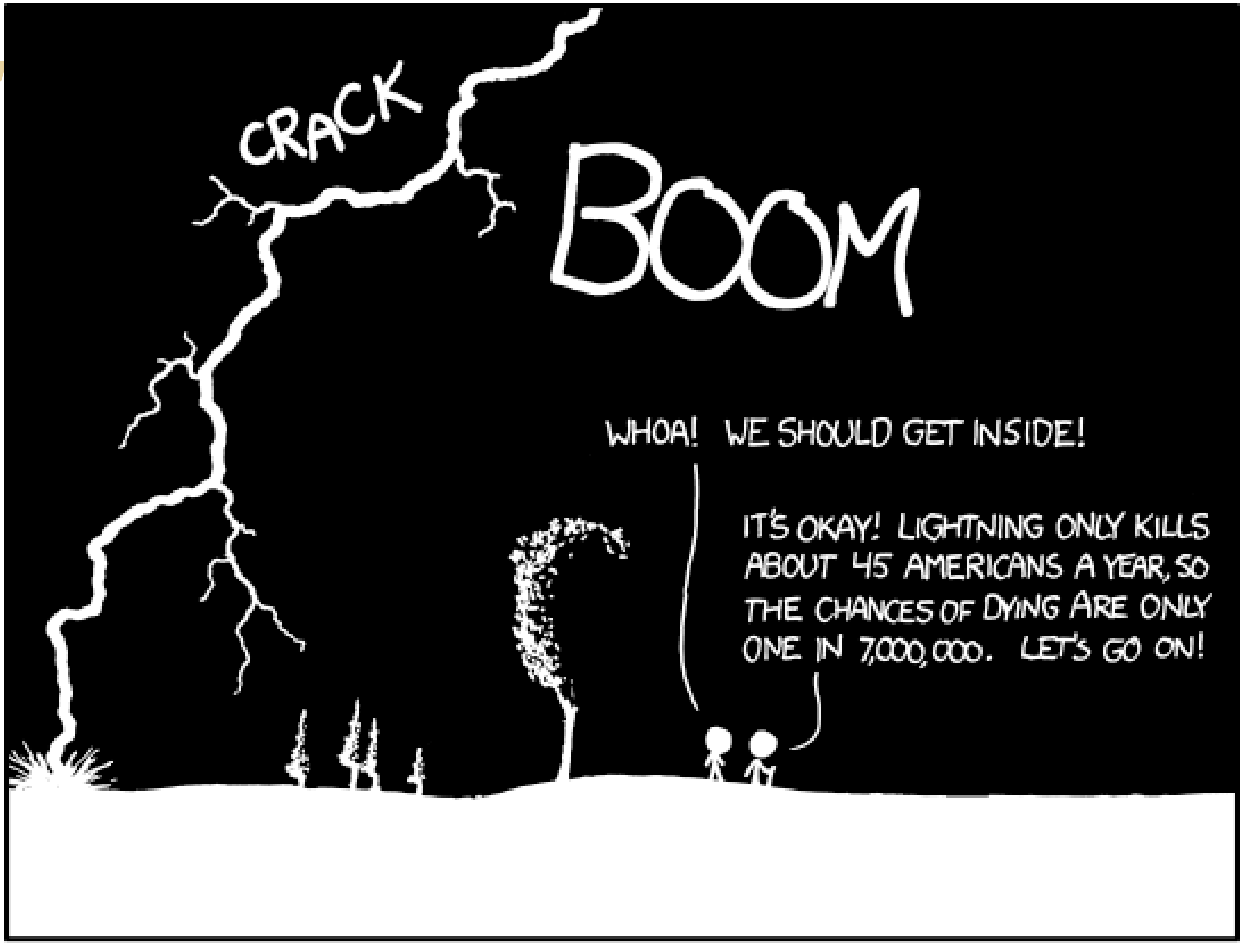
UNIVERSITY *of* WASHINGTON

Data Science UW Methods for Data Analysis

Probability and More on Distributions
Lecture 2
Nick McClure



THE ANNUAL DEATH RATE AMONG PEOPLE WHO KNOW THAT STATISTIC IS ONE IN SIX.

Topics

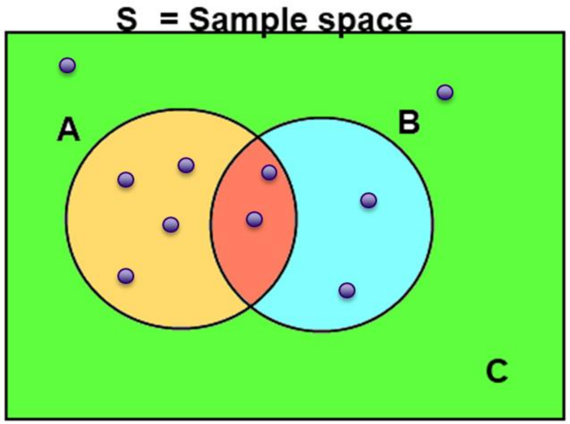
- Review
  - Counting
  - Axioms of Probability
- Probability Examples
- Conditional Probability
- More on Distributions
- Production R code
  - Unit tests and Logging

Probability Examples

> Probability of rolling a sum of 10?

Why is this False?

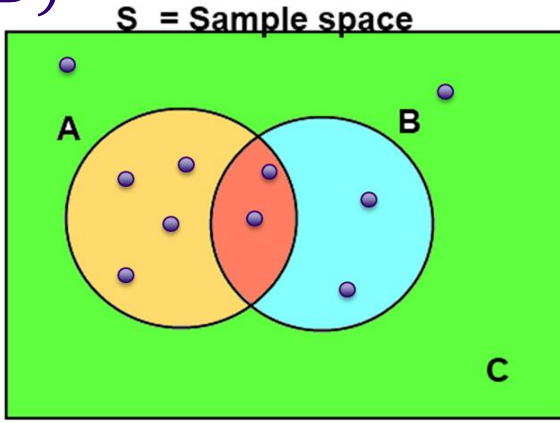
$$P(A \cup B) = P(A) + P(B)$$



Mutually Exclusive Events

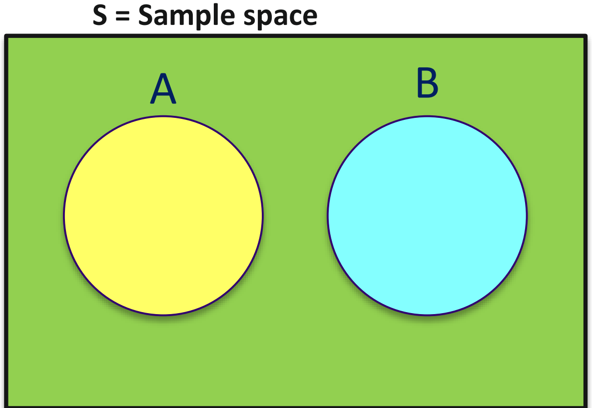
> In all cases, the probability of the union of A and B takes the form:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



> If A and B are mutually exclusive that means that

$$P(A \cap B) = 0$$
$$P(A \cup B) = P(A) + P(B)$$



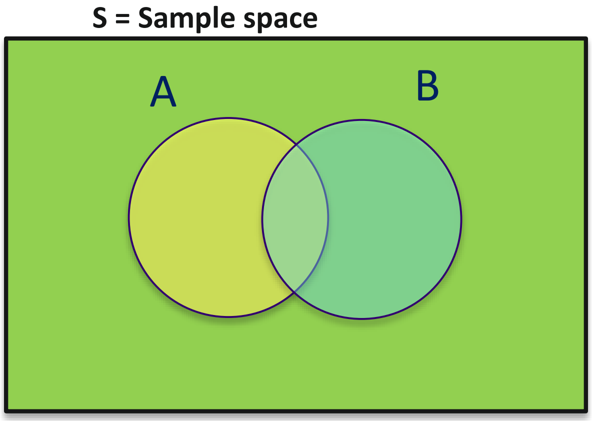
Conditional Probability

> The probability of A *given* B is written:

$$P(A|B)$$

> And is equal to:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$, compare to: $$P(E) = \frac{P(E)}{P(S)}$$



Independent Events

> Events A is independent of B if and only if:

$$P(A|B) = P(A)$$

> A being independent of B does NOT imply B is independent of A.

$$P(A|B) = P(A) \quad \not\Rightarrow \quad P(B|A) = P(B)$$

$$P(A|B) = P(A) = \frac{P(A \cap B)}{P(B)} \Rightarrow P(B)P(A) = P(A \cap B)$$

E.g. The event that my boss takes vacation has an impact on when I take vacation, but when I take vacation has no impact on when my boss takes vacation. (i.e., his vacation is independent of mine, but not vice versa)

Independence vs. Mutually Exclusive

- These are not similar AT ALL and in fact, are nearly opposite ideas.
- If A is M.E. of B then: $P(A|B) = 0$

B occurring has a HUGE impact on P(A)

- If A is independent of B then: $P(A|B) = P(A)$

Example: The probability the sidewalk is wet given it is raining is very high,
But the probability that it is raining given the sidewalk is wet is lower (if I run my sprinklers often).

Odds

> Odds are expressed as (Count in event favor):(Count not in event favor)
  - Make sure you reduce the fraction first

$$P(A) = \frac{n}{m} = \frac{n}{\underset{\uparrow}{n} + (m \underset{\uparrow}{-} n)}$$

Count in favor of A

Count not in favor of A

  - Implies the odds are:

$$n:(m-n)$$

Examples:

If P(A)=5/6, then the odds are 5:1. ‘Five to one’.

If the odds are 3:20, then P(A)=3/23

A straight up sports bet in Vegas has odds 1:1 (50%), but pays 0.95Xbet.

Monty Hall Problem

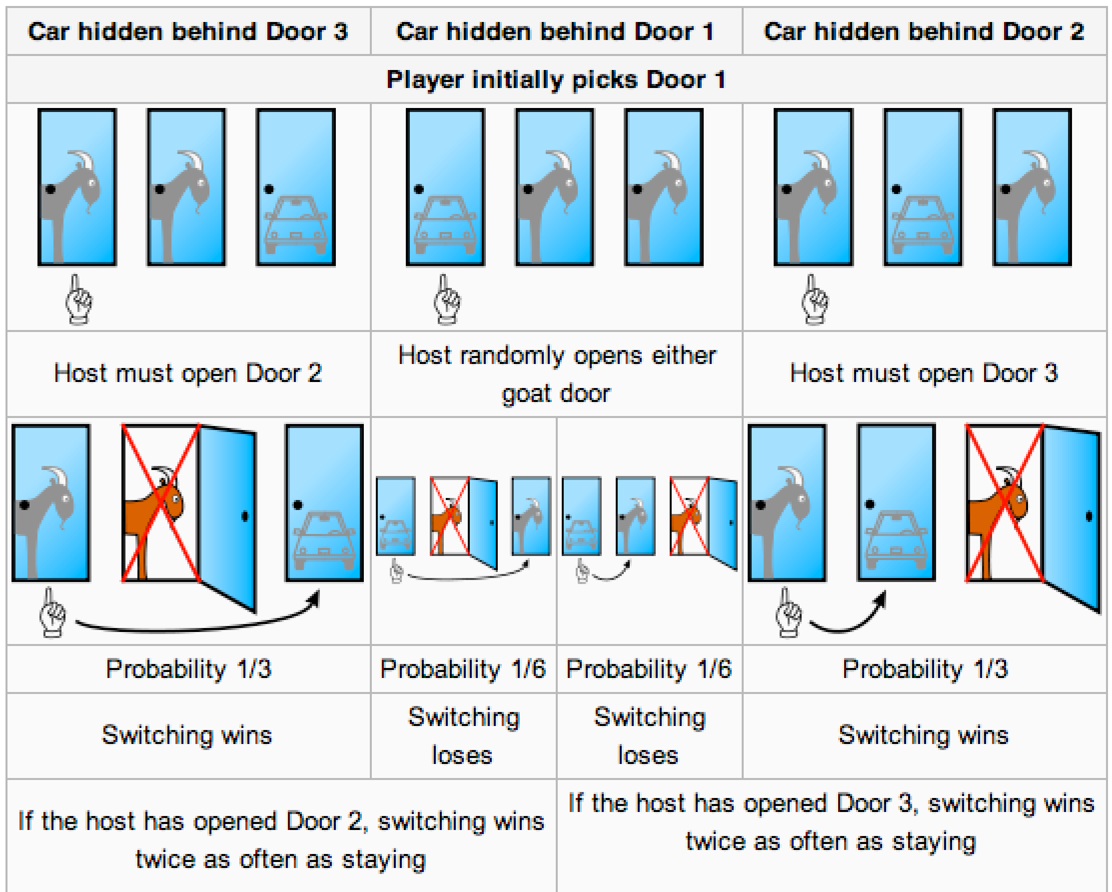
- Famous conditional probability problem that divided statisticians when it came out.
  - Start with 3 doors. One prize behind unknown door. Pick a door. Host reveals a separate door with no prize. Then contestant can switch. Should they?

Monty Hall Problem

> Start with 3 doors. One prize behind unknown door. Pick a door. Host reveals a separate door with no prize. Then contestant can switch. Should they?

Monty Hall Problem

> Start with 3 doors. One prize behind unknown door. Pick a door. Host reveals a separate door with no prize. Then contestant can switch. Should they?



Monty Hall Problem

– http://www.stayorswitch.com/

Back to Die Rolling…

– Consider the probabilities of all potential sums of 2 die:

P(2)=1/36
P(3)=2/36
P(4)=3/36
P(5)=4/36
P(6)=5/36
P(7)=6/36
P(8)=5/36
P(9)=4/36
P(10)=3/36
P(11)=2/36
P(12)=1/36
Sum(all) = 36/36 = 1

	2	3	4	5	6	7
	3	4	5	6	7	8
	4	5	6	7	8	9
	5	6	7	8	9	10
	6	7	8	9	10	11
	7	8	9	10	11	12

If we consider all possibilities together, this is called a *distribution*.

Data Distributions (Discrete)

- Discrete Distribution Properties
  - Sum of all events must equal 1.
  - Probability of event equal to value of distribution at point.
  - No Negative values or values greater than 1.

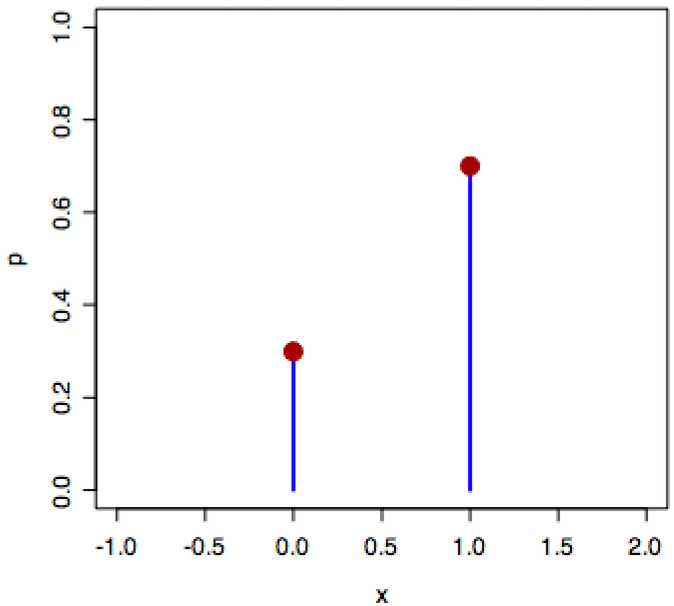
Data Distributions (Discrete)

> Bernoulli (1 event, e.g.: coin flip)

$$P(x) = \begin{cases} p \; if \; x = 1 \\ (1-p) \; if \; x = 0 \end{cases}$$

$$P(x) = p^x(1-p)^{(1-x)} \quad x \in \{0,1\}$$

- Mean = p
- Variance = p(1-p)



Data Distributions (Discrete)

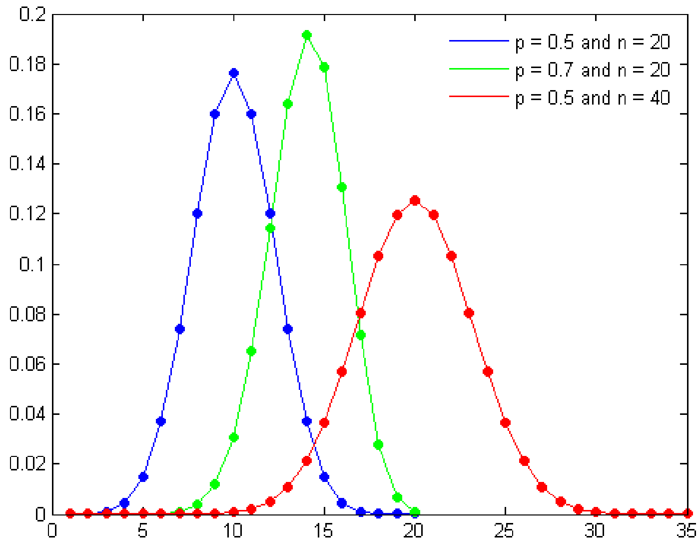
> Binomial (Multiple Bernoulli's Events)

– Multiple Independent events = Product of Bernoulli Probabilities

$$P(x|N,p) = \binom{N}{x} p^x (1-p)^{(N-x)}$$

– Mean = np

– Variance = np(1-p)



Note: for larger n, we approximate this by a normal distribution.

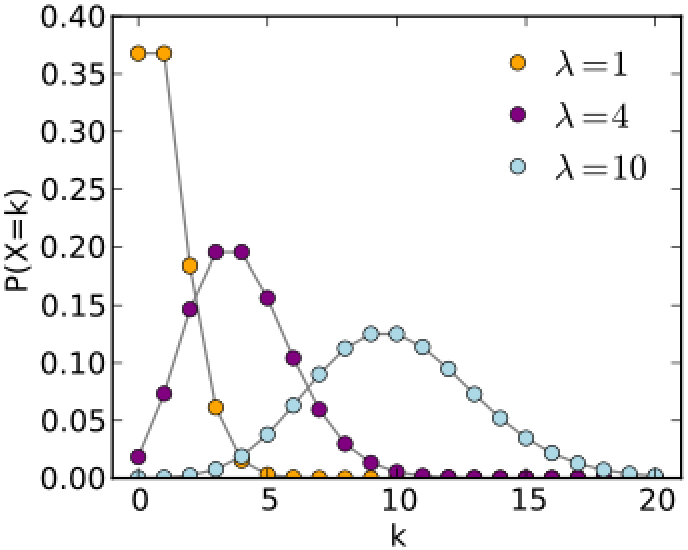
Data Distributions (Discrete)

> Poisson (Count of number of events in a time span)

$$P(x|\lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$$

- Mean = λ
- Variance = λ

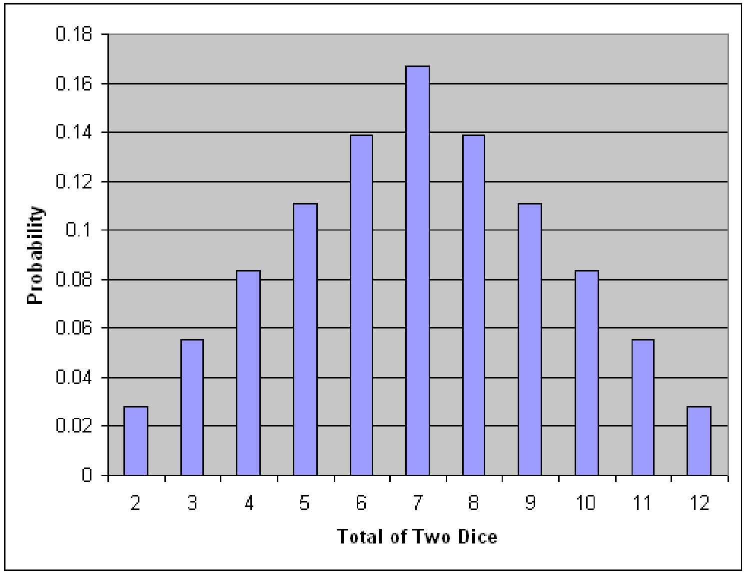
Interpret as the rate of occurrence of an event is equal to lambda in a finite period of time.



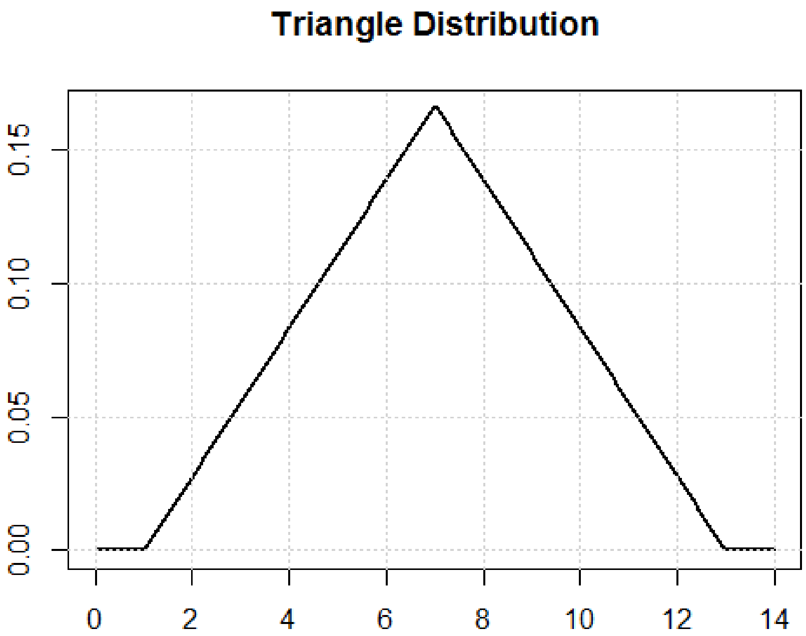
Data Distributions (Continuous)

> Continuous Distribution Properties
  - Area under the curve must be equal to 1.
  - Probability of event equal to AREA under curve.
  - No negative values.
  - Probability of a single, exact value is 0.

Discrete Distribution



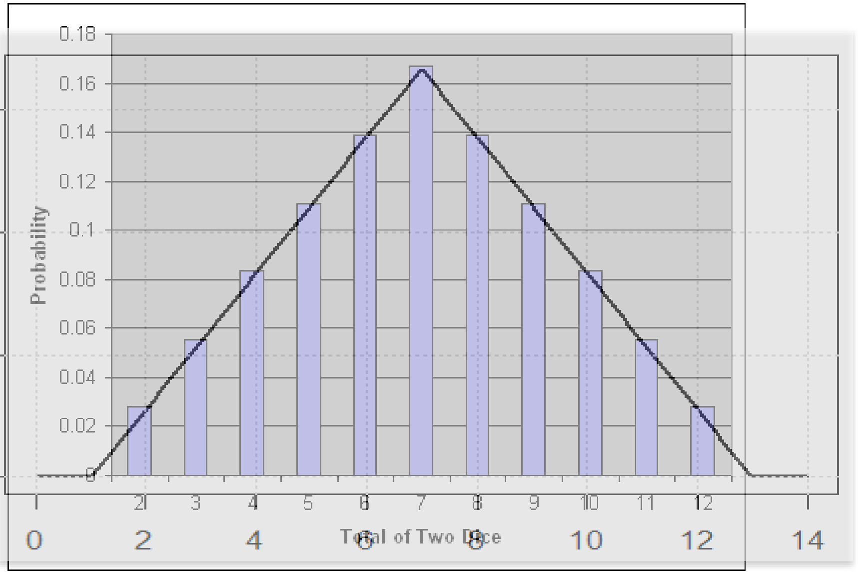
Continuous Distribution



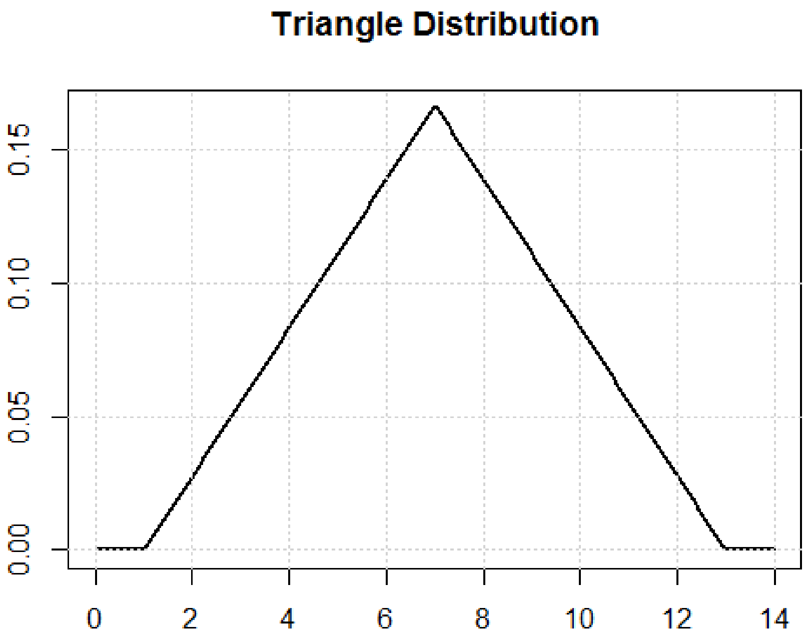
Data Distributions (Continuous)

> Continuous Distribution Properties
  - Area under the curve must be equal to 1.
  - Probability of event equal to AREA under curve.
  - No negative values.
  - Probability of a single, exact value is 0.

Discrete Distribution



Continuous Distribution



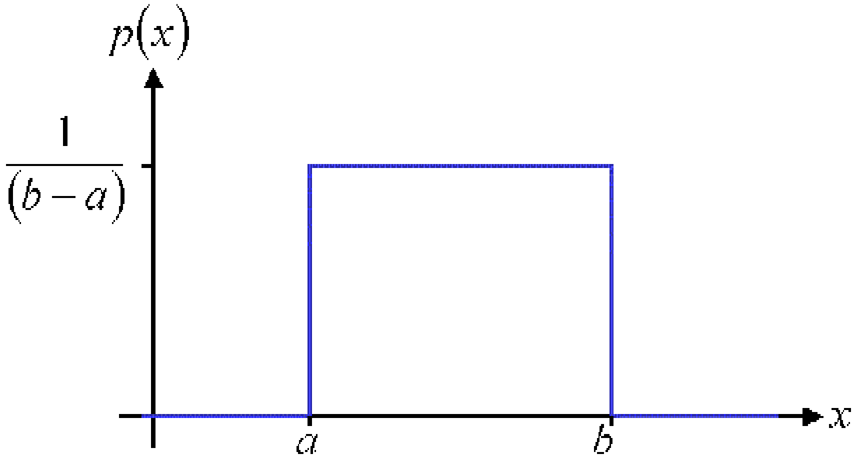
Data Distributions (Continuous)

> Uniform (flat, bounded)

$$P(x) = \begin{cases} \dfrac{1}{(b-a)} \; if \; a \leq x \leq b \\ 0 \; if \; x < a \; or \; x > b \end{cases}$$

> Very useful for parameter priors. (future discussion)

- Mean=(a+b)/2
- Variance=(1/12)(b-a)^2

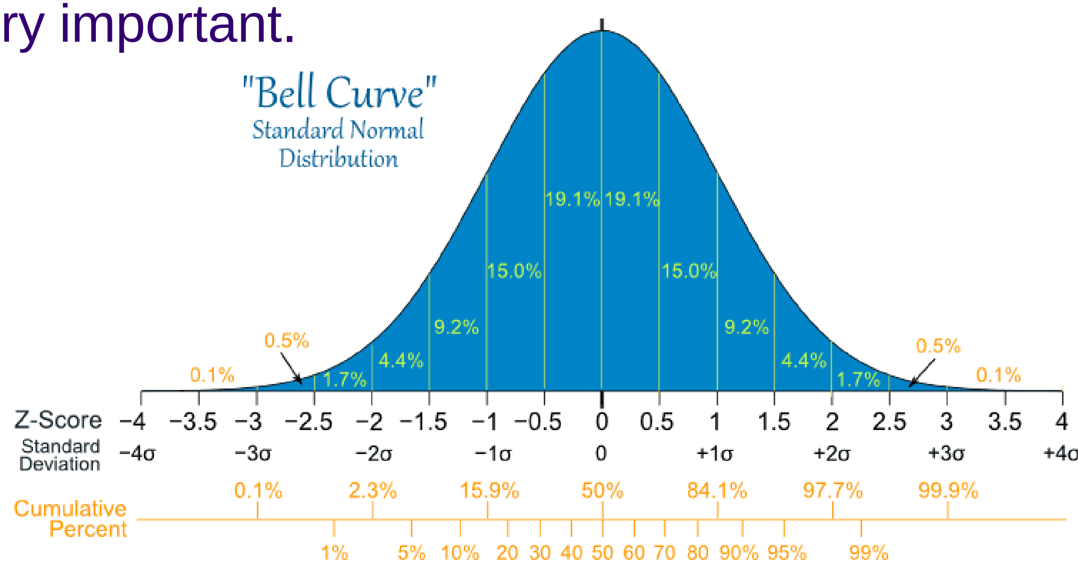


Data Distributions (Continuous)

- Normal (Gaussian) distribution
  - Most common and occurs naturally.
  - Defined by a mean and variance only. (standard = N(0,1))

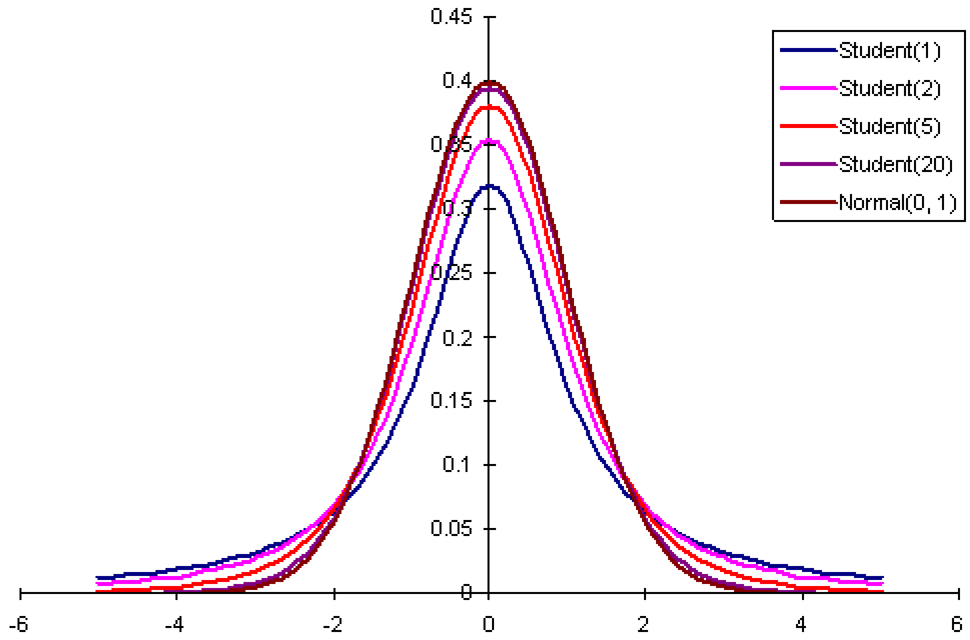
$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

  - Has very nice properties.
  - Tests for normality are very important.



Data Distributions (Continuous)

> Student's T (normal for small samples)
  - Important for hypothesis testing smaller sample sizes.
  - Used for:
    > Testing of mean value when st. dev. is unknown.
    > Testing difference between two distribution means.
  - Looks very similar to the normal distribution.



Distribution Transformations

- The purpose of transforming a variable is to make it easier to distinguish between values.
  - Most commonly we are looking to transform a distribution to be normal.
- Common Transformations
  - Log-based:
    - Log(x), log(x+1), log(x-min(x) + 1)
  - N-th Root based:
    - X^(1/n)
  - Any combination you can think of (remembering math rules).
- We will cover normality tests in a later class.

Simulations in R

- Simulations are used to **verify** probabilities.
  - Why important in business? Need to convince non-statisticians of probabilistic outcomes.
  - In other words, try not to make any statistical assumptions in simulations.
- With these, we can also estimate variation in probabilities.
- Use system.time() from base or microbenchmark() from microbenchmark package.
- Clean up after yourself:
  - gc() or invisible(gc())
- R demo

Dealing with Missing Data

- Reasons for missing data
  - Recording failure (mechanical/software failures)
  - Reporting failure (human decisions)
  - Translation failure (data transferring/parsing errors)
- Many shapes and types
  - Shapes: block, regular, random, sparse
  - Types:
    - Missing At Random (MAR): a particular variable has randomly omitted data.
    - Missing Completely At Random (MCAR) : every piece of data has equal chance of being omitted.
    - Missing Not At Random (MNAR): The value of data is related to chance of being omitted.
- Outliers may also be treated as missing data.

Dealing with Missing Data

Type	Benefits	Disadvantages	Notes
Drop Missing	-Speed	-Data Loss	
Mean/Median/Mode Fill	-No Data Loss	-Variance Reduction	
X~F(independents)	-More Accurate -No Data Loss	-Slower	-Needs most columns to be filled out -Harder on ind. data
knn	-More Accurate -No Data Loss	-Slower -Dependent on distance function	
X~F(y,independents)	-Very accurate -No Data Loss	-Slower -Need y	-Only on training set!

Dealing with Missing Data: Variance and Multiple Imputation

- Dealing with imputation, it is important to try and keep the intrinsic variance in the data set.
- To achieve this, multiple different predictions are made for each missing data point. (Using previous methods)
- These data sets are kept and future hypothesis testing and predictions are made on all imputed sets to gauge the variance in the outcomes.
- R package ‘Amelia’ does this and creates a nested list of data frames.
- Amelia R demo

Dealing with Missing Data: Using Outside or New Data Sources

- Don’t forget to explore outside or new data sources to help fill-in missing data.
- With the advent of free public data and bigger data sources, this is gaining popularity as a tool for imputation.
- Unstructured text is a major source of data.
- Ex:
  - Caesar’s uses public reviews on websites to mine for customer sentiment about hotel rooms.
  - Zillow uses text descriptions of properties to fill in missing data about # bedrooms, # bathrooms, sq. footage, and various amenities.
  - Subject to human stupidity.

Yelp Rating for Circus-Circus: 2/5
Text Description: “My son and I stayed here. The service was great, the room was great, but it turns out my son is deathly afraid of clowns.”

Getting Data

- Files
  - Csv: read.csv
  - Txt: read.table
- Web/HTML
  - readLines
  - XML, xpath
  - http://gastonsanchez.com/work/webdata/getting_web_data_r4_parsing_xml_html.pdf
- API
  - Twitter Example
  - Get consumer/access keys here:
    - https://dev.twitter.com/apps

Storing Data

- .csv – write.csv()
- .txt – write.txt()
- .Rdata – save()
  - Workspaces are very compressed compared to csv
- Databases
  - Sqlite: sqldf, RSQLite packages
    - Sqlite example
  - MongoDB: rmongodb package
  - Postgresql: RPostgreSQL package

Production Level Scripts

- Logging
- Functionalize everything possible
- interactive()
- One Unit Test
- R-example: Weather Scraping R script

Unit Tests

- The purpose of unit tests is ensure the *functionality* of your programs.
- Situations averted by using unit tests:
  - Allows for big changes to code structure to be quickly tested.
  - Helps to realize when we can stop coding.
    - E.g., all foreseeable test cases are covered.
  - Writing tests helps organize code structure.
  - A way to get instant feedback on coding.
  - Good tests help document and define the scope of functions.
  - Make sure that other people using your code don’t break it.
  - “Find a bug, write a unit test for it, fix the bug”, implies that the bug will never appear again.

Unit Tests

- Good unit tests:
  - Test that a function runs over all possible input cases.
    - E.g., a ‘text cleaning function’ cleans lowercase, upper case, punctuation, Unicode, etc…
  - Testing for data structures and integrity.
    - E.g., a data file exists, it was loaded correctly, and that the loaded input is a specific type or structure.
- Bad unit tests:
  - Tests that *might* fail due to the probabilistic nature of the test.
    - E.g., Test that a statistical procedure results in a specific probability.
    - E.g., Testing for a remote server response.
  - Too large tests.
    - E.g., Testing that a whole program or multiple functions ran without error.
  - Complicated tests.
    - E.g., Testing a model fit to a large data set.

Assignment

- Complete Homework 2:
  - Write an R-script to verify the Monty Hall Probabilities with simulations (get probabilities AND variances for switching and not switching).
    - Note that you should do **TWO SEPARATE** simulations for switching and not switching. You will lose points if you do only one simulation.
  - You should submit:
    - **ONE Production level R-script** that outputs the probabilities and variances.
    - Submission should include a text document/log file of your results.
  - Read Intro to Data Science Chapter 7 and 10.
  - Read Statistical Thinking for Programmers Ch. 4.
  - Send an email proposal for your project.
    - nfmcclure@gmail.com or nickmc@uw.edu