**Introduction to Tidyverse and Exploratory Data Visualization with ggplot2**

**1. Introduction**

The **Tidyverse** is an essential collection of R packages designed for data science. It includes packages such as:

* **ggplot2** for data visualization
* **dplyr** for data manipulation
* **tidyr** for data tidying
* **readr** for reading data
* **tibble** for working with data frames
* **forcats** for handling categorical variables

This tutorial will guide you through using tidyverse for data manipulation and creating exploratory data visualizations using ggplot2.

**2. Installing and Loading the Tidyverse**

If you haven't installed tidyverse, install it using:

install.packages("tidyverse")

Then, load the package:

library(tidyverse)

**3. Importing and Inspecting Data**

We will use the **NHANES** dataset from the NHANES package, which contains health survey data.

First, install and load the package if needed:

install.packages("NHANES")

library(NHANES)

Load the dataset:

# Load dataset

health\_data <- NHANES

# View first few rows

glimpse(health\_data)

Key functions for inspecting data:

head(health\_data) # First few rows

dim(health\_data) # Dimensions of the dataset

summary(health\_data) # Summary statistics

colnames(health\_data) # Column names

**4. Data Manipulation with dplyr**

**4.1 Selecting Columns**

health\_data %>% select(Age, Gender, BMI, BPSysAve, TotChol)

**4.2 Filtering Rows**

health\_data %>% filter(Age > 50, BMI > 18.5) #18.5 is normal weight

**4.3 Creating New Variables**

health\_data <- health\_data %>% mutate(BMI\_Category = ifelse(BMI > 25, "Overweight", "Normal"))

**4.4 Summarizing Data**

health\_data %>% group\_by(Gender) %>% summarise(avg\_BMI = mean(BMI, na.rm = TRUE))

**4.5 Identifying Missing Data, Cardinality, and Outliers**

**Identifying Missing Data**

Missing data can impact analysis. We can check for missing values using:

sum(is.na(health\_data)) # Total missing values

colSums(is.na(health\_data)) # Missing values per column

To remove missing values:

health\_data <- health\_data %>% drop\_na()

**Imputing Missing Data**

Instead of dropping missing values, we can impute them using different strategies:

* Fill with the mean:

health\_data <- health\_data %>% mutate(BMI = ifelse(is.na(BMI), mean(BMI, na.rm = TRUE), BMI))

* Fill with the median:

health\_data <- health\_data %>% mutate(BMI = ifelse(is.na(BMI), median(BMI, na.rm = TRUE), BMI))

* Fill using group-wise mean:

health\_data <- health\_data %>% group\_by(Gender) %>% mutate(BMI = ifelse(is.na(BMI), mean(BMI, na.rm = TRUE), BMI)) %>% ungroup()

**Checking Cardinality (Unique Values in Columns)**

Cardinality refers to the number of unique values in a column:

sapply(health\_data, function(x) length(unique(x)))

This helps identify categorical variables with too many unique values, which might not be useful for modeling.

**Detecting Outliers**

Boxplots help visualize outliers:

ggplot(health\_data, aes(x = Gender, y = BMI)) +

geom\_boxplot() +

labs(title = "BMI Outliers by Gender")

Another method is using the **interquartile range (IQR)**:

Q1 <- quantile(health\_data$BMI, 0.25, na.rm = TRUE)

Q3 <- quantile(health\_data$BMI, 0.75, na.rm = TRUE)

IQR <- Q3 - Q1

outliers <- health\_data %>% filter(BMI < (Q1 - 1.5 \* IQR) | BMI > (Q3 + 1.5 \* IQR))

outliers

This identifies extreme BMI values that might need further investigation.

**5. Exploratory Data Visualization with ggplot2**

**5.1 Histogram**

A histogram is useful for visualizing the distribution of a single variable.

ggplot(health\_data, aes(x = BMI)) +

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

labs(title = "BMI Distribution", x = "BMI", y = "Count")

**5.2 Scatterplot**

ggplot(health\_data, aes(x = Age, y = BPSysAve, color = Gender)) +

geom\_point(size = 3) +

labs(title = "Age vs. Blood Pressure", x = "Age", y = "Blood Pressure")

**5.3 Boxplot**

ggplot(health\_data, aes(x = Gender, y = BMI)) +

geom\_boxplot(fill = "lightblue") +

labs(title = "BMI by Gender", x = "Gender", y = "BMI")

**5.4 Density Plot**

ggplot(health\_data, aes(x = BMI, fill = Gender)) +

geom\_density(alpha = 0.5) +

labs(title = "Density Plot of BMI by Gender", x = "BMI", y = "Density")

**5.5 Bar Chart**

ggplot(health\_data, aes(x = BMI\_Category, fill = Gender)) +

geom\_bar(position = "dodge") +

labs(title = "Count of BMI Categories by Gender", x = "BMI Category", y = "Count")

**6. Customizing ggplot2 Visualizations**

**6.1 Modifying Themes**

ggplot(health\_data, aes(x = Age, y = BPSysAve)) +

geom\_point(color = "blue") +

theme\_minimal() +

labs(title = "Customized Scatterplot", x = "Age", y = "Blood Pressure")

Other available themes include:

theme\_classic(), theme\_light(), theme\_dark(), theme\_bw()

**6.2 Customizing Axis Labels and Titles**

ggplot(health\_data, aes(x = Age, y = BMI)) +

geom\_point() +

labs(title = "Age vs BMI", subtitle = "NHANES Dataset", x = "Age (years)", y = "Body Mass Index (BMI)")

**6.3 Changing Legends and Colors**

ggplot(health\_data, aes(x = Age, y = BMI, color = Gender)) +

geom\_point() +

scale\_color\_manual(values = c("Male" = "blue", "Female" = "red")) +

labs(title = "Age vs BMI by Gender")

**6.4 Customizing Grid Lines**

ggplot(health\_data, aes(x = Age, y = BPSysAve)) +

geom\_point() +

theme(panel.grid.major = element\_line(color = "grey", size = 0.5),

panel.grid.minor = element\_line(color = "lightgrey", size = 0.25))

**6.5 Adding Annotations**

ggplot(health\_data, aes(x = Age, y = BMI)) +

geom\_point() +

annotate("text", x = 80, y = 40, label = "Possible Outlier", color = "red", size = 5)

**7. Saving Plots**

To save a plot, use ggsave():

ggplot(health\_data, aes(x = BMI)) +

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

ggsave("bmi\_histogram.png", width = 8, height = 6)