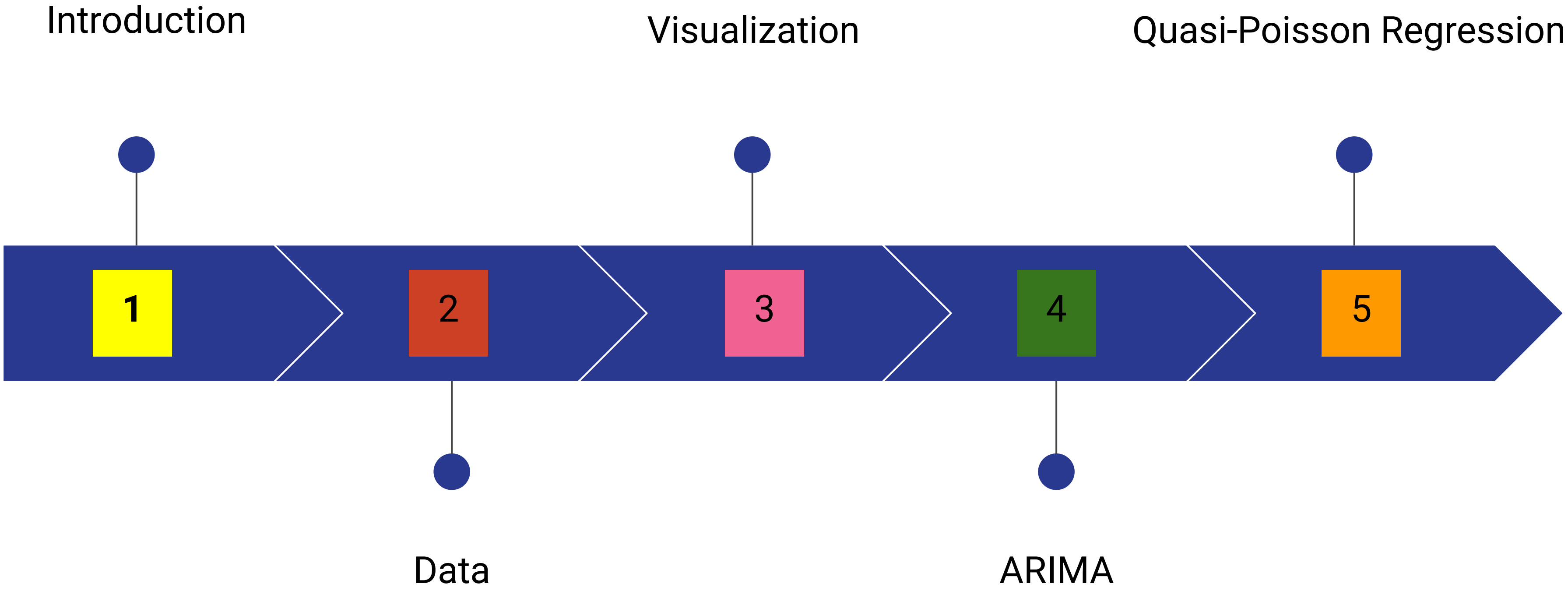
Mexico Covid-19

Alessandro Scardoni, Azad Sadr

July 28, 2020



1 Introduction

2 Data

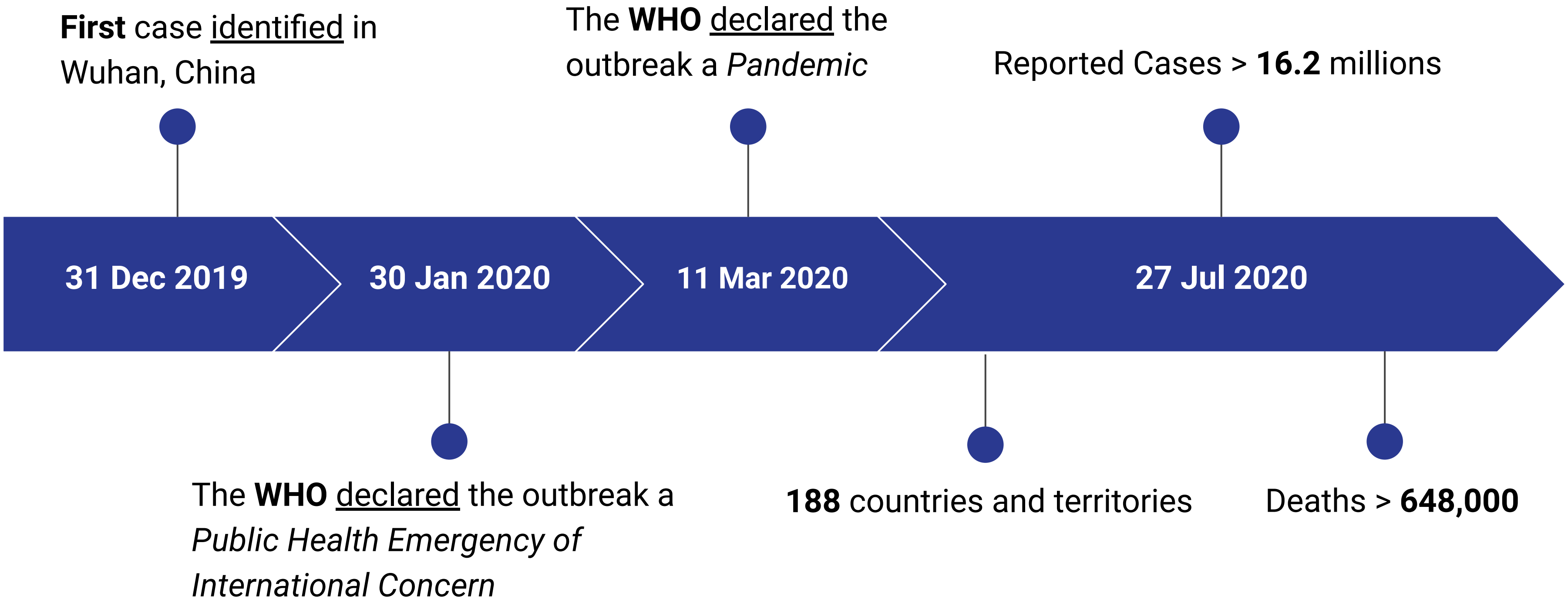
3 Visualization

4 ARIMA

5 Quasi-Poisson Regression

Introduction

The **COVID-19** is global pandemic of coronavirus disease caused by SARS-CoV-2



First case identified in Wuhan, China

31 Dec 2019

30 Jan 2020

The **WHO** declared the outbreak a *Public Health Emergency of International Concern*

11 Mar 2020

The **WHO** declared the outbreak a *Pandemic*

27 Jul 2020

Reported Cases > **16.2** millions

188 countries and territories

Deaths > **648,000**

The Problem Statement

What?

COVID-19 is spreading very easily and sustainably between people.

Accurate outbreak prediction models is essential to obtain insights into the likely spread and consequences of infectious diseases.

Where?

South of Mexico:

- Campeche
- Chiapas
- Guerrero
- Oaxaca
- Quintana Roo
- Tabasco
- Veracruz
- Yucatán

How?

Propose a statistical model to estimate the dynamic of outbreak and forecasting the future No. of the confirmed cases.

Data

Data

Data Acquisition

The dataset is taken from Mexico government website, which is freely available online in following website:

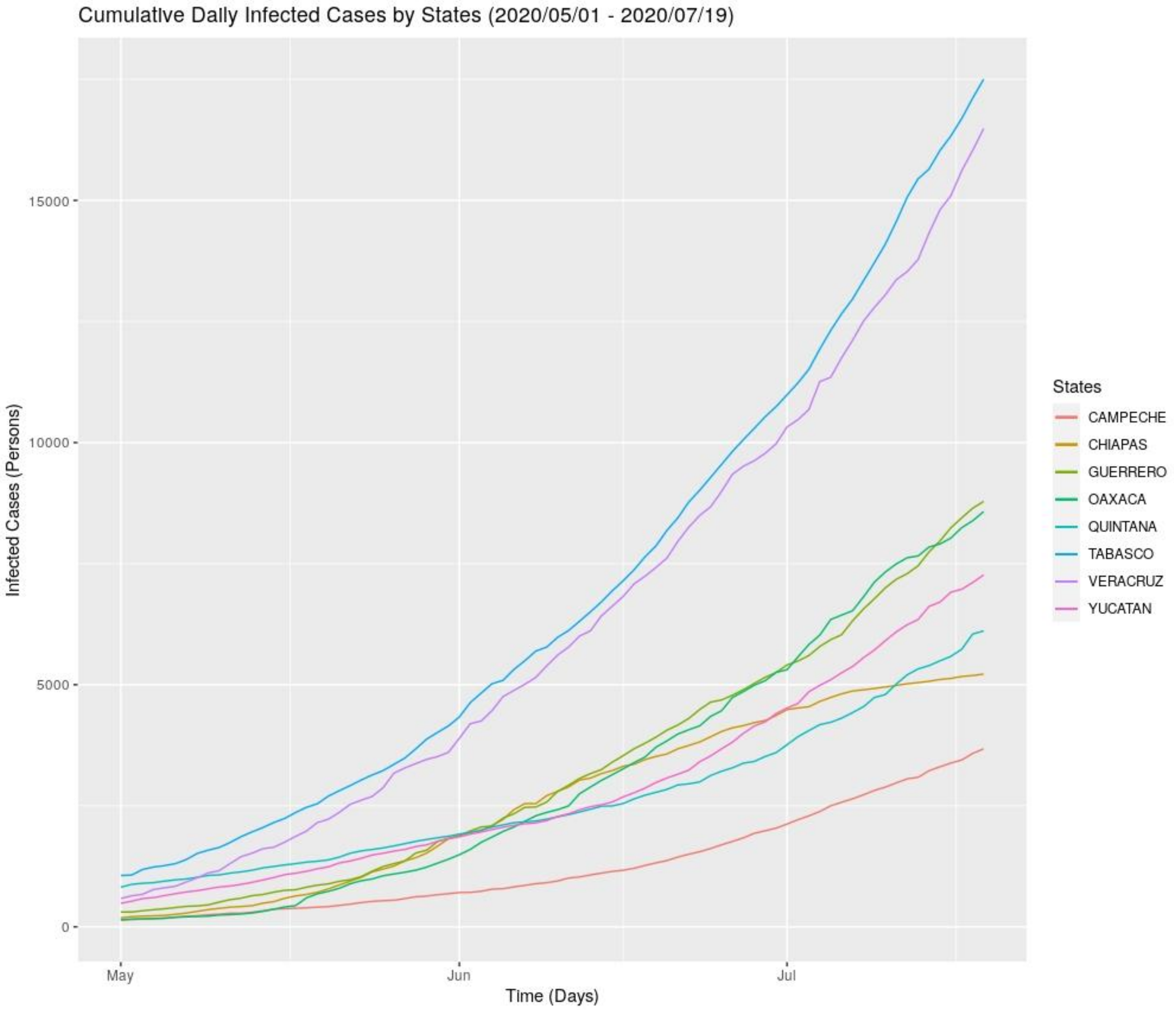
https://www.gob.mx/salud/documentos/datos-abiertos-bases-historicas-direccion-general-de-epidemiologia

Data Description

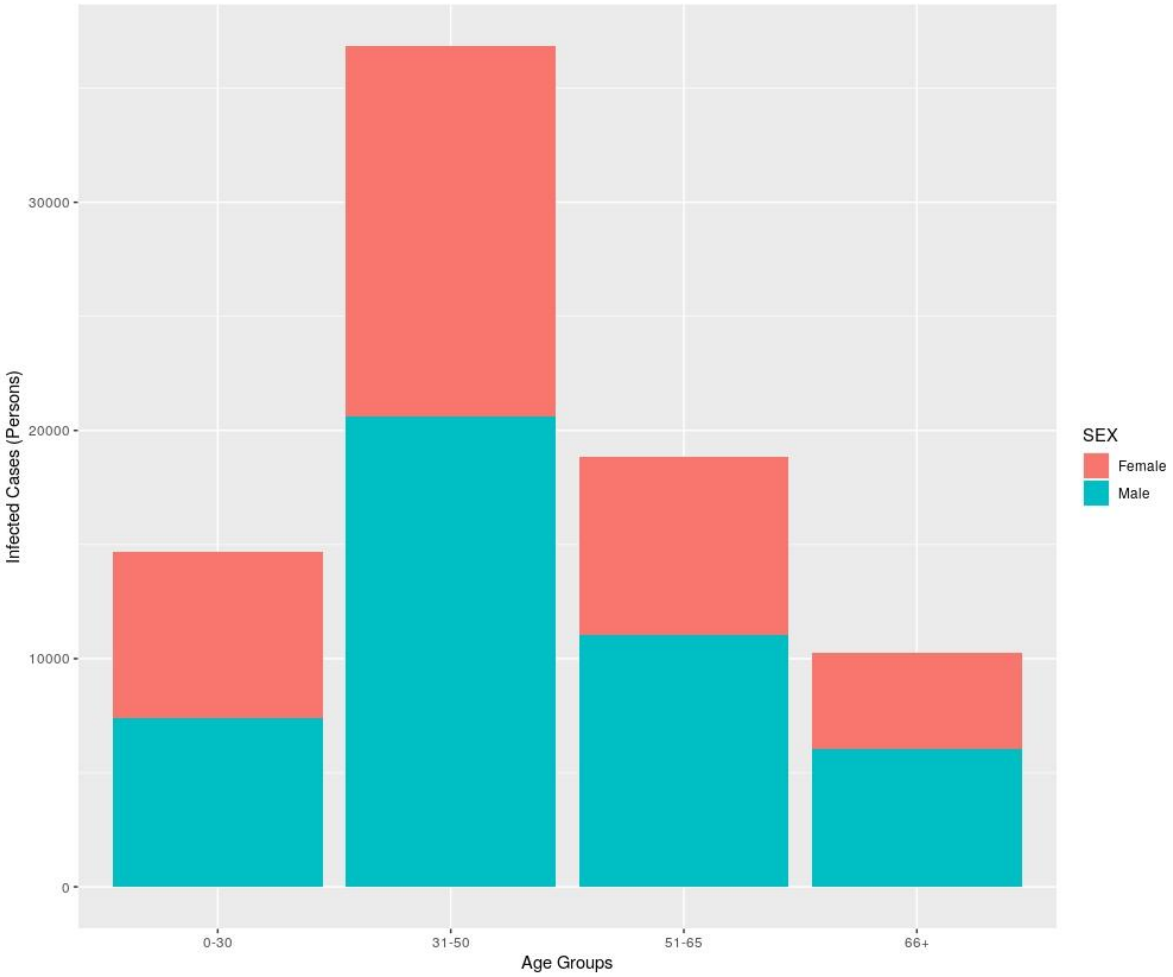
Preprocessing data with R and extract the variables of interest:

- Day counts from May 1 to July 24, 2020.
- Daily confirmed cases.
- Daily cumulative confirmed cases.
- Previous Day cumulative confirmed cases.

Visualization


Cumulative Daily Infected Cases by States (2020/05/01 - 2020/07/19)
Infected Cases (Persons)
15000
10000
5000
0
May
Jun
Jul
Time (Days)
States
CAMPECHE
CHIAPAS
GUERRERO
OAXACA
QUINTANA
TABASCO
VERACRUZ
YUCATAN

Cumulative Daily Infected Cases by States (2020/05/01 - 2020/07/19)



Modeling

The aim is to find the best kind of model to work with the available data, in order to obtain a prediction of the number of daily cases for the south Mexico.

ARIMA(p, d, q)

- Autoregressive (AR)
- Integrated (I)
- Moving Average (MA)

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

- p: The number of lag observations included in the model
- d: The number of times that the raw observations are differenced
- q: The size of the moving average window

Stationarity

Stationary time series: statistical properties are all constant over time

Time series are not stationary if have:

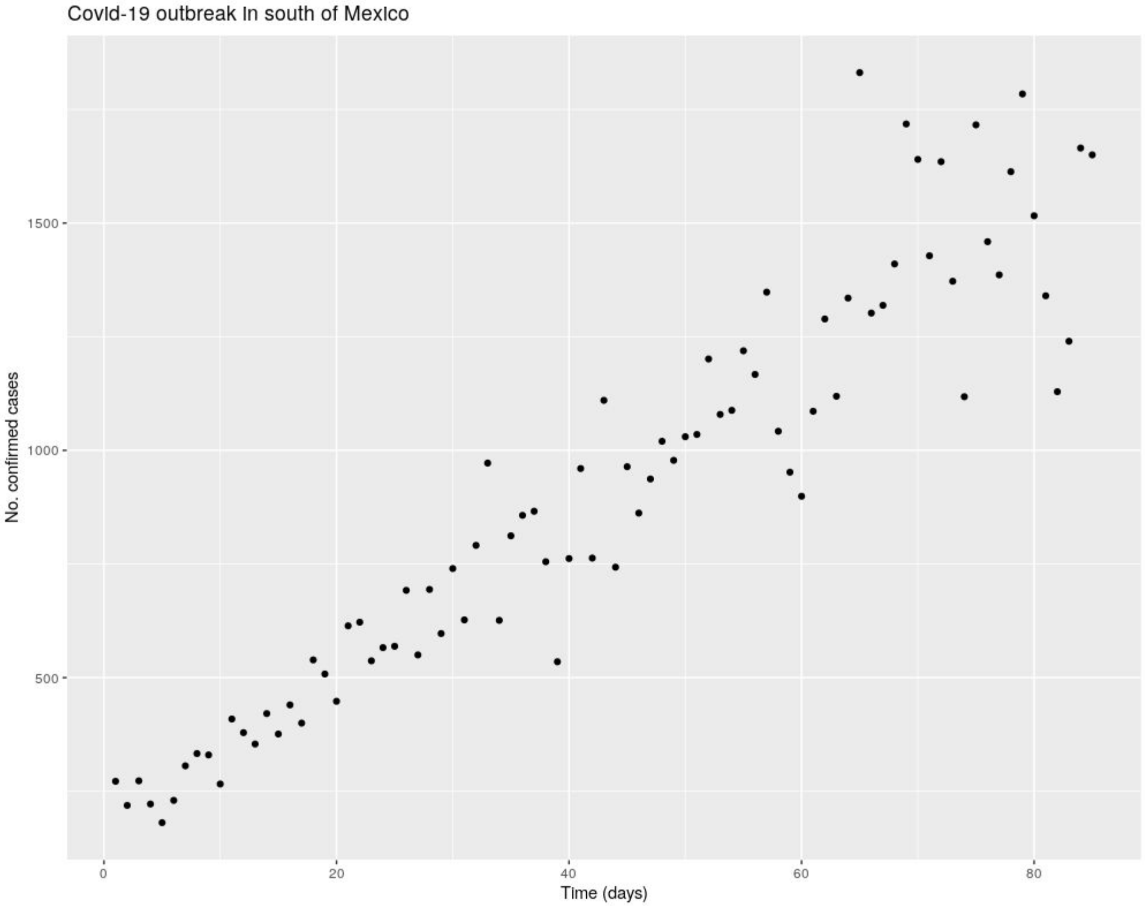
- Trends (exclude unpredictable cyclic behaviour),
- Inconstant variance
- Predictable patterns in the long-term (Seasonality)

Stationarity

Time Series Plot

No Seasonality
+
Inconsistent Variance
+
Trend

Non-Stationarity

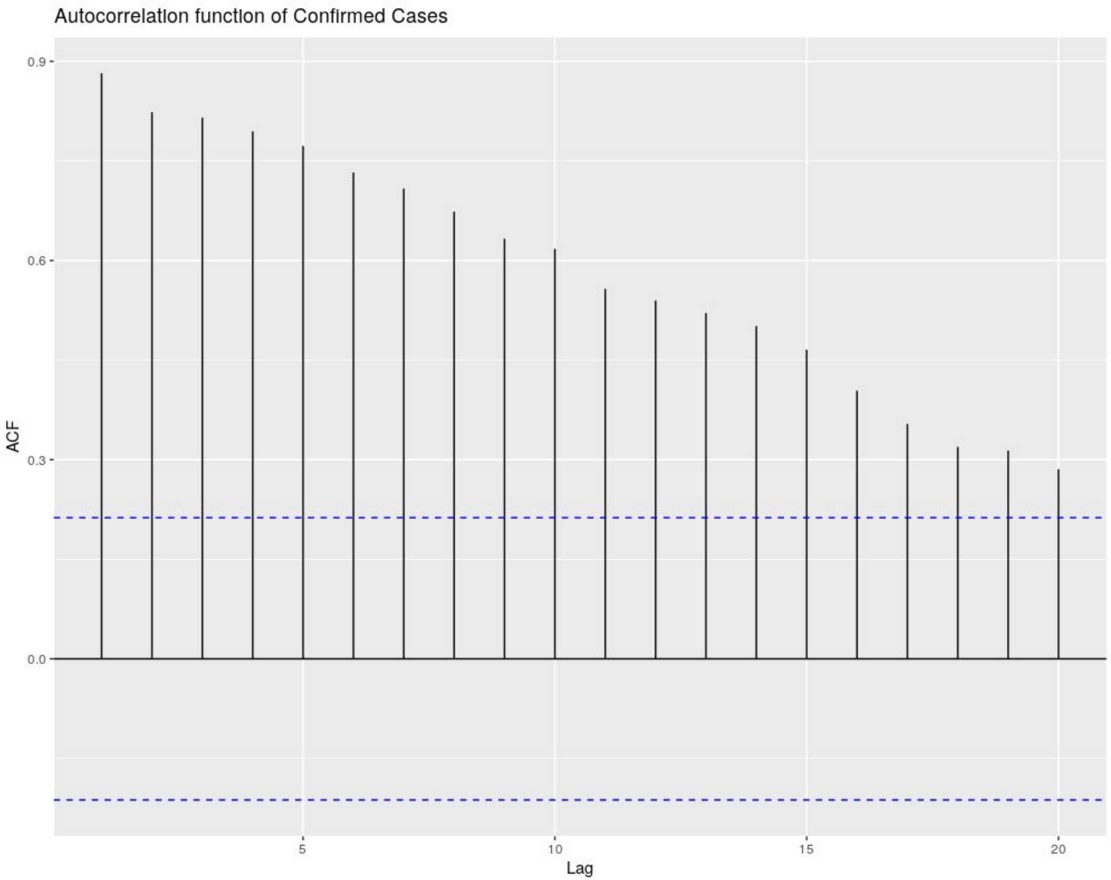


Stationarity

ACF Plot

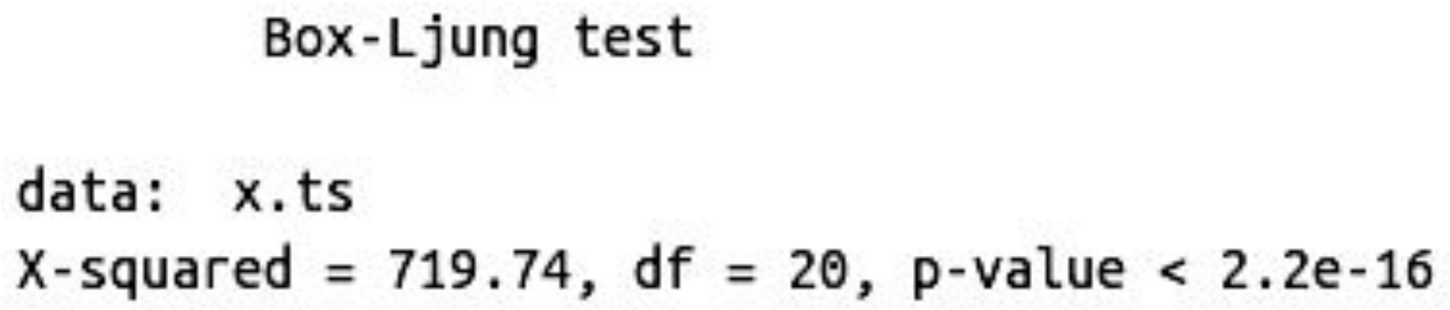
Stationary: ACF will drop to zero relatively quickly

Non-stationary: ACF decreases slowly (large and positive values)



Stationarity

Ljung-Box test



```
	Box-Ljung test

data:  x.ts
X-squared = 719.74, df = 20, p-value < 2.2e-16
```

Stationarity

Non-stationary to Stationary:

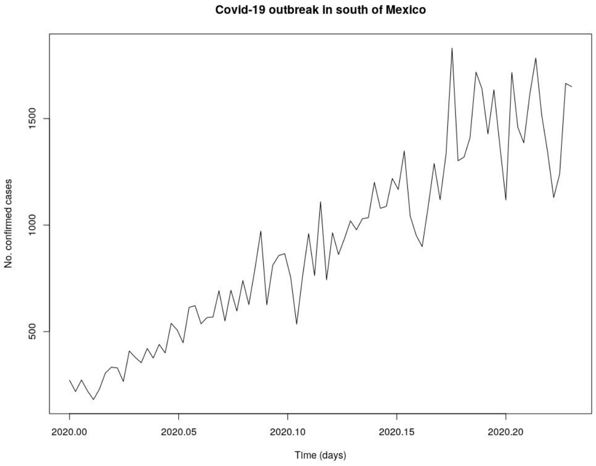
Differencing:

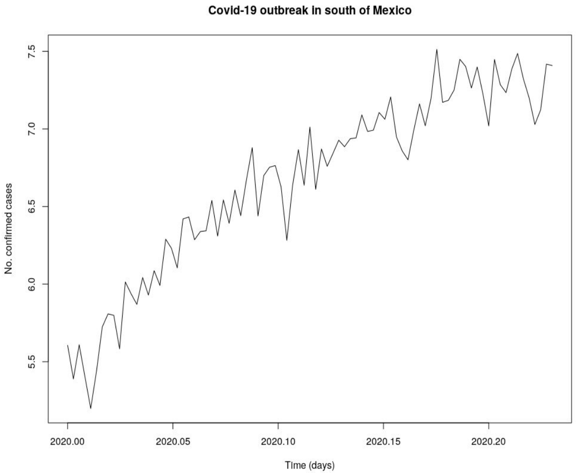
- Stabilise the mean
- Eliminating (or reducing) trend and seasonality.

Transformations (e.g. logarithms)

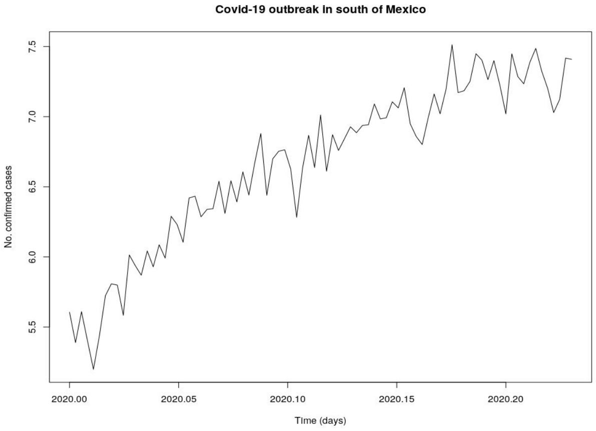
- Stabilise the variance.

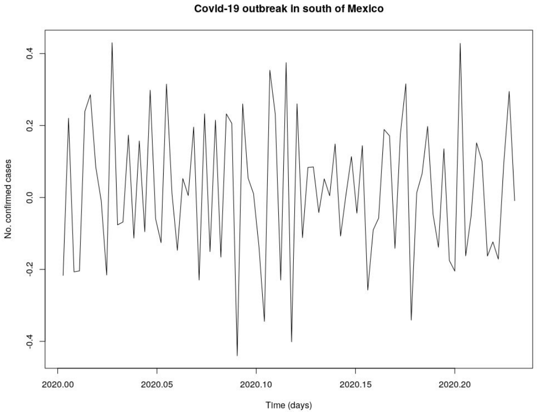
Log Transformation





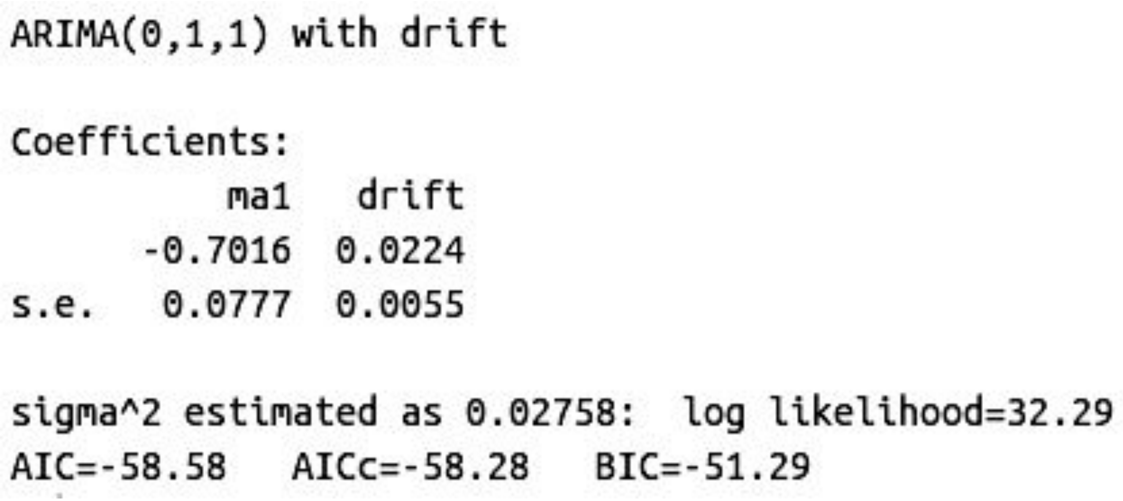
1-Lag Differencing





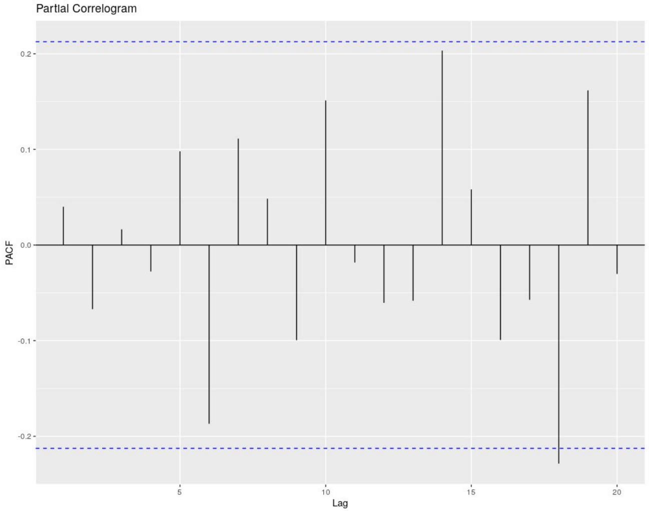
ARIMA

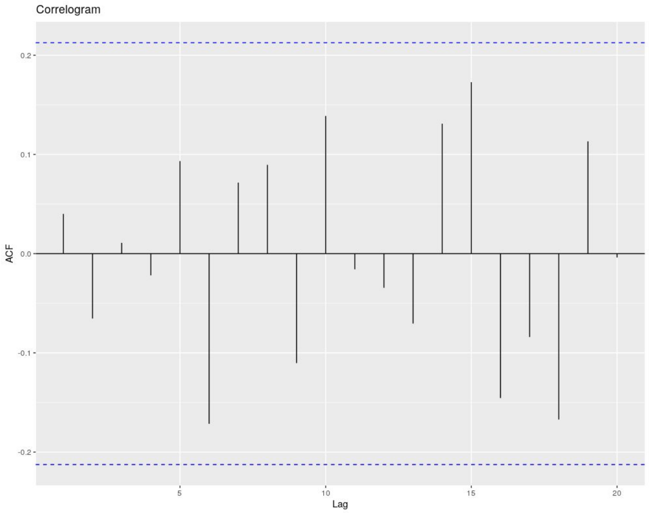
auto.arima()

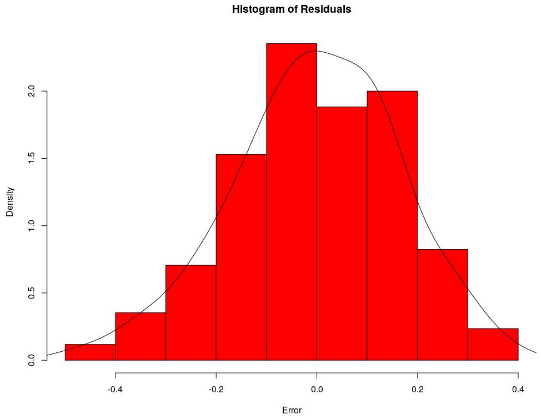


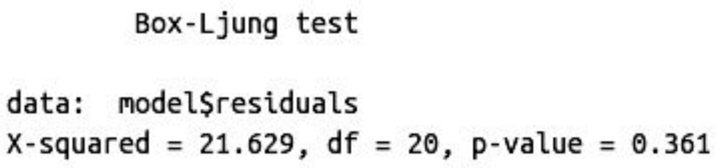
```
ARIMA(0,1,1) with drift

Coefficients:
          ma1   drift
      -0.7016  0.0224
s.e.   0.0777  0.0055

sigma^2 estimated as 0.02758:  log likelihood=32.29
AIC=-58.58   AICc=-58.28   BIC=-51.29
```

Residual Analysis





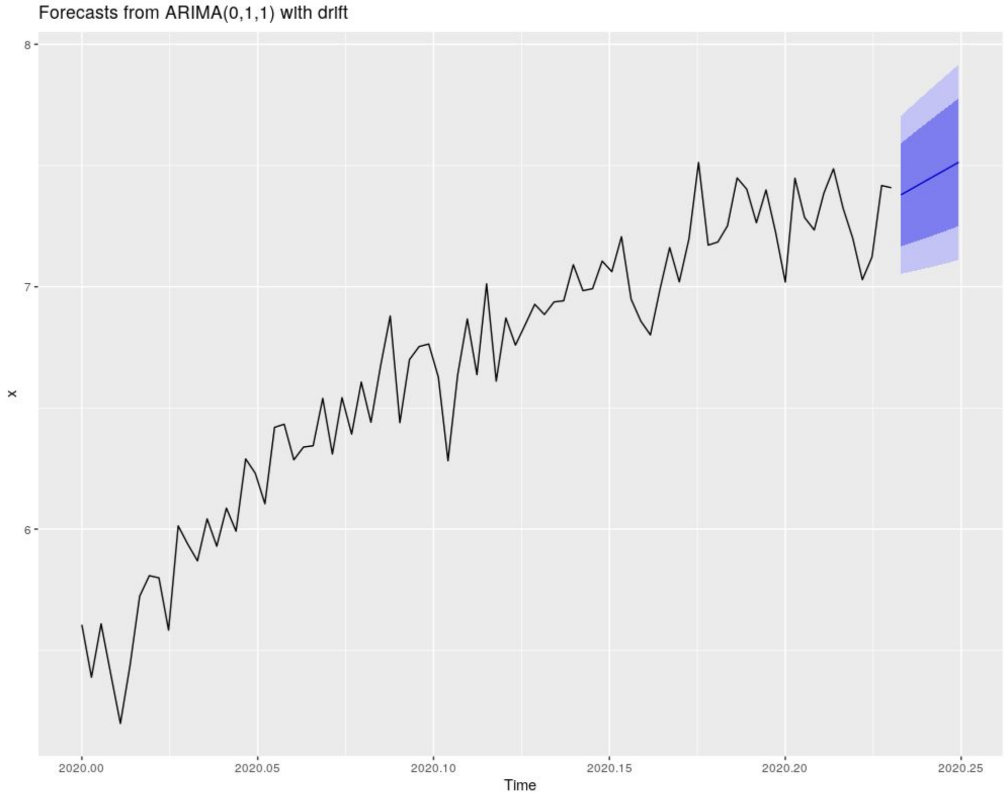




```
        Box-Ljung test

data:  model$residuals
X-squared = 21.629, df = 20, p-value = 0.361
```

Forecast

```
pred <- forecast(model, level = c(95, 80), h = 7)
```



Poisson Distribution

$$
\begin{aligned}
Y_i \mid \lambda_i &\sim P(\lambda_i) \\
p(y_i) &= \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!} \qquad y_i = 0, 1, \ldots, \qquad \lambda_i > 0
\end{aligned}
$$

Generalized Linear Model

- Canonical Link function for Poisson data is the log link

$$
\begin{aligned}
\log(\lambda_i) = \eta_i &= \beta_0 + X_1\beta_1 + \ldots X_p\beta_p \\
\lambda &= \exp(\beta_0 + X_1\beta_1 + \ldots X_p\beta_p)
\end{aligned}
$$

Quasi-Poisson Regression Model

So, what we did is train a glm poisson model with the R software using the dataset with data from 1 May to 10 July, and we performed then the prediction on the data from July 10 to July 24.

The available variables in the dataset created by us for this model were 6:

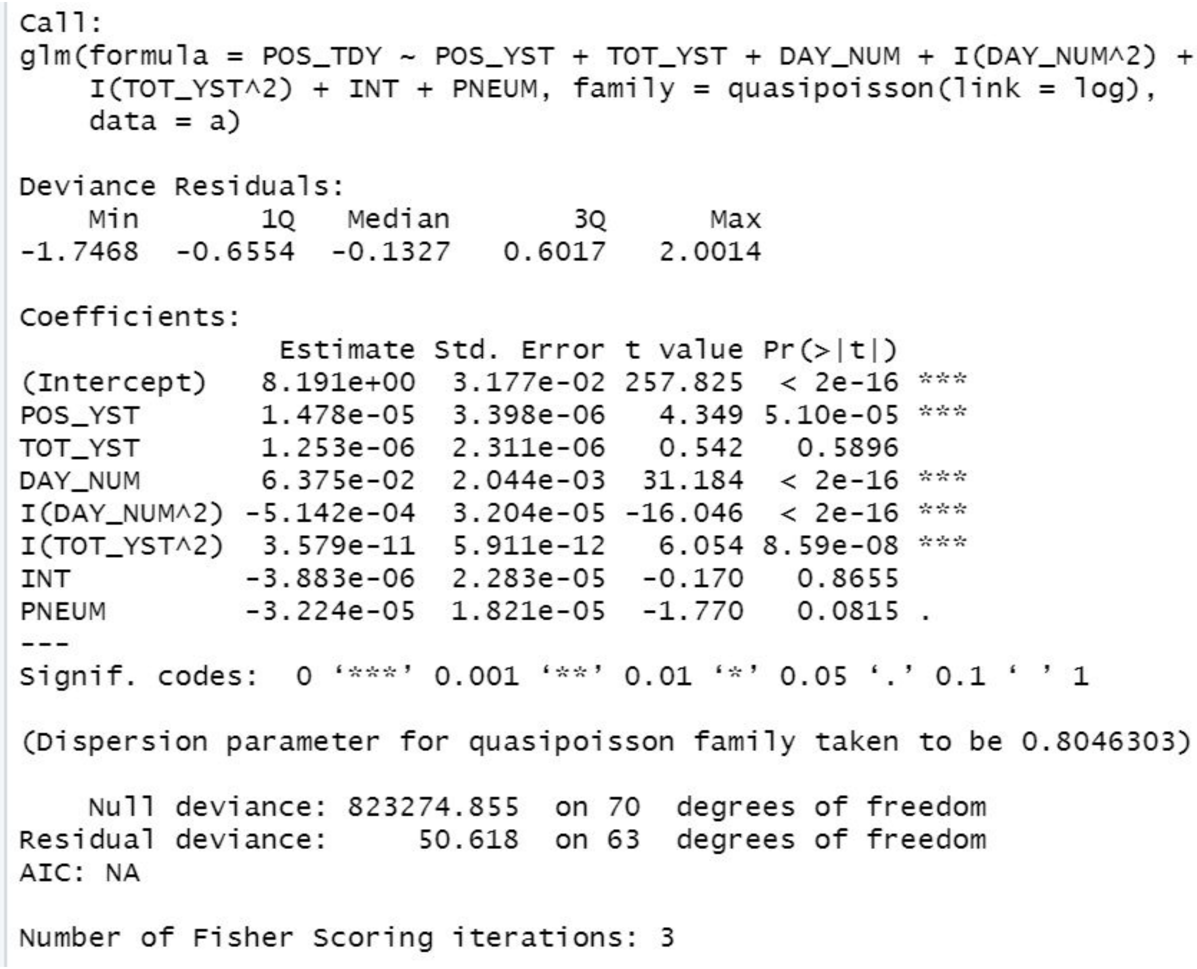
Day number(from 1 to n), total cases today, total cases yesterday, number of tests made yesterday, number of intubated patients, and number of patients with pneumonia.

Quasi-Poisson Regression Model

Obviously, the dependent variable has been setted as number of positive cases today, being what we are interested in predict.

About the independent variables, as first we try to include in the model all the available variables, plus some of them squared.

First model summary:

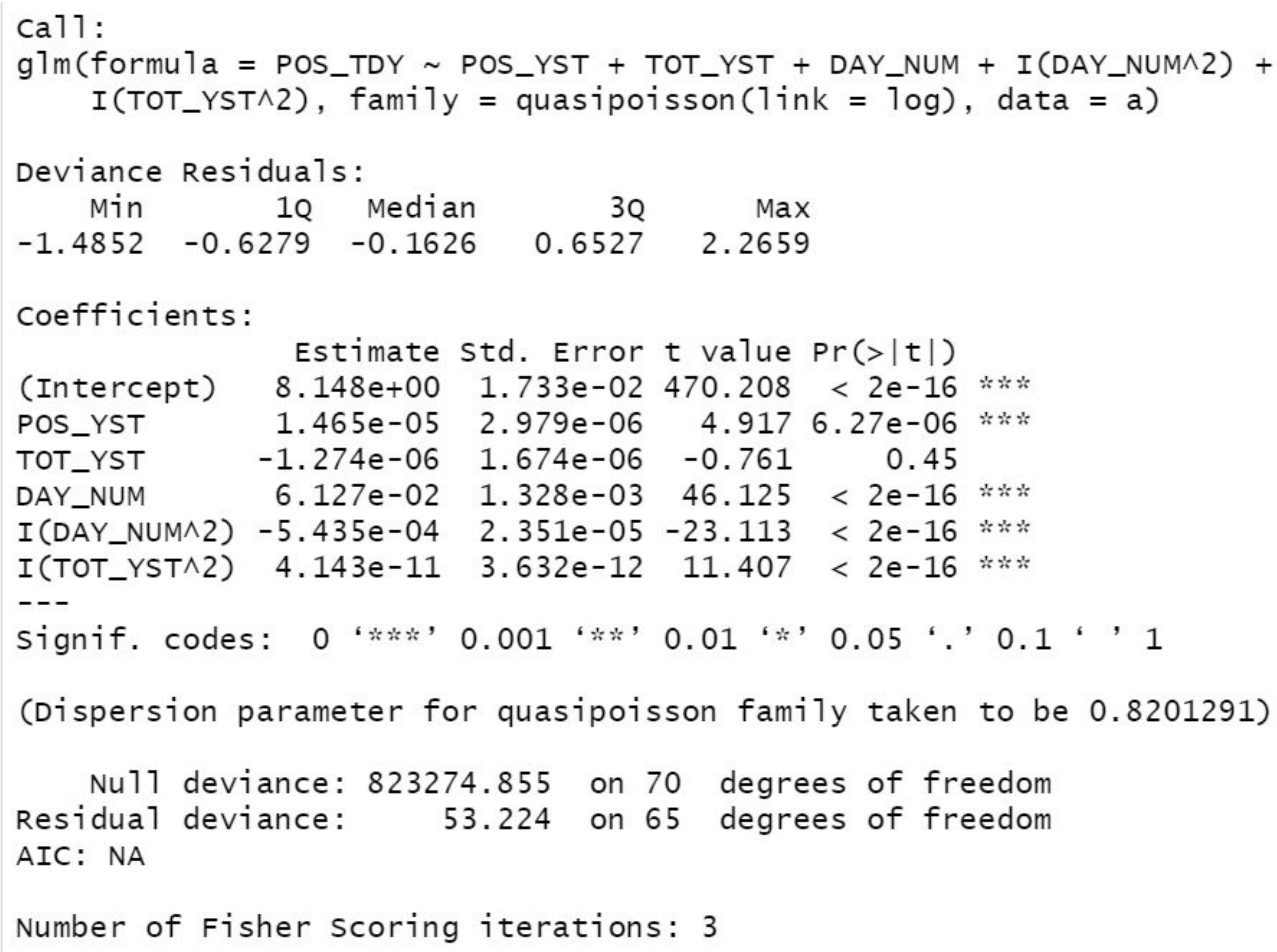


```
Call:
glm(formula = POS_TDY ~ POS_YST + TOT_YST + DAY_NUM + I(DAY_NUM^2) +
    I(TOT_YST^2) + INT + PNEUM, family = quasipoisson(link = log),
    data = a)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7468  -0.6554  -0.1327   0.6017   2.0014

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.191e+00  3.177e-02 257.825  < 2e-16 ***
POS_YST       1.478e-05  3.398e-06   4.349 5.10e-05 ***
TOT_YST       1.253e-06  2.311e-06   0.542   0.5896
DAY_NUM       6.375e-02  2.044e-03  31.184  < 2e-16 ***
I(DAY_NUM^2) -5.142e-04  3.204e-05 -16.046  < 2e-16 ***
I(TOT_YST^2)  3.579e-11  5.911e-12   6.054 8.59e-08 ***
INT          -3.883e-06  2.283e-05  -0.170   0.8655
PNEUM        -3.224e-05  1.821e-05  -1.770   0.0815 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 0.8046303)

    Null deviance: 823274.855  on 70  degrees of freedom
Residual deviance:     50.618  on 63  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 3
```

Model Fitting

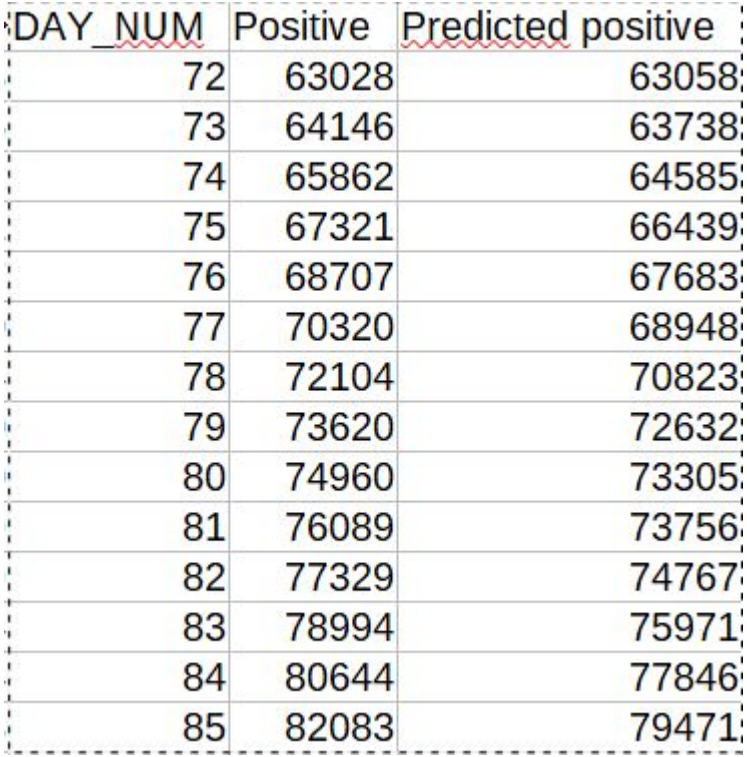
Looking at the model summary, we see that TOT_YST, the No. of patients with pneumonia and patients intubated have a small significance. We performed then a backward selection, arriving at this model:

```
glm(formula = POS_TDY ~  POS_YST+ TOT_YST+DAY_NUM+I(DAY_NUM^2)+I(TOT_YST^2),
data = a, family =quasipoisson(link = log))
```



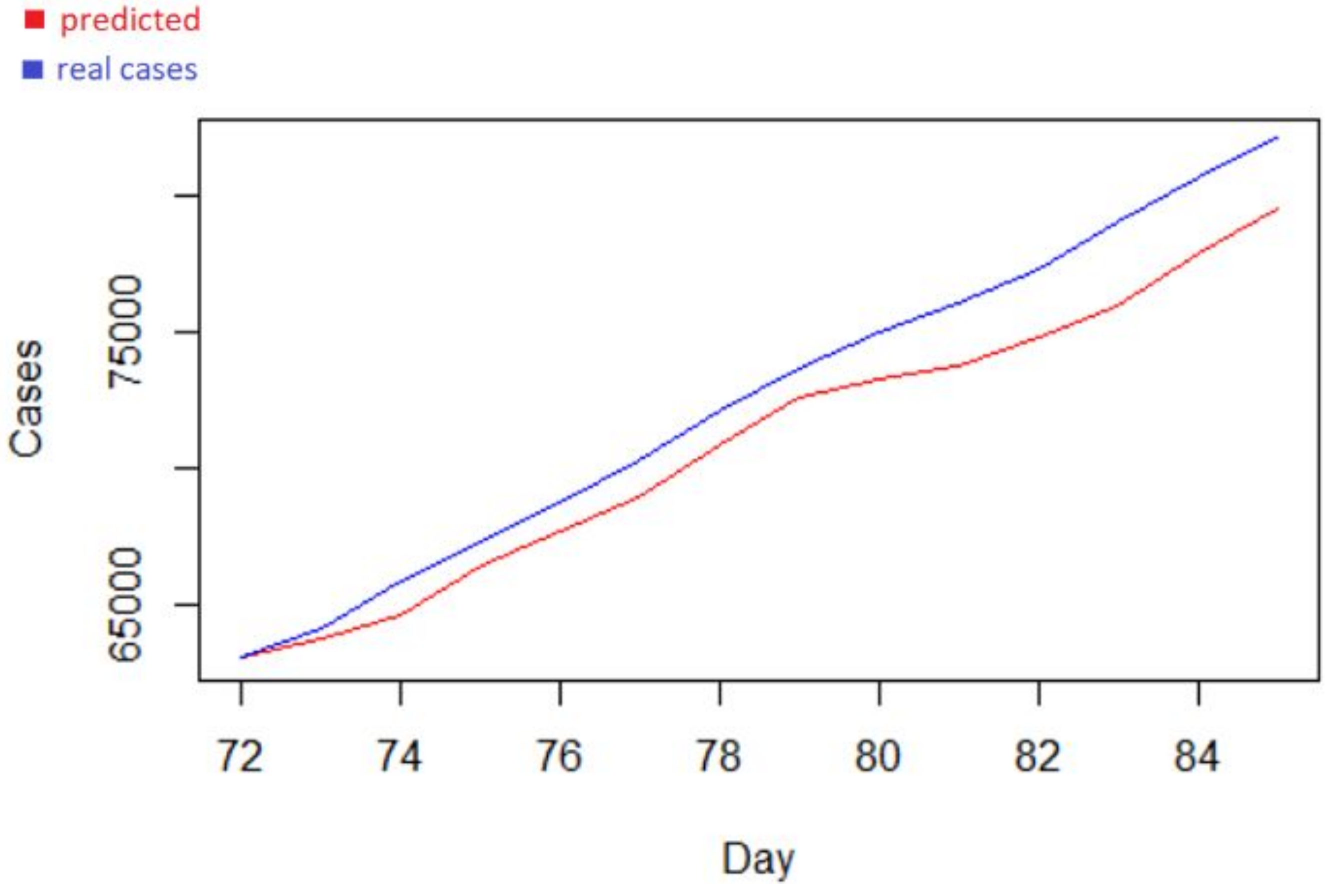
```
Call:
glm(formula = POS_TDY ~ POS_YST + TOT_YST + DAY_NUM + I(DAY_NUM^2) + 
    I(TOT_YST^2), family = quasipoisson(link = log), data = a)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4852  -0.6279  -0.1626   0.6527   2.2659  

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   8.148e+00  1.733e-02 470.208  < 2e-16 ***
POS_YST       1.465e-05  2.979e-06   4.917 6.27e-06 ***
TOT_YST      -1.274e-06  1.674e-06  -0.761     0.45    
DAY_NUM       6.127e-02  1.328e-03  46.125  < 2e-16 ***
I(DAY_NUM^2) -5.435e-04  2.351e-05 -23.113  < 2e-16 ***
I(TOT_YST^2)  4.143e-11  3.632e-12  11.407  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 0.8201291)

    Null deviance: 823274.855  on 70  degrees of freedom
Residual deviance:     53.224  on 65  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 3
```

Prediction Results



DAY_NUM	Positive	Predicted positive
72	63028	63058
73	64146	63738
74	65862	64585
75	67321	66439
76	68707	67683
77	70320	68948
78	72104	70823
79	73620	72632
80	74960	73305
81	76089	73756
82	77329	74767
83	78994	75971
84	80644	77846
85	82083	79471

Prediction Results

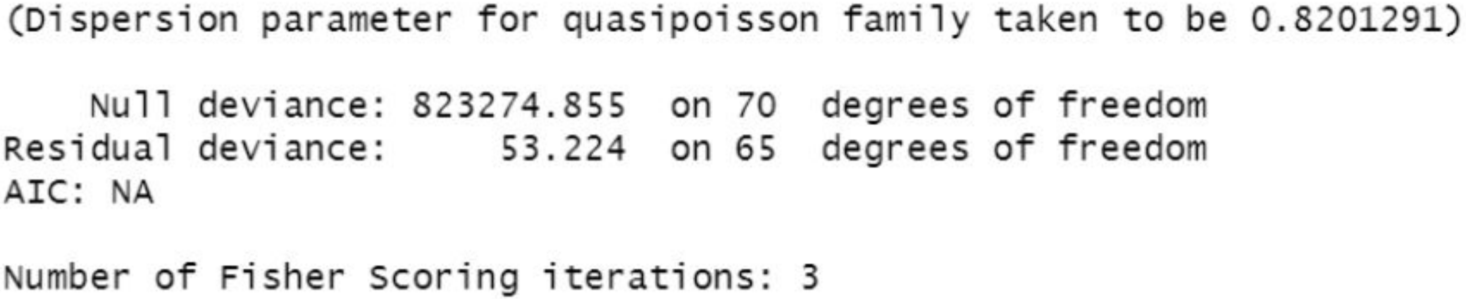


Model goodness of fit

The main way to evaluate a poisson regression is looking at the overdispersion of the model. A way to assess if we have overdispersion is looking at the ratio between the residual deviance and the degrees of freedom, that if is well over 1 show overdispersion; in our case we can see by the model summary:

Residual deviance: 50.618 on 63 degrees of freedom

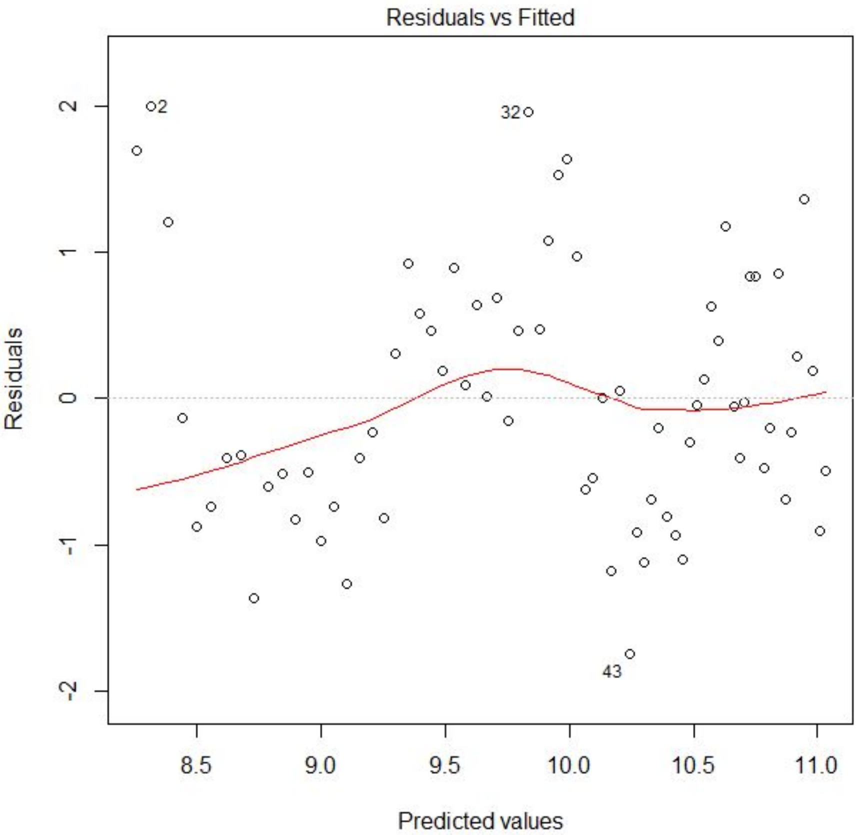
The ratio in this case is even smaller than 1.

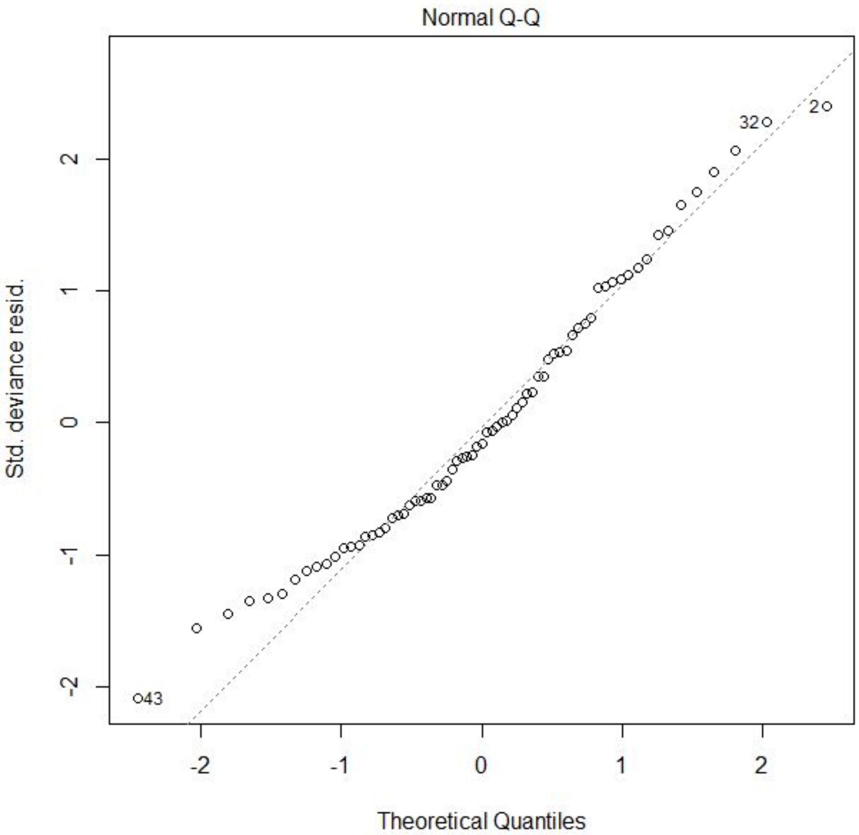


```
(Dispersion parameter for quasipoisson family taken to be 0.8201291)

    Null deviance: 823274.855  on 70  degrees of freedom
Residual deviance:     53.224  on 65  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 3
```

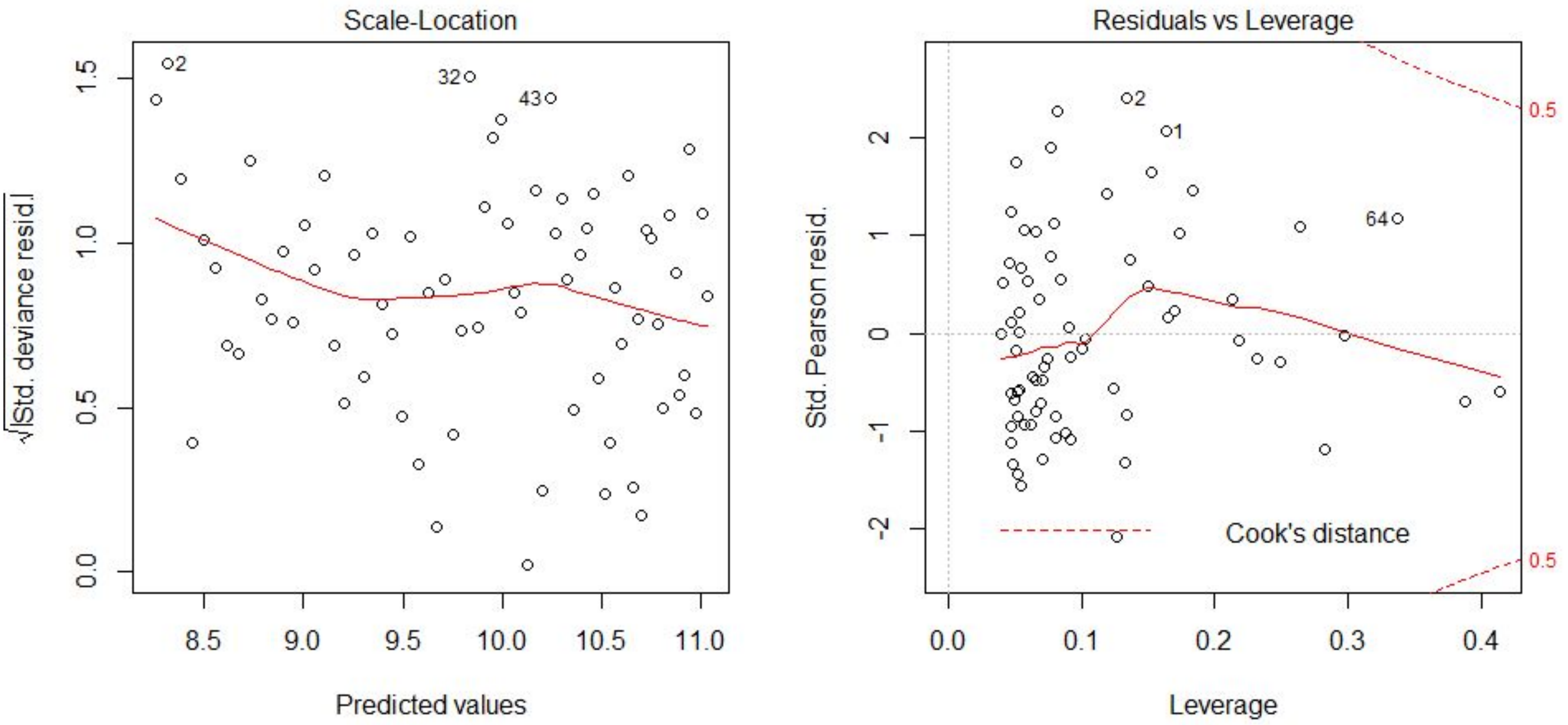
So our model has no overdispersion.

Model checking

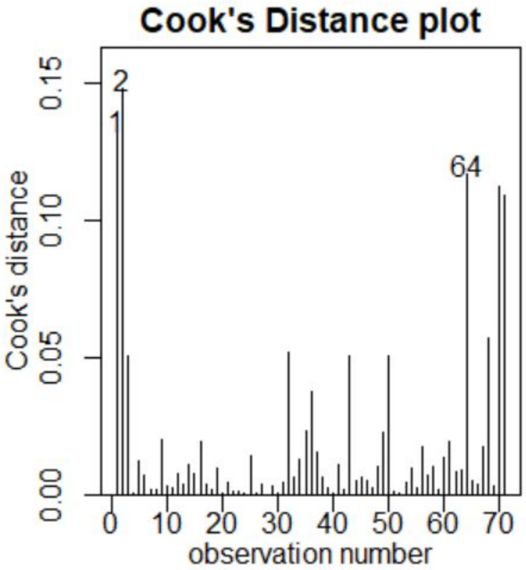




Model checking

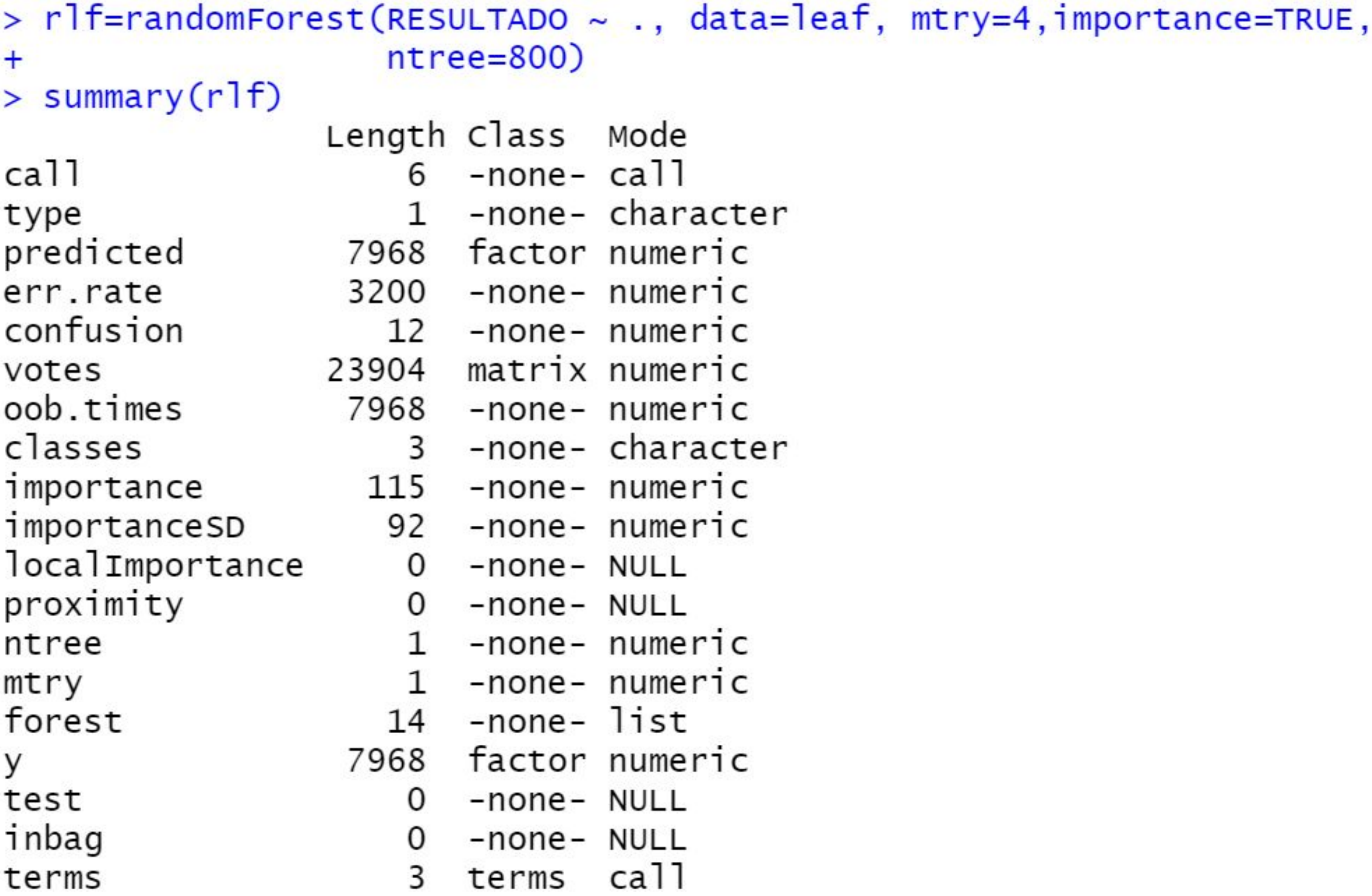


Also if for poisson family models the overdispersion is the most important parameter for the model checking, it worths to take a look at these plots. In the first, residual vs fitter, we don’t observe any strong pattern, that is what we want to.
The QQ plot, doesn’t fit perfectly but besides some observations on the bottom left, the normality of the residual is ok.
In the third, with the standardazied residuals on the y axis, we confirm that the variance doesn’t show a great heterogeneity
Finally, the last plot shows just one point slightly over the cook’s distance lines.
Using a R function we see better wich points have high cook’s distance:



The observation 1,2, 64 are the three with higher distance. Not very high though, so we left them.

An attempt with a classification algorithm: Random forest



```
> rlf=randomForest(RESULTADO ~ ., data=leaf, mtry=4,importance=TRUE,
+                  ntree=800)
> summary(rlf)
                Length Class  Mode     
call                6  -none- call     
type                1  -none- character
predicted        7968  factor numeric  
err.rate         3200  -none- numeric  
confusion          12  -none- numeric  
votes           23904  matrix numeric  
oob.times        7968  -none- numeric  
classes             3  -none- character
importance        115  -none- numeric  
importanceSD       92  -none- numeric  
localImportance     0  -none- NULL     
proximity           0  -none- NULL     
ntree               1  -none- numeric  
mtry                1  -none- numeric  
forest             14  -none- list     
y                7968  factor numeric  
test                0  -none- NULL     
inbag               0  -none- NULL     
terms               3  terms  call     
```

Having the datasets downloaded from the Mexico government more than 20 numerical or binary variables, we wanted to see if a good classification algorithm could help in predict the outcome of a covid test using these variables.
RF is a classification algorithm based on decision trees. We tried to train the rf on a subset of the dataset of june 1, picking all the numerical or binary variables in the dataset on about 100.000 observations, and choosing as response variable “RESULTADO”, that is the outcome of the covid test.
We then used as test sets more than 2*10e6 observations, but we observed that the accuracy was about 55%, basically almost randomness, even after having tried to tune some parameters of the RF(like the number of trees or m) to obtain a better result.
Rf is a good classification algorithm, given the results, we concluded that with these kind of variables a classification to predict the result of the test is not a good approach.

```
> summary(i)
   Mode   FALSE    TRUE 
logical  358279  446875 
```

```
> summary(i)
   Mode   FALSE    TRUE 
logical  280559  364716 
```

```
> summary(i)
 Mode   FALSE    TRUE 
logical  249816  329380 
```

```
> summary(i)
   Mode   FALSE    TRUE 
logical  444176  417676 
```

THE END