R regression analysis

Zhouchen Shen

2021/4/11

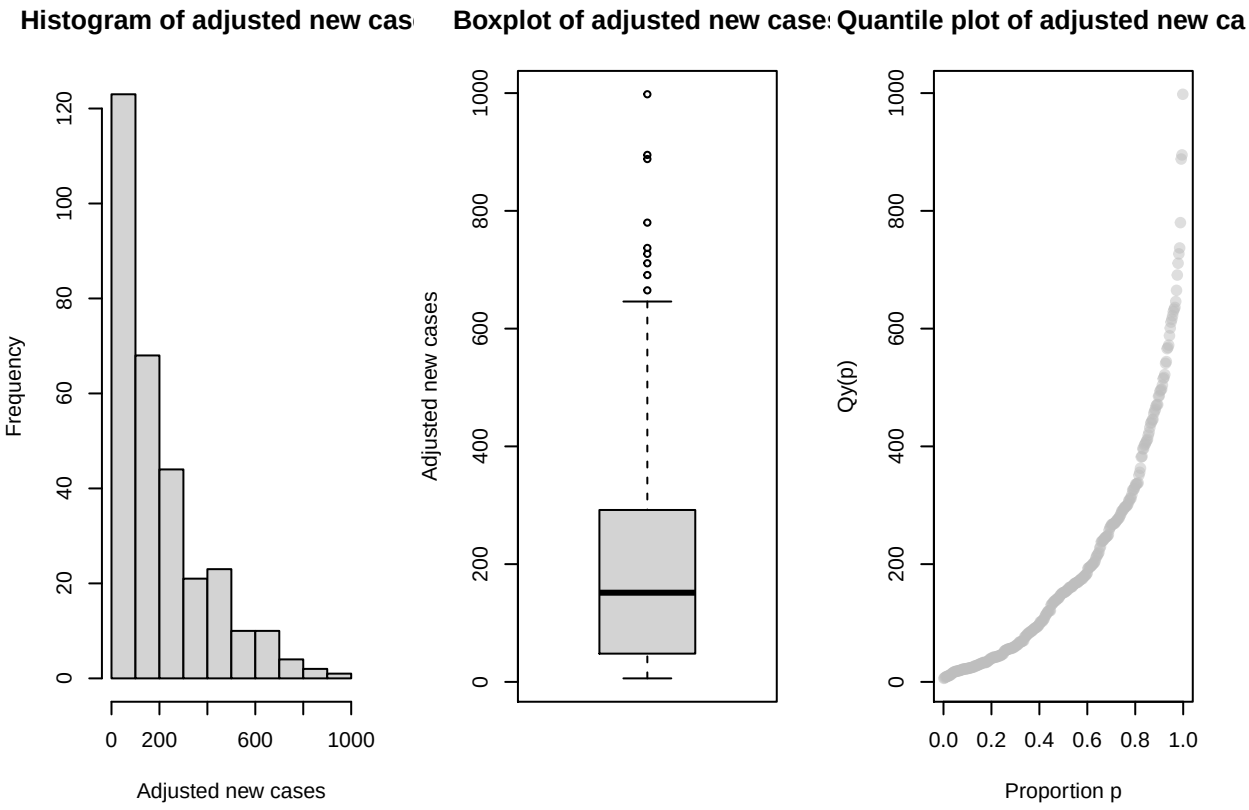
#R regression analysis ## we use the Covid-19 daily new cases of Toronto from March 2020 to December 2020. The data source can be found at https://raw.githubusercontent.com/kentranz/socialMobilityCOVID/master/data/raw/TorontoCovid.toronto.covid

```
library(RCurl)
toronto.covid <- read.csv("toronto.covid.csv", header = TRUE)
```

```
tor <- data.frame(Date = toronto.covid$Episode.Date[307:2], Day_Num = 1:306,
                  New_Cases = toronto.covid$Case.Count[307:2])

head(tor,15)
```

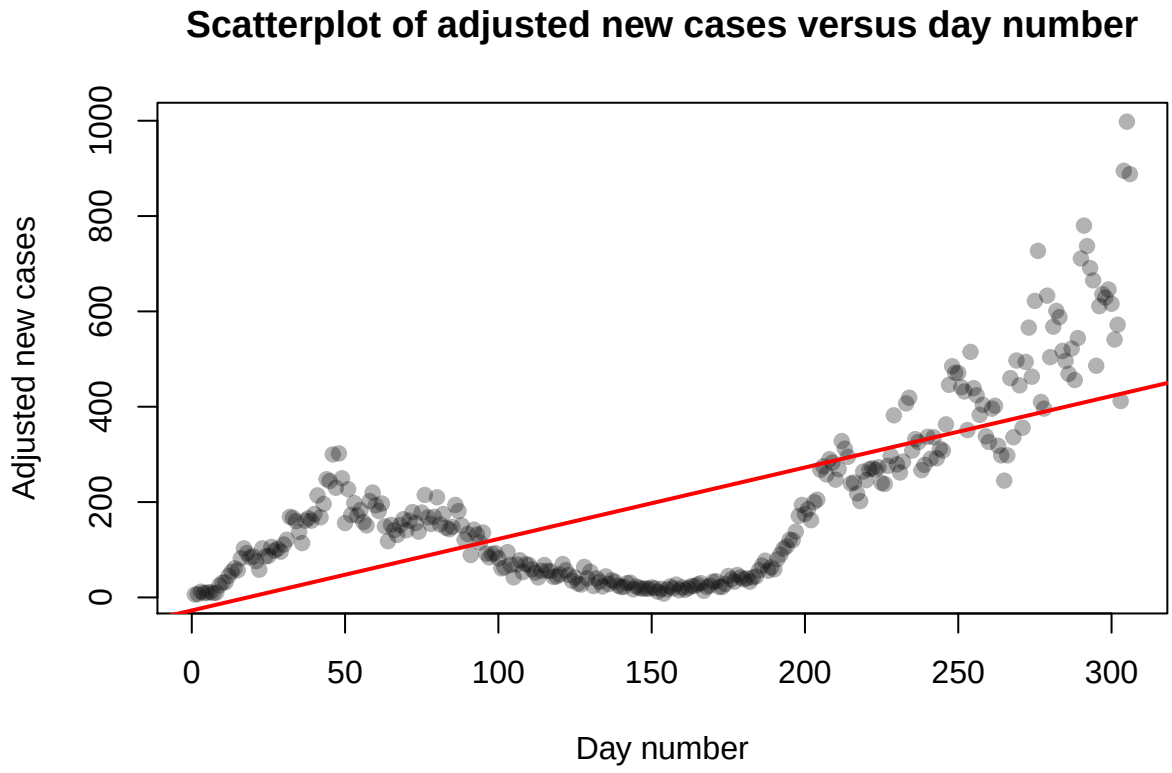
```
##       Date Day_Num New_Cases
## 1   3/1/20       1         6
## 2   3/2/20       2         7
## 3   3/3/20       3        12
## 4   3/4/20       4         9
## 5   3/5/20       5        10
## 6   3/6/20       6        11
## 7   3/7/20       7         9
## 8   3/8/20       8        10
## 9   3/9/20       9        25
## 10 3/10/20      10        31
## 11 3/11/20      11        32
## 12 3/12/20      12        45
## 13 3/13/20      13        54
## 14 3/14/20      14        61
## 15 3/15/20      15        57
```

```
par(mfrow = c(1, 3))
hist(tor$New_Cases, breaks= "Scott", xlab="Adjusted new cases", main = "Histogram of adjusted new cases

boxplot(tor$New_Cases, ylab="Adjusted new cases", main="Boxplot of adjusted new cases")

qvals <- sort(tor$New_Cases)
pvals <- ppoints(length(qvals))
plot(pvals, qvals, pch = 19, col=adjustcolor("grey", alpha = 0.5),
     xlim=c(0,1),
     xlab = "Proportion p",
     ylab = "Qy(p)",
     main = "Quantile plot of adjusted new cases")
```

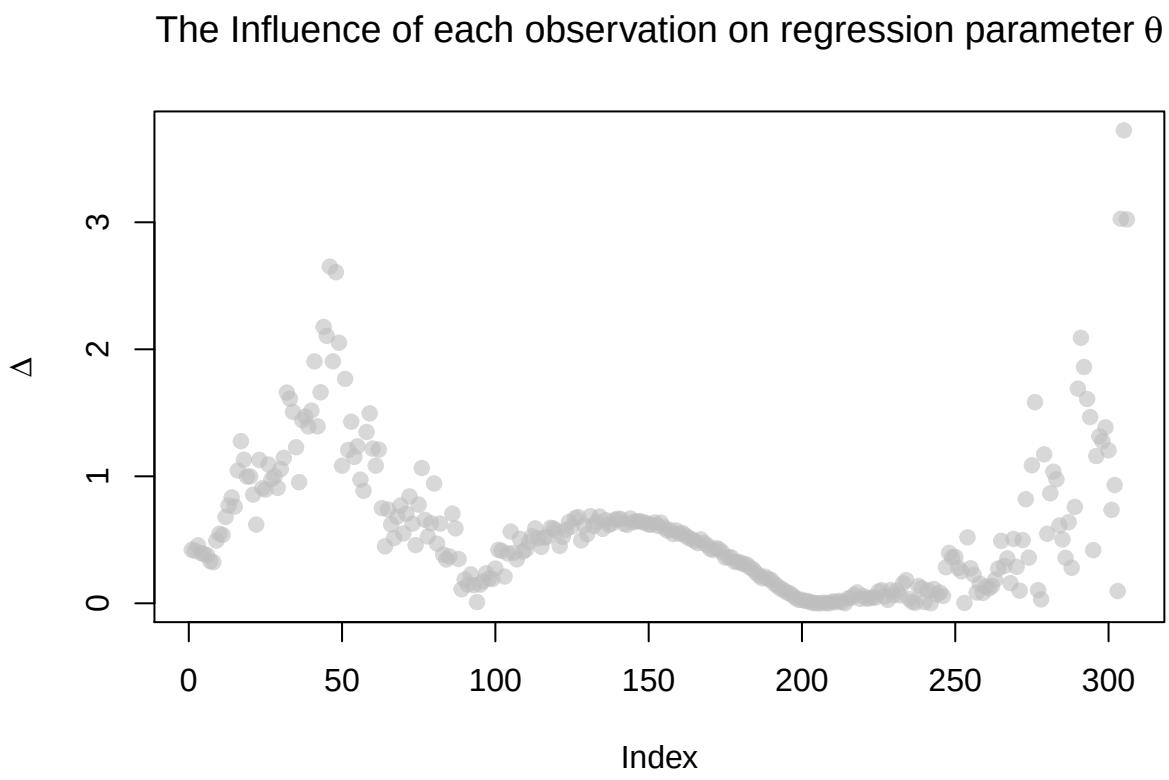

Histogram of adjusted new cas
Frequency
0
20
40
60
80
100
120
0
200
600
1000
Adjusted new cases
Boxplot of adjusted new case
Adjusted new cases
0
200
400
600
800
1000
Quantile plot of adjusted new ca
Qy(p)
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0
Proportion p

(c)

```
plot(tor$Day_Num, tor$New_Cases, main = "Scatterplot of adjusted new cases versus day number", pch = 19
     xlab = "Day number",
     ylab = "Adjusted new cases")
fit = lm(New_Cases~Day_Num, data=tor)
abline(fit, col="red", lwd=2)
```




```
N = nrow(tor)
delta = matrix(0, nrow = N, ncol = 2)
for(i in 1:N) {
  fit.no.i = lm(New_Cases ~ Day_Num, data = tor[-i,])
  delta[i, ] = abs(fit$coef - fit.no.i$coef)
}

delta2 = apply(X = delta, MARGIN = 1, FUN =function(z) {
  sqrt(sum(z^2))})

plot(delta2, main = bquote("The Influence of each observation on regression parameter" ~ theta), ylab =
     col = adjustcolor("grey", 0.6))
```



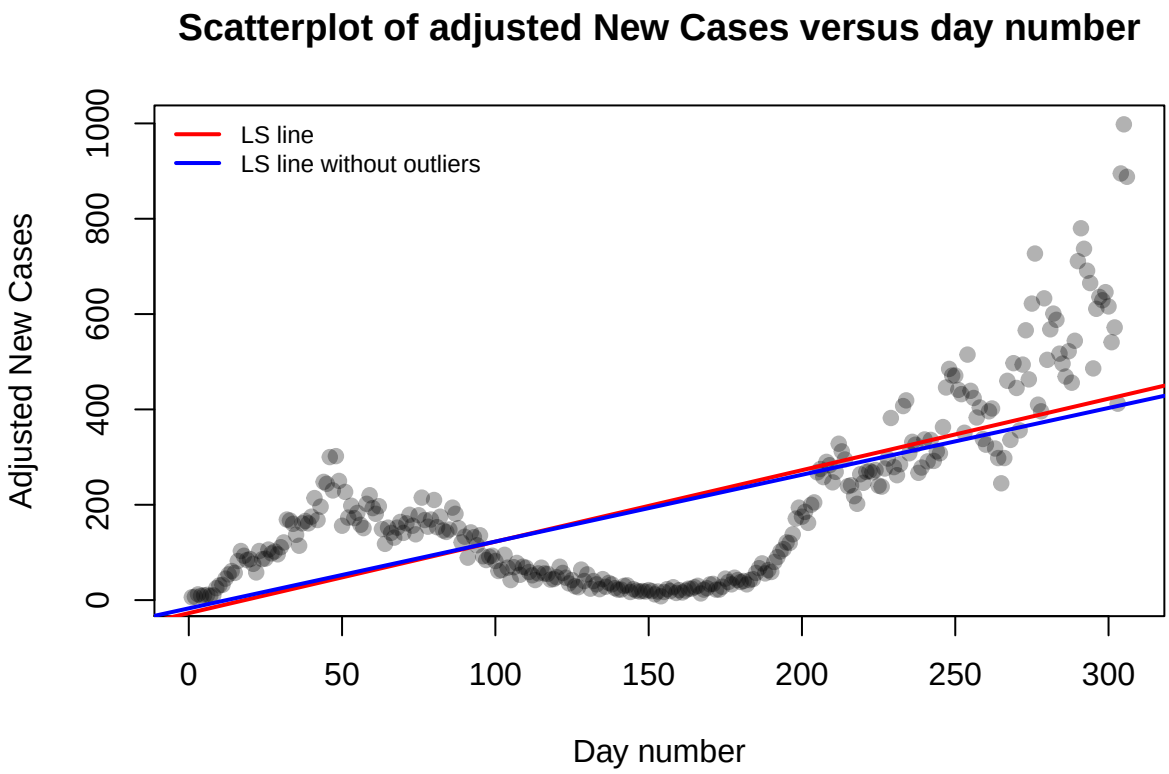
We are going to find the three most influential observations from the population

```
tor$Date[delta2>=sort(delta2, decreasing = TRUE)[3]]
```

```
## [1] "12/29/20" "12/30/20" "12/31/20"
```

Then we are going to remove the three most influential observations from the population and calculate the least squares regression line using the observations that remain.

```
plot(tor$Day_Num, tor$New_Cases, main = "Scatterplot of adjusted New Cases versus day number", pch = 19
     xlab = "Day number",
     ylab = "Adjusted New Cases")

abline(fit, col="red", lwd=2)
index = which(delta2>=sort(delta2, decreasing = TRUE)[3])
w = rep(1, N)
w[index] = 0
fit2 = lm(New_Cases~Day_Num, weights = w, data = tor)

abline(fit2,col="blue",lwd=2)

legend("topleft", legend=c("LS line", "LS line without outliers"), col=c("red", "blue"),
                           cex = 0.75, bty = "n", lty = 1, lwd =2)
```



Instead of removing highly influential observations, we could perform a robust linear regression to mitigate their influence. In addition, we use Turkey's bisquare objective function in the robust linear regression.

$$\rho_k(r) = \begin{cases} \frac{r^2}{2} - \frac{r^4}{2k^2} + \frac{r^6}{6k^4} & \text{for } |r| \leq k \\ \frac{k^2}{6} & \text{for } |r| > k \end{cases}$$

The above is the Turkey Bisquare objective function we are going to use.

```
tukey.fn <- function(r, k) {
  val = (r^2)/2 - (r^4)/(2*k^2) + (r^6)/(6*k^4)
  subr = abs(r) > k
  val[ subr ] = (k^2)/6
  return(val)
}
tukey.fn.prime <- function(r, k) {
  val = r - (2*r^3)/(k^2) + (r^5)/(k^4)
  subr = abs(r) > k
  val[ subr ] = 0
  return(val)
}
```

```
createRobustTukeyRho <- function(x, y, kval) {
    function(theta) {
        alpha <- theta[1]
        beta <- theta[2]
        sum(tukey.fn(y - alpha - beta * x, k = kval))
```

```
    }
}

createRobustTukeyGradient <- function(x, y, kval) {
    function(theta) {
        alpha <- theta[1]
        beta <- theta[2]
        ru = y - alpha - beta * x
        rhok = tukey.fn.prime(ru, k = kval)
        -1 * c(sum(rhok * 1), sum(rhok * x))
    }
}
```

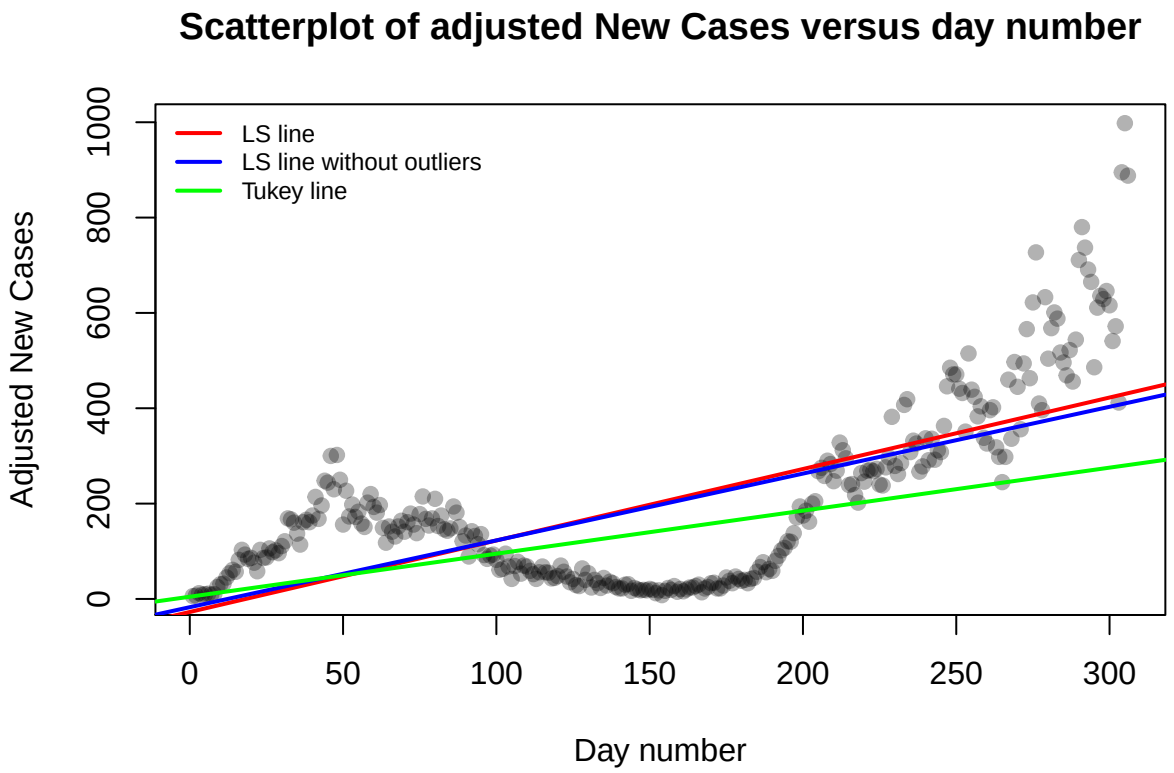
Then, we are going to use k = 6

```
rho <- createRobustTukeyRho(x=tor$Day_Num, y=tor$New_Cases, k=6)
gradient <- createRobustTukeyGradient(x=tor$Day_Num, y=tor$New_Cases, k=6)
result <- nlminb(start = c(0,1), objective = rho, gradient = gradient)
result
```

```
## $par
## [1] 4.6124967 0.9027154
##
## $objective
## [1] 1759.225
##
## $convergence
## [1] 0
##
## $iterations
## [1] 8
##
## $evaluations
## function gradient
##       21        9
##
## $message
## [1] "relative convergence (4)"
```

```
plot(tor$Day_Num, tor$New_Cases, main = "Scatterplot of adjusted New Cases versus day number", pch = 19
     xlab = "Day number",
     ylab = "Adjusted New Cases")

abline(fit, col="red", lwd=2)
index = which(delta2>=sort(delta2, decreasing = TRUE)[3])
w = rep(1, N)
w[index] = 0
fit2 = lm(New_Cases~Day_Num, weights = w, data = tor)

abline(fit2,col="blue",lwd=2)

abline(result$par, col="green", lwd=2)
```

```
legend("topleft", legend=c("LS line", "LS line without outliers","Tukey line"),
       col=c("red", "blue","green"),
       cex = 0.75, bty = "n", lty = 1, lwd =2)
```

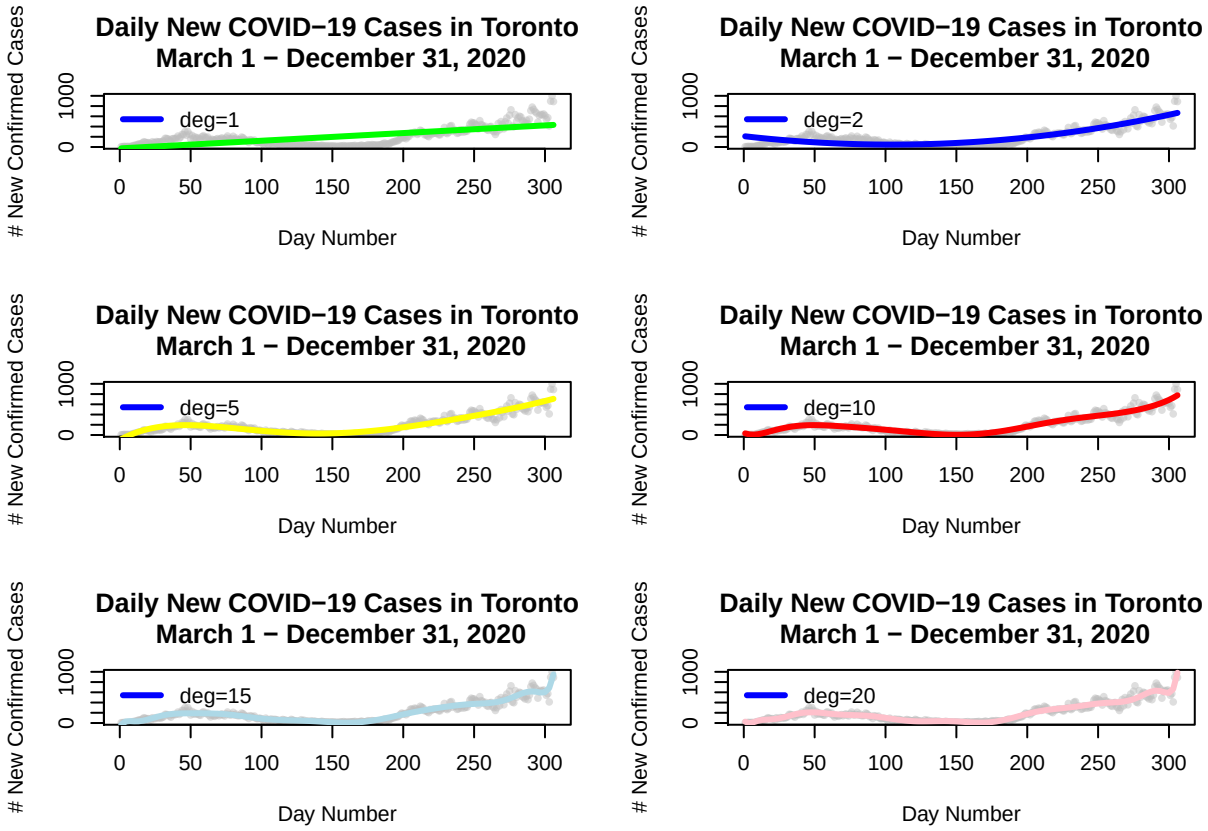


By observing the three linear regression line, we can see that the turkey line(green one) is less affected by the outliers.

Next, we are going to use polynomials to model the covid-19 data.

```
par(mfrow=c(3,2))
plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
muhat1 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 1)
curve(muhat1, from = 1, to = 306, add = TRUE, col = "green", lwd = 3)
legend("topleft", legend = "deg=1", col = "blue", lwd = 3, bty = "n")

plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
```

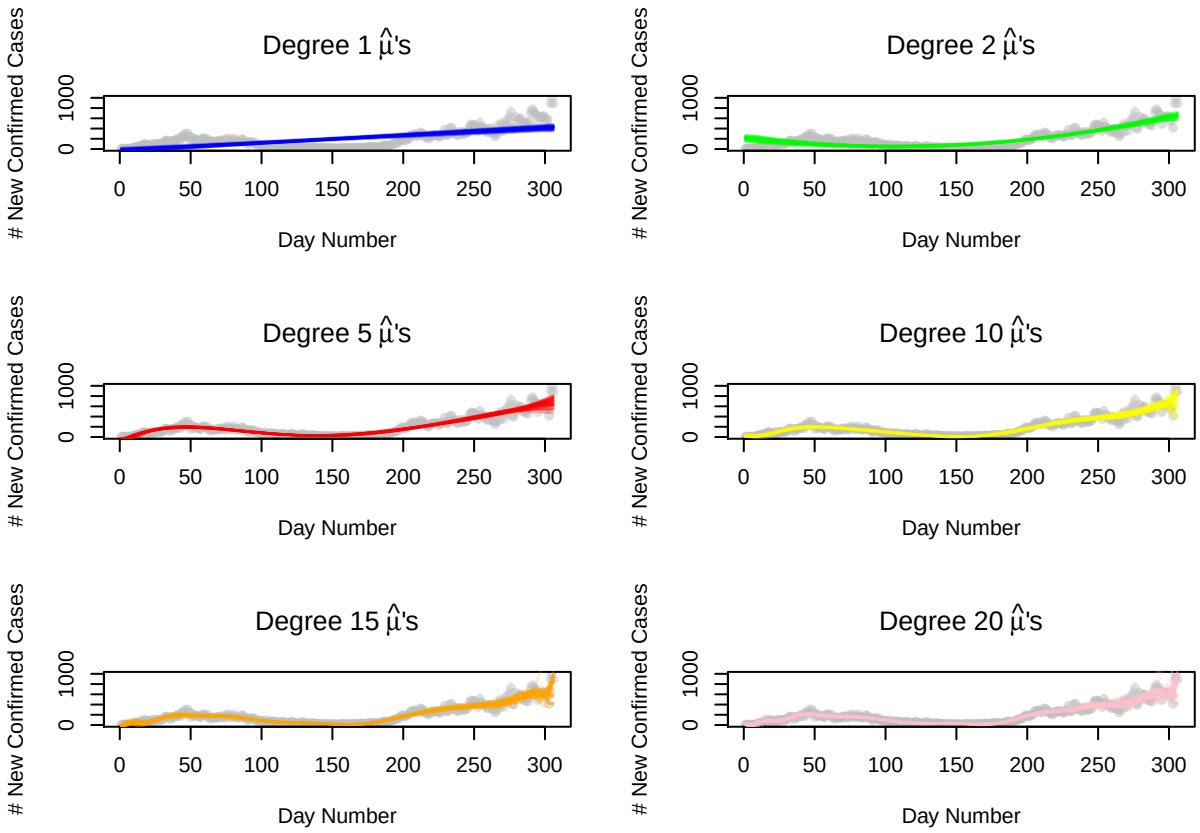
```r
muhat2 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 2)
curve(muhat2, from = 1, to = 306, add = TRUE, col = "blue", lwd = 3)
legend("topleft", legend = "deg=2", col = "blue", lwd = 3, bty = "n")

plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
muhat5 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 5)
curve(muhat5, from = 1, to = 306, add = TRUE, col = "yellow", lwd = 3)
legend("topleft", legend = "deg=5", col = "blue", lwd = 3, bty = "n")

plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
muhat10 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 10)
curve(muhat10, from = 1, to = 306, add = TRUE, col = "red", lwd = 3)
legend("topleft", legend = "deg=10", col = "blue", lwd = 3, bty = "n")

plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
muhat15 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 15)
curve(muhat15, from = 1, to = 306, add = TRUE, col = "lightblue", lwd = 3)
legend("topleft", legend = "deg=15", col = "blue", lwd = 3, bty = "n")

plot(x = tor$Day_Num, y = tor$New_Cases, pch=16, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
muhat20 <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), 20)
curve(muhat20, from = 1, to = 306, add = TRUE, col = "pink", lwd = 3)
legend("topleft", legend = "deg=20", col = "blue", lwd = 3, bty = "n")
```




```
muhats1 <- lapply(Ssamples, getmuhat, complexity = 1)
muhats2 <- lapply(Ssamples, getmuhat, complexity = 2)
muhats5 <- lapply(Ssamples, getmuhat, complexity = 5)
muhats10 <- lapply(Ssamples, getmuhat, complexity = 10)
muhats15 <- lapply(Ssamples, getmuhat, complexity = 15)
muhats20 <- lapply(Ssamples, getmuhat, complexity = 20)
```

```
par(mfrow=c(3,2))

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 1" ~ hat(mu) * "'s"))
for (i in 1:M) {
    curveFn <- muhats1[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("blue",
        0.25))}

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 2" ~ hat(mu) * "'s"))
for (i in 1:M) {
```

```
    curveFn <- muhats2[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("green",
        0.25))}

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 5" ~ hat(mu) * "'s"))
for (i in 1:M) {
    curveFn <- muhats5[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("red",
        0.25))}

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 10" ~ hat(mu) * "'s"))
for (i in 1:M) {
    curveFn <- muhats10[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("yellow",
        0.25))}

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 15" ~ hat(mu) * "'s"))
for (i in 1:M) {
    curveFn <- muhats15[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("orange",
        0.25))}

plot(x = tor$Day_Num, y = tor$New_Cases, pch = 19, cex = 0.8,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
     main = bquote("Degree 20" ~ hat(mu) * "'s"))
for (i in 1:M) {
    curveFn <- muhats20[[i]]
    curve(curveFn, from = 1, to = 306, add = TRUE, col = adjustcolor("pink",
        0.25))}
```

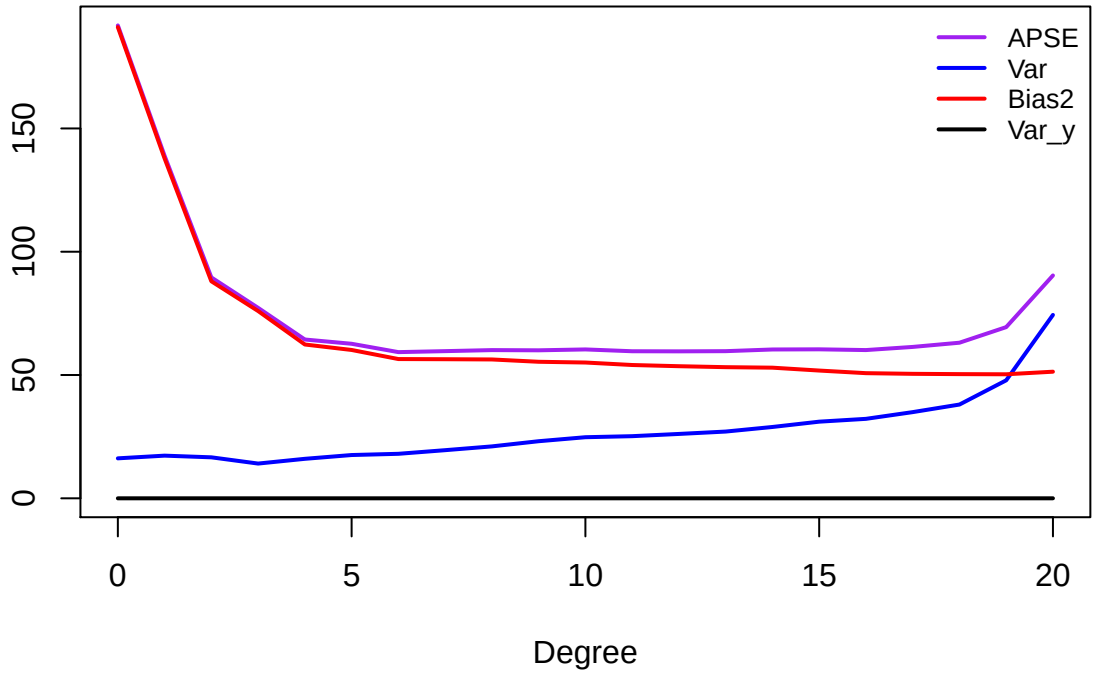



```
tau.day <- gettauFun(tor, "Day_Num", "New_Cases")
degrees <- 0:20
apse_vals <- sapply(degrees, FUN = function(complexity) {
  apse_all(Ssamples, Tsamples, complexity = complexity, tau = tau.day)})

t(rbind(degrees, apse = round(apse_vals, 5)))
```

```
##       degrees      apse var_mutilde     bias2 var_y
##  [1,]       0 36780.539    262.6077 36517.932     0
##  [2,]       1 19297.953    299.2158 18998.738     0
##  [3,]       2  8023.492    276.1977  7747.294     0
##  [4,]       3  5966.170    198.5890  5767.581     0
##  [5,]       4  4147.090    256.5894  3890.501     0
##  [6,]       5  3927.015    308.0987  3618.916     0
##  [7,]       6  3514.476    325.7475  3188.728     0
##  [8,]       7  3562.171    380.9529  3181.218     0
##  [9,]       8  3613.195    443.0774  3170.117     0
## [10,]       9  3601.815    535.9507  3065.864     0
## [11,]      10  3645.402    613.6563  3031.746     0
## [12,]      11  3557.267    634.7055  2922.562     0
## [13,]      12  3552.039    681.6326  2870.407     0
## [14,]      13  3560.835    732.7227  2828.112     0
## [15,]      14  3643.028    836.8560  2806.172     0
## [16,]      15  3650.072    966.1810  2683.891     0
## [17,]      16  3613.607   1038.2253  2575.382     0
```

```
## [18,]      17  3769.307   1221.5320  2547.775     0
## [19,]      18  3981.631   1445.2355  2536.396     0
## [20,]      19  4819.604   2289.9750  2529.629     0
## [21,]      20  8167.623   5531.3125  2636.310     0
```

```
plot(degrees, sqrt(apse_vals[1, ]), xlab = "Degree", ylab = "", type = "l",
    ylim = c(0, max(sqrt(apse_vals))), col = "purple", lwd = 2)
lines(degrees, sqrt(apse_vals[2, ]), col = "blue", lwd = 2)
lines(degrees, sqrt(apse_vals[3, ]), col = "red", lwd = 2)
lines(degrees, sqrt(apse_vals[4, ]), col = "black", lwd = 2)
legend("topright", legend = c("APSE", "Var", "Bias2", "Var_y"), col = c("purple",
    "blue", "red", "black"), lwd = 2, bty = "n", cex = 0.8)
```



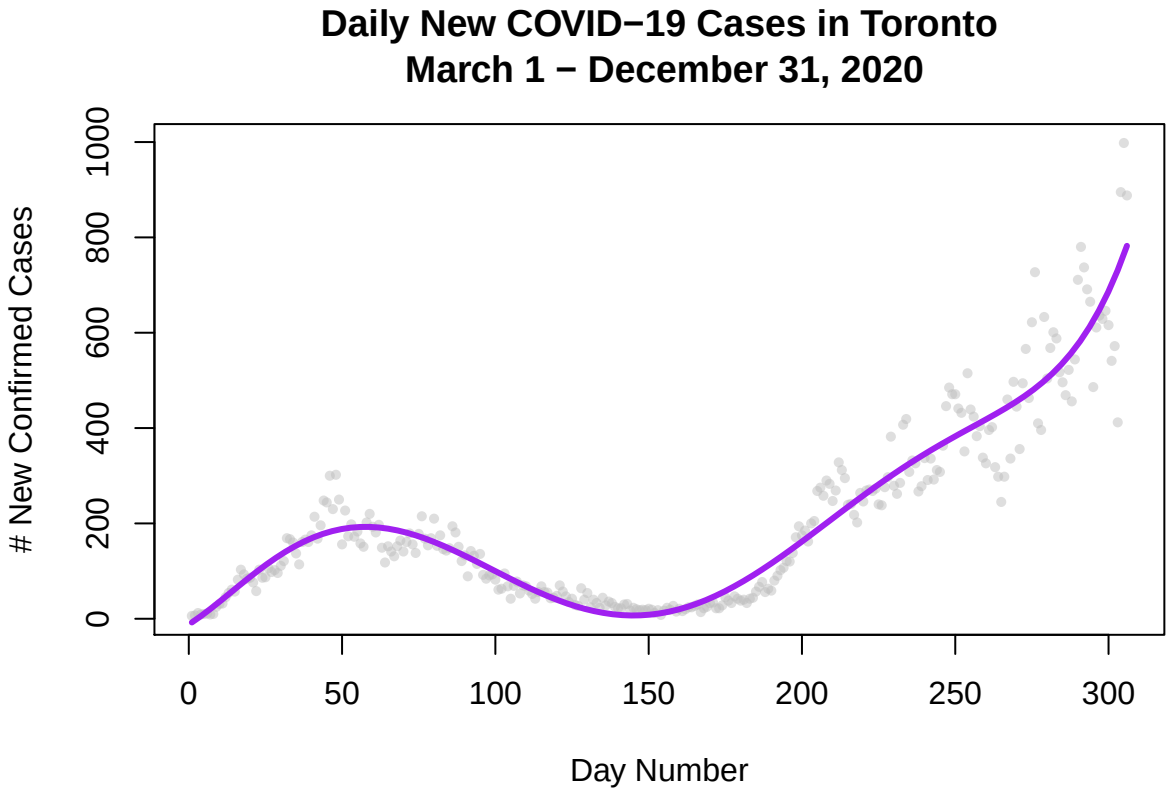

```
best.deg <- degrees[which.min(apse_vals[1, ])]
print(best.deg)
```

```
## [1] 6
```

```
muhat.best <- getmuhat(data.frame(x = tor$Day_Num, y = tor$New_Cases), best.deg)
```

```
plot(x = tor$Day_Num, y = tor$New_Cases, pch = 16, cex = 0.7,
     col = adjustcolor("grey", 0.5),
     xlab = "Day Number",
     ylab = "# New Confirmed Cases",
```

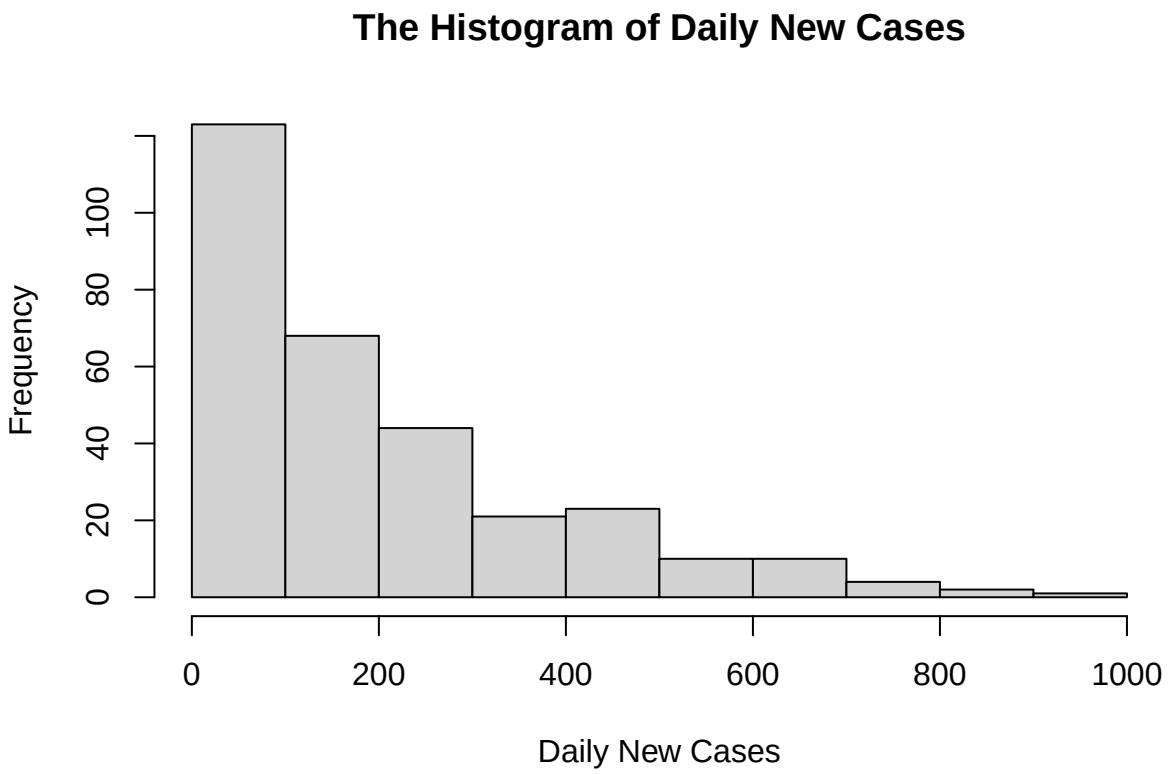
```
    main = "Daily New COVID-19 Cases in Toronto\n March 1 - December 31, 2020")
curve(muhat.best, from = 1, to = 306, add = TRUE, col = "purple", lwd = 3)
```



Thus, we find the best polynomial model with degree = 6.

Then, we are going to use gamma distribution as the potential model for the data and use Method of Moments estimation and Maximum Likelihood (ML) estimation to estimate the paramaters alpha and beta for the distribution.

```
hist(tor$New_Cases, main = 'The Histogram of Daily New Cases', xlab = 'Daily New Cases')
```




```
beta.mm =(mean(tor$New_Cases^2)-(mean(tor$New_Cases))^2) / mean(tor$New_Cases)
alpha.mm = mean(tor$New_Cases) / beta.mm
c(alpha.mm, beta.mm)
```

```
## [1]   1.127008 179.989514
```

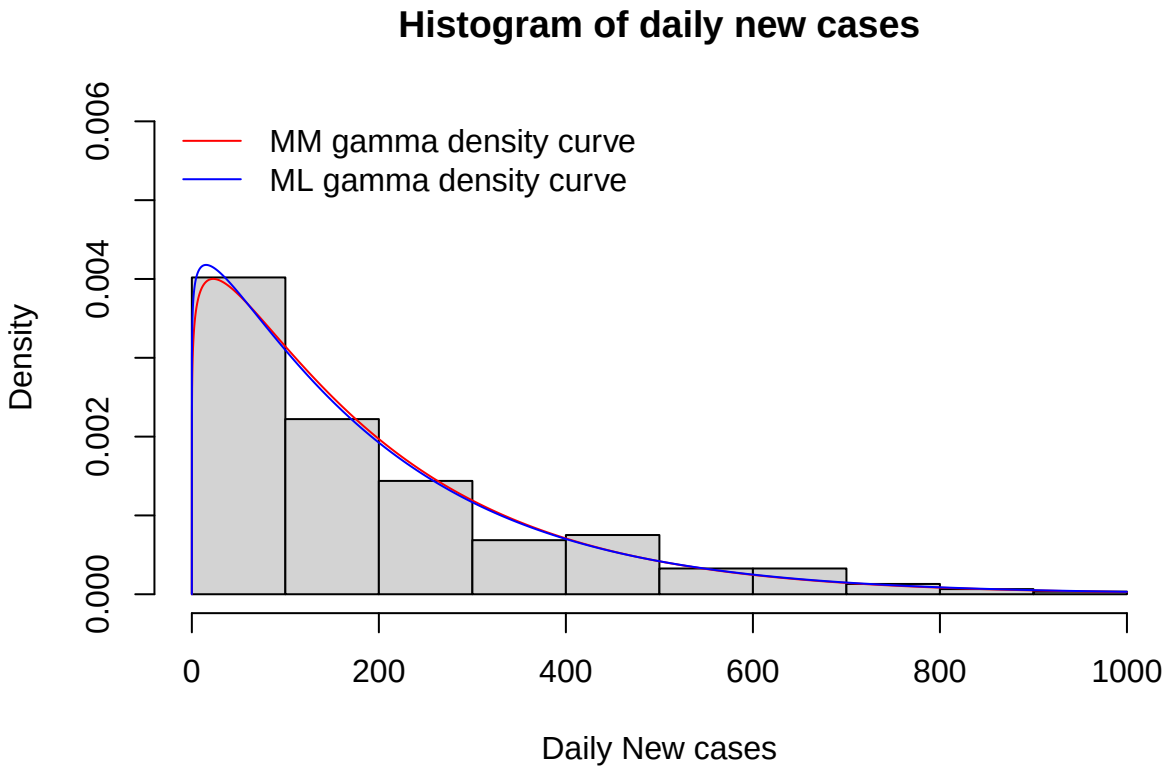
```r
createGammaPsifn <- function(y){
  N <- length(y)
  function(theta){
    alpha <- theta[1]
    beta <- theta[2]
    c(-N* digamma(alpha)- N*log(beta) + sum(log(y)), -alpha*N/beta + sum(y)/beta^2)
  }
}

creatGammaPsiPrimeFn <- function(y){
  N <- length(y)
  function(theta){
    alpha = theta[1]
    beta = theta[2]
    mat = matrix(0, nrow=length(theta), ncol=length(theta))
    mat[1,1] = -N*trigamma(alpha)
    mat[1,2] = -N/beta
    mat[2,1] = -N/beta
    mat[2,2] = alpha*N/beta^2-2*sum(y)/beta^3
    return(mat)
  }
}
```

```r
psi <- createGammaPsifn(tor$New_Cases)
psiPrime <- creatGammaPsiPrimeFn(tor$New_Cases)
Result <- NewtonRaphson(theta = c(alpha.mm, beta.mm), psiFn = psi, psiPrimeFn = psiPrime)
print(Result)
```

```
## $theta
## [1]   1.081193 187.616478
##
## $converged
## [1] TRUE
##
## $iteration
## [1] 4
##
## $fnValue
## [1] 0 0
```

```r
alpha.ml = Result$theta[1]
beta.ml = Result$theta[2]
```

```r
hist(tor$New_Cases, prob=TRUE, ylim = c(0, 0.006), main = 'Histogram of daily new cases', xlab = 'Daily
x = seq(0, 1000, 0.1)
lines(x, dgamma(x, shape = alpha.mm, scale = beta.mm), col='red')
lines(x, dgamma(x, shape = alpha.ml, scale = beta.ml), col='blue')
legend('topleft', legend =c('MM gamma density curve', 'ML gamma density curve'), col = c('red','blue'),
```



From the above plot, we can see that MM gamma density curve and ML gamma density curve is similar. And we can use the gamma model to describe the probability of the certain number of Covid-19 daily new cases of the future.

By comparison, we can clearly see that the polynomial model with degree 6 is better fitted to the data than linear regression model and linear robust regression model. Then we can use the polynomial model with degree 6 to predict the future Covid-19 daily new cases.