```r
#Load the DataSet

dfBat=read.csv("bats.csv")
str(dfBat)
```

```
## 'data.frame':    99999 obs. of  7 variables:
##  $ Bat    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gene.1: logi  FALSE TRUE TRUE FALSE FALSE TRUE ...
##  $ Gene.2: logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ Gene.3: logi  TRUE TRUE TRUE TRUE FALSE FALSE ...
##  $ Gene.4: logi  TRUE FALSE TRUE TRUE TRUE TRUE ...
##  $ Gene.5: logi  TRUE FALSE TRUE TRUE FALSE FALSE ...
##  $ Ebola : logi  TRUE FALSE FALSE TRUE FALSE FALSE ...
```

```r
ncol(dfBat)
```

```
## [1] 7
```

```r
nrow(dfBat)
```

```
## [1] 99999
```

```r
logical_columns = c("Gene.1", "Gene.2", "Gene.3", "Gene.4", "Gene.5", "Ebola")
dfBat[logical_columns] = lapply(dfBat[logical_columns], as.numeric)
```

a) What is the chance of a random bat carrying the Ebola virus?

```r
TotalEbolaBats = sum(dfBat$Ebola)
totalBats = nrow(dfBat)
EbolaChance = TotalEbolaBats / totalBats
cat("Chance of a random bat carrying Ebola is", EbolaChance)
```

```
## Chance of a random bat carrying Ebola is 0.300793
```

b) For each gene, calculate the likelihood that it is expressed in a random bat.

```r
likelihood = colMeans(dfBat[, c("Gene.1", "Gene.2", "Gene.3", "Gene.4", "Gene.5")])
dfLikelihood = data.frame(Gene = c("Gene.1", "Gene.2", "Gene.3", "Gene.4", "Gene.5"),Likelihood = likel
print(dfLikelihood)
```

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

c) Is the presence or absence of any of the genes indicative of a random bat potentially carrying the Ebola virus?

Method 1

```r
genes = c("Gene.1", "Gene.2", "Gene.3", "Gene.4", "Gene.5")

# Calculate the proportion of bats carrying Ebola for each gene's presence and absence
for (gene in genes) {
  Ebola = mean(dfBat[dfBat[[gene]] == TRUE, "Ebola"])
  NoEbola = mean(dfBat[dfBat[[gene]] == FALSE, "Ebola"])

  cat("Gene:", gene, "\n")
  cat("Proportion of bats carrying Ebola with", gene, "present:", Ebola, "\n")
  cat("Proportion of bats carrying Ebola with", gene, "absent:", NoEbola)
  cat( "\n \n")
}
```

```
## Gene: Gene.1
## Proportion of bats carrying Ebola with Gene.1 present: 0.3020206
## Proportion of bats carrying Ebola with Gene.1 absent: 0.2978974
##
## Gene: Gene.2
## Proportion of bats carrying Ebola with Gene.2 present: 0.3022011
## Proportion of bats carrying Ebola with Gene.2 absent: 0.3001873
##
## Gene: Gene.3
## Proportion of bats carrying Ebola with Gene.3 present: 0.5832019
## Proportion of bats carrying Ebola with Gene.3 absent: 0.01737127
##
## Gene: Gene.4
## Proportion of bats carrying Ebola with Gene.4 present: 0.3705418
## Proportion of bats carrying Ebola with Gene.4 absent: 0.01895352
##
## Gene: Gene.5
## Proportion of bats carrying Ebola with Gene.5 present: 0.8999847
## Proportion of bats carrying Ebola with Gene.5 absent: 0.009584807
##
```

Method 2 we can use Binomial also

```r
dfBat$Ebola_numeric = as.numeric(dfBat$Ebola)
model = glm(Ebola_numeric ~ Gene.1 + Gene.2 + Gene.3 + Gene.4 + Gene.5,
             data = dfBat, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Ebola_numeric ~ Gene.1 + Gene.2 + Gene.3 + Gene.4 +
##     Gene.5, family = binomial, data = dfBat)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.76822    0.08298 -57.464  < 2e-16 ***
## Gene.1       0.01726    0.03655   0.472  0.63685
## Gene.2       0.02740    0.03651   0.750  0.45300
## Gene.3       0.20558    0.07542   2.726  0.00641 **
## Gene.4       0.07288    0.08258   0.883  0.37746
## Gene.5       6.67023    0.07489  89.069  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 122306  on 99998  degrees of freedom
## Residual deviance:  28537  on 99993  degrees of freedom
## AIC: 28549
##
## Number of Fisher Scoring iterations: 7
```

Method 3 we can use chi-squared on the Dataset also

```r
genes = c("Gene.1", "Gene.2", "Gene.3", "Gene.4", "Gene.5")

for (gene in genes) {
  cross_tab = table(dfBat[[gene]], dfBat$Ebola)
  chi_result = chisq.test(cross_tab)

  cat("Chi-squared test for", gene, "and Ebola:\n")
  print(chi_result)
}
```

```
## Chi-squared test for Gene.1 and Ebola:
##
##  Pearson’s Chi-squared test with Yates’ continuity correction
##
## data:  cross_tab
## X-squared = 1.6706, df = 1, p-value = 0.1962
##
## Chi-squared test for Gene.2 and Ebola:
##
##  Pearson’s Chi-squared test with Yates’ continuity correction
##
## data:  cross_tab
## X-squared = 0.39597, df = 1, p-value = 0.5292
##
## Chi-squared test for Gene.3 and Ebola:
##
##  Pearson’s Chi-squared test with Yates’ continuity correction
##
## data:  cross_tab
## X-squared = 38054, df = 1, p-value < 2.2e-16
##
## Chi-squared test for Gene.4 and Ebola:
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  cross_tab
## X-squared = 9345.1, df = 1, p-value < 2.2e-16
##
## Chi-squared test for Gene.5 and Ebola:
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  cross_tab
## X-squared = 82960, df = 1, p-value < 2.2e-16
```