Follow this github link for future projects and codes.
https://github.com/Avijit1992/NLP100-R

Session 1

```
#Create string
print("Hello World")

#Store the string in a variable
x<- "Hello World"

#use replicate() function to repet ceratin things multiple times
x2 <- paste(replicate(2, x), collapse = " ")

#concatinate multiple strings(use paste() function)
paste(x,x2)
x3 <- paste(x,"GAP",x2)

#extract string (Left, RIght, Mid (excel type functions))
#We use substr() funtion for that
#we will extract hello from the "Hello World" string
substr(x,1,5)
#indexing starts from 1, that is 1 is the first element

#we will create a vector of strings below

vs <- c("dog", "cat", "cow","cow","dog","cat", "dog","dog","cow")

#to see distribution of each element in the vector vs we will call
#table() function
table(vs)

#to see a plot of distribution of each element we will call plot() function
plot(table(vs))
barplot(table(vs))

#changing with place holder (i)
for(i in 1:4){
  print(paste(i,"rocking"))
}

#Reading webpage with rvest library
require(rvest)

url <- "https://en.wikipedia.org/wiki/Medium_(website)"
webpage <- read_html(url)
x <- html_text(webpage)
head(x)
```

Session 3

```
library(rvest)
#read html page
art <- html("http://news.bbc.co.uk/2/hi/health/2284783.stm")
#see text of html page
art_txt <- html_text(art)
#create token
library(tokenizers)
options(max.print = 20) #reduce the max print to 20
#charecter token
tokenize_characters(art_txt)
#word token
t_w <-  tokenize_words(art_txt)
#n gram tokenization, token of minimum 3 word and maximum 5 word
tokenize_ngrams(art_txt, n = 5, n_min = 3)

str <- "I Love NLP"
#lower string
tolower(str)
#upper string
toupper(str)

#we will use sringr package for following part
library(stringr)
#remove spl charecter, replace with space. we will use sringr package
spl <- "My love@you#rocking%disco.hey!rama"
str_replace_all(spl, "[[:punct:]]", " ")

#split
splt <- "baba@you@are@beautiful"
str_split(splt,"@")
#find specific string position
str_detect(splt,'@')
str_locate(splt,"@")
str_locate_all(splt,"@")

#call dictionary of words
words[1]

#return values ends with "ed"
for (i in 1:length(words)){
  ifelse(str_detect(words[i],'ed$')==TRUE,print(words[i]),"")
}
#alternate code
sapply(words, function(x){ifelse(str_detect(x,'ed$')==TRUE,print(x),"")})
```

```
#return values starts with "ab"
for (i in 1:length(words)){
  ifelse(str_detect(words[i],'^ab')==TRUE,print(words[i]),"")
}
#alternate code
sapply(words, function(x){ifelse(str_detect(x,'^ab')==TRUE,print(x),"")})

#return values  with 3dr word "c"
for (i in 1:length(words)){
  ifelse(str_detect(words[i],'^..c')==TRUE,print(words[i]),"")
}
#alternate code
sapply(words, function(x){ifelse(str_detect(x,'^..c')==TRUE,print(x),"")})

#return complex string
for (i in 1:length(words)){
  ifelse(str_detect(words[i],'^..c...t$')==TRUE,print(words[i]),"")
}
#alternate code
sapply(words, function(x){ifelse(str_detect(x,'^..c...t$')==TRUE,print(x),"")})
```