ISTA 116 Lab: Week 11

Last Revised November 15, 2011

1 Continuous Probability Distributions

- With discrete random variables, we could make a list of possible values, and write down the probability of any given value.
- For a continuous variable, any particular value has (in principle) infinitely many decimal places, and so the probability of any exact value is zero.

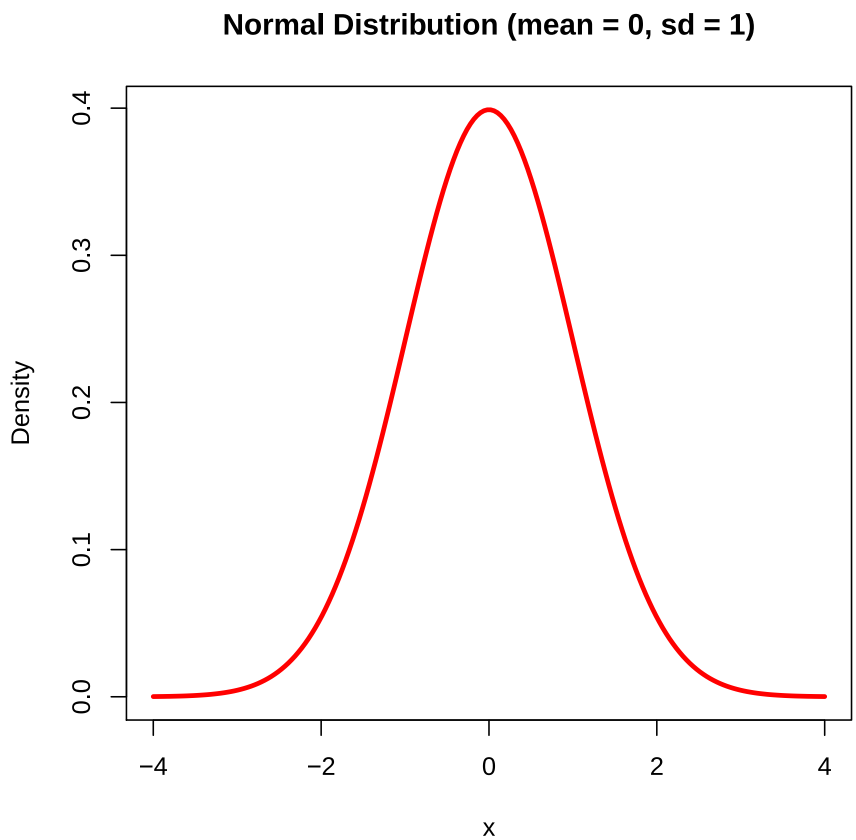
  If X is continuous, then $P(X = x) = 0$ *for any particular value of* x!
- Probabilities only have positive value over *intervals*. So, I can ask for example, "what's the probability of any randomly selected person being *between* 6 ft. 1 in. and 6 ft. 2 in.?"

1.1 Probability Density Functions

- With discrete variables, we defined a function mapping values of a random variable to probabilities.
- With continuous variables, though, since the probability of any particular value is 0, this wouldn't be very interesting.
- Instead, we map values to **probability densities**.
- Essentially, we're asking "How much probability is there per unit length?" around a value. Or, mathily:

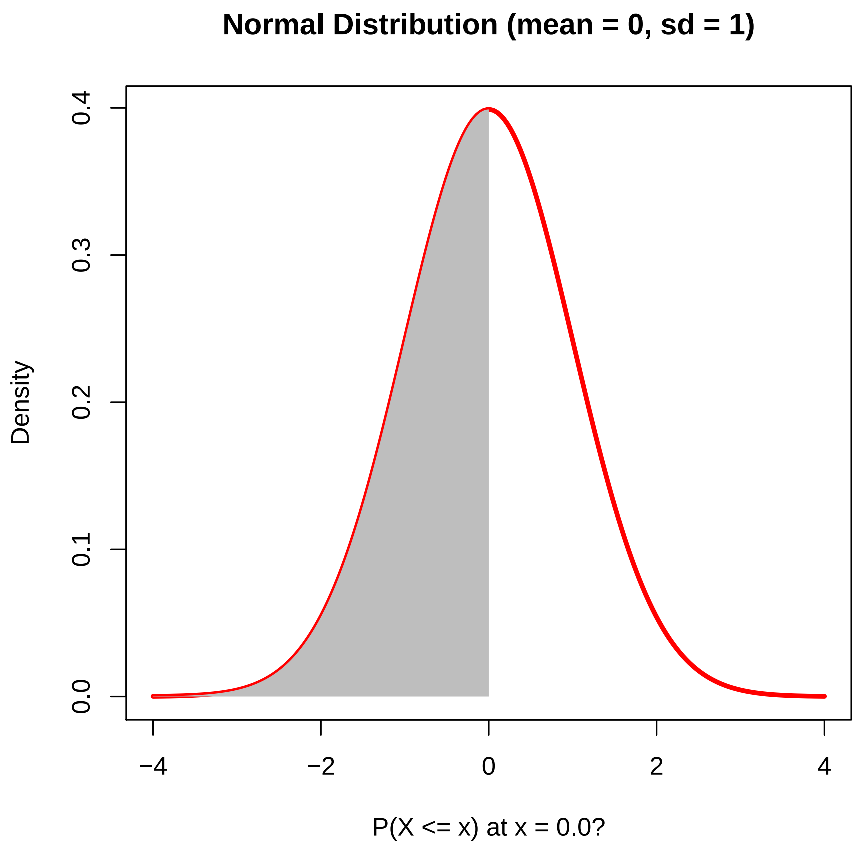
$$f_X(x) \approx \frac{P(x - \delta < X < x + \delta)}{2\delta}, \quad \text{for really, really small values of } \delta$$

- We can visualize a distribution by its **density curve**:



1.2 Cumulative Distribution Functions

- If we can think of the density as the "probability per unit length", then the average density times the length of an interval, in a plot the "area under the curve" is a probability.
- This is the same as when we had a histogram with `prob = TRUE` in `R`: the actual proportion was the width of the bar times the height (density).
- **Question:** Can we ever have $f(x) > 1$? Why or why not?
- The **cumulative distribution function** (or *cdf*), written $F(x)$, gives the "cumulative" probability up to the value x. For example:



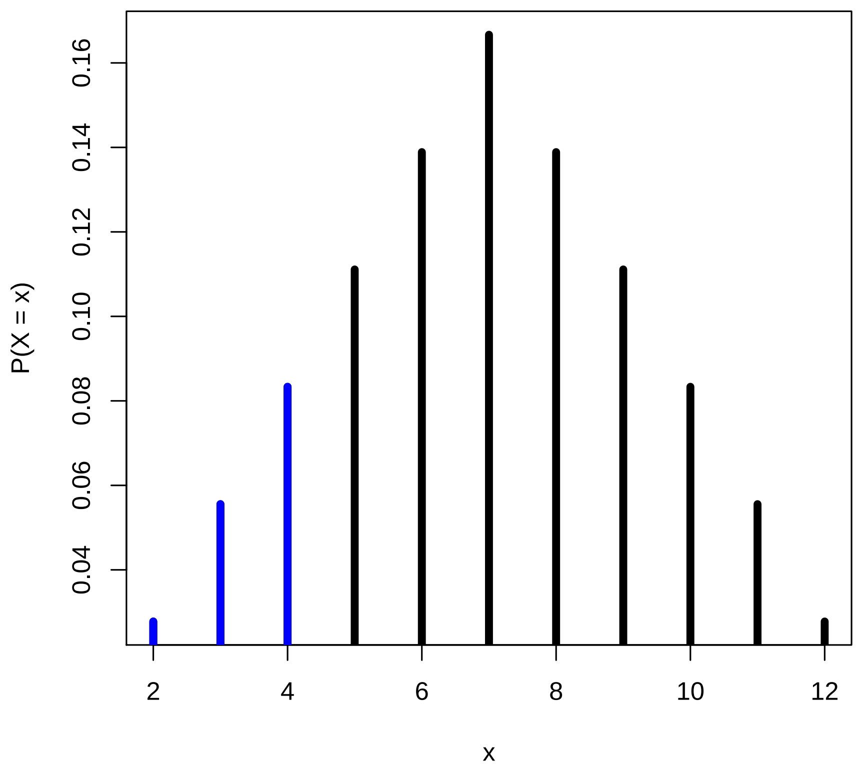
- Calculus fans may have noticed the similarity between the definition of the probability density and the definition of a derivative. This is not a coincidence.
- The density is the instantaneous change in the accumulated probability at a particular point, x.
- Just like probability is distance and density is velocity.
- Therefore, the *cdf* is the integral of the density:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)$$

- In fact, the notion of a *cdf* applies equally well to discrete distributions. Simply apply the definition: $F(x) = P(X \leq x)$, and replace density function with probability function, and integral with sum:

$$F(x) = P(X \leq x) = \sum_{-\infty}^{x} P(X = x)$$

Cumulative probability, $F(4) = P(X \leq 4)$, on a discrete distribution, represented by total blue spike height:



2 Continuous Probability Distributions in R

The functions we introduce here are quite similar to *dbinom* and *pbinom*.

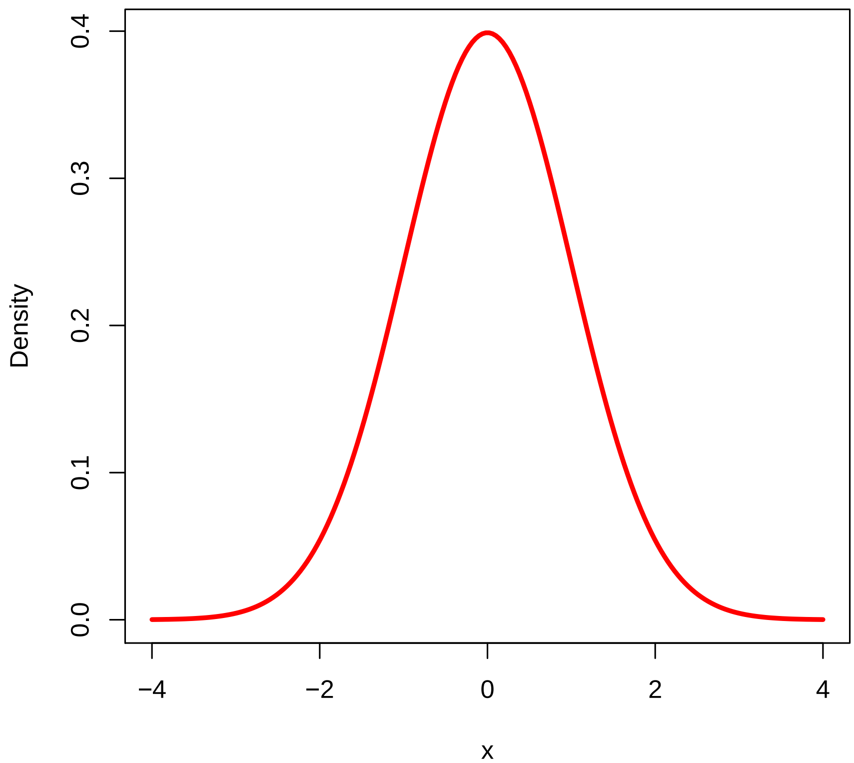
2.1 Probability Density Functions in R

2.1.1 *dnorm*(x, mean = 0, sd = 1)

Given a set of values it returns the height of the normal probability distribution at each point. If you do not specify mean and sd, the defaults are a mean of zero and standard deviation of one.

Here's how we can plot a distribution. We've seen the *seq(a, b, length=n)* function long ago, it creates a sequence of *length* numbers from a to b. Here we create a vector of 200 equally spaced values for x. Then we get the height of the distribution at each of the x values to use as our y values and we plot them:

```
> x = seq(-4, 4, length = 200)
> y = dnorm(x, mean = 0, sd = 1)
> plot(x, y, type = "l", xlab = "x", ylab = "Density", lwd = 3,
+     col = "red")
```



Notice that we've specified the *mean* and *sd* for our distribution (in fact, the values we use here are the defaults, but it doesn't hurt to always specify.)

2.1.2 ***dunif*(x, min = 0, max = 1)**

Similar to *dnorm*, but uses a uniform distribution, so it doesn't get a mean and sd, instead it gets a min and max.

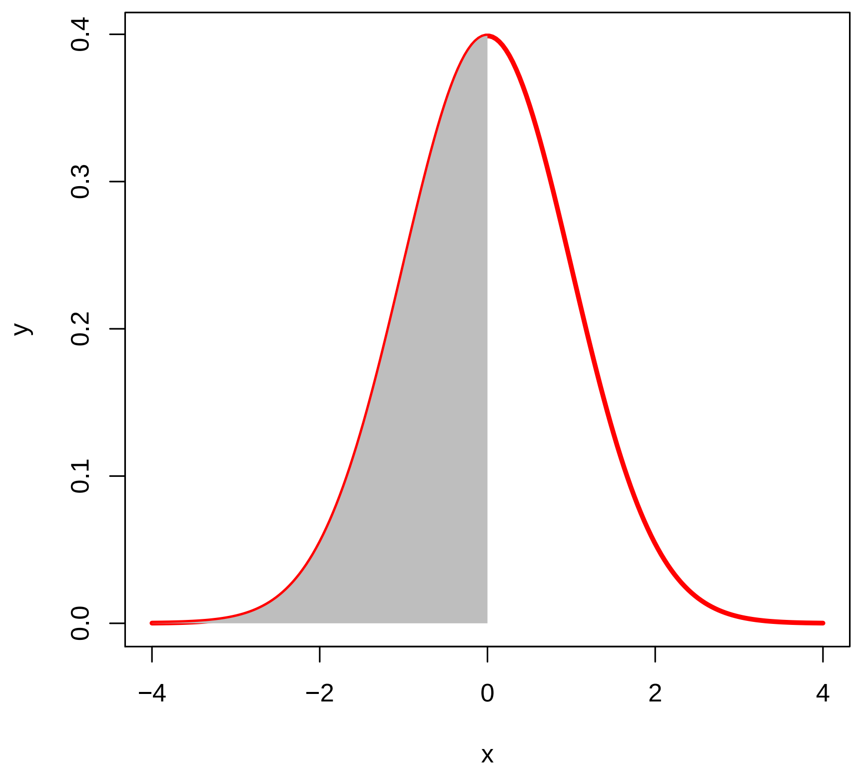
2.2 Cumulative Distribution Functions in R

2.2.1 *pnorm*(x, mean = 0, sd = 1)

Given a number or a list it computes the probability that a normally distributed random number will be less than that number. It accepts the same options as *dnorm*.

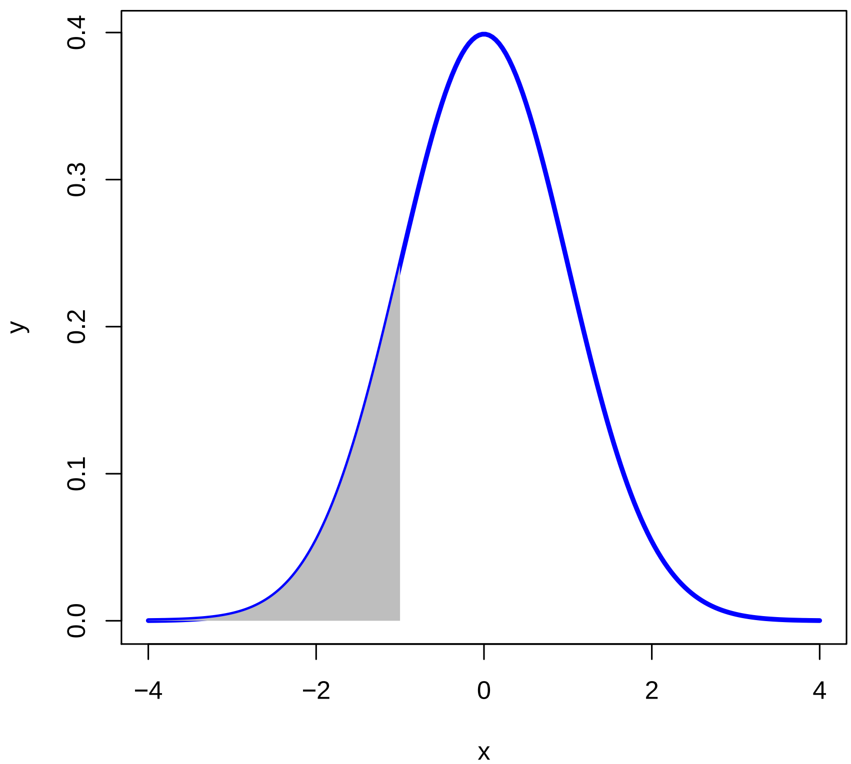
Plotting the shaded region is a bit more complicated (it is not necessary to know how to do this, it just looks nice):

```
> x = seq(-4, 4, length = 200)
> y = dnorm(x)
> plot(x, y, type = "l", lwd = 3, col = "red")
> x = seq(-4, 0, length = 200)
> y = dnorm(x)
> polygon(c(-4, x, 0), c(0, y, 0), col = "gray", border = NA)
```



Exercises:

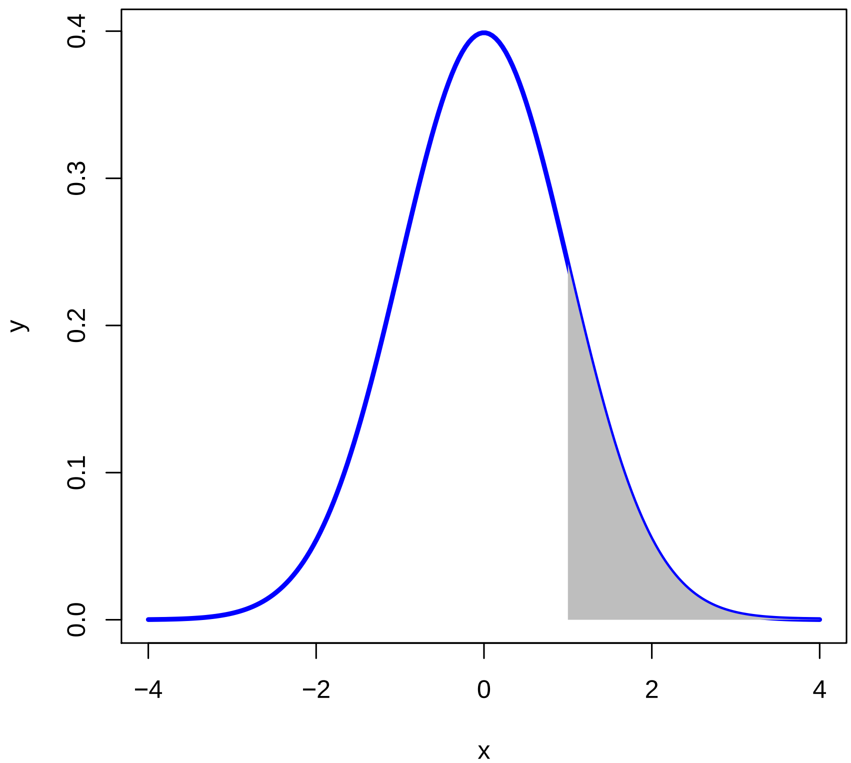
- Find the probability of drawing a number from the standard normal distribution that is smaller than -1:



ANSWER:

```
[1] 0.1586553
```

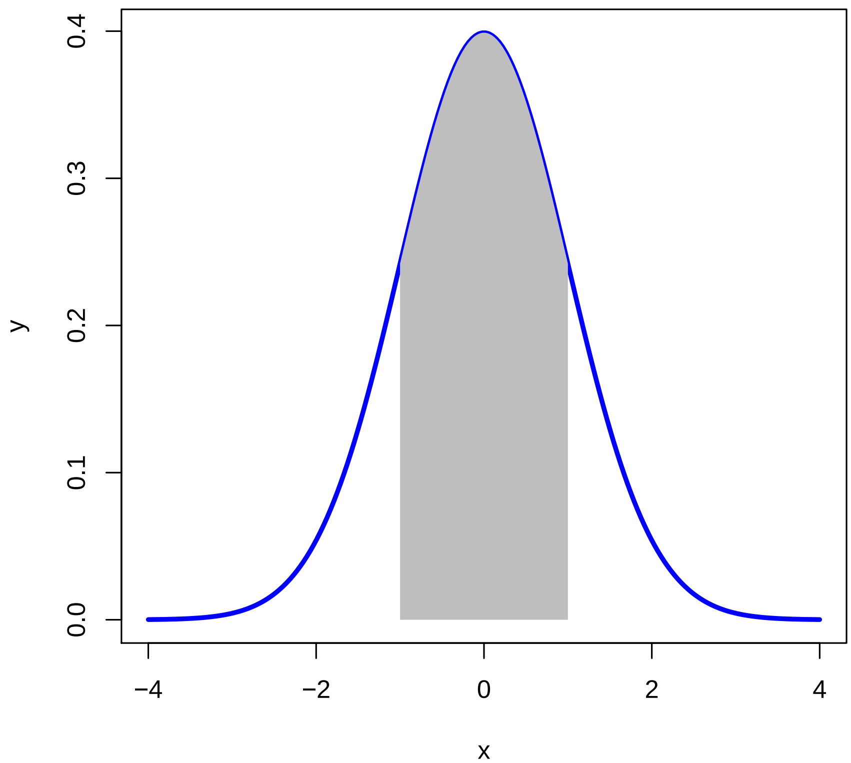
- Find the probability of drawing a number from the standard normal distribution that is greater than 1:



ANSWER:

```
[1] 0.1586553
```

- Find the probability of drawing a number from the standard normal distribution that is between -1 and 1:



ANSWER:

```
[1] 0.6826895
```

- Assume that the test scores of a college entrance exam fits a normal distribution. Furthermore, assume the mean test score is 72, and the standard deviation is 15.2. What is the percentage of students scoring 84 or more on the exam?
  - We apply *pnorm* using a mean of 72 and standard deviation of 15.2.
  - Wait, something looks very wrong. This gives us the percent of students scoring 84 or less.
  - So we subtract the result from 1.
  - The percent of students scoring 84 or more in the college entrance exam is 21.5%.

- Reported data[1] on brain weights of victims of Alzheimer's disease have a mean of 1076.80 grams and a standard deviation of 105.76 grams. Assume the brain weights are normally distributed.
  - What is the probability that a randomly selected victim of Alzheimer's will have a brain that weighs less than 800 grams?
  - What is the probability of a randomly selected victim's brain weighing between 1200 and 1300 grams?
  - Suppose an eccentric zombie likes to dine strictly on Alzheimer's brains. He selects 5 at random and feasts on them. What is the probability of all 5 brains weighing more than 1100 grams?
  - What assumption(s) have we made in the zombie question?

2.2.2 *punif*(x, min = 0, max = 1)

Similar to *pnorm*, but uses a uniform distribution, so it doesn't get a mean and sd, instead it gets a min and max.

[1] "Some Questions Concerning the Pathological Anatomy of Alzheimer's Disease", Dusheiko, 1974