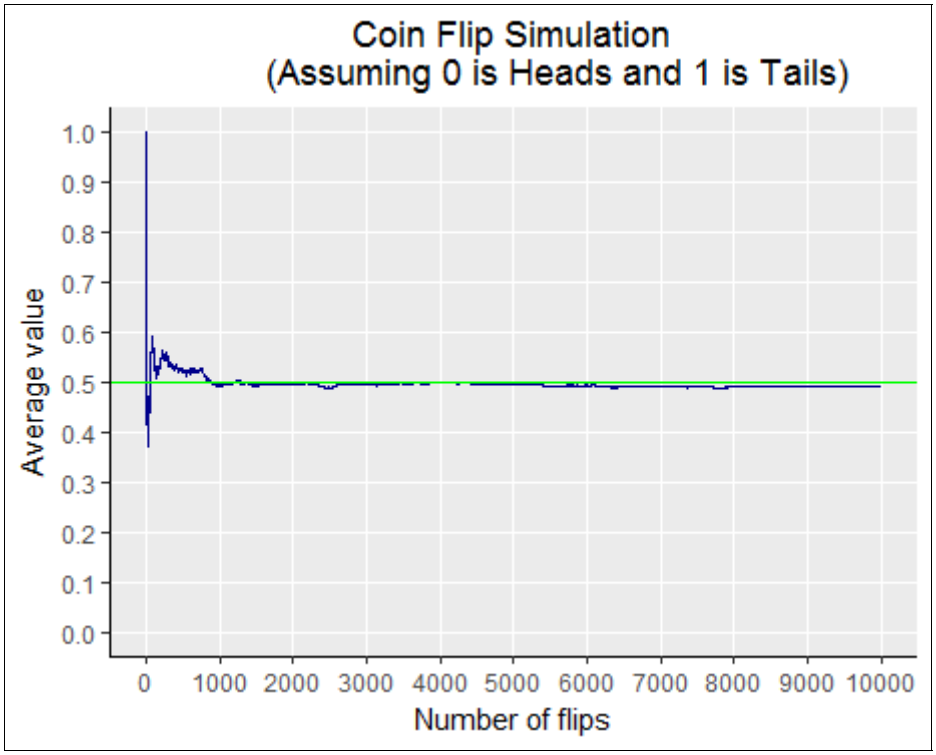
ASSIGNMENT #03
PLOTS AND EXPLANATION

SOLUTION 1.

Command:

```
> CoinFlip()
[1] "The proportion of tails is 0.4916"
```

Solution Plot



Explanation:

[Assumptions -

Total trials: 10000

Heads is 0 and Tails is 1]

The plot starts from trial 1 and has a total of 10000 trials. The blue line represents the cumulative average value till that trial and the green line represents the theoretical value that is expected (= 0.5) as both heads and tails are equally likely events.

Observation:

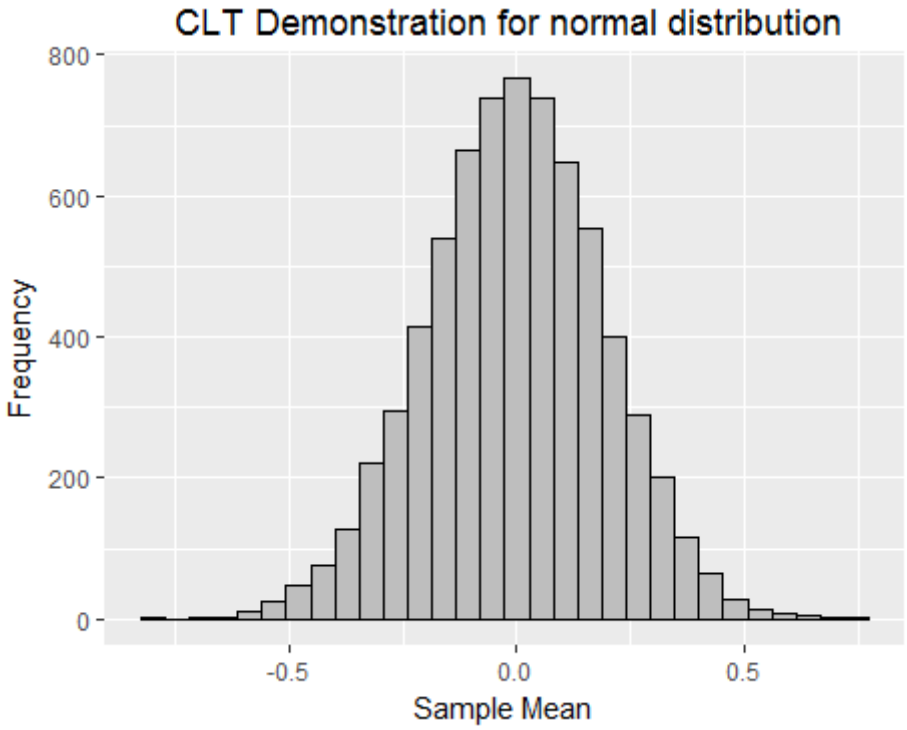
It is evident from the plot that as the number of trails/flips increases, the value approaches 0.5, i.e., the theoretical value. So, it can be concluded that for countably infinite trials, the value will be 0.5

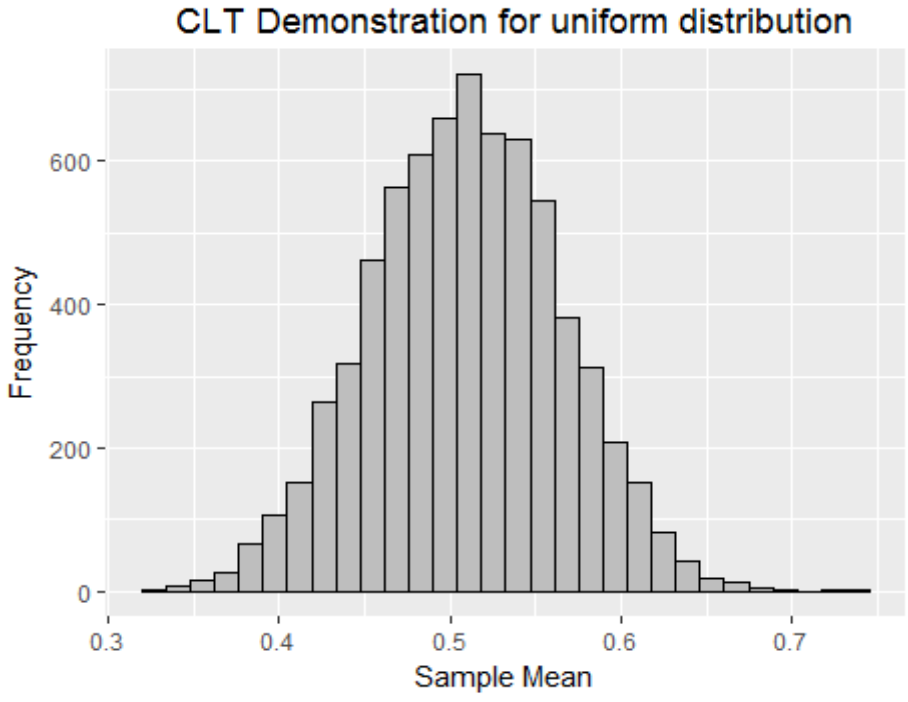
SOLUTION 2.

Command:

```
> CLT("normal",27,7000)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
$mean
[1] -0.004870407

$se
[1] 0.1952874

> CLT("uniform",27,7000)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
$mean
[1] 0.5086828

$se
[1] 0.0556217
```

Solution Plot





Explanation:

[Assumptions -

Population Size = 10000

Types of distribution = normal or uniform]

To demonstrate CLT, I have assumed a population of 10000 elements and then the program generates suitable random population based on the type of distribution passed as argument to the function.

Once, the population is generated, samples for the given size are taken using *sample()* function and mean of these values is calculated.

Now, according to CLT, the mean of each samples should follow a normal distribution. Hence, this can be checked by using *qqnorm(sample_mean);qqline(sample_mean, col = 2)* code which is commented out in the code as it was not required for output.

Mean and standard error of these sample means are returned as output of this function.

Observations:

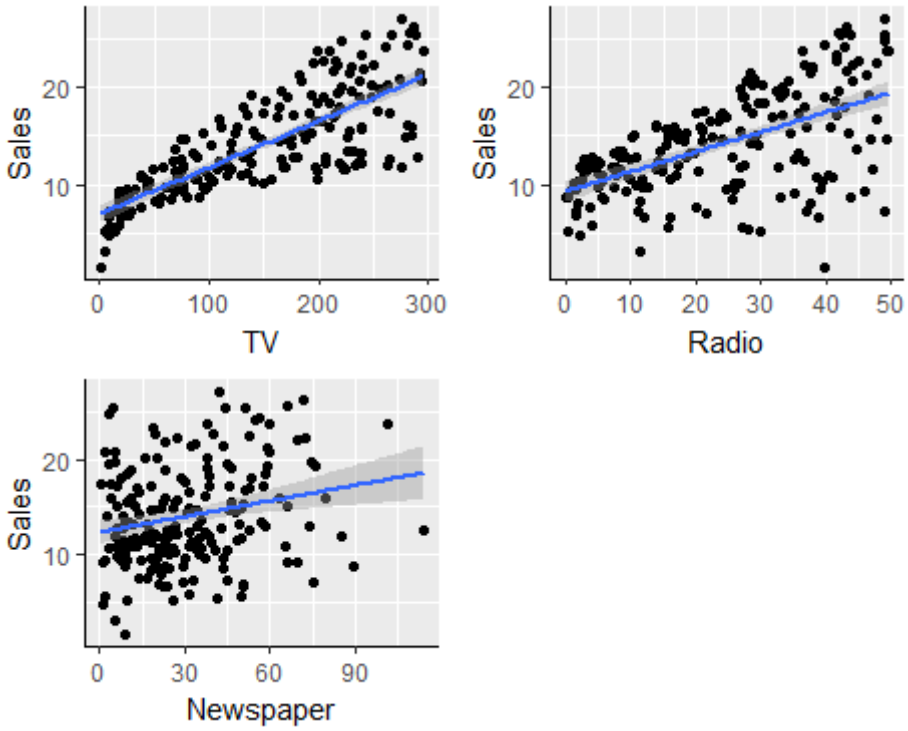
It is clear from the plots above that they follow normal distribution and hence CLT is verified.

SOLUTION 3(a)

Command:

```
> SLR()
[1] "TV"
```

Solution Plot:



Explanation:

ggplot library is used to generate the plot from the imported dataset. There are 3 variables, *TV, Radio and Newspaper*, which affect the *Sales* (dependent variable). For this part, linear regression line is plotted for each of the 3 variables against the dependent variable.

Observation:

Slope for regression line for *TV* is the highest and hence it is the best covariate among all three variable, this is proved by the output as well, as it has the highest correlation value.

SOLUTION 3(b)

Command:

```
> MLR()
$Intercept
(Intercept)
  2.938889

$TVCoeff
    TV
0.04576465

$NewspaperCoeff
 Radio
0.18853

$RadioCoeff
  Newspaper
-0.001037493
```

Solution Equation:

$y\text{-}hat = 2.938889 + 0.045 * x_1 + 0.1885 * x_2 - 0.001 * x_3$

Explanation:

Multiple Linear Regression is Linear Regression with multiple number of variables and hence we take all the 3 variables for this. The formula now becomes *Sales ~ TV + Radio + Newspaper.* Therefore, there are 3 coefficients, one for each variable which is fetch using *coef()* function and the values are returned to the function.

Comparison of β:

Values for Simple Linear Regression

```
> coef(lm(Sales~TV,data = advt_data))
(Intercept)          TV
 7.03259355  0.04753664
> coef(lm(Sales~Radio,data = advt_data))
(Intercept)       Radio
 9.3116381   0.2024958
> coef(lm(Sales~Newspaper,data = advt_data))
(Intercept)   Newspaper
 12.3514071   0.0546931
```

It can be seen that the intercept value is lowest for multiple linear regression model as compared to the value of intercepts in simple linear regression model. Also, since *newspaper* has the lowest correlation with *Sales*, the difference between its value in SLR and MLR is the highest while *TV* and *Radio* have less deviation as compared to newspaper.

SOLUTION 4

Command:

```
> LogisticRegression()
$Intercept
(Intercept)
 -10.15345

$X1Coeff
    X1
0.3312469

$X2Coeff
    X2
0.1808757

$X3Coeff
   X3
5.087466
```

Explanation:
glm() function is used to find the logistic regression and the formula has all the variables (*X1,X2,X3*) on which the value of *Y* is dependent. Here also, *coef()* function is used to extract the coefficients from the resulting logistic model.

SOLUTION 5

Command:

```
> LogisticRegressionImproved()
```

Explanation:
The same function *glm()* is used and accuracy is measured as the number of correct predictions divided by the total number of predictions. Predicting 1 when actual value is 1 and 0 when actual value is 0 are considered as the correct predictions.
In the new model, the resultant accuracy is much higher than the original accuracy.

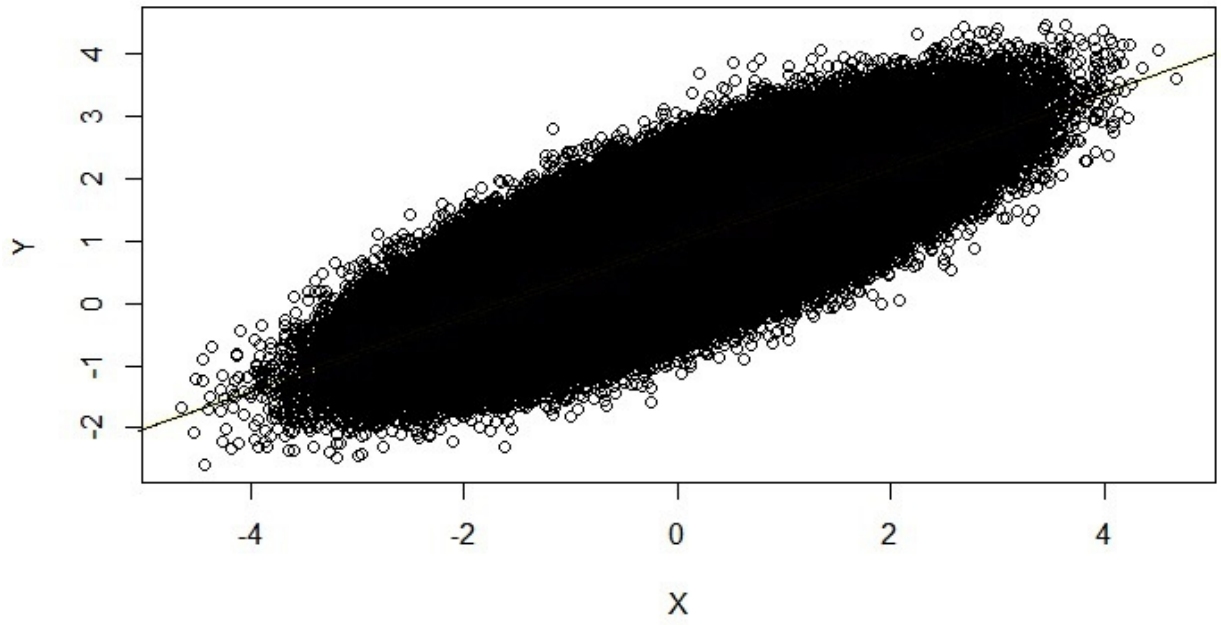
Observation:
The accuracy of the new model is higher than the original model. Hence, the modified one is the better.

SOLUTION 6

Command:

```
> BigSLR()
```

Plot:



Explanation:

Since, this is a large dataset, so it is not feasible to load it with *read.csv()* function, hence *fread()* function present in *data.table* is used for importing this dataset. Now, *lm()* function is again not feasible for this dataset as it takes a lot of time to execute. b*iglm()* function is used for linear regression model for this purpose which is designed to handle large datasets.

For sampling 1%, 2%, 3%, 4% and 5% of the dataset, we use *sample()* function to take those samples and since now the samples are within the capacity of *lm()*, we use it to generation linear regression model for these samples.