UNIVERSITI TEKNOLOGI PETRONAS

TEB2043 DATA SCIENCE

LAB 2

Name: Muhammad Ilham Bin Mohammad Faisal

ID: 22008808

Lecturer: Sir Umar Audi Isma'Ila

Activity 1

Basic Imputation Methods

```
# Create data frame with product labels and prices (including NA)
df <- data.frame(Product = c('A', 'B', 'C', 'D', 'E'), Price = c(612, 447,
NA, 374, 831))
# Display the initial data frame
df
# Replace NA values in 'Price' with 0
df$Price[is.na(df$Price)] <- 0
# Display the updated data frame
df

# Replace NA values in 'Price' with the mean of non-NA prices
df$Price[is.na(df$Price)] <- mean(df$Price, na.rm = TRUE)
df

# Replace NA values in 'Price' with the median of non-NA prices
df$Price[is.na(df$Price)] <- median(df$Price, na.rm = TRUE)
df

#installing titinic packages
install.packages('titanic')

#loading the library
library(titanic)

# Load the 'Titanic' dataset from R's built-in datasets
data("Titanic")

# The 'Titanic' dataset is a table, convert it to a data frame for summary
titanic_df <- as.data.frame(Titanic)

# Summarize the 'titanic_df' data frame
summary(titanic_df)
titanic_train$Age

library(ggplot2)
library(dplyr)
library(cowplot)

ggplot(titanic_train, aes(Age)) +
  geom_histogram(color = "#000000", fill = "#0099F8") + ggtitle("Variable
distribution") +
  theme_classic() +
  theme(plot.title = element_text(size = 18))

value_imputed <- data.frame(
  original = titanic_train$Age, imputed_zero = replace(titanic_train$Age,
```

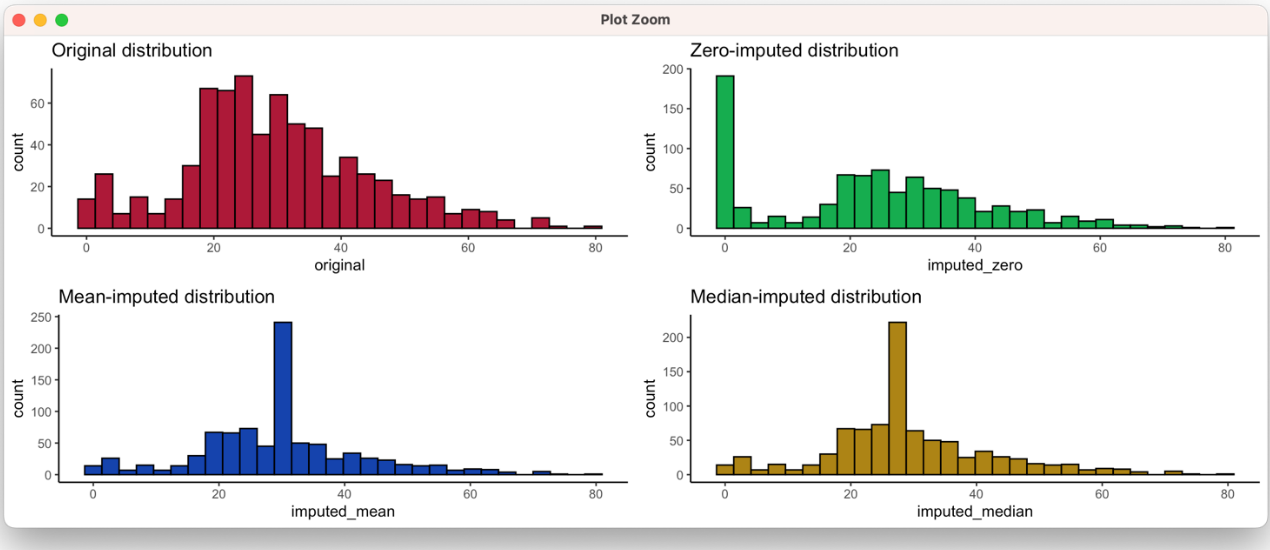
```
is.na(titanic_train$Age), 0),
  imputed_mean = replace(titanic_train$Age,
                         is.na(titanic_train$Age),  mean(titanic_train$Age,
na.rm = TRUE)), imputed_median = replace(titanic_train$Age,

is.na(titanic_train$Age), median(titanic_train$Age, na.rm = TRUE))
)

value_imputed

  h1 <- ggplot(value_imputed, aes(x = original)) + geom_histogram(fill =
"#ad1538", color = "#000000", position = "identity") + ggtitle("Original
distribution") + theme_classic()
  h2 <- ggplot(value_imputed, aes(x = imputed_zero)) + geom_histogram(fill
= "#15ad4f", color = "#000000", position = "identity") + ggtitle("Zero-
imputed distribution") + theme_classic()
  h3 <- ggplot(value_imputed, aes(x = imputed_mean)) + geom_histogram(fill
= "#1543ad", color = "#000000", position = "identity") + ggtitle("Mean-
imputed distribution") + theme_classic()
  h4 <- ggplot(value_imputed, aes(x = imputed_median)) + geom_histogram(fill
= "#ad8415", color = "#000000", position = "identity") + ggtitle("Median-
imputed distribution") + theme_classic()

  #arrange histogram in grid
  plot_grid(h1, h2, h3, h4, nrow = 2, ncol = 2)
```



The image presents a comparative analysis of four different data distributions using histograms,

1. **Original**

   The histogram represents the distribution of the original dataset without any imputation. The data appears to be right skewed with most values concentrated between 10 and 40, with some outliers extending up to 80.

2. **Zero-Imputed Distribution**

   In this histogram, missing values are replaced with zeros. This leads to a significant spike at zero, distorting the original distribution.

3. **Mean-Imputed Distribution**

   Missing values are replaced with the mean of the original dataset. This imputation introduces a pronounced peak around the mean value, causing a distortion in the distribution. The overall shape of the original distribution is somewhat maintained, but the central peak around the mean stands out sharply.

4. **Median-Imputed Distribution**

   Missing values are filled with the median of the original dataset. Like the mean-imputed distribution, there is a prominent spike at the median value. However, this imputation method appears to better preserve the overall distribution pattern compared to mean imputation, with a noticeable central peak where the median lies.

In summary, the histograms illustrate how different imputation methods affect the shape of the data distribution. Zero imputation introduces a large spike at zero, mean imputation creates a central peak around the mean, and median imputation does the same around the median.

Activity 2

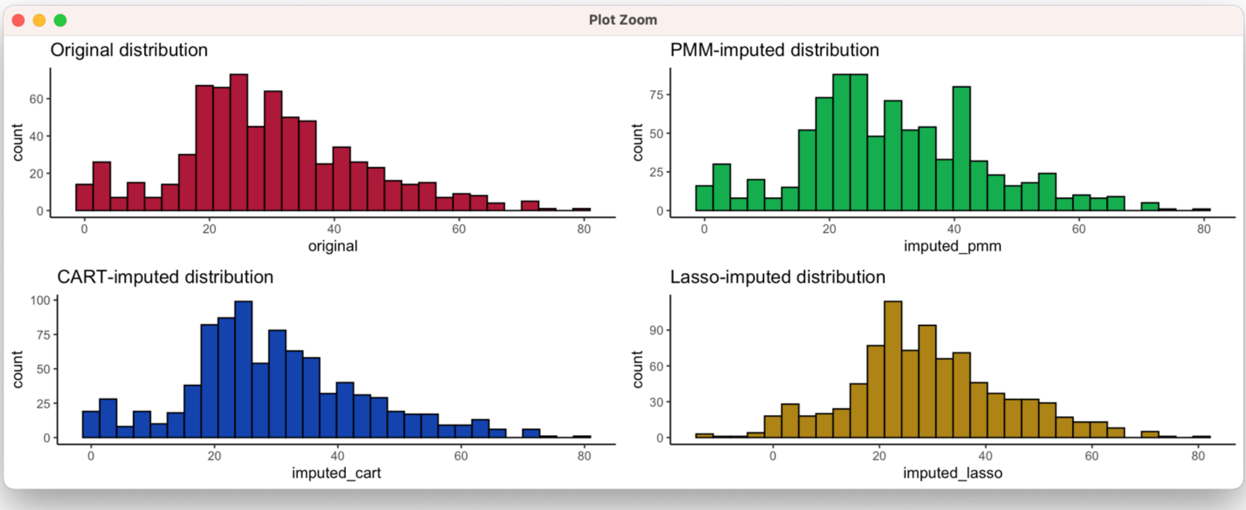
Impute Missing Values in R with MICE

```
library(mice)
titanic_numeric <- titanic_train %>% select(Survived, Pclass, SibSp, Parch,
Age)
md.pattern(titanic_numeric)


mice_imputed <- data.frame(
  original = titanic_train$Age,
  imputed_pmm = complete(mice(titanic_numeric, method = "pmm"))$Age,
  imputed_cart = complete(mice(titanic_numeric, method = "cart"))$Age,
  imputed_lasso = complete(mice(titanic_numeric, method = "lasso.norm"))$Age)
mice_imputed

h1 <- ggplot(mice_imputed, aes(x = original)) + geom_histogram(fill =
"#ad1538", color = "#000000", position = "identity") + ggtitle("Original
distribution") + theme_classic()
h2 <- ggplot(mice_imputed, aes(x = imputed_pmm)) + geom_histogram(fill =
"#15ad4f", color = "#000000", position = "identity") + ggtitle("PMM-imputed
distribution") + theme_classic()
h3 <- ggplot(mice_imputed, aes(x = imputed_cart)) + geom_histogram(fill =
"#1543ad", color = "#000000", position = "identity") + ggtitle("CART-imputed
distribution") + theme_classic()
h4 <- ggplot(mice_imputed, aes(x = imputed_lasso)) + geom_histogram(fill =
"#ad8415", color = "#000000", position = "identity") + ggtitle("Lasso-imputed
distribution") + theme_classic()

#arrange histogram in grid
plot_grid(h1, h2, h3, h4, nrow = 2, ncol = 2)
```



1. **Original Distribution**

   The histogram shows the distribution of the original dataset without any imputation. The data is right skewed.

2. **PMM-Imputed Distribution**

   PMM is used to handle missing values here. The histogram indicates that PMM maintains the general shape and variability of the original distribution. The counts are consistent with the original data, with slight variations but no significant distortions.

3. **CART-Imputed Distribution**

   This histogram shows the distribution after imputation using the Classification and Regression Trees (CART) method. The CART-imputed data retains a similar shape to the original, but there are some subtle changes in the distribution, particularly around the 20 to 30 range, where there is a slight increase in count.

4. **Lasso-Imputed Distribution**

   This distribution shows a noticeable smoothing effect compared to the original. The peak around 20 to 30 is less pronounced, and the data is more evenly spread across the range. This method seems to flatten out some of the variability seen in the original distribution.

In summary, the histograms illustrate how different imputation methods impact the distribution of data with missing values. PMM imputation closely resembles the original distribution, CART imputation introduces minor changes, and Lasso imputation results in a smoother, more evenly distributed dataset.

Activity 3

Imputation with R missForest Package

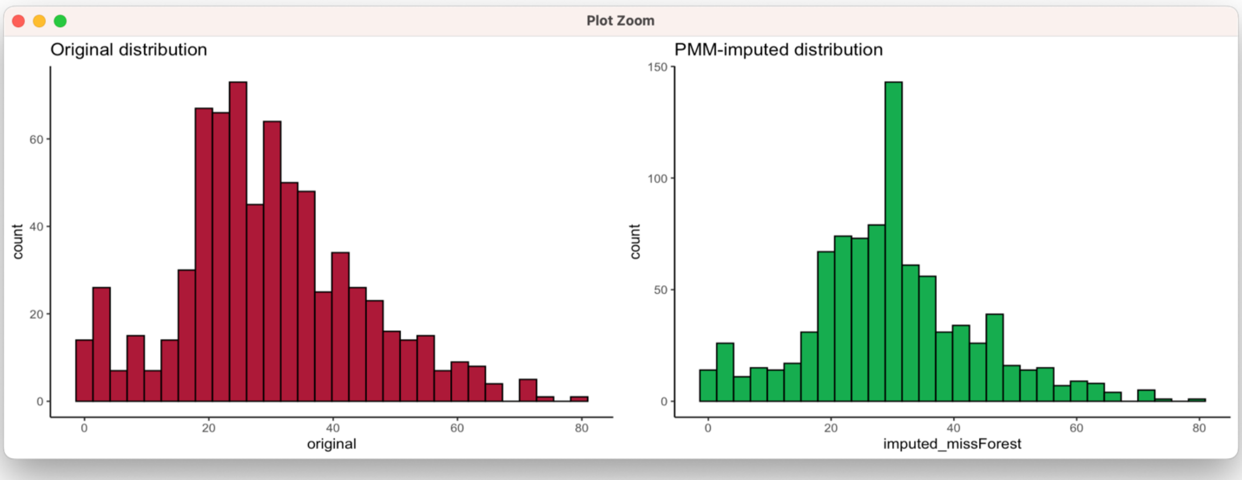
```
install.packages('missForest')

library(missForest)

missForest_imputed <- data.frame (
  original = titanic_numeric$Age,
  imputed_missForest = missForest (titanic_numeric)$ximp$Age
)
missForest_imputed

h1 <- ggplot(missForest_imputed, aes(x = original)) + geom_histogram(fill =
"#ad1538", color = "#000000", position = "identity") + ggtitle("Original
distribution") + theme_classic()
h2 <- ggplot(missForest_imputed, aes(x = imputed_missForest)) +
geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity")
+ ggtitle("PMM-imputed distribution") + theme_classic()

#arrange histogram in grid
plot_grid(h1, h2, nrow = 1, ncol = 2)
```



The image displays two histograms for comparative analysis,

1. **Original Distribution**
   This histogram shows the original dataset's distribution without any imputation. The data is right skewed with a concentration of values between 10 and 40, gradually tapering off towards 80.

2. **PMM-Imputed Distribution using missForest**
   This histogram represents the distribution after applying PMM with the missForest method for handling missing values. The distribution shows a pronounced peak around the value of 30-40, which is higher than any peaks in the original data. Additionally, while the general shape somewhat resembles the original distribution, there is an increase in counts around the peak, suggesting that the missForest imputation has introduced a significant concentration of values in this range.

In summary, the comparison between the original and PMM-imputed distributions highlights how the missForest method affects the dataset. While the overall shape is somewhat preserved, the imputation method has created a notable peak.

Activity 4

Normalize data with scaling methods

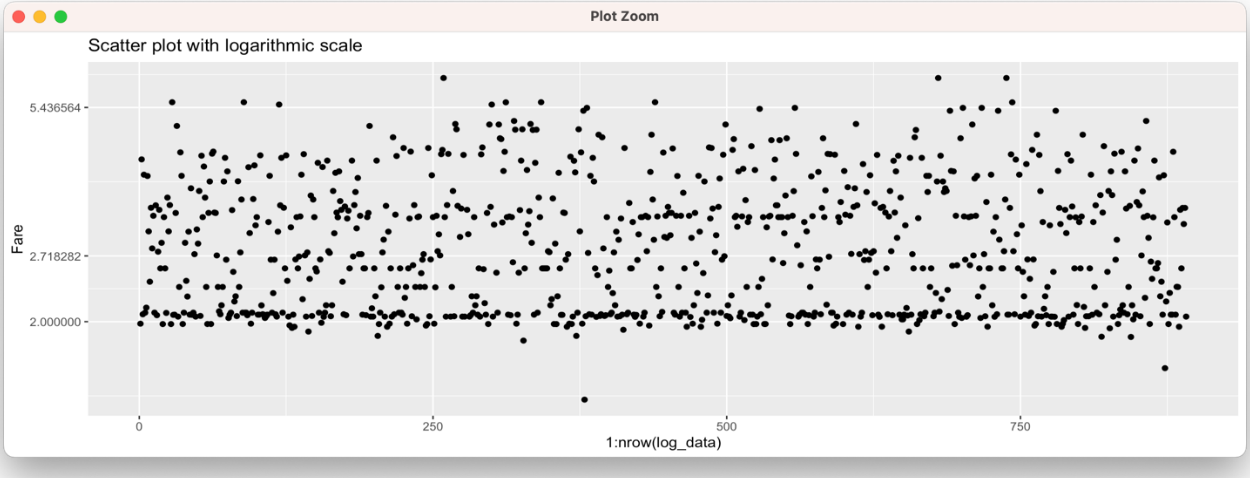
```
log_scale = log(titanic_train$Fare)

log_data <- data.frame(Fare = log_scale)

ggplot(log_data, aes (x = 1:nrow(log_data), y = Fare)) +
  geom_point() +
  scale_y_continuous(trans = 'log') +
  ggtitle("Scatter plot with logarithmic scale")


install.packages('caret')
library(caret)
process <- preProcess(as.data.frame(titanic$Fare), method=c("range"))
norm_scale <- predict(process, as.data.frame(titanic$Fare))

scale_data <- scale(titanic_train$Fare)
```



- **Scatter Plot with Logarithmic Scale** : Visualizes the log-transformed Fare values to handle skewness and identify patterns or outliers more effectively.
- **Normalization** : Normalizes the Fare values to a range of [0, 1], which can be useful for certain machine learning algorithms.
- **Standard Scaling** : Standardizes the Fare values to have a mean of 0 and a standard deviation of 1, which is another common pre-processing step in machine learning.

Activity 5

Feature Encoding

```
gender_encode <- ifelse(titanic_train$Sex == "male",1,0)
table(gender_encode)

new_dat =
data.frame(titanic_train$Fare,titanic_train$Sex,titanic_train$Embarked)
summary(new_dat)


library(caret)
dmy <- dummyVars(" ~ .", data = new_dat, fullRank = T)
dat_transformed <- data.frame(predict(dmy, newdata = new_dat))

glimpse(dat_transformed)

summary(new_dat$titanic_train.Fare)

bins <- c(-Inf, 7.91, 31.00, Inf)

bin_names <- c("Low", "Mid50", "High")

new_dat$new_Fare <- cut(new_dat$titanic_train.Fare, breaks = bins, labels =
bin_names)

summary(new_dat$titanic_train.Fare)

summary(new_dat$new_Fare)
```

Console | Terminal | Compile PDF | Render | Background Jobs

R 4.4.0 · ~/

```
> gender_encode <- ifelse(titanic_train$Sex == "male",1,0)
> table(gender_encode)
gender_encode
  0   1 
314 577 
> 
> new_dat = data.frame(titanic_train$Fare,titanic_train$Sex,titanic_train$Embarked)
> summary(new_dat)
 titanic_train.Fare titanic_train.Sex  titanic_train.Embarked
 Min.   :  0.00     Length:891         Length:891            
 1st Qu.:  7.91     Class :character   Class :character      
 Median : 14.45     Mode  :character   Mode  :character      
 Mean   : 32.20                                              
 3rd Qu.: 31.00                                              
 Max.   :512.33                                              
> 
> 
> library(caret)
> dmy <- dummyVars(" ~ .", data = new_dat, fullRank = T) dat_transformed <- data.frame
(predict(dmy, newdata = new_dat))
```

Console | Terminal | Compile PDF | Render | Background Jobs

R 4.4.0 · ~/

```
> glimpse(dat_transformed)
Rows: 891
Columns: 5
$ titanic_train.Fare      <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.458…
$ titanic_train.Sexmale   <dbl> 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,…
$ titanic_train.EmbarkedC <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,…
$ titanic_train.EmbarkedQ <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ titanic_train.EmbarkedS <dbl> 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,…
> 
> summary(new_dat$titanic_train.Fare)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    7.91   14.45   32.20   31.00  512.33 
> 
> bins <- c(-Inf, 7.91, 31.00, Inf)
> 
> bin_names <- c("Low", "Mid50", "High")
> 
> new_dat$new_Fare <- cut(new_dat$titanic_train.Fare, breaks = bins, labels = bin_name
s)
> 
```

```
> gender_encode <- ifelse(titanic_train$Sex == "male",1,0)
> table(gender_encode)
gender_encode
  0   1 
314 577 
> 
> new_dat = data.frame(titanic_train$Fare,titanic_train$Sex,titanic_train$Embarked)
> summary(new_dat)
 titanic_train.Fare titanic_train.Sex  titanic_train.Embarked
 Min.   :  0.00     Length:891         Length:891            
 1st Qu.:  7.91     Class :character   Class :character      
 Median : 14.45     Mode  :character   Mode  :character      
 Mean   : 32.20                                              
 3rd Qu.: 31.00                                              
 Max.   :512.33                                              
> 
> 
> library(caret)
> dmy <- dummyVars(" ~ .", data = new_dat, fullRank = T)
> dat_transformed <- data.frame(predict(dmy, newdata = new_dat))
> 
```

```
> summary(new_dat$titanic_train.Fare)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    7.91   14.45   32.20   31.00  512.33 
> 
> summary(new_dat$new_Fare)
  Low Mid50  High 
  223   446   222 
> 
```

- **Gender Encoding** : The ‘Sex’ variable is converted to binary with males as 1 and females as 0. The frequency table shows the distribution of genders.
- **Creating a New Data Frame** : A new data frame ‘new_dat’ with ‘Fare’, ‘Sex’, and Embarked is created and summarized to provide basic statistics and counts.
- **Dummy Variable Transformation** : Categorical variables in ‘new_dat’ are converted into dummy/indicator variables to facilitate numerical analysis and modelling. ‘glimpse’ provides a quick view of the transformed data.
- **Summarizing ‘Fare’** : Provides basic statistics for the Fare variable.
- **Binning ‘Fare’** : The ‘Fare’ variable is binned into three categories: "Low", "Mid50", and "High". The summary of ‘new_Fare’ shows the distribution of Fare values across these bins.

This approach ensures that categorical variables are properly encoded for analysis.