Hw2

2023-04-01

the document is by R11323019 .

1.4.2 Let's draw just N = 2, use optim function to get maximum likelihood estimates.

```
library ( stats )
library ( tidyverse )
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
optimfunc <- function ( param ) {
  sig1 <-param [1]
  sig2 <-param [2]

  eplison1 = rnorm (2 ,0 , sig1 ^2) #N=2
  eplison2 = rnorm (2 ,0 , sig2 ^2)
  y = eplison1 + eplison2
  log( sig1 ^2+ sig2 ^2) +log (2*pi ) +(1 /(2*( sig1 ^2+ sig2 ^2) ) )*sum ( y ^2)
}

param <-c(2 ,1) # initial values
optimfunc.result <-optim ( param , optimfunc , method ="L-BFGS-B", hessian = TRUE )
optimfunc.result$par
```

```
## [1] 2.000048 1.001957
```

3.2.1 Pick you favorite 1, 2   0. Go back to your simulation in the first homework and re-simulate the model.

```
## set value

mu0 <- 10
mu1 <- 9
var0 <- 1
```

```r
var1 <- 2
var01 <- 0.6
beta1 <- 1
beta2 <- 2
```

```r
sigmav <-sqrt ( var0 ^2+ var1 ^2 -2* var01 )
rho <- var01 /( var0 * var1 )
sigma <-rbind (c( var0 ^2 , var01 ) ,c( var01 , var1 ^2) )
```

```r
library('dplyr')
library('data.table')
```

```
##
##    'data.table'
##       'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##       'package:dplyr':
##
##     between, first, last
##       'package:purrr':
##
##     transpose
```

```r
library(MASS)
```

```
##
##    'MASS'
##       'package:dplyr':
##
##     select
```

```r
simulation <- data.table(
  mvrnorm ( n =10^7 , mu = c(0,0) , Sigma = sigma),
  x1 =  rnorm(10^7, 1, var0),
  x2 =  rnorm(10^7, 1, var1)
)

colnames ( simulation ) <-c("eplison0", "eplison1", "x1", "x2")
```

```r
## create W0 and W1
# W0 = u0 + beta1*x1 + eplison0
# w1 = u1 + beta1*x1+  beta2*x2 eplison1

simulation[, w0:= mu0 + beta1*x1 +eplison0]
simulation[, w1:= mu1 + beta1*x1 + beta2*x2 + eplison1]
```

```r
simulation[, .(mean(w0),mean(w1),mean(x1),mean(x2))]
```

```
##          V1       V2        V3        V4
## 1: 10.99919 11.99813 0.9995718 0.9997387
```

3.2.5 Create a column in your simulated data for the estimated propensity score using the derived formula above.

```
# create True D
simulation[w1>w0, D:= 1]
simulation[w1<=w0, D:= 0]

#create p(x)
simulation[, propensity_score_formula := 1 - pnorm (( mu0 - mu1 - beta2*x2 )/ sigmav ,0 ,1)]
```

3.2.6 Use logit to estimate the propensity score

```
logit <-glm(D~x2 , data = simulation , family = binomial ( link ="logit") )

simulation[, propensity_score_logit := predict(logit, type ="response")]
```

3.2.7 What is the correlation coefficient of the above two types of propensity scores?

```
cor ( simulation$propensity_score_formula ,simulation$propensity_score_logit )
```

```
## [1] 0.9997168
```

3.2.8 Use both types of propensity score to conduct IPW estimates.

```
# create weight by formula
simulation[D==1, weight_formula := 1/propensity_score_formula]
simulation[D==0, weight_formula:= 1/(1-propensity_score_formula)]

# create weight by logit setimate
simulation[D==1, weight_logit := 1/propensity_score_logit]
simulation[D==0, weight_logit:= 1/(1-propensity_score_logit)]

#create actual wage
simulation[D==1, wage := w1]
simulation[D==0, wage := w0]
```

```
ipw_formula <-lm( wage ~D, weights = weight_formula , data = simulation )
summary ( ipw_formula )
```

```
##
## Call:
## lm(formula = wage ~ D, data = simulation, weights = weight_formula)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -2339.15    -1.24     0.31     2.22   504.15
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 11.288593   0.001111   10163   <2e-16 ***
## D            2.203875   0.001551    1421   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.461 on 9999998 degrees of freedom
## Multiple R-squared:  0.1679, Adjusted R-squared:  0.1679
## F-statistic: 2.018e+06 on 1 and 9999998 DF,  p-value: < 2.2e-16
```

```
ipw_formula$coefficients
```

```
## (Intercept)           D
##   11.288593    2.203875
```

```
ipw_logit <-lm( wage~D, weights = weight_logit , data = simulation )
summary(ipw_logit)
```

```
##
## Call:
## lm(formula = wage ~ D, data = simulation, weights = weight_logit)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -249.997   -1.342    0.226    2.055   98.234
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.212709   0.001117   10040   <2e-16 ***
## D            2.572453   0.001540    1670   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.251 on 9999998 degrees of freedom
## Multiple R-squared:  0.218,  Adjusted R-squared:  0.218
## F-statistic: 2.789e+06 on 1 and 9999998 DF,  p-value: < 2.2e-16
```

```
ipw_logit$coefficients
```

```
## (Intercept)           D
##   11.212709    2.572453
```

3.2.9 regress wi on Xi.

```
wage_to_D_OLS <-lm(wage ~D, data = simulation )
summary (wage_to_D_OLS)
```

```
##
## Call:
## lm(formula = wage ~ D, data = simulation)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9813  -1.6317  -0.1572   1.3332  20.4498
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.083347   0.001270    8725   <2e-16 ***
## D            3.796401   0.001655    2293   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.576 on 9999998 degrees of freedom
## Multiple R-squared:  0.3446, Adjusted R-squared:  0.3446
## F-statistic: 5.259e+06 on 1 and 9999998 DF,  p-value: < 2.2e-16
```

```
wage_to_D_OLS$ coefficients
```

```
## (Intercept)           D
##   11.083347    3.796401
```

3.2.10 Now estimate by adding “control variables.

```
wage_to_D_and_x2_OLS <-lm( wage~D+x2 , data = simulation )
summary ( wage_to_D_and_x2_OLS )
```

```
##
## Call:
## lm(formula = wage ~ D + x2, data = simulation)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1300  -1.4624  -0.1043   1.3513  13.5562
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.181e+01  1.098e-03 10756.6   <2e-16 ***
## D           7.938e-01  1.938e-03   409.6   <2e-16 ***
## x2          1.039e+00  4.769e-04  2179.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.121 on 9999997 degrees of freedom
## Multiple R-squared:  0.5556, Adjusted R-squared:  0.5556
## F-statistic: 6.252e+06 on 2 and 9999997 DF,  p-value: < 2.2e-16
```

```
wage_to_D_and_x2_OLS$ coefficients
```

```
## (Intercept)           D          x2
##  11.8121076   0.7938214   1.0393001
```