

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)

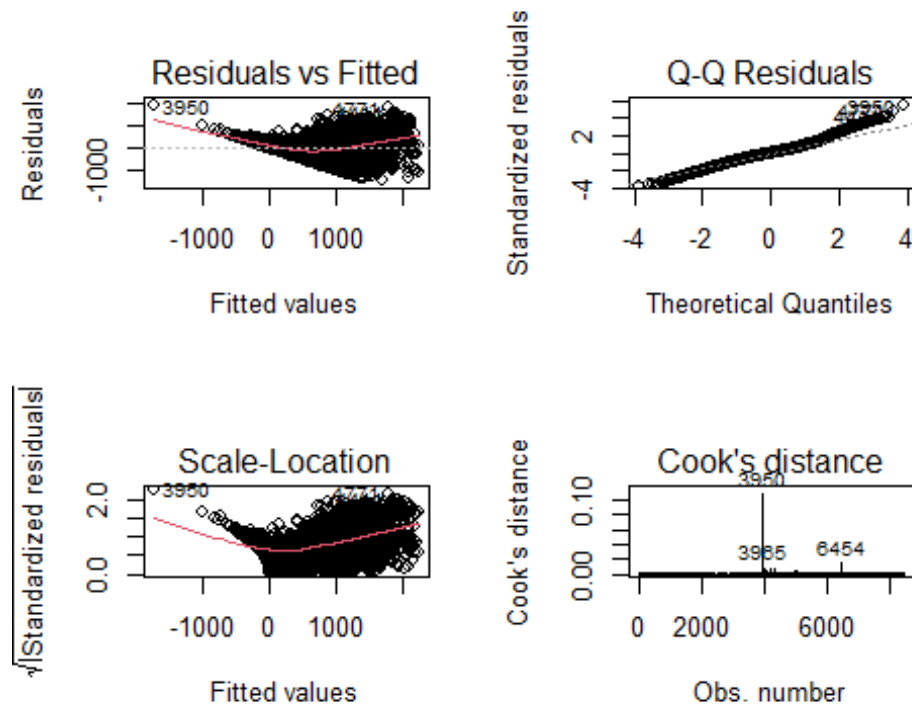
##
## Call:
## lm(formula = rented_bike_count ~ ., data = data_model1)
##
## Residuals:
```

```

##      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 thống 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)

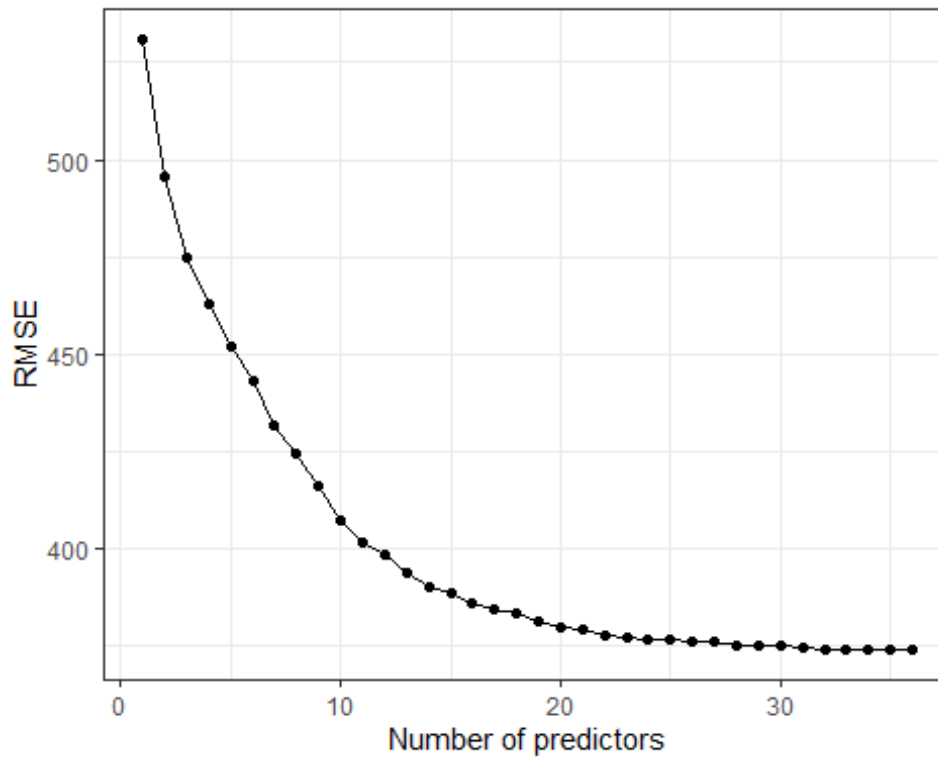
for(r in 1:k){
  data_model1_train_r <- data_model1[folds != r, ]
  data_model1_test_r <- data_model1[folds == r, ]

  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 Mallows 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 Mallows Cp:", best_model_cp,
"\n")

## Số lượng biến tốt nhất dựa trên tiêu chí Mallows 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

# 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 tính vừa tìm được.

```
# 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)

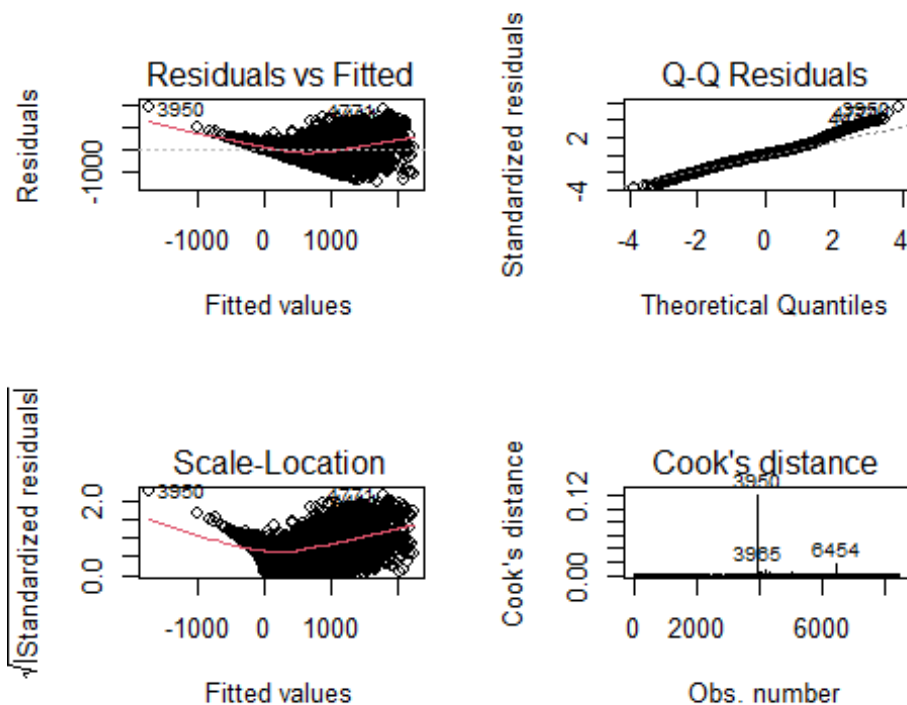
##
## Call:
## lm(formula = formula, data = data_model1)
##
## Residuals:
```

```

##      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 vẽ tốt hơn và các biến trong mô hình đều 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

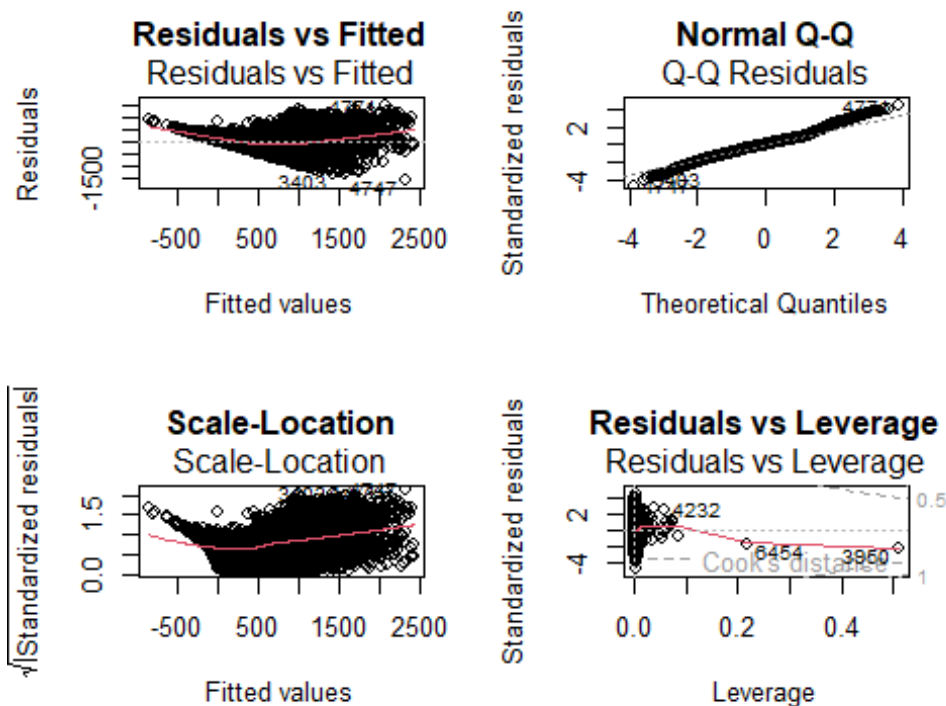
par(mfrow = c(2, 2)) # Hiển thị 4 đồ thị cùng một lúc

# 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_temperature = 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_temperature <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)

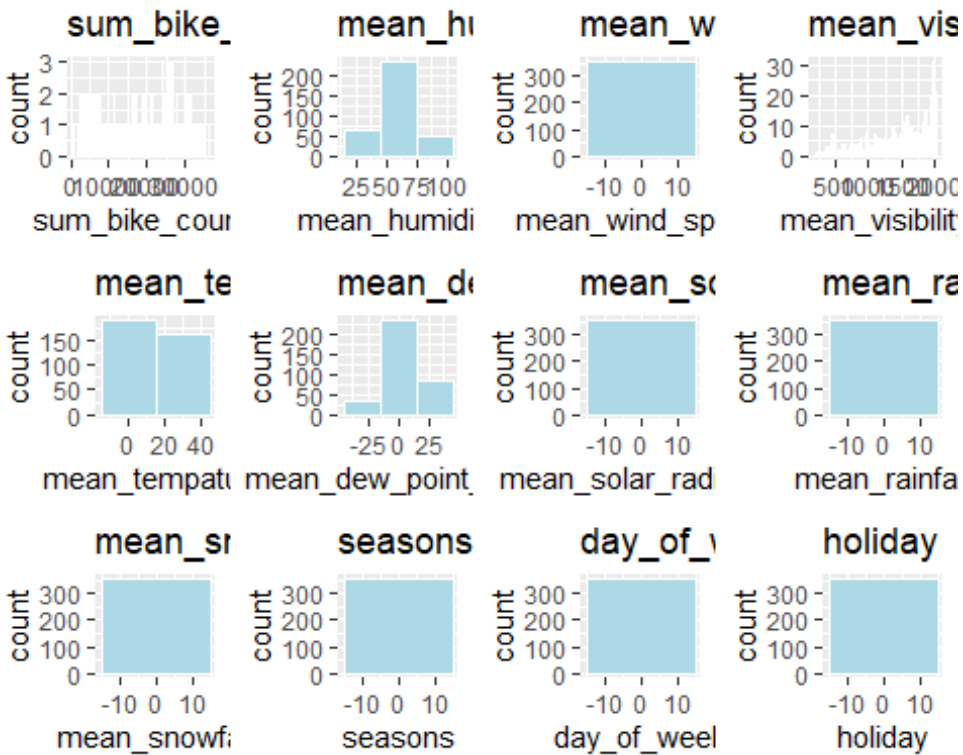
numeric_columns <- sapply(data_model2, is.numeric)
numeric_data <- data_model2[, numeric_columns]

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)
})

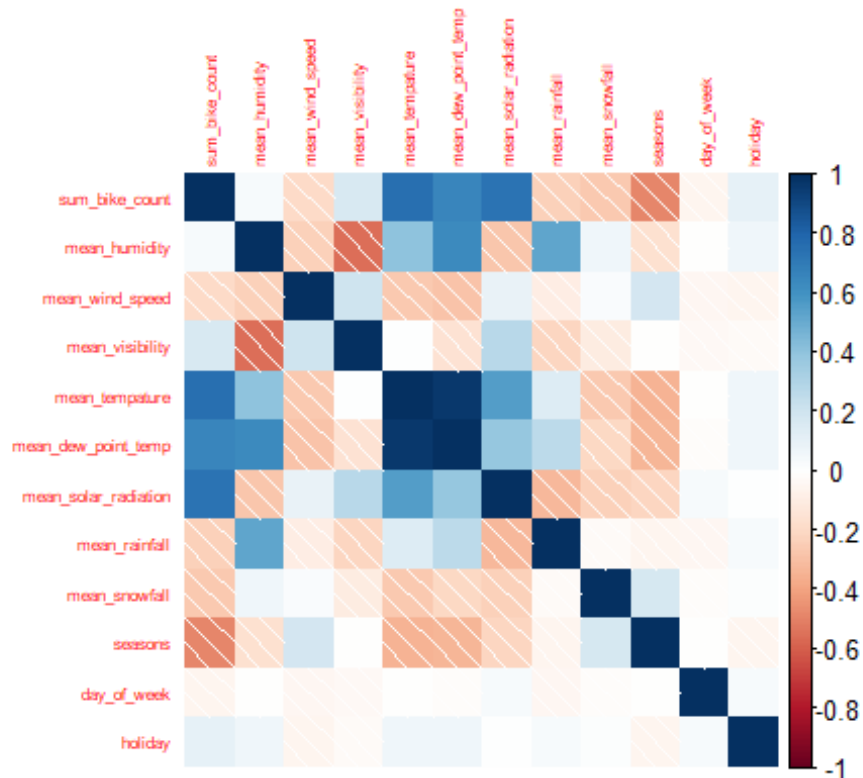
# Sắp xếp các biểu đồ theo dạng lưới
do.call(grid.arrange, c(plots, ncol = num_cols))

```



```
correlation_matrix <- cor(numeric_data)

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



```
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()

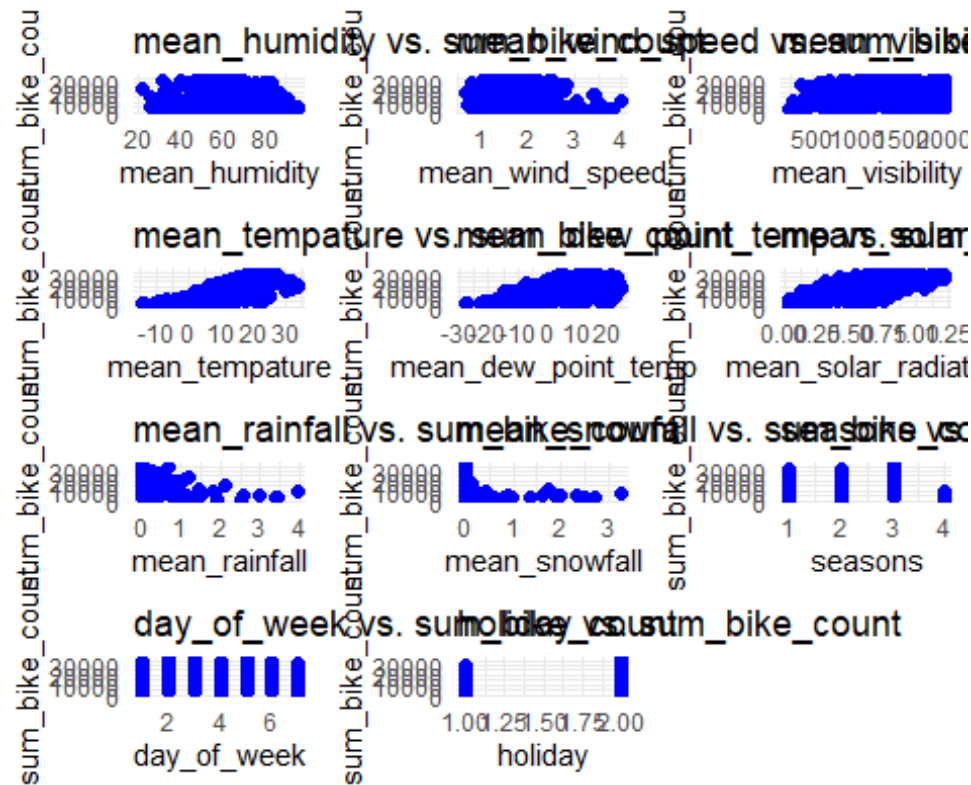
      # Add the plot to the list
      plots[[col]] <- scatter_plot
    }
  }

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

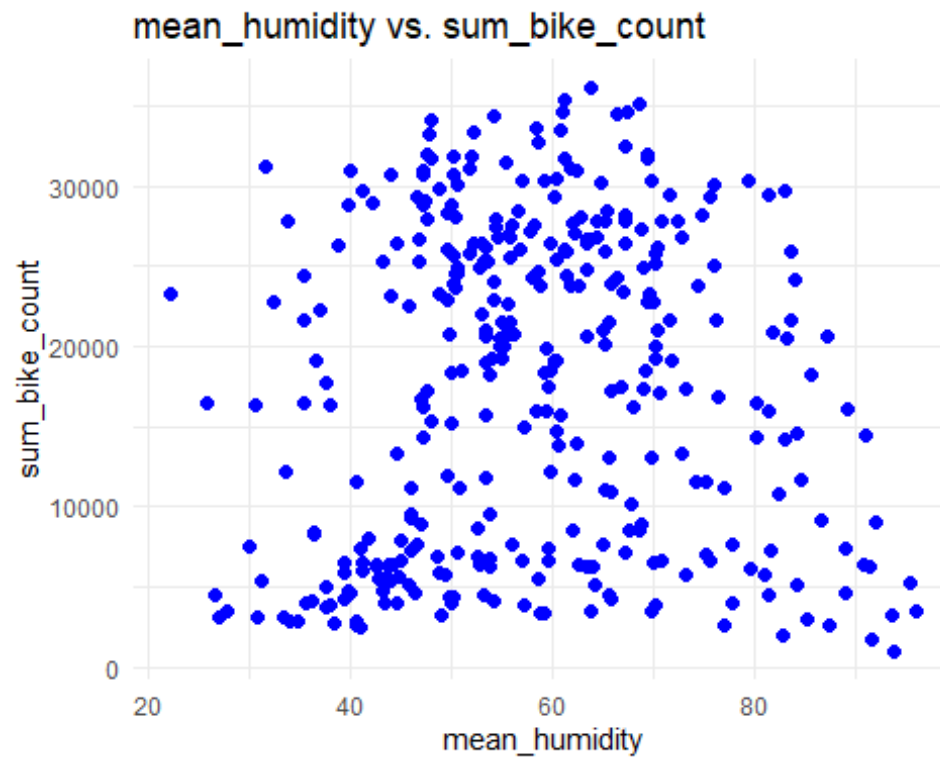
  # Return the list of plots (optional)
  return(plots)
}
```



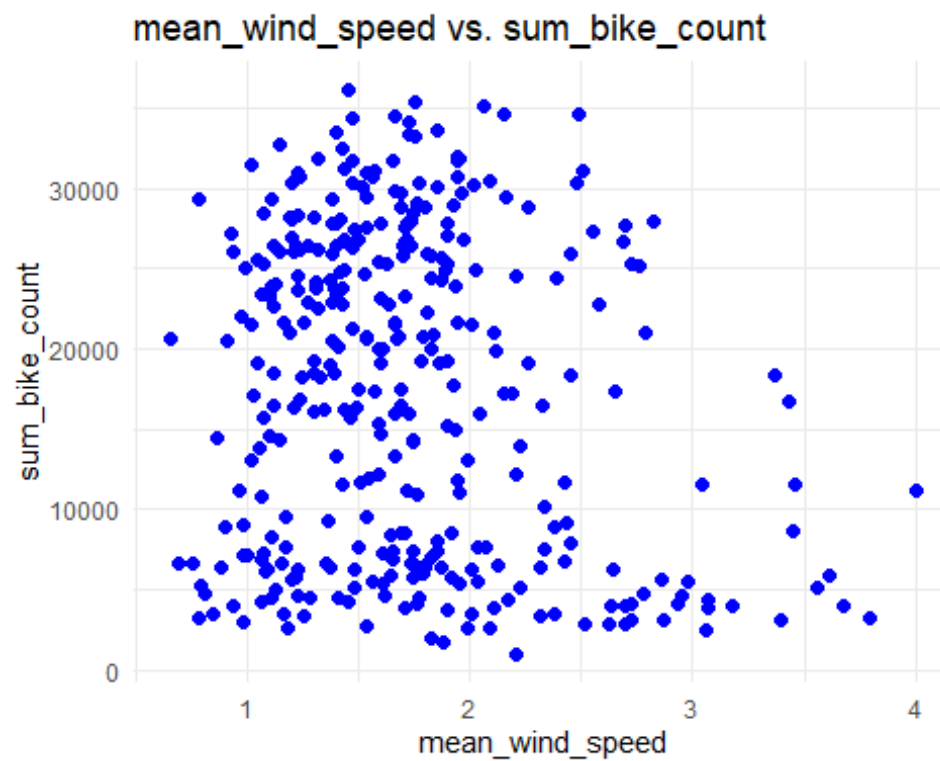
```
# Call the function with numeric columns and target variable
multi.scatter(data_model2, "sum_bike_count")
```



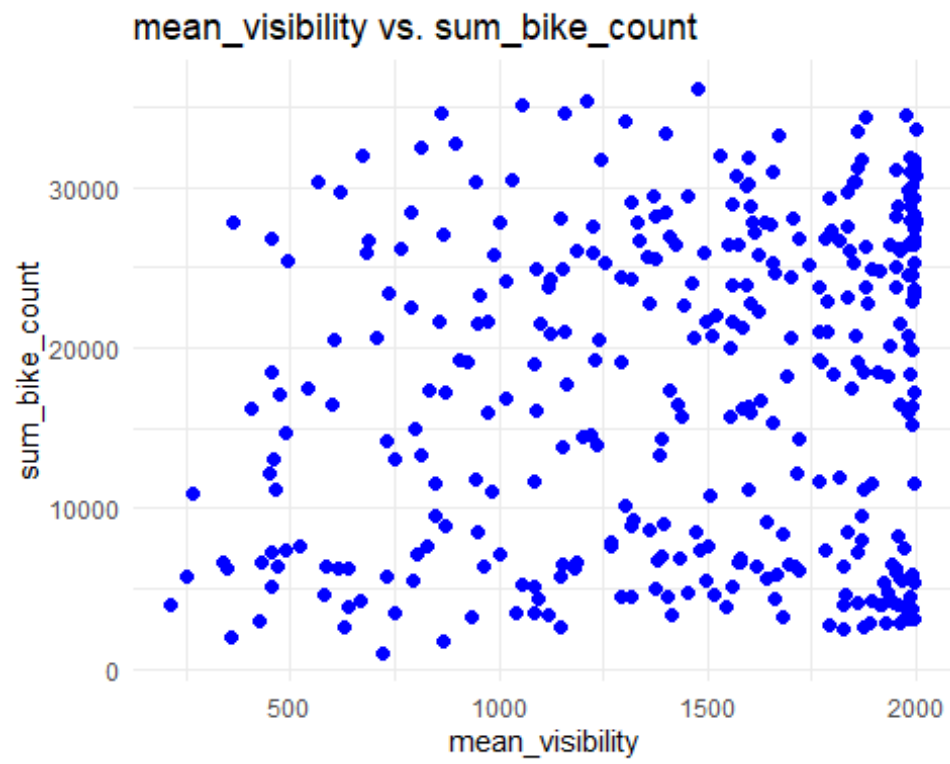
```
## $mean_humidity
```



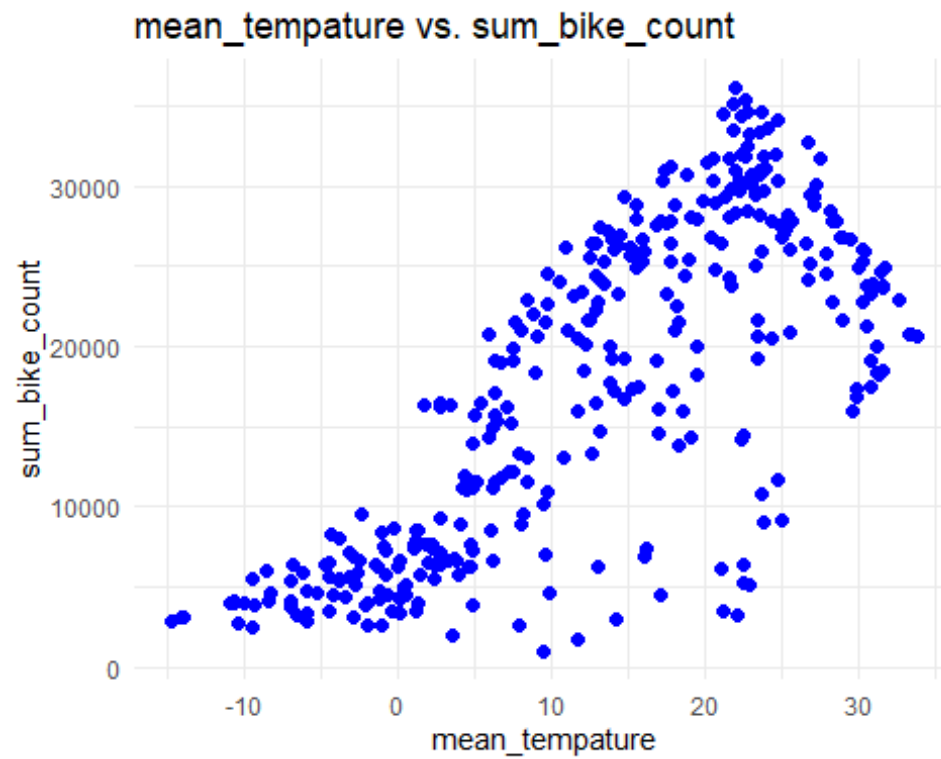
```
##  
## $mean_wind_speed
```



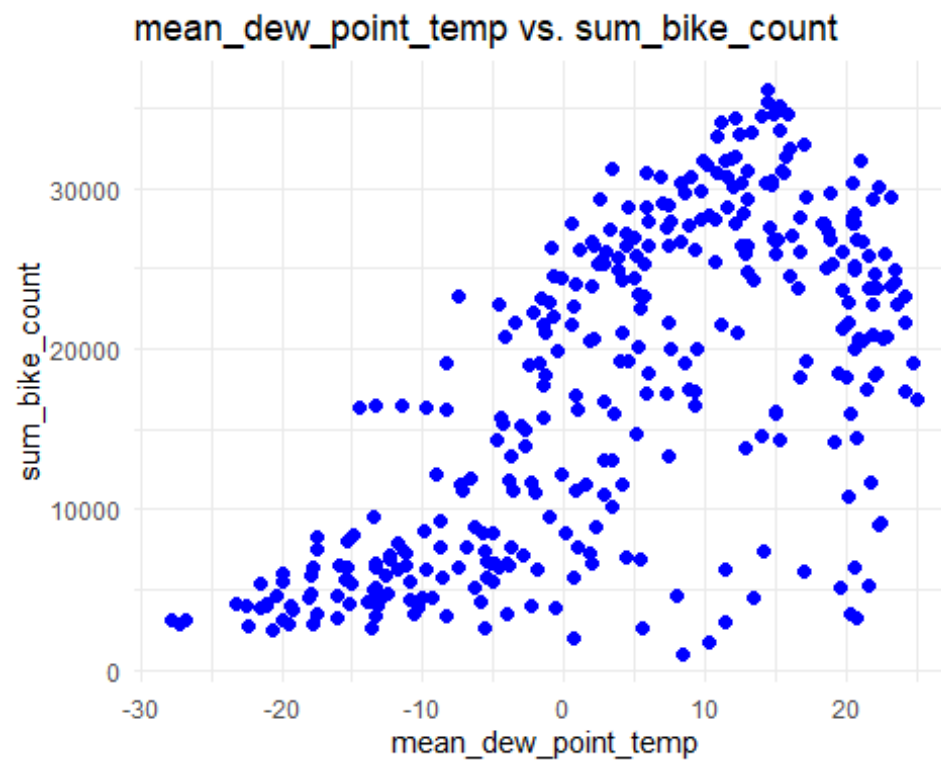
```
##  
## $mean_visibility
```



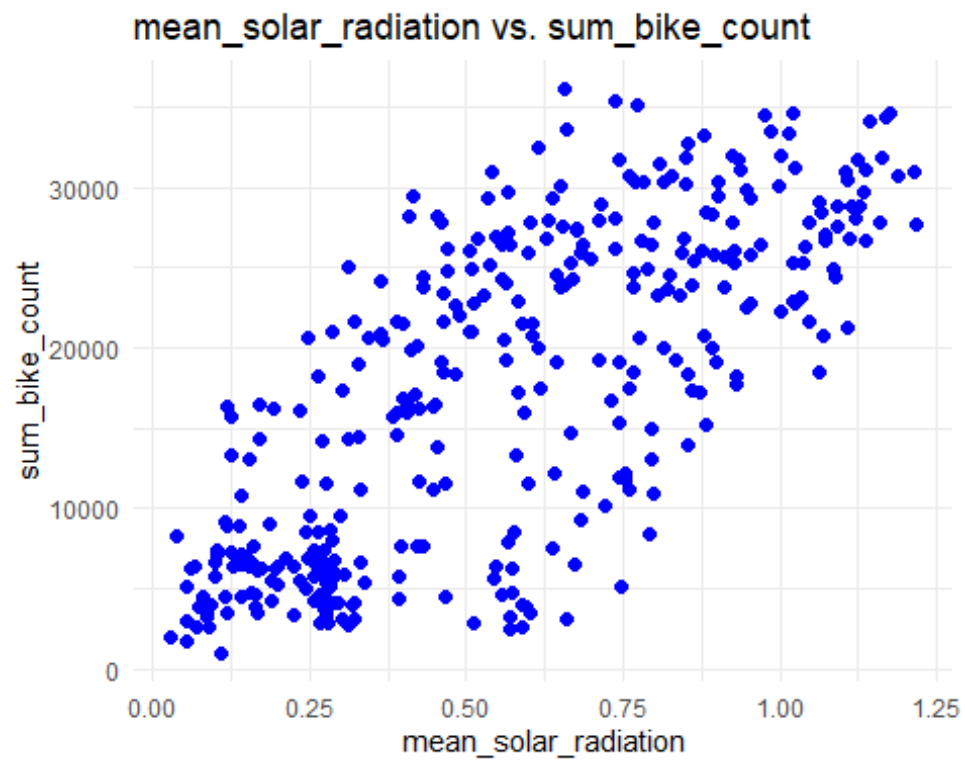
```
##  
## $mean_temperature
```



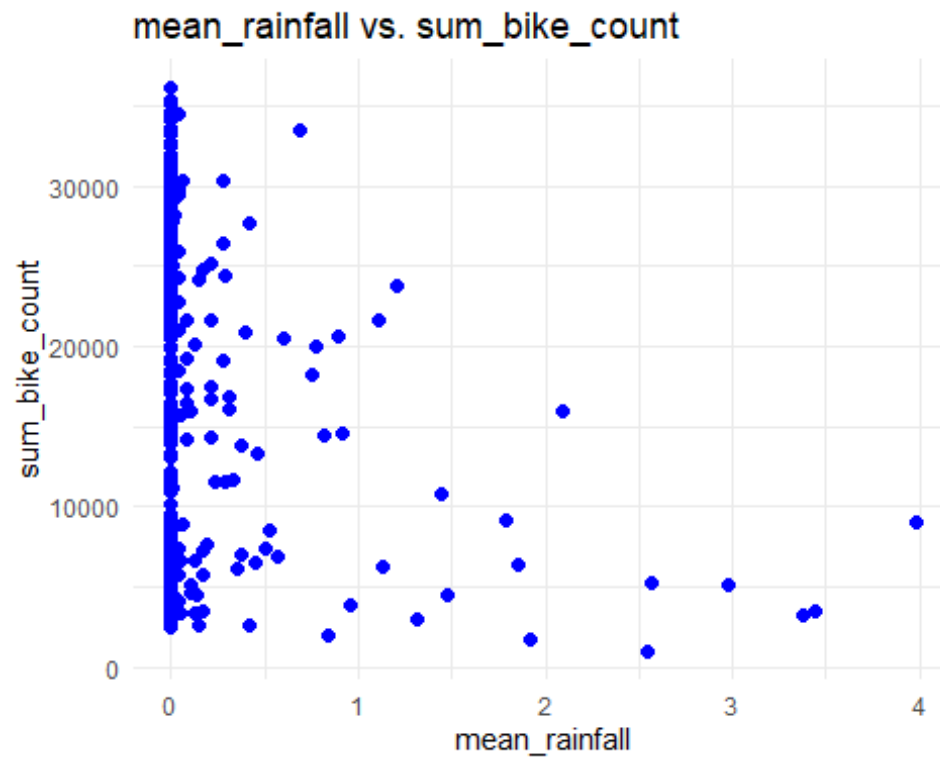
```
##  
## $mean_dew_point_temp
```



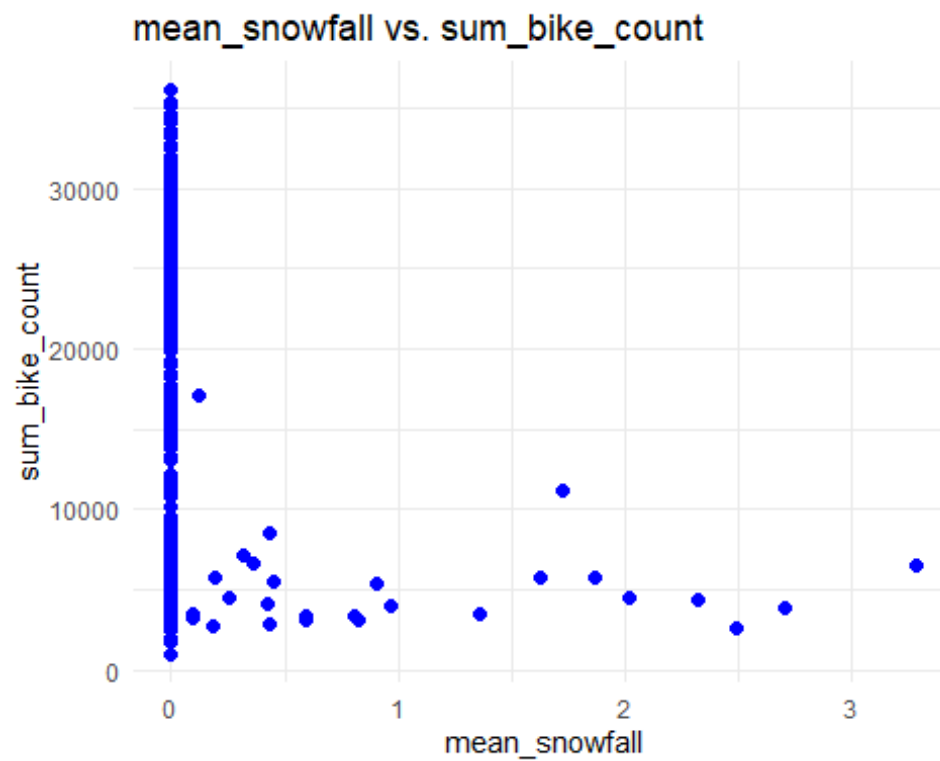
```
##  
## $mean_solar_radiation
```



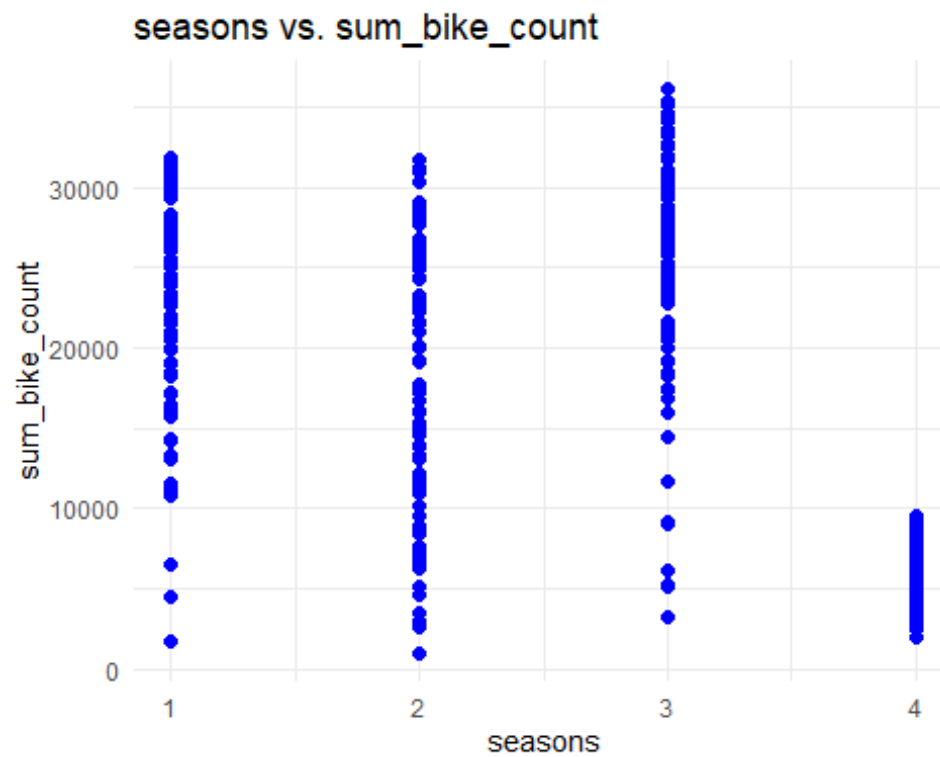
```
##  
## $mean_rainfall
```



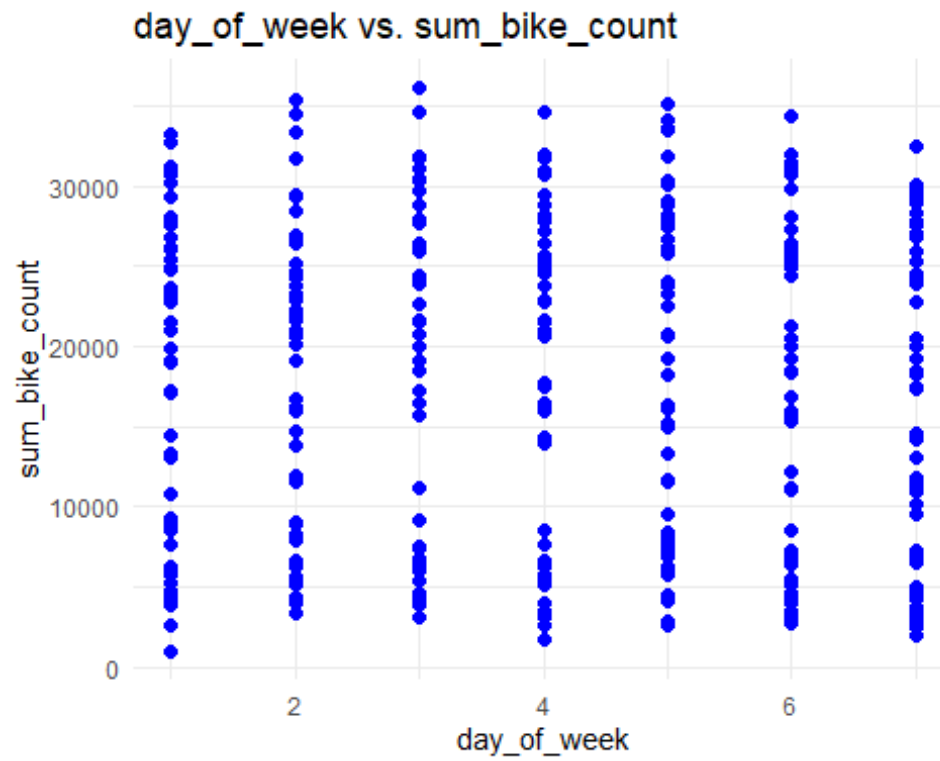
```
##  
## $mean_snowfall
```



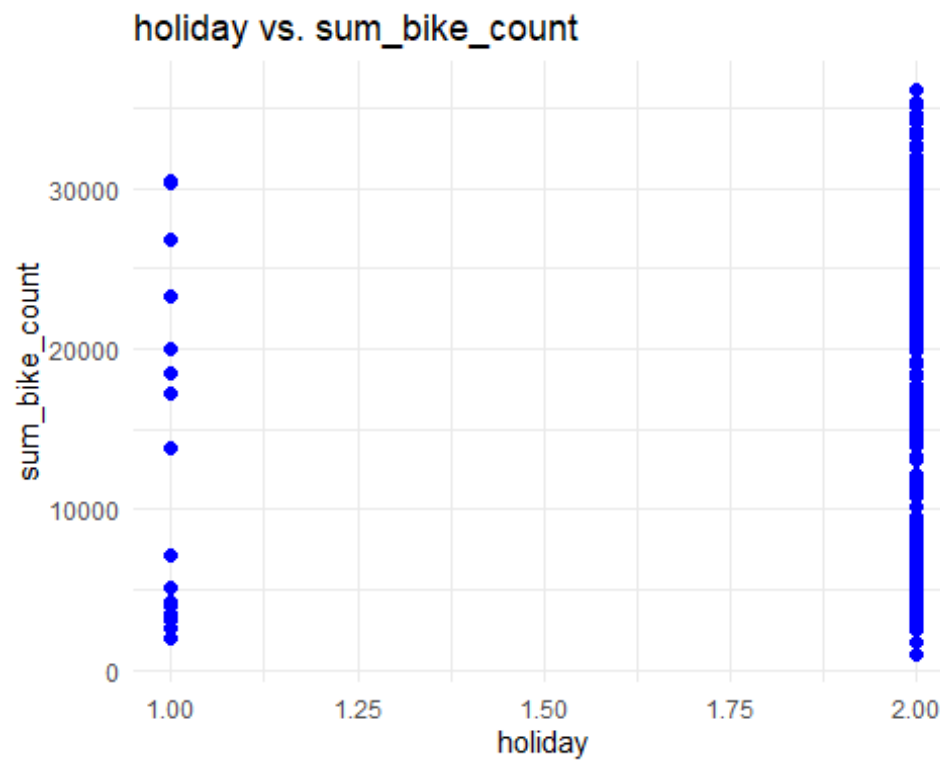
```
##  
## $seasons
```



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



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

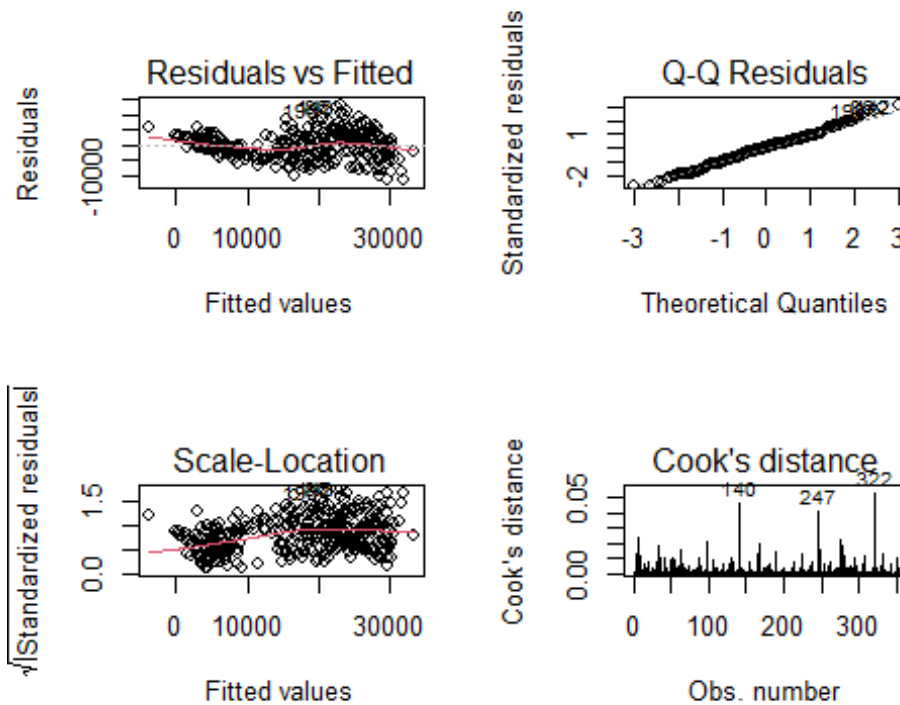


6.2.2 Xây dựng mô hình cơ bản.

```
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_temperature  -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){
```

```

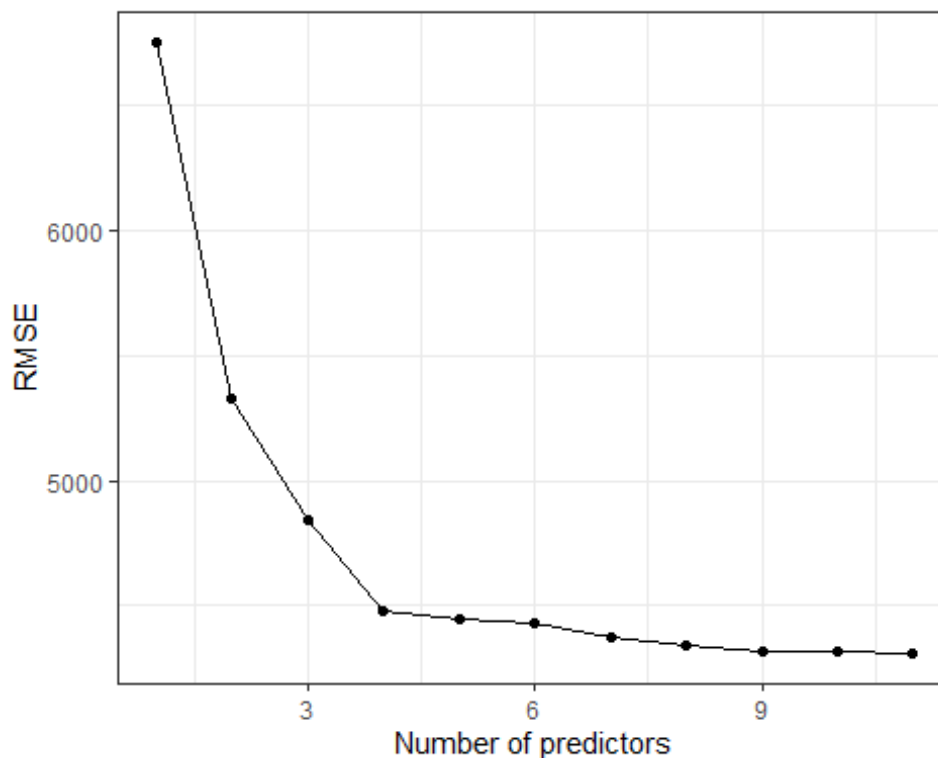
data_model2_train_r <- data_model2[folds != r, ]
data_model2_test_r <- data_model2[folds == r, ]

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.

```

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

reg_summary <- summary(regsubset)

# Tiêu chí Mallows 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 Mallows Cp:", best_model_cp,
"\n")

## Số lượng biến tốt nhất dựa trên tiêu chí Mallows 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 Mallows Cp
best_features_cp <- names(coef(regsubset, best_model_cp))
cat("Các biến tốt nhất dựa trên tiêu chí Mallows Cp:", best_features_cp,
"\n")

## Các biến tốt nhất dựa trên tiêu chí Mallows Cp: (Intercept) mean_humidity
mean_wind_speed mean_temperature 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_temperature 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_temperature mean_dew_point_temp mean_solar_radiation mean_rainfall seasons
day_of_week holiday

```

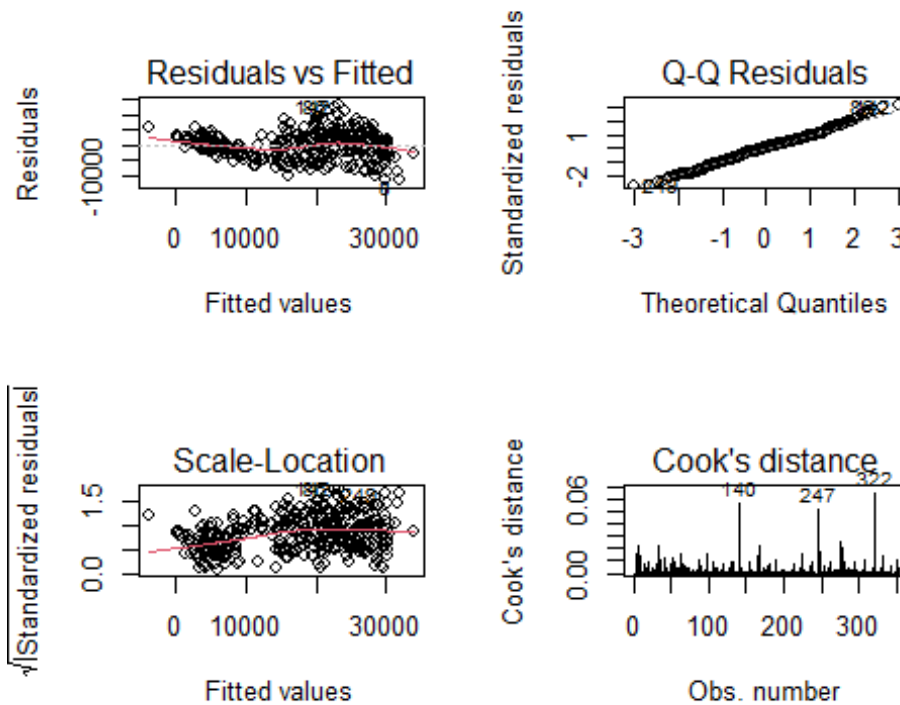
6.2.4 Building model

```
# 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

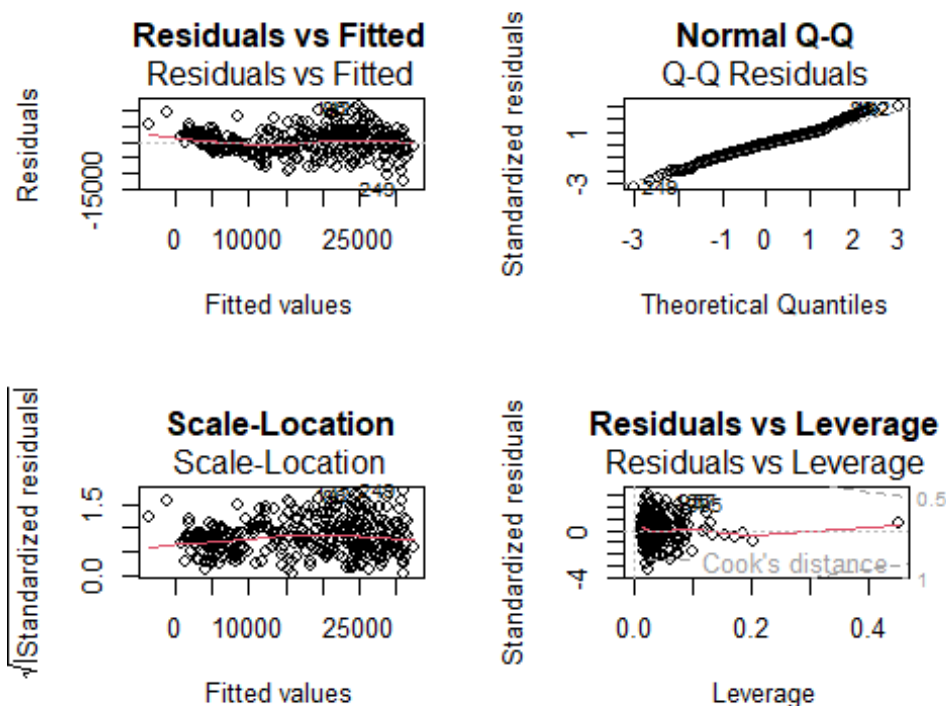
# Residuals vs Fitted
plot(model_2_expand, which = 1, main = "Residuals vs Fitted")

# 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 tuyến 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.