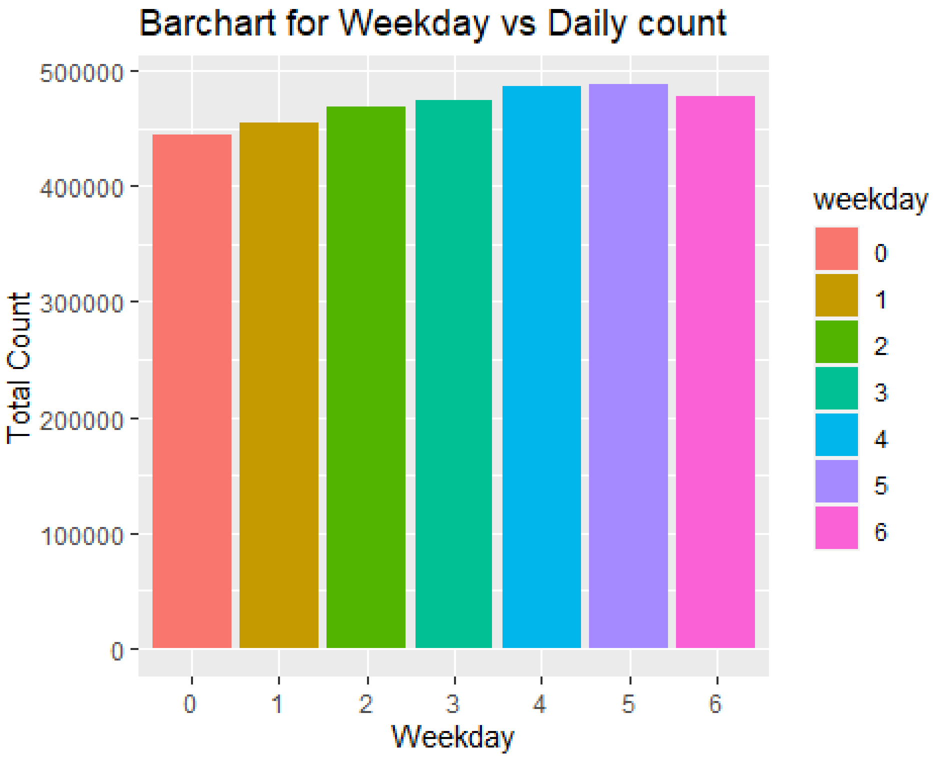
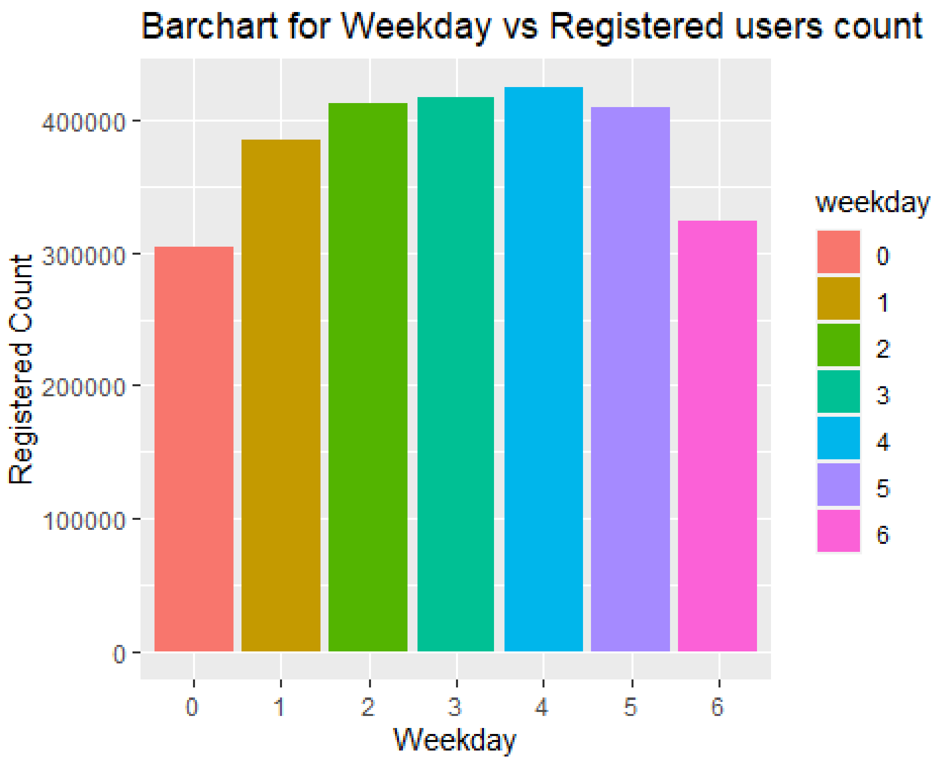
```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidyr)
library(ggplot2)
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(coefplot)
library(corrplot)

## corrplot 0.90 loaded

library(rpart)
library(rpart.plot)
library(DMwR2)

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

library(stringr)
library(data.table)

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose

library(mltools)

## Warning: package 'mltools' was built under R version 4.1.2
```

```
##
## Attaching package: 'mltools'

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

library(olsrr)

## Warning: package 'olsrr' was built under R version 4.1.2

##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers

library(leaps)
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

library(dplyr)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

library(MASS)

##
## Attaching package: 'MASS'
## The following object is masked from 'package:olsrr':
##
##     cement

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

source('eda_grid_funcs.R')


#reading the data into a dataframe and finding the classes of all variables:
d2<-read_csv(‘day.csv’)

## Rows: 731 Columns: 16

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## dbl  (15): instant, season, yr, mnth, holiday, weekday, workingday,
weathers...
## date  (1): dteday

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

sapply(d2, class)

##    instant     dteday     season         yr       mnth    holiday
weekday
##  "numeric"     "Date"  "numeric"  "numeric"  "numeric"  "numeric"
"numeric"
## workingday weathersit       temp      atemp        hum  windspeed
casual
##  "numeric"  "numeric"  "numeric"  "numeric"  "numeric"  "numeric"
"numeric"
## registered        cnt
##  "numeric"  "numeric"

#EDA:
d2$weathersit<-as.factor(d2$weathersit)
d2$yr<-as.factor(d2$yr)
d2$mnth<-as.factor(d2$mnth)
d2$workingday<-as.factor(d2$workingday)
d2$weekday<-as.factor(d2$weekday)
d2$holiday<-as.factor(d2$holiday)
d2$season<-as.factor(d2$season)
```

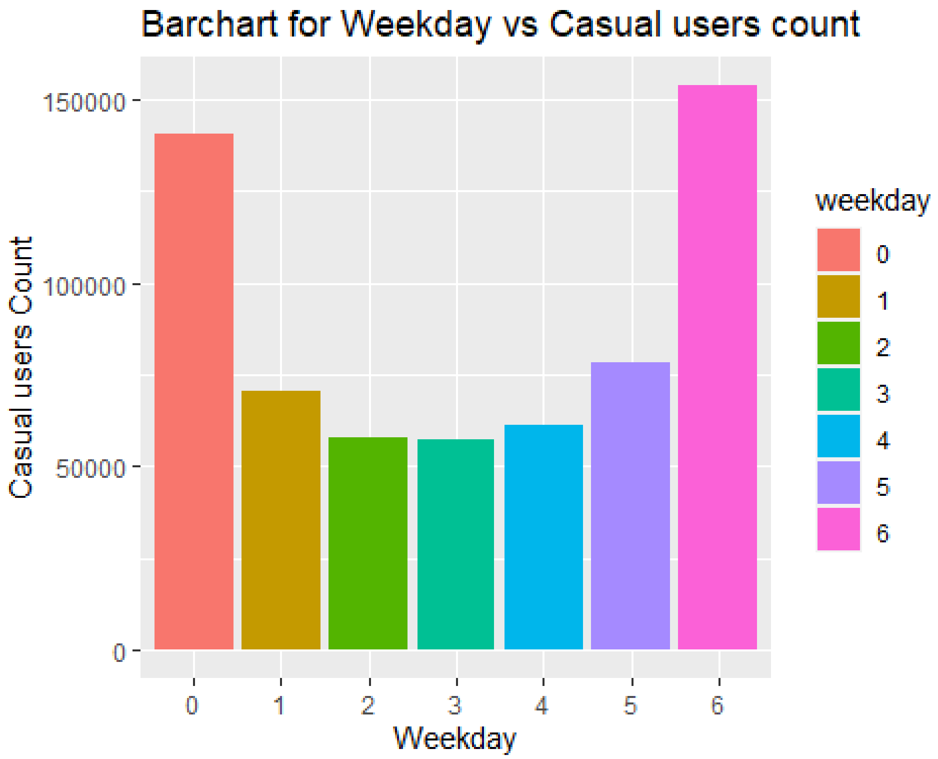
```
options(scipen = 10000)

ggplot(d2, aes(x =weekday, y = cnt,fill = weekday)) +
  geom_col()+labs(title= "Barchart for Weekday vs Daily count",
                  y="Total Count", x = "Weekday")
```



```
ggplot(d2, aes(x =weekday, y = registered,fill = weekday)) +
  geom_col()+labs(title= "Barchart for Weekday vs Registered users count",
                  y="Registered Count", x = "Weekday")
```

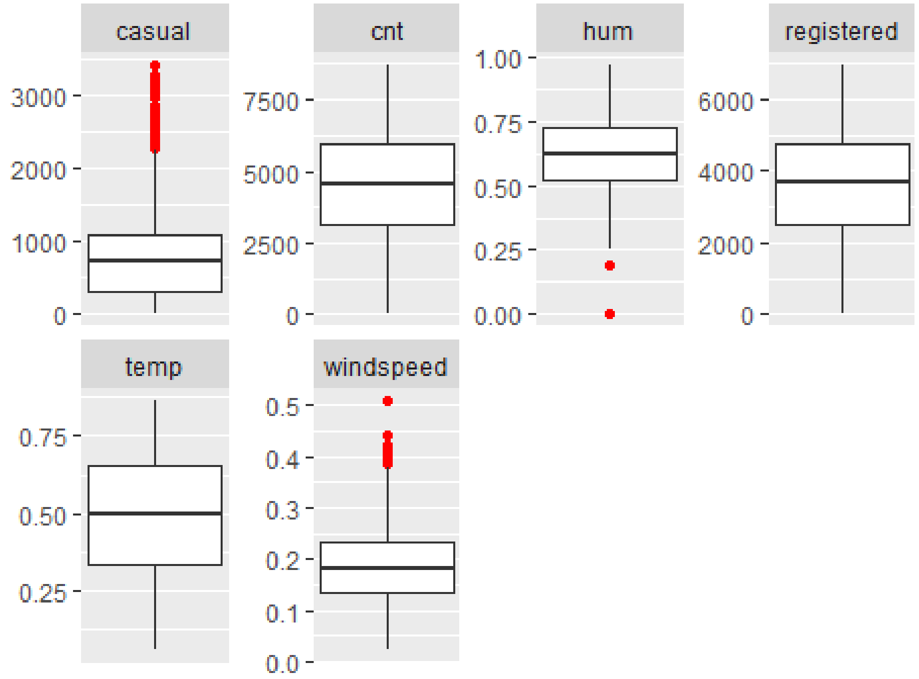


```
ggplot(d2, aes(x =weekday, y = casual,fill = weekday)) +
  geom_col()+labs(title= "Barchart for Weekday vs Casual users count",
                  y="Casual users Count", x = "Weekday")
```

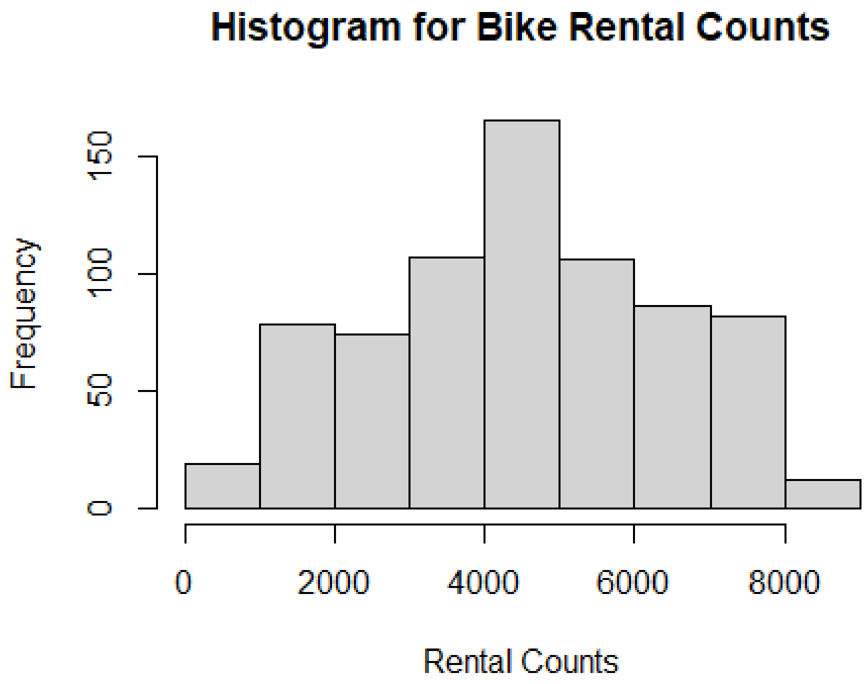


```
boxplot_grid(d2, 1,vars = c("casual", "registered", "cnt","windspeed","hum",
"temp"), ncol=4, nrow = 4)
```

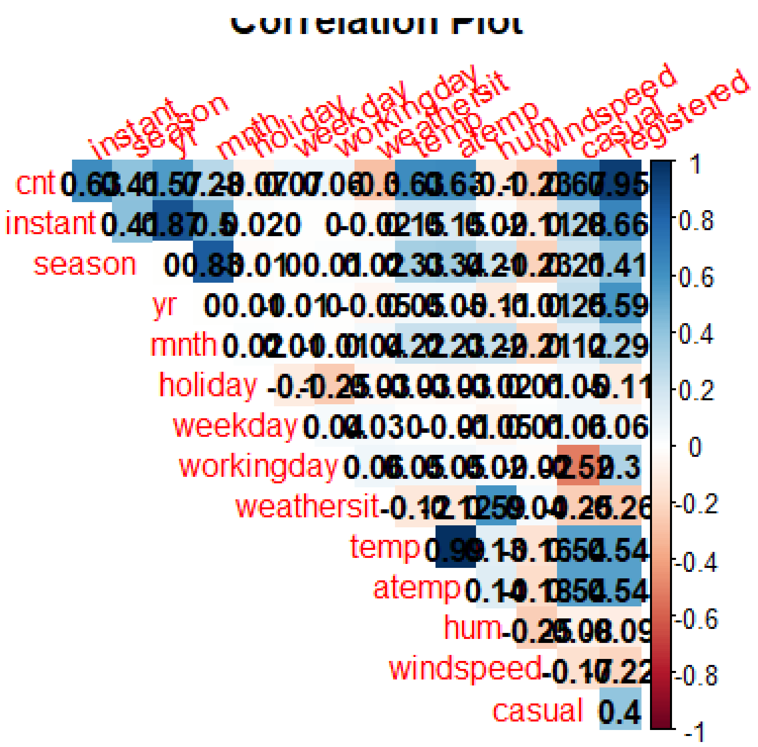
Figure 1. Boxplot grid



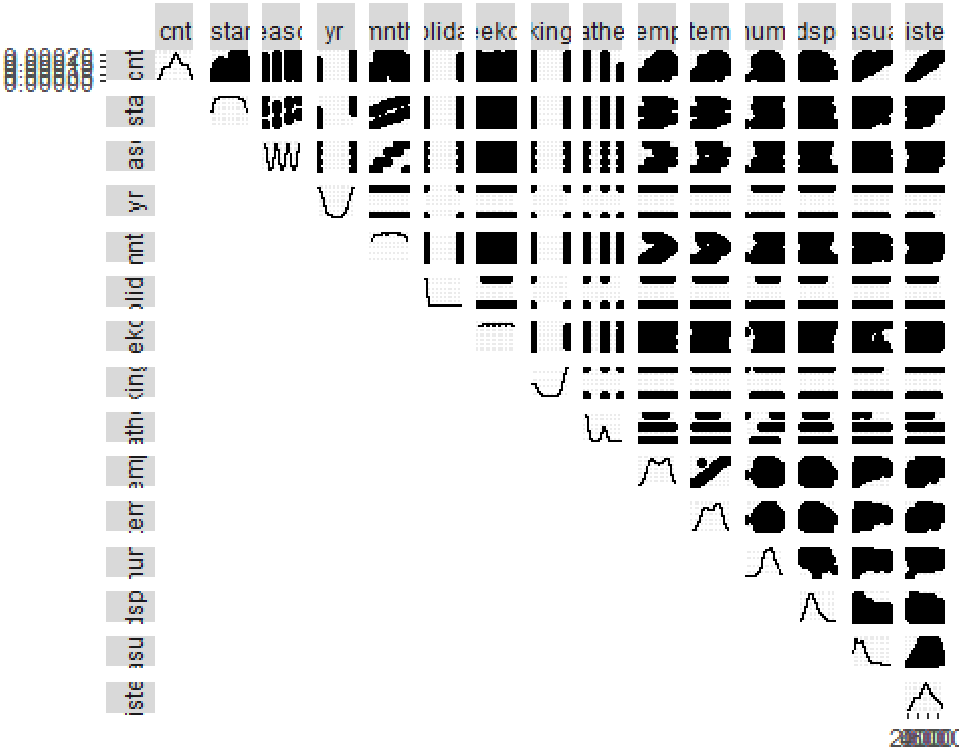
```
hist(d2$cnt, main = "Histogram for Bike Rental Counts", xlab = "Rental
Counts",)
```



```
#Seems to be Normally distributed.

#Constructing the corrplot:

d2<-read_csv('day.csv')

## Rows: 731 Columns: 16

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## dbl  (15): instant, season, yr, mnth, holiday, weekday, workingday,
weathers...
## date  (1): dteday

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

vars <- dplyr::select(d2,`cnt`, everything(), -dteday)
corrplot(cor(vars[sapply(vars, function(x) !is.factor(x))]),type="upper",
method="color", diag=FALSE,
         tl.srt=30, addCoef.col="black", main="Correlation Plot")
```



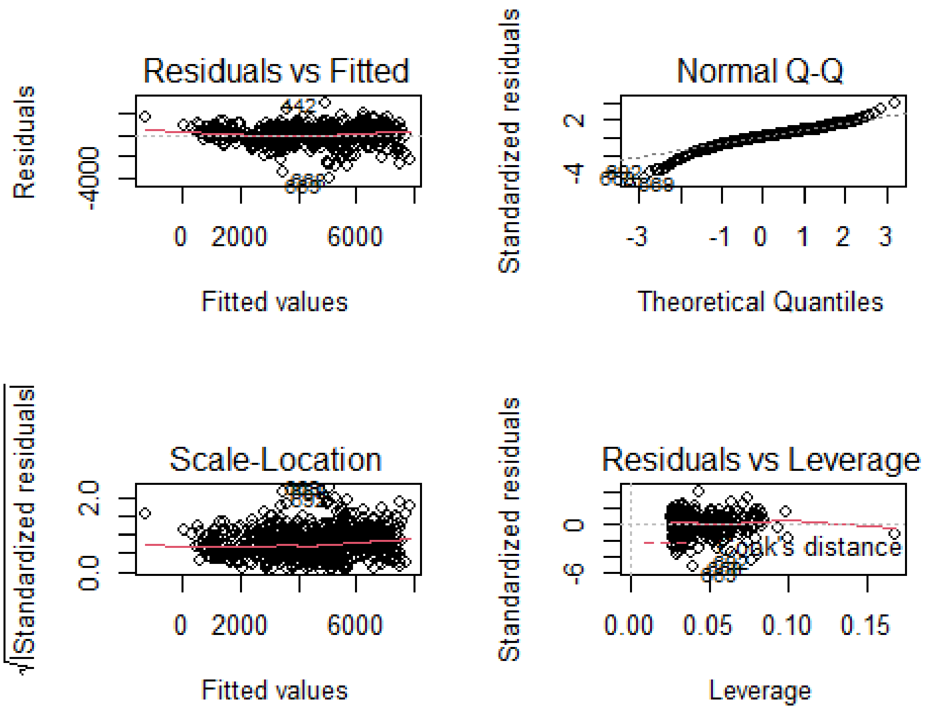
```
#ggpairs plot:
ggpairs(vars, lower=list(continuous='blank',
                         combo='blank',
                         discrete='blank'),

        upper=list(continuous="points",
                   combo="facethist", discrete="facetbar"),
        switch="y")
```



```
#Assigning the appropriate classes to all variables:
d2$weathersit<-as.factor(d2$weathersit)
d2$yr<-as.factor(d2$yr)
d2$mnth<-as.factor(d2$mnth)
d2$workingday<-as.factor(d2$workingday)
d2$weekday<-as.factor(d2$weekday)
d2$holiday<-as.factor(d2$holiday)
d2$season<-as.factor(d2$season)



#One-hot encoding:

df<- one_hot(as.data.table(d2))
head(df,3)

##    instant     dteday season_1 season_2 season_3 season_4 yr_0 yr_1 mnth_1
## 1:       1 2011-01-01        1        0        0        0    1    0      1
## 2:       2 2011-01-02        1        0        0        0    1    0      1
## 3:       3 2011-01-03        1        0        0        0    1    0      1
##    mnth_2 mnth_3 mnth_4 mnth_5 mnth_6 mnth_7 mnth_8 mnth_9 mnth_10 mnth_11
## 1:      0      0      0      0      0      0      0      0       0       0
## 2:      0      0      0      0      0      0      0      0       0       0
## 3:      0      0      0      0      0      0      0      0       0       0
##    mnth_12 holiday_0 holiday_1 weekday_0 weekday_1 weekday_2 weekday_3
## 1:       0         1         0         0         0         0         0
## 2:       0         1         0         1         0         0         0
## 3:       0         1         0         0         1         0         0
##    weekday_4 weekday_5 weekday_6 workingday_0 workingday_1 weathersit_1
## 1:         0         0         1            1            0            0
## 2:         0         0         0            1            0            0
## 3:         0         0         0            0            1            1
##    weathersit_2 weathersit_3     temp    atemp      hum windspeed casual
## 1:            1            0 0.344167 0.363625 0.805833  0.160446    331
## 2:            1            0 0.363478 0.353739 0.696087  0.248539    131
## 3:            0            0 0.196364 0.189405 0.437273  0.248309    120
##    registered  cnt
## 1:        654  985
## 2:        670  801
## 3:       1229 1349
```

```
  df1<-subset(df, select = -c(registered,casual,atemp,instant,dteday))

#Building a linear regression model:

df.lm<- lm((cnt)~ .,df1 )
summary(df.lm)
```

```
##
## Call:
## lm(formula = (cnt) ~ ., data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3960.9  -350.9    74.1   456.0  2919.9
##
## Coefficients: (8 not defined because of singularities)
##                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)     2909.91     404.91   7.187 0.000000000001700011 ***
## season_1       -1578.95     181.04  -8.722 < 0.0000000000000002 ***
## season_2        -689.65     212.36  -3.248             0.001219 **
## season_3        -746.71     191.42  -3.901             0.000105 ***
## season_4             NA         NA      NA                   NA
## yr_0           -2018.06      58.22 -34.660 < 0.0000000000000002 ***
## yr_1                 NA         NA      NA                   NA
## mnth_1            84.39     182.23   0.463             0.643439
## mnth_2           221.24     183.54   1.205             0.228450
## mnth_3           629.52     185.16   3.400             0.000712 ***
## mnth_4           540.88     242.19   2.233             0.025842 *
## mnth_5           807.91     257.73   3.135             0.001792 **
## mnth_6           574.94     262.55   2.190             0.028863 *
## mnth_7            92.79     279.39   0.332             0.739894
## mnth_8           489.30     267.52   1.829             0.067824 .
## mnth_9          1068.34     218.37   4.892 0.000001236091390315 ***
## mnth_10          605.33     163.54   3.701             0.000231 ***
## mnth_11          -26.97     154.85  -0.174             0.861767
## mnth_12             NA         NA      NA                    NA
## holiday_0       603.61     180.07   3.352              0.000845 ***
## holiday_1           NA         NA      NA                    NA
## weekday_0      -438.70     106.59  -4.116 0.0000431678951695 ***
## weekday_1      -223.82     109.49  -2.044              0.041306 *
## weekday_2      -129.57     106.87  -1.212              0.225774
## weekday_3       -61.29     107.03  -0.573              0.567060
## weekday_4       -53.49     106.96  -0.500              0.617163
## weekday_5       -10.09     106.96  -0.094              0.924837
## weekday_6           NA         NA      NA                    NA
## workingday_0        NA         NA      NA                    NA
## workingday_1        NA         NA      NA                    NA
## weathersit_1  1981.36     196.67  10.075 < 0.0000000000000002 ***
## weathersit_2  1516.15     184.23   8.230 0.000000000000000911 ***
## weathersit_3        NA         NA      NA                    NA
## temp          4487.30     411.84  10.896 < 0.0000000000000002 ***
## hum          -1518.18     292.21  -5.196 0.000000267729942701 ***
## windspeed    -2925.44     406.17  -7.202 0.00000000001526049 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 769.5 on 703 degrees of freedom
## Multiple R-squared:  0.848,  Adjusted R-squared:  0.8422
## F-statistic: 145.3 on 27 and 703 DF,  p-value: < 0.00000000000000022

par(mfrow=c(2,2))
plot(df.lm)
```



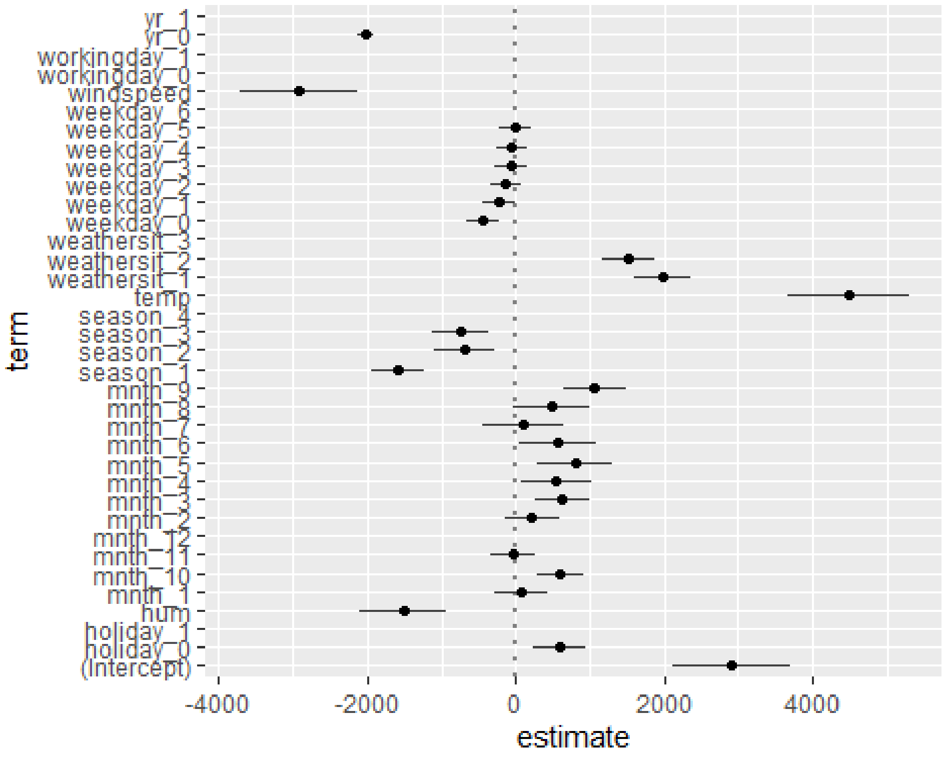
```
confint(df.lm)
```

```
##                    2.5 %      97.5 %
## (Intercept)   2114.93476  3704.88204
## season_1     -1934.39085 -1223.50225
## season_2     -1106.58050  -272.70954
## season_3     -1122.52764  -370.89435
## season_4              NA          NA
## yr_0         -2132.37795 -1903.74791
## yr_1                  NA          NA
## mnth_1        -273.38795   442.16661
## mnth_2        -139.11286   581.60202
## mnth_3         265.99387   993.04864
## mnth_4          65.38480  1016.38275
## mnth_5         301.89148  1313.92764
## mnth_6          59.46897  1090.41462
## mnth_7        -455.75259   641.34017
## mnth_8         -35.94245  1014.54477
## mnth_9         639.60623  1497.06871
## mnth_10        284.23752   926.41424
## mnth_11       -330.99229   277.04664
## mnth_12               NA          NA
## holiday_0      250.07442   957.13595
## holiday_1             NA          NA
## weekday_0     -647.97135  -229.42603
## weekday_1     -438.79102    -8.85278
## weekday_2     -339.38878    80.25537
## weekday_3     -271.42715   148.84470
## weekday_4     -263.50015   156.51517
## weekday_5     -220.08894   199.90032
## weekday_6             NA          NA
## workingday_0          NA          NA
## workingday_1          NA          NA
## weathersit_1  1595.22624  2367.48780
## weathersit_2  1154.44546  1877.86371
## weathersit_3          NA          NA
## temp          3678.72585  5295.88396
## hum          -2091.88206  -944.47302
## windspeed    -3722.89900 -2127.97659
```

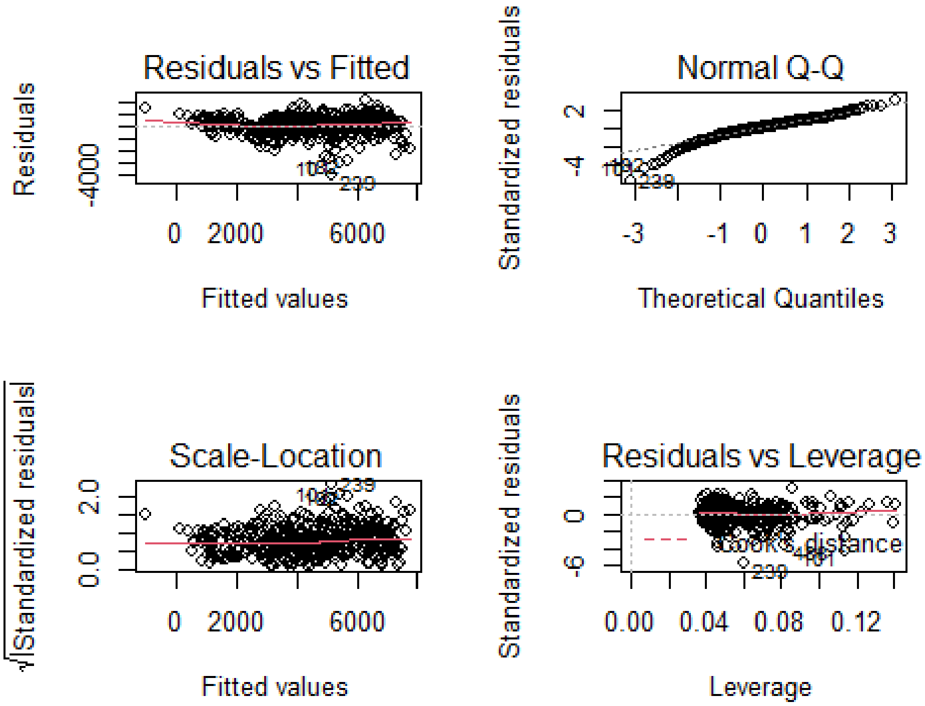
```
ggcoef(df.lm)
```

```
## Warning: Removed 8 rows containing missing values (geom_errorbarh).
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



```
#Best_Subsets selection:
set.seed(2000)
tr<-sample(nrow(df1), size = 500)
train<- df1[tr,]
test<-df1[!tr]







mod1<- lm(cnt~.,data=train)
sum1<-summary(mod1)
sum1$adj.r.squared

## [1] 0.8510152

cat("training-set RMSE: ", sum1$sigma)

## training-set RMSE:  739.5883

plot(mod1)
```



```
leaps<- regsubsets(cnt~.,data=train, nvmax = 37, method = "exhaustive")

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in =
## force.in, : 8 linear dependencies found

## Reordering variables and trying again:

leaps

## Subset selection object
## Call: regsubsets.formula(cnt ~ ., data = train, nvmax = 37, method =
"exhaustive")
## 35 Variables  (and intercept)
##                Forced in Forced out
## season_1           FALSE      FALSE
## season_2           FALSE      FALSE
## season_3           FALSE      FALSE
## yr_0               FALSE      FALSE
## mnth_1             FALSE      FALSE
## mnth_2             FALSE      FALSE
## mnth_3             FALSE      FALSE
## mnth_4             FALSE      FALSE
## mnth_5             FALSE      FALSE
## mnth_6             FALSE      FALSE
## mnth_7             FALSE      FALSE
## mnth_8             FALSE      FALSE
## mnth_9           FALSE      FALSE
## mnth_10          FALSE      FALSE
## mnth_11          FALSE      FALSE
## holiday_0        FALSE      FALSE
## weekday_0        FALSE      FALSE
## weekday_1        FALSE      FALSE
## weekday_2        FALSE      FALSE
## weekday_3        FALSE      FALSE
## weekday_4        FALSE      FALSE
## weekday_5        FALSE      FALSE
## weathersit_1     FALSE      FALSE
## weathersit_2     FALSE      FALSE
## temp             FALSE      FALSE
## hum              FALSE      FALSE
## windspeed        FALSE      FALSE
## season_4         FALSE      FALSE
## yr_1             FALSE      FALSE
## mnth_12          FALSE      FALSE
## holiday_1        FALSE      FALSE
## weekday_6        FALSE      FALSE
## workingday_0     FALSE      FALSE
## workingday_1     FALSE      FALSE
## weathersit_3     FALSE      FALSE
## 1 subsets of each size up to 27
## Selection Algorithm: exhaustive
```

```
reg.summary<-summary(leaps)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(cnt ~ ., data = train, nvmax = 37, method = 
"exhaustive")
## 35 Variables  (and intercept)
##              Forced in Forced out
## season_1         FALSE      FALSE
## season_2         FALSE      FALSE
## season_3         FALSE      FALSE
## yr_0             FALSE      FALSE
## mnth_1           FALSE      FALSE
## mnth_2           FALSE      FALSE
## mnth_3           FALSE      FALSE
## mnth_4           FALSE      FALSE
## mnth_5           FALSE      FALSE
## mnth_6           FALSE      FALSE
## mnth_7           FALSE      FALSE
## mnth_8           FALSE      FALSE
## mnth_9           FALSE      FALSE
## mnth_10          FALSE      FALSE
## mnth_11          FALSE      FALSE
## holiday_0        FALSE      FALSE
## weekday_0        FALSE      FALSE
## weekday_1        FALSE      FALSE
## weekday_2        FALSE      FALSE
## weekday_3        FALSE      FALSE
## weekday_4        FALSE      FALSE
## weekday_5        FALSE      FALSE
## weathersit_1     FALSE      FALSE
## weathersit_2     FALSE      FALSE
## temp             FALSE      FALSE
## hum              FALSE      FALSE
## windspeed        FALSE      FALSE
## season_4         FALSE      FALSE
## yr_1             FALSE      FALSE
## mnth_12          FALSE      FALSE
## holiday_1        FALSE      FALSE
## weekday_6        FALSE      FALSE
## workingday_0     FALSE      FALSE
## workingday_1     FALSE      FALSE
## weathersit_3     FALSE      FALSE
## 1 subsets of each size up to 27
## Selection Algorithm: exhaustive
##           season_1 season_2 season_3 season_4 yr_0 yr_1 mnth_1 mnth_2
mnth_3
## 1  ( 1 )  " "      " "      " "      " "      " "  " "  " "    " "    " "
## 2  ( 1 )  " "      " "      " "      " "      " "  "*"  " "    " "    " "
## 3  ( 1 )  "*"      " "      " "      " "      " "  "*"  " "    " "    " "
## 4  ( 1 )  "*"      " "      " "      " "      " "  "*"  " "    " "    " "
## 5  ( 1 )  "*"      " "      " "      " "      " "  "*"  " "    " "    " "
## 6  ( 1 )  "*"      " "      " "      " "      " "  "*"  " "    " "    " "
## 7  ( 1 )  "*"      " "      " "      "*"      " "  "*"  " "    " "    " "
## 8  ( 1 )  "*"      " "      " "      "*"      " "  "*"  " "    " "    " "
## 9  ( 1 )  "*"      " "      " "      " "      "*"  " "  " "    " "    " "
## 10  ( 1 ) "*"      " "      "*"      " "      "*"  " "  " "    " "    " "
## 11  ( 1 ) "*"      " "      "*"      " "      "*"  " "  " "    " "    " "
## 12  ( 1 ) "*"      " "      "*"      " "      "*"  " "  " "    " "    " "
## 13  ( 1 ) "*"      " "      "*"      " "      "*"  " "  " "    " "    " "
## 14  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    " "    "*"
## 15  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    " "    "*"
## 16  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    " "    "*"
## 17  ( 1 ) "*"      " "      " "      "*"      " "  "*"  "*"    "*"    " "
## 18  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    " "    "*"
## 19  ( 1 ) "*"      " "      " "      "*"      " "  "*"  "*"    "*"    " "
## 20  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    "*"    "*"
## 21  ( 1 ) "*"      " "      " "      "*"      " "  "*"  " "    "*"    "*"
## 22  ( 1 ) "*"      " "      " "      "*"      "*"  " "  "*"    "*"    " "
## 23  ( 1 ) "*"      " "      " "      "*"      " "  "*"  "*"    " "    "*"
## 24  ( 1 ) " "      "*"      "*"      "*"      "*"  " "  "*"    "*"    " "
## 25  ( 1 ) "*"      "*"      "*"      " "      "*"  " "  "*"    "*"    " "
## 26  ( 1 ) "*"      " "      "*"      "*"      " "  "*"  "*"    "*"    "*"
## 27  ( 1 ) "*"      "*"      "*"      " "      "*"  " "  "*"    "*"    "*"
##           mnth_4 mnth_5 mnth_6 mnth_7 mnth_8 mnth_9 mnth_10 mnth_11
mnth_12
## 1  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 2  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 3  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 4  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 5  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 6  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 7  ( 1 )  " "    " "    " "    " "    " "    " "    " "     " "     " "
## 8  ( 1 )  " "    " "    " "    "*"    " "    " "    " "     " "     " "
## 9  ( 1 )  " "    " "    " "    " "    " "    "*"    "*"     " "     " "
## 10  ( 1 ) " "    " "    " "    " "    " "    "*"    "*"     " "     " "
## 11  ( 1 ) " "    " "    " "    " "    " "    "*"    "*"     " "     " "
## 12  ( 1 ) " "    " "    " "    " "    " "    "*"    "*"     " "     " "
## 13  ( 1 ) " "    " "    " "    "*"    " "    "*"    "*"     " "     " "
## 14  ( 1 ) " "    " "    " "    "*"    " "    "*"    "*"     " "     " "
## 15  ( 1 ) " "    "*"    " "    "*"    " "    "*"    "*"     " "     " "
## 16  ( 1 ) " "    "*"    " "    "*"    " "    "*"    "*"     " "     " "
## 17  ( 1 ) " "    " "    " "    "*"    " "    "*"    " "     "*"     "*"
## 18  ( 1 ) "*"    "*"    "*"    " "    "*"    "*"    "*"     " "     " "
## 19  ( 1 ) " "    "*"    " "    "*"    " "    "*"    " "     "*"     "*"
## 20  ( 1 ) "*"    "*"    "*"    " "    "*"    "*"    "*"     " "     " "
## 21  ( 1 ) "*"    "*"    "*"    " "    "*"    "*"    "*"     " "     " "
## 22  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    " "     "*"     "*"
## 23  ( 1 ) "*"    "*"    "*"    " "    "*"    "*"    "*"     " "     "*"
## 24  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    " "     "*"     "*"
## 25  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    " "     "*"     "*"
## 26  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    " "     "*"     "*"
## 27  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"     "*"     " "
##           holiday_0 holiday_1 weekday_0 weekday_1 weekday_2 weekday_3
weekday_4
## 1  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 2  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 3  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 4  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 5  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 6  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 7  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 8  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 9  ( 1 )  " "       " "       " "       " "       " "       " "       " "
## 10  ( 1 ) " "       " "       " "       " "       " "       " "       " "
## 11  ( 1 ) " "       " "       " "       " "       " "       " "       " "
## 12  ( 1 ) " "       "*"       "*"       " "       " "       " "       " "
## 13  ( 1 ) " "       "*"       "*"       " "       " "       " "       " "
## 14  ( 1 ) " "       "*"       "*"       " "       " "       " "       " "
## 15  ( 1 ) "*"       " "       "*"       " "       " "       " "       " "
## 16  ( 1 ) " "       "*"       "*"       " "       " "       " "       " "
## 17  ( 1 ) " "       "*"       "*"       " "       " "       " "       " "
## 18  ( 1 ) "*"       " "       "*"       " "       " "       " "       " "
## 19  ( 1 ) " "       "*"       "*"       " "       " "       " "       "*"
## 20  ( 1 ) "*"       " "       "*"       " "       " "       " "       "*"
## 21  ( 1 ) " "       "*"       "*"       "*"       " "       " "       "*"
## 22  ( 1 ) "*"       " "       "*"       "*"       " "       " "       "*"
## 23  ( 1 ) "*"       " "       "*"       "*"       " "       "*"       "*"
## 24  ( 1 ) "*"       " "       "*"       "*"       " "       "*"       "*"
## 25  ( 1 ) " "       "*"       " "       "*"       "*"       " "       "*"
## 26  ( 1 ) "*"       " "       "*"       "*"       "*"       "*"       " "
## 27  ( 1 ) "*"       " "       "*"       "*"       "*"       "*"       "*"
##           weekday_5 weekday_6 workingday_0 workingday_1 weathersit_1
## 1  ( 1 )  " "       " "       " "          " "          " "
## 2  ( 1 )  " "       " "       " "          " "          " "
## 3  ( 1 )  " "       " "       " "          " "          " "
## 4  ( 1 )  " "       " "       " "          " "          "*"
## 5  ( 1 )  " "       " "       " "          " "          "*"
## 6  ( 1 )  " "       " "       " "          " "          "*"
## 7  ( 1 )  " "       " "       " "          " "          " "
## 8  ( 1 )  " "       " "       " "          " "          " "
## 9  ( 1 )  " "       " "       " "          " "          " "
## 10  ( 1 ) " "       " "       " "          " "          "*"
## 11  ( 1 ) " "       " "       " "          "*"          "*"
## 12  ( 1 ) " "       " "       " "          " "          "*"
## 13  ( 1 ) " "       " "       " "          " "          "*"
## 14  ( 1 ) " "       " "       " "          " "          "*"
## 15  ( 1 ) " "       " "       " "          " "          " "
## 16  ( 1 ) "*"       " "       " "          " "          "*"
## 17  ( 1 ) "*"       " "       " "          " "          "*"
## 18  ( 1 ) "*"       " "       " "          " "          " "
## 19  ( 1 ) "*"       " "       " "          " "          "*"
## 20  ( 1 ) "*"       " "       " "          " "          "*"
## 21  ( 1 ) "*"       " "       " "          " "          " "
## 22  ( 1 ) "*"       " "       " "          " "          "*"
## 23  ( 1 ) "*"       " "       " "          " "          "*"
## 24  ( 1 ) "*"       " "       " "          " "          "*"
## 25  ( 1 ) "*"       "*"       "*"          " "          "*"
## 26  ( 1 ) "*"       "*"       " "          " "          "*"
## 27  ( 1 ) "*"       " "       " "          " "          " "
##           weathersit_2 weathersit_3 temp hum windspeed
## 1  ( 1 )  " "          " "          "*"  " " " "
## 2  ( 1 )  " "          " "          "*"  " " " "
## 3  ( 1 )  " "          " "          "*"  " " " "
## 4  ( 1 )  " "          " "          "*"  " " " "
## 5  ( 1 )  "*"          " "          "*"  " " " "
## 6  ( 1 )  "*"          " "          "*"  " " "*"
## 7  ( 1 )  " "          "*"          "*"  "*" "*"
## 8  ( 1 )  " "          "*"          "*"  "*" "*"
## 9  ( 1 )  "*"          "*"          "*"  "*" "*"
## 10  ( 1 ) "*"          " "          "*"  "*" "*"
## 11  ( 1 ) "*"          " "          "*"  "*" "*"
## 12  ( 1 ) "*"          " "          "*"  "*" "*"
## 13  ( 1 ) " "          "*"          "*"  "*" "*"
## 14  ( 1 ) "*"          " "          "*"  "*" "*"
## 15  ( 1 ) "*"          "*"          "*"  "*" "*"
## 16  ( 1 ) " "          "*"          "*"  "*" "*"
## 17  ( 1 ) " "          "*"          "*"  "*" "*"
## 18  ( 1 ) "*"          "*"          "*"  "*" "*"
## 19  ( 1 ) " "          "*"          "*"  "*" "*"
## 20  ( 1 ) "*"          " "          "*"  "*" "*"
## 21  ( 1 ) "*"          "*"          "*"  "*" "*"
## 22  ( 1 ) "*"          " "          "*"  "*" "*"
## 23  ( 1 ) "*"          " "          "*"  "*" "*"
## 24  ( 1 ) " "          "*"          "*"  "*" "*"
## 25  ( 1 ) " "          "*"          "*"  "*" "*"
## 26  ( 1 ) " "          "*"          "*"  "*" "*"
## 27  ( 1 ) "*"          "*"          "*"  "*" "*"
```

```
mat=model.matrix(cnt~., data = test)
val.error=rep(NA,27)
for(p in 1:27){
  coefp <- coef(leaps,id=p) # Coefficients for selected variables
  pred <- mat[,names(coefp)]%*%coefp # multiply X matrix by coefficients
  val.error[p] <- mean((test$cnt-pred)^2) # mean squared error
}
val.error
```

```
##  [1] 3955253.8 3859547.0 2843884.2 2746426.8 2738147.7 2435119.1 1391745.6
##  [8] 1318152.8 2413460.5 2232049.0 1105265.7 2234887.8 2201139.6 1173545.5
## [15] 1233054.2 1121098.8 1071986.2 1210368.8  870791.0  852793.5  836412.8
## [22]  847482.9  819365.2  864231.0 1711801.4  907455.4 1804184.9
```

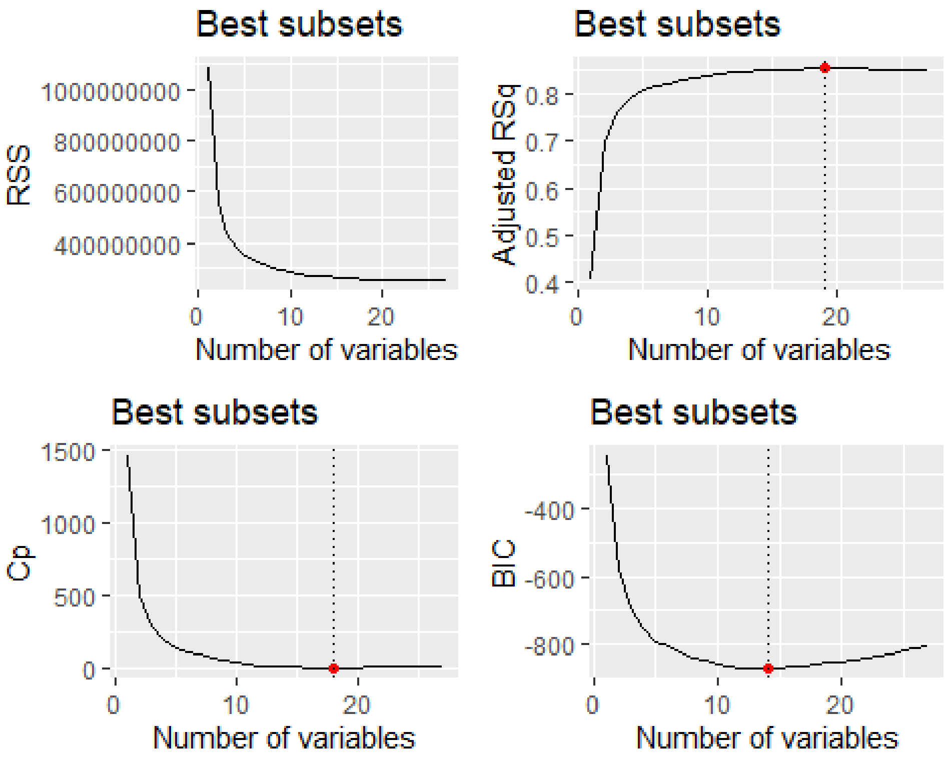
```
which.min(val.error)
```

```
## [1] 23
```

```
#min error was found to be associated with 23 variables model

round(coef(leaps, 23),3)
```

```
##  (Intercept)     season_1         yr_0       mnth_2       mnth_3
mnth_5
##     3838.194    -1401.710    -2026.006       -3.445      193.404      -
59.326
##       mnth_6       mnth_7       mnth_8      mnth_10      mnth_11
holiday_0
##     -224.473     -817.155     -405.609      602.280      116.617
511.499
##    weekday_1    weekday_2    weekday_4    weekday_5 weathersit_2
temp
##     -116.823      -74.203       52.395      148.794     -506.733
5077.039
##          hum      mnth_12    holiday_1 workingday_0 workingday_1
weathersit_3
```

```
##    -1021.458        3.172        0.000     -206.551        0.000    -
1996.251

best.plot <- function(varName, varLabel, minmax=" ") {
  gg <- ggplot(data.frame(varName), aes(x=seq_along(varName), y=varName)) +
    geom_line() +
    labs(x="Number of variables"
         , y=varLabel, title="Best subsets")
  if (minmax=="min") {
    gg <- gg + geom_point(aes(x=which.min(varName), y=min(varName)),
                          color="red") +
    geom_vline(aes(xintercept=which.min(varName)),linetype="dotted")
  }
  if (minmax=="max") {
    gg <- gg + geom_point(aes(x=which.max(varName), y=max(varName)),
                          color="red") +
      geom_vline(aes(xintercept=which.max(varName)), linetype="dotted")
  }
  return(gg)
}
d <- with(reg.summary, data.frame(rss,adjr2,cp,bic))
grid.arrange(best.plot(d$rss, "RSS"),
             best.plot(d$adjr2, "Adjusted RSq"
                       , "max"),
             best.plot(d$cp, "Cp", minmax="min"),
             best.plot(d$bic, "BIC", minmax="min"),
             ncol=2)
```



```
#R-squared train data:
reg.summary$adjr2[17]

## [1] 0.8521662

reg.summary$adjr2[18]

## [1] 0.8529187

reg.summary$adjr2[19]

## [1] 0.8529747

#Rmse train data:
round(sqrt(reg.summary$rss[17]/nrow(train)),2)

## [1] 723.34

round(sqrt(reg.summary$rss[18]/nrow(train)),2)

## [1] 720.75

round(sqrt(reg.summary$rss[19]/nrow(train)),2)

## [1] 719.86

#looking at the results from Best Subsets graphs, model with 19 variables is
selected:

model.f1<-lm(cnt~season_1+season_4+yr_1+ mnth_1+mnth_2+mnth_5+mnth_7+mnth_9+
mnth_11+ mnth_12+ holiday_1
+weekday_0+weekday_5+weekday_4+weathersit_1+weathersit_3 +temp +hum+
windspeed ,train )
summary(model.f1)

## 
## Call:
## lm(formula = cnt ~ season_1 + season_4 + yr_1 + mnth_1 + mnth_2 + 
##     mnth_5 + mnth_7 + mnth_9 + mnth_11 + mnth_12 + holiday_1 + 
##     weekday_0 + weekday_5 + weekday_4 + weathersit_1 + weathersit_3 + 
##     temp + hum + windspeed, data = train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3989.5  -338.2    63.8   450.0  2111.7 
## 
## Coefficients:
##                Estimate Std. Error t value             Pr(>|t|)    
## (Intercept)     3070.10     328.66   9.341 < 0.0000000000000002 ***
## season_1        -820.16     157.16  -5.219        0.000000268516 ***
## season_4         751.14     115.63   6.496        0.000000000207 ***
## yr_1            2030.94      66.98  30.321 < 0.0000000000000002 ***
## mnth_1          -689.38     182.98  -3.768             0.000185 ***
## mnth_2        -520.22     174.30  -2.985             0.002983 **
## mnth_5         242.97     134.48   1.807             0.071418 .
## mnth_7        -484.74     144.33  -3.359             0.000846 ***
## mnth_9         404.41     124.20   3.256             0.001210 **
## mnth_11       -571.86     160.31  -3.567             0.000397 ***
## mnth_12       -630.33     163.35  -3.859             0.000130 ***
## holiday_1     -646.21     189.69  -3.407             0.000713 ***
## weekday_0     -296.62      99.43  -2.983             0.002996 **
## weekday_5      208.52      99.04   2.105             0.035784 *
## weekday_4      109.06      97.65   1.117             0.264591
## weathersit_1   429.07      91.39   4.695       0.000003481461 ***
## weathersit_3 -1124.81     218.72  -5.143       0.000000395013 ***
## temp          4213.83     339.96  12.395 < 0.0000000000000002 ***
## hum          -1823.37     342.53  -5.323       0.000000156854 ***
## windspeed    -2827.86     460.23  -6.144       0.000000001687 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 734.7 on 480 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.853
## F-statistic: 153.4 on 19 and 480 DF,  p-value: < 0.00000000000000022
```

```
model.f2<-lm(cnt~season_1+season_3+yr_1+mnth_7+mnth_9+ mnth_10+ holiday_1
+weekday_0+weathersit_2+weathersit_3 +temp +hum+ windspeed ,train)
plot(model.f2)

#After Removing few of the insignificant variables based on their p values
and reducing the complexity of the model:
model.final<-lm(cnt~season_1+season_4+yr_1+ mnth_1+mnth_2+mnth_7+mnth_9+
mnth_11+ mnth_12+ holiday_1 +weekday_0+weathersit_1+weathersit_3 +temp +hum+
windspeed ,train)
sum.final=summary(model.final)
sum.final$adj.r.squared
```

```
## [1] 0.8513351
```

```
sum.final$sigma
```

```
## [1] 738.7939
```

```
summary(model.final)
```

```
##
## Call:
## lm(formula = cnt ~ season_1 + season_4 + yr_1 + mnth_1 + mnth_2 +
##     mnth_7 + mnth_9 + mnth_11 + mnth_12 + holiday_1 + weekday_0 +
##     weathersit_1 + weathersit_3 + temp + hum + windspeed, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4071.3  -328.0    47.2   465.8  2205.7
```

```
## 
## Coefficients:
##                 Estimate Std. Error t value             Pr(>|t|)    
## (Intercept)      3178.03     326.53   9.733 < 0.0000000000000002 ***
## season_1         -863.26     156.12  -5.530       0.000000052664 ***
## season_4          713.49     113.72   6.274       0.000000000782 ***
## yr_1             2036.06      67.26  30.273 < 0.0000000000000002 ***
## mnth_1           -693.51     183.91  -3.771             0.000183 ***
## mnth_2           -531.01     175.16  -3.032             0.002564 ** 
## mnth_7           -512.59     142.94  -3.586             0.000370 ***
## mnth_9            365.19     122.82   2.973             0.003092 ** 
## mnth_11          -588.68     161.07  -3.655             0.000286 ***
## mnth_12          -635.97     164.19  -3.873             0.000122 ***
## holiday_1        -661.02     190.29  -3.474             0.000559 ***
## weekday_0        -353.25      96.52  -3.660             0.000280 ***
## weathersit_1      414.13      91.52   4.525       0.000007604243 ***
## weathersit_3    -1193.80     218.60  -5.461       0.000000075804 ***
## temp             4200.23     341.82  12.288 < 0.0000000000000002 ***
## hum             -1807.72     340.26  -5.313       0.000000165125 ***
## windspeed       -2838.24     462.70  -6.134       0.000000001784 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 738.8 on 483 degrees of freedom
## Multiple R-squared:  0.8561, Adjusted R-squared:  0.8513 
## F-statistic: 179.6 on 16 and 483 DF,  p-value: < 0.00000000000000022
```

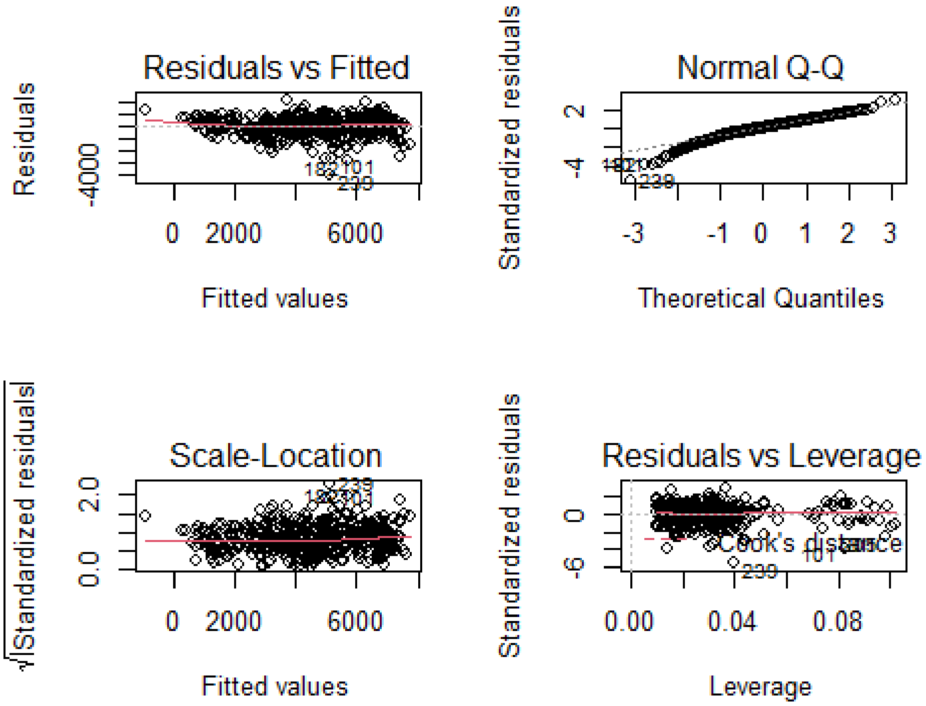
```
#Adj R squared and Rmse of training set after removal of variables:
sum.final$adj.r.squared
```

```
## [1] 0.8513351
```

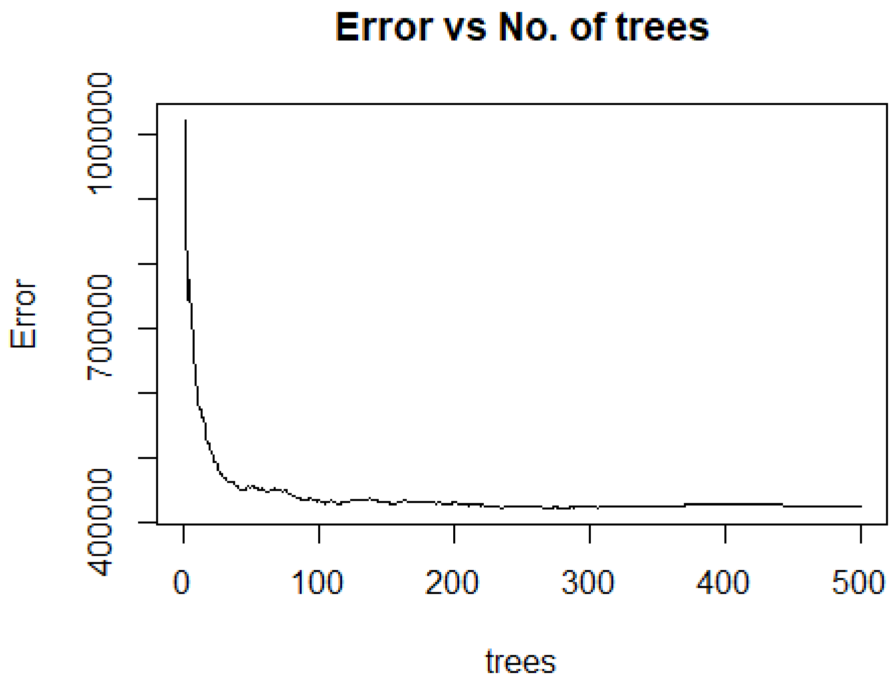
```
cat("training-set RMSE: ", sum.final$sigma)
```

```
## training-set RMSE:  738.7939
```

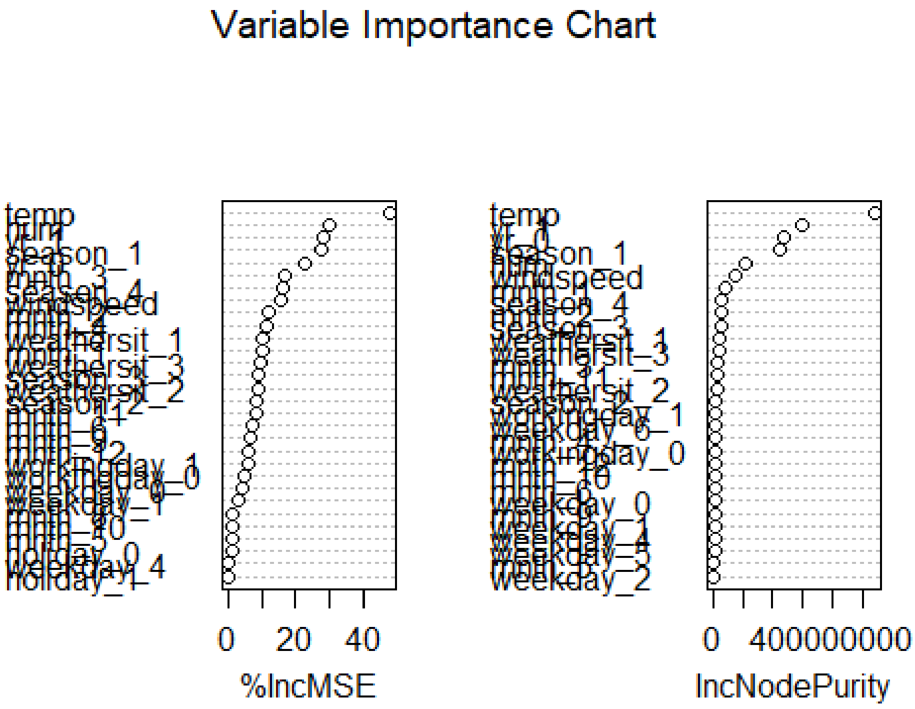
```
#prediction using the test data:
p<- predict.lm(model.final,newdata = test )
comp<-data.frame(test$cnt, p)
par(mfrow=c(1,1))
abline(lm(comp$test.cnt~comp$p), main ="predicted vs real values", xlab=
"predicted vals", ylab="real counts")
```



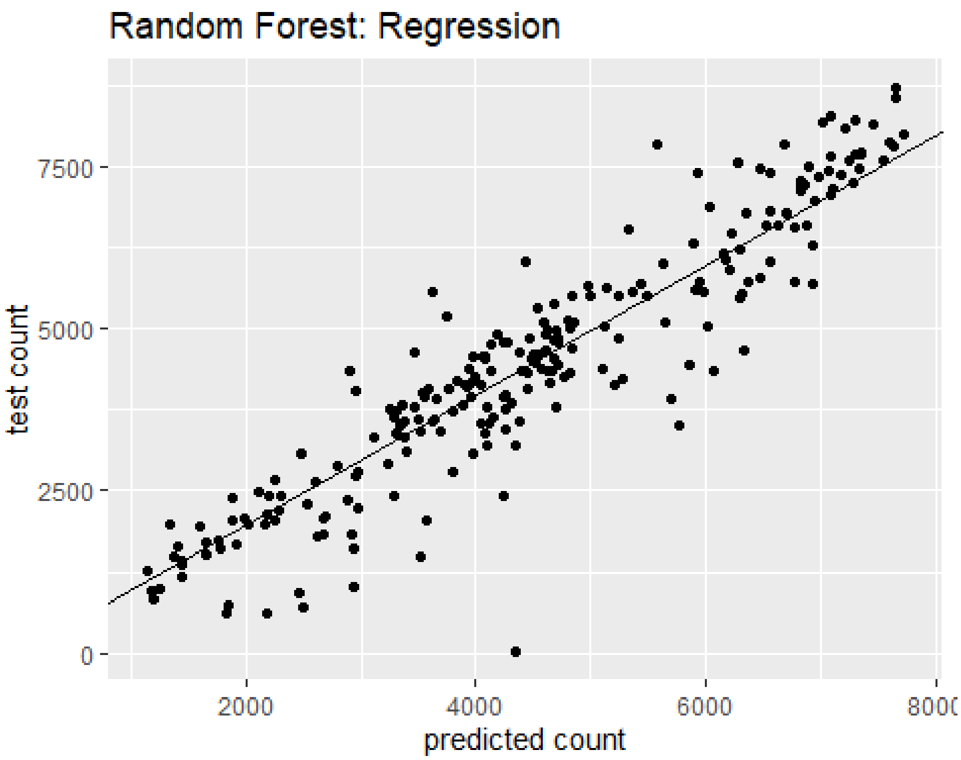
```
#Random Forest:
bag.count=randomForest(cnt~.,data=df1, subset = tr, importance=T)
print(bag.count)

##
## Call:
##  randomForest(formula = cnt ~ ., data = df1, importance = T, subset = tr)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 11
##
##           Mean of squared residuals: 422958
##                     % Var explained: 88.46

plot(bag.count, main="Error vs No. of trees")
```



```
#prediction using the test Data:
yhat.bag<- predict(bag.count, newdata = test)
tst <- test$cnt
mean((as.numeric(yhat.bag)-as.numeric(tst))^2)

## [1] 576599.7

rf<-data.frame(yhat.bag,tst)
#Plot of the important variables:
varImpPlot(bag.count, main = "Variable Importance Chart")
```



```
#Plotting test set count vs. predicted count:
ggplot(rf, aes(x=yhat.bag ,y=tst)) +
  geom_point() +
  geom_abline(slope=1,intercept=0) +
labs(x="predicted count",
     y="test count",
     title="Random Forest: Regression")
```


Random Forest: Regression
test count
7500
5000
2500
0
2000
4000
6000
8000
predicted count