Advanced Microeconometrics: Problem Set 2

Luis Martínez Valdés

Spring 2022

This set of questions is inspired by Rau, Sánchez and Urzúa (2019). The file *data.csv* contains real data for Chilean individuals, that track their schooling and labor market decisions and outcomes from their early teens through their late twenties.

The goal is to estimate a generalized Roy model for the decision to attend a private high school in lieu of a public high school, and labor market outcomes, i.e. wages. The model includes an unobserved factor that approximates a combination of individuals' scholastic abilities.

A description of the variables in the data is as follows:

- *test_X* is students' performance on standardized test X. The unit measure is standard deviations.
- *privateHS* is a dummy indicating having attended a private high school.
- *wage* is the natural log of wage.
- *male*, *momschoolingX*, *dadschoolingX*, *broken_homeX*, *incomehhX*, *north*, *center* are demographic variables.
- *share_private* and *avg_price* are instruments for the decision of attending a private high school, and denote the local share of private high schools and the average fees local private high schools charge.

Formally, the model includes potential outcomes as follows,

$$\begin{aligned} Y_1 &= X\beta_1 + \theta\alpha_1 + U_1 \\ Y_0 &= X\beta_0 + \theta\alpha_0 + U_0, \end{aligned}$$

where Y_1 is the potential outcome (i.e. wages) in the counterfactual of attending a private high school, and Y_0 is similarly defined for the counterfactual of attending a public high school. All relevant observable demographics are included in X, while θ is the unobserved one-dimensional factor (i.e. ability) determining labor market outcomes. The unobserved factor is normally distributed with mean zero and standard deviation σ_θ. The terms U_1 and U_0 are idiosyncratic error terms, that are normally distributed with mean zero and standard deviations σ_1 and σ_0, respectively.

Individuals decide whether or not to attend a private high school based on a latent variable I:

$$I = Z\gamma + \theta\alpha_I + V,$$

where Z include observable demographics and instruments, and V is an idiosyncratic error term with zero mean and unit variance. Note that the unobservable factor is also present in this part of the model. We can thus define a binary variable D indicating treatment status,

$$D = \mathbb{I}\Big[I \geq 0\Big].$$

The model includes a measurement system, that helps with the identification of the distribution of θ. Specifically,

$$T_k = W\omega_k + \theta\alpha_{T_k} + \varepsilon_k \qquad \text{for each } k = 1, 2, 3, 4,$$

where T_k is test score k, W include demographics determining test scores, and ε_k is a normally distributed error term with mean zero and standard deviation σ_{ε_k}.

Finally, we assume that the error terms in the model are all independent from each other conditioning on the observables and the unobserved factor, i.e. $U_1 \perp\!\!\!\perp U_0 \perp\!\!\!\perp V \perp\!\!\!\perp \varepsilon \mid X, Z, W, \theta$, and θ is independent of all observables.

For the empirical implementation of the model, in X include *male*, *north*, and *center*. In Z include all variables in X plus *share_private* and *avg_price*. In W include *male*, *momschoolingX*, *dadschoolingX*, *broken_homeX*, *incomehhX*, *north*, and *center*. The outcome variable is *wage*. D is *privateHS*. And, the measurement system is comprised by the four test scores *test_X*.

```
# Reading data
data <- read.csv('/Volumes/External/AdvMicroEconMetrics/PS_2/data_ps2.csv')

# X (Standarized Test)
X <- data[names(data) %in% c('male', 'north', 'center')]

# Z (Observable demographics and instruments)
Z <- data[names(data) %in% c('male', 'north', 'center', 'share_private',
                             'avg_price')]

# W (demographics determining test scores)

W <- data[names(data) %in% c('male', 'momschooling2', 'momschooling3',
                             'momschooling_miss','dadschooling2',
                             'dadschooling3', 'dadschooling_miss',
                             'broken_home1', 'broken_home_miss',
                             'incomehh2', 'incomehh3', 'incomehh_miss',
                             'north', 'center')]
# D (Treatment Status)
D <- data[names(data) %in% c('privateHS')]

# Y (Model)
Y <- data[names(data) %in% c('wage')]
```

1.

Normalize $\alpha_{T_1} = 1$. Discuss and show a sketch of identification of the remaining loadings in the measurement system (i.e. α_{T_2}, α_{T_3}, α_{T_4}) and of the distribution of θ. Why do we need to assume that one of the loadings in the measurement system is equal to one?

Answer

Following the line of thought in Heckman, et al. (2003) and taking into account that we have four test scores (T_1,T_2, T_3, T_4) available. We have,

$$T_i = W\omega_k + \alpha_{T_i}\theta + \varepsilon_{T_i} \quad \forall \quad i = 1, \ldots, 4$$

Then,assuming independence between demographic determining scores we can get rid of the $W\omega_k$ components when computing the covariances. Thus,

$$
\begin{aligned}
Cov(T_1, T_2) &= \alpha_{T_1}\alpha_{T_2}\sigma^2_\theta \\
Cov(T_1, T_3) &= \alpha_{T_1}\alpha_{T_3}\sigma^2_\theta \\
Cov(T_1, T_4) &= \alpha_{T_1}\alpha_{T_4}\sigma^2_\theta \\
Cov(T_2, T_3) &= \alpha_{T_2}\alpha_{T_3}\sigma^2_\theta \\
Cov(T_2, T_4) &= \alpha_{T_2}\alpha_{T_4}\sigma^2_\theta \\
Cov(T_3, T_4) &= \alpha_{T_3}\alpha_{T_4}\sigma^2_\theta
\end{aligned}
$$

Taking the left hand side, we have that,

$$
\begin{aligned}
\frac{Cov(T_2, T_4)}{Cov(T_1, T_4)} &= \frac{\alpha_{T_2}}{\alpha_{T_1}} \\
\frac{Cov(T_2, T_3)}{Cov(T_1, T_2)} &= \frac{\alpha_{T_3}}{\alpha_{T_1}} \\
\frac{Cov(T_2, T_4)}{Cov(T_1, T_2)} &= \frac{\alpha_{T_4}}{\alpha_{T_1}}
\end{aligned}
$$

As we are normalizing $\alpha_{T_1} = 1$, we can obtain the values for the rest of the loaings in the measuring system, such as

$$
\frac{T_i}{\alpha_{T_i}} = \theta + \frac{\varepsilon_{T_i}}{\alpha_{T_i}} = \theta + \varepsilon^*_{T_i} \quad \forall \quad i = 1, \ .. \ , 4
$$

where $\varepsilon^*_{T_i} = \varepsilon_{T_i}/\alpha_{T_i}$. Thus, we can compute the densities (using Kotlarski's Theorem) for $\varepsilon_{T_i} \quad \forall \quad i = 2, 3, 4$ and θ. On the other hand, assuming zero normal distributions for the error terms, and approximating the distribution of the factor using a mixture of two normal distributions using simulated entries for the parameters. That is,

$$
\theta = \quad p\mathcal{N}(\mu_1, \sigma^2_1) + (1-p)\mathcal{N}(\mu_2, \sigma^2_2)
$$

Therefore,

```
set.seed(180618)
# Approximate Theta
sims<- length(data$test_lect)
p<- runif(sims,0,1)
mu1<- runif(1); mu2<- runif(1)-1
var1<- runif(1,1,2); var2<- runif(1,2,3)
N1<- rnorm(sims,mu1,var1); N2<- rnorm(sims,mu2, var2)

th<- p * N1 + (1-p) * N2

# Create new data frame
T1<- data$test_lect
T2<- data$test_mate
T3<- data$test_soc
T4<- data$test_nat



# Calculate remaining loadings
(alpha_T1<- 1)
```

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

```
(alpha_T2<- cov(T2,T4) / cov(T1,T4))
```

```
## [1] 0.9178072
```

```
(alpha_T3<- cov(T2,T3) / cov(T1,T2))
```

```
## [1] 0.8976903
```

```
(alpha_T4<- cov(T2,T4) / cov(T1,T2))
```

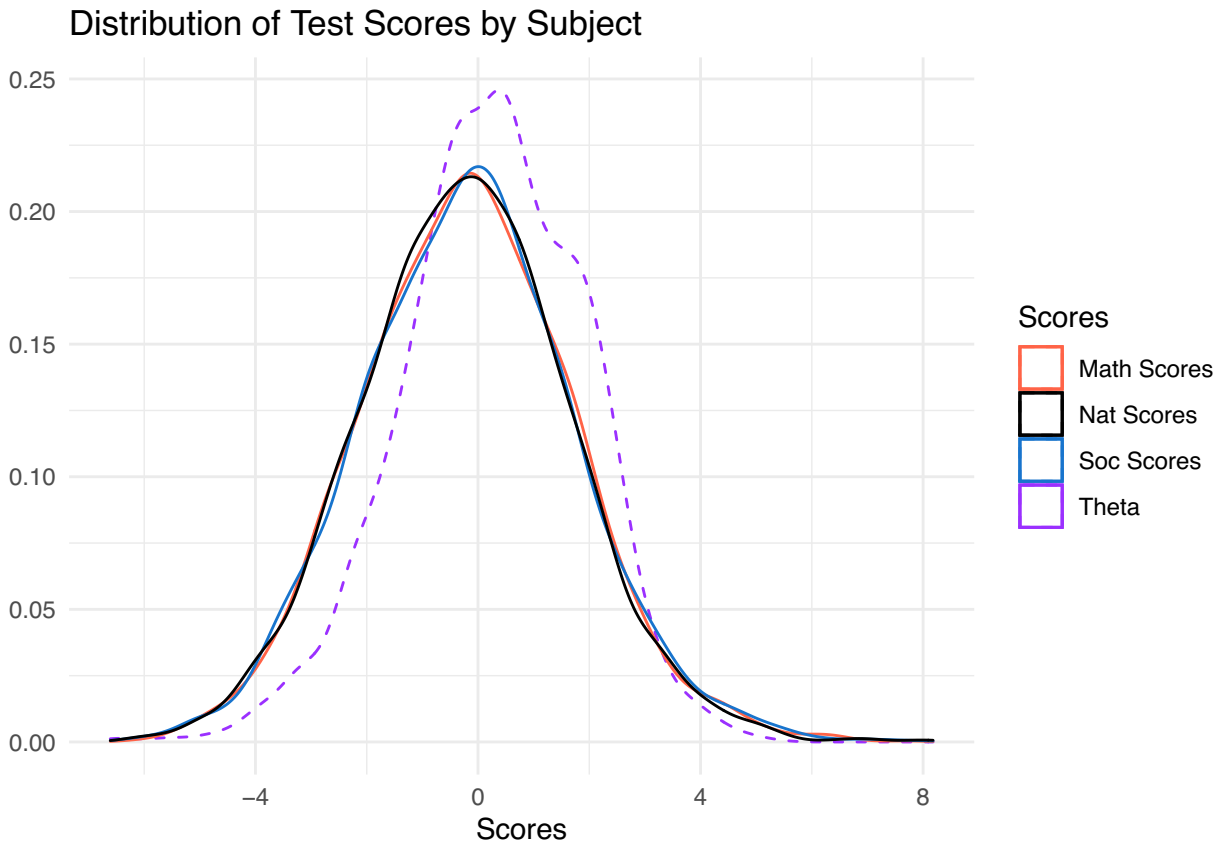
```
## [1] 0.9663841
```

```
# Calculate densities
vareps_T2<- (T2 / alpha_T2) - th
vareps_T3<- (T3 / alpha_T3) - th
vareps_T4<- (T4 / alpha_T4) - th

# Generating new data frame
newdata<- cbind(th, vareps_T2, vareps_T3, vareps_T4) %>% as.data.frame()

colours <-
  c(
    'Theta' = 'purple1',
    'Math Scores' = 'tomato',
    'Soc Scores' = 'dodgerblue3',
    'Nat Scores' = 'black'
  )


ggplot(newdata) +
    geom_density(aes(x = th, y = ..density.. ,
                  colour = 'Theta'),linetype=2)+
    geom_density(aes(x = vareps_T2, y = ..density.. ,
                  colour = 'Math Scores'))+
  geom_density(aes(x = vareps_T3, y = ..density..,
                  colour = 'Soc Scores'))+
   geom_density(aes(x = vareps_T4, y = ..density..,
                  colour = 'Nat Scores'))+
  labs(
        title = 'Distribution of Test Scores by Subject',
        x = 'Scores',
        y = NULL
        )+
  scale_colour_manual(name = 'Scores', values = colours)+
  theme_minimal()
```



Finallly, the assumption that one of the loadings in the measuring system is equal to one is necessary in order to compute the other ones; otherwise, the system of equations wouldn't have a solution, as you have more unknowns than equations.

2.

Run an OLS regression of Y $(= DY_1 + (1-D)Y_0)$ on D and X. Show your results. Are they biased? Why?

Answer

```
# Implementing OLS regression of Y on D and X we have

model_OLS<- lm(wage ~ male + north + center + privateHS, data = data)
summary(model_OLS)
```

```
##
## Call:
## lm(formula = wage ~ male + north + center + privateHS, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6353 -0.4254  0.2473  0.7468  2.4423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.75885    0.05362 144.689  < 2e-16 ***
## male         0.29620    0.04332   6.837 9.86e-12 ***
```

```
## north        0.48118    0.10199   4.718 2.50e-06 ***
## center       0.25462    0.05103   4.990 6.41e-07 ***
## privateHS    0.12157    0.04527   2.685  0.00729 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.151 on 2834 degrees of freedom
## Multiple R-squared:  0.03264,    Adjusted R-squared:  0.03128
## F-statistic: 23.91 on 4 and 2834 DF,  p-value: < 2.2e-16
```

```
# Another Way: Simulating all parameters, using multiple linear regression and
# estimating model
set.seed(180618)
alpha0<- runif(1); alpha1<- runif(1); beta0<- runif(1); beta1<- runif(1)
sigma1<- runif(1)
sigma2<- runif(1)
U0<- rnorm(sims, 0, sigma1)
U1<- rnorm(sims, 0, sigma2)
V<- rnorm(sims, 0, 1)

Z.matrix <- as.matrix.data.frame(Z)
X.matrix<- as.matrix.data.frame(X)
Gamma<- runif(5) %>% as.matrix()

#Using normalized loading in order to simplify instruments
I<- (Z.matrix %*% Gamma) + th * alpha_T1 + V
D.prime<- ifelse(I >= 0, 1,0) %>% as.matrix.data.frame()

#Generate Potential Outcomes
Y1<- X * beta1 + th * alpha1 + U1 %>% as.matrix()
Y0<- X * beta0 + th * alpha0 + U0 %>% as.matrix()


#Generate Y= DY1 + (1-D)Y0
Y.prime<- D.prime * Y1 + (1 - D.prime) * Y0
y.prime<- as.matrix.data.frame(Y.prime)


(OLS_X<- solve(t(X.matrix) %*% X.matrix) %*% t(X.matrix) %*% y.prime)
```

```
##               male       north      center
## male    0.38507103  0.01401893  0.02023693
## north  -0.00767478  0.38301830 -0.02734755
## center -0.05766574 -0.05810669  0.30612437
```

```
(OLS_D<- solve(t(D.prime) %*% D.prime) %*% t(D.prime) %*% y.prime)
```

```
##           male     north    center
## [1,] 0.4306762 0.2121851 0.5158149
```

We know that OLS estimators are BLUEs (Best Unbiased Linear Estimators) therefore, we would expect that our fitted model coefficients are unbiased. Now this is true every time the assumptions are upheld; however, there is a clear endogeinity problem in our case, as our error terms contain part of the 'unobserved abilities'.

Note* : Our estimations using simulated data differ in some manner (~10%) from the fitted model. Why? This is actually an interesting question: As we simulate, we lose efficiency and accuracy, which we can directly

obtain from the data. This is actually part of the endogeinity problem mentioned earlier, when we simulate the parameters $(\alpha_i, beta_i, U_i, V)$ we are not internalizing the unobserved abilities in error terms (losing data).

3.

Using Z as instruments, estimate by 2SLS the "effect'' of D on Y. Show and comment your results.

Answer

```
# First Stage: Estimate D using Z as instrument
D.matrix<- as.matrix.data.frame(D)
D.hat<- Z.matrix %*% solve(t(Z.matrix) %*% Z.matrix) %*% t(Z.matrix) %*%
        D.matrix

data_2sls<- cbind.data.frame(data,D.hat)
#Second Stage: Run regression on X and D.hat
s2_2sls<- lm(wage ~ D.hat, data = data_2sls)
summary(s2_2sls)
```

```
##
## Call:
## lm(formula = wage ~ D.hat, data = data_2sls)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8760 -0.4043  0.2369  0.7341  2.4988
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.76267    0.07282 106.603  < 2e-16 ***
## D.hat        0.71430    0.11593   6.161 8.23e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.162 on 2837 degrees of freedom
## Multiple R-squared:  0.0132, Adjusted R-squared:  0.01286
## F-statistic: 37.96 on 1 and 2837 DF,  p-value: 8.228e-10
```

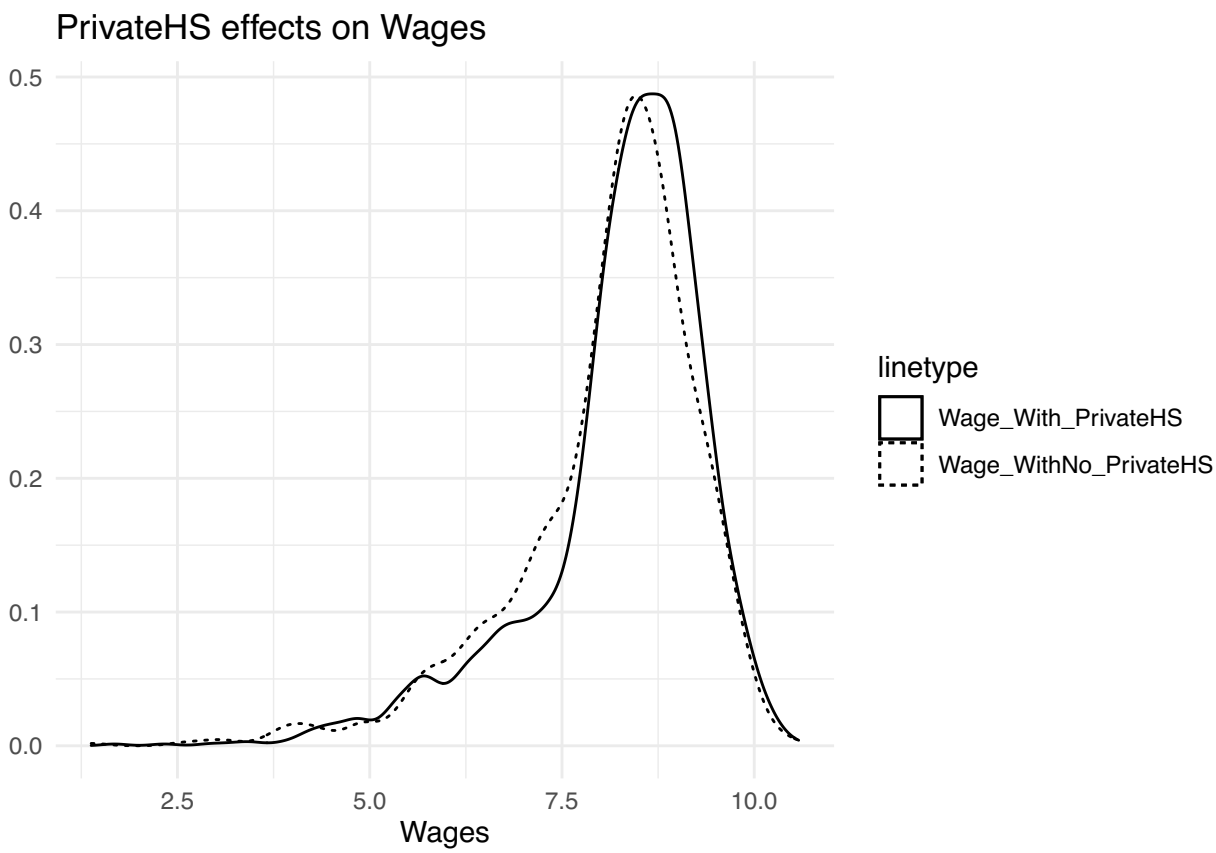
To see the effects more clearly,

```
# Filter out wage by treated and utreated
wage.treated<- data %>% filter(privateHS == 1) %>% pull(wage) %>%
               as.data.frame()
colnames(wage.treated)<- c('Wage_With_PrivateHS')

wage.untreated<- data %>% filter(privateHS == 0) %>% pull(wage) %>%
                 as.data.frame()
colnames(wage.untreated)<- c('Wage_WithNo_PrivateHS')

linetypes<- c(
  'Wage_With_PrivateHS' = 1,
  'Wage_WithNo_PrivateHS' = 2
)
ggplot()+
```

```
  geom_density(data= wage.treated, aes(x=Wage_With_PrivateHS, y= ..density..,
               linetype = 'Wage_With_PrivateHS'))+
  geom_density(data= wage.untreated, aes(x=Wage_WithNo_PrivateHS,
                                         y= ..density..,
              linetype= 'Wage_WithNo_PrivateHS' ))+
  labs(
        title = 'PrivateHS effects on Wages',
        x = 'Wages',
        y = NULL
        )+
  theme_minimal()
```



4.

Now, run your OLS regression in 2. but adding T_1–T_4 as additional controls. What is the effect of D on Y?

Answer

```
data_q4<- data
colnames(data_q4)[c(1,2,3,4)]<- c('T1', 'T2', 'T3', 'T4')

# Implementing OLS regression of Y on D and X we have

model_OLS_Q4<- lm(wage ~ T1 + T2 + T3 + T4 + male + north + center + privateHS,
               data = data_q4)
```

```
summary(model_OLS_Q4)
```

```
##
## Call:
## lm(formula = wage ~ T1 + T2 + T3 + T4 + male + north + center +
##     privateHS, data = data_q4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5270 -0.4147  0.2607  0.7434  2.3179
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.75909    0.05374 144.381  < 2e-16 ***
## T1           0.02457    0.03557   0.691   0.4898
## T2           0.08600    0.03364   2.557   0.0106 *
## T3          -0.02924    0.03234  -0.904   0.3660
## T4           0.02231    0.03335   0.669   0.5036
## male         0.29951    0.04474   6.694 2.60e-11 ***
## north        0.46606    0.10179   4.579 4.88e-06 ***
## center       0.25163    0.05094   4.940 8.27e-07 ***
## privateHS    0.11297    0.04524   2.497   0.0126 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.148 on 2830 degrees of freedom
## Multiple R-squared:  0.03915,    Adjusted R-squared:  0.03643
## F-statistic: 14.41 on 8 and 2830 DF,  p-value: < 2.2e-16
```

Now, it is interesting to see that when we add test scores as controls, -privateHS- estimate decreases significantly, this means the correlation between schooling level (treatment) and wages flattens. As to the effect, it seems that unobserved factor plays a more important role in determining the level of earning an individual will have. Thus, the effect of D on Y has less impact in this case. If we were to replicate the effect graphic like in the previous question, the dashed line would shift to the right, closing in to the solid one.

5.

Define the MTE, ATE, and TT parameters in this framework.

Answer

- The average treatment effect (ATE) is defined by averaging the treatment gains over the entire student population,

$$ATE = \int\int \mathbb{E}\Big[Y_1 - Y_0 | X = x, \theta = \hat{\theta}\Big] dF_{X,\theta}(x, \hat{\theta})$$

- The treatment effect on the treated (TT) is defined by averaging the treatment gains over the subset of students that actually choose to be treated,

$$TT = \int\int \mathbb{E}\Big[Y_1 - Y_0 | X = x, \theta = \hat{\theta}, D = 1\Big] dF_{X,\theta|D=1}(x, \hat{\theta})$$

- The marginal treated effect (MTE) is defined by averaging the gain in terms of $Y_1 - Y_0$ for all students

who would be indifferent between choosing to be treated or not (i.e.$V = v$),

$$\begin{aligned} MTE &= \int\int \mathbb{E}\Big[Y_1 - Y_0 | X = x, \theta = \hat{\theta}, V = v\Big] dF_{X,\theta|V=v}(x, \hat{\theta}) \\ &= \int\int \mathbb{E}\Big[Y_1 - Y_0 | X = x, \theta = \hat{\theta}, V = Z'\gamma - \theta\alpha_I\Big] dF_{X,\theta|V=Z'\gamma-\theta\alpha_I}(x, \hat{\theta}) \end{aligned}$$

6.

Write down the likelihood function of the model, integrating out the factor θ over its distribution.

Answer

$$\begin{aligned} \mathcal{L} =& \Pi_{i=1}^{N} \int f(Y_i, D_i, T_i | Z_i, X_i, \theta) f(\theta) d(\theta) \\ =& \Pi_{i=1}^{N} \int f(Y_i, D_i | Z_i, X_i, \theta) f(T_i | X_i, \theta) f(\theta) d(\theta) \\ =& \Pi_{i=1}^{N} \int \left(\left[\frac{1}{\sigma_0\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{Y_0 - x\beta_0 - \theta\alpha_0}{\sigma_0}\Big)^2 \right\} \right] \left[\Phi\Big(\frac{Z\gamma + \theta\alpha_I}{\sigma_I}\Big) \right] \right)^{1-D} \\ & \left(\left[\frac{1}{\sigma_1\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{Y_1 - x\beta_1 - \theta\alpha_1}{\sigma_1}\Big)^2 \right\} \right] \left[1 - \Phi\Big(\frac{Z\gamma + \theta\alpha_I}{\sigma_I}\Big) \right] \right)^{D} \\ & \left[\frac{1}{\sigma_{T_1}\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{T_1 - W\omega_{T_1} - \theta\alpha_{T_1}}{\sigma_{T_1}}\Big)^2 \right\} \right] \\ & \left[\frac{1}{\sigma_{T_2}\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{T_2 - W\omega_{T_2} - \theta\alpha_{T_2}}{\sigma_{T_2}}\Big)^2 \right\} \right] \\ & \left[\frac{1}{\sigma_{T_3}\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{T_3 - W\omega_{T_3} - \theta\alpha_{T_3}}{\sigma_{T_3}}\Big)^2 \right\} \right] \\ & \left[\frac{1}{\sigma_{T_4}\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{T_4 - W\omega_{T_4} - \theta\alpha_{T_4}}{\sigma_{T_4}}\Big)^2 \right\} \right] \\ & \left[\frac{1}{\sigma_\theta\sqrt{2\pi}} exp\left\{ -\frac{1}{2}\Big(\frac{\theta}{\sigma_\theta}\Big)^2 \right\} \right] d\theta \end{aligned}$$

7.

Estimate the model by maximum likelihood. Show your estimated coefficients and corresponding standard errors. *HINT: I suggest you use a Gauss-Hermite quadrature for numerical integration; however, you are free to use any other numerical integration routine (e.g. Monte Carlo, Quasi-Monte Carlo).*

Answer

Please see Appendix: MATLAB Codes for the Maximum Likelihood Estimation. Once we obtained our results we exported them into a dataset, so we could handle it mre easily.

```
# Reading data
data_betas <-
  read.csv('/Volumes/External/AdvMicroEconMetrics/PS_2/data_betas_ps2.csv')

kable(head(data_betas,10))
```

Betas	Std..Errors
7.8192	0.0487
0.3021	0.0433
0.4764	0.1020
0.2735	0.0506
7.8192	0.0487
0.3021	0.0433
0.4763	0.1020
0.2735	0.0506
-0.4353	0.0831
0.1362	0.0487

8.

Using your estimates, simulate 10,000 observations from your model. Compare simulated and actual data averages for test scores, the decision of attending a private high school, and wages conditional on the decision of attending a private high school.

Answer

```
N<- 10000
set.seed(180618)

# Estimations
beta_0<- data_betas$Betas[1:3]
beta_1<- data_betas$Betas[4:6]
beta_D<- data_betas$Betas[7:11]
omega_T1<- data_betas$Betas[12:25]
omega_T2<- data_betas$Betas[26:39]
omega_T3<- data_betas$Betas[40:53]
omega_T4<- data_betas$Betas[54:67]
sigma_0<- abs(data_betas$Betas[68])
sigma_1<- abs(data_betas$Betas[69])
sigma_T1<- abs(data_betas$Betas[70])
sigma_T2<- abs(data_betas$Betas[71])
sigma_T3<- abs(data_betas$Betas[72])
sigma_T4<- abs(data_betas$Betas[73])
alpha_0<- data_betas$Betas[74]
alpha_1<- data_betas$Betas[75]
alpha_I<- data_betas$Betas[76]
alpha.T1<- data_betas$Betas[77]
alpha.T2<- data_betas$Betas[78]
alpha.T3<- data_betas$Betas[79]
alpha.T4<- data_betas$Betas[80]
sigma.theta<- abs(data_betas$Betas[81])
Gamma<- cbind(data_betas$Betas[82], data_betas$Betas[83], data_betas$Betas[84],
              data_betas$Betas[85], data_betas$Betas[86])

#Start Simulation
eps<-  rnorm(N)
eps1<- rnorm(N,0,sigma_T1)
eps2<- rnorm(N,0, sigma_T2)
```

```r
eps3<- rnorm(N,0, sigma_T3)
eps4<- rnorm(N,0,sigma_T4)

#idiosyncratic Error Terms
U0<- rnorm(N,0,sigma_0)
U1<- rnorm(N,0,sigma_1)
V<-  rnorm(N,0,1)

# Theta(unboserved factor) is normally distributed with mean zero and
# s.d sigma_theta
theta<- rnorm(N, 0, sigma.theta)

#Recalling
W.matrix <- as.matrix.data.frame(W)
D.matrix<- as.matrix.data.frame(D)
Y.matrix<- as.matrix.data.frame(Y)

#Simulating

X.boot <- apply(X.matrix, MARGIN = 2, function(x) sample(x, replace = TRUE,
                size = 10000)) %>% as.matrix.data.frame()
colnames(X.boot)<- c('male', 'north', 'center')

Z.boot <- apply(Z.matrix, MARGIN = 2, function(x) sample(x, replace = TRUE,
                size = 10000)) %>% as.matrix.data.frame()
colnames(Z.boot)<- c('male', 'north', 'center', 'share_private',
                             'avg_price')

W.boot <- apply(W.matrix, MARGIN = 2, function(x) sample(x, replace = TRUE,
                size = 10000)) %>% as.matrix.data.frame()
colnames(W.boot)<- c('male', 'momschooling2', 'momschooling3',
                             'momschooling_miss','dadschooling2',
                             'dadschooling3', 'dadschooling_miss',
                             'broken_home1', 'broken_home_miss',
                             'incomehh2', 'incomehh3', 'incomehh_miss',
                             'north', 'center')

D.boot <- apply(D.matrix, MARGIN = 2, function(x) sample(x, replace = TRUE,
                size = 10000)) %>% as.matrix.data.frame()
colnames(D.boot)<- c('privateHS')

#Using normalized loading in order to simplify instruments
I<- (Z.boot %*% t(Gamma)) + theta * alpha_I + V
D<- ifelse(I >= 0, 1,0) %>% as.matrix.data.frame()

#Generate Potential Outcomes
Y1<- X.boot %*% beta_1 + theta * alpha_1 + U1 %>% as.matrix()
Y0<- X.boot %*% beta_0 + theta * alpha_0 + U0 %>% as.matrix()


# Generate Model
#Generate Y= DY1 + (1-D)Y0
Y<- D * Y1 + (1 - D) * Y0 %>% as.matrix.data.frame()
```

```
# Test Scores
T_1<- W.boot %*% omega_T1 + theta * alpha.T1 + eps1
T_2<- W.boot %*% omega_T2 + theta * alpha.T2 + eps2
T_3<- W.boot %*% omega_T3 + theta * alpha.T3 + eps3
T_4<- W.boot %*% omega_T4 + theta * alpha.T4 + eps4

New_Df<- cbind(T_1, T_2, T_3, T_4, D, Y, theta) %>% as.data.frame()
colnames(New_Df)<- c('T1', 'T2', 'T3', 'T4', 'privateHS', 'wage', 'Theta')

Df2<- subset(New_Df, D== 1)


# Model
model_mean_lect<- New_Df %>% pull(T1) %>% mean()
model_mean_mate<- New_Df %>% pull(T2) %>% mean()
model_mean_soc<- New_Df %>% pull(T3) %>% mean()
model_mean_nat<- New_Df %>% pull(T4) %>% mean()
model_mean_privHS<- New_Df %>% pull(privateHS) %>% mean()
model_mean_wageonprivHS<- Df2 %>% pull(wage) %>% mean()

model_sd_lect<- New_Df %>% pull(T1) %>% sd()
model_sd_mate<- New_Df %>% pull(T2) %>% sd()
model_sd_soc<- New_Df %>% pull(T3) %>% sd()
model_sd_nat<- New_Df %>% pull(T4) %>% sd()
model_sd_privHS<- New_Df %>% pull(privateHS) %>% sd()
model_sd_wageonprivHS<- Df2 %>% pull(wage) %>% sd()

# Actual
Df3<- subset(data_q4, privateHS == 1)
actual_mean_lect<- data_q4 %>% pull(T1) %>% mean()
actual_mean_mate<- data_q4 %>% pull(T2) %>% mean()
actual_mean_soc<- data_q4 %>% pull(T3) %>% mean()
actual_mean_nat<- data_q4 %>% pull(T4) %>% mean()
actual_mean_privHS<- data_q4 %>% pull(privateHS) %>% mean()
actual_mean_wageonprivHS<- Df3 %>% pull(wage) %>% mean()

actual_sd_lect<- data_q4 %>% pull(T1) %>% sd()
actual_sd_mate<- data_q4 %>% pull(T2) %>% sd()
actual_sd_soc<- data_q4 %>% pull(T3) %>% sd()
actual_sd_nat<- data_q4 %>% pull(T4) %>% sd()
actual_sd_privHS<- data_q4 %>% pull(privateHS) %>% sd()
actual_sd_wageonprivHS<- Df3 %>% pull(wage) %>% sd()
```

Goodness of Fit	Actual(mean)	Model(mean)
Lect	0.0736875	0.0528136
Mate	0.0701301	0.0737207
Soc	0.0762443	0.0795918
Nat	0.0343697	0.0312164
PrivateHS	0.6276858	0.8989
Wage(Contidional)	8.2491521	8.8105234

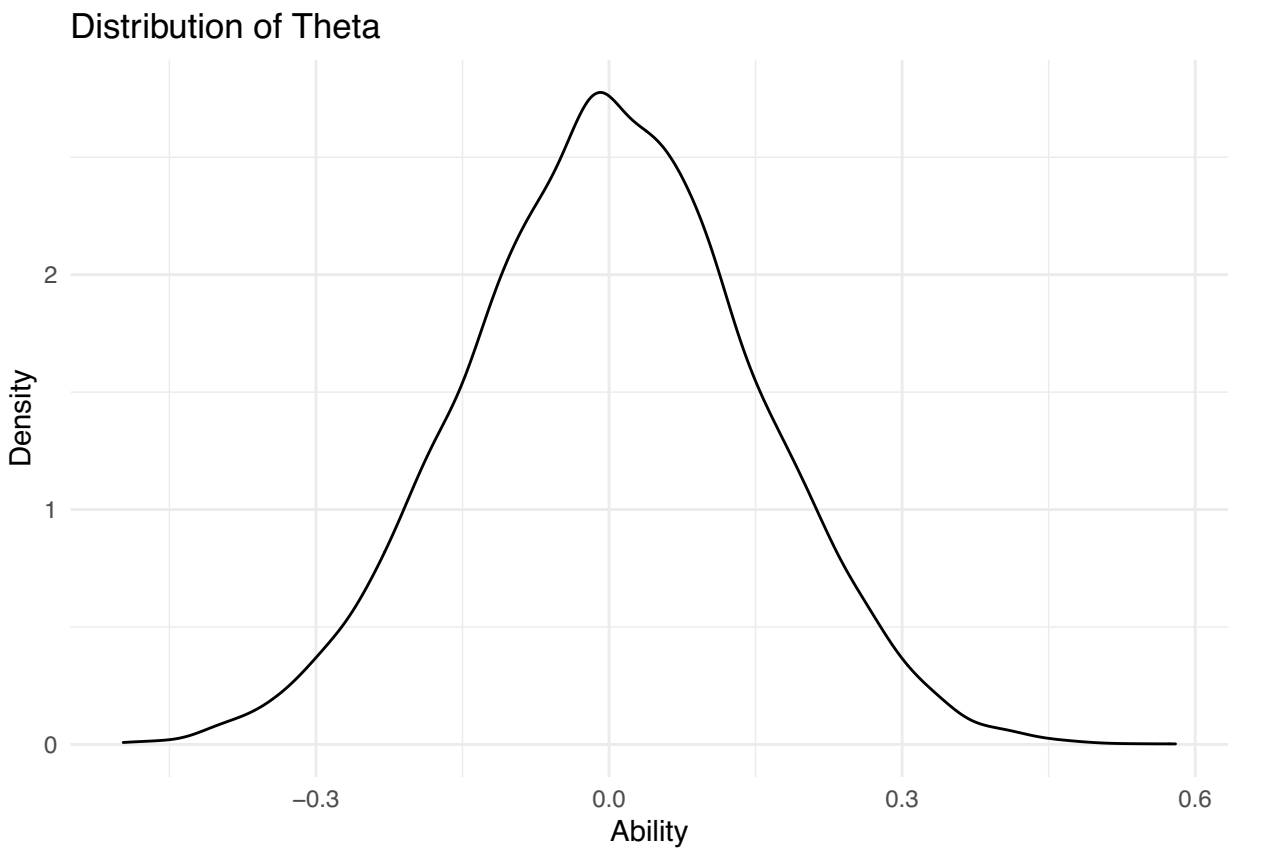
Goodness of Fit	Actual(std.dev.)	Model(std.dev.)
Lect	0.9368779	0.4967984
Mate	0.9134723	0.3484445
Soc	0.9421664	0.2858925
Nat	0.9509775	0.3056515
PrivateHS	0.4835067	0.3014762
Wage(Contidional)	1.1477591	1.879029

9.

Using your simulated data, plot the distribution of θ.

Answer

```
theta_data<- cbind(theta) %>% as.data.frame()
ggplot(theta_data)+
  geom_density(aes(x = theta, y = ..density..),
                colour = 'black')+
  labs(
        title = 'Distribution of Theta',
        x = 'Ability',
        y = 'Density'
        )+
  theme_minimal()
```



10.

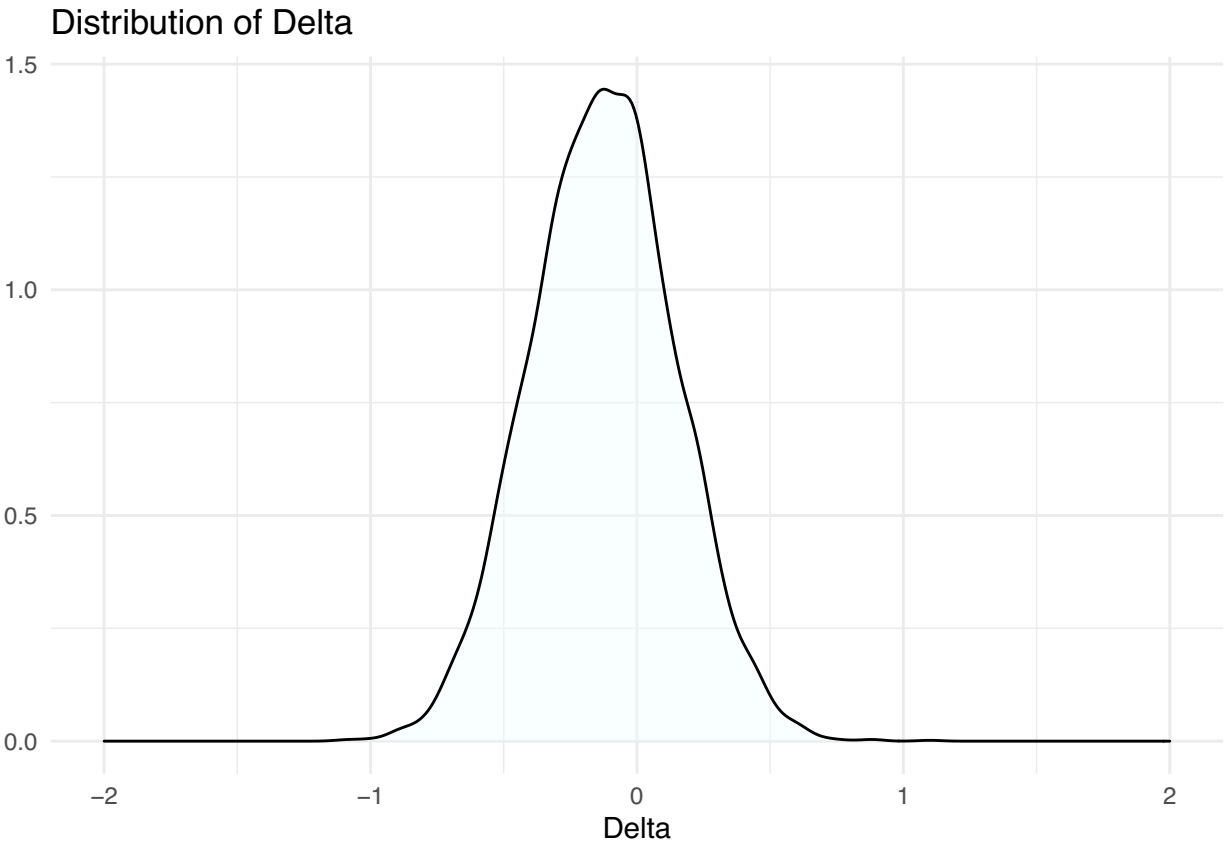
Using your simulated data, plot $Y_1 - Y_0$.

Answer

```
DataFrame<- data.frame('Delta' = Y1 - Y0,
                'Y1' = Y1, 'Y0' = Y0,
                'DecVar' = D, 'Latent' = I, 'Ability' = theta )

ggplot(DataFrame) +
    geom_density(aes(x = Delta, y = ..density..),alpha = 0.4, fill = 'azure')+
  labs(
        title = 'Distribution of Delta',
        x = 'Delta',
        y = NULL
        )+xlim(-2,2)+theme_minimal()
```

```
## Warning: Removed 5088 rows containing non-finite values (stat_density).
```



11.

Using your simulated data, compute and show the ATE and TT parameters.

Answer

```
ATE.aux<- DataFrame %>% pull(Delta) %>% mean()
TT.aux<- DataFrame %>% filter(DecVar == 1) %>% pull(Delta) %>% mean()
```

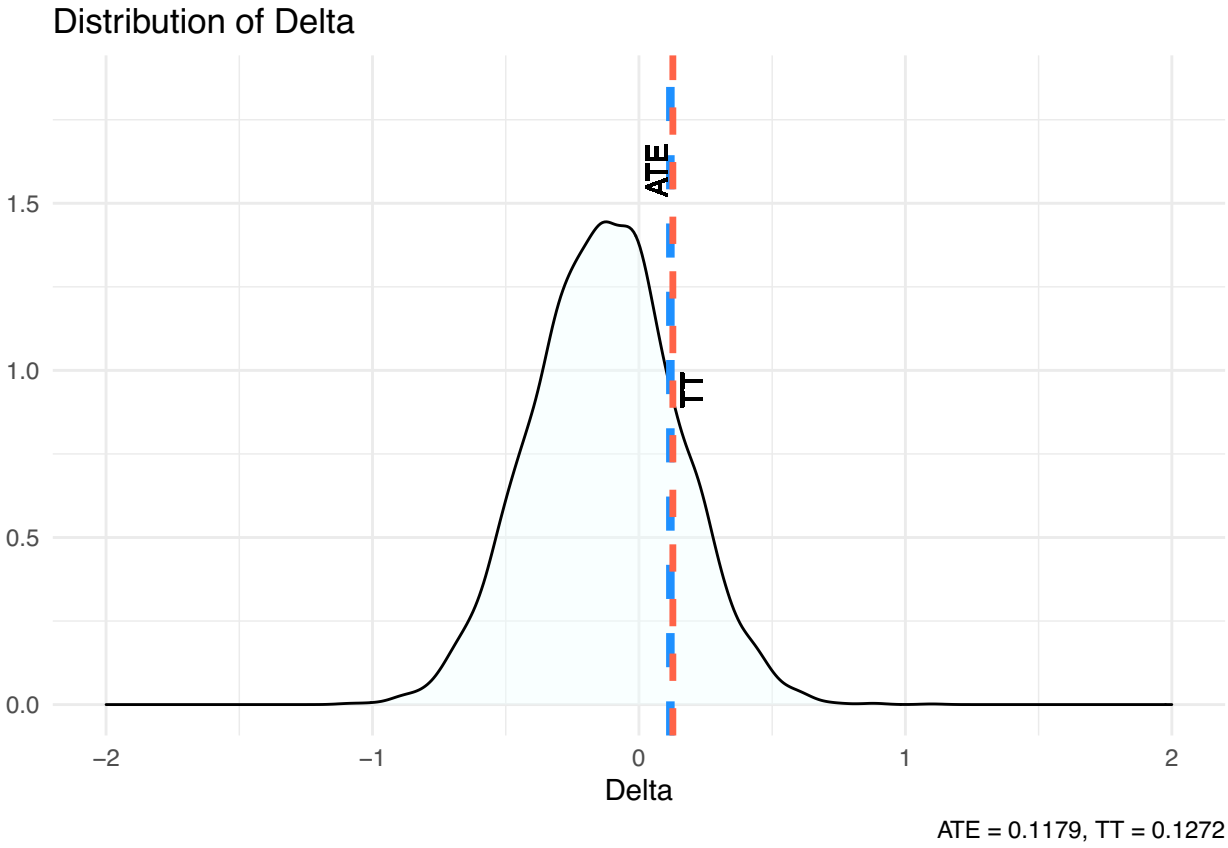
```
## [1] 0.117885
```

```
## [1] 0.1272301
```

```
ggplot(DataFrame) +
    geom_density(aes(x = Delta, y = ..density..),alpha = 0.4, fill = 'azure')+
    geom_vline(xintercept = ATE, linetype = 2, colour = 'dodgerblue',
               size= 1.5)+
    geom_text(aes(x = ATE - 0.05, y = 1.6, label = 'ATE'), angle = 90) +
    geom_vline(xintercept = TT,
               linetype = 2,
               colour = 'tomato', size= 1.2) +
    geom_text(aes(x = TT + 0.07, y =.94 , label = 'TT'), angle = 90)+
  labs(
        title = 'Distribution of Delta',
        x = 'Delta',
        y = NULL,
         caption = paste0(
            'ATE = ' ,round(ATE,4),

            ', TT = ' ,round(TT,4)
        )
        )+xlim(-2,2)+ylim(0,1.85)+theme_minimal()
```

```
## Warning: Removed 5088 rows containing non-finite values (stat_density).
```



12.

Using your simulated data, plot the TT parameter as function of the unobserved ability, θ.

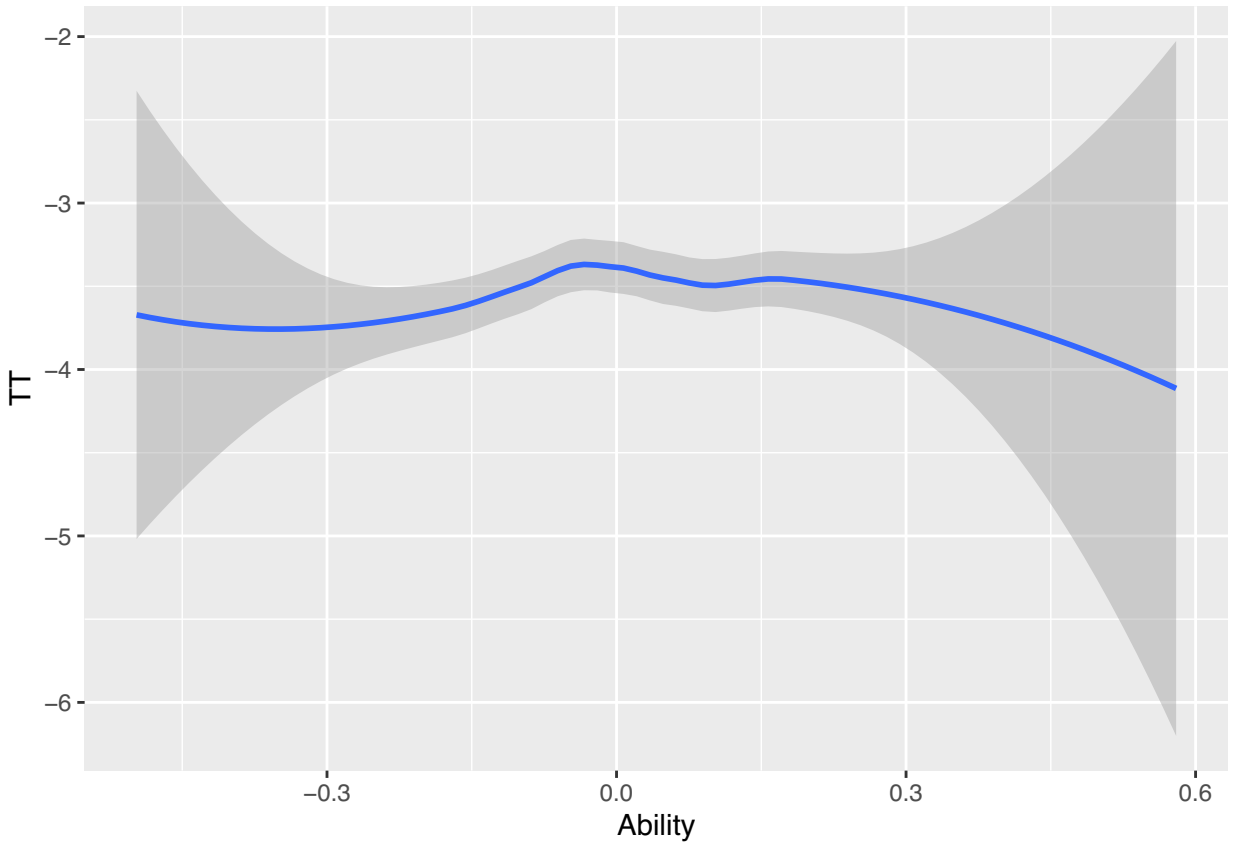
Answer

```
TT.plot<- DataFrame %>% filter(DecVar == 1) %>% pull(Delta) %>% as.data.frame()
theta.ability<- DataFrame %>% filter(DecVar == 1)%>% pull(Ability) %>%
                as.data.frame()

TT.help<- sample(TT.plot,length(theta.ability), replace= TRUE) %>%
         as.data.frame()

DF_aux<- data.frame(TT.plot,theta.ability)
colnames(DF_aux)<- c('TT', 'Ability')
ggplot(DF_aux, aes(x=Ability, y=TT))+
  stat_smooth(method= 'loess')
```

```
## `geom_smooth()` using formula 'y ~ x'
```


−2
−3
−4
−5
−6
TT
−0.3
0.0
0.3
0.6
Ability

MATLAB codes

Listing 1.1: Dummy File For Reading Data

```
data = readmatrix('/Volumes/External/AdvMicroEconMetrics/PS_2/
   data_ps2.csv');

onesVector = ones(size(data, 1), 1);

D = data(:, 5);
T = data(:, 1:4);
W = [onesVector, data(:, 7:20)];
X = [onesVector, data(:, [7, 19, 20])];
Y = data(:, 6);
Z = [onesVector, data(:, [7, 19:22])];
```

Listing 1.2: OLS Regression

```
function [beta, sigma] = OLS_Est(X, Y)
%OLS regression
beta = inv(X' * X) * X' * Y;
eps = Y - X*beta;
sigma = sqrt((eps' * eps) / (size(Y, 1) - size(X, 2)));
end
```

Listing 1.3: MLE

```
function output = loglikelihood_ps2(init)


% init -> initial guess
global D T W X Y Z

colW = size(W, 2);
colX = size(X, 2);
colZ = size(Z, 2);

% Identification auxiliary
aux = [colX, ...
    2 * colX, ...
    2 * colX + colZ, ...
```

```
    2 * colX + colZ + colW, ...
    2 * colX + colZ + 2 * colW, ...
    2 * colX + colZ + 3 * colW, ...
    2 * colX + colZ + 4 * colW, ...
    2 * colX + colZ + 4 * colW + 1, ...
    2 * colX + colZ + 4 * colW + 2, ...
    2 * colX + colZ + 4 * colW + 3, ...
    2 * colX + colZ + 4 * colW + 4, ...
    2 * colX + colZ + 4 * colW + 5, ...
    2 * colX + colZ + 4 * colW + 6, ...
    2 * colX + colZ + 4 * colW + 7, ...
    2 * colX + colZ + 4 * colW + 8, ...
    2 * colX + colZ + 4 * colW + 9, ...
    2 * colX + colZ + 4 * colW + 10, ...
    2 * colX + colZ + 4 * colW + 11, ...
    2 * colX + colZ + 4 * colW + 12, ...
    2 * colX + colZ + 4 * colW + 13];

% Parameters
beta_0 = init(1:aux(1));
beta_1 = init(aux(1) + 1:aux(2));
beta_D = init(aux(2) + 1:aux(3));
omega_T1 = init(aux(3) + 1:aux(4));
omega_T2 = init(aux(4) + 1:aux(5));
omega_T3 = init(aux(5) + 1:aux(6));
omega_T4 = init(aux(6) + 1:aux(7));
sigma_0 = init(aux(7) + 1:aux(8));
sigma_1 = init(aux(8) + 1:aux(9));
sigma_T1 = init(aux(9) + 1:aux(10));
sigma_T2 = init(aux(10) + 1:aux(11));
sigma_T3 = init(aux(11) + 1:aux(12));
sigma_T4 = init(aux(12) + 1:aux(13));
alpha_0 = init(aux(13) + 1:aux(14));
alpha_1 = init(aux(14) + 1:aux(15));
alpha_I = init(aux(15) + 1:aux(16));
alpha_T2 = init(aux(16) + 1:aux(17));
alpha_T3 = init(aux(17) + 1:aux(18));
alpha_T4 = init(aux(18) + 1:aux(19));
sigma_theta = init(aux(19) + 1:aux(20));

MLfunc = @(theta) (normpdf(Y - X * beta_0 - theta * alpha_0, 0,
   exp(sigma_0)) ...
    .* normpdf(Y - X * beta_1 - theta * alpha_1, 0, exp(sigma_1))
       ...
    .* normpdf(T(:, 1) - W * omega_T1 - theta, 0, exp(sigma_T1))
       ...
    .* normpdf(T(:, 2) - W * omega_T2 - theta * alpha_T2, 0, exp(
       sigma_T2)) ...
```

```
    .* normpdf(T(:, 3) - W * omega_T3 - theta * alpha_T3, 0, exp(
       sigma_T3)) ...
    .* normpdf(T(:, 4) - W * omega_T4 - theta * alpha_T4, 0, exp(
       sigma_T4)) ...
    .* (1 - normcdf(Z * beta_D + theta * alpha_I, 0, 1)) .^ (1 - D
       ) ...
    .* normcdf(Z * beta_D + theta * alpha_I, 0, 1) .^ D ...
    .* normpdf(theta, 0, exp(sigma_theta)));


% Can use Gauss- Hermit or Montecarlo
% However there is a tradeoff between computing time and
   programming G-H
% integration.
sol = integral(MLfunc, -Inf, Inf, 'ArrayValued', true);

output = -sum(log(sol));

end
```

Listing 1.4: Estimation of Parameters

```
clear
% Set Seet: LGMV
randn('seed', 180618)

% Set global
global D T W X Y Z

%check dummy file read.m
read;

[beta_T1, sigma_T1] = OLS_Est(W, T(:, 1));
[beta_T2, sigma_T2] = OLS_Est(W, T(:, 2));
[beta_T3, sigma_T3] = OLS_Est(W, T(:, 3));
[beta_T4, sigma_T4] = OLS_Est(W, T(:, 4));
[beta_D, sigma_D] = OLS_Est(Z, D); % Take sigma_D = 1, to simplify
[beta_1, sigma_1] = OLS_Est(X, Y);
[beta_0, sigma_0] = OLS_Est(X, Y);

[alpha_T1, alpha_T2, alpha_T3, alpha_T4, alpha_I, alpha_1, alpha_0
   , sigma_theta] = deal(1);

% Initial guesses for
guesses = [beta_0; beta_1; beta_D; beta_T1; beta_T2; beta_T3;
   beta_T4; ...
    sigma_0; sigma_1; sigma_T1; sigma_T2; sigma_T3; sigma_T4; ...
    alpha_0; alpha_1; alpha_I; alpha_T2; alpha_T3; alpha_T4; ...
```

```
    sigma_theta];

tic
options = optimoptions(@fminunc, 'Algorithm', 'quasi-newton', '
   Display', 'iter',...
    'GradObj', 'off', 'HessUpdate', 'bfgs', 'UseParallel', false,
       ...
    'TolFun', 1e-6, 'TolX', 1e-6, 'MaxIter', 10000, 'MaxFunEvals',
        10000);
[est_beta, est_F, exitflag, output, grad, hessian] = fminunc('
   loglikelihood_ps2', guesses, options);

runtime = toc;

se_Hess=sqrt(diag(inv(hessian)));
[est_beta se_Hess]
```