GLM data analysis

Jiahui Xin

2022/6/3

Contents

Abstract and Prerequisites

This report mainly contain three parts: *data description*, *data preparation* and *data analysis and diagnostic test*. The analysis results are in *data analysis and diagnostic test*.

Due to complex structure of my dataset, data preparation and reconstruction costs some space.

```
library(MASS)
```

Data Description

Data Source

```
bike<-read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/00560/SeoulBikeData.csv")
```

Seoul Bike Sharing Demand Data Set from UCI Machine Learning Repository (page link is here)

The following information is cited from the website. The goal is to predict the bike demand using other variables.

Data Abstract

The dataset contains count of public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information

- Data Set Characteristics: Multivariate
- Number of Instances: 8760
- Attribute Characteristics: Integer, Real
- Number of Attributes: 14
- Associated Tasks: Regression

...

Data Preparation

```
names(bike)[4:11]<-c("Temperature","Humidity","Wind.speed",
                     "Visibility","Dew.point.temperature",
                     "Solar.Radiation","Rainfall","Snowfall")
bike$Hour=as.factor(bike$Hour)
bike$Seasons=as.factor((bike$Seasons))
bike$Holiday=as.factor(bike$Holiday)
bike$Functioning.Day=as.factor(bike$Functioning.Day)
summary(bike)
```

```
##      Date           Rented.Bike.Count      Hour       Temperature
##  Length:8760        Min.   :   0.0    0      : 365   Min.   :-17.80
##  Class :character   1st Qu.: 191.0    1      : 365   1st Qu.:  3.50
##  Mode  :character   Median : 504.5    2      : 365   Median : 13.70
##                     Mean   : 704.6    3      : 365   Mean   : 12.88
##                     3rd Qu.:1065.2    4      : 365   3rd Qu.: 22.50
##                     Max.   :3556.0    5      : 365   Max.   : 39.40
##                                       (Other):6570
##     Humidity       Wind.speed      Visibility   Dew.point.temperature
##  Min.   : 0.00   Min.   :0.000   Min.   :  27   Min.   :-30.600
##  1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 940   1st Qu.: -4.700
##  Median :57.00   Median :1.500   Median :1698   Median :  5.100
##  Mean   :58.23   Mean   :1.725   Mean   :1437   Mean   :  4.074
##  3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 14.800
##  Max.   :98.00   Max.   :7.400   Max.   :2000   Max.   : 27.200
##
##  Solar.Radiation     Rainfall          Snowfall         Seasons
```

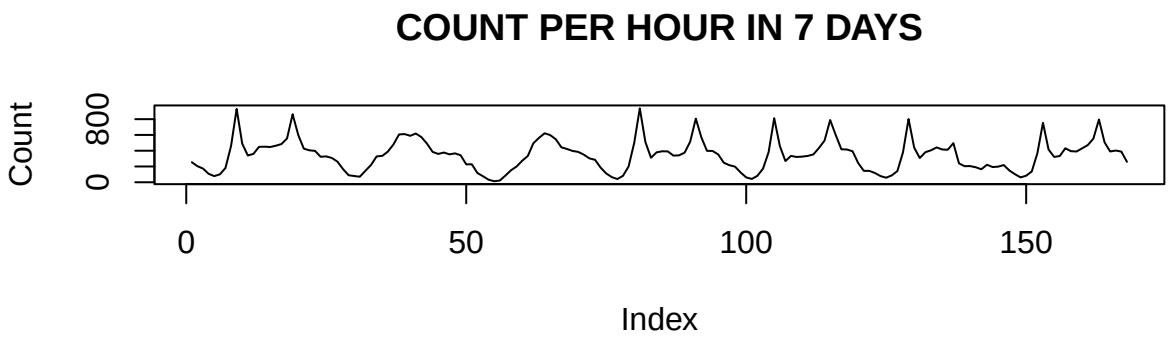
```
##  Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000   Autumn:2184
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   Spring:2208
##  Median :0.0100   Median : 0.0000   Median :0.00000   Summer:2208
##  Mean   :0.5691   Mean   : 0.1487   Mean   :0.07507   Winter:2160
##  3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :3.5200   Max.   :35.0000   Max.   :8.80000
##
##        Holiday     Functioning.Day
##  Holiday   : 432   No : 295
##  No Holiday:8328   Yes:8465
##
##
##
##
##
```

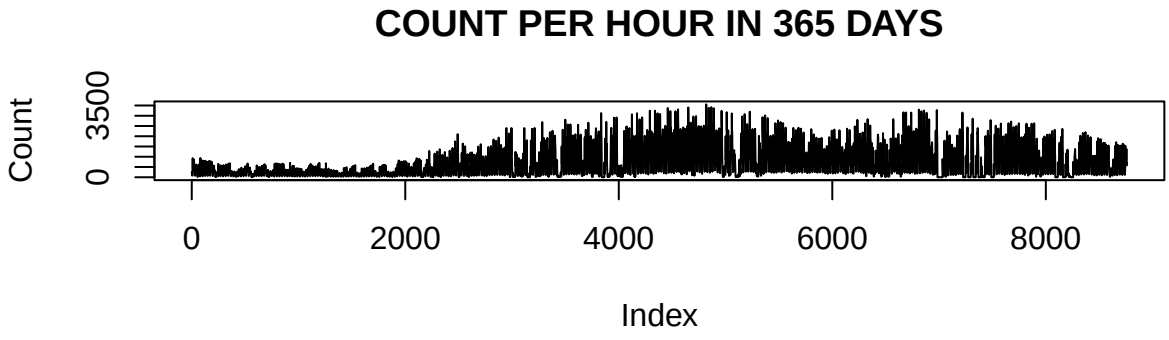
```
head(bike,3)
```

```
##         Date Rented.Bike.Count Hour Temperature Humidity Wind.speed Visibility
## 1 01/12/2017               254    0        -5.2       37        2.2       2000
## 2 01/12/2017               204    1        -5.5       38        0.8       2000
## 3 01/12/2017               173    2        -6.0       39        1.0       2000
##   Dew.point.temperature Solar.Radiation Rainfall Snowfall Seasons    Holiday
## 1                 -17.6               0        0        0  Winter No Holiday
## 2                 -17.6               0        0        0  Winter No Holiday
## 3                 -17.7               0        0        0  Winter No Holiday
##   Functioning.Day
## 1             Yes
## 2             Yes
## 3             Yes
```

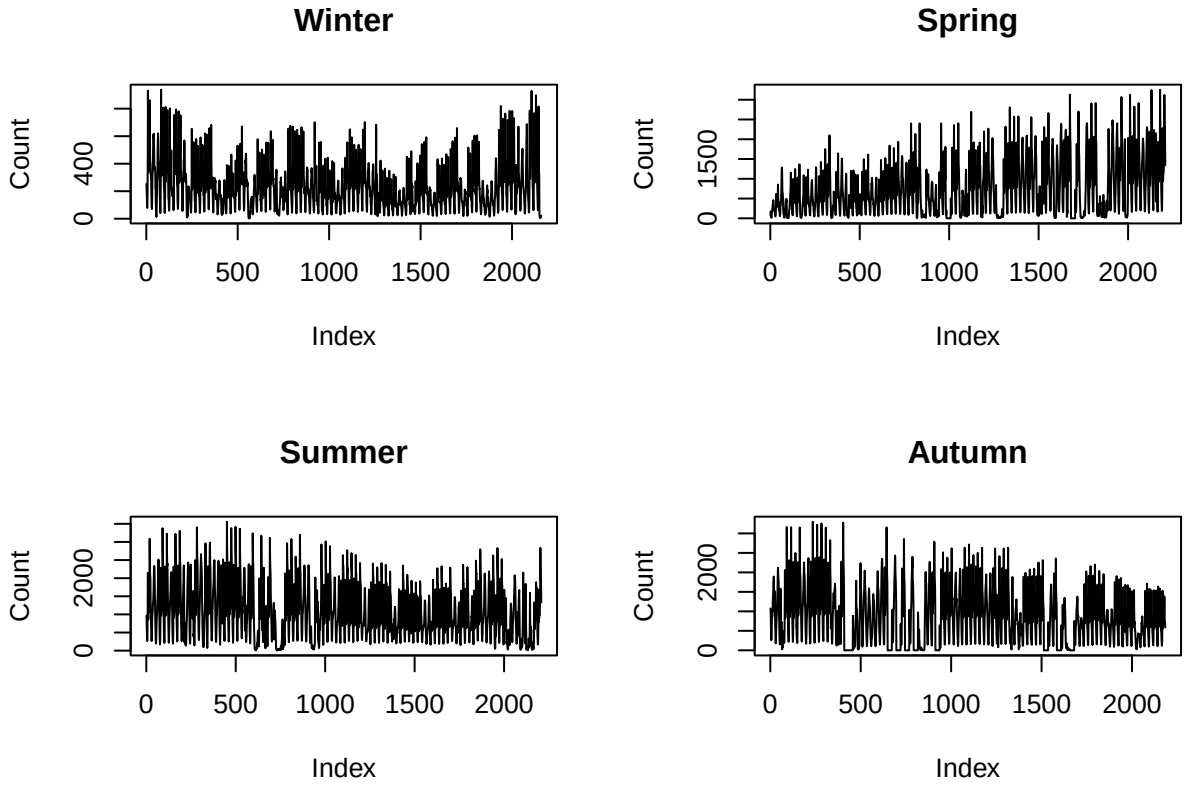
time series plot

```
par(mfrow=c(2,1))
with(bike,plot(Rented.Bike.Count[1:(24*7)],type="l",ylab="Count",main="COUNT PER HOUR IN 7 DAYS"))
with(bike,plot(Rented.Bike.Count[1:(24*365)],type="l",ylab="Count",main="COUNT PER HOUR IN 365 DAYS"))
```






```
par(mfrow=c(2,2))
with(bike,plot(Rented.Bike.Count[Seasons=="Winter"],type="l",ylab="Count",main="Winter"))
with(bike,plot(Rented.Bike.Count[Seasons=="Spring"],type="l",ylab="Count",main="Spring"))
with(bike,plot(Rented.Bike.Count[Seasons=="Summer"],type="l",ylab="Count",main="Summer"))
with(bike,plot(Rented.Bike.Count[Seasons=="Autumn"],type="l",ylab="Count",main="Autumn"))
```



Notice that number of unfunctioning observations 295 cannot be divided by 24 exactly. Find these specific unfunctioning days.

```
summary(as.factor(with(bike,Date[Functioning.Day=="No"])))
```

```
## 02/10/2018 03/11/2018 04/10/2018 06/10/2018 06/11/2018 09/10/2018 09/11/2018
##         24         24         24          7         24         24         24
## 10/05/2018 11/04/2018 18/09/2018 19/09/2018 28/09/2018 30/09/2018
##         24         24         24         24         24         24
```

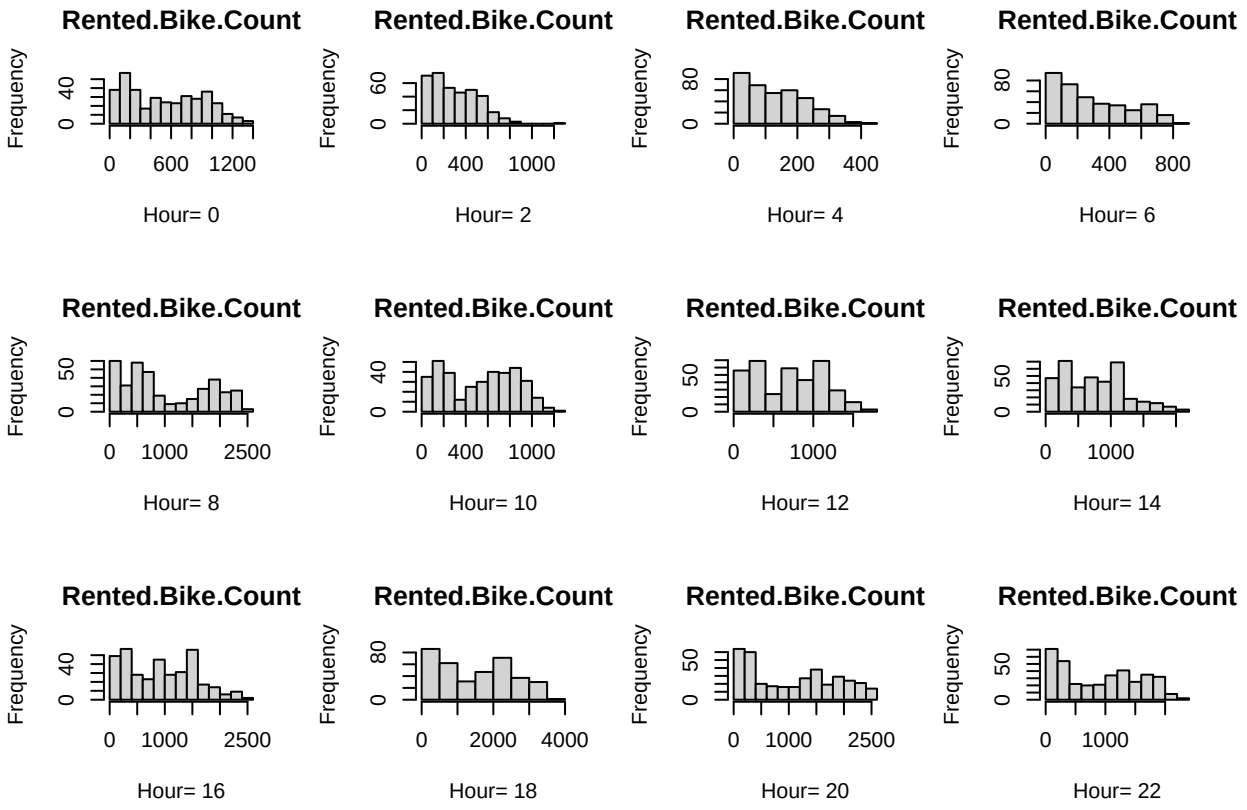
```
#with(bike,bike[Date=="06/10/2018",])
with(bike,Rented.Bike.Count [Date=="06/10/2018"])
```

```
##  [1]    0    0    0    0    0    0    0    5    6    5    2   20  130  529  941
## [16] 1236 1601 1870 2012 1761 1567 1731 1459 1162
```

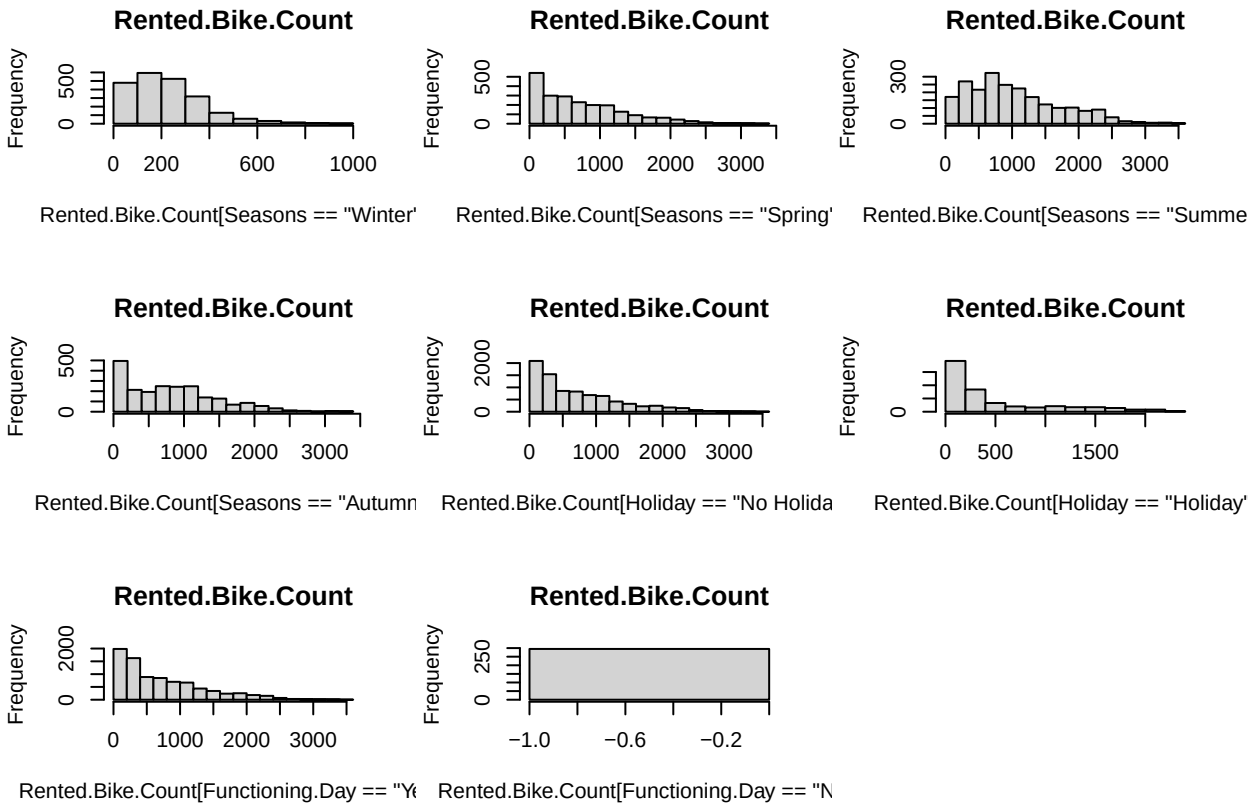
We treated the day *06/10/2018* as "Functioning.Day==Yes" because it only has 7 hours without functioning.

Heterogeneity of variable *Rented.Bike.Count* between other variabls

```
par(mfrow=c(3,4))
for(i in 2*(0:11))
with(bike,hist(Rented.Bike.Count[Hour==i],xlab=paste("Hour=",i),main="Rented.Bike.Count"))
```




```
par(mfrow=c(3,3))
with(bike,hist(Rented.Bike.Count[Seasons=="Winter"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Seasons=="Spring"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Seasons=="Summer"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Seasons=="Autumn"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Holiday=="No Holiday"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Holiday=="Holiday"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Functioning.Day=="Yes"],main="Rented.Bike.Count"))
with(bike,hist(Rented.Bike.Count[Functioning.Day=="No"],main="Rented.Bike.Count"))
```



Data analysis and Diagnostic test

Poisson Regressin and quasi-poisson regression

```
fitGlm1 = glm(Rented.Bike.Count ~ . -Date, family = poisson(link = "log"), data = bike)
```

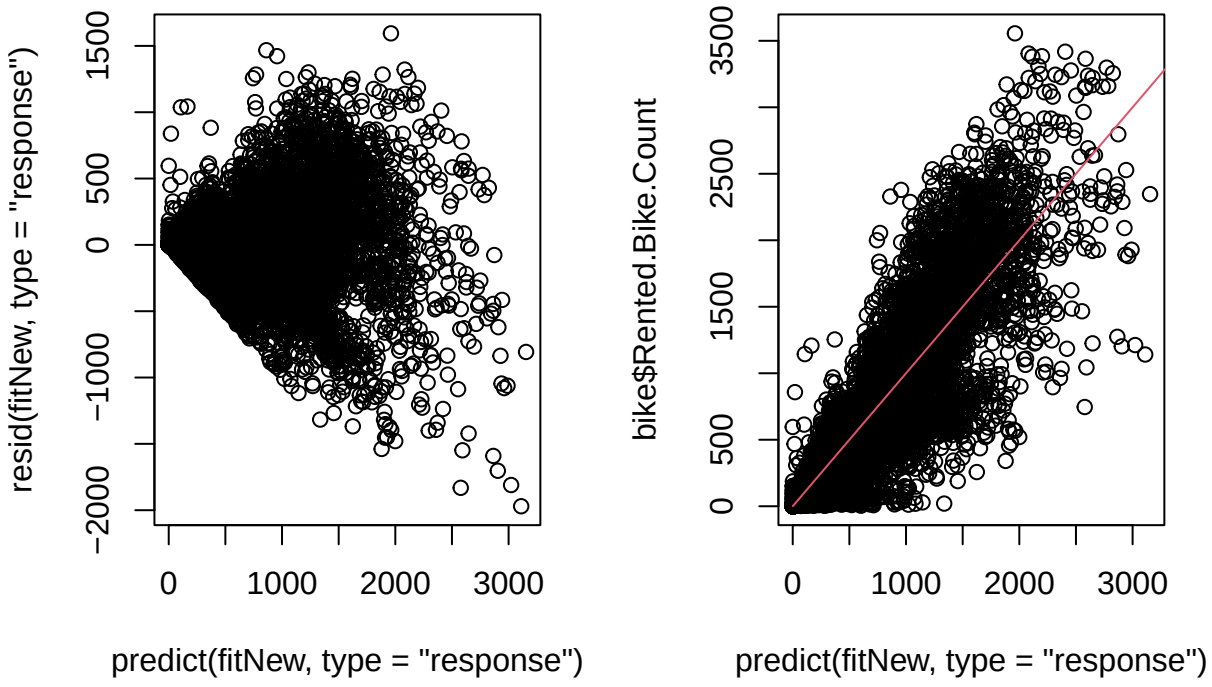
```
fitNew = update(fitGlm1, family = quasipoisson())
summary(fitNew)
```

```
##
## Call:
## glm(formula = Rented.Bike.Count ~ . - Date, family = quasipoisson(),
##     data = bike)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -49.655   -6.027   -0.004    5.208  108.864
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.183e+01  1.448e+04  -0.001    0.999
## Hour1                 -2.280e-01  3.281e+00  -0.069    0.945
```

```
## Hour2                -5.494e-01  3.642e+00  -0.151    0.880
## Hour3                -9.173e-01  4.170e+00  -0.220    0.826
## Hour4                -1.346e+00  4.912e+00  -0.274    0.784
## Hour5                -1.293e+00  4.822e+00  -0.268    0.789
## Hour6                -5.377e-01  3.707e+00  -0.145    0.885
## Hour7                 1.750e-01  3.013e+00   0.058    0.954
## Hour8                 6.399e-01  2.746e+00   0.233    0.816
## Hour9                 1.053e-01  3.093e+00   0.034    0.973
## Hour10               -2.064e-01  3.367e+00  -0.061    0.951
## Hour11               -1.576e-01  3.418e+00  -0.046    0.963
## Hour12               -5.162e-02  3.438e+00  -0.015    0.988
## Hour13               -4.238e-02  3.434e+00  -0.012    0.990
## Hour14               -2.830e-02  3.355e+00  -0.008    0.993
## Hour15                6.709e-02  3.216e+00   0.021    0.983
## Hour16                1.995e-01  3.026e+00   0.066    0.947
## Hour17                4.575e-01  2.824e+00   0.162    0.871
## Hour18                7.956e-01  2.631e+00   0.302    0.762
## Hour19                6.338e-01  2.674e+00   0.237    0.813
## Hour20                5.732e-01  2.701e+00   0.212    0.832
## Hour21                5.743e-01  2.700e+00   0.213    0.832
## Hour22                4.815e-01  2.745e+00   0.175    0.861
## Hour23                1.911e-01  2.926e+00   0.065    0.948
## Temperature           5.025e-03  2.983e-01   0.017    0.987
## Humidity             -1.301e-02  8.790e-02  -0.148    0.882
## Wind.speed           -1.687e-02  4.666e-01  -0.036    0.971
## Visibility           -6.264e-06  8.764e-04  -0.007    0.994
## Dew.point.temperature  2.320e-02  3.084e-01   0.075    0.940
## Solar.Radiation        4.072e-02  9.716e-01   0.042    0.967
## Rainfall             -5.461e-01  2.160e+00  -0.253    0.800
## Snowfall             -1.298e-01  1.934e+00  -0.067    0.947
## SeasonsSpring        -1.923e-01  1.132e+00  -0.170    0.865
## SeasonsSummer        -1.905e-01  1.336e+00  -0.143    0.887
## SeasonsWinter        -9.873e-01  2.025e+00  -0.487    0.626
## HolidayNo Holiday     1.726e-01  2.138e+00   0.081    0.936
## Functioning.DayYes    1.894e+01  1.448e+04   0.001    0.999
##
## (Dispersion parameter for quasipoisson family taken to be 936177.9)
##
##     Null deviance: 4979261  on 8759  degrees of freedom
## Residual deviance:  993729  on 8723  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 10
```

All variables are not significant. And the diagnostic plots seem bad too.

```
par(mfrow=c(1,2))
plot(predict(fitNew, type="response"),resid(fitNew,type="response"))
plot(predict(fitNew, type="response"),bike$Rented.Bike.Count)
lines(1:40000,1:40000,col=2)
```



Reconstruct the data *bike.day* and refit

Sum the counts in one day and use the count per day

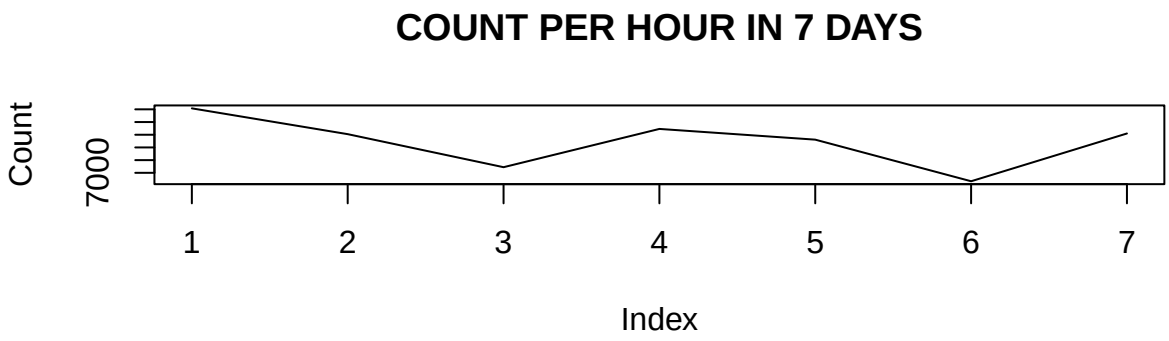
```
bike.day=data.frame(bike[1:365,])
is.num=c()
for(i in 1:14) is.num<-c(is.num, is.numeric(bike[,i]))
for(i in 1:365){
  for(j in 1:14){
    if(is.num[j]==TRUE)
      bike.day[i,j]=mean(bike[(1+24*(i-1)):(24*i),j])
    else
      bike.day[i,j]=bike[24*i,j]
  }
}
bike.day$Hour=NULL

bike.day$Rented.Bike.Count=24*bike.day$Rented.Bike.Count
bike.day$Seasons=as.factor((bike.day$Seasons))
bike.day$Holiday=as.factor(bike.day$Holiday)
bike.day$Functioning.Day=as.factor(bike.day$Functioning.Day)
```

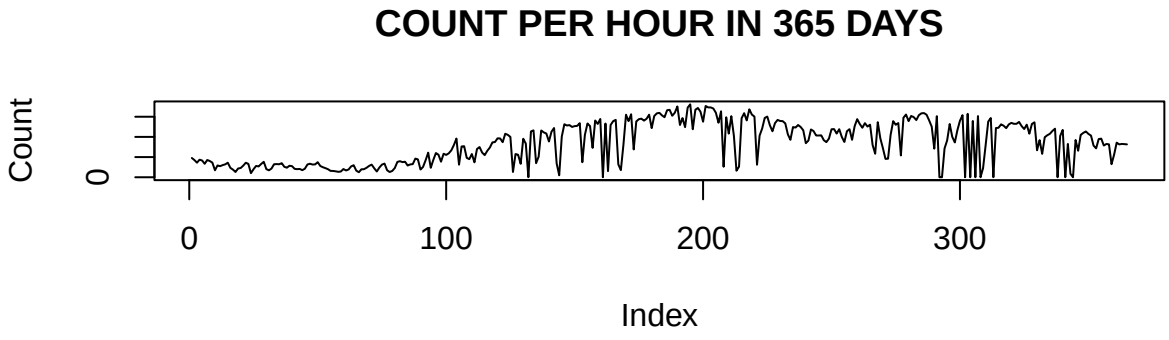
```
summary(bike.day)
```

```
##      Date           Rented.Bike.Count  Temperature         Humidity
##  Length:365         Min.   :    0     Min.   :-14.738   Min.   :22.25
##  Class :character   1st Qu.: 6500     1st Qu.:  3.812   1st Qu.:47.92
##  Mode  :character   Median :17730     Median : 13.838   Median :57.21
##                     Mean   :16910     Mean   : 12.883   Mean   :58.23
##                     3rd Qu.:26075     3rd Qu.: 22.425   3rd Qu.:67.54
##                     Max.   :36149     Max.   : 33.742   Max.   :95.88
##    Wind.speed       Visibility     Dew.point.temperature Solar.Radiation
##  Min.   :0.6625   Min.   : 214.3   Min.   :-27.750       Min.   :0.02917
##  1st Qu.:1.2958   1st Qu.:1087.5   1st Qu.: -4.862       1st Qu.:0.28542
##  Median :1.6417   Median :1557.8   Median :  5.008       Median :0.56500
##  Mean   :1.7249   Mean   :1436.8   Mean   :  4.074       Mean   :0.56911
##  3rd Qu.:1.9542   3rd Qu.:1877.2   3rd Qu.: 14.571       3rd Qu.:0.81583
##  Max.   :4.0000   Max.   :2000.0   Max.   : 25.038       Max.   :1.21667
##     Rainfall          Snowfall         Seasons         Holiday
##  Min.   :0.00000   Min.   :0.00000   Autumn:91   Holiday   : 18
##  1st Qu.:0.00000   1st Qu.:0.00000   Spring:92   No Holiday:347
##  Median :0.00000   Median :0.00000   Summer:92
##  Mean   :0.14869   Mean   :0.07507   Winter:90
##  3rd Qu.:0.02083   3rd Qu.:0.00000
##  Max.   :3.97917   Max.   :3.27917
##  Functioning.Day
##  No : 12
##  Yes:353
##
##
##
##
```

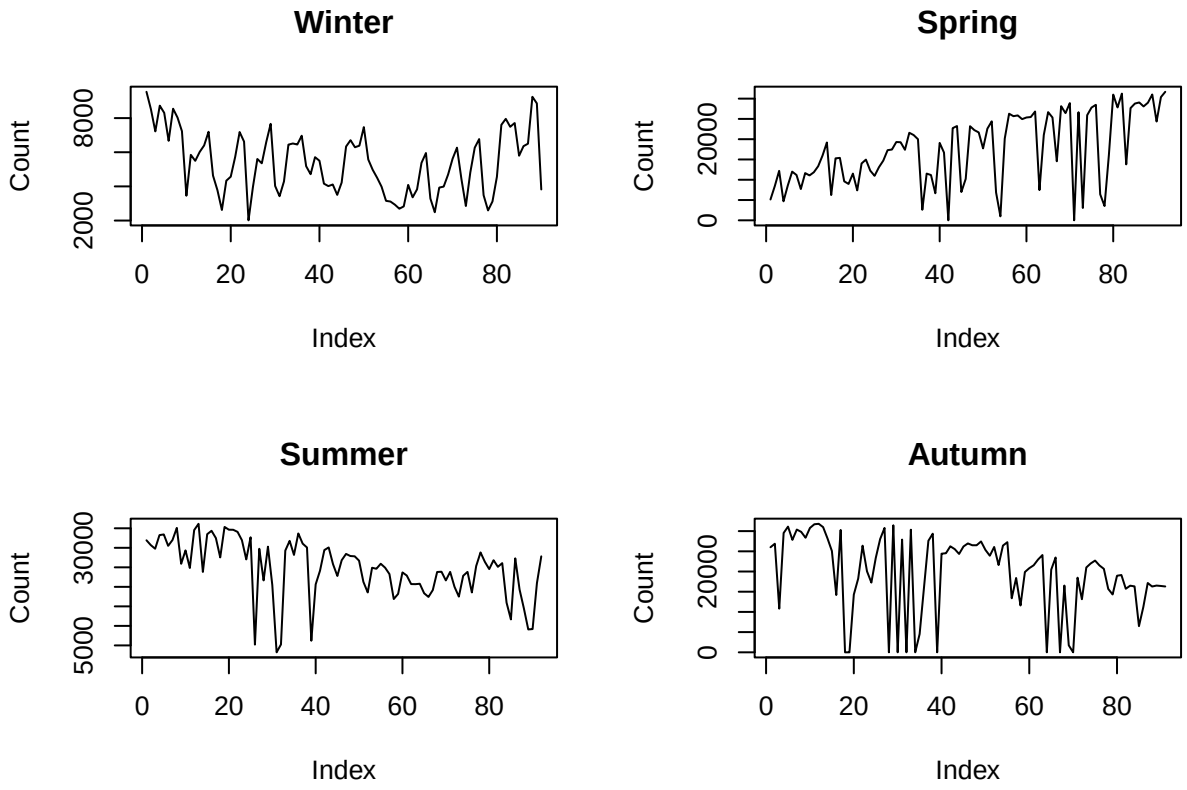
Visualisation agian (easily got the heterogeneity between sesons)

```
par(mfrow=c(2,1))
with(bike.day,plot(Rented.Bike.Count[1:7],type="l",ylab="Count",main="COUNT PER HOUR IN 7 DAYS"))
with(bike.day,plot(Rented.Bike.Count[1:365],type="l",ylab="Count",main="COUNT PER HOUR IN 365 DAYS"))
```






```
par(mfrow=c(2,2))
with(bike.day,plot(Rented.Bike.Count[Seasons=="Winter"],type="l",ylab="Count",main="Winter"))
with(bike.day,plot(Rented.Bike.Count[Seasons=="Spring"],type="l",ylab="Count",main="Spring"))
with(bike.day,plot(Rented.Bike.Count[Seasons=="Summer"],type="l",ylab="Count",main="Summer"))
with(bike.day,plot(Rented.Bike.Count[Seasons=="Autumn"],type="l",ylab="Count",main="Autumn"))
```



Delete obsevations with Functioning.Day==No and refit

The reason is that we can set the count=0 at these days. (And it is truth from data.)

```
vec=bike.day$Functioning.Day=="Yes"# only run once
```

```
bike.day=bike.day[vec,]
bike.day$Functioning.Day=NULL
bike.day$Date=NULL
```

```
fitGlm2 = glm(Rented.Bike.Count ~ ., family = poisson(link = "log"), data = bike.day)
#par(mfrow = c(2, 2))
#plot(fitGlm2) # diagnostic plot
res2 = summary(fitGlm2) # check coefficients
res2
```

```
##
## Call:
## glm(formula = Rented.Bike.Count ~ ., family = poisson(link = "log"),
##     data = bike.day)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -86.765  -22.265   -0.901   17.225   80.167
```

```
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.045e+01  1.738e-02  601.03   <2e-16 ***
## Temperature            -2.644e-02  6.642e-04  -39.81   <2e-16 ***
## Humidity               -1.120e-02  1.952e-04  -57.38   <2e-16 ***
## Wind.speed             -7.841e-02  8.768e-04  -89.43   <2e-16 ***
## Visibility             -3.491e-05  1.168e-06  -29.89   <2e-16 ***
## Dew.point.temperature   4.500e-02  6.950e-04   64.75   <2e-16 ***
## Solar.Radiation         6.397e-01  2.281e-03  280.47   <2e-16 ***
## Rainfall               -3.354e-01  1.522e-03 -220.41   <2e-16 ***
## Snowfall               -1.747e-01  2.112e-03  -82.70   <2e-16 ***
## SeasonsSpring          -2.786e-01  1.276e-03 -218.37   <2e-16 ***
## SeasonsSummer          -2.317e-01  1.432e-03 -161.77   <2e-16 ***
## SeasonsWinter          -9.477e-01  2.166e-03 -437.51   <2e-16 ***
## HolidayNo Holiday       2.151e-01  2.212e-03   97.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2235618  on 352  degrees of freedom
## Residual deviance:  360114  on 340  degrees of freedom
## AIC: 364154
##
## Number of Fisher Scoring iterations: 4
```

```
sum(resid(fitGlm2, type = "pearson")^2) / fitGlm2$df.residual # mean of pearson residuals squared
```

```
## [1] 1035.13
```

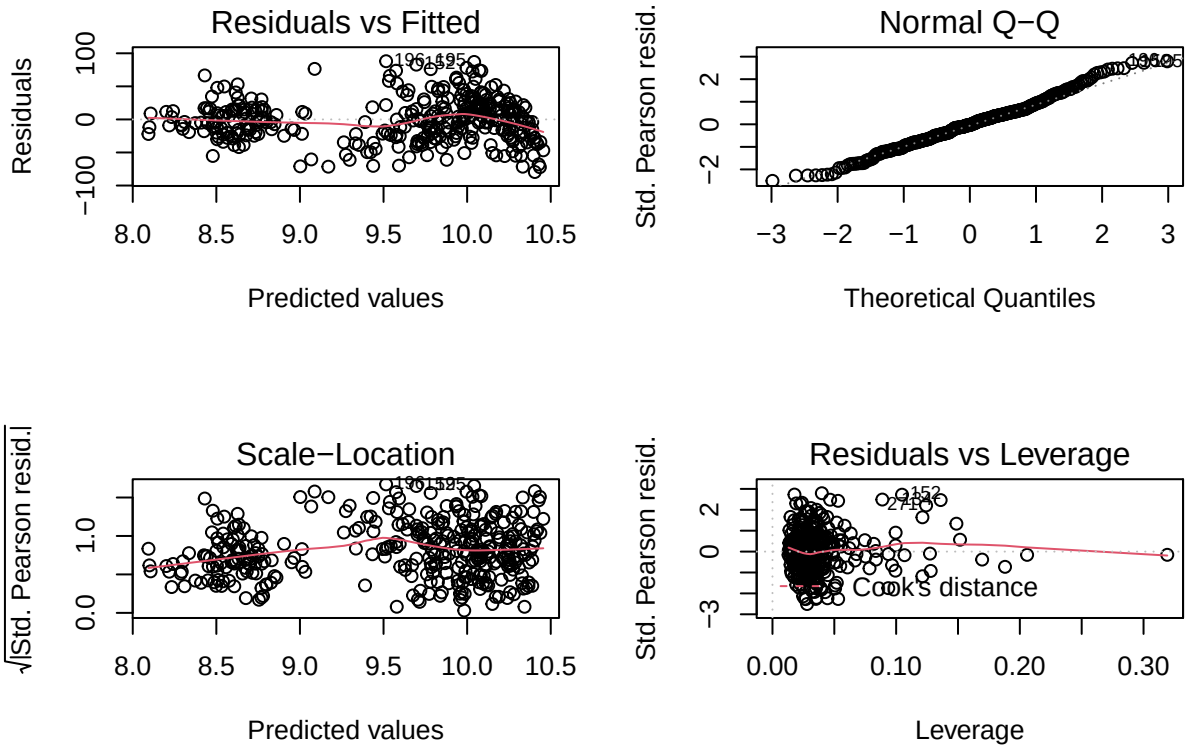
The summary shows that every variables are so significant. But mean of pearson residuals squared is more than 1000 which means great overdispersion.

```
fitNew2 = update(fitGlm2, family = quasipoisson())#refit quasi-poisson
summary(fitNew2)
```

```
##
## Call:
## glm(formula = Rented.Bike.Count ~ ., family = quasipoisson(),
##     data = bike.day)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -86.765  -22.265   -0.901   17.225   80.167
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.045e+01  5.591e-01  18.681  < 2e-16 ***
## Temperature            -2.644e-02  2.137e-02  -1.237  0.21677
## Humidity               -1.120e-02  6.281e-03  -1.783  0.07541 .
## Wind.speed             -7.841e-02  2.821e-02  -2.780  0.00575 **
```

```
## Visibility            -3.491e-05  3.757e-05  -0.929  0.35345
## Dew.point.temperature  4.500e-02  2.236e-02   2.013  0.04494 *
## Solar.Radiation        6.397e-01  7.338e-02   8.717  < 2e-16 ***
## Rainfall              -3.354e-01  4.896e-02  -6.851 3.45e-11 ***
## Snowfall              -1.747e-01  6.796e-02  -2.571  0.01058 *
## SeasonsSpring         -2.786e-01  4.104e-02  -6.787 5.10e-11 ***
## SeasonsSummer         -2.317e-01  4.608e-02  -5.028 8.03e-07 ***
## SeasonsWinter         -9.477e-01  6.969e-02 -13.598  < 2e-16 ***
## HolidayNo Holiday      2.151e-01  7.118e-02   3.022  0.00270 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1035.139)
##
##     Null deviance: 2235618  on 352  degrees of freedom
## Residual deviance:  360114  on 340  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
par(mfrow=c(2,2))
plot(fitNew2)
```



Refit quasi-poisson model and diagnostic plots are all great. (Even Normal Q-Q plot is so good that we do not need robust regression or something.)

Intepretation of results

- Among variables, only *Temperature*, *Humidity* and *Visibility* are not significant (p-value > 0.05); among other variables, *Solar.Radiation*, *Rainfall*, *Wind.speed*, *Seasons* and *Holiday* are super significant (p-value < 0.01).
- Only *Dew.point.temperature*, *Solar.Radiation* and *HolidayNo Holiday* has positive coefficients. It is clear that in sunny workdays without rain or snow, people will rent more bikes.

All the above seem reasonable.

try to remove some unsignificant variables

```
fitNew3 = glm(Rented.Bike.Count ~ .-Temperature-Humidity-Visibility, family = quasipoisson(), data = bi
summary(fitNew3)
```

```
##
## Call:
## glm(formula = Rented.Bike.Count ~ . - Temperature - Humidity -
##     Visibility, family = quasipoisson(), data = bike.day)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -92.422  -22.697   -0.422   17.007   82.791
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             9.516661   0.090962 104.622  < 2e-16 ***
## Wind.speed             -0.085851   0.027536  -3.118  0.00198 **
## Dew.point.temperature   0.013702   0.002193   6.249 1.22e-09 ***
## Solar.Radiation         0.701434   0.058280  12.036  < 2e-16 ***
## Rainfall               -0.373207   0.046157  -8.086 1.07e-14 ***
## Snowfall               -0.197825   0.067229  -2.943  0.00348 **
## SeasonsSpring          -0.278819   0.037866  -7.363 1.34e-12 ***
## SeasonsSummer          -0.218820   0.045039  -4.859 1.80e-06 ***
## SeasonsWinter          -0.939239   0.068179 -13.776  < 2e-16 ***
## HolidayNo Holiday       0.207514   0.071117   2.918  0.00376 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1036.978)
##
##     Null deviance: 2235618  on 352  degrees of freedom
## Residual deviance:  365104  on 343  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fitNew2,fitNew3,test = "Chisq")
```

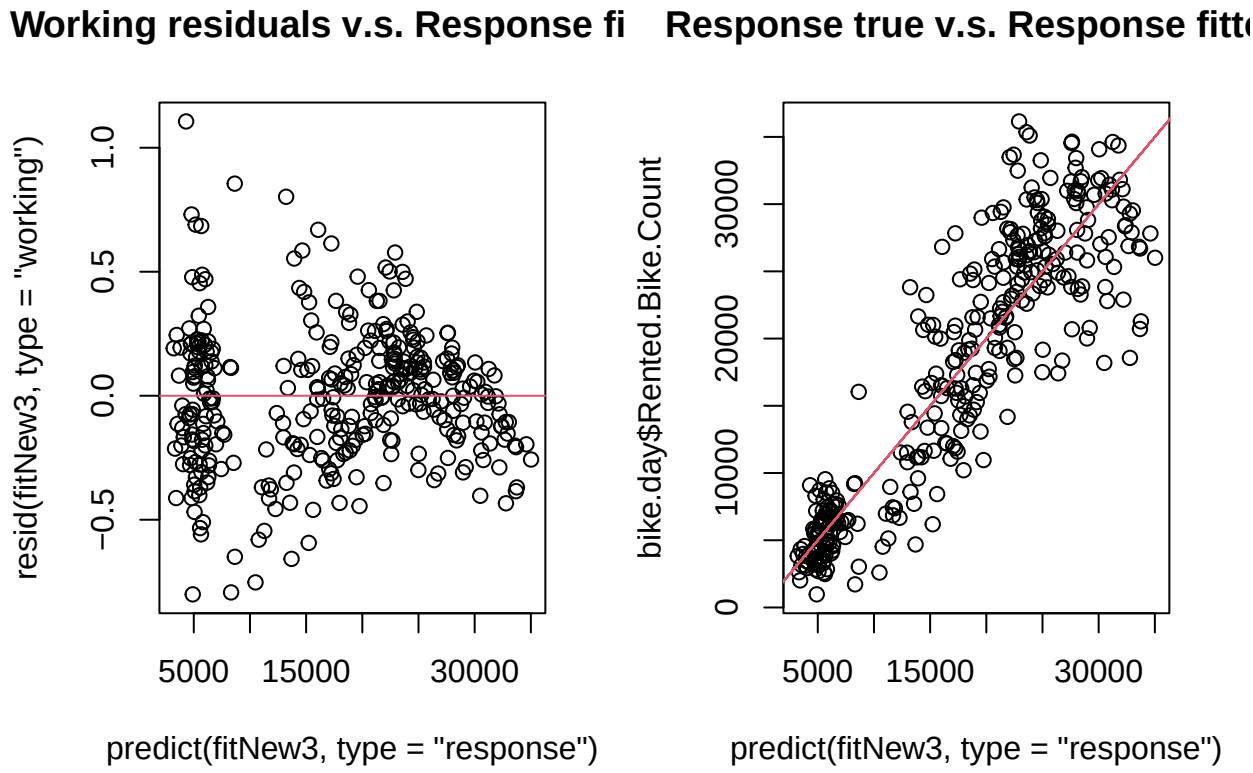
```
## Analysis of Deviance Table
##
## Model 1: Rented.Bike.Count ~ Temperature + Humidity + Wind.speed + Visibility +
```

```
##     Dew.point.temperature + Solar.Radiation + Rainfall + Snowfall +
##     Seasons + Holiday
## Model 2: Rented.Bike.Count ~ (Temperature + Humidity + Wind.speed + Visibility +
##     Dew.point.temperature + Solar.Radiation + Rainfall + Snowfall +
##     Seasons + Holiday) - Temperature - Humidity - Visibility
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       340     360114
## 2       343     365104 -3  -4989.9   0.1854
```

fitNew3 as a simplified model of *fitNew2* is ok. At least it passes the anova chisq test and keeps all left variables significant with similar coefficients.

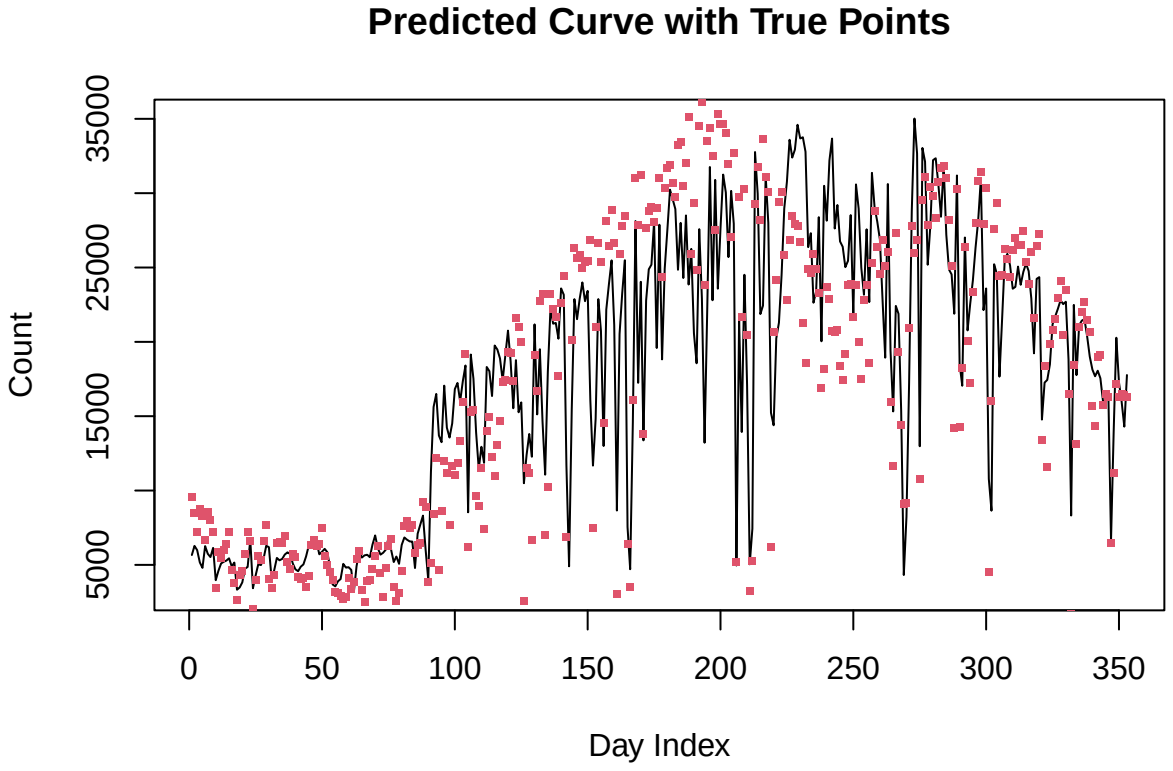
Extended model assessment

```
par(mfrow=c(1,2))
plot(predict(fitNew3, type="response"),resid(fitNew3,type="working"),main="Working residuals v.s. Respo
abline(h=0,col=2)
plot(predict(fitNew3, type="response"),bike.day$Rented.Bike.Count,main="Response true v.s. Response fit
lines(1:40000,1:40000,col=2)
```




```
list=predict(fitNew3, type="response",se.fit=TRUE)
#list$fit
#list$se.fit
```

```
pred.high=list$fit+1.96*list$se.fit
pred.low=list$fit-1.96*list$se.fit
plot(list$fit,type="l",col=1,ylab="Count",xlab="Day Index",main="Predicted Curve with True Points")
#lines(pred.high,col=3)
#lines(pred.low,col=4)
points(bike.day$Rented.Bike.Count,col=2,cex=0.5,pch=15)
```



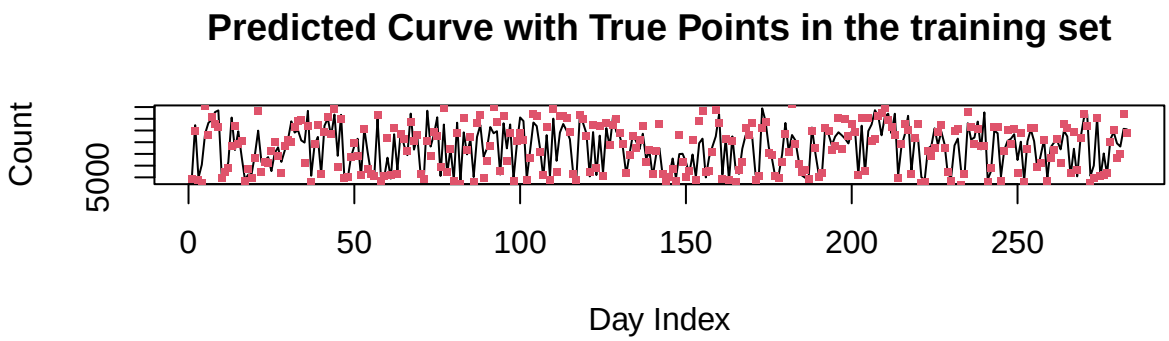
training and test

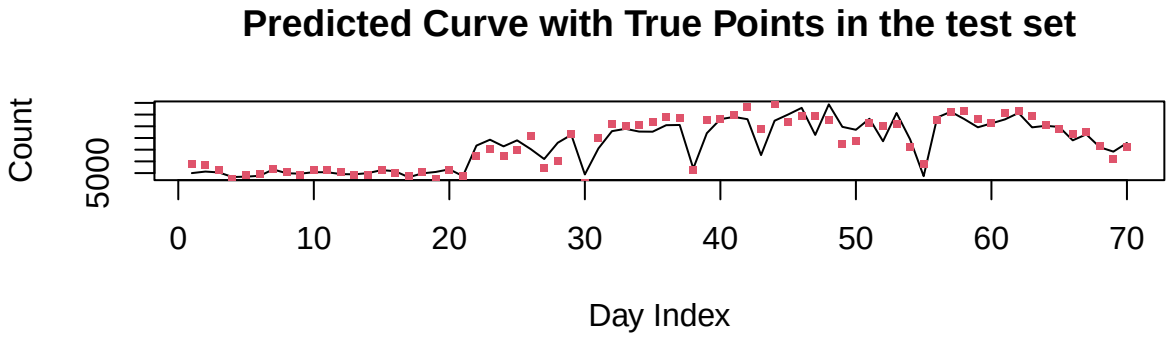
```
set.seed(123456)
num=nrow(bike.day)
n.train=ceiling(num*0.8)
ind.train=sample((1:num),n.train)
ind.test=(1:num)[-ind.train]
fitTrain = glm(Rented.Bike.Count ~ .-Temperature-Humidity-Visibility, family = quasipoisson(), data = bi
summary(fitTrain)
```

```
##
## Call:
## glm(formula = Rented.Bike.Count ~ . - Temperature - Humidity -
##     Visibility, family = quasipoisson(), data = bike.day[ind.train,
##     ])
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -91.619  -22.529   -0.693   18.125   83.250
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.570319   0.103326  92.622  < 2e-16 ***
## Wind.speed              -0.095586   0.031957  -2.991 0.003034 **
## Dew.point.temperature    0.012968   0.002505   5.176 4.40e-07 ***
## Solar.Radiation          0.688924   0.067660  10.182  < 2e-16 ***
## Rainfall                -0.416857   0.056888  -7.328 2.65e-12 ***
## Snowfall                -0.190911   0.097450  -1.959 0.051122 .
## SeasonsSpring           -0.264217   0.043398  -6.088 3.86e-09 ***
## SeasonsSummer           -0.197742   0.052182  -3.789 0.000186 ***
## SeasonsWinter           -0.963619   0.078123 -12.335  < 2e-16 ***
## HolidayNo Holiday        0.172575   0.078037   2.211 0.027834 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1095.747)
##
##     Null deviance: 1749304  on 282  degrees of freedom
## Residual deviance:  307214  on 273  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```r
par(mfrow=c(2,1))

list=predict(fitTrain,type = "response",se.fit=T)
#list$fit
#list$se.fit
pred.high=list$fit+1.96*list$se.fit
pred.low=list$fit-1.96*list$se.fit
plot(list$fit,type="l",col=1,ylab="Count",xlab="Day Index",main="Predicted Curve with True Points in th
#lines(pred.high,col=3)
#lines(pred.low,col=4)
points(bike.day[ind.train,"Rented.Bike.Count"],col=2,cex=0.5,pch=15)



list=predict(fitTrain,newdata = bike.day[ind.test,-1],type = "response",se.fit=T)
#list$fit
#list$se.fit
pred.high=list$fit+1.96*list$se.fit
pred.low=list$fit-1.96*list$se.fit
plot(list$fit,type="l",col=1,ylab="Count",xlab="Day Index",main="Predicted Curve with True Points in th
#lines(pred.high,col=3)
#lines(pred.low,col=4)
points(bike.day[ind.test,"Rented.Bike.Count"],col=2,cex=0.5,pch=15)
```





some metrics

```
fit=list$fit
true=bike.day[ind.test,"Rented.Bike.Count"]
resid = true - fit
mean(abs(resid))#mean of |resid|
```

```
## [1] 2759.045
```

```
sqrt(mean((resid)^2))# mse of resid
```

```
## [1] 3668.924
```

```
mean(abs(resid)/fit)#mean of abs(resid)/fitted
```

```
## [1] 0.2081951
```

```
mean(resid/fit)#mean of resid/fitted
```

```
## [1] 0.04930855
```

The ability of Generalisaion is ok. Everything is ok up to now.

Some extensions

- Actually it lost information to sum hour counts to day counts as above. But we had shown that it was untractable to add crudely 23 dummy variables to the model.
- Raw data is time series and the observations are dependent. So we may use some tools from time series data analysis to exploit more information.
- Or we just forget the interpretability of models and use machine learning.

References

- Kejun He. 'GLM using R.'
- Sathishkumar V E, Jangwoo Park, and Yongyun Cho. 'Using data mining techniques for bike sharing demand prediction in metropolitan city.' Computer Communications, Vol.153, pp.353-366, March, 2020
- Sathishkumar V E and Yongyun Cho. 'A rule-based model for Seoul Bike sharing demand prediction using weather data' European Journal of Remote Sensing, pp. 1-18, Feb, 2020