Linear_regression

Regression

Get data

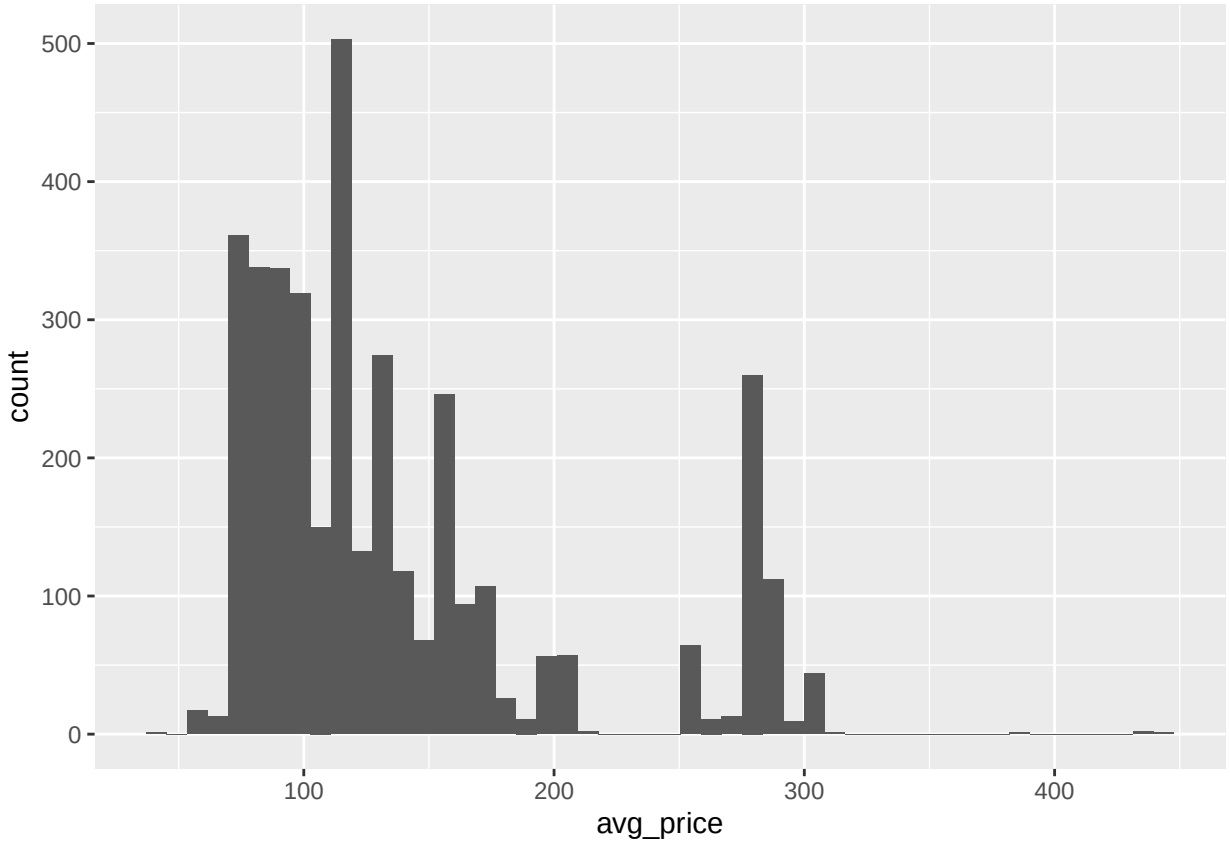
here we load the regression dataset and show the first five lines

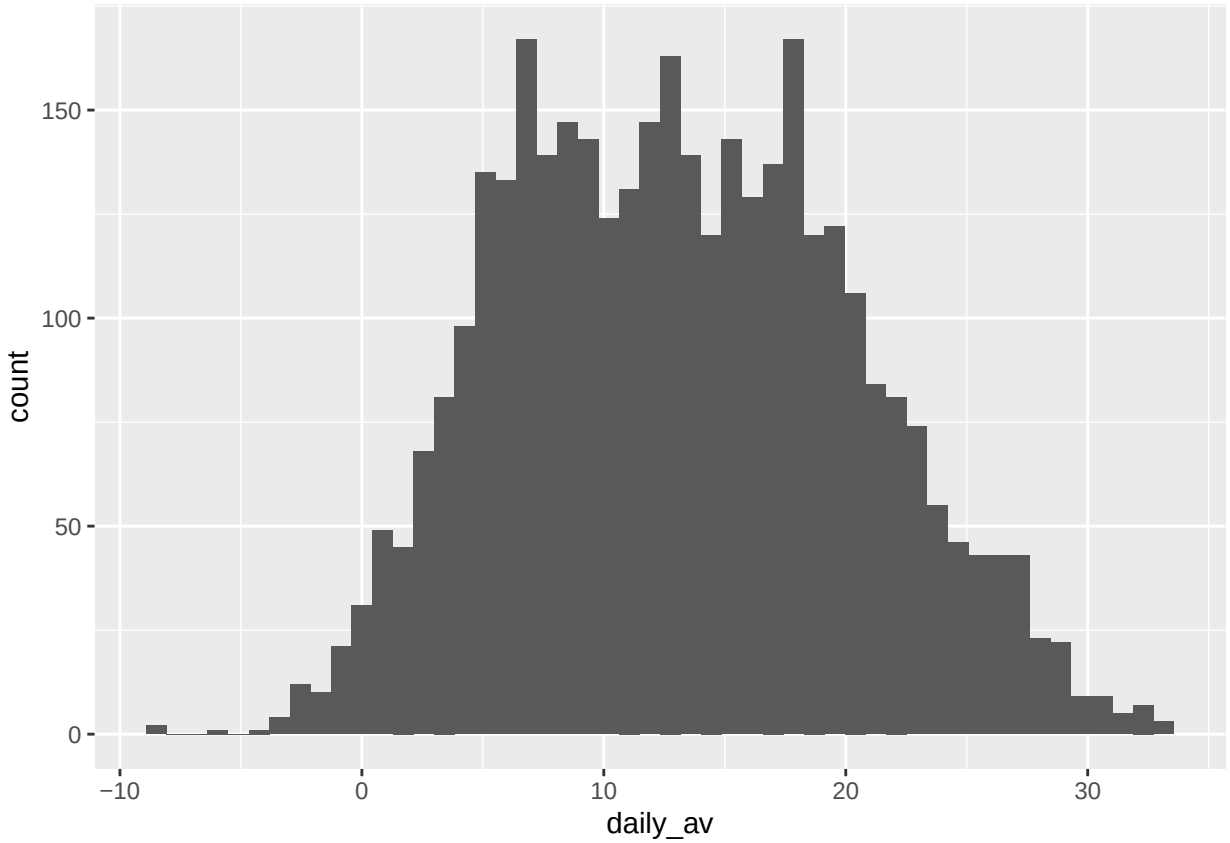
```
regression <- read.csv("../../gen/temp/data_df.csv")
head(regression)
```

```
##         date avg_price      city daily_av is_holiday
## 1 2021-03-05  158.9904 amsterdam     2.96          0
## 2 2021-03-06  156.6292 amsterdam     1.69          0
## 3 2021-03-07  152.7952 amsterdam     2.69          0
## 4 2021-03-08  152.7235 amsterdam     4.87          0
## 5 2021-03-09  152.9122 amsterdam     5.58          0
## 6 2021-03-10  153.1850 amsterdam     5.75          0
```

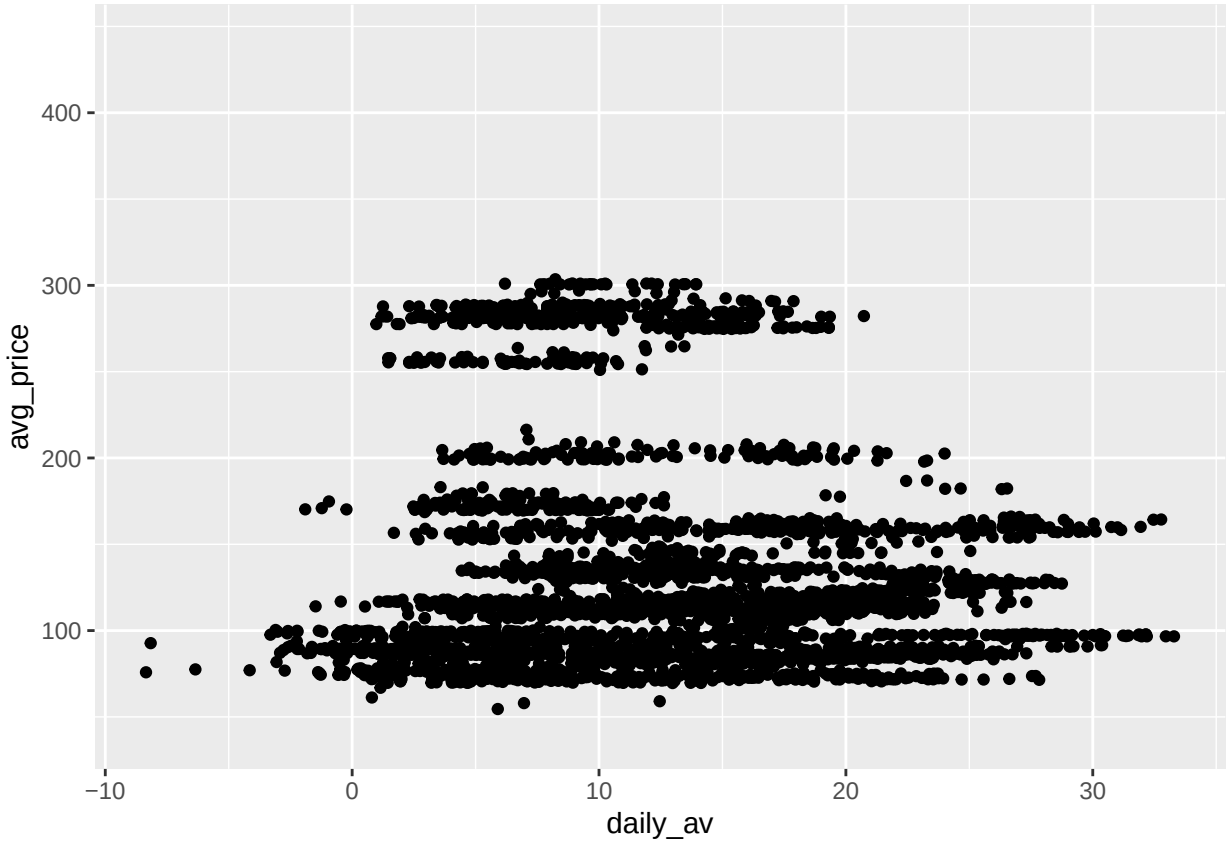
Check for normality

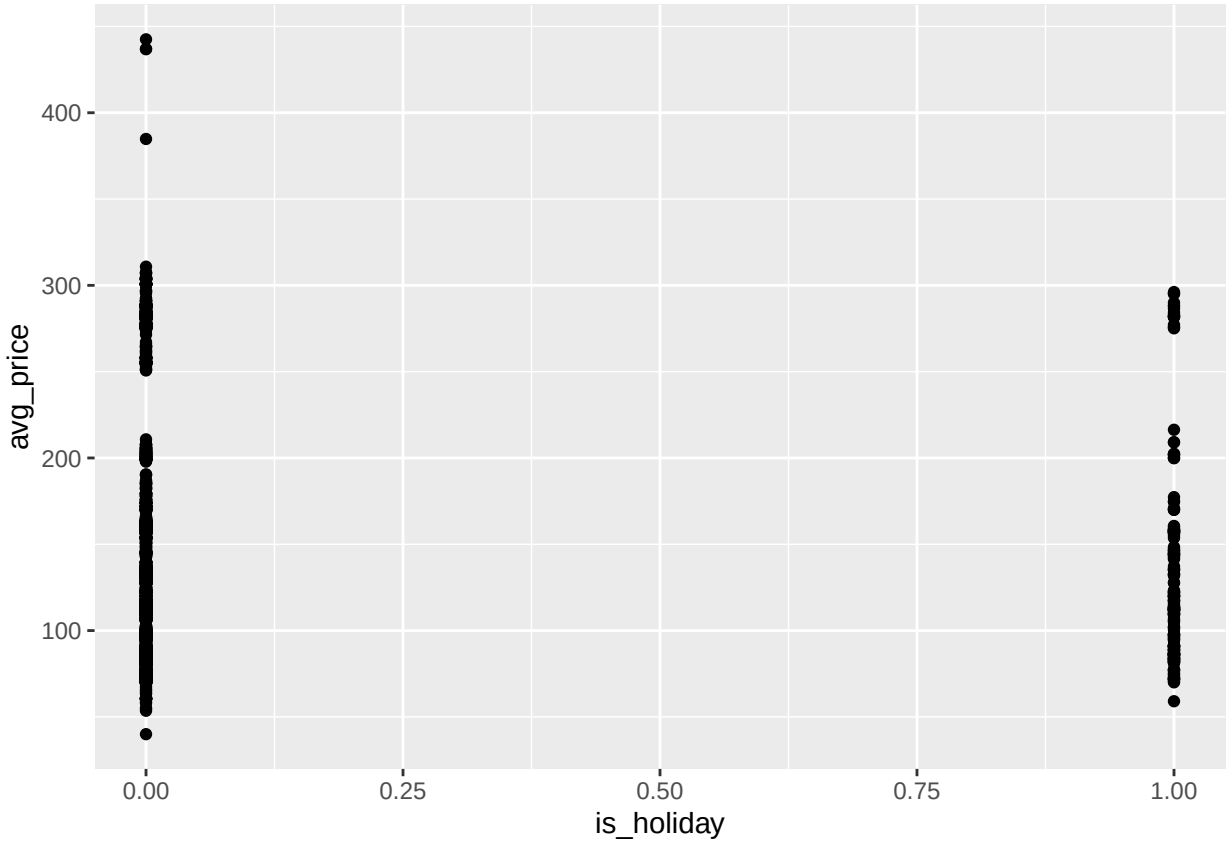
Here we check whether the data is normally distributed.


500
400
300
200
100
0
count
100
200
300
400
avg_price



Check for linearity Here we check if the data is linear.


400
300
200
100
avg_price
−10
0
10
20
30
daily_av



Linear Regresssion The final output obtained via regression:

```
##
## Call:
## lm(formula = avg_price ~ daily_av + as.factor(is_holiday), data = regression)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -92.29 -46.82 -18.15  24.00 166.43
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            153.9170     2.2400  68.712  < 2e-16 ***
## daily_av                -1.1957     0.1479  -8.086 8.35e-16 ***
## as.factor(is_holiday)1  -8.6441     4.5228  -1.911   0.0561 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.39 on 3579 degrees of freedom
##   (166 observations deleted due to missingness)
## Multiple R-squared:  0.01861,    Adjusted R-squared:  0.01806
## F-statistic: 33.93 on 2 and 3579 DF,  p-value: 2.537e-15

##              daily_av as.factor(is_holiday)
##              1.001796              1.001796
```