blog-3

Waheeb Algabri

Introduction

In this blog, we delve into the realm of predictive analytics by constructing regression models using bike sharing data. Through a series of steps, we will navigate the process of model creation, interpretation, and prediction. The dataset under scrutiny, Bike-Sharing-Dataset.zip, comprises two key files: hour.csv and day.csv. For the purpose of this analysis, we will exclusively employ day.csv. This dataset encapsulates various attributes, but for our models, we'll primarily focus on 'dteday', 'temp', and 'cnt' columns. The 'cnt' column serves as our dependent variable, whereas 'temp' and 'month_name', derived from 'dteday', will act as independent variables in our regression models. Leveraging R/RStudio, we will conduct data summary, preparation, and regression modeling tasks, with a keen eye on model diagnostics and interpretation.

Name	Definition
instant	Record index
dteday	Date
season	Season (1:spring, 2:summer, 3:fall, 4:winter)
yr	Year (0: 2011, 1:2012)
mnth	Month
holiday	Whether it is a holiday or not (1: yes, 0: no)
weekday	Day of the week
workingday	Whether it is a working day or not (1: yes, 0: no)
weathersit	Weather situation
temp	Normalized temperature in Celsius
atemp	Normalized feeling temperature in Celsius
hum	Normalized humidity
windspeed	Normalized wind speed
casual	Count of casual users
registered	Count of registered users
cnt	Total count of users (casual + registered)
month_name	Name of the month

Research Question

How do temperature and month impact bike sharing counts, and how effectively can these factors be modeled using regression analysis? Specifically, we aim to address the following:

- How does bike sharing count vary with different months of the year?
- What is the relationship between temperature and bike sharing count?
- How does the inclusion of temperature alongside month affect the predictive power of the regression model compared to a model solely based on month?

Simple and Multiple linear regression

```
# Load necessary libraries
library(lubridate)
library(dplyr)
library(corrplot)
```

```
# Load data
day <- read.csv("day.csv")
```

Loading Necessary Libraries and Data

```
str(day)
```

Data Wrangling

```
## 'data.frame':    731 obs. of  16 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : chr  "2011-01-01" "2011-01-02" "2011-01-03" "2011-01-04" ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

```
summary(day)
```

```
##     instant         dteday              season           yr
##  Min.   :  1.0   Length:731         Min.   :1.000   Min.   :0.0000
##  1st Qu.:183.5   Class :character   1st Qu.:2.000   1st Qu.:0.0000
##  Median :366.0   Mode  :character   Median :3.000   Median :1.0000
##  Mean   :366.0                      Mean   :2.497   Mean   :0.5007
##  3rd Qu.:548.5                      3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :731.0                      Max.   :4.000   Max.   :1.0000
##       mnth          holiday           weekday        workingday
##  Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000
```

```
##  1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000
##  Median : 7.00   Median :0.00000   Median :3.000   Median :1.000
##  Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684
##  3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000
##  Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000
##    weathersit         temp             atemp              hum
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725
##    windspeed          casual        registered        cnt
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
```

```r
# Check for missing values
sum(is.na(day))
```

```
## [1] 0
```

The result is Zero which means no missing values.

```r
# calculates the sum of values across each row for columns 7 through 13 in the dataframe day.

# rowSums(day[,7:13])
```

```r
# calculates the total sum of all the values across each row for columns 7 through 13 in the dataframe
sum(rowSums(day[,7:13]))
```

```
## [1] 5018.115
```

```r
n_distinct(day$site_name)
```

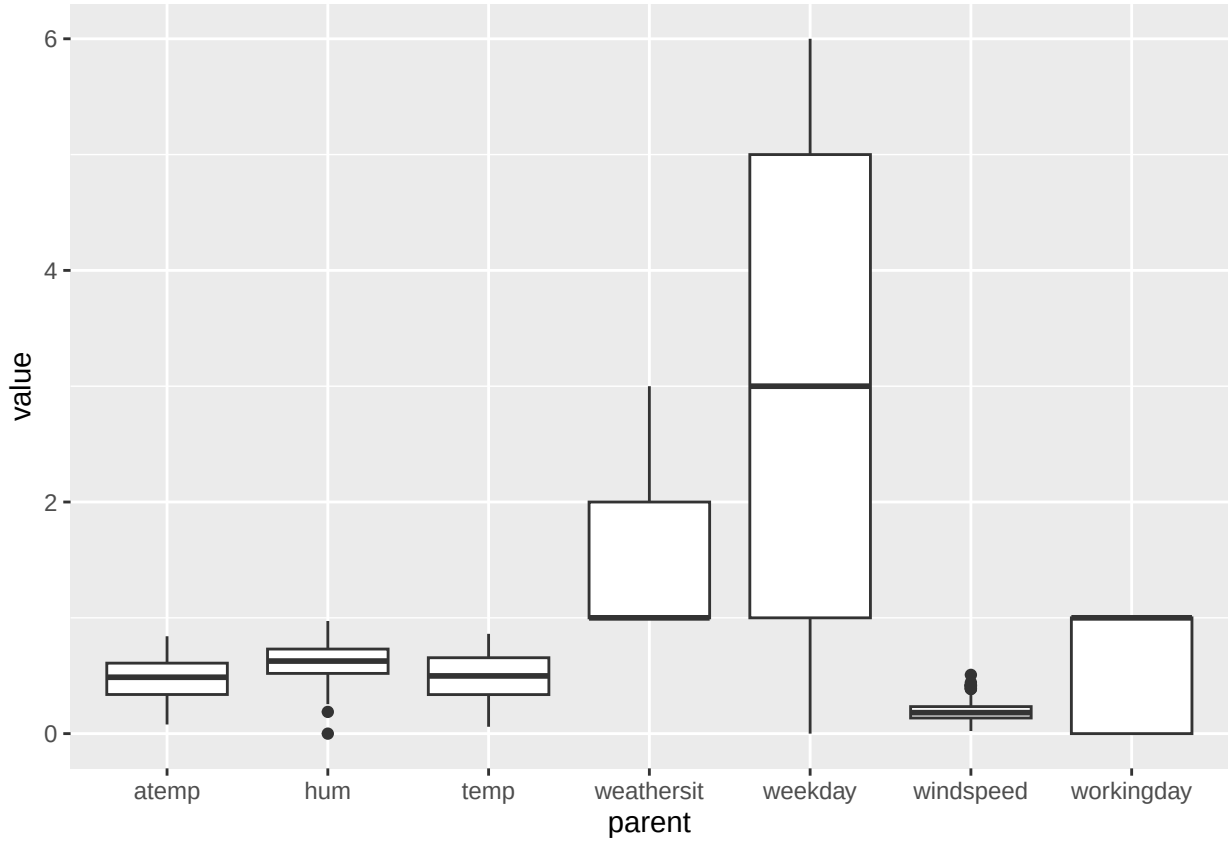
```
## [1] 0
```

To understand the unique values within each column of the day dataframe, you can use the distinct() function from the dplyr package or the unique() function in base

```r
# Display unique values
n_distinct(day$instant)
```

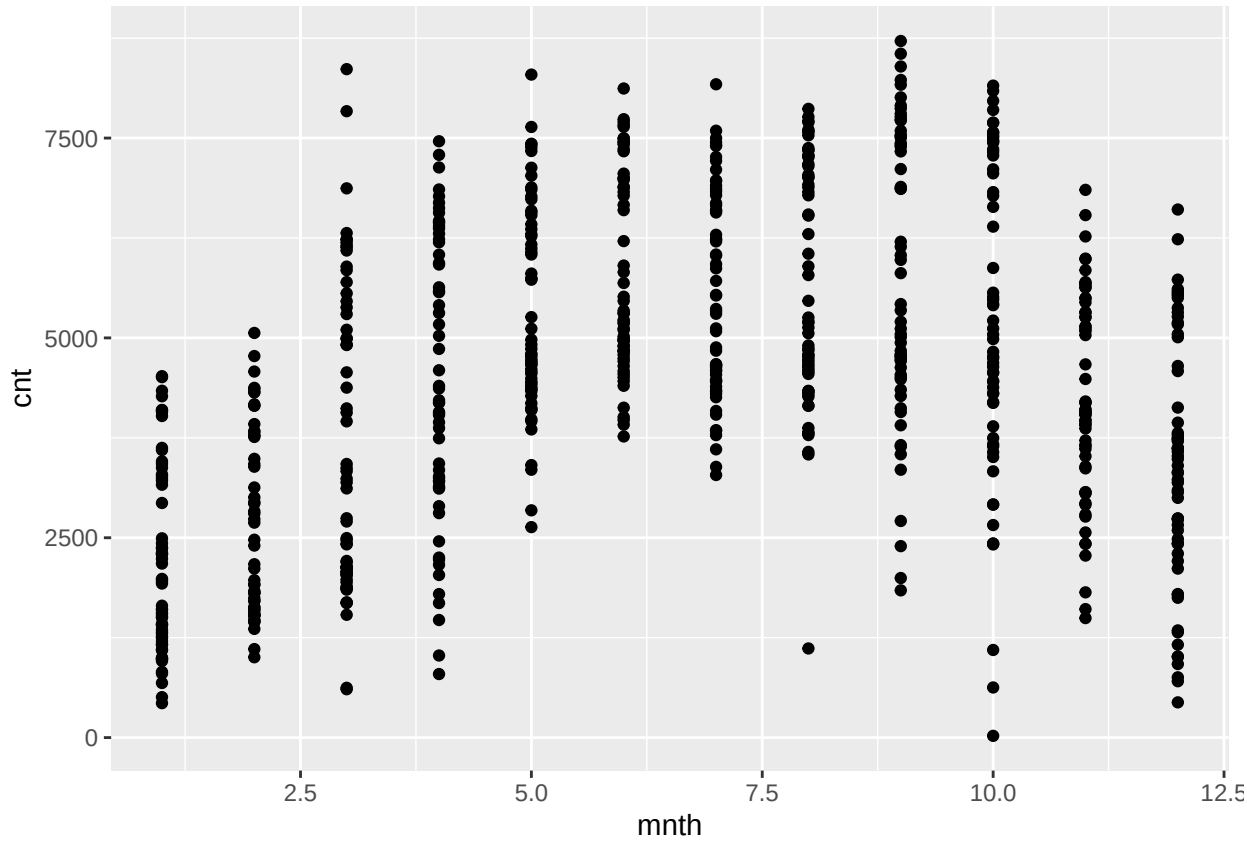
```
## [1] 731
```

```r
day %>%
  pivot_longer(cols = 7:13,
               names_to = "parent") %>%
  mutate(parent = abbreviate(parent, 10)) %>%
  ggplot(aes(x = parent, y = value)) +
  geom_boxplot()
```



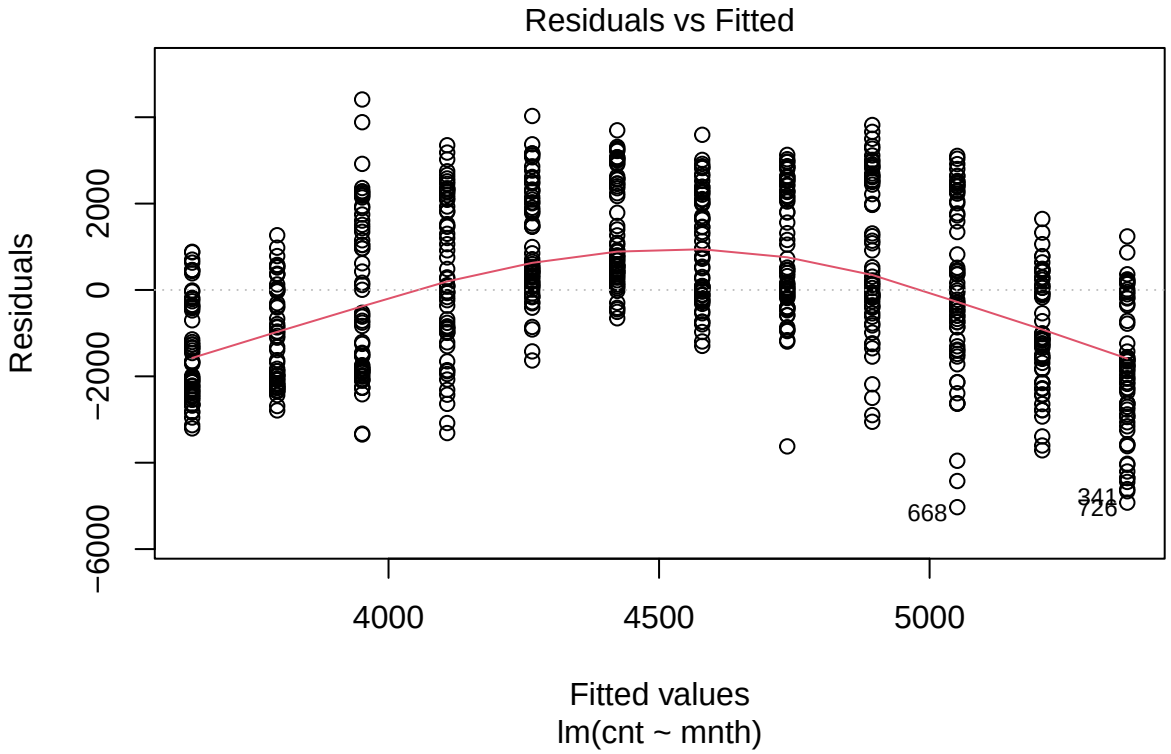

```r
# Model 1: Simple Regression
Model1 <- lm(cnt ~ mnth, data = day)
```

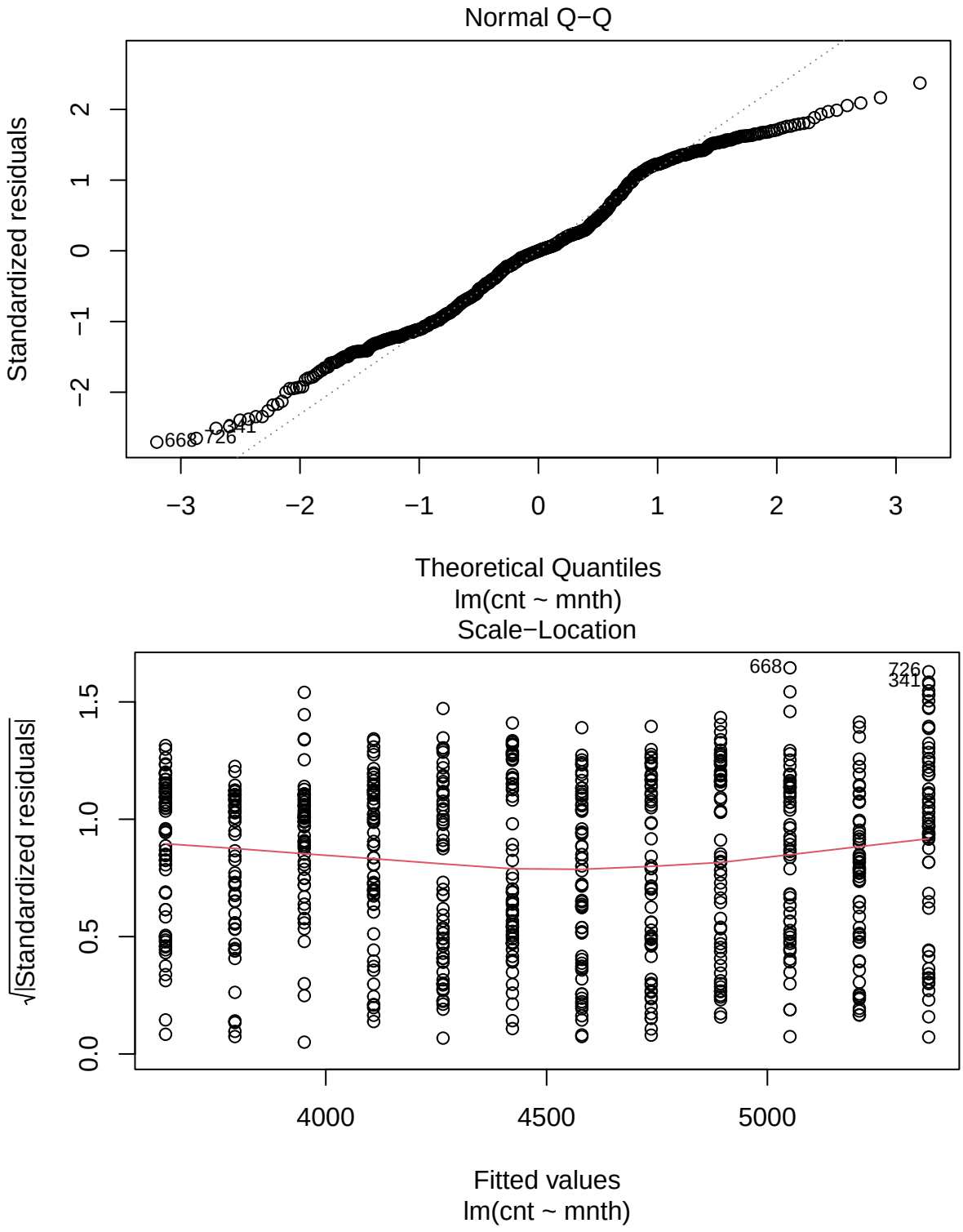
```r
ggplot(day, aes(x= mnth, y = cnt)) +
  geom_point()
```

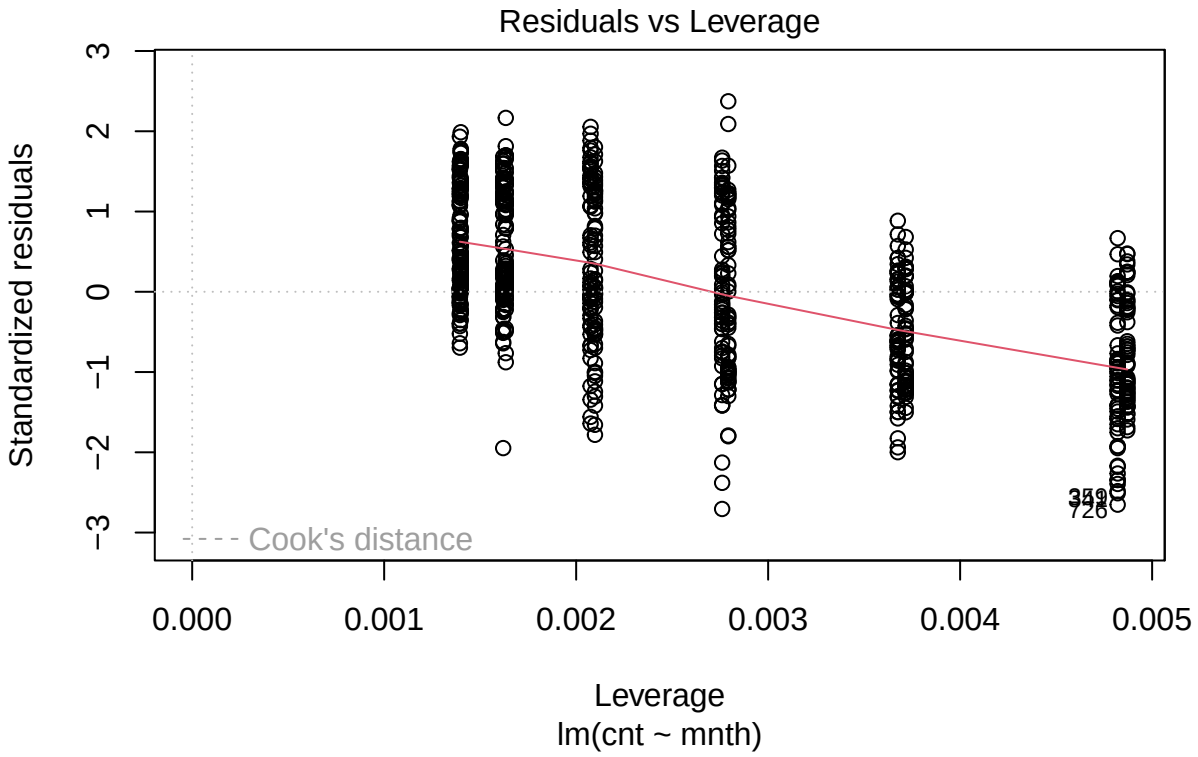


Model 1

```
plot(Model1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cnt ~ mnth)
Scale−Location
√|Standardized residuals|
668
726
341
Fitted values
lm(cnt ~ mnth)




```r
cor(day[,c("cnt", "mnth")])
```

```
##            cnt      mnth
## cnt  1.0000000 0.2799771
## mnth 0.2799771 1.0000000
```

The correlation coefficient between cnt and mnth is approximately 0.28. This positive correlation suggests that there is a weak positive linear relationship between the total count of users and the month variable.

```r
cor(day[,c("cnt", "mnth","temp")])
```

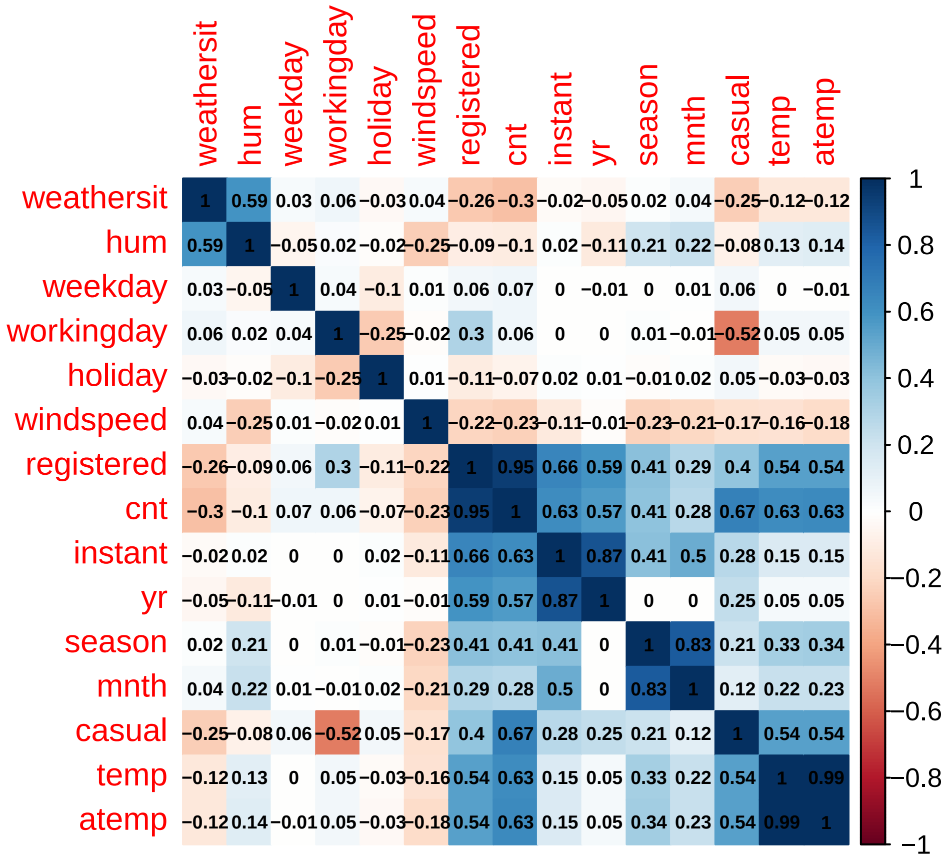
```
##            cnt      mnth      temp
## cnt  1.0000000 0.2799771 0.6274940
## mnth 0.2799771 1.0000000 0.2202053
## temp 0.6274940 0.2202053 1.0000000
```

```r
ctrd <- cor(day %>% select(where(is.numeric)))
print(ctrd)
```

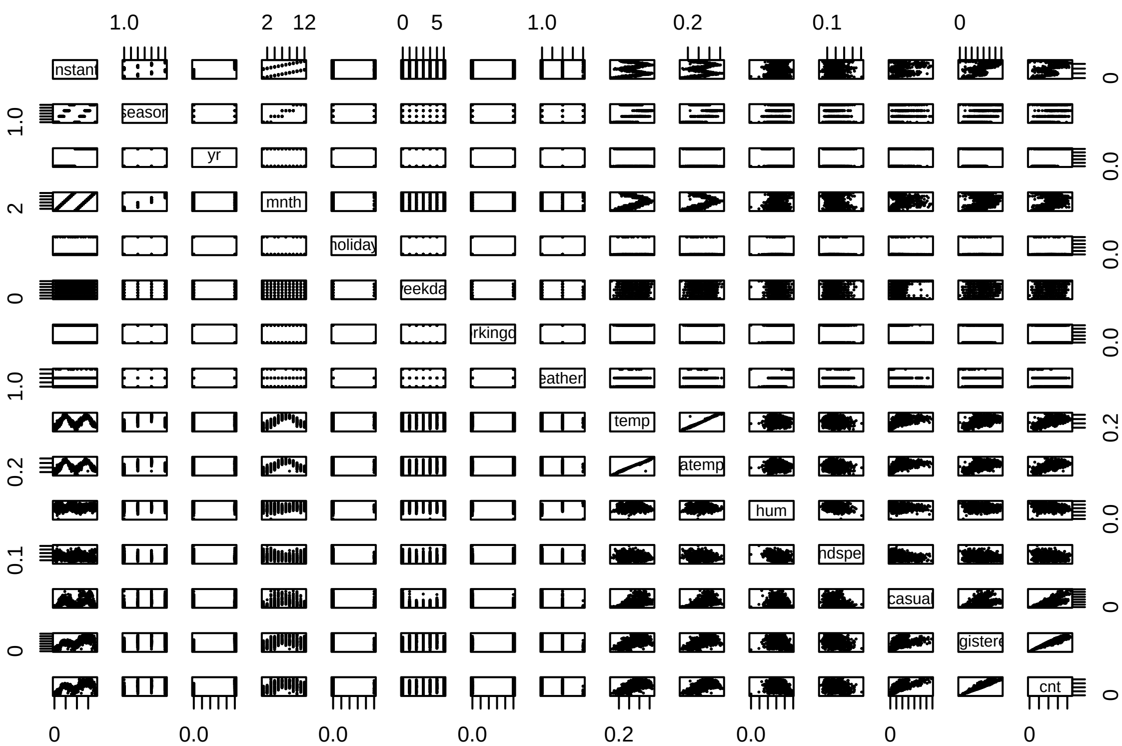
```
##                  instant       season           yr         mnth      holiday
## instant     1.000000e+00  0.412224179  0.866025404  0.496701889  0.016144632
## season      4.122242e-01  1.000000000 -0.001844343  0.831440114 -0.010536659
## yr          8.660254e-01 -0.001844343  1.000000000 -0.001792434  0.007954311
## mnth        4.967019e-01  0.831440114 -0.001792434  1.000000000  0.019190895
## holiday     1.614463e-02 -0.010536659  0.007954311  0.019190895  1.000000000
## weekday    -1.617914e-05 -0.003079881 -0.005460765  0.009509313 -0.101960269
## workingday -4.336537e-03  0.012484963 -0.002012621 -0.005900951 -0.253022700
```

```
## weathersit -2.147721e-02  0.019211028 -0.048726541  0.043528098 -0.034626841
## temp        1.505803e-01  0.334314856  0.047603572  0.220205335 -0.028555535
## atemp       1.526382e-01  0.342875613  0.046106149  0.227458630 -0.032506692
## hum         1.637471e-02  0.205444765 -0.110651045  0.222203691 -0.015937479
## windspeed  -1.126196e-01 -0.229046337 -0.011817060 -0.207501752  0.006291507
## casual      2.752552e-01  0.210399165  0.248545664  0.123005889  0.054274203
## registered  6.596229e-01  0.411623051  0.594248168  0.293487830 -0.108744863
## cnt         6.288303e-01  0.406100371  0.566709708  0.279977112 -0.068347716
##                  weekday   workingday  weathersit          temp        atemp
## instant    -1.617914e-05 -0.004336537 -0.02147721  0.1505803019  0.152638238
## season     -3.079881e-03  0.012484963  0.01921103  0.3343148564  0.342875613
## yr         -5.460765e-03 -0.002012621 -0.04872654  0.0476035719  0.046106149
## mnth        9.509313e-03 -0.005900951  0.04352810  0.2202053352  0.227458630
## holiday    -1.019603e-01 -0.253022700 -0.03462684 -0.0285555350 -0.032506692
## weekday     1.000000e+00  0.035789674  0.03108747 -0.0001699624 -0.007537132
## workingday  3.578967e-02  1.000000000  0.06120043  0.0526598102  0.052182275
## weathersit  3.108747e-02  0.061200430  1.00000000 -0.1206022365 -0.121583354
## temp       -1.699624e-04  0.052659810 -0.12060224  1.0000000000  0.991701553
## atemp      -7.537132e-03  0.052182275 -0.12158335  0.9917015532  1.000000000
## hum        -5.223210e-02  0.024327046  0.59104460  0.1269629390  0.139988060
## windspeed   1.428212e-02 -0.018796487  0.03951106 -0.1579441204 -0.183642967
## casual      5.992264e-02 -0.518044191 -0.24735300  0.5432846617  0.543863690
## registered  5.736744e-02  0.303907117 -0.26038771  0.5400119662  0.544191758
## cnt         6.744341e-02  0.061156063 -0.29739124  0.6274940090  0.631065700
##                    hum    windspeed      casual  registered         cnt
## instant     0.01637471 -0.112619556  0.27525521  0.65962287  0.62883027
## season      0.20544476 -0.229046337  0.21039916  0.41162305  0.40610037
## yr         -0.11065104 -0.011817060  0.24854566  0.59424817  0.56670971
## mnth        0.22220369 -0.207501752  0.12300589  0.29348783  0.27997711
## holiday    -0.01593748  0.006291507  0.05427420 -0.10874486 -0.06834772
## weekday    -0.05223210  0.014282124  0.05992264  0.05736744  0.06744341
## workingday  0.02432705 -0.018796487 -0.51804419  0.30390712  0.06115606
## weathersit  0.59104460  0.039511059 -0.24735300 -0.26038771 -0.29739124
## temp        0.12696294 -0.157944120  0.54328466  0.54001197  0.62749401
## atemp       0.13998806 -0.183642967  0.54386369  0.54419176  0.63106570
## hum         1.00000000 -0.248489099 -0.07700788 -0.09108860 -0.10065856
## windspeed  -0.24848910  1.000000000 -0.16761335 -0.21744898 -0.23454500
## casual     -0.07700788 -0.167613349  1.00000000  0.39528245  0.67280443
## registered -0.09108860 -0.217448981  0.39528245  1.00000000  0.94551692
## cnt        -0.10065856 -0.234544997  0.67280443  0.94551692  1.00000000
```

```r
# Plot the correlation matrix using corrplot with the specified parameters
corrplot(ctrd, method = "color", order = "hclust", addCoef.col = "black", number.cex = 0.6)
```




```r
# Plot pairwise scatterplots for all numeric variables in the dataset
pairs(day %>% select(where(is.numeric)), cex = 0.1)
```



```r
summary(Model1)
```

```
##
## Call:
## lm(formula = cnt ~ mnth, data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5029.2 -1440.4   -10.2  1463.4  4410.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3479.93     147.18  23.644  < 2e-16 ***
## mnth          157.12      19.95   7.874 1.24e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1861 on 729 degrees of freedom
## Multiple R-squared:  0.07839,    Adjusted R-squared:  0.07712
## F-statistic:    62 on 1 and 729 DF,  p-value: 1.243e-14
```

The R-squared value of 0.07839 suggests that only around 7.84% of the variation in bike sharing counts can be explained by the month variable alone. This means that a large portion of the variability in bike sharing counts remains unexplained by the month variable in Model1. It's possible that other factors not included in the model are influencing bike sharing counts.

Reference Month:

To identify the reference month, we need to look at the coefficient estimates. The reference month is the one with a coefficient estimate of 0, as it serves as the baseline for comparison. Let's extract the coefficient estimates:

```r
# Coefficient estimates for Model1
coef(Model1)
```

```
## (Intercept)        mnth
##    3479.933     157.123
```

From the output, we can see that the intercept corresponds to the reference month. Therefore, the reference month is the month with an intercept coefficient. We'll report the predicted count for this month below.

Predicted Count for January and June:

To obtain the predicted count for January and June, we can use the coefficient estimates from Model1. Since January is the reference month, its coefficient estimate directly represents the predicted count for January. Similarly, we can extract the coefficient estimate for June and compute the predicted count.

```r
# Extracting the coefficients from Model1
intercept <- coef(Model1)[1]
coefficient_mnth <- coef(Model1)[2]

# Predicted count for January
january_pred <- intercept + coefficient_mnth
```

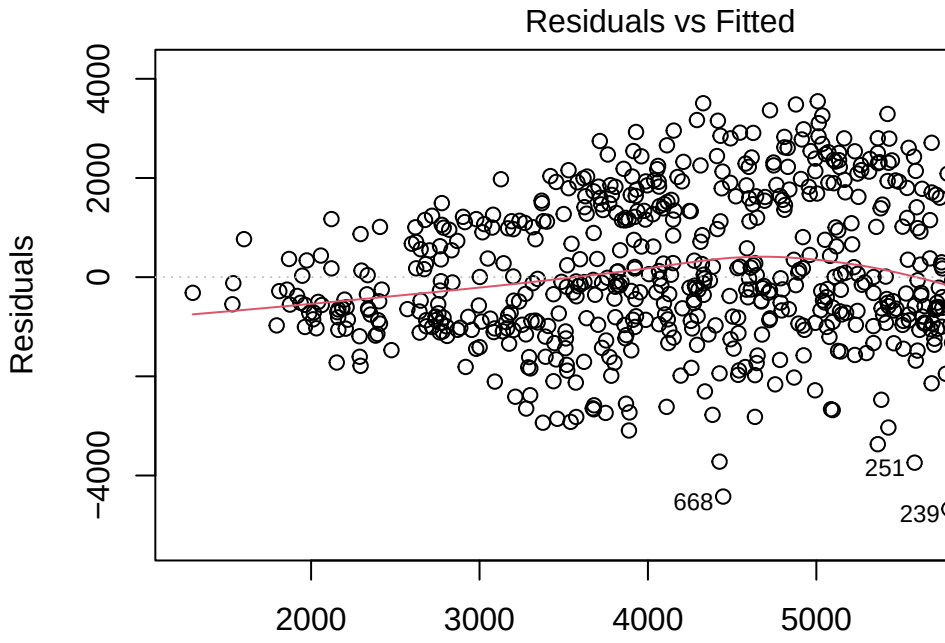
```r
# Predicted count for June (6 months ahead of January)
june_pred <- january_pred + (coefficient_mnth * 6)

# Print the predicted counts
cat("Predicted count for January:", january_pred, "\n")
```

```
## Predicted count for January: 3637.056
```

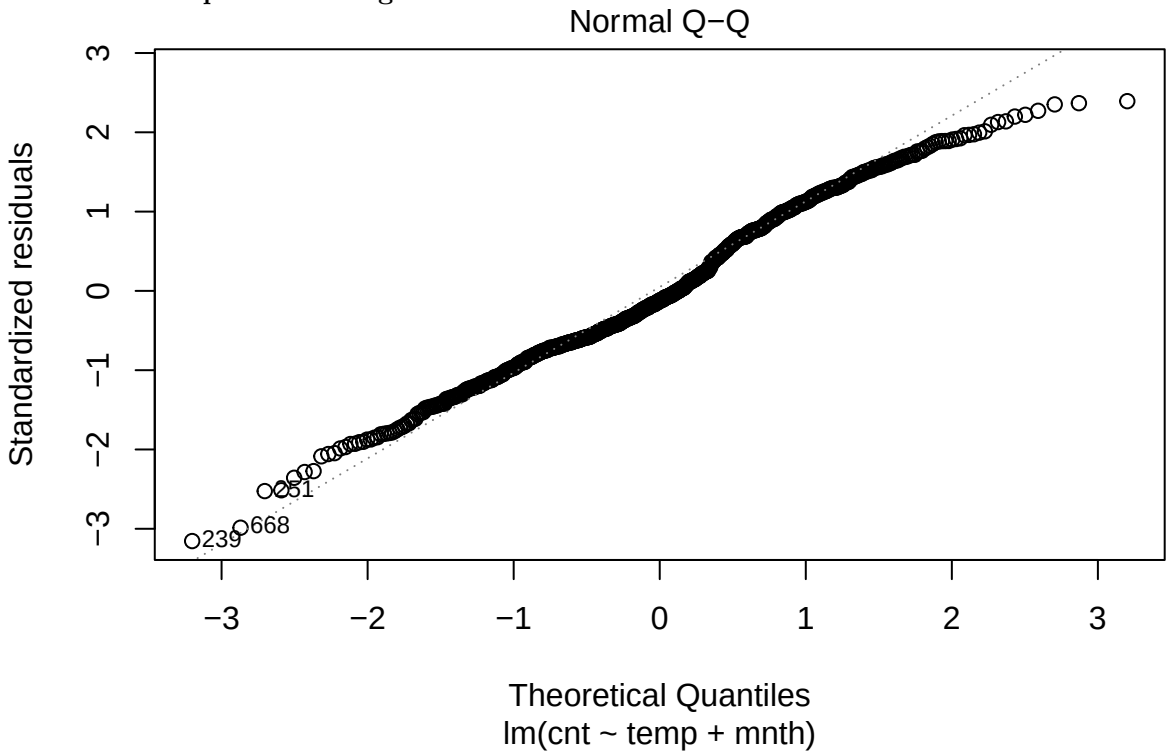
```r
cat("Predicted count for June:", june_pred, "\n")
```

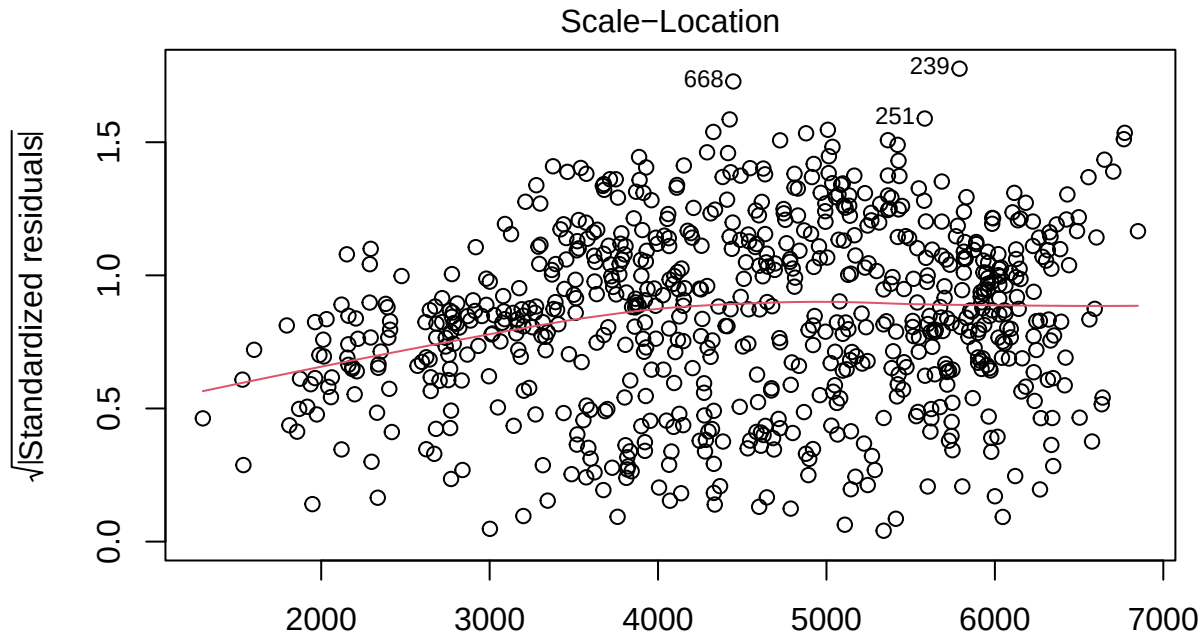
```
## Predicted count for June: 4579.794
```

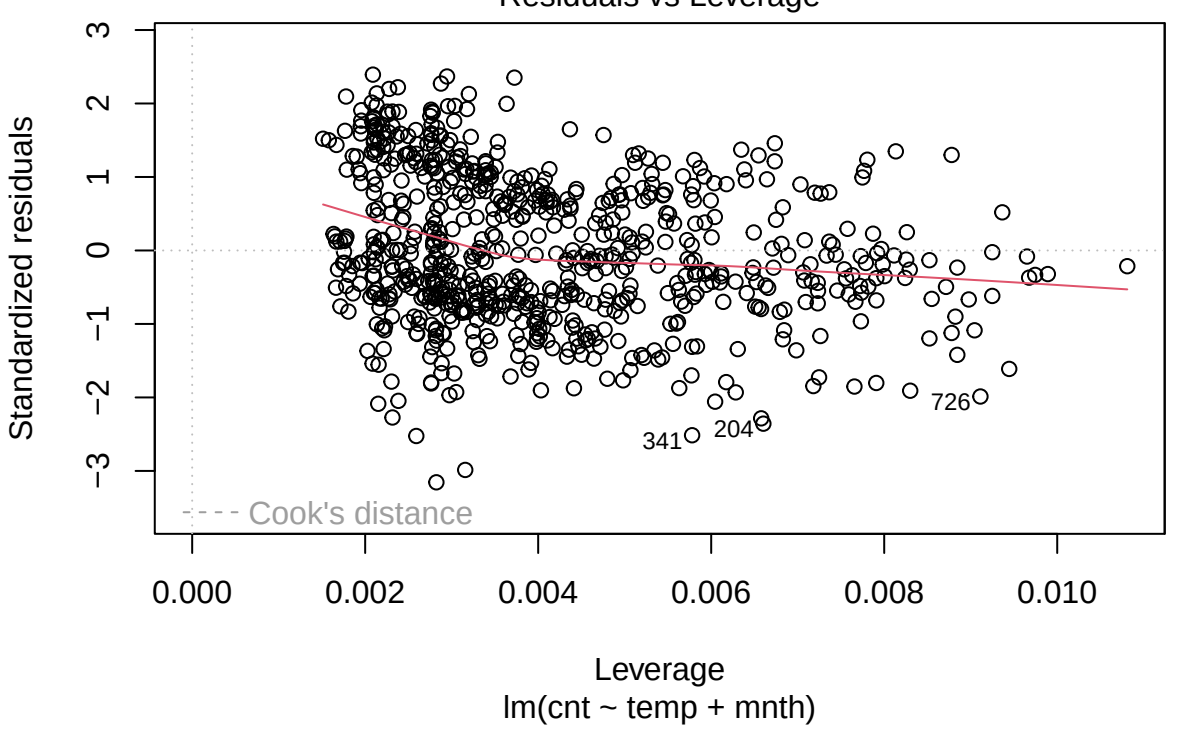
```r


# Model 2: Multiple Linear Regression
Model2 <- lm(cnt ~ temp + mnth, data = day)
plot(Model2)
```



Model2: Multiple Linear Regression








```
# Summary for Model2
summary(Model2)
```

```
##
## Call:
## lm(formula = cnt ~ temp + mnth, data = day)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4675.0 -1005.4  -183.2  1151.9  3546.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   841.41     174.36   4.826 1.70e-06 ***
## temp         6293.42     307.58  20.461  < 2e-16 ***
## mnth           83.63      16.31   5.128 3.77e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1484 on 728 degrees of freedom
## Multiple R-squared:  0.4149, Adjusted R-squared:  0.4133
## F-statistic: 258.1 on 2 and 728 DF,  p-value: < 2.2e-16
```

The R-squared value for Model2 is higher than that of Model1, indicating that Model2, which includes both temperature and month as independent variables, explains more variability in the dependent variable (bike sharing counts) compared to Model1, which only included the month variable. This improvement in the R-squared value suggests that temperature is an important predictor of bike sharing counts and contributes significantly to the model's predictive power, in addition to the month variable.

Comparison of Coefficient Estimates:

We'll compare the coefficient estimates for the month_nameJan variable between Model1 and Model2.

```r
# Coefficient estimates for Model2
coef(Model2)
```

```
## (Intercept)        temp        mnth
##   841.41012  6293.41819    83.63325
```

In Model1 (where only the month variable was used as a predictor), the coefficient estimate for the month variable was 157.12. In Model2 (where both temperature and month were used as predictors), the coefficient estimate for the month variable reduced to 83.63.

Predicted Count for January with Temperature 0.25

To predict the count for January when the temperature is 0.25, we can use the coefficient estimates from Model2.

```r
# Predicted count for January with temp = 0.25
january_pred_temp <- coef(Model2)["(Intercept)"] + coef(Model2)["temp"] * 0.25
cat("Predicted count for January with temp = 0.25:", january_pred_temp)
```

```
## Predicted count for January with temp = 0.25: 2414.765
```

```r
# Coefficients from Model2
intercept <- 841.41012
temp_coef <- 6293.41819
mnth_coef <- 83.63325

# Temperature value for prediction
```

```r
temp_value <- 0.25

# Month value for January (since January is the reference month)
january <- 1

# Predicted count for January with temp = 0.25
predicted_count_january <- intercept + temp_coef * temp_value + mnth_coef * january
predicted_count_january
```

```
## [1] 2498.398
```

Conclusion

the analysis of bike sharing data using regression models reveals insightful relationships between various factors and bike sharing counts. While the month variable alone provides limited predictive power, the inclusion of temperature significantly enhances the model's ability to explain variations in bike sharing counts. Higher temperatures are associated with increased bike sharing activity, underscoring the influence of weather on user behavior.

Comparing the models highlights the importance of considering both temporal and weather-related variables. Model 2, which incorporates both temperature and month, outperforms Model 1, emphasizing the significance of temperature in predicting bike sharing demand.

These findings offer valuable insights for bike sharing systems, enabling them to better anticipate and respond to demand fluctuations. By leveraging weather data alongside seasonal trends, bike sharing providers can optimize resource allocation, enhance service planning, and ultimately improve the overall user experience. This analysis underscores the importance of integrating weather considerations into predictive models to optimize bike sharing system operations effectively.