Bioinformatics Final Project

Al Ashir Intisar

12/6/2023

R Markdown

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(ggplot2)
library(tidyr)

#reading in the data scraped from GenBank file using perl

all_data <- read_tsv("~/Academic/Bio_391_perl/final_project/all_data.tsv") %>%
  mutate(file_name = paste(file_name, row_number())) #creating unique id
```

```
## Rows: 217 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (12): file_name, Genus, Species, type, tRNA_starts, tRNA_ends, rRNA_star...
## dbl  (1): mtDNA_size
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#finding statistical difference between Ectotherm and Endotherm complete mtDNA length

model <- aov(mtDNA_size ~ factor(type), data = all_data) #using anova test
summary(model)
```

```
##                Df    Sum Sq   Mean Sq F value   Pr(>F)
## factor(type)    1 2.627e+08 262711718   13.01 0.000385 ***
## Residuals     215 4.341e+09  20191401
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#finding statistical difference between number of codons in coding regions (CDS)
#in mtDNA of Endotherm and Ectotherm

model <- aov(str_count(CDS_string) ~ factor(type), data = all_data) #using anova test
summary(model)
```

```
##                Df    Sum Sq  Mean Sq F value   Pr(>F)
## factor(type)    1  11542630 11542630   11.52 0.000833 ***
## Residuals     199 199459081  1002307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 16 observations deleted due to missingness
```

```
#calculating and creating new variable for the length of rRNA, tRNA, and Coding
#regions' codons

#function to calculate length from the start and end positions of each regions in the #dataset
calculate_total_length <- function(starts, ends) {
  starts_numeric <- as.numeric(str_split(starts, "\\+")[[1]])
  ends_numeric <- as.numeric(str_split(ends, "\\+")[[1]])
  return(sum(ends_numeric - starts_numeric))
}

#creating the variables
all_lengths <- all_data %>%
  mutate(CDS_length = str_count(CDS_string)) %>%
  rowwise() %>%
  mutate(
    tRNA_length = calculate_total_length(c_across(starts_with("tRNA_starts")), c_across(starts_with("tR
    rRNA_length = calculate_total_length(c_across(starts_with("rRNA_starts")), c_across(starts_with("rR
  )
```

```
#finding statistical difference between rRNA length in mtDNA
#of Endotherm and Ectotherm

model <- aov(rRNA_length ~ factor(type), data = all_lengths) #using anova test
summary(model)
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## factor(type)    1   26593   26593   0.912  0.341
## Residuals     182 5305024   29148
## 33 observations deleted due to missingness
```

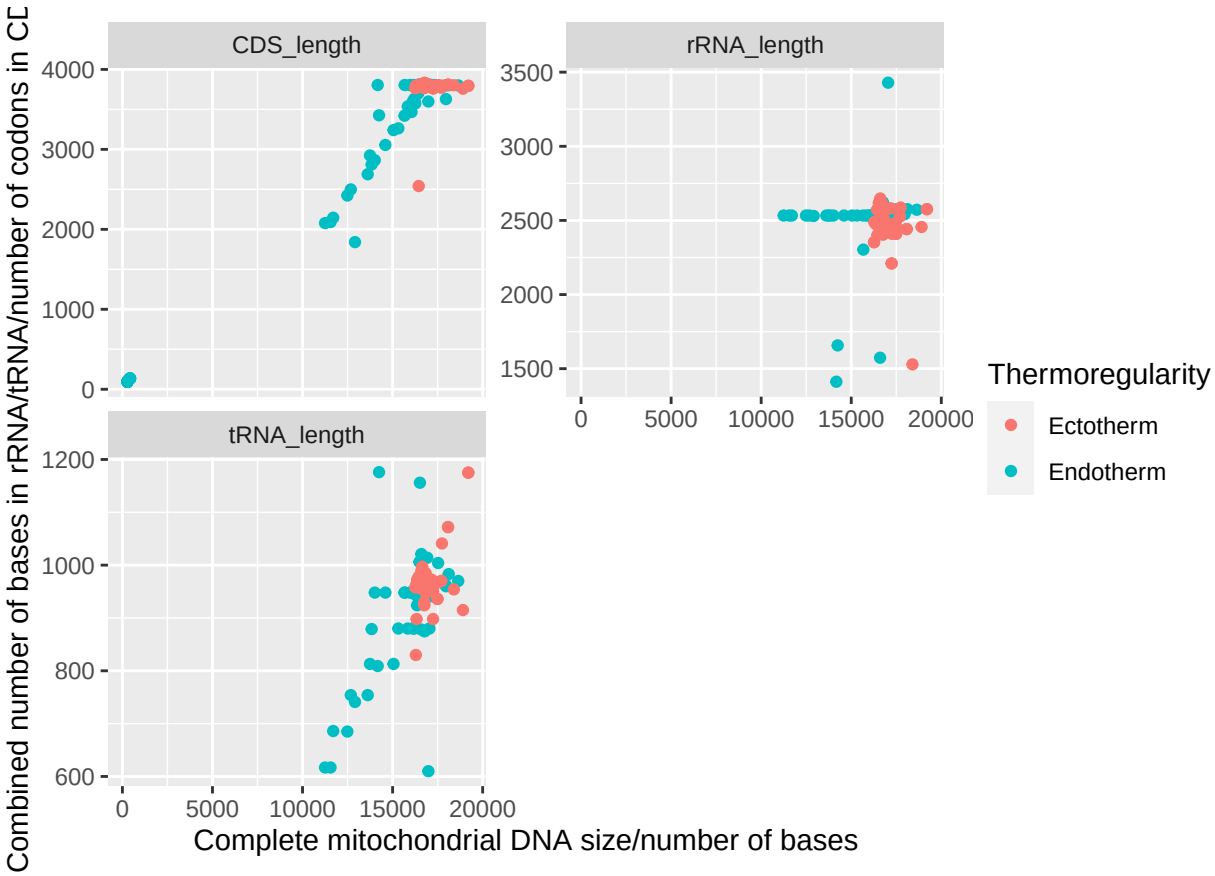
```
#finding statistical difference between tRNA length in mtDNA
#of Endotherm and Ectotherm
```

```
model <- aov(tRNA_length ~ factor(type), data = all_lengths) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## factor(type)    1  80009   80009   15.63 0.00011 ***
## Residuals     182 931600    5119
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 33 observations deleted due to missingness
```

```
#finding the correlation between mtDNA size againts CDS codons, rRNA, and tRNA sizes

#graphing the correlation
all_lengths %>%
  mutate(type = ifelse(type == "cold", "Ectotherm", "Endotherm")) %>%
  pivot_longer(cols = c(CDS_length, tRNA_length, rRNA_length), names_to = "sequence_type", values_to = "
  ggplot(aes(x = mtDNA_size, y = Value, color = type)) +
  geom_point() +
  facet_wrap(~sequence_type, scales = "free_y", ncol = 2) +
  labs(y = "Combined number of bases in rRNA/tRNA/number of codons in CDS", x = "Complete mitochondrial
```

```
## Warning: Removed 82 rows containing missing values (`geom_point()`).
```



```r
#getting rid of the NA values for tRNA and rRNA
cor_data <- all_lengths %>%
  drop_na(CDS_length, tRNA_length, rRNA_length)

#getting the correlation coefficients
correlation1 <- cor(cor_data$mtDNA_size, cor_data$CDS_length)
correlation2 <- cor(cor_data$mtDNA_size, cor_data$rRNA_length)
correlation3 <- cor(cor_data$mtDNA_size, cor_data$tRNA_length)

#printing the correlation coeffcients
cat("Correlation Coefficient mtDNA length ~ Coding region codons:", correlation1, "\n", "Correlation Co
```

```
## Correlation Coefficient mtDNA length ~ Coding region codons: 0.8561144
##  Correlation Coefficient mtDNA length ~ rRNA length: 0.07365597
##  Correlation Coefficient mtDNA length ~ tRNA length: 0.6773914
```

```r
library(tidyr)

#calculating the percent of each amino acid produced from the entire coding region
# of each sample mtDNA


#function to find the frequency of each amino acid produced from the CDS region of the #sequence
process_rows_df <- function(data) {
  rows_df <- data.frame(file_name = character(0))

  for (i in seq_len(nrow(data))) {
    current_row <- data[i, , drop = FALSE]

    name <- current_row[["file_name"]]
    type <- current_row[["type"]]

    amino_acid_freq <- table(strsplit(current_row[["CDS_string"]], NULL))

    amino_acid_freq_df <- as.data.frame(amino_acid_freq) %>%
      pivot_wider(names_from = Var1, values_from = Freq) %>%
      mutate(file_name = name, .before = 1) %>%
      mutate(type = type, .before = 2)

    suppressMessages({
      rows_df <- full_join(rows_df, amino_acid_freq_df)
    })
  }

  return(rows_df)
}

#calculating the number of each amino acid produced from the entire coding region
# of each sample mtDNA and storing it in all_amino_acids

all_amino_acids <-  all_lengths%>%
  drop_na(CDS_string) %>%
  process_rows_df() %>%
```

```
  mutate(type = as.factor(type)) %>%
  rename(plus = `+`) %>%
  full_join(all_lengths, by = "file_name") %>%
  mutate(type = type.x) %>%
  select(-c(type.x, type.y)) %>%
  mutate(type = ifelse(type == "cold", "Ectotherm", "Endotherm"))

#calculating the percent of each amino acid from each CDS
all_data_complete <- all_amino_acids %>%
  drop_na(CDS_length) %>%
  mutate(across(A:X, ~ ./(CDS_length - plus)*100))
```

```
#creating a decision tree model to find out which amino acid are most different in
#determining thermoregulatory (Endotherm, Ectotherm) type


#preparing data for training and testing

model1_data <- all_data_complete %>%
  mutate(type = as.factor(type)) %>%
  select(c(type, A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, X))


#shuffling the dataset for creating a model

set.seed(1234)
shuffled_index1 <- sample(seq_len(nrow(model1_data)))

shuffled_df1 <- model1_data[shuffled_index1, ]

set.seed(2022)
train_index1 <- sample(seq_len(nrow(shuffled_df1)), 0.8 * nrow(shuffled_df1))

train_set1 <- shuffled_df1[train_index1, ]
test_set1 <- shuffled_df1[-train_index1, ]
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.2
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.0.0 --

## v broom        1.0.4     v rsample      1.1.1
## v dials        1.2.0     v tibble       3.2.1
## v infer        1.0.4     v tune         1.1.1
## v modeldata    1.1.0     v workflows    1.1.3
## v parsnip      1.1.0     v workflowsets 1.0.1
## v purrr        1.0.1     v yardstick    1.2.0
## v recipes      1.0.6
```

```
## -- Conflicts ---------------------------------------- tidymodels_conflicts() --
## x purrr::discard()  masks scales::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x dials::prune()    masks rpart::prune()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(dslabs)
```

```
## Warning: package 'dslabs' was built under R version 4.3.2
```

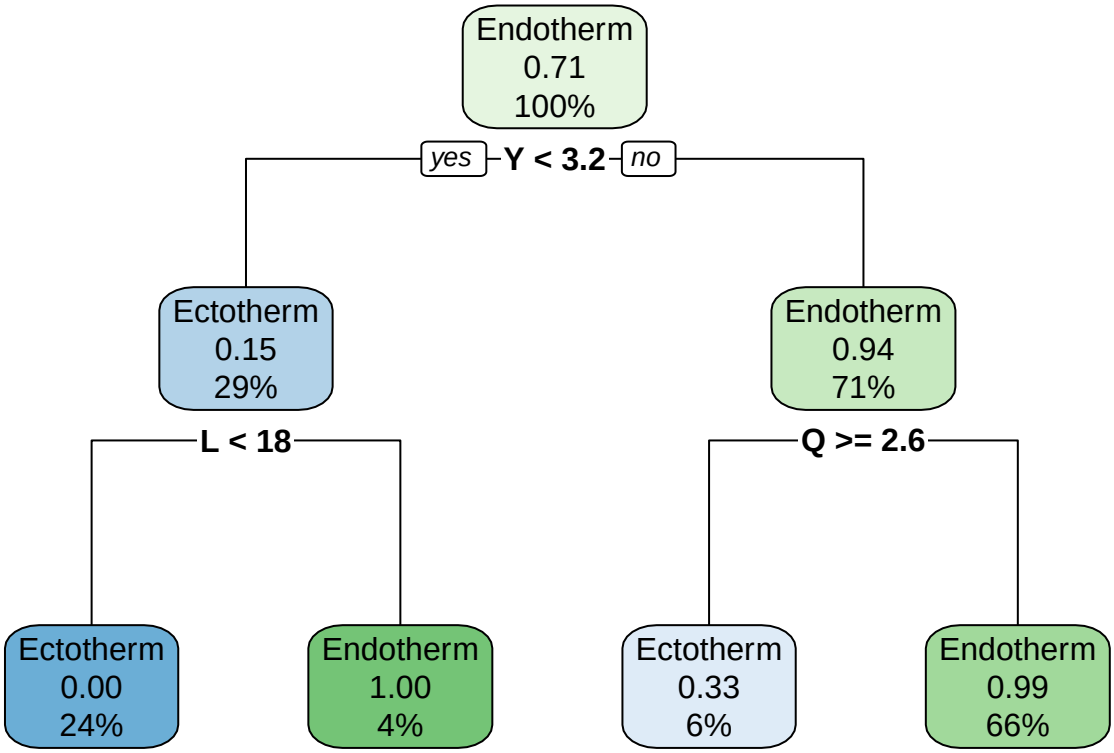
```
#creating a decision tree model

tree_model1 <-
  decision_tree(tree_depth=3) %>%
  set_mode("classification") %>%
  set_engine("rpart")

tree_recipe1 <- recipe(type ~ ., data=train_set1)

tree_wflow1 <- workflow() %>%
    add_recipe(tree_recipe1) %>%
    add_model(tree_model1)

tree_fit1  <- fit(tree_wflow1, train_set1)

#printing the accuracy
augment(tree_fit1, test_set1) %>%
  accuracy(truth=type, estimate=.pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.976
```

```
#printing the confusion matrix
augment(tree_fit1, test_set1) %>%
  conf_mat(truth=type, estimate=.pred_class)
```

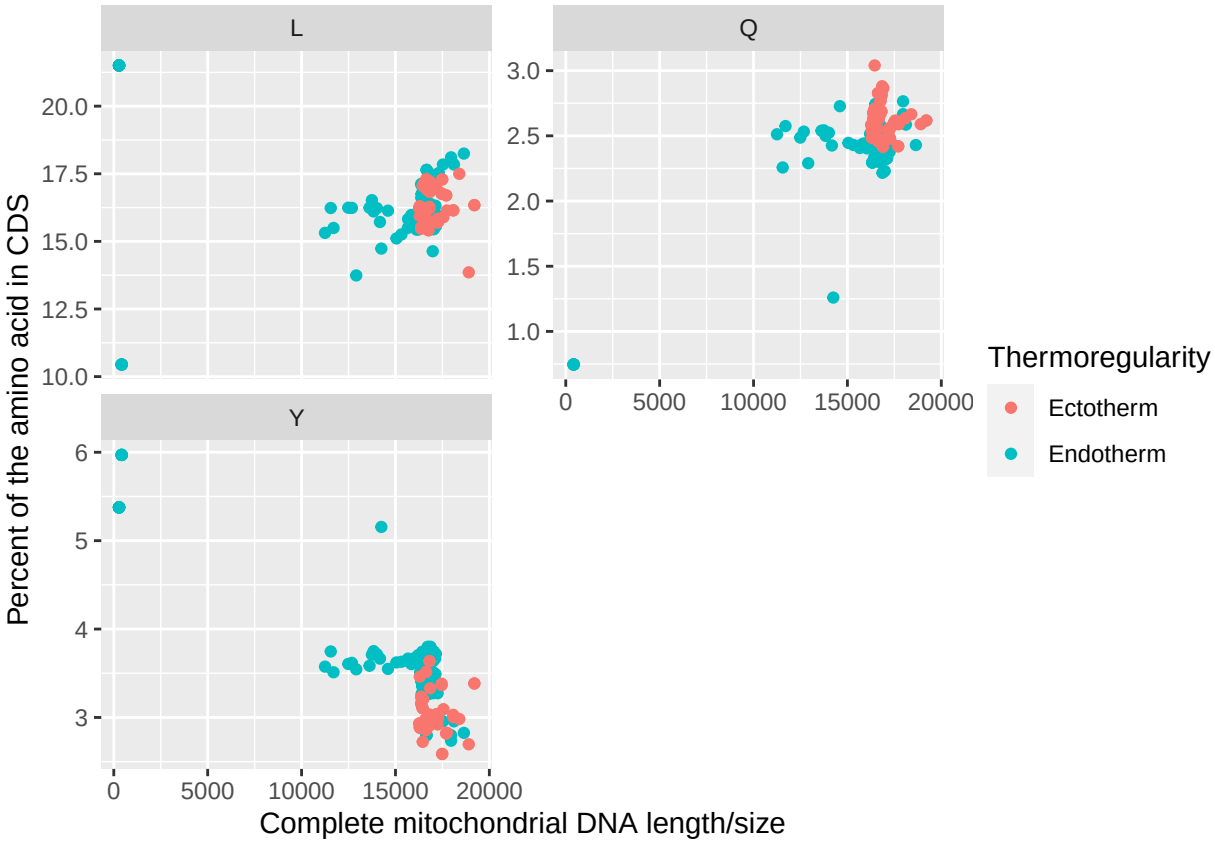
```
##            Truth
## Prediction  Ectotherm Endotherm
##   Ectotherm        14         0
##   Endotherm         1        26
```

```
#printing the decision tree
tree_fit1 %>%
  extract_fit_engine() %>%
  rpart.plot(roundint=FALSE)
```




```
#printing out the percent of amino acids in the decisoin tree against mtDNA size
#to observe their distribution

all_data_complete %>%
  pivot_longer(cols = c(Q, Y, L), names_to = "Amino_Acid", values_to = "Value") %>%
  ggplot(aes(x = mtDNA_size, y = Value, color = type)) +
  geom_point() +
  facet_wrap(~Amino_Acid, scales = "free_y", ncol = 2) +
  labs(y = "Percent of the amino acid in CDS", x = "Complete mitochondrial DNA length/size", color = "T
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```




```
#finding statistical difference between Tyrosine(Y) percent in CDS
#of Endotherm and Ectotherm

model <- aov(Y ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## factor(type)   1  23.13  23.131   64.98 6.9e-14 ***
## Residuals    199  70.84   0.356
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#finding statistical difference between Glutamine(Q) percent in CDS
#of Endotherm and Ectotherm

model <- aov(Q ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## factor(type)   1  3.055  3.0553   33.64 2.77e-08 ***
## Residuals    188 17.077  0.0908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 11 observations deleted due to missingness
```

```r
#finding statistical difference between Leucine(L) percent in CDS
#of Endotherm and Ectotherm

model <- aov(L ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## factor(type)    1    0.2  0.2169   0.076  0.783
## Residuals     199  567.9  2.8536
```

```r
#creating decision tree including the tRNA, rRNA, CDS length

#preparing data for training and testing

model2_data <- all_data_complete %>%
  mutate(type = as.factor(type)) %>%
  select(c(type, A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, X, rRNA_length, tRNA_lengt


#shuffling the dataset for creating a model

set.seed(12345)
shuffled_index2 <- sample(seq_len(nrow(model2_data)))

shuffled_df2 <- model2_data[shuffled_index2, ]

set.seed(2022)
train_index2 <- sample(seq_len(nrow(shuffled_df2)), 0.8 * nrow(shuffled_df2))

train_set2 <- shuffled_df2[train_index2, ]
test_set2 <- shuffled_df2[-train_index2, ]


library(rpart)
library(rpart.plot)
library(tidymodels)
library(dslabs)

#creating a decision tree model

tree_model2 <-
  decision_tree(tree_depth=3) %>%
  set_mode("classification") %>%
  set_engine("rpart")

tree_recipe2 <- recipe(type ~ ., data=train_set2)

tree_wflow2 <- workflow() %>%
    add_recipe(tree_recipe2) %>%
    add_model(tree_model2)

tree_fit2  <- fit(tree_wflow2, train_set2)
```

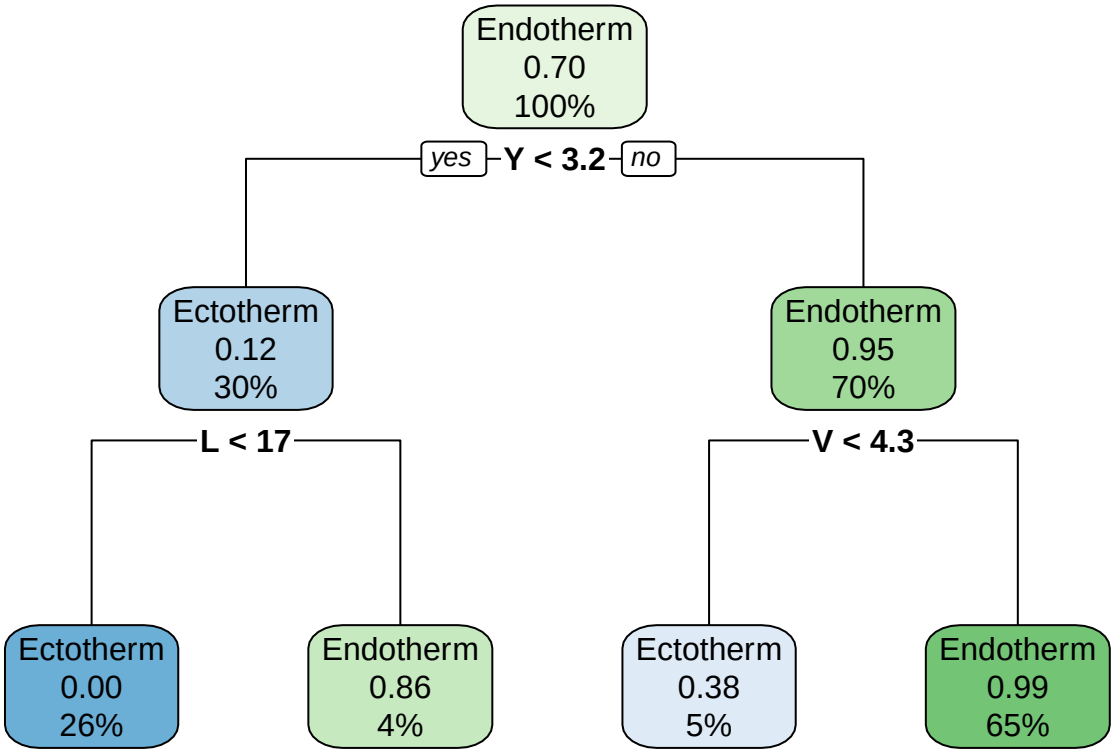
```
#printing the accuracy
augment(tree_fit2, test_set2) %>%
  accuracy(truth=type, estimate=.pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.951
```

```
#printing the confusion matrix
augment(tree_fit2, test_set2) %>%
  conf_mat(truth=type, estimate=.pred_class)
```

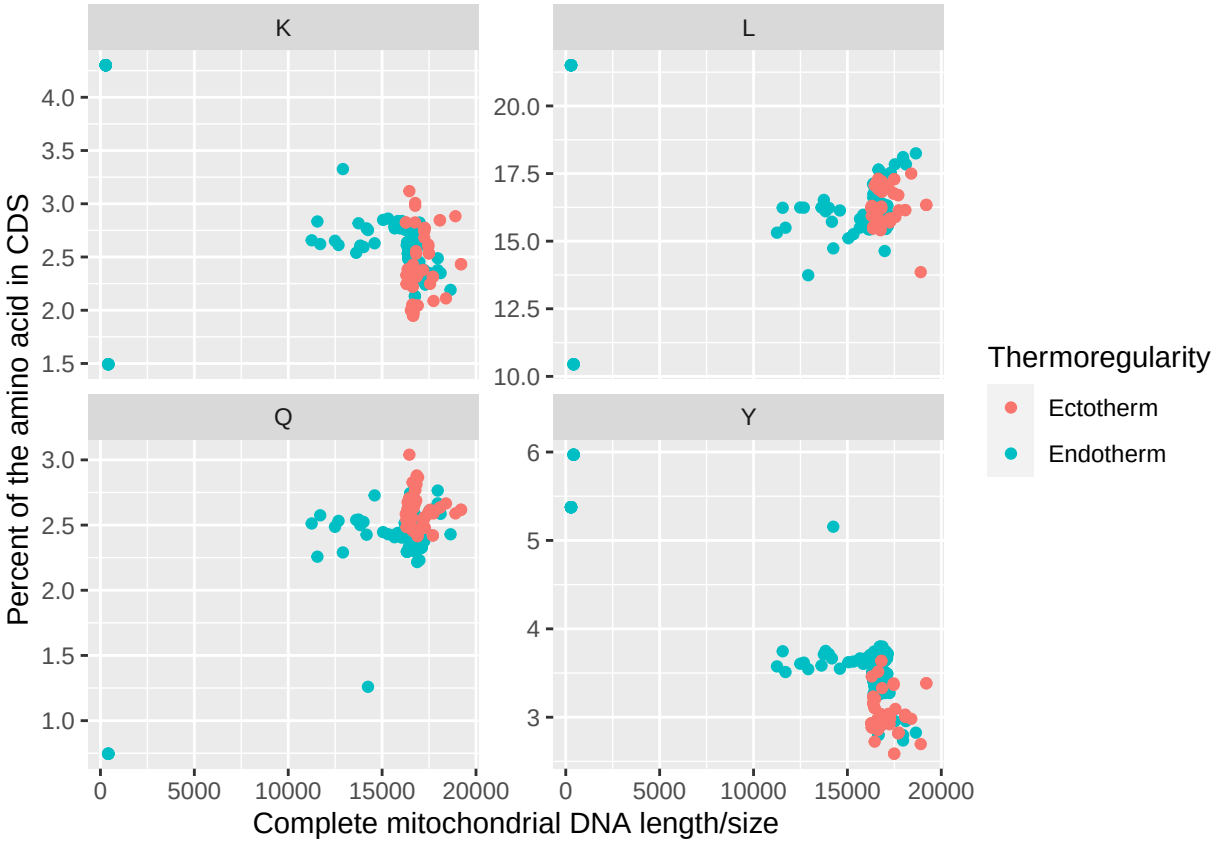
```
##             Truth
## Prediction  Ectotherm Endotherm
##   Ectotherm        13         2
##   Endotherm         0        26
```

```
#printing the decision tree
tree_fit2 %>%
  extract_fit_engine() %>%
  rpart.plot(roundint=FALSE)
```



```
all_data_complete %>%
  pivot_longer(cols = c(Q, Y, L, K), names_to = "Amino_Acid", values_to = "Value") %>%
  ggplot(aes(x = mtDNA_size, y = Value, color = type)) +
  geom_point() +
  facet_wrap(~Amino_Acid, scales = "free_y", ncol = 2) +
  labs(y = "Percent of the amino acid in CDS", x = "Complete mitochondrial DNA length/size", color = "T
```

```
## Warning: Removed 11 rows containing missing values (‘geom_point()‘).
```




```
#finding statistical difference between Tyrosine(Y) percent in CDS
#of Endotherm and Ectotherm

model <- aov(Y ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## factor(type)    1  23.13  23.131   64.98 6.9e-14 ***
## Residuals     199  70.84   0.356
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
#finding statistical difference between Glutamine(Q) percent in CDS
#of Endotherm and Ectotherm
```

```r
model <- aov(Q ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## factor(type)    1  3.055  3.0553   33.64 2.77e-08 ***
## Residuals     188 17.077  0.0908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 11 observations deleted due to missingness
```

```r
#finding statistical difference between Leucine(L) percent in CDS
#of Endotherm and Ectotherm

model <- aov(V ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## factor(type)    1   7.07   7.067    16.9 5.76e-05 ***
## Residuals     199  83.21   0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#finding statistical difference between Valine(V) percent in CDS
#of Endotherm and Ectotherm

model <- aov(V ~ factor(type), data = all_data_complete) #using anova test
summary(model)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## factor(type)    1   7.07   7.067    16.9 5.76e-05 ***
## Residuals     199  83.21   0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

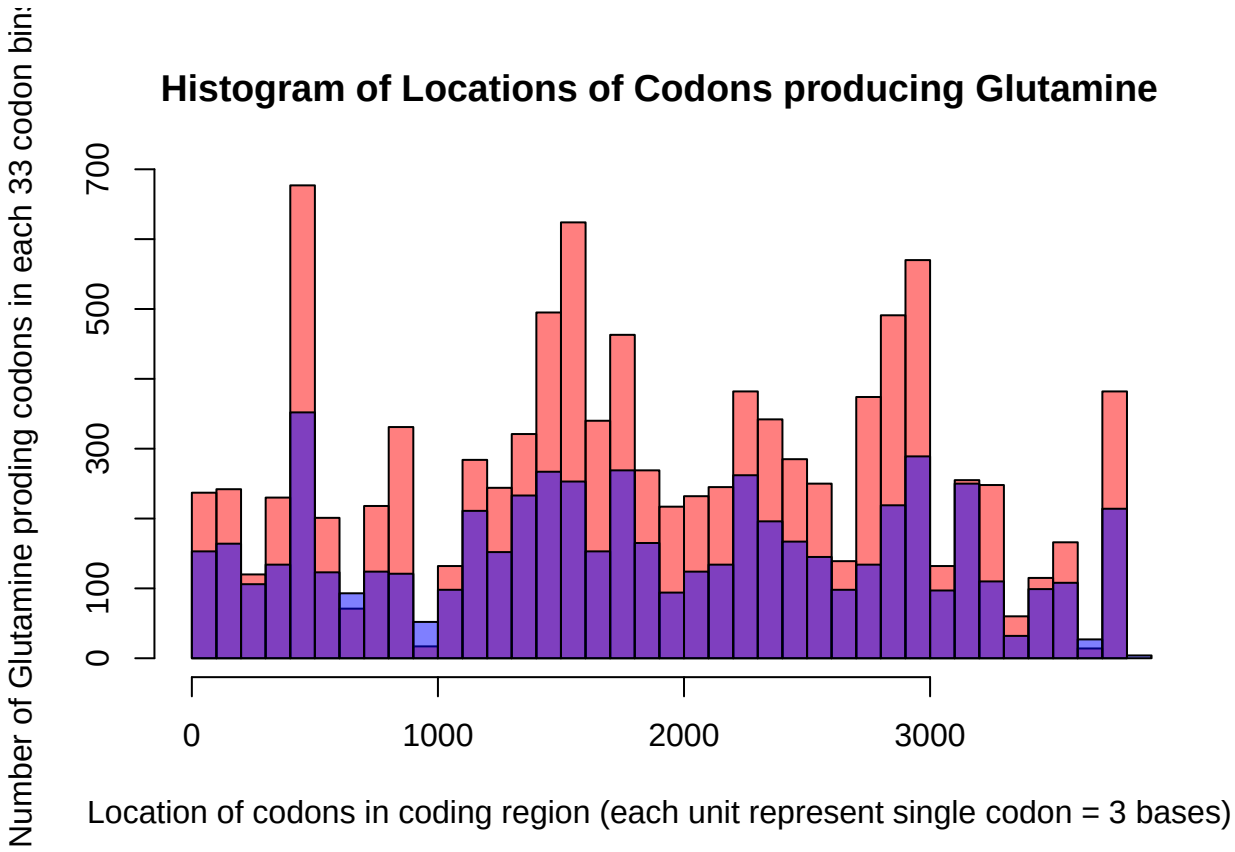
```r
#write.csv(all_data_complete, "~/Academic/Bio_391_perl/final_project/Project_data.csv")
```

```r
#function to get the locations of each codons producing Glutamine, Tyrosine, and Valine


find_indices_in_dataset <- function(dataset) {
  for (letter in c("Y", "V", "Q")) {
    indices_list <- list()

    for (row_index in 1:nrow(dataset)) {
      cds_string <- dataset$CDS_string[row_index]
      indices <- which(strsplit(cds_string, NULL)[[1]] == letter)
      indices_list[[row_index]] <- ifelse(length(indices) > 0, paste(indices, collapse = ","), NA)
    }

    dataset[[paste0("Indices_", letter)]] <- indices_list
```

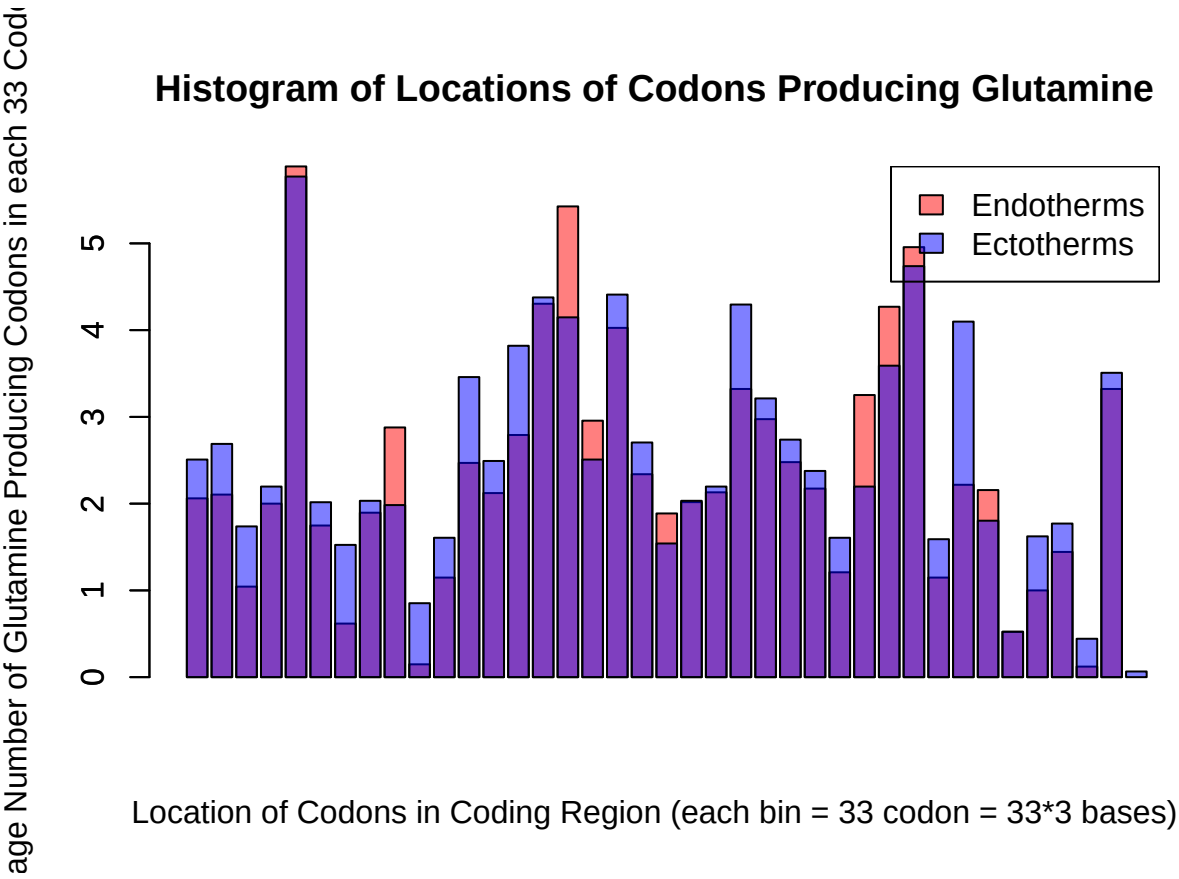
```
  }

  return(dataset)
}



all_locations <- find_indices_in_dataset(all_data_complete)
```

```
#seperating ectotherms and endotherms

ectotherm_locations <- all_locations %>%
  filter(mtDNA_size > 14000) %>%
  select(type, Indices_Q, Indices_Y, Indices_V) %>%
  filter(type == "Ectotherm")

endotherm_locations <- all_locations %>%
  filter(mtDNA_size > 14000) %>%
  select(type, Indices_Q, Indices_Y, Indices_V) %>%
  filter(type == "Endotherm")
```

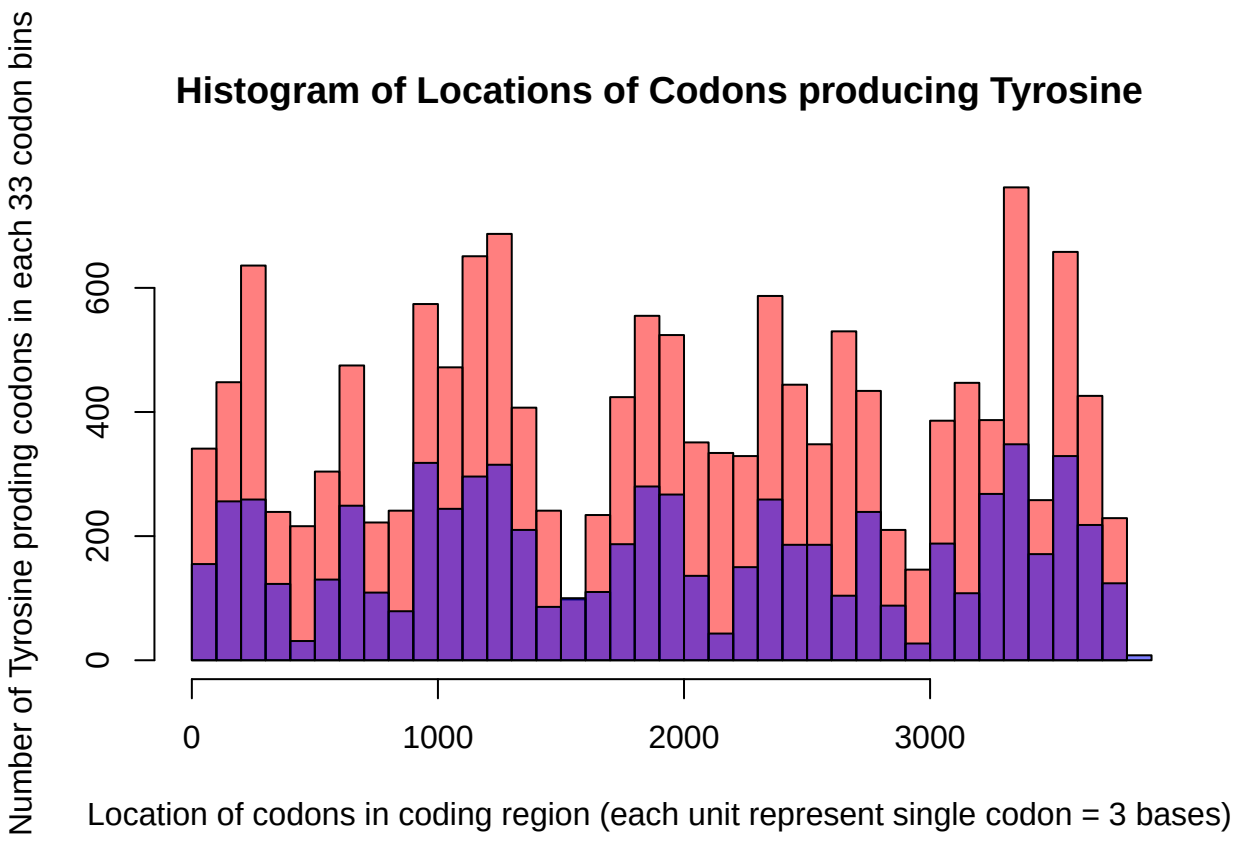
```
#extracting the locations of Glutamine producing codons within the in the coding region
endotherm_hist_dataQ <- numeric()

for (i in seq_along(endotherm_locations$Indices_Q)) {
  temp <- endotherm_locations$Indices_Q[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  endotherm_hist_dataQ <- c(endotherm_hist_dataQ, list_numeric)

}

ectotherm_hist_dataQ <- numeric()

for (i in seq_along(ectotherm_locations$Indices_Q)) {
  temp <- ectotherm_locations$Indices_Q[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  ectotherm_hist_dataQ <- c(ectotherm_hist_dataQ, list_numeric)

}



hist1 <- hist(endotherm_hist_dataQ, breaks = 33, col = rgb(1,0,0,0.5), main = "Histogram of Locations o
hist1$counts <- hist1$counts/115 #sample size 115

hist2 <- hist(ectotherm_hist_dataQ, breaks = 33, add = TRUE, col = rgb(0,0,1,0.5))
```



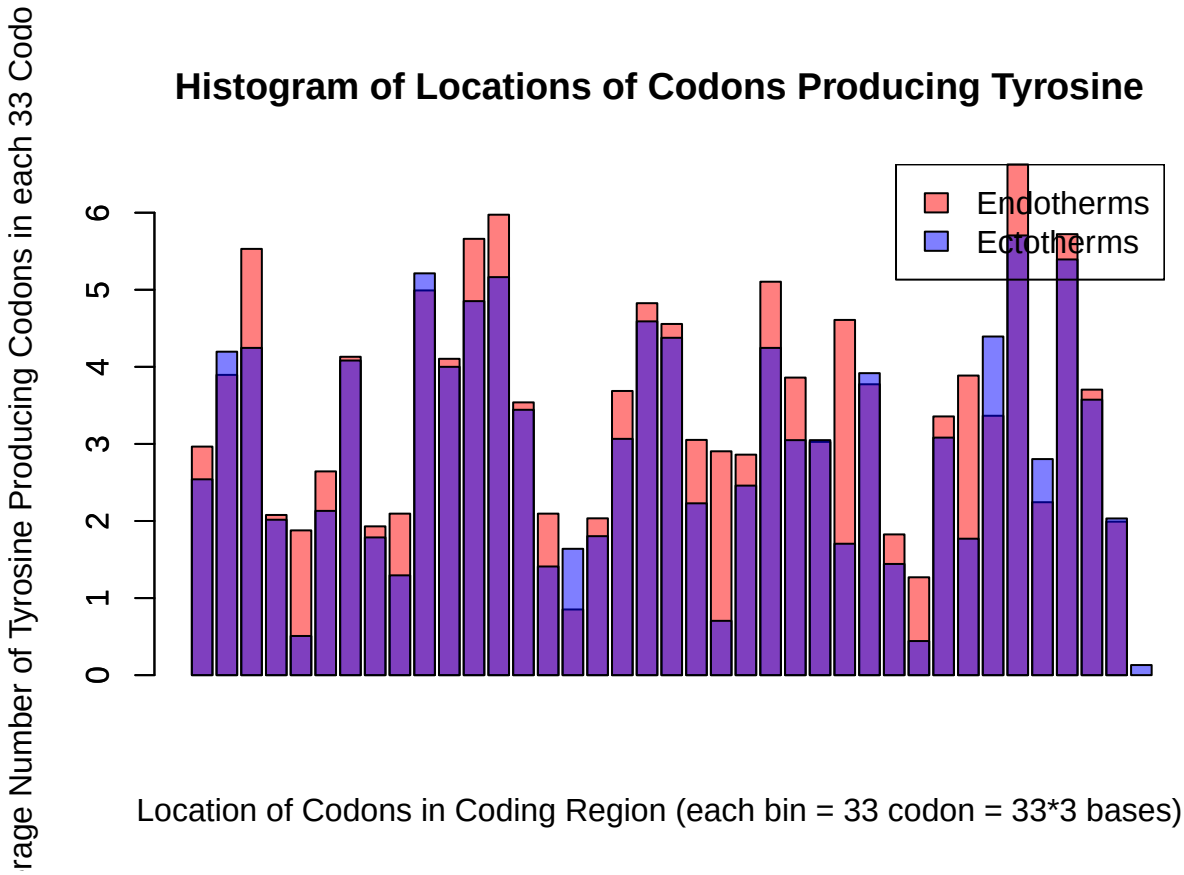

```
hist2$counts <- hist2$counts/61 #sample size 61


barplot(hist1$counts, col = rgb(1, 0, 0, 0.5), main = "Histogram of Locations of Codons Producing Glutar
        xlab = "Location of Codons in Coding Region (each bin = 33 codon = 33*3 bases)",
        ylab = "Average Number of Glutamine Producing Codons in each 33 Codon Bins",
        ylim = c(0, max(c(max(hist1$counts), max(hist2$counts)))))
barplot(hist2$counts, col = rgb(0, 0, 1, 0.5), add = TRUE)
legend("topright", legend = c("Endotherms", "Ectotherms"), fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5)
```



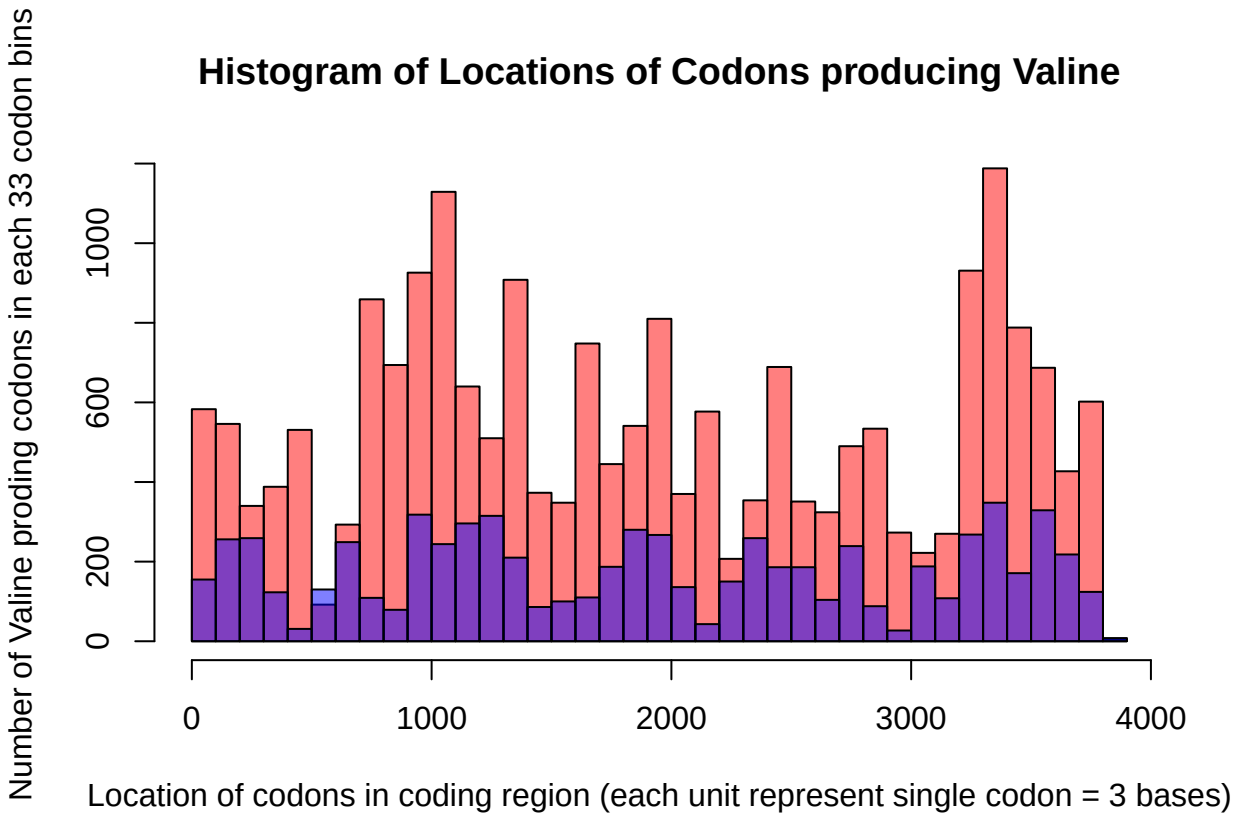

```r
#extracting the locations of Tyrosine producing codons within the in the coding region
endotherm_hist_dataY <- numeric()

for (i in seq_along(endotherm_locations$Indices_Y)) {
  temp <- endotherm_locations$Indices_Y[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  endotherm_hist_dataY <- c(endotherm_hist_dataY, list_numeric)

}

ectotherm_hist_dataY <- numeric()

for (i in seq_along(ectotherm_locations$Indices_Y)) {
  temp <- ectotherm_locations$Indices_Y[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  ectotherm_hist_dataY <- c(ectotherm_hist_dataY, list_numeric)

}

#creating the histogram for tyrosine
hist1 <- hist(endotherm_hist_dataY, breaks = 33, col = rgb(1,0,0,0.5), main = "Histogram of Locations o
hist1$counts <- hist1$counts/115

hist2 <- hist(ectotherm_hist_dataY, breaks = 33, add = TRUE, col = rgb(0,0,1,0.5))
```



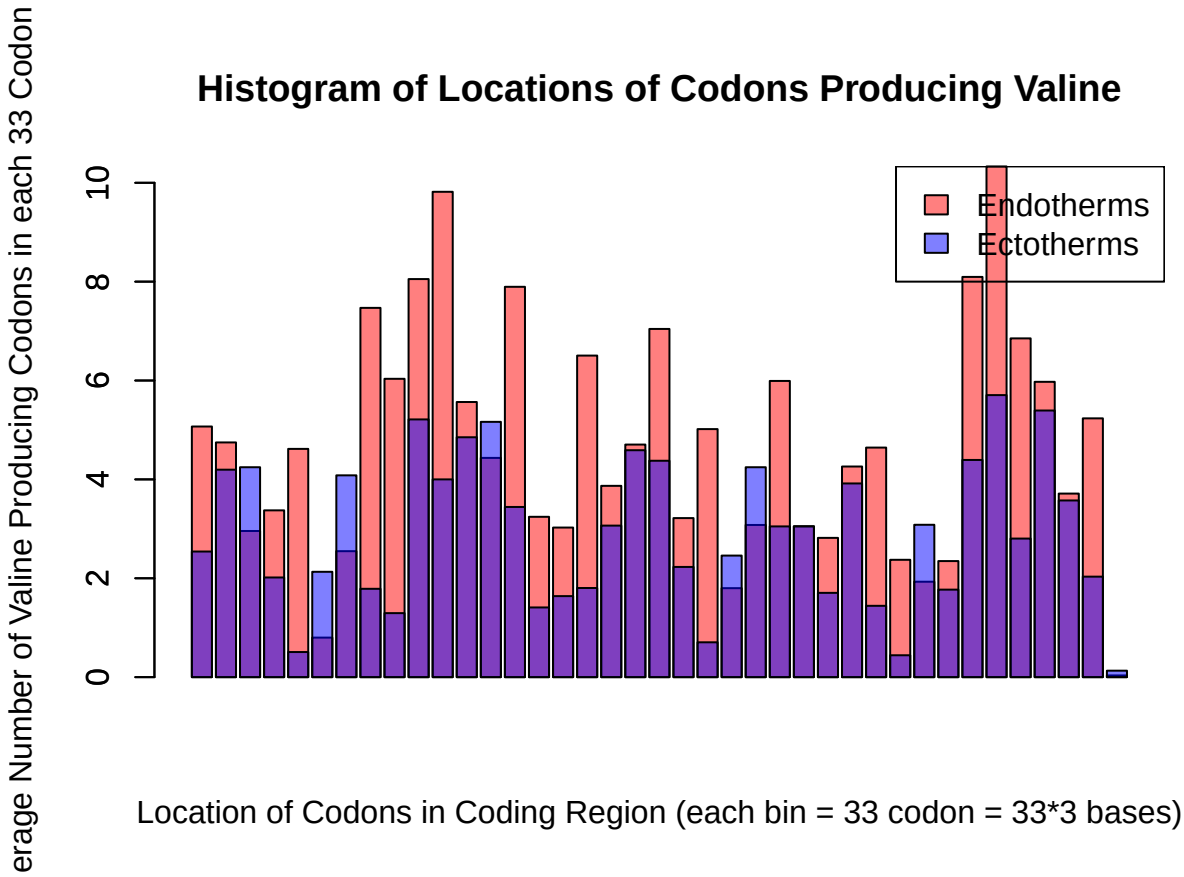

```
hist2$counts <- hist2$counts/61


barplot(hist1$counts, col = rgb(1, 0, 0, 0.5), main = "Histogram of Locations of Codons Producing Tyros
        xlab = "Location of Codons in Coding Region (each bin = 33 codon = 33*3 bases)",
        ylab = "Average Number of Tyrosine Producing Codons in each 33 Codon Bins",
        ylim = c(0, max(c(max(hist1$counts), max(hist2$counts)))))
barplot(hist2$counts, col = rgb(0, 0, 1, 0.5), add = TRUE)
legend("topright", legend = c("Endotherms", "Ectotherms"), fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5
```




```
#extracting the locations of valine producing codons within the in the coding region
endotherm_hist_dataV <- numeric()

for (i in seq_along(endotherm_locations$Indices_V)) {
  temp <- endotherm_locations$Indices_V[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  endotherm_hist_dataV <- c(endotherm_hist_dataV, list_numeric)

}

ectotherm_hist_dataV <- numeric()

for (i in seq_along(ectotherm_locations$Indices_V)) {
  temp <- ectotherm_locations$Indices_V[[i]]
  list_numeric <- as.numeric(unlist(strsplit(temp, ",")))
  ectotherm_hist_dataV <- c(ectotherm_hist_dataV, list_numeric)

}

#creating the histogram for valine
hist1 <- hist(endotherm_hist_dataV, breaks = 33, col = rgb(1,0,0,0.5), main = "Histogram of Locations o
hist1$counts <- hist1$counts/115

hist2 <- hist(ectotherm_hist_dataY, breaks = 33, add = TRUE, col = rgb(0,0,1,0.5))
```




```
hist2$counts <- hist2$counts/61


barplot(hist1$counts, col = rgb(1, 0, 0, 0.5), main = "Histogram of Locations of Codons Producing Valine
        xlab = "Location of Codons in Coding Region (each bin = 33 codon = 33*3 bases)",
        ylab = "Average Number of Valine Producing Codons in each 33 Codon Bins",
        ylim = c(0, max(c(max(hist1$counts), max(hist2$counts)))))
barplot(hist2$counts, col = rgb(0, 0, 1, 0.5), add = TRUE)
legend("topright", legend = c("Endotherms", "Ectotherms"), fill = c(rgb(1, 0, 0, 0.5), rgb(0, 0, 1, 0.5)
```


Histogram of Locations of Codons Producing Valine
erage Number of Valine Producing Codons in each 33 Codon
0
2
4
6
8
10
Endotherms
Ectotherms
Location of Codons in Coding Region (each bin = 33 codon = 33*3 bases)