EDA Visualisation and Preprosessing

Name : Nitesh R

USN : 1BM19EC099

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df1=pd.read_csv("MLaatdataset.csv")
```

My split dataset is from rows 946 to 1441

```
df2=df1[946:1441]
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
946	22	Male	Computer Science	2	7	0	0	1
947	21	Male	Computer Science	1	8	0	0	1
948	21	Male	Information Technology	0	7	0	1	1
949	22	Male	Computer Science	0	7	0	0	1
950	22	Female	Information Technology	0	8	0	0	1

description of thresholds of all attributes

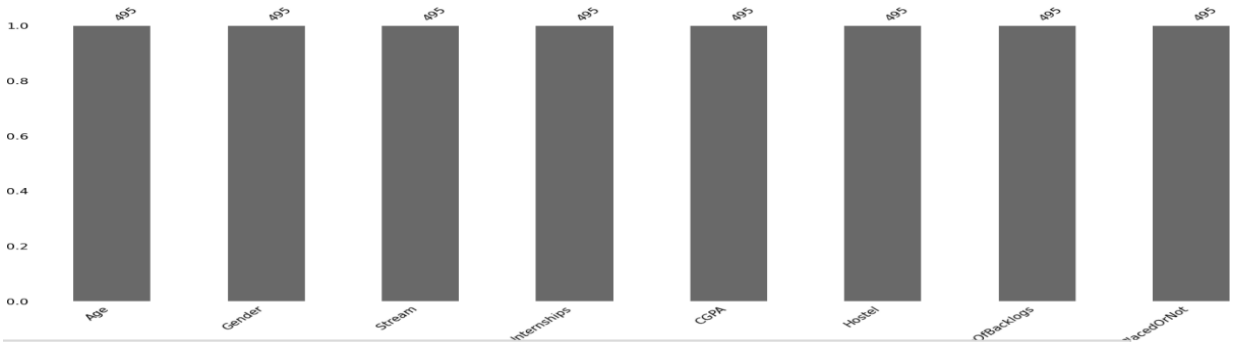
```
df2.describe()
```

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	495.000000	495.000000	495.000000	495.000000	495.000000	495.000000
mean	21.971717	0.870707	7.385859	0.294949	0.214141	0.642424
std	1.367349	0.752493	0.976755	0.456481	0.410640	0.479771
min	21.000000	0.000000	6.000000	0.000000	0.000000	0.000000
25%	21.000000	0.000000	7.000000	0.000000	0.000000	0.000000
50%	22.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	22.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	26.000000	3.000000	9.000000	1.000000	1.000000	1.000000

checking for missing values

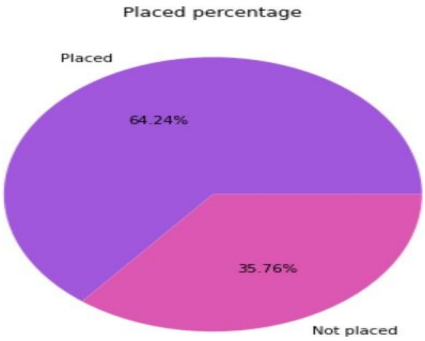
```
import missingno as msno
msno.bar(df2)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f81e9ed1450>



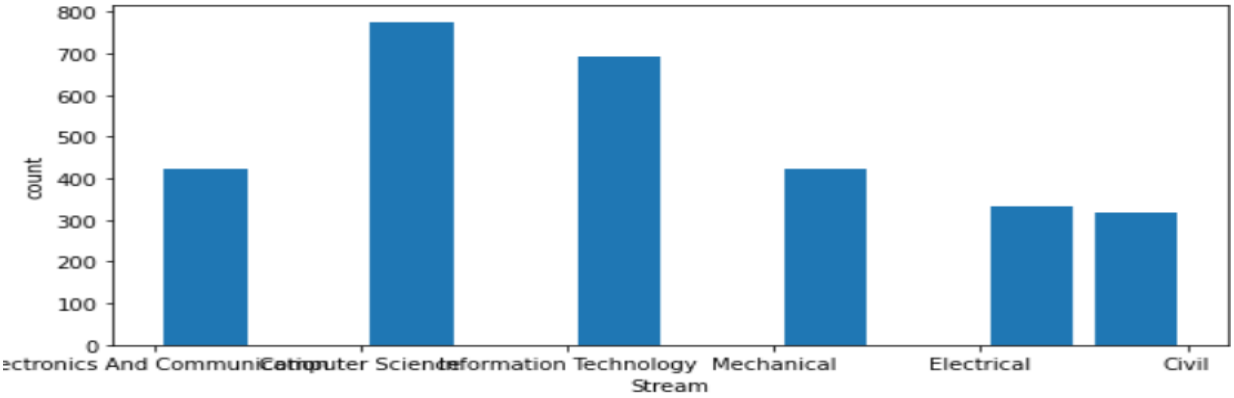
placed percentage using pie chart

```
plt.figure(figsize=(6, 6))
classx = ['Placed','Not placed']
plt.title('Placed percentage')
colors = sns.color_palette("hls", 8)[6:8]
countx = [len(df2[df2.PlacedOrNot == 1]),len(df2[df2.PlacedOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```



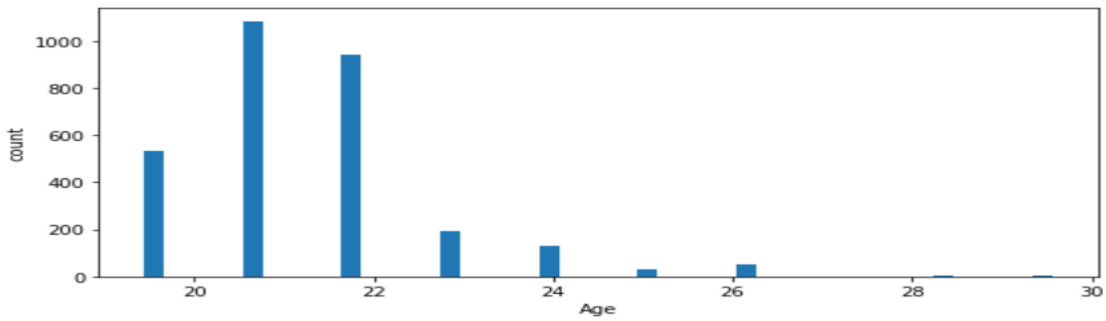
#Analysing number of people in each stream, 750 are from computer science

```
matplotlib.rcParams['figure.figsize']=(9,4)
plt.hist(df1.Stream,rwidth=0.8)
plt.xlabel("Stream")
plt.ylabel('count')
```



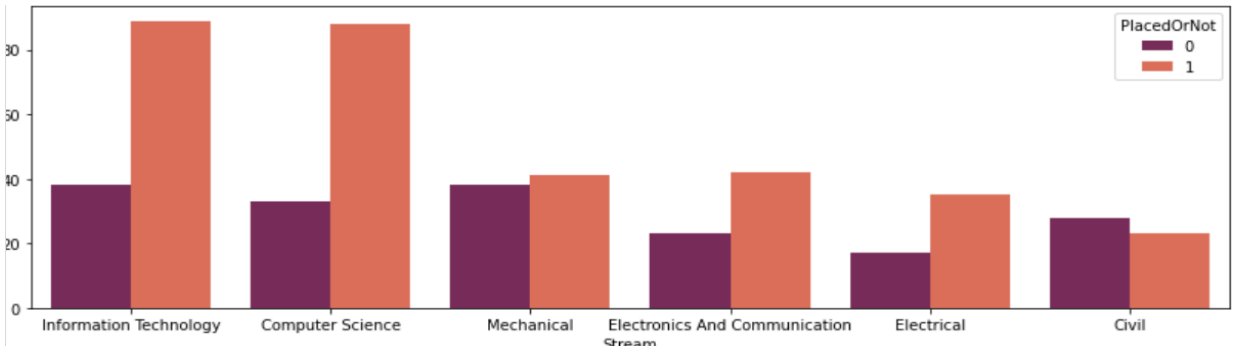
Analysing the people ages, half of them are of the age of 20-22

```
matplotlib.rcParams['figure.figsize']=(9,4)
plt.hist(df1.Age,rwidth=0.2)
plt.xlabel("Age")
plt.ylabel('count')
```



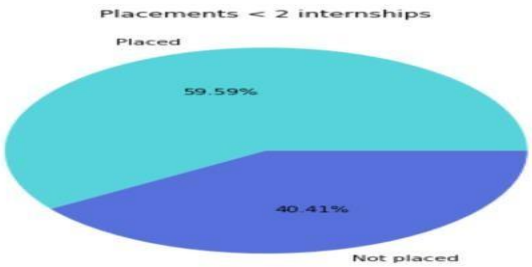
placement check with respect to every branch cs and Is see more placements

```
fig, ax = plt.subplots(figsize=(20,7))
sns.countplot(data=df2,x='Stream', order = df2['Stream'].value_counts().index,palette='rocket',hue='PlacedOrNot')
plt.xticks(rotation=0)
plt.show()
```



Analysing if people with 2 internships are placed or not 59.9% are placed

```
plt.figure(figsize=(6, 6))
classx = ['Placed','Not placed']
plt.title('Placements < 2 internships')
colors = sns.color_palette("hls", 8)[4:8]
dataFrame2 = df2[df2.Internships < 2]
countx = [len(dataFrame2[dataFrame2.PlacedOrNot == 1]),len(dataFrame2[dataFrame2.PlacedOrNot == 0])]
```



Conclusion : Our dataset has negligible null values and categorical data is analysed using labelencoder

And detail analysis is done wrt every attribute checking performance detail prediction is done in phase 2

PHASE -1

Name:Pavan Kumar M
USN:1BM19EC102

EXPLORATORY DATA ANALYSIS

- After importing all the required libraries, in any EDA the first step is to understand how the data is distributed such as what are input attributes and what is the target attribute and this can be done by reading first 5 entries of dataset as shown below:

```
In [3]: dataFrame = pd.read_csv('EngineeringPlacementDataset.csv')
        dataFrame.head()
```

Out[3]:

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

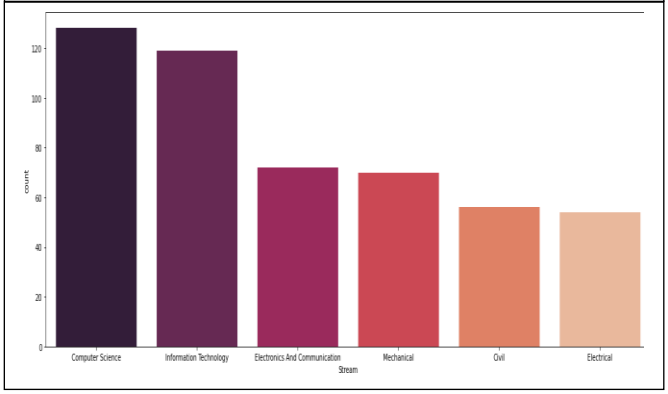
Conclusion:

After performing the EDA following Conclusions were drawn:

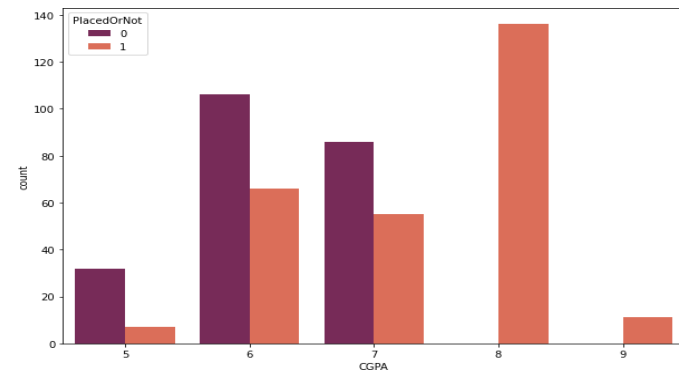
- The sub dataset I picked has 499 rows and 8 features.
- Before proceeding onto training it is very much important to know is there any null values in the dataset and when I perfomed dataFrame.isnull().sum() the result was found out to be there was no null, NAN values. Hence dataset is clean.
- Then using describe () function statistical parameters are calculated which helps in predicting result as accurate as possible.
- From the graphs it is observed that Computer science and Information technology branch has the highest placement opportunity while Civil being the least. Electrical, Electronics & Communication Engineering has decent placement opportunity.
- The number of students getting placed and not placed is almost equal hence dataset is balanced.
- In deciding the Placement prediction the internship also plays an important role.
- From the Pie chart 2 it is observed that nearly 47% of students haven't done any internship, nearly 39% of students have done at least one Internship, nearly 11.5% of students completed two internships, and less than 2.2% of students have completed three internships.
- Nearly 20% of students who are sitting for placements have active backlog.
- Nearly 27% of students stays in hostel remaining 73% lives off campus
- In a extremely rare case it is also found few students who are at the age of 23 & 24
- Nearly 84% of the students who are sitting for placements are Male and while 16%
- Students who had and cleared backlog also got placed before the end of graduation.
- Internships & CGPA directly influences on the Placements. For ex: For ex: The placement percentage of students who did less than 2 Internship is 51.63% while those who did more than 2 Internships the placement percentage is76.81%. There is a drastic change in terms of placement opportunity.
- If a student gets a CGPA of 5 then getting placed percentage is very less.
- If a student gets CGPA between 8 and 9 almost all of them got placed.
- CGPA between 6 and 7 the placement percentage is pretty decent.

Visualization:

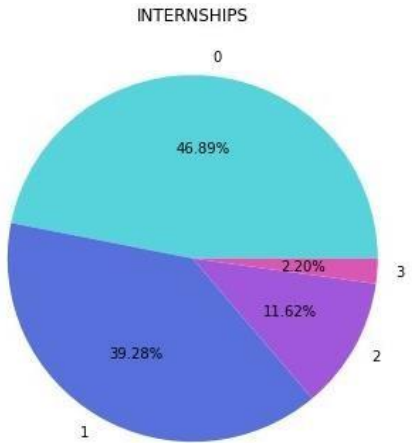
Plot of Placement opportunity with respect to Branches



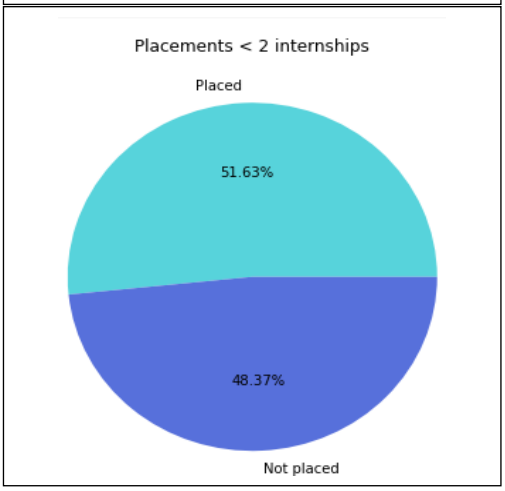
Plot for checking CGPA vs Placed or not



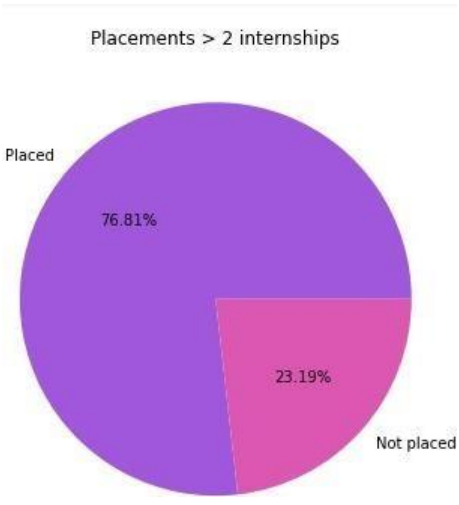
Pie chart of Students vs Number of Internships they did



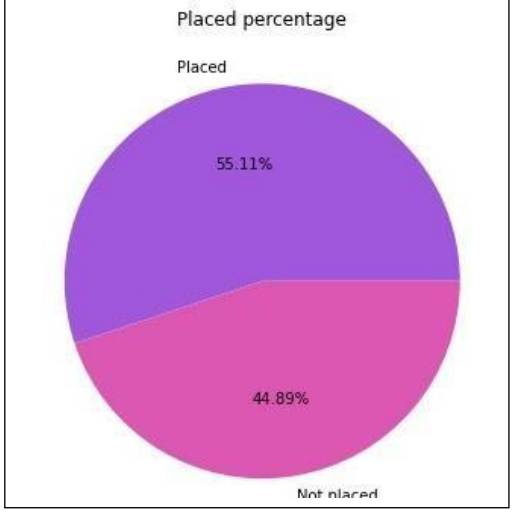
People who did 1 Internship and got placed



People who did more than 2 Internships and got placed



Pie chart showing percentage of student who got placed and not placed



EDA

Nandeesh
1BM19EC087

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df1=pd.read_csv("collegePlace.csv")
from matplotlib import pyplot as plt
matplotlib.rcParams["figure.figsize"]=(20,10)
```

#My data set is from row 2001 to 2967

```
df2=df1[2001:2967] df2
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
2001	19	Male	Electronics And Communication	0	8	0	0	1
2002	21	Male	Electrical	1	8	0	0	1
2003	19	Male	Computer Science	0	8	0	0	1
2004	19	Male	Computer Science	1	8	0	0	1
2005	22	Male	Computer Science	0	7	0	0	0

 # checking the maximum minimum and mean of all attributes

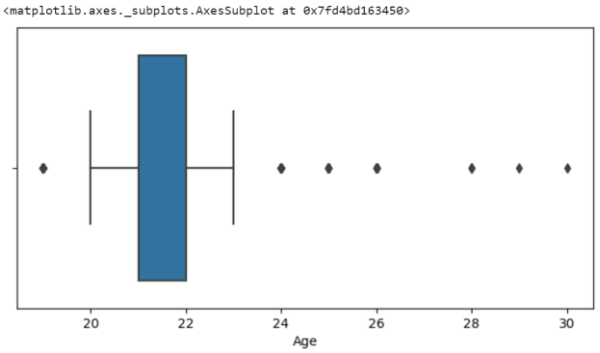
```
df2.describe()
```

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	965.000000	965.000000	965.000000	965.000000	965.000000	965.000000
mean	20.740933	0.644560	7.156477	0.267358	0.196891	0.512953
std	1.160033	0.740839	0.960513	0.442810	0.397856	0.500091
min	19.000000	0.000000	5.000000	0.000000	0.000000	0.000000
25%	20.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	21.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	21.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	29.000000	3.000000	9.000000	1.000000	1.000000	1.000000

```
df2.shape
```

(965, 8)

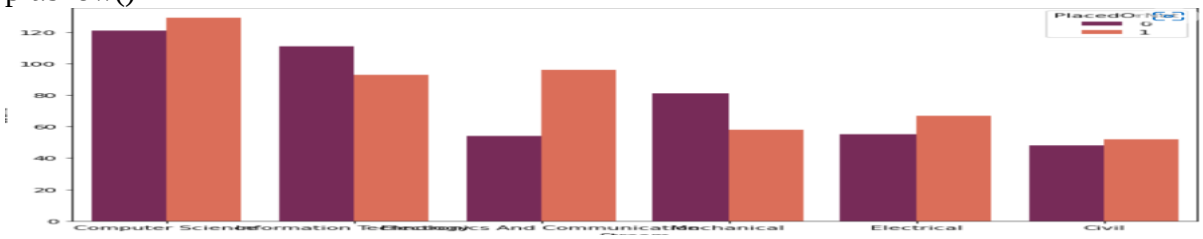
```
import seaborn as sns
plt.figure(figsize = (10, 6), dpi = 100)
sns.boxplot(x = "Age", data = df1)
```



 # placed statistics of each branch

```
fig, ax = plt.subplots(figsize=(20,7))
sns.countplot(data=df2,x='Stream', order = df2['Stream'].value_counts().index,palette='rocket'
,hue='PlacedOrNot')
```

```
plt.xticks(rotation=
plt.show()
```



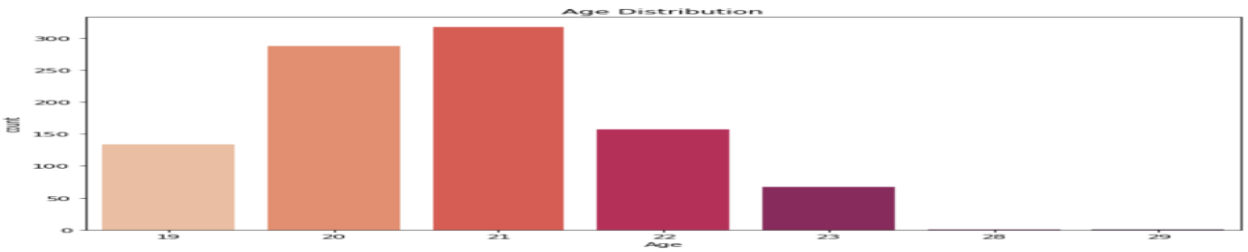
 # Checking count of all Branches

```
Stream_stats=df1.groupby('Stream')['Stream'].agg('count').sort_values(ascending=False)
Stream_stats
```

```
Stream
Computer Science                776
Information Technology          691
Electronics And Communication   424
Mechanical                      424
Electrical                      334
Civil                           317
Name: Stream, dtype: int64
```

 # age distribution

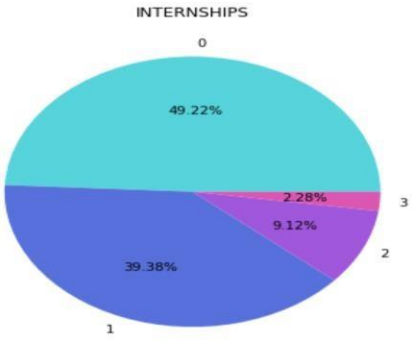
```
fig, ax = plt.subplots(figsize=(10,7))
sns.countplot(df2['Age'],palette='rocket_r')
plt.title('Age Distribution')
plt.show()
```

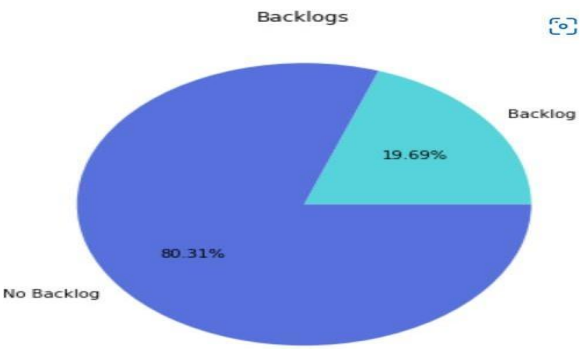



```
plt.figure(figsize=(6, 6))
classx = ['Male','Female']
plt.title('Gender percentage')
colors = sns.color_palette("hls", 8)[6:8]
countx = [len(df2[df2.Gender == 'Male']),len(df2[df2.Gender == 'Female'])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```

#Intership statistics

```
 plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```



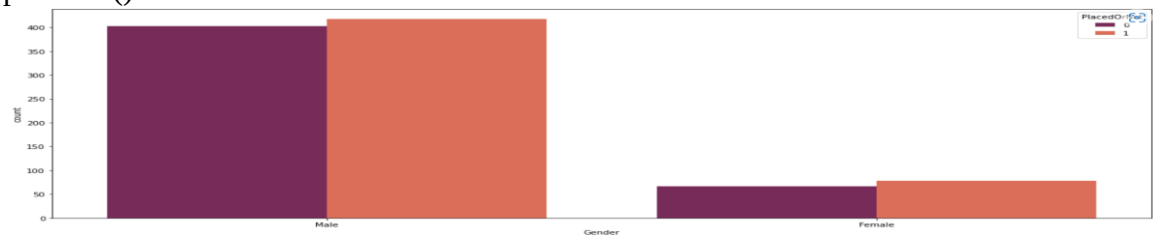


 # people with backlogs 19.69%

```
plt.figure(figsize=(6, 6))
classx = ['Backlog','No Backlog']
plt.title('Backlogs')
colors = sns.color_palette("hls", 8)[4:8]
countx = [len(df2[df2.HistoryOfBacklogs == 1]),len(df2[df2.HistoryOfBacklogs == 0])]
plt.show()
```

 # gender based placement count in dataset

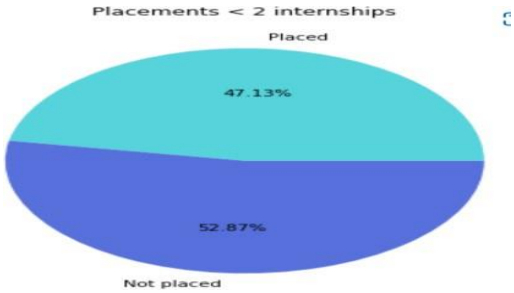
```
plt.xticks(rotation=0)
plt.show()
```



placecement analysis 47 %placed 53% not placed with 2 internships

```
plt.figure(figsize=(6, 6))

classx = ['Placed','Not placed']
plt.title('Placements < 2 internships')
colors = sns.color_palette("hls", 8)[4:8]
dataFrame2 = df2[df2.Internships < 2]
countx = [len(dataFrame2[dataFrame2.PlacedOrNot == 1]),len(dataFrame2[dataFrame2.Plac
edOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```



Name: Padmaj U Naik USN:1BM19EC101

PHASE -1

EXPLORATORY DATA ANALYSIS

- After importing all the required libraries, in any EDA the first step is to understand howthe data is distributed and this can be done by reading first 5 entries of dataset as shownbelow:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
dataFrame = pd.read_csv('EngineeringPlacementDataset.csv')
dataFrame.head()
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	21	Male	Information Technology	0	7	0	1	0
1	22	Male	Information Technology	1	6	0	0	0
2	22	Female	Computer Science	1	6	1	0	0
3	21	Male	Electrical	1	7	0	0	1
4	28	Female	Computer Science	3	8	1	0	1

- Next step is to check the shape of the dataset which will provide number of rows andfeature in a chosen subset. And found that I have 500 rows 8 features.

```
[ ] dataFrame.shape

    (500, 8)
```

- Next used the .info() method to check the data type of each feature and found followingresults:

```
dataFrame.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 500 non-null    int64
 1   Gender              500 non-null    object
 2   Stream              500 non-null    object
 3   Internships         500 non-null    int64
 4   CGPA                500 non-null    int64
 5   Hostel              500 non-null    int64
 6   HistoryOfBacklogs   500 non-null    int64
 7   PlacedOrNot         500 non-null    int64
dtypes: int64(6), object(2)
memory usage: 31.4+ KB
```

- Before proceeding onto training it is very much important to know is there null values inthe dataset and when I performed dataFrame.isnull().sum() the result was there are none.
- Then using describe() function founded few important statistical parameters which helpsin Predicting result as accurate as possible.

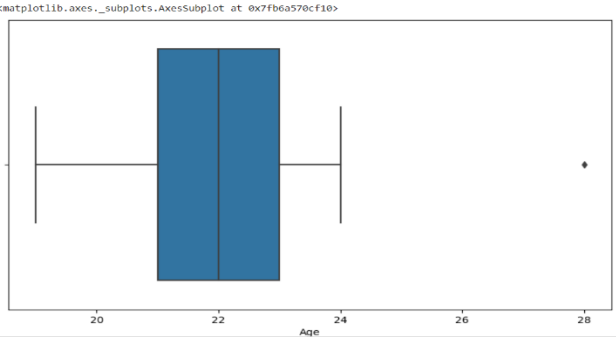
```
[ ]  dataFrame.describe()
```

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.0000
mean	21.896000	0.608000	6.782000	0.260000	0.192000	0.4800
std	1.334446	0.692311	0.955144	0.439074	0.394268	0.5001
min	19.000000	0.000000	5.000000	0.000000	0.000000	0.0000
25%	21.000000	0.000000	6.000000	0.000000	0.000000	0.0000
50%	22.000000	1.000000	7.000000	0.000000	0.000000	0.0000
75%	23.000000	1.000000	8.000000	1.000000	0.000000	1.0000
max	28.000000	3.000000	8.000000	1.000000	1.000000	1.0000

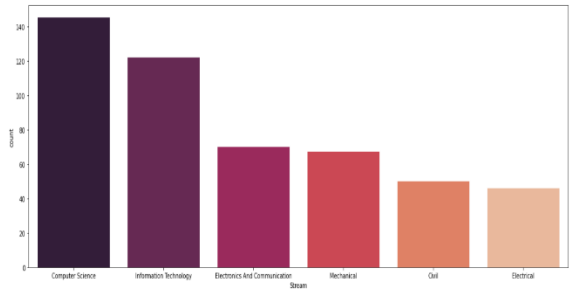
Conclusion: From the above output it can be concluded that the age of the students whoare sitting for placement is between 21-22.

```
plt.figure(figsize = (10, 6), dpi = 100)
sns.boxplot(x = "Age", data = dataFrame)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb6a570cf10>
```

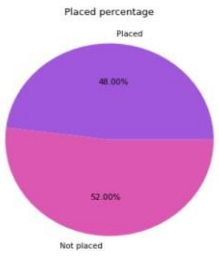



```
fig, ax = plt.subplots(figsize=(20,7))
sns.countplot(data=dataFrame,x='Stream', order = dataFrame['Stream'].value_counts().index,palette='rocket')
plt.xticks(rotation=0)
plt.show()
```

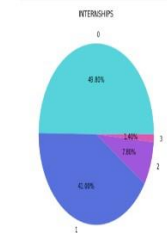


- **Conclusion:**
  - It can be observed that maximum students who are getting placed is from Computer science and Information technology while Electrical being the least. Electronics & Communication engineering having decent number of studentsgetting placed.

```
plt.figure(figsize=(6, 6))
classx = ['Placed','Not placed']
plt.title('Placed percentage')
colors = sns.color_palette("hls", 8)[6:8]
countx = [len(dataFrame[dataFrame.PlacedOrNot == 1]),len(dataFrame[dataFrame.PlacedOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```

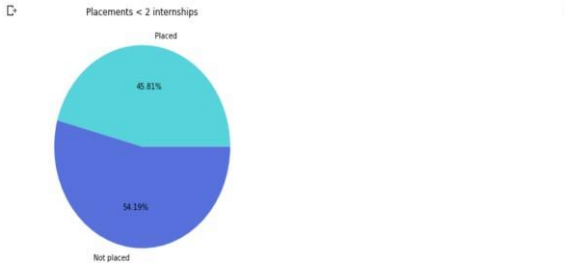



```
plt.figure(figsize=(6, 6))
classx = ['0','1','2','3']
plt.title('INTERNSHIPS')
colors = sns.color_palette("hls", 8)[6:8]
countx = [len(dataFrame[dataFrame.Internships == 0]),len(dataFrame[dataFrame.Internships == 1]),len(dataFrame[dataFrame.Internships==2]),len(dataFrame[dataFrame.Internships==3])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```

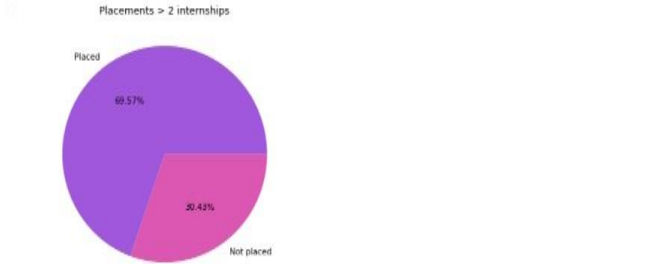


  - The number of students getting placed and not placed is almost equal hence thedataset is balanced.
- Let's see how the internship/s plays a key role in determining the placement predictionof a Graduate.
- **Conclusion:**
  - From the above pie chart, it is clear that Nearly 50% of students haven't done anyinternship
  - Nearly 41% of students have done at least one Internship
  - Nearlly 7.8% of students have completed two Internships.
  - And very less i.e. 1.4% of students completed 3 Internships as well.

```
plt.figure(figsize=(6, 6))
classx = ['Placed','Not placed']
plt.title('Placements < 2 internships')
colors = sns.color_palette("hls", 8)[4:8]
dataFrame2 = dataFrame[dataFrame.Internships < 2]
countx = [len(dataFrame2[dataFrame2.PlacedOrNot == 1]),len(dataFrame2[dataFrame2.PlacedOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```




```
plt.figure(figsize=(6, 6))
classx = ['Placed','Not placed']
plt.title('Placements > 2 internships')
colors = sns.color_palette("hls", 8)[6:8]
dataFrame2 = dataFrame[dataFrame.Internships >= 2]
countx = [len(dataFrame2[dataFrame2.PlacedOrNot == 1]),len(dataFrame2[dataFrame2.PlacedOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```

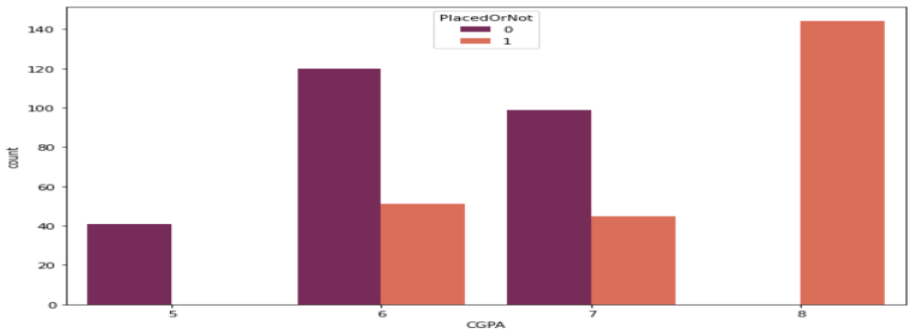


❖ **Conclusion:**

- From the above two figures it is clear the number of Internship/s completed bythe student will directly influence on Placements.
- For ex: The placement percentage of students who did less than 2 Internship is 45.81% while those who did more than 2 Internships the placement percentage is69.57%. There is a drastic change in terms of placement opportunity.

❖ Let's check how the CGPA influences the Placements

```
fig, ax = plt.subplots(figsize=(10,7))
sns.countplot(data=dataFrame,x='CGPA',palette='rocket',hue='PlacedOrNot')
plt.show()
```



❖ **From the above graph it can be concluded that:**

- If a student gets a CGPA of 5 then getting placed is zero percentage.
- If a student gets CGPA between 8 and 9 almost all of them got placed.
- CGPA between 6 and 7 the placement percentage is pretty decent

▪ **All other conclusions in short:**

- Nearly 20% of students who are sitting for placements have active backlog.
- Nearly 27% of students stays in hostel remaining 73% lives off campus
- In a extremely rare case it is also found few students who are at the age of 23 & 24
- Maximum students are getting placed from Computer science and IT, Electrical beingthe least and remaining branches having decent placements.
- Nearly 82% of the students who are sitting for placements are Male and while 18%are females.
- Students who had and cleared backlog also got placed before the end ofgraduation.
- Internships & CGPA directly influences on the Placements.

Name: Nishit Kumar
USN: 1BM19EE077

PHASE -1: EXPLORATORY DATA ANALYSIS

After importing all the required libraries, in any EDA the first step is to understand how the data is distributed such as what are input attributes and what is the target attribute and this can be done by reading first 5 entries of dataset as shown below:

```
[3] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
```

Lets load the Dataset first

```
[4] dataFrame = pd.read_csv('college_Place.csv')
    dataFrame.head()
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Mechanical	0	7	1	1	0
1	22	Male	Mechanical	2	7	1	0	1
2	22	Male	Mechanical	0	7	1	0	1
3	21	Male	Computer Science	0	7	0	0	0
4	22	Male	Computer Science	1	7	0	0	1

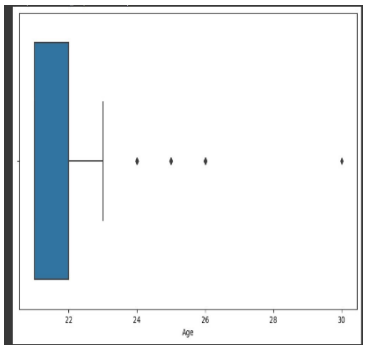
Conclusion:

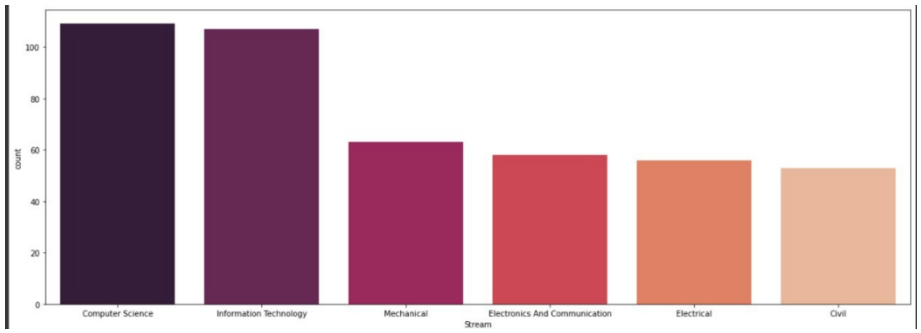
After performing the EDA following Conclusions were drawn:

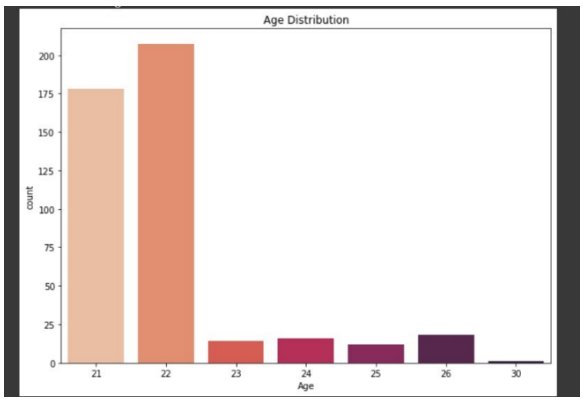
- The sub dataset I picked has 500-945 rows and 8 features.
- Before proceeding onto training it is very much important to know is there any null values in the dataset and when I performed data Frame.is null().sum() the result was found out to be there was no null, NAN values. Hence dataset is clean.
- Then using describe () function statistical parameters are calculated which helps in predictingresult as accurate as possible

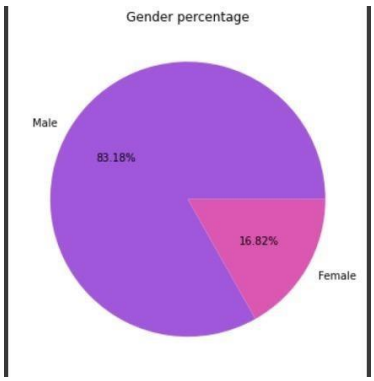
```
[8] dataFrame.describe()
```

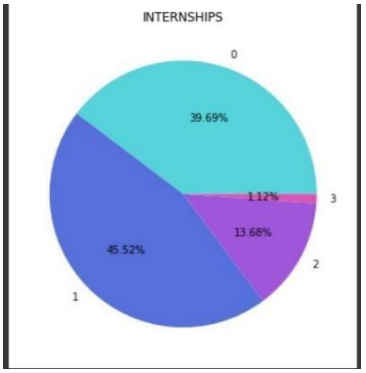
	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	446.000000	446.000000	446.000000	446.000000	446.000000	446.000000
mean	21.964126	0.762332	7.100897	0.253363	0.154709	0.607623
std	1.267951	0.723300	0.832567	0.435426	0.362033	0.488828
min	21.000000	0.000000	6.000000	0.000000	0.000000	0.000000
25%	21.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	22.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	22.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	30.000000	3.000000	9.000000	1.000000	1.000000	1.000000

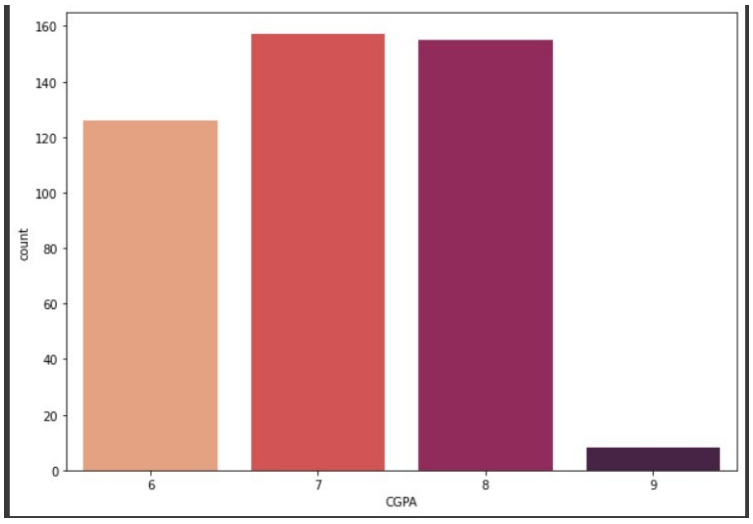


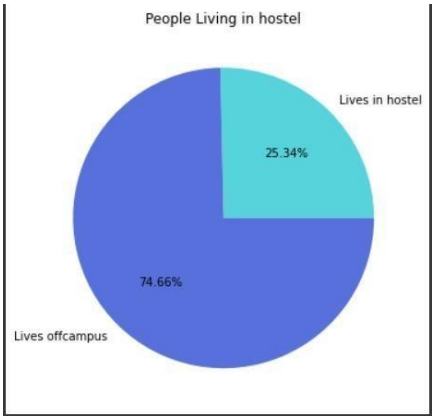






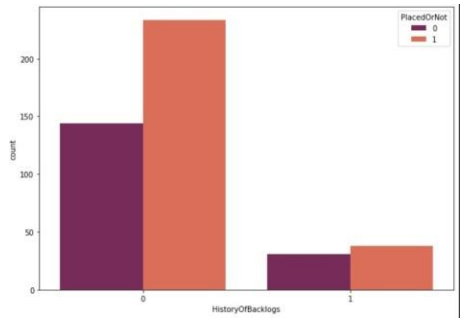


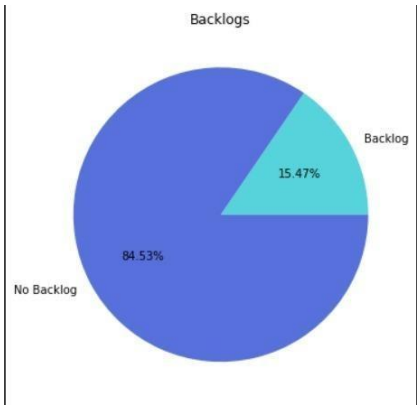


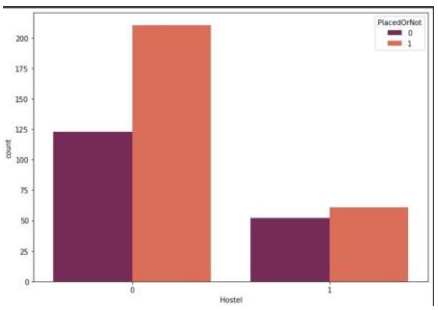


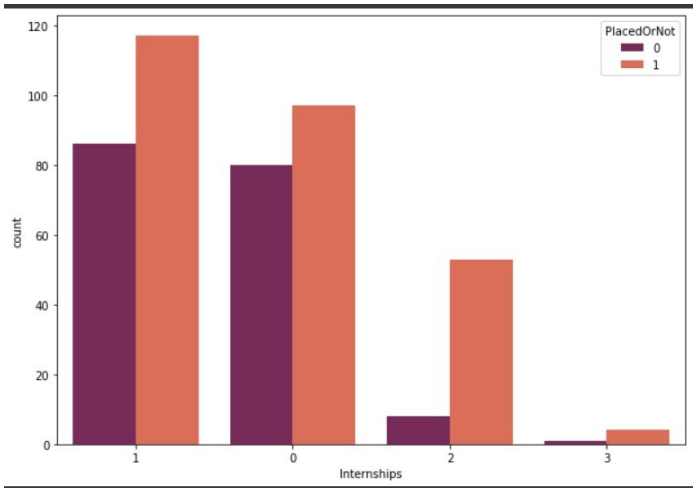
Conclusion:

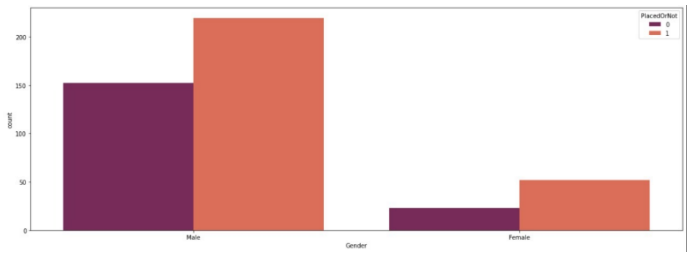
- Electronics & Communication engineering and Electrical almost have the same Competition.
- Almost 61% students are placed.
- Most of the students who are applying for placements are between the age of 21-22.
- Most of the Engineering students are Male.
- Nearly 46% of the students have not done any Internship
- Around 40% students have done atleast one Internship.
- Nearly 14% of the students have done 2 Internships.
- Less than 1% of the students have done 3 Internships
- The CGPA of the students who are sitting for placements ranges between 6-8.
- Very few got CGPA of 9.

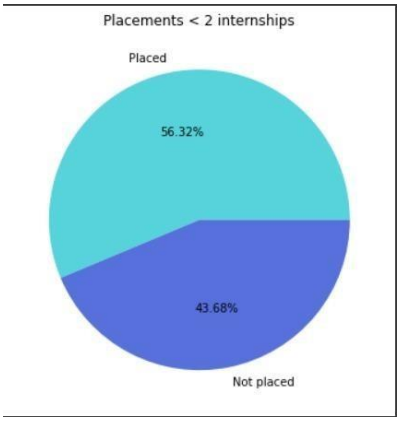


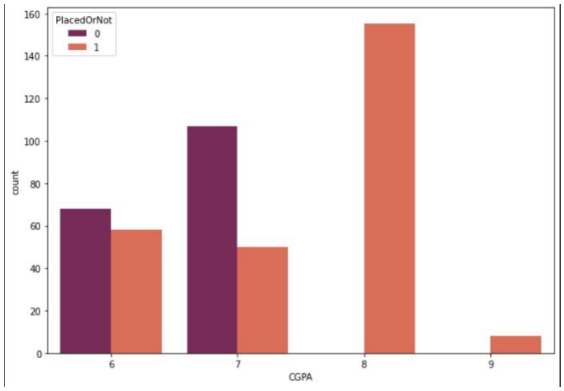


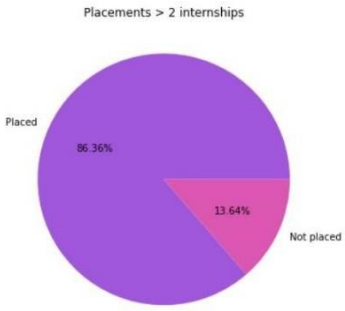












Conclusion:

- Only 25% of the students lives in hostel near to the college premises.
- Nearly 15% of the students sitting for placements have backlog/s.
- Here nearly 86% of the students who did Internships got placed.
- Almost all hostel students got placed. Hence Placement won't affect them much.
- All those who got CGPA above & including 8 got placed
- Almost all hostel students got placed. Hence Placement won't affect them much.
- Those who had Backlog/s also got placed hence even if a student fails in exams he/she should believe in themself and should be positive.

EDA

NiteeshKumar H V
1BM19EC098

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df1=pd.read_csv("collegePlace.csv")
from matplotlib import pyplot as plt
matplotlib.rcParams["figure.figsize"]=(20,10)
```

#My data set is from row 2001 to 2967

```
df2=df1[2001:2967] df2
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
2001	19	Male	Electronics And Communication	0	8	0	0	1
2002	21	Male	Electrical	1	8	0	0	1
2003	19	Male	Computer Science	0	8	0	0	1
2004	19	Male	Computer Science	1	8	0	0	1
2005	22	Male	Computer Science	0	7	0	0	0

 # checking the maximum minimum and mean of all attributes

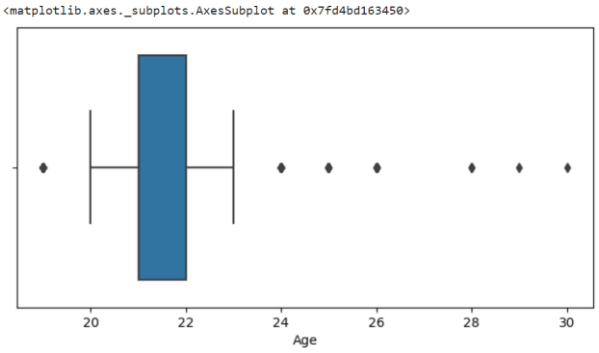
```
df2.describe()
```

	Age	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	965.000000	965.000000	965.000000	965.000000	965.000000	965.000000
mean	20.740933	0.644560	7.156477	0.267358	0.196891	0.512953
std	1.160033	0.740839	0.960513	0.442810	0.397856	0.500091
min	19.000000	0.000000	5.000000	0.000000	0.000000	0.000000
25%	20.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	21.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	21.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	29.000000	3.000000	9.000000	1.000000	1.000000	1.000000

```
df2.shape
```

(965, 8)

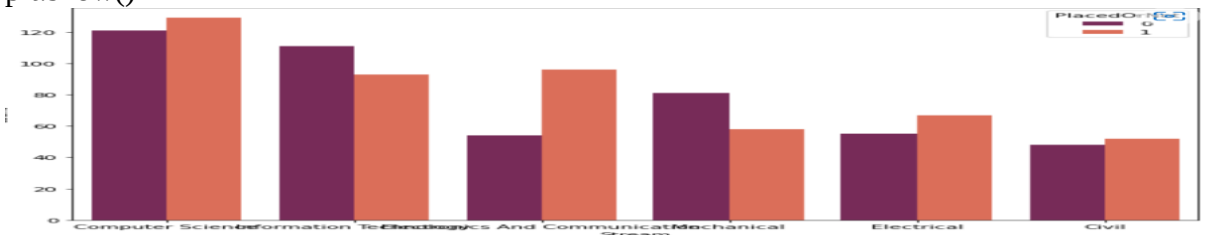
```
import seaborn as sns
plt.figure(figsize = (10, 6), dpi = 100)
sns.boxplot(x = "Age", data = df1)
```



 # placed statistics of each branch

```
fig, ax = plt.subplots(figsize=(20,7))
sns.countplot(data=df2,x='Stream', order = df2['Stream'].value_counts().index,palette='rocket'
,hue='PlacedOrNot')
```

```
plt.xticks(rotation=
plt.show()
```



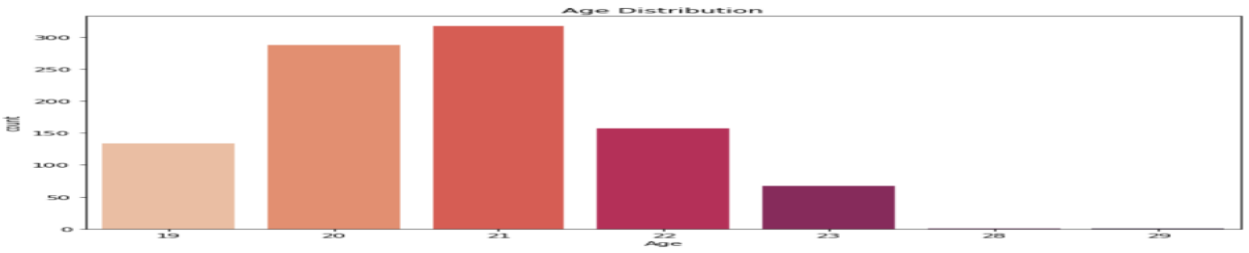
 # Checking count of all Branches

```
Stream_stats=df1.groupby('Stream')['Stream'].agg('count').sort_values(ascending=False)
Stream_stats
```

```
Stream
Computer Science                776
Information Technology          691
Electronics And Communication   424
Mechanical                      424
Electrical                      334
Civil                           317
Name: Stream, dtype: int64
```

 # age distribution

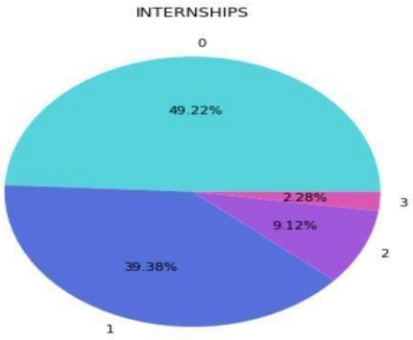
```
fig, ax = plt.subplots(figsize=(10,7))
sns.countplot(df2['Age'],palette='rocket_r')
plt.title('Age Distribution')
plt.show()
```

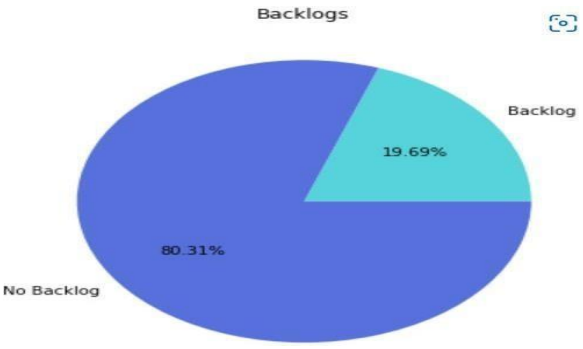



```
plt.figure(figsize=(6, 6))
classx = ['Male','Female']
plt.title('Gender percentage')
colors = sns.color_palette("hls", 8)[6:8]
countx = [len(df2[df2.Gender == 'Male']),len(df2[df2.Gender == 'Female'])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```

#Intership statistics

```
 plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```



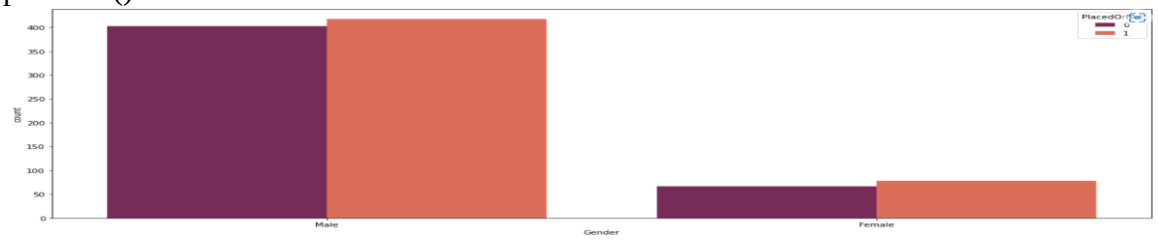


 # people with backlogs 19.69%

```
plt.figure(figsize=(6, 6))
classx = ['Backlog','No Backlog']
plt.title('Backlogs')
colors = sns.color_palette("hls", 8)[4:8]
countx = [len(df2[df2.HistoryOfBacklogs == 1]),len(df2[df2.HistoryOfBacklogs == 0])]
plt.show()
```

 # gender based placement count in dataset

```
plt.xticks(rotation=0)
plt.show()
```



placecement analysis 47 %placed 53% not placed with 2 internships

```
plt.figure(figsize=(6, 6))

classx = ['Placed','Not placed']
plt.title('Placements < 2 internships')
colors = sns.color_palette("hls", 8)[4:8]
dataFrame2 = df2[df2.Internships < 2]
countx = [len(dataFrame2[dataFrame2.PlacedOrNot == 1]),len(dataFrame2[dataFrame2.Plac
edOrNot == 0])]
plt.pie(countx, labels = classx,colors=colors,autopct='%1.2f%%')
plt.show()
```