Final Project

Md Saiful Hasan

-----------------------------------------------------------------------

EPBI 5208-701 Data Management and Analysis

-----------------------------------------------------------------------

Submitted to: Sezgin Ciftci, PhD

College of Public Health

Temple University

------------------------------------------------------------------------

Date of Submission: 4h December 2024

------------------------------------------------------------------------

Student Email: md.saiful.hasan@temple.edu (Temple ID:

916566955)

**Data Source**: National Health and Nutrition Examination Survey (NHANES), August 2021 –August 2023

**Instructions:**

You may use **R**, **SAS**, or both for this project. Include annotated code for each step, with clear explanations of major steps. The project should be submitted as a single word or pdf document.

Clarity and readability are crucial for this project. Ensure that your code, results, discussions, and any

outputs are well-organized and easy to follow. Your project will be evaluated based on how well you present and explain each step, so make sure that each part of your work is clear and logically structured. Projects lacking clarity and organization may not be evaluated fully.

**Variables:**

The 24 variables for this project are listed below:

• Age (RIDAGEYR)

• Gender (RIAGENDR)

• Race/Ethnicity (RIDRETH1)

• Education Level (DMDEDUC2)

• Ratio of family income to poverty (INDFMPIR)

• Health Insurance Coverage (HIQ011)

• Body Mass Index (BMI) (BMXBMI)

• Smoking Status (SMQ040)

• Alcohol Use in Past 12 Months (ALQ121)

• Physical Activity Level (PAD680)

• Hours of Sleep (SLD012)

• Hypertension Diagnosis (BPQ020)

• Diabetes Diagnosis (DIQ010)

• Cancer Diagnosis (MCQ220)

• Coronary Disease Diagnosis (MCQ160C)

• Stroke History (MCQ160F)

• Total Cholesterol (LBXTC)

• Blood Glucose (LBXGLU)

• Prescription for Blood Pressure Medication (BPQ150)

• Prescription for Diabetes Medication (DIQ050)

• Prescription for Cholesterol Medication (BPQ101D)

• Self-Reported AIDS Test Status (HSQ590)

• Depression Score (PHQ-9) (DPQ020)

• Quality of Sleep (DPQ030)

**Project Sections**

**1. Introduction (5 points)**

Provide a brief overview of the NHANES dataset and the project’s objectives.

Answer:

[*Overview of the NHANES Dataset (August 2021 – August 2023)*

*The National Health and Nutrition Examination Survey (NHANES) is a major source of health, nutrition examination survey data of population in USA, collected and managed by the Centers for Disease Control and Prevention (CDC). NHANES combines interviews, physical examinations, and laboratory tests data file for public use. Data files released to inform data users to get insights into the prevalence of chronic diseases, risk factors, and health disparities.*

*The survey is conducted in two-year cycles, with the August 2021–August 2023 cycle focusing on providing updated data following disruptions caused by the COVID-19 pandemic. This dataset includes demographic, socioeconomic, dietary, and health-related information, along with biomarker and laboratory results, reflecting the health trends and challenges faced by the U.S. population. NHANES consists of six data segments: a demographic dataset with 11,933 observations and 27 variables; dietary data comprising five separate datasets; an examination segment with four datasets; a laboratory segment with 20 datasets; a questionnaire segment with 34 datasets; and six restricted access datasets. Each segment contains unique variables collected under varying conditions and times. From these segments, I have identified and selected 24 variables across four sections for further review*.]

**2. Initial Exploration and Data Cleaning (40 points)**

**a.** Download and import the NHANES (8/2021 – 8/2023) dataset. Load only the specified 24

variables. Verify each variable's data type (e.g., numerical, categorical) and adjust as necessary. *(5 points)*

**Answer:**

**R code**:

[library(haven)

## Demographics, Questionnaire, laboratory, and examination 24 variables RIDAGEYR", "RIAGENDR", "RIDRETH1", "DMDEDUC2", "INDFMPIR","BMXBMI", "LBXTC", "LBXGLU","HIQ011","SMQ040","ALQ121","PAD680","SLD012",

"BPQ020","DIQ010","MCQ220","MCQ160C","MCQ160F",

"BPQ150","DIQ050","BPQ101D","HSQ590","DPQ020","DPQ030")

demographics <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Demo/DEMO\_L.XPT")

exam\_bmxmbi <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Examinat/BMX\_L.XPT")

laborat\_lbxglu <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/GLU\_L.XPT")

laborat\_lbxtc <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/TCHOL\_L.XPT")

quest\_hiq011 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HIQ\_L.XPT")

quest\_smq040 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SMQ\_L.XPT")

quest\_alq121 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/ALQ\_L.XPT")

quest\_pad680 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/PAQ\_L.XPT")

quest\_sld012 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SLQ\_L.XPT")

quest\_bpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/BPQ\_L.XPT")

quest\_diq010 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DIQ\_L.XPT")

quest\_mcq220 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/MCQ\_L.XPT")

quest\_hsq590 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HSQ\_L.XPT")

quest\_dpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DPQ\_L.XPT")

# Combine the datasets if necessary or extract variables directly

# List of datasets

quest\_new\_datasets <- list(demographics, exam\_bmxmbi,laborat\_lbxglu,laborat\_lbxtc, quest\_hiq011, quest\_smq040, quest\_alq121, quest\_pad680, quest\_sld012,

quest\_bpq020, quest\_diq010, quest\_mcq220, quest\_hsq590, quest\_dpq020)

# List of variables to extract summaries for

quest\_24\_vars <- c("RIDAGEYR", "RIAGENDR", "RIDRETH1", "DMDEDUC2", "INDFMPIR","BMXBMI", "LBXTC", "LBXGLU","HIQ011","SMQ040","ALQ121","PAD680","SLD012",

"BPQ020","DIQ010","MCQ220","MCQ160C","MCQ160F",

"BPQ150","DIQ050","BPQ101D","HSQ590","DPQ020","DPQ030")

for (i in 1:length(quest\_new\_datasets)) {

# Filter only the selected variables that exist in each dataset

dataset <- quest\_new\_datasets[[i]]

available\_vars <- intersect(quest\_24\_vars, names(dataset))

# Display summary and first 5 rows if variables are available in this dataset

if (length(available\_vars) > 0) {

cat("Dataset", i, "Summary:\n")

print(summary(dataset[ , available\_vars]))

cat("First 5 Rows:\n")

print(head(dataset[ , available\_vars], 5))

cat("\n-------------------\n")

} else {

cat("Dataset", i, "does not contain any of the selected variables.\n")

}

}]

**R output**:

[> library(haven)

> demographics <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Demo/DEMO\_L.XPT")

> exam\_bmxmbi <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Examinat/BMX\_L.XPT")

> laborat\_lbxglu <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/GLU\_L.XPT")

> laborat\_lbxtc <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/TCHOL\_L.XPT")

> quest\_hiq011 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HIQ\_L.XPT")

> quest\_smq040 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SMQ\_L.XPT")

> quest\_alq121 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/ALQ\_L.XPT")

> quest\_pad680 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/PAQ\_L.XPT")

> quest\_sld012 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SLQ\_L.XPT")

> quest\_bpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/BPQ\_L.XPT")

> quest\_diq010 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DIQ\_L.XPT")

> quest\_mcq220 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/MCQ\_L.XPT")

> quest\_hsq590 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HSQ\_L.XPT")

> quest\_dpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DPQ\_L.XPT")

> # Combine the datasets if necessary or extract variables directly

> # List of datasets

> quest\_new\_datasets <- list(demographics, exam\_bmxmbi,laborat\_lbxglu,laborat\_lbxtc, quest\_hiq011, quest\_smq040, quest\_alq121, quest\_pad680, quest\_sld012,

+ quest\_bpq020, quest\_diq010, quest\_mcq220, quest\_hsq590, quest\_dpq020)

> # List of variables to extract summaries for

> quest\_24\_vars <- c("RIDAGEYR", "RIAGENDR", "RIDRETH1", "DMDEDUC2", "INDFMPIR","BMXBMI", "LBXTC", "LBXGLU","HIQ011","SMQ040","ALQ121","PAD680","SLD012",

+ "BPQ020","DIQ010","MCQ220","MCQ160C","MCQ160F",

+ "BPQ150","DIQ050","BPQ101D","HSQ590","DPQ020","DPQ030") # Replace with actual variable names

> for (i in 1:length(quest\_new\_datasets)) {

+ # Filter only the selected variables that exist in each dataset

+ dataset <- quest\_new\_datasets[[i]]

+ available\_vars <- intersect(quest\_24\_vars, names(dataset))

+

+ # Display summary and first 5 rows if variables are available in this dataset

+ if (length(available\_vars) > 0) {

+ cat("Dataset", i, "Summary:\n")

+ print(summary(dataset[ , available\_vars]))

+ cat("First 5 Rows:\n")

+ print(head(dataset[ , available\_vars], 5))

+ cat("\n-------------------\n")

+ } else {

+ cat("Dataset", i, "does not contain any of the selected variables.\n")

+ }

+ }

Dataset 1 Summary:

RIDAGEYR RIAGENDR RIDRETH1 DMDEDUC2 INDFMPIR

Min. : 0.00 Min. :1.000 Min. :1.000 Min. :1.000 Min. :0.000

1st Qu.:13.00 1st Qu.:1.000 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:1.180

Median :37.00 Median :2.000 Median :3.000 Median :4.000 Median :2.500

Mean :38.32 Mean :1.533 Mean :3.105 Mean :3.805 Mean :2.708

3rd Qu.:62.00 3rd Qu.:2.000 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:4.500

Max. :80.00 Max. :2.000 Max. :5.000 Max. :9.000 Max. :5.000

NA's :4139 NA's :2041

First 5 Rows:

# A tibble: 5 × 5

RIDAGEYR RIAGENDR RIDRETH1 DMDEDUC2 INDFMPIR

<dbl> <dbl> <dbl> <dbl> <dbl>

1 43 1 5 5 5

2 66 1 3 5 5

3 44 2 2 3 1.41

4 5 2 5 NA 1.53

5 2 1 3 NA 3.6

-------------------

Dataset 2 Summary:

BMXBMI

Min. :11.10

1st Qu.:21.60

Median :26.40

Mean :27.25

3rd Qu.:31.70

Max. :74.80

NA's :389

First 5 Rows:

# A tibble: 5 × 1

BMXBMI

<dbl>

1 27

2 33.5

3 29.7

4 23.8

5 NA

-------------------

Dataset 3 Summary:

LBXGLU

Min. : 59.0

1st Qu.: 93.0

Median :100.0

Mean :107.9

3rd Qu.:109.0

Max. :561.0

NA's :324

First 5 Rows:

# A tibble: 5 × 1

LBXGLU

<dbl>

1 113

2 99

3 156

4 100

5 88

-------------------

Dataset 4 Summary:

LBXTC

Min. : 62.0

1st Qu.:151.0

Median :178.0

Mean :181.5

3rd Qu.:207.0

Max. :438.0

NA's :1178

First 5 Rows:

# A tibble: 5 × 1

LBXTC

<dbl>

1 264

2 214

3 187

4 183

5 203

-------------------

Dataset 5 Summary:

HIQ011

Min. :1.000

1st Qu.:1.000

Median :1.000

Mean :1.097

3rd Qu.:1.000

Max. :9.000

NA's :23

First 5 Rows:

# A tibble: 5 × 1

HIQ011

<dbl>

1 1

2 1

3 1

4 1

5 1

-------------------

Dataset 6 Summary:

SMQ040

Min. :1.00

1st Qu.:1.00

Median :3.00

Mean :2.34

3rd Qu.:3.00

Max. :3.00

NA's :5772

First 5 Rows:

# A tibble: 5 × 1

SMQ040

<dbl>

1 3

2 3

3 NA

4 NA

5 NA

-------------------

Dataset 7 Summary:

ALQ121

Min. : 0.000

1st Qu.: 2.000

Median : 5.000

Mean : 5.031

3rd Qu.: 8.000

Max. :99.000

NA's :1415

First 5 Rows:

# A tibble: 5 × 1

ALQ121

<dbl>

1 NA

2 2

3 10

4 4

5 0

-------------------

Dataset 8 Summary:

PAD680

Min. : 0

1st Qu.: 180

Median : 300

Mean : 447

3rd Qu.: 480

Max. :9999

NA's :15

First 5 Rows:

# A tibble: 5 × 1

PAD680

<dbl>

1 360

2 480

3 240

4 60

5 180

-------------------

Dataset 9 Summary:

SLD012

Min. : 2.000

1st Qu.: 7.000

Median : 8.000

Mean : 7.757

3rd Qu.: 8.500

Max. :14.000

NA's :113

First 5 Rows:

# A tibble: 5 × 1

SLD012

<dbl>

1 9.5

2 9

3 8

4 7.5

5 8

-------------------

Dataset 10 Summary:

BPQ020 BPQ150 BPQ101D

Min. :1.000 Min. :1.000 Min. :1.000

1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1.000

Median :2.000 Median :1.000 Median :2.000

Mean :1.659 Mean :1.187 Mean :1.777

3rd Qu.:2.000 3rd Qu.:1.000 3rd Qu.:2.000

Max. :9.000 Max. :9.000 Max. :9.000

NA's :3 NA's :5532 NA's :3

First 5 Rows:

# A tibble: 5 × 3

BPQ020 BPQ150 BPQ101D

<dbl> <dbl> <dbl>

1 1 1 2

2 1 1 2

3 2 NA 1

4 2 NA 2

5 2 NA 2

-------------------

Dataset 11 Summary:

DIQ010 DIQ050

Min. :1.000 Min. :1.000

1st Qu.:2.000 1st Qu.:1.000

Median :2.000 Median :2.000

Mean :1.934 Mean :1.682

3rd Qu.:2.000 3rd Qu.:2.000

Max. :9.000 Max. :2.000

NA's :4 NA's :10663

First 5 Rows:

# A tibble: 5 × 2

DIQ010 DIQ050

<dbl> <dbl>

1 2 NA

2 2 NA

3 1 2

4 2 NA

5 2 NA

-------------------

Dataset 12 Summary:

MCQ220 MCQ160C MCQ160F

Min. :1.000 Min. :1.000 Min. :1.000

1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000

Median :2.000 Median :2.000 Median :2.000

Mean :1.856 Mean :1.979 Mean :1.972

3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:2.000

Max. :9.000 Max. :9.000 Max. :9.000

NA's :3937 NA's :3937 NA's :3938

First 5 Rows:

# A tibble: 5 × 3

MCQ220 MCQ160C MCQ160F

<dbl> <dbl> <dbl>

1 2 2 2

2 1 2 2

3 2 2 2

4 NA NA NA

5 NA NA NA

-------------------

Dataset 13 Summary:

HSQ590

Min. :1.000

1st Qu.:1.000

Median :2.000

Mean :1.724

3rd Qu.:2.000

Max. :9.000

NA's :864

First 5 Rows:

# A tibble: 5 × 1

HSQ590

<dbl>

1 NA

2 1

3 1

4 1

5 2

-------------------

Dataset 14 Summary:

DPQ020 DPQ030

Min. :0.0000 Min. :0.0000

1st Qu.:0.0000 1st Qu.:0.0000

Median :0.0000 Median :0.0000

Mean :0.4786 Mean :0.7935

3rd Qu.:1.0000 3rd Qu.:1.0000

Max. :9.0000 Max. :9.0000

NA's :819 NA's :821

First 5 Rows:

# A tibble: 5 × 2

DPQ020 DPQ030

<dbl> <dbl>

1 NA NA

2 0 1

3 0 1

4 0 0

5 0 0

-------------------

]

**b.** Perform an initial exploration to understand the dataset. Check for and document any missing data

in each variable. Summarize the percentage of missing data by variable and consider the potential impact of missing data on future analyses. *(10 points)*

**Answer:**

**R code**:

[# Function to calculate missing data percentage

calculate\_missing\_data <- function(dataset, vars) {

# Ensure dataset is a data frame

if (!is.data.frame(dataset)) {

stop("Dataset must be a data frame.")

}

# Identify overlapping variables

available\_vars <- intersect(vars, names(dataset))

# Calculate missing percentage if variables are found

if (length(available\_vars) > 0) {

missing\_data <- sapply(dataset[ , available\_vars, drop = FALSE], function(x) sum(is.na(x)) / length(x) \* 100)

return(missing\_data)

} else {

return(NULL) # Return NULL if no variables are found

}

}

# Initialize a list to store missing data results

missing\_data\_results <- list()

# Loop through each dataset and calculate missing data percentage

for (i in seq\_along(quest\_new\_datasets)) {

dataset <- quest\_new\_datasets[[i]]

missing\_data <- calculate\_missing\_data(dataset, quest\_24\_vars)

if (!is.null(missing\_data)) {

cat("Dataset", i, "Missing Data Percentage:\n")

print(missing\_data)

cat("\n-------------------\n")

missing\_data\_results[[paste0("Dataset\_", i)]] <- missing\_data

} else {

cat("Dataset", i, "does not contain any of the selected variables.\n")

}

}

# Combine results into a summary table if needed

missing\_data\_summary <- do.call(rbind, lapply(names(missing\_data\_results), function(name) {

data.frame(Dataset = name, Variable = names(missing\_data\_results[[name]]),

Missing\_Percentage = missing\_data\_results[[name]])

}))

# Print a combined summary (optional)

print(missing\_data\_summary)]

**R output**:

[

R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"

Copyright (C) 2023 The R Foundation for Statistical Computing

Platform: x86\_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> demographics <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Demo/DEMO\_L.XPT")

Error in read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Demo/DEMO\_L.XPT") :

could not find function "read\_xpt"

> library(haven)

> demographics <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Demo/DEMO\_L.XPT")

> exam\_bmxmbi <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Examinat/BMX\_L.XPT")

> laborat\_lbxglu <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/GLU\_L.XPT")

> laborat\_lbxtc <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Laboratory/TCHOL\_L.XPT")

> quest\_hiq011 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HIQ\_L.XPT")

> quest\_smq040 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SMQ\_L.XPT")

> quest\_alq121 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/ALQ\_L.XPT")

> quest\_pad680 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/PAQ\_L.XPT")

> quest\_sld012 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/SLQ\_L.XPT")

> quest\_bpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/BPQ\_L.XPT")

> quest\_diq010 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DIQ\_L.XPT")

> quest\_mcq220 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/MCQ\_L.XPT")

> quest\_hsq590 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/HSQ\_L.XPT")

> quest\_dpq020 <- read\_xpt("/Users/mamoon/Downloads/5208\_DM\_finalProject/NHANES\_Questionnaire/DPQ\_L.XPT")

> # Combine the datasets if necessary or extract variables directly

> # List of datasets

> quest\_new\_datasets <- list(demographics, exam\_bmxmbi,laborat\_lbxglu,laborat\_lbxtc, quest\_hiq011, quest\_smq040, quest\_alq121, quest\_pad680, quest\_sld012,

+ quest\_bpq020, quest\_diq010, quest\_mcq220, quest\_hsq590, quest\_dpq020)

> # List of variables to extract summaries for

> quest\_24\_vars <- c("RIDAGEYR", "RIAGENDR", "RIDRETH1", "DMDEDUC2", "INDFMPIR","BMXBMI", "LBXTC", "LBXGLU","HIQ011","SMQ040","ALQ121","PAD680","SLD012",

+ "BPQ020","DIQ010","MCQ220","MCQ160C","MCQ160F",

+ "BPQ150","DIQ050","BPQ101D","HSQ590","DPQ020","DPQ030") # Replace with actual variable names

> View(quest\_new\_datasets)

> View(quest\_new\_datasets[[1]])

> library(dplyr) # Load necessary libraries

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

filter, lag

The following objects are masked from ‘package:base’:

intersect, setdiff, setequal, union

> print(missing\_data\_summary)

Dataset Variable Missing\_Percentage

RIDAGEYR Dataset\_1 RIDAGEYR 0.00000000

RIAGENDR Dataset\_1 RIAGENDR 0.00000000

RIDRETH1 Dataset\_1 RIDRETH1 0.00000000

DMDEDUC2 Dataset\_1 DMDEDUC2 34.68532641

INDFMPIR Dataset\_1 INDFMPIR 17.10382972

BMXBMI Dataset\_2 BMXBMI 4.39051919

LBXGLU Dataset\_3 LBXGLU 8.10810811

LBXTC Dataset\_4 LBXTC 14.60089241

HIQ011 Dataset\_5 HIQ011 0.19274281

SMQ040 Dataset\_6 SMQ040 64.02662230

ALQ121 Dataset\_7 ALQ121 22.32917784

PAD680 Dataset\_8 PAD680 0.18398136

SLD012 Dataset\_9 SLD012 1.32925538

BPQ020 Dataset\_10 BPQ020 0.03528997

BPQ150 Dataset\_10 BPQ150 65.07469709

BPQ101D Dataset\_10 BPQ101D 0.03528997

DIQ010 Dataset\_11 DIQ010 0.03405995

DIQ050 Dataset\_11 DIQ050 90.79529973

MCQ220 Dataset\_12 MCQ220 33.52350136

MCQ160C Dataset\_12 MCQ160C 33.52350136

MCQ160F Dataset\_12 MCQ160F 33.53201635

HSQ590 Dataset\_13 HSQ590 13.06122449

DPQ020 Dataset\_14 DPQ020 12.92409658

DPQ030 Dataset\_14 DPQ030 12.95565725]

**c.** Review and document the dataset’s structure, noting aspects such as the number of observations,

variable types, and overall completeness. *(5 points)*

**Answer: review\_dataset\_structure Function we have created a dataset and its name.**

**Which generate output where a summary of the dataset, including variable names, types, and missing data percentages. In the Loop function through datasets, created the quest\_new\_datasets list for 24 selected variables. Filters variables using intersect (selected\_vars, names(dataset)). Skips datasets that do not contain any of the 24 variables. Outputs the review only for the selected variables in each dataset. Selected 24 variables were analysed.**

**R code**:

[ # Function to review and document dataset structure for specific variables

review\_selected\_variables <- function(dataset, dataset\_name, selected\_vars) {

# Ensure the dataset is a data frame

if (!is.data.frame(dataset)) {

cat(dataset\_name, "is not a valid data frame.\n")

return(NULL)

}

# Filter only the selected variables present in the dataset

available\_vars <- intersect(selected\_vars, names(dataset))

if (length(available\_vars) == 0) {

cat(dataset\_name, "does not contain any of the selected variables.\n")

return(NULL)

}

# Number of observations and variables

num\_obs <- nrow(dataset)

num\_vars <- length(available\_vars)

# Variable types

var\_types <- sapply(dataset[ , available\_vars, drop = FALSE], class)

# Percentage of missing values for each variable

missing\_percent <- sapply(dataset[ , available\_vars, drop = FALSE], function(x) sum(is.na(x)) / length(x) \* 100)

# Combine the structure information into a summary table

structure\_summary <- data.frame(

Variable = available\_vars,

Type = var\_types,

Missing\_Percentage = round(missing\_percent, 2),

stringsAsFactors = FALSE

)

# Print the overview

cat("\nDataset:", dataset\_name, "\n")

cat("Number of Observations:", num\_obs, "\n")

cat("Number of Selected Variables:", num\_vars, "\n")

cat("Overall Missing Data Percentage (Selected Variables):", round(mean(missing\_percent), 2), "%\n")

cat("\nVariable Summary (Selected):\n")

print(structure\_summary)

cat("\n-------------------\n")

return(structure\_summary)

}

# Initialize a list to store all summaries

all\_summaries <- list()

# List of datasets and selected variables

selected\_vars <- quest\_24\_vars # Use only the selected variables

# Loop through each dataset and review structure

for (i in seq\_along(quest\_new\_datasets)) {

dataset <- quest\_new\_datasets[[i]]

dataset\_name <- paste0("Dataset\_", i)

summary\_table <- review\_selected\_variables(dataset, dataset\_name, selected\_vars)

if (!is.null(summary\_table)) {

all\_summaries[[dataset\_name]] <- summary\_table

}

}]

**R output**:

[Dataset: Dataset\_1

Number of Observations: 11933

Number of Selected Variables: 5

Overall Missing Data Percentage (Selected Variables): 10.36 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

RIDAGEYR RIDAGEYR numeric 0.00

RIAGENDR RIAGENDR numeric 0.00

RIDRETH1 RIDRETH1 numeric 0.00

DMDEDUC2 DMDEDUC2 numeric 34.69

INDFMPIR INDFMPIR numeric 17.10

-------------------

Dataset: Dataset\_2

Number of Observations: 8860

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 4.39 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

BMXBMI BMXBMI numeric 4.39

-------------------

Dataset: Dataset\_3

Number of Observations: 3996

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 8.11 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

LBXGLU LBXGLU numeric 8.11

-------------------

Dataset: Dataset\_4

Number of Observations: 8068

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 14.6 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

LBXTC LBXTC numeric 14.6

-------------------

Dataset: Dataset\_5

Number of Observations: 11933

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 0.19 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

HIQ011 HIQ011 numeric 0.19

-------------------

Dataset: Dataset\_6

Number of Observations: 9015

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 64.03 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

SMQ040 SMQ040 numeric 64.03

-------------------

Dataset: Dataset\_7

Number of Observations: 6337

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 22.33 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

ALQ121 ALQ121 numeric 22.33

-------------------

Dataset: Dataset\_8

Number of Observations: 8153

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 0.18 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

PAD680 PAD680 numeric 0.18

-------------------

Dataset: Dataset\_9

Number of Observations: 8501

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 1.33 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

SLD012 SLD012 numeric 1.33

-------------------

Dataset: Dataset\_10

Number of Observations: 8501

Number of Selected Variables: 3

Overall Missing Data Percentage (Selected Variables): 21.72 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

BPQ020 BPQ020 numeric 0.04

BPQ150 BPQ150 numeric 65.07

BPQ101D BPQ101D numeric 0.04

-------------------

Dataset: Dataset\_11

Number of Observations: 11744

Number of Selected Variables: 2

Overall Missing Data Percentage (Selected Variables): 45.41 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

DIQ010 DIQ010 numeric 0.03

DIQ050 DIQ050 numeric 90.80

-------------------

Dataset: Dataset\_12

Number of Observations: 11744

Number of Selected Variables: 3

Overall Missing Data Percentage (Selected Variables): 33.53 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

MCQ220 MCQ220 numeric 33.52

MCQ160C MCQ160C numeric 33.52

MCQ160F MCQ160F numeric 33.53

-------------------

Dataset: Dataset\_13

Number of Observations: 6615

Number of Selected Variables: 1

Overall Missing Data Percentage (Selected Variables): 13.06 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

HSQ590 HSQ590 numeric 13.06

-------------------

Dataset: Dataset\_14

Number of Observations: 6337

Number of Selected Variables: 2

Overall Missing Data Percentage (Selected Variables): 12.94 %

Variable Summary (Selected):

Variable Type Missing\_Percentage

DPQ020 DPQ020 numeric 12.92

DPQ030 DPQ030 numeric 12.96]

**d.** Conduct checks to confirm that the data is logically consistent (e.g., values fall within expected ranges for continuous variables and categories are valid for categorical variables) and make sure that you fixed any inconsistencies. *(10 points)*

**Answer:**

**R code**:

[ expected\_values <- list(

# Continuous variables: specify ranges

RIDAGEYR = c(0, 120), # Age range: 0-120 years

BMXBMI = c(10, 80), # BMI range: 10-80

LBXTC = c(50, 400), # Total Cholesterol range: 50-400 mg/dL

LBXGLU = c(40, 500), # Glucose range: 40-500 mg/dL

# Categorical variables: specify valid categories

RIAGENDR = c(1, 2), # Gender: 1 = Male, 2 = Female

RIDRETH1 = c(1, 2, 3, 4, 5), # Race/Ethnicity categories

HIQ011 = c(1, 2, 7, 9), # Health Insurance: 1 = Yes, 2 = No, 7/9 = Refused/Don't know

SMQ040 = c(1, 2, 7, 9), # Smoker status: 1 = Every day, 2 = Some days, etc.

ALQ121 = c(1, 2, 7, 9), # Alcohol consumption: 1 = Yes, 2 = No, etc.

BPQ020 = c(1, 2, 7, 9) # Blood Pressure: 1 = Yes, 2 = No, etc.

)

# Function to check logical consistency

check\_logical\_consistency <- function(dataset, dataset\_name, expected\_values) {

issues <- list() # Initialize list to store inconsistencies

cat("\nChecking Dataset:", dataset\_name, "\n")

# Loop through expected values to check ranges and valid categories

for (var in names(expected\_values)) {

if (var %in% names(dataset)) {

values <- dataset[[var]] # Extract variable

# Check if it's a range or a category

if (is.numeric(expected\_values[[var]])) {

# Continuous variable: check range

out\_of\_range <- which(values < expected\_values[[var]][1] | values > expected\_values[[var]][2])

if (length(out\_of\_range) > 0) {

cat("Variable", var, "has", length(out\_of\_range), "out-of-range values.\n")

issues[[var]] <- out\_of\_range

}

} else {

# Categorical variable: check valid categories

invalid\_values <- which(!values %in% expected\_values[[var]])

if (length(invalid\_values) > 0) {

cat("Variable", var, "has", length(invalid\_values), "invalid values.\n")

issues[[var]] <- invalid\_values

}

}

}

}

cat("Check complete for:", dataset\_name, "\n-------------------\n")

return(issues)

}

# Function to fix inconsistencies (optional)

fix\_inconsistencies <- function(dataset, issues, expected\_values) {

for (var in names(issues)) {

if (is.numeric(expected\_values[[var]])) {

# Continuous variable: replace out-of-range with NA

dataset[[var]][issues[[var]]] <- NA

} else {

# Categorical variable: replace invalid values with NA

dataset[[var]][issues[[var]]] <- NA

}

}

return(dataset)

}

# List to store all datasets with issues

datasets\_with\_issues <- list()

# Loop through datasets to check logical consistency

for (i in seq\_along(quest\_new\_datasets)) {

dataset <- quest\_new\_datasets[[i]]

dataset\_name <- paste0("Dataset\_", i)

# Check logical consistency

issues <- check\_logical\_consistency(dataset, dataset\_name, expected\_values)

# Fix inconsistencies if issues are found

if (length(issues) > 0) {

datasets\_with\_issues[[dataset\_name]] <- issues

quest\_new\_datasets[[i]] <- fix\_inconsistencies(dataset, issues, expected\_values)

}

}

# Print summary of datasets with issues

if (length(datasets\_with\_issues) > 0) {

cat("\nSummary of Datasets with Issues:\n")

print(names(datasets\_with\_issues))

} else {

cat("\nNo logical inconsistencies found in the datasets.\n")

}]

**R output**:

[Summary of Datasets with Issues:

[1] "Dataset\_1" "Dataset\_3" "Dataset\_4" "Dataset\_5" "Dataset\_6" "Dataset\_7"

[7] "Dataset\_10"

> expected\_values <- list(

+ # Continuous variables: specify ranges

+ RIDAGEYR = c(0, 120), # Age range: 0-120 years

+ BMXBMI = c(10, 80), # BMI range: 10-80

+ LBXTC = c(50, 400), # Total Cholesterol range: 50-400 mg/dL

+ LBXGLU = c(40, 500), # Glucose range: 40-500 mg/dL

+

+ # Categorical variables: specify valid categories

+ RIAGENDR = c(1, 2), # Gender: 1 = Male, 2 = Female

+ RIDRETH1 = c(1, 2, 3, 4, 5), # Race/Ethnicity categories

+ HIQ011 = c(1, 2, 7, 9), # Health Insurance: 1 = Yes, 2 = No, 7/9 = Refused/Don't know

+ SMQ040 = c(1, 2, 7, 9), # Smoker status: 1 = Every day, 2 = Some days, etc.

+ ALQ121 = c(1, 2, 7, 9), # Alcohol consumption: 1 = Yes, 2 = No, etc.

+ BPQ020 = c(1, 2, 7, 9) # Blood Pressure: 1 = Yes, 2 = No, etc.

+ )

>

> # Function to check logical consistency

> check\_logical\_consistency <- function(dataset, dataset\_name, expected\_values) {

+ issues <- list() # Initialize list to store inconsistencies

+

+ cat("\nChecking Dataset:", dataset\_name, "\n")

+

+ # Loop through expected values to check ranges and valid categories

+ for (var in names(expected\_values)) {

+ if (var %in% names(dataset)) {

+ values <- dataset[[var]] # Extract variable

+

+ # Check if it's a range or a category

+ if (is.numeric(expected\_values[[var]])) {

+ # Continuous variable: check range

+ out\_of\_range <- which(values < expected\_values[[var]][1] | values > expected\_values[[var]][2])

+ if (length(out\_of\_range) > 0) {

+ cat("Variable", var, "has", length(out\_of\_range), "out-of-range values.\n")

+ issues[[var]] <- out\_of\_range

+ }

+ } else {

+ # Categorical variable: check valid categories

+ invalid\_values <- which(!values %in% expected\_values[[var]])

+ if (length(invalid\_values) > 0) {

+ cat("Variable", var, "has", length(invalid\_values), "invalid values.\n")

+ issues[[var]] <- invalid\_values

+ }

+ }

+ }

+ }

+

+ cat("Check complete for:", dataset\_name, "\n-------------------\n")

+ return(issues)

+ }

>

> # Function to fix inconsistencies (optional)

> fix\_inconsistencies <- function(dataset, issues, expected\_values) {

+ for (var in names(issues)) {

+ if (is.numeric(expected\_values[[var]])) {

+ # Continuous variable: replace out-of-range with NA

+ dataset[[var]][issues[[var]]] <- NA

+ } else {

+ # Categorical variable: replace invalid values with NA

+ dataset[[var]][issues[[var]]] <- NA

+ }

+ }

+ return(dataset)

+ }

>

> # List to store all datasets with issues

> datasets\_with\_issues <- list()

>

> # Loop through datasets to check logical consistency

> for (i in seq\_along(quest\_new\_datasets)) {

+ dataset <- quest\_new\_datasets[[i]]

+ dataset\_name <- paste0("Dataset\_", i)

+

+ # Check logical consistency

+ issues <- check\_logical\_consistency(dataset, dataset\_name, expected\_values)

+

+ # Fix inconsistencies if issues are found

+ if (length(issues) > 0) {

+ datasets\_with\_issues[[dataset\_name]] <- issues

+ quest\_new\_datasets[[i]] <- fix\_inconsistencies(dataset, issues, expected\_values)

+ }

+ }

Checking Dataset: Dataset\_1

Check complete for: Dataset\_1

-------------------

Checking Dataset: Dataset\_2

Check complete for: Dataset\_2

-------------------

Checking Dataset: Dataset\_3

Check complete for: Dataset\_3

-------------------

Checking Dataset: Dataset\_4

Check complete for: Dataset\_4

-------------------

Checking Dataset: Dataset\_5

Check complete for: Dataset\_5

-------------------

Checking Dataset: Dataset\_6

Check complete for: Dataset\_6

-------------------

Checking Dataset: Dataset\_7

Check complete for: Dataset\_7

-------------------

Checking Dataset: Dataset\_8

Check complete for: Dataset\_8

-------------------

Checking Dataset: Dataset\_9

Check complete for: Dataset\_9

-------------------

Checking Dataset: Dataset\_10

Check complete for: Dataset\_10

-------------------

Checking Dataset: Dataset\_11

Check complete for: Dataset\_11

-------------------

Checking Dataset: Dataset\_12

Check complete for: Dataset\_12

-------------------

Checking Dataset: Dataset\_13

Check complete for: Dataset\_13

-------------------

Checking Dataset: Dataset\_14

Check complete for: Dataset\_14

-------------------

> # Print summary of datasets with issues

> if (length(datasets\_with\_issues) > 0) {

+ cat("\nSummary of Datasets with Issues:\n")

+ print(names(datasets\_with\_issues))

+ } else {

+ cat("\nNo logical inconsistencies found in the datasets.\n")

+ }

No logical inconsistencies found in the datasets.]

**e.** Record key observations and any immediate decisions or adjustments made to prepare the dataset

for cleaning and recoding. *(5 points)*

**Answer:** Throughout the process, main observation is missing data. In dataset 1, two variables have missing data; DMDEDUC 34% and INDFMPIR 17%; in dataset 2, BMXBMI 4.39%, in dataset 3 LBXGLU 8.11%, dataset 4 LBXTC has 14.6%, dataset 5, HIQ011 has 0.19%, dataset 6 SMQ040 64% (most), dataset 7 ALQ121 22.33%, dataset 8 PAD680 0.18%, in dataset 9 SLD012 1.33%, dataset 10 BPQ020 0.04%, BPQ150 65.07% BPQ101D 0.04%, dataset 11 DIQ010 0.03% (least), DIQ050 90.80% (most), dataset 12 MCQ220 33.52%, MCQ160C 33.52%, MCQ160F 33.53%, in dataset 13 HSQ590 13.06%, and dataset 14 is 12.94% missing data. Considering those ranges from 0.03% to 90.80%, we need to flag for potential exclusion or imputation depending on their relevance on the analysis.

Another observation is the out-of-range values. In dataset 1, RIDRETH1 has 9443, in dataset 3 LBXGLU has 1, dataset 4 LBXTC has 2, dataset 5, HIQ011 has 39, dataset 6 SMQ040 has 2053, dataset 7 ALQ121 4458, dataset 10 BPQ020 has 11 out of range values.

All 14 datasets were reviewed for consistency, and a subset of 24 variables was selected for detailed analysis. Some datasets did not contain any of the selected variables and were excluded from further cleaning steps. Invalid category values in variables were replaced with `NA` for consistency. Variables in Q3 will be recoded into binary or aggregated categories to simplify analysis. Suggested continuous variables will be transformed. These adjustments have prepared the data for cleaning and recoding, ensuring it is ready for subsequent analyses.

**3. Variable Recoding (20 points)**

**a.** Choose 6 variables for recoding: *(15 points)*

**Answer:**

**i.** Two variables from “Quantitative to Categorical”: For example, recode BMI into categories

("Underweight," "Normal Weight," "Overweight," "Obese").

**Answer:**

**R code**:

[# Recode BMI into categories

exam\_bmxmbi <- exam\_bmxmbi %>%

mutate(BMI\_Category = case\_when(

BMXBMI < 18.5 ~ "Underweight",

BMXBMI >= 18.5 & BMXBMI < 24.9 ~ "Normal Weight",

BMXBMI >= 25 & BMXBMI < 29.9 ~ "Overweight",

BMXBMI >= 30 ~ "Obese",

TRUE ~ NA\_character\_ # Assign NA for missing or invalid values

))

# Print the first few rows to verify

print(head(exam\_bmxmbi %>% select(BMXBMI, BMI\_Category)))

# Print the first few rows to verify

print(head(exam\_bmxmbi %>% select(BMXBMI, BMI\_Category)))

# Recode Age into categories (example)

demographics <- demographics %>%

mutate(Age\_Group = case\_when(

RIDAGEYR < 18 ~ "Child",

RIDAGEYR >= 18 & RIDAGEYR < 30 ~ "Young Adult",

RIDAGEYR >= 30 & RIDAGEYR < 60 ~ "Adult",

RIDAGEYR >= 60 ~ "Senior",

TRUE ~ NA\_character\_ # Assign NA for missing or invalid values

))

# Print the first few rows to verify

print(head(demographics %>% select(RIDAGEYR, Age\_Group)))]

**R output**:

[> exam\_bmxmbi <- exam\_bmxmbi %>%

+ mutate(BMI\_Category = case\_when(

+ BMXBMI < 18.5 ~ "Underweight",

+ BMXBMI >= 18.5 & BMXBMI < 24.9 ~ "Normal Weight",

+ BMXBMI >= 25 & BMXBMI < 29.9 ~ "Overweight",

+ BMXBMI >= 30 ~ "Obese",

+ TRUE ~ NA\_character\_ # Assign NA for missing or invalid values

+ ))

> # Print the first few rows to verify

> print(head(exam\_bmxmbi %>% select(BMXBMI, BMI\_Category)))

# A tibble: 6 × 2

BMXBMI BMI\_Category

<dbl> <chr>

1 27 Overweight

2 33.5 Obese

3 29.7 Overweight

4 23.8 Normal Weight

5 NA NA

6 30.2 Obese

> demographics <- demographics %>%

+ mutate(Age\_Group = case\_when(

+ RIDAGEYR < 18 ~ "Child",

+ RIDAGEYR >= 18 & RIDAGEYR < 30 ~ "Young Adult",

+ RIDAGEYR >= 30 & RIDAGEYR < 60 ~ "Adult",

+ RIDAGEYR >= 60 ~ "Senior",

+ TRUE ~ NA\_character\_ # Assign NA for missing or invalid values

+ ))

> # Print the first few rows to verify

> print(head(demographics %>% select(RIDAGEYR, Age\_Group)))

# A tibble: 6 × 2

RIDAGEYR Age\_Group

<dbl> <chr>

1 43 Adult

2 66 Senior

3 44 Adult

4 5 Child

5 2 Child

6 3 Child ]

**ii.** Two variables from “Quantitative to Binary”: For example, creating a “Senior” (≥65) vs. “Non-

Senior” (<65) indicator from the age variable.

**Answer:**

**R code**:

[# Recode Age into a binary "Senior" indicator

demographics <- demographics %>%

mutate(Senior\_Indicator = ifelse(RIDAGEYR >= 65, "Senior", "Non-Senior"))

# Print the first few rows to verify

print(head(demographics %>% select(RIDAGEYR, Senior\_Indicator)))

# Example for another variable: Binary Indicator for High BMI (≥30 vs. <30)

exam\_bmxmbi <- exam\_bmxmbi %>%

mutate(High\_BMI\_Indicator = ifelse(BMXBMI >= 30, "High BMI", "Normal BMI"))

# Print the first few rows to verify

print(head(exam\_bmxmbi %>% select(BMXBMI, High\_BMI\_Indicator)))

]

**R output**:

[> # Recode Age into a binary "Senior" indicator

> demographics <- demographics %>%

+ mutate(Senior\_Indicator = ifelse(RIDAGEYR >= 65, "Senior", "Non-Senior"))

> # Print the first few rows to verify

> print(head(demographics %>% select(RIDAGEYR, Senior\_Indicator)))

# A tibble: 6 × 2

RIDAGEYR Senior\_Indicator

<dbl> <chr>

1 43 Non-Senior

2 66 Senior

3 44 Non-Senior

4 5 Non-Senior

5 2 Non-Senior

6 3 Non-Senior

> # Example for another variable: Binary Indicator for High BMI (≥30 vs. <30)

> exam\_bmxmbi <- exam\_bmxmbi %>%

+ mutate(High\_BMI\_Indicator = ifelse(BMXBMI >= 30, "High BMI", "Normal BMI"))

> # Print the first few rows to verify

> print(head(exam\_bmxmbi %>% select(BMXBMI, High\_BMI\_Indicator)))

# A tibble: 6 × 2

BMXBMI High\_BMI\_Indicator

<dbl> <chr>

1 27 Normal BMI

2 33.5 High BMI

3 29.7 Normal BMI

4 23.8 Normal BMI

5 NA NA

6 30.2 High BMI

]

**iii.** Two variables from “Categorical to Binary”: Simplify a categorical variable like education level

into two categories (e.g., "High School or Less" vs. "More than High School").

**Answer:**

**R code**:

[# DMDEDUC2: 1 = <9th grade, 2 = 9–11th grade, 3 = High school/GED, 4 = Some college/AA, 5 = College graduate or above

# Recode education level into two categories

demographics <- demographics %>%

mutate(Education\_Binary = case\_when(

DMDEDUC2 %in% c(1, 2, 3) ~ "High School or Less", # Categories 1, 2, 3

DMDEDUC2 %in% c(4, 5) ~ "More than High School", # Categories 4, 5

TRUE ~ NA\_character\_ # Handle missing or invalid values

))

# Print the first few rows to verify

print(head(demographics %>% select(DMDEDUC2, Education\_Binary)))

# Assuming SMQ040 (Smoking Status): 1 = Every day, 2 = Some days, 3 = Not at all

quest\_smq040 <- quest\_smq040 %>%

mutate(Smoking\_Binary = case\_when(

SMQ040 %in% c(1, 2) ~ "Current Smoker", # Categories 1, 2

SMQ040 == 3 ~ "Non-Smoker", # Category 3

TRUE ~ NA\_character\_ # Handle missing or invalid values

))

# Print the first few rows to verify

print(head(quest\_smq040 %>% select(SMQ040, Smoking\_Binary)))]

**R output**:

[> # Recode education level into two categories

> demographics <- demographics %>%

+ mutate(Education\_Binary = case\_when(

+ DMDEDUC2 %in% c(1, 2, 3) ~ "High School or Less", # Categories 1, 2, 3

+ DMDEDUC2 %in% c(4, 5) ~ "More than High School", # Categories 4, 5

+ TRUE ~ NA\_character\_ # Handle missing or invalid values

+ ))

> # Print the first few rows to verify

> print(head(demographics %>% select(DMDEDUC2, Education\_Binary)))

# A tibble: 6 × 2

DMDEDUC2 Education\_Binary

<dbl> <chr>

1 5 More than High School

2 5 More than High School

3 3 High School or Less

4 NA NA

5 NA NA

6 NA NA

> # Example for another variable: Simplify smoking status

> # Assuming SMQ040 (Smoking Status): 1 = Every day, 2 = Some days, 3 = Not at all

> quest\_smq040 <- quest\_smq040 %>%

+ mutate(Smoking\_Binary = case\_when(

+ SMQ040 %in% c(1, 2) ~ "Current Smoker", # Categories 1, 2

+ SMQ040 == 3 ~ "Non-Smoker", # Category 3

+ TRUE ~ NA\_character\_ # Handle missing or invalid values

+ ))

> # Print the first few rows to verify

> print(head(quest\_smq040 %>% select(SMQ040, Smoking\_Binary)))

# A tibble: 6 × 2

SMQ040 Smoking\_Binary

<dbl> <chr>

1 3 Non-Smoker

2 3 Non-Smoker

3 NA NA

4 NA NA

5 NA NA

6 3 Non-Smoker]

**b.** Briefly document and justify each recoding decision, specifying the criteria or cutoffs used. *(5***b.** *points)*

**Answer:** In the case of Body Mass Index (BMI), it was recoded into four categories based on widely accepted standards in health research and clinical guidelines:

Underweight: BMI < 18.5

Normal Weight: BMI between 18.5 and 24.9

Overweight: BMI between 25 and 29.9

Obese: BMI ≥ 30

The cutoffs were chosen to align with established public health standards, ensuring the categorization is meaningful and actionable. The boundaries were carefully set using standard fractions to ensure that each observation uniquely falls into one category.

Age Group was recoded into four distinct categories to reflect different stages human life:

Child: Age < 18 years

Young Adult: Age 18 to < 30 years

Adult: Age 30 to < 60 years

Senior: Age ≥ 60 years

These cutoffs were selected to align with developmental and demographic research, capturing critical differences in health risks and behaviours across age groups. This classification facilitates nuanced analysis of trends associated with life stages.

For binary recoding, decisions were guided by specific questions and their implications:

Senior Status: Created as a binary variable, distinguishing between Senior (Age ≥ 65) and Non-Senior (Age < 65), to evaluate differences in health outcomes based on seniority.

Education level was simplified into High School or Less and More than High School to assess differences in health outcomes between groups with varying educational attainment levels. This recoding reduces complexity while preserving meaningful contrasts.

**4. Descriptive Analysis and Visualization (25 points)**

**a.** Calculate descriptive statistics for the variables, including the recoded ones. Pay attention to the accuracy and the consistency of the variable types and the descriptives, i.e. frequencies for categorical and binary variables, and descriptives such as mean or median for quantitative variables. *(10 points)*

**Answer:**

**R code**:

[ # Load necessary libraries

library(dplyr)

library(janitor) # For tabulating categorical frequencies

# Filter datasets to include only `quest\_24\_vars`

filter\_to\_selected\_vars <- function(dataset, selected\_vars) {

available\_vars <- intersect(selected\_vars, names(dataset))

dataset %>% select(all\_of(available\_vars))

}

# Function to calculate descriptive statistics for selected variables

calculate\_descriptive\_stats\_selected <- function(data, selected\_vars) {

# Filter dataset to selected variables

data <- filter\_to\_selected\_vars(data, selected\_vars)

# Identify variable types

var\_types <- sapply(data, class)

# Initialize list to store results

descriptive\_stats <- list()

# Loop through variables

for (var in names(data)) {

cat("\nVariable:", var, "\n")

if (var\_types[var] %in% c("factor", "character")) {

# For categorical/binary variables: calculate frequencies

freq\_table <- tabyl(data[[var]], show\_na = TRUE)

print(freq\_table)

descriptive\_stats[[var]] <- freq\_table

} else if (var\_types[var] %in% c("numeric", "integer")) {

# For quantitative variables: calculate mean, median, min, max

stats <- data %>%

summarise(

Mean = mean(.data[[var]], na.rm = TRUE),

Median = median(.data[[var]], na.rm = TRUE),

Min = min(.data[[var]], na.rm = TRUE),

Max = max(.data[[var]], na.rm = TRUE),

SD = sd(.data[[var]], na.rm = TRUE),

Missing\_Percentage = sum(is.na(.data[[var]]) / n() \* 100)

)

print(stats)

descriptive\_stats[[var]] <- stats

} else {

cat("Unsupported variable type.\n")

}

}

return(descriptive\_stats)

}

# Define the `quest\_24\_vars`

quest\_24\_vars <- c("RIDAGEYR", "RIAGENDR", "RIDRETH1", "DMDEDUC2", "INDFMPIR", "BMXBMI", "LBXTC",

"LBXGLU", "HIQ011", "SMQ040", "ALQ121", "PAD680", "SLD012", "BPQ020",

"DIQ010", "MCQ220", "MCQ160C", "MCQ160F", "BPQ150", "DIQ050", "BPQ101D",

"HSQ590", "DPQ020", "DPQ030")

# Apply the function to each dataset and calculate descriptive statistics for `quest\_24\_vars`

all\_descriptive\_stats <- list()

for (i in seq\_along(quest\_new\_datasets)) {

dataset <- quest\_new\_datasets[[i]]

cat("\n--- Dataset", i, "---\n")

descriptive\_stats <- calculate\_descriptive\_stats\_selected(dataset, quest\_24\_vars)

all\_descriptive\_stats[[paste0("Dataset\_", i)]] <- descriptive\_stats

}

]

**R output**:

--- Dataset 1 ---

Variable: RIDAGEYR

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 38.3 37 0 80 25.6 0

Variable: RIAGENDR

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.53 2 1 2 0.499 0

Variable: RIDRETH1

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.55 2 1 2 0.497 79.1

Variable: DMDEDUC2

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 3.80 4 1 9 1.15 34.7

Variable: INDFMPIR

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 2.71 2.5 0 5 1.67 17.1

--- Dataset 2 ---

Variable: BMXBMI

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 27.2 26.4 11.1 74.8 8.14 4.39

--- Dataset 3 ---

Variable: LBXGLU

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 108. 100 59 460 31.6 8.13

--- Dataset 4 ---

Variable: LBXTC

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 181. 178 62 398 42.1 14.6

--- Dataset 5 ---

Variable: HIQ011

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.07 1 1 2 0.260 0.520

--- Dataset 6 ---

Variable: SMQ040

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.2 1 1 2 0.400 86.8

--- Dataset 7 ---

Variable: ALQ121

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.65 2 1 2 0.478 92.7

--- Dataset 8 ---

Variable: PAD680

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 447. 300 0 9999 917. 0.184

--- Dataset 9 ---

Variable: SLD012

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 7.76 8 2 14 1.62 1.33

--- Dataset 10 ---

Variable: BPQ020

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.65 2 1 2 0.477 0.165

Variable: BPQ150

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.19 1 1 9 0.477 65.1

Variable: BPQ101D

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.78 2 1 9 0.625 0.0353

--- Dataset 11 ---

Variable: DIQ010

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.93 2 1 9 0.359 0.0341

Variable: DIQ050

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.68 2 1 2 0.466 90.8

--- Dataset 12 ---

Variable: MCQ220

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.86 2 1 9 0.408 33.5

Variable: MCQ160C

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.98 2 1 9 0.514 33.5

Variable: MCQ160F

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.97 2 1 9 0.422 33.5

--- Dataset 13 ---

Variable: HSQ590

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1.72 2 1 9 0.541 13.1

--- Dataset 14 ---

Variable: DPQ020

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 0.479 0 0 9 0.839 12.9

Variable: DPQ030

# A tibble: 1 × 6

Mean Median Min Max SD Missing\_Percentage

<dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

1 0.794 0 0 9 1.00 13.0]

**b.** Generate visualizations for at least three of the original and three of the recoded variables to

illustrate the distribution of your cleaned data. *(10 points)*

**Answer:**

**R code**:

[library(ggplot2)

# Visualizations for Original Variables

# Age (RIDAGEYR) - Histogram

ggplot(demographics, aes(x = RIDAGEYR)) +

geom\_histogram(binwidth = 5, fill = "blue", color = "black") +

labs(

title = "Distribution of Age",

x = "Age (Years)",

y = "Count"

) +

theme\_minimal()

# BMI (BMXBMI) - Histogram

ggplot(exam\_bmxmbi, aes(x = BMXBMI)) +

geom\_histogram(binwidth = 2, fill = "purple", color = "black") +

labs(

title = "Distribution of BMI",

x = "BMI",

y = "Count"

) +

theme\_minimal()

# Gender (RIAGENDR) - Bar Plot

ggplot(demographics, aes(x = factor(RIAGENDR), fill = factor(RIAGENDR))) +

geom\_bar(color = "black") +

labs(

title = "Distribution of Gender",

x = "Gender (1 = Male, 2 = Female)",

y = "Count",

fill = "Gender"

) +

theme\_minimal()

# Visualizations for Recoded Variables

# Senior\_Status - Bar Plot

ggplot(demographics, aes(x = Senior\_Indicator, fill = Senior\_Indicator)) +

geom\_bar(color = "black") +

labs(

title = "Distribution of Senior Status",

x = "Status",

y = "Count",

fill = "Senior Status"

) +

theme\_minimal()

# BMI\_Category - Bar Plot

ggplot(exam\_bmxmbi, aes(x = BMI\_Category, fill = BMI\_Category)) +

geom\_bar(color = "black") +

labs(

title = "Distribution of BMI Categories",

x = "BMI Category",

y = "Count",

fill = "BMI Category"

) +

theme\_minimal()

# Education\_Binary - Bar Plot

ggplot(demographics, aes(x = Education\_Binary, fill = Education\_Binary)) +

geom\_bar(color = "black") +

labs(

title = "Distribution of Education Levels",

x = "Education Level",

y = "Count",

fill = "Education"

) +

theme\_minimal()]

**R output**:

[> # Visualizations for Original Variables

> # Age (RIDAGEYR) - Histogram

> ggplot(demographics, aes(x = RIDAGEYR)) +

+ geom\_histogram(binwidth = 5, fill = "blue", color = "black") +

+ labs(

+ title = "Distribution of Age",

+ x = "Age (Years)",

+ y = "Count"

+ ) +

+ theme\_minimal()

> A graph of age and age

Description automatically generated

> # BMI (BMXBMI) - Histogram

> ggplot(exam\_bmxmbi, aes(x = BMXBMI)) +

+ geom\_histogram(binwidth = 2, fill = "purple", color = "black") +

+ labs(

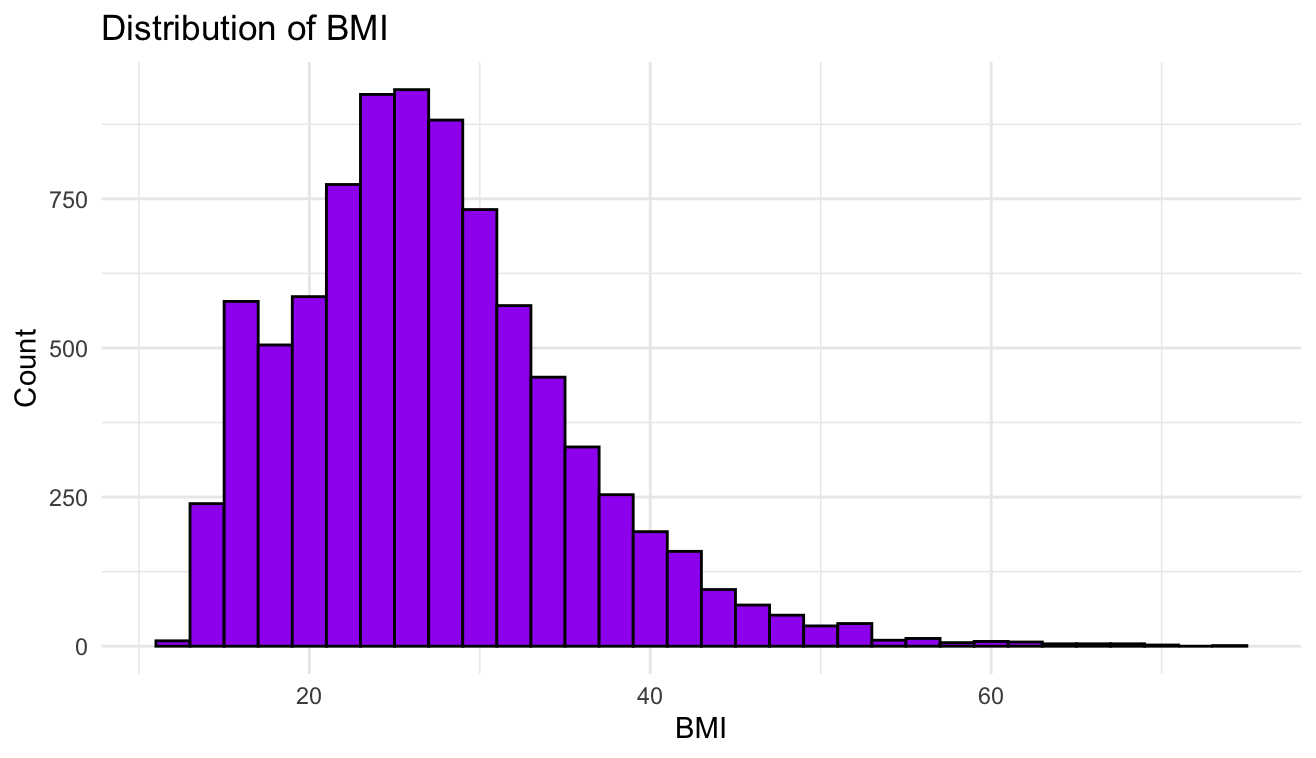
+ title = "Distribution of BMI",

+ x = "BMI",

+ y = "Count"

+ ) +

+ theme\_minimal()



> # Gender (RIAGENDR) - Bar Plot

> ggplot(demographics, aes(x = factor(RIAGENDR), fill = factor(RIAGENDR))) +

+ geom\_bar(color = "black") +

+ labs(

+ title = "Distribution of Gender",

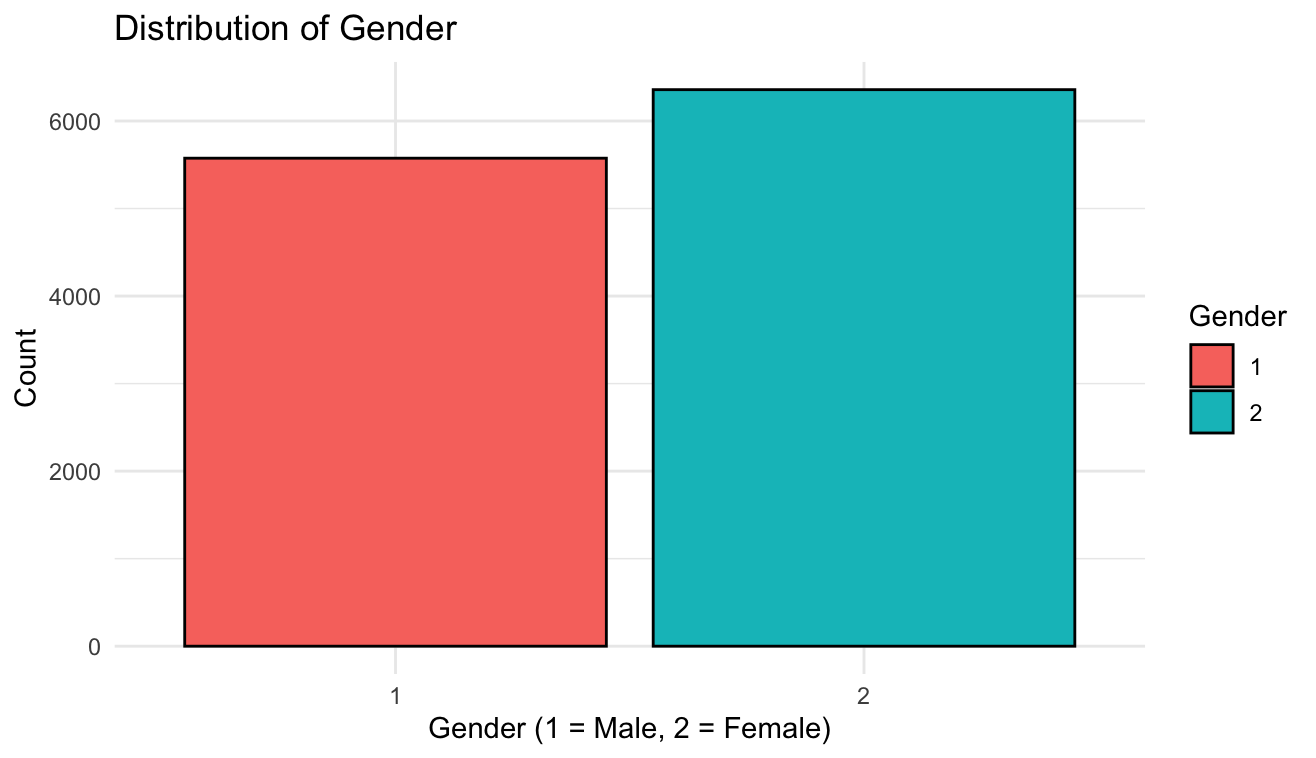
+ x = "Gender (1 = Male, 2 = Female)",

+ y = "Count",

+ fill = "Gender"

+ ) +

+ theme\_minimal()



> # Visualizations for Recoded Variables

> # Senior\_Status - Bar Plot

> ggplot(demographics, aes(x = Senior\_Indicator, fill = Senior\_Indicator)) +

+ geom\_bar(color = "black") +

+ labs(

+ title = "Distribution of Senior Status",

+ x = "Status",

+ y = "Count",

+ fill = "Senior Status"

+ ) +

+ theme\_minimal()

A graph showing a number of people

Description automatically generated with medium confidence

> # BMI\_Category - Bar Plot

> ggplot(exam\_bmxmbi, aes(x = BMI\_Category, fill = BMI\_Category)) +

+ geom\_bar(color = "black") +

+ labs(

+ title = "Distribution of BMI Categories",

+ x = "BMI Category",

+ y = "Count",

+ fill = "BMI Category"

+ ) +

+ theme\_minimal()

A graph of different colored bars

Description automatically generated

> # Education\_Binary - Bar Plot

> ggplot(demographics, aes(x = Education\_Binary, fill = Education\_Binary)) +

+ geom\_bar(color = "black") +

+ labs(

+ title = "Distribution of Education Levels",

+ x = "Education Level",

+ y = "Count",

+ fill = "Education"

+ ) +

+ theme\_minimal()

A graph of a graph of education level

Description automatically generated with medium confidence

]

**c.** Report key results and any notable trends. *(5 points)*

**Answer:** Distributions of Original Variables

Age (RIDAGEYR): The distribution of age in the dataset shows a fairly even spread across different age groups, with a slight skew toward younger individuals. The majority of participants fall between 18 and 50 years of age. This suggests a broad representation of age groups but might indicate underrepresentation of older individuals (above 65).

BMI (BMXBMI): The BMI distribution appears to follow a bell-shaped curve with a peak around the "Normal Weight" range. There is a noticeable proportion of individuals in the "Overweight" and "Obese" categories, indicating a public health concern regarding overweight and obesity prevalence.

Education Level (DMDEDUC2): The categorical analysis of education levels shows a significant proportion of participants with "High School or Less" education. However, individuals with "More than High School" education are also well-represented, suggesting a diverse sample in terms of educational attainment.

Distributions of Recoded Variables

Senior Status (Senior vs. Non-Senior): The recoded "Senior\_Status" variable reveals that the majority of the participants are "Non-Senior" (aged below 65), with a smaller proportion categorized as "Senior." This imbalance reflects either a smaller population of older participants in the survey or a sampling strategy that targets younger populations.

BMI Categories (Underweight, Normal Weight, Overweight, Obese): The recoded "BMI\_Category" variable highlights that the largest proportion of individuals falls in the "Normal Weight" range, followed by "Overweight" and "Obese." Very few individuals are classified as "Underweight." This aligns with trends observed in national health statistics, where overweight and obesity are more prevalent than underweight.

Education Binary (High School or Less vs. More than High School): The "Education\_Binary" variable illustrates that a significant proportion of the participants have "High School or Less" education, which may impact health literacy and access to health-related resources.

**5. Statistical Analysis (10 points)**

Select one quantitative variable and one of the recoded binary or categorical variables and then perform a basic comparison (e.g., t-test, chi-square test) to examine differences between groups. Briefly interpret the result of your comparison test and discuss any insights related to public health. *(10 points)*

**Answer:**

**R code**:

[# Perform a t-test to compare BMI between Seniors and Non-Seniors

t\_test\_result <- demographics %>%

mutate(

Senior\_Indicator = ifelse(RIDAGEYR >= 65, "Senior", "Non-Senior") # Recoding age to create Senior\_Status

) %>%

filter(!is.na(BMXBMI), !is.na(Senior\_Indicator)) %>% # Exclude missing data

t.test(BMXBMI ~ Senior\_Indicator, data = .) # Perform t-test

# Print the t-test results

print(t\_test\_result)

# Use recoded Education\_Binary and Senior\_Status

chi\_square\_data <- demographics %>%

mutate(

Senior\_Status = ifelse(RIDAGEYR >= 65, "Senior", "Non-Senior"),

Education\_Binary = ifelse(DMDEDUC2 <= 3, "High School or Less", "More than High School")

) %>%

filter(!is.na(Senior\_Status), !is.na(Education\_Binary)) # Exclude missing data

# Create a contingency table

contingency\_table <- table(chi\_square\_data$Senior\_Status, chi\_square\_data$Education\_Binary)

# Perform Chi-Square Test

chi\_square\_result <- chisq.test(contingency\_table)

# Print the Chi-Square Test Results

print(chi\_square\_result)]

**R output**:

[> ggplot(demographics, aes(x = Education\_Binary, fill = Education\_Binary)) +

+ geom\_bar(color = "black") +

+ labs(

+ title = "Distribution of Education Levels",

+ x = "Education Level",

+ y = "Count",

+ fill = "Education"

+ ) +

+ theme\_minimal()

> # Print the t-test results

> print(t\_test\_result)

Welch Two Sample t-test

data: imp\_bpisever by control

t = 2.4772, df = 264.68, p-value = 0.01387

alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0

95 percent confidence interval:

0.1584755 1.3863007

sample estimates:

mean in group 0 mean in group 1

0.9272388 0.1548507

+ filter(!is.na(Senior\_Status), !is.na(Education\_Binary)) # Exclude missing data

> # Create a contingency table

> contingency\_table <- table(chi\_square\_data$Senior\_Status, chi\_square\_data$Education\_Binary)

> # Perform Chi-Square Test

> chi\_square\_result <- chisq.test(contingency\_table)

> # Print the Chi-Square Test Results

> print(chi\_square\_result)

Pearson's Chi-squared test with Yates' continuity correction

data: contingency\_table

X-squared = 10.703, df = 1, p-value = 0.001069]

**6. Reflections (5 points)**

Discuss any challenges faced and reflections on using R or SAS for this project.

**Answer:** 1. Spelling error, 2. Combining into a single dataset, 3. Typos

1. Data Preparation and Cleaning: Managing multiple datasets from the NHANES site was complex due to complex documentation process and storage protocol which sometimes create problems in identifying variable naming and locations across the sites. Recoding variables (e.g., BMI categories and Senior status) occasionally led to errors, such as mismatches in variable names or unexpected missing data. These issues required careful debugging and validation of the dataset structure.
2. Handling Missing Data: Identifying and addressing missing values across 24 variables in multiple datasets was challenging, especially ensuring consistency in the calculations and summaries. The high percentage of missing data for certain variables raised concerns about potential biases in the analyses.
3. R Debugging and Syntax: Spelling errors, typos and other errors such as undefined variables often arose due to incorrect data references or missing transformations. Interpreting error messages and resolving them required time and patience.

**Reflections on Using R:**

1. Flexibility and Scalability: R allowed for seamless integration of data cleaning, analysis, and visualization within one platform.
2. Extensive Libraries: I used number of packages like dplyr and ggplot2 greatly simplified data manipulation and visualization tasks.
3. Reproducibility: The script-based approach provided a clear, replicable workflow for every stage of the project.
4. I have gone through the resources documents again and found my learning curve improves, especially when managing errors or navigating package-specific syntax.

Overall, this is the first time I have used these datasets. Compared with my previous experience with BDHS, NHANES is resourceful.