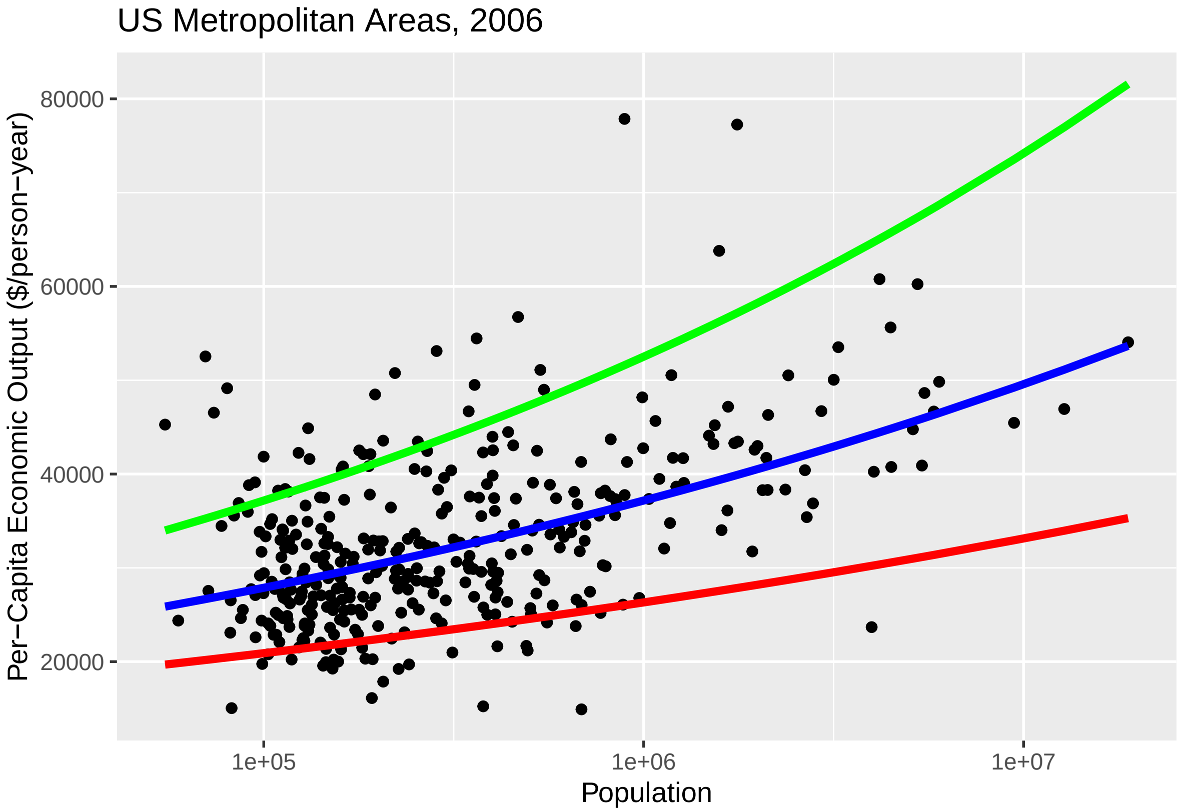
Homework 3: The Death and Life of Great American City Scaling Laws

DingHuan, 3170102085

1.

```r
gmp <- read.table("gmp.dat")
gmp$pop <- round(gmp$gmp/gmp$pcgmp)
gmp <- gmp %>% mutate(nlmfit = 6611*pop^(1/8),
                      a0.1 = 6611*pop^(0.1),
                      a0.15 = 6611*pop^(0.15))
gmp %>% ggplot + geom_point(aes(x = pop, y = pcgmp)) +
  labs(x = "Population", y = "Per-Capita Economic Output ($/person-year)",
    title = "US Metropolitan Areas, 2006") +
  geom_line(aes(x = pop, y = nlmfit), col = 'blue', size = 1.5) +
  geom_line(aes(x = pop, y = a0.1), col = 'red', size = 1.5) +
  geom_line(aes(x = pop, y = a0.15), col = 'green', size = 1.5) +
  scale_x_log10()
```



2.

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

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

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

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

3. The quantity `minimum` represents the optimizing result, the minimum of `mse()` by `nlm()` with Newton-type optimizing algorithm. The quantity `estimate` represents the end of optimizing iteration, the optimal point estimated by `nlm()`. On the other words, `mse()` have a minimum value `minimum`, which is obtained at point `estimate`.

```
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=6600,a=0.1))
```

```
## $minimum
## [1] 61856513
##
## $estimate
## [1] 6600.0000003    0.1264451
##
## $gradient
## [1]   45.28051 -139.63133
##
## $code
## [1] 2
##
## $iterations
## [1] 6
```

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

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

4. The estimates for those two pairs of parameters are different. Because the robustness of the algorithm is not good enough, and there are plenty of local optimal points around the global one, so the results fall on the local ones with an unexpected starting points.

```
plm <- function(y0, a, N = gmp$pop, Y = gmp$pcgmp){
  pari = c(y0, a)
  n <- nlm(mse, pari)
  m = list(final_guess_y0 = n$estimate[1],
           final_guess_a = n$estimate[2],
           final_value_MSE = n$minimum)
  return(m)
}
plm(6611,0.15)
```

```
## $final_guess_y0
## [1] 6611
##
## $final_guess_a
## [1] 0.1263182
##
## $final_value_MSE
## [1] 61857060
```

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

```
## $final_guess_y0
## [1] 5000
##
## $final_guess_a
## [1] 0.1475913
##
## $final_value_MSE
## [1] 62521484
```

5. a.

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

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

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

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

b.

```
except <- function(i){
  m = mean(gmp[-i,"pcgmp"])
  return(m)
}
```

c.

```
jackknifed.means <- sapply(1:nrow(gmp), except)
```

d. It matches quite well.

```
n <- length(jackknifed.means)
m <- mean(jackknifed.means)
sqrt(sum((jackknifed.means-m)^2)*(n-1)/n)
```

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

6.
```
plm.jackknife <- function(y0, a){
  jackknifed.gmp <- data.frame()
  for (i in 1:nrow(gmp)){
    m = data.frame(plm(y0, a, N = gmp[-i,"pop"], Y=gmp[-i,"pcgmp"]))
    jackknifed.gmp <- rbind(jackknifed.gmp, m)
  }
  y0 <- jackknifed.gmp$final_guess_y0
  a <- jackknifed.gmp$final_guess_a
  n <- nrow(jackknifed.gmp)
  y0 <- sqrt(sum((y0 - mean(y0))^2)*(n-1)/n)
  a <- sqrt(sum((a - mean(a))^2)*(n-1)/n)
  v <- c(y0, a)
  return(v)
}
```