R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Hide

```
install.packages("readr")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
'C:/Users/leewq/AppData/Local/R/win-library/4.2'의 위치에 패키지(들)을 설치합니다.
(왜냐하면 'lib'가 지정되지 않았기 때문입니다)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/readr_2.1.4.zip'
Content type 'application/zip' length 1192968 bytes (1.1 MB)
downloaded 1.1 MB
```

```
패키지 'readr'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\Rtmp4K1qIn\downloaded_packages
```

Hide

```
install.packages("FactoMineR")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("FNN")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
'C:/Users/leewq/AppData/Local/R/win-library/4.2'의 위치에 패키지(들)을 설치합니다.
(왜냐하면 'lib'가 지정되지 않았기 때문입니다)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/FNN_1.1.3.2.zip'
Content type 'application/zip' length 450109 bytes (439 KB)
downloaded 439 KB
```

```
패키지 'FNN'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\Rtmp4K1qIn\downloaded_packages
```

Hide

```
install.packages("FactoMineR")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
‘C:/Users/leewq/AppData/Local/R/win-library/4.2’의 위치에 패키지(들)을 설치합니다.
(왜냐하면 ‘lib’가 지정되지 않았기 때문입니다)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/FactoMineR_2.7.zip'
Content type 'application/zip' length 3804028 bytes (3.6 MB)
downloaded 3.6 MB
```

```
패키지 ‘FactoMineR’를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다
Warning in install.packages :
  패키지 ‘FactoMineR’의 이전설치를 삭제할 수 없습니다
Warning in install.packages :
  problem copying C:\Users\leewq\AppData\Local\R\win-library\4.2\00LOCK\FactoMineR\libs\x64\F
actoMineR.dll to C:\Users\leewq\AppData\Local\R\win-library\4.2\FactoMineR\libs\x64\FactoMine
R.dll: Permission denied
Warning in install.packages :
  ‘FactoMineR’를 복구하였습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다
    C:\Users\leewq\AppData\Local\Temp\Rtmp4K1qIn\downloaded_packages
```

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
library(readr)
```

```
경고: 패키지 ‘readr’는 R 버전 4.2.3에서 작성되었습니다
```

Hide

```
# Load the dataset from the CSV file
data <- read_csv("perth.csv")
```

```
Rows: 33656 Columns: 19── Column specification ────────────────────────────────────────────
────────────────────────────────────────────────

Delimiter: ","
chr  (7): ADDRESS, SUBURB, GARAGE, BUILD_YEAR, NEAREST_STN, DATE_SOLD, NEAREST_SCH
dbl (12): PRICE, BEDROOMS, BATHROOMS, LAND_AREA, FLOOR_AREA, CBD_DIST, NEAREST_STN_DIST, POST
CODE, LATITUDE...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

Hide

```
# Convert all non-numeric columns to numeric
# Identify non-numeric columns
non_numeric_columns <- sapply(data, function(x) !is.numeric(x))

# Remove non-numeric columns
data_numeric <- data[, !non_numeric_columns]

# Check if all columns are now numeric
str(data_numeric)
```

```
tibble [33,656 × 12] (S3: tbl_df/tbl/data.frame)
 $ PRICE           : num [1:33656] 565000 365000 287000 255000 325000 409000 400000 370000 56
5000 685000 ...
 $ BEDROOMS        : num [1:33656] 4 3 3 2 4 4 3 4 4 3 ...
 $ BATHROOMS       : num [1:33656] 2 2 1 1 1 2 2 2 2 2 ...
 $ LAND_AREA       : num [1:33656] 600 351 719 651 466 759 386 468 875 552 ...
 $ FLOOR_AREA      : num [1:33656] 160 139 86 59 131 118 132 158 168 126 ...
 $ CBD_DIST        : num [1:33656] 18300 26900 22600 17900 11200 27300 28200 41700 12100 5900
...
 $ NEAREST_STN_DIST: num [1:33656] 1800 4900 1900 3600 2000 1000 3700 1100 2500 508 ...
 $ POSTCODE        : num [1:33656] 6164 6167 6111 6056 6054 ...
 $ LATITUDE        : num [1:33656] -32.1 -32.2 -32.1 -31.9 -31.9 ...
 $ LONGITUDE       : num [1:33656] 116 116 116 116 116 ...
 $ NEAREST_SCH_DIST: num [1:33656] 0.828 5.524 1.649 1.571 1.515 ...
 $ NEAREST_SCH_RANK: num [1:33656] NA 129 113 NA NA NA NA NA NA 29 ...
```

Hide

```
library(FactoMineR)
```

```
경고: 패키지 'FactoMineR'는 R 버전 4.2.3에서 작성되었습니다
```

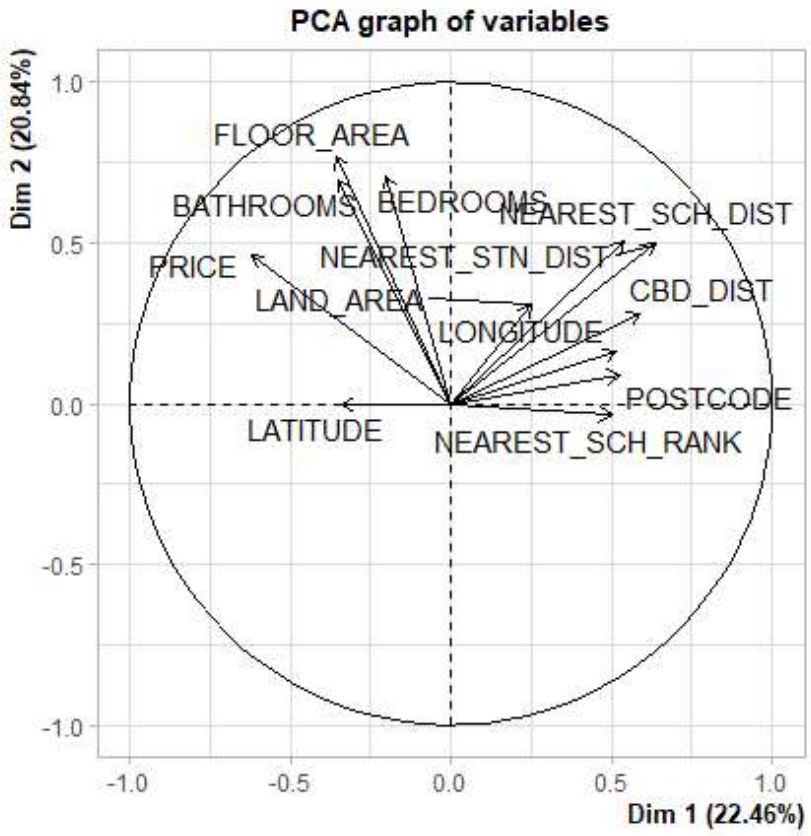
Hide

```
# Perform PCA
res.pca <- PCA(data_numeric, scale.unit = TRUE, graph = FALSE)
```

```
경고: Missing values are imputed by the mean of the variable: you should use the imputePCA fu
nction of the missMDA package
```

Hide

```
# Determine the optimal number of components by plotting the explained variance ratio
plot(res.pca, choix = "var")
```



Hide

```
NA
NA
```

Hide

```
# Set the target variable as 'price'
target <- "PRICE"

# Remove rows with NA values
data_numeric_clean <- na.omit(data_numeric)

# Split the cleaned dataset into features and target variable(s)
X_clean <- data_numeric_clean[, !names(data_numeric_clean) %in% target]
y_clean <- data_numeric_clean[, target]

# Convert y_clean to a vector
y_clean_vector <- as.vector(unlist(y_clean))

# Check the dimensions of X_clean and y_clean
cat("Dimensions of X_clean:", dim(X_clean), "\n")
```

```
Dimensions of X_clean: 22704 11
```

Hide

```
cat("Length of y_clean_vector:", length(y_clean_vector), "\n")
```

```
Length of y_clean_vector: 22704
```

Hide

```
# Perform LDA
library(MASS)
res.lda <- lda(X_clean, y_clean_vector)
```

Hide

```
# Load the necessary libraries

# Load the necessary libraries
library(caret)
```

```
필요한 패키지를 로딩중입니다: ggplot2
필요한 패키지를 로딩중입니다: lattice
```

Hide

```
library(FNN)
```

```
경고: 패키지 'FNN'는 R 버전 4.2.3에서 작성되었습니다
```

Hide

```
# Split the data into train and test sets
set.seed(42)
trainIndex <- createDataPartition(y_vector, p = 0.8, list = FALSE)
train_pca <- res.pca$ind$coord[trainIndex, 1:2] # Replace 2 with the number of components cho
sen
test_pca <- res.pca$ind$coord[-trainIndex, 1:2] # Replace 2 with the number of components cho
sen
y_train <- y_vector[trainIndex]
y_test <- y_vector[-trainIndex]

# Train and evaluate the kNN regression on PCA-reduced data
knn_pca <- knn.reg(train = train_pca, test = test_pca, y = y_train)

# Calculate the mean squared error (MSE) and root mean squared error (RMSE)
mse_pca <- mean((knn_pca$pred - y_test)^2)
```

```
경고: longer object length is not a multiple of shorter object length
```

Hide

```
rmse_pca <- sqrt(mse_pca)

# Print the results
cat("MSE for kNN regression on PCA-reduced data:", mse_pca, "\n")
```

```
MSE for kNN regression on PCA-reduced data: 201988363475
```

Hide

```
cat("RMSE for kNN regression on PCA-reduced data:", rmse_pca, "\n")
```

```
RMSE for kNN regression on PCA-reduced data: 449431.2
```

Hide

```
library(MASS)

# Perform LDA
res_lda <- lda(X_clean, y_clean_vector)

# Project data using LDA
X_lda <- predict(res_lda, X_clean)$x

# Split the LDA-reduced data into train and test sets
trainIndex <- createDataPartition(y_clean_vector, p = 0.8, list = FALSE)
train_lda <- X_lda[trainIndex,]
test_lda <- X_lda[-trainIndex,]
y_train <- y_clean_vector[trainIndex]
y_test <- y_clean_vector[-trainIndex]

# Train and evaluate the kNN regression on LDA-reduced data
knn_lda <- knn.reg(train_lda, test_lda, y_train)
mse_lda <- mean((knn_lda$pred - y_test)^2)
rmse_lda <- sqrt(mse_lda)

cat("MSE for kNN regression on LDA-reduced data:", mse_lda, "\n")
```

```
MSE for kNN regression on LDA-reduced data: 41061045892
```

Hide

```
cat("RMSE for kNN regression on LDA-reduced data:", rmse_lda, "\n")
```

```
RMSE for kNN regression on LDA-reduced data: 202635.3
```

Hide

```
library(caret)
library(FNN)
library(MASS)

# Replace X and y_vector with your dataset and target variable
X <- X_clean
y_vector <- y_clean_vector

# Split the data into train and test sets
set.seed(42)
trainIndex <- createDataPartition(y_vector, p = 0.8, list = FALSE)
train <- X[trainIndex,]
test <- X[-trainIndex,]
y_train <- y_vector[trainIndex]
y_test <- y_vector[-trainIndex]

# Impute missing values in the train and test datasets
# Convert data frames to matrices
train_imputed <- as.matrix(train_imputed)
test_imputed <- as.matrix(test_imputed)

# Train and evaluate the kNN regression on the imputed original data
knn_original <- knn.reg(train = train_imputed, test = test_imputed, y = y_train)

# Make predictions on test set
pred_original <- knn.reg(train = train_imputed, test = test_imputed, y = y_train)$pred


# Perform PCA
res.pca <- prcomp(X, scale. = TRUE)

# Project data using PCA
train_pca <- res.pca$x[trainIndex, 1:2] # Replace 2 with the number of components chosen
test_pca <- res.pca$x[-trainIndex, 1:2] # Replace 2 with the number of components chosen

# Train and evaluate the kNN regression on PCA-reduced data
knn_pca <- knn.reg(train_pca, test_pca, y_train)
mse_pca <- mean((knn_pca$pred - y_test)^2)
rmse_pca <- sqrt(mse_pca)

cat("MSE for kNN regression on PCA-reduced data:", mse_pca, "\n")
```

```
MSE for kNN regression on PCA-reduced data: 95613532827
```

Hide

```
cat("RMSE for kNN regression on PCA-reduced data:", rmse_pca, "\n")
```

```
RMSE for kNN regression on PCA-reduced data: 309214.4
```

Hide

```
# Perform LDA
res_lda <- lda(X, y_vector)

# Project data using LDA
X_lda <- predict(res_lda, X)$x
train_lda <- X_lda[trainIndex,]
test_lda <- X_lda[-trainIndex,]

# Train and evaluate the kNN regression on LDA-reduced data
knn_lda <- knn.reg(train_lda, test_lda, y_train)
mse_lda <- mean((knn_lda$pred - y_test)^2)
rmse_lda <- sqrt(mse_lda)

cat("MSE for kNN regression on LDA-reduced data:", mse_lda, "\n")
```

```
MSE for kNN regression on LDA-reduced data: 42561456705
```

Hide

```
cat("RMSE for kNN regression on LDA-reduced data:", rmse_lda, "\n")
```

```
RMSE for kNN regression on LDA-reduced data: 206304.3
```

Hide

```
# Train and evaluate the kNN regression on the imputed original data
knn_original <- knn.reg(train = train_imputed, test = test_imputed, y = y_train)

# Calculate the mean squared error (MSE) and root mean squared error (RMSE)
mse_original <- mean((knn_original$pred - y_test)^2)
```

```
경고: longer object length is not a multiple of shorter object length
```

Hide

```
rmse_original <- sqrt(mse_original)

# Calculate the difference in performance metrics
rmse_diff_pca <- rmse_original - rmse_pca
rmse_diff_lda <- rmse_original - rmse_lda

cat("RMSE for kNN regression on original imputed data:", rmse_original, "\n")
```

```
RMSE for kNN regression on original imputed data: 503804.7
```

Hide

```
cat("RMSE difference (PCA):", rmse_diff_pca, "\n")
```

```
RMSE difference (PCA): 194590.3
```

Hide

```
cat("RMSE difference (LDA):", rmse_diff_lda, "\n")
```

```
RMSE difference (LDA): 297500.4
```

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.