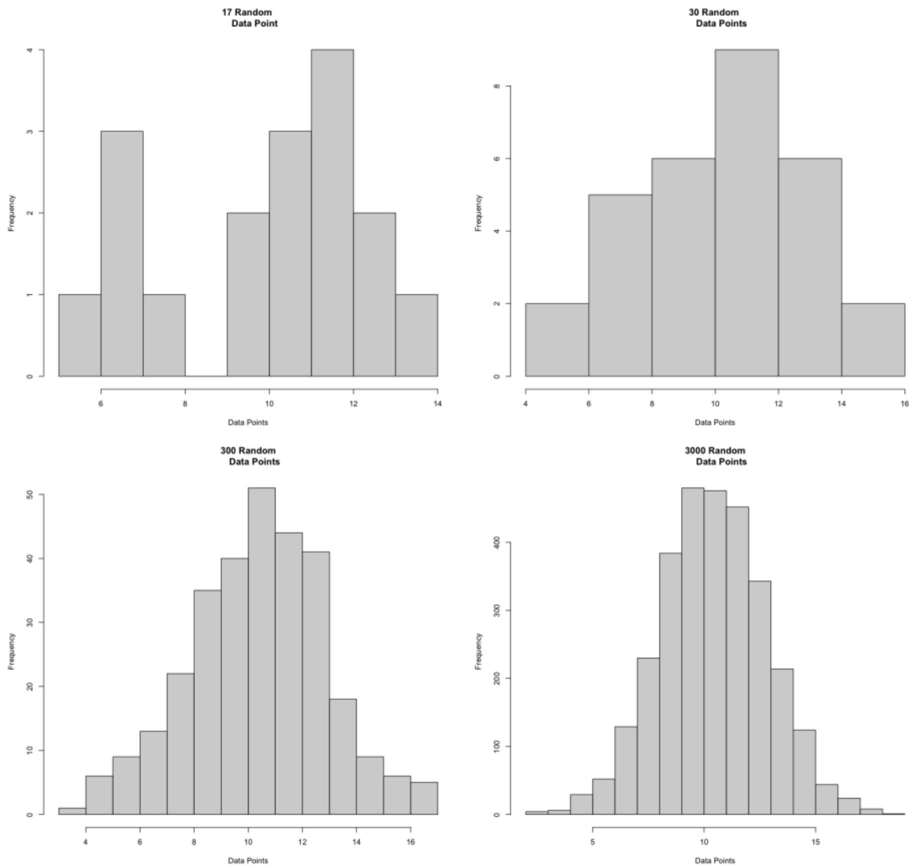
Q3



Q4 The first histogram of only 17 randomly generated values has a gap in its values, almost creating two separate normal distributions within the data. The histogram of 30 random values begins to take the shape of a normal distribution curve, showing a bell-shape with a slight right, or positive skewedness. The histogram with 300 values follows a a similar shape as the previous histogram but is more defined and contains more bin values. It is also slightly right skewed. The last histogram, of 3000 randomly generated values shows a normal bell-shaped distribution. It does not appear to be skewered in either direction and most of the data appears to be within 2/3 standard deviations of the mean (or the bin with the largest frequency).

Q5 The shapes of the histogram are different primarily due to the sample size. As you increase your sample size, the data can follow more closely a normal distribution. This is due to us using the “rnorm” function in R. When there is a small sample size to work with, absences in values (or the lack of certain values) may lead to a less conformed bell-shape, or in this case, breaks in our histogram bins.

Q6 The standard Normal distribution has an area of 1.0 under the “normal curve” (i.e., the shape of the standard normal distribution). 68% of data observed falls between negative one deviation and one deviation. Additionally, 99.8% of all data (typically) falls between three standard deviations in the negative and positive direction. This of course, applies to data that follows the normal distribution curve.

Chloe Lang
Lab 4 Report

*Corrections done in red

Q1

```
norm_17 = rnorm(n = 17, mean = my_mean, sd = my_sd)
norm_30 = rnorm(n = 30, mean = my_mean, sd = my_sd)
norm_300 = rnorm(n = 300, mean = my_mean, sd = my_sd)
norm_3000 = rnorm(n = 3000, mean = my_mean, sd = my_sd)

norm_mean = 10.4
norm_sd = 2.4
```

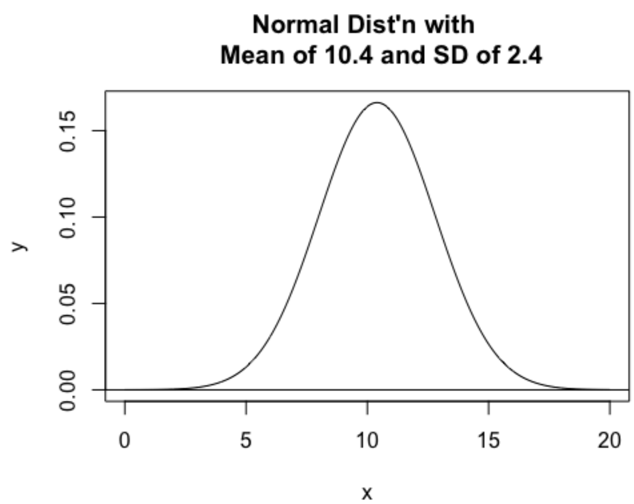
Q2

```
par(mfrow=c(2,2))
 png(filename="lab_4_hist_01.png",
        width=1500,
         height=1600,
        res=180)
hist(norm_17,
main="17 Randomly Generated Data Point",
 xlab="Data Points")
hist(norm_30,
        main= "30 Randomly Generated Data Points",
        xlab="Data Points")
hist(norm_300,
main="300 Randomly Generated Data Points",
        xlab="Data Points")
hist(norm_3000,
        main="3000 Randomly Generated Data Points",
        xlab= "Data Points")
```

Q7

```
png(filename="norm_1.png",
          width=1600,
           height=1500,
          res=180)
mean <- 10.4
sd <- 2.4
x=seq(0,20, length.out=1000)
y=dnorm(x, mean=10.4, sd=2.4)
 plot(x,y, main= "Normal Dist'n with
Mean of 10.4 and SD of 2.4 ",
type= "l", xlim= c(0,20))
abline(h=0)
```

Q8



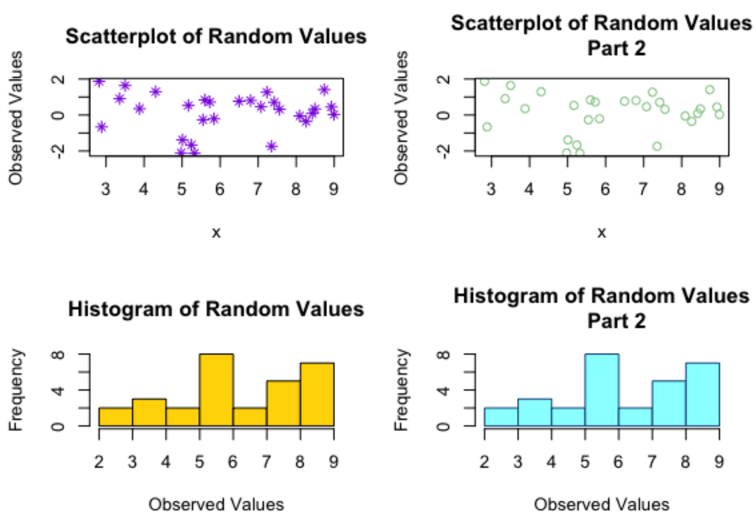
Q9

Q9:

```
n_pts = 29
x_min = 2
x_max = 9
x = runif(n = n_pts, min = x_min, max = x_max)
y_observed = rnorm(n_pts)
dat_random = data.frame(x = x, y_observed = rnorm(n_pts))
dev.off()
par(mfrow=c(2,2))
plot(x, y_observed, dat=dat_random, main = "Scatterplot of Random Values",
          ylab="Observed Values",
          col="blueviolet",
          pch=8)
plot(x, y_observed, dat=dat_random, main ="Scatterplot of Random Values
          Part 2",
```

```
        ylab="Observed Values",
         col="darkseagreen3",
        pch=21)
hist(dat_random$x,
        main="Histogram of Random Values",
        xlab="Observed Values",
        col="gold")
hist(dat_random$x,
        main="Histogram of Random
        Values Part 2",
        xlab="Observed Values", border="dodgerblue4",
        col="darkslategray1")
```

Q10

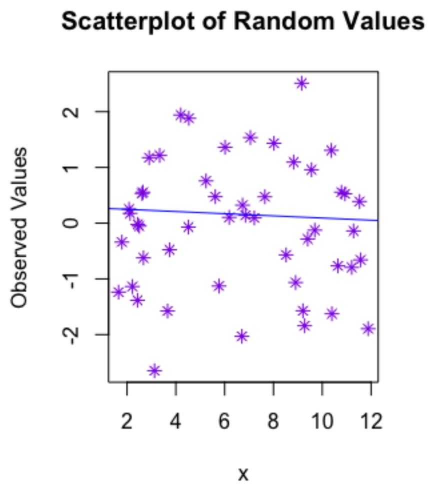


Q11

```
n_pts = 50
x_min = 1
x_max = 12
x = runif(n = n_pts, min = x_min, max = x_max)

y_observed = rnorm(n_pts)
dat_random = data.frame(x = x, y_observed = rnorm(n_pts))
png(filename="Linear_model1",
        width=1600,
        height=1500,
        res=180)
plot(x, y_observed, dat=dat_random, main = "Scatterplot of Random Values",
        ylab="Observed Values",
        col="blueviolet",
        pch=8)
abline(lm(y_observed~x, data=dat_random), col="blue")
```

Q12



Q13
n_pts = 50
x_min = 1
x_max = 12
x = runif(n = n_pts, min = x_min, max = x_max) y_observed = rnorm(n_pts)
dat_random = data.frame(x = x, y_observed = rnorm(n_pts))

guess_x=5
guess_y=1
guess_slope=0.2
dat_random$y_predicted=line_point_slope(dat_random$x, guess_x, guess_y, guess_slope)
dat_random$residuals=line_point_slope(dat_random$x, guess_x, guess_y, guess_slope)

Q14