Ian Eggleston

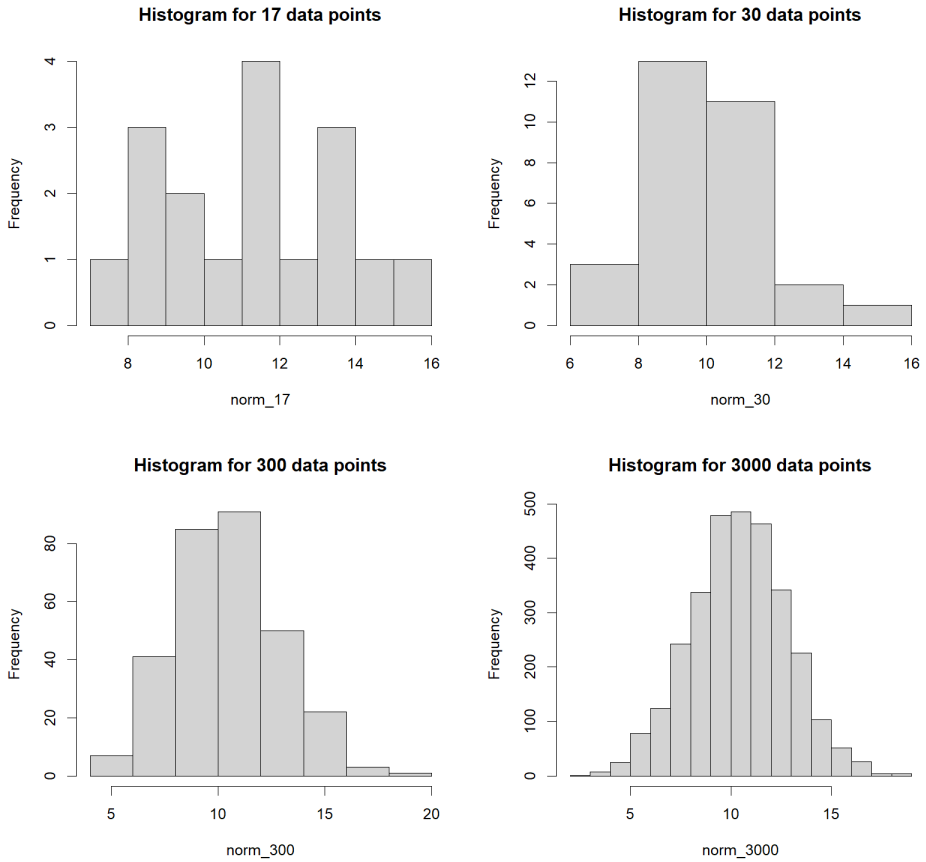
1.

```
norm_mean = 10.4
norm_sd = 2.4
norm_17 = rnorm(17, mean = norm_mean, sd = norm_sd)
norm_30 = rnorm(30, mean = norm_mean, sd = norm_sd)
norm_300 = rnorm(300, mean = norm_mean, sd = norm_sd)
norm_3000 = rnorm(3000, mean = norm_mean, sd = norm_sd)
```

2.

```
png(
  filename = here("lab_04_hist_01.png"),
  width = 1600, height = 1500,
  res=180
)
par(mfrow = c(2,2))
hist(norm_17, main = "Histogram for 17 data points")
hist(norm_30, main = "Histogram for 30 data points")
hist(norm_300, main = "Histogram for 300 data points")
hist(norm_3000, main = "Histogram for 3000 data points")
dev.off()
```

3. Uploaded png to moodle



4. The diagrams change from more random data points to a nicer looking spread of points, with a clear center and distribution. The first graph has a clear center, but the distribution around is not consistent. The second graph shows most data was close to the mean but then quickly dropped off at the ends of the graph. The third and fourth show a better distribution around the mean with more symmetrical plots.
5. The difference among the graphs is due to the number of sampling points. By increasing the number of points, the spread of data is better represented with fewer data points at the extremes and more centered around the mean. As the number of data entries increases, we should visually see the distribution of data form a more symmetrical curve.
6. The two parameters are mean and standard deviation. The values for standard normal deviation are: mean = 0 and std dev = 1.
7.

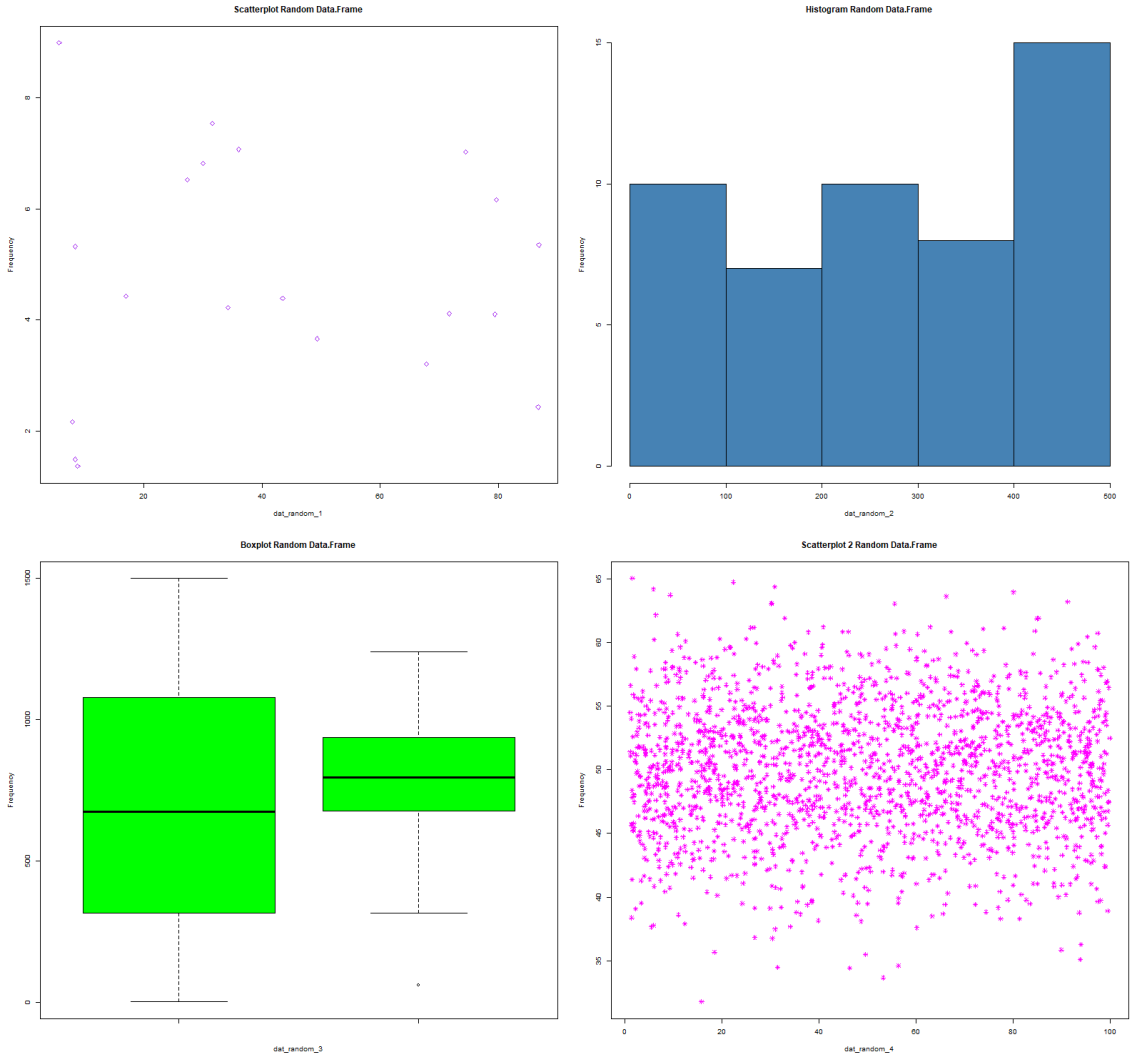
```
par(mfrow = c(1,1))
x = seq(0, 20, length.out = 1000)
y = dnorm(x, mean = 10.4, sd = 2.4)
svg(filename = "norm_1.svg",
    width = 60, height = 50)
plot(x, y, main = "Normal PDF: mean=10.4, SD=2.4", type = "l", xlim = c(0, 20))
abline(h = 0)
```

dev.off()

8. SVG uploaded
9.

```
x_random_4 = runif(n = 2000, min = 1, max = 100)
y_random_4 = rnorm(n = 2000, mean = 50, sd = 5)
dat_random_4 = data.frame(x = x_random_4, y = y_random_4)
plot(y ~ x, data = dat_random_4, pch = 8, main = "Scatterplot 2 Random Data.Frame",
    xlab = "dat_random_4", ylab = "Frequency", col = "magenta")
```

10. PNG uploaded

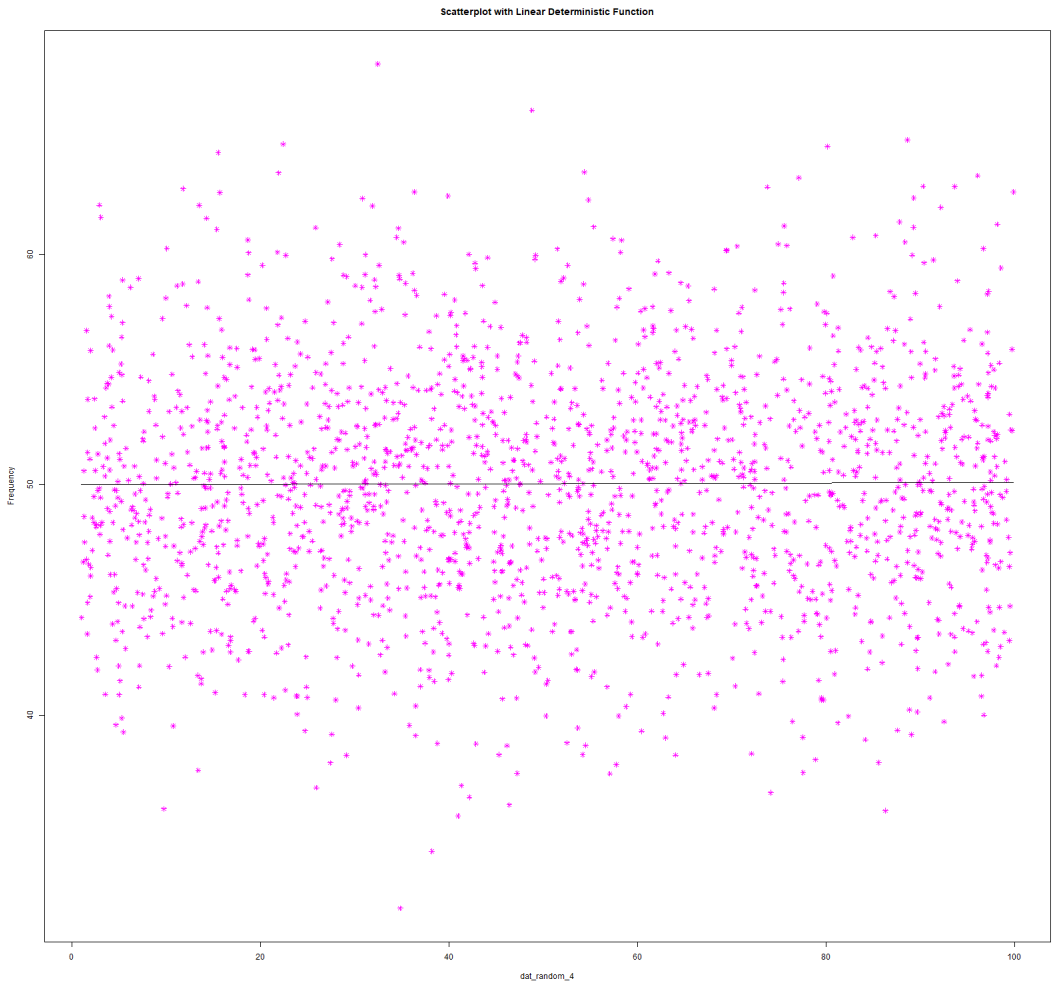


11. png(filename = here("lab_04_linear_function.png"),

    width = 1600, height = 1500
)

    x_random_4 = runif(n = 2000, min = 1, max = 100)
    y_random_4 = rnorm(n = 2000, mean = 50, sd = 5)
    dat_random_4 = data.frame(x = x_random_4, y = y_random_4)
    plot(y ~ x, data = dat_random_4, pch = 8, main = "Scatterplot with Linear Deterministic Function",
        xlab = "dat_random_4", ylab = "Frequency", col = "magenta")
    curve(line_point_slope(x, x1 = 0, y1= 50, slope = 0.001), add = TRUE)
    dev.off()

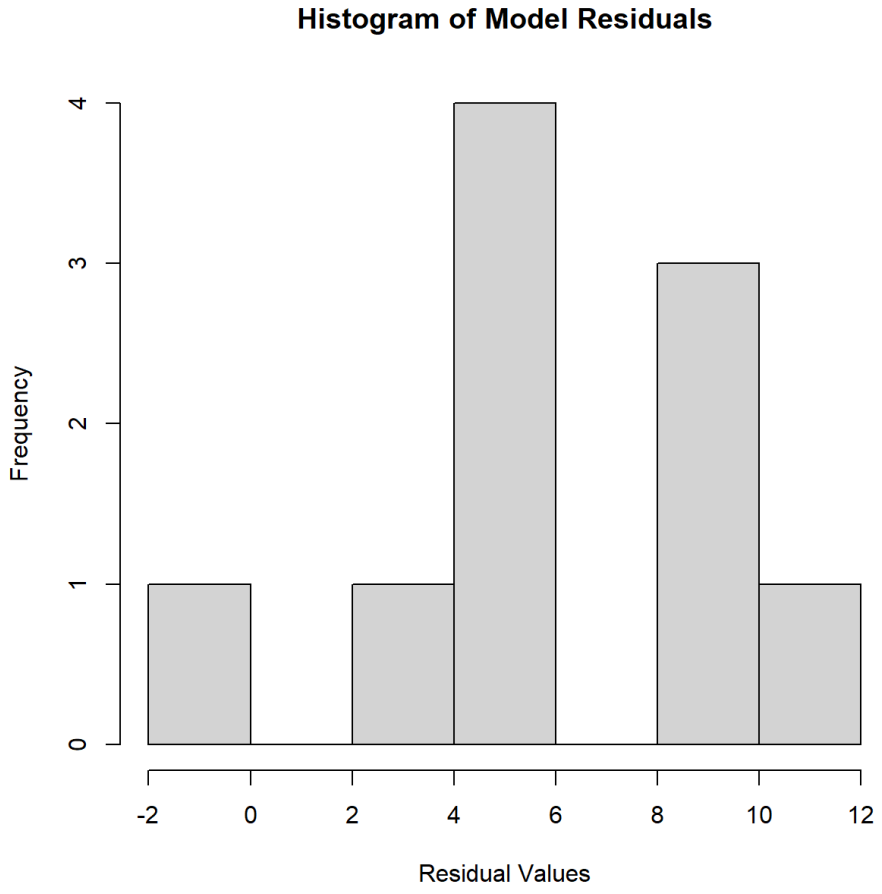
12. Png uploaded



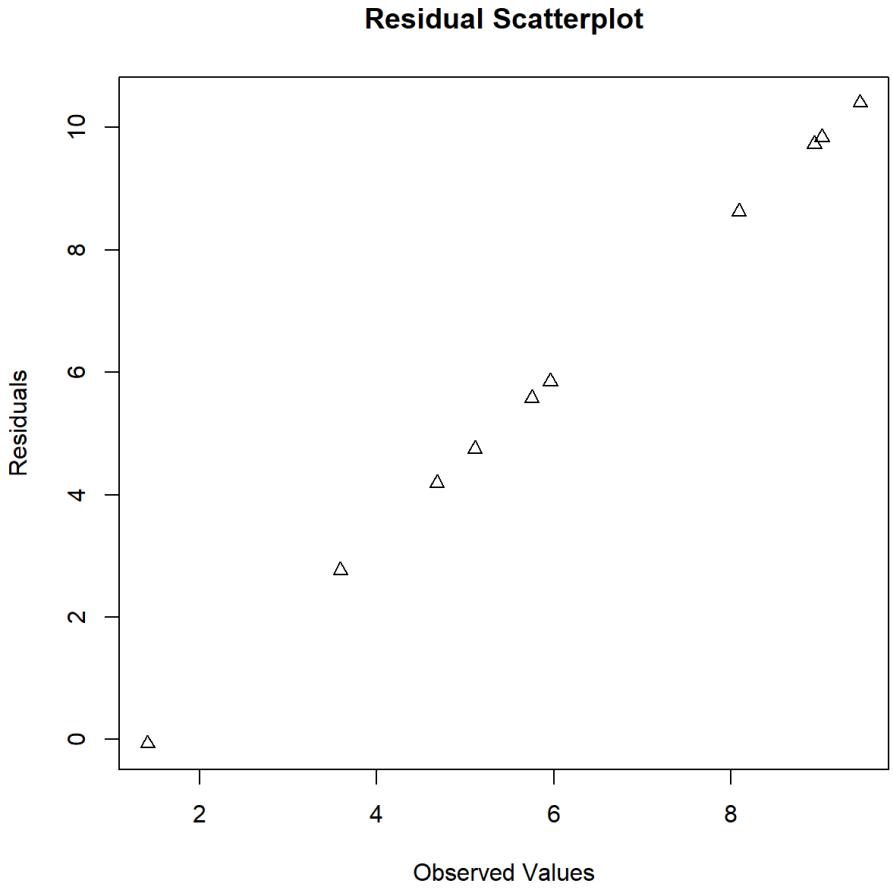
13.

```
predicted_values = line_point_slope(dat_random$x, guess_x, guess_y, guess_slope)
dat_random$predicted_values = predicted_values

resids = dat_random$x - dat_random$predicted_values
dat_random$resids = resids
```



14.


Residual Scatterplot
Residuals
0
2
4
6
8
10
2
4
6
8
Observed Values