STAT 306

Term Project

Bike Share Program
Bike Rental Prediction Project

Christy Choi 34109124

Shijie Xu 50929132

Weining Hu 45606134

Contents

Abstract

Background

Using historical usage patterns with weather data, a model was fitted to predict future bike rental demand in the Capital Bikeshare program in Washington D.C. The model is based on variables from the rental data: including season, hour, date, holiday, weather, temperature, season, etc.

In particular, based on explanatory variables as those stated above, we are interested in registered bike count (bike rental from registered users), casual bike count (casual users), and the total count (total bike rental from registered and casual users).

Methods

We considered two main types of models: first, a linear model, and second, a general linear model of the Poisson family (since our variable of interest is a counter).

Phase1: Data manipulation

The data we are using containing both categorical data and quantitative data. For the categorical variables: We first converting categorical variables to factor variables so that they can be used in statistical modeling and data visualization. We later give them numeric value and assigned the categorical data (for example, weather) with dummy variables before being put into the model. In addition, we noticed that for the datetime category of data, it includes year/month/day/hour/minute/second. We decided to stripe this data and create two additional variables(weekday and hour). The work described above is done with function featureEngineer(). The hour variable created is considered quantitative variable that ranges from 0 to 23 and was further collapsed into 6 categories (based on peak-hour patterns, as well as similarities in variance), to reduce the number of explanatory variables.

Phase2: Variable selection and creation of explanatory variable

Using package “leaps” to conduct exhaustive search, backward selection, forward selection and sequential selection. We selected a subset of the variables that would result in a model with the smallest CP and largest adjusted R values. After the model with the best variables is chosen, we check Cook’s distance to find abnormal data or outliers that largely affect the model and abandon these abnormal data.

Phase3: Variable transformation based on residual plots

Through residual plot analysis, we were able to make transformation on explanatory variables and check the assumption of normal distribution according to QQ-plot. When there are specific pattern found in residual plots, these patterns indicate we might need to adjust the model using logistic, quadratic, and square root terms. This helped to remove residual patterns, and resulted in model with better fit (linear models with smaller residual deviance and adjusted R closer to 1; general linear models with smaller AIC value).

Phase4: Cross validation

The final model was established after performing cross validation, by computing cross-validated root mean squares error of prediction (cvrmse). The smaller the cvrmse, the better the model. We also submit our result to kaggle competition and it ranked 466 among 2627 teams.

Conclusion

Based on our analysis,

Our best fit for registered rental is:

log(**registered**+1)=2.085+0.024(atemp)-0.850(hour1)-1.493(hour2)-1.741(hour3)-1.780(hour4)-0.721(hour5)+0.455(hour6)+1.423(hour7)+2.044(hour8)+1.668(hour9)+1.234(hour10)+1.334(hour11)+1.552(hour12)+1.505(hour13)+1.386(hour14)+1.464(hour15)+1.796(hour16)+2.252(hour17)+2.190(hour18)+1.900(hour19)+1.5936(hour20)+1.333(hour21)+1.088(hour22)+0.691(hour23)+0.5(year2012)-0.002(humidity)+0.161(season)-0.063(weather2)-0.546(weather3)-0.426(weather4)-0.002(windspeed)+0.236(workingday)

with an adjusted R-squared = 0.8493, residual standard error = 0.5482

And our best fit for casual rental is:

log(**casual**+1)=1.266+0.052(atemp)+0.029(weekday)-0.507(hour1)-0.811(hour2)-1.118(hour3)-1.267(hour4)-1.110(hour5)-0.413(hour6)+0.370(hour7)+0.977(hour8)+1.164(hour9)+1.350(hour10)+1.498(hour11)+1.554(hour12)+1.543(hour13)+1.538(hour14)+1.547(hour15)+1.583(hour16)+1.670(hour17)+1.477(hour18)+1.278(hour19)+1.047(hour20)+0.877(hour21)+0.699(hour22)+0.425(hour23)+0.265(year2012-0.081(holiday)-0.005(humidity)+0.099(season)+0.022(temp)-0.088(weather2)-0.639(weather3)-0.366(weather4)-0.0.002(windspeed)-0.549(workingday)

with an adjusted R-squared=0.8167, residual standard error=0.6434

Variable Information

Variables

Variable	Quantitative (Units)	Qualitative (Categories)	Description
datetime			
season		1= spring 2 = summer 3 = fall 4 = winter	seasons: spring, summer, fall, or winter
holiday		0 = not a holiday 1 = holiday	whether the day is considered a holiday
workingday		0 = not a working day 1 = working day	whether the day is neither a weekend nor holiday
weather		1: clear, few clouds, partly cloudy 2: mist + cloudy, mist + broken clouds, mist + few clouds, mist 3: light snow, light rain+thunderstorm+scattered clouds, light rain + scattered clouds 4: heavy rain + ice pallets + thunderstorm + mist, snow + fog	weather condition separated into 4 categories
temp	degrees Celsius		actual temperature
atemp	degrees Celsius		“feels like” temperature
humidity	%		relative humidity
windspeed	kts (knots)		Wind speed
casual	rentals		number of non-registered user rentals initiated per hour
registered	rentals		number of registered user rentals initiated per hour
count	rentals		Casual + registered per hour

Summary Statistics

	Min	1st Qu.	Median	Mean	3rd Qu.	Max	Variance
Temp	0.82	13.94	20.50	20.23	26.24	41.00	60.71
Atemp	0.76	16.66	24.24	23.66	31.06	45.46	71.82
Humidity	0.00	47.00	62.00	61.89	77.00	100.00	370.37
Windspeed	0.00	7.00	13.00	12.80	17.00	57.00	66.66
Casual	0.00	4.00	17.00	36.02	49.00	367.00	2496.05
Registered	0.0	36.0	118.0	155.6	222.0	886.0	22812.79
Count	1.0	42.0	145.0	191.6	284.0	977.0	32813.31

Frequency Tables

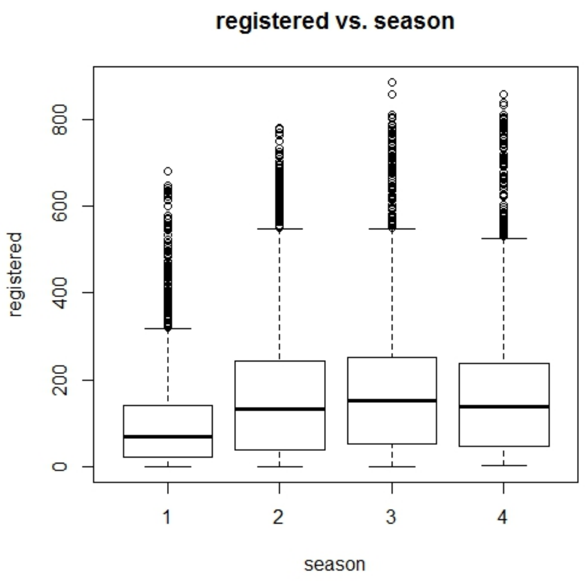
Weather	Total Count
1	7192
2	2834
3	859
4	1

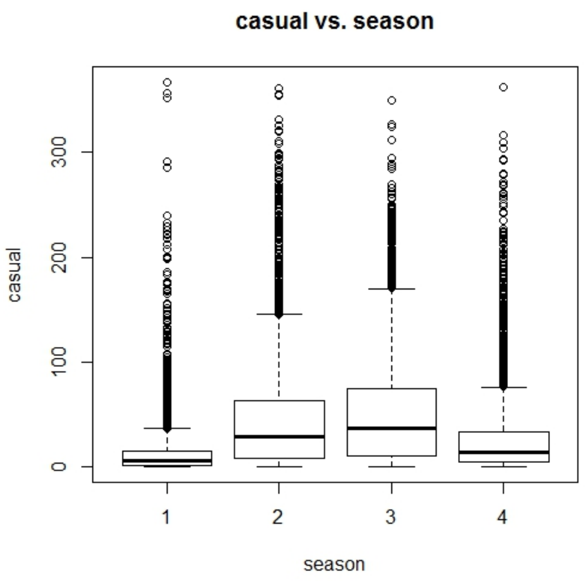
Season	Total Count
1	2686
2	2733
3	2733
4	2734

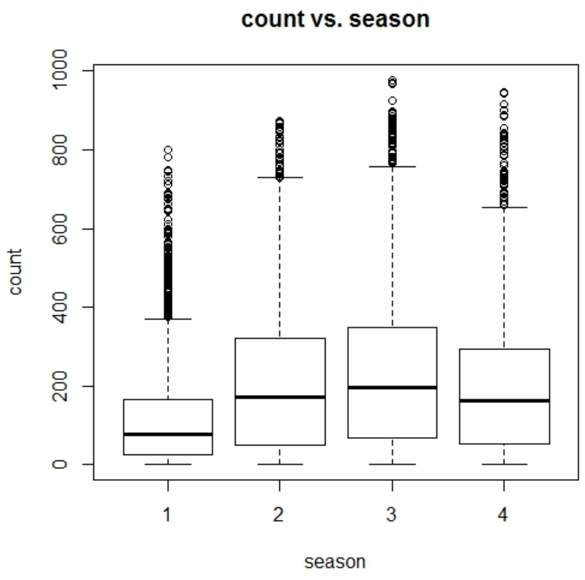
Holiday	Total Count
1	10575
2	311

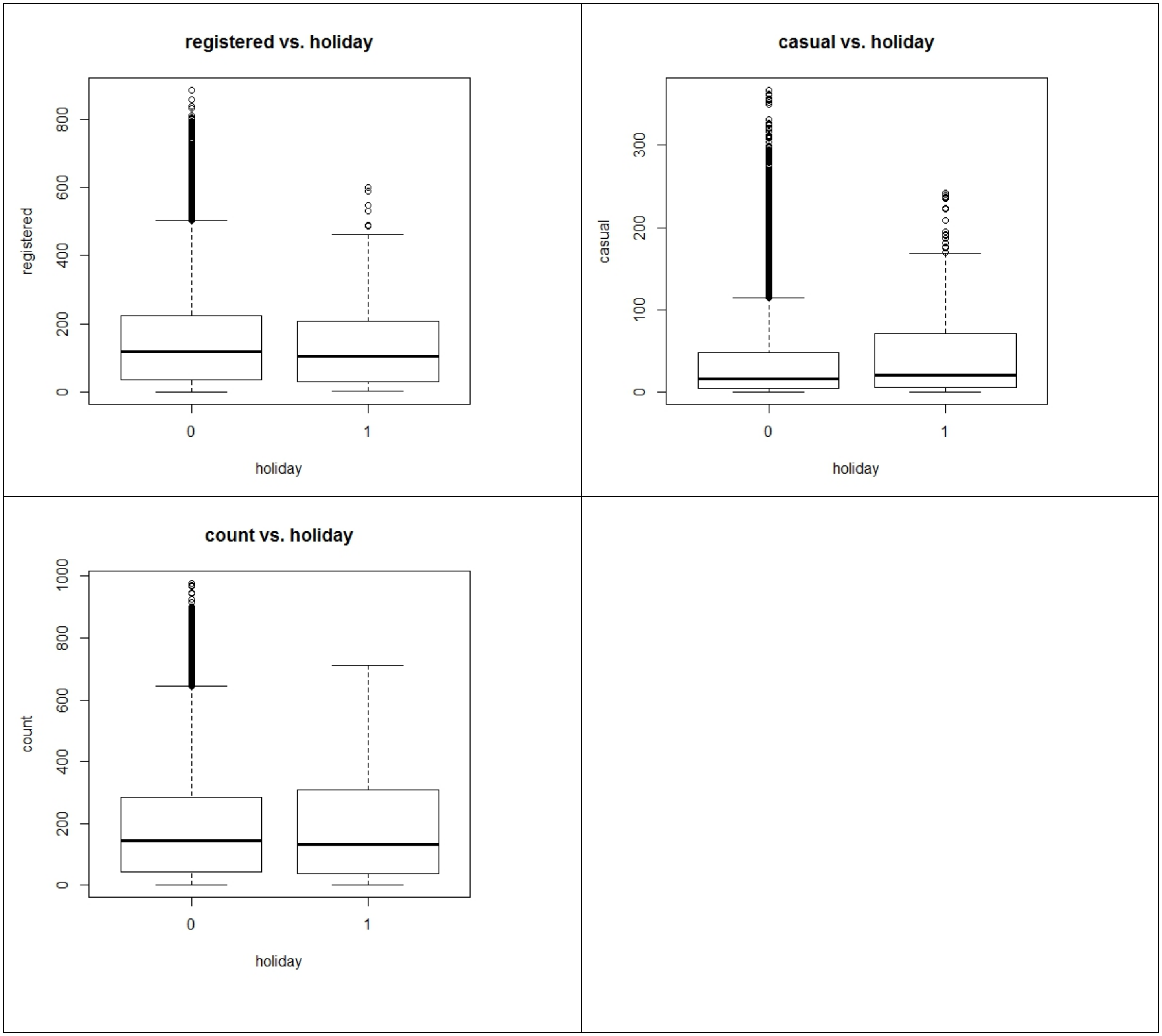
Working Day	Total Count
1	3474
2	7412

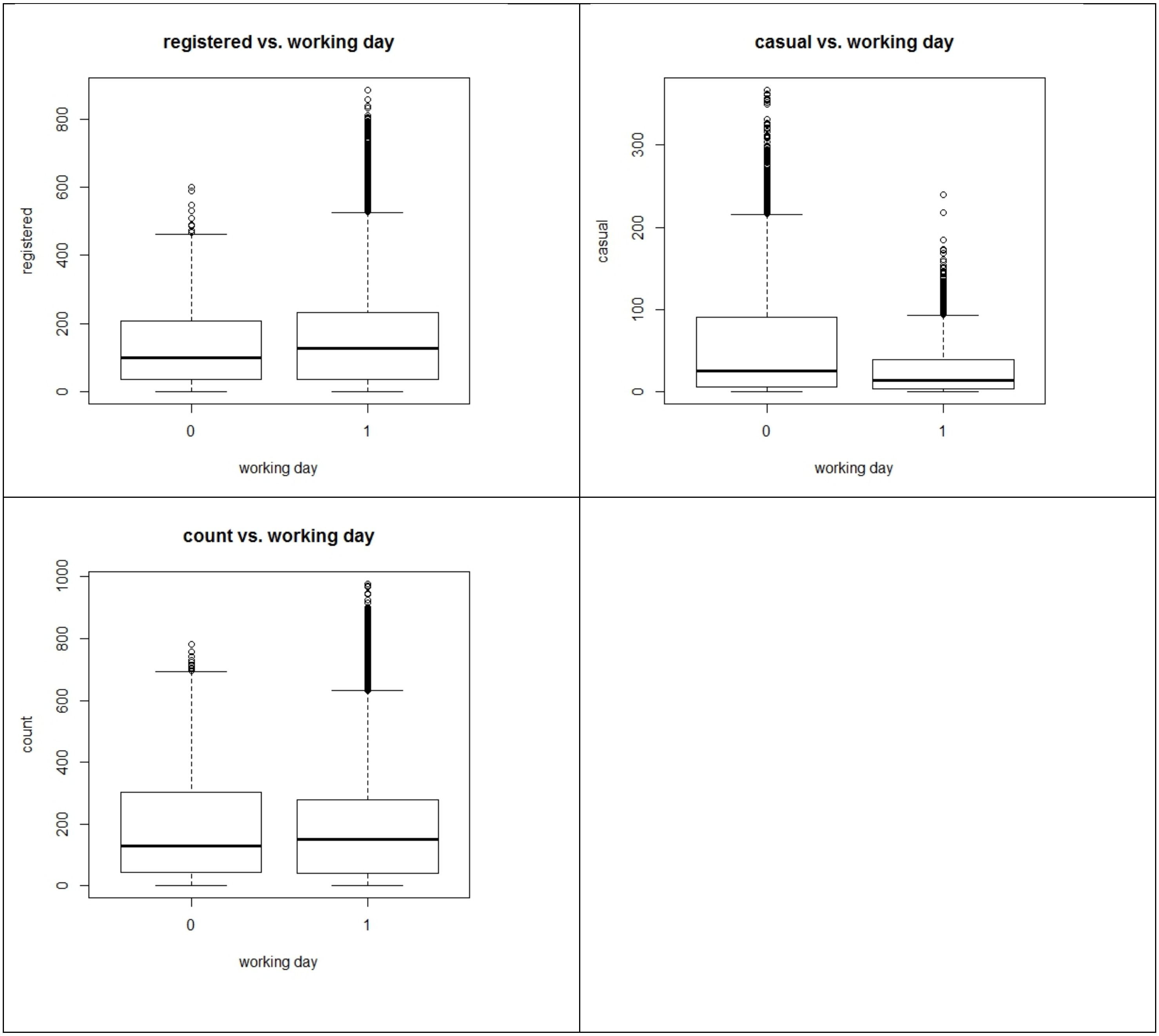
Plots


registered vs. season
registered
0
200
400
600
800
1
2
3
4
season


casual vs. season
casual
0
100
200
300
1
2
3
4
season


count vs. season
count
0
200
400
600
800
1000
1
2
3
4
season

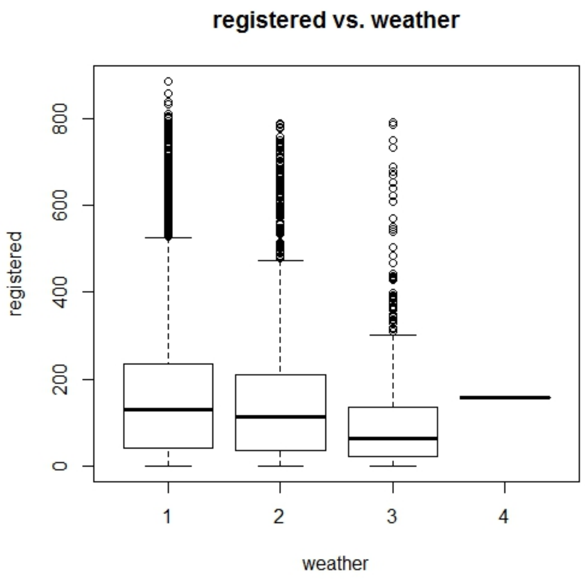

registered vs. holiday
registered
0
200
400
600
800
0
1
holiday
casual vs. holiday
casual
0
100
200
300
0
1
holiday
count vs. holiday
count
0
200
400
600
800
1000
0
1
holiday

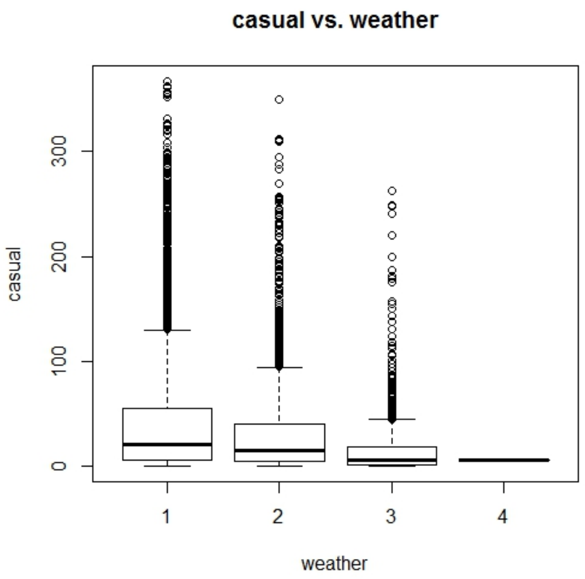


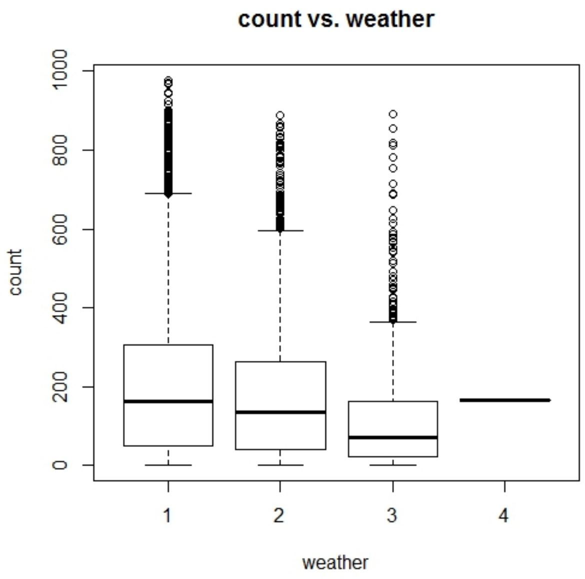
registered vs. working day

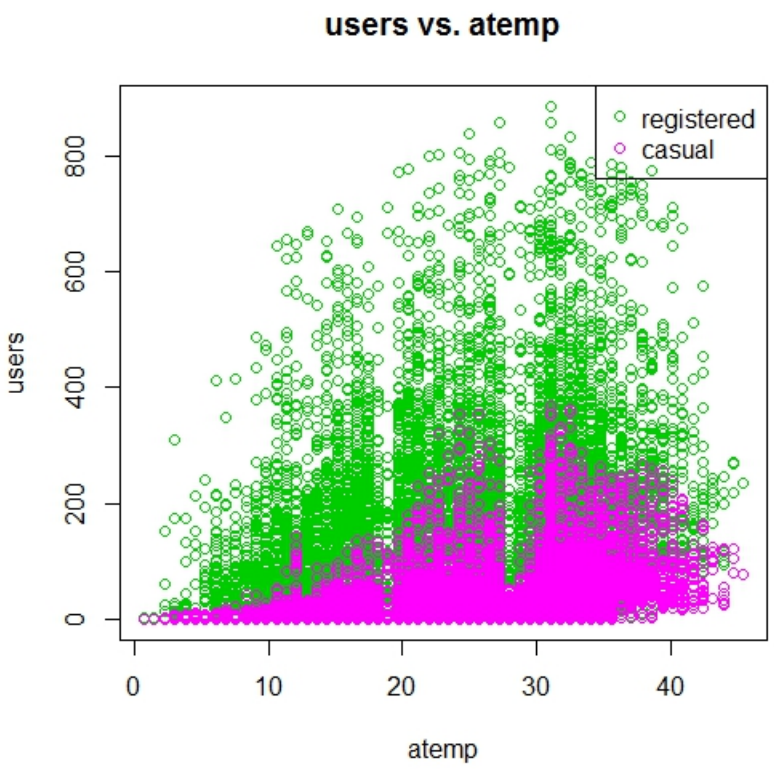
casual vs. working day

count vs. working day

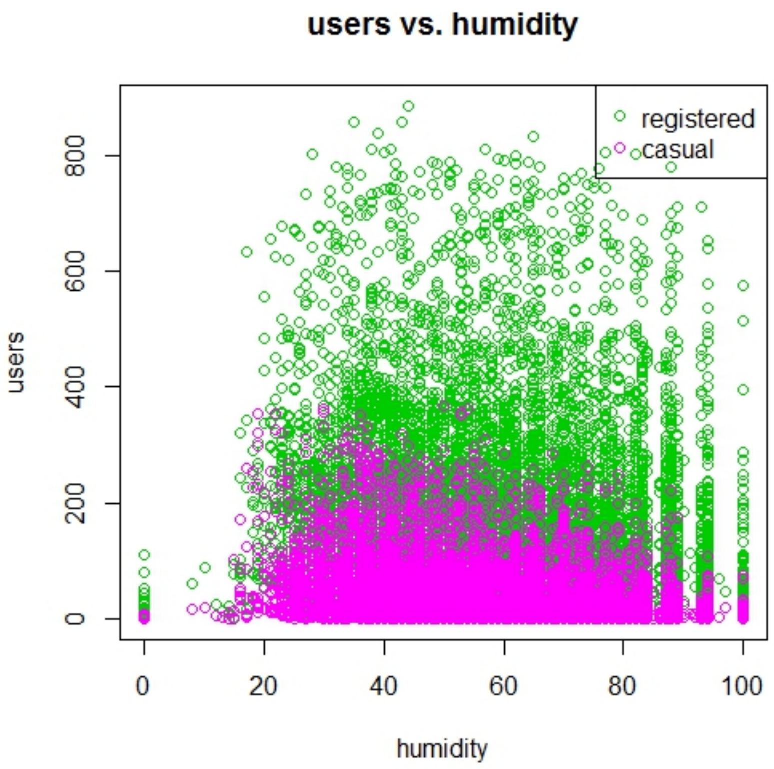

registered vs. weather
registered
0
200
400
600
800
1
2
3
4
weather


casual vs. weather
casual
0
100
200
300
1
2
3
4
weather

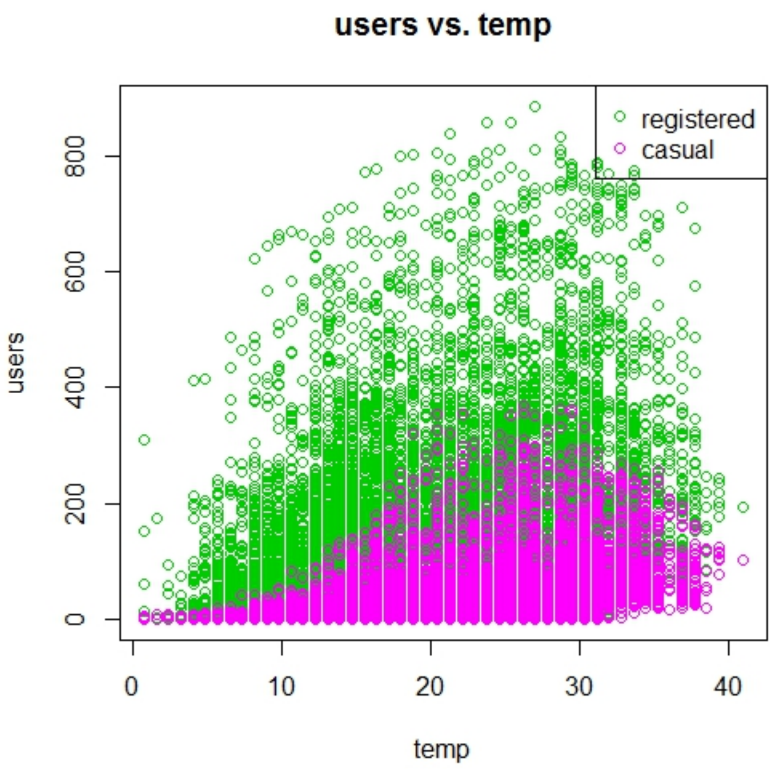

count vs. weather
count
0
200
400
600
800
1000
1
2
3
4
weather

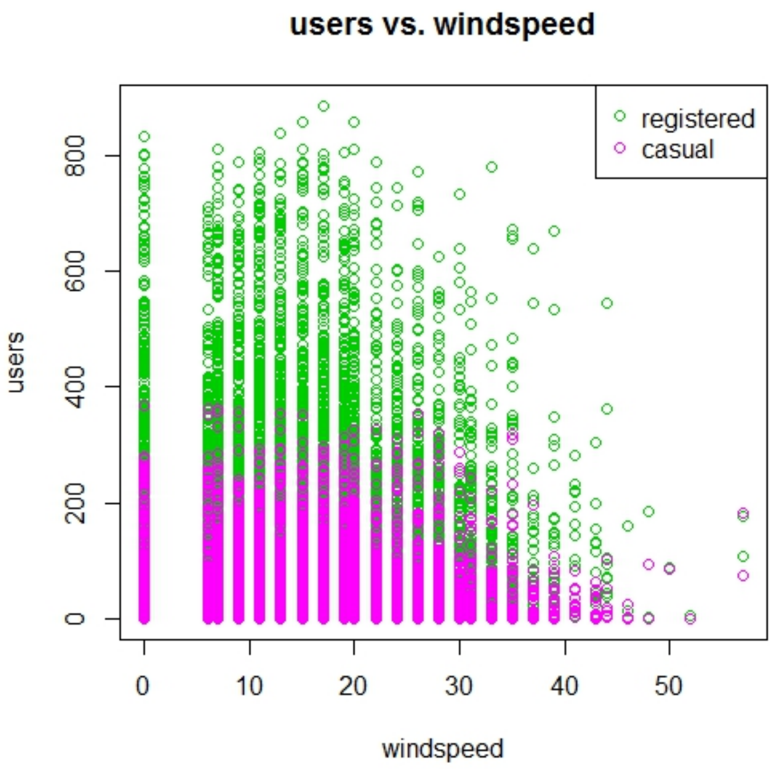


users vs. atemp



users vs. humidity

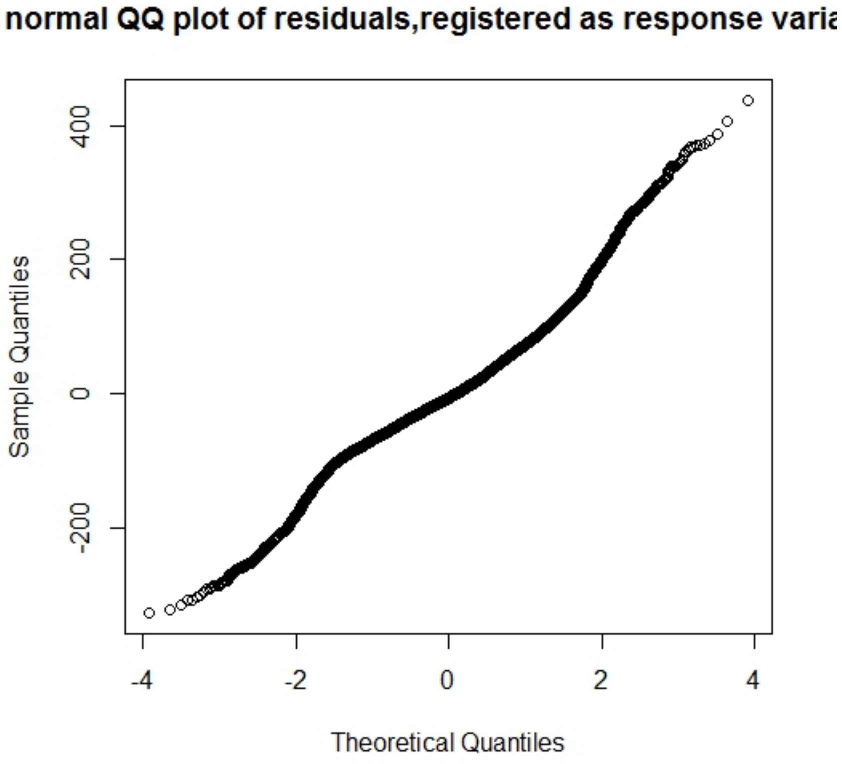

users vs. temp
registered
casual
users
0
200
400
600
800
0
10
20
30
40
temp


users vs. windspeed
registered
casual
users
0
200
400
600
800
0
10
20
30
40
50
windspeed

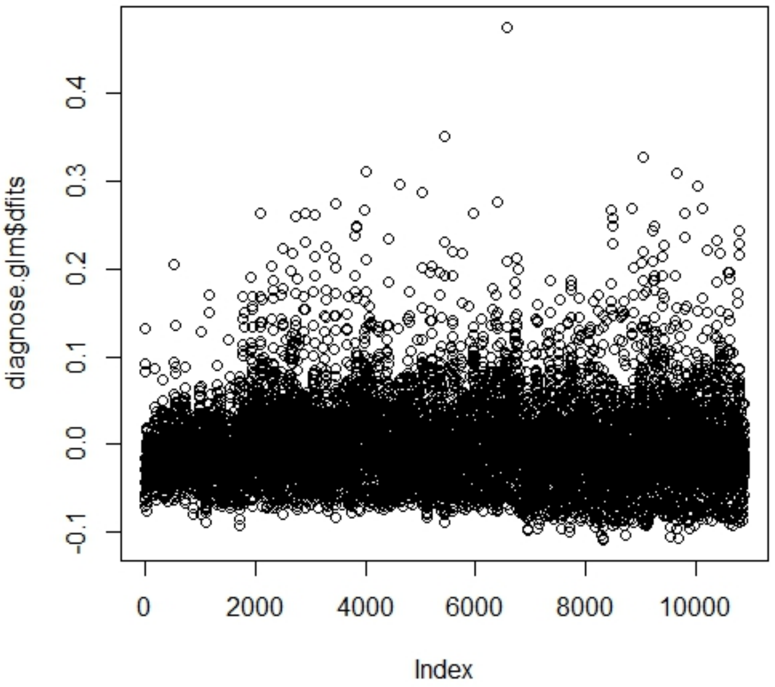
Analysis Outline

Based on exhaustive variable search, without grouping the hour variable, we have atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday as our best selection of variables (since this generates the smallest CP value and almost the largest adjusted R value), and with grouping of the hour variable, we have atemp+hourcat6+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+workingday as our best selection of variables (where hourcat6 is the grouped hour variable, and also this generates the variable selection with the smallest CP value and almost the largest adjusted R value).

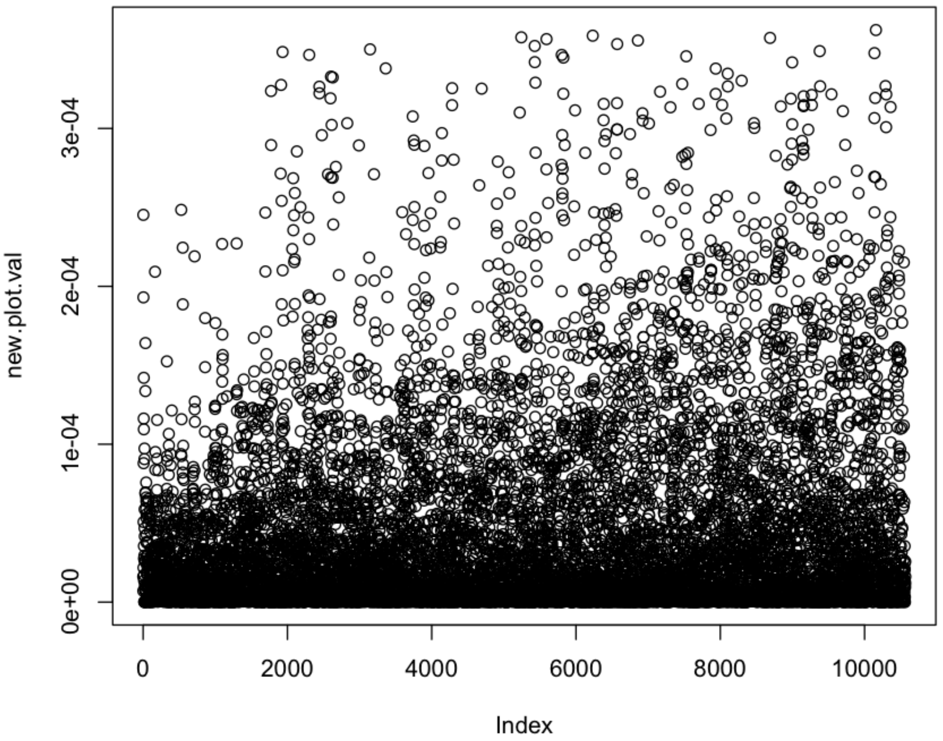
Below is the initial QQ plot for our registered fit (without regrouping hour categories).
The line deviates often from a straight line, and curves near both ends.



From plotting the dfits, we realized that there were outliers, that could possibly alter our fit.



We calculated Cook's distance and removed the outliers. The remaining data points had the following dfit spread.



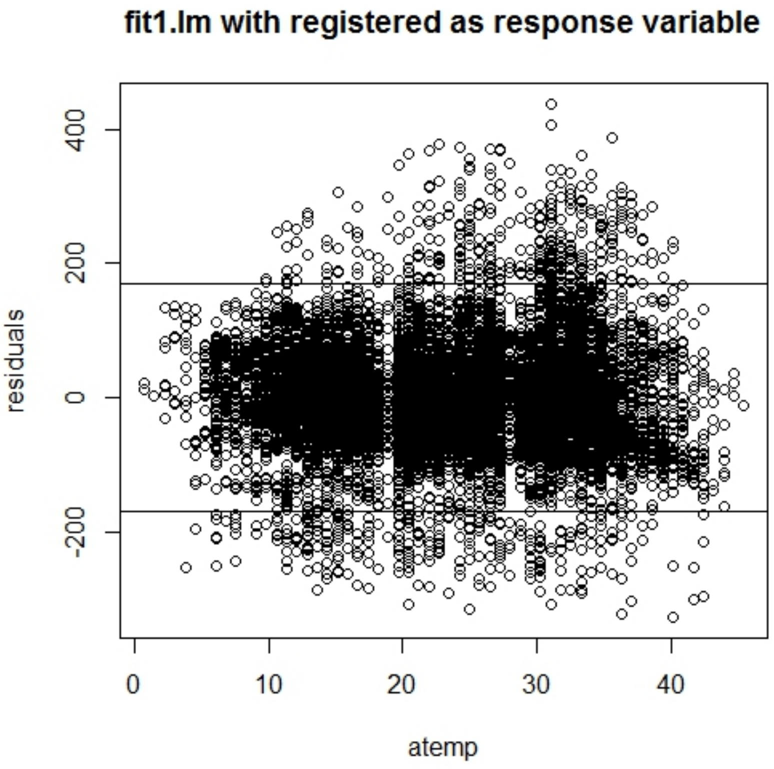
After removing outliers using Cook's distance, we considered the reduced set of data ("subtrain"). There are no longer any "odd" points that are far from the rest of the other datapoint.

We compared the fit of the models by comparing the adjusted R-squared (for linear models), and AIC (for general linear models)

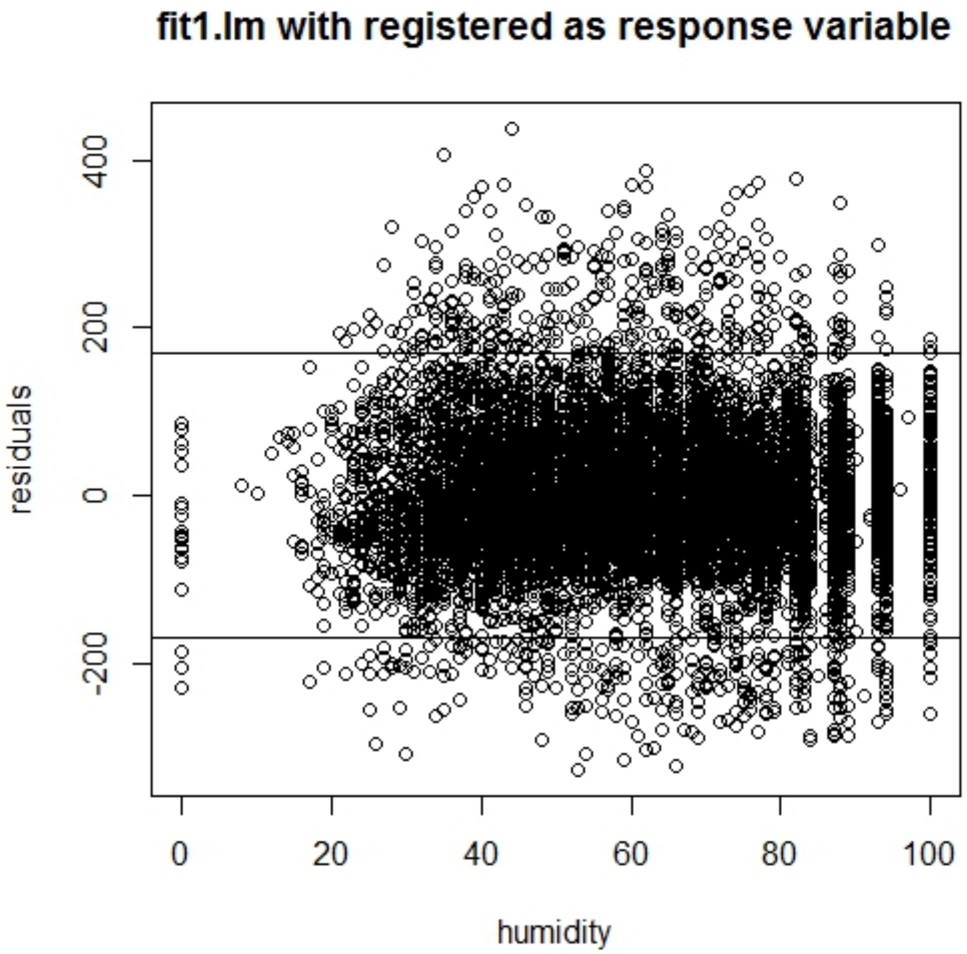
Without grouping of hour variable:
Summary of the linear model:
Multiple R-squared: 0.6878, Adjusted R-squared: 0.6868
Summary of the general linear model:
AIC: 348374

With grouping of hour variable:
Summary of the linear model:
Multiple R-squared: 0.5775, Adjusted R-squared: 0.577
Summary of the general linear model:
AIC: 477430

By analysing the behaviour of the residual plots of atemp and humidity, we decided to add quadratic term of atemp and natural logarithm of humidity to our variables.



As we can see from the atemp vs. residuals plot, the residuals have a certain pattern, i.e., larger at the middle of the range of atemp and smaller at the two ends. This pattern suggests us to add a quadratic term of atemp variable.
After the quadratic term of atemp variable is added, the AIC value of general linear model decreases to 467758 and the adjusted R-squared value of linear model increases to 0.5774.



As we can see from the humidity vs. residuals plot, the residuals have a certain pattern, i.e., the residuals increase as humidity increases. This pattern suggests us to add a natural logarithm term of humidity variable.
After the natural logarithm term of humidity variable is added, the AIC value of general linear model decreases to 463101 and the adjusted R-squared value of linear model increases to 0.5783.

This gives the following results:

Without grouping of hour variable:
Summary of the linear model:
Multiple R-squared: 0.6883, Adjusted R-squared: 0.6873
Summary of the general linear model:
AIC: 339378

With grouping of hour variables:
Summary of the linear model:
Multiple R-squared: 0.5788, Adjusted R-squared: 0.5783
Summary of the general linear model:
AIC: 463101

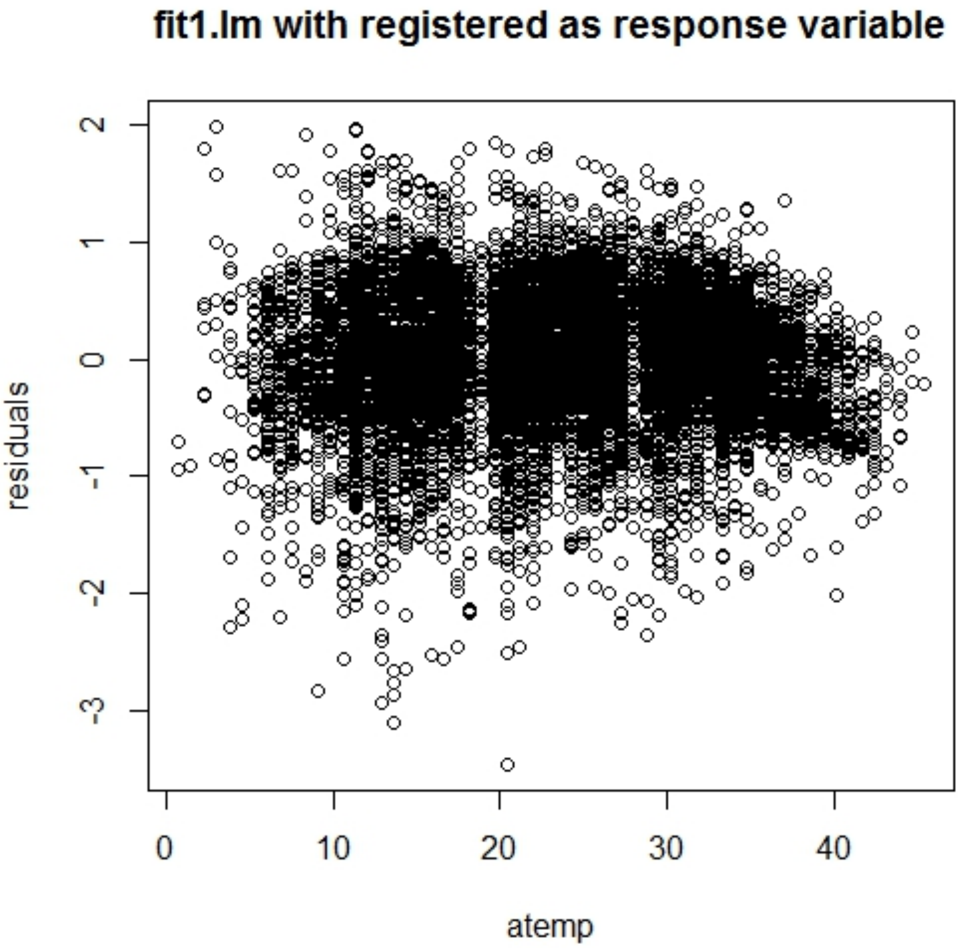
We realized that transforming the explanatory variables was not very effected in improving fit. It was also evident that many of the residual plots had the same patterns. Thus, we performed a transformation on the registered count instead.

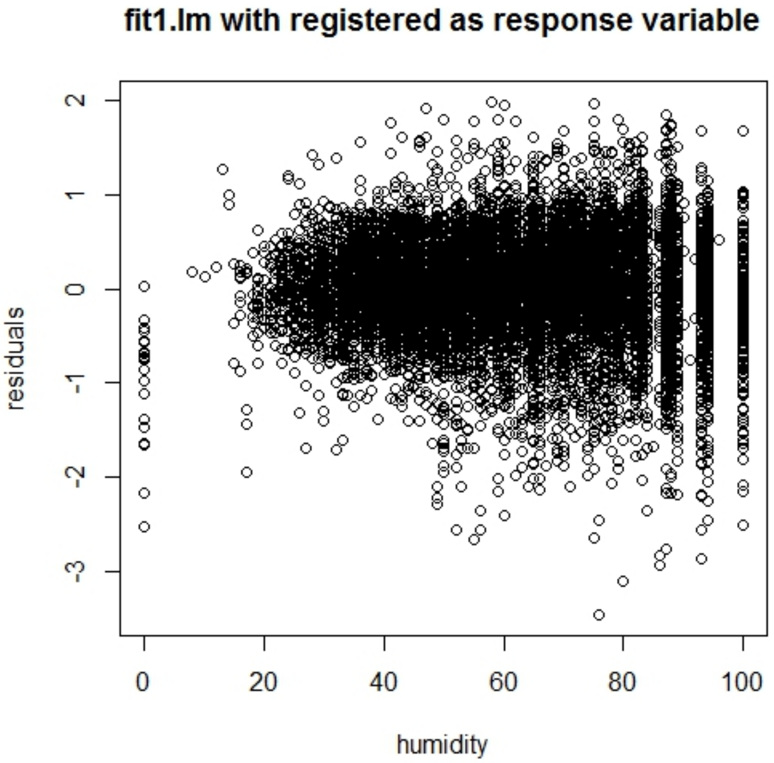
Since our general linear model assumed a Poisson family (response variable is a counter), we could not take the log(registered) as a response variable (it would no longer by nonnegative integers). We only did the log transformation for our linear model.

After comparing the adjusted R-squared of 3 fits (difference was the quadratic term for atemp, and whether or not hour was regrouped), our best fit came from the fit:

fit1.lm =
lm(I(log(registered+1))~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday,data=subtrain)
with an adjusted R squared value= 0.8493 (which is significantly higher than our previous models)

The new fit was also able to remove the patterns that were previously found in the residual plots for atemp and humidity.





In conclusion, the above “fit1.lm” was our best fit for registered user count of bike rentals.

We performed the same steps, but this time for the casual user count of bike rentals.
Through model selection, we were able to conclude the best model had 32 variables.

fit2.lm = lm(casual~atemp+weekday+hour+I(year == 2012)+holiday+humidity+season+temp+I(weather == 2)+ I(weather == 3)+windspeed+workingday,data = subtrain)
Residual standard error= 32.2
Adjusted R-squared=0.5843

Learning from our analysis of the registered user count, we decided to take the log of the response variable again. This resulted in the best fit:
fit2.lm = lm((log(casual+1))~atemp+weekday+hour+I(year == 2012)+holiday+humidity+season+temp+weather+windspeed+workingday,data = subtrain)
Residual standard error= 0.628
Adjusted R-squared=0.8233

By analyzing our residual plots, fit2.lm was able to remove majority of the pattern that was previously present, and also helped reduce heteroscedasticity.

The final step was to perform cross validation.
Special attention was given to the huge difference in rmse. Since variance of prediction in general linear models varies based on covariates, we cannot directly compare the rmse models between a linear and poisson family fit. (While we can compare between linear models, and between poisson family fit models separately, the comparison of two different types could be misleading).

Taking into consideration that rmse varies widely based on covariates, we turned to our adjusted R-squared and AIC values to give us our best fit.

We decided that the optimal models are:

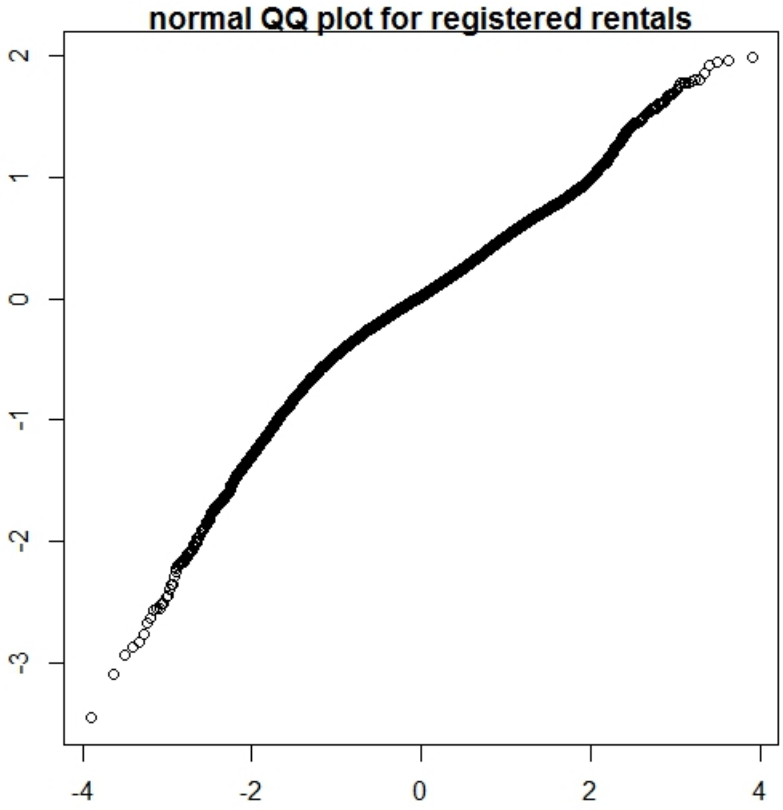
```
fit1.lm =
lm(I(log(registered+1))~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday,d
ata=subtrain)
```

```
fit2.lm = lm((log(casual+1))~atemp+weekday+hour+I(year ==
2012)+holiday+humidity+season+temp+weather+windspeed+workingday,data =
subtrain)
```

On the following pages are the R output that provides the coefficients for the 2 models.

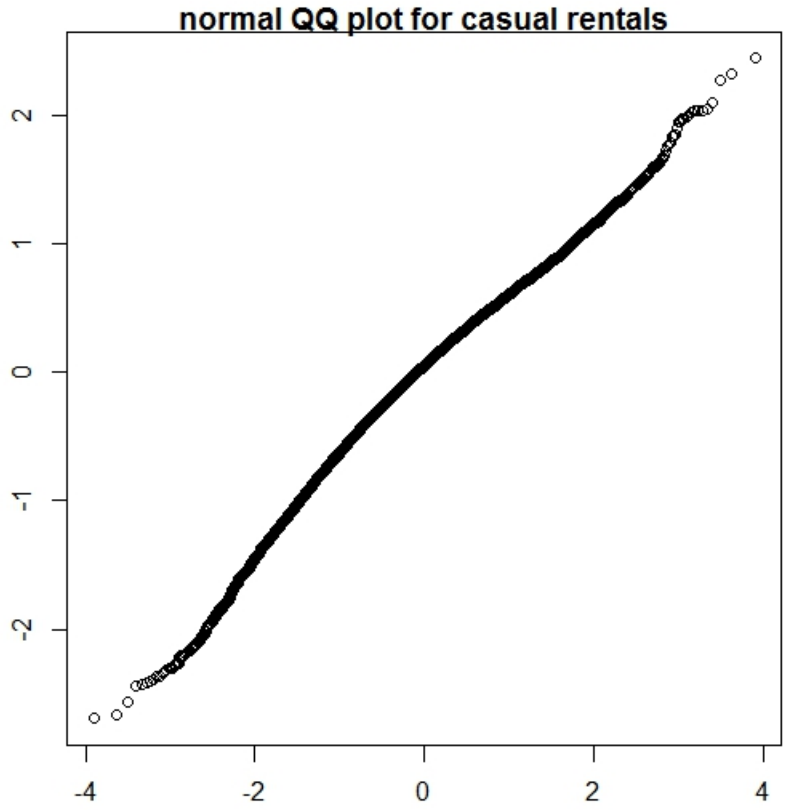
```
> summary(fit1.lm)

Call:
lm(formula = I(log(registered + 1)) ~ atemp + hour + I(year ==
    2012) + humidity + season + weather + windspeed + workingday,
    data = subtrain)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4562 -0.2815  0.0186  0.3401  1.9845

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.0854860  0.0453976  45.938  < 2e-16 ***
atemp                0.0235515  0.0006782  34.726  < 2e-16 ***
hour1               -0.8503390  0.0389245 -21.846  < 2e-16 ***
hour2               -1.4930459  0.0395305 -37.769  < 2e-16 ***
hour3               -1.7414991  0.0390292 -44.620  < 2e-16 ***
hour4               -1.7803247  0.0375256 -47.443  < 2e-16 ***
hour5               -0.7212081  0.0371091 -19.435  < 2e-16 ***
hour6                0.4549617  0.0370614  12.276  < 2e-16 ***
hour7                1.4226557  0.0370456  38.403  < 2e-16 ***
hour8                2.0436895  0.0370641  55.139  < 2e-16 ***
hour9                1.6676640  0.0370451  45.017  < 2e-16 ***
hour10               1.2339093  0.0372096  33.161  < 2e-16 ***
hour11               1.3341826  0.0374027  35.671  < 2e-16 ***
hour12               1.5522820  0.0375859  41.300  < 2e-16 ***
hour13               1.5048101  0.0378323  39.776  < 2e-16 ***
hour14               1.3857016  0.0380201  36.447  < 2e-16 ***
hour15               1.4646092  0.0380085  38.534  < 2e-16 ***
hour16               1.7955228  0.0378579  47.428  < 2e-16 ***
hour17               2.2521340  0.0377812  59.610  < 2e-16 ***
hour18               2.1904503  0.0376495  58.180  < 2e-16 ***
hour19               1.8999133  0.0372875  50.953  < 2e-16 ***
hour20               1.5936237  0.0371480  42.899  < 2e-16 ***
hour21               1.3331016  0.0370807  35.951  < 2e-16 ***
hour22               1.0878636  0.0370410  29.369  < 2e-16 ***
hour23               0.6905982  0.0370036  18.663  < 2e-16 ***
I(year == 2012)TRUE  0.5000953  0.0107609  46.474  < 2e-16 ***
humidity            -0.0019803  0.0003759  -5.268 1.40e-07 ***
season               0.1610211  0.0050989  31.579  < 2e-16 ***
weather2            -0.0631702  0.0132074  -4.783 1.75e-06 ***
weather3            -0.5463790  0.0224640 -24.322  < 2e-16 ***
weather4            -0.4264111  0.5492894  -0.776  0.43759
windspeed           -0.0018338  0.0007064  -2.596  0.00945 **
```

```
workingday           0.2359567  0.0117181  20.136  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5482 on 10531 degrees of freedom
Multiple R-squared:  0.8498,    Adjusted R-squared:  0.8493
F-statistic:  1862 on 32 and 10531 DF,  p-value: < 2.2e-16
```



```
> summary(fit2.lm)

Call:
lm(formula = (log(casual + 1)) ~ atemp + weekday + hour + I(year ==
    2012) + holiday + humidity + season + temp + weather + windspeed +
    workingday, data = subtrain)

Residuals:
     Min       1Q   Median       3Q      Max
-2.69788 -0.38509  0.05556  0.44220  2.45043

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.2659224  0.1066184  11.873  < 2e-16 ***
atemp                0.0519263  0.0044219  11.743  < 2e-16 ***
weekday              0.0294597  0.0051883   5.678 1.40e-08 ***
hour1               -0.5073162  0.0456817 -11.105  < 2e-16 ***
hour2               -0.8110335  0.0463972 -17.480  < 2e-16 ***
hour3               -1.1182250  0.0458042 -24.413  < 2e-16 ***
hour4               -1.2667972  0.0440394 -28.765  < 2e-16 ***
hour5               -1.1099584  0.0435521 -25.486  < 2e-16 ***
hour6               -0.4127001  0.0434963  -9.488  < 2e-16 ***
hour7                0.3695785  0.0434778   8.500  < 2e-16 ***
hour8                0.9773192  0.0434970  22.469  < 2e-16 ***
hour9                1.1642355  0.0434743  26.780  < 2e-16 ***
hour10               1.3497888  0.0436689  30.910  < 2e-16 ***
hour11               1.4980968  0.0438951  34.129  < 2e-16 ***
hour12               1.5544834  0.0441099  35.241  < 2e-16 ***
hour13               1.5432798  0.0443987  34.760  < 2e-16 ***
hour14               1.5376487  0.0446193  34.462  < 2e-16 ***
hour15               1.5469322  0.0446066  34.679  < 2e-16 ***
hour16               1.5831616  0.0444302  35.633  < 2e-16 ***
hour17               1.6697157  0.0443417  37.656  < 2e-16 ***
hour18               1.4767233  0.0441851  33.421  < 2e-16 ***
hour19               1.2781454  0.0437595  29.208  < 2e-16 ***
hour20               1.0467606  0.0435952  24.011  < 2e-16 ***
hour21               0.8774678  0.0435164  20.164  < 2e-16 ***
hour22               0.6994542  0.0434697  16.091  < 2e-16 ***
hour23               0.4254453  0.0434259   9.797  < 2e-16 ***
I(year == 2012)TRUE  0.2649351  0.0126319  20.974  < 2e-16 ***
holiday             -0.0810187  0.0460153  -1.761   0.0783 .
humidity            -0.0046676  0.0004428 -10.542  < 2e-16 ***
season               0.0988577  0.0059901  16.503  < 2e-16 ***
temp                 0.0216869  0.0048012   4.517 6.34e-06 ***
weather2            -0.0882086  0.0155084  -5.688 1.32e-08 ***
weather3            -0.6394616  0.0263816 -24.239  < 2e-16 ***
```

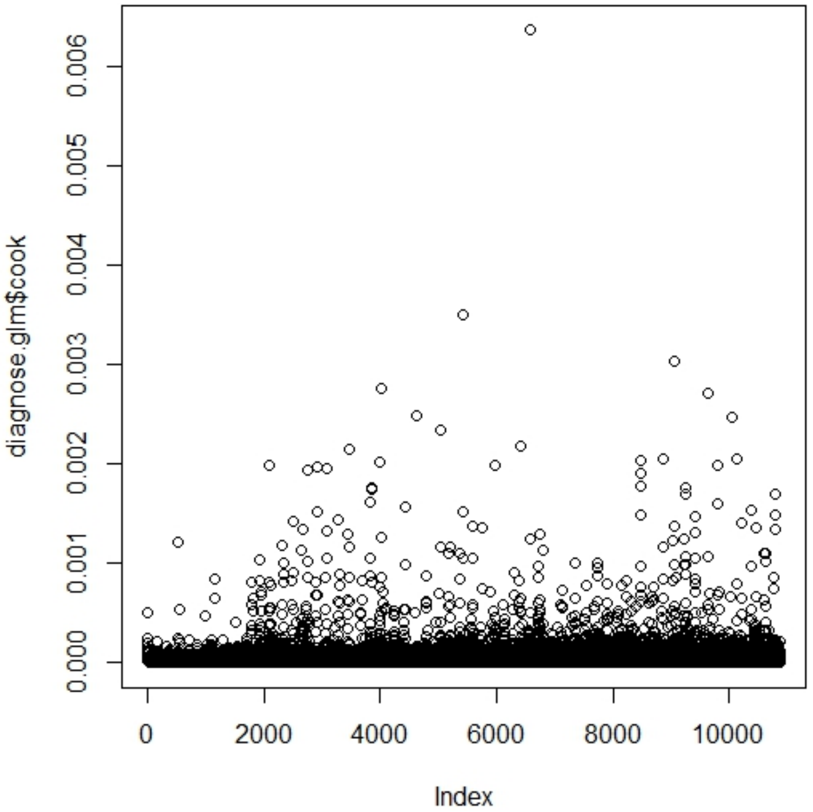
```
weather4            -0.3665702  0.6446977  -0.569   0.5696    
windspeed           -0.0017858  0.0008451  -2.113   0.0346 *  
workingday          -0.5491209  0.0228941 -23.985  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6434 on 10528 degrees of freedom
Multiple R-squared:  0.8173,	Adjusted R-squared:  0.8167 
F-statistic:  1346 on 35 and 10528 DF,  p-value: < 2.2e-16
```



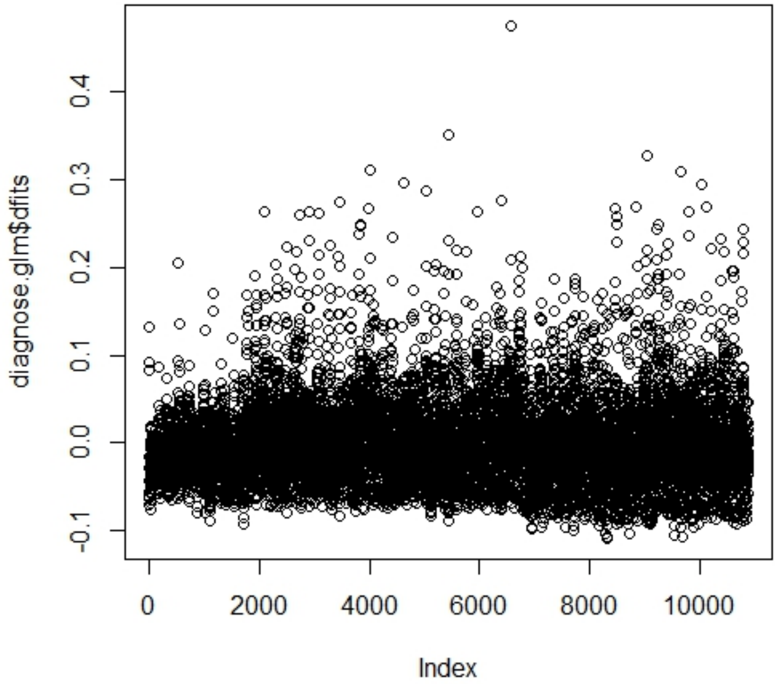
Appendix: R code & Secondary Plots

```
#Visualization part for evaluating the data

library(dplyr)

setwd("/Users/apple/Documents/study/R/BikeSharingDemand/myattempt")
train = read.csv("train.csv")
test = read.csv("test.csv")
attach(train)
# http://www.cookbook-r.com/Manipulating_data/Summarizing_data/
# We have numerical variables: temp, atemp, humidity and windspeed

train$datetime = strptime(train$datetime, format = "%Y-%m-%d%H:%M:%S")
train$weekday = weekdays(train$datetime)
train$hour = train$datetime$hour
# We have categorical variables: season(1/2/3/4), holiday(0/1),
workingday(0/1),weather(1/2/3) and now we add hour and weekday
season_count = aggregate(train["count"], by=train[c("season")], FUN=sum)
holiday_count = aggregate(train["count"], by=train[c("holiday")], FUN=sum)
workingday_count = aggregate(train["count"], by=train[c("workingday")], FUN=sum)
weather_count = aggregate(train["count"], by=train[c("weather")], FUN=sum)
hour_count = aggregate(train["count"], by=train[c("hour")], FUN=sum)
weekday_count = aggregate(train["count"], by = train[c("weekday")], FUN = sum)


par("mar")
par(mar=c(3.9,3.9,2,2))
par(mfrow=c(3,2))
s_c_plot =barplot(season_count$count,names.arg = season_count$season,main = "count VS
season",xlab = "season", ylab = "count")
h_c_plot =barplot(holiday_count$count,names.arg = holiday_count$holiday,main = "count VS
holiday",xlab = "holiday", ylab = "count")
wk_c_plot = barplot(weekday_count$count,names.arg = weekday_count$weekday,main = "count
VS weekday",xlab = "weekday", ylab = "count")
we_c_plot = barplot(weather_count$count,names.arg = weather_count$weather,main = "count
VS weather",xlab = "weather", ylab = "count")
h_c_plot = barplot(hour_count$count,names.arg = hour_count$hour,main = "count VS
hour",xlab = "hour", ylab = "count")
work_c_plot = barplot(workingday_count$count,names.arg = workingday_count$workingday,main
= "count VS workingday",xlab = "workingday", ylab = "count")

# part one for model selection
# first try some parameters

featureEngineer <- function(df){
  # Factorize the data
  names <- c("season", "holiday", "workingday", "weather")
  df[,names]<-lapply(df[,names],factor)
  # Extract the day of the week
  df$datetime <- as.character(df$datetime)
  df$datetime <- strptime(df$datetime, format = "%Y-%m-%d%T", tz = "EST")
  #parse the hour
  df$hour <- as.integer(substr(df$datetime,12,13))
  df$hour <- as.factor(df$hour)
  #get the weekday for each date using weekdays function
  df$weekday <- as.factor(weekdays(df$datetime))
  df$weekday <- factor(df$weekday,
                       levels=c("Monday","Tuesday","Wednesday",
"Thursday","Friday","Saturday","Sunday"))
  #the count also increases as time goes on
  #add year as a factor into our model
  df$year <- as.integer(substr(df$datetime,1,4))
  df$year <- as.factor(df$year)
  # the count also vary among different time
  # to be done next time
  return (df)
}



# Use the function featureEngineer to get the subset of data

subtrain = featureEngineer(train)
attach(subtrain)
head(subtrain)

# Turn the categorical variables into numeric
 subtrain$season = as.numeric(subtrain$season)
 subtrain$holiday = as.numeric(subtrain$holiday)
 subtrain$workingday = as.numeric(subtrain$workingday)
 subtrain$weekday = as.numeric(subtrain$weekday)



# =======================================================
# =======================================================
# Variable selection and creation of explanatory variable
#install.packages('leaps')
library(leaps)

cat("\nexhaustive\n")
out.exh=regsubsets(registered~atemp+weekday+hour+year+holiday+humidity+season+temp+weathe
r+windspeed+workingday,data=subtrain,nbest=1,nvmax=40)
summ.exh=summary(out.exh)
names(summ.exh)
print(summ.exh$outmat)
c =print(summ.exh$cp)
```

```
a = print(summ.exh$adjr)
#Good model should with  small cp and large adjr, we choose the one with variables that
we put in fit1.glm
minc = min(c)
cindex = which(c==minc)
maxa = max(a)
aindex = which(a==maxa)
# The best model according to CP has 32 variables while the best model according to adjr
has 34 variables, we choose the one with 32 variables since the adr are pretty close
# The variables are:
atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday
# cp=33.16922, adjr=0.6840206
# model selection by exhaustive search

# fit1.glm: generalized linear model with 32 variables, response variable as registered
fit1.glm=
glm(registered~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday,fami
ly="poisson")
#AIC: 348374

#fit1.lm: linear model with 32 variables, response variable as registered
fit1.lm =
lm(registered~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday)
# Residual standard error: 84.53 on 10851 degrees of freedom
# Multiple R-squared:  0.6878,  Adjusted R-squared:  0.6868
# F-statistic: 703.1 on 34 and 10851 DF,  p-value: < 2.2e-16
summ1.glm = summary(fit1.glm)
summ1.lm = summary(fit1.lm)


par("mar")
par(mar = c(4,4,1,1))
# Checking Cook's distance for abnormal data or observation
diagnose.glm = ls.diag(fit1.glm)
print(diagnose.glm$cook)
plot(diagnose.glm$cook)
```



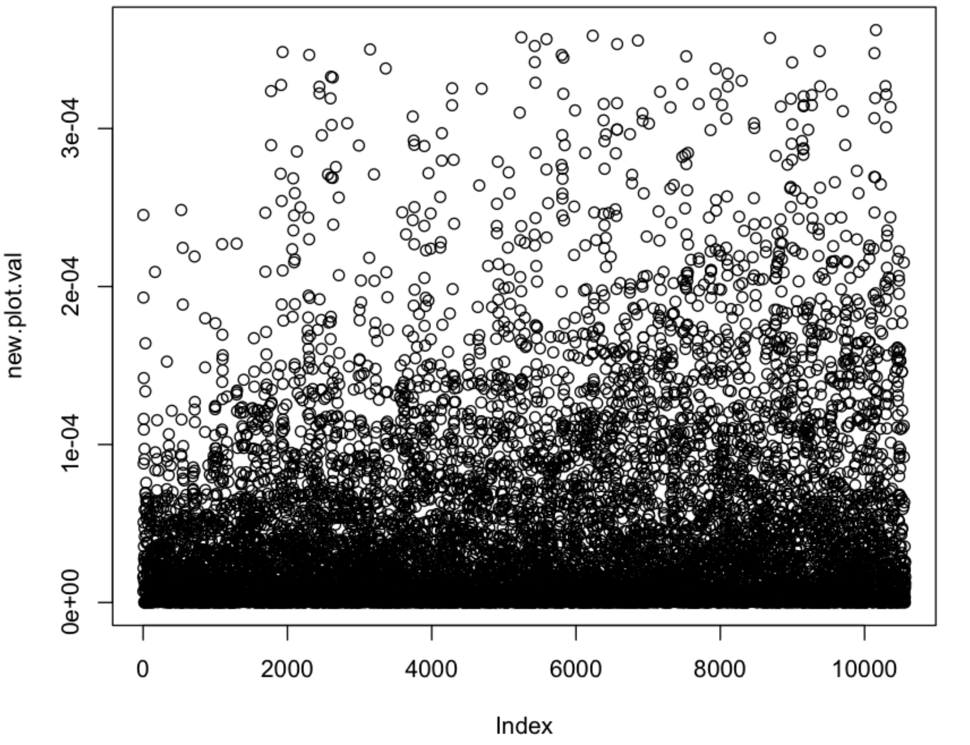
```
print(diagnose.glm$dfits)
plot(diagnose.glm$dfits)
```



```
# Remove the data with large Cook's distance, we use the criteria to remove certain data
that has
# Cook's distance >= (4/n), with n is the number of data
```

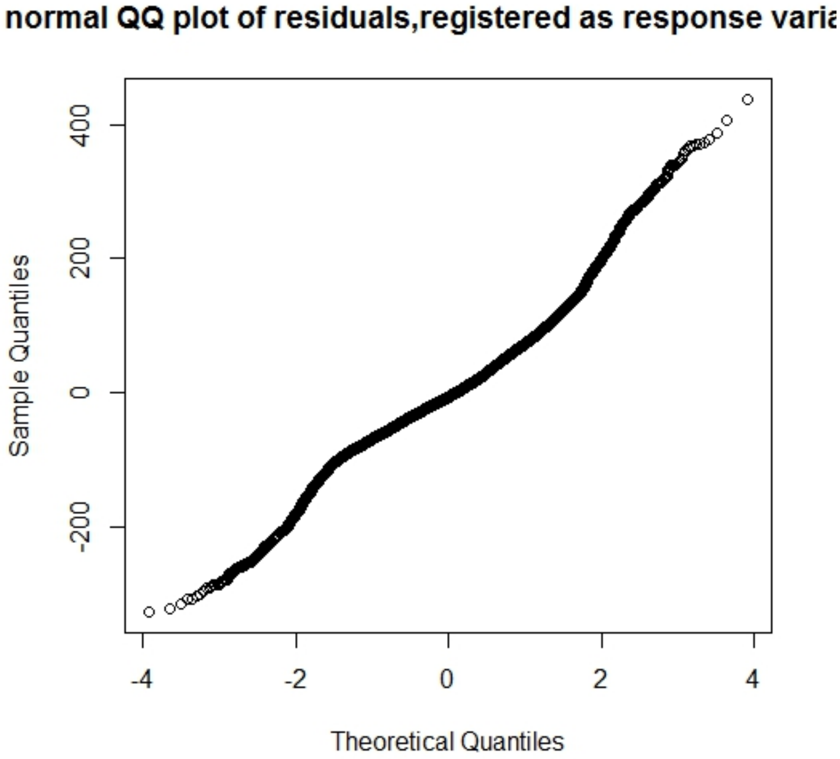
```
n = length(diagnose.glm$cook)
indexvector = c()

for (i in 1:n){


  if (!is.na(diagnose.glm$cook[i])){
    if (diagnose.glm$cook[i]>(4/n)){

    indexvector = c(indexvector,i) }
}
}
# The new subtrain is the one with abnorm data subtracted
subtrain = subtrain[-indexvector,]

# plot the cook's distance without outliers
new.plot.val=c()
for (j in
1:n){if(!is.element(j,indexvector)){new.plot.val=c(new.plot.val,diagnose.glm$cook[j])}}
plot(new.plot.val)
```

After all the outliers of our data set are removed, the plot of cook's distance is shown as below:

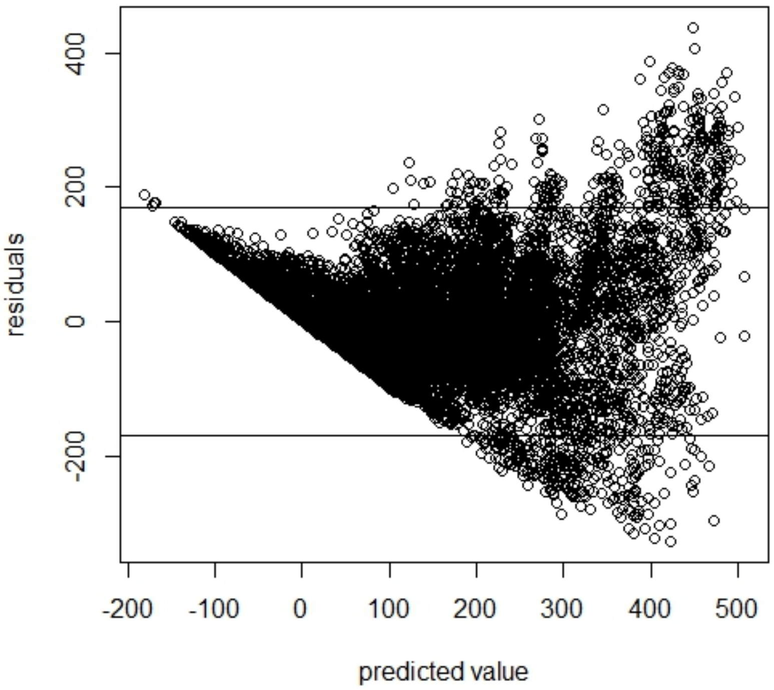


```
# Testing the linear model fit1.lm by residual plot

names(summ1.lm)
```

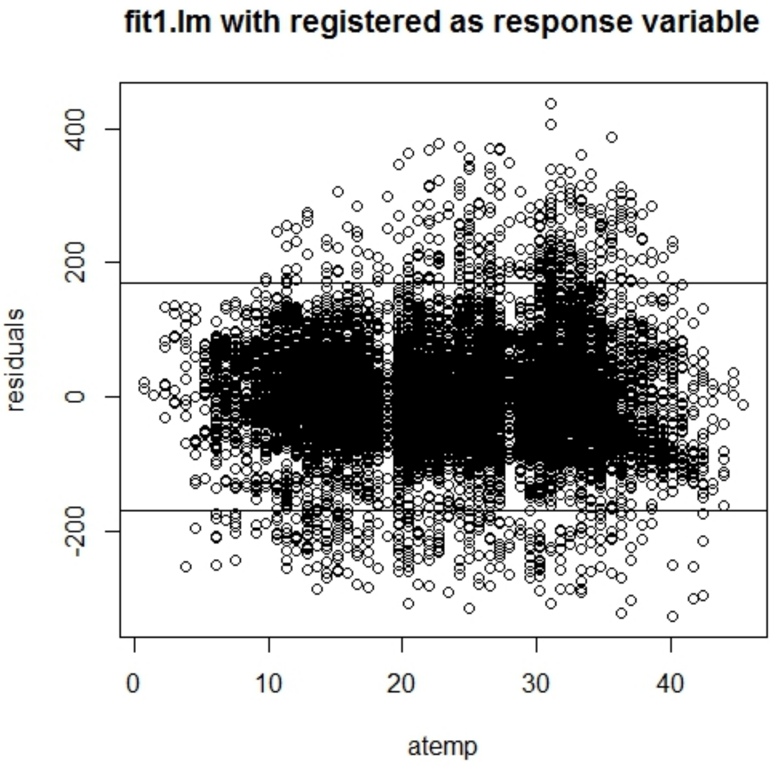
```
pred1.lm = predict(fit1.lm, data = subtrain)  # predicted value from fit1.lm regression
model
res1.lm = resid(fit1.lm, data = subtrain)     # residuals
sigma1.lm = summ1.lm$sigma

pred1.glm = predict(fit1.glm, data = subtrain)  # predicted value from fit1.lm regression
model
res1.glm = resid(fit1.glm, data = subtrain)     # residuals
sigma1.glm = summ1.glm$sigma


# residual plots
par(mfrow = c(1,1))
#plot 1 for all variables
qqnorm(res1.lm,main = "normal QQ plot of residuals,registered as response variable")
```



```
plot(pred1.lm,res1.lm,xlab = "predicted value",ylab = "residuals");abline(h =
2*sigma1.lm);abline(h = -2*sigma1.lm)
```

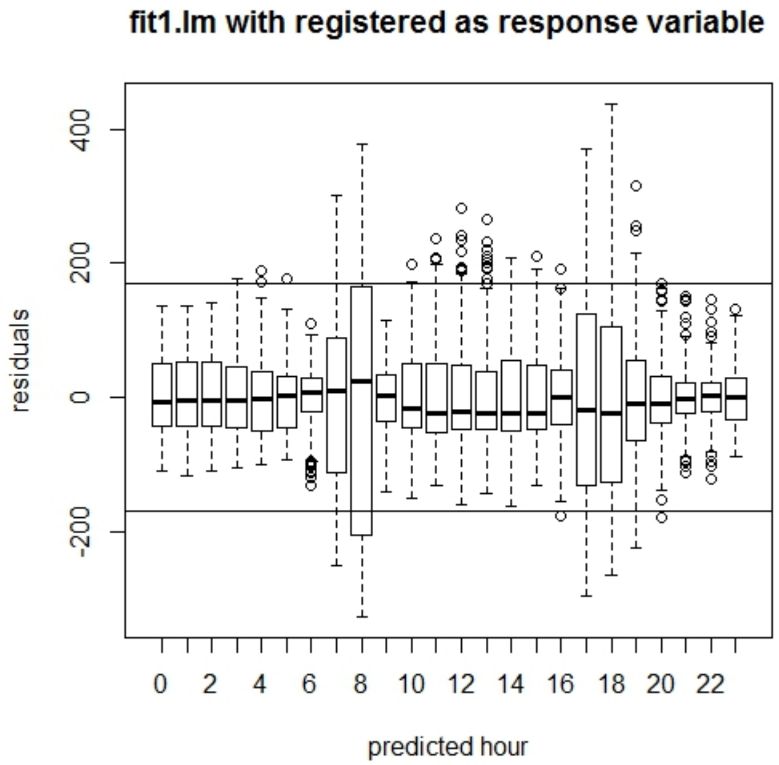

residuals
400
200
0
-200
-200
-100
0
100
200
300
400
500
predicted value

```
#The variables we shall check:
atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday
# Residual plots for quantitative variables
# par(mfrow = c(2,2))

# plot for atemp
plot(atemp,res1.lm,xlab = "atemp",ylab = "residuals",main = "fit1.lm with registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```

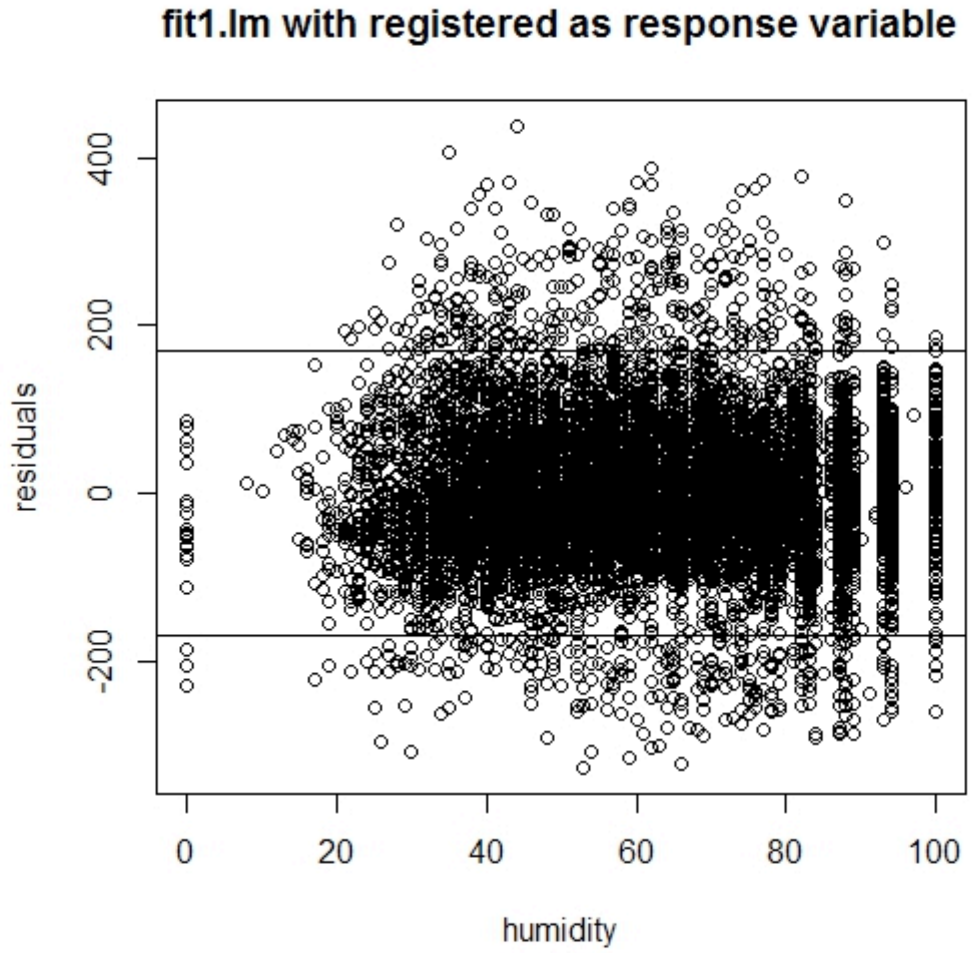


As we can see from the atemp vs. residuals plot, the residuals have a certain pattern, i.e., larger at the middle of the range of atemp and smaller at the two ends. This pattern suggests us to add a quadratic term of atemp variable.
After the quadratic term of atemp variable is added, the AIC value of general linear model decreases to 467758 and the adjusted R-squared value of linear model increases to 0.5774.

```
# plot for hour
plot(hour,res1.lm,xlab = "predicted hour",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



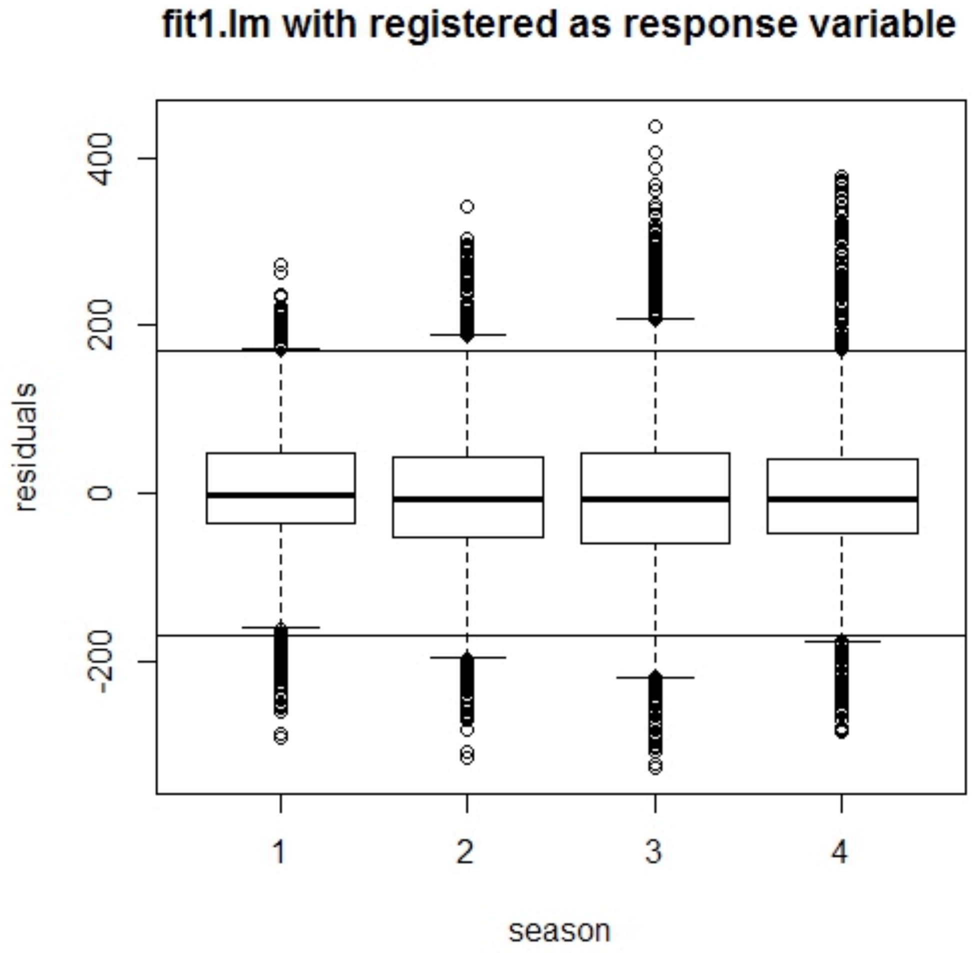
```
# plot for humidity
plot(humidity,res1.lm,xlab = "humidity",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



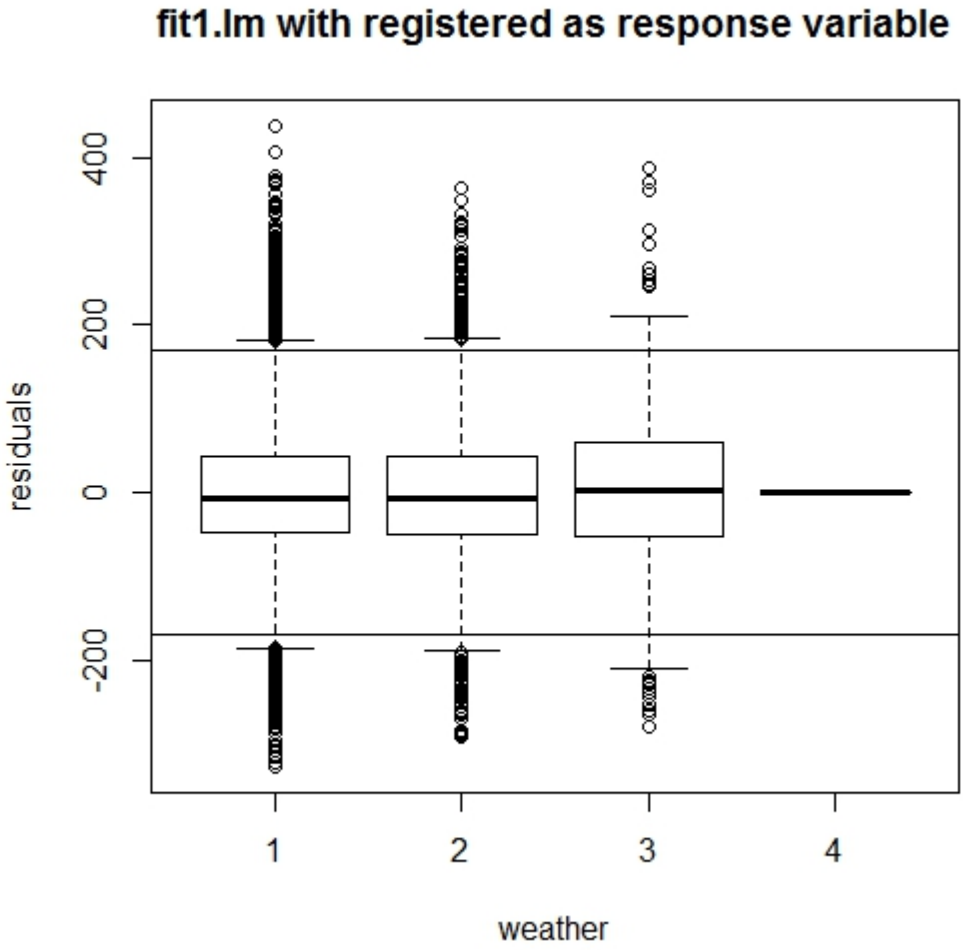
As we can see from the humidity vs. residuals plot, the residuals have a certain pattern, i.e., the residuals increase as humidity increases. This pattern suggests us to add a natural logarithm term of humidity variable.

After the natural logarithm term of humidity variable is added, the AIC value of general linear model decreases to 463101 and the adjusted R-squared value of linear model increases to 0.5783.

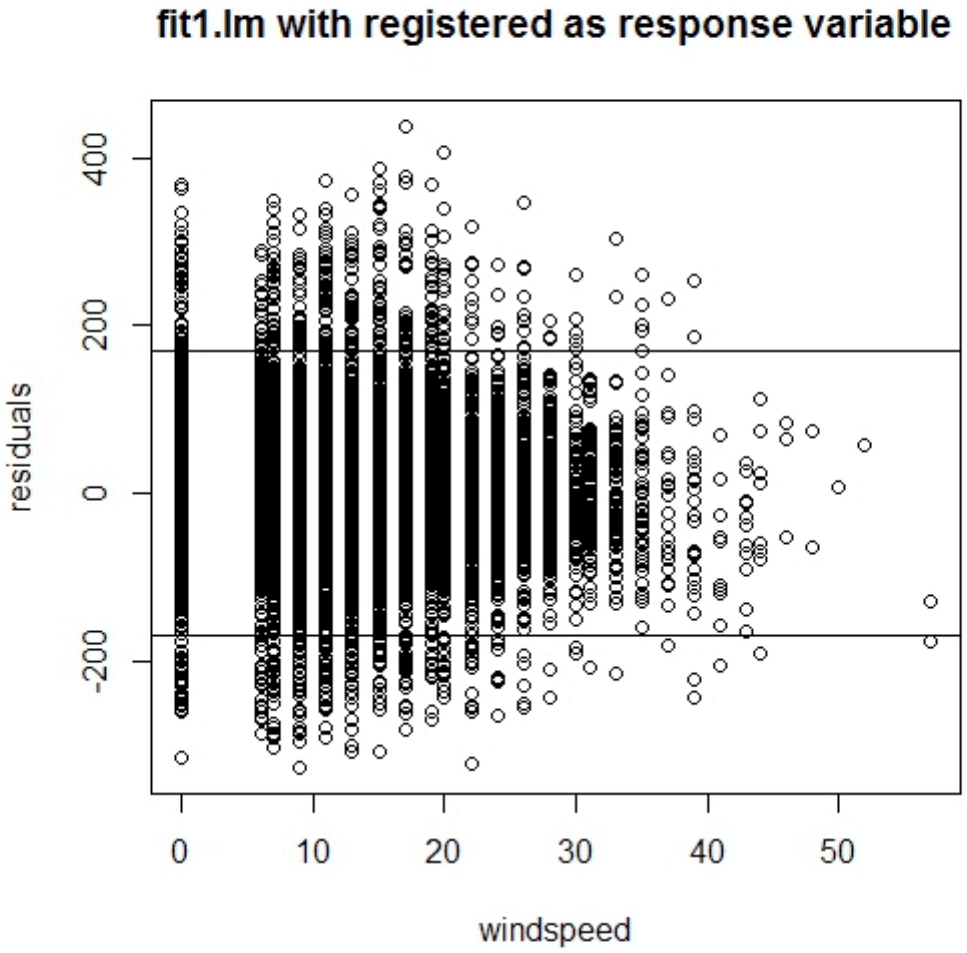
```
#plot for season
plot(season,res1.lm,xlab = "season",ylab = "residuals",main = "fit1.lm with registered as
response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



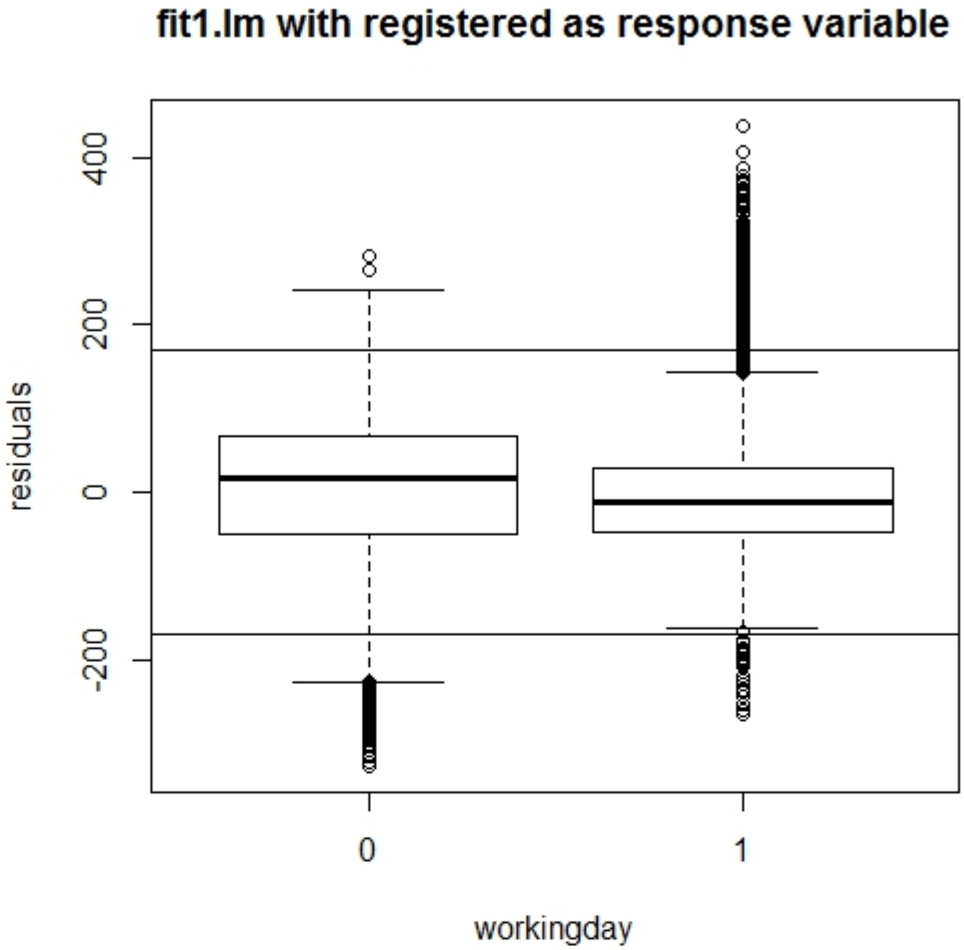
```
#plot for weather
plot(weather,res1.lm,xlab = "weather",ylab = "residuals",main = "fit1.lm with registered
as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



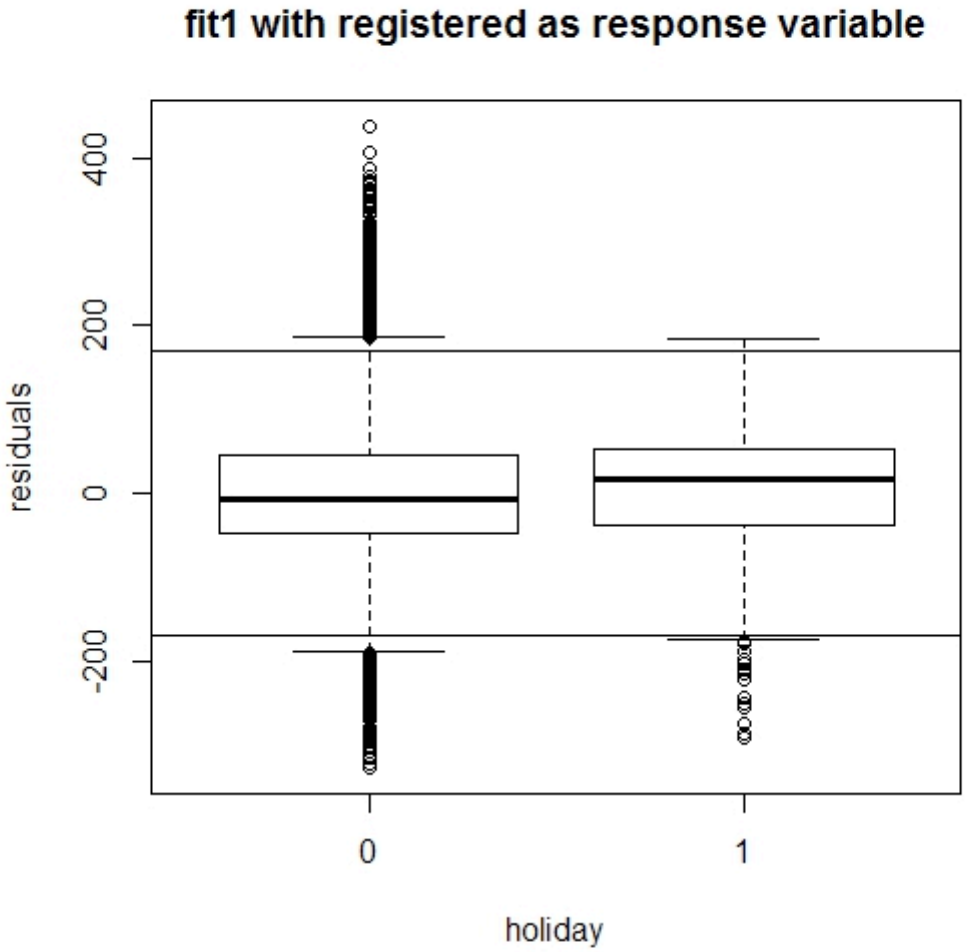
```
# plot for windspeed
plot(windspeed,res1.lm,xlab = "windspeed",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



```
#plot for workingday
plot(workingday,res1.lm,xlab = "workingday",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



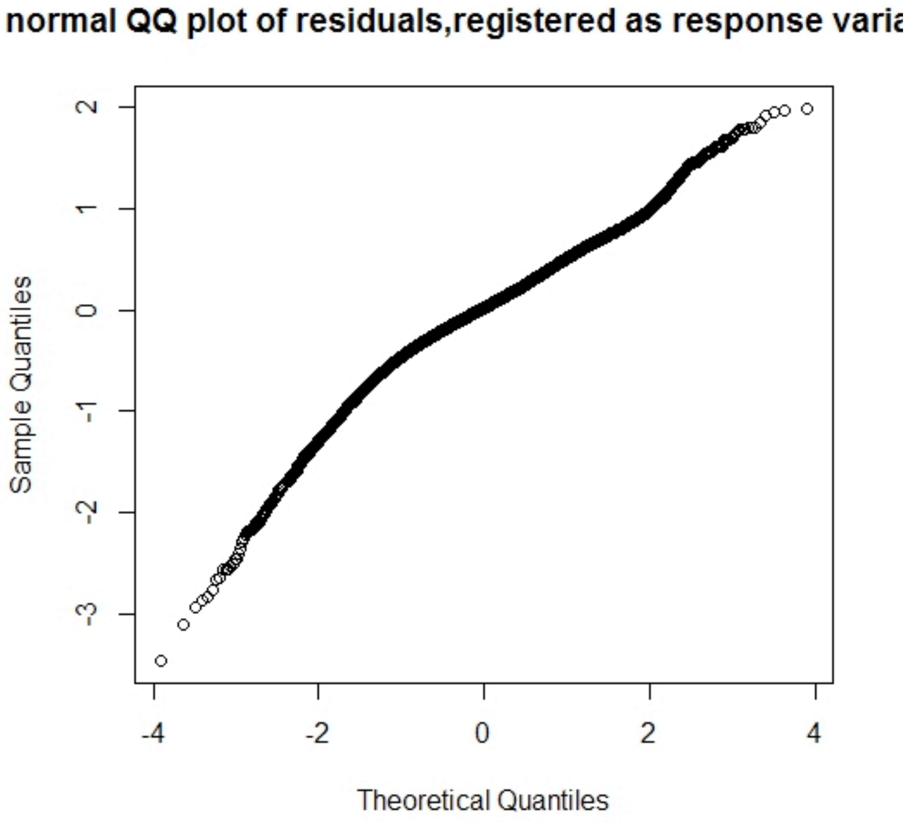
```
#plot for holiday
plot(holiday,res1.lm,xlab = "holiday",ylab = "residuals",main = "fit1 with registered as
response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



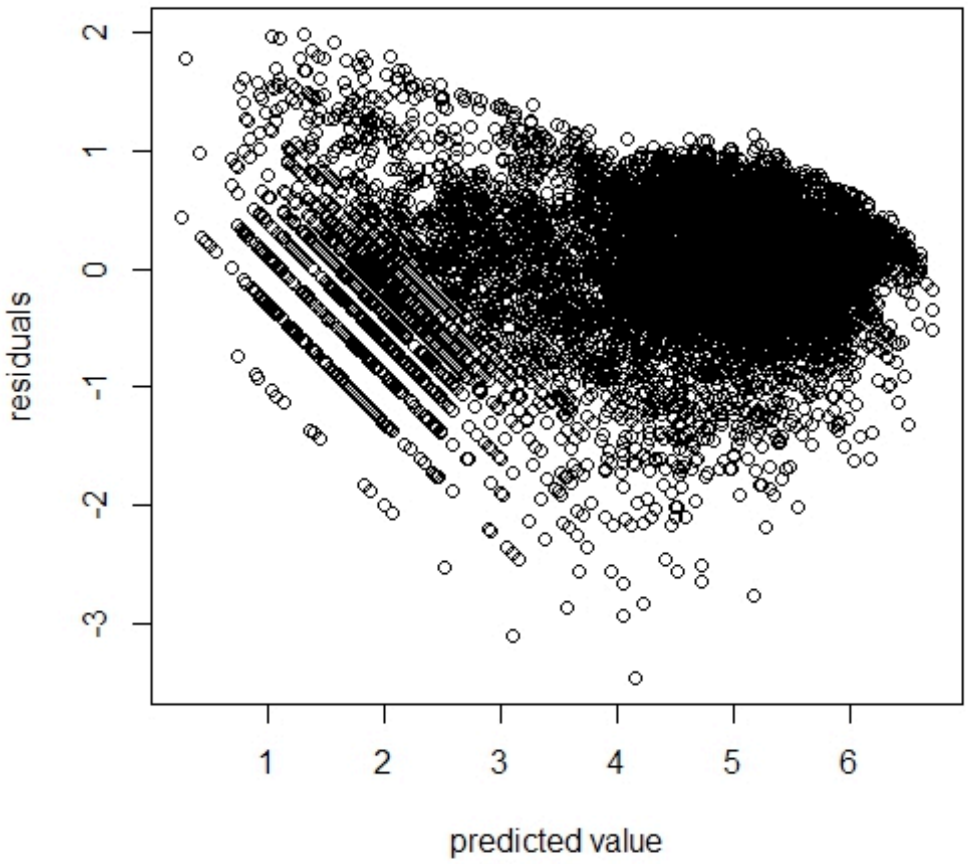
```
# Adjust the linear model glm.lm and nonlinear model fit1.glm with residual plot,mainly
time period
subtrain$numHour = as.numeric(subtrain$hour)
subtrain$hourcat6 = cut(subtrain$numHour,breaks = c(-Inf, 6, 10, 16, 20,Inf), labels =
c(1:5))
out.exh=regsubsets(registered~atemp+weekday+hourcat6+year+holiday+humidity+season+temp+we
ather+windspeed+workingday,data=subtrain,nbest=1,nvmax=40)
summ.exh=summary(out.exh)
names(summ.exh)
print(summ.exh$outmat)
c = print(summ.exh$cp)
a = print(summ.exh$adjr)
minc = min(c)
maxa = max(a)
cindex = which(c == minc)
aindex = which(a == maxa)
# The minimum is with 11 variables
```

```
fit1.lm =
lm(registered~atemp+hourcat6+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+wo
rkingday,data=subtrain)
# Residual standard error: 98.54 on 10552 degrees of freedom
# Multiple R-squared:  0.5775,  Adjusted R-squared:  0.577
# F-statistic:  1311 on 11 and 10552 DF,  p-value: < 2.2e-16


fit1.glm =
glm(registered~atemp+hourcat6+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+w
orkingday,data=subtrain,family=poisson)
summ1.glm = summary(fit1.glm)
#AIC: 477430

attach(subtrain)
#Make transformations to the variables according to residual plot
fit1.lm =
lm(registered~I(atemp^2)+atemp+hourcat6+I(year==2012)+humidity+I(log(humidity+1))+season+
I(weather==2)+I(weather==3)+workingday,data=subtrain)

# Residual standard error: 98.39 on 10550 degrees of freedom
# Multiple R-squared:  0.5788,  Adjusted R-squared:  0.5783
# F-statistic:  1115 on 13 and 10550 DF,  p-value: < 2.2e-16

# We can see that the adjusted-r does not improve much(0.577,0.5783)
# We plan to transform the response variable

fit1.lm =
lm(I(log(registered+1))~I(atemp^2)+atemp+hourcat6+I(year==2012)+humidity+I(log(humidity+1
))+season+I(weather==2)+I(weather==3)+workingday,data=subtrain)
# Residual standard error: 0.6861 on 10550 degrees of freedom
# Multiple R-squared:  0.7644,  Adjusted R-squared:  0.7641
# F-statistic:  2633 on 13 and 10550 DF,  p-value: < 2.2e-16

fit1.lm =
lm(I(log(registered+1))~atemp+hourcat6+I(year==2012)+humidity+season+weather+windspeed+wo
rkingday,data=subtrain)
# Residual standard error: 0.6934 on 10552 degrees of freedom
# Multiple R-squared:  0.7592,  Adjusted R-squared:  0.759
# F-statistic:  3025 on 11 and 10552 DF,  p-value: < 2.2e-16

fit1.lm =
lm(I(log(registered+1))~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workin
gday,data=subtrain)
# Residual standard error: 0.5482 on 10531 degrees of freedom
# Multiple R-squared:  0.8498,  Adjusted R-squared:  0.8493
# F-statistic:  1862 on 32 and 10531 DF,  p-value: < 2.2e-16
```

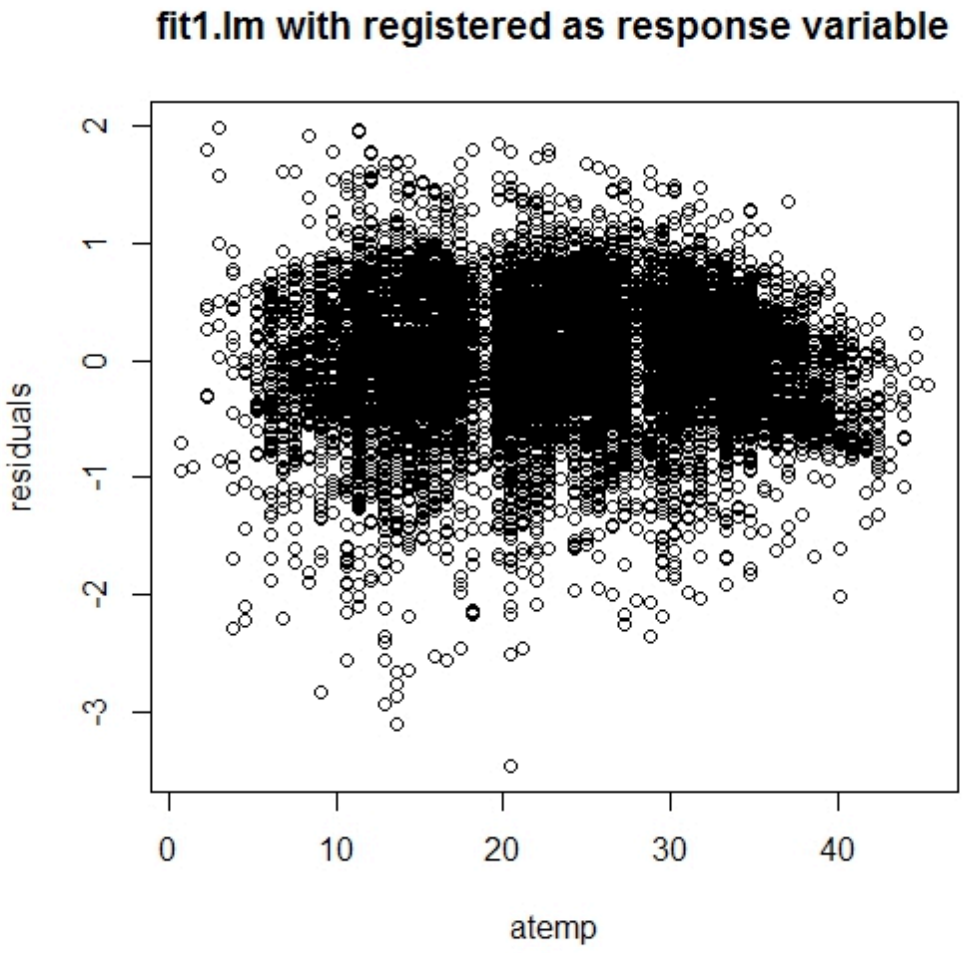
```
pred1.lm = predict(fit1.lm, data = subtrain)  # predicted value from fit1.lm regression
model
res1.lm = resid(fit1.lm, data = subtrain)     # residuals
sigma1.lm = summ1.lm$sigma
summ1.lm = summary(fit1.lm)


# residual plots
par(mfrow = c(1,1))
#plot 1 for all variables
qqnorm(res1.lm,main = "normal QQ plot of residuals,registered as response variable")
```



```
plot(pred1.lm,res1.lm,xlab = "predicted value",ylab = "residuals");abline(h =
2*sigma1.lm);abline(h = -2*sigma1.lm)
```

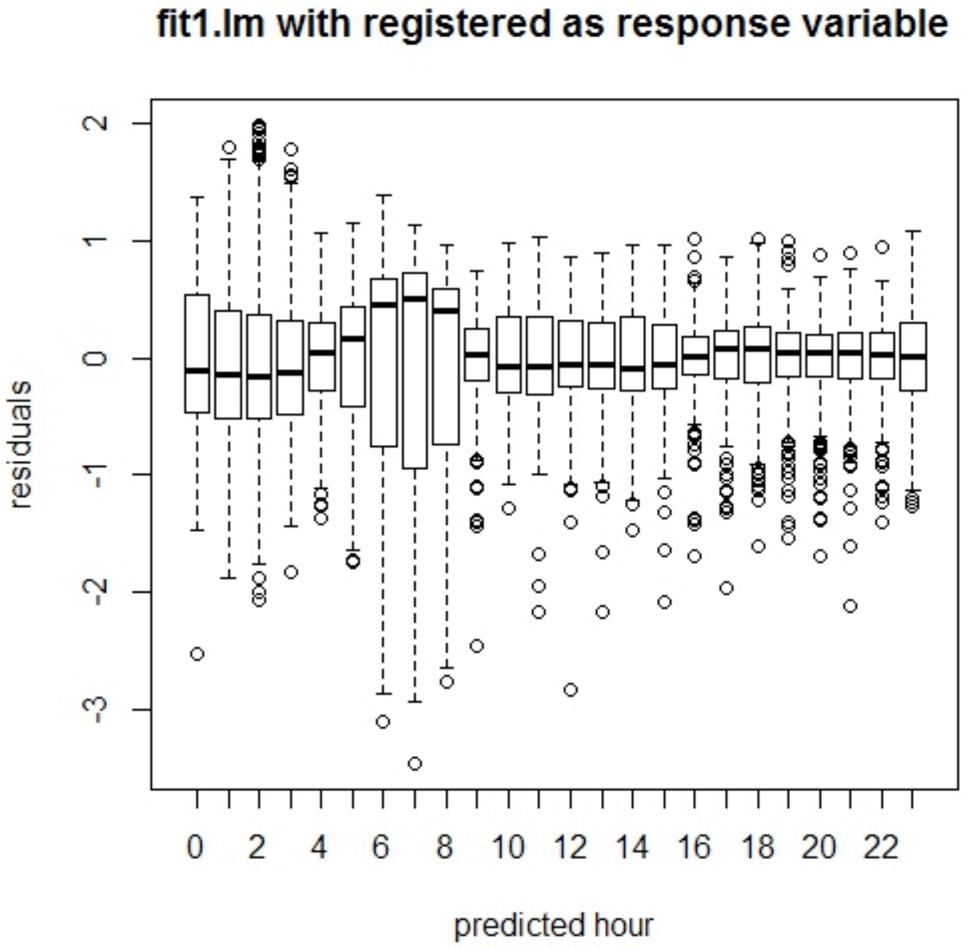


```
#The variables we shall check:
atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday

# plot for atemp
plot(atemp,res1.lm,xlab = "atemp",ylab = "residuals",main = "fit1.lm with registered as
response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```

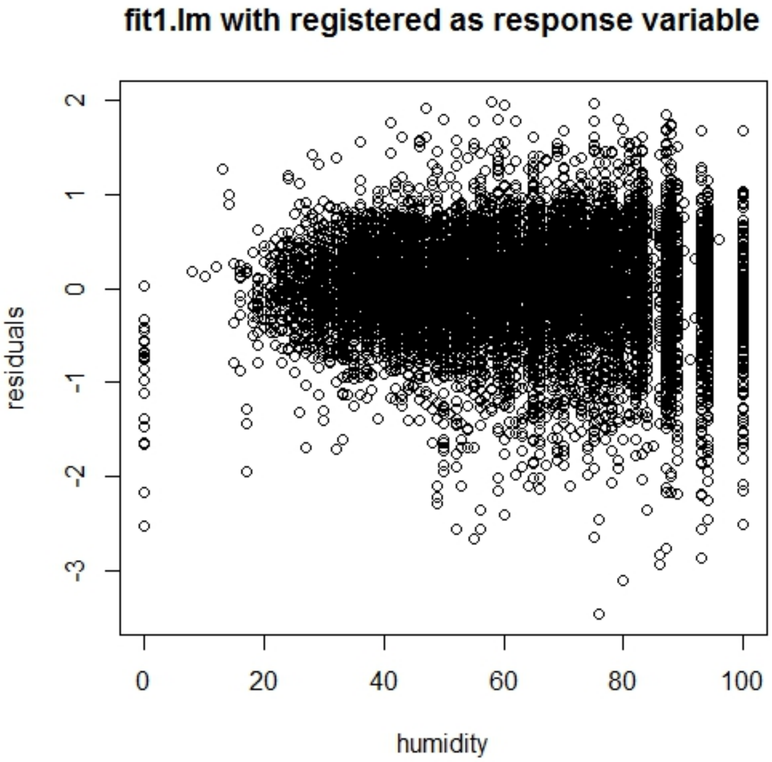


```
#####
```

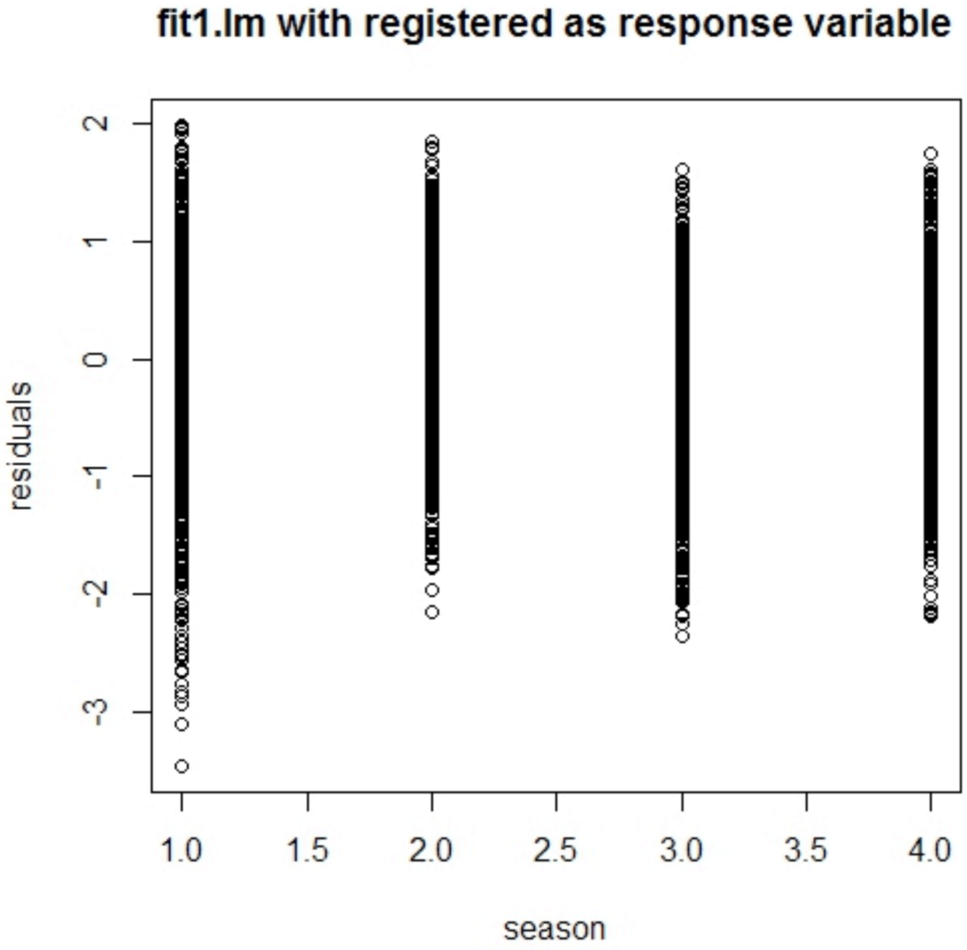
```
# plot for hour
plot(hour,res1.lm,xlab = "predicted hour",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



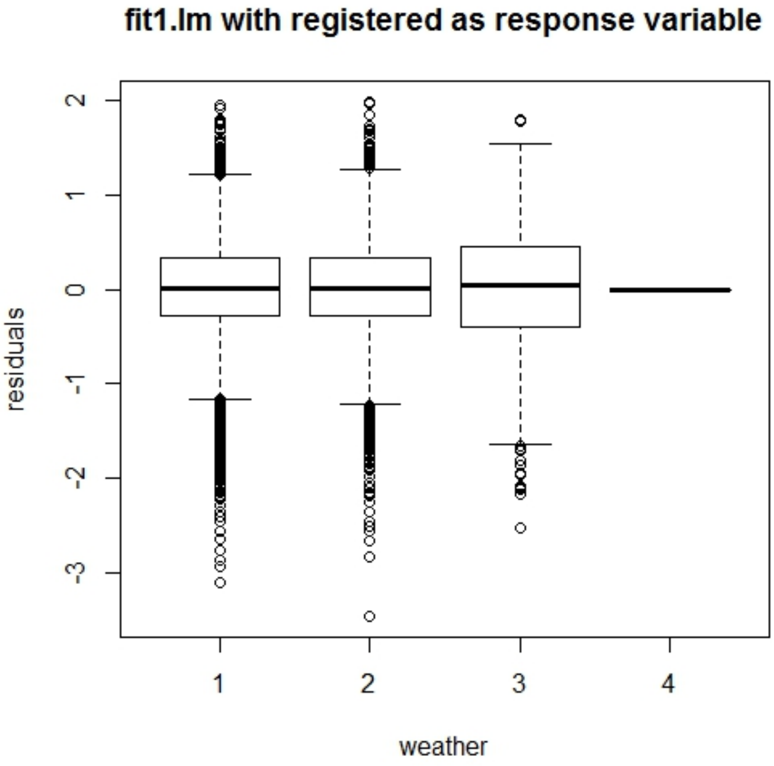
```
# plot for humidity
plot(humidity,res1.lm,xlab = "humidity",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



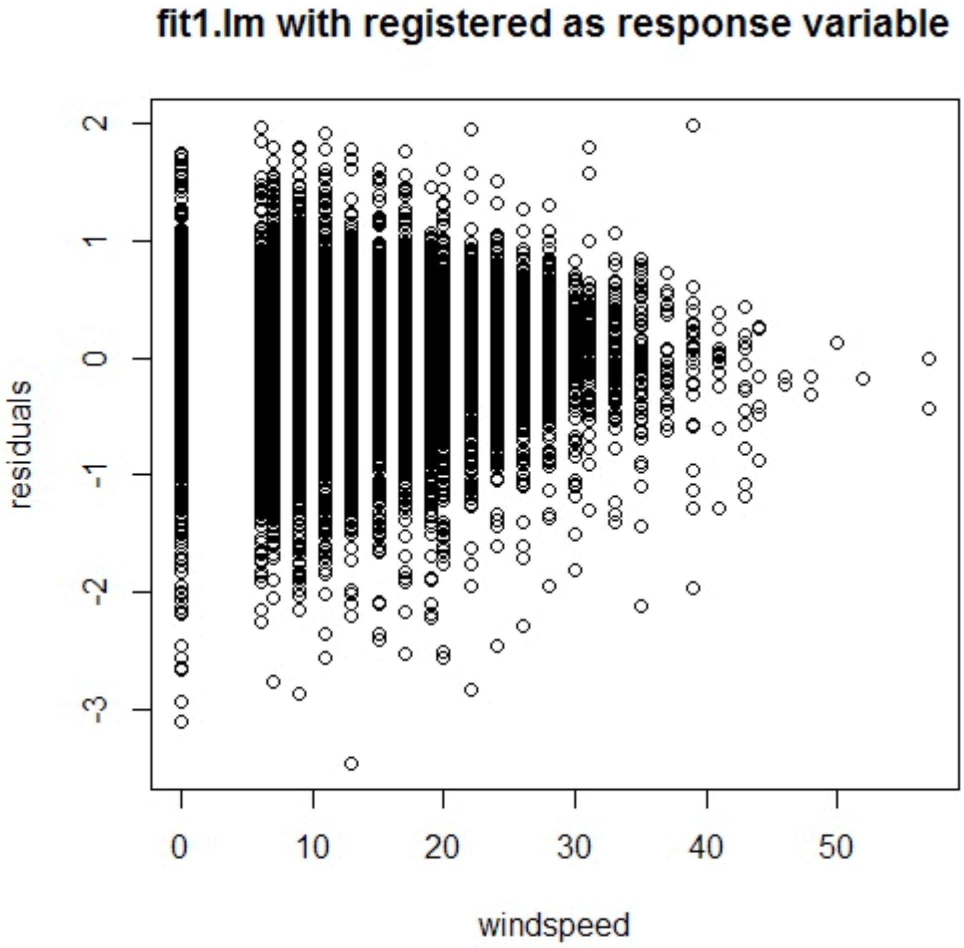
```
#plot for season
plot(season,res1.lm,xlab = "season",ylab = "residuals",main = "fit1.lm with registered as
response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



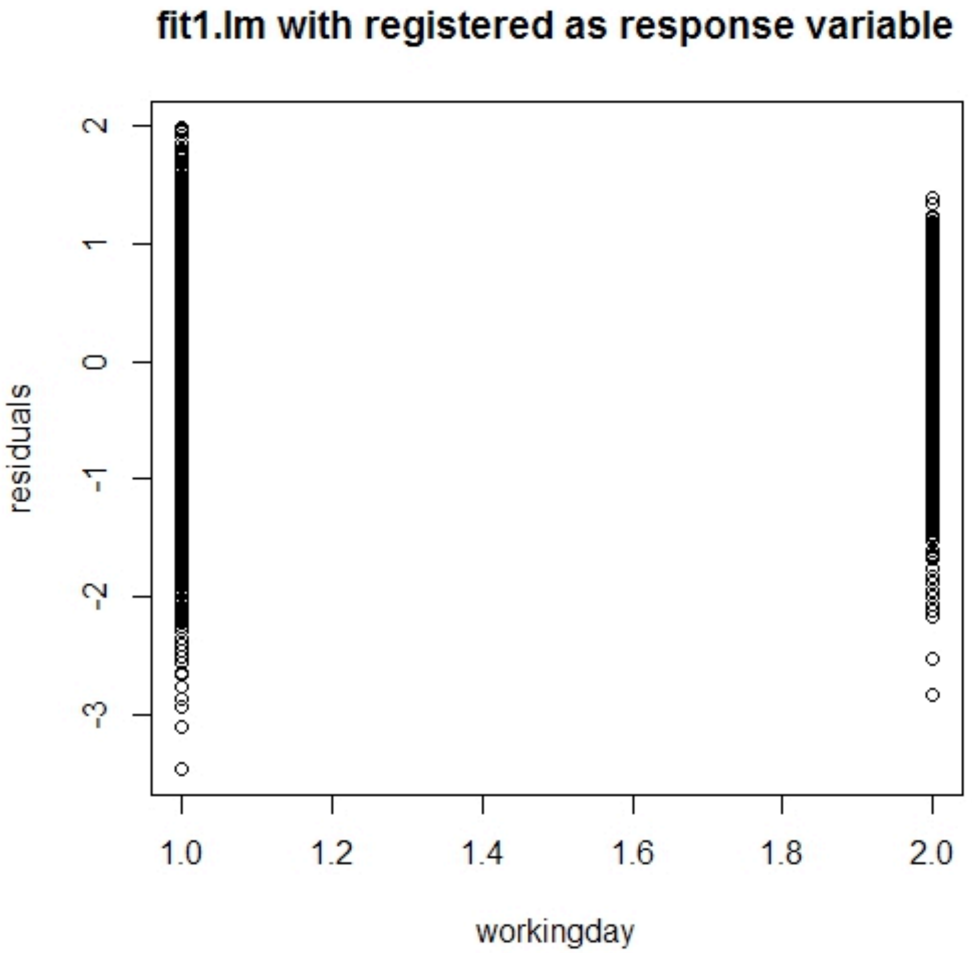
```
#plot for weather
plot(weather,res1.lm,xlab = "weather",ylab = "residuals",main = "fit1.lm with registered
as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



```
# plot for windspeed
plot(windspeed,res1.lm,xlab = "windspeed",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```

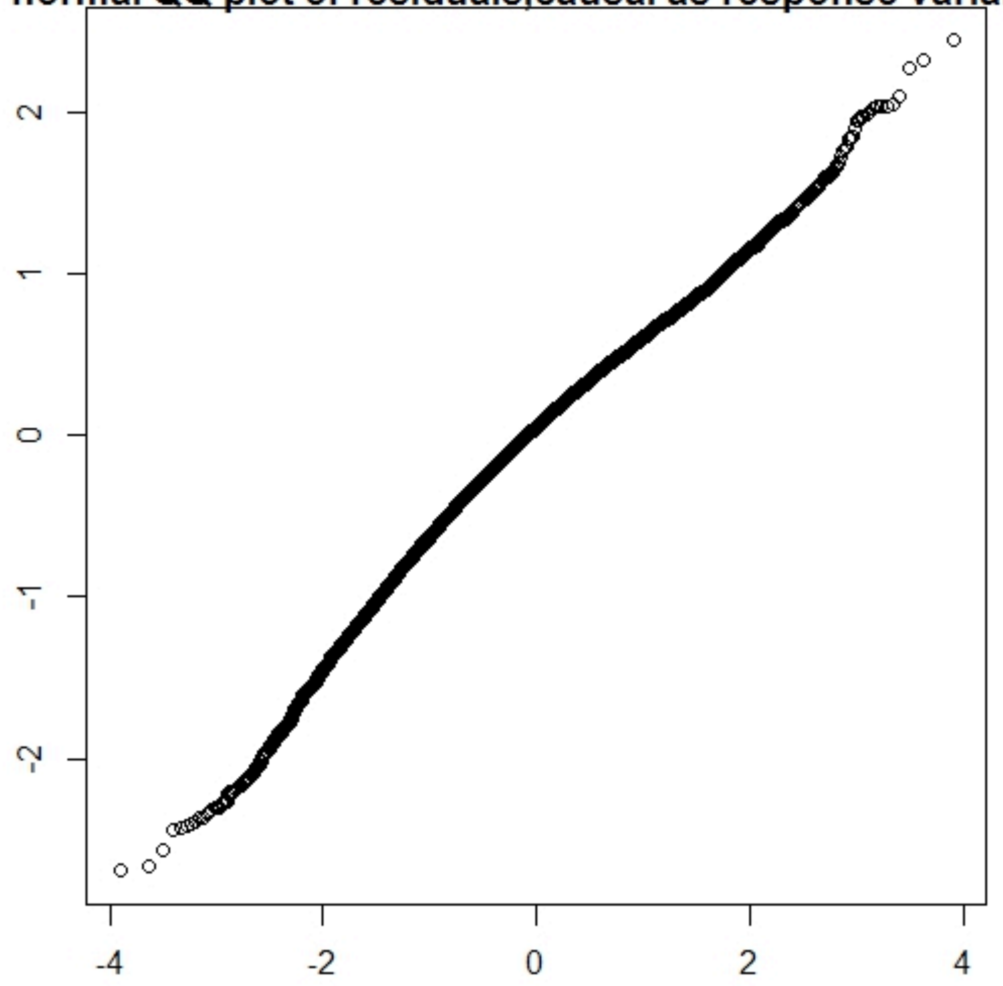


```
#plot for workingday
plot(workingday,res1.lm,xlab = "workingday",ylab = "residuals",main = "fit1.lm with
registered as response variable");abline(h = 2*sigma1.lm);abline(h = -2*sigma1.lm)
```



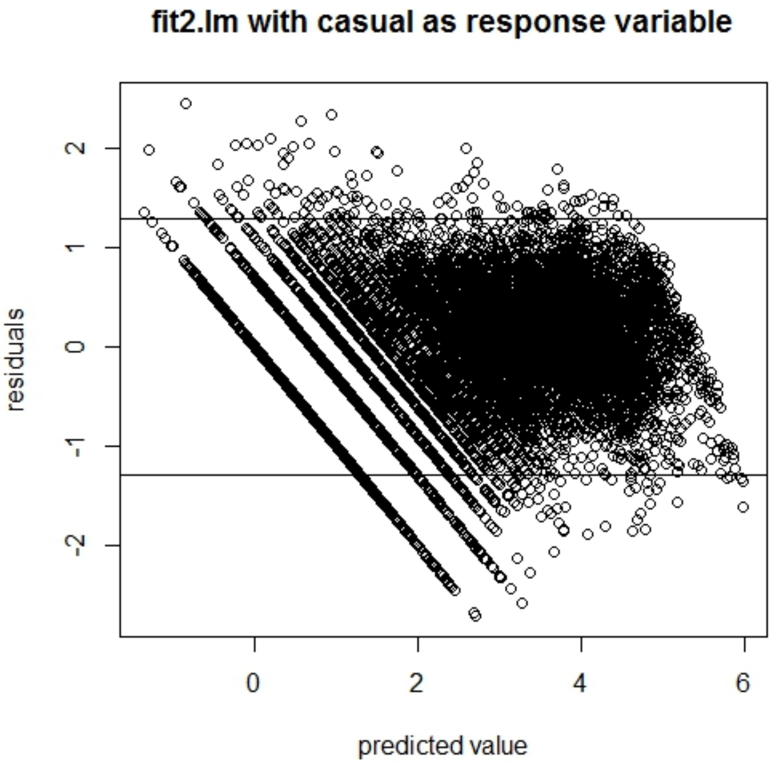
```
# FIT2:
# variable selection and creation of explanatory variable

# The process for model selection
cat("\nexhaustive\n")
out.casual.exh=regsubsets(casual~atemp+weekday+hour+year+holiday+humidity+season+temp+wea
ther+windspeed+workingday,data=subtrain,nbest=1,nvmax = 40)
summ.casual.exh=summary(out.casual.exh)
names(summ.casual.exh)
print(summ.casual.exh$outmat)
c = print(summ.casual.exh$cp)
a = print(summ.casual.exh$adjr)
minc = min(c)
maxa = max(a)
cindex = which(c == minc)
```

```
aindex = which(a == maxa)

# The best model is with 32 variables

# generally want small cp and large adjr
# model selection by exhaustive search
fit2.exh =
lm(casual~atemp+I(hour==13)+I(hour==14)+I(hour==15)+I(hour==16)+I(hour==17)+humidity+work
ingday
# cp=2373.350, adjr=0.4859510

# model selection by backward search
fit2.backw=
lm(casual~I(hour==13)+I(hour==14)+I(hour==15)+I(hour==16)+I(hour==17)+humidity+temp+worki
ngday)
#cp = 2385.866, adjr=0.4854651

# model selection by forward search
fit2.forw=
lm(casual~I(hour==13)+I(hour==14)+I(hour==15)+I(hour==16)+I(hour==17)+humidity+temp+worki
ngday)
# cp = 2385.866, adjr=0.4854651

# model selection by sequential replacement (same result as exhaustive search)

summary(fit2.exh)
summary(fit2.backw)
summary(fit2.forw)

  ## in conclusion, the exhaustive search is the best fit since it has the smallest cp
and largest adjr.

#Fit2: linear model,with response variable as casual, 31 variables
summ.casual.exh$outmat[31,]
subtrain = subtrain[,-(hour == 1)]
subtrain = subtrain[,-(hour == 2)]
fit2.lm = lm(casual~atemp+weekday+hour+I(year ==
2012)+holiday+humidity+season+temp+I(weather == 2)+ I(weather ==
3)+windspeed+workingday,data = subtrain)
# Residual standard error: 32.2 on 10529 degrees of freedom
# Multiple R-squared:  0.5857,  Adjusted R-squared:  0.5843
# F-statistic: 437.8 on 34 and 10529 DF,  p-value: < 2.2e-16

fit2.lm = lm((log(casual+1))~atemp+weekday+hour+I(year ==
2012)+holiday+humidity+season+temp+I(weather == 2)+ I(weather ==
3)+windspeed+workingday,data = subtrain)
# Residual standard error: 0.6434 on 10529 degrees of freedom
# Multiple R-squared:  0.8173,  Adjusted R-squared:  0.8167
# F-statistic:  1386 on 34 and 10529 DF,  p-value: < 2.2e-16

fit2.lm = lm((log(casual+1))~atemp+weekday+hour+I(year ==
2012)+holiday+humidity+season+temp+weather+windspeed+workingday,data = subtrain)
# Residual standard error: 0.628 on 10844 degrees of freedom
# Multiple R-squared:  0.8233,  Adjusted R-squared:  0.8226
# F-statistic:  1232 on 41 and 10844 DF,  p-value: < 2.2e-16

# We could check with the adjr tha the best model is  simply with original explanatory
variables


# Testing the linear model by residual plot
#  season holiday workingday weather temp  atemp humidity windspeed (casual registered
count) weekday hour
summ2 = summary(fit2.lm)
names(summ2)
pred2 = predict(fit2.lm)  # predicted value from fit1 regression model
res2 = resid(fit2.lm)     # residuals
sigma2.lm = summ2$sigma
# residual plots
par(mfrow = c(1,1))
par(mar = c(3,3,1,1))
#plot  for all variables
#atemp+weekday+hour+I(year ==
2012)+holiday+humidity+season+temp+weather+windspeed+workingday
qqnorm(res2,main = "normal QQ plot of residuals,causal as response variable")
```

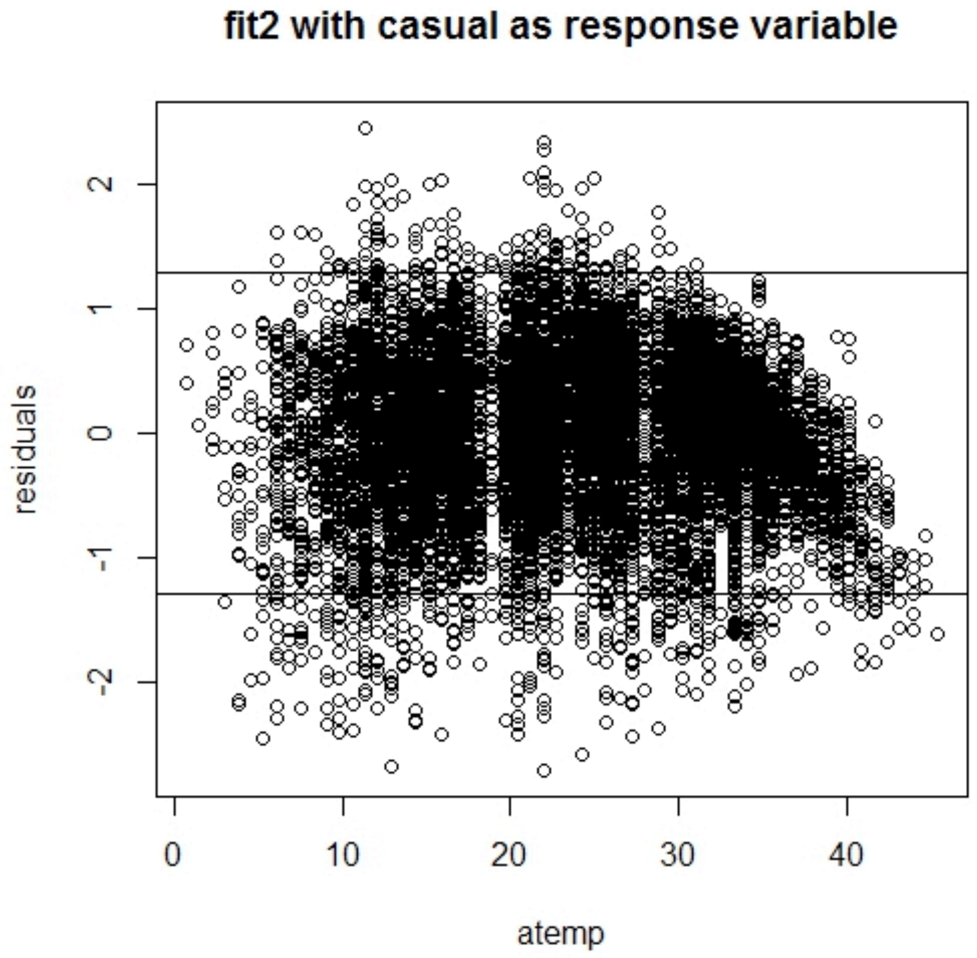


normal QQ plot of residuals,causal as response variab

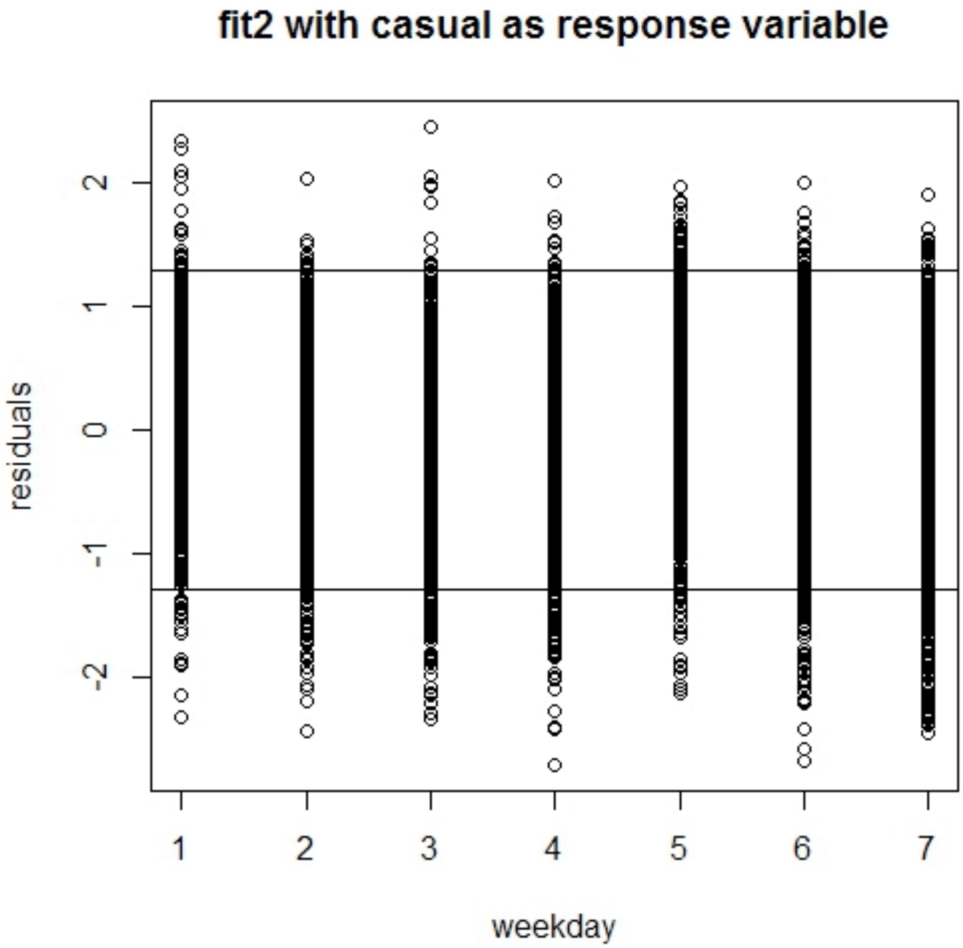
```
plot(pred2,res2,xlab = "predicted value",ylab = "residuals",main = "fit2.lm with casual
as response variable");abline(h = 2*sigma2.lm);abline(h = -2*sigma2.lm)
```



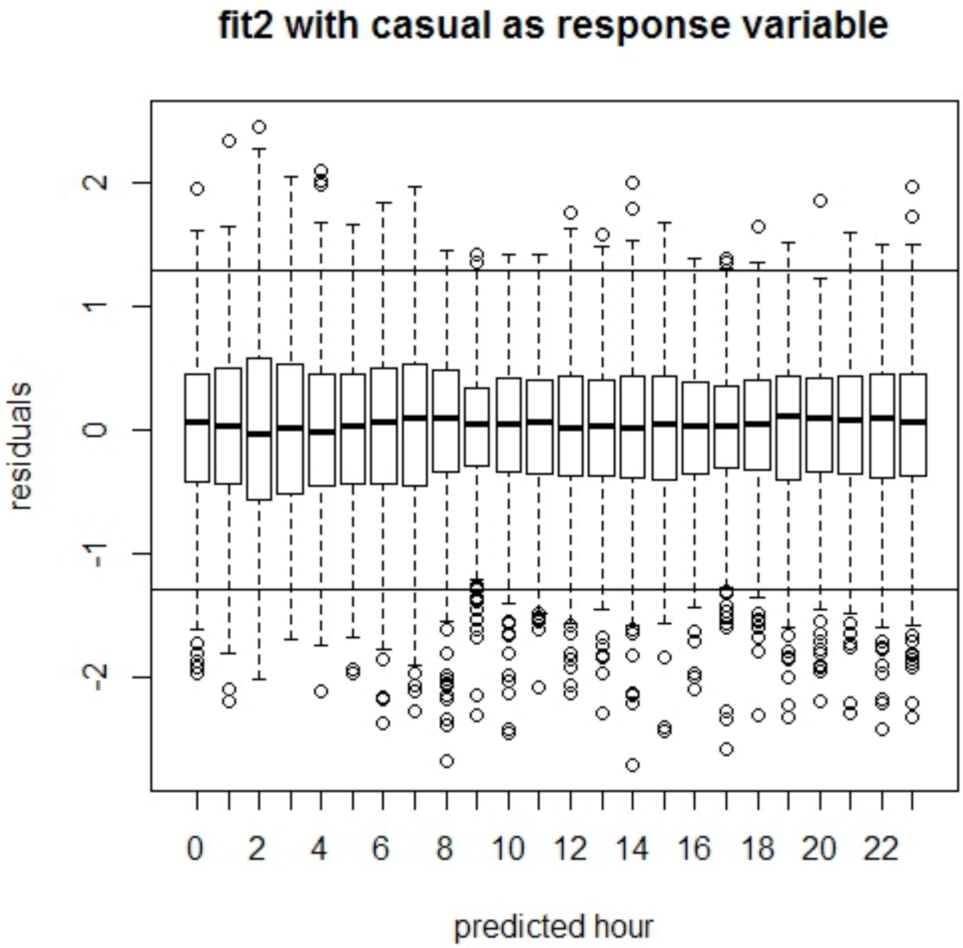
```
#plot for atemp
plot(atemp,res2,xlab = "atemp",ylab = "residuals",main = "fit2 with casual as response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



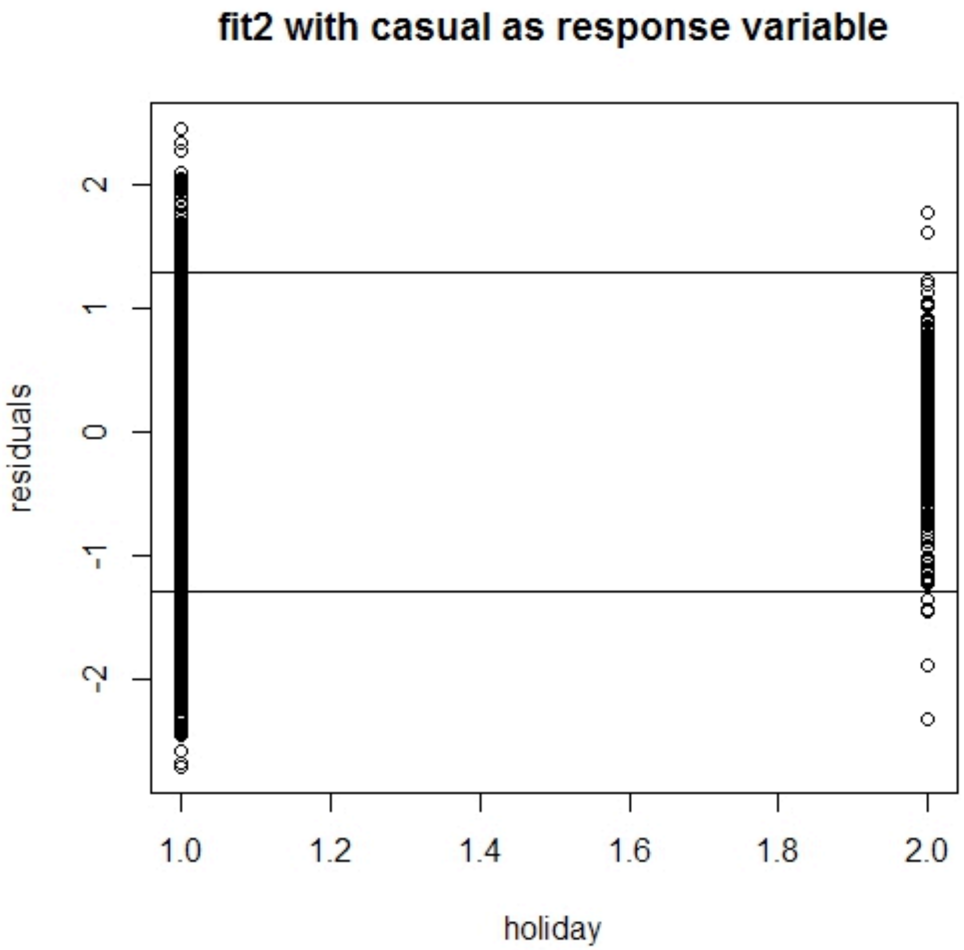
```
# plot for weekday
plot(weekday,res2,xlab = "weekday",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



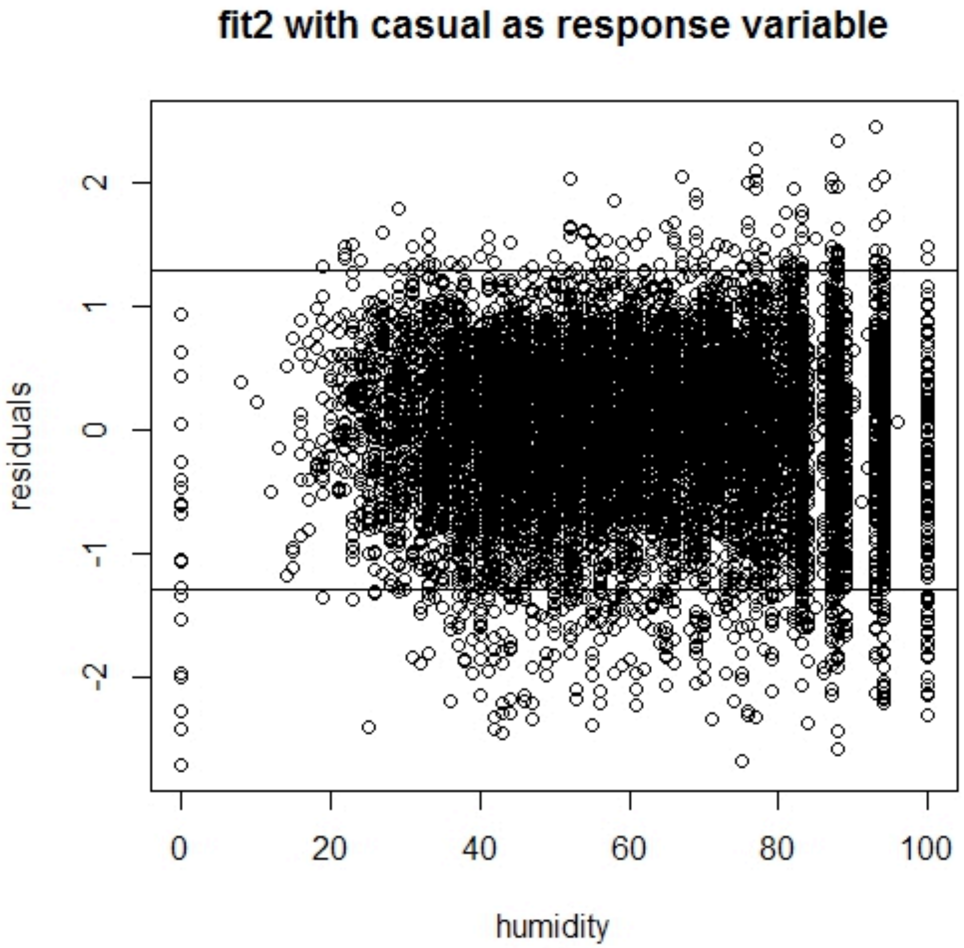
```
# plot for hour
plot(hour,res2,xlab = "predicted hour",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



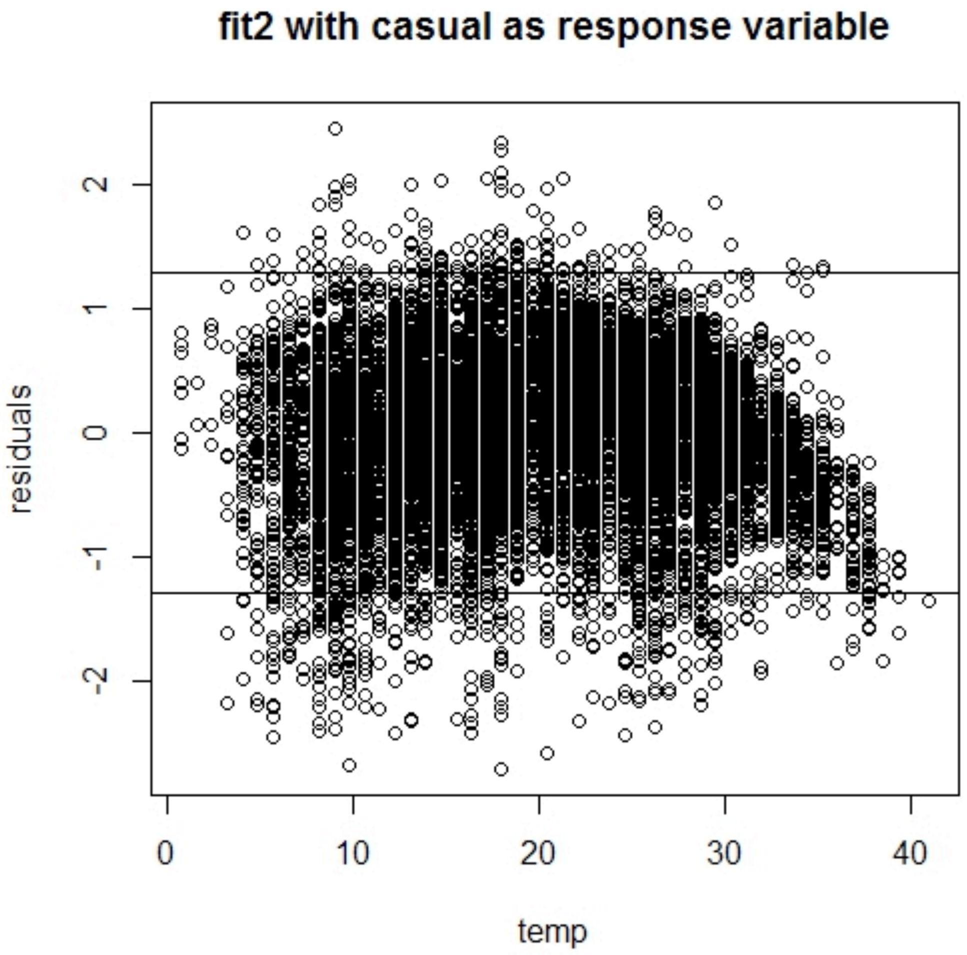
```
#plot for holiday
plot(holiday,res2,xlab = "holiday",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



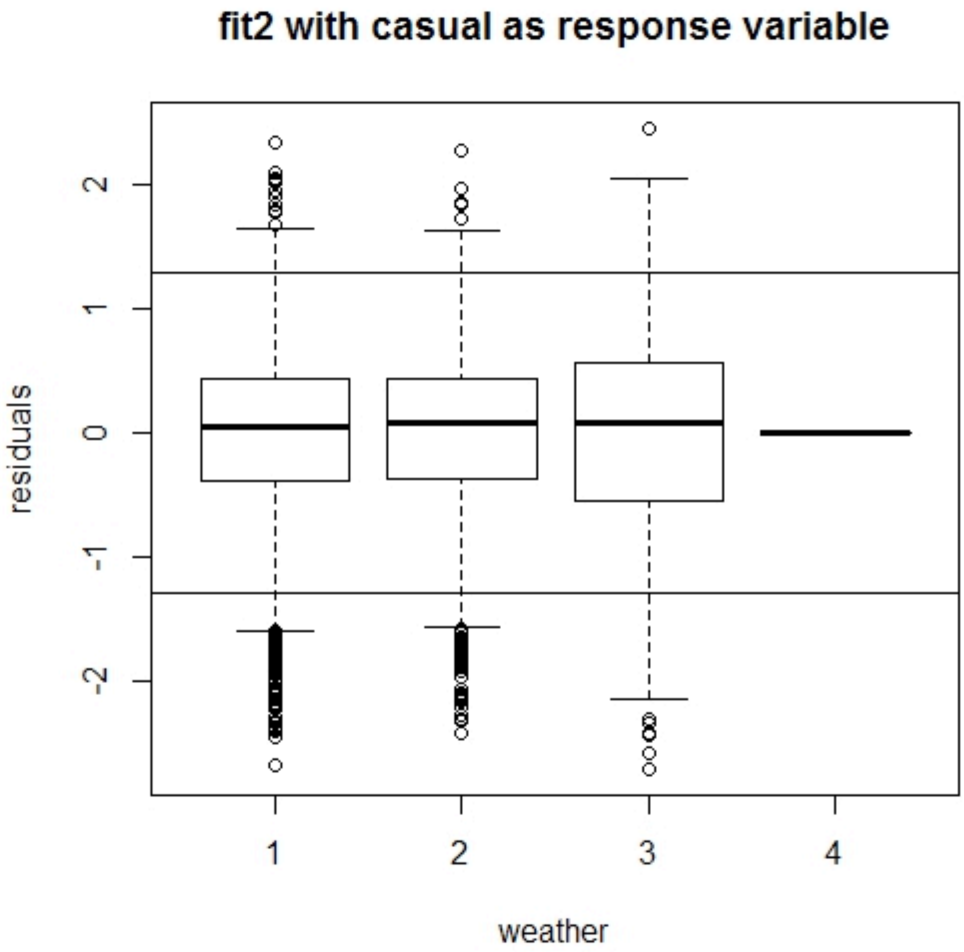
```
# plot for humidity
plot(humidity,res2,xlab = "humidity",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



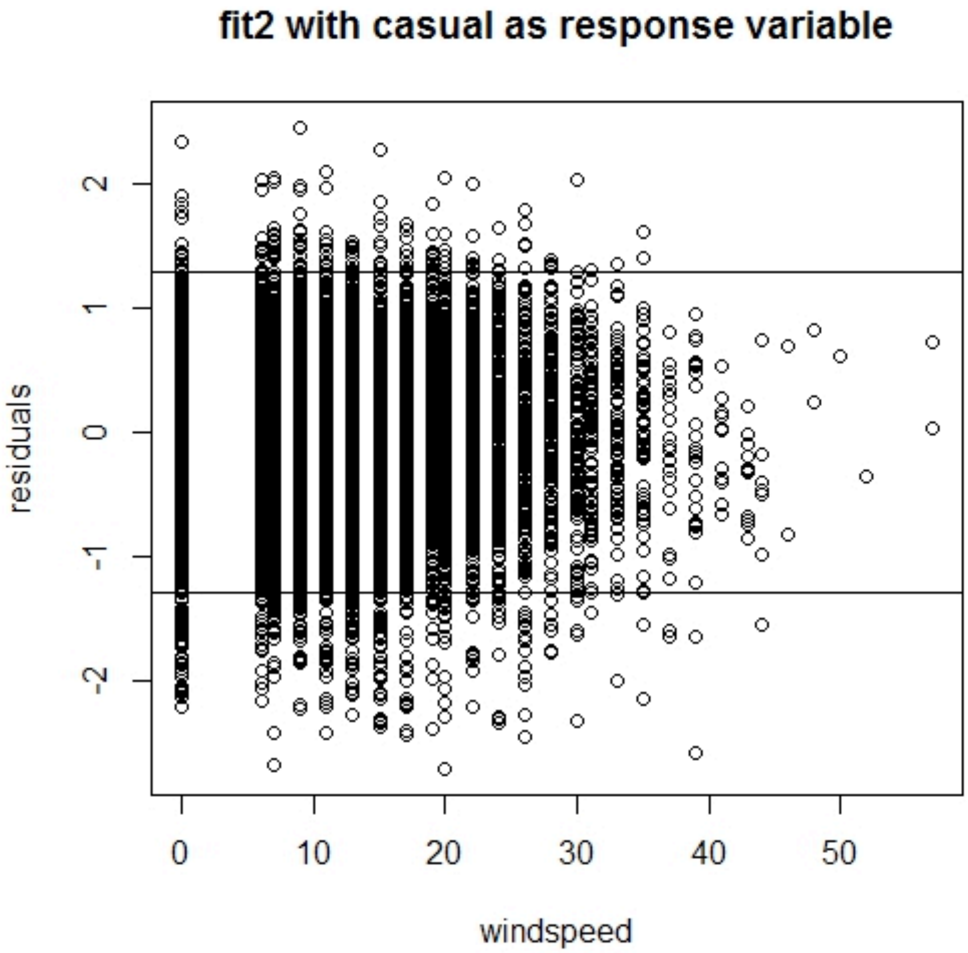
```
#plot for temp
plot(temp,res2,xlab = "temp",ylab = "residuals",main = "fit2 with casual as response
variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



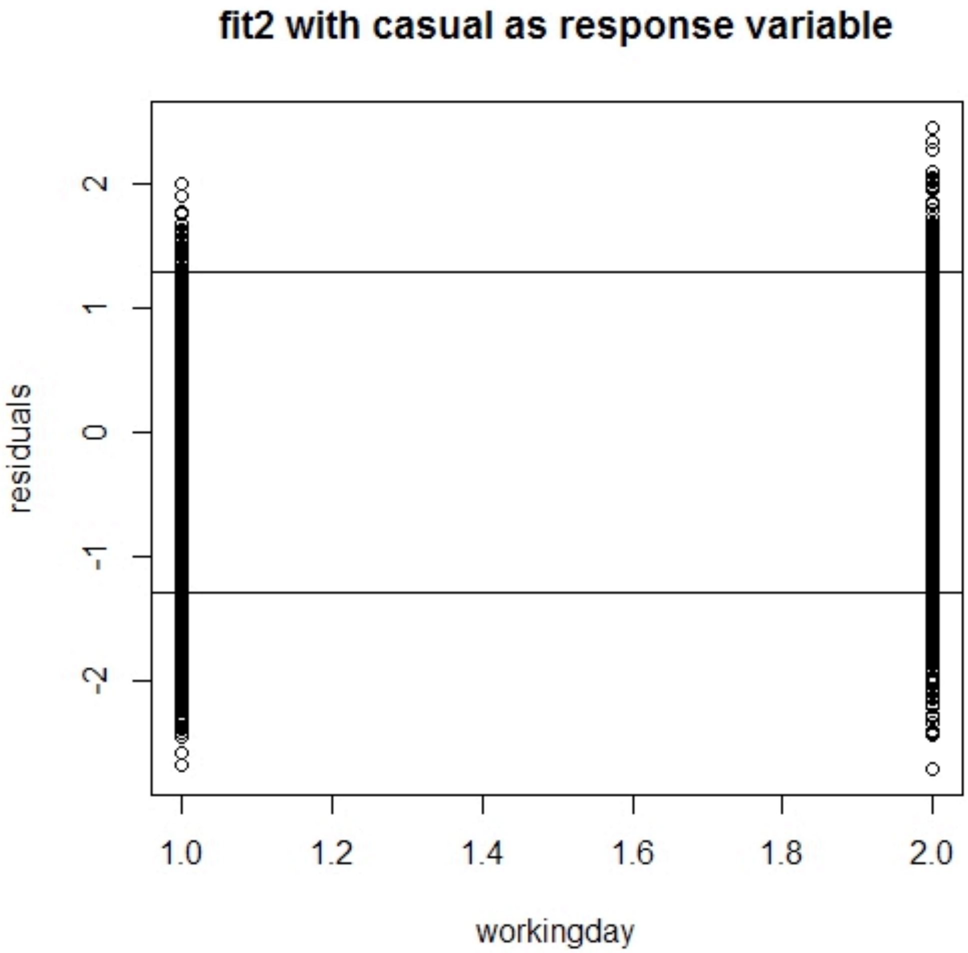
```
#plot for weather
plot(weather,res2,xlab = "weather",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



```
# plot for windspeed
plot(windspeed,res2,xlab = "windspeed",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



```
#plot for workingday
plot(workingday,res2,xlab = "workingday",ylab = "residuals",main = "fit2 with casual as
response variable");abline(h = 2*sigma2);abline(h = -2*sigma2)
```



```
train = read.csv("train.csv")
subtrain = featureEngineer(train)

#CROSS VALIDATION
set.seed(456)
n = nrow(subtrain)
iperm=sample(n,n)
subtrain.train = subtrain[iperm[1:10000],]
subtrain.holdout = subtrain[iperm[10001:n],]

#Linear model with response variable logged
fitMostTrain =
lm(I(log(registered+1))~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workin
gday, data=subtrain.train)
predMostHold = predict(fitMostTrain,new = subtrain.holdout)
rmseMostHold = sqrt(mean((log(subtrain.holdout$registered+1)-predMostHold)^2))
rmseMostHold
#[1] 0.6085023, quite small
```

```
# Linear model
fitMostTrain =
lm(registered~atemp+hour+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+workingday, data=subtrain.train)
predMostHold = predict(fitMostTrain,new = subtrain.holdout)
rmseMostHold = sqrt(mean((subtrain.holdout$registered-predMostHold)^2))
rmseMostHold
# [1] 83.41994
# LARGE!

#fitMostTrain=fit1.atemp.sqrt
#[1] 97.12997 #using fit in pink text
fitMostTrain.pois =
glm(registered~atemp+hour+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+workingday, data=subtrain.train, family = poisson)
predMostHold.pois = predict(fitMostTrain.pois, new = subtrain.holdout)
rmseMostHold.pois = sqrt(mean((subtrain.holdout$registered - predMostHold.pois)^2))
rmseMostHold.pois
# [1] 220.1969

fitMostTrain.atemp2 =
lm(registered~atemp+I(atemp^2)+hour+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+workingday, data=subtrain.train)
predMostHold.atemp2 = predict(fitMostTrain.atemp2, new = subtrain.holdout)
rmseMostHold.atemp2 = sqrt(mean((subtrain.holdout$registered - predMostHold.atemp2)^2))
rmseMostHold.atemp2
# [1] 83.41499

fitMostTrain.pois.atemp2 =
glm(registered~atemp+I(atemp^2)+hour+I(year==2012)+humidity+season+I(weather==2)+I(weather==3)+workingday, data=subtrain.train,family=poisson)
predMostHold.pois.atemp2 = predict(fitMostTrain.pois.atemp2, new = subtrain.holdout)
rmseMostHold.pois.atemp2 = sqrt(mean((subtrain.holdout$registered -
predMostHold.pois.atemp2)^2))
rmseMostHold.pois.atemp2
# [1] 220.1953

fitMostTrain.humidity =
lm(registered~I(atemp^2)+atemp+hour+I(year==2012)+I(log(humidity+1))+humidity+season+I(weather==2)+I(weather==3)+workingday,data=subtrain.train)
predMostHold.humidity = predict(fitMostTrain.humidity, new = subtrain.holdout)
rmseMostHold.humidity = sqrt(mean((subtrain.holdout$registered -
predMostHold.humidity)^2))
rmseMostHold.humidity
# [1] 83.26914
```

```
fitMostTrain.pois.humidity =
glm(registered~I(atemp^2)+atemp+hour+I(year==2012)+I(log(humidity+1))+humidity+season+I(weather==2)+I(weather==3)+workingday,data=subtrain.train,family=poisson)
predMostHold.pois.humidity = predict(fitMostTrain.pois.humidity, new = subtrain.holdout)
rmseMostHold.pois.humidity = sqrt(mean((subtrain.holdout$registered -
predMostHold.pois.humidity)^2))
rmseMostHold.pois.humidity
# [1]220.1953

fitMostTrain.log =
lm(I(log(registered+1))~atemp+hour+I(year==2012)+humidity+season+weather+windspeed+workingday, data=subtrain.train)
predMostHold.log = predict(fitMostTrain,new = subtrain.holdout)
predMostHold = exp(predMostHold.log)-1
rmseMostHold.log = sqrt(mean((subtrain.holdout$registered-predMostHold)^2))
rmseMostHold.log
# [1] 216.5318

#since variance of prediction in GLM varies based on covariates, we cannot directly
compare a linear model with the poisson model

# just rmse cannot verify best fit

fit1.lm
fit2.lm
```