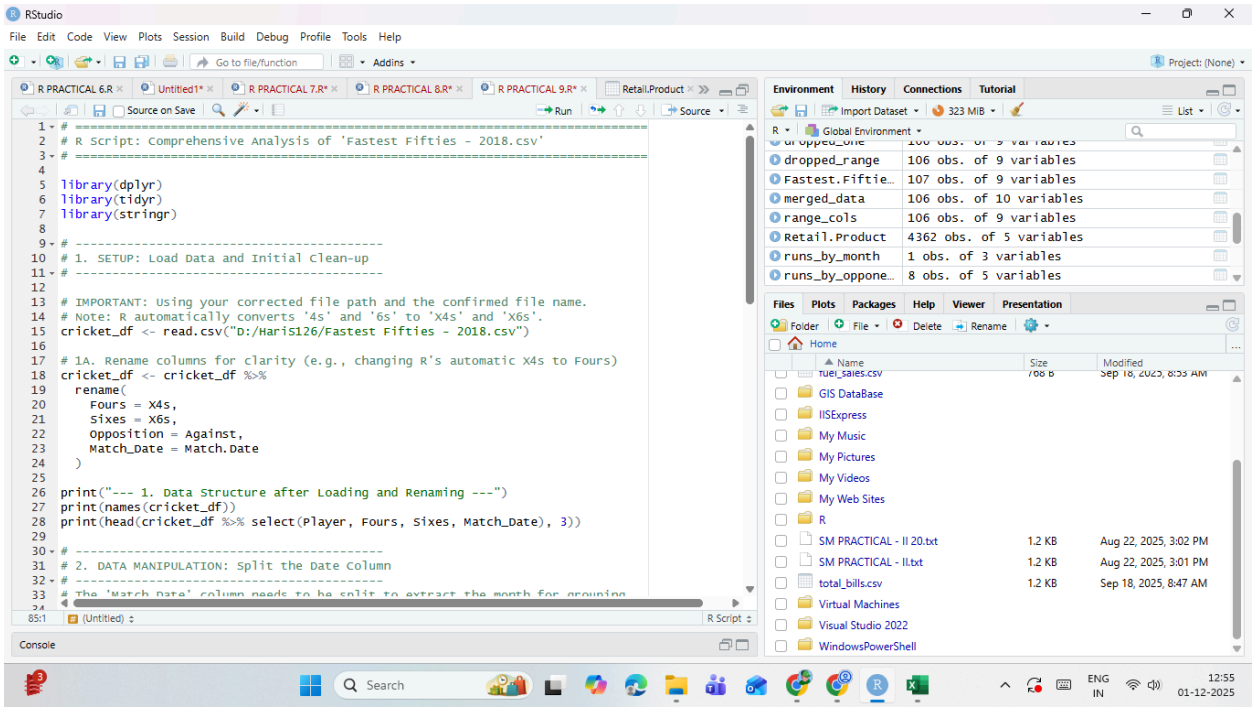
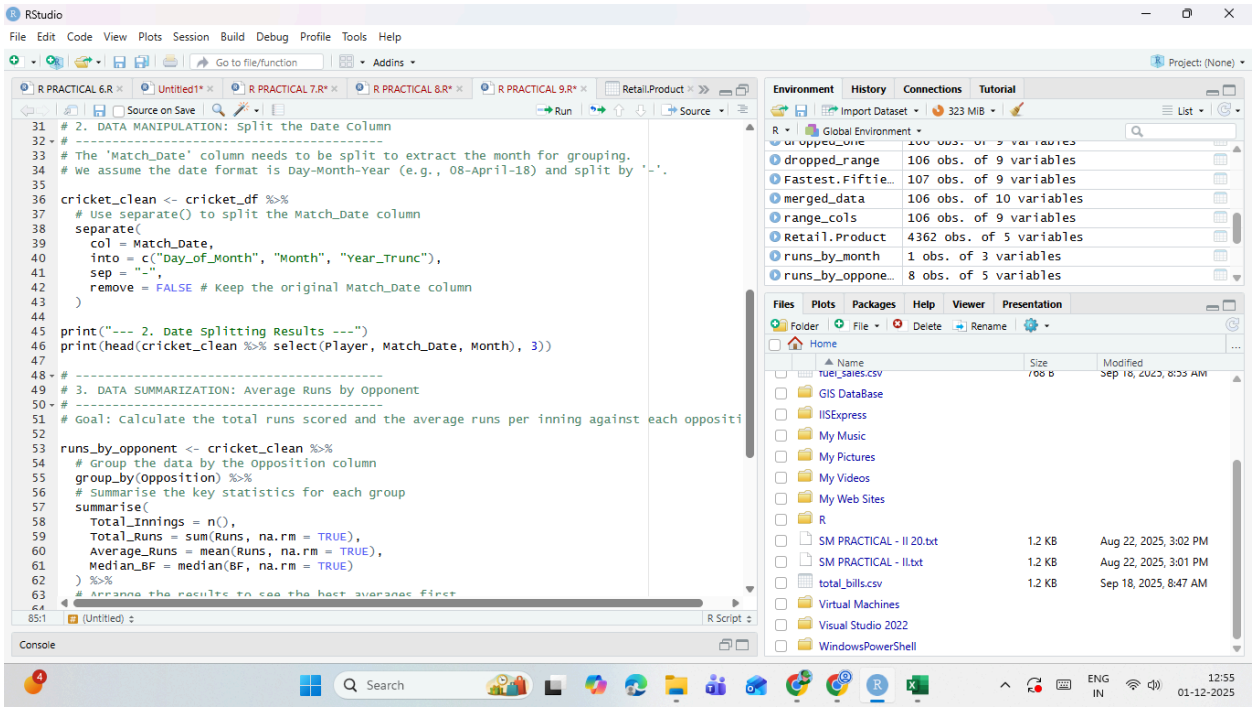
MVLU COLLEGE
PRACTICAL NO.9

Aim: Performing text manipulation using str_sub(), str_split() (R). import dataset

Code:



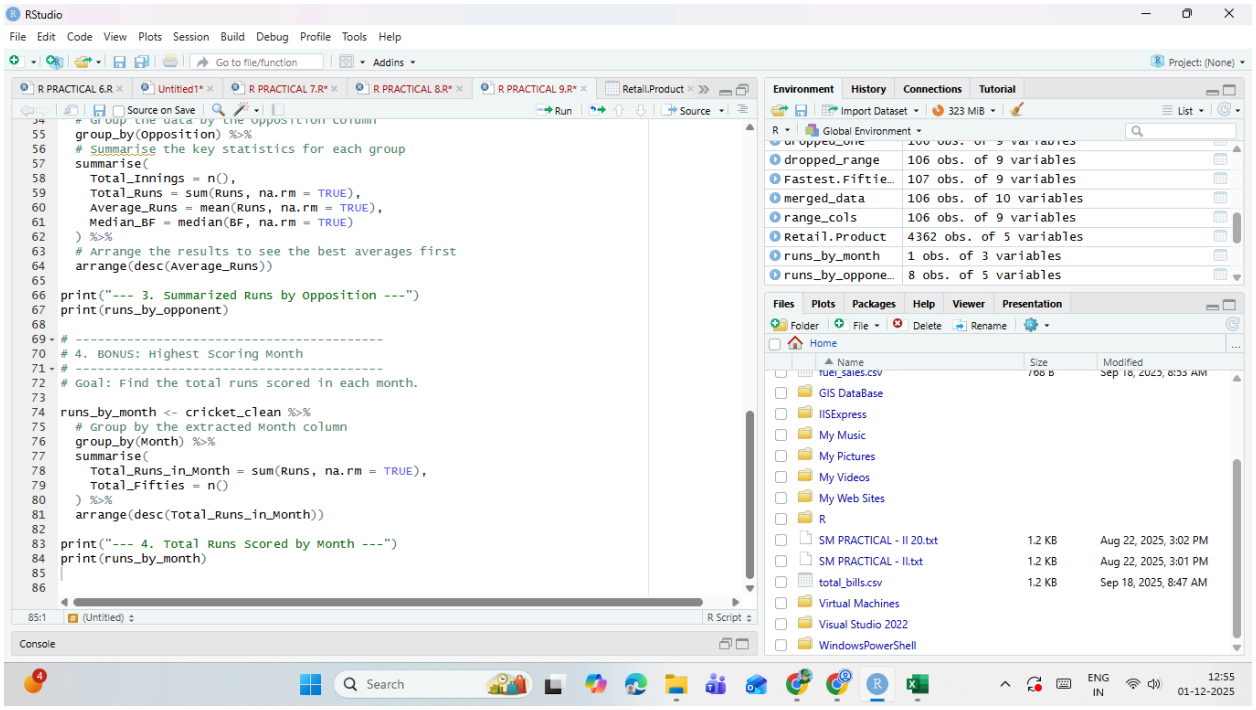
```
# ================================================================================
# R Script: Comprehensive Analysis of 'Fastest Fifties - 2018.csv'
# ================================================================================

library(dplyr)
library(tidyr)
library(stringr)

# ------------------------------------------
# 1. SETUP: Load Data and Initial Clean-up
# ------------------------------------------

# IMPORTANT: Using your corrected file path and the confirmed file name.
# Note: R automatically converts '4s' and '6s' to 'X4s' and 'X6s'.
cricket_df <- read.csv("D:/HariS126/Fastest Fifties - 2018.csv")

# 1A. Rename columns for clarity (e.g., changing R's automatic X4s to Fours)
cricket_df <- cricket_df %>%
  rename(
    Fours = X4s,
    Sixes = X6s,
    Opposition = Against,
    Match_Date = Match.Date
  )

print("--- 1. Data Structure after Loading and Renaming ---")
print(names(cricket_df))
print(head(cricket_df %>% select(Player, Fours, Sixes, Match_Date), 3))

# ------------------------------------------
# 2. DATA MANIPULATION: Split the Date Column
# ------------------------------------------
# The 'Match_Date' column needs to be split to extract the month for grouping.
# We assume the date format is Day-Month-Year (e.g., 08-April-18) and split by '-'.

cricket_clean <- cricket_df %>%
  # Use separate() to split the Match_Date column
  separate(
    col = Match_Date,
    into = c("Day_of_Month", "Month", "Year_Trunc"),
    sep = "-",
    remove = FALSE # Keep the original Match_Date column
  )

print("--- 2. Date Splitting Results ---")
print(head(cricket_clean %>% select(Player, Match_Date, Month), 3))

# ------------------------------------------
# 3. DATA SUMMARIZATION: Average Runs by Opponent
# ------------------------------------------
# Goal: Calculate the total runs scored and the average runs per inning against each oppositi

runs_by_opponent <- cricket_clean %>%
  # Group the data by the Opposition column
  group_by(Opposition) %>%
  # Summarise the key statistics for each group
  summarise(
    Total_Innings = n(),
    Total_Runs = sum(Runs, na.rm = TRUE),
    Average_Runs = mean(Runs, na.rm = TRUE),
    Median_BF = median(BF, na.rm = TRUE)
  ) %>%
  # Arrange the results to see the best averages first
```



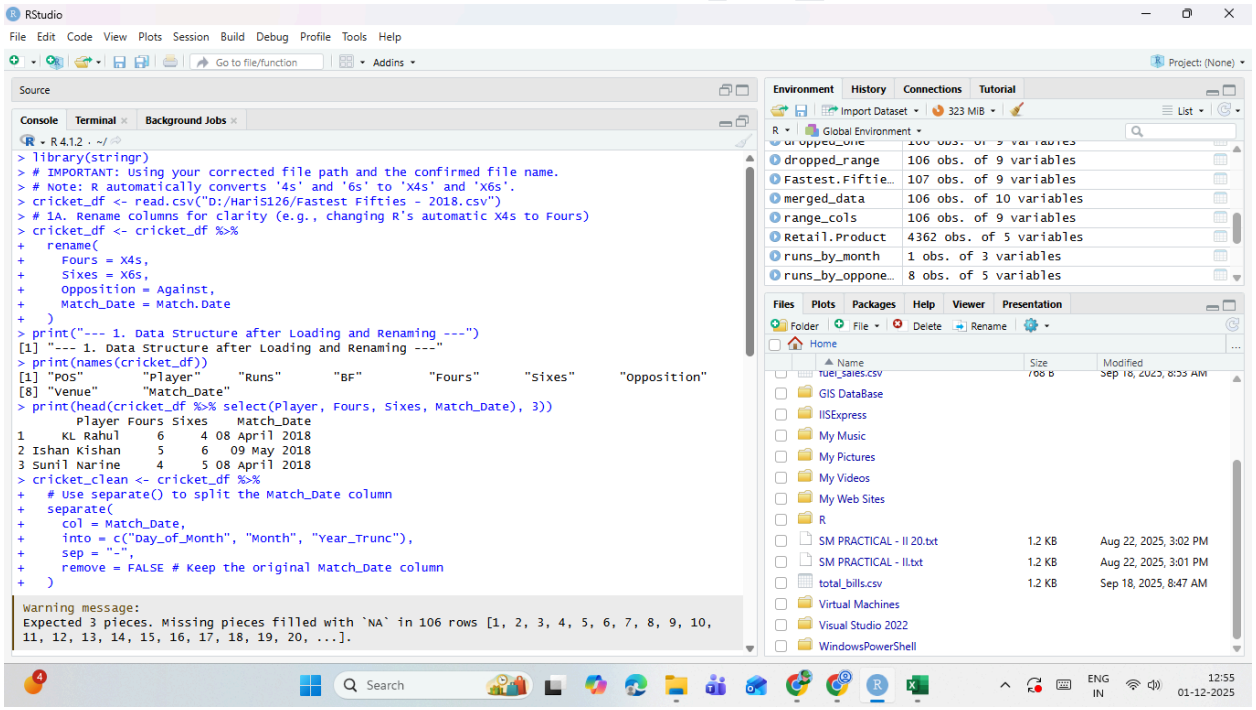

```
# ================================================================================
# R Script: Comprehensive Analysis of 'Fastest Fifties - 2018.csv'
# ================================================================================

library(dplyr)
library(tidyr)
library(stringr)

# ------------------------------------------
# 1. SETUP: Load Data and Initial Clean-up
# ------------------------------------------

# IMPORTANT: Using your corrected file path and the confirmed file name.
# Note: R automatically converts '4s' and '6s' to 'X4s' and 'X6s'.
cricket_df <- read.csv("D:/HariS126/Fastest Fifties - 2018.csv")

# 1A. Rename columns for clarity (e.g., changing R's automatic X4s to Fours)
cricket_df <- cricket_df %>%
  rename(
    Fours = X4s,
    Sixes = X6s,
    Opposition = Against,
    Match_Date = Match.Date
  )

print("--- 1. Data Structure after Loading and Renaming ---")
print(names(cricket_df))
print(head(cricket_df %>% select(Player, Fours, Sixes, Match_Date), 3))

# ------------------------------------------
# 2. DATA MANIPULATION: Split the Date Column
# ------------------------------------------
# The 'Match_Date' column needs to be split to extract the month for grouping.
# We assume the date format is Day-Month-Year (e.g., 08-April-18) and split by '-'.

cricket_clean <- cricket_df %>%
  # Use separate() to split the Match_Date column
  separate(
    col = Match_Date,
    into = c("Day_of_Month", "Month", "Year_Trunc"),
    sep = "-",
    remove = FALSE # Keep the original Match_Date column
  )

print("--- 2. Date Splitting Results ---")
print(head(cricket_clean %>% select(Player, Match_Date, Month), 3))

# ------------------------------------------
# 3. DATA SUMMARIZATION: Average Runs by Opponent
# ------------------------------------------
# Goal: Calculate the total runs scored and the average runs per inning against each oppositi

runs_by_opponent <- cricket_clean %>%
  # Group the data by the Opposition column
  group_by(Opposition) %>%
  # Summarise the key statistics for each group
  summarise(
    Total_Innings = n(),
    Total_Runs = sum(Runs, na.rm = TRUE),
    Average_Runs = mean(Runs, na.rm = TRUE),
    Median_BF = median(BF, na.rm = TRUE)
  ) %>%
  # Arrange the results to see the best averages first
```


Hariprasad Vishwakarma
S126
SYCS

MVLU COLLEGE
PRACTICAL NO.9

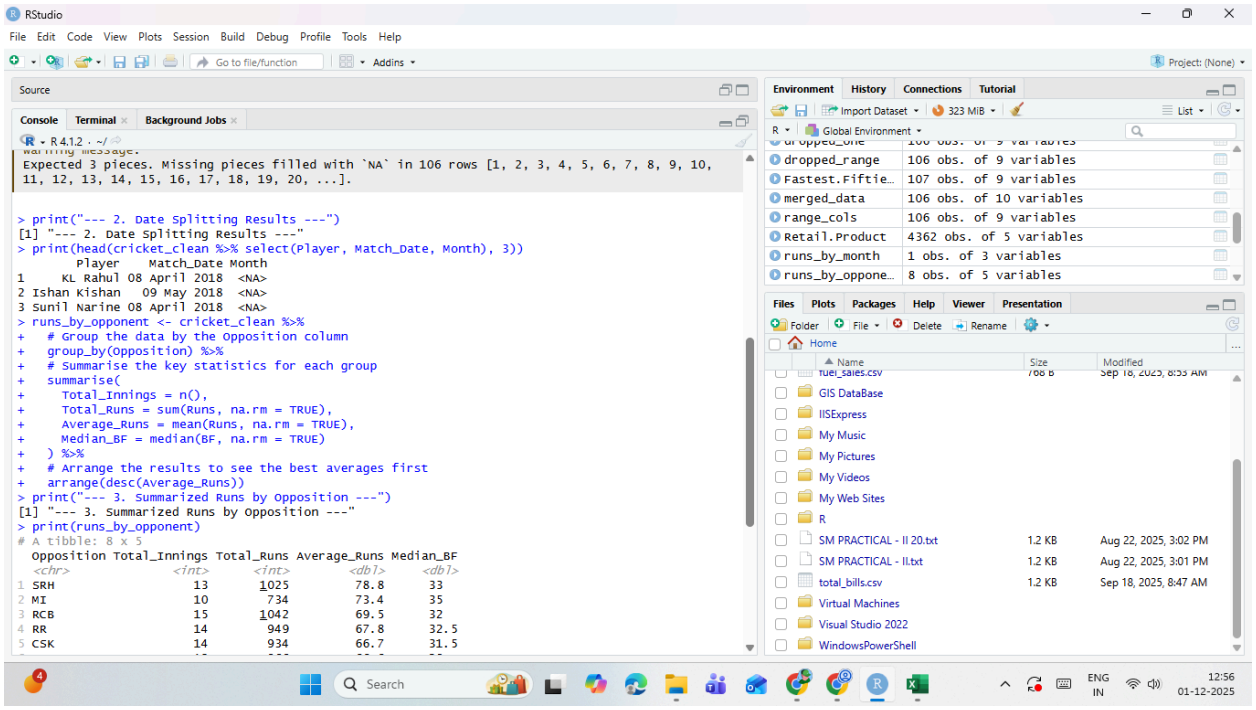


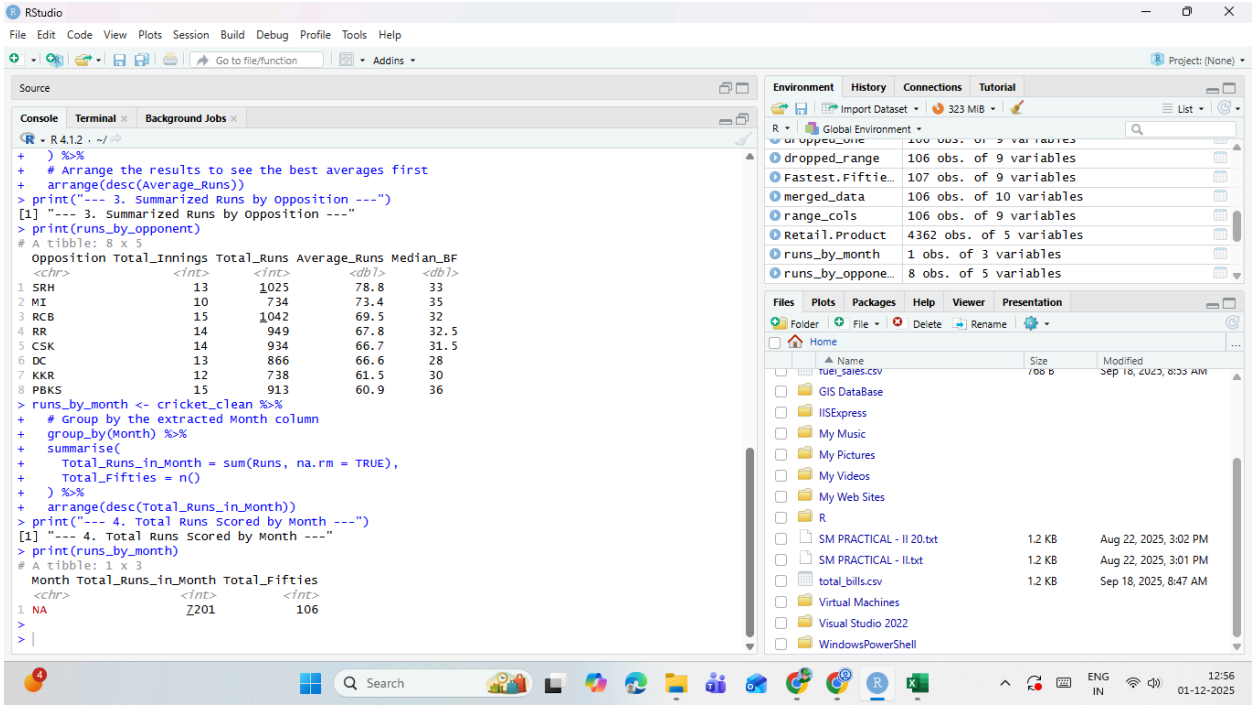
Output:



Hariprasad Vishwakarma
S126
SYCS

MVLU COLLEGE
PRACTICAL NO.9





.

Hariprasad Vishwakarma
S126
SYCS