# WEIGHT CHANGE PREDICTIVE MODEL

## Simulation Modelling

Ngocanh Nguyen
Huyen Truong
Dung Nguyen
Eunjeong Cho

# Acknowledgement

We would like to express our sincere gratitude to Dr. Suchismita Das for their invaluable guidance and support throughout this research. We also extend our deepest appreciation to every team member for their unwavering dedication and unique contributions. Each member brought their distinct skills and expertise, forming a dynamic and collaborative force that elevated the project to new levels of excellence.

The cooperative spirit within our team fostered an environment of open communication and knowledge-sharing, establishing a solid foundation for innovation. The synergy created by our collective efforts not only improved the quality of the project but also became the driving force behind our successful teamwork in achieving ambitious goals.

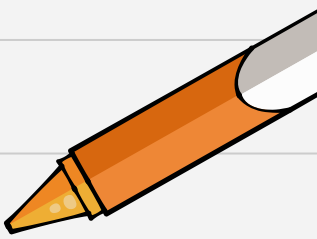
Moreover, we acknowledge and appreciate the dedication and hard work of each team member, whose relentless efforts were crucial to the successful completion of this research. The diverse talents and perspectives each member brought to the table enriched the project, showcasing the power of combining various skills and viewpoints.

In expressing our gratitude, we recognize that this research is a true collective achievement, reflecting the passion and professionalism of our team. We look forward to applying the lessons learned and the camaraderie developed during this project to future endeavors, confident in our ability to overcome challenges and achieve excellence as a united and committed team.

# Content

# 01

# INTRODUCTION



# 1.1. General introduction

Weight management is a critical component of personal health, influencing not only physical well-being but also mental health and quality of life. This project aims to develop a predictive model to analyze and understand the factors contributing to weight change. The dataset focuses on the dietary, lifestyle, and demographic information of 100 participants over time to predict weight fluctuations.

The data includes attributes such as age, gender, current weight, basal metabolic rate (BMR), daily caloric intake, caloric surplus/deficit, sleep quality, stress level, and physical activity level. It aims to provide insights into how these variables interact and affect weight changes, helping identify actionable strategies for healthier weight management.

Our target variable is **Weight Change (lbs)**, a continuous variable representing the change in participants' weight over a specified period. The key predictors include:
**Daily Calories Consumed**: A measure of the caloric intake and its alignment with BMR.
**Physical Activity Level**: The extent to which physical exercise contributes to caloric expenditure.
**Stress Level and Sleep Quality**: Indicators of lifestyle factors that indirectly influence weight change.

This project holds potential benefits for various stakeholders:
**Individuals**: By understanding the factors that drive weight change, individuals can adopt healthier habits tailored to their unique needs.
**Nutritionists and Fitness Professionals**: Insights from the model can guide personalized diet and exercise plans.
**Researchers and Public Health Advocates**: The dataset offers a valuable resource to explore connections between lifestyle factors and weight management, enabling informed interventions to tackle obesity and weight-related issues.

**Source**: This dataset, derived from Kaggle.com, can be accessed at Comprehensive Weight Change Prediction Dataset [1].

## 1.2. Problem Approach

This project aims to develop a predictive model to analyze and understand the factors contributing to weight change. Specifically, we focus on building a **multiple linear regression model** to explore relationships between weight change and predictors such as dietary habits, physical activity, and lifestyle factors.

However, the dataset is relatively small, with only 100 participants. This limited data size presents significant challenges:

**Patterns in Residuals**: The small dataset may lead to non-random residual patterns, such as heteroscedasticity or clustering, which indicate that the model does not fully capture the variability in the data.

**Model Overfitting**: With fewer data points, the model may fit noise in the data rather than the underlying relationships, reducing its generalizability.

**Unstable Estimates**: Regression coefficients can become unreliable and highly sensitive to small changes in the data due to limited sample size.

To address these issues, we incorporate **bootstrap resampling**, a robust statistical technique that generates multiple samples by randomly resampling with replacement from the original data. This method helps:

**Stabilize Model Estimates**: By averaging over many resampled datasets, bootstrap reduces the variability in model coefficients.

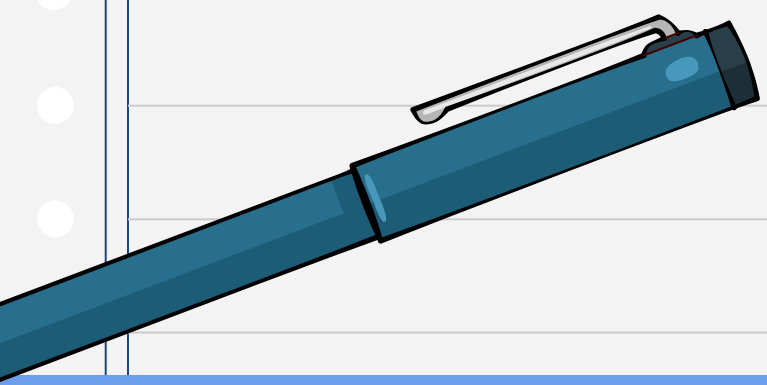
**Improve Residual Patterns**: With more resampled data, the residual plots are expected to show more randomness, indicating a better model fit.

**Enhance Reliability**: Bootstrap increases confidence in predictions, even with small datasets.

Through this approach, we aim to demonstrate the advantages of bootstrap in improving the reliability and robustness of statistical models built on limited datasets.

# 02.

# UNDERSTANDING THE DATA



# 2.1. Closer Look at Key Data Attributes

This project leverages a dataset sourced from Kaggle, comprising detailed demographic, dietary, and lifestyle attributes of 100 participants. Below is a closer examination of the key attributes and their significance to the study:

**1. Participant ID**

A unique identifier assigned to each participant to ensure data integrity and accurate tracking throughout the analysis.

**2. Age**

The participant's age in years. Age is a critical factor affecting metabolism, hormonal balance, and weight change over time.

**3. Gender**

A categorical attribute (M/F) that reflects physiological differences influencing weight management, such as hormonal variations and body composition.

**4. Current Weight (lbs)**

The participant’s initial weight at the start of the study, serving as a baseline for evaluating weight changes.

**5. BMR (Calories)**

Basal Metabolic Rate, calculated using the Mifflin-St Jeor equation, represents the number of calories required for maintaining bodily functions at rest.

**6. Daily Calories Consumed**

Total daily caloric intake, including variability to reflect real-world eating habits and patterns. This variable is pivotal in determining caloric surplus or deficit.

# 2.1. Closer Look at Key Data Attributes

**7. Daily Caloric Surplus/Deficit**

The difference between calories consumed and the BMR. Positive values indicate a surplus (potential weight gain), while negative values reflect a deficit (potential weight loss).

**8. Weight Change (lbs)**

The target variable for this study, representing the estimated change in participants' weight over the specified duration, based on caloric surplus or deficit.

**9. Duration (weeks)**

The time period (ranging from 1 to 12 weeks) over which weight change is measured. This contextualizes the magnitude of weight changes.

**10. Physical Activity Level**

Self-reported activity level, categorized as Sedentary, Lightly Active, Moderately Active, or Very Active. This variable directly impacts caloric expenditure and weight outcomes.

**11. Macronutrient Breakdown**

The dietary composition expressed as percentages of carbohydrates, proteins, and fats. This attribute helps explore the relationship between macronutrient distribution and weight management.

**12. Sleep Quality**

A self-reported measure categorized as Poor, Fair, Good, or Excellent. Sleep quality is linked to metabolic health and the body's ability to manage weight.

**13. Stress Level**

A numerical score (1–10) representing the participant's perceived stress level. Higher stress levels can affect eating behaviors, hormonal balance, and physical activity.

# 03

# DATA INSPECTION

# 3.1 General Inspection

**1. Read and Explore the Dataset**

The dataset is loaded into R and the first few rows are displayed to understand its structure and contents.

We examine the structure of the dataset (number of rows, columns, data types), and the dataset contains **100 rows and 13 columns**.

```
'data.frame':   100 obs. of  13 variables:
 $ Participant.ID              : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Age                         : int  56 46 32 25 38 56 36 40 28 28 ...
 $ Gender                      : chr  "M" "F" "F" "F" ...
 $ Current.Weight..lbs.        : num  228 165 143 146 156 ...
 $ BMR..Calories.              : num  3102 2276 2119 2181 2464 ...
 $ Daily.Calories.Consumed     : num  3916 3823 2785 2587 3313 ...
 $ Daily.Caloric.Surplus.Deficit: num  814 1548 666 406 849 ...
 $ Weight.Change..lbs.         : num  0.2 2.4 1.4 0.8 2 ...
 $ Duration..weeks.            : int  1 6 7 8 10 9 2 11 10 2 ...
 $ Physical.Activity.Level     : chr  "Sedentary" "Very Active" "Sedentary" "Sedentary" ...
 $ Sleep.Quality               : chr  "Excellent" "Excellent" "Good" "Fair" ...
 $ Stress.Level                : int  6 6 3 2 1 6 5 9 1 7 ...
 $ Final.Weight..lbs.          : num  229 168 144 146 158 ...
```

The **summary()** function generates a quick statistical summary of numeric and categorical variables for insights into data distributions (e.g., min, max, median).

```
Participant.ID        Age           Gender          Current.Weight..lbs. BMR..Calories.
Min.   :  1.00   Min.   :18.00   Length:100         Min.   :100.0        Min.   :1566
1st Qu.: 25.75   1st Qu.:26.75   Class :character   1st Qu.:153.7        1st Qu.:2255
Median : 50.50   Median :38.00   Mode  :character   Median :172.2        Median :2520
Mean   : 50.50   Mean   :37.91                      Mean   :171.5        Mean   :2518
3rd Qu.: 75.25   3rd Qu.:46.25                      3rd Qu.:192.5        3rd Qu.:2806
Max.   :100.00   Max.   :59.00                      Max.   :238.2        Max.   :3391
Daily.Calories.Consumed Daily.Caloric.Surplus.Deficit Weight.Change..lbs. Duration..weeks.
Min.   :2031            Min.   :  82.5                Min.   :-35.678     Min.   : 1.00
1st Qu.:3233            1st Qu.: 767.0                1st Qu.: -5.012     1st Qu.: 4.00
Median :3636            Median :1013.1                Median :  0.100     Median : 7.00
Mean   :3518            Mean   :1000.1                Mean   : -2.780     Mean   : 6.92
3rd Qu.:4000            3rd Qu.:1253.3                3rd Qu.:  1.850     3rd Qu.:10.00
Max.   :4000            Max.   :1922.5                Max.   :  5.000     Max.   :12.00
Physical.Activity.Level Sleep.Quality       Stress.Level  Final.Weight..lbs.
Length:100              Length:100         Min.   :1.00   Min.   : 98.2
Class :character        Class :character   1st Qu.:2.75   1st Qu.:149.6
Mode  :character        Mode  :character   Median :5.00   Median :169.8
                                           Mean   :4.81   Mean   :168.8
                                           3rd Qu.:7.00   3rd Qu.:188.3
                                           Max.   :9.00   Max.   :232.5
```

We check for null or missing values using colSums(is.na(df). No missing values were found.

```
                Participant.ID                          Age                       Gender
                             0                            0                            0
         Current.Weight..lbs.                BMR..Calories.       Daily.Calories.Consumed
                             0                            0                            0
Daily.Caloric.Surplus.Deficit           Weight.Change..lbs.             Duration..weeks.
                             0                            0                            0
      Physical.Activity.Level                 Sleep.Quality                 Stress.Level
                             0                            0                            0
           Final.Weight..lbs.
                             0
```

# 3.1 General Inspection

We also identify duplicate rows using duplicated() and count them with sum(duplicated(df)). No duplicates were found.

**2. Feature Engineering**

First, **Participant ID** column is removed as it is not relevant for analysis. This simplifies the dataset and focuses on meaningful variables.

We create **three new columns** to enrich the dataset with derived metrics for better analysis.

**Caloric Intake Per Weight**:

- Formula: Daily Calories Consumed / Current Weight (lbs)
- This column normalizes daily caloric intake relative to an individual's weight, making it easier to compare across participants.

**Physical Activity MET Value**:

- The MET (Metabolic Equivalent of Task) value is a unit used to estimate the amount of energy expended during various physical activities. It provides a way to compare the intensity of activities and is defined as the ratio of the work metabolic rate to the resting metabolic rate.
- MET values are then mapped to activity levels.

```
met_value_map <- c(
  "Sedentary" = 1.2,
  "Very Active" = 1.9,
  "Lightly Active" = 1.375,
  "Moderately Active" = 1.55
)
```

**Activity Weighted Calories**:

- Formula: Physical Activity MET Value × BMR (Basal Metabolic Rate in Calories)
- To calculate activity-weighted calories first we need to convert the Physical Activity Level to a MET value then use the BMR (Basal Metabolic Rate) and MET value to calculate the calories burned.

Lastly, we verify that the new columns are added correctly and confirm the updated structure of the dataset.

# 3.2. Exploratory Data Analysis

## Distribution of continuous numerical variables

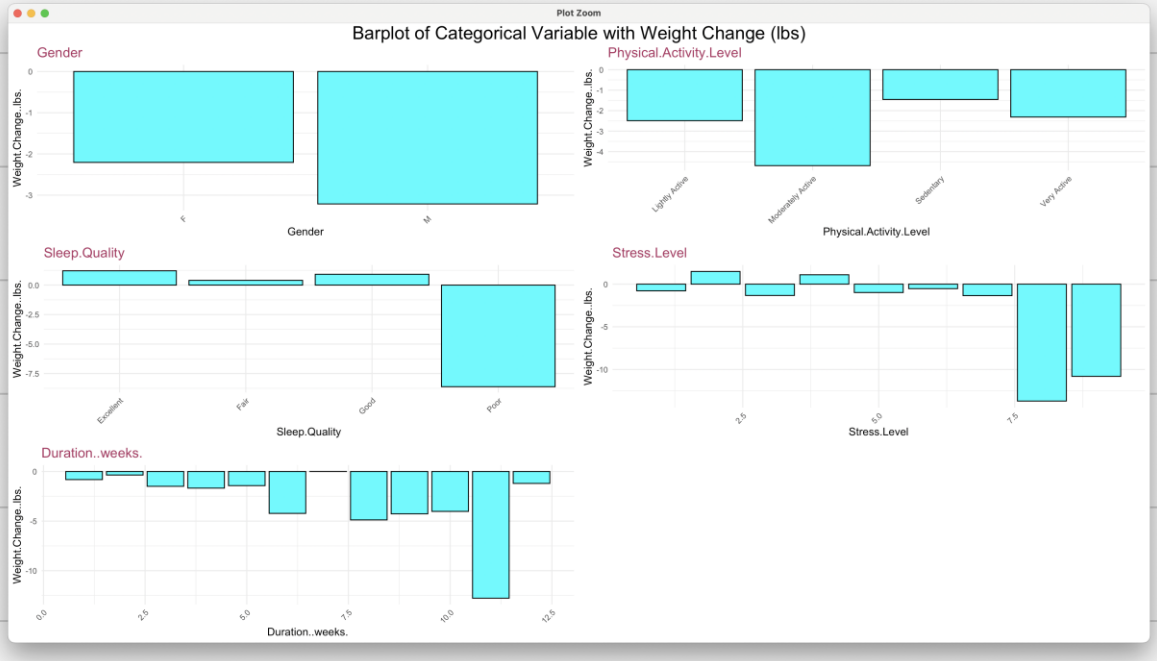

The visualization provides density plots for various variables related to weight and caloric measurements.

**Key Takeaways**

- Most distributions are symmetric, but specific variables like daily caloric intake and weight change exhibit skewness.
- The overall population shows slight weight loss, as indicated by the negative mean in "Weight Change."
- Activity and caloric variables display significant variance, highlighting diversity in energy balance and activity levels among individuals.
- The presence of outliers (e.g., extreme daily calorie consumption or activity) may influence overall trends and should be considered during further analysis.

# 3.2. Exploratory Data Analysis

## Relationship between categorical variables and targeted variable




**1.Gender**:

Females experience slightly more weight loss or smaller gains than males under similar conditions.

**2.Physical Activity**:

Very Active individuals show the greatest weight loss, while Sedentary and Lightly Active groups show minimal change. Higher activity strongly correlates with weight loss.

**3.Sleep Quality**:

Poor sleep leads to greater weight loss compared to Excellent, Good, or Fair sleep, though factors like stress or activity levels may also play a role.

**4.Stress**:

High stress levels (7.5) correspond to greater weight loss, possibly due to reduced caloric intake or other stress-related effects.

**5.Duration**:

Longer durations (10–12.5 weeks) lead to more significant weight loss, while shorter durations (<7.5 weeks) result in minimal change.

**Key Takeaways**

Physical activity and intervention duration are major contributors to weight loss. Stress and poor sleep may also play roles, while gender differences are modest, with females showing slightly greater weight loss.

# 3.2. Exploratory Data Analysis

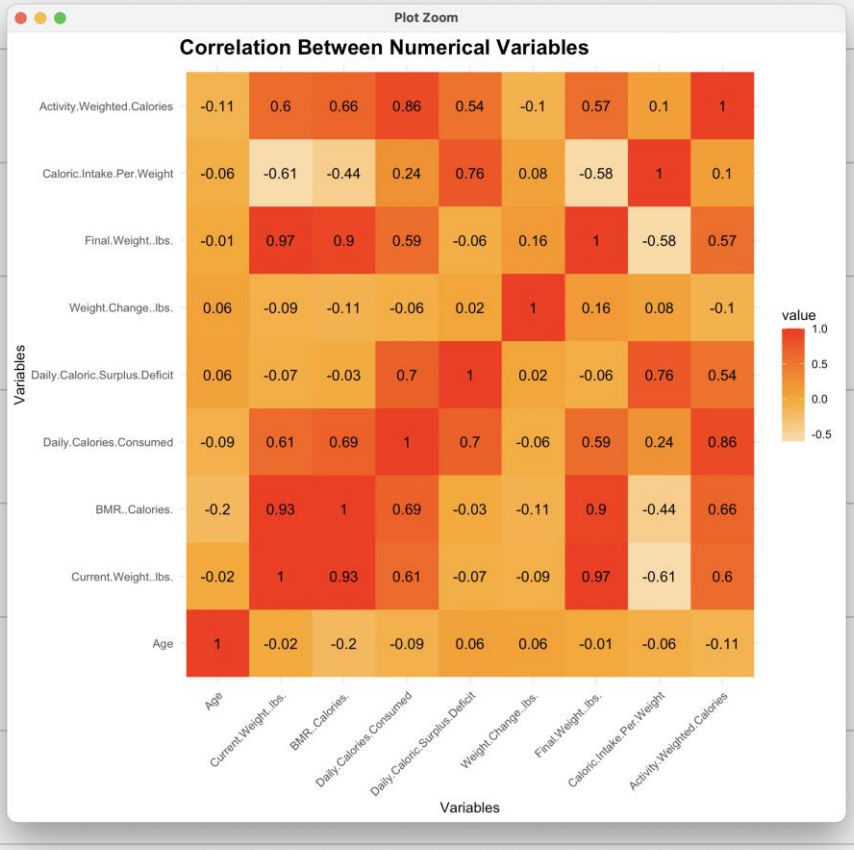
## Relationship between Weight Change (lbs) with all other numerical columns


**Lack of Relationships**: Most variables, including age, initial and final values, BMR, daily intake, normalized intake by weight, and activity-based expenditure, show no relationship with the outcome. This indicates these factors alone do not significantly influence variations in the outcome.

**Caloric Surplus/Deficit as a Key Predictor**: Caloric surplus/deficit shows the strongest relationship but remains only moderately predictive. It aligns with energy balance principles, where a surplus increases and a deficit decreases the outcome. However, variability suggests other interacting factors may contribute. Further analysis should focus on multivariate interactions to enhance understanding.

# 3.2. Exploratory Data Analysis

## Heatmap showing correlation between numerical variables




**Strong Positive Correlations:**

- **Final Weight and Current Weight:** This is a straightforward correlation, as the final weight is naturally influenced by the starting weight.
- **BMR and Current Weight:** Individuals with higher weight tend to have higher BMRs, as more metabolic activity is required to maintain a larger body mass.
- **Daily Calories Consumed and BMR:** Individuals with higher BMRs need to consume more calories to maintain their energy balance.
- **Activity-Weighted Calories and Daily Calories Consumed:** Individuals who are more active tend to burn more calories, and thus need to consume more calories to maintain their energy balance.

**Strong Negative Correlations:**

- **Final Weight and Caloric Intake per Weight:** Individuals with lower caloric intake per weight tend to lose more weight, as they are in a caloric deficit.
- **Caloric Intake per Weight and BMR:** Individuals with higher BMRs can maintain their weight on lower caloric intake per weight.

# 3.2. Exploratory Data Analysis

## Heatmap showing correlation between numerical variables

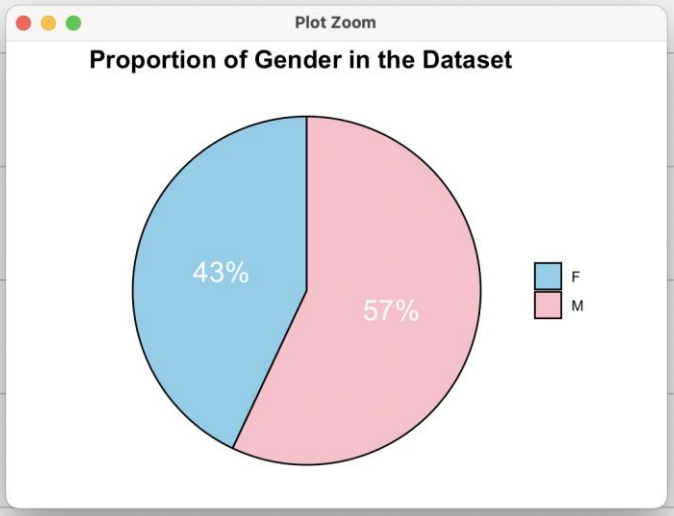
**Moderate Correlations:**

- **Daily Caloric Surplus/Deficit and Daily Calories Consumed:** Individuals who consume more calories are more likely to have a caloric surplus.
- **Weight Change and Final Weight:** Individuals with greater weight loss tend to have lower final weights.

**Conclusion**

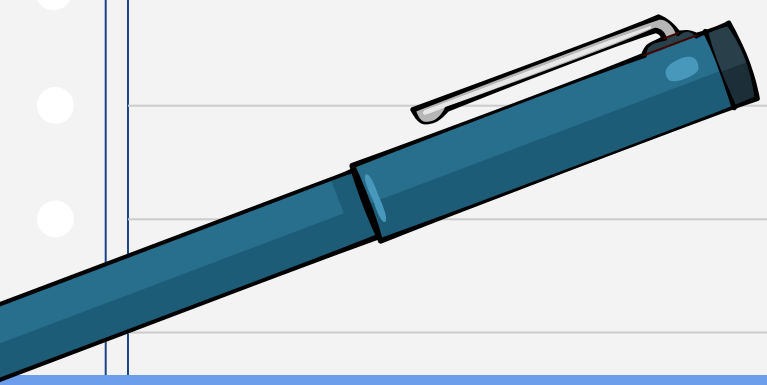
- **Multicollinearity:** Some variables, like **Current Weight** and **BMR**, are strongly correlated. This can affect the interpretation of regression models or other statistical analyses.
- Exploring correlations between features such as Daily Caloric Surplus/Deficit and Weight Change: A correlation of 0.02 between Daily Caloric Surplus/Deficit and Weight Change suggests a weak positive relationship between the two variables. As the Daily Caloric Surplus/Deficit increases, there is a slight increase in Weight Change. A correlation of -0.11 between BMR (calories) and Weight Change indicates a very weak negative relationship between the two variables. As BMR increases, Weight Change tends to slightly decrease.

The dataset also shows a balanced gender distribution with 43% female and 57% male, which is close enough to represent a relatively equitable mix. This balance is important for reducing potential gender-related biases in the analysis, ensuring that insights derived from the data are more generalizable and reflective of diverse perspectives. While not perfectly even, the distribution is reasonable and provides a fair representation for examining weight change trends across genders.



# 04

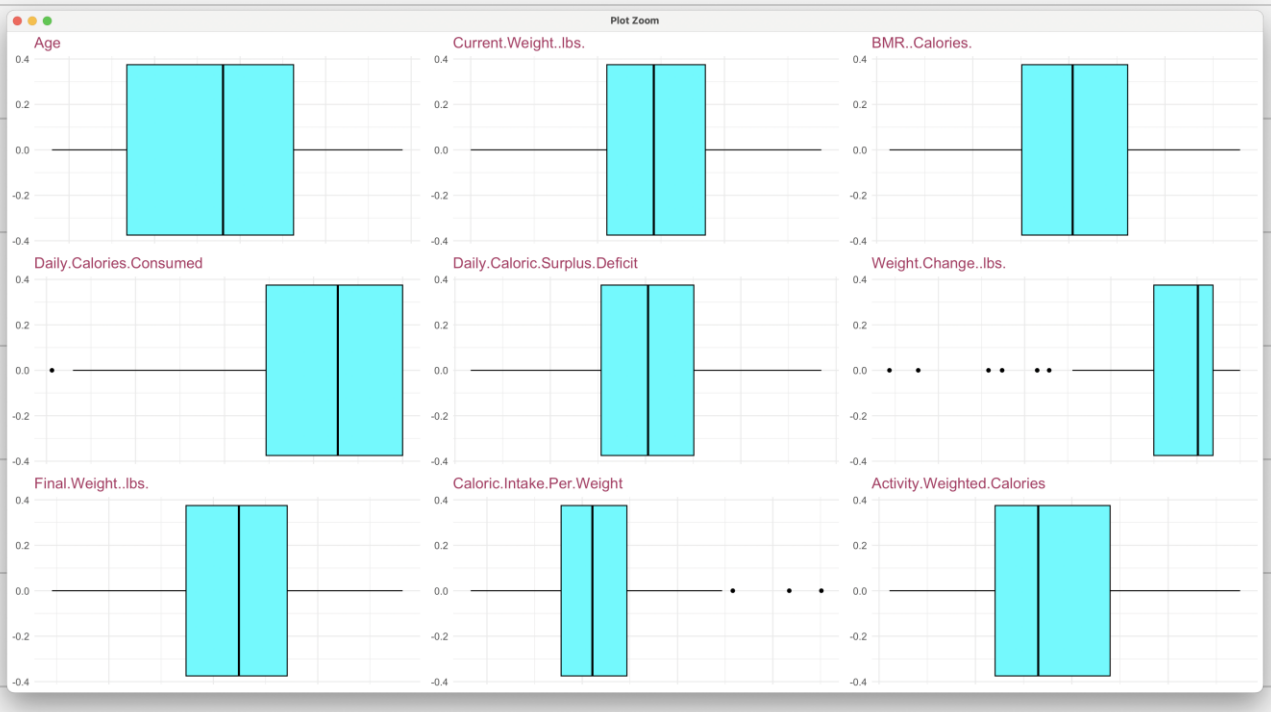
# DATA PREPROCESSING



# 4.1. Dealing with Outliers

Outliers can significantly affect the performance of regression models, especially in small datasets. To identify and address outliers, we begin by visualizing the distribution of continuous variables using box plots.

**Visualizing Outliers with Box Plots**: We used box plots to visualize the distribution of continuous variables and to identify any potential outliers. For each continuous variable, we generated a box plot that shows the spread of data, including the median, quartiles, and potential outliers.
A box plot provides a clear representation of where most of the data is centered and highlights values that fall outside the expected range (outliers).




The box plots indicated outliers in the following columns: *Daily Calories Consumed*, *Caloric Intake Per Weight*, and *Weight Change (lbs)*.

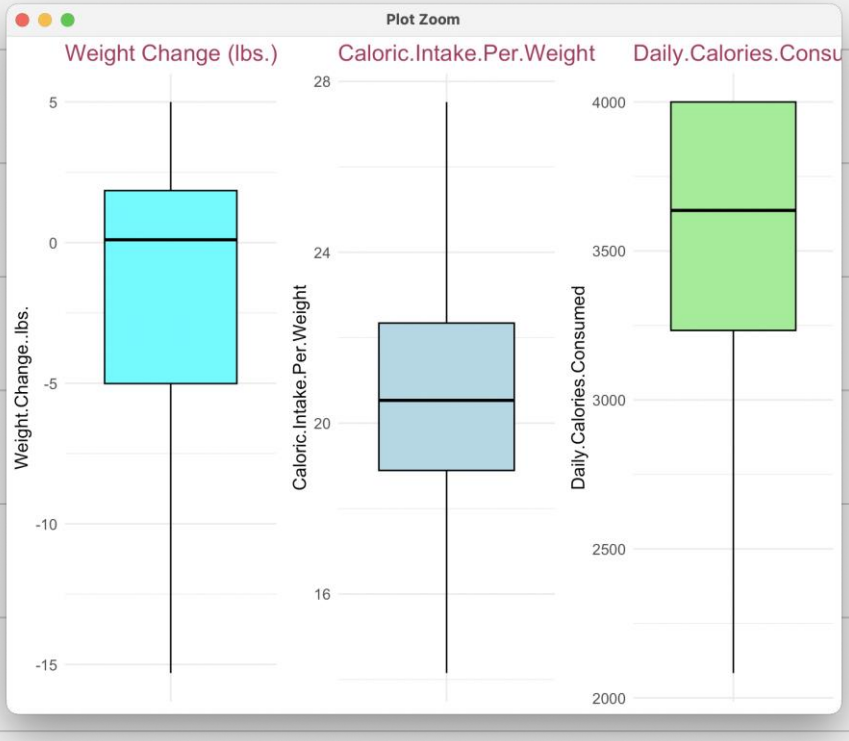
**Handling Outliers:** These column had noticeable outliers, which could distort our regression model. To deal with this, we used the interquartile range (IQR) method to cap the outliers at the acceptable bounds.

# 4.1. Dealing with Outliers

*Steps for Outlier Handling*:

- First, we calculate the first quartile (Q1) and third quartile (Q3) for each column.
- The IQR is computed as the difference between Q3 and Q1.
- Outliers are defined as values below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.
- We then cap the values below the lower bound to the lower bound, and values above the upper bound to the upper bound.

**Outcome**: After applying the capping method [2], we plotted the variables again to confirm that outliers were effectively removed.




The box plot shown above illustrates the targeted columns after handling the outliers, showing a more consistent and representative distribution of the data.

(Osborne & Overbay, 2004, p. 5)

# 4.2. Feature Engineering & Transformation

**1. Feature Selection**

**Purpose:**

Feature selection is the process of identifying and removing variables that are irrelevant, redundant, or highly correlated. The goal is to improve model accuracy, reduce overfitting, and simplify the model by removing features that do not add significant predictive value.

**What we are doing:**

In the dataset, there are both Current.Weight and Final.Weight columns. So, we drop the Final.Weight column because the goal is to predict Weight.Change. Including 'Final Weight' would have made predicting Weight.Change too simple, as it could be calculated directly by subtracting Current.Weight from Final.Weight. This would not provide any additional value to the model and could lead to redundancy. Additionally, using Final.Weight could introduce target leakage, where the model unintentionally gains direct access to information related to the target variable, which could result in overfitting. By removing Final.Weight, we ensure the model focuses on other predictive features and learns meaningful patterns to predict Weight.Change.

To avoid redundancy and ensure the model remains efficient, we drop the Caloric.Intake.Per.Weight feature. This decision was made because it is derived directly from Daily.Calories.Consumed and Current.Weight, and including it would not provide additional value. We also removed Physical.Activity.Level and BMR (Calories) as features, because these are already encapsulated in the Activity.Weighted.Calories feature, thus avoiding duplication and ensuring the model remains lean without losing key information.

We have previously examined the correlation between variables using a heatmap to identify any high correlations between features. Features that are highly correlated (multicollinear) can distort the model and affect its performance. Based on this analysis, we remove the following columns:

# 4.2. Feature Engineering & Transformation

```
# Drop specified columns using base R
df <- df[, !(colnames(df) %in% c('Caloric.Intake.Per.Weight', 'BMR..Calories.',
                                 'Physical.Activity.Level', 'Physical.Activity.MET.Value',
                                 'Final.Weight..lbs.'))]
```

These features were removed due to their high correlation with other variables, which could lead to multicollinearity issues, or because they were deemed irrelevant to the predictive goals of the model.

**Why it's important:**

By addressing multicollinearity and removing redundant variables, we ensure that the model can learn the independent effects of each feature. This step helps improve model interpretability, reduces overfitting, and simplifies the dataset, making the training process more efficient.

**2. Feature Transformation (Encoding Categorical Variables)**

**Purpose**: Encoding categorical variables ensures that they can be used in machine learning models, which generally require numerical inputs.

**What we are doing**:

- The Gender column is converted to a factor (categorical variable) and then encoded as integers (0 for one category, 1 for the other). The - 1 ensures that the encoding starts from 0 instead of 1, making it more standard for machine learning algorithms.
- The Sleep.Quality column, which contains multiple categories such as 'Poor', 'Fair', 'Good', and 'Excellent', is converted into numerical form using label encoding. Each unique level is assigned an integer value based on its quality level, with 'Poor' mapped to 0, 'Fair' to 1, 'Good' to 2, and 'Excellent' to 3.

**Why it's important**: Many machine learning algorithms (like decision trees, linear regression, etc.) require numerical inputs. Encoding helps the model process categorical data efficiently by transforming it into a numerical format that retains the categorical relationships.

# 4.2. Feature Engineering & Transformation

**3. Scaling Data**

**Purpose**: Scaling ensures that all features are on the same scale, which is important for algorithms that rely on distance metrics (e.g., k-NN, SVM) or gradient-based methods (e.g., logistic regression, neural networks).

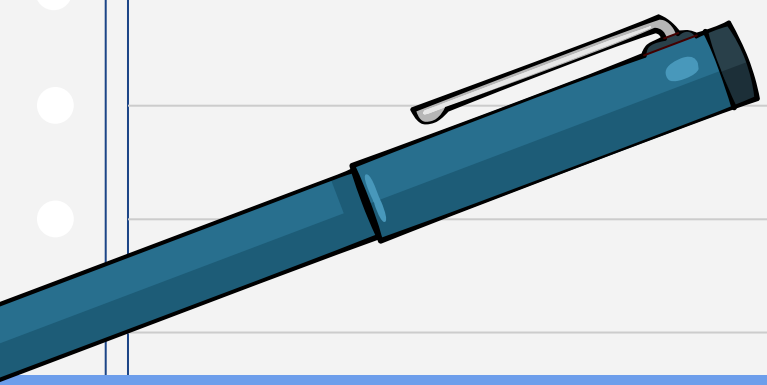
**What we are doing**:

- First, we install and load the caret package, which is a common tool for preprocessing data in R.
- preProcess(df, method = c("center", "scale")) applies two transformations:
  - **Centering**: Subtracts the mean of each feature, ensuring that each feature has a mean of 0.
  - **Scaling**: Divides each feature by its standard deviation, so each feature has a standard deviation of 1.
- The predict(preprocessor, df) applies these transformations to the dataset.

**Why it's important**: Some models (like K-means clustering, neural networks, and gradient-based algorithms) can be sensitive to the scale of the data. Features with larger ranges can dominate the learning process, leading to inaccurate models. Scaling ensures that all features contribute equally.

# 05

# MULTIPLE LINEAR REGRESSION MODEL



# 5.1. Multiple Linear Regression Model:

**Purpose:**

Multiple Linear Regression (MLR) is a statistical technique used to model the relationship between one dependent variable and multiple independent variables. The goal is to understand how the independent variables (predictors) affect the dependent variable and to make predictions based on this relationship.

**What we are doing:**

**Splitting the Data into Training and Testing Sets:**

We allocate 80% of the data for training and the remaining 20% for testing.

**Building the Multiple Linear Regression Model:**

We use the lm() function to fit a linear regression model where Weight.Change..lbs. is the dependent variable, and all other variables in the dataset (~ .) are the independent variables. The model is built using the training data (train_data).

```
# Build the multiple linear regression model
model <- lm(Weight.Change..lbs. ~ ., data = train_data)
```

**Model Summary:**

The summary(model) function provides a detailed summary of the regression model, including coefficients, significance levels, R-squared value, and other diagnostic statistics that help assess the model's performance.

```
Call:
lm(formula = Weight.Change..lbs. ~ ., data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2584 -1.3611  0.3013  1.9317  8.0256

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -2.1108     0.4228  -4.993 4.19e-06 ***
Age                            1.3938     4.8626   0.287    0.775
Gender                        -3.3523     9.9059  -0.338    0.736
Current.Weight..lbs.          -7.3384    24.2784  -0.302    0.763
Daily.Calories.Consumed       12.1085    41.2795   0.293    0.770
Daily.Caloric.Surplus.Deficit -8.5856    29.8119  -0.288    0.774
Duration..weeks.              -0.2713     0.4366  -0.621    0.536
Sleep.Quality                  3.0211     0.4350   6.945 1.58e-09 ***
Stress.Level                  -2.2415     0.4548  -4.928 5.35e-06 ***
Activity.Weighted.Calories     0.1317     0.8714   0.151    0.880
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.763 on 70 degrees of freedom
Multiple R-squared:  0.5378,    Adjusted R-squared:  0.4784
F-statistic: 9.049 on 9 and 70 DF,  p-value: 6.255e-09
```

# 5.1. Multiple Linear Regression Model:

**Model Fit**

The model explains **53.78% of the variance** in weight change ($R^2$), with an adjusted $R^2$ of **47.84%**.

The model is **statistically significant** overall $p<0.001$.

**Significant Key Predictors**

- **Sleep Quality** ($\beta=3.02$, $p<0.001$): Higher sleep quality is associated with an increase in weight change..
- **Stress Level** ($\beta=-2.24$,$p<0.001$): Higher stress levels decrease weight change.

**Non-Significant Key Predictors in this case:** Age, Gender, Current Weight, Caloric Variables, Duration, and Activity-Weighted Calories ($p>0.05$).

To evaluate the model's performance, we calculated the R-squared value, which measures how much of the variability in weight change is explained by the model. The result, an R-squared of approximately 52.61%.

```
> print(paste("R-squared for test data:", R_squared_test))
[1] "R-squared for test data: 0.526071187365508"
```

The moderate R-squared values on both the training set (0.5378) and the test set (0.5261) suggest that the model explains around 53% of the variance in weight change. The similar R-squared values on both the training and test sets suggest that the model generalizes reasonably well and is not overfitting. However, there may still be room for improvement in capturing additional factors that influence weight change. We will consider exploring more features, interactions, or advanced modeling techniques to improve predictive accuracy.

# Diagnostic Plots:

**Purpose:**

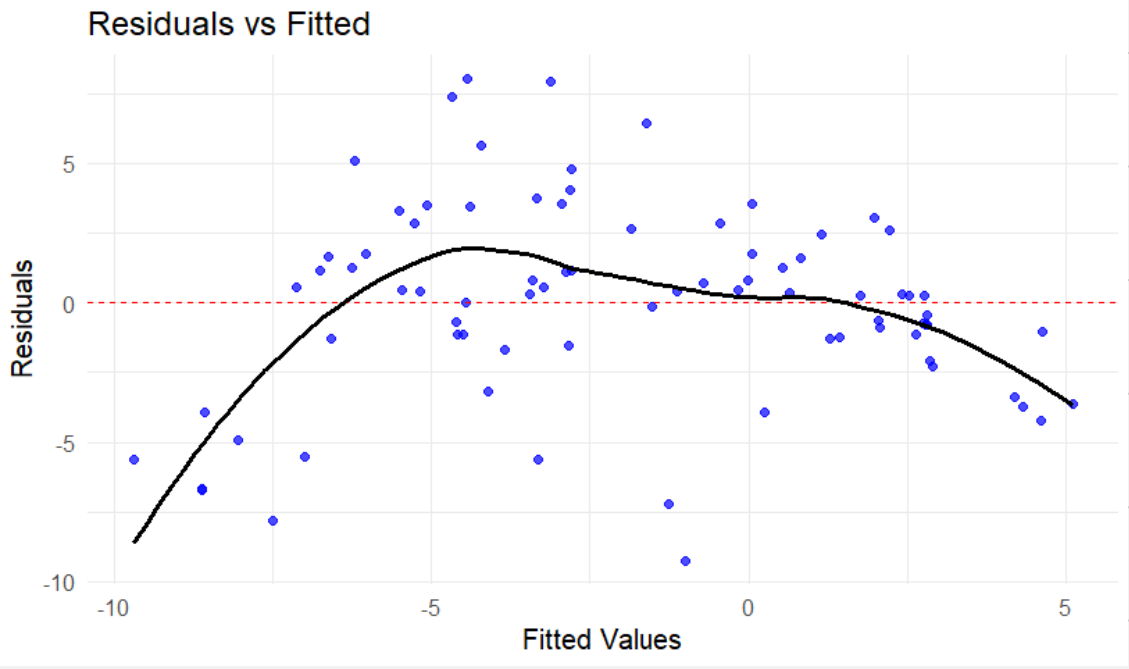
Diagnostic plots are used to assess the validity of the assumptions underlying the multiple linear regression model. These assumptions include linearity, homoscedasticity (constant variance of residuals), normality of residuals, and the absence of influential outliers. Visualizing the residuals helps identify potential problems with the model, such as non-linearity, heteroscedasticity, or non-normality of residuals.

**Extracting Residuals, Fitted Values, and Standardized Residuals:**

- residuals <- resid(model): Extracts the residuals (differences between observed and predicted values) from the fitted regression model.
- fitted_values <- fitted(model): Extracts the fitted values (predicted values) from the model.
- std_residuals <- rstandard(model): Extracts the standardized residuals, which are residuals scaled by their estimated standard deviation.

**Residuals vs Fitted Plot:**

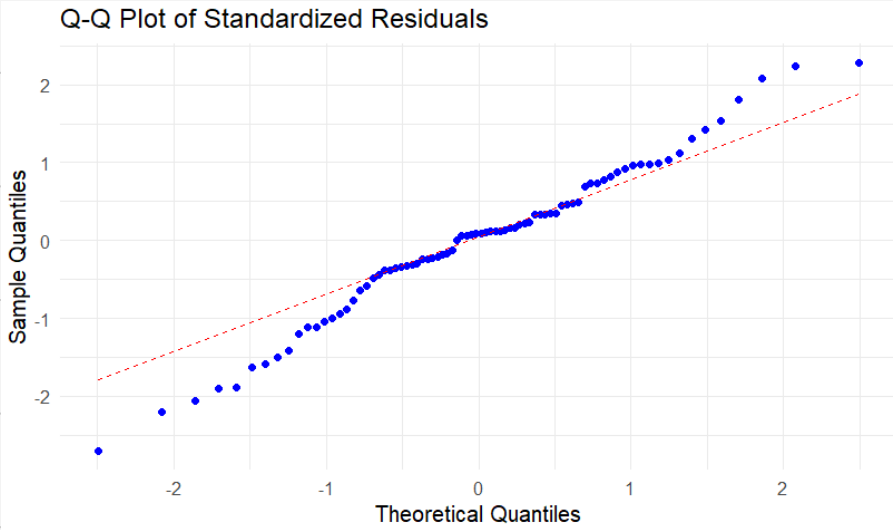
- The first plot shows residuals against fitted values. A random scatter of points around the horizontal line at zero suggests that the model fits the data well. However**, a pattern seems to be visible,** and it may **indicate non-linearity or heteroscedasticity** (unequal variance of residuals).



# Diagnostic Plots:

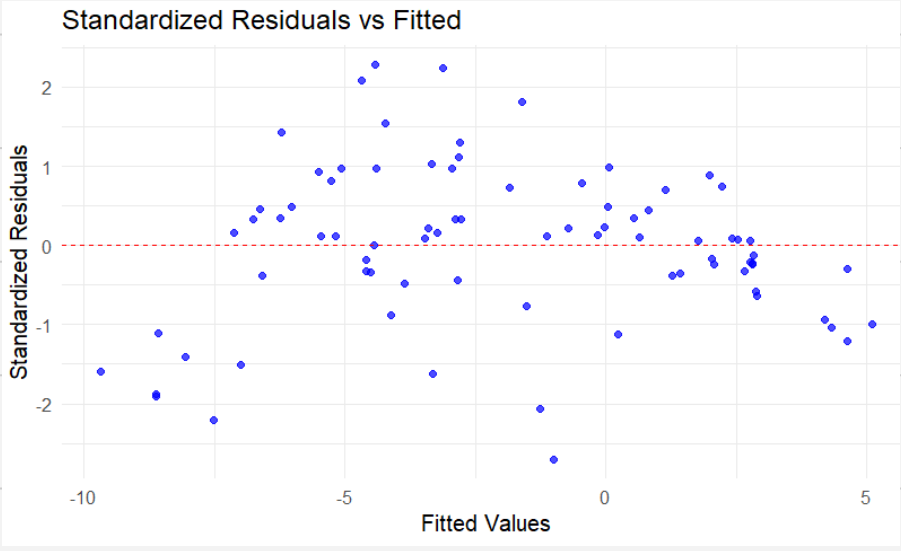
**Q-Q Plot (Quantile-Quantile Plot):**

- The Q-Q plot compares the distribution of the standardized residuals to a normal distribution.
- The residuals fall closely along this line in the middle range, it indicates that the residuals are mostly normally distributed. Deviations from the reference line (especially in the tails) suggest non-normality in the residuals.




**Standardized Residuals vs Fitted Plot:**

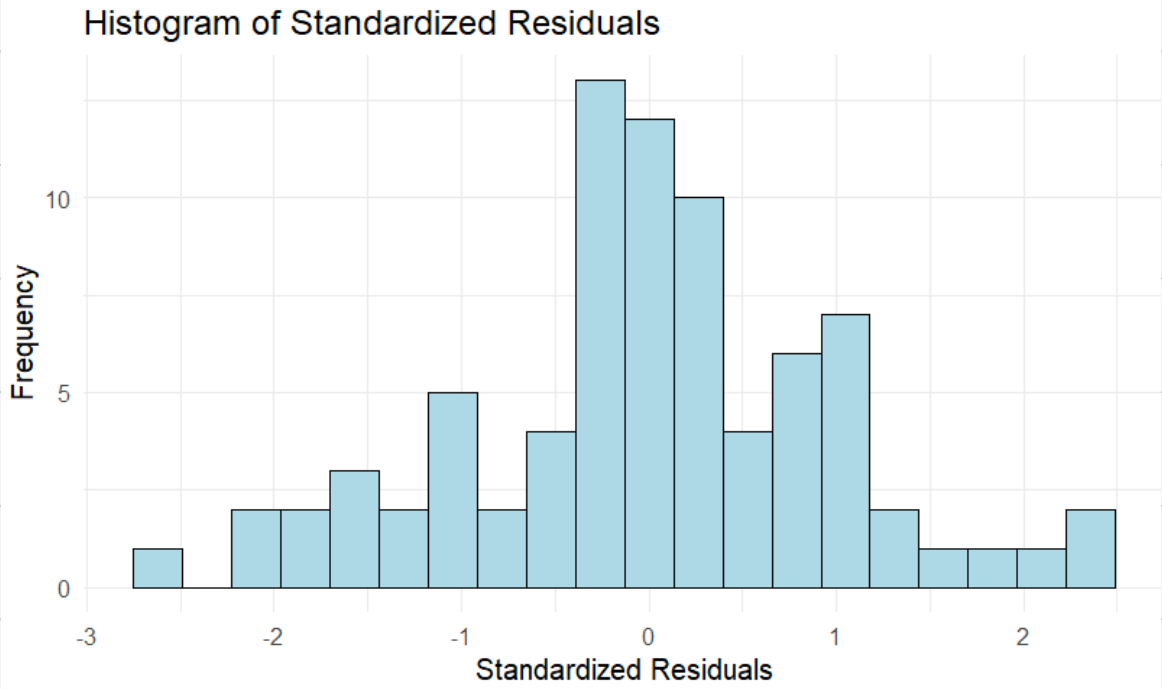
This plot visualizes the standardized residuals against the fitted values, which helps check for heteroscedasticity (non-constant variance). The points should ideally be randomly scattered around zero, without any discernible pattern. A pattern could suggest that the variance of residuals changes with the fitted values, violating the assumption of constant variance.



# Diagnostic Plots:

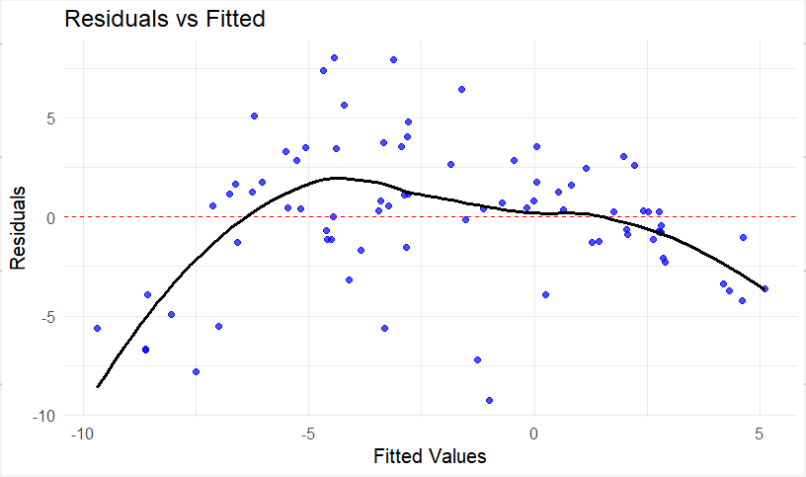
**Histogram of Standardized Residuals:** The histogram of standardized residuals shows that most residuals are centered around zero, indicating no significant bias in predictions. The spread suggests that the model's predictions are reasonably close to the actual values for most data points, with few outliers.

However, the slight left skew indicates a possible deviation from the normality assumption, which could affect the model's performance in certain cases. Overall, the model performs adequately but may benefit from further refinements to address this skew.



# 5.2. Polynomial Regression:

The residuals vs. fitted plot from the initial linear regression model showed patterns such as curvature, indicating the linear model may not adequately capture the relationships in the data.




To address this, we introduced polynomial terms to potentially better fit the data.

**Steps Taken:**

**Polynomial Feature Addition**:

Using a function, polynomial terms (squared and cubic) were added for relevant predictors. This transformation allows the model to capture more complex, non-linear relationships between predictors and the target variable, Weight.Change..lbs..

**Dataset Splitting**:

After adding polynomial terms, the data was split into training and testing sets, ensuring the model's performance can be validated on unseen data.

**Model Building**:

A new multiple linear regression model (model2) was built using both the original and polynomial terms.

```
model2 <- lm(
  Weight.Change..lbs. ~ Age + Gender + Current.Weight..lbs. + Daily.Calories.Consumed +
    Daily.Caloric.Surplus.Deficit + Duration..weeks. + Sleep.Quality + Stress.Level +
    Activity.Weighted.Calories + I(Age^2) + I(Age^3) +
    I(Current.Weight..lbs.^2) + I(Current.Weight..lbs.^3) +
    I(Daily.Calories.Consumed^2) + I(Daily.Calories.Consumed^3) +
    I(Daily.Caloric.Surplus.Deficit^2) + I(Daily.Caloric.Surplus.Deficit^3) +
    I(Duration..weeks.^2) + I(Duration..weeks.^3) +
    I(Sleep.Quality^2) + I(Sleep.Quality^3) +
    I(Stress.Level^2) + I(Stress.Level^3) +
    I(Activity.Weighted.Calories^2) + I(Activity.Weighted.Calories^3),
  data = train_data_poly
)
```

# 5.2. Polynomial Regression:

```
Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                          3.02456    1.45927   2.073   0.0430 *
Age                                  8.00378    5.52887   1.448   0.1535
Gender                             -18.13131   10.79762  -1.679   0.0989 .
Current.Weight..lbs.               -43.33435   26.75220  -1.620   0.1111
Daily.Calories.Consumed             72.47327   45.00589   1.610   0.1132
Daily.Caloric.Surplus.Deficit      -52.08079   32.61272  -1.597   0.1161
Duration..weeks.                    -1.99127    0.94052  -2.117   0.0389 *
Sleep.Quality                        1.42782    1.07541   1.328   0.1899
Stress.Level                        -1.44003    0.87085  -1.654   0.1040
Activity.Weighted.Calories           1.68221    1.72312   0.976   0.3333
I(Age^2)                             0.38297    0.43349   0.883   0.3809
I(Age^3)                             0.28890    0.47214   0.612   0.5432
I(Current.Weight..lbs.^2)            0.16159    0.39636   0.408   0.6851
I(Current.Weight..lbs.^3)            0.02970    0.25928   0.115   0.9092
I(Daily.Calories.Consumed^2)        -1.53271    1.27015  -1.207   0.2328
I(Daily.Calories.Consumed^3)         0.01794    0.51646   0.035   0.9724
I(Daily.Caloric.Surplus.Deficit^2)  -0.30438    0.43258  -0.704   0.4847
I(Daily.Caloric.Surplus.Deficit^3)  -0.43569    0.29694  -1.467   0.1481
I(Duration..weeks.^2)                0.74482    0.46048   1.617   0.1116
I(Duration..weeks.^3)                0.85704    0.49206   1.742   0.0872 .
I(Sleep.Quality^2)                  -4.39690    0.74759  -5.881 2.64e-07 ***
I(Sleep.Quality^3)                   1.65604    0.69956   2.367   0.0215 *
I(Stress.Level^2)                   -2.21379    0.44214  -5.007 6.25e-06 ***
I(Stress.Level^3)                   -0.66379    0.47636  -1.393   0.1692
I(Activity.Weighted.Calories^2)      1.17642    0.66120   1.779   0.0808 .
I(Activity.Weighted.Calories^3)     -0.81304    0.54363  -1.496   0.1406
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.712 on 54 degrees of freedom
Multiple R-squared:  0.8392,    Adjusted R-squared:  0.7648
F-statistic: 11.27 on 25 and 54 DF,  p-value: 1.171e-13
```

**Model Evaluation**:

An R-squared value of 0.8392 indicates that the model explains 84% of the variance in the target variable (in this case, weight change) based on the predictors in the training data. This is generally considered a strong fit, suggesting that the model captures most of the underlying relationships in the data.

```
> print(paste("R-squared for test data:", R_squared_test))
[1] "R-squared for test data: 0.18270155253752"
```

However, the R-squared on the test set is very low (0.18), which suggests the model suffers from **overfitting**.

This occurs because the model becomes overly complex, capturing noise and nuances in the training data that do not generalize to unseen data. To improve performance, simpler models or regularization techniques (e.g., Ridge or Lasso regression) could be explored, or cross-validation could help determine the optimal degree of polynomial features.

# 5.2. Polynomial Regression:

To address multicollinearity, we calculate the Variance Inflation Factor (VIF) for each predictor variable in the model. VIF measures how much the variance of a regression coefficient is inflated due to collinearity with other predictors. If a variable has a high VIF (greater than 10), it suggests that it is highly correlated with other variables and may not provide unique information to the model.

We identify these variables, removes them, and refits the model. After removing the variables with high VIF, the R-squared value for the training set slightly decreases to 0.8097, but it still indicates a strong fit. On the test set, the model is evaluated again using the R-squared metric. The R-squared for test data jumps to 0.40245. The test R-squared indicates how better the model generalizes to new data, compared with the previous model, and removing high VIF variables improves model stability and reduces overfitting.

```
Call:
lm(formula = Weight.Change..lbs. ~ ., data = train_data_cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1343 -1.6247 -0.1103  1.9847  5.7805

Coefficients: (1 not defined because of singularities)
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -0.12625    1.83970  -0.069   0.9455
Duration..weeks.                -1.77756    0.92299  -1.926   0.0589 .
Stress.Level                    -1.49896    0.86800  -1.727   0.0893 .
`Age^2`                          0.51025    0.41019   1.244   0.2184
`Age^3`                         -0.03946    0.17572  -0.225   0.8231
`Gender^2`                       2.04723    1.44880   1.413   0.1628
`Gender^3`                           NA         NA      NA       NA
`Current.Weight..lbs.^2`        -0.14186    0.38046  -0.373   0.7106
`Current.Weight..lbs.^3`         0.01862    0.19493   0.096   0.9242
`Daily.Calories.Consumed^2`     -0.20951    0.92438  -0.227   0.8215
`Daily.Calories.Consumed^3`     -0.03619    0.43037  -0.084   0.9333
`Daily.Caloric.Surplus.Deficit^2` -0.52870  0.40813  -1.295   0.2001
`Daily.Caloric.Surplus.Deficit^3`  0.05595  0.18506   0.302   0.7635
`Duration..weeks.^2`             0.99479    0.44091   2.256   0.0277 *
`Duration..weeks.^3`             0.74979    0.48405   1.549   0.1266
`Sleep.Quality^2`               -4.80874    0.55405  -8.679 3.43e-12 ***
`Sleep.Quality^3`                2.63226    0.24928  10.559 2.61e-15 ***
`Stress.Level^2`                -1.88512    0.42829  -4.401 4.49e-05 ***
`Stress.Level^3`                -0.57927    0.47637  -1.216   0.2287
`Activity.Weighted.Calories^2`   0.72848    0.63190   1.153   0.2535
`Activity.Weighted.Calories^3`  -0.10469    0.31669  -0.331   0.7421
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.799 on 60 degrees of freedom
Multiple R-squared:  0.8097,    Adjusted R-squared:  0.7495
F-statistic: 13.44 on 19 and 60 DF,  p-value: 4.387e-15
```

```
> print(paste("R-squared for test data:", R_squared_test))
[1] "R-squared for test data: 0.402453226194991"
```

# 5.3. Stepwise Regression:

In this analysis, stepwise regression is applied to simplify the model by reducing the number of predictors. The goal is to create a more efficient and interpretable model by automatically adding or removing predictors based on their statistical significance. This process helps reduce complexity, potentially improving generalization to new data and mitigating overfitting.

After performing stepwise regression, the model's performance on the training set shows an R-squared of 0.8081, indicating that the simplified model still explains a substantial amount of variance. However, when tested on the unseen test set, the R-squared drops significantly to 0.3703, suggesting poor generalization.

```
Call:
lm(formula = Weight.Change..lbs. ~ Duration..weeks. + Stress.Level +
    `Age^2` + `Gender^2` + `Daily.Caloric.Surplus.Deficit^2` +
    `Duration..weeks.^2` + `Duration..weeks.^3` + `Sleep.Quality^2` +
    `Sleep.Quality^3` + `Stress.Level^2` + `Stress.Level^3` +
    `Activity.Weighted.Calories^2`, data = train_data_cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1033 -1.5717 -0.0836  1.9114  6.0125

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      -0.3427     1.4710  -0.233   0.8165
Duration..weeks.                 -1.7948     0.8327  -2.155   0.0347 *
Stress.Level                     -1.5064     0.7855  -1.918   0.0594 .
`Age^2`                           0.5250     0.3469   1.513   0.1349
`Gender^2`                        2.2857     1.2192   1.875   0.0652 .
`Daily.Caloric.Surplus.Deficit^2` -0.5796     0.2836  -2.044   0.0449 *
`Duration..weeks.^2`              0.9131     0.3497   2.611   0.0111 *
`Duration..weeks.^3`              0.7639     0.4123   1.853   0.0683 .
`Sleep.Quality^2`                -4.8265     0.5115  -9.435 6.39e-14 ***
`Sleep.Quality^3`                 2.6249     0.2260  11.615  < 2e-16 ***
`Stress.Level^2`                 -1.8736     0.3871  -4.840 7.98e-06 ***
`Stress.Level^3`                 -0.5898     0.4334  -1.361   0.1781
`Activity.Weighted.Calories^2`    0.4729     0.3080   1.536   0.1294
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.66 on 67 degrees of freedom
Multiple R-squared:  0.8081,    Adjusted R-squared:  0.7737
F-statistic: 23.51 on 12 and 67 DF,  p-value: < 2.2e-16
```

```
> print(paste("R-squared for test data:", R_squared_test))
[1] "R-squared for test data: 0.370383321609544"
```

After performing stepwise regression, the selected variables are extracted from the model, and the dataset is updated to include only those variables. This updated dataset is then split into training and testing sets, ensuring that the model is trained on the optimized subset of predictors. By focusing on the most important features, this approach aims to improve model accuracy, reduce overfitting, and enhance the model's generalizability to new data.

# 5.4. Ridge and Lasso Regression:

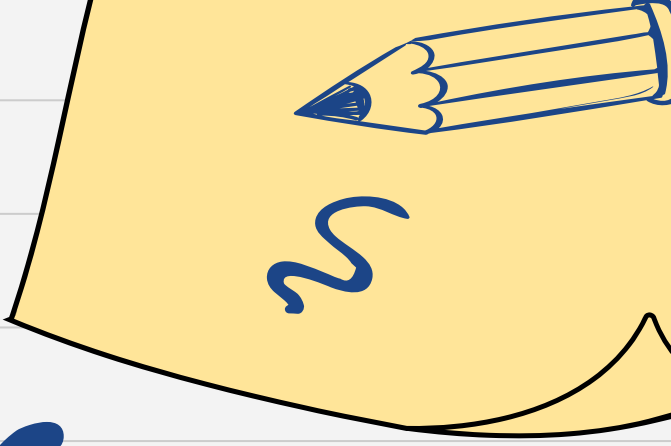
In this part, the goal is to improve the model's generalization ability and prevent overfitting by through **regularization techniques** using **Ridge and Lasso regression**.

**Ridge and Lasso Regression using glmnet package**:

- The cv.glmnet() function fits both Ridge (with alpha = 0) and Lasso (with alpha = 1) regression models with cross-validation to find the optimal penalty (regularization) parameter, lambda.min.
- Lasso regression, or L1 regularization, applies a penalty equal to the absolute value of the coefficients, forcing some to zero. This results in automatic feature selection, eliminating irrelevant predictors and simplifying the model. Lasso is particularly effective for high-dimensional data, improving model interpretability and performance.
- Ridge regression, or L2 regularization, adds a penalty to the loss function proportional to the square of the coefficients. This shrinks the coefficients, reducing multicollinearity and overfitting while keeping all features in the model. It stabilizes the model and improves generalization by limiting the impact of large coefficients.

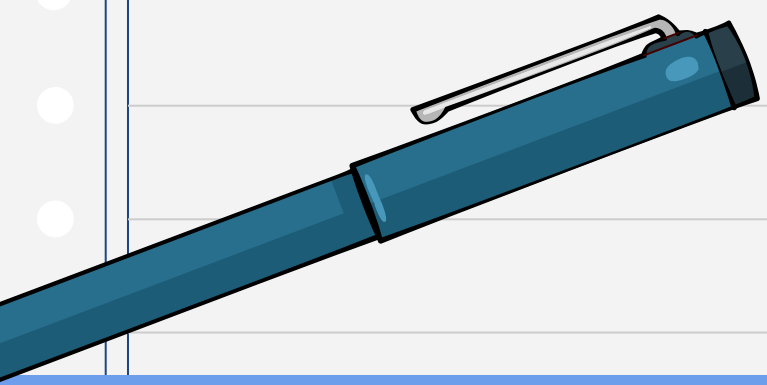
**Model Performance:**

- Predictions are made on the test dataset (x_test) using the best lambda.min value for each model.
- ```
  > cat("R-squared on Test Set:", r_squared, "\n")
  R-squared on Test Set: 0.4568907
  ```
  ng it explains **45.68%** of the variance in the target variable (weight change). This is a moderate improvement over earlier models, suggesting that the regularization approach has enhanced the model's ability to generalize to new data. However, there is still considerable room for improvement, as a significant portion of the variance remains unexplained.
  ```



# 06

# XGBOOST



# 6.1. XGBoost:

This section describes the development and evaluation of an XGBoost model to predict the variable Weight Change (lbs). **The XGBoost model was selected for this task due to its robustness in handling target variables without requiring transformation, which is advantageous in real-world contexts where the target variable may not follow a normal distribution [3].** XGBoost, a tree-based, gradient-boosting algorithm, is known for its capacity to capture complex patterns and relationships in the data without sensitivity to distributional assumptions [4].

**1. XGBoost Model**:

- The model was built using train_data **without polynomial terms or variable selection**, emphasizing simplicity and computational efficiency while leveraging the strength of XGBoost in capturing inherent data patterns.
- The data is first converted into a matrix format that XGBoost can work with (xgb.DMatrix), and then the model is trained with 5 rounds (iterations).
- After training, the model's performance is evaluated on both the **training set** and the **test set**.

**2. Training Performance**:

```
> cat("R-squared on Training Data:", r_squared_train, "\n")
R-squared on Training Data: 0.9174672
```

The model achieved an **R-squared of 0.917** on the training set, demonstrating that it explains **91.7%** of the variance. This result indicates that XGBoost effectively identified patterns in the training data.

**3. Testing Performance**:

```
> cat("R-squared on Test Set:", r_squared, "\n")
R-squared on Test Set: 0.6563695
```

The **R-squared value** for the test set is **0.656**, explaining **65.6%** of the variance in unseen data. Although there is a noticeable drop from the training performance, this is common when generalizing to new data. The model's moderate generalization performance reflects a balance between capturing meaningful patterns and avoiding overfitting.

(Chen & Guestrin, 2016, p. 785)
(Friedman, 2001, p. 1192)

# 6.1. XGBoost:

**4. Hyperparameter Tuning**:

To enhance the model's performance, hyperparameter tuning was conducted using a **grid search approach**. The goal was to fine-tune the XGBoost model to **improve its predictive accuracy on the test dataset**, reducing overfitting and underfitting. This was achieved by systematically varying key parameters and evaluating their impact on the model's R-squared performance.

**Key hyperparameters tuned:**

- Learning Rate (eta): 0.01, 0.1, 0.3 for balanced step size and stable convergence.
- Max Depth (max_depth): 3, 5, 7 to control tree complexity and overfitting.
- Regularization (gamma): 0, 1, 5 to constrain splits and reduce overfitting.
- Column Subsampling (colsample_bytree): 0.5, 0.7, 1 for feature randomness and robustness.
- Min Child Weight (min_child_weight): 1, 3, 5 to balance flexibility and split control.
- Subsample (subsample): 0.7, 0.8, 1 to prevent overfitting by sampling data.

**Process and Result**:

After running 10 boosting rounds, the optimal parameters were:

- eta = 0.3
- max_depth = 7
- gamma = 0
- colsample_bytree = 0.7
- min_child_weight = 3
- subsample = 0.8

```
Best Parameters:
> print(best_params)
$eta
[1] 0.3

$max_depth
[1] 7

$gamma
[1] 0

$colsample_bytree
[1] 0.7

$min_child_weight
[1] 3

$subsample
[1] 0.8
```

# 6.1. XGBoost:

**4. Hyperparameter Tuning**:

**Process and Result**:

This configuration achieved an **R-squared of 0.8219 on the test set**, a significant improvement over the baseline model.

```
> cat("Best R-squared:", best_r2, "\n")
Best R-squared: 0.8219138
```

**Final Model Training:**

The model was retrained using these optimal parameters with **10 rounds**. The final R-squared on the test set was **0.9728**, indicating the model maintained strong generalization capability with slightly reduced variance explanation compared to the hyperparameter search but balanced against overfitting risks.

```
> cat("R-squared on Training Data:", r_squared_train, "\n")
R-squared on Training Data: 0.9728908
```

**Discussion**:

Hyperparameter tuning significantly improved model performance, raising the R-squared in the test set from **0.656** in the baseline model to **0.82** in the tuned model. This demonstrates the effectiveness of parameter optimization in enhancing prediction accuracy. The final model training R-squared of **0.97** reflects a well-generalized model that balances training accuracy with real-world applicability. Further refinements could explore more parameter combinations or advanced tuning techniques if computational resources allow.

## 6.2. Bootstrap Estimation of R-squared:

The bootstrap method is a statistical resampling technique used to estimate the distribution of a metric by repeatedly resampling the data with replacement [5]. In this case, the bootstrap method is applied to **estimate the R-squared** of the XGBoost model. This approach helps to understand the variability and reliability of the model's performance, providing a confidence interval for the R-squared value.

**Steps taken**:

**Initialization**:

Set 1,000 bootstrap resamples and initialize a vector to store R-squared values.

**Bootstrap Sampling & Model Training**:

In each iteration, create a bootstrap sample by randomly selecting rows from the training data with replacement. Train the XGBoost model on this sample using the optimized hyperparameters (best_params).

**Out-of-Bag (OOB) Evaluation**:

Evaluate the model on OOB samples (data not included in the bootstrap sample) to simulate prediction on unseen data.

**R-squared Calculation**:

Compute R-squared for each OOB prediction, measuring the model's ability to explain variance in unseen data.

**Aggregation of Results**:

The mean R-squared value across all bootstrap samples is computed, along with a 95% confidence interval, which represents the range within which the true R-squared value likely falls.

(Efron, 1979, p. 2)

# 6.2. Bootstrap Estimation of R-squared:

**Discussion of Results**:

- **Bootstrap Mean R-squared**: The bootstrap process produced a mean R-squared value of **0.972**, indicating that the model explains approximately 90.8% of the variance in the target variable across various resampled datasets.

```
> cat("Bootstrap mean R-squared:", mean_r_squared, "\n")
Bootstrap mean R-squared: 0.9724859
```

- **95% Confidence Interval**: The confidence interval for R-squared is **[0.900, 0.994]**, suggesting that in 95% of resampled cases, the R-squared value lies within this range. This demonstrates that the model's performance is consistently high and stable across different subsets of the data.

```
> cat("95% confidence interval for R-squared:", lower_ci, "-", upper_ci, "\n")
95% confidence interval for R-squared: 0.9004826 - 0.9947817
```

- **Implications**: The narrow confidence interval and high mean R-squared indicate that the model is robust and generalizes well across different data samples. This enhances confidence in the reliability of the model when applied to unseen data.

The bootstrap method provides a solid estimate of model performance variability. With a high mean R-squared and a narrow confidence interval, it confirms the model's effectiveness and reliability in predicting weight change.

# 6.3. Jackknife Resampling for Feature Importance Estimation:

Jackknife resampling is a statistical technique used to estimate the stability and reliability of a model's feature importance by systematically excluding one observation at a time from the training dataset. This method is particularly useful for evaluating the variance and potential bias in feature importance scores, offering deeper insights into how each feature contributes to model predictions under different data scenarios [6]. In this context, Jackknife resampling is applied to the XGBoost model to assess the robustness of feature importance by calculating bias and standard error, which enhances the interpretability and reliability of the model's results.

**Result Discussion:**

- **Bias of Feature Importance:** The bias values show small deviations from the feature importance derived from the full dataset, indicating that the feature rankings are relatively stable.

```
Bias of Feature Importance:
> print(bias_importance)
[1] -0.019090340 -0.053728964 -0.016185463  0.060044580
0.015685157 -0.008676456  0.016782035
[8]  0.003142152  0.002027299
```

- **Standard Error of Feature Importance:** The standard error values suggest moderate variability in feature importance, reflecting the confidence in their significance. Features with lower standard errors are more reliable for interpretation.

```
Standard Error of Feature Importance:
> print(jackknife_se_importance)
[1] 3.201688 2.147752 1.420485 1.259037 1.426809 1.516425 1.454199
1.444511 1.516405
```

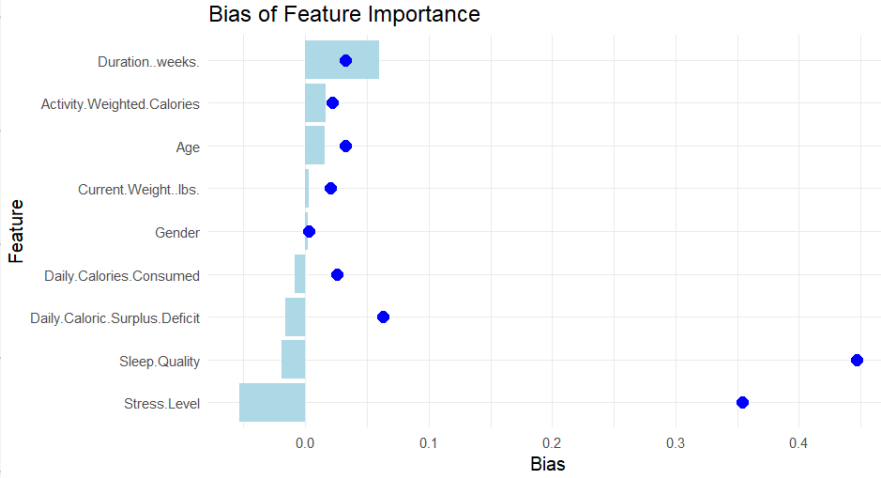
Additionally, the bias and standard error of feature importance are visualized in the 2 plots below.

(Tukey, 1958, p. 615)

# 6.3. Jackknife Resampling for Feature Importance Estimation:

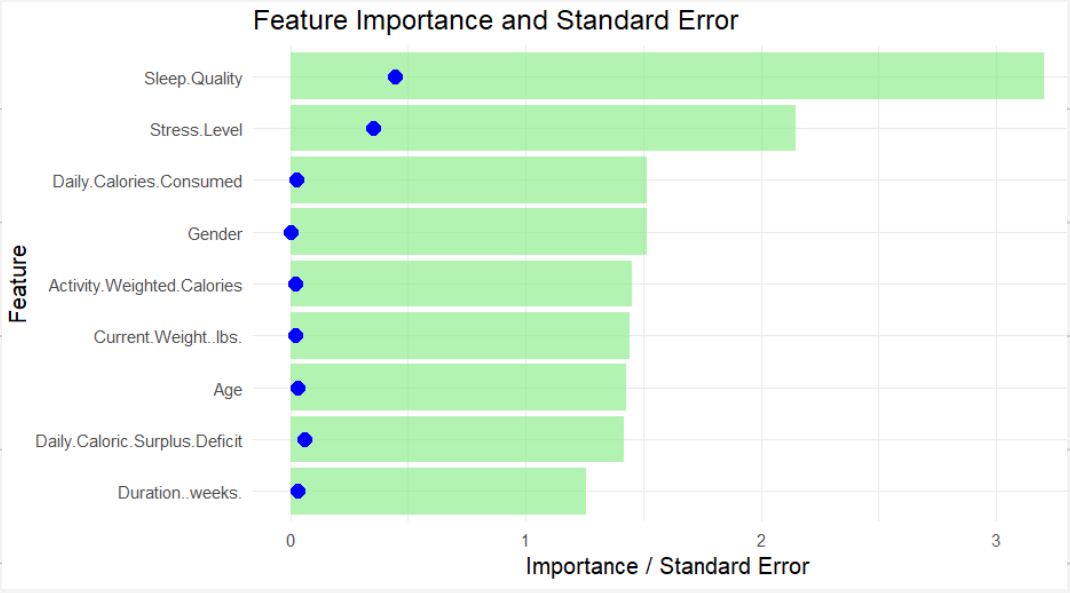
**Bias of Feature Importance:**

The bias plot shows that most features have minimal bias, indicating that their importance rankings are stable across different resampling scenarios. Notably, "Stress Level" exhibits the highest bias, suggesting it is sensitive to data variations.



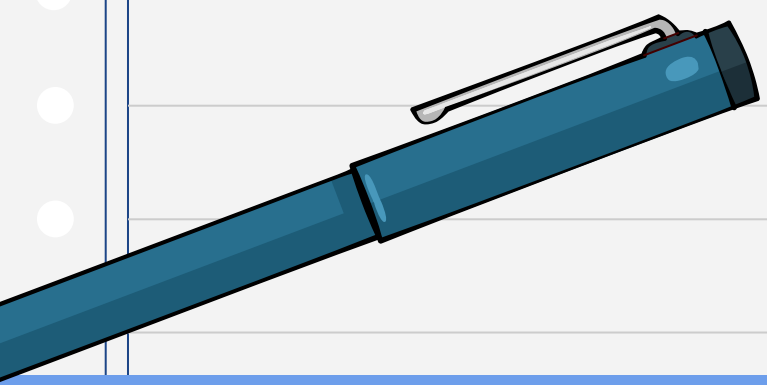

**Feature Importance and Standard Error:**

The plot indicates that "Sleep Quality" and "Stress Level" have the highest importance with relatively low standard errors, making them reliable for interpretation. Other features show moderate variability, reflecting a balance between importance and confidence.



# 07

# MONTE CARLO SIMULATION



# 7. Monte Carlo Simulation for Predicting Weight Change:

Monte Carlo simulation is a statistical technique used to estimate the uncertainty and variability in model predictions by running repeated simulations with random variations in input data [7]. In this context, it is used to predict potential weight change based on various health and lifestyle factors. The purpose of this Monte Carlo simulation is to provide a probabilistic estimate of weight change, considering the inherent variability in key factors such as calorie intake, stress level, and sleep quality. This approach enhances model reliability by accounting for real-world randomness.

**Steps taken:**

**Monte Carlo Simulation:**

- The monte_carlo_simulation function introduces randomness to input variables (e.g., calorie intake and stress level) and runs 1,000 simulations.
- Each simulation involves preprocessing the modified data and predicting weight change.

**Result Aggregation:**

- The predict_pipeline_mc function calculates the mean predicted weight change and a 95% confidence interval (CI) based on the simulation results.
- The mean prediction represents the average expected weight change, while the CI provides a range of possible outcomes.

(Metropolis & Ulam, 1949, p. 336)

# 7. Monte Carlo Simulation for Predicting Weight Change:

**Simulation Results:**

With the provided sample data:

```
> # View the sample data frame
> print(sample_data)
  Age Gender Current.Weight..lbs. Daily.Calories.Consumed Daily.Caloric.Surplus.Deficit
1  56      F                  107                    1399                           320
  Duration..weeks. Sleep.Quality Stress.Level Activity.Weighted.Calories
1               10          Good            2                      316
```

The Monte Carlo simulation was applied to predict the potential weight change based on the given inputs. After running the preprocessor and model (xgb_model_full), the simulation results were as follows:

- **Mean Prediction:** The expected weight change is approximately **-0.61 lbs**, indicating a likely decrease in weight.
- **95% Confidence Interval:** The interval **ranges from -2.40 lbs to 0.098 lbs**, reflecting potential weight loss or gain. This range highlights the variability in the model's predictions, showing that weight outcomes can differ depending on lifestyle factors.

Overall, the Monte Carlo simulation offers a more nuanced understanding of weight change predictions, allowing for uncertainty in real-life scenarios. This approach ensures that decision-making accounts for potential fluctuations in key health indicators.

# 08

# CONCLUSION

# Conclusion

This project aimed to develop a robust predictive model for weight change, considering key factors such as calorie intake, sleep quality, and stress levels. We began with a thorough exploration of the dataset, inspecting key attributes and identifying any potential issues with outliers. Data preprocessing included feature engineering and transformations to ensure the model could effectively capture the relationships between input variables and weight change.

We applied various modeling techniques, starting with Multiple Linear Regression to establish a baseline, and then enhanced model performance with Polynomial Regression and Stepwise Regression to identify the most significant predictors while mitigating the risk of overfitting. Regularization techniques, such as Ridge and Lasso Regression, helped address the overfitting issue, ensuring a more generalizable model.

To better handle the skewed data distribution, we leveraged XGBoost, which significantly improved the model's predictive performance, achieving high R-squared values of 0.9729 on the training set and 0.8219 on the test set. Furthermore, we used Bootstrap Estimation to validate the model's reliability and Jackknife Resampling for assessing feature importance.

A key feature of the project was the use of Monte Carlo simulations to account for uncertainty in the predictions, providing a probabilistic estimate of weight change. The implementation of an automated pipeline allowed for easy scalability, making the model adaptable to new data and ensuring its practical application in real-world scenarios.

Overall, this project demonstrates the power of combining multiple regression techniques, advanced machine learning algorithms, and probabilistic simulations to build a reliable predictive model. Moving forward, improvements in dataset size, input accuracy, and further fine-tuning of model parameters could enhance the model's robustness and predictive power.

# 09

# REFERENCES

# References


[1] Abdullah, A. (2020). Comprehensive weight change prediction dataset. Kaggle. https://www.kaggle.com/datasets/abdullah0a/comprehensive-weight-change-prediction/data

[2] Osborne, J. W., & Overbay, A. (2004). The power of outliers (and why researchers should ALWAYS check for them). Practical Assessment, Research, and Evaluation, 9(1), 1–12.

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16), 785–794.

[4] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.

[5] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. The Annals of Statistics, 7(1), 1–26.

[6] Tukey, J. W. (1958). Bias and confidence in not-quite large samples. Annals of Mathematical Statistics, 29(2), 614–623.

[7] Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. Journal of the American Statistical Association, 44(247), 335–341.

# THE END

Thank you for reading our report. We appreciate your time and consideration!