```
import pandas as pd
import numpy as np
```

```
import os
```

```
path='C:\\Users\\Admin\\Desktop\\student'
```

```
data=pd.read_csv('student-mat.csv')
```

```
data
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reas
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	cour
1	GP	F	17	U	GT3	T	1	1	at_home	other	cour
2	GP	F	15	U	LE3	T	1	1	at_home	other	oth
3	GP	F	15	U	GT3	T	4	2	health	services	hor
4	GP	F	16	U	GT3	T	3	3	other	other	hor
...	...	...	...	...	...	...	...	...	...	...	
390	MS	M	20	U	LE3	A	2	2	services	services	cour
391	MS	M	17	U	LE3	T	3	1	services	services	cour
392	MS	M	21	R	GT3	T	1	1	other	other	cour
393	MS	M	18	R	LE3	T	3	2	services	other	cour
394	MS	M	19	U	LE3	T	1	1	other	at_home	cour

395 rows × 33 columns

```
data.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course
1	GP	F	17	U	GT3	T	1	1	at_home	other	course
2	GP	F	15	U	LE3	T	1	1	at_home	other	other
3	GP	F	15	U	GT3	T	4	2	health	services	home
4	GP	F	16	U	GT3	T	3	3	other	other	home

```
data.tail()
```

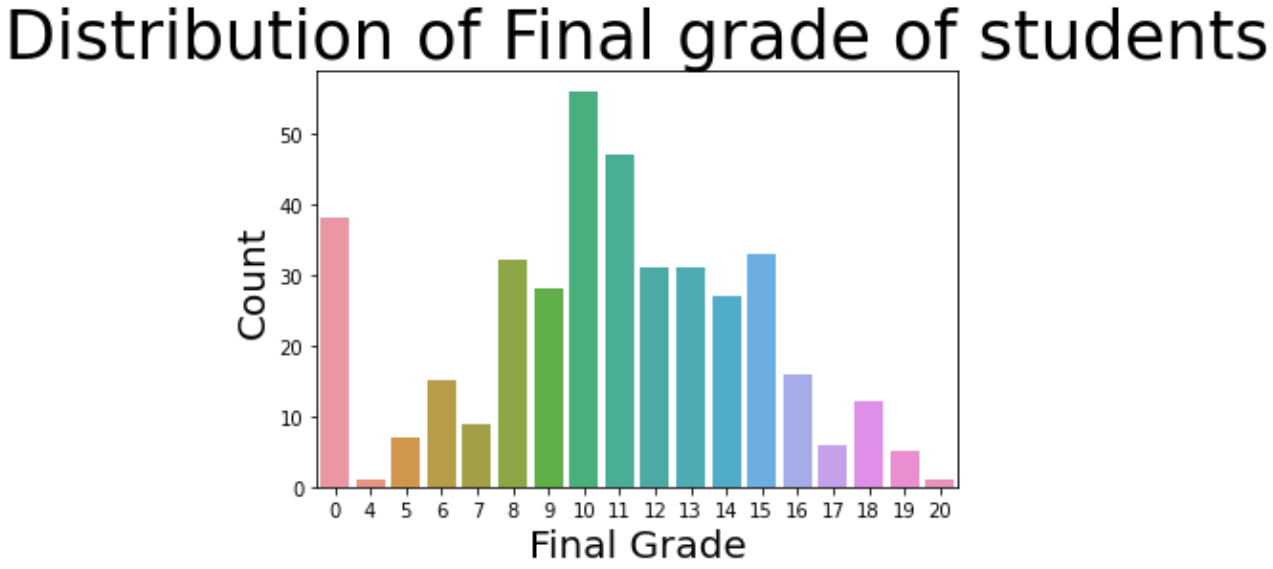
	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reas
390	MS	M	20	U	LE3	A	2	2	services	services	cours
391	MS	M	17	U	LE3	T	3	1	services	services	cours
392	MS	M	21	R	GT3	T	1	1	other	other	cours
393	MS	M	18	R	LE3	T	3	2	services	other	cours
394	MS	M	19	U	LE3	T	1	1	other	at_home	cours

```
data['G3'].describe()
```

```
count    395.000000
mean      10.415190
std        4.581443
min        0.000000
25%        8.000000
50%       11.000000
75%       14.000000
max       20.000000
Name: G3, dtype: float64
```

```
import seaborn as sns
import matplotlib.pyplot as plt
demo= sns.countplot(data['G3'])
demo.axes.set_title('Distribution of Final grade of students', fontsize = 35)
demo.set_xlabel('Final Grade', fontsize = 20)
demo.set_ylabel('Count', fontsize = 20)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```




```
data.isnull().any()
```

```
school        False
sex           False
age           False
address       False
famsize       False
Pstatus       False
Medu          False
Fedu          False
Mjob          False
Fjob          False
reason        False
guardian      False
traveltime    False
studytime     False
failures      False
schoolsup     False
famsup        False
paid          False
activities    False
nursery       False
higher        False
internet      False
romantic      False
famrel        False
freetime      False
goout         False
Dalc          False
Walc          False
health        False
absences      False
G1            False
G2            False
G3            False
dtype: bool
```

```
data.isnull().any()
```

```
school        False
sex           False
age           False
address       False
famsize       False
Pstatus       False
Medu          False
Fedu          False
Mjob          False
Fjob          False
reason        False
guardian      False
traveltime    False
studytime     False
failures      False
schoolsup     False
famsup        False
paid          False
activities    False
nursery       False
higher        False
internet      False
```

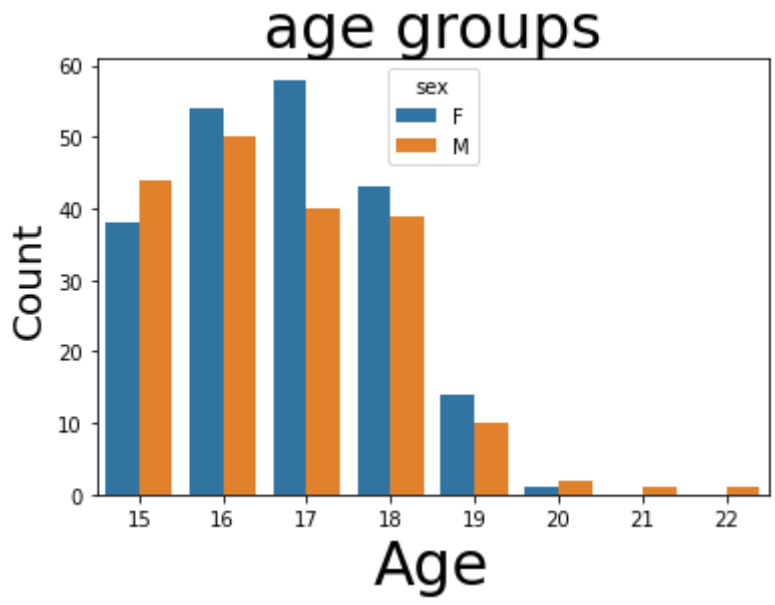
```
romantic      False
famrel        False
freetime      False
goout         False
Dalc          False
Walc          False
health        False
absences      False
G1            False
G2            False
G3            False
dtype: bool
```

```
male_student = len(data[data['sex'] == 'M'])
female_student= len(data[data['sex'] == 'F'])
print('Number of male students:',male_student)
print('Number of female students:',female_student)
```

```
Number of male students: 187
Number of female students: 208
```

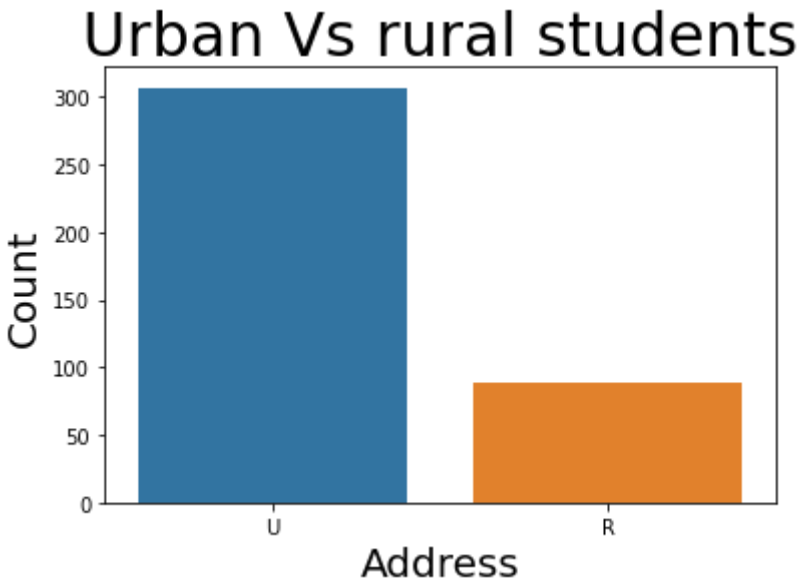
```
demo= sns.countplot('age',hue='sex', data=data)
demo.axes.set_title('age groups',fontsize=30)
demo.set_xlabel("Age",fontsize=30)
demo.set_ylabel("Count",fontsize=20)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```




```
demo = sns.countplot(data['address'])
demo.axes.set_title('Urban Vs rural students', fontsize = 30)
demo.set_xlabel('Address', fontsize = 20)
demo.set_ylabel('Count', fontsize = 20)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```




```
data.corr()['G3'].sort_values()
```

```
failures     -0.360415
age          -0.161579
goout        -0.132791
traveltime   -0.117142
health       -0.061335
Dalc         -0.054660
Walc         -0.051939
freetime      0.011307
absences      0.034247
famrel        0.051363
studytime     0.097820
Fedu          0.152457
Medu          0.217147
G1            0.801468
G2            0.904868
G3            1.000000
Name: G3, dtype: float64
```

```
data['GradeAvg'] = (data['G1'] + data['G2'] + data['G3']) / 3
```

```
data.drop(["school","age"], axis=1, inplace=True)
```

```
data.head()
```

```
      sex  address  famsize  Pstatus  Medu  Fedu     Mjob     Fjob  reason  guardian  tr
```

```
data_dum=data
```

```
1    F        U      GT3        T     1     1  at_home    other  course    father
```

```
#Converting to categorical value
categorical_d = {'yes': 1, 'no': 0}
data_dum['schoolsup'] = data_dum['schoolsup'].map(categorical_d)
data_dum['famsup'] = data_dum['famsup'].map(categorical_d)
data_dum['paid'] = data_dum['paid'].map(categorical_d)
data_dum['activities'] = data_dum['activities'].map(categorical_d)
data_dum['nursery'] = data_dum['nursery'].map(categorical_d)
data_dum['higher'] = data_dum['higher'].map(categorical_d)
data_dum['internet'] = data_dum['internet'].map(categorical_d)
data_dum['romantic'] = data_dum['romantic'].map(categorical_d)
```

```
#Converting to categorical value
categorical_d = {'yes': 1, 'no': 0}
data_dum['schoolsup'] = data_dum['schoolsup'].map(categorical_d)
data_dum['famsup'] = data_dum['famsup'].map(categorical_d)
data_dum['paid'] = data_dum['paid'].map(categorical_d)
data_dum['activities'] = data_dum['activities'].map(categorical_d)
data_dum['nursery'] = data_dum['nursery'].map(categorical_d)
data_dum['higher'] = data_dum['higher'].map(categorical_d)
data_dum['internet'] = data_dum['internet'].map(categorical_d)
data_dum['romantic'] = data_dum['romantic'].map(categorical_d)
```

```
data_dum.columns
```

```
Index(['sex', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob',
       'reason', 'guardian', 'traveltime', 'studytime', 'failures',
       'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher',
       'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc',
       'health', 'absences', 'G1', 'G2', 'G3', 'GradeAvg'],
      dtype='object')
```

```
from sklearn.model_selection import train_test_split
x=data_dum.drop("G3",axis=1)
y=data_dum['G3']
```

```
data_dum['G3']
```

```
0       6
1       6
2      10
3      15
4      10
       ..
390     9
391    16
392     7
393    10
394     9
Name: G3, Length: 395, dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.20, random_state=44
```

```
print (X_train)
```

```
     sex address famsize Pstatus  Medu  ...  health absences  G1  G2   GradeAvg
208   F       U     GT3       T     1  ...       5        6   9   9   9.333333
35    F       U     GT3       T     2  ...       5        0   8   7   7.000000
258   M       U     GT3       T     2  ...       4        8  15  14  14.333333
44    F       U     LE3       T     2  ...       5       14  10  10   9.666667
259   F       U     LE3       T     2  ...       2        0  10   9   6.333333
..   ..     ...     ...     ...   ...  ...     ...      ...  ..  ..        ...
387   F       R     GT3       T     2  ...       5        0   7   5   4.000000
59    F       U     GT3       T     4  ...       5        2  15  16  15.666667
173   F       U     GT3       T     1  ...       3        0   8   7   5.000000
241   M       R     LE3       A     4  ...       4        2  10  11  11.000000
276   F       R     GT3       A     3  ...       5       75  10   9   9.333333

[316 rows x 31 columns]
```

```
print (y_train)
```

```
208    10
35      6
258    14
44      9
259     0
       ..
387     0
59     16
173     0
241    12
276     9
Name: G3, Length: 316, dtype: int64
```

```
print (X_test)
```

```
     sex address famsize Pstatus  Medu  ...  health absences  G1  G2   GradeAvg
78    M       U     GT3       T     2  ...       3        2   8   8   8.666667
157   F       R     GT3       T     1  ...       4        6   9   8   9.000000
211   M       U     LE3       T     4  ...       3       13  12  12  12.333333
294   M       R     LE3       T     3  ...       4        8  14  13  13.666667
303   F       U     GT3       T     3  ...       5        0  17  17  17.333333
..   ..     ...     ...     ...   ...  ...     ...      ...  ..  ..        ...
268   M       U     GT3       T     4  ...       5       10  10   9   9.666667
159   M       U     GT3       T     3  ...       5        4  10  12  11.333333
20    M       U     GT3       T     4  ...       1        0  13  14  14.000000
176   F       U     GT3       T     2  ...       5        2  13  13  12.333333
107   M       U     GT3       T     3  ...       5        2  16  18  17.333333

[79 rows x 31 columns]
```

```
print (y_test)
```

```
78     10
157    10
```

```
211    13
294    14
303    18
       ..
268    10
159    12
20     15
176    11
107    18
Name: G3, Length: 79, dtype: int64
```

```
from sklearn.linear_model import LinearRegression
```

```
model=LinearRegression()
```

```
X_train
```

	sex	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
208	F	U	GT3	T	1	1	at_home	other	home	mother
35	F	U	GT3	T	2	3	other	other	other	father
258	M	U	GT3	T	2	1	other	other	home	mother
44	F	U	LE3	T	2	2	other	at_home	course	father
259	F	U	LE3	T	2	2	services	services	course	father
...	...	...	...	...	...	...	...	...	...	...
387	F	R	GT3	T	2	3	services	other	course	mother
59	F	U	GT3	T	4	2	services	other	course	mother
173	F	U	GT3	T	1	3	at_home	services	home	mother
241	M	R	LE3	A	4	4	teacher	other	course	mother
276	F	R	GT3	A	3	2	other	services	home	mother

316 rows × 31 columns

```
y_train
```

```
208    10
35      6
258    14
44      9
259     0
       ..
387     0
59     16
173     0
241    12
276     9
Name: G3, Length: 316, dtype: int64
```

```
m=model.fit
```

```
print (m)
```

```
<bound method LinearRegression.fit of LinearRegression(copy_X=True, fit_intercept=Tr
```

```
b=model.predict
```

```
print (b)
```

```
<bound method LinearModel.predict of LinearRegression(copy_X=True, fit_intercept=Tru
```

```
y=b
```

```
print (y)
```

```
<bound method LinearModel.predict of LinearRegression(copy_X=True, fit_intercept=Tru
```

0s completed at 10:08 PM