



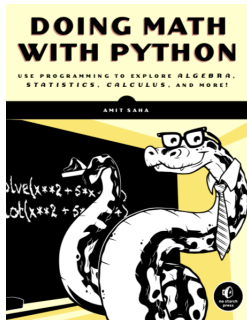
Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

Fall 2018
Week 12

Basic Stats

Mean
Median
Common
Elements
Mode
Range



Saha, Chapter 3: Describing Data with Statistics

- The Three-M's: Mean, Median, Mode
- Common Elements, Minimum & Maximum values, and Range

Basic Stats

Mean

Median

Common

Elements

Mode

Range

- The mean of the set $\{11, 12, 13\}$
 - $(11 + 12 + 13)/3 = 12$
- Could also use a list and the `sum()` function

Find the mean

```
num_list = [11,12,13]  
sum(num_list) / len(num_list)
```

Function for the mean

```
def calculate_mean(numbers_list):  
    print("  Values", numbers_list)  
    s_int = sum(numbers_list)  
    N_int = len(numbers_list)  
    # Calculate the mean  
    mean_flt = s_int/N_int  
    return mean_flt  
#end of calculate_mean()  
if __name__ == '__main__':  
    donations_list = [100, 60, 70, 900, 100,  
200, 500, 500, 503, 600, 1000, 1200]  
    mean_flt = calculate_mean(donations_list)  
    N_int = len(donations_list)  
    print('  The mean of the {0} values  
is {1}'.format(N_int, mean_flt))
```

Find the Mean With Built-In Functions

Basic Stats

Mean

Median

Common

Elements

Mode

Range

statistics - Basic statistics module.

DESCRIPTION

This module provides functions for calculating statistics of data, including averages, variance, and standard deviation.

Calculating averages

Function	Description
mean	Arithmetic mean (average) of data.
harmonic_mean	Harmonic mean of data.
median	Median (middle value) of data.
median_low	Low median of data.
median_high	High median of data.
median_grouped	Median, or 50th percentile, of grouped data.
mode	Mode (most common value) of data.

```
import statistics
statistics.mean([1,2,3])
```

Basic Stats

Mean

Median

Common

Elements

Mode

Range

1, 3, 3, **6**, 7, 8, 9

Median = **6**

1, 2, 3, **4**, **5**, 6, 8, 9

Median = $(4 + 5) \div 2$
= **4.5**

- The median is the value separating the higher half from the lower half of a data sample.

Basic Stats

Mean

Median

Common

Elements

Mode

Range

Median

First, arrange the observations in an ascending order.

If the number of observations (n) is **odd**:
the median is the value at position

$$\left(\frac{n+1}{2} \right)$$

If the number of observations (n) is **even**:

1. Find the value at position $\left(\frac{n}{2} \right)$
2. Find the value at position $\left(\frac{n+1}{2} \right)$
3. Find the average of the two values to get the median.

Basic Stats

Mean

Median

Common

Elements

Mode

Range

Function for the Median

```
''' Calculating the median '''
def calculate_median(numbers_list):
    # print(" calculate_mean()")
    N = len(numbers_list)
    numbers_list.sort()
    # Find the median
    if N % 2 == 0:
        # if N is even
        m1 = N/2
        m2 = (N/2) + 1
        # Convert to integer, match position
        m1 = int(m1) - 1
        m2 = int(m2) - 1
        median_int = (numbers_list[m1] + numbers_list[m2])/2
    else:
        m = (N+1)/2
        # Convert to integer, match position
        m = int(m) - 1
        median_int = numbers_list[m]
    return median_int

if __name__ == '__main__':
    donations_list = [100, 60, 70, 900, 100, 200, 500, 500, 503, 600, 1000, 1200]
    print(" Data:",donations_list)
    median_int = calculate_median(donations_list)
    N = len(donations_list)
    print(' Median donation over the last {0}
    days is {1}'.format(len(donations_list), median_int))
```


Basic Stats

Mean

Median

Common
Elements

Mode

Range

Simple Example

```
import statistics
statistics.median([1,2,3])
```

Another Quick Example with Random Data

```
import random, statistics
nums_list = []
for i in range(10):
    n = int(random.random() * 9 + 1)
    nums_list.append(n)
statistics.median(nums_list)
```

What is the Most Common Element?

Basic Stats

Mean

Median

Common
Elements

Mode

Range

What entry in the set is the most common?

```
simplelist = [4, 2, 1, 3, 4]
from collections import Counter
c = Counter(simplelist)
c.most_common() #[4, 2), (1, 1), (2, 1), (3, 1)]
```

What entry in the set is the most common?

```
c = Counter(['a','a','a','a','a','a','a','b'])
c.most_common() #[('a', 7), ('b', 1)]
```

- Contained in the output is the number of times that an element has been found.

Most Common Values in a List

Basic Stats

Mean

Median

Common
Elements

Mode

Range

- Print the number of times an Integer has occurred in list

```
from collections import Counter
scores_list = [7, 8, 9, 2, 10, 9,1,1,0]
print("  Data: ",scores_list)
x_colCount = Counter(scores_list)
type(x_colCount) # <class 'collections.Counter'>
print(" + One way to do it:\n")
print("  Value \t Count")
for i in x_colCount:
    print("    ",i,"\t",x_colCount[i])
print("\n + Another way to do it:\n")
for i in x_colCount.most_common():
    print("    ",i)
```

Most Common Values in a List

Basic Stats

Mean

Median

Common

Elements

Mode

Range

- Print the number of times a **Character** has occurred in list

```
from collections import Counter
scores_list = ['a','b','a','a','b','c']
print("  Data: ",scores_list)
x_colCount = Counter(scores_list)
type(x_colCount) # <class 'collections.Counter'>
print(" + One way to do it:\n")
print("  Value\tCount")
for i in x_colCount:
    print("    ",i,"\t",x_colCount[i])
print("\n + Another way to do it:\n")
for i in x_colCount.most_common():
    print("    ",i)
```

Basic Stats

Mean

Median

Common

Elements

Mode

Range

Function for the Mode

```
'''Calculating the mode'''
from collections import Counter
def calculate_mode(numbers_list):
    print("  Values: ",numbers_list)
    c = Counter(numbers_list)
    mode_int = c.most_common(1) #print first most common
    return mode_int[0][0]
#end of calculate_mode()
if __name__=='__main__':
    scores_list = [7, 8, 9, 2, 10, 9, 9, 9, 9, 4, 5, 6, 1, 5, 6, 7, 8, 6, 1, 10]
    print("  Set: ",scores_list)
    mode_int = calculate_mode(scores_list)
    print("  Mode: ",mode_int)
```

- The most common (most frequently occurring) data point from discrete or nominal data.

Sorry about the tiny print!

Dispersion

Basic Stats

Mean
Median
Common
Elements
Mode
Range

- *Dispersion*: a measurement of distance between its values and the mean of the data set.
- Three measurements of dispersion: range, variance, and standard deviation
- After finding the mean, one may want to know how *spread-out* the values are found (the, *Variance*)

What kind of distribution?

- The mean of 50 can come from two different distributions
 - $50 = (49 + 50 + 51)/3$
 - $50 = (82 + 23 + 45)/3$
- The **Range** is the maximum and minimum values of a data set.

Function for the Range

```
''' Finding the range '''
def find_range(numbers_list):
    print("  Values: ",numbers_list)
    lowest_int = min(numbers_list)
    highest_int = max(numbers_list)
    # Find the range
    r_int = highest_int - lowest_int # find distance
    return lowest_int, highest_int, r_int
#end of find_range()

if __name__ == '__main__':
    donations_list = [100, 60, 70, 900, 100, 200, 500, 500, 503, 600, 1000, 1200]
    lowest, highest, r = find_range(donations_list)
    print('  Lowest: {0} Highest: {1} Range: {2}'.format(lowest, highest, r))
```

- The most common (most frequently occurring) data point from discrete or nominal data.

Sorry about the tiny print!