Week 8 B

Wali Reheman

2024-10-09

Dataset

We will use the **NHANES** dataset, which contains health and demographic information collected from a representative sample of the U.S. population. The dataset is available in R through the `NHANES` package.

Accessing the Dataset:

```
#install.packages("NHANES")  # Install if not already installed
library(NHANES)
data <- data.frame(NHANES)
```

TASK 1: Data Exploration and Cleaning

- Load the `NHANES` dataset.
- Check for **missing values** and handle them appropriately.
- Summarize key variables: `Age`, `Gender`, `BMI`, `SmokeNow` (whether the person currently smokes), `PhysActive`(physical activity status) and `SleepHrsNight`, recode them if it is necessary.
- Identify and remove **outliers** in `BMI` (e.g., using the IQR method).
- **Transform** `BMI` using log transformation if you think it is appropriate.
- Recode `SmokeNow` and `PhysActive` to have more interpretable categories if necessary.

```
library(tidyverse)
data_clean <- data %>%
  drop_na(Age, BMI, Gender, SmokeNow, PhysActive)
# Remove outliers
Q1 <- quantile(data_clean$BMI, 0.25, na.rm = TRUE)
Q3 <- quantile(data_clean$BMI, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
data_no_outliers <- data_clean %>%
  filter(BMI >= (Q1 - 1.5 * IQR) & BMI <= (Q3 + 1.5 * IQR))

# Log transformation (if distribution is skewed)
data_no_outliers <- data_no_outliers %>%
  mutate(log_BMI = log(BMI))
```

Dealing with Missing Data

When working with datasets like NHANES, missing data can be common. We need to handle missing data because missing values can distort analysis results, lead to biased conclusions, and reduce the dataset size. There are different ways to deal with missing data:

- **Remove Missing Data:** The simplest method is to remove rows with missing values for important variables. This method works when the amount of missing data is small and doesn't significantly affect the dataset's size. However, if there are many missing values, removing them may lead to losing too much data, which can reduce the statistical power of the analysis.
- **Imputation (Advanced):** This is when missing values are filled in with estimates, such as the mean, median, or more complex statistical methods. We're not using this here but it's useful when you have larger portions of missing data.

In our project, we chose to **drop rows with missing data** for important variables (Age, BMI, Gender, SmokeNow, PhysActive) because the dataset is large enough, and this ensures that our results won't be biased by incomplete records.

Dealing with Outliers

Outliers are data points that differ significantly from other observations. They can distort statistical results and lead to incorrect conclusions if not properly addressed. Here's how we deal with outliers in this project:

1. **Why Outliers Matter:** Outliers can affect the mean and standard deviation, inflate regression coefficients, and lead to misleading results. Therefore, identifying and managing outliers is important to ensure that the results of the analysis are robust.
2. **Identifying Outliers:** One common way to detect outliers is by using the Interquartile Range (IQR) method:
    - The **IQR** is the range between the 1st quartile (Q1) and the 3rd quartile (Q3) of the data.
    - Outliers are defined as any values below **Q1 - 1.5 * IQR** or above **Q3 + 1.5 * IQR**. In our case, we used this method to identify outliers in BMI.
3. **Handling Outliers:** Once we've identified the outliers, there are a few ways to handle them:
    - **Remove the Outliers:** If the outliers are errors or extreme values that don't represent the population we're studying, we can remove them. This helps make the data more representative and reduces the chance of the outliers influencing the analysis disproportionately.
    - **Transform the Data:** In some cases, instead of removing outliers, we can transform the data to minimize their effect. For example, a log transformation can reduce the skewness caused by outliers, as we did for BMI if needed.

In this project, we chose to **remove outliers** from the BMI variable to ensure that our regression models and visualizations are not overly influenced by extreme BMI values that may not represent the general population. By doing this, we can obtain more reliable and meaningful results.

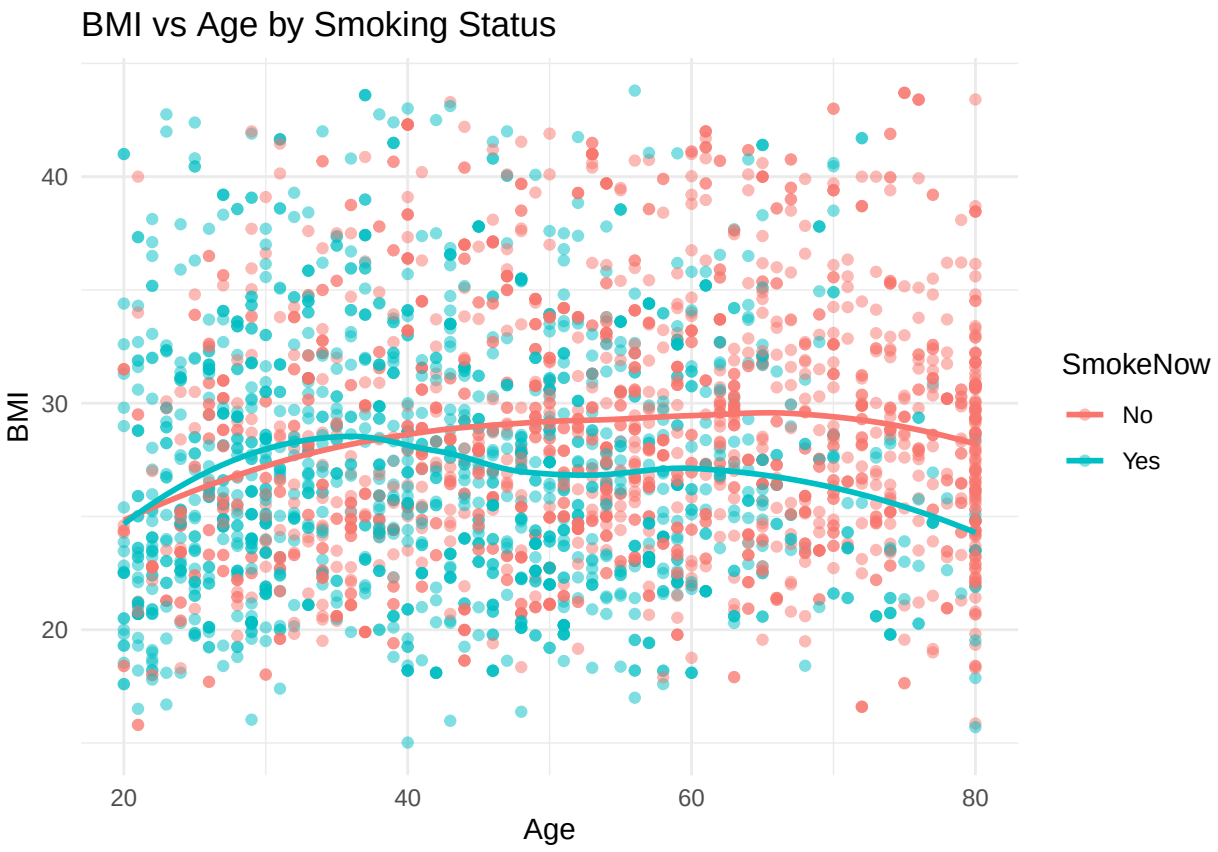
TASK 2: Data Visualization

- Create a **scatter plot** of `BMI` vs. `Age`.
- Color the points by `SmokeNow`.
- Use **smooth lines** to show trends (e.g., using `geom_smooth`).

- Customize the plot with **themes** and **labels**.

```
ggplot(data_no_outliers, aes(x = Age, y = BMI, color = SmokeNow)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE) +
  theme_minimal() +
  labs(title = "BMI vs Age by Smoking Status",
       x = "Age",
       y = "BMI")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



TASK 3: Descriptive Analysis

- Calculate **correlation coefficients** between `Age`, `BMI`, and `SleepHrsNight` (average hours of sleep per night, if available).
- Present the **correlation matrix**.

```
# Ensure 'SleepHrsNight' is available and has no missing values
data_no_outliers <- data_no_outliers %>%
  drop_na(SleepHrsNight)

correlation_matrix <- data_no_outliers %>%
```

```
  select(Age, BMI, SleepHrsNight) %>%
  cor()
print(correlation_matrix)
```

```
##                      Age         BMI SleepHrsNight
## Age           1.00000000  0.09484044    0.10211937
## BMI           0.09484044  1.00000000   -0.04470914
## SleepHrsNight 0.10211937 -0.04470914    1.00000000
```

TASK 4: Multiple Regression Analysis

Run multiple regression models:

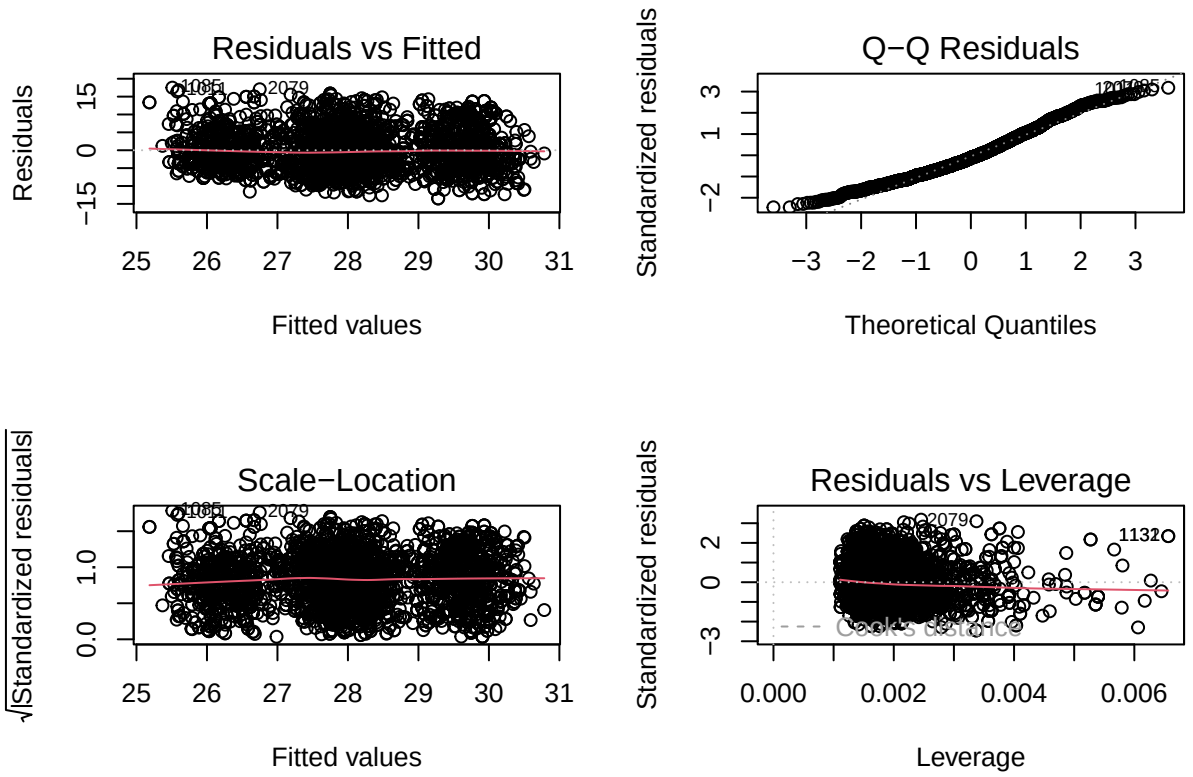
- **Model 1**: `BMI ~ Age`
- **Model 2**: Add `Gender` as an independent variable.
- **Model 3**: Include `PhysActive` , `SmokeNow` and `SleepHrsNight` as independent variables.

```
model1 <- lm(BMI ~ Age, data = data_no_outliers)
model2 <- lm(BMI ~ Age + Gender, data = data_no_outliers)
model3 <- lm(BMI ~ Age + Gender + PhysActive + SmokeNow +SleepHrsNight, data = data_no_outliers)
```

TASK 5: Model Diagnostics

- Check the **assumptions** of linear regression for `model3`:
  - **Linearity**
  - **Independence**
  - **Homoscedasticity**
  - **Normality of residuals**

```
# Residual plots
par(mfrow = c(2, 2))
plot(model3)
```



Four Assumptions of Linear Regression

Linear regression is a powerful tool, but for it to work well, four key assumptions must hold:

1. **Linearity:** This assumes that the relationship between the independent variable(s) and the dependent variable (BMI, in this case) is linear. If this assumption is violated, the model won't accurately capture the relationship. We check this by looking at scatter plots or residual plots, where the data points should align in a linear pattern.

2. **Independence:** This means that the observations (rows of data) are independent of each other. In other words, one person's data shouldn't influence another's. If independence is violated (for example, if the data includes repeated measures for the same individual), the results may be skewed.

3. **Homoscedasticity:** This means that the variability of the residuals (differences between predicted and actual values) is consistent across all levels of the independent variables. If this assumption is violated (heteroscedasticity), the model may give too much importance to certain observations. We check this by looking at a plot of residuals versus predicted values.

4. **Normality of Residuals:** This means that the residuals (errors) of the model should be normally distributed. This assumption can be checked by looking at a histogram or Q-Q plot of the residuals. If the residuals aren't normal, the p-values from the regression model might be unreliable.

By checking and ensuring these assumptions hold, we can be more confident that the results from our regression models are reliable and valid for interpretation.

Checking Four Assumptions of Linear Regression

1. Linearity (Residuals vs Fitted Plot)

- The first plot (Residuals vs Fitted) helps us assess the linearity assumption. Ideally, the red line in the plot should be horizontal and the residuals should be randomly scattered around the line. If there is a clear pattern (e.g., curved or funnel shape), the linearity assumption is likely violated.
- **What to look for:**
  - A random scatter of residuals indicates that the linearity assumption holds.
  - If a clear trend or curve is visible, it suggests a non-linear relationship, and we may need to transform variables or try a non-linear model.

2. Independence (Residuals vs Fitted Plot)

- The same Residuals vs Fitted plot also helps check for independence. If residuals are randomly scattered without showing patterns (like clustering or waves), the independence assumption is likely met.
- **What to look for:**
  - No distinct patterns or grouping in the residuals suggests independence.
  - If there is clustering, autocorrelation, or waves in the residuals, the independence assumption might be violated. This could happen, for instance, if data points are not independent (e.g., repeated measures for the same individuals).

3. Homoscedasticity (Scale-Location Plot)

- The second plot, Scale-Location (also called Spread-Location), helps assess homoscedasticity (constant variance). The red line should be roughly horizontal, and the points should be evenly spread out along the line.
- **What to look for:**
  - If the residuals are evenly spread out and form a horizontal line, this indicates homoscedasticity (constant variance).
  - If the points spread out in a funnel shape (narrow at one end, wider at the other), this suggests heteroscedasticity (non-constant variance). In this case, we might need to transform variables or use robust standard errors.

4. Normality of Residuals (Normal Q-Q Plot)

- The third plot is the Normal Q-Q plot, which helps assess if the residuals are normally distributed. The points should lie on or close to the diagonal line.
- **What to look for:**
  - If the points fall on or very close to the straight diagonal line, it indicates that the residuals are normally distributed.
  - If the points deviate substantially from the line, especially at the tails, it suggests that the residuals are not normally distributed. If the residuals are not normally distributed, transformations or non-parametric methods may be necessary.

Checking Four Assumptions of Linear Regression

5. Residuals vs Leverage (for detecting influential points)

- The fourth plot, Residuals vs Leverage, helps identify influential data points that have a strong effect on the model. The Cook's distance lines help flag these points.
- **What to look for:**
  - Points within Cook's distance lines suggest no influential data points.
  - If points fall outside the Cook's distance lines, they may be influential and disproportionately affect the model. In this case, we may consider removing or investigating these points further.

TASK 6: Presenting Results

- Use `stargazer` to **compare all four models** in a single table with **robust standard errors**.
- Use `coefplot` to **visualize and compare** coefficients across models.

```
library(stargazer)
library(sandwich)  # For robust standard errors
stargazer(model1, model2, model3, type = "text",
          title = "Regression Models Comparison",
          se = list(
            sqrt(diag(vcovHC(model1, type = "HC1"))),
            sqrt(diag(vcovHC(model2, type = "HC1"))),
            sqrt(diag(vcovHC(model3, type = "HC1")))
          ))
```

```
##
## Regression Models Comparison
## ================================================================================================
##                                              Dependent variable:
##                     ---------------------------------------------------------------------------
##                                                      BMI
##                                (1)                   (2)                    (3)
## ------------------------------------------------------------------------------------------------
## Age                          0.032***              0.032***               0.009
##                              (0.006)               (0.006)               (0.006)
##
## Gendermale                                          0.405*                0.397*
##                                                     (0.208)               (0.205)
##
## PhysActiveYes                                                             -1.607***
##                                                                           (0.206)
##
## SmokeNowYes                                                               -1.672***
##                                                                           (0.218)
##
## SleepHrsNight                                                             -0.220***
##                                                                           (0.077)
##
## Constant                    26.489***             26.246***              30.413***
##                              (0.310)               (0.342)               (0.666)
```

```
##
## ---------------------------------------------------------------------------------------------
## Observations                  3,113                    3,113                    3,113
## R2                            0.009                    0.010                    0.044
## Adjusted R2                   0.009                    0.010                    0.043
## Residual Std. Error     5.608 (df = 3111)        5.606 (df = 3110)        5.511 (df = 3107)
## F Statistic          28.237*** (df = 1; 3111) 16.121*** (df = 2; 3110) 28.721*** (df = 5; 3107)
## =============================================================================================
## Note:                                                          *p<0.1; **p<0.05; ***p<0.01
```

```r
library(coefplot)

coefplot(model3, legend = TRUE, title = "Coefficient Comparison",intercept=F)
```