In [31]:

import pandas as pd import numpy as np

In [32]:

df= pd.read\_csv("/content/uber.csv")

In [33]:

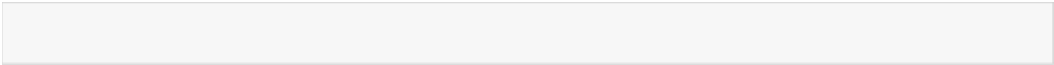
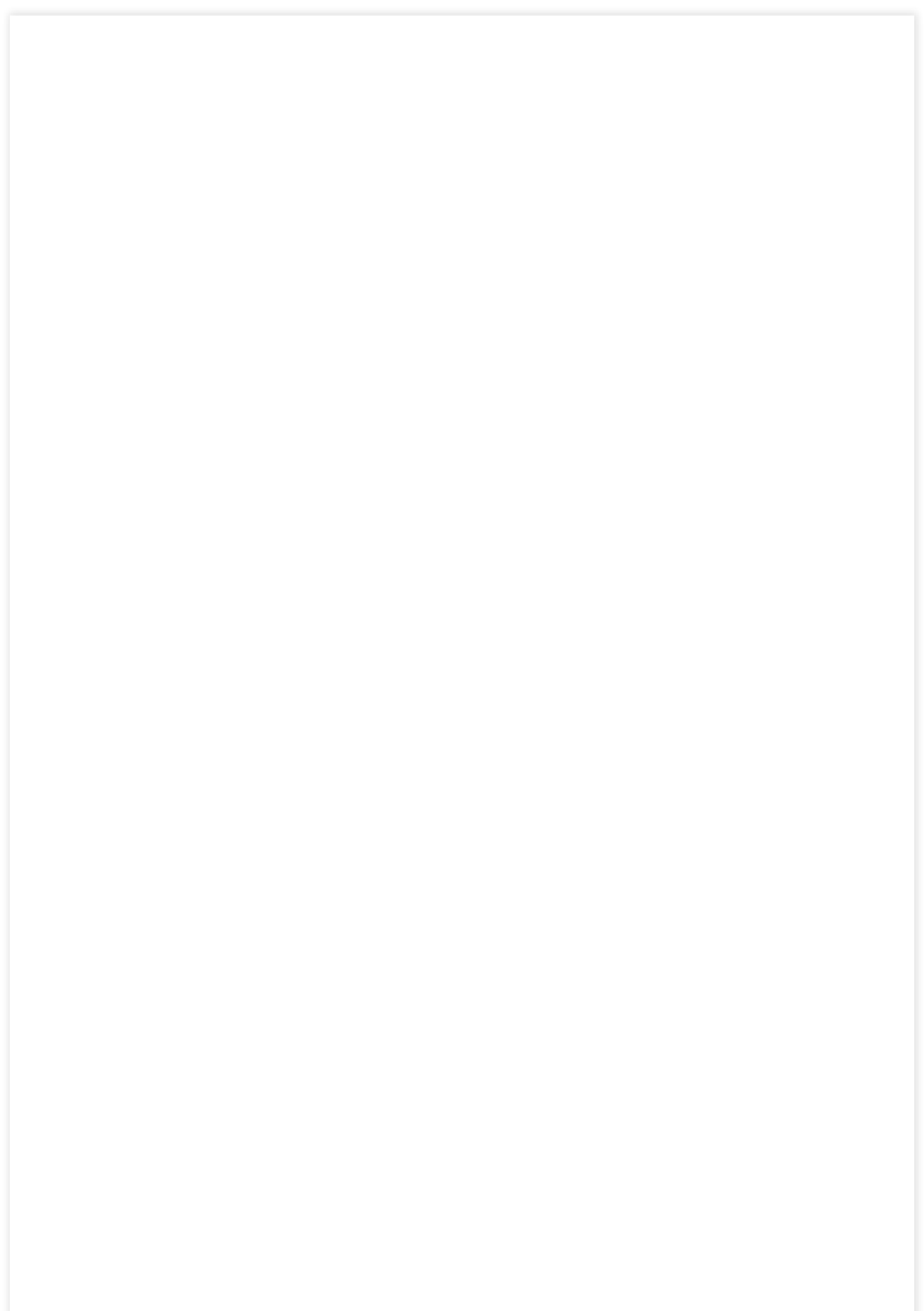
df.head()

Out[33]:

**Unnamed:**

**0**

**key fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude dropoff\_longitude dro**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **0** 24238194 | 2015-05-07  19:52:06.0000003 | 7.5 | 2015-05-07  19:52:06 UTC | -73.999817 | 40.738354 | -73.999512 |
| **1** 27835199 | 2009-07-17  20:04:56.0000002 | 7.7 | 2009-07-17  20:04:56 UTC | -73.994355 | 40.728225 | -73.994710 |
| **2** 44984355 | 2009-08-24  21:45:00.00000061 | 12.9 | 2009-08-24  21:45:00 UTC | -74.005043 | 40.740770 | -73.962565 |
| **3** 25894730 | 2009-06-26  08:22:21.0000001 | 5.3 | 2009-06-26  08:22:21 UTC | -73.976124 | 40.790844 | -73.965316 |
| **4** 17610152 | 2014-08-28  17:47:00.000000188 | 16.0 | 2014-08-28  17:47:00 UTC | -73.925023 | 40.744085 | -73.973082 |

In [34]:

|  |  |
| --- | --- |
| df.isnull().sum() |  |
| Out[34]: |
| Unnamed: 0 | 0 |
| key | 0 |
| fare\_amount | 0 |
| pickup\_datetime | 1 |
| pickup\_longitude | 1 |
| pickup\_latitude | 1 |
| dropoff\_longitude | 2 |
| dropoff\_latitude | 2 |
| passenger\_count | 1 |
| dtype: int64 |  |

In [35]:

df.dropna(inplace=True)

In [36]:

df.drop(labels='Unnamed: 0',axis=1,inplace=True)

In [37]:

df.drop(labels='key',axis=1,inplace=True)

In [38]:

df.head()

Out[38]:

**fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude dropoff\_longitude dropoff\_latitude passenger\_count**

**0**

7.5

2015-05-07

19:52:06 UTC

-73.999817

40.738354

-73.999512

40.723217

1.0

**1** 7.7 2009-07-17

20:04:56 UTC

**2**

12.9

2009-08-24

21:45:00 UTC

-74.005043

40.740770

-73.962565

40.772647

1.0

**3** 5.3 2009-06-26

08:22:21 UTC

**4**

16.0

2014-08-28

17:47:00 UTC

-73.925023

40.744085

-73.973082

40.761247

5.0

In [39]:



df["pickup\_datetime"]=pd.to\_datetime(df['pickup\_datetime'])

In [40]:



df.head()

Out[40]:

-73.994355 40.728225 -73.994710 40.750325 1.0

-73.976124 40.790844 -73.965316 40.803349 3.0

**fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude dropoff\_longitude dropoff\_latitude passenger\_count**

**0**

7.5

2015-05-07

19:52:06+00:00

-73.999817

40.738354

-73.999512

40.723217

1.0

**1** 7.7 2009-07-17

20:04:56+00:00

**2**

12.9

2009-08-24

21:45:00+00:00

-74.005043

40.740770

-73.962565

40.772647

1.0

**3** 5.3 2009-06-26

08:22:21+00:00

**4**

16.0

2014-08-28

17:47:00+00:00

-73.925023

40.744085

-73.973082

40.761247

5.0

In [41]:



df.describe()

Out[41]:

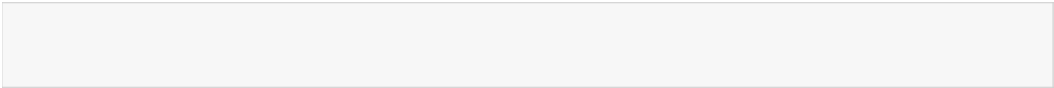
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **fare\_amount** | **pickup\_longitude** | **pickup\_latitude** | **dropoff\_longitude** | **dropoff\_latitude** | **passenger\_count** |
| **count** | 169841.000000 | 169841.000000 | 169841.000000 | 169841.000000 | 169841.000000 | 169841.000000 |
| **mean** | 11.359839 | -72.518528 | 39.932778 | -72.516686 | 39.916603 | 1.684158 |
| **std** | 9.820235 | 11.527704 | 7.999118 | 13.574784 | 6.945321 | 1.398148 |
| **min** | -52.000000 | -1340.648410 | -74.015515 | -3356.666300 | -881.985513 | 0.000000 |
| **25%** | 6.000000 | -73.992065 | 40.734840 | -73.991397 | 40.733822 | 1.000000 |
| **50%** | 8.500000 | -73.981812 | 40.752625 | -73.980080 | 40.753020 | 1.000000 |
| **75%** | 12.500000 | -73.967094 | 40.767182 | -73.963623 | 40.768037 | 2.000000 |
| **max** | 350.000000 | 57.418457 | 1644.421482 | 1153.572603 | 872.697628 | 208.000000 |

In [42]:

-73.994355 40.728225 -73.994710 40.750325 1.0

-73.976124 40.790844 -73.965316 40.803349 3.0





import matplotlib.pyplot as plt import seaborn as sns

%matplotlib inline

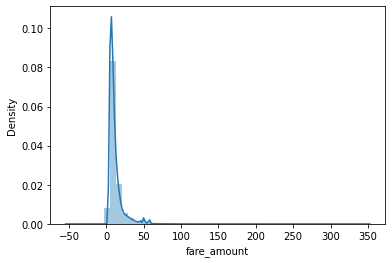
In [43]:



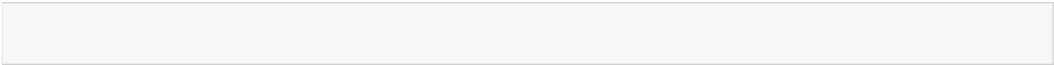
sns.distplot(df['fare\_amount']);

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: ` distplot` is a deprecated function and will be removed in a future version. Please ada pt your code to use either `displot` (a figure-level function with similar flexibility

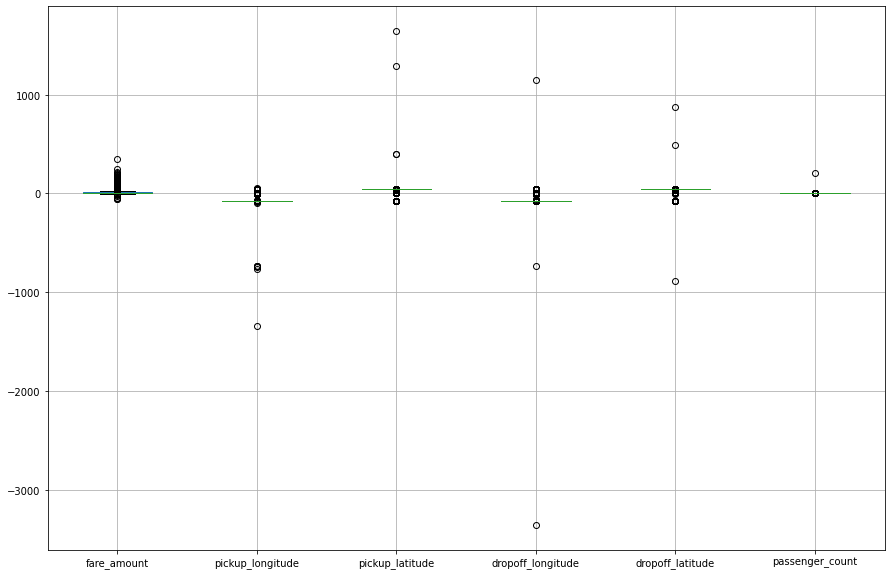
) or `histplot` (an axes-level function for histograms). warnings.warn(msg, FutureWarning)



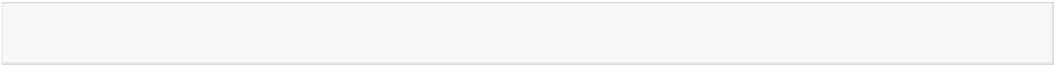
In [44]:



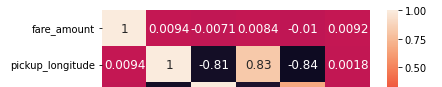
plt.subplots(figsize=(15,10)) df.boxplot();

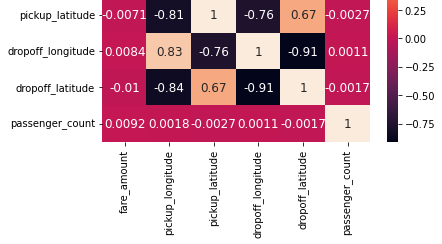


In [45]:



correlation\_matrix = df.corr().round(2) sns.heatmap(df.corr(),annot=True, annot\_kws={'size': 12});





In [45]:

In [45]:

In [45]:

In [46]:



df.drop(["pickup\_datetime"], axis=1, inplace=True)

In [47]:



from sklearn.model\_selection import train\_test\_split

In [48]:



x=df.drop("fare\_amount", axis=1)

In [49]:



y=df['fare\_amount']

In [50]:



x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=101)

In [51]:



x\_train.head()

Out[51]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **pickup\_longitude** | **pickup\_latitude** | **dropoff\_longitude** | **dropoff\_latitude** | **passenger\_count** |
| **118512** | -73.973255 | 40.785025 | -73.982225 | 40.769455 | 1.0 |
| **28135** | -73.862902 | 40.769310 | -73.804105 | 40.762363 | 5.0 |
| **95279** | -73.987110 | 40.739508 | -73.982597 | 40.757473 | 1.0 |
| **19825** | -73.979821 | 40.739303 | -73.994062 | 40.732078 | 1.0 |
| **169707** | -73.984305 | 40.695783 | -73.985131 | 40.713346 | 1.0 |

In [52]:



from sklearn.linear\_model import LinearRegression lm = LinearRegression()





lm.fit(x\_train,y\_train)

Out[52]:

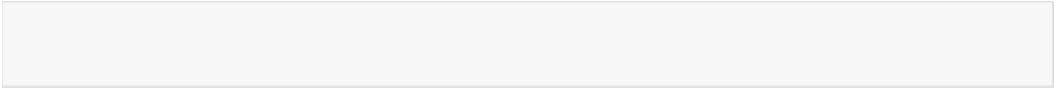
LinearRegression()

In [53]:

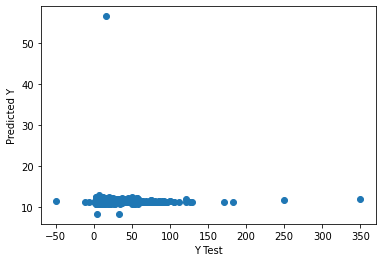


ypred = lm.predict(x\_test)

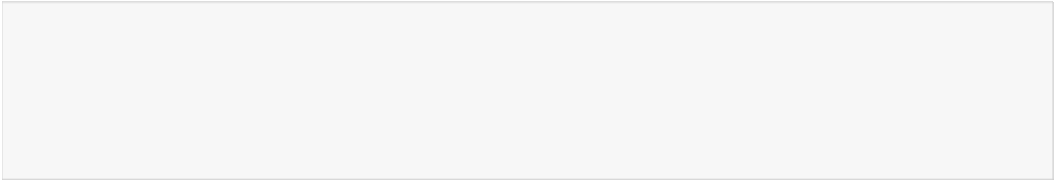
In [54]:



plt.scatter(y\_test,ypred) plt.xlabel('Y Test') plt.ylabel('Predicted Y');



In [72]:



from sklearn import metrics

from sklearn.metrics import r2\_score

print('MAE:', metrics.mean\_absolute\_error(y\_test, ypred)) print('MSE:', metrics.mean\_squared\_error(y\_test, ypred)) print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, ypred))) r2 = r2\_score(y\_test, ypred)

print('r2 score:', r2)

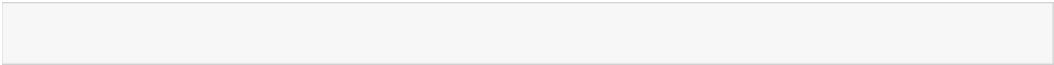
MAE: 6.046505617639961

MSE: 98.61243256389584

RMSE: 9.930379275933817

r2 score: -0.00021133230598224806

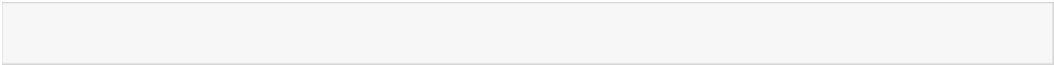
In [56]:



from sklearn.ensemble import RandomForestRegressor

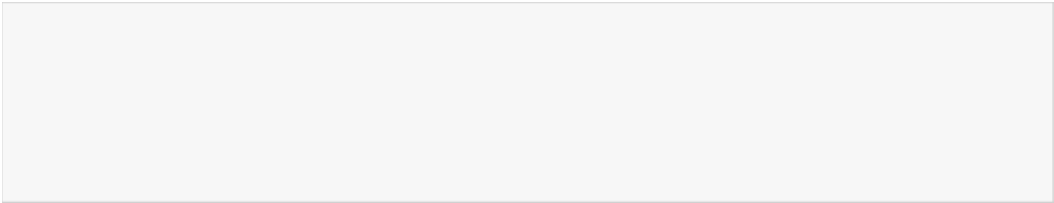
rfrmodel = RandomForestRegressor(n\_estimators=100, random\_state=101)

In [67]:



rfrmodel.fit(x\_train,y\_train) rfrmodel\_pred= rfrmodel.predict(x\_test)

In [73]:



from sklearn.metrics import mean\_squared\_error from sklearn.metrics import r2\_score

print('MAE:', metrics.mean\_absolute\_error(y\_test, rfrmodel\_pred)) rfrmodel\_rmse=np.sqrt(mean\_squared\_error(rfrmodel\_pred, y\_test)) print("RMSE:",rfrmodel\_rmse)

print('MSE:', metrics.mean\_squared\_error(y\_test, rfrmodel\_pred)) r2 = r2\_score(y\_test,rfrmodel\_pred)

print('r2 score:', r2)

MAE: 2.2716559315710136

RMSE: 5.122660388682641



MSE: 26.241649457778195

r2 score: 0.7338348270735129

In [30]: