APBI360 Data Analysis and Visualization Lab

Written by Jens Ulrich

January 2024

Example concepts taken from: Douglas et al., 2015. "Neonicotinoid insecticide travels through a soil food chain, disrupting biological control of non-target pests and decreasing soya bean yield." *Journal of Applied Ecology.*

Overview

Day 1:

We will begin by ensuring that R was properly installed and then famaliarize ourselves with an R environment.

Next, we will simulate samples of slug and soybean densities based on a hypothesized relationship between slug herbivory and soybean seedling survival.

After that, we will visualize our and analyze our simulated data.

Day 2:

We will begin by adjusting the simulation settings and then explore how this might impact our analysis and conclusions.

Finally, you will be given a data set to investigate. Imagine this is data that you or another agroecologist collected from a real study! You will be asked to qualitatively and quantitatively summarize these data and make a conclusion about the relationship between slugs and soybeans.

Lab learning outcomes:

Following the completion of the lab, students will be able to:

- Open an R programming environment, execute pre-written R scripts and interpret the outputs of these executions.
- Read a graphical figure that shows a relationship between a continuous predictor (independent) variable and a continuous outcome (dependent) variable.
- Describe the biological meaning of the intercept and slope estimates given by a linear regression model.
- Understand how changes to study design (sample size) and measurement precision impact the qualitative and quantitative assessment of a hypothesis.

You will be asked to submit responses to key prompts embedded in the activities below. There are 14 labelled questions requiring response. Keep your responses in a word document and then submit them on canvas after the second day of the lab.

Day 1

Day 1 - Part 1: Familiarize yourself with an R environment

This section will be demonstrated to the class by TA on the projection screen.

Before we start with the activity, we will execute a few commands in R to ensure that R is installed properly and that we know how to navigate within an R environment. We will walk through Part 1 as a class with the TA showing navigation and operations on the projection screen.

R commands can be executed in two different ways. First, you can paste or type a chunk of code directly into your console and the hit "enter" on your keyboard. Typing or pasting the following print() command in your console and then pressing "enter" should result in your console printing the input text. Entering the simple arithmetic operation below (2 + 2) will return the result of the operation, just as if you were using a calculator.

```
print("slugs eat soybeans")
```

```
## [1] "slugs eat soybeans"
```

```
2 + 2
```

```
## [1] 4
```

Alternatively, you can create a .R file that holds lines of R script. You can select and then press "ctrl+enter" to execute one or more lines of code. This is a preferable way of interacting with R because it will allow you to save multiple lines of code and re-execute them whenever you like. Try creating a .R file in your working directory and then try the following operations by paste/type the next code chunk into your new .R file.

We might store the result of some command as a new object, say "x". After entering the first line of code, see if you can find the this new object and it's value within your programming environment. You should see it listed in the top right window pane if you are using RStudio. You can remove objects from the environment using the rm() command.

```
x <- 2 + 2
x
```

```
## [1] 4
```

```
rm(x)
```

If you wish to reproduce all outputs exactly as those generated by the TA you will need to initialize a random number generator and then run all code in sequence. It's ok if you you don't use the random number initialization, just be aware that your plots and model outputs won't be fully identical to those produced below.

```
set.seed(19)
```

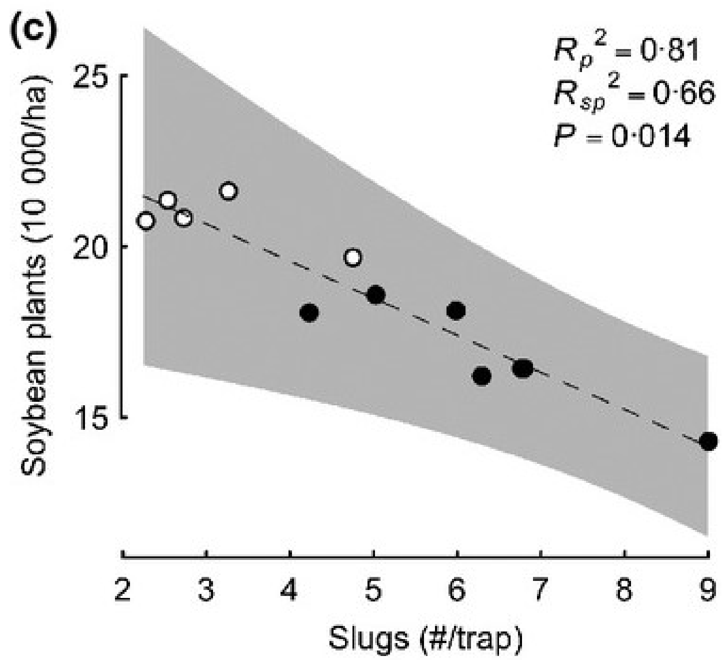
Day 1 - Part 2: Simulate slug and soybean plant densities

Introduction Suppose we wanted to test the sub-hypothesis presented by Douglas et al. 2015: slugs eat (destroy) soybean seedlings thereby limiting soybean plant density. As in Douglas et al., to test this hypothesis we might collect data on the density of slugs in a sample of soybean plots, predicting a negative relationship between slug density and soybean plant density.

Note to TA: As a class we will draw a diagram of our causal hypothesis.

Douglas et al. indeed found a negative relationship, supporting this hypothesis (see figure 3c below). We will recreate this experiment ignoring the additional complexity of variation in neonicotinoid pesticides applied to sample plots. Douglas et al. used neonicotinoids to generate variation in slug densities. Here we might imagine a scenario where slug densities vary naturally among soybean plots.

Note to TA: As a class we will look at the fig 3c and identify the intercept, slope and random variation (precision) around the linear predictor.



Douglas et al., 2015, *Journal of Applied Ecology* - fig 3c

Simulation Create some fake date using a simulation procedure. We will know the relationship between slugs and soybeans (because we will define it before generating our fake data!), and so our visualization and statistical models should return the known relationship that we expect.

Start by defining the dimensions of our experiment. Let's use a sample size equivalent to the one used in the study we intend to replicate. Note that anything following a hashtag is ignored by R. Use hashtags to "commented out" your notes or describe your code without interfering with the R session.

```
# specify a sample size (how many plots are included in the field experiment)
n <- 12
```

Next we will need to generate some values for our independent variable, the average density of slugs per trap in each of the n sample plots. Here we will use R's runif() command. runif() will take three arguments:

(1) n or sample size; (2) a minimum value; and (3) a maximum value. Within the range of the minimum and maxium values, runif() will produce n random numeric values. The probability of generating any value within the range of the minimum and maximum is equal or "uniform".

```
# specify a minimum number of slugs we might expect to see in a trap
# (value estimated from fig 3c)
min_slugs_observed = 2
# specify a maximum number of slugs we might expect to see in a trap
# (value estimated from fig 3c)
max_slugs_observed = 9
# simulate some slug trap data (independent variable), for n plots,
# ranging from minimum to maximum values seen in the field experiment
slugs_per_trap <- runif(n=n, min=min_slugs_observed, max=max_slugs_observed)
```

Great we have some recorded data on slugs per trap! We could even print or plot these data to see the distribution of slug counts that we "recorded". The counts should be fairly evenly distributed within the range of 2 to 9 slugs given use of the runif() function.

```
slugs_per_trap
```

```
##  [1] 2.819914 5.388208 6.558449 2.478777 4.557085 3.567326 4.043171 6.012986
##  [9] 7.856342 7.082365 4.845260 5.148023
```

```
hist(slugs_per_trap,
     main = "",
     xlab = "slugs per trap",
     ylab = "frequency",
     xlim = c(0, 12))
```

QUESTION 1: Why don't we have any plots with 11 or 12 slugs per trap?

Now that we have generated our independent variable data, we can generate dependent data based on a hypothesized relationship. We will assume that there is a linear relationship between slug density and soybean plant density (recall a $y = a + b(x)$ slope-intercept equation from linear algebra). To simulate the outcome of such a relationship we will need to specify two terms: (1) an intercept, i.e., how many soybean plants might you expect to see in a plot given that there are zero slugs plot; (2) a slope, i.e., how much do you expect the density of soybean plants to change for every one unit increase in slugs per trap.

```
# specify an intercept term, i.e.,
# density of soybean plants when there are zero slugs in traps
intercept <- 25 # units are in 10,000 plants / hectare

# specify a slope term, i.e.,
# a '_' change in soybean plant density associated with
# every increase of 1 slug per trap in the plot
slope <- -2
```

QUESTION 2: Our slope term of -2 infers a negative relationship between slugs per trap and soybean plant density. What would a slope term of 1 infer? How about a slope term of 0?

Finally, we will introduce a stochastic or "random" element to our slug-soybean association. In the real world, we might not expect perfect **precision** where a given slugs per trap measurement always corresponds to the exact same density of soybeans. Realistically, we might expect some plots to randomly deviate either slightly lower or slightly higher than expected given our intercept, slope and a measure of slugs per trap.

Assuming normally distributed random variation (bell-curve shaped variation), the actual outcomes fall within 1 standard deviation of the expected value ($a + b(x)$) ~68% of the time and within 2 standard deviations of the expected value (*a + b(x)*) ~95% of the time. E.g. Our intercept of 25, slope of -2 and a measure of 5.5 slugs per trap is expected to yield *25 + -2(5.5) = 14(,000) soybean plants / hectare* on average. **Given a standard deviation of 3**, ~68% of the time plots with 5.5 slugs per trap should have *14(,000) +/- 3(,000)* soybean plants / hectare; ~95% of the time plots with 5.5 slugs per trap should have *14(,000) +/- 6(,000)* soybean plants / hectare.

Don't worry too much if this idea isn't immediately clear! We will follow up on this random element on day 2 of the lab, seeing how increasing it or decreasing it changes our analysis. For now, we will set our standard deviation at 3.

```
# specify precision (how much does the response vary irrespective of the association)
sd <- 3 # standard deviation # units are in 10,000 plants / hectare
```

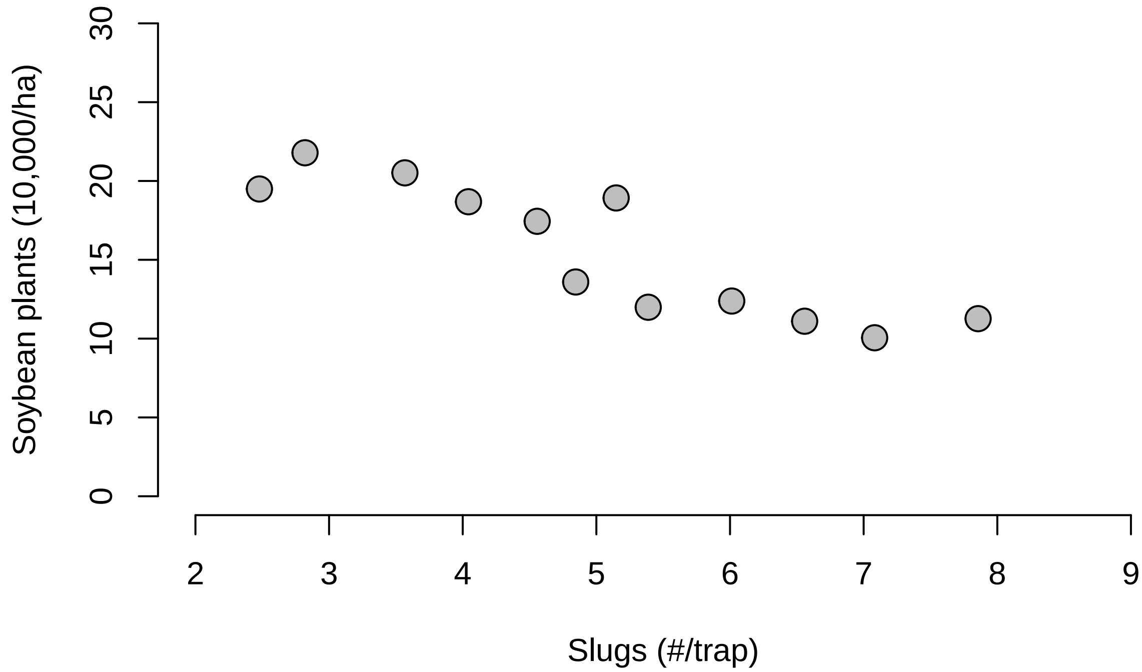
Now that we have defined all of the elements of our imaginary system we are ready to simulate some outcomes. We will simply combine the intercept, slope and measurements into a single linear predictor (again recall the $y = a + b(x)$ formula). Use the rnorm() function to generate soybean densities including both the linear predictor (intercept, slope and some independent data) and an element of normally distributed random variation.

```
# use rnorm() to simulate soybean plant densities
# for "n" plots with 2 to 9 "slugs_per_trap slug"
# an effect size of "slope"
# an intercept of "intercept"
# and a standard deviation of "sd"
linear_model <- (intercept + (slope * slugs_per_trap))
soybean_density <- rnorm(n=n, mean=linear_model, sd=sd)

# join the independent and dependent data into a single 'data frame' structure
mydata <- data.frame(slugs_per_trap, soybean_density)
```

Day 1 - Part 3: Visualization and analysis of simulated data

Preliminary visualization Before we quantitatively analyze patterns in our simulated data set, let's plot the data and conduct a qualitative assessment.

```
# create a plot using base R plotting tools
plot(x = mydata$slugs_per_trap, # independent variable
     y = mydata$soybean_density, # dependent variable
     cex = 1.75, pch = 21, bg = 'gray', # size, shape, and colour of the data points
     xlab = "Slugs (#/trap)", # x-axis title
     ylab = "Soybean plants (10,000/ha)", # y-axis title
     frame = FALSE, # remove frame
     xlim = c(min_slugs_observed, max_slugs_observed), # x-axis limits
     ylim = c(0, 30) # y-axis limits
)
```



QUESTION 3: Which axis in your figure describes the variation in the independent variable? Which axis describes the variation in the dependent variable?

QUESTION 4: Look at the general trend in the data. Does the change in soybean plant density associated with increasing slug counts track with your intuition based on the slope value that we used?

Analysis and final visualization Quantify the association using a linear regression model. the lm() function will find the values of an intercept and slope that in combination have the highest likelihood of producing the data.

We can extract the estimates from the model fit summary.

```
# fit a linear regression model to our data
# lm() fits a linear model
summary(fit1 <- lm(formula = soybean_density ~ slugs_per_trap,
                   data = mydata))
```

```
##
## Call:
## lm(formula = soybean_density ~ slugs_per_trap, data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8016 -1.2490 -0.0653  1.2835  3.5885
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     27.0012     1.9743  13.676 8.47e-08 ***
## slugs_per_trap  -2.2661     0.3742  -6.055 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.064 on 10 degrees of freedom
## Multiple R-squared:  0.7857, Adjusted R-squared:  0.7643
## F-statistic: 36.67 on 1 and 10 DF,  p-value: 0.0001227
```

```r
# save important outputs
# intercept term
(estimate_intercept <- summary(fit1)$coefficients[1,1])
```

```
## [1] 27.00118
```

```r
# effect of slug increase
(estimate_slope <- summary(fit1)$coefficients[2,1])
```

```
## [1] -2.26613
```

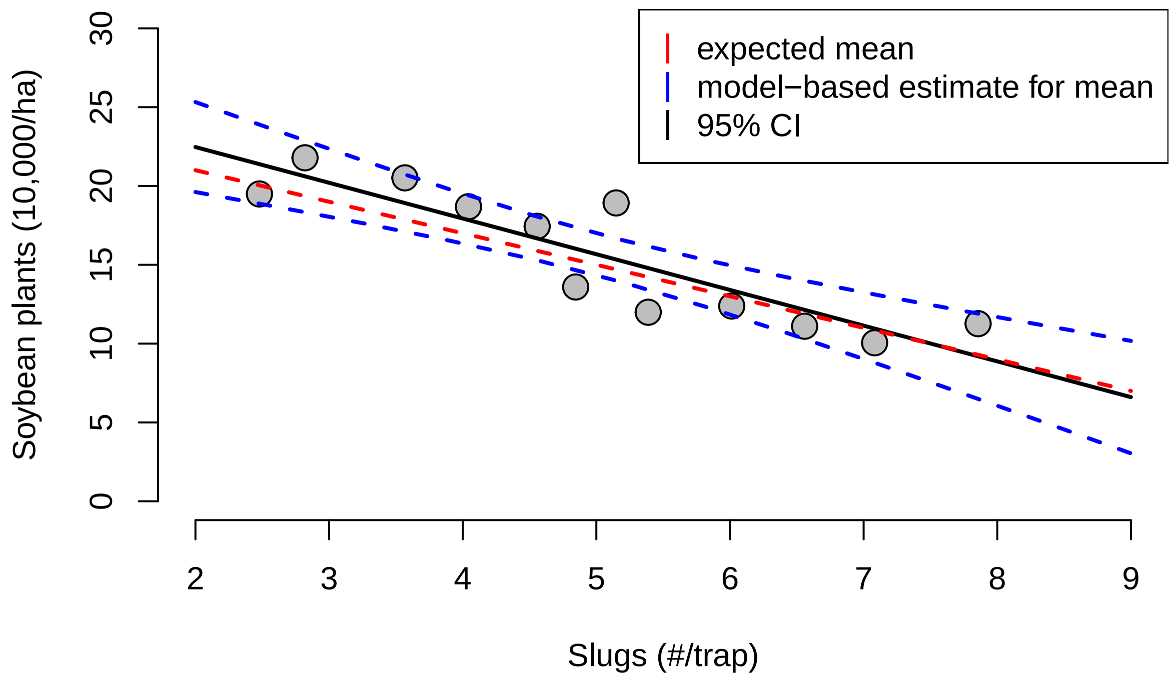
```r
# We can also extract the R-squared value
# (definition)
(R2 <- summary(fit1)$r.squared)
```

```
## [1] 0.785724
```

Last, we will add predictions on our model estimates. Because we know the true values of the intercept and the slope that underly this slug - soybean association, we can assess the accuracy of our model fitting procedure.

```r
# now plot the fit (with confidence intervals)

# first we need to create some new data
# we will make predictions for the mean and confidence across the same range
# of slugs that we "observed" in our simulation
min_slugs_observed <- min_slugs_observed
max_slugs_observed <- max_slugs_observed

# now create some new independent data (slugs_per_trap)
# ranging from min to max and stepping up by interval
newdata <- data.frame(slugs_per_trap = seq(
  min_slugs_observed, max_slugs_observed, length.out=nrow(mydata)))
# View(newdata) # you can view the new data set

# now predict the expected outcome for each value of slugs
# What is the expected soybean density of a plot given a particular slug density?
pred <- predict(object=fit1, newdata, interval = 'confidence')

# create a plot using base R
```

```r
# plot our simulated data
{
  plot(x = mydata$slugs_per_trap, # independent variable
     y = mydata$soybean_density, # dependent variable
     cex = 1.75, pch = 21, bg = 'gray', # size, shape, and colour of the data points
     xlab = "Slugs (#/trap)", # x-axis title
     ylab = "Soybean plants (10,000/ha)", # y-axis title
     frame = FALSE, # remove frame
     xlim = c(min_slugs_observed, max_slugs_observed), # x-axis limits
     ylim = c(0, 30) # y-axis limits
  )

  # plot the predicted mean response for a given number of slugs per trap
  lines(pred[,1] ~ newdata$slugs_per_trap, col = 'black', lwd = 2)

  # Confidence intervals - range in which the 'true' regression line lies
  # given a certain level of confidence (default is 95% confidence).
  # Plot the 95% CI for the mean response across range of slugs per trap
  lines(pred[,2] ~ newdata$slugs_per_trap, col = 'blue', lty = 2, lwd = 2)
  lines(pred[,3] ~ newdata$slugs_per_trap, col = 'blue', lty = 2, lwd = 2)

  # Now let's add the 'true' regression line and see how close our model got
  expected_means <- intercept + slope * newdata$slugs_per_trap
  lines(expected_means ~ newdata$slugs_per_trap, col = 'red', lty = 2, lwd = 2)

  legend("topright",
         legend = c("expected mean", "model-based estimate for mean", "95% CI"),
         pch = "|", col = c("red", "blue", "black"))
}
```



QUESTION 5: What is the estimated intercept and slope? Does the uncertainty around the estimates contain the true values that we used to simulate the data? (Hint: compare the true mean relationship (red dashed line) to the confidence interval (blue dashed lines))

QUESTION 6: Why might the estimates given by our lm() call be slightly different from the ones we used to simulate the data?

QUESTION 7: What does the R-squared included in your model output tell you?

————- END DAY 1 ————-

Day 2

Day 2 - Part 1: Adjust the simulation settings

First, we will try adjusting the simulation settings. We'll take a look at how this effects the our qualitative and quantitative assessments of the association between slugs and soybeans

Download the simulation function To make reproduction of the simulation easy, we've wrapped all of the simulation code from Day 1 into a function stored in a .R file. This will allow us to tweak the simulation settings and instantly see the consequences on the data and our analysis. Save the file "simulation_function.R" in your working directory. Make sure you do not change the name of the file.

Direct R towards the simulation function This section will be demonstrated to the class by TA on the projection screen.

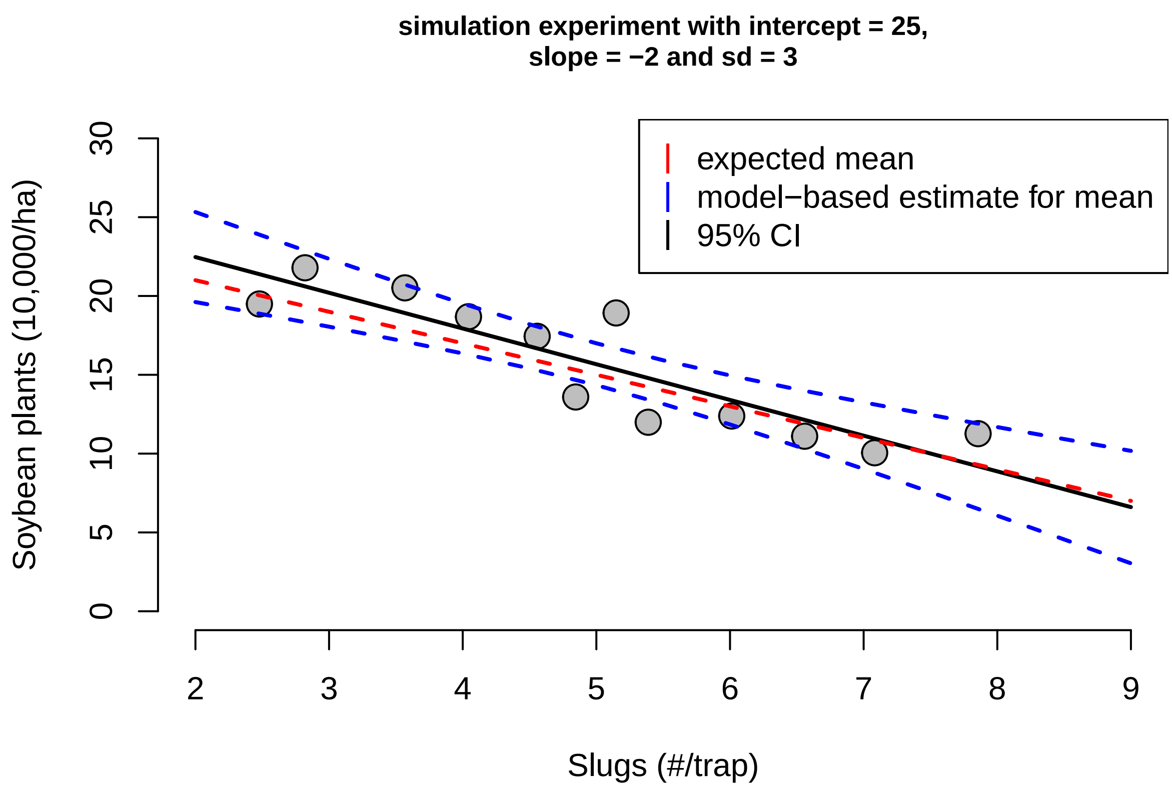
source() points the environment to a different file

The source() function will point your R session towards a file in your directory. Starting the file argument with "./" will place tell R to look in your current working directory.

```r
# the "simulation_function.R" file holds the simulation function.
source(file="./simulation_function.R")
```

To make sure that you are properly connected to the simulation function. Go ahead and recreate the figure that we made as a class by calling the simulate_slugs_and_soybeans() function held in your new file, using the same simulation settings that we used for the simulation in Day 1.

```r
# Now we can just run the following line of code to regenerate our original plot!
set.seed(19)
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                    min_slugs_observed=2, max_slugs_observed=9,
                                    intercept=25, slope=-2, sd=3,
                                    n_reps=1)
```




```
print(paste0("intercept = ", signif(my_simulated_data$estimate_intercept, digits=3)))
```

```
## [1] "intercept = 27"
```

```
print(paste0("slope = ", signif(my_simulated_data$estimate_slope, digits=3)))
```

```
## [1] "slope = -2.27"
```

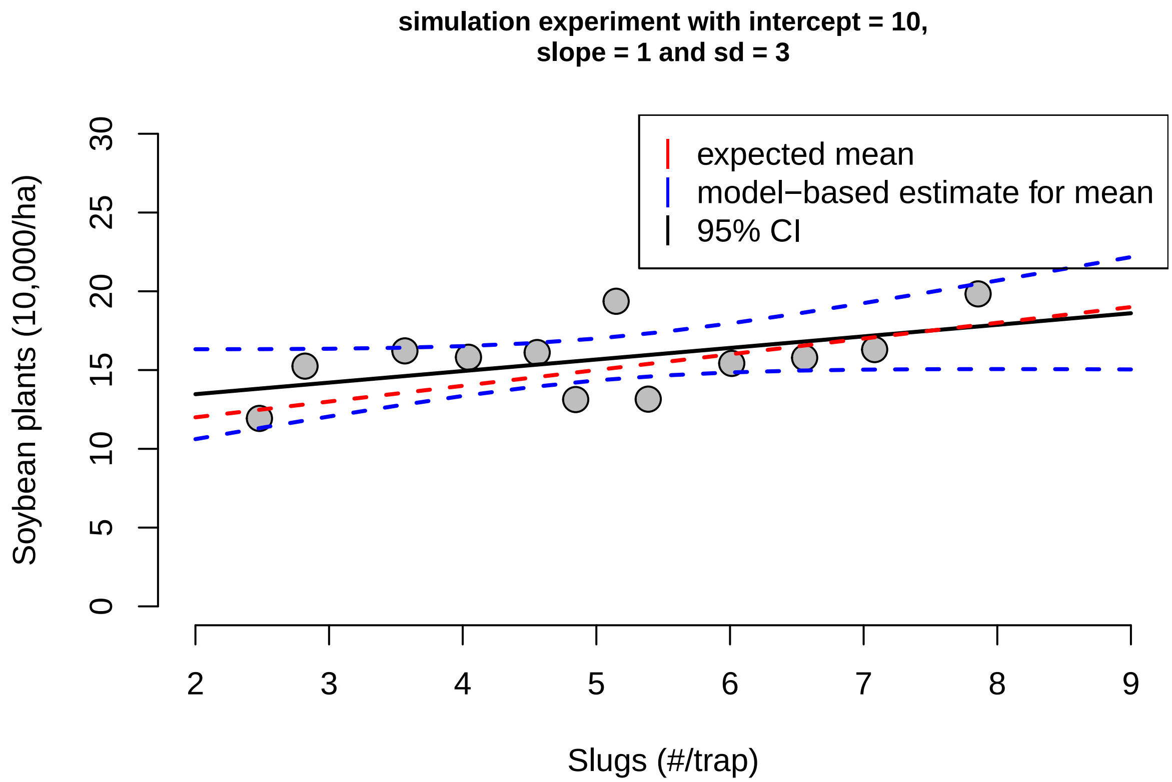
```
print(paste0("R-squared = ", signif(my_simulated_data$R2,3)))
```

```
## [1] "R-squared = 0.786"
```

Reconduct the simulation Next we will adjust our simulation settings to see how this changes the visual spread of the data that emerge, the results of our statistical analysis and our conclusions about the hypothesis introduced on Day 1.

Using the simulation function, alter the intercept and/or the slope. For example, I've chosen a new intercept of 10 and a slope of 1. Feel free to choose any values that you think you might realistically observe in a slug - soybean experiment! Rerun the simulation 1 time (n_reps=1).

```
set.seed(19)
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                   min_slugs_observed=2, max_slugs_observed=9,
                                   intercept=10, slope=1, sd=3,
                                   n_reps=1)
```




```
print(paste0("intercept = ", signif(my_simulated_data$estimate_intercept, digits=3)))
```

```
## [1] "intercept = 12"
```

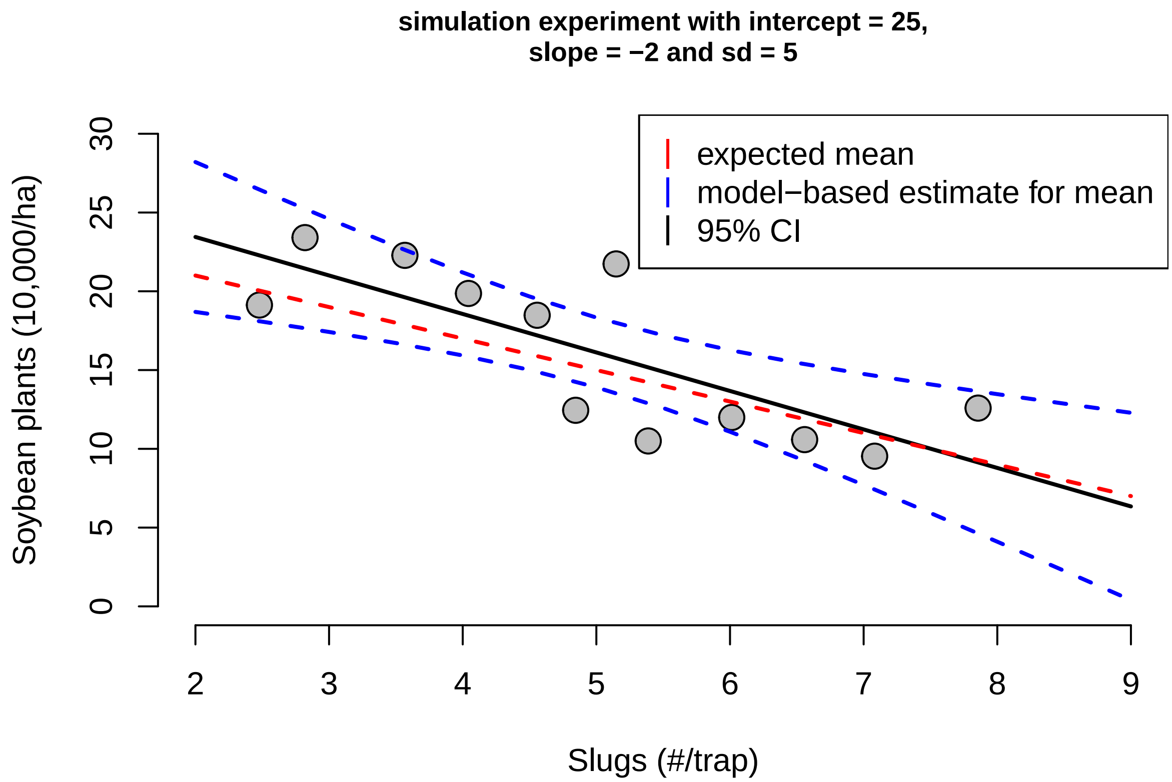
```
print(paste0("slope = ", signif(my_simulated_data$estimate_slope, digits=3)))
```

```
## [1] "slope = 0.734"
```

QUESTION 8: Describe your visual interpretation of how the data and association changed after altering the input intercept and slope. Is the true relationship (red line) still captured within the confidence interval associated with your statistical model's estimates for the intercept and slope (blue lines). Do the results of your analysis still support the hypothesis that slugs destroy soybean plants? Copy and paste the plot WITH THE SIMULATION PARAMETER VALUES THAT YOU CHOSE at the end of your response to this question

Now return to our original intercept and slope values and then decrease the precision (increase the standard deviation) of our experiment. Change the sd from 3 to 5 and rerun the simulation 1 time (n_reps=1).

```
set.seed(19)
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                    min_slugs_observed=2, max_slugs_observed=9,
                                    intercept=25, slope=-2, sd=5,
                                    n_reps = 1)
```




```
print(paste0("intercept = ", signif(my_simulated_data$estimate_intercept, digits=3)))
```

```
## [1] "intercept = 28.3"
```

```
print(paste0("slope = ", signif(my_simulated_data$estimate_slope, digits=3)))
```

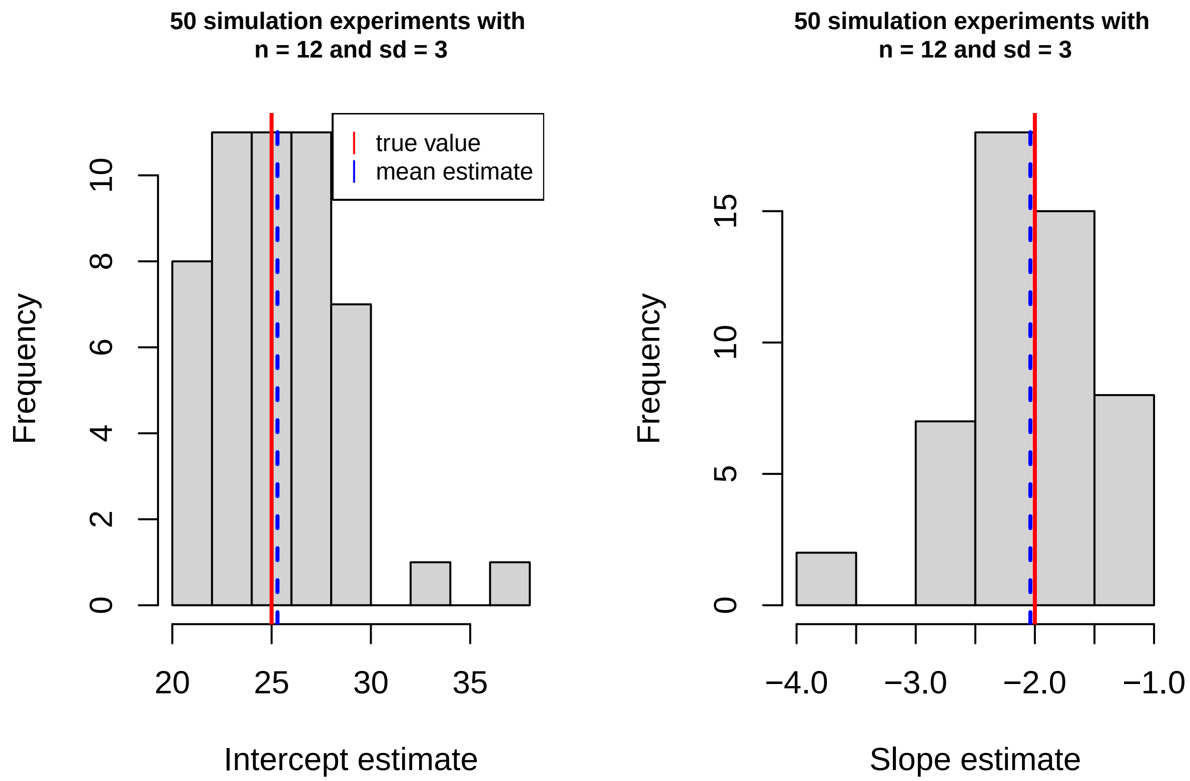
```
## [1] "slope = -2.44"
```

QUESTION 9: Describe your visual interpretation of how the data and association changed after altering the precision (compared to the simulation with intercept=25, slope=-2 and sd=3). Is the true relationship (red line) still captured within the bounds of the confidence interval associated with your statistical model's estimates for the intercept and slope (blue lines). Do the results of your analysis still support the hypothesis? Copy and paste your PLOT WITH THE SIMULATION PARAMETER VALUES THAT YOU CHOSE at the end of your response to this question

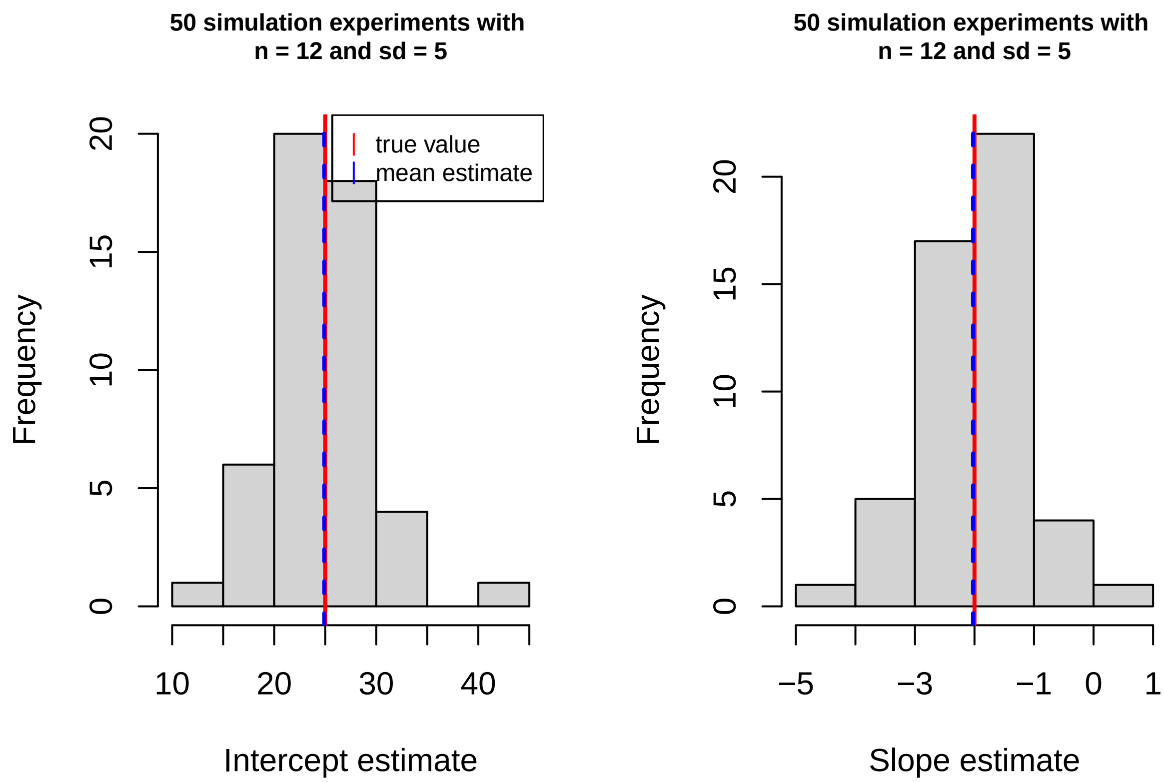
Let's imagine we could conduct the experiment 100 times with higher precision(sd=3) and 100 times (n_reps=100) with lower precision (sd=5). When n_reps is greater than 1, the simulation function will no longer reproduce a plot of data for a single experiment, because we are simulating multiple experiments! Instead, the function will return a plot showing the distribution of estimates for the intercept

```
# Now we can just run the following line of code to regenerate our original plot!
set.seed(19)
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
```

```
                                      min_slugs_observed=2, max_slugs_observed=9,
                                      intercept=25, slope=-2, sd=3,
                                      n_reps = 50)
```



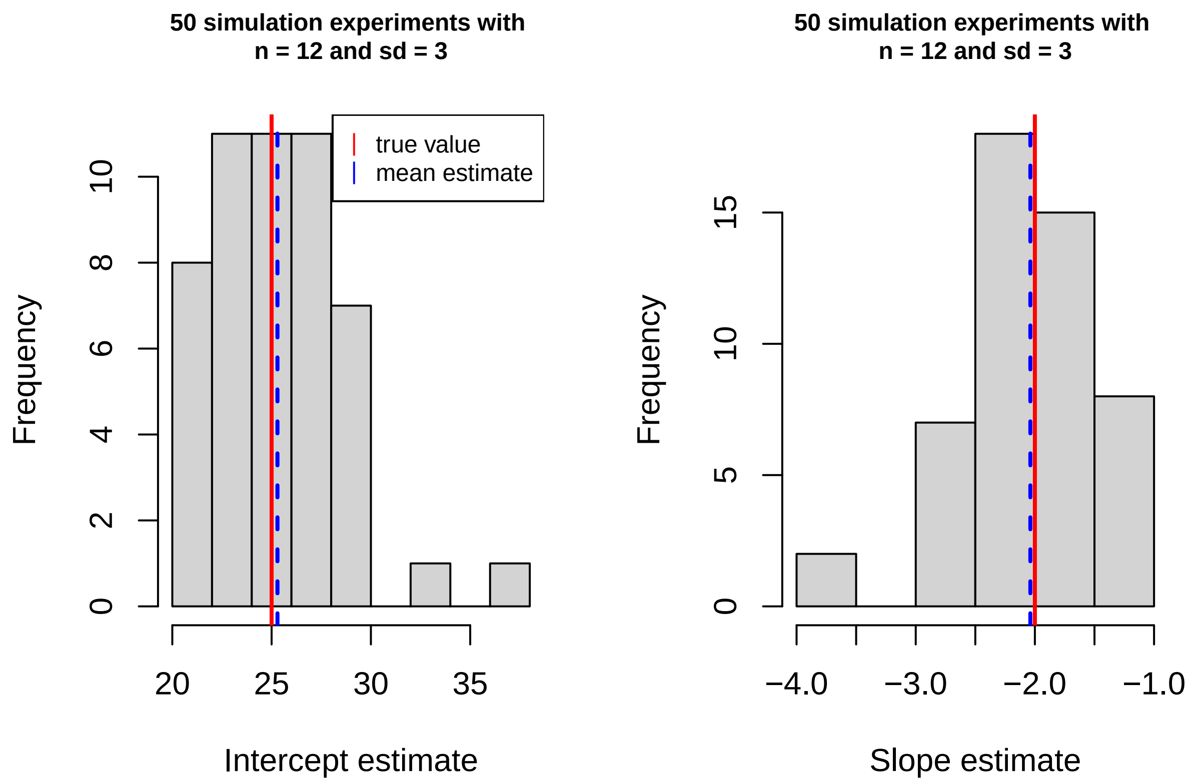

```
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                      min_slugs_observed=2, max_slugs_observed=9,
                                      intercept=25, slope=-2, sd=5,
                                      n_reps = 50)
```



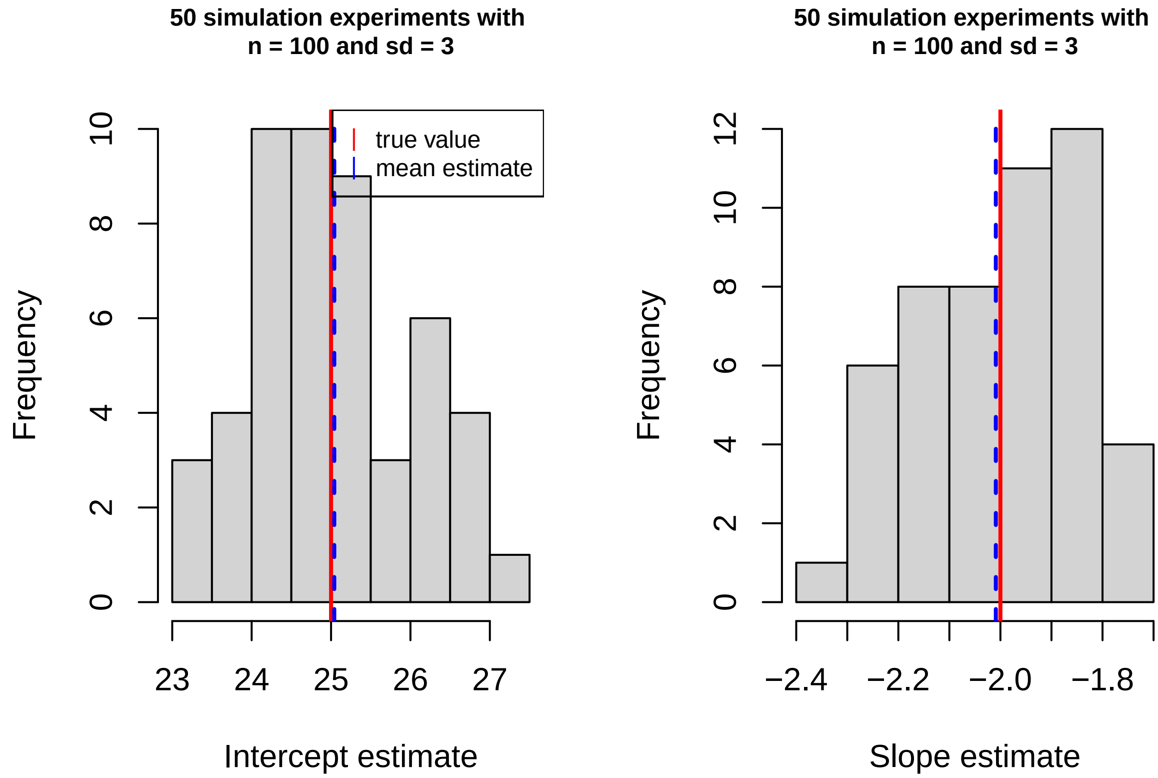
QUESTION 10: Compare the intercept estimates as well as the slope estimates for when sd=3 versus when sd=5. Which set of simulations tends to return estimates for the intercept and slope that are closer to the true values of the intercept and slope used to simulate the data? (Hint: look at how spread out the estimates are around the true value)

Finally, let's adjust the sample size used in our simulation. Say we had the opportunity to measure slugs and soybean plant density in 100 field plots rather than only in 12:

```
# Now we can just run the following line of code to regenerate our original plot!
set.seed(19)
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                   min_slugs_observed=2, max_slugs_observed=9,
                                   intercept=25, slope=-2, sd=3,
                                   n_reps = 50)
```




```
my_simulated_data <- simulate_slugs_and_soybeans(n=100,
                                    min_slugs_observed=2, max_slugs_observed=9,
                                    intercept=25, slope=-2, sd=3,
                                    n_reps = 50)
```



QUESTION 11: Rerun the previous comparison, but replace n=100 with a sample size of your choice. It could be smaller or larger than Douglas et al sample size of 12! Your choice should be a whole integer number. Compare the intercept estimates as well as the slope estimates for when n=12 versus when n=(your choice). Which set of simulations tends to return estimates for the intercept and slope that are closer to the true values of the intercept and slope used to simulate the data? (Hint: look at how spread out the estimates are around the true value). INCLUDE YOUR PLOTS IN YOUR RESPONSE SO THAT I CAN SEE WHAT SAMPLE SIZE YOU CHOSE

Day 2 - Part 2: Visualize and analyze some collected data

Suppose know that you have a good idea of your system and what kind of data would support your hypothesis, you design and conduct a real field experiment measuring slugs per trap and soybean.

You've already entered your data into a .csv file using excel.

Let's first get a summary of the data

```
my_real_data <- read.csv("./new_slug_and_soybean_data.csv")

# view the data
(my_real_data)
```

```
##    plot slugs_per_trap soybean_density
## 1     1       1.693609       28.266239
## 2     2       9.177752        7.233150
```

```
## 3     3      10.426217        6.426407
## 4     4       3.693819       26.819236
## 5     5       2.693481       28.580424
## 6     6       1.338956       29.487020
## 7     7       2.787850       34.380331
## 8     8       7.416654       18.990671
## 9     9       1.228777       35.209413
## 10   10       1.083248       31.972006
## 11   11       4.926972       24.688956
## 12   12       9.138806       22.185393
## 13   13       4.762485       28.647695
## 14   14       4.808122       22.618629
## 15   15       3.649184       22.263439
## 16   16       5.393343       21.313410
## 17   17       5.576072       21.163846
## 18   18       6.407075       20.572363
## 19   19       7.656798       17.514104
## 20   20       2.126989       32.311818
```

```r
# how many plots did we collect data from
print(paste0("data were collected from ", nrow(my_real_data), " field plots."))
```

```
## [1] "data were collected from 20 field plots."
```

```r
# mean slugs per trap measurement
mean(my_real_data$slugs_per_trap)
```

```
## [1] 4.79931
```

```r
# mean soybean density
mean(my_real_data$soybean_density)
```

```
## [1] 24.03223
```

Finally, we will repeat our analysis that we conducted on the simulated data to determine if there is really any association in our "real" data.

```r
# fit a linear regression model to our data
# lm() fits a linear model
summary(fit2 <- lm(formula = soybean_density ~ slugs_per_trap,
                   data = my_real_data))
```

```
##
## Call:
## lm(formula = soybean_density ~ slugs_per_trap, data = my_real_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1219 -1.7900 -0.2479  1.4465  8.7354
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     35.7358     1.6236  22.011 1.83e-14 ***
## slugs_per_trap  -2.4386     0.2927  -8.331 1.37e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.639 on 18 degrees of freedom
## Multiple R-squared:  0.7941, Adjusted R-squared:  0.7826
## F-statistic:  69.4 on 1 and 18 DF,  p-value: 1.37e-07
```

```
# save important outputs
# intercept term
(estimate_intercept <- summary(fit1)$coefficients[1,1])
```

```
## [1] 27.00118
```

```
# effect of slug increase
(estimate_slope <- summary(fit1)$coefficients[2,1])
```

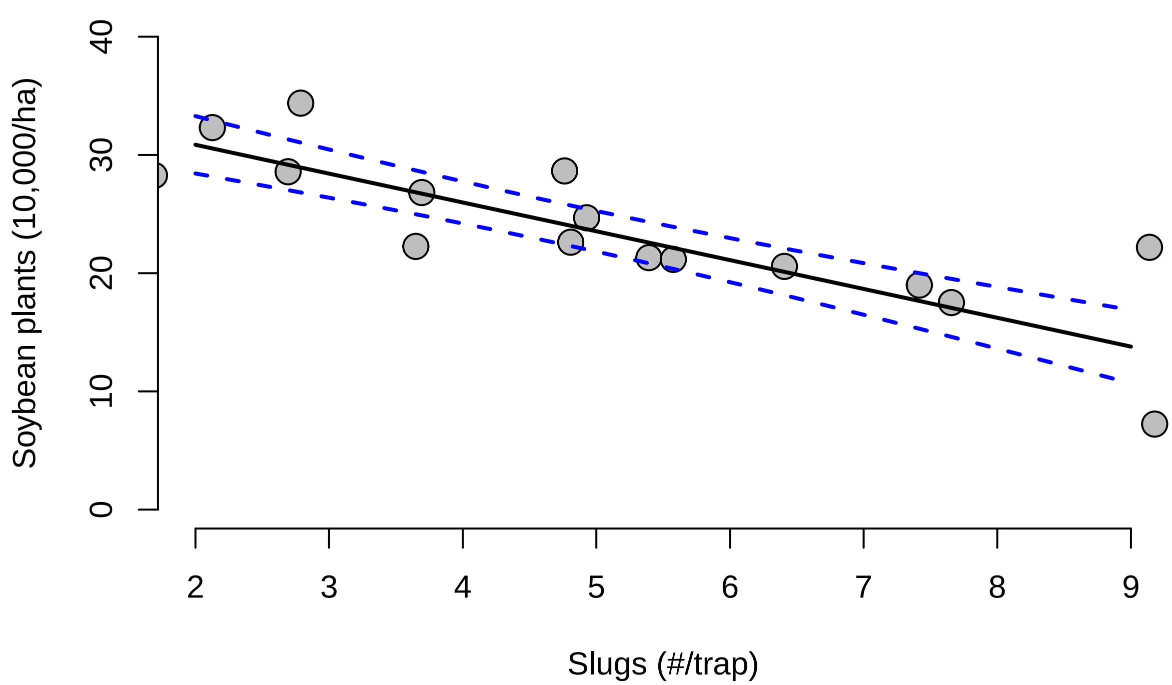
```
## [1] -2.26613
```

```
# We can also extract the R-squared value
# (definition)
(R2 <- summary(fit1)$r.squared)
```

```
## [1] 0.785724
```

```
# now plot the fit (with confidence intervals)

# first we need to create some new data
# we will make predictions for the mean and confidence across the same range
# of slugs that we "observed" in our simulation
# predict the values for every slug_interval slugs added

# now create some new independent data (slugs_per_trap)
# ranging from min to max and stepping up by interval
newdata2 <- data.frame(slugs_per_trap = seq(
  min_slugs_observed, max_slugs_observed, length.out=nrow(my_real_data)))
# View(newdata) # you can view the new data set

# now predict the expected outcome for each value of slugs
# What is the expected soybean density of a plot given a particular slug density?
pred2 <- predict(object=fit2, newdata2, interval = 'confidence')

# create a plot using base R
# plot our simulated data
{
  plot(x = my_real_data$slugs_per_trap, # independent variable
     y = my_real_data$soybean_density, # dependent variable
     cex = 1.75, pch = 21, bg = 'gray', # size, shape, and colour of the data points
     xlab = "Slugs (#/trap)", # x-axis title
     ylab = "Soybean plants (10,000/ha)", # y-axis title
```

```
    frame = FALSE, # remove frame
    xlim = c(min_slugs_observed, max_slugs_observed), # x-axis limits
    ylim = c(0, 40) # y-axis limits
  )

  # plot the predicted mean response for a given number of slugs per trap
  lines(pred2[,1] ~ newdata2$slugs_per_trap, col = 'black', lwd = 2)

  # Confidence intervals - range in which the 'true' regression line lies
  # given a certain level of confidence (default is 95% confidence).
  # Plot the 95% CI for the mean response across range of slugs per trap
  lines(pred2[,2] ~ newdata2$slugs_per_trap, col = 'blue', lty = 2, lwd = 2)
  lines(pred2[,3] ~ newdata2$slugs_per_trap, col = 'blue', lty = 2, lwd = 2)

}
```



QUESTION 12: Paste your plot of the data and model predictions. Describe your visual interpretation of the data. Does soybean density tend to increase, decrease or stay the same as slugs per trap increases? Do the soybean plant densities tend to be close to the expected values (mean trend line) given the number of slugs per trap OR do they vary widely?

QUESTION 13: What are the intercept and slope estimates? What are the biological meanings of these estimates?

QUESTION 14: Do the data support the hypothesis that slugs consume soybean seedlings/plants?

———————- END DAY 1 ———————-