

PSTAT 5LS Lab 3

YOUR NAME HERE

Week of April 22, 2024

Section 1

Announcements & Recap

Announcements

Insert any announcements that you have here

Upcoming due dates:

- Homework 3 due by 11:59pm on Friday, April 26
- ECoach Journal Entry 4 due by 11:59pm on Monday, April 29
- Extra credit! Project CLASSIFIES Task 1 (see Canvas assignments) due by 11:59pm on Monday, April 29

Coming up:

- Exam 1 on Tuesday, April 30 during lecture
- Slide Sets 1-4
- Homework assignments 1-3
- Exam 1 information on Canvas

Section 2

Learning Objectives

R Learning Objectives

- ① Learn how to visualize the normal distribution using `plot_norm()`
- ② Learn how to use `pnorm()` to find probabilities under the normal curve
- ③ Learn how to use `qnorm()` to find values of a normally distributed variable with specified probability to the left or the right

Statistical Learning Objectives

- 1 Understand how area under the normal curve relates to probability
- 2 Understand how to move between probabilities and quantiles of the normal distribution

Functions covered in this lab

- 1 `plot_norm()`
- 2 `pnorm()`
- 3 `qnorm()`

Section 3

Lab Tutorial

Normal Distributions

A “distribution” refers to the possible values a random variable can take as well as the probability that it takes those values. It is commonly used to approximate all sorts of things in nature and life.

A normal distribution is completely described by just two numbers: the **mean** and the **standard deviation**. The mean and standard deviation are called *parameters* of the distribution.

We denote a normal distribution by $N(\mu, \sigma)$, where μ is the population mean and σ is the population standard deviation.

Normal distributions are all bell-shaped, unimodal, and symmetric about their means, regardless of the values of the mean (μ) and the standard deviation (σ).

The mean μ specifies the *center* of the distribution. The standard deviation σ specifies the *variability* of the distribution (meaning, how narrow or wide it is).

The Standard Normal Distribution

Early statisticians did not have the technology we have at our disposal to find probabilities for variables that have normal distributions. They found probabilities under the normal curve by standardizing their variables as follows.

If a random variable X has a $N(\mu, \sigma)$ distribution, then the random variable

$$Z = \frac{X - \mu}{\sigma}$$

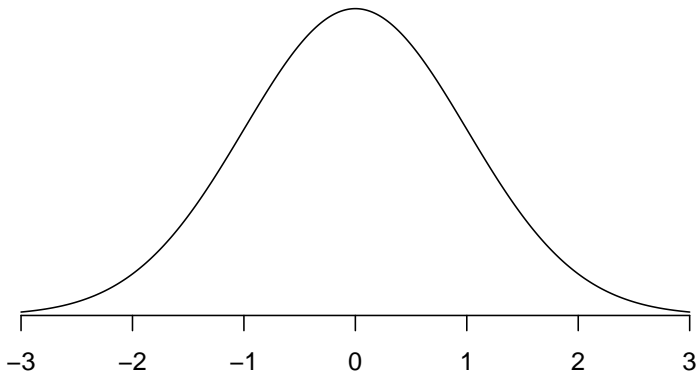
has the **standard normal** $N(0, 1)$ distribution, which is a normal distribution with mean 0 and standard deviation 1.

Then they used a standard normal table to estimate probabilities.

The Standard Normal Distribution

Let's take a look at the standard normal distribution.

$N(0, 1)$ Distribution



Visualizing a Normal Distribution with `plot_norm()`

The `stats250sbi` package that we are using includes the `plot_norm()` function to help you create a graphical display of a normal distribution. You will need to send the function the following arguments:

- `mean`: the mean of the normal distribution you'd like to draw (μ)
- `sd`: the standard deviation or standard error of the normal distribution you'd like to draw (σ or $\sqrt{\frac{p_0(1-p_0)}{n}}$, respectively)
- `shadeValues` (optional): either a number or a vector of two numbers (using `c()`) that are the boundaries of the region you'd like to shade.
- `direction`: where to shade ("less", "greater", "between", or "beyond")
- `col.shade`: the color to use when shading
- any other graphical parameters you want to use to control the appearance of the plot (like `main`, etc.)

Example: Pelican Wingspans

Recall from lecture that the distribution of wingspans in the population of California brown pelicans can be described by a normal distribution with mean 6.5 feet and standard deviation 0.5 feet.

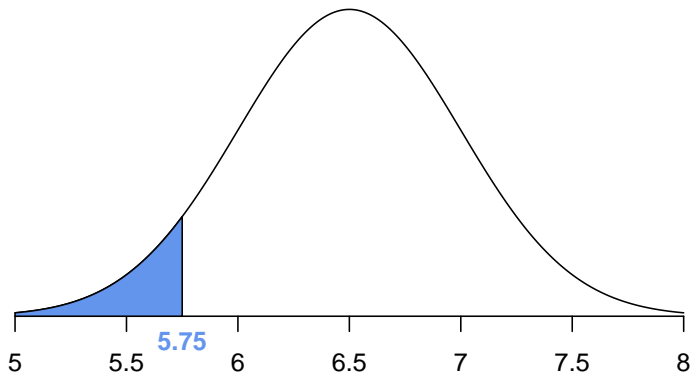
Example: Pelican Wingspans, part a

Let's recreate the images from the first three parts of the pelican example from the lecture notes.

In part a, we are interested in the probability that a randomly selected California brown pelican will have a wingspan of at most 5.75 feet. The visualization using `plot_norm()` is

```
plot_norm(mean = 6.5,  
          sd = 0.5,  
          shadeValues = 5.75,  
          direction = "less",  
          col.shade = "cornflowerblue")
```

Example: Pelican Wingspans, part a

 $N(6.5, 0.5)$ Distribution

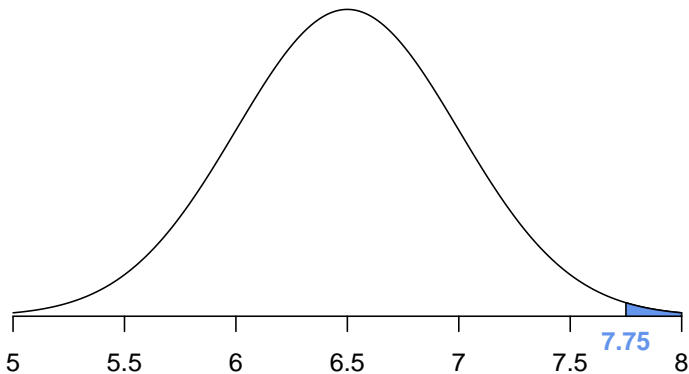
Example: Pelican Wingspans, part b

In part b, we are interested in the probability that a randomly selected California brown pelican will have a wingspan of at least 7.75 feet. The visualization using `plot_norm()` is

```
plot_norm(mean = 6.5,  
          sd = 0.5,  
          shadeValues = 7.75,  
          direction = "greater",  
          col.shade = "cornflowerblue")
```


Example: Pelican Wingspans, part b

$N(6.5, 0.5)$ Distribution

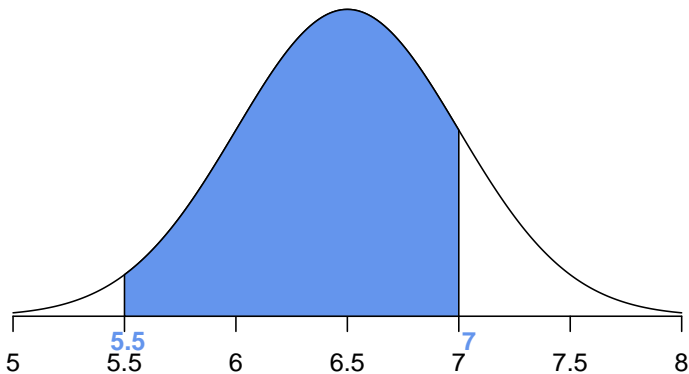


Example: Pelican Wingspans, part c

In part c, we are interested in the probability that a randomly selected California brown pelican will have a wingspan between 5.5 and 7 feet. The visualization using `plot_norm()` is

```
plot_norm(mean = 6.5,  
          sd = 0.5,  
          shadeValues = c(5.5, 7),  
          direction = "between",  
          col.shade = "cornflowerblue")
```

Example: Pelican Wingspans, part c

 $N(6.5, 0.5)$ Distribution

Finding Probabilities with the `pnorm()` Function

The `pnorm()` function gives us a way to compute probabilities when a variable has a normal distribution. The arguments you need to send to `pnorm()` are:

- `q`: the quantile (value on the x-axis) for the normal distribution
- `mean`: the mean of the normal distribution (μ)
- `sd`: the standard deviation of the normal distribution (σ)
- `lower.tail`: set to `'TRUE'` initially, signifying that R will compute the probability **to the LEFT** of `q`; if you would like R to compute the probability *to the right* of `q`, set `lower.tail` to `FALSE`

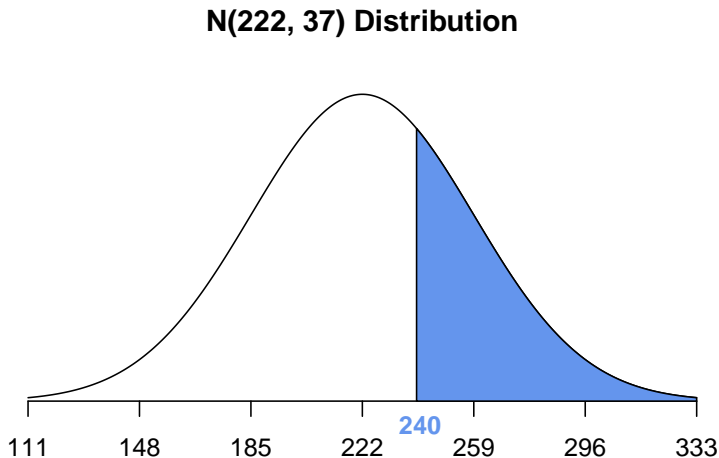
Example: Cholesterol in Middle-aged Men

The blood cholesterol levels of men age 55 to 64 are approximately normal, with mean 222 milligrams per deciliter (mg/dL) and standard deviation 37 mg/dL.

What percent of middle-aged men have high cholesterol (levels above 240 mg/dL)?

Example: Cholesterol in Middle-aged Men

Here is a visual display of the area we are interested in calculating:



Example: Cholesterol in Middle-aged Men

What percent of middle-aged men have high cholesterol (levels above 240 mg/dL)?

```
pnorm(q = 240,  
      mean = 222,  
      sd = 37,  
      lower.tail = FALSE)
```

```
## [1] 0.3133111
```

Note that we set `lower.tail` to **FALSE** because we need to find the area under the curve for cholesterol levels of 240 mg/dL or **more**.

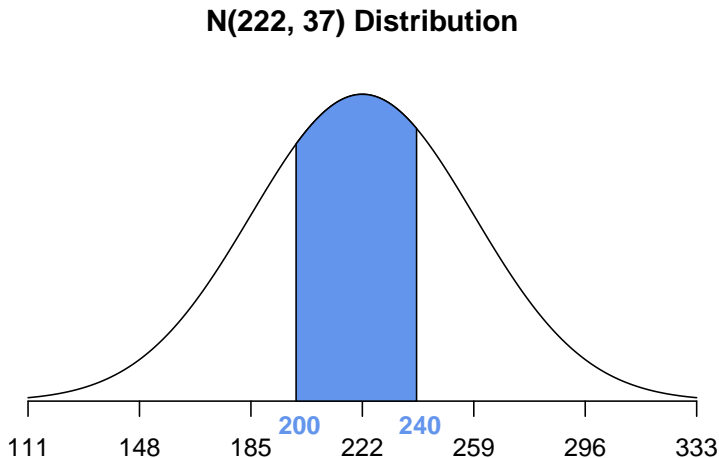
Example: Cholesterol in Middle-aged Men

The blood cholesterol levels of men age 55 to 64 are approximately normal, with mean 222 milligrams per deciliter (mg/dL) and standard deviation 37 mg/dL.

What percent have elevated cholesterol (between 200 and 240 mg/dL)?

Example: Cholesterol in Middle-aged Men

Here is a visual display of the area we are interested in calculating:



Example: Cholesterol in Middle-aged Men

What percent have elevated cholesterol (between 200 and 240 mg/dL)?

To find this, we need to take the area to the left of 240 mg/dL and subtract the area to the left of 200 mg/dL:

```
pnorm(q = 240, mean = 222, sd = 37, lower.tail = FALSE) -  
  pnorm(q = 200, mean = 222, sd = 37, lower.tail = FALSE)
```

```
## [1] -0.4106316
```

Note that we set `lower.tail` to `TRUE` because we need to find the area to the left in both cases. (You can also leave off the `lower.tail` argument if you want the area to the left, since `lower.tail = TRUE` is the default.)

Example: Cholesterol in Middle-aged Men

The blood cholesterol levels of men age 55 to 64 are approximately normal, with mean 222 milligrams per deciliter (mg/dL) and standard deviation 37 mg/dL.

Men in the 95th percentile have blood cholesterol levels of what value?

Um, wait. We can't do this with `pnorm()`!

Finding Values of the Variable with the `qnorm()` Function

The `qnorm()` function gives us a way to find the values of a normally distributed variable when you are given a probability. The arguments you need to send to `qnorm()`:

- `p`: the probability or area under the curve you want to find an x-axis value for
- `mean`: the mean of the normal distribution, defaults to 0
- `sd`: the standard deviation of the normal distribution, defaults to 1
- `lower.tail`: determines whether `qnorm()` finds the value of the variable with area `p` to its left or right. If `lower.tail` is set to `'TRUE'` (the default), the area `p` is to the **LEFT**. If `lower.tail` is set to `'FALSE'`, the area `p` is to the **RIGHT**.

Example: Cholesterol in Middle-aged Men

The blood cholesterol levels of men age 55 to 64 are approximately normal, with mean 222 milligrams per deciliter (mg/dL) and standard deviation 37 mg/dL.

Men in the 95th percentile have blood cholesterol levels of what value?

Um, wait. We can't do this with `pnorm()`!

Example: Cholesterol in Middle-aged Men

We can do this one of two ways:

- 1 $p = 0.95$ and `lower.tail = TRUE` tells R that we want the cholesterol level that has area 0.95 to the left

```
qnorm(p = 0.95,  
      mean = 222,  
      sd = 37,  
      lower.tail = TRUE)
```

- 2 $p = 0.05$ and `lower.tail = FALSE` tells R that we want the cholesterol level that has area 0.05 to the right

```
qnorm(p = 0.05,  
      mean = 222,  
      sd = 37,  
      lower.tail = FALSE)
```

Example: Cholesterol in Middle-aged Men

Hopefully it doesn't surprise you that the probabilities are the same:

1

```
qnorm(p = 0.95,  
      mean = 222,  
      sd = 37,  
      lower.tail = TRUE)
```

```
## [1] 282.8596
```

2

```
qnorm(p = 0.05,  
      mean = 222,  
      sd = 37,  
      lower.tail = FALSE)
```

```
## [1] 282.8596
```