ADS-506 Final Project Initial EDA

Anna

2024-11-01

```r
# Importing the data
bike_data <- read.csv("day.csv")
```

```r
# Getting summary statistics
summary(bike_data)
```

```
##     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  
```

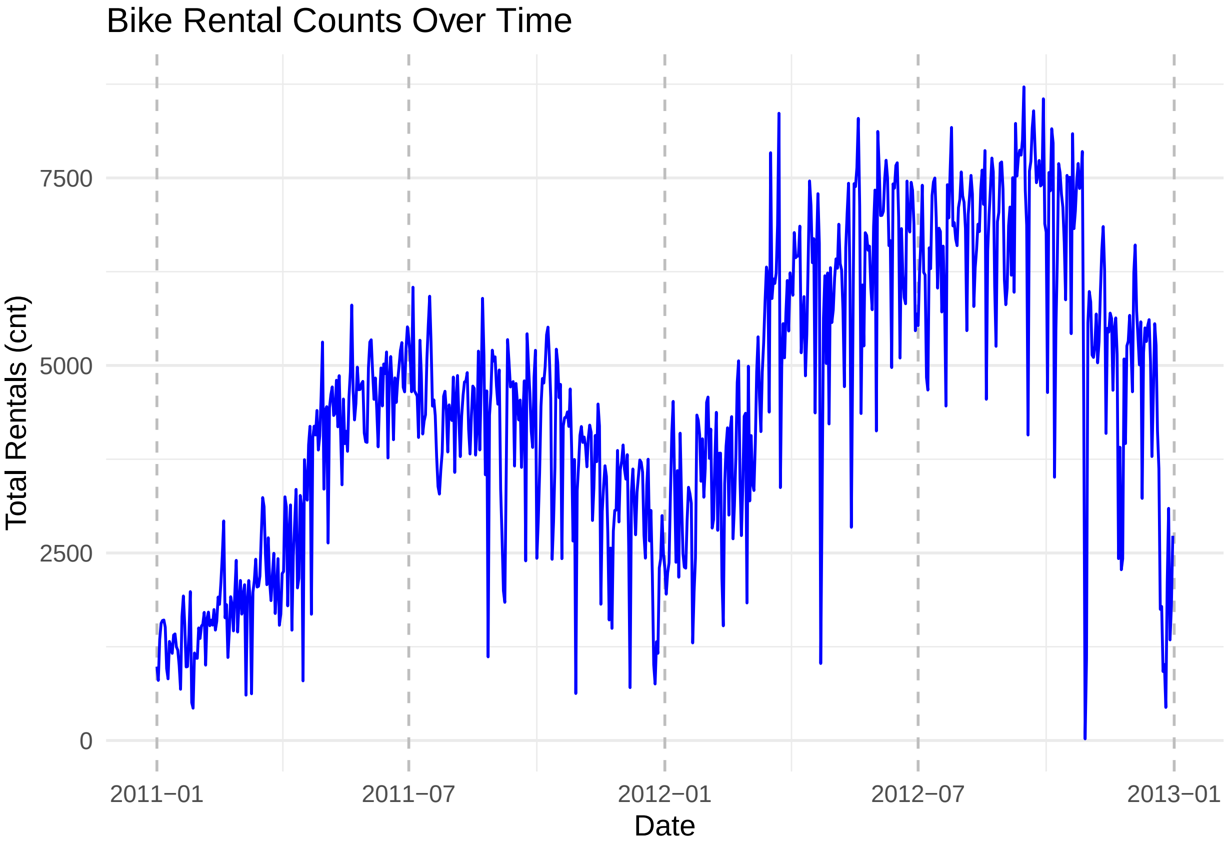
```r
# Exploring unique values in categorical columns
lapply(bike_data[c("season", "mnth", "weekday")], unique)
```

```
## $season
## [1] 1 2 3 4
## 
## $mnth
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
## 
```

```
## $weekday
## [1] 6 0 1 2 3 4 5
```

```
# Converting dteday to Date type and extract year, month, or day
bike_data$dteday <- as.Date(bike_data$dteday)
```

```
# Lets plot a trends over time
library(ggplot2)
ggplot(bike_data, aes(x = dteday, y = cnt)) +
    geom_line(color = "blue") +
    labs(title = "Bike Rental Counts Over Time",
         x = "Date",
         y = "Total Rentals (cnt)") +
    theme_minimal() +  # Use a cleaner theme with grid lines
    theme(panel.grid.major.x = element_line(color = "grey", linetype = "dashed"))
```

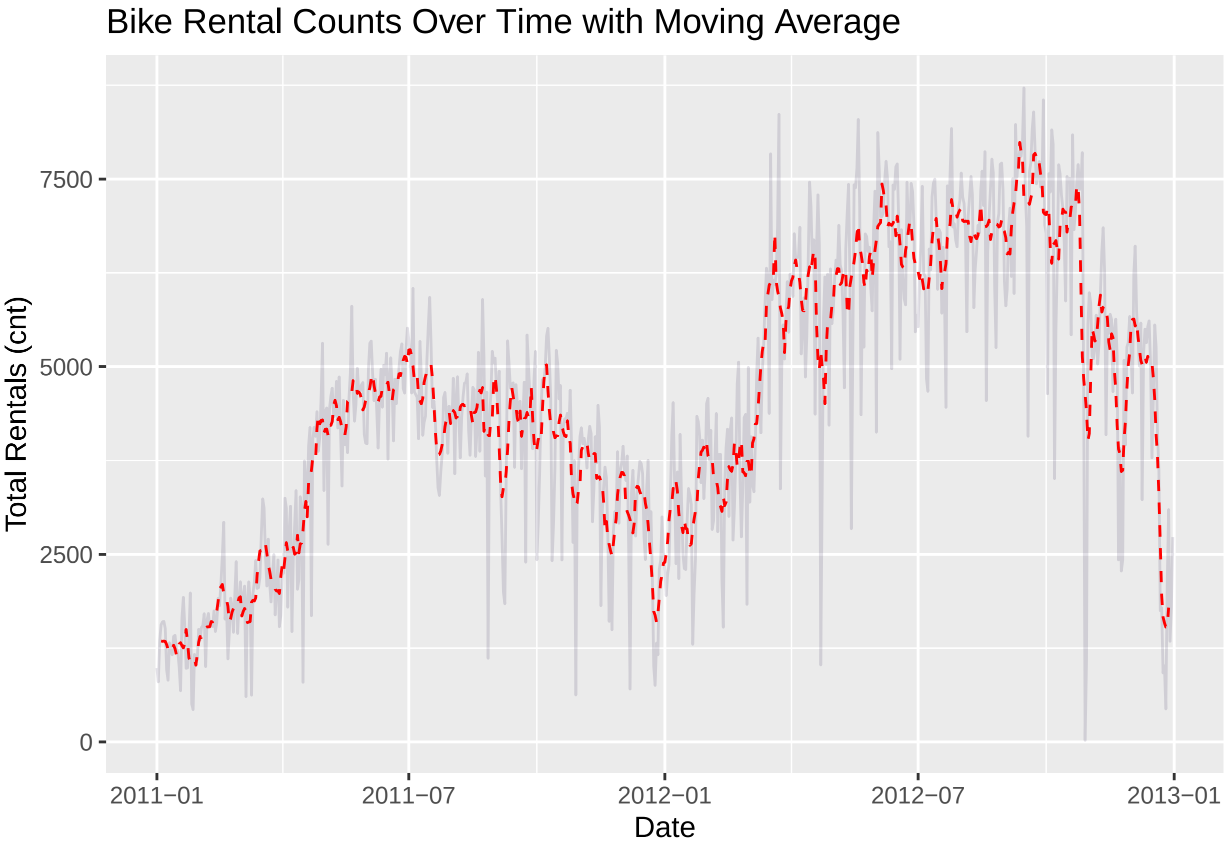


```
library(zoo)
```

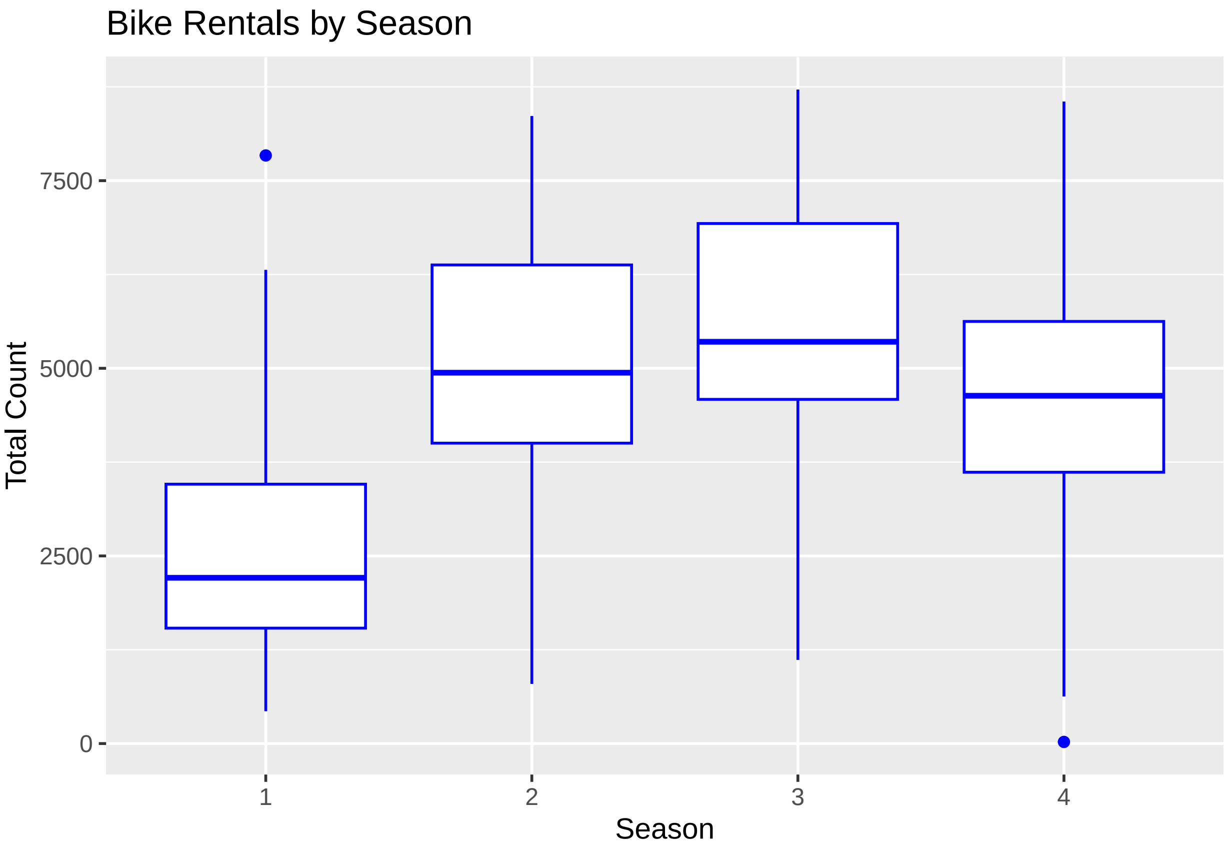
```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bike_data$moving_avg <- rollmean(bike_data$cnt, k = 7, fill = NA)  # 7-day moving average
ggplot(bike_data, aes(x = dteday)) +
    geom_line(aes(y = cnt), color = "#12023a20") +
    geom_line(aes(y = moving_avg), color = "red", linetype = "dashed") +
    labs(title = "Bike Rental Counts Over Time with Moving Average",
         x = "Date",
         y = "Total Rentals (cnt)")
```

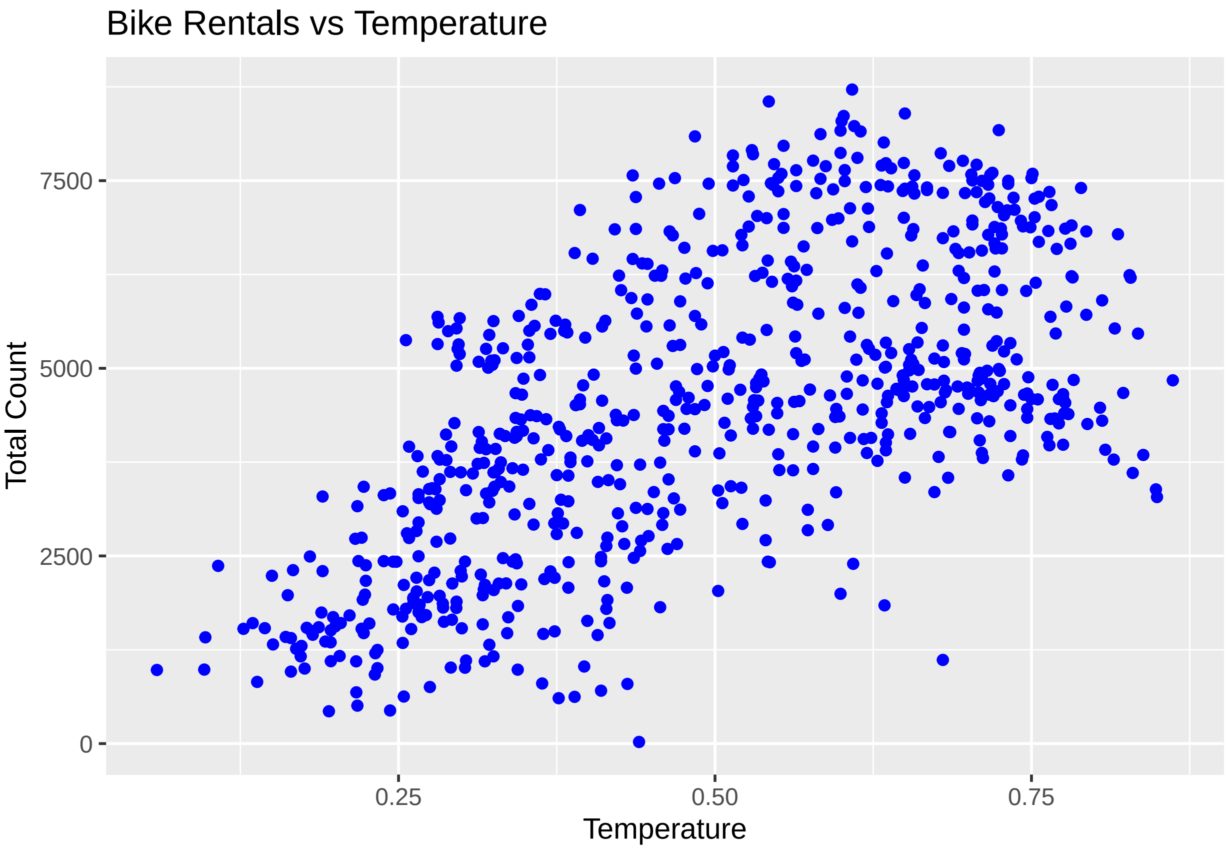
```
## Warning: Removed 6 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



```
# Visualizing distribution of rentals by season and month
ggplot(bike_data, aes(x = as.factor(season), y = cnt)) +
  geom_boxplot(color='blue') +
  labs(title = "Bike Rentals by Season", x = "Season", y = "Total Count")
```



```r
# Visualizing rental distribution by weather or temperature
ggplot(bike_data, aes(x = temp, y = cnt)) +
  geom_point(color='blue') +
  labs(title = "Bike Rentals vs Temperature", x = "Temperature", y = "Total Count")
```



```r
# showing how numeric variables relate to rentals
cor(bike_data[, sapply(bike_data, is.numeric)])
```

```
##                  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
## moving_avg            NA           NA           NA           NA           NA
##                  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
## moving_avg            NA           NA          NA            NA           NA
##                    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
## moving_avg          NA           NA          NA          NA          NA
##            moving_avg
## instant            NA
## season             NA
## yr                 NA
## mnth               NA
## holiday            NA
## weekday            NA
## workingday         NA
## weathersit         NA
## temp               NA
## atemp              NA
## hum                NA
## windspeed          NA
## casual             NA
## registered         NA
## cnt                NA
## moving_avg          1
```

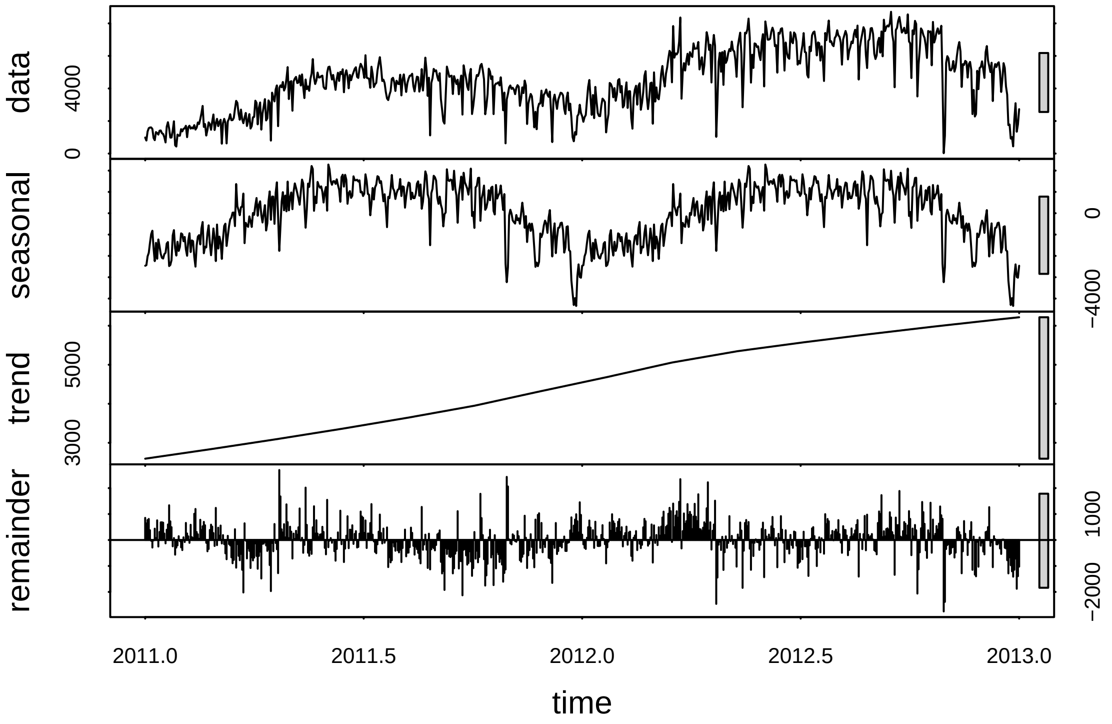
```r
library(forecast)
```

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

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Convert to time series object
bike_ts <- ts(bike_data$cnt, frequency = 365, start = c(2011, 1))
# Decompose the series
```

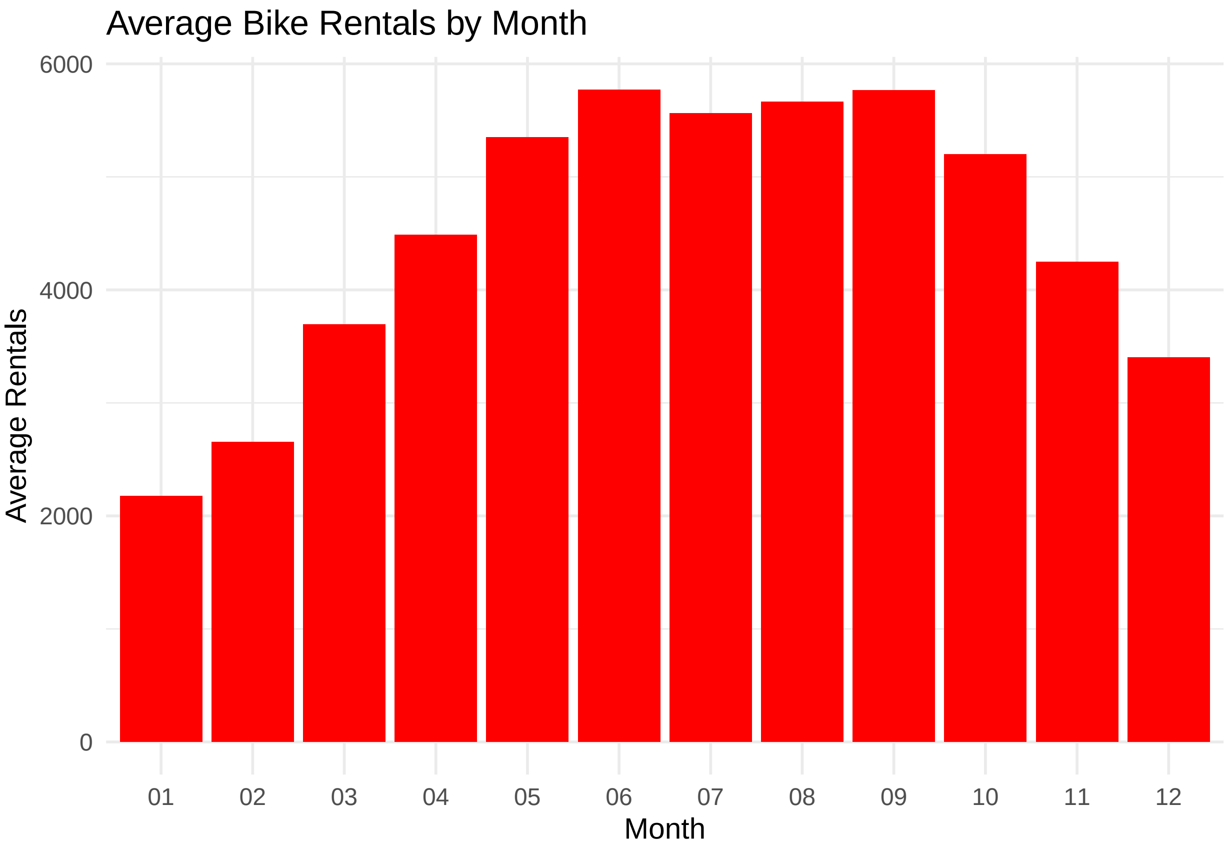
```
decomposition <- stl(bike_ts, s.window = "periodic")
plot(decomposition)
```



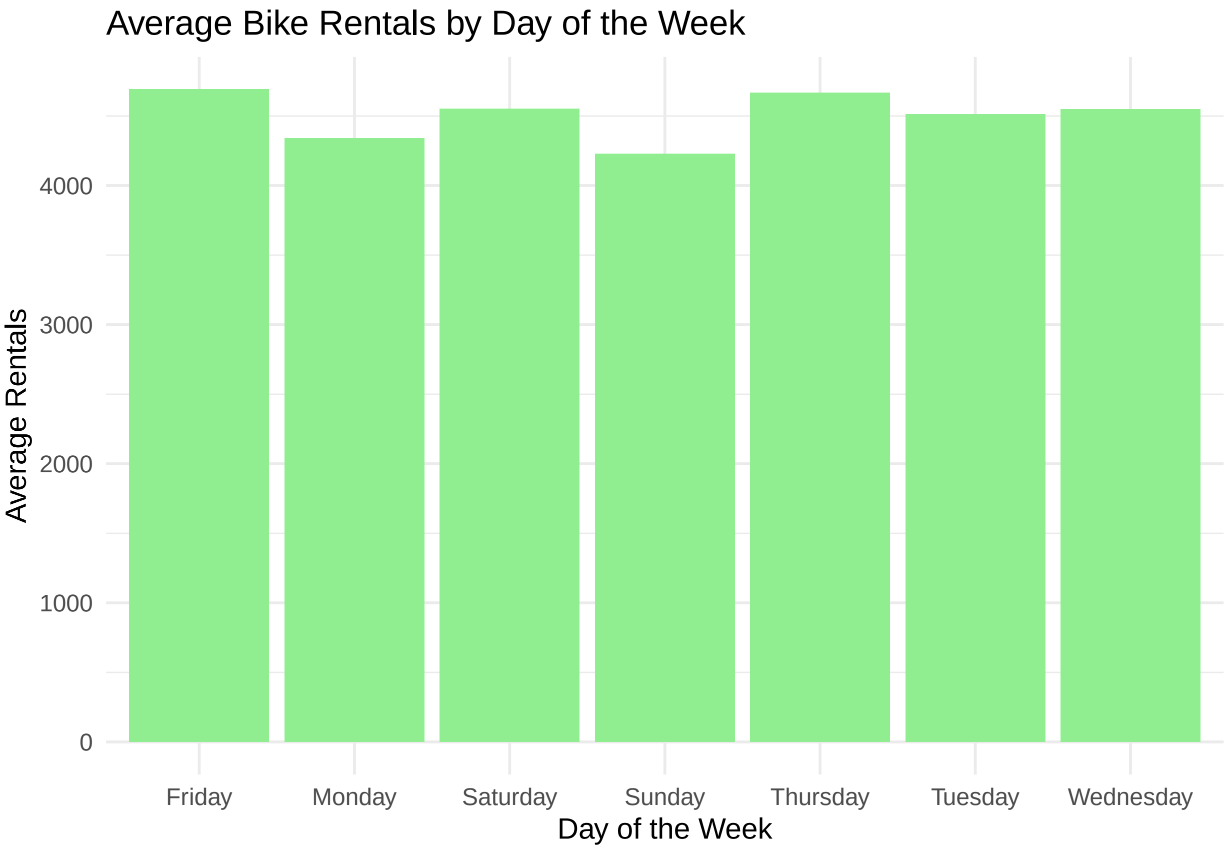
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

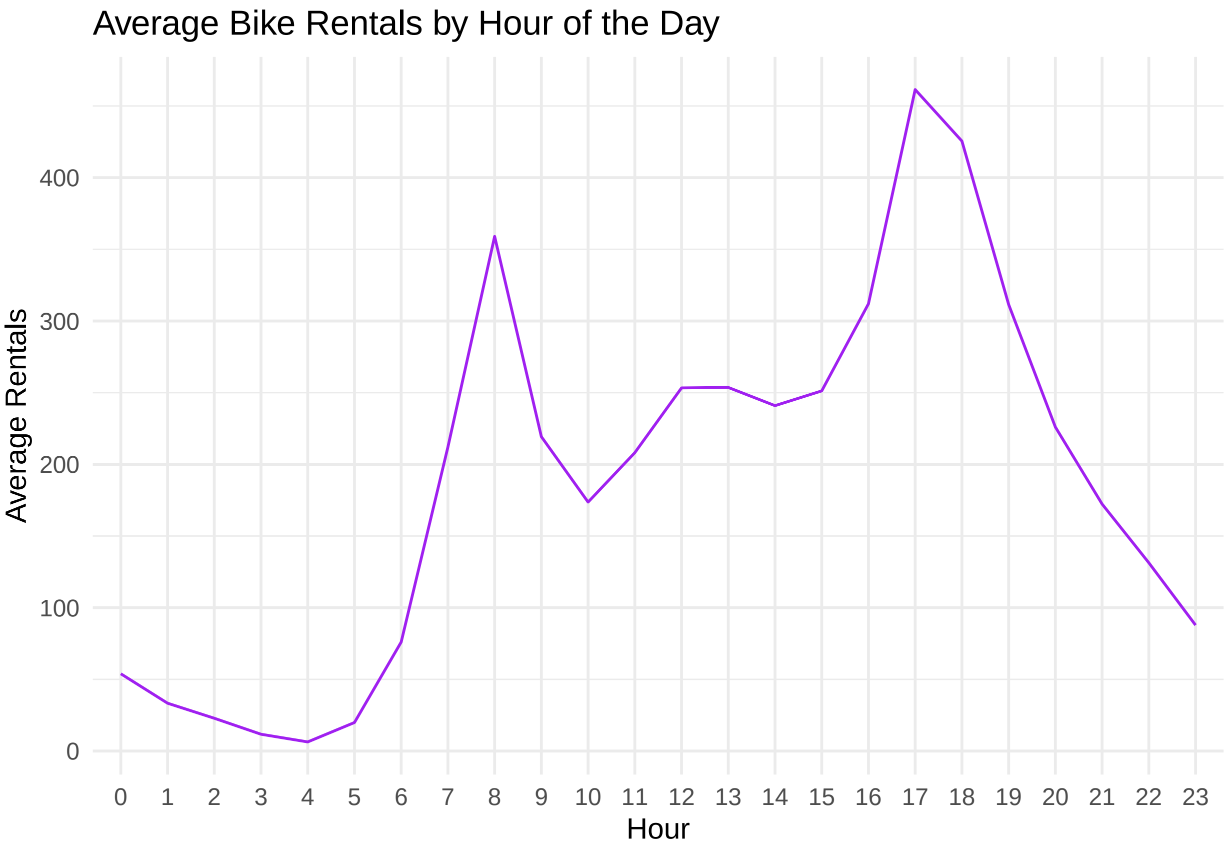
```
library(ggplot2)

bike_data$month <- format(as.Date(bike_data$dteday), "%m")
monthly_avg <- bike_data %>%
    group_by(month) %>%
    summarise(avg_cnt = mean(cnt))

ggplot(monthly_avg, aes(x = month, y = avg_cnt)) +
    geom_bar(stat = "identity", fill = "#ff0000") +
    labs(title = "Average Bike Rentals by Month",
         x = "Month", y = "Average Rentals") +
    theme_minimal()
```



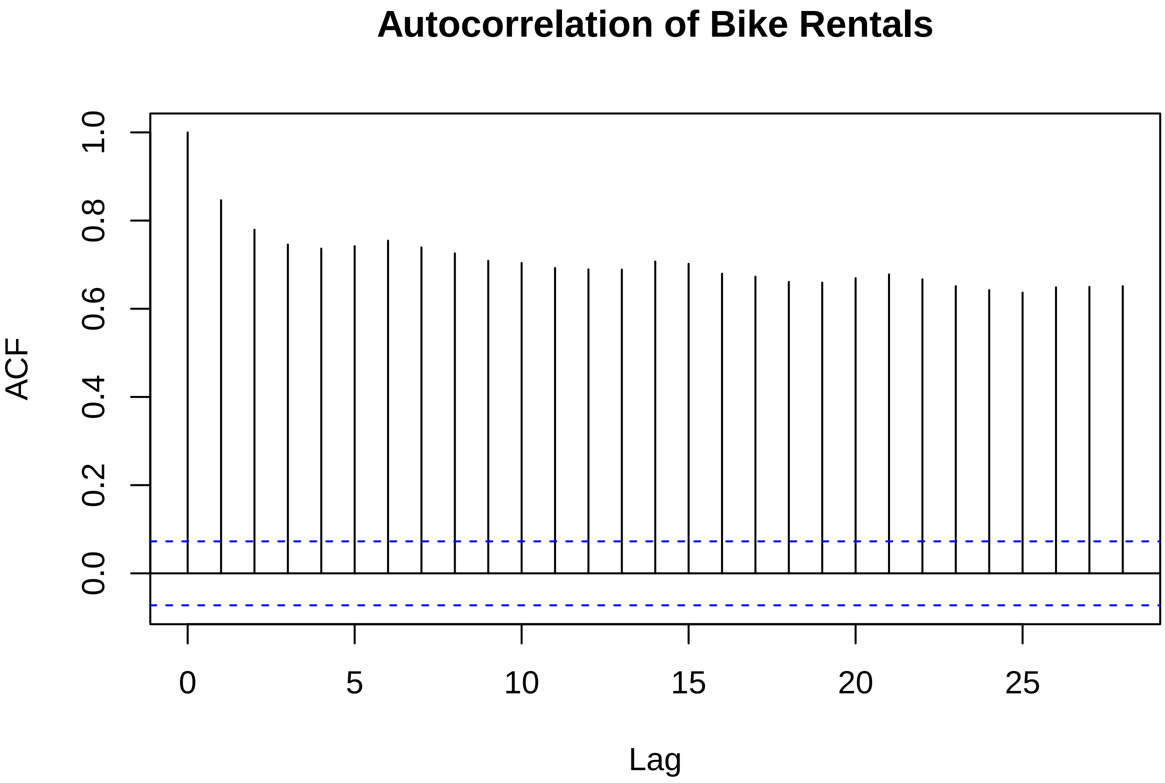
```
bike_data$weekday <- format(as.Date(bike_data$dteday), "%A")
weekday_avg <- bike_data %>%
    group_by(weekday) %>%
    summarise(avg_cnt = mean(cnt)) %>%
    arrange(factor(weekday, levels = c("Monday", "Tuesday", "Wednesday",
                                       "Thursday", "Friday", "Saturday", "Sunday")))

ggplot(weekday_avg, aes(x = weekday, y = avg_cnt)) +
    geom_bar(stat = "identity", fill = "lightgreen") +
    labs(title = "Average Bike Rentals by Day of the Week",
         x = "Day of the Week", y = "Average Rentals") +
    theme_minimal()
```



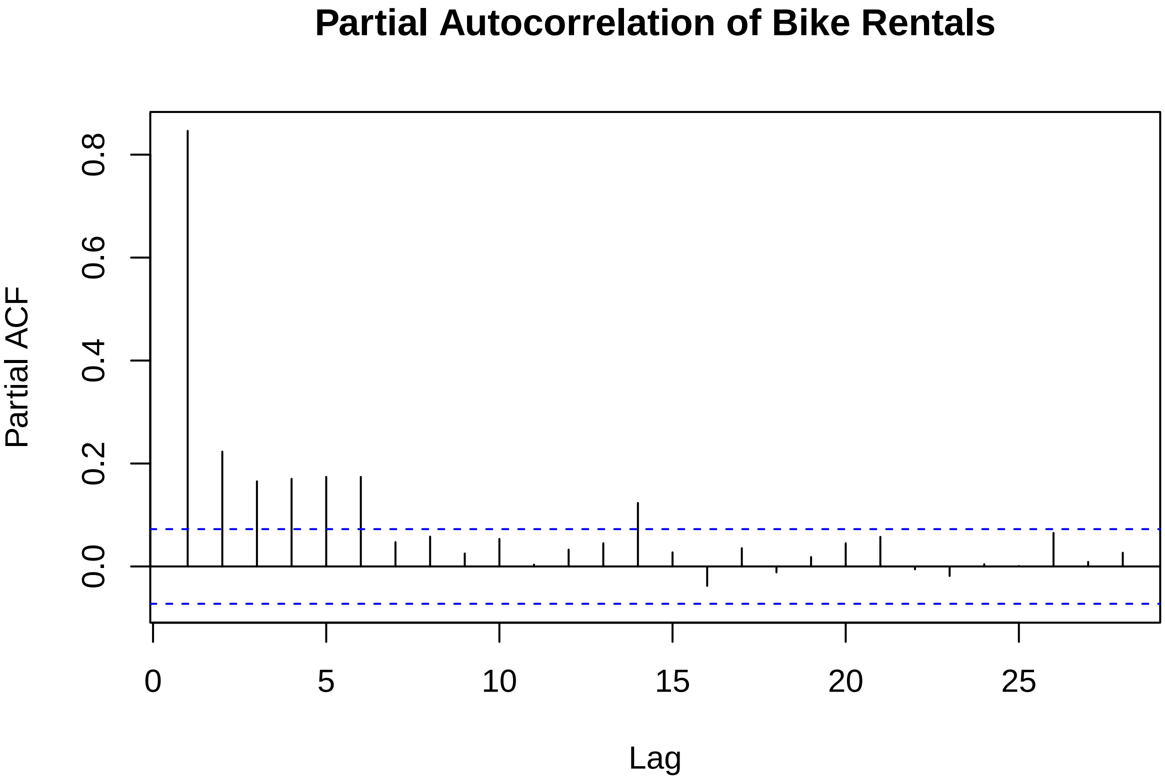
```
bike_data_hourly <- read.csv("/Users/gabrielmancillas/Documents/GitHub/ADS-506-Final-Team-Project/hour.c
bike_data_hourly$hour <- as.factor(bike_data_hourly$hr)

hourly_avg <- bike_data_hourly %>%
    group_by(hour) %>%
    summarise(avg_cnt = mean(cnt))

ggplot(hourly_avg, aes(x = hour, y = avg_cnt)) +
    geom_line(group = 1, color = "purple") +
    labs(title = "Average Bike Rentals by Hour of the Day",
         x = "Hour", y = "Average Rentals") +
    theme_minimal()
```



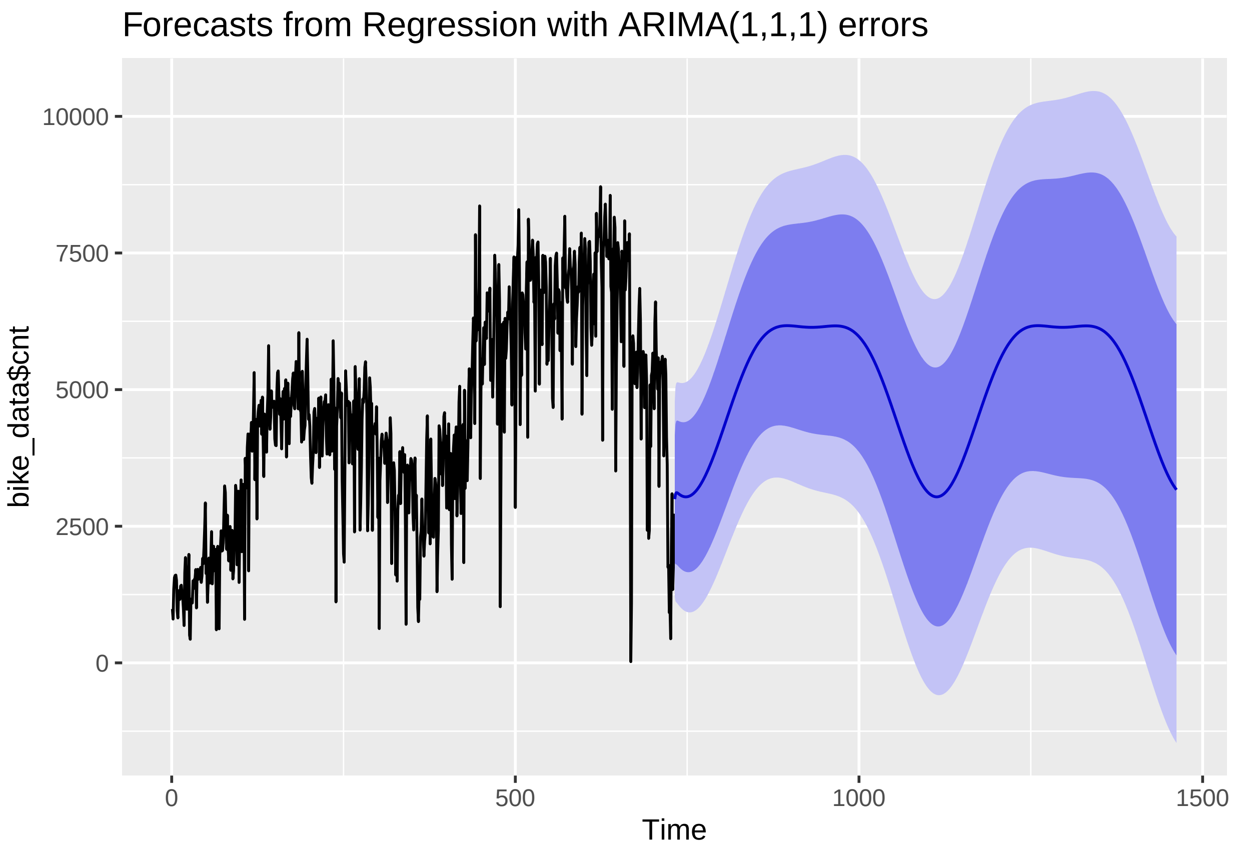
```
acf(bike_data$cnt, main = "Autocorrelation of Bike Rentals")
```



```
pacf(bike_data$cnt, main = "Partial Autocorrelation of Bike Rentals")
```



```
library(forecast)
fourier_terms <- fourier(ts(bike_data$cnt, frequency = 365), K = 2)
fit <- auto.arima(bike_data$cnt, xreg = fourier_terms)
forecast_fit <- forecast(fit, xreg = fourier_terms, h = 365)
autoplot(forecast_fit)
```


Forecasts from Regression with ARIMA(1,1,1) errors
bike_data$cnt
10000
7500
5000
2500
0
Time
0
500
1000
1500