PS6 Lecture 3

Daniel Lim

University of California, Los Angeles

4/7/2014

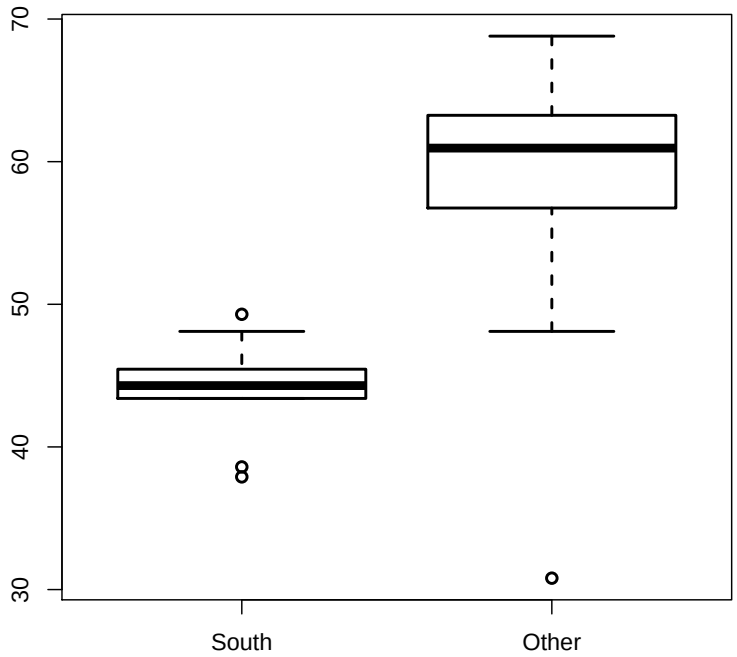
Agenda

- Intro to the Normal distribution
- Density
- Density plots
- Normal distribution redux

Motivation

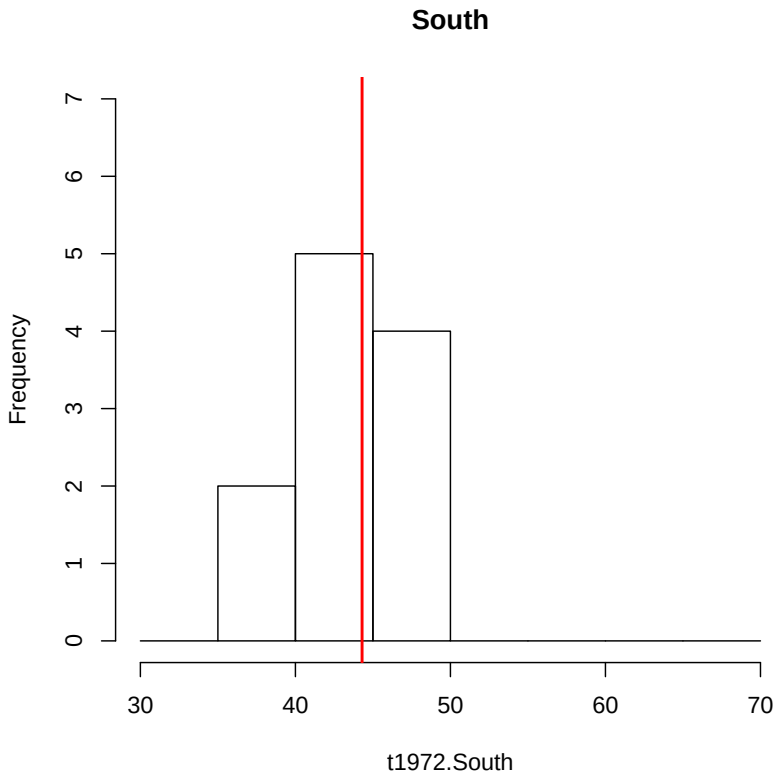
Where did we stop in lecture 2?

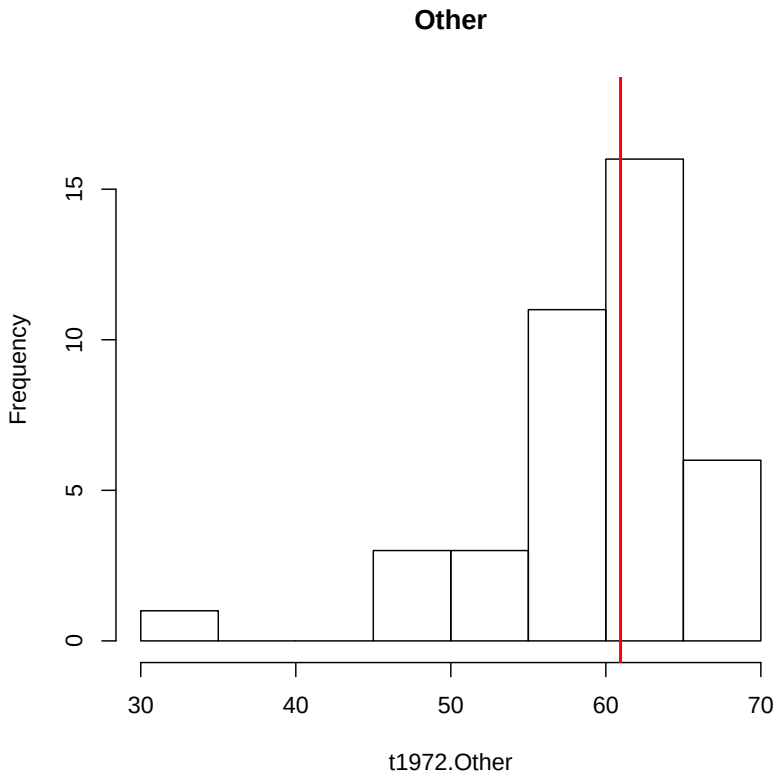
- The 1976 turnout data appeared to be bimodal
- We suspected it's important whether a state is in the South
- By subsetting, we saw that region separates 2 distinct groups of values



Motivation

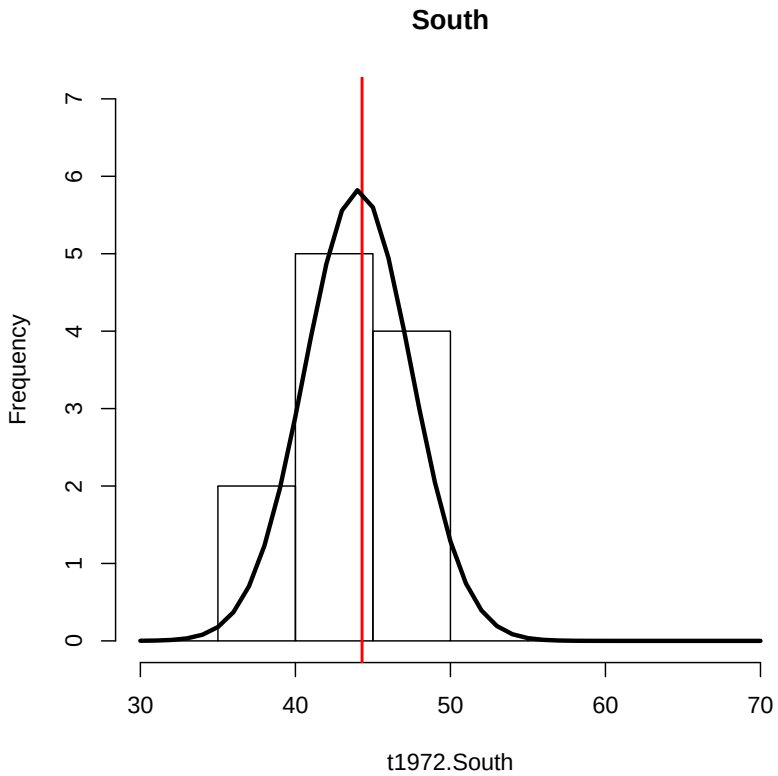
These graphs are pretty satisfying, but why?

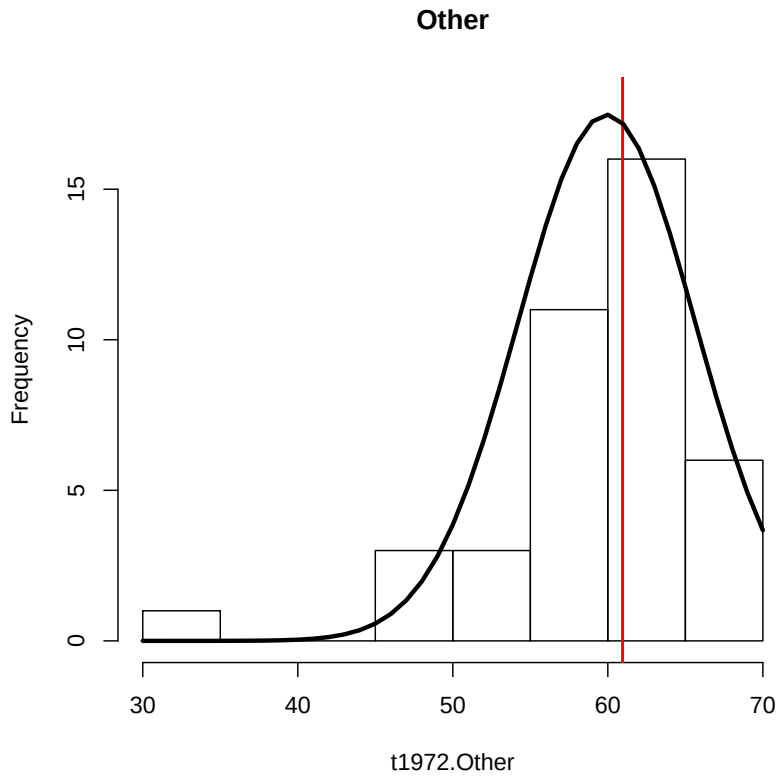




Motivation

These graphs are pretty satisfying, but why?





Because they appear to be normally distributed!

Description/Definition

The **normal distribution** (AKA bell curve, Gaussian distribution) is perhaps the most important distribution in statistics.

It's important because it...

- describes a plethora of real-world phenomena (like turnout)
- is central to many of advanced analytic techniques

It also provides a great introduction to many important concepts including density and density plots, the idea of distributions, random variables (RVs), and discrete vs. continuous RVs.

Description/Definition

We say that some data X are **normally distributed** if its **density** can be described as follows:

The normal density function:

$$\frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Initial reaction?

Description/Definition

We say that some data X are **normally distributed** if its **density** can be described as follows:

The normal density function:

$$\frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Initial reaction?"Eww"

It's not so bad once we parse it out. But, we will need to introduce several new concepts to fully understand this equation. Once we do, we'll return to more fully explore the normal distribution.

Density

First, we need to introduce the concept of **density**.

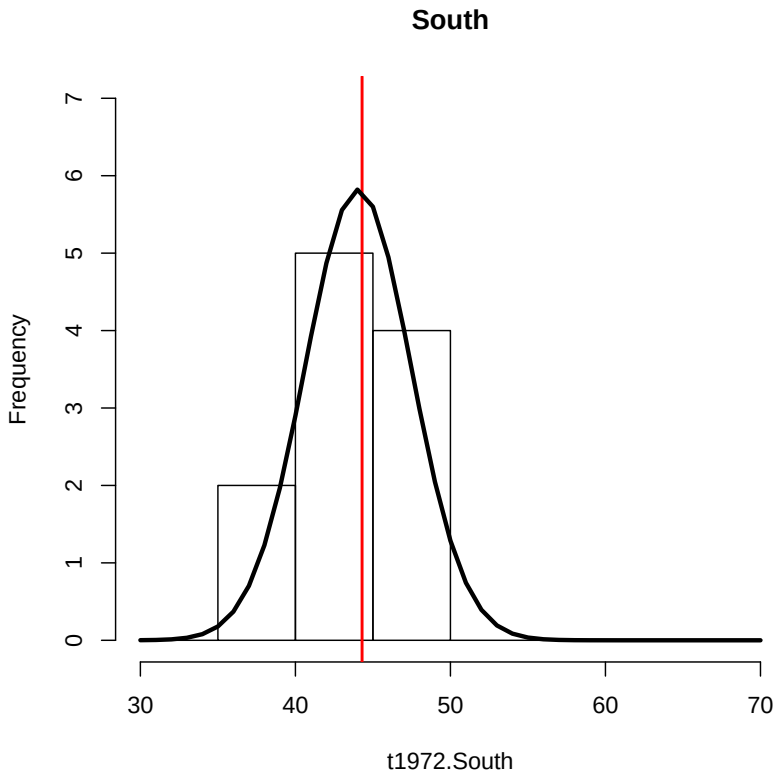
Informal definition: For some data X, the density, $f_X(x)$, is a function describing how likely it is that $X = x$.[1]

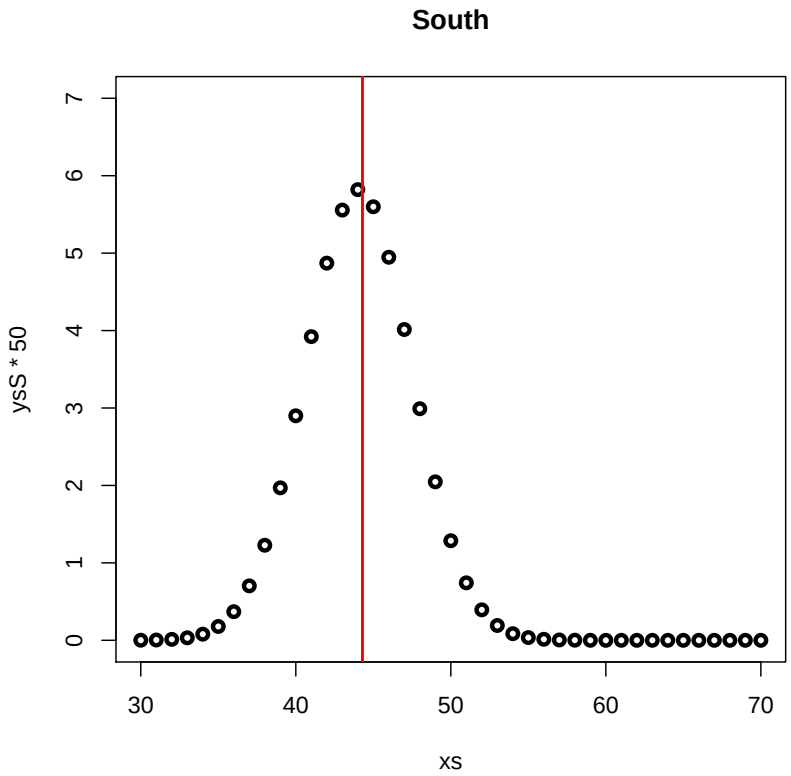
- density is always written with a lower case 'f'
- the subscript 'X' indicates that this is the density for data 'X'
- AKA probability density and PDF
- **density IS NOT a probability**

[1] This is an intuitive definition, but not the formal one. We'll get to that distinction in the next slide.

Density

The left graph is the histogram for Southern turnout. The right is a plot of the corresponding PDF.





Density

The formal definition for a PDF is slightly more complicated:

formal definition:

$$\int_{-\infty}^{+\infty} f(x)dx = 1$$

$$f(x) \geq 0$$

Density

The formal definition for a PDF is slightly more complicated:

formal definition:

$$\int_{-\infty}^{+\infty} f(x)dx = 1$$

$$f(x) \geq 0$$

line 1: don't worry about the integral. . . this simply means that the area under a density curve is equal to 1. We'll get to why in a sec.

line 2: this is a formality arising from the relationship between density and probability.

Density

Density is only meaningful for an infinitesimally small slice of values

What does this mean?

Rather than give some hard-to-understand explanation, let's illustrate using some simulated (fake) data.

Example 1

Let's simulate some normally distributed data. We use the 'rnorm' command to do this.

```
> fakedata = rnorm(1000, 0, 1)
```

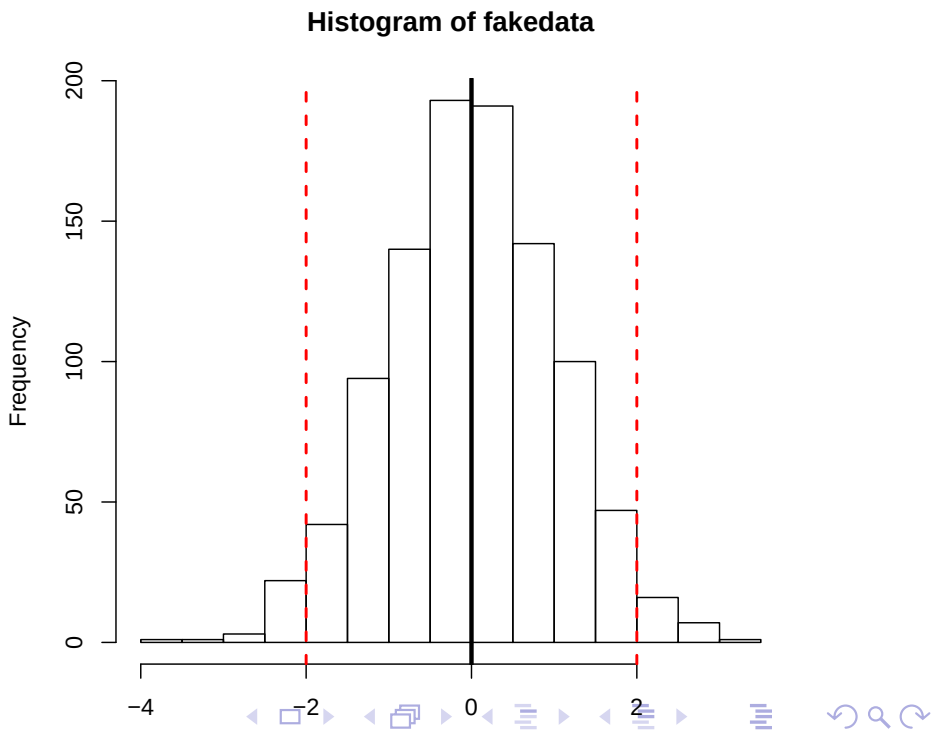
- arg 1: How many data points to create ('1000' above)
- arg 2: the mean of the data (0)
- arg 3: the SD of the data (1)

The 'r' in 'rnorm' stands for random, as in "randomly draw a bunch of numbers from a normal distribution with mean and SD equal to arguments 2 and 3."

Example 1

```
> hist(fakedata)
> abline(v=0, lwd=3)
> abline(v=2, col='red', lty=2, lwd=2)
> abline(v=-2, col='red', lty=2, lwd=2)
```

Using a histogram, we see that these new data look like a bell curve (i.e. are normally distributed)



Example 1

What do we observe?

- The center of the data are around 0, which is what we expect since we specified the mean to be 0.
- most of the data lie within 2 SDs of the mean (between the 2 dotted, red lines). This is a property of normal distributions (more on this later).

We see what we expect, so we're confident the function did what we asked it to do.

Example 1

Now, let's generate some density values.

```
> densAt1 = dnorm(1, 0, 1)
```

The 'd' in 'dnorm' stands for density, akin to the 'r' in 'rnorm'.

Example 1

The function 'dnorm' returns density values for the normal distribution, using arguments similar to those we used for the 'rnorm' function.

arg 1: What values of X to find the density at at. This can be a scalar or a vector. Here, we passed it a scalar: 1

arg 2: the mean of the data (again, 0)

arg 3: the SD of the data (again, 1)

Example 1

Looking at 'densAt1', we see that the *density* for some data X that are normally distributed with mean 0 and SD 1 is 0.242.

```
> densAt1

[1] 0.2419707
```

Given the name of the function, we might think that if we drew 1000 random draws from the same distribution, about 240 of them should equal 1. . . right?

Example 1

Not so much[2].

```
> fakedataEq1 = subset(fakedata, fakedata==1)
> length(fakedataEq1)
[1] 0
```

[2] The 'length' function returns the number of elements in a vector. Here there are 0 elements equalling 1, so it returns 0.

Example 1

Reason 1: The normal distribution can take on any value, not just nicely rounded values like 1, 1.05 or 0.95.

The subset of our fakedata between 0.95 and 1.05 shows several values around 1, but none exactly equal to 1[3].

```
> fakedataApprox1 = subset(fakedata, fakedata < 1.05 & fakedata > 0.95)
> fakedataApprox1

 [1] 0.9566744 0.9615188 1.0435857 1.0363514 0.9717834
 [6] 1.0077769 1.0347709 0.9524631 1.0413132 1.0254592
[11] 1.0457278 0.9758647 1.0300776 0.9721601 1.0307872
[16] 1.0238909 0.9705676 0.9737822 0.9581409 1.0104524
[21] 0.9636334 1.0128088 0.9563298 1.0468416 0.9582823
[26] 1.0383425 0.9646165 0.9618981 1.0355053
```

[3] In this 'subset' call, there are 2 conditions joined by an ampersand (&). That symbol means AND; both conditions 1 AND 2 must be true.

Example 1

Reason 2: Even though we can think of it as a measure of likelihood, **density IS NOT a probability**.

- We want probability (it's an intuitive measure) – how do we get there?
- What use is the density?

Cumulative density

Integrating the PDF gives us the **cumulative distribution function** (CDF) which IS a probability.

Definition: For some data X with PDF f_X, the CDF is:

$$F_X(b) = Pr(X \leq b) = \int_{-\infty}^{b} f_X \, dx$$

Cumulative density

- **The CDF is the probability that the data takes on a value less than or equal to b.**
- The 'cumulative' in the name comes from the fact that it is the accumulation of the density function from $-\infty$ to b.
- This we can easily demonstrate using our fake data.

How many of our simulated observations are less than 1?

```
> fakedataLt1 = subset(fakedata, fakedata<1)
> length(fakedataLt1)
```

```
[1] 829
```

We use 'pnorm' to get the CDF from -∞ to b (arg 1)

```
> ThLt1 = pnorm(1, 0, 1)
> ThLt1 * 1000
```

```
[1] 841.3447
```

841.34 $\approx$ 829 Thus, we see that $F_X(b)$ is indeed a measure of the probability of seeing $X \leq b$.[4]

[4] The difference between the two is due to random chance in using 'rnorm.'

We can also use the CDF to determine the probability that some data fall within the range a to b.

$$Pr(a \leq X \leq b) = F_X(b) - F_X(a)$$

Let's determine, using simulated data as well as theory, the probability that data distributed normally with mean 0 and SD 1 falls between -1 and 1.

Using the simulated data:

```
> a = -1
> b = 1
> fakedataA2B = subset(fakedata, fakedata<1 & fakedata>-1)
> length(fakedataA2B)
```

```
[1] 666
```

Now, according to the definition of the CDF:

```
> pnorm(b, 0, 1) - pnorm(a, 0, 1)

[1] 0.6826895
```

Since $666 \approx 0.6827$, we verify that the CDF does indeed let us find $Pr(a \leq X \leq b)$.

Density plots

We can visualize PDFs using the **density plot**.

The density plot is simply a line-graph, where the points that are being connected are values of the density function f_X for some values of X.

Density plots

To create:

- determine the domain of X you're interested in.
- plug the range of values you're interested in into f_X.
- plot the resulting points, then connect with lines.

Recall from earlier this lecture that the function 'dnorm' returns density values for the normal distribution.

CDF plots

You can also plot the CDF, just like the PDF.

- The X-axis will be the same.
- The Y-axis will range from 0 to 1 for the CDF. This is because of the definition of the CDF
  - CDF is the probability/proportion of observations that will be below some value of x.
  - Probability cannot be less than 0 or greater than 1, so the Y-axis of the CDF is likewise constrained.

Example 3

First let's get the points we need to plot the PDF.

Since we defined the mean and SD of 'fakedata' as 0 and 1, a reasonable domain for X is -4 to 4.

```
> Xs = seq(-4, 4, 0.1)
```

Remember how we used X:Y to define a sequence of integers from X to Y? The function 'seq' is similar, except the third argument lets you specify step size. Here, we are getting a vector from -4 to 4 in increments of 0.1.

Now let's generate the PDF and CDF points using 'dnorm' and 'pnorm'

```
> Ys.PDF = dnorm(Xs, 0, 1)
> Ys.CDF = pnorm(Xs, 0, 1)
```

Finally, we'll plot these CDF and PDF points, using 'plot', the most basic graphing function in R.

```
> plot(Xs, Ys.PDF, main='Normal PDF, mean=0, SD=1', type='l')
> plot(Xs, Ys.CDF, main='Normal CDF, mean=0, SD=1', type='l')
```

The plot function

- requires 2 arguments – 1 for an x coordinate, another for y.
- coordinates can be scalars or vectors, e.g.:

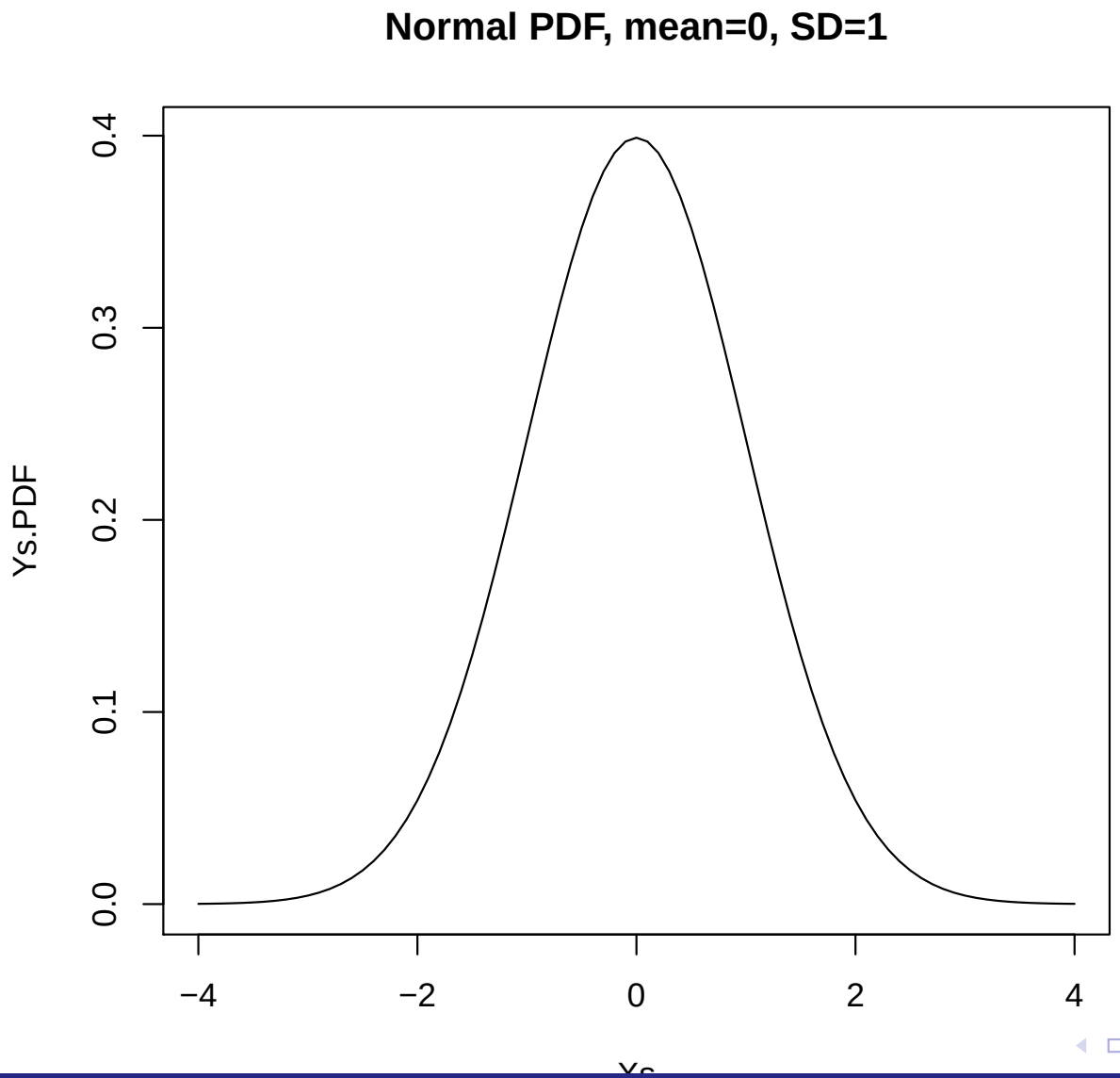
scalar : 'plot(2, 3)' to plot a single point with x=2, y=3

vector : 'plot(c(1, 2), c(3, 4))' to plot two points – one with $x = 1, y = 3$; another with $x = 2, y = 4$.

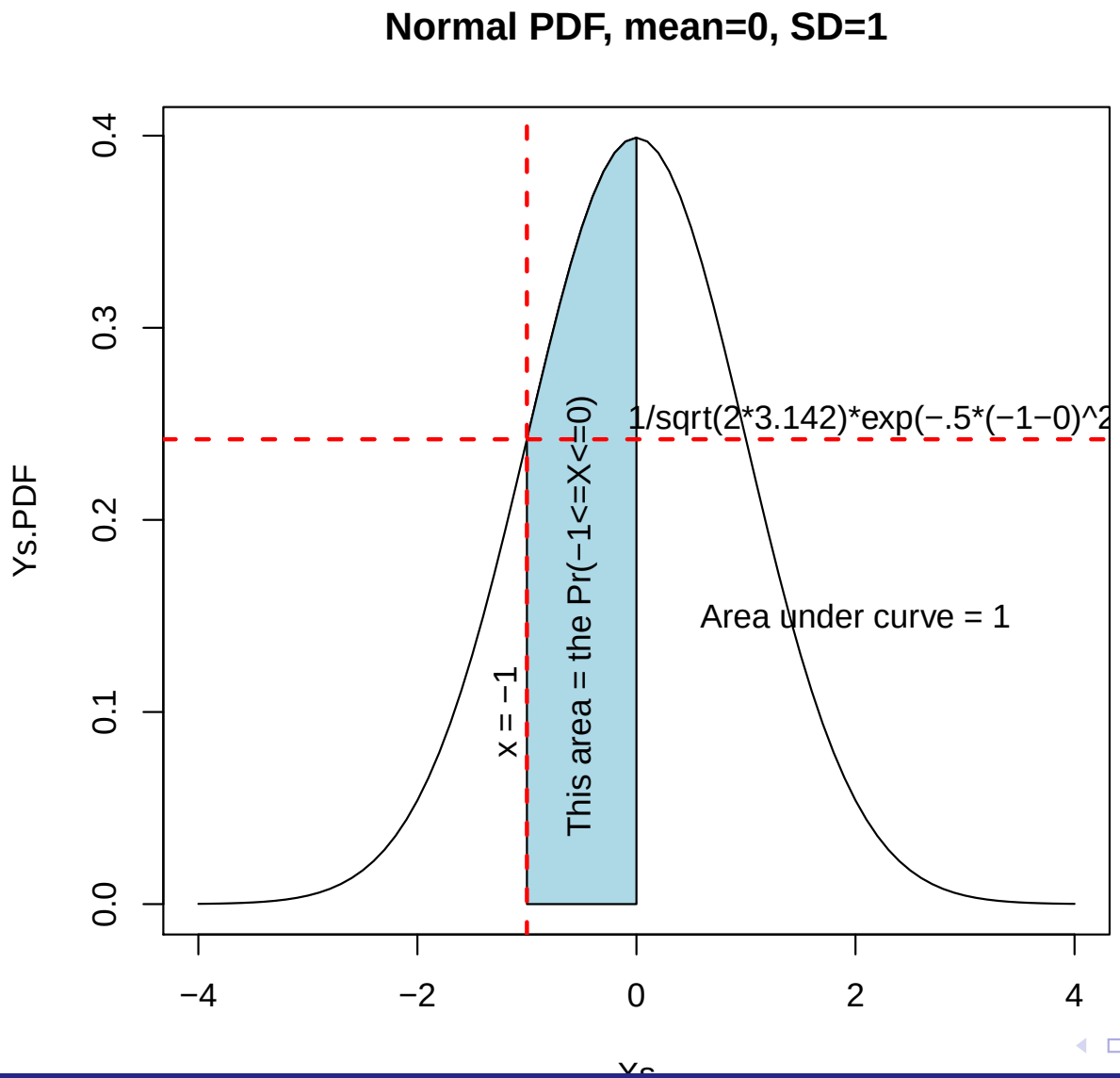
The plot function

- call 'plot' with "type='l'," to get lines, rather than points.
  - Try 'plot(c(1, 2), c(3, 4), type='l')'
- 'hist' & 'boxplot' are specialized versions of 'plot'.
  - Any optional argument they can take, 'plot' can – e.g. 'col', 'lwd', 'lty', 'xlab', 'ylab', 'main', etc.

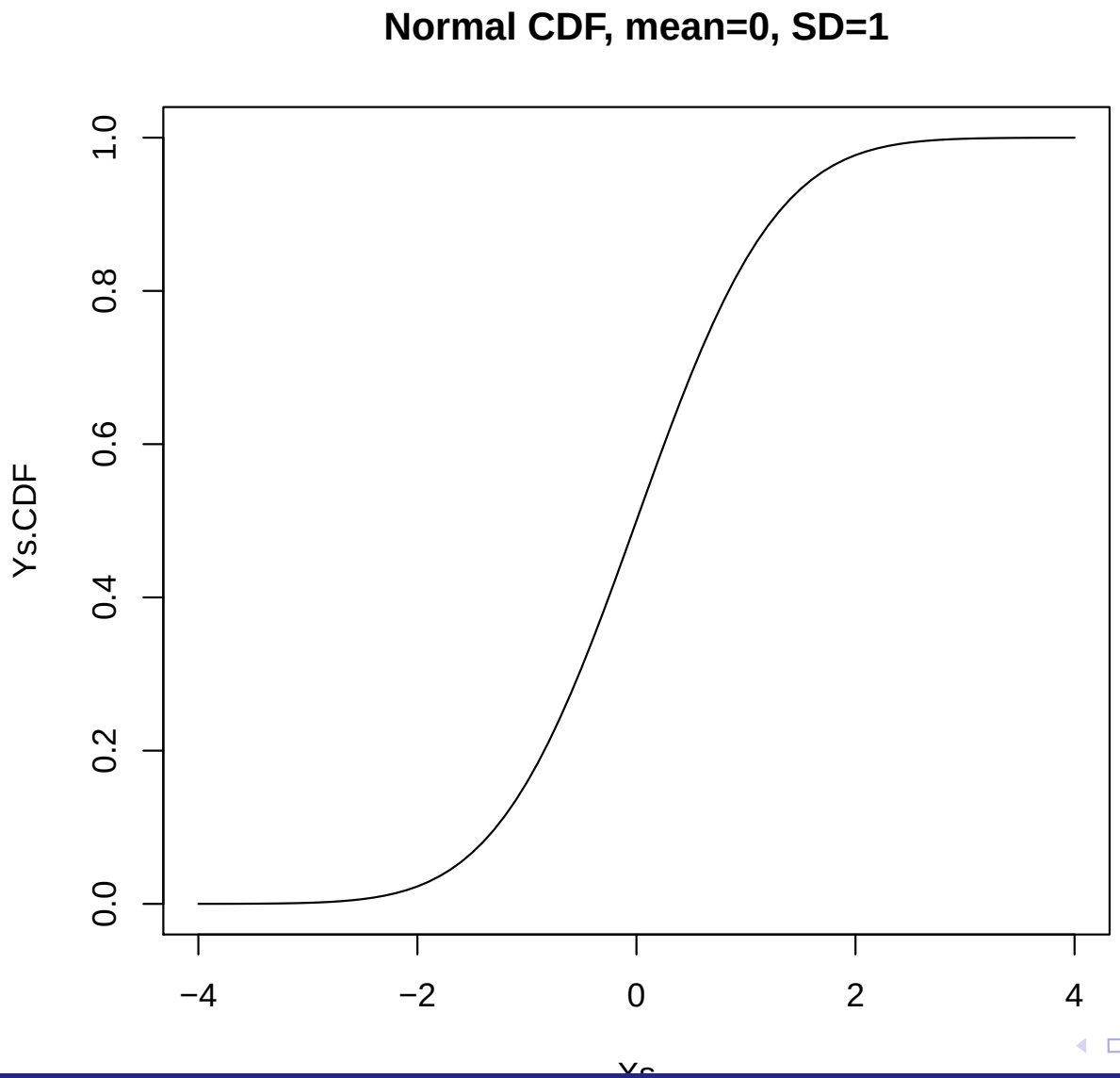
Example 3

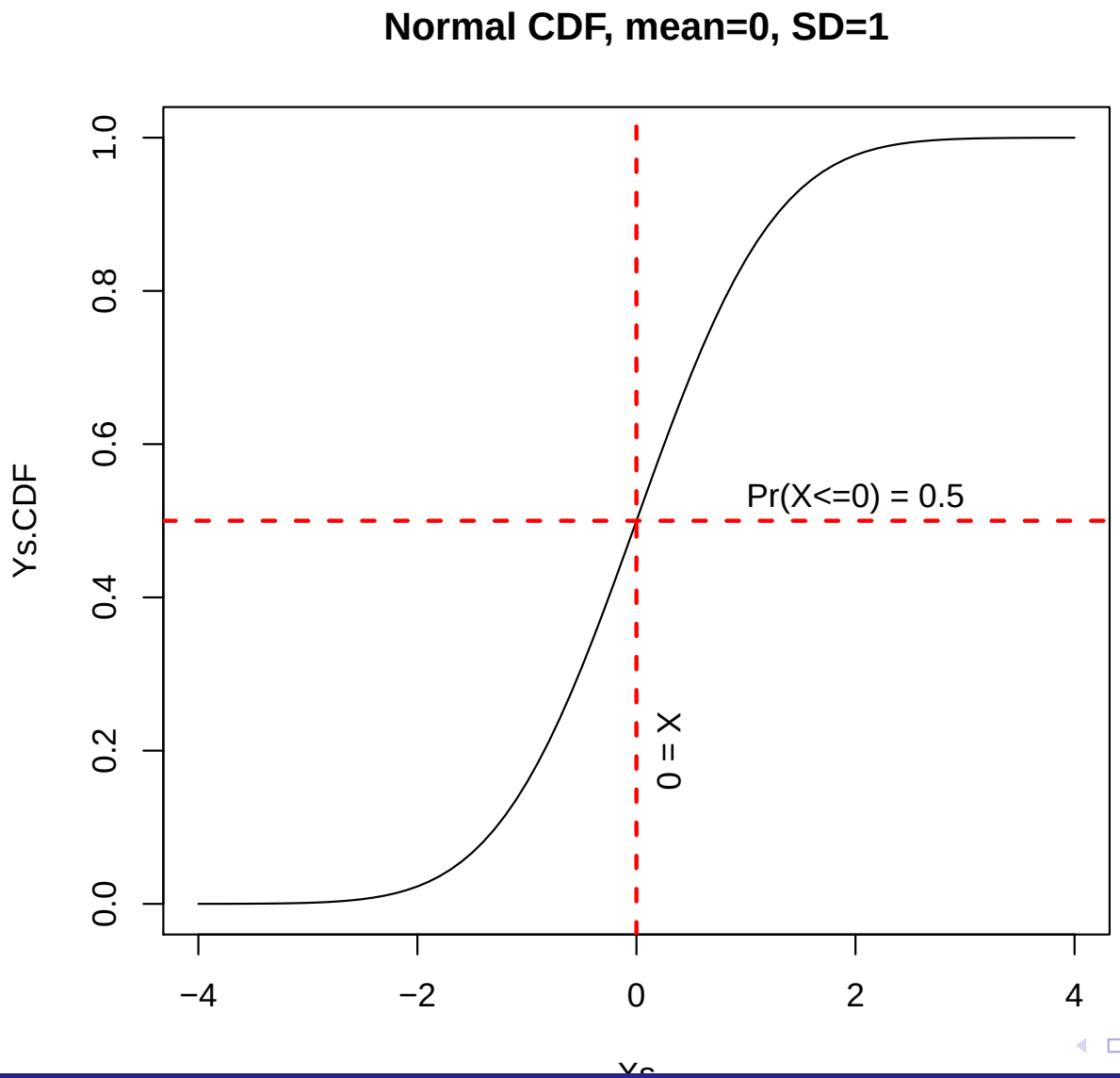


Example 3


Normal PDF, mean=0, SD=1
Ys.PDF
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
1/sqrt(2*3.142)*exp(−.5*(−1−0)^2
This area = the Pr(−1<=X<=0)
x = −1
Area under curve = 1

Example 3




Normal CDF, mean=0, SD=1
Ys.CDF
0.0
0.2
0.4
0.6
0.8
1.0
Pr(X<=0) = 0.5
0 = X
−4
−2
0
2
4

Parameterizing the Normal distribution

As you've seen from the '_norm' functions, we need two numbers to define a normal distribution – the mean and the SD

- The _norm functions each took another argument...
- ...but that extra value was used to do something with the distribution that was specified by mean and SD.

The mean and SD are called the **parameters** of the normal distribution

- We've been calling them 'arguments' since we've been thinking about them as values to pass to a programming function
- This is not incorrect, but in the context of statistics, 'parameter' is the better label.

In prose, we say "Data X [follow a normal distribution]/[is normally distributed] with mean μ and SD σ."

We can also write this more compactly:

$$\text{Data X} \sim N(\mu, \sigma)$$

- The '$\sim$' symbol is read 'is distributed'
- N signifies the normal distribution
- μ is read 'mu' and signifies the mean
- σ is read 'sigma' and signifies the SD
- If $\mu = 0$ & $\sigma = 1$, we have the **standard normal distribution**.

To recap: our goals for this theory/math heavy lecture

- Gain a rudimentary understanding of (one of) the most important distributions in statistics
- Lay groundwork for next lecture where...
- we will acquire the tools necessary to rigourously test the assertion that the turnout data are normal