In [50]:

```
#Importing required library
import pandas as pd
import numpy as np
```

Part 1 -NBA Dataset

In [51]:

```
#loading dataset
df=pd.read_csv("C:/Users/Admin/Desktop/Data3/nba.csv")
```

In [52]:

```
df.shape         #Shows the number of rows and columns as a tuple (number of rows,
```

Out[52]:

```
(458, 9)
```

In [53]:

```
df.size
```

Out[53]:

```
4122
```

In [54]:

```
df.columns
```

Out[54]:

```
Index(['Name', 'Team', 'Number', 'Position', 'Age', 'Height', 'Weigh
t',
       'College', 'Salary'],
      dtype='object')
```

In [55]:

```
df.head(10)
```

Out[55]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
5	Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	NaN	12000000.0
6	Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2165160.0
8	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36.0	PG	22.0	6-4	220.0	Oklahoma State	3431040.0

In [56]:

```
df.tail(10)
```

Out[56]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
448	Gordon Hayward	Utah Jazz	20.0	SF	26.0	6-8	226.0	Butler	15409570.0
449	Rodney Hood	Utah Jazz	5.0	SG	23.0	6-8	206.0	Duke	1348440.0
450	Joe Ingles	Utah Jazz	2.0	SF	28.0	6-8	226.0	NaN	2050000.0
451	Chris Johnson	Utah Jazz	23.0	SF	26.0	6-6	206.0	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41.0	PF	20.0	6-10	234.0	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8.0	PG	26.0	6-3	203.0	Butler	2433333.0
454	Raul Neto	Utah Jazz	25.0	PG	24.0	6-1	179.0	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21.0	C	26.0	7-3	256.0	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24.0	C	26.0	7-0	231.0	Kansas	947276.0
457	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [57]:

```
df.info()   #it returns the total count of not null values and checks the data type
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 458 entries, 0 to 457
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      457 non-null    object 
 1   Team      457 non-null    object 
 2   Number    457 non-null    float64
 3   Position  457 non-null    object 
 4   Age       457 non-null    float64
 5   Height    457 non-null    object 
 6   Weight    457 non-null    float64
 7   College   373 non-null    object 
 8   Salary    446 non-null    float64
dtypes: float64(4), object(5)
memory usage: 32.3+ KB
```

In [58]:

```
df.describe()
```

Out[58]:

	Number	Age	Weight	Salary
count	457.000000	457.000000	457.000000	4.460000e+02
mean	17.678337	26.938731	221.522976	4.842684e+06
std	15.966090	4.404016	26.368343	5.229238e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.044792e+06
50%	13.000000	26.000000	220.000000	2.839073e+06
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

In [59]:

```
df.dtypes           #Returns data type of eah column
```

Out[59]:

```
Name         object
Team         object
Number      float64
Position     object
Age         float64
Height       object
Weight      float64
College      object
Salary      float64
dtype: object
```

checking for missing values

In [60]:

```
df.isnull()          #detect missing values(Missing values gets mapped to True an
```

Out[60]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...
453	False	False	False	False	False	False	False	False	False
454	False	False	False	False	False	False	False	True	False
455	False	False	False	False	False	False	False	True	False
456	False	False	False	False	False	False	False	False	False
457	True	True	True	True	True	True	True	True	True

458 rows × 9 columns

In [61]:

```
df.isnull().sum()
```

Out[61]:

```
Name         1
Team         1
Number       1
Position     1
Age          1
Height       1
Weight       1
College     85
Salary      12
dtype: int64
```

In [62]:

```
df.isnull().sum().sum()      #returns the number of missing values in the data set.
```

Out[62]:

```
104
```

In [63]:

```
df['Salary'].fillna(df['Salary'].mean(),inplace=True)
```

In [64]:

```
df.isnull().sum()
```

Out[64]:

```
Name         1
Team         1
Number       1
Position     1
Age          1
Height       1
Weight       1
College     85
Salary       0
dtype: int64
```

In [65]:

```
mode = df['College'].mode()[0]
df['College'].fillna(mode,inplace=True)
```

In [66]:

```
df.isnull().sum()
```

Out[66]:

```
Name        1
Team        1
Number      1
Position    1
Age         1
Height      1
Weight      1
College     0
Salary      0
dtype: int64
```

In [67]:

```
df['Age'].fillna(df['Age'].median(),inplace=True)
df['Weight'].fillna(df['Weight'].median(),inplace=True)
```

In [68]:

```
df.isnull().sum()
```

Out[68]:

```
Name        1
Team        1
Number      1
Position    1
Age         0
Height      1
Weight      0
College     0
Salary      0
dtype: int64
```

In [69]:

```
df.dropna(inplace=True)
```

In [70]:

```
df.isnull().sum()
```

Out[70]:

```
Name        0
Team        0
Number      0
Position    0
Age         0
Height      0
Weight      0
College     0
Salary      0
dtype: int64
```

Statistical Analysis

In [71]:

```
df['Age'].describe()
```

Out[71]:

```
count    457.000000
mean      26.938731
std        4.404016
min       19.000000
25%       24.000000
50%       26.000000
75%       30.000000
max       40.000000
Name: Age, dtype: float64
```

In [72]:

```
df['Age'].value_counts()
```

Out[72]:

```
24.0    47
25.0    45
27.0    41
23.0    41
26.0    36
28.0    31
30.0    31
29.0    28
22.0    26
31.0    22
21.0    19
20.0    19
33.0    14
32.0    13
36.0    10
34.0    10
35.0     9
38.0     4
37.0     4
40.0     3
39.0     2
19.0     2
Name: Age, dtype: int64
```

In [73]:

```
bins= [11,21,31,41]
labels = ['11 to 20','21 to 30','31 to 40']
df['ageGroup'] = pd.cut(df['Age'], bins=bins,labels=labels, right=False)
print(df)
print(df['ageGroup'].unique())
```

```
              Name            Team  Number Position   Age Height  Weig
ht  \
0    Avery Bradley  Boston Celtics     0.0       PG  25.0    6-2   18
0.0
1      Jae Crowder  Boston Celtics    99.0       SF  25.0    6-6   23
5.0
2     John Holland  Boston Celtics    30.0       SG  27.0    6-5   20
5.0
3      R.J. Hunter  Boston Celtics    28.0       SG  22.0    6-5   18
5.0
4    Jonas Jerebko  Boston Celtics     8.0       PF  29.0   6-10   23
1.0
..             ...             ...     ...      ...   ...    ...
...
452     Trey Lyles       Utah Jazz    41.0       PF  20.0   6-10   23
4.0
453   Shelvin Mack       Utah Jazz     8.0       PG  26.0    6-3   20
3.0
454      Raul Neto       Utah Jazz    25.0       PG  24.0    6-1   17
9.0
455   Tibor Pleiss       Utah Jazz    21.0        C  26.0    7-3   25
6.0
456    Jeff Withey       Utah Jazz    24.0        C  26.0    7-0   23
1.0

               College        Salary  ageGroup
0                Texas  7.730337e+06  21 to 30
1            Marquette  6.796117e+06  21 to 30
2    Boston University  4.842684e+06  21 to 30
3        Georgia State  1.148640e+06  21 to 30
4             Kentucky  5.000000e+06  21 to 30
..                 ...           ...       ...
452           Kentucky  2.239800e+06  11 to 20
453             Butler  2.433333e+06  21 to 30
454           Kentucky  9.000000e+05  21 to 30
455           Kentucky  2.900000e+06  21 to 30
456             Kansas  9.472760e+05  21 to 30

[457 rows x 10 columns]
[21 to 30, 11 to 20, 31 to 40]
Categories (3, object): [11 to 20 < 21 to 30 < 31 to 40]
```

In [74]:

```
df['ageGroup'].value_counts()
```

Out[74]:

```
21 to 30    345
31 to 40     91
11 to 20     21
Name: ageGroup, dtype: int64
```

In [75]:

```
df.head(10)
```

Out[75]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	ageGrou
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7.730337e+06	21 to 3
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6.796117e+06	21 to 3
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	4.842684e+06	21 to 3
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1.148640e+06	21 to 3
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	Kentucky	5.000000e+06	21 to 3
5	Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	Kentucky	1.200000e+07	21 to 3
6	Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1.170960e+06	21 to 3
7	Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2.165160e+06	21 to 3
8	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1.824360e+06	21 to 3
9	Marcus Smart	Boston Celtics	36.0	PG	22.0	6-4	220.0	Oklahoma State	3.431040e+06	21 to 3

In [76]:

```
df.groupby(df['ageGroup']).get_group('11 to 20')
```

Out[76]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	ag
13	James Young	Boston Celtics	13.0	SG	20.0	6-6	215.0	Kentucky	1749840.0	
40	Kristaps Porzingis	New York Knicks	6.0	PF	20.0	7-3	240.0	Kentucky	4131720.0	
56	Jahlil Okafor	Philadelphia 76ers	8.0	C	20.0	6-11	275.0	Duke	4582680.0	
60	Christian Wood	Philadelphia 76ers	35.0	PF	20.0	6-11	220.0	UNLV	525093.0	
62	Bruno Caboclo	Toronto Raptors	20.0	SF	20.0	6-9	205.0	Kentucky	1524000.0	
85	Kevon Looney	Golden State Warriors	36.0	SF	20.0	6-9	220.0	UCLA	1131960.0	
116	D'Angelo Russell	Los Angeles Lakers	1.0	PG	20.0	6-5	195.0	Ohio State	5103120.0	
122	Devin Booker	Phoenix Suns	1.0	SG	19.0	6-6	206.0	Kentucky	2127840.0	
192	Stanley Johnson	Detroit Pistons	3.0	SF	20.0	6-7	245.0	Arizona	2841960.0	
208	Myles Turner	Indiana Pacers	33.0	PF	20.0	6-11	243.0	Texas	2357760.0	
226	Rashad Vaughn	Milwaukee Bucks	20.0	SG	19.0	6-6	202.0	UNLV	1733040.0	
352	Justise Winslow	Miami Heat	20.0	SF	20.0	6-7	225.0	Duke	2481720.0	
356	Aaron Gordon	Orlando Magic	0.0	PF	20.0	6-9	220.0	Arizona	4171680.0	
377	Kelly Oubre Jr.	Washington Wizards	12.0	SF	20.0	6-7	205.0	Kansas	1920240.0	
393	Emmanuel Mudiay	Denver Nuggets	0.0	PG	20.0	6-5	200.0	Kentucky	3102240.0	
401	Tyus Jones	Minnesota Timberwolves	1.0	PG	20.0	6-2	195.0	Duke	1282080.0	
410	Karl-Anthony Towns	Minnesota Timberwolves	32.0	C	20.0	7-0	244.0	Kentucky	5703600.0	
427	Cliff Alexander	Portland Trail Blazers	34.0	PF	20.0	6-8	240.0	Kansas	525093.0	
441	Noah Vonleh	Portland Trail Blazers	21.0	PF	20.0	6-9	240.0	Indiana	2637720.0	
445	Dante Exum	Utah Jazz	11.0	PG	20.0	6-6	190.0	Kentucky	3777720.0	
452	Trey Lyles	Utah Jazz	41.0	PF	20.0	6-10	234.0	Kentucky	2239800.0	

In [77]:

```
list_of_salaries_by_ageGroup = list(df.groupby('ageGroup')['Salary'])
list_of_salaries_by_ageGroup
```

Out[77]:

```
[('11 to 20',
  13     1749840.0
  40     4131720.0
  56     4582680.0
  60      525093.0
  62     1524000.0
  85     1131960.0
  116    5103120.0
  122    2127840.0
  192    2841960.0
  208    2357760.0
  226    1733040.0
  352    2481720.0
  356    4171680.0
  377    1920240.0
  393    3102240.0
  401    1282080.0
  410    5703600.0
  427     525093.0
  441    2637720.0
  445    3777720.0
  452    2239800.0
  Name: Salary, dtype: float64),
 ('21 to 30',
  0      7.730337e+06
  1      6.796117e+06
  2      4.842684e+06
  3      1.148640e+06
  4      5.000000e+06
             ...
  451    9.813480e+05
  453    2.433333e+06
  454    9.000000e+05
  455    2.900000e+06
  456    9.472760e+05
  Name: Salary, Length: 345, dtype: float64),
 ('31 to 40',
  19      6300000.0
  31      1635476.0
  33     22875000.0
  34      7402812.0
  43       947276.0
            ...
  406      947276.0
  413     3750000.0
  415     3135000.0
  420      222888.0
  434     5016000.0
  Name: Salary, Length: 91, dtype: float64)]
```

In [78]:

```
df.groupby('ageGroup')['Salary'].mean()
```

Out[78]:

```
ageGroup
11 to 20    2.650043e+06
21 to 30    4.679140e+06
31 to 40    5.968707e+06
Name: Salary, dtype: float64
```

In [79]:

```
df.groupby('ageGroup')['Salary'].median()
```

Out[79]:

```
ageGroup
11 to 20    2357760.0
21 to 30    2616975.0
31 to 40    4053446.0
Name: Salary, dtype: float64
```

In [80]:

```
df.groupby('ageGroup')['Salary'].describe()
```

Out[80]:

	count	mean	std	min	25%	50%	75%	
ageGroup								
11 to 20	21.0	2.650043e+06	1.454546e+06	525093.0	1733040.0	2357760.0	3777720.0	570
21 to 30	345.0	4.679140e+06	4.996786e+06	30888.0	1015421.0	2616975.0	6486486.0	2235
31 to 40	91.0	5.968707e+06	6.065991e+06	200600.0	1474187.0	4053446.0	7474380.0	2500

In [81]:

```
df['Height'].value_counts()
```

Out[81]:

```
6-9     59
6-10    47
6-7     45
6-8     43
6-6     42
6-11    40
6-3     33
6-5     32
6-4     29
7-0     27
6-1     16
6-2     16
6-0     10
7-1      7
7-3      4
7-2      3
5-11     3
5-9      1
Name: Height, dtype: int64
```

In [82]:

```
df.groupby(df['Height']).get_group('6-11')
```

Out[82]:

	Name	Team	Number	Position	Age	Height	Weight	College	Sa
24	Chris McCullough	Brooklyn Nets	1.0	PF	21.0	6-11	200.0	Syracuse	11402
55	Nerlens Noel	Philadelphia 76ers	4.0	PF	22.0	6-11	228.0	Kentucky	34578
56	Jahlil Okafor	Philadelphia 76ers	8.0	C	20.0	6-11	275.0	Duke	45826
60	Christian Wood	Philadelphia 76ers	35.0	PF	20.0	6-11	220.0	UNLV	5250
73	Jason Thompson	Toronto Raptors	1.0	PF	29.0	6-11	250.0	Rider	2451
81	Festus Ezeli	Golden State Warriors	31.0	C	26.0	6-11	265.0	Vanderbilt	20087
90	Anderson Varejao	Golden State Warriors	18.0	PF	33.0	6-11	273.0	Kentucky	2897
91	Cole Aldrich	Los Angeles Clippers	45.0	C	27.0	6-11	250.0	Kansas	11006
98	DeAndre Jordan	Los Angeles Clippers	6.0	C	27.0	6-11	265.0	Texas A&M	196890
113	Ryan Kelly	Los Angeles Lakers	4.0	PF	25.0	6-11	230.0	Duke	17242
143	DeMarcus Cousins	Sacramento Kings	15.0	C	25.0	6-11	270.0	Kentucky	158519
162	Joakim Noah	Chicago Bulls	13.0	C	31.0	6-11	232.0	Florida	134000
163	Bobby Portis	Chicago Bulls	5.0	PF	21.0	6-11	230.0	Arkansas	13911
167	Channing Frye	Cleveland Cavaliers	9.0	PF	33.0	6-11	255.0	Arizona	81930
173	Sasha Kaun	Cleveland Cavaliers	14.0	C	31.0	6-11	260.0	Kansas	12760
188	Andre Drummond	Detroit Pistons	0.0	C	22.0	6-11	279.0	Connecticut	32720
204	Ian Mahinmi	Indiana Pacers	28.0	C	29.0	6-11	250.0	Kentucky	40000
208	Myles Turner	Indiana Pacers	33.0	PF	20.0	6-11	243.0	Texas	23577
209	Shayne Whittington	Indiana Pacers	42.0	PF	25.0	6-11	250.0	Western Michigan	8450
211	Giannis Antetokounmpo	Milwaukee Bucks	34.0	SF	21.0	6-11	222.0	Kentucky	19539
216	John Henson	Milwaukee Bucks	31.0	PF	25.0	6-11	229.0	North Carolina	29432
220	Greg Monroe	Milwaukee Bucks	15.0	C	26.0	6-11	265.0	Georgetown	164075
224	Miles Plumlee	Milwaukee Bucks	18.0	C	27.0	6-11	249.0	Duke	21092

	Name	Team	Number	Position	Age	Height	Weight	College	Sa
237	Zaza Pachulia	Dallas Mavericks	27.0	C	32.0	6-11	275.0	Kentucky	52000
239	Dwight Powell	Dallas Mavericks	7.0	PF	24.0	6-11	240.0	Stanford	8450
240	Charlie Villanueva	Dallas Mavericks	3.0	PF	31.0	6-11	232.0	Connecticut	9472
251	Dwight Howard	Houston Rockets	12.0	C	30.0	6-11	265.0	Kentucky	223593
294	LaMarcus Aldridge	San Antonio Spurs	12.0	PF	30.0	6-11	240.0	Texas	196890
298	Tim Duncan	San Antonio Spurs	21.0	C	40.0	6-11	250.0	Wake Forest	52500
316	Mike Muscala	Atlanta Hawks	31.0	PF	24.0	6-11	240.0	Bucknell	9472
321	Tiago Splitter	Atlanta Hawks	11.0	C	31.0	6-11	245.0	Kentucky	97562
339	Chris Bosh	Miami Heat	1.0	PF	32.0	6-11	235.0	Georgia Tech	221927
373	Marcin Gortat	Washington Wizards	13.0	C	32.0	6-11	240.0	Kentucky	112173
375	Nene Hilario	Washington Wizards	42.0	C	33.0	6-11	250.0	Kentucky	130000
391	Joffrey Lauvergne	Denver Nuggets	77.0	C	24.0	6-11	220.0	Kentucky	17097
399	Gorgui Dieng	Minnesota Timberwolves	5.0	C	26.0	6-11	241.0	Louisville	14744
400	Kevin Garnett	Minnesota Timberwolves	21.0	PF	40.0	6-11	240.0	Kentucky	85000
405	Nikola Pekovic	Minnesota Timberwolves	14.0	C	30.0	6-11	307.0	Kentucky	121000
418	Enes Kanter	Oklahoma City Thunder	11.0	C	24.0	6-11	245.0	Kentucky	164075
439	Mason Plumlee	Portland Trail Blazers	24.0	C	26.0	6-11	235.0	Duke	14155

In [83]:

```
list_of_salaries_by_HeightGroup = list(df.groupby('Height')['Salary'])
list_of_salaries_by_HeightGroup
```

Out[83]:

```
[('5-11',
  22     1500000.0
  130      55722.0
  203     211744.0
  Name: Salary, dtype: float64),
 ('5-9',
  11    6912869.0
  Name: Salary, dtype: float64),
 ('6-0',
  47       947276.0
  57       947276.0
  67     12000000.0
  100    21468695.0
  142     5013559.0
  152     2250000.0
  228     4290000.0
  305     3578947.0
  384     3000000.0
```

In [84]:

```
df.groupby('Height')['Salary'].describe()
```

Out[84]:

	count	mean	std	min	25%	50%	75%	
Height								
5-11	3.0	5.891553e+05	7.926627e+05	55722.0	133733.0	211744.0	8.558720e+05	15
5-9	1.0	6.912869e+06	NaN	6912869.0	6912869.0	6912869.0	6.912869e+06	69
6-0	10.0	5.784075e+06	6.337144e+06	947276.0	2437500.0	3934473.5	4.846419e+06	214
6-1	16.0	5.217919e+06	4.286013e+06	700902.0	1646160.0	3402626.5	8.633373e+06	135
6-10	47.0	5.185375e+06	5.063120e+06	222888.0	1054584.5	3815000.0	7.025766e+06	196
6-11	40.0	6.544397e+06	6.906416e+06	245177.0	1362370.0	3107656.0	1.143804e+07	223
6-2	16.0	3.523777e+06	3.631376e+06	525093.0	947276.0	1553220.0	4.882013e+06	134
6-3	33.0	5.821784e+06	5.668225e+06	189455.0	1662360.0	4053446.0	8.000000e+06	200
6-4	29.0	4.646163e+06	5.275308e+06	134215.0	1015421.0	2525160.0	5.192520e+06	200
6-5	32.0	4.391786e+06	4.114296e+06	55722.0	1160040.0	3129420.0	6.015152e+06	164
6-6	42.0	3.586813e+06	4.518975e+06	167406.0	955794.0	1903380.0	4.317674e+06	250
6-7	45.0	3.504402e+06	4.337857e+06	30888.0	947276.0	1535880.0	4.000000e+06	164
6-8	43.0	5.950412e+06	6.133934e+06	83397.0	1259700.0	3425510.0	9.321234e+06	229
6-9	59.0	4.157787e+06	4.517154e+06	111444.0	1053814.0	2500000.0	5.250000e+06	201
7-0	27.0	5.287712e+06	4.675298e+06	947276.0	2003580.0	4204200.0	7.574380e+06	196
7-1	7.0	7.400988e+06	6.587462e+06	1175880.0	3441500.0	4950000.0	9.555017e+06	196
7-2	3.0	6.835639e+06	7.825718e+06	525093.0	2457350.0	4389607.0	9.990912e+06	155
7-3	4.0	2.307930e+06	1.484918e+06	1000000.0	1150000.0	2050000.0	3.207930e+06	41

Part2 - IRIS Dataset

In [85]:

```
df1 = pd.read_csv("C:/Users/Admin/Desktop/Data3/iris.csv")
```

In [86]:

```
df1.shape
```

Out[86]:

```
(150, 6)
```

In [87]:

```
df1.size
```

Out[87]:

```
900
```

In [88]:

```
df1.columns
```

Out[88]:

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalW
idthCm',
       'Species'],
      dtype='object')
```

In [89]:

```
df1.head(10)
```

Out[89]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [90]:

```
df1.describe()
```

Out[90]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [91]:

```
df1.dtypes
```

Out[91]:

```
Id                 int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [92]:

```
df1.isnull()
```

Out[92]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
145	False	False	False	False	False	False
146	False	False	False	False	False	False
147	False	False	False	False	False	False
148	False	False	False	False	False	False
149	False	False	False	False	False	False

150 rows × 6 columns

In [93]:

```
df1.isnull().sum()
```

Out[93]:

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [94]:

```
df1['Species'].unique()
```

Out[94]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=obje
ct)
```

In [95]:

```
print('Iris-setosa')
Iris_setos=(df1['Species']=='Iris-setosa')
df1[Iris_setos].describe()
```

```
Iris-setosa
```

Out[95]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.000000	50.000000	50.00000
mean	25.50000	5.00600	3.418000	1.464000	0.24400
std	14.57738	0.35249	0.381024	0.173511	0.10721
min	1.00000	4.30000	2.300000	1.000000	0.10000
25%	13.25000	4.80000	3.125000	1.400000	0.20000
50%	25.50000	5.00000	3.400000	1.500000	0.20000
75%	37.75000	5.20000	3.675000	1.575000	0.30000
max	50.00000	5.80000	4.400000	1.900000	0.60000

In [96]:

```
print('Iris-versicolor')
Iris_versicolor=(df1['Species']=='Iris-versicolor')
df1[Iris_versicolor].describe()
```

```
Iris-versicolor
```

Out[96]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	75.50000	5.936000	2.770000	4.260000	1.326000
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.900000	2.000000	3.000000	1.000000
25%	63.25000	5.600000	2.525000	4.000000	1.200000
50%	75.50000	5.900000	2.800000	4.350000	1.300000
75%	87.75000	6.300000	3.000000	4.600000	1.500000
max	100.00000	7.000000	3.400000	5.100000	1.800000

In [97]:

```
print('Iris-virginica')
Iris_virginica=(df1['Species']=='Iris-virginica')
df1[Iris_virginica].describe()
```

```
Iris-virginica
```

Out[97]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.000000	50.000000	50.00000
mean	125.50000	6.58800	2.974000	5.552000	2.02600
std	14.57738	0.63588	0.322497	0.551895	0.27465
min	101.00000	4.90000	2.200000	4.500000	1.40000
25%	113.25000	6.22500	2.800000	5.100000	1.80000
50%	125.50000	6.50000	3.000000	5.550000	2.00000
75%	137.75000	6.90000	3.175000	5.875000	2.30000
max	150.00000	7.90000	3.800000	6.900000	2.50000