STA 202 Project 1 Group 3

Kshitij Shekhar AU2020165
Laksh Gupta AU2020152
Vinayak C Manoj AU2020015

Question 1

(A)**Suppose $f(x) = \frac{n!}{x!(n-x)!}p^x(1-p)^{n-x}$, where X can take values from 1 to n Take, n = 20 and p = 0.5. Write some R code to plot the probability mass function and and check if the function is a probability probability function or not?**

```
x = 20
n = 20
p = 0.5

fx = array(0, dim=x) # array of size 20 to store the values of f(x)
X = array(0, dim=x) # array of size 20 to store the values of corresponding x

for (i in 0:x)
{
f = (factorial(n)/(factorial(i)*factorial(n-i)))*(p^i)*((1-p)^(n-i)) # f(x) by definition
    X[i] = i # appending the values of x in 'X'
    fx[i] = f # appending the values of f(x) in 'fx'
}

plot(X, fx, xlab = "x", ylab = "f(x)", main = "Probability Mass Function", type='h') # plotting the probability mass function
```

Probability Mass Function

As $\sum_{x=0}^{n} f(x) = 1$, as shown below, this is a valid probability mass function.

```
x = 20
n = 20
p = 0.5
g = 0 # defined to add values of each f(x)

for (i in 0:x)
{
f = (factorial(n)/(factorial(i)*factorial(n-i)))*(p^i)*((1-p)^(n-i)) # f(x) by definition
    g = g + f # sum of all f(x)
    #print(f)
}

print (g)
```

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

(B)Mention the name of the probability distribution.

This distribution is called the Binomial Distribution

(C)Write R codes to draw a random sample of size n = 20, and find sample mean and sample variance of the probability distribution X.

```
x = 20
n = 20
p = 0.5

fx = array(0, dim=x) # array of size 20 to store the values of f(x)

for (i in 0:x)
{
f = (factorial(n)/(factorial(i)*factorial(n-i)))*(p^i)*((1-p)^(n-i)) # f(x) by definition
    fx[i] = f # appending the values of f(x) in 'fx'
}
y = sample(fx, 20, replace = TRUE) # sampling data from f(x) with allowed replacement
    print(y)
```

```
##  [1] 1.761971e-01 3.696442e-02 1.907349e-05 1.201344e-01 4.029552e-03
##  [6] 1.601791e-01 1.478577e-02 7.392883e-02 3.696442e-02 1.907349e-05
## [11] 3.696442e-02 3.696442e-02 1.201344e-01 1.201344e-01 1.601791e-01
## [16] 1.601791e-01 4.029552e-03 1.201344e-01 1.811981e-04 1.907349e-05
```

```
sample_mean = mean(y) #sample mean
sample_variance = var(y) #sample variance

print(sample_mean)
```

```
## [1] 0.06921959
```

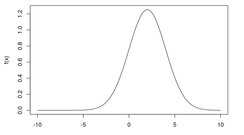
```
print(sample_variance)
```

```
## [1] 0.004269104
```

Question 2

```
f <- function(x) {
  (1/2*sqrt(2*pi)) * exp(-0.125 * (x-2)^2)
}

# Plot the function
curve(f, from = -10, to = 10)
```




```
# Check condition 1: f(x) is non-negative
all(sapply(seq(-10, 10, by = 0.01), f) >= 0) # should return TRUE
```

```
## [1] TRUE
```

```
# Check condition 2: the area under the curve is equal to 1
    integrate(f, lower = -10, upper = 10)
```

```
## 6.282986 with absolute error < 2.3e-08
```

```
# Set the seed for reproducibility
    set.seed(12)

# Values to sample
x_vals <- seq(-5, 10, 0.015)

# Apply PDF to x_vals
y_vals <- sapply(x_vals, f)

samples <- sample(x_vals, 50, replace = TRUE, prob = y_vals)

# Find the sample mean and standard error
    sample_mean <- mean(samples)
sample_se <- sd(samples) / sqrt(length(samples))

# Print the results
print(paste("Sample mean:", sample_mean))
```

```
## [1] "Sample mean: 2.0167"
```

```
print(paste("Sample standard error:", sample_se))
```

```
## [1] "Sample standard error: 0.295690736996647"
```

Question 3

A. **Consider the data given in the above example of Chinese population in Hong Kong in 1937 and estimate the maximum likelihood estimate (MLE) of the parameter θ as $\hat{\theta}$ and estimate the three genotype probabilities for M, MN and N.**

Let X_1, X_2, X_3 denote the counts in the three cells. $n = 1029$. We can find the log likelihood of θ as follows:

$$l(\theta) = log(a\theta) - \sum_{i=1}^{3} logX_i! + X_1 log(1-\theta)^2 + X_2 log2\theta(1-\theta) + X_3 log\theta^2$$

$$= log(a!) - \sum_{i=1}^{3} logX_i! + (2X_1 + X_2)log(1-\theta) + (2X_3 + X_2)log\theta + X_2 log2$$

Setting the derivative to zero to maximize $l(\theta)$,

$$\frac{-2X_1 - X_2}{1-\theta} + \frac{2X_3 + X_2}{\theta} = 0$$

On solving the above equation, we get $\hat{\theta} = 0.4247$

According to the Hardy-Weinberg Equilibrium, if the blood types are in equilibrium, alleles M, MN and N occur in a population with probabilities $(1-\theta)^2, 2\theta(1-\theta)$, and θ^2 respectively. Since we do not know the actual value of θ, we can use $\hat{\theta}$ according to the bootstrap principle.
∴ (M,MN,MN) occur with probability .333 , .489 , .180

(B)How do we assess the preciseness of the estimate $\hat{\theta}$? Make comments.

We can assess the preciseness of $\hat{\theta}$ by using the bootstrap method to estimate the sampling distribution with a histogram, and by extension its standard error

(C)Estimate the sampling distribution of θ̂ (ie, draw the histogram) and report the standard error.

```
observed_freqs=c(342,500,187)
# Calculate the observed value of theta
    f_M <- observed_freqs[1]
f_MN <- observed_freqs[2]
    f_N <- observed_freqs[3]


p_hat <- (2 * f_M + f_MN) / (2 * (f_M + f_MN + f_N))
q_hat <- (2 * f_N + f_MN) / (2 * (f_M + f_MN + f_N))
    theta_hat <- 2 * p_hat * q_hat

# Set the number of bootstrap samples
    n_bootstraps <- 1000

# Create an empty vector to store the bootstrap estimates of theta
theta_hat_boot <- numeric(n_bootstraps)

# Run the bootstrap
    for (i in 1:n_bootstraps) {
# Sample with replacement from the observed frequencies
boot_sample <- sample(observed_freqs, size = length(observed_freqs), replace = TRUE)

# Calculate the estimated value of theta in the bootstrap sample
    f_M_boot <- boot_sample[1]
f_MN_boot <- boot_sample[2]
    f_N_boot <- boot_sample[3]
p_hat_boot <- (2 * f_M_boot + f_MN_boot) / (2 * (f_M_boot + f_MN_boot + f_N_boot))
q_hat_boot <- (2 * f_N_boot + f_MN_boot) / (2 * (f_M_boot + f_MN_boot + f_N_boot))
    theta_hat_boot[i] <- 2 * p_hat_boot * q_hat_boot
}

# Calculate the standard error of the bootstrap estimates of theta
    se_theta_hat_boot <- sd(theta_hat_boot)

# Print the estimated value of theta and the standard error
    cat("Estimated value of theta:", theta_hat, "\n")
```

```
## Estimated value of theta: 0.488655
```

```
cat("Standard error of theta:", se_theta_hat_boot, "\n")
```

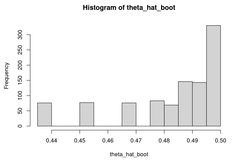
```
## Standard error of theta: 0.01934532
```

```
CI <- quantile(theta_hat_boot, c(0.05, 0.95))
    print(CI)
```

```
##      5%     95%
## 0.4358738 0.5000000
```

```
hist(theta_hat_boot)
```

Histogram of theta_hat_boot



(D)Use bootstrap method (take number of bootstrap sample 500) to estimate the sampling distribution and standard error of the 'θ and 90% confidence interval of θ.

```
n_bootstraps <- 500

# Create an empty vector to store the bootstrap estimates of theta
    theta_hat_boot <- numeric(n_bootstraps)

# Run the bootstrap
    for (i in 1:n_bootstraps) {
# Sample with replacement from the observed frequencies
boot_sample <- sample(observed_freqs, size = length(observed_freqs), replace = TRUE)

# Calculate the estimated frequencies of each blood type in the bootstrap sample
    f_M_boot <- boot_sample[1]
    f_MN_boot <- boot_sample[2]
    f_N_boot <- boot_sample[3]

# Calculate the estimated allele frequencies in the bootstrap sample
p_hat_boot <- (2 * f_M_boot + f_MN_boot) / (2 * (f_M_boot + f_MN_boot + f_N_boot))
q_hat_boot <- (2 * f_N_boot + f_MN_boot) / (2 * (f_M_boot + f_MN_boot + f_N_boot))

# Calculate the estimated theta in the bootstrap sample
    theta_hat_boot[i] <- 2 * p_hat_boot * q_hat_boot
}

# Calculate the standard error of the bootstrap estimates of theta
    se_theta_hat_boot <- sd(theta_hat_boot)

# Calculate the 90% confidence interval for theta
    CI <- quantile(theta_hat_boot, c(0.05, 0.95))
    print(CI)
```

```
##      5%     95%
## 0.4358738 0.5000000
```

```
#print(c(theta_hat_boot - 1.645 * se_theta_hat_boot, theta_hat_boot + 1.645 * se_theta_hat_boot))
```

(E)Use large sample method to estimate SE and the 90% CI for $\hat{\theta}$

We know that Standard Error is given by the following: $SE = \sqrt{pq/n}$
where $\hat{p}$ and $\hat{q}$ are the sample frequencies of the M and N alleles respectively, n is the sample size

The 90% confidence interval is calculated as follows: $(\hat{\theta} - (1.645 * SE), \hat{\theta} + (1.645 * SE))$
where 1.645 is the Z score corresponding to a 90% CI

```
# Sample frequencies of M, MN, and N blood types
    f_M <- 342 / 1029
f_MN <- 500 / 1029
    f_N <- 187 / 1029

# Sample size
    n <- 1029

# Estimate of theta
    theta_hat <- f_MN

# Estimate of p and q
p_hat <- 2 * f_M + f_MN
q_hat <- 2 * f_N + f_MN

# Standard error
SE <- sqrt((p_hat * q_hat) / n)

# 90% confidence interval
CI <- c(theta_hat - 1.645 * SE, theta_hat + 1.645 * SE)

# Print results
    cat("Standard Error:", SE, "\n")
```

```
## Standard Error: 0.03081829
```

```
cat("90% Confidence Interval:", CI, "\n")
```

```
## 90% Confidence Interval: 0.4352126 0.5366047
```

Contributions in the project

Question 1 : Laksh Gupta Question 2 : Vinayak Manoj Question 3 : Kshitij Shekhar

Report : Kshitij Shekhar