

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

### AIM: 11. Reshaping data using pivot\_longer() and pivot\_wider() (R).

The image displays two side-by-side screenshots of the RStudio interface, showing the process of reshaping data using the `tidyverse` packages.

**Screenshot 1 (Top):**

- Console:**

```
R > install.packages("tidyverse")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of R tools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/itlab/AppData/Local/R/win-library/4.5'
(as 'lib' is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.1.zip'
Content type 'application/zip' length 1276983 bytes (1.2 MB)
downloaded 1.2 MB

package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/itlab/AppData/Local/RTemp/Rtmpg3qkBI/downloaded_packages
> library(tidyverse)
> library(dplyr)
```
- Data View:** Shows the global environment with objects: df (4573 obs. of 3 variables), df\_long (9146 obs. of 3 variables), df\_wide (4047 obs. of 3 variables), and spotify (4573 obs. of 15 variables).
- File Bar:** Shows standard file navigation options like Files, Plots, Packages, Help, Viewer, and Presentation.

**Screenshot 2 (Bottom):**

- Console:**

```
R > in addition: warning message:
in file(file, "rt"):
cannot open file 'C:/Users/itlab/OneDrive/Documents/S105/DATASET': Permission denied

> head(spotify)
Error: object 'spotify' not found
```
- Data View:** Shows the global environment with objects: df (4573 obs. of 3 variables), df\_long (9146 obs. of 3 variables), df\_wide (4047 obs. of 3 variables), and spotify (4573 obs. of 15 variables).
- File Bar:** Shows standard file navigation options like Files, Plots, Packages, Help, Viewer, and Presentation.

In both screenshots, the system tray at the bottom right shows the date as 08-12-2025 and the time as 11:02. The status bar at the bottom of each RStudio window also displays the same date and time.

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/Function | Addins |
Environment History Connections Tutorial
R Global Environment
Data
df 4573 obs. of 3 variables
df_long 9146 obs. of 3 variables
df_wide 4047 obs. of 3 variables
spotify 4573 obs. of 15 variables
List | ↻
R 08-12-2025
Source
> head(df)
#> # A tibble: 9,146 × 3
#>   track_name    track_popularity artist_popularity
#>   <chr>                <dbl>                  <dbl>
#> 1 Trippy Mane (ft. Project Pat)        77                   0
#> 2 OMG!                               0                   64
#> 3 Hard 2 Find                         4                   48
#> 4 Still Get Like That (ft. Project Pat & Starrah) 30                   77
#> 5 ride me like a harley                 0                   48
#> 6 BLEED                             2                   46
#> #> df_long <- df %>%
#> #> pivot_longer(
#> #>   cols = c(track_popularity, artist_popularity),
#> #>   names_to = "subject",
#> #>   values_to = "score"
#> #> )
#> #> df_long
#> #> # A tibble: 9,146 × 3
#> #>   track_name      subject      score
#> #>   <chr>          <chr>       <dbl>
#> #> 1 Trippy Mane (ft. Project Pat) track_popularity 0
#> #> 2 Trippy Mane (ft. Project Pat) artist_popularity 77
#> #> 3 OMG!                      track_popularity 0
#> #> 4 OMG!                      artist_popularity 64
#> #> 5 Hard 2 Find                track_popularity 4
#> #> 6 Hard 2 Find                artist_popularity 48
#> #> 7 Still Get Like That (ft. Project Pat & Starrah) track_popularity 30
#> #> 8 Still Get Like That (ft. Project Pat & Starrah) artist_popularity 77
#> #> 9 ride me like a harley        track_popularity 0
#> #> 10 ride me like a harley       artist_popularity 48
#> #> # i use 'print(n ...)' to see more rows
#> #> df_wide <- df_long %>%
#> #> pivot_wider(
#> #>   names_from = subject,
#> #>   values_from = score
#> #> )
warning message:
values from 'score' are not uniquely identified; output will contain list-cols.
• use `values_fn = list` to suppress this warning.
• use `values_fn = `summary_fn`` to summarise duplicates.
• use the following dplyr code to identify duplicates.
  (data) >
  dplyr::summarise(n = dplyr::n(), .by = c(track_name, subject)) |>
  dplyr::filter(n > 1)
  
```

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/Function | Addins |
Environment History Connections Tutorial
R Global Environment
Data
df 4573 obs. of 3 variables
df_long 9146 obs. of 3 variables
df_wide 4047 obs. of 3 variables
spotify 4573 obs. of 15 variables
List | ↻
R 08-12-2025
Source
> head(df)
#> # A tibble: 9,146 × 3
#>   track_name    track_popularity artist_popularity
#>   <chr>                <dbl>                  <dbl>
#> 1 Trippy Mane (ft. Project Pat)        77                   0
#> 2 OMG!                               0                   64
#> 3 Hard 2 Find                         4                   48
#> 4 Still Get Like That (ft. Project Pat & Starrah) 30                   77
#> 5 ride me like a harley                 0                   48
#> 6 BLEED                             2                   46
#> #> # i use 'print(n ...)' to see more rows
#> #> df_wide <- df_long %>%
#> #> pivot_wider(
#> #>   names_from = subject,
#> #>   values_from = score
#> #> )
warning message:
values from 'score' are not uniquely identified; output will contain list-cols.
• use `values_fn = list` to suppress this warning.
• use `values_fn = `summary_fn`` to summarise duplicates.
• use the following dplyr code to identify duplicates.
  (data) >
  dplyr::summarise(n = dplyr::n(), .by = c(track_name, subject)) |>
  dplyr::filter(n > 1)
  
```

> df\_wide

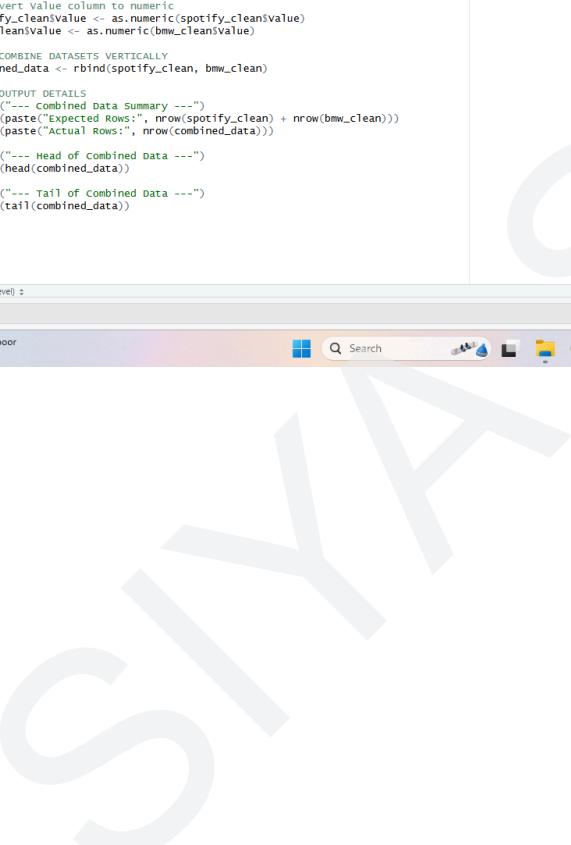
```

#> #> # A tibble: 4,047 × 3
#> #>   track_name      track_popularity artist_popularity
#> #>   <chr>          <list>       <list>
#> #> 1 Trippy Mane (ft. Project Pat) <chr [1]>    <chr [1]>
#> #> 2 OMG!                      <chr [1]>    <chr [1]>
#> #> 3 Hard 2 Find                <chr [1]>    <chr [1]>
#> #> 4 Still Get Like That (ft. Project Pat & Starrah) <chr [1]>    <chr [1]>
#> #> 5 ride me like a harley        <chr [1]>    <chr [1]>
#> #> 6 BLEED                     <chr [1]>    <chr [1]>
#> #> 7 Te Procuru na Cidade      <chr [1]>    <chr [1]>
#> #> 8 come closer                 <chr [2]>    <chr [2]>
#> #> 9 Cupido Vagabundo          <chr [1]>    <chr [1]>
#> #> 10 LET'S GO!                  <chr [1]>    <chr [1]>
#> #> # i use 'print(n ...)' to see more rows
#> #> 
```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

**AIM: 12. Combining datasets vertically (concatenation) using rbind() (R). Write code toCombining datasets vertically (concatenation) using rbind() in R studio.**

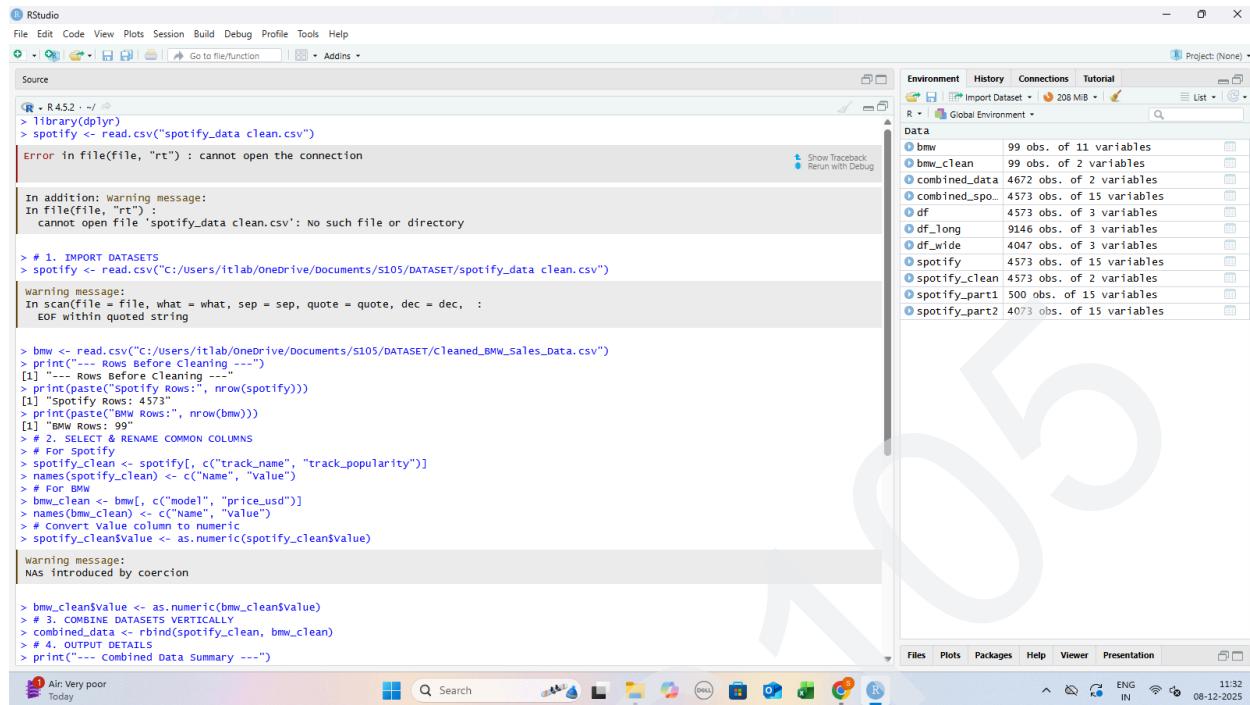


A screenshot of the RStudio interface showing R code for combining datasets vertically using the rbind() function. The code reads two CSV files ('spotify\_data\_clean.csv' and 'cleaned\_BMW\_Sales\_Data.csv'), prints their row counts, renames columns, converts a value column to numeric, and then concatenates them vertically. The resulting dataset is named 'combined\_data'. The RStudio environment pane shows various objects like 'bmw', 'bmw\_clean', and 'combined\_data' with their respective sizes and variable counts.

```
1 library(dplyr)
2
3 # 1. IMPORT DATASETS
4 spotify_clean <- read.csv("c:/users/itlab/onedrive/documents/s105/dataset/spotify_data_clean.csv")
5 bmw_clean <- read.csv("c:/users/itlab/onedrive/documents/s105/dataset/cleaned_BMW_Sales_Data.csv")
6
7 print("---- Rows Before Cleaning ----")
8 print(paste("Spotify Rows:", nrow(spotify)))
9 print(paste("BMW Rows:", nrow(bmw)))
10
11 # 2. SELECT & RENAME COMMON COLUMNS
12 # For Spotify
13 spotify_clean <- spotify[, c("track_name", "track_popularity")]
14 names(spotify_clean) <- c("Name", "Value")
15
16 # For BMW
17 bmw_clean <- bmw[, c("model", "price_usd")]
18 names(bmw_clean) <- c("Name", "Value")
19
20 # Convert Value column to numeric
21 spotify_clean$Value <- as.numeric(spotify_clean$Value)
22 bmw_clean$Value <- as.numeric(bmw_clean$Value)
23
24 # 3. COMBINE DATASETS VERTICALLY
25 combined_data <- rbind(spotify_clean, bmw_clean)
26
27 # 4. OUTPUT DETAILS
28 print("---- Combined Data Summary ----")
29 print(paste("Expected Rows:", nrow(spotify_clean) + nrow(bmw_clean)))
30 print(paste("Actual Rows:", nrow(combined_data)))
31
32 print("---- Head of Combined Data ----")
33 print(head(combined_data))
34
35 print("---- Tail of Combined Data ----")
36 print(tail(combined_data))
37
```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R



The screenshot shows the RStudio interface with two sessions. The top session is titled 'R 4.5.2 · ~' and contains R code for reading CSV files, performing data cleaning, and combining datasets. It includes comments explaining the steps: # 1. IMPORT DATASETS, # 2. SELECT & RENAME COMMON COLUMNS, # 3. COMBINE DATASETS VERTICALLY, and # 4. OUTPUT DETAILS. The bottom session is also titled 'R 4.5.2 · ~' and continues the process, showing the creation of a 'combined\_data' dataset and printing its head and tail. Both sessions show a list of variables in the 'Data' pane on the right.

```

# 1. IMPORT DATASETS
> spotify <- read.csv("spotify_data_clean.csv")
Error in file(file, "rt") : cannot open the connection
In addition: warning message:
In file(file, "rt") :
  cannot open file 'spotify_data_clean.csv': no such file or directory

# 2. SELECT & RENAME COMMON COLUMNS
# For Spotify
> spotify_clean <- spotify[, c("track_name", "track_popularity")]
> names(spotify_clean) <- c("Name", "Value")
# For BMW
> bmw_clean <- bmw[, c("model", "price_usd")]
> names(bmw_clean) <- c("Name", "Value")
# Convert Value column to numeric
> spotify_clean$value <- as.numeric(spotify_clean$value)

warning message:
NAS introduced by coercion

# 3. COMBINE DATASETS VERTICALLY
> combined_data <- rbind(spotify_clean, bmw_clean)
# 4. OUTPUT DETAILS
> print("---- Combined Data Summary ---")

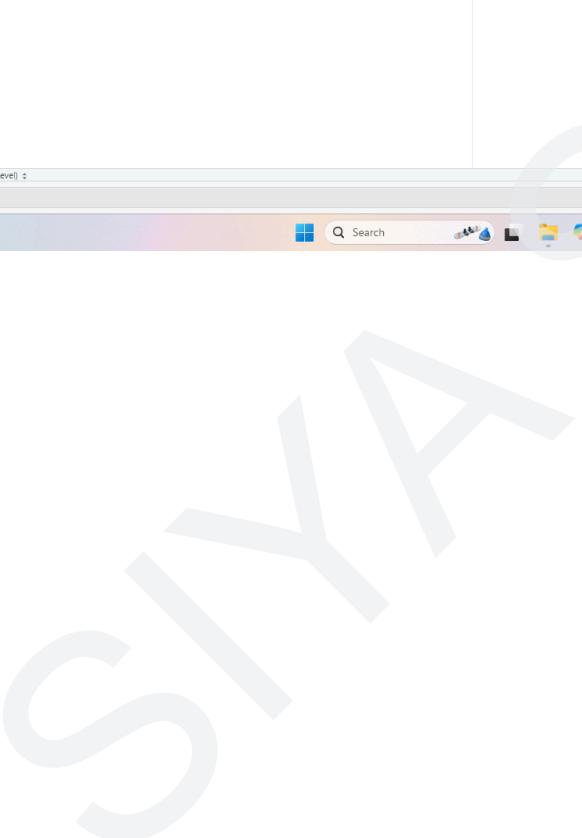
# 1. Combined Data Summary
[1] "---- Combined Data Summary ---"
> print(combined_data)
[1] "Expected Rows: 4672"
> print(paste("Actual Rows:", nrow(combined_data)))
[1] "Actual Rows: 4672"
> print("---- Head of Combined Data ---")
[1] "---- Head of Combined Data ---"
> print(head(combined_data))
      Name      value
1 Trippy Mane (ft. Project Pat) 0
2 Still Get Like That (ft. Project Pat & Starrah) 1
3 Hard 2 Find 4
4 Still Get Like That (ft. Project Pat & Starrah) 30
5 ride me like a harley 0
6 BLEED 2
> print("---- Tail of Combined Data ---")
[1] "---- Tail of Combined Data ---"
> print(tail(combined_data))
      Name      value
4667 13 63285
4668 13 63696
4669 X6 35951
4670 X5 95900
4671 18 44846
4672 X5 95908

```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

### AIM:13 Identifying and handling duplicates using distinct() (R).



A screenshot of the RStudio interface showing an R script for identifying and handling duplicates. The script uses the dplyr package to manipulate a dataset named 'orders\_df'. It first prints the original dataset, then creates a report of duplicate rows, filters out rows where the count is greater than 1, and finally prints unique customers after removing exact duplicates. The RStudio environment pane shows various datasets loaded, and the bottom status bar indicates the date and time.

```
1 library(dplyr)
2
3 orders_df <- data.frame(
4   OrderID = c(201, 202, 203, 204, 201, 204),
5   Customer = c("Reema", "Parth", "Parth", "Sanya", "Dinesh", "Reema", "Amey"),
6   Product = c("Charger", "Tab", "Phone", "Tab", "Monitor", "Laptop", "Mouse")
7 )
8
9 print("--- 1. Original Dataset (Note 7 rows) ---")
10 print(orders_df)
11 duplicates_report <- orders_df %>%
12   group_by(OrderID, Customer, Product) %>%
13   count() %>%
14   filter(n > 1)
15 print("--- 2. Identification Report (Rows that are duplicated) ---")
16 print(duplicates_report)
17
18 clean_exact <- orders_df %>%
19   distinct()
20 print("--- 3. Removed Exact Duplicates (distinct) ---")
21 print(clean_exact)
22
23 unique_customers <- orders_df %>%
24   distinct(Customer, .keep_all = TRUE)
25 print("--- 4. Unique Customers Only (Partial Duplicates removed) ---")
26
27 print(unique_customers)
```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

```
R > R452 ~/ ~
> library(dplyr)
> orders_df <- data.frame(
+   OrderID = c(201, 202, 202, 203, 204, 201, 204),
+   Customer = c("Reema", "Parth", "Parth", "Saniya", "Dinesh", "Reema", "Amey"),
+   Product = c("Charger", "Tab", "Phone", "Tab", "Monitor", "Laptop", "Mouse")
+ )
> print("--- 1. original dataset (Note 7 rows ---")
[1] "--- 1. original dataset (Note 7 rows ---"
> print(orders_df)
  OrderID Customer Product
1    201      Reema Charger
2    202      Parth   Tab
3    202      Parth   Phone
4    203     Saniya   Tab
5    204     Dinesh Monitor
6    201      Reema Laptop
7    204      Amey   Mouse
> duplicates_report <- orders_df %>%
+   group_by(OrderID, Customer, Product) %>%
+   count() %>% # Counts occurrences
+   filter(n > 1) # Keeps only rows that appear more than once
> duplicates_report <- orders_df %>%
+   group_by(OrderID, Customer, Product) %>%
+   count() %>%
+   filter(n > 1)
> print("--- 2. Identification Report (Rows that are duplicated) ---")
[1] "--- 2. Identification Report (Rows that are duplicated) ---"
> print(duplicates_report)
# A tibble: 0 x 4
# Groups:   OrderID, Customer, Product [0]
# i 4 variables: OrderID <dbl>, Customer <chr>, Product <chr>, n <int>
> clean_exact <- orders_df %>%
+   distinct()
> print("--- 3. Removed Exact Duplicates (distinct) ---")
[1] "--- 3. Removed Exact Duplicates (distinct) ---"
> print(clean_exact)
  OrderID Customer Product
1    201      Reema Charger
2    202      Parth   Tab
3    202      Parth   Phone
4    203     Saniya   Tab
5    204     Dinesh Monitor
6    201      Reema Laptop
7    204      Amey   Mouse
```

```
R > R452 ~/ ~
> clean_exact %>%
+   distinct(Customer, keep_all = TRUE)
> print("--- 4. Unique Customers only (Partial duplicates removed) ---")
[1] "--- 4. Unique Customers only (Partial duplicates removed) ---"
> print(unique_customers)
  OrderID Customer Product
1    201      Reema Charger
2    202      Parth   Tab
3    203     Saniya   Tab
4    204     Dinesh Monitor
5    204      Amey   Mouse
```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

### AIM:14 Extracting date components using lubridate:: functions (R).

The screenshot displays two RStudio sessions side-by-side.

**Top Session (Script View):**

```

1 install.packages("lubridate")
2 library(lubridate)
3 library(dplyr)
4
5 movies_df <- data.frame(
+   Movie_ID = 1:5,
+   Release_Date = c("2020-09-04", "2021-12-17", "2022-05-06", "2023-07-21", "2024-11-08")
)
9
10 print("--- original dataset ---")
11 print(movies_df)
12
13 processed_movies <- movies_df %>%
14   mutate(
15     Actual_Date = ymd(Release_Date),
16     Year = year(Actual_Date),
17     Month_No = month(Actual_Date),
18     Month_Name = month(Actual_Date, label = TRUE, abbr = FALSE),
19     Day = day(Actual_Date),
20     Weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE),
21     Quarter = quarter(Actual_Date),
22     Day_of_Year = yday(Actual_Date)
23   )
24
25 print("--- Movies with Extracted Date Components ---")
26 print(processed_movies)
27
28 current_time <- now()
29
30 print("---- System Date-Time Extraction ----")
31 print(paste("Current Year:", year(current_time)))
32 print(paste("Current Month:", month(current_time)))
33 print(paste("Current Day:", day(current_time)))
34 print(paste("Current Hour:", hour(current_time)))
35 print(paste("Current Minute:", minute(current_time)))
36

```

**Bottom Session (Console View):**

```

R > install.packages("lubridate")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of R tools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/it1lab/AppData/Local/R/win-library/4.5'
(as 'lib' is unspecified)

also installing the dependency 'timechange'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/timechange_0.3.0.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/lubridate_1.9.4.zip'
package 'timechange' successfully unpacked and MD5 sums checked
package 'lubridate' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\it1lab\AppData\Local\Temp\Rtmpg3qkBI\downloaded_packages
> library(lubridate)

Attaching package: 'lubridate'

The following objects are masked from 'package:base':
  date, intersect, setdiff, union

> library(dplyr)
> movies_df <- data.frame(
+   Movie_ID = 1:5,
+   Release_Date = c("2020-09-04", "2021-12-17", "2022-05-06", "2023-07-21", "2024-11-08")
+ )
> movies_df <- data.frame(
+   Movie_ID = 1:5,
+   Release_Date = c("2020-09-04", "2021-12-17", "2022-05-06", "2023-07-21", "2024-11-08")
+ )
> print("--- original dataset ---")
[1] "... original dataset ..."
> print(movies_df)
  Movie_ID Release_Date
1          1 2020-09-04
2          2 2021-12-17
3          3 2022-05-06
4          4 2023-07-21
5          5 2024-11-08

```

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

RStudio Environment View showing the following R code:

```

R - R452 ~/ 
[1] "---- original dataset ---"
> print(movies_df)
  Movie_ID Release_Date
1          1 2020-09-04
2          2 2021-12-17
3          3 2022-05-06
4          4 2023-07-21
5          5 2024-11-08

> processed_movies <- movies_df %>%
+   mutate(
+     Actual_Date = ymd(Release_Date),      # Convert character to Date format
+     Year = year(Actual_Date),             # Extract year
+     Month_No = month(Actual_Date),        # Extract month number (1-12)
+     Month_Name = month(Actual_Date, lable = TRUE, abbr = FALSE), # Full month name
+     Day = day(Actual_Date),               # Date number (1-31)
+     weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE), # Full weekday name
+     Quarter = quarter(Actual_Date),       # Quarter of the year (1-4)
+     Day_of_Year = yday(Actual_Date)       # Day number in year (1-366)
  )
> print("---- Movies with Extracted Date Components ---")
[1] "---- Movies with Extracted Date Components ---"
> processed_movies <- movies_df %>%
+   mutate(
+     Actual_Date = ymd(Release_Date),
+     Year = year(Actual_Date),
+     Month_No = month(Actual_Date),
+     Month_Name = month(Actual_Date, lable = TRUE, abbr = FALSE),
+     Day = day(Actual_Date),
+     weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE),
+     Quarter = quarter(Actual_Date),
+     Day_of_Year = yday(Actual_Date)
  )
> print("---- Movies with Extracted Date Components ---")
[1] "---- Movies with Extracted Date Components ---"
> print(processed_movies)
  Movie_ID Release_Date Actual_Date Year Month_No Month_Name Day weekday_Name Quarter Day_of_Year
1          1 2020-09-04 2020-09-04 2020         9 September    4 Friday
2          2 2021-12-17 2021-12-17 2021        12 December   17 Friday
3          3 2022-05-06 2022-05-06 2022         5 May           6 Friday
4          4 2023-07-21 2023-07-21 2023        7 July          21 Friday
5          5 2024-11-08 2024-11-08 2024        11 November    8 Friday
> current_time <- now()
> print("---- System Date-Time Extraction ---")
[1] "---- System Date-Time Extraction ---"
  
```

The RStudio interface includes the Global Environment pane listing various datasets like `bmw`, `df`, and `movies\_df` with their respective dimensions.

RStudio Environment View showing the same R code as the previous screenshot, with additional print statements for system date-time extraction:

```

R - R452 ~/ 
[1] "---- original dataset ---"
> print(movies_df)
  Movie_ID Release_Date
1          1 2020-09-04
2          2 2021-12-17
3          3 2022-05-06
4          4 2023-07-21
5          5 2024-11-08

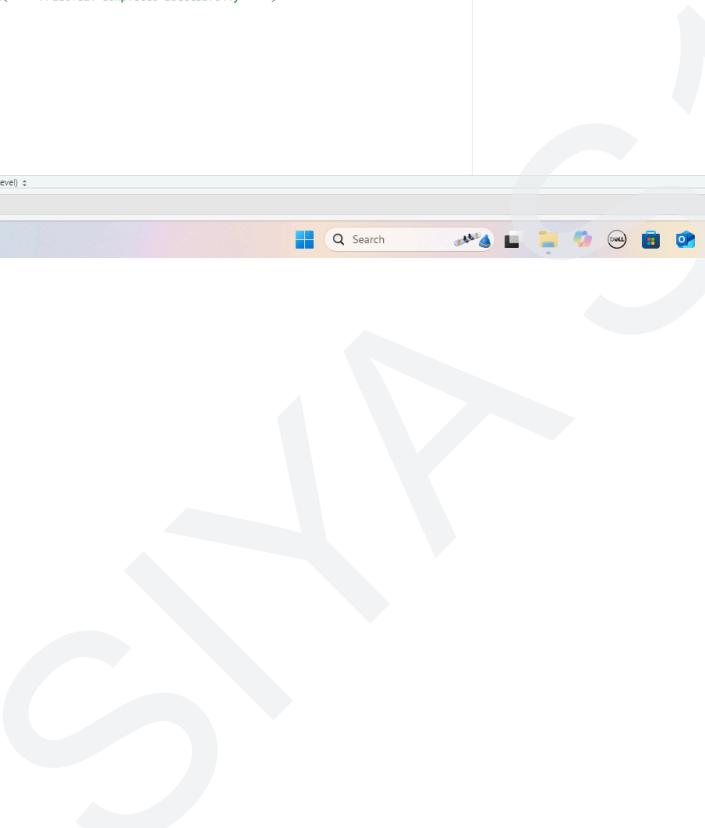
> processed_movies <- movies_df %>%
+   mutate(
+     Actual_Date = ymd(Release_Date),
+     Year = year(Actual_Date),
+     Month_No = month(Actual_Date),
+     Month_Name = month(Actual_Date, lable = TRUE, abbr = FALSE),
+     Day = day(Actual_Date),
+     weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE),
+     Quarter = quarter(Actual_Date),
+     Day_of_Year = yday(Actual_Date)
  )
> print("---- Movies with Extracted Date Components ---")
[1] "---- Movies with Extracted Date Components ---"
> processed_movies <- movies_df %>%
+   mutate(
+     Actual_Date = ymd(Release_Date),
+     Year = year(Actual_Date),
+     Month_No = month(Actual_Date),
+     Month_Name = month(Actual_Date, lable = TRUE, abbr = FALSE),
+     Day = day(Actual_Date),
+     weekday_Name = wday(Actual_Date, label = TRUE, abbr = FALSE),
+     Quarter = quarter(Actual_Date),
+     Day_of_Year = yday(Actual_Date)
  )
> print("---- Movies with Extracted Date Components ---")
[1] "---- Movies with Extracted Date Components ---"
> print(processed_movies)
  Movie_ID Release_Date Actual_Date Year Month_No Month_Name Day weekday_Name Quarter Day_of_Year
1          1 2020-09-04 2020-09-04 2020         9 September    4 Friday
2          2 2021-12-17 2021-12-17 2021        12 December   17 Friday
3          3 2022-05-06 2022-05-06 2022         5 May           6 Friday
4          4 2023-07-21 2023-07-21 2023        7 July          21 Friday
5          5 2024-11-08 2024-11-08 2024        11 November    8 Friday
> current_time <- now()
> print("---- System Date-Time Extraction ---")
[1] "---- System Date-Time Extraction ---"
> print(paste("Current Year:", year(current_time)))
[1] "Current Year: 2025"
> print(paste("Current Month:", month(current_time)))
[1] "Current Month: 12"
> print(paste("Current Day:", day(current_time)))
[1] "Current Day: 8"
> print(paste("Current Hour:", hour(current_time)))
[1] "Current Hour: 11"
> print(paste("Current Minute:", minute(current_time)))
[1] "Current Minute: 53"
  
```

The RStudio interface includes the Global Environment pane listing various datasets like `bmw`, `df`, and `movies\_df` with their respective dimensions.

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

### AIM:15 Generating basic summaries using str() or summary() (R).



A screenshot of an RStudio interface. The code editor shows an R script with the following content:

```
1 marks_df <- read.csv("C:/users/itlab/onedrive/documents/S105/DATASET/College_Marks_Dataset.csv")
2
3 print("--- dataset successfully loaded ---")
4 print(head(marks_df))
5
6 print("---- OUTPUT OF str() ----")
7 str(marks_df)
8
9 print("---- OUTPUT OF summary() ----")
10 summary(marks_df)
11
12 marks_df[sapply(marks_df, is.character)] <-
13   lapply(marks_df[sapply(marks_df, is.character)], as.factor)
14
15 print("---- OUTPUT OF summary() AFTER FACTOR CONVERSION ----")
16 summary(marks_df)
17
18 if("Marks" %in% colnames(marks_df)){
19   avg_marks <- mean(marks_df$Marks, na.rm = TRUE)
20   max_marks <- max(marks_df$Marks, na.rm = TRUE)
21   min_marks <- min(marks_df$Marks, na.rm = TRUE)
22
23   print(paste("Average Marks:", avg_marks))
24   print(paste("Highest Marks:", max_marks))
25   print(paste("Lowest Marks:", min_marks))
26 }
27
28 print("---- Practical completed successfully ----")
29
```

The environment pane on the right lists various datasets:

- bmw 99 obs. of 11 variables
- bmw\_clean 99 obs. of 2 variables
- clean\_exact 7 obs. of 3 variables
- combined\_data 4672 obs. of 2 variables
- combined\_spo\_ 4573 obs. of 15 variables
- df 4573 obs. of 3 variables
- df\_long 9166 obs. of 3 variables
- df\_wide 4047 obs. of 3 variables
- duplicates\_r\_ 0 obs. of 4 variables
- marks\_df 1000 obs. of 8 variables
- movies\_df 5 obs. of 2 variables
- orders\_df 7 obs. of 3 variables
- processed\_mo\_ 5 obs. of 10 variables
- spotify 4573 obs. of 15 variables
- spotify\_clean 4573 obs. of 2 variables
- spotify\_part1 500 obs. of 15 variables
- spotify\_part2 4073 obs. of 15 variables
- unique\_custo\_ 5 obs. of 3 variables

The status bar at the bottom shows the current time as 2025-12-08 11:53:58 IST.

# MVLU COLLEGE

## DATA ANALYSIS WITH SAS/ SPSS/ R

RStudio

```

> marks_df <- read.csv("C:/users/itlab/OneDrive/Documents/S105/DATASET/College_Marks_Dataset.csv")
> print("--- Dataset Successfully Loaded ---")
[1] "---- Dataset Successfully Loaded ----"
> print(head(marks_df))
  Student_ID Name Class SSC_Marks HSC_Marks College_Marks Attendance_Percentage Grade
1     S1001_0 Commerce    525      452       692           84.71     C
2     S1001_1 Commerce    494      513       551           81.99     D
3     S1002_2 Science     542      460       634           80.06     B
4     S1003_3 Science     441      483       686           79.27     D
5     S1004_4 Arts        427      544       569           91.99     A+
6     S1005_5 Science     520      539       519           88.11     B
> print("--- OUTPUT OF str() ---")
[1] "---- OUTPUT OF str() ----"
> str(marks_df)
'data.frame': 1000 obs. of 8 variables:
 $ Student_ID : chr "S1000" "S1001" "S1002" "S1003" ...
 $ Name        : chr "Student_0" "Student_1" "Student_2" "Student_3" ...
 $ Class       : chr "Commerce" "Commerce" "Science" "Science" ...
 $ SSC_Marks   : int 535 494 542 441 427 520 504 509 499 411 ...
 $ HSC_Marks   : int 452 535 460 483 544 539 573 481 474 450 ...
 $ College_Marks: int 692 551 634 686 569 519 646 504 668 636 ...
 $ Attendance_Percentage: num 84.7 82.1 92.1 79.3 92.2 ...
 $ Grade       : chr "C" "D" "B" "D" ...
> print("--- OUTPUT OF summary() ---")
[1] "---- OUTPUT OF summary() ----"
> summary(marks_df)
  Student_ID Name Class SSC_Marks HSC_Marks College_Marks Attendance_Percentage
Length:1000 Length:1000 Length:1000 Min. :400.0 Min. :450.0 Min. :500.0 Min. :60.03
Class :character Class :character Class :character 1st Qu.:437.0 1st Qu.:484.8 1st Qu.:552.0 1st Qu.:69.57
Mode :character Mode :character Mode :character Median:476.0 Median:523.5 Median:602.0 Median:80.57
                                         Mean :476.2 Mean :524.0 Mean :603.1 Mean :79.95
                                         3rd Qu.:516.0 3rd Qu.:564.0 3rd Qu.:655.2 3rd Qu.:89.95
                                         Max. :550.0 Max. :600.0 Max. :700.0 Max. :99.95
  Grade
Length:1000
Class :character
Mode :character

> marks_df[sapply(marks_df, is.character)] <-
+ lapply(marks_df[sapply(marks_df, is.character)], as.factor)
> print("--- OUTPUT OF summary() AFTER FACTOR CONVERSION ---")
[1] "---- OUTPUT OF summary() AFTER FACTOR CONVERSION ----"

```

29°C Sunny

Environment History Connections Tutorial

Data

- bmw 99 obs. of 11 variables
- bmw\_clean 99 obs. of 2 variables
- clean\_exact 7 obs. of 3 variables
- combined\_data 4672 obs. of 2 variables
- combined\_spo... 4573 obs. of 15 variables
- df 4573 obs. of 3 variables
- df\_long 9146 obs. of 3 variables
- df\_wide 4047 obs. of 3 variables
- duplicates\_r... 0 obs. of 4 variables
- marks\_df 1000 obs. of 8 variables
- movies\_df 5 obs. of 2 variables
- orders\_df 7 obs. of 3 variables
- processed\_mo... 5 obs. of 10 variables
- spotify 4573 obs. of 15 variables
- spotify\_clean 4573 obs. of 2 variables
- spotify\_part1 500 obs. of 15 variables
- spotify\_part2 4073 obs. of 15 variables
- unique\_custo... 5 obs. of 3 variables

Values

current\_time 2025-12-08 11:53:58 IST

Files Plots Packages Help Viewer Presentation

RStudio

```

> marks_df[apply(marks_df, 2, mean)]
> print("--- OUTPUT OF summary() ---")
[1] "---- OUTPUT OF summary() ----"
> summary(marks_df)
  Student_ID Name Class SSC_Marks HSC_Marks College_Marks Attendance_Percentage
Length:1000 Length:1000 Length:1000 Min. :400.0 Min. :450.0 Min. :500.0 Min. :60.03
Class :character Class :character Class :character 1st Qu.:437.0 1st Qu.:484.8 1st Qu.:552.0 1st Qu.:69.57
Mode :character Mode :character Mode :character Median:476.0 Median:523.5 Median:602.0 Median:80.57
                                         Mean :476.2 Mean :524.0 Mean :603.1 Mean :79.95
                                         3rd Qu.:516.0 3rd Qu.:564.0 3rd Qu.:655.2 3rd Qu.:89.95
                                         Max. :550.0 Max. :600.0 Max. :700.0 Max. :99.95
  Grade
Length:1000
Class :character
Mode :character

> marks_df[sapply(marks_df, is.character)] <-
+ lapply(marks_df[sapply(marks_df, is.character)], as.factor)
> print("--- OUTPUT OF summary() AFTER FACTOR CONVERSION ---")
[1] "---- OUTPUT OF summary() AFTER FACTOR CONVERSION ----"
> summary(marks_df)
  Student_ID Name Class SSC_Marks HSC_Marks College_Marks Attendance_Percentage Grade
S1000 : 1 Student_0 : 1 Arts      :260 Min. :400.0 Min. :450.0 Min. :500.0 Min. :60.03 A :193
S1001 : 1 Student_1 : 1 Commerce :244 1st Qu.:437.0 1st Qu.:484.8 1st Qu.:552.0 1st Qu.:69.57 A+ :202
S1002 : 1 Student_2 : 1 Math      :260 Median:523.5 Median:562.0 Median:602.0 Median:80.57 B :193
S1003 : 1 Student_3 : 1 Science   :236 Mean  :524.0 Mean  :603.1 Mean  :79.95 C :206
S1004 : 1 Student_101: 1 Arts      :260 3rd Qu.:516.0 3rd Qu.:564.0 3rd Qu.:655.2 3rd Qu.:89.95 D :206
S1005 : 1 Student_102: 1 Max.   :550.0 Max.   :600.0 Max.   :700.0 Max.   :99.95
(Other):994 (Other) :994
> if("Marks" %in% colnames(marks_df)){
+   avg_marks <- mean(marks_df$Marks, na.rm = TRUE)
+   max_marks <- max(marks_df$Marks, na.rm = TRUE)
+   min_marks <- min(marks_df$Marks, na.rm = TRUE)
+
+   print(paste("Average Marks:", avg_marks))
+   print(paste("Highest Marks:", max_marks))
+   print(paste("Lowest Marks:", min_marks))
+ }
> print("--- Practical Completed successfully ---")
[1] "---- Practical Completed Successfully ----"
> |

```

29°C Sunny

Environment History Connections Tutorial

Data

- bmw 99 obs. of 11 variables
- bmw\_clean 99 obs. of 2 variables
- clean\_exact 7 obs. of 3 variables
- combined\_data 4672 obs. of 2 variables
- combined\_spo... 4573 obs. of 15 variables
- df 4573 obs. of 3 variables
- df\_long 9146 obs. of 3 variables
- df\_wide 4047 obs. of 3 variables
- duplicates\_r... 0 obs. of 4 variables
- marks\_df 1000 obs. of 8 variables
- movies\_df 5 obs. of 2 variables
- orders\_df 7 obs. of 3 variables
- processed\_mo... 5 obs. of 10 variables
- spotify 4573 obs. of 15 variables
- spotify\_clean 4573 obs. of 2 variables
- spotify\_part1 500 obs. of 15 variables
- spotify\_part2 4073 obs. of 15 variables
- unique\_custo... 5 obs. of 3 variables

Values

current\_time 2025-12-08 11:53:58 IST

Files Plots Packages Help Viewer Presentation