In [1]:
```
import pandas as pd

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

In [2]:
```
df.sample(5)
```

Out[2]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitud
162949	14564933	2012-02-02 18:54:15.0000005	5.3	2012-02-02 18:54:15 UTC	-73.96012
140856	19739521	2013-11-09 12:37:00.000000152	12.5	2013-11-09 12:37:00 UTC	-73.97896
18415	11328919	2012-07-29 08:28:11.0000002	190.0	2012-07-29 08:28:11 UTC	-73.79723
86014	11414939	2010-05-02 05:54:35.0000001	45.0	2010-05-02 05:54:35 UTC	-73.78766
190449	27635493	2009-04-08 19:13:00.000000244	8.9	2009-04-08 19:13:00 UTC	-74.00467

In [3]:
```
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 [4]:
```
df=df.drop(['Unnamed: 0','key'],axis=1)
```

In [5]:
```
df
```

Out[5]:

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_lon
0	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-73.9
1	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-73.9
2	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-73.9
3	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-73.9
4	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-73.9
...	...	...	...	...	
199995	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-73.9
199996	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-74.0
199997	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-73.8
199998	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-73.9
199999	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-73.9

200000 rows × 7 columns

In [6]:
```
df['pickup_datetime']=pd.to_datetime(df['pickup_datetime'])
```

In [7]:
```
df['hour']=df['pickup_datetime'].dt.hour
df['day']=df['pickup_datetime'].dt.day
df['month']=df['pickup_datetime'].dt.month
df['year']=df['pickup_datetime'].dt.year

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

In [8]:
```
df
```

Out[8]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_la
0	7.5	-73.999817	40.738354	-73.999512	40.
1	7.7	-73.994355	40.728225	-73.994710	40.
2	12.9	-74.005043	40.740770	-73.962565	40.
3	5.3	-73.976124	40.790844	-73.965316	40.
4	16.0	-73.925023	40.744085	-73.973082	40.
...	...	...	...	...	
199995	3.0	-73.987042	40.739367	-73.986525	40.
199996	7.5	-73.984722	40.736837	-74.006672	40.
199997	30.9	-73.986017	40.756487	-73.858957	40.
199998	14.5	-73.997124	40.725452	-73.983215	40.
199999	14.1	-73.984395	40.720077	-73.985508	40.

200000 rows × 10 columns

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
fare_amount          0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
hour                 0
day                  0
month                0
year                 0
dtype: int64
```

In [10]:
```
df=df.dropna()
```

In [11]:
```
df=df[(df['fare_amount']>0) & (df['passenger_count']>0)]
```

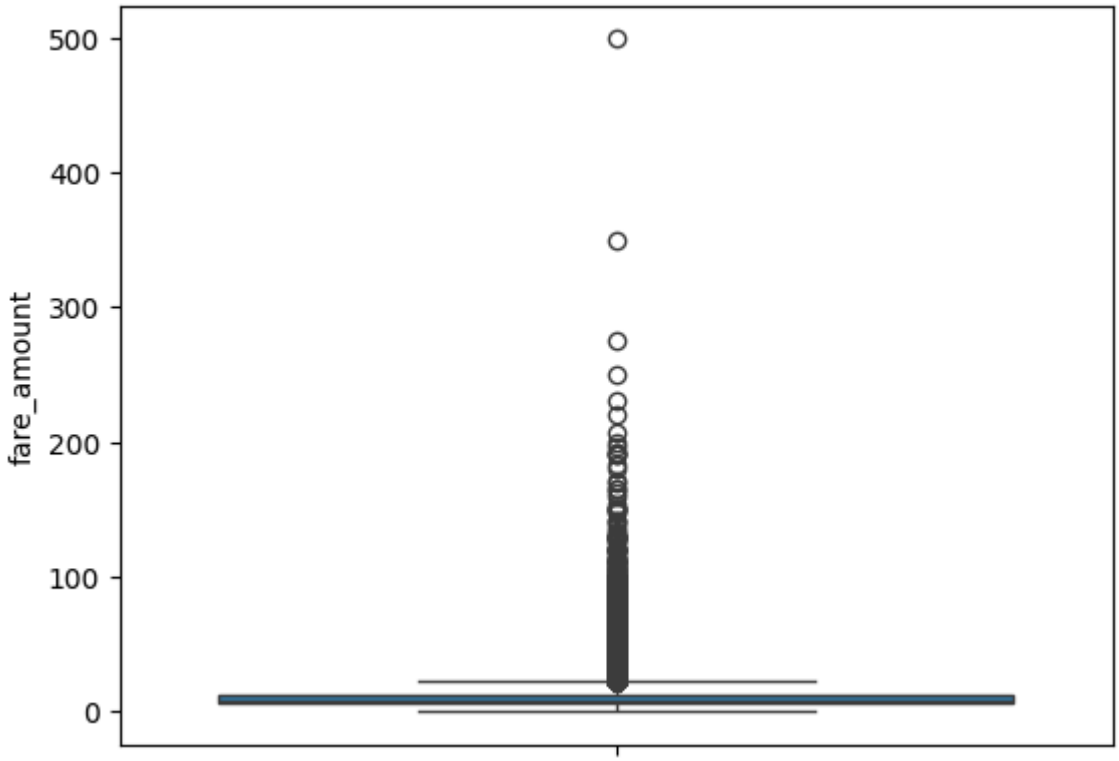
In [12]:
```
df.shape
```

Out[12]:
```
(199269, 10)
```

In [13]:
```
import seaborn as sns

sns.boxplot(df['fare_amount'])
```

Out[13]:
```
<Axes: ylabel='fare_amount'>
```



In [14]:
```
Q1=df['fare_amount'].quantile(0.25)
Q3=df['fare_amount'].quantile(0.75)

IQR=Q3-Q1

lower_bound=Q1-1.5*IQR
upper_bound=Q3+1.5*IQR

df=df[(df['fare_amount']>lower_bound) & (df['fare_amount']<upper_bound)]
```

In [15]:
```
df.shape
```

Out[15]: (182148, 10)

In [16]:
```
corr=df.corr()
```

In [17]:
```
corr
```

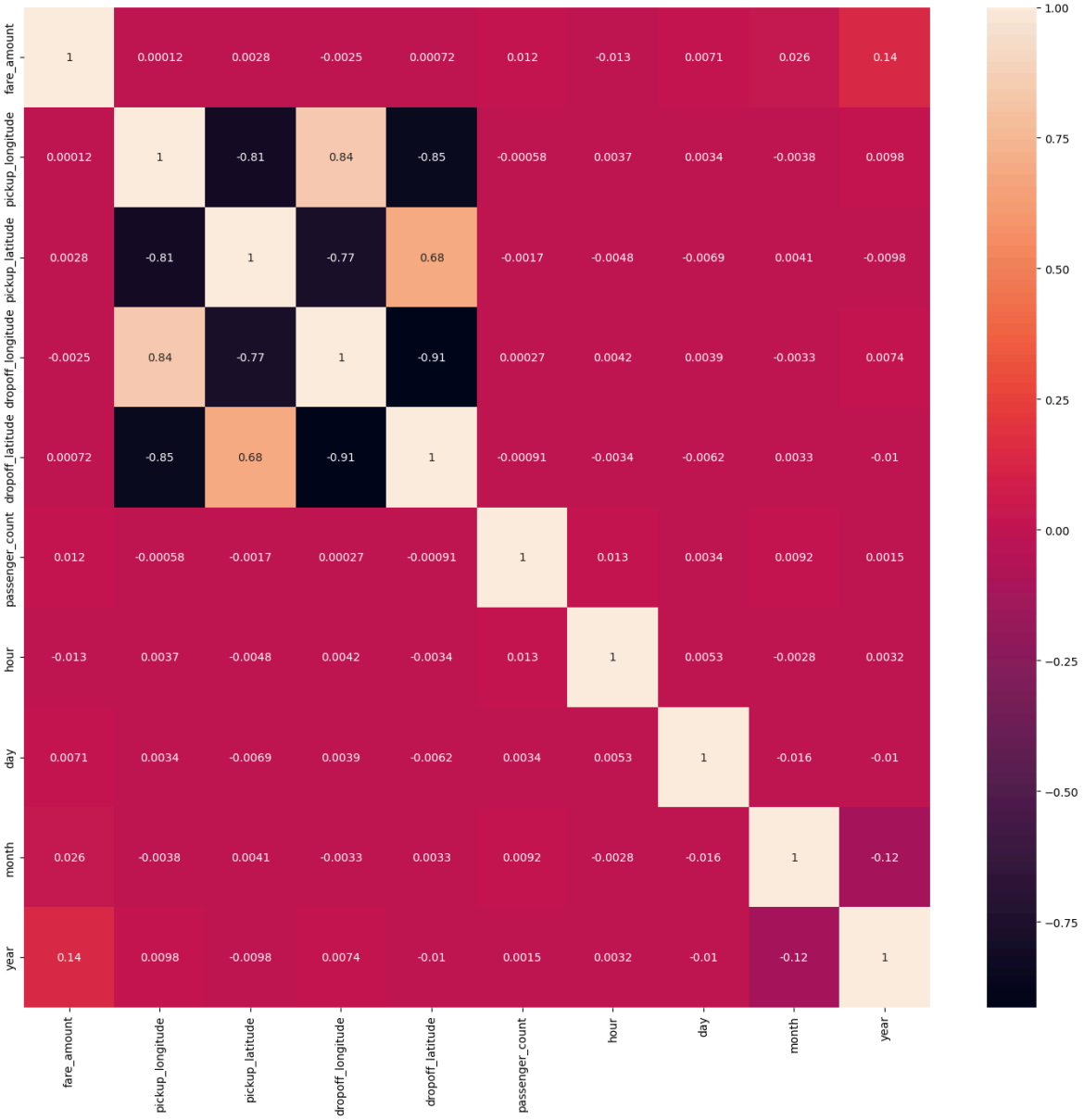
Out[17]:

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude
fare_amount	1.000000	0.000125	0.002826	-0.002534
pickup_longitude	0.000125	1.000000	-0.811435	0.835878
pickup_latitude	0.002826	-0.811435	1.000000	-0.766797
dropoff_longitude	-0.002534	0.835878	-0.766797	1.000000
dropoff_latitude	0.000715	-0.850520	0.683972	-0.913666
passenger_count	0.011995	-0.000582	-0.001742	0.000271
hour	-0.013304	0.003668	-0.004779	0.004220
day	0.007128	0.003436	-0.006933	0.003941
month	0.026188	-0.003842	0.004069	-0.003292
year	0.135630	0.009803	-0.009843	0.007351

In [18]:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(18,16))
sns.heatmap(corr,annot=True)
plt.show()
```



In [19]:
```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, root_mean_squared_error
```

In [20]:
```
lr=LinearRegression()
rig=Ridge()
lass=Lasso()
```

In [21]:
```
x=df.drop(['passenger_count'],axis=1)
y=df['passenger_count']
```

In [22]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=32,test_s
```

In [23]:
```
lr.fit(x_train,y_train)
rig.fit(x_train,y_train)
lass.fit(x_train,y_train)
```

Out[23]:

▾ Lasso

Lasso()

In [24]:
```
y_pred=lr.predict(x_test)
print(r2_score(y_test,y_pred))
print(root_mean_squared_error(y_test,y_pred))
```

```
0.0006000637154700561
1.3013869504918585
```

In [25]:
```
y_pred=rig.predict(x_test)
print(r2_score(y_test,y_pred))
print(root_mean_squared_error(y_test,y_pred))
```

```
0.0006000643961555641
1.301386950048675
```

In [26]:
```
y_pred=lass.predict(x_test)
print(r2_score(y_test,y_pred))
print(root_mean_squared_error(y_test,y_pred))
```

```
-5.340509565687768e-06
1.3017810599218966
```

In [ ]: