Mohamed Noordeen Alaudeen

Senior Data Scientist - Logitech

Statistics Summary

"Statistics is the grammar of science."

Karl Pearson

"If your experiment needs statistics, you ought to have done a better experiment…."

Ernest Rutherford

“Statistics is the study of the collection, analysis, interpretation, presentation and organization of data”.

(Dodge, Y. (2006) The Oxford Dictionary of Statistical Terms)

Importance of Statistics

Statistics has important role in determining

- Existing position of per capita income

Importance of Statistics

Statistics has important role in determining

- Existing position of per capita income
- Unemployment

Importance of Statistics

Statistics has important role in determining

- Existing position of per capita income
- Unemployment
- Population growth rate

Importance of Statistics

Statistics has important role in determining

- Existing position of per capita income
- Unemployment
- Population growth rate
- Housing

Importance of Statistics

Statistics has important role in determining

- Existing position of per capita income
- Unemployment
- Population growth rate
- Housing
- Schooling medical facilities
- Etc……..

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics
- Biology

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics
- Biology
- Botany

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics
- Biology
- Botany
- Psychology

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics
- Biology
- Botany
- Psychology
- Astronomy

Importance of Statistics

Statistics holds a central position in almost every field like

- Industry
- Commerce
- Trade
- Physics
- Chemistry
- Economics
- Mathematics
- Biology
- Botany
- Psychology
- Astronomy
- Information Technology etc..., so application of statistics is very wide.

Different fields of application of statistics

- **Astrostatistics**
- **Biostatistics**
- **Econometrics**
- **Business analytics** is a rapidly developing business process that applies statistical methods to data sets to develop new insights and understanding of business performance & opportunities
- **Environmental statistics Statistical mechanics**
- **Statistical physics**
- **Actuarial science** is the discipline that applies mathematical and statistical methods to assess risk in the insurance and finance industries

USE OF STATISTICS IN REAL LIFE

INDUSTRIES AND BUSINESS

- Performance Measurement
- Performance Boost
- Customer Data

AGRICULTURE

- What amount of crops are grown this year in comparison to previous year or in comparison to required amount of crop for the country
- Creating GDP form
- Quality and size of grains grown due to use of different fertilizer

USE OF STATISTICS IN REAL LIFE

FORESTRY

- How much growth has been occurred in area under forest or how much forest has been depleted in last 5 years?
- How much different species of flora and fauna have increased or decreased in last 5 years?

EDUCATION

- Money spend on girls education in comparison to boys education?
- Increase in no. of girl students who seated in who Seated for different exams?
- Comparison for result for last 10 years.

USE OF STATISTICS IN REAL LIFE

ECOLOGICAL STUDIES

- Comparison of increasing impact of pollution on global warming?
- Increasing effect of nuclear reactors on environment?

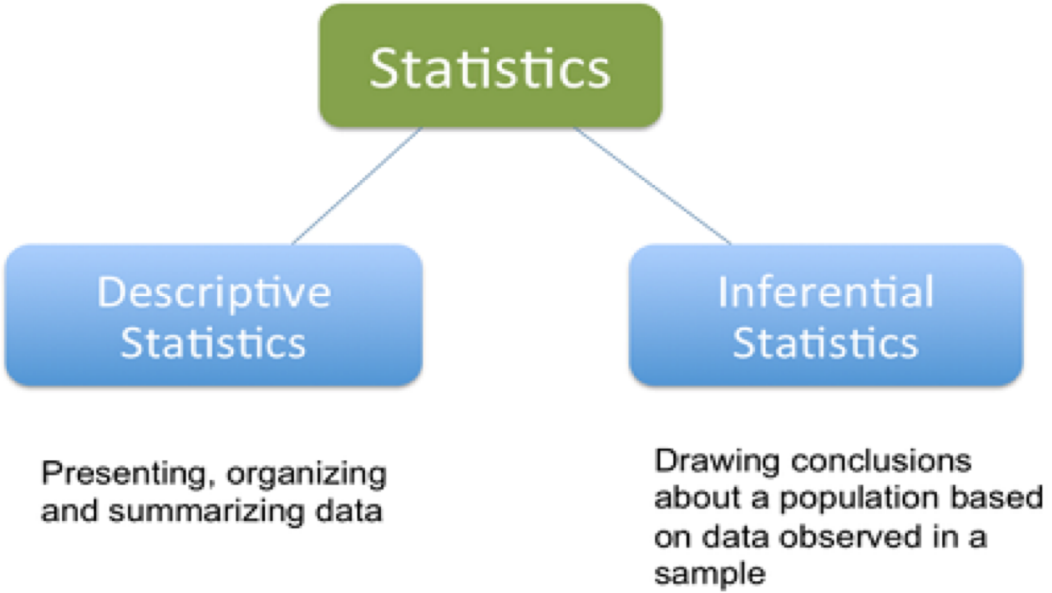
MEDICAL STUDIES

- No. of new diseases grown in last 10 year.
- Increase in no. of patients for a particular disease.

SPORTS

- Used to compare run rates of to different teams.
- Used to compare to different players.

What are the types of Statistics?



What are the types of Statistics?

Quick Reminder: **Inferential vs. Descriptive**

Inferential statistics generalize information from a random sample to a population.

Descriptive statistics do not generalize because they deal with the population itself.

Census and Survey

Census:

Gathering data from the whole population of interest. For example, elections, 10-year census, etc.

Survey:

Gathering data from the sample in order to make conclusions about the population.

For example, opinion polls, quality control checks in manufacturing units, etc

Parameter and Statistic

Parameter: A descriptive measure of the population.

For example, population mean, population variance, population standard deviation, etc.

Statistic: A descriptive measure of the sample.

For example, sample mean, sample variance, sample standard deviation, etc.

Identify Population Data or Sample Data?

- The US Government takes a census of its citizens every 10 years to gather information.

  a) **Population** b) Sample

- You want to know what sports teens prefer so you send out a survey to all the students in your high school.

  a) Population b) **Sample**

- You want data on the shoe size of all West students, so you interview every student at school.

  a) **Population** b) Sample

Identify as Parameter or Statistics?

- You want to know the mean income of the people who subscribe to People magazine, so you question 100 subscribers.

  a) Parameter b) **Statistic**

- You want to know the average height of the students in this math class, so you have everyone in the class write their height on a sheet of paper.

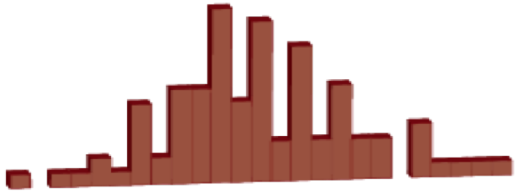
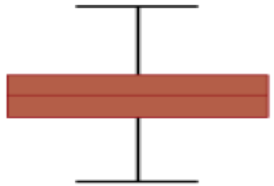
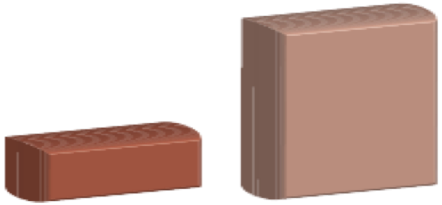
  a) **Parameter** b) Statistic

Parameter and Statistics

- Greek – Population Parameter
- Mean – μ
- Variance – σ^2
- Standard Deviation - σ

- Roman – Sample Statistic
- Mean – $\bar{x}$
- Variance – s^2
- Standard Deviation - s

Types of Data

- **Variable**
  - **Numerical**
    - **Continuous**
      - Ex. weight, systolic blood pressure
      - 
      - Descriptives: mean, median, SD, minimum, maximum, percentile
    - **Discrete**
      - Ex. number of visits
      - 
  - **Categorical**
    - **Nominal**
      - Ex. gender
      - 
      - Descriptives: absolute and relative frequencies, mode
    - **Ordinal**
      - Ex. educational level, NYHA classification
      - 
      - Descriptives: absolute and relative frequencies, median, interquartile range, minimum, maximum, percentile

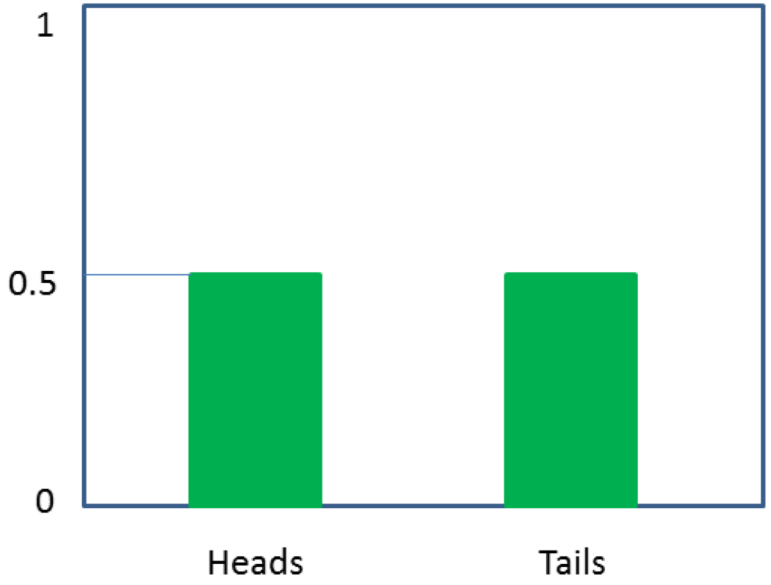
Numerical or Categorical?

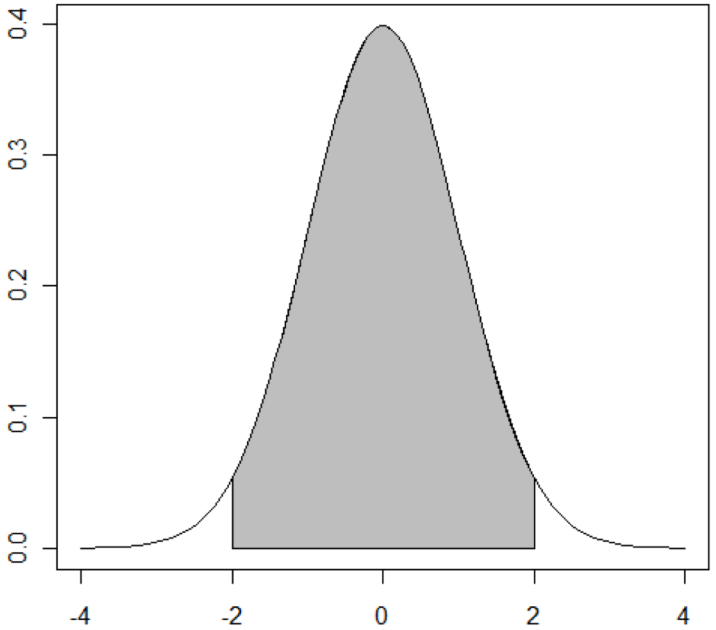
Age	Gender	Major	Units	Housing	GPA
18	Male	Psychology	16	Dorm	3.6
21	Male	Nursing	15	Parents	3.1
20	Female	Business	16	Apartment	2.8

- Numerical
  - Age
  - Units
  - GPA

- Categorical
  - Gender
  - Major
  - Housing

Discrete and Continuous





Discrete or Continuous?

- Time between customer arrivals at a retail outlet
  Continuous
- Sampling 100 voters in an exit poll and determining how many voted for the winning candidate

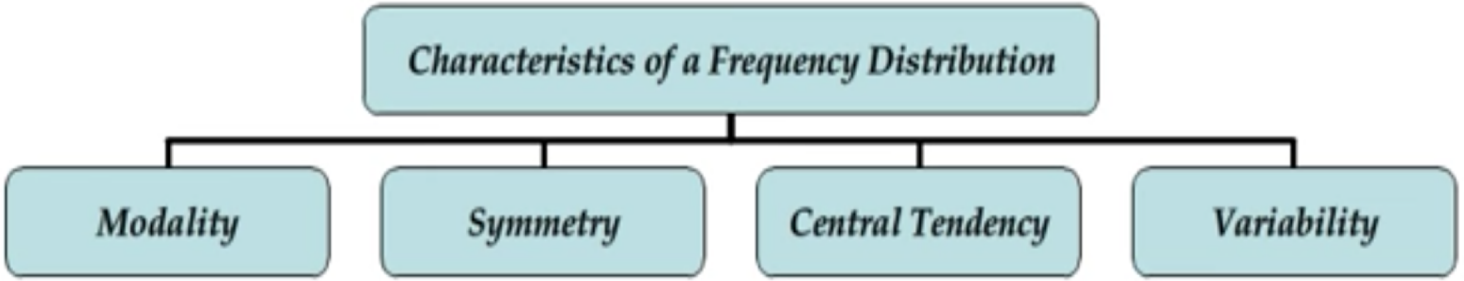
  Discrete
- Lengths of newly designed automobiles -

  Continuous
- No. of customers arriving at a retail outlet during a five- minute period

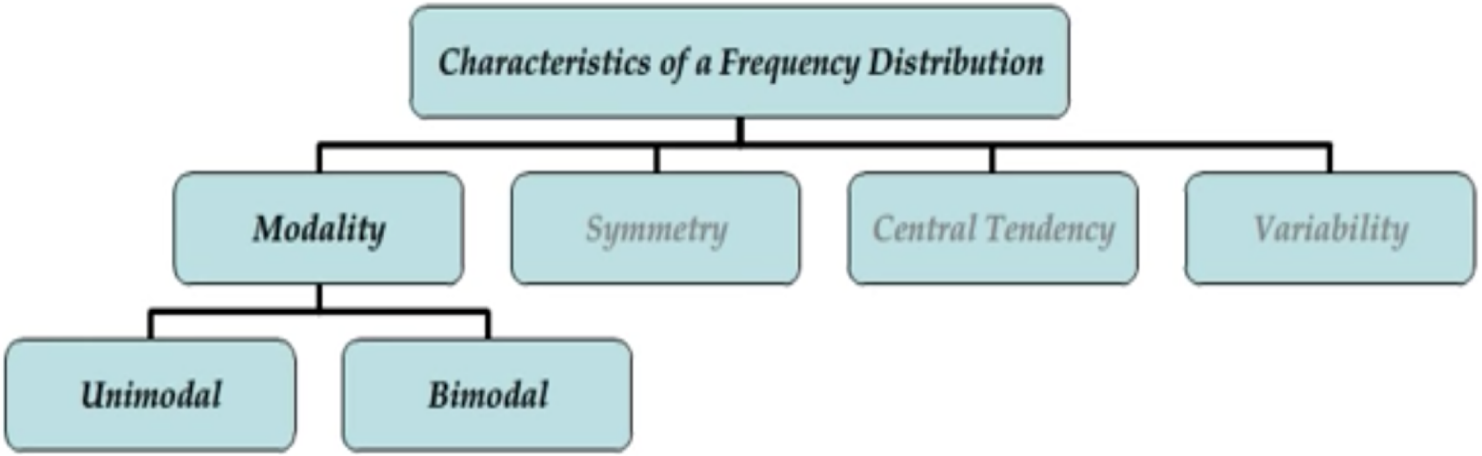
  Discrete
- No. of defects in a batch of 50 items

  Discrete

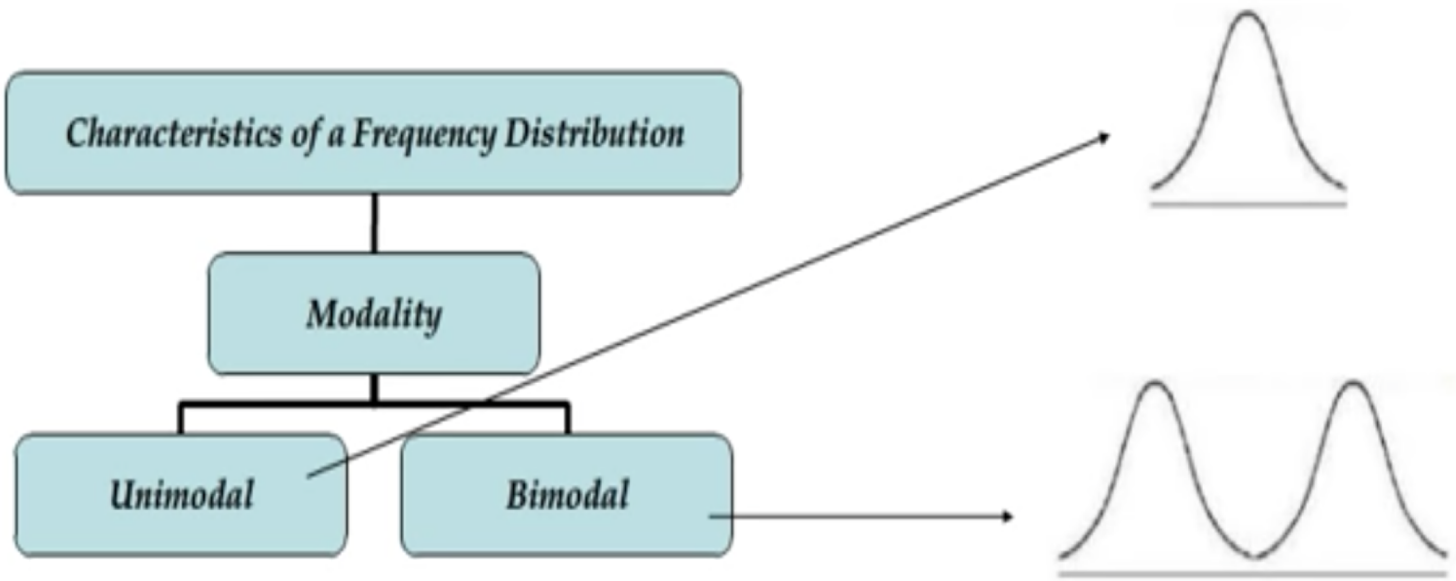
Summarizing the Data



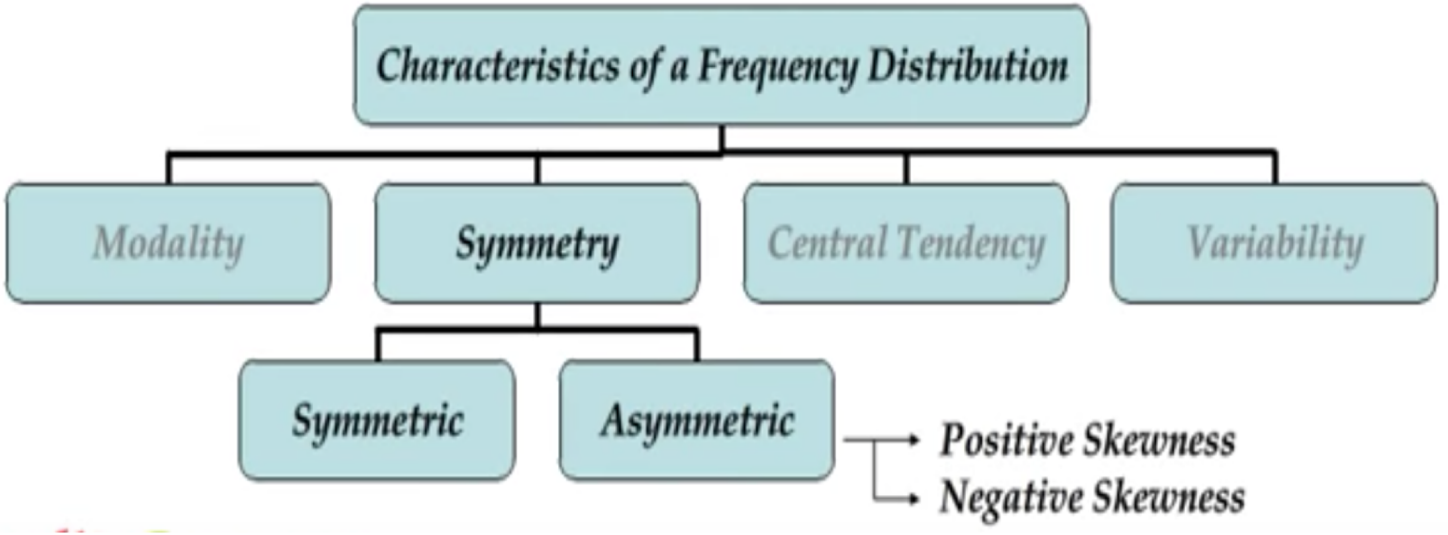
Summarizing the Data



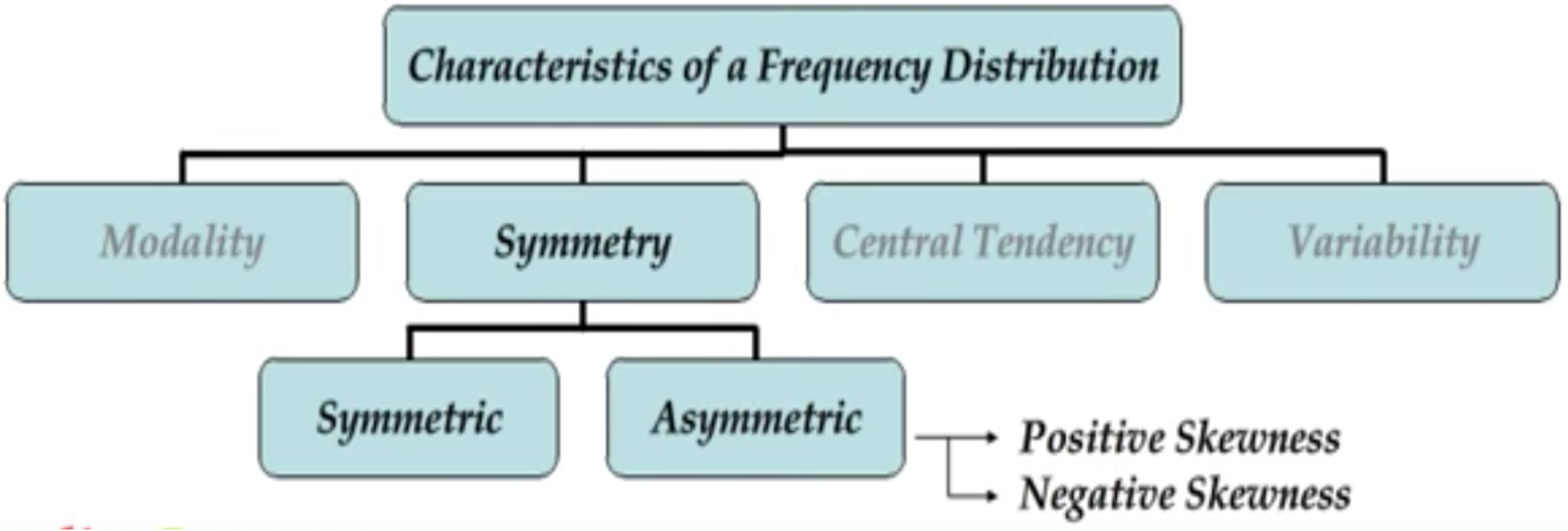
Summarizing the Data



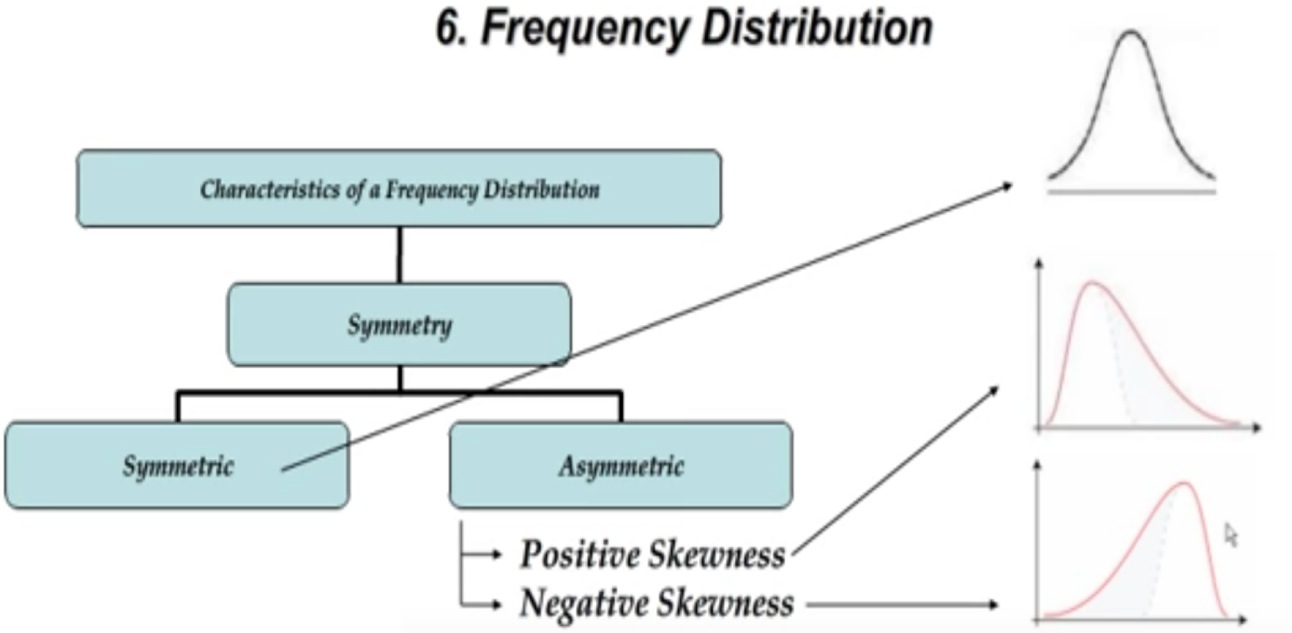
Summarizing the Data



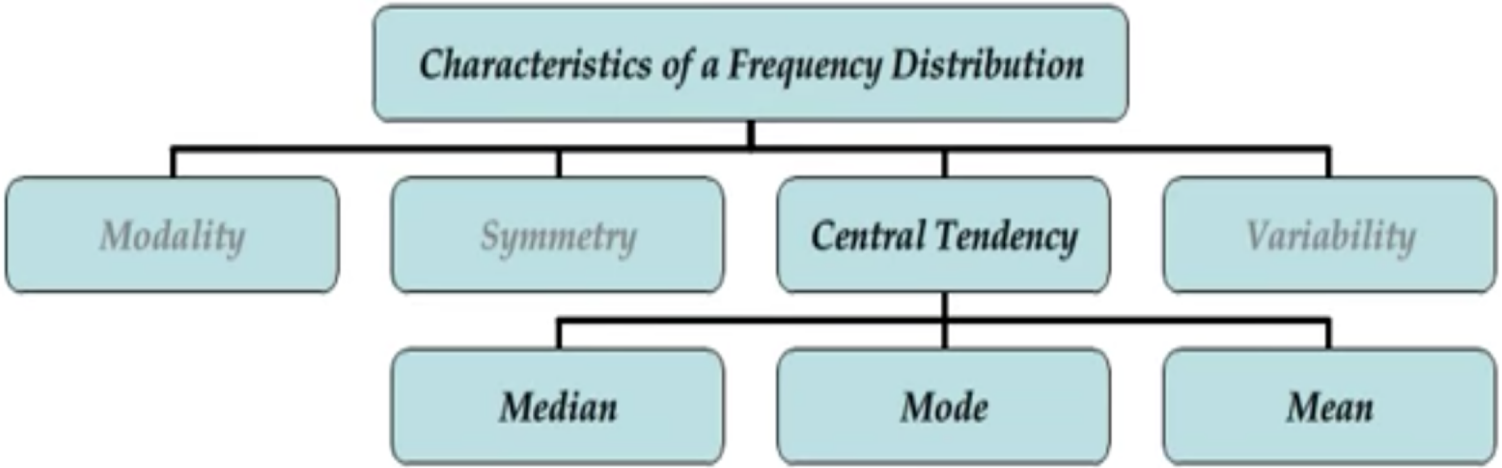
Summarizing the Data



Summarizing the Data



Summarizing the Data



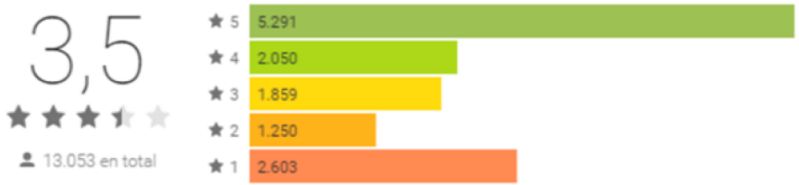
Applications of Mean

- 6ft long man was drowned while crossing swimming pool which was on an average 5ft deep!!

- Swimming pool was 7 ft deep at some places!!!

Applications of Mean

- When you try to search cool game app in play store you always look at rating first right? How that rating is calculated?



- In the above example If you add here all the rating given by users it will come to 45685 you divide it by 13053 you will get 3.5.

Problem of Mean

- **Mean sometime does not represent the data due to extreme values and that case we have to use Median.**

Application of Median

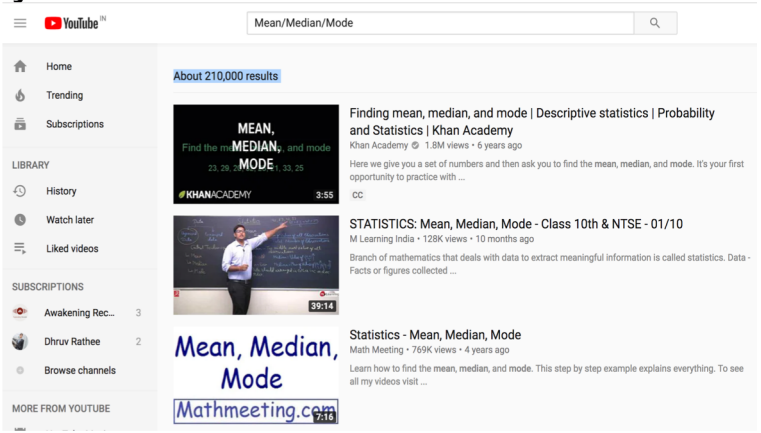
- Suppose in your class there are 11 students and one of them is son of Bill Gate.
- Total number of students = 11
- Total amount of pocket Money = 3300$
- So Mean pocket Money for each one of you is
- = Total Pocket Money/No of students
- = 3300/11
- = 300$

Students	Monthly Pocket Money in $
1	50
2	60
3	70
4	80
5	90
6	100
7	110
8	120
9	130
10	140
11	2350

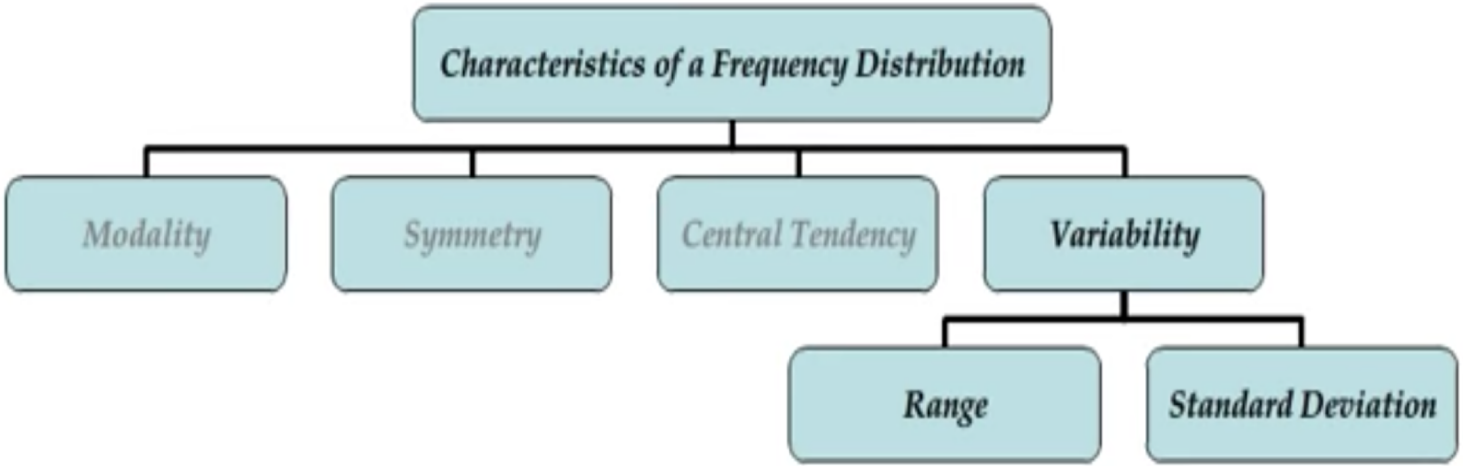
- Is any of the student getting pocket Money near 300$ and answer is No!! So should we use mean to represent the data here? No! Because of one extreme contribution by Son of Bill Gate.
- Median is positional average and its mid-point. So median value here 100$ and this value will represent maximum students.

Application of Mode

- Have you searched videos on YouTube? I am sure you must be doing it daily!!
- Let's say you want to search Mean/Median/Mode videos on YouTube.
- Now there are 210,000 results.
- Are you going to watch all the videos?
- What will you do if you just have to watch only one Video?
- Won't you try to watch the video will maximum views?
- Let me tell you if you do that then you just used "Mode" in real life.
- Remember the definition of Mode Its value in the series which has maximum frequency. Frequency here mean Views so out of 210,000 videos you will try to watch the videos which has maximum Views!



Summarizing the Data



Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37

Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37
SUM	351	351

Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37
SUM	351	351
MEAN	39	39

Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37
SUM	351	351
MEAN	39	39
MEDIAN	40	40

Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37
SUM	351	351
MEAN	39	39
MEDIAN	40	40
RANGE	120	23

Standard Deviation

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n}}$$

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

S.No	x	x^2	x-mean	Abs(x-mean)	(x-mean)^2
1	5	25	0.3333333333	0.3333333333	0.1111111111
2	7	49	2.333333333	2.333333333	5.444444444
3	4	16	-0.6666666667	0.6666666667	0.4444444444
4	2	4	-2.666666667	2.666666667	7.111111111
5	6	36	1.333333333	1.333333333	1.777777778
6	2	4	-2.666666667	2.666666667	7.111111111
7	8	64	3.333333333	3.333333333	11.11111111
8	5	25	0.3333333333	0.3333333333	0.1111111111
9	3	9	-1.666666667	1.666666667	2.777777778
SUM	42	232	0	15.33333333	36
Average	4.666666667	25.77777778			

Match	Player A	Player B
1	40	40
2	40	35
3	7	45
4	40	52
5	0	30
6	90	40
7	3	29
8	11	43
9	120	37
SUM	351	351
MEAN	39	39
MEDIAN	40	40
STANDARD DEVIATION	41.5180683558376	7.28010988928052

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Frequency, f	1	1	2	2	2	1	1

Points scored per game	**7**	**9**	**10**	**11**	**13**
Frequency, f	1	2	4	2	1

Points scored per game	**3**	**6**	**7**	**10**	**11**	**13**	**30**
Frequency, f	2	1	2	3	1	1	1

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Frequency, f	1	1	2	2	2	1	1

Points scored per game	**7**	**9**	**10**	**11**	**13**
Frequency, f	1	2	4	2	1

Points scored per game	**3**	**6**	**7**	**10**	**11**	**13**	**30**
Frequency, f	2	1	2	3	1	1	1

MEAN = ? MEDIAN = ? MODE = ?

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Frequency, f	1	1	2	2	2	1	1

Points scored per game	**7**	**9**	**10**	**11**	**13**
Frequency, f	1	2	4	2	1

Points scored per game	**3**	**6**	**7**	**10**	**11**	**13**	**30**
Frequency, f	2	1	2	3	1	1	1

MEAN = MEDIAN = MODE = 10

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	7	8	9	10	11	12	13
Frequency, f	1	1	2	2	2	1	1

Points scored per game	7	9	10	11	13
Frequency, f	1	2	4	2	1

Points scored per game	3	6	7	10	11	13	30
Frequency, f	2	1	2	3	1	1	1

MEAN = MEDIAN = MODE = 10 RANGE = 5 , 5 , 27

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	7	8	9	10	11	12	13
Frequency, f	1	1	2	2	2	1	1

Points scored per game	7	9	10	11	13
Frequency, f	1	2	4	2	1

Points scored per game	3	6	7	10	11	13	30
Frequency, f	2	1	2	3	1	1	1

MEAN = MEDIAN = MODE = 10 RANGE = 5 , 5 , 27 Reject Player 3

Basketball coach Statson is in a dilemma choosing between 3 players all having the same average scores.

Points scored per game	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Frequency, f	1	1	2	2	2	1	1

Points scored per game	**7**	**9**	**10**	**11**	**13**
Frequency, f	1	2	4	2	1

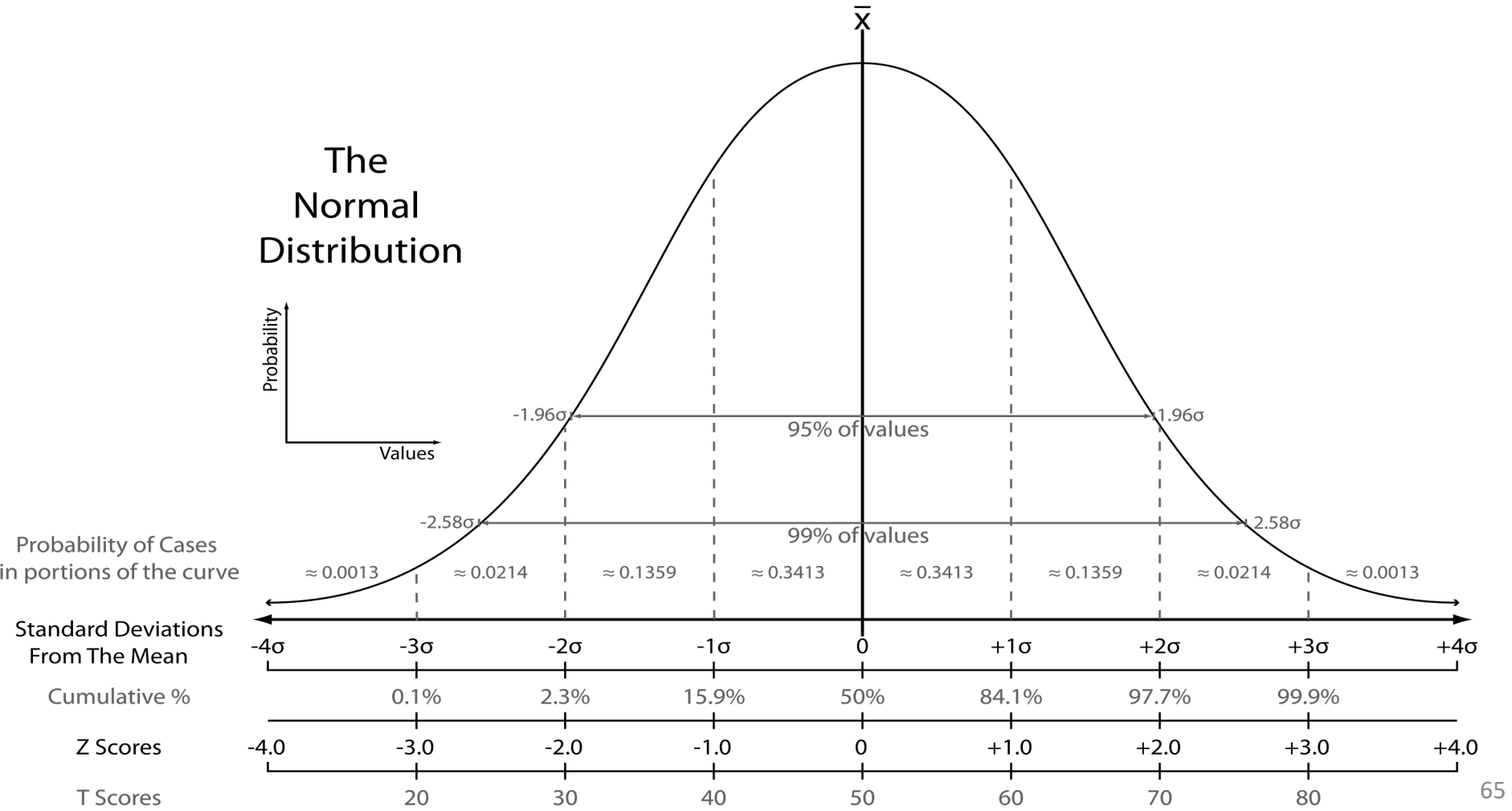
STANDARD DEVIATION

Player 1 = **1.7873008824606**
Player 2 = **3.30823887354653**

What is your Decision?????????

CONSISTENCY INDEX: TOP EIGHT RUN-SCORERS

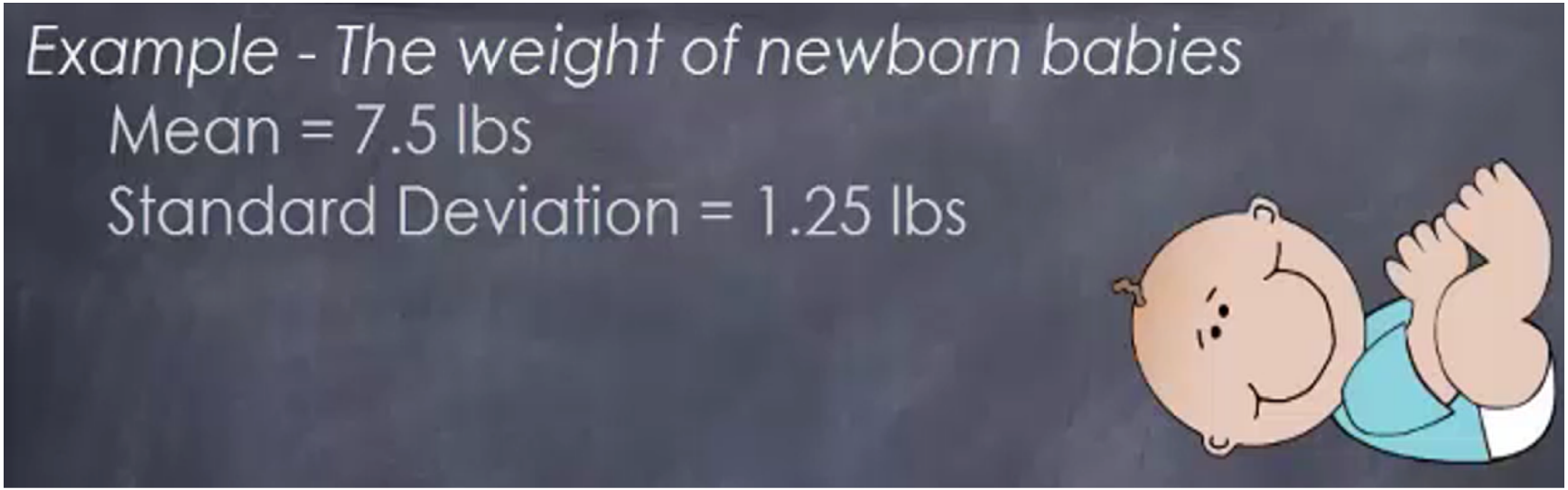
Player	For	Ci	Sd	Ave	M	I	No	Runs
Ricky Ponting	Aus	**1.05**	**59.47**	**56.88**	128	215	26	10750
Sunil Gavaskar	Ind	**1.06**	**54.42**	**51.12**	125	214	16	10122
Allan Border	Aus	**1.08**	**54.45**	**50.37**	156	265	44	11174
Rahul Dravid	Ind	**1.17**	**61.07**	**52.28**	131	227	26	10509
Sachin Tendulkar	Ind	**1.18**	**64.28**	**54.28**	156	256	27	12429
Jacques Kallis	Sa	**1.19**	**64.91**	**54.58**	128	216	33	9988
Brian Lara	Wi	**1.24**	**65.33**	**52.89**	131	232	6	11953
Steve Waugh	Aus	**1.26**	**64.16**	**51.06**	168	260	46	10927


The Normal Distribution
$\bar{X}$
Probability
Values
-1.96σ
1.96σ
95% of values
-2.58σ
2.58σ
99% of values
Probability of Cases in portions of the curve
≈ 0.0013
≈ 0.0214
≈ 0.1359
≈ 0.3413
≈ 0.3413
≈ 0.1359
≈ 0.0214
≈ 0.0013
Standard Deviations From The Mean
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
Cumulative %
0.1%
2.3%
15.9%
50%
84.1%
97.7%
99.9%
Z Scores
-4.0
-3.0
-2.0
-1.0
0
+1.0
+2.0
+3.0
+4.0
T Scores
20
30
40
50
60
70
80


EXECUTIVE OFFICES

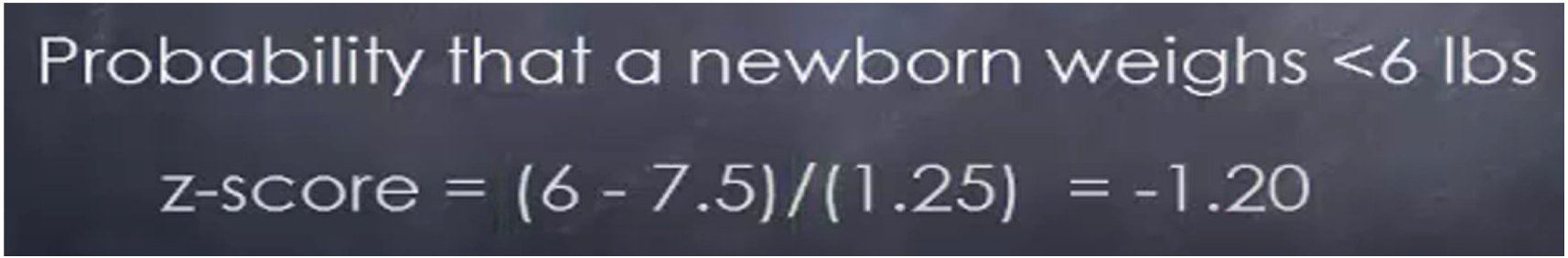
Application of Z-Score

1. Take a look at the weight of new-born babies. Suppose that the mean weight of new-borns is 7.5 pounds and the standard deviation is 1.25 pounds. Say you're interested in determining the probability that a new-born weighs less than 6 pounds. How do you do that?



Application of Z-Score

- The first thing you do is calculate the z-score. To figure out the z-score, take the difference between 6 and 7.5 to arrive at -1.5. When you divide -1.5 by the standard deviation of 1.25, you wind up with a z-score of -1.2.

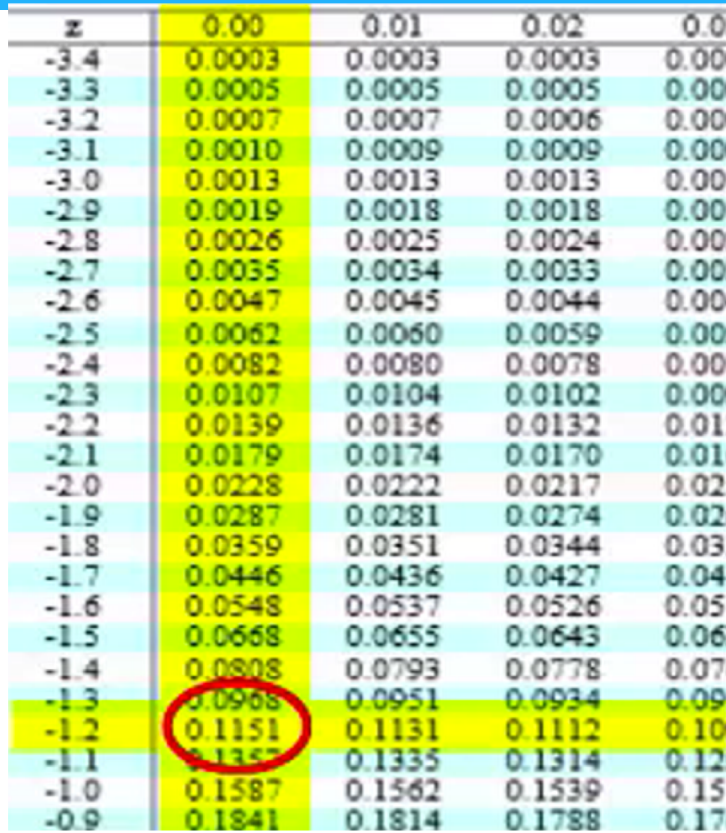


- When you're looking at a z-table, the first step is to determine which row you want to look at. That's going to be represented by the spot to the left of the decimal point and the first spot to the right of the decimal point, which is -1.2. The second step is to look at the digit that is two spots to the right of the decimal point, which is a 0. This determines the column you look at.

Application of Z-Score

- The point at which both the row and the column intersect one another indicates the probability.

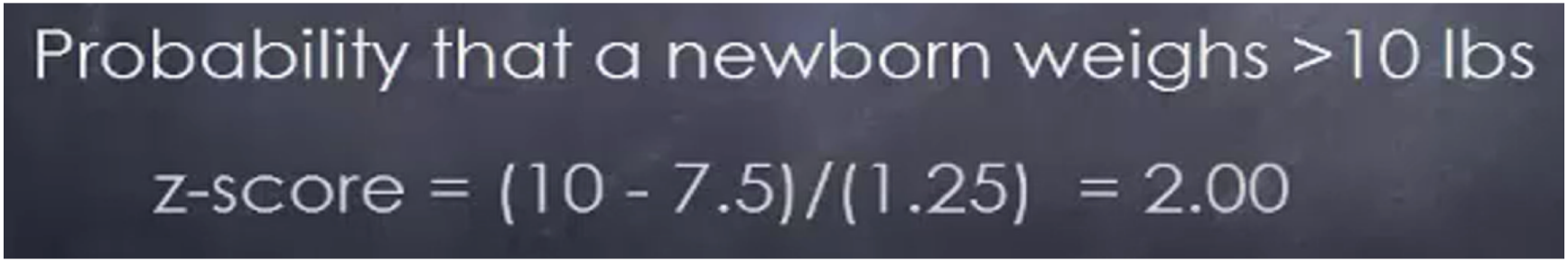
- The intersection in this scenario is 0.1151. That tells you that the probability associated with the baby weighing less than 6 pounds is 0.1151.



z	0.00	0.01	0.02	0.0
-3.4	0.0003	0.0003	0.0003	0.00
-3.3	0.0005	0.0005	0.0005	0.00
-3.2	0.0007	0.0007	0.0006	0.00
-3.1	0.0010	0.0009	0.0009	0.00
-3.0	0.0013	0.0013	0.0013	0.00
-2.9	0.0019	0.0018	0.0018	0.00
-2.8	0.0026	0.0025	0.0024	0.00
-2.7	0.0035	0.0034	0.0033	0.00
-2.6	0.0047	0.0045	0.0044	0.00
-2.5	0.0062	0.0060	0.0059	0.00
-2.4	0.0082	0.0080	0.0078	0.00
-2.3	0.0107	0.0104	0.0102	0.00
-2.2	0.0139	0.0136	0.0132	0.01
-2.1	0.0179	0.0174	0.0170	0.01
-2.0	0.0228	0.0222	0.0217	0.02
-1.9	0.0287	0.0281	0.0274	0.02
-1.8	0.0359	0.0351	0.0344	0.03
-1.7	0.0446	0.0436	0.0427	0.04
-1.6	0.0548	0.0537	0.0526	0.05
-1.5	0.0668	0.0655	0.0643	0.06
-1.4	0.0808	0.0793	0.0778	0.07
-1.3	0.0968	0.0951	0.0934	0.09
-1.2	0.1151	0.1131	0.1112	0.10
-1.1	0.1357	0.1335	0.1314	0.12
-1.0	0.1587	0.1562	0.1539	0.15
-0.9	0.1841	0.1814	0.1788	0.17

Application of Z-Score

2. What if you're interested in looking at the probability that a new-born might weight more than 10 pounds? You could calculate the z-score much the same way you did with the previous calculation:

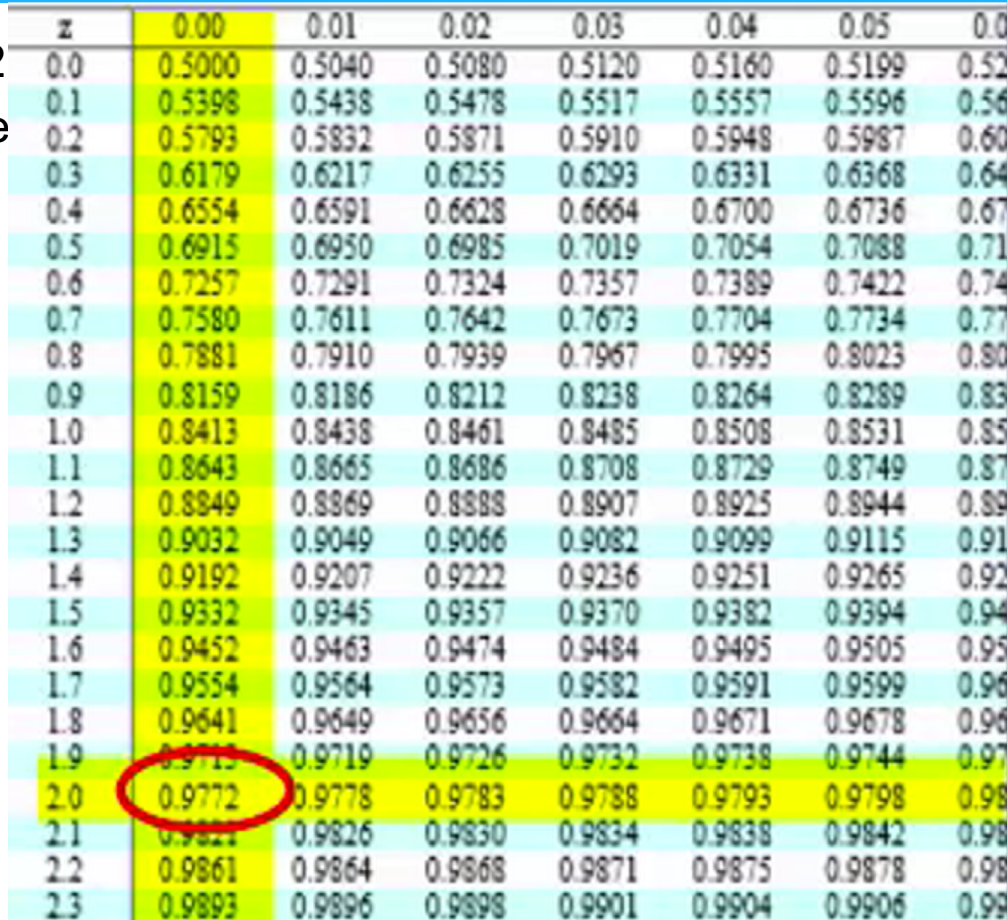


The same process is involved in looking at the z-table. Look at 2.0 and which row that entails. Since there are no numbers other than 0 to the right of the decimal point, you look at the very first column, which is the 0.00 column. **Note, this table looks different from the one above.** The z scores in the left column are negative instead of positive. For a full PDF on z-score tables,

Application of Z-Score

- You wind up with a probability of 0.9772

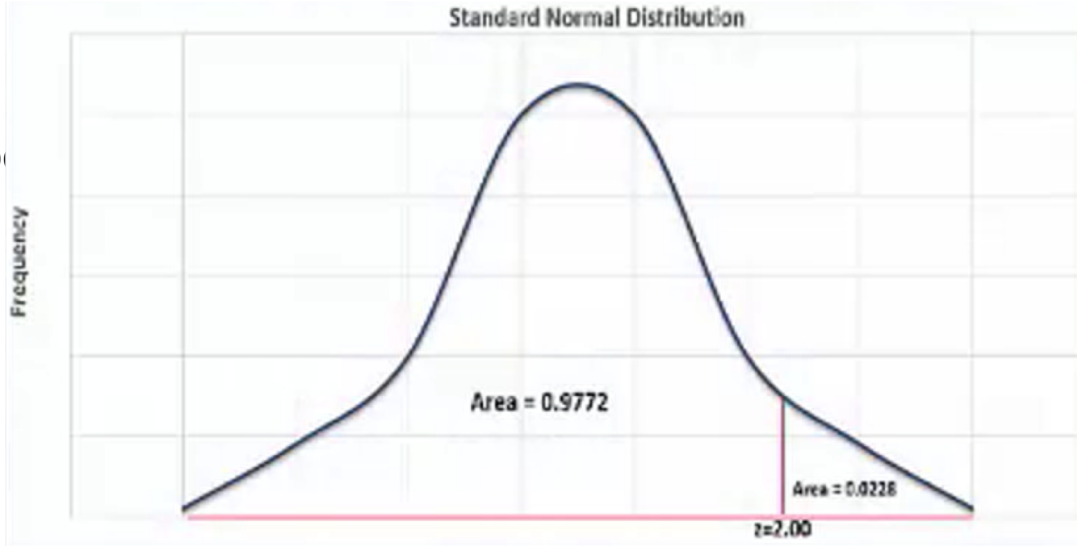
Remember, you are trying to determine the probability that a new-born weighs more than 10 pounds. The value given to you here is telling you the area underneath the curve to the left of +2 standard deviations, as you see on the graph right here:



z	0.00	0.01	0.02	0.03	0.04	0.05	0.0
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.52
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.56
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.60
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.64
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.67
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.71
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.74
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.77
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.80
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.83
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.85
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.87
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.89
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.91
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.92
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.94
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.95
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.96
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.96
1.9	[illegible]	0.9719	0.9726	0.9732	0.9738	0.9744	0.97
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.98
2.1	[illegible]	0.9826	0.9830	0.9834	0.9838	0.9842	0.98
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.98
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.99

Application of Z-Score

- You will need to subtract that value from 1 to get the area that's on the right-hand side of that line. The probability that a new-born would b greater than 10 pounds would be 1 minus 0.9772. The probability is equal to 0.0228.



Application of Z-Test

- **According to a recent survey, the daily one-way commuting distance of U.S. workers averages 13 miles with a standard deviation of 13 miles. An investigator wishes to determine whether the national average describes the mean commuting distance for all workers in the Chicago area. Commuting distances are obtained for a random sample of 169 workers form this area, and the mean distance is found to be 15.5 miles. Test the null hypothesis at the 0.05 level of significance.**

Application of Z-Test

- **Review of the assumptions for z test:**

1. The sample size is large enough (>25) to satisfy the requirement of the central limit theorem.
2. Population standard deviation is known as 13 miles.
3. Scale of measurement of variable "commuting distance" is interval-ratio.

Application of Z-Test

- **Research Question:** Does the national average commuting distance describe the mean commuting distance for all workers in the Chicago area?
- Population of interest: All workers in the Chicago area
- Sample: Randomly selected 169 workers from Chicago area

- **Statistical hypothesis:**
- Null hypothesis (H_0): $\mu = 13$
- Alternative hypothesis (H_1): $\mu \neq 13$

  Where: μ is mean commuting distance for all Chicago workers.

Application of Z-Test

- **Decision Rule:** H_0 should be rejected if observed z equals to or is more positive than the upper critical z (1.96) or if observed z equals to or is more negative than the lower critical z (-1.96) at level of significance (α) of 0.05.

- **Calculations of test statistics (Z):**

- **Given**:
- x (sample mean) = 15.5;
- μ_0 (hypothetical population mean) = 13;
- n (sample size) = 169;
- σ (population standard deviation) = 13

$$z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}} = \frac{15.5 - 13}{13 / \sqrt{169}} = 2.5 > 1.96$$

Application of Z-Test

- **Decision:** Reject H_0

- **Interpretation:** The national average commuting distance does not describe the mean commuting distance for all workers in the Chicago area

Application of t-test

- **Your company wants to improve sales. Past sales data indicate that the average sale was $100 per transaction. After training your sales force, recent sales data (taken from a sample of 25 salesmen) indicates an average sale of $130, with a standard deviation of $15. Did the training work? Test your hypothesis at a 5% alpha level.**

- Step 1: Write your null hypothesis statement (How to state a null hypothesis).
- The accepted hypothesis is that there is no difference in sales, so:
  H_0: μ = $100.

Application of t-test

- Step 2: Write your alternate hypothesis. This is the one you're testing. You think that there *is* a difference (that the mean sales increased), so:
  H_1: $\mu > \$100$.

- Step 3: Identify the following pieces of information you'll need to calculate the test statistic. The question should give you these items:

- **Given:**
- **The sample mean**($\bar{x}$). This is given in the question as $130.
- **The population mean**(μ). Given as $100 (from past data).
- **The sample standard deviation**(s) = $15.
- **Number of observations**(n) = 25.

Application of t-test

- Step 4: Insert the items from above into the t score formula.

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

- t = (130 – 100) / ((15 / √(25))
  t = (30 / 3) = 10
  This is your **calculated t-value**.

Application of t-test

- Step 5: Find the t-table value. You need two values to find this:
- The alpha level: given as 5% in the question.
- The degrees of freedom, which is the number of items in the sample (n) minus 1: 25 – 1 = 24.
- Look up 24 degrees of freedom in the left column and 0.05 in the top row. The intersection is 1.711.This is your one-tailed critical t-value.
- What this critical value means is that we would expect most values to fall under 1.711. If our calculated t-value (from Step 4) falls within this range, the null hypothesis is likely true.
- Step 6: Compare Step 4 to Step 5. The value from Step 4 **does not** fall into the range calculated in Step 5, so we can reject the null hypothesis. The value of 10 falls into the rejection region (the left tail).
- In other words, it's highly likely that the mean sale is greater. The sales training was probably a success.

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Application of Chi Square Test

- **The chi-square formula is a difficult formula to deal with. That's mostly because you're expected to add a large amount of numbers. The easiest way to solve the formula is by making a table.**
  Sample question: 256 visual artists were surveyed to find out their zodiac sign. The results were: Aries (29), Taurus (24), Gemini (22), Cancer (19), Leo (21), Virgo (18), Libra (19), Scorpio (20), Sagittarius (23), Capricorn (18), Aquarius (20), Pisces (23). Test the hypothesis that zodiac signs are evenly distributed across visual artists.
-

Application of Chi Square Test

- Step 1: **Make a table** with columns for “Categories,” “Observed,” “Expected,” “Residual (Obs-Exp)”, “$(Obs-Exp)^2$” and “Component $(Obs-Exp)^2$ / Exp.” Don’t worry what these mean right now; We’ll cover that in the following steps.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Application of Chi Square Test

- Step 2: **Fill in your categories**. Categories should be given to you in the question. There are 12 zodiac signs, so:

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries					
Taurus					
Gemini					
Cancer					
Leo					
Virgo					
Libra					
Scorpio					
Sagittarius					
Capricorn					
Aquarius					
Pisces					

Application of Chi Square Test

- Step 3: **Write your counts**. Counts are the number of each items in each category in column 2. You're given the counts in the question:

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29				
Taurus	24				
Gemini	22				
Cancer	19				
Leo	21				
Virgo	18				
Libra	19				
Scorpio	20				
Sagittarius	23				
Capricorn	18				
Aquarius	20				
Pisces	23				

Application of Chi Square Test

- Step 4: **Calculate your expected value** for column 3. In this question, we would expect the 12 zodiac signs to be evenly distributed for all 256 people, so 256/12=21.333. Write this in column 3.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29	21.333			
Taurus	24	21.333			
Gemini	22	21.333			
Cancer	19	21.333			
Leo	21	21.333			
Virgo	18	21.333			
Libra	19	21.333			
Scorpio	20	21.333			
Sagittarius	23	21.333			
Capricorn	18	21.333			
Aquarius	20	21.333			
Pisces	23	21.333			

Application of Chi Square Test

- Step 5: **Subtract the expected value (Step 4) from the Observed value (Step 3)** and place the result in the “Residual” column. For example, the first row is Aries: 29-21.333=7.667.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29	21.333	7.667		
Taurus	24	21.333	2.667		
Gemini	22	21.333	0.667		
Cancer	19	21.333	-2.333		
Leo	21	21.333	-0.333		
Virgo	18	21.333	-3.333		
Libra	19	21.333	-2.333		
Scorpio	20	21.333	-1.333		
Sagittarius	23	21.333	1.667		
Capricorn	18	21.333	-3.333		
Aquarius	20	21.333	-1.333		
Pisces	23	21.333	1.667		

Application of Chi Square Test

- Step 6: **Square your results from Step 5** and place the amounts in the $(Obs-Exp)^2$ column.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29	21.333	7.667	58.782889	
Taurus	24	21.333	2.667	7.112889	
Gemini	22	21.333	0.667	0.44889	
Cancer	19	21.333	-2.333	5.442889	
Leo	21	21.333	-0.333	0.110889	
Virgo	18	21.333	-3.333	11.108889	
Libra	19	21.333	-2.333	5.442889	
Scorpio	20	21.333	-1.333	1.776889	
Sagittarius	23	21.333	1.667	2.778889	
Capricorn	18	21.333	-3.333	11.108889	
Aquarius	20	21.333	-1.333	1.776889	
Pisces	23	21.333	1.667	2.778889	

Application of Chi Square Test

- Step 7: **Divide the amounts in Step 6 by the expected value (Step 4)** and place those results in the final column.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29	21.333	7.667	58.782889	2.755490976
Taurus	24	21.333	2.667	7.112889	0.333421882
Gemini	22	21.333	0.667	0.44889	0.021042048
Cancer	19	21.333	-2.333	5.442889	0.255139408
Leo	21	21.333	-0.333	0.110889	0.005198003
Virgo	18	21.333	-3.333	11.108889	0.520737308
Libra	19	21.333	-2.333	5.442889	0.255139408
Scorpio	20	21.333	-1.333	1.776889	0.083292973
Sagittarius	23	21.333	1.667	2.778889	0.130262457
Capricorn	18	21.333	-3.333	11.108889	0.520737308
Aquarius	20	21.333	-1.333	1.776889	0.083292973
Pisces	23	21.333	1.667	2.778889	0.130262457

Application of Chi Square Test

- Step 8: **Add up (sum) all the values in the last column**.

Category	Observed	Expected	Residual= (Obs-Exp)	(Obs-Exp)^2	Component = (Obs-Exp)^2 / Exp
Aries	29	21.333	7.667	58.782889	2.755490976
Taurus	24	21.333	2.667	7.112889	0.333421882
Gemini	22	21.333	0.667	0.44889	0.021042048
Cancer	19	21.333	-2.333	5.442889	0.255139408
Leo	21	21.333	-0.333	0.110889	0.005198003
Virgo	18	21.333	-3.333	11.108889	0.520737308
Libra	19	21.333	-2.333	5.442889	0.255139408
Scorpio	20	21.333	-1.333	1.776889	0.083292973
Sagittarius	23	21.333	1.667	2.778889	0.130262457
Capricorn	18	21.333	-3.333	11.108889	0.520737308
Aquarius	20	21.333	-1.333	1.776889	0.083292973
Pisces	23	21.333	1.667	2.778889	0.130262457
					5.094017203

- *This is the chi-square statistic:5.094.*

Application of Chi Square Test

Chi-Square (χ^2) Distribution

Degrees of Freedom	Area to the Right of Critical Value							
	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01
1	—	0.001	0.004	0.016	2.706	3.841	5.024	6.635
2	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210
3	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345
4	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277
5	0.554	0.831	1.145	1.610	9.236	11.071	12.833	15.086
6	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812
7	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475
8	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090
9	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666
10	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209
11	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725
12	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217
13	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688
14	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141
15	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578
16	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000
17	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409
18	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805
19	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191
20	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566
21	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932
22	9.542	10.982	12.338	14.042	30.813	33.924	36.781	40.289
23	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638
24	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980
25	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314
26	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642
27	12.879	14.573	16.151	18.114	36.741	40.113	43.194	46.963
28	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278
29	14.257	16.047	17.708	19.768	39.087	42.557	45.722	49.588

Application of Chi Square Test

- Step 9: Look up 11 degrees of freedom in the column and 0.05 in the top row. The intersection is 19.675.This is your chi-square critical t-value.
- Compare Step 9 to Step 8. The value from Step 8 **does** fall into the range calculated in Step 9, so we can accept the null hypothesis.
- In other words **zodiac signs are evenly distributed**

Summary of Statistical Tests

Statistic Test	Type of Data Needed	Test Statistic	Example
Correlation	Two continuous variables	Pearson's r	Are blood pressure and weight correlated?
T-tests/ANOVA	Means from a continuous variable taken from two or more groups	Student's *t*	Do normal weight (group 1) patients have lower blood pressure than obese patients (group 2)?
Chi-square	Two categorical variables	Chi-square X^2	Are obese individuals (obese vs. not obese) significantly more likely to have a stroke (stroke vs. no stroke)?

Comparison of MEANS	Degrees of Freedom	Application	Assumptions	Test Statistic
One Sample Z-Test	Not Applicable	Testing the difference of a sample mean, x-bar, with a known population mean, μ (fixed mean, historical mean, or targeted mean)	Normal distribution Known population σ.	$Z = \frac{\overline{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$
One Sample t-test	n-1	Testing the difference of one sample mean, x-bar, with a known population mean, μ (fixed mean, historical mean, or targeted mean)	Normal distribution Population standard deviation, σ, is unknown.	$t = \frac{\overline{x} - \mu}{\frac{s}{\sqrt{n}}}$
Two Sample t-test	$n_1 + n_2 - 2$	Testing difference of two sample means when population variances unknown but considered equal	Normal Distribution Requires standard pooled deviation calculation, s_p	$t = \frac{\overline{x}_1 - \overline{x}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$
Paired t-test	n - 1	Testing two sample means when their respective population standard deviations are unknown but considered equal. Data recorded in pairs and each pair has a difference, d.	Normal Distribution Two dependent samples Always two-tailed test S_d = standard deviation of the differences of all samples	$t = \frac{\overline{d}\sqrt{n}}{s_d}$
One-Way ANOVA	$n_1 - 1$ & $n_2 - 1$	Testing the difference of three or more population means	Normal Distribution s_1^2 and s_2^2 represent sample variances	$F = \frac{(s_1)^2}{(s_2)^2}$

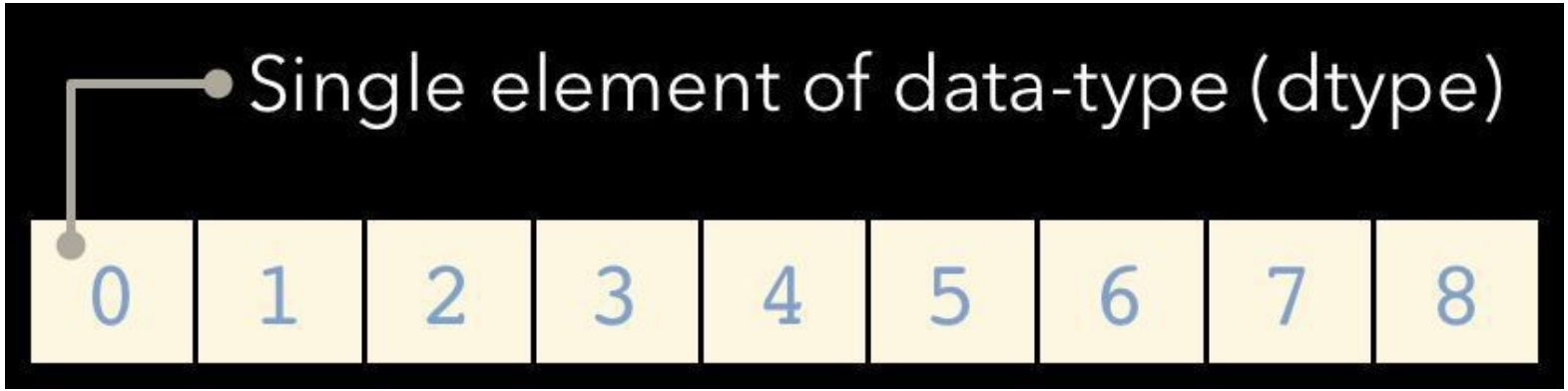
Statistics with Python

- https://github.com/nursnaaz/stats-testing-in-python

Introduction to NumPy

What is NumPy

- NumPy is a Python C extension library for array-oriented computing Efficient
- In-memory
- Contiguous (or Strided)
- Homogeneous (but types can be algebraic)



- NumPy is suited to many applications
- Image processing
- Signal processing, Linear algebra

Fast Computation of NumPy array ¶

In [8]:
```
import numpy as np
import time
import sys

# let's declare the size
Size = 100000

# Creating two lists
list1 = range(Size)
list2 = range(Size)

# Creating two NumPy arrays
arr1 = np.arange(Size)
arr2 = np.arange(Size)

# Calculating time for Python list
start = time.time()
result = [(x+y) for x, y in zip(list1, list2)]

print("Time for Python List in msec: ", (time.time() - start) * 1000)

# Calculating time for NumPy array
start = time.time()
result = arr1+arr2
print("Time for NumPy array in msec: ", (time.time()- start) * 1000)

print("This means NumPy array is faster than Python List")
```

```
Time for Python List in msec:  16.850709915161133
Time for NumPy array in msec:  1.6350746154785156
This means NumPy array is faster than Python List
```

Comparing Memory use

In [1]:
```
import numpy as np
import time
import sys

# Creating a NumPy array with 100 elements
array = np.arange(100)
# array.itemsize : Size of one element
# array.size : length of array
print("Size of NumPy array: ", array.size * array.itemsize)

# Creating a list with 100 elements
# Now I'll print the size of list
list = range(0, 100)
# Multiplying size of 1 element with length of the list
print("Size of list: ", sys.getsizeof(1)*len(list))
```

```
Size of NumPy array:  800
Size of list:  2800
```

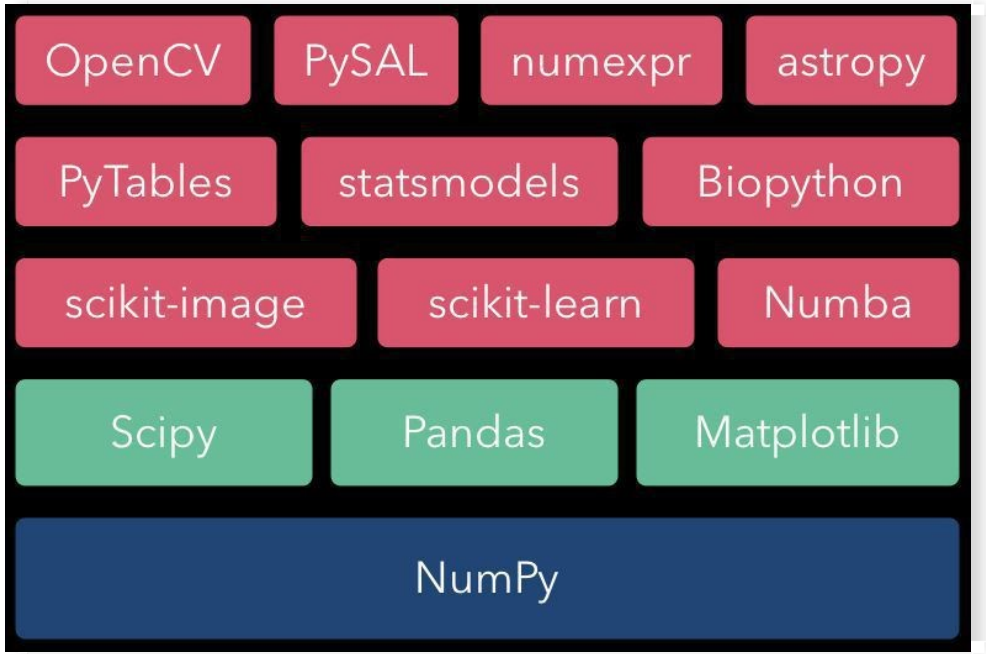
NumPy is Convenient to use

In [11]:
```
import numpy as np
import time

a1 = np.array([1, 2, 3])
a2 = np.array([4, 5, 6])

# To add two array you can simply do it by
print("ADD a1 and a2 elements : ", a1+a2)

# To sub two array you can simply do it by
print("SUB a1 and a2 elements : ", a1 - a2)

# To mul two array you can simply do it by
print("MUL a1 and a2 elements : ", a1 * a2)

# Calculating time for NumPy array
start = time.time()
result = arr1+arr2
print("Time for NumPy array in msec: ", (time.time()- start) * 1000)
```

```
ADD a1 and a2 elements :  [5 7 9]
SUB a1 and a2 elements :  [-3 -3 -3]
MUL a1 and a2 elements :  [ 4 10 18]
Time for NumPy array in msec:  0.5030632019042969
```

NumPy is the foundation of the Python scientific stack

NumPy Ecosystem

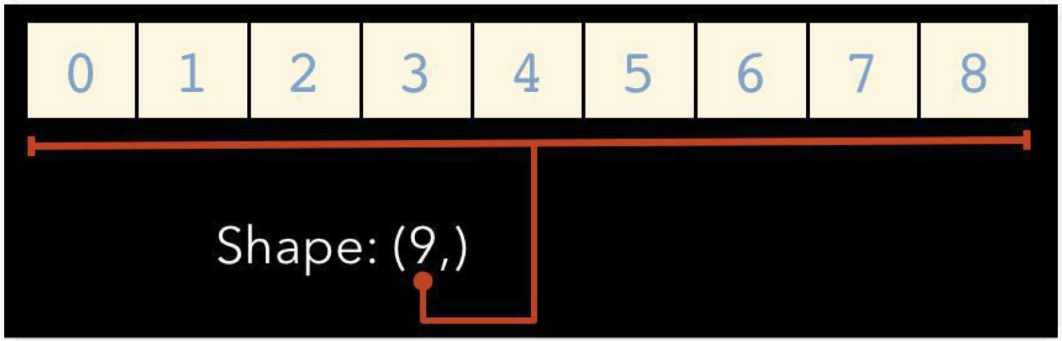


Quick Start

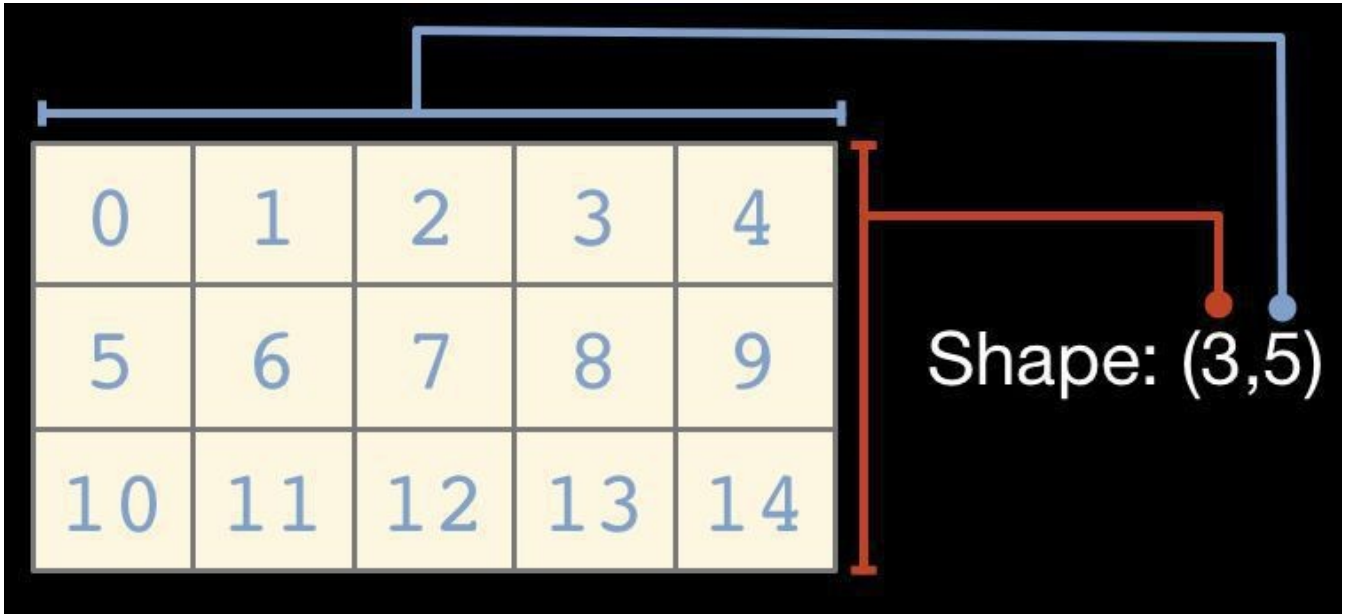
```
In [1]: import numpy as np

In [2]: a = np.array([1,2,3,4,5,6,7,8,9])

In [3]: a
Out[3]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])

In [4]: b = a.reshape((3,3))

In [5]: b
Out[5]:
array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9]])

In [6]: b * 10 + 4
Out[6]:
array([[14, 24, 34],
[44, 54, 64],
[74, 84, 94]])
```

Array Shape

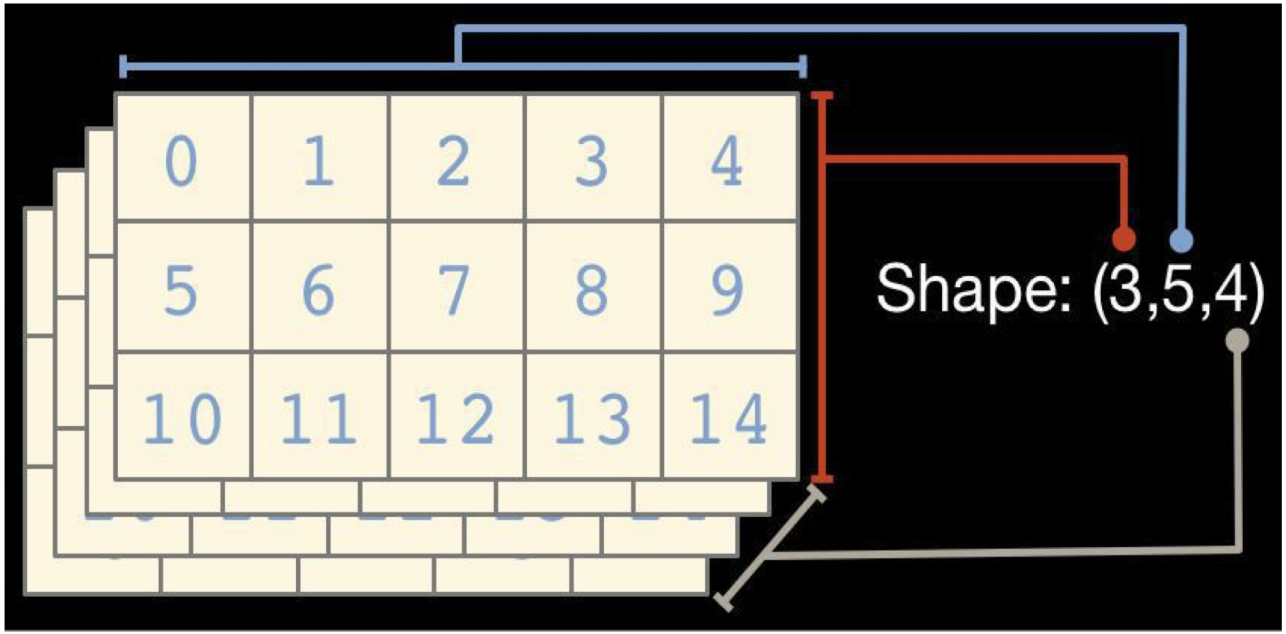
- One dimensional arrays have a 1-tuple for their shape



...Two dimensional arrays have a 2-tuple



...And so on



Array Element Type(dtype)

- NumPy arrays comprise elements of a single data type The type object is accessible through the `.dtype` attribute

- Here are a few of the most important attributes of dtype objects
  `dtype.byteorder` — big or little endian `dtype.itemsize` — element size of this dtype
  - `dtype.name` — a name for this dtype object
  - `dtype.type` — type object used to create scalars
  - There are many others...

Array dtypes are usually inferred automatically

```
In [16]: a = np.array([1,2,3])

In [17]: a.dtype
Out[17]: dtype('int64')

In [18]: b = np.array([1,2,3,4.567])

In [19]: b.dtype
Out[19]: dtype('float64')
```

- But can also be specified explicitly

```
In [20]: a = np.array([1,2,3], dtype=np.float32)

In [21]: a.dtype
Out[21]: dtype('int64')

In [22]: a
Out[22]: array([ 1.,  2.,  3.], dtype=float32)
```

Array Creation

- Explicitly from a list of values

```
In [2]: np.array([1,2,3,4])
Out[2]: array([1, 2, 3, 4])
```

- As a range of values

```
In [3]: np.arange(10)
Out[3]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

- By specifying the number of elements

```
In [4]: np.linspace(0, 1, 5)
Out[4]: array([ 0.  ,  0.25,  0.5 ,  0.75,  1.  ])
```

Array Creation

- **Zero-initialized**

```
In [4]: np.zeros((2,2))
Out[4]:
array([[ 0.,  0.],
[ 0.,  0.]])
```

- **One-initialized**

```
In [5]: np.ones((1,5))
Out[5]: array([[ 1.,  1.,  1.,  1.,  1.]])
```

- **Uninitialized**

```
In [4]: np.empty((1,3))
Out[4]: array([[  2.12716633e-314,   2.12716633e-314,   2.15203762e-314]])
```

Array Creation

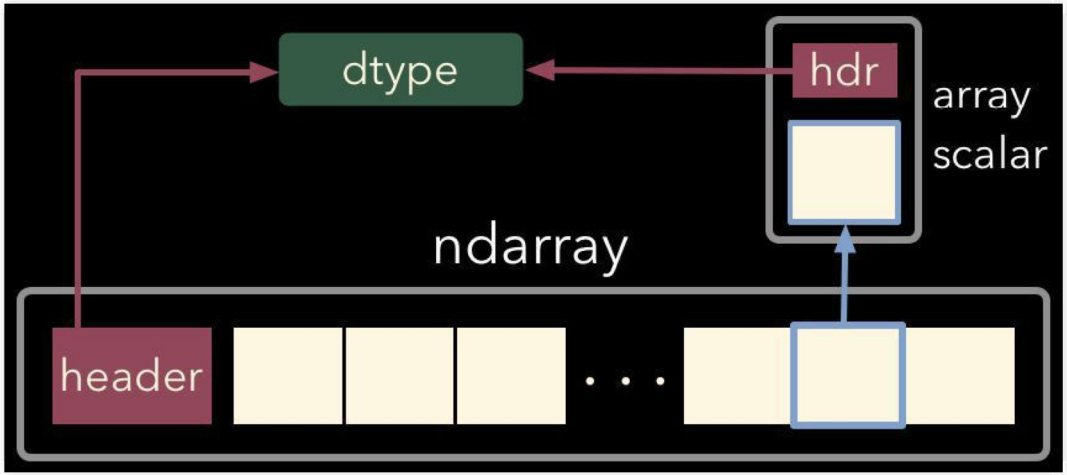
- Constant diagonal value

```
In [6]: np.eye(3)
Out[6]:
array([[ 1.,  0.,  0.],
[ 0.,  1.,  0.],
[ 0.,  0.,  1.]]
            )
```

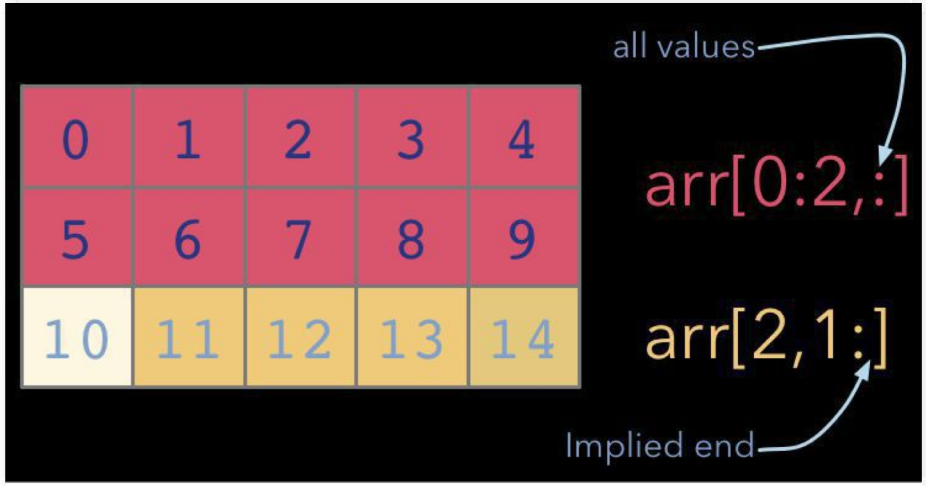
- Multiple diagonal values

```
In [7]: np.diag([1,2,3,4])
Out[7]:
array([[1, 0, 0, 0],
[0, 2, 0, 0],
[0, 0, 3, 0],
[0, 0, 0, 4]])
```

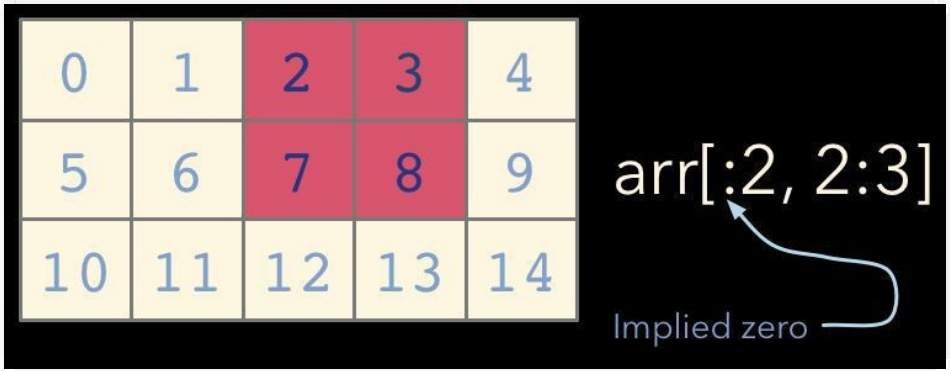
Array Memory Layout



Indexing and Slicing

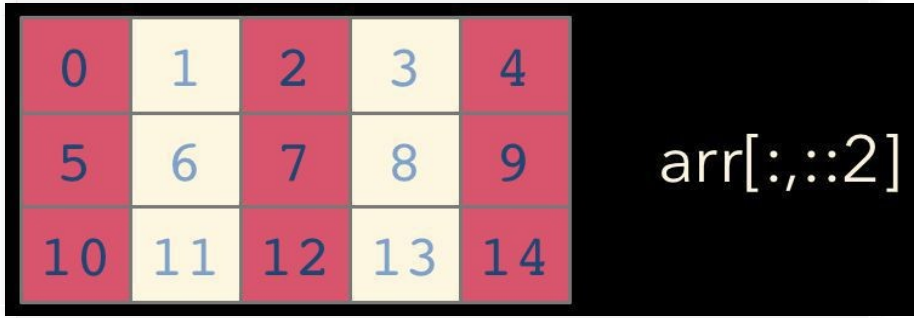


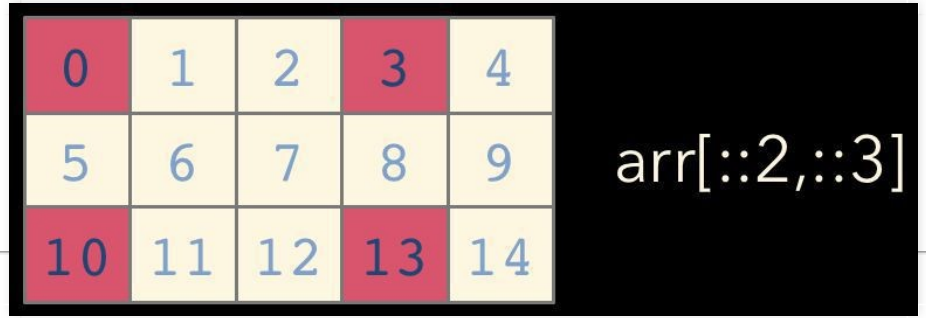
Indexing and Slicing



Indexing and Slicing

- NumPy array can also take an optional stride





Array Views

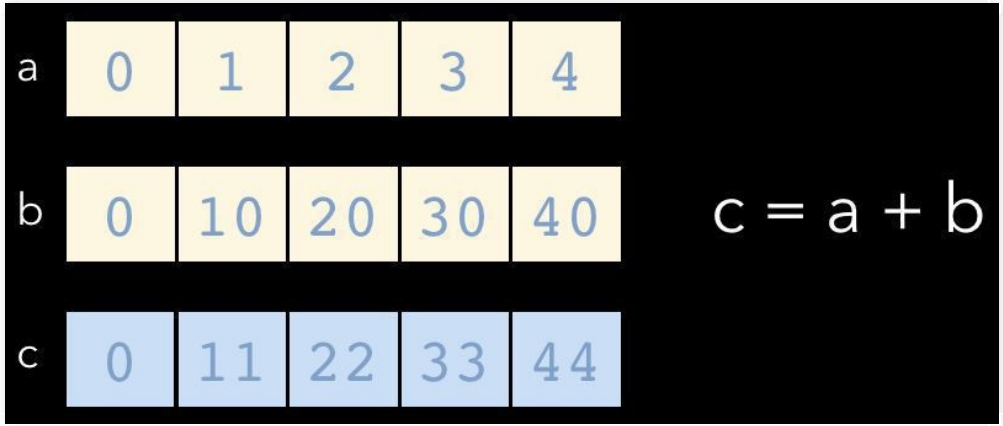
- Simple assignments do not make copies of arrays (same semantics as Python). Slicing operations do not make copies either; they return views on the original array.

```
In [2]: a = np.arange(10)

In [3]: b = a[3:7]

In [4]: b
Out[4]: array([3, 4, 5, 6])

In [5]: b[:] = 0

In [6]: a
Out[6]: array([0, 1, 3, 0, 0, 0, 0, 7, 8, 9])

In [7]: b.flags.owndata
Out[7]: False
```

- Array views contain a pointer to the original data, but may have different shape or stride values. Views always have `flags.owndata` equal to `False`.

Universal Functions (ufuncs)

- NumPy ufuncs are functions that operate element-wise on one or more arrays



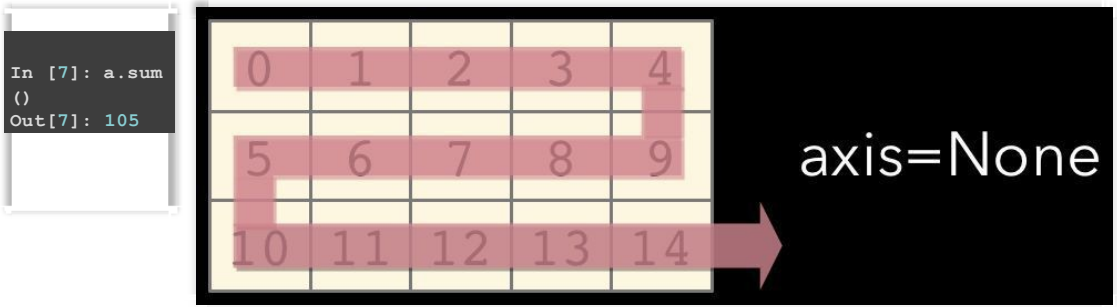
- **ufuncs dispatch to optimized C inner-loops based on array dtype**

NumPy has many built-in ufuncs

- comparison: `<, <=, ==, !=, >=, >`
- arithmetic: `+, -, *, /, reciprocal, square`
- exponential: `exp, expm1, exp2, log, log10, log1p, log2, power, sqrt`
- trigonometric: `sin, cos, tan, acsin, arccos, atctan`
- hyperbolic: `sinh, cosh, tanh, acsinh, arccosh, atctanh`
- bitwise operations: `&, |, ~, ^, left_shift, right_shift`
- logical operations: `and, logical_xor, not, or`
- predicates: `isfinite, isinf, isnan, signbit`
- other: `abs, ceil, floor, mod, modf, round, trunc`

Axis

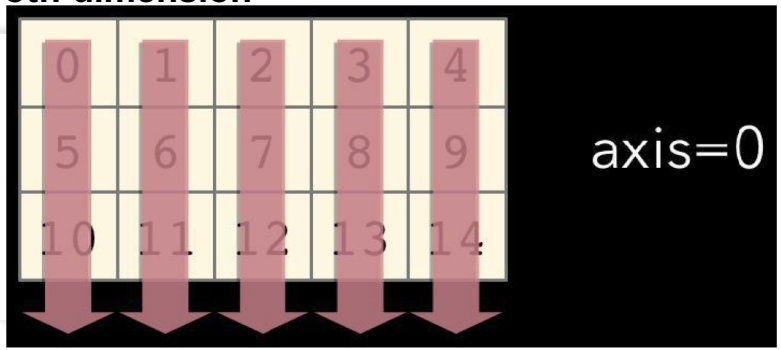
- Array method reductions take an optional `axis` parameter that specifies over which axes to reduce `axis=None` reduces into a single scalar



- `axis=None` is the default

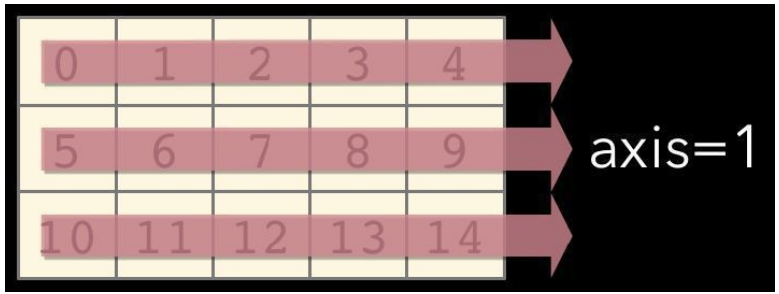
`axis=0` reduces into the zeroth dimension

```
In [8]: a.sum(axis=0)
Out[8]: array([15, 18, 21, 24,
27])
```



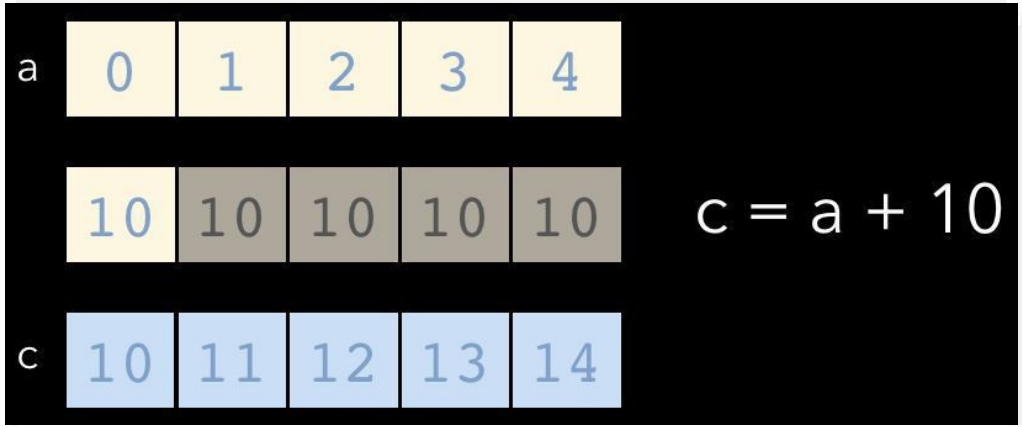
`axis=1` reduces into the first dimension

```
In [9]: a.sum(axis= 1)
Out[9]: array([10,35,60])
```



Broadcasting

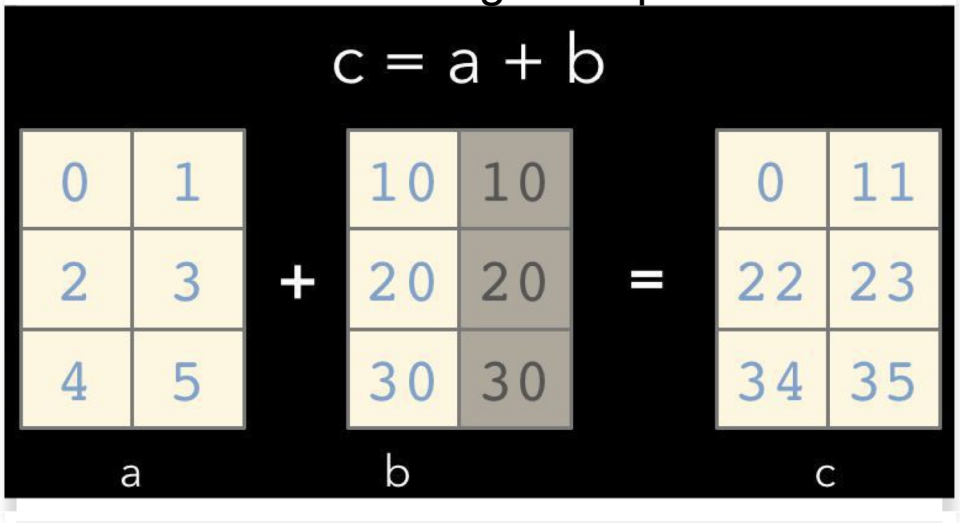
- A key feature of NumPy is broadcasting, where arrays with different, but compatible shapes can be used as arguments to ufuncs



- **In this case an array scalar is broadcast to an array with shape `(5, )`**

Broadcasting

- A slightly more involved broadcasting example in two dimensions



- **Here an array of shape `(3, 1)` is broadcast to an array with shape `(3, 2)`**

Broadcasting Rules

- In order for an operation to broadcast, the size of all the trailing dimensions for both arrays must either:
- be equal OR be one

```
A       (1d array):                 3
B       (2d array):             2 x 3
Result  (2d array):             2 x 3
A       (2d array):             6 x 1
B       (3d array):         1 x 6 x 4
Result  (3d array):         1 x 6 x 4
A       (4d array):     3 x 1 x 6 x 1
B       (3d array):         2 x 1 x 4
Result  (4d array):     3 x 2 x 6 x 4
```

Array Methods

- Predicates
- a.any(), a.all()
- Reductions
- a.mean(), a.argmin(), a.argmax(), a.trace(), a.cumsum(), a.cumprod()
- Manipulation
- a.argsort(), a.transpose(), a.reshape(...), a.ravel(), a.fill(...), a.clip(...)
- Complex Numbers
- a.real, a.imag, a.conj()

Fancy Indexing

- NumPy arrays may be used to index into other arrays

```
In [2]: a = np.arange(15).reshape((3,5))

In [3]: a
Out[3]:
array([[ 0,  1,  2,  3,  4],
[ 5,  6,  7,  8,  9],
[10, 11, 12, 13, 14]])
In [4]: i = np.array([[0,1], [1, 2]])

In [5]: j = np.array([[2, 1], [4, 4]])

In [6]: a[i,j]
Out[6]:
array([[ 2,  6],
[ 9, 14]])
```

Boolean arrays can also be used as indices into other arrays

```
In [2]: a = np.arange(15).reshape((3,5))

In [3]: a
Out[3]:
array([[ 0,  1,  2,  3,  4],
[ 5,  6,  7,  8,  9],
[10, 11, 12, 13, 14]])

In [4]: b = (a % 3 == 0)

In [5]: b
Out[5]:
array([[ True, False, False,  True, False],
[False,  True, False, False,  True],
[False, False,  True, False, False]], dtype=bool)

In [6]: a[b]
Out[6]: array([ 0,  3,  6,  9, 12])
```

NumPy Functions

- **Data I/O**
- fromfile, genfromtxt, load, loadtxt, save, savetxt
- **Mesh Creation**
- mgrid, meshgrid, ogrid
- **Manipulation**
- einsum, hstack, take, vstack

Array Subclasses

- numpy.ma — Masked arrays
- numpy.matrix — Matrix operators
- numpy.memmap — Memory-mapped arrays
- numpy.recarray — Record arrays

Other Subpackages

- **numpy.fft** — Fast Fourier transforms
- **numpy.polynomial** — Efficient polynomials
- **numpy.linalg** — Linear algebra,cholesky, det, eig, eigvals, inv, lstsq, norm, qr, svd
- **numpy.math** — C standard library math functions
- **numpy.random** — Random number generation
  beta, gamma, geometric, hypergeometric, lognormal, normal, poisson, uniform, weibull

Resources

http://docs.scipy.org/doc/numpy/reference/

http://docs.scipy.org/doc/numpy/user/index.html

http://www.scipy.org/Tentative_NumPy_Tutorial

http://www.scipy.org/Numpy_Example_List

THANK YOU