```
import csv
```

```
%precision 2
```

```
'%.2f'
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803

Enter your authorization code:
 · · · · · · · · · · ·
Mounted at /content/gdrive
```

```
with open('/content/gdrive/My Drive/MyFile/car.csv') as csvfile:
    cars=list(csv.DictReader(csvfile))
```

```
cars[:3]
```

```
[OrderedDict([('', '1'),
              ('manufacturer', 'audi'),
              ('model', 'a4'),
              ('displ', '1.8'),
              ('year', '1999'),
              ('cyl', '4'),
              ('trans', 'auto(l5)'),
              ('drv', 'f'),
              ('cty', '18'),
              ('hwy', '29'),
              ('fl', 'p'),
              ('class', 'compact')]),
 OrderedDict([('', '2'),
              ('manufacturer', 'audi'),
              ('model', 'a4'),
              ('displ', '1.8'),
              ('year', '1999'),
              ('cyl', '4'),
              ('trans', 'manual(m5)'),
              ('drv', 'f'),
              ('cty', '21'),
              ('hwy', '29'),
              ('fl', 'p'),
              ('class', 'compact')]),
 OrderedDict([('', '3'),
              ('manufacturer', 'audi'),
              ('model', 'a4'),
              ('displ', '2'),
              ('year', '2008'),
              ('cyl', '4'),
              ('trans', 'manual(m6)'),
              ('drv', 'f'),
              ('cty', '20'),
              ('hwy', '31'),
              ('fl', 'p'),
              ('class', 'compact')])]
```

```
len(cars)
```

```
234
```

```
cars[0].keys()
```

```
odict_keys(['', 'manufacturer', 'model', 'displ', 'year', 'cyl', 'trans', 'drv', 'cty', 'hwy'
```

```
sum(float(d['cty']) for d in cars) / len(cars)
```

```
16.86
```

```
sum(float(d['hwy']) for d in cars) / len(cars)
```

```
23.44
```

```
cylinder = set(d['cyl'] for d in cars) # set은 db의 distinct 기능
cylinder
```

```
{'4', '5', '6', '8'}
```

```
CtyMpgByCyl=[]


for c in cylinder:
    summpg=0
    cyltypecount=0
    for d in cars:
        if d['cyl'] == c:
            summpg+=float(d['cty'])
            cyltypecount+=1
    CtyMpgByCyl.append((c, summpg/cyltypecount))# 기통과 연비

CtyMpgByCyl.sort(key=lambda x:x[0])
CtyMpgByCyl
```

```
[('4', 21.01), ('5', 20.50), ('6', 16.22), ('8', 12.57)]
```

```
vehicleclass = set(d['class'] for d in cars)
vehicleclass
```

```
{'2seater', 'compact', 'midsize', 'minivan', 'pickup', 'subcompact', 'suv'}
```

```
HwyMpgByClass=[]
for c in vehicleclass:
    summpg=0
    cnt=0
    for d in cars:
        if d['class'] == c:
            summpg+=float(d['hwy'])
            cnt+=1
    HwyMpgByClass.append((c,summpg/cnt))
HwyMpgByClass.sort(key=lambda x: x[-1])
HwyMpgByClass
```

```
[('pickup', 16.88),
 ('suv', 18.13),
 ('minivan', 22.36),
 ('2seater', 24.80),
 ('midsize', 27.29),
 ('subcompact', 28.14),
 ('compact', 28.30)]
```

```
import datetime as dt
import time as tm
tm.time()
dtnow= dt.datetime.fromtimestamp(tm.time())


tm.time()
```

```
1585313736.35
```

```
dtnow
```

```
datetime.datetime(2020, 3, 27, 12, 55, 48, 882484)
```

```
dtnow.year
```

```
2020
```

```
dtnow.day
```

```
27
```

```
diff = dt.timedelta(days=100)
diff
```

```
datetime.timedelta(100)
```

```
today = dt.date.today()
```

```
today- diff
```

```
datetime.date(2019, 12, 18)
```

```
today>today-diff
```

```
True
```

```
class Person:
    department = 'School of Information'

    def set_name(self,new_name):
        self.name=new_name
```

```
person = Person()
```

```
person.set_name('kimdh')
```

```
print('{} in department {}'.format(person.name,person.department))
```

```
kimdh in department School of Information
```

```
store1=[10.00,11.00, 12.34, 2.34]
store2=[9.00,11.10,12.34, 2.01]
```

```
cheapest=map(min,store1,store2)
```

```
for i in cheapest:
    print(i)
```

```
9.0
11.0
12.34
2.01
```

```
import numpy as np
```

```
mylist = [1,2,3]
x= np.array(mylist)
```

```
x
```

```
array([1, 2, 3])
```

```
y=np.array([4,5,6])
y
```

```
array([4, 5, 6])
```

```
m= np.array([[7,8,9],[10,11,12]])
```

```
m
```

```
array([[ 7,  8,  9],
       [10, 11, 12]])
```

```
m.shape
```

```
(2, 3)
```

```
n= np.arange(0,30,2)
n
```

```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28])
```

```
n=n.reshape(3,5)
n
```

```
array([[ 0,  2,  4,  6,  8],
       [10, 12, 14, 16, 18],
       [20, 22, 24, 26, 28]])
```

```
o=np.linspace(0,4,9)
o
```

```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. ])
```

```
o.resize(3,3)
```

```
o
```

```
array([[0. , 0.5, 1. ],
       [1.5, 2. , 2.5],
       [3. , 3.5, 4. ]])
```

```
np.ones((4,3))
```

```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

```
np.eye(4)
```

```
array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 0., 0., 1.]])
```

```
np.zeros((2,3))
```

```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

```
y
```

```
array([4, 5, 6])
```

```
np.diag(y)
```

```
array([[4, 0, 0],
       [0, 5, 0],
       [0, 0, 6]])
```

```
np.repeat([1,2,3],3)
```

```
array([1, 1, 1, 2, 2, 2, 3, 3, 3])
```

```
np.array([1,2,3]*3)
```

```
array([1, 2, 3, 1, 2, 3, 1, 2, 3])
```

```
p=np.ones([2,3], int)
p
```

```
array([[1, 1, 1],
       [1, 1, 1]])
```

```
p1= np.vstack([p,2*p])
```

```
p1
```

```
array([[1, 1, 1],
       [1, 1, 1],
       [2, 2, 2],
       [2, 2, 2]])
```

```
p2=np.hstack([p,2*p])
p2
```

```
array([[1, 1, 1, 2, 2, 2],
       [1, 1, 1, 2, 2, 2]])
```

```
x, y
```

```
(array([1, 2, 3]), array([4, 5, 6]))
```

```
x+y
```

```
array([5, 7, 9])
```

```
x-y
```

```
array([-3, -3, -3])
```

```
x*y
```

```
array([ 4, 10, 18])
```

```
x/y
```

```
array([0.25, 0.4 , 0.5 ])
```

```
x**2
```

```
array([1, 4, 9])
```

```
x.dot(y)
```

```
32
```

```
z=np.array([y,y**2])
z
```

```
array([[ 4,  5,  6],
       [16, 25, 36]])
```

```
z.shape
```

```
(2, 3)
```

```
z.T
```

```
array([[ 4, 16],
       [ 5, 25],
       [ 6, 36]])
```

```
z.T.shape
```

```
(3, 2)
```

```
z.dtype
```

```
dtype('int64')
```

```
z=z.astype('f')
z.dtype
```

```
dtype('float32')
```

```
z.min()
```

```
4.0
```

```
z.argmin()
```

```
0
```