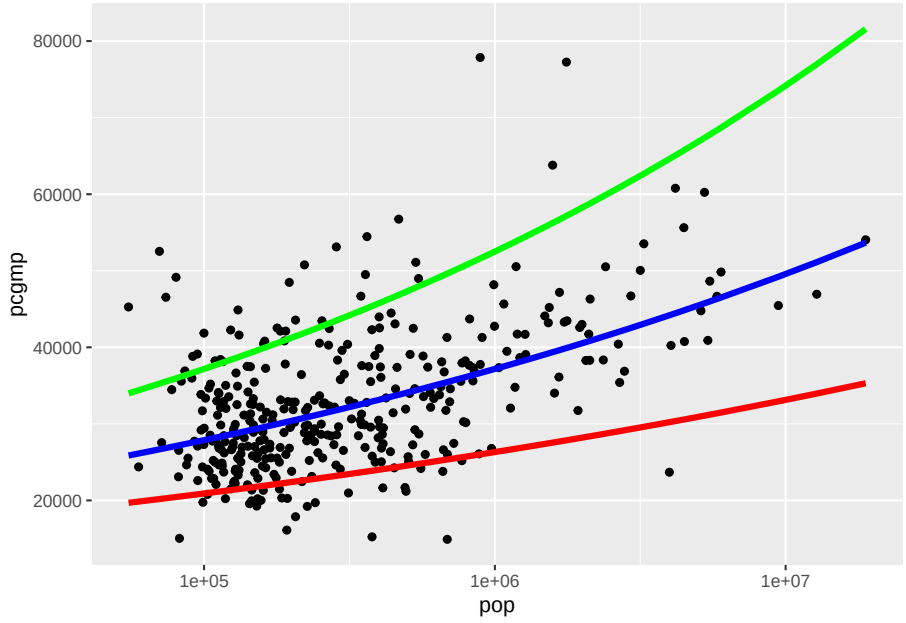
homework3

3170102187　　孙晨旭

```
gmp <- read.table("~/github/Rcourse_1/data/gmp.dat")
gmp$pop <- round(gmp$gmp/gmp$pcgmp)
gmp <- gmp %>% mutate(lmfit_1 = 6611*pop^0.125,
                      lmfit_2 = 6611*pop^0.1,
                      lmfit_3 = 6611*pop^0.15)
```

1.

```
gmp %>% ggplot(aes(x = pop,y = pcgmp))+
  geom_point()+
  geom_line(aes(x = pop,y = lmfit_1),
            col = 'blue',size = 1.5)+
  geom_line(aes(x = pop,y = lmfit_2),
            col = 'red',size = 1.5)+
  geom_line(aes(x = pop,y = lmfit_3),
            col = 'green',size = 1.5)+
  scale_x_log10()
```



2.

```
mse <- function(parameter,POP = gmp$pop,PCGMP = gmp$pcgmp){
  re<-mean((PCGMP - parameter[1]*POP^parameter[2])^2)
  return(re)
}
```

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

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

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

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

4.

```
head(nlm(mse,c(6611,0.15)),2)
```

```
## $minimum
```

```
## [1] 61857060
##
## $estimate
## [1] 6610.9999997    0.1263182
```

```
head(nlm(mse,c(6611,0.125)),2)
```

```
## $minimum
## [1] 61857060
##
## $estimate
## [1] 6611.0000000    0.1263177
```

```
head(nlm(mse,c(6611,0.1)),2)
```

```
## $minimum
## [1] 61857060
##
## $estimate
## [1] 6611.0000003    0.1263177
```

minimun 和 estimate 分别返回函数的最小值与最小值点（估计），即 minimum $=$ mse(estimate)。对上述三点，大都返回 $y_0 \approx 6611, a \approx 0.12631$，对较远的初值，往往返回不同的结果，如

```
head(nlm(mse,c(6600,0.15)),2)
```

```
## $minimum
## [1] 61856513
##
## $estimate
## [1] 6599.9999997    0.1264455
```

```
head(nlm(mse,c(6611,1)),2)
```

```
## $minimum
## [1] 1168662933
##
## $estimate
## [1] 6610.9986 -144.2974
```

5.

```
plm <- function(init,N = gmp$pop,Y = gmp$pcgmp){
  temp <- nlm(mse,init,POP = N,PCGMP = Y)
  re_list <-list(temp$estimate,temp$minimum)
  return(re_list)
}
```

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

```
## [[1]]
## [1] 6610.9999997    0.1263182
##
## [[2]]
## [1] 61857060
```

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

```
## [[1]]
## [1] 5000.0000008    0.1475913
##
## [[2]]
## [1] 62521484
```

两个迭代结果有显著不同，其中前者函数值更小。根据 nlm 函数描述，采用了 Newton-type 算法，因此猜测结果不同的原因是迭代终止于迭代初值附近的某个局部极小值点。

6. a.

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

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

```r
(se <- sd(gmp$pcgmp)/sqrt(nrow(gmp)))
```

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

b.

```r
jackknife.omit <- function(i,data = gmp$pcgmp){
  return(mean(data[-i]))
  }
```

c.

```r
jackknife.mean <- function(data = gmp$pcgmp){
  re <- vector(length = length(data))
  for(i in 1:length(data)){
    re[i] <- jackknife.omit(i)
  }
  return(re)
}
```

d.

```r
jackknift.var <- var(jackknife.mean())*365^2/366
(jackknife.se <- sqrt(jackknift.var))
```

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

```r
all.equal(se,jackknife.se)
```

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

7.

```
plm.jackknife <-function(init,POP = gmp$pop,PCGMP = gmp$pcgmp){
  # ncoefs = 2
  n <- length(POP)
  J <- matrix(0,nrow = n, ncol = 2)
  for(i in 1:n){
    new.coef <- plm(init,N = POP[-i],Y = PCGMP[-i])[[1]]
    J[i,] <- new.coef
  }
  J.var <- apply(J,2,var)
  J.se <- sqrt(((n-1)^2/n)*J.var)
  return(J.se)
}
```

```
plm.jackknife(c(6611,0.125))
```

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

```
plm.jackknife(c(5000,0.125))
```

```
## [1] 1.783567e-08 9.979823e-04
```

应用 plm 处理去除一行的数据时，总是回归到相同的值，也就是说每次去除的元素对 plm 没什么影响。因此，我的理解是，对不同抽样得到的参数都一样，方差和标准误差都为 0。

8.

```
gmp_2013 <- read.table("~/github/Rcourse_1/data/gmp-2013.dat")
gmp_2013$pop <- round(gmp_2013$gmp/gmp_2013$pcgmp)
plm(c(6611,0.1),N = gmp_2013$pop,Y = gmp_2013$pcgmp)
```

```
## [[1]]
## [1] 6611.0000005    0.1433688
##
## [[2]]
## [1] 135210524
```

```
plm.jackknife(c(6611,0.125),POP = gmp_2013$pop,PCGMP = gmp_2013$pcgmp)
```

```
## [1] 2.692652e-08 1.098548e-03
```

对比 2006 的数据，2013 有明显的不同。