Student Alcohol Consumption

Introduction:

This time you will download a dataset from the UCI.

Step 1. Import the necessary libraries

In [2]:
```
import pandas as pd
import numpy as np
```

Step 2. Import the dataset from this address.

Step 3. Assign it to a variable called df.

In [3]:
```
df = pd.read_csv("https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/04_Apply/Students_Alcohol_Consumption/stud
df
```

Out[3]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	11	9	9	9
391	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	0	11	12	10
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	5	8	9	9

In [21]:
```
df.describe()
```

Out[21]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	abse
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.0
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	1.481013	2.291139	3.554430	5.71
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	0.890741	1.287897	1.390303	8.00
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.00
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	1.000000	1.000000	3.000000	0.00
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.000000	4.00
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	2.000000	3.000000	5.000000	8.00
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	75.0

In [22]:
```
df.info
```

Out[22]:
```
<bound method DataFrame.info of      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

     ... freetime goout  Dalc  Walc  health absences  G1  G2  G3 legal_drinker
0    ...        3     4     1     1       3        6   5   6   6          True
1    ...        3     3     1     1       3        4   5   5   6         False
2    ...        3     2     2     3       3       10   7   8  10         False
3    ...        2     2     1     1       5        2  15  14  15         False
4    ...        3     2     1     2       5        4   6  10  10         False
..   ...      ...   ...   ...   ...     ...      ...  ..  ..  ..           ...
390  ...        5     4     4     5       4       11   9   9   9          True
391  ...        4     5     3     4       2        3  14  16  16         False
392  ...        5     3     3     3       3        3  10   8   7          True
393  ...        4     1     3     4       5        0  11  12  10          True
394  ...        2     3     3     3       5        5   8   9   9          True

[395 rows x 34 columns]>
```

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

In [6]:
```
dff = df.loc[: , "school": "guardian"]
dff
```

Out[6]:

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

395 rows × 12 columns

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

In [7]:
```
func = lambda x: x.str.upper()
```

Step 6. Capitalize both Mjob and Fjob

In [8]:
```
df[["Mjob","Fjob"]] = df[["Mjob","Fjob"]].apply(func)
df
```

Out[8]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	AT_HOME	TEACHER	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	AT_HOME	OTHER	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	AT_HOME	OTHER	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	HEALTH	SERVICES	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	OTHER	OTHER	...	4	3	2	1	2	5	4	6	10	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	SERVICES	SERVICES	...	5	5	4	4	5	4	11	9	9	9
391	MS	M	17	U	LE3	T	3	1	SERVICES	SERVICES	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	OTHER	OTHER	...	5	5	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	SERVICES	OTHER	...	4	4	1	3	4	5	0	11	12	10
394	MS	M	19	U	LE3	T	1	1	OTHER	AT_HOME	...	3	2	3	3	3	5	5	8	9	9

395 rows × 33 columns

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

In [13]:
```
df.tail(1)
```

Out[13]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
394	MS	M	19	U	LE3	T	1	1	OTHER	AT_HOME	...	3	2	3	3	3	5	5	8	9	9

1 rows × 33 columns

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

In [14]:
```
df
```

Out[14]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	AT_HOME	TEACHER	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	AT_HOME	OTHER	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	AT_HOME	OTHER	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	HEALTH	SERVICES	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	OTHER	OTHER	...	4	3	2	1	2	5	4	6	10	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	SERVICES	SERVICES	...	5	5	4	4	5	4	11	9	9	9
391	MS	M	17	U	LE3	T	3	1	SERVICES	SERVICES	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	OTHER	OTHER	...	5	5	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	SERVICES	OTHER	...	4	4	1	3	4	5	0	11	12	10
394	MS	M	19	U	LE3	T	1	1	OTHER	AT_HOME	...	3	2	3	3	3	5	5	8	9	9

395 rows × 33 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)

In [16]:
```
majority = lambda x: True if x > 17 else False
```

In [18]:
```
df['legal_drinker'] = df.age.apply(majority)
df
```

Out[18]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	legal_d
0	GP	F	18	U	GT3	A	4	4	AT_HOME	TEACHER	...	3	4	1	1	3	6	5	6	6	
1	GP	F	17	U	GT3	T	1	1	AT_HOME	OTHER	...	3	3	1	1	3	4	5	5	6	
2	GP	F	15	U	LE3	T	1	1	AT_HOME	OTHER	...	3	2	2	3	3	10	7	8	10	
3	GP	F	15	U	GT3	T	4	2	HEALTH	SERVICES	...	2	2	1	1	5	2	15	14	15	
4	GP	F	16	U	GT3	T	3	3	OTHER	OTHER	...	3	2	1	2	5	4	6	10	10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	SERVICES	SERVICES	...	5	4	4	5	4	11	9	9	9	
391	MS	M	17	U	LE3	T	3	1	SERVICES	SERVICES	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	OTHER	OTHER	...	5	3	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	SERVICES	OTHER	...	4	1	3	4	5	0	11	12	10	
394	MS	M	19	U	LE3	T	1	1	OTHER	AT_HOME	...	3	3	3	3	5	5	8	9	9	

395 rows × 34 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

In [19]:
```
df.select_dtypes("number")*10
```

Out[19]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	180	40	40	20	20	0	40	30	40	10	10	30	60	50	60	60
1	170	10	10	10	20	0	50	30	30	10	10	30	40	50	50	60
2	150	10	10	10	20	30	40	30	20	20	30	30	100	70	80	100
3	150	40	20	10	30	0	30	20	20	10	10	50	20	150	140	150
4	160	30	30	10	20	0	40	30	20	10	20	50	40	60	100	100
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	200	20	20	10	20	20	50	50	40	40	50	40	110	90	90	90
391	170	30	10	20	10	0	20	40	50	30	40	20	30	140	160	160
392	210	10	10	10	10	30	50	50	30	30	30	30	30	100	80	70
393	180	30	20	30	10	0	40	40	10	30	40	50	0	110	120	100
394	190	10	10	10	10	0	30	20	30	30	30	50	50	80	90	90

395 rows × 16 columns

In [20]:
```
times_10 = lambda x: x * 10 if (type(x) is int) else x
df.applymap(times_10).head(10)
```

Out[20]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	legal_d
0	GP	F	180	U	GT3	A	40	40	AT_HOME	TEACHER	...	30	40	10	10	30	60	50	60	60	
1	GP	F	170	U	GT3	T	10	10	AT_HOME	OTHER	...	30	30	10	10	30	40	50	50	60	
2	GP	F	150	U	LE3	T	10	10	AT_HOME	OTHER	...	30	20	20	30	30	100	70	80	100	
3	GP	F	150	U	GT3	T	40	20	HEALTH	SERVICES	...	20	20	10	10	50	20	150	140	150	
4	GP	F	160	U	GT3	T	30	30	OTHER	OTHER	...	30	20	10	20	50	40	60	100	100	
5	GP	M	160	U	LE3	T	40	30	SERVICES	OTHER	...	40	20	10	20	50	100	150	150	150	
6	GP	M	160	U	LE3	T	20	20	OTHER	OTHER	...	40	40	10	10	30	0	120	120	110	
7	GP	F	170	U	GT3	A	40	40	OTHER	TEACHER	...	10	40	10	10	10	60	60	50	60	
8	GP	M	150	U	LE3	A	30	20	SERVICES	OTHER	...	20	20	10	10	10	0	160	180	190	
9	GP	M	150	U	GT3	T	30	40	OTHER	OTHER	...	50	10	10	10	50	0	140	150	150	

10 rows × 34 columns