AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

INTRODUCTION TO DATA SCIENCE

SPRING 2024-25

Supervise By

TOHEDUL ISLAM

Assignment: Mid Term Project Report

Section: F

Submitted By:

NAME	ID
MD. SAMIN YEASAR	22-47139-1

CONTENTS NAME — PAGE NO

- **Description of Dataset** 3

1. **Libraries** 3
2. **Loading the Dataset and Initial Exploratio** 3-4
3. **Handling Missing Values and Remove Invalid Data** 4-8
4. **Visualizing Missing Values** 8-9
5. **Outlier Detection and Handling** 9-10
6. **Data Type Conversion** 10
7. **Normalization** 11
8. **Duplicate Row Removal** 12
9. **Data Filtering** 12
10. **Handle Imbalance Dataset** 13-14
11. **Split dataset Into Test and Training** 14-15
12. **Central tendency (mean/median/mode) comparison across gender groups** 15
13. **Age Distribution by Hypertension Status** 16
14. **Spread Metrics by Gender** 16
15. **Statistical Analysis: BMI by Diabetes Status** 17

Description of Dataset:

The Diabetes Prediction Dataset is a collection of medical and demographic data used to predict diabetes in patients. The dataset contains several health parameters and patient characteristics that are commonly associated with diabetes risk. These include age, gender, BMI (Body Mass Index), hypertension status, heart disease history, smoking history, HbA1c level, blood glucose level, and the target variable indicating diabetes status. This dataset is valuable for developing predictive models to identify individuals at risk of diabetes, which can aid in early intervention and prevention strategies.

1. Libraries:

```
library(tidyverse)
library(ggplot2)
library(mice)
library(caret)
library(dplyr)
library(tidyr)
library(zoo)
```

tidyverse: Data manipulation and visualization tools (includes dplyr, ggplot2, tidyr).
ggplot2: Creates advanced plots (e.g., histograms, scatterplots).
mice: Handles missing data (imputation).
caret: Machine learning tools (train/test split, modeling).
dplyr: Data wrangling (filter, mutate, summarize).
tidyr: Cleans data (pivot, separate, drop NAs).
zoo: Time-series data handling (rollmeans, date ops).

2. Loading the Dataset and Initial Exploration

Code:

```
data <- read.csv("D:/Dataset(Updated)_MIdterm_sectoin(F).csv", na.strings = c("", "N

str(data)
```

Output:

```
> str(data)
'data.frame':   122 obs. of  9 variables:
 $ gender             : chr  "Female" "Female" "Male" "Female" ...
 $ age                : int  80 54 28 NA 76 20 79 42 32 53 ...
 $ hypertension       : int  0 0 0 0 1 0 0 0 0 0 ...
 $ heart_disease      : int  1 0 0 0 1 0 0 0 0 0 ...
 $ smoking_history    : chr  "never" "No Info" "never" "current" ...
 $ bmi                : num  25.2 27.3 -27.3 23.4 20.1 ...
 $ HbA1c_level        : num  6.6 6.6 5.7 5 4.8 6.6 5.7 4.8 5 6.1 ...
 $ blood_glucose_level: chr  "140" "80" "158" "155" ...
 $ diabetes           : int  0 0 0 0 0 0 0 0 0 0 ...
>
```

The initial exploration reveals the dataset has 122 observations with 9 variables. Gender and smoking_history are character variables, while age and bmi are numeric. Hypertension, heart_disease, and diabetes are stored as integers but represent binary categorical data.

3. Handling Missing Values and Remove Invalid Data

i) Forward Fill (Top to Bottom)

Code:

```
colSums(is.na(data))
top_bottom_data <- data %>% fill(gender, age, hypertension, smoking_history, bmi, .direction = 'down')
colSums(is.na(top_bottom_data))
```

Output:

```
> colSums(is.na(data))
             gender                 age        hypertension       heart_disease
                  2                   4                   2                   0
    smoking_history                 bmi         HbA1c_level blood_glucose_level
                  5                   2                   0                   0
           diabetes
                  0
> top_bottom_data <- data %>% fill(gender, age, hypertension, smoking_history, bmi, .direc
tion = 'down')
> colSums(is.na(top_bottom_data))
             gender                 age        hypertension       heart_disease
                  0                   0                   0                   0
    smoking_history                 bmi         HbA1c_level blood_glucose_level
                  0                   0                   0                   0
           diabetes
                  0
> |
```

This approach fills missing values by propagating the last known non-NA value downward (from top to bottom). It is useful when data follows a logical sequence (e.g., time-series data) where missing values can reasonably be assumed to carry forward the previous entry.

ii) Backward Fill (Bottom to Top)

Code:

```
colSums(is.na(data))
bottom_top_data <- data %>% fill(gender, age, hypertension, smoking_history, bmi, .direction = 'down')
colSums(is.na(bottom_top_data))
```

Output:

```
> colSums(is.na(data))
            gender                age       hypertension      heart_disease
                 2                  4                  2                  0
   smoking_history                bmi        HbA1c_level blood_glucose_level
                 5                  2                  0                  0
          diabetes
                 0
> bottom_top_data <- data %>% fill(gender, age, hypertension, smoking_history, bmi, .direc
tion = 'down')
> colSums(is.na(bottom_top_data))
            gender                age       hypertension      heart_disease
                 0                  0                  0                  0
   smoking_history                bmi        HbA1c_level blood_glucose_level
                 0                  0                  0                  0
          diabetes
                 0
> |
```

The backward fill method replaces missing values by taking the next available non-NA value and filling upward (from bottom to top). This is helpful when later entries are more reliable or when forward fill would be inappropriate. Like forward filling, it assumes data continuity, but it may not be ideal if missingness occurs in large chunks, leading to repeated values from later records.

iii) Discard Data

Code:

```
colSums(is.na(data))
data_cleaned <- na.omit(data)
colSums(is.na(data_cleaned))
```

Output:

```
> colSums(is.na(data))
            gender                age       hypertension      heart_disease
                 2                  4                  2                  0
   smoking_history                bmi        HbA1c_level blood_glucose_level
                 5                  2                  0                  0
          diabetes
                 0
> data_cleaned <- na.omit(data)
> colSums(is.na(data_cleaned))
            gender                age       hypertension      heart_disease
                 0                  0                  0                  0
   smoking_history                bmi        HbA1c_level blood_glucose_level
                 0                  0                  0                  0
          diabetes
                 0
> |
```

This method removes any rows containing missing values, retaining only complete observations. It is the simplest approach and avoids imputation bias, but it can significantly reduce dataset size if

missingness is widespread. This approach works best when missing data is minimal and random, ensuring that deletion does not distort the dataset's representativeness.

iv) Frequent/Average Value Imputation
Code:

```
newdata <- data
colSums(is.na(newdata))
find_mode <- function(x) {
  tbl <- table(x[!is.na(x)])
  names(tbl)[which.max(tbl)]
}

mode_gender <- find_mode(newdata$gender)
newdata$gender[is.na(newdata$gender)] <- mode_gender

newdata$gender[newdata$gender == "Femalee"] <- "Female"
newdata$gender[newdata$gender == "Malee"] <- "Male"

mode_smoking <- find_mode(newdata$smoking_history)
newdata$smoking_history[is.na(newdata$smoking_history)] <- mode_smoking

mode_hypertension <- find_mode(newdata$hypertension)
newdata$hypertension[is.na(newdata$hypertension)] <- mode_hypertension

mode_heart <- find_mode(newdata$heart_disease)
newdata$heart_disease[is.na(newdata$heart_disease)] <- mode_heart

newdata$age <- as.numeric(as.character(newdata$age))
newdata$age[newdata$age < 0 | newdata$age > 120] <- NA
mean_age <- mean(newdata$age, na.rm = TRUE)
newdata$age[is.na(newdata$age)] <- round(mean_age)

newdata$bmi <- as.numeric(as.character(newdata$bmi))
```

Output:

```
> mode_smoking <- find_mode(newdata$smoking_history)
> newdata$smoking_history[is.na(newdata$smoking_history)] <- mode_smoking
>
> mode_hypertension <- find_mode(newdata$hypertension)
> newdata$hypertension[is.na(newdata$hypertension)] <- mode_hypertension
>
> mode_heart <- find_mode(newdata$heart_disease)
> newdata$heart_disease[is.na(newdata$heart_disease)] <- mode_heart
> colSums(is.na(newdata))
              gender                  age         hypertension        heart_disease
                   0                    0                    0                    0
     smoking_history                  bmi          HbA1c_level blood_glucose_level
                   0                    0                    0                    0
            diabetes
                   0
```

Frequent value imputation replaces missing values in a dataset with the most common value (mode) of that variable. This method is particularly useful for categorical data (e.g., gender, smoking history) where calculating a mean or median would not make sense.

v) Handling Invalid Values (Its in another Question but Answered in here for the relevant question)

Code:

```
newdata$age <- as.numeric(as.character(newdata$age))
newdata$age[newdata$age < 0 | newdata$age > 120] <- NA
mean_age <- mean(newdata$age, na.rm = TRUE)
newdata$age[is.na(newdata$age)] <- round(mean_age)

newdata$bmi <- as.numeric(as.character(newdata$bmi))
newdata$bmi[newdata$bmi < 0] <- NA
mean_bmi <- mean(newdata$bmi, na.rm = TRUE)
newdata$bmi[is.na(newdata$bmi)] <- round(mean_bmi, 2)

mean_hba1c <- mean(newdata$HbA1c_level, na.rm = TRUE)
newdata$HbA1c_level[is.na(newdata$HbA1c_level)] <- round(mean_hba1c, 1)

newdata$blood_glucose_level <- as.numeric(gsub("[^0-9.]", "", as.character(newdata$blood_glucose_level)))
mean_bg <- mean(newdata$blood_glucose_level, na.rm = TRUE)
newdata$blood_glucose_level[is.na(newdata$blood_glucose_level)] <- round(mean_bg)


colSums(is.na(newdata))
```

Output:

```
                   0
> newdata
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level
1   Female  80            0             1           never 25.19         6.6
2   Female  54            0             0         No Info 27.32         6.6
3     Male  28            0             0           never 27.89         5.7
4   Female  50            0             0         current 23.45         5.0
5     Male  76            1             1         current 20.14         4.8
6   Female  20            0             0           never 27.32         6.6
7   Female  79            0             0         No Info 23.86         5.7
8     Male  42            0             0           never 33.64         4.8
9   Female  32            0             0           never 27.32         5.0
10  Female  53            0             0           never 27.32         6.1
11  Female  54            0             0          former 27.89         6.0
12  Female  78            0             0          former 36.05         5.0
13  Female  67            0             0           never 25.69         5.8
14  Female  76            0             0         No Info 27.32         5.0
15  Female  78            0             0         No Info 27.32         6.6
16    Male  15            0             0           never 27.89         6.1
17  Female  42            0             0           never 24.48         5.7
18  Female  42            0             0           never 27.32         5.7
19    Male  50            0             0            ever 25.72         3.5
20    Male  40            0             0         current 36.38         6.0
21    Male   5            0             0         No Info 18.80         6.2
22  Female  69            0             0           never 21.24         4.8
23  Female  72            0             1          former 27.94         6.5
24  Female   4            0             0         No Info 13.99         4.0
25    Male  30            0             0           never 33.76         6.1
26    Male  40            0             0          former 27.85         5.8
27    Male  50            0             0           never 26.47         4.0
28    Male  43            0             0           never 26.08         6.1
```

Invalid data handling involves identifying and correcting illogical or inconsistent values. For categorical variables gender, misspellings (e.g., "Femalee") are fixed by replacing them with valid categories. Numerical variables (age, bmi, blood_glucose_level) are sanitized by converting them to numeric type, removing non-numeric characters, and filtering impossible values (e.g., negative BMI or ages outside 0-120 years). Invalid entries are set to NA and later imputed with mean/mode values,

ensuring the dataset remains consistent and analysis-ready. This step eliminates data errors while preserving meaningful observations.

4. Visualizing Missing Values

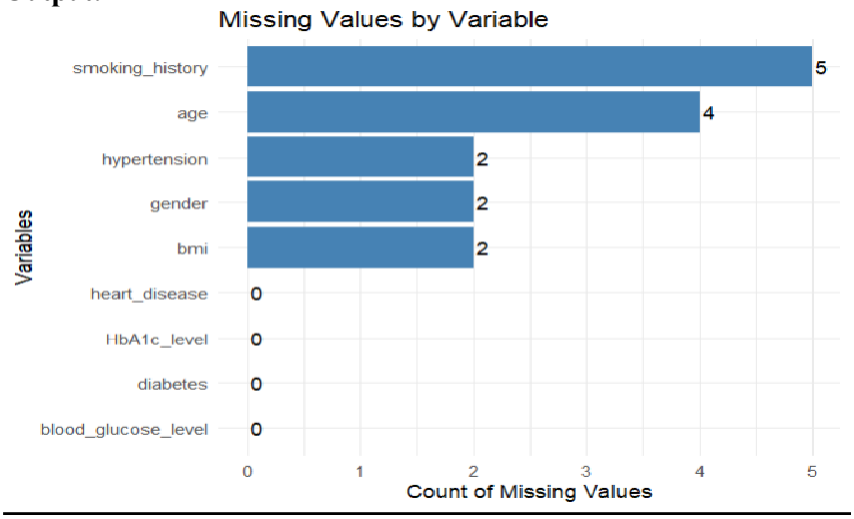
Code:

```
missing_plot <- function(data) {
  missing_data <- data %>% is.na() %>% colSums()
  missing_df <- data.frame(
    Variable = names(missing_data),
    Count = missing_data
  )


  ggplot(missing_df, aes(x = reorder(Variable, Count), y = Count)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    geom_text(aes(label = Count), hjust = -0.2) +
    labs(title = "Missing Values by Variable",
         x = "Variables",
         y = "Count of Missing Values") +
    coord_flip() +
    theme_minimal()
}

md.pattern(data, plot = TRUE)
missing_plot(data)
```

Output:



Missing values are analyzed through two visual methods. A custom bar plot quantifies NAs per variable using ggplot2, while mice::md.pattern() reveals patterns in missing data co-occurrence.

These visualizations help identify columns needing focused cleaning and validate the effectiveness of imputation methods.

5. Outlier Detection and Handling:

Code:

```
detect_outlier <- function(dataframe, columns) {
for (col in columns) {
      if (is.numeric(dataframe[[col]])){
            Quantile1 <- quantile(dataframe[[col]], probs = 0.25)
            Quantile3 <- quantile(dataframe[[col]], probs = 0.75)
            IQR <- Quantile3 - Quantile1
            outlier_flags <- dataframe[[col]] > Quantile3 + (IQR * 1.5) | dataframe[[col]] < Quantile1 - (IQR * 1.5)
            outliers <- dataframe[[col]][outlier_flags]
            if (length(outliers) > 8) {
                  cat ("Outliers detected in column", col, "=\n")
                  print(outliers)
            } else {
              cat("No outliers detected in column", col, "\n")
              }
      } else {
          cat("Column", col, "is not numeric, skipped\n")
      }
    }
}
remove_outlier <- function (dataframe, columns){
  for(col in columns){
    if(is.numeric(dataframe[[col]])) {
      quantile1 <- quantile(dataframe[[col]], probs = 0.25)
      quantile3 <- quantile(dataframe[[col]], probs = 0.75)
      IQR <- quantile3 - quantile1
      dataframe <- dataframe[!(dataframe[[col]] > quantile3 + (IQR * 1.5) | dataframe[[col]] < quantile3 - (IQR * 1.5)),]
    }
  }
  return(dataframe)
}
detect_outlier(newdata, names(newdata))
without_outlierdata <- remove_outlier(newdata, names(newdata))
detect_outlier(without_outlierdata, names(without_outlierdata))
```

Output:

```
> detect_outlier(newdata, names(newdata))
Column gender is not numeric, skipped
No outliers detected in column age
Column hypertension is not numeric, skipped
Column heart_disease is not numeric, skipped
Column smoking_history is not numeric, skipped
Outliers detected in column bmi =
 [1] 36.05 36.38 13.99 15.10 18.03 15.94 15.80 17.98 37.16 63.48 36.49 39.36 36.18 50.30 40.31 36.12 37.24 43.41 49.27 39.00
Outliers detected in column HbA1c_level =
 [1] 3.5 4.0 4.0 4.0 4.0 4.0 3.5 3.5 4.0 4.0 3.5 4.0 9.0 9.0 8.8 8.2 9.0 9.0 8.2 9.0 8.2 8.2 8.2 9.0 8.8 8.2 8.8 9.0 8.8 9.0
[31] 8.8
Outliers detected in column blood_glucose_level =
 [1]  80  80  80 260 220 300 280 280 280 300 280 220 260 260 220 300
No outliers detected in column diabetes
> without_outlierdata <- remove_outlier(newdata, names(newdata))
> detect_outlier(without_outlierdata, names(without_outlierdata))
Column gender is not numeric, skipped
No outliers detected in column age
Column hypertension is not numeric, skipped
Column heart_disease is not numeric, skipped
Column smoking_history is not numeric, skipped
No outliers detected in column bmi
No outliers detected in column HbA1c_level
No outliers detected in column blood_glucose_level
No outliers detected in column diabetes
>
```

Numeric columns are scanned for outliers using the 1.5*IQR rule. The detection function flags extreme values beyond the upper/lower quartile boundaries. Outliers are then removed by filtering

these threshold violations, creating a cleaner dataset for analysis while preserving the central data distribution.

6. Data Type Conversion:

Code:

```
tempdata <- newdata
tempdata$gender<-factor(tempdata$gender,levels=c("Male","Female"),labels=c(1,2))
tempdata$heart_disease<-factor(tempdata$heart_disease,levels=c(1,0),labels=c("yes","no"))
tempdata
```

Output:

```
> tempdata <- newdata
> tempdata$gender<-factor(tempdata$gender,levels=c("Male","Female"),labels=c(1,2))
> tempdata$heart_disease<-factor(tempdata$heart_disease,levels=c(1,0),labels=c("yes","no"))
> tempdata
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1        2  80            0           yes           never 25.19         6.6                 140        0
2        2  54            0            no         No Info 27.32         6.6                  80        0
3        1  28            0            no           never 27.89         5.7                 158        0
4        2  50            0            no         current 23.45         5.0                 155        0
5        1  76            1           yes         current 20.14         4.8                 155        0
6        2  20            0            no           never 27.32         6.6                  85        0
7        2  79            0            no         No Info 23.86         5.7                  85        0
8        1  42            0            no           never 33.64         4.8                 145        0
9        2  32            0            no           never 27.32         5.0                 100        0
10       2  53            0            no           never 27.32         6.1                  85        0
11       2  54            0            no          former 27.89         6.0                 100        0
12       2  78            0            no          former 36.05         5.0                 130        0
13       2  67            0            no           never 25.69         5.8                 200        0
14       2  76            0            no         No Info 27.32         5.0                 160        0
15       2  78            0            no         No Info 27.32         6.6                 126        0
16       1  15            0            no           never 27.89         6.1                 200        0
17       2  42            0            no           never 24.48         5.7                 158        0
18       2  42            0            no           never 27.32         5.7                  80        0
19       1  50            0            no            ever 25.72         3.5                 159        0
20       1  40            0            no         current 36.38         6.0                  90        0
21       1   5            0            no         No Info 18.80         6.2                  85        0
22       2  69            0            no           never 21.24         4.8                  85        0
23       2  72            0           yes          former 27.94         6.5                 130        0
24       2   4            0            no         No Info 13.99         4.0                 140        0
25       1  30            0            no           never 33.76         6.1                 126        0
26       1  40            0            no          former 27.85         5.8                  80        0
27       1  50            0            no           never 26.47         4.0                 158        0
28       1  43            0            no           never 26.08         6.1                 155        0
29       2  53            0            no         No Info 31.75         4.0                 200        0
30       1  50            0            no         No Info 25.15         4.0                 145        0
31       1  43            0            no           never 26.08         6.1                 155        0
32       2  53            0            no         No Info 31.75         4.0                 200        0
33       2  41            0            no         current 22.01         6.2                 126        0
34       2  20            0            no           never 22.19         3.5                 100        0
```

Categorical variables undergo factor conversion with meaningful numeric labels: gender becomes 1 (Male) and 2 (Female), while heart disease is labeled "yes"/"no". This enables statistical analysis of categorical data and prepares features for machine learning algorithms requiring numeric input.

7. Normalization

Code:

```
normalizedata <- newdata
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

normalizedata$age <- normalize(normalizedata$age)
normalizedata$bmi <- normalize(normalizedata$bmi)
normalizedata$HbA1c_level <- normalize(normalizedata$HbA1c_level)
normalizedata$blood_glucose_level <- normalize(normalizedata$blood_glucose_level)
normalizedata
```

Output:

```
> normalizedata <- newdata
> normalize <- function(x) {
+   return((x - min(x)) / (max(x) - min(x)))
+ }
>
> normalizedata$age <- normalize(normalizedata$age)
> normalizedata$bmi <- normalize(normalizedata$bmi)
> normalizedata$HbA1c_level <- normalize(normalizedata$HbA1c_level)
> normalizedata$blood_glucose_level <- normalize(normalizedata$blood_glucose_level)
> normalizedata
    gender        age hypertension heart_disease smoking_history        bmi HbA1c_level blood_glucose_level
1   Female 1.00000000            0             1           never 0.22630835  0.56363636          0.27272727
2   Female 0.66233766            0             0         No Info 0.26934734  0.56363636          0.00000000
3     Male 0.32467532            0             0           never 0.28086482  0.40000000          0.35454545
4   Female 0.61038961            0             0         current 0.19114973  0.27272727          0.34090909
5     Male 0.94805195            1             1         current 0.12426753  0.23636364          0.34090909
6   Female 0.22077922            0             0           never 0.26934734  0.56363636          0.02272727
7   Female 0.98701299            0             0         No Info 0.19943423  0.40000000          0.02272727
8     Male 0.50649351            0             0           never 0.39704991  0.23636364          0.29545455
9   Female 0.37662338            0             0           never 0.26934734  0.27272727          0.09090909
10  Female 0.64935065            0             0           never 0.26934734  0.47272727          0.02272727
11  Female 0.66233766            0             0          former 0.28086482  0.45454545          0.09090909
12  Female 0.97402597            0             0          former 0.44574662  0.27272727          0.22727273
13  Female 0.83116883            0             0           never 0.23641140  0.41818182          0.54545455
14  Female 0.94805195            0             0         No Info 0.26934734  0.27272727          0.36363636
15  Female 0.97402597            0             0         No Info 0.26934734  0.56363636          0.20909091
16    Male 0.15584416            0             0           never 0.28086482  0.47272727          0.54545455
17  Female 0.50649351            0             0           never 0.21196201  0.40000000          0.35454545
18  Female 0.50649351            0             0           never 0.26934734  0.40000000          0.00000000
19    Male 0.61038961            0             0            ever 0.23701758  0.00000000          0.35909091
20    Male 0.48051948            0             0         current 0.45241463  0.45454545          0.04545455
21    Male 0.02597403            0             0         No Info 0.09719135  0.49090909          0.02272727
22  Female 0.85714286            0             0           never 0.14649424  0.23636364          0.02272727
23  Female 0.89610390            0             1          former 0.28187513  0.54545455          0.22727273
24  Female 0.01298701            0             0         No Info 0.00000000  0.09090909          0.27272727
25    Male 0.35064935            0             0           never 0.39947464  0.47272727          0.20909091
```

Continuous variables (age, BMI, HbA1c, glucose levels) are scaled to a 0-1 range using min-max normalization. The transformation preserves relative differences while eliminating scale disparities between features, crucial for distance-based algorithms and comparative analysis.

8. Duplicate Row Removal

Code:

```
duplicate_data <- newdata
duplicated(duplicate_data)
sum(duplicated(duplicate_data))
duplicate_data <- distinct(duplicate_data)
sum(duplicated(duplicate_data))
```

Output:

```
> duplicate_data <- newdata
> duplicated(duplicate_data)
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [18] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
 [35] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [52] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [69] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [86] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
[103] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[120] FALSE FALSE FALSE
> sum(duplicated(duplicate_data))
[1] 3
> duplicate_data <- distinct(duplicate_data)
> sum(duplicated(duplicate_data))
[1] 0
```

Exact duplicate rows are identified and removed using dplyr::distinct(), verified by checking the sum of duplicated entries before and after processing. This ensures each observation is unique, preventing skewed analysis from repeated records.

9. Data Filtering

Code:

```
filtered_data <- subset(newdata, age > 79)
filtered_data
```

Output:

```
> filtered_data <- subset(newdata, age > 79)
> filtered_data
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1   Female  80            0             1           never 25.19         6.6                 140        0
69    Male  80            0             0          former 24.42         4.0                 160        0
82  Female  80            1             0           never 27.32         6.8                 280        1
84    Male  80            0             0           never 22.06         9.0                 155        1
94    Male  80            0             0           never 23.25         6.1                 159        1
97  Female  80            0             0          former 36.18         6.5                 200        1
102   Male  80            0             1          former 24.36         7.5                 280        1
113 Female  80            0             0           never 27.32         6.0                 200        1
```

A subset of elderly patients (age > 79) is isolated using base R's subset() function. This demonstrates targeted data extraction for cohort-specific analysis while maintaining original data structure and variable relationships.

10. Handle Imbalance Dataset

Oversampling Code:

```
class_distribution <- table(newdata$diabetes)
print(class_distribution)
if (class_distribution[1] > class_distribution[2]) {
  majority <- filter(newdata, diabetes == 0)
  minority <- filter(newdata, diabetes == 1)
} else {
  majority <- filter(newdata, diabetes == 1)
  minority <- filter(newdata, diabetes == 0)
}

set.seed(123)
oversampled_minority <- minority %>% sample_n(nrow(majority), replace = TRUE)
oversampled_data <- bind_rows(majority, oversampled_minority)
table(oversampled_data$diabetes)
oversampled_data
```

Output:

```
> class_distribution <- table(newdata$diabetes)
> print(class_distribution)

 0  1
70 52
> if (class_distribution[1] > class_distribution[2]) {
+   majority <- filter(newdata, diabetes == 0)
+   minority <- filter(newdata, diabetes == 1)
+ } else {
+   majority <- filter(newdata, diabetes == 1)
+   minority <- filter(newdata, diabetes == 0)
+ }
>
> set.seed(123)
> oversampled_minority <- minority %>% sample_n(nrow(majority), replace = TRUE)
> oversampled_data <- bind_rows(majority, oversampled_minority)
> table(oversampled_data$diabetes)

 0  1
70 70
> oversampled_data
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1   Female  80            0             1           never 25.19         6.6                 140        0
2   Female  54            0             0         No Info 27.32         6.6                  80        0
3     Male  28            0             0           never 27.89         5.7                 158        0
4   Female  50            0             0         current 23.45         5.0                 155        0
5     Male  76            1             1         current 20.14         4.8                 155        0
6   Female  20            0             0           never 27.32         6.6                  85        0
7   Female  79            0             0         No Info 23.86         5.7                  85        0
8     Male  42            0             0           never 33.64         4.8                 145        0
9   Female  32            0             0           never 27.32         5.0                 100        0
10  Female  53            0             0           never 27.32         6.1                  85        0
11  Female  54            0             0          former 27.89         6.0                 100        0
12  Female  78            0             0          former 36.05         5.0                 130        0
13  Female  67            0             0           never 25.69         5.8                 200        0
```

Under sampling Code:

```
undersampled_majority <- majority %>% sample_n(nrow(minority), replace = FALSE)
undersampled_data <- bind_rows(undersampled_majority, minority)
table(undersampled_data$diabetes)
undersampled_data
```

Output:

```
> undersampled_majority <- majority %>% sample_n(nrow(minority), replace = FALSE)
> undersampled_data <- bind_rows(undersampled_majority, minority)
> table(undersampled_data$diabetes)

 0  1
52 52
> undersampled_data
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
1	Male	5	0	0	No Info	27.32	6.6	130	0
2	Male	43	0	0	never	26.08	6.1	155	0
3	Female	74	0	0	No Info	28.12	5.0	100	0
4	Female	20	0	0	never	22.19	3.5	100	0
5	Female	50	0	0	current	23.45	5.0	155	0
6	Female	67	0	0	never	25.69	5.8	200	0
7	Male	76	1	1	current	20.14	4.8	155	0
8	Female	50	0	0	not current	30.22	5.7	100	0
9	Female	21	0	0	never	26.10	5.8	140	0
10	Male	30	0	0	never	33.76	6.1	126	0
11	Female	30	0	0	current	27.32	6.5	158	0
12	Female	69	0	0	never	21.24	4.8	85	0
13	Male	3	0	0	No Info	15.80	6.2	90	0
14	Female	53	0	0	No Info	31.75	4.0	200	0
15	Female	38	0	0	never	28.27	6.2	155	0
16	Male	56	0	0	never	26.78	4.8	200	0
17	Female	72	0	1	former	27.94	6.5	130	0
18	Female	76	0	0	never	23.55	5.0	85	0
19	Female	77	1	1	never	32.02	5.0	159	0
20	Male	57	0	0	never	27.32	6.1	155	0
21	Male	50	0	0	No Info	25.15	4.0	145	0
22	Female	78	0	0	former	36.05	5.0	130	0
23	Male	80	0	0	former	24.42	4.0	160	0
24	Female	19	0	0	never	27.32	5.7	145	0
25	Male	34	0	0	never	31.16	5.8	90	0

Both oversampling (replicating minority class) and undersampling (reducing majority class) are implemented to balance the diabetes outcome variable. The techniques use dplyr's sample_n() with replacement for oversampling and without replacement for undersampling, followed by row binding to create balanced datasets for modeling.

11. Split dataset Into Test and Training

Code:

```
newdata_copy <- newdata
set.seed(123)
split <- sample(1:nrow(newdata_copy), 0.8 * nrow(newdata_copy))
train_data <- newdata_copy[split, ]
test_data <- newdata_copy[-split, ]
train_data
test_data
```

Output:

```
> newdata_copy <- newdata
> set.seed(123)
> split <- sample(1:nrow(newdata_copy), 0.8 * nrow(newdata_copy))
> train_data <- newdata_copy[split, ]
> test_data <- newdata_copy[-split, ]
> train_data
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
31    Male  43            0             0           never 26.08         6.1                 155        0
79  Female  36            0             0         current 32.27         6.2                 220        1
51  Female  21            0             0           never 26.10         5.8                 140        0
14  Female  76            0             0         No Info 27.32         5.0                 160        0
67  Female  26            0             0           never 26.45         5.7                 158        0
42  Female  67            0             0         No Info 27.32         3.5                 160        0
50  Female  74            0             0         No Info 28.12         5.0                 100        0
43  Female  44            0             0           never 24.93         6.1                 100        0
101   Male  71            0             0           never 27.09         8.2                 200        1
119 Female  43            0             0           never 27.73         8.8                 145        1
25    Male  30            0             0           never 33.76         6.1                 126        0
90    Male  55            0             0         No Info 27.32         6.8                 159        1
91    Male  57            1             1     not current 27.77         6.6                 160        1
69    Male  80            0             0          former 24.42         4.0                 160        0
110   Male  37            0             0           never 37.24         7.0                 126        1
57  Female  50            0             0         No Info 28.16         5.0                  90        0
92  Female  43            0             0           never 27.32         6.2                 155        1
9   Female  32            0             0           never 27.32         5.0                 100        0
> test_data
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1   Female  80            0             1           never 25.19         6.6                 140        0
2   Female  54            0             0         No Info 27.32         6.6                  80        0
10  Female  53            0             0           never 27.32         6.1                  85        0
11  Female  54            0             0          former 27.89         6.0                 100        0
18  Female  42            0             0           never 27.32         5.7                  80        0
19    Male  50            0             0            ever 25.72         3.5                 159        0
20    Male  40            0             0         current 36.38         6.0                  90        0
24  Female   4            0             0         No Info 13.99         4.0                 140        0
28    Male  43            0             0           never 26.08         6.1                 155        0
33  Female  41            0             0         current 22.01         6.2                 126        0
37  Female  15            0             0         No Info 21.76         4.5                 130        0
45  Female  60            0             0           never 18.03         4.0                 159        0
52  Female  30            0             0         current 27.32         6.5                 158        0
65  Female  41            0             0           never 27.45         5.7                 130        0
66  Female  11            0             0         No Info 17.98         6.5                 159        0
71  Female  44            0             0           never 19.31         6.5                 200        1
80  Female  60            0             0           never 27.32         7.5                 300        1
86    Male  53            0             0         current 30.80         6.6                 280        1
88    Male  76            0             0           never 31.90         7.5                 155        1
96  Female  42            0             0           never 24.81         9.0                 159        1
103   Male  59            0             0         current 29.20         8.2                 220        1
107   Male  71            0             0           never 26.53         8.8                 159        1
```

The dataset is partitioned into training (80%) and testing (20%) sets using random sampling with set.seed() for reproducibility. Row indexing separates the data while preserving all variables, creating ready-to-use subsets for model development and validation.

12. Central tendency (mean/median/mode) comparison across gender groups

Code:

```
aggregate(age ~ gender, data = newdata_copy, FUN = function(x) c(mean = mean(x), median = median(x), mode = find_mode(x)))
```

Output:

```
> aggregate(age ~ gender, data = newdata_copy, FUN = function(x) c(mean = mean(x), median = median(x), mode = find_mode(x)))
  gender         age.mean age.median age.mode
1 Female 51.2567567567568         52       43
2   Male 47.8333333333333         50       43
```

The analysis compares mean, median, and mode of age across gender groups using aggregate(). This reveals whether males and females have different age distributions, helping identify demographic patterns in the dataset. The mean shows average age, median indicates the middle value, and mode reflects the most frequent age for each gender.

13. Age Distribution by Hypertension Status

Code:

```
aggregate(age ~ hypertension, data = newdata_copy, FUN = function(x) c(mean = mean(x), median = median(x), mode = find_mode(x)))
```

Output:

```
> aggregate(age ~ hypertension, data = newdata_copy, FUN = function(x) c(mean = mean(x), median = median(x), mode = find_mode(x)))
  hypertension age.mean age.median age.mode
1            0   48.875         50       43
2            1     61.5         60       33
```

There is a substantial difference in age between those with and without hypertension. Individuals with hypertension have a much higher mean age (61.5 years) compared to those without (48.87 years). This aligns with clinical knowledge that hypertension risk increases with age.

14. Spread Metrics by Gender

Code:

```
spread_stats <- aggregate(age ~ gender, data = newdata_copy,
                          FUN = function(x) c(
                            range = max(x) - min(x),
                            IQR = IQR(x),
                            var = var(x),
                            sd = sd(x)
                          ))
spread_stats
```

Output:

```
> spread_stats <- aggregate(age ~ gender, data = newdata_copy,
+                           FUN = function(x) c(
+                             range = max(x) - min(x),
+                             IQR = IQR(x),
+                             var = var(x),
+                             sd = sd(x)
+                           ))
> spread_stats
  gender age.range   age.IQR   age.var    age.sd
1 Female  77.00000  25.75000 397.80989  19.94517
2   Male  77.00000  23.00000 457.16312  21.38137
```

The spread of age is very similar between females and males, with identical ranges and IQRs, and almost identical variances and standard deviations. This suggests similar age distributions across these gender groups.

15. Statistical Analysis: BMI by Diabetes Status

Code:

```
bmi_by_diabetes <- aggregate(bmi ~ diabetes, data = newdata_copy,
                             FUN = function(x) c(mean = mean(x),
                                                 median = median(x),
                                                 sd = sd(x),
                                                 min = min(x),
                                                 max = max(x)
                                                 ))

bmi_by_diabetes
```

Output:

```
> bmi_by_diabetes <- aggregate(bmi ~ diabetes, data = newdata_copy,
+                              FUN = function(x) c(mean = mean(x),
+                                                  median = median(x),
+                                                  sd = sd(x),
+                                                  min = min(x),
+                                                  max = max(x)
+                                                  ))
> bmi_by_diabetes
  diabetes  bmi.mean bmi.median    bmi.sd   bmi.min   bmi.max
1        0 25.728000  27.165000  4.626596 13.990000 36.380000
2        1 30.805962  27.320000  8.023880 19.310000 63.480000
>
```

There is a clear difference in BMI between individuals with and without diabetes. Those who tested positive for diabetes have a higher mean BMI (30.80) compared to those who tested negative (25.72). This supports the established link between obesity and type 2 diabetes risk.