Homework 3

Yuhao Mao 3170102264

2020/7/11

---

0. Read in data

```
gmp <- read.table("data/gmp.dat")
gmp$pop <- round(gmp$gmp/gmp$pcgmp)
```

1. Plot with per capita GMP on the y-axis and population on the x-axis. Blue line is for $a = \frac{1}{8}$, red line is for $a = 0.15$ and purple line is for $a = 0.1$.

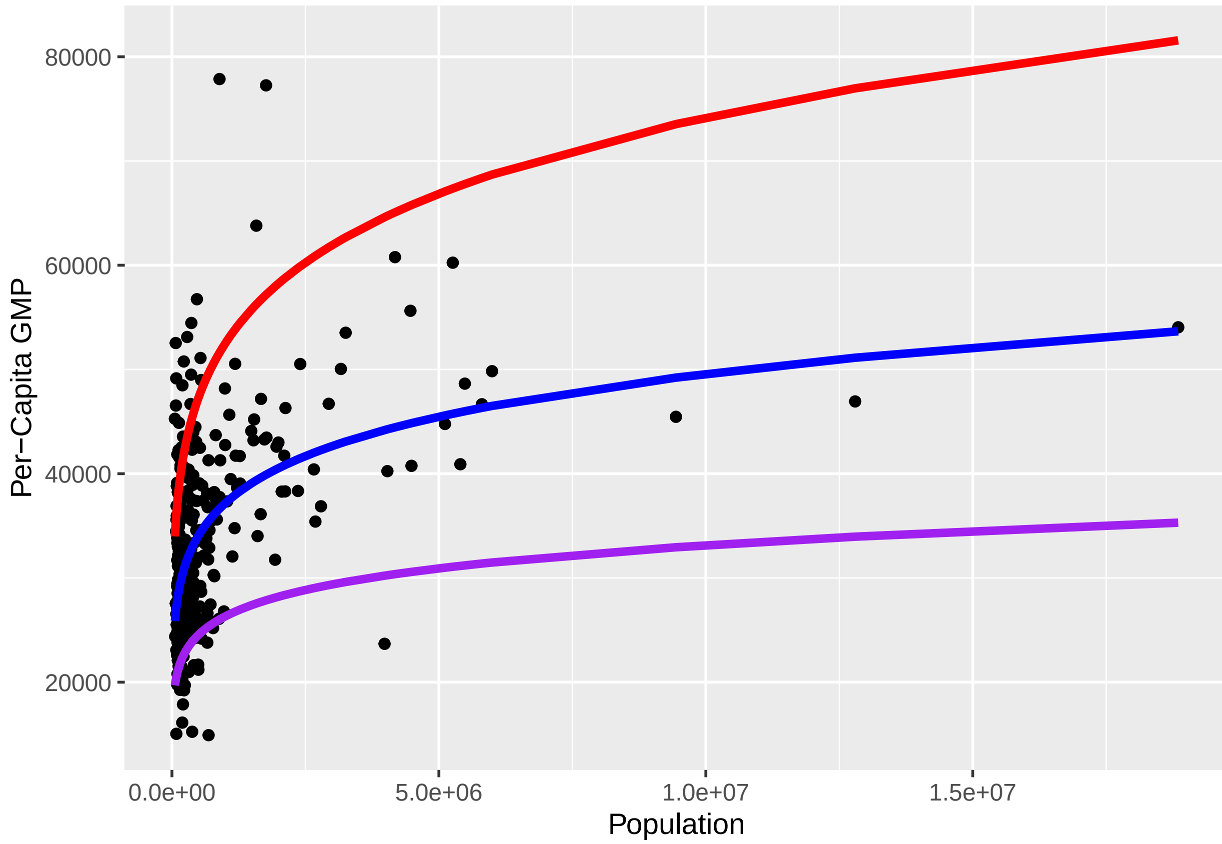
```
library(tidyverse)
```

```
## -- Attaching packages ---------------------- tidyverse 1.3.0.9000 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ----------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
gmp %>% ggplot() +
  geom_point(aes(x = pop, y = pcgmp))+
  labs(x = "Population", y = "Per-Capita GMP")+
  geom_line(aes(x = pop, y = 6611*pop^(1/8)), col = 'blue', size = 1.5)+
  geom_line(aes(x=pop,y=6611*pop^(0.15)), col='red',size=1.5)+
  geom_line(aes(x=pop,y=6611*pop^(0.1)), col='purple',size=1.5)
```



2. The MSE function.

```
mse <- function(params, N=gmp$pop, Y=gmp$pcgmp){
  y0 <- params[1]
  a <- params[2]
  return(mean((Y-y0*N^a)^2))
}
mse(c(6611,0.15))
```

```
## [1] 207057513
```

```
mse(c(5000,0.1))
```

```
## [1] 298459914
```

4. The `nlm()` function.

```
nlm(mse, c(y0=6611,a=1/8))
```

```
## $minimum
## [1] 61857060
##
## $estimate
## [1] 6611.0000000    0.1263177
##
## $gradient
## [1] 50.048639 -9.983778
##
## $code
```

```
## [1] 2
##
## $iterations
## [1] 3
```

```r
nlm(mse, c(y0=5000,a=0.1))
```

```
## $minimum
## [1] 62521484
##
## $estimate
## [1] 5000.0000008    0.1475913
##
## $gradient
## [1] -1028.22544    11.38762
##
## $code
## [1] 2
##
## $iterations
## [1] 5
```

```r
nlm(mse, c(y0=7000,a=0.1))
```

```
## $minimum
## [1] 61908051
##
## $estimate
## [1] 7000.0000002    0.1219422
##
## $gradient
## [1] 203.5103014   0.7078052
##
## $code
## [1] 2
##
## $iterations
## [1] 7
```

`estimate` represents the best parameter that the function derived that achieves smallest function value and `minimum` represents the smallest function value.

5. The `plm()` function.

```r
plm <- function(y0, a, N=gmp$pop, Y=gmp$pcgmp){
  result <- nlm(mse,c(y0,a))
  l <- list(y_final=result$estimate[1],a_final=result$estimate[2],minimum=result$minimum)
  return(l)
}
plm(6611,0.15)
```

```
## $y_final
## [1] 6611
##
## $a_final
## [1] 0.1263182
##
```

```
## $minimum
## [1] 61857060
```

```
plm(5000,0.1)
```

```
## $y_final
## [1] 5000
##
## $a_final
## [1] 0.1475913
##
## $minimum
## [1] 62521484
```

They are not same because it is not a convex optimization and the former one has lower MSE.

6. Jackknife

- a. `mean()` and `sd()`.

```
mean(gmp$pcgmp)
```

```
## [1] 32922.53
```

```
sd(gmp$pcgmp)/sqrt(length(gmp$pcgmp))
```

```
## [1] 481.9195
```

- b. The function `mean.minus.i()`.

```
mean.minus.i <- function(i, v=gmp$pcgmp){
  return(mean(v[-i]))
}
```

+c. The vector `jackknifed.means`.

```
jackknifed.means <- c()
for(i in 1:length(gmp$pcgmp)){
  jackknifed.means <- c(jackknifed.means, mean.minus.i(i))
}
```

+d. The standard deviation derived from jackknife.

```
  n <- dim(gmp)[1]
  sqrt(var(jackknifed.means)/n*(n-1)^2)
```

```
## [1] 481.9195
```

The result is exactly the same to (a).

7. The `plm.jackknife()` function.

```
plm.jackknife <- function(y0, a, N=gmp$pop, Y=gmp$pcgmp){
  y0_estimate <- c()
  a_estimate <- c()
  for(i in 1:length(N)){
    result <- plm(y0,a,N[-i],Y[-i])
    y0_estimate <- c(y0_estimate, result$y_final)
    a_estimate <- c(a_estimate, result$a_final)
  }
  n <- length(N)
  l <- list(y0_sd=sqrt(var(y0_estimate)/n*(n-1)^2), a_sd=sqrt(var(a_estimate)/n*(n-1)^2))
```

```r
  return(l)
}
plm.jackknife(6611, 0.1)
```

```
## $y0_sd
## [1] 0
##
## $a_sd
## [1] 0
```

They are both zero because after deleting one element from the data, the estimate is still the same in this case. That is because the original data size is 366, which is much larger than 1.

8. Estimates for gmp 2013.

```r
gmp <- read.table("data/gmp-2013.dat")
gmp$pop <- round(gmp$gmp/gmp$pcgmp)
plm(6611,0.1)
```

```
## $y_final
## [1] 6611
##
## $a_final
## [1] 0.1433688
##
## $minimum
## [1] 135210524
```

```r
plm.jackknife(6611,0.1)
```

```
## $y0_sd
## [1] 0
##
## $a_sd
## [1] 0
```

The estimate for a is changed significantly.