## Explanation for AAD Analysis R Script

This document explains each section of the R script we wrote for the Antibiotics Associated Diarrhea (AAD) dataset. I’ll break it down step by step, telling you **what the code does**, **why it’s done**, and **where you can make changes**.

### 1. Data Reading

# Load required packages  
library(dplyr)  
library(ggplot2)  
library(psych)  
library(car)  
library(ggpubr)  
library(rstatix)  
  
# Load data  
load("AAD.RData")

**Explanation:** - library(...) loads all the necessary R packages. If you don’t have them, install with install.packages("name"). - load("AAD.RData") loads the dataset into the R environment. After this, your dataframe is named AAD.

**Changes:** - If your file has a different name (e.g., AAD\_v2.RData), change it inside load().

### 2. Descriptive Statistics

# Summary statistics  
summary(AAD)  
  
# Descriptive statistics for continuous variables  
describe(AAD[, c("D1.Shannon.diversity", "D6.Shannon.diversity",  
 "D1.Chao1.diversity", "D6.Chao1.diversity",  
 "D1.D6.Jaccard.distance")])  
  
# Frequency table for categorical variables  
table(AAD$Outcome)  
  
# Correlation coefficients  
cor(AAD$D1.Shannon.diversity, AAD$D6.Shannon.diversity)  
cor(AAD$D1.Chao1.diversity, AAD$D6.Chao1.diversity)

**Explanation:** - summary() gives quick stats like mean, median, min, max. - describe() from **psych** gives detailed stats including quartiles. - table() gives frequency counts of categorical variables (e.g., Outcome). - cor() calculates correlation between two numeric variables.

**Changes:** - If you want to summarize other variables, add them to the describe() vector.

### 3. Graphics

# Bar chart of Outcome  
ggplot(AAD, aes(x = Outcome)) +  
 geom\_bar(fill = "steelblue") +  
 theme\_minimal()  
  
# Bar chart of mean Outcome by Antibiotic class  
ggplot(AAD, aes(x = Antibiotic.class, fill = Outcome)) +  
 geom\_bar(position = "dodge") +  
 theme\_minimal()  
  
# Histograms of Chao diversity  
ggplot(AAD, aes(x = D1.Chao1.diversity)) +  
 geom\_histogram(bins = 30, fill = "darkgreen", alpha = 0.7)  
  
ggplot(AAD, aes(x = D6.Chao1.diversity)) +  
 geom\_histogram(bins = 30, fill = "darkred", alpha = 0.7)  
  
# Scatterplot D1 vs D6 Chao  
ggplot(AAD, aes(x = D1.Chao1.diversity, y = D6.Chao1.diversity, color = Antibiotic.class)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE)  
  
# Boxplot of Jaccard distance by Antibiotic class  
ggplot(AAD, aes(x = Antibiotic.class, y = D1.D6.Jaccard.distance, fill = Antibiotic.class)) +  
 geom\_boxplot() +  
 theme\_minimal()

**Explanation:** - The bar charts show categorical distributions. - Histograms show distribution shapes of Chao diversities. - Scatterplot shows relation between Day1 and Day6 Chao, with regression lines. - Boxplot compares Jaccard distance between antibiotic classes.

**Changes:** - You can adjust bins in histograms or change fill colors.

### 4. Outlier Detection

# Boxplot method for outliers  
boxplot(AAD$D1.Chao1.diversity, main="D1 Chao Outlier Check")  
boxplot(AAD$D6.Chao1.diversity, main="D6 Chao Outlier Check")  
  
# Identify actual values  
boxplot.stats(AAD$D1.Chao1.diversity)$out  
boxplot.stats(AAD$D6.Chao1.diversity)$out

**Explanation:** - Boxplots visualize outliers. - boxplot.stats()$out shows exact outlier values.

**Changes:** - Replace variable names to check other columns.

### 5. Normality & Homoscedasticity

# Normality tests  
shapiro.test(AAD$D1.Chao1.diversity)  
shapiro.test(AAD$D6.Chao1.diversity)  
  
# QQ plots  
ggqqplot(AAD$D1.Chao1.diversity)  
ggqqplot(AAD$D6.Chao1.diversity)  
  
# Homoscedasticity tests  
leveneTest(D1.Chao1.diversity ~ Antibiotic.class, data = AAD)  
fligner.test(D1.Chao1.diversity ~ Antibiotic.class, data = AAD)

**Explanation:** - shapiro.test() checks normality. - ggqqplot() (from ggpubr) gives visual QQ plots. - leveneTest() and fligner.test() test equal variances.

**Changes:** - Replace variables if you want to test others.

### 6. Confidence Intervals

# Confidence intervals for Chao diversity by Antibiotic class  
AAD %>%  
 group\_by(Antibiotic.class) %>%  
 summarise(  
 mean\_chao\_D1 = mean(D1.Chao1.diversity, na.rm = TRUE),  
 ci90 = t.test(D1.Chao1.diversity, conf.level=0.90)$conf.int,  
 ci95 = t.test(D1.Chao1.diversity, conf.level=0.95)$conf.int,  
 ci99 = t.test(D1.Chao1.diversity, conf.level=0.99)$conf.int  
 )

**Explanation:** - This gives mean and confidence intervals at 90%, 95%, and 99%. - Wider intervals appear at higher confidence levels.

**Changes:** - Replace D1.Chao1.diversity with Shannon or other variables.

### 7. Hypothesis Testing

# Paired t-test: Shannon Day1 vs Day6  
t.test(AAD$D1.Shannon.diversity, AAD$D6.Shannon.diversity, paired = TRUE)  
  
# Independent t-test: Chao between PBL and FQN  
t.test(D1.Chao1.diversity ~ Antibiotic.class, data = AAD %>% filter(Antibiotic.class %in% c("PBL","FQN")))  
  
# ANOVA: Chao across all antibiotic classes  
aov\_model <- aov(D1.Chao1.diversity ~ Antibiotic.class, data = AAD)  
summary(aov\_model)  
TukeyHSD(aov\_model)

**Explanation:** - Paired t-test compares D1 vs D6 within the same patients. - Independent t-test compares two antibiotic groups. - ANOVA checks if differences exist across all antibiotic groups, with Tukey post-hoc testing.

**Changes:** - Swap Shannon for Chao where needed.

### 8. Linear Models

# Simple linear regression  
lm\_model <- lm(D6.Chao1.diversity ~ D1.Chao1.diversity + Antibiotic.class, data = AAD)  
summary(lm\_model)  
confint(lm\_model)  
  
# Repeated measures (mixed effects model)  
library(lme4)  
lmer\_model <- lmer(D6.Chao1.diversity ~ D1.Chao1.diversity + Antibiotic.class + (1|Patient.ID), data = AAD)  
summary(lmer\_model)

**Explanation:** - lm() fits a linear regression. - confint() gives confidence intervals for regression slopes. - lmer() (from lme4) fits a mixed model that accounts for repeated measures per patient.

**Changes:** - Replace outcome variable with Shannon if needed.

✅ This explanation file works as a companion to the main R script. You can run the code in the script, then read this file side by side to fully understand what’s happening.