Uber Data Analysis w/ Graphics

In [13]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
#Import Dataset and inspect the first few lines
```

In [15]:
```
data = pd.read_csv("uber-raw-data-sep14.csv")
data["Date/Time"] = data["Date/Time"].map(pd.to_datetime)
data.head()
```

Out[15]:

	Date/Time	Lat	Lon	Base
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512

In [ ]:
```
#Expand Date to express day, week, and hour
```

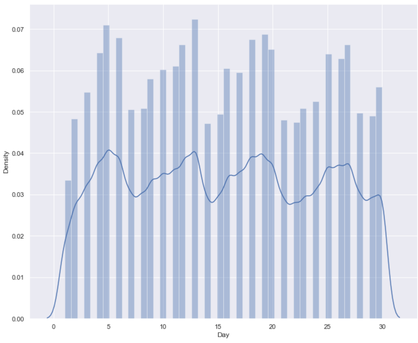
In [16]:
```
data["Day"] = data["Date/Time"].apply(lambda x: x.day)
data["Weekday"] = data["Date/Time"].apply(lambda x: x.weekday())
data["Hour"] = data["Date/Time"].apply(lambda x: x.hour)
print(data.head())
```

```
            Date/Time      Lat      Lon    Base  Day  Weekday  Hour
0 2014-09-01 00:01:00  40.2201 -74.0021  B02512    1        0     0
1 2014-09-01 00:01:00  40.7500 -74.0027  B02512    1        0     0
2 2014-09-01 00:03:00  40.7559 -73.9864  B02512    1        0     0
3 2014-09-01 00:06:00  40.7450 -73.9889  B02512    1        0     0
4 2014-09-01 00:11:00  40.8145 -73.9444  B02512    1        0     0
```

In [ ]:
```
#first graph describes distribution of Uber rides per day, if this data was a live connection the fluctuation would continue,
#but since this data is limited to only a month you will see a positive slope at day 0 and a negative slope at day 30
```

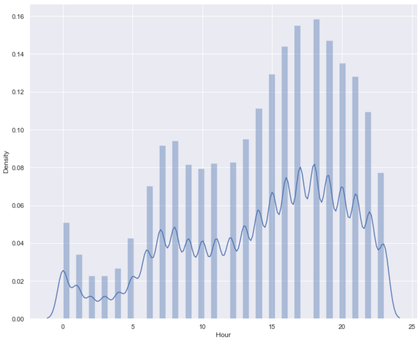
In [17]:
```
sns.set(rc={'figure.figsize':(12, 10)})
sns.distplot(data["Day"])
```

Out[17]:
```
<AxesSubplot:xlabel='Day', ylabel='Density'>
```



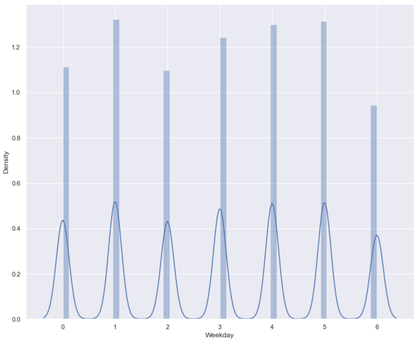
In [18]:
```
sns.distplot(data["Hour"])
```

Out[18]:
```
<AxesSubplot:xlabel='Hour', ylabel='Density'>
```



In [19]:
```
sns.distplot(data["Weekday"])
```

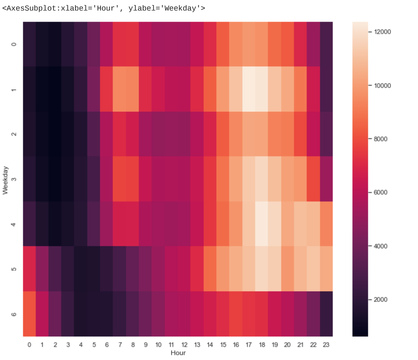
Out[19]:
```
<AxesSubplot:xlabel='Weekday', ylabel='Density'>
```



In [ ]:
```
#this heat map defines which days through out the week are the most busy.
```

In [20]:
```
df = data.groupby(["Weekday", "Hour"]).apply(lambda x: len(x))
df = df.unstack()
sns.heatmap(df, annot = False)
```

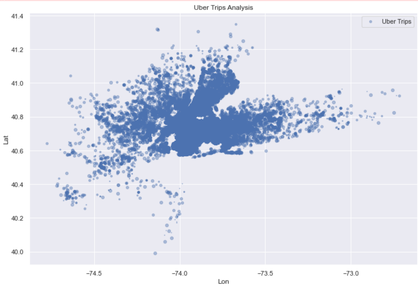
Out[20]:
```
<AxesSubplot:xlabel='Hour', ylabel='Weekday'>
```



In [ ]:
```
# With this scatter plot we can see that all data points are abundant around corrdinates -74.0 long and 40.8 lat,
# these corrdinates navigate to New York City
```

In [22]:
```
data.plot(kind='scatter', x='Lon', y='Lat', alpha=0.4, s=data['Day'], label='Uber Trips',
          figsize=(12, 8), cmap=plt.get_cmap('jet'))
plt.title("Uber Trips Analysis")
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with a single row if you intend to specify the same RGB or RGBA value for all point s.
```



In [ ]:
```
```