

# DATA 622 Assignment 1

CUNY: Spring 2021

Philip Tanofsky

19 February 2021

## Introduction

The purpose of this project is to apply logistic regression approaches to the Palmer Penguin data set available at <https://allisonhorst.github.io/palmerpenguins/articles/intro.html>. The first approach performs binary logistic regression on the dataset in order to predict a species or not of the penguin subjects. The second approach utilizes all three species to perform a multinomial logistic regression in order to predict the species of penguin subjects. While the primary goal of the logistic regression is to predict the penguin species, statistical interpretation also presents the context for the prediction models.

Data – 622 Homework # 1 Due date Feb 19, 2021- 11:59 EST Let's use the Penguin dataset for our assignment. To learn more about the dataset, please visit: <https://allisonhorst.github.io/palmerpenguins/articles/intro.html> For this assignment, let us use 'species' as our outcome or the dependent variable. 1. Logistic Regression with a binary outcome. (40) a. The penguin dataset has 'species' column. Please check how many categories you have in the species column. Conduct whatever data manipulation you need to do to be able to build a logistic regression with binary outcome. Please explain your reasoning behind your decision as you manipulate the outcome/dependent variable (species). b. Please make sure you are evaluating the independent variables appropriately in deciding which ones should be in the model. c. Provide variable interpretations in your model. 2. For your model from #1, please provide: AUC, Accuracy, TPR, FPR, TNR, FNR (20) 3. Multinomial Logistic Regression. (40) a. Please fit it a multinomial logistic regression where your outcome variable is 'species'. b. Please be sure to evaluate the independent variables appropriately to fit your best parsimonious model. c. Please be sure to interpret your variables in the model. 4. Extra credit: what would be some of the fit statistics you would want to evaluate for your model in question #3? Feel free to share whatever you can provide. (10)

```
# Import required R libraries
library(palmerpenguins)
library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)
library(MASS)
library(pROC)
library(nnet) # Used for multinomial logistic regression
library(mlogit)
library(stargazer)
library(popbio)
# Set theme, based on the Penguin vignettes
theme_set(theme_minimal())
```

The palmer penguins dataset consists of 8 variables, 7 independent variables and 1 dependent variable (species).

## Variables

- species: species of the penguin observed (dependent variable)
- island: island of penguin's inhabitation
- bill\_length\_mm: penguin bill length in millimeters
- bill\_depth\_mm: penguin bill depth in millimeters
- flipper\_length\_mm: penguin flipper length in millimeters
- body\_mass\_g: penguin body mass in grams
- sex: penguin sex
- year: year of observation

## EDA

Initial data summary and exploratory data analysis.

```
ds <- penguins
```

```
head(ds)
```

```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex
##   <fct>   <fct>         <dbl>         <dbl>          <int>        <int> <fct>
## 1 Adelie  Torge~           39.1           18.7            181         3750 male
## 2 Adelie  Torge~           39.5           17.4            186         3800 fema~
## 3 Adelie  Torge~           40.3            18             195         3250 fema~
## 4 Adelie  Torge~            NA            NA              NA           NA <NA>
## 5 Adelie  Torge~           36.7           19.3            193         3450 fema~
## 6 Adelie  Torge~           39.3           20.6            190         3650 male
## # ... with 1 more variable: year <int>
```

```
summary(ds)
```

```
##           species           island bill_length_mm bill_depth_mm
## Adelie    :152  Biscoe       :168  Min.    :32.10  Min.    :13.10
## Chinstrap: 68  Dream        :124  1st Qu.:39.23  1st Qu.:15.60
## Gentoo    :124  Torgersen: 52   Median :44.45  Median :17.30
##                                     Mean    :43.92  Mean    :17.15
##                                     3rd Qu.:48.50  3rd Qu.:18.70
##                                     Max.    :59.60  Max.    :21.50
##                                     NA's    :2      NA's    :2
## flipper_length_mm body_mass_g      sex      year
## Min.    :172.0    Min.    :2700  female:165  Min.    :2007
## 1st Qu.:190.0    1st Qu.:3550  male  :168  1st Qu.:2007
## Median :197.0    Median :4050  NA's   : 11  Median :2008
## Mean    :200.9    Mean    :4202                Mean    :2008
## 3rd Qu.:213.0    3rd Qu.:4750                3rd Qu.:2009
## Max.    :231.0    Max.    :6300                Max.    :2009
## NA's     :2      NA's     :2
```

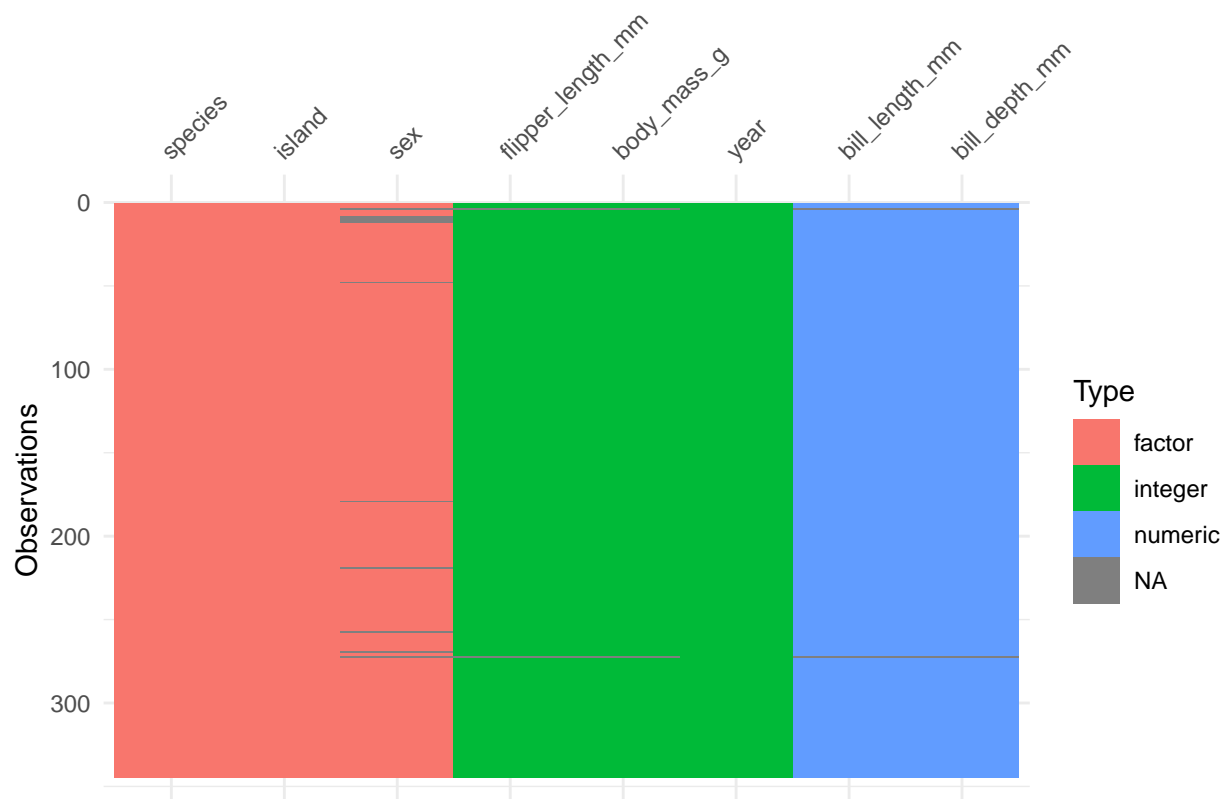
```
dim(ds)
```

```
## [1] 344 8
```

```
glimpse(ds)
```

```
## Rows: 344
## Columns: 8
## $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, A...
## $ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torge...
## $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34....
## $ bill_depth_mm <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18....
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, ...
## $ body_mass_g   <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 347...
## $ sex          <fct> male, female, female, NA, female, male, female, m...
## $ year         <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2...
```

```
visdat::vis_dat(ds)
```



Initial summary outputs show 344 instances of the 8 variables. The final graph indicates the missing values among the 8 variables. Variables with missing values include *sex*, *bill\_length\_mm*, *bill\_depth\_mm*, *flipper\_length\_mm* and *body\_mass\_g*.

```
# Penguins data has three factor variables
```

```
ds %>%  
  dplyr::select(where(is.factor)) %>%  
  glimpse()
```

```
## Rows: 344
```

```
## Columns: 3
```

```
## $ species <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Ade...
```

```
## $ island <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgersen, Torg...
```

```
## $ sex <fct> male, female, female, NA, female, male, female, male, NA, N...
```

```
# Count penguins for each species / island
```

```
ds %>%  
  count(species, island, .drop=F)
```

```
## # A tibble: 9 x 3
```

```
##   species   island     n
```

```
##   <fct>     <fct>   <int>
```

```
## 1 Adelie   Biscoe     44
```

```
## 2 Adelie   Dream      56
```

```
## 3 Adelie   Torgersen   52
```

```
## 4 Chinstrap Biscoe      0
```

```
## 5 Chinstrap Dream      68
```

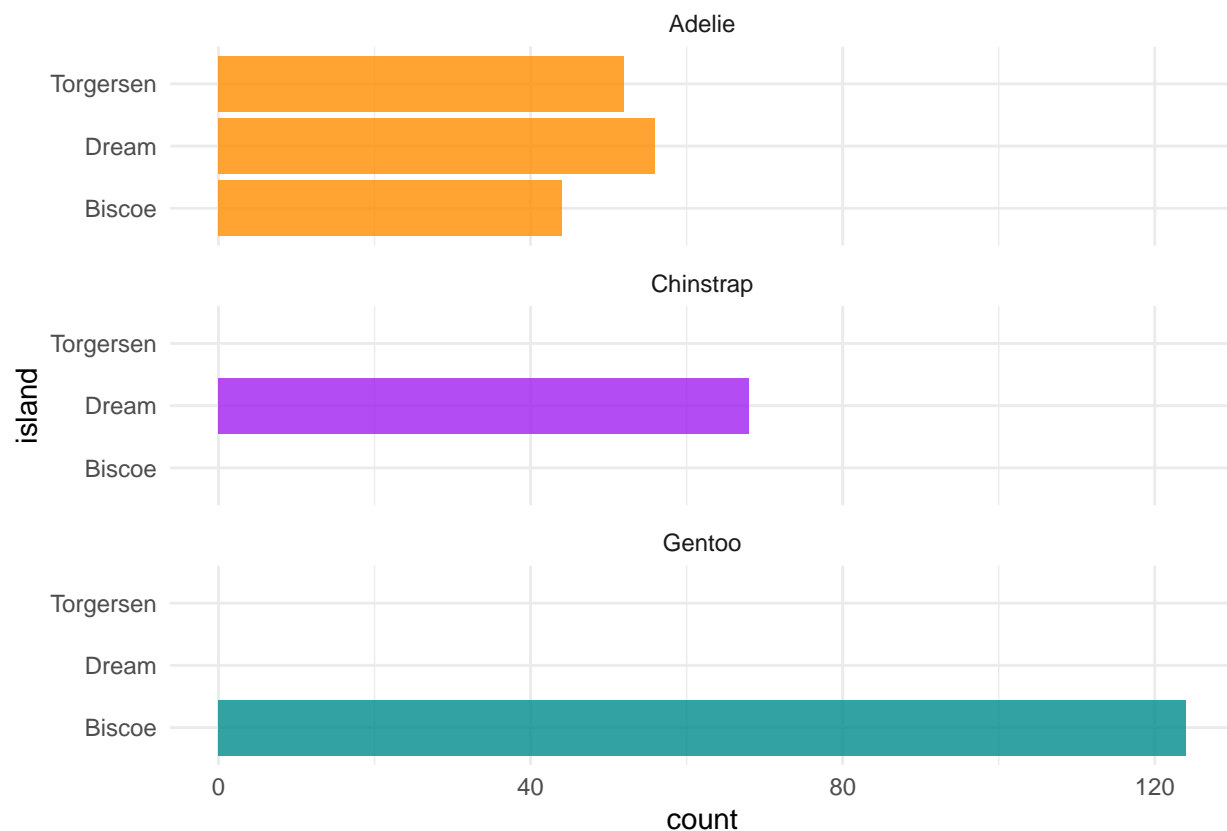
```
## 6 Chinstrap Torgersen    0
```

```
## 7 Gentoo   Biscoe    124
```

```
## 8 Gentoo   Dream      0
```

```
## 9 Gentoo   Torgersen    0
```

```
ggplot(ds, aes(x = island, fill = species)) +  
  geom_bar(alpha = 0.8) +  
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),  
                    guide = F) +  
  theme_minimal() +  
  facet_wrap(~species, ncol = 1) +  
  coord_flip()
```



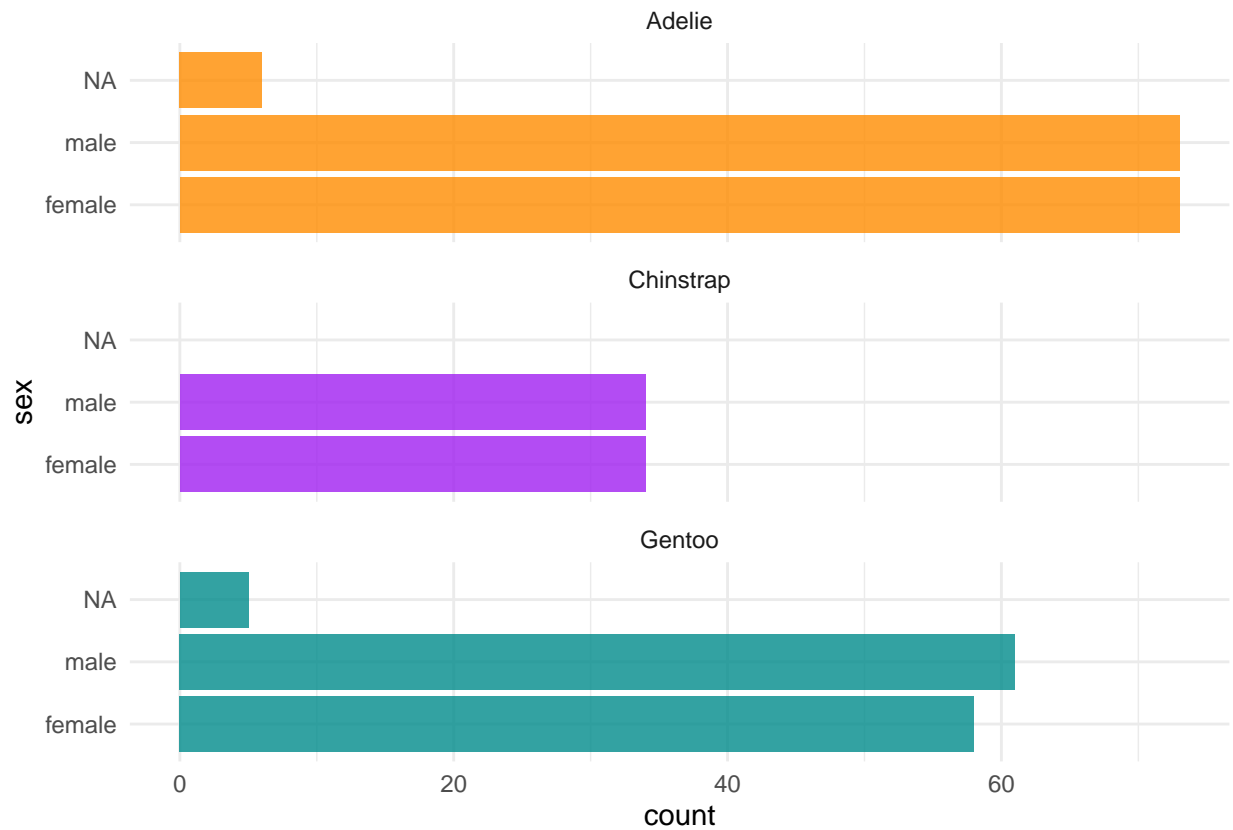
The above plot indicates the population of penguin species based on island. Interesting finding that Chinstrap are only observed on Dream island and Gentoo are only observed on Biscoe island, while the Adelie species are observed on all three islands in the study.

```
# Count penguins for each species / sex
ds %>%
  count(species, sex, .drop = F)
```

```
## # A tibble: 8 x 3
##   species sex      n
##   <fct>   <fct> <int>
## 1 Adelie female   73
## 2 Adelie male    73
## 3 Adelie <NA>     6
## 4 Chinstrap female 34
## 5 Chinstrap male   34
## 6 Gentoo female   58
## 7 Gentoo male    61
## 8 Gentoo <NA>     5
```

```
ggplot(ds, aes(x = sex, fill = species)) +
  geom_bar(alpha = 0.8) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"),
    guide = F) +
  theme_minimal() +
```

```
facet_wrap(~species, ncol = 1) +
coord_flip()
```

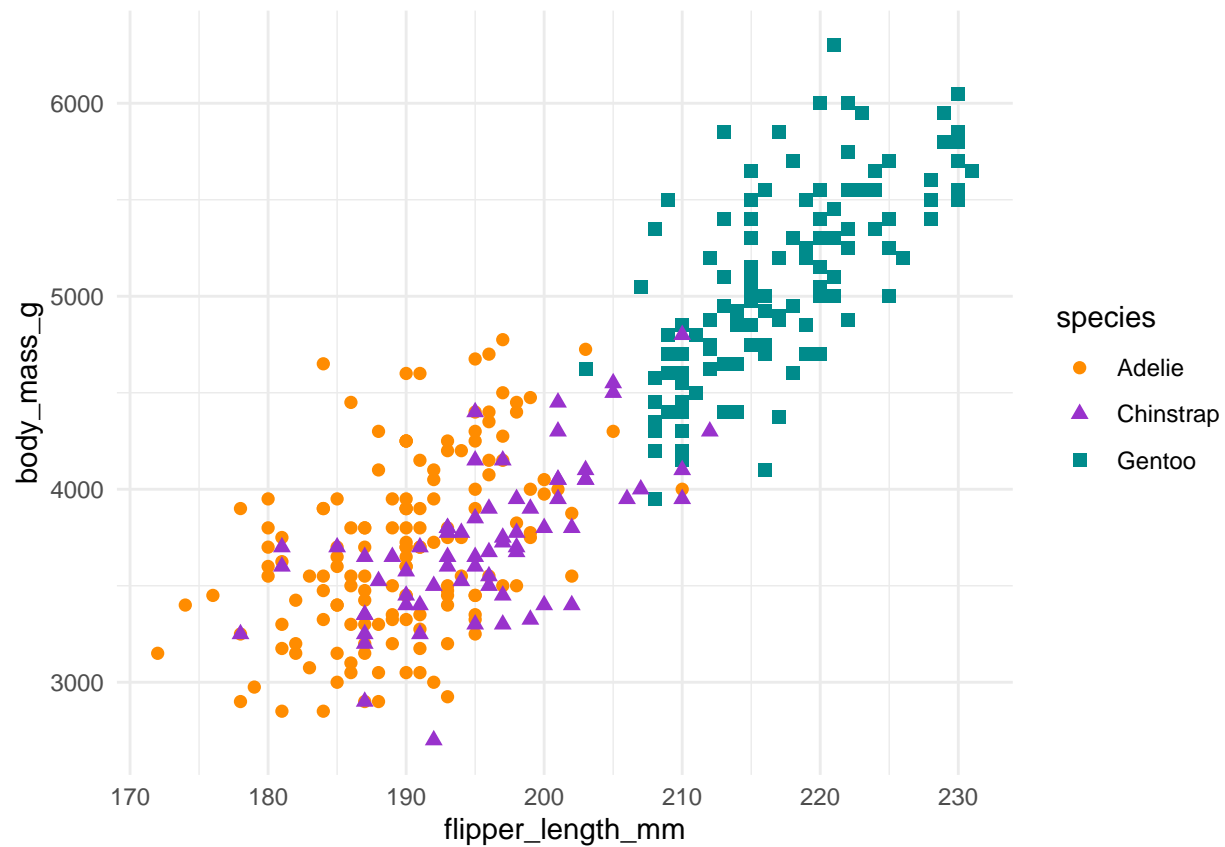


The above breakdown of penguin species by sex shows an expected ratio of near 50-50 for each species.

```
# Penguins data also has four continuous variables, making six unique scatterplots
ds %>%
  dplyr::select(body_mass_g, ends_with("_mm")) %>%
  glimpse()
```

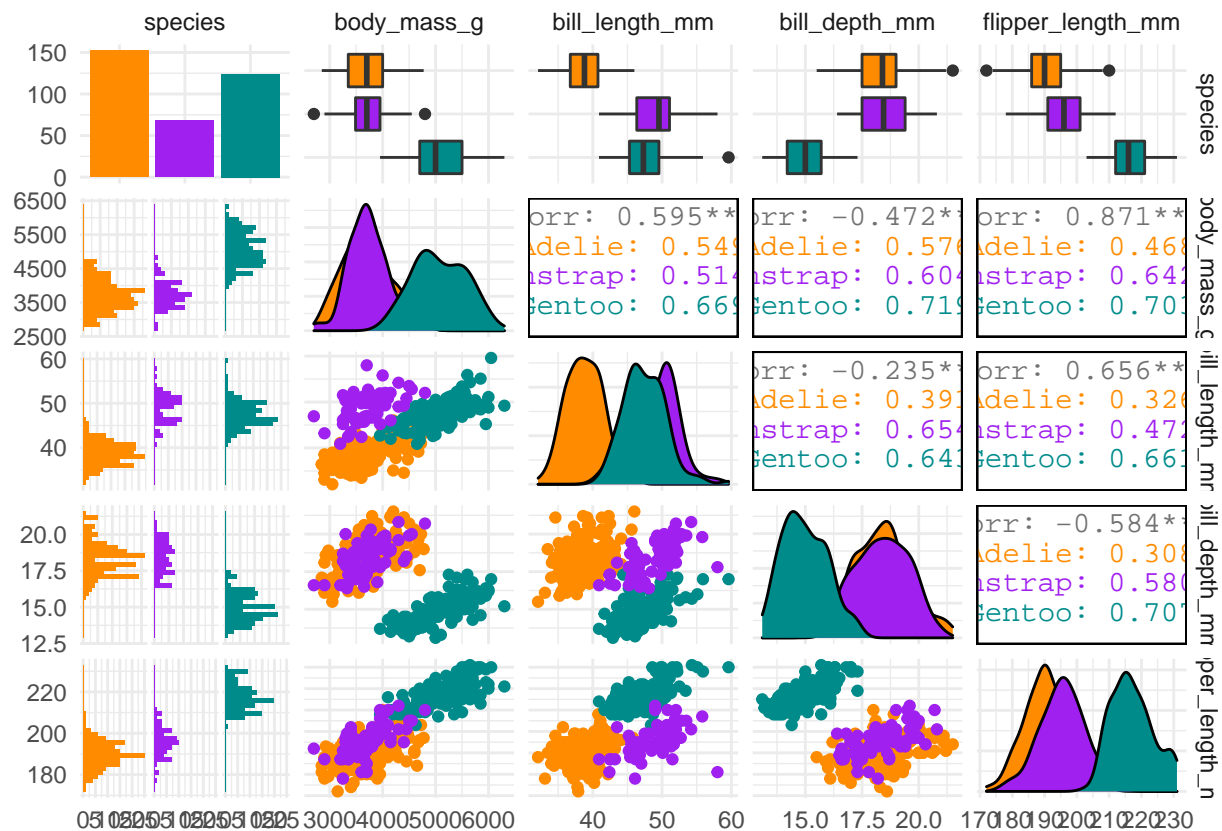
```
## Rows: 344
## Columns: 4
## $ body_mass_g      <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 347...
## $ bill_length_mm   <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34....
## $ bill_depth_mm    <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18....
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, ...
```

```
# Scatterplot example 1: penguin flipper length versus body mass
ggplot(data = penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point(aes(color = species,
                  shape = species),
            size = 2) +
  scale_color_manual(values = c("darkorange", "darkorchid", "cyan4"))
```



The above scatterplot of body mass shows a strong similarity between the species Adelie and Chinstrap with a clear difference from Gentoo. A valuable observation in regards to the binary logistic regression model.

```
ds %>%
  dplyr::select(species, body_mass_g, ends_with("_mm")) %>%
  GGally::ggpairs(aes(color = species)) +
  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
  scale_fill_manual(values = c("darkorange", "purple", "cyan4"))
```



The above plot shows the additional strong similarities between the Adelie and Chinstrap species as compared to the Gentoo species.

## Model Definitions

Prep dataset for logistic regression. First step is to remove rows containing an NA.

```
# Create dataset for binary logistic regression: species Gentoo or Not
data_binary <- penguins

# Only use complete instances ... actually come back to this as I don't want to exclude because of sex
train_data_binary <- na.omit(data_binary)

dim(train_data_binary)

## [1] 333 8
```

Based on the result, 11 rows are removed, which would equal the number of NAs in variable *sex*.

## Binary Logistic Regression

The following approach attempts to construct a logistic regression model based on a binary outcome. As the penguins dataset is based on a dependent variable (species) containing three values, a dummy variable



*Gentoo* is defined to identify penguins of the species *Gentoo* or of the other two values (*Adelie* and *Chinstrap*). Based on the exploratory data analysis indicating independent variable overlap for body mass, bill depth, and flipper length between the *Adelie* and *Chinstrap* species, the decision was made to group these two species based on the similarities.

```
# Create new column
train_data_binary$gentoo <- ifelse(train_data_binary$species=="Gentoo", 1, 0)

summary(train_data_binary)
```

```
##      species      island bill_length_mm bill_depth_mm
## Adelie   :146   Biscoe   :163   Min.    :32.10   Min.    :13.10
## Chinstrap: 68   Dream    :123   1st Qu.:39.50   1st Qu.:15.60
## Gentoo   :119   Torgersen: 47   Median :44.50   Median :17.30
##                                     Mean    :43.99   Mean    :17.16
##                                     3rd Qu.:48.60   3rd Qu.:18.70
##                                     Max.    :59.60   Max.    :21.50
## flipper_length_mm body_mass_g      sex      year      gentoo
## Min.      :172      Min.      :2700 female:165   Min.      :2007   Min.      :0.0000
## 1st Qu.   :190      1st Qu.   :3550 male  :168   1st Qu.   :2007   1st Qu.   :0.0000
## Median    :197      Median    :4050                Median :2008   Median    :0.0000
## Mean      :201      Mean      :4207                Mean   :2008   Mean      :0.3574
## 3rd Qu.   :213      3rd Qu.   :4775                3rd Qu.:2009   3rd Qu.   :1.0000
## Max.      :231      Max.      :6300                Max.    :2009   Max.      :1.0000
```

With the derived dummy variable *Gentoo*, the variable *species* is removed from the initial dataset, so as not to impact the logistic regression models.

```
# Drop species column, as now just using gentoo column as Y variable

drops <- c("species")
train_data_binary <- train_data_binary[ , !(names(train_data_binary) %in% drops)]

summary(train_data_binary)
```

```
##      island      bill_length_mm bill_depth_mm flipper_length_mm
## Biscoe   :163   Min.    :32.10   Min.    :13.10   Min.    :172
## Dream    :123   1st Qu.:39.50   1st Qu.:15.60   1st Qu.:190
## Torgersen: 47   Median :44.50   Median :17.30   Median :197
##                                     Mean    :43.99   Mean    :17.16   Mean    :201
##                                     3rd Qu.:48.60   3rd Qu.:18.70   3rd Qu.:213
##                                     Max.    :59.60   Max.    :21.50   Max.    :231
## body_mass_g      sex      year      gentoo
## Min.      :2700 female:165   Min.      :2007   Min.      :0.0000
## 1st Qu.   :3550 male  :168   1st Qu.   :2007   1st Qu.   :0.0000
## Median    :4050                Median :2008   Median    :0.0000
## Mean      :4207                Mean   :2008   Mean      :0.3574
## 3rd Qu.   :4775                3rd Qu.:2009   3rd Qu.   :1.0000
## Max.      :6300                Max.    :2009   Max.      :1.0000
```

In order to validate the models properly, the initial penguins dataset is partitioned into training data at 70% of the given dataset with the remaining 30% used as test data completely unseen by the model.

```
set.seed(123)
trainIndex <- createDataPartition(train_data_binary$gentoo, p = 0.7, list = FALSE, times = 1)
train <- train_data_binary[trainIndex,]
test <- train_data_binary[-trainIndex,]
```

Three versions of a binary logistic regression model are constructed in order to evaluate the accuracy of each and also provide to narrow the model to the least number of variables to identify the most parsimonious model.

## Baseline Model

The first model uses all the available independent variables in order to define a baseline evaluation of the model.

```
# All variables
modell1 <- glm(gentoo ~ ., data = train, family = "binomial"(link="logit"))
#Accuracy 100%, AIC is 18
summary(modell1)
```

```
##
## Call:
## glm(formula = gentoo ~ ., family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.722e-05 -2.100e-08 -2.100e-08  2.100e-08  2.985e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.439e+03  1.581e+08      0      1
## islandDream    -1.043e+01  1.197e+05      0      1
## islandTorgersen -1.180e+01  1.124e+05      0      1
## bill_length_mm   7.064e-01  1.136e+04      0      1
## bill_depth_mm   -9.278e+00  3.578e+04      0      1
## flipper_length_mm 9.491e-01  6.324e+03      0      1
## body_mass_g      1.516e-02  1.527e+02      0      1
## sexmale          1.990e+00  1.590e+05      0      1
## year            -3.773e+00  7.865e+04      0      1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 3.0884e+02  on 233  degrees of freedom
## Residual deviance: 4.3248e-09  on 225  degrees of freedom
## AIC: 18
##
## Number of Fisher Scoring iterations: 25
```

Resulting AIC: 18.

## Stepwise Model

Next, the *stepAIC* function is applied to the full model to determine the most predictive variables for the model.

```
# All variables then applied with stepAIC
model2 <- glm(gentoo ~ ., data = train, family = "binomial"(link="logit")) %>% stepAIC(trace=F, direction="both")
# Accuracy 100% an AIC is 6
summary(model2)
```

```
##
## Call:
## glm(formula = gentoo ~ bill_depth_mm + flipper_length_mm, family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6.826e-05 -2.100e-08 -2.100e-08  2.100e-08  6.510e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -428.130  474946.766  -0.001    0.999
## bill_depth_mm    -14.834  12021.243  -0.001    0.999
## flipper_length_mm   3.274   1957.819   0.002    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 3.0884e+02  on 233  degrees of freedom
## Residual deviance: 9.5967e-09  on 231  degrees of freedom
## AIC: 6
##
## Number of Fisher Scoring iterations: 25
```

Resulting AIC: 6.

## Hand Selected Model

Finally, a hand-selected list of independent variables are chosen based on the evaluation of the exploratory data analysis.

```
# Hand selected variables
model3 <- glm(gentoo ~ island + bill_depth_mm + flipper_length_mm + body_mass_g, data = train, family = "binomial"(link="logit"))
# Accuracy 100%, AIC is 12
summary(model3)
```

```
##
## Call:
## glm(formula = gentoo ~ island + bill_depth_mm + flipper_length_mm +
##      body_mass_g, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -4.069e-05 -2.100e-08 -2.100e-08 2.100e-08 2.804e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.402e+02  6.440e+05  0.000    1.000
## islandDream   -1.390e+00  4.055e+04  0.000    1.000
## islandTorgersen -5.044e+00  7.407e+04  0.000    1.000
## bill_depth_mm  -1.049e+01  1.142e+04 -0.001    0.999
## flipper_length_mm 1.098e+00  3.818e+03  0.000    1.000
## body_mass_g     1.958e-02  5.657e+01  0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3.0884e+02 on 233 degrees of freedom
## Residual deviance: 4.7527e-09 on 228 degrees of freedom
## AIC: 12
##
## Number of Fisher Scoring iterations: 25
```

Resulting AIC: 12.

## Make predictions

Predictions are performed on the test dataset based on the three above binary logistic regression models.

```
## use the test data set to make predictions for the 3 models
mod1.predict.probs <- predict.glm(model1, type="response", newdata=test)
mod1.predict.manual <- ifelse(mod1.predict.probs > 0.5, '1','0')
attach(test)

mod2.predict.probs <- predict.glm(model2, type="response", newdata=test)
mod2.predict.manual <- ifelse(mod2.predict.probs > 0.5, '1','0')
attach(test)

mod3.predict.probs <- predict.glm(model3, type="response", newdata=test)
mod3.predict.manual <- ifelse(mod3.predict.probs > 0.5, '1','0')
attach(test)
```

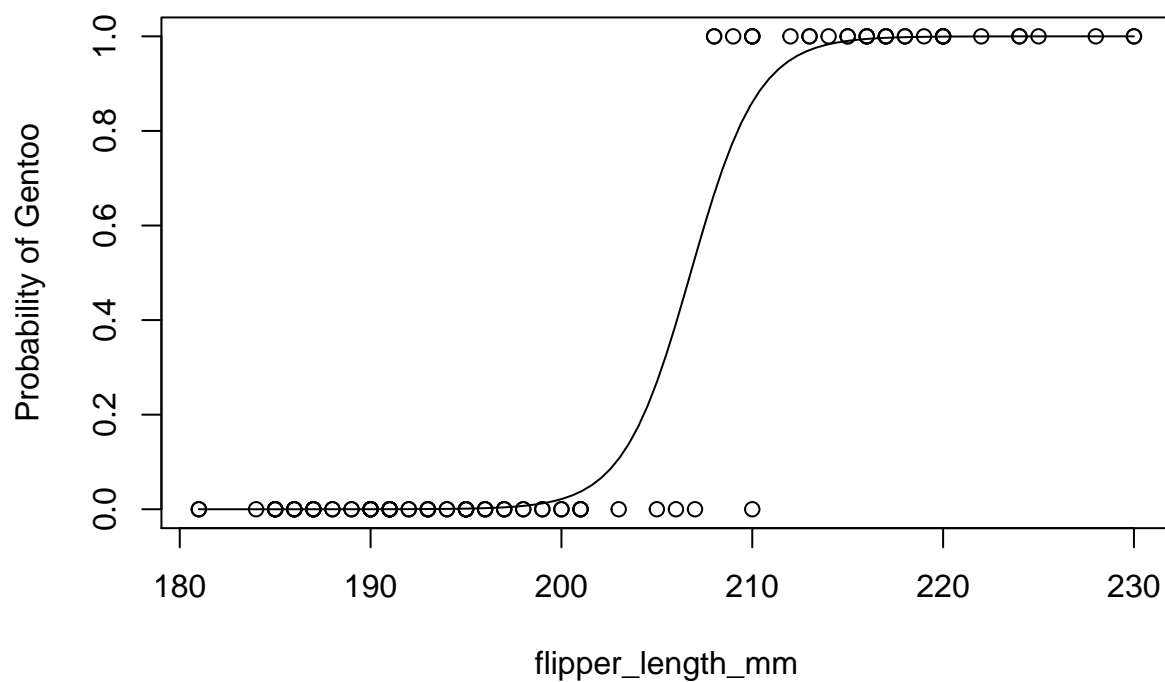
## Model Visualizations

Plots of the data to visualize the independent variable's value compared to the logit function of the dependent variable.

First, plot the variable *flipper\_length<sub>m</sub>* against the logit value of the *Gentoo* result.

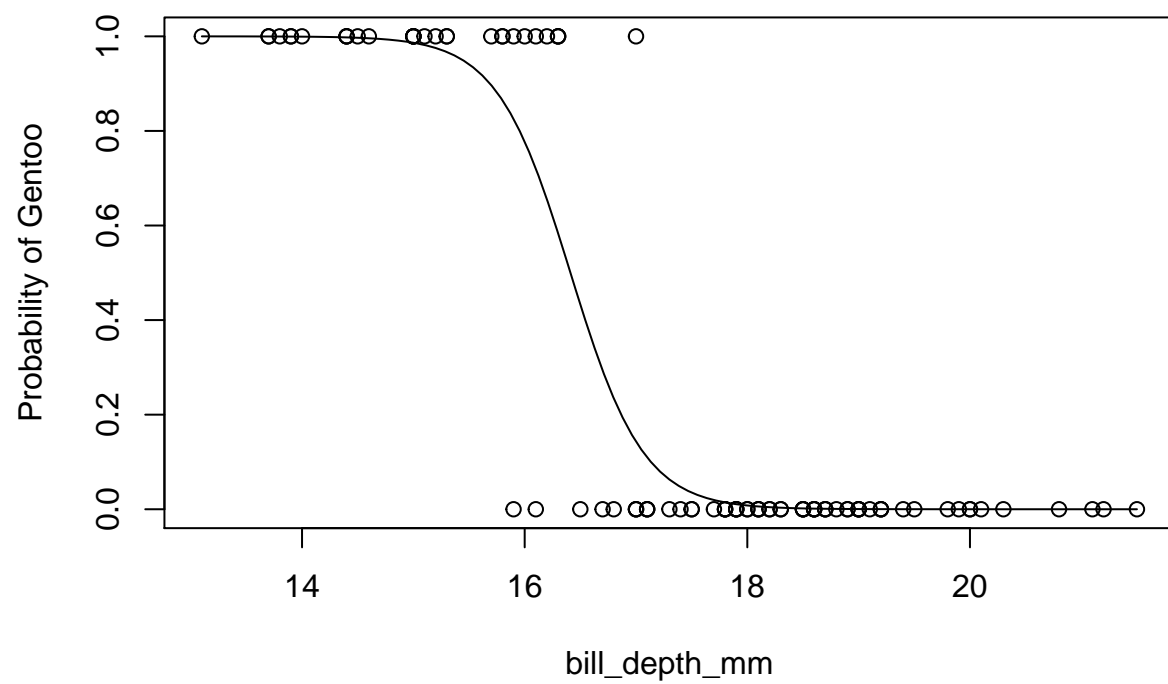
```
# Plot the dependent variable interpretation
# https://sites.google.com/site/daishizuka/toolkits/plotting-logistic-regression-in-r

# plot with flipper_length_mm on x-axis and Gentoo species (0 or 1) on y-axis
plot(flipper_length_mm, gentoo, xlab="flipper_length_mm", ylab="Probability of Gentoo")
g=glm(gentoo ~ flipper_length_mm, data = train, family = "binomial"(link="logit"))
curve(predict(g, data.frame(flipper_length_mm=x), type="resp"), add=TRUE)
```

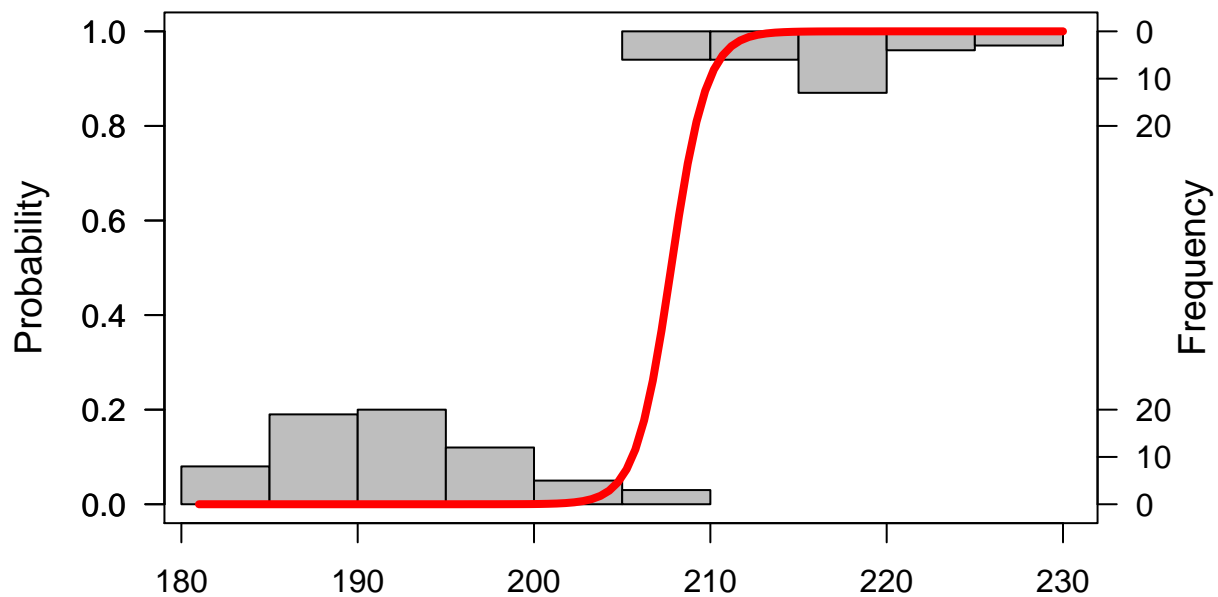


Second, plot the variable *bill\_depth\_mm* against the logit value of the *Gentoo* result.

```
# plot with bill_depth_mm on x-axis and Gentoo species (0 or 1) on y-axis
plot(bill_depth_mm, gentoo, xlab="bill_depth_mm", ylab="Probability of Gentoo")
g=glm(gentoo ~ bill_depth_mm, data = train, family = "binomial"(link="logit"))
curve(predict(g, data.frame(bill_depth_mm=x), type="resp"), add=TRUE)
```



```
# plot using another function
logi.hist.plot(flipper_length_mm,gentoo,boxp=FALSE,type="hist",col="gray")
```



Third plot above was just an attempt to use another library function for plotting the data against the logit function.

## Model 1 Results

The baseline model shows:

- Accuracy: 1 or 100%
- Area Under the Curve: 1.0
- True Positive Rate (Sensitivity): 1.0
- True Negative Rate (Specificity): 1.0
- False Negative Rate (Miss Rate: 1-TPR): 0
- False Positive Rate (Fall-out: 1-TNR): 0

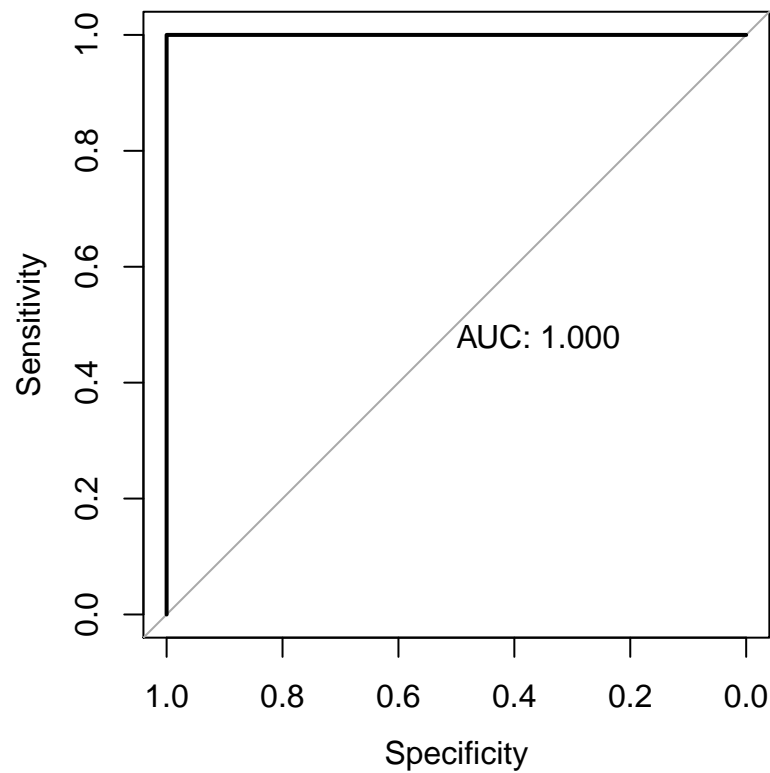
```
# Model1
# now can use the caret function
cm.var <- caret::confusionMatrix(factor(mod1.predict.manual), factor(test$gentoo), positive='1')
cm.var$table
```

```
##           Reference
## Prediction  0  1
##           0 67  0
##           1  0 32
```

```
# print metrics
mod1.CMmetrics <- c(cm.var$overall[c(1)], cm.var$byClass[c(1,2,5,6,7)])
mod1.CMmetrics

##      Accuracy Sensitivity Specificity Precision Recall      F1
##           1           1           1           1           1           1

# ROC and AUC
par(pty="s")
roc.stepwise <- roc(train$gentoo, model1$fitted.values, plot=TRUE, print.auc=TRUE)
```



## Model 2 Results

The *stepAIC* model shows:

- Accuracy: 1 or 100%
- Area Under the Curve: 1.0
- True Positive Rate (Sensitivity): 1.0
- True Negative Rate (Specificity): 1.0
- False Negative Rate (Miss Rate: 1-TPR): 0
- False Positive Rate (Fall-out: 1-TNR): 0



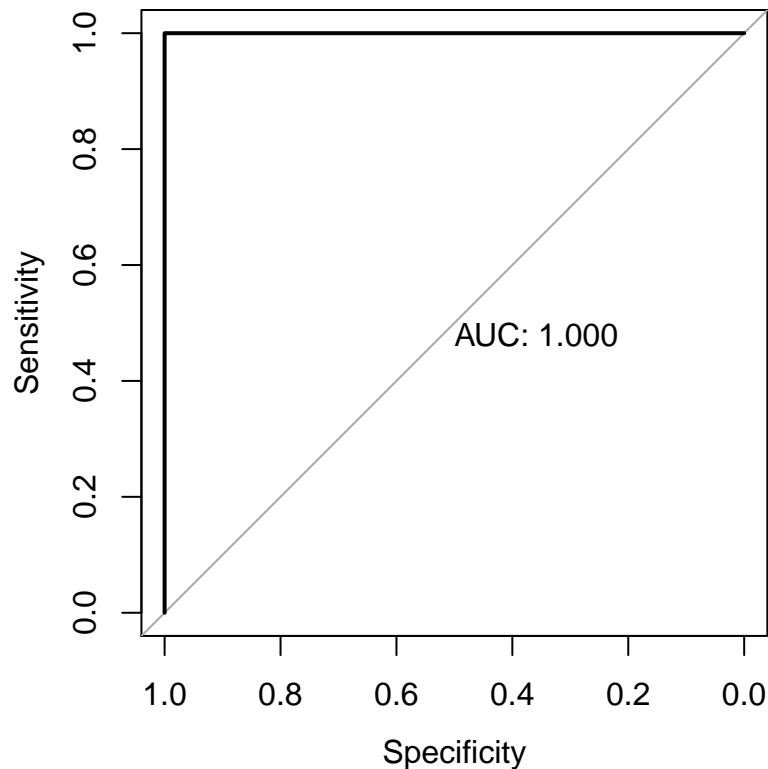
```
# Model2
# now can use the caret function
cm.var <- caret::confusionMatrix(factor(mod2.predict.manual), factor(test$gentoo), positive='1')
cm.var$table
```

```
##           Reference
## Prediction  0  1
##           0 67  0
##           1  0 32
```

```
# print metrics
mod2.CMmetrics <- c(cm.var$overall[c(1)], cm.var$byClass[c(1,2,5,6,7)])
mod2.CMmetrics
```

```
##      Accuracy Sensitivity Specificity Precision Recall      F1
##           1           1           1           1           1           1
```

```
# ROC and AUC
par(pty="s")
roc.stepwise <- roc(train$gentoo, model2$fitted.values, plot=TRUE, print.auc=TRUE)
```



### Model 3 Results

The hand-selected model shows:

- Accuracy: 1 or 100%
- Area Under the Curve: 1.0
- True Positive Rate (Sensitivity): 1.0
- True Negative Rate (Specificity): 1.0
- False Negative Rate (Miss Rate: 1-TPR): 0
- False Positive Rate (Fall-out: 1-TNR): 0

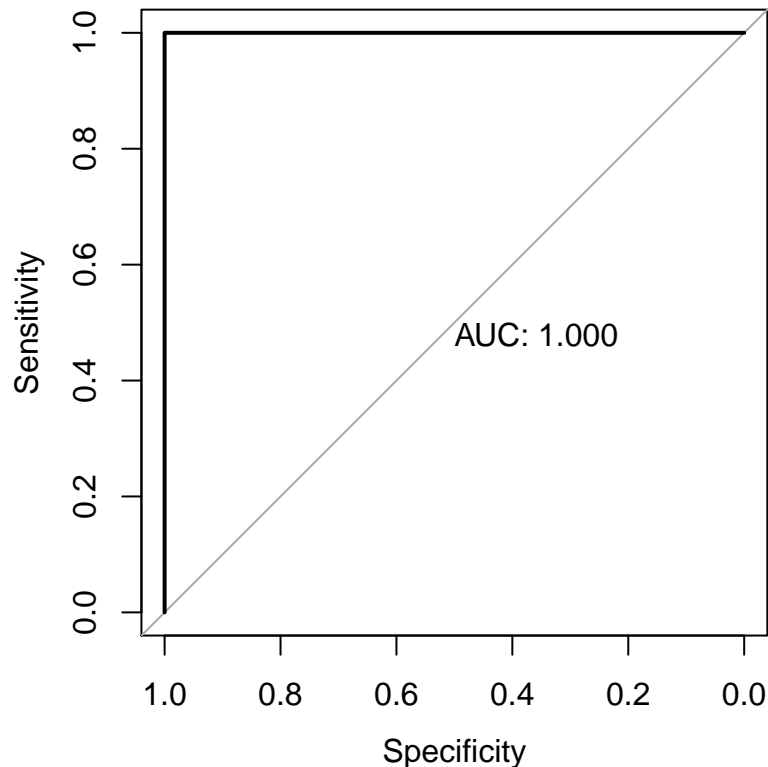
```
# Model3
# now can use the caret function
cm.var <- caret::confusionMatrix(factor(mod3.predict.manual), factor(test$gentoo), positive='1')
cm.var$table
```

```
##           Reference
## Prediction  0   1
##           0 67   0
##           1   0 32
```

```
# print metrics
mod3.CMmetrics <- c(cm.var$overall[c(1)], cm.var$byClass[c(1,2,5,6,7)])
mod3.CMmetrics
```

```
##      Accuracy Sensitivity Specificity   Precision      Recall      F1
##           1           1           1           1           1           1
```

```
# ROC and AUC
par(pty="s")
roc.stepwise <- roc(train$gentoo, model3$fitted.values, plot=TRUE, print.auc=TRUE)
```



Looking at the summary results for the three models attempting to identify Gentoo species or not, I decided to make the goal a bit more difficult as Adelie species appears on all three islands, and statistically, the Adelie species does overlap with Chinstrap, I decided to create logistic regression models to identify Adelie instead of Gentoo.

The first model uses the same baseline model approach as above including all independent variables. The second model re-uses the *stepAIC* approach to algorithmically select the best independent variables.

```
# Create dataset for binary logistic regression: species Adelie or Not
data_binary <- penguins

# Only use complete instances ... actually come back to this as I don't want to exclude because of sex
train_data_binary <- na.omit(data_binary)

train_data_binary$adelie <- ifelse(train_data_binary$species=="Adelie", 1, 0)

summary(train_data_binary)
```

```
##      species      island  bill_length_mm  bill_depth_mm
## Adelie   :146  Biscoe   :163   Min.    :32.10   Min.    :13.10
## Chinstrap: 68  Dream    :123   1st Qu.:39.50   1st Qu.:15.60
## Gentoo   :119  Torgersen: 47   Median :44.50   Median :17.30
##                                     Mean    :43.99   Mean    :17.16
##                                     3rd Qu.:48.60   3rd Qu.:18.70
##                                     Max.    :59.60   Max.    :21.50
## flipper_length_mm  body_mass_g      sex      year      adelie
```

```
## Min.      :172      Min.      :2700   female:165   Min.      :2007   Min.      :0.0000
## 1st Qu.:190      1st Qu.:3550   male  :168   1st Qu.:2007   1st Qu.:0.0000
## Median :197      Median :4050                      Median :2008   Median :0.0000
## Mean    :201      Mean    :4207                      Mean    :2008   Mean    :0.4384
## 3rd Qu.:213      3rd Qu.:4775                      3rd Qu.:2009   3rd Qu.:1.0000
## Max.    :231      Max.    :6300                      Max.    :2009   Max.    :1.0000
```

```
drops <- c("species")
train_data_binary <- train_data_binary[ , !(names(train_data_binary) %in% drops)]

set.seed(123)
trainIndex <- createDataPartition(train_data_binary$adelie, p = 0.7, list = FALSE, times = 1)
train <- train_data_binary[trainIndex,]
test <- train_data_binary[-trainIndex,]

# All variables
modell1_ad <- glm(adelie ~ ., data = train, family = "binomial"(link="logit"))
summary(modell1_ad)
```

```
##
## Call:
## glm(formula = adelie ~ ., family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -8.484e-05 -2.100e-08 -2.100e-08  2.100e-08  8.052e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.625e+04  5.517e+07  0.000    1.000
## islandDream   -8.202e+00  1.179e+05  0.000    1.000
## islandTorgersen 1.094e+01  1.264e+05  0.000    1.000
## bill_length_mm -2.301e+01  7.614e+03 -0.003    0.998
## bill_depth_mm  2.986e+01  1.415e+04  0.002    0.998
## flipper_length_mm -1.174e+00  4.091e+03  0.000    1.000
## body_mass_g    3.811e-02  6.454e+01  0.001    1.000
## sexmale        9.225e+00  4.531e+04  0.000    1.000
## year           8.359e+00  2.785e+04  0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 3.2103e+02  on 233  degrees of freedom
## Residual deviance: 2.8522e-08  on 225  degrees of freedom
## AIC: 18
##
## Number of Fisher Scoring iterations: 25
```

```
# All variables then applied with stepAIC
modell2_ad <- glm(adelie ~ ., data = train, family = "binomial"(link="logit")) %>% stepAIC(trace=F, direction="both")
summary(modell2_ad)
```

```
##
```

```
## Call:
## glm(formula = adellie ~ bill_length_mm + bill_depth_mm + body_mass_g,
##      family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.024e-04 -2.100e-08 -2.100e-08  2.100e-08  1.099e-04
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.328e+02  2.819e+05   0.001   0.999
## bill_length_mm -2.980e+01  7.288e+03  -0.004   0.997
## bill_depth_mm   4.525e+01  1.268e+04   0.004   0.997
## body_mass_g     3.582e-02  1.075e+01   0.003   0.997
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3.2103e+02 on 233 degrees of freedom
## Residual deviance: 4.3658e-08 on 230 degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25
```

```
## use the test data set to make predictions for the 3 models
mod1_ad.predict.probs <- predict.glm(model1_ad, type="response", newdata=test)
mod1_ad.predict.manual <- ifelse(mod1_ad.predict.probs > 0.5, '1','0')
attach(test)

mod2_ad.predict.probs <- predict.glm(model2_ad, type="response", newdata=test)
mod2_ad.predict.manual <- ifelse(mod2_ad.predict.probs > 0.5, '1','0')
attach(test)

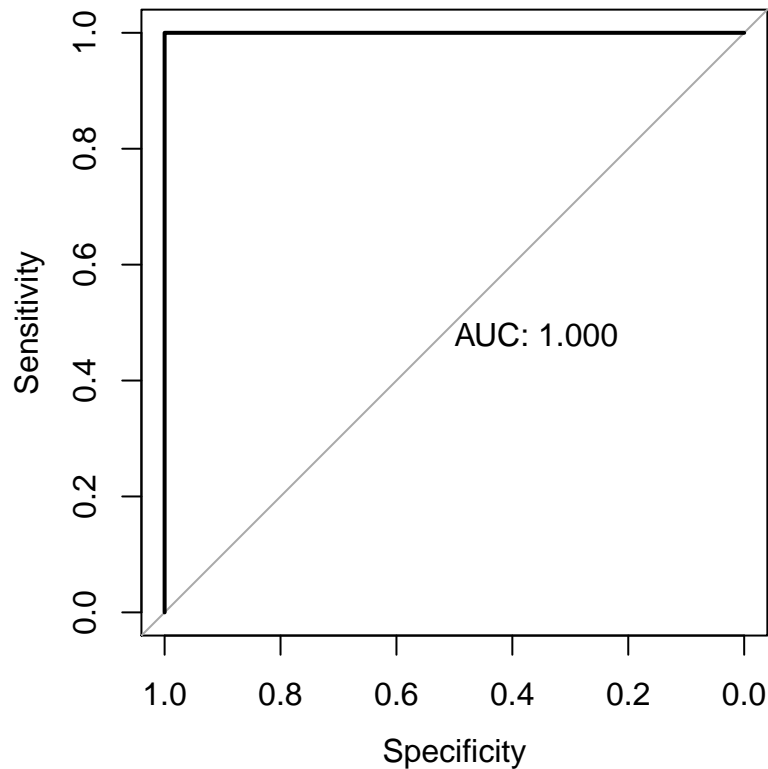
# Model1
# now can use the caret function
cm.var <- caret::confusionMatrix(factor(mod1_ad.predict.manual), factor(test$adellie), positive='1')
cm.var$table
```

```
##           Reference
## Prediction  0  1
##           0 56  2
##           1  0 41
```

```
# print metrics
mod1_ad.CMmetrics <- c(cm.var$overall[c(1)], cm.var$byClass[c(1,2,5,6,7)])
mod1_ad.CMmetrics
```

```
## Accuracy Sensitivity Specificity Precision Recall F1
## 0.9797980 0.9534884 1.0000000 1.0000000 0.9534884 0.9761905
```

```
# ROC and AUC
par(pty="s")
roc.stepwise <- roc(train$adellie, model1_ad$fitted.values, plot=TRUE, print.auc=TRUE)
```



The baseline model shows:

- Accuracy: 0.9797980 or ~98%
- Area Under the Curve: 1.0
- True Positive Rate (Sensitivity): 0.9534884
- True Negative Rate (Specificity): 1.0
- False Negative Rate (Miss Rate: 1-TPR): 0.0465116
- False Positive Rate (Fall-out: 1-TNR): 0

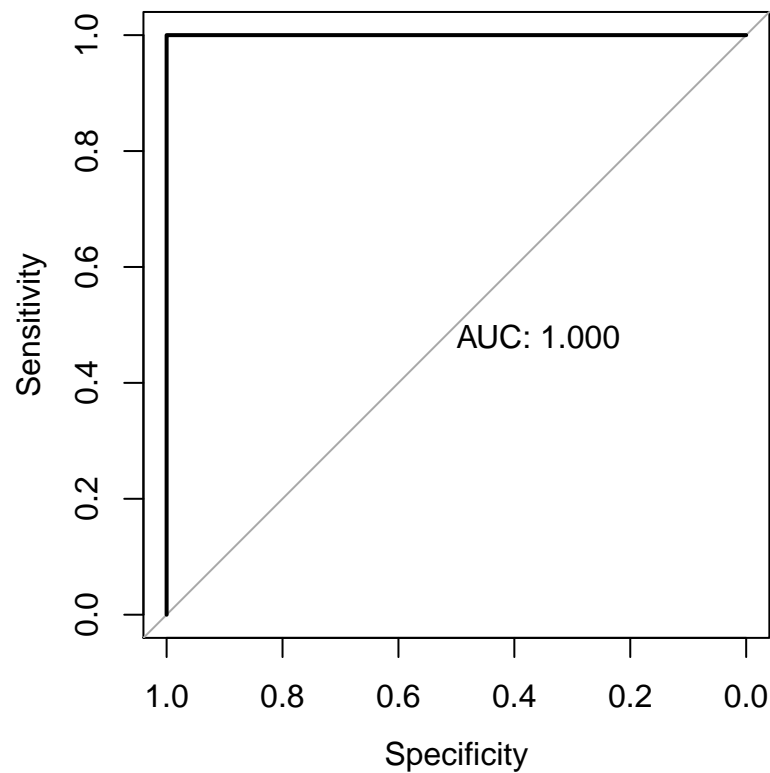
```
# Model2
# now can use the caret function
cm.var <- caret::confusionMatrix(factor(mod2_ad.predict.manual), factor(test$adelie), positive='1')
cm.var$table
```

```
##           Reference
## Prediction  0  1
##           0 56  2
##           1  0 41
```

```
# print metrics
mod2_ad.CMmetrics <- c(cm.var$overall[c(1)], cm.var$byClass[c(1,2,5,6,7)])
mod2_ad.CMmetrics
```

##	Accuracy	Sensitivity	Specificity	Precision	Recall	F1
##	0.9797980	0.9534884	1.0000000	1.0000000	0.9534884	0.9761905

```
# ROC and AUC
par(pty="s")
roc.stepwise <- roc(train$adelie, model2_ad$fitted.values, plot=TRUE, print.auc=TRUE)
```



The *stepAIC* model shows:

- Accuracy: 0.9797980 or ~98%
- Area Under the Curve: 1.0
- True Positive Rate (Sensitivity): 0.9534884
- True Negative Rate (Specificity): 1.0
- False Negative Rate (Miss Rate: 1-TPR): 0.0465116
- False Positive Rate (Fall-out: 1-TNR): 0

Interestingly, the *stepAIC* did not perform at 100% accuracy. The *stepAIC* model indicates *bill\_length\_mm*, *bill\_depth\_mm*, and *body\_mass\_g* are the three most predictive independent variables.

## Multinomial Logistic Regression

The following approach attempts to construct a multinomial logistic regression model based on a multivariate outcome. As the penguins dataset is based on a dependent variable (species) containing three values, these models attempt to predict the species of each penguin subject.

```
# Initial walk-through: https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/
```

```
# Start with initial dataset
```

```
mlr_data <- penguins
```

```
summary(mlr_data)
```

```
##      species      island bill_length_mm bill_depth_mm
## Adelie   :152  Biscoe   :168   Min.    :32.10   Min.    :13.10
## Chinstrap: 68  Dream    :124   1st Qu.:39.23   1st Qu.:15.60
## Gentoo   :124  Torgersen: 52   Median :44.45   Median :17.30
##
##                               Mean    :43.92   Mean    :17.15
##                               3rd Qu.:48.50   3rd Qu.:18.70
##                               Max.    :59.60   Max.    :21.50
##                               NA's    :2       NA's    :2
## flipper_length_mm body_mass_g      sex      year
## Min.    :172.0     Min.    :2700   female:165   Min.    :2007
## 1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
## Median :197.0     Median :4050   NA's   : 11   Median :2008
## Mean    :200.9     Mean    :4202                   Mean    :2008
## 3rd Qu.:213.0     3rd Qu.:4750                   3rd Qu.:2009
## Max.    :231.0     Max.    :6300                   Max.    :2009
## NA's    :2        NA's    :2
```

```
mlr_data$species2 <- relevel(mlr_data$species, ref = "Gentoo")
```

```
test <- multinom(species2 ~ body_mass_g + bill_length_mm + bill_depth_mm + flipper_length_mm + island, data = mlr_data)
```

```
## # weights:  24 (14 variable)
## initial value 375.725403
## iter  10 value 20.138634
## iter  20 value 1.481353
## iter  30 value 0.035234
## iter  40 value 0.000933
## iter  50 value 0.000233
## final value 0.000089
## converged
```

```
summary(test)
```

```
## Call:
```

```
## multinom(formula = species2 ~ body_mass_g + bill_length_mm +
##          bill_depth_mm + flipper_length_mm + island, data = mlr_data)
##
```

```
## Coefficients:
```

```
##          (Intercept) body_mass_g bill_length_mm bill_depth_mm
## Adelie      179.0566 -0.01156190    -11.117549     15.793011
```



```
## Chinstrap    -151.2783 -0.04191271      4.802821      -5.639087
##              flipper_length_mm islandDream islandTorgersen
## Adelie       0.4109570    22.48619      69.34019
## Chinstrap    0.6913637    117.34484      16.71717
##
## Std. Errors:
##              (Intercept) body_mass_g bill_length_mm bill_depth_mm
## Adelie       0.736784    0.4581624      49.98112      14.88575
## Chinstrap    0.732472    0.7914301      49.74986      14.71006
##              flipper_length_mm islandDream islandTorgersen
## Adelie       6.554627    0.732472    1.236702e-24
## Chinstrap    6.545027    0.732472    1.876044e-50
##
## Residual Deviance: 0.0001778793
## AIC: 28.00018
```

```
stargazer(test, type="text", out="test.htm")
```

```
##
## =====
##              Dependent variable:
##              -----
##              Adelie      Chinstrap
##              (1)         (2)
##              -----
## body_mass_g      -0.012      -0.042
##                  (0.458)      (0.791)
##
## bill_length_mm   -11.118       4.803
##                  (49.981)      (49.750)
##
## bill_depth_mm    15.793      -5.639
##                  (14.886)      (14.710)
##
## flipper_length_mm 0.411       0.691
##                  (6.555)      (6.545)
##
## islandDream      22.486***     117.345***
##                  (0.732)      (0.732)
##
## islandTorgersen  69.340***     16.717***
##                  (0.000)      (0.000)
##
## Constant         179.057***    -151.278***
##                  (0.737)      (0.732)
##
## -----
## Akaike Inf. Crit.    28.000      28.000
## =====
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
test.rrr = exp(coef(test))
test.rrr
```

```
##          (Intercept) body_mass_g bill_length_mm bill_depth_mm
## Adelie      5.798378e+77  0.9885047  1.484942e-05  7.224660e+06
## Chinstrap 1.998367e-66  0.9589535  1.218537e+02  3.556113e-03
##          flipper_length_mm islandDream islandTorgersen
## Adelie          1.508260 5.829443e+09  1.300353e+30
## Chinstrap        1.996436 9.166779e+50  1.820439e+07
```

```
stargazer(test, type="text", coef=list(test.rrr), p.auto=FALSE, out="testrrr.htm")
```

```
##
## =====
##
##          -----
##                                     Adelie
##                                     (1)
## -----
## body_mass_g                      0.989
##                                     (0.458)
##
## bill_length_mm                   0.00001
##                                     (49.981)
##
## bill_depth_mm                   7,224,660.000
##                                     (14.886)
##
## flipper_length_mm                1.508
##                                     (6.555)
##
## islandDream                      5,829,443,146.000***
##                                     (0.732)
##
## islandTorgersen                  1,300,352,545,036,810,191,042,951,249,920.000***
##                                     (0.000)
##
## Constant          579,837,803,916,479,821,887,587,742,166,729,353,269,199,669,971,417,058,436,431,37
##                                     (0.737)
##
## -----
## Akaike Inf. Crit.                      28.000
## =====
## Note:
```

```
# Again with https://www.r-bloggers.com/2020/05/multinomial-logistic-regression-with-r/
```

```
index <- createDataPartition(mlr_data$species, p = .70, list = FALSE)
train <- mlr_data[index,]
test <- mlr_data[-index,]
```

```
# Set the reference
```

```
train$species <- relevel(train$species, ref = "Adelie")
```

```
# Training the multinomial model
```

```
#multinom_model <- multinom(species ~ ., data = mlr_data)
```

```
multinom_model <- multinom(species ~ island + bill_depth_mm + bill_length_mm, data = mlr_data)
```

```
## # weights: 18 (10 variable)
## initial value 375.725403
## iter 10 value 7.109357
## iter 20 value 3.080002
## iter 30 value 1.165609
## iter 40 value 0.972679
## iter 50 value 0.791804
## iter 60 value 0.629141
## iter 70 value 0.239584
## iter 80 value 0.210170
## iter 90 value 0.197773
## iter 100 value 0.180365
## final value 0.180365
## stopped after 100 iterations
```

```
#multinom_model <- multinom(species ~ flipper_length_mm + body_mass_g, data = mlr_data)
```

```
#with flipper length, and with body mass, will also get to one hundred accuracy
# but these 3 are required for 100: island + bill_depth_mm + bill_length_mm
```

```
# Checking the model
summary(multinom_model)
```

```
## Call:
## multinom(formula = species ~ island + bill_depth_mm + bill_length_mm,
## data = mlr_data)
##
## Coefficients:
## (Intercept) islandDream islandTorgersen bill_depth_mm bill_length_mm
## Chinstrap -94.4661532 4.04063 -16.83291 -14.19469 8.083423
## Gentoo -0.9047992 -18.48151 -14.78798 -21.82820 9.070280
##
## Std. Errors:
## (Intercept) islandDream islandTorgersen bill_depth_mm bill_length_mm
## Chinstrap 82.71917 55.74159 107.82684 13.12430 6.673409
## Gentoo 10.28023 86.00558 87.37543 23.86346 9.995773
##
## Residual Deviance: 0.3607304
## AIC: 20.36073
```

```
z <- summary(multinom_model)$coefficients/summary(multinom_model)$standard.errors
z
```

```
## (Intercept) islandDream islandTorgersen bill_depth_mm bill_length_mm
## Chinstrap -1.14201039 0.07248861 -0.1561106 -1.081558 1.2112884
## Gentoo -0.08801355 -0.21488736 -0.1692464 -0.914712 0.9074115
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##           (Intercept) islandDream islandTorgersen bill_depth_mm bill_length_mm
## Chinstrap  0.2534497   0.9422131      0.8759459      0.2794489      0.2257849
## Gentoo     0.9298659   0.8298551      0.8656028      0.3603428      0.3641892
```

```
# Convert the coefficients to odds by taking the exponential of the coefficients.
exp(coef(multinom_model))
```

```
##           (Intercept) islandDream islandTorgersen bill_depth_mm
## Chinstrap 9.416097e-42 5.686217e+01  4.892843e-08 6.844208e-07
## Gentoo    4.046231e-01 9.409816e-09  3.781492e-07 3.312341e-10
##           bill_length_mm
## Chinstrap      3240.307
## Gentoo         8693.059
```

```
head(pp <- fitted(multinom_model))
```

```
##   Adelie   Chinstrap   Gentoo
## 1      1 4.445222e-27 8.568186e-31
## 2      1 1.164683e-17 6.798019e-17
## 3      1 1.499238e-18 1.976066e-19
## 5      1 3.339494e-39 6.179533e-46
## 6      1 4.336355e-38 5.116583e-48
## 7      1 3.118785e-22 4.752617e-23
```

```
#dses <- data.frame(ses = c("low", "middle", "high"), write = mean(ml$write))
#predict(test, newdata = dses, "probs")
```

```
# What is this?
head(round(fitted(multinom_model), 2))
```

```
##   Adelie Chinstrap Gentoo
## 1      1         0      0
## 2      1         0      0
## 3      1         0      0
## 5      1         0      0
## 6      1         0      0
## 7      1         0      0
```

```
# Predicting and validating the model
```

```
# Predicting the values for train dataset
```

```
train$speciesPredicted <- predict(multinom_model, newdata = train, "class")
```

```
# Building classification table
```

```
tab <- table(train$species, train$speciesPredicted)
```

```
# Calculating accuracy - sum of diagonal elements divided by total obs
```

```
round((sum(diag(tab))/sum(tab))*100,2)
```

```
## [1] 100
```

```
# Predicting the class for test dataset
test$speciesPredicted <- predict(multinom_model, newdata = test, "class")

# Building classification table
tab <- table(test$species, test$speciesPredicted)
tab
```

```
##
##           Adelie Chinstrap Gentoo
## Adelie           44           0      0
## Chinstrap         0          20      0
## Gentoo            0           0     37
```

### Attempt at Extra Credit

After some Internet searching, it appears there isn't much direction in how to measure model fit for multinomial logistic regression models. Pearson residual and Overdispersion are ways to measure the model. Approaches to comparing two or more models would include likelihood ratio test, wald test, cross validation and parallel lines assumption.

Sex doesn't matter, Considering that I'm grouping Orange and Purple, probably don't use bill\_length\_mm, as that one shows purple and green have similar distribution

## Prompt

Considers for multinomial wald test LR test: likelihood ratio Cross validation parallel lines assumption

Ideas here: <https://stats.stackexchange.com/questions/145203/how-to-assess-if-a-model-is-good-in-multinomial-logistic-regression>