EXPT NO:10	DATA IMPORTING AND EXPORTING
DATE:2.04.2025	

AIM:

To implement R programming on data importing and exporting (csv, excel, json)

CSV:

Create a CSV file containing fitness tracking data for a group of users with the attributes (User_ID,Name, Age, Gender, Daily_Steps, Calories_Burned, Workout_Type, Workout_Duration (mins),Heart_Rate)

CODE:

```
data <-data.frame(
 user_id = c("1","2","3","4"),
 Name= c("Harini","Bhaanu","aquina","jigisha"),
 Age=c("19","20","18","21"),
 Gender=c("Female","Female","Female","Female"),
 Daily_step=c("7000","9000","8000","10000"),
 calories_burned=c("5000","7000","6000","8000"),
 Workout_type=c("Cardio", "Strength", "Cardio", "Yoga"),
 workout_Duration =c("18","60","55","45"),
 Heart_rate = c("180","150","190","172"))
write.csv(data,"fitness_data.csv",row.names=FALSE)
data <- read.csv("fitness_data.csv")
```

1. Calculate the average number of steps taken per day.

```
avg <- mean(data$Daily_step)
avg
```

2. Identify the most common workout type among users.

```
common_workout <- names(sort(table(data$Workout_type),decreasing=TRUE)[1])
common_workout
```

3. Find the user with the highest calories burned in a single workout session.

```
max_cal <- data[which.max(data$calories_burned),c("Name","calories_burned")]
print("Maximum Calories Burned:")
print(max_cal)
```

4. Determine the percentage of users who exercise for more than 45 minutes per day.

```
per <- sum(data$workout_Duration>45)/nrow(data)*100
per
```

5. Analyze the average heart rate recorded during different workout types.

```
data$Heart_rate <- as.numeric(data$Heart_rate)
avg_heart_rate <- data %>%
  group_by(Workout_type) %>%
  summarise(avg_hr = mean(Heart_rate))
print(avg_heart_rate)
```

OUTPUT:

```
> avg <- mean(data$Daily_step)
> avg
[1] 8500
> common_workout <- names(sort(table(data$workout_type),decreasing=TRUE)[1])
> common_workout
[1] "Cardio"
> max_cal <- data[which.max(data$calories_burned),c("Name","calories_burned")]
> print("Maximum Calories Burned:")
[1] "Maximum Calories Burned:"
> print(max_cal)
     Name calories_burned
4 jigisha            8000
> per <- sum(data$workout_Duration>45)/nrow(data)*100
> per
[1] 50
```

```
> data$Heart_rate <- as.numeric(data$Heart_rate)
> avg_heart_rate <- data %>%
+   group_by(workout_type) %>%
+   summarise(avg_hr = mean(Heart_rate))
>
> print(avg_heart_rate)
# A tibble: 3 × 2
  workout_type avg_hr
  <chr>          <dbl>
1 Cardio           185
2 Strength         150
3 Yoga             172
```

EXCEL:

Create an Excel file containing real-time traffic data from a smart city surveillance system with the attributes (Camera_ID, Location, Vehicle_Count, Average_Speed (km/h), Peak_Hour,Accident_Reported (Yes/No), Weather_Condition)

```
traffic_data <- data.frame(
  Camera_ID = c("C101", "C102", "C103", "C104", "C105"),
  Location = c("Main Street", "Central Ave", "Airport Road", "Tech Park", "City Mall"),
  Vehicle_Count = c(120, 300, 150, 400, 275),
  Average_Speed = c(45.2, 52.5, 39.8, 60.0, 50.3),
  Peak_Hour = c("08:00-09:00", "17:00-18:00", "07:30-08:30", "18:00-19:00", "16:30-
17:30"),
  Accident_Reported = c("No", "Yes", "No", "No", "Yes"),
  Weather_Condition = c("Clear", "Rainy", "Foggy", "Clear", "Cloudy")
)
write_xlsx(traffic_data, "smart_city_traffic_data.xlsx")
traf_data <- read_excel("smart_city_traffic_data.xlsx")
```

1. Identify the location with the highest traffic volume.

```
max_veh <- traf_data[which.max(traf_data$Vehicle_Count),c("Location","Vehicle_Count")]
max_veh
```

2. Determine the percentage of accidents reported during peak hours.

```
percent <- sum(traf_data$Accident_Reported == "Yes")/nrow(traf_data)*100
percent
```

3. Find the average speed of vehicles in different weather conditions.

```
avg_spd <- traf_data %>% group_by(Weather_Condition) %>%
summarise(avg_speed=mean(Average_Speed))
avg_spd
```

4. Analyze the most common peak hour for traffic congestion.

```
common_peak <- names(sort(table(traf_data$Peak_Hour),decreasing=TRUE)[1])
common_peak
```

5. Identify locations where accidents frequently occur.

```
accident_locations <- traf_data[traf_data$Accident_Reported == "Yes", ]
accident_count <- table(accident_locations$Location)
most_frequent_accident_location <- names(accident_count[accident_count ==
max(accident_count)])
print(most_frequent_accident_location)
```

OUTPUT:

```
> max_veh
# A tibble: 1 × 2
  Location  Vehicle_Count
  <chr>             <dbl>
1 Tech Park           400
> percent <- sum(traf_data$Accident_Reported == "Yes")/nrow(traf_data)*100
> percent
[1] 40
```

```
> avg_spd <- traf_data %>% group_by(Weather_Condition) %>% summarise(avg_speed
=mean(Average_Speed))
> avg_spd
# A tibble: 4 × 2
  Weather_Condition avg_speed
  <chr>                 <dbl>
1 Clear                  52.6
2 Cloudy                 50.3
3 Foggy                  39.8
4 Rainy                  52.5
> common_peaK <- names(sort(table(traf_data$Peak_Hour),decreasing=TRUE)[1])
> common_peaK
[1] "07:30-08:30"
```

```
> accident_locations
# A tibble: 2 × 7
  Camera_ID Location     Vehicle_Count Average_Speed Peak_Hour
  <chr>     <chr>                <dbl>         <dbl> <chr>
1 C102      Central Ave            300          52.5 17:00-18:00
2 C105      City Mall              275          50.3 16:30-17:30
# ℹ 2 more variables: Accident_Reported <chr>, Weather_Condition <chr>
> accident_count

Central Ave   City Mall
          1           1
```

JSON:

Create a JSON file containing daily weather data for a city with the attributes (Date, City, Temperature (°C), Humidity (%), Wind Speed (km/h), Precipitation (mm), Weather_Condition)

```
install.packages("jsonlite")
library(jsonlite)
weather_data <- data.frame(
  Date = as.Date(c("2025-04-01", "2025-04-02", "2025-04-03", "2025-04-04", "2025-04-
05")),
  City = rep("Coimbatore", 5),
  Temperature_C = c(34, 33, 32, 30, 31),
  Humidity_percent = c(48, 50, 55, 60, 58),
  Wind_Speed_kmph = c(12, 15, 10, 8, 11),
  Precipitation_mm = c(0, 0, 2, 5, 0),
  Weather_Condition = c("Sunny", "Partly Cloudy", "Light Rain", "Rainy", "Cloudy"))
weather_json <- toJSON(weather_data, pretty = TRUE)
write(weather_json, file = "weather_data.json")
```

1. Find the average temperature for the past 7 days.

```
average_temp <-mean(weather_data$Temperature_C)
average_temp
```

2. Identify the day with the highest humidity.

```
max_humidity_day <- weather_data[which.max(weather_data$Humidity_percent), ]
cat("Day with highest humidity:\n")
print(max_humidity_day)
```

3. Determine the most common weather condition (e.g., Sunny, Rainy, Cloudy).

```
most_common_condition <- names(sort(table(weather_data$Weather_Condition), decreasing
= TRUE))[1]
cat("Most common weather condition:", most_common_condition, "\n")
```

4. Calculate the total precipitation for the past 30 days.

```
total_precipitation <- sum(weather_data$Precipitation_mm)
cat("Total precipitation (30 days):", total_precipitation, "mm\n")
```

5. Identify the windiest day based on wind speed.

```
windiest_day <- weather_data[which.max(weather_data$Wind_Speed_kmph), ]
cat("Windiest day:\n")
print(windiest_day)
```

OUTPUT:

```
> weather_json <- toJSON(weather_data, pretty = TRUE)
> write(weather_json, file = "weather_data.json")
> average_temp <-mean(weather_data$Temperature_C)
> average_temp
[1] 32
> max_humidity_day <- weather_data[which.max(weather_data$Humidity_percent), ]
> cat("Day with highest humidity:\n")
Day with highest humidity:
> print(max_humidity_day)
        Date       City Temperature_C Humidity_percent Wind_Speed_kmph
4 2025-04-04 Coimbatore            30               60               8
  Precipitation_mm Weather_Condition
4                5             Rainy
> most_common_condition <- names(sort(table(weather_data$Weather_Condition), d
ecreasing = TRUE))[1]
> cat("Most common weather condition:", most_common_condition, "\n")
Most common weather condition: Cloudy
> total_precipitation <- sum(weather_data$Precipitation_mm)
> cat("Total precipitation (30 days):", total_precipitation, "mm\n")
Total precipitation (30 days): 7 mm
> windiest_day <- weather_data[which.max(weather_data$Wind_Speed_kmph), ]
> cat("Windiest day:\n")
Windiest day:
> print(windiest_day)
        Date       City Temperature_C Humidity_percent Wind_Speed_kmph
2 2025-04-02 Coimbatore            33               50              15
  Precipitation_mm Weather_Condition
2                0     Partly Cloudy
```

RESULT:

Thus the implementation on R programming on data importing and exporting (csv, excel, json)has been executed successfully.