∨ Student Alcohol Consumption

∨ Introduction:

This time you will download a dataset from the UCI.

Step 1. Import the necessary libraries

```
import pandas as pd
```

Step 2. Import the dataset from this [address](#).

∨ Step 3. Assign it to a variable called df.

```
df = pd.read_csv('student_alcohol.csv')
```

∨ Step 4. For the purpose of this exercise slice the dataframe from 'school' until the 'guardian' column

```
step4 = df.loc[:, 'school':'guardian']
print(step4)
```

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

         reason guardian
    0    course   mother
    1    course   father
    2     other   mother
    3      home   mother
    4      home   father
    ..      ...      ...
    390  course    other
    391  course   mother
    392  course    other
    393  course   mother
    394  course   father

    [395 rows x 12 columns]
```

∨ Step 5. Create a lambda function that will capitalize strings.

```
step5 = lambda x: x.str.capitalize()
print(step5)
```

```
    <function <lambda> at 0x79084809ae80>
```

∨ Step 6. Capitalize both Mjob and Fjob

```
step6 = df[['Mjob', 'Fjob']].apply(step5)
print(step6)
```

```
            Mjob      Fjob
    0    At_home   Teacher
    1    At_home     Other
    2    At_home     Other
```

```
3      Health  Services
4       Other     Other
..        ...       ...
390  Services  Services
391  Services  Services
392     Other     Other
393  Services     Other
394     Other   At_home

[395 rows x 2 columns]
```

Step 7. Print the last elements of the data set.

```
step7 = df.tail()
print(step7)
```

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

     ... famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
390  ...      5        5      4     4     5      4       11   9   9   9
391  ...      2        4      5     3     4      2        3  14  16  16
392  ...      5        5      3     3     3      3        3  10   8   7
393  ...      4        4      1     3     4      5        0  11  12  10
394  ...      3        2      3     3     3      5        5   8   9   9

[5 rows x 33 columns]
```

Step 8. Did you notice the original dataframe is still lowercase? Why is that? Fix it and capitalize Mjob and Fjob.

```
df[['Mjob', 'Fjob']] = df[['Mjob', 'Fjob']].apply(step5)
step8 = df[['Mjob', 'Fjob']]
print(step8)
```

```
         Mjob      Fjob
0     At_home   Teacher
1     At_home     Other
2     At_home     Other
3      Health  Services
4       Other     Other
..        ...       ...
390  Services  Services
391  Services  Services
392     Other     Other
393  Services     Other
394     Other   At_home

[395 rows x 2 columns]
```

Step 9. Create a function called majority that returns a boolean value to a new column called legal_drinker (Consider majority as older than 17 years old)

```
def majority(age):
    return age > 17

df['legal_drinker'] = df['age'].apply(majority)
step9 = df[['age', 'legal_drinker']]
print(step9)
```

```
     age  legal_drinker
0     18           True
1     17          False
2     15          False
3     15          False
4     16          False
..   ...            ...
390   20           True
391   17          False
392   21           True
```

```
393   18            True
394   19            True

[395 rows x 2 columns]
```

∨ Step 10. Multiply every number of the dataset by 10.

I know this makes no sense, don't forget it is just an exercise

```
step10 = df.select_dtypes(include=['int64', 'float64']) * 10
print(step10)
```

```
       age  Medu  Fedu  traveltime  studytime  failures  famrel  freetime  \
0      180    40    40          20         20         0      40        30
1      170    10    10          10         20         0      50        30
2      150    10    10          10         20        30      40        30
3      150    40    20          10         30         0      30        20
4      160    30    30          10         20         0      40        30
..     ...   ...   ...         ...        ...       ...     ...       ...
390    200    20    20          10         20        20      50        50
391    170    30    10          20         10         0      20        40
392    210    10    10          10         10        30      50        50
393    180    30    20          30         10         0      40        40
394    190    10    10          10         10         0      30        20

     goout  Dalc  Walc  health  absences   G1   G2   G3
0       40    10    10      30        60   50   60   60
1       30    10    10      30        40   50   50   60
2       20    20    30      30       100   70   80  100
3       20    10    10      50        20  150  140  150
4       20    10    20      50        40   60  100  100
..     ...   ...   ...     ...       ...  ...  ...  ...
390     40    40    50      40       110   90   90   90
391     50    30    40      20        30  140  160  160
392     30    30    30      30        30  100   80   70
393     10    30    40      50         0  110  120  100
394     30    30    30      50        50   80   90   90

[395 rows x 16 columns]
```