Fruit Flies Logistic Regression

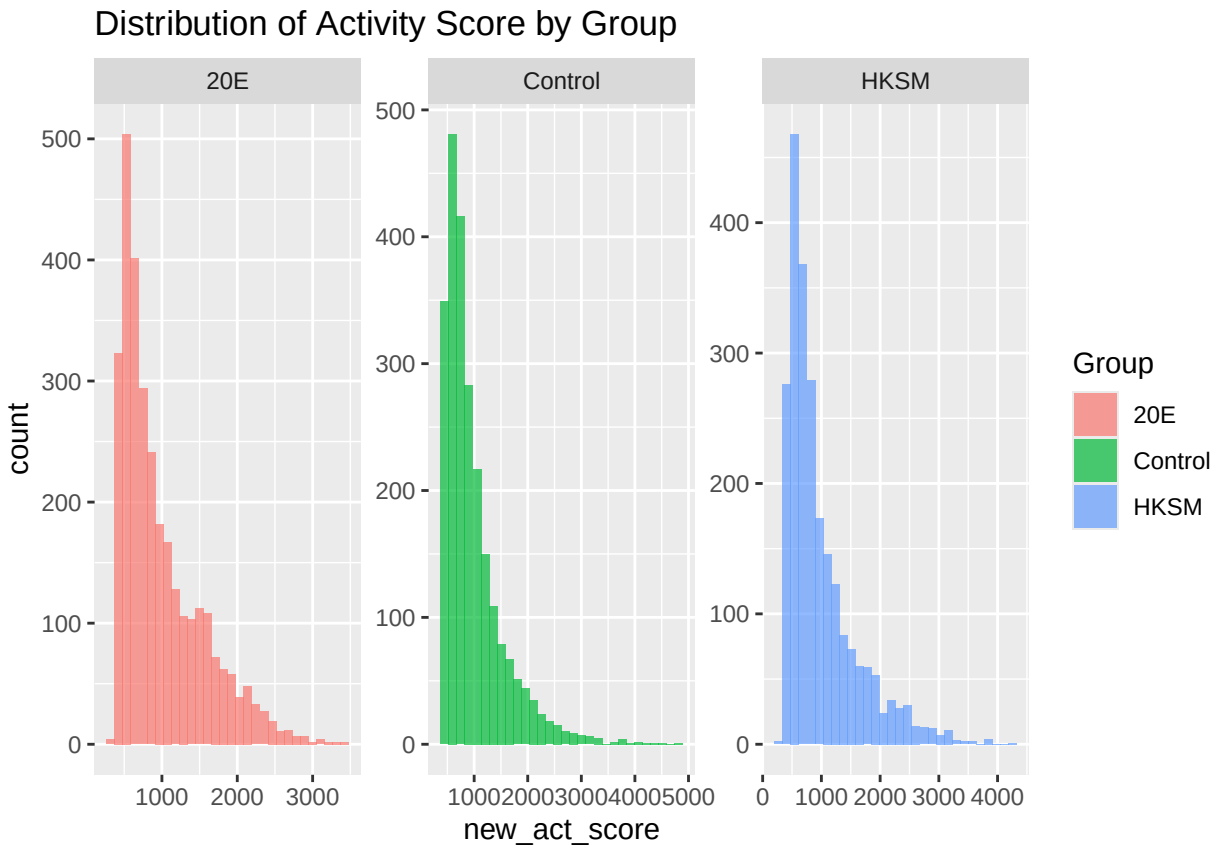
```
## Warning: package 'knitr' was built under R version 4.4.3

## Warning: package 'ggplot2' was built under R version 4.4.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

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

##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
##
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## corrplot 0.95 loaded
```

```r
# load data
control <- read.csv("EG_Con_col_03_11.csv") %>% mutate(Group = "Control")
twentye <- read.csv("EG_20E_col_03_11.csv") %>% mutate(Group = "20E")
hksm <- read.csv("EG_HKSM_20E_col_03_11.csv") %>% mutate(Group = "HKSM")

# combined data
all_data <- bind_rows(control, twentye, hksm)
```

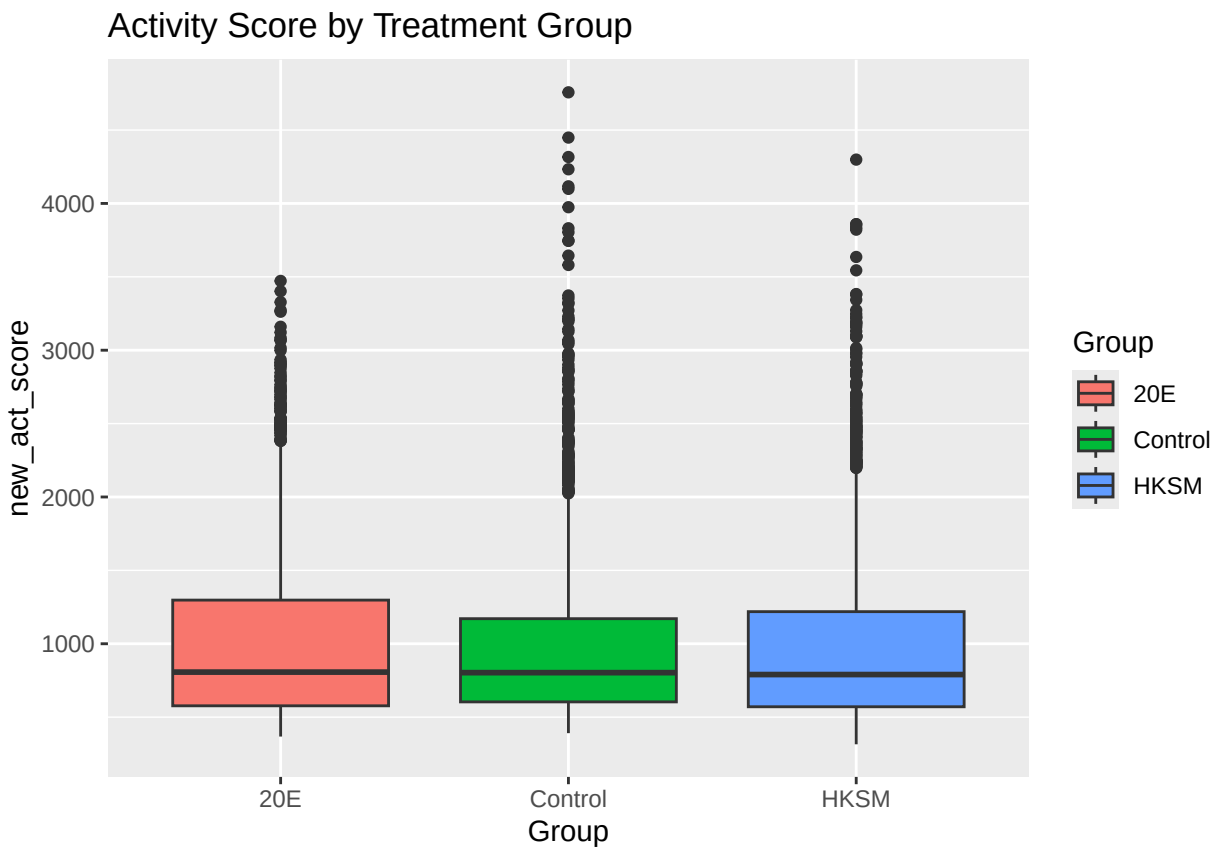
EDA

Here is a small EDA of the distribution of the activity score.

```
# histogram of activity score by group
ggplot(all_data, aes(x = new_act_score, fill = Group)) +
  geom_histogram(bins = 30, alpha = 0.7) +
  facet_wrap(~Group, scales = "free") +
  labs(title = "Distribution of Activity Score by Group")
```



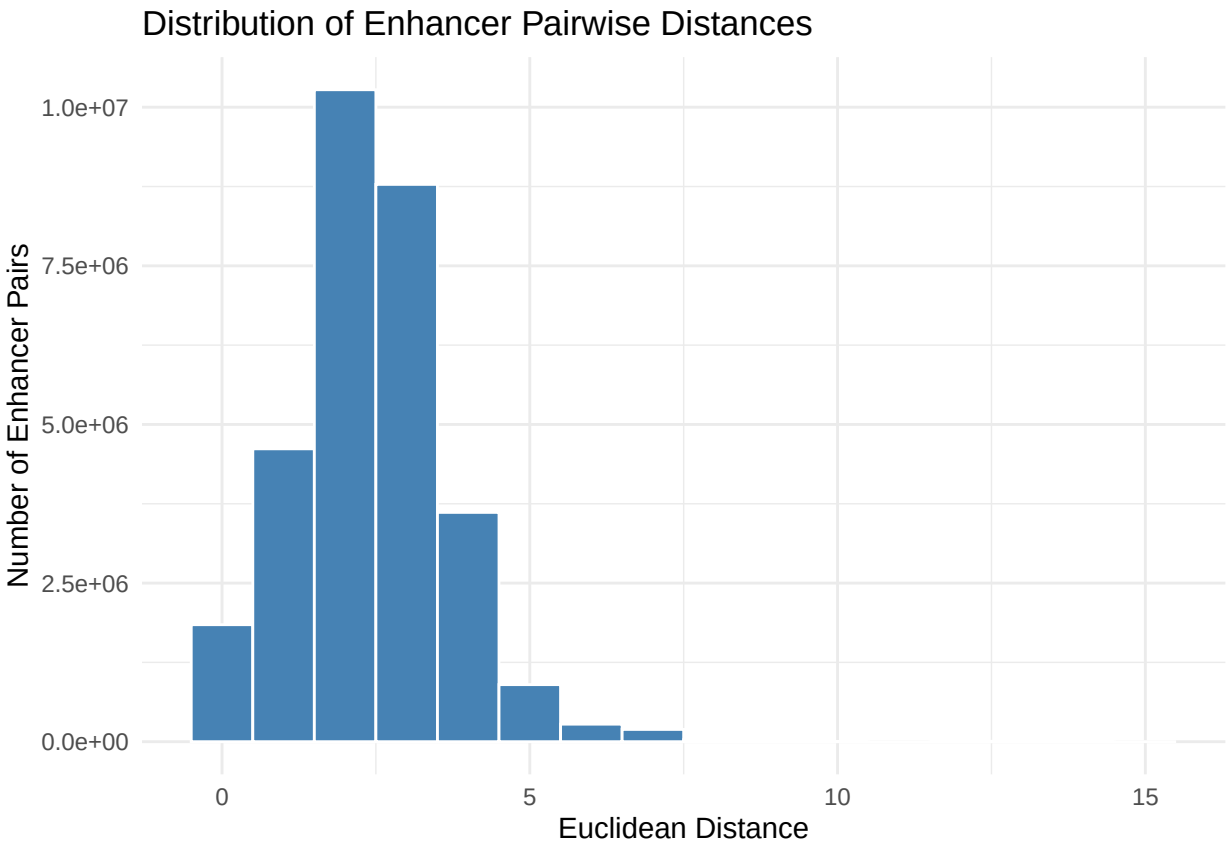

```
# box plot of the three groups
ggplot(all_data, aes(x = Group, y = new_act_score, fill = Group)) +
  geom_boxplot() +
  labs(title = "Activity Score by Treatment Group")
```



From the histogram, the activity scores for all of the groups are right skewed. The above box plot shows that all three groups have similar median activity scores, where the control group has high outliers, while HKSM shows more spread.

```
# motif columns and activity score column
motif_cols <- 8:28
activity_col <- 30

# use median as the threshold for evaluating activity score
threshold <- median(all_data[[activity_col]])
all_data$activity_class <- ifelse(all_data[[activity_col]] > threshold, "High", "Low")
all_data$activity_class <- as.factor(all_data$activity_class)

# cleaned data
cleaned_data <- all_data[, c(motif_cols, which(names(all_data) == "activity_class"))]

# euclidean distance approach
motif_matrix <- as.matrix(cleaned_data[, 1:21])
dist_matrix <- as.matrix(dist(motif_matrix, method = "euclidean"))

upper_triangle <- dist_matrix[upper.tri(dist_matrix)]

ggplot(data.frame(dist = upper_triangle), aes(x = dist)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "white") +
  labs(title = "Distribution of Enhancer Pairwise Distances",
       x = "Euclidean Distance",
```

```
       y = "Number of Enhancer Pairs") +
  theme_minimal()
```




```
library(stringr)

all_data <- all_data %>%
  mutate(chr = str_extract(Enhancer, "^[^:]+"))

cleaned_data$seq <- all_data$chr
```

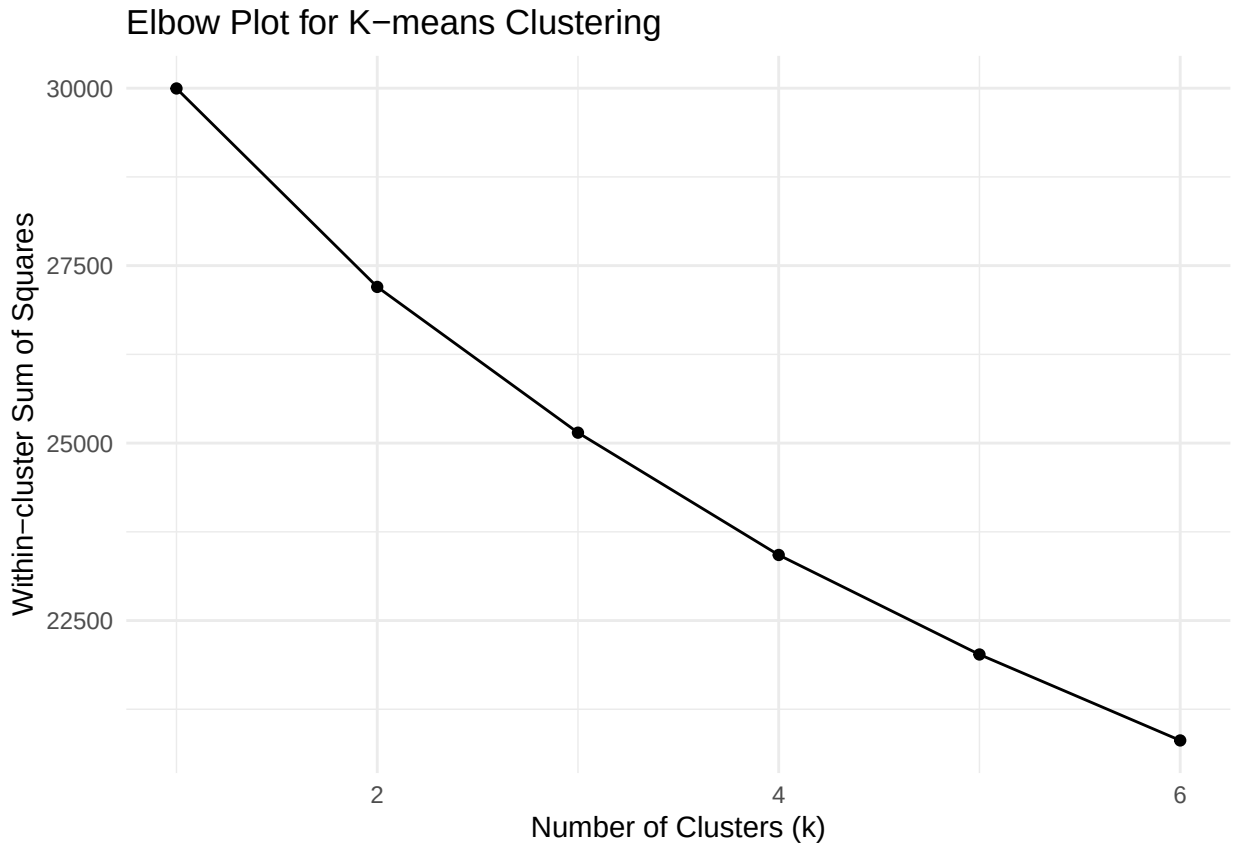
The histogram above shows that the distribution of the pairwise distances of enhancers. Majority of enhancer pairs have small Euclidean distances, with a peak around 2–3, indicating high similarity or clustering in feature space. Only a small proportion of pairs are far apart (less similar enhancer combinations).

Clustering

The following code clusters the enhancers based on the 21 Transcription Factors Motifs using Euclidean distance. The method used is k-means clustering, where it is a way to group data into K clusters based on similarity. In this case, similarity of the TFs are being considered.

```
# Calculate Within cluster sum of squares
set.seed(679)
wss <- map_dbl(1:6, function(k) {
  kmeans(motif_matrix, centers = k, nstart = 10)$tot.withinss
```

```
})

# Elbow Plot
elbow_df <- data.frame(k = 1:6, wss = wss)
ggplot(elbow_df, aes(x = k, y = wss)) +
  geom_line() + geom_point() +
  labs(title = "Elbow Plot for K-means Clustering",
       x = "Number of Clusters (k)", y = "Within-cluster Sum of Squares") +
  theme_minimal()
```



The elbow plot above is a way to choose the number of clusters. The y-axis is the within cluster sum of squares (WSS), which shows how close the points are to their clusters. The x-axis is the number of clusters. The elbow point is usually the best number of clusters since adding more clusters does not improve much. From the above plot, the elbow point is located at k = 5. Hence, 5 clusters are being considered.

```
# k-mean cluster for enhancers
set.seed(123)
km <- kmeans(motif_matrix, centers = 5, nstart = 10)
cleaned_data$enhancer_group <- as.factor(km$cluster)

cleaned_data %>%
  count(enhancer_group, name = "count") %>%
  arrange(desc(count))
```

```
##   enhancer_group count
```

```
## 1              2  4352
## 2              3  1250
## 3              4  1126
## 4              1   656
## 5              5   433
```

Logistic Model

The response variable of the logistic model is activity class (a binary variable) where enhancers with activity score above the overall median activity score are grouped as high class, and the rest are grouped as low class. The predictors are the 21 transcription factors. The two random effects are the enhancer cluster and a factor variable for the main region of the enhancer (3R,2L,X, etc.).

```
# model
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 4.4.3
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
glmer_model <- glmer(activity_class ~ . - enhancer_group - seq + (1 | enhancer_group)
                     + (1|seq),
                     data = cleaned_data,
                     family = binomial)

summary(glmer_model)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: activity_class ~ . - enhancer_group - seq + (1 | enhancer_group) +
##     (1 | seq)
##    Data: cleaned_data
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##   10669.6   10836.7   -5310.8   10621.6      7793
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1488 -1.0186  0.4159  0.9670  2.1144
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  seq            (Intercept) 0.03509  0.1873
```

```
##  enhancer_group (Intercept) 0.04619  0.2149
## Number of obs: 7817, groups:  seq, 7; enhancer_group, 5
##
## Fixed effects:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.135142   0.140361   0.963  0.33564
## XBP1             0.017124   0.048178   0.355  0.72227
## Trl             -0.046573   0.052411  -0.889  0.37421
## xrp1            -0.273844   0.063251  -4.330 1.49e-05 ***
## USP              0.440434   0.112192   3.926 8.65e-05 ***
## bergman_EcR_usp  0.047059   0.056679   0.830  0.40638
## CF2              0.052185   0.044653   1.169  0.24253
## bergman_Rel      0.096834   0.069701   1.389  0.16475
## da               0.235260   0.077025   3.054  0.00226 **
## crp              0.230233   0.052909   4.351 1.35e-05 ***
## EcR             -0.238609   0.123508  -1.932  0.05337 .
## GATA_elemento   -0.228082   0.088550  -2.576  0.01000 *
## ERR             -0.253631   0.085158  -2.978  0.00290 **
## Eip74EF         -0.033029   0.095116  -0.347  0.72840
## h                0.006086   0.064302   0.095  0.92460
## gcm              0.021062   0.075340   0.280  0.77982
## kay_Jra         -0.117045   0.061066  -1.917  0.05528 .
## Hnf4             0.095975   0.094954   1.011  0.31213
## srp_SANGER      -0.202906   0.068981  -2.941  0.00327 **
## slp2_forkhead   -0.116131   0.058816  -1.974  0.04833 *
## Rel_FFS         -0.036758   0.063296  -0.581  0.56142
## Tgo             -0.107488   0.076008  -1.414  0.15731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.4.2
```

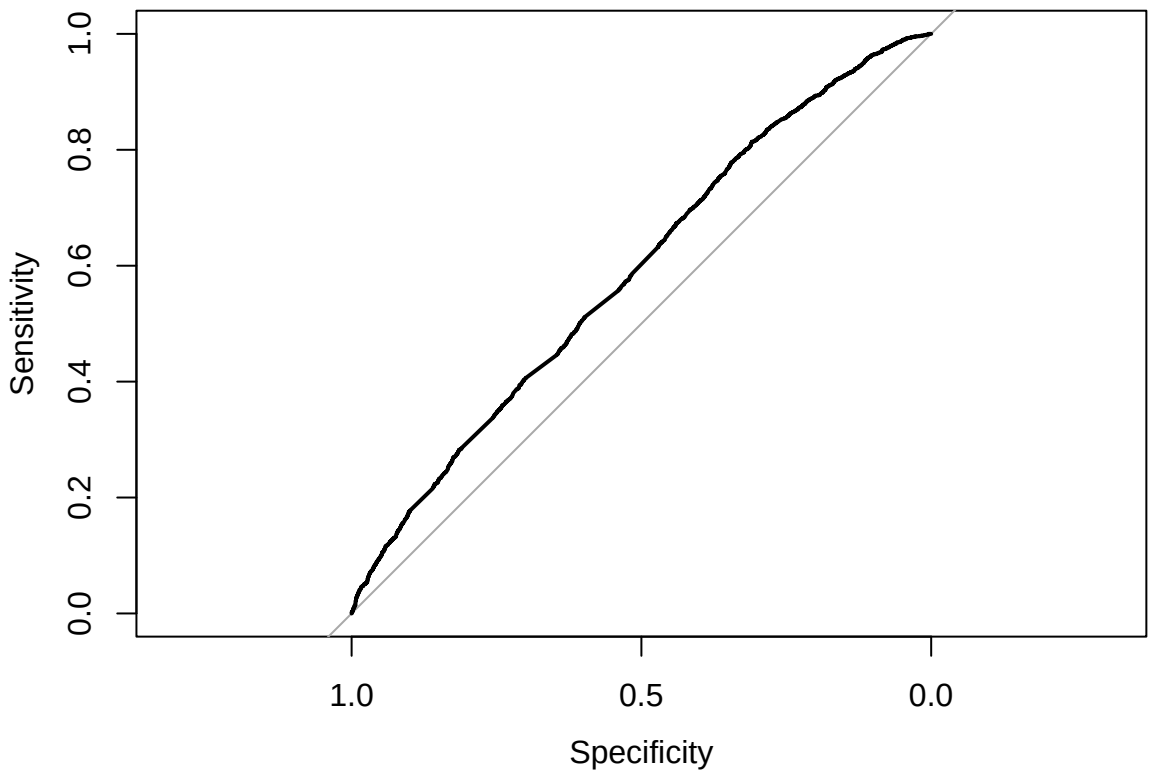
```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
pred_probs_1 <- predict(glmer_model, type = "response")
roc_curve <- roc(cleaned_data$activity_class, pred_probs_1)
```

```
## Setting levels: control = High, case = Low
```

```
## Setting direction: controls < cases
```

```
plot(roc_curve)
```




```
auc(roc_curve)
```

```
## Area under the curve: 0.5872
```

The above result shows the summary of the logistic regrerssion. The transcription factors of xrp1, usp, da, crp, GATA_elemento, ERR, srp_SANGER, and slp2_forkhead are statistically significant. The AUC (Area under the curve) here is 0.5872.

```
# prediction
pred_class_1 <- ifelse(pred_probs_1 > 0.5, 1, 0)
table(Predicted = pred_class_1, Actual = cleaned_data$activity_class)
```

```
##           Actual
## Predicted High  Low
##         0 1669 1242
##         1 2239 2667
```

```
# confusion matrix
TP <- 2239  # True Positives
```

```r
FP <- 2667  # False Positives
FN <- 1669  # False Negatives
TN <- 1242  # True Negatives

# Accuracy
accuracy <- (TP + TN) / (TP + TN + FP + FN)

# Precision
precision <- TP / (TP + FP)

# Recall
recall <- TP / (TP + FN)

# F1-score
f1 <- 2 * (precision * recall) / (precision + recall)

# print result
cat("Accuracy: ", round(accuracy, 4), "\n")
```

```
## Accuracy:  0.4453
```

```r
cat("Precision: ", round(precision, 4), "\n")
```

```
## Precision:  0.4564
```

```r
cat("Recall: ", round(recall, 4), "\n")
```

```
## Recall:  0.5729
```

```r
cat("F1-score: ", round(f1, 4), "\n")
```

```
## F1-score:  0.5081
```

Above shows a summary of the overall prediction of the logistic regression as an evaluation of the performance.