1) Here is an example of how to perform basic R operations including data input, handling missing values, and importing data from different formats:
Data input:

```
# Input data using the c() function
x <- c(1, 2, 3, 4, 5)

# View the data
x

#Handling missing values:

# Input data with missing values
y <- c(1, NA, 3, 4, NA)

# View the data with missing values
y

# Remove missing values
y_clean <- y[!is.na(y)]

# View the data without missing values
y_clean
```

2) Importing data into R using different formats:

```
# Import data from a .xlsx file
library(readxl)
data_xlsx <- read_excel("data.xlsx")

# View the imported data
head(data_xlsx)

# Import data from a .csv file
data_csv <- read.csv("data.csv")

# View the imported data
head(data_csv)

# Import data from a .txt file
data_txt <- read.table("data.txt")

# View the imported data
head(data_txt)
```

3 Creating dataframes:

```
#From scratch
df <- data.frame(
  Name = c("Gaurav", "Maya", "Prakash"),
  Age = c(30, 28, 35),
  Gender = c("Male", "Female", "Male")
)

#From a matrix
mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
df2 <- as.data.frame(mat)
```

```
colnames(df2) <- c("A", "B", "C")

#From a list
my_list <- list(A = c(1, 2), B = c("A", "B"))
df3 <- as.data.frame(my_list)
```

b) Indexing and selecting data

```
# Selecting a specific row
df[2, ]

# Selecting a specific column
df$Age

# Selecting multiple columns
df[, c("Name", "Age")]

# Selecting rows where a condition is true
df[df$Age > 30, ]
```

c) Adding, removing, and renaming columns:

```
# Adding a new column
df$City <- c("NYC", "LA", "Chicago")

# Removing a column
df$City <- NULL

# Renaming a column
colnames(df) <- c("Full Name", "Years", "Sex")
```

d) Sorting data:

```
# Sorting by a single column
df[order(df$Years), ]

# Sorting by multiple columns
df[order(df$Sex, df$Years), ]
```

e Grouping and aggregating data:

```
library(dplyr)

# Grouping by a single column
df %>% group_by(Sex) %>% summarize(mean_age = mean(Years))

# Grouping by multiple columns
df %>% group_by(Sex, Years) %>% summarize(count = n())
```

3) Create the contingency table for the given raw data in R

```
# Sample data
raw_data <- c("A", "A", "B", "B", "A", "B", "C", "C", "C")

# Creating the contingency table
table(raw_data)
# Output:
# raw_data
# A B C
# 3 3 3
```

In this example, the contingency table shows the frequency of each unique value in the raw_data vector. The result indicates that the values "A", "B", and "C" each appear 3 times in the data.
Convert the contingency table to original format of the given data. in R

To convert a contingency table back to its original format in R, you can use the unlist function along with rep to repeat the values according to their frequencies in the contingency table. Here's an example:

```
# Contingency table

contingency_table <- table(c("A", "A", "B", "B", "A", "B", "C", "C", "C"))

# Converting the contingency table back to the original format

original_data <- unlist(lapply(names(contingency_table), function(x) rep(x, contingency_table[x])))
```

In this example, contingency_table is the input contingency table, and original_data is the output vector that represents the original data format. The lapply function is used to loop through the names of the contingency table, and the rep function is used to repeat each value according to its frequency in the contingency table. The unlist function is then used to convert the resulting list into a vector.

4) Analyse and give interpretation of summary statistics for the given data.

To get a summary of statistics for a given data set in R, you can use the summary function. The summary function provides a variety of information about the data, including mean, median, minimum, maximum, and quartiles. Here's an example:

```
# Sample data
data <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Getting the summary statistics
summary(data)
# Output:
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#    1.00    3.25    5.50    5.50    7.75   10.00
```

In this example, the summary statistics give you a quick overview of the data distribution. For example, the mean and median are both 5.5, which suggests that the data are symmetrically distributed around the center. The 1st quartile and 3rd quartile values (3.25 and 7.75, respectively) give you an idea of the spread of the data. The minimum value is 1 and the maximum value is 10, which gives you a sense of the range of the data.
It's important to remember that the summary function provides a basic overview of the data, but it is not a complete analysis. To perform a more in-depth analysis of the data, you may need to use other functions and techniques, such as plotting the data, calculating standard deviations, and performing hypothesis tests.

5) Calculate mean, median and mode for the grouped data and compare the results for the given data.

To calculate the mean, median, and mode in R, you can use the mean, median, and table functions, respectively. Here's an example:

```
data <- c(7, 9, 10, 10, 11, 12, 13, 14, 14, 15, 15, 16, 16, 17, 18, 19)

#We can create a frequency table using the table() function:

freq_table <- table(data)
freq_table
#Now, we can use the weighted.mean() function to calculate the mean:

mean <- weighted.mean(as.numeric(names(freq_table)), as.numeric(freq_table))
mean
```

To calculate the median, we first need to find the cumulative frequency of the data. We can do this using the cumsum() function:

```
cum_freq <- cumsum(freq_table)
cum_freq
#Now, we can use the which() function to find the index of the median value:

median_index <- which(cum_freq >= sum(freq_table)/2)[1]
median_index
#Finally, we can find the median value by taking the average of the value at the median index and the value at the previous index:

median <- mean(as.numeric(names(freq_table)[median_index-1:median_index]))
median

#To calculate the mode, we can use the which.max() function to find the index of the maximum frequency:
mode_index <- which.max(freq_table)
mode_index
#Finally, we can find the mode value by taking the value at the mode index:

mode <- as.numeric(names(freq_table)[mode_index])
mode
```

Comparing the results, we can see that the mean is ......, the median is ...., and the mode is .... This is because the data is slightly skewed to the right, with a longer tail on the right side. As a result, the mean is pulled slightly higher than the median, but both are still fairly close to each other. The mode is lower than both the mean and median, indicating that there is no clear peak in the data.

6) Use R for test the given data In order to compare the effectiveness of two sources of nitrogen, namely ammonium chloride (NH4Cl) and urea, on grain yield of Coarse cereal, an experiment was conducted. The results on the grain yield of Coarse Cereal (kg/plot) under the two treatments are given below.
NH4Cl : 13.4, 10.9, 11.2, 11.8, 14.0, 15.3, 14.2, 12.6, 17.0, 16.2, 16.5, 15.7.
Urea : 12.0, 11.7, 10.7, 11.2, 14.8, 14.4, 13.9, 13.7, 16.9, 16.0, 15.6, 16.0.
Assess which source of nitrogen is better for Coarse Cereal.

To assess which source of nitrogen is better for Coarse Cereal, we can compare the mean grain yield for each treatment. Here's how you can do this in R:

```
# Define the grain yield data for each treatment
NH4Cl <- c(13.4, 10.9, 11.2, 11.8, 14.0, 15.3, 14.2, 12.6, 17.0, 16.2, 16.5, 15.7)
Urea <- c(12.0, 11.7, 10.7, 11.2, 14.8, 14.4, 13.9, 13.7, 16.9, 16.0, 15.6, 16.0)


# Calculate the mean for each treatment
mean_NH4Cl <- mean(NH4Cl)
mean_Urea <- mean(Urea)

# Compare the means
if (mean_NH4Cl > mean_Urea) {
  print("NH4Cl is better for Coarse Cereal")
} else if (mean_Urea > mean_NH4Cl) {
  print("Urea is better for Coarse Cereal")
} else {
  print("Both sources of nitrogen are equally effective for Coarse Cereal")
}
```

This code calculates the mean grain yield for each treatment, then compares the means and prints a statement indicating which source of nitrogen is better. However, it's important to keep in mind that this is just a simple comparison based on the mean and doesn't take into account other factors that may influence the grain yield, such as soil type, weather conditions, and so on. Further statistical analysis may be necessary to fully assess the effectiveness of the two sources of nitrogen

```
t.test(NH4Cl,Urea,paired=TRUE,alt="less")

#INTERPRETATION:The p-value (0.753) is greater than the level of significance (0.05).
#So we do not reject our null hypothesis. Hence, the ammonium chloride
#is the better nitrogen source for paddy.
```

7) Before an increasing in exercise duty on tea, 800 persons out of a sample of 1000 persons were found to be tea drinkers. After an increasing in duty, 800 people were tea drinkers in a sample of 1200 people. Using SE of a proportion, state whether there is a significant decrease in consumption of tea after the increase in the exercise duty. in R

To determine whether there is a significant decrease in consumption of tea after the increase in the exercise duty, you can use the standard error (SE) of a proportion. The SE of a proportion can be used to calculate a confidence interval for the difference between two proportions. If the confidence interval for the difference does not include 0, it can be concluded that there is a significant difference between the two proportions. Here's how you can do this in R:

```
# Define the sample sizes and proportions
n1 <- 1000
p1 <- 800/1000
n2 <- 1200
p2 <- 800/1200

# Calculate the SE of the difference between the two proportions
SE_diff <- sqrt(p1 * (1 - p1)/n1 + p2 * (1 - p2)/n2)

# Calculate the confidence interval for the difference between the two proportions
CI_diff_lower <- p1 - p2 - qnorm(0.975) * SE_diff
CI_diff_upper <- p1 - p2 + qnorm(0.975) * SE_diff

# Test for a significant difference
if (CI_diff_lower > 0 || CI_diff_upper < 0) {
  print("There is a significant decrease in consumption of tea after the increase in the exercise duty")
}
else {
  print("There is not a significant decrease in consumption of tea after the increase in the exercise duty")
}
```

This code calculates the SE of the difference between the two proportions, and then calculates the 95% confidence interval for the difference. If the confidence interval for the difference does not include 0, it can be concluded that there is a significant decrease in consumption of tea after the increase in the exercise duty.

To determine if there is a significant decrease in consumption of tea after the increase in exercise duty, you can perform a hypothesis test using the standard error (SE) of a proportion.
Here's how you can perform this test in R using the prop.test function:

```
# Define the proportions of tea drinkers before and after the increase in duty
p1 <- 800/1000
p2 <- 800/1200

# Perform the hypothesis test using the prop.test function
result <- prop.test(c(800, 800), c(1000, 1200), correct = FALSE)

# Extract the p-value from the result
p_value <- result$p.value

# Check if the p-value is less than the significance level (0.05)
if (p_value < 0.05) {
  print("There is a significant decrease in consumption of tea after the increase in the exercise duty.")
}
else {
  print("There is not a significant decrease in consumption of tea after the increase in the exercise duty.")
}
```

The prop.test function takes three arguments: the number of successful trials (tea drinkers), the number of total trials (sample size), and correct = FALSE to indicate that the sample proportions are not corrected for continuity.
The result object contains various test statistics, such as the estimated difference in proportions, the standard error of the difference, and the p-value. If the p-value is less than the significance level (0.05), you can reject the null hypothesis that there is no difference in the proportions of tea drinkers before and after the increase in duty, and conclude that there is a significant decrease in consumption of tea after the increase in the exercise duty.

8) Use R for test the given data
A health status survey in a few villages revealed that the normal serum protein value of children in that locality is 7.0 g/100ml. A group of 16 children who received high protein food for a period of six months
had serum protein values shown below. Can we consider that the mean serum protein level of those who
were fed on high protein diet is different from that of the general population?

S.No. (Child No.)	1	2	3	4	5	6	7	8
Protein level (g%)	7.10	7.70	8.20	7.56	7.05	7.08	7.21	7.25
S.No. (Child No.)	9	10	11	12	13	14	15	16
Protein level (g%)	7.36	6.59	6.85	7.90	7.27	6.56	7.93	8.56

To determine if the mean serum protein level of the children who received a high protein diet is significantly different from the normal level of 7.0 g/100ml in the general population, you can perform a t-test.
Here's how you can perform the t-test in R:

```
# Define the protein levels of the children who received the high protein diet
protein_levels <- c(7.10, 7.70, 8.20, 7.56, 7.05, 7.08, 7.21, 7.25, 7.36, 6.59, 6.85, 7.90, 7.27, 6.56, 7.93,
8.56)

# Perform the t-test using the t.test function
result <- t.test(protein_levels, mu = 7.0)

# Extract the p-value from the result
p_value <- result$p.value
```

```
# Check if the p-value is less than the significance level (0.05)
if (p_value < 0.05) {
  print("We can consider that the mean serum protein level of those who were fed on high protein diet is different from that of the general population.")
} else {
  print("We cannot consider that the mean serum protein level of those who were fed on high protein diet is different from that of the general population.")
}
```

The t.test function takes two arguments: the numeric vector protein_levels representing the serum protein levels of the children who received the high protein diet, and mu = 7.0 representing the population mean of 7.0 g/100ml.
The result object contains various test statistics, such as the estimated mean difference, the standard error of the mean difference, and the t-value. If the p-value is less than the significance level (0.05), you can reject the null hypothesis that the mean serum protein level of the children who received a high protein diet is equal to the normal level of 7.0 g/100ml in the general population, and conclude that the mean serum protein level is significantly different.

9) Students were selected to training. Their performance was noted by giving a test and the marks recorded
out of 50. They were given effective 6 months training and again they were given a test and marks were
recorded out of 50.
Students 1 2 3 4 5 6 7 8 9 10
Before training 25 20 35 15 42 28 26 44 35 48
After training 26 20 34 13 43 40 29 41 36 46
By applying the t-test can it be concluded that the students have benefited by the training?

To determine if the students have benefited from the training, you can perform a paired sample t-test. This test compares the mean difference in the test scores before and after the training, to determine if there is a significant improvement.

Here's how you can perform the paired sample t-test in R:

```
# Define the test scores before and after the training
before_training <- c(25, 20, 35, 15, 42, 28, 26, 44, 35, 48)
after_training <- c(26, 20, 34, 13, 43, 40, 29, 41, 36, 46)

# Perform the paired sample t-test using the t.test function
result <- t.test(after_training - before_training, mu = 0, alternative = "greater")

# Extract the p-value from the result
p_value <- result$p.value

# Check if the p-value is less than the significance level (0.05)
if (p_value < 0.05) {
  print("We can conclude that the students have benefited by the training.")
} else {
  print("We cannot conclude that the students have benefited by the training.")
}
```

The t.test function takes two arguments: the numeric vector after_training - before_training representing the difference in the test scores, and mu = 0 representing the expected mean difference of 0 if there is no improvement. The argument alternative = "greater" indicates that we are testing the alternative hypothesis that the mean difference is greater than 0, implying that the students have benefited from the training.
The result object contains various test statistics, such as the estimated mean difference, the standard error of the mean difference, and the t-value. If the p-value is less than the significance level (0.05),

you can reject the null hypothesis that the mean difference is equal to 0, and conclude that the students have benefited from the training.

11)100 individuals of a particular race were tested with an intelligence test and classified into two classes.
Another group of 120 individuals belong to another race were administered the same intelligence test and classified into the same two classes. The following are the observed frequencies of the two races:
Race Intelligence
Intelligent Non-intelligent Total
Race I 42 58 100
Race II 55 65 120
Total 97 123 220
Test whether the intelligence is anything to do with the race.

To test whether the intelligence is associated with the race, you can perform a chi-squared test for independence. The chi-squared test compares the observed frequencies in the contingency table to the expected frequencies under the assumption of independence between the two variables.
Here's how you can perform the chi-squared test in R:

```
# Define the contingency table
table <- matrix(c(42, 58, 55, 65), nrow = 2, byrow = TRUE)
colnames(table) <- c("Intelligent", "Non-intelligent")
rownames(table) <- c("Race I", "Race II")

# Perform the chi-squared test using the chisq.test function
result <- chisq.test(table)

# Extract the p-value from the result
p_value <- result$p.value

# Check if the p-value is less than the significance level (0.05)
if (p_value < 0.05) {
  print("There is evidence that the intelligence is associated with the race.")
} else {
  print("There is no evidence that the intelligence is associated with the race.")
}
```

The chisq.test function takes the contingency table as an argument and returns the test statistics, including the chi-squared statistic, the degrees of freedom, and the p-value. If the p-value is less than the significance level (0.05), you can reject the null hypothesis that the intelligence and the race are independent, and conclude that there is evidence that the intelligence is associated with the race.

14) Obtain the correlation coefficient between the heights of father(X) and of the son (Y) from the following
data
X 65 66 67 68 69 70 71 72
Y 67 68 65 68 72 72 69 71
And also test its significance. Using R functions.

To obtain the correlation coefficient between the heights of the father (X) and the son (Y), you can use the cor function in R. The cor function calculates the Pearson correlation coefficient, which measures the linear relationship between two variables.
Here's how you can use the cor function in R:

```
# Define the data
x <- c(65, 66, 67, 68, 69, 70, 71, 72)
y <- c(67, 68, 65, 68, 72, 72, 69, 71)

# Calculate the correlation coefficient
```

```
correlation <- cor(x, y)

# Print the correlation coefficient
cat("The correlation coefficient is", correlation, "\n")
```

The output of the above code will give you the correlation coefficient between X and Y.
To test the significance of the correlation coefficient, you can perform a hypothesis test to see if the correlation coefficient is significantly different from zero. This test is known as the Pearson's correlation significance test. In R, you can use the cor.test function to perform this test.
Here's how you can perform the significance test in R:

```
# Perform the significance test using the cor.test function
result <- cor.test(x, y)

# Extract the p-value from the result
p_value <- result$p.value

# Check if the p-value is less than the significance level (0.05)
if (p_value < 0.05) {
  print("The correlation is significant.")
}
else {
  print("The correlation is not significant.")
}
```

The cor.test function takes the two variables as arguments and returns the test statistics, including the correlation coefficient, the p-value, and the degrees of freedom. If the p-value is less than the significance level (0.05), you can reject the null hypothesis that the correlation coefficient is equal to zero and conclude that the correlation is significant.

15) Consider the inbuilt data set cars.
Find Correlation between possible variables and pairwise correlation
Find regression line between appropriate variables
Display the summary statistics and comment on the results

The cars data set in R contains the speed and stopping distances of 50 cars. Here's how you can perform the steps you mentioned in your question:
Find the correlation between possible variables and pairwise correlation:

```
# Load the cars data set
data(cars)

# View the data
head(cars)

# Calculate the pairwise correlation matrix
cor_matrix <- cor(cars)

# View the correlation matrix
cor_matrix
```

Find the regression line between appropriate variables:

```
# Fit a linear regression model to predict stopping distance from speed
model <- lm(dist ~ speed, data = cars)

# Summarize the model
```

```
summary(model)
#Display the summary statistics and comment on the results:
r

# Extract the summary statistics
model_summary <- summary(model)

# View the summary statistics
model_summary
```

The summary function returns a variety of statistics about the model fit, including the coefficients, residuals, goodness of fit, and more. The most important statistics for evaluating the model are the coefficient of determination (R-squared) and the p-value for the speed variable.

The R-squared value measures the proportion of variance in the dependent variable (stopping distance) that is explained by the independent variable (speed). A high R-squared value indicates that the model fits the data well.

The p-value for the speed variable tests the null hypothesis that the slope of the regression line is equal to zero. If the p-value is less than the significance level (0.05), you can reject the null hypothesis and conclude that the speed has a significant effect on the stopping distance.

Based on the results of the summary statistics, you can conclude that there is a strong, positive relationship between speed and stopping distance. The speed variable has a significant effect on the stopping distance, as evidenced by the high R-squared value and the low p-value.