* **To perform basic commands on Ubuntu operating system.**

|  |  |
| --- | --- |
| **Linux Commands** | **Functions** |
| Is | Displays information about files in the current directory. |
| pwd | Displays the current working directory. |
| mkdir | Creates a directory. |
| cd | To navigate between different folders. |
| rmdir | Removes empty directories from the directory lists. |
| cp | Moves files from one directory to another. |
| mv | Rename and Replace the files |
| rm | Delete files |
| touch | Create empty files |
| cat | Display file contents on terminal |
| clear | Clear terminal |
| ps | Display the processes in terminal |
| echo | Display active processes on the terminal |
| wget | download files from the internet. |
| whoami | Create or update passwords for existing users |
| cal | View Calendar in terminal |

**R Programming**

To implement basic functions and commands in R Programming. better visualization than a data table.

**# Taking Input from user & display it.**

Name <- readline(prompt = "Enter Your Name :")

> Age <- readline(prompt = "Enter Your Age :")

> print(paste("Hi,", Name, "Welcome to the world of R Programming"))

**# Take user input & perform arithmetic operations.**

> Num1 <- as.numeric(readline(prompt = "Enter the 1st Number : "))

> Num2 <- as.numeric(readline(prompt = "Enter the 2nd Number : "))

> Addition <- Num1 + Num2

> print(Addition)

|  |  |
| --- | --- |
| abs() | calculates a number’s absolute value. |
| sqrt() | calculates a number’s square root. |
| round() | rounds a number to the nearest integer. |
| exp() | calculates a number’s exponential value |
| log() | which calculates a number’s natural logarithm. |
| cos(), sin(), and tan() | calculates a number’s cosine, sine, and tang. |
| mean() | A vector’s arithmetic mean is determined by the mean() function |
| median() | A vector’s median value is determined by the median() function. |

**# Create the data for the chart**

> Runs <- c(778, 678, 567, 789, 697)

> Players <- c('Rohit Sharma', 'Virat Kohli', 'S Gill', 'Ishan Kishan', 'SKY')

**> # Plot the simple bar chart**

> barplot(Runs, names.arg = Players, xlab = 'Players', ylab = 'Runs', main = "CricStat")

**> # Plot the coloured bar chart**

> barplot(Runs, names.arg = Players, xlab = 'Players', ylab = 'Runs', main = "CricStat", col = 'Green')

**> # Plot the horizontal bar chart**

> barplot(Runs, names.arg = Players, xlab = 'Players', ylab = 'Runs', main = "CricStat", col = 'Green', horiz = TRUE)

**> # Plot the bar chart with data labels on top of each bar**

> barplot(Runs, names.arg = Players, xlab = 'Players', ylab = 'Runs', main = "CricStat", col = 'Green', horiz = TRUE, cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)

**> # Add data labels on top of each bar**

> text(x = barplot(Runs, names.arg = Players, col = 'Green', ylim = c(0, max(Runs)\*1.2)), y = Runs + 1, labels = Runs, pos = 3, cex = 1.2, col = 'Red') >

To install Hadoop framework, configure it and setup a single node cluster.

<https://codewitharjun.medium.com/install-hadoop-on-ubuntu-operating-system-6e0ca4ef9689>

To implement a word count application using the MapReduce API.

<https://codewitharjun.medium.com/word-count-example-using-hadoop-and-java-8ef3d665e331>

To implement file management tasks in Hadoop HDFS like adding, retrieving and deleting.

<https://www.geeksforgeeks.org/hdfs-commands/>

Hdfs dfs - ls /

Hdfs dfs - mkdir

Hdfs dfs - touchz <file path>

Hdfs dfs - copyFromLocal <local file path> <dest(present on hdfs)>

Hdfs dfs - cat <path>

Hdfs dfs - copyToLocal <<srcfile(on hdfs)> <local file dest>

Hdfs dfs - moveFromLocal <local src> <dest(on hdfs)>