Poisson regression analysis to investigate the associations between transmission controls measures and the number of reported cases in the first seven days of the outbreaks in cities

Chieh-Hsi Wu

Here we investigate the associations between transmission control measures and the number of reported cases in the first week of the outbreaks in cities.

```
library(R330)
```

```
## Loading required package: s20x

## Loading required package: leaps

## Loading required package: rgl

## Loading required package: lattice
```

```
library(readxl)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:s20x':
##
##     levene.test
```

```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
covid2019FilePath = "/Users/chwu/Documents/research/nCov-2019_TCM/nCoV-data.xlsx"
covid2019.df = read_excel(path = covid2019FilePath, sheet = "3resp-7days")

covid2019GroupFilePath = "/Users/jessiewu/Documents/research/phylodynamic/nCov-2019/TCM/groups.xlsx"
covid2019Group.df = read_excel(path = covid2019FilePath, sheet = "3resp-7days")
```

Processing the data

Some of the cities have a inflow from Wuhan recorded as 0, which causes calculations to run into an error when we use it as an offset variable. To resolve this issue, 0 values are changed to 10^{-6}, which is equivalent to only one person arriving to a city from Wuhan.

```
covid2019.df$new.totalflow_million = covid2019.df$totalflow_million
covid2019.df$new.totalflow_million[covid2019.df$totalflow_million == 0] = 1e-6
```

The arrival time is processed so that 31 December 2019 is coded as day 0.

```
covid2019.df$new.arr.time = covid2019.df$arr.time - 1
```

The timing of suspending intra-city public transport is processed so that 31 December 2019 is coded as day 0.

```
bus.resp.tab = table(covid2019.df$Bus.resp)
bus.resp.tab
```

```
##
##   0   1
## 207  89
```

```
bus.date.tab1 = table(covid2019.df$Bus.date[which(covid2019.df$Bus.resp==1)])
bus.date.tab1
```

```
##
## 24 25 26 27 28 29 30 31 32 33
##  3 10  5 14 20 20 11  2  3  1
```

```
covid2019.df$new.Bus.date = covid2019.df$Bus.date - 1
new.bus.date.tab1 = table(covid2019.df$new.Bus.date[which(covid2019.df$Bus.resp==1)])
new.bus.date.tab1
```

```
##
## 23 24 25 26 27 28 29 30 31 32
##  3 10  5 14 20 20 11  2  3  1
```

```
covid2019.df$new.Bus.date[which(covid2019.df$Bus.resp==0)] = 0
new.bus.date.tab = table(covid2019.df$new.Bus.date)
new.bus.date.tab
```

```
##
##   0  23  24  25  26  27  28  29  30  31  32
## 207   3  10   5  14  20  20  11   2   3   1
```

```
## Sanity check
## Codes below should return (0)
# bus.date.tab1 - new.bus.date.tab1
# (as.numeric(names(bus.date.tab1)) - 1) - as.numeric(names(new.bus.date.tab1))
# bus.date.tab1 - new.bus.date.tab[-1]
# (as.numeric(names(bus.date.tab1)) - 1) - as.numeric(names(new.bus.date.tab[-1]))
# bus.resp.tab["0"] - new.bus.date.tab["0"]
## Sanity check complete
```

The timing of suspending inter-city passenger traffic is processed so that 31 December 2019 is coded as day 0.

```
rail.resp.tab =  table(covid2019.df$Railway.resp)
rail.resp.tab
```

```
##
##   0   1
## 125 171
```

```
rail.date.tab1 = table(covid2019.df$Railway.date[which(covid2019.df$Railway.resp == 1)])
rail.date.tab1
```

```
##
## 24 25 26 27 28 29 30 31 32
##  1 10 14 48 66 21  8  1  2
```

```r
covid2019.df$new.Railway.date = covid2019.df$Railway.date - 1
new.rail.date.tab1 = table(covid2019.df$new.Railway.date[which(covid2019.df$Railway.resp == 1)])
new.rail.date.tab1
```

```
##
## 23 24 25 26 27 28 29 30 31
##  1 10 14 48 66 21  8  1  2
```

```r
covid2019.df$new.Railway.date[which(covid2019.df$Railway.resp == 0)] = 0
new.rail.date.tab = table(covid2019.df$new.Railway.date)
new.rail.date.tab
```

```
##
##   0  23  24  25  26  27  28  29  30  31
## 125   1  10  14  48  66  21   8   1   2
```

```r
## Sanity check
## Codes below should return 0(s)
# rail.date.tab1 - new.rail.date.tab1
# (as.numeric(names(rail.date.tab1)) - 1) - as.numeric(names(new.rail.date.tab1))
# rail.date.tab1 - new.rail.date.tab[-1]
# (as.numeric(names(rail.date.tab1)) - 1) - as.numeric(names(new.rail.date.tab[-1]))
# rail.resp.tab["0"] - new.rail.date.tab["0"]
## Sanity check complete
```

The timing of closure of entertainment venues and banning public gathering is processed so that 31 December 2019 is coded as day 0.

```r
enter.resp.tab = table(covid2019.df$Enter.resp)
enter.resp.tab
```

```
##
##   0   1
## 127 169
```

```r
enter.date.tab1 = table(covid2019.df$Enter.date[which(covid2019.df$Enter.resp == 1)])
enter.date.tab1
```

```
##
## 24 25 26 27 28 29 30 32 35
##  5 64 42 27  5  7 16  2  1
```

```r
covid2019.df$new.Enter.date = covid2019.df$Enter.date - 1
new.enter.date.tab1 = table(covid2019.df$new.Enter.date[which(covid2019.df$Enter.resp == 1)])
new.enter.date.tab1
```

```
##
## 23 24 25 26 27 28 29 31 34
##  5 64 42 27  5  7 16  2  1
```

```r
covid2019.df$new.Enter.date[which(covid2019.df$Enter.resp == 0)] = 0
new.enter.date.tab = table(covid2019.df$new.Enter.date)
new.enter.date.tab
```

```
##
##   0  23  24  25  26  27  28  29  31  34
```

```
## 127   5  64  42  27   5   7  16   2   1
```

```r
## Sanity check
## Codes should return 0(s)
# enter.date.tab1 - new.enter.date.tab1
# (as.numeric(names(enter.date.tab1)) - 1) - as.numeric(names(new.enter.date.tab1))
# enter.date.tab1 - new.enter.date.tab[-1]
# (as.numeric(names(enter.date.tab1)) - 1) - as.numeric(names(new.enter.date.tab[-1]))
# enter.resp.tab["0"] - new.enter.date.tab["0"]
## Sanity check complete
```

Regression analysis

Poisson regression

The analysis using a Poisson regression model is presented below.

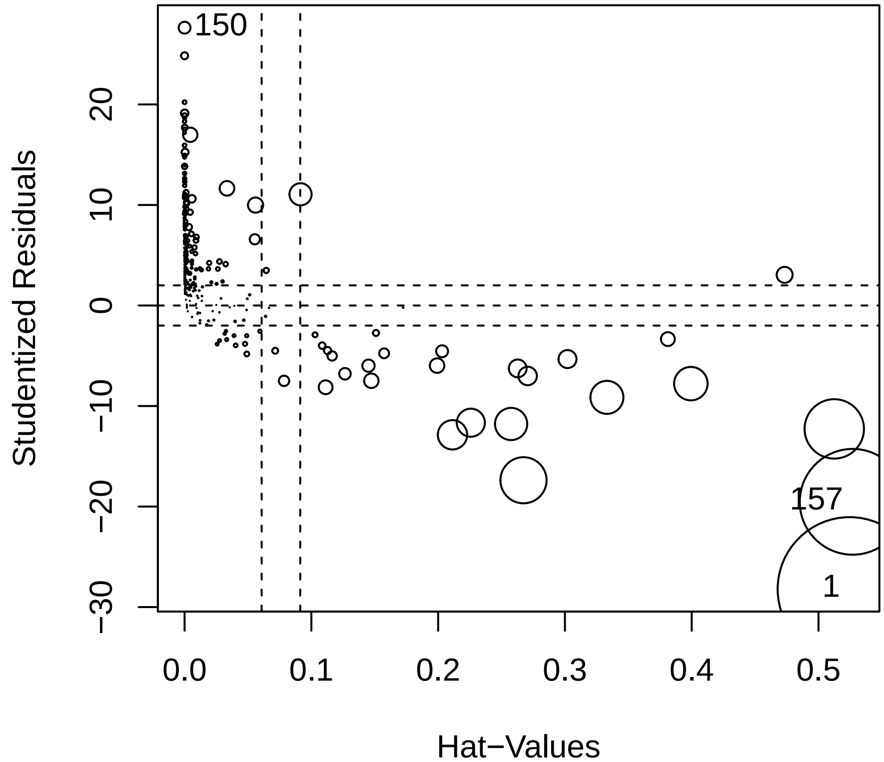
```r
yfc.resp.date.glm1 = glm(sevendays.cucase ~ new.arr.time + log10.Dis.WH+
                          Bus.resp + new.Bus.date +
                          Railway.resp + new.Railway.date +
                          Enter.resp + new.Enter.date +
                          offset(log(Pop_million_2018*new.totalflow_million)),
                        family = "poisson",
                        data = covid2019.df)
summary(yfc.resp.date.glm1)
```

```
##
## Call:
## glm(formula = sevendays.cucase ~ new.arr.time + log10.Dis.WH +
##     Bus.resp + new.Bus.date + Railway.resp + new.Railway.date +
##     Enter.resp + new.Enter.date + offset(log(Pop_million_2018 *
##     new.totalflow_million)), family = "poisson", data = covid2019.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -21.522    0.996    3.436    6.828   27.547
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.012402   0.262692 -34.308  < 2e-16 ***
## new.arr.time      0.449957   0.008711  51.651  < 2e-16 ***
## log10.Dis.WH      0.262743   0.058299   4.507 6.58e-06 ***
## Bus.resp         -3.798081   0.380528  -9.981  < 2e-16 ***
## new.Bus.date      0.153348   0.014383  10.662  < 2e-16 ***
## Railway.resp      2.359289   0.488727   4.827 1.38e-06 ***
## new.Railway.date -0.077271   0.018554  -4.165 3.12e-05 ***
## Enter.resp       -6.762999   0.317329 -21.312  < 2e-16 ***
## new.Enter.date    0.272404   0.012282  22.179  < 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: 20643  on 295  degrees of freedom
## Residual deviance: 15279  on 287  degrees of freedom
```

```
## AIC: 16453
##
## Number of Fisher Scoring iterations: 11
```

There appears to be three influential points.

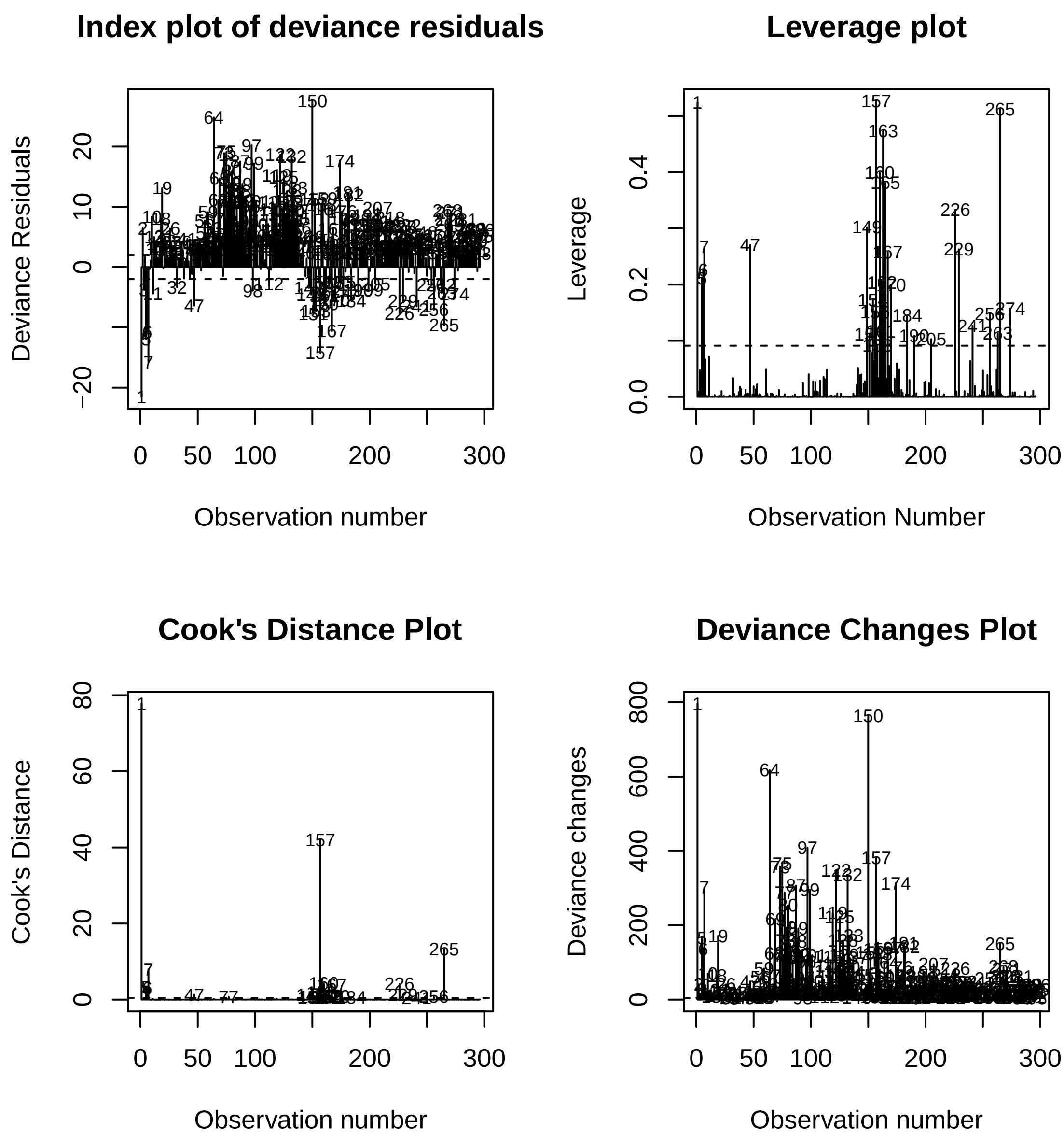
```r
influencePlot(yfc.resp.date.glm1)
```




```
##       StudRes          Hat      CookD
## 1   -28.21211 5.245598e-01 77.7548556
## 150  27.63330 7.147685e-06  0.5279906
## 157 -19.52850 5.269354e-01 41.9935411
```

Below is a visualisation of the influential points indicators.

```r
influenceplots(yfc.resp.date.glm1)
```



Removing the three influential points does not affect the conclusions.

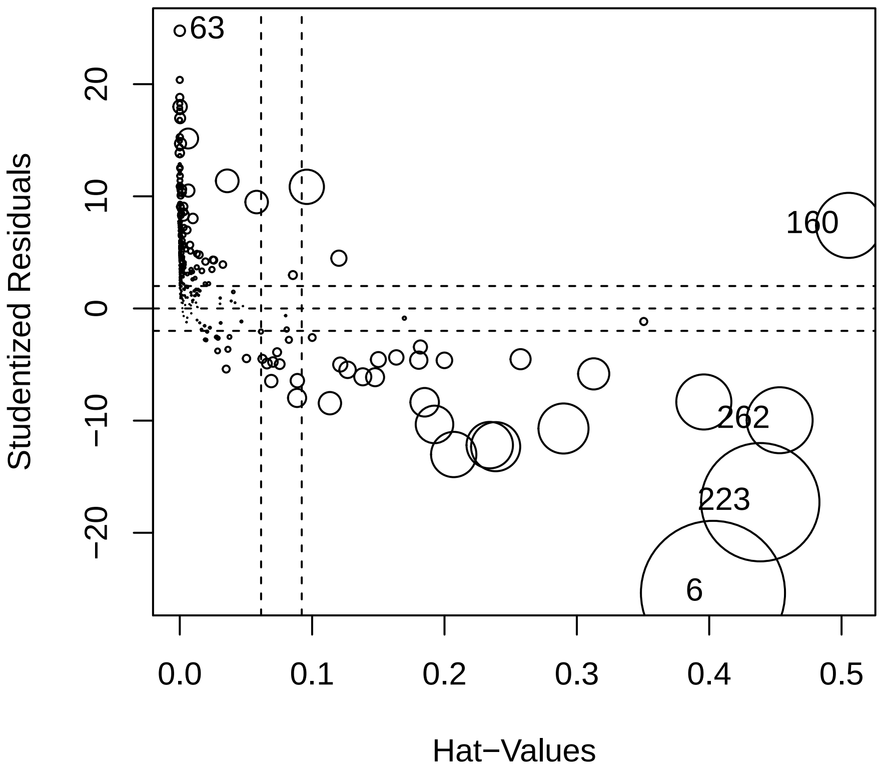
```
yfc.resp.date.glm2 = glm(sevendays.cucase ~ new.arr.time + log10.Dis.WH+
                          Bus.resp + new.Bus.date +
                          Railway.resp + new.Railway.date +
                          Enter.resp + new.Enter.date +
                          offset(log(Pop_million_2018*new.totalflow_million)),
                        family = "poisson",
                        data = covid2019.df[-c(1, 150, 157),])
summary(yfc.resp.date.glm2)
```

```
##
## Call:
## glm(formula = sevendays.cucase ~ new.arr.time + log10.Dis.WH +
##     Bus.resp + new.Bus.date + Railway.resp + new.Railway.date +
##     Enter.resp + new.Enter.date + offset(log(Pop_million_2018 *
##     new.totalflow_million)), family = "poisson", data = covid2019.df[-c(1,
```

```
##     150, 157), ])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -21.6415    0.6517    3.1769    6.4401   24.7307
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.918294   0.246656 -36.157  < 2e-16 ***
## new.arr.time      0.456095   0.008042  56.713  < 2e-16 ***
## log10.Dis.WH      0.307842   0.062989   4.887 1.02e-06 ***
## Bus.resp         -4.665591   0.403407 -11.565  < 2e-16 ***
## new.Bus.date      0.166875   0.015129  11.030  < 2e-16 ***
## Railway.resp      2.983175   0.502211   5.940 2.85e-09 ***
## new.Railway.date -0.094721   0.019133  -4.951 7.40e-07 ***
## Enter.resp       -4.088143   0.354539 -11.531  < 2e-16 ***
## new.Enter.date    0.162662   0.013953  11.658  < 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: 18049  on 292  degrees of freedom
## Residual deviance: 13501  on 284  degrees of freedom
## AIC: 14656
##
## Number of Fisher Scoring iterations: 10
```

There appears to be more observations with outstanding Cook's distances.

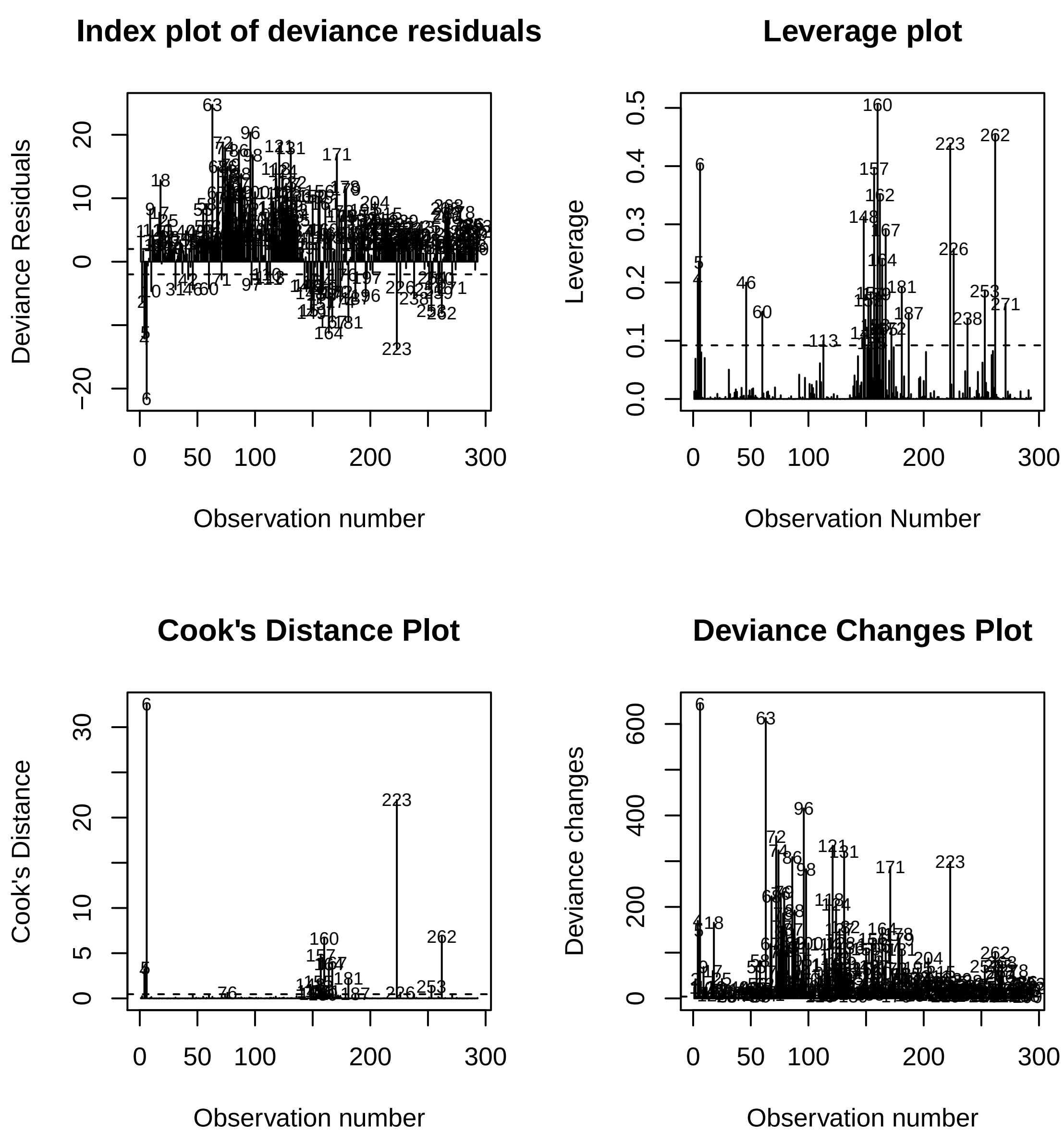
```r
influencePlot(yfc.resp.date.glm2)
```




```
##         StudRes          Hat      CookD
## 6   -25.362400 4.028576e-01 32.5434952
## 63   24.762349 7.183167e-07  0.1739565
## 160   7.408214 5.053251e-01  6.6349053
## 223 -17.273884 4.385894e-01 21.9439577
## 262  -9.965340 4.532306e-01  6.8218686
```

The plots below show that the outstanding Cook's distance a substantially far apart from the rest.

```
influenceplots(yfc.resp.date.glm2)
```



The conclusions are not affected by removing the observations with the outstanding Cook's distances.

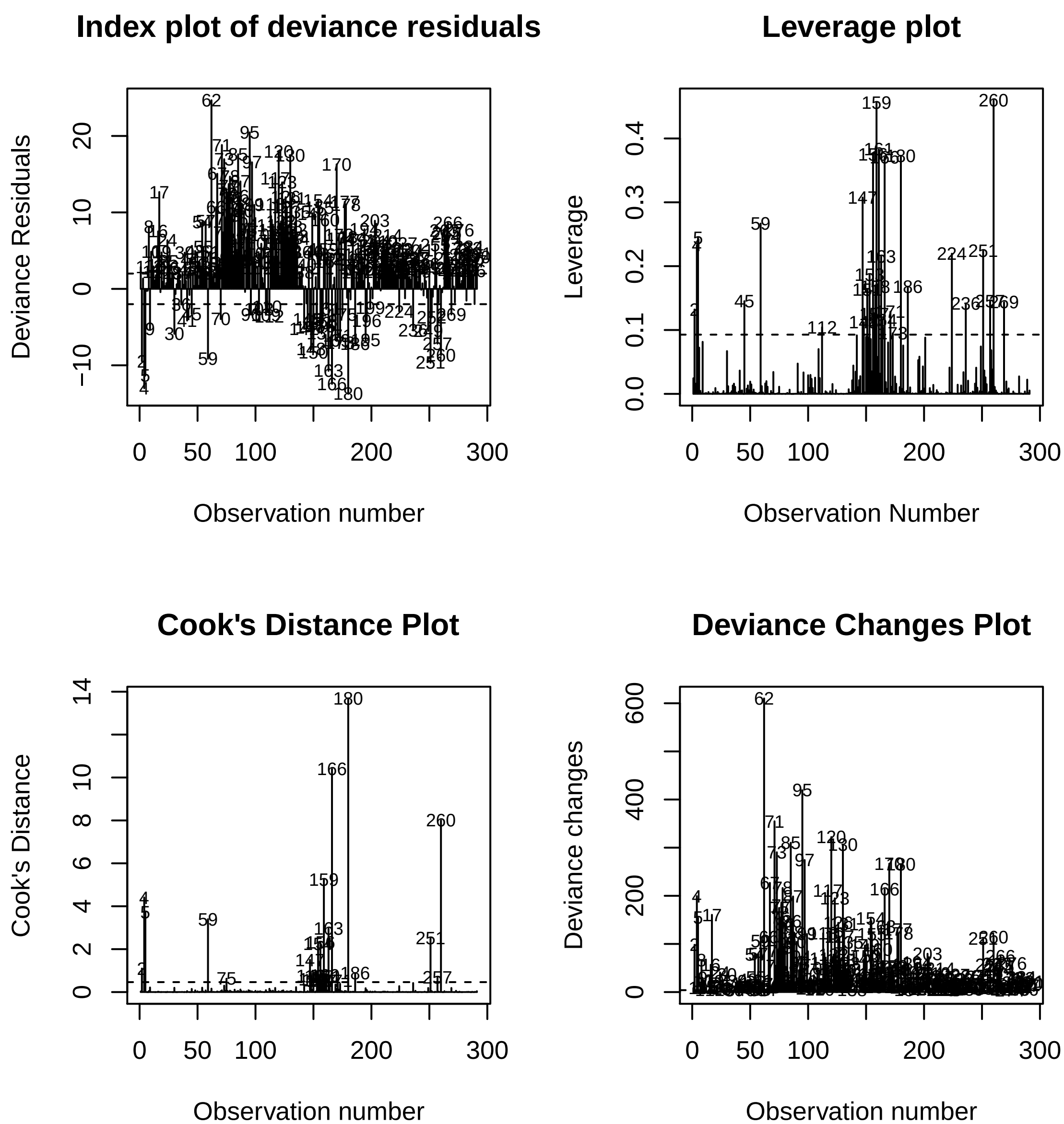
```r
yfc.resp.date.glm = glm(sevendays.cucase ~ new.arr.time + log10.Dis.WH+
                          Bus.resp + new.Bus.date +
                          Railway.resp + new.Railway.date +
                          Enter.resp + new.Enter.date +
                          offset(log(Pop_million_2018*new.totalflow_million)),
                        family = "poisson",
                        data = covid2019.df[-c(1, 7, 150, 157, 226),])
summary(yfc.resp.date.glm)
```

```
##
## Call:
## glm(formula = sevendays.cucase ~ new.arr.time + log10.Dis.WH +
##     Bus.resp + new.Bus.date + Railway.resp + new.Railway.date +
##     Enter.resp + new.Enter.date + offset(log(Pop_million_2018 *
##     new.totalflow_million)), family = "poisson", data = covid2019.df[-c(1,
```

```
##     7, 150, 157, 226), ])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -13.7343    0.3432    2.6513    5.9258   24.6688
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.102497   0.234468 -38.822  < 2e-16 ***
## new.arr.time      0.441667   0.007643  57.785  < 2e-16 ***
## log10.Dis.WH      0.610339   0.061075   9.993  < 2e-16 ***
## Bus.resp         -3.504380   0.397201  -8.823  < 2e-16 ***
## new.Bus.date      0.112641   0.015061   7.479 7.48e-14 ***
## Railway.resp      2.325827   0.491613   4.731 2.23e-06 ***
## new.Railway.date -0.065610   0.018747  -3.500 0.000466 ***
## Enter.resp       -2.275964   0.359759  -6.326 2.51e-10 ***
## new.Enter.date    0.086014   0.014254   6.034 1.60e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16982  on 290  degrees of freedom
## Residual deviance: 12618  on 282  degrees of freedom
## AIC: 13762
##
## Number of Fisher Scoring iterations: 10
```

There still are several data points with large Cook's Distances. However there are a large number of these and by standard practice, it would be too many to remove. In general, we should only remove one or two influential points, as this is to check that the results observed is not just purely the result of one or two points. If too many points are removed, the defies the purpose of this check.

```
influenceplots(yfc.resp.date.glm)
```

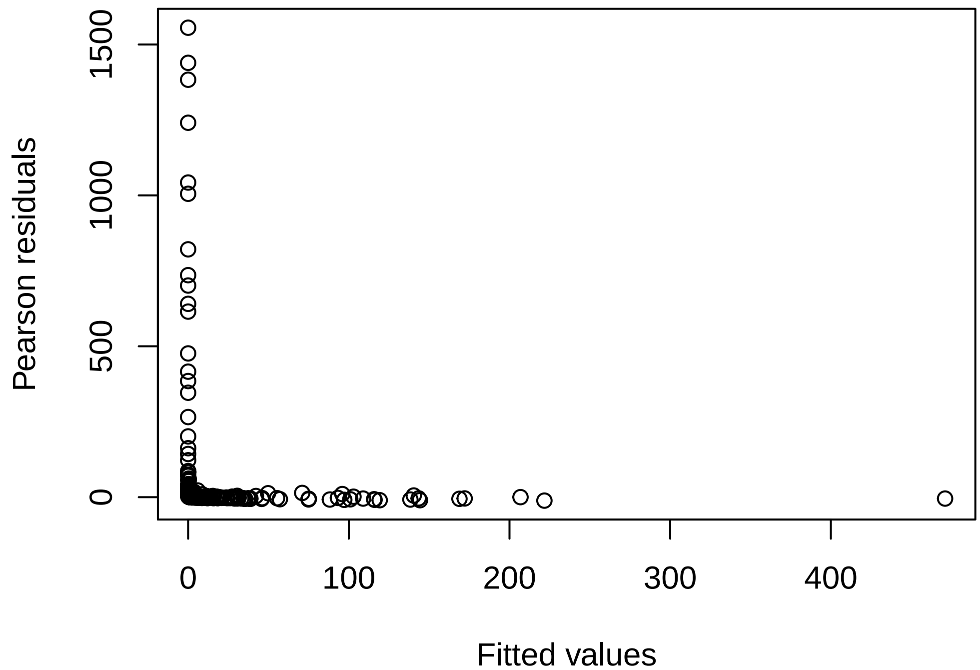



```
yfc.resp.date.glm.est = coef(summary(yfc.resp.date.glm))
yfc.resp.date.glm.est.tab = cbind(yfc.resp.date.glm.est[,"Estimate"],
yfc.resp.date.glm.est[,"Estimate"]-1.96*yfc.resp.date.glm.est[,"Std. Error"],
yfc.resp.date.glm.est[,"Estimate"]+1.96*yfc.resp.date.glm.est[,"Std. Error"])
colnames(yfc.resp.date.glm.est.tab) = c("Estimate", "Lower 95% CI", "Upper 95% CI")
round(yfc.resp.date.glm.est.tab, 2)
```

```
##                  Estimate Lower 95% CI Upper 95% CI
## (Intercept)         -9.10        -9.56        -8.64
## new.arr.time         0.44         0.43         0.46
## log10.Dis.WH         0.61         0.49         0.73
## Bus.resp            -3.50        -4.28        -2.73
## new.Bus.date         0.11         0.08         0.14
## Railway.resp         2.33         1.36         3.29
## new.Railway.date    -0.07        -0.10        -0.03
## Enter.resp          -2.28        -2.98        -1.57
## new.Enter.date       0.09         0.06         0.11
```

If the model is correct, the Pearson residuals should have constant spread across fitted values. However, the plot below clearly shows that this is not the case—there is evident heteroscedasticity in the pearson residuals.

```
plot((fitted(yfc.resp.date.glm)),
     residuals(yfc.resp.date.glm, type= "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals")
```



Due to these issues we report the parameter estimates from the regression with influential points excluded as that give us the most conservative estimates.

Quasi-Poisson regression

A quasi-Poisson regression model is fitted to see whether it can rectify this problem. This model uses a quasi-likelihood and suggests that there is no evidence that any of the coefficients are significant. This is dubious as we demonstrate in the next step.

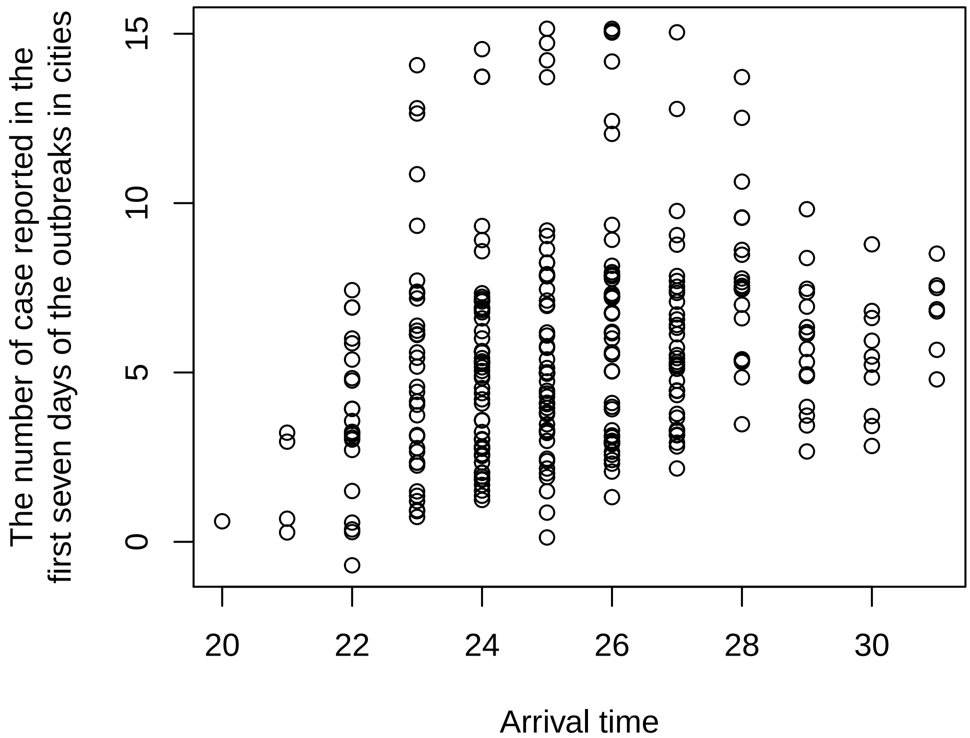
```
yfc.resp.date.qp.glm = glm(sevendays.cucase ~ new.arr.time + log10.Dis.WH+
                          Bus.resp + new.Bus.date +
                          Railway.resp + new.Railway.date +
                          Enter.resp + new.Enter.date +
                          offset(log(Pop_million_2018*new.totalflow_million)),
                        family = "quasipoisson",
                        data = covid2019.df[-c(1, 7, 150, 157, 226),])
summary(yfc.resp.date.qp.glm)
```

```
##
## Call:
## glm(formula = sevendays.cucase ~ new.arr.time + log10.Dis.WH +
##     Bus.resp + new.Bus.date + Railway.resp + new.Railway.date +
##     Enter.resp + new.Enter.date + offset(log(Pop_million_2018 *
##     new.totalflow_million)), family = "quasipoisson", data = covid2019.df[-c(1,
##     7, 150, 157, 226), ])
##
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -13.7343   0.3432   2.6513   5.9258  24.6688
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -9.10250   51.21896  -0.178    0.859
## new.arr.time       0.44167    1.66965   0.265    0.792
## log10.Dis.WH       0.61034   13.34172   0.046    0.964
## Bus.resp          -3.50438   86.76741  -0.040    0.968
## new.Bus.date       0.11264    3.28995   0.034    0.973
## Railway.resp       2.32583  107.39152   0.022    0.983
## new.Railway.date  -0.06561    4.09513  -0.016    0.987
## Enter.resp        -2.27596   78.58832  -0.029    0.977
## new.Enter.date     0.08601    3.11374   0.028    0.978
##
## (Dispersion parameter for quasipoisson family taken to be 47719.25)
##
##     Null deviance: 16982  on 290  degrees of freedom
## Residual deviance: 12618  on 282  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 10
```

We consider the arrival time, and plot it against the logarithem of the number of reported cases divided by the population size and inflow from Wuhan. It is apparente that the arrival time increases as the number of reported cases increases.

```r
par(mar = c(5, 6, 2, 1) + 0.2)
log.std.seven.cucase = log(covid2019.df$sevendays.cucase/
                             (covid2019.df$Pop_million_2018*covid2019.df$new.totalflow_million))
plot(covid2019.df$arr.time,log.std.seven.cucase,
     xlab = "Arrival time",
     ylab = "The number of case reported in the\nfirst seven days of the outbreaks in cities")
```



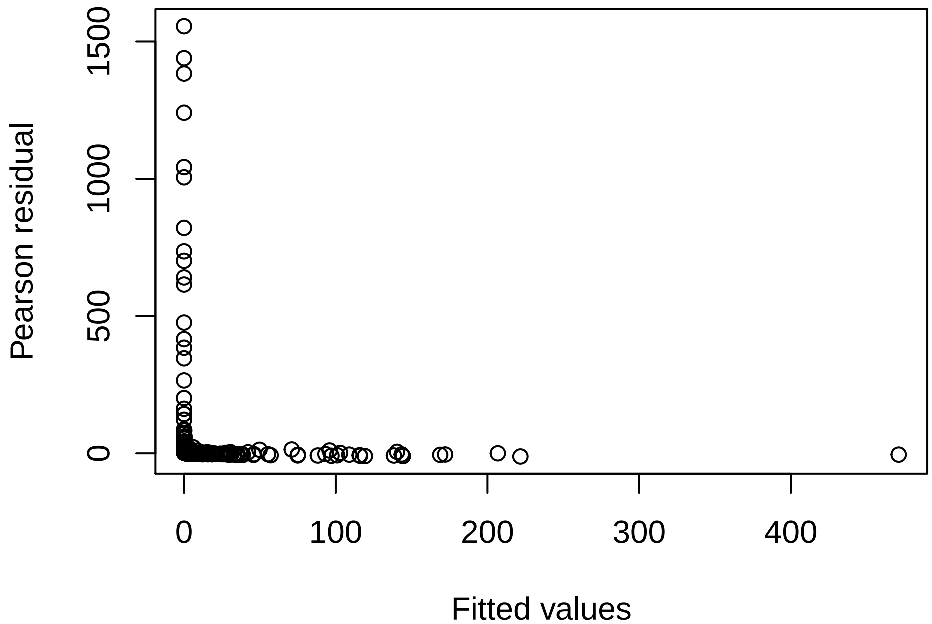
However, the quasi-Poisson regression again indicates the no evidence for coefficient of arrival time. This could be because of the heteroscedasticity in the Pearson residuals is extremely severe, the quasi-Poisson regression is over-compensating and is unable to detect any signal in the data efficiently. Furthermore, the plot below shows that the quasi-Poisson model provides no indication that the heteroscedasticity in the Pearson residuals have been rectified. (Pearson residuals in the plot is from the full quasi-Poisson model including all the control measure varaibles.) Therefore, we are also uncertain about the reliability of the quasi-Poisson regression analysis.

```
yfc.arr.time.qp.glm = glm(sevendays.cucase ~ new.arr.time +
                          offset(log(Pop_million_2018*new.totalflow_million)),
                        family = "quasipoisson",
                        data = covid2019.df[-c(1, 7, 150, 157, 226),])
summary(yfc.arr.time.qp.glm)
```

```
##
## Call:
## glm(formula = sevendays.cucase ~ new.arr.time + offset(log(Pop_million_2018 *
##     new.totalflow_million)), family = "quasipoisson", data = covid2019.df[-c(1,
##     7, 150, 157, 226), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -15.205    1.125    3.542    6.627   24.916
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.7062    50.3550  -0.153    0.878
## new.arr.time   0.4385     2.1250   0.206    0.837
##
## (Dispersion parameter for quasipoisson family taken to be 78984.86)
##
```

```
##     Null deviance: 16982  on 290  degrees of freedom
## Residual deviance: 13937  on 289  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 7
```

```
par(mar = c(5, 5, 5, 2) + 0.2)
plot((fitted(yfc.resp.date.qp.glm)),
     residuals(yfc.resp.date.qp.glm, type= "pearson"),
     xlab = "Fitted values",
     ylab = "Pearson residual")
```



Negative binomial regression

A negative regression model is fitted to see whether it can rectify the heteroscedasticity in the Pearson residuals.

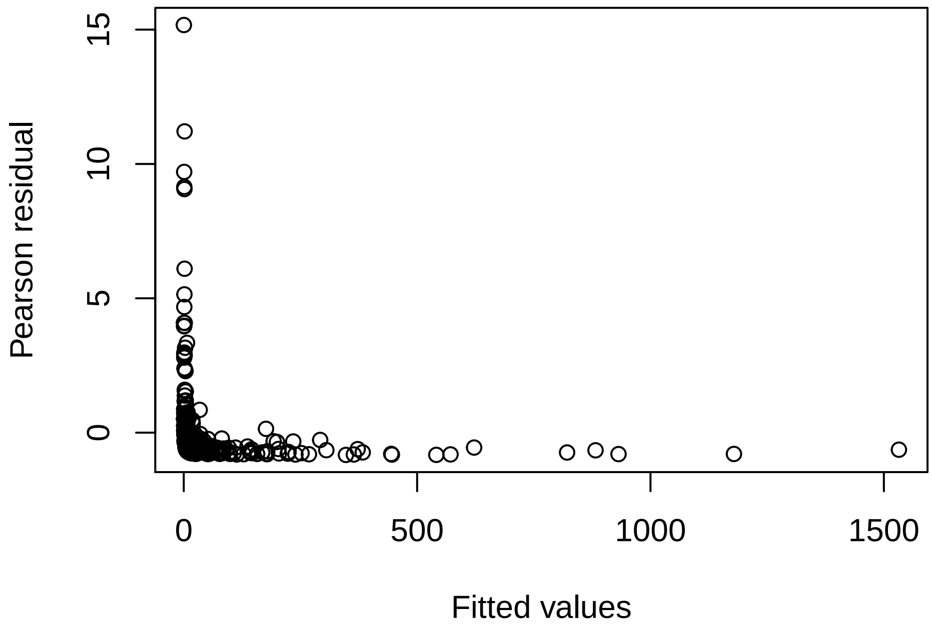
```
library(MASS)
yfc.resp.date.nb.glm = glm.nb(sevendays.cucase ~ new.arr.time + log10.Dis.WH+
                             Bus.resp + new.Bus.date +
                             Railway.resp + new.Railway.date +
                             Enter.resp + new.Enter.date +
                             offset(log10(Pop_million_2018*new.totalflow_million)),
                           data = covid2019.df[-c(1, 7, 150, 157, 226),])
summary(yfc.resp.date.nb.glm)
```

```
##
## Call:
## glm.nb(formula = sevendays.cucase ~ new.arr.time + log10.Dis.WH +
##     Bus.resp + new.Bus.date + Railway.resp + new.Railway.date +
##     Enter.resp + new.Enter.date + offset(log10(Pop_million_2018 *
##     new.totalflow_million)), data = covid2019.df[-c(1, 7, 150,
##     157, 226), ], init.theta = 0.7009678275, link = log)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max  
## -2.2634  -1.0328  -0.5568   0.0296   4.3927  
## 
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       7.088496   1.107077   6.403 1.52e-10 ***
## new.arr.time      0.116918   0.034715   3.368 0.000757 ***
## log10.Dis.WH     -2.240389   0.310856  -7.207 5.71e-13 ***
## Bus.resp         -8.064656   2.301858  -3.504 0.000459 ***
## new.Bus.date      0.289603   0.083825   3.455 0.000551 ***
## Railway.resp     -5.673621   2.338304  -2.426 0.015250 *  
## new.Railway.date  0.264473   0.087200   3.033 0.002422 ** 
## Enter.resp        0.361755   1.521028   0.238 0.812008    
## new.Enter.date   -0.002557   0.059153  -0.043 0.965527    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(0.701) family taken to be 1)
## 
##     Null deviance: 511.31  on 290  degrees of freedom
## Residual deviance: 361.83  on 282  degrees of freedom
## AIC: 2289.3
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  0.7010 
##           Std. Err.:  0.0550 
## 
##  2 x log-likelihood:  -2269.3030
```

The plot below shows that the Pearson residuals of the negative-binomial regression still display heteroscedasticity. And therefore, we are also uncertain about the reliability of the results from the negative binomial regression.

```
par(mar = c(5, 5, 5, 2) + 0.2)
plot((fitted(yfc.resp.date.nb.glm)),
     residuals(yfc.resp.date.nb.glm, type= "pearson"),
     xlab = "Fitted values",
     ylab = "Pearson residual")
```



Pearson residual
15
10
5
0
0
500
1000
1500
Fitted values