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.

```
url = "https://raw.githubusercontent.com/thieu1995/csv-files/main/data/pandas/student-mat.csv"
```

Step 3. Assign it to a variable called df.

```
df = pd.read_csv(url)
```

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

```
df = df.iloc[:, :12]

print(df)
```

```
    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.

```
capitalize = lambda x: x.capitalize()
```

Step 6. Capitalize both Mjob and Fjob

```
df[['Mjob', 'Fjob']] = df[['Mjob', 'Fjob']].applymap(capitalize)

print(df)
```

```
    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]
<ipython-input-13-54578b1a9087>:1: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
  df[['Mjob', 'Fjob']] = df[['Mjob', 'Fjob']].applymap(capitalize)
```

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

```
print(df.tail(1))
```

```
     school sex  age address famsize Pstatus  Medu  Fedu   Mjob     Fjob  \
394     MS   M   19       U     LE3       T     1     1  Other  At_home

     reason guardian
394  course   father
```

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

```
df['Mjob'] = df['Mjob'].apply(lambda x: x.capitalize())
df['Fjob'] = df['Fjob'].apply(lambda x: x.capitalize())

print(df)
```

```
     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 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)

```
df['legal_drinker'] = df['age'].apply(lambda x: x > 17)

print(df)
```

```
     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  legal_drinker
0    course   mother           True
1    course   father          False
2     other   mother          False
3      home   mother          False
4      home   father          False
..      ...      ...            ...
390  course    other           True
391  course   mother          False
392  course    other           True
393  course   mother           True
394  course   father           True

[395 rows x 13 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

```
df[['age', 'Medu', 'Fedu']] = df[['age', 'Medu', 'Fedu']].apply(lambda x: x * 10)

print(df)
```

```
     school sex  age address famsize Pstatus  Medu  Fedu      Mjob      Fjob  \
0       GP   F  180       U     GT3       A    40    40   At_home   Teacher
1       GP   F  170       U     GT3       T    10    10   At_home     Other
2       GP   F  150       U     LE3       T    10    10   At_home     Other
3       GP   F  150       U     GT3       T    40    20    Health  Services
4       GP   F  160       U     GT3       T    30    30     Other     Other
..     ...  ..  ...     ...     ...     ...   ...   ...       ...       ...
390     MS   M  200       U     LE3       A    20    20  Services  Services
391     MS   M  170       U     LE3       T    30    10  Services  Services
392     MS   M  210       R     GT3       T    10    10     Other     Other
393     MS   M  180       R     LE3       T    30    20  Services     Other
394     MS   M  190       U     LE3       T    10    10     Other   At_home

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

[395 rows x 13 columns]
```