Matthew Jusino

Lab 4 Uncertainty and Error

Q1 Vector Code

- (1 pt.) Show the code you used to create your vectors.

```
q1_sd = 2.4

q1_mean = 10.4


norm_17 = rnorm(n = 17, mean = q1_mean, sd = q1_sd)

norm_30 = rnorm(n = 30, mean = q1_mean, sd = q1_sd)

norm_300 = rnorm(n = 300, mean = q1_mean, sd = q1_sd)

norm_3000 = rnorm(n = 3000, mean = q1_mean, sd = q1_sd)
```

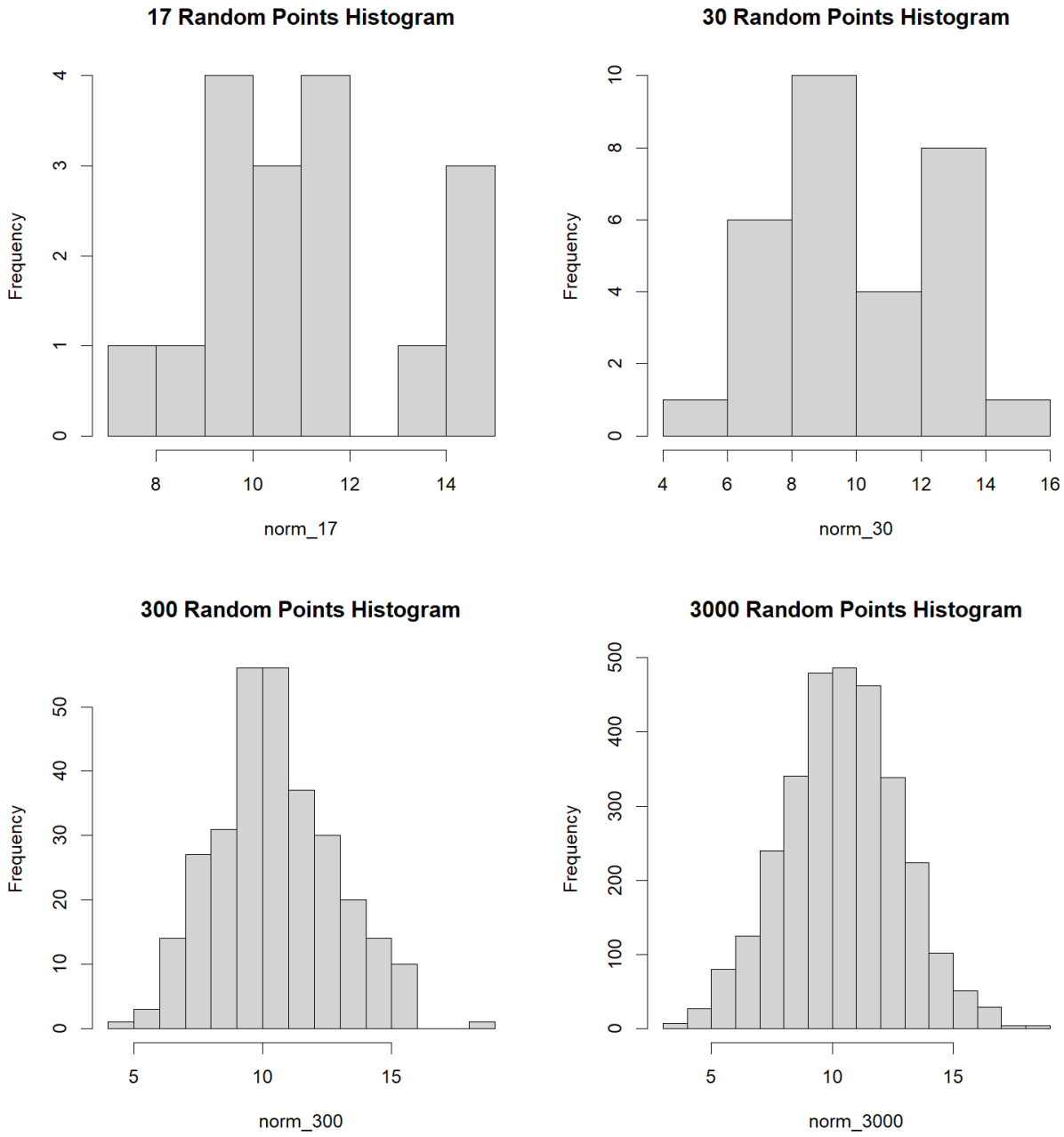
Q2 Histogram Code

- (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.

```
par(mfrow = c(2, 2))

hist(norm_17, main = "17 Random Points Histogram")

hist(norm_30, main = "30 Random Points Histogram")

hist(norm_300, main = "300 Random Points Histogram")

hist(norm_3000, main = "3000 Random Points Histogram")

png(filename = "lab_04_hist_01.png",

   width = 1500, height = 1600, res = 180)

dev.off()
```

Q3 Histogram Figure

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



Q4 Histogram Shapes 1

- (2 pts.) Qualitatively describe the differences among the histograms.

The histograms appear to become more uniform and more similar to each other the more point are included in the random sample. At 17 points, the chart is all over the place, while it shifts closer to a normal distribution as 30 points, is almost a normal distribution at 300 points, and looks almost exactly like a normal distribution at 3000 points.

Q5 Histogram Shapes 2

Explain why the shapes of the histograms are different.

The histogram shapes are different because in a rnorm distribution, the more points you add, the closer it should get to a normal distribution. With only 17 random points in the first histogram, it doesn't look anything like a normal distribution, but by the time you reach 3000, it is almost exact.

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.

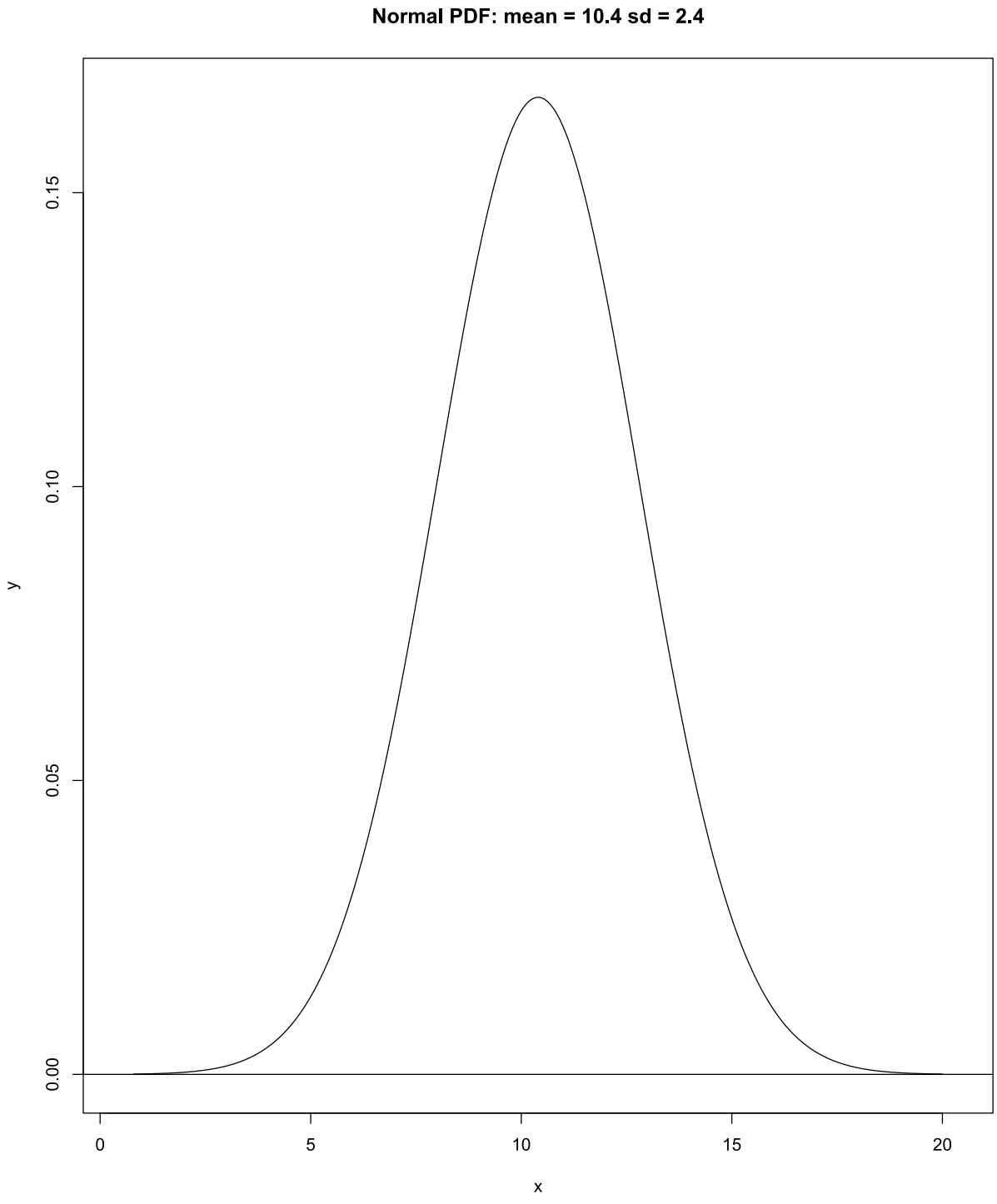
Q7 Density Figure Code

- (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(0.8, 20, length.out = 1000)

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


plot(x, y, main = "Normal PDF: mean = 10.4 sd = 2.4", type = "l", xlim = c(0.4, 20.4))

abline(h = 0)

svg(filename = "norm_1.svg",

   width = 1500, height = 1600)

dev.off()
```

Q8 Density Figure File

- (2 pts.) Upload `norm_1.svg` (or `norm_1.pdf`).



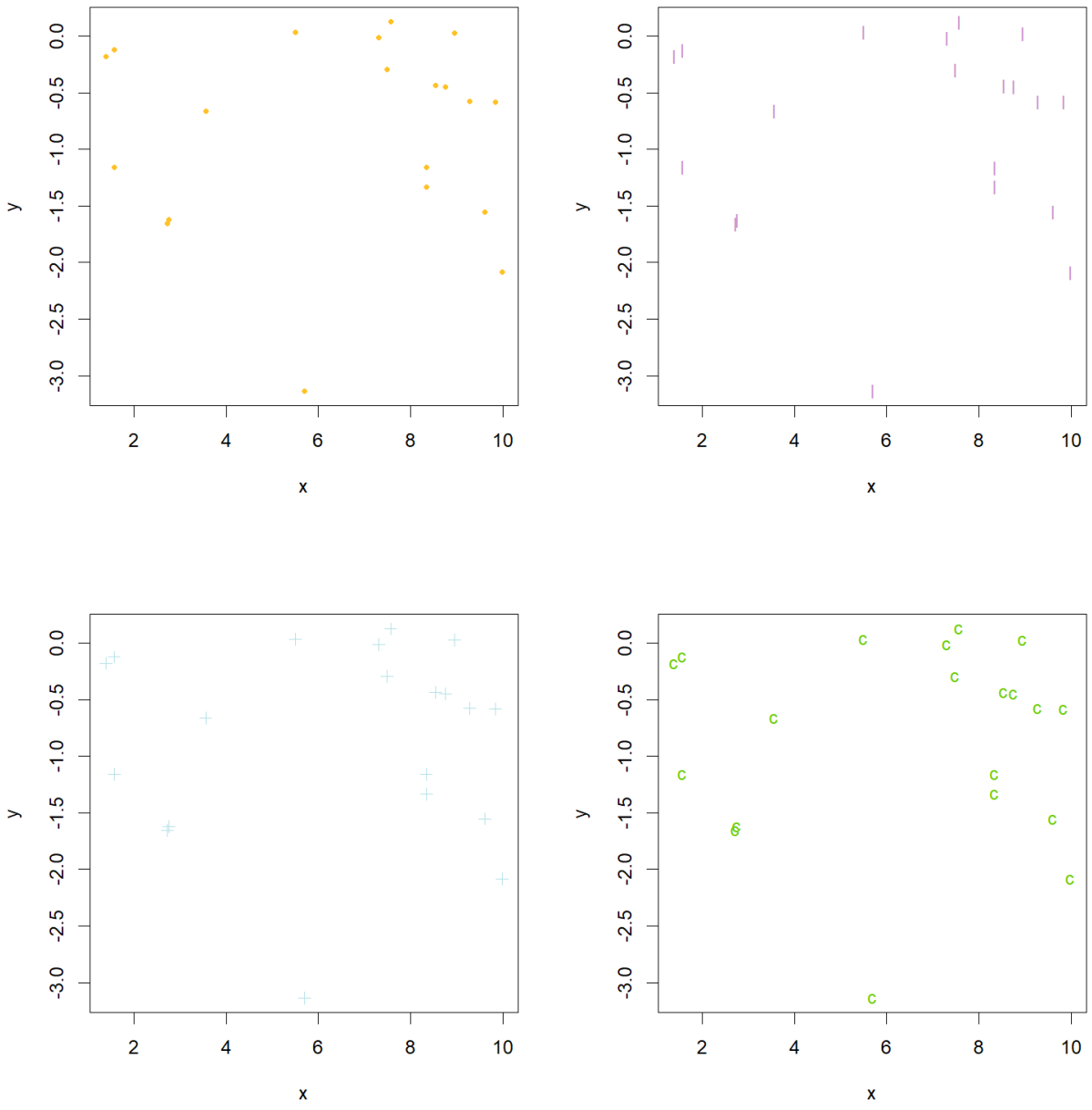
Q9 Random Data Set

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

```
n_pts = 33

x_min = 0

x_max = 20


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

y_random = rnorm(n = n_pts)


dat_ran = data.frame(x = x_random, y = y_random)


plot(y ~ x, data = dat_ran, pch = 20, col = "goldenrod1")
```

Q10 Random Data Image File

- (2 pts.) Upload an image file of your figure. It may be in png (raster graphics), svg (vector graphics), or pdf (vector graphics) format.
    - Check out the corresponding functions `png()`, `pdf()`, and `svg()` for info.



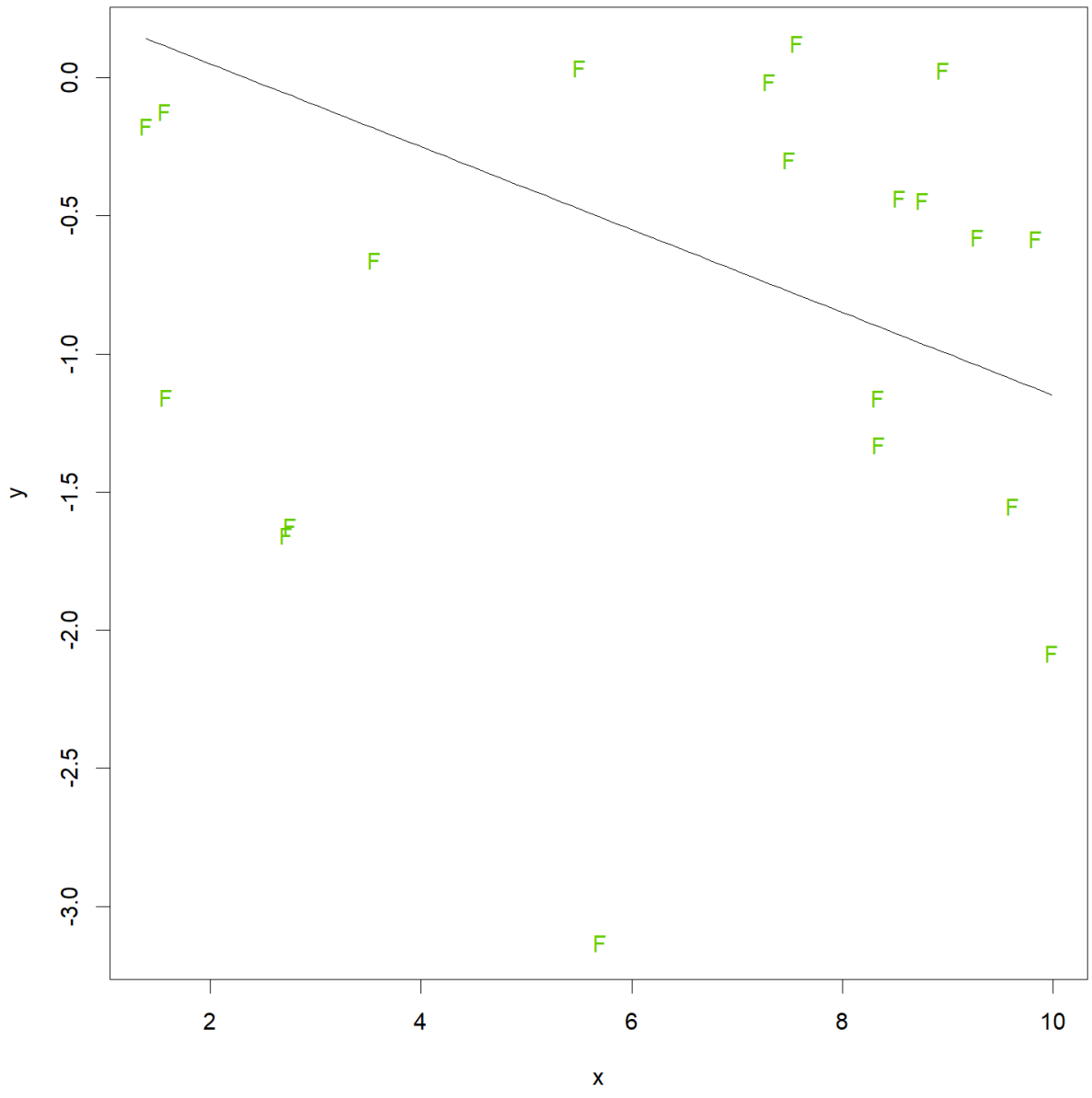
Q11 Random Data Model Fit (Q 13-14)

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

```
par(mfrow =c(1,1))

n_pts = 30

slope = 0.7

intcp = 0.2


guess_x = 9

guess_y = -1

guess_slope = -0.15


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

y_random = rnorm(n = n_pts)


plot(y ~ x, data = dat_ran, pch = 70, col = "chartreuse3")

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

Q12 Random Data Model Fit Image File

- (2 pts.) Upload an image file of your figure. It may be in png (raster graphics), svg (vector graphics), or pdf (vector graphics) format.
    - Check out the corresponding functions `png()`, `pdf()`, and `svg()` for info.



Q13 Random Data Model Residuals (Q 13-14)

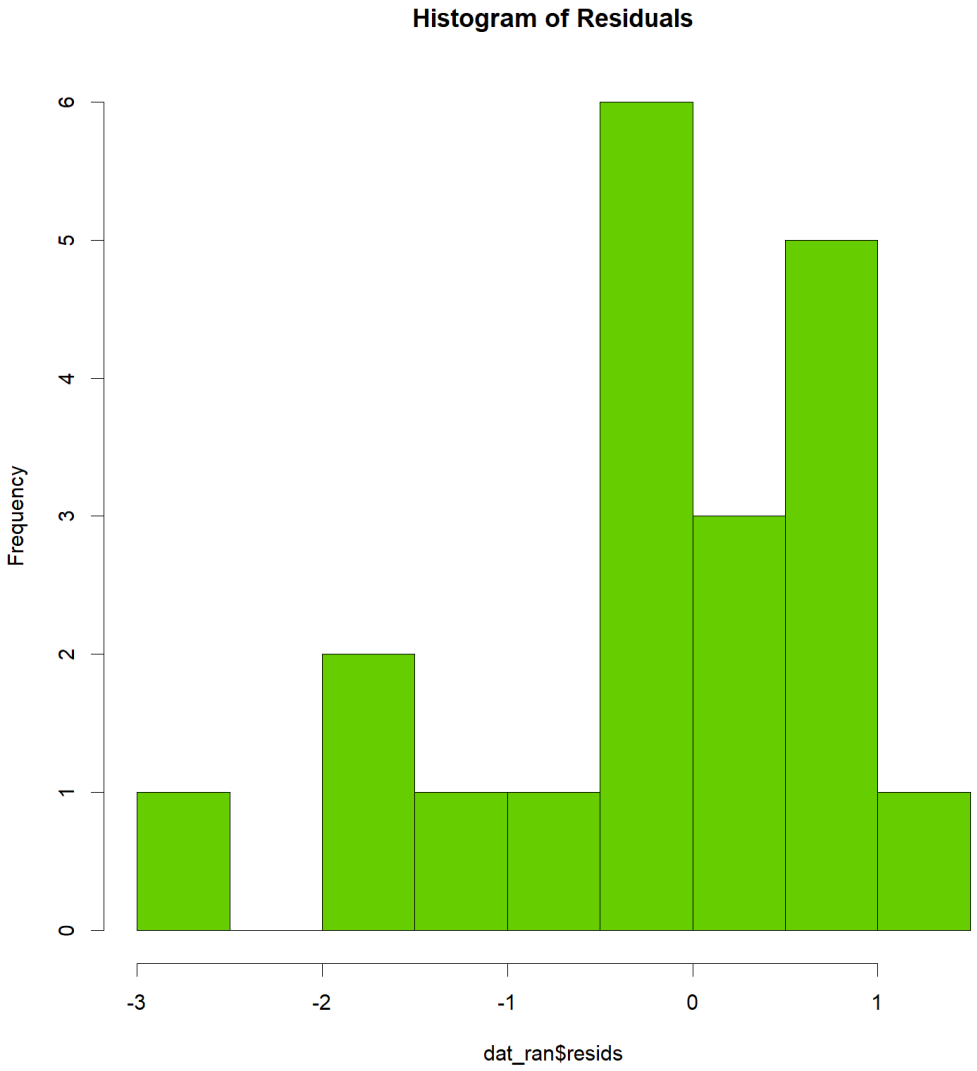
- (2 pts.) Paste the R code you used to create create the columns of predicted values and residuals.

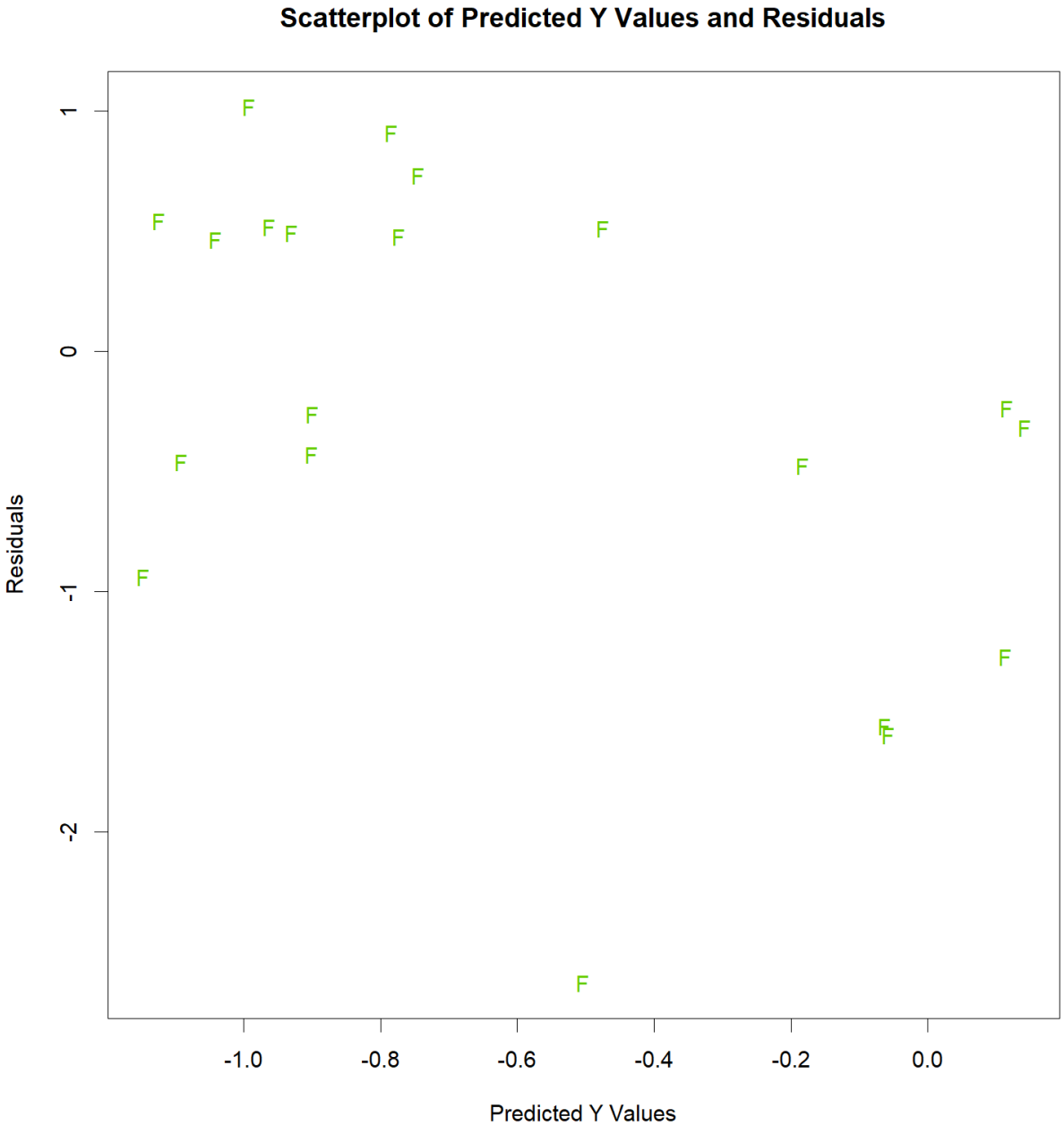
line_point_slope(dat_ran$x, guess_x, guess_y, guess_slope)

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

dat_ran$resids <- dat_ran$y - dat_ran$y_predicted
```

Q14 Random Data Model Residual Plot

- (3 pts.) In your report, include the two following figures
  - A histogram of the model's residuals.
  - A scatterplot of your model's predicted values (on the x-axis) and residuals (on the y-axis)




Scatterplot of Predicted Y Values and Residuals
Residuals
1
0
-1
-2
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
Predicted Y Values
F