Assignment 7

Dipesh Poudel

12/28/2021

Instructions:

Use the attached Nepal COVID-19 data extracted from Wikipedia to fit the following models with daily deaths as dependent variable and time as independent variable.

First plot the daily deaths by time and distribute the three outliers (added deaths around timeline of 400) before fitting the following models in the outlier adjusted data on training and testing datasets:

Loading the excel data

```
library(readxl)
covid_tbl<-read_excel('covid_tbl_final.xlsx')

str(covid_tbl)
```

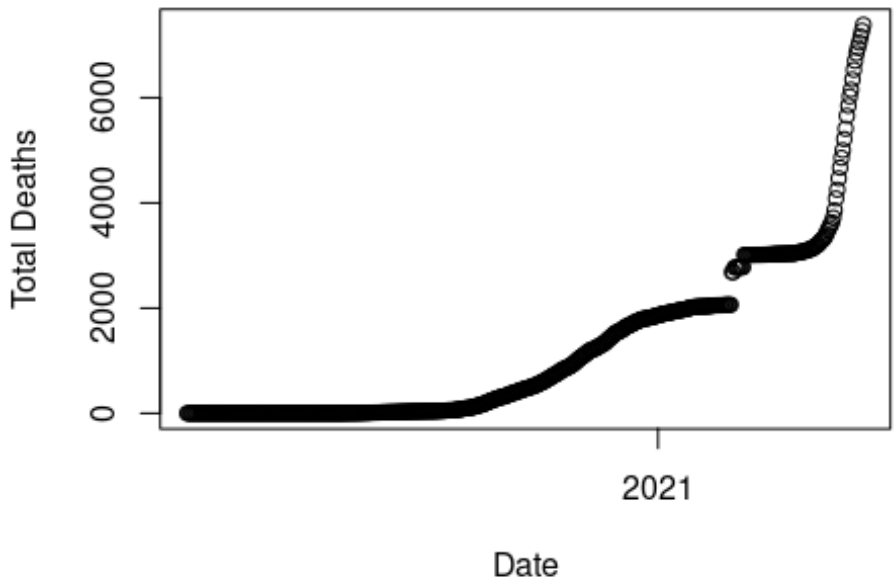
```
## tibble [495 × 14] (S3: tbl_df/tbl/data.frame)
##  $ SN                    : num [1:495] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Date                  : POSIXct[1:495], format: "2020-01-23"
"2020-01-24" ...
##  $ Confirmed_cases_total  : num [1:495] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Confirmed_cases_new    : num [1:495] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Confirmed _cases_active: num [1:495] 1 1 1 1 1 1 0 0 0 0 ...
##  $ Recoveries_total       : num [1:495] 0 0 0 0 0 0 1 1 1 1 ...
##  $ Recoveries_daily       : num [1:495] 0 0 0 0 0 0 1 0 0 0 ...
##  $ Deaths_total           : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths_daily           : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
##  $ RT-PCR_tests_total     : num [1:495] NA NA NA NA NA 3 4 5 5
NA ...
##  $ RT-PCR_tests_daily     : num [1:495] NA NA NA NA NA NA 1 1 0
NA ...
##  $ Test_positivity_rate   : num [1:495] NA NA NA NA NA ...
##  $ Recovery_rate          : num [1:495] 0 0 0 0 0 0 100 100 100 100
...
##  $ Case_fatality_rate     : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
```

```
covid_tbl$Date<-as.Date(as.POSIXct(covid_tbl$Date))

str(covid_tbl)
```

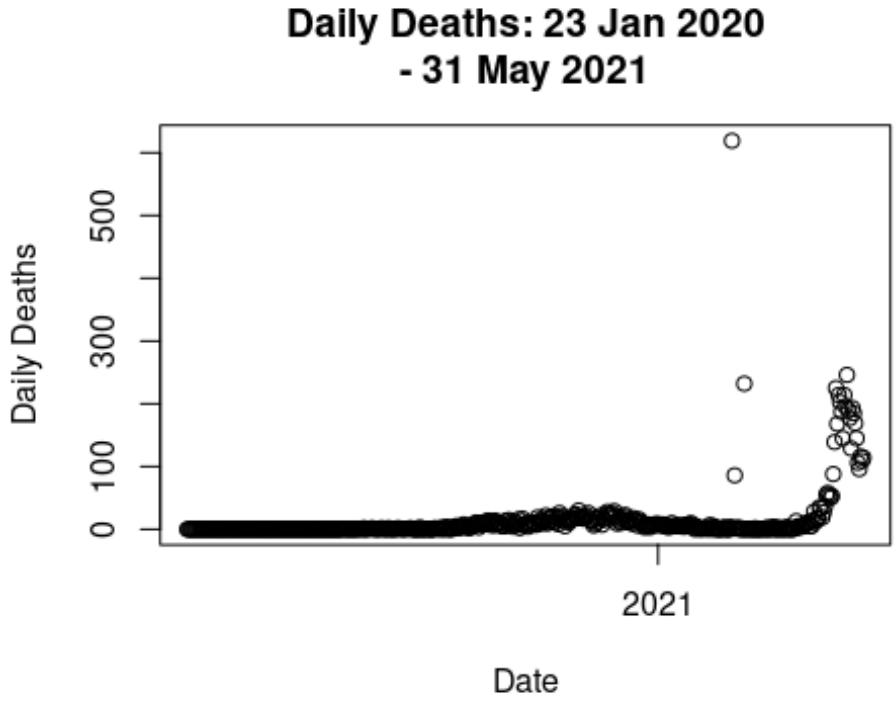
```
## tibble [495 × 14] (S3: tbl_df/tbl/data.frame)
##  $ SN                    : num [1:495] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Date                  : Date[1:495], format: "2020-01-23"
"2020-01-24" ...
```

```
##  $ Confirmed_cases_total  : num [1:495] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Confirmed_cases_new    : num [1:495] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Confirmed _cases_active: num [1:495] 1 1 1 1 1 1 0 0 0 0 ...
##  $ Recoveries_total       : num [1:495] 0 0 0 0 0 0 1 1 1 1 ...
##  $ Recoveries_daily       : num [1:495] 0 0 0 0 0 0 1 0 0 0 ...
##  $ Deaths_total           : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths_daily           : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
##  $ RT-PCR_tests_total     : num [1:495] NA NA NA NA NA 3 4 5 5
NA ...
##  $ RT-PCR_tests_daily     : num [1:495] NA NA NA NA NA NA 1 1 0
NA ...
##  $ Test_positivity_rate   : num [1:495] NA NA NA NA NA ...
##  $ Recovery_rate          : num [1:495] 0 0 0 0 0 0 100 100 100 100
...
##  $ Case_fatality_rate     : num [1:495] 0 0 0 0 0 0 0 0 0 0 ...
```

```
plot(covid_tbl$Date,covid_tbl$Deaths_total,xlab = "Date",ylab = "Total
Deaths")
```

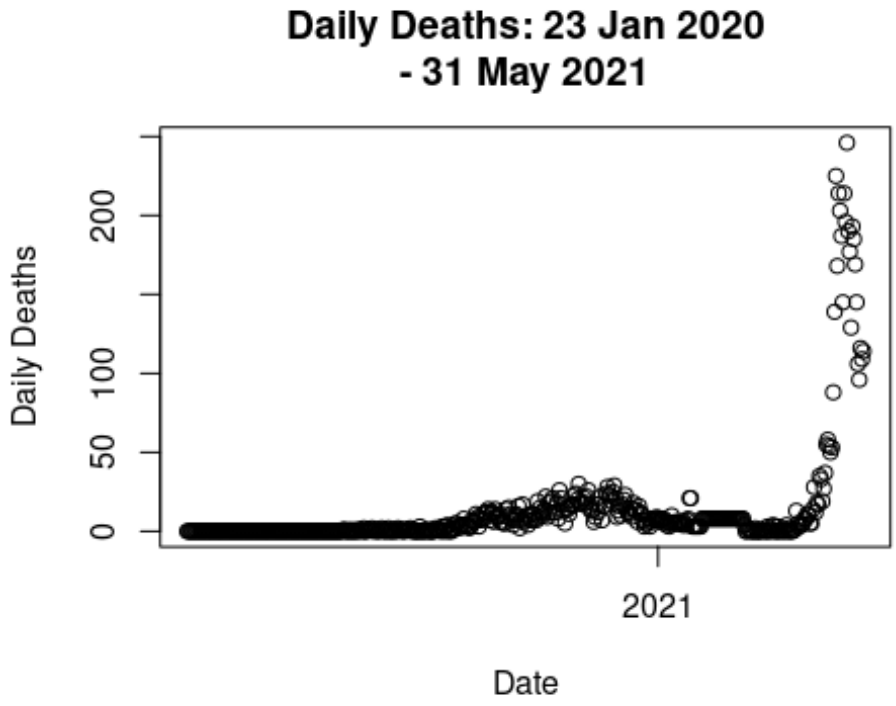



```
plot(covid_tbl$Date,
covid_tbl$Deaths_daily,
main = "Daily Deaths: 23 Jan 2020
- 31 May 2021",
xlab = "Date",
ylab = "Daily Deaths")
```




```
summary(covid_tbl$Deaths_daily)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    2.00   14.92   11.00  619.00

library(dplyr)
filter(covid_tbl,Deaths_daily>=50&Date<=as.Date("2021-03-05"))

## # A tibble: 3 × 14
##      SN Date       Confirmed_cases_total Confirmed_cases_new
`Confirmed _cases_…
##   <dbl> <date>                     <dbl>               <dbl>
<dbl>
## 1   399 2021-02-24               273760                  94
937
## 2   401 2021-02-26               273984                 112
936
## 3   408 2021-03-05               274608                 120
832
## # … with 9 more variables: Recoveries_total <dbl>, Recoveries_daily
<dbl>,
## #   Deaths_total <dbl>, Deaths_daily <dbl>, RT-PCR_tests_total
<dbl>,
## #   RT-PCR_tests_daily <dbl>, Test_positivity_rate <dbl>,
Recovery_rate <dbl>,
## #   Case_fatality_rate <dbl>
```

```r
wsn<-c(399,401,408)
for(i in 1:length(wsn)){

temp_sn = wsn[i]
# Get the Value to be adjusted
curr_val<-covid_tbl[covid_tbl$SN==temp_sn,"Deaths_daily"]
# Calculate the average daily deaths for last 30 days
avg_daily_deaths<-ceiling(mean(covid_tbl[covid_tbl$SN %in% c((temp_sn-
1):(temp_sn-1-30)),]$Deaths_daily))

# Change the Value for given SN
covid_tbl[covid_tbl$SN==temp_sn,"Deaths_daily"]=avg_daily_deaths
# Change values for last 30 days
covid_tbl[covid_tbl$SN %in% c((temp_sn-1):(temp_sn-1-30)),]
$Deaths_daily=as.integer( round(curr_val/30))
}

plot(covid_tbl$Date,
covid_tbl$Deaths_daily,
main = "Daily Deaths: 23 Jan 2020
- 31 May 2021",
xlab = "Date",
ylab = "Daily Deaths")
```



Splitting the data into training and testing set

```r
set.seed(1234)
ind<-sample(2,nrow(covid_tbl),replace=T,prob = c(0.7,0.3))
```

```
train_data<-covid_tbl[ind==1,]
test_data<-covid_tbl[ind==2,]
```

1. Linear regression model

```
library(caret)

lm1<-train(Deaths_daily~SN,data=train_data,method="lm")
predict1<-predict(lm1,newdata = test_data)

predict_eval<-function(predicted_values){
  return(data.frame(
    R2=R2(predicted_values,test_data$Deaths_daily),
RMSE = RMSE(predicted_values,test_data$Deaths_daily),
MAE = MAE(predicted_values,test_data$Deaths_daily)
  ))
}

predict_eval(predict1)

##          R2    RMSE      MAE
## 1 0.1887896 32.1613 17.61361
```

2. Quadratic linear regression model

```
lm2<-train(Deaths_daily~poly(SN,2),data=train_data,method="lm")
predict2<-predict(lm2,newdata = test_data)
predict_eval(predict2)

##          R2     RMSE      MAE
## 1 0.3143297 29.52953 18.11123
```

3. Cubic linear regression model

```
lm3<-train(Deaths_daily~poly(SN,3),data = train_data,method="lm")
predict3<-predict(lm3,newdata = test_data)
predict_eval(predict3)

##          R2    RMSE      MAE
## 1 0.4823308 25.6787 16.66555
```

4. Double quadratic linear regression model

```
lm4<-train(Deaths_daily~poly(SN,4),data = train_data,method="lm")
predict4<-predict(lm4,newdata = test_data)
predict_eval(predict4)

##          R2     RMSE      MAE
## 1 0.6857402 19.98498 14.03474
```

5. Fifth order polynomial regression model

```
lm5<-train(Deaths_daily~poly(SN,5),data = train_data,method="lm")
predict5<-predict(lm5,newdata = test_data)
predict_eval(predict5)
```

```
##          R2     RMSE      MAE
## 1 0.8005885 15.90596 8.879888
```

6. KNN regression model

```
knnmodel<-train(Deaths_daily~SN,data = train_data,method="knn")
predict6<-predict(knnmodel,newdata = test_data)
predict_eval(predict6)
```

```
##          R2     RMSE      MAE
## 1 0.9777022 5.806763 2.827703
```

7. ANN-MLP regression model with 2 hidden layers with 3 and neurons

```
library(neuralnet)
nn<-neuralnet(Deaths_daily~SN,data = train_data,hidden =
c(3,2),linear.output = F)
predict7<-predict(nn,newdata = test_data)
predict_eval(predict7)
```

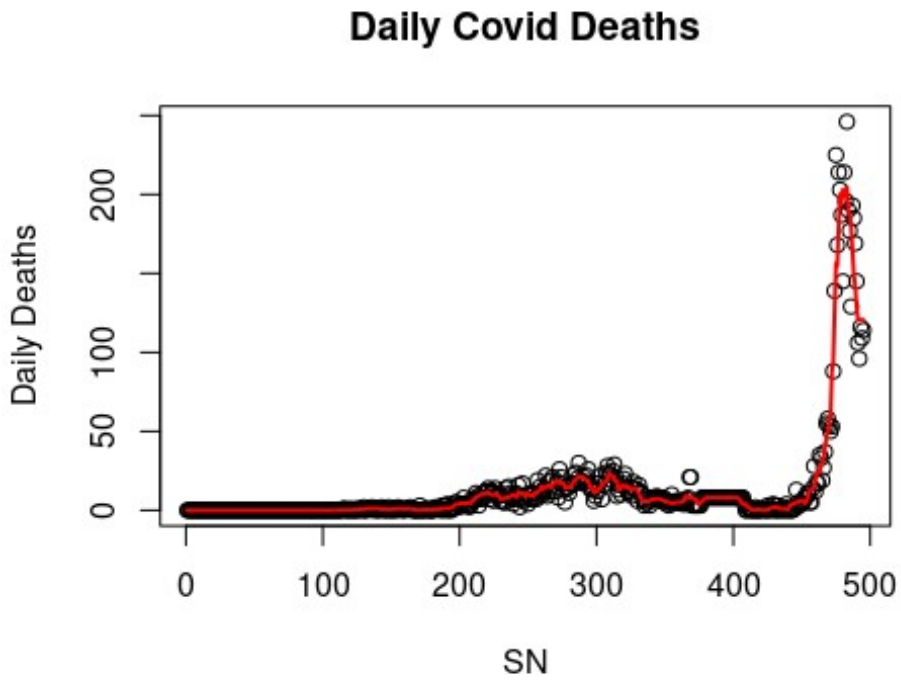
```
##           R2     RMSE      MAE
## 1 0.03179269 37.80605 13.32432
```

8. Select the best model with lowest RMSE on the test data

Based on the RMSE value on the test data the best model is KNN model which gave us the RMSE value of 5.806763 on the test data.

9. Write a summary and recommendation for Ministry of Health, Nepal

```
#Plot with linear model
plot(covid_tbl$SN, covid_tbl$Deaths_daily,
main = "Daily Covid Deaths",
xlab = "SN",
ylab = "Daily Deaths")
lines(predict(knnmodel,newdata = covid_tbl), col = "red", lwd=2)
```



The model shows that the number of deaths will increase, reach a peak and go down. So, I would recommend that the vaccine to be provided to as many people as possible as fast as possible and ease the lock down with great care.