dem0

January 27, 2020

Let us do some basic numpy exercises.

[0]:
```
import numpy as np
```

Numpy lets us create and manipulate vector arrays using basic linear algebra routines

[24]:
```
x = np.array([1,2,3])
print(x)
```

```
[1 2 3]
```

[25]:
```
x = np.arange(10)
print(x)
x = np.arange(2,7)
print(x)
x = np.arange(-5,5,2)
print(x)
```

```
[0 1 2 3 4 5 6 7 8 9]
[2 3 4 5 6]
[-5 -3 -1  1  3]
```

[26]:
```
y = np.arange(0,5.1,0.5)
print(y)
```

```
[0.  0.5 1.  1.5 2.  2.5 3.  3.5 4.  4.5 5. ]
```

Let us now do a couple of quick exercises. How to enumerate the array 2,4,...,20? 30,20,10,0,10,20,30?

[27]:
```
y = np.arange(2,21,2)
print(y)
```

```
[ 2  4  6  8 10 12 14 16 18 20]
```

[28]:
```
z = np.arange(-30,31,10)
print(np.abs(z))
```

```
[30 20 10  0 10 20 30]
```

Indexing can be a bit funny.

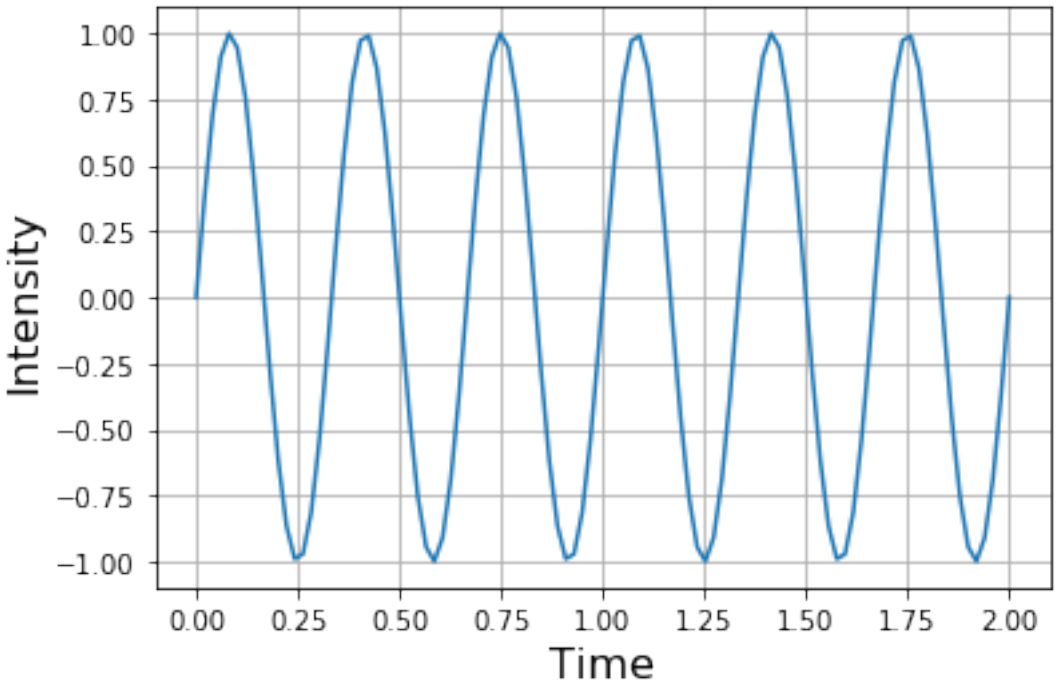
[29]:
```
y1 = y[2:5]
y2 = y[-1]
print(y1)
print(y2)
```

```
[ 6  8 10]
20
```

Let's now plot stuff. A popular plotting library is matplotlib.

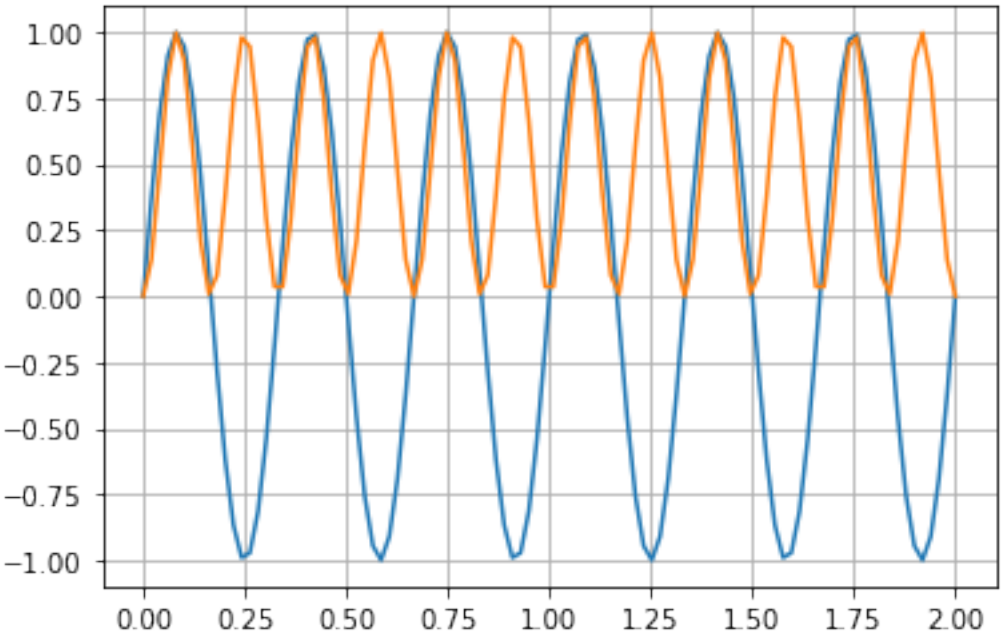
[30]:
```
import matplotlib.pyplot as plt
f = 3
t = np.linspace(0,2,100)
x = np.sin(2*np.pi*f*t)

plt.plot(t,x)
plt.grid()
plt.xlabel('Time', fontsize=16)
plt.ylabel('Intensity', fontsize=16)
```

[30]:
```
Text(0, 0.5, 'Intensity')
```



You can plot multiple curves at once.

[31]:
```
y = x**2
plt.plot(t,x)
plt.plot(t,y)
plt.grid()
```



OK, enough. Let's now do some data science (TM).

Pandas is a nice library that supports basic data analysis (reading and writing from files, querying, and visualization.

[32]:
```
import pandas as pd

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
 ↪auto-mpg/auto-mpg.data')
df.head(6)
```

[32]:

	18.0	8	307.0	130.0	3504.	12.0	70	1\t"chevrolet chevelle malibu"
0	15.0	8	350.0	165.0	3693.	11…		
1	18.0	8	318.0	150.0	3436.	11…		
2	16.0	8	304.0	150.0	3433.	12…		
3	17.0	8	302.0	140.0	3449.	10…		
4	15.0	8	429.0	198.0	4341.	10…		
5	14.0	8	454.0	220.0	4354.	9…		

Not delimited correctly! Let's use the correct names.

[0]:
```
names = ['mpg', 'cylinders','displacement', 'horsepower',
         'weight', 'acceleration', 'model year', 'origin', 'car name']
```

[34]:
```
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
↪auto-mpg/auto-mpg.data',
                 header=None,delim_whitespace=True,names=names,na_values='?')
df.head(6)
```

[34]:

	mpg	cylinders	displacement	…	model year	origin	car name
0	18.0	8	307.0	…	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	…	70	1	buick skylark 320
2	18.0	8	318.0	…	70	1	plymouth satellite
3	16.0	8	304.0	…	70	1	amc rebel sst
4	17.0	8	302.0	…	70	1	ford torino
5	15.0	8	429.0	…	70	1	ford galaxie 500

[6 rows x 9 columns]

[35]:
```
df.shape
```

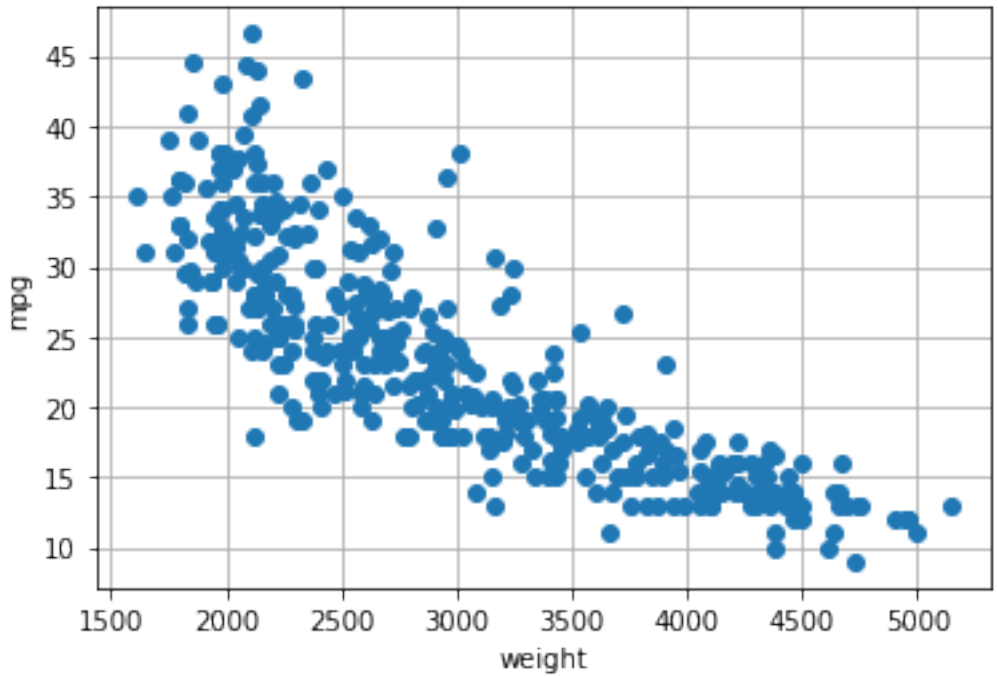
[35]:
```
(398, 9)
```

[36]:
```
df.columns.to_list()
```

[36]:
```
['mpg',
 'cylinders',
 'displacement',
 'horsepower',
 'weight',
 'acceleration',
 'model year',
 'origin',
 'car name']
```

[37]:
```
x = np.array(df['weight'])
y = np.array(df['mpg'])
plt.plot(x,y,'o')
plt.grid()
plt.xlabel('weight')
plt.ylabel('mpg')
```

[37]: Text(0, 0.5, 'mpg')



OK, now that we know how to load and visualize the data, let's do some analysis. We can extract individual data features and do some basic statistics using numpy.

[38]:
```
mx = np.mean(x)
my = np.mean(y)
print(mx)
print(my)
```

```
2970.424623115578
23.514572864321607
```

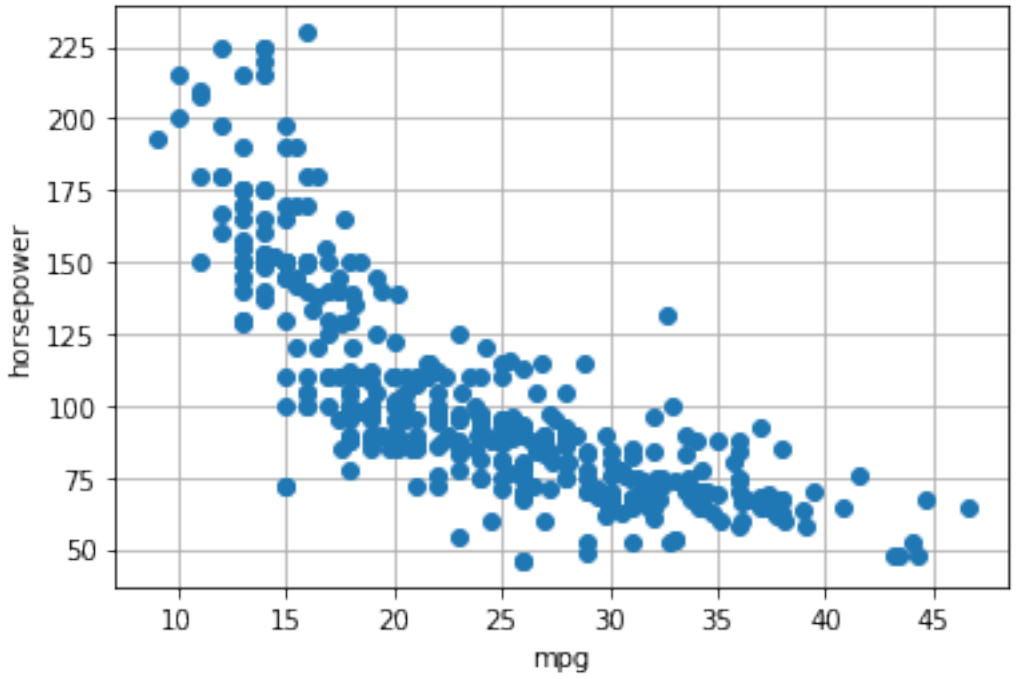
[39]:
```
z = np.array(df['horsepower'])
np.mean(z)
```

[39]: nan

Ouch! Some hp values are missing. Let's drop those rows and retabulate.

[40]:
```
df1 = df[['mpg','horsepower']]
df2 = df.dropna()
x = np.array(df2['mpg'])
y = np.array(df2['horsepower'])
plt.plot(x,y,'o')
```

```
plt.xlabel('mpg')
plt.ylabel('horsepower')
plt.grid()
```



That's all! Let's save our work.