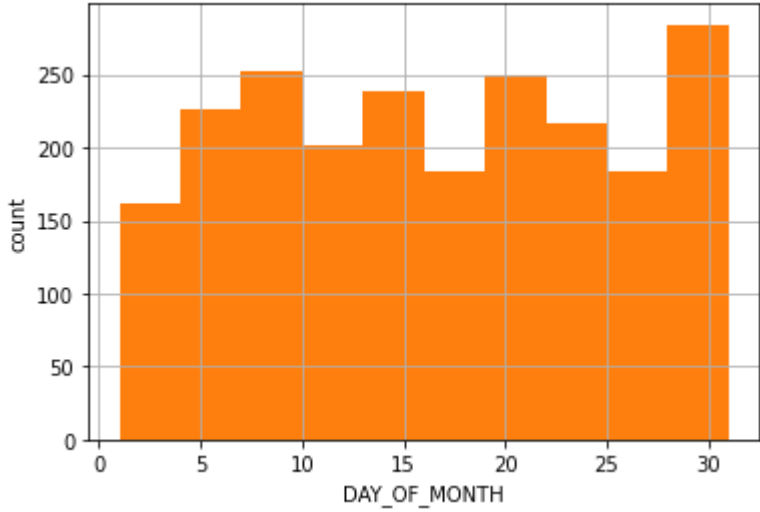


```
In [30]: import pandas as pd
flightdata=pd.read_csv("FormattedData.csv")
flightdata.head()
```

Out[30]:

	CARRIER	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN	WEATHER	DAY_WEEK	DAY_OF_MONTH	TAIL_NUM	FLIGHT_STATUS	UPDATED CRS DEP_TIME	UPDATED DEP_TIME	DELAY_IN_MINS
0	DH	JFK	213	2004-01-01	6155	DCA	0	Thursday	1	N405FJ	ontime	16:40	16:40	0
1	DH	LGA	229	2004-01-01	7208	IAD	0	Thursday	1	N695BR	ontime	12:45	12:45	0
2	DH	LGA	229	2004-01-01	7215	IAD	0	Thursday	1	N662BR	ontime	17:15	17:09	-6
3	DH	LGA	229	2004-01-01	7792	IAD	0	Thursday	1	N698BR	ontime	10:39	10:35	-4
4	DH	JFK	228	2004-01-01	7800	IAD	0	Thursday	1	N687BR	ontime	8:40	8:39	-1

```
In [16]: #For DAY_OF_MONTH- Histograms
DAY_OF_MONTH= flightdata.DAY_OF_MONTH.hist()
DAY_OF_MONTH.set_xlabel('DAY_OF_MONTH');
DAY_OF_MONTH.set_ylabel('count');
```

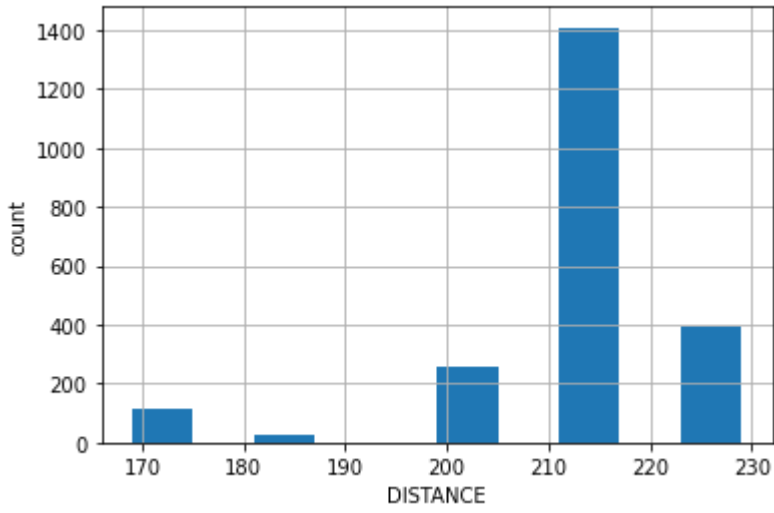


```
In [17]: #For DAY_OF_MONTH - statistics
flightdata['DAY_OF_MONTH'].describe()
```

Out[17]:

```
count      2198.000000
mean        16.027298
std          8.675885
min           1.000000
25%           8.000000
50%          16.000000
75%          23.000000
max          31.000000
Name: DAY_OF_MONTH, dtype: float64
```

```
In [21]: #For DISTANCE- Histograms
DISTANCE= flightdata.DISTANCE.hist()
DISTANCE.set_xlabel('DISTANCE');
DISTANCE.set_ylabel('count');
```

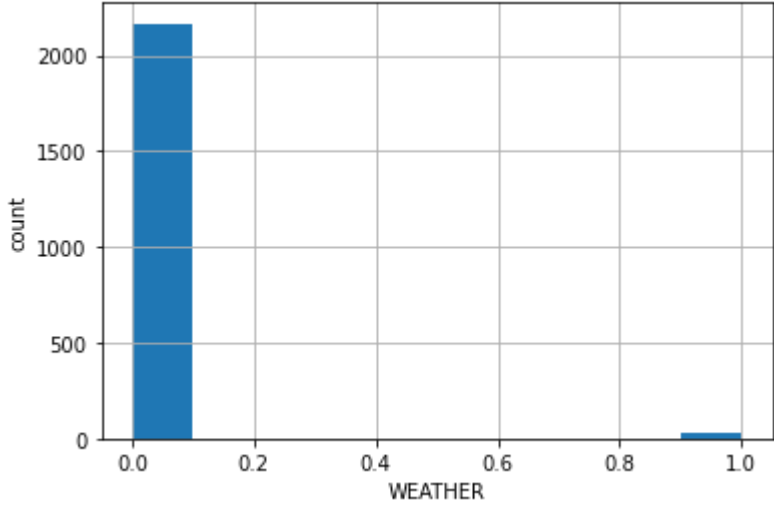


```
In [22]: #For DISTANCE - statistics
flightdata['DISTANCE'].describe()
```

Out[22]:

```
count      2198.000000
mean       211.868517
std        13.302595
min       169.000000
25%       213.000000
50%       214.000000
75%       214.000000
max       229.000000
Name: DISTANCE, dtype: float64
```

```
In [23]: #For WEATHER- Histograms
WEATHER= flightdata.WEATHER.hist()
WEATHER.set_xlabel('WEATHER');
WEATHER.set_ylabel('count');
```

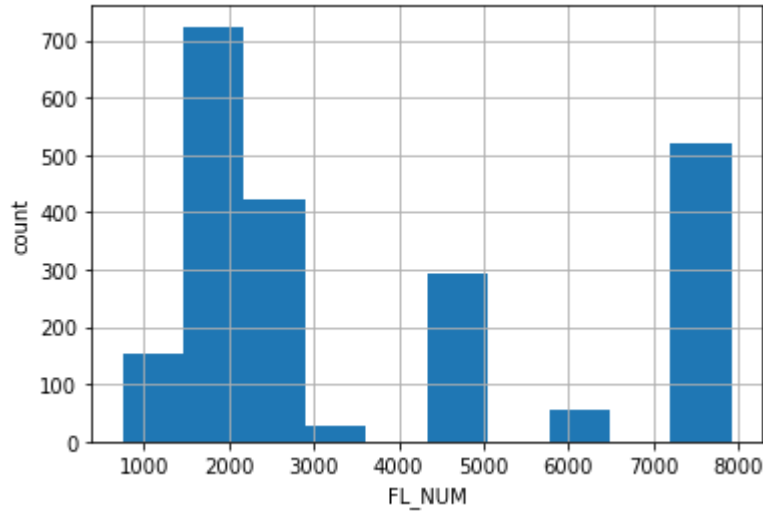


```
In [24]: #For WEATHER - statistics
flightdata['WEATHER'].describe()
```

Out[24]:

```
count      2198.000000
mean        0.014559
std         0.119805
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
Name: WEATHER, dtype: float64
```

```
In [26]: #For FL_NUM- Histograms
FL_NUM= flightdata.FL_NUM.hist()
FL_NUM.set_xlabel('FL_NUM');
FL_NUM.set_ylabel('count');
```

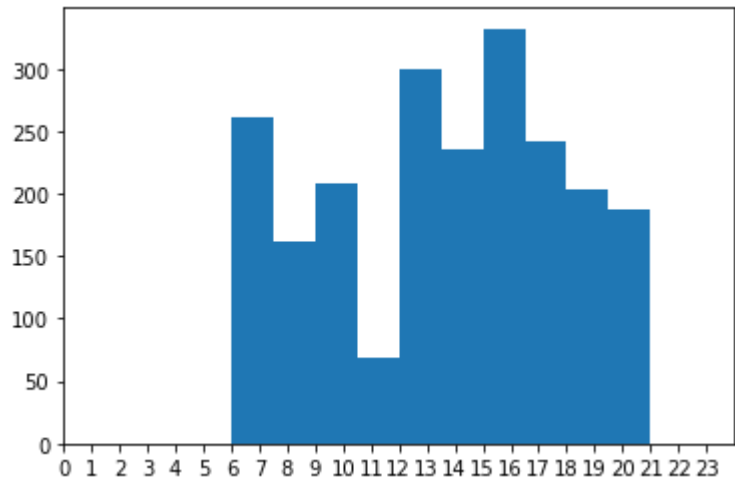


```
In [27]: #For FL_NUM - statistics
flightdata['FL_NUM'].describe()
```

```
Out[27]: count    2198.000000
mean     3810.601456
std      2408.139011
min       746.000000
25%      2156.000000
50%      2385.000000
75%      5935.000000
max      7924.000000
Name: FL_NUM, dtype: float64
```

```
In [63]: #For UPDATED_CRS_DEP_TIME- Histograms- converting in to 24 hour continous format for a better histogram
#Reference - https://sparkbyexamples.com/pandas/pandas-convert-string-column-to-datetime/#:~:text=Use%20pandas%20to_datetime()%20function,string%20you%20wanted%20to%20convert.
```

```
flightdata['UPDATED_CRS_DEP_TIME'] = pd.to_datetime(flightdata['UPDATED_CRS_DEP_TIME'], infer_datetime_format=True)
hour_list = [t.hour for t in flightdata['UPDATED_CRS_DEP_TIME']]
numbers=[x for x in range(0,24)]
labels=map(lambda x: str(x), numbers)
plt.xticks(numbers, labels)
plt.xlim(0,24)
plt.hist(hour_list)
plt.show()
```

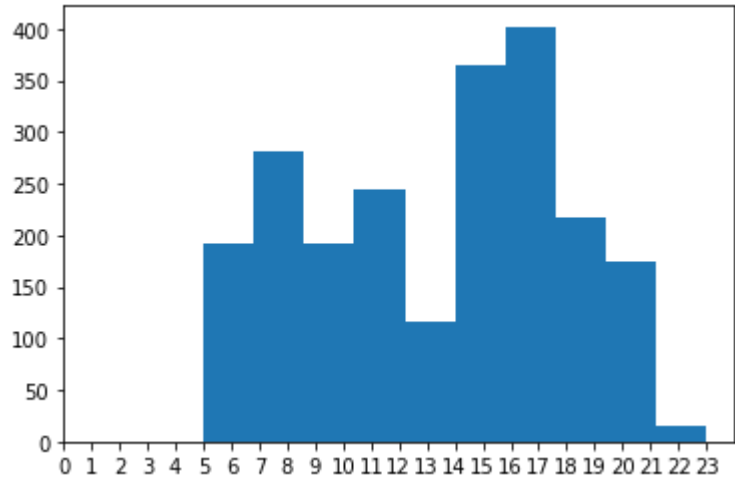


```
In [37]: flightdata['UPDATED_CRS_DEP_TIME'].describe()
```

```
Out[37]: count      2198
unique        59
top          14:55
freq         137
Name: UPDATED_CRS_DEP_TIME, dtype: object
```

```
In [64]: #For UPDATED_DEP_TIME- Histograms- converting in to 24 hour continous format for a better histogram
#Reference - https://sparkbyexamples.com/pandas/pandas-convert-string-column-to-datetime/#:~:text=Use%20pandas%20to_datetime()%20function,string%20you%20wanted%20to%20convert.
```

```
flightdata['UPDATED_DEP_TIME'] = pd.to_datetime(flightdata['UPDATED_DEP_TIME'], infer_datetime_format=True)
hour_list = [t.hour for t in flightdata['UPDATED_DEP_TIME']]
numbers=[x for x in range(0,24)]
labels=map(lambda x: str(x), numbers)
plt.xticks(numbers, labels)
plt.xlim(0,24)
plt.hist(hour_list)
plt.show()
```



```
In [44]: flightdata['UPDATED_DEP_TIME'].describe()
```

```
Out[44]: count      2198
unique      631
top         14:55
freq        25
Name: UPDATED_DEP_TIME, dtype: object
```

```
In [49]: #PIVOT table to summarize total flights between each airports
flightdata['NUM']=1
import numpy as np
table = pd.pivot_table(flightdata,values='NUM',index=['ORIGIN'],columns=['DEST'], aggfunc=np.sum)
table
```

```
Out[49]:
```

	DEST	EWR	JFK	LGA
ORIGIN				
	BWI	115.0	29.0	NaN
	DCA	256.0	149.0	965.0
	IAD	294.0	207.0	183.0

```
In [52]: #Pivot table to show delayed and ontime flight count across the days of the week
table2 = pd.pivot_table(flightdata,values='NUM',index=['FLIGHT_STATUS'],columns=['DAY_WEEK'], aggfunc=np.sum)
table2
```

```
Out[52]:
```

	DAY_WEEK	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
FLIGHT_STATUS								
	delayed	74	83	24	68	57	63	57
	ontime	316	224	226	185	314	244	263

```
In [53]: #Pivot table to show number of flights per carrier across the days of the week
table3 = pd.pivot_table(flightdata,values='NUM',index=['CARRIER'],columns=['DAY_WEEK'], aggfunc=np.sum)
table3
```

Out[53]: DAY\_WEEK Friday Monday Saturday Sunday Thursday Tuesday Wednesday

CARRIER								
CO	18	14	5	11	17	14	15	
DH	93	75	73	69	88	74	77	
DL	69	55	43	42	66	57	56	
MQ	53	41	25	31	55	45	45	
OH	5	4	4	4	4	4	4	
RU	77	53	60	42	70	49	57	
UA	5	4	5	4	5	4	4	
US	70	61	35	50	66	60	62	

In [54]: #Pivot table to show number of flights per carrier across the days of the week  
table4 = pd.pivot\_table(flightdata,values='NUM',index=['CARRIER'],columns=['FLIGHT\_STATUS'], aggfunc=np.sum)  
table4

Out[54]: FLIGHT\_STATUS delayed ontime

CARRIER			
CO	26	68	
DH	135	414	
DL	47	341	
MQ	80	215	
OH	4	25	
RU	94	314	
UA	5	26	
US	35	369	

In [ ]:

In [ ]: