

## PRACTICAL NO 8

**AIM:- Applying basic data cleaning functions: handling missing values using `na.omit()`/`replace_na()` in R. import dataset.**

## **OUTPUT:-**

The screenshot shows the RStudio interface with the following details:

- Title Bar:** DATA SET - RStudio
- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar:** Go to file/function, Addins
- Source Editor:** Contains R code for dataset cleaning, including handling missing values and auto-cleaning.
- Environment Tab:** Shows the Global Environment with variables like id\_cols, input\_file, nm, numeric\_patterns, out\_file, p, packages, pat, pkg, and possible\_paths.
- Files Tab:** Lists files in the current directory, including PRACTICAL.NW.R, Processed\_Global\_Mobile\_Prices.csv, SALARY.xlsx, synthetic\_freelance.jobs.csv, WhatsApp Image 2025-11-30 at 13:41:44\_0156d80c0e9433 KB, WhatsApp Image 2025-11-30 at 13:41:46\_6541516e.je 93.5 KB, WhatsApp Image 2025-11-30 at 13:41:47\_401b4e63.jc 146.6 KB, PRACTICAL NO 6.R, amazon\_merged\_jan\_feb\_big.csv, amazon\_final\_list\_big.csv, merged\_output.csv, final\_output.csv, nyc\_flights.csv, PRACTICAL NO 7.R, Breast\_Cancer.csv, PRACTICAL NO 8.R, and Breast\_Cancer\_cleaned.csv.

# Sheth L.U.J. College of Arts And Sir M.V. College of Science and Commerce

## Data Analysis with SAS / SPSS / R

**R DATA SET - RStudio**

File Edit Code View Plots Session Build Debug Profile Tools Help

Console Terminal Background Jobs

```
R 4.5.2 · C:/Users/as993/DATA SET/ 
  * specify the certain types or set strcuretypes = TRUE to ignore this message.
> cat("Loaded:", input_file, "\nrows:", nrow(df), "cols:", ncol(df), "\n\n")
Loaded: Breast_Cancer.csv
Rows: 200 cols: 8

> # ---- 2. Quick inspect ----
> cat("Column names:\n"); print(names(df)); cat("\nMissing per column (before):\n")
Column names:
[1] "Patient_ID"      "Age"           "Tumor_Size_mm"   "Tumor_Location" "Mean_Radius"
[6] "Mean_Texture"    "Mean_Smoothness" "Diagnosis"
[7] "Mean_Convexity"   "Mean_Symmetry"  "Mean_Vessel_Strength" "Class"

Missing per column (before):
> print(cols(is.na(df))); cat("\n")
  Patient_ID      Age Tumor_Size_mm Tumor_Location Mean_Radius Mean_Texture
  0             0          0            0            0            0            0
Mean_Smoothness Diagnosis
  0             0

> # ---- 3. Helpers ----
> get_mode <- function(x) {
+   x_no <- na.omit(x)
+   if (length(x_no) == 0) return(NA_character_)
+   ux <- unique(x_no)
+   ux[which.max(tabulate(match(x_no, ux)))]
+ }
+ is.numeric_like <- function(x) {
+   if (!is.character(x)) return(FALSE)
+   s <- stringr::str_trim(x)
+   # allow decimal and integer numbers, drop empty/NA
+   s <- s[s != "" & !is.na(s)]
+   if (length(s) == 0) return(FALSE)
+   suppressWarnings(all(is.na(as.numeric(s))))
+ }
+ convert_numeric_like <- function(dat) {
+   for (nm in names(dat)) {
+     if (is.character(dat[[nm]]) & is.numeric_like(dat[[nm]])) {
+       dat[[nm]] <- as.numeric(stringr::str_trim(dat[[nm]]))
+       message("Converted to numeric-like: ", nm)
+     }
+   }
+ }

  Patient_ID      Age Tumor_Size_mm Tumor_Location Mean_Radius Mean_Texture
  0             0          0            0            0            0            0
Mean_Smoothness Diagnosis
  0             0

> # ---- 4. Preprocess conversions ----
> df <- convert_numeric_like(df)
>
> # If there's an 'id' like column, standardize name to Patient_ID (non-destructive)
> id_cols <- names(df)[tolower(names(df)) %in% c("id", "patient_id", "patientid", "pid")]
> if (length(id_cols) > 0) {
+   # rename first matching to Patient_ID
+   names(df)[names(df) == id_cols[1]] <- "Patient_ID"
+   # Renamed column "", id_cols[1], "" -> 'Patient_ID'
+ } else {
+   # if no id, create Patient_ID sequentially
+   df$Patient_ID <- paste0("Bc", sprintf("%04d", seq_len(nrow(df))))
+   message("No ID column found -> created Patient_ID")
+ }

Renamed column 'Patient_ID' -> 'Patient_ID'

> # ---- 5. Dataset-aware custom rules (only apply if those columns exist) ----
> # - Diagnosis: keep as-is; if missing, mark as "Unknown"
> if ("Diagnosis" %in% names(df)) {
+   df$Diagnosis <- as.character(df$Diagnosis)
+   df$Diagnosis[is.na(df$Diagnosis)] <- "Unknown"
+   message("Diagnosis missing values -> 'Unknown'")
+ }

Diagnosis missing values -> 'Unknown'

> # - Common numeric columns may have columns with names containing 'radius', 'texture', 'smooth', 'perimeter', 'area'
> numeric_patterns <- c("radius", "texture", "smooth", "perimeter", "area", "compactness", "concavity", "symmetry", "fractal")
> for (pat in numeric_patterns) {
+   hits <- grep(pat, names(df), ignore.case = TRUE, value = TRUE)
+   if (length(hits) > 0) {
+     for (h in hits) {
+       if (!is.numeric(df[[h]])) {
+         df[[h]] <- suppressWarnings(as.numeric(df[[h]]))
+       }
+       message("Checked numeric-like column: ", h)
+     }
+   }
+ }
```

Environment History Connections Tutorial

Global Environment

- id\_cols "Patient\_ID"
- input\_file "Breast\_Cancer.csv"
- nm "Diagnosis"
- numeric\_patterns chr [1:9] "radius" "texture" "smooth" "perimeter" "area" "compactness" "concavity" "symmetry" "fractal"
- out\_file "Breast\_Cancer\_cleaned.csv"
- p "Breast\_Cancer.csv"
- packages "dplyr" "tidyverse" "readr" "stringr"
- pat "fractal"
- pkg "stringr"
- possible\_paths chr [1:3] "/mnt/data/Breast\_Cancer.csv" "Breast\_Cancer\_cleaned.csv"

Files Plots Packages Help Viewer Presentation

New Folder New File Delete Rename More

PRACTICAL NO 3.R SALARY.xlsx synthetic\_freelance.jobs.csv WhatsApp Image 2025-11-30 at 13.41.44\_6541518e0.jpe WhatsApp Image 2025-11-30 at 13.41.47\_401b4e63.jpe PRACTICAL NO 6.R amazon\_merged\_jan\_feb\_big.csv amazon\_final\_list\_big.csv merged\_output.csv final\_output.csv nyc.flights.csv PRACTICAL NO 7.R Breast.Cancer.csv PRACTICAL NO 8.R Breast\_Cancer\_cleaned.csv

Snipping Tool

Screenshot copied to clipboard Automatically saved to screenshots folder.

Mark-up and share

**R DATA SET - RStudio**

File Edit Code View Plots Session Build Debug Profile Tools Help

Console Terminal Background Jobs

```
R 4.5.2 · C:/Users/as993/DATA SET/ 
  * specify the certain types or set strcuretypes = TRUE to ignore this message.
>
> # ---- 4. Preprocess conversions ----
> df <- convert_numeric_like(df)
>
> # If there's an 'id' like column, standardize name to Patient_ID (non-destructive)
> id_cols <- names(df)[tolower(names(df)) %in% c("id", "patient_id", "patientid", "pid")]
> if (length(id_cols) > 0) {
+   # rename first matching to Patient_ID
+   names(df)[names(df) == id_cols[1]] <- "Patient_ID"
+   # Renamed column "", id_cols[1], "" -> 'Patient_ID'
+ } else {
+   # if no id, create Patient_ID sequentially
+   df$Patient_ID <- paste0("Bc", sprintf("%04d", seq_len(nrow(df))))
+   message("No ID column found -> created Patient_ID")
+ }

Renamed column 'Patient_ID' -> 'Patient_ID'

> # ---- 5. Dataset-aware custom rules (only apply if those columns exist) ----
> # - Diagnosis: keep as-is; if missing, mark as "Unknown"
> if ("Diagnosis" %in% names(df)) {
+   df$Diagnosis <- as.character(df$Diagnosis)
+   df$Diagnosis[is.na(df$Diagnosis)] <- "Unknown"
+   message("Diagnosis missing values -> 'Unknown'")
+ }

Diagnosis missing values -> 'Unknown'

> # - Common numeric columns may have columns with names containing 'radius', 'texture', 'smooth', 'perimeter', 'area'
> numeric_patterns <- c("radius", "texture", "smooth", "perimeter", "area", "compactness", "concavity", "symmetry", "fractal")
> for (pat in numeric_patterns) {
+   hits <- grep(pat, names(df), ignore.case = TRUE, value = TRUE)
+   if (length(hits) > 0) {
+     for (h in hits) {
+       if (!is.numeric(df[[h]])) {
+         df[[h]] <- suppressWarnings(as.numeric(df[[h]]))
+       }
+       message("Checked numeric-like column: ", h)
+     }
+   }
+ }
```

Environment History Connections Tutorial

Global Environment

- id\_cols "Patient\_ID"
- input\_file "Breast\_Cancer.csv"
- nm "Diagnosis"
- numeric\_patterns chr [1:9] "radius" "texture" "smooth" "perimeter" "area" "compactness" "concavity" "symmetry" "fractal"
- out\_file "Breast\_Cancer\_cleaned.csv"
- p "Breast\_Cancer.csv"
- packages "dplyr" "tidyverse" "readr" "stringr"
- pat "fractal"
- pkg "stringr"
- possible\_paths chr [1:3] "/mnt/data/Breast\_Cancer.csv" "Breast\_Cancer\_cleaned.csv"

Files Plots Packages Help Viewer Presentation

New Folder New File Delete Rename More

PRACTICAL NO 3.R SALARY.xlsx synthetic\_freelance.jobs.csv WhatsApp Image 2025-11-30 at 13.41.44\_6541518e0.jpe WhatsApp Image 2025-11-30 at 13.41.47\_401b4e63.jpe PRACTICAL NO 6.R amazon\_merged\_jan\_feb\_big.csv amazon\_final\_list\_big.csv merged\_output.csv final\_output.csv nyc.flights.csv PRACTICAL NO 7.R Breast.Cancer.csv PRACTICAL NO 8.R Breast\_Cancer\_cleaned.csv

# Sheth L.U.J. College of Arts And Sir M.V. College of Science and Commerce

## Data Analysis with SAS / SPSS / R

**R DATA SET - RStudio**

```

File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

Source
Console Terminal Background Jobs
R + R 4.5.2 - C:/Users/as993/DATA SET/...
y ~ 'fractai'
> for (pat in numeric_patterns) {
+   hits <- grep(pat, names(df), ignore.case = TRUE, value = TRUE)
+   if (length(hits) > 0) {
+     for (h in hits) {
+       if (is.numeric(df[[h]])) {
+         df[[h]] <- suppressWarnings(as.numeric(df[[h]]))
+       }
+       message("Checked numeric-like column:", h)
+     }
+   }
+ }

Checked numeric-like column:Mean_Radius
Checked numeric-like column:Mean_Texture
Checked numeric-like column:Mean_Smoothness

> # ---- 6. Imputation strategy ----
> Cleaned <- df # copy
>

> # If Price-like logic not relevant here; we use general rules:
> for (nm in names(cleaned)) {
+   col <- cleaned[[nm]]
+   if (any(is.na(col))) {
+     if (is.numeric(col)) {
+       med <- median(col, na.rm = TRUE)
+       if (is.na(med)) med <- 0
+       cleaned[[nm]]<-na(cleaned[[nm]]) <- med
+       message(sprintf("Numeric imputed (median) for %s -> %s", nm, format(med, digits=6)))
+     } else if (is.logical(col)) {
+       cleaned[[nm]]<-na(cleaned[[nm]]) <- FALSE
+       message(sprintf("Logical imputed (FALSE) for %s", nm))
+     } else { # character/factor
+       modev <- charTable(modev)
+       if (is.na(modev)) modev <- "Unknown"
+       cleaned[[nm]]<-na(cleaned[[nm]]) <- modev
+       message(sprintf("Categorical imputed (mode/'Unknown') for %s -> %s", nm, as.character(modev)))
+     }
+   }
+ }

#----- 7. Data Cleaning ----#
#----- 8. Data Transformation ----#
#----- 9. Provide quick counts ----#
#----- 10. Agar aur customization chahiye (e.g., specific columns ka different rule, or scaling, or encoding)
#----- 11. Batao. \n"

```

Environment History Connections Tutorial

id\_cols "Patient\_ID"
input\_file "Breast\_Cancer.csv"
nm "Diagnosis"
numeric\_patterns chr [1:9] "radius" "texture" "smooth" "perimeter" "area" "symmetry" "compactness" "concavity" "concave points"
out\_file "Breast\_Cancer\_cleaned.csv"
p "Breast\_Cancer.csv"
packages "dplyr" "tidyverse" "readr" "stringr"
pat "fractai"
pkg "stringr"
possible\_paths chr [1:3] "/mnt/data/Breast\_Cancer.csv" "Breast\_Cancer\_cleaned.csv"

Files Plots Packages Help Viewer Presentation

13:03  
ENG US 01-12-2025

**R DATA SET - RStudio**

```

File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

Source
Console Terminal Background Jobs
R + R 4.5.2 - C:/Users/as993/DATA SET/...
Missing per column (after):
  Patient_ID    Age    Tumor_Size_mm Tumor_Location    Mean_Radius    Mean_Texture
  0             0           0              0            0             0
Mean_Smoothness Diagnosis
  0             0

> num_summary <- cleaned %>% select(where(is.numeric))
> if (ncol(num_summary) > 0) {
+   cat("Numeric summary (selected):\n"); print(summary(num_summary)); cat("\n")
+ }
Numeric summary (selected):
  Age    Tumor_Size_mm Mean_Radius  Mean_Texture Mean_Smoothness
  Min. :30.00  Min. :10.00  Min. :10.01  Min. :15.10  Min. :0.07000
  1st Qu.:41.75  1st Qu.:21.00  1st Qu.:13.32  1st Qu.:19.88  1st Qu.:0.08975
  Median :54.50  Median :33.00  Median :17.21  Median :24.39  Median :0.10750
  Mean   :54.84  Mean   :33.95  Mean   :17.36  Mean   :24.69  Mean   :0.10981
  3rd Qu.:68.00  3rd Qu.:47.00  3rd Qu.:21.34  3rd Qu.:29.53  3rd Qu.:0.13025
  Max.  :80.00  Max.  :60.00  Max.  :24.87  Max.  :34.90  Max.  :0.15000

> 
> out_file <- "Breast_Cancer_cleaned.csv"
> readr::write_csv(cleaned, out_file)
> cat("Cleaned file written to:", out_file, "\n")
Cleaned file written to: Breast_Cancer_cleaned.csv
>
> # ---- 9. Provide quick counts ----#
> cat(sprintf("Rows: %d | Columns: %d\n", nrow(cleaned), ncol(cleaned)))
Rows: 200 | Columns: 8
> cat("Diagnosis value counts:\n"); if ("Diagnosis" %in% names(cleaned)) print(table(cleaned$Diagnosis)) else cat("No Diagnosis column.\n")
Diagnosis value counts:
  Benign Malignant
  101      99
> cat("\nDone. Agar aur customization chahiye (e.g., specific columns ka different rule, or scaling, or encoding)\n, batao.\n")
Done. Agar aur customization chahiye (e.g., specific columns ka different rule, or scaling, or encoding), batao.

#----- 10. Agar aur customization chahiye (e.g., specific columns ka different rule, or scaling, or encoding)
#----- 11. Batao. \n"

```

Environment History Connections Tutorial

id\_cols "Patient\_ID"
input\_file "Breast\_Cancer.csv"
nm "Diagnosis"
numeric\_patterns chr [1:9] "radius" "texture" "smooth" "perimeter" "area" "symmetry" "compactness" "concavity" "concave points"
out\_file "Breast\_Cancer\_cleaned.csv"
p "Breast\_Cancer.csv"
packages "dplyr" "tidyverse" "readr" "stringr"
pat "fractai"
pkg "stringr"
possible\_paths chr [1:3] "/mnt/data/Breast\_Cancer.csv" "Breast\_Cancer\_cleaned.csv"

Files Plots Packages Help Viewer Presentation

13:04  
ENG US 01-12-2025