STAT 474 - Summer 2021
Project
Thakshila Herath

==

I used the ‘Bike Sharing’ Dataset for my regression project. The dataset can be found here : https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset[1]

I downloaded ‘day.csv’ as the ‘hours.csv’ file is bulky. It contains the daily count of rental bikes in the Capital bikeshare system in 2011-2012. It also contains corresponding information about days, seasons and weather. We are trying to find the demand for bike rentals depending on factors such as day, season and weather.

Attribute Information:

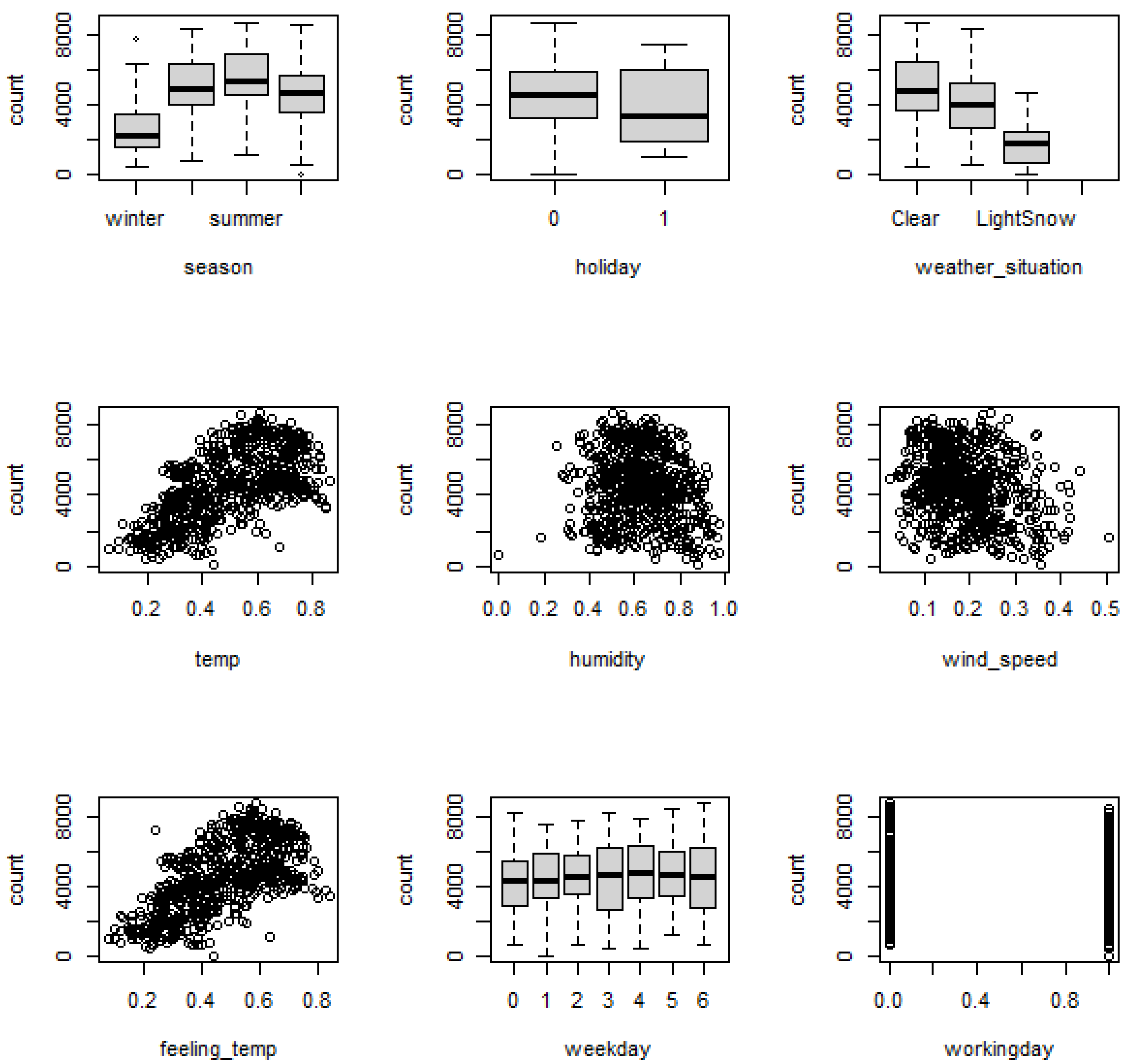
- instant: record index
- dteday : date
- season : season (1:winter, 2:spring, 3:summer, 4:fall)
- yr : year (0: 2011, 1:2012)
- mnth : month (1 to 12)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (extracted from [Web Link])
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.

+ weathersit :

- 1: Clear, Few clouds, Partly cloudy, 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
- temp : Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
- atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

Visualizing data

First, the dataset was graphically visualized by plotting dependent variable (y=count here) vs independent variables (x's). Here pairwise plots were not generated as we do not want to visualize all the predictors. Because some of them have auto-correlation by default. The plots itself clearly indicates that counts highly depend on season, holiday, weather situation, temperature, humidity level and wind speed.



Analysis methods

It is count data. So I started with a 'Poisson loglinear model' including all the parameters.

Poisson loglinear model

```
> fit_bike <-
glm(cnt~season+holiday+weekday+weathersit+temp+atemp+hum+windspeed,
+                      family = poisson, data = data_bike)
> fit_bike <-
glm(cnt~season+holiday+weekday+weathersit+temp+atemp+hum+windspeed,
+                      family = poisson, data = data_bike)
> summary(fit_bike)

Call:
glm(formula = cnt ~ season + holiday + weekday + weathersit +
    temp + atemp + hum + windspeed, family = poisson, data =
data_bike)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-59.987  -15.903   -2.579   15.817   64.279

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          7.866794   0.004584 1716.22   <2e-16 ***
seasonspring         0.308743   0.002269  136.07   <2e-16 ***
seasonsummer         0.196415   0.002833   69.33   <2e-16 ***
seasonfall)          0.454923   0.001980  229.77   <2e-16 ***
holiday1            -0.167408   0.003748  -44.67   <2e-16 ***
weekday1             0.050374   0.002160   23.33   <2e-16 ***
weekday2             0.066412   0.002100   31.62   <2e-16 ***
weekday3             0.078144   0.002098   37.24   <2e-16 ***
weekday4             0.080811   0.002088   38.70   <2e-16 ***
weekday5             0.085122   0.002089   40.75   <2e-16 ***
weekday6             0.089116   0.002090   42.64   <2e-16 ***
weathersitMist      -0.058836   0.001497  -39.31   <2e-16 ***
weathersitLightSnow -0.755518   0.005493 -137.54   <2e-16 ***
temp                 0.899895   0.024569   36.63   <2e-16 ***
atemp                0.548231   0.027140   20.20   <2e-16 ***
hum                 -0.553910   0.005494 -100.82   <2e-16 ***
windspeed           -0.697644   0.008188  -85.20   <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: 668801  on 730  degrees of freedom
Residual deviance: 293733  on 714  degrees of freedom
AIC: 301166

Number of Fisher Scoring iterations: 4
```

Model Diagnostics

Checking the correlation using 'vif'

```
> vif(fit_bike)
                GVIF Df GVIF^(1/(2*Df))
season      3.392721  3        1.225814
holiday     1.075449  1        1.037038
weekday     1.113293  6        1.008984
weathersit  1.652626  2        1.133819
temp       60.666564  1        7.788874
atemp      57.585282  1        7.588497
hum         1.771636  1        1.331028
windspeed   1.188379  1        1.090128
```

⇒ 'temp' and 'atemp' have large values which indicate the auto-correlation. It is true as they are 'temperature' and 'feeling temperature'. So I removed 'atemp' from my model.

```
> fit_bike_red <-
glm(cnt~season+holiday+weekday+weathersit+temp+hum+windspeed,
+                  family = poisson, data = data_bike)
> summary(fit_bike_red)

Call:
glm(formula = cnt ~ season + holiday + weekday + weathersit +
    temp + hum + windspeed, family = poisson, data = data_bike)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-59.051  -16.145   -2.543   15.757   64.711

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          7.888164   0.004463 1767.51   <2e-16 ***
seasonspring         0.310556   0.002269  136.87   <2e-16 ***
seasonsummer         0.193745   0.002831   68.44   <2e-16 ***
```

```
seasonfall)           0.457347   0.001977  231.31   <2e-16 ***
holiday1             -0.170312   0.003745  -45.47   <2e-16 ***
weekday1              0.050830   0.002160   23.54   <2e-16 ***
weekday2              0.066156   0.002100   31.50   <2e-16 ***
weekday3              0.076784   0.002097   36.62   <2e-16 ***
weekday4              0.080463   0.002088   38.53   <2e-16 ***
weekday5              0.082063   0.002084   39.37   <2e-16 ***
weekday6              0.088193   0.002089   42.21   <2e-16 ***
weathersitMist       -0.060449   0.001495  -40.44   <2e-16 ***
weathersitLightSnow -0.760769   0.005488 -138.63   <2e-16 ***
temp                  1.383015   0.005599  247.02   <2e-16 ***
hum                  -0.546436   0.005481  -99.69   <2e-16 ***
windspeed            -0.718536   0.008130  -88.39   <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: 668801  on 730  degrees of freedom
Residual deviance: 294162  on 715  degrees of freedom
AIC: 301592

Number of Fisher Scoring iterations: 4
```

Then re-check autocorrelation

```
> vif(fit_bike_red)
               GVIF Df GVIF^(1/(2*Df))
season     3.311746  3        1.220889
holiday    1.074099  1        1.036388
weekday    1.104688  6        1.008331
weathersit 1.646798  2        1.132818
temp       3.153458  1        1.775798
hum        1.765477  1        1.328712
windspeed  1.171122  1        1.082184
```

⇒ vif values are less than 5 and look better now. I then check for overdispersion.

```
> #If Residual Deviance/df >1, then overdispersion >
deviance(fit_bike_red)/df.residual(fit_bike_red)
[1] 411.4147
```

Overdispersion

Overdispersion is detected. I have two ways to fix the overdispersion.

1. Quasi-poisson which assumes variance is a linear function of mean.
2. Negative binomial which assumes variance is a quadratic function of mean

First, I tried quasipoisson. But overdispersion is still there. Then I used negative binomial and it helps solve the overdispersion issue.

```
> nb_fit_bike <-
glm.nb(cnt~season+holiday+weekday+weathersit+temp+hum+windspeed,
+                        data = data_bike)
> deviance(nb_fit_bike)/df.residual(nb_fit_bike)
[1] 1.045605

> summary(nb_fit_bike)

Call:
glm.nb(formula = cnt ~ season + holiday + weekday + weathersit +
    temp + hum + windspeed, data = data_bike, init.theta =
8.73466464,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.5215  -0.7691  -0.1375   0.6684   2.8505

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          7.80095    0.09532  81.840  < 2e-16 ***
seasonspring         0.25284    0.04672   5.411 6.25e-08 ***
seasonsummer         0.10485    0.06163   1.701  0.08888 .
seasonfall)          0.44187    0.03984  11.090  < 2e-16 ***
holiday1            -0.21012    0.07838  -2.681  0.00735 **
weekday1             0.06590    0.04816   1.368  0.17119
weekday2             0.08393    0.04706   1.783  0.07453 .
weekday3             0.07528    0.04717   1.596  0.11051
weekday4             0.09134    0.04714   1.938  0.05267 .
weekday5             0.10672    0.04715   2.263  0.02361 *
weekday6             0.09682    0.04689   2.065  0.03896 *
weathersitMist      -0.05486    0.03350  -1.637  0.10154
weathersitLightSnow -0.73528    0.08597  -8.553  < 2e-16 ***
temp                 1.77153    0.12582  14.080  < 2e-16 ***
hum                 -0.64656    0.12120  -5.335 9.57e-08 ***
windspeed           -0.82400    0.17639  -4.671 2.99e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for Negative Binomial(8.7347) family taken to
be 1)

    Null deviance: 1645.54  on 730  degrees of freedom
Residual deviance:  747.61  on 715  degrees of freedom
AIC: 12667

Number of Fisher Scoring iterations: 1


              Theta:  8.735
          Std. Err.:  0.451

 2 x log-likelihood:  -12633.261
```

Searching for the best model

Searched for the best model using the function called 'dredge' in the package 'MuMIn'. It checks for every possible combination of all variables and estimates the best model.

```
> # fiding the best model with all possible variables
> #install.packages(MuMIn)
> library(MuMIn)
> nb_fit_bike2 <-
glm.nb(cnt~season+holiday+weekday+weathersit+temp+hum+windspeed,
+                       data = data_bike, na.action = na.pass)
> nb_fit_best <- dredge(nb_fit_bike2)
Fixed term is "(Intercept)"
> head(nb_fit_best)
Global model call: glm.nb(formula = cnt ~ season + holiday + weekday
+ weathersit +
    temp + hum + windspeed, data = data_bike, na.action = na.pass,
    init.theta = 8.73466464, link = log)
---
Model selection table
    (Intrc) holdy     hum seasn  temp wthrs wekdy    wndsp df    logLik    AICc delta weight
96    7.886     + -0.6733     + 1.784     +        -0.8264 11 -6319.966 12662.3  0.00  0.885
95    7.885       -0.6818     + 1.782     +        -0.8352 10 -6323.652 12667.6  5.31  0.062
128   7.801     + -0.6466     + 1.772     +     + -0.8240 17 -6316.630 12668.1  5.82  0.048
127   7.805       -0.6520     + 1.768     +     + -0.8315 16 -6319.959 12672.7 10.38  0.005
32    7.616     + -0.4976     + 1.788     +               10 -6330.369 12681.0 18.75  0.000
31    7.613       -0.5043     + 1.784     +                9 -6334.194 12686.6 24.34  0.000
Models ranked by AICc(x)
```

According to the output, the best model has the lowest AIC. New model should include the predictors : season, holiday, weathersit, temp, hum, and windspeed.

Best model using negative binomial regression

So I fit the model excluding 'weekday' and check the anova results to make sure of the predictor elimination.

```
> nb_fit_bike3 <- update(nb_fit_bike, . ~ . - weekday)
> anova(nb_fit_bike, nb_fit_bike3)
Likelihood ratio tests of Negative Binomial Models

Response: cnt
                                                                Model    theta Resid. df
1           season + holiday + weathersit + temp + hum + windspeed 8.657569       721
2 season + holiday + weekday + weathersit + temp + hum + windspeed 8.734665       715

     2 x log-lik.   Test    df LR stat.   Pr(Chi)
1       -12639.93
2       -12633.26 1 vs 2     6 6.670105 0.3524355
```

p-value = 0.3524 > significant level

i.e. the predictor 'weekday' is not a statistically significant predictor in the model.

Summary and overdispersion parameter of the new model

```
> summary(nb_fit_bike3)

Call:
glm.nb(formula = cnt ~ season + holiday + weathersit + temp +
    hum + windspeed, data = data_bike, init.theta = 8.657568839,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.5080  -0.7607  -0.1639   0.6697   2.8872

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)               7.88570    0.08985  87.763  < 2e-16 ***
seasonspring              0.24991    0.04690   5.329 9.88e-08 ***
seasonsummer              0.10152    0.06182   1.642  0.10059
seasonfall)               0.44091    0.04000  11.022  < 2e-16 ***
```

```
holiday1             -0.21231    0.07551  -2.812  0.00493 **
weathersitMist       -0.04780    0.03346  -1.429  0.15313
weathersitLightSnow -0.71828    0.08580  -8.372  < 2e-16 ***
temp                  1.78406    0.12602  14.157  < 2e-16 ***
hum                  -0.67325    0.12095  -5.566 2.60e-08 ***
windspeed            -0.82636    0.17701  -4.669 3.03e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(8.6576) family taken to
be 1)

    Null deviance: 1631.08  on 730  degrees of freedom
Residual deviance:  747.69  on 721  degrees of freedom
AIC: 12662

Number of Fisher Scoring iterations: 1


              Theta:  8.658
          Std. Err.:  0.447

 2 x log-likelihood:  -12639.931

> deviance(nb_fit_bike3)/df.residual(nb_fit_bike3) # checking
overdispersion parameter
[1] 1.037021
```

Summary of the the poisson model, negative binomial (nb) model and best-nb subset model :

	Poisson loglinear model	Negative binomial (nb) loglinear model	Best-nb model
Null deviance	668801	1645.54	1631.08
Residual deviance	294162	747.61	747.69

AIC	301592	12667	12662
Overdispersion	411.41	1.046	1.037

The best negative binomial loglinear model (best-nb) has lower deviances, lower AIC and lower overdispersion parameter than the poisson and negative binomial loglinear models. So, the best-nb loglinear model gives a better fit than the other models. All the predictors and overall best-nb model is statistically significant.

Prediction equation

Poisson loglinear model :

$\log(\hat{\mu})$= 7.89 + 0.25*sesonspring +0.102*seasonsummer +0.441*seasonfall - 0.212*holiday1 - 0.0478*weatherMist - 0.718*weathersitLightSnow + 1.784*temp - 0.573*hum -0.826*windspeed

Estimated log coefficients and 95% confidence interval

```
> ##Confidence Intervals
> cbind(coef(nb_fit_bike3),confint.default(nb_fit_bike3,level=0.95))
#Confidence intervals for beta parameters in log[E(Y)]
                                         2.5 %      97.5 %
(Intercept)          7.88569808  7.70959160  8.06180457
seasonspring         0.24991274  0.15799390  0.34183157
seasonsummer         0.10151673 -0.01965668  0.22269015
seasonfall)          0.44091205  0.36250431  0.51931979
holiday1            -0.21231339 -0.36030896 -0.06431781
weathersitMist      -0.04779834 -0.11337649  0.01777980
weathersitLightSnow -0.71827959 -0.88643537 -0.55012380
temp                 1.78405844  1.53705853  2.03105835
hum                 -0.67325259 -0.91031525 -0.43618992
windspeed           -0.82635817 -1.17328535 -0.47943099
```

Estimated odd-ratios and 95% confidence interval

```
> exp(cbind(oddR =
coef(nb_fit_bike3),confint.default(nb_fit_bike3,level=0.95))) ## odd
ratios, Conf Intervals of beta parameters
```

```
                          oddR        2.5 %       97.5 %
(Intercept)          2658.9805701 2229.6314910 3171.0072722
seasonspring            1.2839134    1.1711590    1.4075232
seasonsummer            1.1068484    0.9805353    1.2494334
seasonfall)             1.5541240    1.4369234    1.6808839
holiday1                0.8087112    0.6974608    0.9377069
weathersitMist          0.9533260    0.8928145    1.0179388
weathersitLightSnow     0.4875904    0.4121222    0.5768784
temp                    5.9539713    4.6508897    7.6221490
hum                     0.5100469    0.4023973    0.6464949
windspeed               0.4376402    0.3093489    0.6191356
```

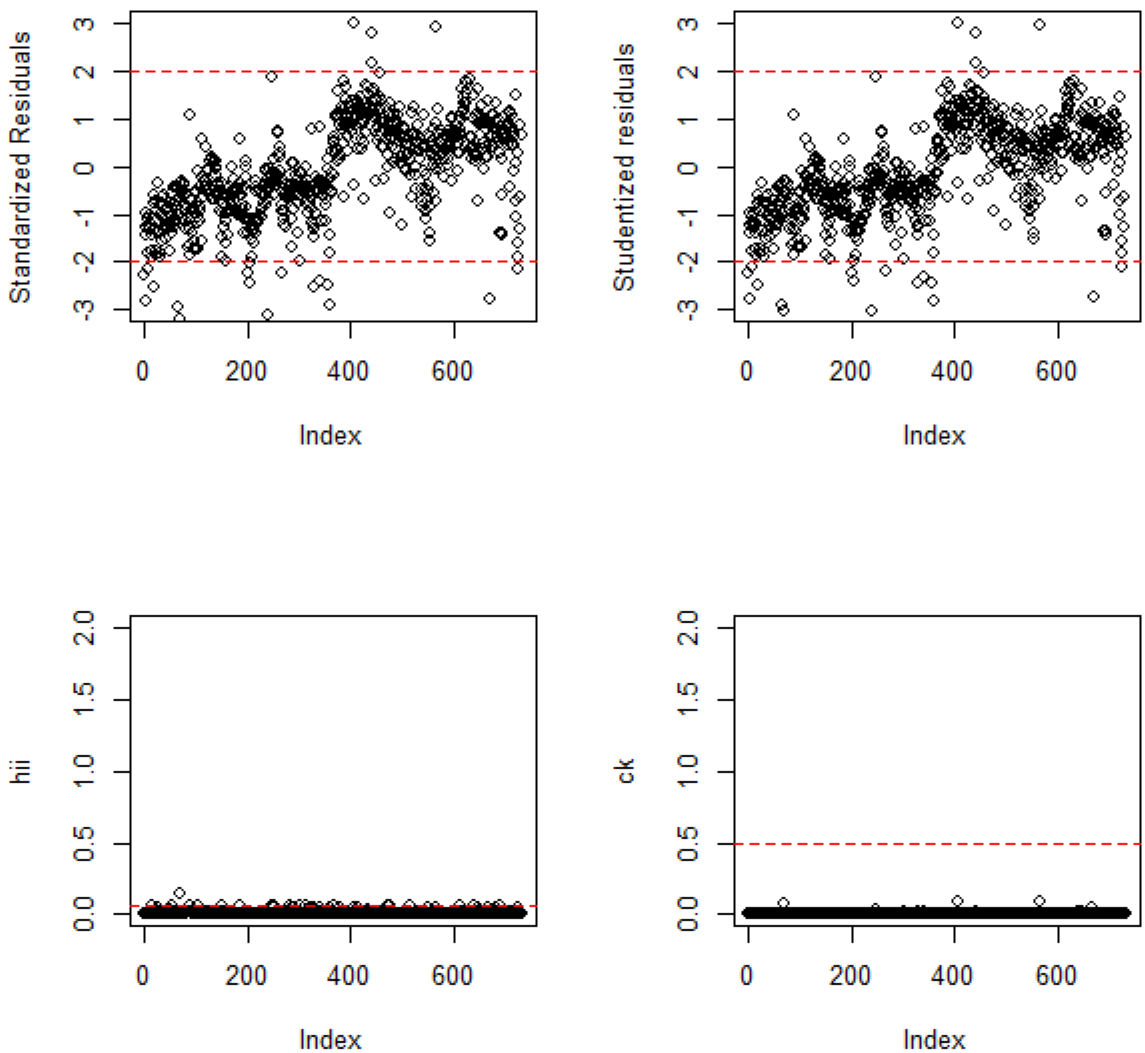
The odds of mean count of total rental bikes in Spring season is 1.28 times the odds ratio of rental bike count in Winter season, holding other predictors fixed. The odds of mean count of total rental bikes in the Summer season is ~1.11 times the odds ratio of rental bike count in the Winter season, holding other predictors fixed. The odds of mean count of total rental bikes in Fall season is 1.55 times the odds ratio of rental bike count in Winter season, holding other predictors fixed.
If it is a holiday the odds of the mean count of total rental bikes is ~0.81 times the odds ratio of rental bike count on another day, holding other predictors fixed.
For every 1 celsius increment in temperature, the odds of bike rental count increased by 495% (exp(1.784)-1), holding other variables fixed.

McFadden's R2

```
> r2=with(nb_fit_bike3,1-deviance/null.deviance) #Produced McFadden's
R2...
> r2
[1] 0.5415954
```

McFadden's R^2 says that the model describes 54% variability of data.

Residual analysis to search for outliers



According to the plots of ‘standardized residuals’ and ‘Studentized residuals’ there are several outliers. But we cannot see any high influential points in the plot of hi-leverage and cook distance.

References :

[1] Fanaee-T, Hadi, and Gama, Joao, 'Event labeling combining ensemble detectors and background knowledge', Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg.