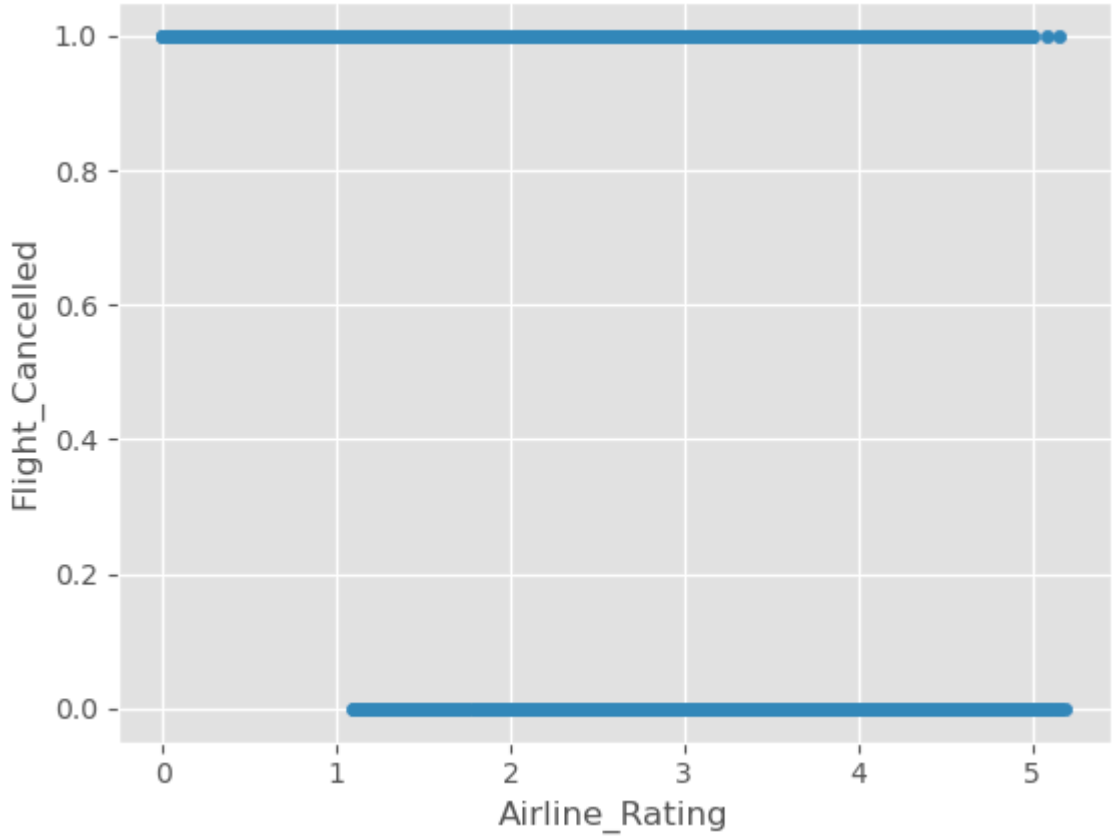
In [145…
```
#USING SCATTER PLOT TO ANALYZE THE RELATIONSHIP BETWEEN DELAYS IN PREVIOUS FLIGHTS
dataset.plot(kind='scatter',
             x='Previous_Flight_Delay_Minutes',
             y='Flight_Cancelled')
```

Out[145]:
```
<Axes: xlabel='Previous_Flight_Delay_Minutes', ylabel='Flight_Cancelled'>
```



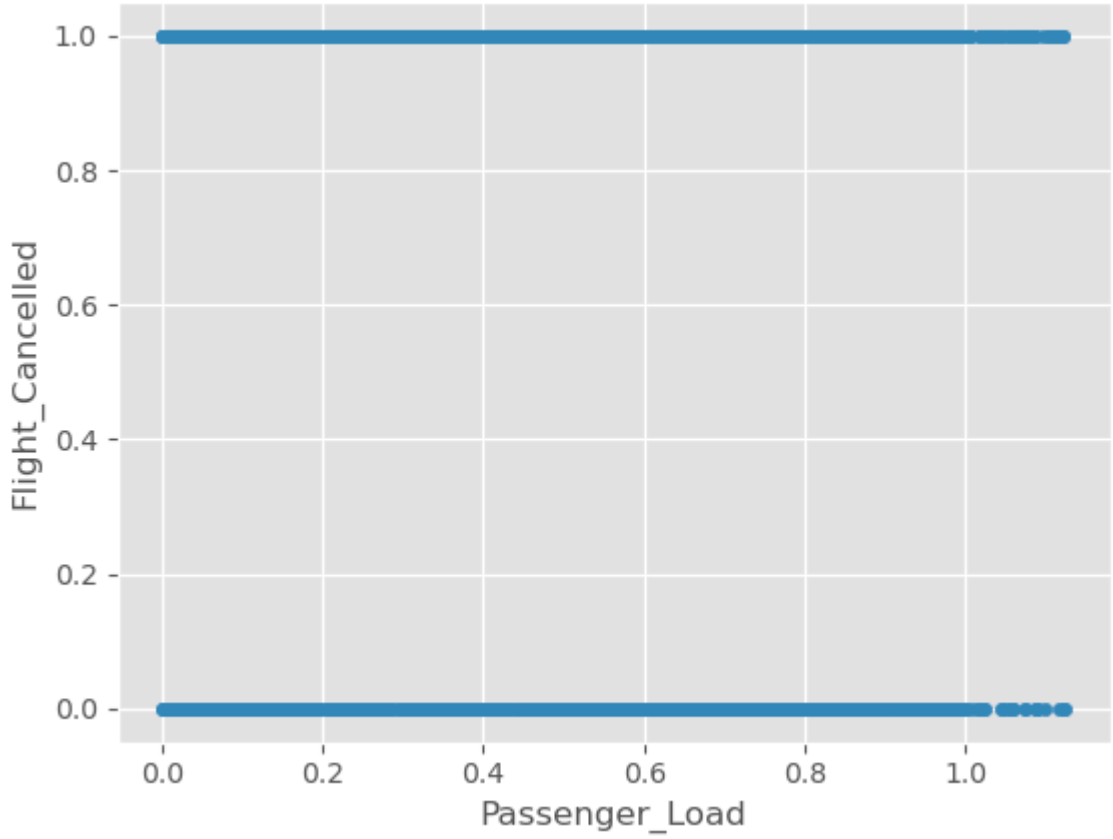
In [146…
```
#USING SCATTER PLOT TO ANALYZE THE RELATONSHIP BETWEEN AIRLINE RATINGS AND THE NUMB
#IDENTIFY IF HIGHER -RATED AIRLINES HAVE FEWER CANCELLATIONS AND LOWER-RATED AIRLIN
dataset.plot(kind='scatter',
             x='Airline_Rating',
             y='Flight_Cancelled')
```

Out[146]:
```
<Axes: xlabel='Airline_Rating', ylabel='Flight_Cancelled'>
```



In [147…
```
#USING SCATTER PLOT TO HELP US IDENTIFY IF FLIGHTS WITH HIGHER OR LOWER PASSENGER L
dataset.plot(kind='scatter',
             x='Passenger_Load',
             y='Flight_Cancelled')
```

Out[147]:
```
<Axes: xlabel='Passenger_Load', ylabel='Flight_Cancelled'>
```



In [ ]:

In [ ]: