Hannah Korte
ECO 634
10/10/21

Lab 4

1. set.seed(1)
   n_17=17
   n_30 = 30
   n_300= 300
   n_3000= 3000
   vec_sd = 2.4
   vec_mean = 10.4
   norm_17 = rnorm(n = n_17, mean = vec_mean, sd = vec_sd)
   norm_30 = rnorm(n = n_30, mean = vec_mean, sd = vec_sd)
   norm_300 = rnorm(n = n_300, mean = vec_mean, sd = vec_sd)
   norm_3000 = rnorm(n = n_3000, mean = vec_mean, sd = vec_sd)

2. require(here)
   png(
     here("images", "lab_04_hist_01.png"),
     width= 1500, height=1600, res=180)

   par(mfrow=c(2,2))
   hist(norm_17, main= "17 Element Vector",xlab="Random Generated Points")
   hist(norm_30, main= "30 Element Vector",xlab="Random Generated Points")
   hist(norm_300, main= "300 Element Vector",xlab="Random Generated Points")
   hist(norm_3000, main= "3000 Element Vector",xlab="Random Generated Points")
   dev.off()

3. Upload of my lab_04_hist_01.png on moodle

4. The four histograms all have different shapes from each other due to their different sample sizes. The “17 Element Vector” histogram has a gap in the data, while the rest of them don’t. As the data points keep increasing, the amount of columns and the frequency range on the y-axis increase as well. The histograms with the sample sizes n=300 and n=3000 are more positively skewed and become more bell shaped, the histogram with n=3000 is more bell shaped though. The histogram with n=30 is negatively skewed and histogram n=17 has an odd shape but from what I can tell it is negatively skewed as well.

5. The shapes of the histograms are different due to the large variations in the sample sizes. The more randomly generated points there are the more likely a normal distribution will be present. In my histograms as the sample size increases the more symmetrical they become. The n=17 and n=30 histograms are skewed to the left. Then as you look at the n=300 histogram you can see it being skewed to the right but starting

to look more symmetrical and lastly when you look at the n=3000 histogram you can tell it is becoming more symmetrical than the last as the data points increase.

6. The parameters for a standard normal distribution are the number of observations to create, the mean, and the standard deviation. The values are mean = 0, standard deviation = 1, and the number of observances to create doesn't have a set value.

7.
```
require(here)
n_mean=10.4
n_sd=2.4
x=seq(-50,50,length.out = 1000)
y=dnorm(x,mean=n_mean,sd=n_sd)

pdf(
  here("images","norm_1.pdf"),
  width = 7,height = 7
  )
plot(x, y, main = "Normal PDF: Mean = 10.4, SD = 2.4", type = "l", xlim = c(3,20))
abline(h = 0)
dev.off()
```

8. Upload of my norm_1.pdf on moodle

9.
```
set.seed(223)
n_pts = 500
x3 = rnorm(n = n_pts, mean=7,sd=2)
y3 = rnorm(n = n_pts, mean=5,sd=1)
dat4= data.frame(x=x3,y=y3)
plot(dat4,col=15, main="dat4 Plot")
```

10. Upload of my random_data_1.pdf on moodle

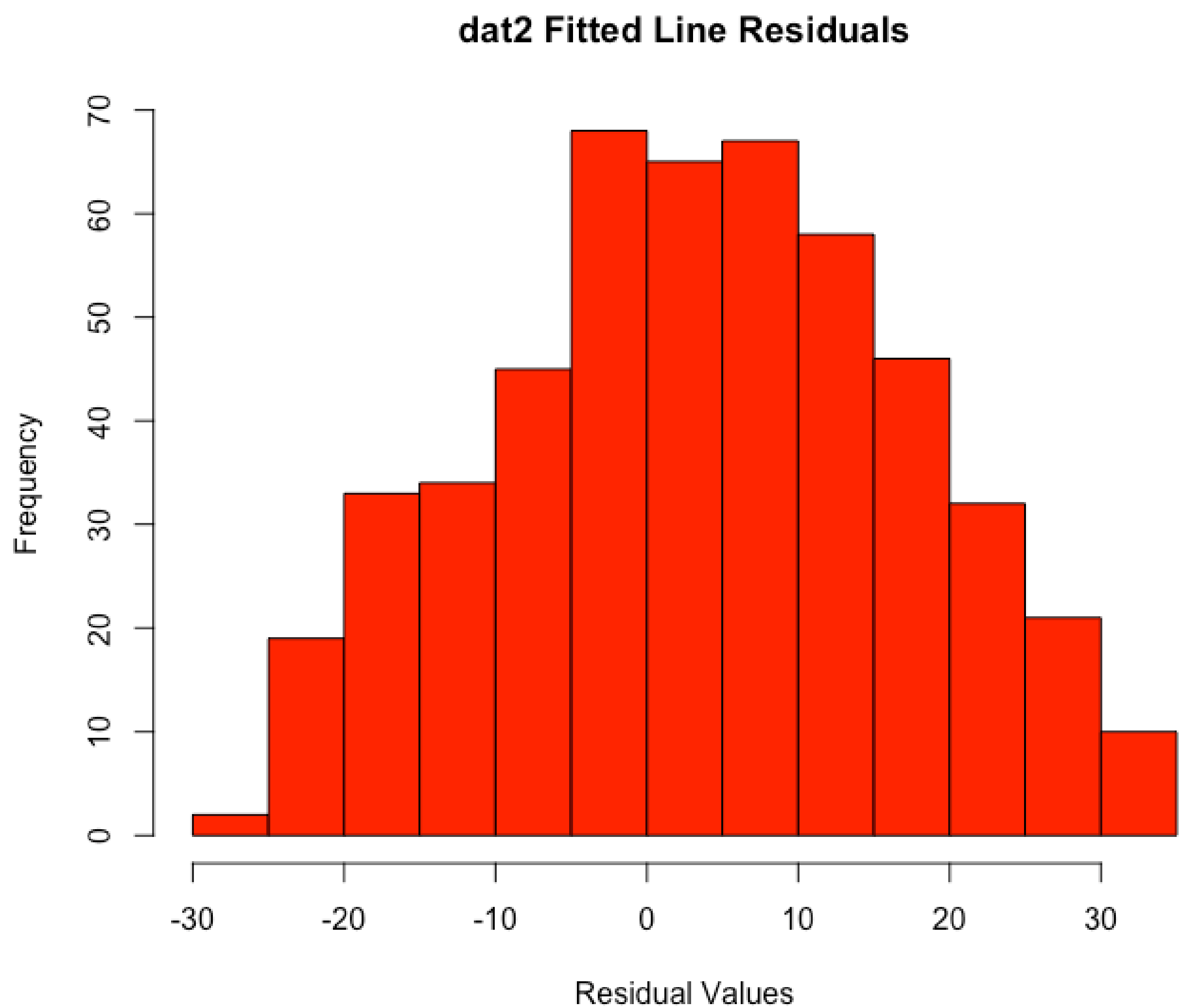
11.
```
set.seed(223)
n_pts = 500
x_min = 1
x_max = 35
x = runif(n = n_pts, min = x_min, max = x_max)
y = runif(n = n_pts, min = x_min, max = x_max)
dat2 = data.frame(x = x,y=y)
plot(dat2,col=2, main=  "dat2 Plot")
```

12. Upload of my random_data_2.pdf on moodle

13. #calculate and add predicted y-values
    dat2$y_predicted = line_point_slope(dat2$x, guess_x, guess_y, guess_slope)

    #calculate and add residuals
    dat2$resids = (dat2$y - dat2$y_predicted)

14. A histogram of the model's residuals



A scatterplot of your model's predicted values and residuals