COVID-19 TIME SERIES FORECASTING (India)

Objective: The objective of this project is to analyse the time-series of SARS-CoV-2/Covid-19 confirmed and death cases and study the trends to forecast expected future trends in India.

The following libraries were used:

library(dplyr) #data manipulation

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(forecast) #forecast()

## Warning: package 'forecast' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(openxlsx) #write.xlsx()

## Warning: package 'openxlsx' was built under R version 4.0.3

library(ggplot2)

Raw data is imported and manipulated as per requirements. It is ensured that the datatype in the imported file is interpreted exactly as what it was before, or at least what we want it to be interpreted as.

##working with data  
#importing data  
data.world<-read.csv("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv",sep = ",",header = T)  
str(data.world)

## 'data.frame': 93740 obs. of 59 variables:  
## $ iso\_code : chr "AFG" "AFG" "AFG" "AFG" ...  
## $ continent : chr "Asia" "Asia" "Asia" "Asia" ...  
## $ location : chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ date : chr "2020-02-24" "2020-02-25" "2020-02-26" "2020-02-27" ...  
## $ total\_cases : num 1 1 1 1 1 1 1 1 2 4 ...  
## $ new\_cases : num 1 0 0 0 0 0 0 0 1 2 ...  
## $ new\_cases\_smoothed : num NA NA NA NA NA 0.143 0.143 0 0.143 0.429 ...  
## $ total\_deaths : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_smoothed : num NA NA NA NA NA 0 0 0 0 0 ...  
## $ total\_cases\_per\_million : num 0.026 0.026 0.026 0.026 0.026 0.026 0.026 0.026 0.051 0.103 ...  
## $ new\_cases\_per\_million : num 0.026 0 0 0 0 0 0 0 0.026 0.051 ...  
## $ new\_cases\_smoothed\_per\_million : num NA NA NA NA NA 0.004 0.004 0 0.004 0.011 ...  
## $ total\_deaths\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_deaths\_smoothed\_per\_million : num NA NA NA NA NA 0 0 0 0 0 ...  
## $ reproduction\_rate : num NA NA NA NA NA NA NA NA NA NA ...  
## $ icu\_patients : num NA NA NA NA NA NA NA NA NA NA ...  
## $ icu\_patients\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ hosp\_patients : num NA NA NA NA NA NA NA NA NA NA ...  
## $ hosp\_patients\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_icu\_admissions : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_icu\_admissions\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_hosp\_admissions : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekly\_hosp\_admissions\_per\_million : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_tests : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_tests\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_smoothed : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_tests\_smoothed\_per\_thousand : num NA NA NA NA NA NA NA NA NA NA ...  
## $ positive\_rate : num NA NA NA NA NA NA NA NA NA NA ...  
## $ tests\_per\_case : num NA NA NA NA NA NA NA NA NA NA ...  
## $ tests\_units : chr "" "" "" "" ...  
## $ total\_vaccinations : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_vaccinated : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_fully\_vaccinated : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations\_smoothed : num NA NA NA NA NA NA NA NA NA NA ...  
## $ total\_vaccinations\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_vaccinated\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ people\_fully\_vaccinated\_per\_hundred : num NA NA NA NA NA NA NA NA NA NA ...  
## $ new\_vaccinations\_smoothed\_per\_million: num NA NA NA NA NA NA NA NA NA NA ...  
## $ stringency\_index : num 8.33 8.33 8.33 8.33 8.33 ...  
## $ population : num 38928341 38928341 38928341 38928341 38928341 ...  
## $ population\_density : num 54.4 54.4 54.4 54.4 54.4 ...  
## $ median\_age : num 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 18.6 ...  
## $ aged\_65\_older : num 2.58 2.58 2.58 2.58 2.58 ...  
## $ aged\_70\_older : num 1.34 1.34 1.34 1.34 1.34 ...  
## $ gdp\_per\_capita : num 1804 1804 1804 1804 1804 ...  
## $ extreme\_poverty : num NA NA NA NA NA NA NA NA NA NA ...  
## $ cardiovasc\_death\_rate : num 597 597 597 597 597 ...  
## $ diabetes\_prevalence : num 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 9.59 ...  
## $ female\_smokers : num NA NA NA NA NA NA NA NA NA NA ...  
## $ male\_smokers : num NA NA NA NA NA NA NA NA NA NA ...  
## $ handwashing\_facilities : num 37.7 37.7 37.7 37.7 37.7 ...  
## $ hospital\_beds\_per\_thousand : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...  
## $ life\_expectancy : num 64.8 64.8 64.8 64.8 64.8 ...  
## $ human\_development\_index : num 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 0.511 ...

Raw data is cleaned and only the required variables are retrieved.

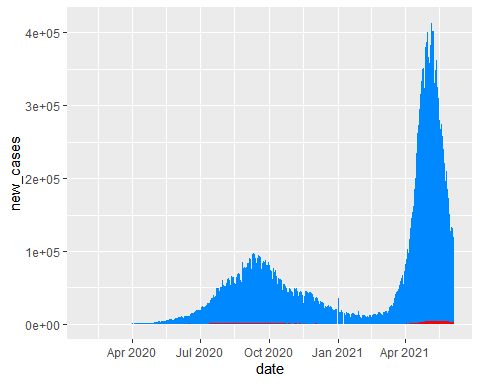
#preprocessing data  
{  
data.india<-data.world %>%  
 filter(location=="India") %>%  
 select(date,new\_cases,new\_deaths)  
data.india$date<-as.Date(data.india$date)  
n<-nrow(data.india)  
str(data.india)  
}

## 'data.frame': 493 obs. of 3 variables:  
## $ date : Date, format: "2020-01-30" "2020-01-31" ...  
## $ new\_cases : num 1 0 0 1 1 0 0 0 0 0 ...  
## $ new\_deaths: num NA NA NA NA NA NA NA NA NA NA ...

#new vector: date  
{  
start\_date = as.Date("2020-01-30", format = "%Y-%m-%d")  
end\_date = as.Date(data.india$date[n], format = "%Y-%m-%d")+15  
forecast\_date = seq.Date(from=start\_date,to=end\_date,by = 1)  
#View(forecast\_date)  
}

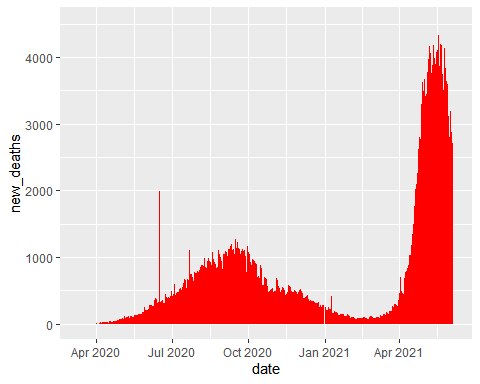
ggplot(data.india)+  
 geom\_area(aes(x=date,y=new\_cases),fill="#0088FF")+  
 geom\_area(aes(x=date,y=new\_deaths),fill="#FF0000",na.rm = TRUE)

## Warning: Removed 41 rows containing missing values (position\_stack).



ggplot(data=data.india)+  
 geom\_area(aes(x=date,y=new\_deaths),fill="#FF0000",na.rm = TRUE)

## Warning: Removed 41 rows containing missing values (position\_stack).



1. The first model represents ARIMA model for daily new cases.

model.1<-auto.arima(data.india$new\_cases); model.1

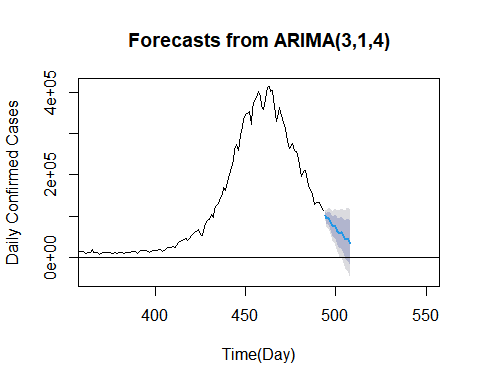
## Series: data.india$new\_cases   
## ARIMA(3,1,4)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2 ma3 ma4  
## 0.4759 -0.5008 0.9384 -0.3648 0.4571 -0.8281 0.0733  
## s.e. 0.0226 0.0219 0.0220 0.0522 0.0343 0.0350 0.0482  
##   
## sigma^2 estimated as 47107802: log likelihood=-5042.15  
## AIC=10100.3 AICc=10100.6 BIC=10133.89

model.1.forecast<-forecast(data.india$new\_cases,model = model.1,h = 15)  
head(model.1.forecast$mean)

## Time Series:  
## Start = 494   
## End = 499   
## Frequency = 1   
## [1] 102752.46 94384.96 95358.08 88545.12 76963.36 75777.48

{   
plot(model.1.forecast,xlab="Time(Day)",ylab="Daily Confirmed Cases",xlim=c(365,550),  
 main = cat("Forecasted New Cases till ",end\_date))  
abline(h=0)  
}

## Forecasted New Cases till 18798

 The model suggested is ARIMA(3,1,4); 3 parameter for Autoregression, 1 parameter for difference and 4 parameters for Moving Average.

1. The second model represents ARIMA model for daily new deaths.

model.2<-auto.arima(data.india$new\_deaths); model.2

## Series: data.india$new\_deaths   
## ARIMA(1,1,2)   
##   
## Coefficients:  
## ar1 ma1 ma2  
## 0.9391 -1.5916 0.6879  
## s.e. 0.0262 0.0387 0.0377  
##   
## sigma^2 estimated as 21562: log likelihood=-2889.25  
## AIC=5786.5 AICc=5786.59 BIC=5802.94

model.2.forecast<-forecast(data.india$new\_deaths,model = model.2,h = 15)  
head(model.2.forecast$mean)

## Time Series:  
## Start = 494   
## End = 499   
## Frequency = 1   
## [1] 2708.569 2648.853 2592.773 2540.109 2490.651 2444.206

{   
plot(model.2.forecast,xlab="Time(Day)",ylab="Daily Deaths",xlim=c(365,550),  
 main = cat("Forecasted New Deaths till ",end\_date))  
abline(h=0)  
}

## Forecasted New Deaths till 18798

