SeoulBikeData program

```
SeoulBikeData = read.csv('C:/Users/kevin/Desktop/SeoulBikeData.csv',hea
der = TRUE)
head(SeoulBikeData)

##         Date Rented.Bike.Count Hour Temperature.癈. Humidity...
## 1 01/12/2017               254    0            -5.2          37
## 2 01/12/2017               204    1            -5.5          38
## 3 01/12/2017               173    2            -6.0          39
## 4 01/12/2017               107    3            -6.2          40
## 5 01/12/2017                78    4            -6.0          36
## 6 01/12/2017               100    5            -6.4          37
##   Wind.speed..m.s. Visibility..10m. Dew.point.temperature.癈.
## 1              2.2             2000                     -17.6
## 2              0.8             2000                     -17.6
## 3              1.0             2000                     -17.7
## 4              0.9             2000                     -17.6
## 5              2.3             2000                     -18.6
## 6              1.5             2000                     -18.7
##   Solar.Radiation..MJ.m2. Rainfall.mm. Snowfall..cm. Seasons    Holi
day
## 1                       0            0             0  Winter No Holi
day
## 2                       0            0             0  Winter No Holi
day
## 3                       0            0             0  Winter No Holi
day
## 4                       0            0             0  Winter No Holi
day
## 5                       0            0             0  Winter No Holi
day
## 6                       0            0             0  Winter No Holi
day
##   Functioning.Day
## 1             Yes
## 2             Yes
## 3             Yes
## 4             Yes
## 5             Yes
## 6             Yes

sum(is.na(SeoulBikeData))

## [1] 0

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyve
rse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_co
nflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
SeoulBikeData <- SeoulBikeData %>% rename(
    rental_count = Rented.Bike.Count,
    Temperature = Temperature.癤.,
    Humidity = Humidity...,
    Wind_speed = Wind.speed..m.s.,
    Visibility_in_10m = Visibility..10m.,
    Dew_point_temperature = Dew.point.temperature.癤.,
    Solar_Radiation = Solar.Radiation..MJ.m2.,
    Rainfall_in_mm = Rainfall.mm.,
    Snowfall_in_cm = Snowfall..cm.,
    Functioning_Day = Functioning.Day
    )
head(SeoulBikeData)
```

```
##         Date rental_count Hour Temperature Humidity Wind_speed
## 1 01/12/2017          254    0        -5.2       37        2.2
## 2 01/12/2017          204    1        -5.5       38        0.8
## 3 01/12/2017          173    2        -6.0       39        1.0
## 4 01/12/2017          107    3        -6.2       40        0.9
## 5 01/12/2017           78    4        -6.0       36        2.3
## 6 01/12/2017          100    5        -6.4       37        1.5
##   Visibility_in_10m Dew_point_temperature Solar_Radiation Rainfall_i
n_mm
## 1              2000                -17.6              0
   0
## 2              2000                -17.6              0
   0
## 3              2000                -17.7              0
   0
## 4              2000                -17.6              0
   0
## 5              2000                -18.6              0
   0
## 6              2000                -18.7              0
   0
##   Snowfall_in_cm Seasons    Holiday Functioning_Day
## 1              0  Winter No Holiday             Yes
```

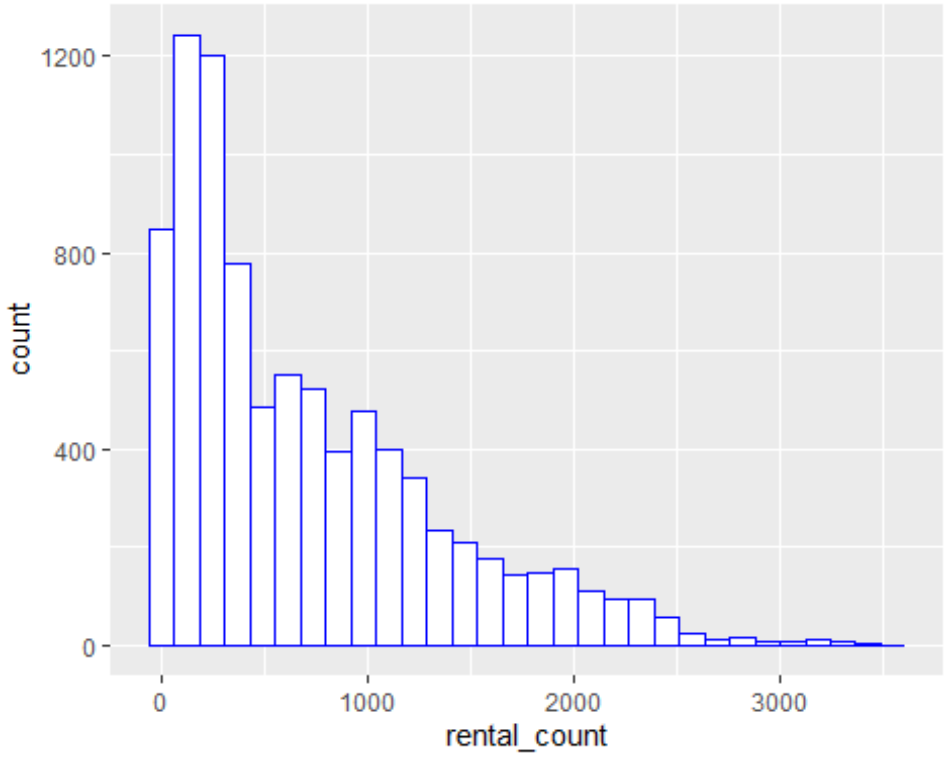
```
## 2              0  Winter No Holiday             Yes
## 3              0  Winter No Holiday             Yes
## 4              0  Winter No Holiday             Yes
## 5              0  Winter No Holiday             Yes
## 6              0  Winter No Holiday             Yes
```

```
str(SeoulBikeData)
```

```
## 'data.frame':    8760 obs. of  14 variables:
##  $ Date                 : chr  "01/12/2017" "01/12/2017" "01/12/2017
" "01/12/2017" ...
##  $ rental_count         : int  254 204 173 107 78 100 181 460 930 49
0 ...
##  $ Hour                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temperature          : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -
7.6 -6.5 ...
##  $ Humidity             : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind_speed           : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5
 ...
##  $ Visibility_in_10m    : int  2000 2000 2000 2000 2000 2000 2000 20
00 2000 1928 ...
##  $ Dew_point_temperature: num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -
19.5 -19.3 -19.8 -22.4 ...
##  $ Solar_Radiation      : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rainfall_in_mm       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snowfall_in_cm       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Seasons              : chr  "Winter" "Winter" "Winter" "Winter"
 ...
##  $ Holiday              : chr  "No Holiday" "No Holiday" "No Holiday
" "No Holiday" ...
##  $ Functioning_Day      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

```
ggplot(SeoulBikeData, aes(x=rental_count)) + geom_histogram(color="blue
", fill="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
nrow(subset(SeoulBikeData, rental_count==0))
```

```
## [1] 295
```

```
SeoulBikeData %>% count(Functioning_Day)
```

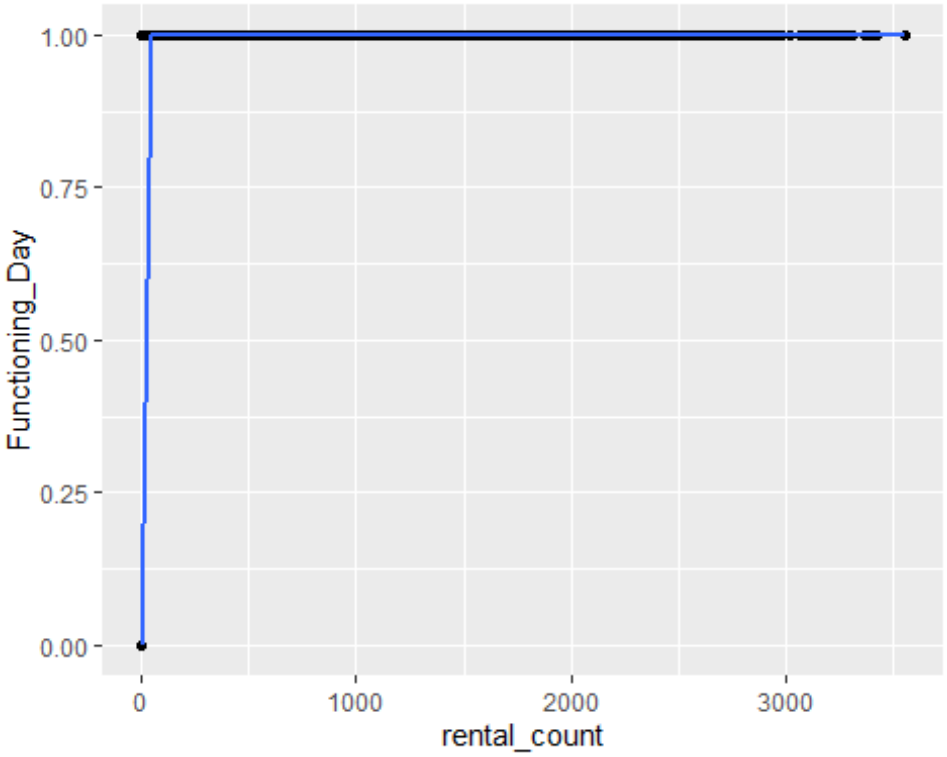
```
##   Functioning_Day    n
## 1              No  295
## 2             Yes 8465
```

```
SeoulBikeData$Functioning_Day[SeoulBikeData$Functioning_Day == 'No'] <-
 0
SeoulBikeData$Functioning_Day[SeoulBikeData$Functioning_Day == 'Yes'] <
- 1
SeoulBikeData$Functioning_Day <- as.numeric(SeoulBikeData$Functioning_D
ay)
ggplot(SeoulBikeData, aes(x=rental_count, y=Functioning_Day)) +
  geom_point() +
  geom_smooth(method = "glm",
    method.args = list(family = "binomial"),
    se = FALSE)
```

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

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

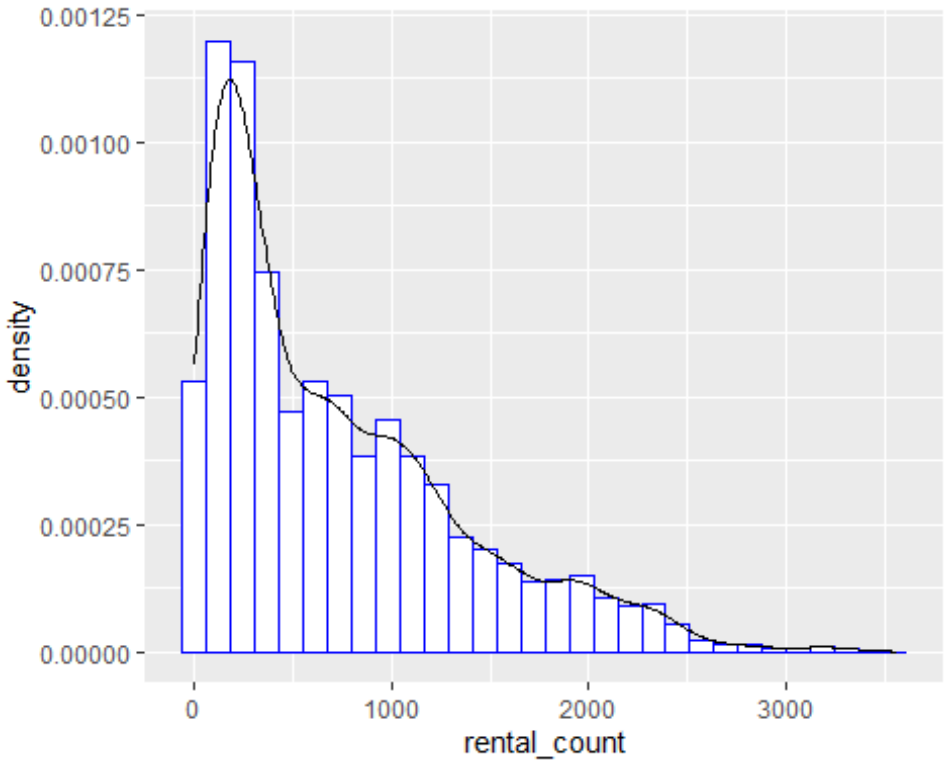



```
SeoulBikeData_nonFunction <- filter(SeoulBikeData, Functioning_Day == 0)
SeoulBikeData_nonFunction %>% count(rental_count)

##   rental_count   n
## 1            0 295

SeoulBikeData_nonFunction %>% count(Functioning_Day)

##   Functioning_Day   n
## 1               0 295

SeoulBikeData_Function <- filter(SeoulBikeData, Functioning_Day == 1)
str(SeoulBikeData_Function)

## 'data.frame':    8465 obs. of  14 variables:
##  $ Date                 : chr  "01/12/2017" "01/12/2017" "01/12/2017
" "01/12/2017" ...
##  $ rental_count         : int  254 204 173 107 78 100 181 460 930 49
0 ...
##  $ Hour                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temperature          : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -
7.6 -6.5 ...
##  $ Humidity             : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind_speed           : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5
 ...
##  $ Visibility_in_10m    : int  2000 2000 2000 2000 2000 2000 2000 20
00 2000 1928 ...
##  $ Dew_point_temperature: num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -
```

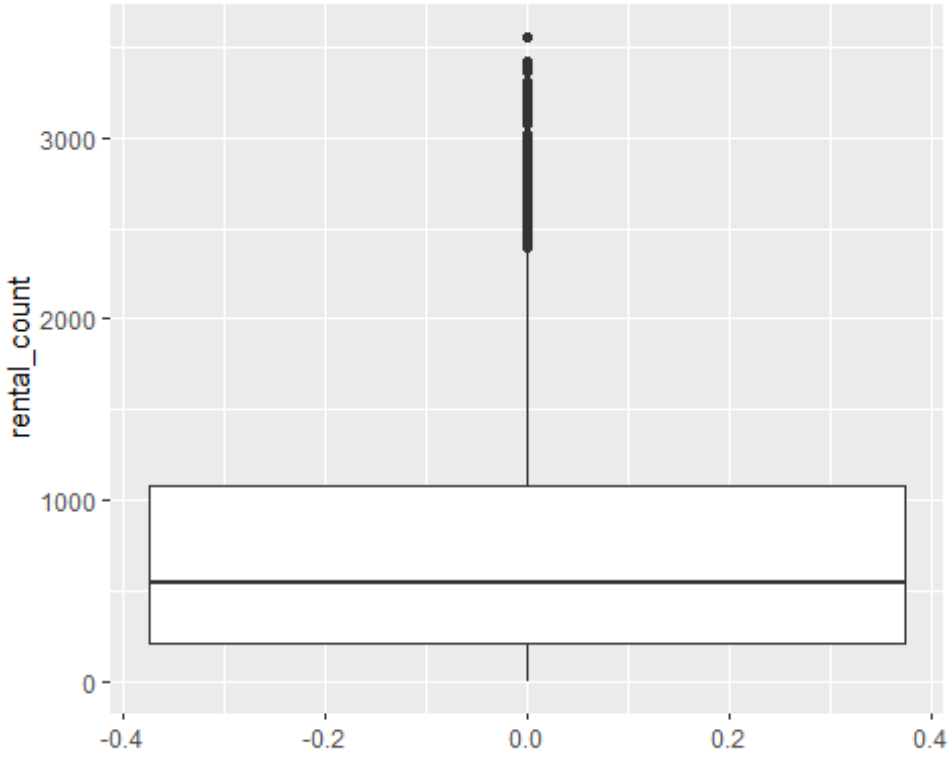
```
19.5 -19.3 -19.8 -22.4 ...
##  $ Solar_Radiation      : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rainfall_in_mm       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snowfall_in_cm       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Seasons              : chr  "Winter" "Winter" "Winter" "Winter"
 ...
##  $ Holiday              : chr  "No Holiday" "No Holiday" "No Holiday
" "No Holiday" ...
##  $ Functioning_Day      : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
ggplot(SeoulBikeData_Function, aes(x=rental_count)) + geom_histogram(ae
s(y=..density..),color="blue", fill="white") + geom_density()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(SeoulBikeData_Function, aes(y=rental_count)) +
  geom_boxplot()
```




```r
summary(SeoulBikeData_Function$rental_count)
```

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

```r
summary(SeoulBikeData_Function$Temperature)
```

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

```r
summary(SeoulBikeData_Function$Humidity)
```

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

```r
summary(SeoulBikeData_Function$Wind_speed)
```

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

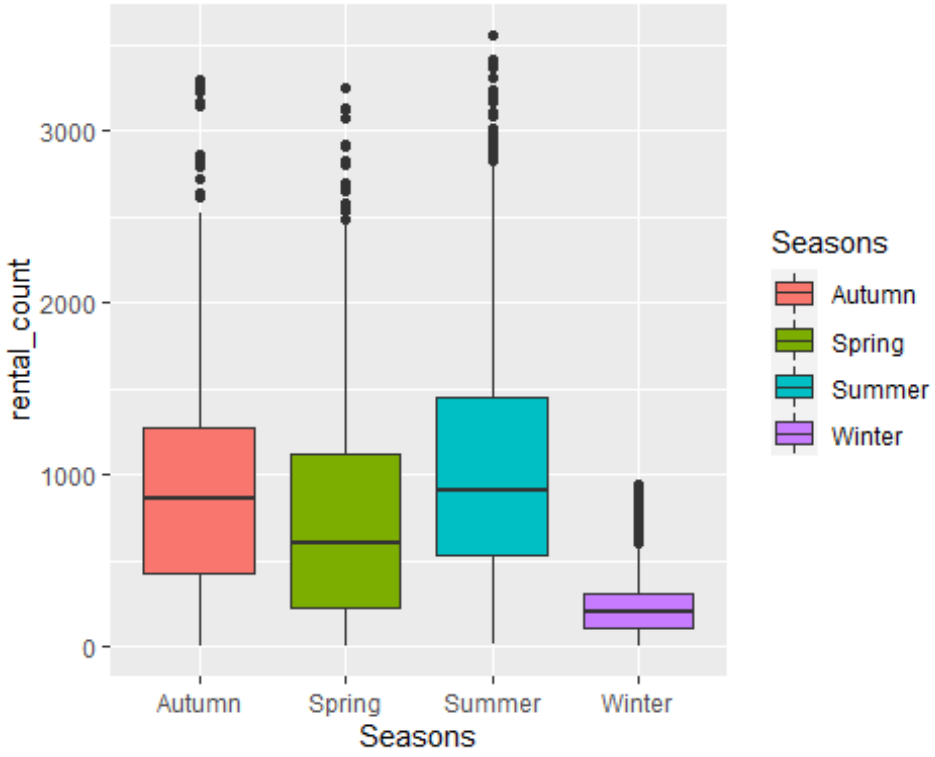
```r
summary(SeoulBikeData_Function$Visibility_in_10m)
```

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

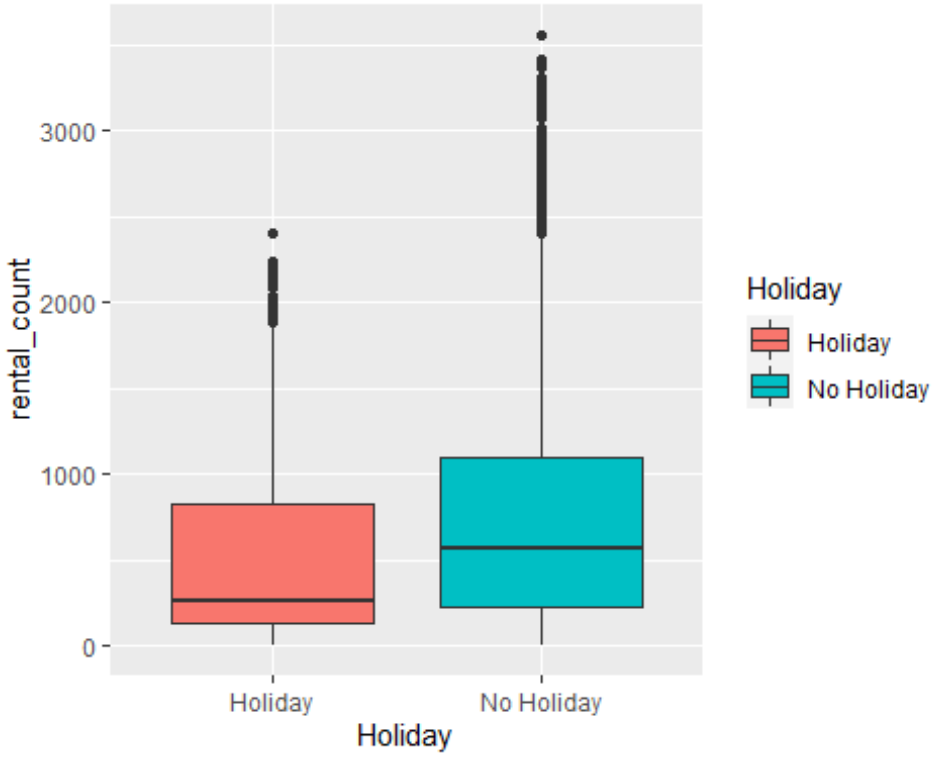
```r
summary(SeoulBikeData_Function$Dew_point_temperature)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -30.600  -5.100   4.700   3.945  15.200  27.200
```

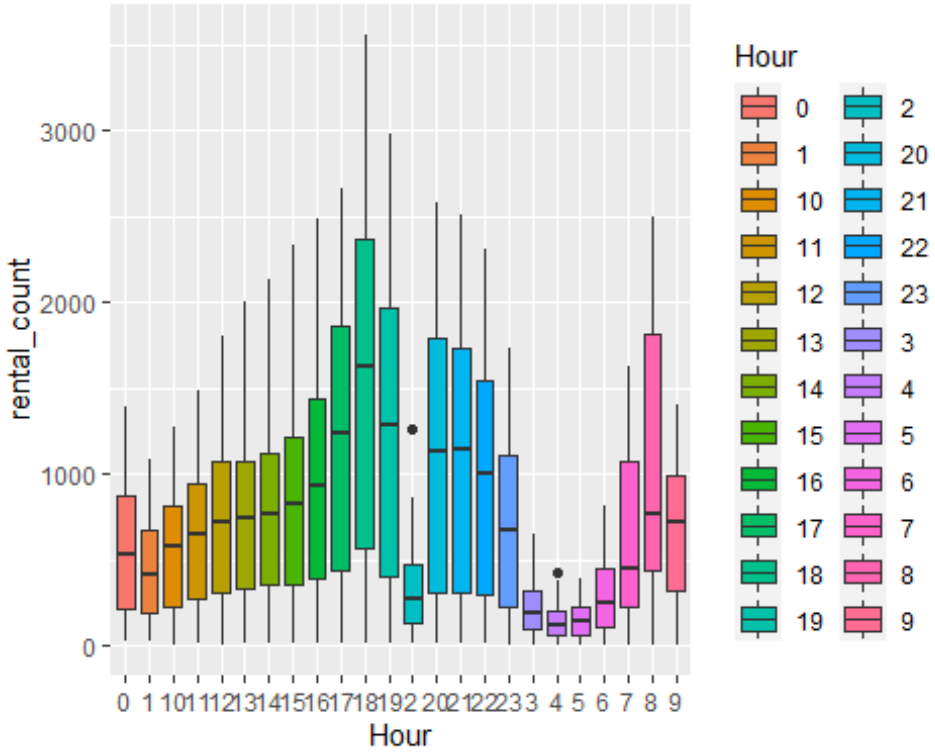
```
summary(SeoulBikeData_Function$Solar_Radiation)

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

summary(SeoulBikeData_Function$Rainfall_in_mm)

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

summary(SeoulBikeData_Function$Snowfall_in_cm)

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

ggplot(SeoulBikeData_Function, aes(x=Seasons , y=rental_count, fill=Sea
sons)) +
  geom_boxplot()
```




```
ggplot(SeoulBikeData_Function, aes(x=Holiday , y=rental_count, fill=Hol
iday)) +
  geom_boxplot()
```




```
count_vs_hour <- SeoulBikeData_Function[2:3]
count_vs_hour$Hour <- as.character(count_vs_hour$Hour)
str(count_vs_hour)

## 'data.frame':    8465 obs. of  2 variables:
##  $ rental_count: int  254 204 173 107 78 100 181 460 930 490 ...
##  $ Hour        : chr  "0" "1" "2" "3" ...

ggplot(count_vs_hour,aes(x=Hour , y=rental_count,fill = Hour)) + geom_b
oxplot()
```

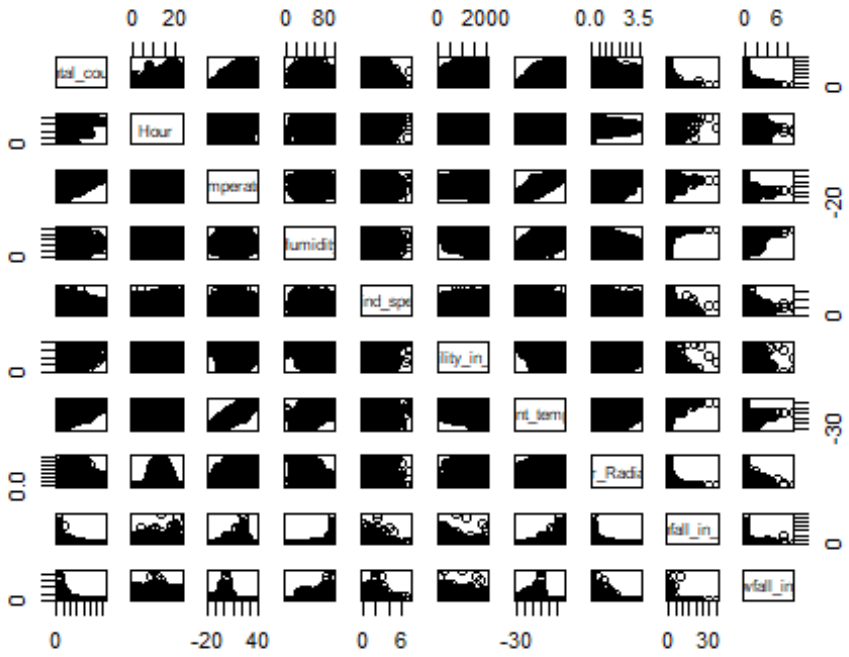



```
SeoulBikeData_Function$Seasons[SeoulBikeData_Function$Seasons == 'Winte
r'] <- 1
SeoulBikeData_Function$Seasons[SeoulBikeData_Function$Seasons == 'Sprin
g'] <- 2
SeoulBikeData_Function$Seasons[SeoulBikeData_Function$Seasons == 'Summe
r'] <- 3
SeoulBikeData_Function$Seasons[SeoulBikeData_Function$Seasons == 'Autum
n'] <- 4
SeoulBikeData_Function$Seasons <- as.numeric(SeoulBikeData_Function$Sea
sons)
SeoulBikeData_Function %>% count(Seasons)

##   Seasons    n
## 1       1 2160
## 2       2 2160
## 3       3 2208
## 4       4 1937

SeoulBikeData_Function %>% count(Holiday)

##      Holiday    n
## 1    Holiday  408
## 2 No Holiday 8057

SeoulBikeData_Function$Holiday[SeoulBikeData_Function$Holiday == 'Holid
ay'] <- 1
SeoulBikeData_Function$Holiday[SeoulBikeData_Function$Holiday == 'No Ho
```

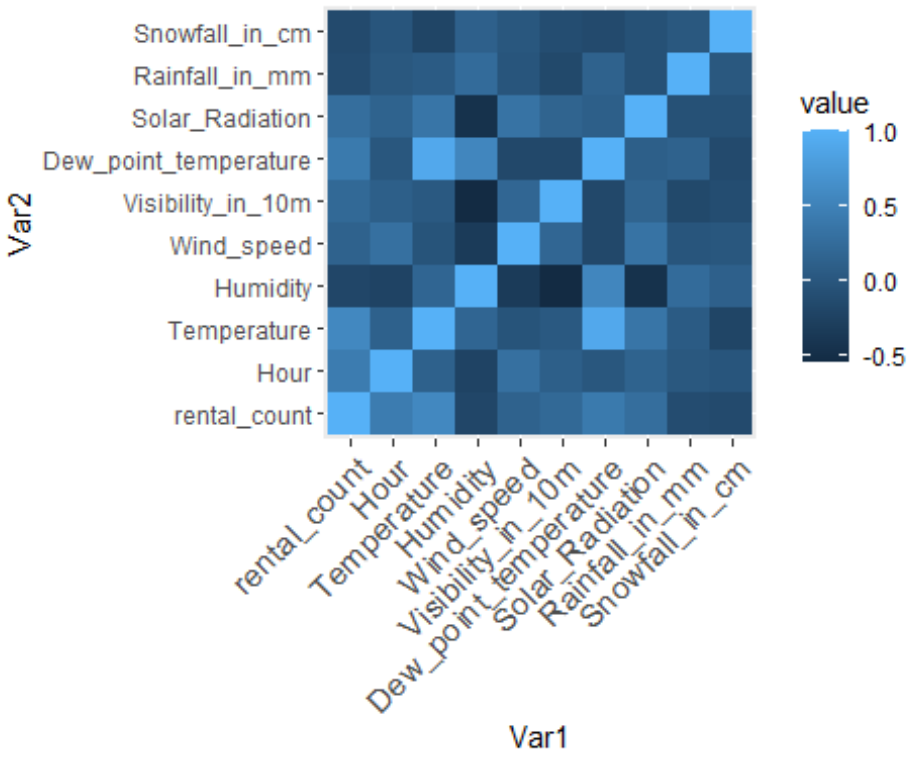
```
liday'] <- 2
SeoulBikeData_Function$Holiday <- as.numeric(SeoulBikeData_Function$Hol
iday)
SeoulBikeData_Function %>% count(Holiday)

##   Holiday    n
## 1       1  408
## 2       2 8057

correlation_test <- SeoulBikeData_Function[2:11]
str(correlation_test)

## 'data.frame':    8465 obs. of  10 variables:
##  $ rental_count         : int  254 204 173 107 78 100 181 460 930 49
0 ...
##  $ Hour                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temperature          : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -
7.6 -6.5 ...
##  $ Humidity             : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind_speed           : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5
 ...
##  $ Visibility_in_10m    : int  2000 2000 2000 2000 2000 2000 2000 20
00 2000 1928 ...
##  $ Dew_point_temperature: num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -
19.5 -19.3 -19.8 -22.4 ...
##  $ Solar_Radiation      : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rainfall_in_mm       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snowfall_in_cm       : num  0 0 0 0 0 0 0 0 0 0 ...

cor(correlation_test)

##                       rental_count         Hour Temperature   Humidi
ty
## rental_count             1.0000000  0.425255882  0.56274017 -0.20197
27
## Hour                     0.4252559  1.000000000  0.12274182 -0.23593
72
## Temperature              0.5627402  0.122741816  1.00000000  0.16642
52
## Humidity                -0.2019727 -0.235937180  0.16642522  1.00000
00
## Wind_speed               0.1250219  0.287779819 -0.03848085 -0.33735
24
## Visibility_in_10m        0.2123228  0.103869318  0.02826206 -0.54854
18
## Dew_point_temperature    0.4002628  0.004691479  0.91446695  0.53940
24
## Solar_Radiation          0.2738616  0.144658482  0.35484355 -0.45727
27
## Rainfall_in_mm          -0.1286261  0.014344645  0.05214889  0.23691
69
```

```
## Snowfall_in_cm          -0.1516108 -0.022082499 -0.21774581  0.11012
65
##                         Wind_speed Visibility_in_10m Dew_point_tempe
rature
## rental_count           0.125021946        0.21232278           0.400
262829
## Hour                    0.287779819        0.10386932           0.004
691479
## Temperature            -0.038480848        0.02826206           0.914
466952
## Humidity               -0.337352380       -0.54854183           0.539
402446
## Wind_speed              1.000000000        0.18042765          -0.177
170126
## Visibility_in_10m       0.180427649        1.00000000          -0.182
586445
## Dew_point_temperature -0.177170126       -0.18258645           1.000
000000
## Solar_Radiation         0.326221868        0.15304614           0.098
524979
## Rainfall_in_mm         -0.024931327       -0.17035180           0.126
812453
## Snowfall_in_cm         -0.003789344       -0.12285973          -0.149
759793
##                        Solar_Radiation Rainfall_in_mm Snowfall_in_cm
## rental_count                0.27386155   -0.128626093   -0.151610753
## Hour                        0.14465848    0.014344645   -0.022082499
## Temperature                 0.35484355    0.052148891   -0.217745811
## Humidity                   -0.45727269    0.236916912    0.110126502
## Wind_speed                  0.32622187   -0.024931327   -0.003789344
## Visibility_in_10m           0.15304614   -0.170351798   -0.122859731
## Dew_point_temperature       0.09852498    0.126812453   -0.149759793
## Solar_Radiation             1.00000000   -0.074157271   -0.073379874
## Rainfall_in_mm             -0.07415727    1.000000000    0.008604092
## Snowfall_in_cm             -0.07337987    0.008604092    1.000000000
```

```r
plot(correlation_test)
```

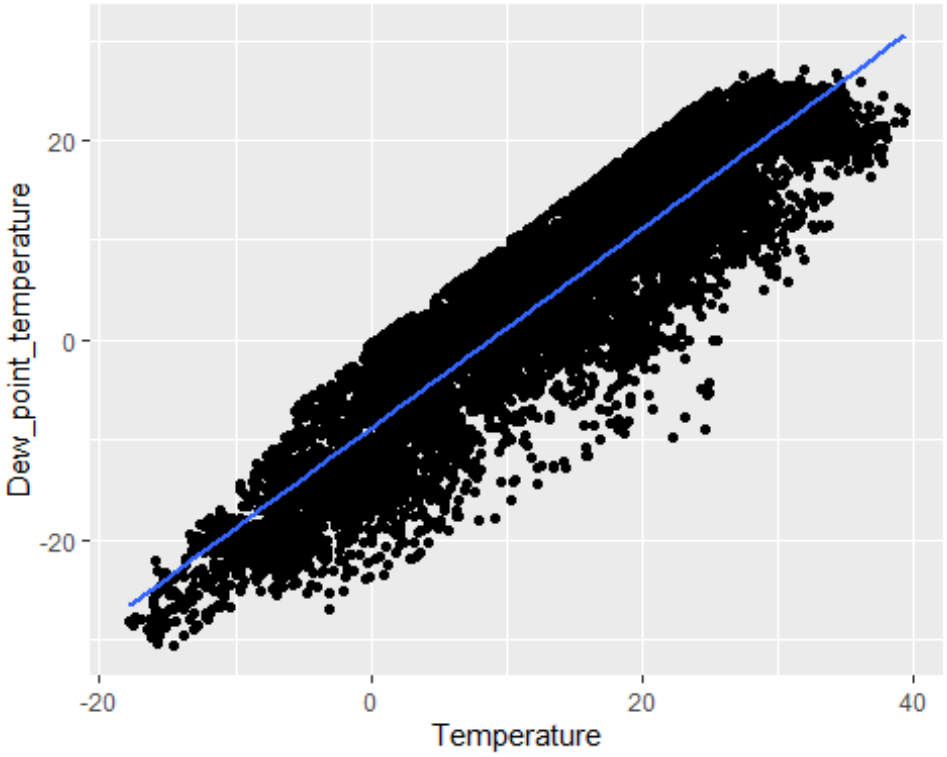



```
library(reshape2)

##
## 载入程辑包：'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

melted<- melt(round(cor(correlation_test),2))
library(ggplot2)
ggplot(data = melted, aes(x=Var1, y=Var2, fill=value)) + geom_tile() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hj
ust = 1))+ coord_fixed()
```




```
ggplot(correlation_test, aes(x=Temperature, y=Dew_point_temperature)) +

  geom_point()+
  geom_smooth(method=lm, se=FALSE)
```

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




```
test_model_1 <- lm(Temperature~Dew_point_temperature,data = correlation
_test)
summary(test_model_1)

##
## Call:
## lm(formula = Temperature ~ Dew_point_temperature, data = correlation
_test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2582 -3.7426 -0.6475  3.1734 22.6494
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9.473515   0.055553   170.5   <2e-16 ***
## Dew_point_temperature 0.835880   0.004021   207.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.898 on 8463 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8362
## F-statistic: 4.322e+04 on 1 and 8463 DF,  p-value: < 2.2e-16

SeoulBikeData_Function_mlr <- SeoulBikeData_Function[c(2:7,9:13)]
head(SeoulBikeData_Function_mlr)
```

```
##   rental_count Hour Temperature Humidity Wind_speed Visibility_in_10
m
## 1          254    0        -5.2       37        2.2              200
0
## 2          204    1        -5.5       38        0.8              200
0
## 3          173    2        -6.0       39        1.0              200
0
## 4          107    3        -6.2       40        0.9              200
0
## 5           78    4        -6.0       36        2.3              200
0
## 6          100    5        -6.4       37        1.5              200
0
##   Solar_Radiation Rainfall_in_mm Snowfall_in_cm Seasons Holiday
## 1               0              0              0       1       2
## 2               0              0              0       1       2
## 3               0              0              0       1       2
## 4               0              0              0       1       2
## 5               0              0              0       1       2
## 6               0              0              0       1       2
```

```
library(MASS)
```

```
##
## 载入程辑包：'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(leaps)
```

```
library(caret)
```

```
## 载入需要的程辑包：lattice
```

```
##
## 载入程辑包：'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
train_control <- trainControl(method = "cv", number = 10,p = 0.75)
```

```
set.seed(903)
model_mlr <- train(rental_count~., data=SeoulBikeData_Function_mlr,meth
od = "lm", trControl = train_control,metric = "RMSE")
model_mlr
```

```
## Linear Regression
##
## 8465 samples
##   10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7617, 7620, 7619, 7618, 7619, 7618, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   438.2147  0.535284  326.9658
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
summary(model_mlr)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1236.44  -278.37   -54.88   213.52  2231.49
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        57.345482  55.244293   1.038 0.299284
## Hour               28.706993   0.746413  38.460  < 2e-16 ***
## Temperature        26.609709   0.579858  45.890  < 2e-16 ***
## Humidity           -8.431415   0.372911 -22.610  < 2e-16 ***
## Wind_speed         19.218473   5.236120   3.670 0.000244 ***
## Visibility_in_10m  -0.007595   0.009984  -0.761 0.446853
## Solar_Radiation   -79.693998   7.577910 -10.517  < 2e-16 ***
## Rainfall_in_mm    -63.431446   4.375804 -14.496  < 2e-16 ***
## Snowfall_in_cm     18.817067  11.200192   1.680 0.092981 .
## Seasons           110.785998   5.623107  19.702  < 2e-16 ***
## Holiday           128.077284  22.298547   5.744 9.58e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 437.8 on 8454 degrees of freedom
## Multiple R-squared:  0.536,  Adjusted R-squared:  0.5354
## F-statistic: 976.5 on 10 and 8454 DF,  p-value: < 2.2e-16
```

```
set.seed(903)
step_model <- train(rental_count~., data = SeoulBikeData_Function_mlr,
                    method = "lmStepAIC",
                    trControl = train_control,
```

```
                    trace = FALSE)
step_model

## Linear Regression with Stepwise Selection
##
## 8465 samples
##   10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7617, 7620, 7619, 7618, 7619, 7618, ...
## Resampling results:
##
##   RMSE     Rsquared   MAE
##   438.212  0.5352863  326.8619

summary(step_model)

##
## Call:
## lm(formula = .outcome ~ Hour + Temperature + Humidity + Wind_speed +

##     Solar_Radiation + Rainfall_in_mm + Snowfall_in_cm + Seasons +
##     Holiday, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1239.21  -278.14   -54.79   213.14  2233.86
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      39.0364    49.7244   0.785 0.432443
## Hour             28.7498     0.7443  38.628  < 2e-16 ***
## Temperature      26.5632     0.5766  46.068  < 2e-16 ***
## Humidity         -8.2726     0.3090 -26.773  < 2e-16 ***
## Wind_speed       18.8501     5.2135   3.616 0.000301 ***
## Solar_Radiation -78.3414     7.3662 -10.635  < 2e-16 ***
## Rainfall_in_mm  -63.2961     4.3721 -14.477  < 2e-16 ***
## Snowfall_in_cm   18.9046    11.1993   1.688 0.091446 .
## Seasons         109.9105     5.5039  19.969  < 2e-16 ***
## Holiday         128.2228    22.2972   5.751  9.2e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 437.8 on 8455 degrees of freedom
## Multiple R-squared:  0.536,  Adjusted R-squared:  0.5355
## F-statistic:  1085 on 9 and 8455 DF,  p-value: < 2.2e-16

subsets<-regsubsets(rental_count~.,
data=SeoulBikeData_Function_mlr, nbest=1)
```

```
sub.sum <- summary(subsets)
as.data.frame(sub.sum$outmat)
```

```
##          Hour Temperature Humidity Wind_speed Visibility_in_10m Sola
r_Radiation
## 1  ( 1 )               *
## 2  ( 1 )    *          *
## 3  ( 1 )    *          *        *
## 4  ( 1 )    *          *        *
## 5  ( 1 )    *          *        *
## 6  ( 1 )    *          *        *
          *
## 7  ( 1 )    *          *        *
          *
## 8  ( 1 )    *          *        *          *
          *
##          Rainfall_in_mm Snowfall_in_cm Seasons Holiday
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )                                   *
## 5  ( 1 )              *                    *
## 6  ( 1 )              *                    *
## 7  ( 1 )              *                    *       *
## 8  ( 1 )              *                    *       *
```

```
library(rpart)
library(rpart.plot)
```

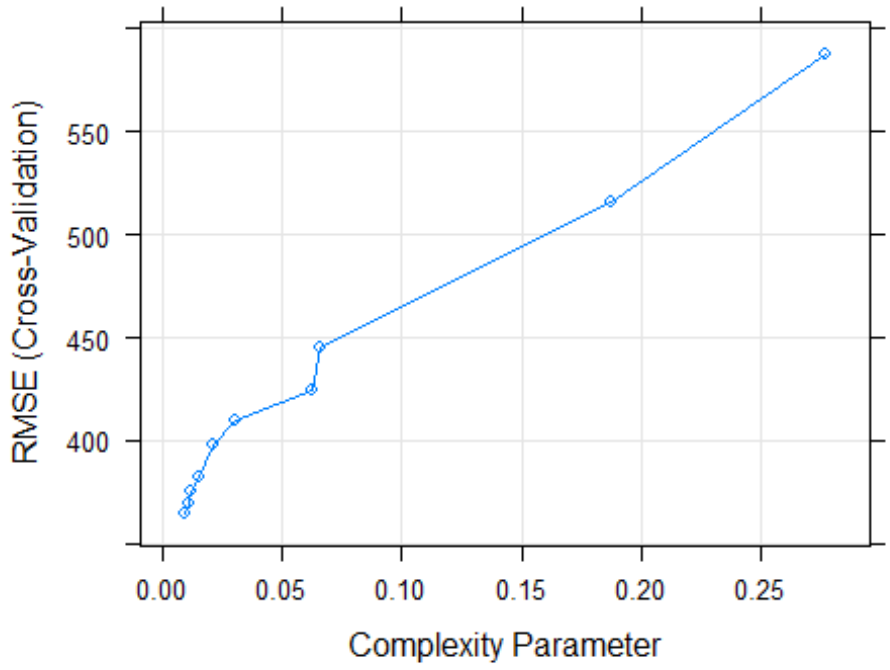
```
set.seed(903)
dt_model <- train(rental_count ~ .,
   tuneLength = 10,metric = "RMSE", data = SeoulBikeData_Function_mlr,
method = "rpart",trControl =train_control)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
trainInfo, :
## There were missing values in resampled performance measures.
```

```
print(dt_model)
```

```
## CART
##
## 8465 samples
##   10 predictor
##
## No pre-processing
```

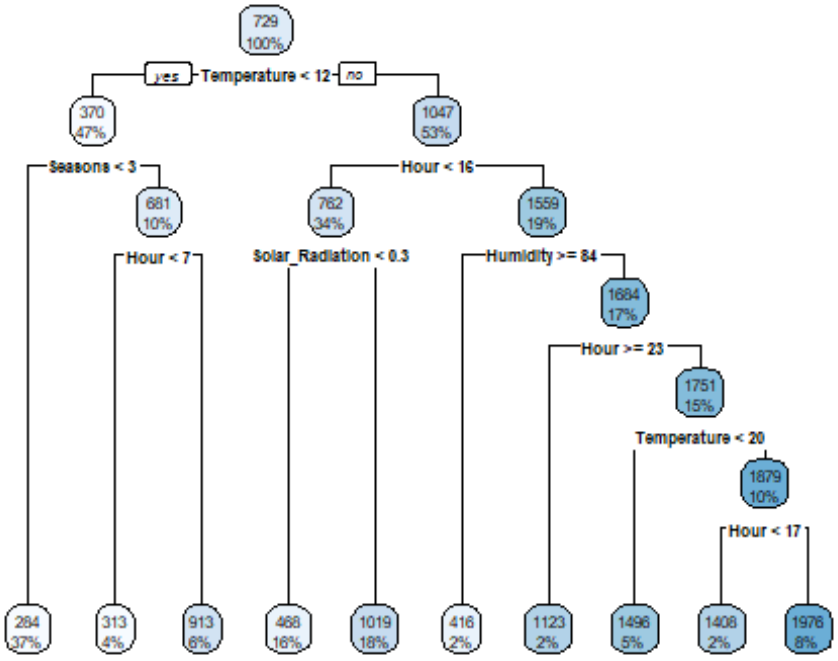
```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7617, 7620, 7619, 7618, 7619, 7618, ...
## Resampling results across tuning parameters:
##
##   cp           RMSE      Rsquared   MAE
##   0.009541003  364.5774  0.6785039  259.3501
##   0.011345923  369.3711  0.6699230  263.2851
##   0.012109507  375.2878  0.6593205  267.7438
##   0.015410349  381.8530  0.6471406  273.2710
##   0.021076976  398.3559  0.6158454  287.3467
##   0.030598818  410.0091  0.5929280  300.4972
##   0.062278865  424.3441  0.5634851  313.4617
##   0.065964898  445.3929  0.5174085  330.9601
##   0.187538412  515.4791  0.3547558  380.1379
##   0.276927210  587.5918  0.2643416  452.1301
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.009541003.

plot(dt_model)
```




```
dt_model$finalModel

## n= 8465
##
## node), split, n, deviance, yval
##       * denotes terminal node
```

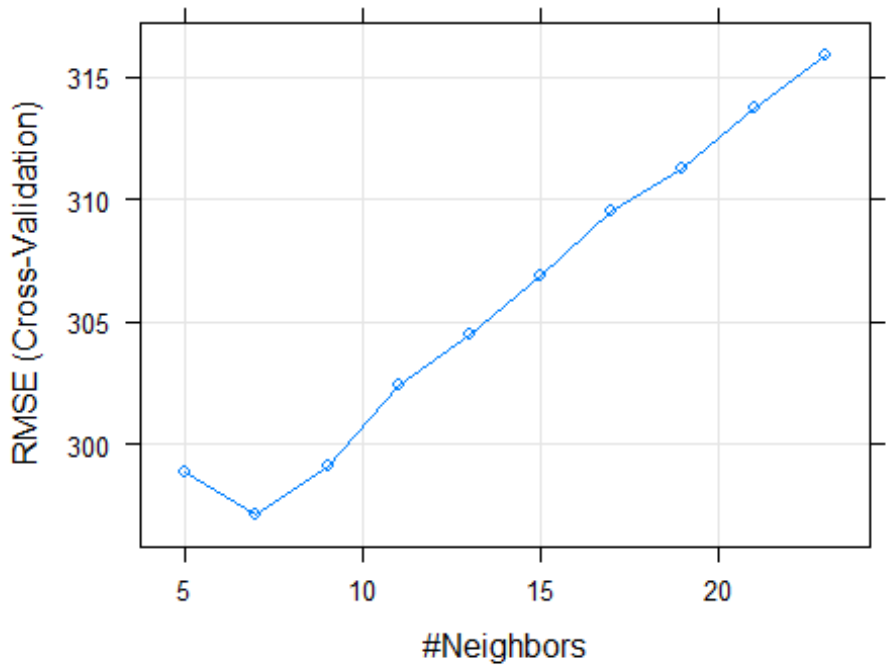
```
##
##   1) root 8465 3492374000  729.1570
##     2) Temperature< 12.05 3976  484876500  370.0025
##       4) Seasons< 3 3112  194179300  283.6199 *
##       5) Seasons>=3 864  183834700  681.1400
##        10) Hour< 6.5 334   11483650  313.4581 *
##        11) Hour>=6.5 530   98742340  912.8491 *
##     3) Temperature>=12.05 4489 2040364000 1047.2680
##       6) Hour< 15.5 2882  649750900  762.0399
##        12) Solar_Radiation< 0.295 1344  163436500  468.1652 *
##        13) Solar_Radiation>=0.295 1538  268813300 1018.8460 *
##       7) Hour>=15.5 1607  735658700 1558.7960
##        14) Humidity>=83.5 159   35515400  416.1950 *
##        15) Humidity< 83.5 1448  469769300 1684.2610
##          30) Hour>=22.5 153   11268640 1123.3790 *
##          31) Hour< 22.5 1295  404681900 1750.5270
##            62) Temperature< 19.95 433   96791000 1495.5520 *
##            63) Temperature>=19.95 862  265600000 1878.6070
##             126) Hour< 16.5 148   29729470 1407.6890 *
##             127) Hour>=16.5 714  196246400 1976.2200 *
```

```
rpart.plot(dt_model$finalModel)
```




```
set.seed(903)
model_knn <- train(
   rental_count~., data = SeoulBikeData_Function_mlr, method = "knn",
  trControl = train_control,
```

```
  preProcess = c("center","scale"),
  tuneLength = 10)
plot(model_knn)
```




```
model_knn
```

```
## k-Nearest Neighbors
##
## 8465 samples
##   10 predictor
##
## Pre-processing: centered (10), scaled (10)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7617, 7620, 7619, 7618, 7619, 7618, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    5  298.9053  0.7844021  192.3623
##    7  297.1353  0.7863595  194.3197
##    9  299.0939  0.7834965  197.6511
##   11  302.4515  0.7787076  200.7736
##   13  304.4817  0.7758977  203.0266
##   15  306.8399  0.7724442  204.9082
##   17  309.5172  0.7685921  207.8098
##   19  311.2797  0.7660385  209.7445
##   21  313.7222  0.7624218  211.8016
##   23  315.8738  0.7591868  213.7759
```

```
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 7.
```

```
model_knn$bestTune
```

```
##   k
## 2 7
```