4_3

2023-03-30

```
library(car)
```

```
## Loading required package: carData
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and

##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(GGally)
```

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

```
library(gridExtra)
```

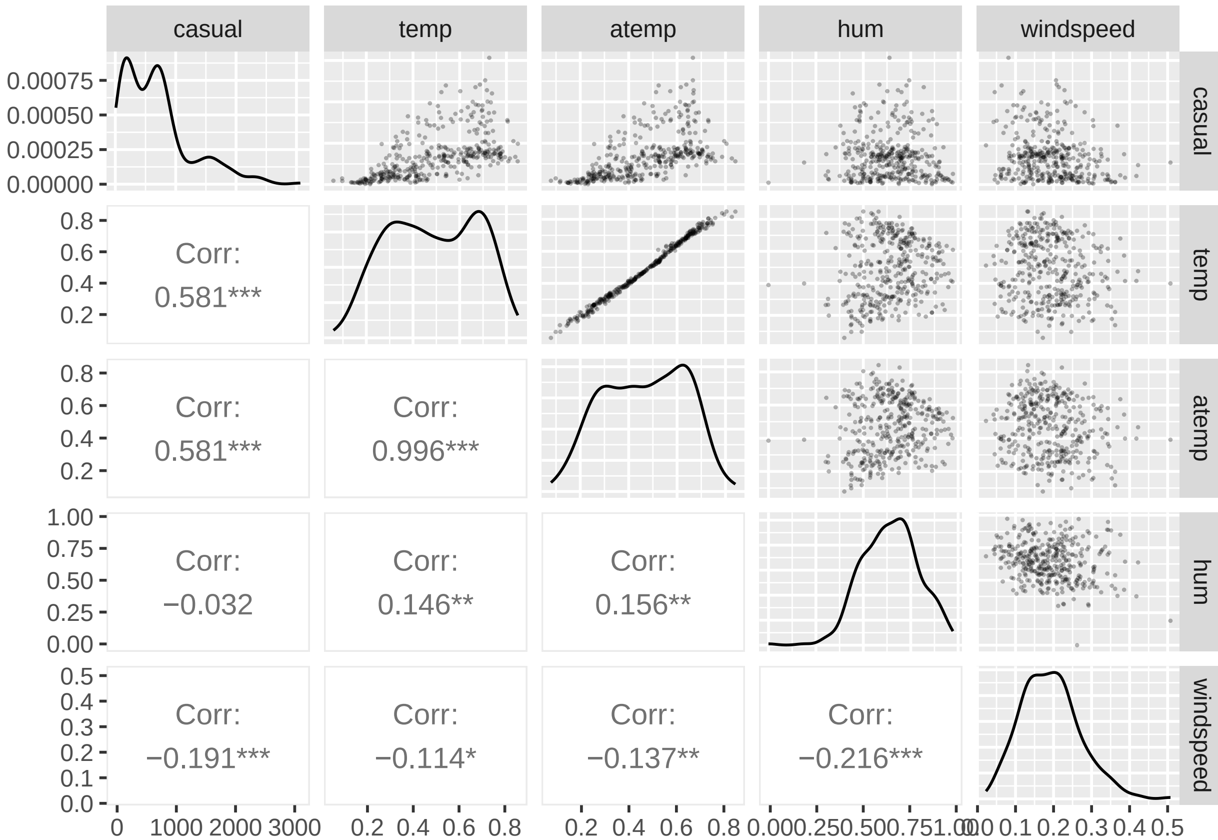
```
##
## Attaching package: 'gridExtra'

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

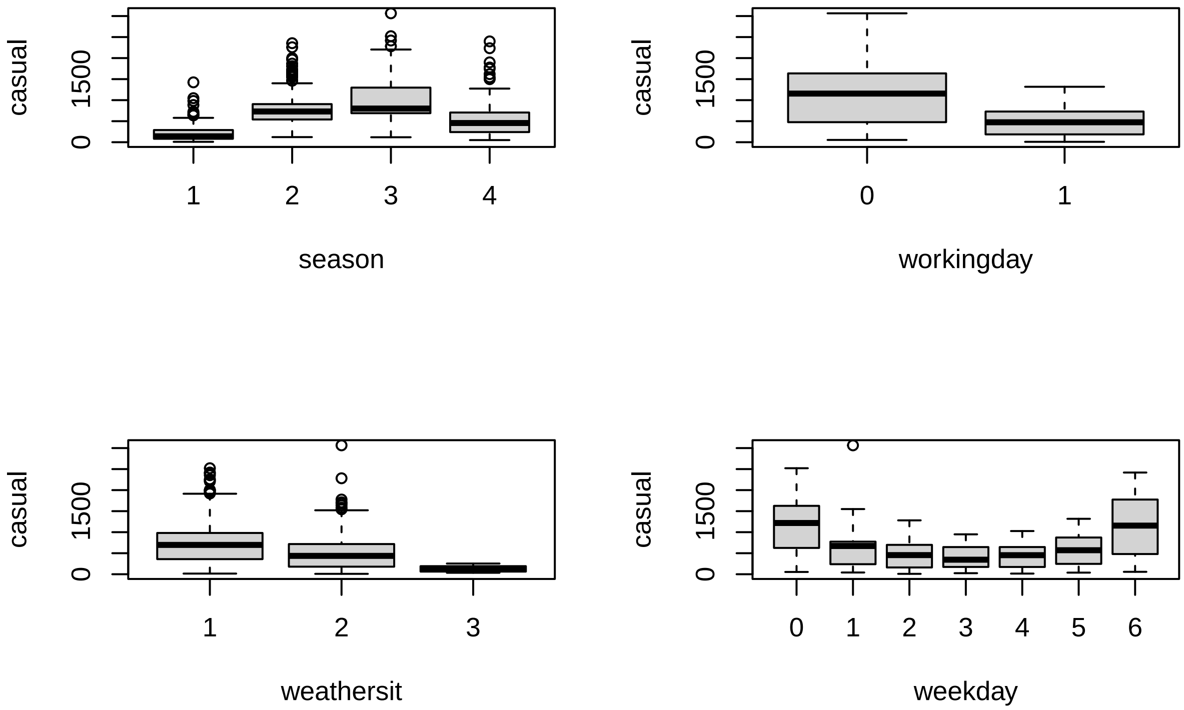
1. Model selection

```
daydata <- read.csv("/Users/bach_nguyen/MA 575/Labs/Project/Data/day.csv",header=TRUE)
df = data.frame(daydata)
train = df[df$yr == '0',]
validation = df[df$yr == '1',]
attach(train)
```

```r
data <- data.frame(casual, temp, atemp, hum, windspeed)
ggpairs(data, upper = list(continuous = wrap("points", alpha = 0.3,     size=0.1)),
lower = list(continuous = wrap('cor', size = 4)))
```




```r
par(mfrow=c(2,2))
boxplot(casual~as.factor(season),ylab="casual", xlab="season")
boxplot(casual~as.factor(workingday),ylab="casual", xlab="workingday")
boxplot(casual~as.factor(weathersit),ylab="casual", xlab="weathersit")
boxplot(casual~as.factor(weekday),ylab="casual", xlab="weekday")
```



first try:

```r
m1 = lm(casual ~ temp + atemp + hum + windspeed)
summary(m1)
```

```
##
## Call:
## lm(formula = casual ~ temp + atemp + hum + windspeed)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -843.8 -261.3 -118.6  101.1 1850.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    418.0      157.6   2.653 0.008326 **
## temp          1621.2     1510.0   1.074 0.283706
## atemp          108.4     1701.8   0.064 0.949245
## hum           -566.7      161.3  -3.512 0.000500 ***
## windspeed    -1125.8      320.4  -3.514 0.000499 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 442.3 on 360 degrees of freedom
## Multiple R-squared:  0.3747, Adjusted R-squared:  0.3678
## F-statistic: 53.94 on 4 and 360 DF,  p-value: < 2.2e-16
```

```r
vif(m1)
```

```
##       temp      atemp        hum  windspeed
## 152.509345 153.609171   1.071621   1.129341
```

-> From the multi collinearity plot, we can see temp and atemp are highly correlated. And temp is more significant, keep temp

```
summary(aov(casual~season, train))
```

```
##              Df    Sum Sq Mean Sq F value   Pr(>F)    
## season        1   7071501 7071501   24.32 1.25e-06 ***
## Residuals   363 105562961  290807                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

-> season is significant to add to the model

```
summary(aov(casual~workingday, train))
```

```
##              Df   Sum Sq  Mean Sq F value Pr(>F)    
## workingday    1 33017099 33017099   150.5 <2e-16 ***
## Residuals   363 79617363   219332                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

-> workingday is significant to add to the model

```
summary(aov(casual~weekday, train))
```

```
##              Df    Sum Sq Mean Sq F value Pr(>F)
## weekday       1     43286   43286    0.14  0.709
## Residuals   363 112591176  310169
```

-> weekday is not significant

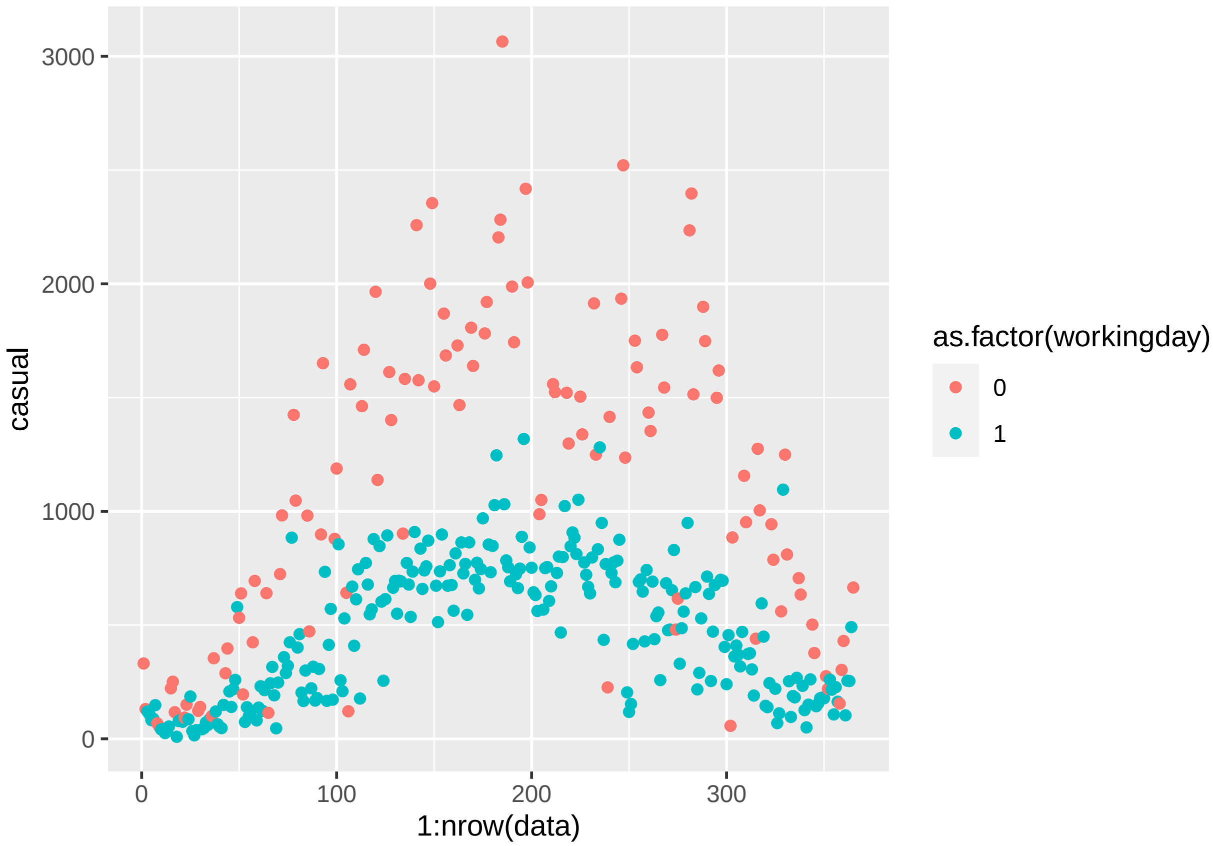
```
summary(aov(casual~holiday, train))
```

```
##              Df    Sum Sq Mean Sq F value Pr(>F)  
## holiday       1    909365  909365   2.955 0.0865 .
## Residuals   363 111725096  307783                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

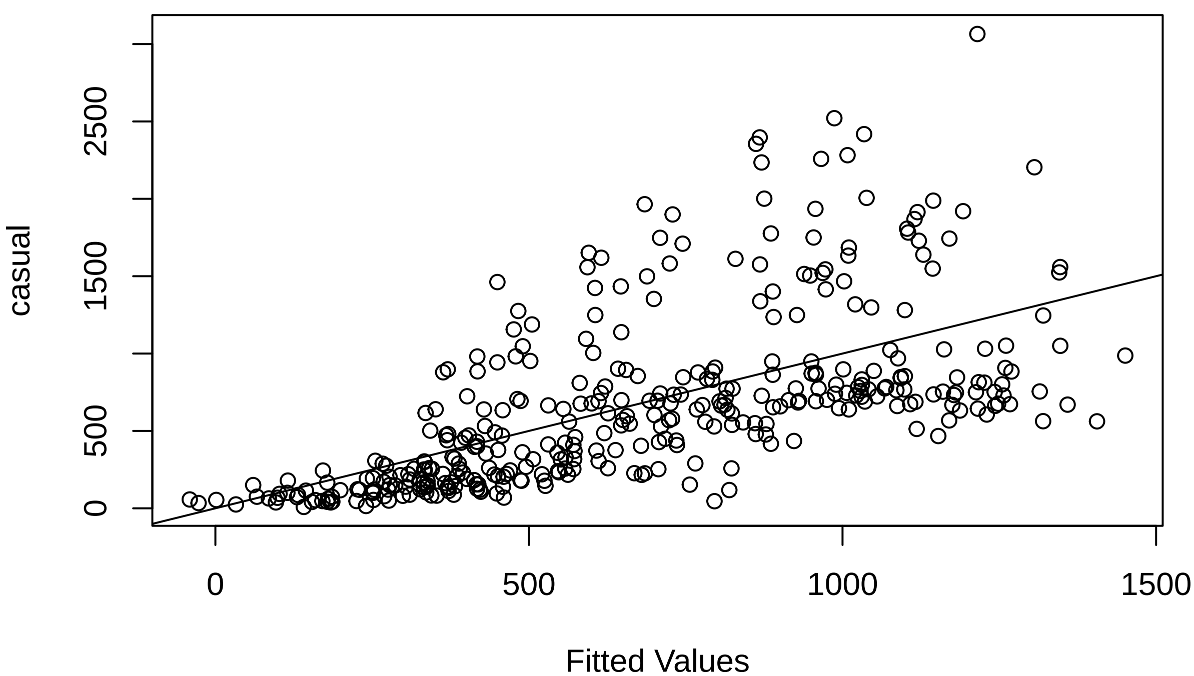
->holiday is not significant

from the boxplot and below scatter plot, non-working days have higher bike rents.

```
ggplot(data = train) + geom_point(data=train, aes(x=1:nrow(data), y=casual, color=as.factor(workingday))
```

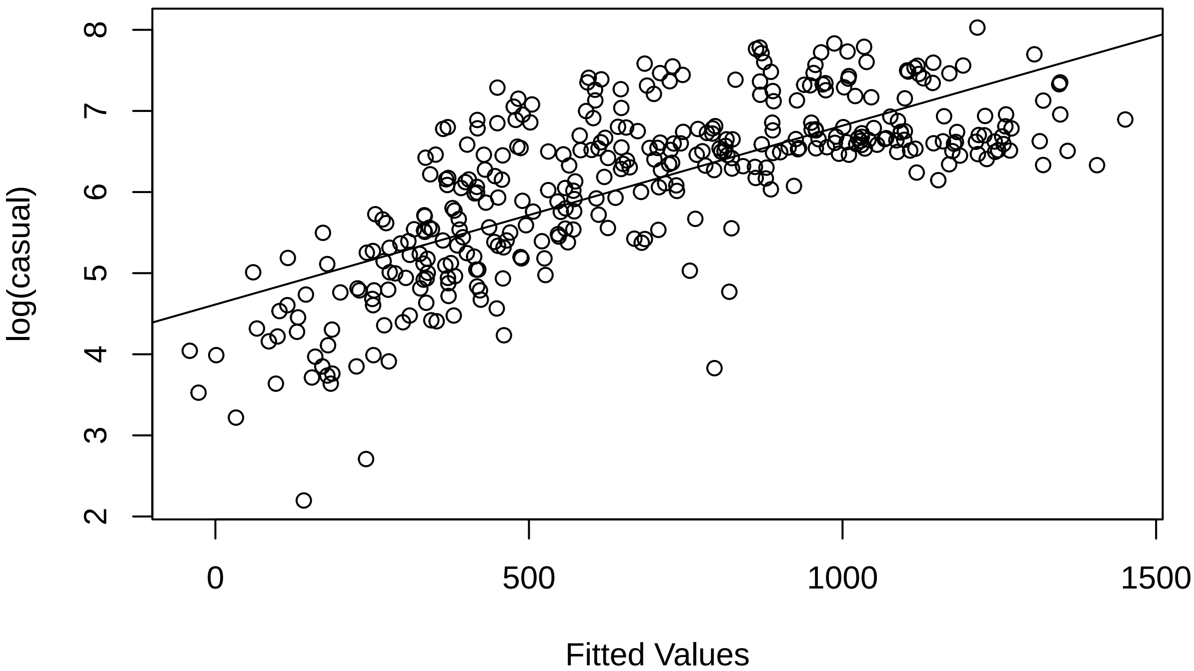



```r
plot(m1$fitted.values,casual,xlab="Fitted Values")
abline(lsfit(m1$fitted.values,casual))
```



-> Use log(casual)

```r
plot(m1$fitted.values,log(casual),xlab="Fitted Values")
abline(lsfit(m1$fitted.values,log(casual)))
```



-> fit better

2. Choosing MLR

```r
m = lm(log(casual) ~ temp + I(temp^2) + hum + windspeed + as.factor(season)+ as.factor(weathersit)+ as.f
summary(m)
```

```
##
## Call:
## lm(formula = log(casual) ~ temp + I(temp^2) + hum + windspeed +
##     as.factor(season) + as.factor(weathersit) + as.factor(workingday))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86271 -0.19973 -0.01298  0.24217  1.26307
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.12815    0.17279  23.891  < 2e-16 ***
## temp                    12.56256    0.74254  16.918  < 2e-16 ***
## I(temp^2)               -9.86823    0.76051 -12.976  < 2e-16 ***
## hum                     -1.16283    0.19130  -6.078 3.14e-09 ***
## windspeed               -1.72473    0.29274  -5.892 8.90e-09 ***
## as.factor(season)2       0.53244    0.07768   6.854 3.20e-11 ***
## as.factor(season)3       0.49733    0.09996   4.975 1.02e-06 ***
## as.factor(season)4       0.32262    0.07159   4.507 8.96e-06 ***
## as.factor(weathersit)2 -0.21040    0.05368  -3.920 0.000106 ***
## as.factor(weathersit)3 -1.09211    0.11927  -9.157  < 2e-16 ***
## as.factor(workingday)1 -0.86134    0.04384 -19.646  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3854 on 354 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8596
## F-statistic: 223.8 on 10 and 354 DF,  p-value: < 2.2e-16
```

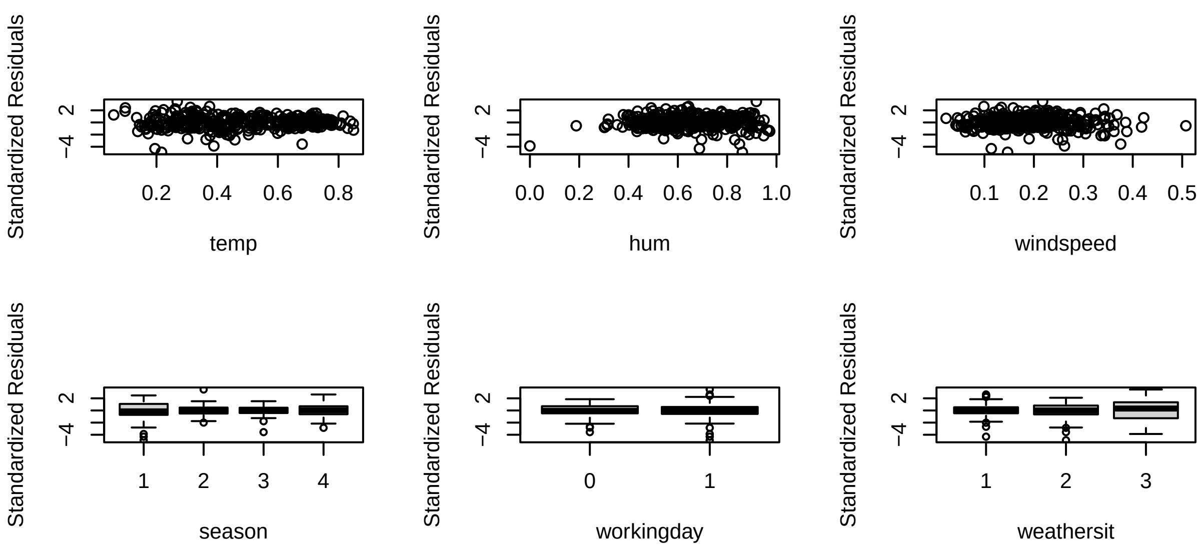
```
m1_with_log <- lm(log(casual) ~ temp + hum + windspeed)
anova(m1_with_log, m)
```

```
## Analysis of Variance Table
##
## Model 1: log(casual) ~ temp + hum + windspeed
## Model 2: log(casual) ~ temp + I(temp^2) + hum + windspeed + as.factor(season) +
##     as.factor(weathersit) + as.factor(workingday)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    361 179.257
## 2    354  52.587  7    126.67 121.82 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

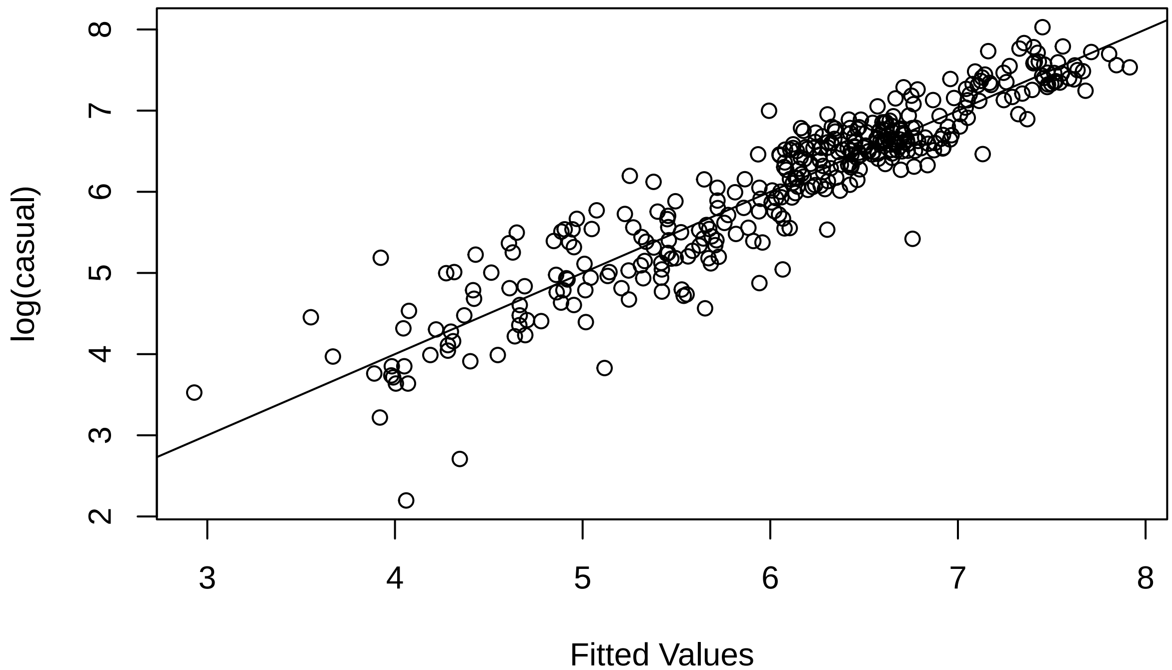
-> there is statistically evidence to use full model.

Begin the diagnostics:

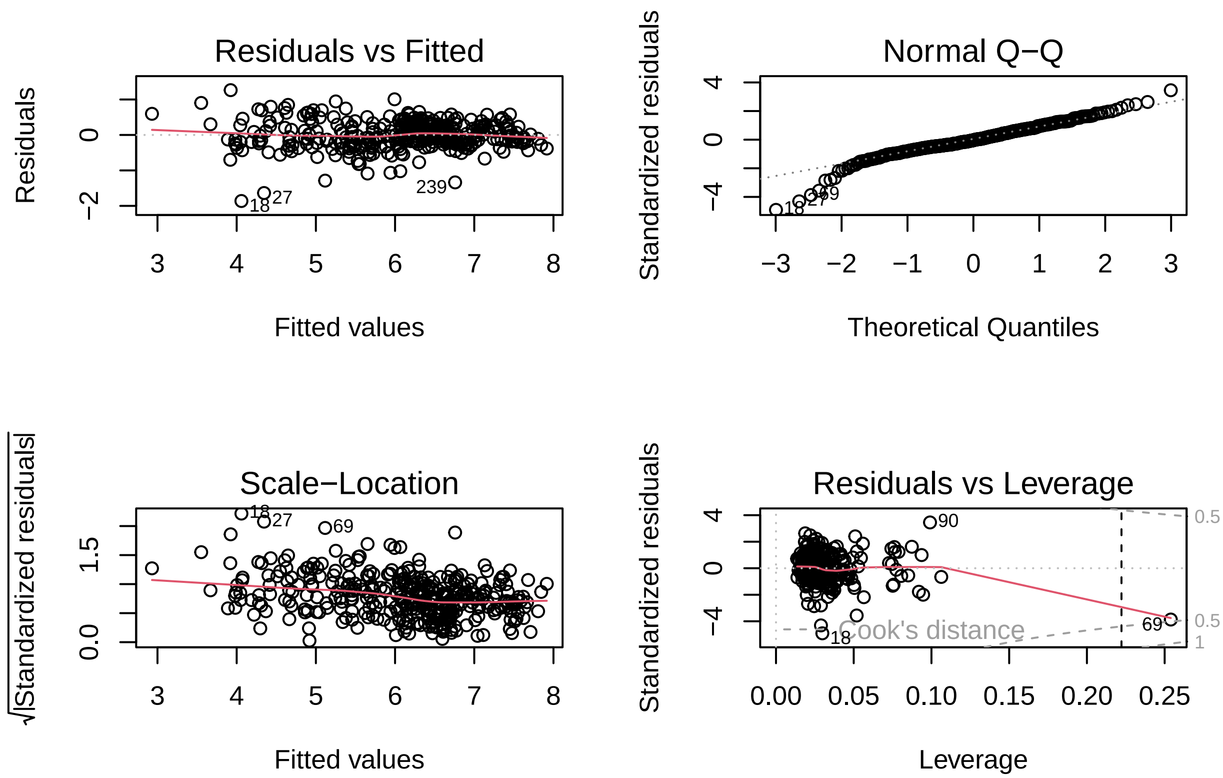
```
StanRes <- rstandard(m)
par(mfrow=c(3,3))
plot(temp,StanRes, ylab="Standardized Residuals")
plot(hum,StanRes, ylab="Standardized Residuals")
plot(windspeed,StanRes, ylab="Standardized Residuals")
boxplot(StanRes~as.factor(season),ylab="Standardized Residuals",xlab="season")
boxplot(StanRes~as.factor(workingday),ylab="Standardized Residuals",xlab="workingday")
boxplot(StanRes~as.factor(weathersit),ylab="Standardized Residuals",xlab="weathersit")
```




```
plot(m$fitted.values,log(casual),xlab="Fitted Values")
abline(lsfit(m$fitted.values,log(casual)))
```




```
par(mfrow=c(2,2))
plot(m)
abline(v=2*8/72,lty=2)
```

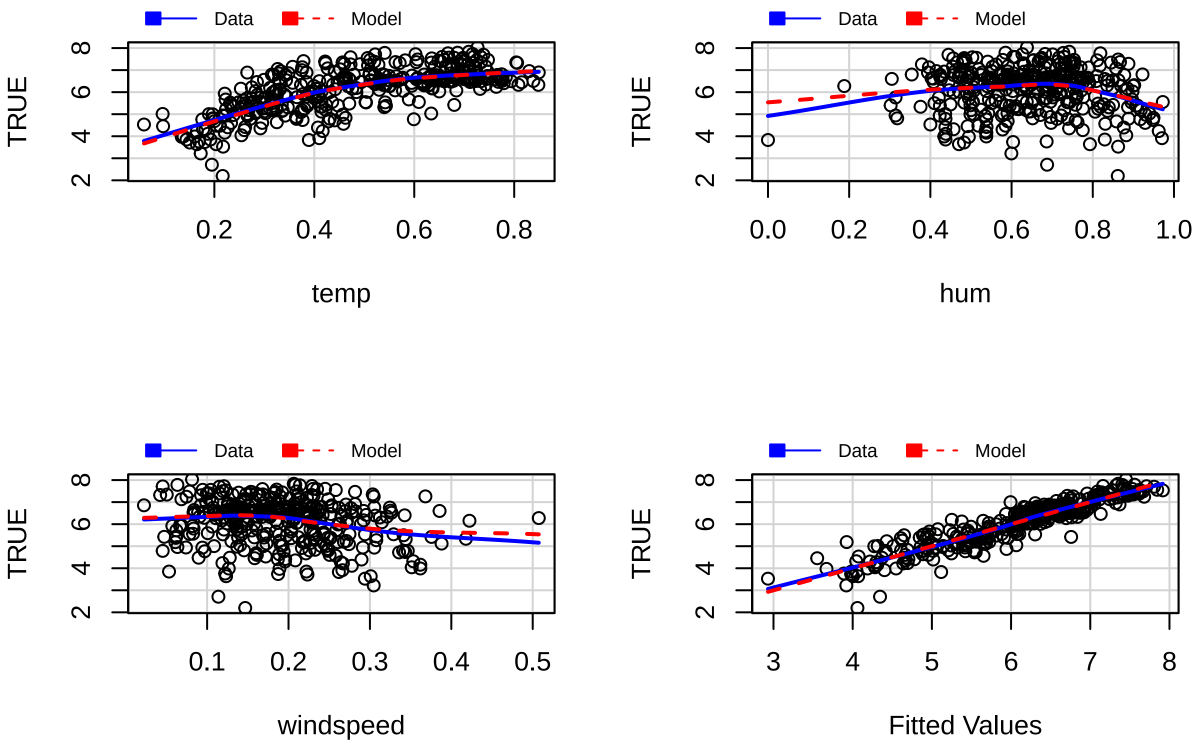



```
vif(m)
```

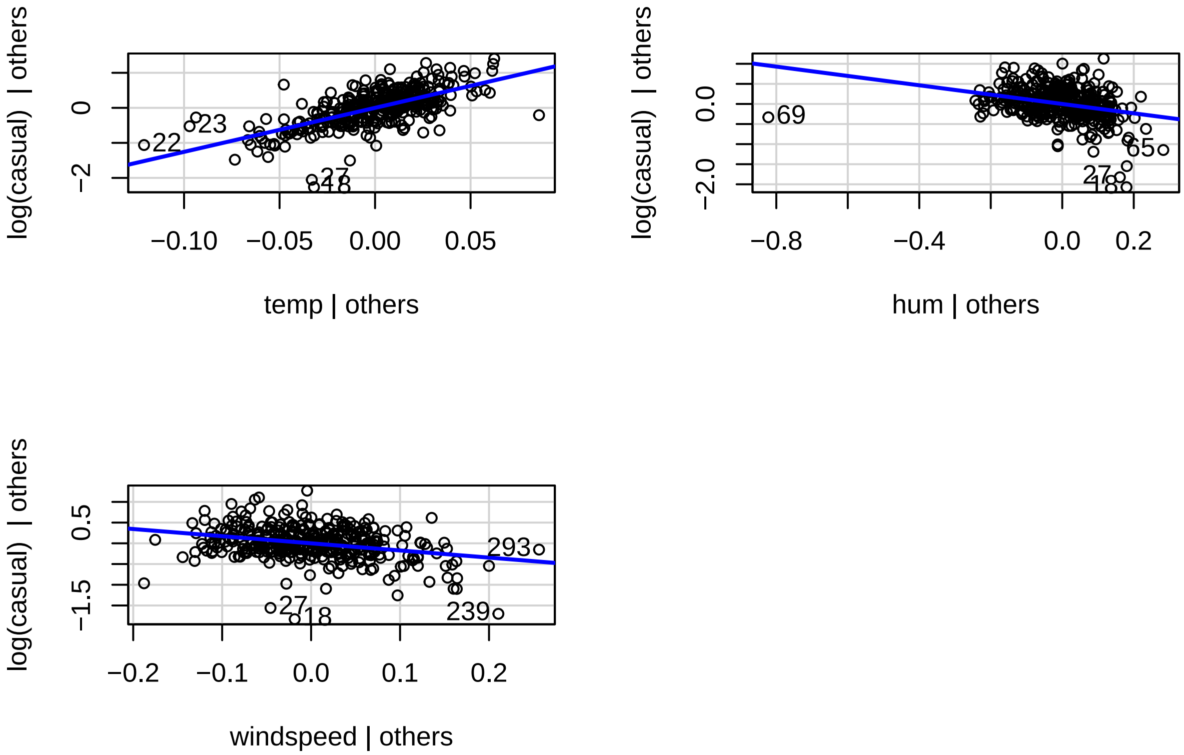
```
##                               GVIF Df GVIF^(1/(2*Df))
## temp                     48.564918  1        6.968853
```

```
## I(temp^2)              48.739860  1        6.981394
## hum                     1.984080  1        1.408574
## windspeed               1.241480  1        1.114217
## as.factor(season)       4.924616  3        1.304354
## as.factor(weathersit)   1.820096  2        1.161512
## as.factor(workingday)   1.019189  1        1.009549
```

```
par(mfrow=c(2,2))
mmp(m,temp)
mmp(m,hum)
mmp(m,windspeed)
mmp(m,m$fitted.values,xlab="Fitted Values")
```




```
library(car)
par(mfrow=c(2,2))
avPlot(m,variable=temp,ask=FALSE, main="")
avPlot(m,variable=hum,ask=FALSE, main="")
avPlot(m,variable=windspeed,ask=FALSE, main="")
```



Validation

```
# Residuals for training data
ResMLS <- resid(m)

# Mean Square Error for training data
mean((ResMLS)^2)
```

```
## [1] 0.1440728
```

```
# Mean Square Error for validation data

# Residuals for validation data
#If the logical se. fit is TRUE , standard errors of the predictions are also calculated.
new_data <- data.frame(temp=validation$temp,hum=validation$hum, windspeed =validation$windspeed, season

output <- predict(m,se.fit = TRUE, newdata = new_data)

ResMLSValidation <- log(validation$casual) - output$fit

mean((ResMLSValidation)^2)
```

```
## [1] 0.2801909
```

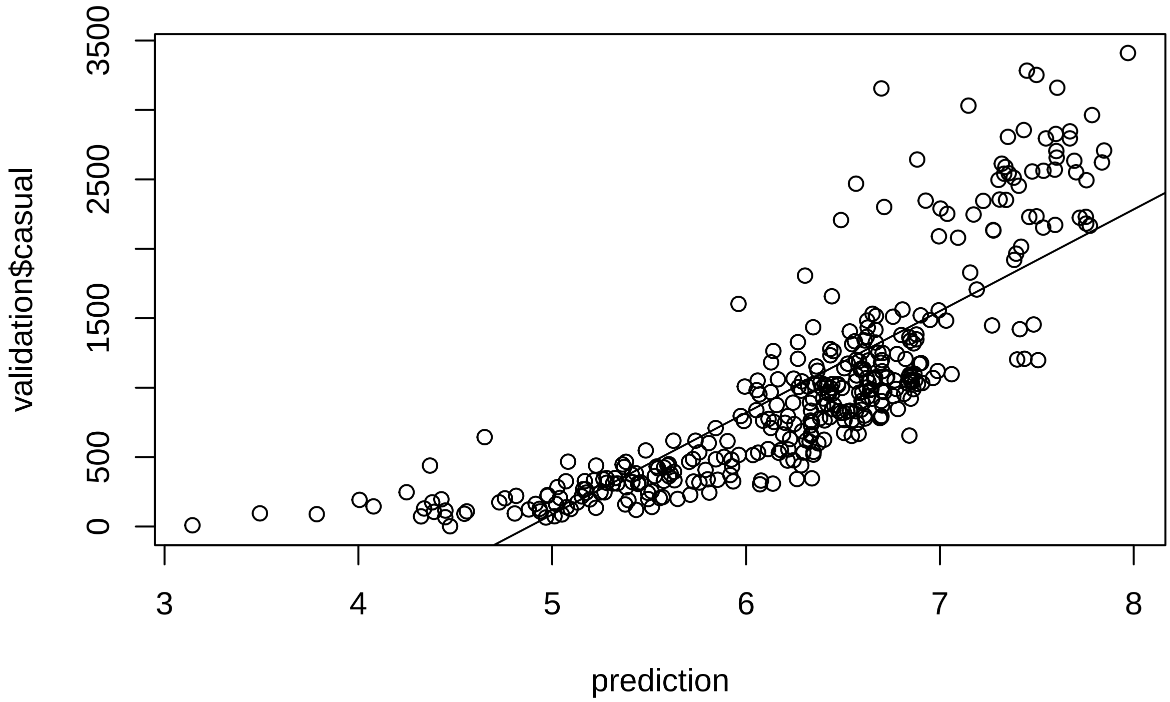
```
#mean((output$residual.scale)^2)
```

```
# Relative Mean Square Error for validation data
mean((ResMLSValidation)^2) / mean((validation$casual)^2)
```

```
## [1] 1.738372e-07
```

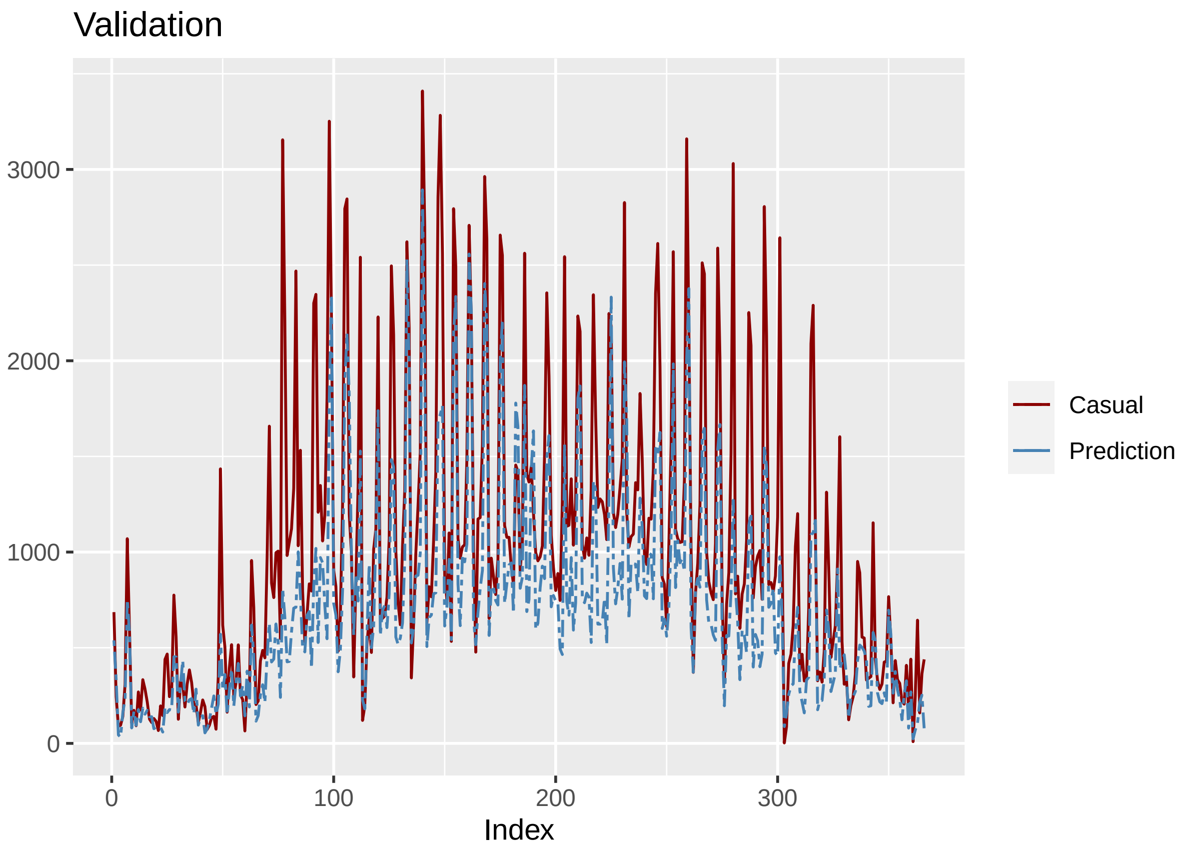
```
#mean((output$residual.scale)^2) / mean((validation$casual)^2)
```

```
plot(output$fit,validation$casual,xlab="prediction")
abline(lsfit(output$fit,validation$casual))
```




```
# Create data frame with validation observation and prediction
test = data.frame(validation$casual,exp(output$fit), 1:length(output$fit));
colnames(test)[1] = "Casual"
colnames(test)[2] = "Prediction"
colnames(test)[3] = "Index"
```

```
ggplot(data = test, aes(x = Index)) +
  geom_line(aes(y = Casual, color = "Casual")) +
  geom_line(aes(y = Prediction, color="Prediction"), linetype="twodash") +
  scale_color_manual(name = element_blank(), labels = c("Casual","Prediction"),
                     values = c("darkred", "steelblue")) + labs(y = "") +
  ggtitle("Validation")
```


Validation
3000
2000
1000
0
0
100
200
300
Index
Casual
Prediction