AML5103 Odd Semester 2024 Coding Problem Set-1

#### Execute the following cells to load the libraries

library(ggplot2)  
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

#### Load the bats data to a dataframe where the columns correspond to whether a gene is expressed or not expressed in the bats and whether the Ebola virus is present or absent in the bats

dfBat = read.csv("bats.csv", header = TRUE,row.names = 1 , stringsAsFactors = TRUE)  
head(dfBat)

## Gene.1 Gene.2 Gene.3 Gene.4 Gene.5 Ebola  
## 1 FALSE FALSE TRUE TRUE TRUE TRUE  
## 2 TRUE FALSE TRUE FALSE FALSE FALSE  
## 3 TRUE FALSE TRUE TRUE TRUE FALSE  
## 4 FALSE TRUE TRUE TRUE TRUE TRUE  
## 5 FALSE FALSE FALSE TRUE FALSE FALSE  
## 6 TRUE FALSE FALSE TRUE FALSE FALSE

##### Get the genes that are expressed for the 1st bat by using two approaches: (1) logical indexing for the columns in base R (2) the dplyr library

colnames

## function (x, do.NULL = TRUE, prefix = "col")   
## {  
## if (is.data.frame(x) && do.NULL)   
## names(x)  
## else dimnames(x)[[2L]] %||% if (do.NULL)   
## NULL  
## else {  
## nc <- NCOL(x)  
## if (nc > 0L)   
## paste0(prefix, seq\_len(nc))  
## else character()  
## }  
## }  
## <bytecode: 0x0000020aa6211df0>  
## <environment: namespace:base>

colnames(dfBat[1, dfBat[1, -ncol(dfBat)] == TRUE])

## [1] "Gene.3" "Gene.4" "Gene.5"

colnames(dfBat)[(which(dfBat %>% slice(1) %>% select(-c(Ebola)) == TRUE))]

## [1] "Gene.3" "Gene.4" "Gene.5"

#### Calculate the probability that a random bat carries the Ebola virus using two approaches: (1) base R (2) the dplyr library.

mean(dfBat$Ebola)

## [1] 0.300793

dfBat %>% summarise(mean(Ebola))

## mean(Ebola)  
## 1 0.300793

apply(dfBat %>% select(Ebola), 2, mean)

## Ebola   
## 0.300793

colMeans(dfBat %>% select(Ebola))

## Ebola   
## 0.300793

#### Calculate the probability that gene-1 through gene-5 is expressed in a random bat.

dfBat %>% summarise(across(-Ebola,mean))

## Gene.1 Gene.2 Gene.3 Gene.4 Gene.5  
## 1 0.702277 0.300763 0.500895 0.801618 0.3270533

apply(dfBat %>% select(-c(Ebola)), 2, mean)

## Gene.1 Gene.2 Gene.3 Gene.4 Gene.5   
## 0.7022770 0.3007630 0.5008950 0.8016180 0.3270533

#### Is the presence of Ebola virus dependent on the expression of any of the genes 1 through 5? Answer this by noting that if P(Gene-j expressed | Ebola ) = P(Gene-j expressed) approximately based on the data, then the j-th gene’s expression is most likely independent of the bat carrying the Ebola virus and vice versa. Note that P(Gene-j expressed) is calculated in the previous question.

# Subset the dataframe to those bats that carry the Ebola virus  
dfBatEbola = dfBat %>% filter(Ebola==TRUE)  
head(dfBatEbola)

## Gene.1 Gene.2 Gene.3 Gene.4 Gene.5 Ebola  
## 1 FALSE FALSE TRUE TRUE TRUE TRUE  
## 2 FALSE TRUE TRUE TRUE TRUE TRUE  
## 3 TRUE FALSE TRUE TRUE TRUE TRUE  
## 4 TRUE FALSE TRUE TRUE TRUE TRUE  
## 5 TRUE TRUE TRUE TRUE TRUE TRUE  
## 6 TRUE FALSE TRUE TRUE TRUE TRUE

# Conditional probability that gene-1 through gene-5 is expressed given that the bat carries the Ebola virus  
dfBat %>% summarise(across(-Ebola,mean))

## Gene.1 Gene.2 Gene.3 Gene.4 Gene.5  
## 1 0.702277 0.300763 0.500895 0.801618 0.3270533

#### Suppose a bat has one of the genes identified in the previous question expressed. What is the probability that it carries the Ebola virus? From you answers, conclude which gene’s expression is most indicative of a random bat carrying the Ebola virus.

selected\_genes = colnames(dfBat %>% select(-c(Ebola)))   
for (gene in selected\_genes){  
 # Subset the bats that carry Gene-j  
 dfBatGene = dfBat %>% filter(get(gene)==TRUE)  
 # Conditional probability that a random bat carries the Ebola virus given that Gene-j is expressed in it  
 print(dfBatGene %>% summarise(mean(Ebola)))  
}

## mean(Ebola)  
## 1 0.3020206  
## mean(Ebola)  
## 1 0.3022011  
## mean(Ebola)  
## 1 0.5832019  
## mean(Ebola)  
## 1 0.3705418  
## mean(Ebola)  
## 1 0.8999847

#### Load the ‘emergency.csv’ file to a dataframe and answer the following questions:

dfPatient = read.csv("emergency.csv" , header=TRUE ,row.names = 1, stringsAsFactors = TRUE)  
head(dfPatient)

## Condition Insurance  
## 1 serious insured  
## 2 serious insured  
## 3 serious not insured  
## 4 serious insured  
## 5 fair not insured  
## 6 serious not insured

#1. What is the probability that the next incoming patient is in good, fair, and serious condition?

dfPatient %>% filter(Condition == 'good') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.112

dfPatient %>% filter(Condition == 'fair') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.203

dfPatient %>% filter(Condition == 'serious') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.685

1. What is the probability that the next incoming patient is insured and not insured?

dfPatient %>% filter(Insurance == 'insured') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.614

dfPatient %>% filter(Insurance == 'not insured') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.386

1. What is the probability that the next incoming patient is serious and insured?

dfPatient %>% filter((Condition == 'serious') & (Insurance == 'insured')) %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.423

1. Given that the next incoming patient is known to be serious, what is the probability that they are insured?

A = dfPatient %>% filter(Condition == 'serious') %>% group\_by(Insurance) %>% summarise(n = n()) %>% mutate(ConditionalProbability = n / sum(n))   
print(A[1, 'ConditionalProbability'])

## # A tibble: 1 × 1  
## ConditionalProbability  
## <dbl>  
## 1 0.618

1. Given that the next incoming patient is known to be insured, what is the probability that they are serious?

A = dfPatient %>% filter(Insurance == 'insured') %>% group\_by(Condition) %>% summarise(n = n()) %>% mutate(ConditionalProbability = n / sum(n))   
print(A[1, 'ConditionalProbability'])

## # A tibble: 1 × 1  
## ConditionalProbability  
## <dbl>  
## 1 0.205

1. Given that the next incoming patient is known to be not insured, what is the probability that they are in a good or fair condition?

A = dfPatient %>% filter(Insurance == 'not insured') %>% group\_by(Condition) %>% summarise(n = n()) %>% mutate(ConditionalProbability = n / sum(n))   
print(A[1, 'ConditionalProbability'] + A[2, 'ConditionalProbability'])

## ConditionalProbability  
## 1 0.3212435

# For Questions 1. Repeat this for Questions 2 and 3  
dfPatient %>% filter(Condition == 'good') %>% summarise(n()/nrow(dfPatient))

## n()/nrow(dfPatient)  
## 1 0.112

# For Question-4. Repeat this for Questions 5 and 6  
A = dfPatient %>% filter(Condition == 'serious') %>% group\_by(Insurance) %>% summarise(n = n()) %>% mutate(ConditionalProbability = n / sum(n))   
print(A[1, 'ConditionalProbability'])

## # A tibble: 1 × 1  
## ConditionalProbability  
## <dbl>  
## 1 0.618