Assignment 3: Inference in Linear Regression

Nicole McCarthy

September 22, 2025

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)

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.

H 0 : The estimate for the coefficient of the parameter sugar is equal to 0. H A : The estimate for the coefficient of the parameter sugar is not equal to 0.

$$H_0 : \widehat{\beta}_{sugar} = 0$$

$$H_A : \widehat{\beta}_{sugar} \neq 0$$

Because the given p-value for this test, $2.95e-09$, is less than our given α, 0.05, we conclude that there is enough evidence to reject the null hypothesis H_0.

Does this mean the coefficient is statistically significant?

Yes, $\widehat{\beta}_{sugar}$, -0.022797, is statistically significant.

What does it mean for a coefficient to be statistically significant?

It means that the coefficient makes a noticable difference in the outputs of the model. Basically, it means that the parameter is important, at least statistically, for determining at least some of the variance of the response values.

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.

```
chew.lmod.2 = lm(chewiness ~ sugar + nacl, data=chew.data)
summary(chew.lmod.2)
```

```
##
## Call:
## lm(formula = chewiness ~ sugar + nacl, 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
## sugar        -0.4223     0.3685  -1.146    0.255
## nacl          0.6555     0.6045   1.084    0.281
##
## 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
```

Look at the results and answer the following questions:

Which, if any, of the slope parameters are statistically significant?

Neither of the slope parameters are statistically significant because neither has a p-value less than 0.05.

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.

No, the statistical significance of sugar did not stay the same as the model in 1(a). This is because the addition of a second parameter meant that some of the variance that was accounted for by sugar was not attributed to nacl. However, neither accounted for enough variance for the coefficients to be statistically significant. This can happen when two parameters are correlated.

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.

```
anova(chew.lmod,chew.lmod.2)
```

```
## Analysis of Variance Table
##
## Model 1: chewiness ~ sugar
## Model 2: chewiness ~ sugar + nacl
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     88 74.126
## 2     87 73.138  1   0.98839 1.1757 0.2812
```

The F-test for sugar-sugar+nacl models has a p-value higher than 0.05, our given α. This means that the amount of variance accounted for in the sugar and sugar+nacl models is not statistically significant. In other words, adding nacl as a predictor did not improve the model's ability to predict responses correctly. Therefore,

I would choose the model that only uses sugar as a predictor. The equation for this model is below.

$$chewiness = 7.662878 - 0.022797 * sugar$$

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.

```
# get summary into variable to easily grab information from it
mod.items <- summary(chew.lmod)

# grab estimates
inter <- mod.items$coefficients[1, "Estimate"]
sugar <- mod.items$coefficients[2, "Estimate"]

# grab standard errors
inter.se <- mod.items$coefficients[1, "Std. Error"]
sugar.se <- mod.items$coefficients[2, "Std. Error"]

# get degrees of freedom for t-statistic
n <- dim(chew.data)[1]
p <- 1 # sugar is the only predictor in this model
deg.free <- n - p - 1

# calculate t-statistic
alpha <- 0.05
t <- qt(1 - alpha/2, deg.free)

# 1. calculate intercept confidence interval
inter.margin <- abs(t * inter.se)
inter.low <- inter - inter.margin
inter.upp <- inter + inter.margin
print("Intercept 95% Confidence Interval")
print(c(inter.low, inter.upp))

# 2. calculate sugar coefficient confidence interval
sugar.margin <- abs(t * sugar.se)
sugar.low <- sugar - sugar.margin
sugar.upp <- sugar + sugar.margin
print("Sugar Coefficient 95% Confidence Interval")
print(c(sugar.low, sugar.upp))
```

```
## [1] "Intercept 95% Confidence Interval"
## [1] 6.159274 9.166482
## [1] "Sugar Coefficient 95% Confidence Interval"
## [1] -0.02965862 -0.01593536
```

The 95% confidence interval means that there is a high possibility that the true value of the specified parameter (intercept or sugar coefficient) falls within the given interval.

Problem 2: Variability of Slope in SLR

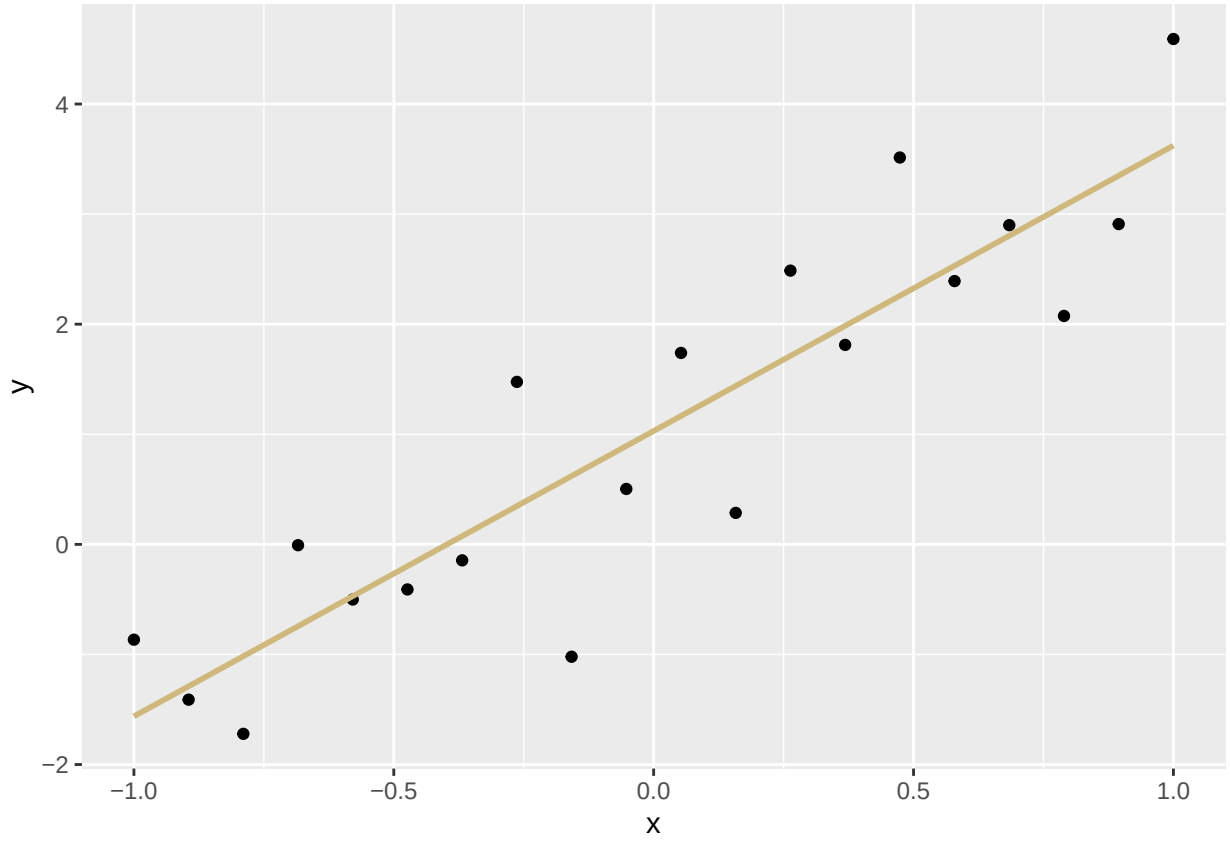
In this exercise, we'll look at the variability of slopes of simple linear regression models fitted to realizations of simulated data.

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 function to understand how fitted slopes can vary, even for the same underlying population.

2. (a) Fit a slope

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

```
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")
```



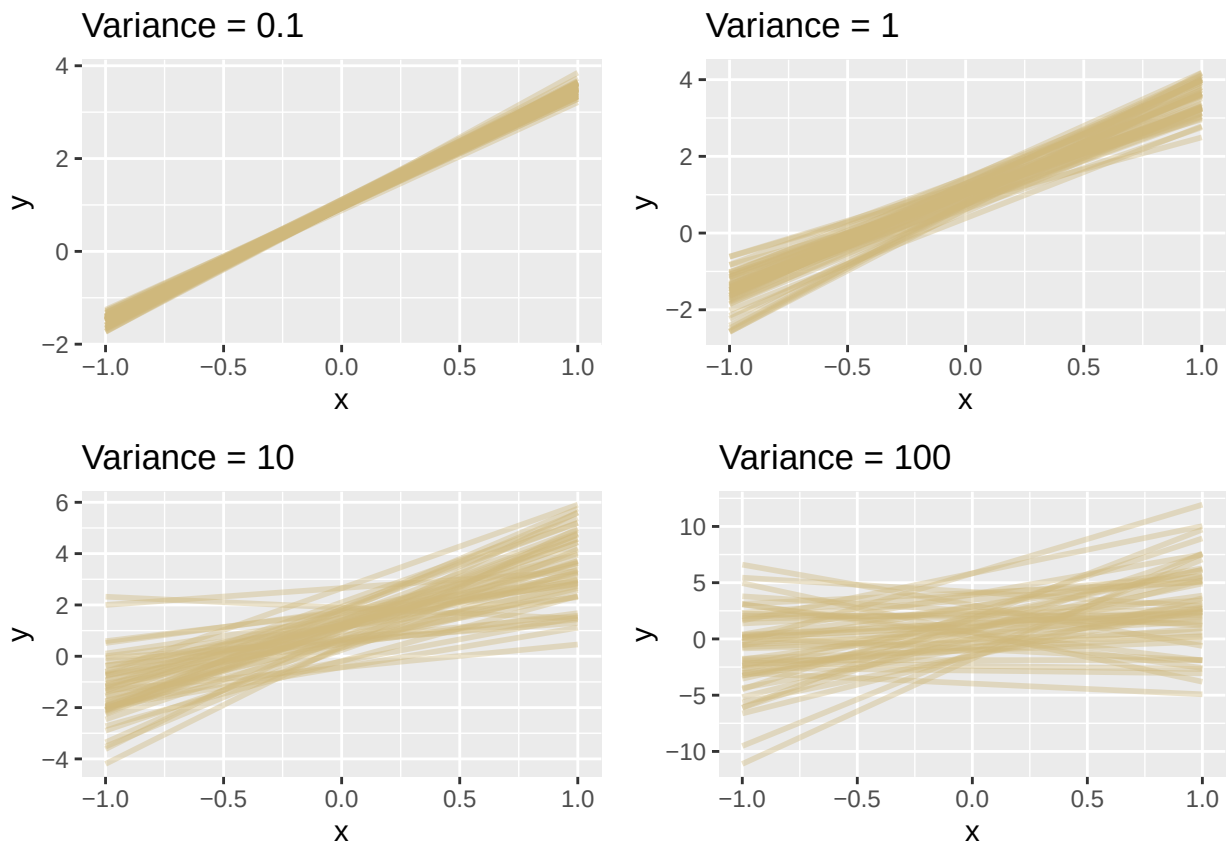
Just based on this plot, how well does our linear model fit the data?

The linear model looks like it fits the data relatively well since that data themselves are fairly spread out.

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 generation function 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.

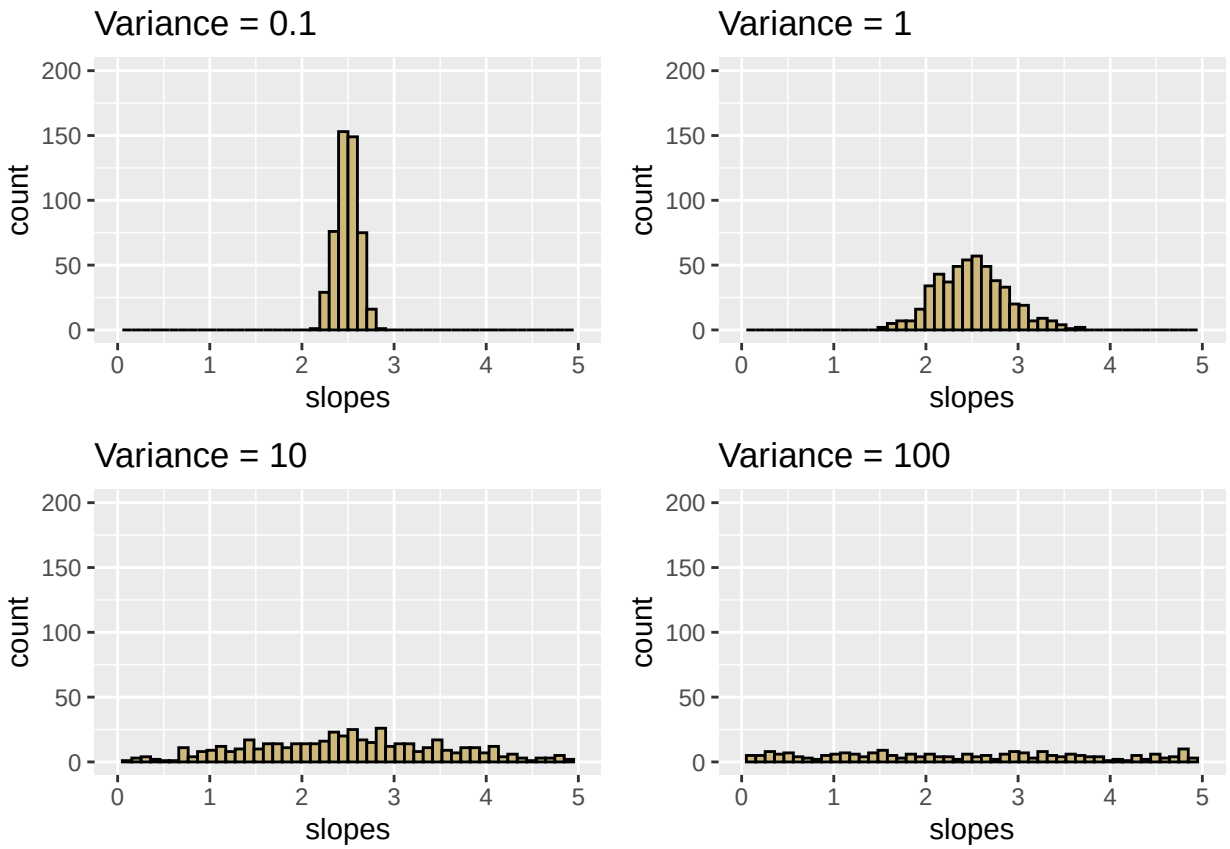


The larger the variance, the less close the resulting slopes are. Therefore, the less the variance there is in the data, the less potential problems there will be in fitting a regression line that is close to the true line of best fit.

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?



The historgrams show that as the variance increases, the kurtosis distribution of the slopes increases. The peak of the curve becomes less defined and the tails continually increase in length. Basically, the distribution curve of the slopes gets smooshed down as variance increases.

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 re-sample 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?

```
# get sample
samp <- sim_data()

# create linear model and get slope
lmod <- lm(y~x, samp)
```

```r
slope <- lmod$coefficients[2]

# look at summary and grab standard error
lmod.sum <- summary(lmod)
se <- lmod.sum$coefficients[2, "Std. Error"]

# calculate t-statistic
alpha <- 0.05
t <- qt(1 - alpha/2, n-1)

# get margins
ci.margin <- abs(t * se)

# calculate confidence interval
lower <- slope - ci.margin
upper <- slope + ci.margin
ci <- c(lower, upper)

# confidence interval view
names(ci) <- c("lower.bound", "upper.bound")
print(ci)
```

```
## lower.bound upper.bound
##    2.161788    3.525864
```

Yes, the known slope, 2.5, falls within the 95% confidence interval given. The value of σ^2 makes the confidence interval wider to account for more variance.