Question: Using R execute the basic commands, array, list and frames.
Solution:

Array:

```
> x <- array(1:20, dim = c(4,5))
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

List:

```
> x <- list(a = 1:4, b = letters[1:4], c = c("red", "blue", "green"))
> x
$a
[1] 1 2 3 4

$b
[1] "a" "b" "c" "d"

$c
[1] "red"  "blue" "green"
```

Data Frame:

```
> x <- data.frame(name = c("John", "Paul", "George", "Ringo"), age = c(21, 22, 23, 24))
> x
   name age
1  John  21
2  Paul  22
3 George  23
4 Ringo  24
```

Question:
Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operati ons.
Solution:

```
# Create a Matrix

matrix1 <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)

# Addition

matrix2 <- matrix(c(5,6,7,8), nrow = 2, ncol = 2)
matrix_add <- matrix1 + matrix2
print(matrix_add)

# Output
     [,1] [,2]
```

```
[1,]    6    8
[2,]   10   12
```

```
# Inverse

matrix_inv <- solve(matrix1)
print(matrix_inv)

# Output
          [,1]      [,2]
[1,] -2.000000  1.000000
[2,]  1.500000 -0.500000

# Transpose

matrix_transpose <- t(matrix1)
print(matrix_transpose)

# Output
     [,1] [,2]
[1,]    1    3
[2,]    2    4

 # Multiplication

 matrix_multiply <- matrix1 %*% matrix2
 print(matrix_multiply)

 # Output

      [,1] [,2]
[1,]   23   31
[2,]   34   46
```

Question:
Using R Execute the statistical functions:mean, median, mode, quartiles, range, inter quartile range histog
ram
Solution:

Mean:

```
mean(x)
```

Median:

```
median(x)
```

Mode:

```
mode(x)
```

Quartiles:

```
quantile(x)
```

Range:

```
range(x)
```

Inter Quartile Range:
IQR(x)

Histogram:
hist(x)

Question:
Using R import the data from Excel / .CSV file and Perform the above functions.
Solution:

```
# Importing the data from Excel / .CSV file

data <- read.csv("data.csv")

# Calculating the mean of the data

mean(data$value)

# Calculating the median of the data

median(data$value)

# Calculating the mode of the data

mode(data$value)
```

Question:
Using R import the data from Excel / .CSV file and Calculate the standard deviation, variance, co-variance
.
Solution:

To import data from an Excel or .CSV file into R, you can use the read.csv() or read.xlsx() functions. For e
xample, to import a .csv file called "data.csv" into R, you would use the following code:

```
data <- read.csv("data.csv")
```

Once the data is imported, you can calculate the standard deviation, variance, and co-variance using the f
ollowing functions:

Standard Deviation: sd(data)
Variance: var(data)
Covariance: cov(data)

Question:
Using R import the data from Excel / .CSV file and draw the skewness.
Solution:

To import data from an Excel or .CSV file into R, you can use the read.csv() or read.xlsx() functions. For e

xample:

data <- read.csv("mydata.csv")

Once the data is imported, you can use the skewness() function to calculate the skewness of the data. Fo r example:

skewness(data)

Question:
Import the data from Excel / .CSV and perform the hypothetical testing using R.
Solution:

To import the data from Excel / .CSV and perform the hypothetical testing using R, you can use the read.c sv() function. This function will read in the data from the file and store it in a data frame. Once the data is stored in a data frame, you can then use the t.test() function to perform the hypothetical testing. The t.test () function will take two vectors as arguments and will return a p-value that can be used to determine if the null hypothesis is accepted or rejected.

Question:
Import the data from Excel / .CSV and perform the Chi-squared Test using R.
Solution:

The Chi-squared Test is a statistical test used to determine if there is a significant difference between two or more groups. It is used to compare observed data with expected data.

To perform the Chi-squared Test using R, first import the data from Excel or .CSV into R. This can be don e using the read.csv() function.

Once the data is imported, the chisq.test() function can be used to perform the Chi-squared Test. This fun ction takes two arguments: the observed data and the expected data. The observed data should be in a matrix or data frame format, while the expected data should be in a vector format.

For example, if we have a dataset with two columns (A and B) and 10 rows of data, we can use the followi ng code to perform the Chi-squared Test:

observed <- as.matrix(read.csv("data.csv"))
expected <- c(5, 5)
chisq.test(observed, expected)

Question:
Using R perform the binomial and normal distribution on the data.
Solution:

For the binomial distribution, you can use the dbinom() function in R. This function takes three arguments: the number of successes, the probability of success, and the number of trials. For example, if you wanted to calculate the probability of getting 3 successes out of 10 trials with a probability of success of 0.5, you would use the following code:

dbinom(3, 10, 0.5)

For the normal distribution, you can use the dnorm() function in R. This function takes three arguments: th e mean, standard deviation, and value for which you want to calculate the probability. For example, if you wanted to calculate the probability of getting a value of 5 with a mean of 10 and a standard deviation of 2, you would use the following code:

dnorm(5, 10, 2)

Question:
Perform the Linear Regression using R.
Solution:

Linear regression is a statistical technique used to model the relationship between a dependent variable (also known as the response variable) and one or more independent variables (also known as explanatory variables).

In R, linear regression can be performed using the lm() function. The syntax for this function is:

lm(formula, data)

Where formula is a formula specifying the model to be fit and data is the data frame containing the variabl es in the model.

For example, if we have a dataset with two variables, x and y, we can fit a linear regression model using t he following command:

lm(y ~ x, data = mydata)

Question:
Compute the Least squares means using R.
Solution:

The least squares means can be computed in R using the lm() function. This function takes a formula and a data frame as arguments and returns the least squares means for the specified model. For example, to compute the least squares means for a linear regression model with two predictor variables, x1 and x2, t he following code can be used:

```
model <- lm(y ~ x1 + x2, data = mydata)
least_squares_means <- summary(model)$coefficients[,1]
```

Question:
Compute the Linear Least Square Regression using R.
Solution:

Linear least squares regression is a method for fitting a linear model to a set of data points. It is used to fi nd the best-fitting line for a given set of data points by minimizing the sum of the squares of the vertical di stances between each data point and the line.

In R, linear least squares regression can be performed using the lm() function. The syntax for this function is:

lm(formula, data)

where formula is an R formula specifying the model to be fit and data is an optional data frame containing the variables in the model.

For example, to fit a linear model to a set of x and y values, we could use the following code:

```
x <- c(1,2,3,4,5)
y <- c(2,4,6,8,10)
model <- lm(y ~ x, data = data.frame(x, y))
summary(model)
```