Section 5 Activity

Main

The goal of this question is to practice simulating data and running regressions in R. It is taken from your textbook.

1. Set the RNG seed to be 1

```
library(ggplot2)
set.seed(1)
```

2. Create a vector `x` that has 30 observations randomly drawn from a standard normal distribution. (Hint: use the `rnorm()` function).

```
N = 30
x = rnorm(N)
```

3. Create a second vector `eps` that has 30 observations randomly drawn from a normal distribution with a mean of 0 and a standard deviation of 0.25.

```
eps = rnorm(N, 0, 0.25)
```

4. Using `x` and `eps` create a vector `y` according to the following data generating process

$$Y = -1 + 0.5X + \epsilon$$

```
y = -1 + 0.5*x + eps
```

5. What is the length of y? What are the values of β in the DGP?

```
length(y)
```

```
[1] 30
```

$\beta_0 = -1, \beta_1 = 0.5$

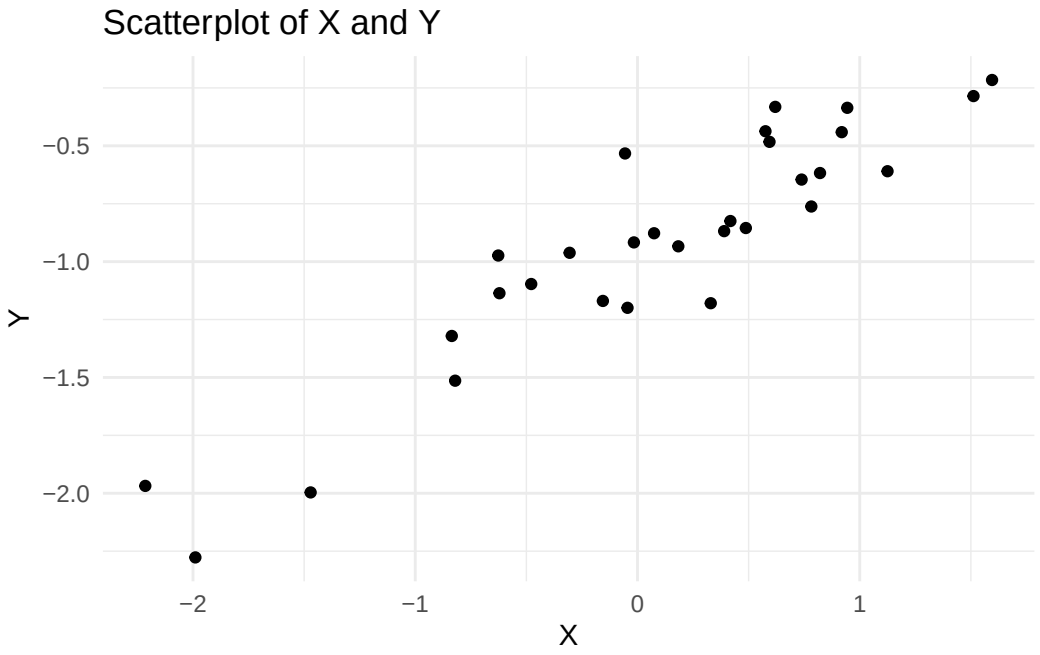
6. Create a data frame called `dgp` with the variables created in 2-4.

```
dgp = data.frame(y, x, eps)
```

7. Using `ggplot2` create a scatterplot of the relationship between `x` and `y`.

```
plot = dgp |>
  ggplot(aes(x,y))+
  geom_point()+
  theme_minimal()+
  labs(title = "Scatterplot of X and Y", x = "X", y = "Y")

plot
```



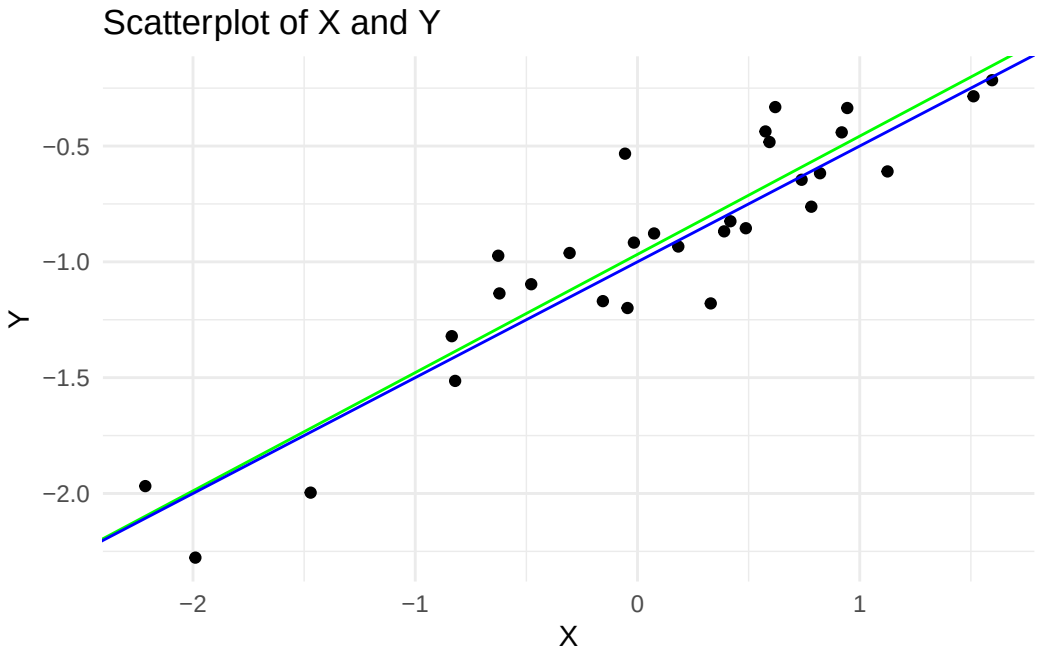
8. Run the regression of `y` on `x` and report the summary of the model. Call this `m1`. Comment on why you expect the result. (Hint: consider the discussion of the Conditional Expectation Function from lecture)

```
m1 = lm(y~x, data = dgp)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.9676698	0.0370517	-26.11673	0
x	0.5104716	0.0406125	12.56931	0

9. Using `ggplot2` add the least squares line to your previous plot. Give it a color other than black. Draw the population regression line on the plot in a different color.

```
plot +
  geom_abline(intercept = coef(m1)[1],
              slope = coef(m1)[2],
              color = "green")+
  geom_abline(intercept = -1,
              slope = 0.5,
              color = "blue")
```



10. Create a second model `m2` that adds a squared term x^2 to the model. Is there evidence that the term improves the model fit? Which model is "correct"?

```
m2 = lm(y~x + I(x^2), data = dgp)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.9316842	0.0472934	-19.700071	0.0000000
x	0.4891827	0.0439631	11.127110	0.0000000
I(x^2)	-0.0411256	0.0340192	-1.208893	0.2371844

We'd generally think that the model that best estimates the true data generating process is correct. Adding in the interaction term would give us a higher R^2 in our model, but the fit is misspecified. We will see that in action in the next question.

11. For both models, manually predict the result of `y` when `x = 4`. Would you trust either prediction?

```
m1_new_data = c(1,4)
m2_new_data = c(1,4,4^2)
```

The prediction from the first model is (rounded) 1.07 and the prediction from the second model is (rounded) 0.37.

I would probably trust the first one more than the second because it estimates the true data generating process. I might be "worried" about the first prediction as well because the value is far outside of my training data set.

Bonus

1. Add a new variable `z` to the `dgp` data frame that has 30 observations randomly drawn from a Poisson distribution. Set `lambda=3`.

In R, the way to get random observations from a Poisson distribution is ***`rpois`***

```
dgp$z = rpois(N, lambda = 3)
```

2. Update the `y` variable in the `dgp` data frame so that Y is now drawn from the following data generating process.

$$-1 + 0.5X + .25Z + .75(XZ) + \epsilon$$

```
dgp$y = -1 + 0.5*dgp$x + .25*dgp$z + .75*(dgp$x*dgp$z) + dgp$eps
```

Note that we can and should just operate directly with the data frame rather than using objects outside of it here.

3. Run a new model called `m3` that would perfectly estimate the CEF in expectation. Report the summary of this model.

```
m3 = lm(y~x*z, data = dgp)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.9782745	0.0782729	-12.498248	0e+00
x	0.5940044	0.0927435	6.404812	9e-07
z	0.2548088	0.0223525	11.399563	0e+00
x:z	0.7218926	0.0277883	25.978265	0e+00

Since the data generating process is linear, the CEF is also linear. The best fit is thus a linear model with the appropriate interactions.