Preprocessing, EDA and Feature Engineering

Jan Philip Richter

2023-07-12

Loading the required packages

```
library(tidyverse)
library(ggcorrplot)
library(patchwork)
library(moments)
```

Data set

The data set for the statistical analysis is the Seoul Bike Sharing Demand data, originally obtained from the SEOUL OPEN DATA PLAZA and can be downloaded at https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand.

The data set provides information about the number of bikes which are rented from the Seoul Public Bike service, as well as information about the weather conditions, for each hour of the day over a span of 365 days.

The goal of this analysis is to predict the number of bikes which are rented at a given hour, given the current weather conditions during that time.

Variables The variables contained in the data set are:

- Date: The date of the observation
- Rented Bike Count: The number of bikes rented
- Hour: The hour of the day
- Temperature: The temperature measured in Celsius
- Humidity: The air humidity measured in %
- Wind speed: The wind speed measured in metres per second
- Visibility: The visibility measured in 10 metres
- Dew point temperature: The dew point temperature measured in Celsius
- Solar Radiation: The solar radiation measured in millijoule per square metre
- Rainfall: The rainfall measured in millimetres
- Snowfall: The snowfall measured in centimetres
- Seasons: The season of the year (Spring, Summer, Autumn, Winter)
- Holiday: Indicator if the day was a public holiday
- Functioning Day: Indicator if the bike sharing service was available on the day

```
bike <- read.csv(paste0("/Users/philip/Documents/Milano University/",
                        "2. Semester/Statistical Learning/",
                        "Statistical Learning Project/",
                        "SeoulBikeSharing/SeoulBikeData.csv"),
                 check.names = FALSE)
```

Loading the dataset The column names of the data set contain characters which raise an error, which is why we set check.names = FALSE.

Renaming Columns We remove the troublesome characters and then rename the columns to a simpler description.

```
iconv(names(bike), to = "ASCII", sub = "")

bike <- bike %>%
  rename("date" = "Date",
         "count" = "Rented Bike Count",
         "hour" = "Hour",
         "temperature" = "Temperature(\xb0C)",
         "humidity" = "Humidity(%)",
         "wind_speed" = "Wind speed (m/s)",
         "visibility" = "Visibility (10m)",
         "dp_temperature" = "Dew point temperature(\xb0C)",
         "solar_radiation" = "Solar Radiation (MJ/m2)",
         "rainfall" = "Rainfall(mm)",
         "snowfall" = "Snowfall (cm)",
         "seasons" = "Seasons",
         "holiday" = "Holiday",
         "functioning_day" = "Functioning Day"
         )
```

Data Classes Now we can take a look at the data classes and values of our variables to see if we have to apply further changes.

```
glimpse(bike)
```

```
## Rows: 8,760
## Columns: 14
## $ date            <chr> "01/12/2017", "01/12/2017", "01/12/2017", "01/12/2017"~
## $ count           <int> 254, 204, 173, 107, 78, 100, 181, 460, 930, 490, 339, ~
## $ hour            <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ~
## $ temperature     <dbl> -5.2, -5.5, -6.0, -6.2, -6.0, -6.4, -6.6, -7.4, -7.6, ~
## $ humidity        <int> 37, 38, 39, 40, 36, 37, 35, 38, 37, 27, 24, 21, 23, 25~
## $ wind_speed      <dbl> 2.2, 0.8, 1.0, 0.9, 2.3, 1.5, 1.3, 0.9, 1.1, 0.5, 1.2,~
## $ visibility      <int> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, ~
## $ dp_temperature  <dbl> -17.6, -17.6, -17.7, -17.6, -18.6, -18.7, -19.5, -19.3~
## $ solar_radiation <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.01, ~
## $ rainfall        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ snowfall        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ seasons         <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Win~
## $ holiday         <chr> "No Holiday", "No Holiday", "No Holiday", "No Holiday"~
## $ functioning_day <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"~
```

The numerical variables were all automatically assigned with an appropirate data class, whereas we need to change the data classes of the date column and the categorical variables.

```
bike$date <- as.Date(bike$date, format = "%d/%m/%Y")
bike$holiday <- as.factor(bike$holiday)
bike$functioning_day <- as.factor(bike$functioning_day)
bike$holiday <- as.factor(bike$holiday)
bike$seasons <- as.factor(bike$seasons)
```

Missing values and duplicates Checking for missing values

```r
sum(is.na(bike))
```

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

Checking for duplicates

```r
sum(duplicated(bike))
```

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

There are no missing values or duplicates in the data set.

Removing non functioning days The goal if this analysis is to predict the number of bikes rented at a given hour. The column **functioning_day** indicates if the service was available at a given day or not. If the service is not available the number of rented bikes is obviously 0 and if the service is available there is no information about the number of rented bikes to be gathered solely from that fact. The column is therefore removed, so our problem trivially changes to now predict the amount of bikes rented, given that the bike sharing service is available.

```r
bike <- bike[bike$functioning_day != "No",]
```

Exploratory Data Anlysis, Data Transformation and Feature Engineering

In this section we take a look at the individual variables contained in the data set, check the values for plausibility, check the respective distributions to decide whether some form of data transformation is necessary and if some further feature engineering seems appropriate and useful.

Count

Summary-Statistics of Count

```r
summary(bike$count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0   214.0   542.0   729.2  1084.0  3556.0
```

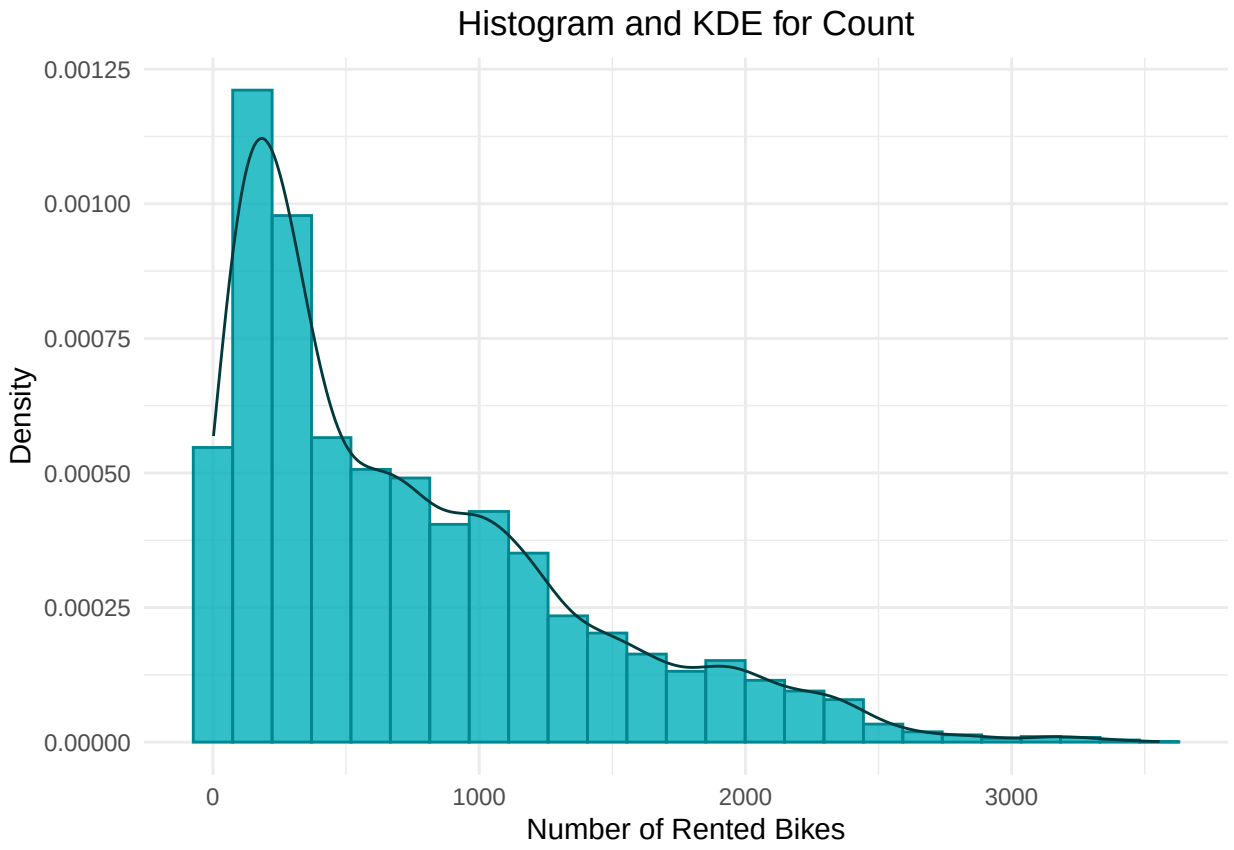
The maximum number of bikes rented seems plausible, while the lowest number of bikes rented in an hour appears to be extremely small. There could be potential outliers in the data set, which could be explained by the bike service not functioning properly at that given time, or some other unknown exogenous circumstances. But the extremely low minimum could also be logically explained by extreme weather or the time of day. For now we don't remove any of the observations as even the extreme points lie within the realm of plausibility.

Distribution of Count

We take a look at the distribution of the Count variable to evaluate if some data-transformations are necessary. For this we use a histogram and a kernel density estimation. For the KDE we use a gaussian kernel and Silverman's rule of thumb for the bandwidth.

```r
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$count)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 25) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
```

```
  labs(x = "Number of Rented Bikes",
       y = "Density",
       title = "Histogram and KDE for Count") +
  theme(plot.title = element_text(hjust = 0.5))
```

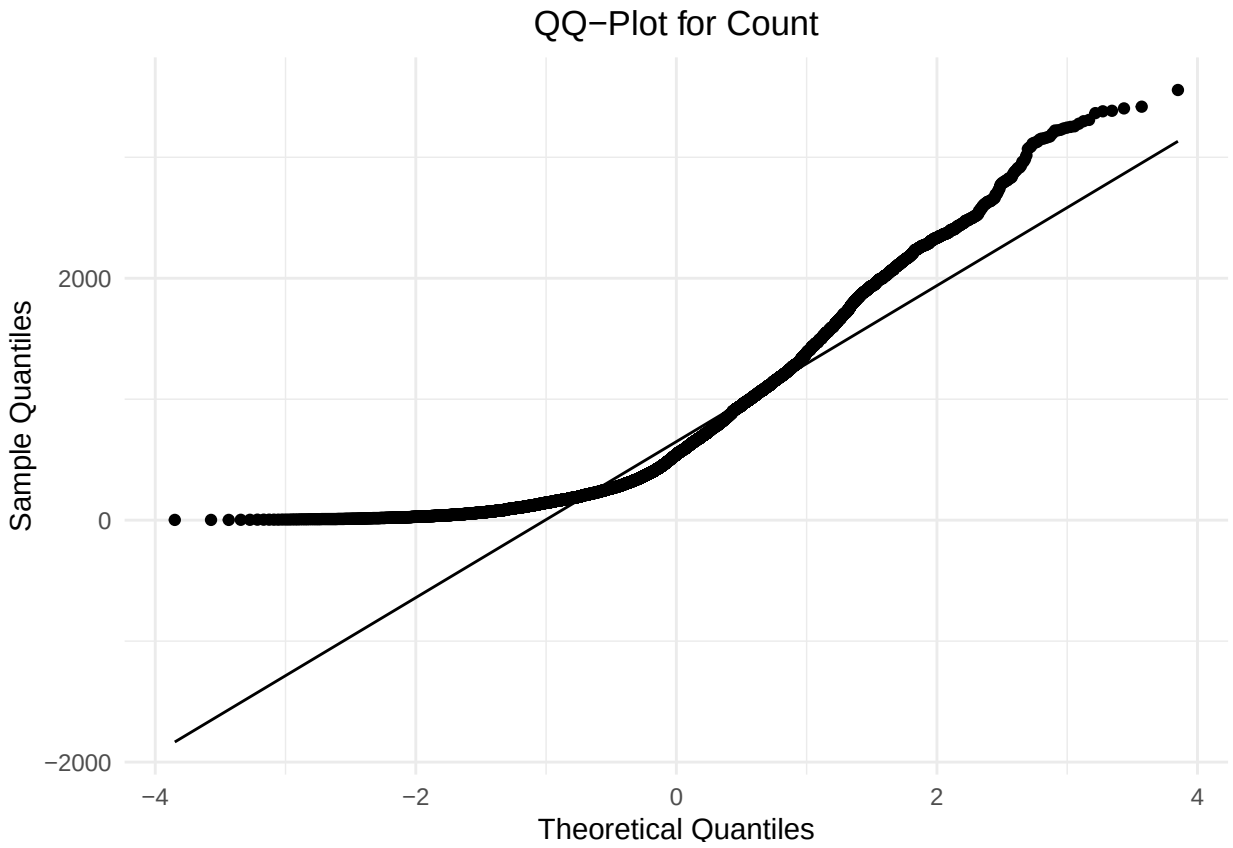


Skewness of Count

```
skewness(bike$count)
```

```
## [1] 1.139498
```

```
ggplot(data = bike, mapping = aes(sample = count)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for Count") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

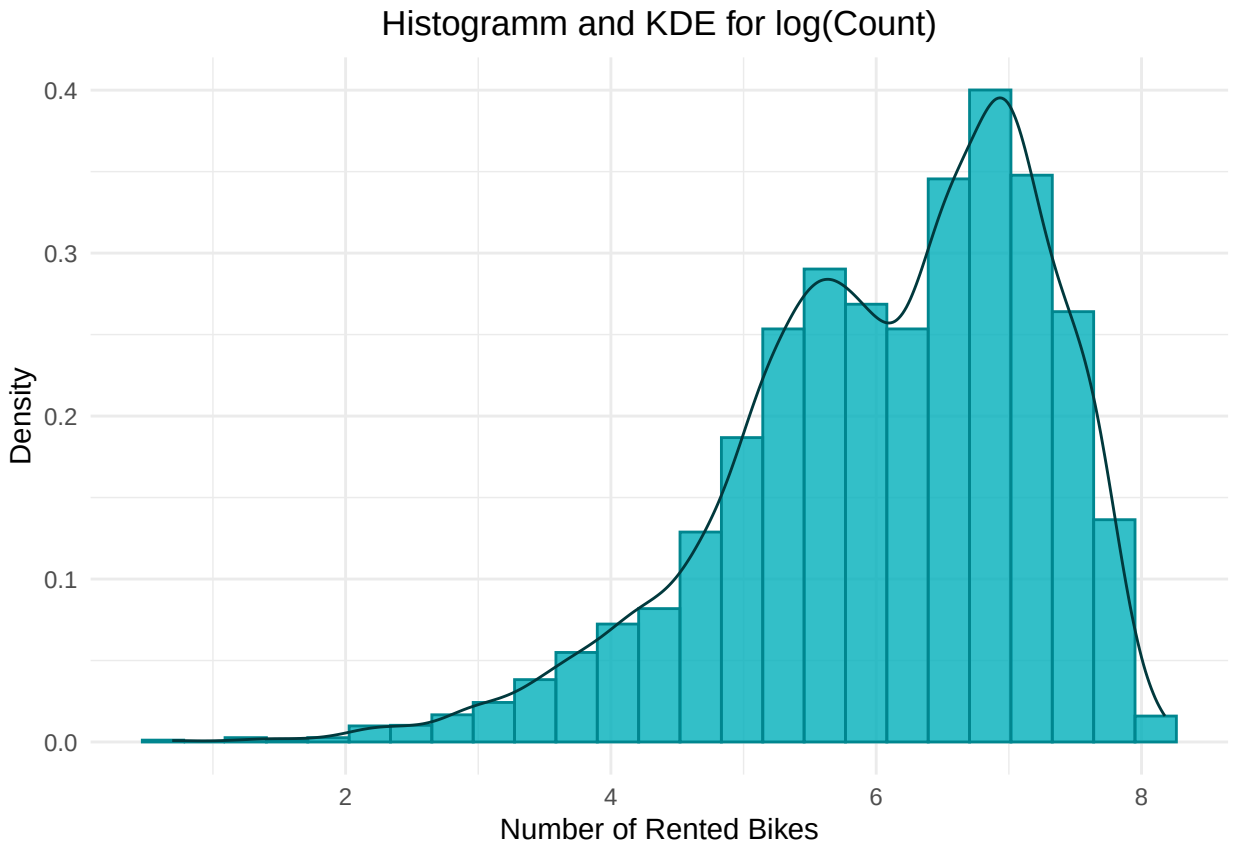


We can see that the distribution appears to right-skewed and has an estimated moment coefficient of skewness of 1.14. The QQ-Plot also shows, that the distribution is far away from a normal distribution, so it seems appropriate to use a log-transformation to obtain a distribution which is closer to a normal distribution, to improve the results for the regression analysis later on.

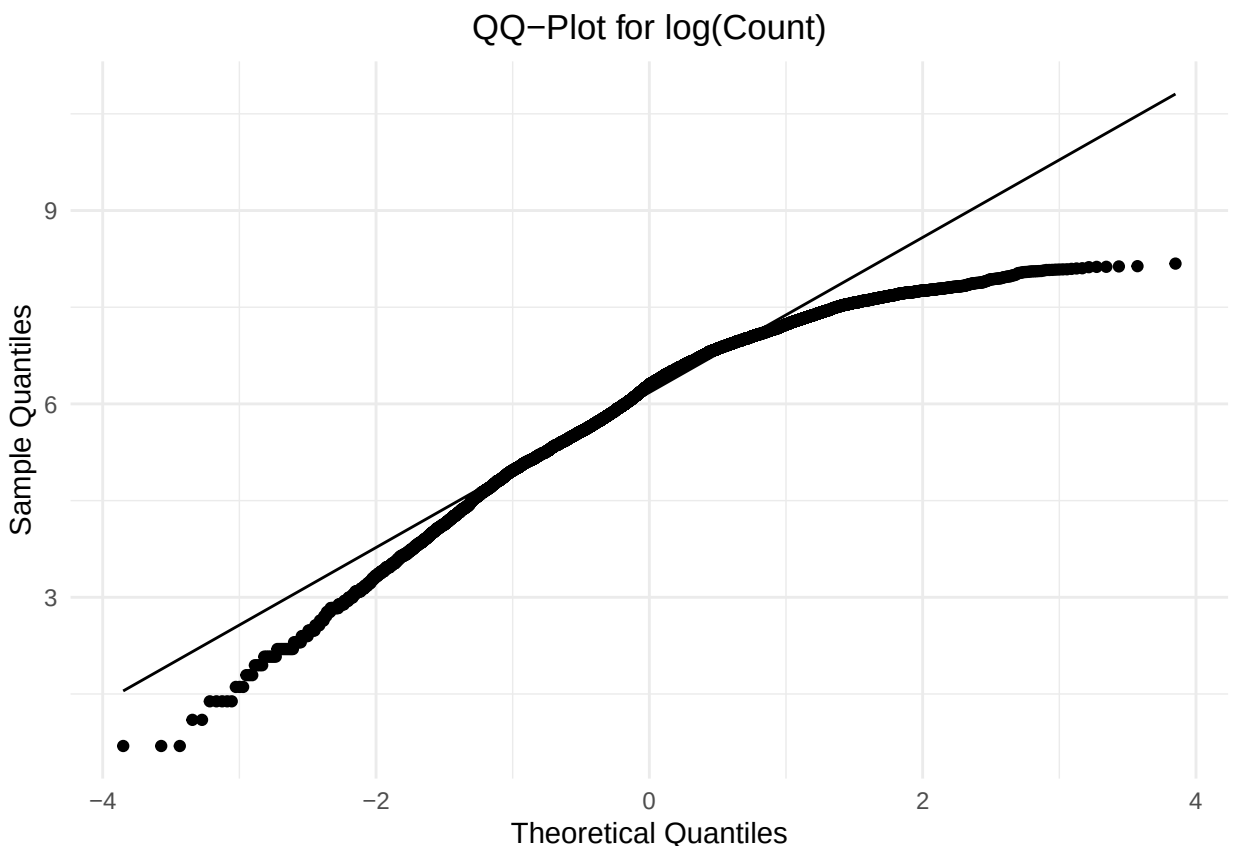
Log-transformation and new distribution

```
bike$log_count <- log(bike$count)

# New distribution
ggplot(bike, mapping = aes(x = bike$log_count)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 25) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "Number of Rented Bikes",
       y = "Density",
       title = "Histogramm and KDE for log(Count)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = log_count)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for log(Count)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# New Skewness
skewness(bike$log_count)
```

```
## [1] -0.8037832
```

After the log-transformation the data is still skewed, albeit now left-skewed, and not normally distributed but the skewness, as well as the variance, have been significantly reduced, which will lead to better predictability in the regression analysis.

Hour

Unique values of Hour

```
unique(bike$hour)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

There are no implausible values for the hour of the day.

Cyclical-Encoding of Hour The hour of day could be a could predictor when trying to estimate the number of bikes rented at a given time, as it would seem plausible that the number of rentals decrease in the night for example.

But using the hour of day as it is, would not be a good predictor, as we would lose the cyclical property of the hour of day. Hour 0 is furthest away from hour 23 in absolute values, but hour 0 comes right after hour 23 in the cyclical path of time.

Treating the hours of day as a categorical feature is also not an appropriate encoding method as all measurements of distance between the different hours get lost.

To counter these problems we use Sine-Cosine-Encoding as a method of cyclical encoding, which gives us two new features, which together represent the hour of the day.

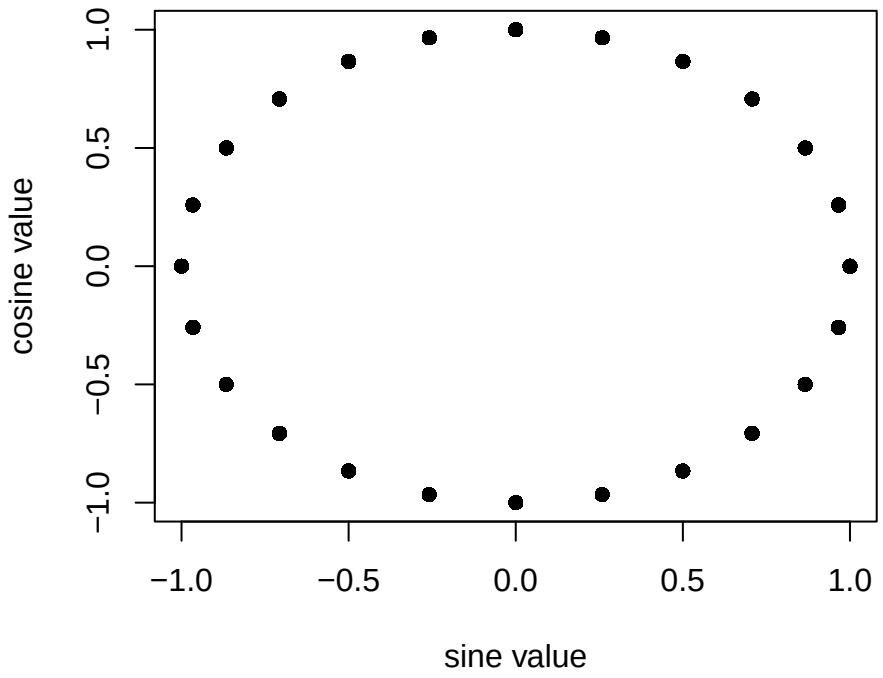
$$x_{sin} = sin(\frac{2\pi * hour}{24})$$

$$x_{cos} = cos(\frac{2\pi * hour}{24})$$

```
bike$sin_hour = sin(2 * pi * bike$hour / 24)
bike$cos_hour = cos(2 * pi * bike$hour / 24)
```

The graphical representation of our new features for the time of day now resembles a clock with the 24 different possible times

```
plot(bike$sin_hour, bike$cos_hour,
     xlab = "sine value",
     ylab = "cosine value",
     pch = 16)
```



Date

From the date column we could extract a number of possible predictors. To keep it simple we only extract the day of the week as a possible predictor and encode it in the same way as we did for the hour of the day.

```
bike$week_day <- wday(bike$date)

bike$sin_dow = sin(2 * pi * bike$week_day / 7)
bike$cos_dow = cos(2 * pi * bike$week_day / 7)
```

Temperature

Summary-Statistics of Temperature

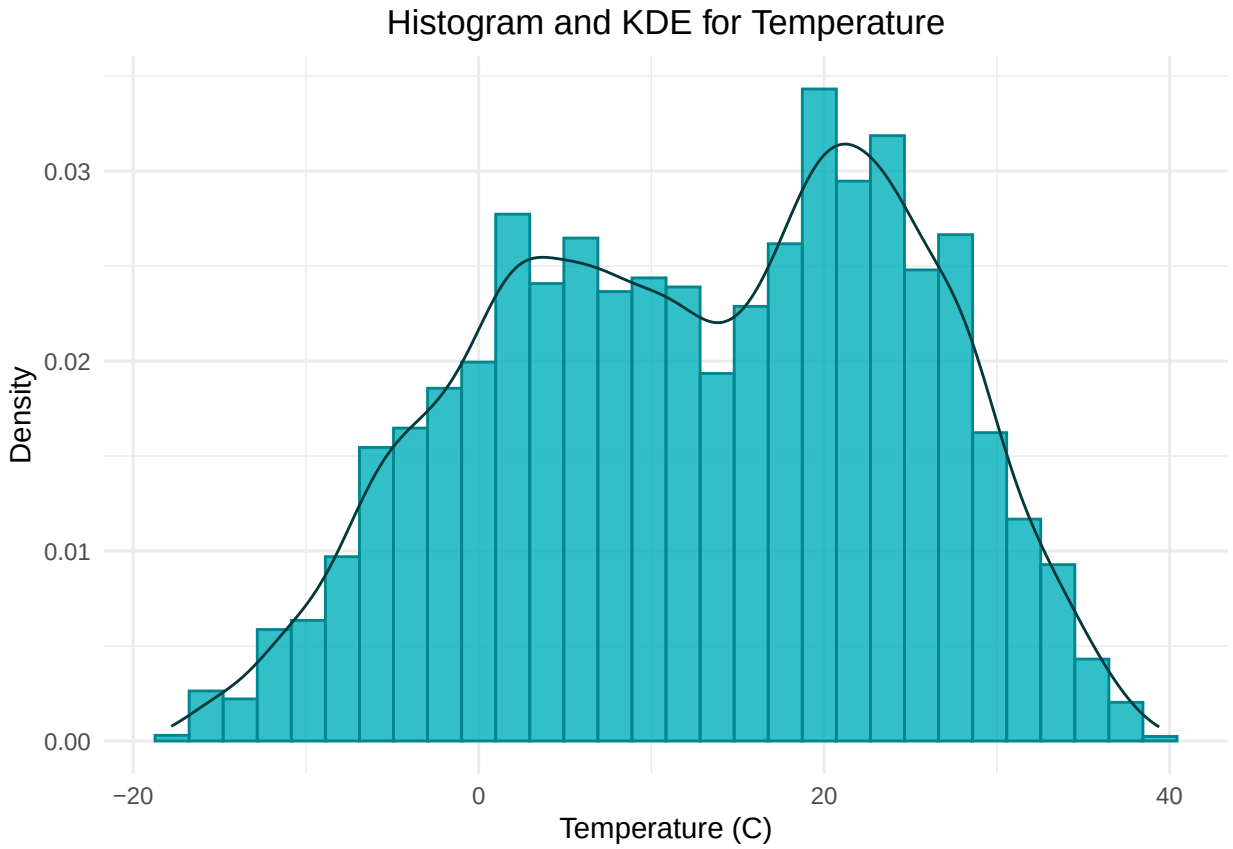
```
summary(bike$temperature)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -17.80    3.00   13.50   12.77   22.70   39.40
```

The weather in Seoul can be fairly extreme compared to central european levels. So neither extremely hot temperatures, like almost 40C, nor extremely low temperatures, like almost -20C are unusual.

Distribution of Temperature

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$temperature)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "Temperature (C)",
       y = "Density",
       title = "Histogram and KDE for Temperature") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Skewness
skewness(bike$temperature)
```

```
## [1] -0.1745189
```

The data looks somewhat normally distributed and has a fairly low skewness, so no transformation is necessary.

Dew-point Temperature

Summary-Statistics of Dew-Point Temperature

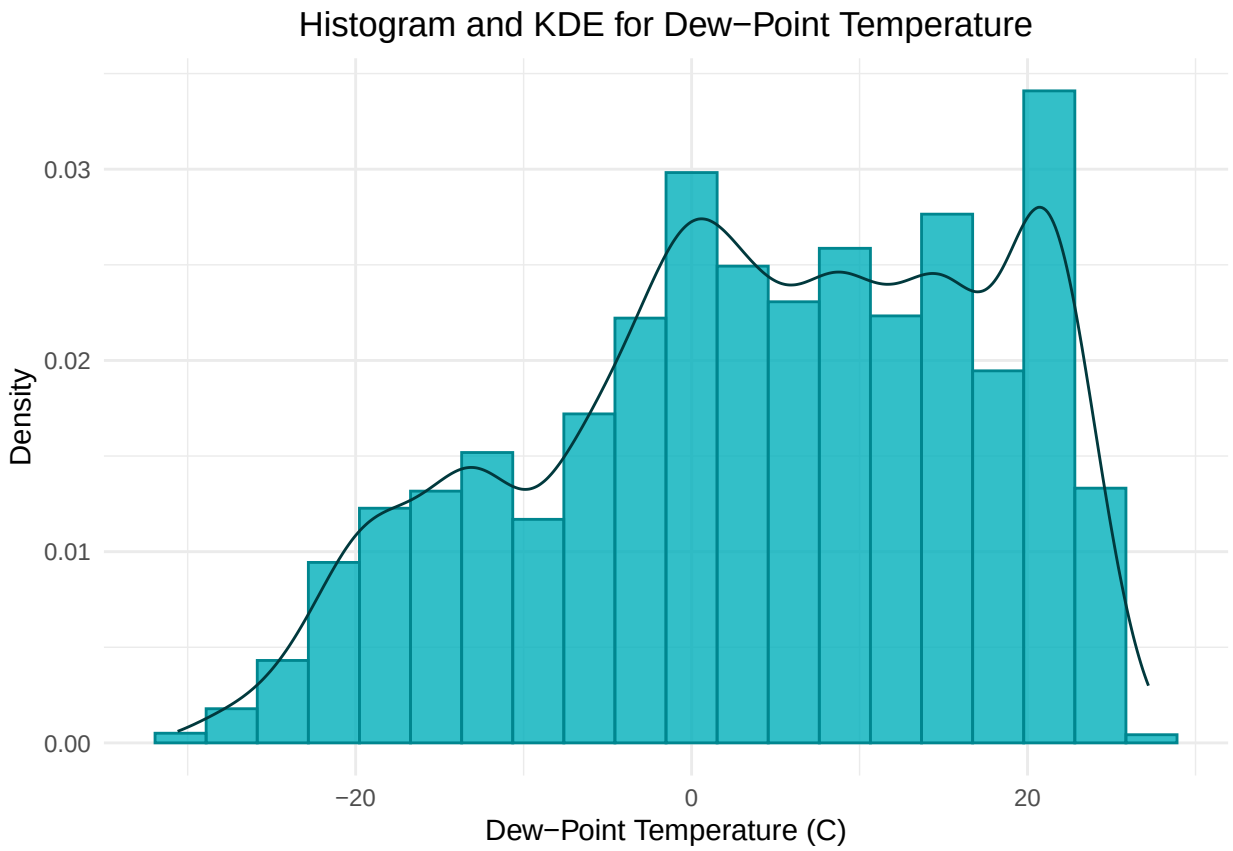
```r
summary(bike$dew_point_temperature)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

Again there are no implausible values to be found in the range of dew-point temperature.

Distribution of Dew-Point Temperature

```r
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$dp_temperature)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "Dew-Point Temperature (C)",
       y = "Density",
       title = "Histogram and KDE for Dew-Point Temperature") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# Skewness
skewness(bike$dp_temperature)
```

```
## [1] -0.3387147
```

The distribution seems to have a slight skew but it cannot reasonably be improved with a transformation, so the data stays as it is.

Humidity

Summary-Statistics of Humidity

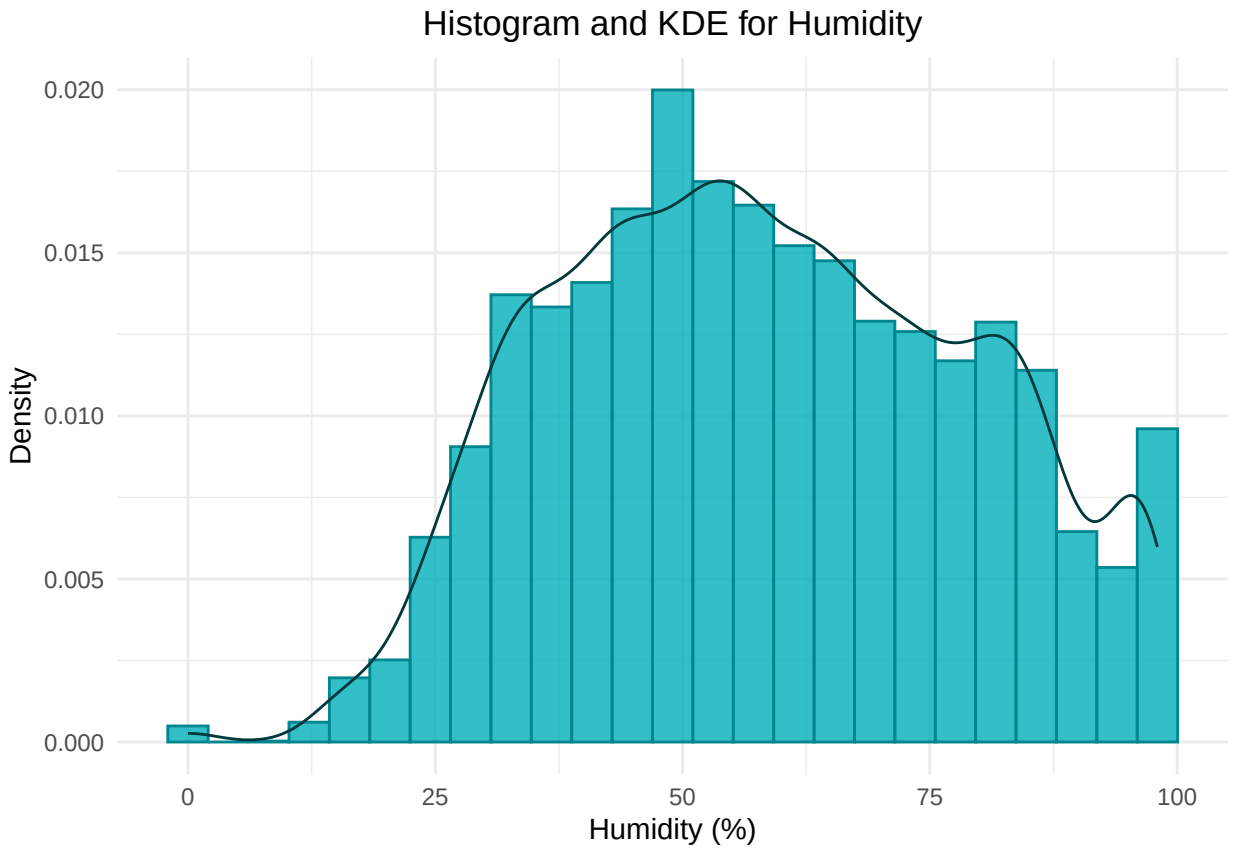
```
summary(bike$humidity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   42.00   57.00   58.15   74.00   98.00
```

The range of values for humidity is very plausible, going from 0% up to 98%.

Distribution of Humidity

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$humidity)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 25) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "Humidity (%)",
       y = "Density",
       title = "Histogram and KDE for Humidity") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Skewness
skewness(bike$humidity)
```

```
## [1] 0.06863681
```

The distribution for Humidity is not very skewed and is appropriate for usage in statistical modeling as it is and need no transformation.

Rainfall

Summary-Statistics of Rainfall

```
summary(bike$rainfall)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.1491  0.0000 35.0000
```

Both maximum and minimum of rainfall appear to be plausible, so there is no justification to remove any potential outliers based on the rainfall value.

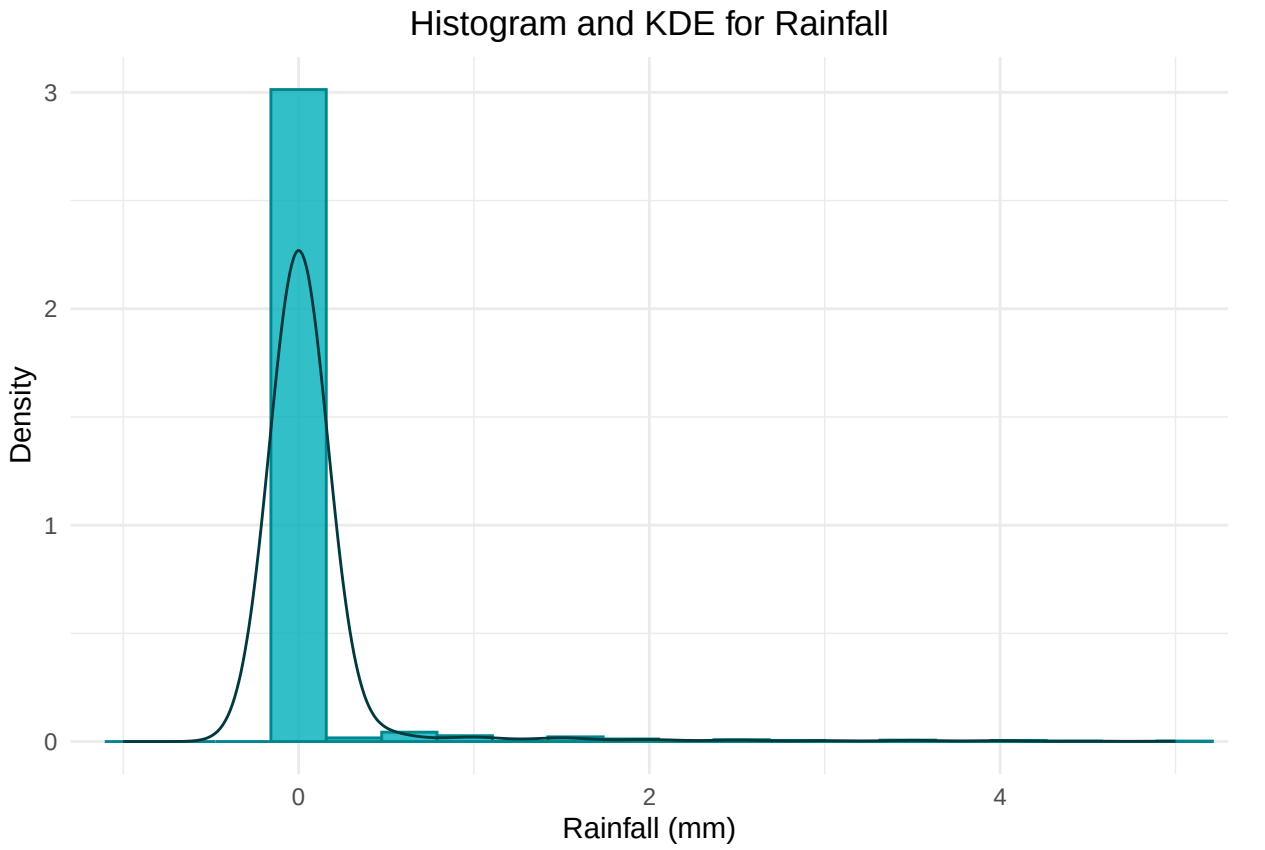
Distribution of Rainfall

```
# Skewness
skewness(bike$rainfall)
```

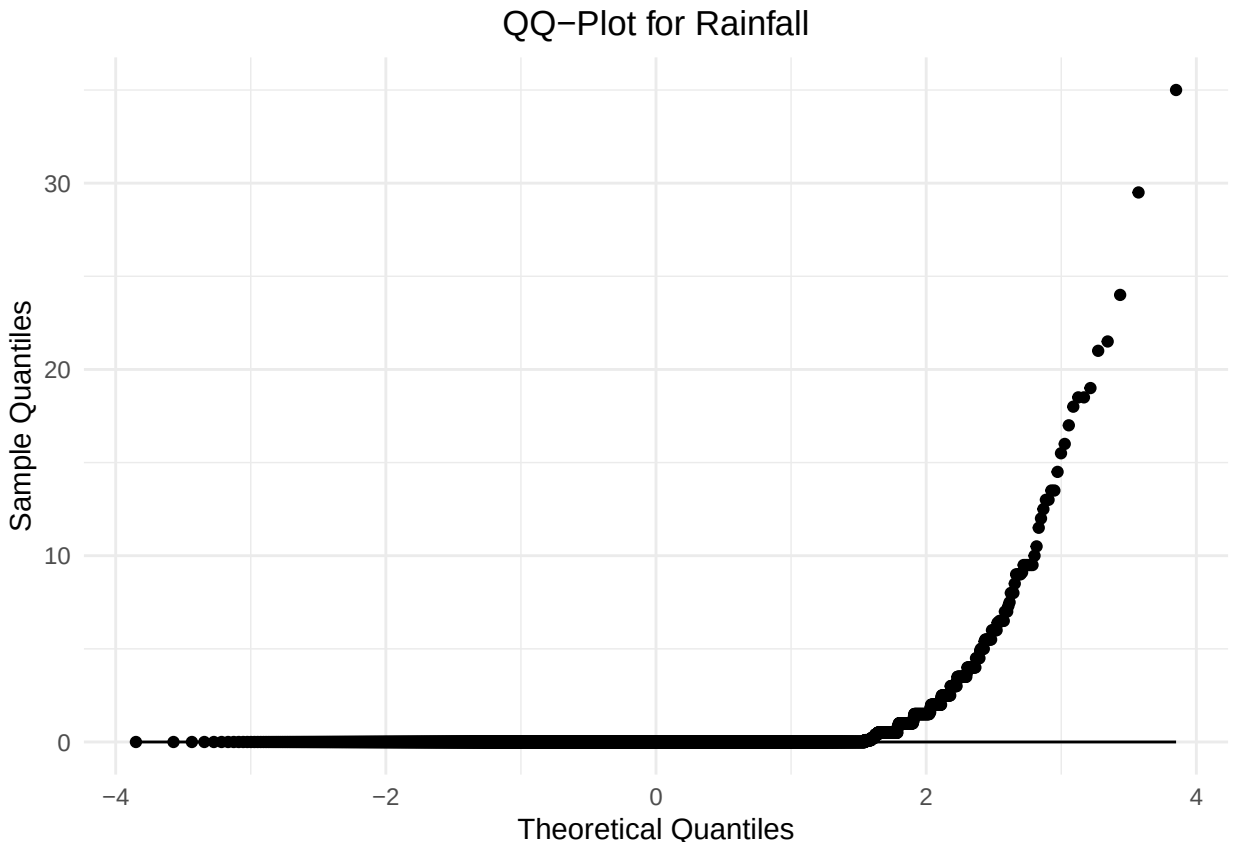
```
## [1] 14.61433
```

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$rainfall)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
```

```
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  scale_x_continuous(limits = c(-1, 5),
                     oob = scales::oob_keep) +
  theme_minimal() +
  labs(x = "Rainfall (mm)",
       y = "Density",
       title = "Histogram and KDE for Rainfall") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = rainfall)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for Rainfall") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

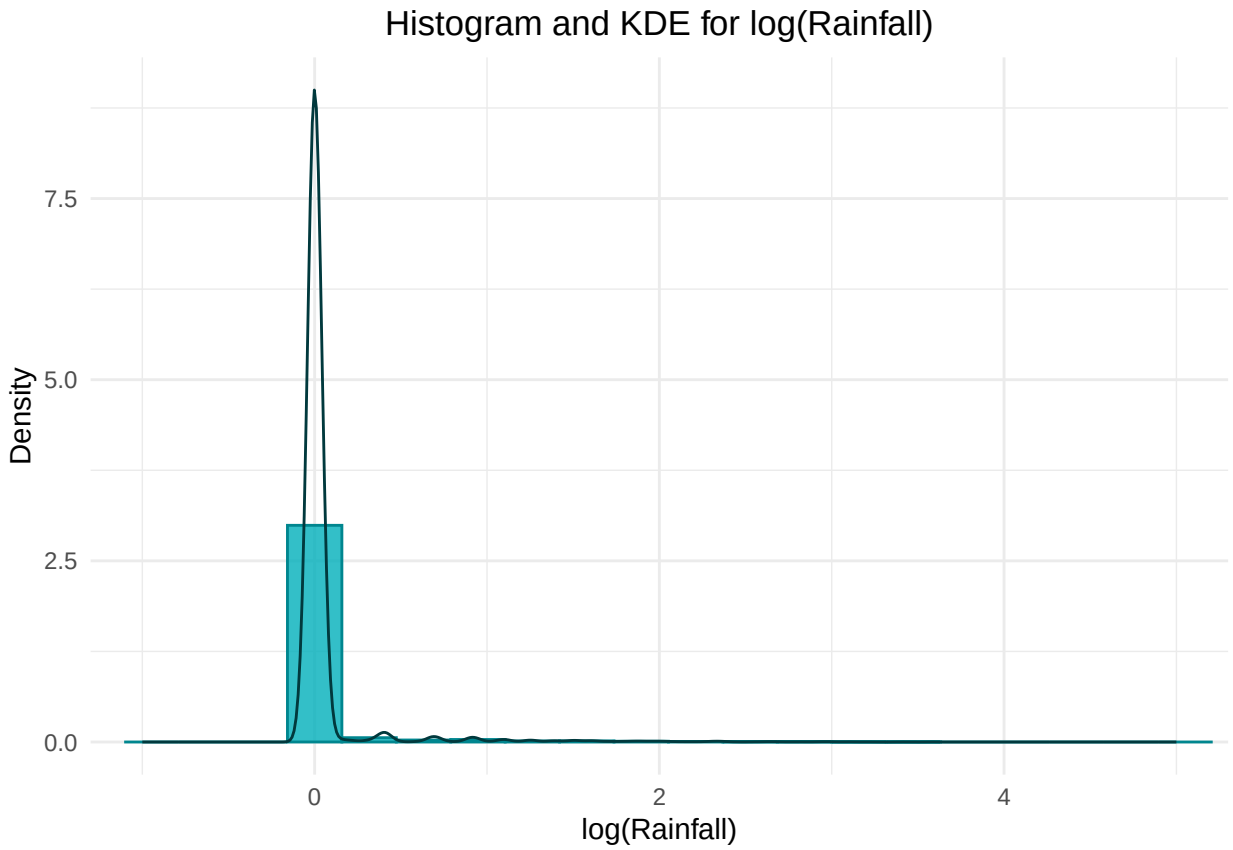


In the histogram for rainfall we can see that the distribution is extremely skewed, which is definitely problematic for the regression analysis. (Note that the x-axis is only plotted up to a value of 5 eventhough rainfall ranges up to a value of 35. The height of the bins for extreme values of rainfall is barely visible if plotted) To reduce the skewness to some extend we perform a log-transformation again. To account for the fact, that there are many observations, where rainfall $= 0$ we add a constant to the rainfall variable and calculate $log(\text{rainfall} + 1)$

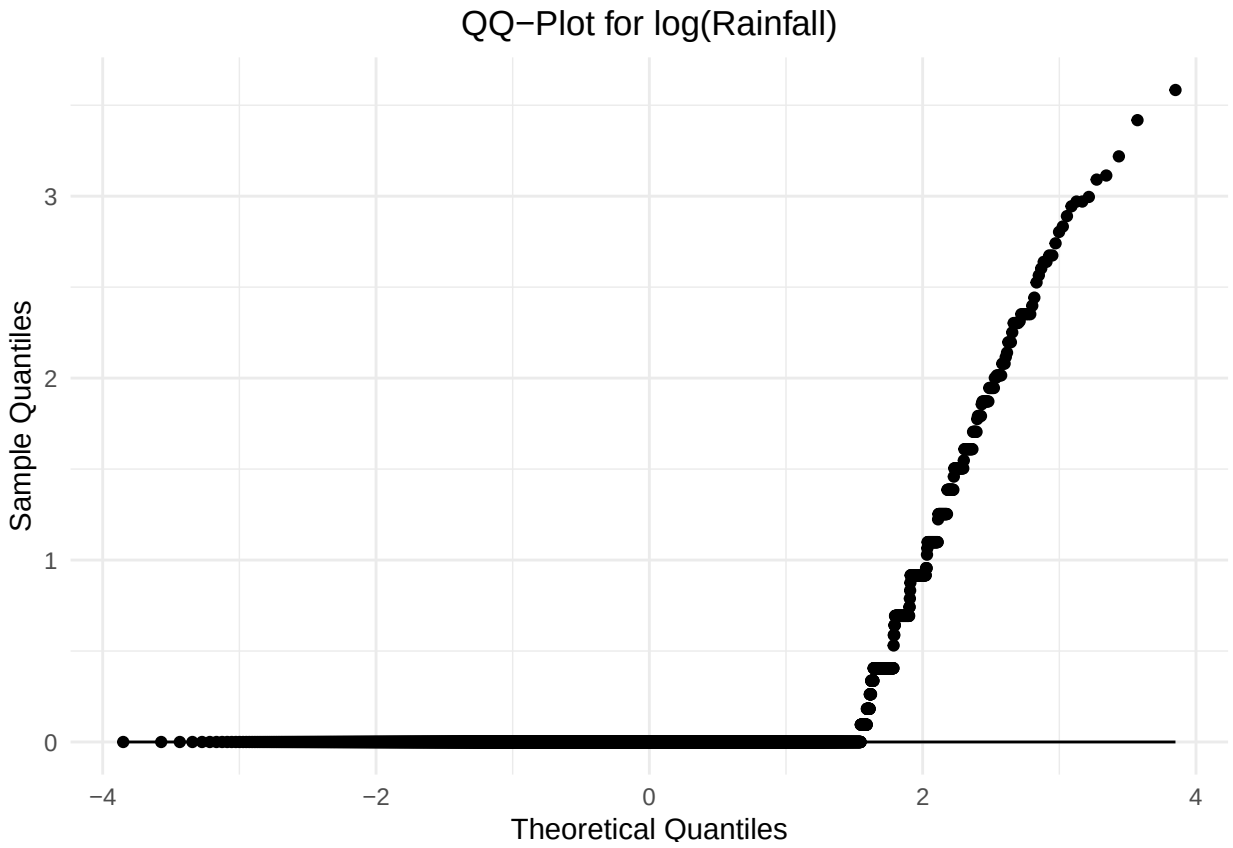
Log-transformation and new distribution

```
bike$log_rainfall <- log(bike$rainfall + 1)

# New distribution
ggplot(bike, mapping = aes(x = bike$log_rainfall)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  scale_x_continuous(limits = c(-1, 5), oob = scales::oob_keep) +
  theme_minimal() +
  labs(x = "log(Rainfall)",
       y = "Density",
       title = "Histogram and KDE for log(Rainfall)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = log_rainfall)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for log(Rainfall)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# New Skewness
skewness(bike$log_rainfall)
```

```
## [1] 6.416821
```

Even after the log transformation the data is heavily skewed, but at least the variance is now reduced. Different transformation techniques, like Box-Cox-Transformation, log-transformation with a higher base or inverse-transformation did not yield a better distribution. Using the regular log-transformation comes with the benefit of better interpretability for the linear regression.

Snowfall

Summary-Statistics of Snowfall

```
summary(bike$snowfall)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.07769 0.00000 8.80000
```

Both maximum and minimum of snowfall appear to be plausible, so there is no justification to remove any potential outliers based on the snowfall value.

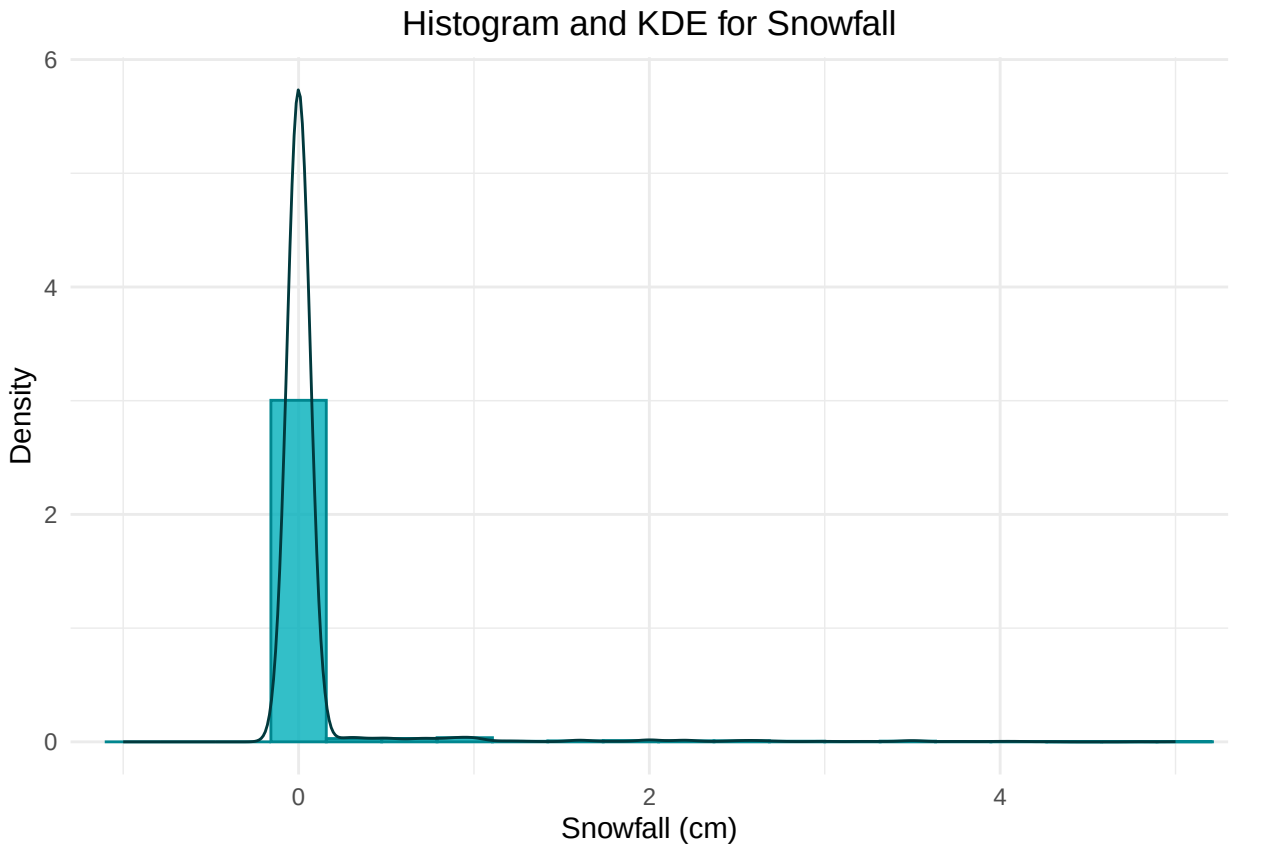
Distribution of Snowfall

```
# Skewness
skewness(bike$snowfall)
```

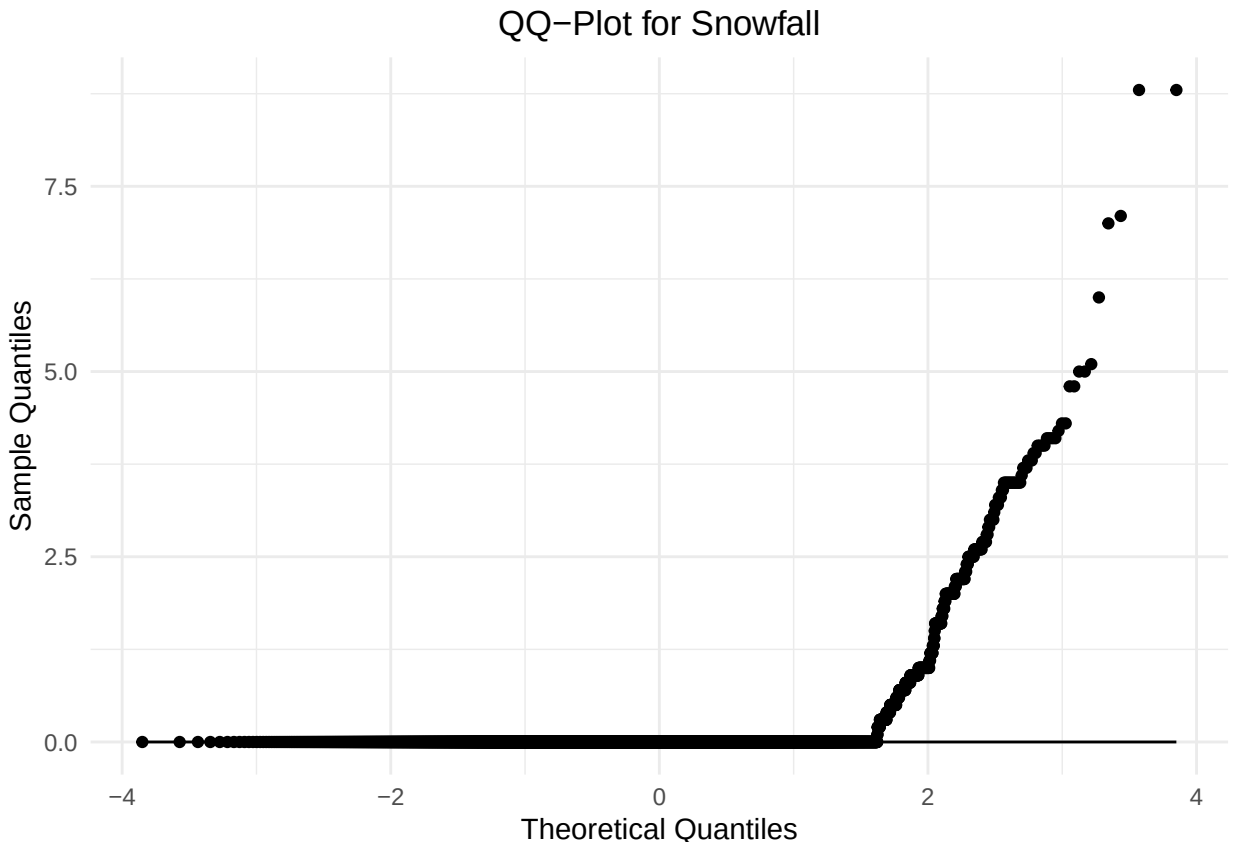
```
## [1] 8.291361
```

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$snowfall)) +
```

```
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  scale_x_continuous(limits = c(-1, 5),
                     oob = scales::oob_keep) +
  theme_minimal() +
  labs(x = "Snowfall (cm)",
       y = "Density",
       title = "Histogram and KDE for Snowfall") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = snowfall)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for Snowfall") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

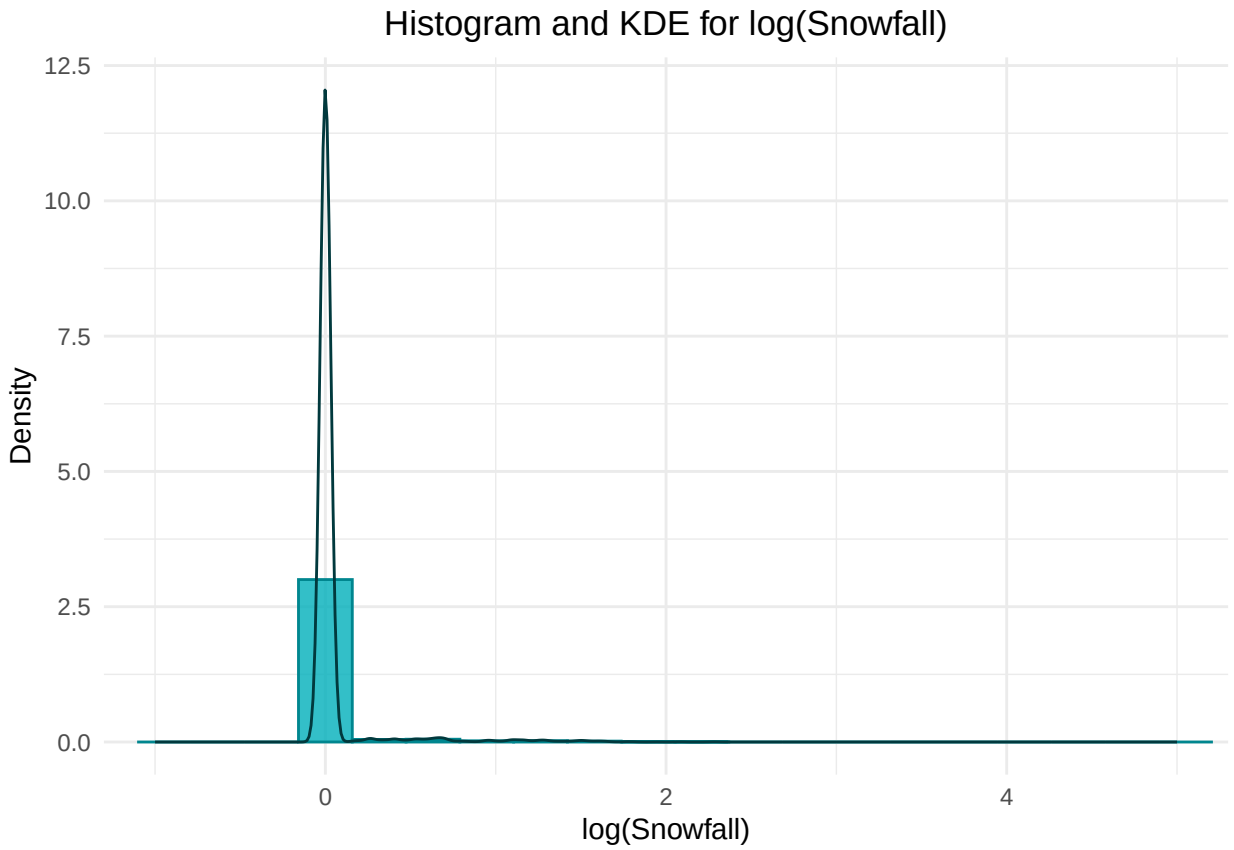


Same as for rainfall, also the snowfall distribution is heavily skewed to the right. (Note that again the x-axis has been limited to a maximum value of 5) We also log-transform the variable in the same way as with rainfall

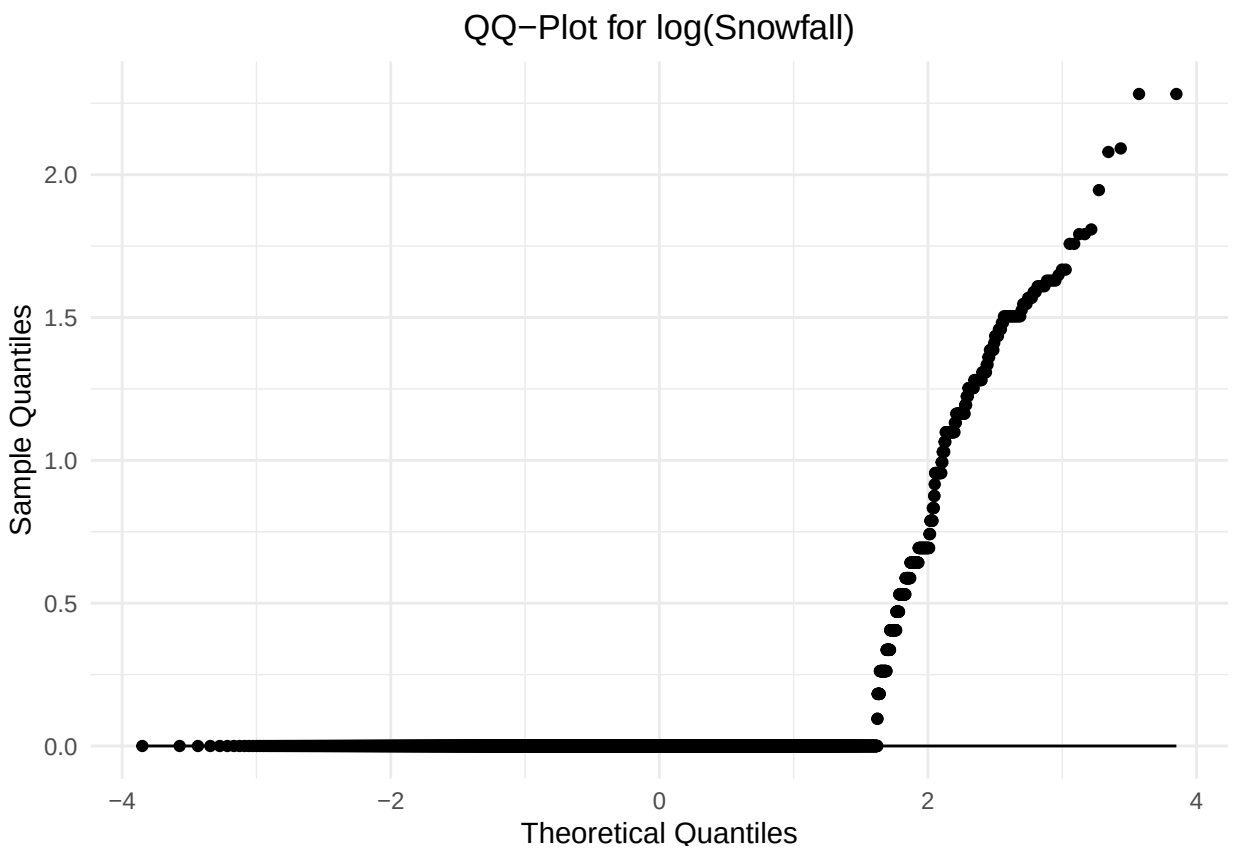
Log-transformation and new distribution

```
bike$log_snowfall <- log(bike$snowfall + 1)

# New distribution
ggplot(bike, mapping = aes(x = bike$log_snowfall)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  scale_x_continuous(limits = c(-1, 5), oob = scales::oob_keep) +
  theme_minimal() +
  labs(x = "log(Snowfall)",
       y = "Density",
       title = "Histogram and KDE for log(Snowfall)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = log_snowfall)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for log(Snowfall)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# New Skewness
skewness(bike$log_snowfall)
```

```
## [1] 5.70384
```

Same as with the rainfall, also the snowfall stays very skewed after the log-transformation, albeit with reduced variance.

Wind Speed

Summary-Statistics of Wind Speed

```
summary(bike$wind_speed)
```

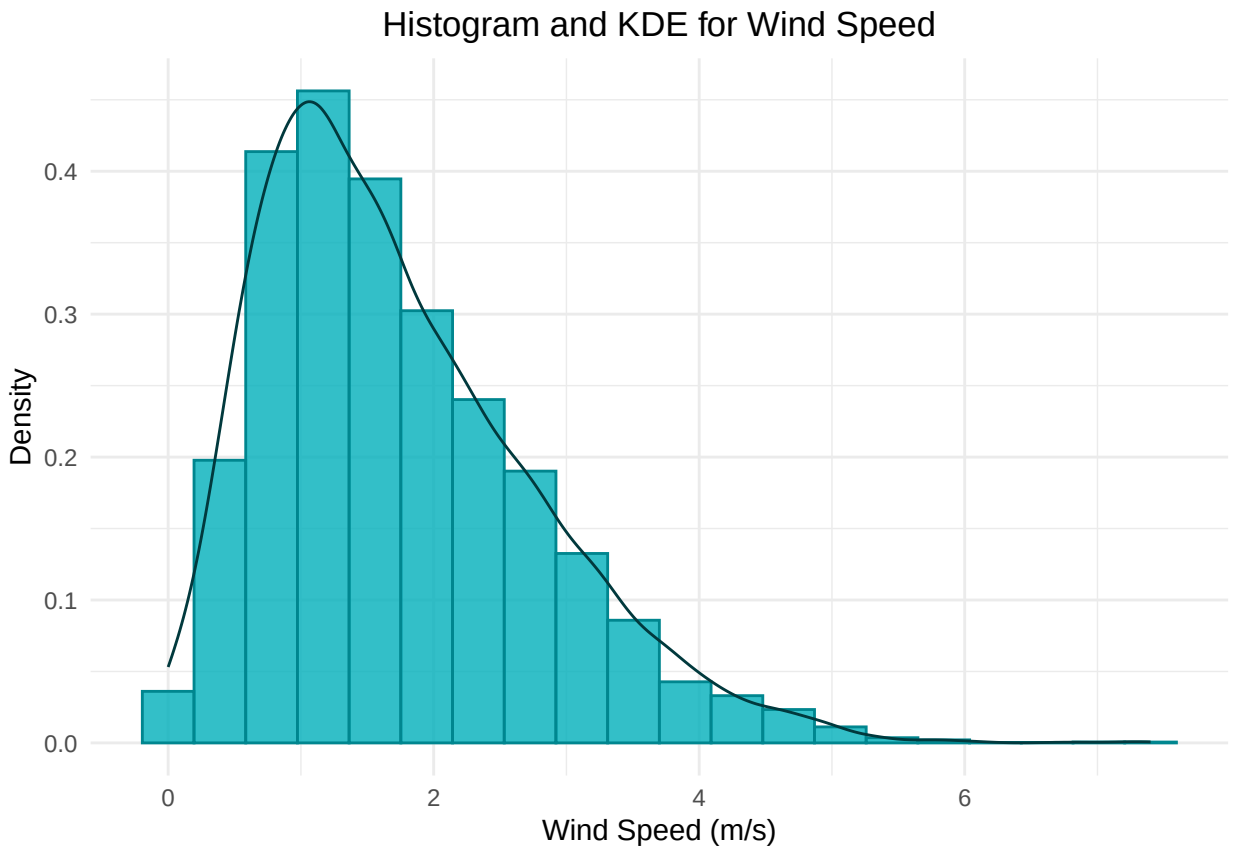
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.900   1.500   1.726   2.300   7.400
```

The range of values for wind speed is very plausible.

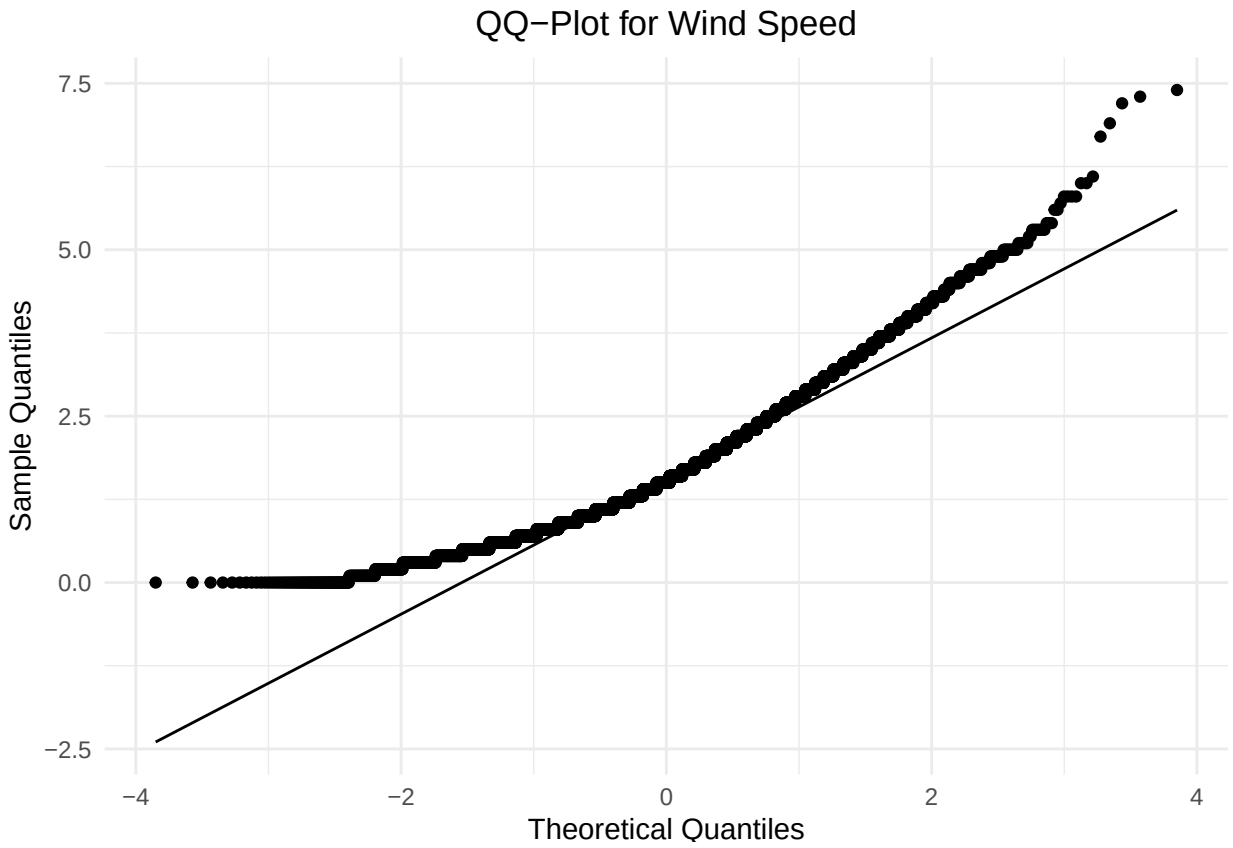
Distribution of Wind Speed

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = wind_speed)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
```

```
              color = "#01393d") +
  theme_minimal() +
  labs(x = "Wind Speed (m/s)",
       y = "Density",
       title = "Histogram and KDE for Wind Speed") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = wind_speed)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for Wind Speed") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Skewness
skewness(bike$wind_speed)
```

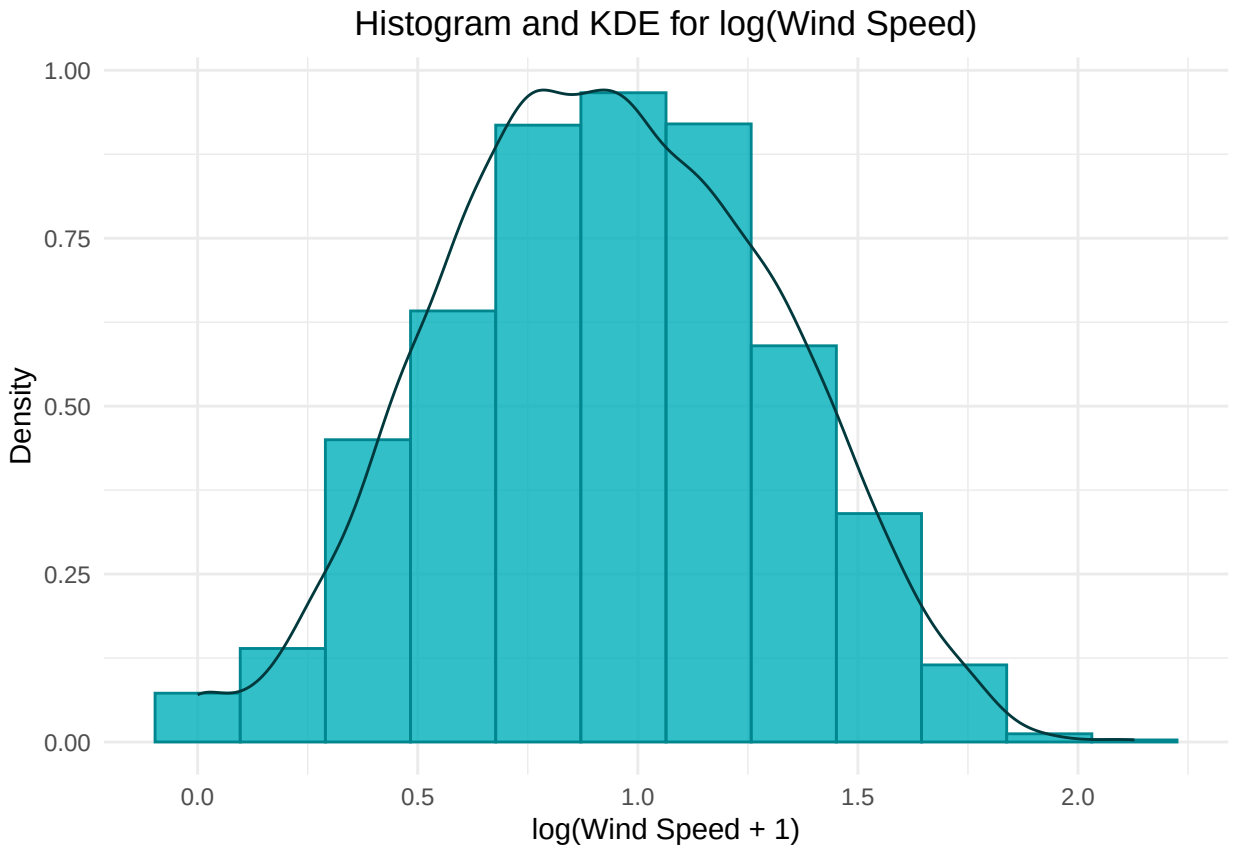
```
## [1] 0.894063
```

The distribution is slightly right-skewed and can be significantly improved with a log-transformation.

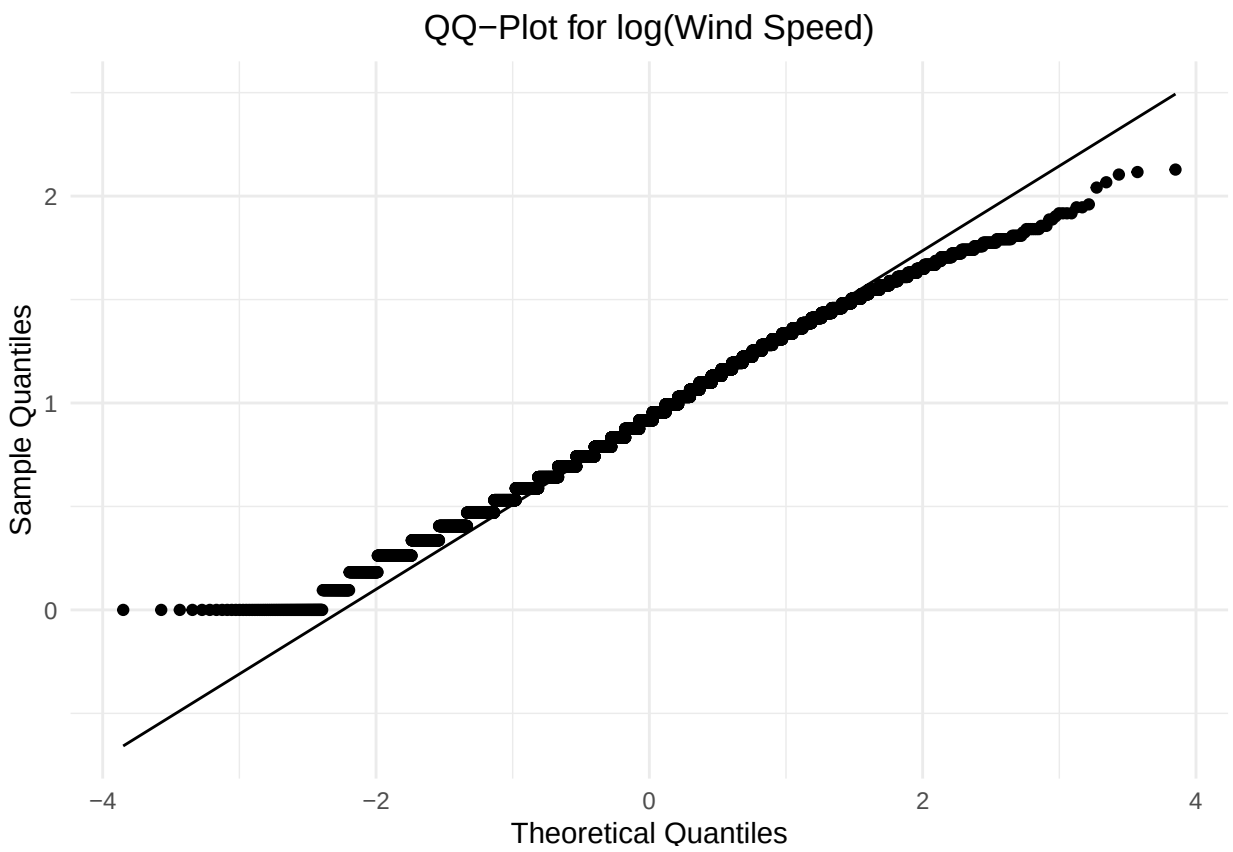
Log-transformation and new distribution

```
bike$log_wind_speed <- log(bike$wind_speed + 1)

# New distribution
ggplot(bike, mapping = aes(x = log_wind_speed)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 12) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "log(Wind Speed + 1)",
       y = "Density",
       title = "Histogram and KDE for log(Wind Speed)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = log_wind_speed)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for log(Wind Speed)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# New Skewness
skewness(bike$log_wind_speed)
```

```
## [1] 0.01373256
```

The distribution after the log-transformation looks approximately normal which is very desirable.

Visibility

Summary-Statistics of Visibility

```
summary(bike$visibility)
```

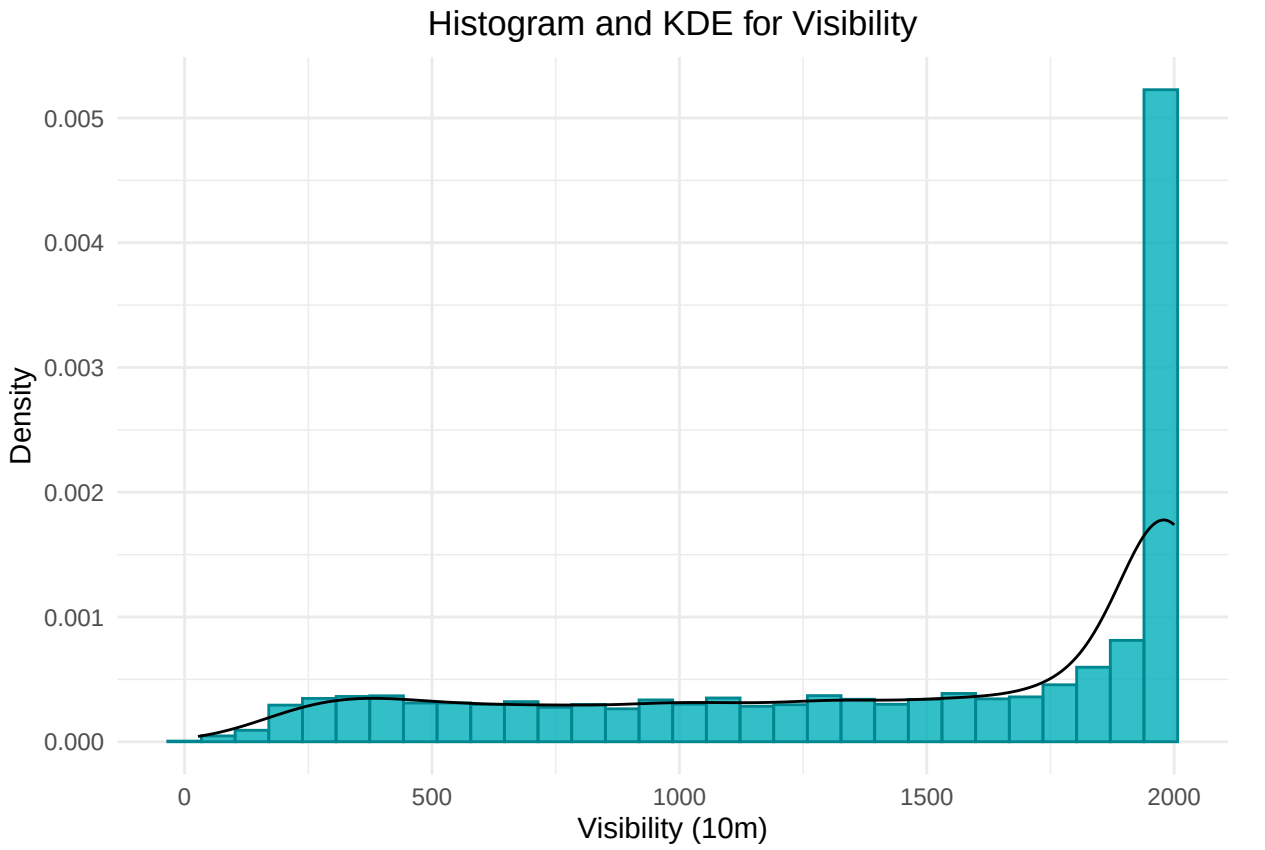
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      27     935    1690    1434    2000    2000
```

The minimum value for visibility seems to be fairly low, but it could be due to smog, which is not uncommon for asian metropolises.

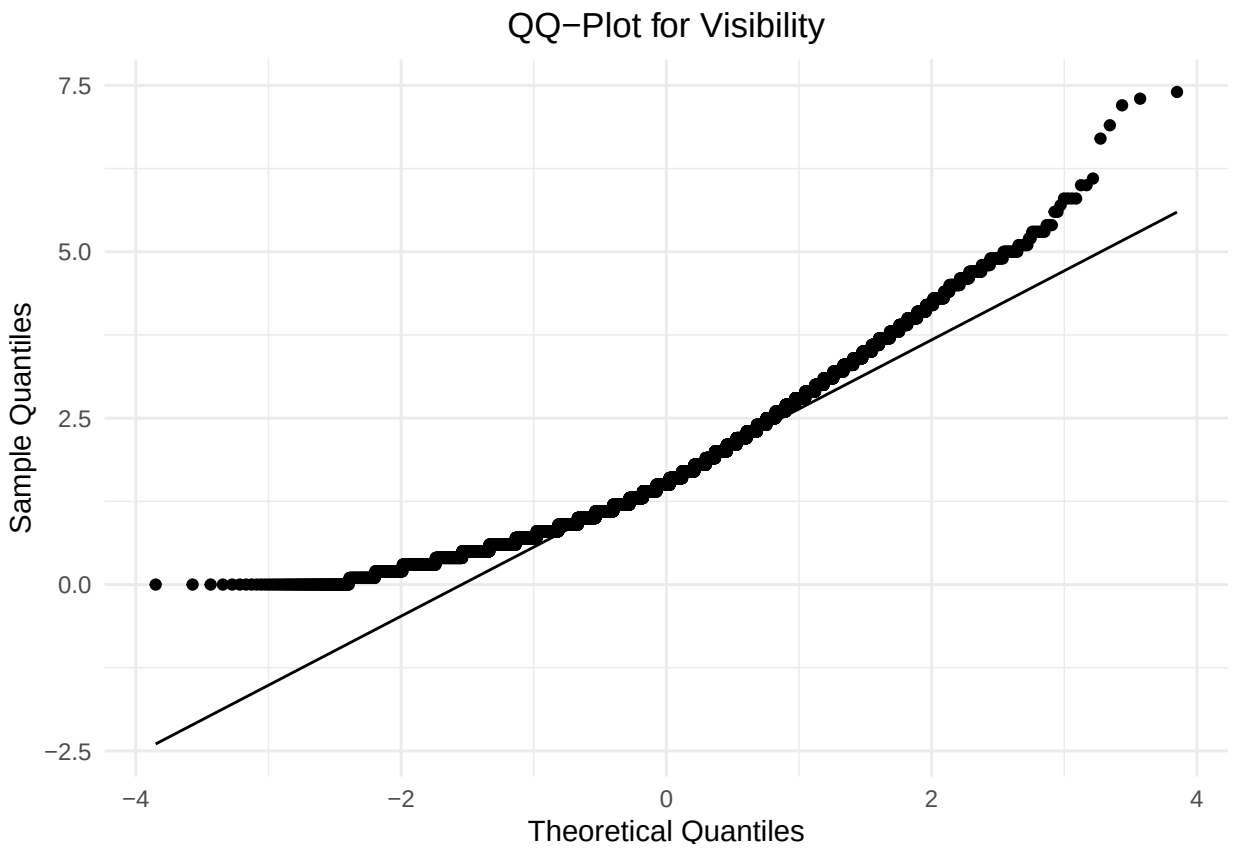
Distribution of Visibility

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$visibility)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8) +
  geom_density(kernel = "gaussian", bw = "nrd0") +
  theme_minimal() +
  labs(x = "Visibility (10m)",
```

```
       y = "Density",
       title = "Histogram and KDE for Visibility") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = wind_speed)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for Visibility") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Skewness
skewness(bike$visibility)
```

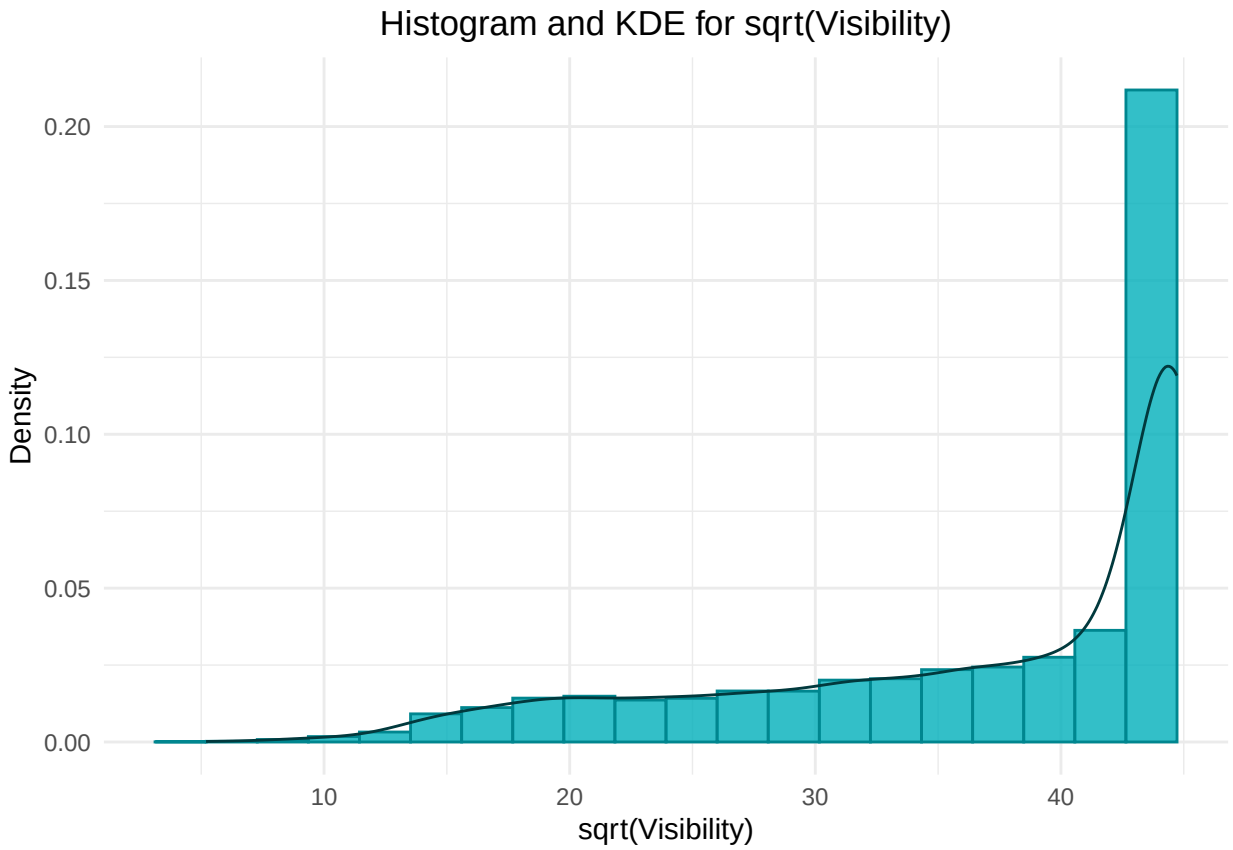
```
## [1] -0.695183
```

The data is extremely left-skewed. In this case it is reasonable to perform a square root-transformation.

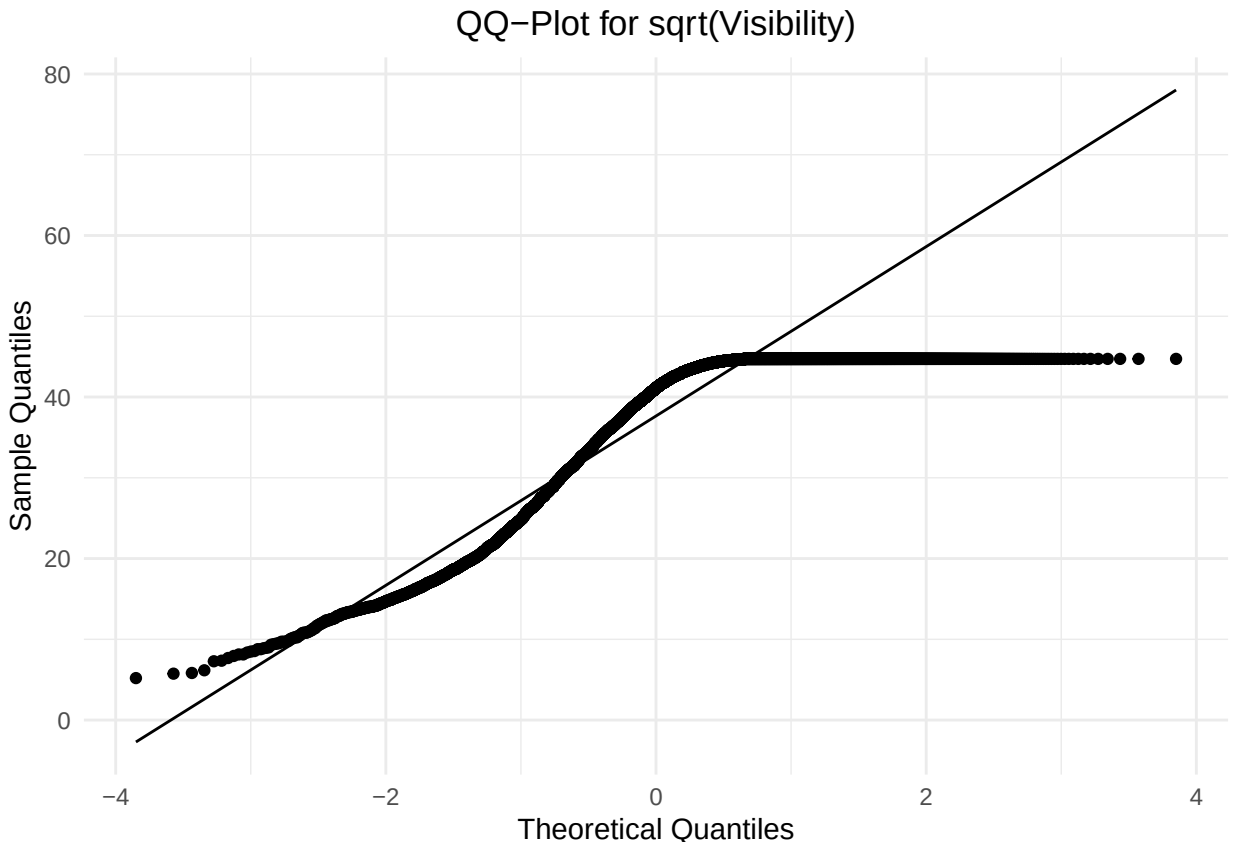
Square root-transformation and new distribution

```
bike$sqrt_visibility <- sqrt(bike$visibility)

# New distribution
ggplot(bike, mapping = aes(x = bike$sqrt_visibility)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8,
                 bins = 20) +
  geom_density(kernel = "gaussian",
               bw = "nrd0",
               color = "#01393d") +
  theme_minimal() +
  labs(x = "sqrt(Visibility)",
       y = "Density",
       title = "Histogram and KDE for sqrt(Visibility)") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# QQ-Plot
ggplot(data = bike, mapping = aes(sample = sqrt_visibility)) +
  stat_qq() +
  stat_qq_line() +
  labs(x = "Theoretical Quantiles",
       y = "Sample Quantiles",
       title = "QQ-Plot for sqrt(Visibility)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# New Skewness
skewness(bike$sqrt_visibility)
```

```
## [1] -1.009785
```

The data is still very left-skewed, but has a lower variance now.

Solar Radiation

Summary-Statistics of Solar Radiation

```
summary(bike$solar_radiation)
```

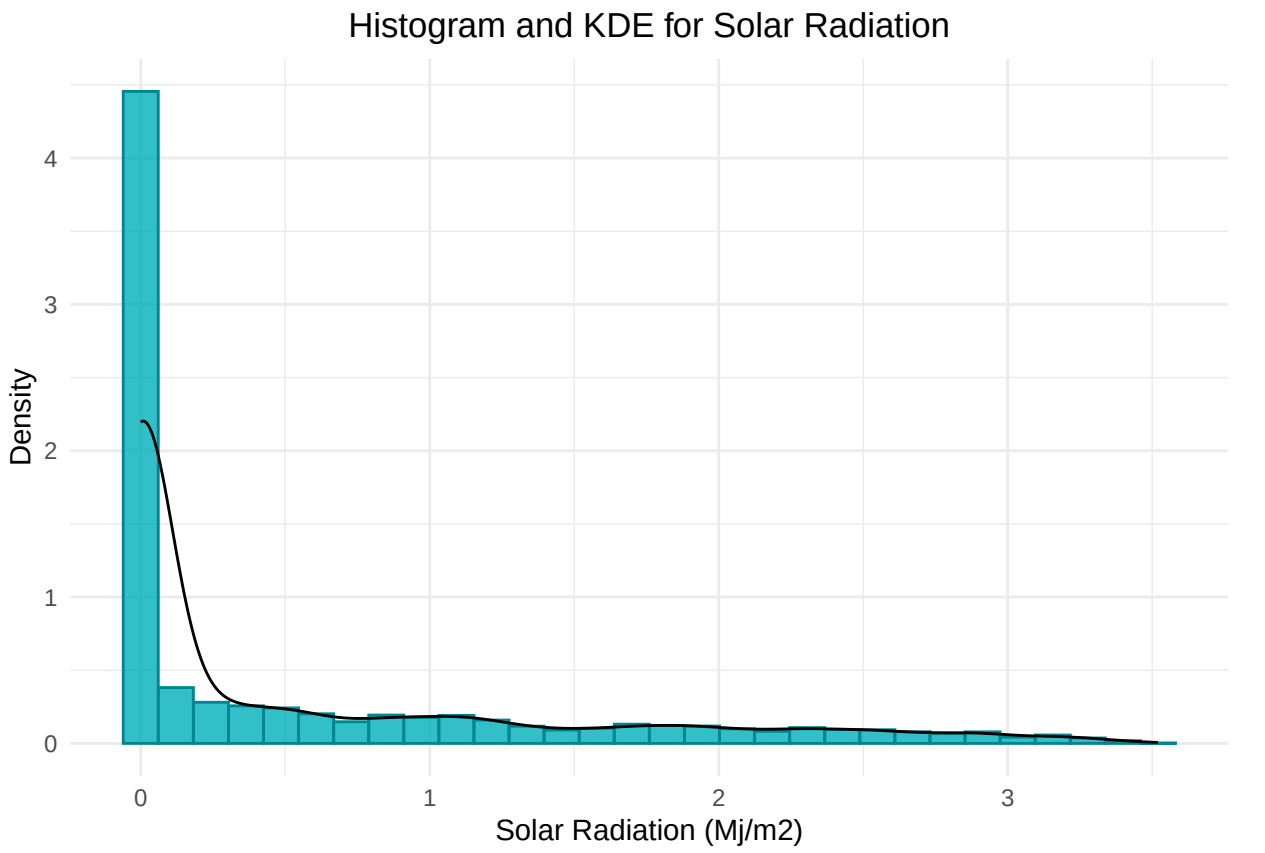
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0100  0.5679  0.9300  3.5200
```

After some brief research it seems like the range of values for solar radiation is plausible

Distribution of Solar Radiation

```
# Histogram and Kernel Density Estimation
ggplot(bike, mapping = aes(x = bike$solar_radiation)) +
  geom_histogram(mapping = aes(y = after_stat(density)),
                 fill = "#00AFBB",
                 color = "#00868f",
                 alpha = 0.8) +
  geom_density(kernel = "gaussian", bw = "nrd0") +
  theme_minimal() +
  labs(x = "Solar Radiation (Mj/m2)",
       y = "Density",
```

```
      title = "Histogram and KDE for Solar Radiation") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Skewness
skewness(bike$solar_radiation)
```

```
## [1] 1.509797
```

Again we have an extremely skewed distribution but performing a log-transformation does almost not change the distribution at all, so we just keep this variable as it is.

Holiday

Values for Holiday

```
levels(bike$holiday)
```

```
## [1] "Holiday"    "No Holiday"
```

The holiday variable is binary as expected.

The only transformation necessary is to perform one-hot encoding to enable usage for different algorithms.

```
bike$holiday <- ifelse(bike$holiday == "Holiday", 1, 0)
```

Seasons

Values for Seasons

```
levels(bike$seasons)
```

```
## [1] "Autumn" "Spring" "Summer" "Winter"
```

There are no implausible values for the seasons variable.

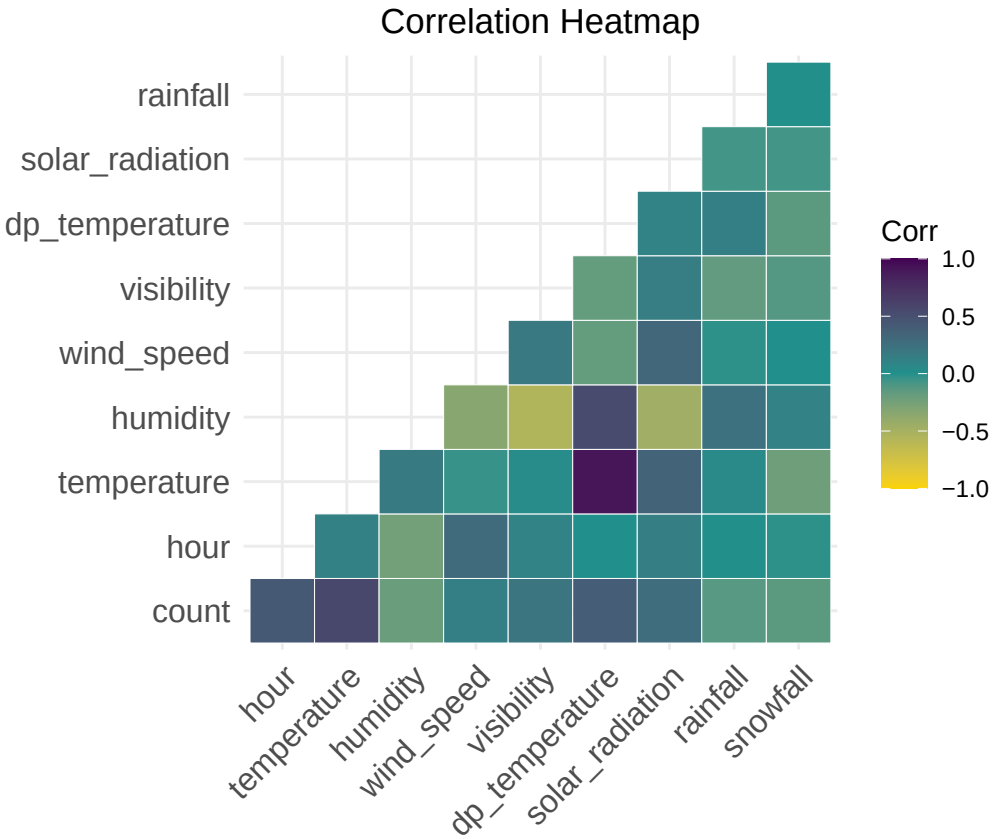
We also perform one-hot encoding for each of the different seasons.

```
bike$winter <- ifelse(bike$seasons == "Winter", 1, 0)
bike$spring <- ifelse(bike$seasons == "Spring", 1, 0)
bike$summer <- ifelse(bike$seasons == "Summer", 1, 0)
bike$autumn <- ifelse(bike$seasons == "Autumn", 1, 0)
```

Correlation Analysis

```
corr <- cor(bike[, c("count", "hour", "temperature", "humidity", "wind_speed",
                     "visibility", "dp_temperature", "solar_radiation",
                     "rainfall", "snowfall")])

ggcorrplot(corr, outline.col = "white",
           type = "lower",
           colors = c("#fcd303",  "#21908CFF", "#440154FF")) +
  labs(title = "Correlation Heatmap") +
  theme(plot.title = element_text(hjust = 0.5))
```



The temperature, hour of the day and the dew point temperature seem to have the strongest relationship with the number of bikes rented. With a correlation coefficient of 0.91 there also appears to be a strong connection between the temperature and the dew point temperature. The latter also shows a relatively strong correlation with humidity. In fact, the dew point temperature can be almost perfectly approximated using the so called Magnus formula.

Based on this knowledge it would be reasonable to exclude dew point temperature at this point already from any further analysis but we intentionally keep it in the data set to observe the effects of this relationship on the models which will be introduced later on in the analysis.