## 3.2 Xử lí dữ liệu.

### 3.2.1 Kiểm tra tổng thể dữ liệu.

```
# Hàm để lấy thông tin dữ liệu bao gồm loại dữ liệu, giá trị duy nhất, và giá trị null
datainfo <- function(data) {
  # Tạo một khung dữ liệu để lưu trữ thông tin
  temp_ps <- data.frame(matrix(ncol = 6, nrow = ncol(data)))
  colnames(temp_ps) <- c("Column_Name", "DataType", "Non_null_Values", "Unique_Values",
"NaN_Values_Percentage", "Duplicates")

  # Điền thông tin vào khung dữ liệu
  temp_ps$Column_Name <- colnames(data)  # Lấy tên các cột
  temp_ps$DataType <- sapply(data, class)
  # Lấy loại dữ liệu sử dụng sapply
  temp_ps$Non_null_Values <- sapply(data, function(x) sum(!is.na(x)))
  # Đếm số giá trị không null sử dụng sapply và is.na
  temp_ps$Unique_Values <- sapply(data, function(x) length(unique(na.omit(x))))
  # Đếm số giá trị duy nhất loại bỏ các giá trị NA
  temp_ps$NaN_Values <- sapply(data, function(x) sum(is.na(x)))
  # Đếm số giá trị null sử dụng sapply và is.na
  temp_ps$NaN_Values_Percentage <- (temp_ps$NaN_Values / nrow(data)) * 100
  # Tính tỷ lệ phần trăm của các giá trị null

  # Đếm số lượng bản ghi bị trùng lặp trong dữ liệu
  temp_ps$Duplicates <- sum(duplicated(data))

  # Trả về khung dữ liệu chứa thông tin
  return(temp_ps)
}
data_info <- datainfo(data)
print(data_info)
```

```
##                Column_Name  DataType Non_null_Values Unique_Values
## 1                     date character            8760           365
## 2        rented_bike_count   numeric            8760          2166
## 3                     hour   numeric            8760            24
## 4           temperature_c   numeric            8760           546
## 5         humidity_percent   numeric            8760            90
## 6           wind_speed_m_s   numeric            8760            65
## 7           visibility_10m   numeric            8760          1789
## 8  dew_point_temperature_c   numeric            8760           556
## 9    solar_radiation_mj_m2   numeric            8760           345
## 10             rainfall_mm   numeric            8760            61
## 11             snowfall_cm   numeric            8760            51
## 12                 seasons character            8760             4
## 13                 holiday character            8760             2
## 14         functioning_day character            8760             2
##    NaN_Values_Percentage Duplicates NaN_Values
## 1                      0          0          0
## 2                      0          0          0
```

```
## 3                       0          0          0
## 4                       0          0          0
## 5                       0          0          0
## 6                       0          0          0
## 7                       0          0          0
## 8                       0          0          0
## 9                       0          0          0
## 10                      0          0          0
## 11                      0          0          0
## 12                      0          0          0
## 13                      0          0          0
## 14                      0          0          0
```

=> Bộ dữ liệu không chứa NaN

### 3.2.2 Kiểm tra hàm functioning_day với rented_bike_coun

```
result <- data %>%
  group_by(functioning_day) %>%
  summarise(rented_bike_count = sum(rented_bike_count, na.rm = TRUE)) %>%
  arrange(desc(rented_bike_count)) %>%
  ungroup()
print(result)
```

```
## # A tibble: 2 × 2
##   functioning_day rented_bike_count
##   <chr>                       <dbl>
## 1 Yes                       6172314
## 2 No                              0
```

```
summary(factor(data$functioning_day))
```

```
##   No  Yes
##  295 8465
```

=> **Nhận xét**: Theo thống kê tóm tắt của Working_day, có 295 ngày không hoạt động trong tập dữ liệu. Những ngày này sẽ không có số lượng xe đạp vì không thể thuê. Do đó, các điểm dữ liệu của ngày không hoạt động sẽ bị xóa khỏi tập dữ liệu vì điều này không liên quan đến mục đích của bài toán. Tập dữ liệu hiện còn lại 8465 điểm dữ liệu.

```
data <- data %>%
  filter(functioning_day != "No") %>% dplyr::select(-functioning_day)
```

### 3.2.3 Chuyển đổi kiểu dữ liệu cho các biến holiday, seasons.

```
  data <- data |> mutate(
  holiday = as.factor(holiday),
  seasons = as.factor(seasons)
)
```

### 3.2.4 Tách thời gian thành ngày, tháng, năm và thêm cột ngày trong tuần.

```
data <- data %>%
  mutate(
    Date = dmy(date),
    # Chuyển đổi cột 'date' sang định dạng ngày tháng
    year = year(Date),
```

```
    month = month(Date),
    day = day(Date)
  )
# Loại bỏ cột 'Date'
data <- subset(data, select = -c(Date))
# Tạo cột day bằng cách kết hợp year, month và day
data$full_date <- as.Date(with(data, paste(year, month, day, sep = "-")), format = "%Y-
%m-%d")

# Thêm cột day_of_week dựa trên cột full_date
data$day_of_week <- weekdays(data$full_date)

# Loại bỏ cột 'full_date'
data <- subset(data, select = -c(full_date))

# Hiển thị kết quả
print(head(data))
```

```
## # A tibble: 6 × 17
##   date     rented_bike_count  hour temperature_c humidity_percent wind_speed_m_s
##   <chr>                <dbl> <dbl>         <dbl>            <dbl>          <dbl>
## 1 01/12/2…               254     0          -5.2               37            2.2
## 2 01/12/2…               204     1          -5.5               38            0.8
## 3 01/12/2…               173     2          -6                 39            1
## 4 01/12/2…               107     3          -6.2               40            0.9
## 5 01/12/2…                78     4          -6                 36            2.3
## 6 01/12/2…               100     5          -6.4               37            1.5
## # ℹ 11 more variables: visibility_10m <dbl>, dew_point_temperature_c <dbl>,
## #   solar_radiation_mj_m2 <dbl>, rainfall_mm <dbl>, snowfall_cm <dbl>,
## #   seasons <fct>, holiday <fct>, year <dbl>, month <dbl>, day <int>,
## #   day_of_week <chr>
```