XỬ LÝ SỐ LIỆU THỐNG KÊ - FINAL PROJECT

PROJECT - 2

2024-07-06

Các thành viên nhóm:

1. Trần Thị Bích Tuyền - 21280059
2. Nguyễn Thị Yến Như - 21280082
3. Nguyễn Hoài Linh - 21280097
4. Lê Nguyễn Hoàng Uyên - 21280118
5. Trần Trịnh Mai Vy - 21280122

Giới Thiệu.

Dịch vụ cho thuê xe đạp đã được triển khai ở nhiều thành phố lớn trên thế giới nhằm nâng cao sự thoải mái và tiện lợi trong việc di chuyển. Tại Seoul, dịch vụ cho thuê xe đạp không chỉ giúp giảm thiểu thời gian chờ đợi mà còn đóng vai trò quan trọng trong việc giảm tắc nghẽn giao thông và bảo vệ môi trường.

Một thách thức lớn đối với các nhà quản lý dịch vụ là đảm bảo nguồn cung cấp xe đạp ổn định và đủ để đáp ứng nhu cầu của công chúng vào mọi thời điểm. Việc dự đoán chính xác số lượng xe đạp cần thiết mỗi giờ là yếu tố then chốt để cung cấp dịch vụ hiệu quả và giảm thiểu thời gian chờ đợi của khách hàng.

Với dữ liệu `SeoulBikeData.csv` đã cho trước. Mục tiêu của dự án này là phân tích và dự đoán số lượng xe đạp cần thiết mỗi giờ để đảm bảo cung cấp dịch vụ thuê xe đạp ổn định, góp phần nâng cao trải nghiệm của người sử dụng và tối ưu hóa hoạt động kinh doanh của công ty cung cấp dịch vụ.

Bảng đề xuất phân tích và xử lý số liệu.

Bảng đề xuất phân tích dữ liệu.

Đề xuất	Mô tả	Phương pháp
Phân tích các yếu tố ảnh hưởng đến số lượng xe đạp được thuê.	Phân tích các yếu tố như thời gian, môi trường, sự kiện để kiểm tra nó ảnh hưởng đến số lượng xe đạp như thế	Đồ thị boxplot

	nào	
Phân tích sự độc lập giữa các nhóm dữ liệu.	Sự độc lập của số xe thuê theo ngày lễ, theo mùa, các ngày trong tuần, ...	A/B Testing
Xây dựng các bảng thống kê.	Xây dựng các bản thống kê với các thuộc tính để tìm ra min, max, trung bình, ... của các thuộc tính.	Các biến trong thống kê.

Bảng đề xuất xử lý số liệu.

Đề xuất	Mô tả	Phương pháp
Loại bỏ các thuộc tính không cần thiết.	Các thuộc tính không sử dụng, có một giá trị, ...	
Chuyển đổi, tách dữ liệu	Chuyển đổi các biến dữ liệu định tính sang dạng số. Trích suất các thuộc tính cần thiết như ngày, tháng, năm.	
Tạo các thuộc tính mới.	Gom cụm các dữ liệu có chung ngày, để giảm bớt số lượng mẫu.	

Bảng đề xuất xây dựng mô hình

Đề xuất	Mô tả	Phương pháp
Xây dựng mô hình tuyến tín	Sử dụng các thuộc tính có sẵn để xây dựng mô hình hồi quy tuyến tính về dự đoán số lượng xe đạp thuê.	Hồi quy tuyến tính
Lựa chọn thuộc tính.	Sử dụng CV hoặc hồi quy từng phần để lựa chọn các thuộc tính có ý nghĩa về mặt thống kê cho mô hình.	Cross validation, Hồi quy từng phần.
Kiểm tra giả định	Kiểm tra các đồ thị residual vs fit, normal Q-Q plot, ... Để kiểm tra các giả định như. homoscedasticity, normality	Chuẩn đoán mô hình, kiểm tra tuyến tính, đồ thị thặng dư.
Mở rộng mô hình.	Sử dụng các phương pháp hồi quy khác như hồi quy đa thức, hồi quy đa thức từng phần để cải thiện các giả đinh.	Hồi quy đa thức, Hồi quy đa thực từng

		phần.

1. Import library

```
library(boot)
```

```
## Warning: package 'boot' was built under R version 4.3.3
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.0     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(matrixStats)
```

```
## Warning: package 'matrixStats' was built under R version 4.3.3
```

```
##
## Attaching package: 'matrixStats'
##
## The following object is masked from 'package:dplyr':
##
##     count
```

```
library(lubridate)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.3.3
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(ggplot2)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```r
library(dplyr)
library(splines2)
```

```
## Warning: package 'splines2' was built under R version 4.3.3
```

```r
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 4.3.3
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
library(mgcViz)
```

```
## Warning: package 'mgcViz' was built under R version 4.3.3
```

```
## Loading required package: qgam
```

```
## Warning: package 'qgam' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## Warning in check_dep_version(): ABI version mismatch:
## lme4 was built with Matrix ABI version 1
## Current Matrix ABI version is 0
## Please re-install lme4 from source or restore original 'Matrix' package
```

```
## Registered S3 method overwritten by 'mgcViz':
##   method from
##   +.gg   GGally
##
## Attaching package: 'mgcViz'
##
## The following objects are masked from 'package:stats':
##
##     qqline, qqnorm, qqplot
```

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.3.3
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(grid)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(leaps)
library(lmPerm)
```

```
## Warning: package 'lmPerm' was built under R version 4.3.3
```

```r
library(corrplot)
library(splines2)
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.3.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:mgcViz':
##
##     qq
##
## The following object is masked from 'package:boot':
##
##     melanoma
##
##
```

```
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(mgcv)
library(mgcViz)
library(splines)
library(lmPerm)
library(readr)
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.3.3
```

```
## Thank you for using fastDummies!
## To acknowledge our work, please cite the package:
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy
(Binary) Columns and Rows from Categorical Variables. Version 1.7.1. URL:
https://github.com/jacobkap/fastDummies,
https://jacobkap.github.io/fastDummies/.
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
##
## The following object is masked from 'package:boot':
##
##     logit
```

2. Read data

```
locale=locale(encoding="latin1")
data <- read_csv("SeoulBikeData.csv", locale = locale)
```

```
## Rows: 8760 Columns: 14
## ── Column specification
──────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): Date, Seasons, Holiday, Functioning Day
## dbl (10): Rented Bike Count, Hour, Temperature(°C), Humidity(%), Wind
speed ...
##
## 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.
```

```
head(data)
```

```
## # A tibble: 6 × 14
##   Date       `Rented Bike Count`  Hour `Temperature(°C)` `Humidity(%)`
##   <chr>                    <dbl> <dbl>             <dbl>         <dbl>
## 1 01/12/2017                 254     0              -5.2            37
## 2 01/12/2017                 204     1              -5.5            38
## 3 01/12/2017                 173     2              -6              39
## 4 01/12/2017                 107     3              -6.2            40
## 5 01/12/2017                  78     4              -6              36
## 6 01/12/2017                 100     5              -6.4            37
## # ℹ 9 more variables: `Wind speed (m/s)` <dbl>, `Visibility (10m)` <dbl>,
## #   `Dew point temperature(°C)` <dbl>, `Solar Radiation (MJ/m2)` <dbl>,
## #   `Rainfall(mm)` <dbl>, `Snowfall (cm)` <dbl>, Seasons <chr>, Holiday
<chr>,
## #   `Functioning Day` <chr>
```

```
data <- data |> janitor::clean_names()
glimpse(data)
```

```
## Rows: 8,760
## Columns: 14
## $ date                   <chr> "01/12/2017", "01/12/2017", "01/12/2017",
"01/…
## $ rented_bike_count      <dbl> 254, 204, 173, 107, 78, 100, 181, 460,
930, 49…
## $ hour                   <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, …
## $ temperature_c          <dbl> -5.2, -5.5, -6.0, -6.2, -6.0, -6.4, -6.6,
-7.4…
## $ humidity_percent       <dbl> 37, 38, 39, 40, 36, 37, 35, 38, 37, 27,
24, 21…
## $ wind_speed_m_s         <dbl> 2.2, 0.8, 1.0, 0.9, 2.3, 1.5, 1.3, 0.9,
1.1, 0…
## $ visibility_10m         <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000,
2000…
## $ dew_point_temperature_c <dbl> -17.6, -17.6, -17.7, -17.6, -18.6, -18.7,
-19.…
## $ solar_radiation_mj_m2  <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
```

```
0.00…
## $ rainfall_mm             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ snowfall_cm             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ seasons                 <chr> "Winter", "Winter", "Winter", "Winter",
"Winte…
## $ holiday                 <chr> "No Holiday", "No Holiday", "No Holiday",
"No …
## $ functioning_day         <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
"Yes…
```

3. EDA.

3.1 Bảng tổng hợp dữ liệu.

```
summary(data)
```

```
##      date           rented_bike_count      hour       temperature_c
##  Length:8760        Min.   :   0.0    Min.   : 0.00   Min.   :-17.80
##  Class :character   1st Qu.: 191.0    1st Qu.: 5.75   1st Qu.:  3.50
##  Mode  :character   Median : 504.5    Median :11.50   Median : 13.70
##                     Mean   : 704.6    Mean   :11.50   Mean   : 12.88
##                     3rd Qu.:1065.2    3rd Qu.:17.25   3rd Qu.: 22.50
##                     Max.   :3556.0    Max.   :23.00   Max.   : 39.40
##  humidity_percent wind_speed_m_s  visibility_10m dew_point_temperature_c
##  Min.   : 0.00    Min.   :0.000   Min.   :  27   Min.   :-30.600
##  1st Qu.:42.00    1st Qu.:0.900   1st Qu.: 940   1st Qu.: -4.700
##  Median :57.00    Median :1.500   Median :1698   Median :  5.100
##  Mean   :58.23    Mean   :1.725   Mean   :1437   Mean   :  4.074
##  3rd Qu.:74.00    3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 14.800
##  Max.   :98.00    Max.   :7.400   Max.   :2000   Max.   : 27.200
##  solar_radiation_mj_m2  rainfall_mm       snowfall_cm        seasons
##  Min.   :0.0000        Min.   : 0.0000   Min.   :0.00000   Length:8760
##  1st Qu.:0.0000        1st Qu.: 0.0000   1st Qu.:0.00000   Class
:character
##  Median :0.0100        Median : 0.0000   Median :0.00000   Mode
:character
##  Mean   :0.5691        Mean   : 0.1487   Mean   :0.07507
##  3rd Qu.:0.9300        3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :3.5200        Max.   :35.0000   Max.   :8.80000
##    holiday          functioning_day
##  Length:8760        Length:8760
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

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.

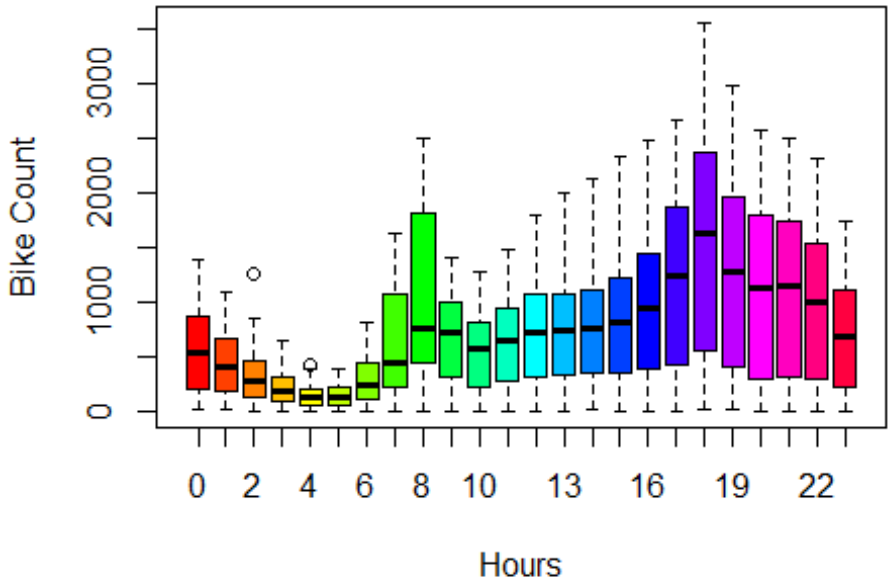
```
data <- data %>%
  mutate(
    Date = dmy(date),
    # Chuyển đổi cột 'date' sang định dạng ngày tháng
    year = year(Date),
    month = month(Date),
    day = day(Date)
  )
# Loại bỏ cột 'Date'
data <- subset(data, select = -c(Date))
# Tạo cột day bằng cách kết hợp year, month và day
data$full_date <- 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>
```

3.3 Trực quan dữ liệu.

3.3.1 Vẽ biểu đồ boxplot nhằm xem xét phân phối của số lượng xe đạp được thuê (rented_bike_count) vào từng giờ trong ngày (hour)

```
# Vẽ biểu đồ boxplot
boxplot(data$rented_bike_count ~ data$hour,
        xlab = "Hours",
        ylab = "Bike Count",
        col = rainbow(length(unique(data$hour))))
```

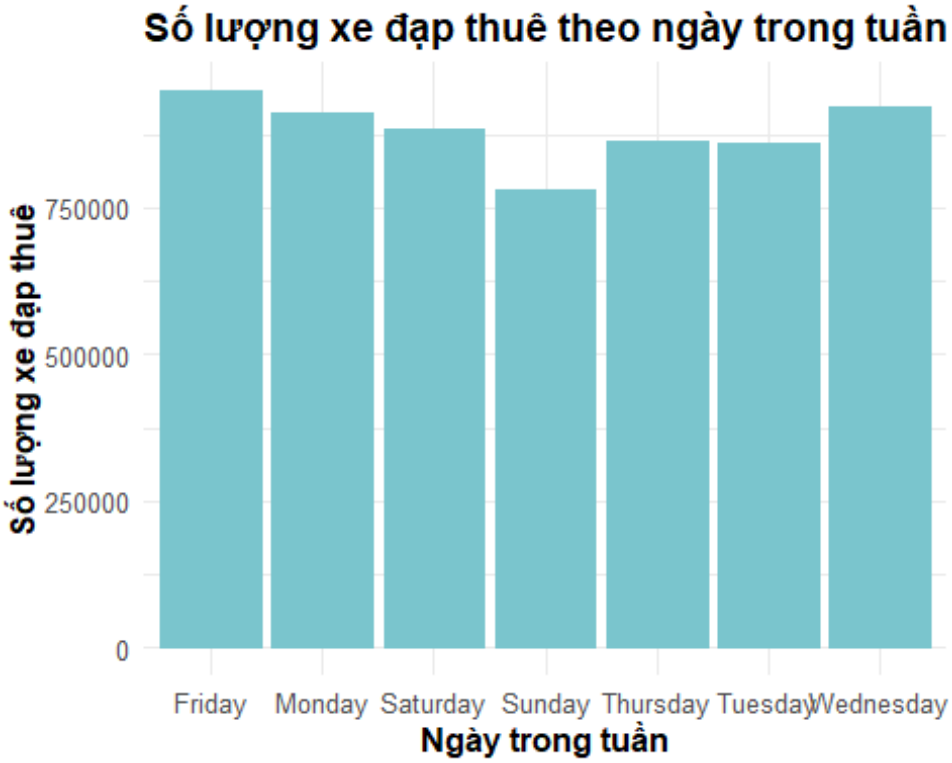


Biểu đồ ở trên cho thấy số lượng xe đạp được thuê thay đổi theo giờ. Dễ dàng nhận thấy từ biểu đồ rằng nhiều xe đạp được thuê nhất từ 7 giờ sáng đến 9 giờ sáng. Sau đó, số lượng xe thuê duy trì tương đối ổn định cho đến 15:00 chiều và bắt đầu tăng trở lại sau 15:00 chiều.

3.3.2 Vẽ biểu đồ thanh cho số lượng xe đạp thuê theo ngày trong tuần.

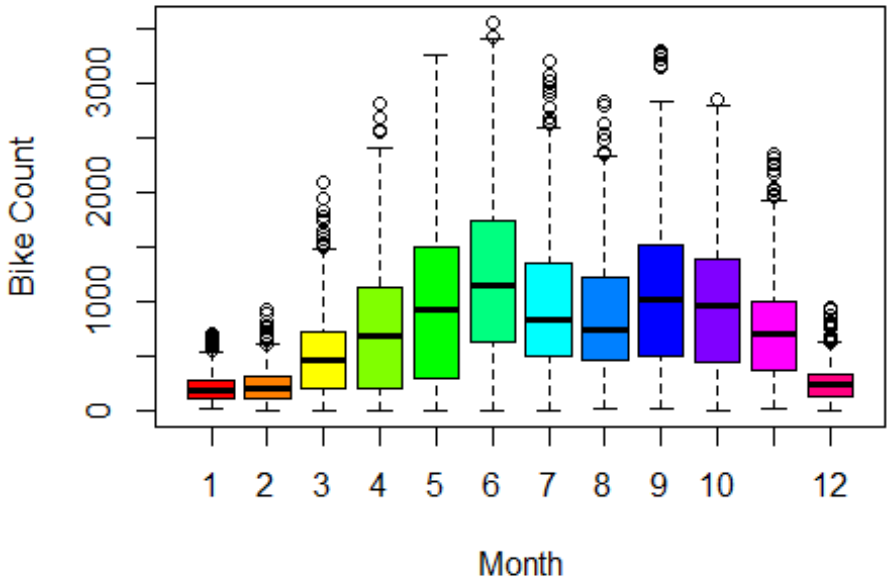
```
# Biểu đồ thanh cho số lượng xe đạp thuê theo ngày trong tuần
ggplot(data, aes(x = day_of_week, y = rented_bike_count, fill = day_of_week))
+
  geom_bar(stat = "identity", color = "#7AC5CD") +
  labs(
    title = "Số lượng xe đạp thuê theo ngày trong tuần",
    x = "Ngày trong tuần",
    y = "Số lượng xe đạp thuê"
  ) +
  theme_minimal() +
  theme(
```

```
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.position = "none"
  ) + scale_fill_brewer(palette = "Set3")
```



3.3.3 Vẽ biểu đồ boxplot cho số lượng xe đạp thuê theo tháng trong năm.

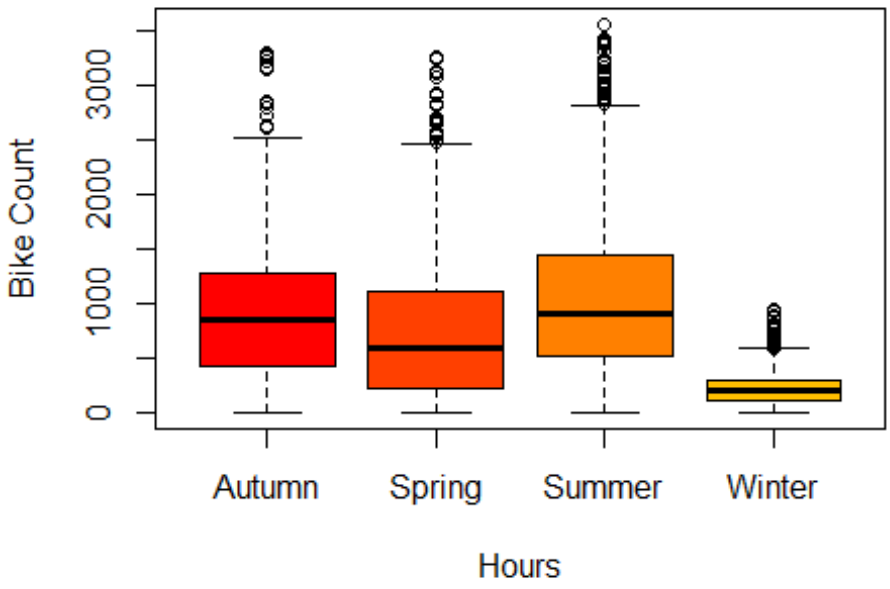
```
# Vẽ biểu đồ boxplot
boxplot(data$rented_bike_count ~ data$month,
        xlab = "Month",
        ylab = "Bike Count",
        col = rainbow(length(unique(data$month))))
```



=> Nhận xét: Qua biểu đồ ta thấy tháng 6 là tháng có số lượng xe đạp thuê cao nhất, trong khi tháng 1 và tháng 12 là các tháng có số lượng xe đạp thuê thấp nhất.

3.3.4 Vẽ biểu đồ boxplot cho số lượng xe đạp thuê theo mùa.

```
# Vẽ biểu đồ boxplot
boxplot(data$rented_bike_count ~ data$seasons,
        xlab = "Hours",
        ylab = "Bike Count",
        col = rainbow(length(unique(data$hour))))
```

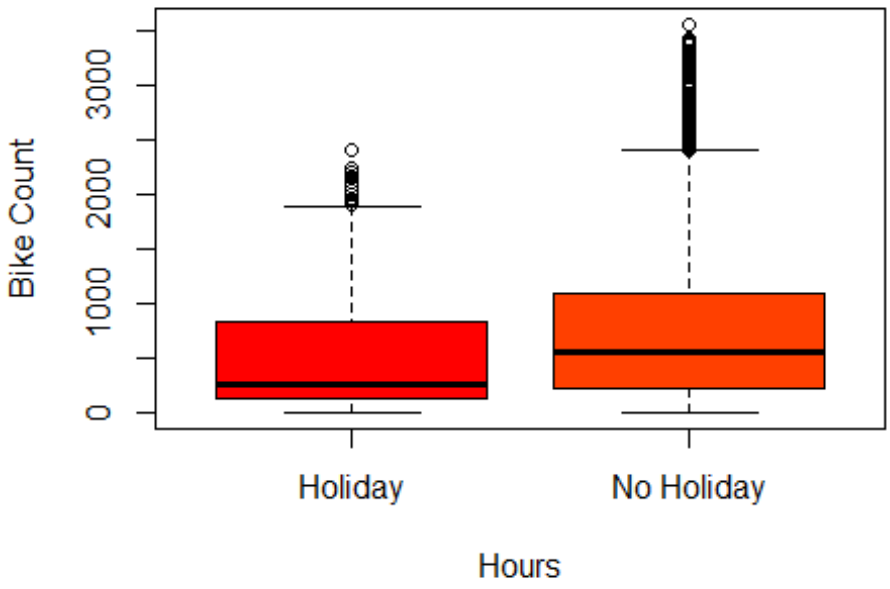


=> **Nhận xét**: Ta có thể thấy rằng trong những thời điểm thời tiết tốt hơn như mùa hè, mọi người thường thích đạp xe hơn và vào mùa đông xe đạp được thuê với giá thấp nhất với số lượng nhu cầu thấp hơn 500 chiếc. Điều này có thể là do thời tiết lạnh cũng như tuyết trong mùa đông. Có một số ngoại lệ với mật độ cao hơn các mùa còn lại khi nhu cầu xe đạp trong Mùa đông là trên 500. Đây có thể là một số lý do cơ bản chẳng hạn như ngày có thời tiết tốt hơn những ngày khác trong mùa đông. + Vào mùa thu, nhu cầu xe đạp cao hơn mùa xuân mặc dù chúng tôi dự kiến nhu cầu vào mùa xuân sẽ cao hơn mùa thu. Điều này có thể là do mùa Xuân có nhiều ngày mưa hơn mùa Thu.

Việc sử dụng xe đạp cao nhất vào mùa thu và mùa xuân, với trung vị và sự biến đổi tương tự nhau. Mùa hè có trung vị thấp hơn nhưng sự biến đổi số lượng xe đạp cao hơn. Mùa đông cho thấy việc sử dụng xe đạp thấp nhất với sự biến đổi ít nhất.

3.3.5 Vẽ biểu đồ boxplot cho số lượng xe đạp thuê theo ngày lễ.

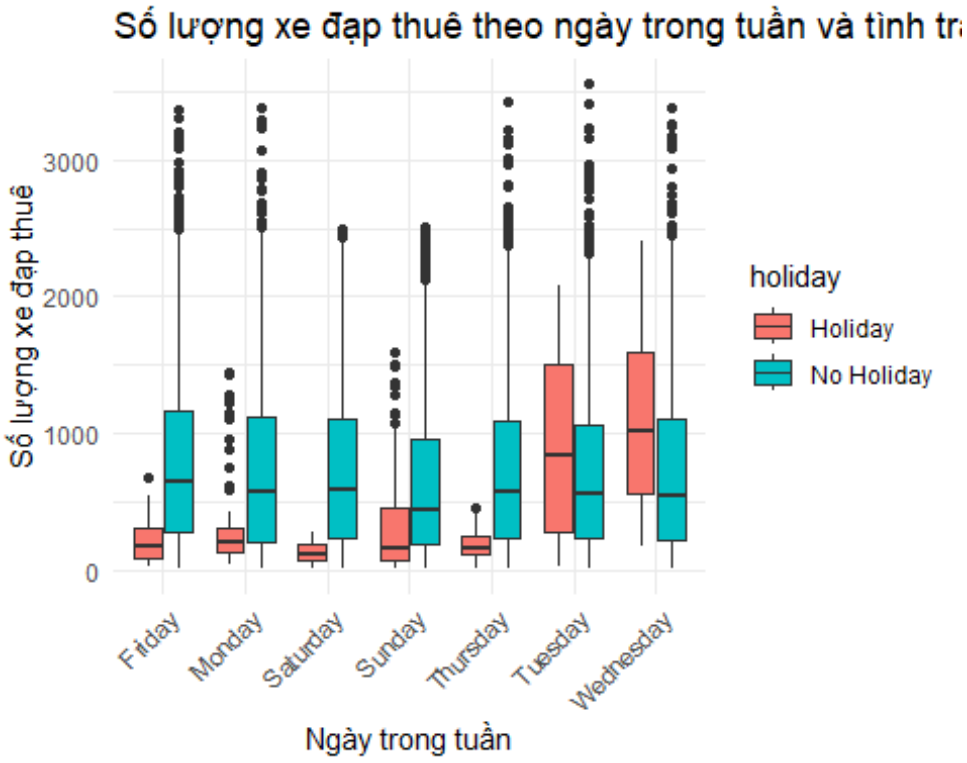
```
# Vẽ biểu đồ boxplot
boxplot(data$rented_bike_count ~ data$holiday,
        xlab = "Hours",
        ylab = "Bike Count",
        col = rainbow(length(unique(data$hour))))
```



=> **Nhận xét**: Ta có thấy rằng nhu cầu về xe đạp cao hơn vào những ngày không phải ngày nghỉ lễ, điều đó có nghĩa là hầu hết người dùng có thể thuê xe đạp vì những lý do khác ngoài mục đích giải trí.

3.3.6 Vẽ biểu đồ boxplot cho số lượng xe đạp thuê theo ngày trong tuần phân loại ngày lễ.

```
ggplot(data, aes(x = day_of_week, y = rented_bike_count, fill = holiday)) +
  geom_boxplot() +
  labs(title = "Số lượng xe đạp thuê theo ngày trong tuần và tình trạng ngày lễ",
       x = "Ngày trong tuần",
       y = "Số lượng xe đạp thuê") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



=> Nhận xét:

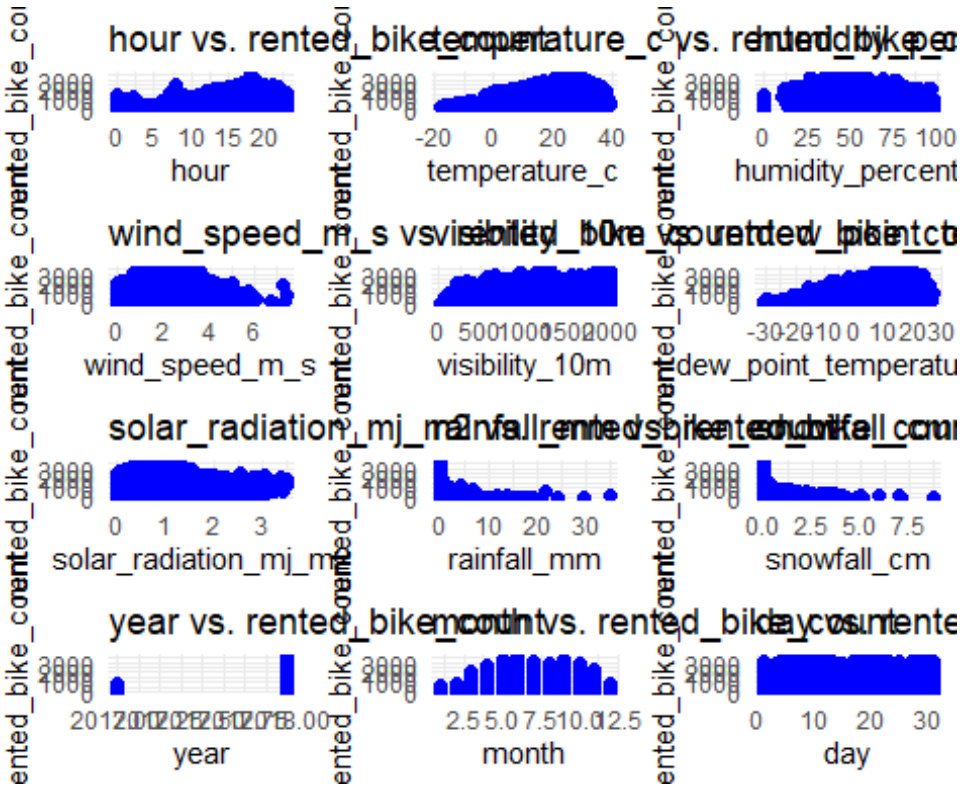
- Biểu đồ cho thấy số lượng xe đạp thuê vào các ngày lễ cuối tuần thường cao hơn so với ngày cuối tuần không phải ngày lễ . Ngày cuối tuần (thứ Bảy và Chủ Nhật) thường có nhiều hoạt động giải trí và du lịch, do đó, số lượng xe đạp thuê vào các ngày này cao và biến động lớn.
- Nếu một ngày lễ rơi vào cuối tuần, có thể tăng cường thêm hoạt động ngoài trời, dẫn đến số lượng xe đạp thuê cao hơn
- Ta cũng có thể thấy thứ Ba và thứ Tư là những ngày làm việc trong tuần. Khi ngày lễ trùng vào những ngày này, nhiều người có thể sử dụng thời gian nghỉ để tham gia các hoạt động ngoài trời, du lịch ngắn ngày hoặc đơn giản là thư giãn => sử dụng xe đạp tương đối lớn.

3.3.7 Biểu đồ tất cả các biến định lượng so với số lượng xe đạp thuê.

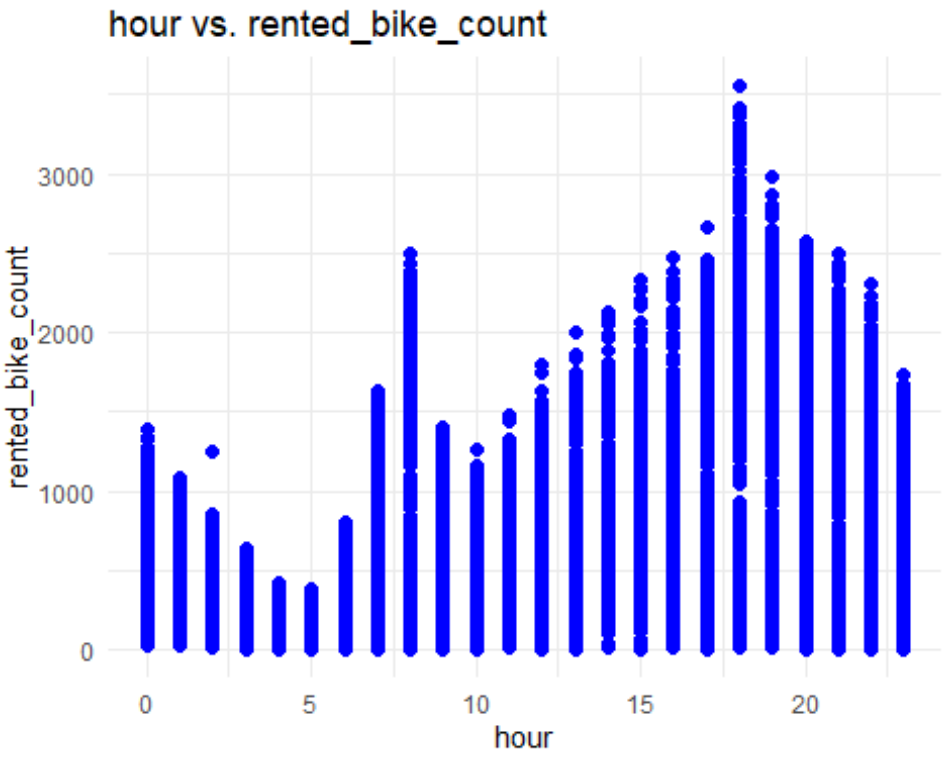
```
multi.scatter <- function(data, target) {
  # Initialize an empty list to store plots
  plots <- list()

  # Loop through each numeric variable
  for (col in names(data)) {
    # Check if current variable is numeric and not the target variable
    if (is.numeric(data[[col]]) && col != target) {
      # Create scatter plot
      scatter_plot <- ggplot(data, aes_string(x = col, y = target)) +
```

```
        geom_point(size = 2, color = "blue") +
        ggtitle(paste(col, "vs.", target)) +
        theme_minimal()

      # Add the plot to the list
      plots[[col]] <- scatter_plot
    }
  }

  # Arrange plots in a grid
  grid.arrange(grobs = plots, ncol = 3)  # Adjust ncol as needed

  # Return the list of plots (optional)
  return(plots)
}

# Call the function with numeric columns and target variable
multi.scatter(data, "rented_bike_count")
```

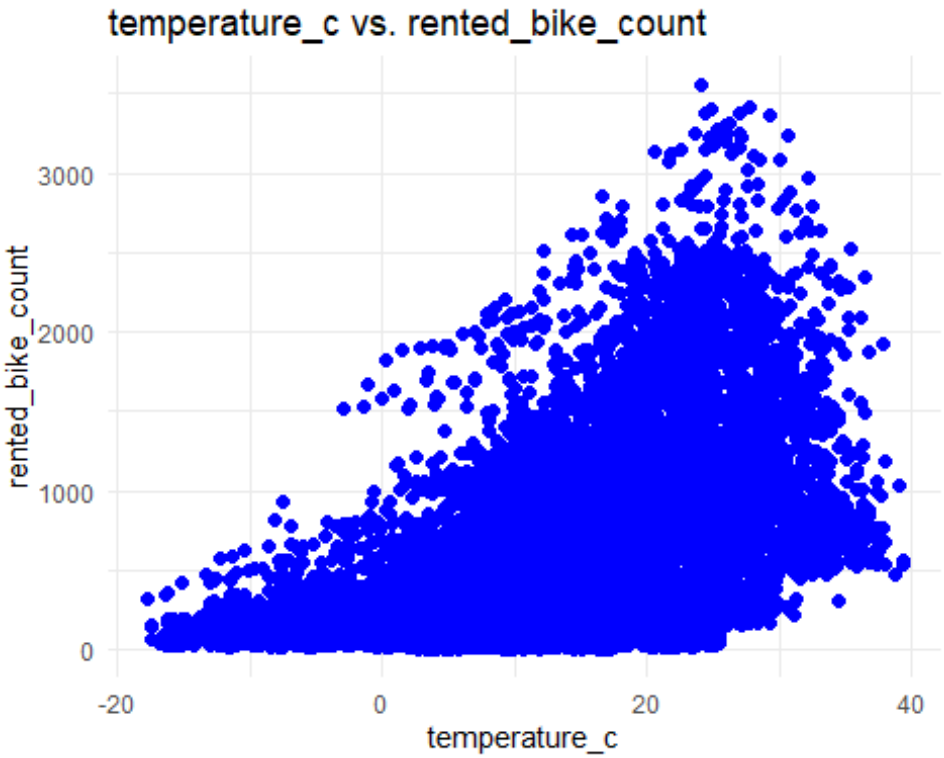
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



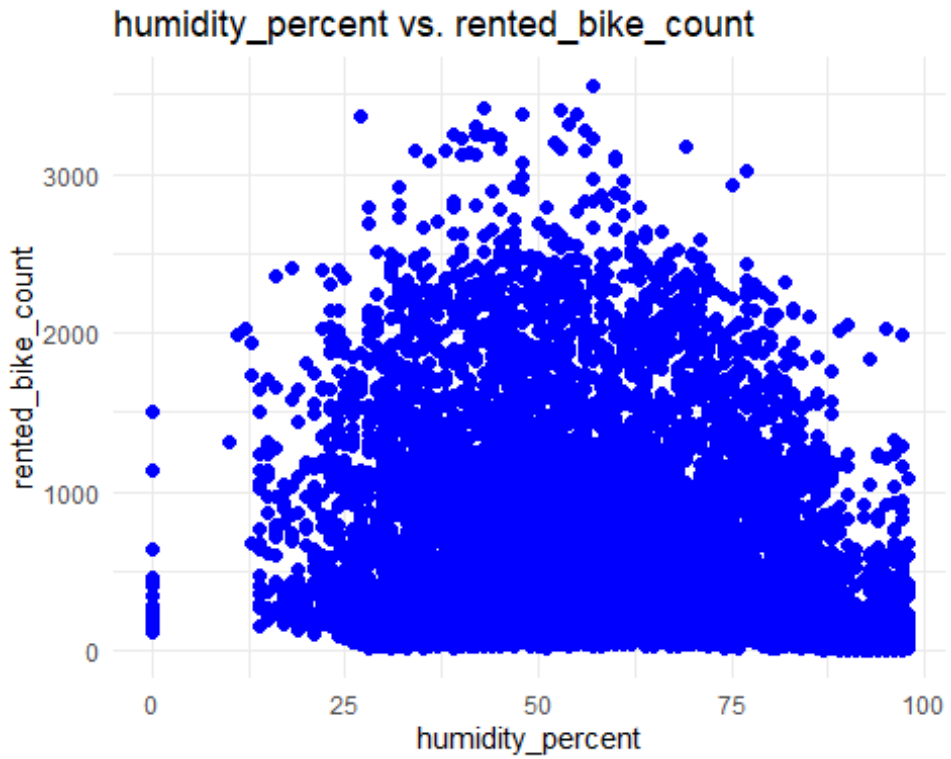

```
## $hour
```



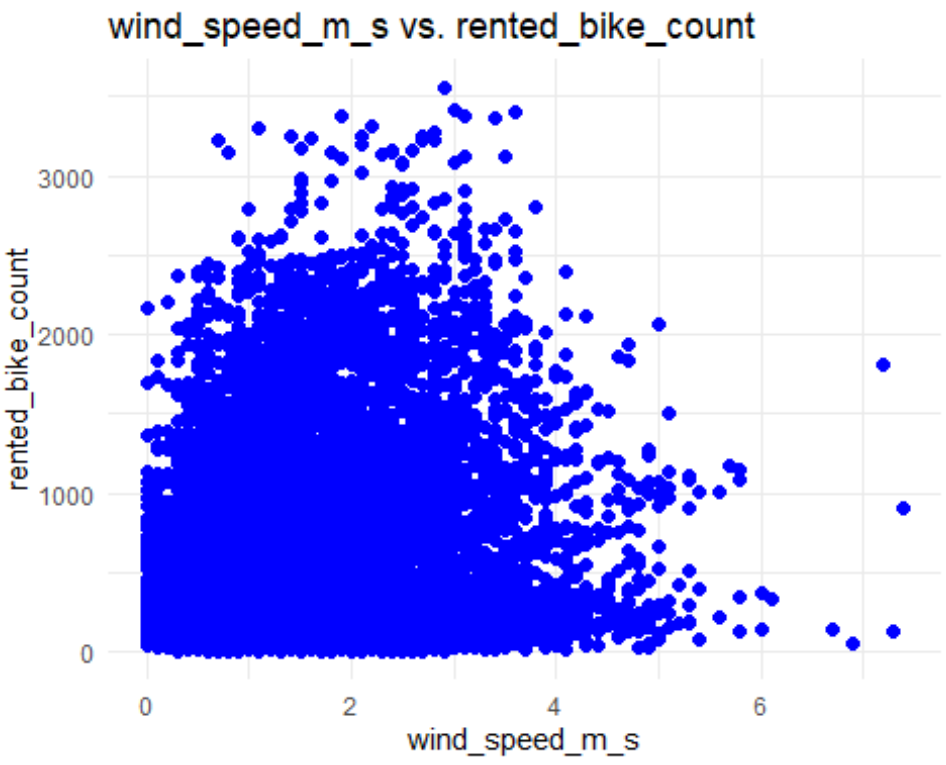

```
##
## $temperature_c
```



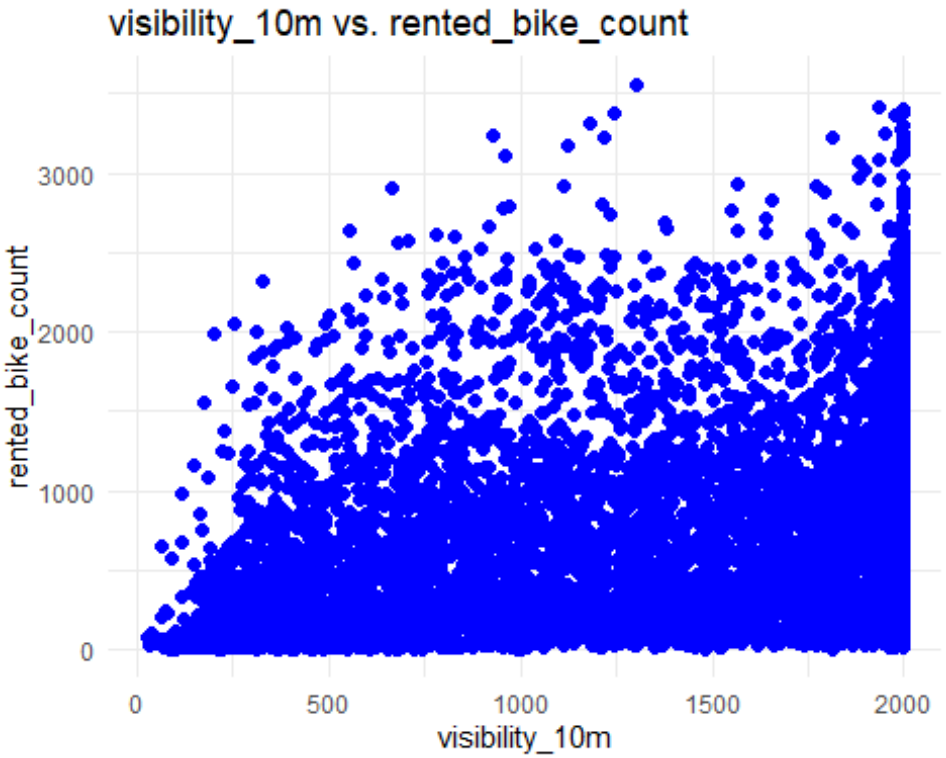
```
##
## $humidity_percent
```



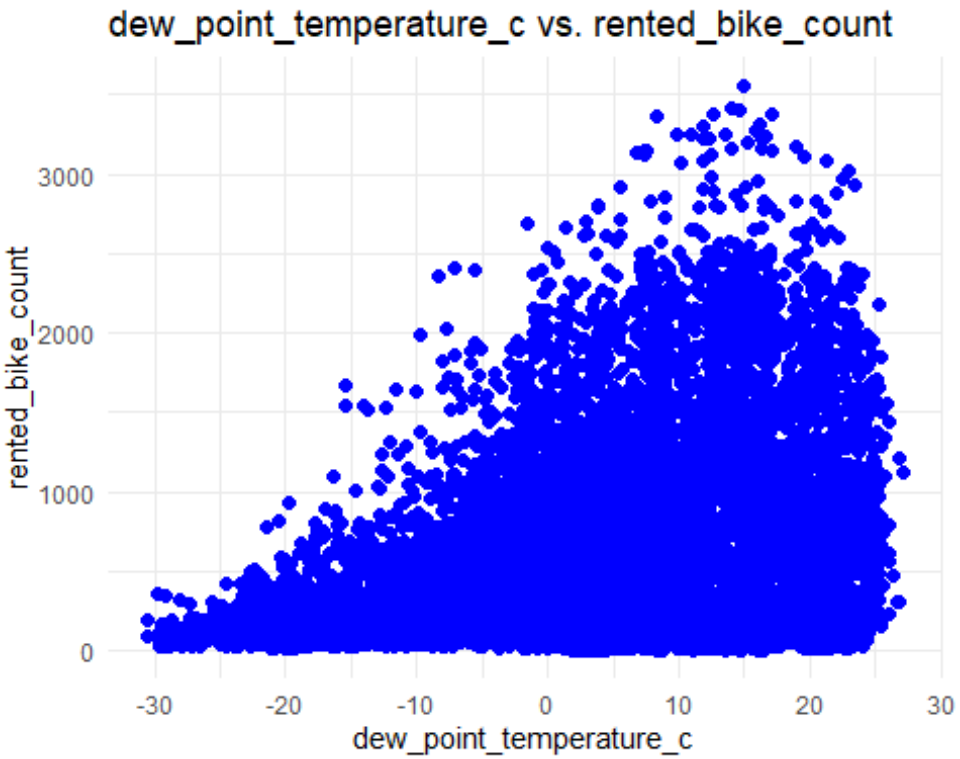

```
##
## $wind_speed_m_s
```



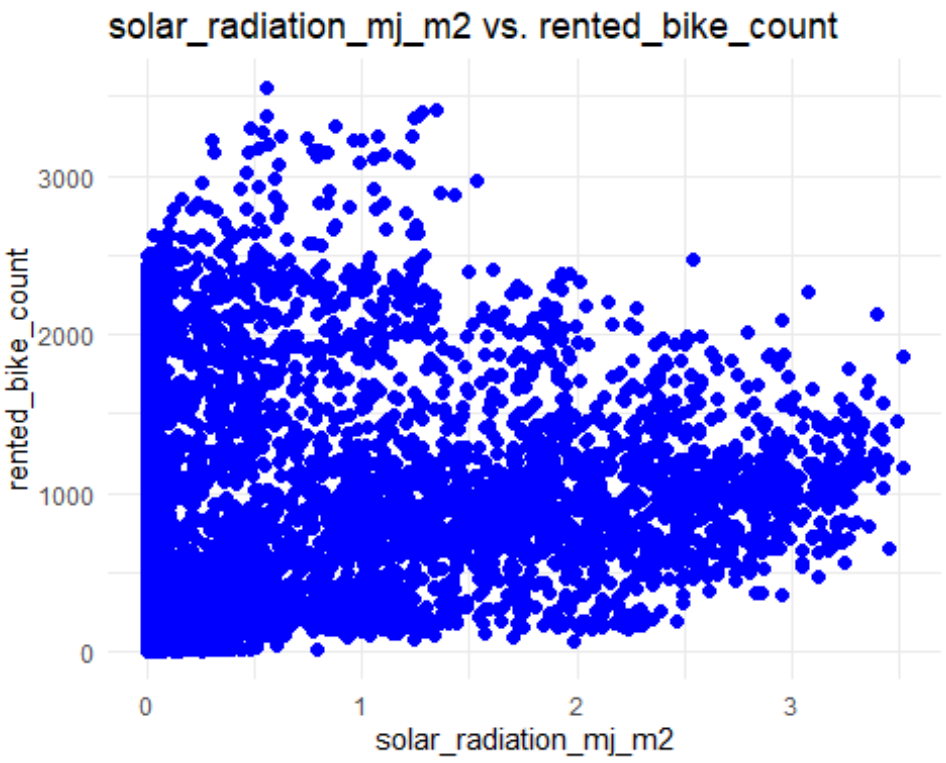

```
##
## $visibility_10m
```



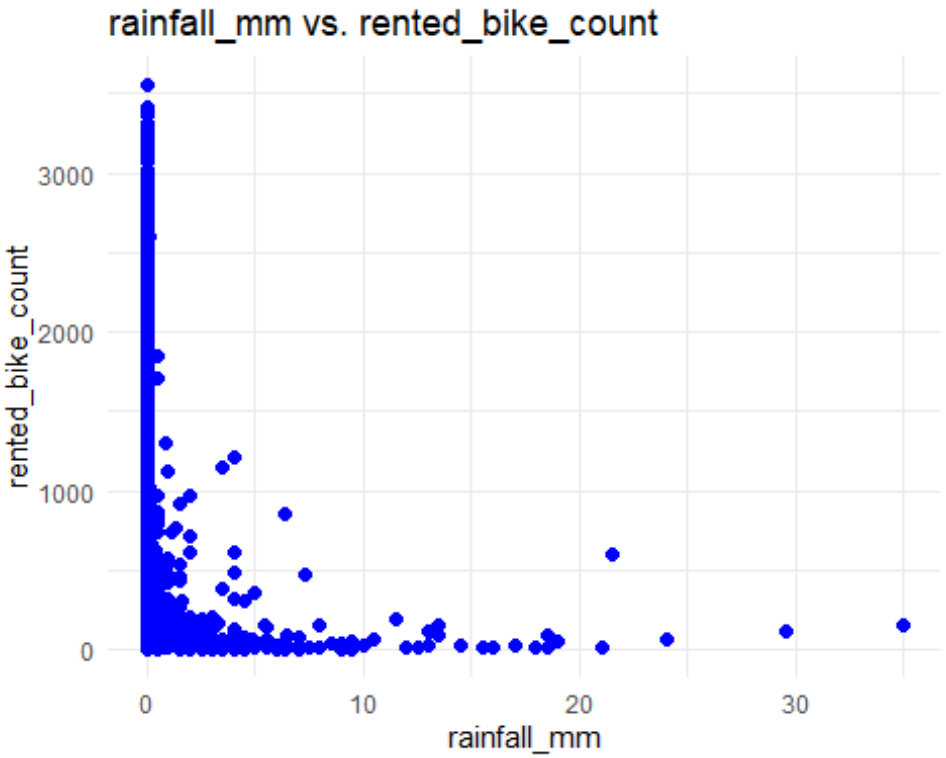
```
##
## $dew_point_temperature_c
```



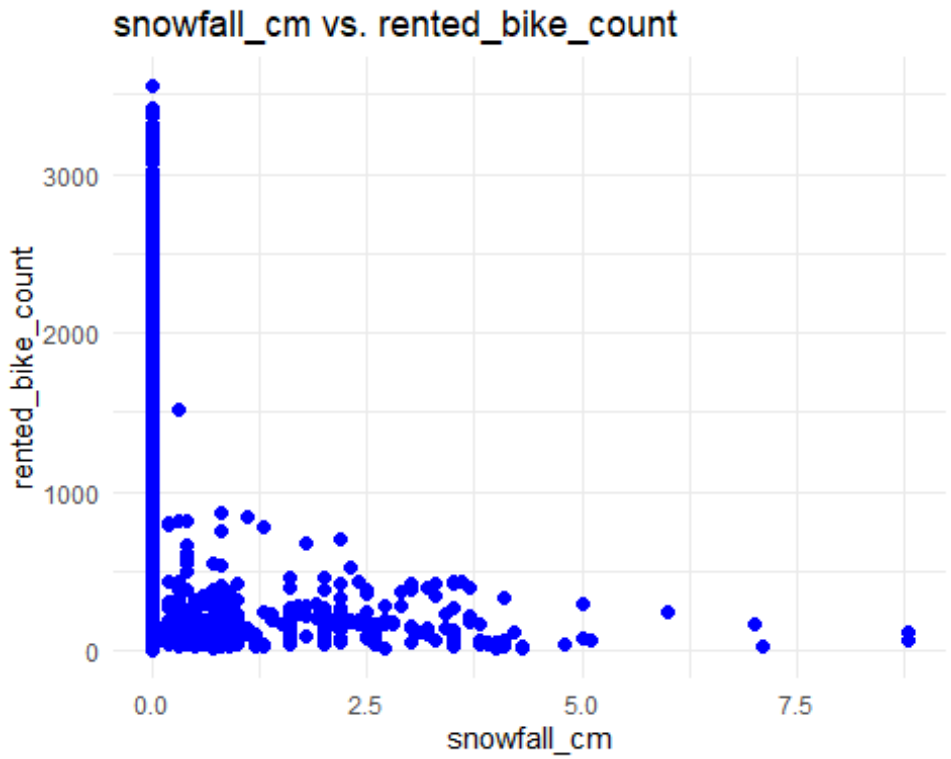

```
##
## $solar_radiation_mj_m2
```


solar_radiation_mj_m2 vs. rented_bike_count
rented_bike_count
3000
2000
1000
0
0
1
2
3
solar_radiation_mj_m2

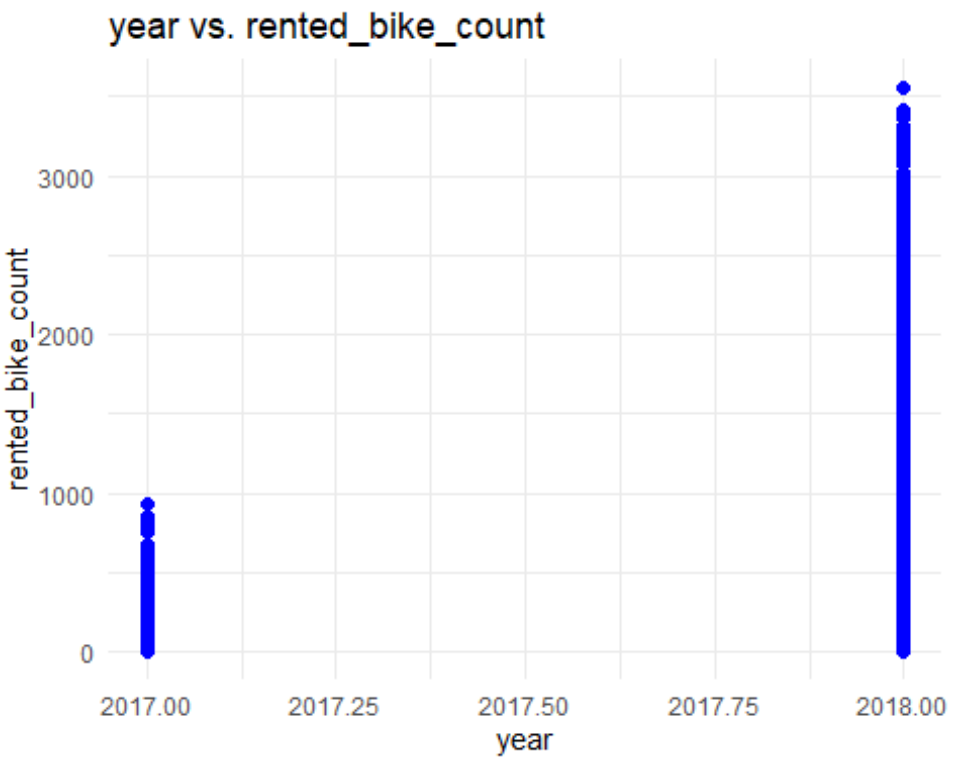
```
##
## $rainfall_mm
```


rainfall_mm vs. rented_bike_count
rented_bike_count
3000
2000
1000
0
0
10
20
30
rainfall_mm

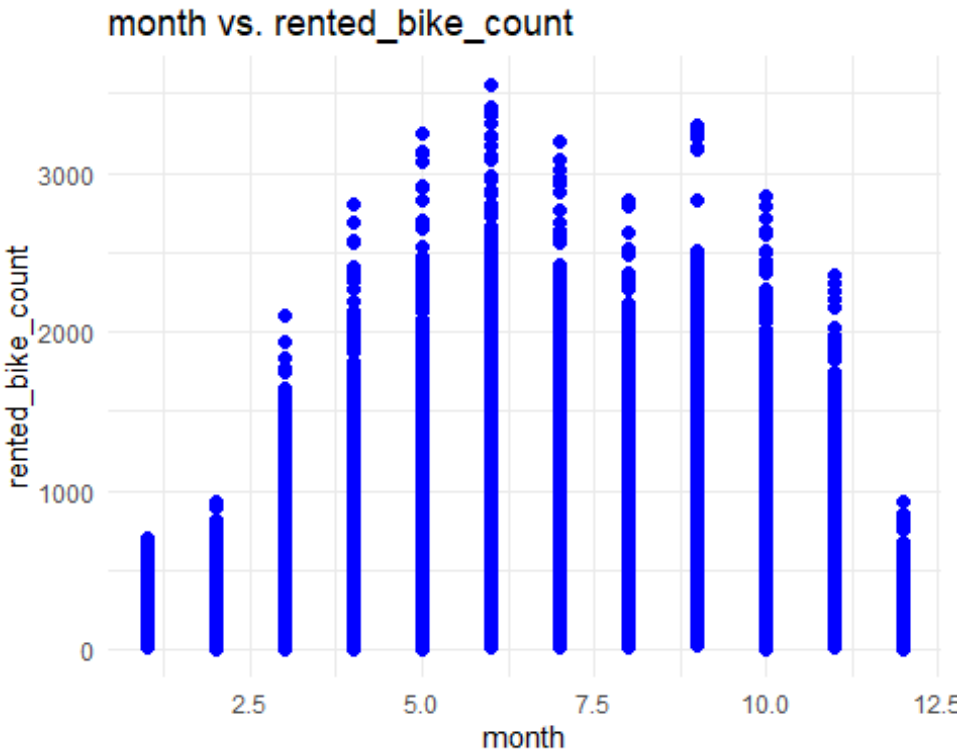
```
##
## $snowfall_cm
```



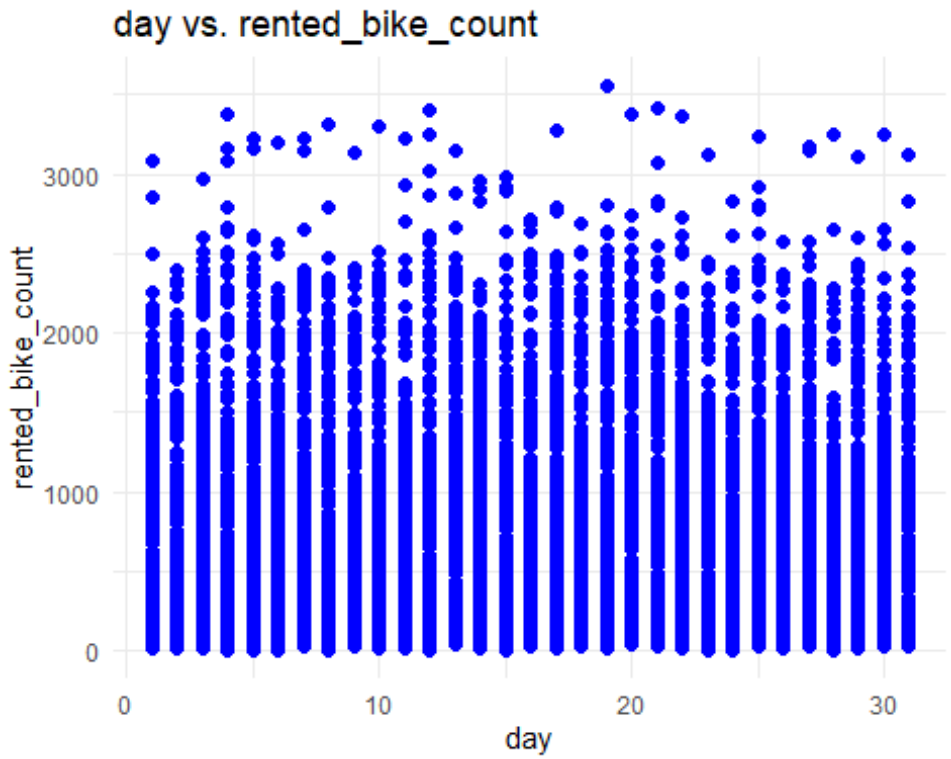

```
##
## $year
```


year vs. rented_bike_count
rented_bike_count
3000
2000
1000
0
2017.00
2017.25
2017.50
2017.75
2018.00
year

```
##
## $month
```


month vs. rented_bike_count
rented_bike_count
3000
2000
1000
0
2.5
5.0
7.5
10.0
12.5
month

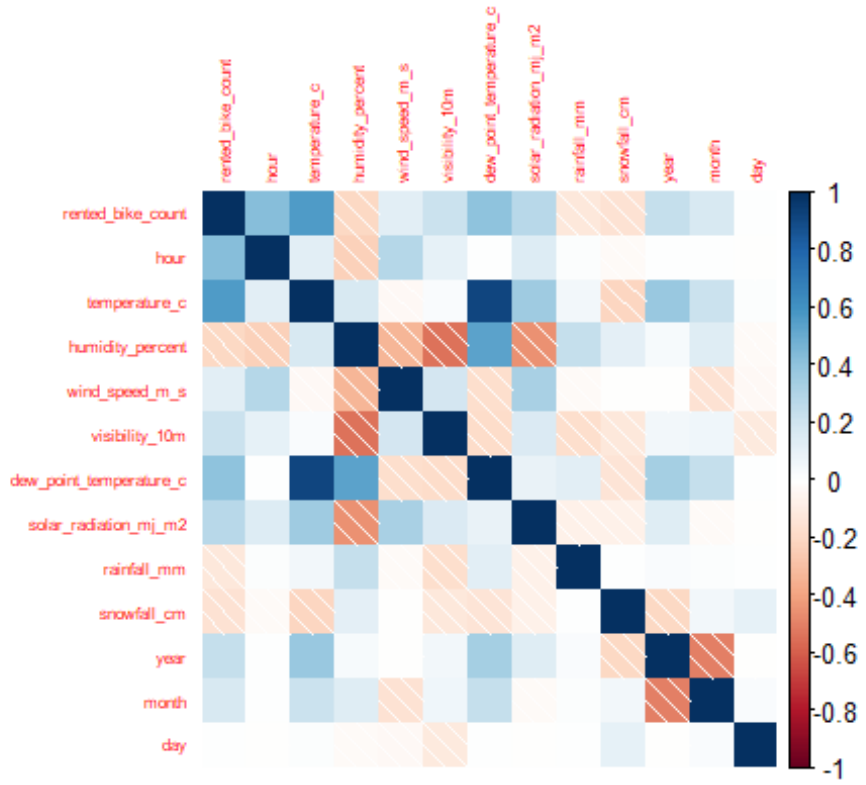
```
##
## $day
```



=> Nhận xét: Hầu hết tất cả các định lượng thể hiện thông tin môi trường bên ngoài điều chỉ là tuyến tính yếu, ngoài ra còn có độc lập tuyến tính.

3.3.8 Correlation Matrix

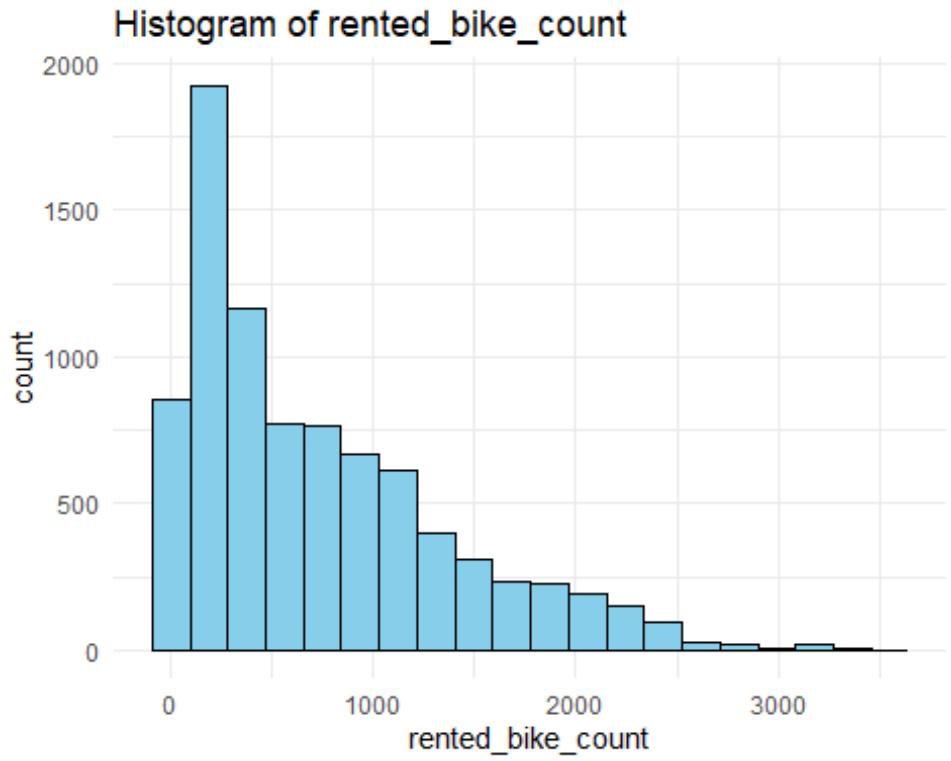
```
numeric_data <- data[, sapply(data, is.numeric)]
correlation_matrix <- cor(numeric_data)

# Load the corrplot package
library(corrplot)

# Plot the correlation matrix
corrplot(correlation_matrix, method = "shade",tl.cex = 0.5)
```



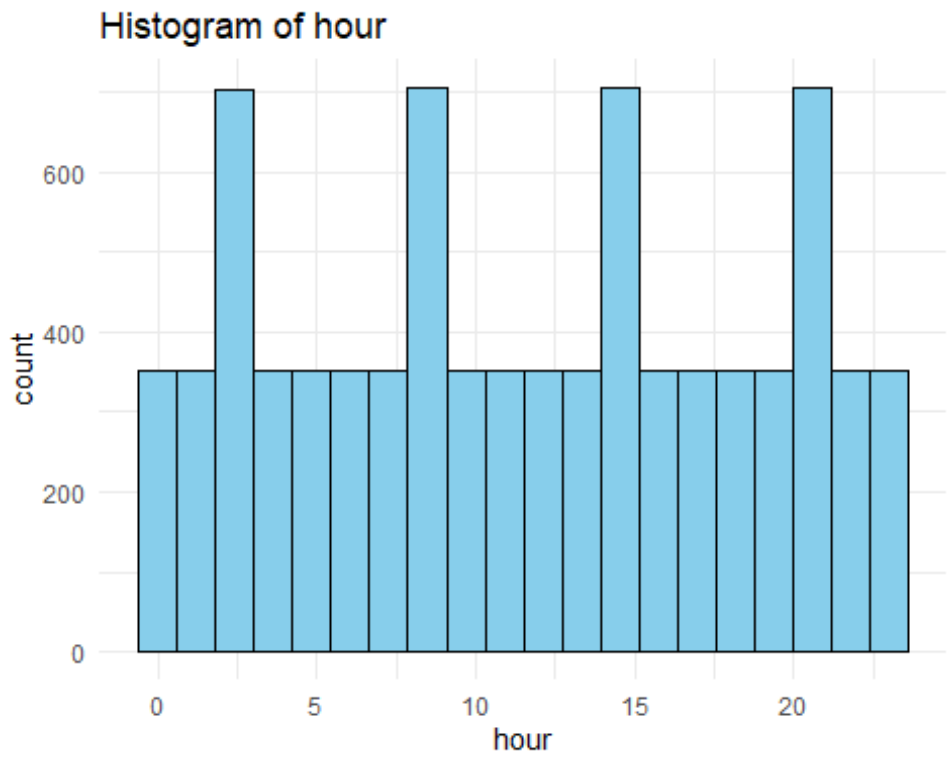

```r
# Function to create histograms
multi.hist <- function(data) {
  # Initialize an empty list to store plots
  plots <- list()

  # Loop through each numeric variable
  for (col in names(data)) {
    # Create histogram plot
    hist_plot <- ggplot(data, aes_string(x = col)) +
      geom_histogram(fill = "skyblue", color = "black", bins = 20) +  #
Customize fill and color
      ggtitle(paste("Histogram of", col)) +
      theme_minimal()

    # Add the plot to the list
    plots[[col]] <- hist_plot
  }

  # Arrange plots in a grid
  grid.arrange(grobs = plots, ncol = 3)  # Adjust ncol as needed

  # Return the list of plots (optional)
  return(plots)
}
```

```r
# Call the function with numeric columns of 'bike'
multi.hist(data[, sapply(data, is.numeric)])
```

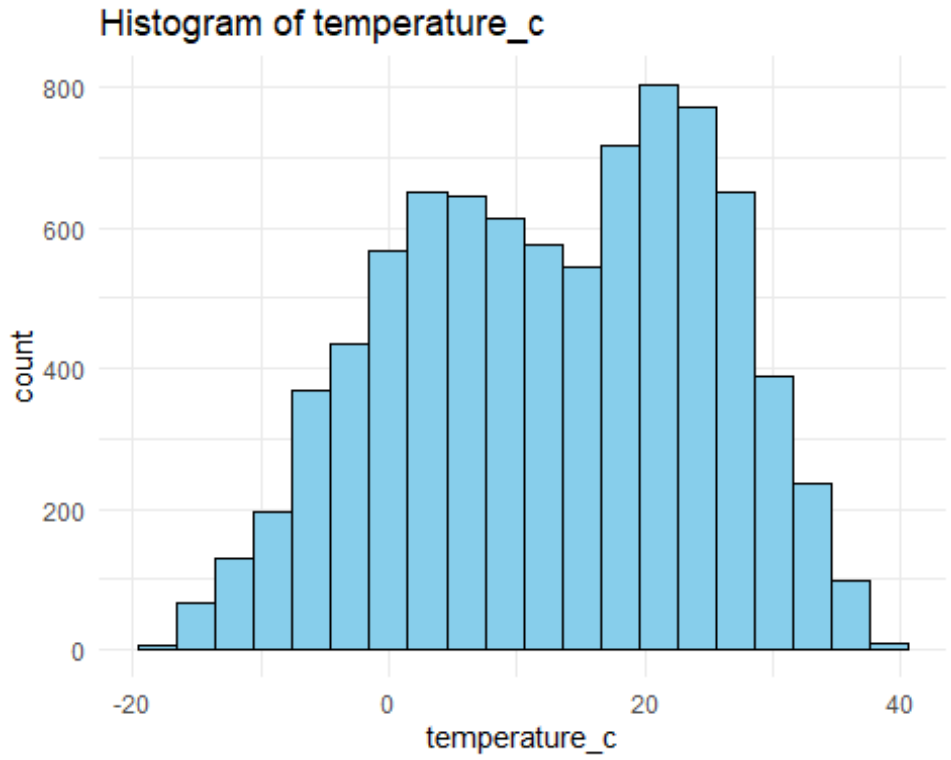
```
## $rented_bike_count
```



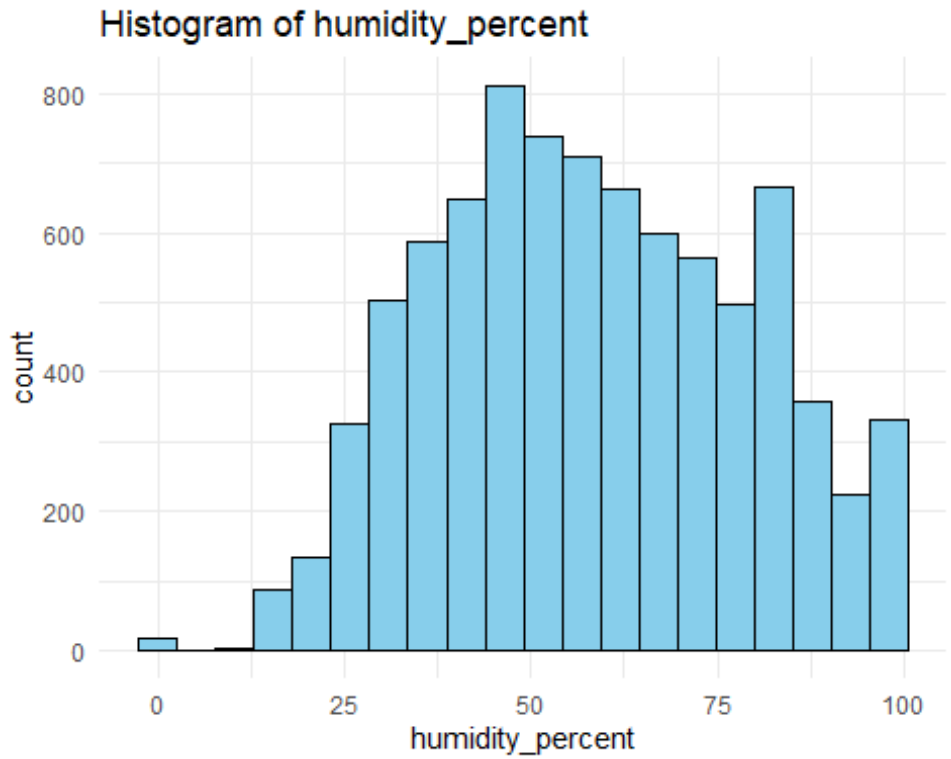

```
##
## $hour
```



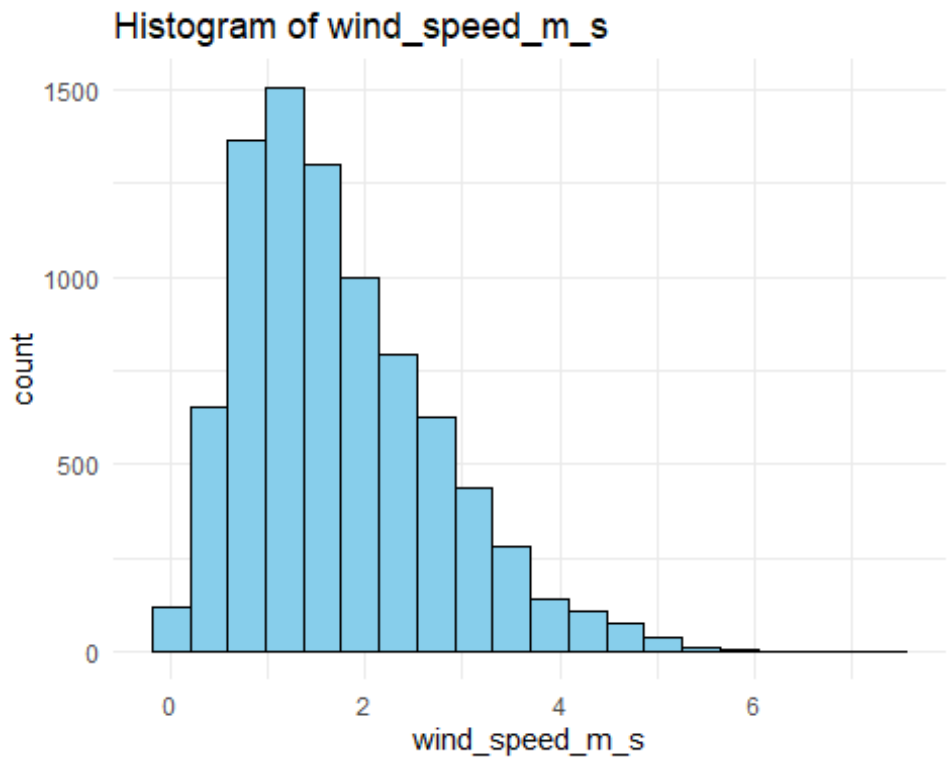

```
##
## $temperature_c
```



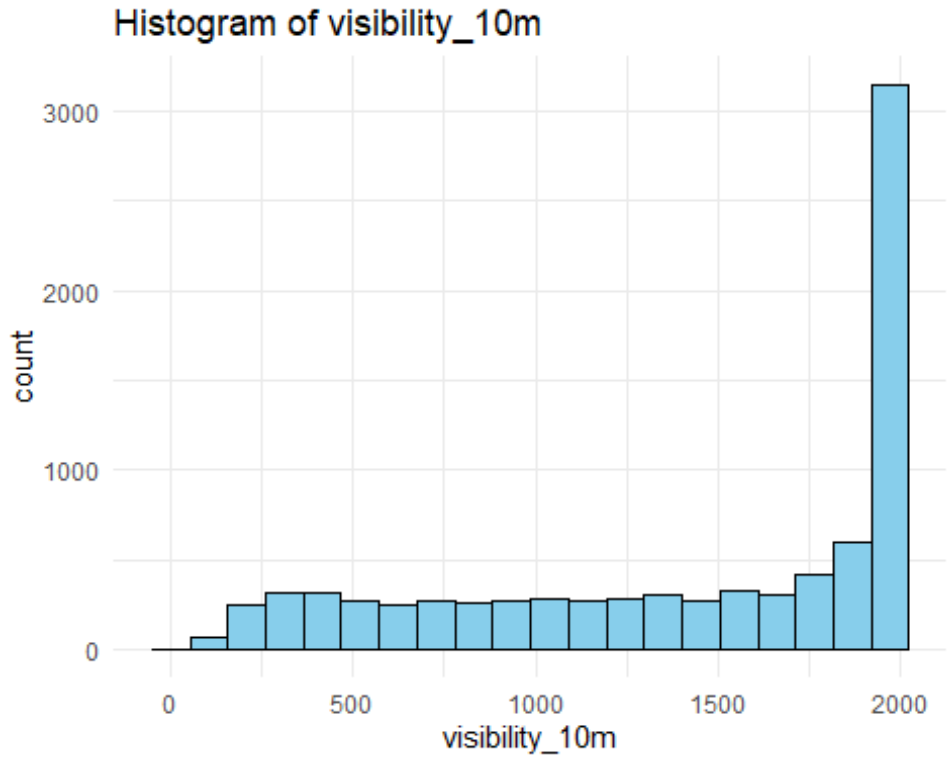
```
##
## $humidity_percent
```



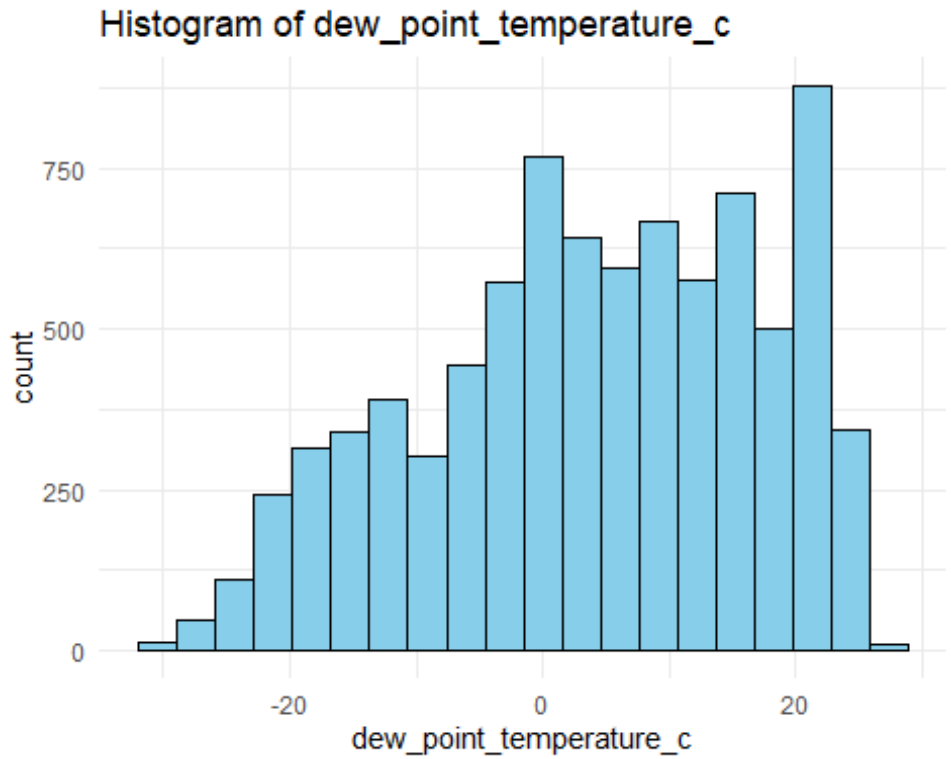

```
##
## $wind_speed_m_s
```



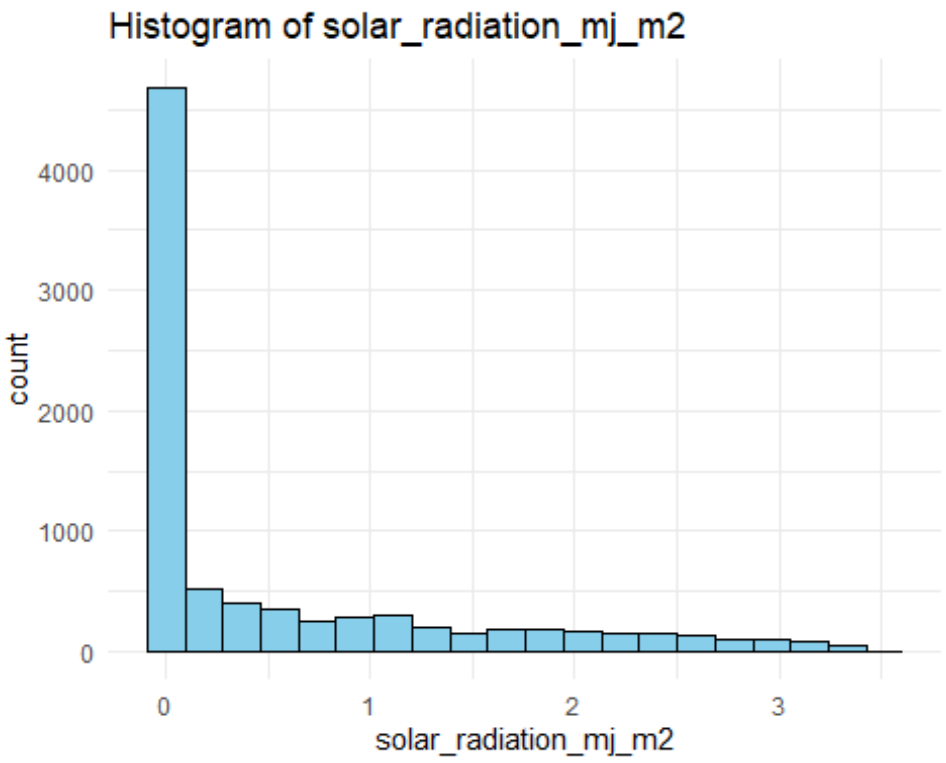

```
##
## $visibility_10m
```



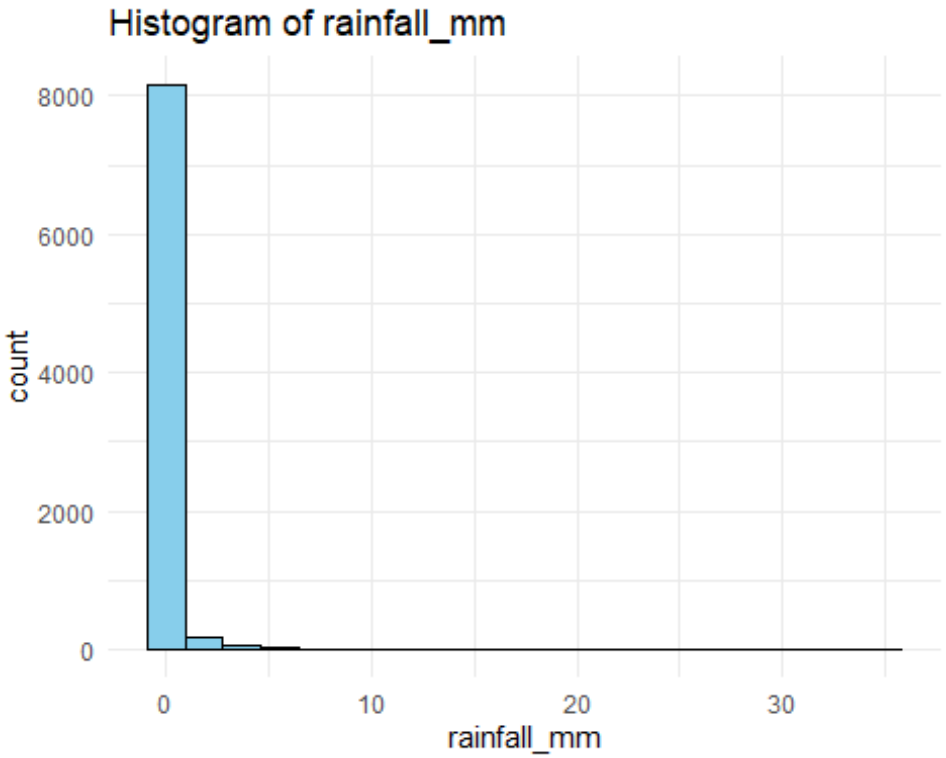
```
##
## $dew_point_temperature_c
```



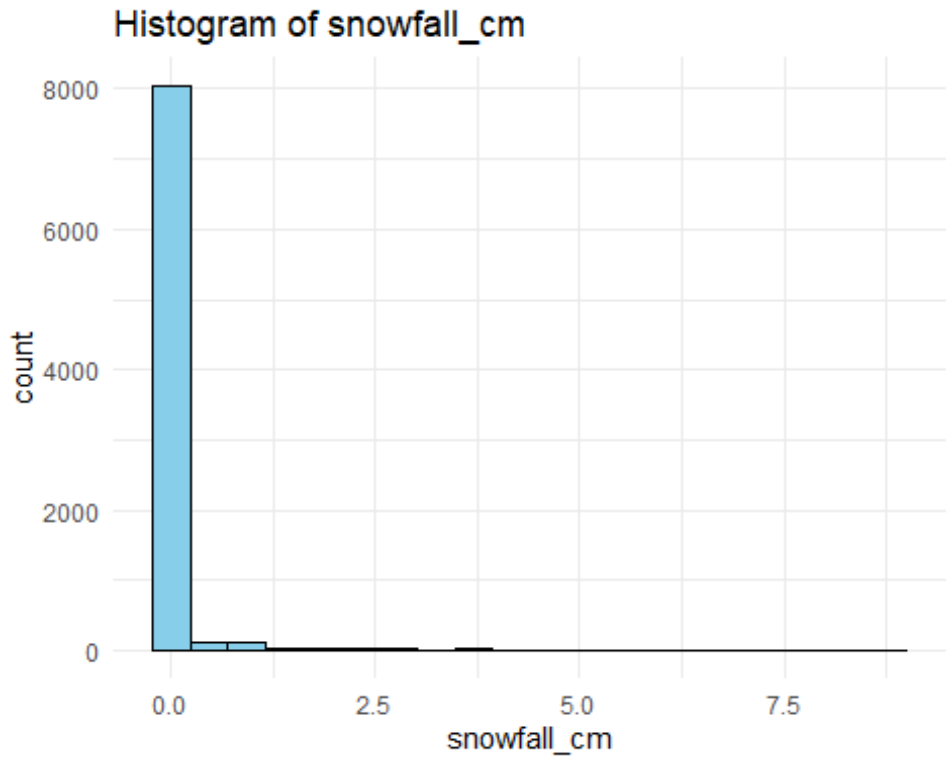

```
##
## $solar_radiation_mj_m2
```


Histogram of solar_radiation_mj_m2
count
4000
3000
2000
1000
0
0
1
2
3
solar_radiation_mj_m2

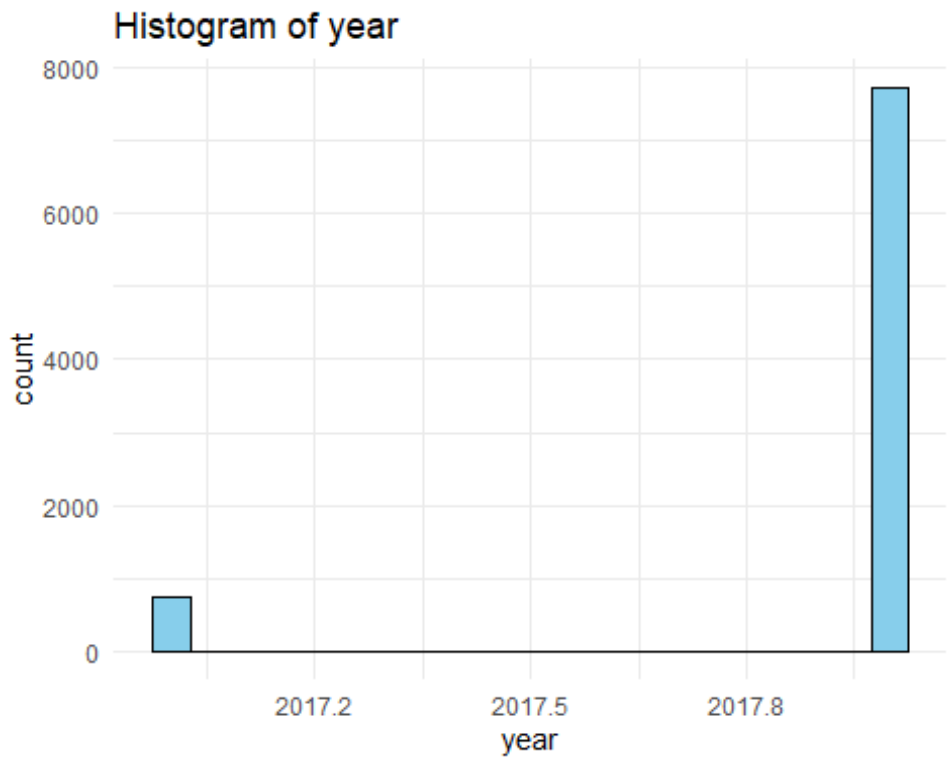
```
##
## $rainfall_mm
```


Histogram of rainfall_mm
count
8000
6000
4000
2000
0
0
10
20
30
rainfall_mm

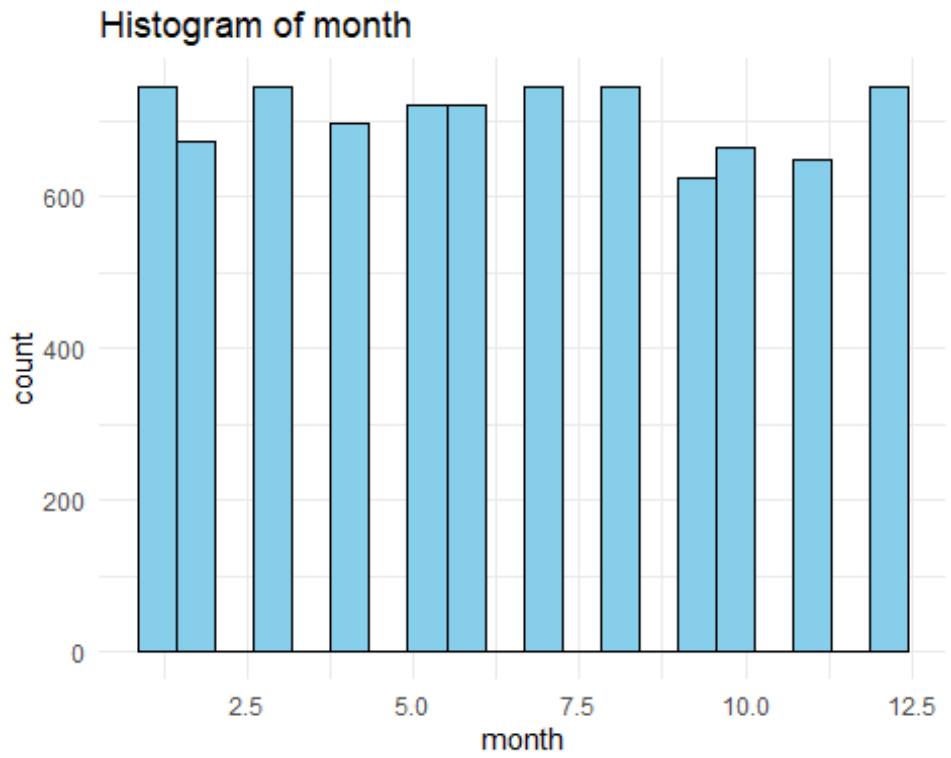
```
##
## $snowfall_cm
```



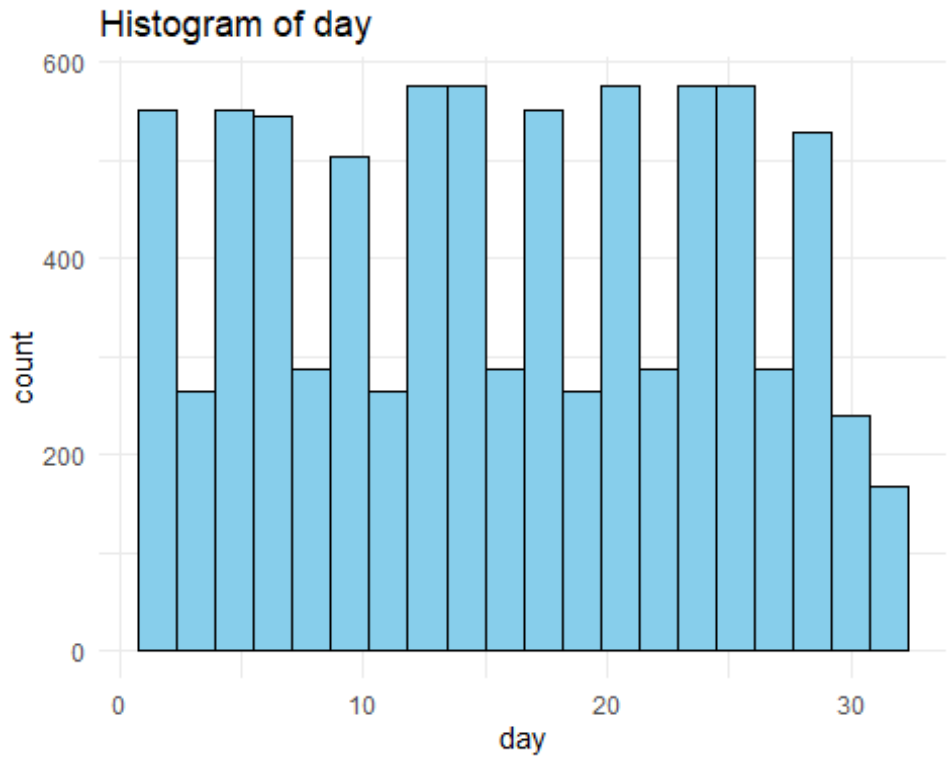

```
##
## $year
```


Histogram of year
8000
6000
4000
2000
0
count
2017.2
2017.5
2017.8
year

```
##
## $month
```


Histogram of month
600
400
200
0
count
2.5
5.0
7.5
10.0
12.5
month

```
##
## $day
```



=> Nhận xét: các thuộc tính không tuân theo phân phối chuẩn.

3.4 Bảng Tổng hợp theo Giờ trong Ngày

```
summary_hour <- data %>%
  group_by(hour) %>%
  summarize(
    total_bike_count = sum(rented_bike_count),
    mean_bike_count = mean(rented_bike_count),
    median_bike_count = median(rented_bike_count),
    max_bike_count = max(rented_bike_count),
    min_bike_count = min(rented_bike_count)
  )
print(summary_hour)
```

```
## # A tibble: 24 × 6
##     hour total_bike_count mean_bike_count median_bike_count max_bike_count
##    <dbl>            <dbl>           <dbl>             <dbl>          <dbl>
##  1     0           197633            561.              532.           1394
##  2     1           155557            442.              412.           1088
##  3     2           110095            313.              277            1254
##  4     3            74216            211.              184.            644
##  5     4            48396            137.              124.            421
##  6     5            50765            144.              136.            383
##  7     6           104961            298.              245             807
```

```
##  8     7           221192            627.              443            1629
##  9     8           370731           1050.              767            2495
## 10     9           235784            668.              719            1401
## # ℹ 14 more rows
## # ℹ 1 more variable: min_bike_count <dbl>
```

=> **Nhận xét**: Bảng cho biết tổng số, trung bình, trung vị, giá trị lớn nhất và nhỏ nhất của số lượng xe đạp thuê theo từng giờ trong ngày. Giúp xác định giờ cao điểm và giờ thấp điểm.

3.4.1 Bảng Tổng hợp theo Ngày trong Tuần

```
data$day_of_week <- weekdays(as.Date(data$date, format="%d/%m/%Y"))

summary_day_of_week <- data %>%
  group_by(day_of_week) %>%
  summarize(
    total_bike_count = sum(rented_bike_count),
    mean_bike_count = mean(rented_bike_count),
    median_bike_count = median(rented_bike_count),
    max_bike_count = max(rented_bike_count),
    min_bike_count = min(rented_bike_count)
  )

print(summary_day_of_week)
```

```
## # A tibble: 7 × 6
##   day_of_week total_bike_count mean_bike_count median_bike_count
max_bike_count
##   <chr>                 <dbl>           <dbl>             <dbl>
<dbl>
## 1 Friday               950334            776.              600
3365
## 2 Monday               911743            731.              533
3380
## 3 Saturday             885492            728.              560
2497
## 4 Sunday               780194            637.              424
2514
## 5 Thursday             861999            718.              538
3418
## 6 Tuesday              858596            745.              572.
3556
## 7 Wednesday            923956            770.              570.
3384
## # ℹ 1 more variable: min_bike_count <dbl>
```

=> Kết quả sẽ cho thấy sự khác biệt về nhu cầu thuê xe đạp giữa các ngày trong tuần.

3.4.2 Bảng Tổng hợp theo Mùa

```
summary_season <- data %>%
  group_by(seasons) %>%
```

```
  summarize(
    total_bike_count = sum(rented_bike_count),
    mean_bike_count = mean(rented_bike_count),
    median_bike_count = median(rented_bike_count),
    max_bike_count = max(rented_bike_count),
    min_bike_count = min(rented_bike_count)
  )

print(summary_season)
```

```
## # A tibble: 4 × 6
##   seasons total_bike_count mean_bike_count median_bike_count
max_bike_count
##   <fct>              <dbl>           <dbl>             <dbl>
<dbl>
## 1 Autumn           1790002            924.              856
3298
## 2 Spring           1611909            746.              599
3251
## 3 Summer           2283234           1034.              906.
3556
## 4 Winter            487169            226.              203
937
## # ℹ 1 more variable: min_bike_count <dbl>
```

=> Kết quả sẽ cho thấy sự khác biệt về nhu cầu thuê xe đạp giữa các mùa. Ví dụ, mùa hè có thể có nhu cầu cao hơn mùa đông.

3.4.3 Bảng Tổng hợp theo Ngày Lễ

```
summary_holiday <- data %>%
  group_by(holiday) %>%
  summarize(
    total_bike_count = sum(rented_bike_count),
    mean_bike_count = mean(rented_bike_count),
    median_bike_count = median(rented_bike_count),
    max_bike_count = max(rented_bike_count),
    min_bike_count = min(rented_bike_count)
  )

print(summary_holiday)
```

```
## # A tibble: 2 × 6
##   holiday    total_bike_count mean_bike_count median_bike_count
max_bike_count
##   <fct>                 <dbl>           <dbl>             <dbl>
<dbl>
## 1 Holiday              215895            529.               259
2400
## 2 No Holiday          5956419            739.               561
```

```
3556
## # ℹ 1 more variable: min_bike_count <dbl>
```

Kết quả sẽ cho thấy sự khác biệt về nhu cầu thuê xe đạp giữa các ngày lễ và ngày thường.

3.4.4 Bảng Tổng hợp theo Nhiệt độ

```
data$temp_range <- cut(data$temperature_c, breaks =
seq(floor(min(data$temperature_c)), ceiling(max(data$temperature_c)), by =
5))

summary_temp <- data %>%
  group_by(temp_range) %>%
  summarize(
    total_bike_count = sum(rented_bike_count),
    mean_bike_count = mean(rented_bike_count),
    median_bike_count = median(rented_bike_count),
    max_bike_count = max(rented_bike_count),
    min_bike_count = min(rented_bike_count)
  )
data <- data |> dplyr::select(-temp_range)
print(summary_temp)
```

```
## # A tibble: 12 × 6
##    temp_range total_bike_count mean_bike_count median_bike_count
max_bike_count
##    <fct>                 <dbl>           <dbl>             <dbl>
<dbl>
##  1 (-18,-13]             10113            119.               94
469
##  2 (-13,-8]              42724            161.              136.
812
##  3 (-8,-3]              123377            195.              168.
1516
##  4 (-3,2]               227180            255.              216
1883
##  5 (2,7]                440565            408.              341
2000
##  6 (7,12]               627171            614.              555
2254
##  7 (12,17]              775206            829.              814
2857
##  8 (17,22]             1245008            967.              931
3130
##  9 (22,27]             1421600           1212.             1117
3556
## 10 (27,32]              937282           1188.             1078
3418
## 11 (32,37]              305711           1076.              821
2965
## 12 <NA>                  16377            780.              665
```

```
1929
## # ℹ 1 more variable: min_bike_count <dbl>
```

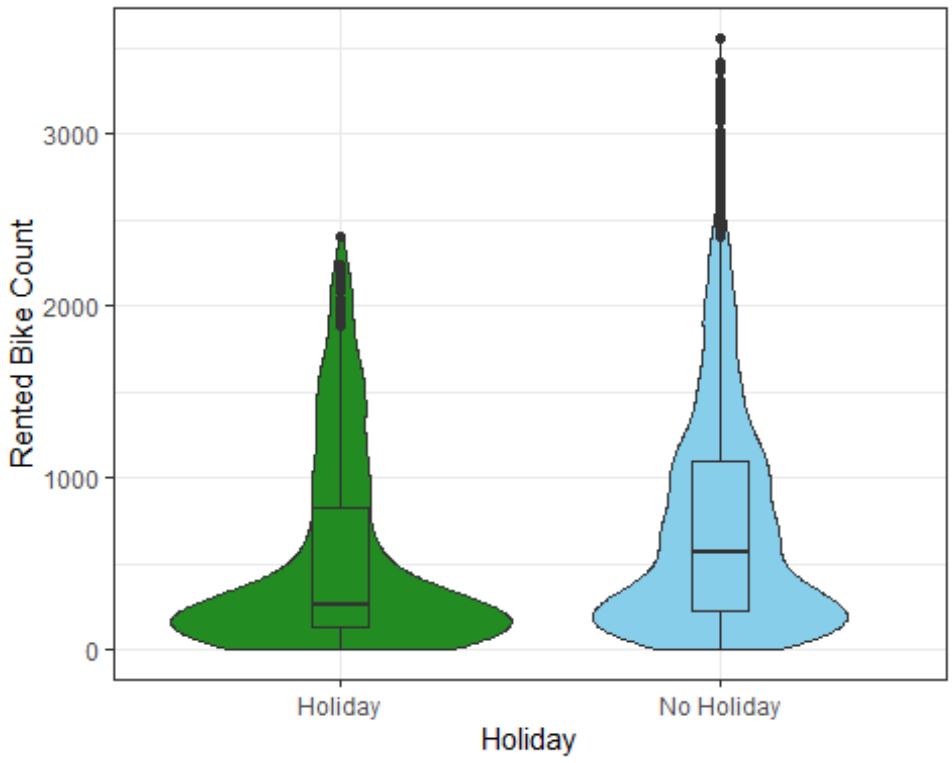
Kết quả sẽ cho thấy mối quan hệ giữa nhiệt độ và số lượng xe đạp thuê. Điều này có thể giúp dự đoán nhu cầu thuê xe đạp dựa trên dự báo thời tiết.

=> Những bảng tổng hợp và biểu đồ này sẽ cung cấp cái nhìn tổng quan và chi tiết về các yếu tố ảnh hưởng đến nhu cầu thuê xe đạp, giúp đưa ra các quyết định quản lý và phát triển dịch vụ hiệu quả hơn.

5. A/B Testing

5.1 Phân loại dữ liệu thành hai nhóm: Holiday và No Holiday

```r
ggplot(data, aes(x = holiday, y = rented_bike_count, fill = holiday)) +
  geom_violin() +
  geom_boxplot(width = 0.15) +
  scale_fill_manual(breaks = c("Holiday", "No Holiday"),
                    values = c("forestgreen", "skyblue")) +
  labs(x = "Holiday", y = "Rented Bike Count") +
  theme_bw() +
  theme(legend.position = "none")
```



Nhóm A: Dữ liệu vào các ngày không phải là ngày nghỉ lễ.

Nhóm B: Dữ liệu vào các ngày nghỉ lễ.

Thông qua bảng tổng hợp và biểu đồ violin, một giả định có thể là "Số lượng sử dụng xe đạp vào những ngày No Holiday là nhiều hơn những ngày Holiday".

Do đó, ta cần kiểm chứng giả thuyết và đối thuyết sau:

- Giả thuyết: μA = μB
- Đối thuyết: μA < μB

```
summary_data <- data %>%
  group_by(holiday) %>%
  summarise(
    n = n(),
    mean_rented_bike_count = mean(rented_bike_count, na.rm = TRUE),
    sd_rented_bike_count = sd(rented_bike_count, na.rm = TRUE)
  )
print(summary_data)
```

```
## # A tibble: 2 × 4
##   holiday        n mean_rented_bike_count sd_rented_bike_count
##   <fct>      <int>                  <dbl>                <dbl>
## 1 Holiday      408                   529.                 574.
## 2 No Holiday  8057                   739.                 644.
```

```
perm_fun <- function(x, nA, nB, R) {
n <- nA + nB
mean_diff <- numeric(R)
for (i in 1:R){
idx_a <- sample(x = 1:n, size = nA)
idx_b <- setdiff(x = 1:n, y = idx_a)
mean_diff[i] <- mean(x[idx_a]) - mean(x[idx_b])
}
return(mean_diff)
}
set.seed(42)
diff_mean_perm <- perm_fun(data$rented_bike_count, nA = 8328 , nB = 432  , R
= 10000)
mean_a <- mean(data$rented_bike_count[data$holiday == 'Holiday'])
mean_b <- mean(data$rented_bike_count [data$holiday == 'No Holiday'])
mean(diff_mean_perm < (mean_a - mean_b))
```

```
## [1] NA
```

Với mức ý nghĩa α = 0, kết quả cho thấy Giả thuyết là không thể bị bác bỏ. Do đó, số lượng sử dụng xe đối với những ngày No Holiday nhiều hơn so với nhứng ngày Holiday là không có ý nghĩa thống kê, hay chỉ là kết quả của sự ngẫu nhiên.

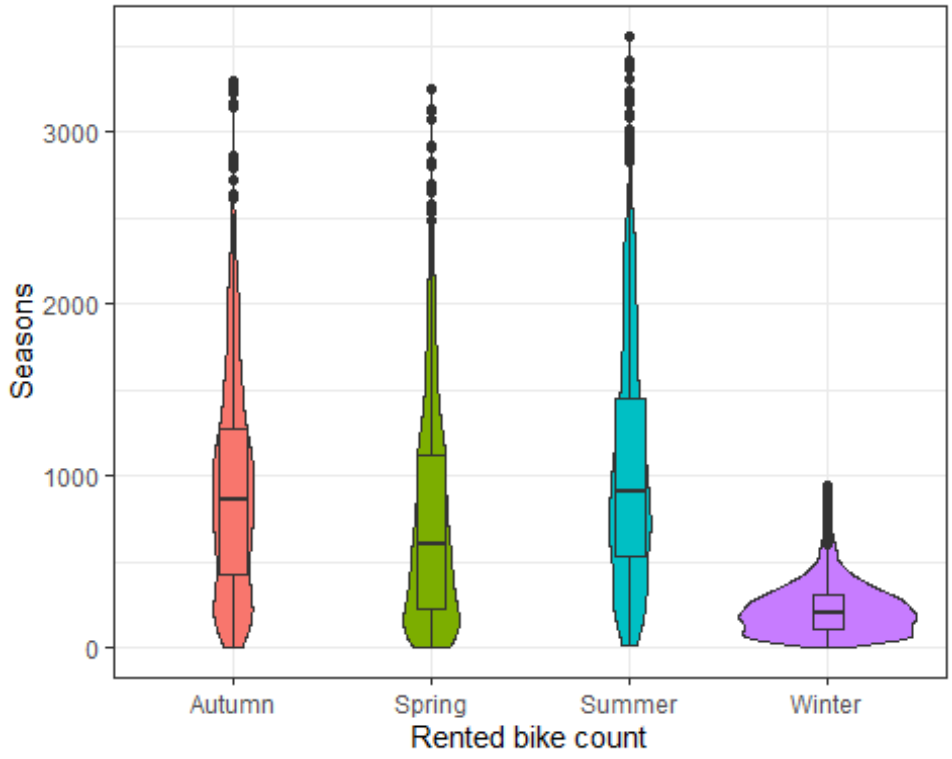
5.2 Season

```
# Thực hiện nhóm dữ liệu theo 'seasons' và tính toán các giá trị
summary_seasons <- data %>%
  group_by(seasons) %>%
```

```
  summarise(
    n = n(),
    mean_rented_bike_count = mean(rented_bike_count, na.rm = TRUE),
    sd_rented_bike_count = sd(rented_bike_count, na.rm = TRUE)
  )
print(summary_seasons)
```

```
## # A tibble: 4 × 4
##   seasons     n mean_rented_bike_count sd_rented_bike_count
##   <fct>   <int>                  <dbl>                <dbl>
## 1 Autumn   1937                   924.                 618.
## 2 Spring   2160                   746.                 619.
## 3 Summer   2208                  1034.                 690.
## 4 Winter   2160                   226.                 150.
```

```
ggplot(data, aes(x = seasons, y = rented_bike_count, fill = seasons)) +
  geom_violin() +
  geom_boxplot(width = 0.15) +
  labs(x = "Rented bike count", y = "Seasons") +
  theme_bw() +
  theme(legend.position = "none")
```



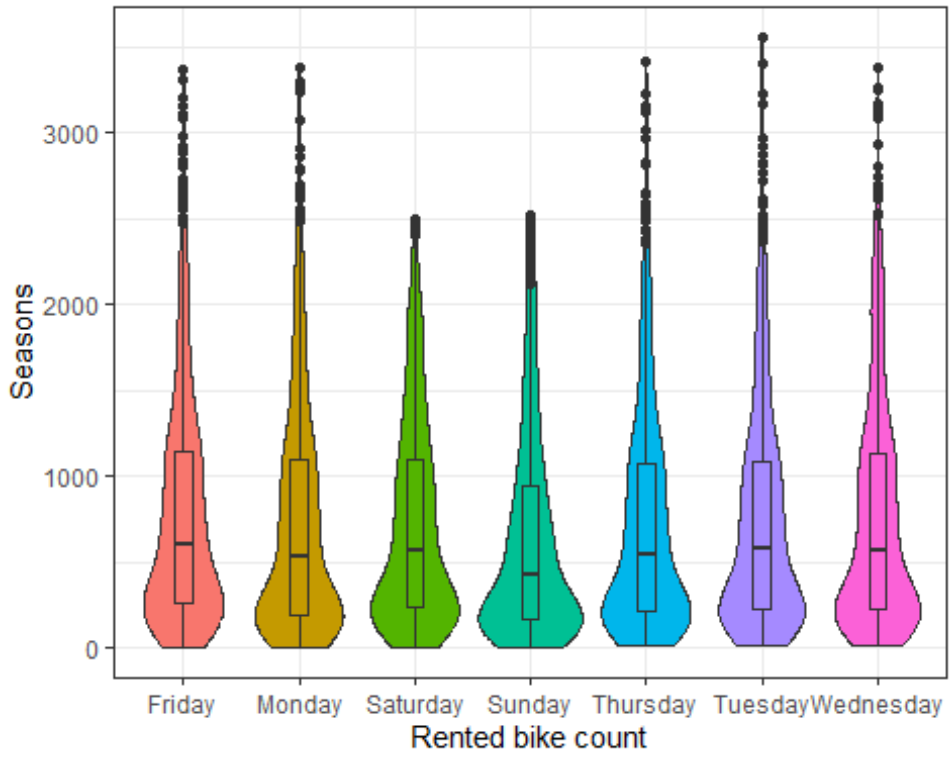

```
library(lmPerm)

set.seed(56)
out_aov_1 <- aovp(formula = rented_bike_count ~ seasons,
                  data = data |> mutate(across(seasons, as.factor)))
```

```
## [1] "Settings:  unique SS "

summary(out_aov_1)

## Component 1 :
##               Df   R Sum Sq R Mean Sq Iter  Pr(Prob)
## seasons        3  827375618 275791873 5000 < 2.2e-16 ***
## Residuals   8461 2664997916    314974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

=> Giá trị p-value rất nhỏ (< 2.2e-16) cho thấy sự khác biệt về số lượng xe đạp thuê giữa các mùa là có ý nghĩa thống kê. Điều này có nghĩa là các mùa khác nhau ảnh hưởng đến số lượng xe đạp thuê một cách đáng kể.

5.3 Ngày trong tuần.

```
# Thực hiện nhóm dữ liệu theo 'day_of_week' và tính toán các giá trị
summary_seasons <- data %>%
  group_by(day_of_week) %>%
  summarise(
    n = n(),
    mean_rented_bike_count = mean(rented_bike_count, na.rm = TRUE),
    sd_rented_bike_count = sd(rented_bike_count, na.rm = TRUE)
  )
print(summary_seasons)

## # A tibble: 7 × 4
##   day_of_week     n mean_rented_bike_count sd_rented_bike_count
##   <chr>       <int>                  <dbl>                <dbl>
## 1 Friday       1224                   776.                 654.
## 2 Monday       1248                   731.                 665.
## 3 Saturday     1217                   728.                 601.
## 4 Sunday       1224                   637.                 605.
## 5 Thursday     1200                   718.                 639.
## 6 Tuesday      1152                   745.                 647.
## 7 Wednesday    1200                   770.                 674.

ggplot(data, aes(x = day_of_week, y = rented_bike_count, fill = day_of_week))
+
  geom_violin() +
  geom_boxplot(width = 0.15) +
  labs(x = "Rented bike count", y = "Seasons") +
  theme_bw() +
  theme(legend.position = "none")
```




```
set.seed(69)

# Thực hiện phân tích ANOVA với dữ liệu đã được xử lý
out_aov_2<- aovp(formula = rented_bike_count ~ day_of_week, data = data, perm
= "Prob")

## [1] "Settings:  unique SS "

summary(out_aov_2)

## Component 1 :
##                Df   R Sum Sq R Mean Sq Iter  Pr(Prob)
## day_of_week1    6   15480791   2580132 5000 < 2.2e-16 ***
## Residuals    8458 3476892744    411077
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

=> Phân tích cho thấy rằng có sự khác biệt đáng kể (p < 2.2e-16) về số lượng xe đạp được thuê giữa các ngày trong tuần (các nhóm của biến day_of_week).

6. Model

6.1 Model dự đoán số xe trong ngày.

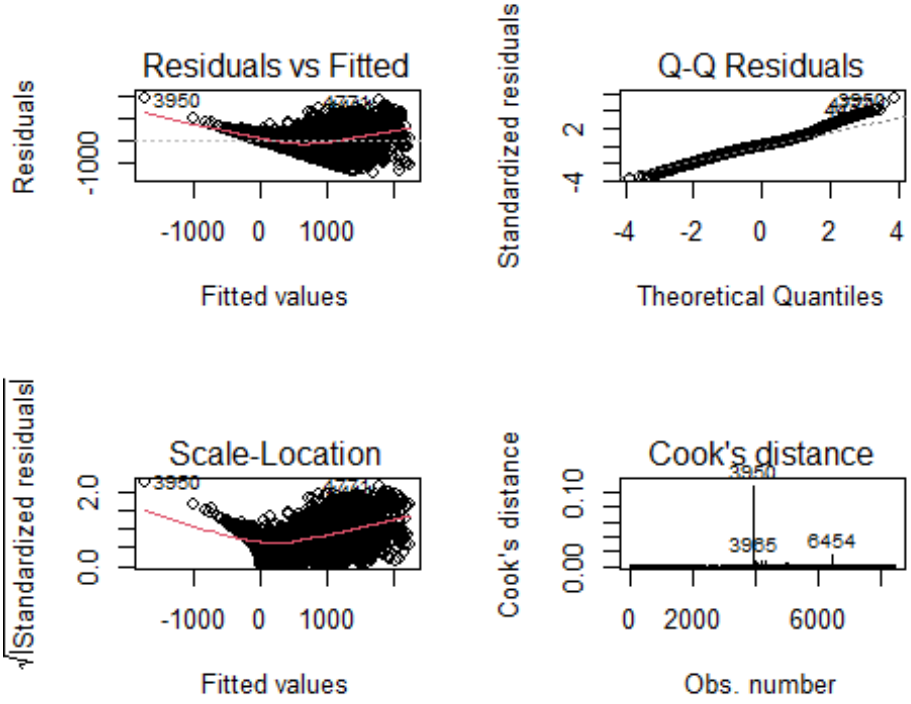
6.1.1 Feature engineering.

```
data_model1 <- data
# Số hóa các giá trị categorical bằng mutate và case_when
data_model1 <- data_model1 |>
  mutate(
    seasons = case_when(
      seasons == "Autumn" ~ 1,
      seasons == "Spring" ~ 2,
      seasons == "Summer" ~ 3,
      seasons == "Winter" ~ 4,
      TRUE ~ NA_real_
    ),
    holiday = case_when(
      holiday == "Holiday" ~ 1,
      holiday == "No Holiday" ~ 2,
      TRUE ~ NA_real_
    ),
    day_of_week = case_when(
      day_of_week == "Monday" ~ 1,
      day_of_week == "Tuesday" ~ 2,
      day_of_week == "Wednesday" ~ 3,
      day_of_week == "Thursday" ~ 4,
      day_of_week == "Friday" ~ 5,
      day_of_week == "Saturday" ~ 6,
      day_of_week == "Sunday" ~ 7,
      TRUE ~ NA_real_
    )
  )
data_model1 <- dummy_cols(data_model1, select_columns = c("hour", "seasons"),
                          remove_first_dummy = TRUE,
                          remove_selected_columns = TRUE)
data_model1 <- data_model1 |> janitor::clean_names()
data_model1 <- data_model1 |> dplyr::select(-c(date, day, day_of_week,
month))
```

6.1.2 Xây dựng mô hình cơ bản.

```
model_1 <- lm(rented_bike_count ~ ., data = data_model1)
summary(model_1)

##
## Call:
## lm(formula = rented_bike_count ~ ., data = data_model1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1380.95  -218.39    -8.47   198.59  1864.72
```

```
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.912e+05  3.474e+04   5.503 3.85e-08 ***
## temperature_c             9.120e+00  3.266e+00   2.793 0.005240 **
## humidity_percent         -1.056e+01  9.010e-01 -11.722  < 2e-16 ***
## wind_speed_m_s            7.445e-02  4.653e+00   0.016 0.987236
## visibility_10m            6.352e-03  8.765e-03   0.725 0.468620
## dew_point_temperature_c   1.528e+01  3.383e+00   4.518 6.33e-06 ***
## solar_radiation_mj_m2     8.140e+01  1.007e+01   8.084 7.11e-16 ***
## rainfall_mm              -6.085e+01  3.795e+00 -16.034  < 2e-16 ***
## snowfall_cm               2.634e+01  9.732e+00   2.707 0.006808 **
## holiday                   1.307e+02  1.925e+01   6.789 1.20e-11 ***
## year                     -9.425e+01  1.722e+01  -5.474 4.53e-08 ***
## hour_1                   -1.053e+02  2.818e+01  -3.738 0.000187 ***
## hour_2                   -2.169e+02  2.820e+01  -7.690 1.63e-14 ***
## hour_3                   -3.038e+02  2.822e+01 -10.768  < 2e-16 ***
## hour_4                   -3.666e+02  2.823e+01 -12.989  < 2e-16 ***
## hour_5                   -3.542e+02  2.827e+01 -12.530  < 2e-16 ***
## hour_6                   -1.862e+02  2.829e+01  -6.581 4.94e-11 ***
## hour_7                    1.262e+02  2.829e+01   4.461 8.25e-06 ***
## hour_8                    4.977e+02  2.847e+01  17.484  < 2e-16 ***
## hour_9                    2.262e+01  2.911e+01   0.777 0.437038
## hour_10                  -2.158e+02  3.024e+01  -7.136 1.04e-12 ***
## hour_11                  -2.269e+02  3.144e+01  -7.216 5.80e-13 ***
## hour_12                  -1.867e+02  3.234e+01  -5.773 8.05e-09 ***
## hour_13                  -1.846e+02  3.265e+01  -5.653 1.62e-08 ***
## hour_14                  -1.785e+02  3.220e+01  -5.542 3.07e-08 ***
## hour_15                  -9.058e+01  3.143e+01  -2.882 0.003962 **
## hour_16                   4.739e+01  3.037e+01   1.560 0.118721
## hour_17                   3.295e+02  2.946e+01  11.188  < 2e-16 ***
## hour_18                   7.881e+02  2.884e+01  27.324  < 2e-16 ***
## hour_19                   5.273e+02  2.853e+01  18.479  < 2e-16 ***
## hour_20                   4.539e+02  2.840e+01  15.984  < 2e-16 ***
## hour_21                   4.466e+02  2.826e+01  15.805  < 2e-16 ***
## hour_22                   3.488e+02  2.818e+01  12.379  < 2e-16 ***
## hour_23                   1.108e+02  2.815e+01   3.935 8.40e-05 ***
## seasons_2                -1.635e+02  1.230e+01 -13.294  < 2e-16 ***
## seasons_3                -1.634e+02  1.521e+01 -10.740  < 2e-16 ***
## seasons_4                -3.965e+02  1.846e+01 -21.479  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 373.6 on 8428 degrees of freedom
## Multiple R-squared:  0.6631, Adjusted R-squared:  0.6617
## F-statistic: 460.8 on 36 and 8428 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(model_1 , which=1:4)
```



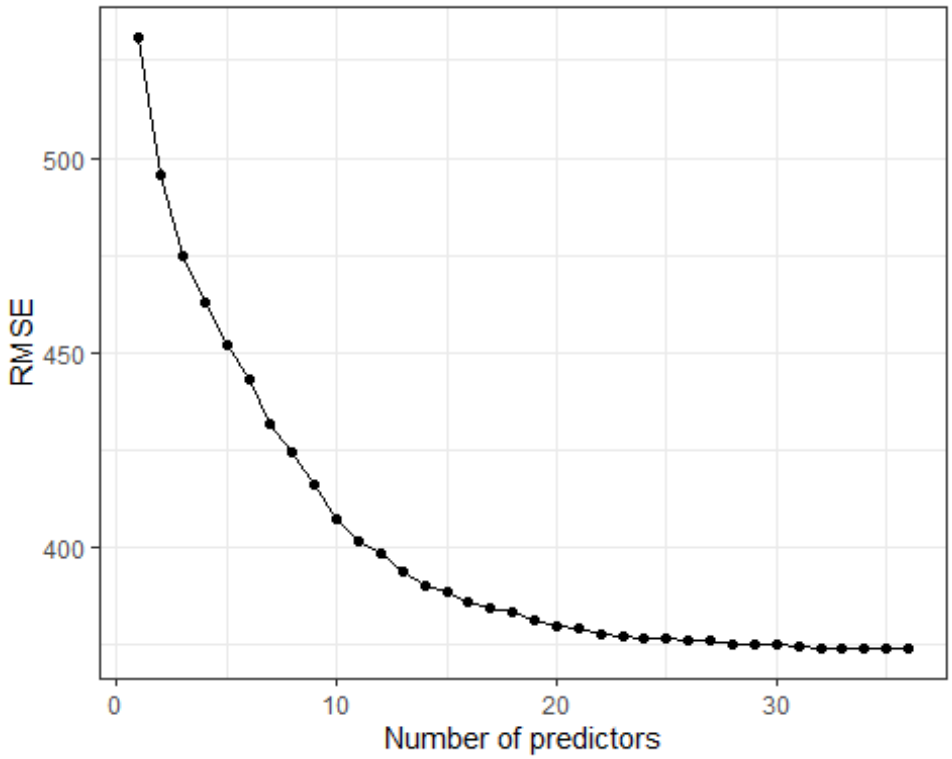
Nhận thấy mô hình có nhiều biến không có ý nghĩa thông kê và

6.1.3 Select feature.

Sử dụng phương pháp hồi quy từng bước kết hợp với CV.

```
# Tạo hàm predict cho regsubsets
predict.regsubsets <- function(object, newdata, id_model){
  form <- as.formula(object$call[[2]])
  x_mat <- model.matrix(form, newdata)
  coef_est <- coef(object, id = id_model)
  x_vars <- names(coef_est)
  x_mat <- x_mat[, x_vars, drop = FALSE]  # Đảm bảo rằng x_mat có các biến
cần thiết
  res <- x_mat %*% coef_est
  return(as.numeric(res))
}

n_data_model1 <- nrow(data_model1)
k <- 5
set.seed(21)
folds <- sample(rep(1:k, length = n_data_model1))

# Đảm bảo rằng số lượng tối đa các biến dự đoán không vượt quá số biến thực
tế
nvmax_actual <- min(37, ncol(data_model1) - 1)
cv_error_model1_rj <- matrix(0, nrow = k, ncol = nvmax_actual)
```

```r
for(r in 1:k){
  data_model1_train_r <- data_model1[folds != r, ]
  data_model1_test_r <- data_model1[folds == r, ]

  out_subset_model1_folds <- regsubsets(x = rented_bike_count ~ ., data =
data_model1_train_r,
                                        method = "exhaustive", nvmax =
nvmax_actual)

  for(j in 1:nvmax_actual){
    pred_rj <- predict.regsubsets(out_subset_model1_folds,
                                  newdata = data_model1_test_r, id_model = j)
    cv_error_model1_rj[r, j] <-
sqrt(mean((data_model1_test_r$rented_bike_count - pred_rj)^2))
  }
}

cv_error_model1 <- colMeans(cv_error_model1_rj)
ggplot(data = data.frame(x = c(1:36), y = cv_error_model1),
  mapping = aes(x = x, y = y)) +
  geom_point() +
  geom_line() +
  labs(x = "Number of predictors", y = "RMSE") +
  theme_bw()
```



Nhận thấy khi ta dùng càng nhiều thuộc tính thì RMSE càng giảm, điều này không có ý nghĩa gì cả cho việc chọn lựa thuộc tính, nên ta sẽ dùng hồi quy từng phần với tiêu chí BIC.

```
data_model1 <- data_model1
regsubset <- regsubsets(x = rented_bike_count ~ ., data = data_model1, nvmax
= 36,
                                    method = "exhaustive")

reg_summary <- summary(regsubset)

# Tiêu chí Mallow's Cp
best_model_cp <- which.min(reg_summary$cp)

# Tiêu chí BIC
best_model_bic <- which.min(reg_summary$bic)

# Tiêu chí Adjusted R²
best_model_adjr2 <- which.max(reg_summary$adjr2)

# Hiển thị số lượng biến tốt nhất dựa trên các tiêu chí
cat("Số lượng biến tốt nhất dựa trên tiêu chí Mallow's Cp:", best_model_cp,
"\n")

## Số lượng biến tốt nhất dựa trên tiêu chí Mallow's Cp: 32

cat("Số lượng biến tốt nhất dựa trên tiêu chí BIC:", best_model_bic, "\n")

## Số lượng biến tốt nhất dựa trên tiêu chí BIC: 31

# Lấy các biến của mô hình tốt nhất dựa trên tiêu chí Mallow's Cp
best_features_cp <- names(coef(regsubset, best_model_cp))
cat("Các biến tốt nhất dựa trên tiêu chí Mallow's Cp:", best_features_cp,
"\n")

## Các biến tốt nhất dựa trên tiêu chí Mallow's Cp: (Intercept) temperature_c
humidity_percent dew_point_temperature_c solar_radiation_mj_m2 rainfall_mm
snowfall_cm holiday year hour_1 hour_2 hour_3 hour_4 hour_5 hour_6 hour_7
hour_8 hour_10 hour_11 hour_12 hour_13 hour_14 hour_15 hour_17 hour_18
hour_19 hour_20 hour_21 hour_22 hour_23 seasons_2 seasons_3 seasons_4

# Lấy các biến của mô hình tốt nhất dựa trên tiêu chí BIC
best_features_bic <- names(coef(regsubset, best_model_bic))
cat("Các biến tốt nhất dựa trên tiêu chí BIC:", best_features_bic, "\n")

## Các biến tốt nhất dựa trên tiêu chí BIC: (Intercept) temperature_c
humidity_percent dew_point_temperature_c solar_radiation_mj_m2 rainfall_mm
holiday year hour_1 hour_2 hour_3 hour_4 hour_5 hour_6 hour_7 hour_8 hour_10
hour_11 hour_12 hour_13 hour_14 hour_15 hour_17 hour_18 hour_19 hour_20
hour_21 hour_22 hour_23 seasons_2 seasons_3 seasons_4
```

```r
# Hợp nhất các biến của cả 3 tiêu chí
all_best_features <- unique(c(best_features_cp, best_features_bic))

# Hiển thị tổng hợp các biến từ 3 tiêu chí
cat("Tổng hợp các biến từ 2 tiêu chí:", all_best_features, "\n")
```

```
## Tổng hợp các biến từ 2 tiêu chí: (Intercept) temperature_c
humidity_percent dew_point_temperature_c solar_radiation_mj_m2 rainfall_mm
snowfall_cm holiday year hour_1 hour_2 hour_3 hour_4 hour_5 hour_6 hour_7
hour_8 hour_10 hour_11 hour_12 hour_13 hour_14 hour_15 hour_17 hour_18
hour_19 hour_20 hour_21 hour_22 hour_23 seasons_2 seasons_3 seasons_4
```

6.1.4 Building model

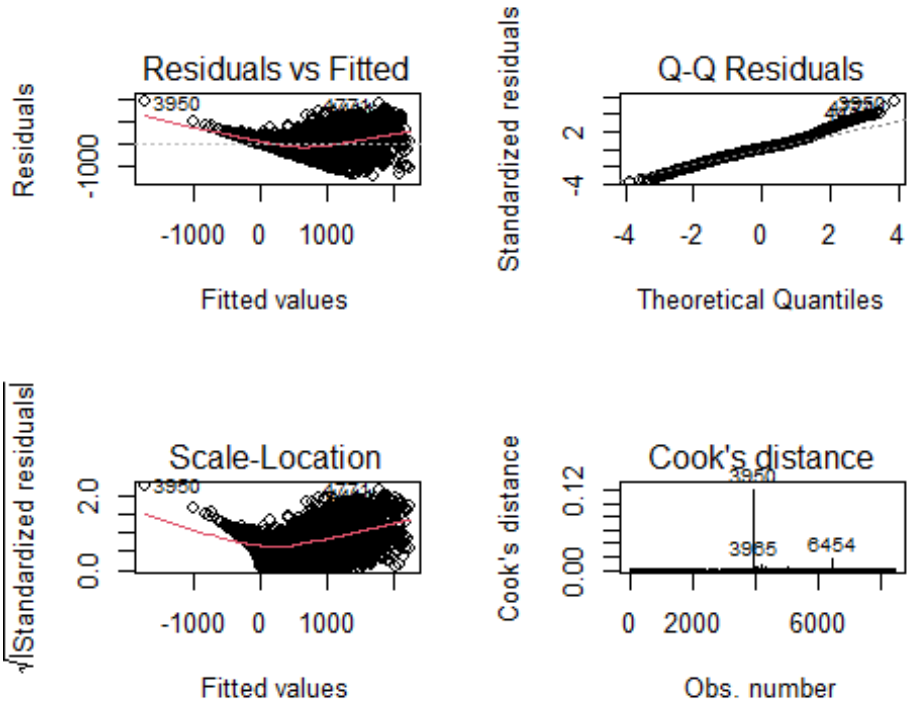
Xây dựng mô hình mới với các thuộc tình vừa tìm được.

```r
# Tạo công thức hồi quy tuyến tính
formula <- as.formula(paste("rented_bike_count ~", paste(all_best_features[-
1], collapse = " + ")))

# Xây dựng mô hình hồi quy tuyến tính
model_1 <- lm(formula, data = data_model1)
summary(model_1)
```

```
##
## Call:
## lm(formula = formula, data = data_model1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1380.34  -218.58    -8.73   197.55  1866.31
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.927e+05  3.468e+04   5.555 2.85e-08 ***
## temperature_c             9.355e+00  3.231e+00   2.895 0.003801 **
## humidity_percent         -1.067e+01  8.702e-01 -12.267  < 2e-16 ***
## dew_point_temperature_c   1.512e+01  3.361e+00   4.499 6.93e-06 ***
## solar_radiation_mj_m2     8.491e+01  9.523e+00   8.916  < 2e-16 ***
## rainfall_mm              -6.060e+01  3.786e+00 -16.006  < 2e-16 ***
## snowfall_cm               2.654e+01  9.727e+00   2.728 0.006384 **
## holiday                   1.312e+02  1.924e+01   6.818 9.87e-12 ***
## year                     -9.496e+01  1.719e+01  -5.526 3.38e-08 ***
## hour_1                   -1.249e+02  2.361e+01  -5.290 1.25e-07 ***
## hour_2                   -2.365e+02  2.361e+01 -10.016  < 2e-16 ***
## hour_3                   -3.234e+02  2.361e+01 -13.697  < 2e-16 ***
## hour_4                   -3.862e+02  2.363e+01 -16.346  < 2e-16 ***
## hour_5                   -3.736e+02  2.366e+01 -15.789  < 2e-16 ***
## hour_6                   -2.057e+02  2.366e+01  -8.694  < 2e-16 ***
## hour_7                    1.062e+02  2.350e+01   4.517 6.37e-06 ***
## hour_8                    4.763e+02  2.322e+01  20.518  < 2e-16 ***
```

```
## hour_10                -2.417e+02  2.358e+01 -10.248  < 2e-16 ***
## hour_11                -2.546e+02  2.430e+01 -10.477  < 2e-16 ***
## hour_12                -2.155e+02  2.490e+01  -8.654  < 2e-16 ***
## hour_13                -2.140e+02  2.507e+01  -8.533  < 2e-16 ***
## hour_14                -2.076e+02  2.470e+01  -8.405  < 2e-16 ***
## hour_15                -1.188e+02  2.412e+01  -4.925 8.61e-07 ***
## hour_17                 3.048e+02  2.323e+01  13.124  < 2e-16 ***
## hour_18                 7.654e+02  2.338e+01  32.739  < 2e-16 ***
## hour_19                 5.062e+02  2.362e+01  21.430  < 2e-16 ***
## hour_20                 4.336e+02  2.372e+01  18.277  < 2e-16 ***
## hour_21                 4.266e+02  2.366e+01  18.034  < 2e-16 ***
## hour_22                 3.290e+02  2.361e+01  13.932  < 2e-16 ***
## hour_23                 9.094e+01  2.359e+01   3.855 0.000116 ***
## seasons_2              -1.661e+02  1.186e+01 -14.001  < 2e-16 ***
## seasons_3              -1.647e+02  1.506e+01 -10.937  < 2e-16 ***
## seasons_4              -3.968e+02  1.811e+01 -21.914  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 373.6 on 8432 degrees of freedom
## Multiple R-squared:  0.663,  Adjusted R-squared:  0.6617
## F-statistic: 518.4 on 32 and 8432 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model_1 , which=1:4)
```



Nhận thấy đồ thị thặng dư khá vẻ tốt hơn và các biến trong mô hình điều có ý nghĩa thống kê, nhưng vẫn chưa tốt nên ta sẽ thực hiện thêm việc mở rộng mô hình.

6.1.5 Mở rộng mô hình.

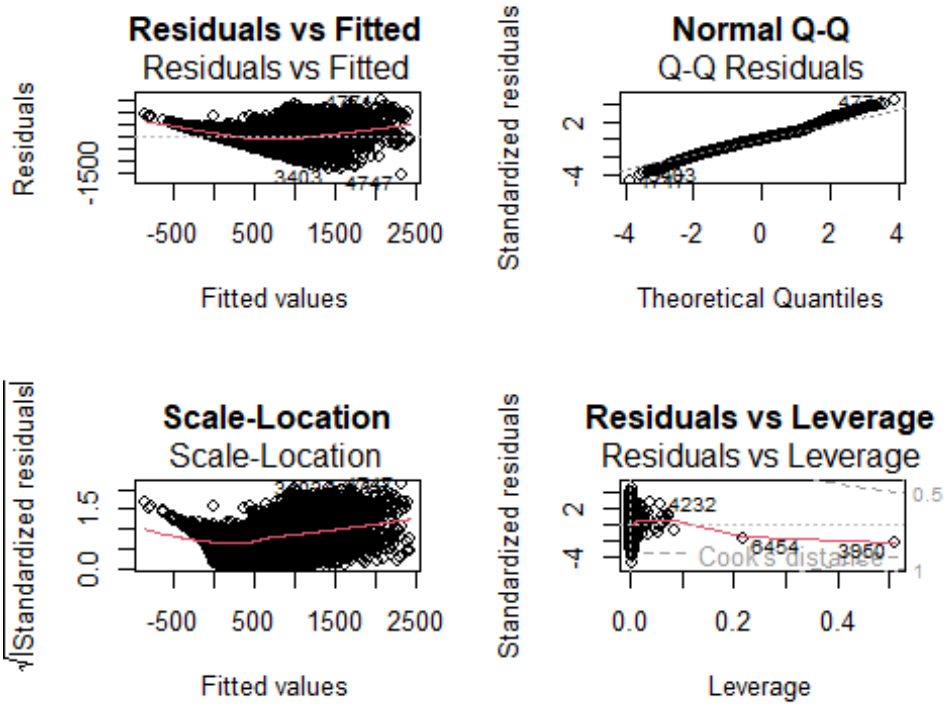
```
knots_temperature_c <- quantile(data_model1$temperature_c, probs = c(0.25,
0.75))

model_1_expand <- lm(formula = rented_bike_count ~
                       bs(temperature_c, knots = knots_temperature_c, degree
= 2) +
                       poly(humidity_percent, 2) +
                       poly(dew_point_temperature_c, 3) +
                       poly(rainfall_mm, 2) + holiday + year + hour_1 +
                       hour_2 + hour_3 + hour_4 + hour_5 +
                       hour_6  + hour_8 + hour_10 +
                       hour_11 + hour_12 + hour_13 + hour_14 +
                       hour_17 + hour_18 + hour_19 +
                       hour_20 + hour_21  +
                       seasons_2 + seasons_3 + seasons_4, data = data_model1)
summary(model_1_expand)

##
## Call:
## lm(formula = rented_bike_count ~ bs(temperature_c, knots =
knots_temperature_c,
##     degree = 2) + poly(humidity_percent, 2) +
poly(dew_point_temperature_c,
##     3) + poly(rainfall_mm, 2) + holiday + year + hour_1 + hour_2 +
##     hour_3 + hour_4 + hour_5 + hour_6 + hour_8 + hour_10 + hour_11 +
##     hour_12 + hour_13 + hour_14 + hour_17 + hour_18 + hour_19 +
##     hour_20 + hour_21 + seasons_2 + seasons_3 + seasons_4, data =
data_model1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1578.07  -199.12    -2.43   186.01  1470.56
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                    148215.01
## bs(temperature_c, knots = knots_temperature_c, degree = 2)1      -230.70
## bs(temperature_c, knots = knots_temperature_c, degree = 2)2       266.21
## bs(temperature_c, knots = knots_temperature_c, degree = 2)3      1827.60
## bs(temperature_c, knots = knots_temperature_c, degree = 2)4       531.61
## poly(humidity_percent, 2)1                                     -4176.64
## poly(humidity_percent, 2)2                                     -5324.87
## poly(dew_point_temperature_c, 3)1                             -24123.50
## poly(dew_point_temperature_c, 3)2                             -12743.35
## poly(dew_point_temperature_c, 3)3                              -2730.35
## poly(rainfall_mm, 2)1                                          -5226.85
```

```
## poly(rainfall_mm, 2)2                                                3322.95
## holiday                                                               102.16
## year                                                                  -73.38
## hour_1                                                               -238.60
## hour_2                                                               -346.06
## hour_3                                                               -423.51
## hour_4                                                               -486.20
## hour_5                                                               -469.26
## hour_6                                                               -302.31
## hour_8                                                                378.25
## hour_10                                                              -252.52
## hour_11                                                              -225.87
## hour_12                                                              -152.41
## hour_13                                                              -146.67
## hour_14                                                              -136.44
## hour_17                                                               288.36
## hour_18                                                               706.45
## hour_19                                                               416.72
## hour_20                                                               322.49
## hour_21                                                               313.49
## seasons_2                                                            -136.50
## seasons_3                                                             -63.87
## seasons_4                                                            -344.32
##                                                                  Std. Error t
value
## (Intercept)                                                        31485.40
4.707
## bs(temperature_c, knots = knots_temperature_c, degree = 2)1          80.06  -
2.882
## bs(temperature_c, knots = knots_temperature_c, degree = 2)2         119.10
2.235
## bs(temperature_c, knots = knots_temperature_c, degree = 2)3         168.08
10.873
## bs(temperature_c, knots = knots_temperature_c, degree = 2)4         193.69
2.745
## poly(humidity_percent, 2)1                                         1702.04  -
2.454
## poly(humidity_percent, 2)2                                          475.02 -
11.210
## poly(dew_point_temperature_c, 3)1                                  4216.35  -
5.721
## poly(dew_point_temperature_c, 3)2                                   660.46 -
19.295
## poly(dew_point_temperature_c, 3)3                                   453.83  -
6.016
## poly(rainfall_mm, 2)1                                               365.05 -
14.318
## poly(rainfall_mm, 2)2                                               351.57
9.452
## holiday                                                              17.43
```

```
5.861
## year                                                          15.60  -
4.703
## hour_1                                                        19.29 -
12.367
## hour_2                                                        19.32 -
17.915
## hour_3                                                        19.34 -
21.893
## hour_4                                                        19.38 -
25.082
## hour_5                                                        19.45 -
24.127
## hour_6                                                        19.45 -
15.540
## hour_8                                                        19.30
19.595
## hour_10                                                       19.16 -
13.178
## hour_11                                                       19.24 -
11.739
## hour_12                                                       19.38  -
7.864
## hour_13                                                       19.54  -
7.507
## hour_14                                                       19.66  -
6.941
## hour_17                                                       19.43
14.840
## hour_18                                                       19.29
36.628
## hour_19                                                       19.18
21.723
## hour_20                                                       19.18
16.815
## hour_21                                                       19.18
16.343
## seasons_2                                                     10.66 -
12.799
## seasons_3                                                     15.74  -
4.058
## seasons_4                                                     18.09 -
19.033
##                                                             Pr(>|t|)
## (Intercept)                                                 2.55e-06 ***
## bs(temperature_c, knots = knots_temperature_c, degree = 2)1  0.00397 **
## bs(temperature_c, knots = knots_temperature_c, degree = 2)2  0.02543 *
## bs(temperature_c, knots = knots_temperature_c, degree = 2)3  < 2e-16 ***
## bs(temperature_c, knots = knots_temperature_c, degree = 2)4  0.00607 **
## poly(humidity_percent, 2)1                                   0.01415 *
```

```
## poly(humidity_percent, 2)2                                    < 2e-16 ***
## poly(dew_point_temperature_c, 3)1                            1.09e-08 ***
## poly(dew_point_temperature_c, 3)2                             < 2e-16 ***
## poly(dew_point_temperature_c, 3)3                            1.86e-09 ***
## poly(rainfall_mm, 2)1                                         < 2e-16 ***
## poly(rainfall_mm, 2)2                                         < 2e-16 ***
## holiday                                                      4.76e-09 ***
## year                                                         2.61e-06 ***
## hour_1                                                        < 2e-16 ***
## hour_2                                                        < 2e-16 ***
## hour_3                                                        < 2e-16 ***
## hour_4                                                        < 2e-16 ***
## hour_5                                                        < 2e-16 ***
## hour_6                                                        < 2e-16 ***
## hour_8                                                        < 2e-16 ***
## hour_10                                                       < 2e-16 ***
## hour_11                                                       < 2e-16 ***
## hour_12                                                      4.17e-15 ***
## hour_13                                                      6.66e-14 ***
## hour_14                                                      4.17e-12 ***
## hour_17                                                       < 2e-16 ***
## hour_18                                                       < 2e-16 ***
## hour_19                                                       < 2e-16 ***
## hour_20                                                       < 2e-16 ***
## hour_21                                                       < 2e-16 ***
## seasons_2                                                     < 2e-16 ***
## seasons_3                                                    5.00e-05 ***
## seasons_4                                                     < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.2 on 8431 degrees of freedom
## Multiple R-squared:  0.7271, Adjusted R-squared:  0.726
## F-statistic: 680.7 on 33 and 8431 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))  # Hiển thị 4 đồ thị cùng một lúc

# Residuals vs Fitted
plot(model_1_expand, which = 1, main = "Residuals vs Fitted")

# Normal Q-Q
plot(model_1_expand, which = 2, main = "Normal Q-Q")

# Scale-Location
plot(model_1_expand, which = 3, main = "Scale-Location")

# Residuals vs Leverage
plot(model_1_expand, which = 5, main = "Residuals vs Leverage")
```




```
par(mfrow = c(1, 1))  # Trở về chế độ hiển thị mặc định
```

Nhận xét:

- Theo đồ thị **Residuals vs Fitted Values:** Điều này có thể chỉ ra rằng mô hình hồi quy tuyến tính không phù hợp.
- Theo đồ thị **Normal Q-Q Plot:** Thặng dư không có phân phối chuẩn, không ứng giả định normality của thặng dư.

Nhận thấy mô hình không được tốt cho lắm, việc tiếp cận mô hình cũng không hay, nên ta thực hiện một hướng tiếp cận khác của dữ liệu để kiểm tra.

6.2 Model dự đoán số xe theo ngày.

6.2.1 Feature engineering.

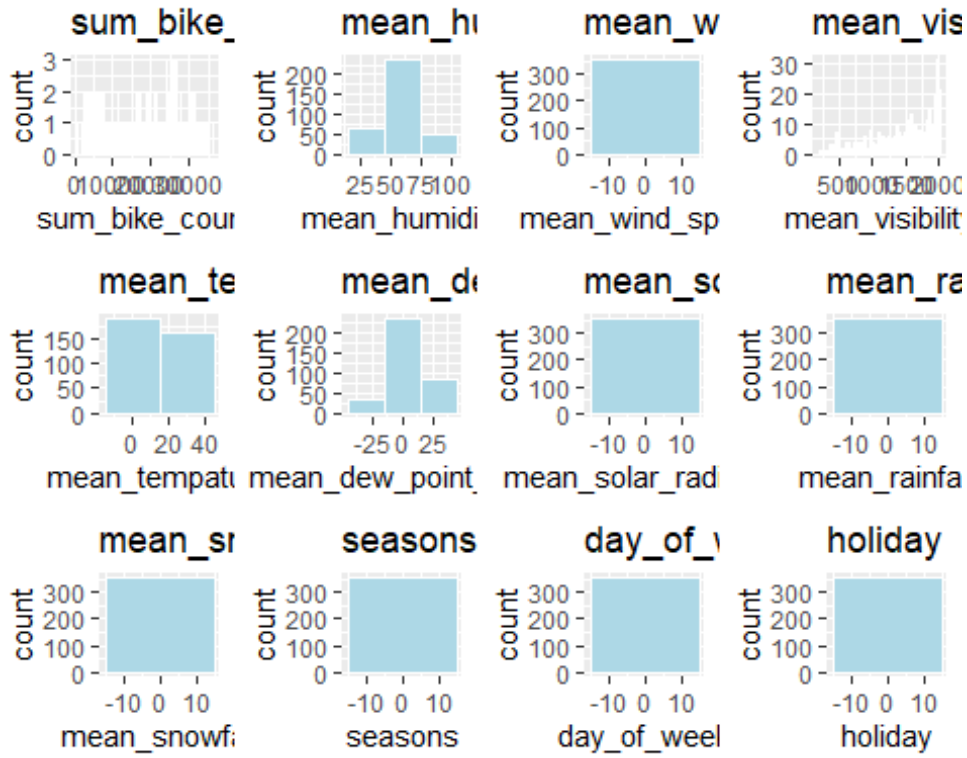
```
data_model2 <- data |> dplyr::select(-c(hour, day, month))
data_model2 <- data_model2 |>
  mutate(
    seasons = case_when(
      seasons == "Autumn" ~ 1,
      seasons == "Spring" ~ 2,
      seasons == "Summer" ~ 3,
      seasons == "Winter" ~ 4,
      TRUE ~ NA_real_
    ),
    holiday = case_when(
```

```
      holiday == "Holiday" ~ 1,
      holiday == "No Holiday" ~ 2,
      TRUE ~ NA_real_
    ),
    day_of_week = case_when(
      day_of_week == "Monday" ~ 1,
      day_of_week == "Tuesday" ~ 2,
      day_of_week == "Wednesday" ~ 3,
      day_of_week == "Thursday" ~ 4,
      day_of_week == "Friday" ~ 5,
      day_of_week == "Saturday" ~ 6,
      day_of_week == "Sunday" ~ 7,
      TRUE ~ NA_real_
    )
  )
data_model2 <- data_model2 |>
  group_by(date) |>
  dplyr::summarise(sum_bike_count = sum(rented_bike_count),
            mean_humidity = mean(humidity_percent),
            mean_wind_speed = mean(wind_speed_m_s),
            mean_visibility = mean(visibility_10m),
            mean_tempature = mean(temperature_c),
            mean_dew_point_temp = mean(dew_point_temperature_c),
            mean_solar_radiation = mean(solar_radiation_mj_m2),
            mean_rainfall = mean(rainfall_mm),
            mean_snowfall = mean(snowfall_cm),
            seasons = mean(seasons),
            day_of_week = mean(day_of_week),
            holiday = mean(holiday)
            )
head(data_model2)
```

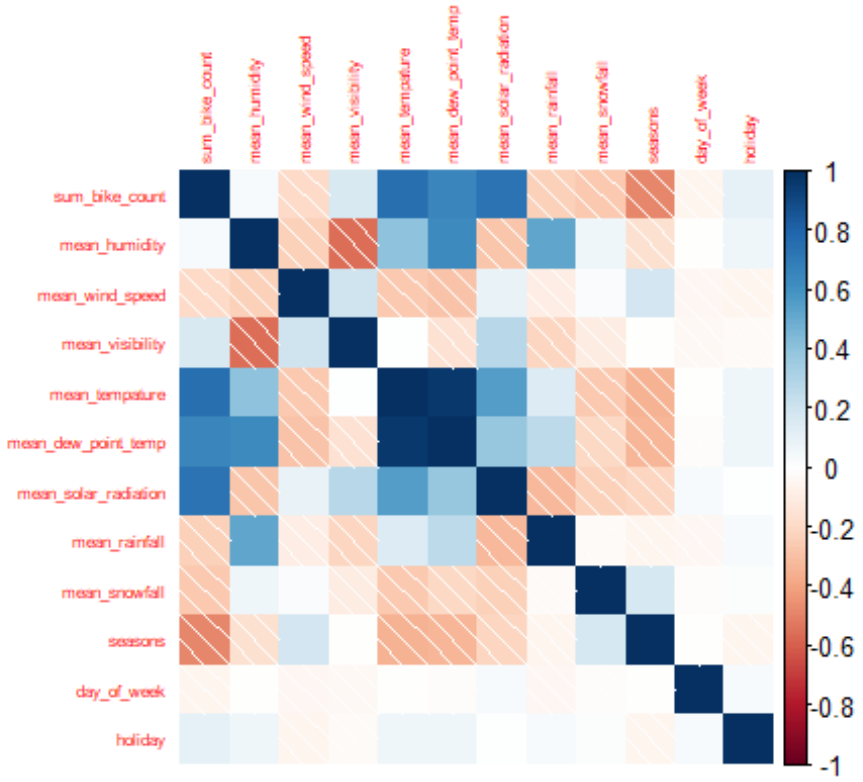
```
## # A tibble: 6 × 13
##   date       sum_bike_count mean_humidity mean_wind_speed mean_visibility
##   <chr>               <dbl>         <dbl>           <dbl>           <dbl>
## 1 01/01/2018           4290          39.3            1.45           1895.
## 2 01/02/2018           5377          44              1.61           1924.
## 3 01/03/2018           5132          64.2            3.55           1084
## 4 01/04/2018          17388          68.9            1.57            832.
## 5 01/05/2018          26820          72.8            1.44            456.
## 6 01/06/2018          31928          50.1            1.95           1598.
## # ℹ 8 more variables: mean_tempature <dbl>, mean_dew_point_temp <dbl>,
## #   mean_solar_radiation <dbl>, mean_rainfall <dbl>, mean_snowfall <dbl>,
## #   seasons <dbl>, day_of_week <dbl>, holiday <dbl>
```

```
library(ggplot2)
library(gridExtra)


numeric_columns <- sapply(data_model2, is.numeric)
```

```
numeric_data <- data_model2[, numeric_columns]

num_cols <- ceiling(sqrt(sum(numeric_columns)))
num_rows <- ceiling(sum(numeric_columns) / num_cols)



# Tạo danh sách các biểu đồ histogram
plots <- lapply(names(numeric_data), function(col) {
  ggplot(numeric_data, aes_string(x = col)) +
    geom_histogram(binwidth = 30, fill = "lightblue", color = "white") +
    ggtitle(col)
})

# Sắp xếp các biểu đồ theo dạng lưới
do.call(grid.arrange, c(plots, ncol = num_cols))
```

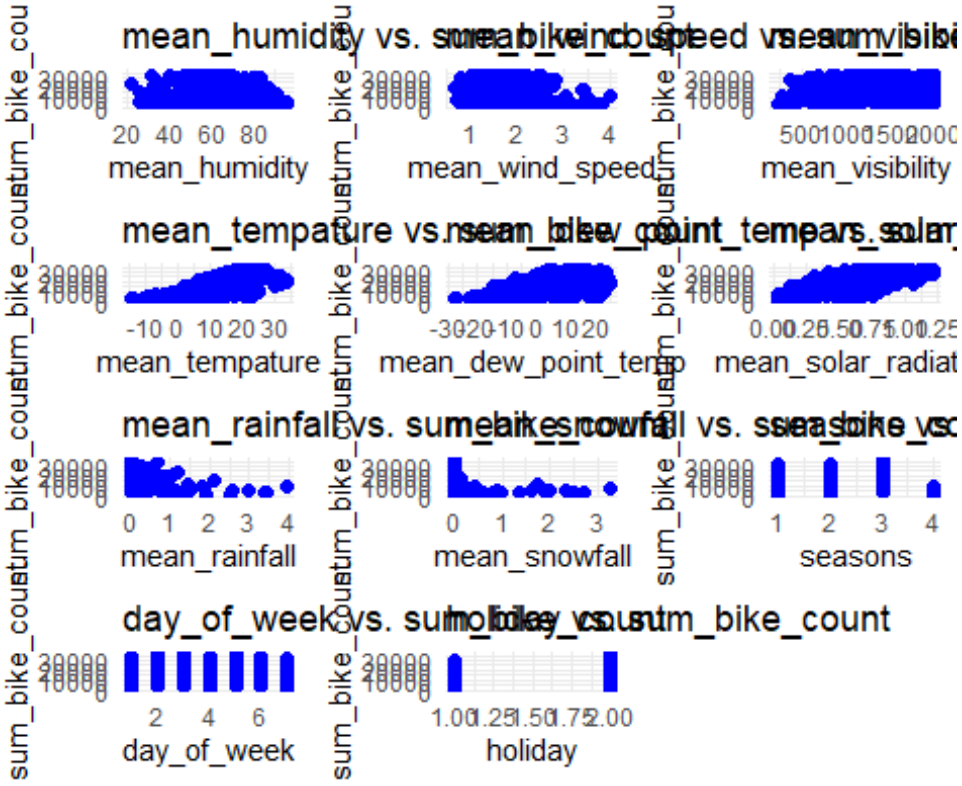



```
correlation_matrix <- cor(numeric_data)

# Plot the correlation matrix
corrplot(correlation_matrix, method = "shade",tl.cex = 0.5)
```

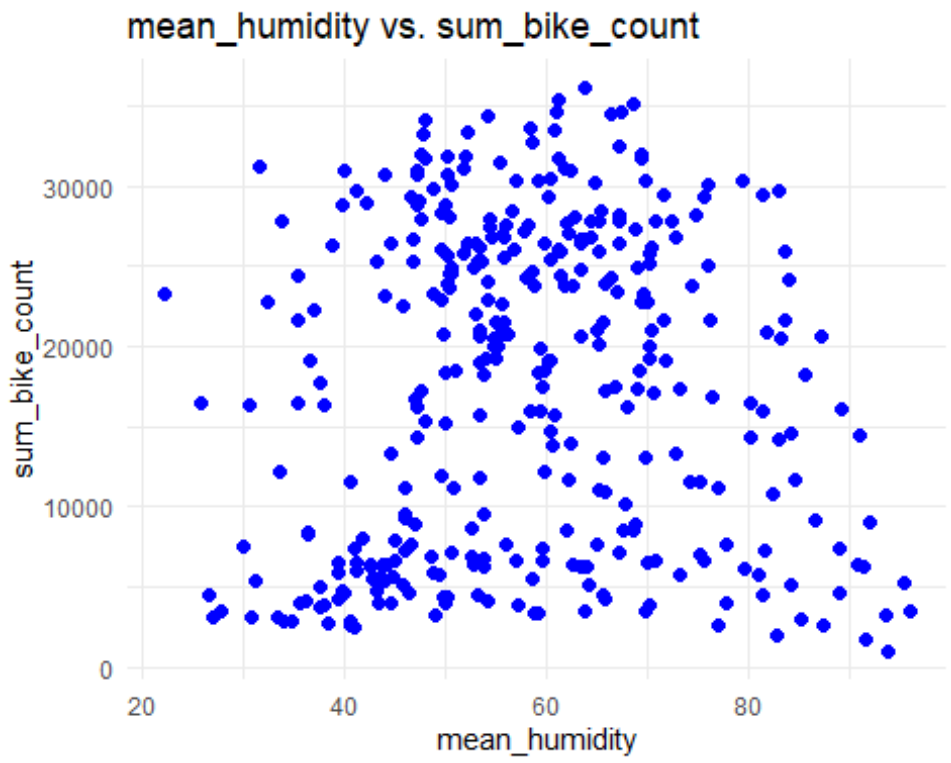



```r
multi.scatter <- function(data, target) {
  # Initialize an empty list to store plots
  plots <- list()

  # Loop through each numeric variable
  for (col in names(data)) {
    # Check if current variable is numeric and not the target variable
    if (is.numeric(data[[col]]) && col != target) {
      # Create scatter plot
      scatter_plot <- ggplot(data, aes_string(x = col, y = target)) +
        geom_point(size = 2, color = "blue") +
        ggtitle(paste(col, "vs.", target)) +
        theme_minimal()

      # Add the plot to the list
      plots[[col]] <- scatter_plot
    }
  }

  # Arrange plots in a grid
  grid.arrange(grobs = plots, ncol = 3)  # Adjust ncol as needed

  # Return the list of plots (optional)
  return(plots)
}
```

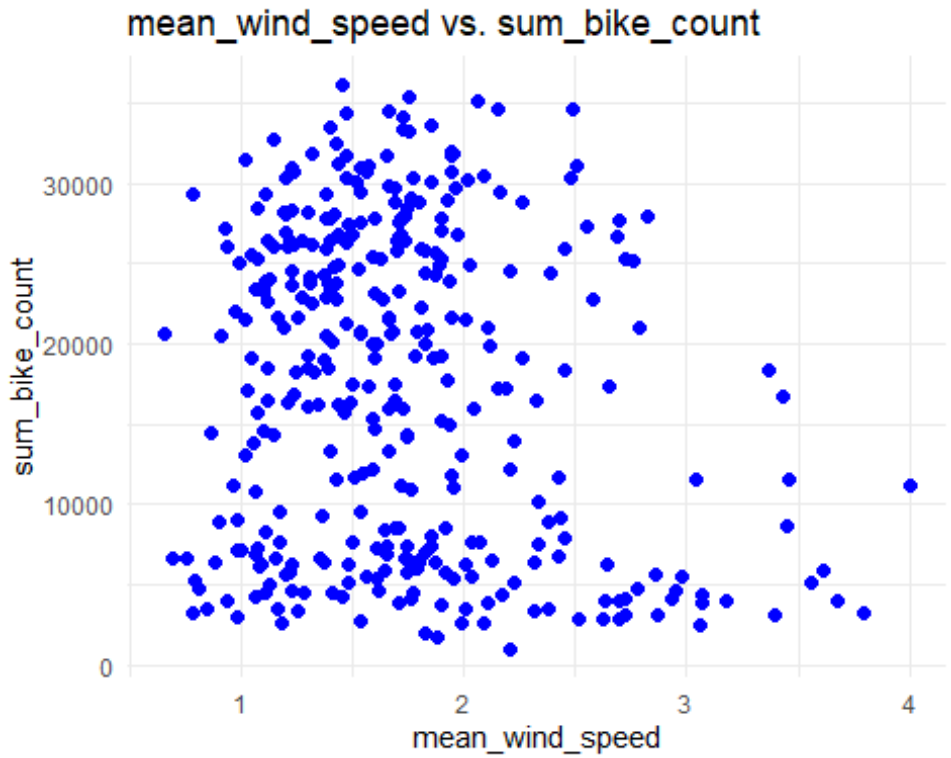
```
# Call the function with numeric columns and target variable
multi.scatter(data_model2, "sum_bike_count")
```



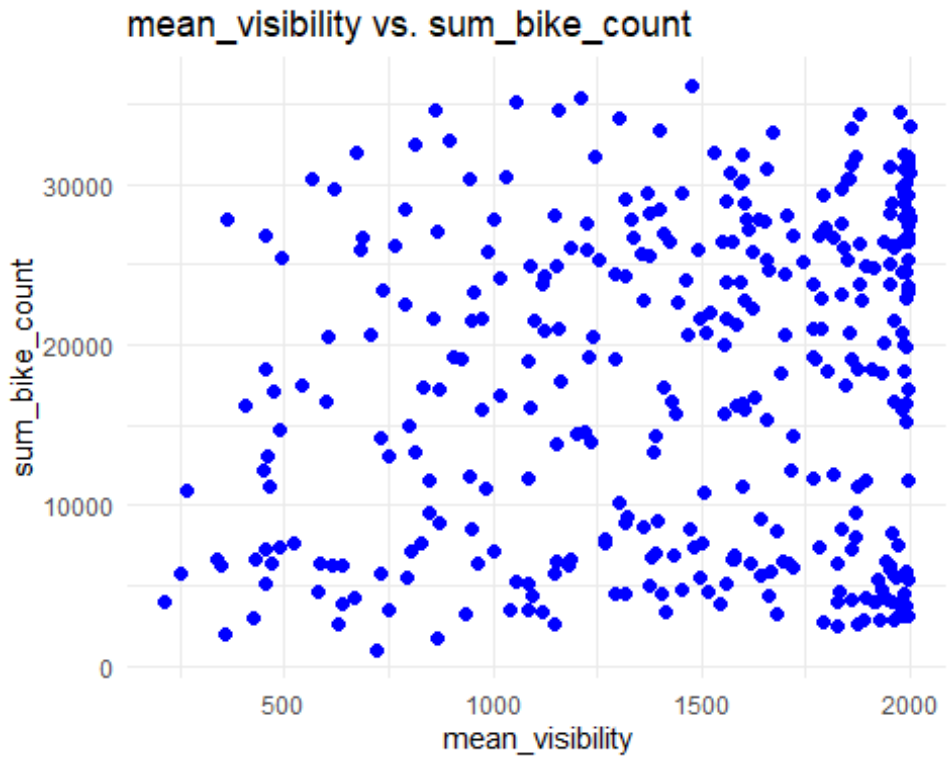

```
## $mean_humidity
```



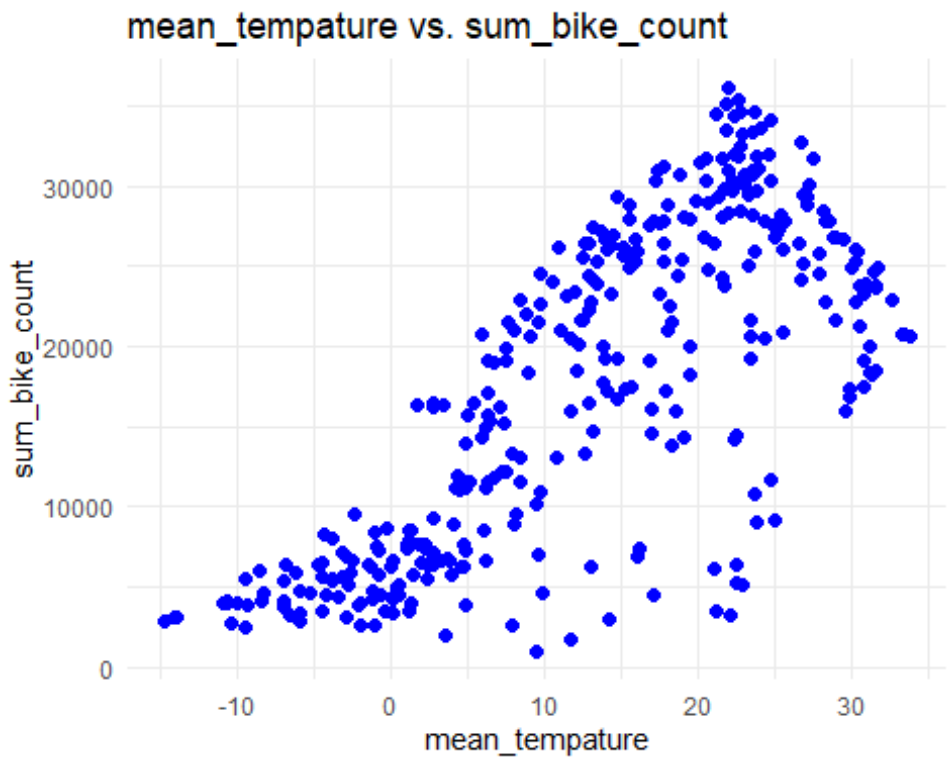

```
##
## $mean_wind_speed
```



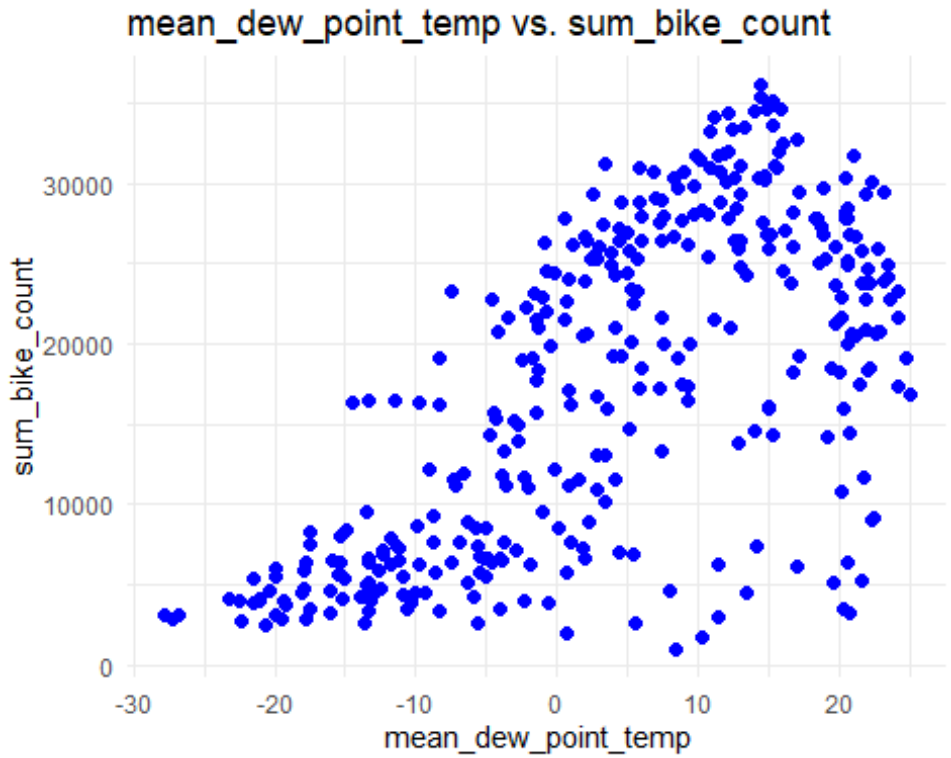
```
##
## $mean_visibility
```



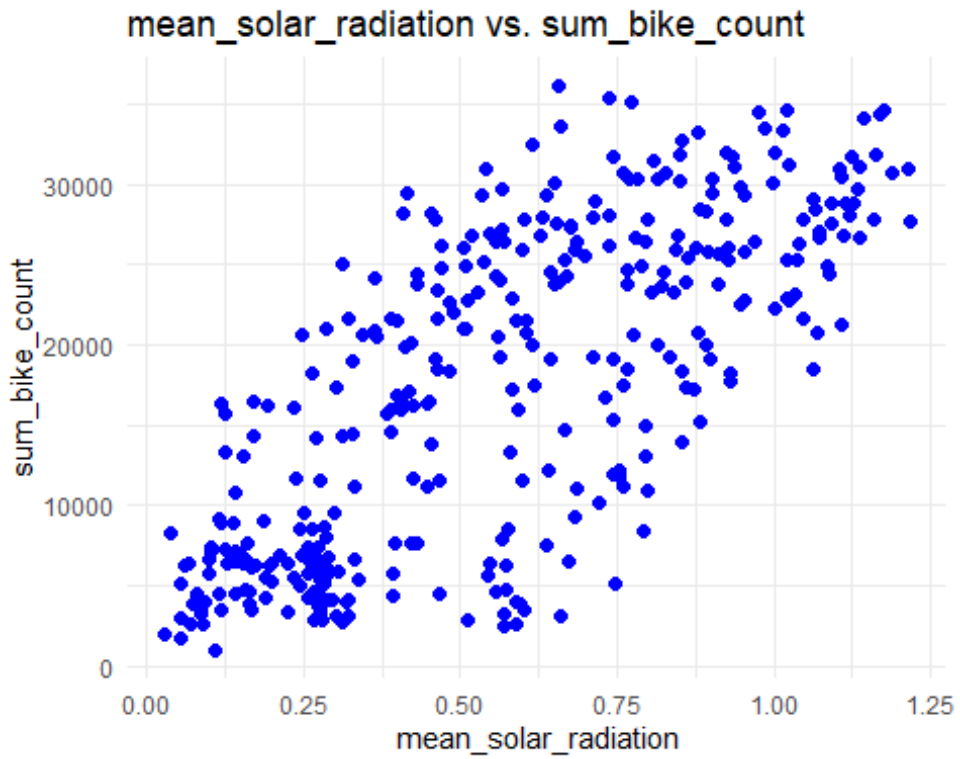

```
##
## $mean_tempature
```



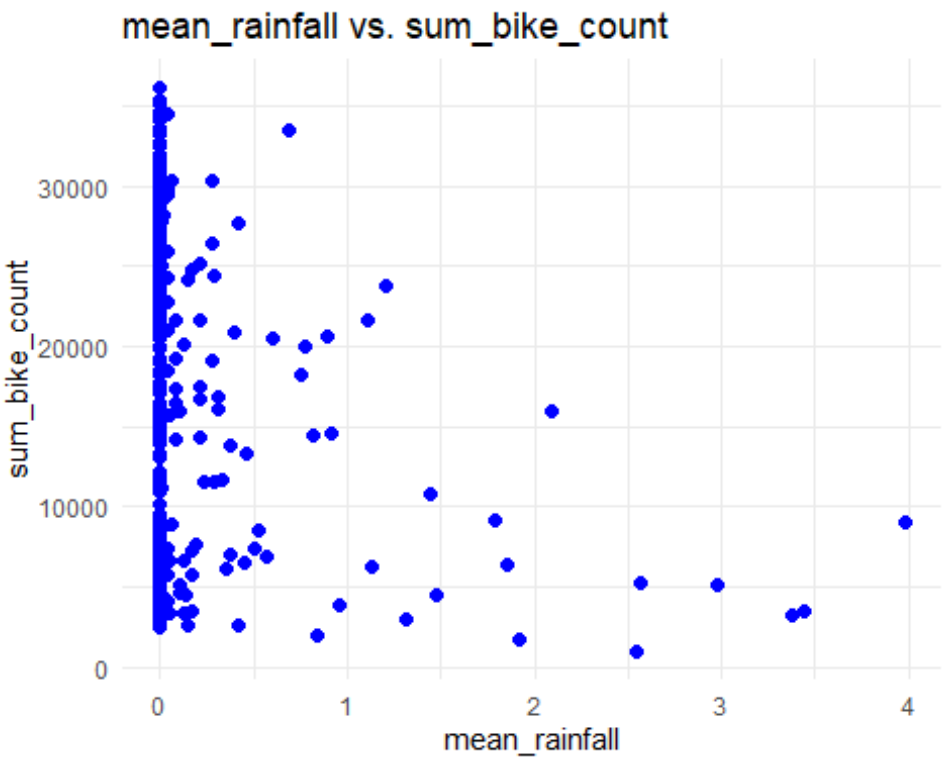

```
##
## $mean_dew_point_temp
```



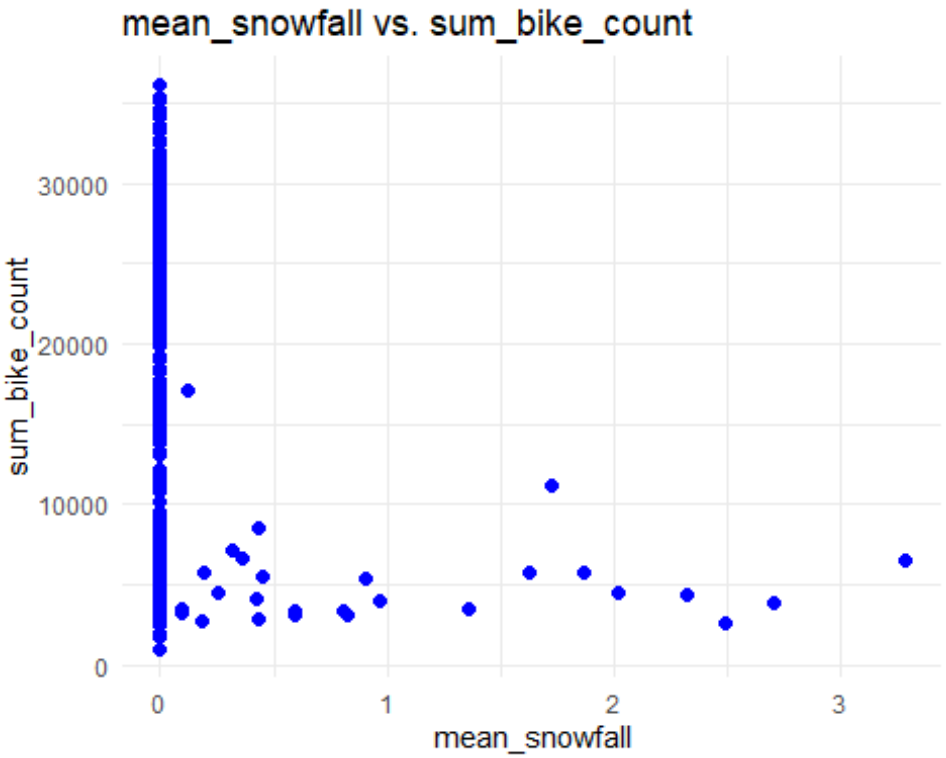
```
##
## $mean_solar_radiation
```



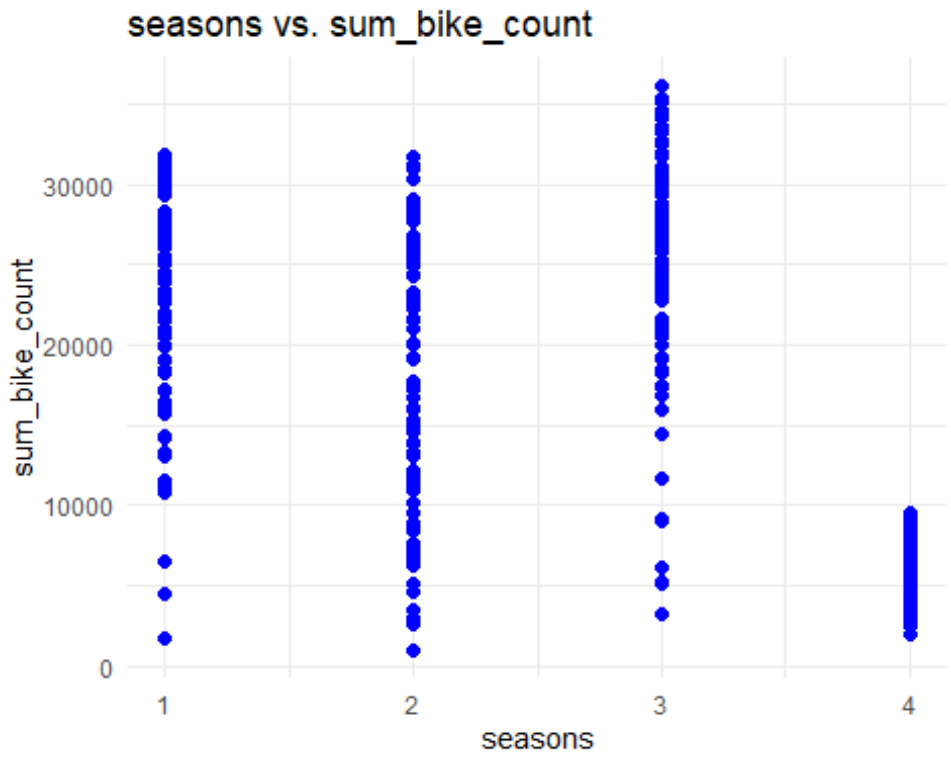

```
##
## $mean_rainfall
```



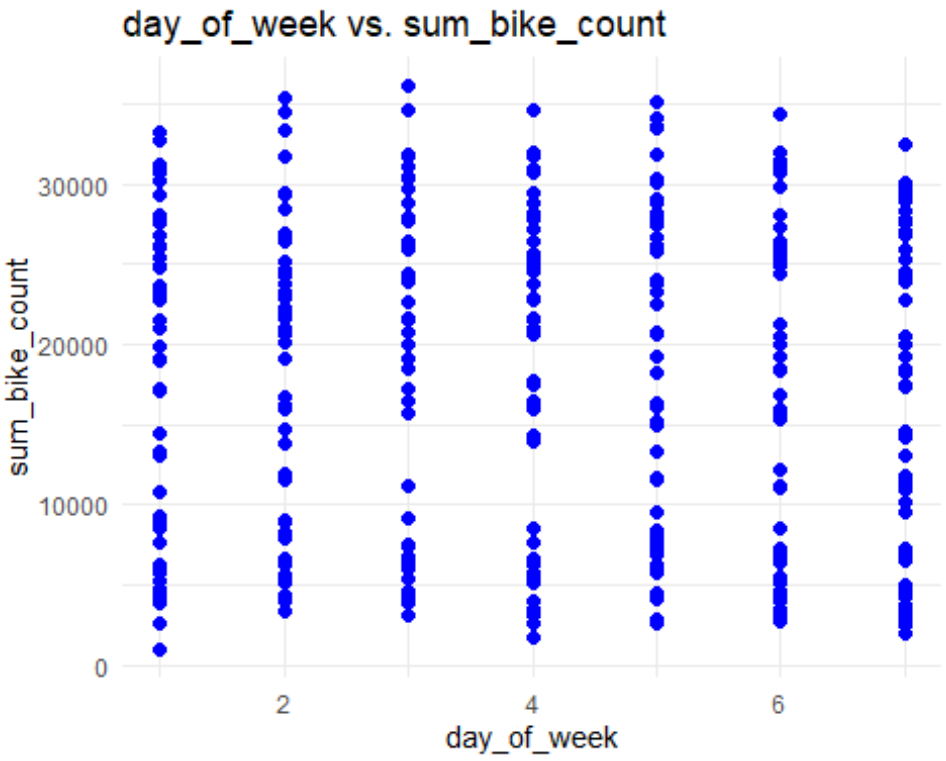

```
##
## $mean_snowfall
```



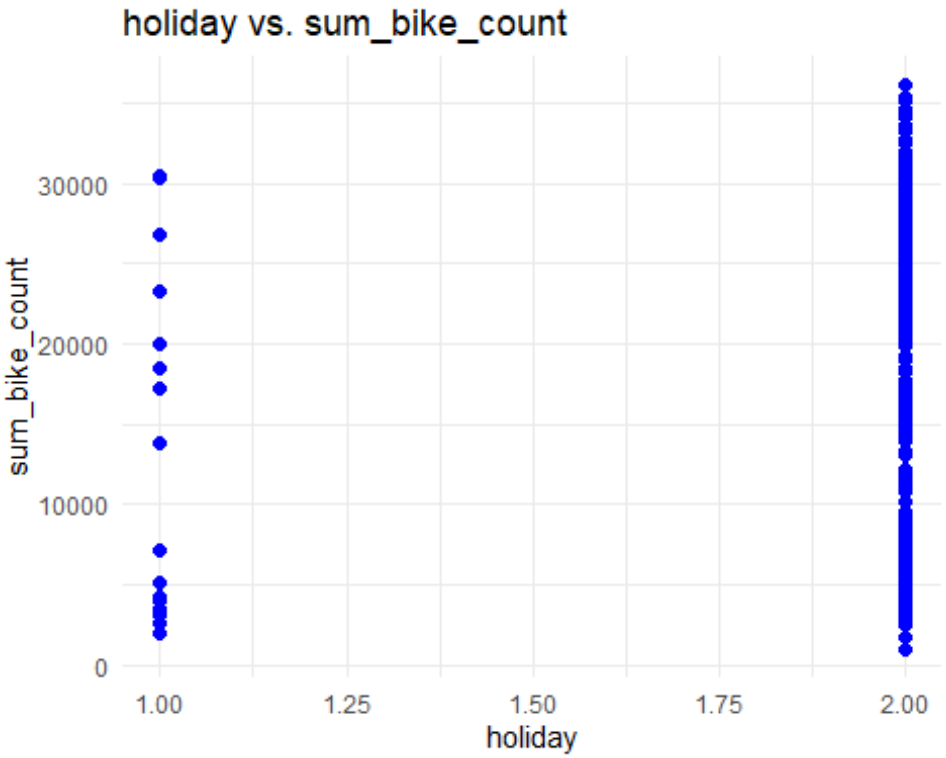
```
##
## $seasons
```




```
##
## $day_of_week
```


day_of_week vs. sum_bike_count
sum_bike_count
30000
20000
10000
0
2
4
6
day_of_week

```
##
## $holiday
```

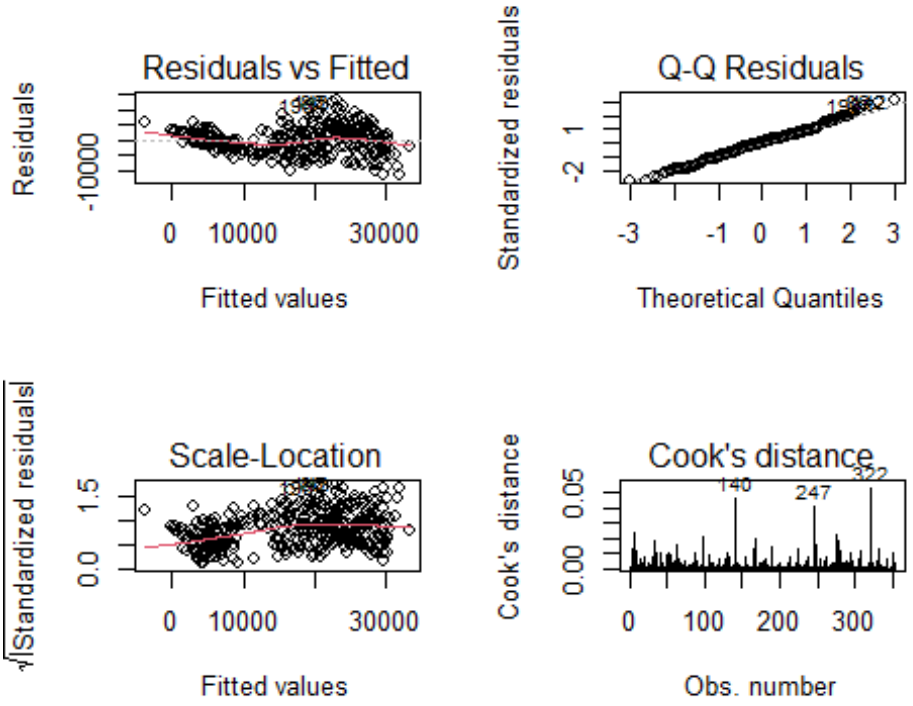

holiday vs. sum_bike_count
sum_bike_count
30000
20000
10000
0
1.00
1.25
1.50
1.75
2.00
holiday

6.2.2 Xây dựng mô hình cơ bản.

```
data_model2 <- data_model2 |> dplyr::select(-date)
model_3 <- lm(sum_bike_count ~ ., data = data_model2 )
summary(model_3)
```

```
##
## Call:
## lm(formula = sum_bike_count ~ ., data = data_model2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11294.2  -2722.3    323.8   2596.2  12930.1
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           29476.0428  9788.2396   3.011 0.002795 **
## mean_humidity          -231.2355   107.8403  -2.144 0.032720 *
## mean_wind_speed       -1465.2355   426.9315  -3.432 0.000673 ***
## mean_visibility           0.6342     0.6209   1.022 0.307721
## mean_tempature         -477.1974   365.3256  -1.306 0.192357
## mean_dew_point_temp     927.0607   387.0279   2.395 0.017144 *
## mean_solar_radiation 11641.7484  1237.1334   9.410  < 2e-16 ***
## mean_rainfall        -3883.8047   595.6025  -6.521 2.52e-10 ***
## mean_snowfall         -355.6750   666.4449  -0.534 0.593904
## seasons              -2137.7477   226.4028  -9.442  < 2e-16 ***
## day_of_week           -355.2070   114.8823  -3.092 0.002153 **
## holiday               3263.8718  1060.5901   3.077 0.002257 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4245 on 341 degrees of freedom
## Multiple R-squared:  0.8232, Adjusted R-squared:  0.8175
## F-statistic: 144.3 on 11 and 341 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model_3 , which=1:4)
```

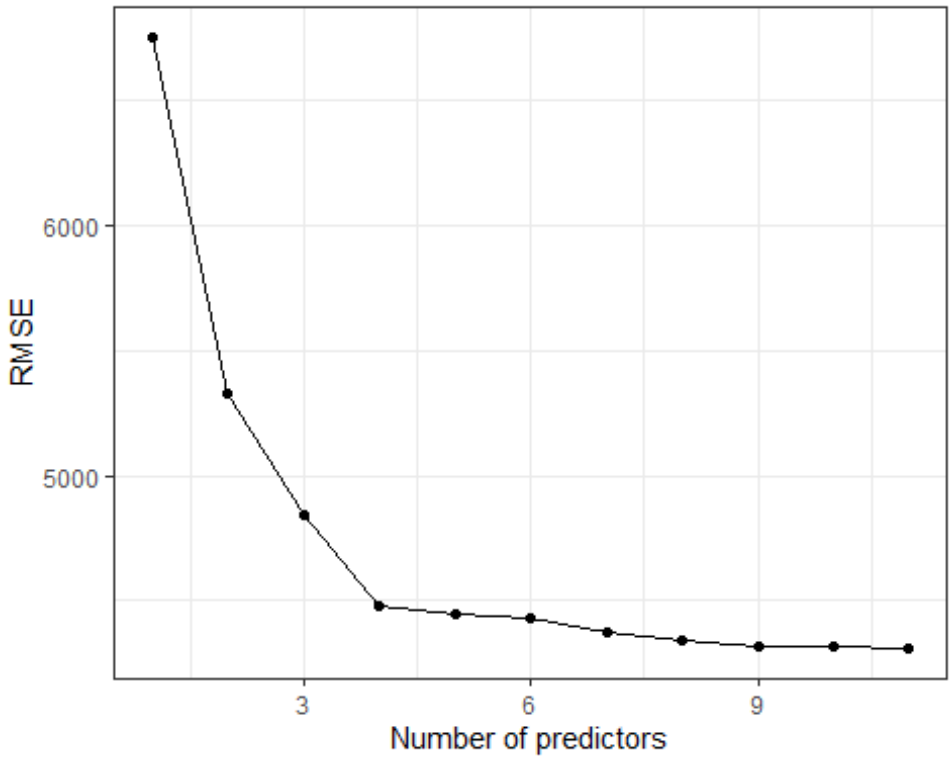


6.2.3 Select feature.

Sử dụng phương pháp hồi quy từng bước.

```
# Tạo hàm predict cho regsubsets
predict.regsubsets <- function(object, newdata, id_model){
  form <- as.formula(object$call[[2]])
  x_mat <- model.matrix(form, newdata)
  coef_est <- coef(object, id = id_model)
  x_vars <- names(coef_est)
  x_mat <- x_mat[, x_vars, drop = FALSE]  # Đảm bảo rằng x_mat có các biến
cần thiết
  res <- x_mat %*% coef_est
  return(as.numeric(res))
}

n_data_model2 <- nrow(data_model2)
k <- 5
set.seed(21)
folds <- sample(rep(1:k, length = n_data_model2))

# Đảm bảo rằng số lượng tối đa các biến dự đoán không vượt quá số biến thực
tế
nvmax_actual <- min(13, ncol(data_model2) - 1)
cv_error_model2_rj <- matrix(0, nrow = k, ncol = nvmax_actual)

for(r in 1:k){
```

```
  data_model2_train_r <- data_model2[folds != r, ]
  data_model2_test_r <- data_model2[folds == r, ]

  out_subset_model2_folds <- regsubsets(x = sum_bike_count ~ ., data =
data_model2_train_r,
                                        method = "exhaustive", nvmax =
nvmax_actual, really.big = TRUE)

  for(j in 1:nvmax_actual){
    pred_rj <- predict.regsubsets(out_subset_model2_folds,
                                  newdata = data_model2_test_r, id_model = j)
    cv_error_model2_rj[r, j] <- sqrt(mean((data_model2_test_r$sum_bike_count
- pred_rj)^2))
  }
}

cv_error_model2 <- colMeans(cv_error_model2_rj)
ggplot(data = data.frame(x = c(1:nvmax_actual), y = cv_error_model2),
  mapping = aes(x = x, y = y)) +
  geom_point() +
  geom_line() +
  labs(x = "Number of predictors", y = "RMSE") +
  theme_bw()
```

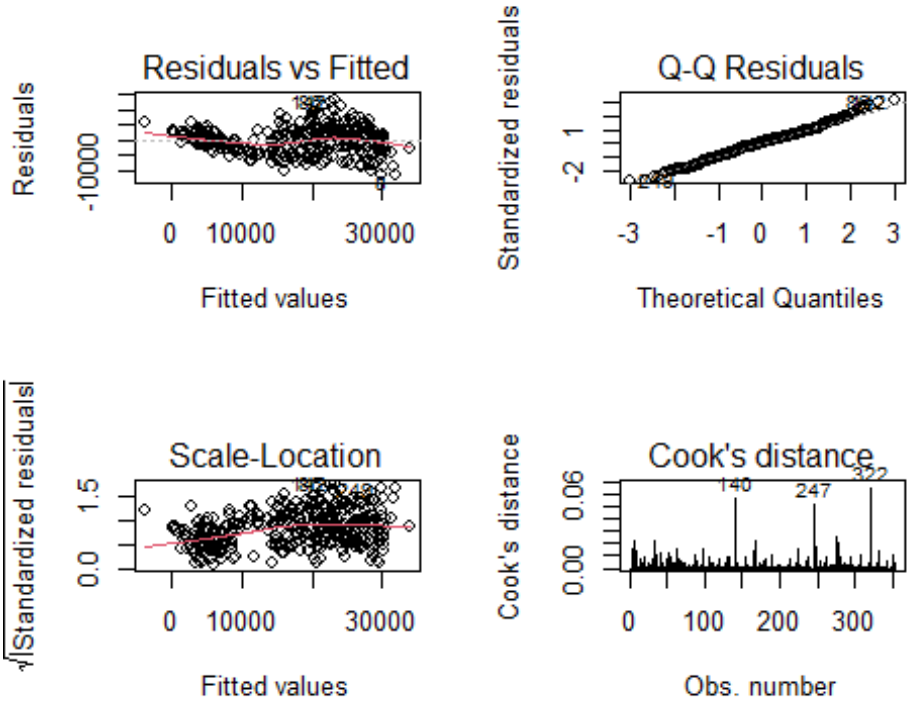


Việc sử dụng RMSE không thấy được số lượng thuộc tính hợp lý, nên ta sẽ chuyển sang phương pháp khác.

```r
data_model2 <- data_model2
regsubset <- regsubsets(x = sum_bike_count ~ ., data = data_model2, nvmax =
15,
                                  method = "exhaustive")

reg_summary <- summary(regsubset)

# Tiêu chí Mallow's Cp
best_model_cp <- which.min(reg_summary$cp)

# Tiêu chí BIC
best_model_bic <- which.min(reg_summary$bic)

# Tiêu chí Adjusted R²
best_model_adjr2 <- which.max(reg_summary$adjr2)

# Hiển thị số lượng biến tốt nhất dựa trên các tiêu chí
cat("Số lượng biến tốt nhất dựa trên tiêu chí Mallow's Cp:", best_model_cp,
"\n")

## Số lượng biến tốt nhất dựa trên tiêu chí Mallow's Cp: 9

cat("Số lượng biến tốt nhất dựa trên tiêu chí BIC:", best_model_bic, "\n")

## Số lượng biến tốt nhất dựa trên tiêu chí BIC: 7

# Lấy các biến của mô hình tốt nhất dựa trên tiêu chí Mallow's Cp
best_features_cp <- names(coef(regsubset, best_model_cp))
cat("Các biến tốt nhất dựa trên tiêu chí Mallow's Cp:", best_features_cp,
"\n")

## Các biến tốt nhất dựa trên tiêu chí Mallow's Cp: (Intercept) mean_humidity
mean_wind_speed mean_tempature mean_dew_point_temp mean_solar_radiation
mean_rainfall seasons day_of_week holiday

# Lấy các biến của mô hình tốt nhất dựa trên tiêu chí BIC
best_features_bic <- names(coef(regsubset, best_model_bic))
cat("Các biến tốt nhất dựa trên tiêu chí BIC:", best_features_bic, "\n")

## Các biến tốt nhất dựa trên tiêu chí BIC: (Intercept) mean_wind_speed
mean_tempature mean_solar_radiation mean_rainfall seasons day_of_week holiday

# Hợp nhất các biến của cả 3 tiêu chí
all_best_features <- unique(c(best_features_cp, best_features_bic))

# Hiển thị tổng hợp các biến từ 3 tiêu chí
cat("Tổng hợp các biến từ 2 tiêu chí:", all_best_features, "\n")

## Tổng hợp các biến từ 2 tiêu chí: (Intercept) mean_humidity mean_wind_speed
mean_tempature mean_dew_point_temp mean_solar_radiation mean_rainfall seasons
day_of_week holiday
```

6.2.4 Building model

```
# Tạo công thức hồi quy tuyến tính
formula <- as.formula(paste("sum_bike_count ~", paste(all_best_features[-1],
collapse = " + ")))

# Xây dựng mô hình hồi quy tuyến tính
model_2 <- lm(formula, data = data_model2)
summary(model_2)

##
## Call:
## lm(formula = formula, data = data_model2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11238.6  -2759.9    164.7   2514.5  12982.2
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           33764.66    8987.59   3.757 0.000202 ***
## mean_humidity          -278.28      99.19  -2.805 0.005311 **
## mean_wind_speed       -1370.76     417.51  -3.283 0.001132 **
## mean_tempature         -561.85     357.03  -1.574 0.116483
## mean_dew_point_temp    1033.33     374.79   2.757 0.006144 **
## mean_solar_radiation  11474.40    1225.48   9.363  < 2e-16 ***
## mean_rainfall         -3744.08     581.20  -6.442 3.99e-10 ***
## seasons               -2176.48     223.05  -9.758  < 2e-16 ***
## day_of_week            -351.47     114.64  -3.066 0.002343 **
## holiday                3256.45    1059.06   3.075 0.002275 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4241 on 343 degrees of freedom
## Multiple R-squared:  0.8225, Adjusted R-squared:  0.8179
## F-statistic: 176.7 on 9 and 343 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(model_2 , which=1:4)
```



6.2.5 Mở rộng mô hình.

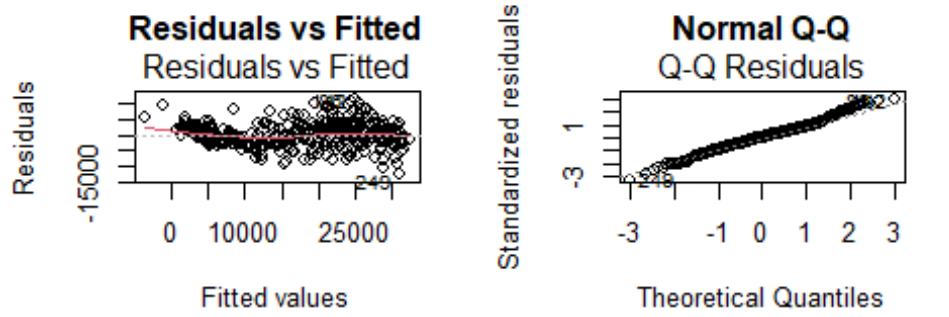
```
knots_mean_solar_radiation<- quantile(data_model2$mean_solar_radiation, probs
= c(0.5))

model_2_expand <- lm(sum_bike_count ~ mean_humidity +
                       mean_wind_speed +
                       bs(mean_solar_radiation, knots =
knots_mean_solar_radiation, degree = 2) +
                       poly(mean_dew_point_temp, degree = 3) +
                       poly(mean_rainfall, degree = 2) +
                       poly(seasons, 2) +
                       holiday, data = data_model2
    )
summary(model_2_expand)

##
## Call:
## lm(formula = sum_bike_count ~ mean_humidity + mean_wind_speed +
##     bs(mean_solar_radiation, knots = knots_mean_solar_radiation,
##         degree = 2) + poly(mean_dew_point_temp, degree = 3) +
##     poly(mean_rainfall, degree = 2) + poly(seasons, 2) + holiday,
##     data = data_model2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12342.2  -2217.3     -0.5   2258.2  11066.4
```

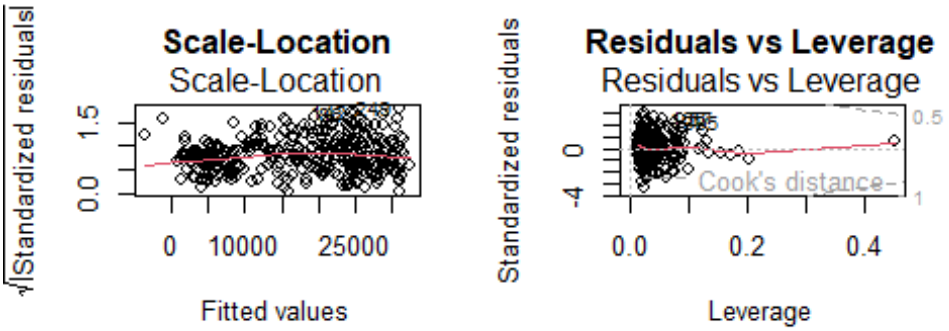
```
##
## Coefficients:
##
Estimate
## (Intercept)
12842.48
## mean_humidity
-166.95
## mean_wind_speed
-971.30
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
9863.27
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
10090.63
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
16013.27
## poly(mean_dew_point_temp, degree = 3)1
145443.76
## poly(mean_dew_point_temp, degree = 3)2
-25027.88
## poly(mean_dew_point_temp, degree = 3)3
-31176.59
## poly(mean_rainfall, degree = 2)1
-30089.19
## poly(mean_rainfall, degree = 2)2
10212.53
## poly(seasons, 2)1
-15044.45
## poly(seasons, 2)2
22479.78
## holiday
3536.96
##
Std. Error
## (Intercept)
2941.51
## mean_humidity
28.69
## mean_wind_speed
397.16
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
1611.90
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
1466.45
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
1886.12
## poly(mean_dew_point_temp, degree = 3)1
9274.89
## poly(mean_dew_point_temp, degree = 3)2
5392.83
```

```
## poly(mean_dew_point_temp, degree = 3)3
4230.52
## poly(mean_rainfall, degree = 2)1
5269.53
## poly(mean_rainfall, degree = 2)2
4249.39
## poly(seasons, 2)1
6011.22
## poly(seasons, 2)2
5669.13
## holiday
972.16
##
t value
## (Intercept)
4.366
## mean_humidity
-5.819
## mean_wind_speed
-2.446
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
6.119
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
6.881
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
8.490
## poly(mean_dew_point_temp, degree = 3)1
15.681
## poly(mean_dew_point_temp, degree = 3)2
-4.641
## poly(mean_dew_point_temp, degree = 3)3
-7.369
## poly(mean_rainfall, degree = 2)1
-5.710
## poly(mean_rainfall, degree = 2)2
2.403
## poly(seasons, 2)1
-2.503
## poly(seasons, 2)2
3.965
## holiday
3.638
##
Pr(>|t|)
## (Intercept)
1.68e-05
## mean_humidity
1.37e-08
## mean_wind_speed
0.014968
```

```
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
2.60e-09
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
2.88e-11
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
6.56e-16
## poly(mean_dew_point_temp, degree = 3)1
< 2e-16
## poly(mean_dew_point_temp, degree = 3)2
4.96e-06
## poly(mean_dew_point_temp, degree = 3)3
1.32e-12
## poly(mean_rainfall, degree = 2)1
2.47e-08
## poly(mean_rainfall, degree = 2)2
0.016785
## poly(seasons, 2)1
0.012794
## poly(seasons, 2)2
8.94e-05
## holiday
0.000317
##
## (Intercept)
***
## mean_humidity
***
## mean_wind_speed
*
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
***
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
***
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
***
## poly(mean_dew_point_temp, degree = 3)1
***
## poly(mean_dew_point_temp, degree = 3)2
***
## poly(mean_dew_point_temp, degree = 3)3
***
## poly(mean_rainfall, degree = 2)1
***
## poly(mean_rainfall, degree = 2)2
*
## poly(seasons, 2)1
*
## poly(seasons, 2)2
***
## holiday
```

```
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3839 on 339 degrees of freedom
## Multiple R-squared:  0.8563, Adjusted R-squared:  0.8507
## F-statistic: 155.3 on 13 and 339 DF,  p-value: < 2.2e-16

# Kiểm tra sự độ lập thặng dư của mô hình
par(mfrow = c(2, 2))  # Hiển thị 4 đồ thị cùng một lúc

# Residuals vs Fitted
plot(model_2_expand, which = 1, main = "Residuals vs Fitted")

# Normal Q-Q
plot(model_2_expand, which = 2, main = "Normal Q-Q")

# Scale-Location
plot(model_2_expand, which = 3, main = "Scale-Location")

# Residuals vs Leverage
plot(model_2_expand, which = 5, main = "Residuals vs Leverage")
```






```
par(mfrow = c(1, 1))  # Trở về chế độ hiển thị mặc định
```

Nhận xét:

- Theo đồ thị **Residuals vs Fitted Values:** Điều này cho thấy rằng mối quan hệ tuyến tính là hợp lý và giả định homoscedasticity (phương sai đồng nhất) được đáp ứng.
- Theo đồ thị **Normal Q-Q Plot:** Thặng dư có phân phối gần với phân phối chuẩn, đáp ứng giả định normality của thặng dư.
- Theo đồ thị **Scale-Location Plot:** Dù có mở rộng mô hình nhưng đường nằm ngang không hoàn toàn, thế nên có thể sử dụng mô hình phi tiến có thể sẽ tốt hơn.

```
# Kiểm tra đa cộng tuyến (VIF)
library(car)
vif(model_2_expand)

##
GVIF
## mean_humidity
4.343968
## mean_wind_speed
1.343859
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)
5.303889
## poly(mean_dew_point_temp, degree = 3)
10.797242
## poly(mean_rainfall, degree = 2)
2.143055
## poly(seasons, 2)
4.779637
## holiday
1.037560
##
Df
## mean_humidity
1
## mean_wind_speed
1
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)
3
## poly(mean_dew_point_temp, degree = 3)
3
## poly(mean_rainfall, degree = 2)
2
## poly(seasons, 2)
2
## holiday
1
##
GVIF^(1/(2*Df))
## mean_humidity
2.084219
## mean_wind_speed
```

```
1.159249
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)
1.320583
## poly(mean_dew_point_temp, degree = 3)
1.486684
## poly(mean_rainfall, degree = 2)
1.209925
## poly(seasons, 2)
1.478593
## holiday
1.018607
```

Nhận xét: `GVIF^(1/(2*Df))` của tất cả các biến đều dưới 2, cho thấy không có vấn đề đa cộng tuyến nghiêm trọng giữa các biến độc lập. Điều này có nghĩa là các biến không bị phụ thuộc tuyến tính lẫn nhau một cách đáng kể và mô hình có độ ổn định cao.

```
# Kiểm tra Normality
shapiro.test(residuals(model_2_expand))

##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_2_expand)
## W = 0.99213, p-value = 0.05889
```

Nhận xét: P-value > 0.05 cho thấy thặng dư có phân phối gần với phân phối chuẩn. Điều này đáp ứng giả định normality của thặng dư, cho phép sử dụng các kiểm định thống kê tiếp theo và tăng độ tin cậy của các ước lượng từ mô hình.

```
summary(model_2_expand)$coefficients

##
Estimate
## (Intercept)
12842.4758
## mean_humidity
-166.9517
## mean_wind_speed
-971.2965
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
9863.2675
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
10090.6329
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
16013.2692
## poly(mean_dew_point_temp, degree = 3)1
145443.7578
## poly(mean_dew_point_temp, degree = 3)2
-25027.8831
## poly(mean_dew_point_temp, degree = 3)3
-31176.5862
```

```
## poly(mean_rainfall, degree = 2)1
-30089.1947
## poly(mean_rainfall, degree = 2)2
10212.5272
## poly(seasons, 2)1
-15044.4516
## poly(seasons, 2)2
22479.7803
## holiday
3536.9579
##
Std. Error
## (Intercept)
2941.50775
## mean_humidity
28.68981
## mean_wind_speed
397.15736
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
1611.90175
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
1466.44801
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
1886.12153
## poly(mean_dew_point_temp, degree = 3)1
9274.89499
## poly(mean_dew_point_temp, degree = 3)2
5392.82527
## poly(mean_dew_point_temp, degree = 3)3
4230.52313
## poly(mean_rainfall, degree = 2)1
5269.53022
## poly(mean_rainfall, degree = 2)2
4249.38738
## poly(seasons, 2)1
6011.21949
## poly(seasons, 2)2
5669.12527
## holiday
972.15511
##
t value
## (Intercept)
4.365950
## mean_humidity
-5.819198
## mean_wind_speed
-2.445621
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
6.119025
```

```
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
6.881003
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
8.490052
## poly(mean_dew_point_temp, degree = 3)1
15.681445
## poly(mean_dew_point_temp, degree = 3)2
-4.640959
## poly(mean_dew_point_temp, degree = 3)3
-7.369440
## poly(mean_rainfall, degree = 2)1
-5.710034
## poly(mean_rainfall, degree = 2)2
2.403294
## poly(seasons, 2)1
-2.502729
## poly(seasons, 2)2
3.965300
## holiday
3.638265
##
Pr(>|t|)
## (Intercept)
1.683348e-05
## mean_humidity
1.370033e-08
## mean_wind_speed
1.496837e-02
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)1
2.598762e-09
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)2
2.881365e-11
## bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)3
6.560454e-16
## poly(mean_dew_point_temp, degree = 3)1
4.682534e-42
## poly(mean_dew_point_temp, degree = 3)2
4.963490e-06
## poly(mean_dew_point_temp, degree = 3)3
1.319340e-12
## poly(mean_rainfall, degree = 2)1
2.469944e-08
## poly(mean_rainfall, degree = 2)2
1.678476e-02
## poly(seasons, 2)1
1.279414e-02
## poly(seasons, 2)2
8.944948e-05
## holiday
3.171993e-04
```

Nhận xét:

- **Hệ số ước lượng (Estimate)**:
  - `mean_humidity`: Hệ số này âm (-166.9517) cho thấy rằng khi độ ẩm trung bình tăng, số lượng xe đạp thuê giảm.
  - `mean_wind_speed`: Hệ số này âm (-971.2965) cho thấy rằng khi tốc độ gió tăng, số lượng xe đạp thuê giảm.
  - `holiday`: Hệ số này dương (3536.9579) cho thấy rằng vào ngày nghỉ lễ, số lượng xe đạp thuê tăng
- **Giá trị t và p-value**:
  - Tất cả các biến trong mô hình đều có giá trị p-value nhỏ hơn 0.05, cho thấy rằng chúng có ý nghĩa thống kê trong mô hình.
- **Phân tích các thành phần hàm cơ sở (bs) và đa thức (poly)**:
  - `bs(mean_solar_radiation, knots = knots_mean_solar_radiation, degree = 2)`: Các hệ số này đều có ý nghĩa thống kê cao (p-value rất nhỏ), cho thấy rằng biến `mean_solar_radiation` có ảnh hưởng phi tuyến tính đáng kể đến số lượng xe đạp thuê.
  - `poly(mean_dew_point_temp, degree = 3)`: Tương tự, các hệ số này cũng có ý nghĩa thống kê cao, cho thấy rằng `mean_dew_point_temp` có ảnh hưởng phi tuyến tính quan trọng.
  - `poly(mean_rainfall, degree = 2)` và `poly(seasons, 2)`: Các hệ số này cũng có ý nghĩa thống kê, cho thấy rằng các biến này có ảnh hưởng phi tuyến tính đến số lượng xe đạp thuê.

7. Tổng kết.

7.1 Tình hình sử dụng xe đạp thuê tại Seoul.

Từ các bảng, đồ thị phân phối đã trình bày ở trên, ta có kết luận rằng.

Dữ liệu từ dịch vụ cho thuê xe đạp ở Seoul từ 01/12/2017 đến 30/11/2018 tiết lộ rằng số lượng xe đạp thuê biến đổi theo giờ trong ngày, với cao điểm từ 7 giờ sáng đến 9 giờ sáng và tăng trở lại sau 15:00 chiều. Về mặt thời gian trong năm, tháng 6 có số lượng thuê cao nhất, trong khi tháng 1 và tháng 12 ghi nhận số lượng thuê thấp nhất. Vào mùa hè, nhu cầu thuê xe đạp tăng lên nhờ thời tiết thuận lợi, trong khi mùa đông có nhu cầu thấp nhất do thời tiết lạnh và tuyết. Thú vị là mùa thu có nhu cầu xe đạp cao hơn mùa xuân, có thể do số ngày mưa ít hơn.

Nhu cầu xe đạp cũng cao hơn vào những ngày làm việc so với ngày nghỉ lễ, cho thấy rằng người dân thường thuê xe đạp vì các lý do khác ngoài mục đích giải trí. Cuối tuần, đặc biệt

khi trùng với ngày lễ, số lượng thuê xe tăng cao nhờ các hoạt động ngoài trời và du lịch. Thời tiết tốt và các ngày lễ cuối tuần có thể làm tăng nhu cầu thuê xe. Đáng chú ý, hầu hết các yếu tố môi trường bên ngoài chỉ có quan hệ tuyến tính yếu, cho thấy sự phức tạp và đa dạng trong các yếu tố ảnh hưởng đến nhu cầu thuê xe đạp.

7.2 Mô hình dự đoán.

Mô hình đã chi tiết hóa mối quan hệ giữa các biến thời tiết, mùa và ngày lễ với lượng xe đạp được thuê. Với tỉ lệ dự đoán đúng là 85,63 % và các tham số sử dụng là là độ ẩm, mức gió, lượng mưa, sương mù, bức xạ mặt trời, thời gian, dịp lễ đặt biệt.

- Tất cả các biến trong mô hình đều có ý nghĩa thống kê (p-value < 0.05), cho thấy chúng có tác động đáng kể đến số lượng xe đạp thuê.
- Các biến như mean_solar_radiation và mean_dew_point_temp có ảnh hưởng phi tuyến tính mạnh mẽ, được xác định bởi các hàm bs và poly với các hệ số có ý nghĩa thống kê cao.
- Thặng dư có phân phối gần với chuẩn và không có vấn đề đa cộng tuyến lớn giữa các biến, làm tăng độ tin cậy của mô hình.
- Mô hình đáp ứng được giả định về homoscedasticity và normality của thặng dư, mặc dù có sự mở rộng.

Tổng kết lại, mô hình tuyến tính hiện tại đáp ứng các giả định quan trọng và có ý nghĩa thống kê. Tuy nhiên, một số biến có ảnh hưởng phi tuyến tính đáng kể và việc sử dụng mô hình phi tuyến có thể cải thiện độ chính xác của dự đoán.

7.3 Chiến lược kinh doanh đề ra.

Từ những tổng kết trên, ta có thể đề xuất một số chiến lược kinh doanh sau:

- Tận dụng thời tiết và ngày lễ:
  - Mùa cao điểm (mùa xuân): Tập trung vào chiến dịch tiếp thị và quảng cáo, cung cấp các gói khuyến mãi đặc biệt cho các hoạt động ngoài trời khi nhu cầu thuê xe cao.
  - Mùa thấp điểm (mùa hè): Cung cấp các gói thuê xe dài hạn với giá cả cạnh tranh để thu hút khách du lịch.
- Chú trọng vào các yếu tố môi trường:
  - Độ ẩm và tốc độ gió: Cung cấp bảo hiểm cho thuê xe để giảm thiểu rủi ro từ thời tiết xấu. Cung cấp thông tin dự báo thời tiết để hỗ trợ quyết định thuê xe của khách hàng.
  - Bức xạ mặt trời: Cung cấp nón che và các lựa chọn bảo vệ khỏi ánh nắng mặt trời.

- Chiến lược tận dụng thời tiết:
  - Thời tiết lạnh (mùa đông): Quảng cáo và khuyến mãi vào những ngày thời tiết thuận lợi để kích thích nhu cầu thuê xe.
  - Thời tiết nóng (mùa hè): Cung cấp các dịch vụ bảo vệ khỏi nắng nóng để thu hút khách hàng muốn tham gia các hoạt động ngoài trời.
- Các mùa trong năm:
  - Thiết kế chiến dịch quảng cáo và khuyến mãi dựa trên các mùa vụ để tối ưu hóa nhu cầu thuê xe đạp theo từng mùa.

Tăng cường hợp tác với các sự kiện và lễ hội để nâng cao thương hiệu và thu hút khách hàng mới.