Sample No-1

How is Mortality affected by CO_2, Methane, Nitrous Oxide Emissions and Economic Factors

Data Description

The data are collected from https://data.worldbank.org/indicator/SH.DYN.NCOM.ZS . In this data,

(i) Response Variable:

Mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 (%) and it's converted into No_of_deaths (mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70)

(ii) Independent Variable:

pop_total_2018: Population of ages from 30 to 70

CO_2(2018): C02 emission per capita

Methane (2018): Methane emission (kt of Co2 equivalent)

nitrous_oxide (2018): Nitrous oxide emissions (thousand metric tons of CO2 equivalent)

exp(2018): Current Health Expenditure(% of GDP)

renewable energy (2018): Renewable energy consumption (% of total final energy consumption)

tobacco(2018): Prevalence of current tobacco use (% of adults)

All of these variables are taken for the year 2018 and have been compiled in one single dataset. Linkof final dataset is:
https://drive.google.com/drive/u/0/folders/11TE7KFyUzwKBx8XHGvBgTDOezMwQa5EG

Objective

Here, we have analyzed data of 141 countries mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 of year 2018.

- To find out contributable predictors towards mortality from CVD, cancer, diabetes or CRD.

Data Exploration

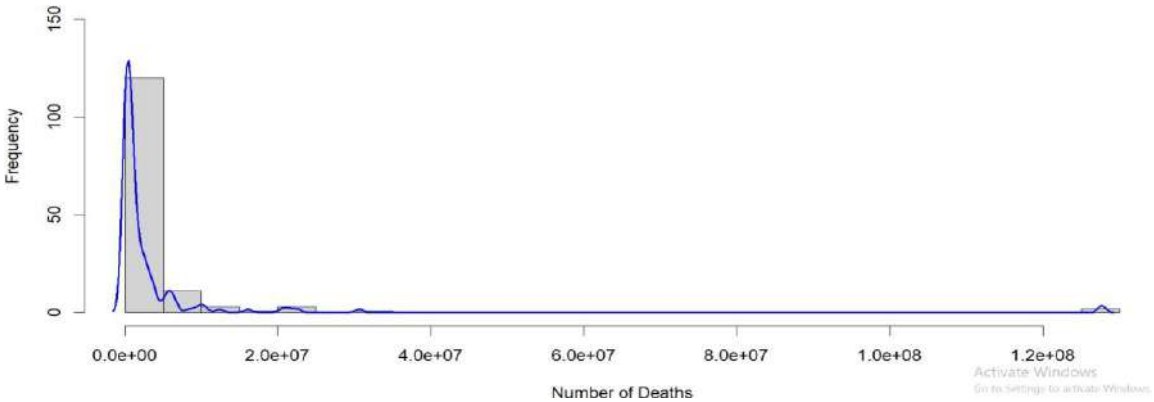
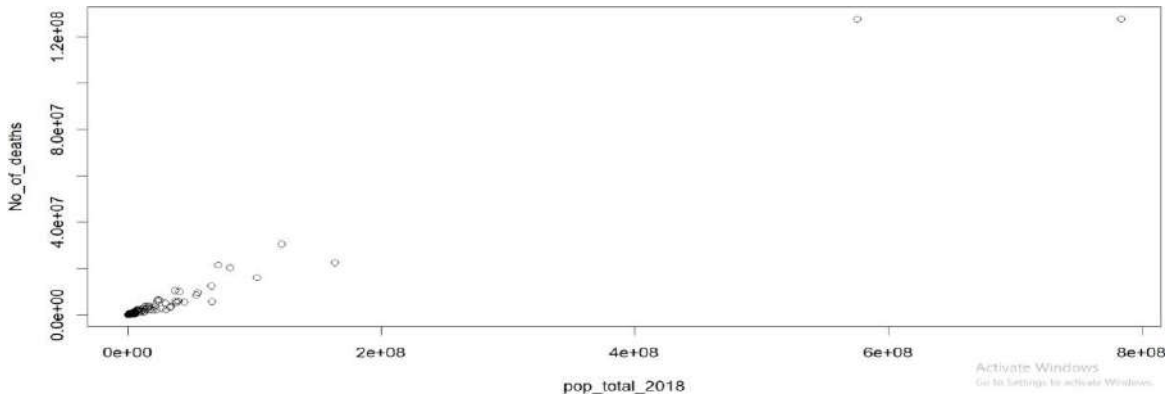


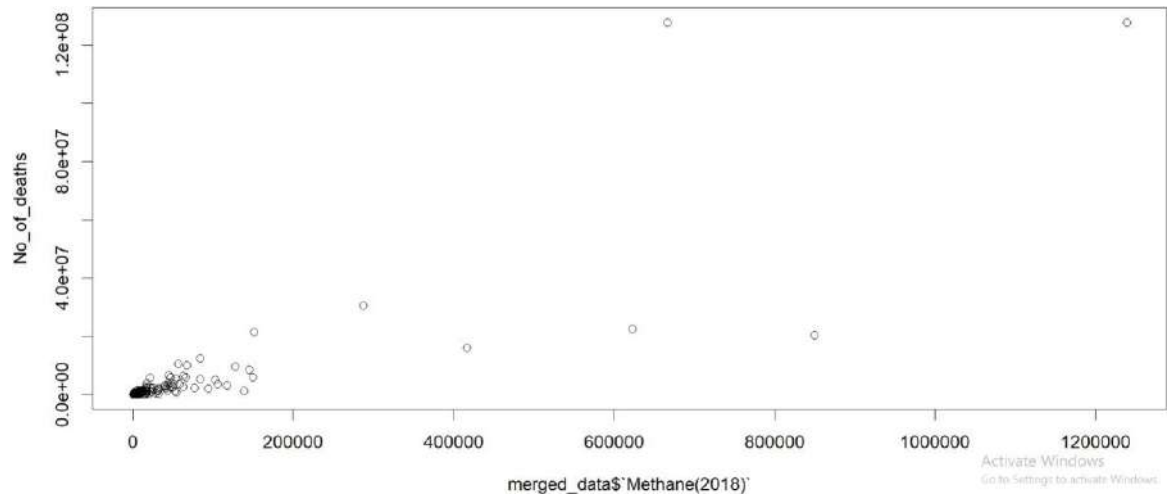
Fig 1.1: Frequency distribution for Number of deaths

From fig 1.1, we see that the distribution of Mortality is not Normally distributed rather it has a very high spike at initial values and a very long tail i.e. positively skewed. So, we can say there is an inflation from 0 values to 5e+06.

A natural second step in the exploratory analysis is to look at pairwise bivariate displays of the dependent variable against each of the regressors bringing out the partial relationships. In R, such bivariate displays can easily be generated with the plot () method for formulas, e.g., via plot (y ~ x).



It is obvious that, mortality increases with number of population. So, this should be in mind when fitting model and also it's obvious. For most of countries , population is from 0 to 2e+08.



Here, also mortality increases with methane emission(kt). We see that, for most of the countries methane emission is from 0 to 2,00,000 kt and very few is more than that.

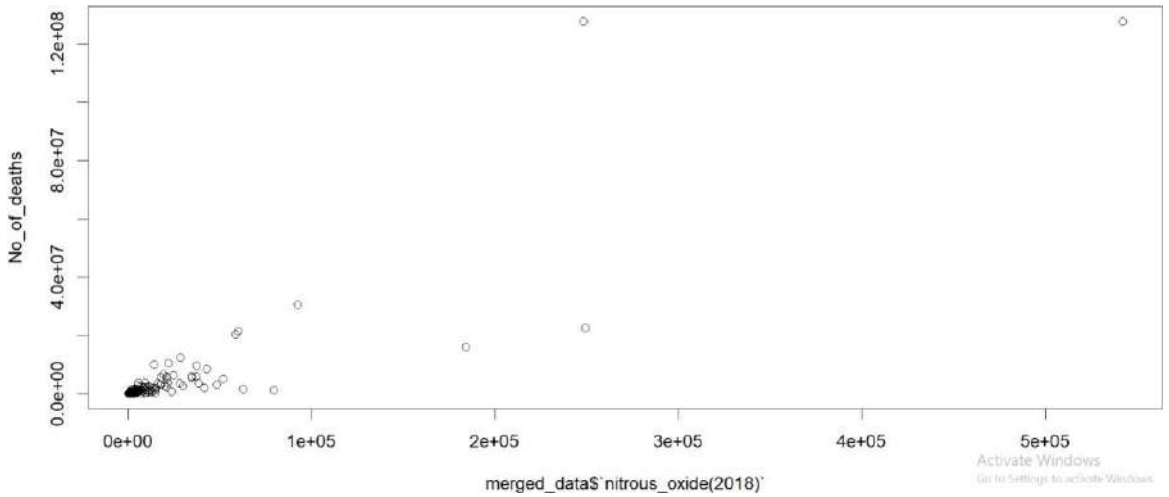
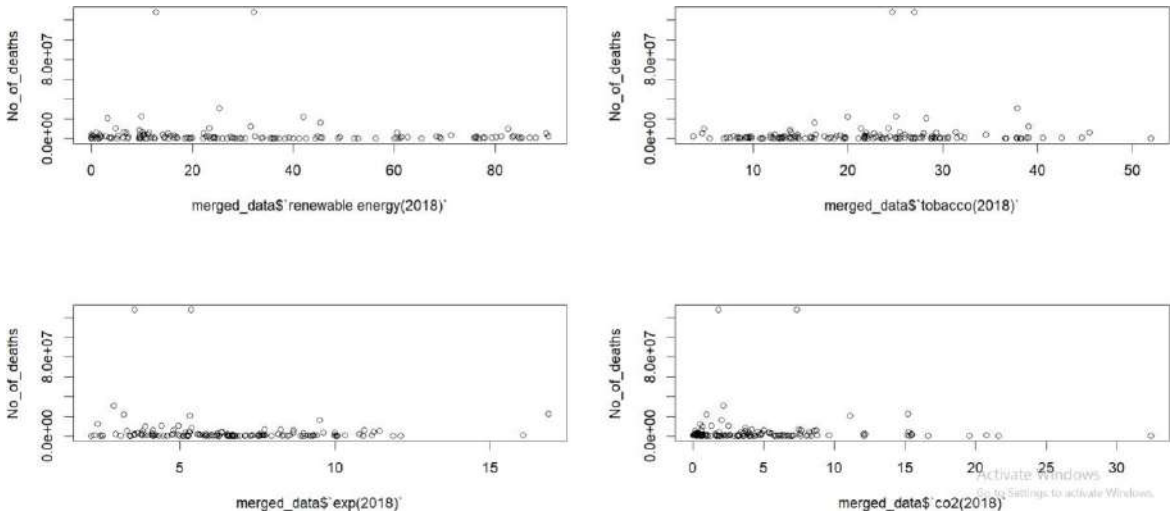


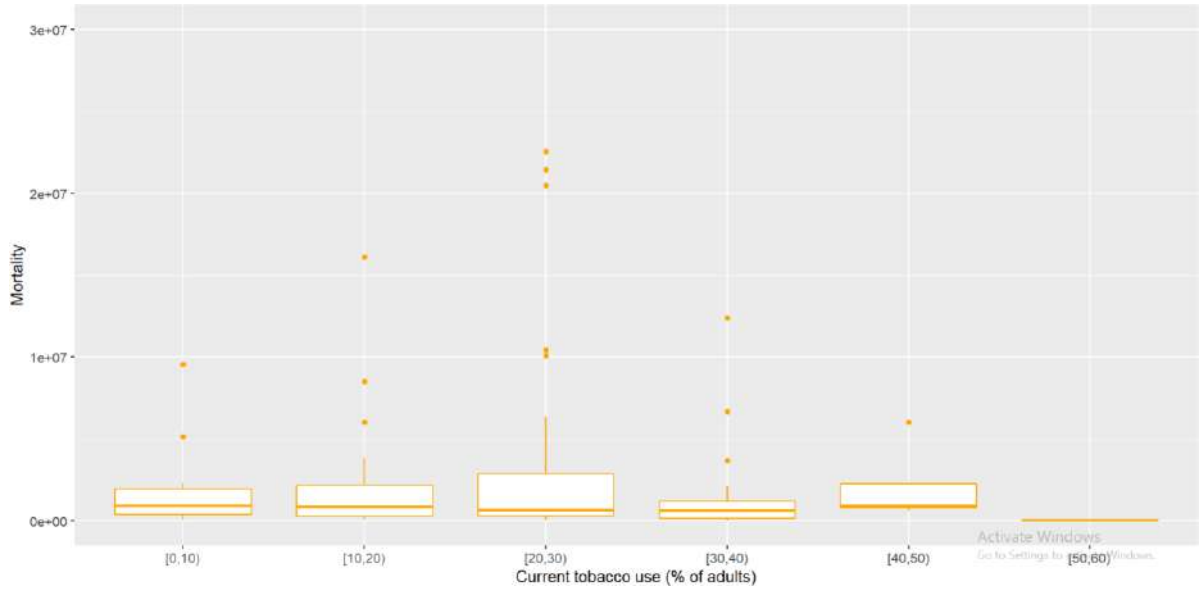
Fig 1.2: Scatterplots of Mortality vs regressors

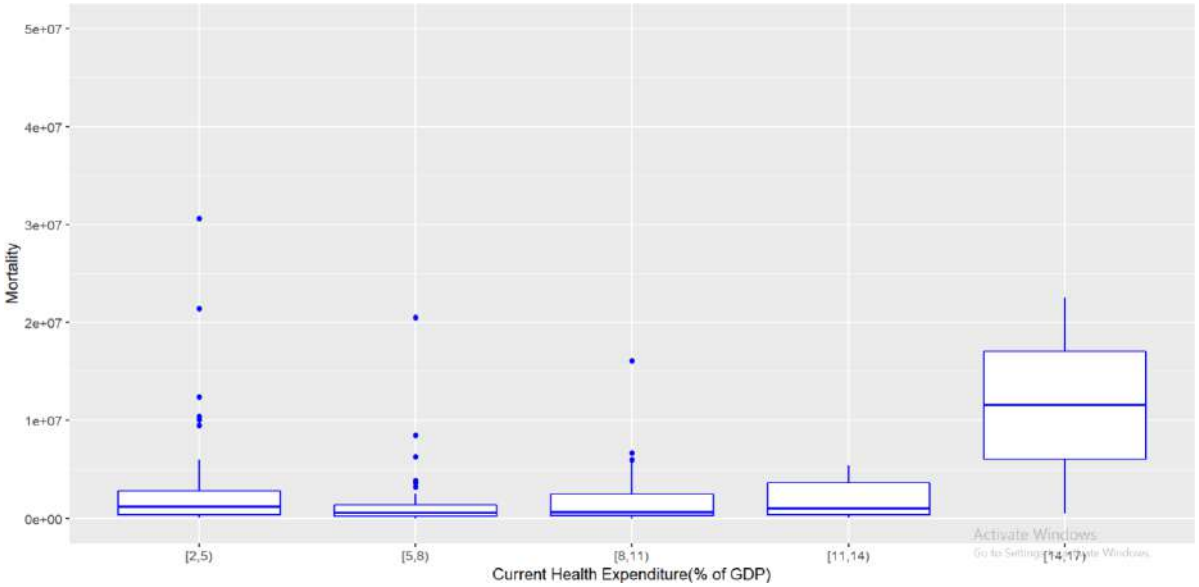
Also, same picture depicting for mortality vs Nitrous Oxide. For most of the countries is from 0 to1e+05.



Now, for Renewable energy, tobacco, Current Health expenditure and CO_2 we plot all these with response variable but can't get any clear idea from that.

This is clearly not useful as both variables are count variables producing numerous ties in the bivariate distribution and thus obscuring a large number of points in the display. To overcome the problem, it is useful to group the number of Current Health Expenditure (% of total GDP) into a factor with levels [2,5), [5,8) and more produce a boxplot instead of a scatterplot.


3e+07
2e+07
1e+07
0e+00
Mortality
[0,10)
[10,20)
[20,30)
[30,40)
[40,50)
[50,60)
Current tobacco use (% of adults)


5e+07
4e+07
3e+07
2e+07
1e+07
0e+00
Mortality
[2,5)
[5,8)
[8,11)
[11,14)
[14,17)
Current Health Expenditure(% of GDP)

P.T.O.

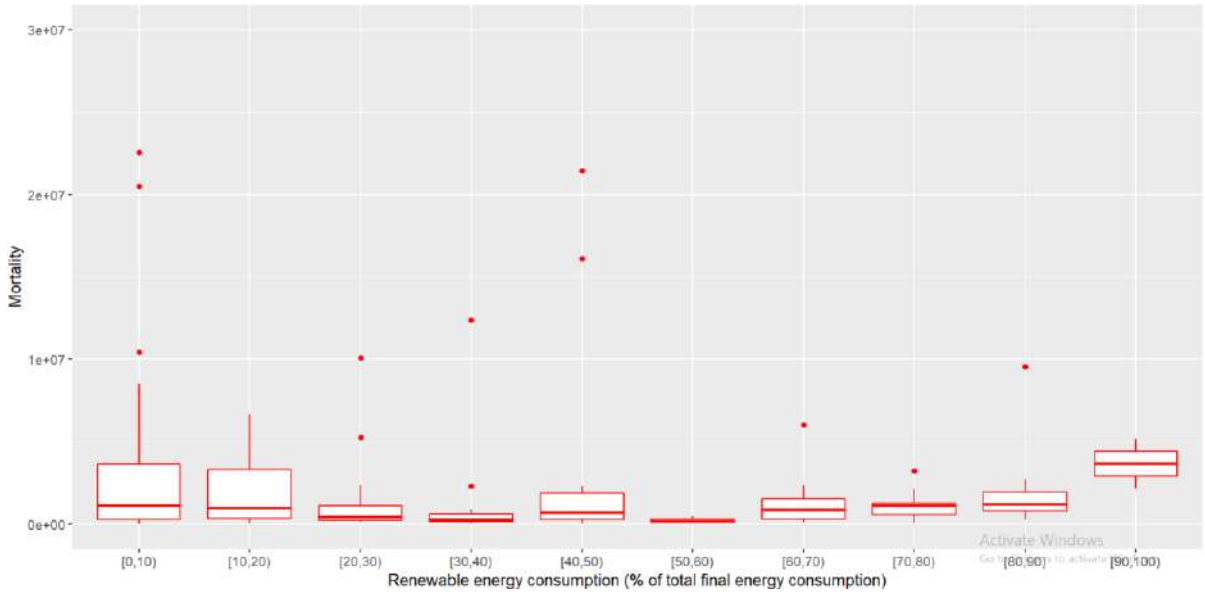


Fig 1.3: Plot of Mortality vs all regressors

Here, we see that when Prevelance of current tobacco use is from 20% and more, variation inmortality increases and also huge outliers also present.

A very surprising fact that when Current Health Expenditure is up to 8%, variation of mortality is low but outliers are present. After 8%, variation increases also outliers present. When health exp is 14%to 17% variation is most; which is not expected. There is something hidden thing which is depicting this.

Also, Renewable energy consumption is higher than 50%, variation in mortality decreases and quite small and less outliers present; which is natural.

Modelling Strategies

We are going to predict Mortality using Population, Renewable energy (% of Total energy) , Methane emissions (kt of CO2 equivalent) , Nitrous oxide emissions (thousand metric tons of CO2 equivalent) ,Current Health Expenditure(% of GDP) , Prevalence of current tobacco use (%).

Here, Mortality is the number of deaths from CVD, Cancer, Diabetes or CRD.So, here are some following choices for fitting the data.

1. Poisson Regression
2. Quassi-Poisson Regression; which is adjusted to accounted for overdispersion.
3. Negeative Binomial Regression
4. Quassi-Negeative Regression; adjusted to accounted for overdispersion.

Here, ZIP models are not appropriate as there is absence of zero values and also no inflation at zero.

1. ***Poisson Regression***

Here, Y_I = Mortality per country I; I =1(1)n, n=# of Countries

We already see that Y_I doesn't follow Normal distribution, rather we can assume that log (Y_I)follows poisson distribution.

Thus, $Y_I \sim Poi(\lambda_i)$ and assume that Y_I's are independently distributed.

As, w.k.t mortality increases with increase in number of population. Thus, it's appropriate tomodel log (λ_i / n_i)

Model

$$\log\left(\frac{\lambda_i}{n_i}\right) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

$$\log(\lambda_i) = \log(n_i) + \beta_0 + \cdots + \beta_p x_{pi}$$

Here, p=5; Offset = log(Population) ;linear predictor with known coefficient.

```
#Poisson Regression

model_pois = glm(No_of_deaths~.-pop_total_2018+offset(log(pop_total_2018)),family =poisson(link="log"),data = merged_data)

summary(model_pois)
```

Output

We obtain the coefficient estimates along with associated partial Wald tests as follows:

```
> summary(model_pois)

Call:
glm(formula = No_of_deaths ~ . - pop_total_2018 + offset(log(pop_total_2018)),
    family = poisson(link = "log"), data = merged_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1177.96   -231.32     35.09    196.14   1521.34

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -1.595e+00  2.504e-04 -6369.4   <2e-16 ***
`Methane(2018)`          6.754e-07  3.521e-10  1917.9   <2e-16 ***
`exp(2018)`             -3.354e-02  2.037e-05 -1646.2   <2e-16 ***
`nitrous_oxide(2018)`   -1.626e-06  7.564e-10 -2149.2   <2e-16 ***
`renewable energy(2018)` 1.457e-03  3.042e-06   479.2   <2e-16 ***
`tobacco(2018)`          5.231e-03  6.042e-06   865.8   <2e-16 ***
`co2(2018)`             -1.664e-02  1.990e-05  -836.1   <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: 48852376  on 140  degrees of freedom
Residual deviance: 25450828  on 134  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 4
```

Comment

All coefficients of covariates are highly significant with the Mortality variable leading to somewhat larger Wald statistics. Here also residual deviance is smaller than null deviance thatimplies covariates are somewhat explaining the model that’s why deviance decreasing.

However, the Wald test results might be too optimistic due to a misspecification of the likelihood. As the exploratory analysis suggested that over-dispersion is present in this data set,we re-compute the Wald tests using sandwich standard errors via

coeftest(model_pois,vcov. = sandwich)

```
> coeftest(model_pois,vcov. = sandwich)

z test of coefficients:

                           Estimate  Std. Error z value  Pr(>|z|)
(Intercept)             -1.5948e+00  1.9123e-01 -8.3393 < 2.2e-16 ***
`Methane(2018)`          6.7535e-07  1.3177e-07  5.1252 2.972e-07 ***
`exp(2018)`             -3.3541e-02  1.4993e-02 -2.2371   0.02528 *
`nitrous_oxide(2018)`   -1.6257e-06  2.1301e-07 -7.6319 2.314e-14 ***
`renewable energy(2018)` 1.4574e-03  1.8042e-03  0.8078   0.41921
`tobacco(2018)`          5.2312e-03  3.4472e-03  1.5175   0.12913
`co2(2018)`             -1.6635e-02  1.0267e-02 -1.6202   0.10519
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comment

So, Methane, nitrous oxide are highly significant and expenditure is significant at 0.01% level of significance which is unlike as above figure. Rest of 3 covariates aren’t significant for the model.

All of these estimates are done using covariance matrix “Sandwich” which is $Var(\hat{\beta}) =$

$\left(-E\left(\frac{\delta^2 l}{\delta\beta\delta\beta'}\right)\right)^{-1}$. Also, here standard errors of estimates are more stable than before. This will also be confirmed by the following models that deal with over-dispersion.

Goodness of fit of Poisson model

W.k.t, if model is good then (Null deviance – Residual deviance) is small and $(Null\ deviance - Residual\ deviance) \sim x_p^2$; p=# of parameters

H_0: model is adequate vs H_1: model is not adequate

```
chisq = qchisq(0.95,(model_pois$df.null-model_pois$df.residual))
T = model_pois$null.deviance - model_pois$deviance
ifelse(T>chisq,"Reject Ho","Can't reject Ho")
```

Output: "Reject Ho"

```
qchisq(0.95,model_pois$df.residual)
162.0156
deviance(model_pois)
25450828
sum((resid(model_pois,type = "pearson"))^2)
25957610
```

Comment

Here, both the Deviance and Pearson Chi-square residuals are at an extreme distance from the 95% critical point, both are in 25000000, where as the critical point is only 162.0156. Clearly, the model doesn't fit well at all.

Assuming, variance is proportional to mean rather than equal to mean, estimate of ϕ is,

```
phi = sum((resid(model_pois,type =
"pearson"))^2)/df.residual(model_pois)round(c(phi, sqrt(phi)),3)
193713.506  440.129
```

It can be seen that the variance extremely higher than the mean, and standard errors should also beadjusted by multiplication with 440.129. This can be done by a Quasi-Poisson model.

2. *Quassi-Poisson Regression*

As from exploratory part we see that, there is presence of overdispersion. So, here we want dispersion to be estimated from the data. So, we use dispersion parameter as

$$\hat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

Where, n and p =# of rows and columns of data.

Model

$$\log\left(\frac{\lambda_i}{\eta_i}\right) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

$$\log(\lambda_i) = \log(\eta_i) + \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

Here, p=5; Offset = log (Population); linear predictor with known coefficient.

```
#Quasi-Poisson Regression
model_qpois <- glm (No_of_deaths~.-pop_total_2018+offset(log(pop_total_2018)),data = merged_data,family=quasipoisson)
summary(model_qpois)
```

```
> model_qpois <- glm(No_of_deaths~.-pop_total_2018+offset(log(pop_total_2018)),data = merged_data,family=quasipoisson)
> summary(model_qpois)

Call:
glm(formula = No_of_deaths ~ . - pop_total_2018 + offset(log(pop_total_2018)),
    family = quasipoisson, data = merged_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1177.96   -231.32     35.09    196.14   1521.34

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -1.595e+00  1.102e-01 -14.472  < 2e-16 ***
`Methane(2018)`          6.754e-07  1.550e-07   4.358 2.60e-05 ***
`exp(2018)`             -3.354e-02  8.967e-03  -3.740 0.000271 ***
`nitrous_oxide(2018)`   -1.626e-06  3.329e-07  -4.883 2.93e-06 ***
`renewable energy(2018)` 1.457e-03  1.339e-03   1.089 0.278257
`tobacco(2018)`          5.231e-03  2.659e-03   1.967 0.051223 .
`co2(2018)`             -1.664e-02  8.757e-03  -1.900 0.059639 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 193713.7)

    Null deviance: 48852376  on 140  degrees of freedom
Residual deviance: 25450828  on 134  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

Comment

This leads to an estimate of ϕ = 1,93,713.7 >>>1 that implies confirming that overdispersion presentin the data. The resulting partial Wald tests of the coefficients are rather similar to the results obtained from the Poisson regression with sandwich standard errors, leading to the same conclusions. Also, Residual deviance is same as that of Poisson regression.

Goodness of fit of Quassi-Poisson model:

H_0: Model is adequate; H_1: Model is inadequate

```
chisq = qchisq(0.95, (model_qpois$df.null-model_qpois$df.residual))
T = model_qpois$null.deviance-model_qpois$deviance
ifelse(T>chisq,"Reject Ho","Can't reject Ho")
```

Output: "Reject H_0"

```
library(knitr)
deviance_ = model_qpois$deviance
dev_df = model_qpois$df.residual
pearson_chisq = sum(residuals(model_qpois, type = "pearson")^2)
a = data.frame(Estimate = c(deviance_, pearson_chisq))
a$df = rep(dev_df,2)
a$P.Value = pchisq(a$Estimate, df = a$df, lower.tail = FALSE)
row.names(a) = c("Deviance", "Pearson")
kable(a, caption = "Goodness of fit")
```

Table: Goodness of fit

	Estimate	df	P.Value
Deviance	25450828	134	0
Pearson	25957610	134	0

Comment

Here, both the p-value for Deviance and Pearson goodness-of-fit tests is highly significant and wereject the null hypothesis that the model is adequate.

3. *Negative binomial regression*

A more formal way to accommodate over-dispersion in a count data regression model is to use a negative binomial model

```
fm_nbin <- MASS::glm.nb(No_of_deaths ~ .-pop_total_2018+offset(log(pop_total_2018)), data =merged_data,link=log)
summary(fm_nbin)
```

```
> summary(fm_nbin)

Call:
MASS::glm.nb(formula = No_of_deaths ~ . - pop_total_2018 + offset(log(pop_total_2018)),
    data = merged_data, link = log, init.theta = 8.95413603)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1785  -0.8439  -0.1303   0.5260   2.9884

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)             -1.478e+00  1.244e-01 -11.875  < 2e-16 ***
`Methane(2018)`          6.855e-07  4.542e-07   1.509 0.131195
`exp(2018)`             -5.050e-02  1.084e-02  -4.659 3.18e-06 ***
`nitrous_oxide(2018)`   -1.780e-06  1.203e-06  -1.480 0.139003
`renewable energy(2018)` 2.001e-03  1.358e-03   1.473 0.140770
`tobacco(2018)`          9.619e-03  3.105e-03   3.098 0.001947 **
`co2(2018)`             -2.595e-02  6.812e-03  -3.809 0.000139 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(8.9541) family taken to be 1)

    Null deviance: 206.66  on 140  degrees of freedom
Residual deviance: 143.63  on 134  degrees of freedom
AIC: 3930.8

Number of Fisher Scoring iterations: 1


              Theta:  8.95
          Std. Err.:  1.05

 2 x log-likelihood:  -3914.785
```

Comment

Here, CO_2, Health expenditure, intercept are highly significant and tobacco is significant at 0.001% level of significance for the model. Rest of the covariates are not significant for the model unlikely the case in quassi-poisson and poisson moel. In this model, Residual deviance is 143.46 which is very very less than poisson/quassi-poisson model. Estimated Values of AIC, Theta are 3930.8 and 8.95 respectively.

coeftest(fm_nbin,vcov. = sandwich)

```
> coeftest(fm_nbin,vcov. = sandwich)

z test of coefficients:

                            Estimate  Std. Error z value  Pr(>|z|)
(Intercept)               -1.4776e+00  1.6666e-01 -8.8660 < 2.2e-16 ***
`Methane(2018)`            6.8551e-07  1.7002e-07  4.0319 5.532e-05 ***
`exp(2018)`               -5.0496e-02  1.4295e-02 -3.5323 0.0004119 ***
`nitrous_oxide(2018)`     -1.7797e-06  4.8691e-07 -3.6550 0.0002572 ***
`renewable energy(2018)`   2.0006e-03  1.2188e-03  1.6414 0.1007139
`tobacco(2018)`            9.6189e-03  4.2115e-03  2.2840 0.0223741 *
`co2(2018)`               -2.5948e-02  5.8261e-03 -4.4538 8.435e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comment

Now, when we test using sandwich Covariance matrix, then all covariates except Renewable energyis highly significant and tobacco is significant at 0.01% level of significance. Standard errors of coefficients are same as upper table and stable.

Now, we will again check for significance of parameter estimates using Likelihood Ratio test, Rao’sScore test for the above model.

Likelihood Ratio Statistic

drop1(fm_nbin, test = "LRT")

```
> ## Likelihood Ratio Statistic
> drop1(fm_nbin, test = "LRT")
Single term deletions

Model:
No_of_deaths ~ merged_data$`Methane(2018)` + merged_data$`exp(2018)` +
    merged_data$`nitrous_oxide(2018)` + merged_data$`renewable energy(2018)` +
    merged_data$`tobacco(2018)` + merged_data$`co2(2018)` + offset(log(pop_total_2018))
                                  Df Deviance    AIC     LRT  Pr(>Chi)
<none>                                143.63 3928.8
merged_data$`Methane(2018)`        1   146.05 3929.2  2.4206  0.119746
merged_data$`exp(2018)`            1   166.24 3949.4 22.6127 1.982e-06 ***
merged_data$`nitrous_oxide(2018)`  1   145.86 3929.0  2.2316  0.135214
merged_data$`renewable energy(2018)` 1 145.80 3929.0  2.1691  0.140807
merged_data$`tobacco(2018)`        1   153.62 3936.8  9.9932  0.001571 **
merged_data$`co2(2018)`            1   157.90 3941.1 14.2749  0.000158 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Rao's Score Statistic

drop1(fm_nbin, test = "Rao")

```
> ## Rao's Score Statistic
> drop1(fm_nbin, test = "Rao")
Single term deletions

Model:
No_of_deaths ~ merged_data$`Methane(2018)` + merged_data$`exp(2018)` +
    merged_data$`nitrous_oxide(2018)` + merged_data$`renewable energy(2018)` +
    merged_data$`tobacco(2018)` + merged_data$`co2(2018)` + offset(log(pop_total_2018))
                                    Df Deviance    AIC Rao score  Pr(>Chi)
<none>                                   143.63 3928.8
merged_data$`Methane(2018)`          1   146.05 3929.2    2.9598 0.0853574 .
merged_data$`exp(2018)`              1   166.24 3949.4   22.7377 1.857e-06 ***
merged_data$`nitrous_oxide(2018)`    1   145.86 3929.0    2.4440 0.1179730
merged_data$`renewable energy(2018)` 1   145.80 3929.0    2.1717 0.1405688
merged_data$`tobacco(2018)`          1   153.62 3936.8   10.5893 0.0011374 **
merged_data$`co2(2018)`              1   157.90 3941.1   13.0507 0.0003032 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comment

All three tests, Wald test (not using sandwich variance covariance matrix), LRT test, Rao’s score test is depicting that all the variables except nitrous oxide and renewable energy are significant. Only difference is that when we are doing Wald test using Sandwich variance-covariance matrix, all the variables except renewable energy are highly significant. Now, for confirming and choosing the best-model, we will do Stepwise variable selection.

Stepwise Variable Selection

stepAIC(fm_nbin,trace = 0)

```
> stepAIC(fm_nbin,trace = 0)

Call:  MASS::glm.nb(formula = No_of_deaths ~ merged_data$`Methane(2018)` +
    merged_data$`exp(2018)` + merged_data$`nitrous_oxide(2018)` +
    merged_data$`renewable energy(2018)` + merged_data$`tobacco(2018)` +
    merged_data$`co2(2018)` + offset(log(pop_total_2018)), data = merged_data,
    link = log, init.theta = 8.95413603)

Coefficients:
                          (Intercept)           merged_data$`Methane(2018)`
                           -1.478e+00                             6.855e-07
              merged_data$`exp(2018)`     merged_data$`nitrous_oxide(2018)`
                           -5.050e-02                            -1.780e-06
merged_data$`renewable energy(2018)`           merged_data$`tobacco(2018)`
                            2.001e-03                             9.619e-03
              merged_data$`co2(2018)`
                           -2.595e-02

Degrees of Freedom: 140 Total (i.e. Null);  134 Residual
Null Deviance:      206.7
Residual Deviance: 143.6        AIC: 3931
```

Comment

The above Stepwise model selection criterion gives Methane (2018), exp(2018), nitrous_oxide(2018) ,renewable energy(2018),tobacco(2018) and cco2(2018) as the significant variables for Mortality.

Model Comparison

Having fitted several count data regression models to the mortality data, it is, of course, of interest to understand what these models have in common and what their differences are. In this section, we show how to compute the components of Table 2 and provide some further comments and interpretations.

Type Distribution Method object	ML model_pois	Poisson Adjusted model_pois (vcov.=sandwich)	Quassi-Poisson Model_qpois	NB ML fm_nbin
Intercept	-1.595e+00 (0.0002504)	-1.595e+00 (0.1912342)	-1.595e+00 (0.1101989)	-1.478e+00 (0.1244268)
Methane	6.754e-07 (3.521268e-10)	6.754e-07 (1.317709e-07)	6.754e-07 (0.0000002)	6.855e-07 (0.0000005)
Expenditure	-3.354e-02 (0.0000204)	-3.354e-02 (0.0149931)	-3.354e-02 (0.0089674)	-5.050e-02 (0.0108382)
Nitrous oxide	-1.626e-06 (7.564155e-10)	-1.626e-06 (2.130128e-07)	-1.626e-06 (0.0000003)	-1.780e-06 (0.0000012)
Renewable energy	1.457e-03 (0.0000030)	1.457e-03 (0.0018042)	1.457e-03 (0.0013387)	2.001e-03 (0.0013582)
Tobacco	5.231e-03 (0.0000060)	5.231e-03 (0.0034472)	5.231e-03 (0.0026592)	9.619e-03 (0.0031047)
CO_2	-1.664e-02 (0.0000199)	-1.664e-02 (0.0102672)	-1.664e-02 (0.0087573)	-2.595e-02 (0.0068116)

No of parameters	7	7	7	8
logL	-Inf	-Inf	NA	-1957
AIC	Inf	Inf	NA	3931
BIC	Inf	Inf	NA	3954

Table 2: Summary of fitted count regression models for mortality data: coefficient estimates fromcount model (both with standard errors in parentheses), number of estimated parameters, maximized log-likelihood, AIC, BIC

```
fm <- list("ML-Pois" = model_pois, "Quasi-Pois" = model_qpois, "NB"
= fm_nbin)sapply(fm, function(x) coef(x)[1:8]) %>% na.omit()
```

```
kable(cbind("ML-Pois" =
    sqrt(diag(vcov(model_pois))),"Adj-Pois" =
    sqrt(diag(sandwich(model_pois))),
    sapply(fm[-1], function(x) sqrt(diag(vcov(x)))[1:8])) %>% na.omit())
```

```
kable(rbind(logLik = sapply(fm, function(x) round(logLik(x),
        digits = 0)),AIC = sapply(fm, function(x) round(AIC(x),
        digits = 0)),
        BIC = sapply(fm, function(x) round(BIC(x), digits
    = 0)),Df = sapply(fm, function(x) attr(logLik(x),
    "df"))))
```

From table 2, we see that overall estimated mean functions of all glm models are similar; whereas standard errors of Quassi-Poisson and Negeative binomial are similar than other two models.

In summary, the models are not too different with respect to their fitted mean functions

	ML-Pois	Quasi-Pois	NB
logLik	-Inf	NA	-1957
AIC	Inf	NA	3931
BIC	Inf	NA	3954
Df	7	7	8

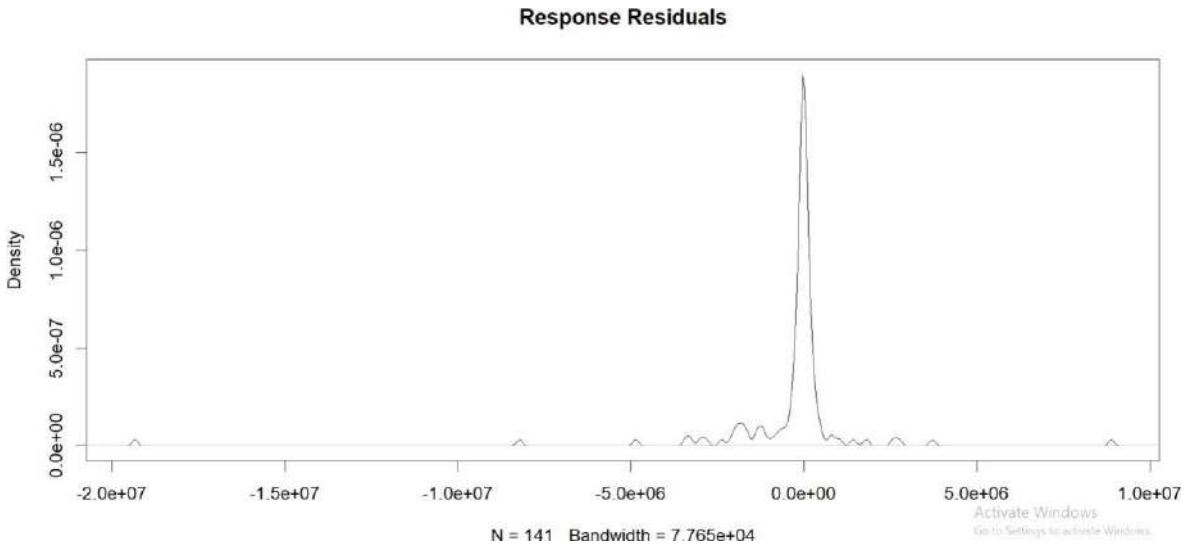
The **ML Poisson model is clearly inferior to all other fits**. The quasi-Poisson model and the sandwich- adjusted Poisson model are not associated with a fitted likelihood. **The negative binomial already improves the fit dramatically and also accommodate overdispersions**. This also reflects that the over-dispersion in the data is captured better by the negative-binomial-based models than the plain Poisson model.

So, here **We prefer to work with Negeative-Biomial model for further diagnostic checking**.

Residual diagnostic Checking

(a) Response Residuals

```
rr = resid (fm_nbin, type = "response")
plot(density(rr), main="Response Residuals")
```



Response residuals are $y_i - \hat{\mu}_i$ and used in linear regression, but not meaningful for Generalized linear model.

(b) Pearsonian Residuals

W.k.t. Pearsonian residuals are defined by,

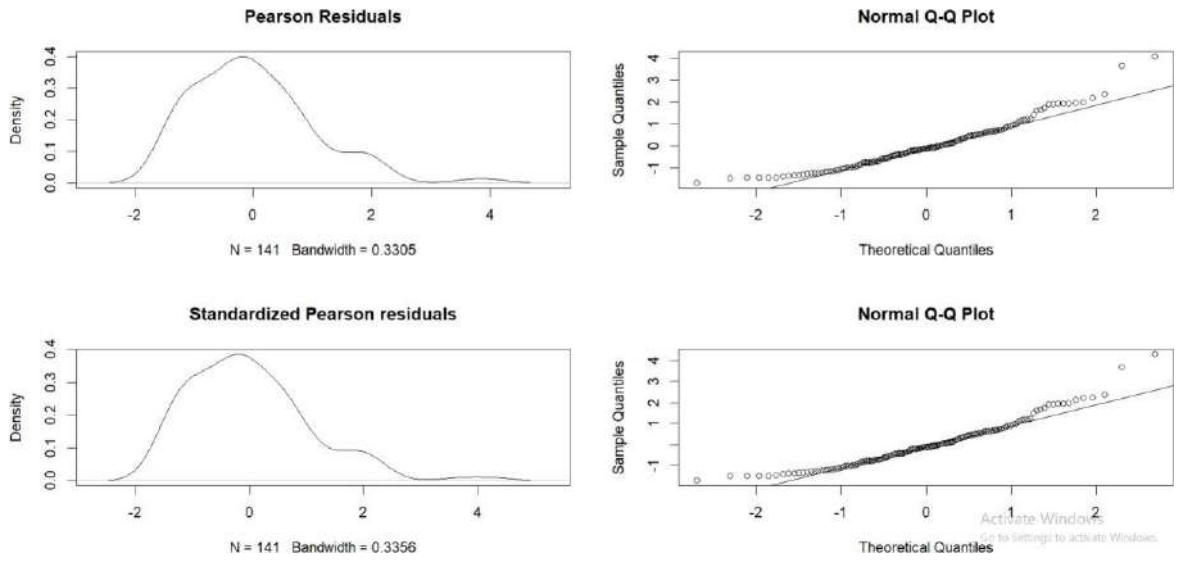
$$\hat{e}_i^p = \frac{y_i - \hat{\mu}_i}{V(\hat{\mu}_i)}$$; V(.) is the variance function

In this way, effect of non-constant variance is considering by dividing.

```
par(mfrow=c(2,2))
#Original Pearson residuals
rp = resid(fm_nbin, type = "pearson")
plot(density(rp), main = "Pearson Residuals")
qqnorm(rp)
qqline(rp)
```

Standardized Pearson residuals

```
srp = rstandard (fm_nbin, type = "pearson")
plot(density(srp), main = "Standardized Pearson residuals")
qqnorm(srp)
qqline(srp)
```



Comment

Density plot of Pearsonian and standardized Pearsonian residuals are similar. Q-Q plot shows that, tail values are deviating from diagonal line by a large extent. So, distribution of Pearsonian residuals don't have normal distribution.

(c) Deviance Residuals

Deviance residuals are,

$$D = 2logL(y,y) - 2logL(y,\hat{\mu}) = \sum_{i=1}^{n} d_i \ ;$$

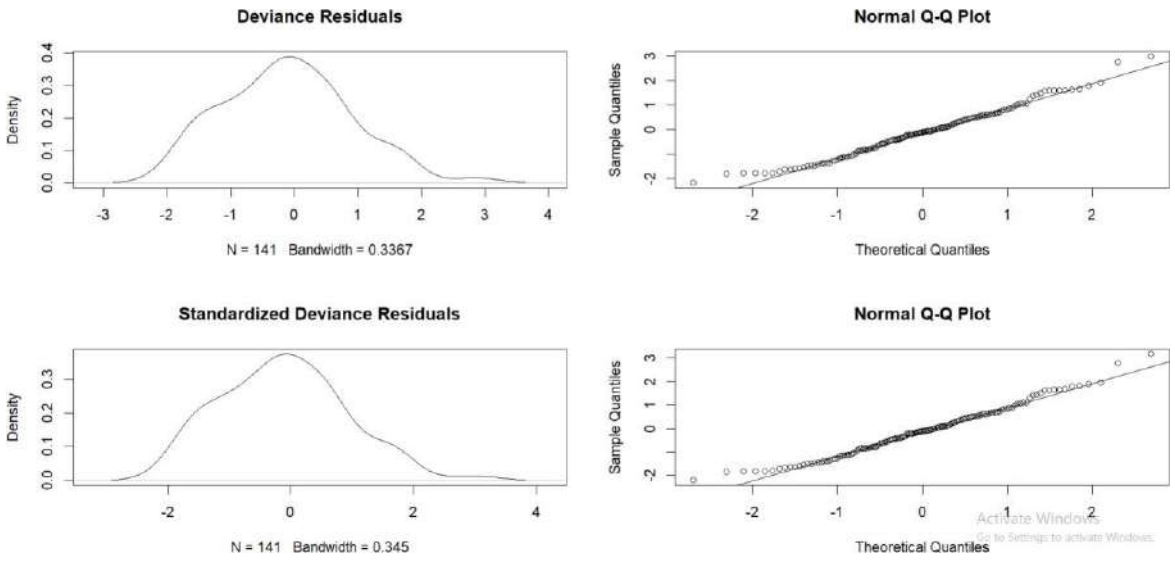
Where logL(y,$\hat{y}$) = Saturated model with all data points y

logL(y,$\hat{\mu}$) = Fitted model with $\hat{\mu}$.

```
par(mfrow=c(2,2))
#Original Deviance Residual
rd = resid(fm_nbin, type = "deviance")
plot(density(rd), main="Deviance Residuals")
qqnorm(rd)
qqline(rd)
```

Standardized Deviance Residuals

```
srd = rstandard(fm_nbin, type = "deviance")
plot(density(srd),main="Standardized Deviance
Residuals")
qqnorm(srd)
qqline(srd)
```



Comment

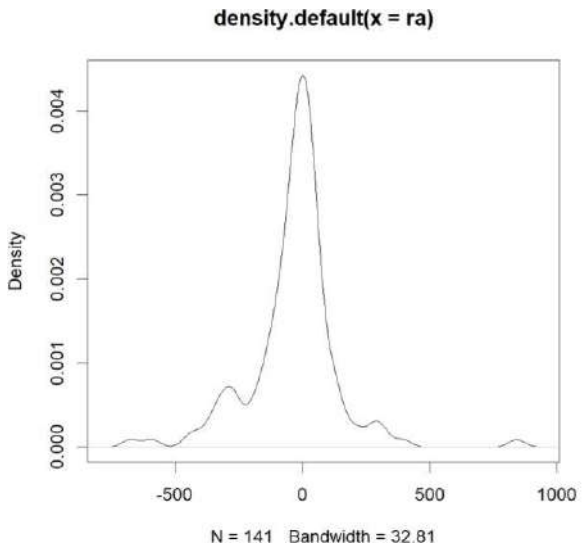
Also, Q-Q plot shows that, tail values are deviating but not like Deviance residuals also it doesn't have normal distribution. Deviance residuals are preferable to Pearsonian residuals. Density plot of Deviance and standardized Deviance residuals are similar.

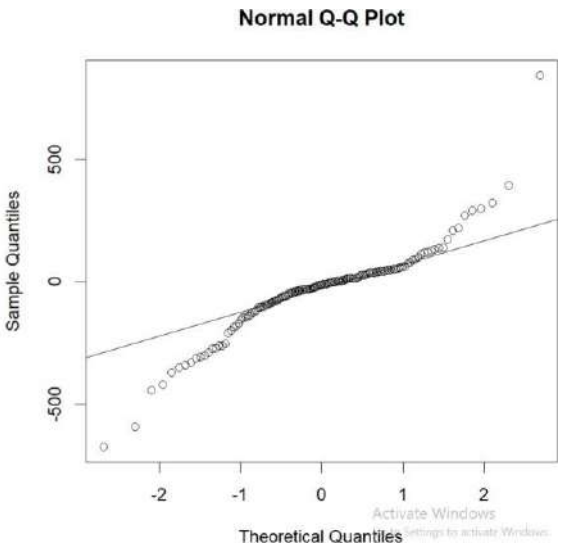
(d) Anscombe Residuals

Unlike Pearsonian, Deviance resiuals Anscombe proposed a residual based on A(y) instead of ywhere the function A(.) is so choosen as to make the dist^n of the residual closest to normality.

$$A(\mu) = \int_{-\infty}^{\mu} V^{-\frac{1}{3}}(t)dt$$

```
par(mfrow=c(1,2))
ra = anscombe.residuals(fm_nbin, fm_nbin$theta)
plot(density(ra))
qqnorm(ra)
qqline(ra)
```





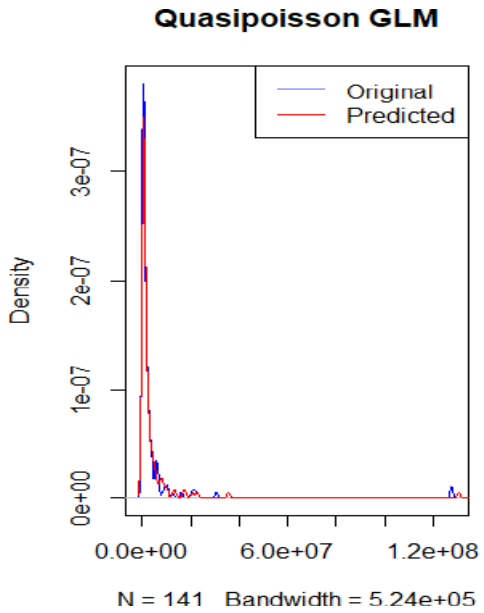
Comment

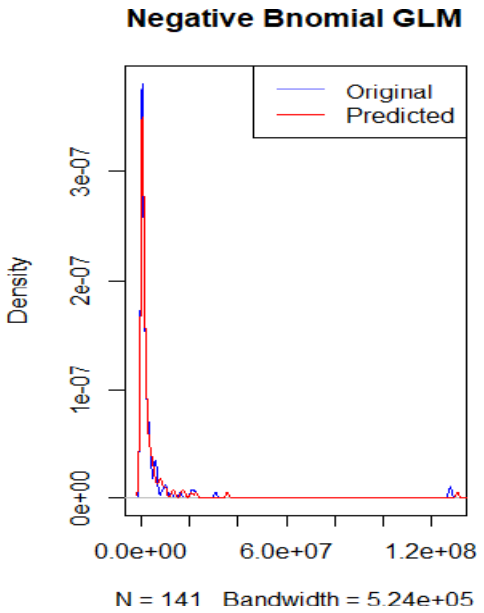
Q-Q plot shows that tail values are deviating by a large extent i.e. flat tailed; non-normal. Also, density curves of Anscombe residuals are long tail than normal dist^{n} and also spikes at tails. So,skewed distribution rather than normal.

Checking the Model Assumptions

I. Checking densities of y and $\hat{y}$

If predicted and original values are close, then fitted model can be said to be of good enough





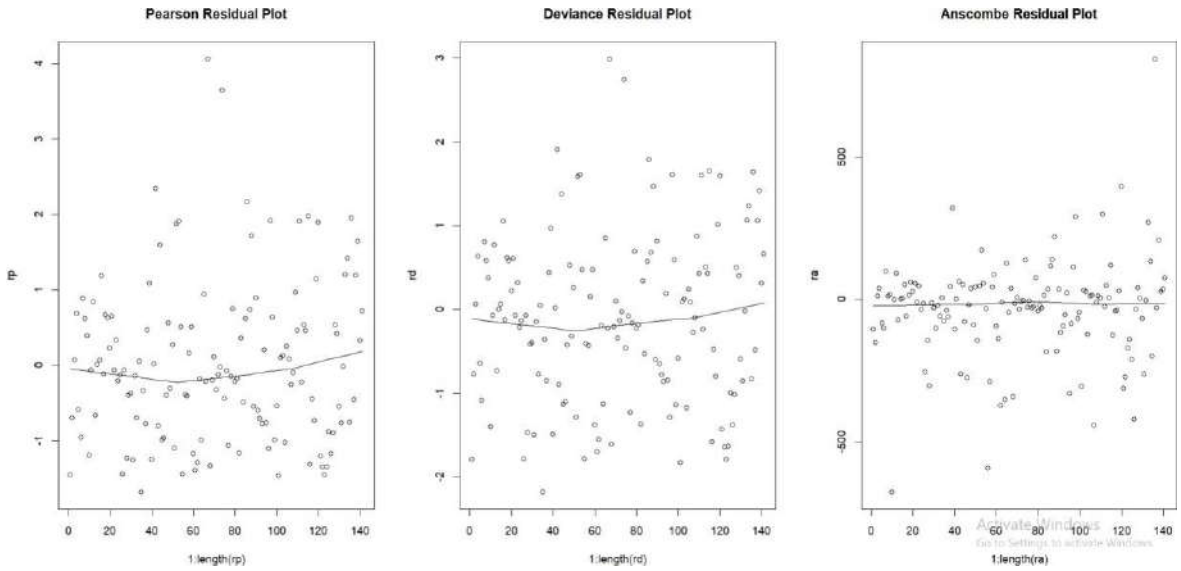
Comment

For both the models fitted and original values are similar. So, both the models are capable of considering overdispersion of data and also fitted well.

II. ***Checking Independence of observations***

W.k.t, no pattern in residual plot indicates indepedence of observatios.

```
par(mfrow=c(1,3))
scatter.smooth(1:length(rp), rp, col = "grey",main = "Pearson Residual Plot")
scatter.smooth(1:length(rd), rd, col = "grey",main = "Deviance Residual Plot")
scatter.smooth(1:length(ra), ra, col = "grey",main = "Anscombe Residual Plot")
scatter.smooth (rstandard(fm_nbin, type='deviance'), col='red',ylab = "Deviance Residuals")
```



Comment

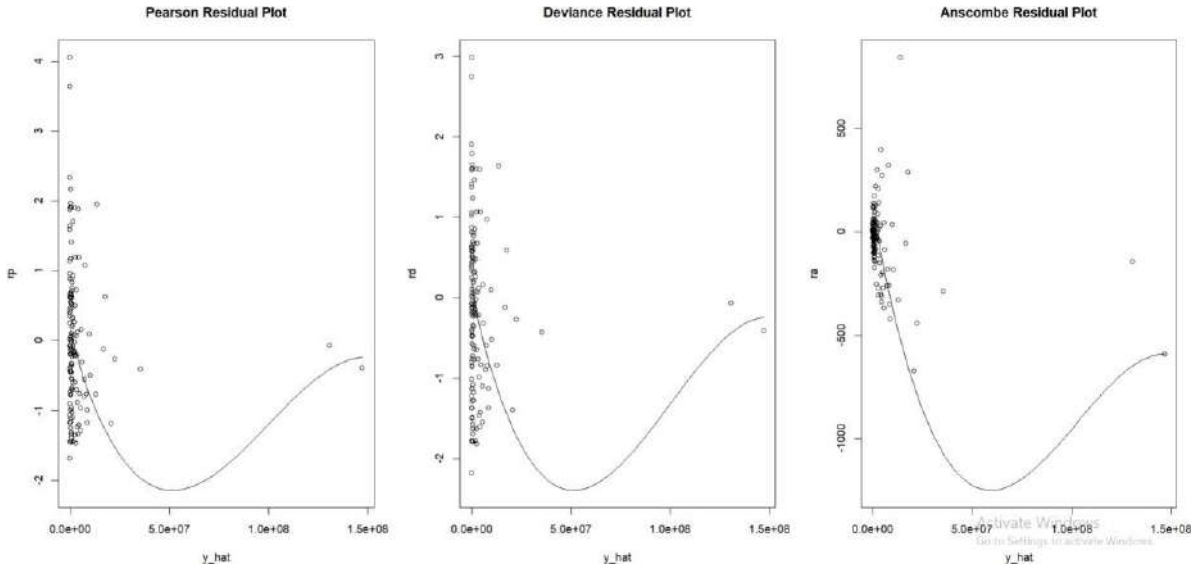
All the residuals except Anscombe residuals show that residuals having downward trend rather than random; which indicated that, some of the observations may not be independent.

III. ***Plot to check systematic component***

To check adequacy of systematic component g(η) we have to plot residuals against fitted values $\hat{y}$. No pattern in plot indicates adequacy of systematic component.

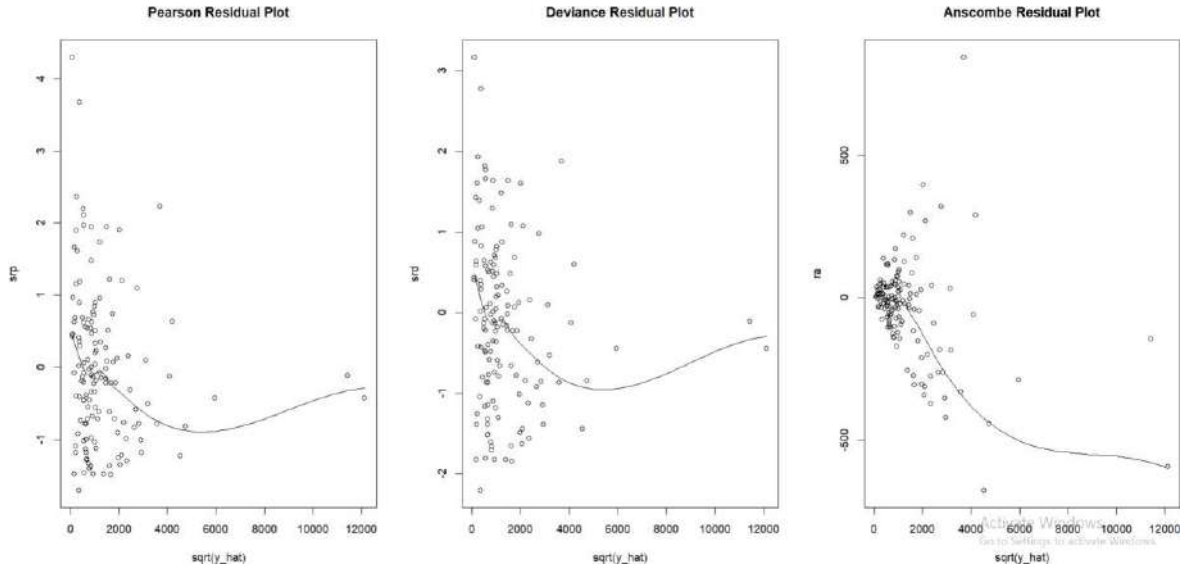
```
par(mfrow=c(1,3))
scatter.smooth(y_hat, rp,main = "Pearson Residual Plot")
scatter.smooth(y_hat, rd,main = "Deviance Residual Plot")

scatter.smooth(y_hat, ra,main = "Anscombe Residual Plot")
```

```
scatter.smooth(y_hat, ra,main = "Anscombe Residual Plot")
scatter.smooth(predict(fm_nbin, type='response'), rstandard(fm_nbin, type='deviance'),
col='blue', xlab="Fitted values",ylab="Deviance Residuals")
```



In this figure, we see that first the curve decreases then increases, so there is showing a pattern (like presence of quadratic term) from the plot. So, in this case we can do variance stabilization transformation on $\hat{y}$ that is square root of $\hat{y}$. Again, plotting we get,

```
par(mfrow=c(1,3))
scatter.smooth(sqrt(y_hat), rp,main = "Pearson Residual Plot")
scatter.smooth(sqrt(y_hat), rd,main = "Deviance Residual Plot")
scatter.smooth(sqrt(y_hat), ra,main = "Anscombe Residual Plot")
```

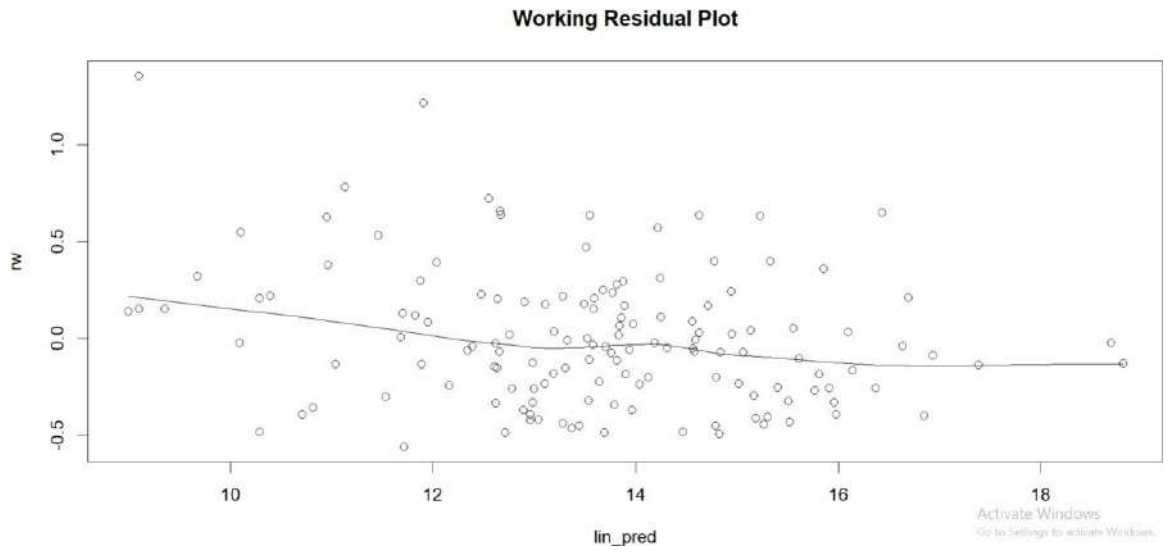


After Variance stabilizing transformation, now the points spread along horizontal axis exceptAnscombe residuals than before. For anscombe it's tend in downward.

IV. ***Plot to check link function***

Here, we plot working residuals against linear predictor as a check for adequacy of selected linkfunction.

```
par(mfrow=c(1,1))
rw = resid(fm_nbin, type="working")
lin_pred = unique(fm_nbin$linear.predictors)#plot
scatter.smooth(lin_pred, rw, main = "Working Residual Plot")
```



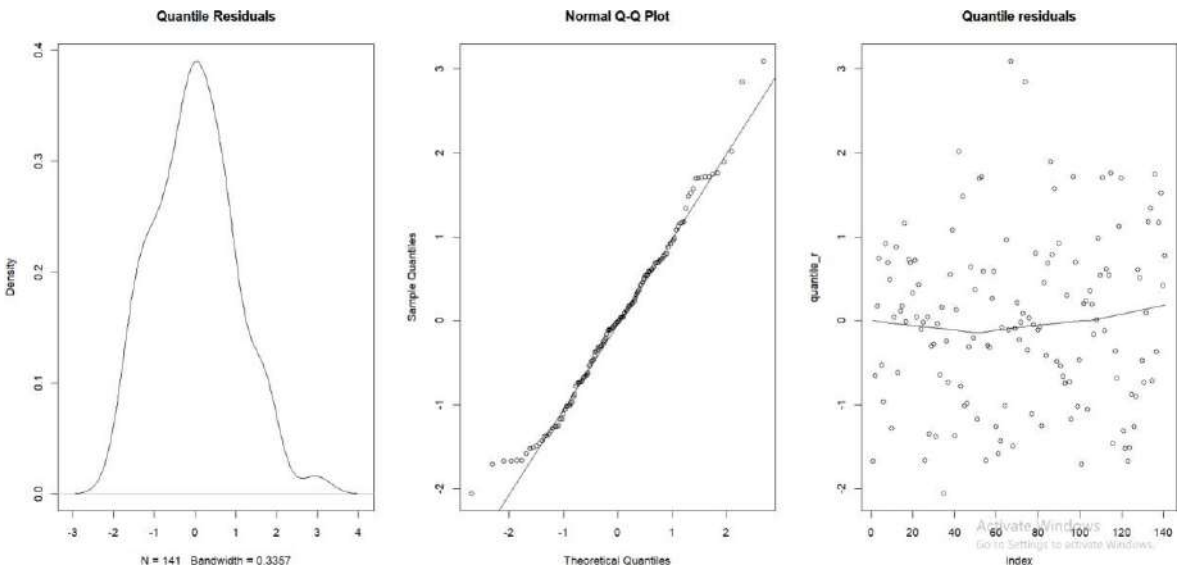
Comment:

Above plot has no pattern showing that the the selected link is appropriate.

V. ***Plot to check the random component***

Another alternative method rather than Pearson, deviance and Anscombe residuals is Quantile residuals, which are exactly normally distributed apart from the sampling variability in estimating μ and ϕ, assuming that the correct EDM is used. Quantile residuals are best used for checking randomcomponent in residual plots where trends and patterns are of interest.

```
library("statmod")
par(mfrow=c(1,3))
quantile_r = qresid(fm_nbin)
plot(density(quantile_r), main="Quantile Residuals")
qqnorm(quantile_r)
qqline(quantile_r)
scatter.smooth(quantile_r, main = "Quantile residuals")
```

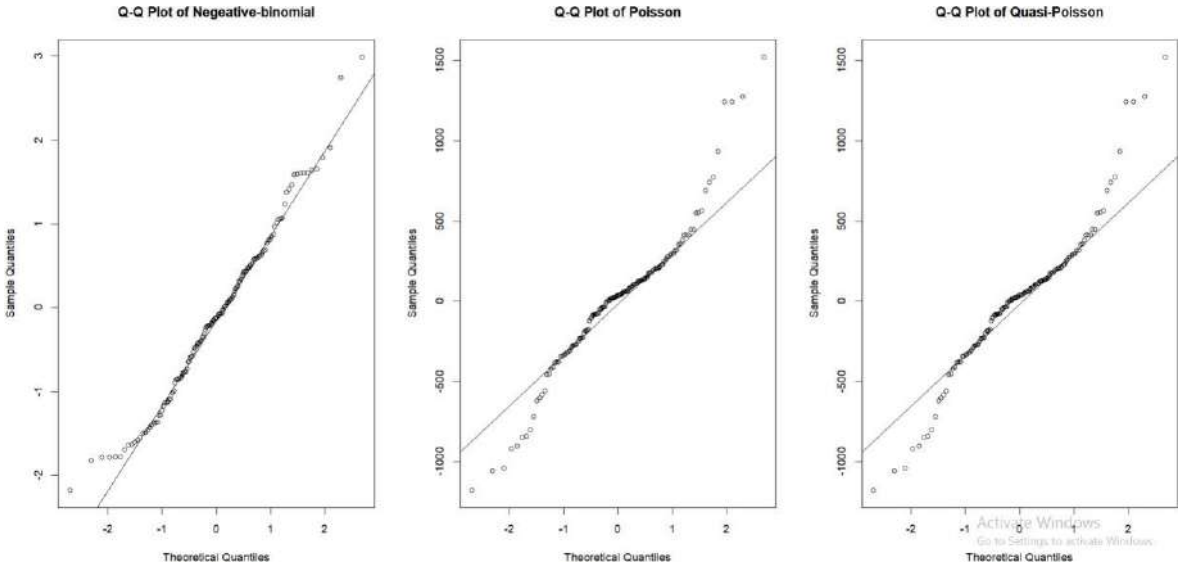


Comment:

Q-Q plot shows that tail values deviating from diagonal line, density curves of this residuals have flat tails. Hence non-normally distributed. there is no certain pattern visible in Quantile residuals.

So, as Q-Q plot shows there is some large residuals present in the model. So, distributionalassmption might be inappropriate.

*VI. **Checking if distribution of y is appropriate***



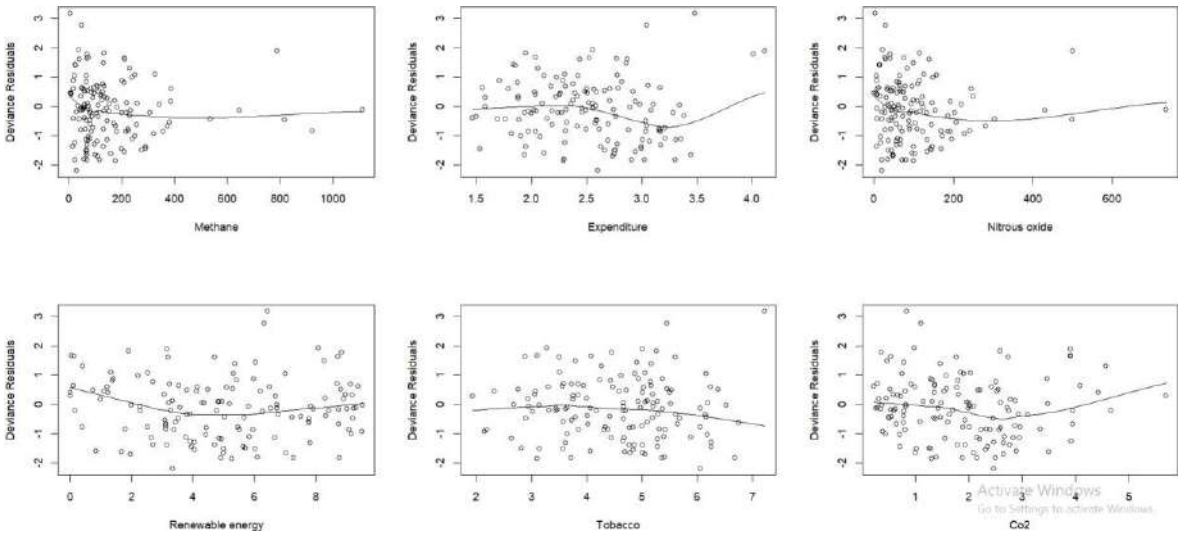
Here, we have done QQ-plot of all three models and see that for poisson as well as Quassi-Poissonlargely deviates in middle as well as in tail values. For Negeative-biomial model, middle values accurately fall on the diagonal line but tail values are deviating from line but not in large extent as other two models. So, we can say that response variable's distribution as Negative-binomial is appropriate

VII. Plot to check linearity of covariates

Linearity between residuals and covariates are the main assumptional block of GLM. If this assumptional block fails to meet, then all model building will go into vain.

Comment:

For all plots, there is no certain pattern exist. so, we say that model is created with adequate variables.



VIII. Plot to check Influential observations

Influential measures of current GLM are,

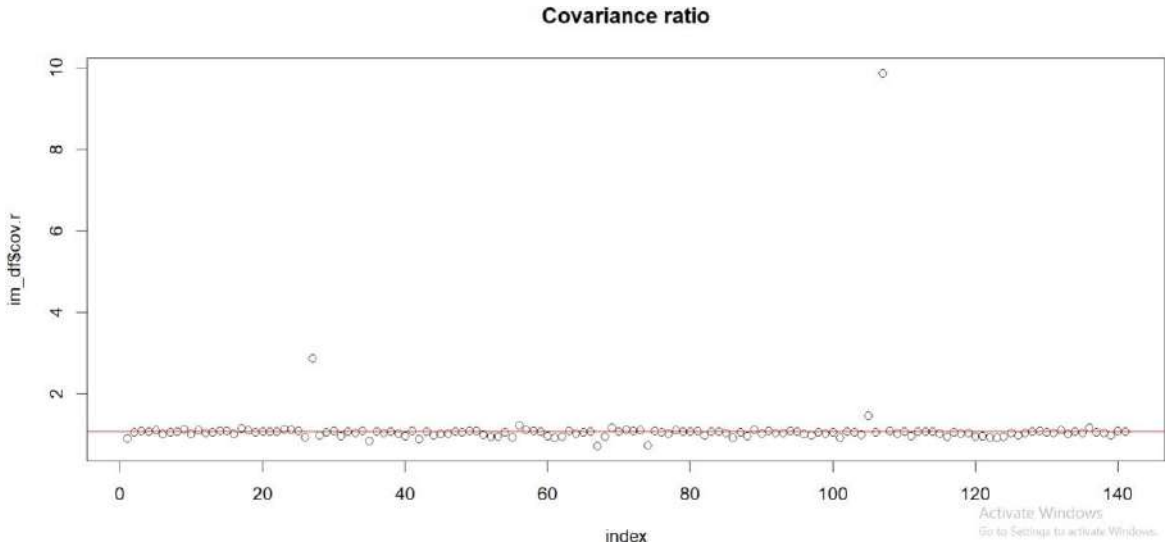
```
im = influence.measures(fm_nbin)
im_df = data.frame(im$infmat)
colSums(im$is.inf)
```

```
> colSums(im$is.inf)
             dfb.1_          dfb.m_$`M dfb.mrgd_dt$`x(2018)`          dfb.m_$`_
                  0                  1                     0                  1
      dfb.m_$e dfb.mrgd_dt$`t(2018)`          dfb.m_$`2              dffit
                  0                  0                     0                  3
         cov.r              cook.d                   hat
                  8                  0                     5
```

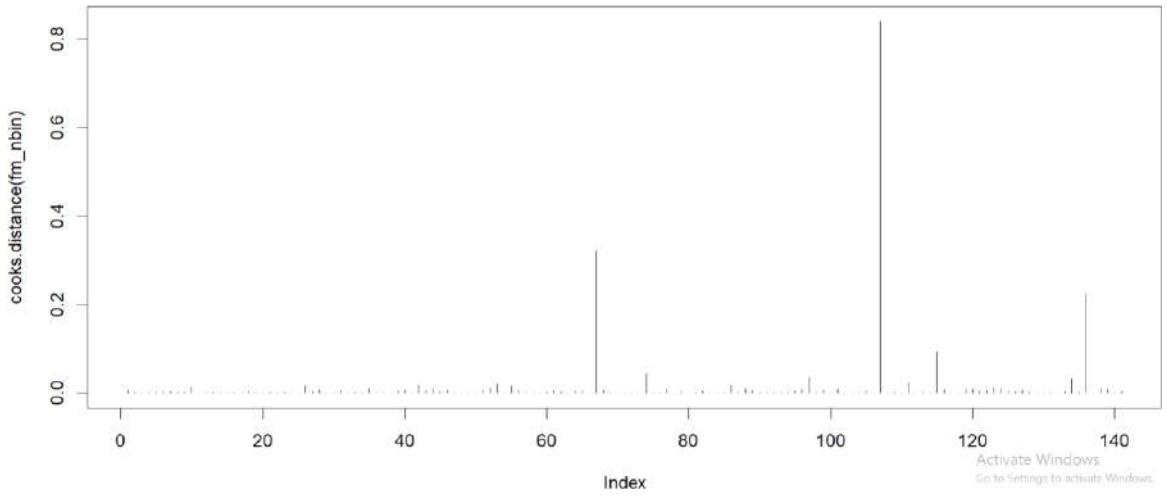
8 observations are identified as influential by Covariance Ratio, dffit identified 3 observations asinfluential.

We can plot the covariance ratio to check the influential observations

```
par(mfrow=c(1,1))
plot(1:length(im_df$cov.r), im_df$cov.r, main = "Covariance ratio", xlab = "index")
abline(h = 1.07, col = "red")
```



```
cd = cooks.distance(fm_nbin)
plot(cd, type="h")
cd[which(cd>0.15)]
```



67	104	136
0.3208772	0.8396847	0.2236554

Thus 67th ,104th and 136th observation i.e. "Malaysia", "United States" and "Portugal" are influential observation in the data.

Conclusion

From the above analyzed methods, Negative Binomial method outer performed to predict systemic component with consideration of overdispersion present in the data.

- The contributable predictors towards response are Methane, Nitrous oxide, CO_2 emission, use of tobacco and health expenditure percentage of GDP.
- We have checked fitted model's diagnostics of fitted models residuals and assumptions. All satisfied theoretical assumptions.

Possible research areas

- In Fig-1.3, when health expenditure is 14%to 17%(highest) variation of mortality is most; which is not expected. So, further investigation can be done about this.
- In this analysis, we have analyzed for all accumulated countries data. How, different countries mortality from CVD, cancer, diabetes or CRD differs from each other and what are those main contributing predictors towards this.

Sample 1 Code

Data Preparation

```
library(MASS) #glm.nb() function
library(pscl)
library(sandwich) #Sandwich Covariances
library(lmtest) #coef(),waldtest()
library(car) #linearHypothesis()
library(MixAll)
library(knitr)
mortality = read.csv("C:/Users/admin/Desktop/STAT 302 /Mortality from CVD, cancer, diabetes orCRD between exact ages 30 and 70 (%).csv",header=TRUE)
methane = read.csv("C:/Users/admin/Desktop/ STAT 302 /Methane emissions (kt of CO2 equivalent).csv",header=TRUE)
exp =read.csv("C:/Users/admin/Desktop/ STAT 302 /Current Health Expenditure(% of GDP).csv",header=TRUE)
nitrous_oxide =read.csv("C:/Users/admin/Desktop/ STAT 302 /Nitrous oxide emissions (thousand metric tons of CO2 equivalent).csv",header=TRUE)
ren_energy =read.csv("C:/Users/admin/Desktop/ STAT 302 /Renewable energy consumption (% of total final energy consumption).csv",header=TRUE)
co2 = read.csv("C:/Users/admin/Desktop/ STAT 302 /CO2 Emission per capita.csv",header=TRUE)
colnames(mortality) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
colnames(methane) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
colnames(exp) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
colnames(nitrous_oxide) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
colnames(ren_energy) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
colnames(co2) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
alcohol_tobacco = read.csv("C:/Users/admin/Desktop/ STAT 302 /Alcohol & Tobacco.csv",header=TRUE)
colnames(alcohol_tobacco) = c("Country","Series Name","2011","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
tobacco = subset(alcohol_tobacco,`Series Name`=="Prevalence of current tobacco use (% of adults)")
pop = read.csv("C:/Users/admin/Desktop/ STAT 302 /Population(30-70).csv",header=TRUE)
colnames(pop) = c("Country","Series Name","2012","2013", "2014","2015","2016","2017","2018","2019","2020")
pop = na.omit(pop)
```

```
attach(pop)
library(magrittr)
library(dplyr)
#Population for Year 2018
pop1 <- pop %>% group_by(Country) %>% dplyr::select(`2018`) %>%
summarise(pop_total_2018 = sum(`2018`))
sum(pop[1:16,9])
mortality = mortality[,c(1,10)]
names(mortality) = c("Country","Mortality(2018)")
methane = methane[,c(1,10)]
names(methane) = c("Country","Methane(2018)")
exp = exp[,c(1,10)]
names(exp) = c("Country","exp(2018)")
nitrous_oxide = nitrous_oxide[,c(1,10)]
names(nitrous_oxide) = c("Country","nitrous_oxide(2018)")
ren_energy = ren_energy[,c(1,10)]
names(ren_energy) = c("Country","renewable energy(2018)")
co2 = co2[,c(1,10)]
names(co2) = c("Country","co2(2018)")
tobacco = tobacco[,c(1,10)]
names(tobacco) = c("Country","tobacco(2018)")
tobacco = na.omit(tobacco)

#Now, data of all variables are ready to go; we need now mortality rate = mortality*pop
merged_data = merge(mortality,pop1,by=c("Country"))
merged_data = na.omit(merged_data)
sum(is.na(merged_data))
mortality_deaths = merged_data
mortality_deaths = mortality_deaths %>% mutate(No_of_deaths =
(`Mortality(2018)`*pop_total_2018)/100)
mortality_deaths = mortality_deaths[,-2]

#Constructing Mortality rate
prob = read.csv("C:/Users/admin/Desktop/ STAT 302 /prob_2018.csv",header = TRUE)
df_new = merge(mortality_deaths,prob,by=c("Country"))
df_new$death_exposed = df_new$pop_total_2018*df_new$Probability_2018.
df_new$rate = df_new$No_of_deaths/df_new$death_exposed#Mortality rate(in 10^5 or 0.1M
population)
df_new$rate = df_new$rate*10^5
df_new$rate = as.integer(df_new$rate)
df_new=df_new[,c(1,2,6)]
#Now, we want to merge data of all variables
merged_data = merge(mortality_deaths,mortality,by=c("Country"))
```

```
merged_data = merge(merged_data,methane,by=c("Country"))
merged_data = merge(merged_data,exp,by=c("Country"))
merged_data = merge(merged_data,nitrous_oxide,by=c("Country"))
merged_data = merge(merged_data,ren_energy,by=c("Country"))
merged_data = merge(merged_data,tobacco,by=c("Country"))
merged_data = merge(merged_data,co2,by=c("Country"))
merged_data=na.omit(merged_data)
merged_data=merged_data[,-4] #Removing Rate
merged_data1 = merged_data[,]
merged_data = merged_data[,-1] #Removing Country



attach(merged_data)
```

To get a first overview of dependent variable, we employ histogram of the observed count frequencies.

Data Exploration

```
#Data Exploration #Histogram of Mortality
h=hist(No_of_deaths,ylim=c(0,170),
breaks=20,xlab="Number of Deaths",ylab="Frequency",main=NULL)h$breaks
h$counts
#Calculate Smoothed Count Densitydrate = density(No_of_deaths) #Add density as frequency
lines(drate$x, drate$y*nrow(merged_data)*2.386e+06 , col = "blue", lwd = 2)
```

Output:

```
h$breaks
[1] 0.00e+00 5.00e+06 1.00e+07 1.50e+07 2.00e+07 2.50e+07 3.00e+07 [8] 3.50e+07 4.00e+07
4.50e+07 5.00e+07 5.50e+07 6.00e+07 6.50e+07 [15] 7.00e+07 7.50e+07 8.00e+07 8.50e+07
9.00e+07 9.50e+07 1.00e+08[22] 1.05e+08 1.10e+08 1.15e+08 1.20e+08 1.25e+08 1.30e+08
h$counts
[1] 154  13  3   1   3   0   1   0   0  0  0   0   0  0  0   0   0
[18]   0   0 0   0   0   0   0   0   2
```

Plot of response variable with independent regressors

```
plot(No_of_deaths~pop_total_2018) plot(No_of_deaths~merged_data$`co2(2018)`) #Percentage
plot(No_of_deaths~merged_data$`Methane(2018)`)
plot(No_of_deaths~merged_data$`exp(2018)`) #Percentage
plot(No_of_deaths~merged_data$`nitrous_oxide(2018)`)
plot(No_of_deaths~merged_data$`renewable energy(2018)`) #Percentage
plot(No_of_deaths~merged_data$`tobacco(2018)`) #Percentage


#Mortality changes with Current Health Expenditure exp_cut =cut(`exp(2018)`,seq(2,17,3),right =
FALSE) ggplot(merged_data, aes(x = exp_cut, y = No_of_deaths)) +geom_boxplot(col =
"blue")+
```

```
labs(x="Current Health Expenditure(% of GDP)",y="Mortality")+scale_y_continuous(limits = c(0,5e+07))

#Mortality rate changes with Renewable energy used
energy_cut = cut(`renewable energy(2018)`,seq(0,100,10),right = FALSE)ggplot(merged_data, aes(x = energy_cut, y = No_of_deaths)) + geom_boxplot(col = "red")+
labs(x="Renewable energy consumption (% of total final energy consumption)",y="Mortality")+
scale_y_continuous(limits = c(0,3e+07))

#Mortality rate changes with Tobacco
tobacco_cut = cut(`tobacco(2018)`,seq(0,100,10),right = FALSE) ggplot(merged_data, aes(x = tobacco_cut, y = No_of_deaths)) +geom_boxplot(col = "orange")+
labs(x="Current tobacco use (% of adults)",y="Mortality")+
scale_y_continuous(limits = c(0,3e+07))
```

Sample No-2

Gamma-Inverse Gamma Models

Saikat Kar

13/08/2021

Primary endpoints

- To draw samples from Posterior distribution where likelihood is gamma distribution and prior is Inverse-gamma model.
- Diagnostics of checking whether chain has converged or not.

```
library(rstan)
library(ggmcmc)
library(bayesplot)
```

Data & model description

- ***Gamma-Inverse Gamma models***

Here, in our model we take likelihood i.e. $X_1, X_2, \ldots, X_p|\theta\ Gamma(n,\theta)$ where pdf of Gamma distribution is

$$p(\underset{\sim}{x}|\theta) = \frac{1}{(\theta^n)^p}\frac{e^{-\frac{\sum_{i=1}^{p} x_i}{\theta}}}{(\Gamma n)^p}\prod_{i=1}^{p} x_i^{n-1}$$

and taking prior distribution of θ ~ InverseGamma (α,β). where pdf of InverseGamma (α,β) is

$$\tau(\theta) = \frac{\beta^\alpha}{\Gamma\alpha}e^{-\frac{\beta}{\theta}}\left(\frac{1}{\theta}\right)^{\alpha+1}$$

Now, our target is to draw random sample from the posterior distribution i.e. from distribution of $\pi(\theta|\underset{\sim}{x})$

Now, we defining models of likelihood and prior as well as parameters as the following:

```
data{
  int<lower=0> N; //Number of Trials
  real<lower=0> x; //random sample values taken from Gamma distribution
}

parameters{
```

```
 real<lower=0> theta; //scale parameter of Gamma distribution; theta(>0) }
model{
 target+= inv_gamma_lpdf(theta|3,4); //Prior distribution
 target+= gamma_lpdf(x|10,theta); //Likelihood
}
```

Here we taking hyperparameters α,β as 3,4 respectively (as W.K.T in InverseGamma(α,β) α,β >0).

As θ~InverseGamma; so theta belong to (0,∞). That's why we define theta in *parameter block* and that belong to $\mathbb{R}^+$.

Taking n of Gamma(n,θ) as 10 and theta is unknown parameter. Now, constructing 4 chains of markov models to draw random samples from required posterior distribution.

```
model_file=file.choose()
N=100
theta=0.5
x=rgamma(N,shape=1,scale = theta)

stan_data = list(N=N,x=sum(x))
stan_model=stan(model_file,      #Stan Program
                data=stan_data,  #named list of data
                chains=4,        #Number of Markov Chain
                iter=2000,       #Total number of iterations per chain
                warmup= 1000,    #Number of Warmup iterations per chain
                refresh=0)
```

Note: Here, instead of running a Markov chain of 8,000; we run 4 chains each for 2000 iterations and treat 1,000 as warmup value (T_0)

After finishing the computations, summaries of the inferences and convergences are shown below:

```
print(stan_model,digits=2)

## Inference for Stan model: Gamma_Inv_Gamma.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## theta   0.34    0.00 0.06   0.24   0.30   0.34   0.38   0.48  1464    1
## lp__  -11.78    0.02 0.70 -13.82 -11.94 -11.51 -11.34 -11.28  1527    1
##
## Samples were drawn using NUTS(diag_e) at Sat Aug 21 18:57:25 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
```

```
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

$\hat{R}$ - potential scale reduction factor on split chains

n_eff is a crude measure of effective sample size(# of samples required to reach convergence)

In the above table, mean is the estimated standard posterior mean,se_mean is the estimated standard error of mean of simulations,sd is the standard deviation.

Diagnostics of checking convergence of chain

1. **Use multiple starting points**
2. **Trace Plots**
3. **Autocorrelation Plots/Autocorrelogram**

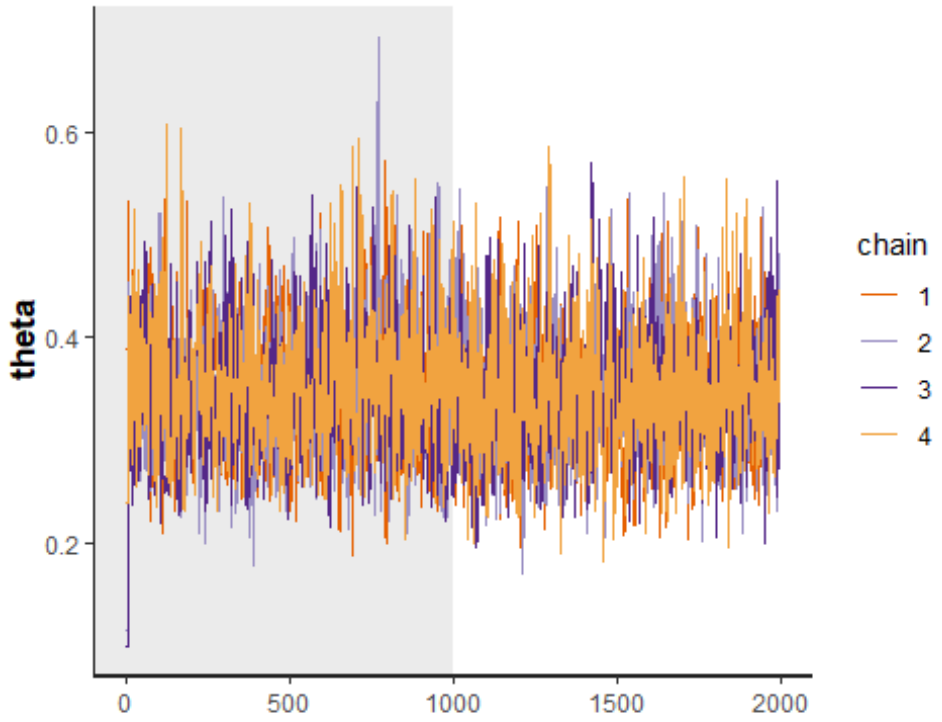
1.*Use multiple starting points*

- ***(Gelman Rubin Test):***

In this model, we see that $\hat{R}$ i.e. estimated potential scale reduction factor is 1; means that the between chains the average variance isn't so large; after certain points it converges.

2. Trace Plots

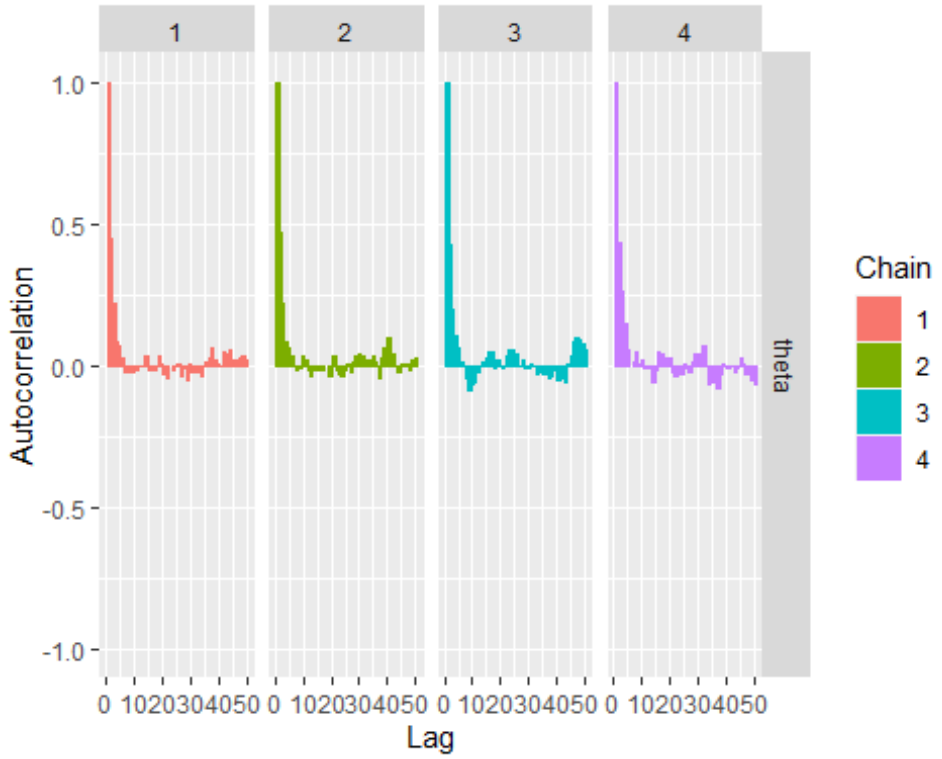
```
traceplot(stan_model,inc_warmup=T)
```



In the dark shaded area i.e., that is of first 1,000 iteration and after 1,000 we see that 4 chains are converging in such a manner that, we can't distinguish 4 chains one from another. So, for confirmation in convergence; we have to look at only at the Trace plot of warmup values.

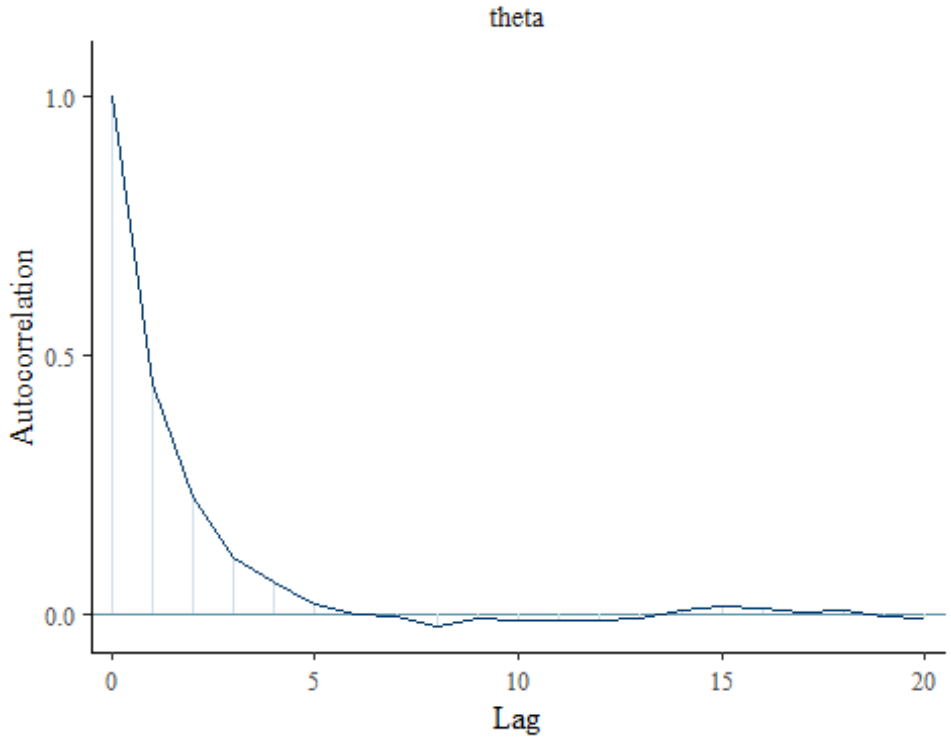
3. Autocorrelation Plot/Autocorrelogram:

```
obj =ggs(stan_model)
ggs_autocorrelation(obj)
```



1st plot showing autocorrelation plot for each chain; after how much lag it is going to thin that means there is absence of dependency/non stationarity in the samples as per our required. We always want to draw random samples from required/target distribution; so if from autocorrelation plot we understand autocorrelation is not coming thin (i.e. significant) after certain points; then we have to take more iterations for drawing random samples. In this fig, we can say there is no dependency present after certain point for each chain.

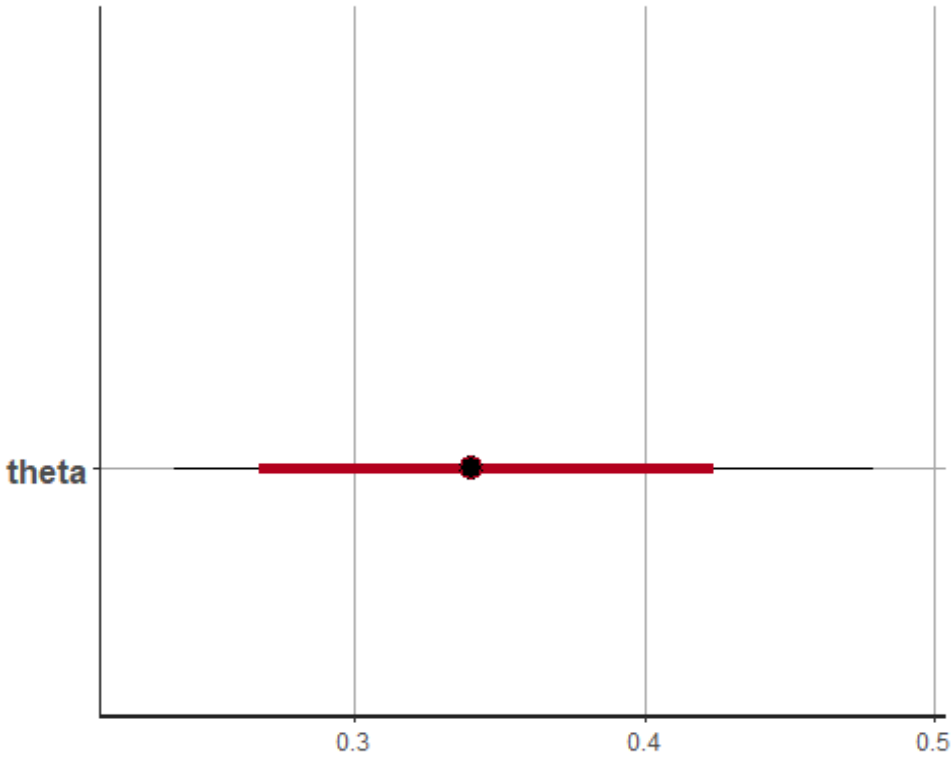
```
gamma_invgamma_fit_score = as.data.frame(stan_model)
mcmc_acf(gamma_invgamma_fit_score,pars="theta")
```



W.K.T. for stationary series the ACF plot dies off very quickly.2nd plot showing after lag 5 there is no more significant values. So, the series of data/samples are stationary

Credible Interval for theta

```
plot(stan_model,ci_level=0.8,outer_level=0.95)
```



Here red highlighted line is 95% credible interval for θ; whereas light black line is the 80% credible interval for θ.

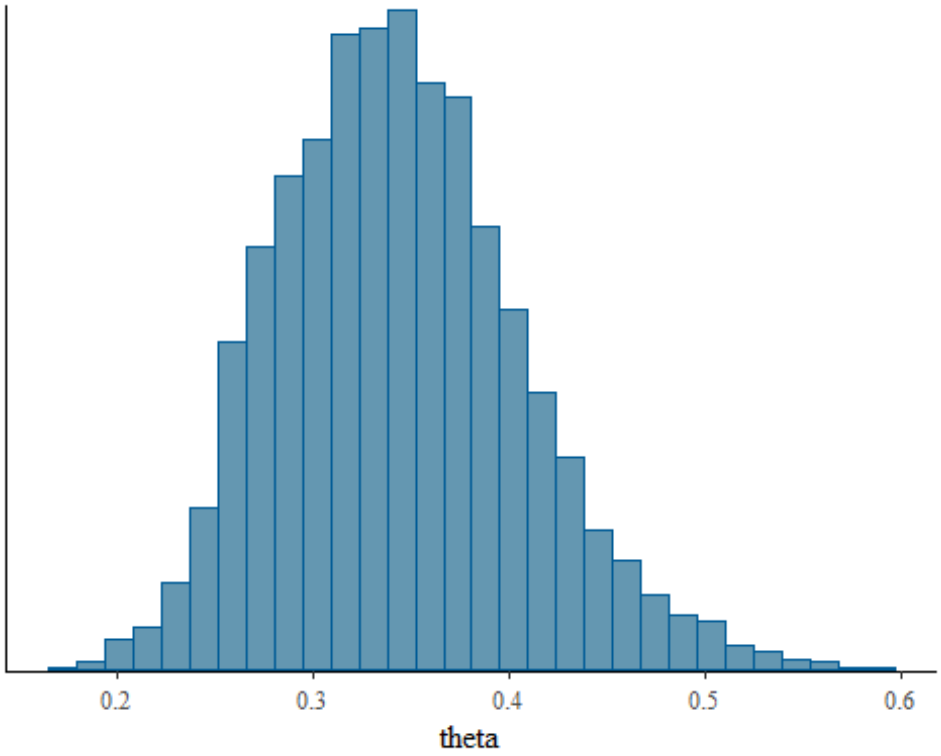
95% confidence Interval

```
print(stan_model,"theta",probs = c(0.05,0.95))
```

```
## Inference for Stan model: Gamma_Inv_Gamma.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##       mean se_mean   sd   5%  95% n_eff Rhat
## theta 0.34       0 0.06 0.25 0.45  1464    1
##
## Samples were drawn using NUTS(diag_e) at Sat Aug 21 18:57:25 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

So, here (0.25,0.45) is the 95% credible interval for θ . Also, we can see Posterior mean i.e. $E(\theta|\underset{\sim}{x})$ = 0.34 which is also Bayes Estimator; if we consider loss function as squared error loss. $L(\theta, \delta(\underset{\sim}{x})) = (\delta(\underset{\sim}{x}) - \theta)^2$

Histogram of Posterior distribution:

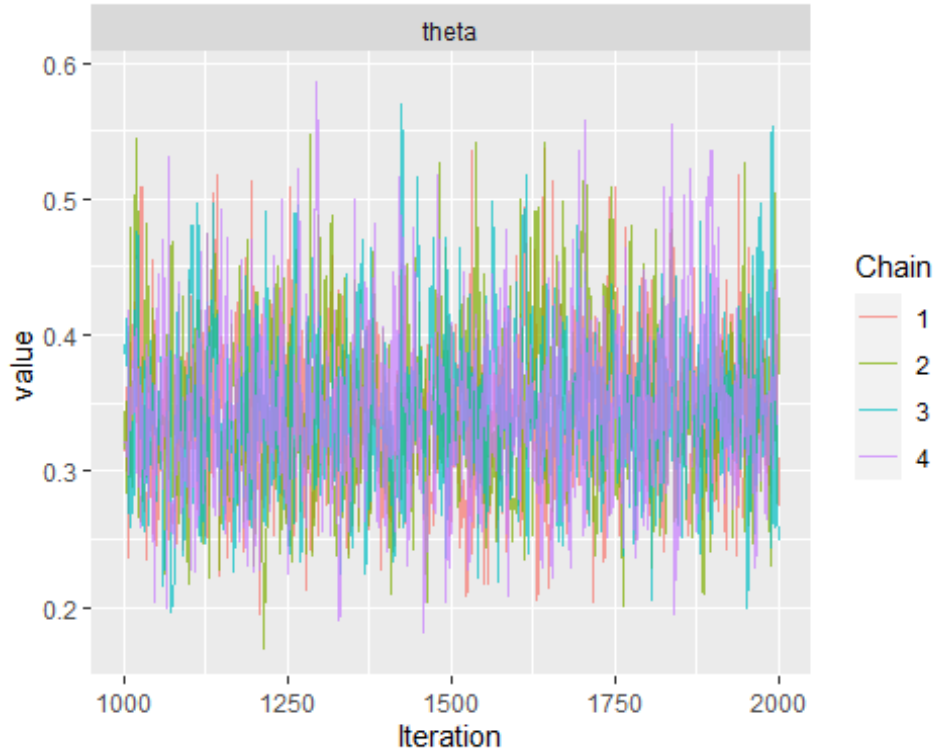


This is the histogram of Posterior distribution;$\pi(\theta|\underset{\sim}{x})$. Basically we drawing likelihood from Gamma(n=10,θ) and prior from InverseGamma(α=3,β=4); so from calculation W.K.T posterior should come from InverseGamma(α+np=1003,β+$\sum_{i=1}^{p} x_i$=59.38003) as $\sum_{i=1}^{p} x_i$=55.38003.

So, basically the above histogram is of distribution of $\pi(\theta|\underset{\sim}{x})$~InverseGamma(1003,59.38003)

Traceplot Of Warmup values

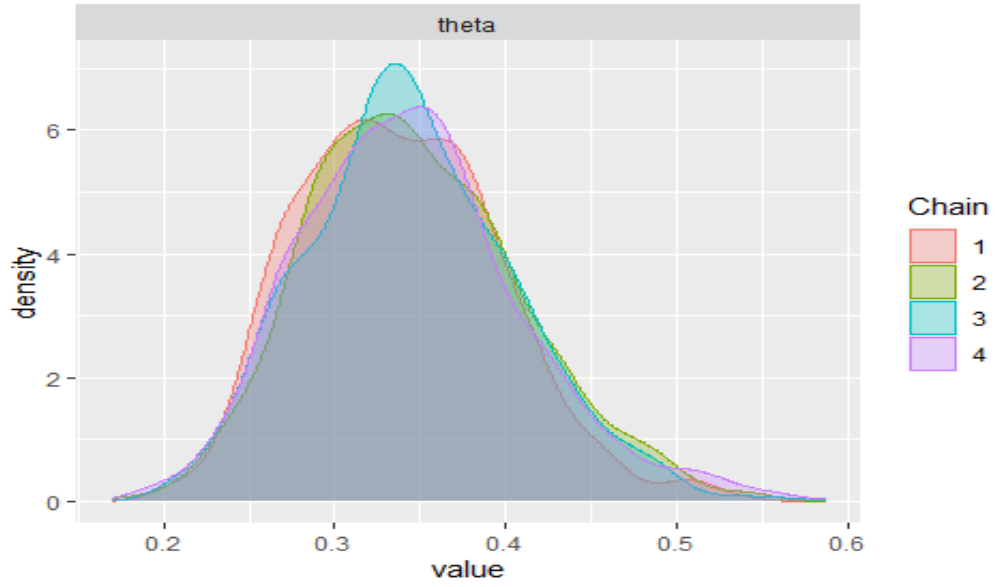
```
ggs_traceplot(obj)
```



This is the traceplot of iteration of 4 chains after warmup i.e. after T_0. From this, we get more clear idea about convergence of marcov chains.

Density plot

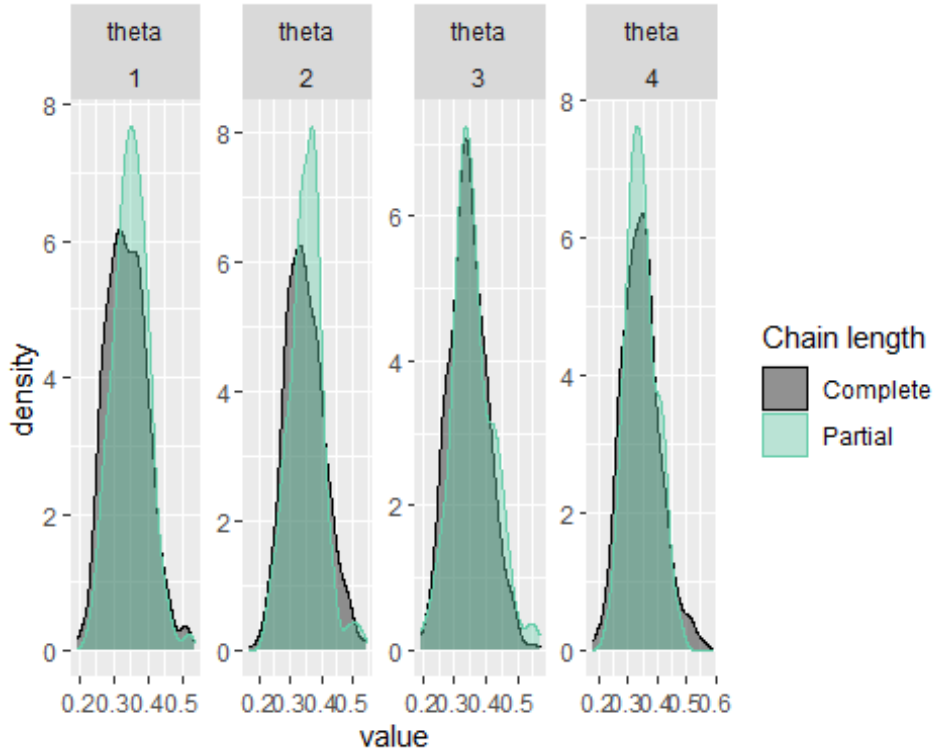
```
ggs_density(obj)
```



This is the density plot of 4 chains of posterior distribution.These density plots are same;overlapping on each other.

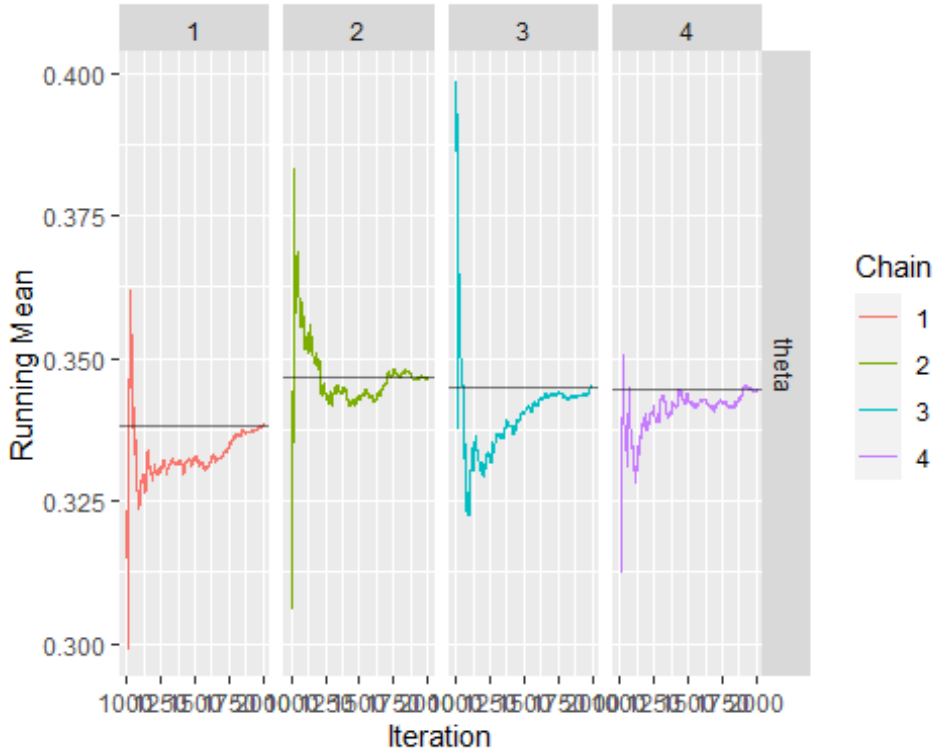
Complete vs Partial plots for each chain

```
ggs_compare_partial(obj)
```



Complete means density plot of total 4000 post warmup samples and partial means 1000 samples for each chain and here is 4 graphs fro each chain.This also shows for each chain; how samples partially drawn for each chain.

```
ggs_running(obj)
```



This shows how samples are going for each chain; i.e. whether it is going to converge quickly or it is taking time for any chain or how much time it's taking.

Conclusion

From ACF plot, we can see after lag5 the chain is showing there is no more significant values. So, the drawn samples from Posterior distribution are stationary. Also, trace plot showing after warmup iterations 4 chains are showing more convergence.

Sample No-3

COVID19's effect on airport traffic: A Data Analysis in R

Dataset info

This dataset shows traffic to and from the Airport as a Percentage of the Traffic volume during the baseline period. The baseline period used for computing this metric is from 1st February to 15th March in 2020.

Download the data from https://drive.google.com/drive/folders/1LWFC344Ng-M8JborpJ2qoeQ8vIkesYi2

Objective

- To analyze reason of difference of percentage of traffic volume across country's airport within country in baseline period.
- To predict percentage of traffic volume for each country's airport for next 15 days.

Let's see how we can analyze step by step

Data Loading and Basics

```
data <- read.csv ("Location/file name.csv")
```

Have a look on data

```
head(data)

str(data)
```

```
tibble [7,247 x 11] (S3: tbl_df/tbl/data.frame)
 $ AggregationMethod: Factor w/ 1 level "Daily": 1 1 1 1 1 1 1 1 1 1 ...
 $ Date             : Date[1:7247], format: "2020-04-03" "2020-04-13" "2020-07-10" "2020-09-02" ...
 $ Version          : num [1:7247] 1 1 1 1 1 1 1 1 1 1 ...
 $ AirportName      : Factor w/ 28 levels "Boston Logan International",..: 14 14 14 14 14 14 14 14 14 14 ...
 $ PercentOfBaseline: num [1:7247] 64 29 54 18 22 59 59 48 20 27 ...
 $ Centroid         : Factor w/ 28 levels "POINT(-104.700315559089 39.8643468206413)",..: 28 28 28 28 28 28 28 28 28 28 ...
 $ City             : Factor w/ 27 levels "Boston","Calgary",..: 25 25 25 25 25 25 25 25 25 25 ...
 $ State            : Factor w/ 23 levels "Alberta","British Columbia",..: 14 14 14 14 14 14 14 14 14 14 ...
 $ ISO_3166_2       : Factor w/ 23 levels "AU","CA-AB","CA-BC",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Country          : Factor w/ 4 levels "Australia","Canada",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Geography        : Factor w/ 28 levels "POLYGON((-104.661254882812 39.8242265704646, -104.661340713501 39.9048641265029, -104.7093200
359 39.905654221"| __truncated__,..: 28 28 28 28 28 28 28 28 28 28 ...
```

Summary of data

```
library(skimr) # if not work, then do install.packages ("skimr")
skim(data)
```

Understanding the Data

Continuous Variables: Version, Percent of Baseline

Categorical Variables: Aggregation Method, Airport Name, Centroid, City, State, ISO_3166_2, Country, Geography

Comments: This is time series data.

Now, Version is somewhat like Id we can drop it. Also, ISO_3166_2, Centroid, Geography these all are being useful when we visualize this in world map in Power BI/Tableau. For analysis, we are left with following variables:

Percent Of Baseline: Numerical

Airport Name, City, State, Country: Categorical Variables.

Now, we will Perform some EDA to bring out insights.

Exploratory Data Analysis

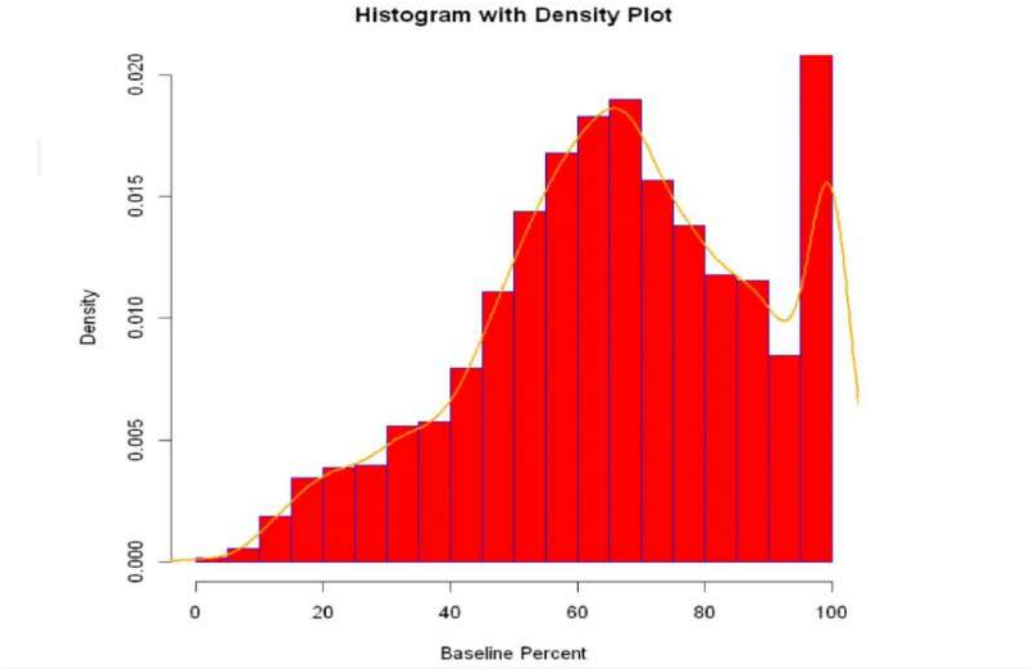
- ***Univariate***

Histogram + density Plot of Percent of Baseline

The only important numerical random variable is the percentage of baseline, represents the numerical random variable in the following:

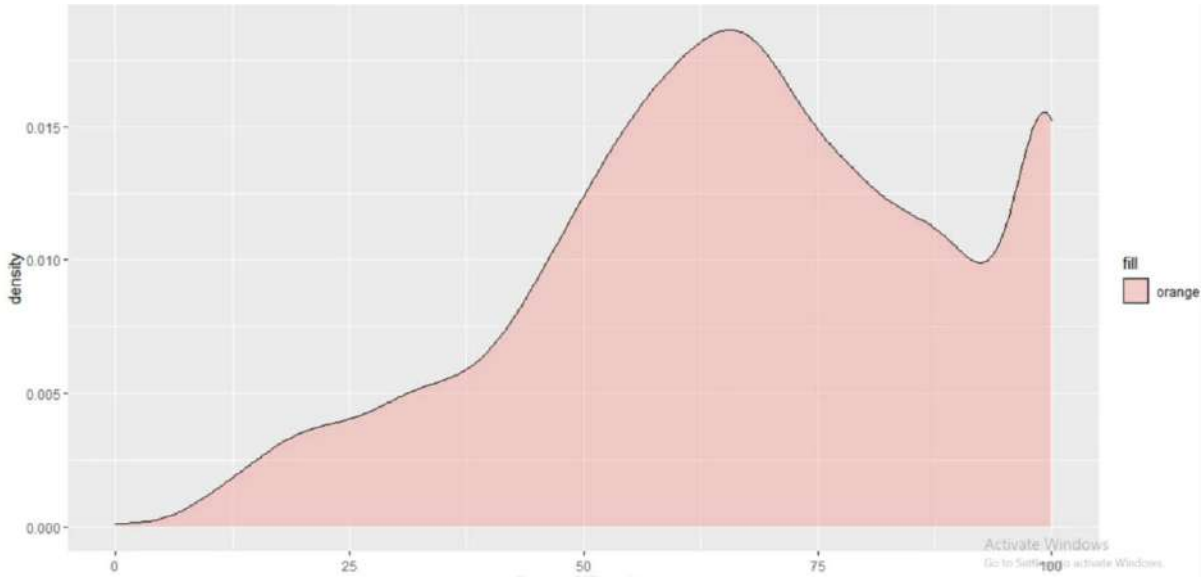
1. *Histogram*

```
hist (Percent Of Baseline,col="red",breaks = seq(0,100,by=5),ylim = c(0,0.02),border =
"blue",prob=TRUE,xlab = "Baseline Percent",ylab = "Density",main="Histogram with Density
Plot")lines(density(percent of Baseline),lwd=2,col="orange")
```



2. ***Kernel Density of Percent Of Baseline***

```
ggplot(data = data,aes(x=PercentOfBaseline,fill=Country)) + geom_density(alpha = 0.2)
```



Comments:

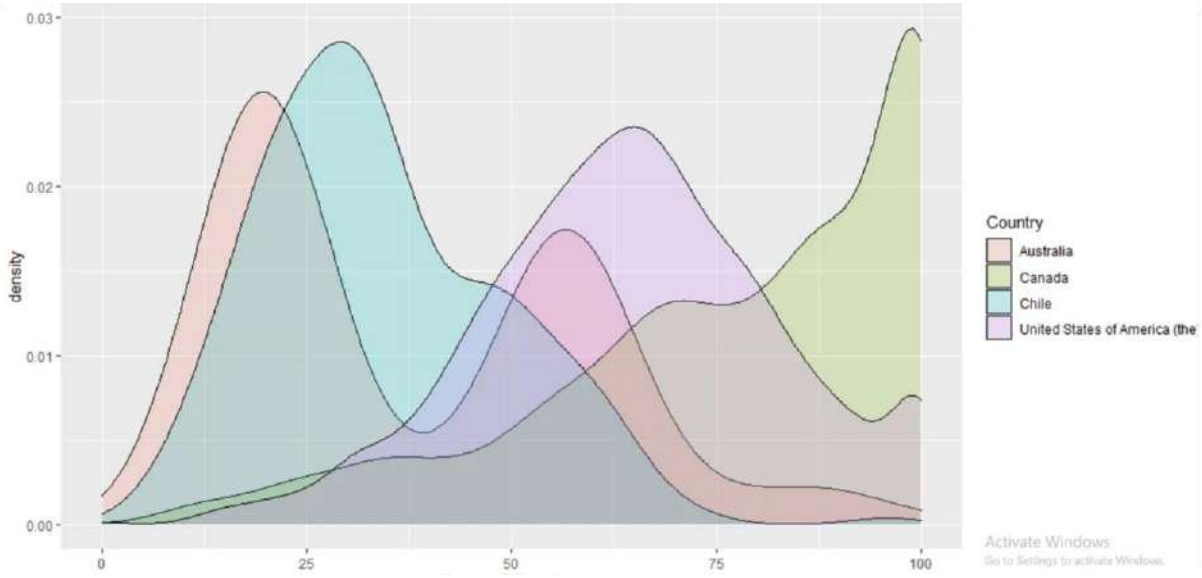
Why there is a sudden increase in densities at tail points? Is it due to there is two modes of this distribution? May be! or, Is it due to influence of some predictors on Percent of Baseline? Let's Explore!

- ***Bivariate***

Numerical vs Categorical

Kernel Density of Percent of Baseline vs Country

```
ggplot (data = data,aes(x=PercentOfBaseline,fill=Country)) + geom_density(alpha = 0.2)
```



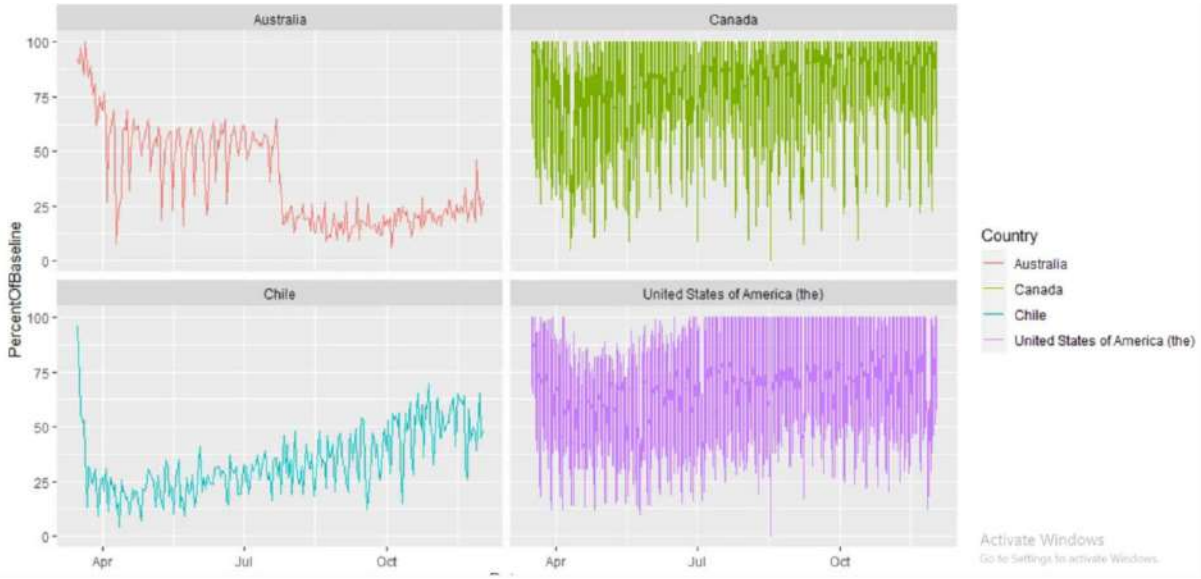
Comments:

1. Australia has two distinct modes of Normal Mixtures.
2. All the countries behaving like normal mixtures.
3. Canada behaves totally different than others country (higher than others).

That's why in the graph of Percent of Baseline we have a sudden increase in tail. That's due to Canada Country.

Percent Of Baseline vs Date (Time Series Plot)

```
ggplot (data = data,aes(x=Date,y=PercentOfBaseline,color=Country)) + geom_line() +
facet_wrap(~Country)
```



Observations:

1. Australia has random fluctuations over time (no trend) as well as sudden dip. It's may be due to some good measurement taken from government to prevent Covid19.
2. Chilie has clearly an increasing trend except at first few days of April month.
3. Canada and USA certain number of random fluctuations rather than any trend over time.

Time Series Modelling

Now, we will only focus on dealing with Time series model of Percent of Baseline over time for each country.

Loading required packages

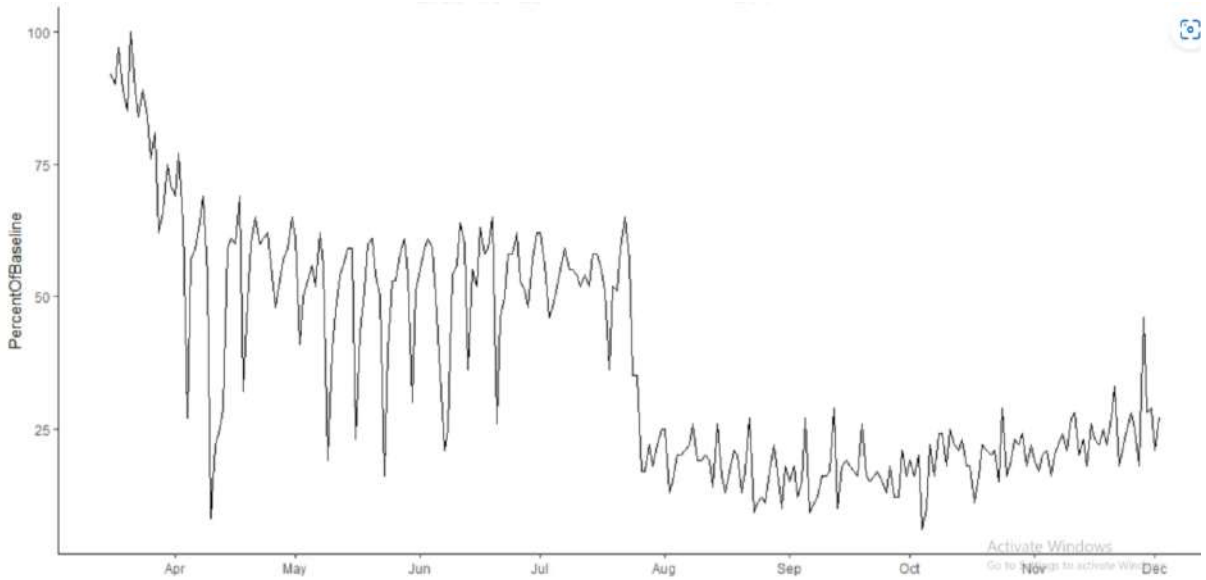
```
library(FinTS)     ##for Arch Test
library(rugarch)  ## for Garch Models
library(tseries)  ## For unit root test
library(dynlm) ##for using lags in the model
```

```
library(vars)  ## for using VAR
library(nlWaldTest)  ## For testing non linear wald test
library(lmtest)     ##for BP test
library(broom)      ## For table presentations
library(car)     ## for robust standard errors
library(sandwich)
library(knitr)
library(forecast)
library(ggplot2)
library(pdfetch)
library(tsbox)
library(stats)
library(zoo)
library(vrtest)
library(nortsTest)
library(rugarch)
library(rmgarch)
```

Australia

First, here we have to organise data according to ordering of time.

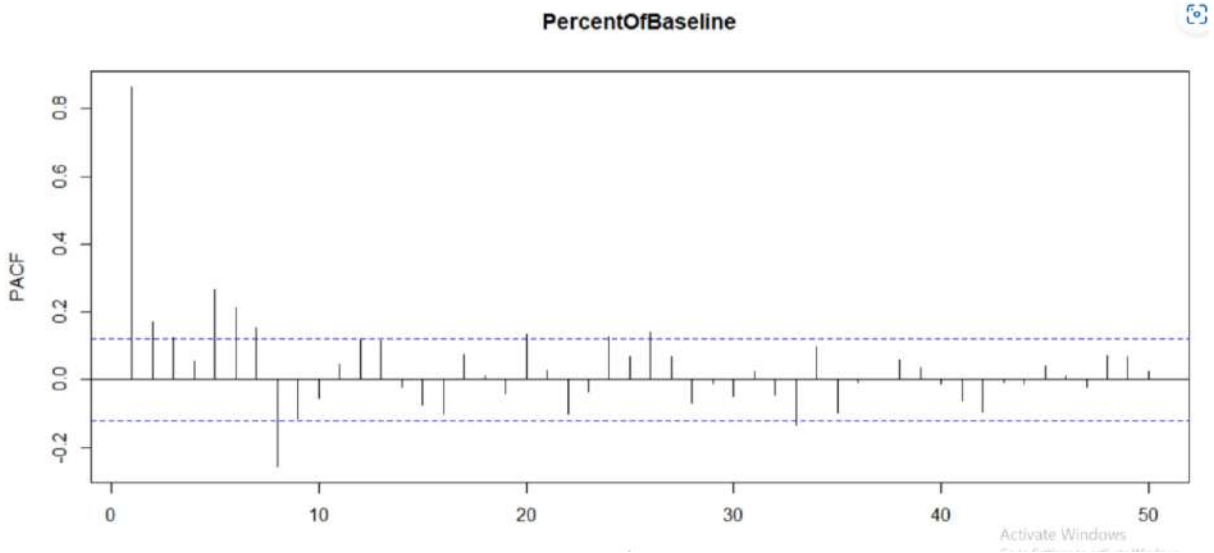
```
data$Date <- as.Date(data$Date, format = "%Y-%b-%d") #converting into time formate
head(data$Date)
data <- data[order(as.Date(data$Date, format = "%d-%m-%Y")),]
View(data)
class(data$Date)#gives date; we can proceed nowdata_australia = data %>% filter(Country ==
"Australia")
australia = data_australia[,c(2,5)]
head(australia)##Visualising time series data
ggplot(australia,aes(x=Date,y=PercentOfBaseline)) + geom_line()+
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  theme_classic()australia <- australia[,-1]##Converting into time series object
australia_ts <- ts(australia,frequency = 366)#leap year and daily data
```



From this plot, there is little trend (not sure as showing its presence in mean level) but seasonality is present but lag is not clear from this picture. So, we need to go for Unit root testing for non-stationarity.

Checking for non-stationarity

The PACF plot of the given time series is as below:



The above PACF plot shows several significant values up to lag of order 8. Now, choosing a lag order for testing is a tough choice in practice. If lag order is too small then the remaining serial correlation in the errors will bias the test. If lag order is too large then the power of the test will suffer. So, taking a justifiable number of significant value for testing the unit root is sufficient. Choosing the lag order as 5(Not too big, not too small), the Dickey-Fuller test is given below,

```
adfTest(australia_ts,lags = 5,type ="c")@test
```

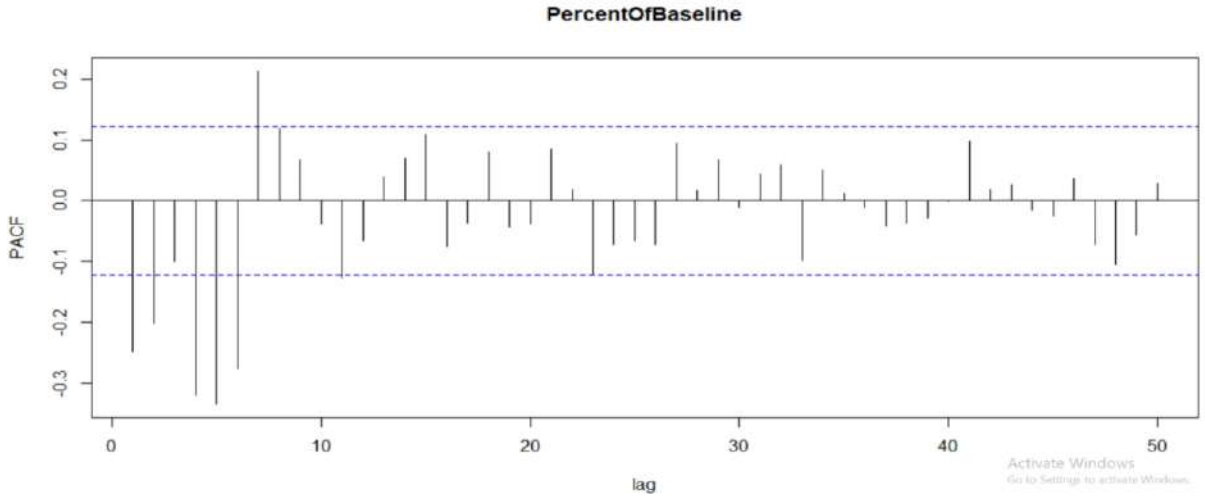
```
> adfTest(australia_ts,lags = 5,type ="c")@test
$data.name
[1] "australia_ts"

$statistic
Dickey-Fuller
    -2.899468

$p.value

0.04810496

$parameter
Lag Order
        5

$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2  y.diff.lag3  y.diff.lag4
    2.43585     -0.08585     -0.40088     -0.32594     -0.27819     -0.41621
y.diff.lag5
   -0.33174
```

The model for the above DF test

$$\Delta X_t = 2.43585 + (-0.08585)X_{t-1} + (-0.40088)\Delta X_{t-1} + (-0.32594)\Delta X_{t-2} + (-0.27819)X_{t-3} + (-0.41621)\Delta X_{t-4} + (-0.33174)\Delta X_{t-5}$$

Here, p value is very close to 0.05; indicating presence of unit root in the data. So, first differencing is required.

```
#1st differencing
australia_ts_1diff <- diff(australia_ts)
pacfPlot(australia_ts_1diff)
```



The above PACF plot also shows several significant values. So, taking a small number of significant values for testing the unit root is sufficient. Here also, choosing the lag order as 5, the Dickey-Fuller test is given below,

```
adfTest(australia_ts_1diff,lags = 5,type ="c")@test
```

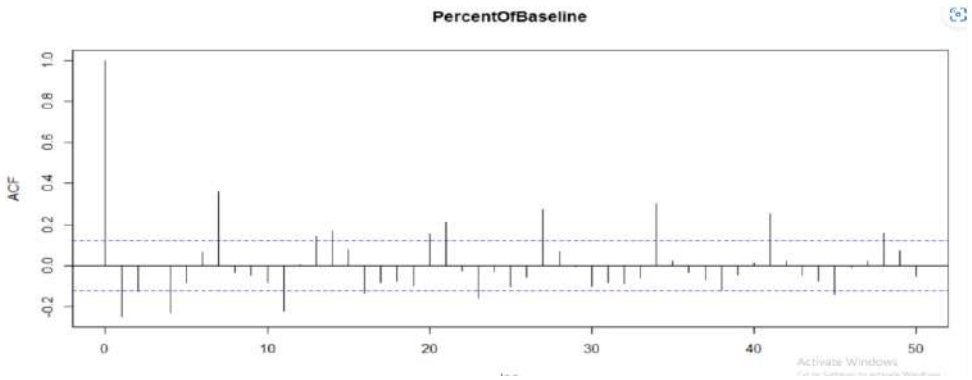
```
> adfTest(australia_ts_1diff,lags = 5,type ="c")@test
$data.name
[1] "australia_ts_1diff"

$statistic
Dickey-Fuller
     -13.9988

$p.value

0.01

$parameter
Lag Order
        5

$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2  y.diff.lag3  y.diff.lag4  y.diff.lag5
    -0.9443      -3.7100       2.1631       1.6727       1.2830       0.7425       0.2797


Warning message:
In adfTest(australia_ts_1diff, lags = 5, type = "c") :
  p-value smaller than printed p-value
```

The model for the above Dickey-Fuller test is(Taking the first differenced series as X*t),

$$\Delta^* X_t = -0.9443 + (-3.7100)X_{t-1} + (2.1631)\Delta X_{t-1} + (1.6727)\Delta X_{t-2} + (1.2830)X_{t-3} + (0.7425)\Delta X_{t-4} + (0.2797)\Delta X_{t-5}$$

Now, this test suggests there is no more unit root is present in the model. So, the number of differencing required to get stationarity is 1(d=1).

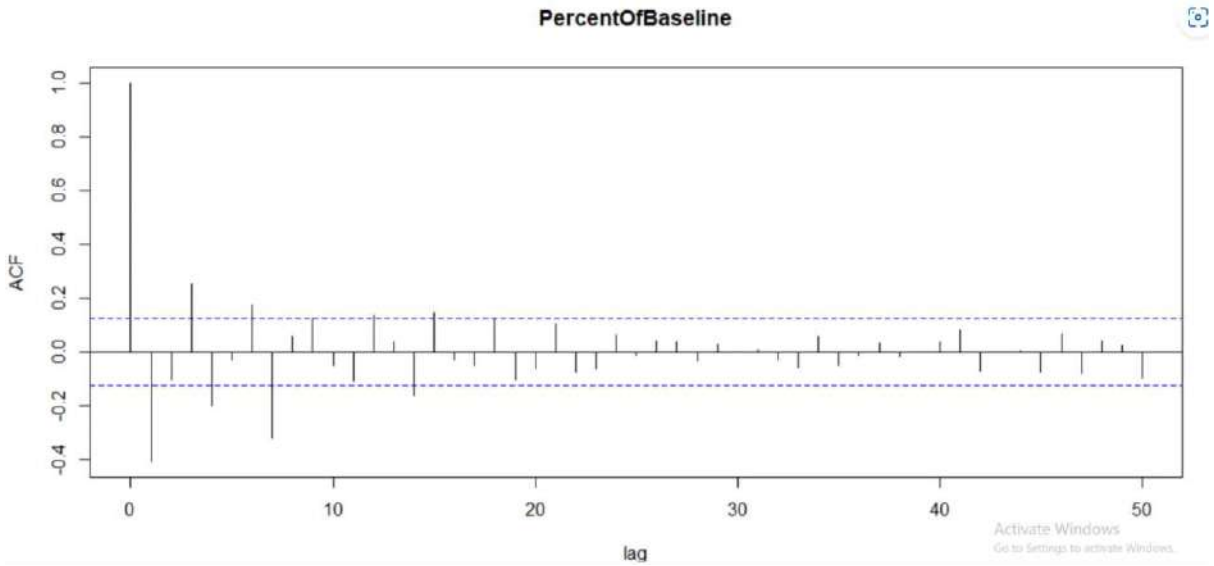
Now, to check for the seasonality, need to plot ACF of the first differenced series. Since first differenced series is mean stationary, it only contains the seasonal component. The plots shown below



From the above ACF Plot, it is clear that their weekly seasonality is present in the data. At an interval of 7 days, there is a presence of highly significant value. Which indicates weekly

seasonality. So, at least one seasonal differencing is required. Now, the ACF plot of seasonal differenced series of the first differenced series is shown below,

```
seas_diff_aus <- diff(australia_ts_1diff,7)
acfPlot(seas_diff_aus,lag.max=50)
```



Now, the above ACF Plot shows that, there is now very less seasonality present in the data and the series becomes stationary. So, the number of seasonal differencing required is at least 1(D=1).

So, after analysing all the previous graphs and tests, a SARIMA model of order (4, 1, 2) × (2, 1, 2)$_7$ model would be a good choice for the given time series data. The fitted model is shown below,

```
#Model fitting
sarima_fit_aus = arima(seas_diff_aus,order = c(4,1,2),seasonal = list(order=c(2,1,2),period=7))
sarima_fit_aus
```

So, the estimated SARIMA (4, 1, 2) × (2, 1, 2)$_7$ model is,

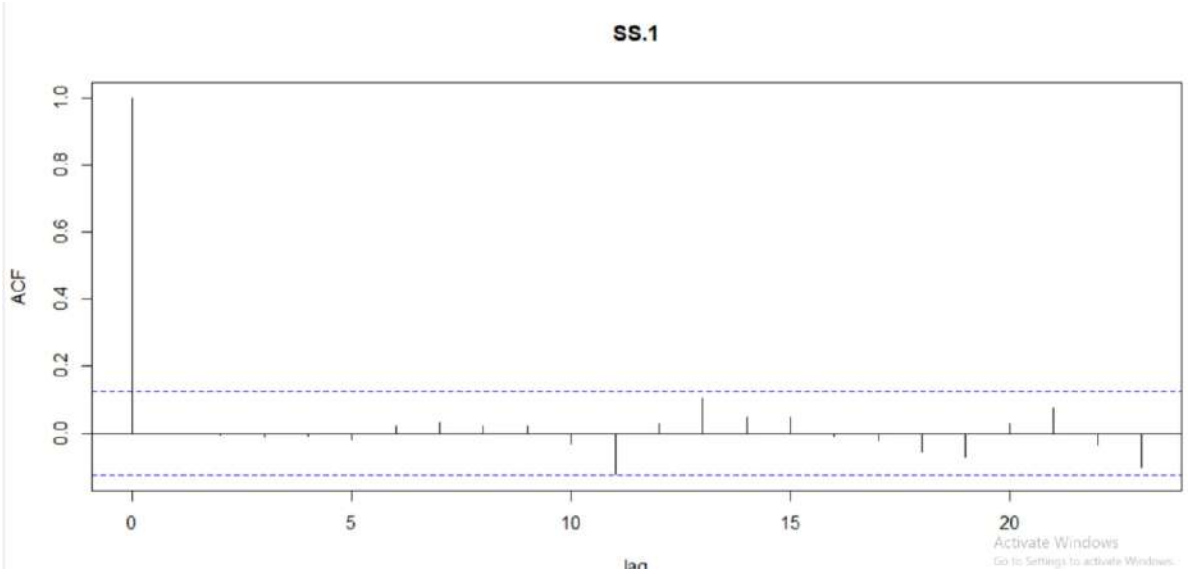
$$(1-0.3321B-0.0946B^2-0.1830B^3+-0.1651B^4)(1-0.3923B^7-0.0181B^{14})(1-B^7)(1-B)X_t = (1-1.8627B+0.8683B^2)(1-1.9573B^7+0.9576B^{14})\varepsilon_t$$

; where B is the Backward shift Eliminator.

Diagnostic Checking

After fitting the data, need to do diagnostic checks for the model, which is done below,

```
sarima_fit_aus_res = residuals(sarima_fit_aus)
acfPlot(sarima_fit_aus_res)
```



Clearly, the ACF of the residuals resembles the pattern of white noise process, so the fitted model can be assumed to be a good fitted model (as no significant value except at lag 0).

- ***Portmanteau Test***

Portmanteau test statistic tests for the joint hypothesis that all the residuals up to a certain lags are simultaneously equal to zero. Tests for the the fitted sarima model is done as follows,

```
BoxPierce(sarima_fit_aus_res,lag=50)
```

```
lags statistic df   p-value
  50  38.04294 50 0.8923018
```

In the test taking up to lag 50, we can't reject the null hypothesis that the residuals are random (Based on the P-value).

- ***Ljung-Box Test:***

Similar to the previous test, Ljung-Box test statistic also tests for the joint hypothesis that all the residuals up to certain lags are simultaneously equal to zero. Tests for the fitted sarima model is done as follows,

```
LjungBox(sarima_fit_aus_res,lag=50)
```

```
lags statistic df   p-value
  50  43.78686 50 0.7195983
```

In the test taking up to lag 50, we can't reject the null hypothesis that the residuals are random (Based on the P-value).

So, all the diagnostic checks for the fitted SARIMA model or order (4, 1, 2) × (2, 1, 2)7 implies this as a good fitted model for the given data.

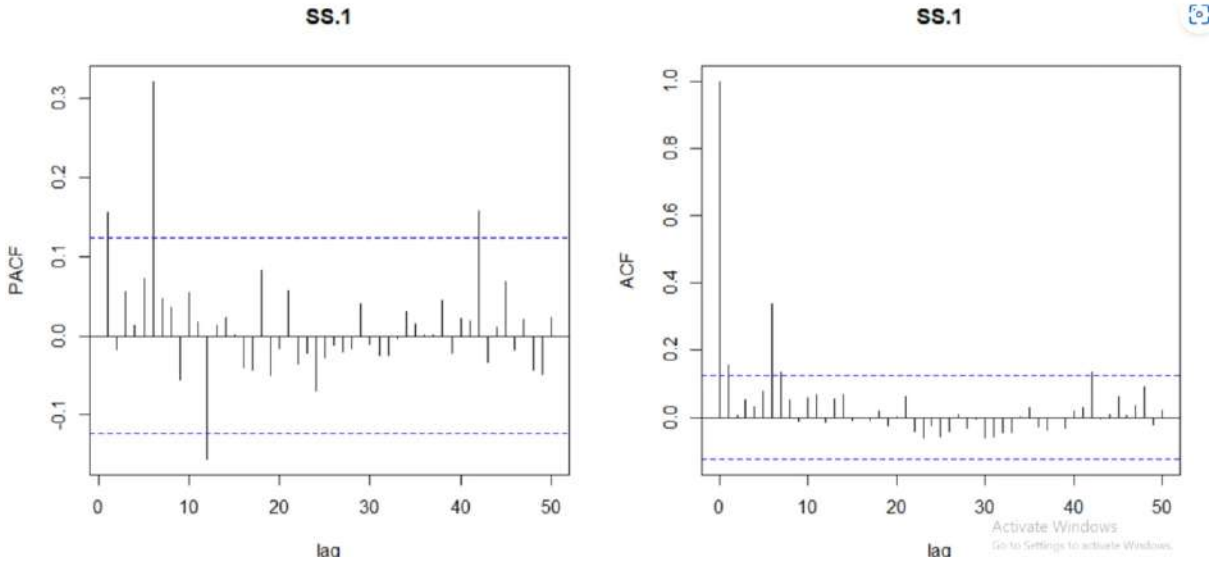
Checking for Volatility

The residuals square of the above fitted model is obtained as follows,

```
res= sarima_fit_aus_res
res_sq = res^2
res_sq_ts = ts(res_sq,frequency = 1)
```

Now, to check for the order of an GARCH model, we need PACF Plot of the residual squares, which is shown below,

```
par(mfrow=c(1,2))
pacfPlot(res_sq_ts,lag.max=50)
acfPlot(res_sq_ts,lag.max=50)
```



There is small highly significant values present in the PACF Plot of the residual squares. But some values are larger than the remaining ones. And, similarly in the ACF plot also, there is small significant values present.

Taking a small lag order 1, we fit a regression model on the residual square time series. Now, the testing for the presence of ARCH effect is done using

Lagrange Multiplier Test of Engle (1982). This uses the linear regression

$$X_t^2 = \alpha_o + \alpha_1 X_{t-1}^2 + ... + \alpha_p X_{t-p}^2 + u_t \ (the\ unknown\ \sigma_t^2\ is\ replaced\ by\ X_t^2)$$
$$Here\ we\ test\ H_o : \alpha_1 = \alpha_2 = ... = \alpha_p = 0.\ If\ rejected\ there\ is\ volatility.$$

Taking the lag order, s=6(Since, upto lag 6, the values are larger than the remaining ones), the testing is done below,

```
Lm.test(res_sq_ts,lag.max = 1,alpha = 0.05)
```

Now, the p-value of the above test is <2.2e-16, which clearly suggests that we reject the null hypothesis and conclude that, ARCH effect is present in the data.

How to fit model to capture volatility?

There is no certain way to decide best (G)ARCH model for the conditional variance. You will find it by trial and error.

Rugarch package

The rugarch package aims to provide for a comprehensive set of methods for modelling univariate GARCH processes, including fitting, filtering, forecasting, simulation as well as diagnostic tools including plots and various tests. Additional methods such as rolling estimation, bootstrap forecasting and simulated parameter density to evaluate model uncertainty provide a rich environment for the modelling of these processes.
In this function, you need to start by specifiying your model. This package seeks and finds the most appropriate GARCH model for your series. To do so, use ugarchspec() function.

Now, by looking at ACF and PACF, we will specify the order of GARCH model manually. The orders are (1,2).

```
spec=ugarchspec(mean.model = list(armaOrder=c(4,2),
                        seasonal=list(order=c(2,1,2),period=7)),variance.model =
list(model="sGARCH",garchOrder = c(1, 2)),
            distribution.model = "std")def.fit1= ugarchfit(spec = spec, data = seas_diff_aus)
print(def.fit1)
```

Then, some information criteria are given to be used for selecting the best GARCH models for the series. The model having smallest information criteria is the better than the others.

Ljung Box Tests are used to test serial autocorrelation among the residuals. (Null: No autocorrelation)
The results show that residuals have autocorrelation, but squared residuals not. (Look p values.)

ARCH LM test is used to check presence of ARCH effect. (Null: Adequately fitted ARCH process)
The results show that the GARCH process is adequately fitted. (Look p values.)
Sign Bias Test is used to test leverage effect in the standardized residuals. (Null: no significant negative and positive reaction shocks (if exist an ARCH type models))

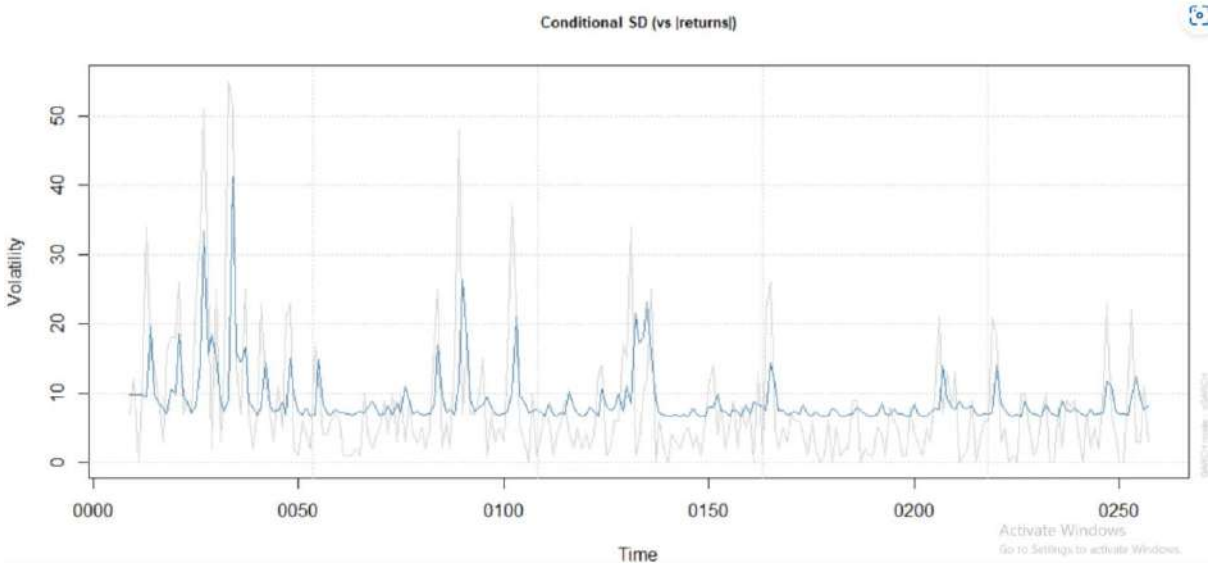
The results show that there is a leverage effect. Fit an ARCH type model.

The Nyblom stability test provides a means of testing for structural change within a time series. A structural change implies that the relationship between variables changes overtime e.g. for the regression **y=βxy=βx** beta changes over time. (Null: the parameter values are constant i.e. zero variance, the alternative hypothesis is that their variance > 0.) (Reject Ho, if Test Stat > CV.)
Therefore, we can say that omega, alpha and beta have stability problem. We should also consider TGARCH models.

Adjusted Pearson Goodness-of-Fit Test calculates the chi-squared goodness of fit test, which compares the empirical distribution of the standardized residuals with the theoretical ones from the chosen density. In this case, the chosen density is student t, not

standard normal. However, this is not a problem because t-distribution approaches the normal distribution for n>30.

It is seen that we have a normality problem.
I have tried several orders of GARCH model, but looking at ACF, PACF of square of residuals, the proper order should be p=1, q=2. Other orders like p=1, q=1 or p=2, q=1 for this models AIC, BIC are larger than AIC of models with p=1, q=2.
Here, AIC = 7.0244, BIC = 7.1939

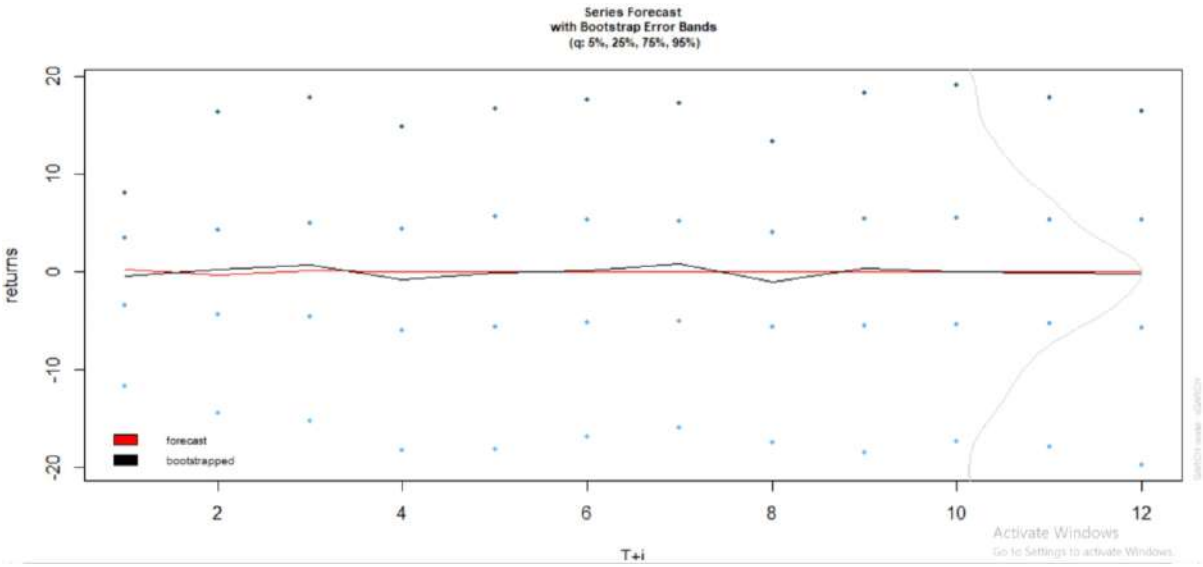


Grey line is the plot of the series. Blue line represents the volatility of the series. By looking at this plot, you can see the periods in which volatility was high.

Forecasting

There are two forecasting options in this package. Rolling and Bootstrap method.
The bootstrap methods are based on resampling standardized residuals from the empirical distribution of the fitted model to generate future realizations of the series and sigma.
In other words, with bootstrap forecasting, you can forecast both series and conditional variances.

```
bootp=ugarchboot(def.fit1,method=c("Partial","Full")[1],n.ahead =
12,n.bootpred=1000,n.bootfit=1000)
bootp
plot(bootp,which=2)
```



Conclusion

To capture mean functions, we checked and removed trend as well as seasonality. At the end, to predict mean function using training data, we have integrated those after modelling using SARIMA model of order $(4, 1, 2) \times (2, 1, 2)_7$. We checked for diagnostics of model and all tests, figures showed good fit except presence of volatility. We were able to predict not only mean functions but also captured inherent volatility using GARCH models within time series. We predicted percentage of traffic volume for next 12 days.

Possible research areas

- It can be investigated about the contribution of factors towards percentage of airport traffic in airports in certain country.
- How particular airports became significant for handling airport traffic unusually compared to others in same country.