In [1]:
```
import statistics
```

In [ ]:
```
# mean, median, mode
```

In [2]:
```
val = [5,3,6,8,12,5]
```

In [3]:
```
statistics.mean(val)
```

Out[3]: 6.5

In [4]:
```
statistics.median(val)
```

Out[4]: 5.5

In [5]:
```
statistics.mode(val)
```

Out[5]: 5

In [ ]:
```
# average and weighted average
```

In [6]:
```
import numpy as np
```

In [7]:
```
np.average(val)
```

Out[7]: 6.5

In [9]:
```
arr = np.arange(5)
arr
```

Out[9]: array([0, 1, 2, 3, 4])

In [11]:
```
w = np.arange(5,10)
w
```

Out[11]: array([5, 6, 7, 8, 9])

In [12]:
```
weighted_avg = np.average(arr,weights = w)
weighted_avg
```

Out[12]: 2.2857142857142856

In [13]:
```
# Spread : minimum, maximum, range, variance, std dev
```

In [14]:
```
data = [1,2,3,4,5]
```

In [15]:
```
# min

minimum = min(data)
minimum
```

Out[15]: 1

In [16]:
```
# max

maximum = max(data)
maximum
```

Out[16]: 5

In [17]:
```
# range
range_data = maximum - minimum
range_data
```

Out[17]: 4

In [18]:
```
# variance
```

In [19]:
```
from statistics import variance

sample_data = (1,2,4,5,7,8,9,11,12)
sample_var = variance(sample_data)
sample_var
```

Out[19]: 14.777777777777779

In [20]:
```
from statistics import stdev
sample_std = stdev(sample_data)
sample_std
```

Out[20]: 3.844187531556932

In [24]:
```
# covariance

arr1 = np.array([1,2,3])
arr2 = np.array([2,4,5])

covariance = np.cov(arr1,arr2)
covariance
```

Out[24]:
```
array([[1.        , 1.5       ],
       [1.5       , 2.33333333]])
```

In [26]:
```
# correlation coeff

x = np.arange(10,20)
y = np.array([2,1,4,5,8,12,34,67,18,96])

r = np.corrcoef(x,y)
r
```

Out[26]:
```
array([[1.        , 0.78100034],
       [0.78100034, 1.        ]])
```

In [28]:
```
import scipy.stats

scipy.stats.pearsonr(x,y)
```

Out[28]: (0.7810003430715909, 0.007652641631823555)

In [29]:
```
scipy.stats.spearmanr(x,y)
```

Out[29]: SpearmanrResult(correlation=0.9515151515151514, pvalue=2.279854920641689e-05)

In [30]:
```
# percentiles
```

In [31]:
```
d = [19,3,7,1,36]

```

In [38]:
```
# 25th percentile
np.percentile(d,25)
```

Out[38]: 3.0

In [33]:
```
# 50th percentile
np.percentile(d,50)
```

Out[33]: 7.0

In [34]:
```
# 75th percentile

np.percentile(d,75)
```

Out[34]: 19.0

In [ ]:
```
# Normal distribution
```

In [39]:
```
import matplotlib.pyplot as plt
from scipy.stats import norm
```

In [40]:
```
x = np.arange(-20,20,0.01)
x
```

Out[40]: array([-20. , -19.99, -19.98, ..., 19.97, 19.98, 19.99])

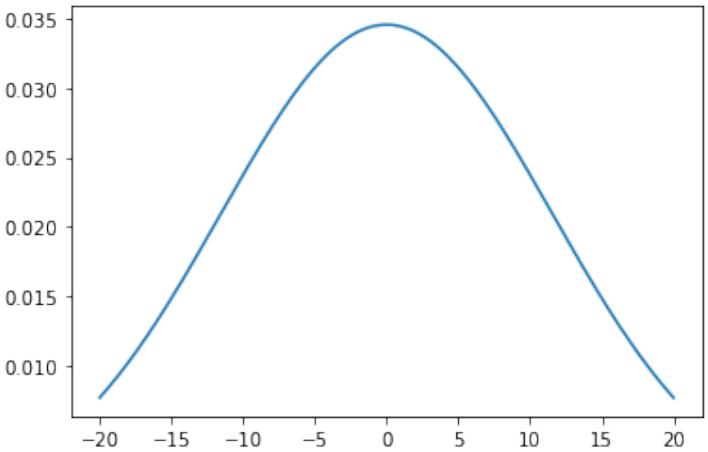
In [41]:
```
mean  = statistics.mean(x)
mean
```

Out[41]: -0.0049999999968743936

In [42]:
```
std = statistics.stdev(x)
std
```

Out[42]: 11.54844866926759

In [43]:
```
plt.plot(x,norm.pdf(x,mean,std))
plt.show()
```



In [44]:
```
1 np.percentile(x,75)
```

Out[44]:
```
9.99250000000469
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```