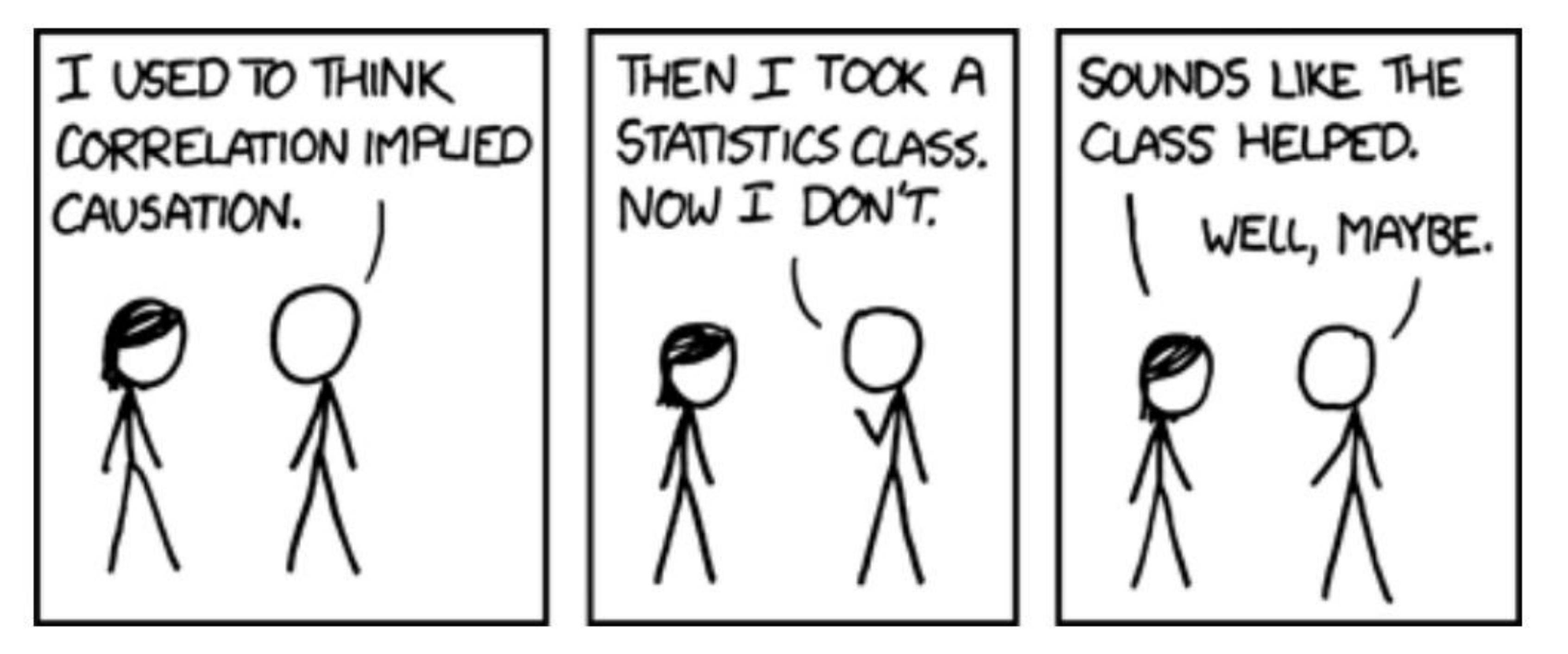
Intro to Statistics Part I: Types of data, distributions, random variables, and randomness

AST1501 Guest Lecture, Nov. 23, 2023

Prof. Gwendolyn Eadie

Slides adapted from what was formerly "Starfish School" (thank you's to Mubdi Rahman, Renee Hlozek)

Introduction to Basics in Statistics



https://xkcd.com/552/

Types of Data

Types of Data

Quantitative

- Continuous
  - Real or complex numbers
- Discrete
  - integers

Categorical

- Nominal
  - e.g., categories A, B, C, or I, II, III
- Ordinal
  - Ordering matters, e.g., a *Likert Scale* used in a survey: 1,2,3,4,5

Types of Data

Quantitative

- Continuous
  - Real or complex numbers
- Discrete
  - integers

Categorical

- Nominal
  - e.g., categories A, B, C, or I, II, III
- Ordinal
  - Ordering matters, e.g., a *Likert Scale* used in a survey: 1,2,3,4,5

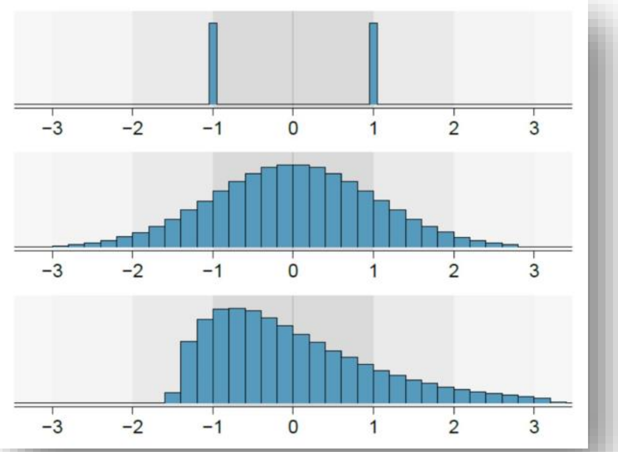
What astronomy examples can you think of for each type?

Distributions

What exactly is a distribution?

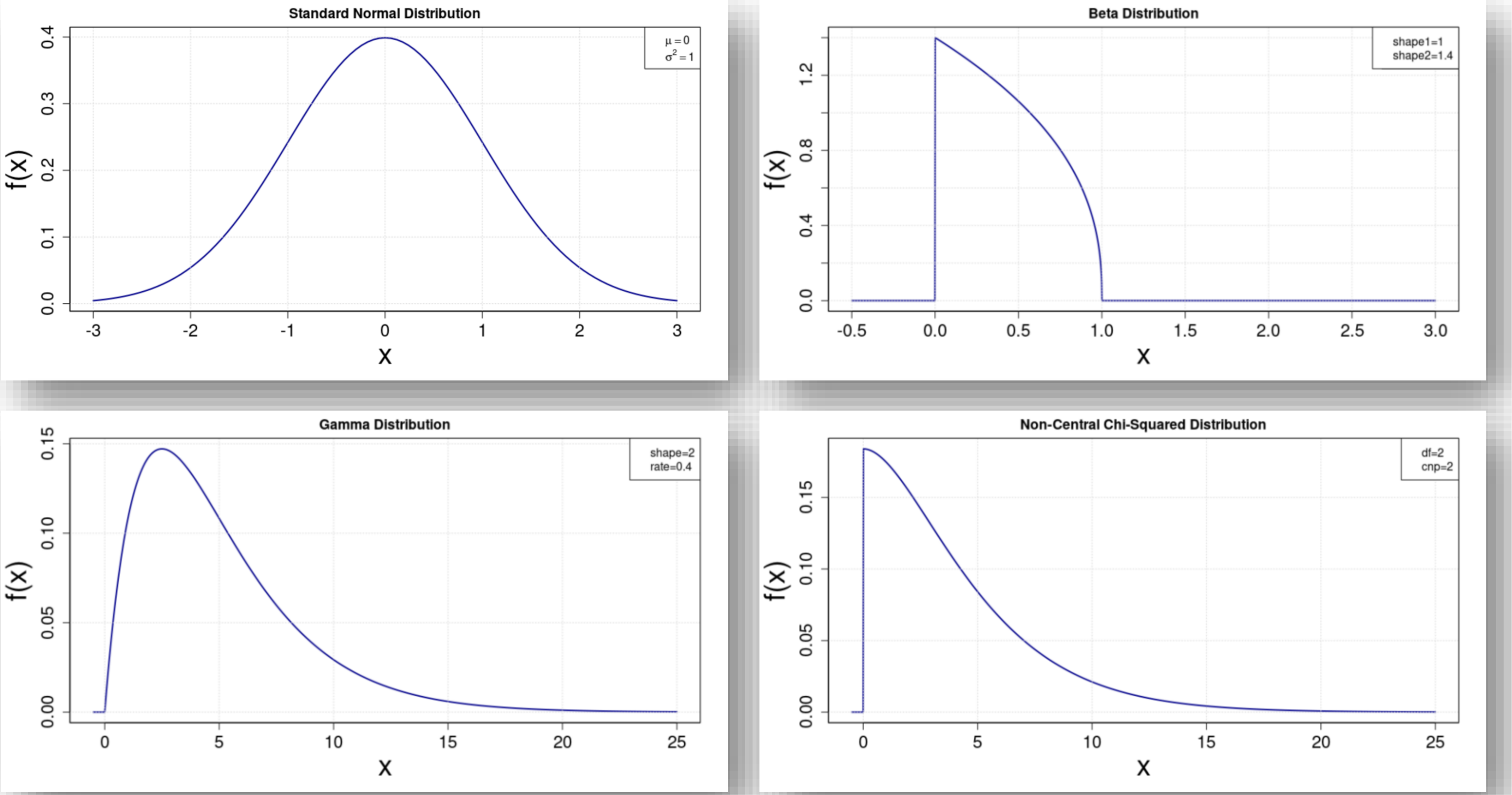
A distribution…

- Tells you the frequency or relative frequency of each possible value/event, or of some data that was collected
- **Could be empirical or analytic**
- Can be useful for modelling a population of objects
- Is often a foundation of statistical reasoning
- Can be continuous or discrete
- That is analytic has parameters that define its shape
- Can be univariate or multivariate



Example histograms (figure from Open Intro Statistics 4th ed.)

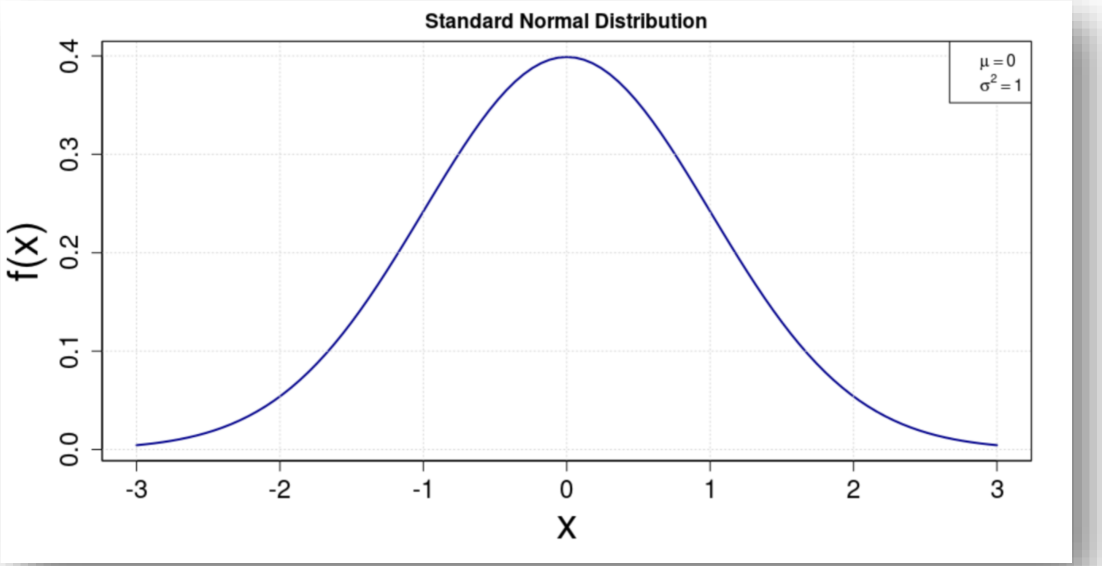
Some analytic probability distributions (plotted by me)



Probability Distributions

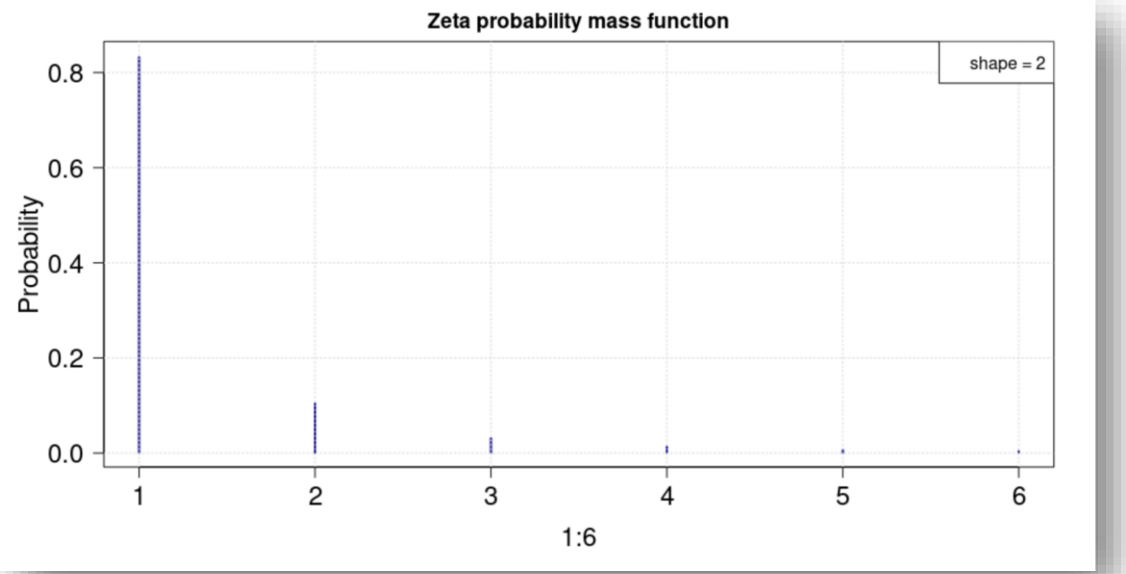
Continuous quantities

probability density function (pdf)

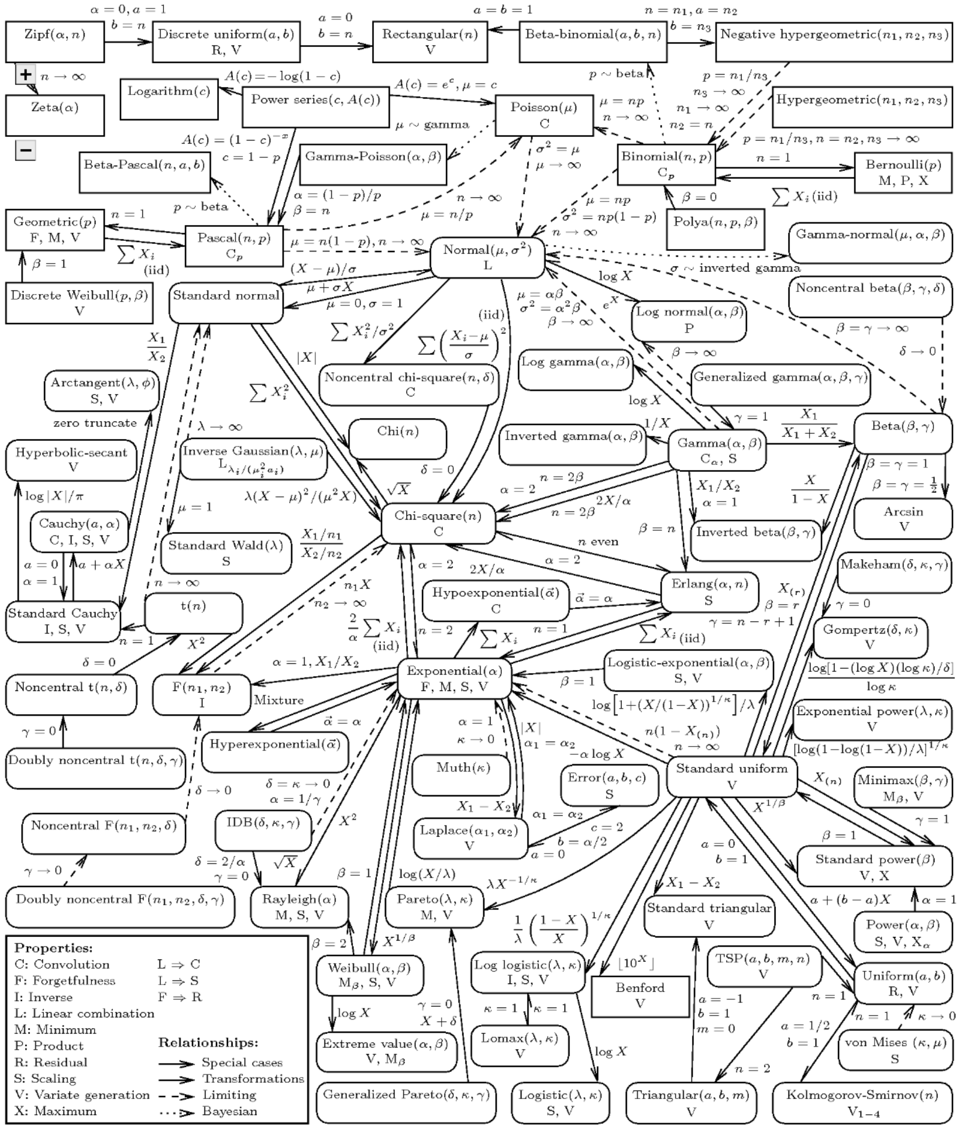


Discrete quantities

Probability mass function (pmf)

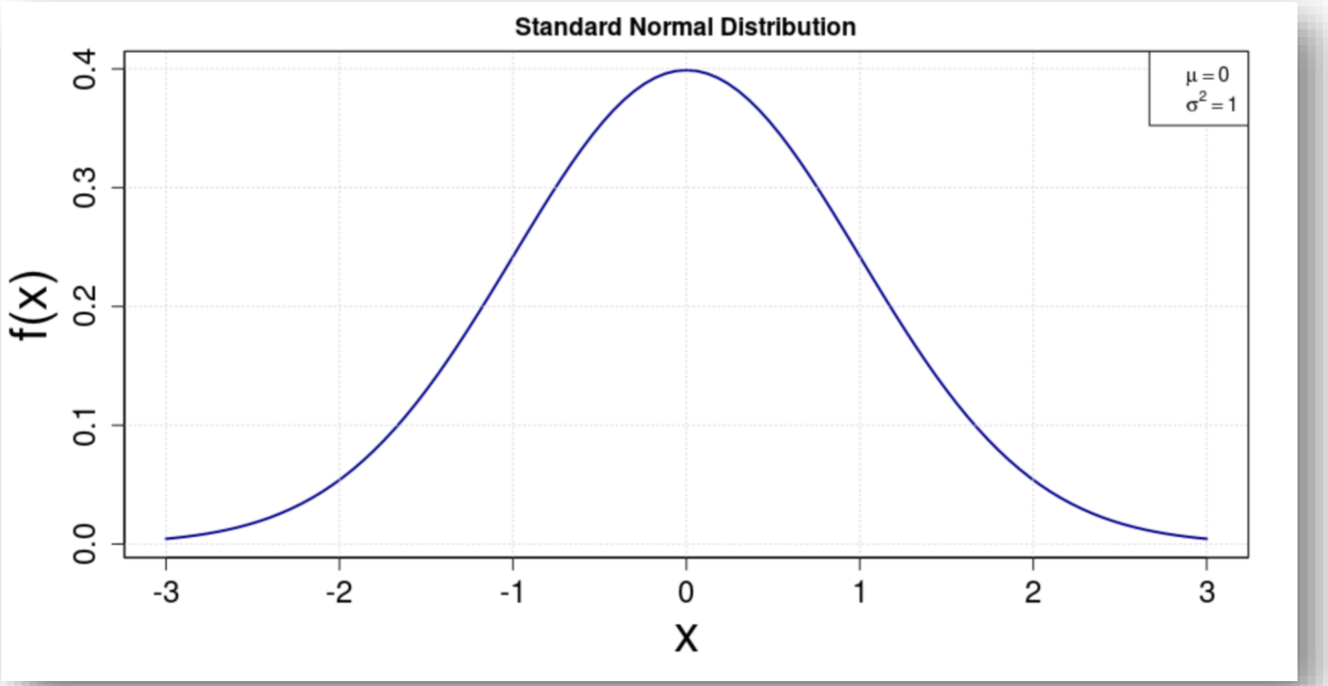


There are many univariate distributions!



http://www.math.wm.edu/~leemis/chart/UDR/UDR.html

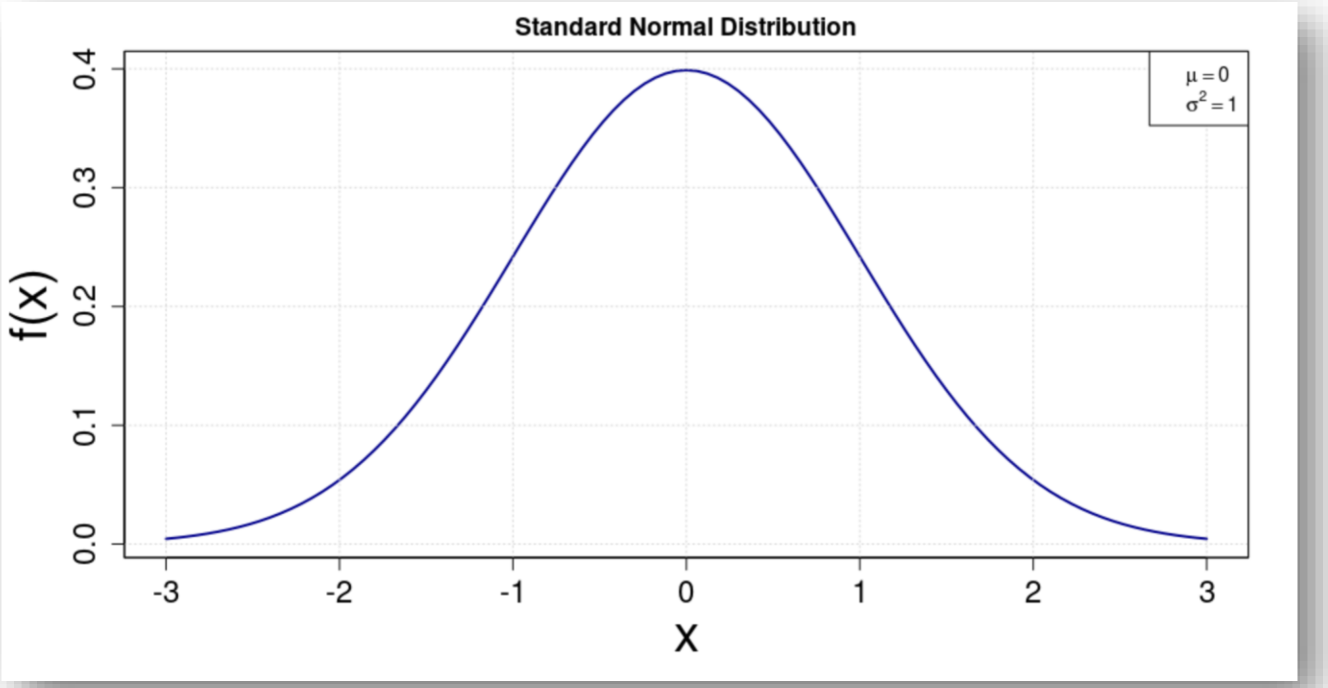
The Normal Distribution



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

The Normal Distribution

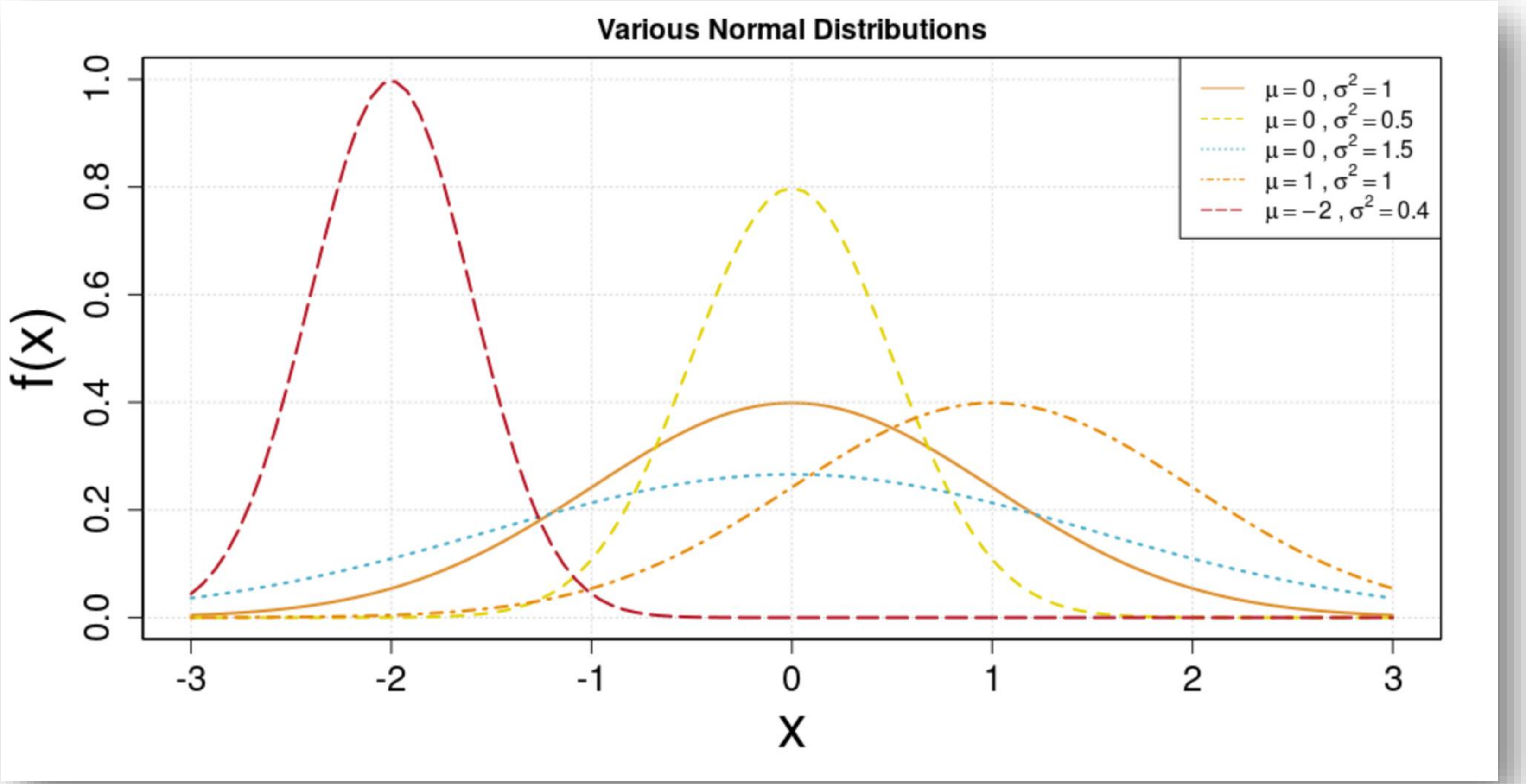
$N(\mu, \sigma^2)$



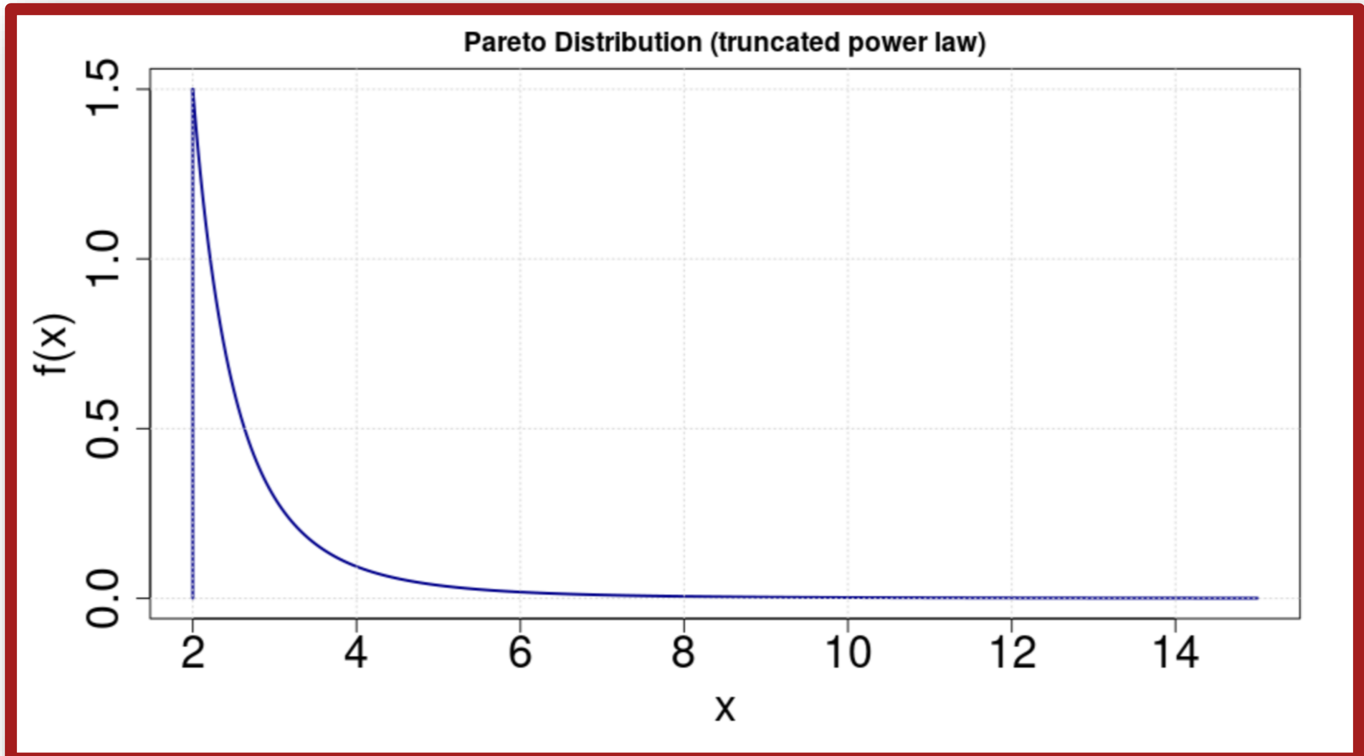
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

The Normal Distribution

The mean and variance are all you need to plot the Normal



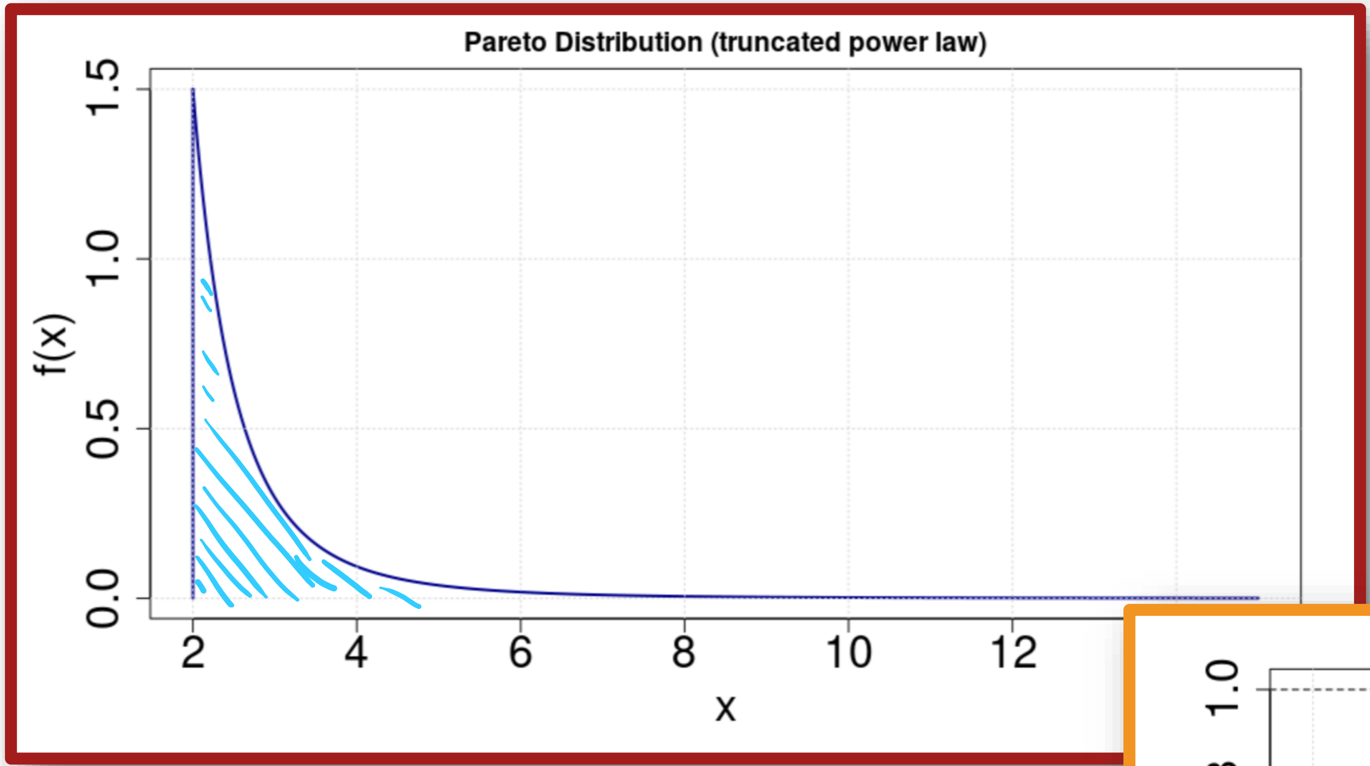
Example:
Pareto distribution (truncated power-law)



Probability distribution function (pdf)

$$f(x) = \frac{\alpha x_{\min}^{\alpha}}{x^{\alpha+1}}$$

Example:
Pareto distribution (truncated power-law)

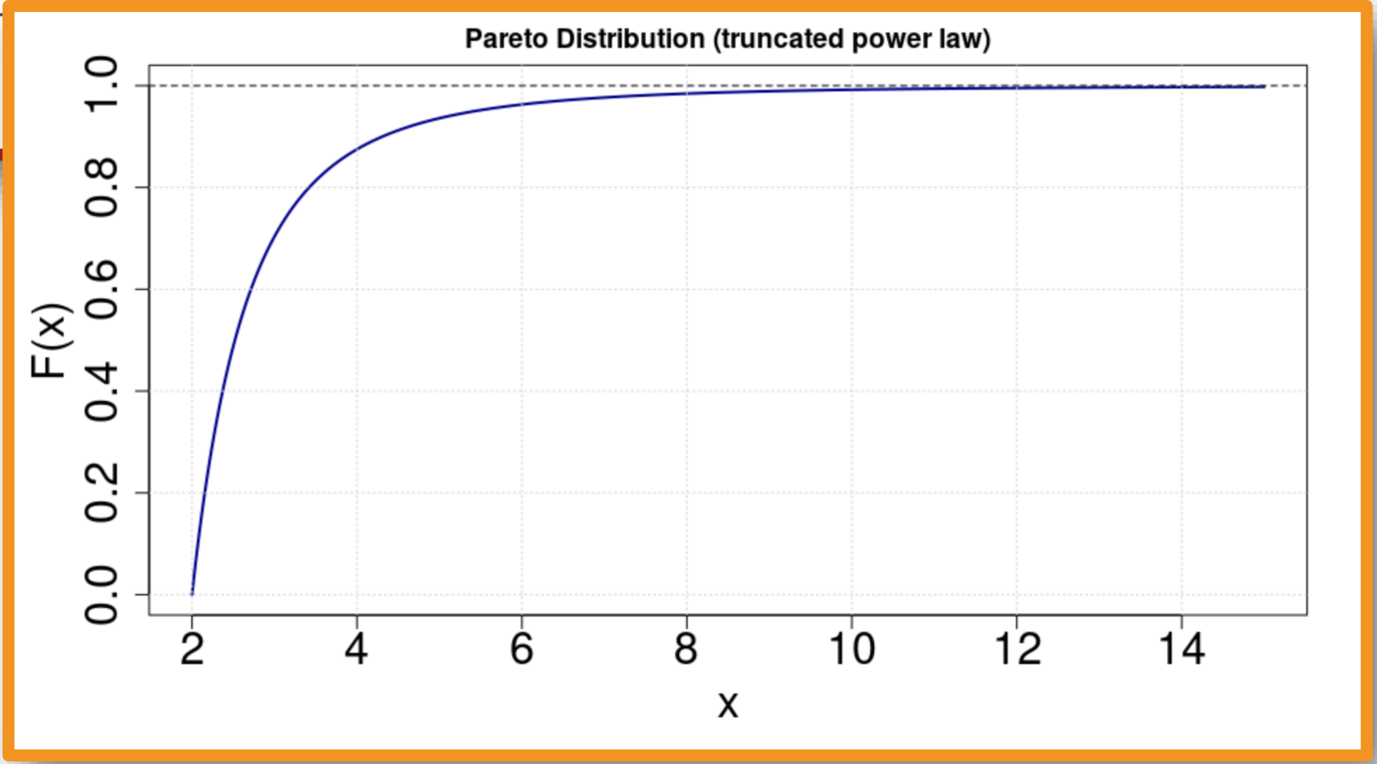


Probability distribution function (pdf)

$$f(x) = \frac{\alpha x_{\min}^{\alpha}}{x^{\alpha+1}}$$

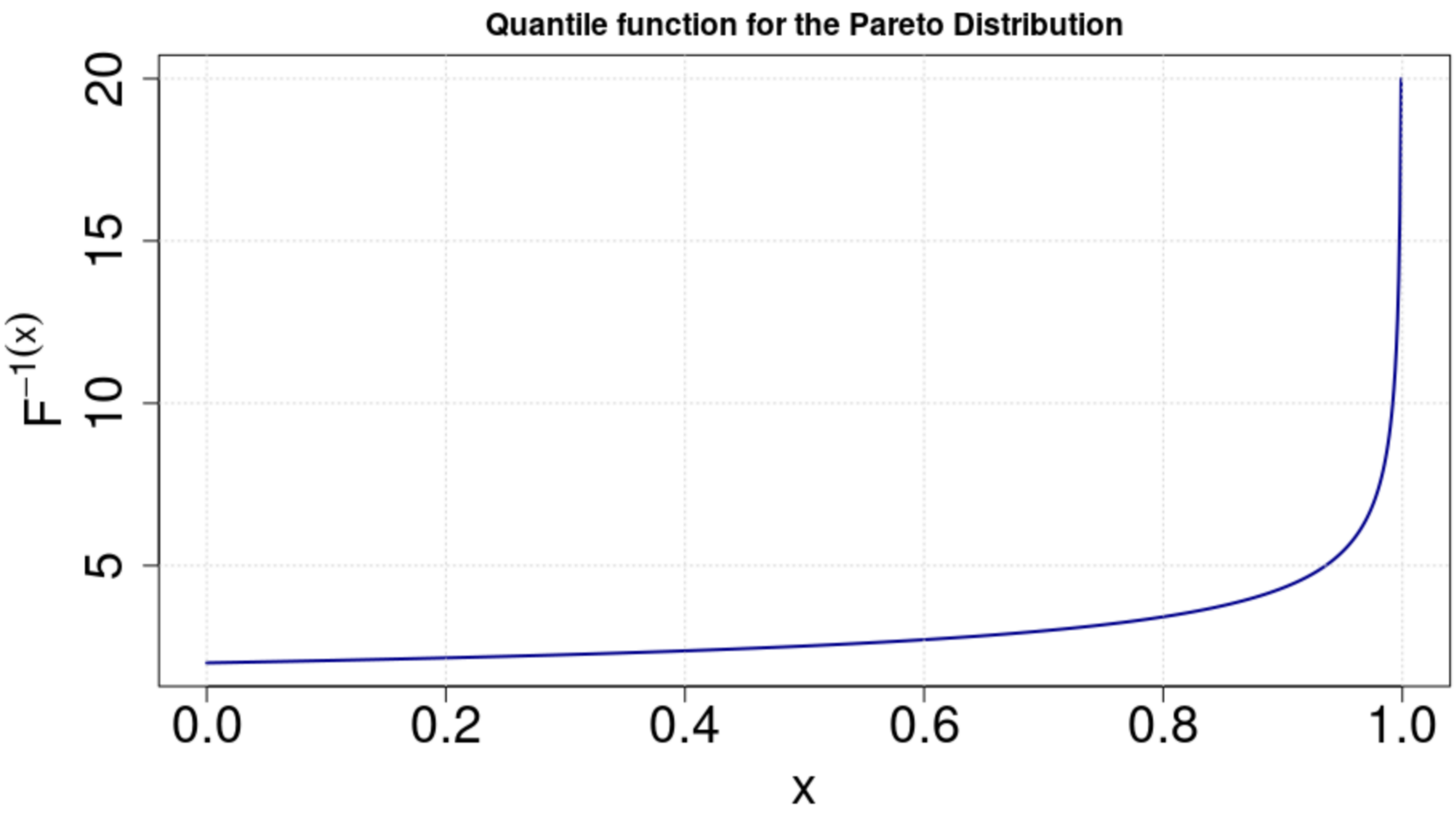
$$F(x) = P(X \leq x) = 1 - \left(\frac{x_{\min}}{x}\right)^{\alpha}$$

Cumulative distribution function (cdf)



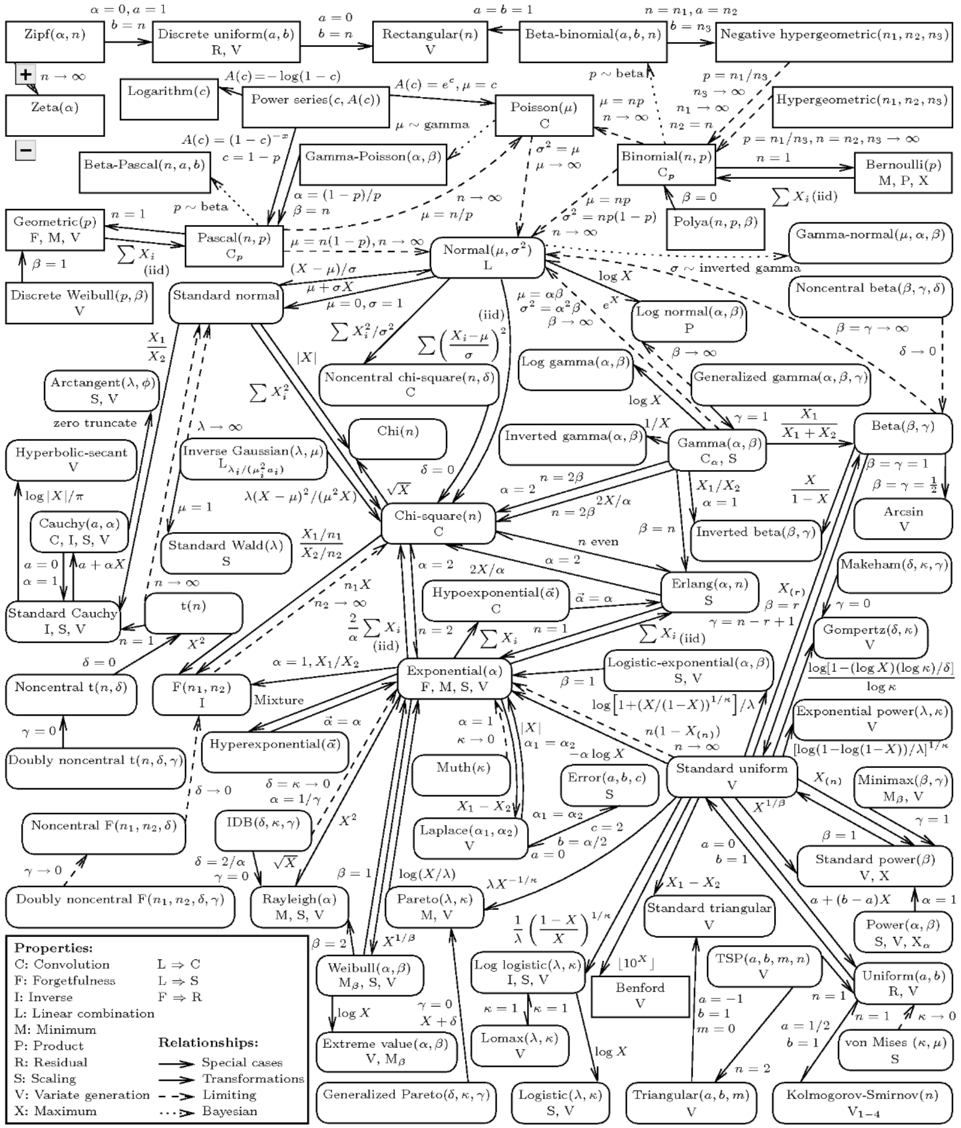
Quantile Function

- This is the inverse of the cumulative distribution function (cdf)



This chart has a pdf file for every distribution, including pdf, cdf, etc.

There are many univariate distributions!



http://www.math.wm.edu/~leemis/chart/UDR/UDR.html

Random Variables

Random Variable

- A random variable X is a *function* that maps an *outcome* to a *real number*
    - e.g., Let's say we decide to flip a coin repeatedly, and each time we flip the coin we record whether we get heads or tails with a 1 or a 0 respectively.
    - *We are mapping the outcome of the flip (heads or tails) to a numeric value (a 1 or a 0)*

- In other words, X is a **function**. Little x represents the data --- *realizations* of that random variable.

A Random Variable follows a distribution

The standard statistics notation to show what distribution a random variable follows is:

$$X \sim N(\mu, \sigma^2)$$

For example, we might assume that are data x (e.g. the photon counts from a star) follows a Poisson distribution

$$X \sim Pois(\lambda)$$

A Random Variable follows a distribution

The standard statistics notation to show what distribution a random variable follows is:

$$X \sim N(\mu, \sigma^2)$$

For example, we might assume that are data x (e.g. the photon counts from a star) follows a Poisson distribution

$$X \sim Pois(\lambda)$$

Poisson distribution is a discrete probability distribution often used to describe count data

A Random Variable follows a distribution

Another example:

$$X \sim Bernoulli$$

$$X = \begin{cases} 1 \text{ if success with probability } p \\ 0 \text{ if failure} \end{cases}$$

If we have n Bernoulli trials, then the sum of these are distributed as

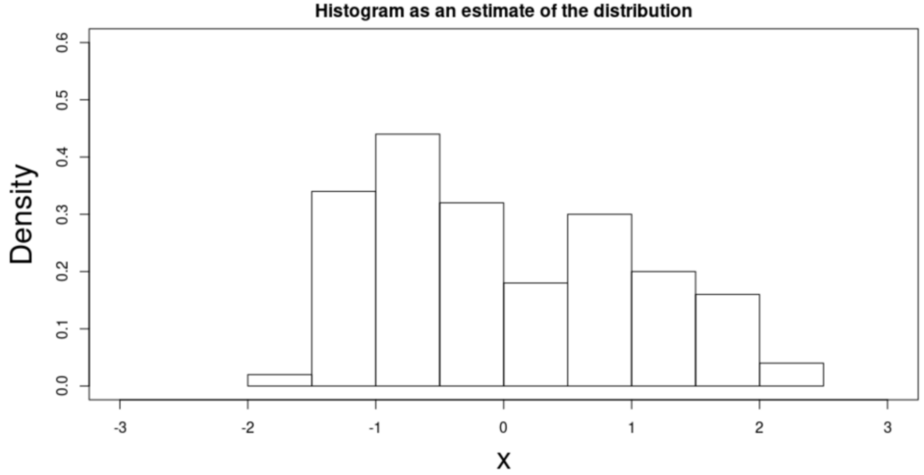
$$Y \sim Binom(n, p)$$

where $Y = \sum_i^n X_i$

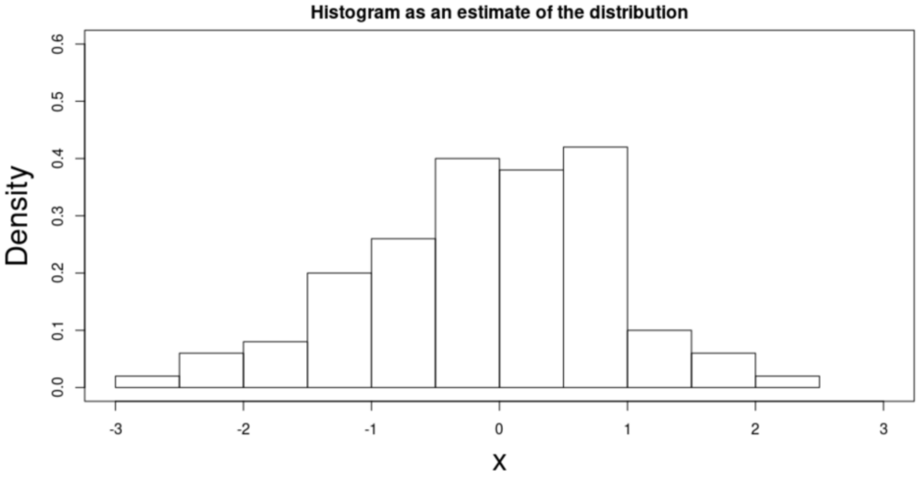
Randomness in Data

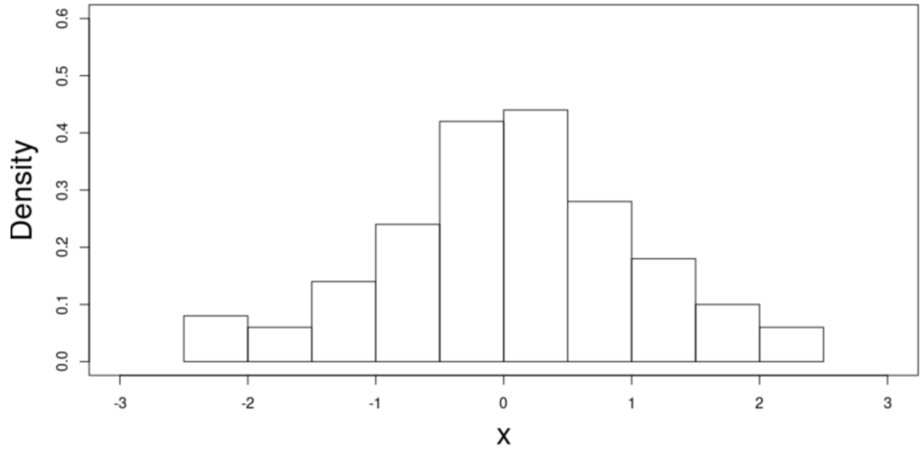
Randomness in Data

- All these histograms were generated from 100 draws from a standard normal

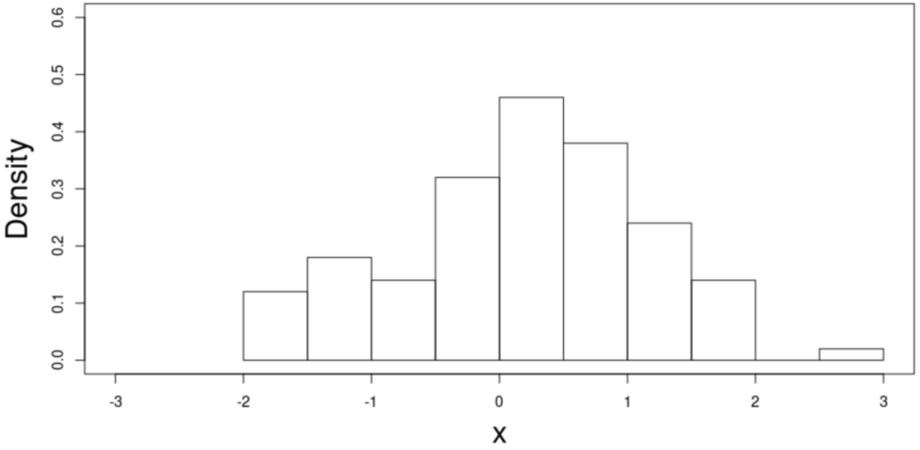


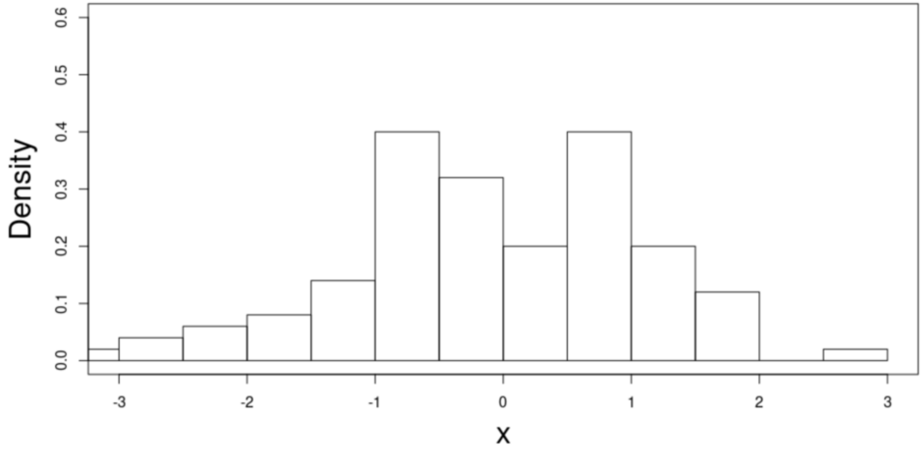
From the data, we can try to *estimate* the true distribution. We can also try to estimate the underlying parameters of the distribution.





These estimates are **random variables.**

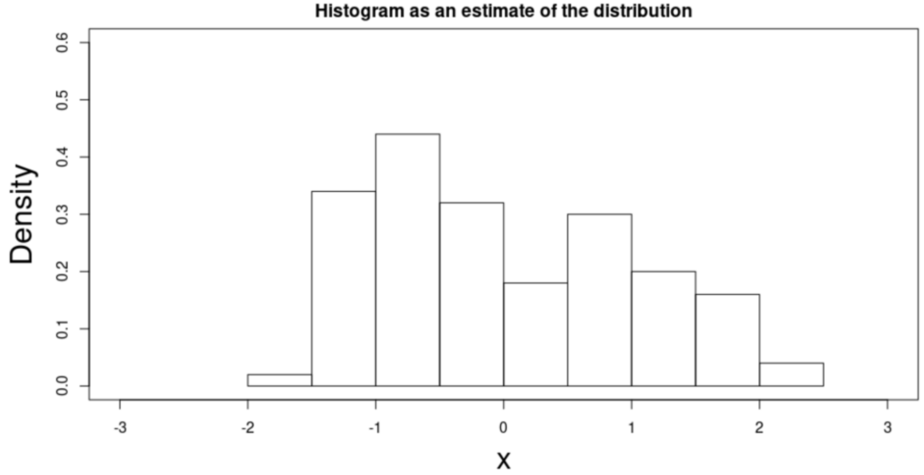


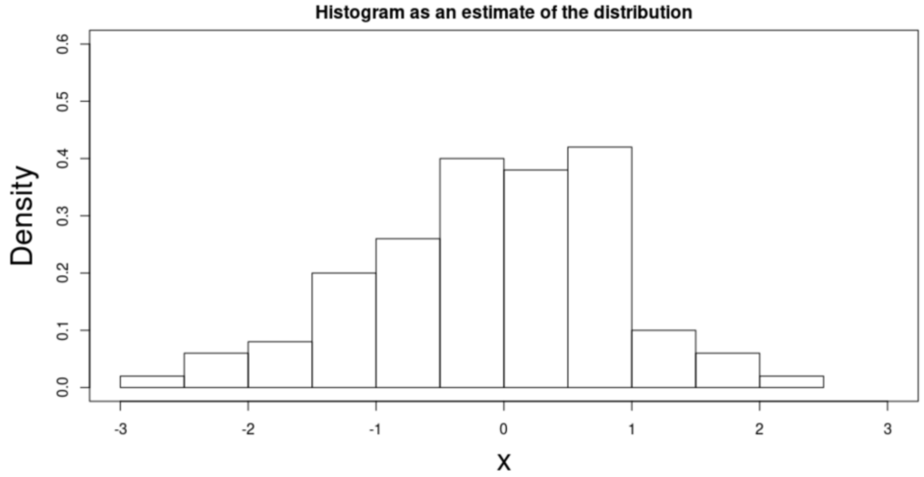


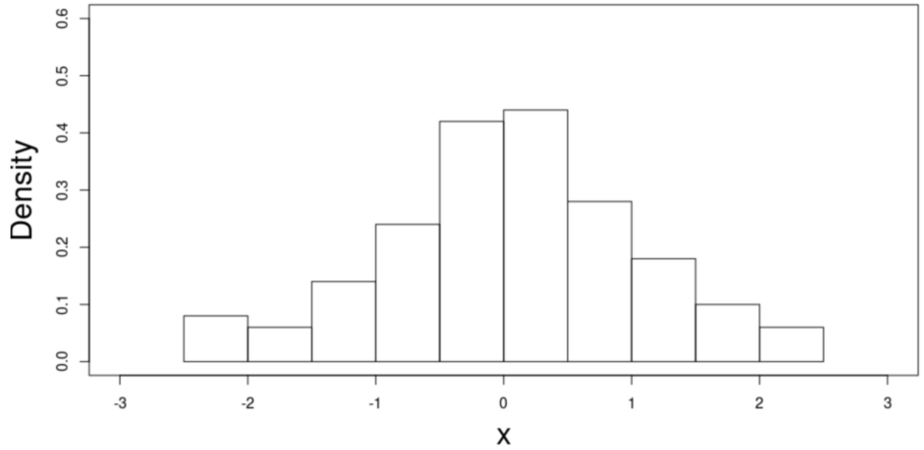
Randomness in Data

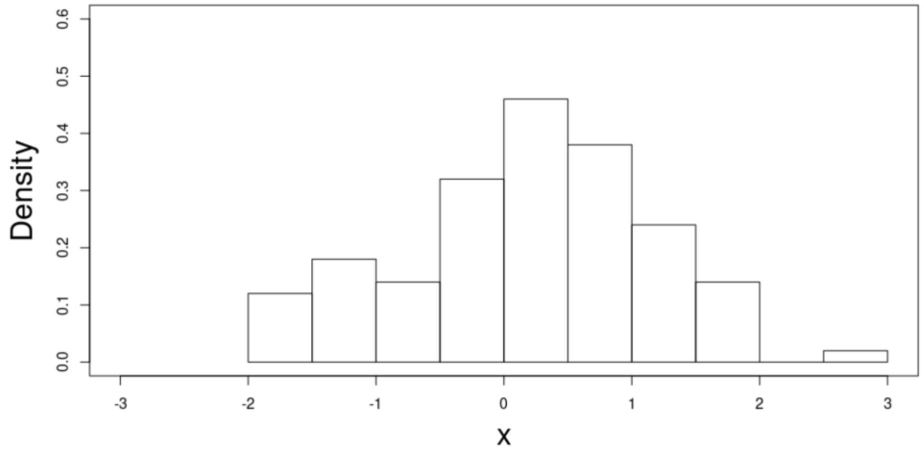
- All these histograms were generated from 100 draws from a standard normal

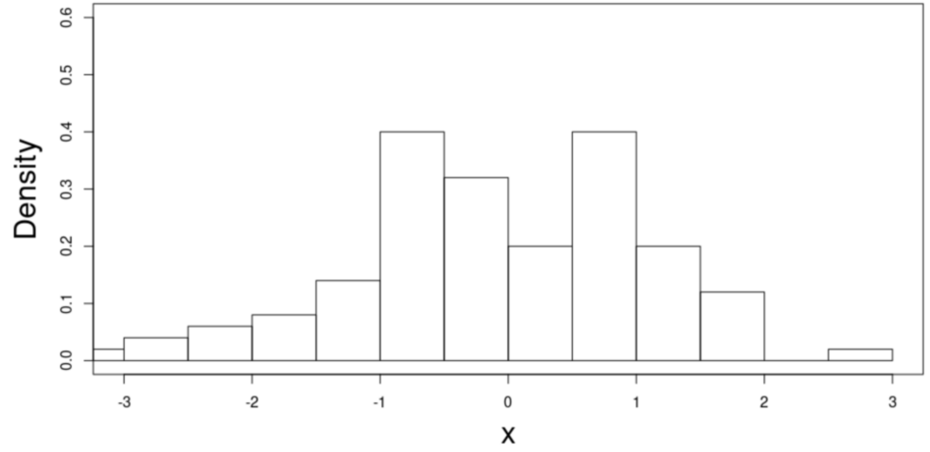
Human eyes like to look for patterns and trends. Be careful, and don't mistake randomness for a signal











Sampling from a distribution

- Sometimes you want to generate mock data that looks "real", and that follows some distribution

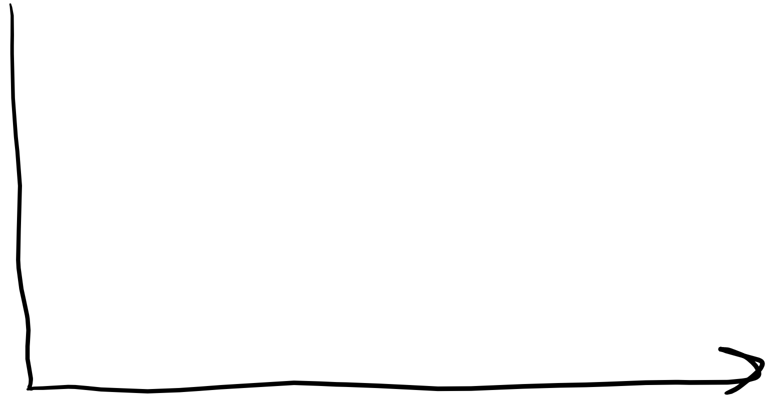
- Statistical software programs and packages (e.g., in R, Python, Julia, etc.) have built in functions to draw random values from a distribution.

- Sometimes, however, the distribution you want to draw from isn't coded up already, so you need to do it yourself

Sampling from a distribution (two basic approaches)

Inverse cdf Method

- First choice if the inverse cdf is tractable



Accept/Reject Algorithm

- Useful when you can't write down the inverse cdf

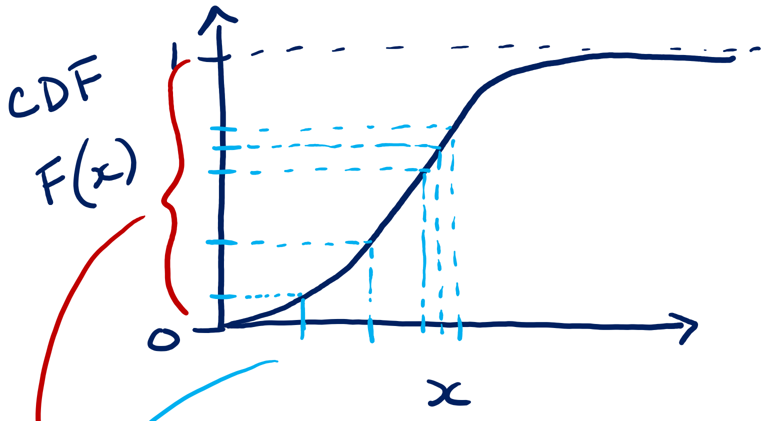
Sampling from a distribution (two basic approaches)

Inverse cdf Method

inverse CDF

$\boxed{F^{-1}(x)}$

- First choice if the inverse cdf is tractable



draw random values between 0 and 1

drawing from $U(0,1)$, and pass to $F^{-1}(x)$

uniform distribution

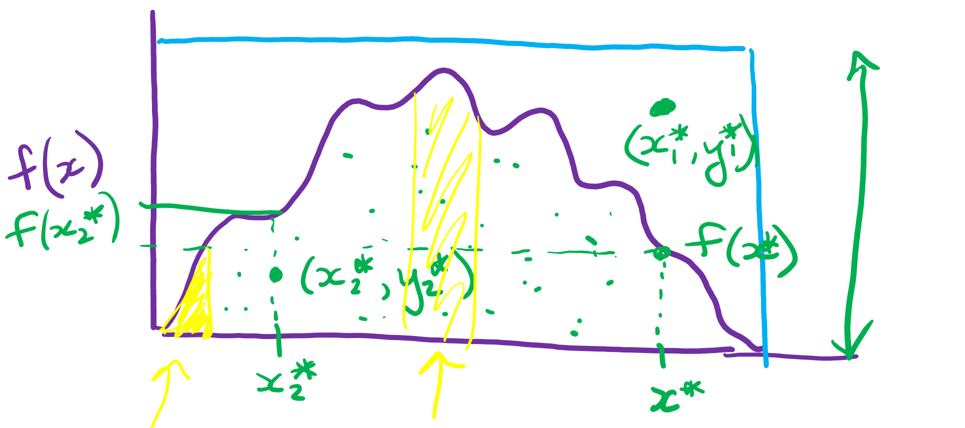
a bunch of x values that have $f(x)$ → Pareto distribution

α

x_{min}

Accept/Reject Algorithm

- Useful when you can't write down the inverse cdf



$y_1^*, f(x^*)$

$y_2^*, f(x_2^*)$

Estimates of Distributions

Visualizing Empirical Distributions

- Histograms (in frequency or relative frequency)
- Boxplots
- Kernel Density Estimators
- Empirical cumulative distribution functions (ecdfs)
- Bar charts, stacked bar chart, mosaic plots, contingency tables, ...

Visualizing Empirical Distributions

- Histograms (in frequency or relative frequency)
- Boxplots
- Kernel Density Estimators
- Empirical cumulative distribution functions (ecdfs)
- Bar charts, stacked bar chart, mosaic plots, contingency tables, ...

Estimating Parameters of Distributions

- Method of moments
- Maximum Likelihood Estimators
- Bayesian inference

Box Plots

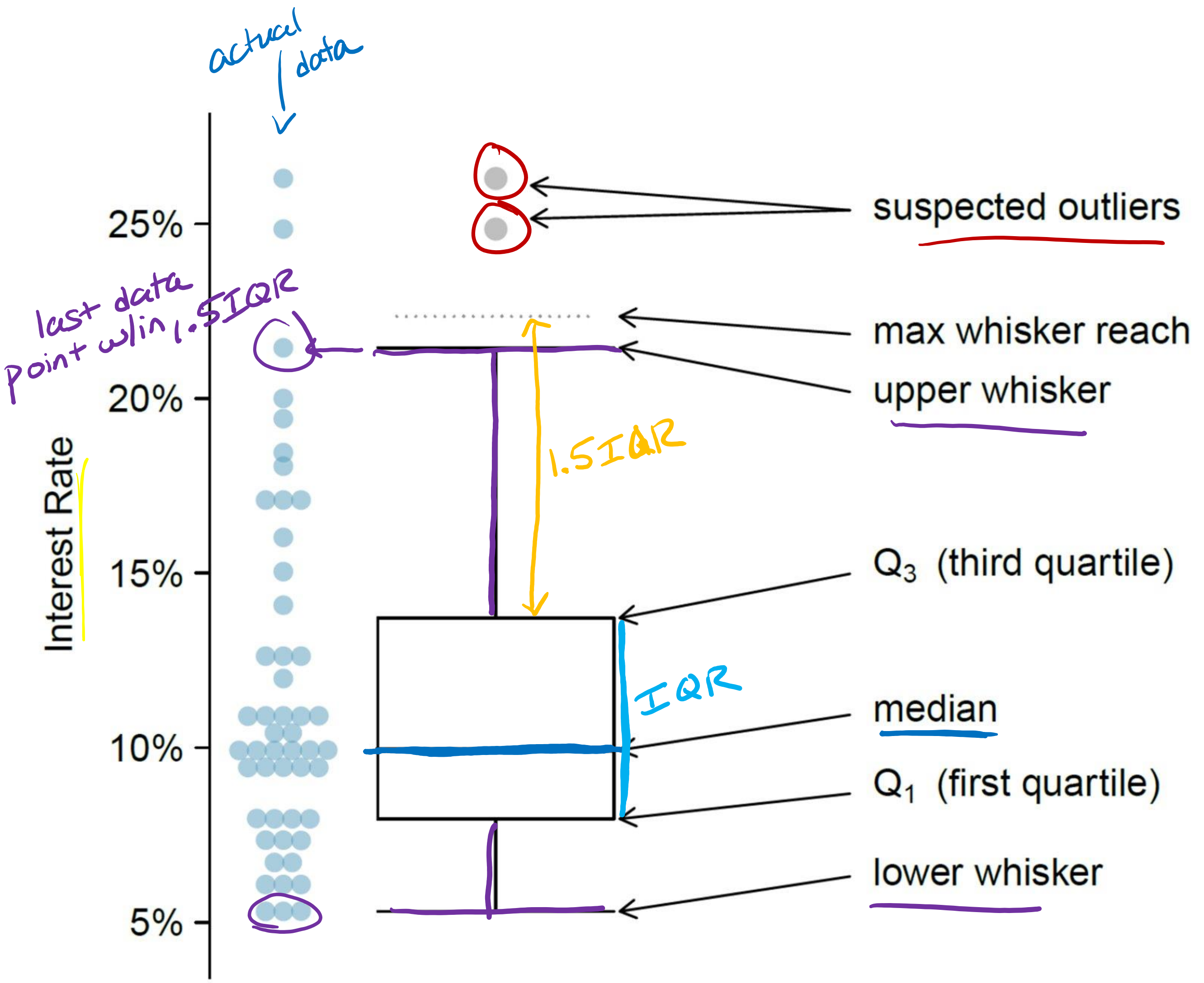
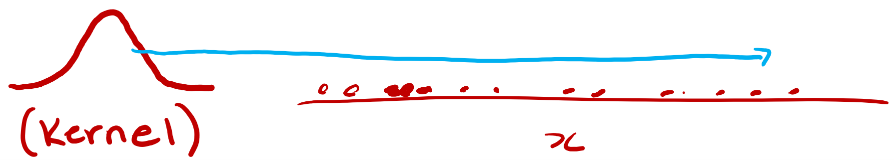


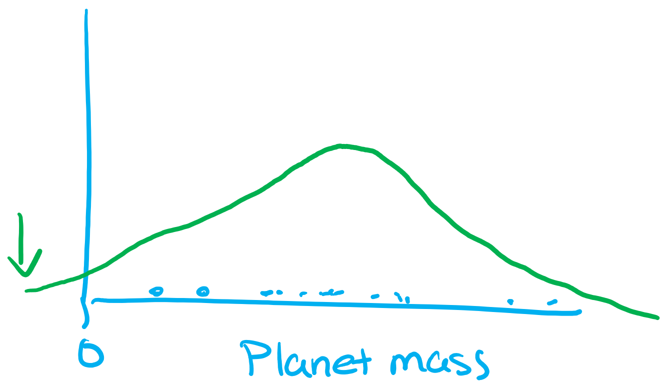
Figure 2.10, OpenIntro (4th ed.)

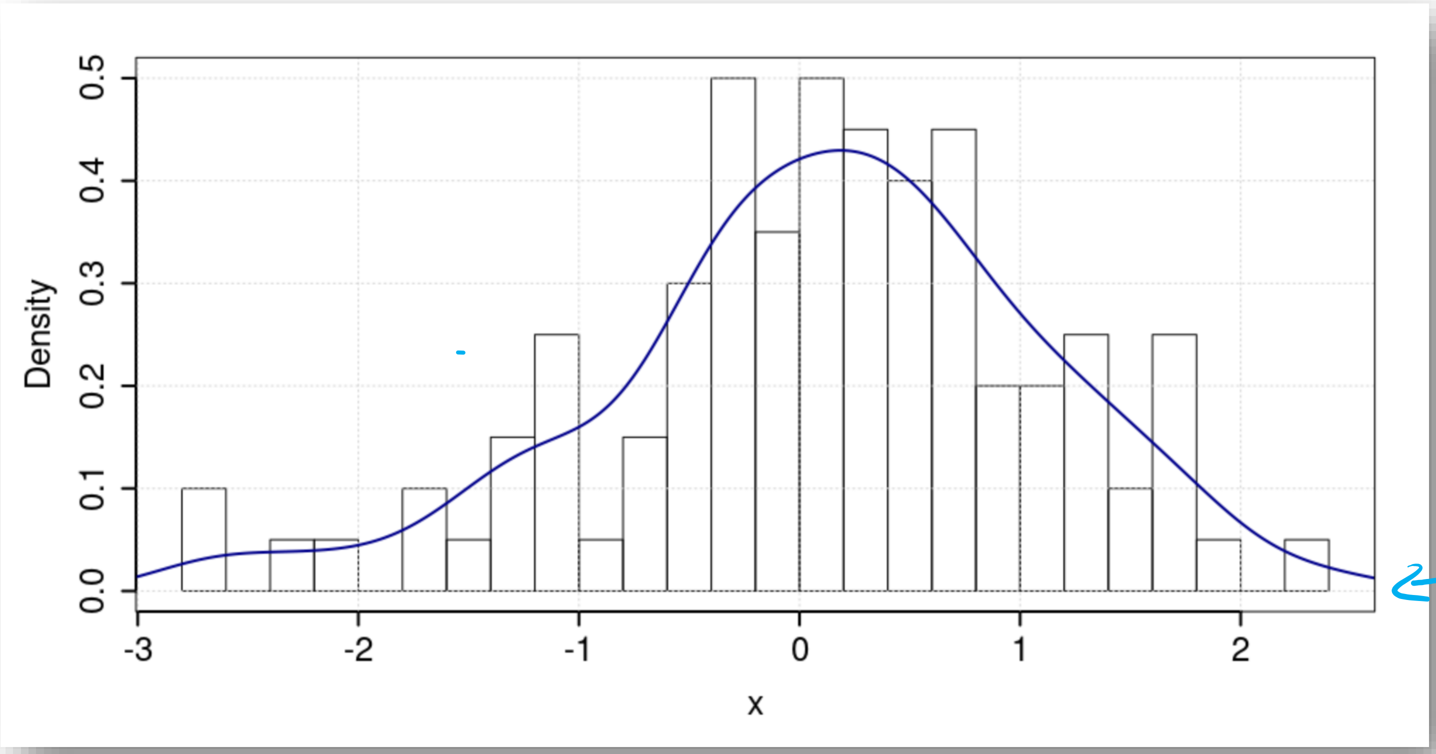
- Building a box plot:
  - Find the median first
  - Draw a rectangle that shows the interquartile range (IQR) → contains 50% of data
  - Extend the whiskers out to the furthest data point that is still within 1.5xIQR
  - Show the individual points that are outside the whiskers

Kernel Density Estimates

- Less sensitive to bin size, bin choice



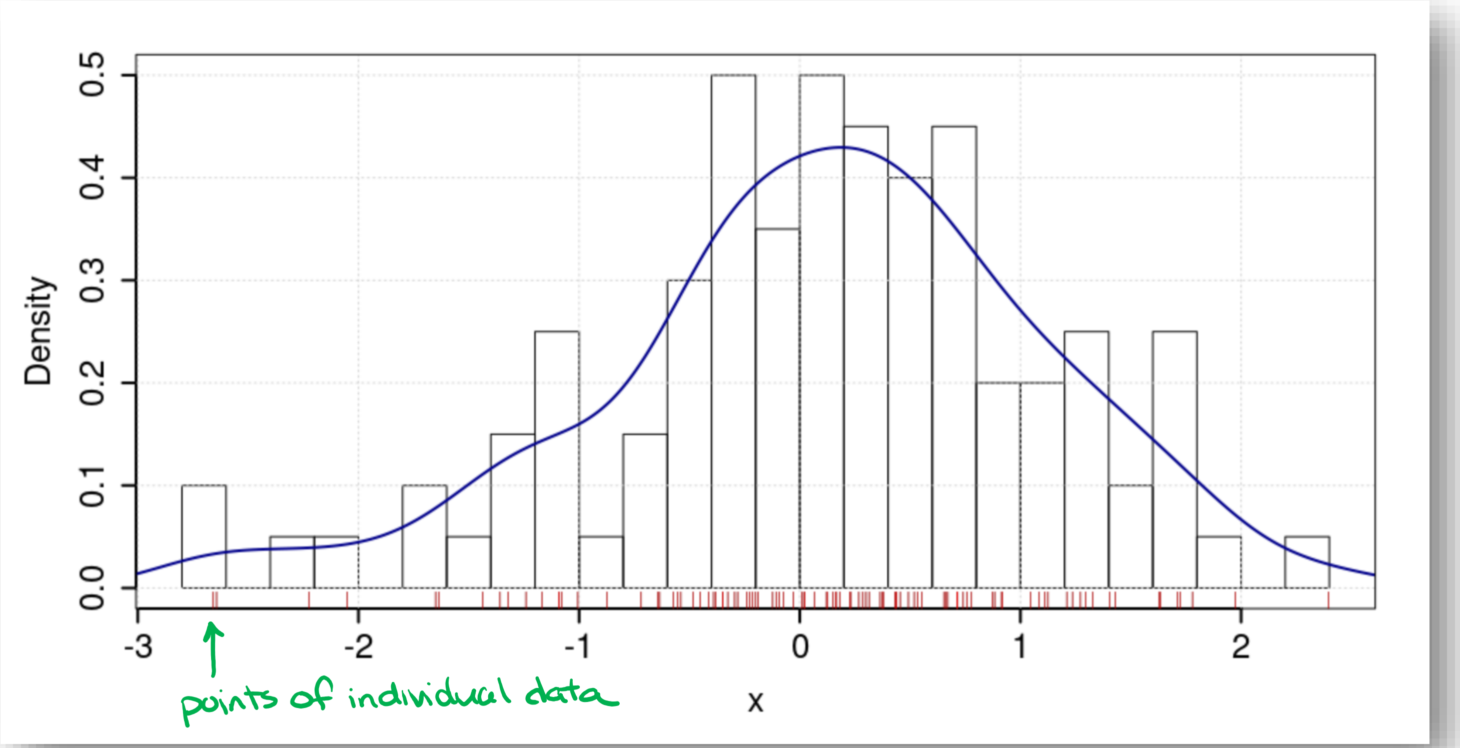




smooth estimate of distribution

Kernel Density Estimates

- Less sensitive to bin size, bin choice
- Helpful to add a "rug"



Kernel Density Estimates

- Less sensitive to bin size, bin choice
- Helpful to add a "rug"
- (really quick to plot in R)

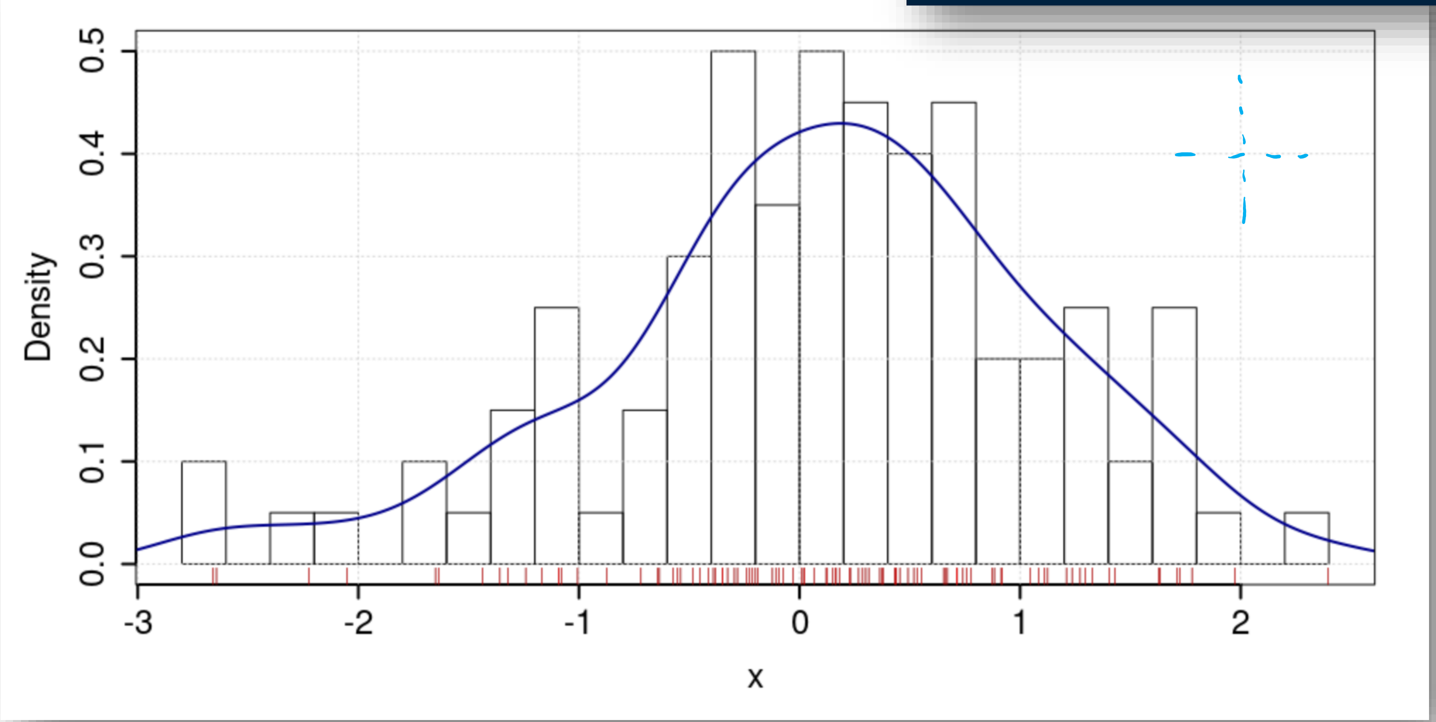
plots the KDE estimate

```
par(mar=c(5,5,2,2))
hist(x, breaks = 20, freq = FALSE, lwd=2, main="", cex.lab=1.5, cex.axis=1.5)
grid()
lines(density(x), col="darkblue", lwd=2)
box()
rug(x, col="red")
```

histogram

background grid

calculates KDE



NOTE:

You do not have to import any modules or packages to make these kinds of plots in R. The functions are just there already.

Summary Statistics

Five-Number Summary

You almost get all five from a boxplot.

- Minimum
- 1st quartile
- Median
- 3rd quartile
- Maximum

Five-Number Summary

You almost get all five from a boxplot.

- Minimum
- 1st quartile
- Median
- 3rd quartile
- Maximum

Fivenum function is in base R

```
> moons <- c(0, 0, 1, 2, 63, 61, 27, 13)
 > fivenum(moons)
[1]  0.0  0.5  7.5 44.0 63.0
```

Examples above from:
https://en.wikipedia.org/wiki/Five-number_summary

Five-Number Summary

You almost get all five from a boxplot.

- Minimum
- 1st quartile
- Median
- 3rd quartile
- Maximum

Fivenum function is in base R

```
> moons <- c(0, 0, 1, 2, 63, 61, 27, 13)
 > fivenum(moons)
[1]  0.0  0.5  7.5 44.0 63.0
```

Fivenum function must be written in Python

```python
import numpy as np

def fivenum(data):
    """Five-number summary."""
    return np.percentile(data, [0, 25, 50, 75, 100], interpolation='midpoint')

moons = [0, 0, 1, 2, 63, 61, 27, 13]
print(fivenum(moons))
[  0.    0.5   7.5  44.    63. ]
```

Examples above from:
https://en.wikipedia.org/wiki/Five-number_summary

Five-Number Summary

You almost get all five from a boxplot.

- Minimum
- 1st quartile
- Median
- 3rd quartile
- Maximum

Fivenum function is in base R

```
> moons <- c(0, 0, 1, 2, 63, 61, 27, 13)
 > fivenum(moons)
[1]  0.0  0.5  7.5 44.0 63.0
```

Fivenum function must be written in Python

```
import numpy

def fivenum(c
    """Five-r
    return np

moons = [0, 0
print(fivenum
[  0.     0.5
```

Want a quick way to see distribution details in python? Pip install "knowyourdata"

```
In [3]:  kyd(x)
```

Basic Statistics

Mean: -0.0007368   Std Dev: 0.9813

Min:	-3.161	-99% CI:	-2.53
1Q:	-0.6406	-95% CI:	-1.905
Median:	0.01018	-68% CI:	-0.9957
3Q:	0.6503	+68% CI:	0.9445
Max:	3.651	+95% CI:	1.938
		+99% CI:	2.493

Array Structure

Number of Dimensions:	1
Shape of Dimensions:	(2000,)
Array Data Type:	float64
Memory Size:	15.7KiB
Number of NaN:	0
Number of Inf:	0

Five-Number Summary

You almost get all five from a boxplot.

- Minimum
- 1st quartile
- Median
- 3rd quartile
- Maximum

Summary Statistics

Includes the mean too:

- Minimum
- 1st quartile
- Median
- Mean
- 3rd quartile
- Maximum

summary() behaves differently depending on class of object

```
> fivenum(x)
[1] -2.6609228 -0.4021807  0.1650809  0.7280392  2.3974525
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.6609 -0.3964  0.1651  0.1059  0.7216  2.3975
```

Pandas in Python

Exercises to try

Exercise 1

1. Plot the PDF for the χ^2 distribution, for different values of the degrees of freedom (N) of that distribution.
2. Compare this to the normal distribution.
3. What do you notice about the two distributions?
4. Now compare the normal distribution to the log normal distribution for a range of values of the mean and variance. How do the mean and variance of the log normal distribution map onto the mean, variance of the normal distribution?

COOL CATCH

Remember that R, Python, etc. have distributions coded up – don't reinvent the wheel!

Exercise 2

- Use the *accept-reject* approach to transform numbers generated from a uniform distribution into those following the distribution $P(x) = (1/(e-1))\exp(x)$ for $0<x<1$ and 0 elsewhere
    - Draw two random samples x^*, y^* from the U(0,1) distribution
    - If $y^* < c\,f(x^*)$, keep x^* [remember the normalization *c* here]
    - If not, draw another two random samples from the distribution
    - Continue until you have 100 samples
    - Histogram the samples and over plot the PDF

- Use *CDF sampling* to do the same thing above.
    - To do this, compute the CDF F(X) by integrating the PDF P(x) from $-\infty$ to X
    - Then find the inverse $F^{-1}(X)$ of the CDF.
      [HINT: Remember an inverse function $F^{-1}(x)$ is such that $F(F^{-1}(x)) = x$]
    - Draw a random samples x_1 from the U(0,1) distribution
    - Then the variable $y = F^{-1}(x_1)$ will have the probability distribution you seek
    - Continue until you have 100 samples
    - Histogram the samples and over plot the PDF