C1M3_peer_reviewed

December 24, 2021

1 Module 3: Peer Reviewed Assignment

1.0.1 Outline:

The objectives for this assignment:

1. Learn how to read and interpret p-values for coefficients in R.
2. Apply Partial F-tests to compare different models.
3. Compute confidence intervals for model coefficients.
4. Understand model significance using the Overall F-test.
5. Observe the variability of coefficients using the simulated data.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

```
[2]: # Load Required Packages
     library(ggplot2)
```

1.1 Problem 1: Individual t-tests

The dataset below measures the chewiness (mJ) of different berries along with their sugar equivalent and salt (NaCl) concentration. Let's use these data to create a model to finally understand chewiness.

Here are the variables: 1. `nacl`: salt concentration (NaCl) 2. `sugar`: sugar equivalent 3. `chewiness`: chewiness (mJ)

Dataset Source: I. Zouid, R. Siret, F. Jourjion, E. Mehinagic, L. Rolle (2013). "Impact of Grapes Heterogeneity According to Sugar Level on Both Physical and Mechanical Berries Properties and their Anthocyanins Extractability at Harvest," Journal of Texture Studies, Vol. 44, pp. 95-103.

1. (a) Simple linear regression (SLR) parameters In the below code, we load in the data and fit a SLR model to it, using `chewiness` as the response and `sugar` as the predictor. The summary of the model is printed. Let $\alpha = 0.05$.

Look at the results and answer the following questions: * What is the hypothesis test related to the p-value `2.95e-09`? Clearly state the null and alternative hypotheses and the decision made based on the p-value. * Does this mean the coefficient is statistically significant? * What does it mean for a coefficient to be statistically significant?

```
[3]: # Load the data
     chew.data = read.csv("berry_sugar_chewy.csv")

     chew.lmod = lm(chewiness~sugar, data=chew.data)
     summary(chew.lmod)
```

```
Call:
lm(formula = chewiness ~ sugar, data = chew.data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4557 -0.5604  0.1045  0.5249  1.9559

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.662878   0.756610  10.128  < 2e-16 ***
sugar       -0.022797   0.003453  -6.603 2.95e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9178 on 88 degrees of freedom
Multiple R-squared:  0.3313,	Adjusted R-squared:  0.3237
F-statistic: 43.59 on 1 and 88 DF,  p-value: 2.951e-09
```

1.1.1 Answer:

The hypothesis test is as follows:
$H_0 : Y_i = \beta_0 + \varepsilon_i$
H_1 : A predictor is needed in the model
The p-value is very, very close to 0. Therfore, we can state a predictor is needed in the model.

Because this is simple linear regression model, it does mean this coefficient is statistically significant. However, for multiple linear regression model we wouldn't be able to say the same. We would need to look at the individual t test. This is because the F-test tests whether the predictors are jointly significant. To test individual significance, we would need to take a look at the t test. In this case, since there is only 1 predictor, the t test is the same as the F-test.

For our coefficient to be statistically significant, the p-value of the t-test needs to be less than .05. In other words, Is there a reason the believe the predictor associated with sugar is meaningfully different than 0.

1. (b) MLR parameters   Now let's see if the second predictor/feature `nacl` is worth adding to the model. In the code below, we create a second linear model fitting `chewiness` as the response with `sugar` and `nacl` as predictors.

Look at the results and answer the following questions: * Which, if any, of the slope parameters are statistically significant? * Did the statistical significance of the parameter for `sugar` stay the same, when compared to 1 (a)? If the statistical signficance changed, explain why it changed. If it didn't change, explain why it didn't change.

```
[4]: chew.lmod.2 = lm(chewiness ~ ., data=chew.data)
     summary(chew.lmod.2)
```

```
Call:
lm(formula = chewiness ~ ., data = chew.data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3820 -0.6333  0.1234  0.5231  1.9731

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.1107    13.6459  -0.521    0.604
nacl          0.6555     0.6045   1.084    0.281
sugar        -0.4223     0.3685  -1.146    0.255

Residual standard error: 0.9169 on 87 degrees of freedom
Multiple R-squared:  0.3402,Adjusted R-squared:  0.325
F-statistic: 22.43 on 2 and 87 DF,  p-value: 1.395e-08
```

1.1.2   Answer:

- None of the slope parameters are statistically significant. None of their t-tests have a value less than .05
- The stastical significance of the sugar parameter changed. Adding `nacl` cause a change in the `chewiness` variance. This could be due to `nacl` being closely related, or correlated, to `sugar`. We can tell `nacl` didn't add much information, if any, because R^2 didn't increase by much.

1. (c) Model Selection   Determine which of the two models we should use. Explain how you arrived at your conclusion and write out the actual equation for your selected model.

```
[5]: anova(chew.lmod, chew.lmod.2)
```

A anova: 2×6

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	88	74.12640	NA	NA	NA	NA
2	87	73.13801	1	0.9883882	1.175719	0.2812249

We should use the following model:

$$Y_i = 7.662878 + -0.022797 X_{sugar} + \varepsilon_i$$

The F-test has a p-value of 0.2812. Therefore, we fail to reject the null hypothesis: the reduced model (only `sugar`) is better.

1. (d) Parameter Confidence Intervals Compute 95% confidence intervals for each parameter in your selected model. Then, in words, state what these confidence intervals mean.

```
[19]: # Your Code Here
      confint(chew.lmod)
```

A matrix: 2×2 of type dbl

	2.5 %	97.5 %
(Intercept)	6.15927388	9.16648152
sugar	-0.02965862	-0.01593536

1.1.3 Answer:

The confidence interval for sugar is (-0.0297, -0.0159). This means we are 95% confident the true change that 1 increase of sugar concentration makes on chewiness is between -0.0297 and -.0169 mJ. Because 0 is not contained in teh interval, we are confiderent there is a significant relationship between sugar and chewiness.

2 Problem 2: Variability of Slope in SLR

In this exercise we'll look at the variability of slopes of simple linear regression models fitted to realizations of simulated data.

Write a function, called `sim_data()`, that returns a simulated sample of size $n = 20$ from the model $Y = 1 + 2.5X + \epsilon$ where $\epsilon \overset{iid}{\sim} N(0, 1)$. We will then use this generative funciton to understand how fitted slopes can vary, even for the same underlying population.

```
[20]: sim_data <- function(n=20, var=1, beta.0=1, beta.1=2.5){
          # BEGIN SOLUTION HERE
          x = seq(-1, 1, length.out = n); beta0 = 1; beta1 = 2.5; e = rnorm(n, 0,␣
      ↪sqrt(var))
          y = beta0 + beta1*x + e
          # END SOLUTION HERE
          data = data.frame(x=x, y=y)
          return(data)
      }
```

2. (a) Fit a slope   Execute the following code to generate 20 data points, fit a simple linear regression model and plot the results.

Just based on this plot, how well does our linear model fit the data?

```
[8]: data = sim_data()
     lmod = lm(y~x, data=data)
     ggplot(aes(x=x, y=y), data=data) +
         geom_point() +
         geom_smooth(method="lm", formula=y~x, se=FALSE, color="#CFB87C")
```
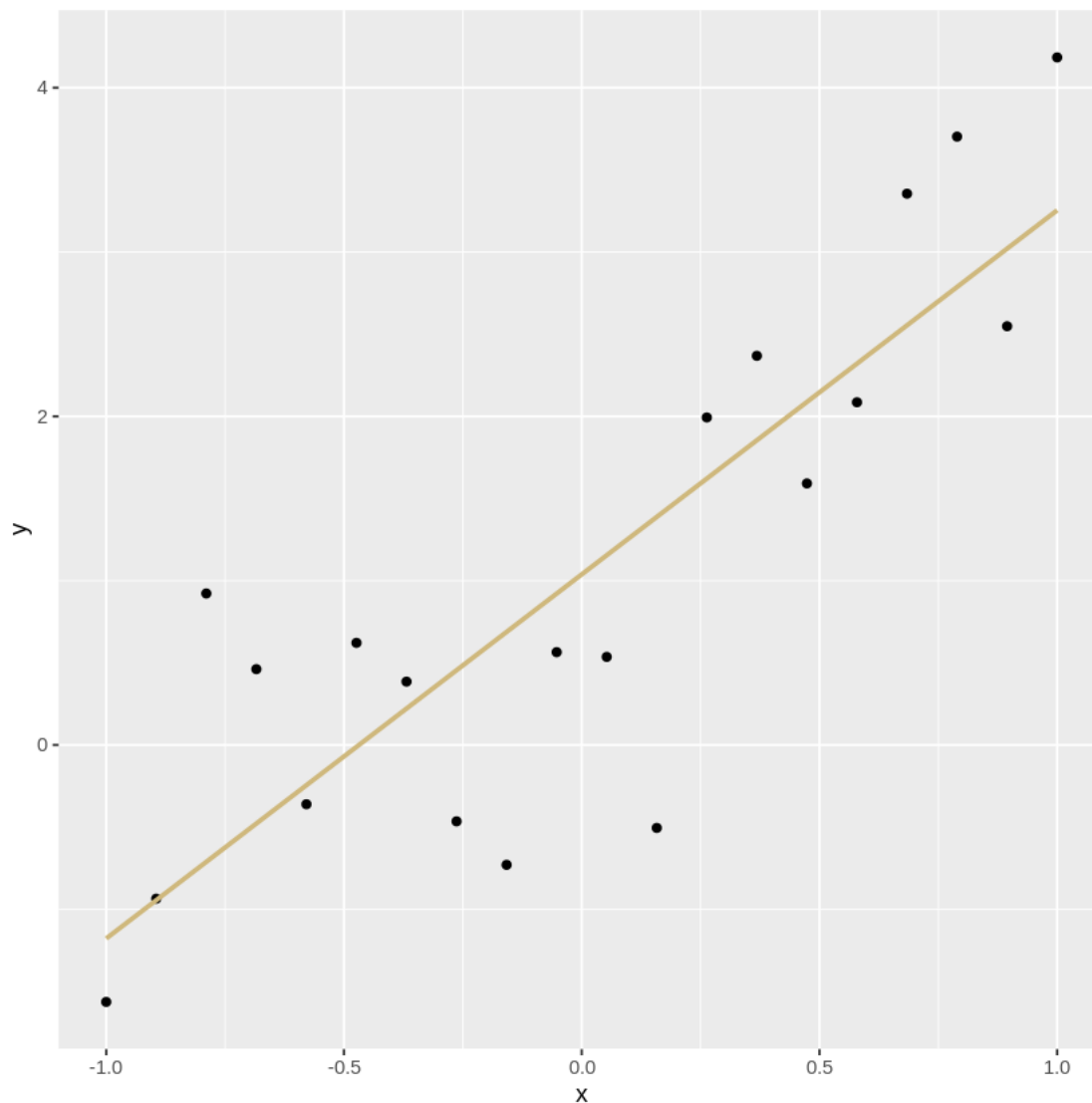
2.0.1 Answer:

Although the model isn't perfect, it does fit the data well.

2. (b) Do the slopes change? Now we want to see how the slope of our line varies with different random samples of data. Call our data genaration funciton 50 times to gather 50 independent samples. Then we can fit a SLR model to each of those samples and plot the resulting slope. The function below performs this for us.

Experiment with different variances and report on what effect that has to the spread of the slopes.
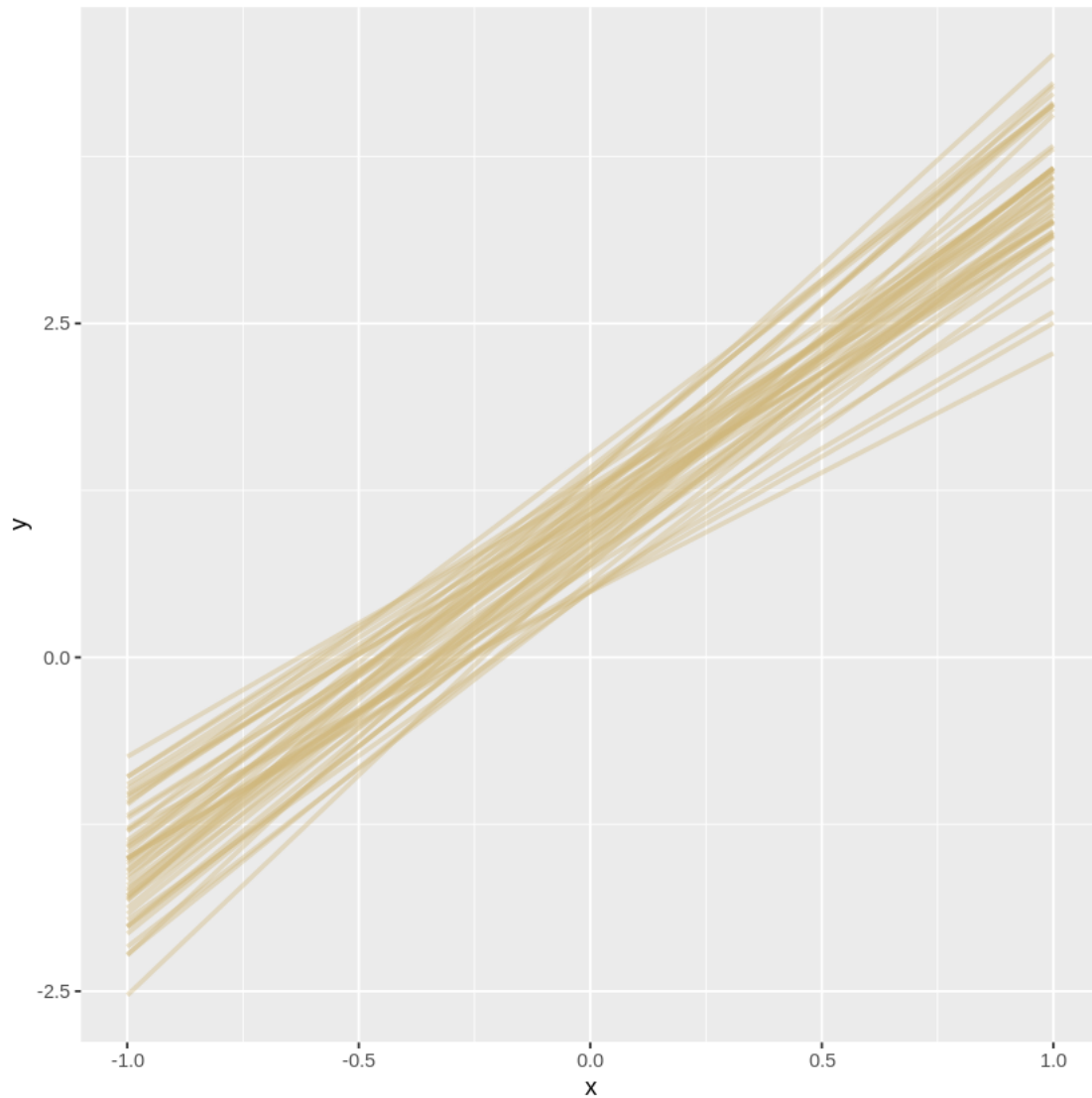
```
[9]: gen_slopes <- function(num.slopes=50, var=1, num.samples=20){
         g = ggplot()
         # Repeat the sample for the number of slopes
         for(ii in 1:num.slopes){
             # Generate a random sampling of data
             data = sim_data(n=num.samples, var=var)
             # Add the slope of the best fit linear model to the plot
             g = g + stat_smooth(aes(x=x, y=y), data=data, method="lm", geom="line",
                                 se=FALSE, alpha=0.4, color="#CFB87C", size=1)
         }
         return(g)
     }
```

```
[10]: gen_slopes()
```

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```

[11]: `gen_slopes(var=4)`

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```
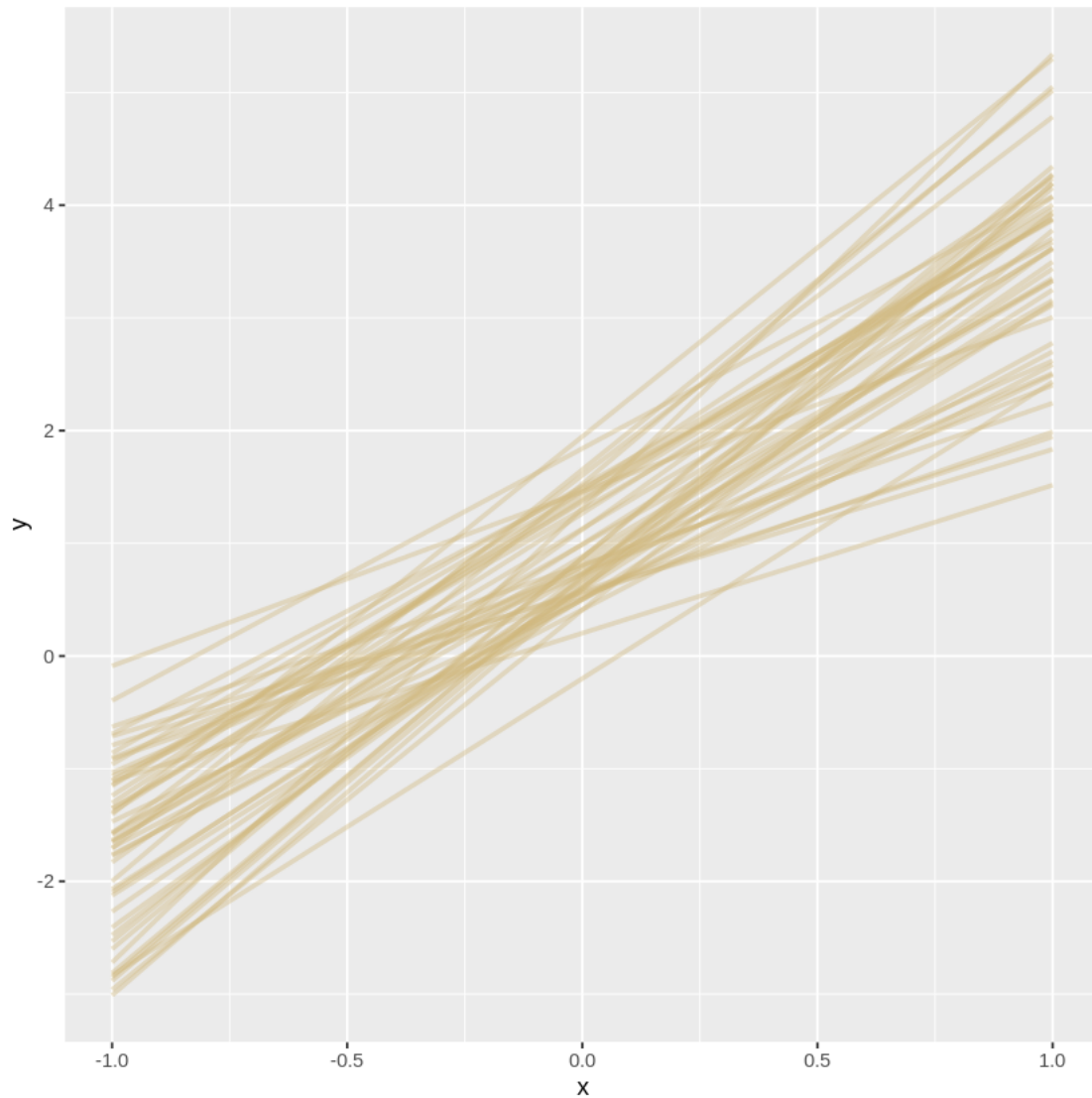
```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```
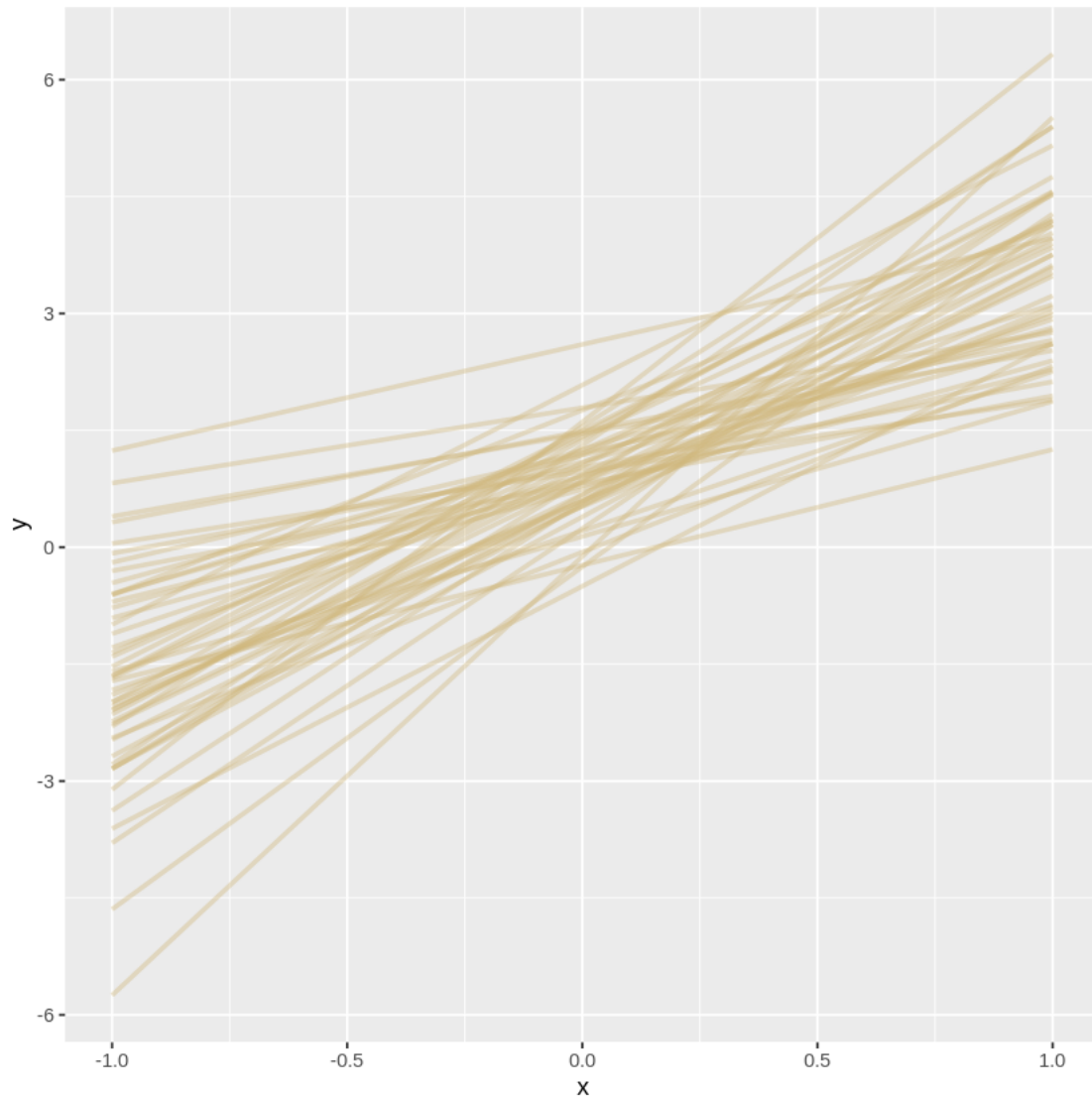
```
[12]: gen_slopes(var=8)
```

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```

```
`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'

`geom_smooth()` using formula 'y ~ x'
```

2.0.2 Answer:

As the variance increases, the slope variability does as well.

2. (c) Distributions of Slopes   As we see above, the slopes are somewhat random. That means that they follow some sort of distribution, which we can try to discern. The code below computes `num_samples` independent realizations of the model data, computes the SLR model, and generates a histogram of the resulting slopes.

Again, experiment with different variances for the simulated data and record what you notice. What do you notice about the shapes of the resulting histograms?

```
[13]: hist_slopes <- function(num.slopes=500, var=1, num.samples=20){
          slopes = rep(0, num.slopes)
          # For num.slopes, compute a SLR model slope
          for(i in 1:num.slopes){
              # Simulate the desired data
              data = sim_data(var=var, n=num.samples)
              # Fit an SLR model to the data
              lmod = lm(y~x, data=data)
              # Add the slopes to the vector of slopes
              slopes[i] = lmod$coef[2]
          }
          # Plot a histogram of the resulting slopes
          g = ggplot() + aes(slopes) + geom_histogram(color="black", fill="#CFB87C")
          return(g)
      }
```
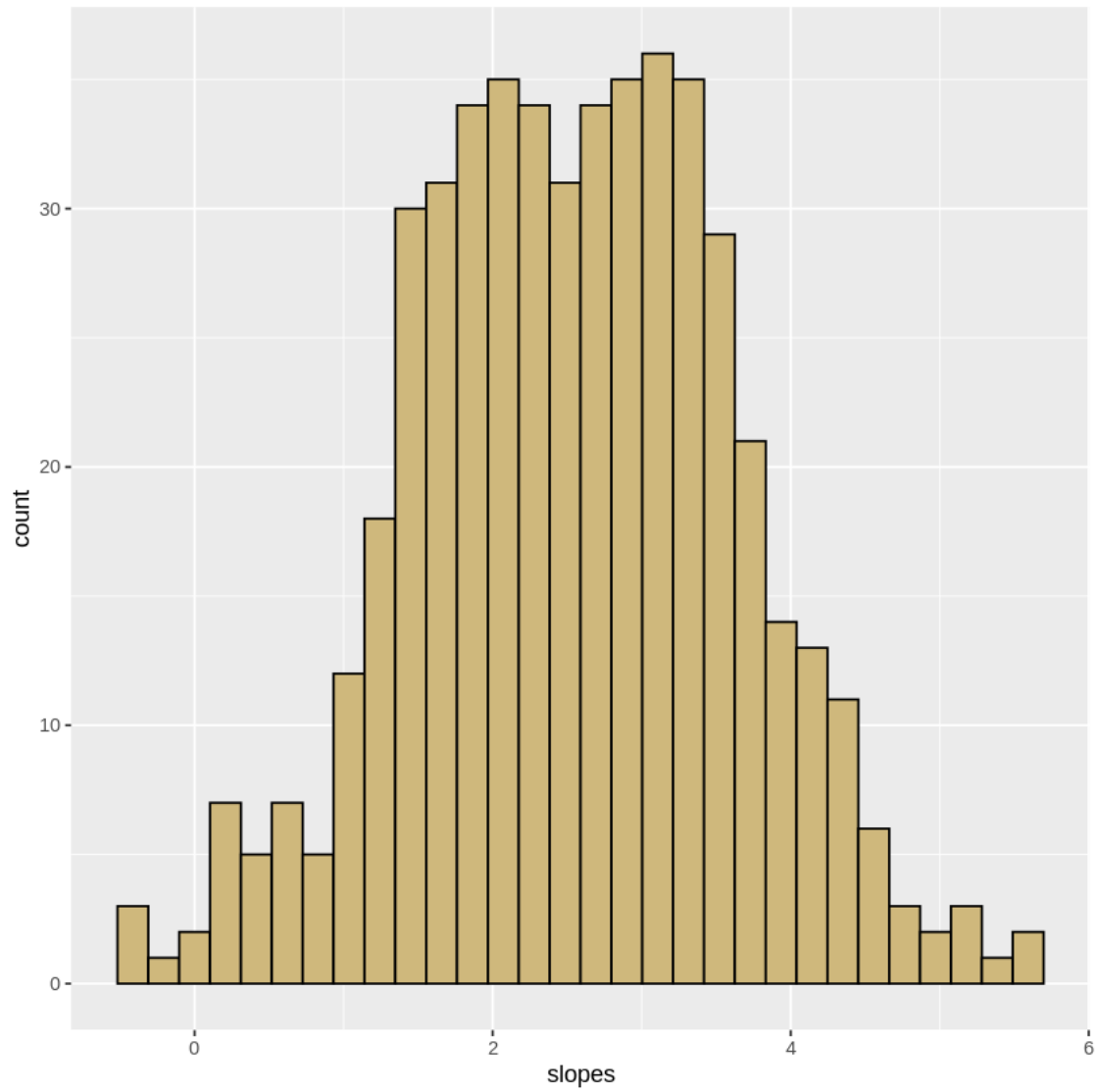
```
[14]: hist_slopes()
```

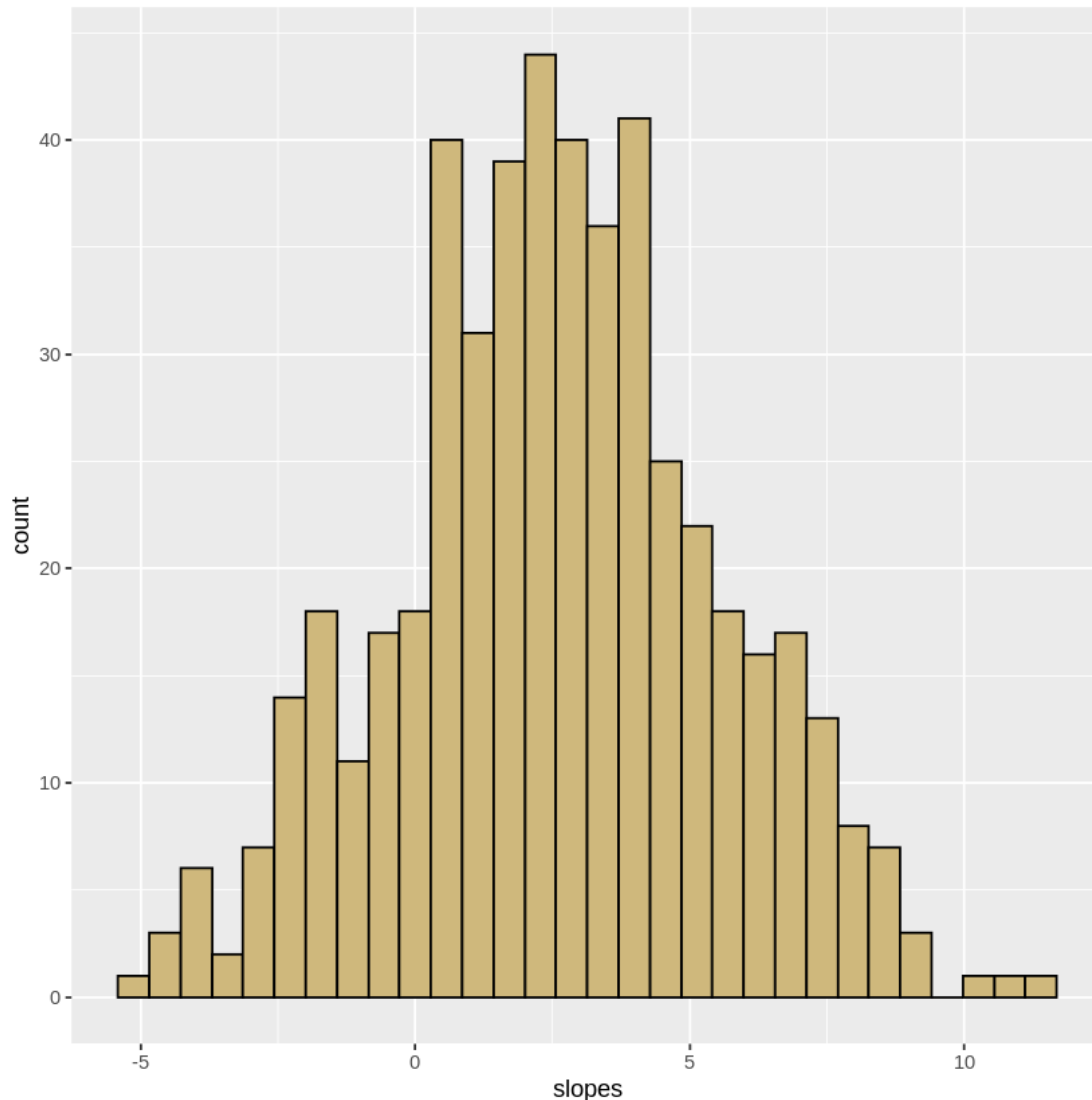`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
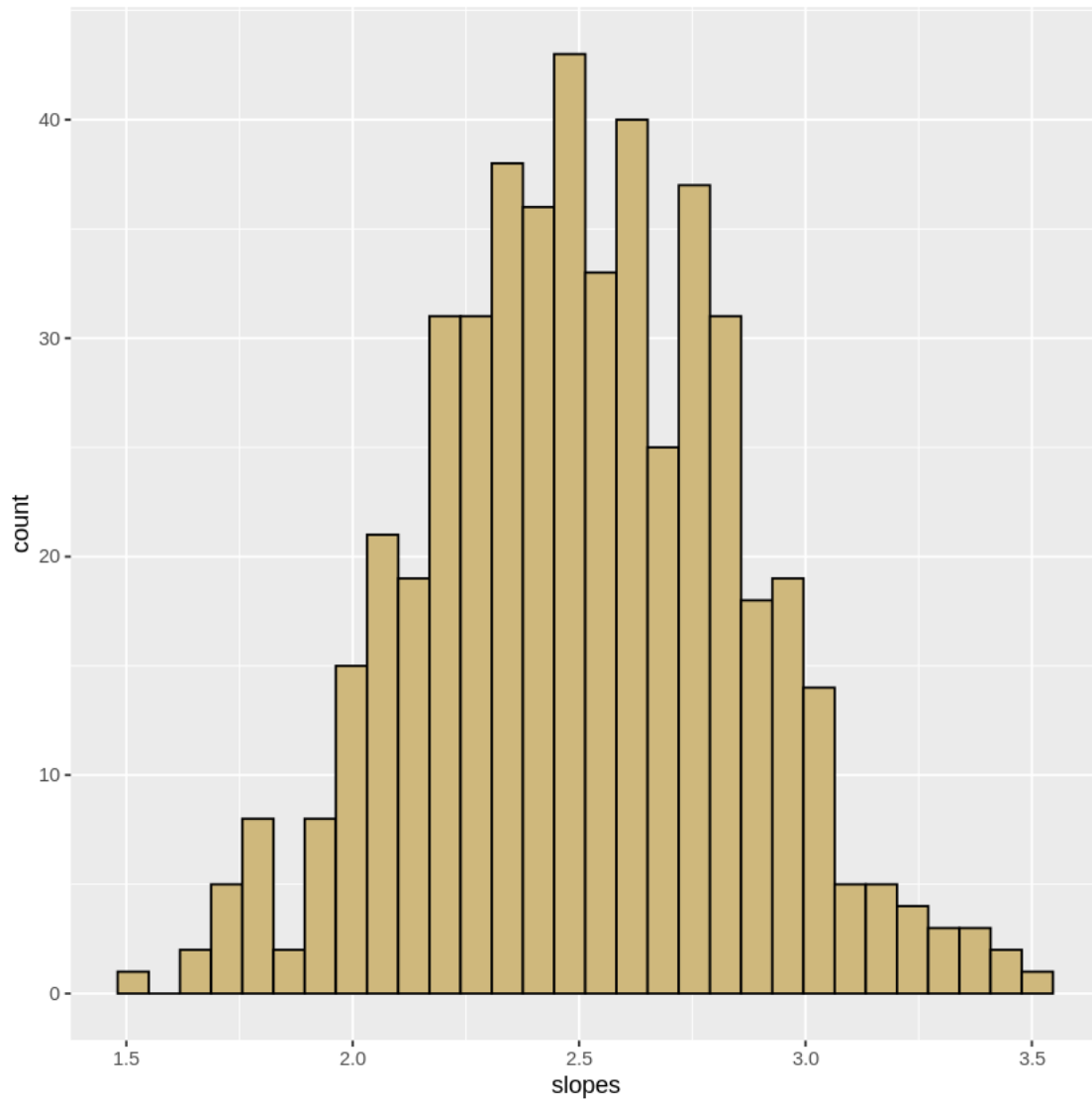
```
[15]: hist_slopes(var=8)
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```
[16]: hist_slopes(var=64)
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

2.0.3 Answer:

The histogram of slopes follow a bell curve; in other words, the slope seem to follow the normal distribution. However, as the variance increases, the range of values of the slopes do as well.

2. (d) Confidence Intervals of Slopes   What does that all mean? It means that when we fit a linear regression model, our parameter *estimates* will not be equal to the true parameters. Instead, the estimates will vary from sample to sample, and form a distribution. This is true for any linear regression model with any data - not just simulated data - as long as we assume that there is a large population that we can resample the response from (at fixed predictor values). Also note that we only demonstrated this fact with the slope estimate, but the same principle is true for the intercept, or if we had several slope parameters.

This simulation shows that there is a chance for a linear regression model to have a slope that is very different from the true slope. But with a large sample size, n, or small error variance, σ^2, the distribution will become narrower. Confidence intervals can help us understand this variability. The procedure that generates confidence intervals for our model parameters has a high probability of covering the true parameter. And, the higher n is, for a fixed σ^2, or the smaller σ^2 is, for a fixed n, the narrower the confidence interval will be!

Draw a single sample of size $n = 20$ from `sim_data()` with variance $\sigma^2 = 1$. Use your sample to compute a 95% confidence interval for the slope. Does the known slope for the model (which we can recall is 2.5) fall inside your confidence interval? How does the value of σ^2 affect the CI width?

```
[23]: # Your code here
      set.seed(23923)
      data.ci = sim_data(n=20)
      slope_lm = lm(y ~ x, data = data.ci)
      confint(slope_lm)
```

A matrix: 2×2 of type dbl		2.5 %	97.5 %
	(Intercept)	0.4678678	1.112756
	x	1.8132176	2.875677

```
[24]: set.seed(23028)
      data.ci2 = sim_data(n=20, var = 8)
      slope_lm2 = lm(y ~ x, data = data.ci2)
      confint(slope_lm2)
```

A matrix: 2×2 of type dbl		2.5 %	97.5 %
	(Intercept)	-0.1129461	2.399668
	x	1.1446465	5.284201

As shown in the first cell, the confidence interval for the slope of our model is (1.813, 2.876). 2.5 is contained within the interval. The larger σ^2 is thae wider the width of the confidence interval.

```
[ ]:
```