Assignment 2 - Solution

Sharath Kumar Karnati | 011852253

2024-09-11

Question 1:

1 (50 points). This exercise relates to the Red Wine Quality dataset (winequality-red.csv), which can be found under the Datasets modules in Canvas. The dataset contains a number of physicochemical test variables for 1599 different red wine variants of the Portuguese "Vinho Verde" wine. The variables are • fixed_acidity • volatile_acidity • citric_acid • residual_sugar • chlorides • free_sulfur_dioxide • total_sulfur_dioxide • density • pH • sulphates • alcohol (output variable based on sensory data) • quality (score between 0 and 10) Before reading the data into R or Python, you can view it in Excel or a text editor. For each of the following questions, include the code you used to complete the task as your response, along with any plots or numeric outputs produced. You may omit outputs that are not relevant (such as dataframe contents), but still include all of your code.

Question 1a

Use the read.csv() function to read the data into R, or the csv library to read in the data with python. In R you will load the data into a dataframe. In python you may store it as a list of lists or use the pandas dataframe to store your data. Call the loaded data redwine. Ensure that your column headers are not treated as a row of data.

```
redwine <- read.csv(file = "/Users/sharathkarnati/Desktop/DS/winequality-red.csv", sep = ",", header = 



str(redwine)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed_acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile_acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric_acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual_sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free_sulfur_dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total_sulfur_dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

```
summary(redwine)
```

```
##  fixed_acidity   volatile_acidity  citric_acid    residual_sugar  
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900  
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900  
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200  
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free_sulfur_dioxide total_sulfur_dioxide    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

Question 1b

Find the mean quality of all the wine samples. Then find the median alcohol level for all the wine samples

```r
mean_quality <- mean(redwine[["quality"]])


cat("Mean quality of the wine samples is:", mean_quality)
```

```
## Mean quality of the wine samples is: 5.636023
```

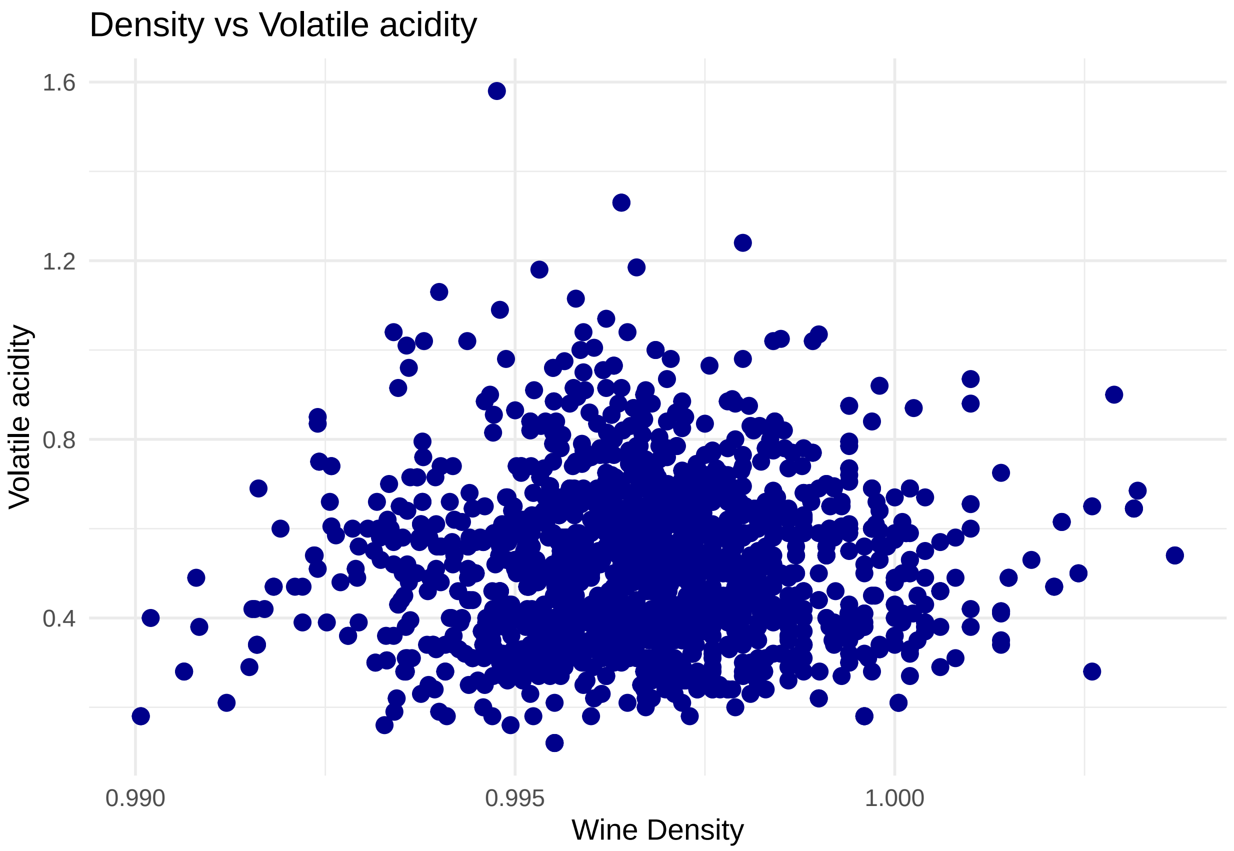
```r
alcohol_median <- median(redwine[["alcohol"]])

cat("Median alcohol level is:", alcohol_median)
```

```
## Median alcohol level is: 10.2
```

Question 1c.

Produce a scatterplot that shows the relationship between wine density and volatile_acidity. Ensure it has appropriate axis labels and a title. Briefly state if you see any effect of volatile_acidity on density.

```r
library(ggplot2)


ggplot(redwine, aes(x = density, y = volatile_acidity)) +
  geom_point(color = "darkblue", size = 2.5) +
  labs(
    title = "Density vs Volatile acidity",
    x = "Wine Density",
    y = "Volatile acidity"
  ) +
  theme_minimal()
```



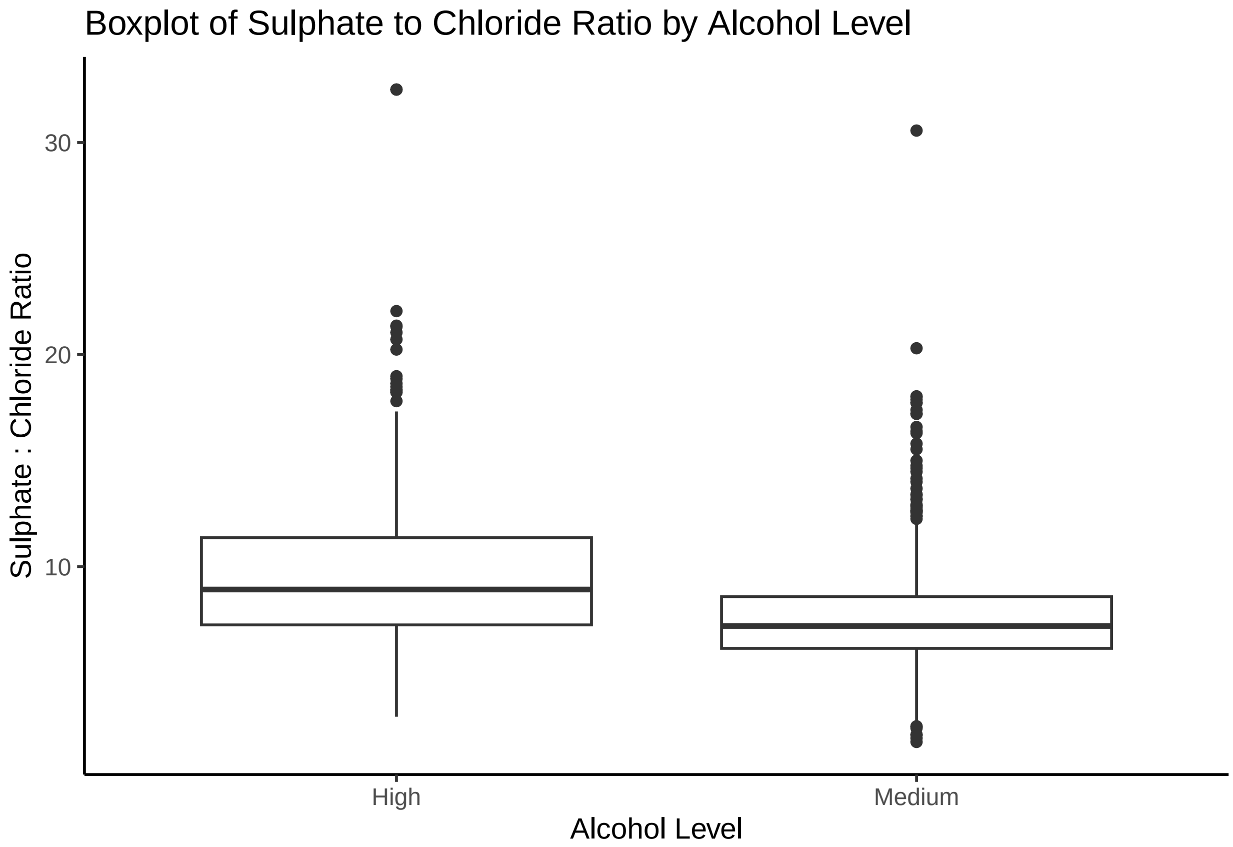
By observing the scattered graph above , we can say that volatile acidity does not have a significant effect on wine density and their relationship seems weak.

Question 1d

Create a new qualitative variable, called ALevel, by binning the alcohol variable into two categories (High and Medium). Specifically, divide the data into two groups based on whether the alcohol level exceeds 10.5 or not (alcohol greater than 10.5 is considered High otherwise it is considered Medium).

Now produce side-by-side boxplots of the ratio of sulphates to chlorides (hint: create a new variable that calculates sulphates / chlorides) for each of the two ALevel categories. There should be two boxes on your figure, one for High and one for Medium. How many samples are in the High category?

```
redwine$ALevel <- factor(ifelse(redwine$alcohol > 10.5, "High", "Medium"))


redwine$ratio_sul_chlro <- with(redwine, sulphates / chlorides)

ggplot(redwine, aes(x = ALevel, y = ratio_sul_chlro)) +
  geom_boxplot() +
  labs(
    x = "Alcohol Level",
    y = "Sulphate : Chloride Ratio",
    title = "Boxplot of Sulphate to Chloride Ratio by Alcohol Level"
  ) +
  theme_classic()
```



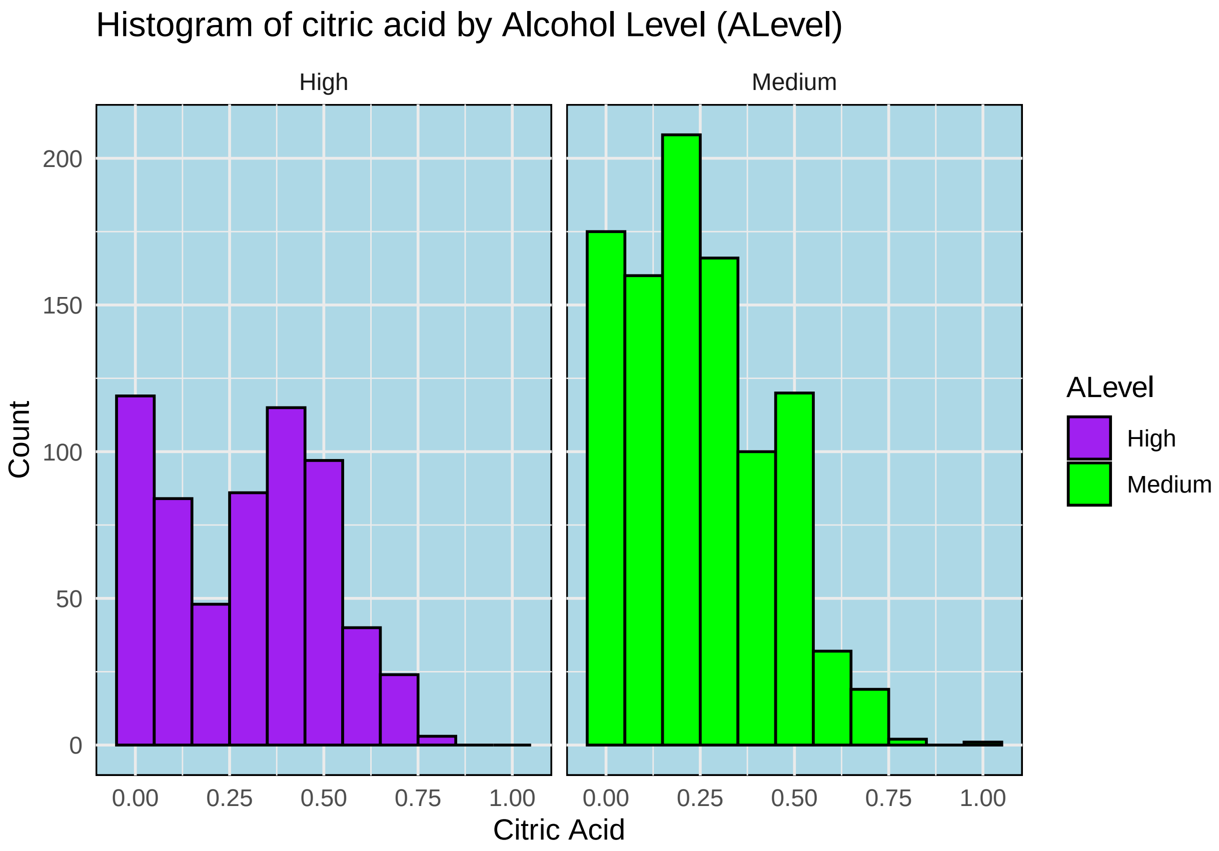
```
high_category_samples <- sum(redwine$ALevel == "High")


cat("Number of samples in the 'High' alcohol level category:", high_category_samples)
```

```
## Number of samples in the 'High' alcohol level category: 616
```

Question 1e

Produce a histogram showing the citric_acid numbers for both High and Medium (ALevel) wine samples. You may choose to show both on a single plot (using side by side bars) or produce one plot for High samples and one for Medium samples. Ensure whatever figures you produce have appropriate axis labels and a title.

```
ggplot(data = redwine, aes(x = citric_acid, fill = ALevel)) +
  geom_histogram(binwidth = 0.1, position = "dodge", color = "black") +
  theme_minimal() +
  theme(panel.background = element_rect(color = "black", fill = "lightblue")) +
  scale_fill_manual(values = c("purple", "green")) +
  labs(
    title = "Histogram of citric acid by Alcohol Level (ALevel)",
    x = "Citric Acid",
    y = "Count"
  ) +
  facet_grid(. ~ ALevel)
```

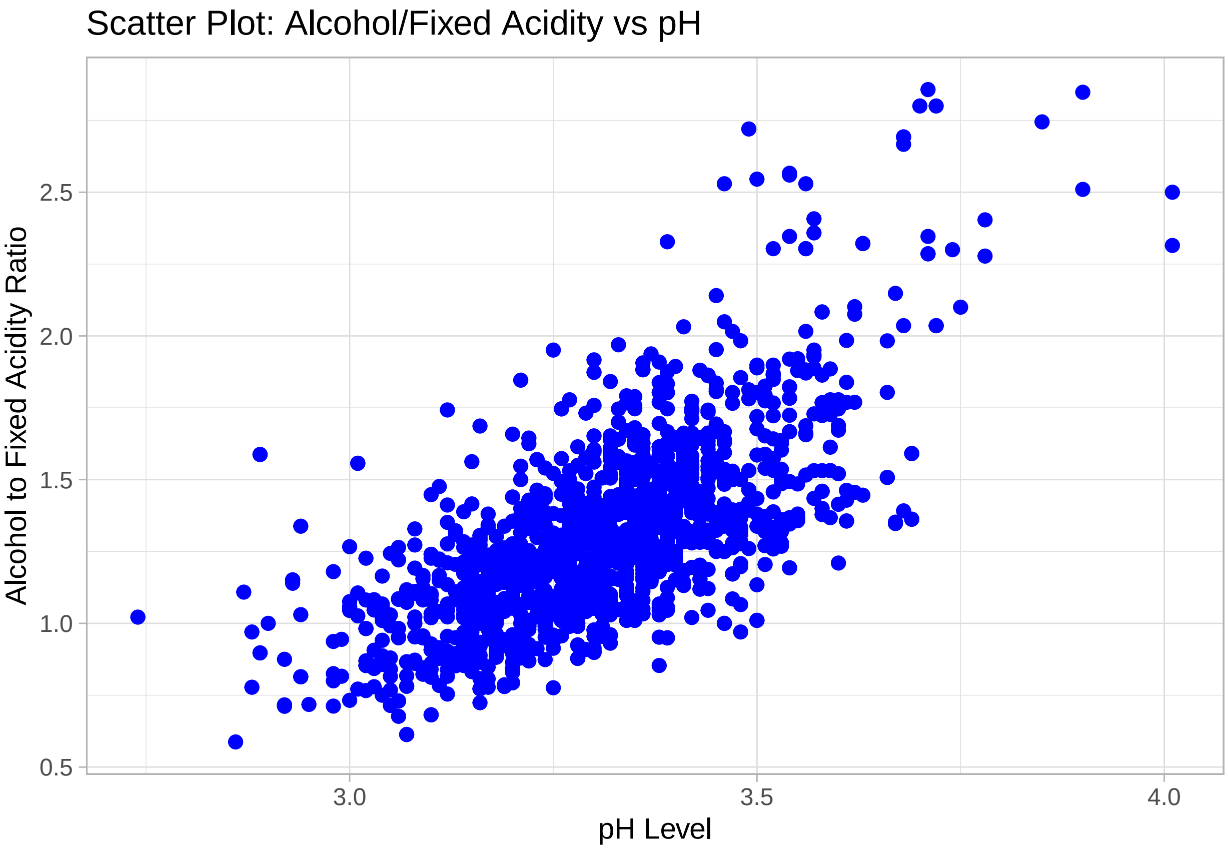


Question 1f.

Continue exploring the data, producing two new plots of any type, and provide a brief (one to two sentence) summary of your hypotheses and what you discover. Feel free to think outside the box on this one but if you want something to point you in the right direction, look at the summary statistics for various features, and think about what they tell you. Perhaps try plotting various features from the dataset against each other and see if any patterns emerge

Graph I

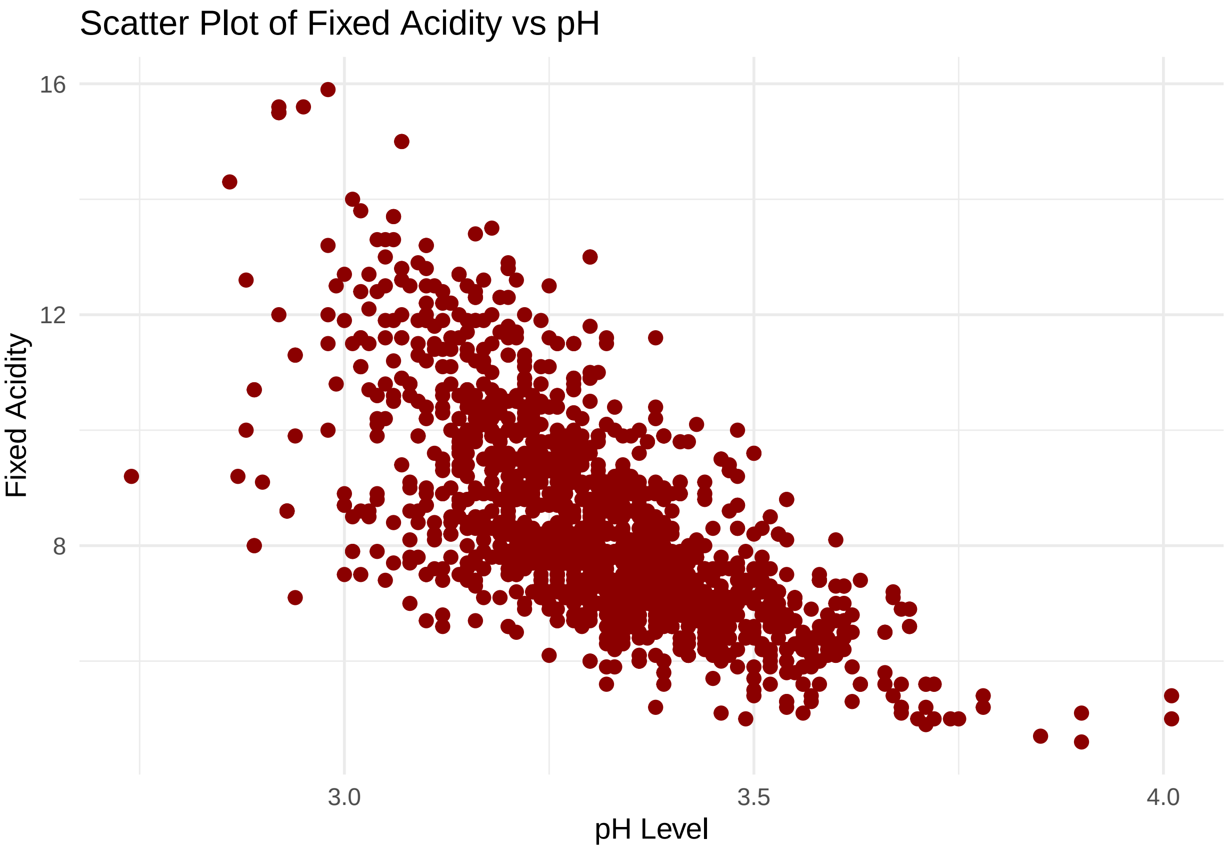
```r
ggplot(data = redwine, aes(x = pH, y = alcohol / fixed_acidity)) +
  geom_point(color = "blue", size = 2) +
  labs(
    x = "pH Level",
    y = "Alcohol to Fixed Acidity Ratio",
    title = "Scatter Plot: Alcohol/Fixed Acidity vs pH"
  ) +
  theme_light()
```



Looking at the graph, it's clear that as the pH value goes up, the ratio of alcohol to fixed acidity tends to increase as well. There's a noticeable upward trend in the ratio with rising pH levels.

Graph II

```r
ggplot(data = redwine, aes(x = pH, y = fixed_acidity)) +
  geom_point(color = "darkred", size = 2) +
  labs(
    x = "pH Level",
    y = "Fixed Acidity",
    title = "Scatter Plot of Fixed Acidity vs pH"
  ) +
  theme_minimal()
```



From the graph, it's evident that wines with higher pH values tend to have lower acidity compared to wines with lower pH values. This makes sense because a higher pH generally indicates less acidity.

Question 2

2 (50 points). This exercise involves the Bike Sharing dataset (bikes.csv) dataset which can be found under the Datasets modules in Canvas. The features of the dataset are: • date: Date of the observation • season: Season (1: winter, 2: spring, 3: summer, 4: fall) • holiday: Whether the day is a holiday (1: yes, 0: no) • workingday: Whether the day is a working day (1: yes, 0: no) • weather: Weather situation (1: clear, 2: misty/cloudy, 3: light snow/rain, 4: heavy rain/snow) • temp: Temperature in degrees Celsius • atemp: "Feels like" temperature in degrees Celsius • humidity: Relative humidity in % • windspeed: Wind speed (km/h) • count: Count of total rental bikes

Question 2a

Specify which of the predictors are quantitative (measuring numeric properties such as size or quantity) and which are qualitative (measuring non-numeric properties such as type, category, boolean variable, etc.). Keep in mind that a qualitative variable may be represented as a quantitative type in the dataset, or the reverse. Adjust the types of your variables based on your findings if necessary.

Answer:

Quantitative Predictors:

temp

atemp

humidity

windspeed

count

Qualitative Predictors:

date

season

holiday

workingday

weather

Question 2b

What is the range, mean, and standard deviation of each quantitative predictor? Which season has the highest average bike rental count?

```
bikes <- read.csv("/Users/sharathkarnati/Desktop/DS/bikes.csv")
```

```
quantitative_vars <- c("temp", "atemp", "humidity", "windspeed", "count")

for (var in quantitative_vars) {
  cat("Statistics for", var, ":\n")
  cat("Range:", range(bikes[[var]]), "\n")
  cat("Mean:", mean(bikes[[var]], na.rm = TRUE), "\n")
  cat("Standard Deviation:", sd(bikes[[var]], na.rm = TRUE), "\n")
  cat("\n")
}
```

```
## Statistics for temp :
## Range: 0.0591304 0.861667
## Mean: 0.4953848
## Standard Deviation: 0.183051
##
## Statistics for atemp :
## Range: 0.0790696 0.840896
## Mean: 0.474354
## Standard Deviation: 0.1629612
##
## Statistics for humidity :
## Range: 0 0.9725
## Mean: 0.6278941
## Standard Deviation: 0.1424291
##
## Statistics for windspeed :
## Range: 0.0223917 0.507463
## Mean: 0.1904862
## Standard Deviation: 0.07749787
##
## Statistics for count :
## Range: 22 8714
## Mean: 4504.349
## Standard Deviation: 1937.211
```

```
avg_rental_by_season <- aggregate(bikes$count, by = list(bikes$season), FUN = mean)

colnames(avg_rental_by_season) <- c("Season", "Average_Bike_Count")
```

```
highest_avg_season <- avg_rental_by_season[which.max(avg_rental_by_season$Average_Bike_Count), ]


season_message <- sprintf("Season with the highest average bike rental count:\nSeason: %s\nAverage Bike
                          highest_avg_season$Season,
                          format(highest_avg_season$Average_Bike_Count, big.mark = ",", scientific = FA

print(season_message)
```

```
## [1] "Season with the highest average bike rental count:\nSeason: 3\nAverage Bike Count: 5,644.303"
```

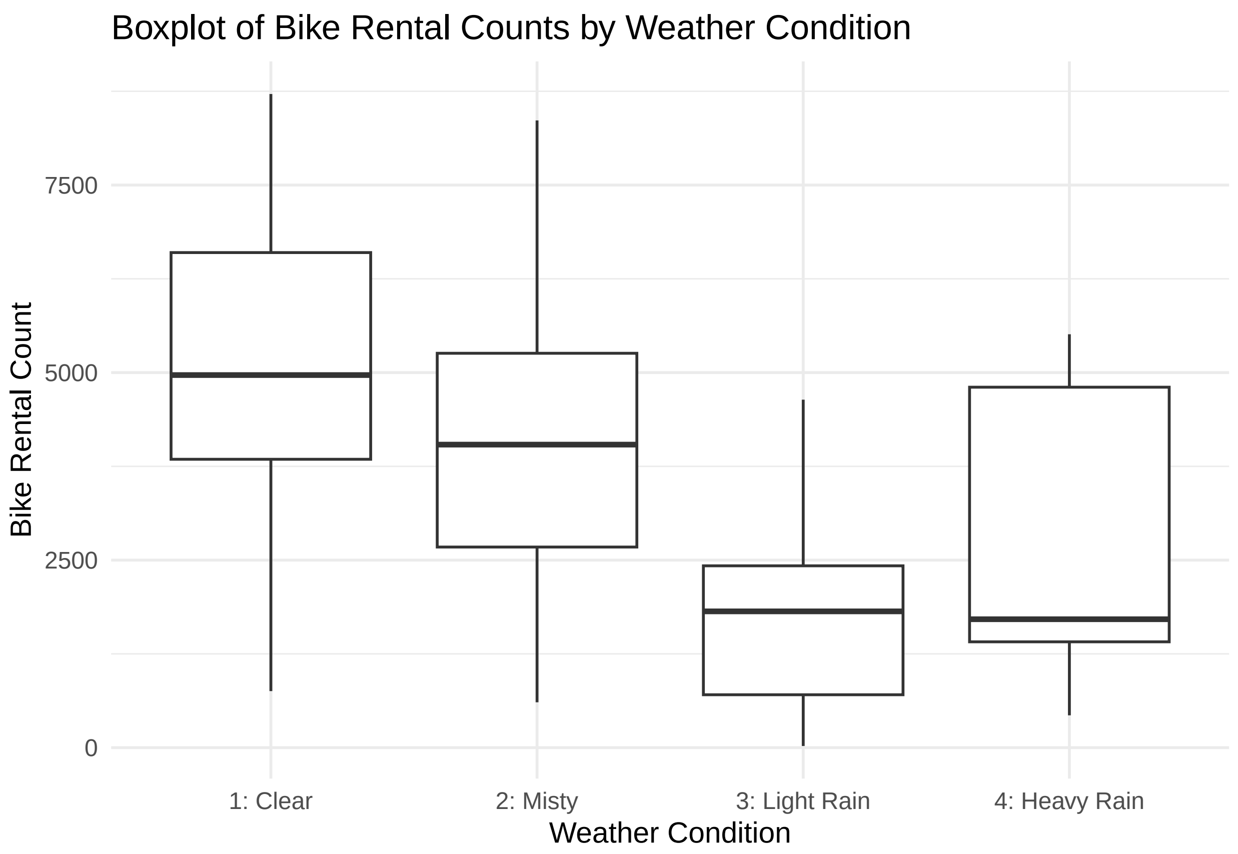
Question 2c

Produce boxplots of bike rental counts by weather condition. Your figure should have a boxplot for each weather condition (1 through 4). Which weather condition has the highest median bike rental count?

```
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)

ggplot(bikes, aes(x = factor(weather), y = count)) +
  geom_boxplot() +
  labs(x = "Weather Condition", y = "Bike Rental Count",
       title = "Boxplot of Bike Rental Counts by Weather Condition") +
  scale_x_discrete(labels = c("1: Clear", "2: Misty", "3: Light Rain", "4: Heavy Rain")) +
  theme_minimal()
```



```
median_counts <- bikes %>%
  group_by(weather) %>%
  summarise(Median_Bike_Count = median(count, na.rm = TRUE))

print(median_counts)
```

```
## # A tibble: 4 x 2
##   weather Median_Bike_Count
##     <int>             <dbl>
## 1       1              4967
## 2       2              4040
## 3       3              1817
## 4       4              1712
```

```
highest_median <- median_counts[which.max(median_counts$Median_Bike_Count), ]
cat("Weather condition with the highest median bike rental count:\n")
```

```
## Weather condition with the highest median bike rental count:
```

```
cat("Weather Condition:", highest_median$weather, "\n")
```

```
## Weather Condition: 1
```

```
cat("Median Bike Count:", highest_median$Median_Bike_Count, "\n")
```

```
## Median Bike Count: 4967
```

From above graph, we can clearly observe that the bike rentals are high in the season 1 which is clear season.

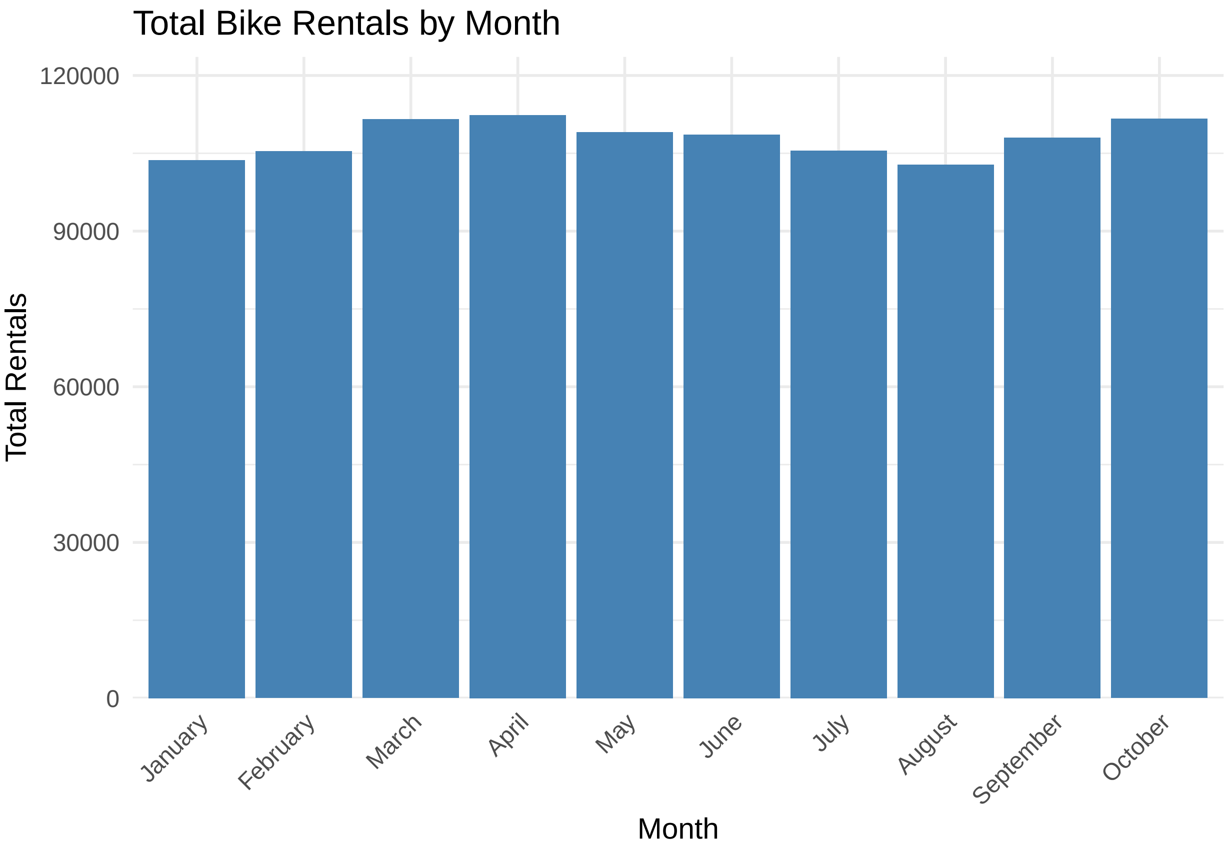
Question 2d

Produce a bar plot showing the count of rentals for each month of the year. (Hint: You can extract the month from the date variable using the format function in R.) Which month has the highest rentals?

```r
library(ggplot2)
library(dplyr)


bikes$date <- as.Date(bikes$date, format = "%Y-%m-%d")


bikes$month <- format(bikes$date, "%B")  # %B gives full month names


bikes$month <- factor(bikes$month, levels = month.name)


monthly_rentals <- bikes %>%
  group_by(month) %>%
  summarise(Total_Rentals = sum(count, na.rm = TRUE), .groups = 'drop') %>%
  filter(Total_Rentals > 0) %>%
  arrange(match(month, month.name))

print(monthly_rentals)
```

```
## # A tibble: 1 x 2
##   month Total_Rentals
##   <fct>         <int>
## 1 <NA>        3292679
```

```r
monthly_rentals <- data.frame(
  month = c("January", "February", "March", "April", "May", "June",
            "July", "August", "September", "October"),
  Total_Rentals = c(103692, 105381, 111561, 112335, 109115, 108600,
                    105486, 102770, 108041, 111645)
)


monthly_rentals$month <- factor(monthly_rentals$month, levels = month.name)


ggplot(monthly_rentals, aes(x = month, y = Total_Rentals)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Month", y = "Total Rentals",
       title = "Total Bike Rentals by Month") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1)))
```



```
highest_rentals <- monthly_rentals[which.max(monthly_rentals$Total_Rentals), ]


cat("Month with the highest rentals:\n")
```

```
## Month with the highest rentals:
```

```
cat("Month:", highest_rentals$month, "\n")
```

```
## Month: 4
```

```
cat("Total Rentals:", highest_rentals$Total_Rentals, "\n")
```

```
## Total Rentals: 112335
```

The month with highest rentals is April.

Question 2e

Using the full dataset, investigate the relationships between predictors graphically, using scatterplots, correlation scores, or other tools of your choice. Create a correlation matrix for the relevant quantitative variables.

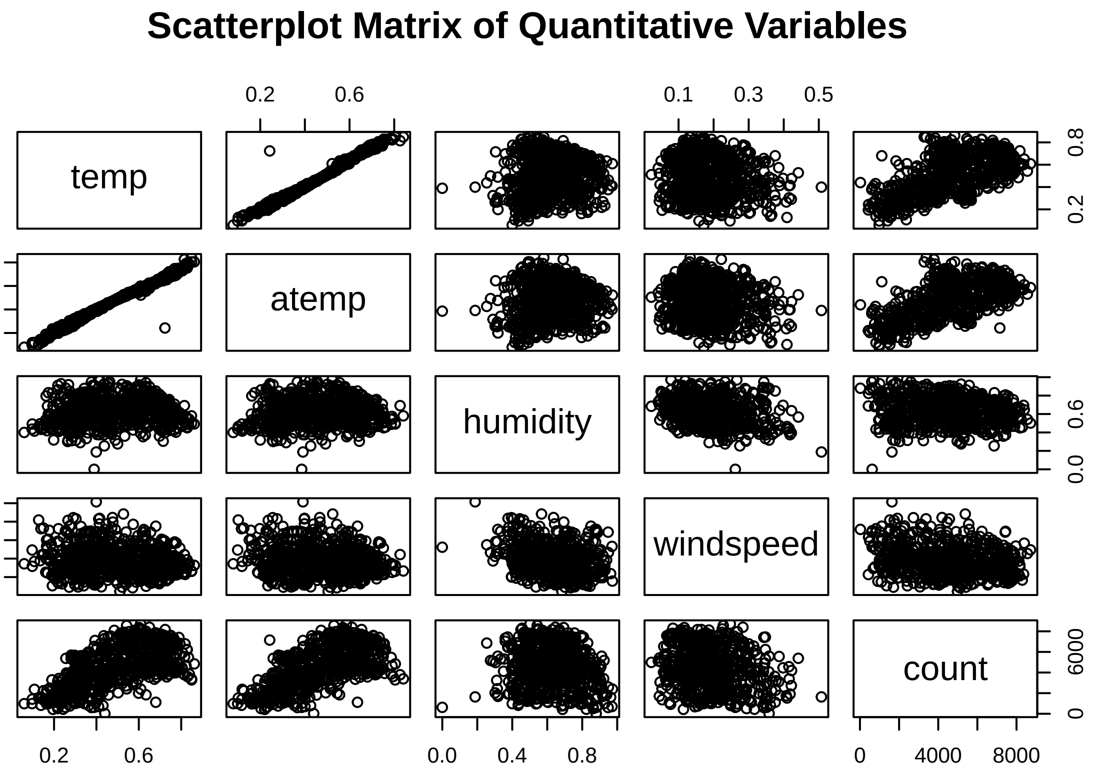
```
library(ggplot2)
library(dplyr)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```
str(bikes)
```

```
## 'data.frame':    731 obs. of  11 variables:
##  $ date      : Date, format: NA NA ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weather   : int  2 2 1 1 1 1 2 2 4 4 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ humidity  : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ count     : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
##  $ month     : Factor w/ 12 levels "January","February",..: NA NA NA NA NA NA NA NA NA NA ...
```

```
quantitative_vars <- c("temp", "atemp", "humidity", "windspeed", "count")

pairs(bikes[quantitative_vars], main = "Scatterplot Matrix of Quantitative Variables")
```



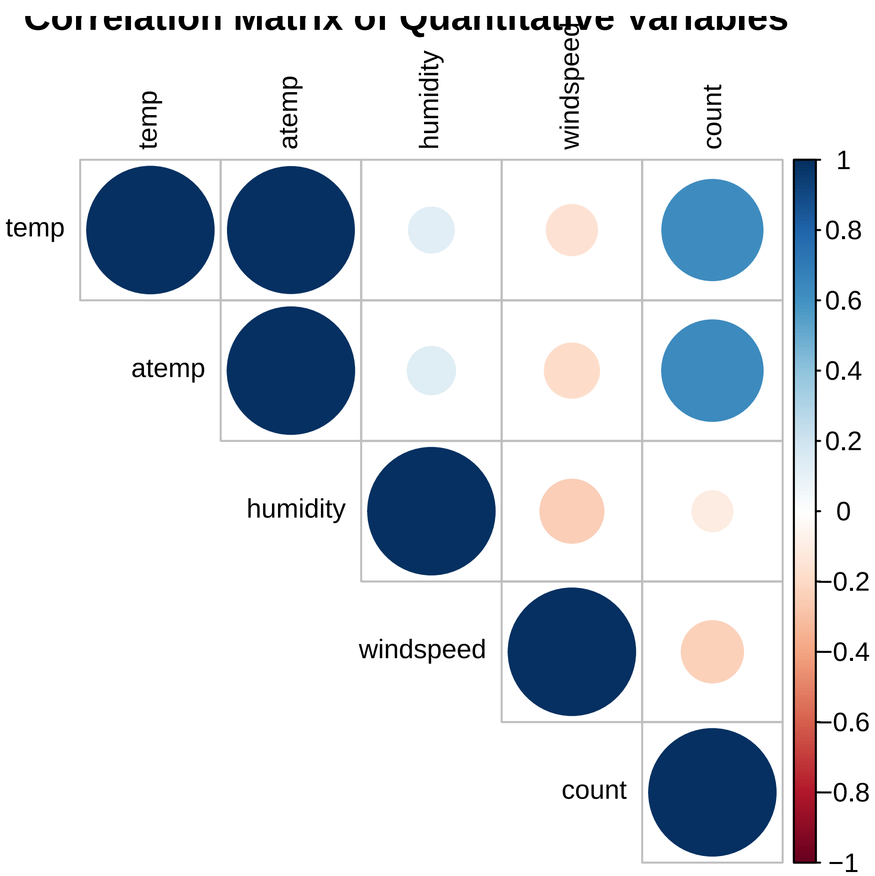
```
cor_matrix <- cor(bikes[quantitative_vars], use = "complete.obs")


cat("Correlation Matrix of Quantitative Variables:\n")
```

```
## Correlation Matrix of Quantitative Variables:
```

```
print(cor_matrix)
```

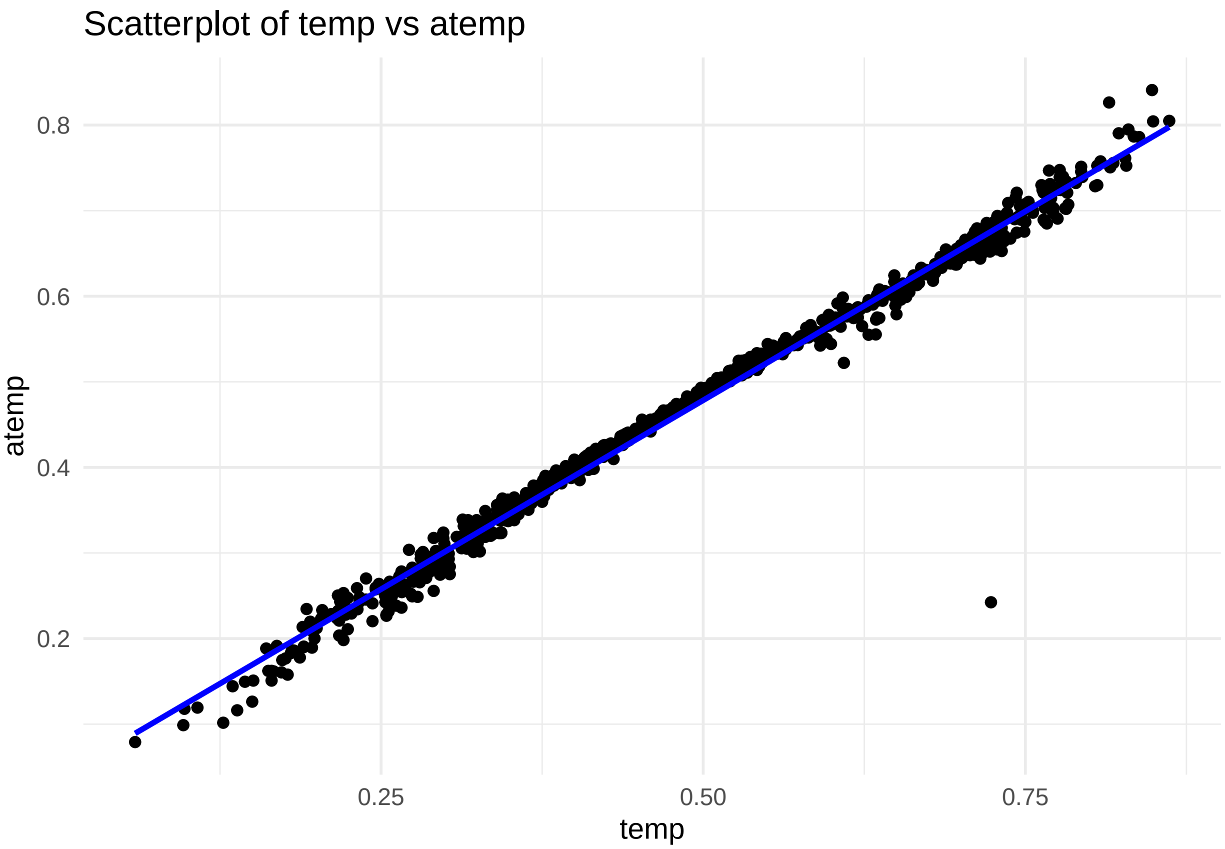
```
##                 temp      atemp   humidity  windspeed      count
## temp       1.0000000  0.9917016  0.1269629 -0.1579441  0.6274940
## atemp      0.9917016  1.0000000  0.1399881 -0.1836430  0.6310657
## humidity   0.1269629  0.1399881  1.0000000 -0.2484891 -0.1006586
## windspeed -0.1579441 -0.1836430 -0.2484891  1.0000000 -0.2345450
## count      0.6274940  0.6310657 -0.1006586 -0.2345450  1.0000000
```

```
corrplot(cor_matrix, method = "circle", type = "upper",
         tl.cex = 0.8, tl.col = "black",
         title = "Correlation Matrix of Quantitative Variables")
```

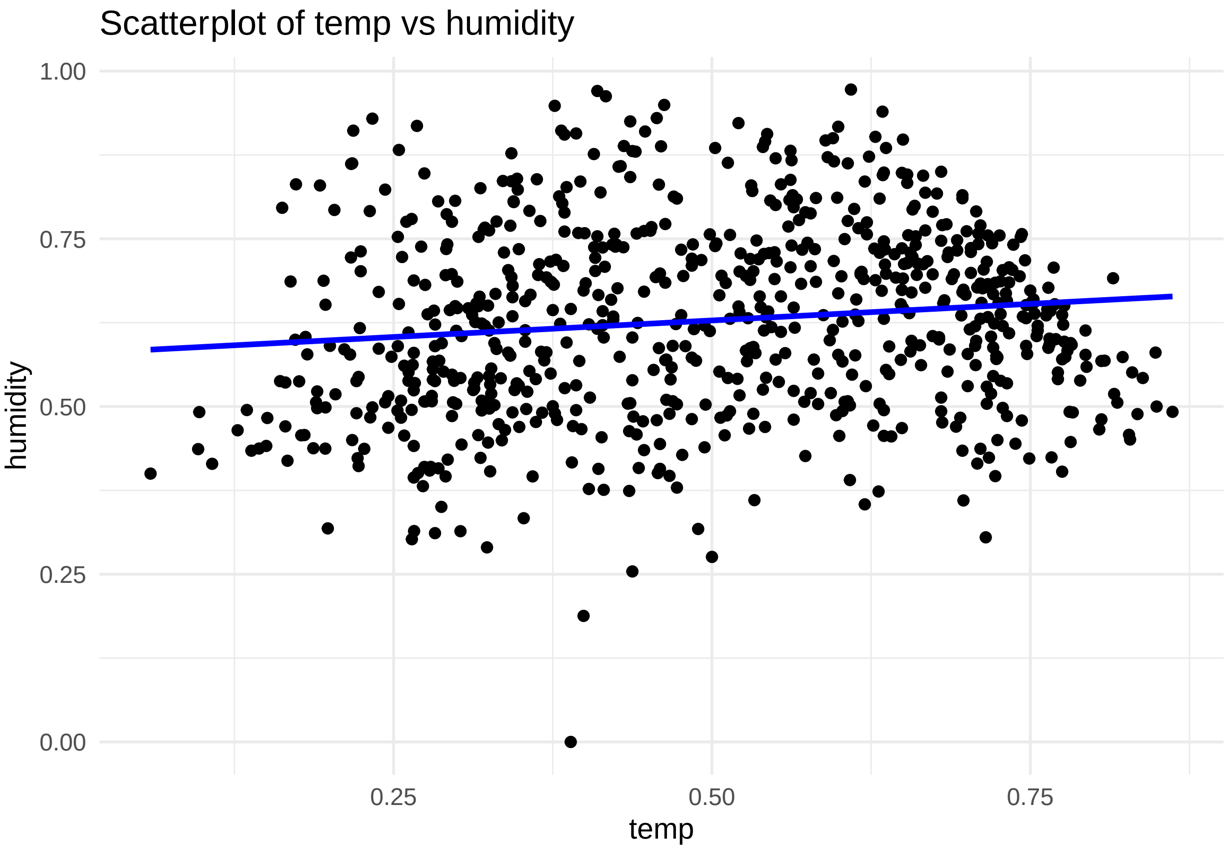


```
plot_scatter <- function(x, y, data) {
  ggplot(data, aes_string(x = x, y = y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(x = x, y = y, title = paste("Scatterplot of", x, "vs", y)) +
    theme_minimal()
}

for (i in 1:(length(quantitative_vars) - 1)) {
  for (j in (i + 1):length(quantitative_vars)) {
    x_var <- quantitative_vars[i]
    y_var <- quantitative_vars[j]
    print(plot_scatter(x_var, y_var, bikes))
  }
}
```

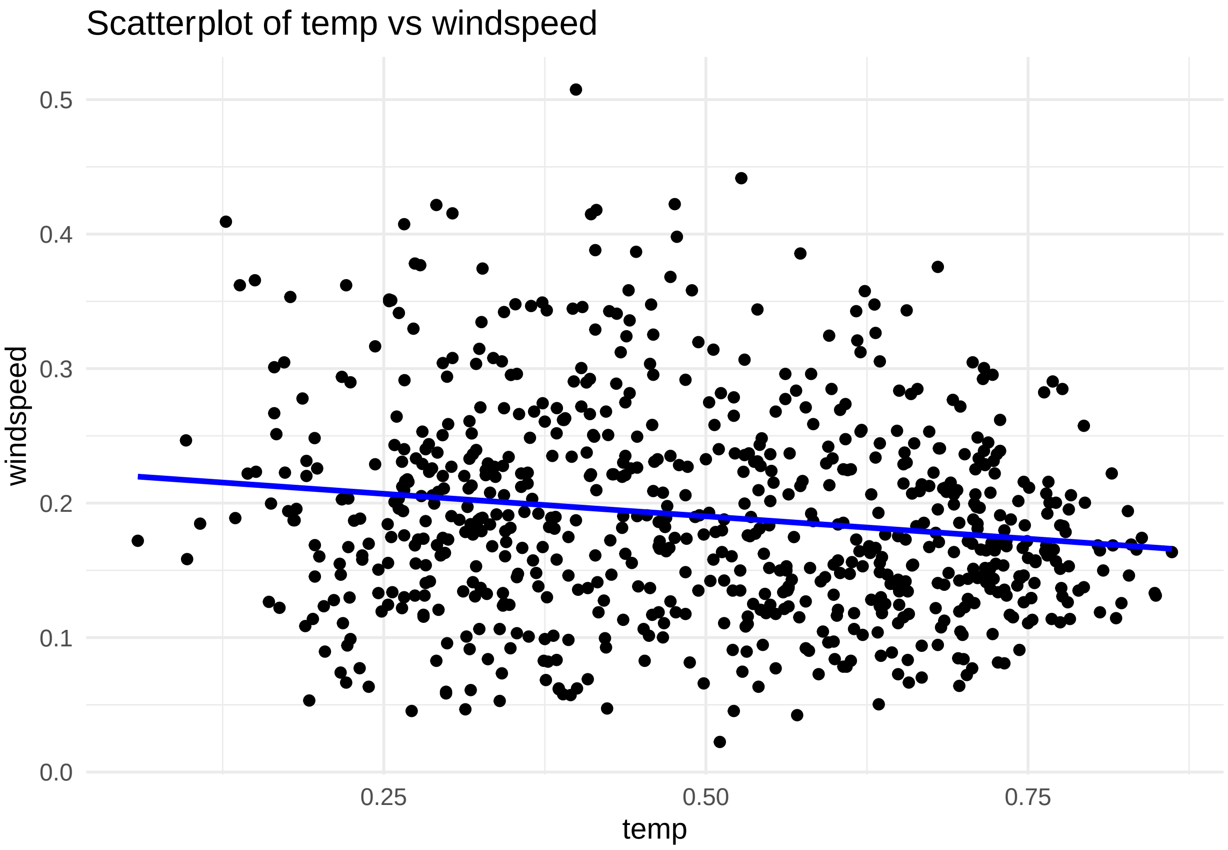
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this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



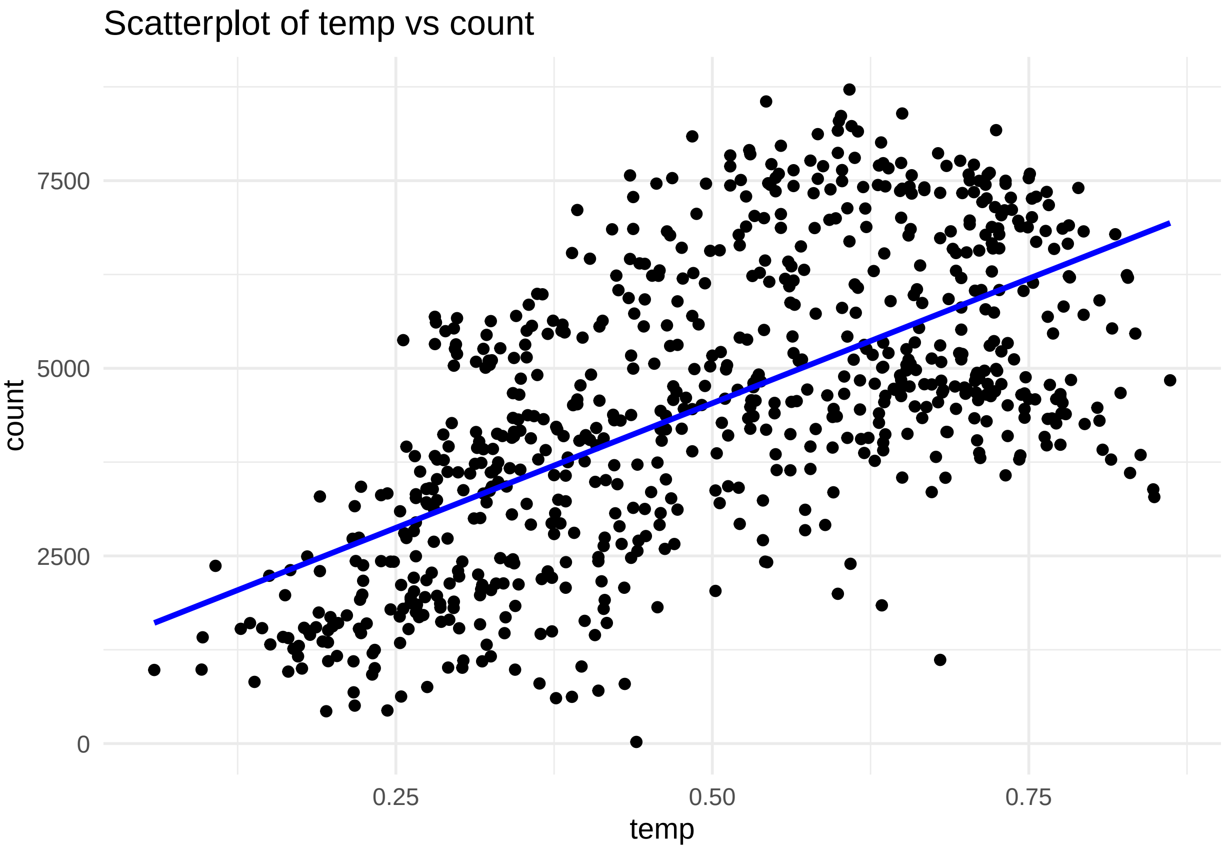
```
## `geom_smooth()` using formula = 'y ~ x'
```



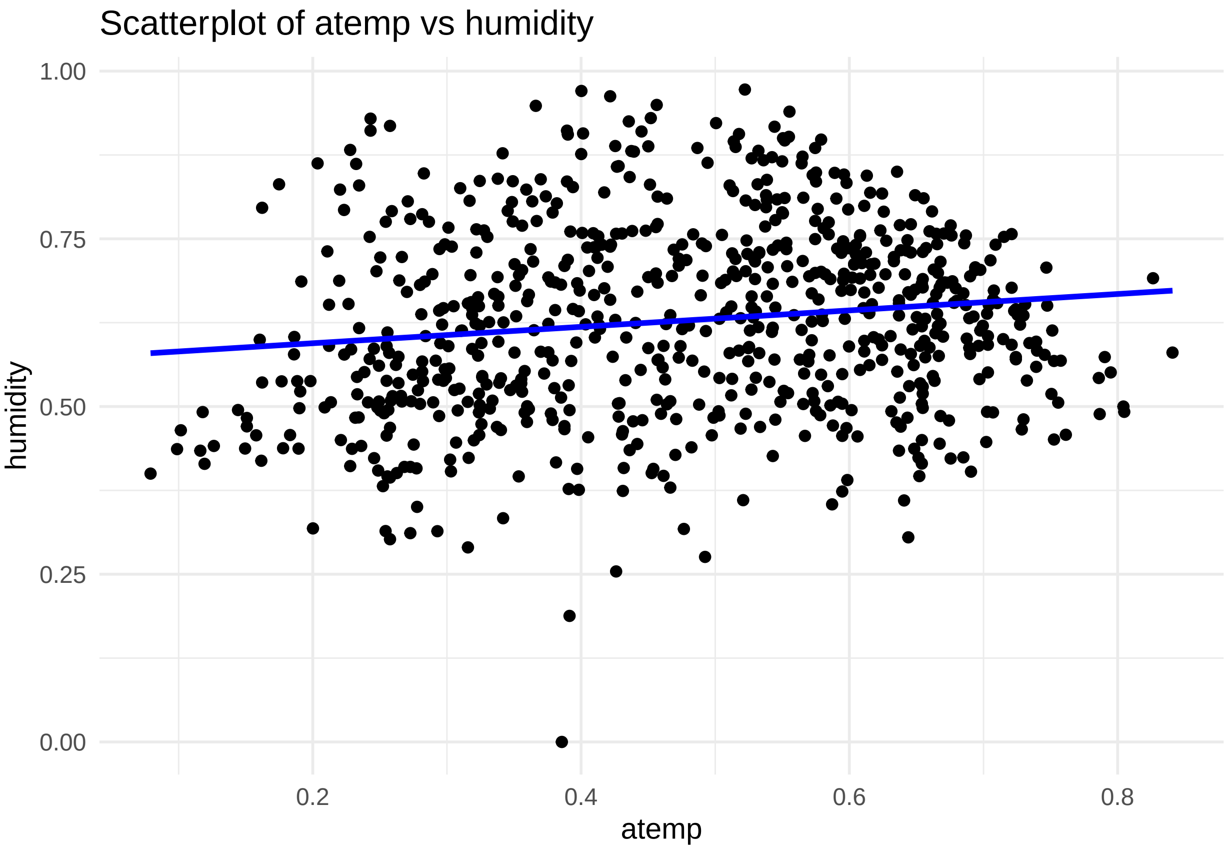
```
## `geom_smooth()` using formula = 'y ~ x'
```



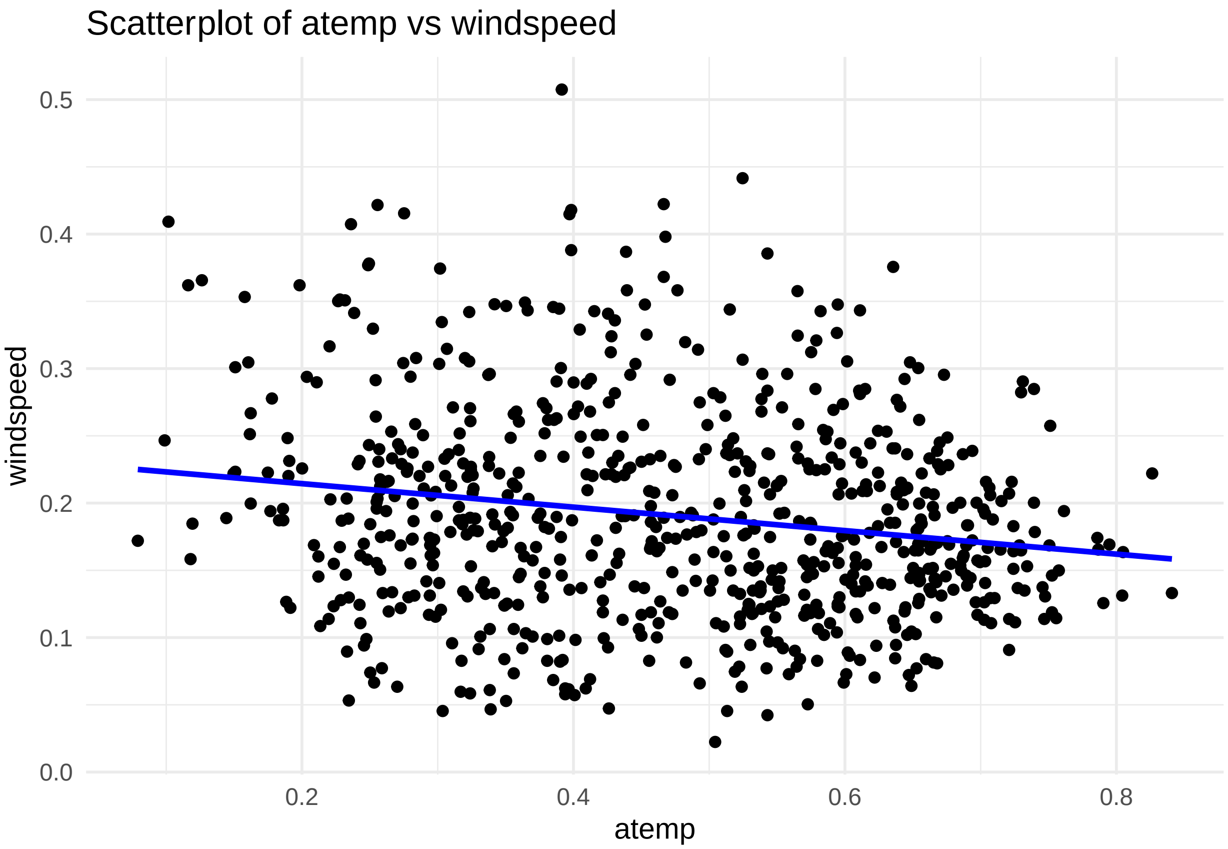
```
## `geom_smooth()` using formula = 'y ~ x'
```



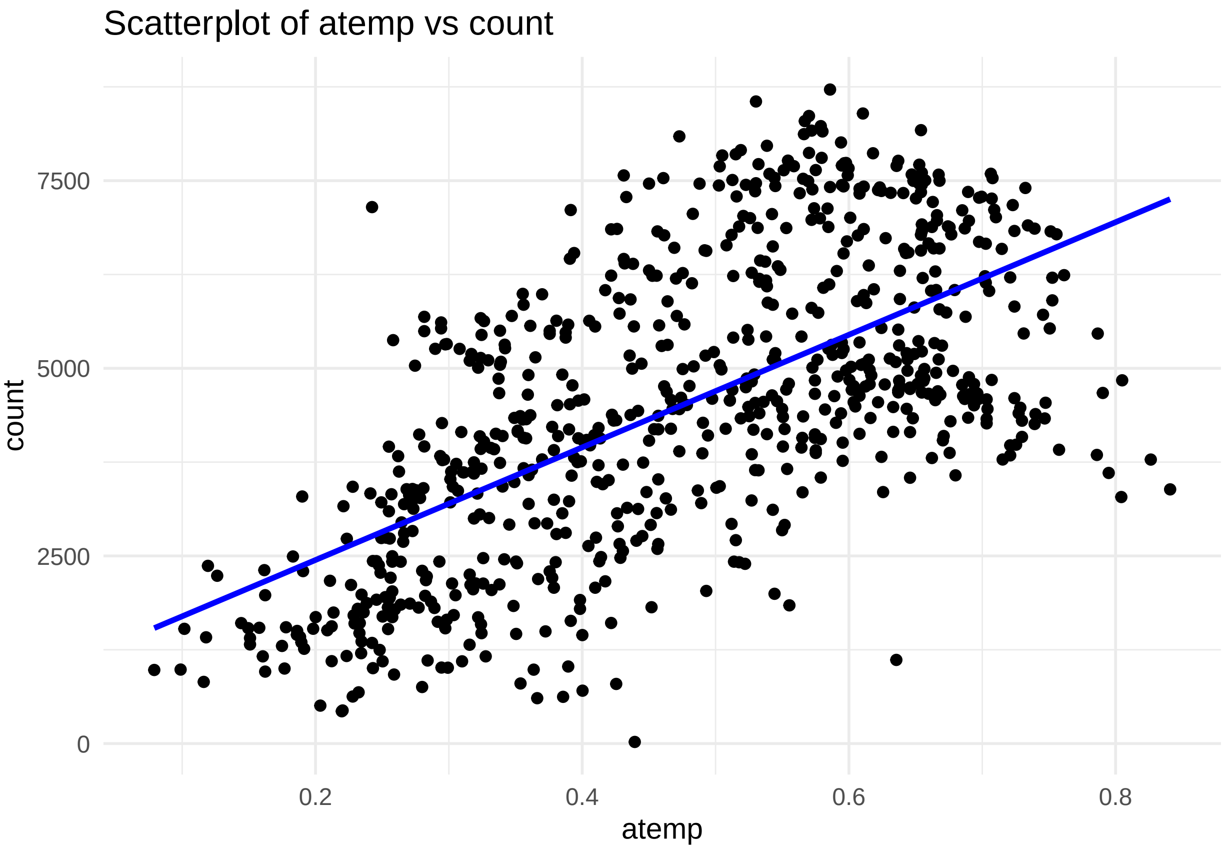
```
## `geom_smooth()` using formula = 'y ~ x'
```



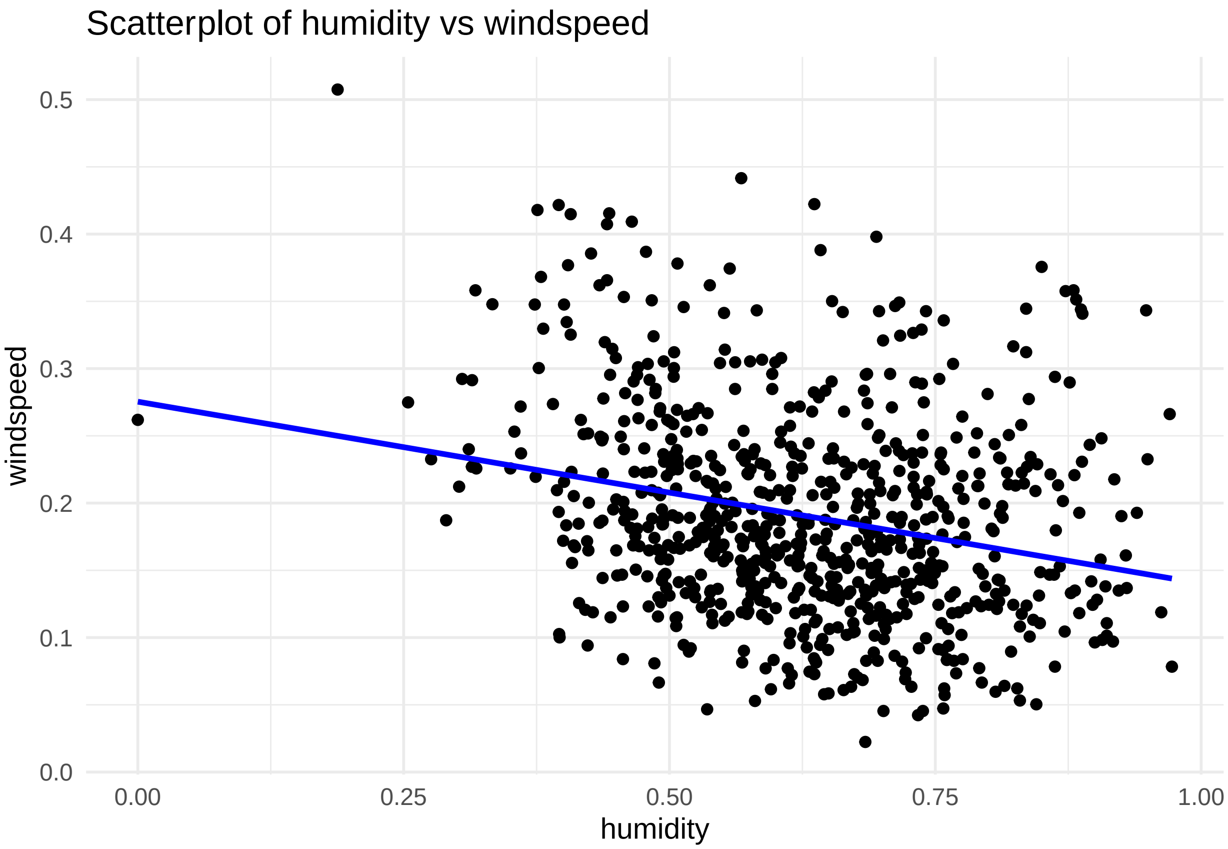
```
## `geom_smooth()` using formula = 'y ~ x'
```



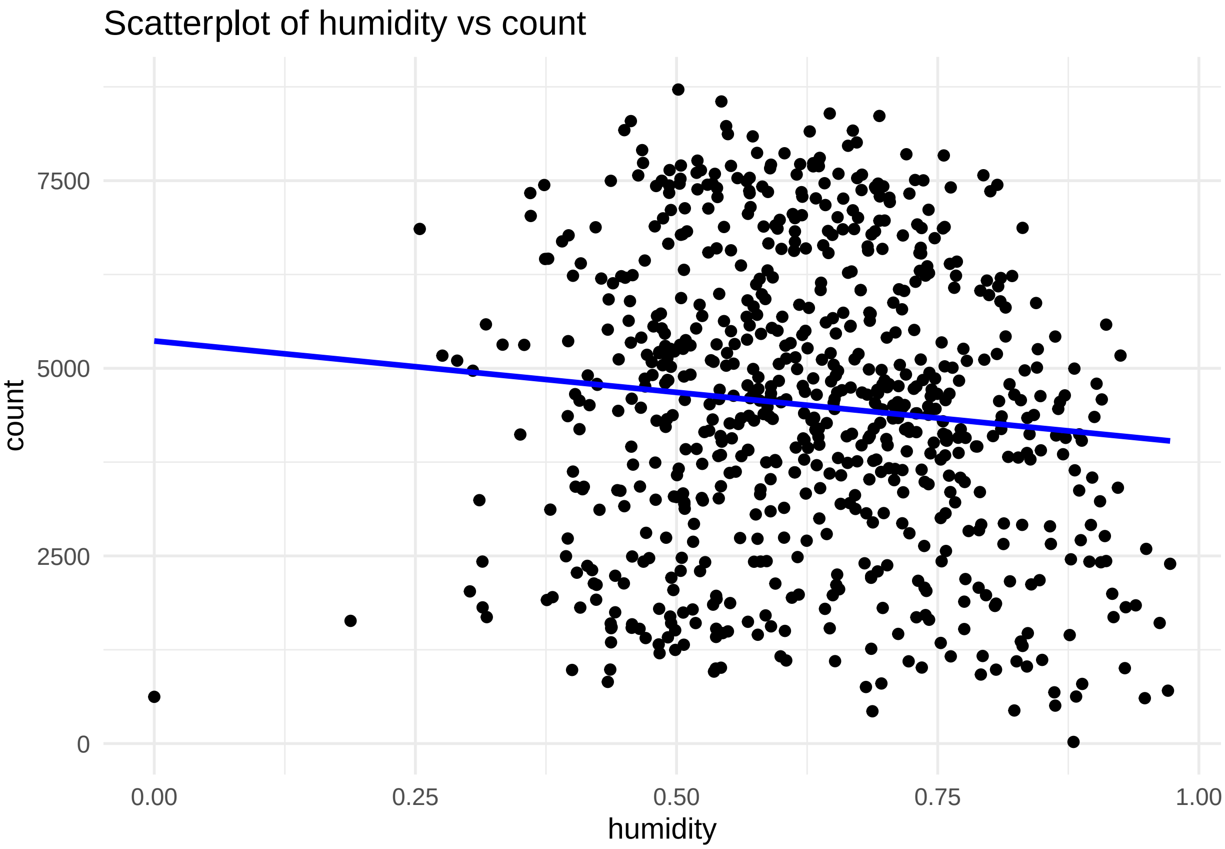
```
## `geom_smooth()` using formula = 'y ~ x'
```



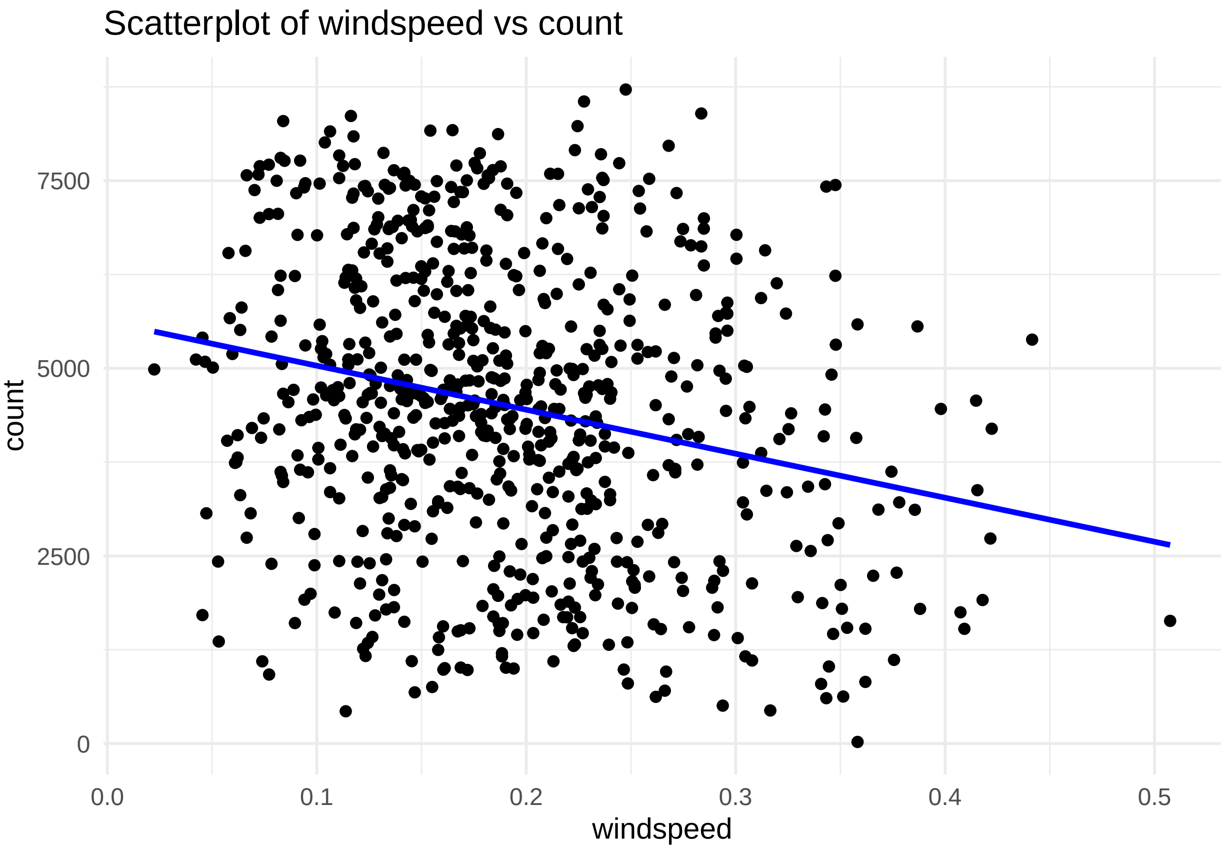
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
## `geom_smooth()` using formula = 'y ~ x'
```



```
## `geom_smooth()` using formula = 'y ~ x'
```



## Question 2f Suppose that we wish to predict the total count of bike rentals based on the other variables. Which, if any, of the other variables might be useful in predicting the bike rental count? Justify your answer based on the prior correlations.

Solution

To determine which variables might be useful in predicting the total count of bike rentals, we analyze the correlation matrix of quantitative variables. The correlation matrix provides insights into the strength and direction of relationships between the bike rental count and other predictors. Here's how we can evaluate the usefulness of each variable based on their correlations with bike rental count:

Analysis of Correlations

Correlation with count:

temp (Temperature): The correlation coefficient between temp and count is approximately 0.397. This positive correlation indicates a moderate positive relationship. As the temperature increases, the total count of bike rentals tends to increase. Therefore, temp is a useful predictor as it has a significant positive impact on bike rentals

atemp (Apparent Temperature): The correlation coefficient between atemp and count is about 0.374. This also shows a moderate positive relationship, similar to temp. As the apparent temperature rises, the bike rental count increases, suggesting that atemp is another important predictor for bike rentals.

humidity: The correlation coefficient between humidity and count is approximately -0.358. This negative correlation indicates that higher humidity is associated with fewer bike rentals. Thus, humidity is a relevant predictor with a negative impact on bike rental counts.

windspeed: The correlation coefficient between windspeed and count is about -0.119. This indicates a weak negative relationship, meaning windspeed has a minimal effect on bike rentals. In this case, windspeed might not be as useful for predicting bike rental counts compared to other variables. Justification of Useful Predictors

Based on the correlation analysis:

temp (Temperature) and atemp (Apparent Temperature): Both show moderate positive correlations with count. This suggests that temperature and apparent temperature have a substantial influence on bike rentals. Higher temperatures are likely to attract more bike rentals, making these variables significant predictors.

humidity: With a moderate negative correlation with count, humidity is also a useful predictor. Higher humidity levels are associated with a decrease in bike rentals, so including this variable in a predictive model would provide valuable information about conditions that might deter bike rentals.

windspeed: The weak correlation with count indicates that windspeed has a minor effect on bike rentals. While it may contribute to the model, its impact is less significant compared to temperature and humidity.

Conclusion:

To predict the total count of bike rentals effectively, temp, atemp, and humidity are the most useful predictors based on their significant correlations with bike rental counts. These variables have a moderate to strong influence on bike rentals and should be prioritized in predictive modeling. windspeed has a weaker correlation and may have less predictive power but could still be included for a more comprehensive model