Practical 1

In [ ]:

```
NAME: Gayatri Ghorpade
ROLLNO:14144
```

In [30]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [31]:

```
df=pd.read_csv("uber.csv")
df
```

Out[31]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	p
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	
...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	

200000 rows × 9 columns

In [32]:

```
df.head()
```

Out[32]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	4
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	4
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	4
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	4
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	4

In [33]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 9 columns):
 #   Column             Non-Null  Count   Dtype
---  ------             --------------  -----
 0   Unnamed: 0         200000  non-null  int64
 1   key                200000  non-null  object
 2   fare_amount        200000  non-null  float64
 3   pickup_datetime    200000  non-null  object
 4   pickup_longitude   200000  non-null  float64
 5   pickup_latitude    200000  non-null  float64
 6   dropoff_longitude  199999  non-null  float64
 7   dropoff_latitude   199999  non-null  float64
 8   passenger_count    200000  non-null  int64
dtypes: float64(5), int64(2), object(2)
memory usage: 13.7+ MB
```

In [34]:

```
df.columns
```

Out[34]:

```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

In [35]:

```
df =df.drop(['Unnamed: 0', 'key'], axis =1)
```

In [36]:

```
df.head()
```

Out[36]:

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dr
0	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-73.999512	
1	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-73.994710	
2	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-73.962565	
3	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-73.965316	
4	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-73.973082	

In [37]:

```
df.shape
```

Out[37]:

```
(200000, 7)
```

In [38]:

```
df.dtypes
```

Out[38]:

```
fare_amount            float64
pickup_datetime         object
pickup_longitude       float64
pickup_latitude        float64
dropoff_longitude      float64
dropoff_latitude       float64
passenger_count          int64
dtype: object
```

In [39]:

```
df.describe()
```

Out[39]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitud
count	200000.000000	200000.000000	200000.000000	199999.000000	199999.00000
mean	11.359955	-72.527638	39.935885	-72.525292	39.92389
std	9.901776	11.437787	7.720539	13.117408	6.79482
min	-52.000000	-1340.648410	-74.015515	-3356.666300	-881.98551
25%	6.000000	-73.992065	40.734796	-73.991407	40.73382
50%	8.500000	-73.981823	40.752592	-73.980093	40.75304
75%	12.500000	-73.967154	40.767158	-73.963658	40.76800
max	499.000000	57.418457	1644.421482	1153.572603	872.69762

In [40]:

```
df.isnull().sum()
```

Out[40]:

```
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
dtype: int64
```

In [41]:

```
df['dropoff_latitude'].fillna(value=df['dropoff_latitude'].mean(), inplace=True)
```

In [42]:

```
df.isnull().sum()
```

Out[42]:

```
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     0
passenger_count      0
dtype: int64
```

In [43]:

```
df['dropoff_longitude'].fillna(value=df['dropoff_longitude'].median(), inplace=True
```

In [44]:

```
df.isnull().sum()
```

Out[44]:

```
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    0
dropoff_latitude     0
passenger_count      0
dtype: int64
```

In [45]:

```
df.dtypes
```

Out[45]:

```
fare_amount          float64
pickup_datetime       object
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
passenger_count        int64
dtype: object
```

In [46]:

```
df.pickup_datetime = pd.to_datetime(df.pickup_datetime, errors='coerce')
```

In [47]:

```
df.dtypes
```

```
Out[47]: fare_amount                       float64
         pickup_datetime       datetime64[ns, UTC]
         pickup_longitude                  float64
         pickup_latitude                   float64
         dropoff_longitude                 float64
         dropoff_latitude                  float64
         passenger_count                     int64
         dtype: object
```

```
In [48]: df= df.assign(hour = df.pickup_datetime.dt.hour,
          day = df.pickup_datetime.dt.day,
          month = df.pickup_datetime.dt.month,
          year = df.pickup_datetime.dt.year,
          dayofweek = df.pickup_datetime.dt.dayofweek)
```

```
In [49]: df.head
```

Out[49]:

```
<bound method NDFrame.head of         fare_amount           pickup_datetime  picku
p_longitude  \
0               7.5 2015-05-07 19:52:06+00:00        -73.999817   
1               7.7 2009-07-17 20:04:56+00:00        -73.994355   
2              12.9 2009-08-24 21:45:00+00:00        -74.005043   
3               5.3 2009-06-26 08:22:21+00:00        -73.976124   
4              16.0 2014-08-28 17:47:00+00:00        -73.925023   
...             ...                       ...               ...   
199995          3.0 2012-10-28 10:49:00+00:00        -73.987042   
199996          7.5 2014-03-14 01:09:00+00:00        -73.984722   
199997         30.9 2009-06-29 00:42:00+00:00        -73.986017   
199998         14.5 2015-05-20 14:56:25+00:00        -73.997124   
199999         14.1 2010-05-15 04:08:00+00:00        -73.984395   

        pickup_latitude  dropoff_longitude  dropoff_latitude  passenger_count  \
0             40.738354         -73.999512         40.723217                1   
1             40.728225         -73.994710         40.750325                1   
2             40.740770         -73.962565         40.772647                1   
3             40.790844         -73.965316         40.803349                3   
4             40.744085         -73.973082         40.761247                5   
...                 ...                ...               ...              ...   
199995        40.739367         -73.986525         40.740297                1   
199996        40.736837         -74.006672         40.739620                1   
199997        40.756487         -73.858957         40.692588                2   
199998        40.725452         -73.983215         40.695415                1   
199999        40.720077         -73.985508         40.768793                1   

        hour  day  month  year  dayofweek  
0         19    7      5  2015          3  
1         20   17      7  2009          4  
2         21   24      8  2009          0  
3          8   26      6  2009          4  
4         17   28      8  2014          3  
...      ...  ...    ...   ...        ...  
199995    10   28     10  2012          6  
199996     1   14      3  2014          4  
199997     0   29      6  2009          0  
199998    14   20      5  2015          2  
199999     4   15      5  2010          5  

[200000 rows x 12 columns]>
```

In [50]:

```
df = df.drop('pickup_datetime', axis =1)
```

In [51]:

```
df.head
```

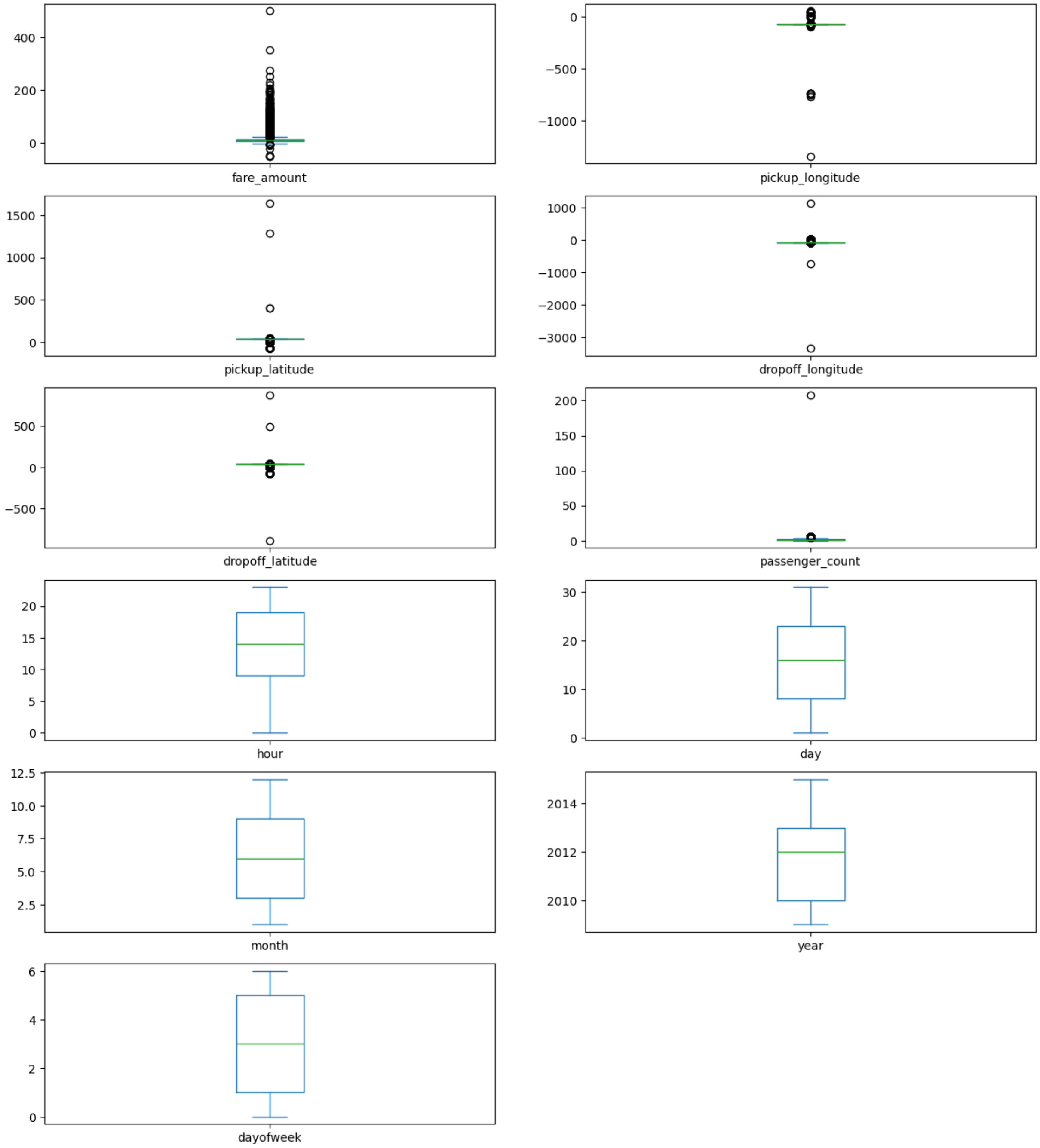
```
Out[51]: <bound method NDFrame.head of         fare_amount  pickup_longitude  pickup_latitu
         de  dropoff_longitude  \
         0               7.5        -73.999817        40.738354         -73.999512
         1               7.7        -73.994355        40.728225         -73.994710
         2              12.9        -74.005043        40.740770         -73.962565
         3               5.3        -73.976124        40.790844         -73.965316
         4              16.0        -73.925023        40.744085         -73.973082
         ...             ...               ...              ...                ...
         199995          3.0        -73.987042        40.739367         -73.986525
         199996          7.5        -73.984722        40.736837         -74.006672
         199997         30.9        -73.986017        40.756487         -73.858957
         199998         14.5        -73.997124        40.725452         -73.983215
         199999         14.1        -73.984395        40.720077         -73.985508

                 dropoff_latitude  passenger_count  hour  day  month  year  dayofweek
         0              40.723217                1    19    7      5  2015          3
         1              40.750325                1    20   17      7  2009          4
         2              40.772647                1    21   24      8  2009          0
         3              40.803349                3     8   26      6  2009          4
         4              40.761247                5    17   28      8  2014          3
         ...                  ...              ...   ...  ...    ...   ...        ...
         199995         40.740297                1    10   28     10  2012          6
         199996         40.739620                1     1   14      3  2014          4
         199997         40.692588                2     0   29      6  2009          0
         199998         40.695415                1    14   20      5  2015          2
         199999         40.768793                1     4   15      5  2010          5

         [200000 rows x 11 columns]>
```

```
In [52]: df.dtypes
```

```
Out[52]: fare_amount          float64
         pickup_longitude     float64
         pickup_latitude      float64
         dropoff_longitude    float64
         dropoff_latitude     float64
         passenger_count        int64
         hour                   int32
         day                    int32
         month                  int32
         year                   int32
         dayofweek              int32
         dtype: object
```

```
In [53]: df.plot(kind="box",  subplots = True,layout =(7,2),figsize=(15,20))
```

```
Out[53]: fare_amount              Axes(0.125,0.786098;0.352273x0.0939024)
         pickup_longitude       Axes(0.547727,0.786098;0.352273x0.0939024)
         pickup_latitude           Axes(0.125,0.673415;0.352273x0.0939024)
         dropoff_longitude      Axes(0.547727,0.673415;0.352273x0.0939024)
         dropoff_latitude          Axes(0.125,0.560732;0.352273x0.0939024)
         passenger_count        Axes(0.547727,0.560732;0.352273x0.0939024)
         hour                     Axes(0.125,0.448049;0.352273x0.0939024)
         day                    Axes(0.547727,0.448049;0.352273x0.0939024)
         month                    Axes(0.125,0.335366;0.352273x0.0939024)
         year                   Axes(0.547727,0.335366;0.352273x0.0939024)
         dayofweek                Axes(0.125,0.222683;0.352273x0.0939024)
         dtype: object
```



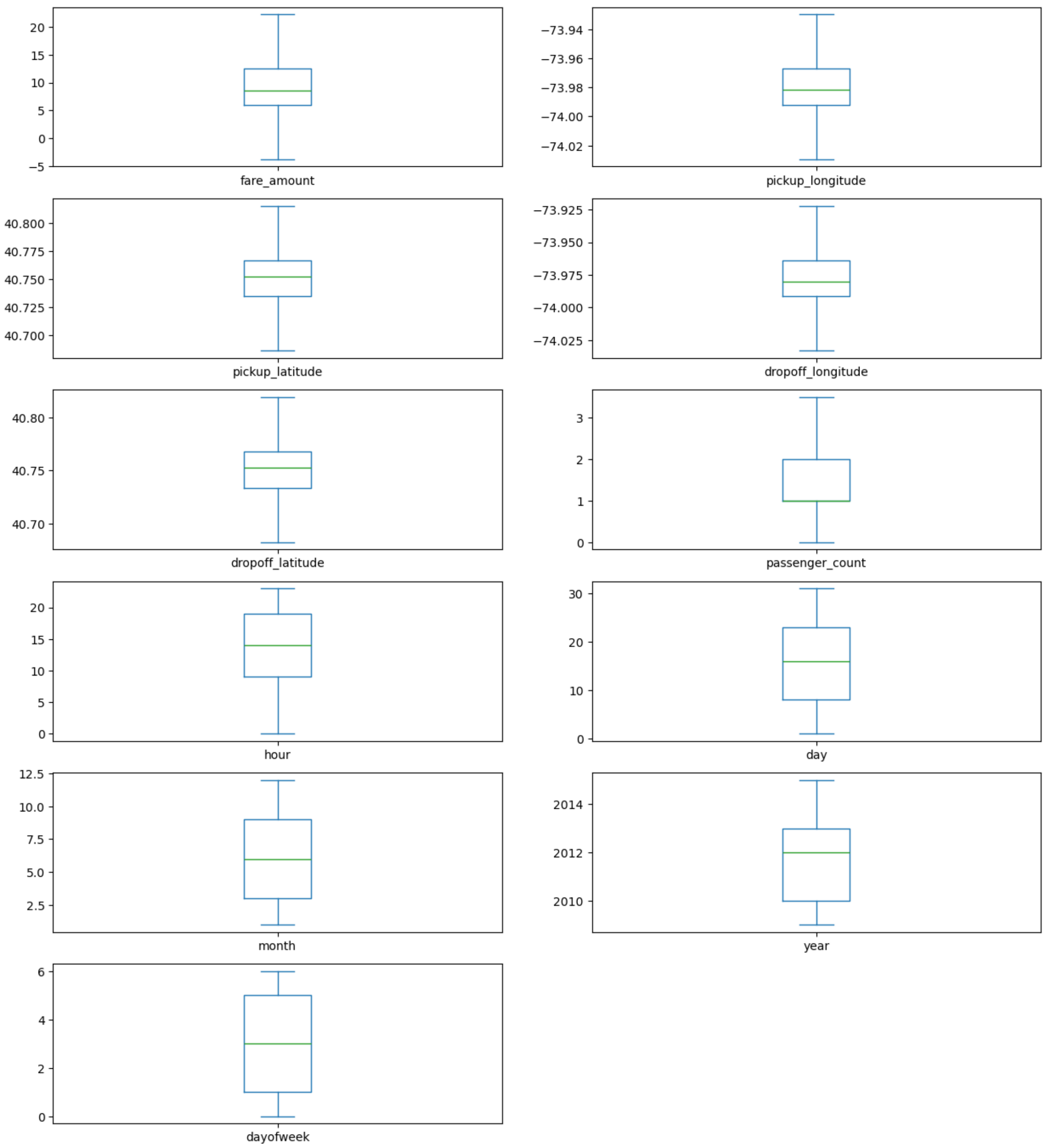
```
In [62]: def remove_outlier(df1, col):
          Q1 =df1[col].quantile(0.25)
          Q3= df1[col].quantile (0.75)
```

```
  IQR= Q3 - Q1
  lower_whisker= Q1-1.5*IQR
  upper_whisker= Q3+1.5*IQR
  df[col] = np.clip(df1[col], lower_whisker, upper_whisker)
  return df1
def treat_outliers_all(df1, col_list):
  for c in col_list:
   df1 = remove_outlier(df, c)
  return df1
```

In [63]:
```
df = treat_outliers_all(df,df.iloc[:, 0::])
```

In [64]:
```
df.plot(kind = "box",subplots = True,layout=(7,2),figsize=(15,20))
```

Out[64]:
```
fare_amount             Axes(0.125,0.786098;0.352273x0.0939024)
pickup_longitude     Axes(0.547727,0.786098;0.352273x0.0939024)
pickup_latitude         Axes(0.125,0.673415;0.352273x0.0939024)
dropoff_longitude    Axes(0.547727,0.673415;0.352273x0.0939024)
dropoff_latitude        Axes(0.125,0.560732;0.352273x0.0939024)
passenger_count      Axes(0.547727,0.560732;0.352273x0.0939024)
hour                    Axes(0.125,0.448049;0.352273x0.0939024)
day                  Axes(0.547727,0.448049;0.352273x0.0939024)
month                   Axes(0.125,0.335366;0.352273x0.0939024)
year                 Axes(0.547727,0.335366;0.352273x0.0939024)
dayofweek               Axes(0.125,0.222683;0.352273x0.0939024)
dtype: object
```



In [65]:

```
pip install haversine
```

```
Collecting haversine
  Downloading haversine-2.9.0-py2.py3-none-any.whl.metadata (5.8 kB)
Downloading haversine-2.9.0-py2.py3-none-any.whl (7.7 kB)
Installing collected packages: haversine
Successfully installed haversine-2.9.0
Note: you may need to restart the kernel to use updated packages.
```

In [72]:

```
import haversine as hs
travel_dist = []

for pos in range(len(df['pickup_longitude'])):
    long1 = df['pickup_longitude'][pos]
    lati1 = df['pickup_latitude'][pos]
    long2 = df['dropoff_longitude'][pos]
    lati2 = df['dropoff_latitude'][pos]  # <-- corrected from 'dropoff_lloc1'

    loc1 = (lati1, long1)
    loc2 = (lati2, long2)

    c = hs.haversine(loc1, loc2)
    travel_dist.append(c)

df['dist_travel_km'] = travel_dist
df.head()
```

Out[72]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	pa
0	7.5	-73.999817	40.738354	-73.999512	40.723217	
1	7.7	-73.994355	40.728225	-73.994710	40.750325	
2	12.9	-74.005043	40.740770	-73.962565	40.772647	
3	5.3	-73.976124	40.790844	-73.965316	40.803349	
4	16.0	-73.929786	40.744085	-73.973082	40.761247	

In [73]:

```
df = df.loc[(df.dist_travel_km>=1) |(df.dist_travel_km<=130) ]
print("Remaining obervation:" , df.shape)
```

```
Remaining obervation: (200000, 12)
```

In [74]:

```
incorrect_coordinates =df.loc[(df.pickup_latitude>90) | (df.pickup_latitude< -90)|
                              (df.dropoff_latitude>90)  |  (df.dropoff_latitude< -90
                              (df.pickup_longitude>180)|  (df.pickup_longitude< -180
                              (df.dropoff_latitude>90) | (df.dropoff_latitude< -90
                              ]
```

In [75]:

```
df.drop(incorrect_coordinates, inplace = True, errors ='ignore')
```

In [76]:

```
df.head()
```

Out[76]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	pa
0	7.5	-73.999817	40.738354	-73.999512	40.723217	
1	7.7	-73.994355	40.728225	-73.994710	40.750325	
2	12.9	-74.005043	40.740770	-73.962565	40.772647	
3	5.3	-73.976124	40.790844	-73.965316	40.803349	
4	16.0	-73.929786	40.744085	-73.973082	40.761247	

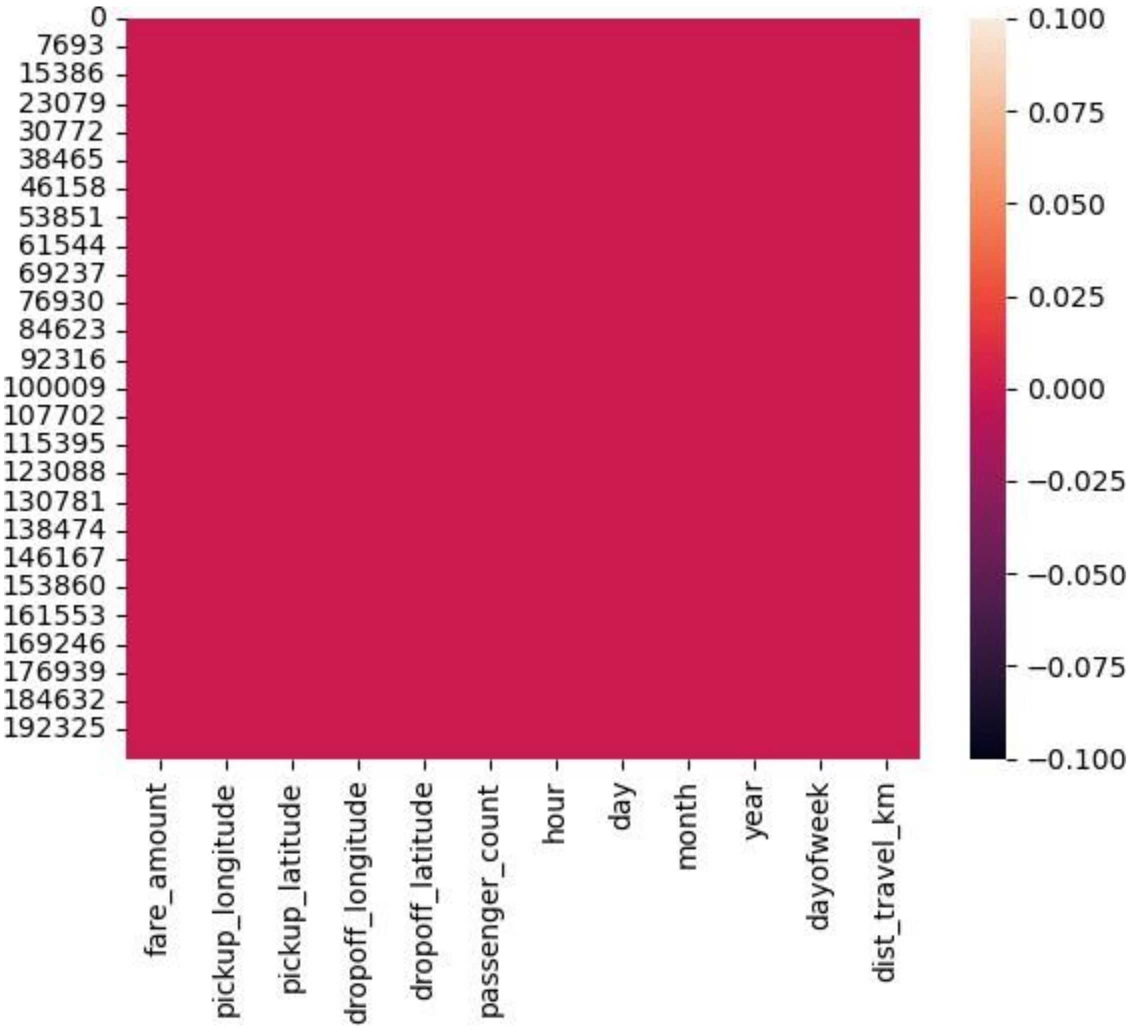
In [77]:

```
df.isnull().sum()
```

```
Out[77]: fare_amount          0
         pickup_longitude     0
         pickup_latitude      0
         dropoff_longitude    0
         dropoff_latitude     0
         passenger_count      0
         hour                 0
         day                  0
         month                0
         year                 0
         dayofweek            0
         dist_travel_km       0
         dtype: int64
```

```
In [78]: sns.heatmap(df.isnull())
```

```
Out[78]: <Axes: >
```



```
In [79]: corr = df.corr()
         corr
```

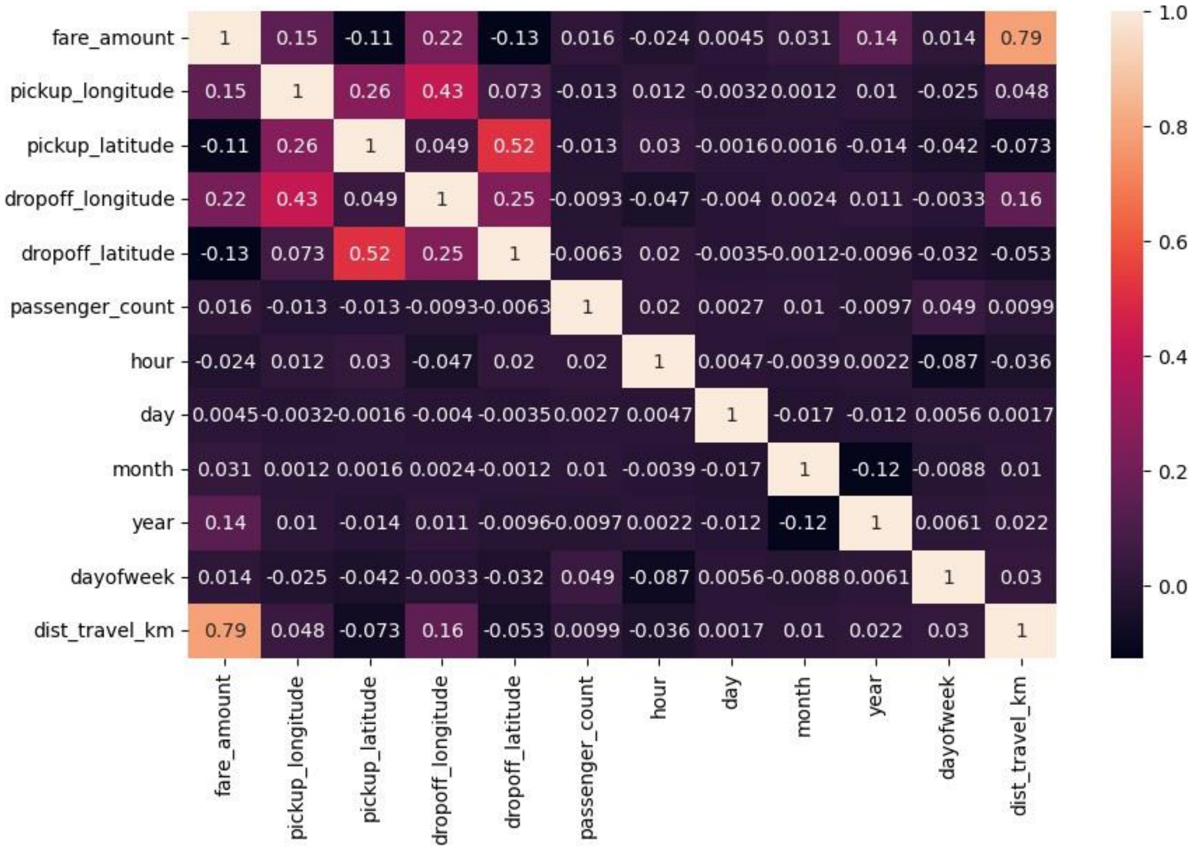
Out[79]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	drop
fare_amount	1.000000	0.154069	-0.110842	0.218675	
pickup_longitude	0.154069	1.000000	0.259497	0.425619	
pickup_latitude	-0.110842	0.259497	1.000000	0.048889	
dropoff_longitude	0.218675	0.425619	0.048889	1.000000	
dropoff_latitude	-0.125898	0.073290	0.515714	0.245667	
passenger_count	0.015778	-0.013213	-0.012889	-0.009303	
hour	-0.023623	0.011579	0.029681	-0.046558	
day	0.004534	-0.003204	-0.001553	-0.004007	
month	0.030817	0.001169	0.001562	0.002391	
year	0.141277	0.010198	-0.014243	0.011346	
dayofweek	0.013652	-0.024652	-0.042310	-0.003336	
dist_travel_km	0.786385	0.048446	-0.073362	0.155191	

In [80]:

```
fig,axis = plt.subplots(figsize= (10,6))
sns.heatmap(df.corr(), annot = True)
```

Out[80]: <Axes: >



```
In [81]: df.dtypes
```

```
Out[81]: fare_amount            float64
         pickup_longitude       float64
         pickup_latitude        float64
         dropoff_longitude      float64
         dropoff_latitude       float64
         passenger_count        float64
         hour                     int32
         day                      int32
         month                    int32
         year                     int32
         dayofweek                int32
         dist_travel_km         float64
         dtype: object
```

```
In [84]: x = df[['pickup_longitude', 'pickup_latitude','dropoff_longitude', 'passenger_count
```

```
In [85]: y = df['fare_amount']
```

```
In [87]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train,y_test = train_test_split(x,y,test_size = 0.33)
```

```
In [89]: from sklearn.linear_model import LinearRegression
         regression = LinearRegression()
```

```
In [90]: regression.fit(x_train,y_train)
```

```
Out[90]: ▾ LinearReg ession

         LinearRegression()
```

```
In [91]: regression.intercept_
```

```
Out[91]: 3256.513201301484
```

```
In [92]: regression.coef_
```

```
Out[92]: array([ 3.10758620e+01, -1.85910612e+01,  1.27429851e+01,  5.41800674e-02,
                 5.77478080e-03,  3.38249411e-03,  5.89651957e-02,  3.71366494e-01,
                -3.27278737e-02,  1.86084919e+00])
```

```
In [93]: prediction = regression.predict(x_test)
```

```
In [94]: print(prediction)
```

```
[13.00853888  7.57316671 13.25903141 ... 13.71316448 13.18778424
 11.11547073]
```

```
In [95]: y_test
```

Out[95]:

```
1086      22.25
96074     11.30
104027    12.50
133659     7.50
58255      7.70
          ...  
27046     10.50
178816    14.00
42112     12.50
53060     14.50
83308      8.90
Name: fare_amount, Length: 66000, dtype: float64
```

In [97]:

```
from sklearn.metrics import r2_score
r2_score(y_test,prediction)
```

Out[97]:

```
0.6584561379465177
```

In [100…

```
from sklearn.metrics import mean_squared_error
MSE = mean_squared_error(y_test,prediction)
MSE
```

Out[100…

```
10.140972773285915
```

In [ ]: