Unit 2 - R Programming Work Report

Submitted By:

2021UCA1536 - Aparna Srivastava

2021UCA1804 - Pulkit Batra

2021UCA1814 - Muskan Sinha

1. Implement Basic Data Structure in R

2. Implement Linear Regression in R and Visualize the results.

3. Implement Logistic Regression in R and Visualize the results.

4. Implement any Machine learning Algorithm along with feature selection and data visualization on any dataset of your choice.

1. Implement Basic Data Structure in R

RPubs Link

BasicDataStructures.html

Vectors:

```
# Creating a numeric vector
numeric_vector <- c(1, 2, 3, 4, 5)
print(numeric_vector)

# Creating a character vector
character_vector <- c("apple", "orange", "banana")
print(character_vector)
```

Matrices:

```
# Creating a matrix
matrix_data <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol =
3)
print(matrix_data)
```

Lists:

```
# Creating a list
list_data <- list(numbers = c(1, 2, 3), fruits = c("apple",
"orange", "banana"))
print(list_data)
```

Data Frames:

```
# Creating a data frame
data_frame_data <- data.frame(
  name = c("Alice", "Bob", "Charlie"),
  age = c(25, 30, 22),
  gender = c("Female", "Male", "Male")
)
print(data_frame_data)
```

Factors:

```
# Creating a factor
gender_factor <- factor(c("Male", "Female", "Male", "Femal
e"))
print(gender_factor)
```

Basic Data Structures in R

Vectors

```
# Creating a numeric vector
numeric_vector <- c(1, 2, 3, 4, 5)
print(numeric_vector)
```

```
## [1] 1 2 3 4 5
```

```
# Creating a character vector
character_vector <- c("apple", "orange", "banana")
print(character_vector)
```

```
## [1] "apple"  "orange" "banana"
```

Matrices

```
# Creating a matrix
matrix_data <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
print(matrix_data)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Lists

```
# Creating a list
list_data <- list(numbers = c(1, 2, 3), fruits = c("apple", "orange", "banana"))
print(list_data)
```

```
## $numbers
## [1] 1 2 3
##
## $fruits
## [1] "apple"  "orange" "banana"
```

Data Frames

```
# Creating a data frame
data_frame_data <- data.frame(
  name = c("Alice", "Bob", "Charlie"),
  age = c(25, 30, 22),
  gender = c("Female", "Male", "Male")
)
print(data_frame_data)
```

```
##      name age gender
## 1   Alice  25 Female
## 2     Bob  30   Male
## 3 Charlie  22   Male
```

Factors

```
# Creating a factor
gender_factor <- factor(c("Male", "Female", "Male", "Female"))
print(gender_factor)
```

```
## [1] Male   Female Male   Female
## Levels: Female Male
```

2. Implement Linear Regression in R and Visualize the results.

RPubs Link

LinearRegression.nb.pdf

LinearReggresion

Code

Loading ggplot

Hide

```
library(ggplot2)
```

Print the head of the dataset

Hide

```
path <- "/Users/pulkitbatra/Downloads/archive-2/train.csv"
trainingSet = read.csv(path)
```

Check for NA and missing values is.na return a vector with value TT for missing values.

Hide

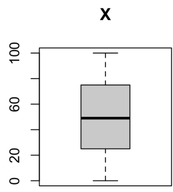
```
numberOfNA = length(which(is.na(trainingSet)==T))
if(numberOfNA > 0) {
  cat('Number of missing values found: ', numberOfNA)
  cat('\nRemoving missing values...')
  trainingSet = trainingSet[complete.cases(trainingSet), ]
}
```

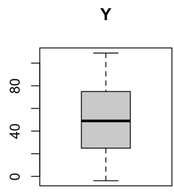
```
Number of missing values found:  1
Removing missing values...
```

Check for outliers Divide the graph area in 2 columns

Hide

```
par(mfrow = c(1, 2))
# Boxplot for X
boxplot(trainingSet$x, main='X', sub=paste('Outliers: ', boxplot.stats(trainingSet$x)$out))
# Boxplot for Y
boxplot(trainingSet$y, main='Y', sub=paste('Outliers: ', boxplot.stats(trainingSet$y)$out))
```





Outliers:

Outliers:

Hide

```
cor(trainingSet$x, trainingSet$y)
```

```
[1] 0.9953399
```

0.99 shows a very strong relation.

Hide

```
regressor = lm(formula = y ~.,
               data = trainingSet)
```

Hide

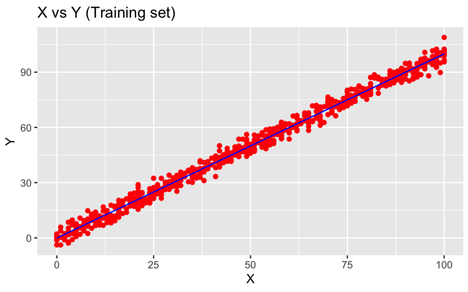
```
summary(regressor)
```

```

Call:
lm(formula = y ~ ., data = trainingSet)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.1523 -2.0179  0.0325  1.8573  8.9132 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.107265   0.212170  -0.506    0.613    
x            1.000656   0.003672 272.510   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.809 on 697 degrees of freedom
Multiple R-squared:  0.9907,    Adjusted R-squared:  0.9907 
F-statistic: 7.426e+04 on 1 and 697 DF,  p-value: < 2.2e-16
```

plot

Hide

```
ggplot() +
  geom_point(aes(x = trainingSet$x, y = trainingSet$y),
             colour = 'red') +
  geom_line(aes(x = trainingSet$x, y = predict(regressor, newdata = trainingSet)),
            colour = 'blue') +
  ggtitle('X vs Y (Training set)') +
  xlab('X') +
  ylab('Y')
```



Test

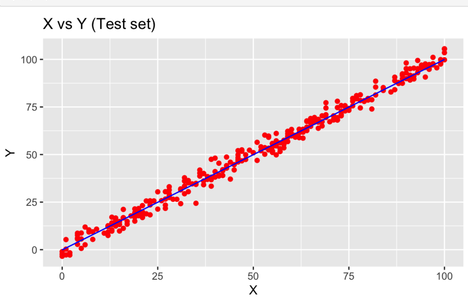
Hide

```
testPath <- "/Users/pulkitbatra/Downloads/archive-2/test.csv"
testSet = read.csv(testPath)

y_pred = predict(regressor, newdata = testSet)
```

Visualsing the result

Hide

```
ggplot() +
  geom_point(aes(x = testSet$x, y = testSet$y),
             colour = 'red') +
  geom_line(aes(x = trainingSet$x, y = predict(regressor, newdata = trainingSet)),
            colour = 'blue') +
  ggtitle('X vs Y (Test set)') +
  xlab('X') +
  ylab('Y')
```



Plot shows model was a good fit.

Hide

```
compare <- cbind (actual=testSet$x, y_pred)  # combine actual and predicted
mean (apply(compare, 1, min)/apply(compare, 1, max))
```

```
[1] -Inf
```

Hide

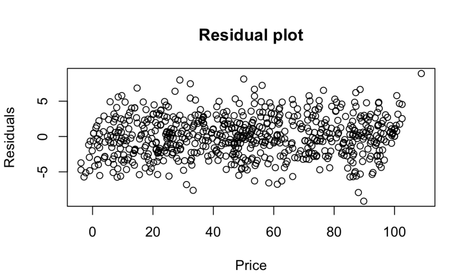
```
mean(0.9,0.9,0.9,0.9,0.9)
```

```
[1] 0.9
```

Check for residual mean and distribution

Hide

```
plot(trainingSet$y, resid(regressor),
     ylab="Residuals", xlab="Price",
     main="Residual plot")
```



Hide

```
mean(regressor$residuals)
```

```
[1] -1.353233e-16
```

3. Implement Logistic Regression in R and Visualize the results.

RPubs Link

Logistic.nb.pdf

Logistic Reggression

Code

Load Data

```
path <- "/Users/pulkitbatra/Desktop/CACSC19/Unit-2 R Programming/Learning R/Assignment/Placement_Data_Full_Class.
csv"



library(dplyr)
library(ggplot2)
location <- "../input/factors-affecting-campus-placement/Placement_Data_Full_Class.csv"
placement.df <- read.csv(path)
# select only relevant columns
placement.lr <- placement.df %>% select(ends_with("_p"), -etest_p, status)
table(placement.lr$status)
```

```
Not Placed     Placed
        67        148
```

```
placement.lr$status <- ifelse(placement.lr$status == "Not Placed", 1, 0)
table(placement.lr$status)
```

```
  0   1
148  67
```

```
library(caTools)
```

```
# Train and Test data
library(caTools) # to split data into train and test
set.seed(101)
sample <- sample.split(placement.lr$status, SplitRatio = 0.80)
train.lr = subset(placement.lr, sample == TRUE)
test.lr = subset(placement.lr, sample == FALSE)
#check the splits
prop.table(table(train.lr$status))
```

```
        0         1
0.6860465 0.3139535
```

```
prop.table(table(test.lr$status))
```

```
        0         1
0.6976744 0.3023256
```

```
# Train the model
model.lr <- glm(status ~ degree_p, family = binomial, data = train.lr)
summary(model.lr)
```

```
Call:
glm(formula = status ~ degree_p, family = binomial, data = train.lr)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.43688    2.24817   5.087 3.63e-07 ***
degree_p    -0.18851    0.03509  -5.372 7.79e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 214.05  on 171  degrees of freedom
Residual deviance: 173.35  on 170  degrees of freedom
AIC: 177.35

Number of Fisher Scoring iterations: 5
```

```
# prediction
lr.pred <- predict(model.lr, newdata = test.lr, type = "response")
head(lr.pred)
```

```
        15         17         22         25         33         35
0.88198047 0.28303502 0.01008494 0.03139780 0.25345579 0.83675747
```

```
# The probabilities always refer to the class dummy-coded as “1”
head(test.lr$status)
```

```
[1] 1 0 0 0 0 1
```

```
# Classification Table
# categorize into groups based on the predicted probability
lr.pred.class <- ifelse(lr.pred>=0.5, 1, 0)
head(lr.pred.class)
```

```
15 17 22 25 33 35
 1  0  0  0  0  1
```

```
table(lr.pred.class)
```

```
lr.pred.class
 0  1
34  9
```

```
table(test.lr$status)
```

```
 0  1
30 13
```

```
conf.matrix <- table(test.lr$status, lr.pred.class)
conf.matrix
```

```
   lr.pred.class
     0  1
  0 30  0
  1  4  9
```

```
rownames(conf.matrix) <- c("Placed", "Not Placed")
colnames(conf.matrix) <- c("Placed", "Not Placed")
addmargins(conf.matrix)
```

```
            lr.pred.class
             Placed Not Placed Sum
  Placed         30          0  30
  Not Placed      4          9  13
  Sum            34          9  43
```

```
# model accuracy
mean((test.lr$status == lr.pred.class))
```

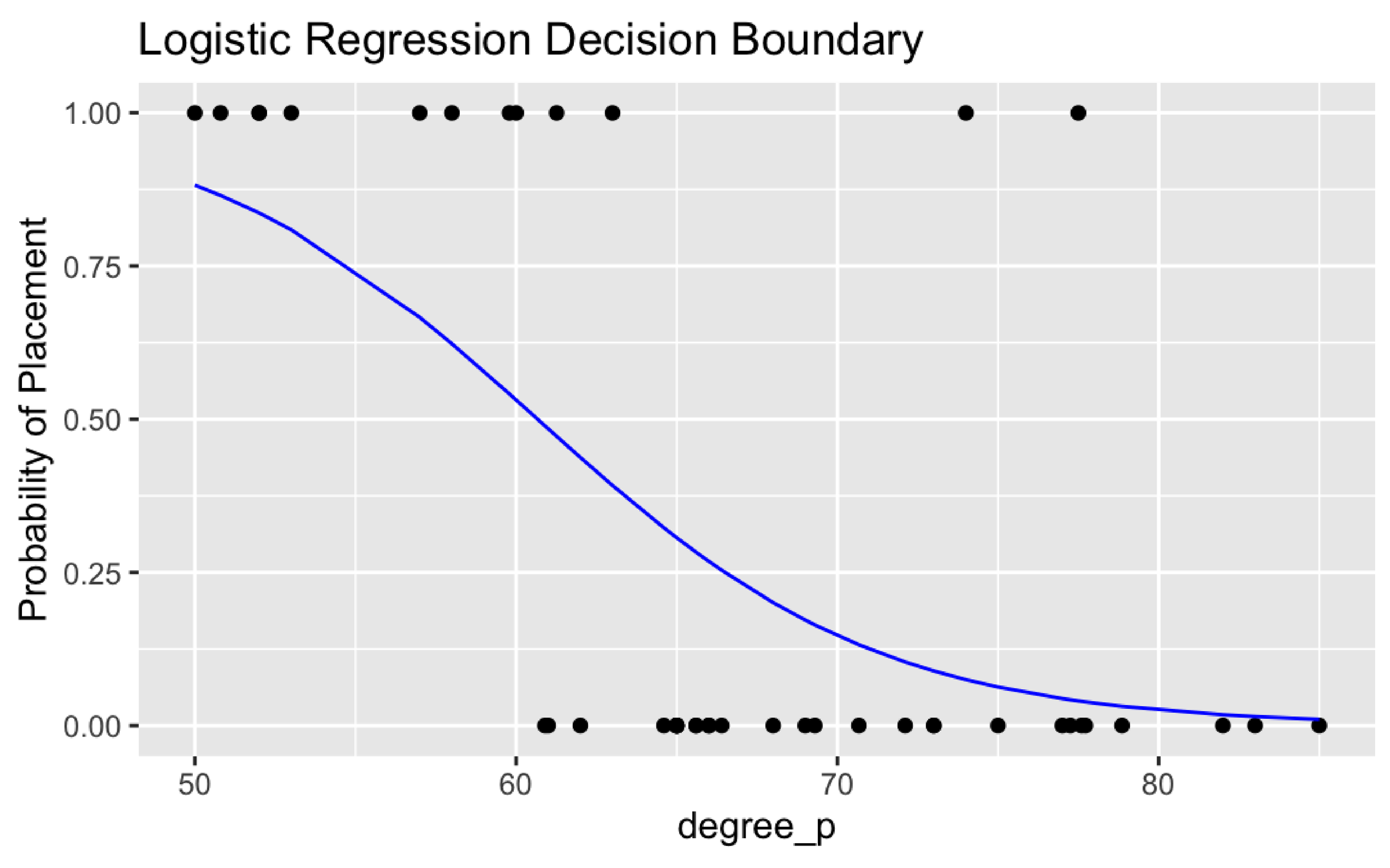
```
[1] 0.9069767
```

```
# different cut-off
lr.pred.class1 <- ifelse(lr.pred>=0.35, 1, 0)
conf.matrix1 <- table(test.lr$status, lr.pred.class1)
conf.matrix1
```

```
   lr.pred.class1
     0  1
  0 27  3
  1  2 11
```

Plots

```
ggplot(data = test.lr, aes(x = degree_p, y = status)) +
  geom_point() +
  geom_line(aes(y = lr.pred), color = "blue") +
  labs(title = "Logistic Regression Decision Boundary",
       x = "degree_p",
       y = "Probability of Placement")
```




```
install.packages("pROC")
```

```
Installing package into ‘/opt/homebrew/lib/R/4.3/site-library’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/src/contrib/pROC_1.18.5.tar.gz'
Content type 'application/x-gzip' length 696162 bytes (679 KB)
==================================================
downloaded 679 KB

* installing *source* package ‘pROC’ ...
** package ‘pROC’ successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C++ compiler: ‘Apple clang version 15.0.0 (clang-1500.0.40.1)’
using SDK: ‘MacOSX14.2.sdk’
```

```
clang++ -std=gnu++17 -I"/opt/homebrew/Cellar/r/4.3.2/lib/R/include" -DNDEBUG  -I'/opt/homebrew/lib/R/4.3/site-lib
rary/Rcpp/include' -I/opt/homebrew/opt/gettext/include -I/opt/homebrew/opt/readline/include -I/opt/homebrew/opt/x
z/include -I/opt/homebrew/include    -fPIC  -g -O2  -c RcppExports.cpp -o RcppExports.o
clang++ -std=gnu++17 -I"/opt/homebrew/Cellar/r/4.3.2/lib/R/include" -DNDEBUG  -I'/opt/homebrew/lib/R/4.3/site-lib
rary/Rcpp/include' -I/opt/homebrew/opt/gettext/include -I/opt/homebrew/opt/readline/include -I/opt/homebrew/opt/x
z/include -I/opt/homebrew/include    -fPIC  -g -O2  -c RcppVersion.cpp -o RcppVersion.o
clang++ -std=gnu++17 -I"/opt/homebrew/Cellar/r/4.3.2/lib/R/include" -DNDEBUG  -I'/opt/homebrew/lib/R/4.3/site-lib
rary/Rcpp/include' -I/opt/homebrew/opt/gettext/include -I/opt/homebrew/opt/readline/include -I/opt/homebrew/opt/x
z/include -I/opt/homebrew/include    -fPIC  -g -O2  -c delong.cpp -o delong.o
clang++ -std=gnu++17 -I"/opt/homebrew/Cellar/r/4.3.2/lib/R/include" -DNDEBUG  -I'/opt/homebrew/lib/R/4.3/site-lib
rary/Rcpp/include' -I/opt/homebrew/opt/gettext/include -I/opt/homebrew/opt/readline/include -I/opt/homebrew/opt/x
z/include -I/opt/homebrew/include    -fPIC  -g -O2  -c perfsAll.cpp -o perfsAll.o
clang++ -std=gnu++17 -dynamiclib -Wl,-headerpad_max_install_names -undefined dynamic_lookup -L/opt/homebrew/Cella
r/r/4.3.2/lib/R/lib -L/opt/homebrew/opt/gettext/lib -L/opt/homebrew/opt/readline/lib -L/opt/homebrew/opt/xz/lib -
L/opt/homebrew/lib -o pROC.so RcppExports.o RcppVersion.o delong.o perfsAll.o -L/opt/homebrew/Cellar/r/4.3.2/lib/
R/lib -lR -lintl -Wl,-framework -Wl,CoreFoundation
```

```
installing to /opt/homebrew/lib/R/4.3/site-library/00LOCK-pROC/00new/pROC/libs
** R
** data
*** moving datasets to lazyload DB
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary location
** checking absolute paths in shared objects and dynamic libraries
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (pROC)

The downloaded source packages are in
    ‘/private/var/folders/gs/jr7fg_pj3kdbfx9sj3vfs7680000gn/T/RtmpLbjxS0/downloaded_packages’
```

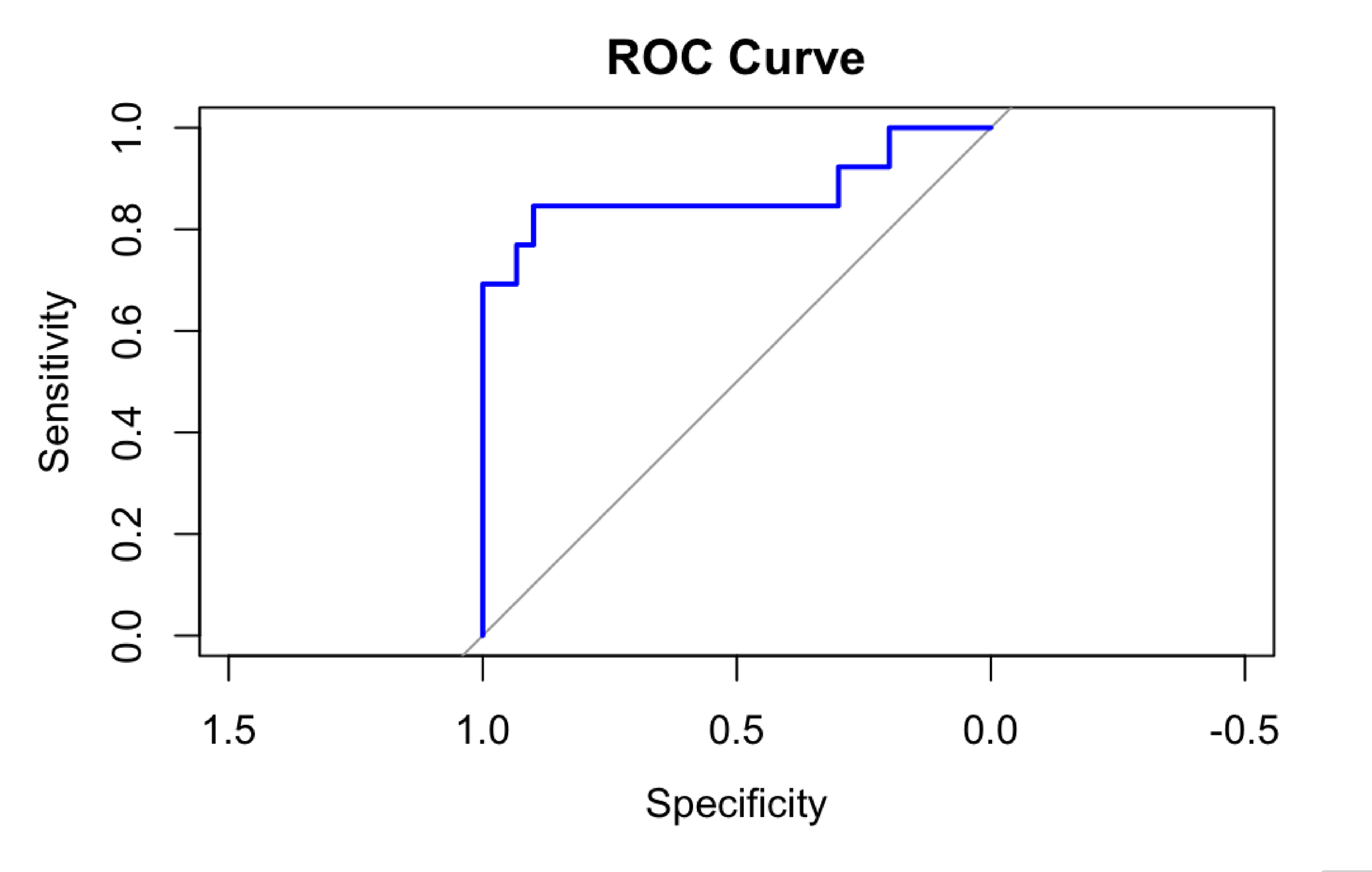
```
library(pROC)
```

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

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:

    cov, smooth, var
```

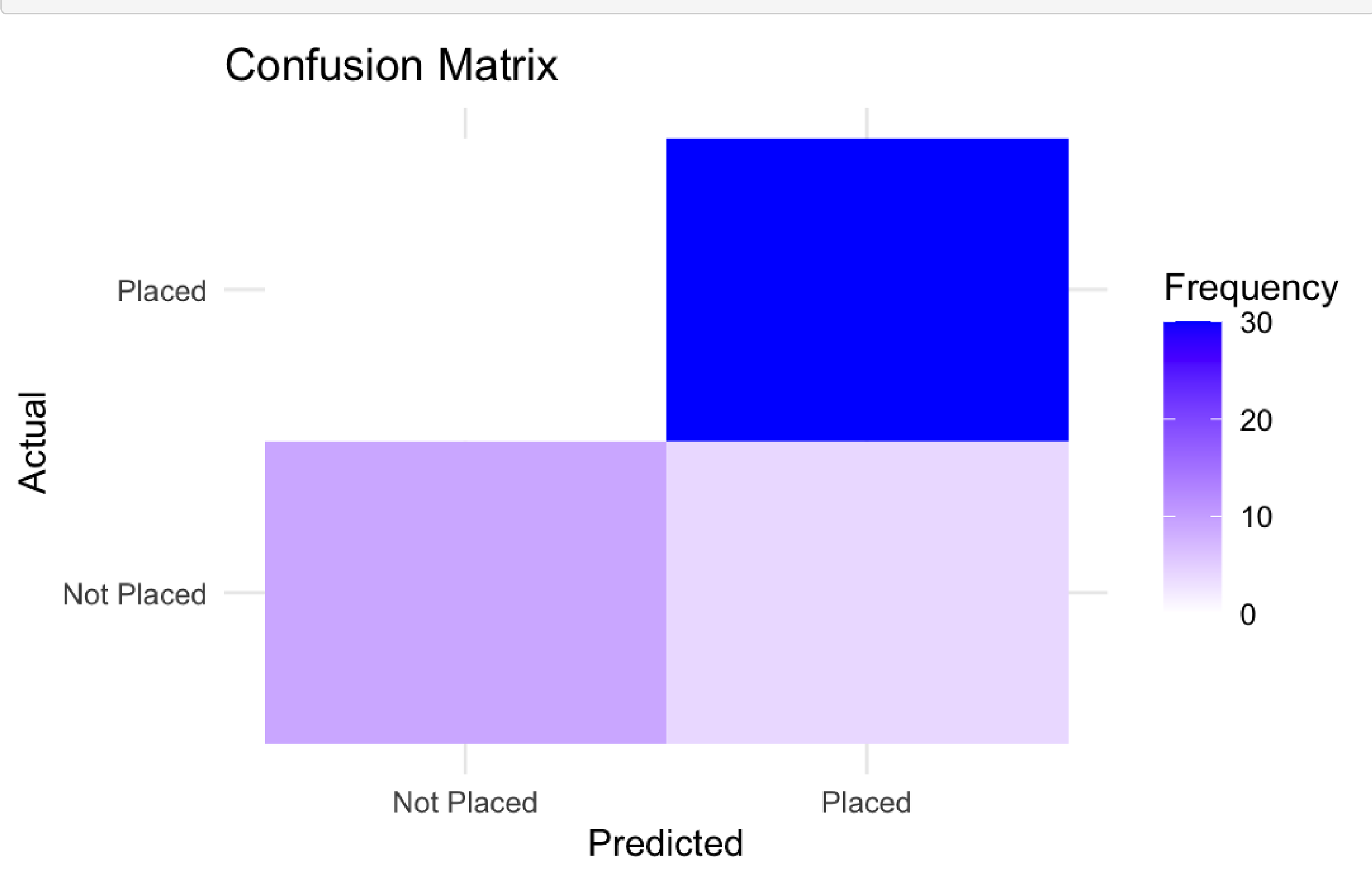
```
roc_curve <- roc(test.lr$status, lr.pred)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
```




```
library(ggplot2)

# Convert confusion matrix to a data frame
conf_matrix_df <- as.data.frame.matrix(conf.matrix)
conf_matrix_df <- cbind(Actual = rownames(conf_matrix_df), conf_matrix_df)

# Reshape data for ggplot
conf_matrix_long <- tidyr::gather(conf_matrix_df, key = "Predicted", value = "Frequency", -Actual)

# Create heatmap using ggplot2
ggplot(data = conf_matrix_long, aes(x = Predicted, y = Actual, fill = Frequency)) +
  geom_tile() +
  labs(title = "Confusion Matrix", x = "Predicted", y = "Actual") +
  scale_fill_gradient(low = "white", high = "blue") +
  theme_minimal()
```




```
NA
NA
```

4. Implement any Machine learning Algorithm along with feature selection and data visualization on any dataset of your choice.

RPubs Link

SVM.nb.pdf

R Notebook

Code

Hide

```
install.packages("caret")
```

```
Installing package into '/Users/pulkitbatra/Library/R/arm64/4.3/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/src/contrib/caret_6.0-94.tar.gz'
Content type 'application/x-gzip' length 2274203 bytes (2.2 MB)
==================================================
downloaded 2.2 MB

* installing *source* package 'caret' ...
** package 'caret' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C compiler: 'Apple clang version 15.0.0 (clang-1500.0.40.1)'
using SDK: 'macOSX14.2.sdk'
```

```
clang -I"/opt/homebrew/Cellar/r/4.3.2/lib/R/include" -DNDEBUG   -I/opt/homebrew/opt/gettext/include -I/opt/homebrew/opt/readline/include -I/opt/homebrew/opt/xz/include -I/opt/homebrew/include    -fPIC  -g -O2  -c caret.c -o caret.o
clang -dynamiclib -Wl,-headerpad_max_install_names -undefined dynamic_lookup -L/opt/homebrew/Cellar/r/4.3.2/lib/R/lib -L/opt/homebrew/opt/gettext/lib -L/opt/homebrew/opt/readline/lib -L/opt/homebrew/opt/xz/lib -L/opt/homebrew/lib -o caret.so caret.o -L/opt/homebrew/Cellar/r/4.3.2/lib/R/lib -lR -lintl -Wl,-framework -Wl,CoreFoundation
```

```
installing to /Users/pulkitbatra/Library/R/arm64/4.3/library/00LOCK-caret/00new/caret/libs
** R
** data
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** checking absolute paths in shared objects and dynamic libraries
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (caret)

The downloaded source packages are in
    '/private/var/folders/gs/jr7fg_pj3kdbfx9sj3vfs7680000gn/T/RtmpCXxqzT/downloaded_packages'
```

Hide

```
install.packages("ggplot2")
```

```
Installing package into '/Users/pulkitbatra/Library/R/arm64/4.3/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/src/contrib/ggplot2_3.4.4.tar.gz'
Content type 'application/x-gzip' length 3159578 bytes (3.0 MB)
==================================================
downloaded 3.0 MB

* installing *source* package 'ggplot2' ...
** package 'ggplot2' successfully unpacked and MD5 sums checked
** using staged installation
** R
** data
*** moving datasets to lazyload DB
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
*** copying figures
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (ggplot2)

The downloaded source packages are in
    '/private/var/folders/gs/jr7fg_pj3kdbfx9sj3vfs7680000gn/T/RtmpCXxqzT/downloaded_packages'
```

Hide

```
library(caret)
library(randomForest)
library(ggplot2)
```

Hide

```
# Load mtcars dataset
data(mtcars)

# Explore the dataset
str(mtcars)
```

```
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Hide

```
set.seed(123)
indices <- createDataPartition(mtcars$mpg, p = 0.7, list = FALSE)
train_data <- mtcars[indices, ]
test_data <- mtcars[-indices, ]
```

Hide

```
rf_model <- randomForest(mpg ~ ., data = train_data, ntree = 100)
```

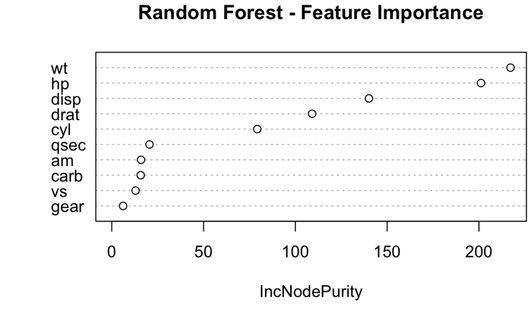
Hide

```
importance <- importance(rf_model)
print(importance)
```

```
     IncNodePurity
cyl       79.24559
disp     140.08452
hp       201.12318
drat     109.11472
wt       217.19134
qsec      20.53858
vs        12.94892
am        15.93240
gear       6.17807
carb      15.76233
```

Hide

```
# Plot feature importance
varImpPlot(rf_model, main = "Random Forest - Feature Importance")
```



Hide

```
# Predictions on the test set
predictions <- predict(rf_model, test_data)
```

Hide

```
# Scatter plot of predicted vs actual mpg
ggplot() +
  geom_point(aes(x = test_data$mpg, y = predictions), color = "blue") +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  ggtitle("Scatter Plot of Actual vs Predicted mpg") +
  xlab("Actual mpg") +
  ylab("Predicted mpg") +
  theme_minimal()
```