Econometrics-4-Wei

Wei Ye[*]

March 14,2022

a.

The code associated with a is in below:

```
set.seed(1)
N=100
x=rnorm(N)
b0=1
b1=0
u=rnorm(N)
m=exp(b0+b1*x)
y=m+u
```

b.

```
theta_int=cbind(0.5,0)
source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solution
source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solution
source('~/Dropbox/My Mac (Wei's MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solution
out1=qderivfun(y,x,t(theta_int))
ssum=out1$ssum
out2=qderivfun2(y,x,t(theta_int))
hsum=out2$H
QgPlusOne=t(theta_int)-solve(hsum)%*%ssum
QgPlusOne
```

```
##              [,1]
## [1,]  2.680596909
## [2,] -0.001060814
```

c.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

[*] First year Phd Student in Economics at Fordham University

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
thetastore=matrix(0,nrow=2,ncol=11)
thetastore[,1]=c(0.5,0)
d=matrix(0,nrow=2,ncol=10)
for (i in 1:10){
  out1=qderivfun(y,x,thetastore[,i])
  ssum=out1$ssum
  out2=qderivfun2(y,x,thetastore[,i])
  hsum=out2$H
  thetastore[,i+1]=thetastore[,i]-solve(hsum)%*% ssum
  d[,i]=abs(thetastore[,i+1]-thetastore[,i])
}

print(thetastore)
```

```
##      [,1]         [,2]         [,3]         [,4]         [,5]         [,6]
## [1,]  0.5  2.680596909  2.680589215  2.680589215  2.680589215  2.680589215
## [2,]  0.0 -0.001060814 -0.001060386 -0.001060386 -0.001060386 -0.001060386
##              [,7]         [,8]         [,9]        [,10]        [,11]
## [1,]  2.680589215  2.680589215  2.680589215  2.680589215  2.680589215
## [2,] -0.001060386 -0.001060386 -0.001060386 -0.001060386 -0.001060386
```

```
print("\n")
```

```
## [1] "\n"
```

```
print(d)
```

```
##             [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 2.180596909 7.694387e-06 1.400213e-12 1.211919e-12 8.197887e-13
## [2,] 0.001060814 4.279194e-07 1.415500e-12 2.639066e-12 1.992350e-12
##              [,6]         [,7]         [,8]         [,9]        [,10]
## [1,] 8.659740e-13 8.379963e-13 1.403766e-12 5.413447e-13 1.334932e-12
## [2,] 1.440198e-13 3.642193e-12 4.447252e-12 1.068699e-12 1.700681e-12
```

```
print("\n")
```

```
## [1] "\n"
```

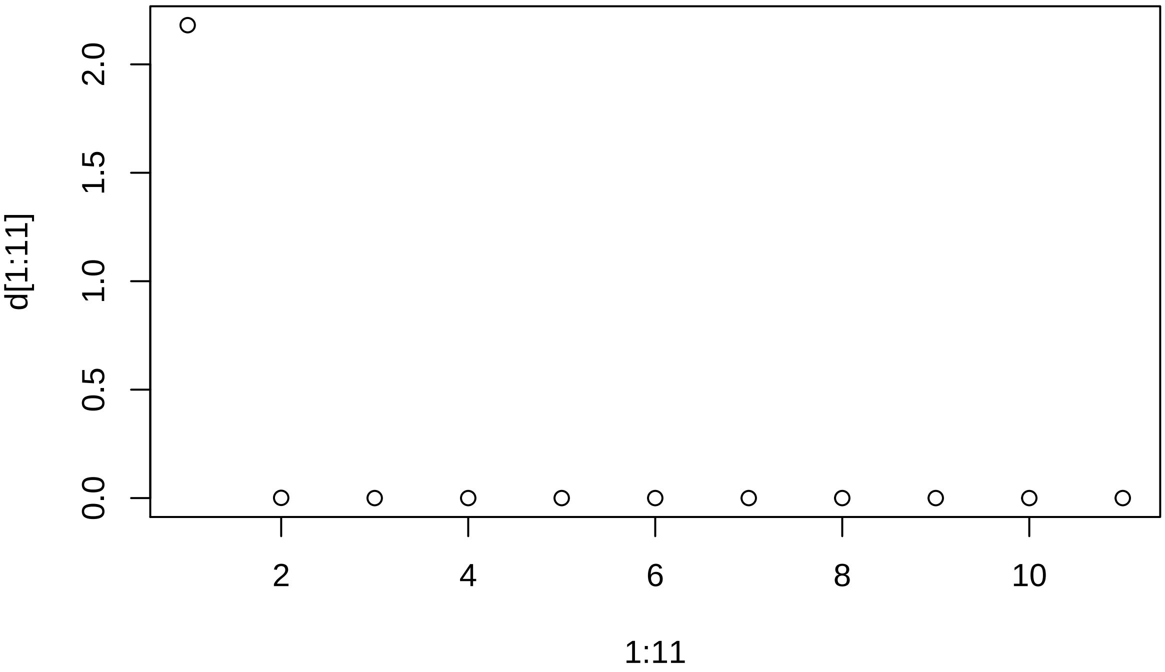
```
print(max(d))
```

```
## [1] 2.180597
```

From the d value matrix you can see above, the last couple of colums' power is -12, it's extreme number, so it will converge.

Plot as below:

```
plot(1:11,d[1:11])
```



d:Use NLS method to estimate the model:

```
nlsout=nls(y~exp(b0+b1*x),start = list(b0=1,b1=0))
summary(nlsout)
```

```
##
## Formula: y ~ exp(b0 + b1 * x)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## b0  0.986034   0.036180  27.254   <2e-16 ***
## b1 -0.000368   0.040189  -0.009    0.993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
##
## Number of iterations to convergence: 2
## Achieved convergence tolerance: 5.255e-06
```

```
#Here the result is homoskedasticity, we dig into more to make some robust test.
```

```
X1=cbind(rep(1,100),x)
o=optim(cbind(1,0),fn=qfunction,x=X1,y=y,method="BFGS")
o
```

```
## $par
##          [,1]         [,2]
## [1,] 2.680591 -0.001060647
##
## $value
## [1] 0.9083561
##
```

```
## $counts
## function gradient
##       19        5
##
## $convergence
## [1] 0
##
## $message
## NULL
```

From the result above, we can get $o\$par[,1] = 2.680591$ and $o\$par[,2] = -0.001060647$, which is the same in roud 5 digits with the result we get from QgPlusOne in b.

e. Build R script and test θ^g in our R script.

```
##This chunk is codes for 'nrmethod.R', in this case i don't need to attach my R file anymore.
##Created by Wei Ye
##Date: March 14, 2022
#Note: I don't treat $\theta$ as vectors, but as matrix for my own convenience.
#source('~/Dropbox/My Mac (Wei’s MacBook Air)/Downloads/PhD-Coursework/22Spring/Econometrics/My_solutio
```

Test for $\theta^{\{g\}} = [.5, 0]'$

Note: I suffered from an error below in comment. I will check this tomorrow or in a few days, and to test it later. I have attached my nrmethod.R file in the email.

```
#I comment this chunk because after implementing nrmethod to this chunk,
#there are erros: 'Only first element is used', and it follows 10 times.
#I didn't find a method to solve this at the moment. My initial suspect is due to
#my while loop bug in "nrmethod.R".

#thetatest1=matrix(0,nrow=2,ncol=100)
#thetatest1[,1]=c(0.5,0)
#nrmethod(y,x,thetatest1)
```

```
##thetatest2=matrix(0,nrow=2,ncol=100)
#thetatest2[,1]=c(-2,0)
#nrmethod(y,x,thetatest2)
```