Predicting Crab Age

Chloe Barker, Drew Nunnally, Tracy Dower

2025-04-21

# Load Required Libraries

Load libraries required for data manipulation, machine learning, and plotting

library(ggplot2)  
library(gtsummary) # pretty tables

library(overviewR)

# library(pROC)  
library(tidyverse)

# library(UBL)  
library(GGally)

intVersionNumber <- 10   
# Set colors  
hexBlue <- "#354CA1"  
hexRed <- "#CC0035"   
hexGold <- "#DD923B"  
sex\_colors <- c("Female" = hexRed, "Male" = hexBlue, "Indeterminate" = hexGold)  
age\_group\_colors <- c("Young" = "#1f78b4", "Adult" = "#e31a1c", "Old" = "#33a02c")

# EDA

## Load Test and combinedData

train <- read.csv("train-1.csv")  
test <- read.csv("competition-1.csv")

## CompareTest and Train, Compare

Do test and train have the same columns and the same levels?

# Columns only in train  
setdiff(names(train), names(test))

## [1] "Age"

# Columns only in test  
setdiff(names(test), names(train))

## character(0)

# Columns in both  
intersect(names(train), names(test))

## [1] "id" "Sex" "Length" "Diameter"   
## [5] "Height" "Weight" "Shucked.Weight" "Viscera.Weight"  
## [9] "Shell.Weight"

# Check for levels in one dataset that aren't in the other  
categoricals <- names(train)[sapply(train, is.factor)]  
for (var in categoricals) {  
 cat("Levels in test only for", var, ":\n")  
 print(setdiff(unique(test[[var]]), unique(train[[var]])))  
}

Returns NO ROWS so we’re good!

## Combine Test and Train before cleaning

train <- train %>% mutate(DataSet = "train")  
test <- test %>% mutate(DataSet = "test")  
combinedData <- bind\_rows(train, test)  
  
# Set factor levels for Sex  
combinedData$Sex <- factor(combinedData$Sex, levels = c("F", "M", "I"), labels = c("Female", "Male", "Indeterminate"))  
names(combinedData)[names(combinedData) == "id"] <- "ID"  
  
# Order the columns for consistent nice plots and tables.  
orderColumns <- c("Age", "Sex", "Diameter", "Height", "Length", "Weight","Shell.Weight", "Shucked.Weight", "Viscera.Weight")  
  
# Reorder the columns for nice reporting  
combinedData <- combinedData[ , c("ID","DataSet", orderColumns)]  
# names(combinedData)

## Handle Missing and Nonsense Values

# names(combinedData)  
combinedData |>   
 tbl\_summary(   
 by = Sex  
 ,include = c(Length, Diameter, Height, Weight, Shell.Weight, Shucked.Weight, Viscera.Weight)  
 ,type = all\_continuous() ~ "continuous2"  
 ,statistic = all\_continuous() ~ c(  
 "{N\_nonmiss}",  
 "{median} ({p25}, {p75})",  
 "{min}, {max}"  
 ),  
 missing = "no"  
 ) |>  
 add\_p(pvalue\_fun = label\_style\_pvalue(digits = 2))

| **Characteristic** | **Female** N = 7,686 | **Male** N = 9,026 | **Indeterminate** N = 8,288 | **p-value***1* |
| --- | --- | --- | --- | --- |
| Length |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 1.50 (1.38, 1.59) | 1.48 (1.33, 1.56) | 1.06 (0.88, 1.24) |  |
| Min, Max | 0.61, 2.04 | 0.20, 1.95 | 0.19, 1.68 |  |
| Diameter |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 1.18 (1.06, 1.25) | 1.15 (1.04, 1.23) | 0.81 (0.66, 0.95) |  |
| Min, Max | 0.45, 1.58 | 0.28, 1.58 | 0.00, 1.31 |  |
| Height |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 0.40 (0.36, 0.44) | 0.39 (0.35, 0.43) | 0.26 (0.21, 0.31) |  |
| Min, Max | 0.15, 0.60 | 0.06, 0.59 | 0.00, 0.51 |  |
| Weight |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 30 (23, 37) | 28 (22, 35) | 10 (6, 16) |  |
| Min, Max | 2, 75 | 0, 75 | 0, 49 |  |
| Shell.Weight |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 8.5 (6.8, 10.3) | 8.1 (6.2, 9.8) | 3.0 (1.7, 4.7) |  |
| Min, Max | 0.6, 23.1 | 0.1, 29.1 | 0.1, 18.9 |  |
| Shucked.Weight |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 12.6 (9.6, 15.8) | 12.1 (8.8, 15.3) | 4.5 (2.4, 7.1) |  |
| Min, Max | 0.6, 38.2 | 0.2, 42.2 | 0.1, 18.9 |  |
| Viscera.Weight |  |  |  | <0.001 |
| N Non-missing | 7,686 | 9,026 | 8,288 |  |
| Median (Q1, Q3) | 6.44 (4.89, 8.05) | 6.11 (4.54, 7.56) | 2.13 (1.16, 3.42) |  |
| Min, Max | 0.38, 20.13 | 0.13, 16.81 | 0.01, 9.54 |  |
| *1*Kruskal-Wallis rank sum test | | | | |

# Number of rows in our data where values are NA  
sum(sapply(names(combinedData), function(col) {  
 if (col == "Age") {  
 sum(is.na(combinedData[[col]]) & combinedData$DataSet != "Train") # We expect NAs for Age in our Train set.  
 } else {  
 sum(is.na(combinedData[[col]]))  
 }  
}))

## [1] 10000

# number of rows in our data where values are nonsense (0 is not a valid value)  
sum(sapply(combinedData[ , names(combinedData) != "id"], function(x) sum(x == 0, na.rm = TRUE)))

## [1] 23

# nrow(combinedData)  
  
summaryMissing <- function(data, ignore\_cols = c("ID", "Age")) {  
 data %>%  
 summarise(across(where(is.numeric), ~sum(is.na(.) | . == 0))) %>%  
 select(-all\_of(ignore\_cols)) %>%  
 pivot\_longer(everything(), names\_to = "Variable", values\_to = "Missing\_Count") %>%  
 filter(Missing\_Count > 0) %>%  
 mutate(Missing\_Percent = round(Missing\_Count / nrow(data) \* 100, 2)) %>%  
 arrange(desc(Missing\_Count))  
}  
missingStart <- summaryMissing(combinedData)  
missingStart

## # A tibble: 2 × 3  
## Variable Missing\_Count Missing\_Percent  
## <chr> <int> <dbl>  
## 1 Height 19 0.08  
## 2 Diameter 3 0.01

We have data for 25000 crabs; of those, only 19 crabs are missing Height, and only 3 crabs are missing Diameter. We impute the missing values for Diameter based on a simple linear regression model using Sex, Height, Weight, and Length. We impute the missing values for Height based on a simple linear regression model using Sex, Diameter, Weight, and Length.

fitDiameter <- lm(Diameter ~ Weight + Length + Height + Sex, data = combinedData[combinedData$Height > 0 & combinedData$Diameter > 0, ])  
 missing <- which(combinedData$Diameter == 0 & combinedData$Height > 0 & combinedData$Length > 0 & combinedData$Weight > 0) # Predict Diameter for those rows using the fitted model  
 combinedData$Diameter[missing] <- predict(fitDiameter, newdata = combinedData[missing, ]) # Replace Diameter with predicted values  
# Predict Height for those rows using the fitted model  
fitHeight <- lm(Height ~ Weight + Length + Diameter + Sex, data = combinedData %>% filter(Height > 0, Diameter > 0, Length > 0, Weight > 0))  
# Predict Diameter for those   
missing <- which(combinedData$Height == 0 & combinedData$Diameter > 0 & combinedData$Length > 0 & combinedData$Weight > 0)   
# Replace Height with predicted values   
 combinedData$Height[missing] <- predict(fitHeight, newdata = combinedData[missing, ])   
missingEnd <- summaryMissing(combinedData)  
missingEnd

## # A tibble: 0 × 3  
## # ℹ 3 variables: Variable <chr>, Missing\_Count <int>, Missing\_Percent <dbl>

# Missing data has been resolved.

## Plot and Summarize Data

# A different summary table   
tbl\_summary(  
 combinedData[ , names(combinedData) != "ID"],  
 by = Sex,  
 label = NULL,  
 statistic = list(all\_continuous() ~ "{median} ({p25}, {p75})", all\_categorical() ~ "{n} ({p}%)"),  
 digits = NULL,  
 type = NULL,  
 value = NULL,  
 missing = c("ifany", "no", "always"),  
 missing\_text = "Unknown",  
 missing\_stat = "{N\_miss}",  
 sort = all\_categorical(FALSE) ~ "alphanumeric",  
 percent = c("column", "row", "cell"),  
 include = orderColumns #everything()  
)

| **Characteristic** | **Female** N = 7,686*1* | **Male** N = 9,026*1* | **Indeterminate** N = 8,288*1* |
| --- | --- | --- | --- |
| Age | 11.0 (9.0, 13.0) | 10.0 (9.0, 12.0) | 7.0 (6.0, 8.0) |
| Unknown | 3,110 | 3,569 | 3,321 |
| Diameter | 1.18 (1.06, 1.25) | 1.15 (1.04, 1.23) | 0.81 (0.66, 0.95) |
| Height | 0.40 (0.36, 0.44) | 0.39 (0.35, 0.43) | 0.26 (0.21, 0.31) |
| Length | 1.50 (1.38, 1.59) | 1.48 (1.33, 1.56) | 1.06 (0.88, 1.24) |
| Weight | 30 (23, 37) | 28 (22, 35) | 10 (6, 16) |
| Shell.Weight | 8.5 (6.8, 10.3) | 8.1 (6.2, 9.8) | 3.0 (1.7, 4.7) |
| Shucked.Weight | 12.6 (9.6, 15.8) | 12.1 (8.8, 15.3) | 4.5 (2.4, 7.1) |
| Viscera.Weight | 6.44 (4.89, 8.05) | 6.11 (4.54, 7.56) | 2.13 (1.16, 3.42) |
| *1*Median (Q1, Q3) | | | |

### Correlations

trainClean <- combinedData %>% filter(DataSet == "train")  
# names(trainClean)  
plot(trainClean[,3:11],col=hexBlue) # This is NOT superuseful  
# Do not include Age or ID in the list of predictors.  
all\_predictors <- setdiff(names(trainClean), c("Age", "ID","Dataset"))  
# Keep only numeric predictors  
numeric\_predictors <- all\_predictors[sapply(trainClean[ , all\_predictors], is.numeric)]  
# Compute correlations with Age  
myCorrelations <- cor(trainClean[ , numeric\_predictors], trainClean$Age, use = "complete.obs")  
myCorrelationsALL <- cor(trainClean[ , numeric\_predictors])  
myCorrelationsALL  
## Graphical Correlation Matrix:  
symnum(myCorrelationsALL)   
# Length and Diameter = B → correlation > 0.95 → extremely strong  
# Weight and Height = \* → very strong correlation  
# Shell Weight is strongly or very strongly correlated with most others  
# Shucked Weight and Viscera Weight = + → strong correlation

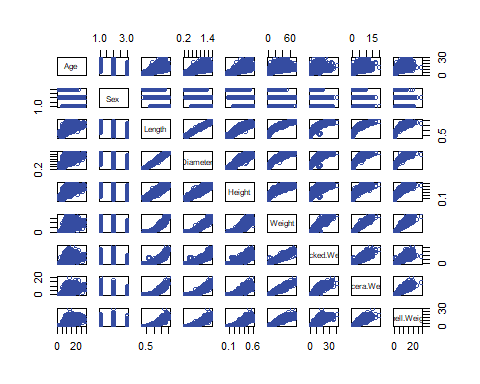
## Transformed and Derived Variables

Using raw Weight, the coefficient for slope was highly significant (<0.001) but difficult to interpret (Beta = 0.00 with a 95% confidence level of [0,0]) therefore, we explored transformations of weight (\* 100, or natural log). The natural log improved the correlation with age, from 0.614 0.631.

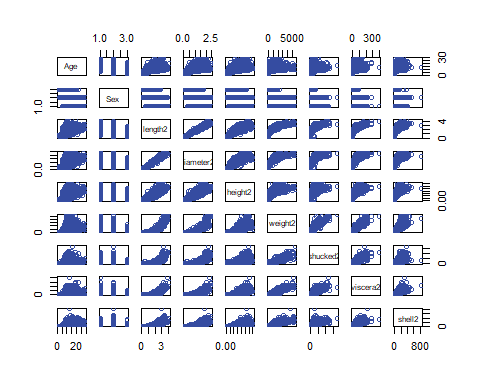
# Log-transform the variables  
# Derived metrics == length \* diameter = volume  
combinedData = combinedData %>%  
 mutate(  
 ### Log-transformed terms  
 log\_Age = log(Age),  
   
 log\_Length = log(Length),  
 log\_Diameter = log(Diameter),  
 log\_Height = log(Height),  
 log\_Weight = log(Weight),  
 log\_ShuckedWeight = log(Shucked.Weight),  
 log\_VisceraWeight = log(Viscera.Weight),  
 log\_ShellWeight = log(Shell.Weight),  
 ### Squared terms  
 length2 = Length^2,  
 diameter2 = Diameter^2,  
 height2 = Height^2,  
 weight2 = Weight^2,  
 shucked2 = Shucked.Weight^2,  
 viscera2 = Viscera.Weight^2,  
 shell2 = Shell.Weight^2,  
 ### Cubic terms  
 length3 = Length^3,  
 diameter3 = Diameter^3,  
 height3 = Height^3,  
 weight3 = Weight^3,  
 shucked3 = Shucked.Weight^3,  
 viscera3 = Viscera.Weight^3,  
 shell3 = Shell.Weight^3,  
 ### Interaction terms  
 dl.int = Diameter \* Length,  
 dh.int = Diameter \* Height,  
 dshuck.int = Diameter \* Shucked.Weight,  
 dvisc.int = Diameter \* Viscera.Weight,  
 dshell.int = Diameter \* Shell.Weight,  
 hl.int = Length \* Height,  
 hw.int = Height \* Weight,  
 shellww = Shell.Weight \* Weight,  
 shuckvisc = Shucked.Weight \* Viscera.Weight,  
 shellshuck = Shell.Weight \* Shucked.Weight,  
 shellvisc = Shell.Weight \* Viscera.Weight,  
 # DERIVED variables  
 BodySize = Length \* Diameter,  
 ShellDensity = Shell.Weight/Weight,  
 logShellDensity = log\_ShellWeight/log\_Weight,  
 BodySizeLog = log\_Length \* log\_Diameter,  
 # Times 100 versions of base variables  
 Length\_c = Length \* 100,  
 Diameter\_c = Diameter \* 100,  
 Height\_c = Height \* 100,  
 Weight\_c = Weight \* 100,  
 ShuckedWeight\_c = Shucked.Weight \* 100,  
 VisceraWeight\_c = Viscera.Weight \* 100,  
 ShellWeight\_c = Shell.Weight \* 100,  
 BodySize\_c = BodySize \* 100  
 )

## Correlations

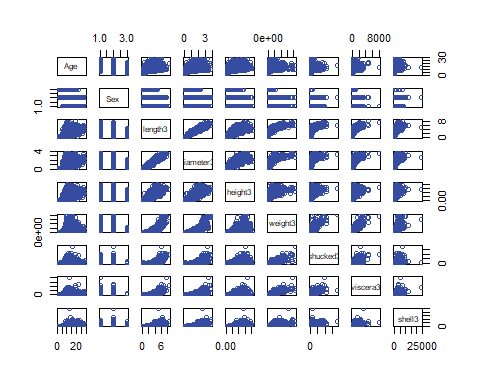
# names(trainClean)  
trainClean <- combinedData %>% filter(DataSet == "train")  
pRawAgeRawElse <- plot(trainClean[,c("Age", "Sex", "Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")],col=hexBlue)



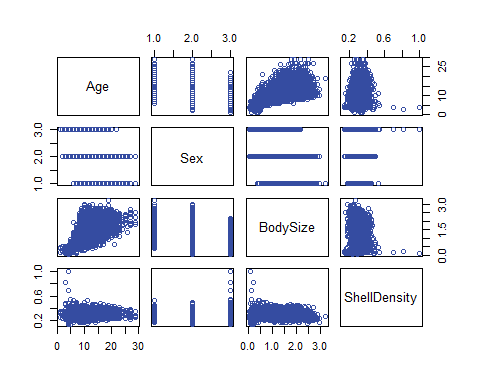
pRawAgeSquared <- plot(trainClean[,c("Age", "Sex", "length2", "diameter2", "height2", "weight2", "shucked2", "viscera2", "shell2")],col=hexBlue)



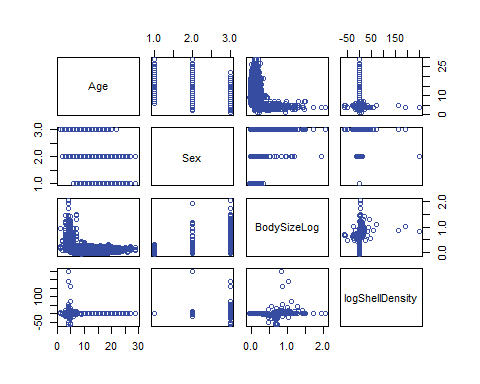
pRawAgeCubed <- plot(trainClean[,c("Age", "Sex", "length3", "diameter3", "height3", "weight3", "shucked3", "viscera3", "shell3")],col=hexBlue)



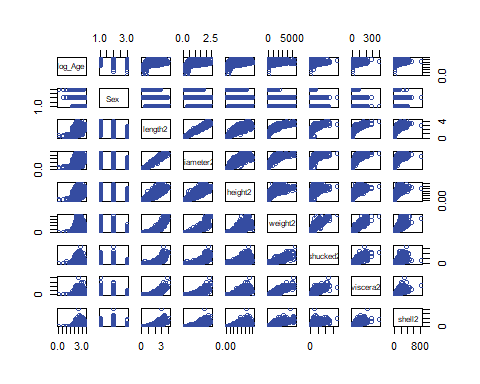
pRawAgeDerived <- plot(trainClean[,c("Age", "Sex", "BodySize", "ShellDensity")],col=hexBlue)



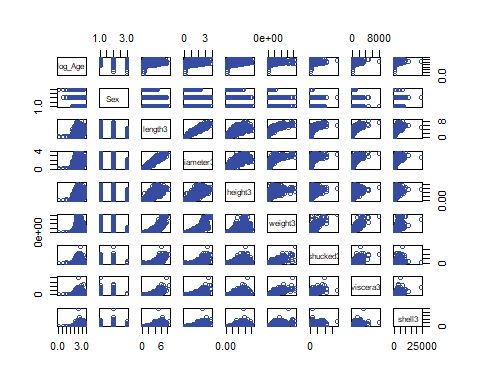
pRawAgeDerivedLog <- plot(trainClean[,c("Age", "Sex", "BodySizeLog", "logShellDensity")],col=hexBlue)



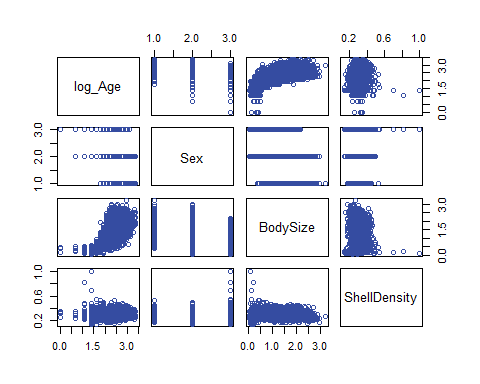
pLogAgeSquared <- plot(trainClean[,c("log\_Age", "Sex", "length2", "diameter2", "height2", "weight2", "shucked2", "viscera2", "shell2")],col=hexBlue)



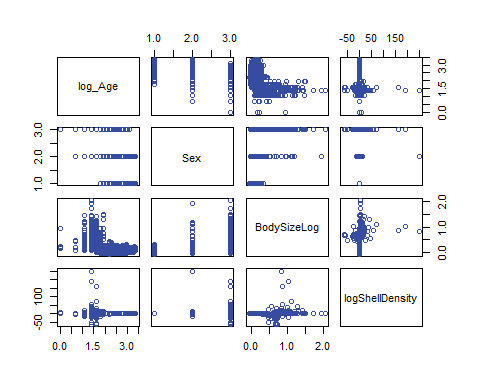
pLogAgeCube <- plot(trainClean[,c("log\_Age", "Sex", "length3", "diameter3", "height3", "weight3", "shucked3", "viscera3", "shell3")],col=hexBlue)



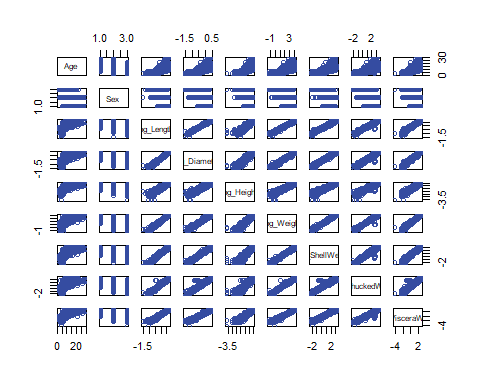
pLogAgeDerived <- plot(trainClean[,c("log\_Age", "Sex", "BodySize", "ShellDensity")],col=hexBlue)



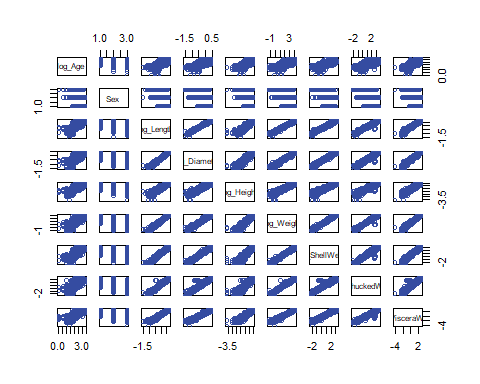
pLogAgeDerivedLog <- plot(trainClean[,c("log\_Age", "Sex", "BodySizeLog", "logShellDensity")],col=hexBlue)



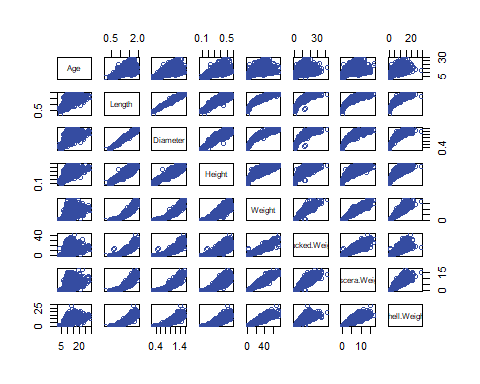
pRawAgeLogElse <- plot(trainClean[,c( "Age", "Sex", "log\_Length", "log\_Diameter", "log\_Height", "log\_Weight", "log\_ShellWeight", "log\_ShuckedWeight", "log\_VisceraWeight")],col=hexBlue)



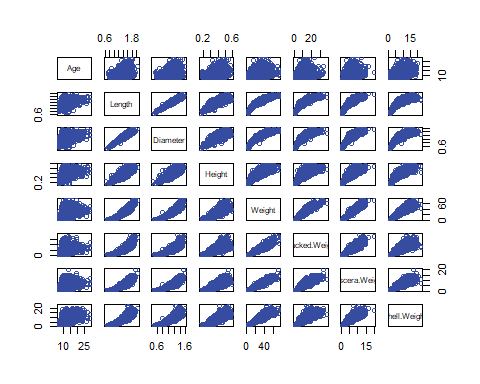
pLogAgeLogElse <- plot(trainClean[,c("log\_Age", "Sex", "log\_Length", "log\_Diameter", "log\_Height", "log\_Weight", "log\_ShellWeight", "log\_ShuckedWeight", "log\_VisceraWeight")],col=hexBlue)



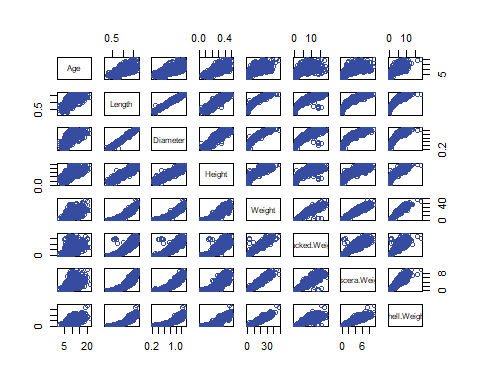
train1Sex<- combinedData %>% filter(Sex == "Male")  
pRawAgeRawMale<- plot(train1Sex[,c("Age", "Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")],col=hexBlue)



train1Sex<- combinedData %>% filter(Sex == "Female")  
pRawAgeRawFemale<- plot(train1Sex[,c("Age", "Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")],col=hexBlue)



train1Sex<- combinedData %>% filter(Sex == "Indeterminate")  
pRawAgeRawInterdeterminate <- plot(train1Sex[,c("Age", "Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")],col=hexBlue)



## SLR – Transformations

MAE(SLR: Age ~ Raw Values): 1.442213

trainClean <- combinedData %>% filter(DataSet == "train")  
  
fit\_SLR\_untransformed <- glm(Age ~ Sex + Length + Diameter + Height + Weight + Shell.Weight + Shucked.Weight + Viscera.Weight, data = trainClean)  
  
tableCoefficients <- tbl\_regression(fit\_SLR\_untransformed, exponentiate = FALSE)  
tableCoefficients

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| Sex |  |  |  |
| Female | — | — |  |
| Male | -0.13 | -0.22, -0.05 | 0.001 |
| Indeterminate | -1.1 | -1.2, -0.98 | <0.001 |
| Length | 0.73 | -0.20, 1.6 | 0.12 |
| Diameter | 1.0 | -0.16, 2.1 | 0.091 |
| Height | 9.8 | 8.6, 11 | <0.001 |
| Weight | 0.22 | 0.19, 0.24 | <0.001 |
| Shell.Weight | 0.59 | 0.55, 0.64 | <0.001 |
| Shucked.Weight | -0.63 | -0.66, -0.60 | <0.001 |
| Viscera.Weight | -0.34 | -0.39, -0.29 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

cat("MAE(SLR: Age ~ Raw Values):", mean(abs(residuals(fit\_SLR\_untransformed))),"\n")

## MAE(SLR: Age ~ Raw Values): 1.442213

# Derive Shell Density and Body Size

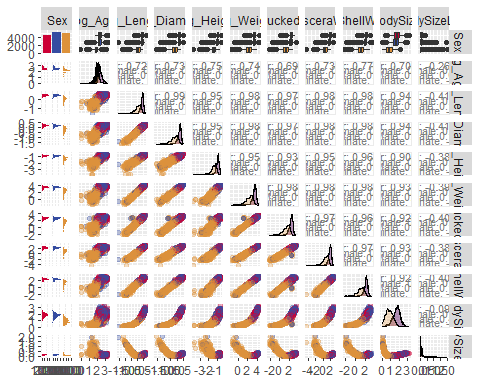
MAE(SLR Sex ~ Shell Density and Body Size): 0.1501884

combinedData$ShellDensity <- combinedData$log\_ShellWeight / combinedData$log\_Weight  
combinedData$BodySizeLog <- combinedData$log\_Length \* combinedData$log\_Diameter  
combinedData$BodySize <- combinedData$Length \* combinedData$Diameter  
  
trainClean <- combinedData %>% filter(DataSet == "train")  
fitDensityBodySize <- lm(log\_Age ~ Sex + ShellDensity + BodySize + log\_ShellWeight,data=trainClean)  
summary(fitDensityBodySize) # Multiple R-squared: 0.6214, Adjusted R-squared: 0.6213   
cat("MAE(SLR Sex ~ Shell Density and Body Size):", mean(abs(residuals(fitDensityBodySize))),"\n")  
# MAE = 0.1501884

## Plot by Sex

To demonstrate the necessity of modeling by interactions with Sex.

createPairsPlotBySex <- function(myVariables) {  
 plot <- ggpairs(  
 trainClean[ , myVariables],  
 mapping = aes(color = Sex),  
 upper = list(continuous = wrap("cor", size = 3)),  
 lower = list(continuous = wrap("points", alpha = 0.3, position = position\_jitter(width = 0.09, height = 0.09))),  
 diag = list(continuous = wrap("densityDiag", alpha = 0.3))  
 ) +  
 theme(legend.position = "bottom") +  
 scale\_color\_manual(values = sex\_colors) +  
 scale\_fill\_manual(values = sex\_colors)  
 return(plot)  
}  
## Plot Correlations  
InterestingVariables <- c("Sex","log\_Age", "log\_Length", "log\_Diameter", "log\_Height", "log\_Weight", "log\_ShuckedWeight", "log\_VisceraWeight", "log\_ShellWeight", "BodySize", "BodySizeLog")  
Plot\_Sex\_and\_DerivedVariables <- createPairsPlotBySex(InterestingVariables)  
suppressWarnings(print(Plot\_Sex\_and\_DerivedVariables))



## Create Interaction Variables

combinedData$Sex\_Female <- ifelse(combinedData$Sex == "Female", 1, 0)  
combinedData$Sex\_Male <- ifelse(combinedData$Sex == "Male", 1, 0)  
combinedData$Sex\_Indeterminate <- ifelse(combinedData$Sex == "Indeterminate", 1, 0)  
combinedData$Sex\_Female\_BodySize <- combinedData$Sex\_Female \* combinedData$BodySize  
combinedData$Sex\_Male\_BodySize <- combinedData$Sex\_Male \* combinedData$BodySize  
combinedData$Sex\_Indeterminate\_BodySize <- combinedData$Sex\_Indeterminate \* combinedData$BodySize  
combinedData$Sex\_Female\_ShellDensity <- combinedData$Sex\_Female \* combinedData$ShellDensity  
combinedData$Sex\_Male\_ShellDensity <- combinedData$Sex\_Male \* combinedData$ShellDensity  
combinedData$Sex\_Indeterminate\_ShellDensity <- combinedData$Sex\_Indeterminate \* combinedData$ShellDensity  
  
trainClean <- combinedData %>% filter(DataSet == "train")

## OLSRR Step Forward

library(olsrr)  
# names(combinedData)  
  
fullModel <- lm(log\_Age ~ Sex +  
ShellDensity + BodySize +  
log\_Length + log\_Diameter + log\_Height + log\_Weight + log\_ShuckedWeight + log\_VisceraWeight + log\_ShellWeight +  
Sex\_Female\_BodySize + Sex\_Male\_BodySize + Sex\_Indeterminate\_BodySize +  
Sex\_Female\_ShellDensity + Sex\_Male\_ShellDensity + Sex\_Indeterminate\_ShellDensity,  
data = trainClean  
)  
  
# summary(fullModel) # Multiple R-squared: 0.695, Adjusted R-squared: 0.6945  
# cat("MAE(everything):", mean(abs(residuals(fullModel))),"\n")  
  
stepForward\_AIC <- ols\_step\_forward\_aic(fullModel,include = NULL,exclude = NULL,progress = FALSE,details = FALSE)  
suppressWarnings(plot(stepForward\_AIC, print\_plot = TRUE, details = TRUE, digits = 3))  
writeLines(capture.output(print(stepForward\_AIC)),"ols\_step\_forward\_aic.txt")  
newModel = stepForward\_AIC$model  
# summary(newModel) # Multiple R-squared: 0.5592, Adjusted R-squared: 0.559 We Went BACKWARD :(  
cat("MAE(",deparse(formula(stepForward\_AIC$model)),"):", mean(abs(residuals(stepForward\_AIC$model))),"\n")

## OLSRR Step Backward

fullModel <- lm(log\_Age ~ Sex +  
 ShellDensity + BodySize +  
 log\_Length + log\_Diameter + log\_Height + log\_Weight +  
 log\_ShuckedWeight + log\_VisceraWeight + log\_ShellWeight ,  
 data = trainClean  
)  
stepBackward\_AIC <- ols\_step\_backward\_aic(fullModel, penter = 0.05, details = FALSE)  
suppressWarnings(plot(stepBackward\_AIC, print\_plot = TRUE, details = TRUE, digits = 3))  
  
writeLines(capture.output(print(stepBackward\_AIC)),"ols\_step\_backward\_aic.txt")  
newModel = stepBackward\_AIC$model  
# summary(newModel) # Multiple R-squared: 0.6789, Adjusted R-squared: 0.6787 :(  
cat("MAE(",deparse(formula(stepBackward\_AIC$model)),"):", mean(abs(residuals(stepBackward\_AIC$model))),"\n")

## OLSRR All Possible

# fullModel <- lm(log\_Age ~ Sex + ShellDensity + Sex\_Female\_BodySize + Sex\_Male\_BodySize + Sex\_Indeterminate\_BodySize + log\_Length + log\_Diameter + log\_Height + log\_Weight + log\_ShuckedWeight + log\_VisceraWeight + log\_ShellWeight,data = trainClean)  
  
# fullModel <- lm(log\_Age ~ Sex + BodySize + ShellDensity + Sex\_Female\_BodySize + Sex\_Male\_BodySize + Sex\_Indeterminate\_BodySize + length2 + diameter2 + height2 + weight2 + shucked2 + viscera2 + shell2 + shucked3 + viscera3 + weight3 + shellshuck,data = trainClean)  
# fullModel <- lm(log\_Age ~ Sex + BodySize + ShellDensity + Sex\_Female\_BodySize + Sex\_Male\_BodySize + height2 + weight2 + shucked2 + viscera2 + shell2 + shucked3 + viscera3 + weight3 + shellshuck,data = trainClean)  
# summary(fullModel)  
# fullModel <- lm(log\_Age ~ Sex + BodySize + ShellDensity + Sex\_Female\_BodySize + Sex\_Male\_BodySize + height2 + weight2 + shucked2 + viscera2 + shell2 + shucked3 + shellshuck,data = trainClean)  
# summary(fullModel)  
# fullModel <- lm(log\_Age ~ Sex + BodySize + Sex\_Female\_BodySize + Sex\_Male\_BodySize + height2 + weight2 + shucked2 + viscera2 + shell2 + shucked3 + shellshuck,data = trainClean)  
# summary(fullModel) # Last run 7:20   
print(Sys.time())  
# length(fullModel$residuals) # Should match after NA omission  
  
all\_possible <- ols\_step\_all\_possible(fullModel, details = TRUE)   
writeLines(capture.output(print(all\_possible)),"ols\_step\_all\_possible.txt")  
  
Results\_Feature\_Selection <- all\_possible$result  
Results\_Feature\_Selection$mae <- NA\_real\_ # Add a null column for MAE  
# Just before the loop  
Results\_Feature\_Selection$predictors <- sapply(  
 Results\_Feature\_Selection$predictors,  
 function(x) paste(unlist(strsplit(as.character(x), "\\s+")), collapse = " ")  
)  
  
# loop through the models, calculating MAE as we go  
for (i in seq\_len(nrow(Results\_Feature\_Selection))) {  
 predictor\_string <- Results\_Feature\_Selection$predictors[i]  
 # Skip if predictor string is missing or empty  
 if (is.na(predictor\_string) || predictor\_string == "") next  
 predictors <- unlist(strsplit(trimws(predictor\_string), "\\s\*,\\s\*"))  
 # predictors <- unlist(strsplit(predictor\_string, "[,\\s]+"))  
 # print("predictors")  
 # print(predictors)  
 # gsub(pattern, replacement, x, ignore.case=FALSE, fixed=FALSE)  
 predictorsAndPluses <- gsub("\\s+", " + ", predictors)  
 # print(predictorsAndPluses)  
formula\_str <- paste("log\_Age ~", predictorsAndPluses)  
 # print("formula\_str")  
model <- lm(as.formula(formula\_str), data = trainClean)  
Results\_Feature\_Selection$mae[i] <- mean(abs(residuals(model)))  
}

# Objective A: Final Model

Develop a robust predictive model to accurately predict crab age based on various features. We aim to achieve the lowest possible Mean Absolute Error (MAE) on the test set, ensuring high precision in our predictions. Use the model identified above to predict age, then run it a LOT and fine-tune it. GOAL: Mean Absolute Error (MAE) = mean(abs(actual - predicted)) ## Simple Linear Regression We seek a model with the lowest MAE (mean of the absolute residuals).

fit\_SLR\_untransformed <- glm(Age ~ Sex + Length + Diameter + Height + Weight + Shell.Weight + Shucked.Weight + Viscera.Weight, data = trainClean)  
  
tableCoefficients <- tbl\_regression(fit\_SLR\_untransformed, exponentiate = FALSE)  
tableCoefficients

| **Characteristic** | **Beta** | **95% CI** | **p-value** |
| --- | --- | --- | --- |
| Sex |  |  |  |
| Female | — | — |  |
| Male | -0.13 | -0.22, -0.05 | 0.001 |
| Indeterminate | -1.1 | -1.2, -0.98 | <0.001 |
| Length | 0.73 | -0.20, 1.6 | 0.12 |
| Diameter | 1.0 | -0.16, 2.1 | 0.091 |
| Height | 9.8 | 8.6, 11 | <0.001 |
| Weight | 0.22 | 0.19, 0.24 | <0.001 |
| Shell.Weight | 0.59 | 0.55, 0.64 | <0.001 |
| Shucked.Weight | -0.63 | -0.66, -0.60 | <0.001 |
| Viscera.Weight | -0.34 | -0.39, -0.29 | <0.001 |
| Abbreviation: CI = Confidence Interval | | | |

### Plots

library(ggplot2)  
library(broom)

## Warning: package 'broom' was built under R version 4.4.3

plot\_assumptions <- function(model) {  
 library(ggplot2)  
 library(broom)  
 model\_df <- augment(model)  
 model\_df$Sex <- model$model$Sex # Pull Sex from model data  
 model\_terms <- attr(model$terms, "term.labels")  
 # model\_title <- paste("Predictors:", paste(model\_terms, collapse = " + "))  
 response <- deparse(formula(model)[[2]]) # left-hand side  
 model\_title <- paste("Model for", response, "~", paste(model\_terms, collapse = " + "))  
 p\_Residuals\_v\_Fitted <- ggplot(model\_df, aes(.fitted, .resid, color = Sex)) +  
 geom\_point() +  
 geom\_hline(yintercept = 0, linetype = "dashed") +  
 scale\_color\_manual(values = sex\_colors) +  
 labs(title = model\_title,  
 subtitle = "Residuals vs Fitted",  
 x = "Fitted Values", y = "Residuals", color = "Sex")  
  
 p\_QQ <- ggplot(model\_df, aes(sample = .std.resid, color = Sex)) +  
 stat\_qq() +  
 stat\_qq\_line(color = "black") +  
 scale\_color\_manual(values = sex\_colors) +  
 labs(title = model\_title,  
 subtitle = "Normal Q-Q Plot",  
 x = "Theoretical Quantiles", y = "Standardized Residuals", color = "Sex")  
  
 print(p\_Residuals\_v\_Fitted)  
 print(p\_QQ)  
}  
  
Model\_All\_Possible\_Best <- lm(log\_Age ~ Sex + BodySize + height2 + shell2 + shucked3 + shellshuck, data = trainClean)  
  
 summary(Model\_All\_Possible\_Best)

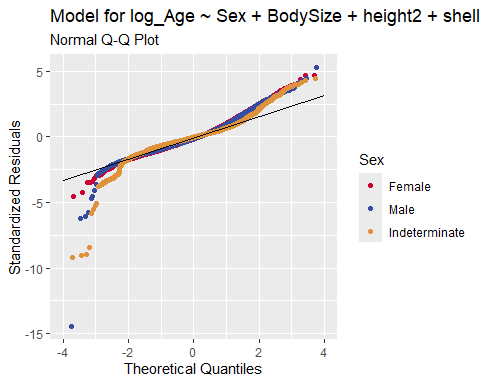
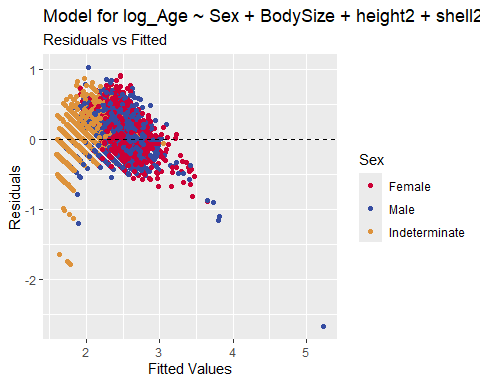
pred2 <- predict(Model\_All\_Possible\_Best, newdata = trainClean)  
 mae2 <- mean(abs(pred2 - trainClean$log\_Age))  
 cat("MAE (reduced model, log‐age):", mae2, "\n")

## MAE (reduced model, log‐age): 0.1426794

cat("Approx. MAE on original age scale:", exp(mae2), "\n")

## Approx. MAE on original age scale: 1.15336

plot\_assumptions(Model\_All\_Possible\_Best)



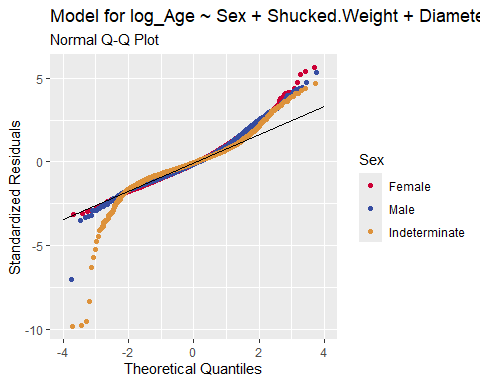
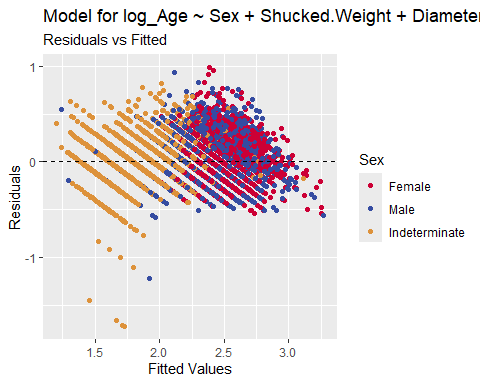
# 2) Reduced logged‐Age model  
Model\_Absolute\_Lowest\_MAE <- lm(log\_Age ~ Sex \* Shucked.Weight + Diameter + Height + Weight + Shell.Weight + Viscera.Weight +  
 # only the squared & cubic terms you kept  
 height2 + weight2 + shucked2 + viscera2 + shell2 + shell3 +  
 # key interactions & size metrics  
 dh.int + shellww + shellshuck + shellvisc + ShellDensity,  
 data = trainClean)  
 pred2 <- predict(Model\_Absolute\_Lowest\_MAE, newdata = trainClean)  
 mae2 <- mean(abs(pred2 - trainClean$log\_Age))  
 cat("MAE (reduced model, log‐age):", mae2, "\n")

## MAE (reduced model, log‐age): 0.1306896

cat("Approx. MAE on original age scale:", exp(mae2), "\n")

## Approx. MAE on original age scale: 1.139614

plot\_assumptions(Model\_Absolute\_Lowest\_MAE)



Final\_Model <- Model\_Absolute\_Lowest\_MAE

## Correlations

install.packages("psych")  
library(psych)  
  
# Select only numeric columns, excluding "ID"  
numeric\_vars <- names(trainClean)[sapply(trainClean, is.numeric)]  
numeric\_vars <- setdiff(numeric\_vars, "ID")  
result <- corr.test(trainClean[ , numeric\_vars], use = "pairwise")  
cor\_df <- as.data.frame(result$r)  
cor\_df$Variable <- rownames(cor\_df)  
cor\_df <- cor\_df[ , c("Variable", setdiff(names(cor\_df), "Variable"))]  
View(cor\_df) # tab-delimited table in Viewer pane

### Generate CSVs

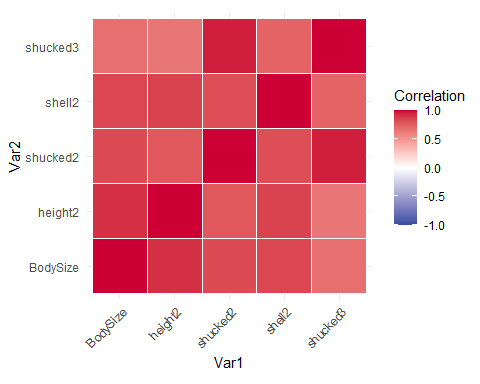
trainClean <- combinedData %>% filter(DataSet == "train")  
testClean <- combinedData %>% filter(DataSet == "test")  
# generate csv for the RSHINY app   
# intVersionNumber  
write.csv(trainClean, paste0("CrabsCleanTrain", intVersionNumber, ".csv"), row.names = FALSE)  
  
# generate csv of predictions from the competition data   
names(testClean)[names(testClean) == "ID"] <- "id"  
# names(testClean)   
  
predictions <- data.frame(Id = testClean$id,Age = predict(Final\_Model, newdata = testClean))  
colnames(predictions)[2] <- "Age"  
write.csv(predictions, "CrabsPredictions8.csv", row.names = FALSE)

# Objective B: Mining Data for Insights

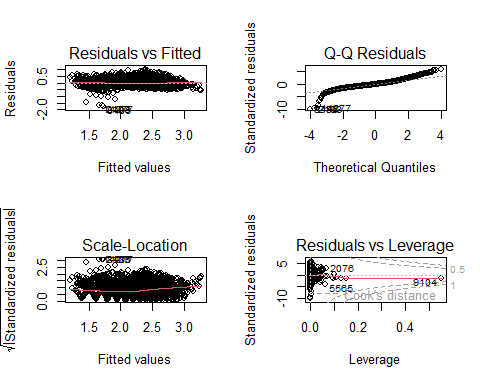
Identify the key determinants of crab age by analyzing relationships between various crab metrics and age provide valuable insights to help scientists understand growth rate, reproductive cycles, and lifespan which can help commercial markets as well as scientific studies on climate change, effects of ocean pollution, shifts in predatory trends, etc. Viscera.Weight relative to Body Size predicts female fertility? (Roe?)

library(reshape2) # For melt()

predictors <- combinedData[, c("Sex", "BodySize", "height2", "shucked2", "shell2", "shucked3")]  
  
# Keep only numeric columns  
numeric\_predictors <- predictors[sapply(predictors, is.numeric)]  
# Final\_Model  
# Compute correlation matrix  
CorrelationMatrix <- cor(numeric\_predictors, use = "complete.obs") # Creates a square matrix (table) of feature-to-feature correlations  
myCorrelations <- melt(CorrelationMatrix) # Melt() from the reshape2 package turns a matrix a long table so ggplot2 can plot it  
ggplot(myCorrelations, aes(Var1, Var2, fill = value)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient2(low = hexBlue, high = hexRed, mid = "white", midpoint = 0, limit = c(-1,1), name = "Correlation") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1)) +  
 coord\_fixed()



par(mfrow = c(2, 2)) # 2x2 layout for plots  
plot(Final\_Model)

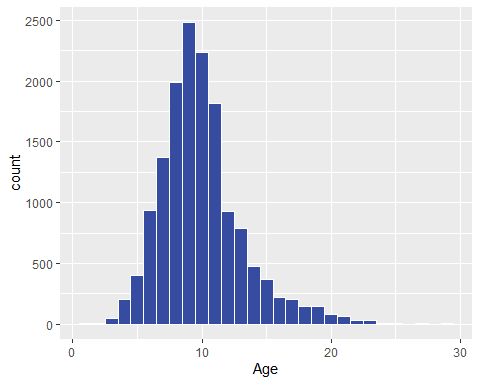
 ## Chloe plots

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.4.3

## corrplot 0.95 loaded

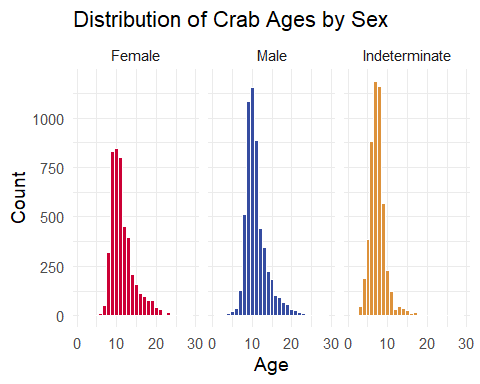
# Age Histograms  
trainClean %>%  
 ggplot(aes(x = Age)) +  
 geom\_histogram(binwidth = 1, fill = hexBlue, color = "white")



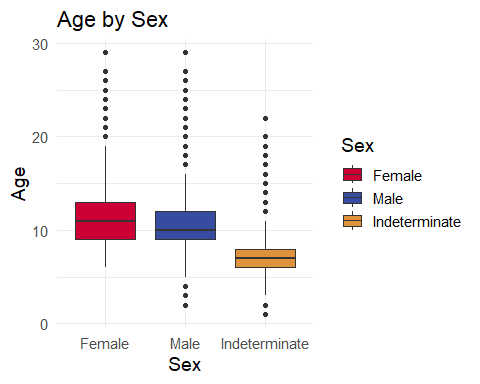
labs(title = "Distribution of Crab Ages", x = "Age", y = "Count") +  
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "none")

## NULL

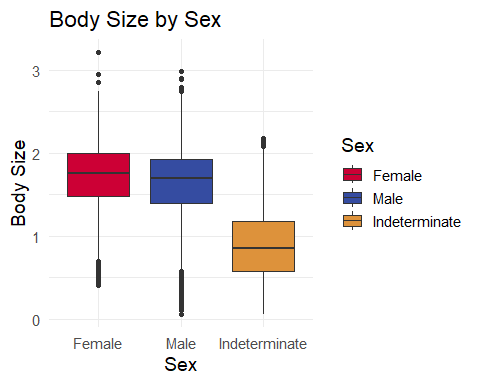
# Faceted histogram, one for each sex  
trainClean %>%  
 ggplot(aes(x = Age, fill = Sex)) +  
 geom\_histogram(binwidth = 1, color = "white") +  
 scale\_fill\_manual(values = sex\_colors) +  
 facet\_wrap(~ Sex) +  
 labs(title = "Distribution of Crab Ages by Sex", x = "Age", y = "Count") +  
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "none")



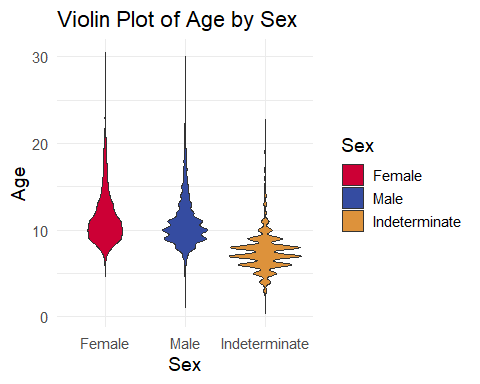
# Does one sex live longer?  
ggplot(trainClean, aes(x = Sex, y = Age, fill = Sex)) +   
 geom\_boxplot() +  
 scale\_fill\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Age by Sex", x = "Sex", y = "Age")



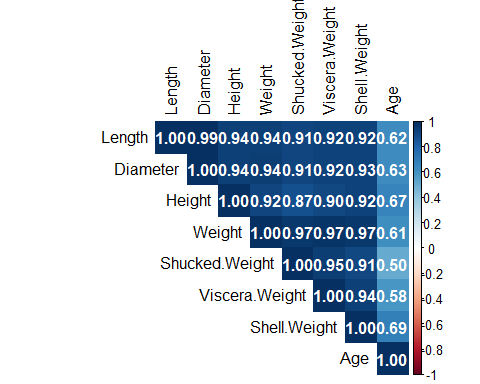
# Is one sex larger?  
ggplot(trainClean, aes(x = Sex, y = BodySize, fill = Sex)) +   
 geom\_boxplot() +  
 scale\_fill\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Body Size by Sex", x = "Sex", y = "Body Size")



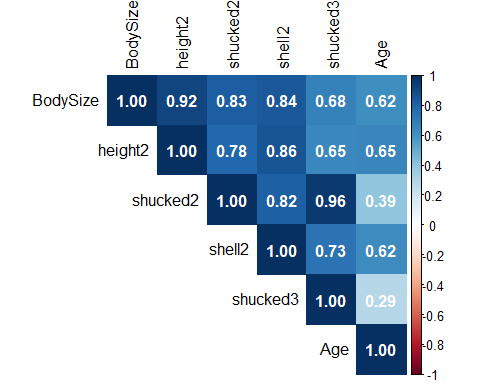
ggplot(trainClean, aes(x = Sex, y = Age, fill = Sex)) +  
 geom\_violin(trim = FALSE) +  
 scale\_fill\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Violin Plot of Age by Sex", x = "Sex", y = "Age")



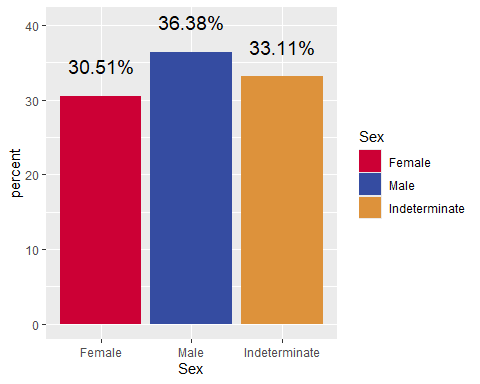
# Correlation Plot (Numeric Only) Raw Variables  
trainClean %>%  
 select(Length, Diameter, Height, Weight, Shucked.Weight, Viscera.Weight, Shell.Weight, Age) %>%  
 cor(use = "complete.obs") %>%  
 corrplot(method = "color", type = "upper", tl.col = "black", addCoef.col = "white")



# Correlation Plot (Numeric Only) Our Final Model == Sex + BodySize + height2 + shucked2 + shell2 + shucked3  
trainClean %>% select(BodySize, height2, shucked2, shell2, shucked3, Age) %>% cor() %>% corrplot(method = "color", type = "upper", tl.col = "black", addCoef.col = "white")

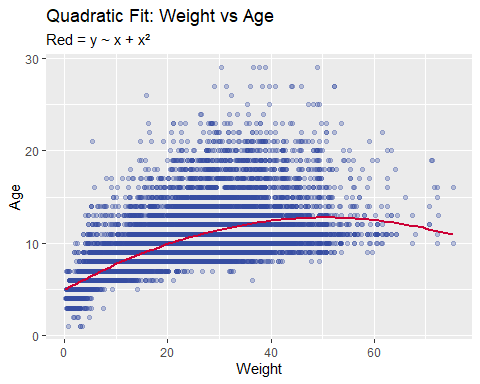


# Plot of percentage of crabs by Sex   
trainClean %>%   
 group\_by(Sex) %>%   
 dplyr::summarize(count=n()) %>%  
 mutate(percent = (count / sum(count))\*100) %>%  
 ggplot(aes(x=Sex, y = percent, fill=Sex)) +   
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = paste0(round(percent,2),"%")), nudge\_y = 4, size = 5) +   
 scale\_fill\_manual(values = sex\_colors)

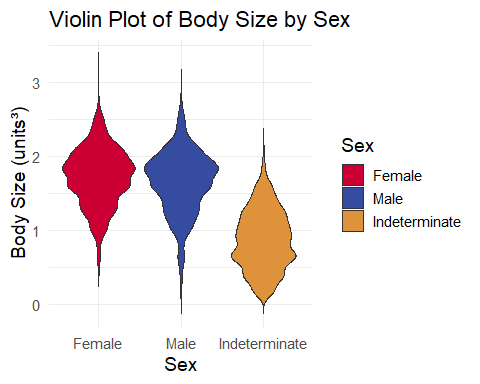


theme\_minimal(base\_size = 14) +   
 theme(legend.position = "none", text=element\_text(family = "Corbel")) +  
 ggtitle("Percentage of Crabs by Sex") +   
 xlab("Sex") +   
 ylab("Percent")

# Plot Age vs. Weight  
# Create a new column for weight squared   
ggplot(trainClean, aes(x = Weight, y = Age)) +  
 geom\_point(alpha = 0.3, color = hexBlue) +  
 geom\_smooth(method = "lm", formula = y ~ x + I(x^2), color = hexRed, se = FALSE) +  
 labs(  
 title = "Quadratic Fit: Weight vs Age",  
 subtitle = "Red = y ~ x + x²",  
 x = "Weight", y = "Age"  
 )

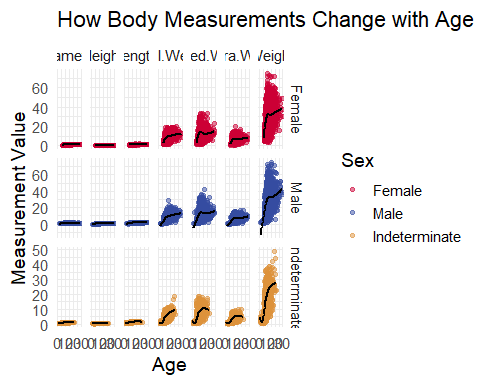


ggplot(trainClean, aes(x = Sex, y = BodySize, fill = Sex)) +  
 geom\_violin(trim = FALSE) +  
 scale\_fill\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Violin Plot of Body Size by Sex", x = "Sex", y = "Body Size (units³)")



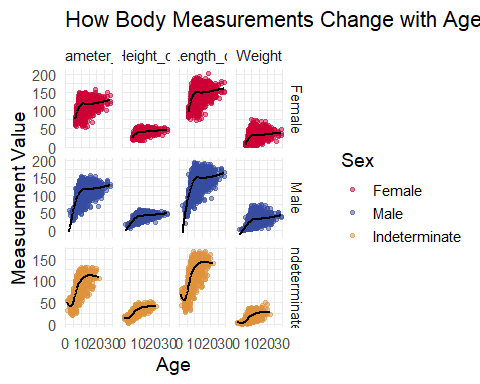
# How do body measurements change with age?  
# Scatterplot: Age vs Body measurements, one plot per Sex  
  
trainClean %>%  
 pivot\_longer(cols = c(Length, Diameter, Height, Weight, Shell.Weight, Shucked.Weight, Viscera.Weight),  
 names\_to = "Measurement", values\_to = "Value") %>%  
 ggplot(aes(x = Age, y = Value, color = Sex)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "black") +  
 facet\_grid(Sex ~ Measurement, scales = "free\_y") +  
 scale\_color\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "How Body Measurements Change with Age by Sex",  
 x = "Age",  
 y = "Measurement Value",  
 color = "Sex")

## `geom\_smooth()` using formula = 'y ~ x'

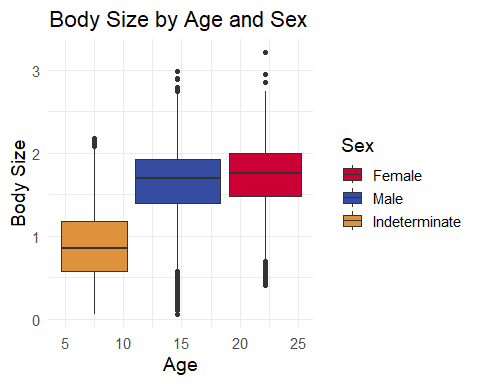


trainClean %>%  
 pivot\_longer(cols = c(Length\_c, Diameter\_c, Height\_c, Weight),  
 names\_to = "Measurement", values\_to = "Value") %>%  
 ggplot(aes(x = Age, y = Value, color = Sex)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "black") +  
 facet\_grid(Sex ~ Measurement, scales = "free\_y") +  
 scale\_color\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "How Body Measurements Change with Age by Sex",  
 x = "Age",  
 y = "Measurement Value",  
 color = "Sex")

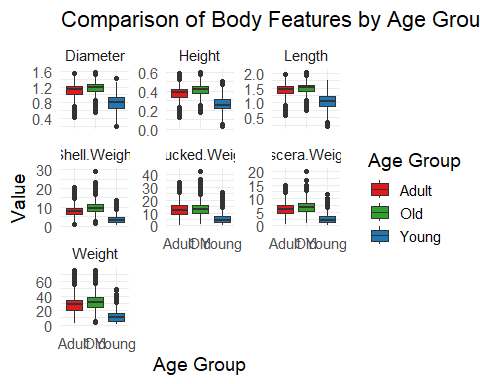
## `geom\_smooth()` using formula = 'y ~ x'



ggplot(trainClean, aes(x = Age, y = BodySize, fill = Sex)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = sex\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Body Size by Age and Sex", x = "Age", y = "Body Size", fill = "Sex")

 ### By Age GROUP

# Create Age bins  
trainClean <- trainClean %>%  
 mutate(AgeGroup = case\_when(  
 Age <= 8 ~ "Young",  
 Age <= 11 ~ "Adult",  
 TRUE ~ "Old"  
 ))  
  
# Define body features to compare  
features <- c("Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")  
  
# Pivot and plot  
trainClean %>%  
 pivot\_longer(cols = all\_of(features), names\_to = "Feature", values\_to = "Value") %>%  
 ggplot(aes(x = AgeGroup, y = Value, fill = AgeGroup)) +  
 geom\_boxplot() +  
 facet\_wrap(~ Feature, scales = "free\_y") +  
 scale\_fill\_manual(values = age\_group\_colors) +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Comparison of Body Features by Age Group", x = "Age Group", y = "Value", fill = "Age Group")

 ## Which variables change less as crabs age?

# Correlation between Age and other numeric variables  
age\_corr <- trainClean %>%  
 select(where(is.numeric)) %>%  
 summarise(across(everything(), ~ cor(Age, ., use = "complete.obs"))) %>%  
 pivot\_longer(cols = everything(), names\_to = "Variable", values\_to = "Correlation") %>%  
 arrange(abs(Correlation))  
  
# View variables weakly correlated with Age (e.g., abs(correlation) < 0.1)  
age\_corr %>% filter(abs(Correlation) < 0.15)

## # A tibble: 18 × 2  
## Variable Correlation  
## <chr> <dbl>  
## 1 ID -0.0000308  
## 2 logShellDensity 0.00652   
## 3 ShellDensity 0.126   
## 4 BodySizeLog -0.142

### Compare mean(variable) change year-over-year

From this we see that most body size metrics level off around 20 years. This is why our model fits less well as crab age increases beyond 20 years.

basic\_variables <- c("Length", "Diameter", "Height", "Weight", "Shucked.Weight", "Viscera.Weight", "Shell.Weight")  
  
# Scatterplot: Base variable values by Age and by Sex   
trainClean %>%  
 filter(!is.na(Age)) %>%  
 pivot\_longer(cols = all\_of(basic\_variables), names\_to = "Variable", values\_to = "Value") %>%  
 ggplot(aes(x = Age, y = Value, color = Sex)) +  
 geom\_point(alpha = 0.3, size = 1) +  
 geom\_smooth(method = "loess", se = FALSE, linewidth = 1, color = "black") +  
 scale\_color\_manual(values = sex\_colors) +  
 facet\_wrap(~ Variable, scales = "free\_y") +  
 theme\_minimal(base\_size = 14) +  
 labs(title = "Metrics by Age and Sex",  
 x = "Age", y = "Value", color = "Sex")

