Final project

Leena

Sridhar

2023-04-13

#reading the data

# Load required packages  
library(dplyr) # For data manipulation

##   
## 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(readxl) # For reading Excel files  
library(ggplot2) # For data visualization  
  
  
# Load the data from Excel file  
file\_path <- "C:/Users/kavur/OneDrive/Desktop/Advanced R/final project/statistics-clinical-20201221.xlsx" # Update with your file path  
df <- read\_excel(file\_path)  
  
# Perform basic data cleaning and preprocessing  
# For example, removing unnecessary columns and renaming columns for better readability  
df <- select(df, -No.)  
colnames(df) <- c("PatientID", "Sex", "Age", "Weight", "T\_Stage", "N\_Stage", "M\_Stage", "Histopath\_Grading", "Smoking\_History")

# Perform exploratory data analysis (EDA)

# For example, checking data summary, generating plots, etc.  
summary(df$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 28.00 55.00 62.00 61.24 68.00 90.00 1

summary(df$Weight)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 35.50 57.65 65.00 65.32 72.75 103.00 4

age: Min. 1st Qu. Median Mean 3rd Qu. Max. NA’s 28.00 55.00 62.00 61.24 68.00 90.00 1

weight Min. 1st Qu. Median Mean 3rd Qu. Max. NA’s 35.50 57.65 65.00 65.32 72.75 103.00 4

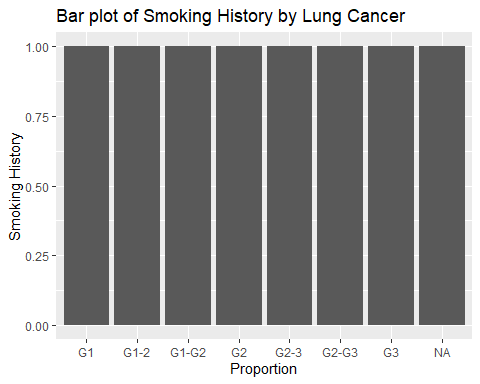
# Remove rows with missing values in "Age" and "Weight" columns  
df <- df[complete.cases(df[c("Age", "Weight")]), ]

library(ggplot2)  
  
# Verify that df is a data frame and has the correct column names  
str(df)

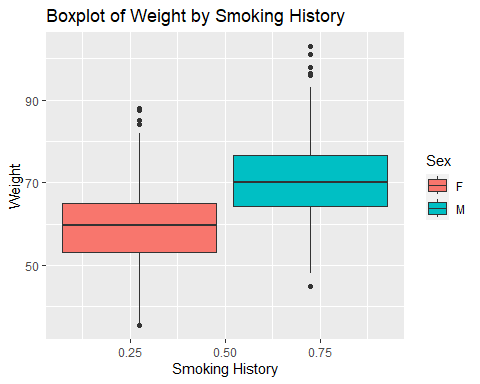
## tibble [350 × 9] (S3: tbl\_df/tbl/data.frame)  
## $ PatientID : chr [1:350] "A0001" "A0002" "A0003" "A0004" ...  
## $ Sex : chr [1:350] "M" "F" "M" "F" ...  
## $ Age : num [1:350] 58 53 60 48 70 36 69 66 59 57 ...  
## $ Weight : num [1:350] 65 55 62 56 75 57 74 60.5 75 53 ...  
## $ T\_Stage : chr [1:350] "2b" "2b" "1c" "1c" ...  
## $ N\_Stage : num [1:350] 3 1 1 3 0 2 0 0 3 3 ...  
## $ M\_Stage : chr [1:350] "1b" "0" "0" "0" ...  
## $ Histopath\_Grading: chr [1:350] "G3" NA "G3" NA ...  
## $ Smoking\_History : num [1:350] 1 0 0 0 1 0 1 0 0 0 ...

ggplot(df, aes(x =Histopath\_Grading , fill =Smoking\_History )) +  
 geom\_bar(position = "fill") +  
 labs(x ="Proportion" , y = "Smoking History", title = "Bar plot of Smoking History by Lung Cancer")

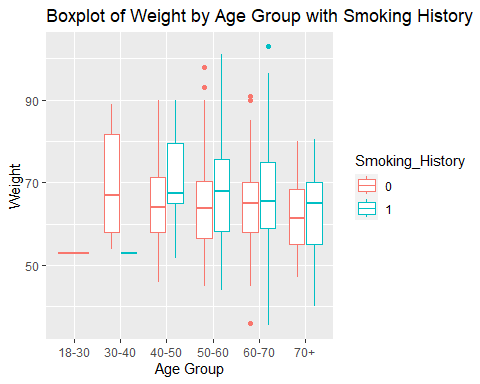
## Warning: The following aesthetics were dropped during statistical transformation: fill  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



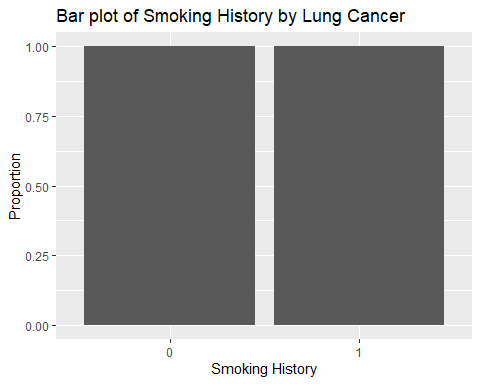
ggplot(df, aes(x = Smoking\_History , y = Weight, fill = Sex)) +  
 geom\_boxplot() +  
 labs(x = "Smoking History", y = "Weight", title = "Boxplot of Weight by Smoking History")



# Create age groups  
age\_breaks <- c(0, 18, 30, 40, 50, 60, 70, Inf) # Define the age group breaks  
age\_labels <- c("0-18", "18-30", "30-40", "40-50", "50-60", "60-70", "70+")  
df$Age\_Group <- cut(df$Age, breaks = age\_breaks, labels = age\_labels)  
df$Age\_Group<-as.factor(df$Age\_Group)  
df$Smoking\_History<-as.factor(df$Smoking\_History)  
  
# Plot data based on age groups  
  
ggplot(df, aes(x = Age\_Group, y = Weight, color = Smoking\_History)) +  
 geom\_boxplot() +  
 labs(x = "Age Group", y = "Weight", title = "Boxplot of Weight by Age Group with Smoking History")



# Perform feature engineering  
# For example, calculating BMI from weight and age  
df$BMI <- df$Weight / ((df$Age / 100)^2)  
  
ggplot(df, aes(x = Smoking\_History, fill = )) +  
 geom\_bar(position = "fill") +  
 labs(x = "Smoking History", y = "Proportion", title = "Bar plot of Smoking History by Lung Cancer")



# Perform statistical analysis

# For example, conducting t-tests to compare BMI between different groups  
t.test(df$BMI[df$Smoking\_History == 0], df$BMI[df$Smoking\_History == 1])

##   
## Welch Two Sample t-test  
##   
## data: df$BMI[df$Smoking\_History == 0] and df$BMI[df$Smoking\_History == 1]  
## t = 3.1182, df = 323.58, p-value = 0.001983  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 10.67732 47.18173  
## sample estimates:  
## mean of x mean of y   
## 205.1750 176.2455

# data analysis on smoking analysis

# Perform chi-squared test for smoking history and T-Stage  
chi\_square\_result\_tstage <- chisq.test(df$Smoking\_History, df$T\_Stage)

## Warning in chisq.test(df$Smoking\_History, df$T\_Stage): Chi-squared approximation  
## may be incorrect

print(chi\_square\_result\_tstage)

##   
## Pearson's Chi-squared test  
##   
## data: df$Smoking\_History and df$T\_Stage  
## X-squared = 23.041, df = 9, p-value = 0.006105

# Perform chi-squared test for smoking history and N-Stage  
chi\_square\_result\_nstage <- chisq.test(df$Smoking\_History, df$N\_Stage)

## Warning in chisq.test(df$Smoking\_History, df$N\_Stage): Chi-squared approximation  
## may be incorrect

print(chi\_square\_result\_nstage)

##   
## Pearson's Chi-squared test  
##   
## data: df$Smoking\_History and df$N\_Stage  
## X-squared = 5.8774, df = 3, p-value = 0.1177

# Perform chi-squared test for smoking history and M-Stage  
chi\_square\_result\_mstage <- chisq.test(df$Smoking\_History, df$M\_Stage)

## Warning in chisq.test(df$Smoking\_History, df$M\_Stage): Chi-squared approximation  
## may be incorrect

print(chi\_square\_result\_mstage)

##   
## Pearson's Chi-squared test  
##   
## data: df$Smoking\_History and df$M\_Stage  
## X-squared = 9.5308, df = 6, p-value = 0.1459

# Perform chi-squared test for smoking history and Histopathological grading  
chi\_square\_result\_histopath\_grade <- chisq.test(df$Smoking\_History, df$Histopath\_Grading)

## Warning in chisq.test(df$Smoking\_History, df$Histopath\_Grading): Chi-squared  
## approximation may be incorrect

print(chi\_square\_result\_histopath\_grade)

##   
## Pearson's Chi-squared test  
##   
## data: df$Smoking\_History and df$Histopath\_Grading  
## X-squared = 7.6552, df = 6, p-value = 0.2645

ML modeling

# For example, training a simple logistic regression model for predicting smoking history  
#library(caret) # For machine learning  
library(glmnet) # For logistic regression

## Warning: package 'glmnet' was built under R version 4.2.3

## Loading required package: Matrix

## Loaded glmnet 4.1-7

# Convert smoking history to a binary factor variable  
df$Smoking\_History <- as.factor(df$Smoking\_History)  
# Assuming that "Na" indicates absence of lung cancer and other values indicate presence of lung cancer  
df$Outcome <- ifelse(is.na(df$Histopath\_Grading), 0, 1)  
  
# Split data into training and testing sets  
set.seed(123) # For reproducibility  
train\_indices <- sample(1:nrow(df), nrow(df) \* 0.8)  
train\_data <- df[train\_indices, ]  
test\_data <- df[-train\_indices, ]  
# Train a logistic regression model  
  
# Fit the logistic regression model  
model1 <- glm(Outcome ~ Age + Weight + Smoking\_History, data = df, family = "binomial")  
  
# Make predictions on the test data  
predictions <- predict(model1, newdata = test\_data, type = "response")  
  
# Convert predicted probabilities to binary outcomes  
predicted\_outcome <- ifelse(predictions > 0.5, 1, 0)  
  
# Generate confusion matrix  
confusion <- table(predicted\_outcome, test\_data$Outcome)  
  
# Print the confusion matrix  
print(confusion)

##   
## predicted\_outcome 0 1  
## 0 46 24

accuracy <- sum(diag(confusion)) / sum(confusion)  
accuracy

## [1] 0.6571429

# Perform CART analysis  
# For example, building a classification tree to predict Smoking History based on Age, Weight, and Histopathological Grading  
# Convert Smoking History to a binary factor variable  
df$Smoking\_History <- as.factor(df$Smoking\_History)  
library(rpart)

## Warning: package 'rpart' was built under R version 4.2.3

# Split data into training and testing sets  
set.seed(3) # For reproducibility  
train\_indices <- sample(1:nrow(df), nrow(df) \* 0.8)  
train\_data <- df[train\_indices, ]  
test\_data <- df[-train\_indices, ]  
# Train a classification tree  
model2 <- rpart(Histopath\_Grading ~ Age + Weight + Smoking\_History, data = train\_data, method = "class")  
  
  
# Visualize the tree  
#par(xpd = NA)  
#plot(model2, box.palette = c("red", "blue"),  
# fallen.leaves = TRUE, under = TRUE, faclen = 0)  
  
# Add class labels to the plot  
#text(model2, use.n = TRUE, all = TRUE, cex = 0.8)  
# Make predictions on the testing set  
predictions <- predict(model2, newdata = test\_data, type = "class")  
# Evaluate model performance  
confusion\_matrix <- table(predictions, test\_data$Histopath\_Grading)  
confusion\_matrix

##   
## predictions G1 G1-2 G1-G2 G2 G2-3 G2-G3 G3  
## G1 0 0 0 1 1 0 1  
## G1-2 0 0 0 0 0 0 0  
## G1-G2 0 0 0 0 0 0 0  
## G2 0 1 0 0 0 0 0  
## G2-3 0 0 0 0 2 0 2  
## G3 2 1 1 4 5 1 7

# Calculate accuracy  
accuracy <- sum(diag(confusion\_matrix)) / sum(confusion\_matrix)  
accuracy

## [1] 0.1034483