Stats 10 Lab 4: Normal Random Variable, Sampling Distribution, and the Central Limit Theorem

Objectives

1. Calculate theory-based probabilities for normal distributions
2. Reinforce understanding of simulating and random sampling
3. Demonstrate the Central Limit Theorem's application

Section 1. The Normal Random Variable

Applet for the normal density curve:
https://digitalfirst.bfwpub.com/stats_applet/stats_applet_7_norm.html

The purpose of this interactive activity is to:

- get a better feel for the behavior of normal random variables.
- check how accurate the Standard Deviation Rule is.
- do some other calculations for the normal distribution that are beyond what the Empirical Rule can help us with.

As we've seen, the Empirical Rule for normal random variables is very useful. The following applet will help you see how accurate the rule is and do some other calculations for normal distributions that cannot be done using that rule and that will further enhance your understanding of the normal distribution.

Activity:

1. Set the mean to 0 and the standard deviation to 1.
2. The numbers on the horizontal axis represent the number of SD above or below the mean. So, 0 is the mean, +1 is one SD above the mean, -1 is one SD below the mean etc.

To see how accurate the Empirical Rule is, drag one flag across the other, so that the applet shows the area under the curve between the two flags.

Exercise 1

a. Place the flags 1 standard deviation on either side of the mean. What is the area between these two values? What does the empirical rule say this area is?
b. Repeat for 2 and 3 standard deviations on either side of the mean. Again, compare the empirical rule with the area given in the applet.

If you drag the flags across each other again, you'll be able to see the probabilities in the tails (above and below 1, 2, and 3 standard deviations from the mean). Recall that we mentioned that according to the Empirical Rule, these tails have probabilities .16, .025 and .0015, corresponding to 1, 2 and 3 standard deviations from the mean). Try it.

Recall from the EDA section the quartiles Q1 and Q3. In the context of random variables, Q1 is the value such that P(X < Q1) = .25. In other words, the random variable has a 25% chance of having a value that is below Q1. Similarly, Q3 is the value such that P(X < Q3) = .75. In other words, the random variable has a 75% chance of having a value below Q3. Another way to think about the quartiles is that they mark the highest and lowest 25% of the distribution.

Exercise 2
Using the applet, how many standard deviations above and below the mean do the quartiles of any normal distribution lie? Use the closest available values (the applet can't hit every value exactly).

Section 2. Finding Probabilities in Normal Distribution

The following activity will show you how to solve word problems involving the normal distribution. There are two types of problems that are of interest: finding probabilities given values, and finding values given probabilities. Below are the instructions for both types.

Finding Probabilities (Given Values)
The pnorm() function computes probabilities from a normal distribution with specified mean and standard deviation. The function inputs a value in the q argument and computes the probability that a value drawn from a normal distribution will be less than or equal to q. The exact normal distribution to compare with is specified using the mean and sd arguments. The default mean is mean = 0 and the default standard deviation is sd = 1.

Example:
Suppose X is the SAT-M score which has a normal distribution with a mean of 507 and standard deviation of 111.
To find the probability P(X < 700), enter the command in R:

```
pnorm(700, mean=507, sd=111)
```

You should see that the probability is equal to 0.9589596. To calculate the probability P(X > 700) = 1 - P(X < 700), enter

```
1- pnorm(700, mean=507, sd=111)
```

The optional argument lower.tail inputs a logical value (TRUE or FALSE) and changes the direction of the probability. The default is lower.tail = TRUE, so the pnorm() function will, as noted above, compute the probability that a value drawn from the normal distribution will be less than or equal to q. If we set lower.tail = FALSE, then the pnorm() function will compute the probability that a value drawn from the normal distribution will be greater than or equal to q.

Example:

```
pnorm(700, mean = 507, sd = 111, lower.tail = FALSE)
```

This gives you the exact same value as 1- pnorm(700, mean=507, sd=111)

If we wanted to find P(400 < X < 600), we would need to do two separate calculations; one for P(X < 600), and one for P(X < 400), and subtract.

Exercise 3
Adult male height (X) follows (approximately) a normal distribution with a mean of 69 inches and a standard deviation of 2.8 inches.

a. What proportion of males are less than 65 inches tall? In other words, what is P(X < 65)?
b. What proportion of males are more than 75 inches tall? In other words, what is P(X > 75)?
c. What proportion of males are between 66 and 72 inches tall? In other words, what is P(66 < X < 72)?

Finding Value (Given Probability)

This time, we're given the population mean and standard deviation, but instead of being given an X and asked to find the probability, we're given a probability and asked to find the corresponding X value.

The qnorm() function aims to do the opposite of pnorm(). The function inputs a value in the p argument and computes a value that determines this area from a normal distribution with specified mean and standard deviation.

We will illustrate with the previous example: Suppose X is the SAT-M score which has a normal distribution with a mean of 507 and standard deviation of 111.

Enter the probability calculated from previous activity for P(X < 700).

```
qnorm(0.9589596, mean=507, sd=111)
```

You should see that the output value is exactly 700.
Now let's find the value of x that satisfies P(X > x) = 0.02 where X is the SAT-M score, which has a normal distribution with a mean of 507 and standard deviation of 111. Before we start, it will be useful to rephrase the problem in terms of X < x; we are looking for the value of x that satisfies P(X < x) = 0.98.

To find the X value, enter the command:

```
qnorm(0.98, mean=507, sd=111)
```

R tells us that the value that we are looking for (the 98th percentile) is 734.966.
Alternatively, you can use the lower.tail option in qnorm()

Exercise 4
Suppose adult male height follows a normal distribution with a mean of 69 inches and a standard deviation of 2.8 inches.

a. How tall must a male be in order to be among the shortest 0.5% of males?
b. How tall must a male be in order to be among the tallest 0.25% of males?

Section 3. Simulations of samples and sample distributions using a for loop

We want to illustrate the sampling variability (also called sample-to-sample variability) of the sample mean and the sample proportion. That is, when we take different random samples, how does the sample mean of the arsenic levels vary from sample to sample? How does the sample proportion of households who experienced a new health issue vary from sample to sample?

We can simulate sampling many (1000) random samples of size 30 from the population of households in the Pawnee data using a for loop. For each random sample, we can compute the mean arsenic levels and the proportion of the sample who experience a new health issue.

Here the code is a for loop to simulate the sample proportions from 1000 random samples of size 30. Please carefully read the comments for each line to understand the code. We will also discuss the code in lab section.

```
# We first create objects for common quantities we will use for this
exercise.
n <- 30 # The sample size
N <- 541 # The population size
M <- 1000 # Number of samples/repetitions
# Create vectors to store the simulated proportions from each repetition.
phats <- numeric(M) # for sample proportions
# Set the seed for reproduceability
set.seed(123)
# Always set the seed OUTSIDE the for loop.
# Now we start the loop. Let i cycle over the numbers 1 to 1000 (i.e.,
iterate 1000 times).
for(i in seq_len(M)){
# The i-th iteration of the for loop represents a single repetition.
# Take a simple random sample of size n from the population of size N.
index <- sample(N, size = n)
# Save the random sample in the sample_i vector.
sample_i <- pawnee[index, ]
# Compute the proportion of the i-th sample of households with a new health
issue.
phats[i] <- mean(sample_i$New_hlth_issue == "Y")
}
```

Note that the replicate() function from the last lab could have been used here, but for loops are much more versatile and can be used in a wider variety of settings.

Exercise 5 - Proportions

a. Run the entire chunk of code in the previous page to run a for loop that creates a vector of sample proportions. Using the results, create a relative frequency histogram of the sampling distribution of sample proportions.

Superimpose a normal curve to your histogram with following instructions:

- If you use the histogram() function from the mosaic package, add the argument: fit = "normal".
- If you use the hist() function from base R, add the argument: prob = TRUE, then run the command: curve(dnorm(x, mean(phats), sd(phats)), add = TRUE).

b. What is the mean and standard deviation of the simulated sample proportions?

c. Do you think the simulated distribution of sample proportions is approximately normal? Explain why or why not.

d. Using the theory-based method (i.e., normal approximation by invoking the Central Limit Theorem), what would you predict the mean and standard deviation of the sampling distribution of sample proportions to be? How close are these predictions to your answers from Part B?