Formula1

Assignment - 1

तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED ‘A’ GRADE

Brief of dataset – dataset contain the races occur in history of Formula1 which contain PlayerId, their fastestlaptime, total lap, points, time, position, grid, resultId, Fastestlapspeed, rank and many other columns.

Dataset contain total of 1096 races and total of 856 drivers with total of 25845 rows and 18 columns.

ResultId contain the name of winner in each race, raceId contain each race with a specific number, driverId contain names of driver, constructorId contain name of constructor, position contain position at end of race, grid contain the position at start of race, points contain point acquired by driver in that race, laps contain how many laps completed by driver, time contain the time required by driver to complete race, fastestlapnumber contain the lap number of fastest lap of each driver, fastestlaptime contain the time taken by driver to complete fastest lap, fastestlapspeed contain average speed of fastest lap.

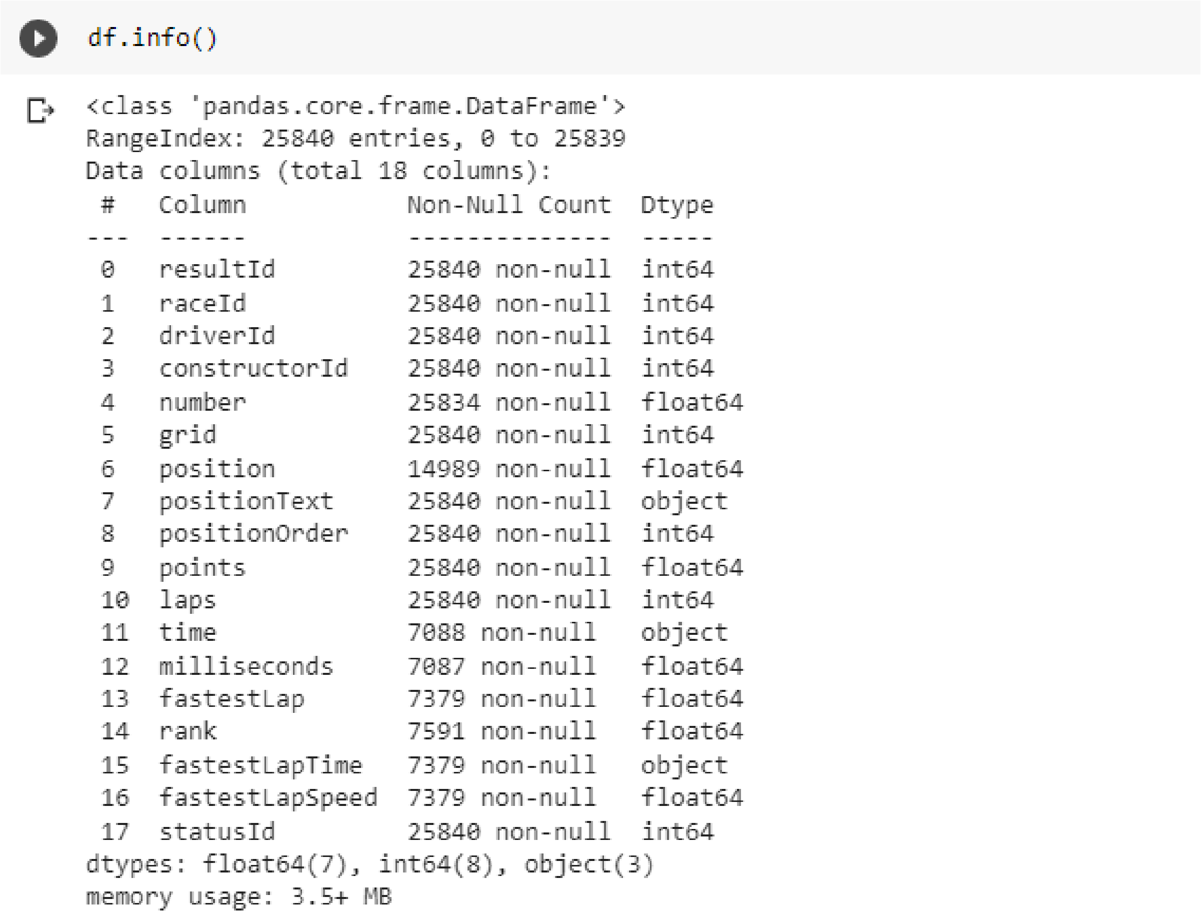
Source – https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

Objective of datamining – Data mining's main goal is to automatically analyse a lot of data. this helps to uncover fascinating patterns. We discuss the collection of data records, peculiar records, and dependencies.

Typically, this calls for the usage of database techniques like spatial indexes. Consequently, it is possible to think of these patterns as a sort of input data summary. as well as being applicable to further analysis such as machine learning or predictive analysis

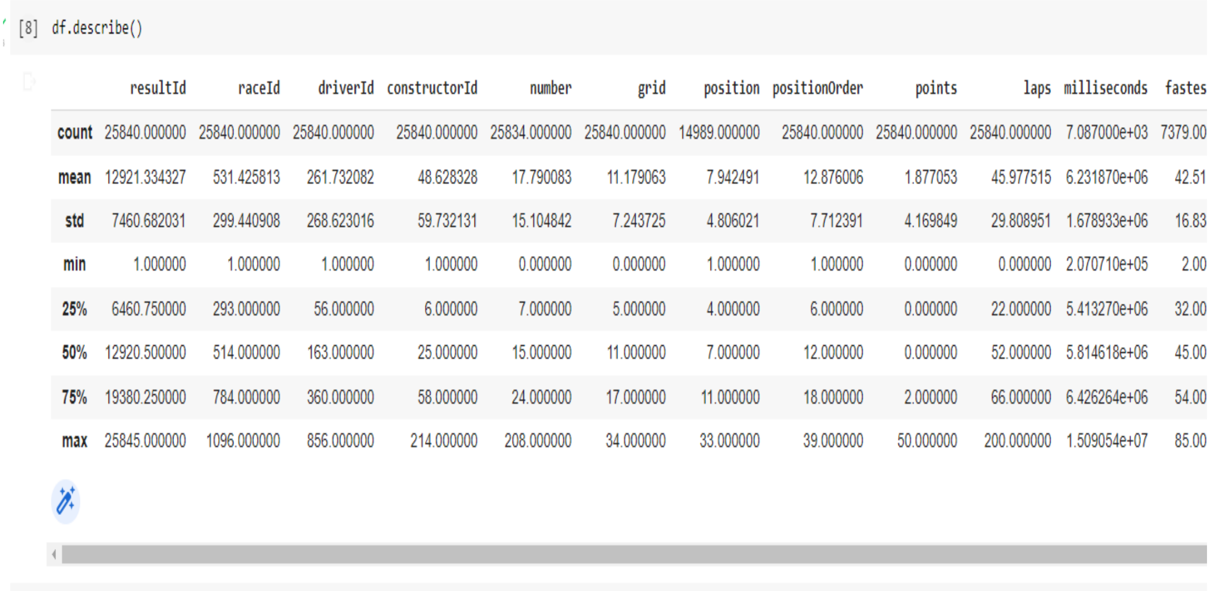
Assignment-1

Preprocessing of data –



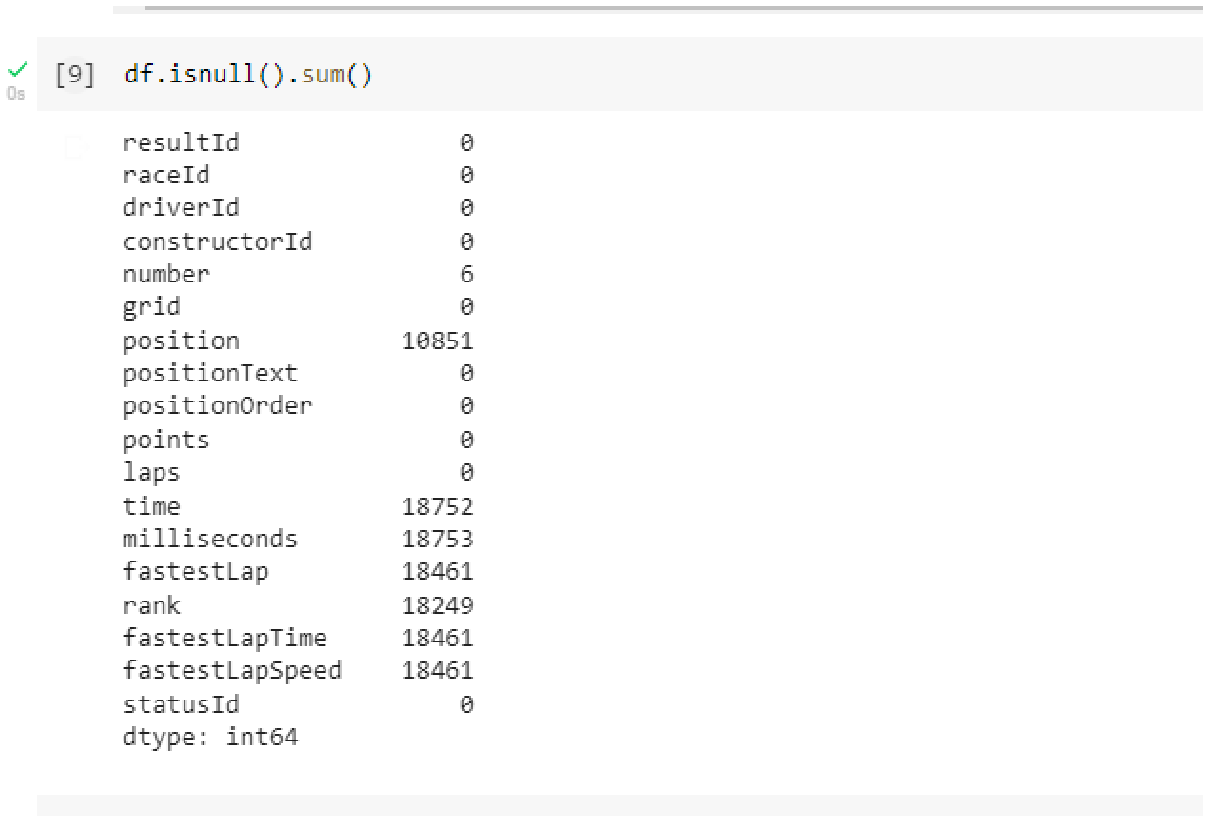
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25840 entries, 0 to 25839
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   resultId        25840 non-null  int64  
 1   raceId          25840 non-null  int64  
 2   driverId        25840 non-null  int64  
 3   constructorId   25840 non-null  int64  
 4   number          25834 non-null  float64
 5   grid            25840 non-null  int64  
 6   position        14989 non-null  float64
 7   positionText    25840 non-null  object 
 8   positionOrder   25840 non-null  int64  
 9   points          25840 non-null  float64
 10  laps            25840 non-null  int64  
 11  time            7088 non-null   object 
 12  milliseconds    7087 non-null   float64
 13  fastestLap      7379 non-null   float64
 14  rank            7591 non-null   float64
 15  fastestLapTime  7379 non-null   object 
 16  fastestLapSpeed 7379 non-null   float64
 17  statusId        25840 non-null  int64  
dtypes: float64(7), int64(8), object(3)
memory usage: 3.5+ MB
```



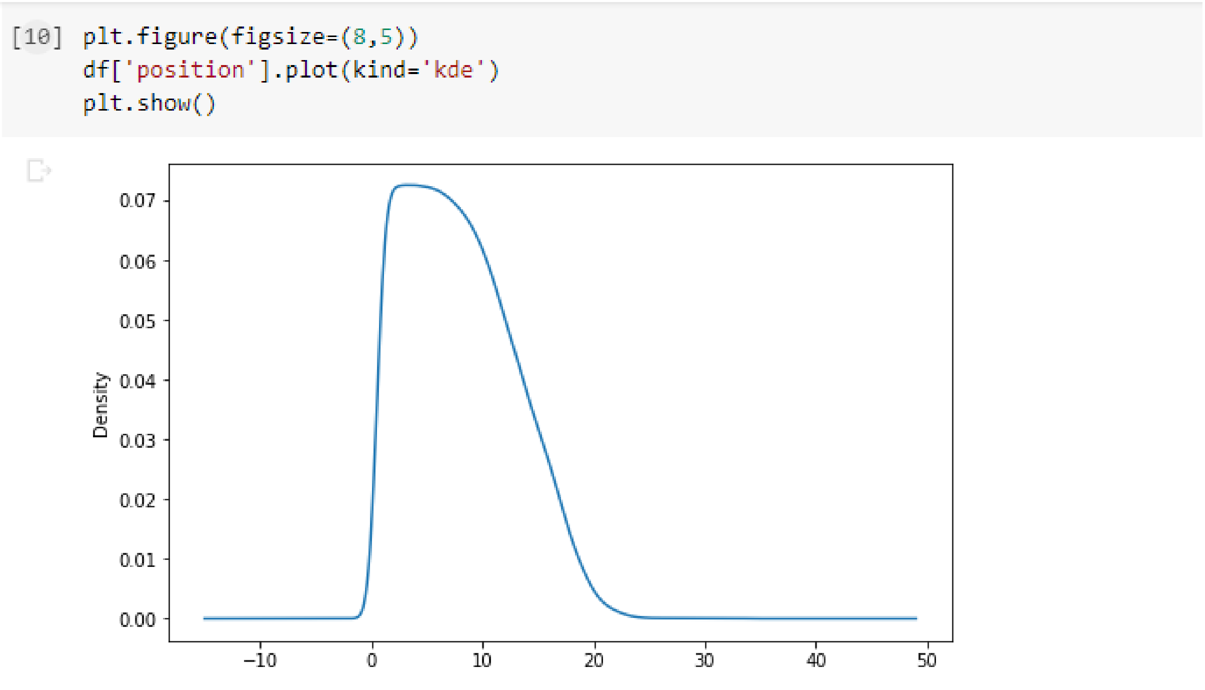
```
[8] df.describe()
```

	resultId	raceId	driverId	constructorId	number	grid	position	positionOrder	points	laps	milliseconds	fastes
count	25840.000000	25840.000000	25840.000000	25840.000000	25834.000000	25840.000000	14989.000000	25840.000000	25840.000000	25840.000000	7.087000e+03	7379.00
mean	12921.334327	531.425813	261.732082	48.628328	17.790083	11.179063	7.942491	12.876006	1.877053	45.977515	6.231870e+06	42.51
std	7460.682031	299.440908	268.623016	59.732131	15.104842	7.243725	4.806021	7.712391	4.169849	29.808951	1.678933e+06	16.83
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	2.070710e+05	2.00
25%	6460.750000	293.000000	56.000000	6.000000	7.000000	5.000000	4.000000	6.000000	0.000000	22.000000	5.413270e+06	32.00
50%	12920.500000	514.000000	163.000000	25.000000	15.000000	11.000000	7.000000	12.000000	0.000000	52.000000	5.814618e+06	45.00
75%	19380.250000	784.000000	360.000000	58.000000	24.000000	17.000000	11.000000	18.000000	2.000000	66.000000	6.426264e+06	54.00
max	25845.000000	1096.000000	856.000000	214.000000	208.000000	34.000000	33.000000	39.000000	50.000000	200.000000	1.509054e+07	85.00



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

```
resultId               0
raceId                 0
driverId               0
constructorId          0
number                 6
grid                   0
position           10851
positionText           0
positionOrder          0
points                 0
laps                   0
time               18752
milliseconds       18753
fastestLap         18461
rank               18249
fastestLapTime     18461
fastestLapSpeed    18461
statusId               0
dtype: int64
```

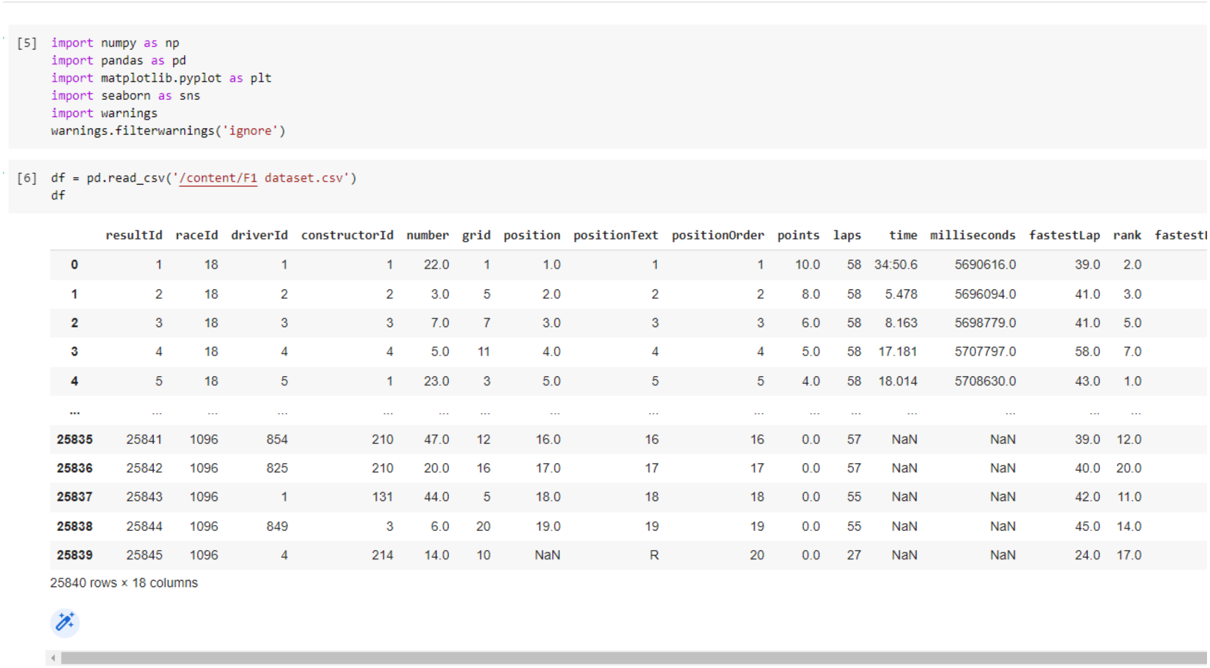


```
[10] plt.figure(figsize=(8,5))
     df['position'].plot(kind='kde')
     plt.show()
```

Assignment-1

Outcomes – By processing the data of formula1 over the years we can able to make some points like which player is making process and which are not. So, the team can then decide how much they can bid for them for next year season. Also, player can also see their record in each race of fastest speed, points, total time. By processing more in data set teams can also find that their cars need some modification with respect to other team performance.

Code –

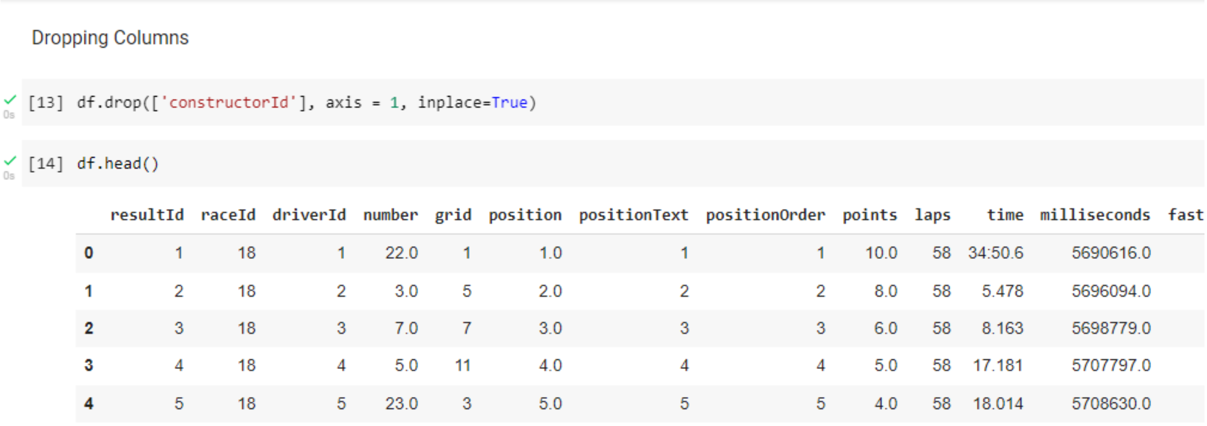


```
[5]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
[6]  df = pd.read_csv('/content/F1 dataset.csv')
     df
```

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fastestl
0	1	18	1	1	22.0	1	1.0	1	1	10.0	58	34:50.6	5690616.0	39.0	2.0	
1	2	18	2	2	3.0	5	2.0	2	2	8.0	58	5.478	5696094.0	41.0	3.0	
2	3	18	3	3	7.0	7	3.0	3	3	6.0	58	8.163	5698779.0	41.0	5.0	
3	4	18	4	4	5.0	11	4.0	4	4	5.0	58	17.181	5707797.0	58.0	7.0	
4	5	18	5	1	23.0	3	5.0	5	5	4.0	58	18.014	5708630.0	43.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25835	25841	1096	854	210	47.0	12	16.0	16	16	0.0	57	NaN	NaN	39.0	12.0	
25836	25842	1096	825	210	20.0	16	17.0	17	17	0.0	57	NaN	NaN	40.0	20.0	
25837	25843	1096	1	131	44.0	5	18.0	18	18	0.0	55	NaN	NaN	42.0	11.0	
25838	25844	1096	849	3	6.0	20	19.0	19	19	0.0	55	NaN	NaN	45.0	14.0	
25839	25845	1096	4	214	14.0	10	NaN	R	20	0.0	27	NaN	NaN	24.0	17.0	

25840 rows × 18 columns

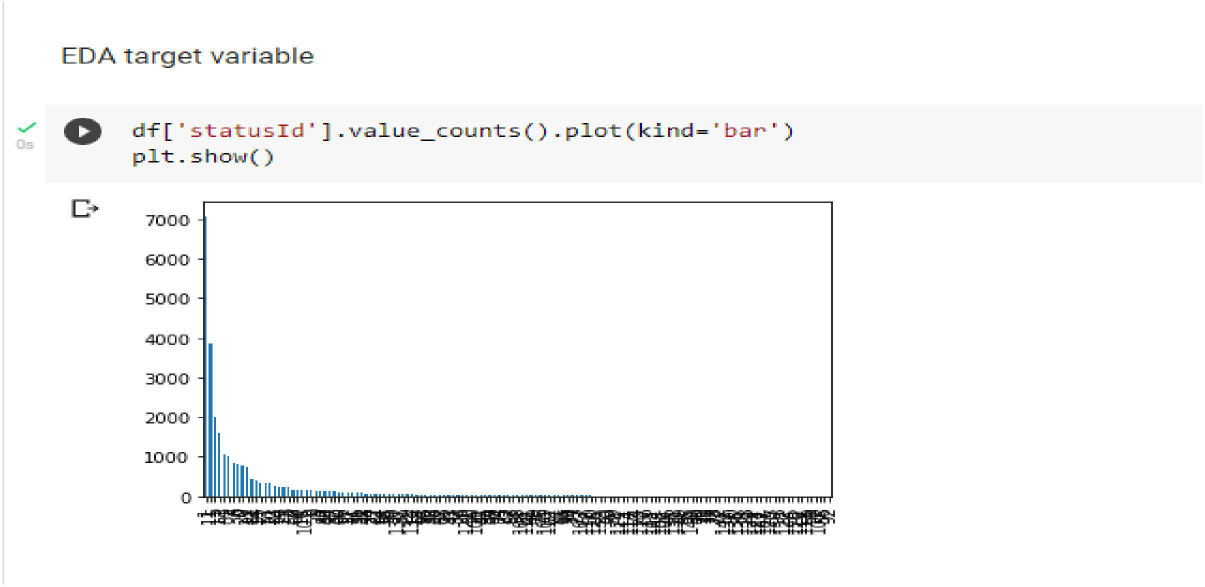


Dropping Columns

```
[13] df.drop(['constructorId'], axis = 1, inplace=True)
```

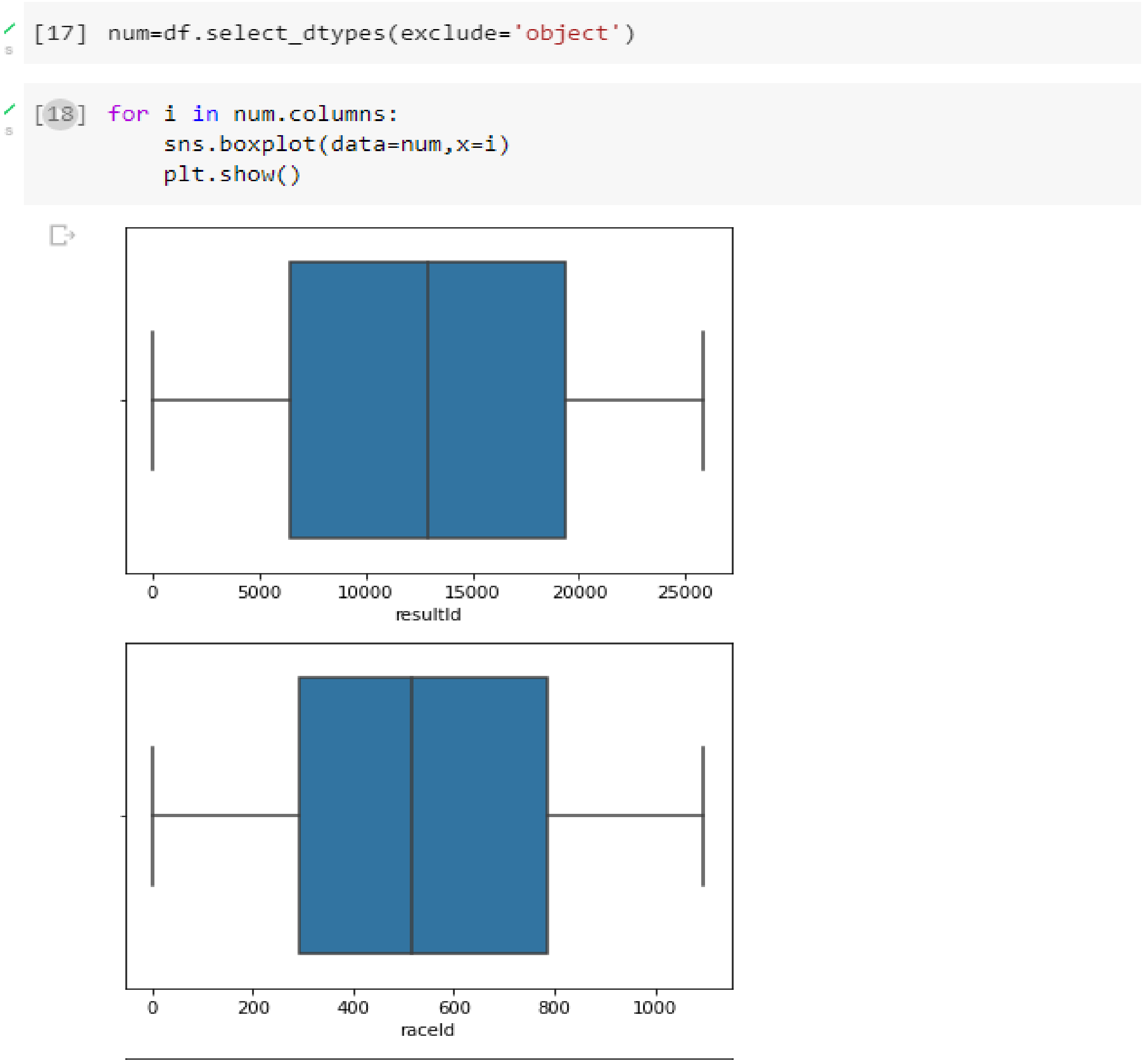
```
[14] df.head()
```

	resultId	raceId	driverId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fast
0	1	18	1	22.0	1	1.0	1	1	10.0	58	34:50.6	5690616.0	
1	2	18	2	3.0	5	2.0	2	2	8.0	58	5.478	5696094.0	
2	3	18	3	7.0	7	3.0	3	3	6.0	58	8.163	5698779.0	
3	4	18	4	5.0	11	4.0	4	4	5.0	58	17.181	5707797.0	
4	5	18	5	23.0	3	5.0	5	5	4.0	58	18.014	5708630.0	



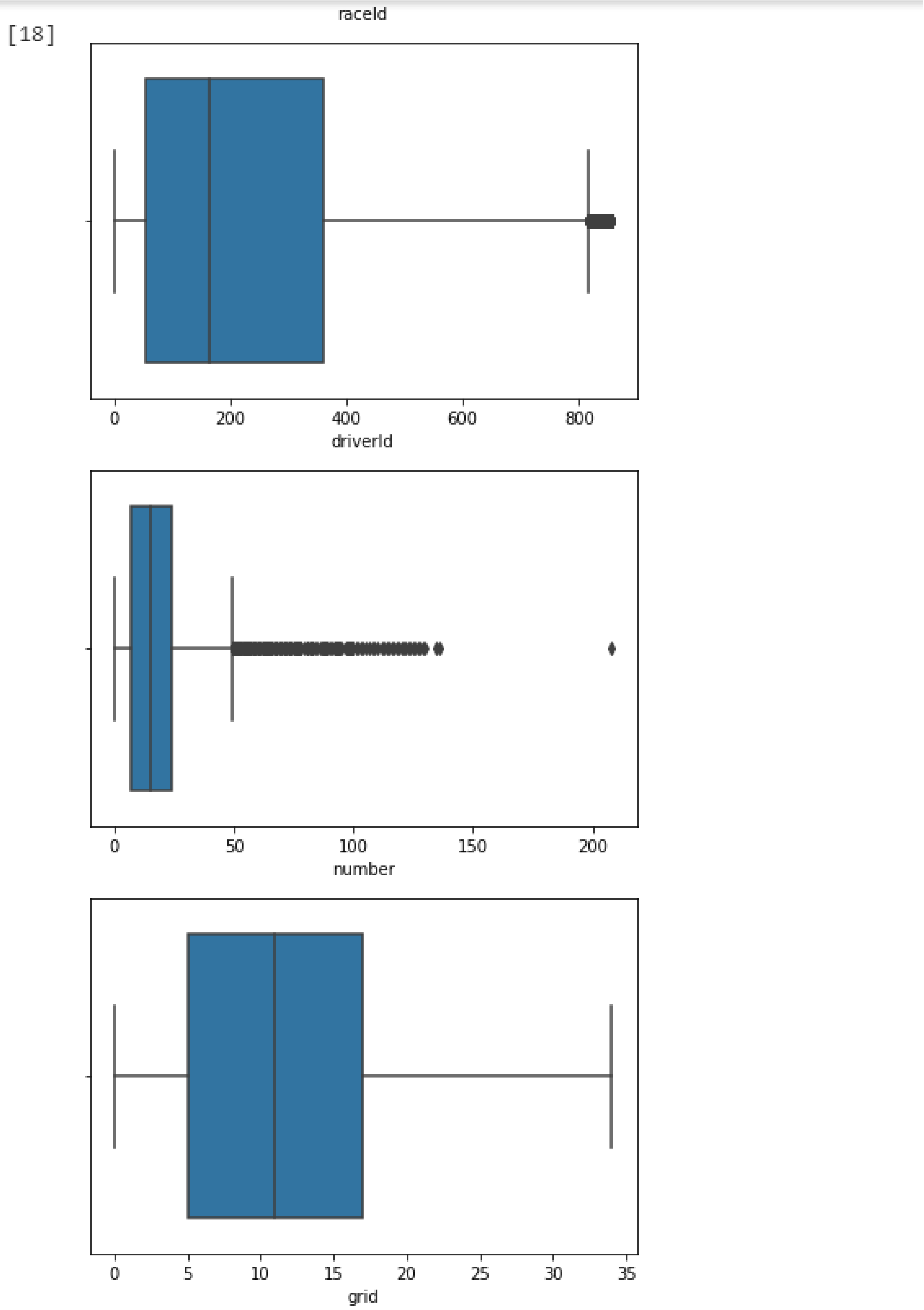
EDA target variable

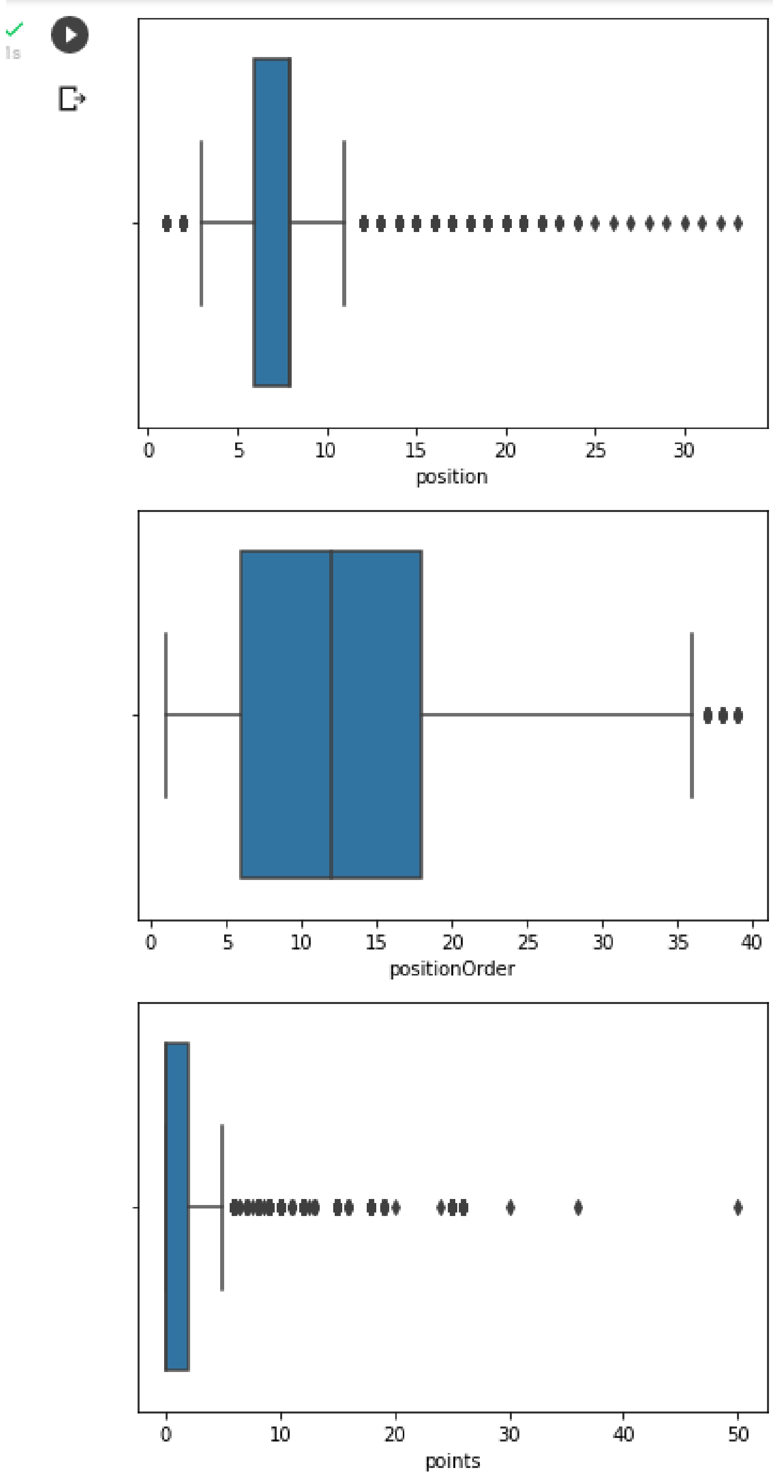
```
df['statusId'].value_counts().plot(kind='bar')
plt.show()
```



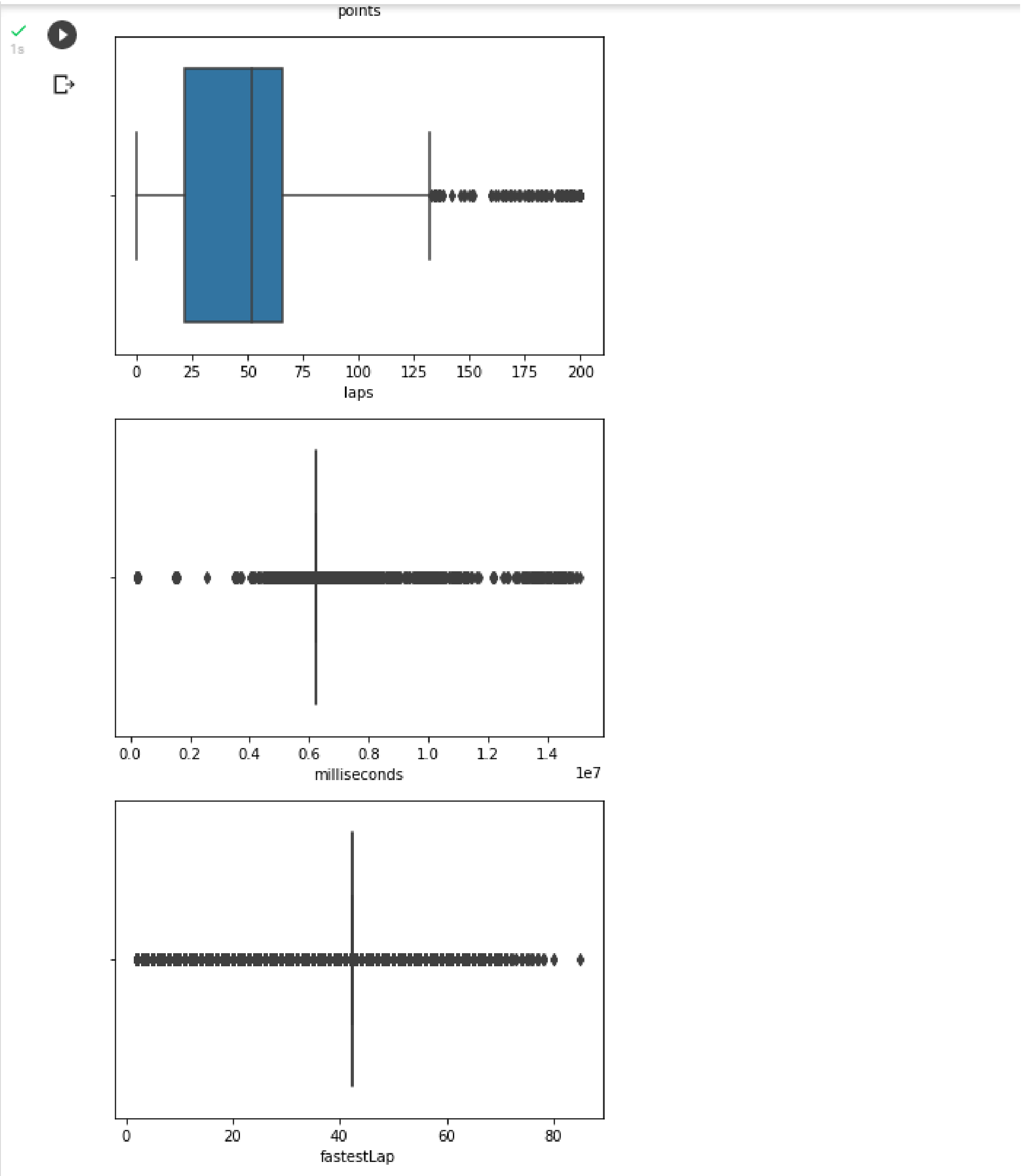
```
[17] num=df.select_dtypes(exclude='object')
```

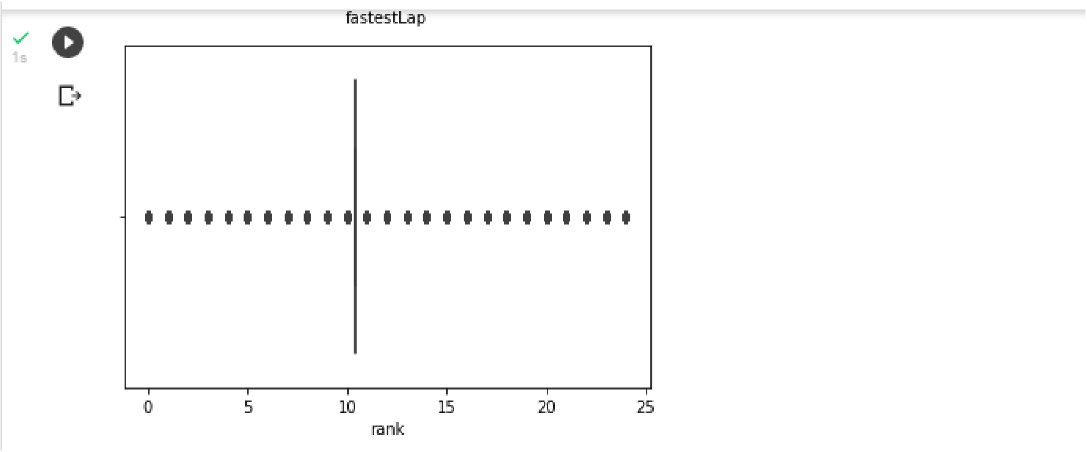
```
[18] for i in num.columns:
         sns.boxplot(data=num,x=i)
         plt.show()
```

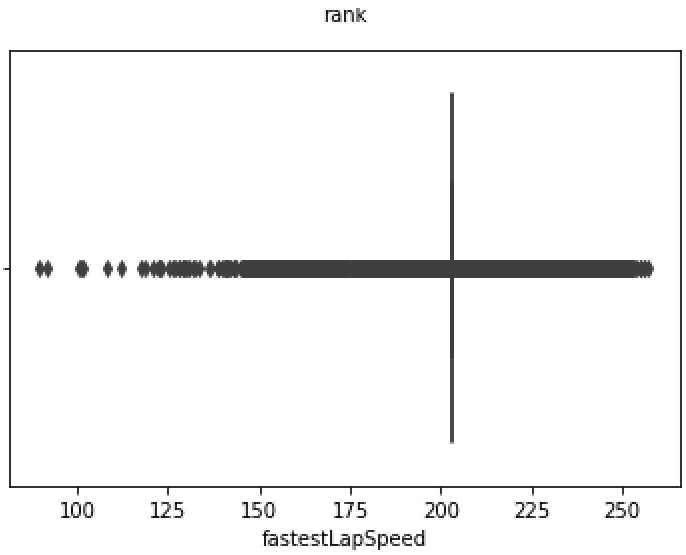

[18]
raceId
0
200
400
600
800
driverId
0
50
100
150
200
number
0
5
10
15
20
25
30
35
grid

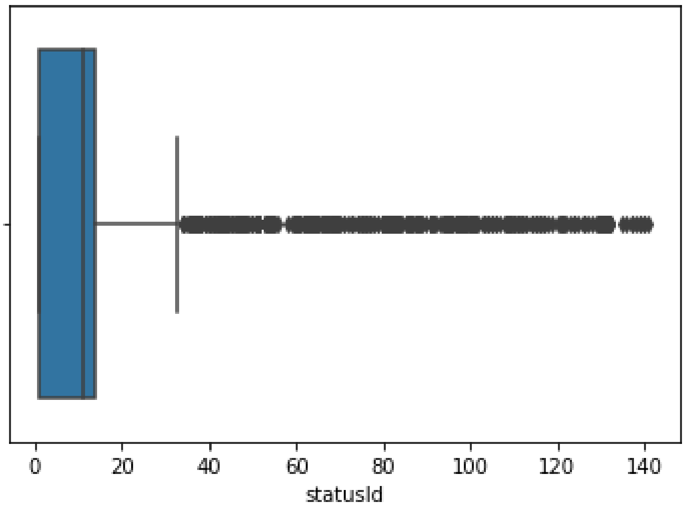

1s
0
5
10
15
20
25
30
position
0
5
10
15
20
25
30
35
40
positionOrder
0
10
20
30
40
50
points

Assignment-1

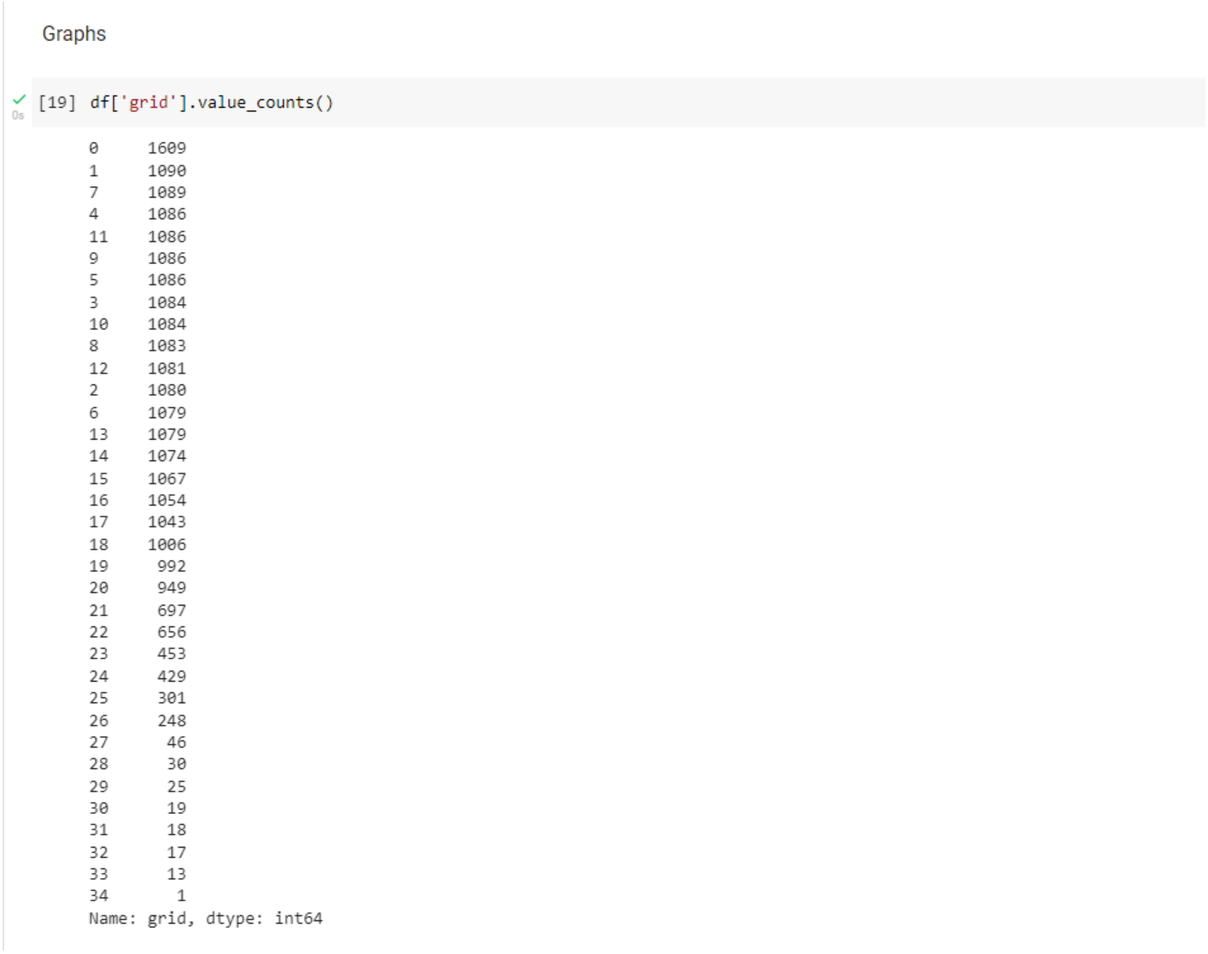





rank
100
125
150
175
200
225
250
fastestLapSpeed


0
20
40
60
80
100
120
140
statusId

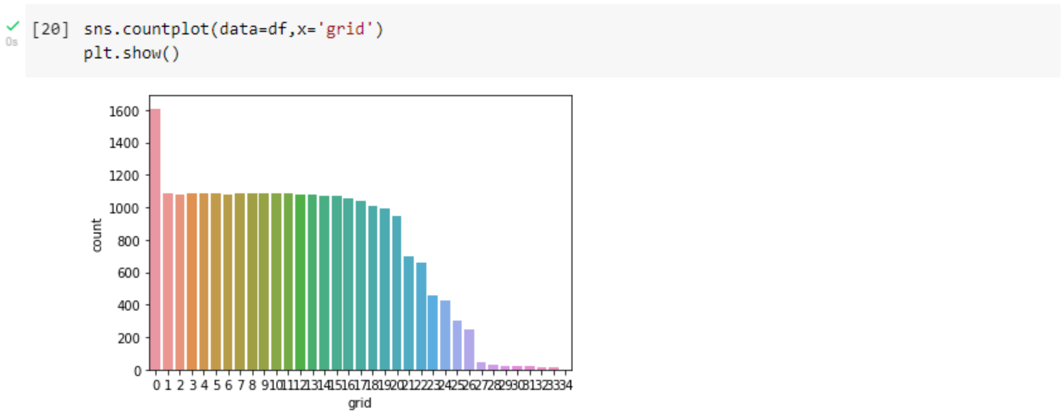
Assignment-1



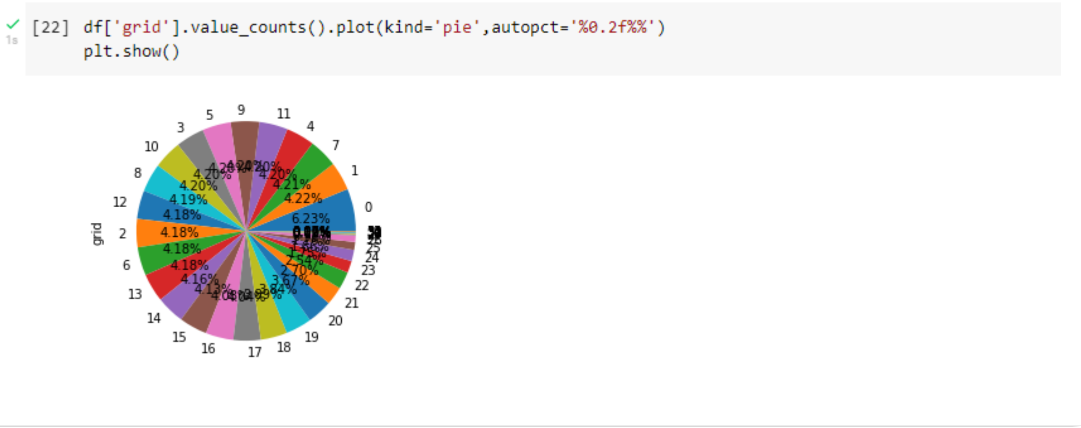
Graphs

```
[19] df['grid'].value_counts()
```

```
0     1609
1     1090
7     1089
4     1086
11    1086
9     1086
5     1086
3     1084
10    1084
8     1083
12    1081
2     1080
6     1079
13    1079
14    1074
15    1067
16    1054
17    1043
18    1006
19     992
20     949
21     697
22     656
23     453
24     429
25     301
26     248
27      46
28      30
29      25
30      19
31      18
32      17
33      13
34       1
Name: grid, dtype: int64
```



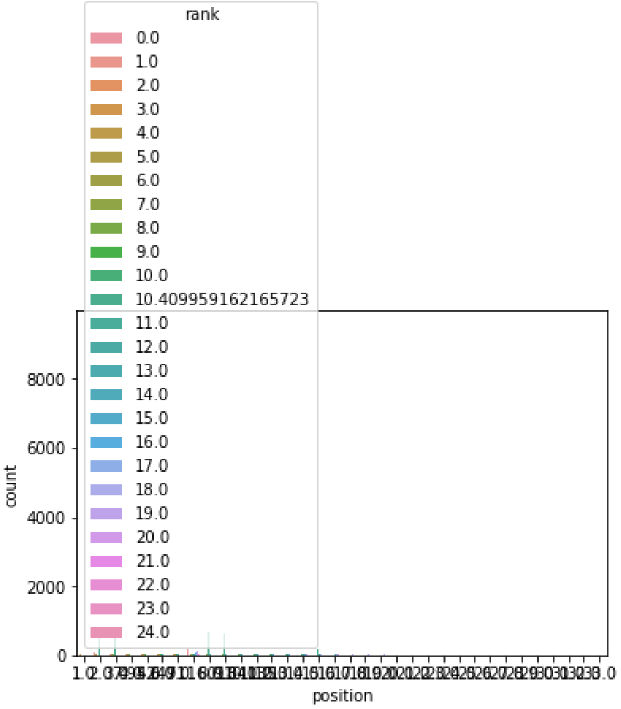
```
[20] sns.countplot(data=df,x='grid')
     plt.show()
```



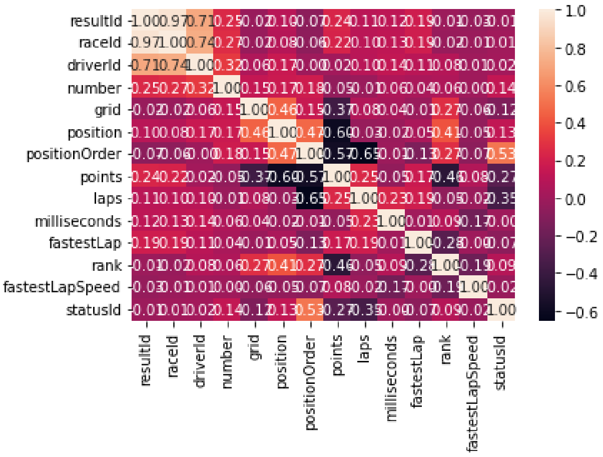
```
[22] df['grid'].value_counts().plot(kind='pie',autopct='%0.2f%%')
     plt.show()
```

Assignment-1

```
[27] sns.countplot(data=df,x='position',hue='rank')
     plt.show()
```



```
[28] sns.heatmap(df.corr(),annot=True,fmt='.2f')
     plt.show()
```


resultId 1.00 0.97 0.71 0.25 0.02 0.10 0.07 0.24 0.11 0.12 0.19 0.01 0.03 0.01
raceId 0.97 1.00 0.74 0.27 0.02 0.08 0.06 0.22 0.10 0.13 0.19 0.02 0.01 0.01
driverId 0.71 0.74 1.00 0.32 0.06 0.17 0.00 0.02 0.10 0.14 0.11 0.08 0.01 0.02
number 0.25 0.27 0.32 1.00 0.15 0.17 0.18 0.05 0.01 0.06 0.04 0.06 0.00 0.14
grid 0.02 0.02 0.06 0.15 1.00 0.46 0.15 0.37 0.08 0.04 0.01 0.27 0.06 0.12
position 0.10 0.08 0.17 0.17 0.46 1.00 0.47 0.60 0.03 0.02 0.05 0.41 0.05 0.13
positionOrder 0.07 0.06 0.00 0.18 0.15 0.47 1.00 0.57 0.65 0.01 0.13 0.27 0.07 0.53
points 0.24 0.22 0.02 0.05 0.37 0.60 0.57 1.00 0.25 0.05 0.17 0.46 0.08 0.27
laps 0.11 0.10 0.10 0.01 0.08 0.03 0.65 0.25 1.00 0.23 0.19 0.05 0.02 0.35
milliseconds 0.12 0.13 0.14 0.06 0.04 0.02 0.01 0.05 0.23 1.00 0.01 0.09 0.17 0.00
fastestLap 0.19 0.19 0.11 0.04 0.01 0.05 0.13 0.17 0.19 0.01 1.00 0.28 0.09 0.07
rank 0.01 0.02 0.08 0.06 0.27 0.41 0.27 0.46 0.05 0.09 0.28 1.00 0.19 0.09
fastestLapSpeed 0.03 0.01 0.01 0.00 0.06 0.05 0.07 0.08 0.02 0.17 0.09 0.19 1.00 0.02
statusId 0.01 0.01 0.02 0.14 0.12 0.13 0.53 0.27 0.35 0.09 0.07 0.09 0.02 1.00
resultId
raceId
driverId
number
grid
position
positionOrder
points
laps
milliseconds
fastestLap
rank
fastestLapSpeed
statusId