|  |  |  |
| --- | --- | --- |
| **Block** | **R Code** | **Process Code** |
| 1 | names(RPMA2Growth) | inspect data |
| 2 | early <- subset(RPMA2Growth, StockYear < 2006) | filter data, relational statement, create new dataframe |
| 2 | mid <- subset(RPMA2Growth, StockYear < 2014 & StockYear > 2003) | filter data, relational statement, joined by logical (&), create new dataframe |
| 2 | RPMA2GrowthSub <- transform(RPMA2Growth, Age = as.integer(Age)) | converts to dataframe, adds new column, mutates existing column |
| 2 | Early <- subset(RPMA2GrowthSub, StockYear < 2004) | filter data, relational statement, create new dataframe |
| 2 | Mid <- subset(RPMA2GrowthSub, StockYear < 2018 & StockYear > 2005) | filter data, relational statement, joined by logical (&), create new dataframe |
| 3 | EarlyWeightAge <- ddply(Early, ~Age, summarise, meanWE=mean(Weight, na.rm = T)) | data summaries (mean) of a variable for groups of another variable |
| 3 | EarlyLengthAge <- ddply(Early, ~Age, summarise, meanLE = mean(ForkLength, na.rm = T)) | data summaries (mean) of a variable for groups of another variable |
| 3 | MidLengthAge <- ddply(Mid, ~Age, summarise, meanLM = mean(ForkLength, na.rm = T)) | data summaries (mean) of a variable for groups of another variable |
| 4 | WeightChange <- rep(NA, 9) | create new variable |
| 5 | library(plyr) | load in package, need for ddply function |
| 5 | WeightAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanW=mean(Weight, na.rm = T)) | data summaries (mean) of a variable for groups of another variable |
| 5 | LengthAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanL=mean(ForkLength, na.rm = T)) | data summaries (mean) of a variable for groups of another variable |
| 6 | #Tanner's code/help | comment on where code is coming from |
| 6 | WeightChange <- rep(NA, 9) | redundant, already included |
| 6 | library(plyr) | redundant, already included |
| 6 | WeightAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanW=mean(Weight, na.rm = T)) | redundant, already included |
| 6 | LengthAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanL=mean(ForkLength, na.rm = T)) | redundant, already included |
| 6 | plot(WeightAge$meanW ~ WeightAge$Age) | scatterplot of data summary and data variable |
| 6 | plot(LengthAge$mean ~ LengthAge$Age) | scatterplot of data summary and data variable, non-existent variable, incorrect name for data summary created |
| 6 | WeightChange | inspect variable created |
| 7 | Weight1 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 1], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight1 | inspect variable created |
| 7 | Length1 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 1], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight2 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 2], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length2 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 2], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight3 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 3], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length3 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 3], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight4 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 4], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length4 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 4], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight5 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 5], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length5 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 5], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight6 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 6], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length6 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 6], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight7 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 7], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length7 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 7], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight8 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 8], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length8 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 8], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Weight9 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 9], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | Length9 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 9], na.rm = TRUE) | data summary (mean), filter variable, using brackets, relational statement (==), create new variable |
| 7 | x <- data.frame("Age" = 1:9, "Growth" = Weight1, Weight2, Weight3, Weight4, Weight5,  Weight6, Weight7, Weight8, Weight9) | create new dataframe, create new variables, insert values from data summaries calculated |