

```
if (!require(ggplot2)) install.packages("ggplot2")
```

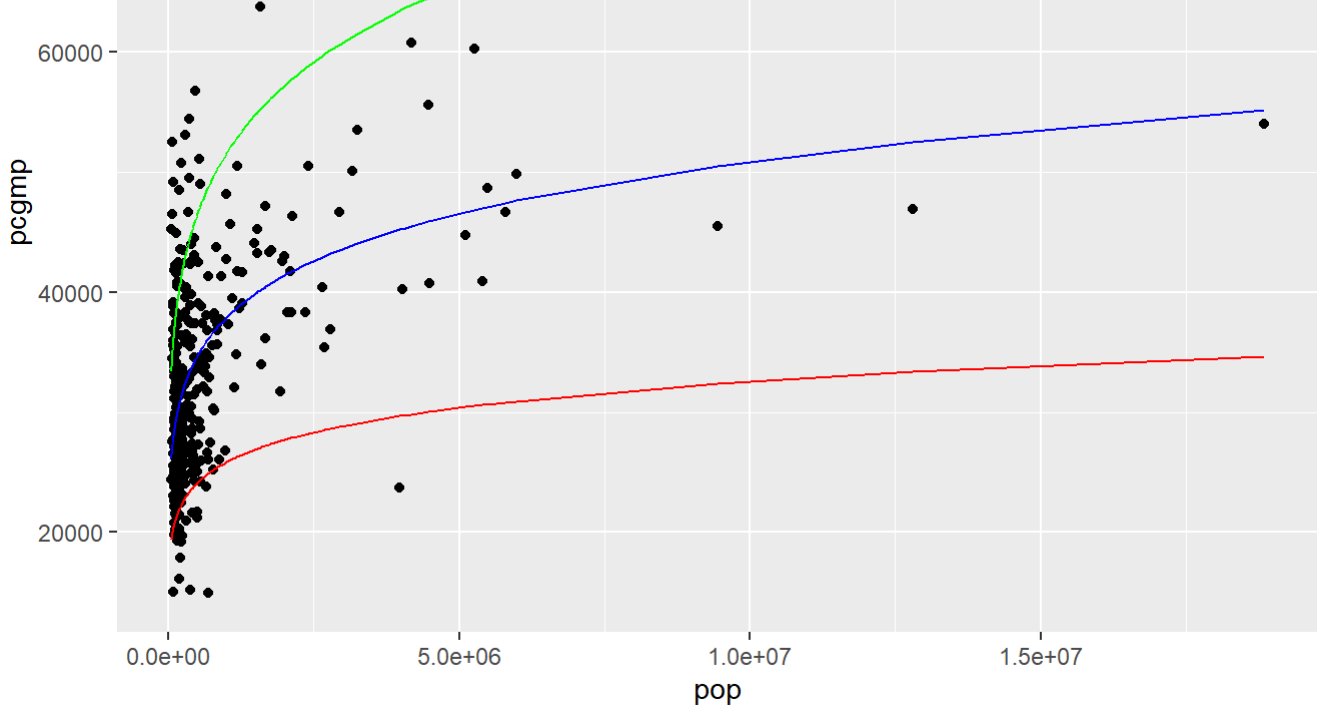
```
## 载入需要的程序包：ggplot2
```

```
library(ggplot2)
```

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

1.

```
func <- function(x, y0, a) {  
  return(y0 * x ^ a)  
}  
ggplot(data = gmp) +  
  geom_point(aes(x = pop, y = pcgmp)) +  
  geom_line(aes(x = pop, y = func(pop, 6493.2563726, 0.1276921)), col = "blue") +  
  geom_line(aes(x = pop, y = func(pop, 6493.2563726, 0.1)), col = "red") +  
  geom_line(aes(x = pop, y = func(pop, 6493.2563726, 0.15)), col = "green")
```



2.

```
mse <- function(para, N = gmp$pop, Y = gmp$pcgmp) {
  return(mean((Y - para[1] * N ^ para[2]) ^ 2))
}
```

```
mse(c(6611, 0.15))
```

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

```
mse(c(5000, 0.10))
```

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

3.

```
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
```

```
nlm(mse, c(y0=6000,a=0.1))
```

```
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
## Warning in nlm(mse, c(y0 = 6000, a = 0.1)): NA/Inf被换成最大的正值
```

```
## $minimum
## [1] 61914531
##
## $estimate
## [1] 6000.0000004    0.1337231
##
## $gradient
## [1] -257.9030009   -0.7674098
##
## $code
## [1] 2
##
## $iterations
## [1] 5
```

```
nlm(mse, c(y0=6500,a=0.15))
```

minimum 表示其找到的函数最小值，estimate 表示找到 minimum 值时对应的变量数值。

[illegible]

```
## $MSE
## [1] 62521484
```

plm 中调用的 nlm 函数使用的是牛顿型算法来寻找最小值，其结果会与初始值的选取有关。此处当初始值选为 (y0 = 6611, a = 0.15) 时会得到更小的 MSE。

5.

a.

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

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

```
sd(gmp$pcgmp) / (length(gmp$pcgmp) ** 0.5)
```

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

```
(sum((gmp$pcgmp - mean(gmp$pcgmp)) ^ 2) / ((length(gmp$pcgmp) - 1) * length(gmp$pcgmp))) ^ 0.5
```

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

b.

```
mean_pcgmp_except <- function(i) {
  return(mean(gmp$pcgmp[-i]))
}
```

```
mean_pcgmp_except(i = 2)
```

```
## [1] 32922.62
```

c.

```
jackknifed.means <- c()
for (i in 1:length(gmp$pcgmp))
  jackknifed.means <- c(jackknifed.means, mean_pcgmp_except(i))
```

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

和 a 中的结果基本相符。## 6

```
plm.jackknife <- function(y0, a, N = gmp$pop, Y = gmp$pcgmp) {  
  jackknife.estimates <- list(y0 = c(), a = c())  
  for (i in 1:length(N)) {  
    estimates <- plm(y0, a, N[-i], Y[-i])  
    jackknife.estimates$y0 <- c(jackknife.estimates$y0, estimates$y0)  
    jackknife.estimates$a <- c(jackknife.estimates$a, estimates$a)  
  }  
  n <- length(N)  
  y0.variance = (n - 1) ^ 2 / n * var(jackknife.estimates$y0)  
  a.variance = (n - 1) ^ 2 / n * var(jackknife.estimates$a)  
  return(list(y0.sd = y0.variance ^ 0.5, a.sd = a.variance ^ 0.5))  
}
```

```
options(warn = -1)  
plm.jackknife(y0 = 6611, a = 1/8)
```

```
## $y0.sd  
## [1] 1.136653e-08  
##  
## $a.sd  
## [1] 0.0009901003
```

7.

```
gmp2013 <- read.table('data/gmp-2013.dat', header = T)  
gmp2013$pop <- gmp2013$gmp / gmp2013$pcgmp
```

```
options(warn = -1)  
plm(y0 = 6611, a = 1/8, N = gmp2013$pop, Y = gmp2013$pcgmp)
```

```
## $y0  
## [1] 6611  
##  
## $a  
## [1] 0.1433688  
##  
## $MSE  
## [1] 135210520
```

```
plm.jackknife(y0 = 6611, a = 1/8, N = gmp2013$pop, Y = gmp2013$pcgmp)
```

