Formula1

Assignment - 2

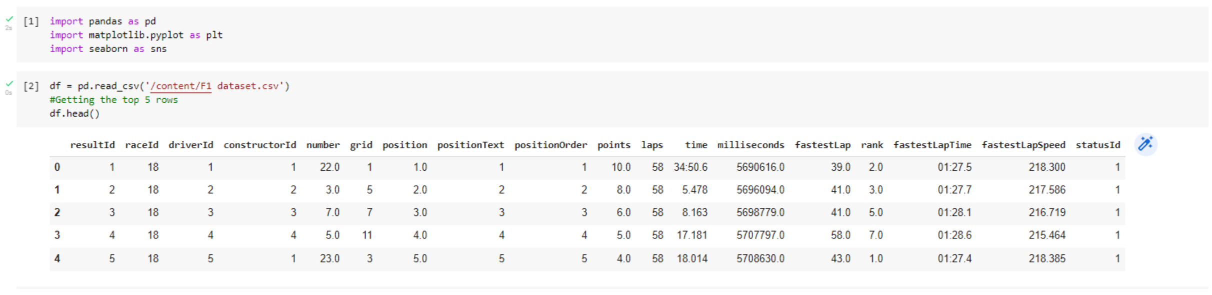
तमसो मा ज्योतिर्गमय

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

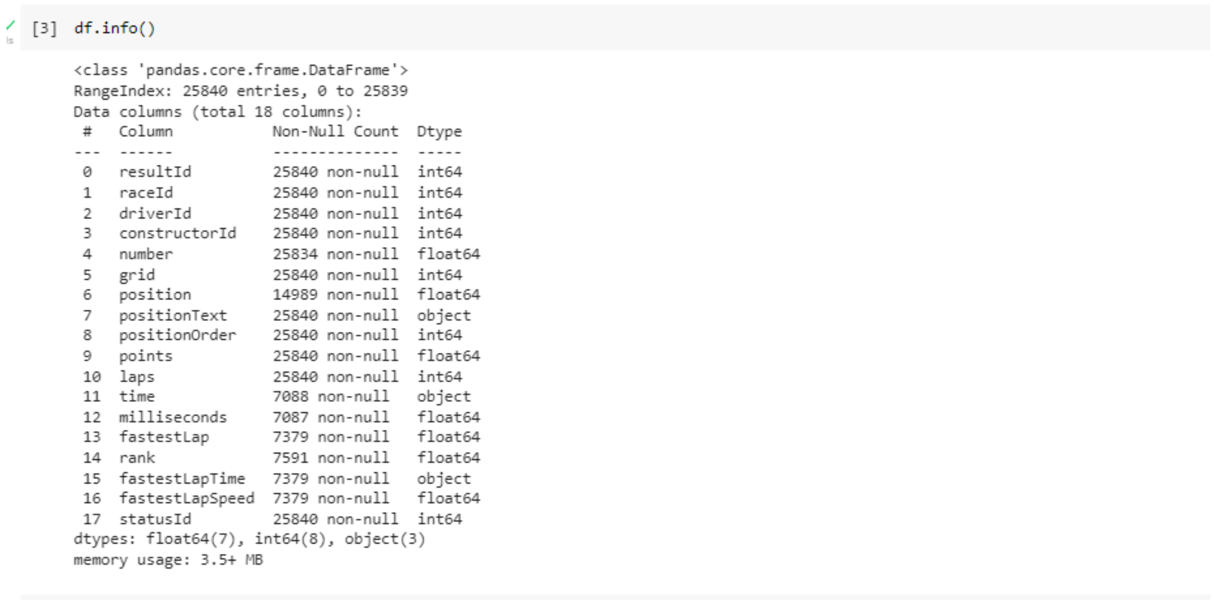
Code –



```
[1] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
```

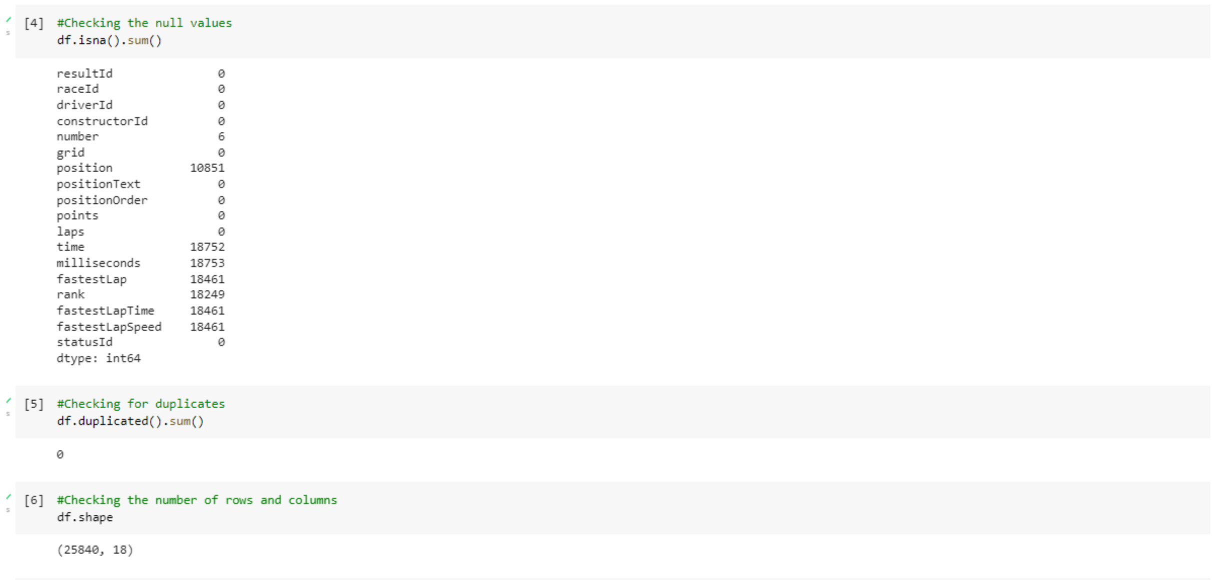
```
[2] df = pd.read_csv('/content/F1 dataset.csv')
    #Getting the top 5 rows
    df.head()
```

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fastestLapTime	fastestLapSpeed	statusId
0	1	18	1	1	22.0	1	1.0	1	1	10.0	58	34:50.6	5690616.0	39.0	2.0	01:27.5	218.300	1
1	2	18	2	2	3.0	5	2.0	2	2	8.0	58	5.478	5696094.0	41.0	3.0	01:27.7	217.586	1
2	3	18	3	3	7.0	7	3.0	3	3	6.0	58	8.163	5698779.0	41.0	5.0	01:28.1	216.719	1
3	4	18	4	4	5.0	11	4.0	4	4	5.0	58	17.181	5707797.0	58.0	7.0	01:28.6	215.464	1
4	5	18	5	1	23.0	3	5.0	5	5	4.0	58	18.014	5708630.0	43.0	1.0	01:27.4	218.385	1



```
[3] 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
```



```
[4] #Checking the null values
    df.isna().sum()
```

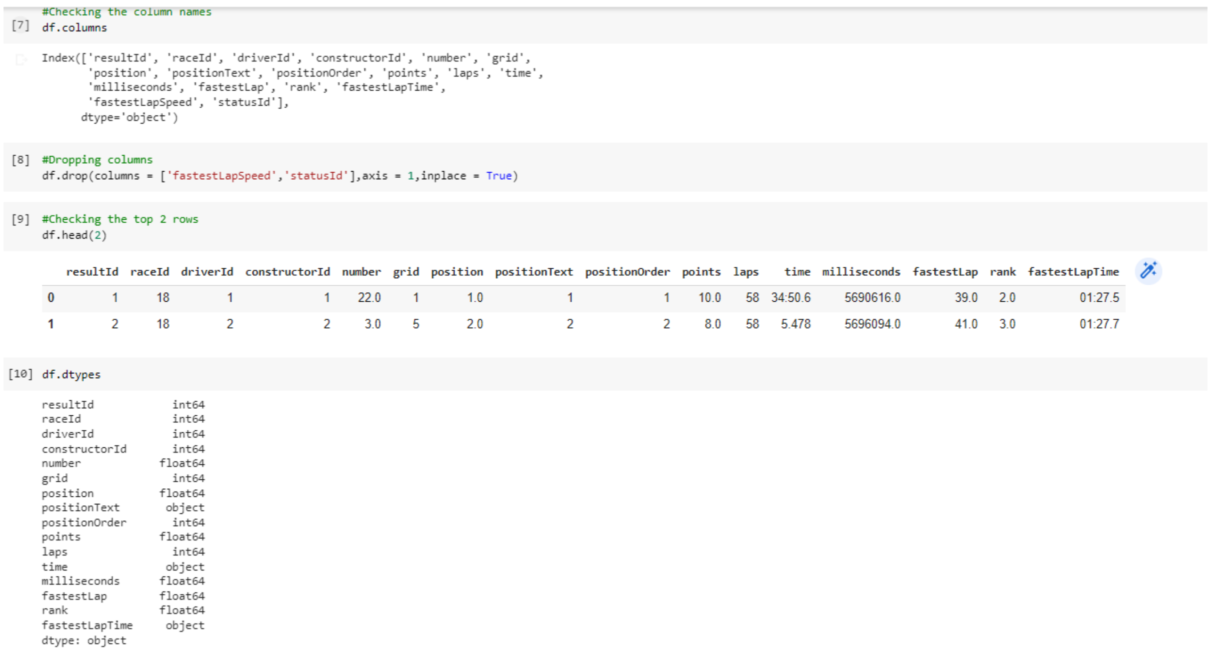
```
resultId               0
raceId                 0
driverId               0
constructorId          0
number                 6
grid                   0
position           10851
positionText           0
positionOrder          0
points                 0
laps                   0
time               18752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

```
[5] #Checking for duplicates
    df.duplicated().sum()
```

```
0
```

```
[6] #Checking the number of rows and columns
    df.shape
```

```
(25840, 18)
```



```
#Checking the column names
[7] df.columns
```

```
Index(['resultId', 'raceId', 'driverId', 'constructorId', 'number', 'grid',
       'position', 'positionText', 'positionOrder', 'points', 'laps', 'time',
       'milliseconds', 'fastestLap', 'rank', 'fastestLapTime',
       'fastestLapSpeed', 'statusId'],
      dtype='object')
```

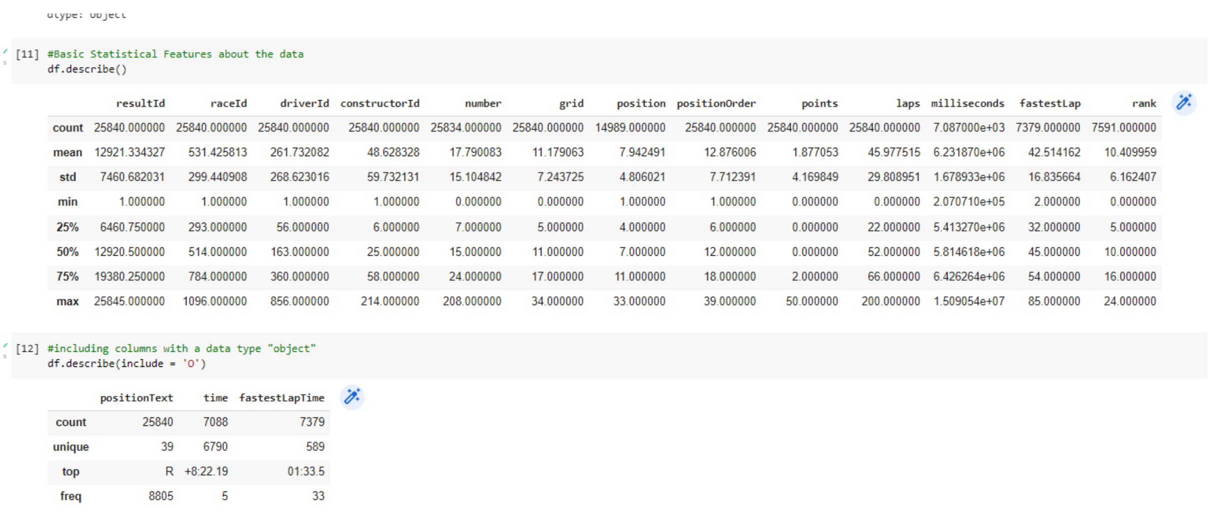
```
[8] #Dropping columns
    df.drop(columns = ['fastestLapSpeed','statusId'],axis = 1,inplace = True)
```

```
[9] #Checking the top 2 rows
    df.head(2)
```

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fastestLapTime
0	1	18	1	1	22.0	1	1.0	1	1	10.0	58	34:50.6	5690616.0	39.0	2.0	01:27.5
1	2	18	2	2	3.0	5	2.0	2	2	8.0	58	5.478	5696094.0	41.0	3.0	01:27.7

```
[10] df.dtypes
```

```
resultId            int64
raceId              int64
driverId            int64
constructorId       int64
number            float64
grid                int64
position          float64
positionText       object
positionOrder       int64
points            float64
laps                int64
time               object
milliseconds      float64
fastestLap        float64
rank              float64
fastestLapTime     object
dtype: object
```



```
dtype: object
```

```
[11] #Basic Statistical Features about the data
     df.describe()
```

	resultId	raceId	driverId	constructorId	number	grid	position	positionOrder	points	laps	milliseconds	fastestLap	rank
count	25840.000000	25840.000000	25840.000000	25840.000000	25834.000000	25840.000000	14989.000000	25840.000000	25840.000000	25840.000000	7.087000e+03	7379.000000	7591.000000
mean	12921.334327	531.425813	261.732082	48.628328	17.790083	11.179063	7.942491	12.876006	1.877053	45.977515	6.231870e+06	42.514162	10.409959
std	7460.682031	299.440908	268.623016	59.732131	15.104842	7.243725	4.806021	7.712391	4.169849	29.808951	1.678933e+06	16.835664	6.162407
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	2.070710e+05	2.000000	0.000000
25%	6460.750000	293.000000	56.000000	6.000000	7.000000	5.000000	4.000000	6.000000	0.000000	22.000000	5.413270e+06	32.000000	5.000000
50%	12920.500000	514.000000	163.000000	25.000000	15.000000	11.000000	7.000000	12.000000	0.000000	52.000000	5.814618e+06	45.000000	10.000000
75%	19380.250000	784.000000	360.000000	58.000000	24.000000	17.000000	11.000000	18.000000	2.000000	66.000000	6.426264e+06	54.000000	16.000000
max	25845.000000	1096.000000	856.000000	214.000000	208.000000	34.000000	33.000000	39.000000	50.000000	200.000000	1.509054e+07	85.000000	24.000000

```
[12] #including columns with a data type "object"
     df.describe(include = 'O')
```

	positionText	time	fastestLapTime
count	25840	7088	7379
unique	39	6790	589
top	R	+8:22.19	01:33.5
freq	8805	5	33

```
[13] #Checking the null values
     df.isna().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
dtype: int64
```

```
[15] #Dropping few morecolumns that may not be required for analysis
     df.drop(columns = ['positionText','positionOrder'],axis = 1,inplace =True)
```

```
[16] #Checking the unique number of drivers
     df['driverId'].unique()
```

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,  39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52,
        53,  56,  63,  62,  59,  66,  54,  55,  57,  58,  60,  61,  64,
        65,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,  79,
        80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,  92,
        93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104, 105,
       106, 107, 108, 110, 109, 111, 112, 113, 114, 115, 116, 117, 118,
       119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131,
       132, 135, 136, 137, 138, 139, 133, 140, 141, 142, 143, 144, 145,
       146, 147, 148, 151, 149, 150, 152,  67, 153, 154, 155, 156, 157,
       158, 159, 163, 160, 161, 162, 164, 134, 165, 166, 167, 168, 169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
       196, 197, 198, 199, 200, 206, 201, 202, 203, 204, 205, 207, 208,
       209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
       222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
       235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247,
       248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
       261, 262, 263, 267, 264, 265, 266, 268, 269, 270, 271, 272, 273,
       274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286,
       287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299,
       300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312,
       314, 313, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325,
       326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338,
       339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351,
       352, 353, 354, 355, 356, 357, 362, 358, 359, 360, 361, 363, 364,
       365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377,
       378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
       391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403,
       404, 405, 406, 408, 407, 409, 410, 411, 412, 413, 414, 415, 416,
       417, 418, 420, 419, 421, 422, 423, 424, 425, 426, 427, 428, 429,
       430, 431, 432, 433, 434, 435, 440, 436, 437, 438, 439, 441, 442,
       443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455,
       456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468,
       469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481,
       482, 483, 485, 484, 487, 488, 489, 490, 486, 491, 492, 493, 494,
       495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507,
       508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520,
       521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533,
       534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546,
       547, 548, 554, 549, 550, 551, 552, 553, 555, 556, 557, 558, 559,
       560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572,
       573, 574, 575, 576, 577, 578, 579, 580, 581, 590, 582, 583, 584,
       585, 586, 587, 588, 589, 591, 592, 593, 594, 595, 596, 597, 598,
       599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611,
       612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624,
       625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637,
       638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650,
[17] #Checking the column names
     df.columns
```

```
Index(['resultId', 'raceId', 'driverId', 'constructorId', 'number', 'grid',
       'position', 'points', 'laps', 'time', 'milliseconds', 'fastestLap',
       'rank', 'fastestLapTime'],
      dtype='object')
```

```
[18] #Renaming the column names
     df.rename(columns ={'resultId':'result_Id','raceId':'race_Id','driverId': 'driver_Id','constructorId':'constructor_Id'},inplace=True)
     #df.rename(columns={'old_name': 'new_name'}, inplace=True)
     df.head()
```

	result_Id	race_Id	driver_Id	constructor_Id	number	grid	position	points	laps	time	milliseconds	fastestLap	rank	fastestLapTime
0	1	18	1	1	22.0	1	1.0	10.0	58	34:50.6	5690616.0	39.0	2.0	01:27.5
1	2	18	2	2	3.0	5	2.0	8.0	58	5.478	5696094.0	41.0	3.0	01:27.7
2	3	18	3	3	7.0	7	3.0	6.0	58	8.163	5698779.0	41.0	5.0	01:28.1
3	4	18	4	4	5.0	11	4.0	5.0	58	17.181	5707797.0	58.0	7.0	01:28.6
4	5	18	5	1	23.0	3	5.0	4.0	58	18.014	5708630.0	43.0	1.0	01:27.4

```
[20] #Checking the null values
     df.isna().sum()
```

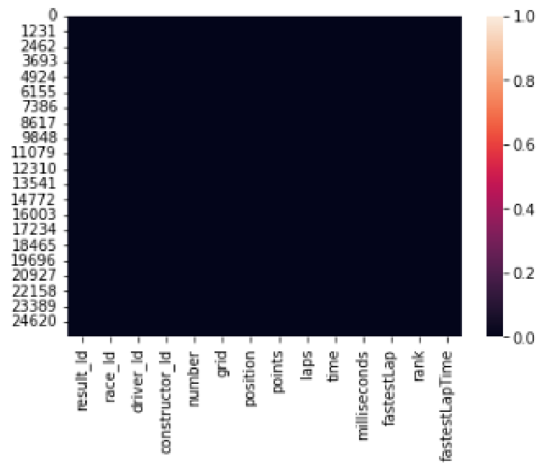
```
result_Id             0
race_Id               0
driver_Id             0
constructor_Id        0
number                6
grid                  0
position          10851
points                0
laps                  0
time              18752
milliseconds      18753
fastestLap        18461
rank              18249
fastestLapTime    18461
dtype: int64
```

```
[22] #Filling the missing values with 'Not_Specified'
     df['position'].fillna("Not_Specified", inplace=True)
     df['time'].fillna("Not_Specified", inplace=True)
     df['milliseconds'].fillna("Not_Specified", inplace=True)
     df['fastestLap'].fillna("Not_Specified", inplace=True)
     df['rank'].fillna("Not_Specified", inplace=True)
     df['fastestLapTime'].fillna("Not_Specified", inplace=True)
     df.head(2)
```

	result_Id	race_Id	driver_Id	constructor_Id	number	grid	position	points	laps	time	milliseconds	fastestLap	rank	fastestLapTime
0	1	18	1	1	22.0	1	1.0	10.0	58	34:50.6	5690616.0	39.0	2.0	01:27.5
1	2	18	2	2	3.0	5	2.0	8.0	58	5.478	5696094.0	41.0	3.0	01:27.7

```
[25] #Confirming if all the missing values has been removed
     sns.heatmap(df.isna())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c8a197f0>
```



```
[26] df.columns
```

```
Index(['result_Id', 'race_Id', 'driver_Id', 'constructor_Id', 'number', 'grid',
       'position', 'points', 'laps', 'time', 'milliseconds', 'fastestLap',
       'rank', 'fastestLapTime'],
      dtype='object')
```

```
[27] #Checking the unique categories present
     df['race_Id'].unique()
```

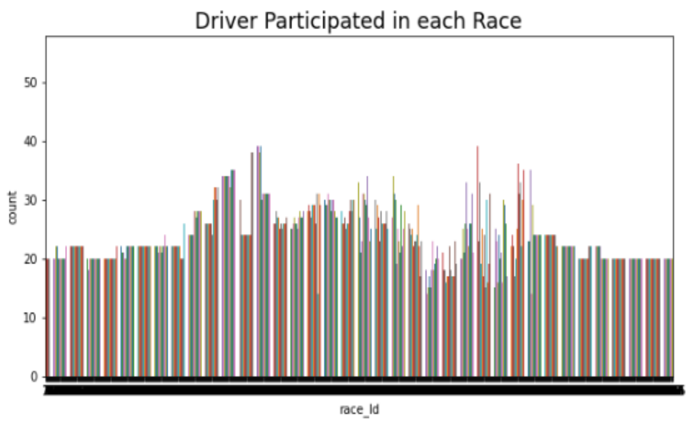
```
array([  18,   19,   20, ..., 1094, 1095, 1096])
```

```
[29] # setting the view for side by side layout
     plt.figure(figsize = (10,5))

     # Setting the Supertitle,font size and font color
     plt.title("Driver Participated in each Race",color="black",size=18)

     #Distribution of Job preference
     ax = sns.countplot(data = df, x = 'race_Id',palette="tab10" )

     #Showning the plot
     plt.show()
```



```
[30] #Checking the unique values
     df['laps'].unique()
```

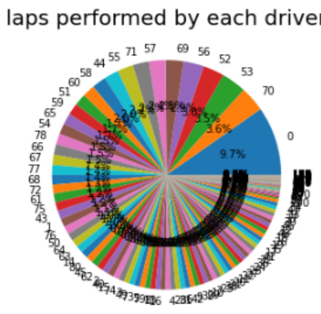
```
     array([ 58,  57,  55,  53,  47,  43,  32,  30,  29,  25,  19,   8,   0,
             56,  54,  39,   5,   1,  40,  66,  65,  41,  34,  21,   7,   6,
             24,  76,  75,  72,  67,  59,  36,  70,  69,  51,  46,  44,  13,
             16,  60,  38,  35,  10,  50,  68,  62,  22,  45,  42,  52,  11,
             61,  49,  14,   2,  71,  48,  28,  26,  64,  63,   9,  78,  77,
             17,  73,  18,   4,  33,  20,  15,  37,  23,  12,  31,   3,  27,
             74,  83,  82,  81,  80,  79,  84,  85,  96,  95,  93,  92,  90,
             94,  89,  88,  87,  86, 108, 107, 106, 105, 104, 101, 100,  98,
            103,  99,  97, 110, 109, 102,  91, 200, 196, 194, 191, 185, 152,
            134, 132, 129, 125, 182, 163, 162, 150, 147, 136, 115, 112, 189,
            151, 148, 122, 116, 199, 197, 195, 192, 170, 138, 111, 187, 175,
            173, 160, 131, 178, 168, 142, 120, 119, 193, 181, 172, 165, 130,
            190, 184, 183, 177, 176, 169, 166, 146, 180, 135, 126, 123, 137,
            133, 128, 127])
```

```
✓  [31] # setting the size
2s      plt.figure(figsize=(5,5))

        #Distribution of preference of work
        status_of_product= df['laps'].value_counts()
        plt.pie(status_of_product, labels=status_of_product.index,autopct='%1.1f%%')

        #Setting up the title , font size and font color
        plt.title('laps performed by each driver',color = 'black',fontsize = 20)

        #to show
        plt.show()
```



```
✓  [32] #CHecking the unique quantity
0s      df['rank'].unique()
```

```
        array([2.0, 3.0, 5.0, 7.0, 1.0, 14.0, 12.0, 4.0, 9.0, 13.0, 15.0, 16.0,
               6.0, 11.0, 10.0, 17.0, 'Not_Specified', 8.0, 18.0, 19.0, 20.0,
               21.0, 22.0, 23.0, 24.0, 0.0], dtype=object)
```