MACHINE LEARNNING PREDICTION OF SCHOOL GRADES IN SIERRA LEONE

HELLO MENTORS

MY NAME IS SHEKU KAMARA

ELECTRONICS AND TELECOMMUNICATION DEPARTMENT

UNIVERSITY OF MANAGEMENT AND TECHNOLOGY (UNIMTECH)

FREETOWN SIERRA LEONE AND DIVE INTO CODE MACHINE LEARNNINNG COURSE (JAPAN)

- INTRODUTION :
- In this session I am going to explain to you how to use machine learning technology to predict grades in schools.

- However, this data set is also available in kaggle website were I took it and implemented it as a prediction of schools grades in Sierra Leone for both junior and senior secondary school pupils.
- CONTEXT:
- It is important for all teachers in Sierra Leone to use machine learning technology to predict the grades of student . Infact, that will help the country to mach to international standards
- CONTENT:
- I was hoping to have real dataset from Sierra Leone since its impossible to have it, I decided to took it from kaggle website in order to built a module so that in the future I can do the same thing for a real dataset in Sierra Leone to predict student grades.
- The dataset contains prediction of student grades from schools within Sierra Leone.


A+

We want to know Student's Final Grade through ML !

Library

+ Code + Markdown

[1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
import pandas_profiling as pp
```

Data Loading

[2]:

```
data = pd.read_csv("../input/student-grade-prediction/student-mat.csv")
```

[3]:

```
data.head()
```

[3]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goou
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	

5 rows × 33 columns

Variable

Attribute Information:

1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€" 5th to 9th grade, 3 â€" secondary education or 4 â€" higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€" 5th to 9th grade, 3 â€" secondary education or 4 â€" higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course'

12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (numeric: n if 1<=n<3, else 4)
16. schoolsup - extra educational support (binary: yes or no)
17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (numeric: from 1 - very low to 5 - very high)
26. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27. Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. health - current health status (numeric: from 1 - very bad to 5 - very good)
30. absences - number of school absences (numeric: from 0 to 93)

30. absences - number of school absences (numeric: from 0 to 93)
31. G1 - score
32. G2 - score
33. G3 - socre

[4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64
 7   Fedu        395 non-null    int64
```

HELLOW EVERYONE

```
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64
 13  studytime   395 non-null    int64
 14  failures    395 non-null    int64
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64
 24  freetime    395 non-null    int64
 25  goout       395 non-null    int64
 26  Dalc        395 non-null    int64
 27  Walc        395 non-null    int64
 28  health      395 non-null    int64
 29  absences    395 non-null    int64
 30  G1          395 non-null    int64
 31  G2          395 non-null    int64
 32  G3          395 non-null    int64
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
```

```
[5]: import pandas_profiling as pp
     pp.ProfileReport(data)
```

```
Summarize dataset: 100%          46/46 [00:15<00:00, 1.39it/s, Completed]
Generate report structure: 100%  1/1 [00:12<00:00, 12.66s/it]
Render HTML: 100%                1/1 [00:01<00:00, 1.71s/it]
```

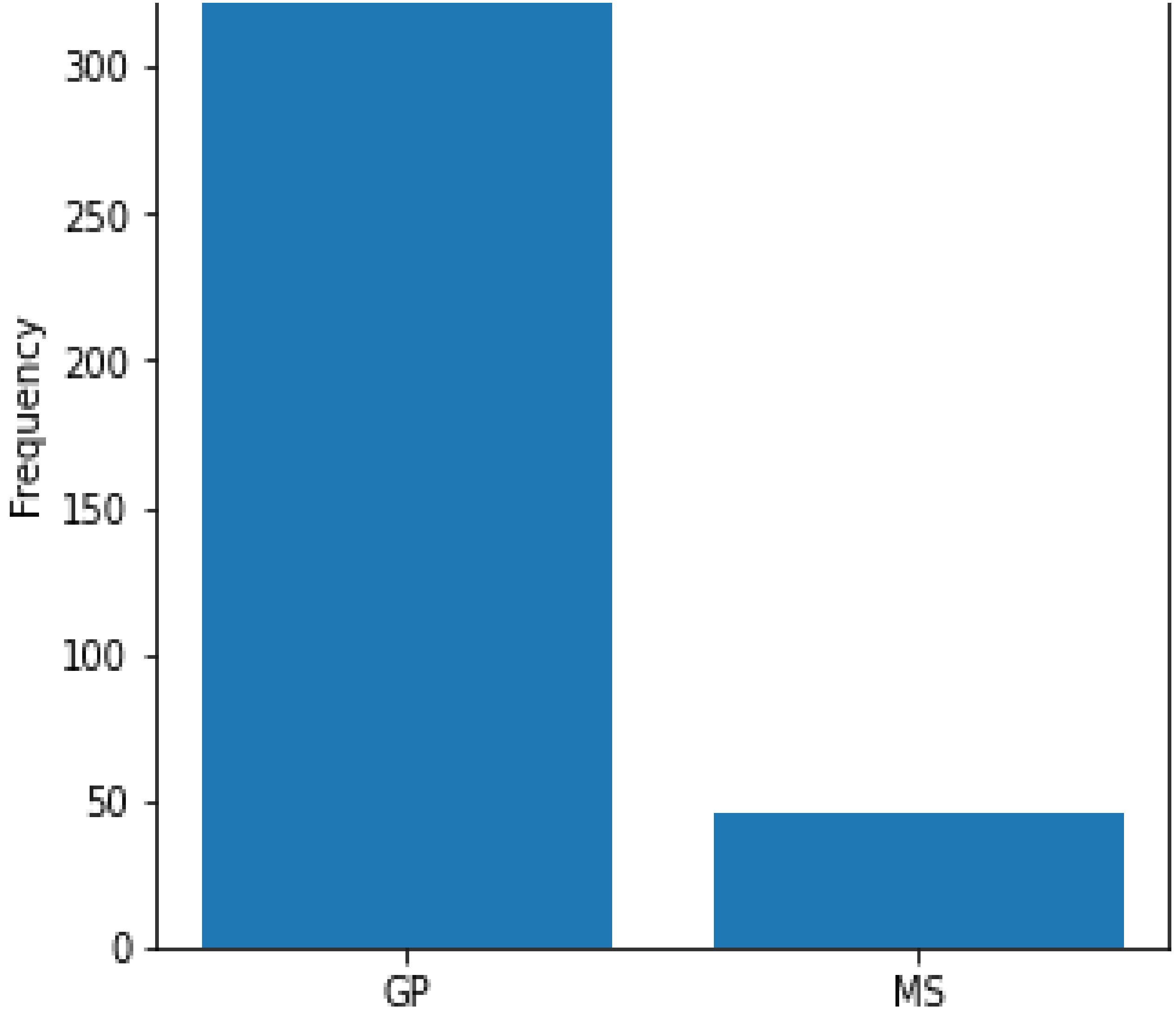
Pandas Profiling Report

Visualization

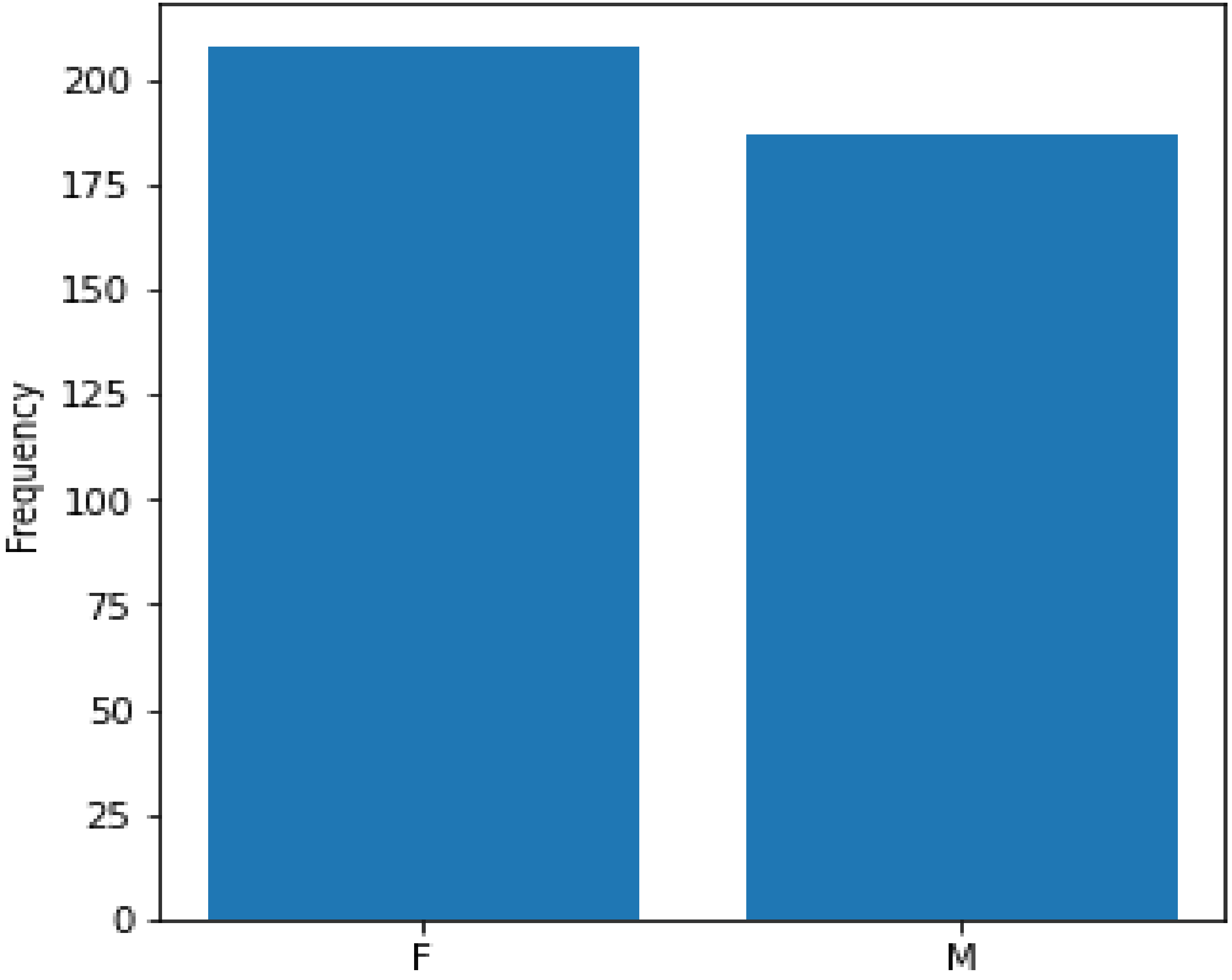
[6]:

```
def bar_plot(variable):
    var = data[variable]
    var_c = var.value_counts()

    plt.figure(figsize= (5,5))
    plt.bar(var_c.index, var_c)
    plt.ylabel('Frequency')
    plt.show()
    print("{}\n{}".format(variable, var_c))
```

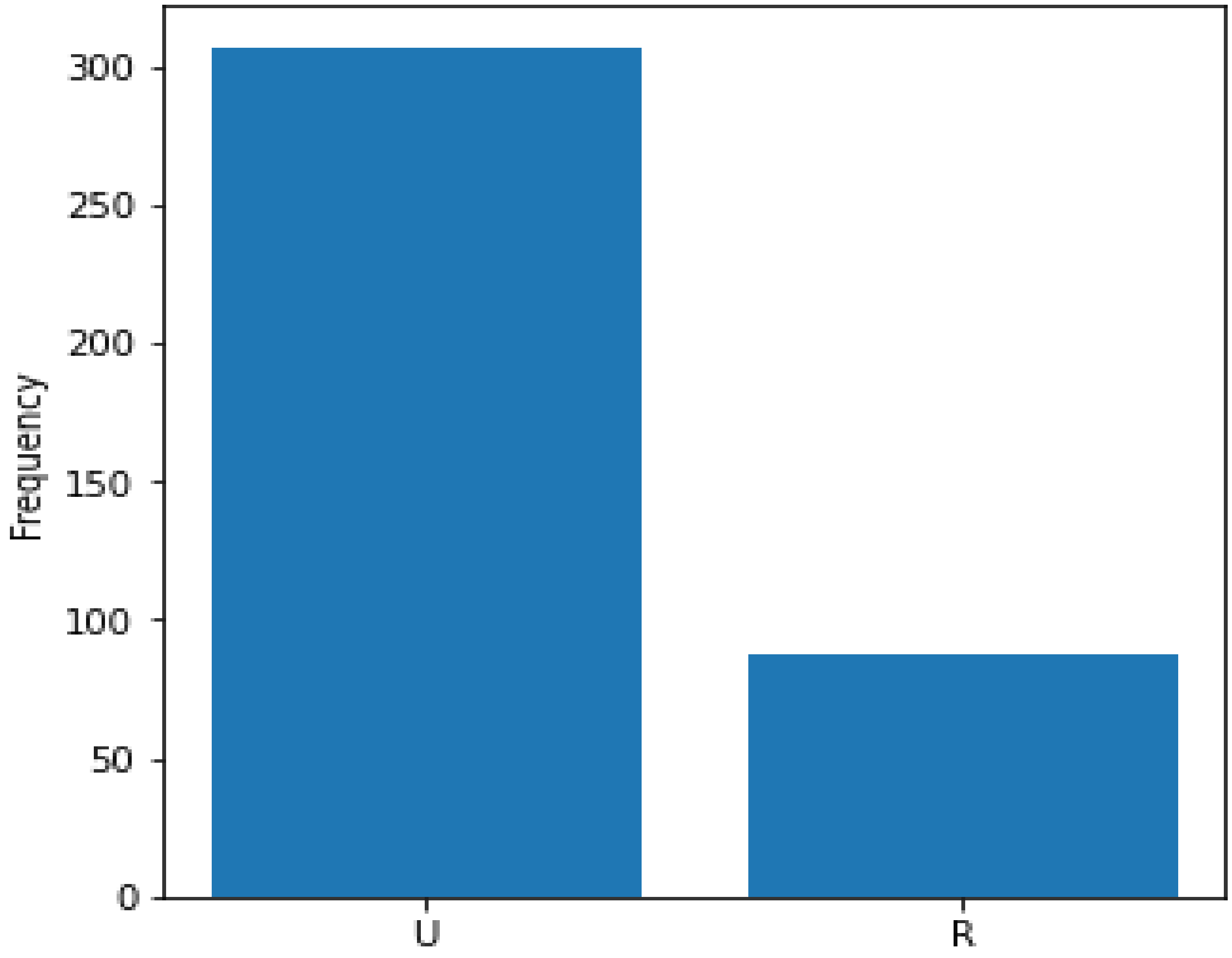
```
categorical = data.dtypes=='object'
categorical_list = list(categorical[categorical].index)
categorical_list


for i in categorical_list:
    bar_plot(i)
```



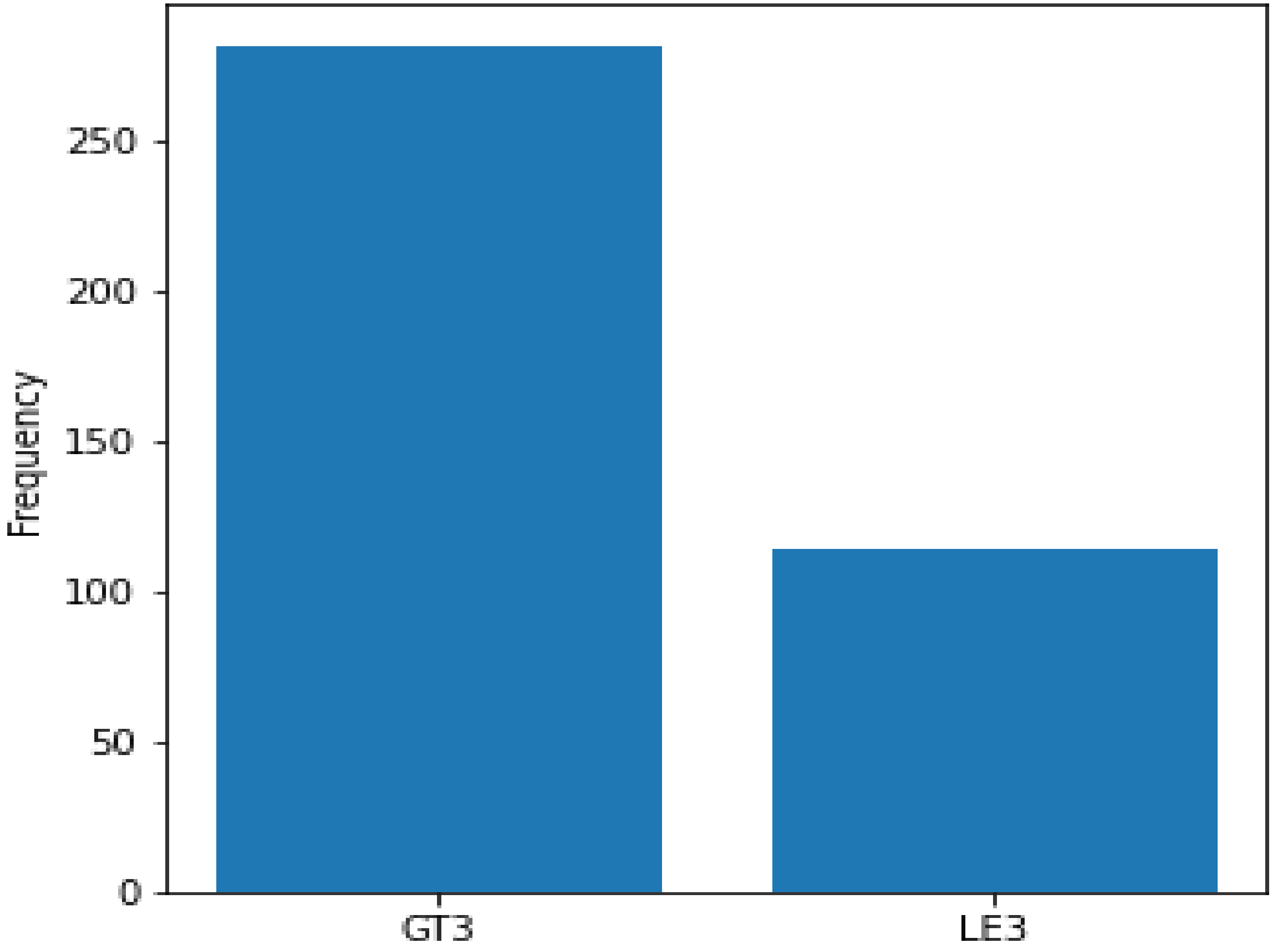

```
school
GP    349
MS     46
Name: school, dtype: int64
```



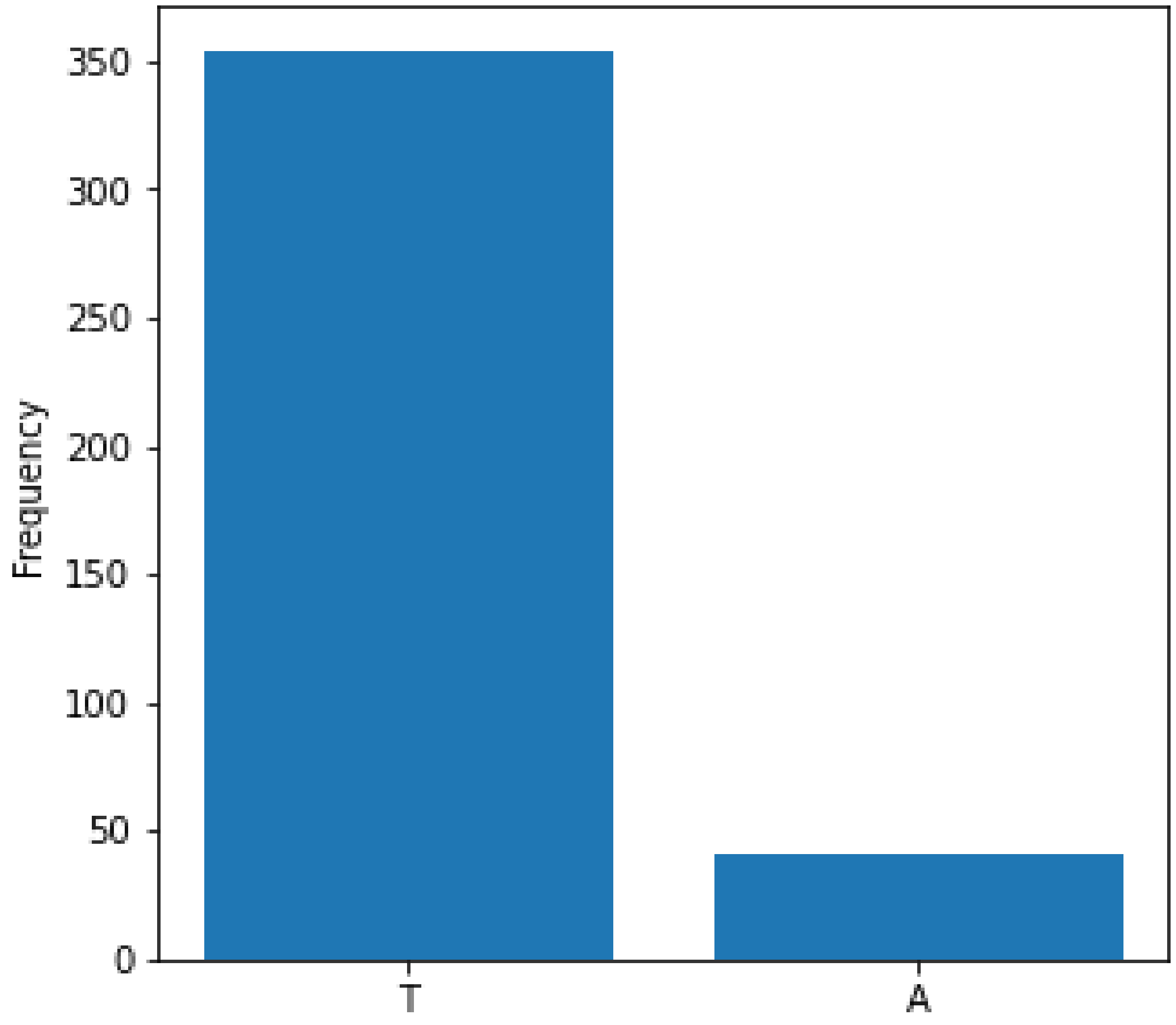

```
sex
F    208
M    187
Name: sex, dtype: int64
```



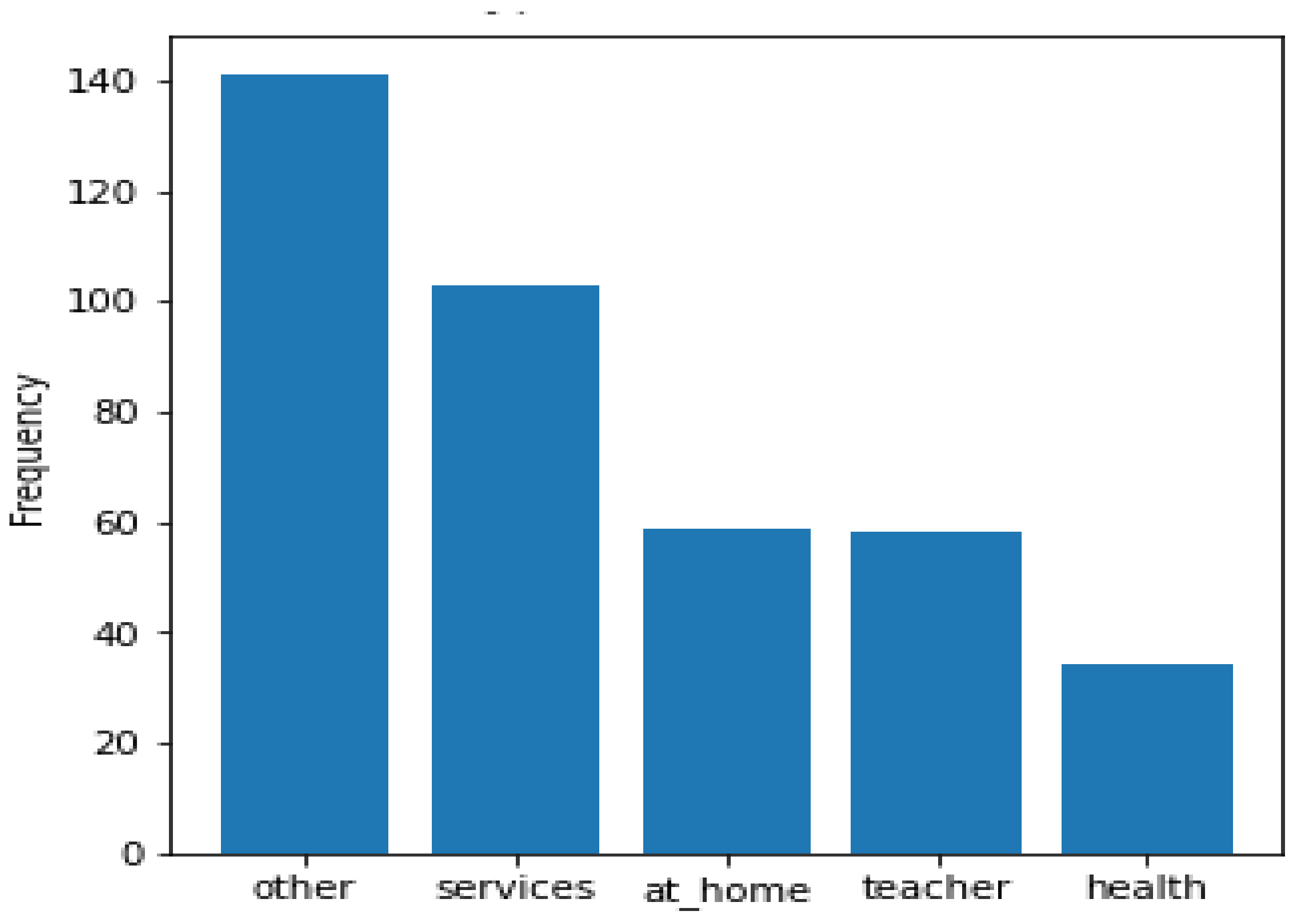

```
address
U    307
R     88
Name: address, dtype: int64
```



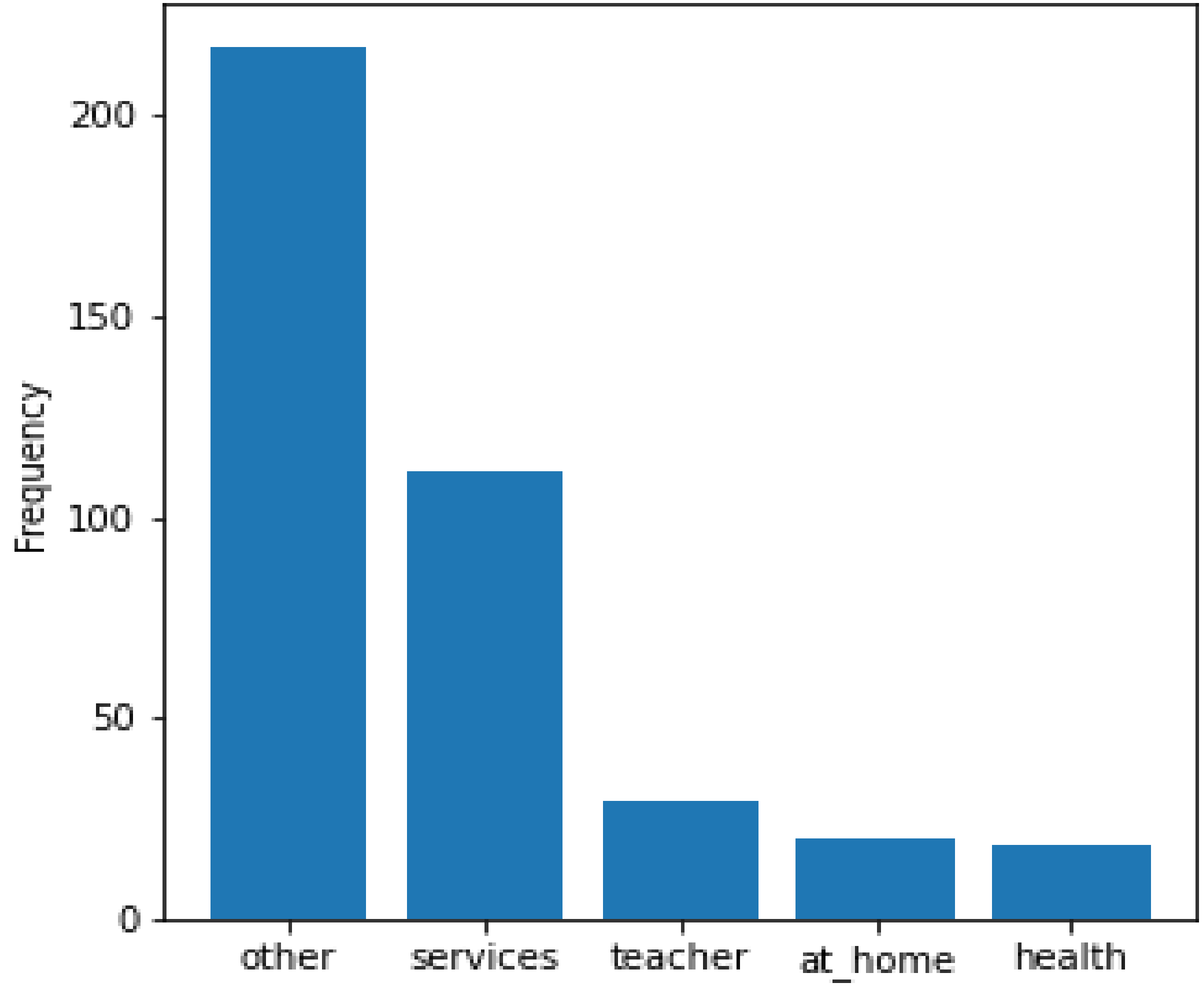

```
famsize
GT3    281
LE3    114
Name: famsize, dtype: int64
```



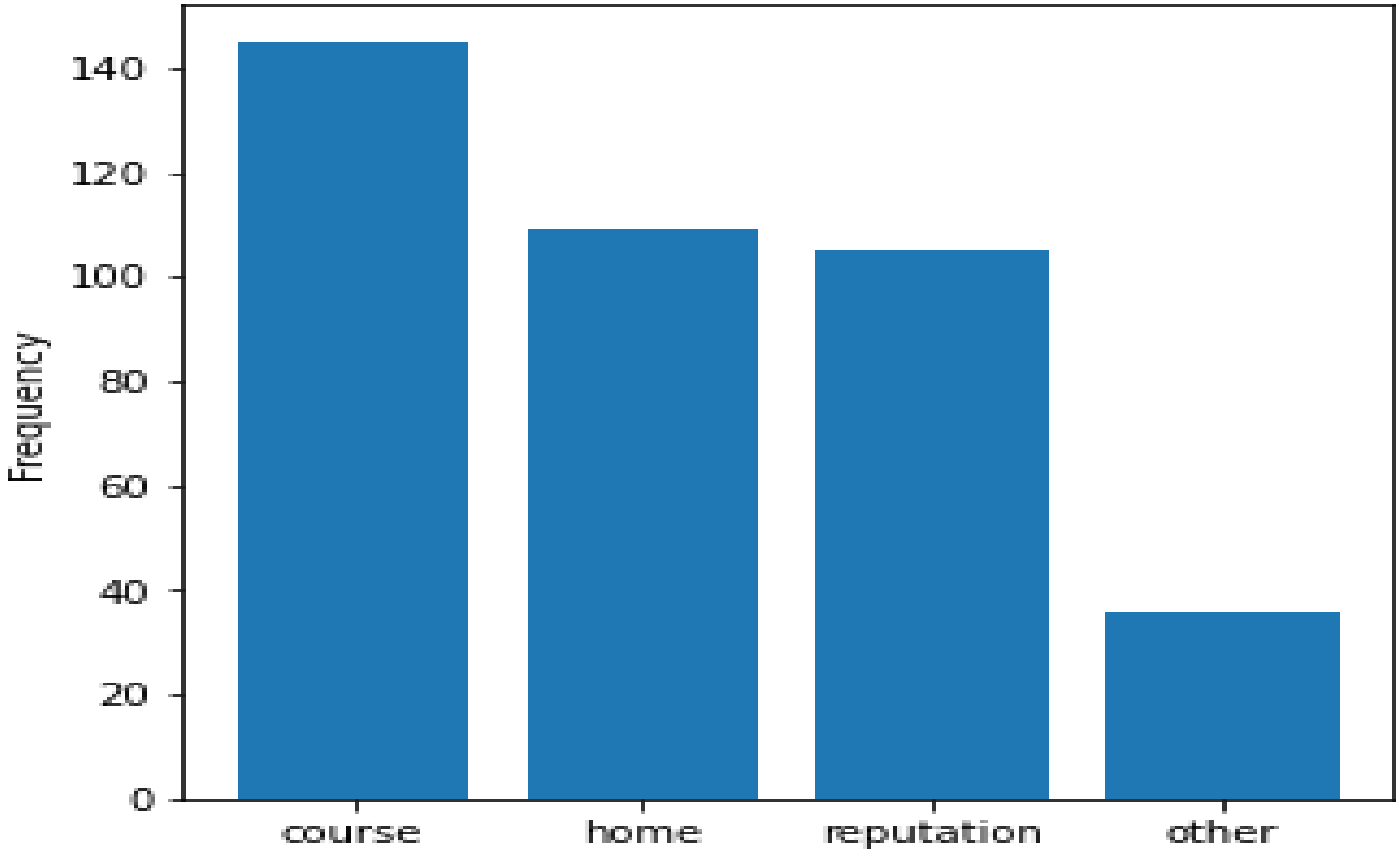

```
Pstatus
T    354
A     41
Name: Pstatus, dtype: int64
```



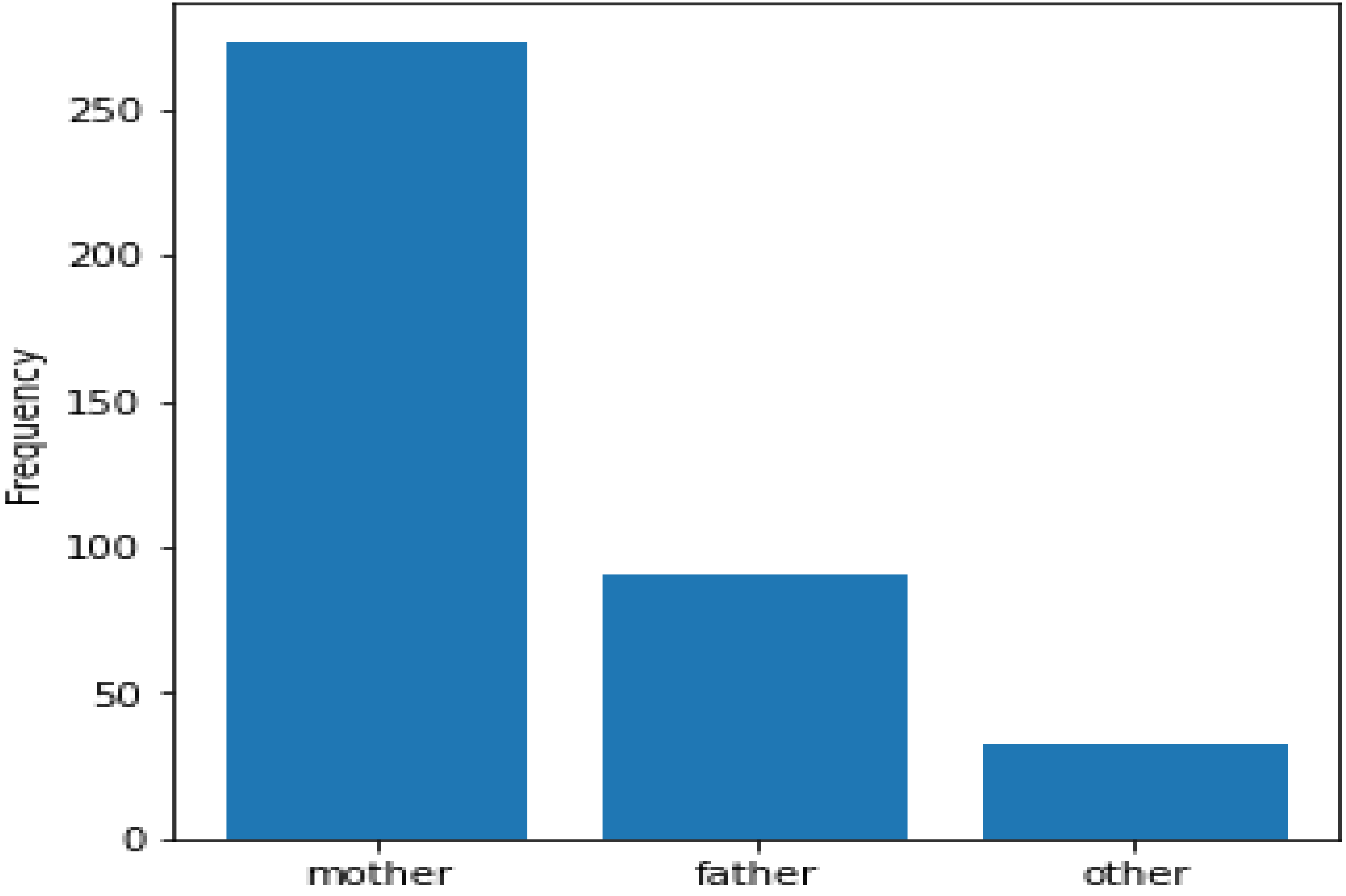

```
Mjob
other       141
services    103
at_home      59
teacher      58
health       34
Name: Mjob, dtype: int64
```



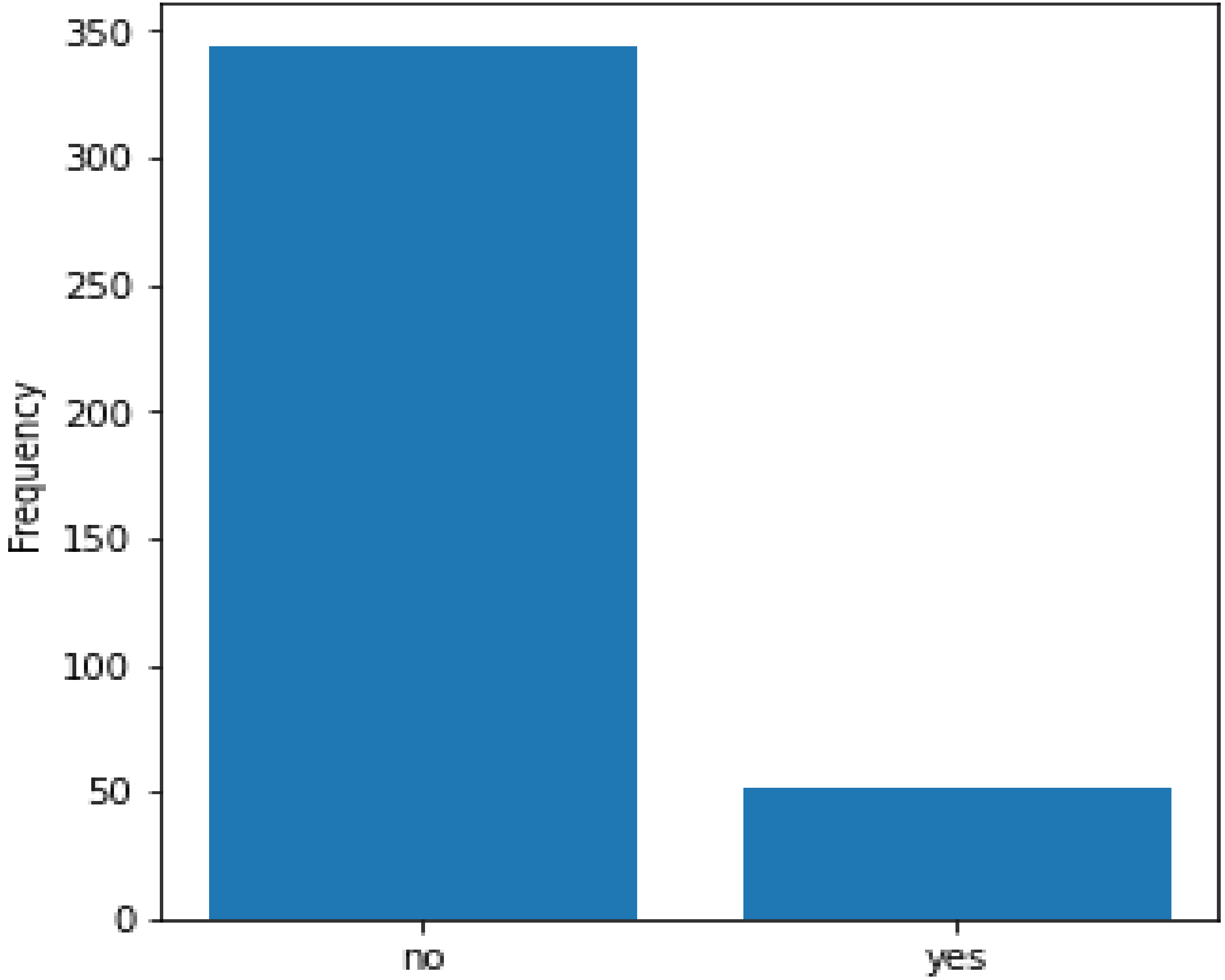

```
Fjob
other       217
services    111
teacher      29
at_home      20
health       18
Name: Fjob, dtype: int64
```




```
reason
course        145
home          109
reputation    105
other          36
Name: reason, dtype: int64
```




```
guardian
mother    273
father     90
other      32
Name: guardian, dtype: int64
```

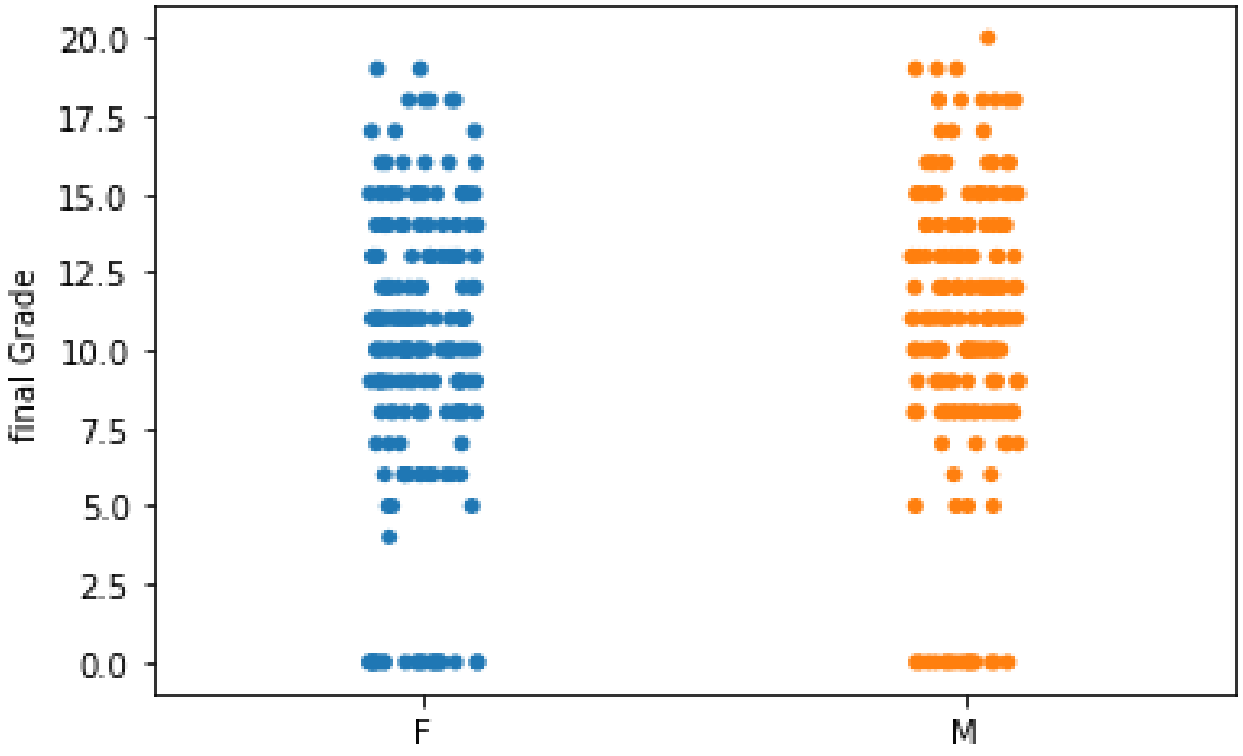



```
schoolsup
no      344
yes      51
Name: schoolsup, dtype: int64
```

Sex

[8]:

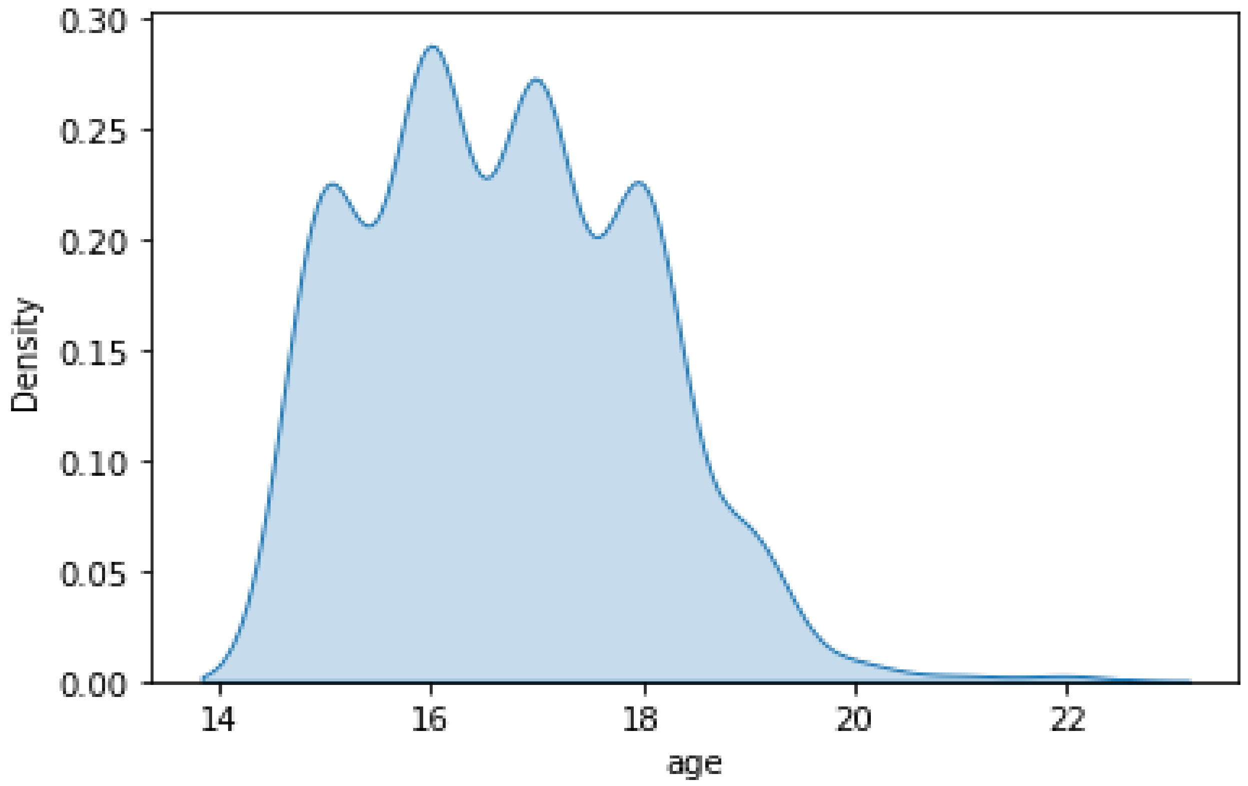
```
sns.stripplot(x=data['sex'], y=data['G3'])
plt.ylabel('final Grade')
plt.show()
```



Age

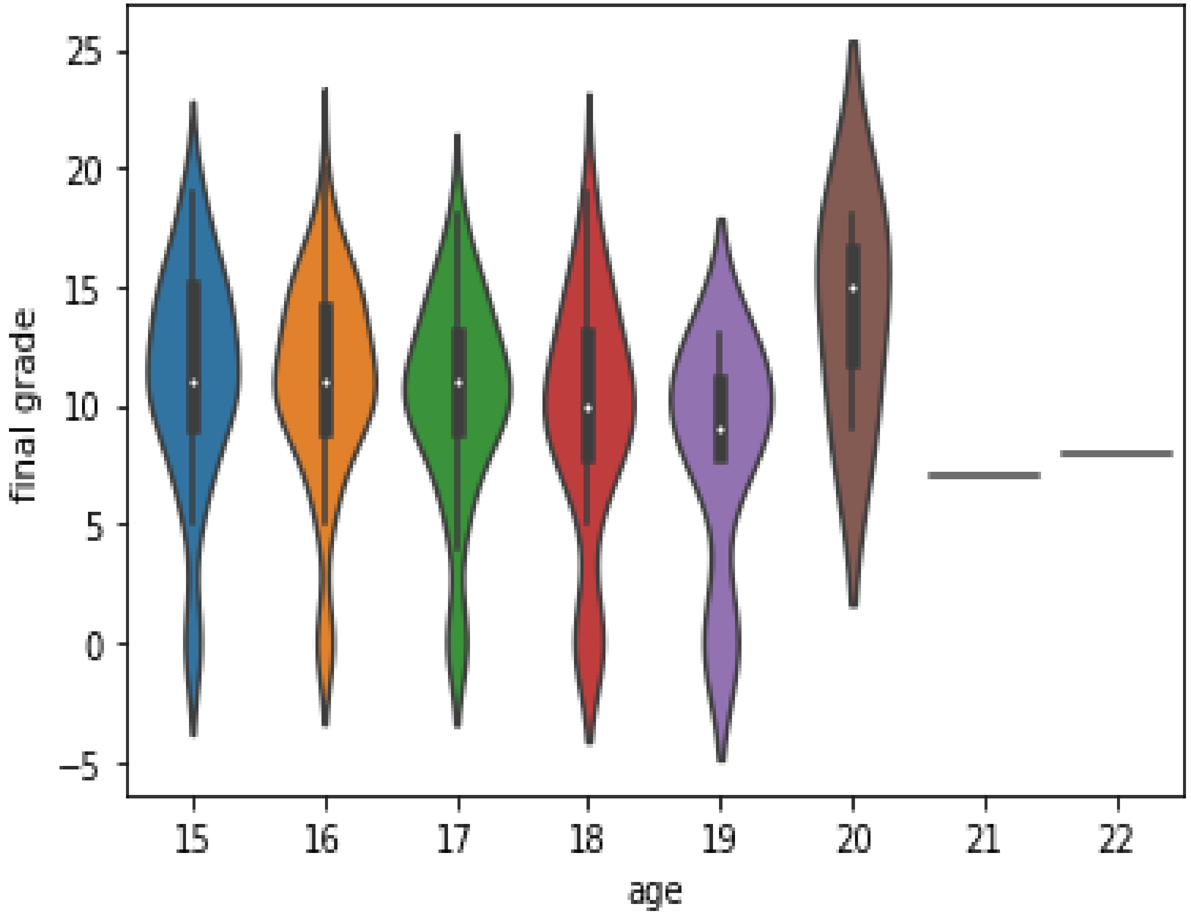
[9]:

```
#1 age's dist
sns.kdeplot(data['age'], shade=True)
plt.show()
```



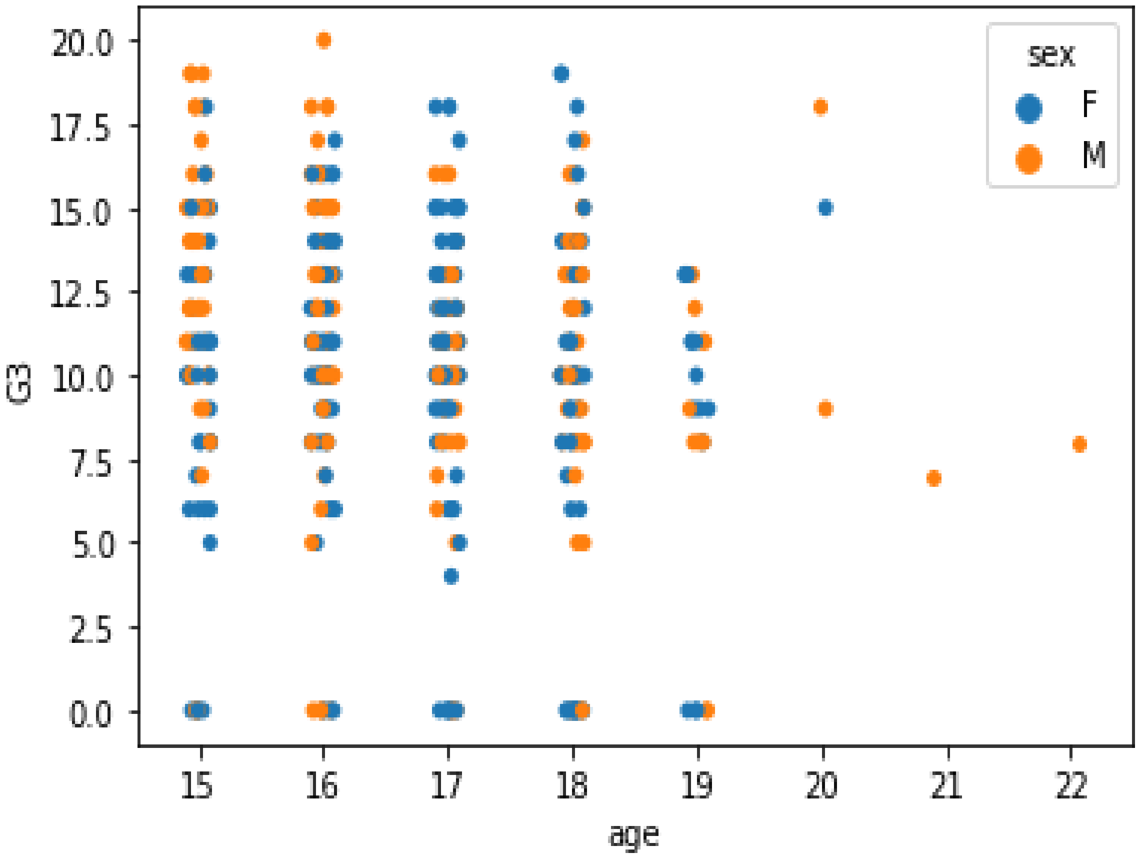
[10]:

```
sns.violinplot(data=data, x='age',y='G3')
plt.ylabel("final grade")
plt.show()
```



[11]:

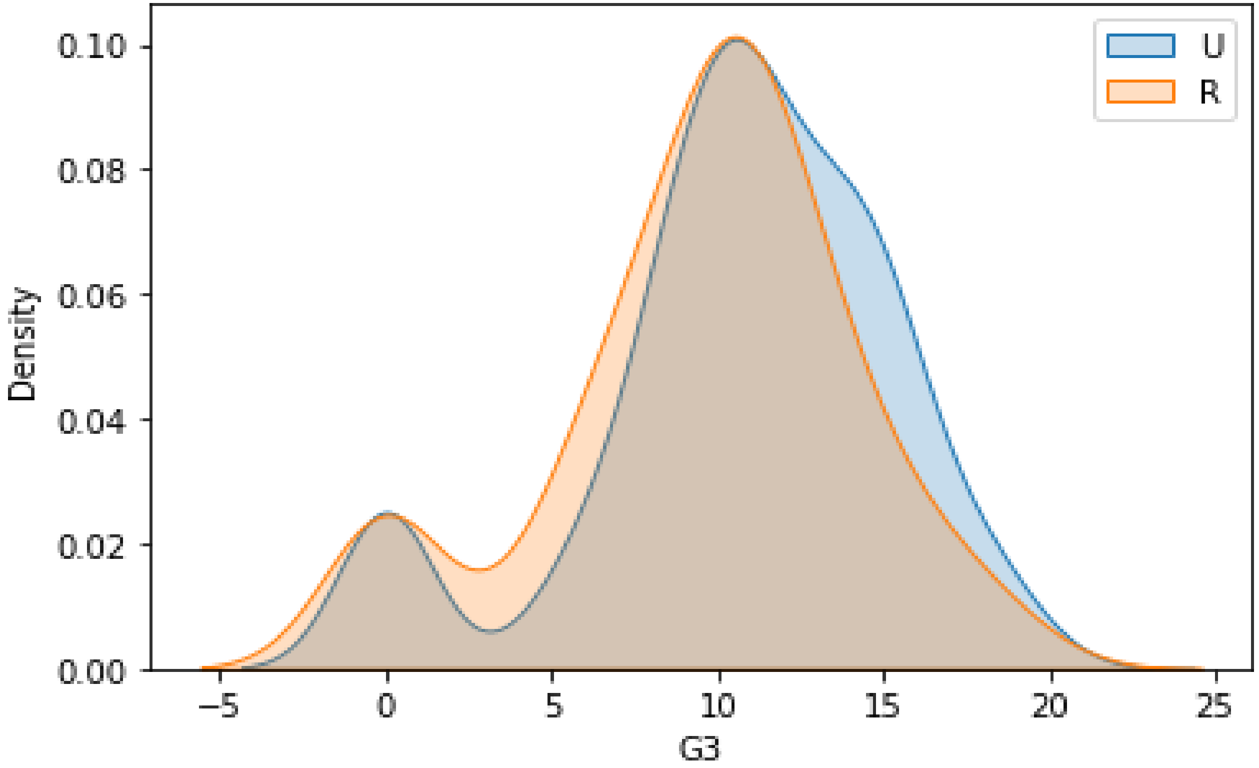
```
sns.stripplot(data=data, x='age',y='G3', hue='sex')
plt.show()
```



address

[12]:

```
sns.kdeplot(data.loc[data['address']== 'U', 'G3'], shade=True)
sns.kdeplot(data.loc[data['address']== 'R','G3'],shade= True)
plt.legend(data['address'].unique())
plt.show()
```



We can find out urban students make high scores.

I think it is inefficient to find significant with all variables,

So then, I will use variables with high correlation to G3

[13]:

```
numeric = data.dtypes=='int64'
numeric_list= numeric[numeric].index
```

[14]:

```
for i in numeric_list:
    print(i ,':', np.round(data['G3'].corr(data[i]), 2))
```

```
age : -0.16
Medu : 0.22
Fedu : 0.15
traveltime : -0.12
studytime : 0.1
failures : -0.36
famrel : 0.05
freetime : 0.01
goout : -0.13
Dalc : -0.05
Walc : -0.05
health : -0.06
absences : 0.03
G1 : 0.8
G2 : 0.9
G3 : 1.0
```

[26]:

```python
#e.g.,
pred = best_grid.predict(test_input)
pd.DataFrame(pred, test_target)
```

[26]:

	0
G3	
12	11.75
10	9.25
12	10.50
9	10.00
12	12.75
...	...
0	2.00
7	6.50

[27]:

```python
from sklearn.metrics import mean_absolute_error, mean_squared_error
print(mean_absolute_error(test_target, pred)) #평균오차
print(mean_squared_error(test_target, pred))
```

```
1.4113924050632911
4.780063291139241
```

Dtr

[28]:

```python
from sklearn.tree import DecisionTreeRegressor
dtr= DecisionTreeRegressor()
dtr.fit(train_input, train_target)
print(dtr.score(test_input, test_target))
```

```
0.6328056142218084
```

CONCLUSION

I couldn’t use dataset of Sierra Leone, but in the further, since I have built a model, I am planning to use a real dataset from Cards Fraud in Sierra Leone.