Blue Group Final Project

Sophie Kearney, Matthew Jensen, Caleb Ackman

Summary

It has been established that in the penguin dataset, penguin species can be predicted using the four body measurements: bill length, bill depth, flipper length, and body mass.

In this project, we were interested in using these variables to predict the sex of the penguin. In our data exploration we found clear differences between the sexes but also between the penguin species. Therefore, we decided to build three separate logistic models for each penguin species.

Each of our models performed realtively well, with accuracies around 80%. This means that we can accurate determine the sex of the penguin based on the four body measurements for each species of penguin.

Data Exploration

Filter out missing data

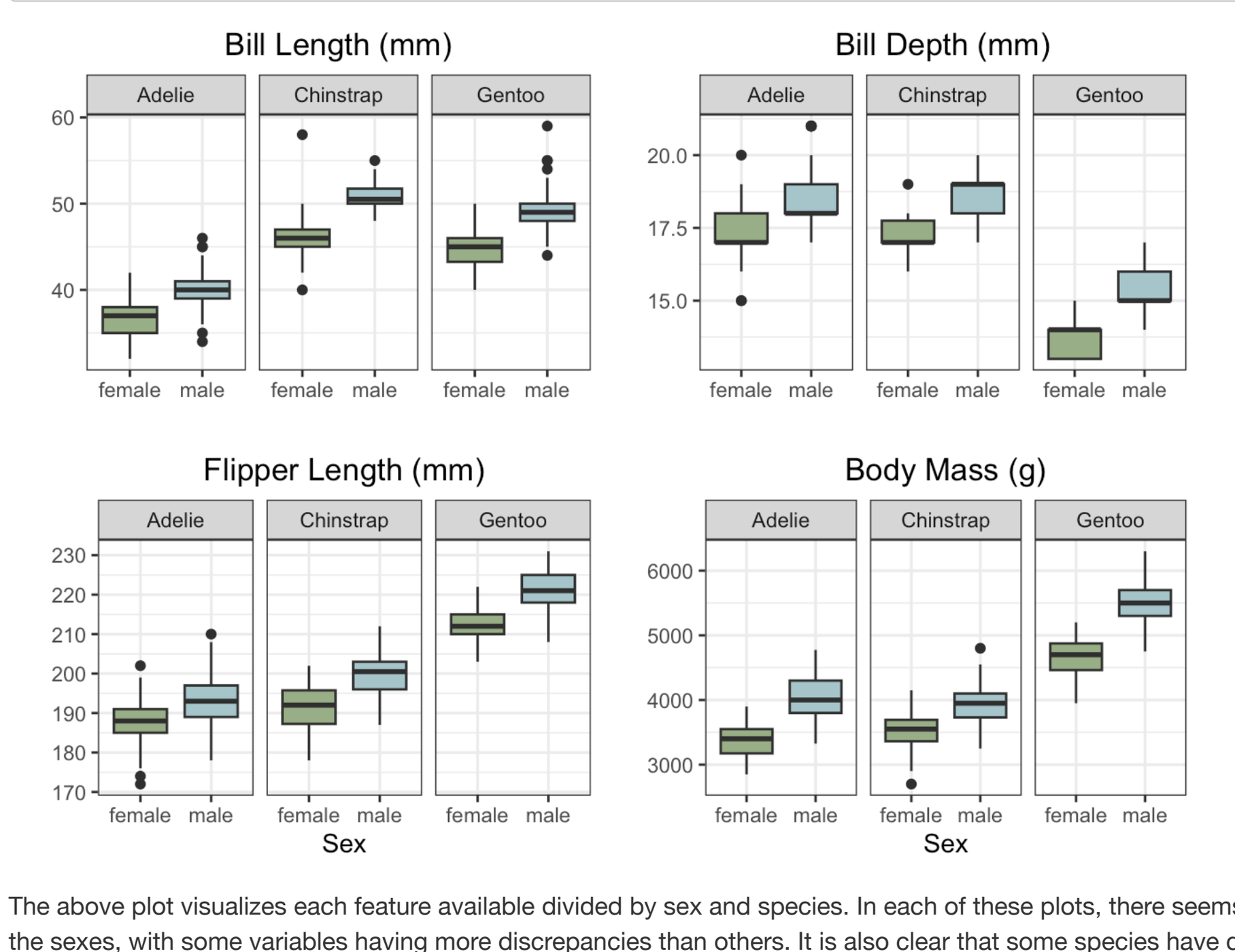
```
penguin_filter <- penguin[penguin$sex!=".",]
```

Visualize differences in response variables by sex and species

```
p1 <- ggplot(penguin_filter, aes(x=sex,y=as.integer(bill_length_mm),fill=sex)) +
  geom_boxplot() +
  facet_wrap(~species) +
  labs(x="",y="",title="Bill Length (mm)") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("#9caf88","#A7C5CB"))+
  guides(fill = FALSE)
```

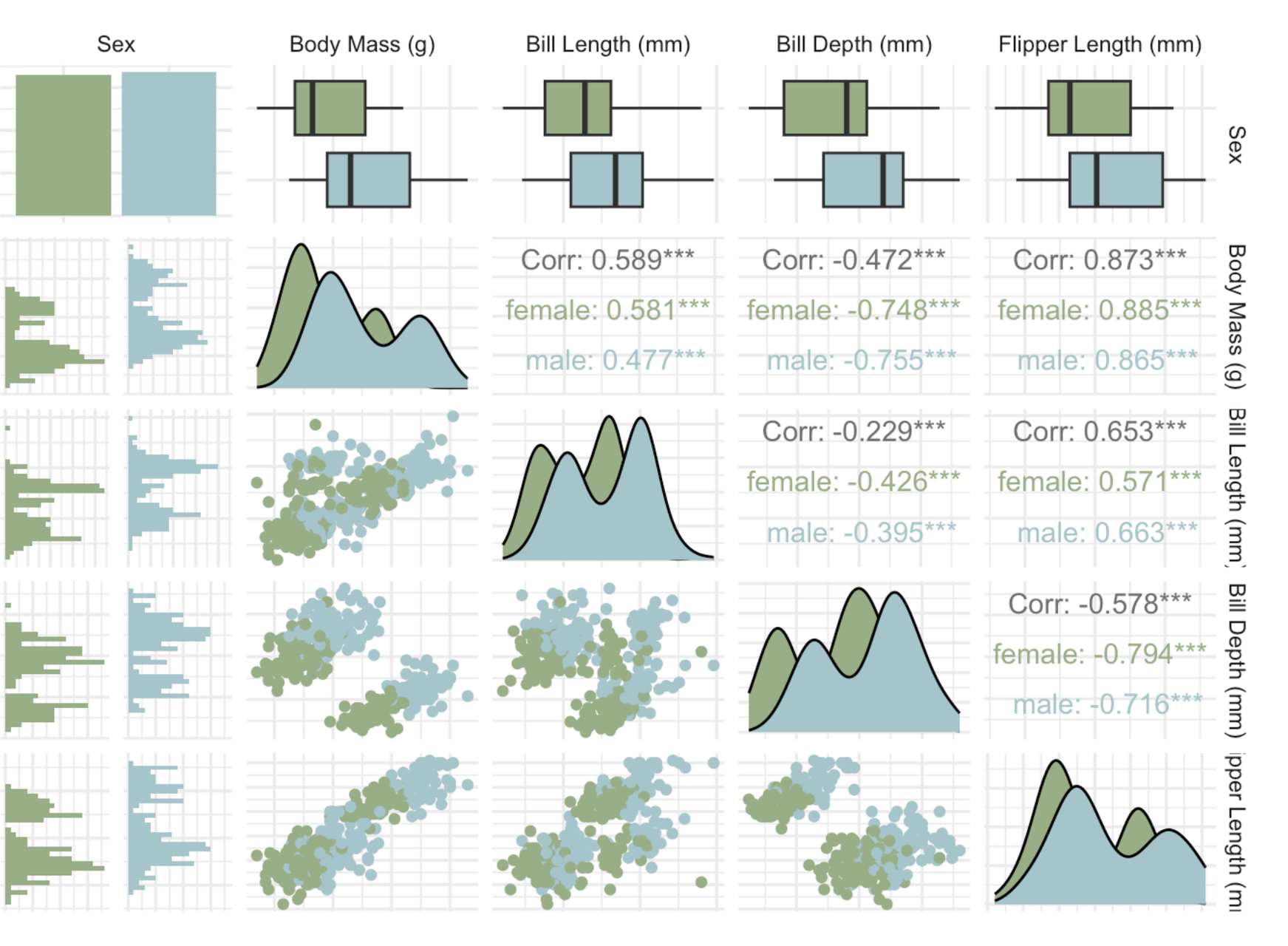
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p2 <- ggplot(penguin_filter, aes(x=sex,y=as.integer(bill_depth_mm),fill=sex)) +
  geom_boxplot() +
  facet_wrap(~species) +
  labs(x="",y="",title="Bill Depth (mm)") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("#9caf88","#A7C5CB"))+
  guides(fill = FALSE)

p3 <- ggplot(penguin_filter, aes(x=sex,y=as.integer(flipper_length_mm),fill=sex)) +
  geom_boxplot() +
  facet_wrap(~species) +
  labs(x="Sex",y="",title="Flipper Length (mm)") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("#9caf88","#A7C5CB"))+
  guides(fill = FALSE)

p4 <- ggplot(penguin_filter, aes(x=sex,y=as.integer(body_mass_g),fill=sex)) +
  geom_boxplot() +
  facet_wrap(~species) +
  labs(x="Sex",y="", title="Body Mass (g)") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("#9caf88","#A7C5CB"))+
  guides(fill = FALSE)

combined_plot <- grid.arrange(p1, p2, p3, p4, ncol = 2)
```



The above plot visualizes each feature available divided by sex and species. In each of these plots, there seems to be a clear distinction between the sexes, with some variables having more discrepancies than others. It is also clear that some species have drastically different values, such as the Gentoo bill depth being much lower than the Adelie and Chinstrap.

```
penguins_filter %>%
  select(Sex, `Body Mass (g)`, ends_with("(mm)")) %>%
  GGally::ggpairs(aes(color = Sex)) +
  scale_colour_manual(values = c("#9caf88","#A7C5CB")) +
  scale_fill_manual(values = c("#9caf88","#A7C5CB")) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank())
```



The plot above divides the dataset into male and female, with female being the light green and the male being the light blue values. Here, we wanted to visualize in several different ways the difference between the male and female values for each variable.

For example, the top row provides boxplots for each body measurement by sex. The next row down, body_mass_g, shows the distribution of the body mass across sexes in the first two cells.

The last few cells provide information on the correlation between body mass and each other numeric variable. A correlation closer to 1 or -1 signifies a strong positive or negative correlation. We can see a particularly strong correlation between body mass and flipper length with an overall correlation of .873. This makes sense because bigger, heavier penguins would have longer flippers.

This graph provides a good overall view of the data and highlights that there is a difference in each body measurement between sex, but there seems to be a significant overlap that comes with grouping all of the species together.

```
penguin_table <- table(penguin_filter$species, penguin_filter$sex)
kable(penguin_table, caption = "Penguin Species by Sex")
```

Penguin Species by Sex

	female	male
Adelie	73	73
Chinstrap	34	34
Gentoo	58	61

As we can see in this table, there seems to be a very even distribution of sexes across each penguin species. Additionally, there are quite a few samples in each sex still even when divided by species.

Because of the even distribution of sexes and the physical differences between each species, we decided to create 3 separate models to predict sex in each species.

Predicting Sex for the Adelie Species

Data Processing

```
# encode the female and male species into 0 and 1
penguin_filter$sex<-gsub("female", 0, penguin_filter$sex)
penguin_filter$sex<-gsub("male", 1, penguin_filter$sex)

# separate out the adelie species
adelie <- penguin_filter[penguin_filter$species=="Adelie",]

# convert feature to numeric
adelie$bill_length_mm <- as.numeric(adelie$bill_length_mm)
adelie$bill_depth_mm <- as.numeric(adelie$bill_depth_mm)
adelie$flipper_length_mm <- as.numeric(adelie$flipper_length_mm)
adelie$body_mass_g <- as.numeric(adelie$body_mass_g)
adelie$sex <- factor(adelie$sex)

# split the data into training and testing datasets 70/30
split <- sample.split(adelie$sex, SplitRatio = 0.7)
train_data <- adelie[split, ]
test_data <- adelie[!split, ]
```

Build Binary Logistic Regression Model

```
# build model
model <- glm(sex ~ bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g, data = train_data, family =
binomial)

# predict on new test data
predicted_values <- predict(model, newdata = test_data, type = "response")

# get the classes of the predict values from the probabilities
predicted_classes <- ifelse(predicted_values > 0.5, 1, 0)
```

```
# calculate the confusion matrix
confusion_matrix <- confusionMatrix(factor(predicted_classes, levels = c(0, 1)), test_data$sex)

# extract the performance metrics
accuracy <- confusion_matrix$overall['Accuracy']
precision <- confusion_matrix$byClass['Sensitivity']
recall <- confusion_matrix$byClass['Specificity']
f1_score <- confusion_matrix$byClass['F1']
```

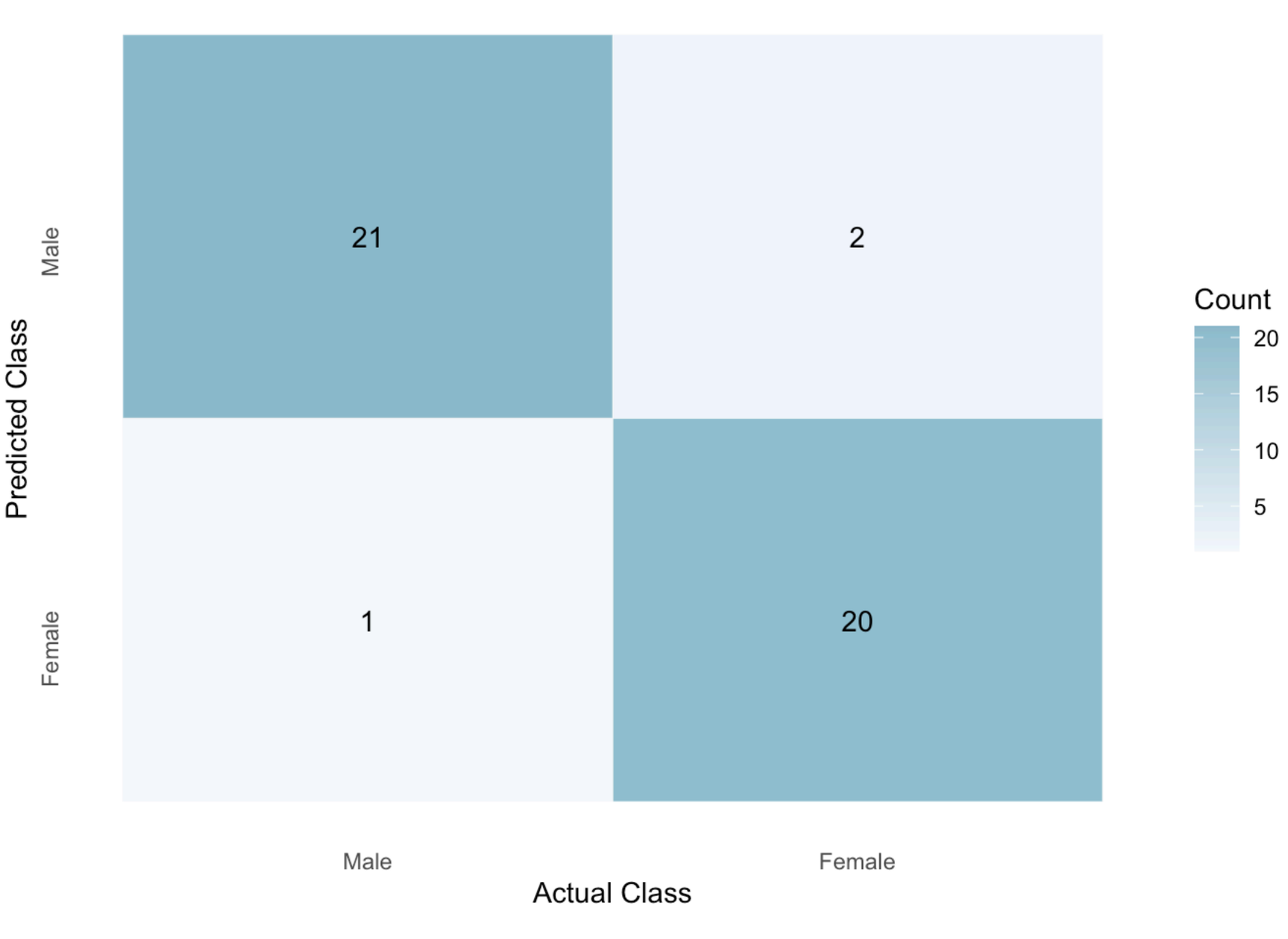
Model Evaluation Metrics

Metric	Value
Accuracy	0.9318182
Precision	0.9090909
Recall	0.9545455
F1 Score	0.9302326

From the accuracy, we can see that the model is overall predicting the right sex around 93% of the time. The precision suggests that among the positive classes (male), around 90% were correctly classified. The recall measures the ratio of predicted positives to actual positives, which is 95%. Lastly, the F1 score combines precision and recall into one score, 93%.

Overall, the model is performing relatively well across a diverse range of evaluation metrics.

```
# parse out relevant confusion matrix values
conf_matrix_values <- as.matrix(confusion_matrix$table)
conf_matrix_df <- as.data.frame(conf_matrix_values)
act <- factor(c("0", "0", "1", "1"))
pred <- factor(c("0", "1", "0", "1"))
Y <- conf_matrix_df$Freq

# create a dataframe
df <- data.frame(act, pred, Y)
df$act <- factor(df$act, levels = c("1", "0"))

# plot confusion matrix
ggplot(data =  df, mapping = aes(x = act, y = pred)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "#f4f8fd", high = "#8dbbcc") +
  theme_minimal() +
  theme(panel.grid = element_blank(),
        axis.text.y = element_text(angle = 90)) +
  scale_x_discrete(labels=c("Male","Female")) +
  scale_y_discrete(labels=c("Female","Male")) +
  labs(x="Actual Class", y="Predicted Class",fill="Count")
```



The confusion matrix visualizes the actual classes compared to the predicted classes. As we can see here, the model is classifying most of the test values correctly with only a few outliers.

Predicting Sex for the Chinstrap Species

Data Processing

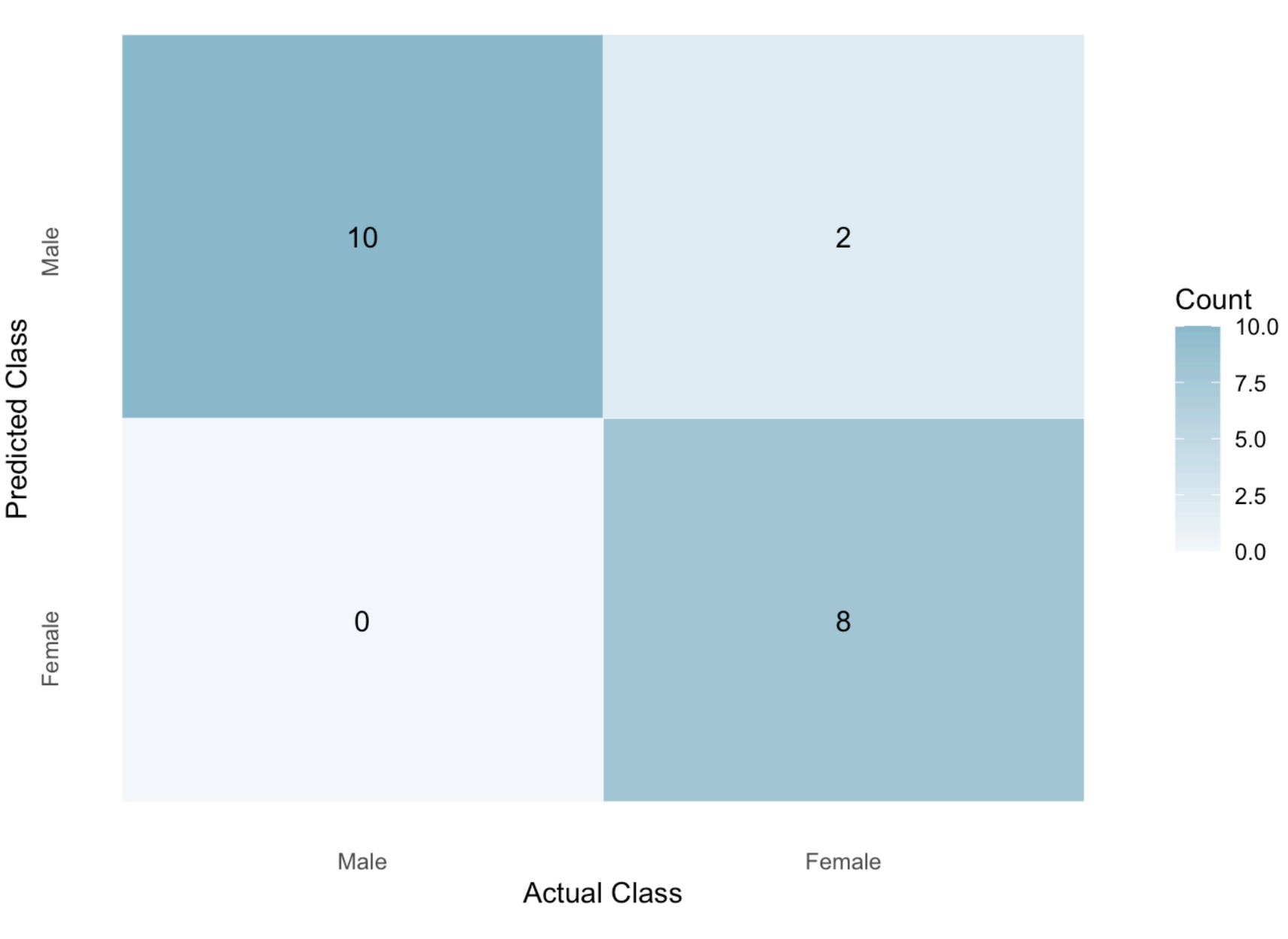
```
# separate out the chinstrap species
chinstrap <- penguin_filter[penguin_filter$species=="Chinstrap",]

# convert features to numeric
chinstrap$bill_length_mm <- as.numeric(chinstrap$bill_length_mm)
chinstrap$bill_depth_mm <- as.numeric(chinstrap$bill_depth_mm)
chinstrap$flipper_length_mm <- as.numeric(chinstrap$flipper_length_mm)
chinstrap$body_mass_g <- as.numeric(chinstrap$body_mass_g)
chinstrap$sex <- factor(chinstrap$sex)

# split the data into training and testing datasets 70/30
split <- sample.split(chinstrap$sex, SplitRatio = 0.7)
train_data <- chinstrap[split, ]
test_data <- chinstrap[!split, ]
```

Build Binary Logistic Regression Model

```
# build model
model <- glm(sex ~ bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g, data = train_data, family =
binomial)

# predict on new test data
predicted_values <- predict(model, newdata = test_data, type = "response")

# get the classes of the predict values from the probabilities
predicted_classes <- ifelse(predicted_values > 0.5, 1, 0)
```

Model Evaluation Metrics

Metric	Value
Accuracy	0.9000000
Precision	0.8000000
Recall	1.0000000
F1 Score	0.8888889



From the evaluation metrics, we can see that the model has a good accuracy of 90%. Interestingly, because all of the males were correctly predicted, the recall is 1. The precision reflects the two females sorted as males.

Based on the metrics and confusion matrix, it seems like the model can accurately classify males better than females for the Chinstrap species. However, we had the least amount of data in this species, which could affect model performance.

Predicting Sex for the Gentoo Species

Data Processing

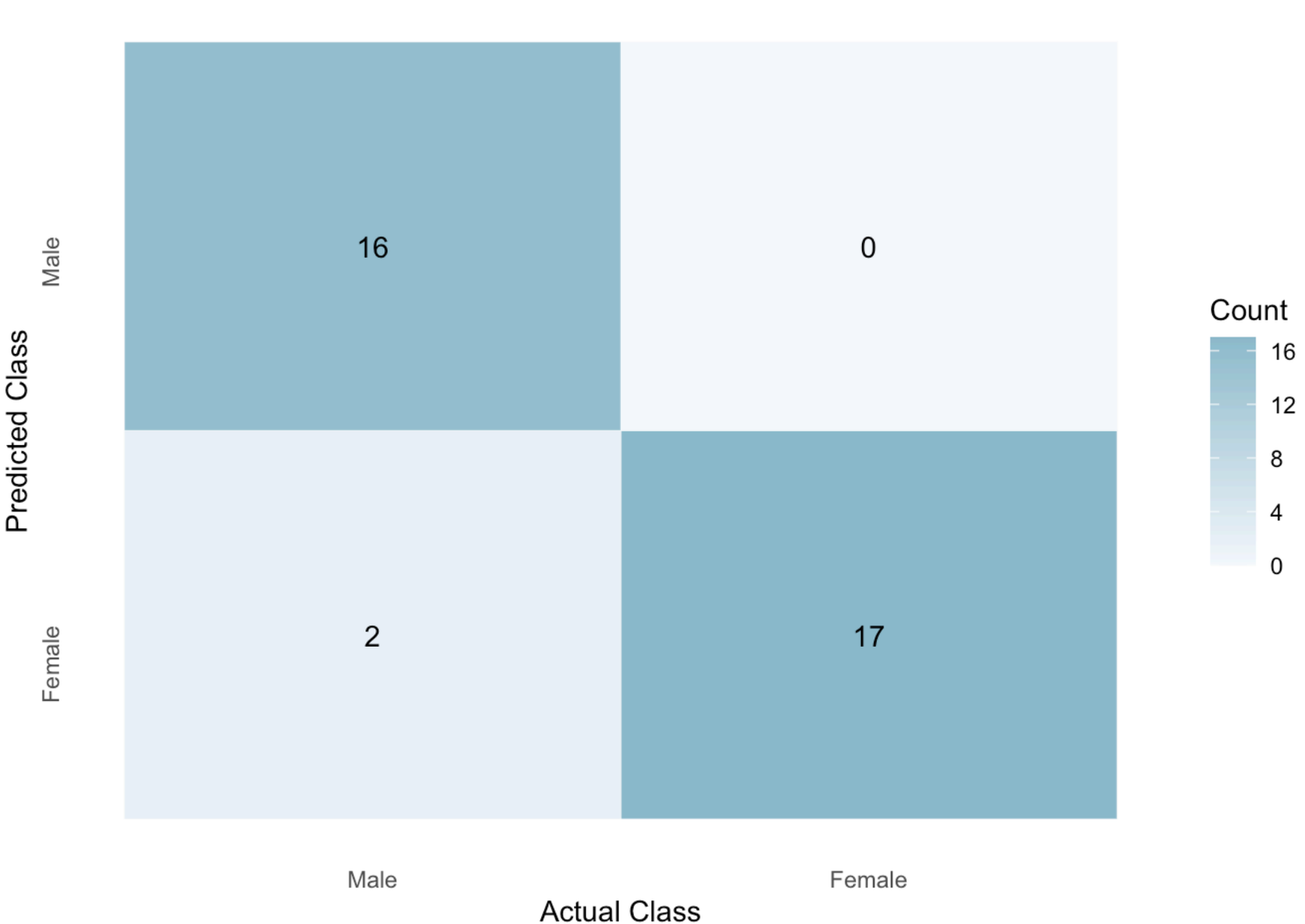
```
# separate out the chinstrap species
gentoo <- penguin_filter[penguin_filter$species=="Gentoo",]

# convert features to numeric
gentoo$bill_length_mm <- as.numeric(gentoo$bill_length_mm)
gentoo$bill_depth_mm <- as.numeric(gentoo$bill_depth_mm)
gentoo$flipper_length_mm <- as.numeric(gentoo$flipper_length_mm)
gentoo$body_mass_g <- as.numeric(gentoo$body_mass_g)
gentoo$sex <- factor(gentoo$sex)

# split the data into training and testing datasets 70/30
split <- sample.split(gentoo$sex, SplitRatio = 0.7)
train_data <- gentoo[split, ]
test_data <- gentoo[!split, ]
```

Build Binary Logistic Regression Model

```
# build model
model <- glm(sex ~ bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g, data = train_data, family =
binomial)

# predict on new test data
predicted_values <- predict(model, newdata = test_data, type = "response")

# get the classes of the predict values from the probabilities
predicted_classes <- ifelse(predicted_values > 0.5, 1, 0)
```

Model Evaluation Metrics

Metric	Value
Accuracy	0.9428571
Precision	1.0000000
Recall	0.8888889
F1 Score	0.9444444

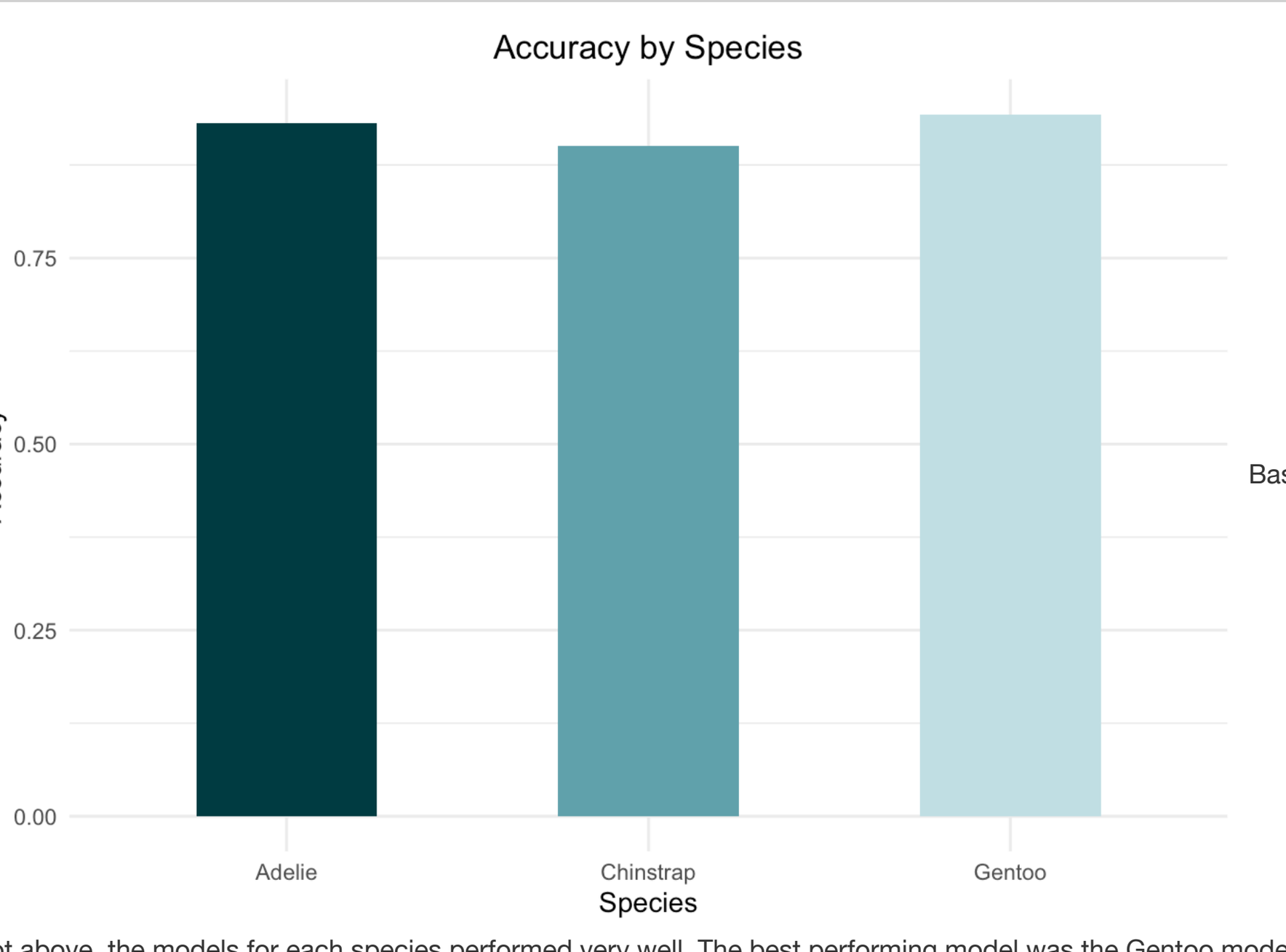


In the metrics, we can see that this time there is a precision of 1. This is reflecting that all of the females were sorted correctly whereas two males were sorted as females. This can be seen in the confusion matrix.

This model also performed well with an accuracy of 94%.

Conclusion

```
# create a dataframe from the accuracies
Accuracy <- c(.9318182, .9, .9428571)
Species <- c("Adelie", "Chinstrap","Gentoo")
acc <- data.frame(Accuracy, Species)

# create barplot
ggplot(acc, aes(x = Species, y = Accuracy, fill = Species)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Accuracy by Species", x = "Species", y = "Accuracy") +
  theme_minimal() +
  scale_fill_manual(values=c("#003b46","#61a4ad","#c0dee5")) +
  theme(plot.title = element_text(hjust = 0.5),legend.position = "none")
```



Based on the accuracy seen in the plot above, the models for each species performed very well. The best performing model was the Gentoo model, but with different testing and training groups these numbers can differ.

Overall, it is possible to successfully predict the sex of a penguin within each of the three species based only on the body mass, bill length, flipper length, and bill depth.