Neurotechnology, Brains and Machines: Week 2

Sam Michalka

Some of this should be attributed to *Statistics in MATLAB: A Primer* by MoonJung Cho and Wendy L. Martinez. Chapter 4 should be a reference for this notebook.

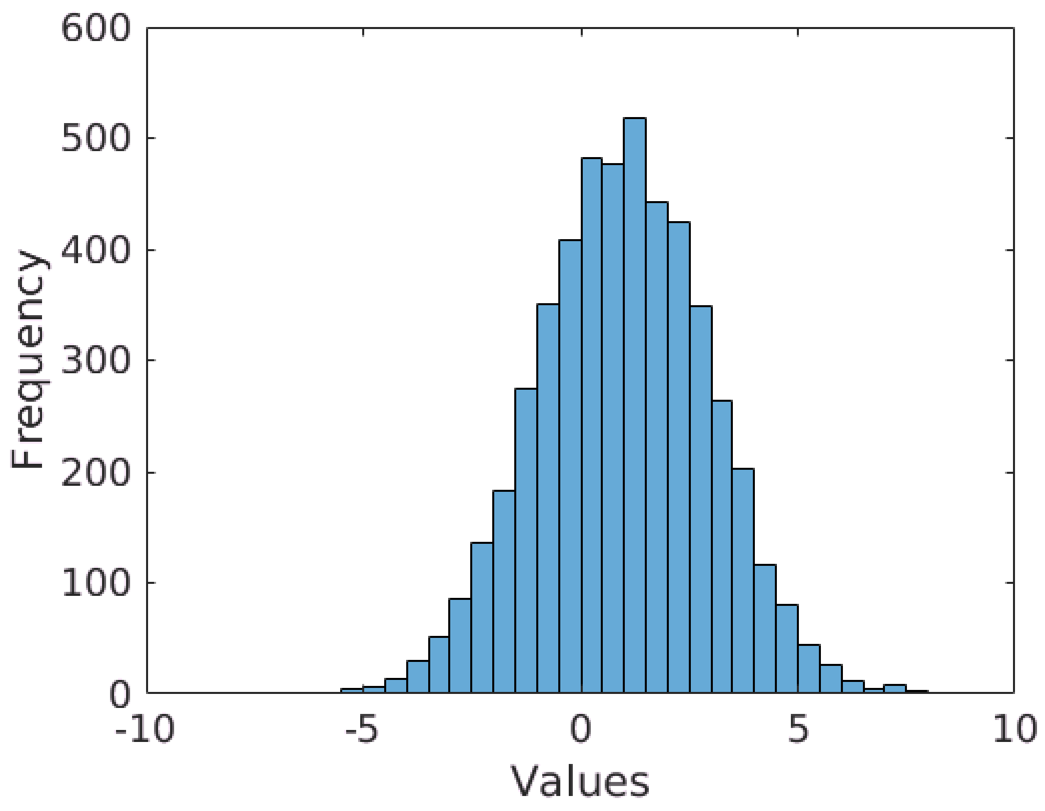
This notebook is not a stand-alone information source. You will need to seek out external resources.

Distributions and Plots

MATLAB allows us to create a distribution object, which we can then sample from to create a vector of random numbers. This probability distribution class is very useful... more on this later. For now, we'll just focus on normal distributions.

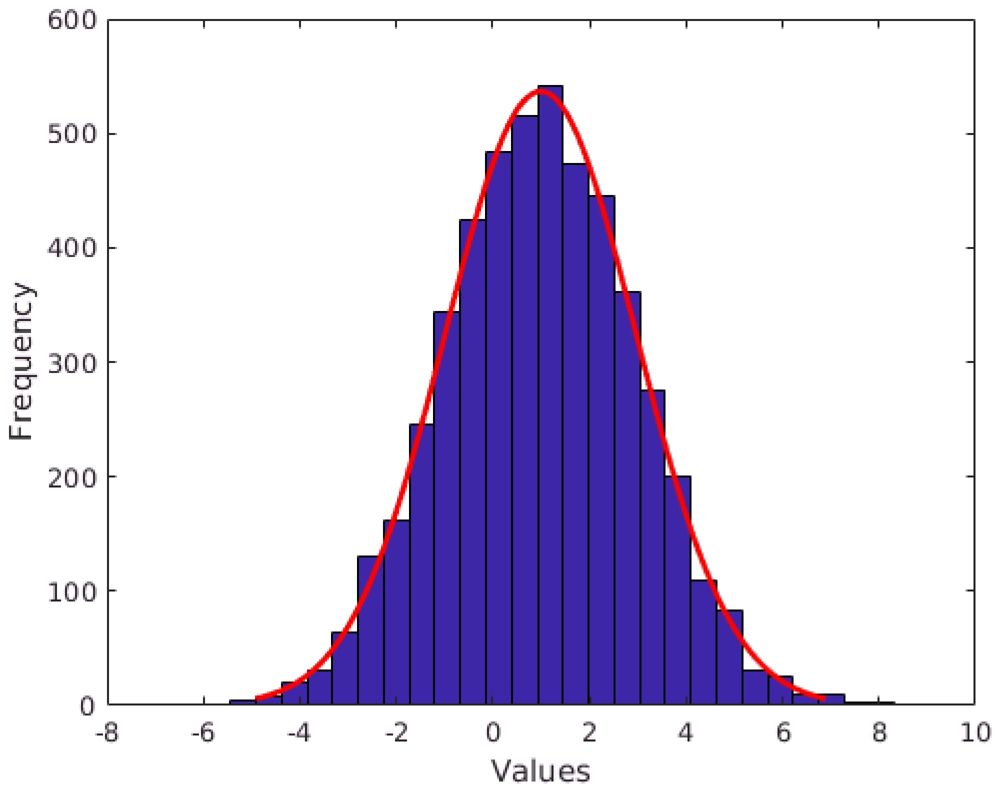
Exercise: Manipulate the value of n and numbins to get a sense of these two factors.

```
distobj1 = makedist('Normal','mu',1,'sigma',2);
n = 5000; %Number of data points in sample
yval = distobj1.random(n,1); %Select a random sample from the distribution
histogram(yval); %If you don't set the number of bins, matlab chooses.
xlabel('Values'); ylabel('Frequency');
set(gca,'FontSize',14); % Set font size to make things prettier
```



```
% You can use histfit to create a histogram with a line fit to your distribution.
```

```
numbins = 30; %Number of bins in histogram
histfit(yval,numbins,'Normal'); xlabel('Values'); ylabel('Frequency');
```



```
56
```

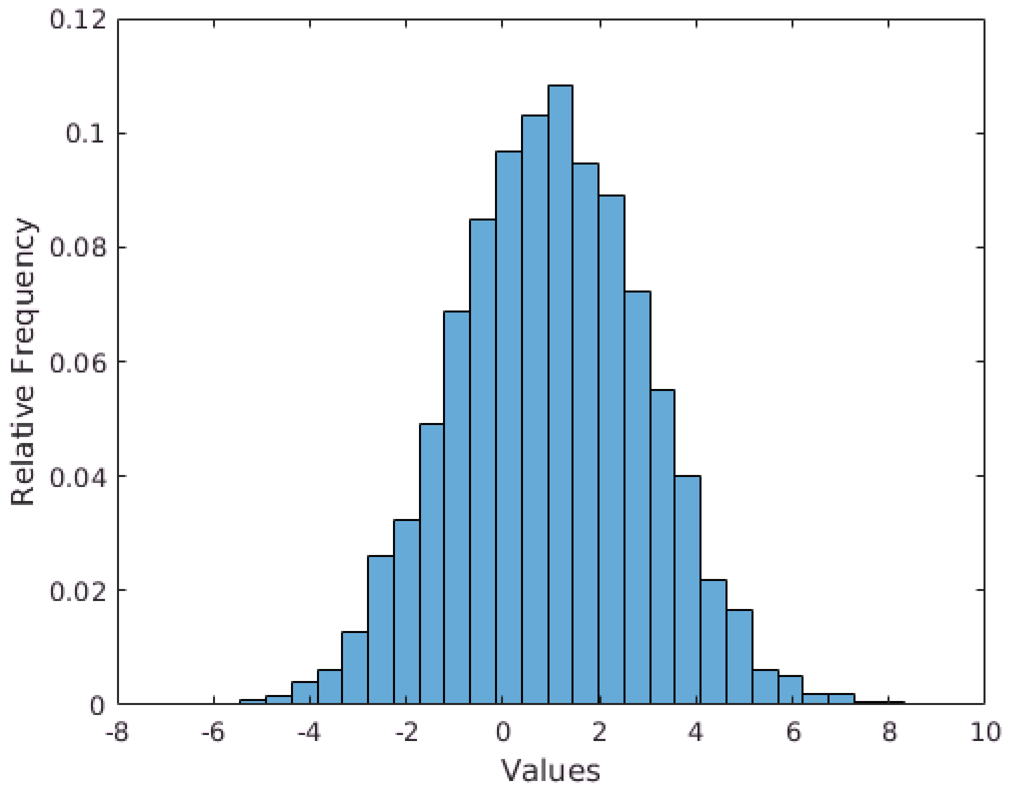
```
ans = 56
```

Exercise: Why might we want to represent our data as a smooth distribution?

Representing the data as a smooth distribution makes it easier to reason about intuitively and mathematically, and reduces the amount of uncertainty when looking at a region containing a small part of the data. For example, asking "what fraction of values are greater than 10" is less usefully answered by "none so far" than by "we expect 0.00001%".

We can use the histogram function to plot the relative probability instead of the actual frequency.

```
histogram(yval,numbins,'Normalization','probability');
xlabel('Values'); ylabel('Relative Frequency');
```



Probability density function (pdf)

We can use a probability density function to represent the shape of the distribution.

The result of evaluating a PDF for a particular value of x is not a probability; it is a probability density. The probability density measure probability per unit of x, so to get a probability mass, you have to integrate over x.

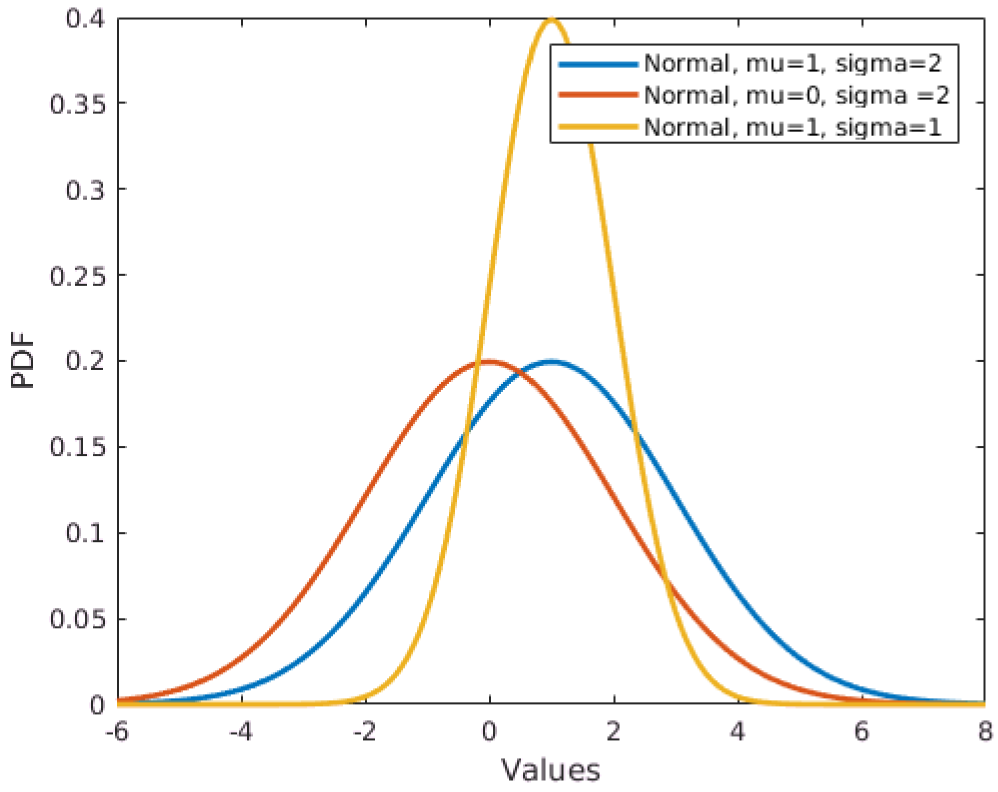
$$\mathrm{PDF}_{normal}(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

The probability density functions can be useful for comparing different distributions.

```
X = -6:.1:8; %Set some values of x to evaluate at
distobj1 = makedist('Normal','mu',1,'sigma',2);
plot(X,distobj1.pdf(X),'LineWidth',2);
```

```
hold on;
distobj2 = makedist('Normal','mu',0,'sigma',2);
plot(X,distobj2.pdf(X),'LineWidth',2);

distobj3 = makedist('Normal','mu',1,'sigma',1);
```

```
plot(X,distobj3.pdf(X),'LineWidth',2);
xlabel('Values'); ylabel('PDF');
legend({'Normal, mu=1, sigma=2','Normal, mu=0, sigma =2','Normal, mu=1, sigma=1'});
hold off;
```



Cumulative Distribution Function (cdf)

Like the pdf, the cdf can be called as a method for the distribution object.

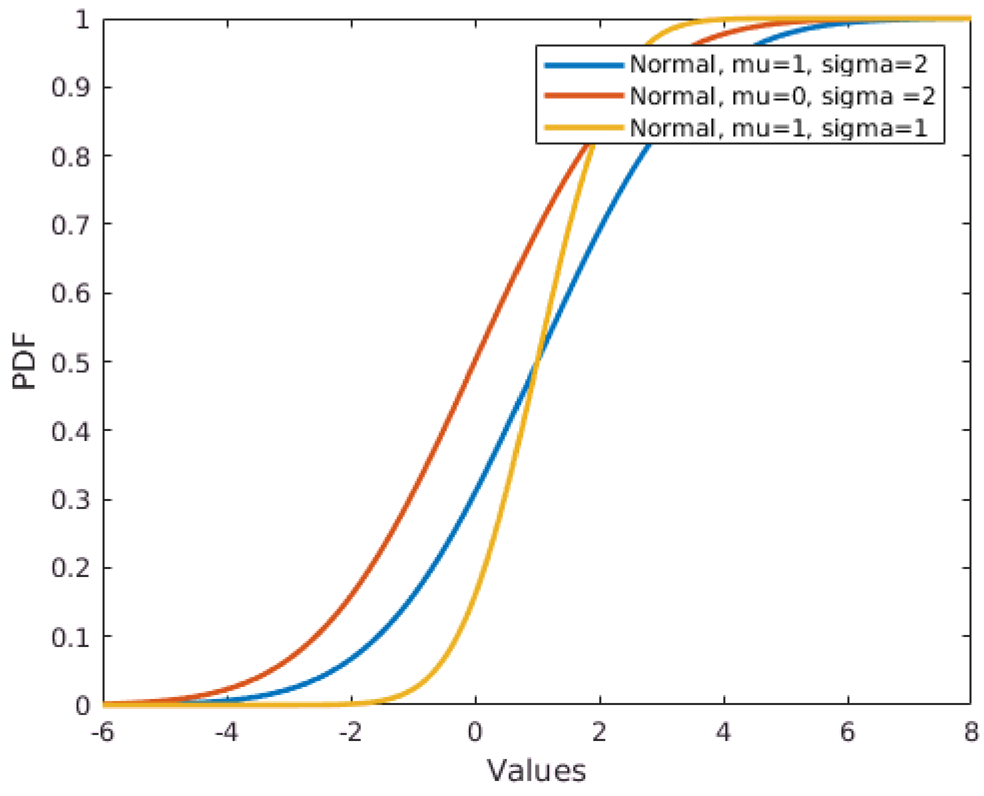
You can see all available methods for an object by calling methods()

```
methods(distobj1)
```

```
Methods for class prob.NormalDistribution:

cat        icdf       median     pdf        std        vertcat
cdf        iqr        negloglik  proflik    truncate
horzcat    mean       paramci    random     var
```

Exercise: Plot and label the cumulative distribution functions for the 3 distributions above.

```
plot(X,distobj1.cdf(X),'LineWidth',2);
hold on;
plot(X,distobj2.cdf(X),'LineWidth',2);
plot(X,distobj3.cdf(X),'LineWidth',2);
xlabel('Values'); ylabel('PDF');
legend({'Normal, mu=1, sigma=2','Normal, mu=0, sigma =2','Normal, mu=1, sigma=1'});
hold off;
```



Exercise: Describe the effects of the mean and standard deviation on the cdf.

The mean (mu) determines the value it which the PDF intersects 0.5. The standard deviation affects the slope of the crossing, with higher sigma creating a less steep crossing.

Now let's return to a single distribution, distobj1 (mu = 1, sigma =2). The methods for cdf and pdf take in numerical values and output the probability distribution value at the given numerical value. You can confirm that these values at mean =1 correspond to the pdf and cdf plots.

```
distobj1.pdf(1)
```

```
ans = 0.1995
```

```
distobj1.cdf(1)
```

```
ans = 0.5000
```

```
distobj1.cdf(distobj1.mu)  % just another way of writing the line directly before this one
```

```
ans = 0.5000
```

Exercise: Look up the area under the curve of a normal distribution. Remember that the cdf outputs the area under the pdf for values up to and including the input value. Determine/Justify what value you should expect the cdf to output if the input value is 1 standard deviation below the mean. Then confirm by calling the cdf method.

```
(1-0.68)/2
```

```
ans = 0.1600
```

68% of the samples should fall within 1 standard deviation of the mean. That means 16% should fall to each side of those lines.

```
% Answer code (and text) goes here (or below)
distobj1.cdf(distobj1.mu-distobj1.sigma)
```

```
ans = 0.1587
```

Close enough.

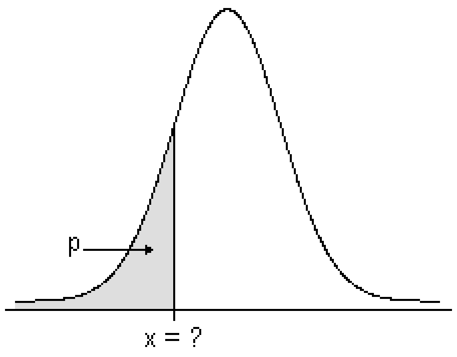
Exercise: Look up the area under the curve of a normal distribution. Determine/Justify what value you should expect the cdf to output if the input value is 1.96 standard deviations above the mean. Then confirm by calling the cdf method.

95% of the density falls within +-1.96mu, so 2.5% falls on each side. That means 0.975 falls before that line.

```
% Answer code (and text) goes here (or below)
distobj1.cdf(distobj1.mu+1.96*distobj1.sigma)
```

```
ans = 0.9750
```

The inverse cumulative distribution function takes in a proportion (betwen 0 and 1), and outputs a value x such that the probability of X ≤ x is greater than or equal to p. This is illustrated as the area under the pdf between negative infinity and x. Remember that the cdf is the intergral of the pdf.



Let's think about an example. We know that distobj1 contains a normal distribution with a mean of 1 and a standard deviation of 2. When x = the mean = 1, we expect p = 0.5. The area under the pdf curve to the left of x = 1 should be half (0.5) of the the total area under the curve (which is 1). We can confirm this with the icdf method.

```
p = 0.5; %proportion of total area under the pdf curve for X<=x OR y-value of the cdf at x.
distobj1.icdf(p)
```

```
ans = 1
```

```
distobj1.mu %This is the mean
```

```
ans = 1
```

Exercise: Confirm your answer for the previous exercise (case of 1.96 stdevs above the mean).

```
% Answer code goes here

distobj1.icdf(0.975)
```

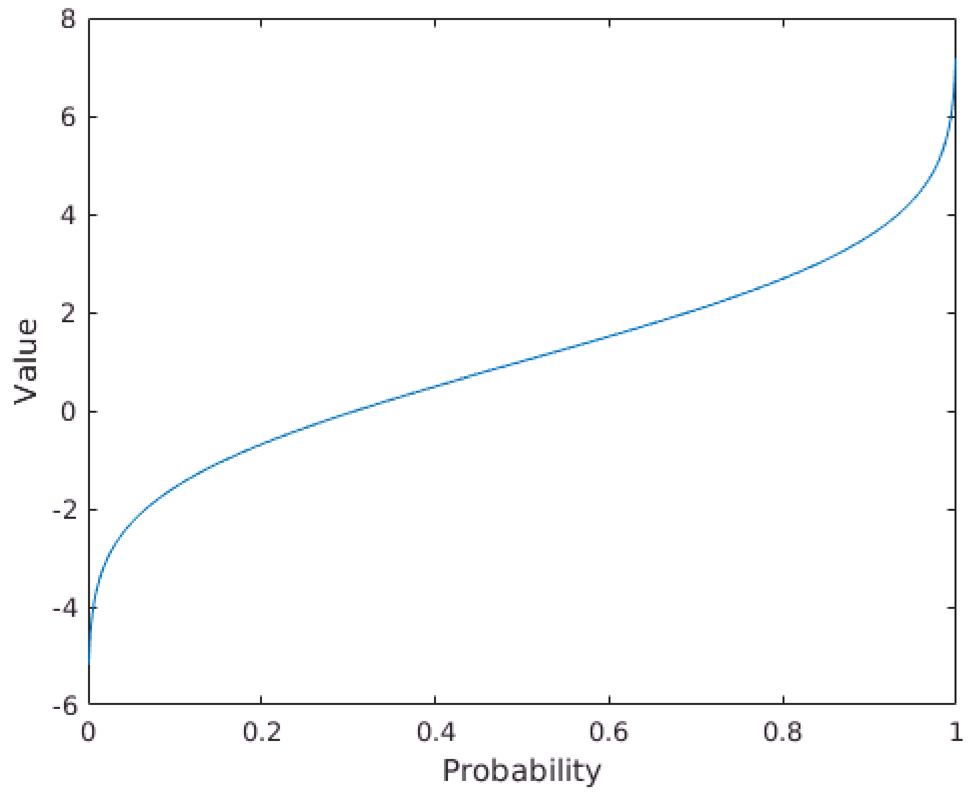
```
ans = 4.9199
```

```
(ans-distobj1.mu)/distobj1.sigma
```

```
ans = 1.9600
```

Exercise: Plot/Label the inverse cumulative distribution function. Consider what values are appropriate for the x-axis.

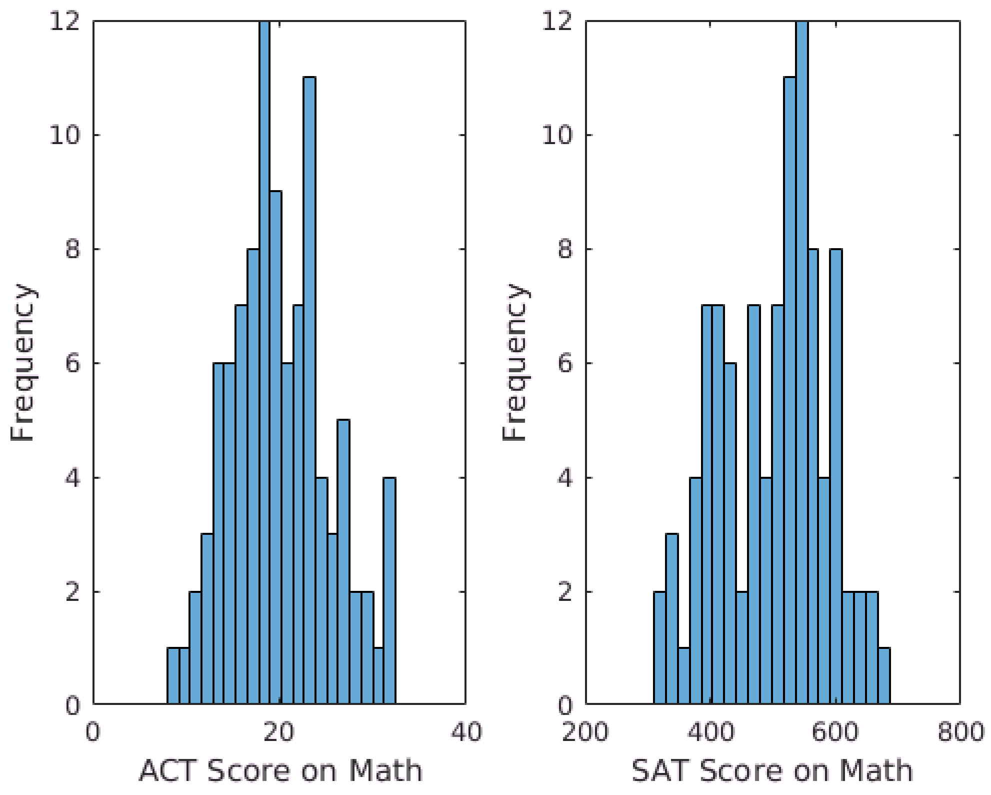
```
% Answer code goes here.
r = 0:0.001:1;
plot(r, distobj1.icdf(r))
xlabel("Probability")
ylabel("Value")
```



Standardized normal distribution

The standardized normal distribution (or z-distribution) is a normal distribution, where the mean is 0 and the standard deviation is 1. Converting our measurements into a z-score can be useful for comparing information from two scales. Regardless of your thoughts on standardized testing (e.g., the SATs and the ACTs), they are great for examples of comparing scores from different distributions.

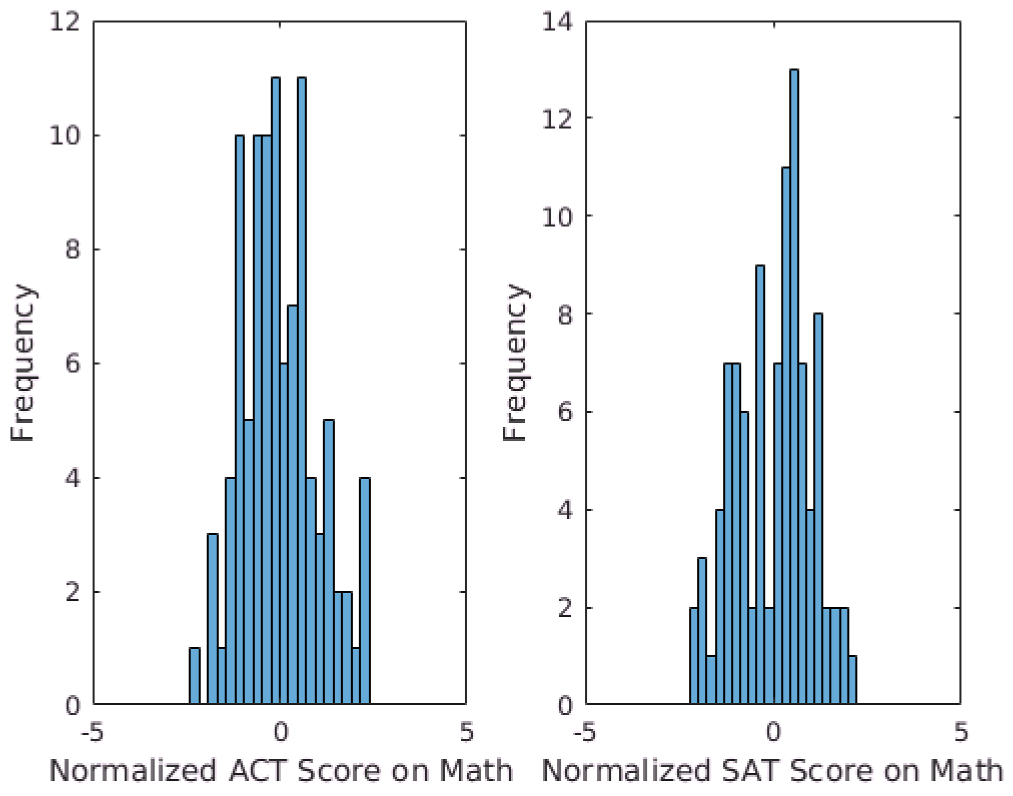
```
avals = normrnd(20.7,5.4,100,1); %Random values from ACT-like distribution (mean, stdev, num m
svals = normrnd(500,100,100,1); %Random values from SAT-like distribution (mean, stdev, num me
subplot(1,2,1); histogram(avals,20); xlabel('ACT Score on Math'); ylabel('Frequency')
subplot(1,2,2); histogram(svals,20); xlabel('SAT Score on Math'); ylabel('Frequency')
```



Exercise: Convert each of the test scores to z-scores and plot them in their own subplot (just like above).

I chose to use the means and std. dev's from the actual data here, not the generating function above.

```
azvals = (avals-mean(avals))/std(avals);
szvals = (svals-mean(svals))/std(svals);
subplot(1,2,1); histogram(azvals,20); xlabel('Normalized ACT Score on Math'); ylabel('Frequenc
subplot(1,2,2); histogram(szvals,20); xlabel('Normalized SAT Score on Math'); ylabel('Frequenc
```



Exercise: One ambitious student has decided to take the ACT and the SAT (just focusing on math). Now they are deciding which scores they want to report to get that sweet sweet college scholarship money. They scored a 34 on the ACT and a 720 on the SAT. Use z-scores to determine which test they did "better" on. You can get the means and standard deviations in the code above.

```
%Answer code and text goes here
(34-20.7)/5.4
```

```
ans = 2.4630
```

```
(720-500)/100
```

```
ans = 2.2000
```

Because the ACT score is higher as a z-score, it is the more impressive number to send.

Effect sizes

Read questions/sections 1, 2, 3, 4, and 7 of this paper: http://www.cem.org/attachments/ebe/ESguide.pdf about effect sizes. We will talk more about statistical significance and hypothesis testing later, so don't worry about the details of those things (especially if this is new to you).

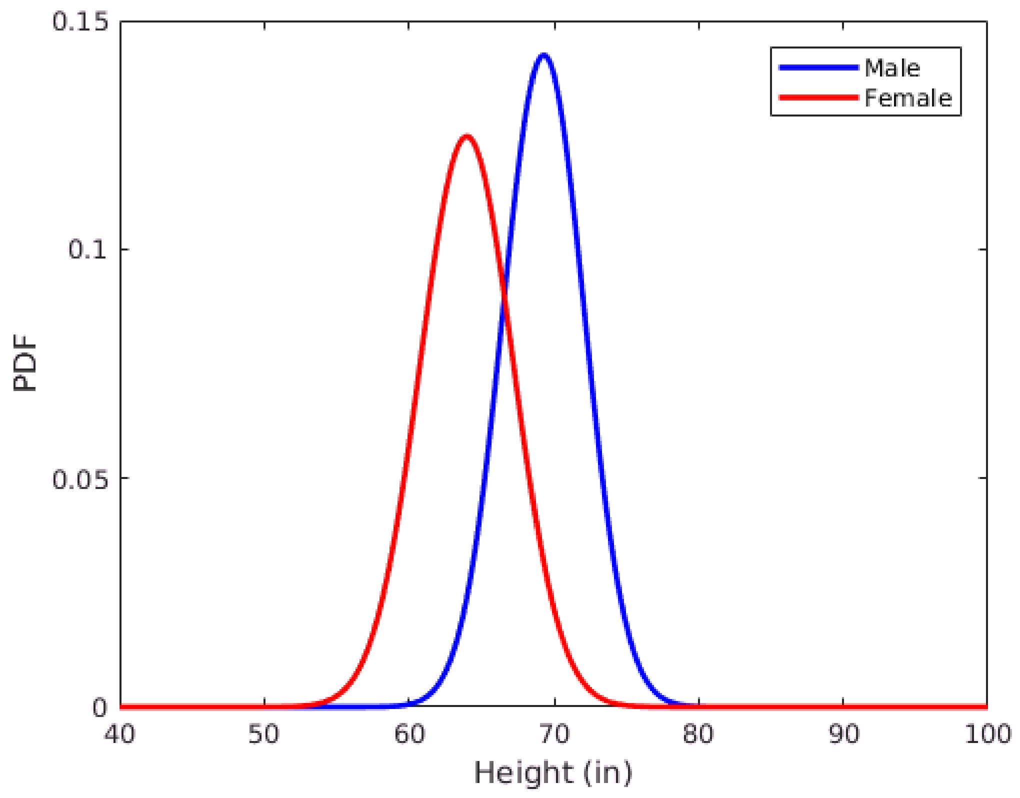
The following is an example looking at sex differences in person height and brain weight.

Reference: Witelson, S. F., Beresh, H., & Kigar, D. L. (2006). Intelligence and brain size in 100 postmortem brains: sex, lateralization Jarrett, Christian. Great Myths of the Brain (Great Myths of Psychology) (p. 125). Wiley. Kindle Edition.

Here, we'll only look at right-handed people (separated in study table). This is just a quick peek at a complex data set. There were many additional differences between these two groups in addition to their sex.

First, we will look at height, measured in inches. We will assume these are normal distributions.

```
% Right handed subjects from from Witelson et al.
nf = 38; % num female
nm = 20; % num male

% height in inches
hmeanf = 64; % height mean (females)
hmeanm = 69.3; % height mean (males)
hsdf = 3.2; %height stdev females
hsdm = 2.8; %height stdev males
hvals = 40:.2:100; % height range to look at (inches)
fheight = makedist('Normal','mu',hmeanf,'sigma',hsdf); % Normal distribution of right handed f
mheight = makedist('Normal','mu',hmeanm,'sigma',hsdm); % Normal distribution of right handed f

% Plot the distributions
subplot(1,1,1)
plot(hvals,mheight.pdf(hvals),'b','LineWidth',2)
hold on
plot(hvals,fheight.pdf(hvals),'r','LineWidth',2);
legend({'Male','Female'}); xlabel('Height (in)'); ylabel('PDF');
hold off;
```



Note that the two groups have different standard deviations, which affects the shape and height of the pdf. To calculate effect sizes, we need to use a standard deviation. In some cases, it might make sense to use the standard deviation for one group or the other, but usually we need to find the pooled standard deviation between our two groups. The equation can be found here:

$$SD_{pooled} = \sqrt{\frac{(N_E - 1)SD_E^2 + (N_C - 1)SD_C^2}{N_E + N_C - 2}}$$

where SD is standard deviation, N is number of participants, E and C are the two groups (classically referred to as experimental and control, but the naming doesn't apply in our case).

```
% Calculate the pooled standard deviation for males & females
heightsdpooled = sqrt(((nm-1)*(hsdm.^2) + (nf-1)*(hsdf.^2))./(nf+nm-2))
```

```
heightsdpooled = 3.0701
```

Note that the pooled standard deviation is slightly closer to the standard deviation for females, as there are more females in the study.

```
% Calculate the effect size
heightcohensd = (hmeanm-hmeanf)./heightsdpooled
```

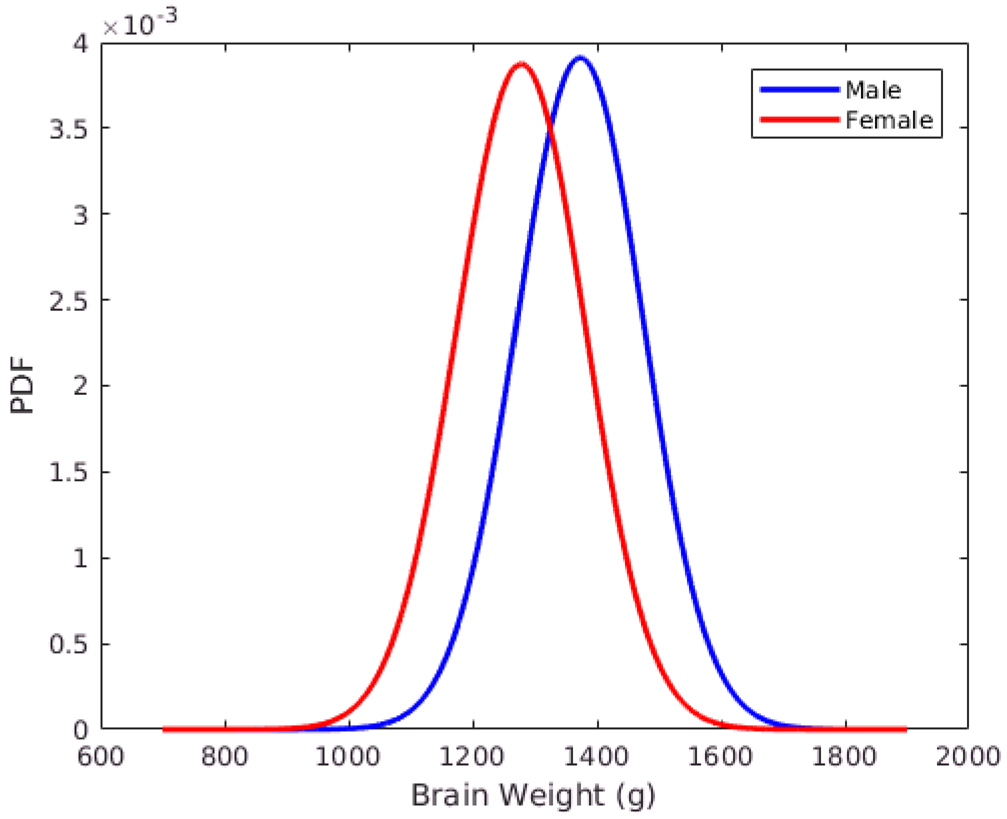
```
heightcohensd = 1.7263
```

Studies have consistently shown differences in brain mass between sexes, so we expect to find those differeneces here. Later on, we will read about how brain weight is not a good correlate of intelligence.

The numbers in this study represent data from a small to medium sized sample, so these will only give a rough estimate of the effect size. We are going to plot distributions and evaluate effect sizes around whole brain weight differences, though there are many ways to weigh a brain.

```
% Fresh whole brain weight
fbwmean = 1278; % mean female brain weight
mbwmean = 1373; % mean male brain weight
fbwsd = 103; %sd female
mbwsd = 102; % sd male
w = 700:1900; %Range of weights to look at
fbrainweight = makedist('Normal','mu',fbwmean,'sigma',fbwsd); % Normal distribution of right h
mbrainweight = makedist('Normal','mu',mbwmean,'sigma',mbwsd); % Normal distribution of right h

% Plot the distributions
plot(w,mbrainweight.pdf(w),'b','LineWidth',2)
hold on
plot(w,fbrainweight.pdf(w),'r','LineWidth',2);
```

```
legend({'Male','Female'}); xlabel('Brain Weight (g)'); ylabel('PDF');
hold off;
```



Exercise: Calculate the pooled standard deviation of brain weights for males & females

```
% Answer code goes here
brainsdpooled = sqrt(((nm-1)*(mbwsd.^2) + (nf-1)*(fbwsd.^2))./(nf+nm-2))
```

```
brainsdpooled = 102.6618
```

Exercise: Calculate the effect size for the differences in brain weight. Comment on how this compares to the effect size for height differences.

```
% Answer code goes here
(mbwmean-fbwmean)/brainsdpooled
```

```
ans = 0.9254
```

This effect size is smaller than that associated with height differences. I feel like a multivariate analysis is in order here.

Exploring distribution types

We spend a lot of time talking about normal distributions, but they are certainly not the only type of distribution. Table 4.1 in the Martinez & Cho Statistics in Matlab book lists several other continuous distributions that are easily accessible in Matlab.

One way to experiment with these distributions is to use disttool. This funciton will show something in the notebook (that you can't click on), but will also pop out a GUI to experiment with. If you like GUIs, play around with the values and dropdown menus. If you hate them, go ahead to the details below about how to script this.

```
disttool();
```

Just like with the normal distribution, we can create a distribution object for other distributions. Depending on the distribution, different parameters and methods may be available.

Let's use the exponential distribution as an example. From Table 4.1 (or other online matlab/stats references), we know that the exponential distribution takes in one parameter: mu (the mean). This is also a good resource: https://www.mathworks.com/help/stats/pdf.html

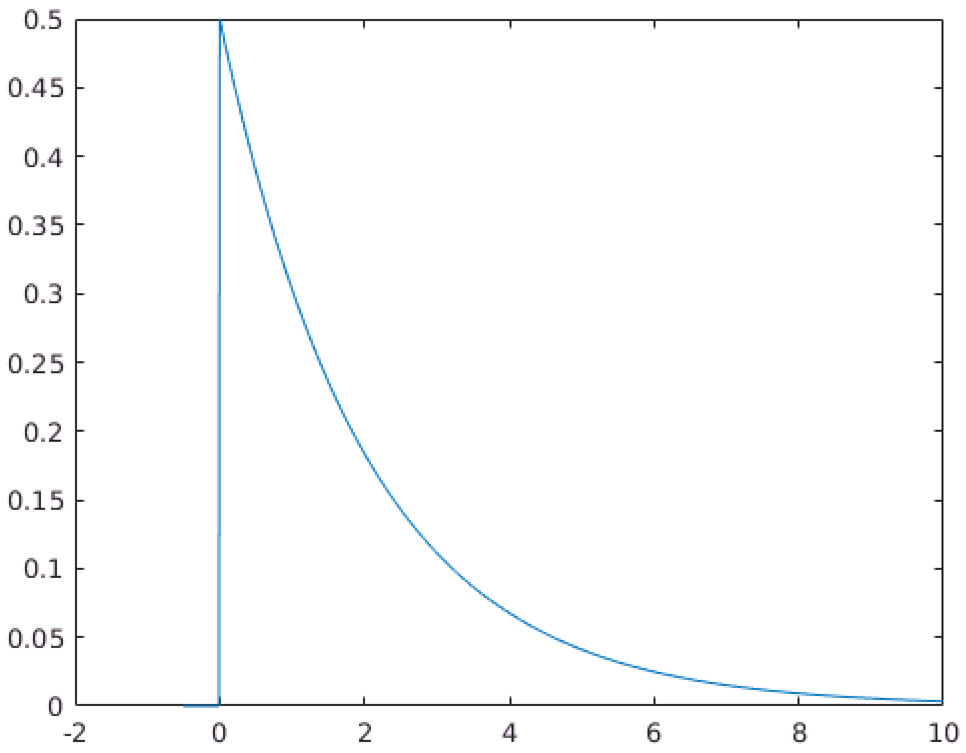
We will start by making a distribution object and choosing some values of x to evaluate over.

```
mydistobj = makedist('Exp','mu',2);
```

```
Warning: Error occurred while executing the listener callback for event POST_REGION defined
for class matlab.internal.editor.RegionEvaluator:
Struct contents reference from a non-struct array object.

Error in matlab.graphics.internal.getframeWithDecorations (line 23)
c = jf.getAxisComponent();

Error in matlab.internal.editor.FigureManager

Error in matlab.internal.editor.FigureManager.getCurrentFigurePixels

Error in matlab.internal.editor.FigureManager

Error in matlab.internal.editor.FigureManager

Error in matlab.internal.editor.FigureManager.getFigureSnapshots

Error in matlab.internal.editor.FigureManager.compareFigures

Error in matlab.internal.editor.OutputsManager/handleFigures

Error in matlab.internal.editor.OutputsManager/postRegionCallback

Error in
matlab.internal.editor.OutputsManager>@(varargin)obj.postRegionCallback(varargin{:})

Error in matlab.internal.editor.RegionEvaluator/doPostRegion

Error in matlab.internal.editor.RegionEvaluator/internalPreLineCallback

Error in
matlab.internal.editor.RegionEvaluator>@(varargin)obj.internalPreLineCallback(varargin{:})
```

```
Error in matlab.internal.editor.Evaluator/internalPreLineCallback

Error in
matlab.internal.editor.Evaluator>@(lineNumber,~)obj.internalPreLineCallback(lineNumber,callbackData)

Error in LiveEditorEvaluationHelperE1584257543 (line 111)
mydistobj = makedist('Exp','mu',2);

Error in matlab.internal.editor.Evaluator/evalFile

Error in matlab.internal.editor.Evaluator/evaluateCode

Error in matlab.internal.editor.RegionEvaluator/evaluateRegions

Error in matlab.internal.editor.evaluateRegions

Error in matlab.internal.editor.EvaluationOutputsService.evalRegions
```

```
X = -0.5:.01:10;
plot(X,mydistobj.pdf(X));
```



We can also learn more about our new object by calling the methods function.

```
methods(mydistobj)
```

```
Methods for class prob.ExponentialDistribution:

cat         icdf        median      pdf         std         vertcat
cdf         iqr         negloglik   proflik     truncate
horzcat     mean        paramci     random      var
```

We can also type mydistobject. and then hit the tab button to see all of our methods and value options. (make sure to include the period)

```
mydistobj.median
```

```
ans = 1.3863
```

```
mydistobj.var
```

```
ans = 4
```

We can select random variables from this distribution and assign them to a vector.

```
vals = mydistobj.random(6,1) % These values determine the vector size
```

```
vals =
    3.3131
    0.5901
    3.0943
    4.6027
    0.2188
    0.3912
```

Exercise:

1) Select a new continuous distribution type, research the relevant parameters and describe how they affect the shape of the distribution. (You may want to do some of the other steps before you write this part).

2) Plot the pdf and cdf for your distribution over some range of values that fully captures the distribution.

3) Create a large vector (at least 100 values) of values that have been randomly selected from your distribution. Plot these in a histogram.

4) Find an example of this distribution in the real world. You don't need to find data for this part, just describe a situation where this distribution is used.

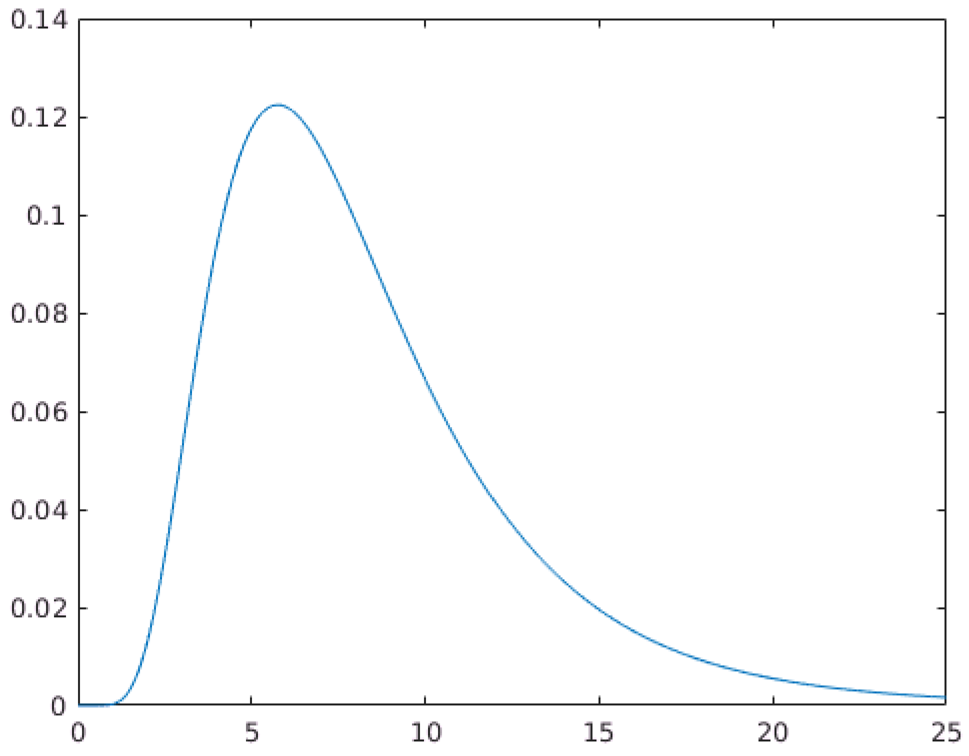
5) Repeat steps 1-4 for 2 other distributions.

The lognormal distribution

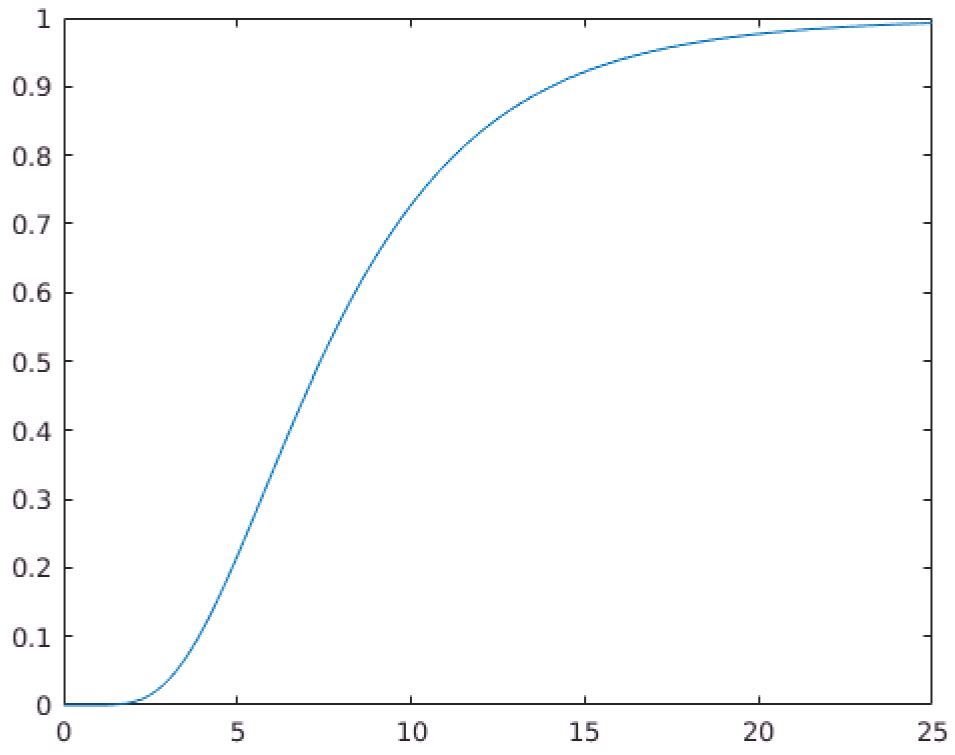
Similar to a normal distribution, mu controls the position of the mean, and sigma controls the width.

It is used to model the product of a bunch of positive random variables, and provides a useful representation of values that cannot be negative.

```
X = 0:0.01:25;
dist=makedist('Lognormal','mu',2,'sigma',.5);

plot(X, dist.pdf(X));
```

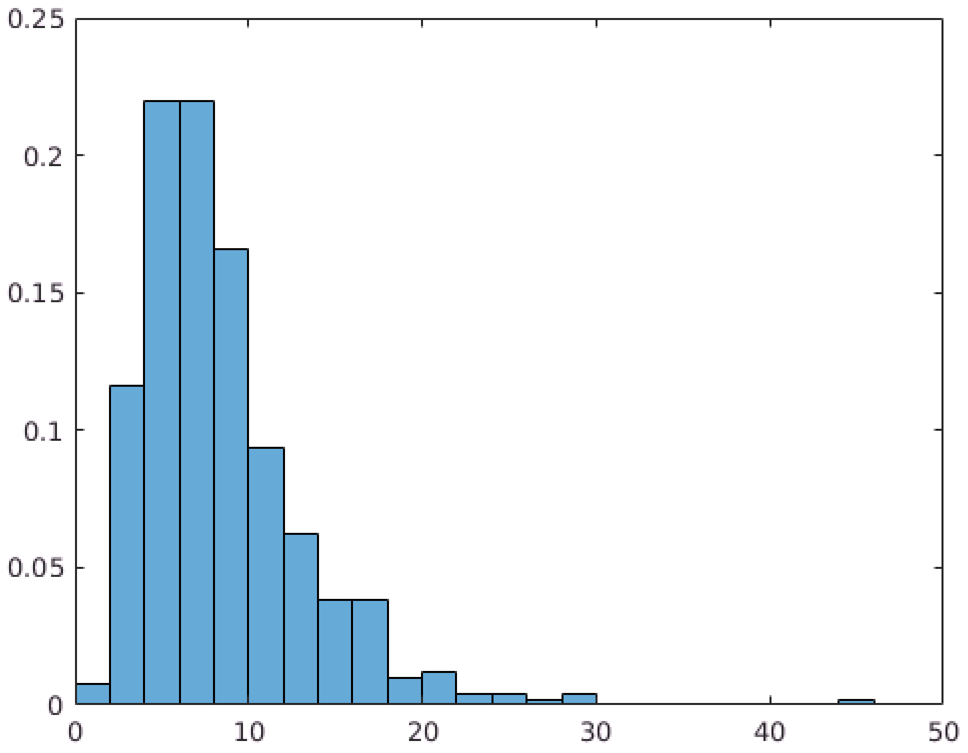


```
plot(X, dist.cdf(X));
```



```
data = dist.random(500,1);
```

```
histogram(data, 'Normalization','probability')
```

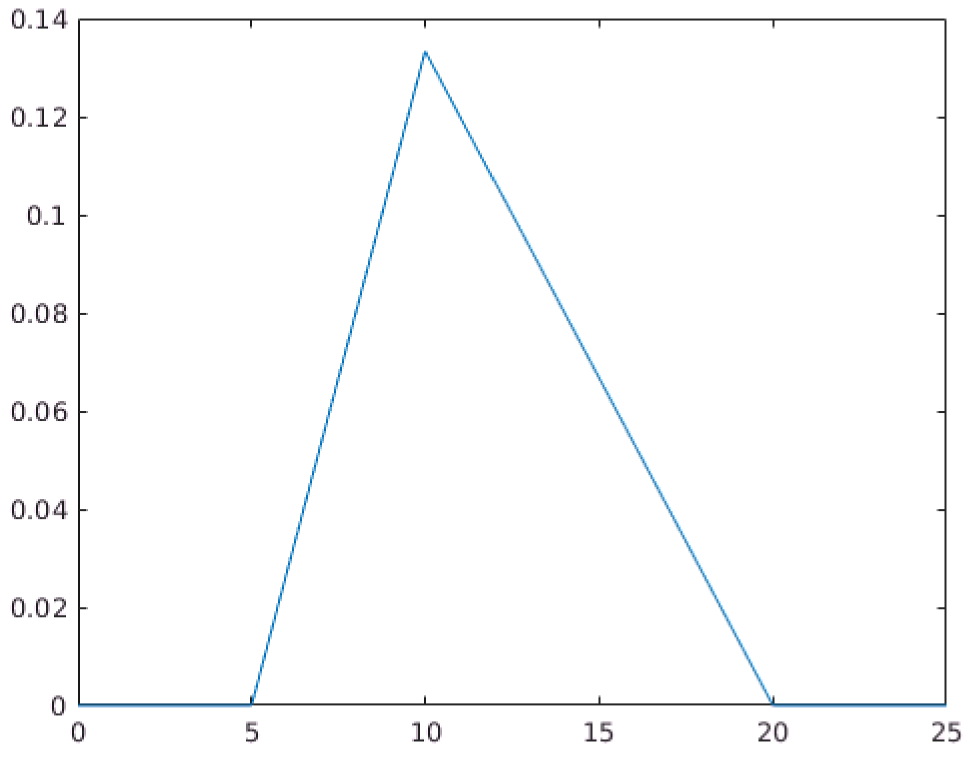


The Triangular distribution

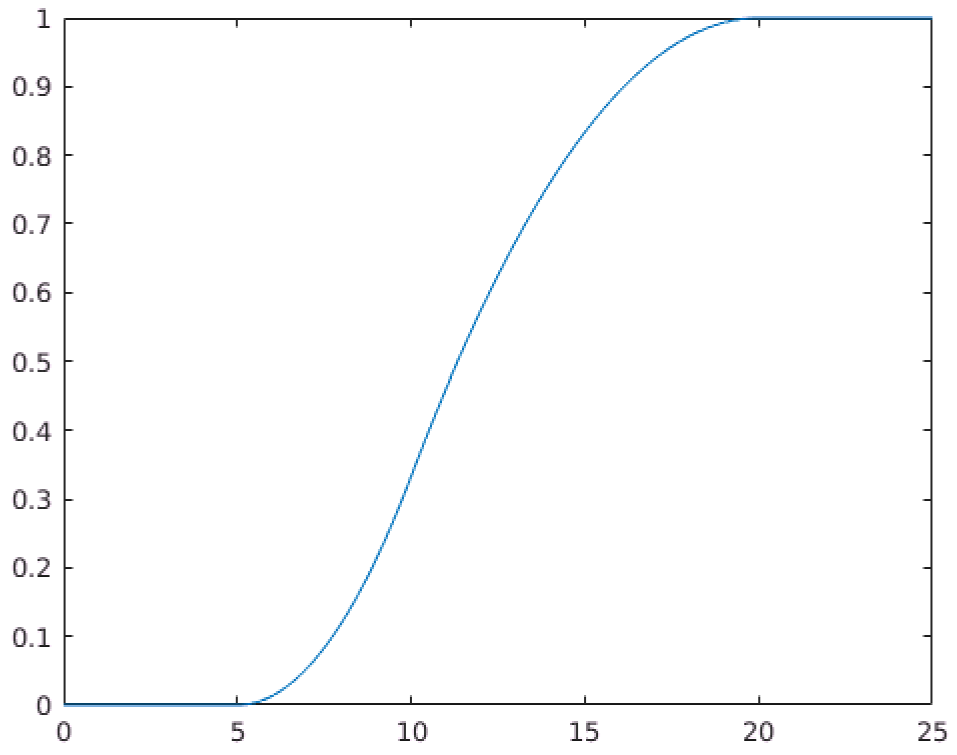
a controls the minimum value, b controls the mode, and c controls the maximum value

It can be useful as a first-guess distribution because human experts are pretty good at estimating "smallest possible", "most likely" and "largest possible' values.

```
X = 0:0.01:25;
dist=makedist('Triangular','a',5,'b',10,'c',20);

plot(X, dist.pdf(X));
```

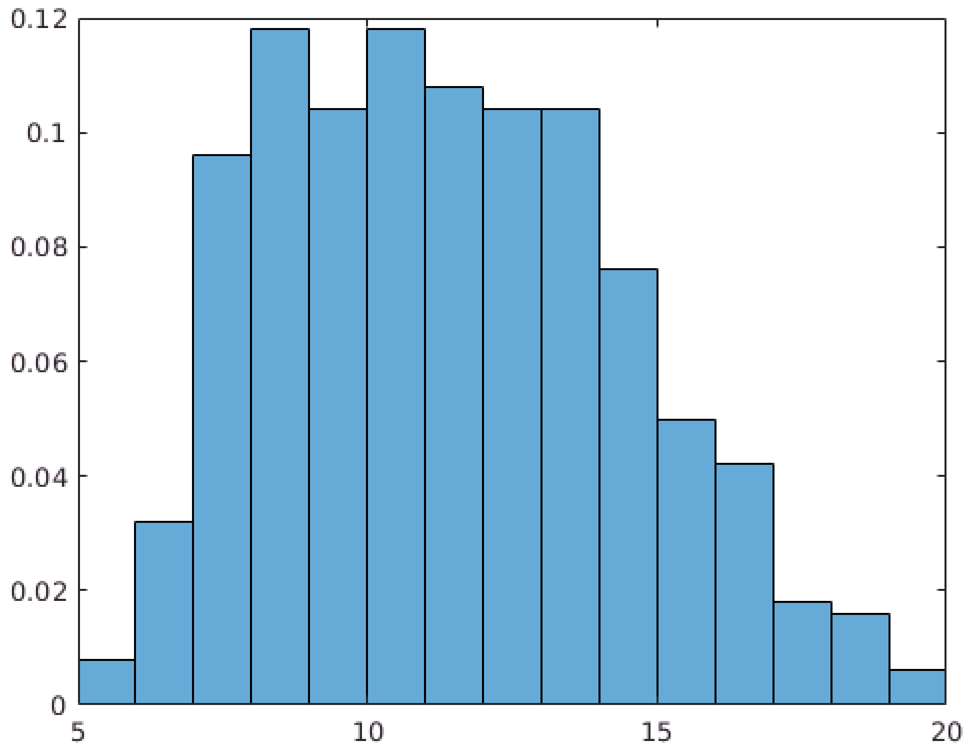


```
plot(X, dist.cdf(X));
```



```
data = dist.random(500,1);
```

```
histogram(data, 'Normalization','probability')
```

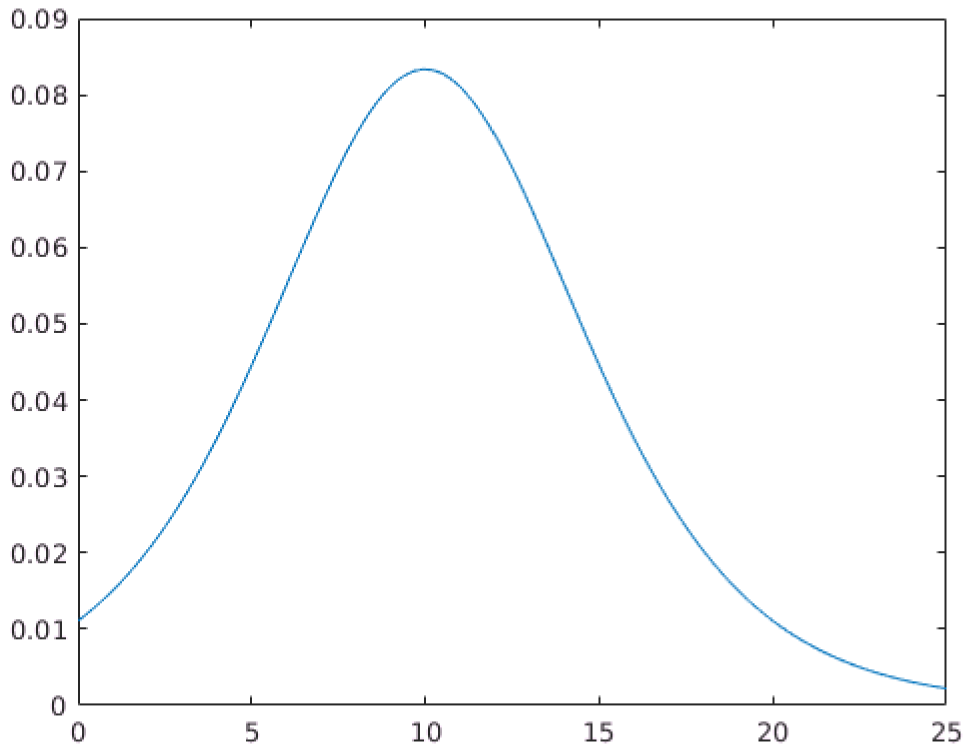


The Logistic distribution

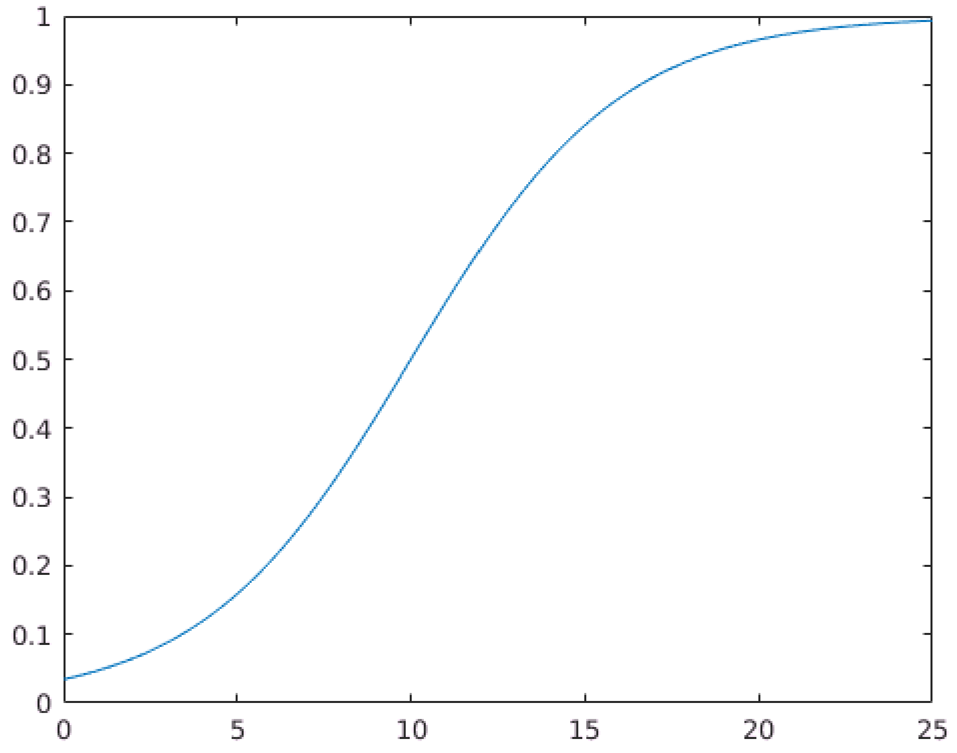
Similar to a normal distribution, mu controls the position of the mean, and sigma controls the width.

It is used in feedforward neural networks as a normal-distribution-like-thing that has fatter tails and is fast to calculate.

```
X = 0:0.01:25;
dist=makedist('Logistic','mu',10,'sigma',3);

plot(X, dist.pdf(X));
```

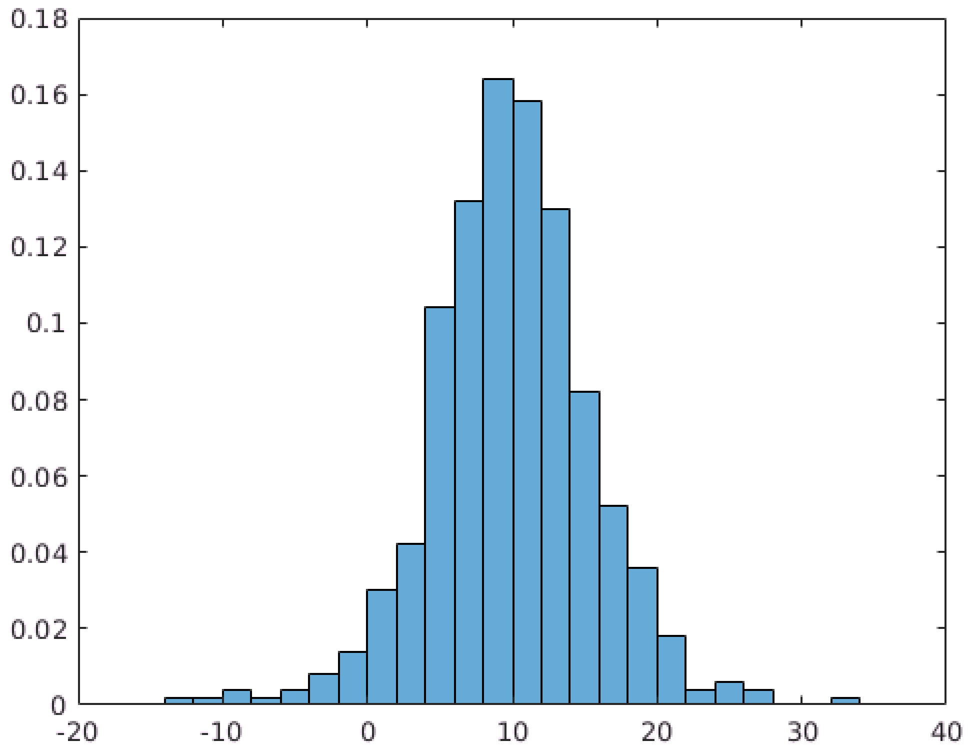


```
plot(X, dist.cdf(X));
```



```
data = dist.random(500,1);
```

```
histogram(data, 'Normalization','probability')
```


0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-20
-10
0
10
20
30
40