lab-4-NFSU

March 6, 2023

[2]:
```python
import numpy as np
```

Unit 1 (as stated in NFSU syllabus) Introduction to mathematics for ML and Python

- Working With Numpy, Pandas, Matplotlib [**Done in lab 1, 2**]
- Vector, Matrices [**Done in lab 1 ,2**] - revision today
- Linear Equation [**Done in lab 3**]
- Mean, Median, Mode, Standard Deviation, Variance [**Done with Numpy in Lab 1,2**] - revision today
- Probability
- Correlation [**Done in lab 3**] - revision today
- Regression - [**Done in lab 3**] - revision today
- Handling and representing data [**Done in lab 3**]

[11]:
```python
# Mean, Median, Mode,
x = np.array([11.5,12.5,12.8,16.3,17.8,19.2])

sum_x = 0
for i in x:
    sum_x = sum_x + i

print("The mean is = {}".format(sum_x/len(x)))
```

```
The mean is = 15.016666666666666
```

[15]:
```python
print("The mean is = {}".format(np.mean(x)))
```

```
The mean is = 15.016666666666666
```

[29]:
```python
x = np.array([11.5,12.5,12.8,16.3,17.8,19.2])
```

[30]:
```python
x
```

[30]:
```
array([11.5, 12.5, 12.8, 16.3, 17.8, 19.2])
```

[24]:
```python
(12.8+16.3)/2
```

[24]:
```
14.55
```

[28]:

```
np.median(x)
```

[28]:

```
16.3
```

[31]:

```
np.median(x) == (12.8+16.3)/2
```

[31]:

```
True
```

[42]:

```
# our variance
x_bar = np.mean(x)
sum_sqr = 0
for x_i in x:
    x_sqr = (x_i - x_bar)**2
    sum_sqr = sum_sqr + x_sqr
print("Variance of {} is {}".format(x,sum_sqr/len(x)))


# numpy's variance
print("Variance of {} is {}".format(x,np.var(x)))
```

```
Variance of [11.5 12.5 12.8 16.3 17.8 19.2] is 8.418055555555554
Variance of [11.5 12.5 12.8 16.3 17.8 19.2] is 8.418055555555554
```

[49]:

```
# standard deviation, our version
print("Standard Deviation of {} is {}".format(x,np.sqrt(sum_sqr/len(x))))

# standard deviation, numpy
print("Standard Deviation of {} is {}".format(x, np.std(x) ))
```

```
Standard Deviation of [11.5 12.5 12.8 16.3 17.8 19.2] is 2.9013885564597435
Standard Deviation of [11.5 12.5 12.8 16.3 17.8 19.2] is 2.9013885564597435
```

[50]:

```
# covariance
toyX = np.array([12, 13, 25, 39])
toyY = np.array([67, 45, 32, 21])

# Step 1: Find Mean
toyX_mean = np.mean(toyX)
toyY_mean = np.mean(toyY)

print(toyX_mean)
print(toyY_mean)
```

```
22.25
41.25
```

[59]:

```
# Step 2:
diffX = np.array([i-toyX_mean for i in toyX])
```

[60]:
```
# Step 2:
diffY = np.array([i-toyY_mean for i in toyY])
```

[61]:
```
diffX*diffY
```

[61]:
```
array([-263.9375,  -34.6875,  -25.4375, -339.1875])
```

[66]:
```
# Step 3
cov = np.sum((diffX*diffY))/(len(diffX))
```

[68]:
```
cov
```

[68]:
```
-165.8125
```

[69]:
```
# Correlation is standardised covariance
cov/(np.std(toyX)*np.std(toyY))
```

[69]:
```
-0.8851566660693809
```

[52]:
```
# find correlationn using numpy
np.corrcoef(toyX, toyY) # 1D variable
```

[52]:
```
array([[ 1.        , -0.88515667],
       [-0.88515667,  1.        ]])
```

[70]:
```
np.corrcoef?
```

[ ]:
```

```