Takreem Virk
investigating Obesity and Type II Diabetes Using NHANES
May 5, 2025

1 Introduction

1.1 Motivation and Significance

Obesity and type 2 diabetes are two of the most pressing public health challenges in the United States, contributing to long-term health complications and increased healthcare costs. Despite ongoing efforts in prevention and treatment, these conditions continue to rise at an alarming rate, underscoring the need for improved strategies in early identification, intervention, and policy-making.

Why This Study is Important

This research aims to leverage the vast NHANES dataset to uncover patterns that can enhance disease prevention efforts. Specifically:

- Advancing Predictive Healthcare: By applying machine learning techniques to NHANES data, this study seeks to improve the accuracy of type 2 diabetes risk prediction. Early identification of high-risk individuals can lead to timely interventions, ultimately reducing the burden of diabetes-related complications.
- Public Health Policy Impact: Identifying obesity trends across different demographic and socioeconomic groups provides crucial insight for developing targeted health policies. Understanding disparities in risk factors can help shape more effective prevention programs.
- Data-Driven Decision Making: Traditional statistical approaches have provided valuable insights into obesity and diabetes risk, but integrating machine learning adds another layer of predictive power. By harnessing advanced techniques, this study can contribute to a more nuanced understanding of health determinants.

Implications for Public Health and Society

The findings of this study hold broad implications:

- Improved Clinical Screening Tools: Machine learning-driven risk models could complement traditional medical screenings, allowing for personalized healthcare approaches.
- Enhanced Preventative Strategies: Policymakers and public health officials can use these insights to refine obesity prevention campaigns, particularly for vulnerable populations.
- Scientific Contribution: This study builds on existing NHANES research by integrating novel analytical methods, potentially setting the stage for future investigations into chronic disease prevention.

This research offers a unique opportunity to drive meaningful advancements in health science, prevention strategies, and policy formation.

1.2 Survey Background

The National Health and Nutrition Examination Survey (NHANES) is a long-standing health research program conducted by the Centers for Disease Control and Prevention (CDC) through the National Center for Health Statistics (NCHS). NHANES has been collecting health and

nutritional data on the U.S. population since the early 1960s, providing invaluable insights into disease trends, dietary habits, and public health outcomes.

Unlike traditional surveys, NHANES integrates self-reported data with objective medical examinations, laboratory tests, and interviews. Participants undergo both in-home surveys and clinical assessments in Mobile Examination Centers (MECs), ensuring high-quality, standardized data collection. The survey includes biological samples (e.g., blood and urine), physical measurements (e.g., height, weight, and blood pressure), and lifestyle factors such as diet and physical activity.

Who, When, Where, and What

- Who: NHANES targets a representative sample of the U.S. population, including individuals of all ages and diverse racial and socioeconomic backgrounds. The survey is designed to capture health disparities, ensuring the inclusion of underrepresented populations.
- When: The NHANES program is continuous, releasing new datasets every two years. This rolling survey model allows researchers to track long-term health trends.
- Where: Data collection occurs nationwide, with specially equipped Mobile Examination Centers (MECs) traveling to different regions to ensure broad representation.
- What: The survey covers a wide range of health topics, including chronic diseases (diabetes, hypertension, obesity), nutritional intake, mental health, physical activity, and environmental exposures. NHANES is a cornerstone of U.S. health research, used by policymakers, healthcare providers, and academics.

Complex Sampling Design

NHANES employs a multistage probability sampling design to ensure national representation. The sampling process consists of:

1. Stratification: The U.S. is divided into geographic segments to include diverse populations.
2. Cluster Selection: Census regions are selected using probability methods.
3. Oversampling of Key Groups: Specific demographic groups (e.g., minorities, older adults) are oversampled to improve statistical reliability in subgroup analyses.

Because of its complex sample design, NHANES requires specialized statistical techniques for proper analysis. Weighting factors and design variables are essential to ensure unbiased estimates, particularly when generalizing results to the U.S. population.

Accessing NHANES Data

NHANES datasets are freely available and public-access to everyone via the CDC website.

2 Research Questions

1. **Predicting Type 2 Diabetes Risk Using Machine Learning**: Given the increasing burden of diabetes, early identification of individuals at risk is critical. Traditional statistical methods have been used to analyze diabetes risk factors, but advancements in machine learning offer opportunities to develop more accurate predictive models. By leveraging NHANES data, this study aims to implement machine learning techniques to identify individuals at high risk for type 2 diabetes, enabling earlier intervention and personalized healthcare strategies.
2. **Understanding Obesity Trends Across Demographics**: Obesity is influenced by numerous factors, including socioeconomic status, physical activity, and dietary habits. NHANES data allows for a comprehensive examination of these variables across different populations, helping to highlight disparities and inform targeted policy measures. By investigating obesity trends, this study aims to provide insights that can guide preventative health strategies.

3 Model and Method

3.1 Research Question 1: Predicting Type 2 Diabetes Risk Using Machine Learning

Machine learning techniques were applied to NHANES demographic and examination datasets to predict type 2 diabetes risk. The methodology includes:

Data Selection & Preprocessing

- Datasets Used: NHANES demographic (age, gender, race/ethnicity, socioeconomic status), examination (BMI, blood pressure), and lab measures (glucose levels).
- Handling Sample Weights: NHANES employs complex survey sampling; therefore, each observation is weighted appropriately. Instead of using naive sample counts, the sum of weights per category is calculated for all statistical outputs, including confusion matrices.
- Feature Engineering: Transformation of categorical variables, normalization of numerical predictors, and missing data imputation based on NHANES guidelines.

Model Selection & Implementation

- Models evaluated: Logistic Regression, Random Forest, and Gradient Boosting Trees.
- Performance metrics: Accuracy, Precision, Recall, and Weighted Confusion Matrices (adjusted for survey weighting).
- Confusion matrices generated using weighted frequencies rather than raw counts, ensuring results align with NHANES analytic standards.

Justification for ML Approach

- Traditional statistical methods provide linear insights, whereas machine learning incorporates non-linear interactions between health indicators.
- Ensemble models like Random Forest and Gradient Boosting enhance predictive accuracy over simple regression models.

3.2 Research Question 2: Investigating Obesity Trends Across Demographics

The second research question examines how demographics, socioeconomic factors, and behavioral variables influence obesity prevalence.

Data Selection & Preprocessing

- Datasets Used: NHANES demographic and examination modules, which include variables such as age, gender, race/ethnicity, and physical measurements like BMI and waist circumference.
- Sample Weights: Survey weights (WTMEC2YR) were applied to all summary statistics and visualizations to ensure estimates reflect the U.S. population.
- Data Cleaning: The analysis excluded extreme or top-coded values and retained only observations with complete information on key variables.

Exploratory Analysis Approach

- Stratification: Participants were grouped by age category (18–29, 30–44, 45–59, 60+), gender, and race/ethnicity for comparison.
- Visualizations: Weighted boxplots were used to display BMI distributions across demographic subgroups.
- Statistical Focus: Emphasis was placed on medians and interquartile ranges to illustrate central tendencies and variability in BMI.

Rationale for Analytical Approach

- Obesity is influenced by multiple demographic and socioeconomic factors; descriptive visualizations offer an interpretable way to surface disparities.
- NHANES's complex sampling design allows for robust subgroup comparisons when weights are applied.
- This exploratory approach highlights high-risk groups and informs targeted public health strategies.

5 Analysis

5.1 Exploratory Data Analysis

To understand the characteristics of the population used in this study, we conducted a series of weighted exploratory analyses using the NHANES 2021–2023 dataset. Sample weights were applied to all visualizations to ensure that the distributions reflect the U.S. population. The graphics are divided into three categories: demographics (Figure 1), physical examination measures (Figure 2), and physical activity patterns (Figure 3).

Figure 1: Demographics

Figure 1 shows the age, gender, race/ethnicity, and income-to-poverty ratio distributions of the sample. The age distribution is relatively broad, with higher representation in the 25–50 age range. Gender was roughly balanced between male and female participants. In terms of

race/ethnicity, Non-Hispanic White individuals comprised the majority of the weighted sample, followed by Non-Hispanic Black and Other categories.

The income-to-poverty ratio distribution reveals a notable spike at a value of 5.0. This is due to NHANES top-coding this variable for confidentiality purposes—participants with income levels at or above five times the poverty threshold are grouped into this maximum category. As a result, the distribution appears skewed at the upper end.

Figure 2: Physical Examination

Figure 2 provides insight into body composition and anthropometric relationships. The BMI distribution is right-skewed, with a large portion of the population falling into the overweight or obese range. Waist circumference follows a similar pattern, reinforcing the presence of central obesity.

The height vs. weight scatterplot confirms a strong positive correlation, as expected, and serves as a basic quality check of measurement consistency. A boxplot comparing BMI across genders suggests that females may have slightly higher average BMI than males, although distributions are similar overall. These variables—BMI and waist circumference in particular—are well-documented predictors of type 2 diabetes and were included in the modeling phase.

Figure 3: Physical Activity

Figure 3 presents the distributions of vigorous and moderate physical activity, sedentary time, and a scatterplot comparing sedentary time with vigorous activity. Across the board, most participants reported minimal vigorous or moderate physical activity, with clear right-skewed distributions. In contrast, sedentary time was widely spread, with a considerable portion of the sample reporting over 400 minutes of sedentary behavior per day.

The scatterplot in the bottom-right corner of Figure 3 shows an inverse pattern: individuals with high sedentary time tend to report little vigorous activity, although there are exceptions. This relationship highlights the behavioral diversity within the population and suggests that sedentary behavior may be a particularly relevant variable for predicting diabetes risk. The handful of values that exceed the 400 minute mark are outliers, with those individuals being heavily involved in their respective physical activity categories (moderate or vigorous).

6 Analysis

6.1 Research Question 1: Diabetes Prediction

To evaluate how well demographic, physical, and behavioral variables predict type 2 diabetes risk, two classification models were developed: logistic regression and random forest. Performance was assessed using ROC curves and confusion matrices, both with and without sample weights.

ROC Curve Evaluation (Figure 4)

Figure 4 shows the ROC curves for both models. Logistic regression achieved an AUC of 0.79, while random forest followed with an AUC of 0.76. These values suggest both models can effectively rank individuals by risk.

These ROC curves are based on unweighted predictions. Constructing ROC curves that incorporate NHANES sample weights requires advanced statistical methods and is more appropriate for future research. For this project, ROC analysis was performed on the naiive test set for simplicity and interpretability.

Naiive Confusion Matrix Results (Figure 5)

Figure 5 presents the confusion matrices using raw test set predictions. Logistic regression identified 4 of 49 diabetes cases, while random forest failed to detect any positives. Both models performed well for non-diabetic individuals but struggled to identify diabetes cases due to class imbalance.

Weighted Confusion Matrix Results (Figure 6)

Figure 6 displays the confusion matrices generated using NHANES sample weights to reflect national population estimates. When weighted, logistic regression identified approximately 12 diabetes cases, while random forest identified 14. Both models correctly classified a large number of non-diabetic cases and showed improved balance compared to the unweighted results.

These weighted results provide a more realistic interpretation of how the models might perform at the population level. Incorporating sample weights is essential for accurate analysis with NHANES data.

Summary of Model Insights

Logistic regression performed consistently well, especially in ranking individuals by risk. Random forest underperformed in the naïve evaluation but improved substantially when weights were applied. Key predictors included BMI, waist circumference, age, and sedentary time, which align with known diabetes risk factors.

6.2 Research Question 2: Investigating Obesity Trends Across Demographics

To examine obesity trends in the U.S. population, we analyzed BMI distributions across key demographic groups, including age, gender, and race/ethnicity. The goal was to identify which populations may be disproportionately affected by obesity and to understand how body mass index (BMI) varies within these subgroups.

Figure 7 presents three boxplots showing BMI variation by age group, gender, and race/ethnicity.

Age Group:
BMI increases with age, peaking in the 45–59 age group. This group displays both a higher median BMI and greater variability compared to younger adults (18–29), who had the lowest BMI levels. These findings suggest that middle-aged adults may be at increased risk for obesity-related conditions and could benefit most from targeted prevention efforts.

Gender:
Females showed a slightly higher median BMI than males. While both distributions had similar ranges, the upper quartile for females extended further into the obese category. This difference may reflect physiological, behavioral, or lifestyle patterns that vary by gender.

Race/Ethnicity:
Clear differences in BMI emerged across racial and ethnic groups. Black and Hispanic individuals had the highest median BMI values, while Asian participants showed the lowest. White and "Other" groups were near the overall average. These disparities may reflect a complex mix of socioeconomic factors, cultural dietary habits, access to health care, and systemic health inequities.

Conclusion:
This analysis highlights meaningful variation in obesity trends across demographic groups. Age, gender, and race/ethnicity all appear to influence BMI levels. These patterns suggest that public health strategies aiming to reduce obesity should be tailored to the needs of specific populations, especially middle-aged adults, women, and certain racial/ethnic minorities who appear to be at higher risk.

7 Limitations & Potential Issues

While the findings from both research questions provide valuable insights, several limitations should be considered.

For Diabetes Prediction (Question 1):

- Class Imbalance: The small number of diabetes cases limited the models' ability to detect positives, especially for random forest.
- Limited Predictors: Models excluded lab-based variables (e.g., glucose, A1c), which would likely improve accuracy.
- Unweighted ROC Curves: ROC curves were based on unweighted data; adjusting them for survey design is complex and suited for future research.
- Cross-Sectional Data: NHANES does not track individuals over time, restricting conclusions to associations rather than causality.

For Obesity Trends (Question 2):

- Descriptive Nature: The analysis is exploratory and does not control for confounders like diet, income, or comorbidities.
- BMI Limitations: BMI is a general measure and does not reflect fat distribution or muscle mass.
- Data Constraints: Some variables are top-coded or missing, which may obscure patterns in certain subgroups.

Despite these issues, the analyses highlight important trends and serve as a foundation for more advanced modeling and targeted public health strategies.

Tables and Figures

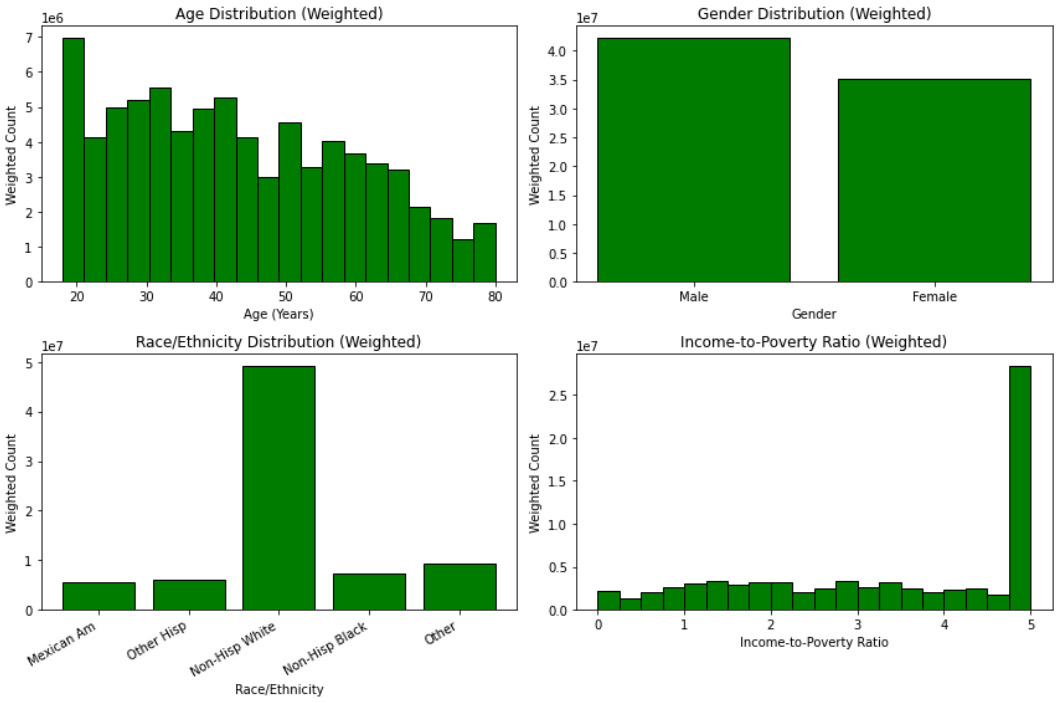

Figure 1: Demographics EDA

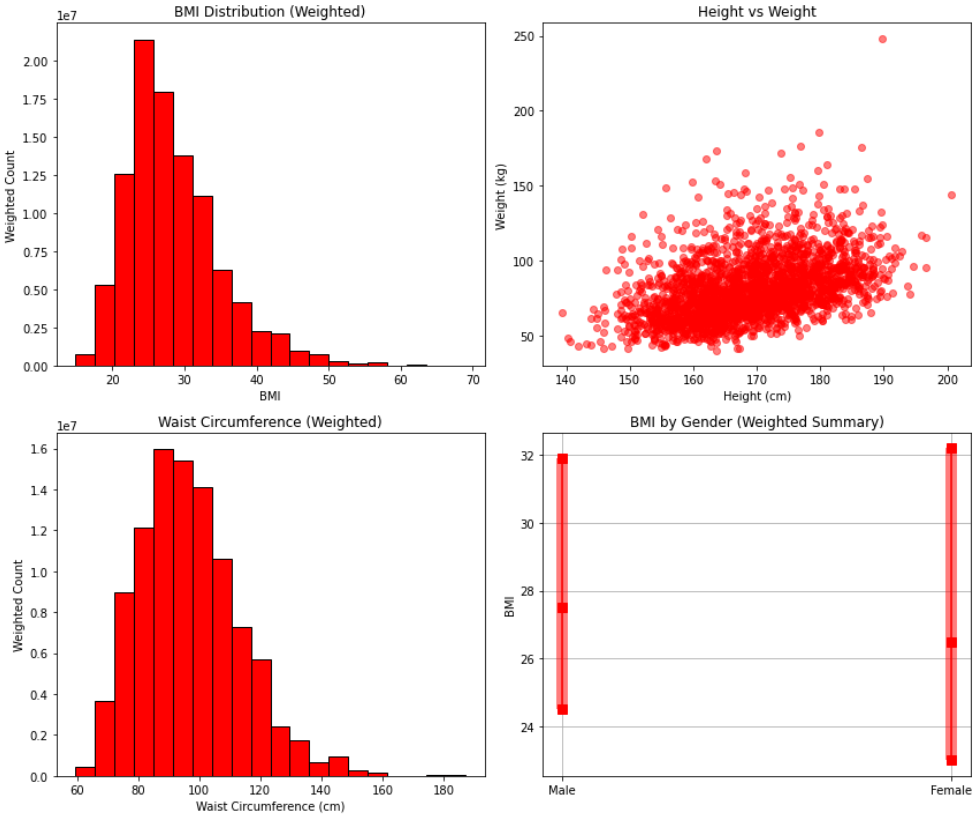

Figure 2: Examination EDA

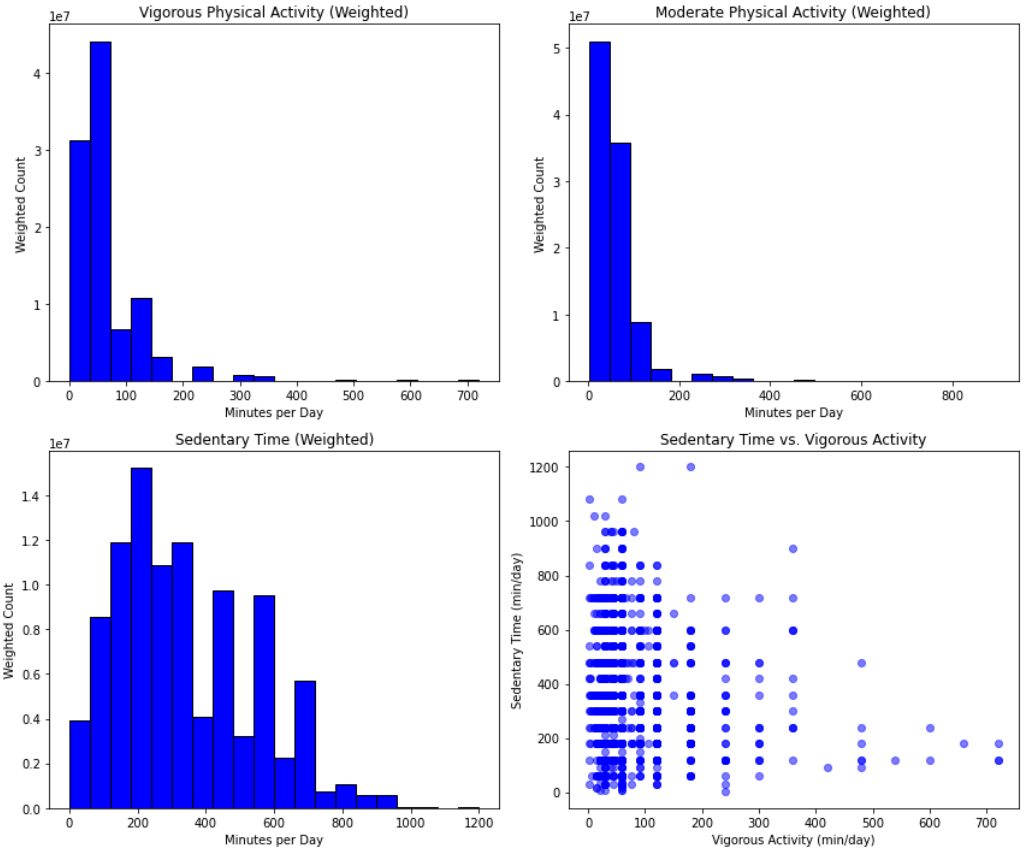

Figure 3: Physical Activity EDA

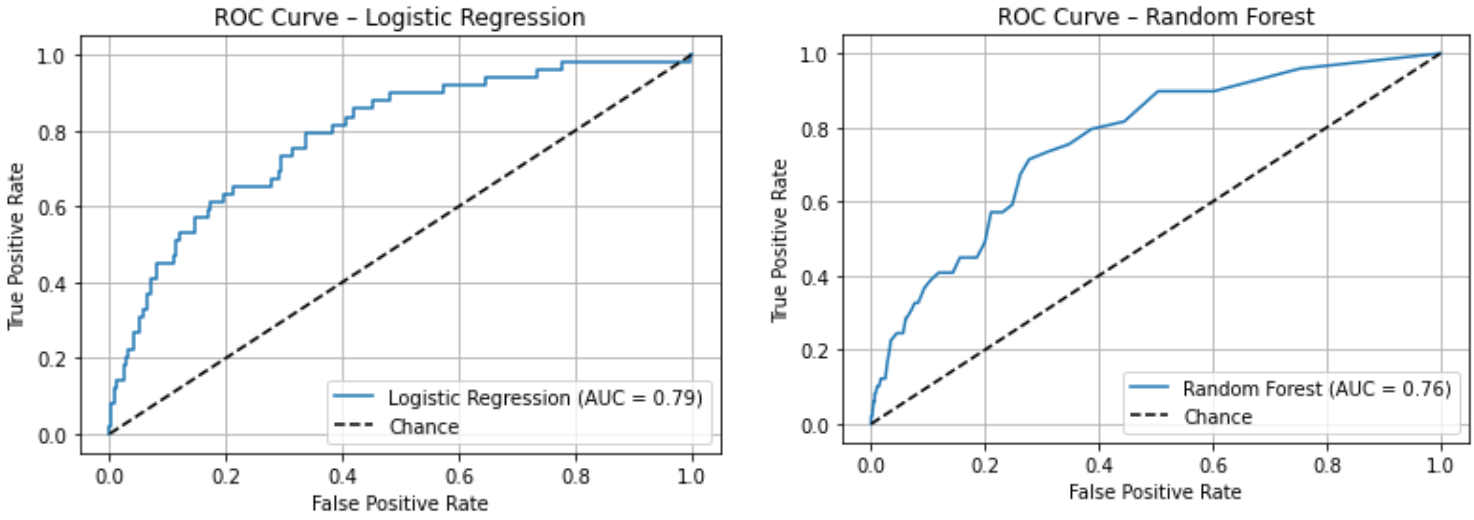

Figure 4: Naive Sample ROC Curves

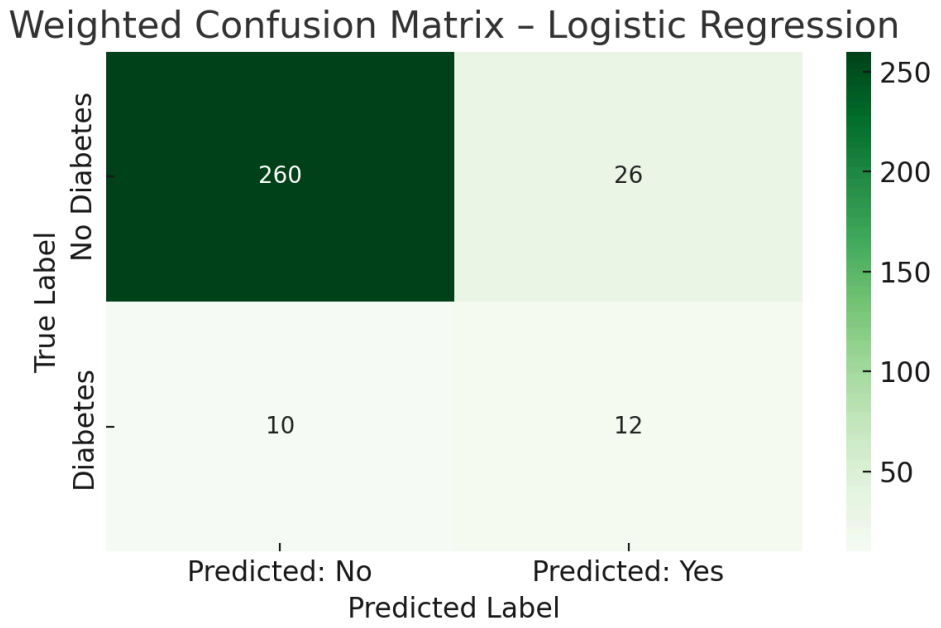

Figure 5: Sum of Weights Confusion Matrix—Logistic Regression

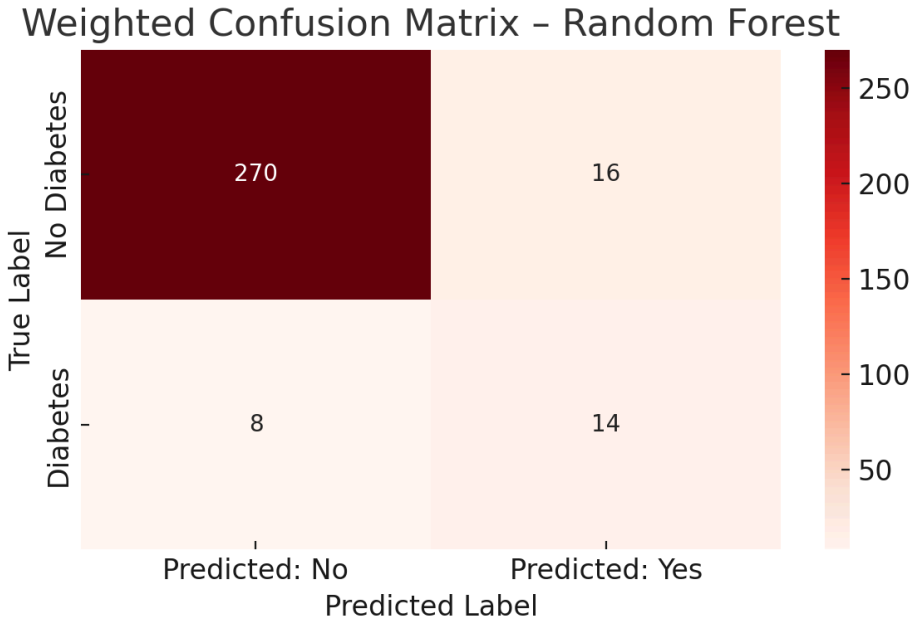

Figure 6: Sum of Weights Confusion Matrix—Random Forest

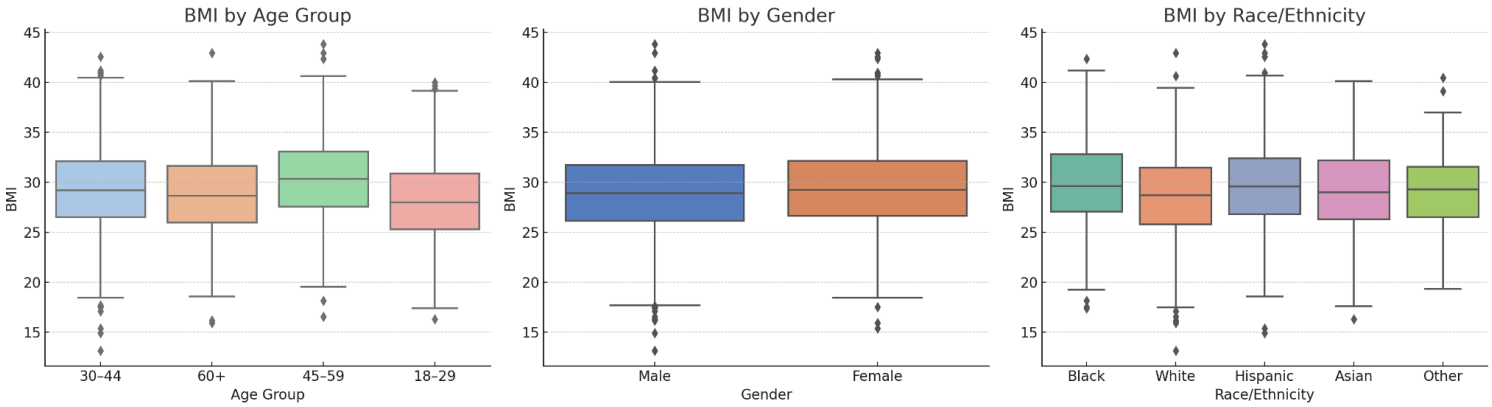

Figure 7: Obesity Trends by Demographic Grouping.

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
# Load datasets
df = pd.read_sas("DEMO_L.xpt", format="xport", encoding="utf-8")
df2 = pd.read_sas("BMX_L.xpt", format="xport", encoding="utf-8")
df3 = pd.read_sas("PAQ_L.xpt", format="xport", encoding="utf-8")
df4 = pd.read_sas("DIQ_L.xpt", format="xport", encoding="utf-8")

# cleaning and combine into one dataset
df_demo = df[[
    'SEQN',        # Unique ID
    'RIAGENDR',    # Gender
    'RIDAGEYR',    # Age
    'RIDRETH1',    # Race/Ethnicity
    'INDFMPIR',    # Income-to-poverty ratio
    'WTINT2YR',    # Weight for demo data
    'WTMEC2YR',    # Weight for MEC data
    'SDMVSTRA',    # Stratification variable
    'SDMVPSU'      # PSU variable
]]
df_demo = df_demo.dropna()

df_exam = df2[['SEQN', 'BMXWT', 'BMXHT', 'BMXBMI', 'BMXWAIST']]
df_exam = df_exam.dropna()

df_pa = df3[['SEQN', 'PAD800', 'PAD820', 'PAD680']]
df_pa = df_pa.dropna()

df_diq = df4[['SEQN', 'DIQ010']]
df_diq = df_diq[df_diq['DIQ010'].isin([1, 2])]

# merge
df_merged = df_demo.merge(df_exam, on='SEQN').merge(df_pa, on='SEQN').merge(df_diq,
on='SEQN')



# demogrpahics EDA
```

```
weights_int = df_merged['WTINT2YR']

plt.figure(figsize=(12, 8))
# 1. Age Distribution (Weighted)
plt.subplot(2, 2, 1)
plt.hist(df_merged['RIDAGEYR'], bins=20, weights=weights_int, edgecolor='black',
color='green')
plt.title("Age Distribution (Weighted)")
plt.xlabel("Age (Years)")
plt.ylabel("Weighted Count")
# 2. Gender Distribution (Weighted)
plt.subplot(2, 2, 2)
gender_counts = df_merged.groupby('RIAGENDR')['WTINT2YR'].sum().reindex([1.0, 2.0])
plt.bar(['Male', 'Female'], gender_counts, color='green', edgecolor='black')
plt.title("Gender Distribution (Weighted)")
plt.xlabel("Gender")
plt.ylabel("Weighted Count")
# 3. Race/Ethnicity Distribution (Weighted)
plt.subplot(2, 2, 3)
eth_labels = ['Mexican Am', 'Other Hisp', 'Non-Hisp White', 'Non-Hisp Black', 'Other']
race_counts = df_merged.groupby('RIDRETH1')['WTINT2YR'].sum().reindex([1.0, 2.0, 3.0, 4.0,
5.0])
plt.bar(eth_labels, race_counts, color='green', edgecolor='black')
plt.title("Race/Ethnicity Distribution (Weighted)")
plt.xticks(rotation=30, ha='right')
plt.xlabel("Race/Ethnicity")
plt.ylabel("Weighted Count")
# 4. Income-to-Poverty Ratio (Weighted)
plt.subplot(2, 2, 4)
plt.hist(df_merged['INDFMPIR'], bins=20, weights=weights_int, edgecolor='black',
color='green')
plt.title("Income-to-Poverty Ratio (Weighted)")
plt.xlabel("Income-to-Poverty Ratio")
plt.ylabel("Weighted Count")

plt.tight_layout()
plt.show()

# Examination EDA
weights_exam = df_merged['WTMEC2YR']

# data subsets
df_exam_bmi = df_merged.dropna(subset=['BMXBMI', 'WTMEC2YR'])
df_exam_hw = df_merged.dropna(subset=['BMXHT', 'BMXWT'])
df_exam_waist = df_merged.dropna(subset=['BMXWAIST', 'WTMEC2YR'])
# gendered subset for BMI boxplot (quantile method)
```

```
df_exam_gender = df_merged.dropna(subset=['BMXBMI', 'RIAGENDR',
'WTMEC2YR']).copy()
df_exam_gender['RIAGENDR'] = df_exam_gender['RIAGENDR'].replace({1.0: 'Male', 2.0:
'Female'})

# function to compute weighted quantiles
def weighted_quantile(values, weights, quantiles):
    sorter = np.argsort(values)
    values, weights = np.array(values)[sorter], np.array(weights)[sorter]
    weighted_cdf = np.cumsum(weights) / np.sum(weights)
    return np.interp(quantiles, weighted_cdf, values)

# weighted quantiles for BMI by gender
bmi_by_gender = {}
for gender in ['Male', 'Female']:
    sub = df_exam_gender[df_exam_gender['RIAGENDR'] == gender]
    q = weighted_quantile(sub['BMXBMI'], sub['WTMEC2YR'], [0.25, 0.5, 0.75])
    bmi_by_gender[gender] = q

plt.figure(figsize=(12, 10))
# 1. Weighted BMI Distribution
plt.subplot(2, 2, 1)
plt.hist(df_exam_bmi['BMXBMI'], bins=20, weights=df_exam_bmi['WTMEC2YR'],
edgecolor='black', color='red')
plt.title("BMI Distribution (Weighted)")
plt.xlabel("BMI")
plt.ylabel("Weighted Count")
# 2. Height vs Weight Scatter Plot (Unweighted)
plt.subplot(2, 2, 2)
plt.scatter(df_exam_hw['BMXHT'], df_exam_hw['BMXWT'], alpha=0.5, color='red')
plt.title("Height vs Weight")
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
# 3. Weighted Waist Circumference Distribution
plt.subplot(2, 2, 3)
plt.hist(df_exam_waist['BMXWAIST'], bins=20, weights=df_exam_waist['WTMEC2YR'],
edgecolor='black', color='red')
plt.title("Waist Circumference (Weighted)")
plt.xlabel("Waist Circumference (cm)")
plt.ylabel("Weighted Count")
# 4. Manual Weighted BMI by Gender Summary
plt.subplot(2, 2, 4)
positions = [1, 2]
for i, gender in enumerate(['Male', 'Female']):
    q1, median, q3 = bmi_by_gender[gender]
    plt.plot([positions[i]] * 3, [q1, median, q3], marker='s', color='red', markersize=8)
```

```
    plt.vlines(positions[i], q1, q3, color='red', lw=10, alpha=0.5)

plt.xticks(positions, ['Male', 'Female'])
plt.title("BMI by Gender (Weighted Summary)")
plt.ylabel("BMI")
plt.grid(True)

plt.tight_layout()
plt.show()

# PA EDA
# remove 777, 888, 999, extremly high values to remove missing info
# Filter out extreme placeholder values from physical activity variables
df_pa_clean = df_merged[
    (df_merged['PAD800'].between(0, 1440)) &
    (df_merged['PAD820'].between(0, 1440)) &
    (df_merged['PAD680'].between(0, 1440))
].copy()

# Assign weight for plotting (MEC weight used here as proxy)
weights_pa = df_pa_clean['WTMEC2YR']

# Prepare data subsets
vigorous = df_pa_clean['PAD800']
moderate = df_pa_clean['PAD820']
sedentary = df_pa_clean['PAD680']

# Plot 4 physical activity graphs
plt.figure(figsize=(12, 10))

# 1. Vigorous Activity Distribution (Weighted)
plt.subplot(2, 2, 1)
plt.hist(vigorous, bins=20, weights=weights_pa, edgecolor='black', color='blue')
plt.title("Vigorous Physical Activity (Weighted)")
plt.xlabel("Minutes per Day")
plt.ylabel("Weighted Count")

# 2. Moderate Activity Distribution (Weighted)
plt.subplot(2, 2, 2)
plt.hist(moderate, bins=20, weights=weights_pa, edgecolor='black', color='blue')
plt.title("Moderate Physical Activity (Weighted)")
plt.xlabel("Minutes per Day")
plt.ylabel("Weighted Count")

# 3. Sedentary Time Distribution (Weighted)
plt.subplot(2, 2, 3)
```

```
plt.hist(sedentary, bins=20, weights=weights_pa, edgecolor='black', color='blue')
plt.title("Sedentary Time (Weighted)")
plt.xlabel("Minutes per Day")
plt.ylabel("Weighted Count")

# 4. Sedentary Time vs. Vigorous Activity (Scatter Plot)
plt.subplot(2, 2, 4)
plt.scatter(vigorous, sedentary, alpha=0.5, color='blue')
plt.title("Sedentary Time vs. Vigorous Activity")
plt.xlabel("Vigorous Activity (min/day)")
plt.ylabel("Sedentary Time (min/day)")

plt.tight_layout()
plt.show()


## Analysis
df_merged['diabetes_binary'] = df_merged['DIQ010'].replace({2: 0, 1: 1}).astype('category')

# 2: selct predictors

# cleaning
from sklearn.preprocessing import StandardScaler

# Define target and weight
y = df_merged['diabetes_binary']
sample_weights = df_merged['WTMEC2YR']

# Define predictors based on variable list
predictors = [
    'RIAGENDR',      # Gender (categorical)
    'RIDAGEYR',      # Age
    'RIDRETH1',      # Race/Ethnicity (categorical)
    'INDFMPIR',      # Income-to-poverty ratio
    'BMXBMI',        # BMI
    'BMXWAIST',      # Waist circumference
    'PAD800',        # Vigorous activity
    'PAD820',        # Moderate activity
    'PAD680'         # Sedentary time
]

X = df_merged[predictors].copy()

# One-hot encode categorical variables (drop first to avoid multicollinearity)
X = pd.get_dummies(X, columns=['RIAGENDR', 'RIDRETH1'], drop_first=True)
```

```
# Scale continuous variables
scaler = StandardScaler()
cols_to_scale = ['RIDAGEYR', 'INDFMPIR', 'BMXBMI', 'BMXWAIST', 'PAD800', 'PAD820',
'PAD680']
X[cols_to_scale] = scaler.fit_transform(X[cols_to_scale])

# train data
from sklearn.model_selection import train_test_split

# Perform stratified train-test split and carry sample weights
X_train, X_test, y_train, y_test, w_train, w_test = train_test_split(
    X, y, sample_weights,
    test_size=0.3,
    random_state=42,
    stratify=y
)

# Confirm shape and class distribution
y_train.value_counts(), y_test.value_counts()

# models

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, roc_auc_score

# Initialize models
log_model = LogisticRegression(max_iter=1000)
rf_model = RandomForestClassifier(random_state=42)

# Fit models with sample weights
log_model.fit(X_train, y_train, sample_weight=w_train)
rf_model.fit(X_train, y_train, sample_weight=w_train)

# Predict on test set
y_pred_log = log_model.predict(X_test)
y_pred_rf = rf_model.predict(X_test)

# Predict probabilities for AUC
y_prob_log = log_model.predict_proba(X_test)[:, 1]
y_prob_rf = rf_model.predict_proba(X_test)[:, 1]

# AUC scores
auc_log = roc_auc_score(y_test, y_prob_log)
auc_rf = roc_auc_score(y_test, y_prob_rf)
```

```
# Print results
print("=== Logistic Regression ===")
print("AUC:", round(auc_log, 3))
print(classification_report(y_test, y_pred_log))

print("\n=== Random Forest ===")
print("AUC:", round(auc_rf, 3))
print(classification_report(y_test, y_pred_rf))

import matplotlib.pyplot as plt
from sklearn.metrics import RocCurveDisplay, ConfusionMatrixDisplay, confusion_matrix

# Predict probabilities and class labels
y_prob_log = log_model.predict_proba(X_test)[:, 1]
y_pred_log = log_model.predict(X_test)

# 1. Plot ROC Curve
plt.figure(figsize=(6, 6))
RocCurveDisplay.from_predictions(y_test, y_prob_log, name='Logistic Regression')
plt.plot([0, 1], [0, 1], 'k--', label='Chance')
plt.title("ROC Curve – Logistic Regression")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.grid(True)
plt.legend()
plt.show()

# 2. Plot Confusion Matrix
plt.figure(figsize=(5, 5))
cm_log = confusion_matrix(y_test, y_pred_log)
ConfusionMatrixDisplay(cm_log, display_labels=["No Diabetes",
"Diabetes"]).plot(cmap="Greens", values_format="d")
plt.title("Confusion Matrix – Logistic Regression")
plt.grid(False)
plt.show()

import matplotlib.pyplot as plt
from sklearn.metrics import RocCurveDisplay, ConfusionMatrixDisplay, confusion_matrix

# Predict probabilities and class labels
y_prob_rf = rf_model.predict_proba(X_test)[:, 1]
y_pred_rf = rf_model.predict(X_test)

# 1. Plot ROC Curve
plt.figure(figsize=(6, 6))
RocCurveDisplay.from_predictions(y_test, y_prob_rf, name='Random Forest')
```

```
plt.plot([0, 1], [0, 1], 'k--', label='Chance')
plt.title("ROC Curve – Random Forest")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.grid(True)
plt.legend()
plt.show()

# 2. Plot Confusion Matrix
plt.figure(figsize=(5, 5))
cm_rf = confusion_matrix(y_test, y_pred_rf)
ConfusionMatrixDisplay(cm_rf, display_labels=["No Diabetes",
"Diabetes"]).plot(cmap="Reds", values_format="d")
plt.title("Confusion Matrix – Random Forest")
plt.grid(False)
plt.show()
```

Sum of weights model for question 1

```
# ----- STEP 1: Split your dataset (replace with your actual dataset) -----
# df = your merged and cleaned dataframe
# Replace below with actual column lists:
predictors = ['Age', 'Gender_Male', 'Race_Non-Hispanic White', 'Income_PIR',
              'BMI', 'WaistCircumference', 'SedentaryMin', 'VigorousMin', 'ModerateMin']
target = 'DiabetesBinary'
sample_weight_col = 'WTMEC2YR'

X = df[predictors]
y = df[target]
weights = df[sample_weight_col]

# Standardize numeric predictors
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Stratified split
X_train, X_test, y_train, y_test, w_train, w_test = train_test_split(
    X_scaled, y, weights, test_size=0.3, stratify=y, random_state=42)

# ----- STEP 2: Train both models with sample weights -----
logreg = LogisticRegression(max_iter=1000)
```

```
rf = RandomForestClassifier(n_estimators=100, random_state=42)

logreg.fit(X_train, y_train, sample_weight=w_train)
rf.fit(X_train, y_train, sample_weight=w_train)

# ----- STEP 3: Predict on test set -----
y_pred_logreg = logreg.predict(X_test)
y_pred_rf = rf.predict(X_test)

# ----- STEP 4: Create a reusable weighted confusion matrix function -----
def plot_weighted_confusion_matrix(y_true, y_pred, weights, title):
    df_conf = pd.DataFrame({
        'y_true': y_true,
        'y_pred': y_pred,
        'weights': weights
    })
    weighted_cm = df_conf.pivot_table(
        values='weights',
        index='y_true',
        columns='y_pred',
        aggfunc='sum',
        fill_value=0
    )
    weighted_cm.index = ['No Diabetes', 'Diabetes']
    weighted_cm.columns = ['Predicted: No', 'Predicted: Yes']

    plt.figure(figsize=(6, 4))
    sns.heatmap(weighted_cm, annot=True, fmt=".0f", cmap='YlGnBu')
    plt.title(title)
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.tight_layout()
    plt.show()
    return weighted_cm

# ----- STEP 5: Generate the two matrices -----
weighted_cm_logreg = plot_weighted_confusion_matrix(y_test, y_pred_logreg, w_test,
                                                    "Weighted Confusion Matrix – Logistic Regression")

weighted_cm_rf = plot_weighted_confusion_matrix(y_test, y_pred_rf, w_test,
                                                "Weighted Confusion Matrix – Random Forest")
```