Julian Burgoff

10/10/2022

Analysis of Environmental Data Lab

Lab 4

1. n_samples_1= 17
   n_samples_2= 30
   n_samples_3= 300
   n_samples_4= 3000
   pop_sd = 2.4
   pop_mean = 10.4

   dat_1 = rnorm(n = n_samples_1, mean = pop_mean, sd = pop_sd)
   dat_2 = rnorm(n = n_samples_2, mean = pop_mean, sd = pop_sd)
   dat_3 = rnorm(n = n_samples_3, mean = pop_mean, sd = pop_sd)
   dat_4 = rnorm(n = n_samples_4, mean = pop_mean, sd = pop_sd)

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

   par(mfrow = c(2, 2))

   hist(dat_1, main= "Histogram of 17 randomly generated numbers")
   hist(dat_2, main= "Histogram of 30 randomly generated numbers")
   hist(dat_3, main= "HIstogram of 300 randomly generated numbers")
   hist(dat_4, main= "Histogram of 3000 randomly generated numbers")
   dev.off()

3. PNG file attached

4. The histogram of 17 randomly generated numbers is relatively normally distributed, minus the big gap where the frequency should be highest around 11. The histogram of 30 numbers is skewed to the right and more uniformly distributed. The histogram of 300 numbers is much more normally distributed and the histogram of 3000 is almost perfectly normally distributed, where the histogram peaks at the mean and falls evenly on each side.

5. The shapes are different because of the different sample sizes represented in each histogram. As the sample size goes up, the random numbers get closer and closer to a normal distribution about the mean.

6. The parameters are the mean and standard deviation (mean=0, sd=1).

7. ```
   require(here)
   svg(
    filename = here("norm_1.svg"),
    width = 10, height = 10)

   x = seq(-100, 100, length.out = 1000)
   y = dnorm(x, mean= 10.4, sd= 2.4)

   plot(x, y, main = "Normal PDF with Mean = 10.4, SD = 2.4", type = "l", xlim = c(3, 19))
   abline(h = 0)
   dev.off()
   ```

8. norm_1.svg attached

9. ```
   n_pts = 111

   x_min = 0

   x_max = 10


   set.seed(1)
   dat_unif_1 = runif(n = n_pts, min = x_min, max = x_max)
   ```

10. quad_plot.svg attached

11. ```
    n_pts = 111
    x_min = 0
    x_max = 10

    x_random = runif(n = n_pts, min = x_min, max = x_max)

    y_random = rnorm(n = n_pts)

    dat_random = data.frame(x = x_random, y = y_random)
    ```

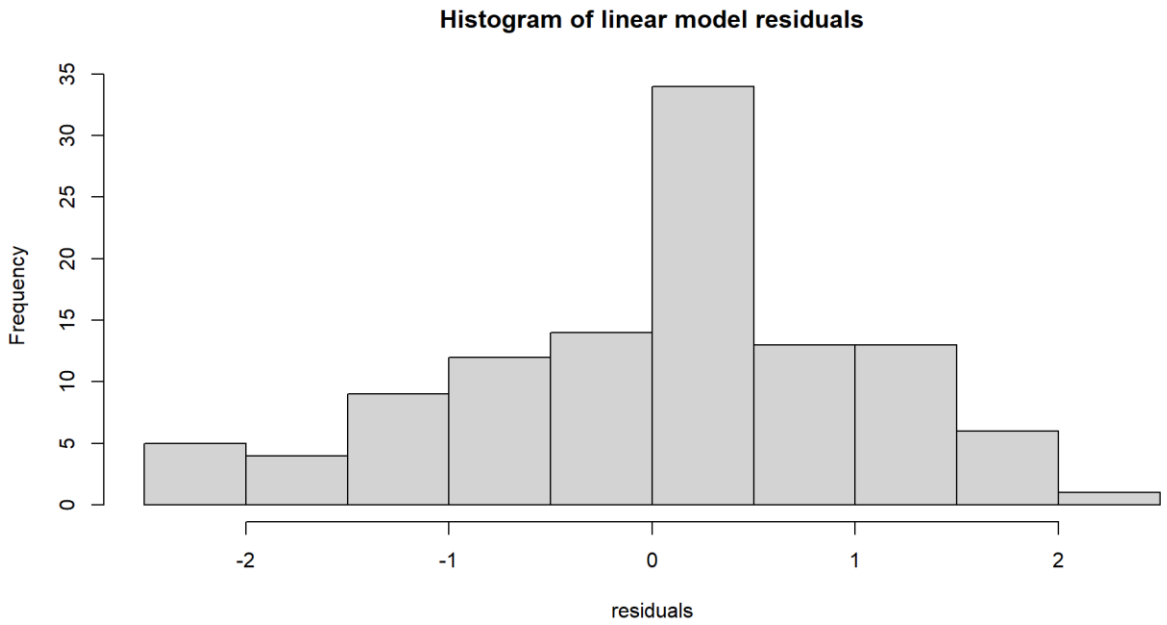
12. lin_function.svg attached
    ```

13. y_predicted= line_point_slope(dat_random$x, guess_x, guess_y, guess_slope)

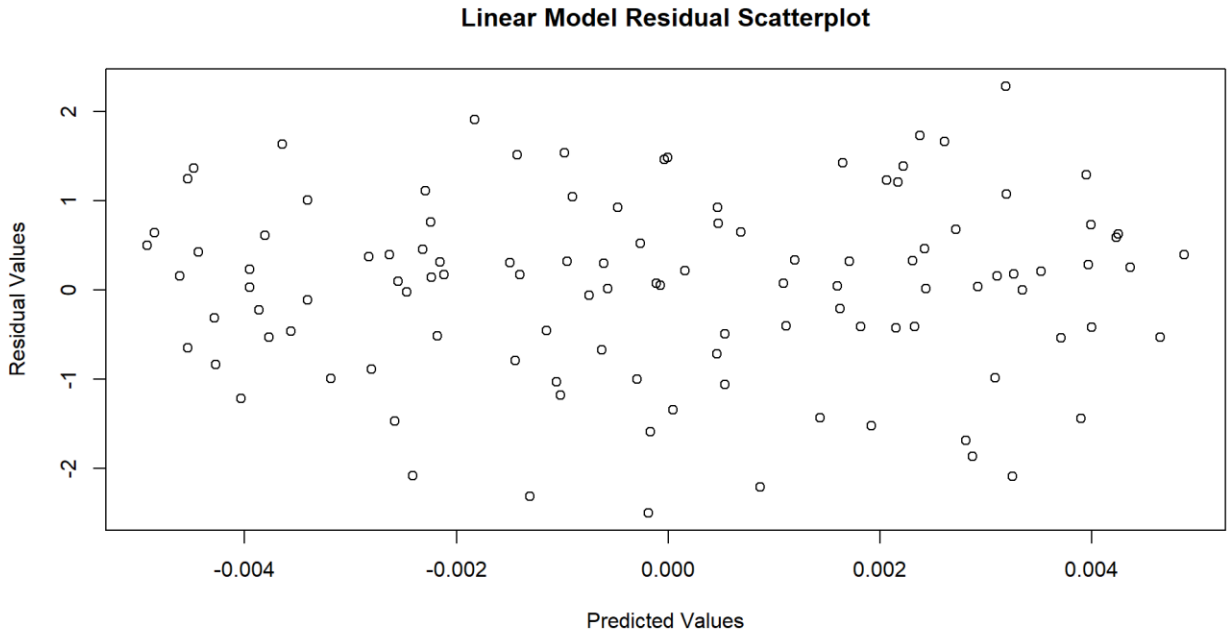
    dat_random$y_predicted<- y_predicted

    resids= dat_random$y_predicted- dat_random$y

    dat_random$resids<- resids



14. _____


Linear Model Residual Scatterplot
Residual Values
2
1
0
-1
-2
-0.004
-0.002
0.000
0.002
0.004
Predicted Values