Using the normal distribution in Matlab

`normpdf, normcdf,and norminv`

Learning objectives

- How to use normpdf, normcdf, and norminv in Matlab
- How to make simple pdf and cdf plots
- How to make confidence intervals with different ranges than 95%

Calculating the normal distribution pdf

Matlab has a built-in function that calculates the value of the normal pdf for any value of x, mu (mean), and sigma (standard deviation). The syntax is

```
f = normpdf(x,mu,sigma)
```

where f is the pdf value at x.

The function is vectorized – you can pass an array of values as x, and f will be an array of pdf values. (This is useful for plotting.)

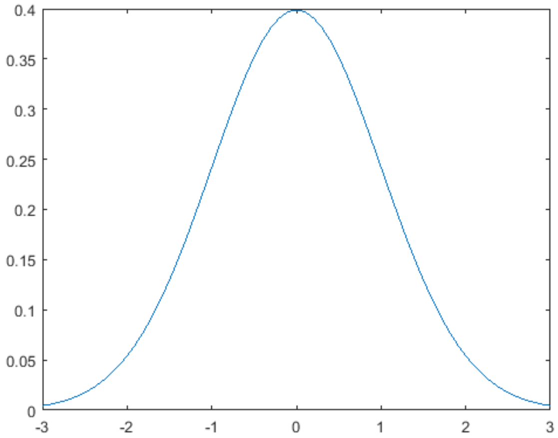
It is important to remember the differences between a pdf and cdf – we use cdfs for confidence intervals. cdfs have values between 0 and 1. pdfs can take on any non-negative value.

How to use normpdf at the command line with a vector input:

```
>> mu = 3;
>> sigma = 2;
>> x = [-0.4 0.4 4.0]

x =

   -0.4000    0.4000    4.0000

>> f = normpdf(x,mu,sigma)

f =

    0.0470    0.0857    0.1760
```

Quick and dirty plotting:

```
>> x = -3:0.1:3;
>> f = normpdf(x,0,1);
>> figure
>> plot(x,f)
```

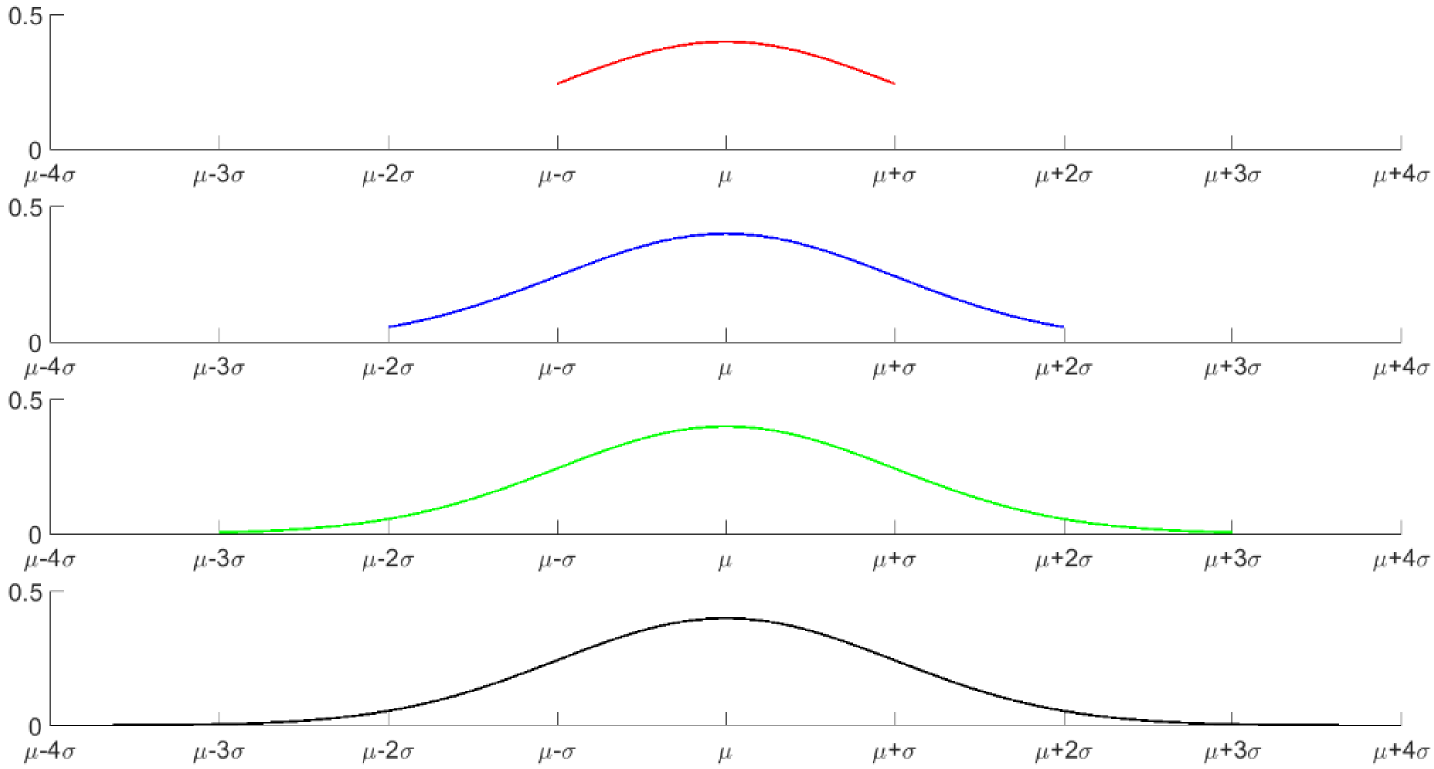


Plotting the normal pdf

For the normal distribution, a large, large proportion of the pdf is contained between mu – 3*sigma and mu + 3*sigma. For plotting a normal pdf, you rarely have to plot beyond those bounds.

At 1 sigma, the function is still changing.

Very little changes between 3 and 4 sigma.



Calculating the normal distribution cdf

As we discussed in class, there is no analytical solution to the normal cdf (you can't write it as a combination of simple functions). It has to be evaluated numerically (one of the reasons for the creation of z tables.)

However, Matlab has a built-in function for this numerical evaluation of the normal cdf for any value of x, given mean mu and standard deviation sigma. The syntax is

```
P = normcdf(x,mu,sigma)
```

where p (a value between 0 and 1) is the cumulative probability of x.

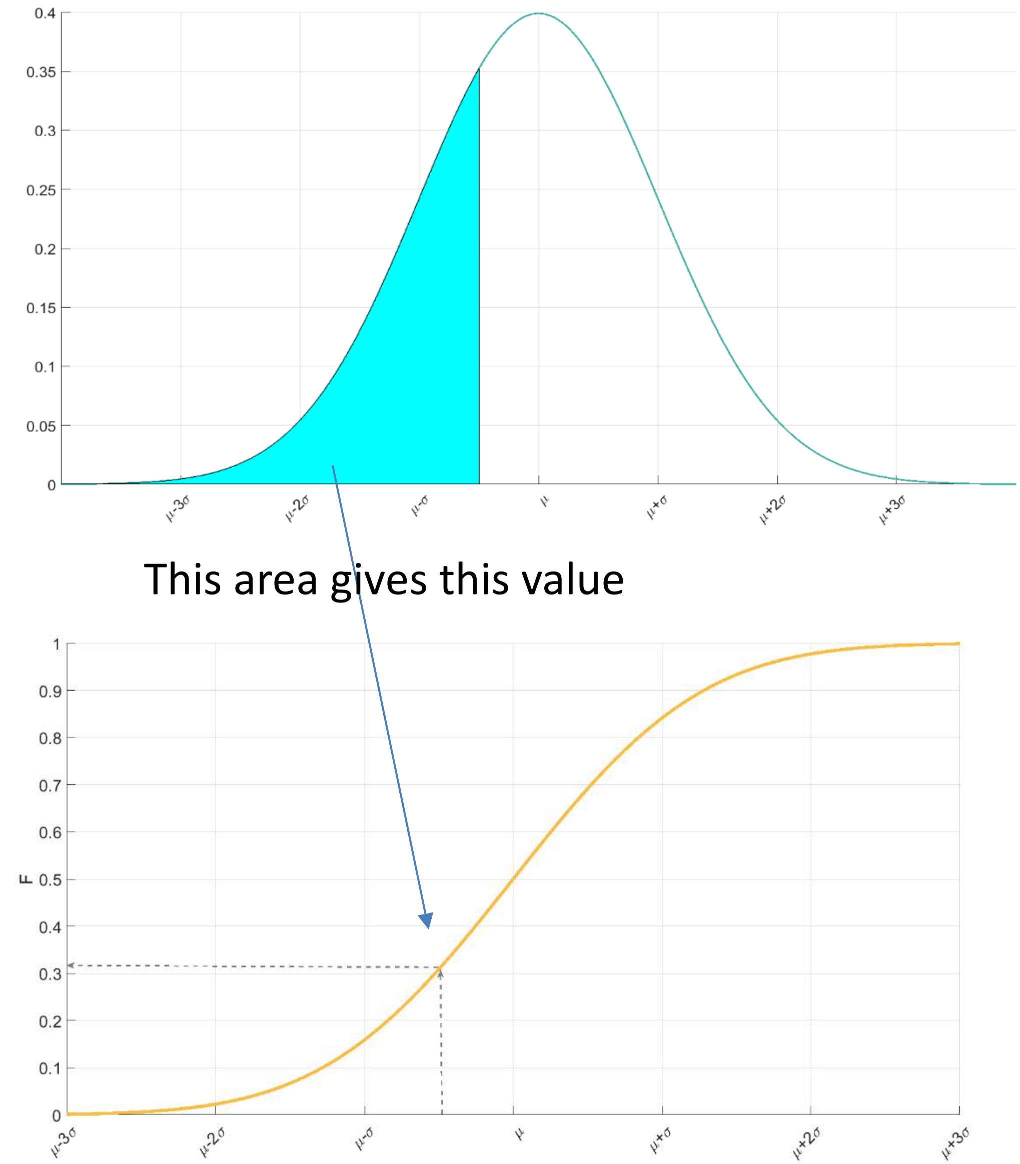
This function is also vectorized, which means you can pass an array of x values, and get back an array of p values. This is again useful for plotting.

How to use normcdf at the command line with vector input:

```
>> mu = 3;
>> sigma = 2;

>> x = [-.4 .4 4]

x =

   -0.4000    0.4000    4.0000

>> F = normcdf(x,mu,sigma)

F =

    0.0446    0.0968    0.6915
```

Explaining the relationship between the normal pdf and cdf graphically

Since the cdf is defined as the integral of the pdf (and vice versa), the value of the cdf at x is the same as the area under the curve of the pdf taken from minus infinity to x.



Using normcdf to make a z-table

If we use mu = 0 and sigma = 1 (the values for the standard normal), then normcdf will return the probabilities found on the z-table.

Now you can create your own z-table (more fun than watching paint dry!)

```
clear
fprintf('Hey, I''m part of a z-table!\n\n')

% we will add p1 and p2 values to get the z-value,
% just like in a real z-table
p1 = 0:0.01:0.09;
np1 = length(p1);

p2 = 0:0.1:1.5;
np2 = length(p2);

% top line of table – we do some fancy fprintf formatting
% don’t worry if you don’t understand it
ncol = 10;
fmt = ['   ', repmat('\t%3.2f', 1, ncol - 1), '\t%3.2f\n'];
fprintf(fmt, p1)

% now make a double loop to sequentially print out probabilities
% corresponding to z-values, using normcdf; this could also be
% done vectorally, but it might be more confusing?
for i = 1:np2
    % print left-most column, which is p2 in our implementation
    fprintf('%2.1f',p2(i));

    % now loop over all the p1 values, add then to the current
    % p2 value for this row, and calculate using normcdf
    for j = 1:np1
        z = p2(i)+p1(j);
        fprintf('\t%5.4f', normcdf(z,0,1))
    end

    % move to a new line
    fprintf('\n')
end
```

Results from the script on the previous slide

```
>> makeMeAzTable
Hey, I'm part of a z-table!
```

	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441

Calculating the inverse of the normal cdf

As we just discussed, normcdf gives you a cumulative probability P, given x, mu, and sigma. But what if you want to find out the x-value that corresponds to a given P, mu, and sigma?

This is actually a harder problem numerically than finding P (root-finding of non-analytic function vs. numerical integration of smooth function – don't worry if that's all gobbly-gook now.)

Fortunately, Matlab's got your back.

The built-in function norminv provides the value of x given P, mu, and sigma. Its syntax is

```
x = norminv(P,mu,sigma)
```

Again, it's vectorized; P can be an array, and x will then be an array.

How to use normcdf at the command line with vector input:

```
>> p = [0.1 0.3 0.9]

p =

    0.1000    0.3000    0.9000

>> x = norminv(p,3,2)

x =

    0.4369    1.9512    5.5631
```

Note we get the special confidence interval value for $\frac{\alpha}{2} = 0.025$

```
>> x = norminv(0.975,0,1)

x =

    1.9600
```

Note norminv doesn't work for values outside [0 1]

```
Trial>> x = norminv(-0.2,0,1)

x =

   NaN
```

Norminv and Confidence interval alphas

You can use `norminv` to calculate z-values for different percentage confidence intervals.

The value of $z = 1.96$ you've seen before is for a 95% two-sided confidence interval.

It's the value from the z-table for 0.975 of the probability – which means that 0.025 of the probability is in values greater than 1.96.

Since the distribution is symmetric, there is also 0.025 of the probability beyond -1.96.

0.025+0.025 = 0.05 chance you end beyond ±1.96 randomly

95% of the probability is between -1.96 and 1.96

Hey, I'm part of a z-table!

	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	**0.9750**	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817

Norminv and Confidence interval alphas (2)

What if you want an 80% two-sided confidence interval?

That means we want to leave 10% beyond the z-value, and a corresponding 10% below the negative of the z-value.

So we want to find a value on the table close to 0.90.

Looking at the table, we see that for z=1.28 we are very close to 0.90, and so we can use 1.28 instead of 1.96 to get a 80% confidence interval.

Hey, I'm part of a z-table!

	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	**0.08**	0.09
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	**0.8997**	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441

Norminv and Confidence interval alphas (3)

But we have a computer. And Matlab. We can do better.

The z-table gives cumulative probability values for the standard normal distribution (mu = 0, sigma = 1)

Given a probability p, `norminv(p,0,1)` gives the corresponding z-value

```
Trial>> norminv(0.9,0,1)

ans =

    1.2816
```

A more exact z-value for an 80% confidence interval is 1.2816