covid_tbl.final.R

Dell

2023-06-02

```
#Shital Bhandary
#2 July 2023
#Based on 21st Lecture of Statistical Computing with R course
#Masters in Data Science program, School of Mathematical Sciences
#Tribhuvan University, Kirtipur, Nepal

covid_tbl_final <- read.csv("C:/Users/Dell/Documents/covid19_nepal/data/covid_tbl_final.csv")
str(covid_tbl_final)
```

```
## 'data.frame':    495 obs. of  14 variables:
##  $ SN                     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Date                   : chr  "23-Jan" "24-Jan" "25-Jan" "26-Jan" ...
##  $ Confirmed_cases_total  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Confirmed_cases_new    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Confirmed._cases_active: int  1 1 1 1 1 1 0 0 0 0 ...
##  $ Recoveries_total       : int  0 0 0 0 0 0 1 1 1 1 ...
##  $ Recoveries_daily       : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Deaths_total           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths_daily           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RT.PCR_tests_total     : int  NA NA NA NA NA 3 4 5 5 NA ...
##  $ RT.PCR_tests_daily     : int  NA NA NA NA NA NA 1 1 0 NA ...
##  $ Test_positivity_rate   : num  NA NA NA NA NA ...
##  $ Recovery_rate          : num  0 0 0 0 0 0 100 100 100 100 ...
##  $ Case_fatality_rate     : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
mean(covid_tbl_final$Confirmed_cases_total)
```

```
## [1] 138125.2
```

```
mean(covid_tbl_final$Confirmed_cases_new)
```

```
## [1] 1133.943
```

```
mean(covid_tbl_final$Confirmed._cases_active)
```

```
## [1] 14243.52
```

```r
mean(covid_tbl_final$Recoveries_total)
```

```
## [1] 122680.1
```

```r
mean(covid_tbl_final$Recoveries_daily)
```

```
## [1] 903.9313
```

```r
mean(covid_tbl_final$Deaths_total)
```

```
## [1] 1201.515
```

```r
mean(covid_tbl_final$Deaths_daily)
```

```
## [1] 14.92121
```

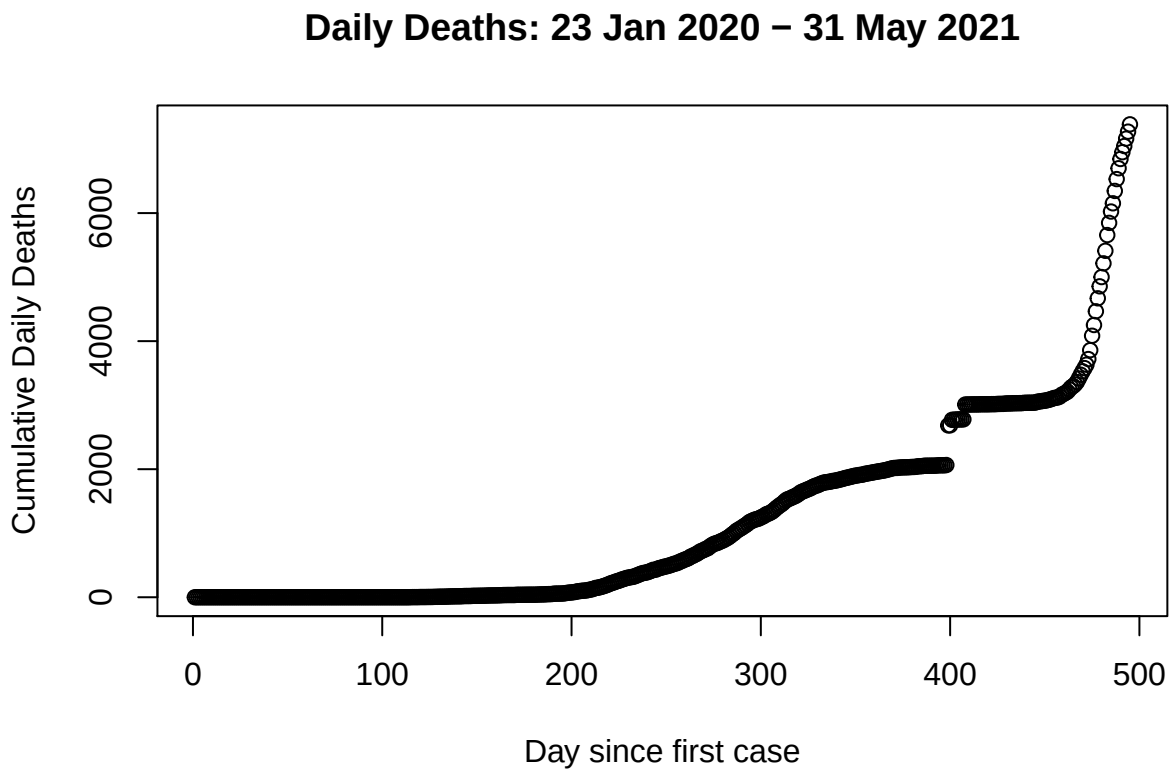
```r
mean(covid_tbl_final$RT.PCR_tests_total)
```

```
## [1] NA
```

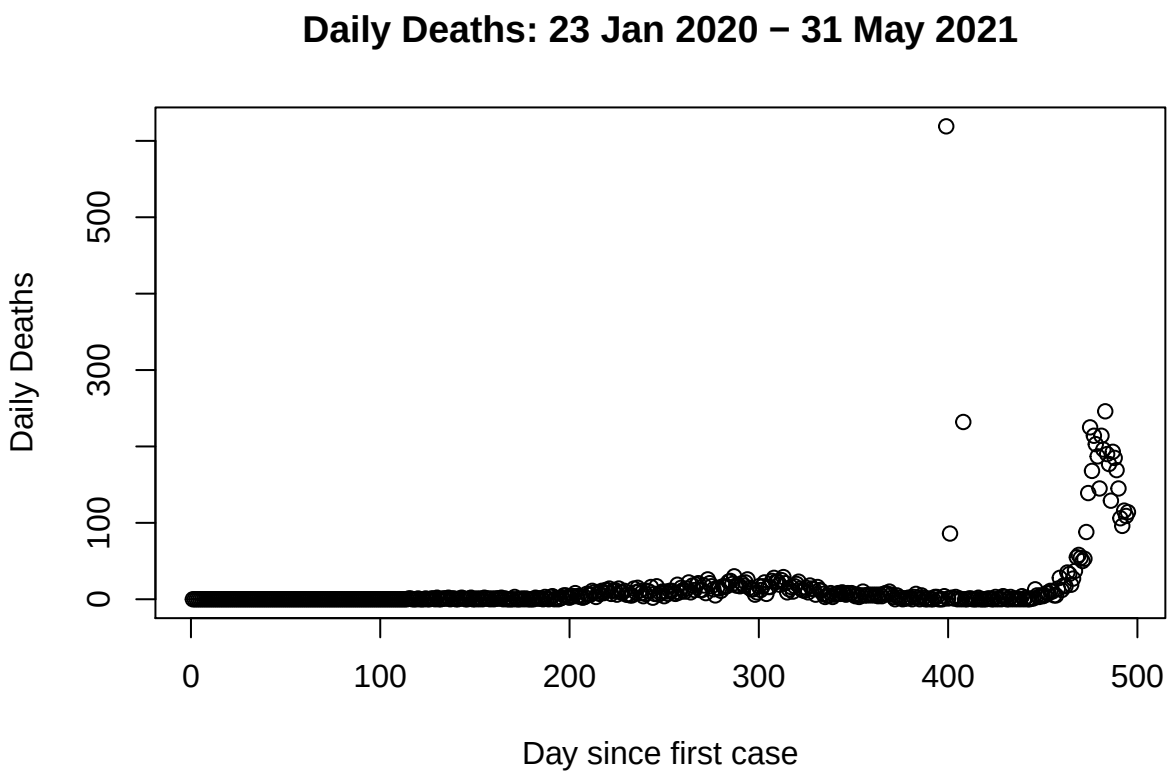
```r
mean(covid_tbl_final$RT.PCR_tests_daily)
```

```
## [1] NA
```

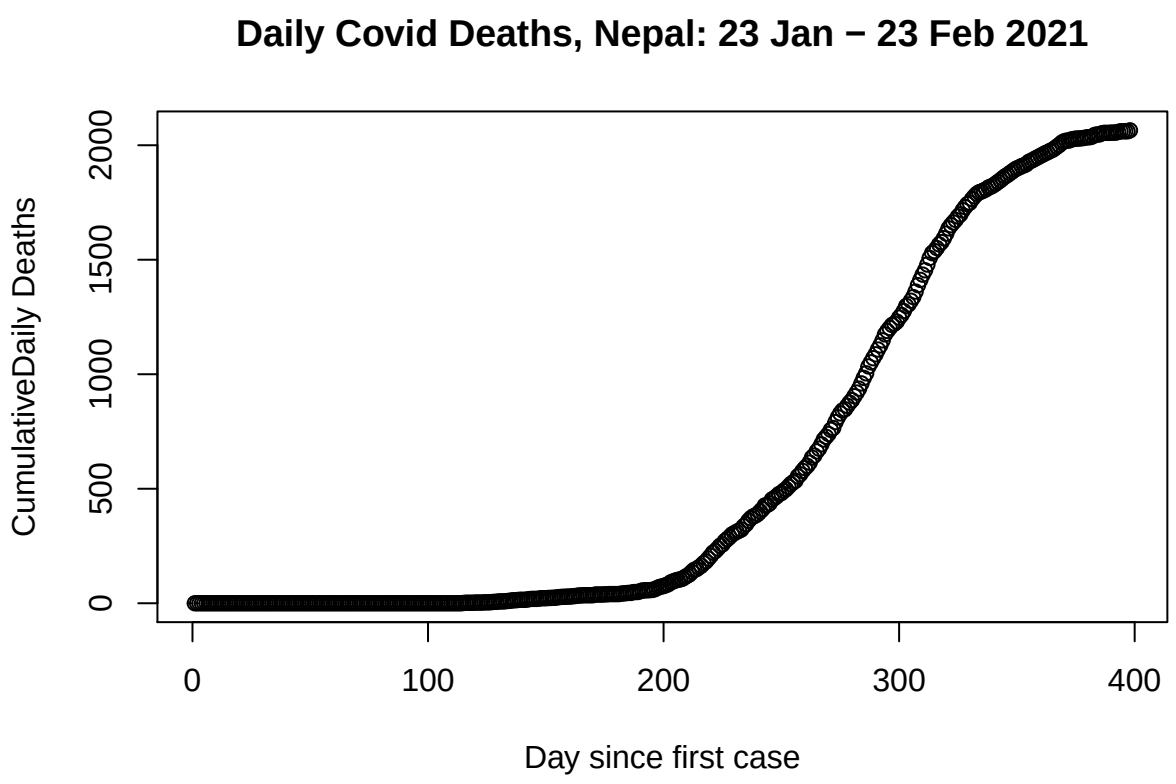
```r
#Cumulative Daily deaths
#There is a gap and we need to find why
plot(covid_tbl_final$SN, covid_tbl_final$Deaths_total,
     main = "Daily Deaths: 23 Jan 2020 - 31 May 2021",
     xlab = "Day since first case",
     ylab = "Cumulative Daily Deaths")
```




```
#Daily deaths
#There are problems as we can see three daily death values that are way more thann usual
#This happened as the death tally from MoHP and Nepali Army did not match
plot(covid_tbl_final$SN, covid_tbl_final$Deaths_daily,
     main = "Daily Deaths: 23 Jan 2020 - 31 May 2021",
     xlab = "Day since first case",
     ylab = "Daily Deaths")
```




```
#Cumulative deaths upto 398 cases i.e. 23 Feb 2021
#There are the cumulative daily deaths without adding the deaths reported by Army
plot.data <- covid_tbl_final[covid_tbl_final$SN <=398,-15]
plot(plot.data$SN, plot.data$Deaths_total,
     main = "Daily Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Day since first case",
     ylab = "CumulativeDaily Deaths")
```



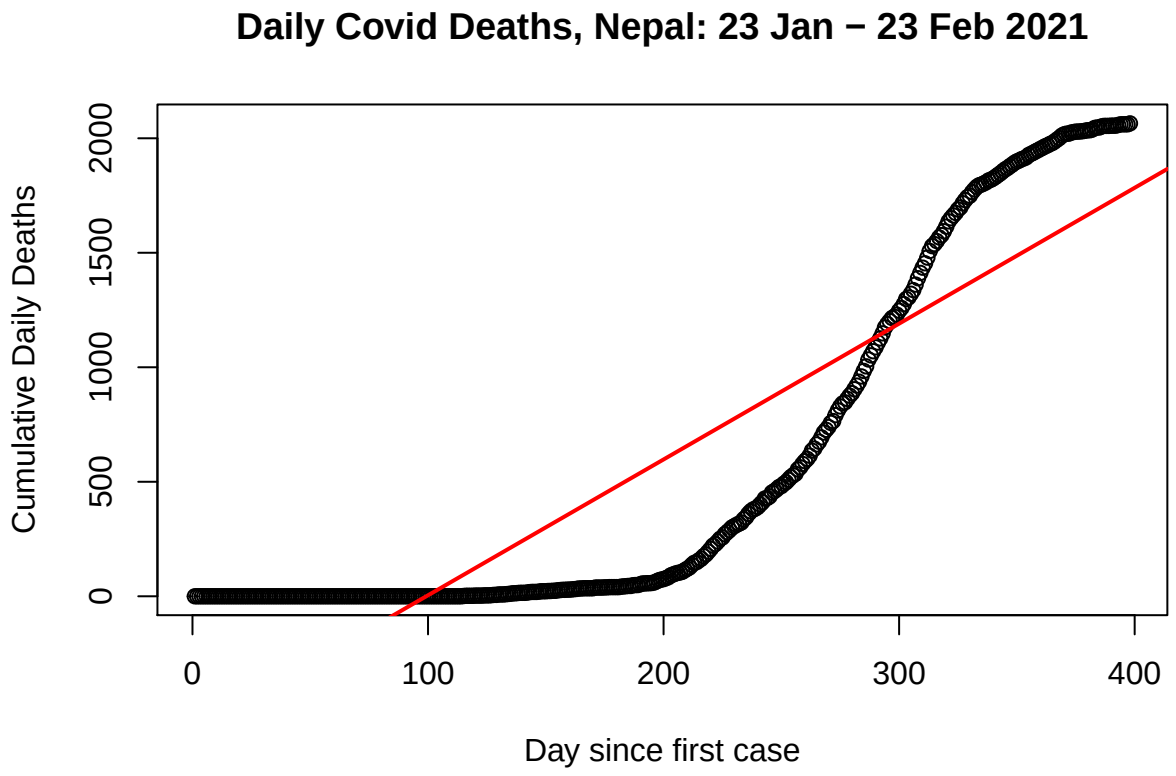

```
#We will fit polynomial models on this data now
#Linear model

#Linear model:
#R-square = 0.7921, F-test is statistically significant
#Coefficient is also statistically significant
lm <- lm(Deaths_total ~ SN, data = plot.data)
summary(lm)
```

```
##
## Call:
## lm(formula = Deaths_total ~ SN, data = plot.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -537.91 -344.76   22.38  351.50  582.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -588.8326    35.1575  -16.75   <2e-16 ***
## SN             5.9315     0.1527   38.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 350 on 396 degrees of freedom
## Multiple R-squared:  0.7921, Adjusted R-squared:  0.7916
```

```
## F-statistic:  1509 on 1 and 396 DF,  p-value: < 2.2e-16
```

```
#Plot with linear model
#Clearly shows the underfitting
plot(plot.data$SN, plot.data$Deaths_total,
     main = "Daily Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Day since first case",
     ylab = "Cumulative Daily Deaths")
abline(lm(Deaths_total ~ SN, data = plot.data), col = "red", lwd=2)
```

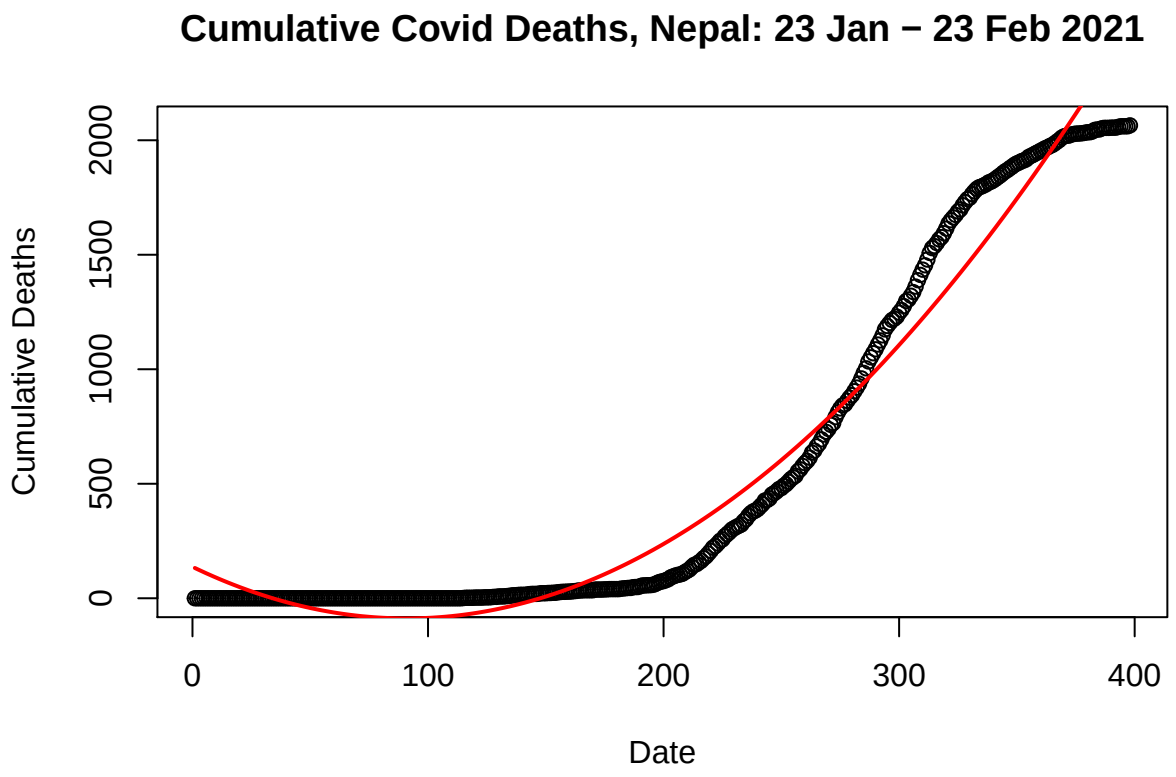



```
#Quadratic Linear model
#R-squared = 0.9692, F-test is statistically significant
#Coefficients are also statistically significant
qlm <- lm(Deaths_total ~ poly(SN, 2, raw=T), data = plot.data)
summary(qlm)
```

```
##
## Call:
## lm(formula = Deaths_total ~ poly(SN, 2, raw = T), data = plot.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -422.04 -110.87    8.94   81.97  282.94
##
## Coefficients:
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.372e+02  2.039e+01   6.727 6.11e-11 ***
## poly(SN, 2, raw = T)1 -4.959e+00  2.360e-01 -21.009  < 2e-16 ***
## poly(SN, 2, raw = T)2  2.729e-02  5.728e-04  47.646  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 134.9 on 395 degrees of freedom
## Multiple R-squared:  0.9692, Adjusted R-squared:  0.969
## F-statistic:  6211 on 2 and 395 DF,  p-value: < 2.2e-16
```

```
#Plot with quadratic linear model
#Clearly shows the good fit but can we do it better
plot(Deaths_total ~ SN, data=plot.data,
     main = "Cumulative Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Date",
     ylab = "Cumulative Deaths")
lines(fitted(qlm) ~ SN, data=plot.data, col="red", lwd=2)
```

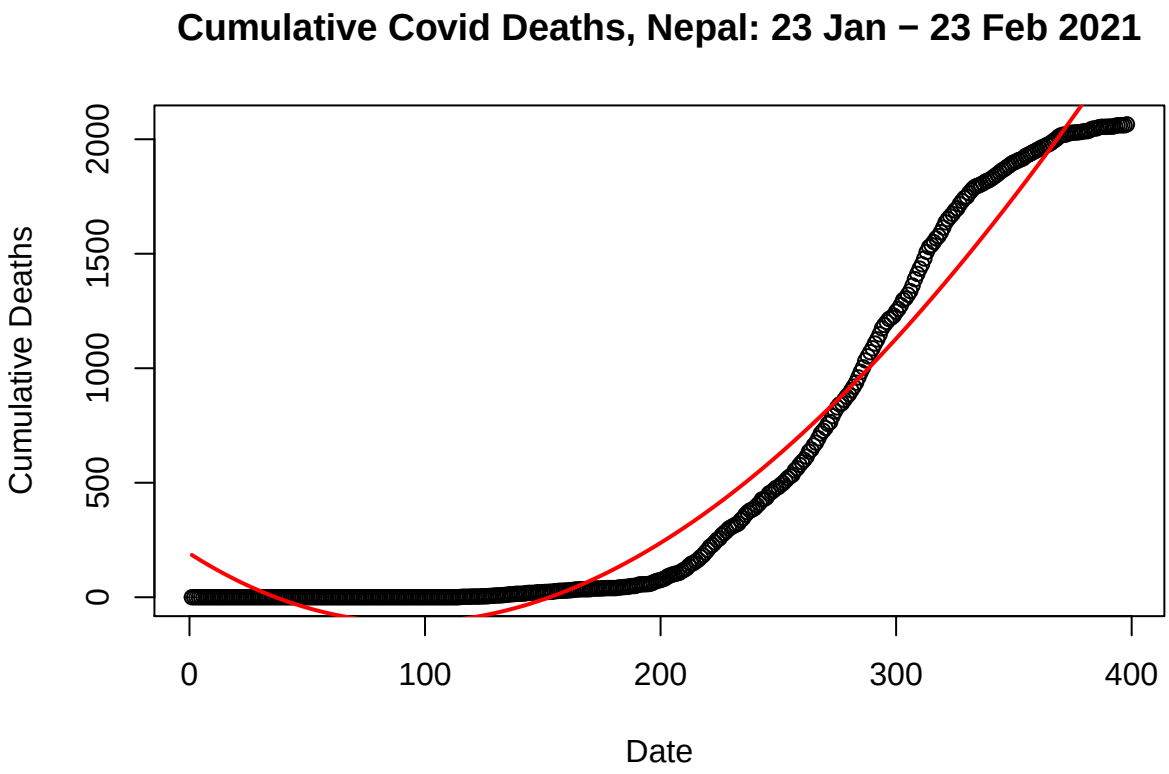



```
#Cubic Linear model
#R-squared = 0.9699, F-test is statistically significant
#Coefficients are also statistically signfincat
clm <- lm(Deaths_total ~ poly(SN, 3, raw=T), data = plot.data)
summary(clm)
```

```
##
```

```
## Call:
## lm(formula = Deaths_total ~ poly(SN, 3, raw = T), data = plot.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -369.58 -123.49   12.82   99.36  267.65
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.912e+02  2.704e+01   7.073 6.95e-12 ***
## poly(SN, 3, raw = T)1 -6.574e+00  5.861e-01 -11.217  < 2e-16 ***
## poly(SN, 3, raw = T)2  3.740e-02  3.411e-03  10.966  < 2e-16 ***
## poly(SN, 3, raw = T)3 -1.689e-05  5.620e-06  -3.006  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 133.6 on 394 degrees of freedom
## Multiple R-squared:  0.9699, Adjusted R-squared:  0.9696
## F-statistic:  4228 on 3 and 394 DF,  p-value: < 2.2e-16
```

```
#Plot with cubic linear model
#Clearly shows the good fit but can we do it better?
plot(Deaths_total ~ SN, data=plot.data,
     main = "Cumulative Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Date",
     ylab = "Cumulative Deaths")
lines(fitted(clm) ~ SN, data=plot.data, col="red", lwd=2)
```

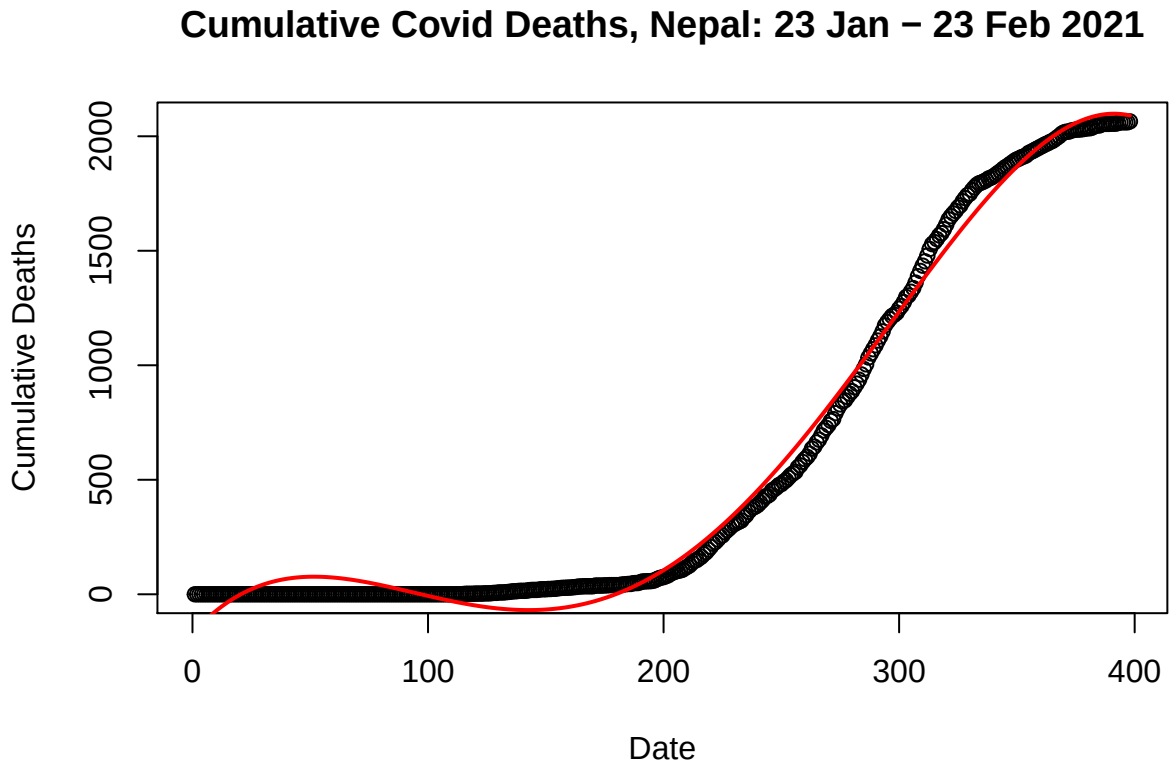



```
#Double quadratic or 4th order linear model
#R-squared = 0.9934, F-test is statistically significant
#All the coefficients are also significant
dqlm <- lm(Deaths_total ~ poly(SN, 4, raw=T), data = plot.data)
summary(dqlm)
```

```
##
## Call:
## lm(formula = Deaths_total ~ poly(SN, 4, raw = T), data = plot.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -105.44  -53.22  -12.50   53.61  159.13
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.703e+02  1.589e+01  -10.72   <2e-16 ***
## poly(SN, 4, raw = T)1  1.135e+01  5.504e-01   20.61   <2e-16 ***
## poly(SN, 4, raw = T)2 -1.641e-01  5.599e-03  -29.31   <2e-16 ***
## poly(SN, 4, raw = T)3  7.681e-04  2.107e-05   36.45   <2e-16 ***
## poly(SN, 4, raw = T)4 -9.837e-07  2.620e-08  -37.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.45 on 393 degrees of freedom
## Multiple R-squared:  0.9934, Adjusted R-squared:  0.9934
```

```
## F-statistic: 1.486e+04 on 4 and 393 DF,  p-value: < 2.2e-16
```

```
#Plot with fourth order polynomial model
#Clearly shows the good fit but can we do it better?
plot(Deaths_total ~ SN, data=plot.data,
     main = "Cumulative Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Date",
     ylab = "Cumulative Deaths")
lines(fitted(dqlm) ~ SN, data=plot.data, col="red", lwd=2)
```

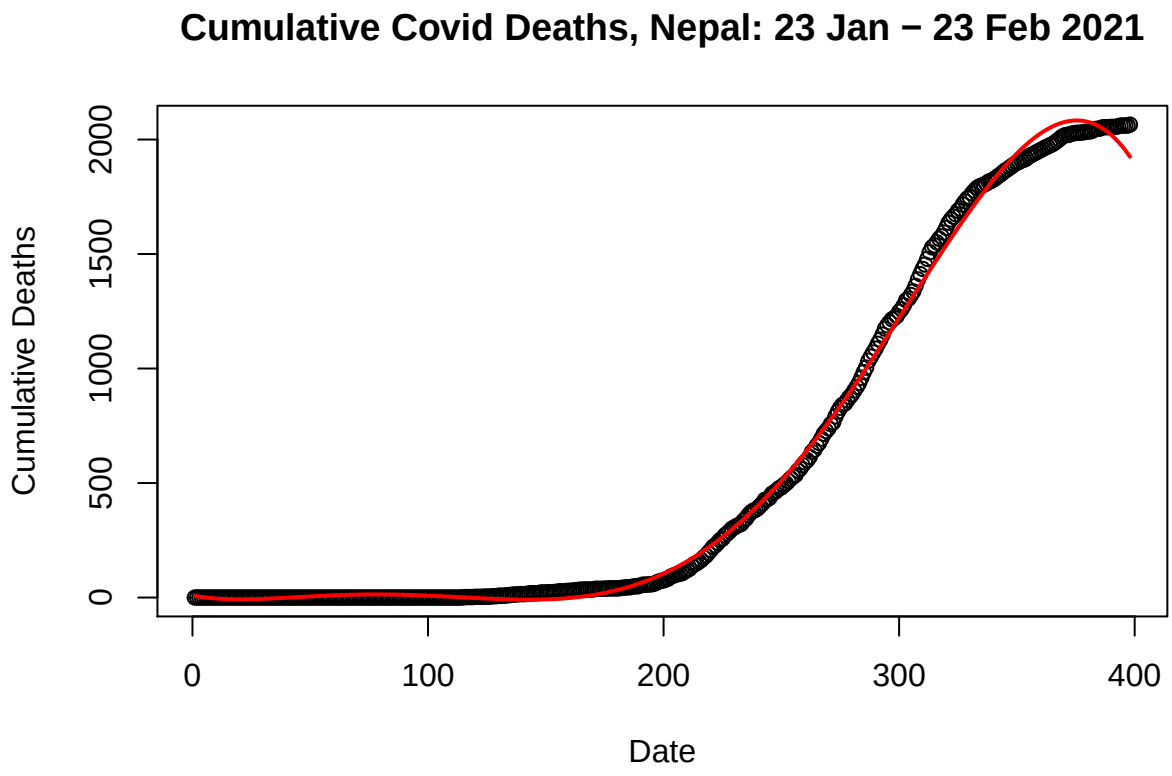



```
#Fifth order polynomial fit
#R-squared = 0.998, F-test is statistically significant
#All the coefficients are also statistically significant
folm <- lm(Deaths_total ~ poly(SN, 5, raw=T), data = plot.data)
summary(folm)
```

```
##
## Call:
## lm(formula = Deaths_total ~ poly(SN, 5, raw = T), data = plot.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.300 -16.980  -3.571  19.199 140.089
##
## Coefficients:
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.510e+00  1.053e+01   0.808 0.419427
## poly(SN, 5, raw = T)1 -1.867e+00  5.309e-01  -3.517 0.000488 ***
## poly(SN, 5, raw = T)2  6.653e-02  8.219e-03   8.095 7.31e-15 ***
## poly(SN, 5, raw = T)3 -7.705e-04  5.216e-05 -14.773  < 2e-16 ***
## poly(SN, 5, raw = T)4  3.352e-06  1.440e-07  23.270  < 2e-16 ***
## poly(SN, 5, raw = T)5 -4.346e-09  1.437e-10 -30.251  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.24 on 392 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.998
## F-statistic: 3.973e+04 on 5 and 392 DF,  p-value: < 2.2e-16
```

```r
#Plot with fourth order polynomial model
#Clearly shows the good fit but can we do it better?
#Is this a overfitting?
#Does the cumulative death declined? NO!
plot(Deaths_total ~ SN, data=plot.data,
     main = "Cumulative Covid Deaths, Nepal: 23 Jan - 23 Feb 2021",
     xlab = "Date",
     ylab = "Cumulative Deaths")
lines(fitted(folm) ~ SN, data=plot.data, col="red", lwd=2)
```



```r
#Based on the results obtained above the most likely/plausible model is the 4th order polynomial model

#Let us use the validation set approach in this model
#Creating partition variable ind with 70% and 30% probabilities
ind <- sample(2, nrow(plot.data), replace=T, prob=c(0.7,0.3))
#Creating training data with 70% cases
train.pd <- plot.data[ind==1,]
trest.pd <- plot.data[ind==2,]


#Fifth order polynomial fit on train data
#R-squared = 0.9931, F-test is statistically significant
#All the coefficients are also statistically significant
dqlm.train <- lm(Deaths_total ~ poly(SN, 4, raw=T), data = train.pd)
(dqlm.train.summary <- summary(dqlm.train))
```

```
##
## Call:
## lm(formula = Deaths_total ~ poly(SN, 4, raw = T), data = train.pd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -106.21  -54.10  -13.66   58.28  153.93
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.649e+02  1.869e+01  -8.824   <2e-16 ***
## poly(SN, 4, raw = T)1  1.112e+01  6.718e-01  16.547   <2e-16 ***
## poly(SN, 4, raw = T)2 -1.622e-01  6.843e-03 -23.701   <2e-16 ***
## poly(SN, 4, raw = T)3  7.628e-04  2.571e-05  29.671   <2e-16 ***
## poly(SN, 4, raw = T)4 -9.791e-07  3.195e-08 -30.650   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.57 on 268 degrees of freedom
## Multiple R-squared:  0.9933, Adjusted R-squared:  0.9932
## F-statistic:  9908 on 4 and 268 DF,  p-value: < 2.2e-16
```

```r
#Plot with fourth order polynomial model of train data
plot(train.pd$Deaths_total ~ train.pd$SN,
     main = "Cumulative Covid Deaths, Nepal",
     xlab = "Date",
     ylab = "Cumulative Deaths")
lines(fitted(dqlm.train) ~ SN, data=train.pd, col="red", lwd=2)

#MSE and RMSE
(MSE.dqlm.train <- mean(dqlm.train$residuals^2))
```

```
## [1] 3967.409
```

```r
(RMSE.dqlm.train <- sqrt(MSE.dqlm.train))
```

```
## [1] 62.98737
```

```
#Prediction with test data
prediction <- predict(dqlm.train, trest.pd)
library(caret)
```

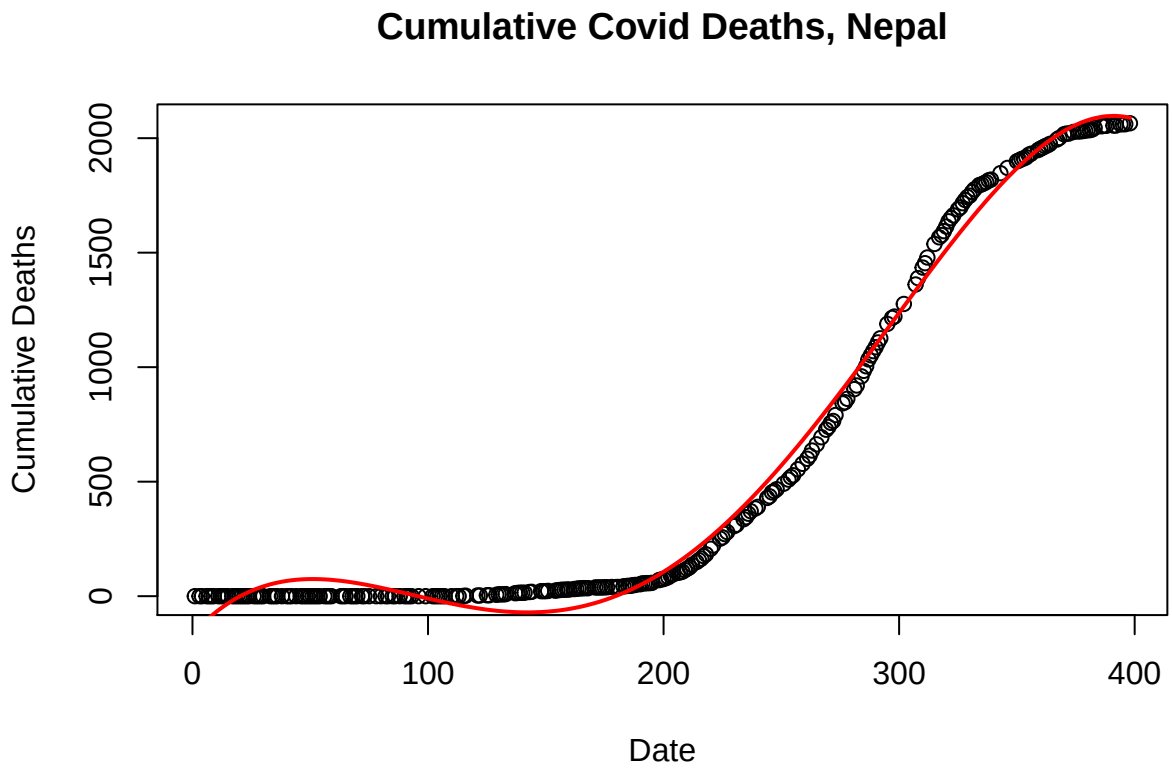
```
## Warning: package 'caret' was built under R version 4.0.5
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.0.5
```




```
(R2.test.pd <- R2(prediction, trest.pd$Deaths_total))
```

```
## [1] 0.9937432
```

```
(RMSE.test.pd <- RMSE(prediction, trest.pd$Deaths_total))
```

```
## [1] 60.11464
```

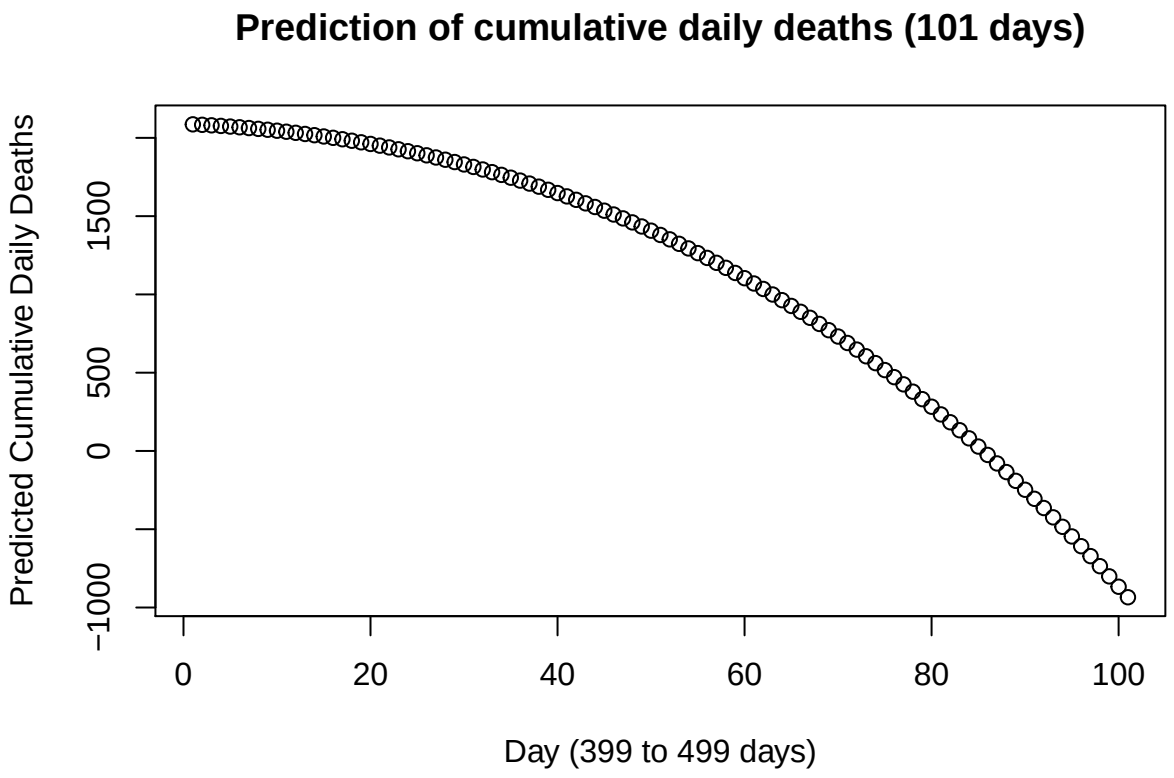
```r
#Comparing R-square and RMSE
fit_indices_train <- c(R2 = dqlm.train.summary$r.squared, RMSE = RMSE.dqlm.train)
fit_indices_test <- c(R2 = R2.test.pd, RMSE = RMSE.test.pd)
(fit_indices <- cbind(train=fit_indices_train, test=fit_indices_test))
```

```
##           train       test
## R2    0.9932829  0.9937432
## RMSE 62.9873699 60.1146424
```

```r
#Prediction for new data
new_death <- data.frame(SN=seq(399,499,1))
predicted_cd <- predict(dqlm.train, newdata = new_death)
predicted_cd
```

```
##          1          2          3          4          5          6          7
## 2086.37575 2083.40259 2080.05579 2076.33050 2072.22187 2067.72502 2062.83503
##          8          9         10         11         12         13         14
## 2057.54699 2051.85594 2045.75690 2039.24488 2032.31485 2024.96176 2017.18055
##         15         16         17         18         19         20         21
## 2008.96613 2000.31337 1991.21713 1981.67225 1971.67355 1961.21582 1950.29381
##         22         23         24         25         26         27         28
## 1938.90228 1927.03593 1914.68947 1901.85757 1888.53487 1874.71600 1860.39556
##         29         30         31         32         33         34         35
## 1845.56814 1830.22827 1814.37050 1797.98933 1781.07925 1763.63471 1745.65015
##         36         37         38         39         40         41         42
## 1727.11999 1708.03861 1688.40038 1668.19965 1647.43073 1626.08791 1604.16548
##         43         44         45         46         47         48         49
## 1581.65768 1558.55873 1534.86284 1510.56418 1485.65692 1460.13517 1433.99305
##         50         51         52         53         54         55         56
## 1407.22465 1379.82402 1351.78519 1323.10220 1293.76901 1263.77961 1233.12794
##         57         58         59         60         61         62         63
## 1201.80790 1169.81341 1137.13833 1103.77651 1069.72178 1034.96794  999.50877
##         64         65         66         67         68         69         70
##  963.33803  926.44944  888.83671  850.49354  811.41357  771.59045  731.01780
##         71         72         73         74         75         76         77
##  689.68920  647.59821  604.73839  561.10326  516.68630  471.48101  425.48081
##         78         79         80         81         82         83         84
##  378.67914  331.06941  282.64499  233.39924  183.32548  132.41704   80.66719
##         85         86         87         88         89         90         91
##   28.06920  -25.38370  -79.69830 -134.88139 -190.93981 -247.88042 -305.71010
##         92         93         94         95         96         97         98
## -364.43575 -424.06429 -484.60268 -546.05789 -608.43692 -671.74678 -735.99452
##         99        100        101
## -801.18721 -867.33194 -934.43582
```

```r
#Plot
SNN = seq(1,101,1)
plot(SNN, predicted_cd, main="Prediction of cumulative daily deaths (101 days)",
     xlab="Day (399 to 499 days)", ylab="Predicted Cumulative Daily Deaths")
```




```
#The question is "Does it make sense?"
#Did it happen like this in Nepal between 399 and 499 days since the first COVID19 case?
#Can we improve it further with cross-validation methods?
```