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 familiarize ourselves with an R environment.

Next, we will simulate samples of slug and soybean densities based on a hypothesized effect of slug herbivory on soybean plant survival.

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

Day 2:

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

Finally, you will be given a data set to investigate. 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 independent variable and a continuous dependent variable.
- Describe the ecological 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.
- Apply a linear regression model to a new data set to test an ecological 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. First, the TA should orient you to the 4 main display panels in your RStudio environment.

Before we start with the activity, we will execute a few commands in R to ensure that R is installed properly and that we can navigate within an R environment.

After opening your R environment, R commands can be executed in two different ways. First, you can paste or type a chunk of code directly into the console and then hit "enter". Typing or pasting the 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
```

We might store the result of some command as a new object, say "new_object". 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.

```
new_object <- 2 + 2
new_object
```

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

```
rm(new_object)
```

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

Alternatively, you can create an .R file that holds lines of R script. 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. For example, when we come back to class on Day 2, you can rerun your saved R code rather than retyping everything.

Before creating a .R file we will define a working directory - the folder on your computer where all of the files for this project will be stored. Make a folder somewhere on your computer that is accessible for you. Give the folder an informative name, e.g., "apbi_360_R_activity". To set the working directory either select the gear icon menu in the bottom right panel of RStudio, then click on "Set as working directory". Using RStudio's interface is the easiest way to do this. Alternatively you can manually enter the working directory:

```
# To manually enter the working directory, use the setwd() function:
# setwd("./Documents/apbi_360_R_activity")
# replace "./Documents/apbi_360_R_activity" with the directory on your computer
```

Again, the TA will show you how to set the working directory. Please ask for help if you have difficulties.

Now that we've set up a folder where we want to store our materials for the lab, let's try creating a .R file. Click on the new file icon to make a new .R file. If you've navigated to your working directory folder in the bottom right hand corner of your RStudio panel, you should see this new file pop up in that folder. Confirm that the file is there and give it an informative name, e.g., "apbi360_R_activity.R".

Now paste/type the previous print or calculation code chunks into your new .R file. To run the commands, highlight all of the lines of code that you would like to run and then press "ctrl+enter" on your keyboard. You can also place your cursor anywhere on any line of code and press "ctrl+enter" to run that line of code.

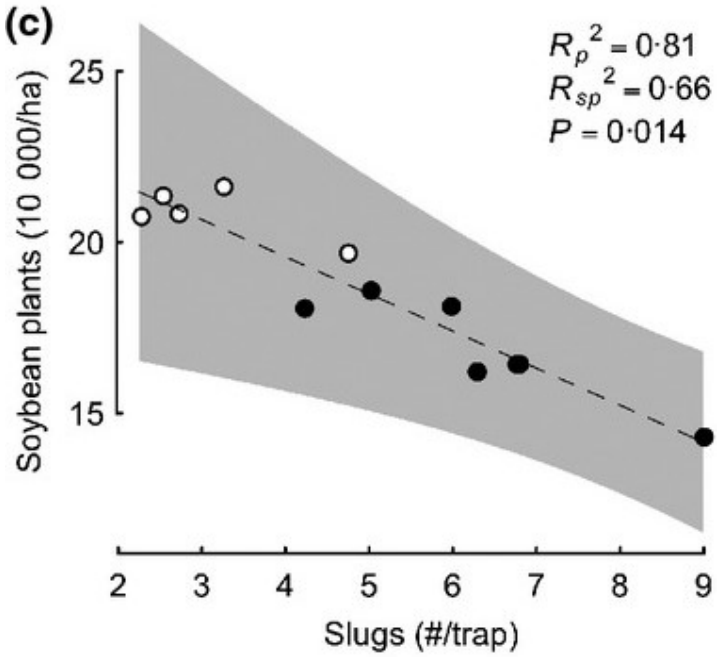
Day 1 - Part 2: Simulate slug and soybean plant densities

Introduction

Suppose we wanted to test the idea presented by Douglas et al. 2015: slugs eat (destroy) soybean seedlings thereby reducing soybean plant density. We might test this by collecting data on the number of slugs and the density of soybean plants in a sample of soybean plots, predicting a negative association between slug counts and soybean plant density.

Douglas et al. found a negative association between slug count and soybean plant establishment, supporting the idea that slugs eat soybean seedlings (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

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 the following 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.

```
# initialize random number generator
set.seed(19)
```

We will create and then visualize/analyze some fake date using a simulation procedure. We will know the association between slugs and soybeans (because we will define it during our simulation), and so our visualization and statistical models should return the association that we expect.

Start by defining the dimensions of the experiment. Let's use a sample size equivalent to the one used in the study we intend to replicate (n = 12 plots). Note that anything following a hashtag is ignored by R. Use hashtags to "comment 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)
# note that any lines of code beginning with "#" will not be read by R.
# You can use the "#" symbol to take notes or make some comments about your code.
n <- 12
```

Next we will need to generate some values for our independent variable, the average measure 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 maximum 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
min_slugs_observed <- 2
# specify a maximum number of slugs we might expect to see in a trap
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)
```

We have some 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

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? How might you change the runif() arguments to allow for the possibility of plots with these higher numbers of slugs per trap?

Now that we've generated our independent variable data, we can generate dependent data based on some association that we define. 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 where y is the soybean plant density and x is the measure of slugs per trap). To simulate the outcome of such a relationship we will need to specify two parameters: (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 may change 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 we are ready to simulate some outcomes. We will combine the intercept, slope and slug count measurements into a 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 (the sd) for a finite number of samples (n).

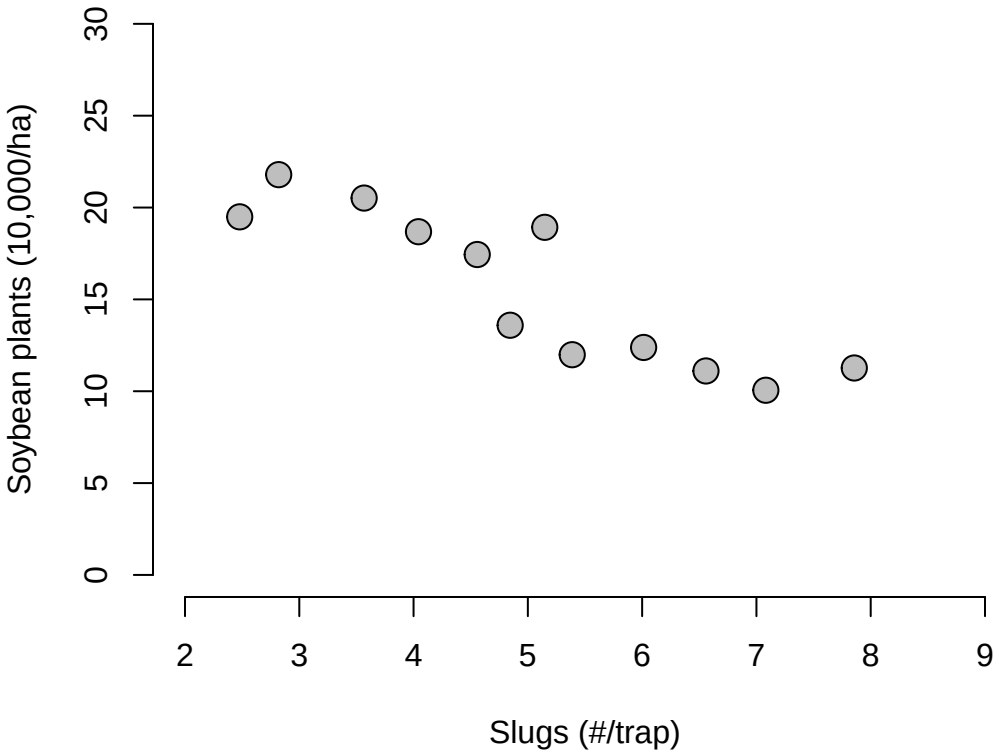
```
# use rnorm() to simulate soybean plant densities
# for "n" plots with 2 to 9 "slugs_per_trap slug"
# an effect size of "slope"
# an intercept of "intercept"
# and a standard deviation of "sd"
linear_model <- (intercept + (slope * slugs_per_trap))
soybean_density <- rnorm(n=n, mean=linear_model,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

```r
# create a plot using base R plotting tools
par(mfrow=c(1,1), mar = c(5, 5, 2, 5)) # Set the margin on the sides to 5

plot(x = mydata$slugs_per_trap, # independent variable
     y = mydata$soybean_density, # dependent variable
     # size, shape, and colour of the data points
     cex = 1.75, pch = 21, bg = 'gray',
     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: Based on a visual assessment of the plot, describe the general association between slugs per trap and soybean density (as slugs per trap increases, does soybean density increase, decrease or stay the same?). Does this association agree with your intuition based on the input slope value of -2?

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.

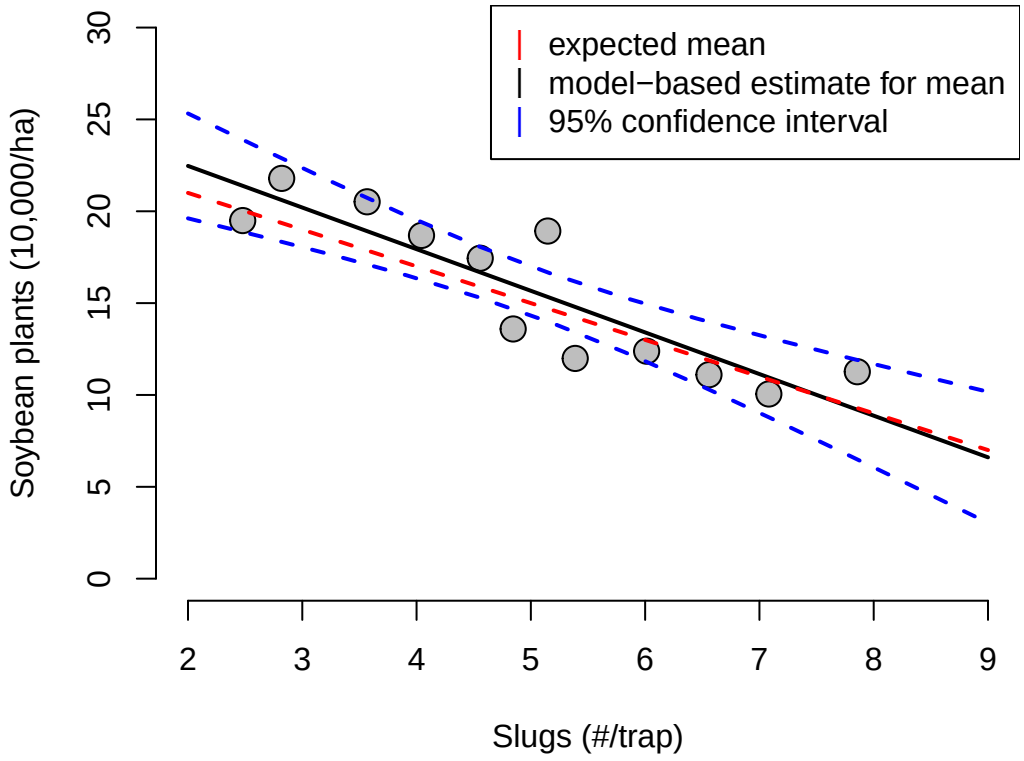
We can extract the estimates from the model fit summary.

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

# save important outputs
# intercept term
(estimate_intercept <- summary(fit1)$coefficients[1,1])
# effect of slug increase
(estimate_slope <- summary(fit1)$coefficients[2,1])
```

Last, we will add predictions for the expected mean value of soybean density for any given slugs per trap measurement based on our model estimates. Because we know the true values of the intercept and the slope that underlie the association, we can assess the accuracy of our model fitting procedure.

```
# 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 equal intervals
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
# plot our simulated data
{
  par(mfrow=c(1,1), mar = c(5, 5, 2, 5)) # Set the margin on the sides to 5

  plot(x = mydata$slugs_per_trap, # independent variable
     y = mydata$soybean_density, # dependent variable
     # size, shape, and colour of the data points
     cex = 1.75, pch = 21, bg = 'gray',
     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% confidence interval"),
         pch = "|", col = c("red", "black", "blue"))
}
```



QUESTION 5: What is the estimated intercept and slope?

QUESTION 6: Do the estimates for the intercept and slope reasonably describe the association between slugs and soybeans that we simulated? Hint: compare the expected mean association (red dashed line) to the model-based estimated mean association (black line) and confidence interval around the model-based estimate for the mean association (blue dashed lines)

QUESTION 7: Why might the estimates given by our linear regression model call be slightly different from the ones we used to simulate the data?

Take Home: Linear regressions describe linear associations between independent and dependent variables. The intercept term describes the expected value of the dependent variable at the baseline level of the independent variable (in this case the expected soybean density at a level of 0 slugs). The slope term describes the change in the dependent variable associated with an increase in one unit of the independent variable (in this case the change in expected soybean density associated with an increase of 1 slugs per trap).The linear regression estimates may deviate from the true underlying values because we specified a finite sample size where the outcomes are subject to additional random variation (in this case we specified 12 sites where actual soybean density tends to be within +/- 3 thousand of the expected mean for a given slugs per trap count).

————- END DAY 1 ————-

Day 2

Day 2 - Part 1: Adjust the simulation settings

We will try adjusting the simulation settings. We'll examine how this effects our qualitative and quantitative assessments of the slug - soybean hypothesis.

Specify your working directory

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

The following activities will require you to interact with another .R file and, later, a .csv spreadsheet of new data. In order to interact with these other files you will need to set them in your working directory so that your computer will be able to find them.

Download the simulation_function.R and the new_slug_and_soybean.csv file into this same folder that you were using during the previous day. Use getwd() function to confirm that you're computer still recognizes the same working directory. The working directory may have changed if you restarted your computer or opened a different project in RStudio.

```
getwd() # get current working directory

# setwd("C:/Users/my_name/Documents/R/APBI360") # set working directory
# ^ your working directory may look something like what I've pasted above.
# ^ replace the above with the path that you are using on your own computer.
# ^ remove the hastag at the front of the above line.
# ^ make sure that you use forward slashes.

list.files() # print out the files in your working directory.
# You should see the simulation function and new data files printed out here
```

Download and source the simulation function

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

To make reproduction of the simulation easy, we've wrapped all of the simulation code from Day 1 into a function stored in an .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.

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.

```
# 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 during the previous lab day by calling the simulate_slugs_and_soybeans() function held in your new file. Use the same simulation inputs that we used for the previous simulation.

```
# Now we can just run the following line of code to resimulate
# our original data and regenerate our original plot!
set.seed(19)
```

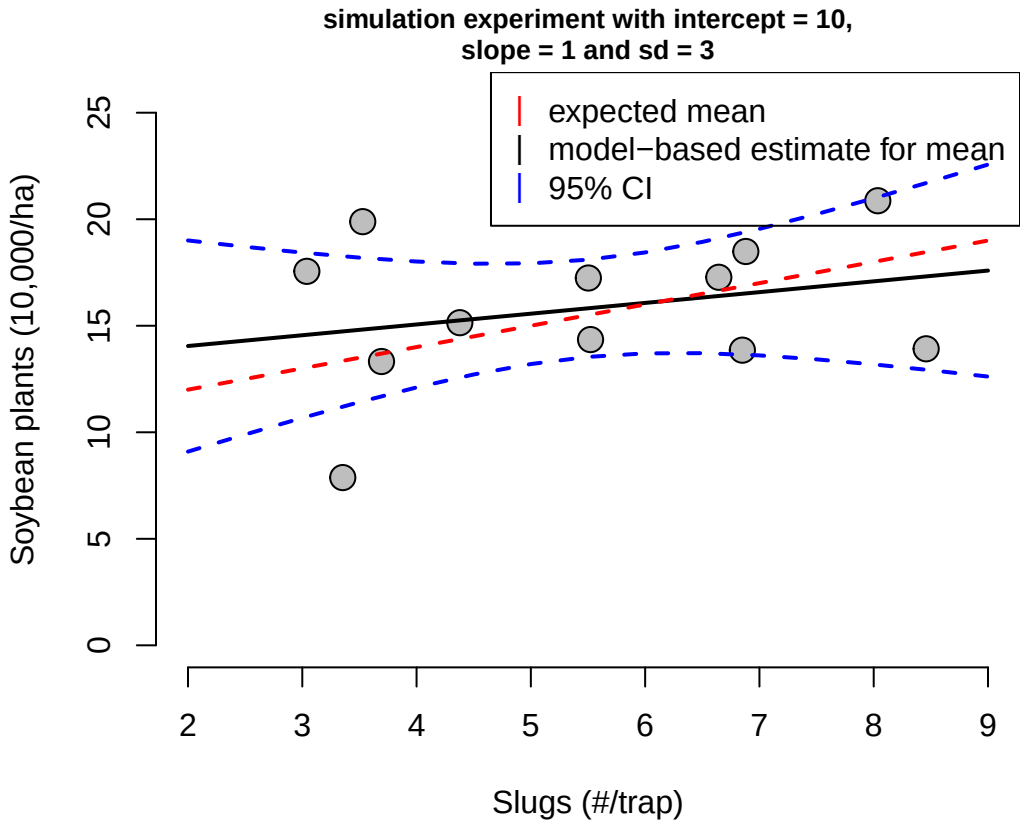
```
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                    min_slugs_observed=2, max_slugs_observed=9,
                                    intercept=25, slope=-2, sd=3,
                                    n_reps=1)

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

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).

```
# simulate data with a new intercept and/or slope
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)
```

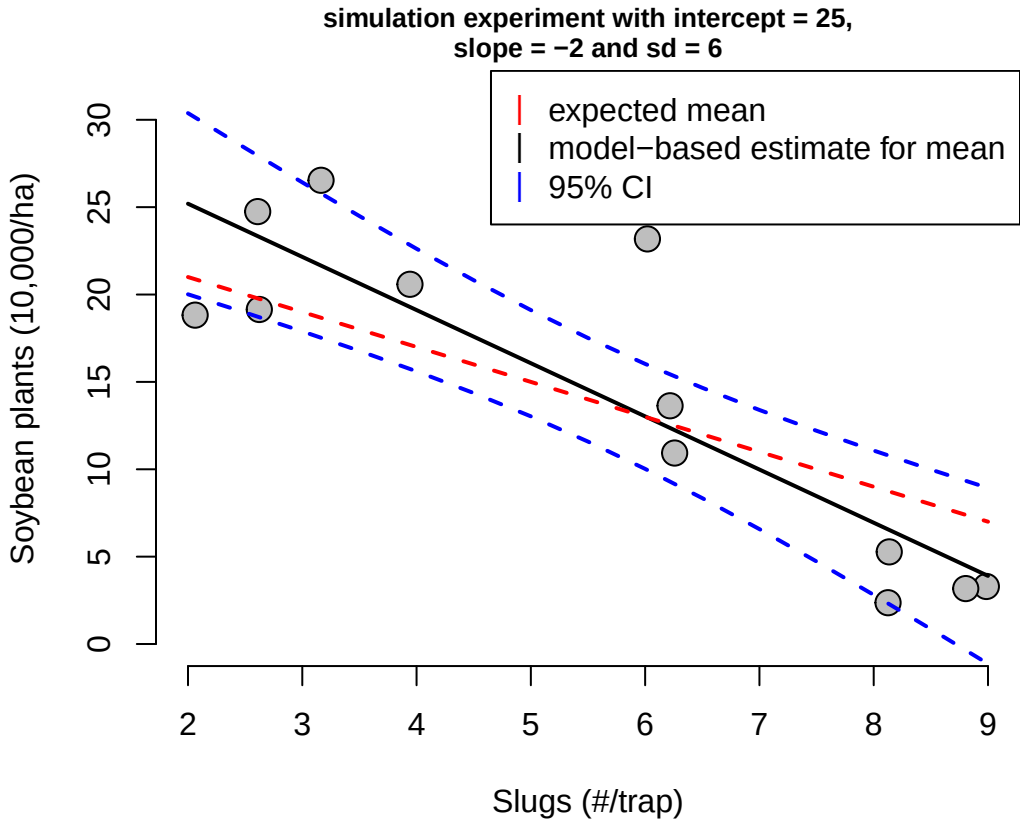


```
print(paste0("intercept = ", signif(
  my_simulated_data$estimate_intercept, digits=3)))
print(paste0("slope = ", signif(my_simulated_data$estimate_slope, digits=3)))
```

QUESTION 8: Describe your visual interpretation of how the data and association changed after altering the input intercept and slope (comparing with the plot generated with intercept=25 and slope =-2). 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 idea that slugs destroy soybean plants? COPY AND PASTE YOUR PLOT at the end of your response to this question

Now **re-enter the original intercept and slope values (intercept = 25; slope = -2)**. Now we will vary the precision of our measurements. Decrease the precision (increase the standard deviation) of our experiment. Change the sd from 3 to 6 and rerun the simulation 1 time (n_reps=1).

```
# simulate data with a new sd
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                  min_slugs_observed=2, max_slugs_observed=9,
                                  intercept=25, slope=-2, sd=6,
                                  n_reps = 1)
```




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

QUESTION 9: Describe your visual interpretation of how the data and association changed after altering the precision (compared to the simulation with 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 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=6). 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

```
# compare parameter estimates for relatively low sd
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 = 100)
# versus for relatively high sd
my_simulated_data <- simulate_slugs_and_soybeans(n=12,
                                    min_slugs_observed=2, max_slugs_observed=9,
                                    intercept=25, slope=-2, sd=6,
                                    n_reps = 100)
```

QUESTION 10: Compare the intercept estimates as well as the slope estimates for when sd=3 versus when sd=6. 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. Make sure to use a consistent standard deviation across the sets of simulations for both small and large sample sizes (e.g., sd = 3).

```
# compare parameter estimates for relatively low sample size
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 = 100)
# versus relatively high sample size
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 = 100)
```

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

Take Home: If the system were different (e.g., if the association between slugs and soybeans were positive), the linear regression model will provide different estimates for the intercept and slope (in that case, a slope

estimate greater than 0). Reducing precision (i.e., increasing the sd) or reducing sample size (reducing n) can make it more difficult to estimate the underlying relationship. We more frequently get estimates that are far from the true values governing the system.

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

Imagine that now that you have a good idea of your system and what kind of data would support or fail to support your hypothesis, you designed and conducted a real field experiment measuring slugs per trap and soybean.

Load the slug and soybean data

Congratulations! After a long summer of data collection the numbers are in! You've neatly organized all of your slug and soybean data into a .csv spreadsheet. The .csv is available in Canvas, save the file into your working directory folder. Once you've got this on your personal computer, read the spreadsheet into our R session and first get a summary of those data.

```
# read in the spreadsheet (after saving it in your working directory folder)
my_real_data <- read.csv("./new_slug_and_soybean_data.csv")

# view the data
(my_real_data)

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

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

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

Apply the linear regression analysis to the new data

Finally, repeat the analysis that we conducted on the simulated data to determine the association in our "real" data.

```
# 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))

# save important outputs
# intercept term
(estimate_intercept <- summary(fit2)$coefficients[1,1])
# effect of slug increase
(estimate_slope <- summary(fit2)$coefficients[2,1])

# now plot the fit (with confidence intervals)

# first we need to create some new data
```

```r
# we will make predictions for the mean and confidence across the same range
# of slugs that we "observed" in our simulation
min_slugs_observed <- floor(min(my_real_data$slugs_per_trap)) # round down
max_slugs_observed <- ceiling(max(my_real_data$slugs_per_trap)) # round up

# 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
{
  par(mfrow=c(1,1), mar = c(5, 5, 2, 5)) # Set the margin on the sides to 5

  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)

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

}
```

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 ecological meanings of these estimates (what do these estimates tell us about slug and soybean densities)?

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

Take Home: A linear regression model can tell us about the association between an independent and dependent variable. This provides us with a way to quantitatively test an ecological hypothesis.

————- END DAY 2 ————-