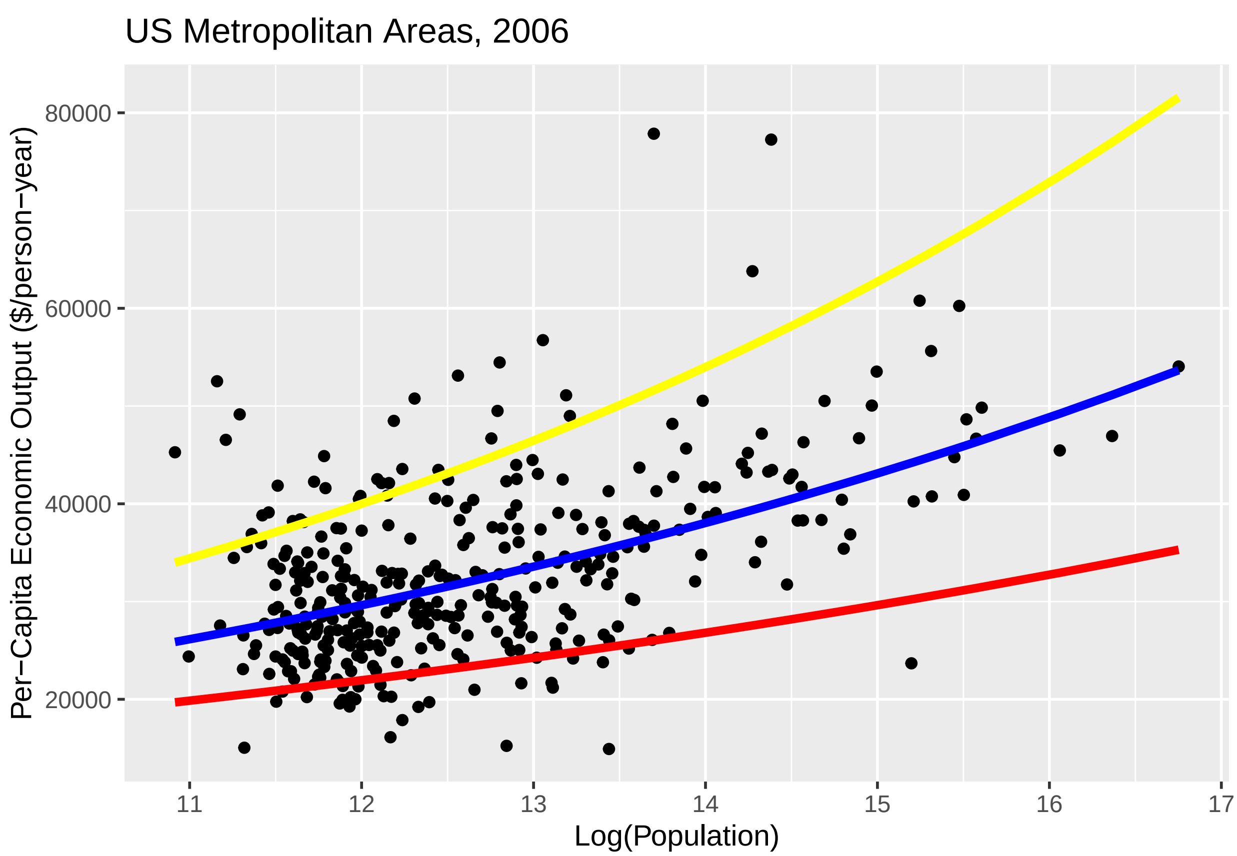
Homework 3: The Death and Life of Great American City Scaling Laws

Shen Dingtao 3170104764

1. Solution

```
gmp <- read.table("data/gmp.dat")
gmp <- gmp %>% mutate(log_pop = log(gmp/pcgmp),
      nlmfit1 = 6611*(gmp/pcgmp)^(1/8),
      nlmfit2 = 6611*(gmp/pcgmp)^(0.1),
      nlmfit3 = 6611*(gmp/pcgmp)^(0.15))
gmp %>% ggplot() + geom_point(aes(x = log_pop, y = pcgmp))+
  labs(x = "Log(Population)", y = "Per-Capita Economic Output ($/person-year)",
  title = "US Metropolitan Areas, 2006")+
  geom_line(aes(x = log_pop, y = nlmfit1), col = 'blue', size = 1.5)+
  geom_line(aes(x = log_pop, y = nlmfit2), col = 'red', size = 1.5)+
  geom_line(aes(x = log_pop, y = nlmfit3), col = 'yellow', size = 1.5)
```



2. Solution

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

```r
# Calculate mean squared error of nonlinear model

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

Check:

```r
mse(c(6611,0.15)) #The result should be 207057513
```

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

```r
mse(c(5000,0.10)) #The result should be 298459915
```

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

The second result has a little error with expected result because of the machine error between different computers.

4. Solution Case 1:

```r
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
```

Case 2:

```r
nlm(mse, c(y0=7000,a=0.12))
```

```
## $minimum
## [1] 61908051
```

```
##
## $estimate
## [1] 7000.0000000    0.1219426
##
## $gradient
## [1] 205.132132   4.426353
##
## $code
## [1] 2
##
## $iterations
## [1] 5
```

Case 3:

```
nlm(mse, c(y0=6000,a=0.15))
```

```
## $minimum
## [1] 61914531
##
## $estimate
## [1] 5999.9999998    0.1337231
##
## $gradient
## [1] -257.90269   24.00577
##
## $code
## [1] 2
##
## $iterations
## [1] 6
```

-`minimum` represents the minimized MSE

-`estimate` represents the estimates for `y0` and `a`

5. Solution

```
plm <- function(params, N=gmp$pop, Y=gmp$pcgmp) {
  nlm_result <- nlm(mse, params, N=N, Y=Y)
  y0_esti <- nlm_result[['estimate']][1]
  a_esti <- nlm_result[['estimate']][2]
  mse_esti <- nlm_result[['minimum']]
  return(c(y0_esti, a_esti, mse_esti))
}
```

(i)paras: `y0 = 6611 a = 0.15`

```
plm(c(6611, 0.15))
```

```
## [1] 6.611000e+03 1.263182e-01 6.185706e+07
```

The parameter estimates are 6611 and 0.1263182 for `y0` and `a`. And the MSE is 6.185706e+07.

(ii)paras: `y0 = 5000 a = 0.10`

```
plm(c(5000, 0.10))
```

```
## [1] 5.000000e+03 1.475913e-01 6.252148e+07
```

The parameter estimates are 5000 and 0.1475913 for `y0` and `a`. And the MSE is 6.252148e+07.

They are not the same. Because the convergence of `plm` depends on the initial parameters, which means they result from different fixed `y0` and `a`. For some initial parameters, it may converges to a local minimum.

The first case in which `y0=6611 a=0.15` has lower MSE.

6. *Convince yourself the jackknife can work.*

Solution:

(a)

```
avg.pcgmp<-mean(gmp$pcgmp)
avg.pcgmp
```

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

Using `sd()` and the formula for the standard error of the mean(SEM), we can calculate the SEM:

```
sem.pcgmp<-sd(gmp$pcgmp)/sqrt(length(gmp$pcgmp))
sem.pcgmp
```

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

(b) Following function calculates the mean per-capita GMP for every city except city number i.

```
exc_i_mean<-function(i,data=gmp$pcgmp){
  exp_i_mean<-mean(data[-i])
  return(exp_i_mean)
}
```

(c)

```
jackknifed.means<-sapply(c(1:length(gmp$pcgmp)),exc_i_mean)
```

(d)

```
mean.jackknife <- function(a_vector) {
  n <- length(a_vector)
  variance.of.ests <- var(a_vector)
  jackknife.var <- ((n-1)^2/n)*variance.of.ests
  jackknife.stderr <- sqrt(jackknife.var)
  return(jackknife.stderr)
}
mean.jackknife(jackknifed.means)
```

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

The result is equal to the answer in (a). We can also verify this by the following command:

```r
all.equal(sem.pcgmp,mean.jackknife(jackknifed.means))
```

```
## [1] TRUE
```

7 `plm.jackknife()`

Solution:

```r
plm.jackknife<-function(params,N=gmp$pop,Y=gmp$pcgmp){
  y0_jk<-c()
  a_jk<-c()
  n<-length(gmp$pcgmp)
  for(i in 1:n){
    plm_jk<-plm(params,N[-i],Y[-i])
    y0_jk<-c(y0_jk,plm_jk[1])
    a_jk<-c(a_jk,plm_jk[2])
  }
  y0_se<-(n-1)*sd(y0_jk)/sqrt(n)
  a_se<-(n-1)*sd(a_jk)/sqrt(n)
  return(c(y0_se,a_se))
}
```

```r
params_se<-plm.jackknife(c(6611,0.125))
params_se
```

```
## [1] 1.136653e-08 9.901003e-04
```

8. Solution

Load the data set:

```r
gmp2013 <- read.table("data/gmp-2013.dat")
gmp2013$pop <- round(gmp2013$gmp/gmp2013$pcgmp)
```

use `plm()` and `plm.jackknife` to estimate the parameters of the model for 2013, and their standard errors:

```r
params_esti<-plm(c(6611,0.125),N=gmp2013$pop,Y=gmp2013$pcgmp)
print(paste("Estimation:y0=",params_esti[1],",a=",params_esti[2],",MSE:",params_esti[3]))
```

```
## [1] "Estimation:y0= 6611.00000023012 ,a= 0.143368784715643 ,MSE: 135210524.492386"
```

```r
params_esti_se<-plm.jackknife(c(6611,0.125),N=gmp2013$pop,Y=gmp2013$pcgmp)
print(paste("SE:y0=",params_esti_se[1],",a=",params_esti_se[2]))
```

```
## [1] "SE:y0= 2.67739123994886e-08 ,a= 0.00109088274013141"
```

The estimation for `y0` doesn't change significantly, it's still near to 6611, while the estimation for `a` changes pretty significantly, from 0.126 to 0.143.