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	drop
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	-73.994355	40.728225	-73.994710	40.750325	1.0
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	-73.976124	40.790844	-73.965316	40.803349	3.0
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]:

	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	-73.994355	40.728225	-73.994710	40.750325	1.0
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	-73.976124	40.790844	-73.965316	40.803349	3.0
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]:

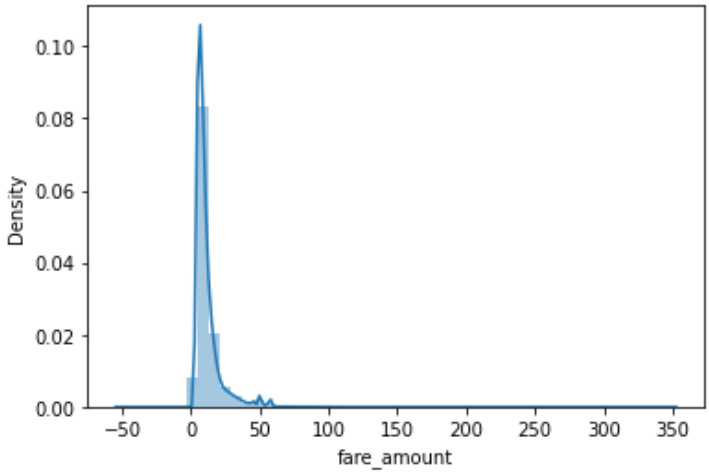
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [43]:

In [43]:

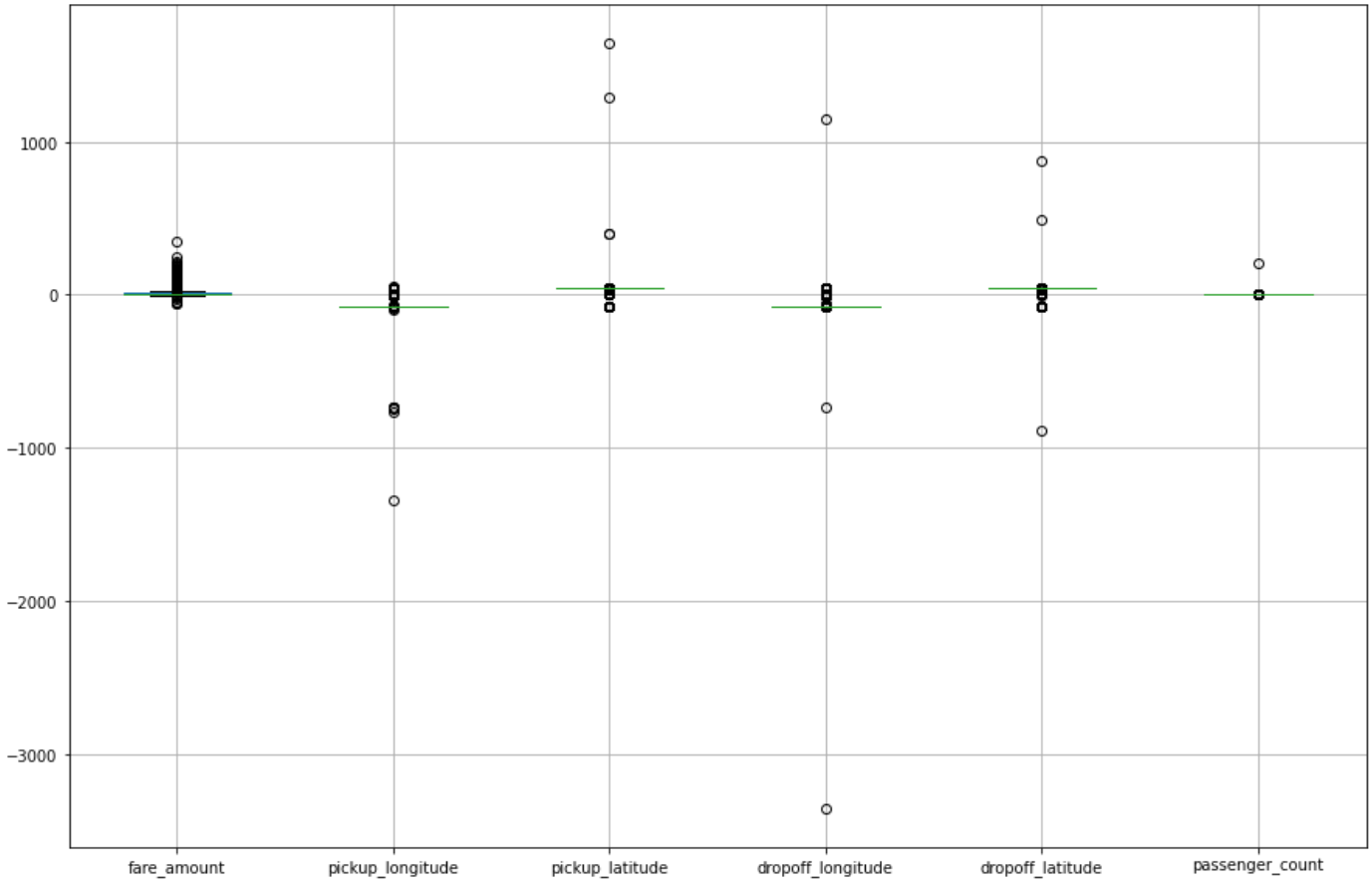
```
sns.distplot(df['fare_amount']);
```

```
/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 adapt 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)
```



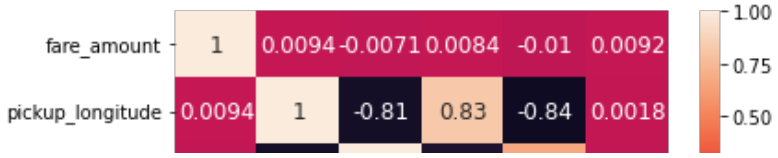
In [44]:

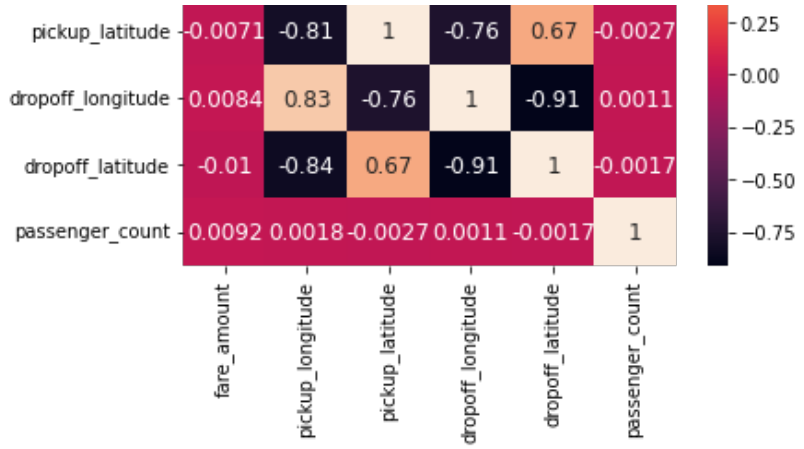
```
plt.subplots(figsize=(15,10))
df.boxplot();
```



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]:

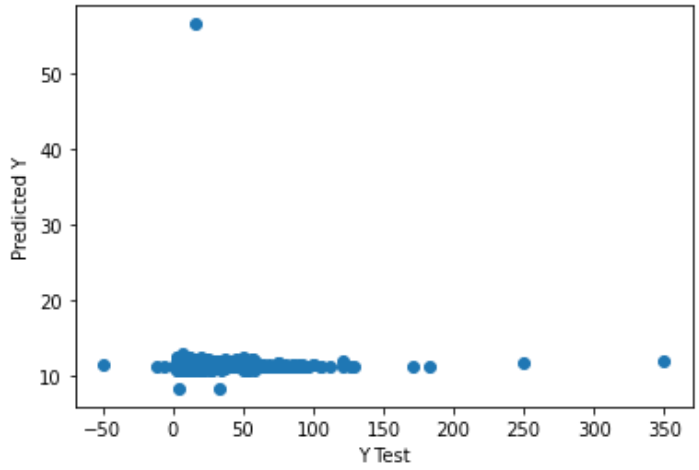
```
LinearRegression()
```

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]:

```
```