R Notebook

```r
# Lab 17_1
data(cars)  # Load built-in dataset
x <- cars$speed  # Independent variable (speed)
y <- cars$dist   # Dependent variable (stopping distance)
```

```r
#Lab 17_A
x_mean <- mean(x)
y_mean <- mean(y)

Sxx <- sum((x - x_mean)^2)
Sxy <- sum((x - x_mean) * (y - y_mean))
Syy <- sum((y - y_mean)^2)

Sxx
```

```
## [1] 1370
```

```r
Sxy
```

```
## [1] 5387.4
```

```r
Syy
```

```
## [1] 32538.98
```

```r
#Lab 17_B
Beta1 <- Sxy / Sxx  # Slope
Beta0 <- y_mean - Beta1 * x_mean  # Intercept

Beta0
```

```
## [1] -17.57909
```

```r
Beta1
```

```
## [1] 3.932409
```

```r
#Lab 17_C
y_pred_8 <- Beta0 + Beta1 * 8
y_pred_8
```

```
## [1] 13.88018
```

```r
#Lab 17_D
actual_y_8 <- cars[5, "dist"]
residual_8 <- actual_y_8 - y_pred_8
residual_8
```

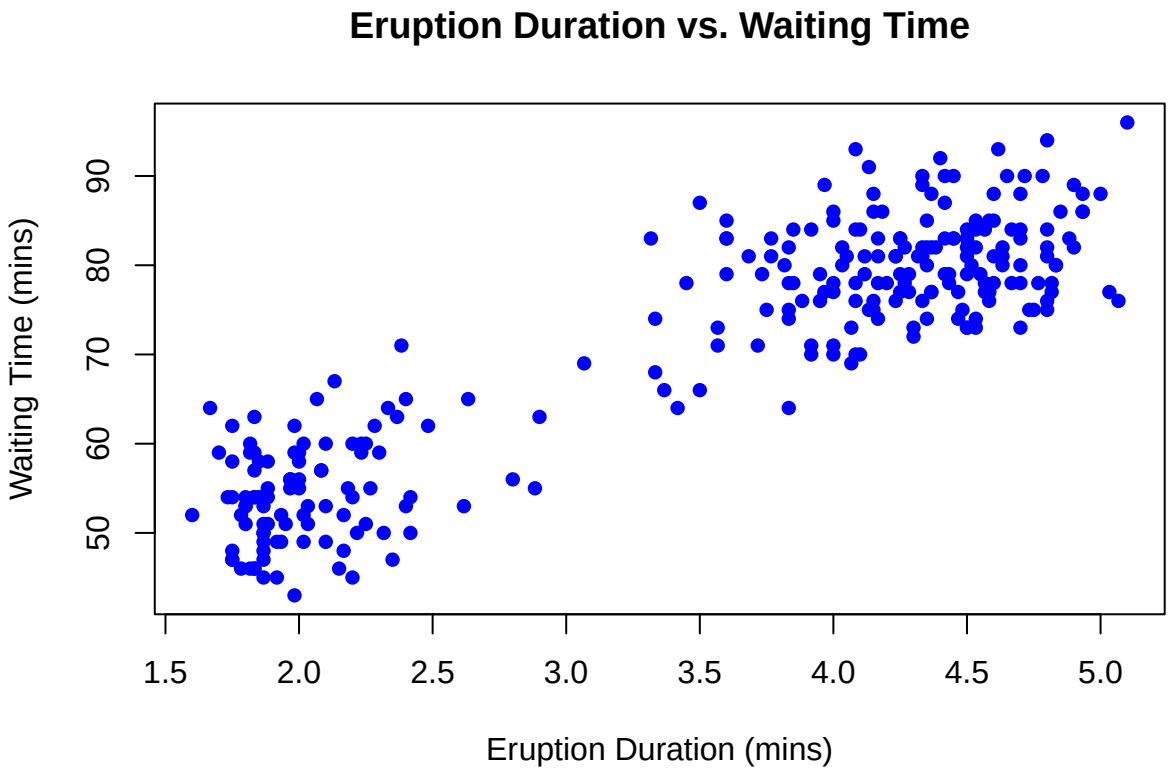
```
## [1] 2.119825
```

```r
model <- lm(dist ~ speed, data = cars)
summary(model)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```r
# Lab 17_2
data(faithful)  # Load built-in dataset
```

```r
# Lab 17_2A
plot(faithful$eruptions, faithful$waiting,
     xlab = "Eruption Duration (mins)",
     ylab = "Waiting Time (mins)",
     main = "Eruption Duration vs. Waiting Time",
     col = "blue", pch = 16)
```




```r
# Lab 17_2B
correlation <- cor(faithful$eruptions, faithful$waiting)
```

```r
correlation
```

```
## [1] 0.9008112
```

```r
# Lab 17_2C
# Fit a linear regression model
lab_model <- lm(waiting ~ eruptions, data = faithful)
```

```r
# Lab 17_2D
summary(lab_model)
```

```
##
## Call:
## lm(formula = waiting ~ eruptions, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0796  -4.4831   0.2122   3.9246  15.9719
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.4744     1.1549   28.98   <2e-16 ***
## eruptions    10.7296     0.3148   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.914 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

```r
# Lab 17_2E
new_data <- data.frame(eruptions = 3.667)
predicted_waiting <- predict(lab_model, new_data)
predicted_waiting
```

```
##        1
## 72.81999
```

```r
# Lab 17_3A
euclidean_distance <- function(v1, v2) {
  # Check if vectors are of the same length
  if (length(v1) != length(v2)) {
    stop("Error: Vectors must be of the same length.")
  }

  # Compute Euclidean distance
  distance <- sqrt(sum((v1 - v2)^2))
  return(distance)
}
```

```r
# Lab 17_3B
# Define the vectors
v1 <- c(10.09, 2.33, 9.71, 101.46)
v2 <- c(12.21, 9.41, 7.65, 163.12)

# Compute the Euclidean distance
distance_result <- euclidean_distance(v1, v2)
```

```
distance_result
```

```
## [1] 62.1355
```

```
# Lab 17_4
# Load the dataset
vehicle_data <- read.csv("vehicle_info.csv")

# Browse the data
head(vehicle_data)
```

```
##   origin price mileage repair headspace trunkspace weight length turningcircle
## 1    usa  4099     8.8      3      6.25        308 1318.5  465.0         12.20
## 2    usa  4749     6.8      3      7.50        308 1507.5  432.5         12.20
## 3    usa  3799     8.8      3      7.50        336 1188.0  420.0         10.68
## 4    usa  4816     8.0      3     11.25        448 1462.5  490.0         12.20
## 5    usa  7827     6.0      4     10.00        560 1836.0  555.0         13.12
## 6    usa  5788     7.2      3     10.00        588 1651.5  545.0         13.12
##   gear_ratio
## 1       3.58
## 2       2.53
## 3       3.08
## 4       2.93
## 5       2.41
## 6       2.73
```

```
# Lab 17_4A
# View structure and dimensionality
str(vehicle_data)
```

```
## 'data.frame':    74 obs. of  10 variables:
##  $ origin       : chr  "usa" "usa" "usa" "usa" ...
##  $ price        : num  4099 4749 3799 4816 7827 ...
##  $ mileage      : num  8.8 6.8 8.8 8 6 7.2 10.4 8 6.4 7.6 ...
##  $ repair       : num  3 3 3 3 4 3 3 3 3 3 ...
##  $ headspace    : num  6.25 7.5 7.5 11.25 10 ...
##  $ trunkspace   : num  308 308 336 448 560 588 280 448 476 364 ...
##  $ weight       : num  1318 1508 1188 1462 1836 ...
##  $ length       : num  465 432 420 490 555 ...
##  $ turningcircle: num  12.2 12.2 10.7 12.2 13.1 ...
##  $ gear_ratio   : num  3.58 2.53 3.08 2.93 2.41 2.73 2.87 2.93 2.93 3.08 ...
```

```
dim(vehicle_data)
```

```
## [1] 74 10
```

```
#lab 17_4B
# Identify the label variable for classification
names(vehicle_data)
```

```
##  [1] "origin"        "price"         "mileage"       "repair"
##  [5] "headspace"     "trunkspace"    "weight"        "length"
##  [9] "turningcircle" "gear_ratio"
```

```
cat("\nUnique vales in the 'origin'column:\n")
```

```
##
## Unique vales in the 'origin'column:
```

```r
table(vehicle_data$origin)
```

```
##
## other   usa
##    22    52
```

```r
#lab 17_4C
# Function to normalize values between 0 and 1
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Create a copy of the dataframe for normalization
vehicle_data_norm <- vehicle_data

# Normalize all numeric columns (exclude the "origin" column)
vehicle_data_norm[, 2:ncol(vehicle_data)] <- lapply(vehicle_data[, 2:ncol(vehicle_data)], normalize)

# Check the normalized data
head(vehicle_data_norm)
```

```
##   origin      price   mileage repair headspace trunkspace    weight    length
## 1    usa 0.06405073 0.3448276   0.50 0.2857143  0.3333333 0.3798701 0.4835165
## 2    usa 0.11557669 0.1724138   0.50 0.4285714  0.3333333 0.5162338 0.3406593
## 3    usa 0.04026952 0.3448276   0.50 0.4285714  0.3888889 0.2857143 0.2857143
## 4    usa 0.12088783 0.2758621   0.50 0.8571429  0.6111111 0.4837662 0.5934066
## 5    usa 0.35957194 0.1034483   0.75 0.7142857  0.8333333 0.7532468 0.8791209
## 6    usa 0.19793896 0.2068966   0.50 0.7142857  0.8888889 0.6201299 0.8351648
##   turningcircle gear_ratio
## 1     0.4500818  0.8176471
## 2     0.4500818  0.2000000
## 3     0.2013093  0.5235294
## 4     0.4500818  0.4352941
## 5     0.6006547  0.1294118
## 6     0.6006547  0.3176471
```

```r
#Lab 17_4D
# Load required libraries
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
library(ggplot2)

# Set seed for reproducibility
set.seed(123)

# Stratified sampling to maintain class distribution
train_index <- createDataPartition(vehicle_data_norm$origin,
                                   p = 0.8,
                                   list = FALSE,
                                   times = 1)

# Create training and test sets
```

```r
train_data <- vehicle_data_norm[train_index, ]
test_data <- vehicle_data_norm[-train_index, ]

# Check distribution of classes in both sets
table(train_data$origin)
```

```
##
## other   usa
##    18    42
```

```r
table(test_data$origin)
```

```
##
## other   usa
##     4    10
```

```r
# Lab 17_4E
# Load required package
library(class)

# Prepare training and test sets
train_labels <- train_data$origin
test_labels <- test_data$origin

# Remove the label column from the feature sets
train_features <- train_data[, -1]  # Excluding the "origin" column
test_features <- test_data[, -1]    # Excluding the "origin" column

# Build KNN model with k=9
knn_pred <- knn(train = train_features,
                test = test_features,
                cl = train_labels,
                k = 9)
```

```r
# Lab 17_4F
# Load required library
library(caret)

# Create confusion matrix
conf_matrix <- confusionMatrix(data = as.factor(knn_pred),
                               reference = as.factor(test_labels))

# Display confusion matrix
conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction other usa
##      other     3   2
##      usa       1   8
##
##                Accuracy : 0.7857
##                  95% CI : (0.492, 0.9534)
##     No Information Rate : 0.7143
##     P-Value [Acc > NIR] : 0.4001
```

```
##
##                  Kappa : 0.5116
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.7500
##             Specificity : 0.8000
##          Pos Pred Value : 0.6000
##          Neg Pred Value : 0.8889
##              Prevalence : 0.2857
##          Detection Rate : 0.2143
##    Detection Prevalence : 0.3571
##       Balanced Accuracy : 0.7750
##
##        'Positive' Class : other
##
```