Examining Trends in BikeShare Data

Martin Garcia-Angel

2023-12-03

```
hour =read.csv("C:\\Users\\Marti\\Downloads\\hour.csv")
head(hour)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
##   temp  atemp  hum windspeed casual registered cnt
## 1 0.24 0.2879 0.81    0.0000      3         13  16
## 2 0.22 0.2727 0.80    0.0000      8         32  40
## 3 0.22 0.2727 0.80    0.0000      5         27  32
## 4 0.24 0.2879 0.75    0.0000      3         10  13
## 5 0.24 0.2879 0.75    0.0000      0          1   1
## 6 0.24 0.2576 0.75    0.0896      0          1   1
```

```
day =read.csv("C:\\Users\\Marti\\Downloads\\day.csv")
head(day)
```

```
##   instant     dteday season yr mnth holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1       0       6          0          2
## 2       2 2011-01-02      1  0    1       0       0          0          2
## 3       3 2011-01-03      1  0    1       0       1          1          1
## 4       4 2011-01-04      1  0    1       0       2          1          1
## 5       5 2011-01-05      1  0    1       0       3          1          1
## 6       6 2011-01-06      1  0    1       0       4          1          1
##       temp    atemp      hum windspeed casual registered  cnt
## 1 0.344167 0.363625 0.805833 0.1604460    331        654  985
## 2 0.363478 0.353739 0.696087 0.2485390    131        670  801
## 3 0.196364 0.189405 0.437273 0.2483090    120       1229 1349
## 4 0.200000 0.212122 0.590435 0.1602960    108       1454 1562
## 5 0.226957 0.229270 0.436957 0.1869000     82       1518 1600
## 6 0.204348 0.233209 0.518261 0.0895652     88       1518 1606
```

```
nrow(hour)
```

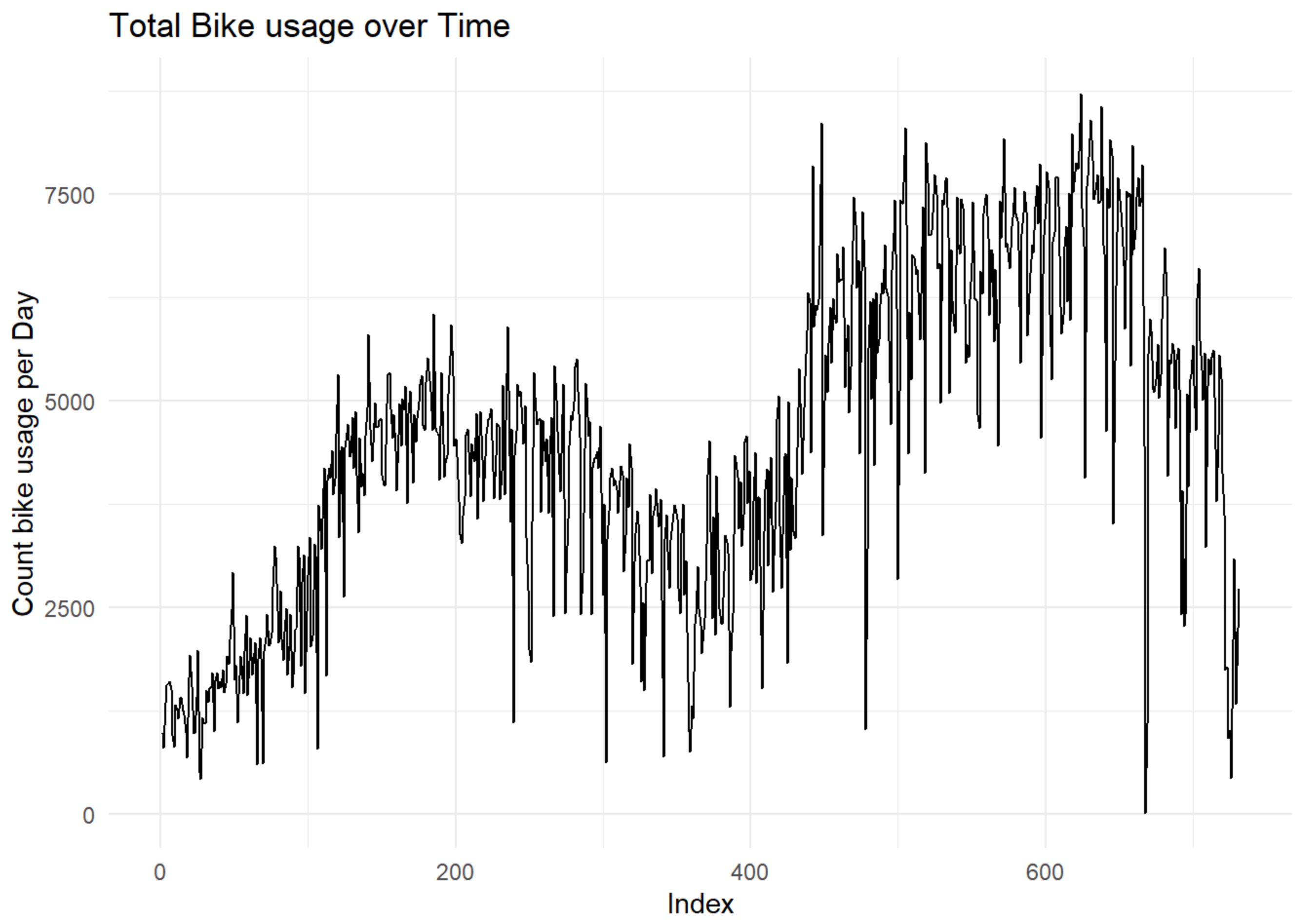
```
## [1] 17379
```

```
nrow(day)
```

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

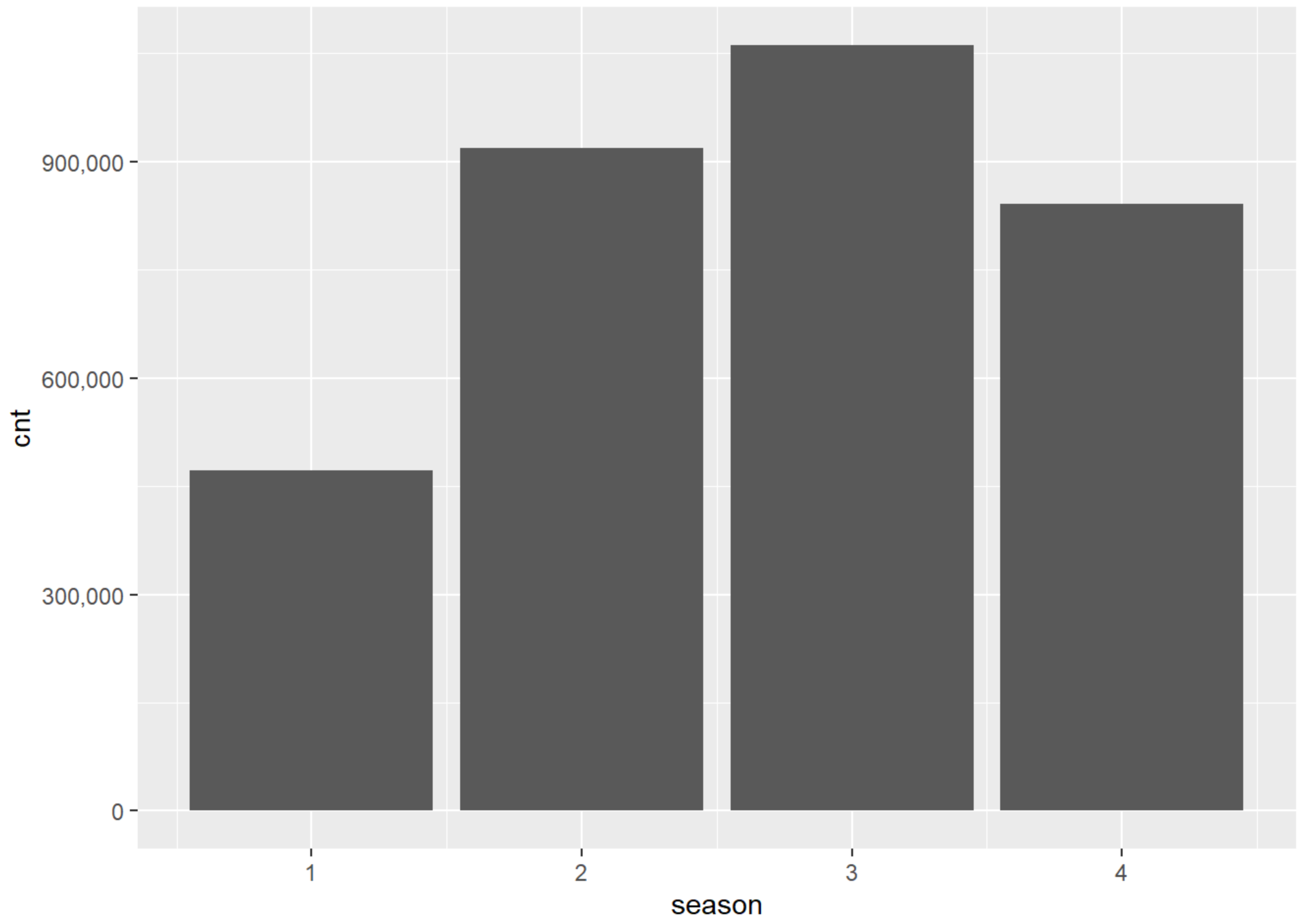
Line plot showing bike usage per day

```
# Create the line plot
ggplot(day, aes(x = instant, y = cnt)) +
  geom_line() +
  labs(x = "Index", y = "Count bike usage per Day", title = "Total Bike usage over Time") +
  theme_minimal()
```



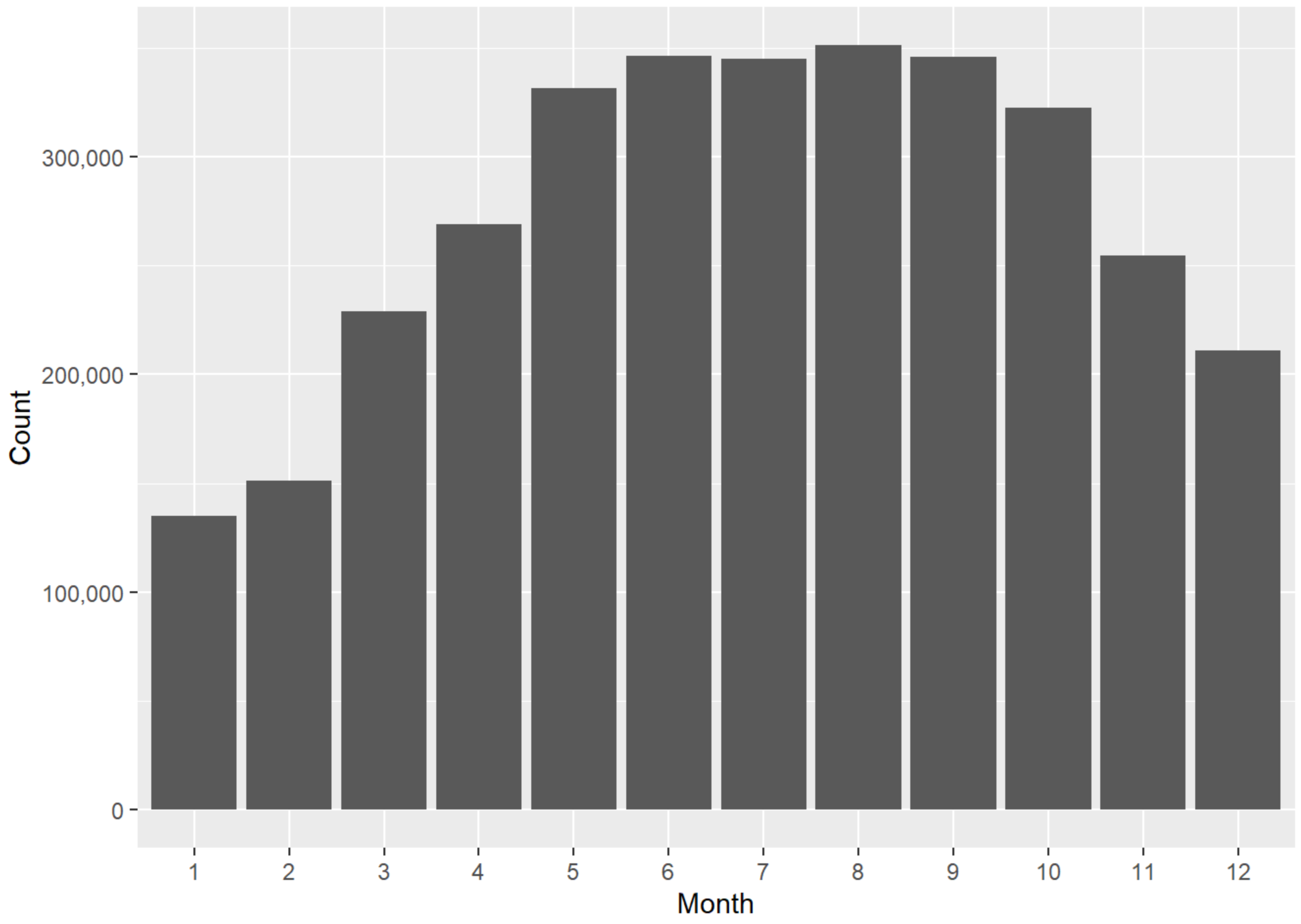
Checking total count of bikes per season

```
ggplot(day, aes(x = season, y = cnt)) + geom_bar(stat = 'identity')+ scale_y_continuous(labels =
scales::comma)
```



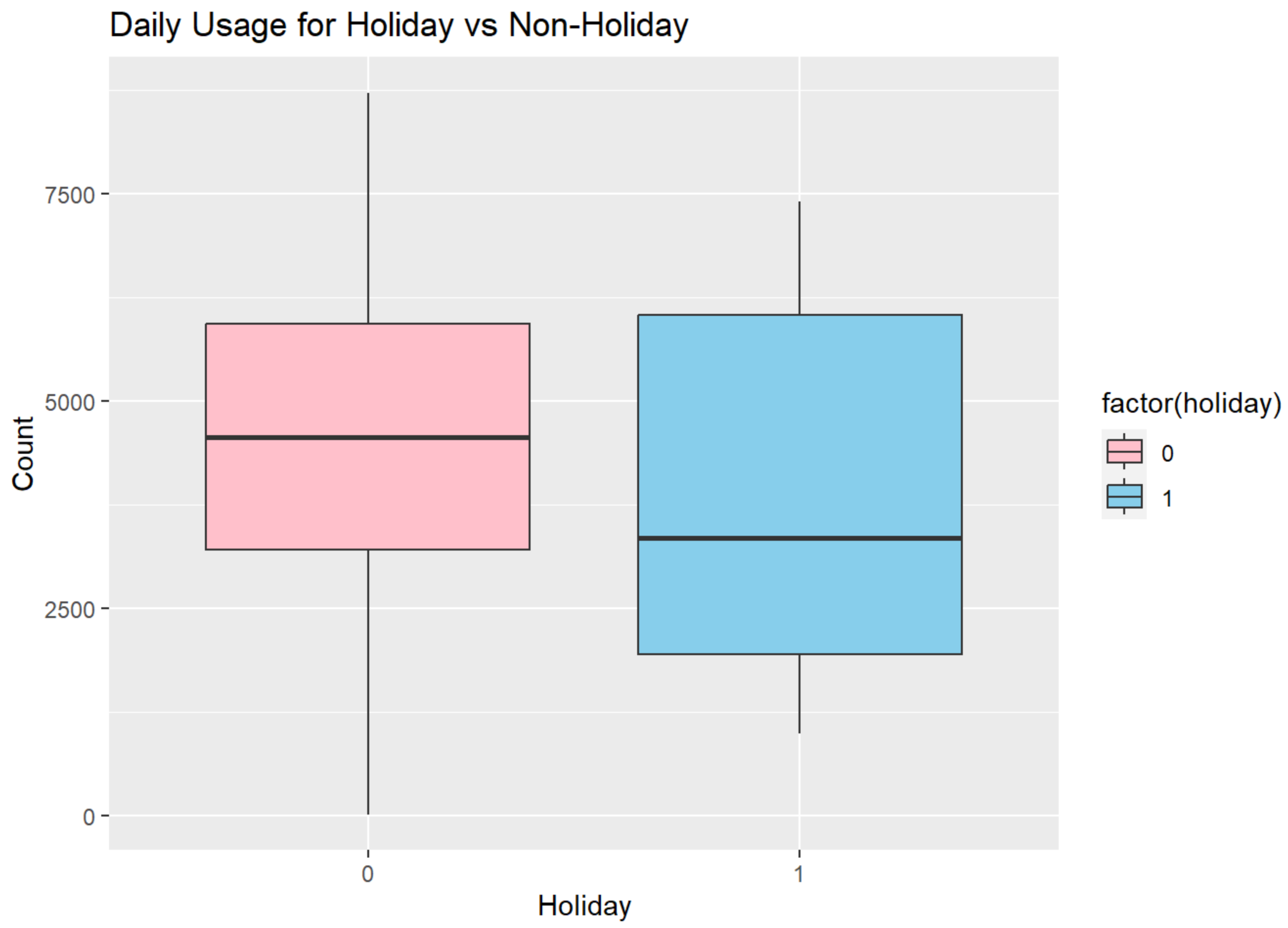
Checking total number of bikes per every month of the year

```
day$mnth <- as.factor(day$mnth)

ggplot(day, aes(x = factor(mnth), y = cnt)) +
  geom_bar(stat = 'identity') +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Month", y = "Count")
```



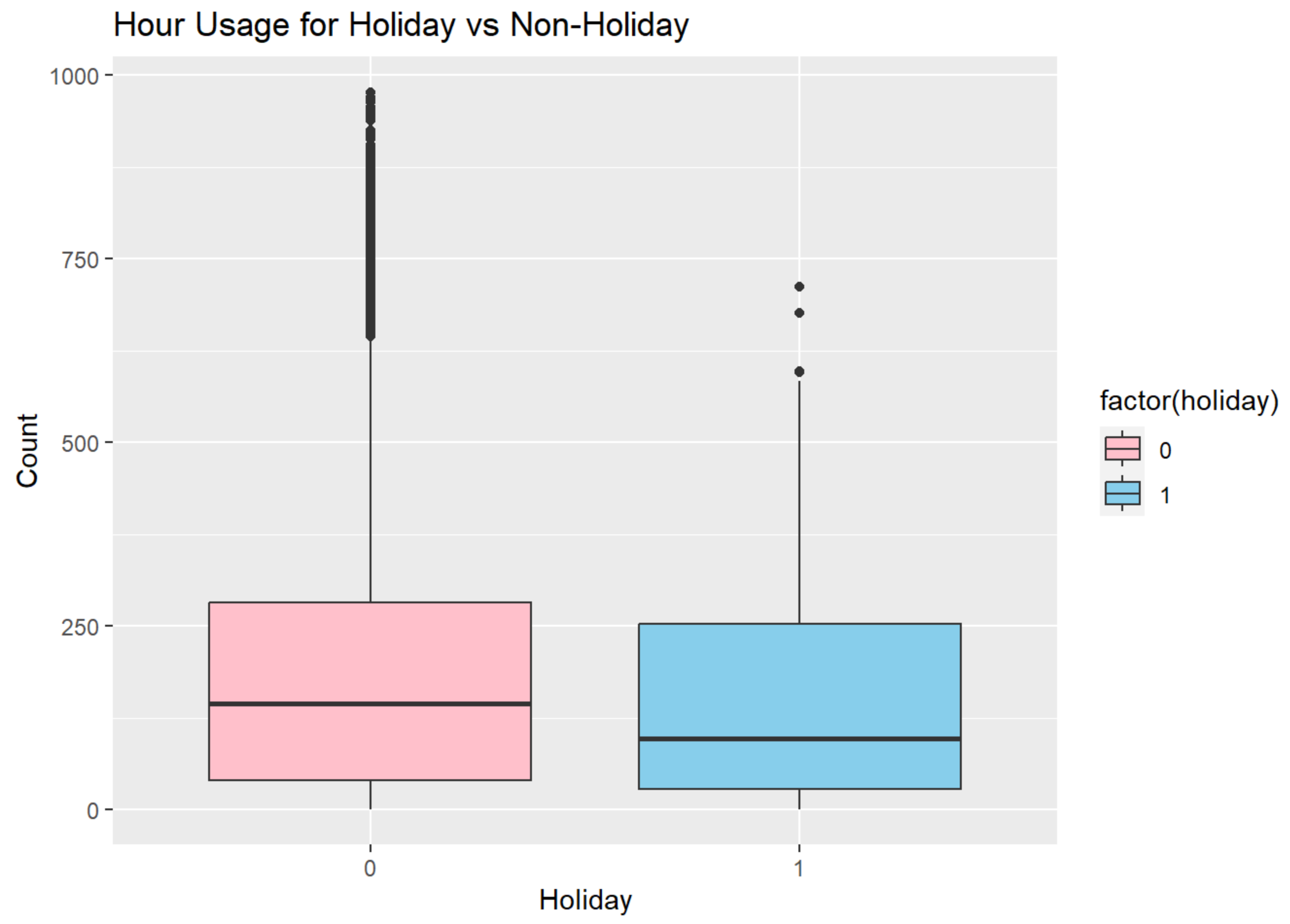
box plot of Daily usage during Holiday vs Non-Holiday

```
box_plot <- ggplot(day, aes(x = factor(holiday), y = cnt, fill = factor(holiday))) +
  geom_boxplot() +
  labs(x = "Holiday", y = "Count", title = "Daily Usage for Holiday vs Non-Holiday") +
  scale_fill_manual(values = c("pink", "skyblue"))  # Specify colors for boxes
box_plot
```



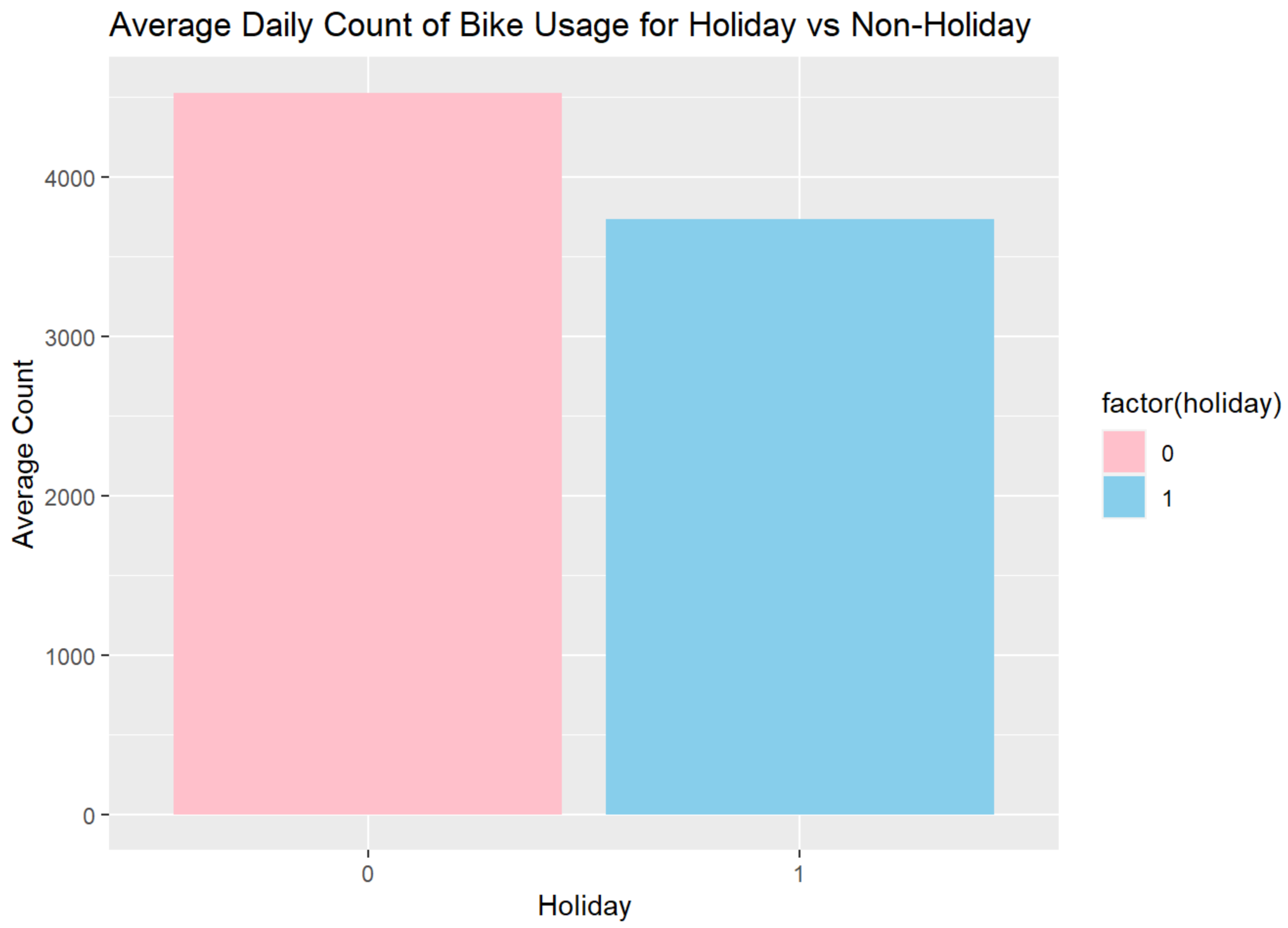
box plot of Hourly usage during Holiday vs Non-Holiday

```
box_plot <- ggplot(hour, aes(x = factor(holiday), y = cnt, fill = factor(holiday))) +
  geom_boxplot() +
  labs(x = "Holiday", y = "Count", title = "Hour Usage for Holiday vs Non-Holiday") +
  scale_fill_manual(values = c("pink", "skyblue"))  # Specify colors for boxes
box_plot
```

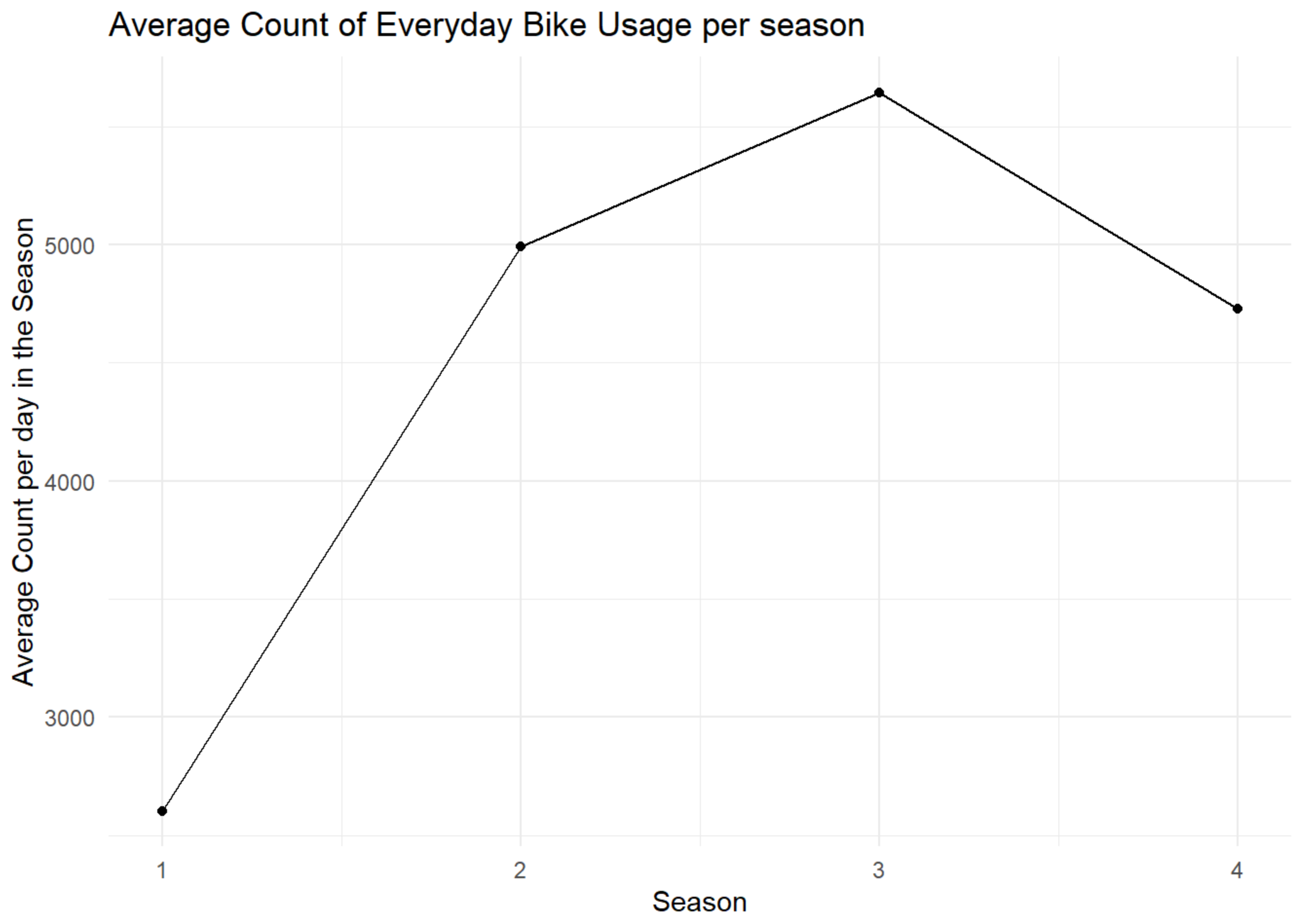


Bar plot of average ridership for holiday vs non-holiday

```
avg_counts <- day %>%
  group_by(holiday) %>%
  summarise(avg_count = mean(cnt, na.rm = TRUE))

bar_chart <- ggplot(avg_counts, aes(x = factor(holiday), y = avg_count, fill = factor(holiday)))
+
  geom_bar(stat = 'identity') +
  labs(x = "Holiday", y = "Average Count", title = "Average Daily Count of Bike Usage for Holida
y vs Non-Holiday") +
  scale_fill_manual(values = c("pink", "skyblue"))

bar_chart
```

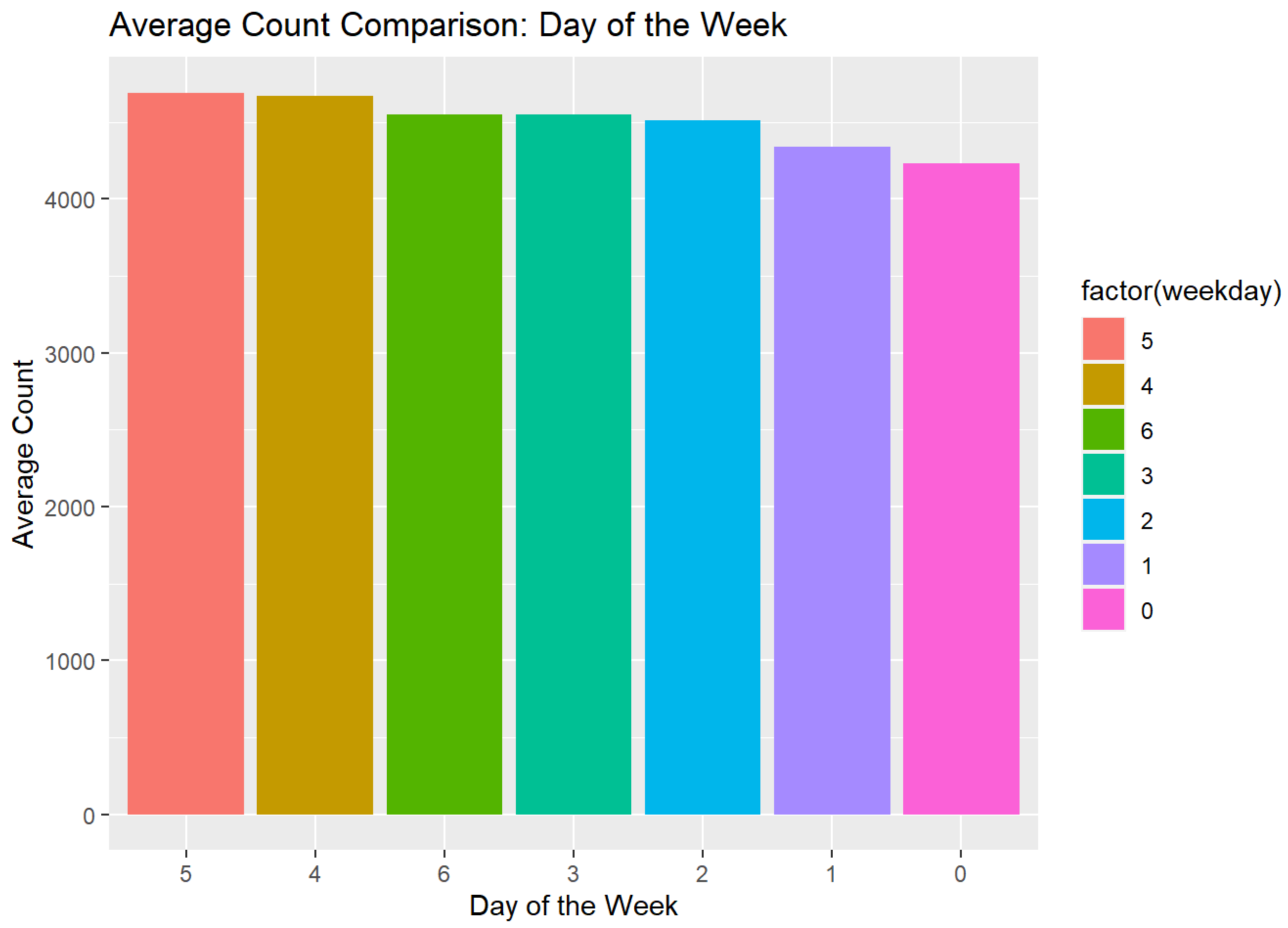


```
# Calculate average counts per hour
avg_counts <- day %>%
  group_by(season) %>%
  summarise(avg_count = mean(cnt, na.rm = TRUE))

# Create the line plot
ggplot(avg_counts, aes(x = season, y = avg_count)) +
  geom_point() +
  geom_line() +
  labs(x = "Season", y = "Average Count per day in the Season", title = "Average Count of Everyd
ay Bike Usage per season") +
  theme_minimal()
```



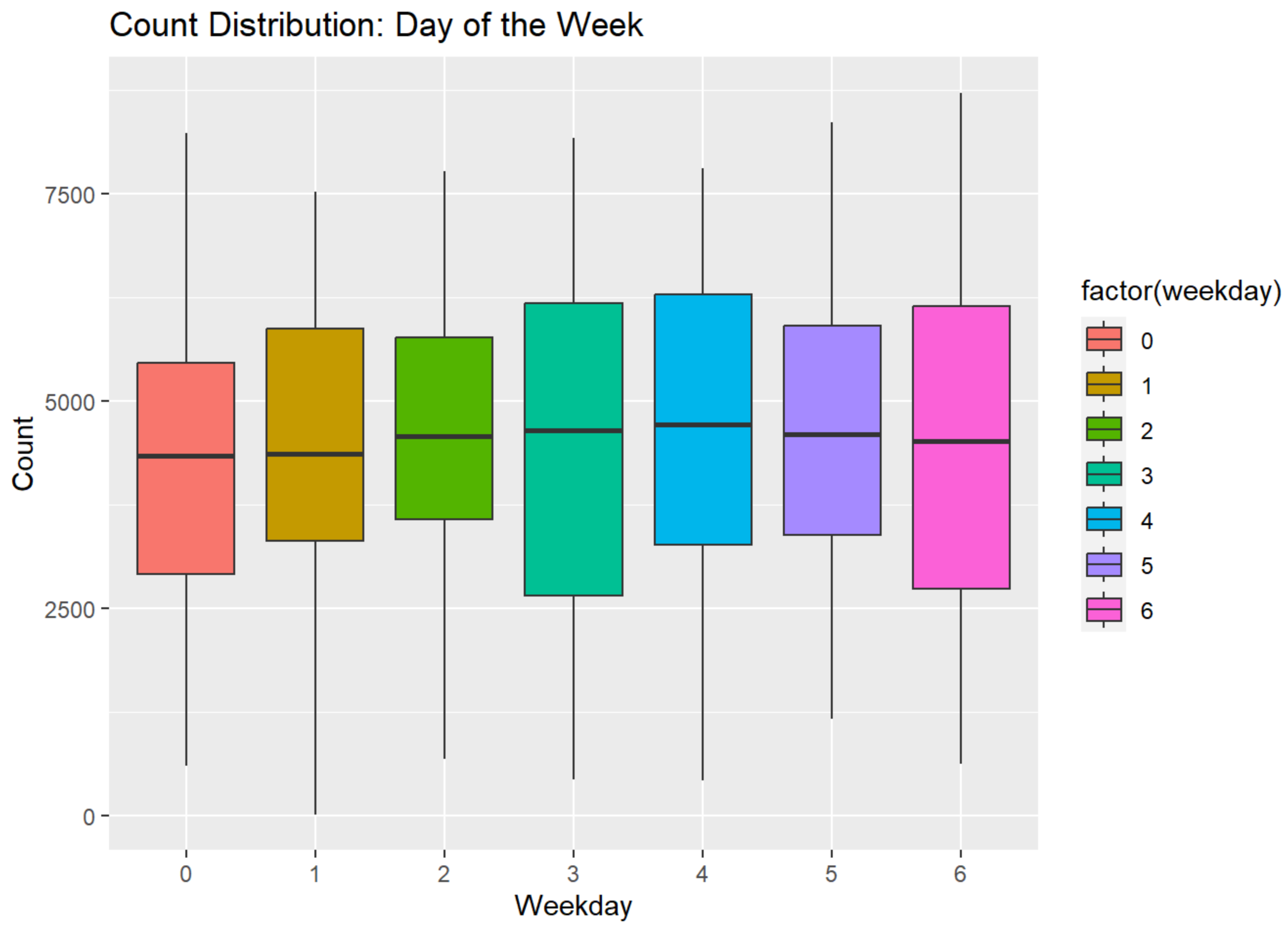
Bar plot of average ridership for per day of the week

```
avg_counts <- day %>%
  group_by(weekday) %>%
  summarise(avg_count = mean(cnt, na.rm = TRUE))

# Reorder the levels of 'weekday' based on average count (descending order)
avg_counts <- avg_counts %>%
  mutate(weekday = reorder(factor(weekday), -avg_count))

# Create the bar chart
ggplot(avg_counts, aes(x = weekday, y = avg_count, fill = factor(weekday))) +
  geom_bar(stat = 'identity') +
  labs(x = "Day of the Week", y = "Average Count", title = "Average Count Comparison: Day of the
Week")
```



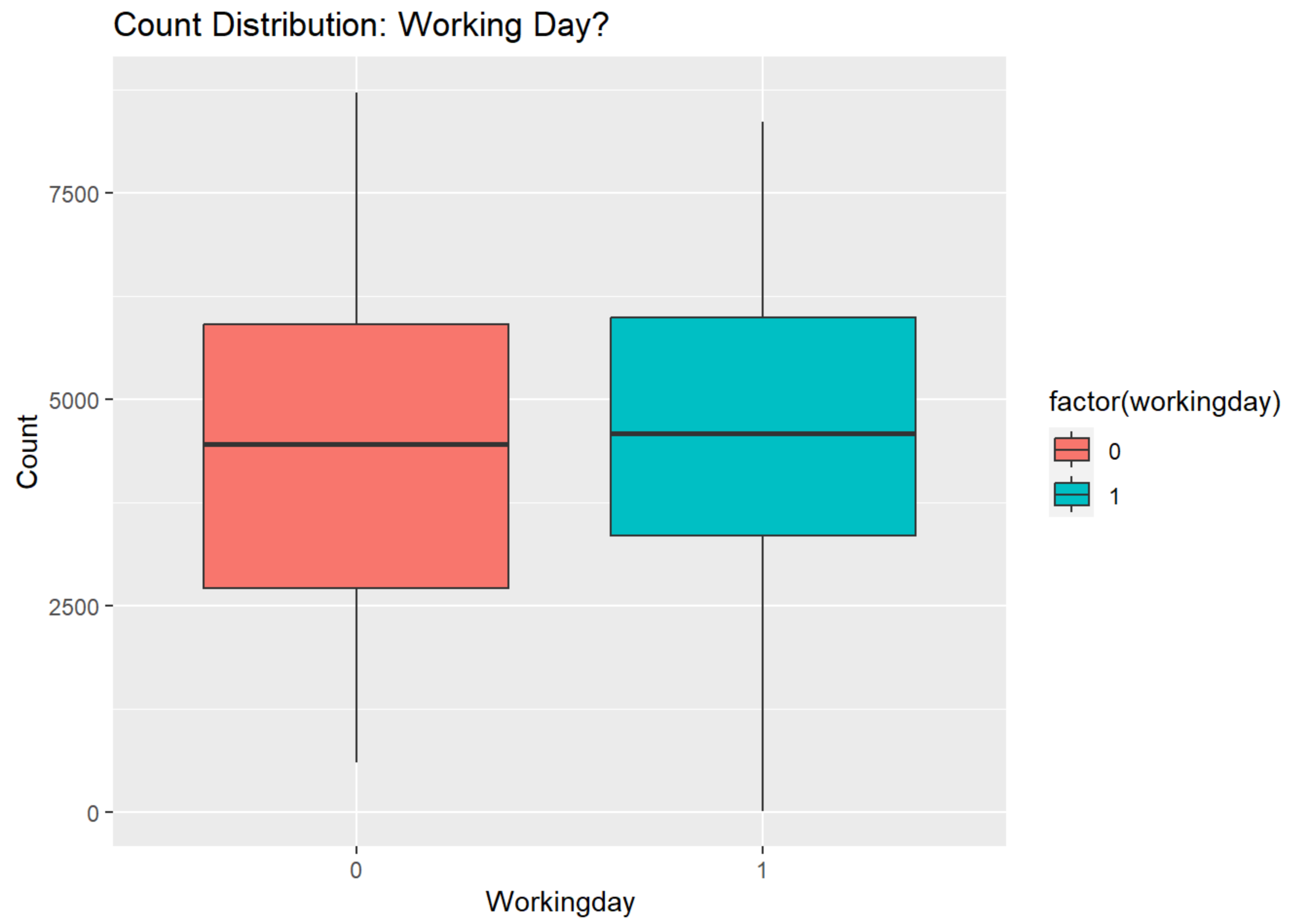
Box plot of Ridership for per day of the week

```
ggplot(day, aes(x = factor(weekday), y = cnt, fill = factor(weekday))) +
  geom_boxplot() +
  labs(x = "Weekday", y = "Count", title = "Count Distribution: Day of the Week")
```



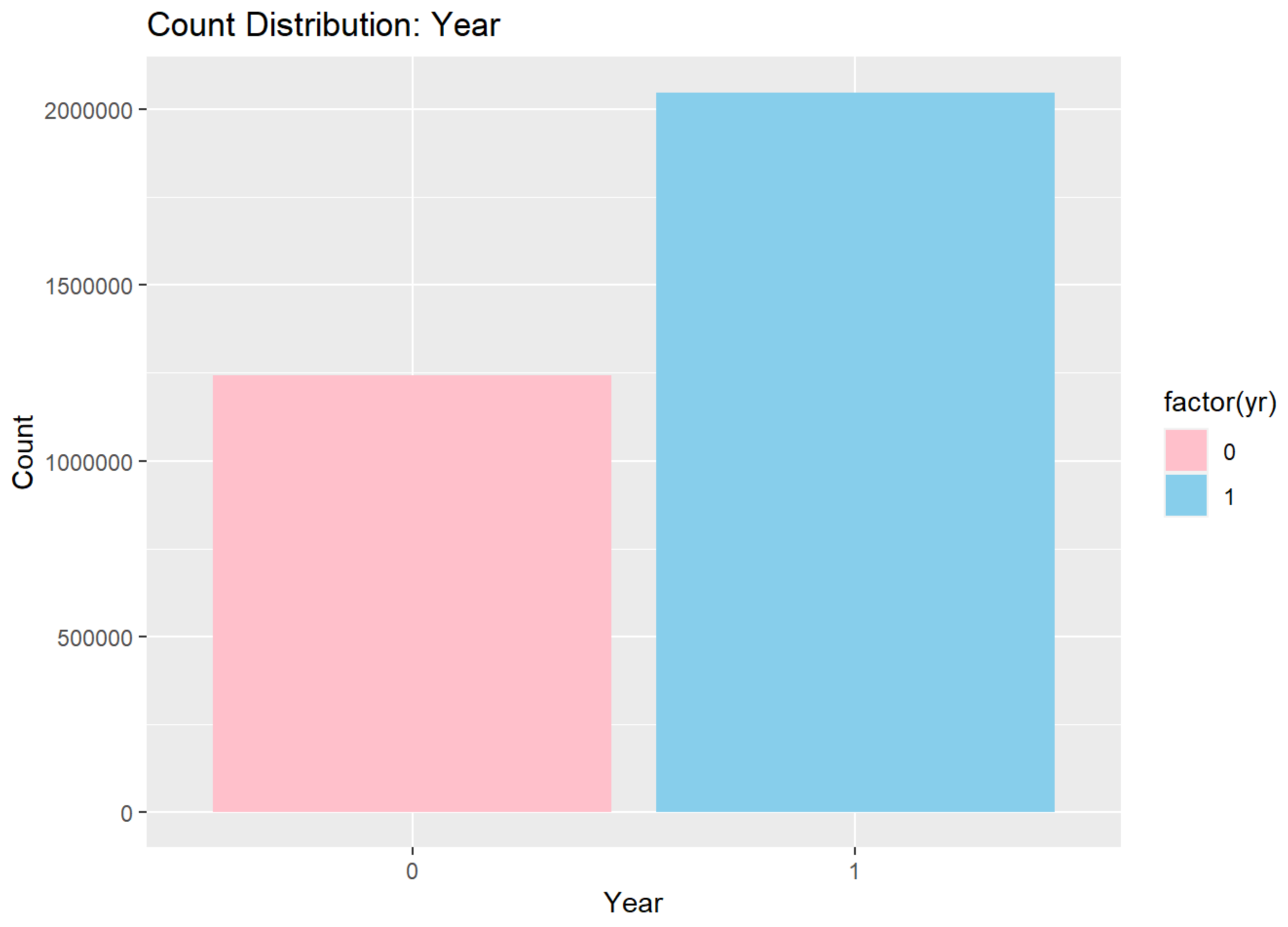
Box plot of Ridership for per Workingday

```
ggplot(day, aes(x = factor(workingday), y = cnt, fill = factor(workingday))) +
  geom_boxplot() +
  labs(x = "Workingday", y = "Count", title = "Count Distribution: Working Day?")
```



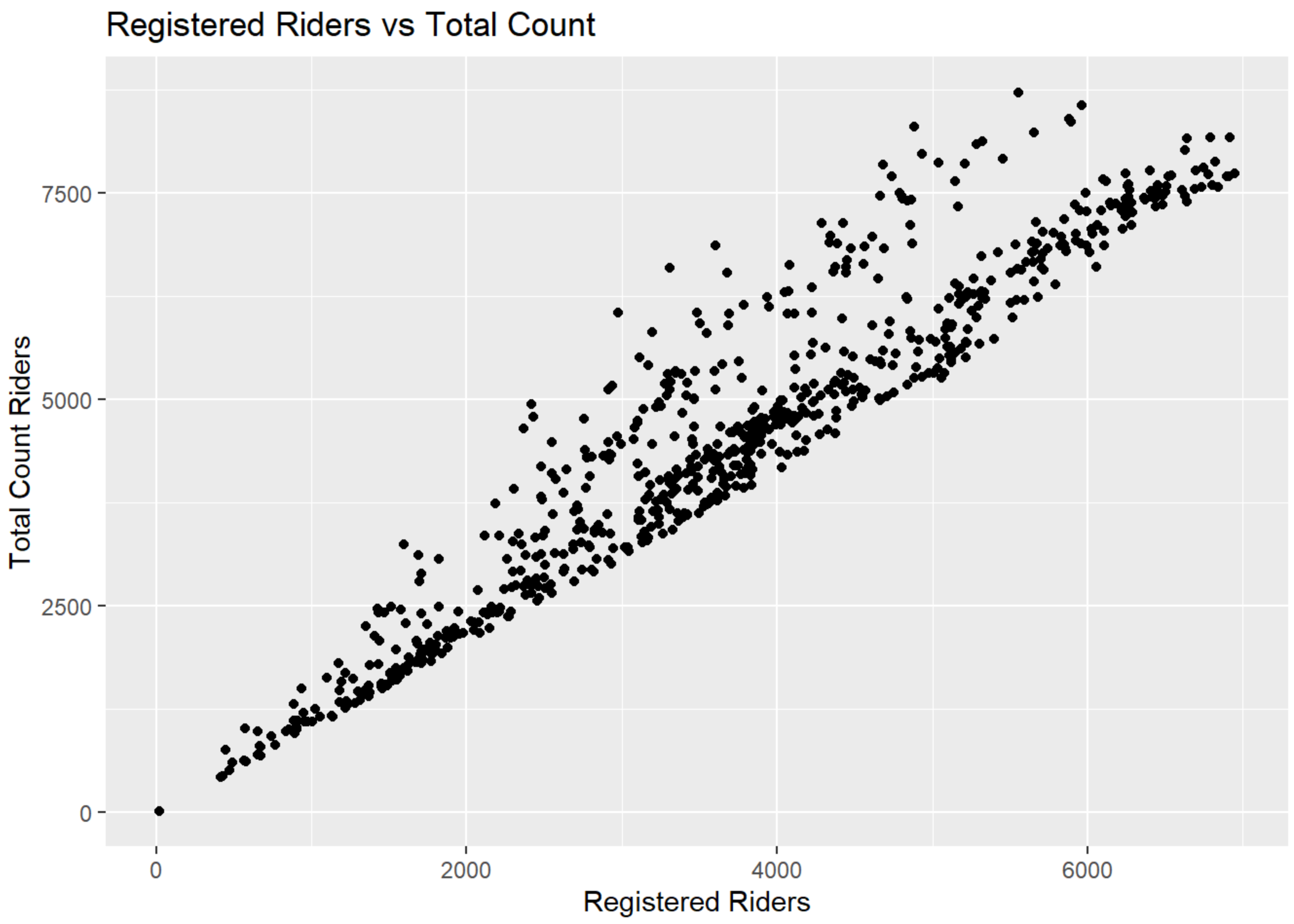
Total number of riders during 2011 and 2012

```
ggplot(day, aes(x = factor(yr), y = cnt, fill = factor(yr))) +
  geom_bar(stat = 'identity') +
  labs(x = "Year", y = "Count", title = "Count Distribution: Year") +
  scale_fill_manual(values = c("pink", "skyblue"))  # Specify colors for boxes
```



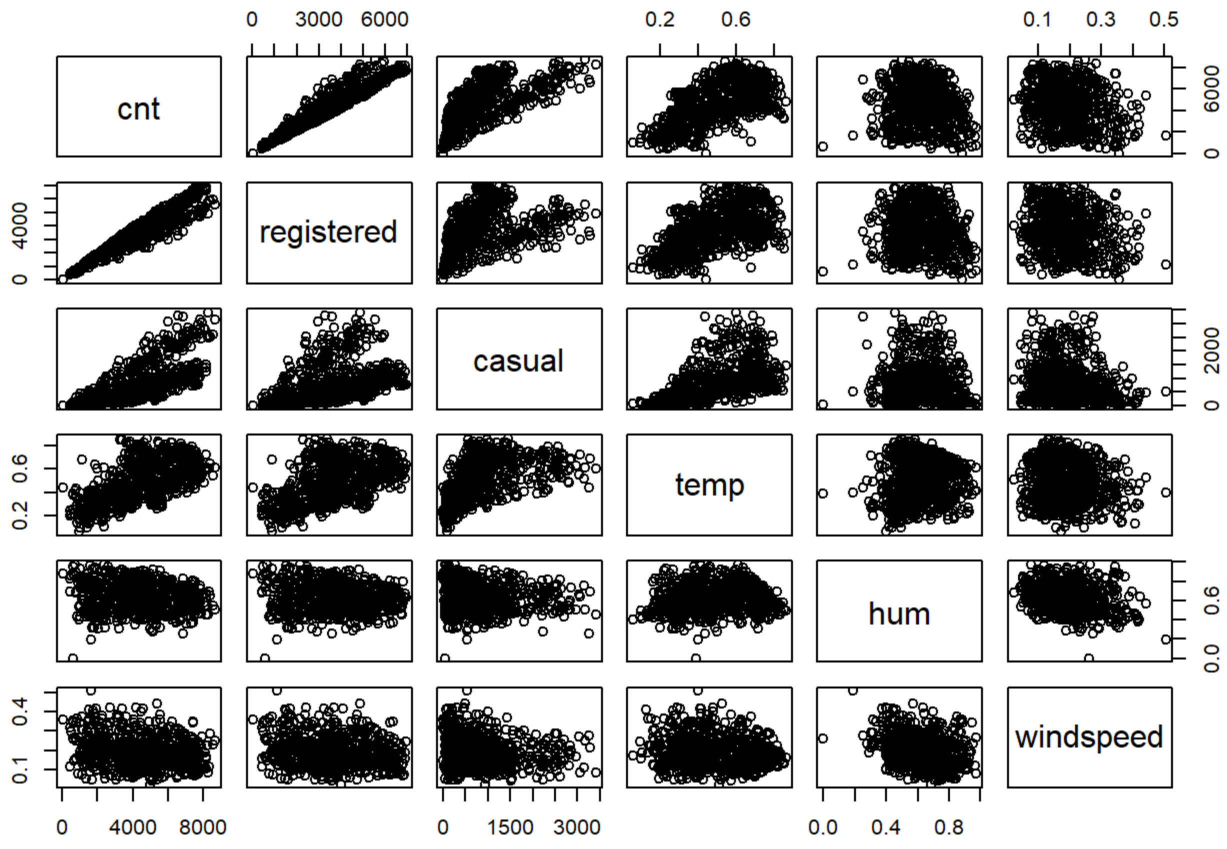
Registered Riders vs Total count

```
ggplot(day, aes( x = registered, y = cnt)) + geom_point() + labs(x = 'Registered Riders', y = 'T
otal Count Riders', title = 'Registered Riders vs Total Count')
```



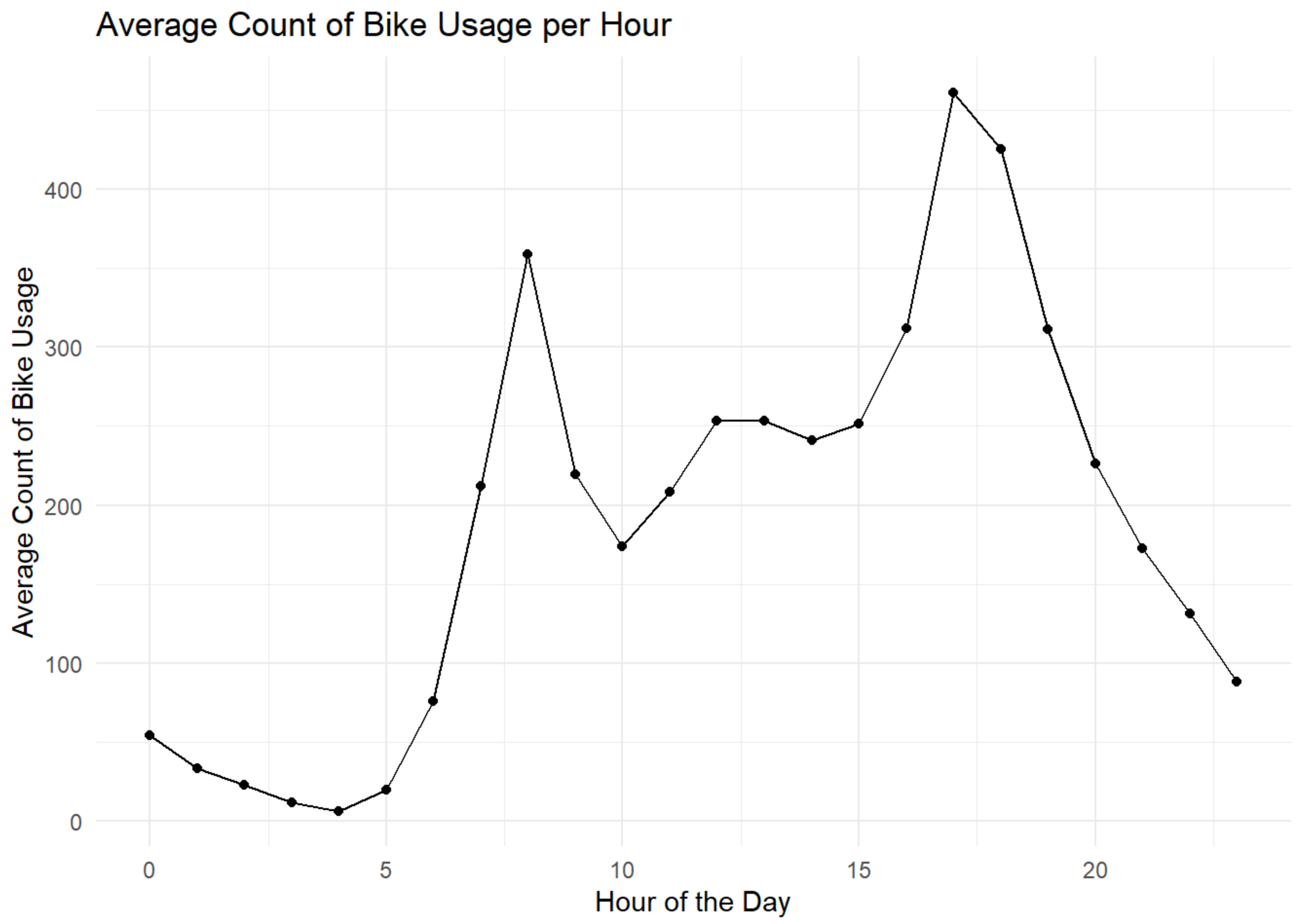
Pairs plot of all numerical variables

```
day_data_2 = day[, c('cnt','registered', 'casual', 'temp', 'hum', 'windspeed')]
pairs(day_data_2)
```



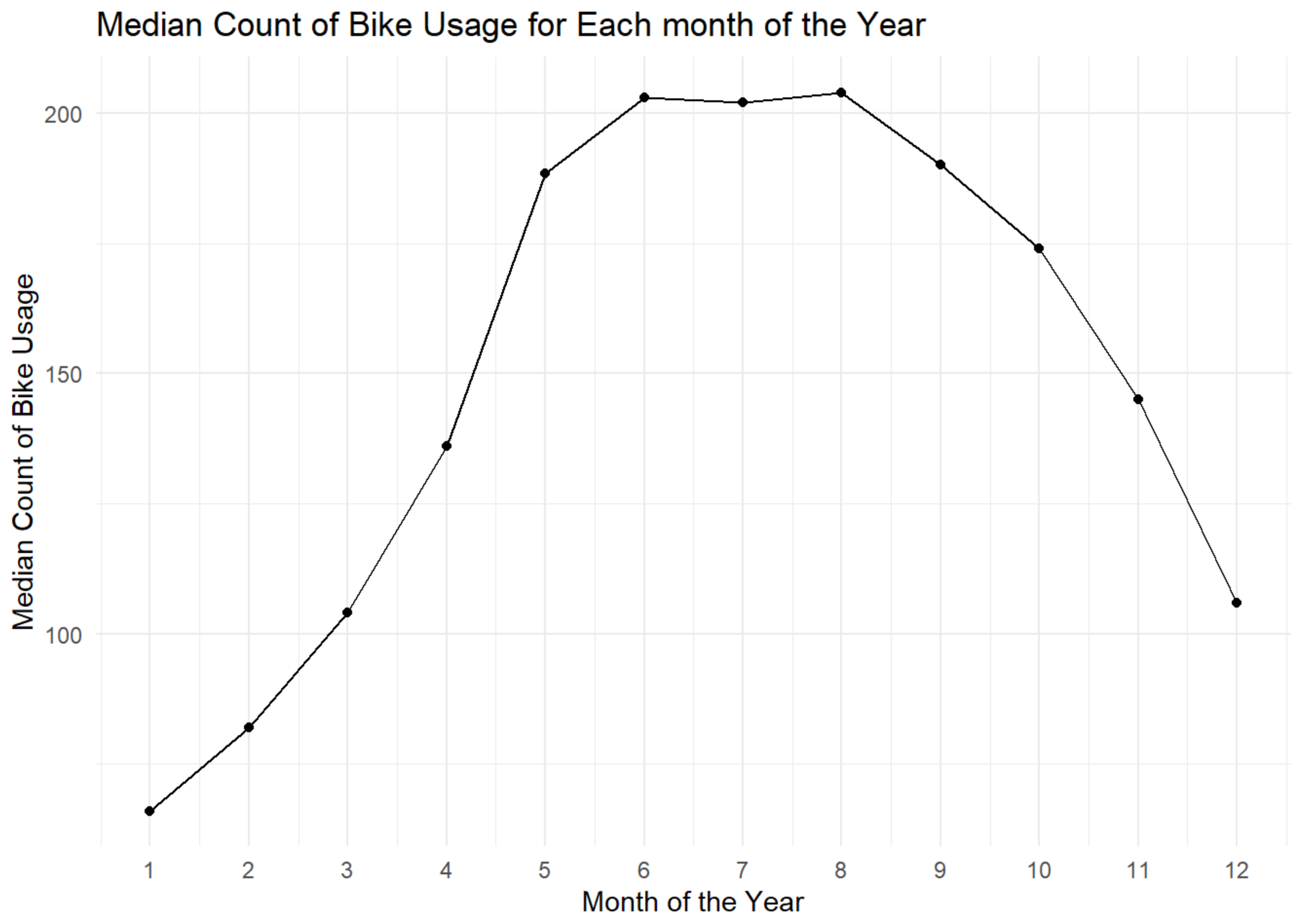
Average bike usage per hour of the day

```
# Calculate average counts per hour
avg_counts <- hour %>%
  group_by(hr) %>%
  summarise(avg_count = mean(cnt, na.rm = TRUE))

# Create the line plot
ggplot(avg_counts, aes(x = hr, y = avg_count)) +
  geom_point() +
  geom_line() +
  labs(x = "Hour of the Day", y = "Average Count of Bike Usage", title = "Average Count of Bike
Usage per Hour") +
  theme_minimal()
```



Median bike usage per month of the Year

```
# Calculate average counts per hour
median_count <- hour %>%
  group_by(mnth) %>%
  summarise(avg_count = median(cnt, na.rm = TRUE))

# Create the line plot
ggplot(median_count, aes(x = mnth, y = avg_count)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:12, labels = 1:12) +
  labs(x = "Month of the Year", y = "Median Count of Bike Usage", title = "Median Count of Bike
Usage for Each month of the Year") +
  theme_minimal()
```


Median Count of Bike Usage for Each month of the Year
Median Count of Bike Usage
200
150
100
1
2
3
4
5
6
7
8
9
10
11
12
Month of the Year

ONE HOT ENCODING

```
# Create binary columns for each season
data <- hour

# Create binary columns for each season
data <- data %>%
  mutate(
    Season1 = ifelse(season == 1, 1, 0),
    Season2 = ifelse(season == 2, 1, 0),
    Season3 = ifelse(season == 3, 1, 0),
    Season4 = ifelse(season == 4, 1, 0)
  )

# Create binary columns for each season
data <- data %>%
  mutate(
    Weather1 = ifelse(weathersit == 1, 1, 0),
    Weather2 = ifelse(weathersit == 2, 1, 0),
    Weather3 = ifelse(weathersit == 3, 1, 0),
    Weather4 = ifelse(weathersit == 4, 1, 0)
  )

data <- data %>%
  mutate(
    Weekday0 = ifelse(weekday == 0, 1, 0),
    Weekday1 = ifelse(weekday == 1, 1, 0),
    Weekday2 = ifelse(weekday == 2, 1, 0),
    Weekday3 = ifelse(weekday == 3, 1, 0),
    Weekday4 = ifelse(weekday == 4, 1, 0),
    Weekday5 = ifelse(weekday == 5, 1, 0),
    Weekday6 = ifelse(weekday == 6, 1, 0)
  )
data <- data %>%
  mutate(
    hr0 = ifelse(hr == 0, 1, 0),
    hr1 = ifelse(hr == 1, 1, 0),
    hr2 = ifelse(hr == 2, 1, 0),
    hr3 = ifelse(hr == 3, 1, 0),
    hr4 = ifelse(hr == 4, 1, 0),
    hr5 = ifelse(hr == 5, 1, 0),
    hr6 = ifelse(hr == 6, 1, 0),
    hr7 = ifelse(hr == 7, 1, 0),
    hr8 = ifelse(hr == 8, 1, 0),
    hr9 = ifelse(hr == 9, 1, 0),
    hr10 = ifelse(hr == 10, 1, 0),
    hr11= ifelse(hr == 11, 1, 0),
    hr12 = ifelse(hr == 12, 1, 0),
    hr13 = ifelse(hr == 13, 1, 0),
    hr14 = ifelse(hr == 14, 1, 0),
    hr15 = ifelse(hr == 15, 1, 0),
    hr16 = ifelse(hr == 16, 1, 0),
    hr17 = ifelse(hr == 17, 1, 0),
    hr18 = ifelse(hr == 18, 1, 0),
    hr19 = ifelse(hr == 19, 1, 0),
    hr20 = ifelse(hr == 20, 1, 0),
    hr21 = ifelse(hr == 21, 1, 0),
    hr22 = ifelse(hr == 22, 1, 0),
    hr23 = ifelse(hr == 23, 1, 0)
  )
data <- data %>%
  mutate(
    month1 = ifelse(mnth == 1, 1, 0),
    month2 = ifelse(mnth == 2, 1, 0),
    month3 = ifelse(mnth == 3, 1, 0),
    month4 = ifelse(mnth == 4, 1, 0),
    month5 = ifelse(mnth == 5, 1, 0),
    month6 = ifelse(mnth == 6, 1, 0),
    month7 = ifelse(mnth == 7, 1, 0),
    month8 = ifelse(mnth == 8, 1, 0),
    month9 = ifelse(mnth == 9, 1, 0),
    month10 = ifelse(mnth == 10, 1, 0),
    month11 = ifelse(mnth == 11, 1, 0),
    month12 = ifelse(mnth == 12, 1, 0)
  )
```

##OHE for DAY

```r
# Create binary columns for each season
data1<- day

# Create binary columns for each season
data1 <- data1 %>%
  mutate(
    Season1 = ifelse(season == 1, 1, 0),
    Season2 = ifelse(season == 2, 1, 0),
    Season3 = ifelse(season == 3, 1, 0),
    Season4 = ifelse(season == 4, 1, 0)
  )

# Create binary columns for each season
data1 <- data1 %>%
  mutate(
    Weather1 = ifelse(weathersit == 1, 1, 0),
    Weather2 = ifelse(weathersit == 2, 1, 0),
    Weather3 = ifelse(weathersit == 3, 1, 0),
    Weather4 = ifelse(weathersit == 4, 1, 0)
  )

data1 <- data1 %>%
  mutate(
    Weekday0 = ifelse(weekday == 0, 1, 0),
    Weekday1 = ifelse(weekday == 1, 1, 0),
    Weekday2 = ifelse(weekday == 2, 1, 0),
    Weekday3 = ifelse(weekday == 3, 1, 0),
    Weekday4 = ifelse(weekday == 4, 1, 0),
    Weekday5 = ifelse(weekday == 5, 1, 0),
    Weekday6 = ifelse(weekday == 6, 1, 0)
  )

data1 <- data1 %>%
  mutate(
    month1 = ifelse(mnth == 1, 1, 0),
    month2 = ifelse(mnth == 2, 1, 0),
    month3 = ifelse(mnth == 3, 1, 0),
    month4 = ifelse(mnth == 4, 1, 0),
    month5 = ifelse(mnth == 5, 1, 0),
    month6 = ifelse(mnth == 6, 1, 0),
    month7 = ifelse(mnth == 7, 1, 0),
    month8 = ifelse(mnth == 8, 1, 0),
    month9 = ifelse(mnth == 9, 1, 0),
    month10 = ifelse(mnth == 10, 1, 0),
    month11 = ifelse(mnth == 11, 1, 0),
    month12 = ifelse(mnth == 12, 1, 0)
  )
```

-Testing the best subset for our model on all categorical variables

```r
names(data)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"      
##  [6] "hr"         "holiday"    "weekday"    "workingday" "weathersit"
## [11] "temp"       "atemp"      "hum"        "windspeed"  "casual"    
## [16] "registered" "cnt"        "Season1"    "Season2"    "Season3"   
## [21] "Season4"    "Weather1"   "Weather2"   "Weather3"   "Weather4"  
## [26] "Weekday0"   "Weekday1"   "Weekday2"   "Weekday3"   "Weekday4"  
## [31] "Weekday5"   "Weekday6"   "hr0"        "hr1"        "hr2"       
## [36] "hr3"        "hr4"        "hr5"        "hr6"        "hr7"       
## [41] "hr8"        "hr9"        "hr10"       "hr11"       "hr12"      
## [46] "hr13"       "hr14"       "hr15"       "hr16"       "hr17"      
## [51] "hr18"       "hr19"       "hr20"       "hr21"       "hr22"      
## [56] "hr23"       "month1"     "month2"     "month3"     "month4"    
## [61] "month5"     "month6"     "month7"     "month8"     "month9"    
## [66] "month10"    "month11"    "month12"
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.2
```

```
data$hrssn= data$season * data$workingday
data$weathersitwork= data$hr8 * data$weathersit
bhour <- regsubsets(cnt~registered+Season1+Season2+Season3+Season4+Weekday0+Weekday1+Weekday2+We
ekday3+Weekday4+Weekday5+Weekday6+Weather1+Weather2+Weather3+Weather4
                    +hr1+hr2+hr3+hr4+hr5+hr6+hr7+hr8+hr9+hr10+hr11+hr12+hr13+hr14+hr15+hr16+hr17
+hr18+hr19+hr20+hr21+hr22+hr23+month1+month2
                    +month3+month4+month5+month6+month7+month8+month9+month10+month11+month12+ho
liday+workingday+yr+temp+windspeed+atemp+hum+weathersitwork+hrssn, data,really.big=T)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 5 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(bhour)
```

```
## Subset selection object
## Call: regsubsets.formula(cnt ~ registered + Season1 + Season2 + Season3 +
##     Season4 + Weekday0 + Weekday1 + Weekday2 + Weekday3 + Weekday4 +
##     Weekday5 + Weekday6 + Weather1 + Weather2 + Weather3 + Weather4 +
##     hr1 + hr2 + hr3 + hr4 + hr5 + hr6 + hr7 + hr8 + hr9 + hr10 +
##     hr11 + hr12 + hr13 + hr14 + hr15 + hr16 + hr17 + hr18 + hr19 +
##     hr20 + hr21 + hr22 + hr23 + month1 + month2 + month3 + month4 +
##     month5 + month6 + month7 + month8 + month9 + month10 + month11 +
##     month12 + holiday + workingday + yr + temp + windspeed +
##     atemp + hum + weathersitwork + hrssn, data, really.big = T)
## 60 Variables  (and intercept)
##                Forced in Forced out
## registered         FALSE      FALSE
## Season1            FALSE      FALSE
## Season2            FALSE      FALSE
## Season3            FALSE      FALSE
## Weekday0           FALSE      FALSE
## Weekday1           FALSE      FALSE
## Weekday2           FALSE      FALSE
## Weekday3           FALSE      FALSE
## Weekday4           FALSE      FALSE
## Weekday5           FALSE      FALSE
## Weather1           FALSE      FALSE
## Weather2           FALSE      FALSE
## Weather3           FALSE      FALSE
## hr1                FALSE      FALSE
## hr2                FALSE      FALSE
## hr3                FALSE      FALSE
## hr4                FALSE      FALSE
## hr5                FALSE      FALSE
## hr6                FALSE      FALSE
## hr7                FALSE      FALSE
## hr8                FALSE      FALSE
## hr9                FALSE      FALSE
## hr10               FALSE      FALSE
## hr11               FALSE      FALSE
## hr12               FALSE      FALSE
## hr13               FALSE      FALSE
## hr14               FALSE      FALSE
## hr15               FALSE      FALSE
## hr16               FALSE      FALSE
## hr17               FALSE      FALSE
## hr18               FALSE      FALSE
## hr19               FALSE      FALSE
## hr20               FALSE      FALSE
## hr21               FALSE      FALSE
## hr22               FALSE      FALSE
## hr23               FALSE      FALSE
## month1             FALSE      FALSE
## month2             FALSE      FALSE
## month3             FALSE      FALSE
## month4             FALSE      FALSE
## month5              FALSE      FALSE
## month6              FALSE      FALSE
## month7              FALSE      FALSE
## month8              FALSE      FALSE
## month9              FALSE      FALSE
## month10             FALSE      FALSE
## month11             FALSE      FALSE
## holiday             FALSE      FALSE
## yr                  FALSE      FALSE
## temp                FALSE      FALSE
## windspeed           FALSE      FALSE
## atemp               FALSE      FALSE
## hum                 FALSE      FALSE
## weathersitwork      FALSE      FALSE
## hrssn               FALSE      FALSE
## Season4             FALSE      FALSE
## Weekday6            FALSE      FALSE
## Weather4            FALSE      FALSE
## month12             FALSE      FALSE
## workingday          FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##           registered Season1 Season2 Season3 Season4 Weekday0 Weekday1 Weekday2
## 1  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 2  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 3  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 4  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 5  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 6  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 7  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 8  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
## 9  ( 1 ) "*"        " "     " "     " "     " "     " "      " "      " "     
##           Weekday3 Weekday4 Weekday5 Weekday6 Weather1 Weather2 Weather3
## 1  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 2  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 3  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 4  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 5  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 6  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 7  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 8  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
## 9  ( 1 ) " "      " "      " "      " "      " "      " "      " "     
##           Weather4 hr1 hr2 hr3 hr4 hr5 hr6 hr7 hr8 hr9 hr10 hr11 hr12 hr13 hr14
## 1  ( 1 ) " "      " " " " " " " " " " " " " " " " " " " "  " "  " "  " "  " " 
## 2  ( 1 ) " "      " " " " " " " " " " " " " " " " " " " "  " "  " "  " "  " " 
## 3  ( 1 ) " "      " " " " " " " " " " " " " " " " " " " "  " "  " "  " "  " " 
## 4  ( 1 ) " "      " " " " " " " " " " " " " " " " " " " "  " "  " "  " "  " " 
## 5  ( 1 ) " "      " " " " " " " " " " " " " " "*" " " " "  " "  " "  " "  " " 
## 6  ( 1 ) " "      " " " " " " " " " " " " " " "*" " " " "  " "  " "  " "  "*" 
## 7  ( 1 ) " "      " " " " " " " " " " " " " " "*" " " " "  " "  " "  " "  "*" 
## 8  ( 1 ) " "      " " " " " " " " " " " " " " "*" " " " "  " "  " "  "*"  "*" 
## 9  ( 1 ) " "      " " " " " " " " " " " " " " "*" " " " "  " "  "*"  "*"  "*" 
##           hr15 hr16 hr17 hr18 hr19 hr20 hr21 hr22 hr23 month1 month2 month3
## 1  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 2  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 3  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 4  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 5  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 6  ( 1 ) " "  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 7  ( 1 ) "*"  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 8  ( 1 ) "*"  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
## 9  ( 1 ) "*"  " "  " "  " "  " "  " "  " "  " "  " "  " "    " "    " "   
##           month4 month5 month6 month7 month8 month9 month10 month11 month12
## 1  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 2  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 3  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 4  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 5  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 6  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 7  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 8  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
## 9  ( 1 ) " "    " "    " "    " "    " "    " "    " "     " "     " "    
##           holiday workingday yr  temp windspeed atemp hum weathersitwork hrssn
## 1  ( 1 ) " "     " "        " " " "  " "       " "   " " " "            " "  
## 2  ( 1 ) " "     "*"        " " " "  " "       " "   " " " "            " "  
## 3  ( 1 ) " "     "*"        " " "*"  " "       " "   " " " "            " "  
## 4  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
## 5  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
## 6  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
## 7  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
## 8  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
## 9  ( 1 ) " "     "*"        " " "*"  " "       " "   "*" " "            " "  
```

```
summary(bhour)$adjr2
```

```
## [1] 0.9450739 0.9553170 0.9624928 0.9655591 0.9668426 0.9676995 0.9685001
## [8] 0.9693260 0.9700993
```

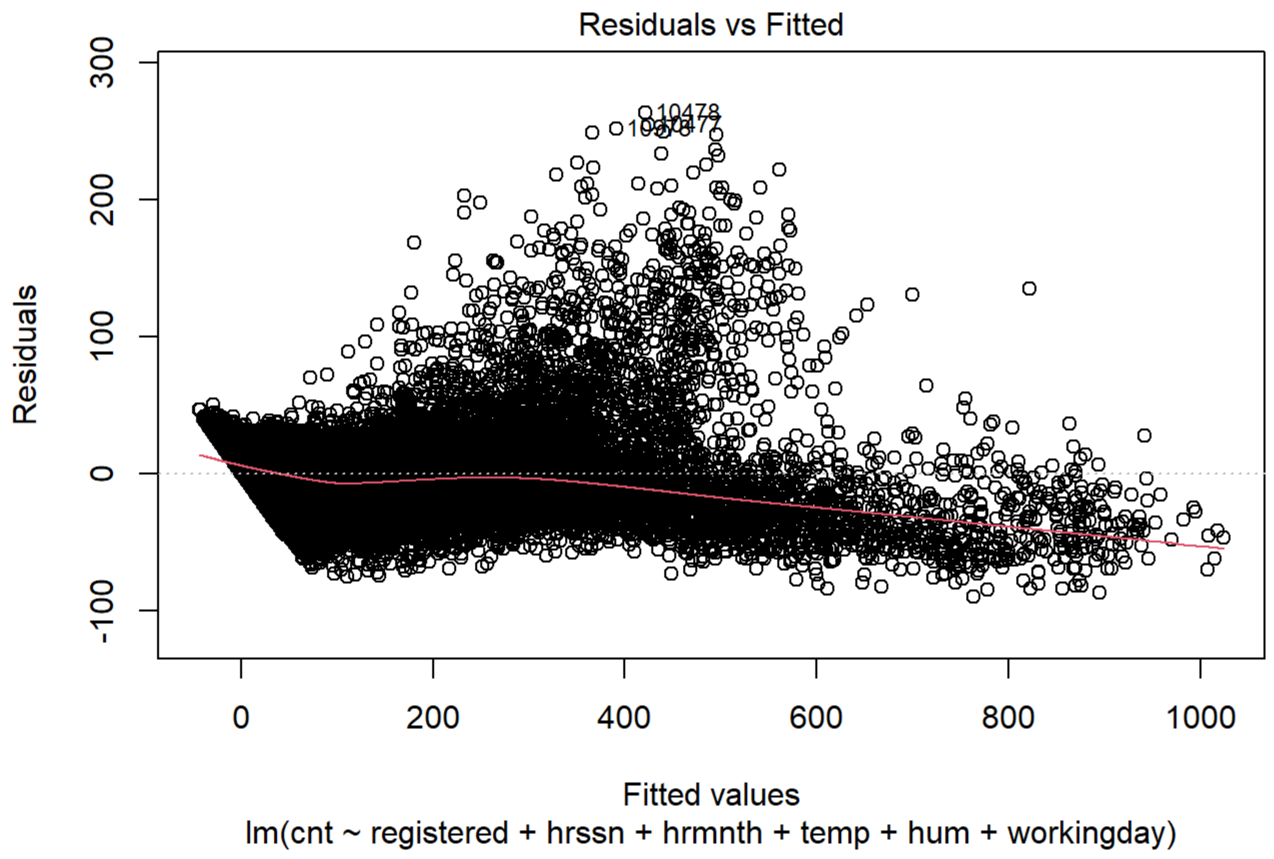
```
which.max(summary(bhour)$adjr2)
```

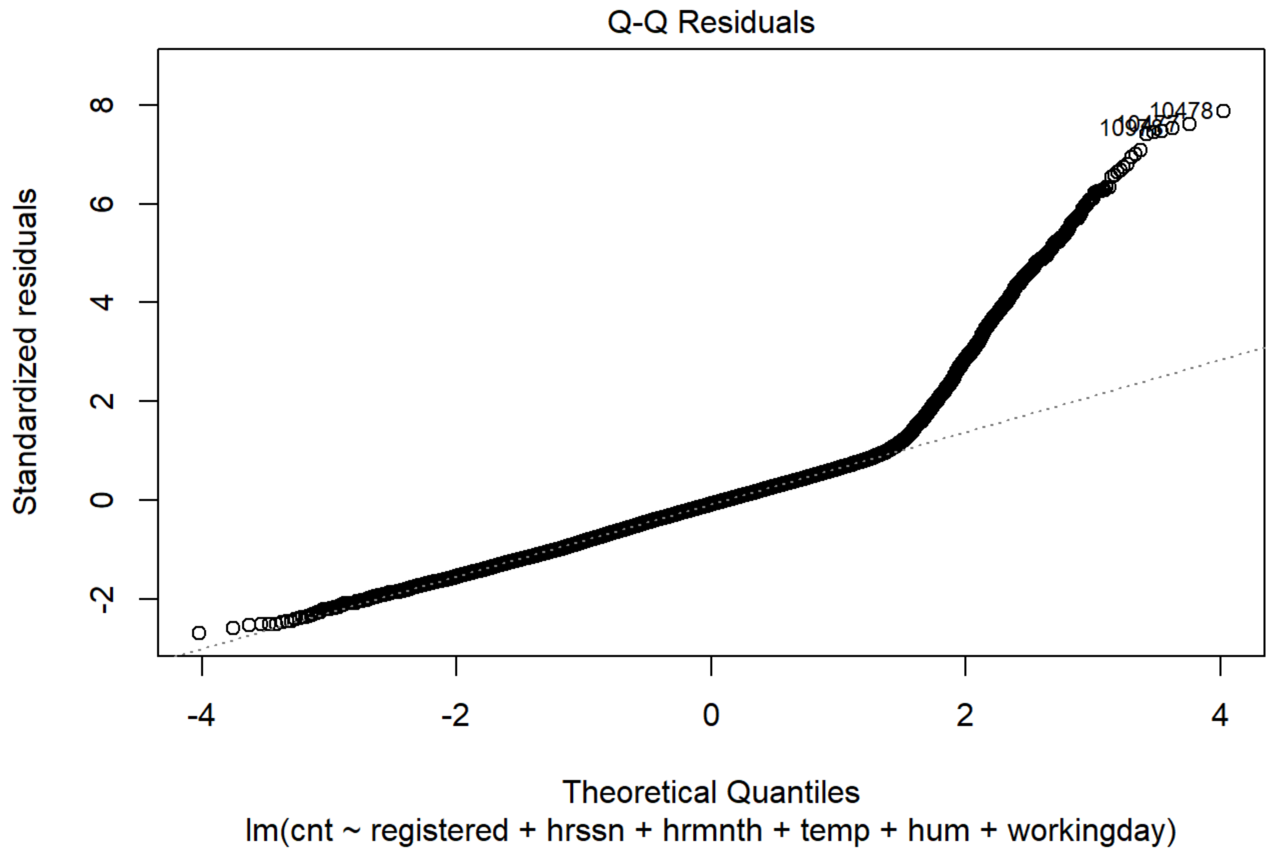
```
## [1] 9
```

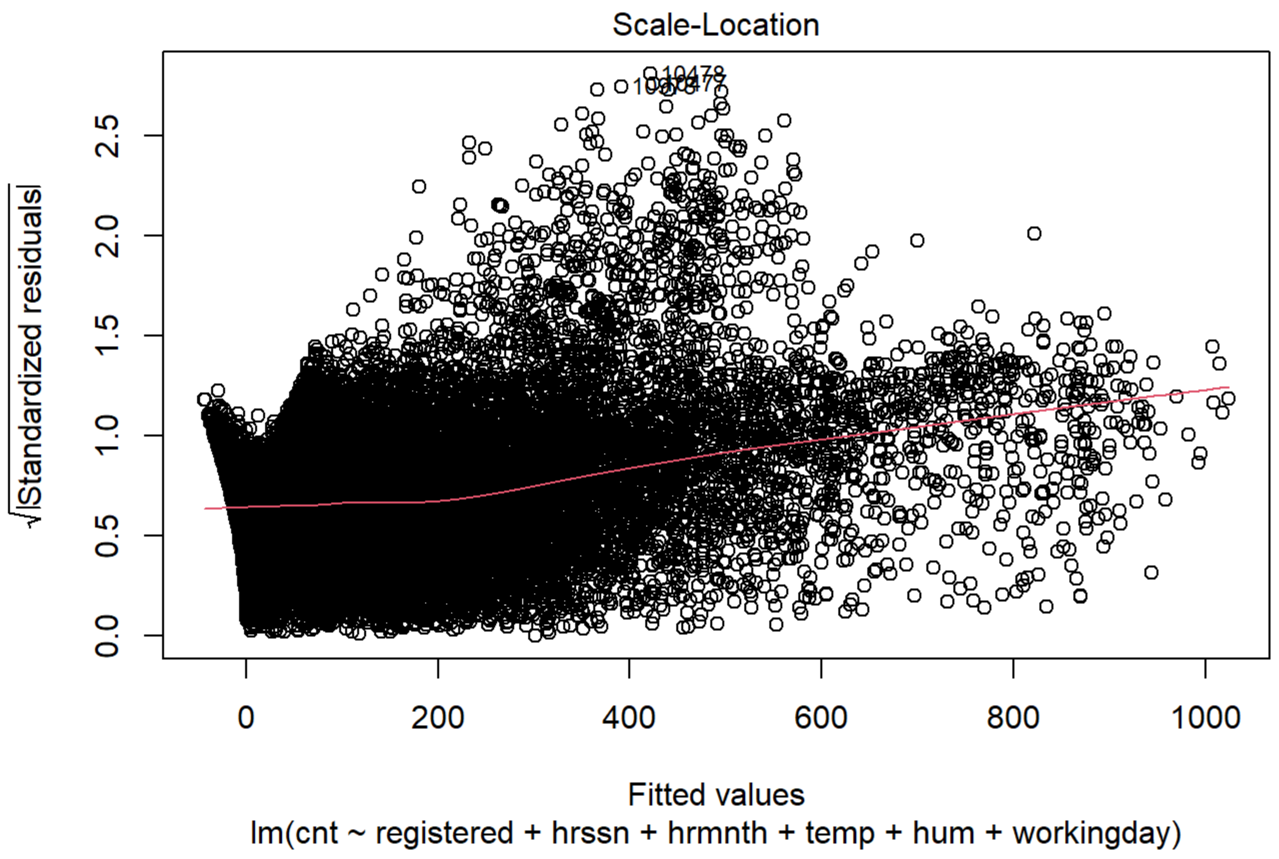
```
day1 = subset(day, yr == 1)
hour$hrssn= hour$season * hour$workingday
hour$hrmnth= hour$workingday * hour$weathersit
big_model = lm(cnt ~registered+ hrssn +hrmnth+temp +hum+workingday, data = hour)
summary(big_model)
```

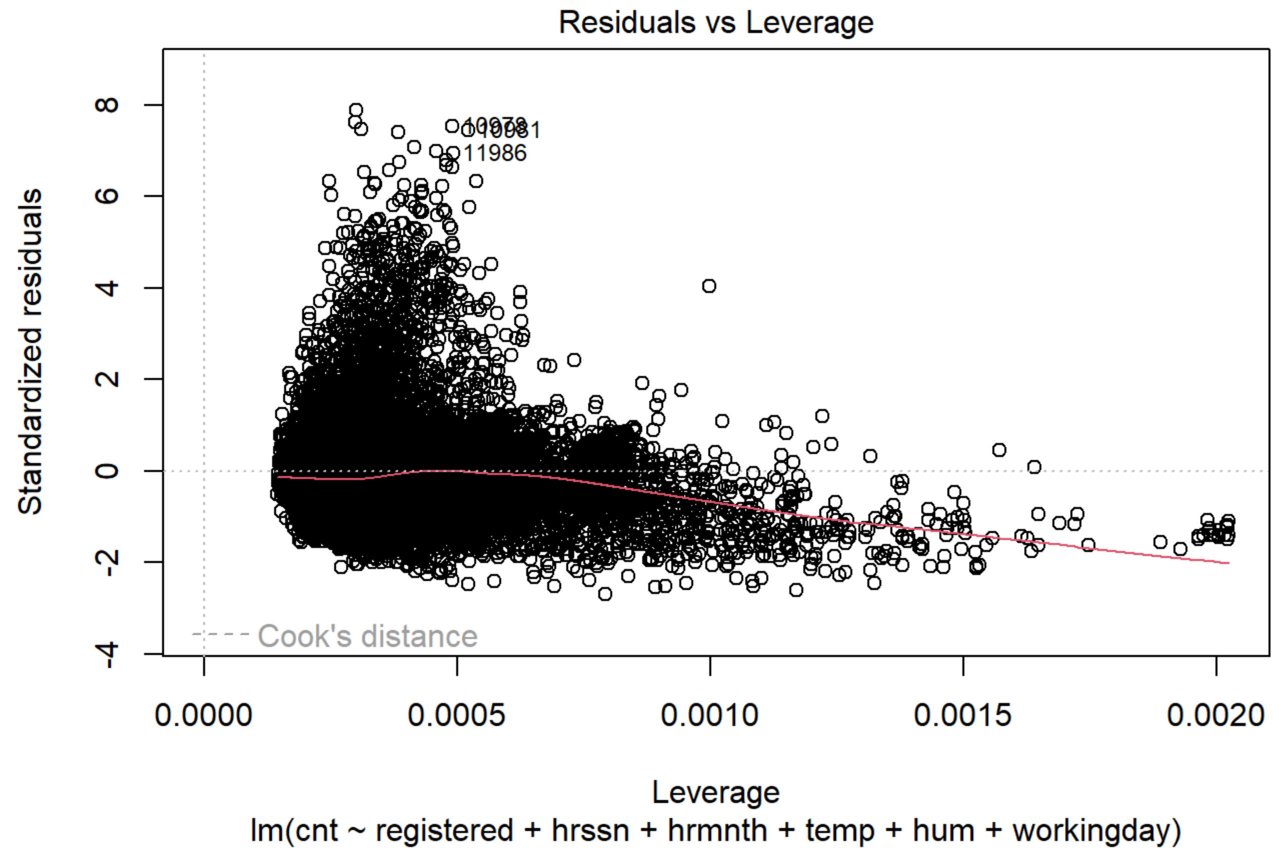
```
##
## Call:
## lm(formula = cnt ~ registered + hrssn + hrmnth + temp + hum +
##     workingday, data = hour)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.165 -19.584  -2.762  13.401 263.433
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  34.266900   1.265089  27.087   <2e-16 ***
## registered    1.127499   0.001874 601.627   <2e-16 ***
## hrssn        -2.792263   0.294990  -9.466   <2e-16 ***
## hrmnth        6.114442   0.504587  12.118   <2e-16 ***
## temp         90.075603   1.436433  62.708   <2e-16 ***
## hum         -57.785052   1.475828 -39.154   <2e-16 ***
## workingday  -40.957645   1.179222 -34.733   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.43 on 17372 degrees of freedom
## Multiple R-squared:  0.966,  Adjusted R-squared:  0.966
## F-statistic: 8.237e+04 on 6 and 17372 DF,  p-value: < 2.2e-16
```

```
plot(big_model)
```


Residuals vs Fitted
Residuals
300
200
100
0
-100
0
200
400
600
800
1000
10478
10477
10476
Fitted values
lm(cnt ~ registered + hrssn + hrmnth + temp + hum + workingday)


Q-Q Residuals
Standardized residuals
8
6
4
2
0
-2
-4
-2
0
2
4
10478
10477
10476
Theoretical Quantiles
lm(cnt ~ registered + hrssn + hrmnth + temp + hum + workingday)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
10478
10477
0
200
400
600
800
1000
Fitted values
lm(cnt ~ registered + hrssn + hrmnth + temp + hum + workingday)


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
6
8
10978
10981
11986
Cook's distance
0.0000
0.0005
0.0010
0.0015
0.0020
Leverage
lm(cnt ~ registered + hrssn + hrmnth + temp + hum + workingday)

```
library(caret)
```

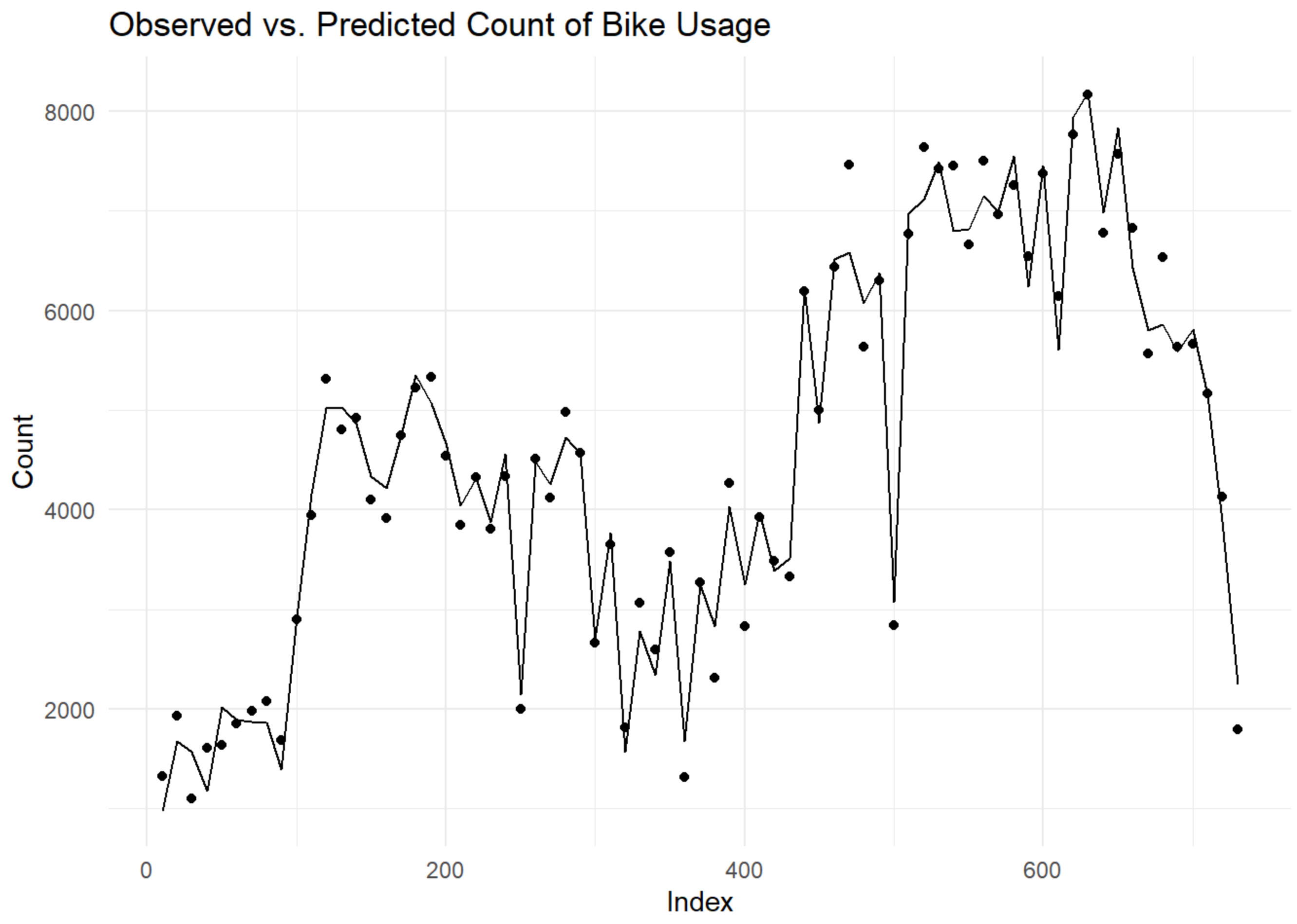
```
## Warning: package 'caret' was built under R version 4.3.2
```

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

```
day_model = lm(cnt ~registered+ mnth +weathersit+temp +hum+workingday+season, data = day)
predicted_values <- predict(day_model, newdata = day)
RMSE(predicted_values, day$cnt)
```

```
## [1] 353.8768
```

```
comparison_data <- data.frame(
  Instant = day$instant,
  Observed = day$cnt,
  Predicted = predicted_values
)
filtered_data <- comparison_data %>% filter(row_number() %% 10 == 0)

ggplot(filtered_data, aes(x = Instant, y = Observed)) +
  geom_point() +
  geom_line(aes(x = Instant, y = Predicted)) +
  labs(x = "Index", y = "Count", title = "Observed vs. Predicted Count of Bike Usage") +
  theme_minimal()
```



```
subset(comparison_data, Predicted < 0)
```

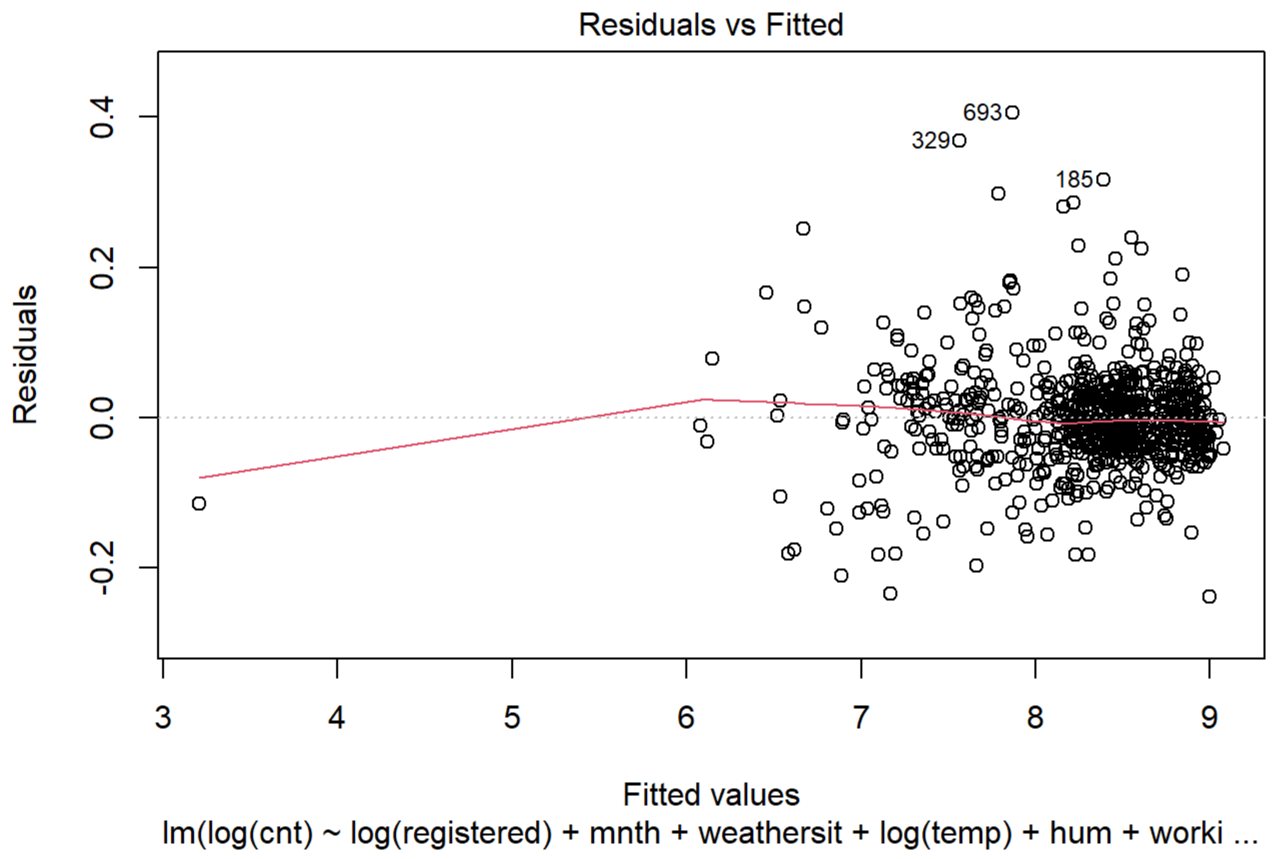
```
##     Instant Observed  Predicted
## 26       26      506  -98.19399
## 27       27      431  -26.45746
## 668     668       22 -195.19768
## 726     726      441  -84.37476
```

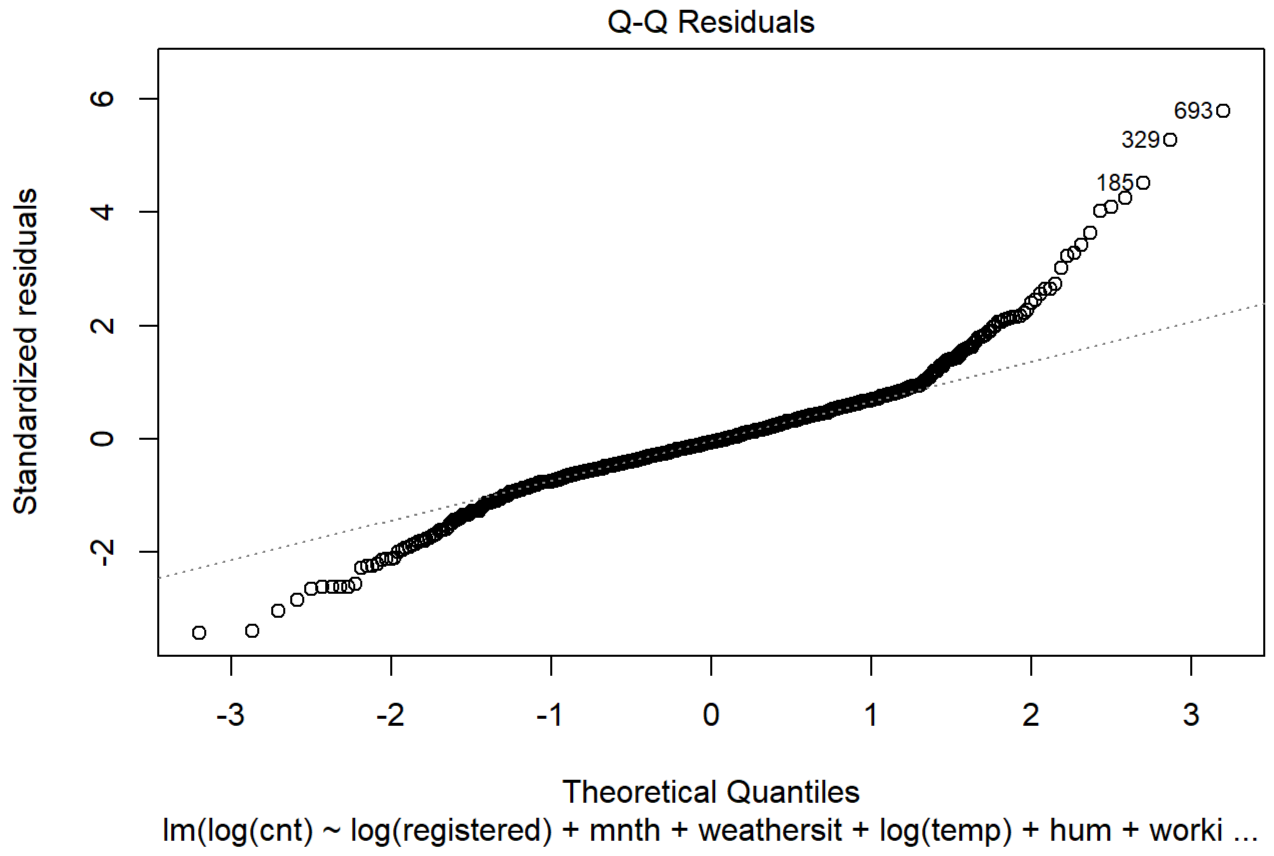
```
comparison_data
```

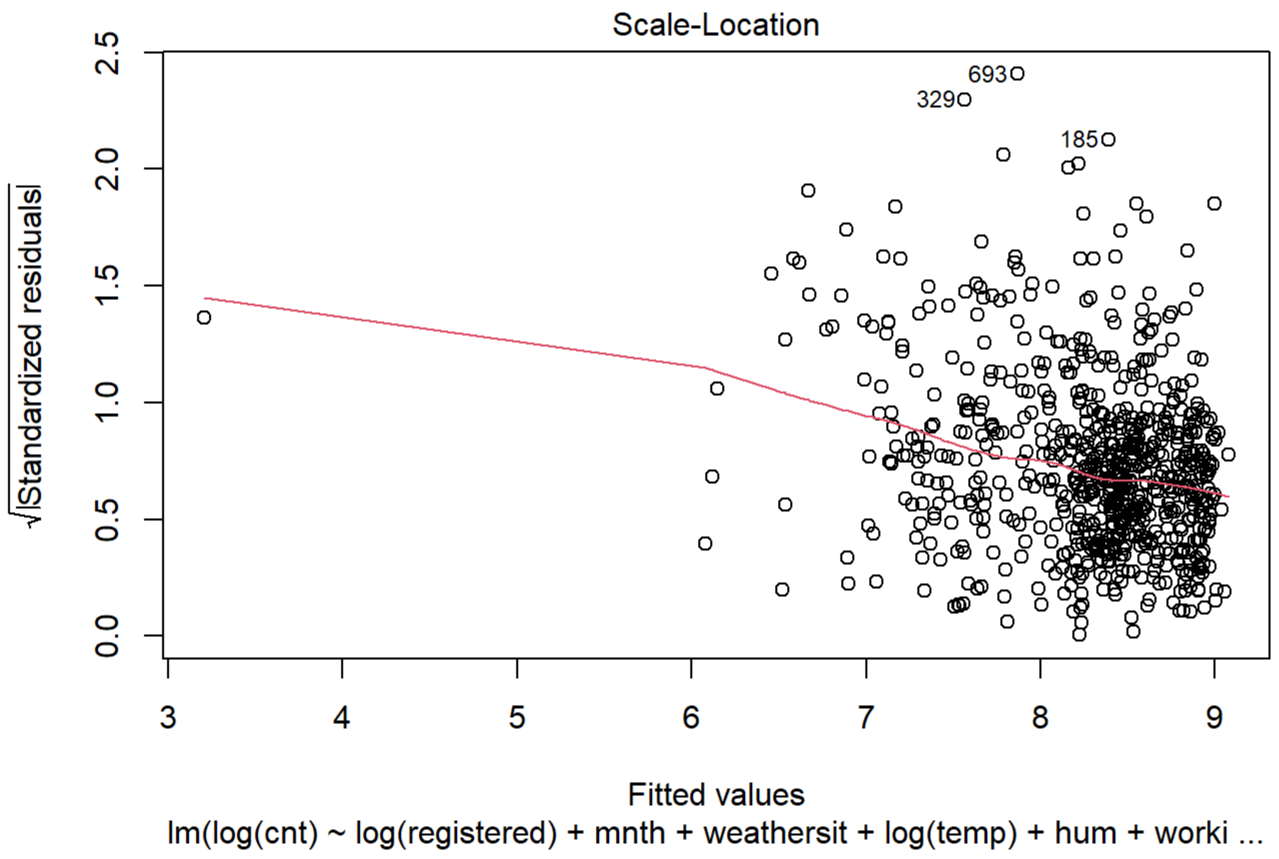
```
##     Instant Observed  Predicted
## 1         1      985 1275.97945
## 2         2      801 1338.83443
## 3         3     1349  975.89997
## 4         4     1562 1208.90400
## 5         5     1600 1345.24015
## 6         6     1606 1303.62844
## 7         7     1510 1056.12777
## 8         8      959 1419.10817
## 9         9      822 1330.92227
## 10       10     1321  976.92854
## 11       11     1263  821.06660
## 12       12     1162  808.59495
## 13       13     1406 1097.23565
## 14       14     1421 1077.15622
## 15       15     1248 1657.21891
## 16       16     1204 1635.28229
## 17       17     1000 1420.98276
## 18       18      683  198.04751
## 19       19     1650 1350.43777
## 20       20     1927 1678.53427
## 21       21     1543 1229.92303
## 22       22      981 1393.59915
## 23       23      986 1365.03863
## 24       24     1416  976.28240
## 25       25     1985 1568.46565
## 26       26      506  -98.19399
## 27       27      431  -26.45746
## 28       28     1167  728.77051
## 29       29     1098 1587.26964
## 30       30     1096 1571.30360
## 31       31     1501 1129.77567
## 32       32     1360  886.94379
## 33       33     1526 1135.17645
## 34       34     1550 1232.36897
## 35       35     1708 1317.55712
## 36       36     1005 1387.37254
## 37       37     1623 2006.69069
## 38       38     1712 1377.99965
## 39       39     1530 1220.15725
## 40       40     1605 1176.20993
## 41       41     1538 1189.35378
## 42       42     1746 1345.69737
## 43       43     1472 1845.33981
## 44       44     1589 1973.60375
## 45       45     1913 1740.51228
## 46       46     1815 1559.99928
## 47       47     2115 1850.74822
## 48       48     2475 2330.17335
## 49       49     2927 2572.95888
## 50       50     1635 2016.06986
## 51       51     1812 1928.75656
## 52       52     1107 1539.95456
## 53       53     1450 1065.56346
## 54       54     1917 1609.29046
## 55       55     1807 1484.70391
## 56       56     1461 1128.43716
## 57       57     1969 2331.18784
## 58       58     2402 2555.50470
## 59       59     1446 1166.82647
## 60       60     1851 1886.95047
## 61       61     2134 2198.36846
## 62       62     1685 1683.65745
## 63       63     1944 1822.37559
## 64       64     2077 2526.46991
## 65       65      605 1382.10367
## 66       66     1872 1778.00680
## 67       67     2133 2059.05593
## 68       68     1891 1788.34688
## 69       69      623  684.91614
## 70       70     1977 1872.44649
## 71       71     2132 2537.56860
## 72       72     2417 2642.32913
## 73       73     2046 1926.12619
## 74       74     2056 1915.16313
## 75       75     2192 2061.29350
## 76       76     2744 2736.42575
## 77       77     3239 2927.22270
## 78       78     3117 3068.77800
## 79       79     2471 2585.77107
## 80       80     2077 1860.81896
## 81       81     2703 2619.20782
## 82       82     2121 2031.70228
## 83       83     1865 1717.78530
## 84       84     2210 2069.91230
## 85       85     2496 2586.51386
## 86       86     1693 2147.28485
## 87       87     2028 1990.46604
## 88       88     2425 2380.36091
## 89       89     1536 1382.47100
## 90       90     1685 1389.32834
## 91       91     2227 2121.43277
## 92       92     2252 2438.35179
## 93       93     3249 2889.16665
## 94       94     3115 3067.53611
## 95       95     1795 1912.54648
## 96       96     2808 2879.62804
## 97       97     3141 3104.81434
## 98       98     1471 1403.97224
## 99       99     2455 2678.11536
## 100     100     2895 2924.77488
## 101     101     3348 3097.57941
## 102     102     2034 2159.41311
## 103     103     2162 2251.23300
## 104     104     3267 3345.95839
## 105     105     3126 3952.94572
## 106     106      795 1657.62271
## 107     107     3744 3657.73618
## 108     108     3429 3419.12551
## 109     109     3204 3364.59119
## 110     110     3944 4156.81828
## 111     111     4189 4188.17164
## 112     112     1683 1668.95066
## 113     113     4036 3963.71658
## 114     114     4191 4001.72319
## 115     115     4073 4098.16974
## 116     116     4400 4626.62996
## 117     117     3872 4067.21685
## 118     118     4058 4284.55978
## 119     119     4595 4549.62709
## 120     120     5312 5021.92276
## 121     121     3351 3566.26524
## 122     122     4401 4285.99362
## 123     123     4451 4707.53367
## 124     124     2633 2770.61633
## 125     125     4433 4621.88195
## 126     126     4608 4489.50707
## 127     127     4714 4783.88850
## 128     128     4333 4575.28398
## 129     129     4362 4528.01295
## 130     130     4803 5028.21066
## 131     131     4182 4452.28830
## 132     132     4864 5045.16079
## 133     133     4105 4053.94399
## 134     134     3409 3947.96533
## 135     135     4553 4544.82813
## 136     136     3958 3935.55061
## 137     137     4123 4149.12291
## 138     138     3855 3983.00135
## 139     139     4575 4578.15904
## 140     140     4917 4865.40370
## 141     141     5805 5371.84782
## 142     142     4660 4807.83194
## 143     143     4274 4221.62670
## 144     144     4492 4726.85375
## 145     145     4978 5270.75809
## 146     146     4677 4953.85489
## 147     147     4679 4801.22981
## 148     148     4758 4486.32151
## 149     149     4788 4102.15142
## 150     150     4098 4336.27190
## 151     151     3982 4323.16778
## 152     152     3974 4278.10978
## 153     153     4968 5265.53311
## 154     154     5312 5365.64721
## 155     155     5342 5216.76983
## 156     156     4906 4833.54994
## 157     157     4548 4747.07145
## 158     158     4833 5005.77347
## 159     159     4401 4671.56212
## 160     160     3915 4221.64713
## 161     161     4586 4706.60693
## 162     162     4966 4995.14447
## 163     163     4460 4656.16700
## 164     164     5020 5052.04604
## 165     165     4891 5024.62886
## 166     166     5180 5343.96958
## 167     167     3767 3852.42967
## 168     168     4844 4810.20579
## 169     169     5119 5048.88682
## 170     170     4744 4749.58520
## 171     171     4010 3950.74581
## 172     172     4835 4816.77609
## 173     173     4507 4697.50542
## 174     174     4790 4862.06800
## 175     175     4991 4921.62836
## 176     176     5202 5161.88932
## 177     177     5305 5098.66501
## 178     178     4708 4601.55255
## 179     179     4648 4806.38528
## 180     180     5225 5355.83386
## 181     181     5515 5465.08827
## 182     182     5362 5043.05315
## 183     183     5119 4603.76788
## 184     184     4649 3829.45401
## 185     185     6043 4561.45072
## 186     186     4665 4465.56160
## 187     187     4629 4649.99504
## 188     188     4592 4693.65473
## 189     189     4040 3994.71606
## 190     190     5336 5067.39677
## 191     191     4881 4844.51905
## 192     192     4086 4156.99003
## 193     193     4258 4478.67769
## 194     194     4342 4410.08741
## 195     195     5084 5074.58816
## 196     196     5538 5059.12830
## 197     197     5923 5205.71856
## 198     198     5302 4992.79307
## 199     199     4458 4432.66404
## 200     200     4541 4665.19329
## 201     201     4332 4526.45616
## 202     202     3784 3893.61317
## 203     203     3387 3633.75146
## 204     204     3285 3991.30818
## 205     205     3606 4260.46271
## 206     206     3840 4004.44701
## 207     207     4590 4742.85842
## 208     208     4656 4847.15730
## 209     209     4390 4676.47529
## 210     210     3846 4040.11880
## 211     211     4475 4670.56442
## 212     212     4302 4508.36008
## 213     213     4266 4398.61042
## 214     214     4845 5014.46611
## 215     215     3574 3770.45229
## 216     216     4576 4506.28383
## 217     217     4866 4672.96630
## 218     218     4294 4297.64552
## 219     219     3785 4054.13618
## 220     220     4326 4316.22370
## 221     221     4602 4581.99868
## 222     222     4780 4838.73266
## 223     223     4792 4884.23946
## 224     224     4905 4729.57353
## 225     225     4150 4122.29656
## 226     226     3820 3902.55833
## 227     227     4338 4281.32457
## 228     228     4725 4861.14738
## 229     229     4694 4911.39639
## 230     230     3805 3880.12608
## 231     231     4153 3996.65746
## 232     232     5191 4943.34695
## 233     233     3873 4176.27624
## 234     234     4758 4782.66741
## 235     235     5895 5534.19243
## 236     236     5130 5032.26294
## 237     237     3542 3695.22261
## 238     238     4661 4691.58419
## 239     239     1115 2043.47829
## 240     240     4334 4560.71252
## 241     241     4634 4682.51045
## 242     242     5204 5296.91390
## 243     243     5058 5236.17536
## 244     244     5115 5244.86656
## 245     245     4727 4592.71573
## 246     246     4484 4119.57168
## 247     247     4940 4005.32386
## 248     248     3351 3540.93867
## 249     249     2710 2818.54431
## 250     250     1996 2143.61702
## 251     251     1842 1955.67051
## 252     252     3544 3718.52115
## 253     253     5345 5320.26795
## 254     254     5046 5109.76210
## 255     255     4713 4862.15208
## 256     256     4763 4910.12236
## 257     257     4785 5025.96531
## 258     258     3659 3802.98295
## 259     259     4760 4629.77061
## 260     260     4511 4483.33054
## 261     261     4274 4385.16449
## 262     262     4539 4495.61928
## 263     263     3641 3716.43999
## 264     264     4352 4458.52142
## 265     265     4795 4991.01926
## 266     266     2395 2454.40325
## 267     267     5423 5187.61349
## 268     268     5010 5009.86518
## 269     269     4630 4631.03796
## 270     270     4120 4256.76953
## 271     271     3907 4011.27267
## 272     272     4839 4969.70438
## 273     273     5202 5141.50557
## 274     274     2429 3061.91165
## 275     275     2918 3408.07425
## 276     276     3570 3583.44246
## 277     277     4456 4613.02750
## 278     278     4826 5030.12088
## 279     279     4765 4824.60522
## 280     280     4985 4723.90652
## 281     281     5409 4680.90814
## 282     282     5511 4625.80418
## 283     283     5117 5226.07267
## 284     284     4563 4531.90237
## 285     285     2416 2447.59948
## 286     286     2913 3054.09176
## 287     287     3644 3625.16442
## 288     288     5217 4879.56162
## 289     289     5041 4855.05423
## 290     290     4570 4562.79077
## 291     291     4748 4768.96396
## 292     292     2424 2414.49332
## 293     293     4195 4333.45718
## 294     294     4304 4183.45077
## 295     295     4308 4165.49439
## 296     296     4381 4085.73046
## 297     297     4187 4015.97860
## 298     298     4687 4644.06981
## 299     299     3894 3990.00069
## 300     300     2659 2707.40539
## 301     301     3747 3623.64512
## 302     302      627 1200.01278
## 303     303     3331 3633.61737
## 304     304     3669 3688.41274
## 305     305     4068 3996.35838
## 306     306     4186 4147.99140
## 307     307     3974 3998.08039
## 308     308     4046 3855.04286
## 309     309     3926 3871.53367
## 310     310     3649 3763.23302
## 311     311     4035 3978.65320
## 312     312     4205 4195.38736
## 313     313     4109 4149.48659
## 314     314     2933 2818.74511
## 315     315     3368 4068.61310
## 316     316     4067 3921.88812
## 317     317     3717 3939.97855
## 318     318     4486 4426.32618
## 319     319     4195 4173.88870
## 320     320     1817 1566.16709
## 321     321     3053 3028.18299
## 322     322     3392 3324.95875
## 323     323     3663 3819.63363
## 324     324     3520 3876.92144
## 325     325     2765 2577.62832
## 326     326     1607 1360.42228
## 327     327     2566 2558.89907
## 328     328     1495 1777.30379
## 329     329     2792 1693.10936
## 330     330     3068 2781.19361
## 331     331     3071 3381.34686
## 332     332     3867 4041.80226
## 333     333     2914 2985.87831
## 334     334     3613 3659.23247
## 335     335     3727 3649.33739
## 336     336     3940 3777.25881
## 337     337     3614 3827.32565
## 338     338     3485 3759.62802
## 339     339     3811 3639.21892
## 340     340     2594 2339.95535
## 341     341      705  167.57427
## 342     342     3322 3153.14576
## 343     343     3620 3372.70280
## 344     344     3190 3567.97409
## 345     345     2743 3138.88872
## 346     346     3310 3098.44271
## 347     347     3523 3397.08160
## 348     348     3740 3582.86880
## 349     349     3709 3661.60399
## 350     350     3577 3489.47203
## 351     351     2739 3216.14229
## 352     352     2431 2956.27742
## 353     353     3403 3118.54068
## 354     354     3750 3637.83246
## 355     355     2660 2637.71344
## 356     356     3068 2989.56912
## 357     357     2209 2087.16057
## 358     358     1011 1609.61488
## 359     359      754 1078.40615
## 360     360     1317 1673.89338
## 361     361     1162  807.83074
## 362     362     2302 2048.44419
## 363     363     2423 2121.26535
## 364     364     2999 2570.24269
## 365     365     2485 2831.40329
## 366     366     2294 2500.85678
## 367     367     1951 2579.77051
## 368     368     2236 1994.95197
## 369     369     2368 2040.49688
## 370     370     3272 3248.02600
## 371     371     4098 4084.69094
## 372     372     4521 4707.21588
## 373     373     3425 3933.98172
## 374     374     2376 2099.61989
## 375     375     3598 3608.70463
## 376     376     2177 1906.04762
## 377     377     4097 4061.54067
## 378     378     3214 3153.33307
## 379     379     2493 2991.83325
## 380     380     2311 2830.93854
## 381     381     2298 2896.48083
## 382     382     2935 2881.97868
## 383     383     3376 3462.67392
## 384     384     3292 3207.84348
## 385     385     3163 3052.95251
## 386     386     1301 1763.95436
## 387     387     1977 2397.12007
## 388     388     2432 2068.08695
## 389     389     4339 4157.31630
## 390     390     4270 4034.12849
## 391     391     4075 4028.66604
## 392     392     3456 3373.45538
## 393     393     4023 4385.58940
## 394     394     3243 3744.17841
## 395     395     3624 3704.47253
## 396     396     4509 4630.18735
## 397     397     4579 4763.56119
## 398     398     3761 3771.67972
## 399     399     4151 4087.78877
## 400     400     2832 3249.84823
## 401     401     2947 3482.19820
## 402     402     3784 3728.10495
## 403     403     4375 4528.11353
## 404     404     2802 2586.19753
## 405     405     3830 3825.93507
## 406     406     3831 3712.53951
## 407     407     2169 2607.29532
## 408     408     1529 2078.00686
## 409     409     3422 3418.18961
## 410     410     3922 3973.31727
## 411     411     4169 4341.95728
## 412     412     3005 2920.96343
## 413     413     4154 4054.61594
## 414     414     4318 3958.93027
## 415     415     2689 2884.35070
## 416     416     3129 3595.38889
## 417     417     3777 3781.59348
## 418     418     4773 4793.58830
## 419     419     5062 5053.19974
## 420     420     3487 3382.25689
## 421     421     2732 3384.09610
## 422     422     3389 3903.28133
## 423     423     4322 4417.96854
## 424     424     4363 4506.13652
## 425     425     1834 1585.73874
## 426     426     4990 5540.89293
## 427     427     3194 3328.89584
## 428     428     4066 4543.43568
## 429     429     3423 4095.33703
## 430     430     3333 3517.71201
## 431     431     3956 4249.12782
## 432     432     4916 5264.93647
## 433     433     5382 5864.72649
## 434     434     4569 4811.58649
## 435     435     4118 4574.89734
## 436     436     4911 4746.80936
## 437     437     5298 5308.83199
## 438     438     5847 6114.11441
## 439     439     6312 6413.80075
## 440     440     6192 6233.09505
## 441     441     4378 4404.46752
## 442     442     7836 6451.11783
## 443     443     5892 5234.81307
## 444     444     6153 6169.01771
## 445     445     6093 6018.59258
## 446     446     6230 5948.55741
## 447     447     6871 6531.77510
## 448     448     8362 6964.98478
## 449     449     3372 3631.35866
## 450     450     4996 4873.50806
## 451     451     5558 5590.69478
## 452     452     5102 5276.85525
## 453     453     5698 5934.88817
## 454     454     6133 6274.95028
## 455     455     5459 5309.53870
## 456     456     6235 5437.45292
## 457     457     6041 5277.49836
## 458     458     5936 5635.99712
## 459     459     6772 6504.85517
## 460     460     6436 6515.55601
## 461     461     6457 6290.71414
## 462     462     6460 5543.41584
## 463     463     6857 5338.76989
## 464     464     5169 4624.95294
## 465     465     5585 5679.22708
## 466     466     5918 6096.69268
## 467     467     4862 5147.23346
## 468     468     5409 5626.33456
## 469     469     6398 6152.73813
## 470     470     7460 6581.30564
## 471     471     7132 6257.82605
## 472     472     6370 7339.64956
## 473     473     6691 6981.10823
## 474     474     4367 4767.04315
## 475     475     6565 6835.80113
## 476     476     7290 7117.43281
## 477     477     6624 5944.58532
## 478     478     1027 1903.99318
## 479     479     3214 3407.42081
## 480     480     5633 6075.65712
## 481     481     6196 6645.08229
## 482     482     5026 5382.55281
## 483     483     6233 6255.38422
## 484     484     4220 4573.31582
## 485     485     6304 5836.81278
## 486     486     5572 5800.99325
## 487     487     5740 6155.32482
## 488     488     6169 6618.36381
## 489     489     6421 6739.01696
## 490     490     6296 6378.83733
## 491     491     6883 6281.14861
## 492     492     6359 6031.76628
## 493     493     6273 6280.81610
## 494     494     5728 6002.91302
## 495     495     4717 4943.01720
## 496     496     6572 6660.14835
## 497     497     7030 6922.93573
## 498     498     7429 6830.18729
## 499     499     6118 5858.14781
## 500     500     2843 3072.33024
## 501     501     5115 5429.06081
## 502     502     7424 7801.51892
## 503     503     7384 7454.62789
## 504     504     7639 7394.85070
## 505     505     8294 6963.29703
## 506     506     7129 6434.87299
## 507     507     4359 4524.78727
## 508     508     6073 6328.76477
## 509     509     5260 5448.68952
## 510     510     6770 6975.99688
## 511     511     6734 6537.15088
## 512     512     6536 5601.13580
## 513     513     6591 5171.59691
## 514     514     6043 5407.42247
## 515     515     5743 6067.07565
## 516     516     6855 7390.95660
## 517     517     7338 7664.70763
## 518     518     4127 4298.87070
## 519     519     8120 7299.01060
## 520     520     7641 7124.12220
## 521     521     6998 7075.49075
## 522     522     7001 7049.71393
## 523     523     7055 7120.20413
## 524     524     7494 7679.81862
## 525     525     7736 7508.63786
## 526     526     7498 6835.89887
## 527     527     6598 6346.40347
## 528     528     6664 6797.49835
## 529     529     4972 5330.75752
## 530     530     7421 7491.17145
## 531     531     7363 7410.14753
## 532     532     7665 7301.71319
## 533     533     7702 6677.62690
## 534     534     6978 6155.37541
## 535     535     5099 5194.96291
## 536     536     6825 7047.13544
## 537     537     6211 6565.23038
## 538     538     5905 6297.06523
## 539     539     5823 5953.40983
## 540     540     7458 6798.64534
## 541     541     6891 6286.00952
## 542     542     6779 6812.38969
## 543     543     7442 7594.43905
## 544     544     7335 7543.76216
## 545     545     6879 7235.94554
## 546     546     5463 5770.04102
## 547     547     5687 6157.03009
## 548     548     5531 6062.46959
## 549     549     6227 6501.20566
## 550     550     6660 6822.61774
## 551     551     7403 6881.24785
## 552     552     6241 5980.45836
## 553     553     6207 5988.69591
## 554     554     4840 5280.71066
## 555     555     4672 5311.05825
## 556     556     6569 6602.05431
## 557     557     6290 6342.38202
## 558     558     7264 7517.06223
## 559     559     7446 7684.11460
## 560     560     7499 7152.53615
## 561     561     6969 6429.11057
## 562     562     6031 5946.11372
## 563     563     6830 6927.13171
## 564     564     6786 7159.01993
## 565     565     5713 6009.23091
## 566     566     6591 6898.40833
## 567     567     5870 5997.58330
## 568     568     4459 4564.08742
## 569     569     7410 6670.81241
## 570     570     6966 6997.11681
## 571     571     7592 7738.70700
## 572     572     8173 8148.12415
## 573     573     6861 7048.43947
## 574     574     6904 6844.54344
## 575     575     6685 6375.34621
## 576     576     6597 6328.51489
## 577     577     7105 7263.61422
## 578     578     7216 7450.49959
## 579     579     7580 7772.08044
## 580     580     7261 7547.67103
## 581     581     7175 7001.86087
## 582     582     6824 6458.81318
## 583     583     5464 5583.57497
## 584     584     7013 6907.08473
## 585     585     7273 7129.08493
## 586     586     7534 7472.33399
## 587     587     7286 7340.66052
## 588     588     5786 5621.11096
## 589     589     6299 5766.49638
## 590     590     6544 6241.72891
## 591     591     6883 6837.22422
## 592     592     6784 6789.74311
## 593     593     7347 7357.13639
## 594     594     7605 7529.53648
## 595     595     7148 6821.60568
## 596     596     7865 6989.49140
## 597     597     4549 4882.41764
## 598     598     6530 6443.94530
## 599     599     7006 7023.22712
## 600     600     7375 7456.75397
## 601     601     7765 7636.79719
## 602     602     7582 7417.83850
## 603     603     6053 5938.27869
## 604     604     5255 5374.25867
## 605     605     6917 7072.28678
## 606     606     7040 7328.89712
## 607     607     7697 7780.72732
## 608     608     7713 7820.04377
## 609     609     7350 7155.11102
## 610     610     6140 5607.81665
## 611     611     5810 4820.92030
## 612     612     6034 5915.08848
## 613     613     6864 7234.97868
## 614     614     7112 7579.15025
## 615     615     6203 6658.52771
## 616     616     7504 7753.19070
## 617     617     5976 6207.80978
## 618     618     8227 7713.30713
## 619     619     7525 7614.69587
## 620     620     7767 7942.78998
## 621     621     7870 8096.88736
## 622     622     7804 8016.64940
## 623     623     8009 7891.45765
## 624     624     8714 7601.41368
## 625     625     7333 7105.68512
## 626     626     6869 6848.36721
## 627     627     4073 4365.35492
## 628     628     7591 8036.01835
## 629     629     7720 7986.59163
## 630     630     8167 8190.14777
## 631     631     8395 7997.87771
## 632     632     7907 7355.48595
## 633     633     7436 7524.58792
## 634     634     7538 7846.74437
## 635     635     7733 8219.00649
## 636     636     7393 7806.31987
## 637     637     7415 7456.77388
## 638     638     8555 7950.95903
## 639     639     6889 6653.33630
## 640     640     6778 6985.73639
## 641     641     4639 4981.04765
## 642     642     7572 8066.03412
## 643     643     7328 7607.17921
## 644     644     8156 7880.51148
## 645     645     7965 6773.72065
## 646     646     3510 3986.50725
## 647     647     5478 6135.71188
## 648     648     6392 6621.49776
## 649     649     7691 8087.94480
## 650     650     7570 7835.72381
## 651     651     7282 7223.32261
## 652     652     7109 6549.01522
## 653     653     6639 6307.34837
## 654     654     5875 5968.34214
## 655     655     7534 7706.37699
## 656     656     7461 7512.64353
## 657     657     7509 7536.60003
## 658     658     5424 5429.77614
## 659     659     8090 7126.41898
## 660     660     6824 6429.03091
## 661     661     7058 7277.35403
## 662     662     7466 7784.45201
## 663     663     7693 8149.83832
## 664     664     7359 7530.60682
## 665     665     7444 7266.12697
## 666     666     7852 6994.62296
## 667     667     4459 4907.91433
## 668     668       22 -195.19768
## 669     669     1096  900.33936
## 670     670     5566 5796.68231
## 671     671     5986 6088.35019
## 672     672     5847 5812.29311
## 673     673     5138 5393.43051
## 674     674     5107 5190.97725
## 675     675     5259 5374.71475
## 676     676     5686 5713.03042
## 677     677     5035 5076.34647
## 678     678     5315 5554.00100
## 679     679     5992 5878.26977
## 680     680     6536 5863.33066
## 681     681     6852 6029.26470
## 682     682     6269 6790.50508
## 683     683     4094 4004.84857
## 684     684     5495 5610.98943
## 685     685     5445 5572.78620
## 686     686     5698 5783.61783
## 687     687     5629 5664.99506
## 688     688     4669 4989.09548
## 689     689     5499 5547.88831
## 690     690     5634 5585.74887
## 691     691     5146 4977.70387
## 692     692     2425 2358.23608
## 693     693     3910 2412.71486
## 694     694     2277 2650.66284
## 695     695     2424 3033.28396
## 696     696     5087 5207.01594
## 697     697     3959 4003.48949
## 698     698     5260 5558.96116
## 699     699     5323 5552.44863
## 700     700     5668 5814.11266
## 701     701     5191 5274.69078
## 702     702     4649 4760.90298
## 703     703     6234 6232.17817
## 704     704     6606 6702.31687
## 705     705     5729 5950.55204
## 706     706     5375 5327.97229
## 707     707     5008 4842.67043
## 708     708     5582 5561.20704
## 709     709     3228 3652.41150
## 710     710     5170 5142.69323
## 711     711     5501 5565.71172
## 712     712     5319 5274.10503
## 713     713     5532 5459.43233
## 714     714     5611 5497.87737
## 715     715     5047 5444.11585
## 716     716     3786 4180.66145
## 717     717     4585 4556.06833
## 718     718     5557 5563.03592
## 719     719     5267 5266.96523
## 720     720     4128 3888.84925
## 721     721     3623 3583.20684
## 722     722     1749 2393.73855
## 723     723     1787 2164.24178
## 724     724      920  337.10789
## 725     725     1013 1164.98002
## 726     726      441  -84.37476
## 727     727     2114 1697.09770
## 728     728     3095 2390.03598
## 729     729     1341 1829.99003
## 730     730     1796 2243.55031
## 731     731     2729 2165.19227
```

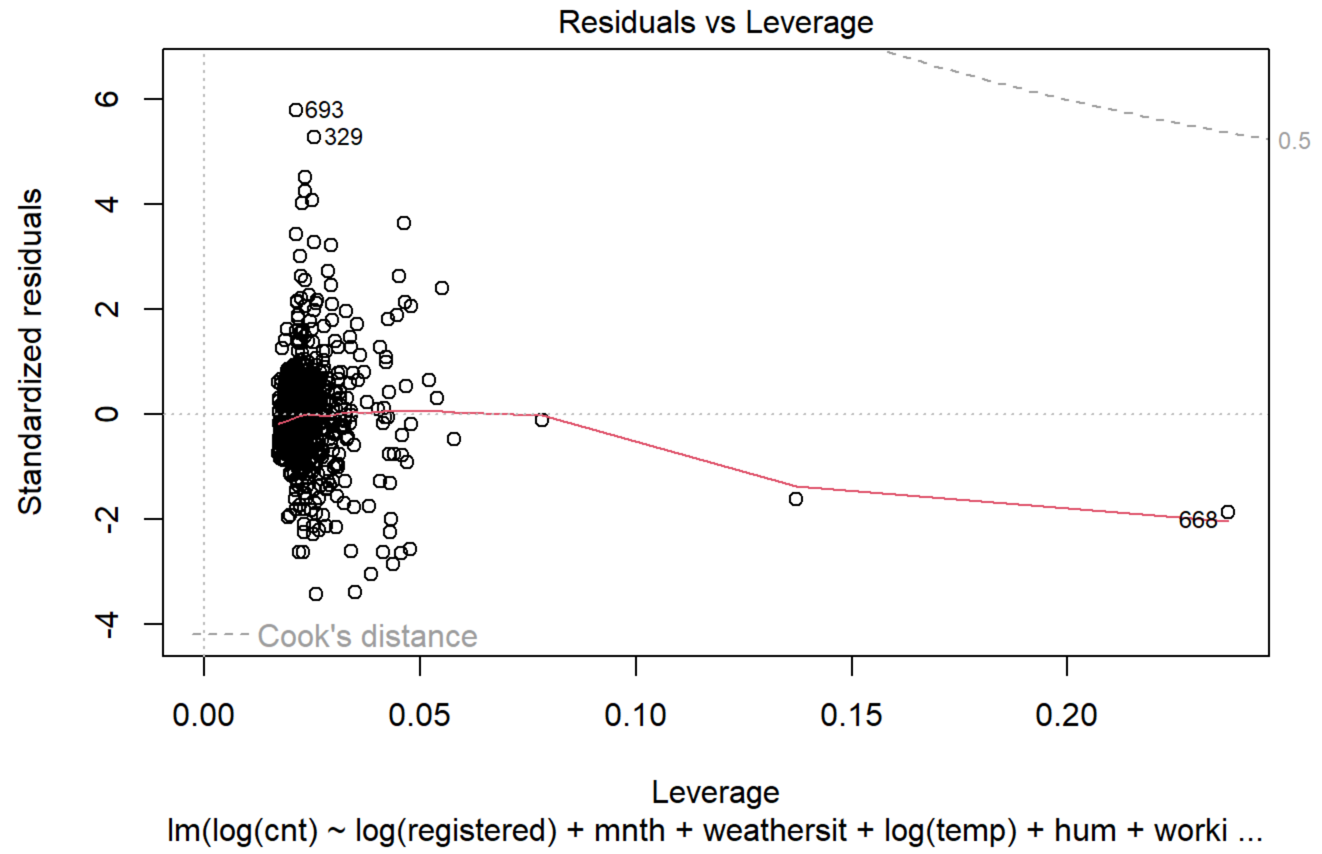
##log t

```
day_model_logged = lm(log(cnt) ~log(registered)+ mnth +weathersit + log(temp) +hum+workingday+se
ason, data = day)

plot(day_model_logged)
```


Residuals vs Fitted
Residuals
0.4
0.2
0.0
-0.2
693
329
185
3
4
5
6
7
8
9
Fitted values
lm(log(cnt) ~ log(registered) + mnth + weathersit + log(temp) + hum + worki ...


Q-Q Residuals
Standardized residuals
6
4
2
0
-2
693
329
185
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(log(cnt) ~ log(registered) + mnth + weathersit + log(temp) + hum + worki ...

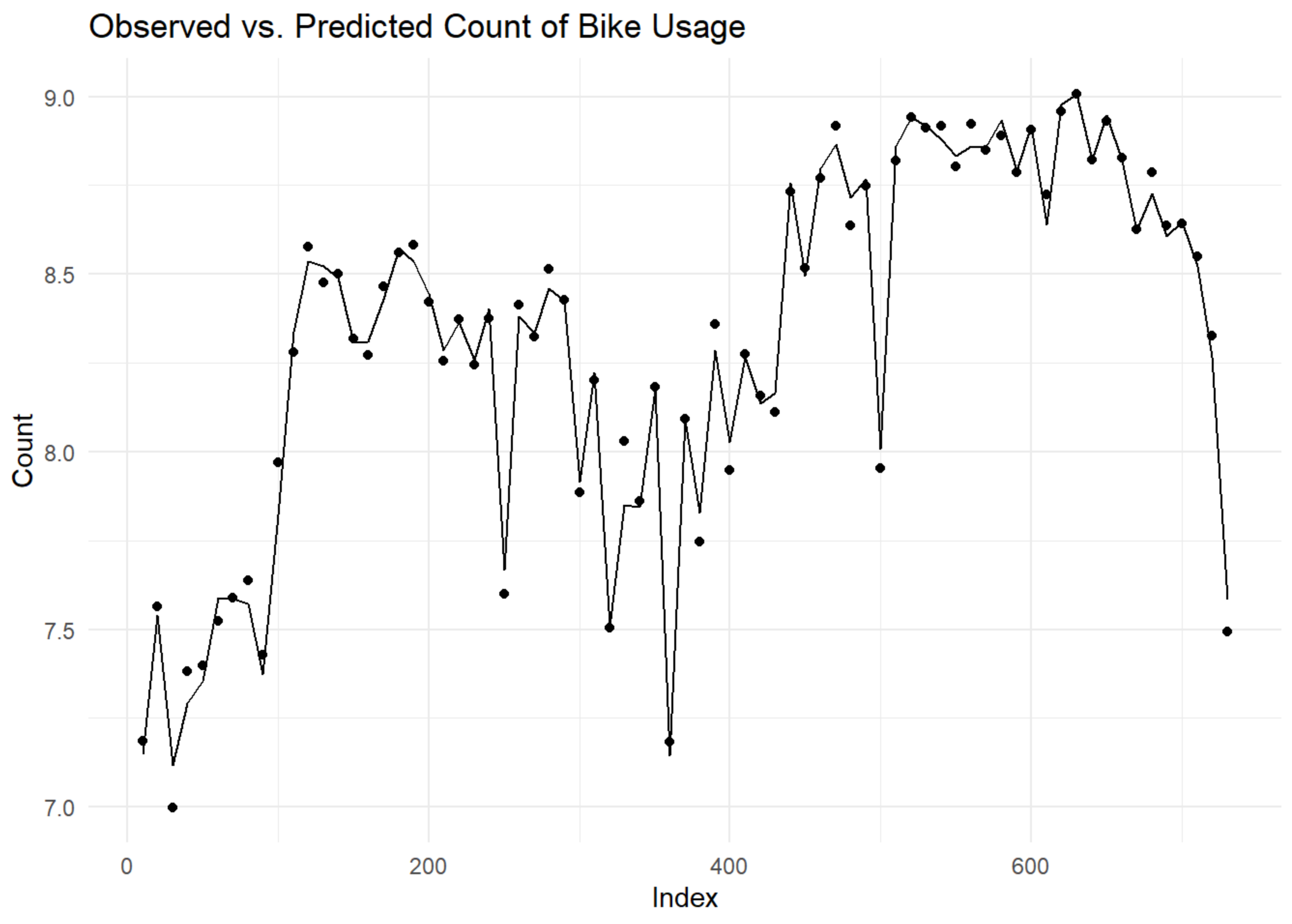

Scale-Location
693
329
185
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3
4
5
6
7
8
9
Fitted values
lm(log(cnt) ~ log(registered) + mnth + weathersit + log(temp) + hum + worki ...


Residuals vs Leverage
693
329
668
0.5
Cook's distance
Standardized residuals
-4
-2
0
2
4
6
0.00
0.05
0.10
0.15
0.20
Leverage
lm(log(cnt) ~ log(registered) + mnth + weathersit + log(temp) + hum + worki ...

```
predicted_values <- predict(day_model_logged, newdata = day)
RMSE(predicted_values, log(day$cnt))
```

```
## [1] 0.06987392
```

```
comparison_data <- data.frame(
  Instant = day$instant,
  Observed = log(day$cnt),
  Predicted = predicted_values
)
filtered_data <- comparison_data %>% filter(row_number() %% 10 == 0)

ggplot(filtered_data, aes(x = Instant, y = Observed)) +
  geom_point() +
  geom_line(aes(x = Instant, y = Predicted)) +
  labs(x = "Index", y = "Count", title = "Observed vs. Predicted Count of Bike Usage") +
  theme_minimal()
```



```
selected_years <- c(1, 448, 185, 463, 668, 669, 693)

subset(day, instant %in% selected_years , select = c('dteday', 'workingday','weathersit', 'tem
p', 'hum', 'windspeed','holiday', 'registered', 'cnt')  )
```

```
##         dteday workingday weathersit     temp      hum windspeed holiday
## 1   2011-01-01          0          2 0.344167 0.805833 0.1604460       0
## 185 2011-07-04          0          2 0.726667 0.637917 0.0814792       1
## 448 2012-03-23          1          2 0.601667 0.694167 0.1163000       0
## 463 2012-04-07          0          1 0.437500 0.254167 0.2748710       0
## 668 2012-10-29          1          3 0.440000 0.880000 0.3582000       0
## 669 2012-10-30          1          2 0.318182 0.825455 0.2130090       0
## 693 2012-11-23          1          1 0.368333 0.568750 0.1480210       0
##     registered  cnt
## 1          654  985
## 185       2978 6043
## 448       5893 8362
## 463       3605 6857
## 668         20   22
## 669       1009 1096
## 693       2307 3910
```

```
smallest_cnt <- day[which.min(day$cnt), ]
largest_cnt <- day[which.max(day$cnt), ]
largest_cnt
```

```
##     instant     dteday season yr mnth holiday weekday workingday weathersit
## 624     624 2012-09-15      3  1    9       0       6          0          1
##         temp    atemp      hum windspeed casual registered  cnt
## 624 0.608333 0.585867 0.501667  0.247521   3160       5554 8714
```

```
smallest_cnt
```

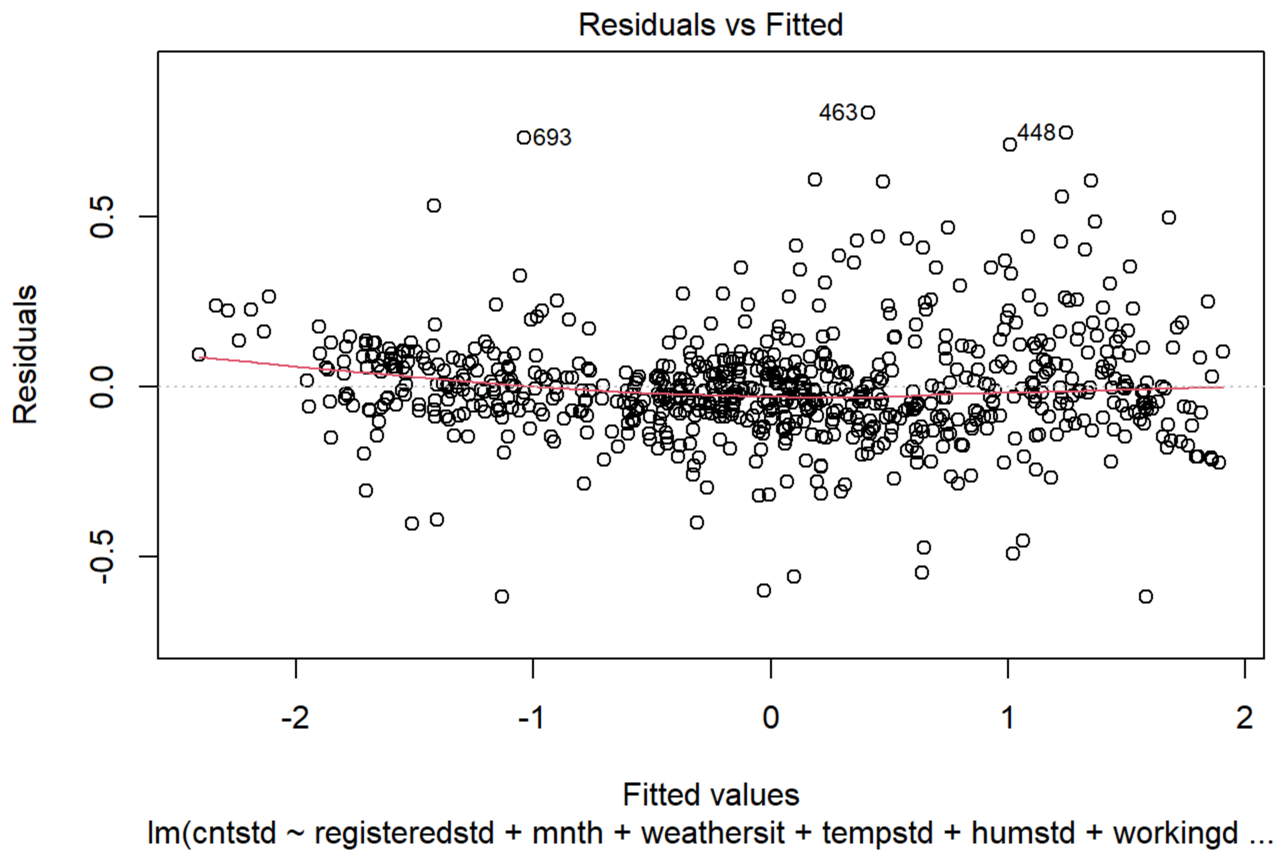
```
##     instant     dteday season yr mnth holiday weekday workingday weathersit
## 668     668 2012-10-29      4  1   10       0       1          1          3
##     temp  atemp  hum windspeed casual registered cnt
## 668 0.44 0.4394 0.88    0.3582      2         20  22
```

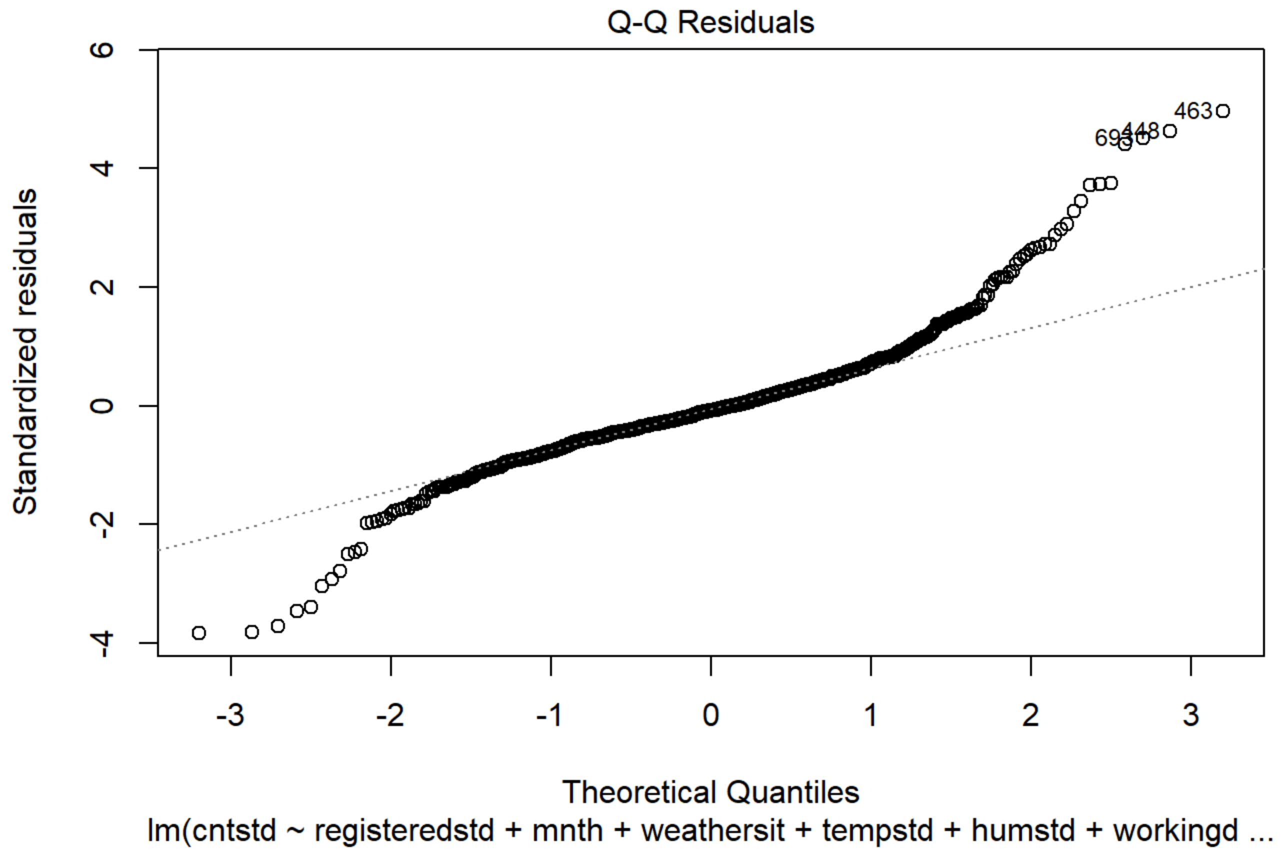
##standardize

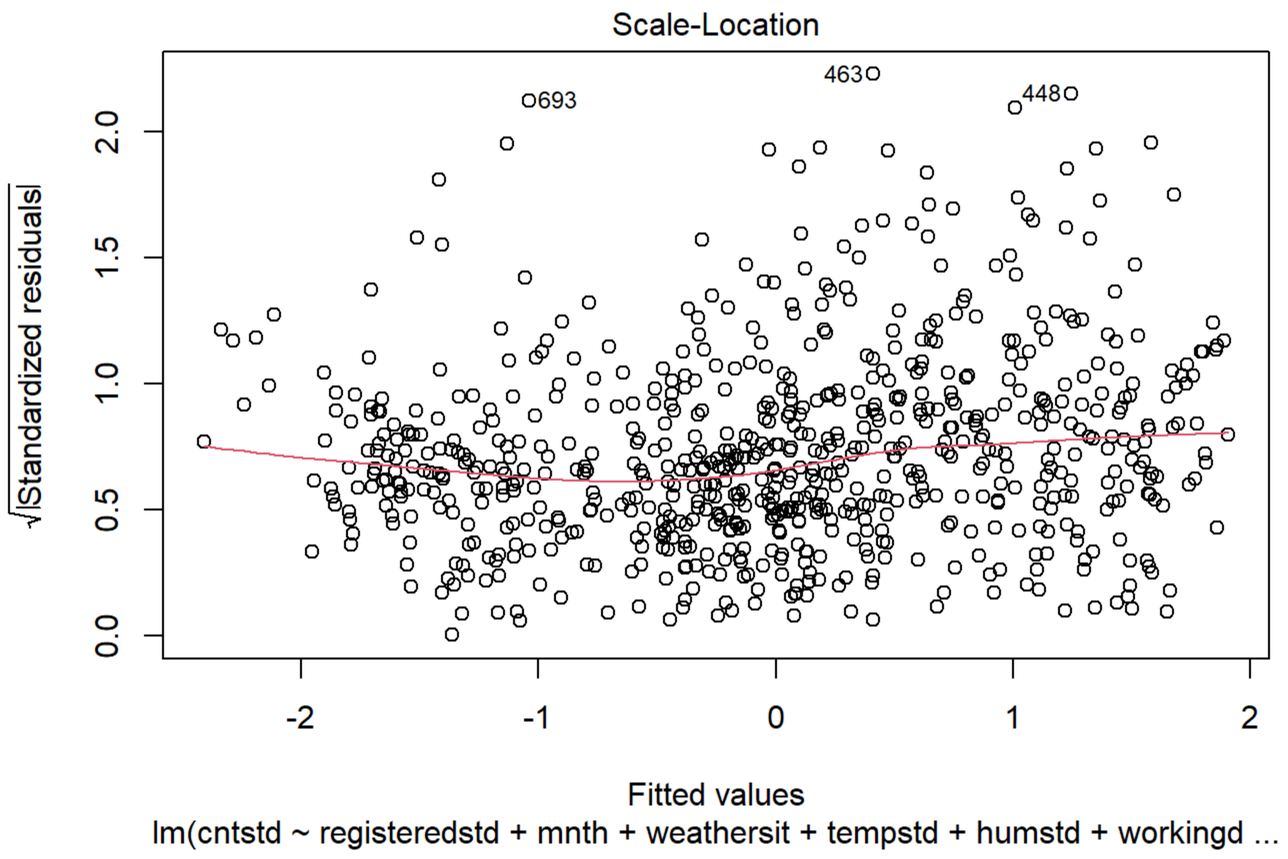
```
data1$tempstd = (day$temp - mean(day$temp))/sd(day$temp)
data1$humstd = (day$hum - mean(day$hum))/sd(day$hum)
data1$registeredstd = (day$registered - mean(day$registered))/sd(day$registered)
data1$cntstd = (day$cnt - mean(day$cnt))/sd(day$cnt)

day_model_standardized = lm(cntstd ~registeredstd+ mnth +weathersit + tempstd +humstd+workingday
+ (workingday * tempstd), data = data1)
summary(day_model_standardized)
```

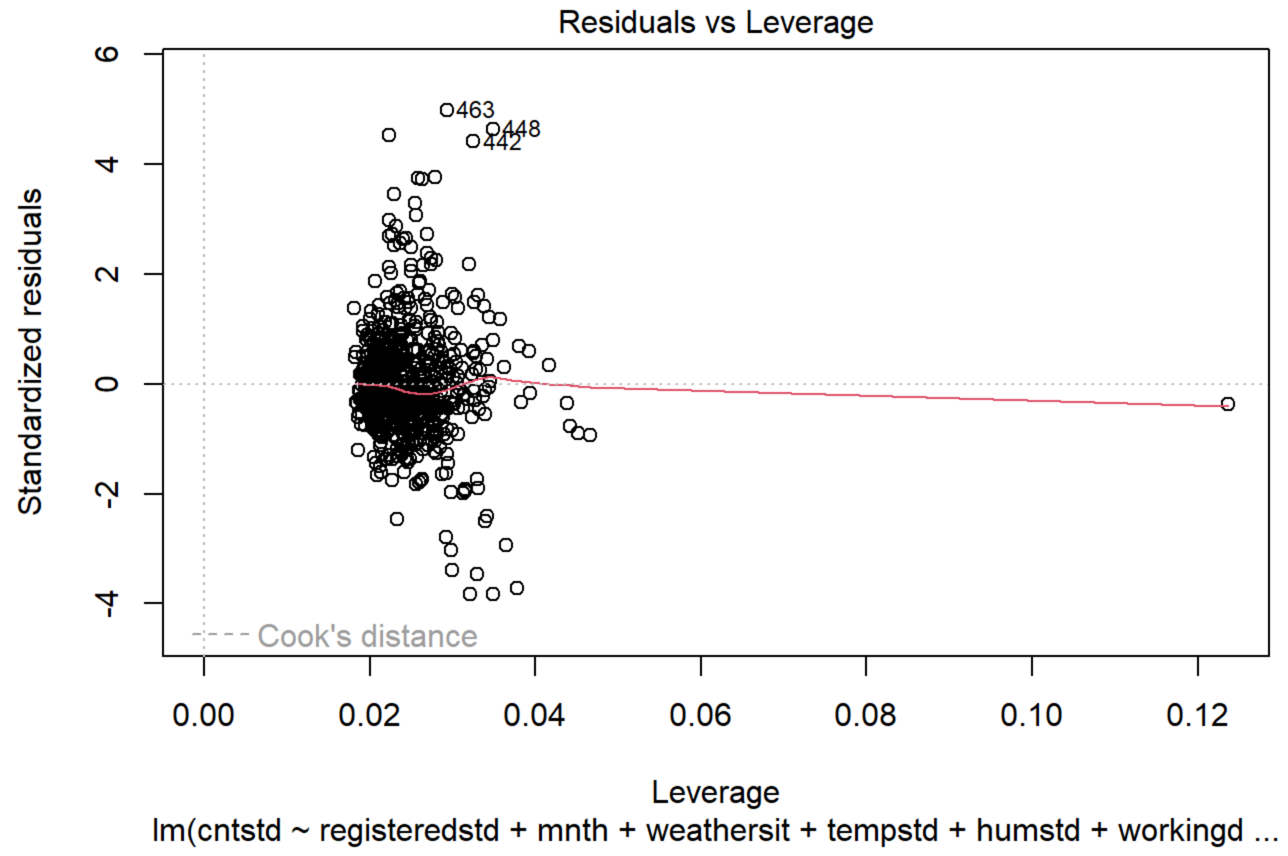
```
##
## Call:
## lm(formula = cntstd ~ registeredstd + mnth + weathersit + tempstd +
##     humstd + workingday + (workingday * tempstd), data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61938 -0.08355 -0.01261  0.06718  0.80573
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.314295   0.036060   8.716  < 2e-16 ***
## registeredstd       0.940779   0.008584 109.596  < 2e-16 ***
## mnth2              -0.022698   0.030666  -0.740  0.45944
## mnth3               0.132168   0.032356   4.085 4.91e-05 ***
## mnth4               0.176715   0.035853   4.929 1.03e-06 ***
## mnth5               0.179839   0.041988   4.283 2.10e-05 ***
## mnth6               0.109763   0.048166   2.279  0.02297 *
## mnth7               0.070107   0.052699   1.330  0.18384
## mnth8               0.094685   0.049255   1.922  0.05496 .
## mnth9               0.098312   0.043786   2.245  0.02505 *
## mnth10              0.108243   0.036785   2.943  0.00336 **
## mnth11              0.008736   0.032246   0.271  0.78654
## mnth12             -0.036053   0.030845  -1.169  0.24286
## weathersit         -0.031297   0.014754  -2.121  0.03425 *
## tempstd             0.217065   0.018343  11.834  < 2e-16 ***
## humstd             -0.018486   0.008445  -2.189  0.02893 *
## workingday         -0.501883   0.014252 -35.214  < 2e-16 ***
## tempstd:workingday -0.181194   0.013012 -13.925  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1644 on 713 degrees of freedom
## Multiple R-squared:  0.9736, Adjusted R-squared:  0.973
## F-statistic:  1547 on 17 and 713 DF,  p-value: < 2.2e-16
```

```
plot(day_model_standardized)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cntstd ~ registeredstd + mnth + weathersit + tempstd + humstd + workingd ...
693
463
448
0.5
0.0
-0.5
-2
-1
0
1
2


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(cntstd ~ registeredstd + mnth + weathersit + tempstd + humstd + workingd ...
463
448
693
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

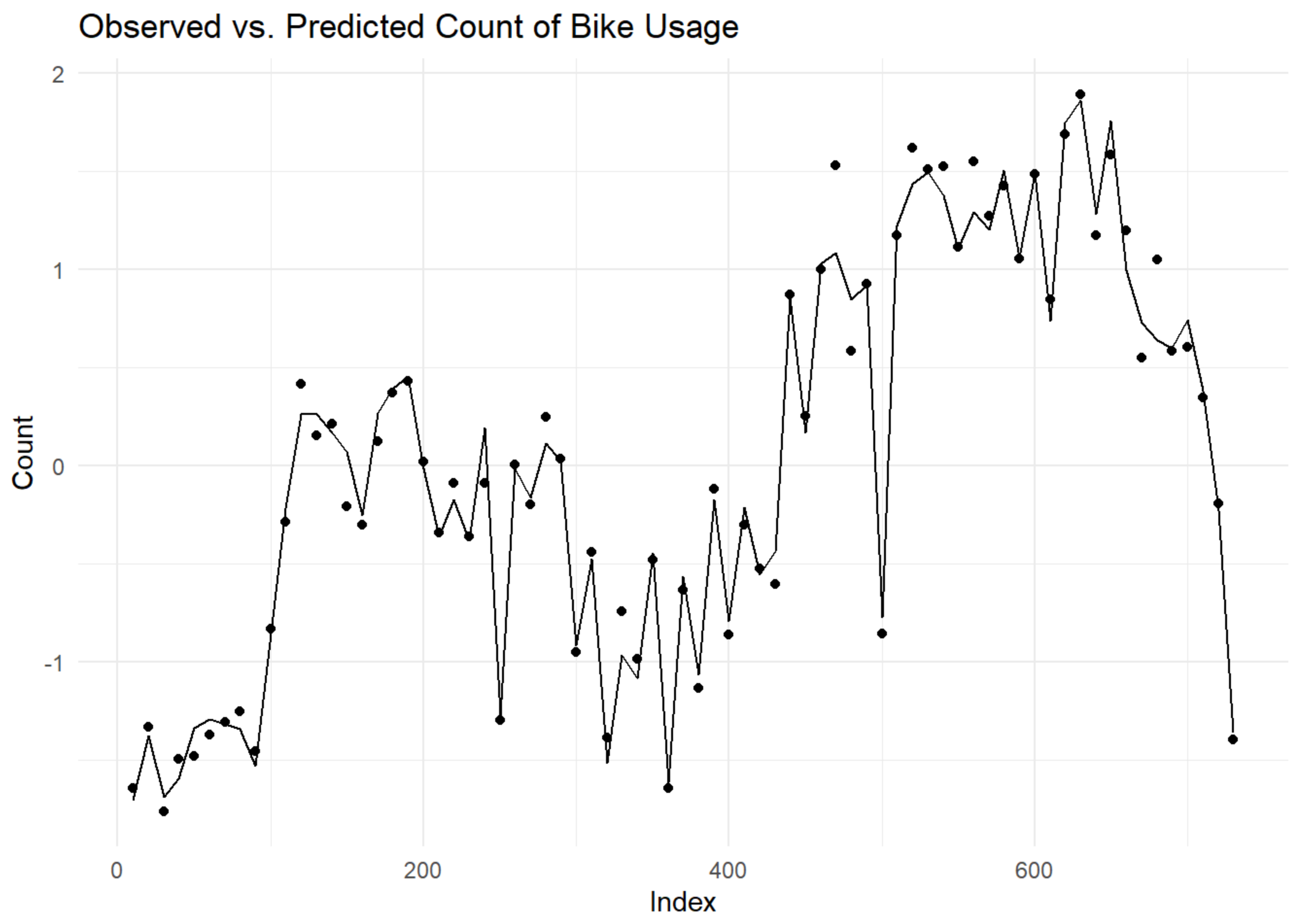

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
693
463
448
-2
-1
0
1
2
Fitted values
lm(cntstd ~ registeredstd + mnth + weathersit + tempstd + humstd + workingd ...


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
6
463
448
442
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
lm(cntstd ~ registeredstd + mnth + weathersit + tempstd + humstd + workingd ...

```
predicted_values <- predict(day_model_standardized, newdata = data1)
RMSE(predicted_values, data1$cntstd)
```

```
## [1] 0.1623359
```

```
comparison_data <- data.frame(
  Instant = day$instant,
  Observed = data1$cntstd,
  Predicted = predicted_values
)
filtered_data <- comparison_data %>% filter(row_number() %% 10 == 0)

ggplot(filtered_data, aes(x = Instant, y = Observed)) +
  geom_point() +
  geom_line(aes(x = Instant, y = Predicted)) +
  labs(x = "Index", y = "Count", title = "Observed vs. Predicted Count of Bike Usage") +
  theme_minimal()
```



```
names(day)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"      
##  [6] "holiday"    "weekday"    "workingday" "weathersit" "temp"      
## [11] "atemp"      "hum"        "windspeed"  "casual"     "registered"
## [16] "cnt"
```

```
bday <- regsubsets(cntstd~registeredstd+season+Weather1+Weather2+Weather3+Weather4+holiday+yr+mo
nth1+month2+month3+month4+month5+month6+month7+month8+month9+month10+month11+month12+Weekday1+We
ekday2+Weekday3+Weekday4+Weekday5+Weekday6+Weekday0+tempstd+windspeed+humstd+workingday + Season
1+Season2+Season3+Season4+(workingday * tempstd)+(humstd * tempstd), data1)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 7 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(bday)
```

```
## Subset selection object
## Call: regsubsets.formula(cntstd ~ registeredstd + season + Weather1 +
##     Weather2 + Weather3 + Weather4 + holiday + yr + month1 +
##     month2 + month3 + month4 + month5 + month6 + month7 + month8 +
##     month9 + month10 + month11 + month12 + Weekday1 + Weekday2 +
##     Weekday3 + Weekday4 + Weekday5 + Weekday6 + Weekday0 + tempstd +
##     windspeed + humstd + workingday + Season1 + Season2 + Season3 +
##     Season4 + (workingday * tempstd) + (humstd * tempstd), data1)
## 37 Variables  (and intercept)
##                     Forced in Forced out
## registeredstd           FALSE      FALSE
## season                  FALSE      FALSE
## Weather1                FALSE      FALSE
## Weather2                FALSE      FALSE
## holiday                 FALSE      FALSE
## yr                      FALSE      FALSE
## month1                  FALSE      FALSE
## month2                  FALSE      FALSE
## month3                  FALSE      FALSE
## month4                  FALSE      FALSE
## month5                  FALSE      FALSE
## month6                  FALSE      FALSE
## month7                  FALSE      FALSE
## month8                  FALSE      FALSE
## month9                  FALSE      FALSE
## month10                 FALSE      FALSE
## month11                 FALSE      FALSE
## Weekday1                FALSE      FALSE
## Weekday2                FALSE      FALSE
## Weekday3                FALSE      FALSE
## Weekday4                FALSE      FALSE
## Weekday5                FALSE      FALSE
## Weekday6                FALSE      FALSE
## tempstd                 FALSE      FALSE
## windspeed               FALSE      FALSE
## humstd                  FALSE      FALSE
## Season1                 FALSE      FALSE
## Season2                 FALSE      FALSE
## tempstd:workingday      FALSE      FALSE
## tempstd:humstd          FALSE      FALSE
## Weather3                FALSE      FALSE
## Weather4                FALSE      FALSE
## month12                 FALSE      FALSE
## Weekday0                FALSE      FALSE
## workingday              FALSE      FALSE
## Season3                 FALSE      FALSE
## Season4                 FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##          registeredstd season Weather1 Weather2 Weather3 Weather4 holiday yr
## 1  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 2  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 3  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 4  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 5  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 6  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 7  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 8  ( 1 ) "*"           " "    " "      " "      " "      " "      " "     " "
## 9  ( 1 ) "*"           " "    " "      " "      " "      " "      "*"     " "
##           month1 month2 month3 month4 month5 month6 month7 month8 month9 month10
## 1  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 2  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 3  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 4  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 5  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 6  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 7  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 8  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
## 9  ( 1 ) " "    " "    " "    " "    " "    " "    " "    " "    " "    " "
##           month11 month12 Weekday1 Weekday2 Weekday3 Weekday4 Weekday5 Weekday6
## 1  ( 1 ) " "     " "     " "      " "      " "      " "      " "      " "
## 2  ( 1 ) " "     " "     " "      " "      " "      " "      " "      " "
## 3  ( 1 ) " "     " "     " "      " "      " "      " "      " "      " "
## 4  ( 1 ) " "     " "     " "      " "      " "      " "      " "      " "
## 5  ( 1 ) " "     " "     " "      " "      " "      " "      " "      " "
## 6  ( 1 ) " "     " "     " "      " "      " "      " "      "*"      " "
## 7  ( 1 ) " "     " "     " "      " "      " "      " "      "*"      " "
## 8  ( 1 ) " "     " "     " "      " "      " "      " "      "*"      " "
## 9  ( 1 ) " "     " "     " "      " "      " "      " "      "*"      " "
##           Weekday0 tempstd windspeed humstd workingday Season1 Season2 Season3
## 1  ( 1 ) " "      " "     " "       " "    " "        " "     " "     " "
## 2  ( 1 ) " "      " "     " "       " "    "*"        " "     " "     " "
## 3  ( 1 ) " "      "*"     " "       " "    "*"        " "     " "     " "
## 4  ( 1 ) " "      "*"     " "       " "    "*"        " "     " "     " "
## 5  ( 1 ) " "      "*"     " "       " "    "*"        " "     "*"     " "
## 6  ( 1 ) " "      "*"     " "       " "    "*"        " "     "*"     " "
## 7  ( 1 ) " "      "*"     " "       "*"    "*"        " "     "*"     " "
## 8  ( 1 ) " "      "*"     "*"       "*"    "*"        " "     "*"     " "
## 9  ( 1 ) " "      "*"     "*"       "*"    "*"        " "     "*"     " "
##           Season4 tempstd:workingday tempstd:humstd
## 1  ( 1 ) " "     " "                 " "
## 2  ( 1 ) " "     " "                 " "
## 3  ( 1 ) " "     " "                 " "
## 4  ( 1 ) " "     "*"                 " "
## 5  ( 1 ) " "     "*"                 " "
## 6  ( 1 ) " "     "*"                 " "
## 7  ( 1 ) " "     "*"                 " "
## 8  ( 1 ) " "     "*"                 " "
## 9  ( 1 ) " "     "*"                 " "
```

```
summary(bday)$adjr2
```

```
## [1] 0.8938568 0.9502356 0.9616492 0.9690085 0.9711947 0.9721365 0.9727133
## [8] 0.9732158 0.9736021
```

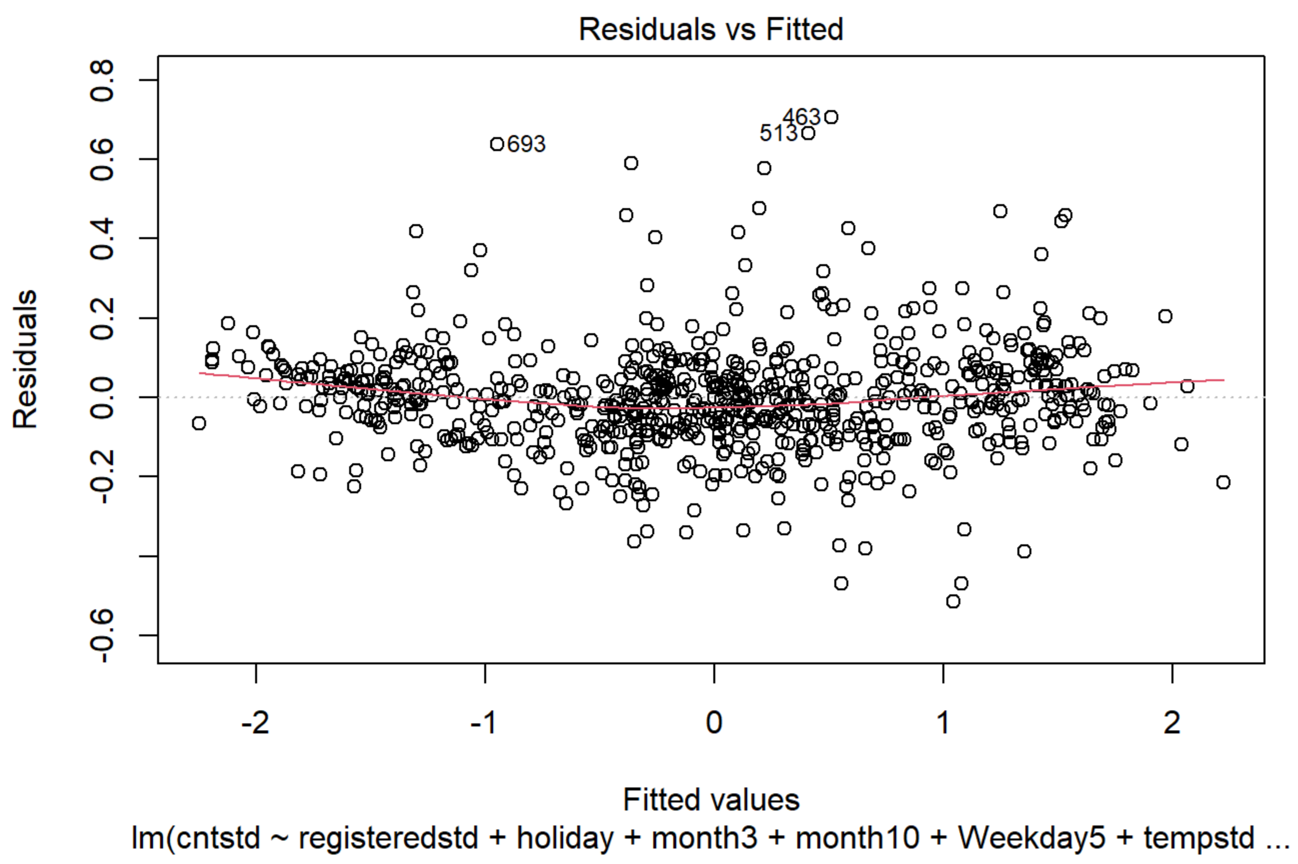
```
which.max(summary(bday)$adjr2)
```

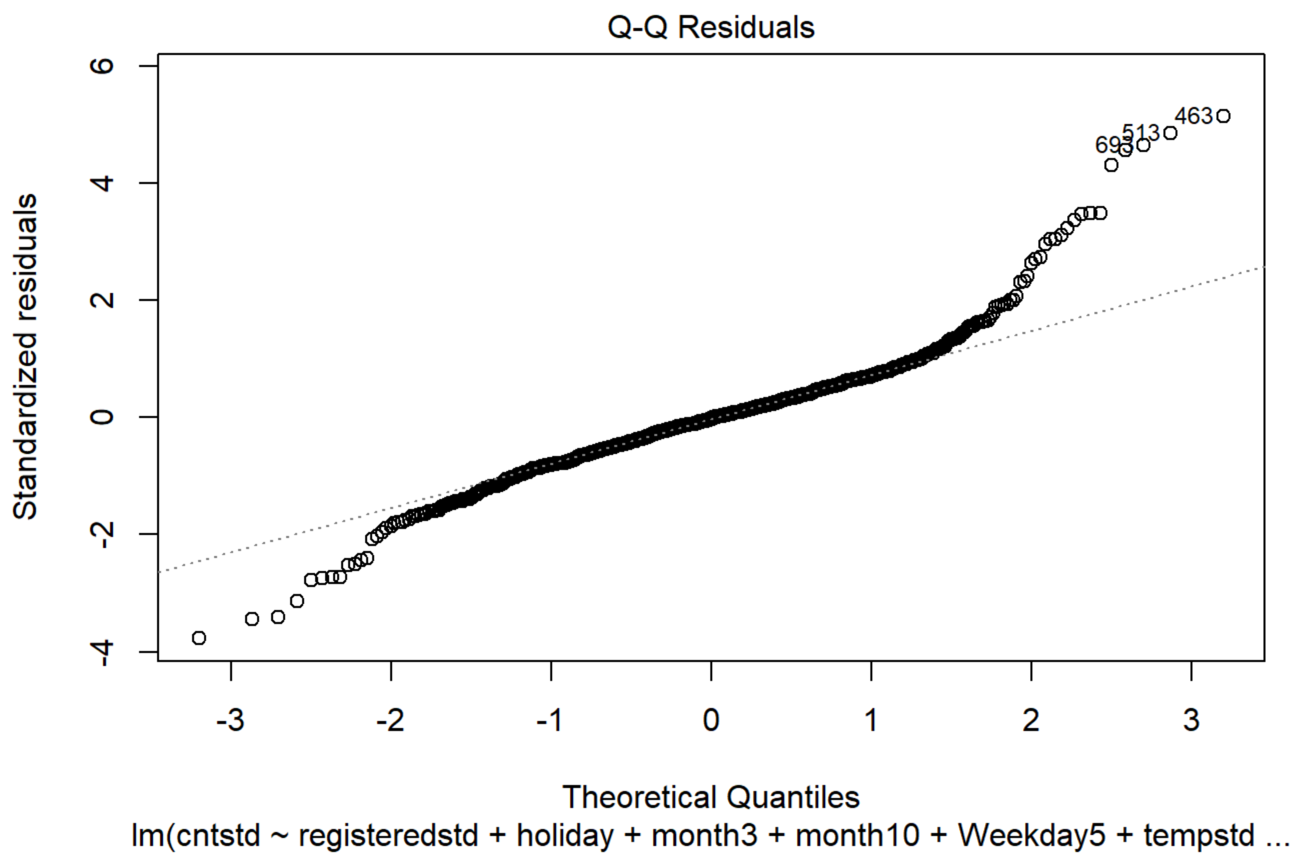
```
## [1] 9
```

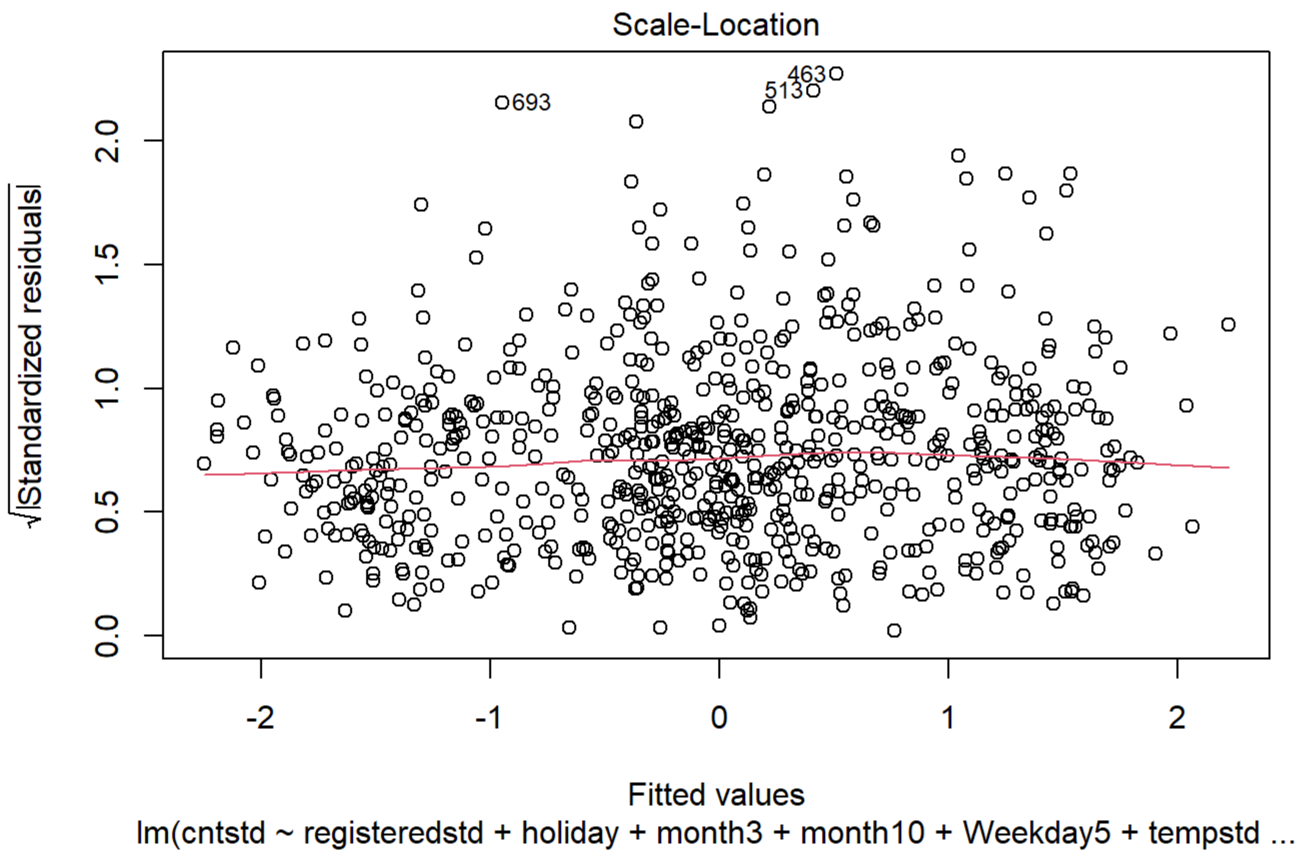
```
big_std_model = lm(cntstd~registeredstd+holiday+month3+month10+Weekday5+tempstd+humstd+workingda
y +Season2+(workingday * tempstd)+(registeredstd * workingday)+
                     (month3 * tempstd)+(holiday * registeredstd)+(holiday * tempstd)+(registere
dstd * Season3), data1)

summary(big_std_model)
```

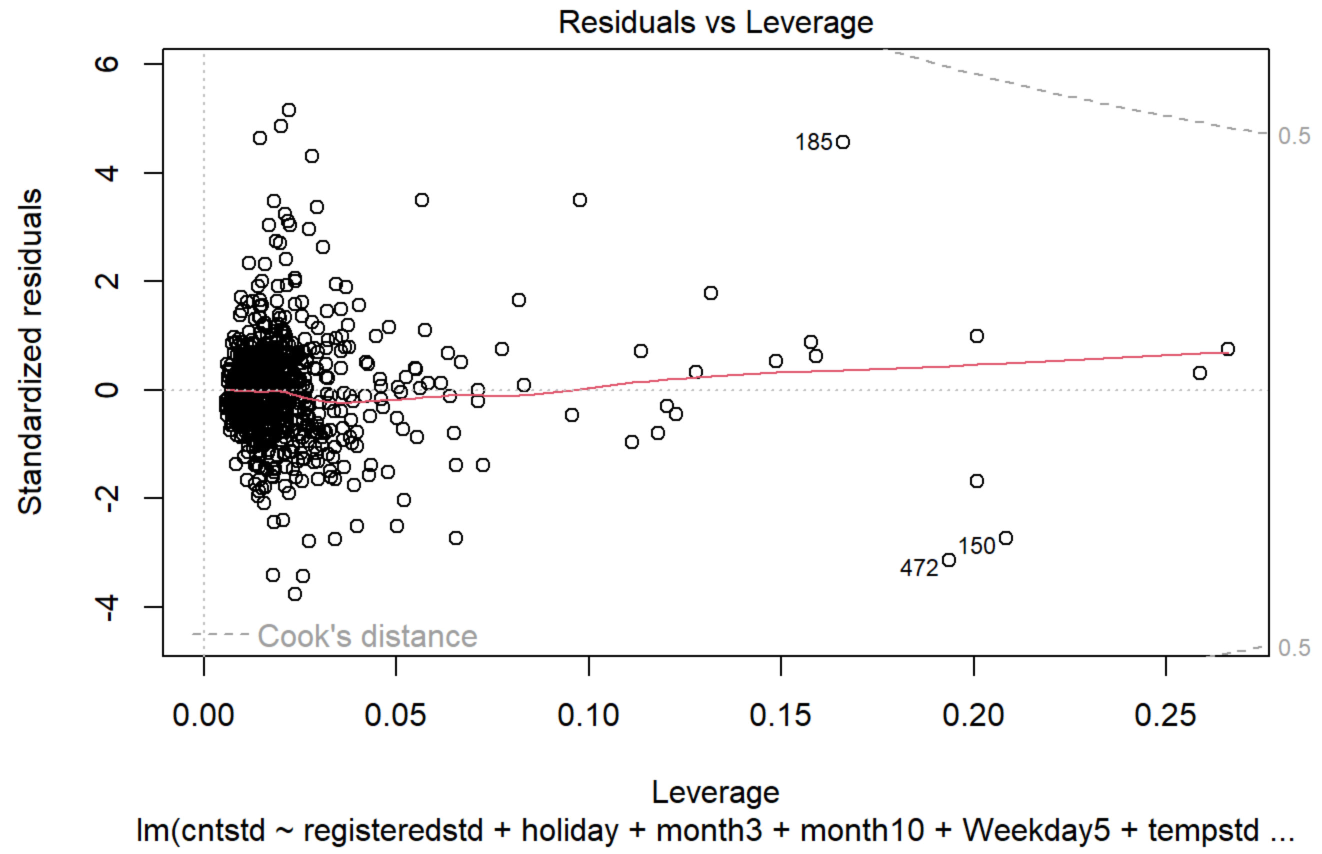
```
##
## Call:
## lm(formula = cntstd ~ registeredstd + holiday + month3 + month10 +
##     Weekday5 + tempstd + humstd + workingday + Season2 + (workingday *
##     tempstd) + (registeredstd * workingday) + (month3 * tempstd) +
##     (holiday * registeredstd) + (holiday * tempstd) + (registeredstd *
##     Season3), data = data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51560 -0.07435 -0.00300  0.06582  0.70515
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.358767   0.015403  23.292  < 2e-16 ***
## registeredstd             1.114425   0.014985  74.368  < 2e-16 ***
## holiday                  -0.261909   0.039513  -6.628 6.69e-11 ***
## month3                    0.155656   0.026345   5.908 5.35e-09 ***
## month10                   0.115184   0.021329   5.400 9.07e-08 ***
## Weekday5                  0.093957   0.015214   6.176 1.10e-09 ***
## tempstd                   0.149124   0.015039   9.916  < 2e-16 ***
## humstd                   -0.029513   0.005466  -5.399 9.12e-08 ***
## workingday               -0.603933   0.013272 -45.505  < 2e-16 ***
## Season2                   0.156803   0.016972   9.239  < 2e-16 ***
## Season3                   0.060629   0.024956   2.429   0.0154 *
## tempstd:workingday       -0.081297   0.014055  -5.784 1.09e-08 ***
## registeredstd:workingday -0.236702   0.016001 -14.793  < 2e-16 ***
## month3:tempstd            0.149379   0.033120   4.510 7.57e-06 ***
## registeredstd:holiday    -0.323088   0.043478  -7.431 3.09e-13 ***
## holiday:tempstd           0.193468   0.037358   5.179 2.91e-07 ***
## registeredstd:Season3     0.063085   0.014827   4.255 2.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1384 on 714 degrees of freedom
## Multiple R-squared:  0.9813, Adjusted R-squared:  0.9809
## F-statistic:  2339 on 16 and 714 DF,  p-value: < 2.2e-16
```

```
plot(big_std_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cntstd ~ registeredstd + holiday + month3 + month10 + Weekday5 + tempstd ...
693
463
513
0.8
0.6
0.4
0.2
0.0
-0.2
-0.6
-2
-1
0
1
2


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(cntstd ~ registeredstd + holiday + month3 + month10 + Weekday5 + tempstd ...
463
513
693
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

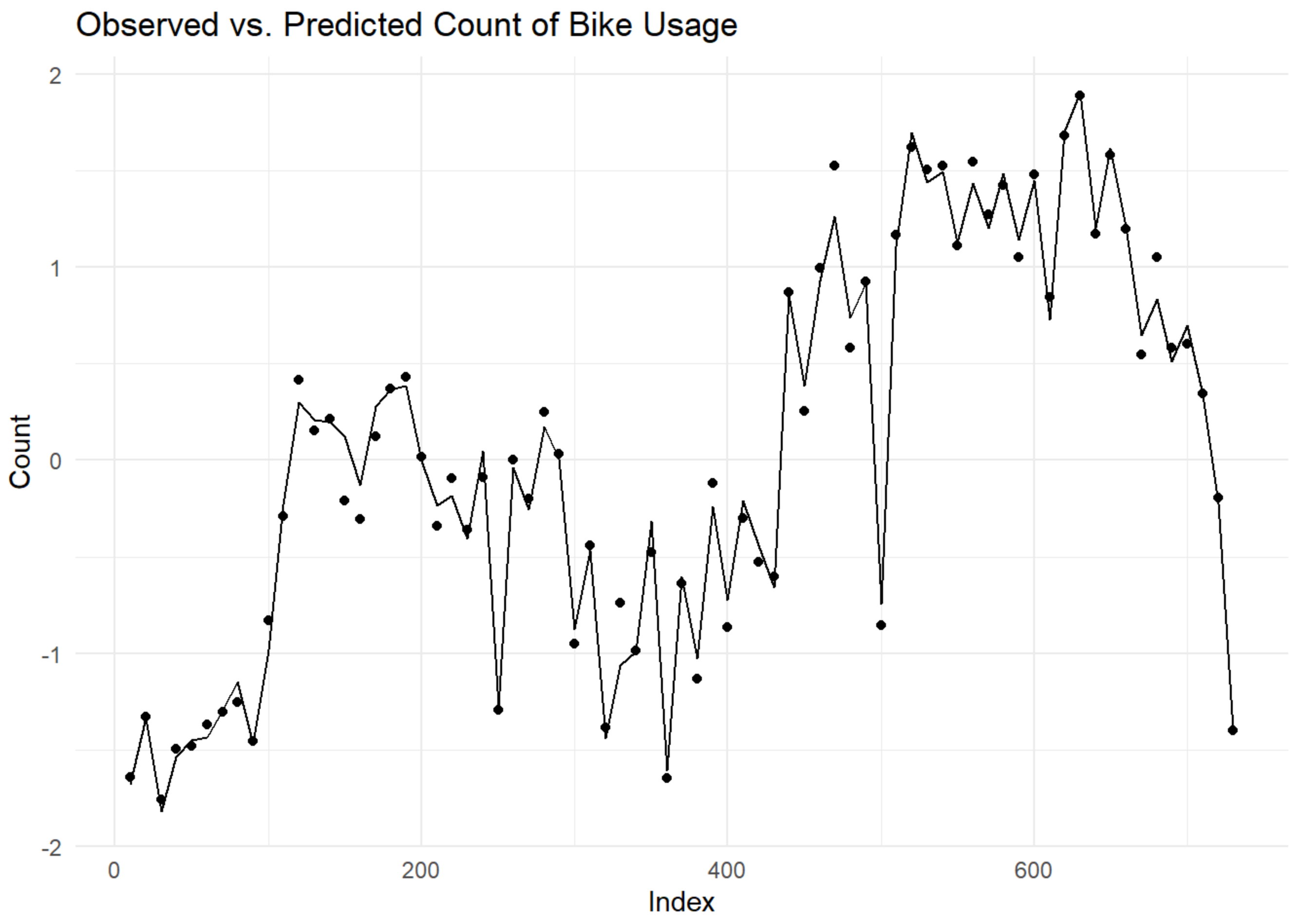

Scale-Location
463
513
693
2.0
1.5
1.0
0.5
0.0
√|Standardized residuals|
-2
-1
0
1
2
Fitted values
lm(cntstd ~ registeredstd + holiday + month3 + month10 + Weekday5 + tempstd ...


Residuals vs Leverage
185
150
472
6
4
2
0
-2
-4
Standardized residuals
0.5
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(cntstd ~ registeredstd + holiday + month3 + month10 + Weekday5 + tempstd ...

```
predicted_values <- predict(big_std_model, newdata = data1)
RMSE(predicted_values, data1$cntstd)
```

```
## [1] 0.1367364
```

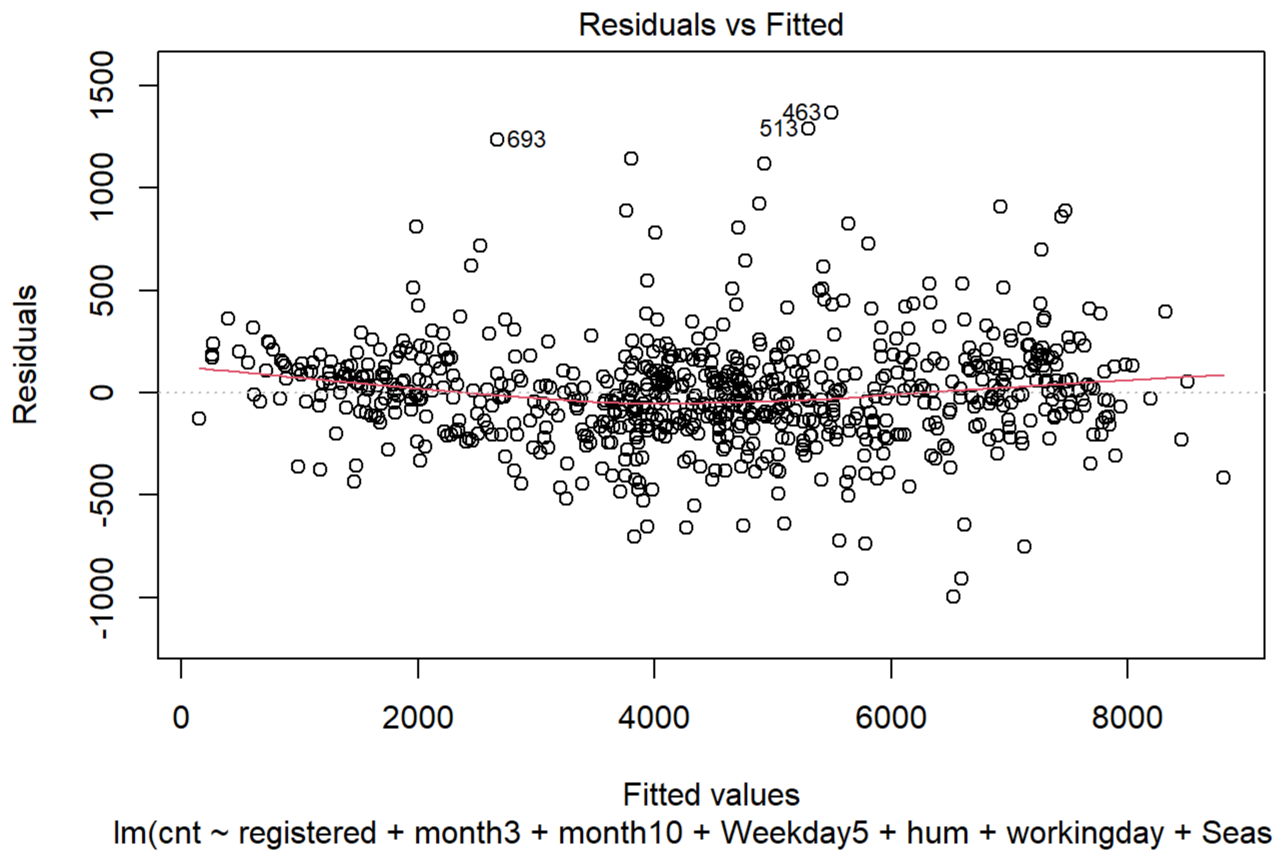
```
comparison_data <- data.frame(
  Instant = data1$instant,
  Observed = data1$cntstd,
  Predicted = predicted_values
)
filtered_data <- comparison_data %>% filter(row_number() %% 10 == 0)

ggplot(filtered_data, aes(x = Instant, y = Observed)) +
  geom_point() +
  geom_line(aes(x = Instant, y = Predicted)) +
  labs(x = "Index", y = "Count", title = "Observed vs. Predicted Count of Bike Usage") +
  theme_minimal()
```

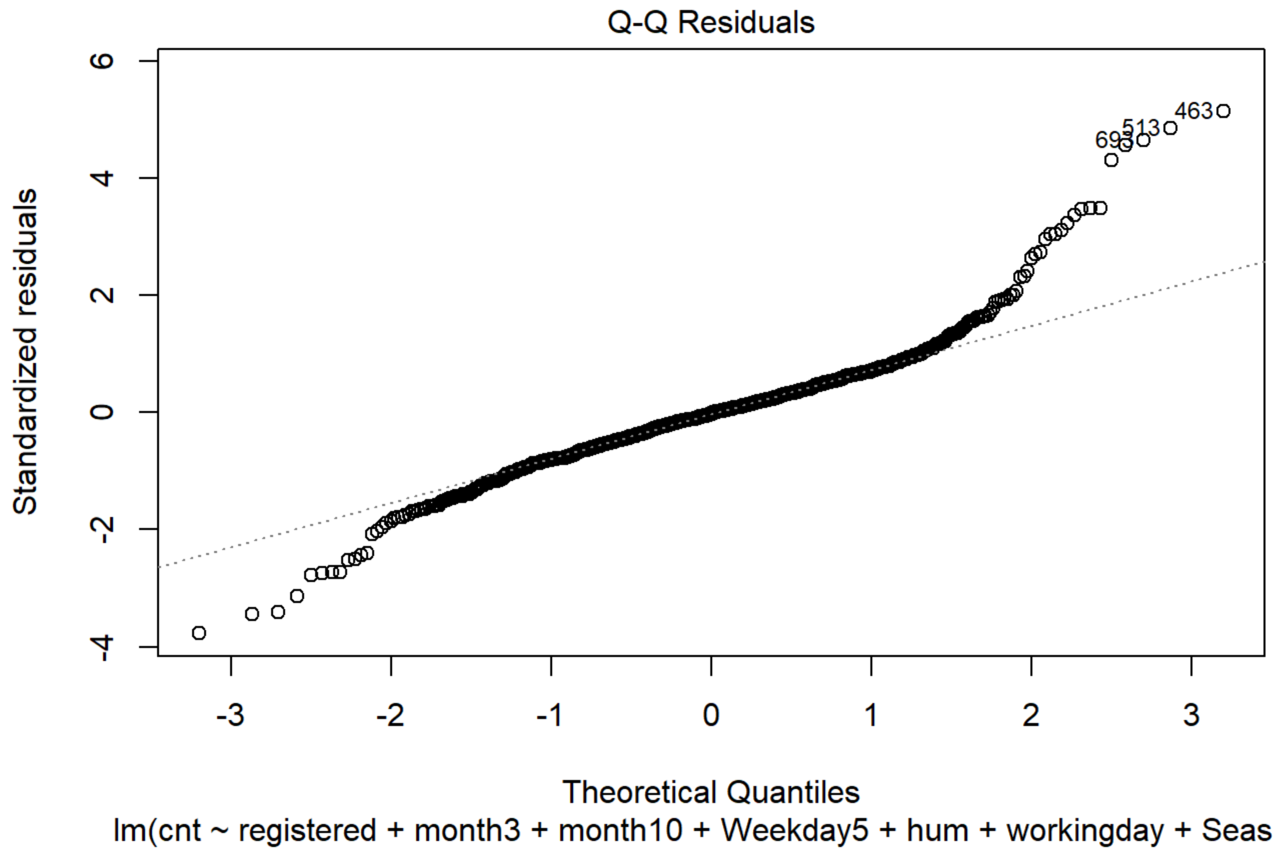


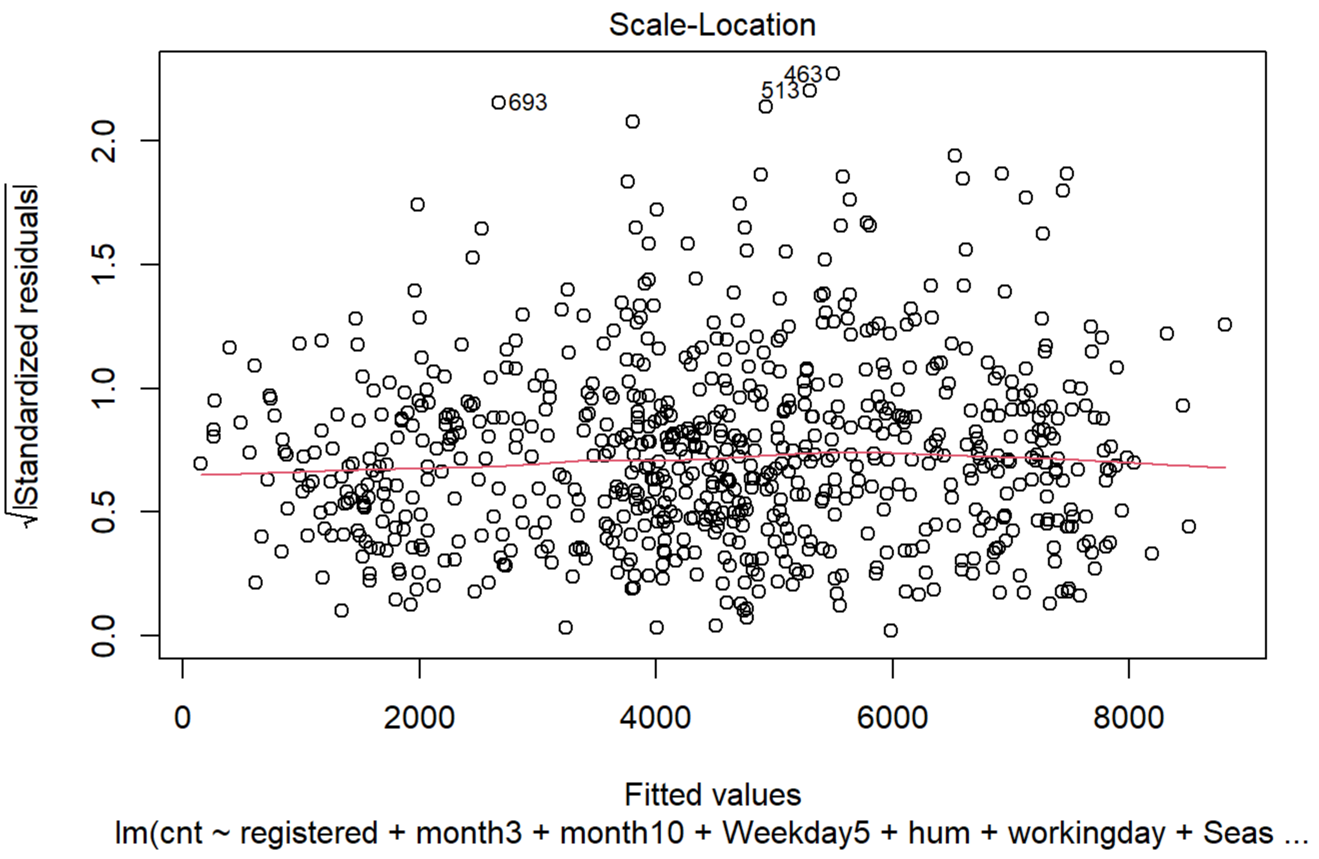
```
day_model_unstd =  lm(cnt~registered+month3+month10+Weekday5+hum+workingday +Season2+(workingday
* temp)+(registered * workingday)+
                     (month3 * temp)+(holiday * registered)+(holiday * temp)+(registered * Seaso
n3), data1)
summary(day_model_unstd)
```

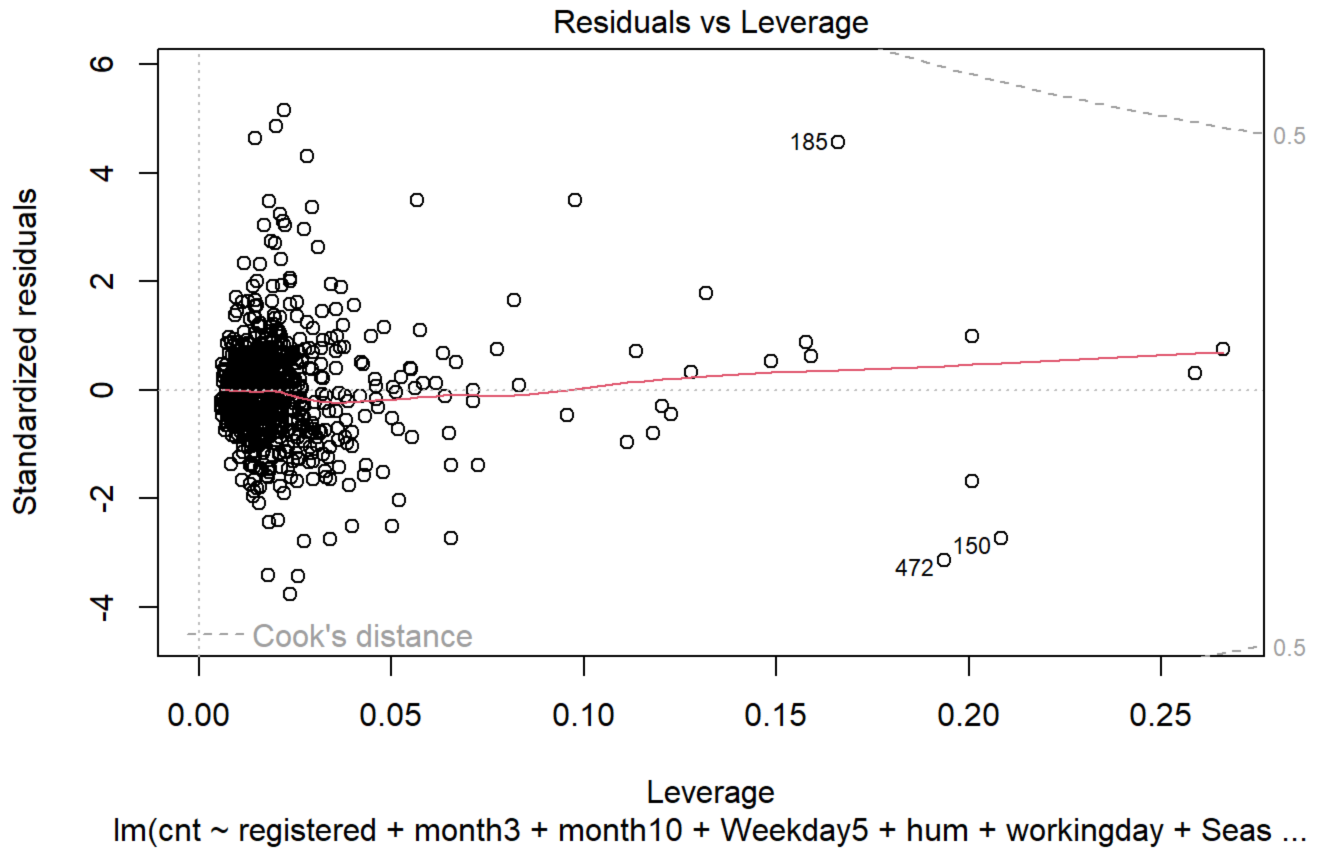
```
##
## Call:
## lm(formula = cnt ~ registered + month3 + month10 + Weekday5 +
##     hum + workingday + Season2 + (workingday * temp) + (registered *
##     workingday) + (month3 * temp) + (holiday * registered) +
##     (holiday * temp) + (registered * Season3), data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -998.83 -144.02   -5.82  127.52 1366.02
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -389.32798   73.49446  -5.297 1.57e-07 ***
## registered              1.38367    0.01861  74.368  < 2e-16 ***
## month3               -481.59763  143.10463  -3.365 0.000805 ***
## month10               223.13493   41.31972   5.400 9.07e-08 ***
## Weekday5              182.01523   29.47196   6.176 1.10e-09 ***
## hum                  -401.41153   74.34748  -5.399 9.12e-08 ***
## workingday            330.77292   69.36812   4.768 2.25e-06 ***
## Season2               303.75970   32.87878   9.239  < 2e-16 ***
## temp                 1578.16939  159.15753   9.916  < 2e-16 ***
## holiday               -54.99294  162.71864  -0.338 0.735490
## Season3              -168.91984   95.73855  -1.764 0.078094 .
## workingday:temp      -860.36298  148.73902  -5.784 1.09e-08 ***
## registered:workingday   -0.29389    0.01987 -14.793  < 2e-16 ***
## month3:temp          1580.86401  350.50587   4.510 7.57e-06 ***
## registered:holiday      -0.40115    0.05398  -7.431 3.09e-13 ***
## temp:holiday         2047.45442  395.35829   5.179 2.91e-07 ***
## registered:Season3      0.07833    0.01841   4.255 2.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 268 on 714 degrees of freedom
## Multiple R-squared:  0.9813, Adjusted R-squared:  0.9809
## F-statistic:  2339 on 16 and 714 DF,  p-value: < 2.2e-16
```

```
plot(day_model_unstd)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cnt ~ registered + month3 + month10 + Weekday5 + hum + workingday + Seas ...
693
463
513
1500
1000
500
0
-500
-1000
0
2000
4000
6000
8000


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(cnt ~ registered + month3 + month10 + Weekday5 + hum + workingday + Seas ...
463
513
693
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

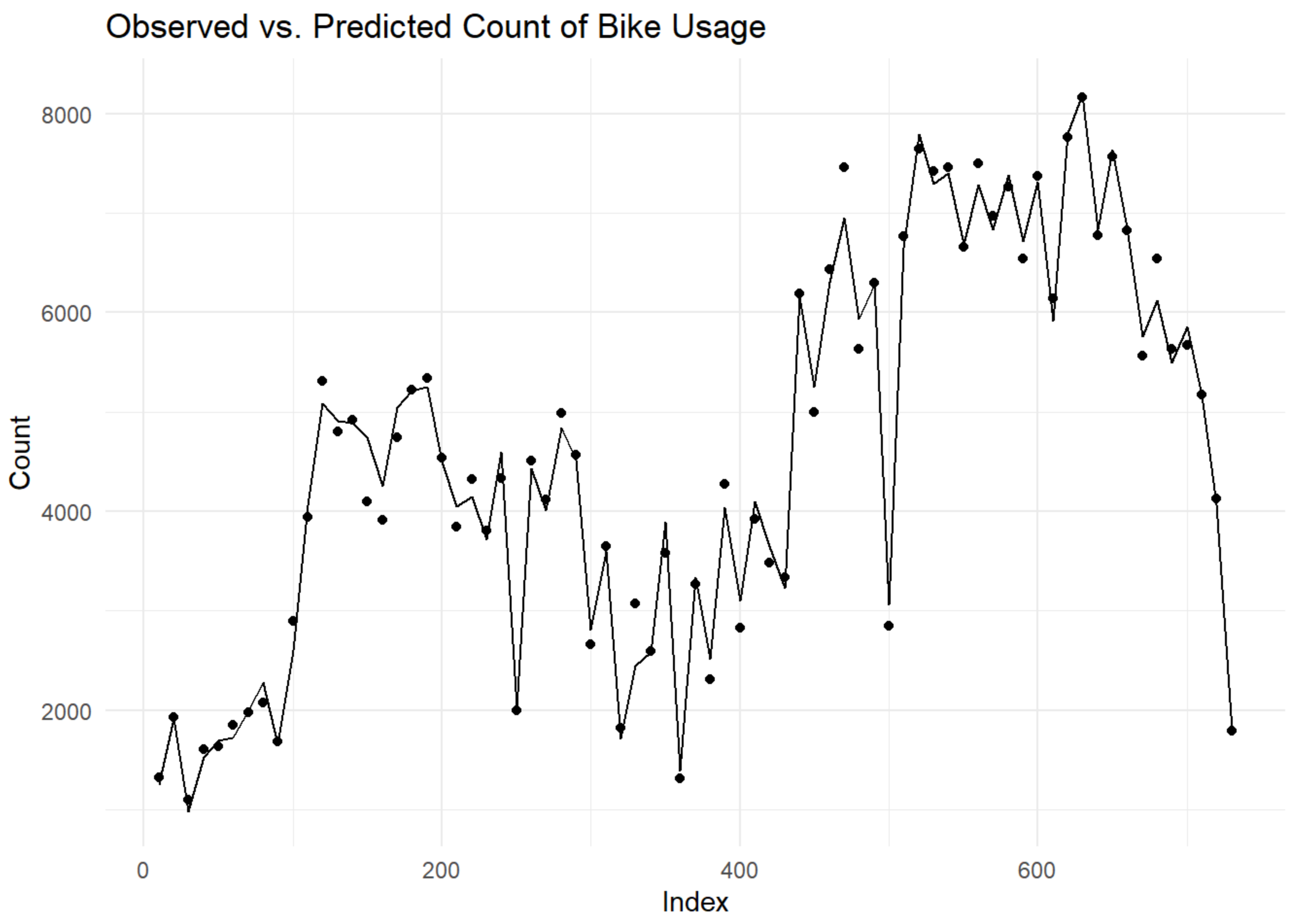

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
693
513
463
0
2000
4000
6000
8000
Fitted values
lm(cnt ~ registered + month3 + month10 + Weekday5 + hum + workingday + Seas ...


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
6
185
150
472
0.5
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(cnt ~ registered + month3 + month10 + Weekday5 + hum + workingday + Seas ...

```
predicted_values <- predict(day_model_unstd, newdata = data1)
RMSE(predicted_values, data1$cnt)
```

```
## [1] 264.8874
```

```
comparison_data <- data.frame(
  Instant = data1$instant,
  Observed = data1$cnt,
  Predicted = predicted_values
)
filtered_data <- comparison_data %>% filter(row_number() %% 10 == 0)

ggplot(filtered_data, aes(x = Instant, y = Observed)) +
  geom_point() +
  geom_line(aes(x = Instant, y = Predicted)) +
  labs(x = "Index", y = "Count", title = "Observed vs. Predicted Count of Bike Usage") +
  theme_minimal()
```


Observed vs. Predicted Count of Bike Usage
Count
Index