# ASSIGNMENT-1

Data Science with R

Question-1:

```
library(rvest)

library(dplyr)

 # URL of the webpage-

 url <- "https://www.moneyworks4me.com/best-index/nse-stocks/top-nifty50-companies-list/"
# Read the webpage

 webpage <- read_html(url)

# Extract data-

company_data <- webpage %>%

html_nodes("table") %>%

html_table(fill = TRUE)

 # Since the data is in the first table, we select it

 nifty50_data <- company_data[[1]]

# Display the data-

print(nifty50_data)
```

Question-3:

```
tennis <- function(p) {

 # Initialize the number of sets won by each player

 wins_A <- 0

 wins_B <- 0

 sets_played <- 0


 while (wins_A < 3 && wins_B < 3) {

  sets_played <- sets_played + 1


  # Simulate the outcome of one set

  if (runif(1) < p) {

   wins_A <- wins_A + 1
```

```
  } else {
    wins_B <- wins_B + 1
  }
 }

 return(sets_played)
}

# Example usage
set.seed(123)  # For reproducibility
p <- 0.6  # Probability of player A winning a set
number_of_sets <- tennis(p)
print(number_of_sets)
```

Question-4

```
MontyHall <- function() {
 # Randomly place the car behind one of the three doors
 doors <- c("goat", "goat", "car")
 doors <- sample(doors)

 # Contestant makes an initial choice
 initial_choice <- sample(1:3, 1)

 # Host opens one of the other doors with a goat
 available_doors <- setdiff(1:3, initial_choice)
 door_to_open <- available_doors[doors[available_doors] == "goat"]

 if(length(door_to_open) > 1) {
  door_to_open <- sample(door_to_open, 1)
 }
```

```
  # Contestant switches to the remaining door
  remaining_door <- setdiff(1:3, c(initial_choice, door_to_open))

  # Debugging: Print the choices and the doors
  cat("Initial choice:", initial_choice, "\n")
  cat("Door to open:", door_to_open, "\n")
  cat("Remaining door:", remaining_door, "\n")
  cat("Doors:", doors, "\n")

  # Check if the contestant wins by switching
  if (doors[remaining_door] == "car") {
    return(1)  # Contestant wins
  } else {
    return(0)  # Contestant loses
  }
}

# Example usage
set.seed(123)  # For reproducibility
result <- MontyHall()
print(result)

# Simulate the Monty Hall game 1000 times
simulate_games <- function(n) {
  results <- numeric(n)
  for (i in 1:n) {
    results[i] <- MontyHall()
  }
  return(results)
}
```

```
# Number of simulations
num_simulations <- 1000
results <- simulate_games(num_simulations)

# Estimate the probability of winning if the contestant switches
probability_of_winning <- mean(results)
print(probability_of_winning)
```

Question-5

```
library(rvest)
library(dplyr)
#Read the webpage
webpage <- read_html("https://editorial.rottentomatoes.com/guide/best-netflix-movies-to-watch-right-now/")

# Extract movie rankings, names, Tomato % scores, and years
rankings <- webpage %>%
  html_nodes(".countdown-index") %>%
  html_text(trim = TRUE) %>%
  as.numeric()

names <- webpage %>%
  html_nodes(".article_movie_title a") %>%
  html_text(trim = TRUE)

scores <- webpage %>%
  html_nodes(".tMeterScore") %>%
  html_text(trim = TRUE) %>%
  gsub("%", "", .) %>%
  as.numeric()
```

```
years <- webpage %>%
  html_nodes(".start-year") %>%
  html_text(trim = TRUE) %>%
  gsub("\\(", "", .) %>%
  gsub("\\)", "", .) %>%
  as.numeric()

# Combine the extracted information into a data frame
movies_data <- data.frame(
  Ranking = rankings,
  Name = names,
  Tomato_Score = scores,
  Year = years,
  stringsAsFactors = FALSE
)

# Display the data frame
print(movies_data)
```