

EXPERIMENT-13

MIN,MAX,MEAN,MINMAX:

AIM:

To write the program for the minimum,maximum,mean and minmax using r-TOOL

PROGRAM:

```
library(readxl) # Load the readxl package

file_path <- "C:/Users/harik/Downloads/Data Excel sheet for study on DM.xlsx"

diabetest1 <- read_excel(file_path)

colnames(diabetest1) <- trimws(colnames(diabetest1))


# Function to compute statistics and Min-Max normalization
compute_stats <- function(data, column_name) {
  if (column_name %in% colnames(data)) {
    # Convert column to numeric
    data[[column_name]] <- as.numeric(as.character(data[[column_name]]))

    # Remove NA values
    values <- na.omit(data[[column_name]])

    # Compute statistics
    min_value <- min(values, na.rm = TRUE)
    max_value <- max(values, na.rm = TRUE)
    mean_value <- mean(values, na.rm = TRUE)

    # Compute Min-Max normalization
    min_max_normalized <- (values - min_value) / (max_value - min_value)

    # Print results
    print(paste("Statistics for column:", column_name))
    print(paste("Minimum:", min_value))
    print(paste("Maximum:", max_value))
```

```

    print(paste("Mean:", mean_value))

    print("Min-Max Normalization Values:")

    print(min_max_normalized)

} else {

    print(paste("Column", column_name, "not found in the dataset."))

}

}

```

Example: Compute statistics for "AGE" column

```
compute_stats(diabetest1, "AGE")
```

Example: Compute statistics for another column (e.g., "Glucose")

```
compute_stats(diabetest1, "Glucose")
```

Output:

```

> diabetest1 <- read_excel(file_path)

> # Clean column names (remove extra spaces)
> colnames(diabetest1) <- trimws(colnames(diabetest1))

> # Function to compute statistics and Min-Max normalization
> compute_stats <- function(data, column_name) {
+   if (column_name %in% colnames(data)) .... [TRUNCATED]

> # Example: Compute statistics for "AGE" column
> compute_stats(diabetest1, "AGE")
[1] "Statistics for column: AGE"
[1] "Minimum: 35"
[1] "Maximum: 69"
[1] "Mean: 53.4428571428571"
[1] "Min-Max Normalization Values:"
[1] 0.5588235 0.9117647 0.5000000 0.4411765 0.4117647 0.5000000 0.7941176 1.0000000 0.3823529 0.2352941
[11] 0.2647059 0.7058824 0.5294118 0.7352941 0.4411765 0.8235294 0.3529412 0.6470588 0.8823529 0.8529412
[21] 0.8529412 0.6764706 0.2941176 0.2647059 0.2941176 0.3235294 0.2647059 0.6176471 0.8235294 0.2941176
[31] 0.2352941 0.8235294 0.1470588 0.9117647 0.1176471 0.7941176 0.0000000 0.4117647 0.4705882 0.3235294
[41] 0.4411765 0.9117647 0.1176471 0.7352941 0.4705882 0.6764706 0.6764706 0.5882353 0.6764706 0.7941176
[51] 0.4117647 0.8235294 0.9117647 0.4411765 0.7647059 0.1470588 0.3823529 0.4705882 0.4705882 0.3529412
[61] 0.5294118 0.8823529 0.9117647 0.1470588 0.3529412 0.6764706 0.3235294 0.2941176 0.7647059 0.9117647

> # Example: Compute statistics for another column (e.g., "Glucose")
> compute_stats(diabetest1, "Glucose")
[1] "Column Glucose not found in the dataset."
> |

```
