**Linear Model – 25 pts**

PLEASE READ THIS BEFORE CONTINUING

This assignment will help you practice writing, executing, and interpreting a linear model in R. It will also involve some more practice with GitHub. You may collaborate with a partner to enhance your learning experience. Please ensure the following:

* **Collaboration**: If you work with a partner, include both names on the final submission by editing the YAML header.
* **Submission**: Only one person should submit the assignment to Canvas *in a Word document or .pdf file generated through R markdown*. Additionally, you should provide a link to your GitHub, where the assignment should be viewable by rendering it as a GitHub-flavored markdown file.
* **Setup**: It is also assumed you already have a GitHub repository for this class.
* **Time**: This should take you no longer than the class period to complete.

1. 4 pts. Read in the data called “PlantEmergence.csv” using a relative file path and load the following libraries. *tidyverse, lme4, emmeans, multcomp,* and *multcompView*. Turn the Treatment , DaysAfterPlanting  and Rep into factors using the function *as.factor*

STAND$Treatment <- as.factor(STAND$Treatment) # example shown here.

1. 5 pts. Fit a linear model to predict Emergence using Treatment and DaysAfterPlanting along with the interaction. Provide the summary of the linear model and ANOVA results.
2. 5 pts. Based on the results of the linear model in question 2, do you need to fit the interaction term? Provide a simplified linear model without the interaction term but still testing both main effects. Provide the summary and ANOVA results. Then, interpret the intercept and the coefficient for Treatment 2.
3. 5 pts. Calculate the least square means for Treatment using the *emmeans* package and perform a Tukey separation with the compact letter display using the *cld* function. Interpret the results.
4. 4 pts. The provided function lets you dynamically add a linear model plus one factor from that model and plots a bar chart with letters denoting treatment differences. Use this model to generate the plot shown below. Explain the significance of the letters.

plot\_cldbars\_onefactor <- **function**(lm\_model, factor) {

data <- lm\_model$model

variables <- colnames(lm\_model$model)

dependent\_var <- variables[1]

independent\_var <- variables[2:length(variables)]

lsmeans <- emmeans(lm\_model, as.formula(paste("~", factor))) *# estimate lsmeans*

Results\_lsmeans <- cld(lsmeans, alpha = 0.05, reversed = TRUE, details = TRUE, Letters = letters) *# contrast with Tukey adjustment by default.*

*# Extracting the letters for the bars*

sig.diff.letters <- data.frame(Results\_lsmeans$emmeans[,1],

str\_trim(Results\_lsmeans$emmeans[,7]))

colnames(sig.diff.letters) <- c(factor, "Letters")

*# for plotting with letters from significance test*

ave\_stand2 <- lm\_model$model %>%

group\_by(!!sym(factor)) %>%

dplyr::summarize(

ave.emerge = mean(.data[[dependent\_var]], na.rm = TRUE),

se = sd(.data[[dependent\_var]]) / sqrt(n())

) %>%

left\_join(sig.diff.letters, by = factor) %>%

mutate(letter\_position = ave.emerge + 10 \* se)

plot <- ggplot(data, aes(x = !! sym(factor), y = !! sym(dependent\_var))) +

stat\_summary(fun = mean, geom = "bar") +

stat\_summary(fun.data = mean\_se, geom = "errorbar", width = 0.5) +

ylab("Number of emerged plants") +

geom\_jitter(width = 0.02, alpha = 0.5) +

geom\_text(data = ave\_stand2, aes(label = Letters, y = letter\_position), size = 5) +

xlab(as.character(factor)) +

theme\_classic()

**return**(plot)

}

A graph of a number of patients

AI-generated content may be incorrect.

1. 2 pts. Generate the gfm .md file along with a .html, .docx, or .pdf. Commit, and push the .md file to github and turn in the .html, .docx, or .pdf to Canvas. Provide me a link here to your github.