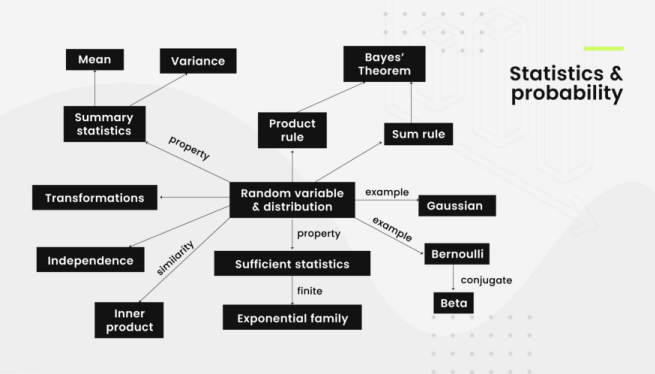
Data Science
Compilation: Yogesh Kulkarni

Descriptive Statistics

Statistics

Landscape: Statistics



(Ref: The NOT definitive guide to learning math for machine learning - Favio Vazquez)

Introduction to Descriptive Statistics

Descriptive Statistics

- Describes the data characteristics
- To make sense of the data
- To make rational decisions
- E.g. Demographics, clinical data.
- Measures of Central Tendencies
- Measures of Variability
- Measures of Shape

Why Descriptive Statistics?

- Population: the whole
- Sample: small subset of the population
- Gauging Population by examining traits of Sample.
- Example Question: Finding height of Americans?
- Not going to measure everyone height, but that in a **representative** sample.
- Example: Election sampling?

Why Descriptive Statistics?

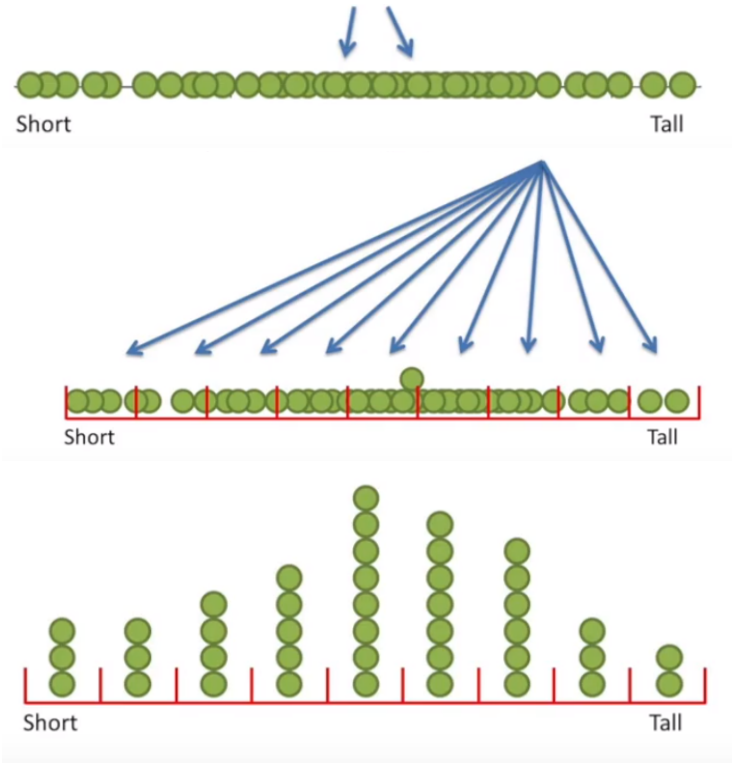
- To check the accuracy and precision of the process
- To reduce variability and improve process capability
- To know the truth about the real world

Basic Terms

Histogram

Example

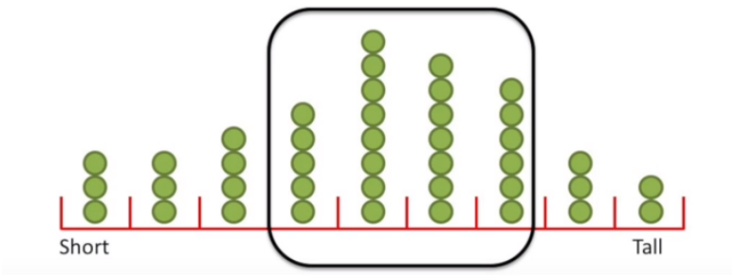
- Say, we are measuring height of people.
- Plotting them on X axis.
- The dots would look very crowded where there are many close or repetitive observations.
- Some dots get hidden.
- We can improve the visualization, by plotting frequency (number of occurrences) on Y axis.
- But in case of contiguous variable, like, exact measurements are rare. So we 'bin' them and measure occurrences.
- Thats Histogram.

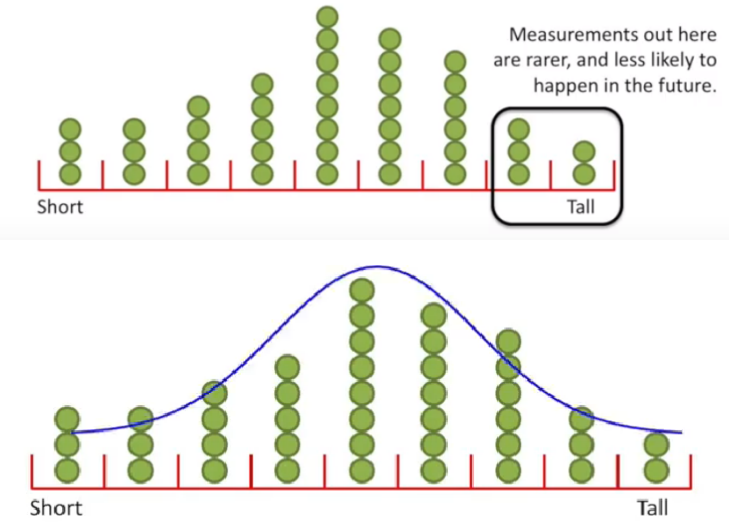


(Ref: StatQuest: What is a Histogram? - Josh Starmer)

Histogram

- Histogram can be used to predict probability of getting (future) measurements.
- Getting measurement (as shown in the box) in the middle region is more likely.
- Measurements at both the ends are rare.
- We can approximate this histogram of observations by a 'distribution'.
- Looks like 'Normal' distribution, or a bell-curve

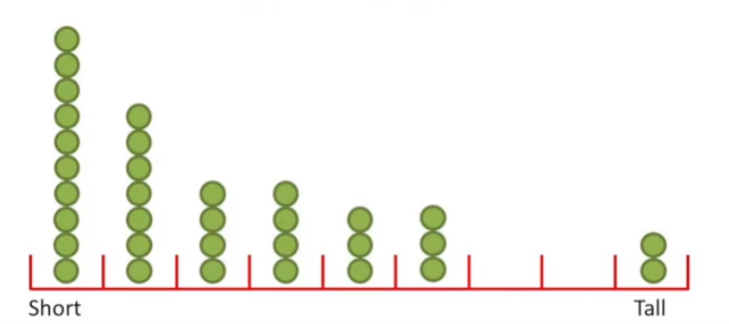


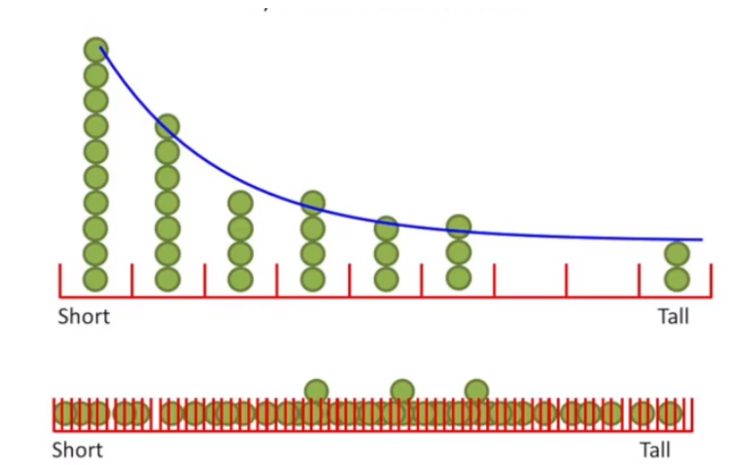


(Ref: StatQuest: What is a Histogram? - Josh Starmer)

Histogram

- If the frequency of measurements seem decreasing, it may be an exponential distribution.
- Binning criterion is critical. They can not be too narrow or too wide.
- Try different bin widths/formulas to plot a histogram.





(Ref: StatQuest: What is a Histogram? - Josh Starmer)

Descriptive Statistics

- Describes features of data sets using numbers
- Individual row: Data
- Full table: Dataset
- Purpose: Answer questions.

Mrs. Graham's 5th Grade Class	Scores on Spelling Test
Bella	43
Betty	45
Bobby	32
Bonnie	45
Booker	38
Boston	45
Botania	50
Boyle	45
Bunder	31

Questions

Mrs. Graham's 5th Grade Class	Scores on Spelling Test
Bella	43
Betty	45
Bobby	32
Bonnie	45
Booker	38
Boston	45
Botania	50
Boyle	45
Bunder	31

- What is Bobby's score?
- Out of? (Total # entries)
- Highest/Lowest scores?

Questions

Mrs. Graham's 5th Grade Class	Scores on Spelling Test
Bella	43
Betty	45
Bobby	32
Bonnie	45
Booker	38
Boston	45
Botania	50
Boyle	45
Bunder	31

- Class average?
- Most Common/frequent Score?
- Any other questions?

Numerical Measures

- Highest to Lowest Score: RANGE
- Most Common Score: MODE
- Average Score: MEAN
- Any other measures?

Descriptive Statistics

- Examines ALL data (not sample)
- Cannot generalize to other datasets

Descriptive Statistics Example

Descriptive Tasks

id	Home Owner	Marital Status	Annual Income	Defaulted Barrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Objective: Derive patterns, summarize underlying relationships
- More exploratory of current state

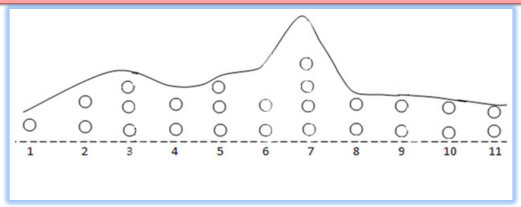
Data Example

1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

What sense it makes? Any pattern?

Visualize

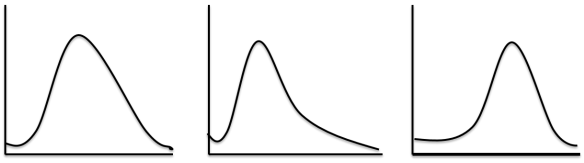
1	2	2	3	3	3	4	4	5
5	5	6	6	7	7	7	7	8
8	9	9	10	10	11	11		



Makes sense?

The Shape of The Distribution

Better to see



Symmetric? Skewed right/left?

Describing Data

Describing Data

- Univariate Analysis: Statistical moments (based on degree)
  - 1st degree: Central tendency: mean, median, mode
  - 2nd degree: Standard Deviation: how wide data is around mean
  - 3rd degree: Skewness: Asymmetric around mean
  - 4th degree: Kurtosis: shape of skewness.
- BiVariate Analysis: covariance, correlation
- MultiVariate Analysis

Univariate Analysis

- Measure of Central Tendency
- Measure of Spread
- Measure of Asymmetry
- Measure of Skewness

Measure of Central Tendency

Mean

- Measure the "location" of a set of values
- Mean is a very, very common measurement
- But is sensitive to outliers

$mean(x) = \bar{x} = 1/n \sum x_i$

Mean

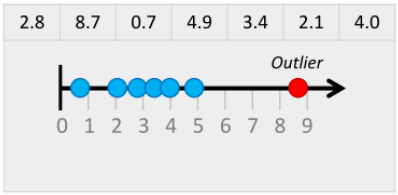
Data set from 25 subjects

1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Sum = 153
Mean = 153/25 = 6.12

Outliers

- Extreme data point
- May affect calculations
- Can occur in any given data set and in any distribution
- May indicate an experimental error or incorrect recording of data



Mean

Implement mean

```
def mean(datalist):
  :
  return m

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
result = mean(lst)
print("Mean : {}".format(result))
```

Mean

```
def mean(datalist):
  total = 0
  m = 0
  for item in datalist:
    total += item
  m = total / float(len(datalist))
  return m

Mean : 13.157894736842104
```

Median

- Commonly used instead of mean if outliers are present
- Median is the middle value
- if odd number of values are present; average of the two middle values if even number of values
- Not easily affected by outliers (extreme values).
- Always exists and unique.

$median(x) = x_{r=1} for \quad odd, 1/2(x_r + x_{r+1}) for \quad even$

Median

Data set from 25 subjects

1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Median = 6

Medians are less reliable: medians of samples drawn from same population vary more widely than sample means.

Median

Implement median

```
def median(datalist):
  :
  return m

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
result = median(lst)
print("Median : {}".format(result))
```

Median

```
def median(datalist):
    n = len(datalist)
    numsort = sorted(datalist)
    mid = n // 2
    m = 1
    if n % 2 == 0:
        lo = mid - 1
        hi = mid
        m = (numsort[lo] + numsort[hi])/2
    else:
        m = numsort[mid]
    return m

Median : 9
```

Mode

- The value that has the highest frequency.
- Requires no calculation, only counting
- Often used with categorical values.
- The mode (especially with discrete / continuous data) may reveal value that symbolizes a missing value.
- Not a stable measure : it depends only a few values
- May not exist
- May not be unique

Mode

Data set from 25 subjects

1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Median = 7

Mode

Implement mode

```
def mode(datalist):
  :
  return m

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
result = mode(lst)
print("Mode : {}".format(result))
```

Mode

```
def frequency_distribution(datalist):
  freqs = dict()
  for item in datalist:
    if item not in frees.keys():
      freqs[item] = 1
    else:
      freqs[item] += 1
  return freqs

def mode(datalist):
    d = frequency_distribution(datalist)
    print(d)
    most_often = 0
    m = 0
    for item in d.keys():
        if d[item] > most_often:
            most_often = d[item]
            m = item
    return m

Mode : 3
```

Mode

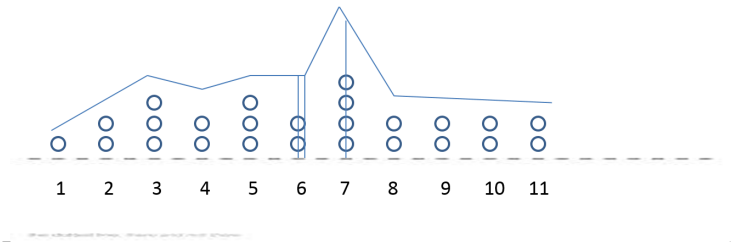
Another implementation. Counter returns dictionary of frquencies and values.

```
from collections import Counter

def mode2(x):
    counts = Counter(x)
    max_count = max(counts.values())
    return [x_i for x_i, count in counts.items() if count ==
     max_count]  # multiple modes are possible

Mode : [3]
```

Central Tendency

- Mean: Summarizes all the information in the data set
- Median: Splits the data sets into two halves: there are an equal number of values above and below it.
- Mode: The most common value in the data set.

Locating Central Tendency



Mean = 6.12
Median = 6
Mode = 7

Descriptive Statistics Exercise

Exercise

Our Data: Store as list of integers and calculate Mean, Median and Mode

```
lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
```

Descriptive Statistics Exercise

Exercise

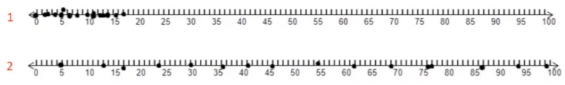
```
crater_diameter = [46, 51, 49, 82, 74, 63, 49, 70, 48, 47,
    79, 48, 52, 55, 49, 51, 58, 82, 72, 45]
print mean(crater_diameter)
print median(crater_diameter)
print mode(crater_diameter)
```

Code: *Result? 58.5,51.5,49*

Measure of Spread

Measure of Spread

In which example (below), the data is spread?



How do you quantify the spread?

Measure of Spread

- Range: The largest value minus the smallest value. Suffers from Outliers.
- Semi-Interquartile range: One half of the difference between the 75th percentile and the 25th percentile. Not affected by Outliers.
- Standard Deviation: The square root of the average of the squared deviations from the mean

Range

- Variation between the smallest and the largest values
- Can be misleading if most values are concentrated, but a few values are extreme

$range(x) = max(x) - min(x)$

Range

Data set from 25 subjects

1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Range = 11 -1 = 10

- A 'quick and easy' indication of variability
- No indication of dispersion within
- Unstable, as depends ONLY on Outliers/Extremes

Code: *Calculate and verify the answer*

Range

Implement my_range. It cannot be called as "range" is already there in Python, so a different name

```
def my_range(datalist):
  :
  return min, max, diff

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
min,max,diff = my_range(lst)
print("Range: Min {}, Max {}, Diff {}".format(min,max,diff))
```

Range

```
def my_range(abclist):
    smallest = abclist[0]
    largest = abclist[0]
    range_of_values = 0
    for item in abclist[1:]:
        if item < smallest:
            smallest = item
        elif item > largest:
            largest = item
    range_of_values = largest - smallest
    return smallest, largest, range_of_values

Range: Min 2, Max 44, Diff 42
```

Range

min max functions are available on list

```
def my_range2(x):
  return max(x) - min(x)

diff = my_range2(lst)
print("Range: {}".format(diff))
Range: 42
```

Percentiles

- For ordered data, percentile is useful.
- Given an ordinal or continuous attribute x and a number p between 0 and 100, the pth percentile x_p is a value of x such that p% of the observed values are less than x_p.
- Example: the 75th percentile is the value such that 75% of all values are less than it.

Quantile

Quantile can be of any number between 0 to 1. Quartiles are about quarters so they are quantiles of 0.25, 0.5,0.75
Quantiles are cut points in set of data. They can represent the bottom ten percent of the data or the top 75% or any % from 0 to 100.

Quantile

Implement quantile

```
def quantile(datalist):
  :
  return q

def interquartile_range(x):
  :
  return iqr

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
result1 = quantile(lst,0.10)
result2 = quantile(lst,0.25)
result3 = quantile(lst,0.75)
result4 = quantile(lst,0.90)
result5 = interquartile_range(lst)
print("Q10 {}, Q25 {}, Q50 {} Q90 {} IQR
    {}".format(result1,result2,result3,result4,result5))
```

Quantile

```
def quantile(datalist,num):
    index = int(num * len(datalist)) # slicing parameter
    return sorted(datalist)[index]
    # For values :
    # if num > .5:
    #     return sorted(datalist)[index:]
    # else:
    #     return sorted(datalist)[:index]

def interquartile_range(x):
    return  quantile(x, 0.75) - quantile(x, 0.25)

# Q10 [2], Q25 [2, 3, 3, 3], Q50 [21, 22, 23, 42, 44] Q90
    [42, 44]
Q10 3, Q25 3, Q50 21 Q90 42 IQR 18
```

Semi-Interquartile Range

- Quartiles are Quantiles at 25% and 75%.
- Inter Quartile Range (IQR) is between 25% and 75%.
- More resistant to extreme values than the range
- Does not utilize all the values in the data or set for its computation
- If small, the values are concentrated near the median

Semi-Interquartile Range

- 75th percentile: the value in the date set which is exceeded by 75% of the total number of items in the set
- 25 x (0.75) = 18.75
- 18.75 : rank of the 75th percentile
- 18th and 19th items, both 8
- 75th percentile = 8

Data Set from 25 subjects				
1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Semi-Interquartile Range

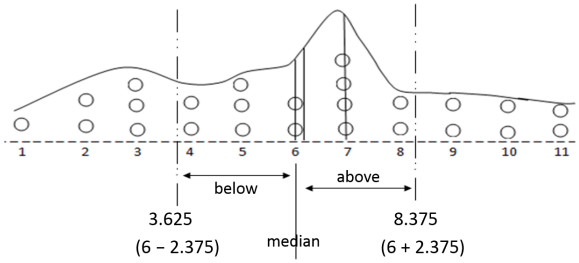
- 25th percentile: the value in the date set which is exceeded by 25% of the total number of items in the set
- 25 x (0.25) = 6.25
- 6.25 : rank of the 25th percentile
- 6th item = 3 and 7th item = 4
- 25th percentile = 3 + (0.25)(4-3)
- 25th percentile = 3.25

Data Set from 25 subjects				
1	2	2	3	3
3	4	4	5	5
5	6	6	7	7
7	7	8	8	9
9	10	10	11	11

Code: *Calculate and verify the answer*

Semi-Interquartile Range

- 75th percentile = 8
- 25th percentile = 3.25
- SIQR =1/2 (8 - 3.25)
- SIQR = 2.375
- Semi-interquartile range = 2.375

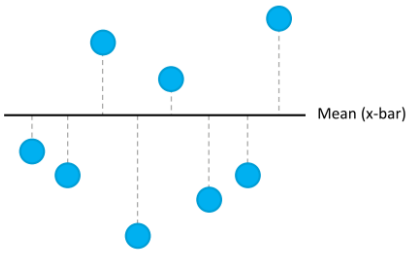


Standard Deviation

- How far each value if from the mean
- Uses all the values in the data for its computation
- If small, the values are concentrated near the mean.
- If LARGE, the values are scattered widely about the mean
- z score: how many std deviations from the mean.

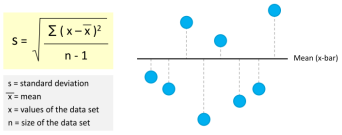
$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

s = standard deviation
$\bar{x}$ = mean
x = values of the data set
n = size of the data set



Variance, Standard Deviation

Implement variance and standard deviation.




```
def variance(datalist):
  :
  return v

def std_dev(datalist):
  :
  return s

lst = [9,3,7,2,7,10,23,44,12,42,19,11,22,5,3,4,3,21,3]
result1 = variance(lst)
result2 = std_dev(lst)
print("Variance {}, Std Dev {}".format(result1,result2))
```

Standard Deviation

To reparametrize Covariance value between -1 and 1, need to divide by std devs. Empirical Rule for symmetric bell-shaped distributions

- About 68% of the values will lie within 1 standard deviation of the mean
- About 95% of the values will lie Within 2 standard deviation
- About 99.7% of the values will lie within 3 standard deviation of the mean

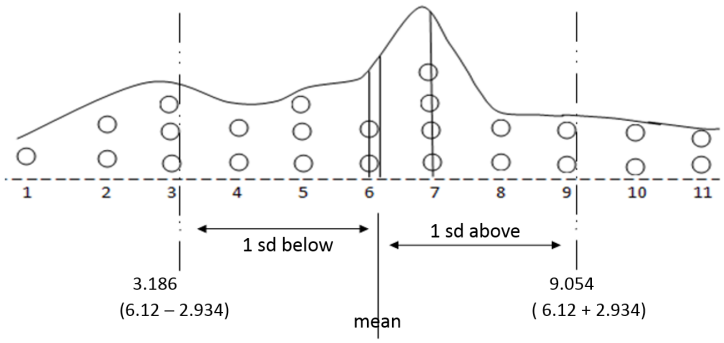
$variance(x) = s_x^2 = 1/(n-1)\sum(x_i - \bar{x})^2$ $sd(x) = s_x = \sqrt{1/(n-1)\sum(x_i - \bar{x})^2}$

Variance, Standard Deviation

```
def de_mean(x):
  """translate x by subtracting its mean"""
  x_bar = mean(x)
  return  [x_i - x_bar for x_i in x]

def sum_of_squares(diffs):
  sum_of_squares = 0
  for df in diffs:
          sum_of_squares += (df) ** 2
     return sum_of_squares

def variance(x):
  """assumes x has at least two elements"""
  n = len(x)
  deviations = de_mean(x)
  return  sum_of_squares(deviations) / (n - 1)

def std_dev(anotherlist):
  std_dev = variance(anotherlist) ** 0.5
  return std_dev

Variance 158.36257309941527, Std Dev 12.584219208970229
```

Standard Deviation

Standard Deviation = 2.934



Descriptive Statistics Exercise

Exercise

```
crater_diameter = [46, 51, 49, 82, 74, 63, 49, 70, 48, 47,
     79, 48, 52, 55, 49, 51, 58, 82, 72, 45]

print range_min_max(crater_diameter)
print avg_dev(crater_diameter)
print variance(crater_diameter)
print std_dev(crater_diameter)
```

Code: *Result?(45,82,37),11.25,161.45,12.7062*

Exercise

Find the mean, median, range and standard deviation for the following set of data:

2.8, 8.7, 0.7, 4.9, 3.4, 2.1 & 4.0

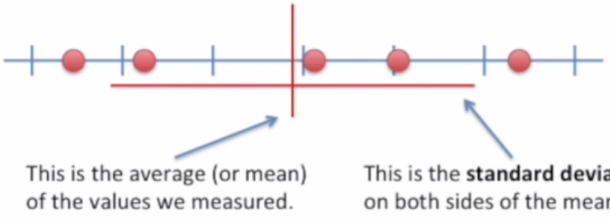
Exercise

Find the mean, median, range and standard deviation for the following set of data:

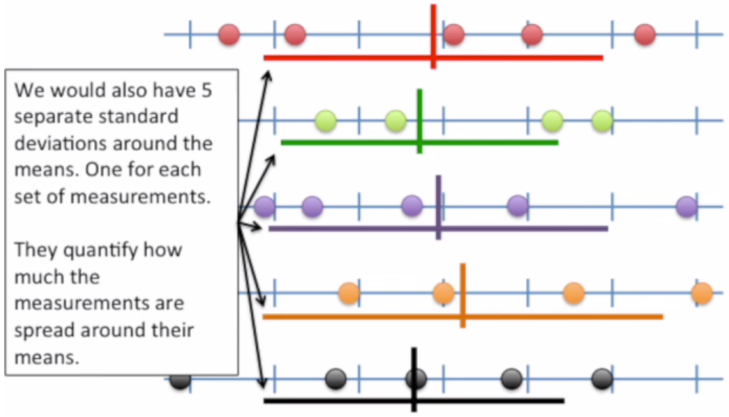
21	19	20	24	23	21	26	23
25	24	19	19	21	19	25	19
23	23	15	22	23	20	14	20
15	19	20	21	17	15	16	19
13	17	19	17	22	20	18	16
17	18	21	21	17	20	21	21
21	17	17	19	21	22	25	20
19	20	24	28	26	26	25	24

Difference between Standard Deviation and Standard Error

- For a set of normally distributed observations you have mean and standard deviation.
- If you do this for different samples, you get their own respective means and standard deviations.



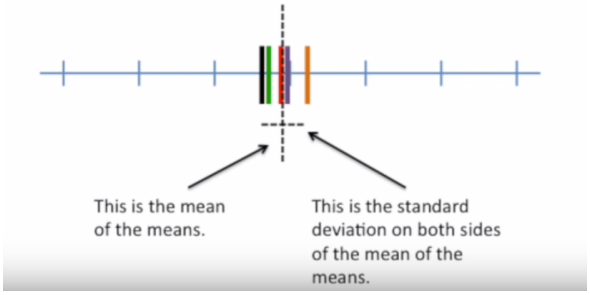
It quantifies of how much the data are spread out.



(Ref: StatQuest: Difference between Standard Deviation and Standard Error - Josh Starmer)

Difference between Standard Deviation and Standard Error

- Plotting those sample means, and sample standard deviations, can form another (meta?) distribution
- Standard deviation of this meta distribution is called Standard Error

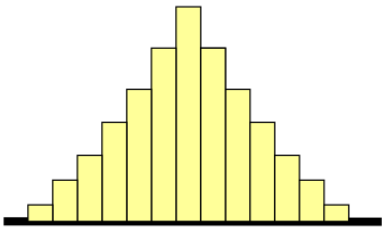


(Ref: StatQuest: Difference between Standard Deviation and Standard Error - Josh Starmer)

Measure of Asymmetry

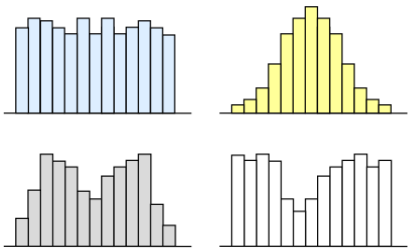
Measures of Shape

- To have a general idea of its shape, or distribution
- Helps identifying which descriptive statistic to use
- Symmetrical or nonsymmetrical
- Skewness.
- Kurtosis.



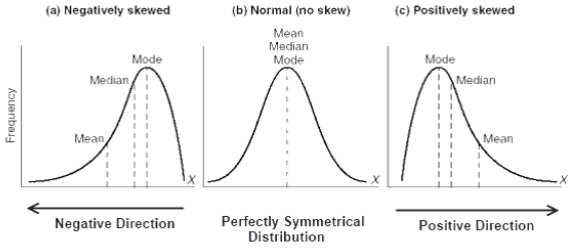
Symmetric

- Uniform.
- Normal.
- Camel-back.
- Bow-tie shaped.



Skewness

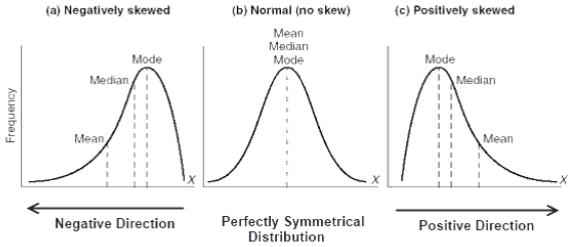
Measures the degree to which the values are symmetrically distributed about the center



If the distribution of values is skewed, then the median is a better indicator of the middle, compare to the mean.

Skewness

For perfectly symmetrical distribution, like Normal Distribution (middle figure):

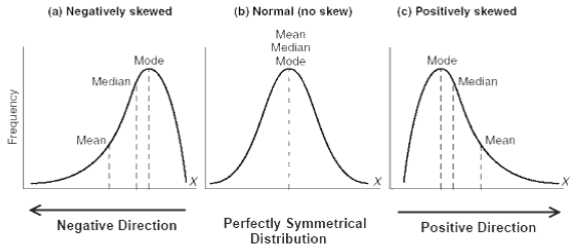


- Whats the mean?: the middle axis point
- Whats the mode?: Highest frequency, top most point
- Whats the median?: Half split of the curve is at the middle.

All Points/Axes are same.

Skewness

For skewed distribution (left and right figures):



- Whats the mean?: towards tail, as most of the heavy (+ve or -ve) points are there
- Whats the mode?: Highest frequency, top most point
- Whats the median?: somewhere between these two

All Points/Axes are different. Sides of Mean and Mode can decide right/left skewness.

Pearson's Skewness Coefficient

Karl Pearson coefficient of Skewness $sk_p = \frac{3(\mu - median)}{\sigma}$

- The direction of skewness is given by the sign.
- The coefficient compares the sample distribution with a normal distribution. The larger the value, the larger the difference.
- A value of zero means no skewness at all.
- A large negative value means the distribution is negatively skewed.
- A large positive value means the distribution is positively skewed.

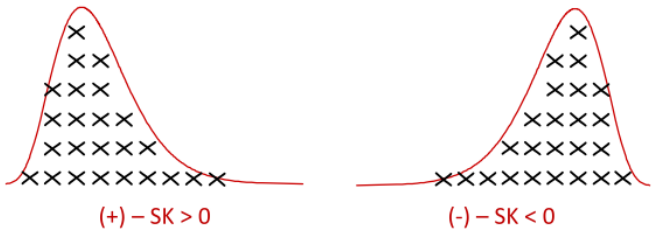
3rd Moment Skewness Coefficient

$sk_t = \frac{\sum(x_i - \mu)^3}{\sigma^3}$

- If the power would have been 1 (instead of 3) then $\sum(x_i - \mu)$ would have been 0. +ve and -ve will cancel each other.
- Odd moments are increased when there is a long tail to the right and decreased when there is a long tail to the left.

Skewness

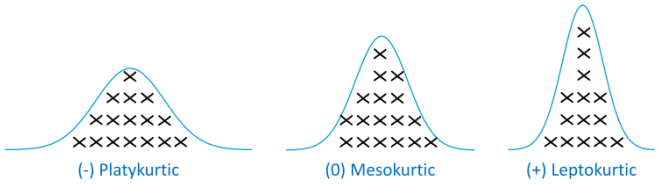
- Zero indicates perfect symmetry
- Negative value implies left-skewed data
- Positive value implies right-skewed data.



Measure of Skewness

Kurtosis

- Measures the degree of flatness (or peakness)
- Clustered around middle? More peak, more kurtosis value
- If values spread evenly, flatted, less kurtosis value



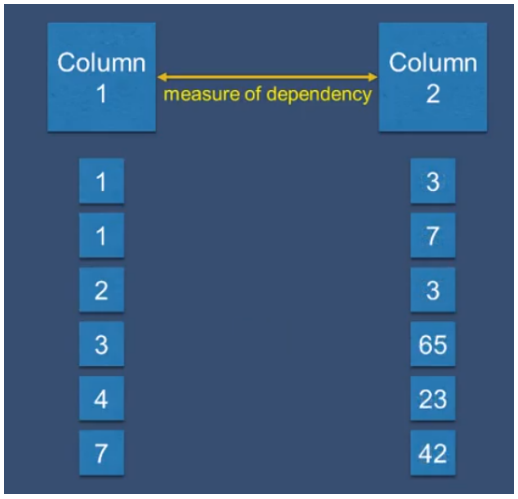
Kurtosis Skewness Coefficient

$sk_k = \frac{\sum(x_i - \mu)^4}{\sigma^4}$

- Since the exponent in the above is 4, the term in the summation will always be positive
- Moments of even order are increased when either tail is long.
- Kurtosis is a measure of outlier content. High if longer the tails so more the outliers.
- The third and fourth moments are the smallest examples of these so are used for skewness and kurtosis measures.

Bi-variate Analysis

Bi-variate Analysis



Correlations and Covariance

Covariance and Correlation

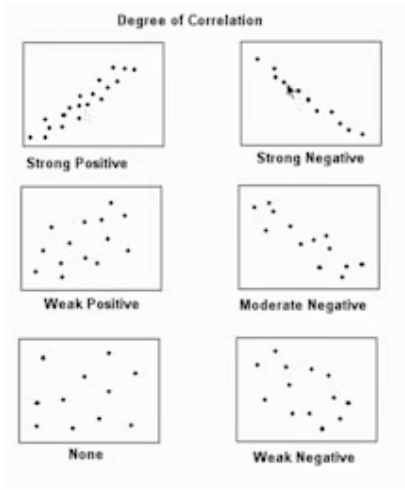
Both show association between two variables

- Positive: If one goes up, the other does too and vice versa.
- Example: Height and weight
- Not always, but tendency
- Another example: Temperature and Ice-creame sales
- Negative: Temperature and sale of woolen clothes

Correlation

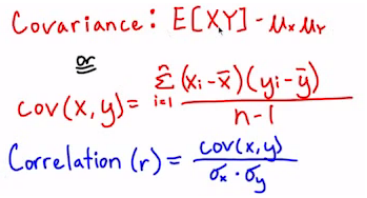
- Correlation is a value standardized between -1 to 1
- Relation between two variables is linear,
- Directly proportional in case of Positive Corr
- Inversely proportional in case of Negative Corr
- The value of corr is the factor of proportionality
- No correlation, ie no dependence so value = 0

Covariance and Correlation



Covariance

Implement covariance, the paired analogue of variance. The variance measures how a single variable deviates from its mean, covariance measures how two variables vary in tandem from their means.




```
x = [2, 3, 0, 1, 3]
y = [ 2, 1, 0, 1, 2]
result1 = covariance(x,y)
result2 = correlation(x,y)print("CoVariance {}, Correlation
    {}".format(result1,result2))
```

Covariance

Covariance is like a dot product and tell how two quantities (centered, meaning subtracted by Mean) are together/similar.

```
def elemwise_multi(v, w):
    """v_1 * w_1 + ... + v_n * w_n"""
    return sum(v_i * w_i for v_i, w_i in zip(v, w))

def covariance(x, y):
  n = len(x)
  return  elemwise_multi(de_mean(x), de_mean(y)) / (n - 1)


CoVariance 0.8
```

Correlation

Covariance is like a dot product normalized by standard deviation.

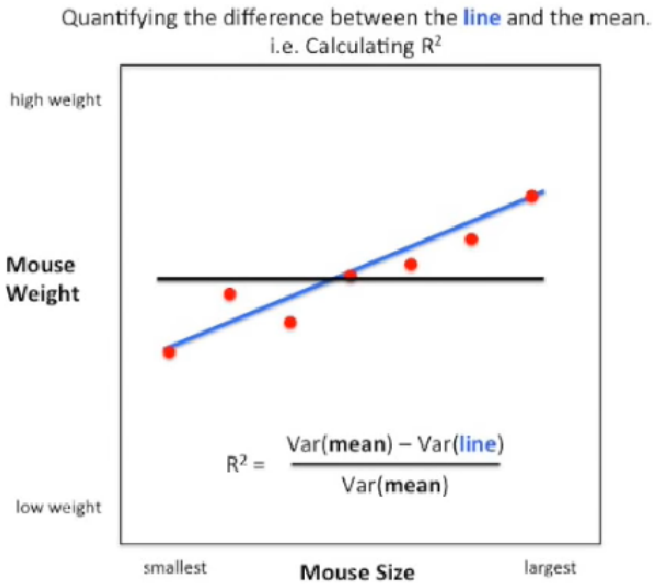
```
def correlation(x, y):
  stdev_x = std_dev(x)
    stdev_y = std_dev(y)
  if stdev_x > 0 and stdev_y > 0:
    return  covariance(x, y) / stdev_x / stdev_y
  else:
    return  0 # if  no variation, correlation is zero

Correlation 0.7333587976225691
```

R^2

- Correlation, the regular 'R' has values from -1 to 1 and is good enough to tell you that the two quantitative variables are strongly related.
- Why do you need R^2 then?
- Plain R is not easier to interpret.
- Example: $R = 0.7$ is twice as good as $R = 0.5$
- But its more clear when $R^2 = 0.7$ is 1.4 times as good as $R^2 = .5$

(Ref: StatQuest: R-squared explained - Josh Starmer)

R^2



- R^2 is used to decide the quality of the linear fitting.
- $Var(mean)$ represents the variation of just the mean line, ie black line.
- $Var(line)$ represents the variation calculated using he fitted line, ie blue line.
- Taking just relative ratio to make R^2 in range 0 t 1 and as a percentage.
- If the value is 0.81, it means there is 81% less variation around fitted line than the benchmark black line.
- So, if one variable is input (size) and one is output (weight), then we say that 81% of weight variation is explained by size.

(Ref: StatQuest: R-squared explained - Josh Starmer)

Copyleft ⓒ Send suggestions to yogeshkulkarni@yahoo.com