Advanced programming

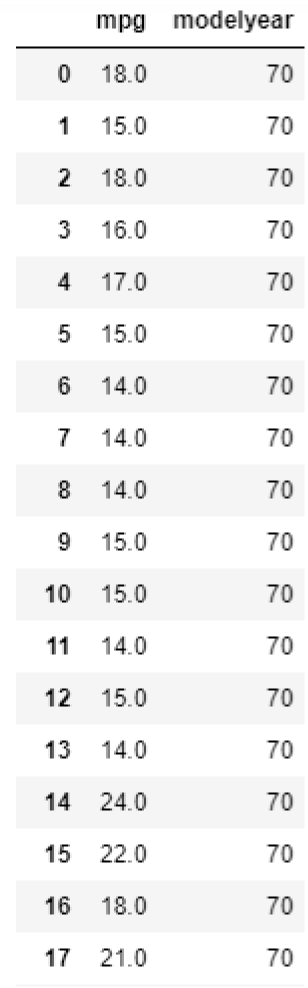
Wanwen liu Sun He

First we upload our datasets

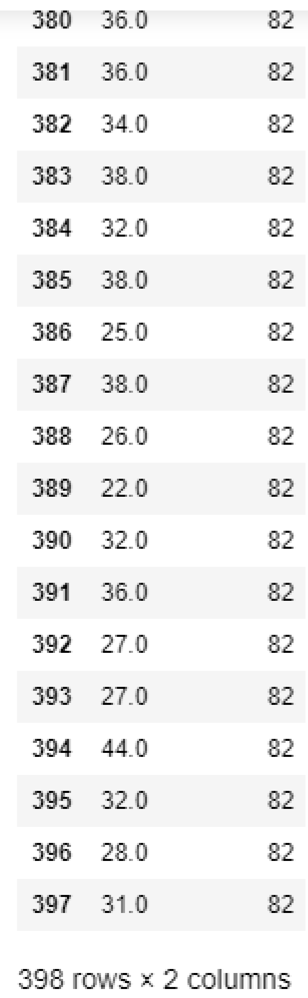
```
import pandas as pd

cars = pd.read_csv('auto-mpg.data', delim_whitespace=True,header=None)

cars.describe()

modelyear = cars.iloc[:,6].tolist()

mpg = cars.iloc[:,0].tolist()
```

Second we modify our data for preparing next steps. We built a new dataset of modelyear and mpg.

```
mymodelyear=list(set(modelyear))

cars1=cars.iloc[:,np.r_[0,6]]

cars1.columns=["mpg","modelyear"]

cars1
```



	mpg	modelyear
0	18.0	70
1	15.0	70
2	18.0	70
3	16.0	70
4	17.0	70
5	15.0	70
6	14.0	70
7	14.0	70
8	14.0	70
9	15.0	70
10	15.0	70
11	14.0	70
12	15.0	70
13	14.0	70
14	24.0	70
15	22.0	70
16	18.0	70
17	21.0	70



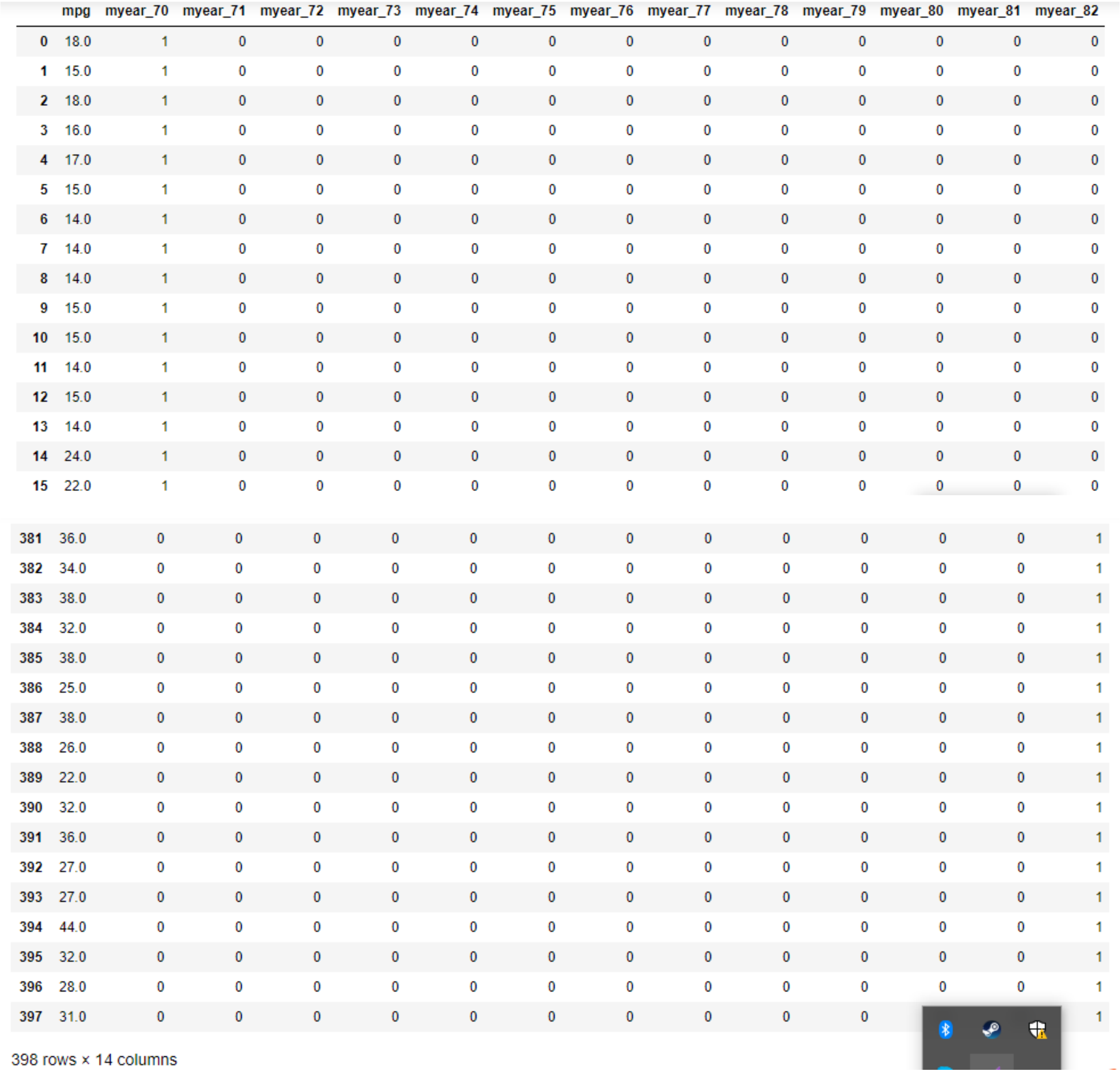
380	36.0	82
381	36.0	82
382	34.0	82
383	38.0	82
384	32.0	82
385	38.0	82
386	25.0	82
387	38.0	82
388	26.0	82
389	22.0	82
390	32.0	82
391	36.0	82
392	27.0	82
393	27.0	82
394	44.0	82
395	32.0	82
396	28.0	82
397	31.0	82

398 rows × 2 columns

then we get a new dataset with modelyear and mpg

We build a dictionary of the dataset ‘cars1’, and use”get_dummies” to classify the data.

```
dict=cars1.set_index('modelyear').T.to_dict('list')

dict

cars2=pd.get_dummies(cars1,prefix=['myear'],columns=['modelyear'])

cars2
```

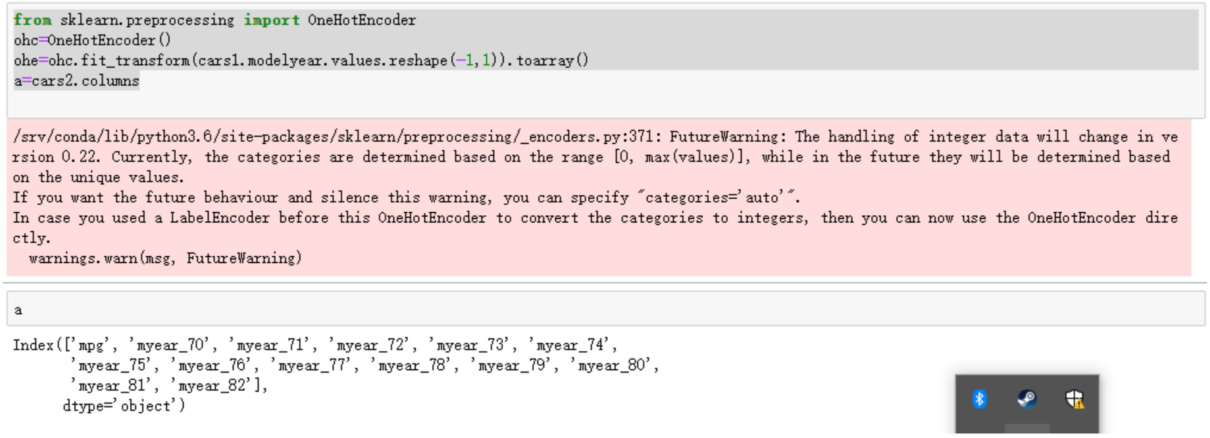


	mpg	myear_70	myear_71	myear_72	myear_73	myear_74	myear_75	myear_76	myear_77	myear_78	myear_79	myear_80	myear_81	myear_82
0	18.0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	15.0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	18.0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	16.0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	17.0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	15.0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	14.0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	14.0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	14.0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	15.0	1	0	0	0	0	0	0	0	0	0	0	0	0
10	15.0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	14.0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	15.0	1	0	0	0	0	0	0	0	0	0	0	0	0
13	14.0	1	0	0	0	0	0	0	0	0	0	0	0	0
14	24.0	1	0	0	0	0	0	0	0	0	0	0	0	0
15	22.0	1	0	0	0	0	0	0	0	0	0	0	0	0
381	36.0	0	0	0	0	0	0	0	0	0	0	0	0	1
382	34.0	0	0	0	0	0	0	0	0	0	0	0	0	1
383	38.0	0	0	0	0	0	0	0	0	0	0	0	0	1
384	32.0	0	0	0	0	0	0	0	0	0	0	0	0	1
385	38.0	0	0	0	0	0	0	0	0	0	0	0	0	1
386	25.0	0	0	0	0	0	0	0	0	0	0	0	0	1
387	38.0	0	0	0	0	0	0	0	0	0	0	0	0	1
388	26.0	0	0	0	0	0	0	0	0	0	0	0	0	1
389	22.0	0	0	0	0	0	0	0	0	0	0	0	0	1
390	32.0	0	0	0	0	0	0	0	0	0	0	0	0	1
391	36.0	0	0	0	0	0	0	0	0	0	0	0	0	1
392	27.0	0	0	0	0	0	0	0	0	0	0	0	0	1
393	27.0	0	0	0	0	0	0	0	0	0	0	0	0	1
394	44.0	0	0	0	0	0	0	0	0	0	0	0	0	1
395	32.0	0	0	0	0	0	0	0	0	0	0	0	0	1
396	28.0	0	0	0	0	0	0	0	0	0	0	0	0	1
397	31.0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	1

398 rows × 14 columns

Building an index

```
from sklearn.preprocessing import OneHotEncoder

ohc=OneHotEncoder()

ohe=ohc.fit_transform(cars1.modelyear.values.reshape(-1,1)).toarray()

a=cars2.columns
```



```
from sklearn.preprocessing import OneHotEncoder
ohc=OneHotEncoder()
ohe=ohc.fit_transform(cars1.modelyear.values.reshape(-1,1)).toarray()
a=cars2.columns
```

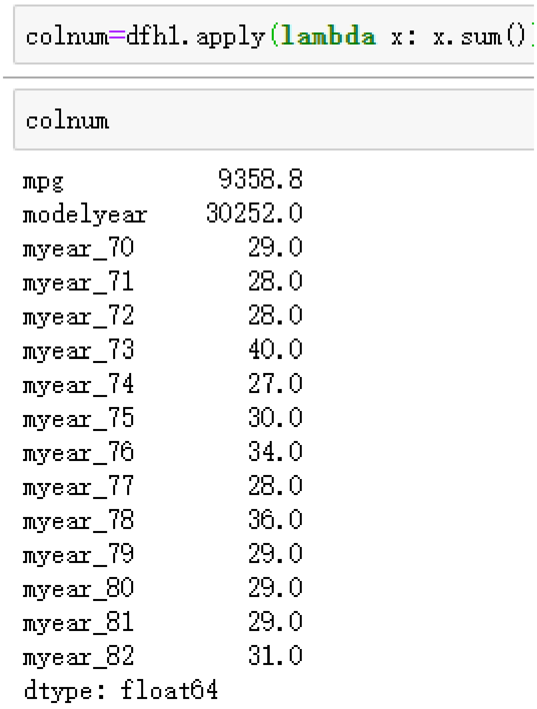
```
/srv/conda/lib/python3.6/site-packages/sklearn/preprocessing/_encoders.py:371: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, max(values)], while in the future they will be determined based
on the unique values.
If you want the future behaviour and silence this warning, you can specify "categories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the categories to integers, then you can now use the OneHotEncoder dire
ctly.
  warnings.warn(msg, FutureWarning)
```

```
a
```

```
Index(['mpg', 'myear_70', 'myear_71', 'myear_72', 'myear_73', 'myear_74',
       'myear_75', 'myear_76', 'myear_77', 'myear_78', 'myear_79', 'myear_80',
       'myear_81', 'myear_82'],
      dtype='object')
```

Calculating the sum of the numbers and the whole number of data ,modify them

```
dfOneHot=pd.DataFrame(ohe,columns=a[1:])

dfOneHot

dfh1=pd.concat([cars1,dfOneHot],axis=1)

dfh1

colnum=dfh1.apply(lambda x: x.sum())
```



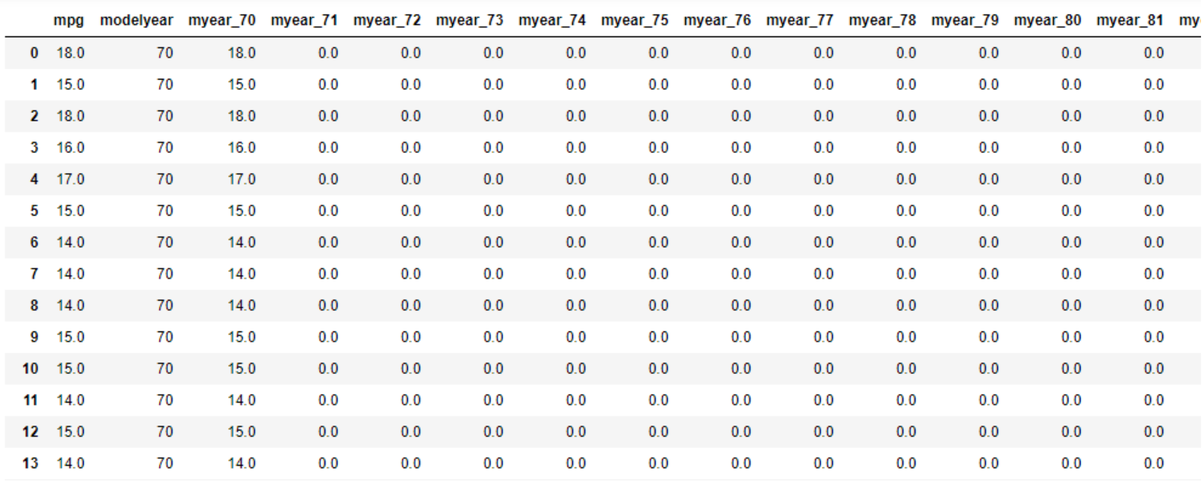
```
colnum=dfh1.apply(lambda x: x.sum()
```

```
colnum
```

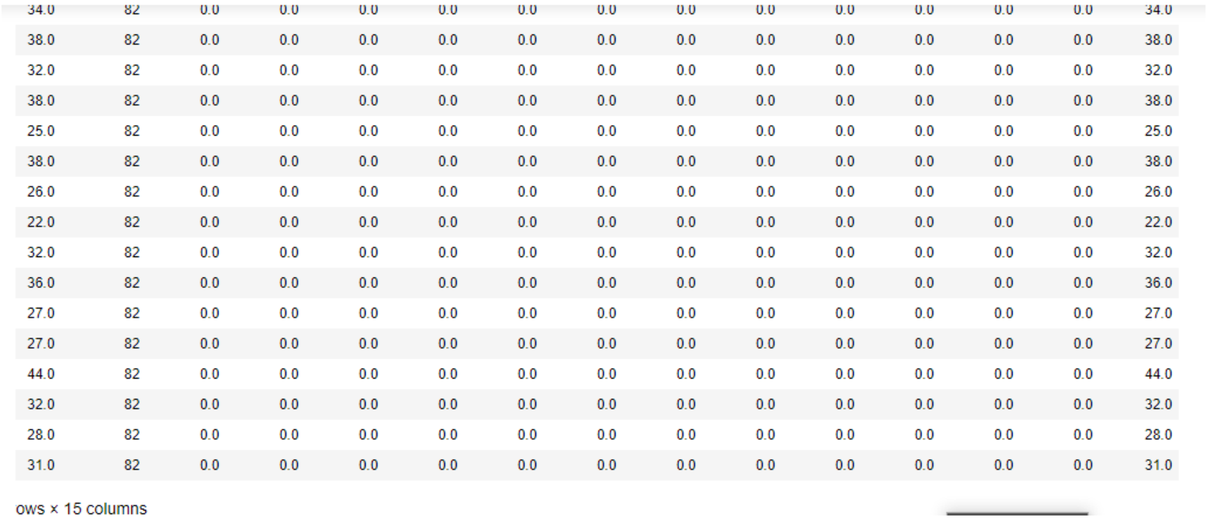
```
mpg            9358.8
modelyear     30252.0
myear_70         29.0
myear_71         28.0
myear_72         28.0
myear_73         40.0
myear_74         27.0
myear_75         30.0
myear_76         34.0
myear_77         28.0
myear_78         36.0
myear_79         29.0
myear_80         29.0
myear_81         29.0
myear_82         31.0
dtype: float64
```

Using a while to multipy every column in the dataset with mpg. So that ,we can replace number of “year” to the mpg of them.

```
i=2
while i<15:
  dfh1.iloc[:,i]=dfh1.iloc[:,i]*mpg
    i=i+1;
dfh1
```



	mpg	modelyear	myear_70	myear_71	myear_72	myear_73	myear_74	myear_75	myear_76	myear_77	myear_78	myear_79	myear_80	myear_81	my
0	18.0	70	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	15.0	70	15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	18.0	70	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	16.0	70	16.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	17.0	70	17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	15.0	70	15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	14.0	70	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	14.0	70	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	14.0	70	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	15.0	70	15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	15.0	70	15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
11	14.0	70	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12	15.0	70	15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
13	14.0	70	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	



34.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	34.0
38.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.0
32.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32.0
38.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.0
25.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	25.0
38.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.0
26.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	26.0
22.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22.0
32.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32.0
36.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	36.0
27.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27.0
27.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27.0
44.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	44.0
32.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32.0
28.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	28.0
31.0	82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	31.0

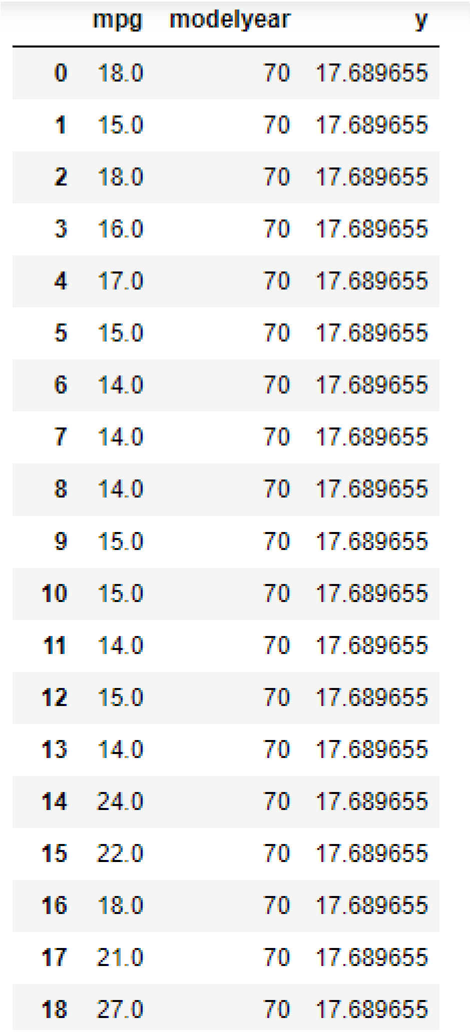
ows × 15 columns

Calculate the sum of each column and average

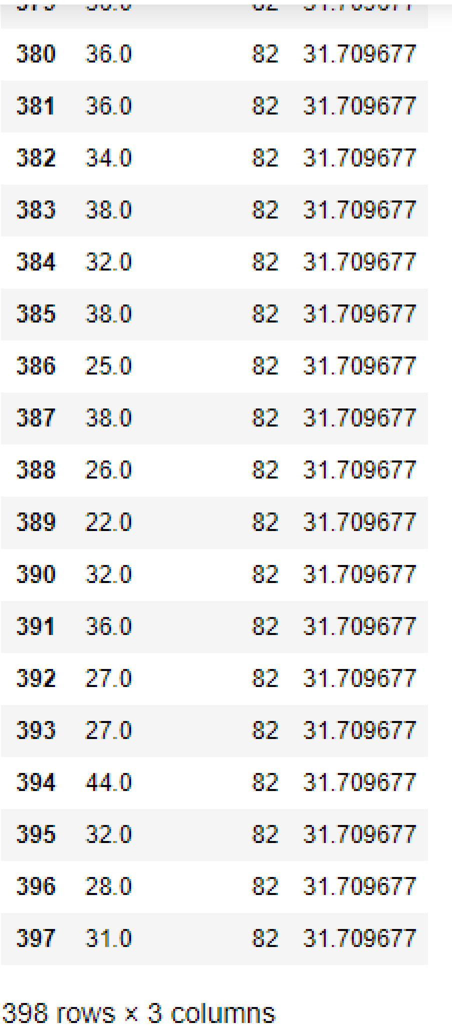
```
colsum=dfh1.apply(lambda x: x.sum())
average=colsum/colnum
average=average[2:]
```

Using a for loop to replace every column to mpg.

```
for i in mymodelyear:
  dict[i]=average[i-70]
dict
cars1['y']=cars1.modelyear.map(dict)
cars1
```



	mpg	modelyear	y
0	18.0	70	17.689655
1	15.0	70	17.689655
2	18.0	70	17.689655
3	16.0	70	17.689655
4	17.0	70	17.689655
5	15.0	70	17.689655
6	14.0	70	17.689655
7	14.0	70	17.689655
8	14.0	70	17.689655
9	15.0	70	17.689655
10	15.0	70	17.689655
11	14.0	70	17.689655
12	15.0	70	17.689655
13	14.0	70	17.689655
14	24.0	70	17.689655
15	22.0	70	17.689655
16	18.0	70	17.689655
17	21.0	70	17.689655
18	27.0	70	17.689655



380	36.0	82	31.709677
381	36.0	82	31.709677
382	34.0	82	31.709677
383	38.0	82	31.709677
384	32.0	82	31.709677
385	38.0	82	31.709677
386	25.0	82	31.709677
387	38.0	82	31.709677
388	26.0	82	31.709677
389	22.0	82	31.709677
390	32.0	82	31.709677
391	36.0	82	31.709677
392	27.0	82	31.709677
393	27.0	82	31.709677
394	44.0	82	31.709677
395	32.0	82	31.709677
396	28.0	82	31.709677
397	31.0	82	31.709677

398 rows × 3 columns