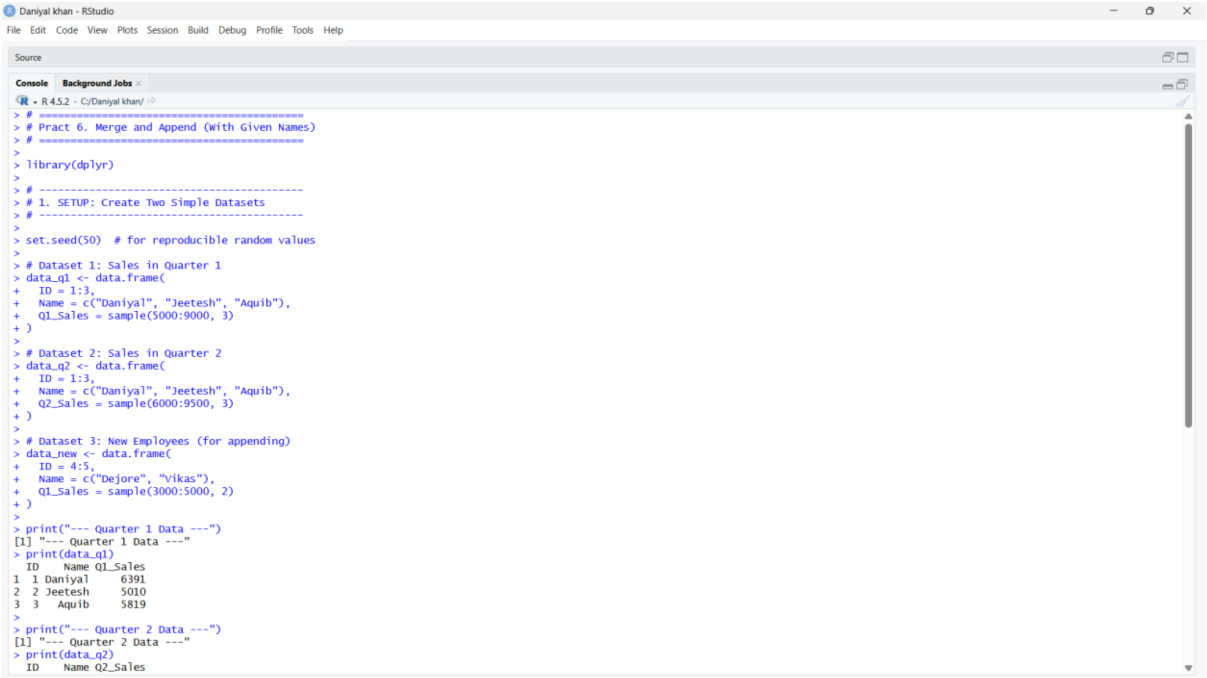
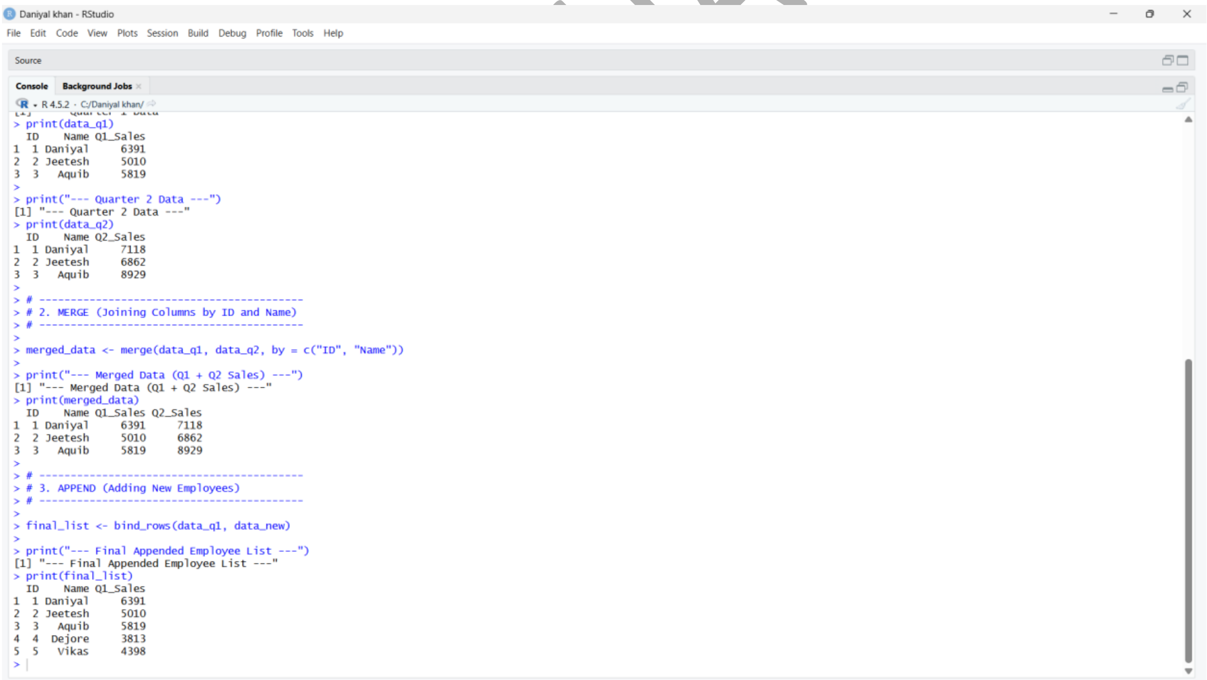
SHETH L.U.J COLLEGE

R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10

6:


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> # ==========================================
> # Pract 6. Merge and Append (With Given Names)
> # ==========================================
>
> library(dplyr)
>
> # ------------------------------------------
> # 1. SETUP: Create Two Simple Datasets
> # ------------------------------------------
>
> set.seed(50)  # for reproducible random values
>
> # Dataset 1: Sales in Quarter 1
> data_q1 <- data.frame(
+   ID = 1:3,
+   Name = c("Daniyal", "Jeetesh", "Aquib"),
+   Q1_Sales = sample(5000:9000, 3)
+ )
>
> # Dataset 2: Sales in Quarter 2
> data_q2 <- data.frame(
+   ID = 1:3,
+   Name = c("Daniyal", "Jeetesh", "Aquib"),
+   Q2_Sales = sample(6000:9500, 3)
+ )
>
> # Dataset 3: New Employees (for appending)
> data_new <- data.frame(
+   ID = 4:5,
+   Name = c("Dejore", "Vikas"),
+   Q1_Sales = sample(3000:5000, 2)
+ )
>
> print("--- Quarter 1 Data ---")
[1] "--- Quarter 1 Data ---"
> print(data_q1)
  ID    Name Q1_Sales
1  1 Daniyal     6391
2  2 Jeetesh     5010
3  3   Aquib     5819
>
> print("--- Quarter 2 Data ---")
[1] "--- Quarter 2 Data ---"
> print(data_q2)
  ID    Name Q2_Sales
```

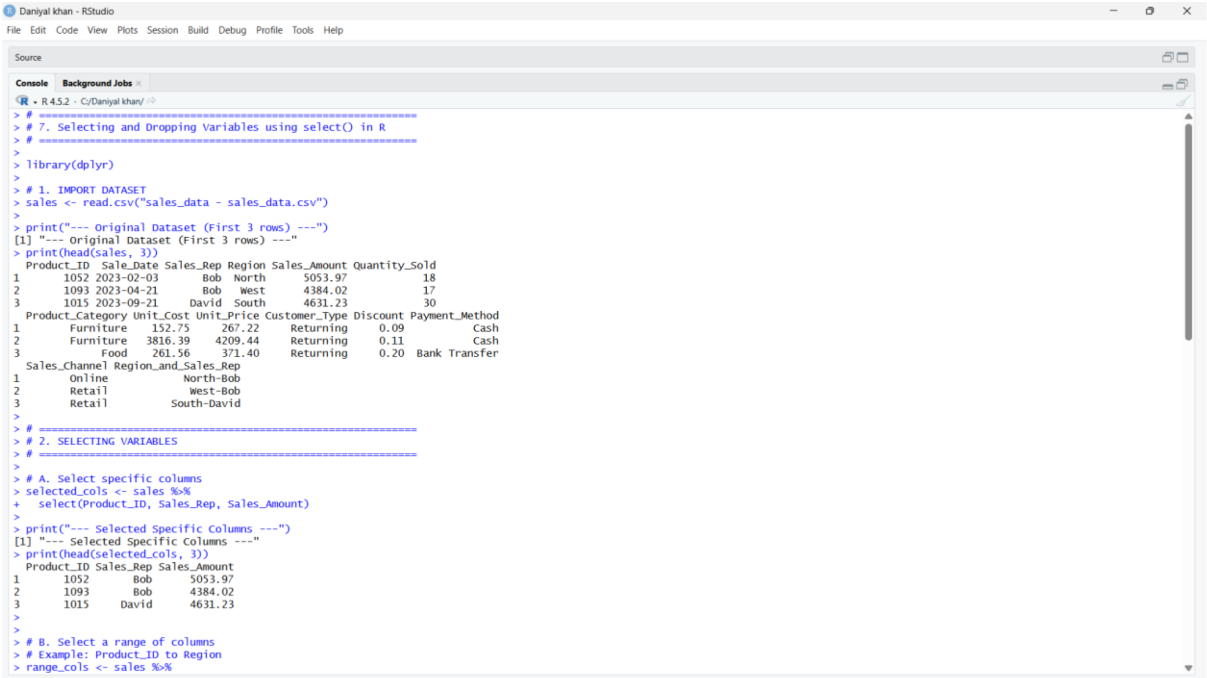

Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> print(data_q1)
  ID    Name Q1_Sales
1  1 Daniyal     6391
2  2 Jeetesh     5010
3  3   Aquib     5819
>
> print("--- Quarter 2 Data ---")
[1] "--- Quarter 2 Data ---"
> print(data_q2)
  ID    Name Q2_Sales
1  1 Daniyal     7118
2  2 Jeetesh     6862
3  3   Aquib     8929
>
> # ------------------------------------------
> # 2. MERGE (Joining Columns by ID and Name)
> # ------------------------------------------
>
> merged_data <- merge(data_q1, data_q2, by = c("ID", "Name"))
>
> print("--- Merged Data (Q1 + Q2 Sales) ---")
[1] "--- Merged Data (Q1 + Q2 Sales) ---"
> print(merged_data)
  ID    Name Q1_Sales Q2_Sales
1  1 Daniyal     6391     7118
2  2 Jeetesh     5010     6862
3  3   Aquib     5819     8929
>
> # ------------------------------------------
> # 3. APPEND (Adding New Employees)
> # ------------------------------------------
>
> final_list <- bind_rows(data_q1, data_new)
>
> print("--- Final Appended Employee List ---")
[1] "--- Final Appended Employee List ---"
> print(final_list)
  ID    Name Q1_Sales
1  1 Daniyal     6391
2  2 Jeetesh     5010
3  3   Aquib     5819
4  4  Dejore     3813
5  5   Vikas     4398
>
```

SHETH L.U.J COLLEGE

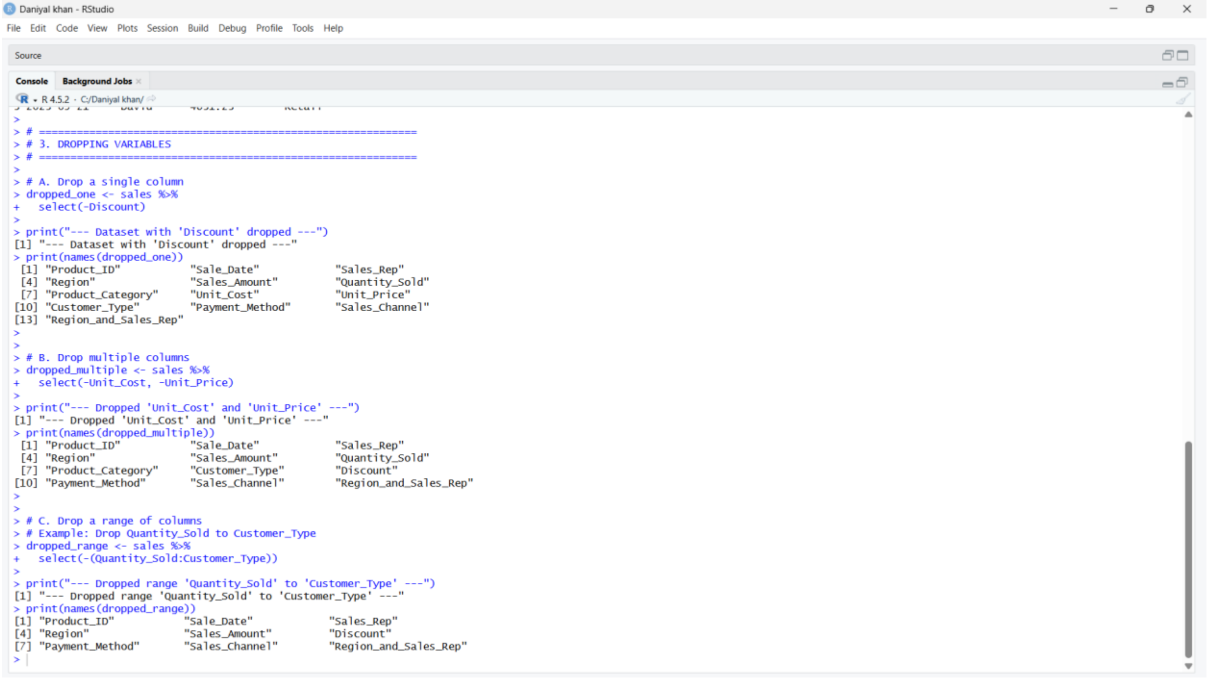
R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10

7:



Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

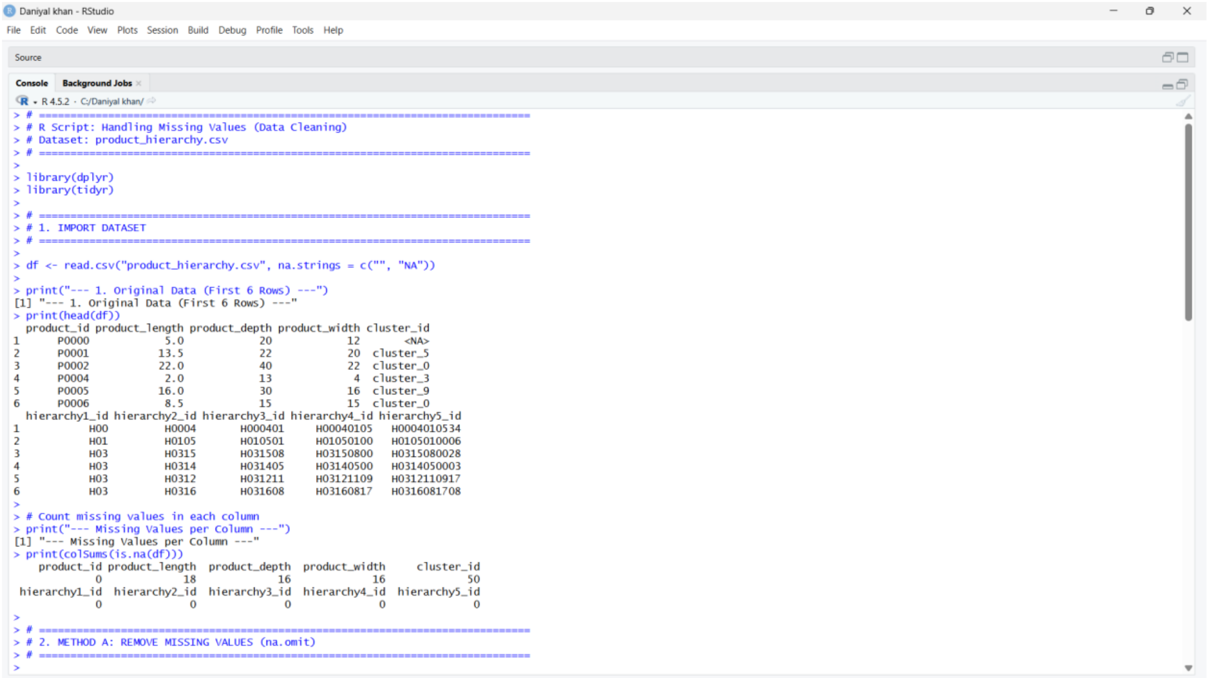
```
> # ============================================================
> # 7. Selecting and Dropping Variables using select() in R
> # ============================================================
>
> library(dplyr)
>
> # 1. IMPORT DATASET
> sales <- read.csv("sales_data - sales_data.csv")
>
> print("--- Original Dataset (First 3 rows) ---")
[1] "--- Original Dataset (First 3 rows) ---"
> print(head(sales, 3))
  Product_ID  Sale_Date Sales_Rep Region Sales_Amount Quantity_Sold
1       1052 2023-02-03       Bob  North      5053.97            18
2       1093 2023-04-21       Bob   West      4384.02            17
3       1015 2023-09-21     David  South      4631.23            30
  Product_Category Unit_Cost Unit_Price Customer_Type Discount Payment_Method
1        Furniture    152.75     267.22     Returning     0.09           Cash
2        Furniture   3816.39    4209.44     Returning     0.11           Cash
3             Food    261.56     371.40     Returning     0.20  Bank Transfer
  Sales_Channel Region_and_Sales_Rep
1        Online            North-Bob
2        Retail             West-Bob
3        Retail          South-David
>
> # ============================================================
> # 2. SELECTING VARIABLES
> # ============================================================
>
> # A. Select specific columns
> selected_cols <- sales %>%
+   select(Product_ID, Sales_Rep, Sales_Amount)
>
> print("--- Selected Specific Columns ---")
[1] "--- Selected Specific Columns ---"
> print(head(selected_cols, 3))
  Product_ID Sales_Rep Sales_Amount
1       1052       Bob      5053.97
2       1093       Bob      4384.02
3       1015     David      4631.23
>
>
> # B. Select a range of columns
> # Example: Product_ID to Region
> range_cols <- sales %>%
```



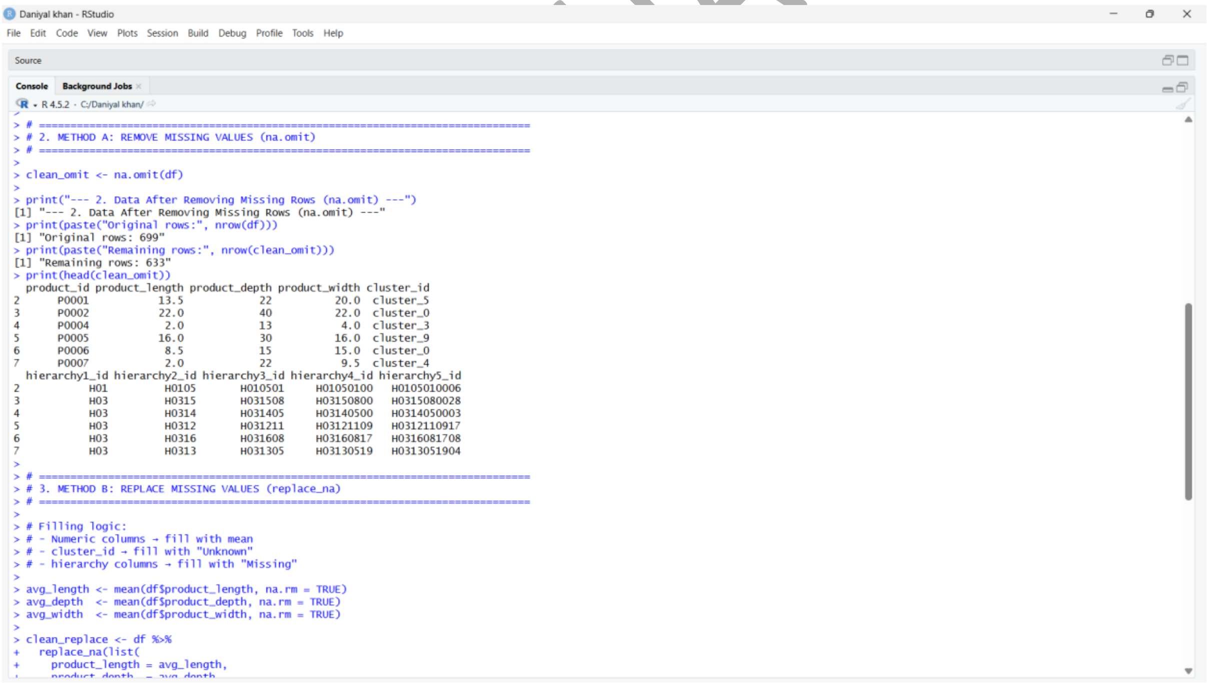
Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
>
> # ============================================================
> # 3. DROPPING VARIABLES
> # ============================================================
>
> # A. Drop a single column
> dropped_one <- sales %>%
+   select(-Discount)
>
> print("--- Dataset with 'Discount' dropped ---")
[1] "--- Dataset with 'Discount' dropped ---"
> print(names(dropped_one))
 [1] "Product_ID"           "Sale_Date"            "Sales_Rep"
 [4] "Region"               "Sales_Amount"         "Quantity_Sold"
 [7] "Product_Category"     "Unit_Cost"            "Unit_Price"
[10] "Customer_Type"        "Payment_Method"       "Sales_Channel"
[13] "Region_and_Sales_Rep"
>
>
> # B. Drop multiple columns
> dropped_multiple <- sales %>%
+   select(-Unit_Cost, -Unit_Price)
>
> print("--- Dropped 'Unit_Cost' and 'Unit_Price' ---")
[1] "--- Dropped 'Unit_Cost' and 'Unit_Price' ---"
> print(names(dropped_multiple))
 [1] "Product_ID"           "Sale_Date"            "Sales_Rep"
 [4] "Region"               "Sales_Amount"         "Quantity_Sold"
 [7] "Product_Category"     "Customer_Type"        "Discount"
[10] "Payment_Method"       "Sales_Channel"        "Region_and_Sales_Rep"
>
>
> # C. Drop a range of columns
> # Example: Drop Quantity_Sold to Customer_Type
> dropped_range <- sales %>%
+   select(-(Quantity_Sold:Customer_Type))
>
> print("--- Dropped range 'Quantity_Sold' to 'Customer_Type' ---")
[1] "--- Dropped range 'Quantity_Sold' to 'Customer_Type' ---"
> print(names(dropped_range))
[1] "Product_ID"           "Sale_Date"            "Sales_Rep"
[4] "Region"               "Sales_Amount"         "Discount"
[7] "Payment_Method"       "Sales_Channel"        "Region_and_Sales_Rep"
>
```

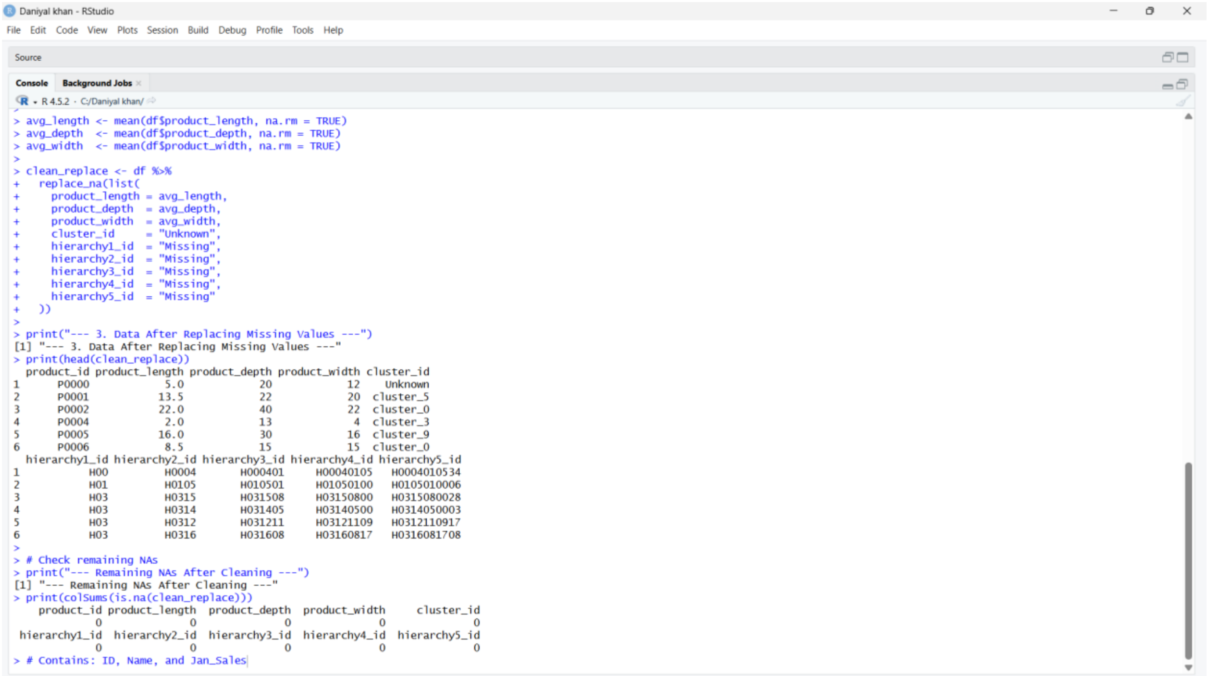
8:


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> # ==============================================================================
> # R Script: Handling Missing Values (Data Cleaning)
> # Dataset: product_hierarchy.csv
> # ==============================================================================
>
> library(dplyr)
> library(tidyr)
>
> # ==============================================================================
> # 1. IMPORT DATASET
> # ==============================================================================
>
> df <- read.csv("product_hierarchy.csv", na.strings = c("", "NA"))
>
> print("--- 1. Original Data (First 6 Rows) ---")
[1] "--- 1. Original Data (First 6 Rows) ---"
> print(head(df))
  product_id product_length product_depth product_width cluster_id
1      P0000            5.0            20            12       <NA>
2      P0001           13.5            22            20  cluster_5
3      P0002           22.0            40            22  cluster_0
4      P0004            2.0            13             4  cluster_3
5      P0005           16.0            30            16  cluster_9
6      P0006            8.5            15            15  cluster_0
  hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id hierarchy5_id
1           H00         H0004       H000401     H00040105   H0004010534
2           H01         H0105       H010501     H01050100   H0105010006
3           H03         H0315       H031508     H03150800   H0315080028
4           H03         H0314       H031405     H03140500   H0314050003
5           H03         H0312       H031211     H03121109   H0312110917
6           H03         H0316       H031608     H03160817   H0316081708
>
> # Count missing values in each column
> print("--- Missing Values per Column ---")
[1] "--- Missing Values per Column ---"
> print(colSums(is.na(df)))
    product_id product_length  product_depth  product_width     cluster_id
             0             18             16             16             50
 hierarchy1_id  hierarchy2_id  hierarchy3_id  hierarchy4_id  hierarchy5_id
             0              0              0              0              0
>
> # ==============================================================================
> # 2. METHOD A: REMOVE MISSING VALUES (na.omit)
> # ==============================================================================
>
```


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> # ==============================================================================
> # 2. METHOD A: REMOVE MISSING VALUES (na.omit)
> # ==============================================================================
>
> clean_omit <- na.omit(df)
>
> print("--- 2. Data After Removing Missing Rows (na.omit) ---")
[1] "--- 2. Data After Removing Missing Rows (na.omit) ---"
> print(paste("Original rows:", nrow(df)))
[1] "Original rows: 699"
> print(paste("Remaining rows:", nrow(clean_omit)))
[1] "Remaining rows: 633"
> print(head(clean_omit))
  product_id product_length product_depth product_width cluster_id
2      P0001           13.5            22          20.0  cluster_5
3      P0002           22.0            40          22.0  cluster_0
4      P0004            2.0            13           4.0  cluster_3
5      P0005           16.0            30          16.0  cluster_9
6      P0006            8.5            15          15.0  cluster_0
7      P0007            2.0            22           9.5  cluster_4
  hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id hierarchy5_id
2           H01         H0105       H010501     H01050100   H0105010006
3           H03         H0315       H031508     H03150800   H0315080028
4           H03         H0314       H031405     H03140500   H0314050003
5           H03         H0312       H031211     H03121109   H0312110917
6           H03         H0316       H031608     H03160817   H0316081708
7           H03         H0313       H031305     H03130519   H0313051904
>
> # ==============================================================================
> # 3. METHOD B: REPLACE MISSING VALUES (replace_na)
> # ==============================================================================
>
> # Filling logic:
> # - Numeric columns → fill with mean
> # - cluster_id → fill with "Unknown"
> # - hierarchy columns → fill with "Missing"
>
> avg_length <- mean(df$product_length, na.rm = TRUE)
> avg_depth  <- mean(df$product_depth, na.rm = TRUE)
> avg_width  <- mean(df$product_width, na.rm = TRUE)
>
> clean_replace <- df %>%
+   replace_na(list(
+     product_length = avg_length,
+     product_depth  = avg_depth,
```

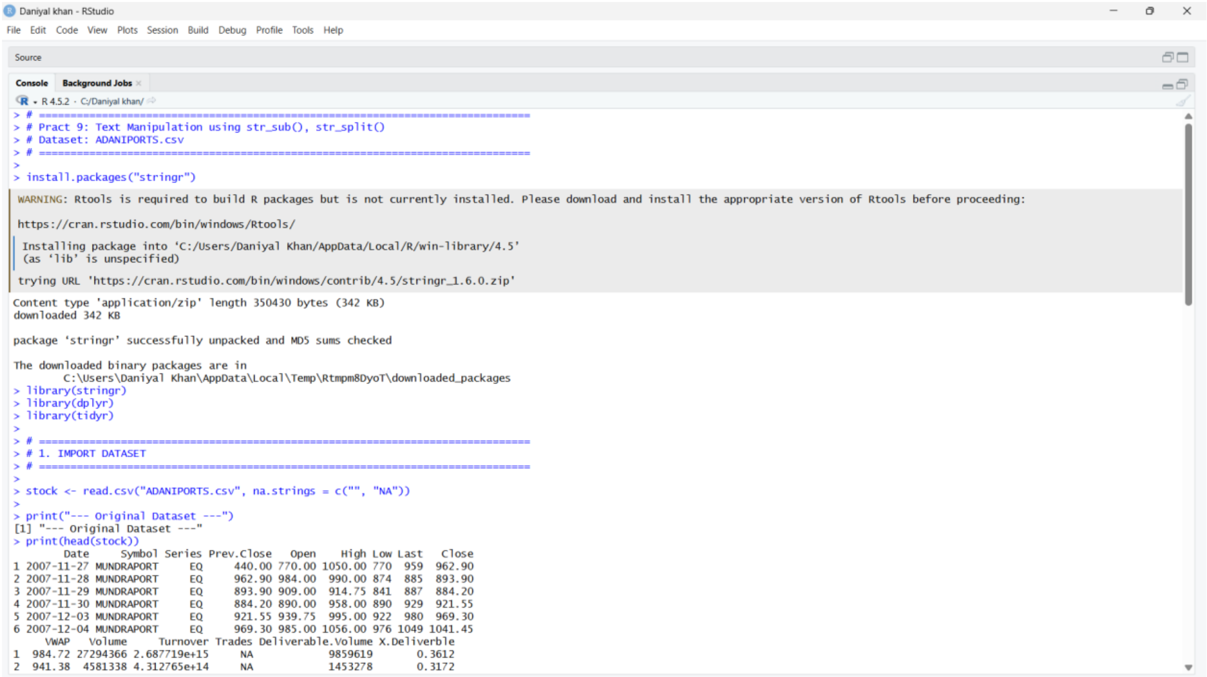

Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> avg_length <- mean(df$product_length, na.rm = TRUE)
> avg_depth  <- mean(df$product_depth, na.rm = TRUE)
> avg_width  <- mean(df$product_width, na.rm = TRUE)
>
> clean_replace <- df %>%
+   replace_na(list(
+     product_length = avg_length,
+     product_depth  = avg_depth,
+     product_width  = avg_width,
+     cluster_id     = "Unknown",
+     hierarchy1_id  = "Missing",
+     hierarchy2_id  = "Missing",
+     hierarchy3_id  = "Missing",
+     hierarchy4_id  = "Missing",
+     hierarchy5_id  = "Missing"
+   ))
>
> print("--- 3. Data After Replacing Missing Values ---")
[1] "--- 3. Data After Replacing Missing Values ---"
> print(head(clean_replace))
  product_id product_length product_depth product_width cluster_id
1      P0000            5.0            20            12    Unknown
2      P0001           13.5            22            20  cluster_5
3      P0002           22.0            40            22  cluster_0
4      P0004            2.0            13             4  cluster_3
5      P0005           16.0            30            16  cluster_9
6      P0006            8.5            15            15  cluster_0
  hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id hierarchy5_id
1           H00         H0004       H000401     H00040105   H0004010534
2           H01         H0105       H010501     H01050100   H0105010006
3           H03         H0315       H031508     H03150800   H0315080028
4           H03         H0314       H031405     H03140500   H0314050003
5           H03         H0312       H031211     H03121109   H0312110917
6           H03         H0316       H031608     H03160817   H0316081708
>
> # Check remaining NAs
> print("--- Remaining NAs After Cleaning ---")
[1] "--- Remaining NAs After Cleaning ---"
> print(colSums(is.na(clean_replace)))
    product_id product_length  product_depth  product_width     cluster_id
             0              0              0              0              0
 hierarchy1_id  hierarchy2_id  hierarchy3_id  hierarchy4_id  hierarchy5_id
             0              0              0              0              0
> # Contains: ID, Name, and Jan_Sales
```

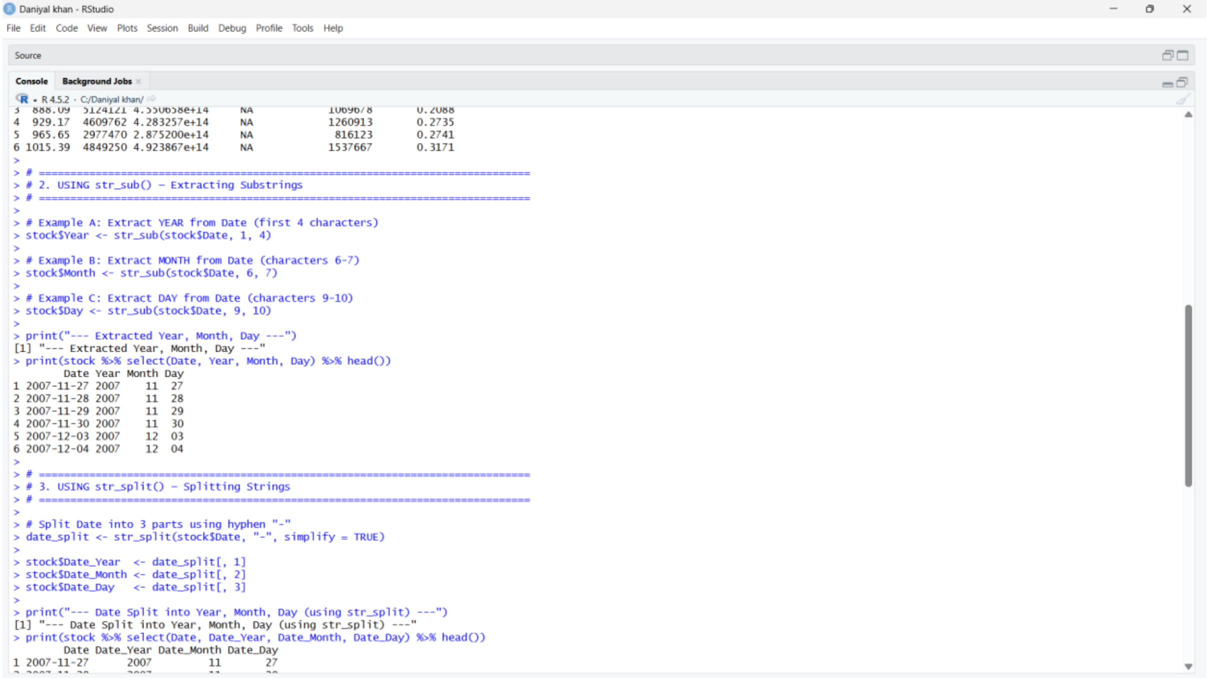
SHETH L.U.J COLLEGE

R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10

9:


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

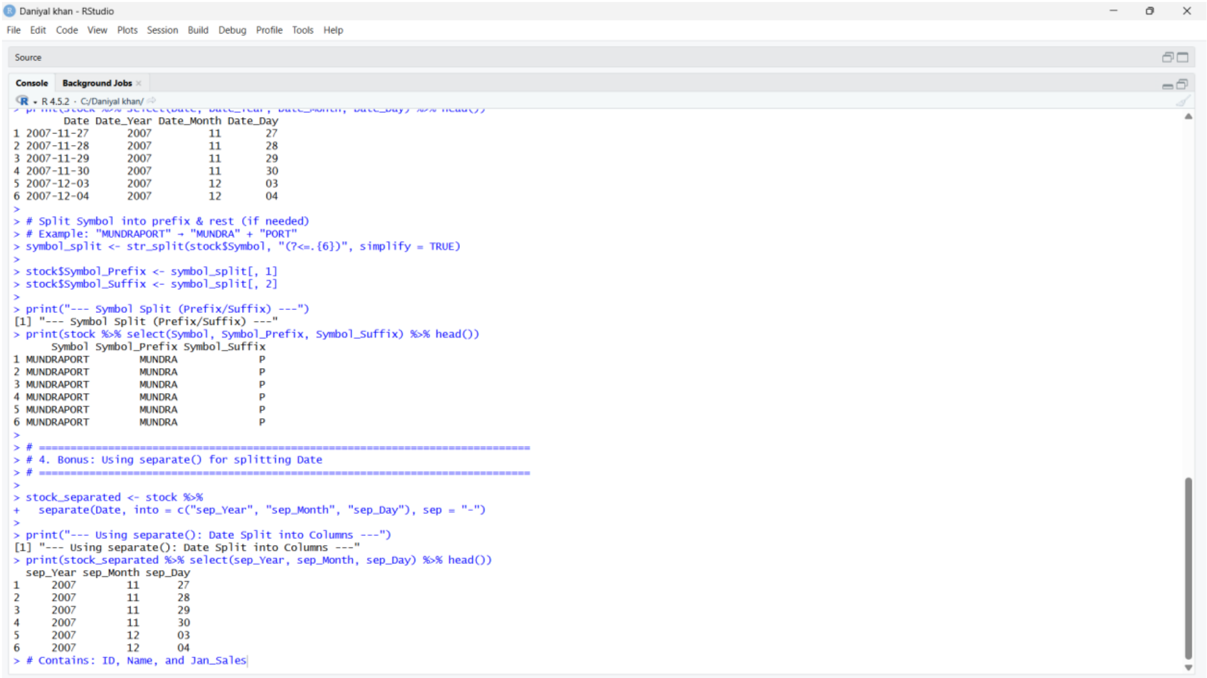
```
> # ============================================================================
> # Pract 9: Text Manipulation using str_sub(), str_split()
> # Dataset: ADANIPORTS.csv
> # ============================================================================
>
> install.packages("stringr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/

 Installing package into ‘C:/Users/Daniyal Khan/AppData/Local/R/win-library/4.5’
 (as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/stringr_1.6.0.zip'
Content type 'application/zip' length 350430 bytes (342 KB)
downloaded 342 KB

package ‘stringr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
	C:\Users\Daniyal Khan\AppData\Local\Temp\Rtmpm8DyoT\downloaded_packages
> library(stringr)
> library(dplyr)
> library(tidyr)
>
> # ============================================================================
> # 1. IMPORT DATASET
> # ============================================================================
>
> stock <- read.csv("ADANIPORTS.csv", na.strings = c("", "NA"))
>
> print("--- Original Dataset ---")
[1] "--- Original Dataset ---"
> print(head(stock))
        Date     Symbol Series Prev.Close   Open    High Low Last   Close
1 2007-11-27 MUNDRAPORT     EQ     440.00 770.00 1050.00 770  959  962.90
2 2007-11-28 MUNDRAPORT     EQ     962.90 984.00  990.00 874  885  893.90
3 2007-11-29 MUNDRAPORT     EQ     893.90 909.00  914.75 841  887  884.20
4 2007-11-30 MUNDRAPORT     EQ     884.20 890.00  958.00 890  929  921.55
5 2007-12-03 MUNDRAPORT     EQ     921.55 939.75  995.00 922  980  969.30
6 2007-12-04 MUNDRAPORT     EQ     969.30 985.00 1056.00 976 1049 1041.45
     VWAP   Volume     Turnover Trades Deliverable.Volume X.Deliverble
1  984.72 27294366 2.687719e+15     NA            9859619       0.3612
2  941.38  4581338 4.312765e+14     NA            1453278       0.3172
```


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
3  888.09  5124121 4.550658e+14     NA            1069678       0.2088
4  929.17  4609762 4.283257e+14     NA            1260913       0.2735
5  965.65  2977470 2.875200e+14     NA             816123       0.2741
6 1015.39  4849250 4.923867e+14     NA            1537667       0.3171
>
> # ============================================================================
> # 2. USING str_sub() - Extracting Substrings
> # ============================================================================
>
> # Example A: Extract YEAR from Date (first 4 characters)
> stock$Year <- str_sub(stock$Date, 1, 4)
>
> # Example B: Extract MONTH from Date (characters 6-7)
> stock$Month <- str_sub(stock$Date, 6, 7)
>
> # Example C: Extract DAY from Date (characters 9-10)
> stock$Day <- str_sub(stock$Date, 9, 10)
>
> print("--- Extracted Year, Month, Day ---")
[1] "--- Extracted Year, Month, Day ---"
> print(stock %>% select(Date, Year, Month, Day) %>% head())
        Date Year Month Day
1 2007-11-27 2007    11  27
2 2007-11-28 2007    11  28
3 2007-11-29 2007    11  29
4 2007-11-30 2007    11  30
5 2007-12-03 2007    12  03
6 2007-12-04 2007    12  04
>
> # ============================================================================
> # 3. USING str_split() - Splitting Strings
> # ============================================================================
>
> # Split Date into 3 parts using hyphen "-"
> date_split <- str_split(stock$Date, "-", simplify = TRUE)
>
> stock$Date_Year  <- date_split[, 1]
> stock$Date_Month <- date_split[, 2]
> stock$Date_Day   <- date_split[, 3]
>
> print("--- Date Split into Year, Month, Day (using str_split) ---")
[1] "--- Date Split into Year, Month, Day (using str_split) ---"
> print(stock %>% select(Date, Date_Year, Date_Month, Date_Day) %>% head())
        Date Date_Year Date_Month Date_Day
1 2007-11-27      2007         11       27
```

SHETH L.U.J COLLEGE

R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10

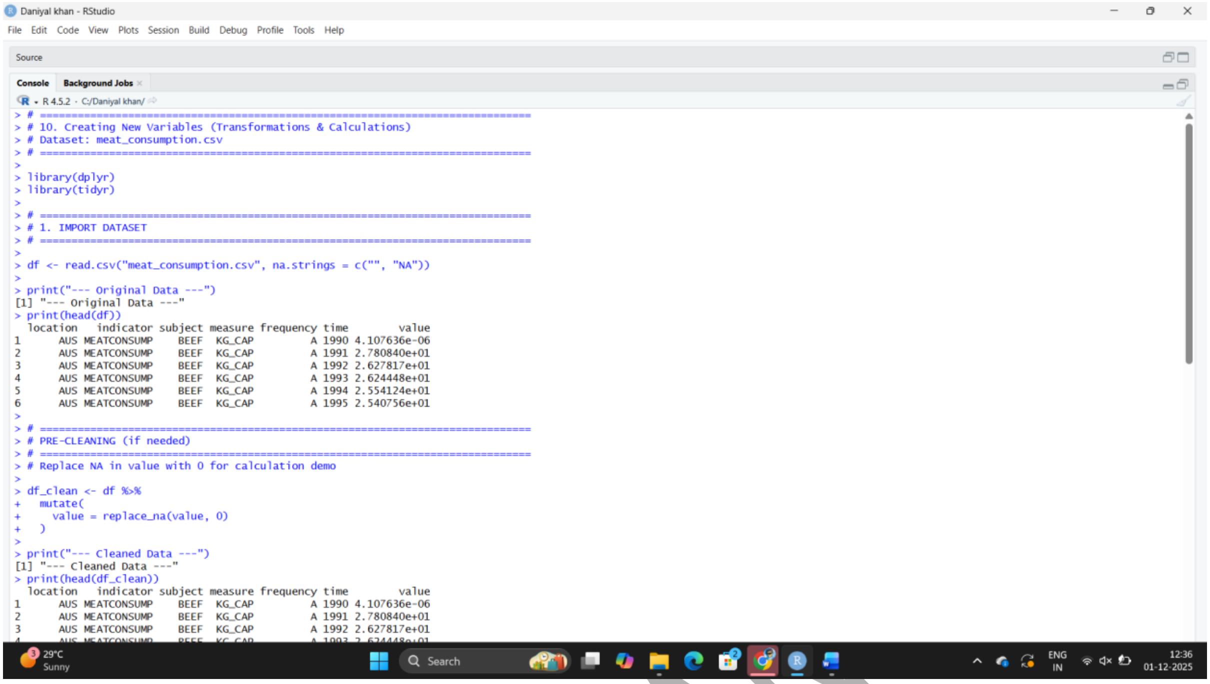

Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/

```
> print(stock %>% select(Date, Date_Year, Date_Month, Date_Day) %>% head())
        Date Date_Year Date_Month Date_Day
1 2007-11-27      2007         11       27
2 2007-11-28      2007         11       28
3 2007-11-29      2007         11       29
4 2007-11-30      2007         11       30
5 2007-12-03      2007         12       03
6 2007-12-04      2007         12       04
>
> # Split Symbol into prefix & rest (if needed)
> # Example: "MUNDRAPORT" → "MUNDRA" + "PORT"
> symbol_split <- str_split(stock$Symbol, "(?<=.{6})", simplify = TRUE)
>
> stock$Symbol_Prefix <- symbol_split[, 1]
> stock$Symbol_Suffix <- symbol_split[, 2]
>
> print("--- Symbol Split (Prefix/Suffix) ---")
[1] "--- Symbol Split (Prefix/Suffix) ---"
> print(stock %>% select(Symbol, Symbol_Prefix, Symbol_Suffix) %>% head())
      Symbol Symbol_Prefix Symbol_Suffix
1 MUNDRAPORT        MUNDRA             P
2 MUNDRAPORT        MUNDRA             P
3 MUNDRAPORT        MUNDRA             P
4 MUNDRAPORT        MUNDRA             P
5 MUNDRAPORT        MUNDRA             P
6 MUNDRAPORT        MUNDRA             P
>
> # ============================================================================
> # 4. Bonus: Using separate() for splitting Date
> # ============================================================================
>
> stock_separated <- stock %>%
+   separate(Date, into = c("sep_Year", "sep_Month", "sep_Day"), sep = "-")
>
> print("--- Using separate(): Date Split into Columns ---")
[1] "--- Using separate(): Date Split into Columns ---"
> print(stock_separated %>% select(sep_Year, sep_Month, sep_Day) %>% head())
  sep_Year sep_Month sep_Day
1     2007        11      27
2     2007        11      28
3     2007        11      29
4     2007        11      30
5     2007        12      03
6     2007        12      04
> # Contains: ID, Name, and Jan_Sales
```

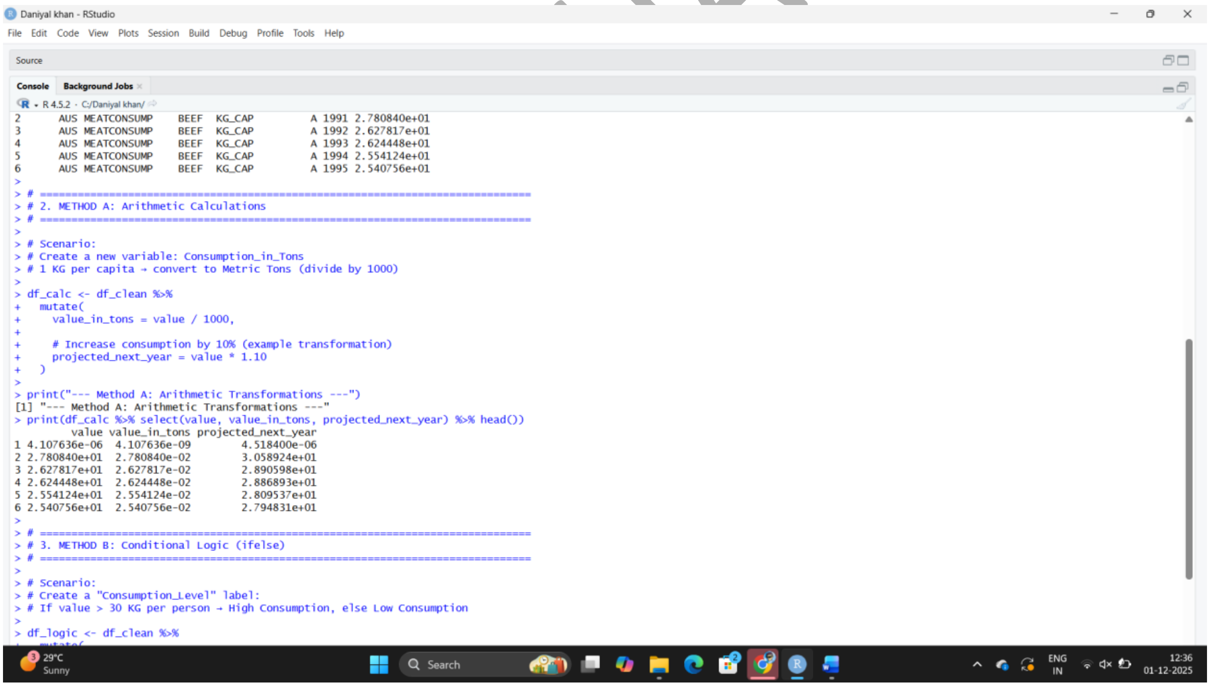
SHETH L.U.J COLLEGE

R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10

10:


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/
29°C Sunny
Search
ENG IN
12:36 01-12-2025

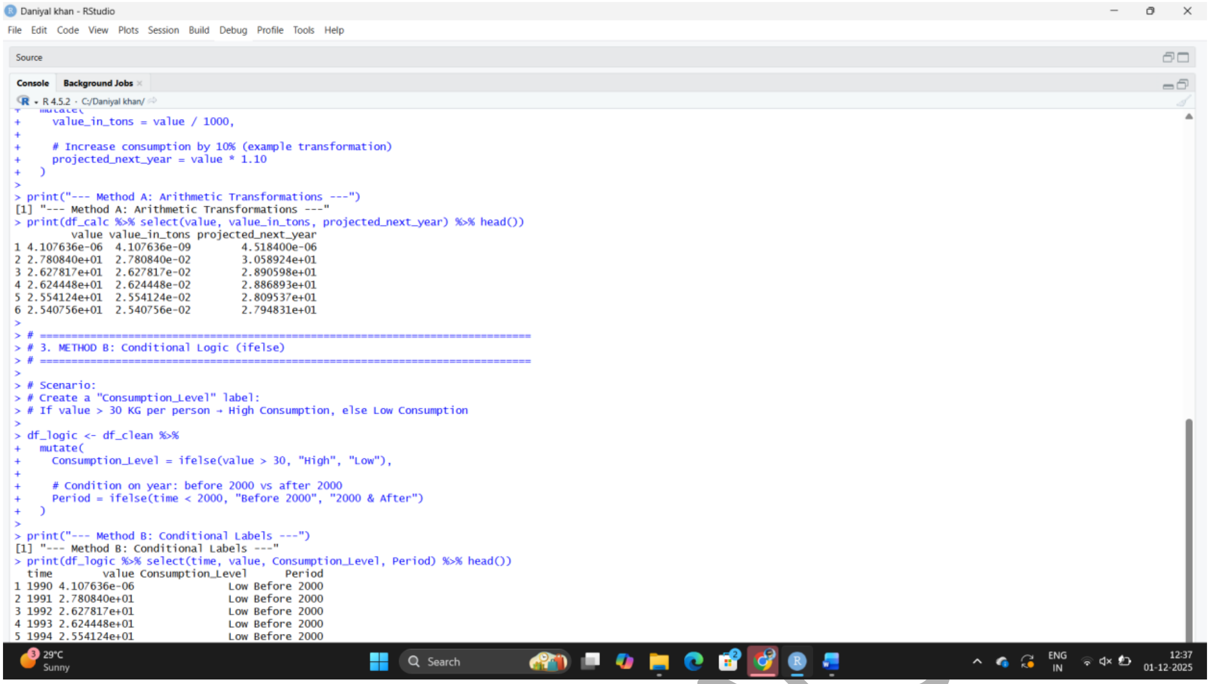
```
> # ============================================================================
> # 10. Creating New Variables (Transformations & Calculations)
> # Dataset: meat_consumption.csv
> # ============================================================================
>
> library(dplyr)
> library(tidyr)
>
> # ============================================================================
> # 1. IMPORT DATASET
> # ============================================================================
>
> df <- read.csv("meat_consumption.csv", na.strings = c("", "NA"))
>
> print("--- Original Data ---")
[1] "--- Original Data ---"
> print(head(df))
  location   indicator subject measure frequency time        value
1      AUS MEATCONSUMP    BEEF  KG_CAP         A 1990 4.107636e-06
2      AUS MEATCONSUMP    BEEF  KG_CAP         A 1991 2.780840e+01
3      AUS MEATCONSUMP    BEEF  KG_CAP         A 1992 2.627817e+01
4      AUS MEATCONSUMP    BEEF  KG_CAP         A 1993 2.624448e+01
5      AUS MEATCONSUMP    BEEF  KG_CAP         A 1994 2.554124e+01
6      AUS MEATCONSUMP    BEEF  KG_CAP         A 1995 2.540756e+01
>
> # ============================================================================
> # PRE-CLEANING (if needed)
> # ============================================================================
> # Replace NA in value with 0 for calculation demo
>
> df_clean <- df %>%
+   mutate(
+     value = replace_na(value, 0)
+   )
>
> print("--- Cleaned Data ---")
[1] "--- Cleaned Data ---"
> print(head(df_clean))
  location   indicator subject measure frequency time        value
1      AUS MEATCONSUMP    BEEF  KG_CAP         A 1990 4.107636e-06
2      AUS MEATCONSUMP    BEEF  KG_CAP         A 1991 2.780840e+01
3      AUS MEATCONSUMP    BEEF  KG_CAP         A 1992 2.627817e+01
4      AUS MEATCONSUMP    BEEF  KG_CAP         A 1993 2.624448e+01
```


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/
29°C Sunny
Search
ENG IN
12:36 01-12-2025

```
2      AUS MEATCONSUMP    BEEF  KG_CAP         A 1991 2.780840e+01
3      AUS MEATCONSUMP    BEEF  KG_CAP         A 1992 2.627817e+01
4      AUS MEATCONSUMP    BEEF  KG_CAP         A 1993 2.624448e+01
5      AUS MEATCONSUMP    BEEF  KG_CAP         A 1994 2.554124e+01
6      AUS MEATCONSUMP    BEEF  KG_CAP         A 1995 2.540756e+01
>
> # ============================================================================
> # 2. METHOD A: Arithmetic Calculations
> # ============================================================================
>
> # Scenario:
> # Create a new variable: Consumption_in_Tons
> # 1 KG per capita → convert to Metric Tons (divide by 1000)
>
> df_calc <- df_clean %>%
+   mutate(
+     value_in_tons = value / 1000,
+
+     # Increase consumption by 10% (example transformation)
+     projected_next_year = value * 1.10
+   )
>
> print("--- Method A: Arithmetic Transformations ---")
[1] "--- Method A: Arithmetic Transformations ---"
> print(df_calc %>% select(value, value_in_tons, projected_next_year) %>% head())
         value value_in_tons projected_next_year
1 4.107636e-06  4.107636e-09        4.518400e-06
2 2.780840e+01  2.780840e-02        3.058924e+01
3 2.627817e+01  2.627817e-02        2.890598e+01
4 2.624448e+01  2.624448e-02        2.886893e+01
5 2.554124e+01  2.554124e-02        2.809537e+01
6 2.540756e+01  2.540756e-02        2.794831e+01
>
> # ============================================================================
> # 3. METHOD B: Conditional Logic (ifelse)
> # ============================================================================
>
> # Scenario:
> # Create a "Consumption_Level" label:
> # If value > 30 KG per person → High Consumption, else Low Consumption
>
> df_logic <- df_clean %>%
+   mutate(
```

SHETH L.U.J COLLEGE

R-PROGRAMMING OUTOUT PRAC-6,7,8,9,10


Daniyal khan - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console
Background Jobs
R 4.5.2 · C:/Daniyal khan/
29°C
Sunny
Search
ENG
IN
12:37
01-12-2025

```
+   mutate(
+     value_in_tons = value / 1000,
+
+     # Increase consumption by 10% (example transformation)
+     projected_next_year = value * 1.10
+   )
>
> print("--- Method A: Arithmetic Transformations ---")
[1] "--- Method A: Arithmetic Transformations ---"
> print(df_calc %>% select(value, value_in_tons, projected_next_year) %>% head())
         value value_in_tons projected_next_year
1 4.107636e-06  4.107636e-09        4.518400e-06
2 2.780840e+01  2.780840e-02        3.058924e+01
3 2.627817e+01  2.627817e-02        2.890598e+01
4 2.624448e+01  2.624448e-02        2.886893e+01
5 2.554124e+01  2.554124e-02        2.809537e+01
6 2.540756e+01  2.540756e-02        2.794831e+01
>
> # ==============================================================================
> # 3. METHOD B: Conditional Logic (ifelse)
> # ==============================================================================
>
> # Scenario:
> # Create a "Consumption_Level" label:
> # If value > 30 KG per person → High Consumption, else Low Consumption
>
> df_logic <- df_clean %>%
+   mutate(
+     Consumption_Level = ifelse(value > 30, "High", "Low"),
+
+     # Condition on year: before 2000 vs after 2000
+     Period = ifelse(time < 2000, "Before 2000", "2000 & After")
+   )
>
> print("--- Method B: Conditional Labels ---")
[1] "--- Method B: Conditional Labels ---"
> print(df_logic %>% select(time, value, Consumption_Level, Period) %>% head())
  time        value Consumption_Level      Period
1 1990 4.107636e-06               Low Before 2000
2 1991 2.780840e+01               Low Before 2000
3 1992 2.627817e+01               Low Before 2000
4 1993 2.624448e+01               Low Before 2000
5 1994 2.554124e+01               Low Before 2000
```