Johanna Ravenhurst
Submitted: October 10, 2022
I did not work with other students.

ECO 634 - Lab 04: Uncertainty and Error

Q1 (1 pt): Show the code you used to create your vectors

```
norm_mean <- 10.4

norm_sd <- 2.4

norm_17 <- rnorm(n = 17, mean = norm_mean, sd = norm_sd)

norm_30 <- rnorm(n = 30, mean = norm_mean, sd = norm_sd)

norm_300 <- rnorm(n = 300, mean = norm_mean, sd = norm_sd)

norm_3000 <- rnorm(n = 3000, mean = norm_mean, sd = norm_sd)
```

Q2 (2 pts): Include the R code you used to create your figure. Your answer should include code that builds the figure as well as saves it to a file.

```
png(filename = here("lab_04_hist_01.png"), width = 1500, height = 1600, res=180)

par(mfrow = c(2, 2))

hist(norm_17, main = "Histogram of 17 Random Numbers", xlab = "Normally-Distributed Random Numbers")

hist(norm_30, main = "Histogram of 30 Random Numbers", xlab = "Normally-Distributed Random Numbers")

hist(norm_300, main = "Histogram of 300 Random Numbers", xlab = "Normally-Distributed Random Numbers")

hist(norm_3000, main = "Histogram of 3000 Random Numbers", xlab = "Normally-Distributed Random Numbers")

dev.off()
```

Q3 (4 pts): Upload your lab_04_hist_01.png file to Moodle. Make sure you double check the image size and resolution requirements.

Q4 (2 pts): Qualitatively describe the differences among the histograms.

The first histogram is of 17 randomly generated numbers, so it represents the smallest sample size of this group. The distribution is skewed to the right instead of the symmetrical normal distribution we might expect from the function we used to create this vector. The second histogram is of the sample of 30 randomly generated numbers and it looks closer to a normal distribution. The distribution looks almost symmetrical, but with a slight skew to the right. The last two histograms have larger sample sizes and therefore have much larger frequencies displayed in the bins. The range of numbers is more evenly distributed, and each histogram has a distribution that looks closer to a symmetrical normal distribution as the sample size of random numbers increases.

These vectors were created with a mean of 10.4 and a standard deviation of 2.4. The vectors with a larger sample size create a histogram that more clearly shows the mean and standard deviation because the values are evenly distributed.

Q5: Explain *why* the shapes of the histograms are different.

The shapes of the histograms are different because a smaller sample is less likely to end up with an even distribution of random numbers. This is due to sampling error because when we randomly select a few numbers it's unlikely that they will also be evenly distributed. We need to use a larger sample to have a better representation of the range of values in the population or in the distribution we are sampling from.

Q6: (1 pt) What are the parameters and their values for the *standard Normal* distribution?

The standard Normal distribution has a mean of 0 and a standard deviation of 1.

Q7: (2 pts) Include the R code you used to create your figure. Your answer should include code that builds the figure as well as saves it to a file.

```
x = seq(-2, 21, length.out = 1000)

y = dnorm(x, mean = 10.4, sd = 2.4, log=FALSE)

svg(filename = here("norm_1.svg"),

   width = 7, height = 7)

plot(x, y, main = "Normal PDF with Mean of 10.4\nStandard Deviation of 2.4", type = "l", xlim = c(4, 18))

abline(h = 0)
```

Q8: (2 pts.) Upload norm_1.svg (or norm_1.pdf).

Q9: (3 pts.) Show the R code you used to create one **of the random datasets in your figure.**

```
n_pts = 100

y_prob = 0.25

y_size = 50

set.seed(50)

# X values are normally-distributed, Y values follow binomial distribution

x_random_q9_2 = rnorm(n = n_pts)

y_random_q9_2 = rbinom(n = n_pts, size = y_size, prob = y_prob)

dat_random_q9_2 = data.frame(x = x_random_q9_2, y = y_random_q9_2)
```

```
plot(y ~ x, data = dat_random_q9_2, pch = 5,

    main = "Two Randomly Generated Samples",

    col =

      adjustcolor(col = "red", alpha.f =.9),

     xlab = "Random Normal Sample X",

     ylab = "Random Binomial Sample Y")
```

Q10: (2 pts.) Upload an image file of your figure. It may be in png (raster graphics), svg (vector graphics), or pdf (vector graphics) format.

Q11:Code for simulated data and fitted linear model

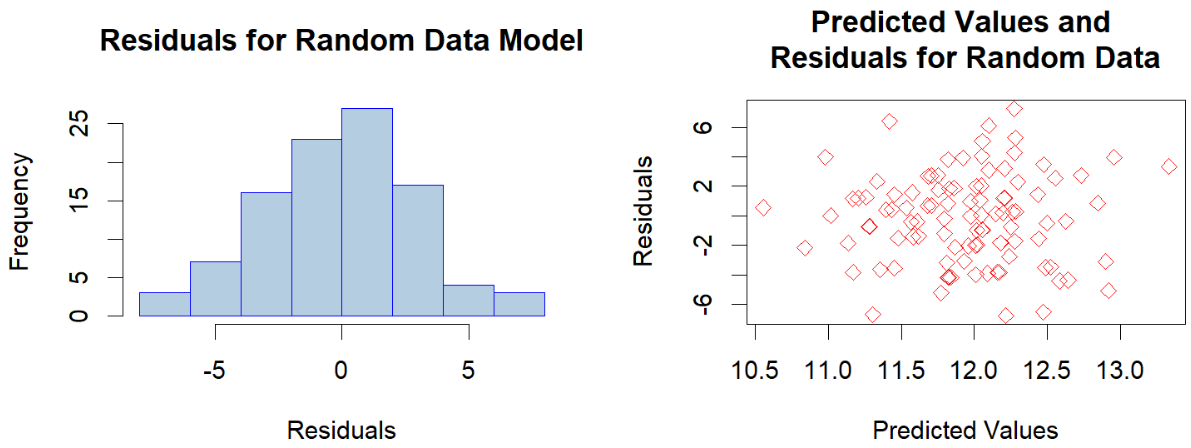
```
n_pts = 100
y_prob = 0.25
y_size = 50

set.seed(50)

x_random_q9_2 = rnorm(n = n_pts)

y_random_q9_2 = rbinom(n = n_pts, size = y_size, prob = y_prob)

dat_random_q9_2 = data.frame(x = x_random_q9_2, y = y_random_q9_2)

plot(y ~ x, data = dat_random_q9_2, pch = 5,

    main = "Two Randomly Generated Samples",

    col =

      adjustcolor(col = "red", alpha.f =.9),

     xlab = "Random Normal Sample X",

     ylab = "Random Binomial Sample Y")


guess_x = 0

guess_y = 12

guess_slope = 0.5

curve(line_point_slope(x, guess_x, guess_y, guess_slope), add = T)
```

Q12: Upload an image file

Q13: Random Data Model Residuals Code

```
#calculate the predicted yvalues based on my estimated model parameters

#add y_predicted variable as a new column in the dat_random dataset

dat_random_q9_2$y_predicted <- line_point_slope(dat_random_q9_2$x, guess_x, guess_y, guess_slope)


#calculate residuals and add to dat_random as a new column

#residuals are the difference between the predicted and observed values


dat_random_q9_2$resids <- dat_random_q9_2$y_predicted - dat_random_q9_2$y
```

Q14: Random Data Model Residual Plot


Residuals for Random Data Model
Frequency
Residuals
Predicted Values and
Residuals for Random Data
Residuals
Predicted Values