Problem Set 2

Topics in Advanced Econometrics (ResEcon 703)
University of Massachusetts Amherst

Solutions

Rules

Email a single .pdf file of your problem set writeup, code, and output to `mwoerman@umass.edu` by the date and time above. You may work in groups of up to three and submit one writeup for the group, and I strongly encourage you to do so. This problem set requires you to code your own estimators, rather than using R's "canned" routines (e.g., `glm()` and `mlogit()`).

Data

Download the file `commute_datasets.zip` from the course website. This zipped file contains two datasets—`commute_binary.csv` and `commute_multinomial.csv`—that you will use for this problem set. Both datasets contain simulated data on the travel mode choice of 1000 UMass graduate students who commute to campus from more than one mile away. The `commute_binary.csv` dataset corresponds to commuting in the middle of winter when only driving a car or taking a bus are feasible options. The `commute_multinomial.csv` dataset corresponds to commuting in the spring when riding a bike and walking are feasible alternatives. See the file `commute_descriptions.txt` for descriptions of the variables in each dataset.

```r
### Load packages for problem set
library(tidyverse)
library(gmm)
```

Problem 1: Maximum Likelihood Estimation

We are again studying how UMass graduate students choose how to commute to campus in the spring when riding a bike and walking are feasible alternatives—as in problem 3 of problem set 1—but we are now estimating the model "by hand" to better understand the maximum likelihood estimation method. Use the `commute_multinomial.csv` dataset for this question.

```r
### Create functions for use with maximum likelihood
## Function to summarize MLE model results
summarize_mle <- function(model, names){
  ## Extract model parameter estimates
```

```r
  parameters <- model$par
  ## Calculate parameters standard errors
  std_errors <- model$hessian %>%
    solve() %>%
    diag() %>%
    sqrt()
  ## Calculate parameter z-stats
  z_stats <- parameters / std_errors
  ## Calculate parameter p-values
  p_values <- 2 * pnorm(-abs(z_stats))
  ## Summarize results in a list
  model_summary <- tibble(names = names,
                          parameters = parameters,
                          std_errors = std_errors,
                          z_stats = z_stats,
                          p_values = p_values)
  ## Return model_summary object
  return(model_summary)
}
## Function to conduct likelihood ratio test
test_likelihood_ratio <- function(model_rest, model_unrest){
  ## Calculate likelihood ratio test statistic
  test_stat <- 2 * (model_rest$value - model_unrest$value)
  ## Calculate the number of restrictions
  df <- length(model_unrest$par) - length(model_rest$par)
  ## Test if likelihood ratio test statistic is greater than critical value
  test <- test_stat > qchisq(0.95, df)
  ## Calculate p-value of test
  p_value <- 1 - pchisq(test_stat, df)
  ## Return test result and p-value
  return(list(reject = test, p_value = p_value))
}

## Load dataset
data_multi <- read_csv('commute_multinomial.csv')

## Rows:  1000 Columns:  13
## - Column specification --------------------------
## Delimiter:  ","
## chr  (2):  mode, marital_status
## dbl (11):  id, time.car, cost.car, time.bus, cost.bus, time.bike, cost.b...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a. Model the commute choice during spring as a multinomial logit model. Express the representative utility of each alternative as a linear function of its cost and time with common parameters on these variables. That is, the representative utility to student n from alternative j is

$$V_{nj} = \beta C_{nj} + \gamma T_{nj}$$

where C_{nj} is the cost to student n of alternative j, T_{nj} is the time for student n of alternative j, and the β and γ parameters are to be estimated. Estimate the parameters of this model by maximum likelihood estimation. The following steps can provide a rough guide to creating your own maximum likelihood estimator:

I. Create a function that takes a set of parameters and data as inputs: `function(parameters, data)`.

II. Within that function, make the following calculations:
   i. Calculate the representative utility of every alternative for each decision maker.
   ii. Calculate the choice probability of the chosen alternative for each decision maker.
   iii. Sum the log of these choice probabilities to get the log-likelihood.
   iv. Return the negative of the log-likelihood.

III. Maximize the log-likelihood (by minimizing its negative) using `optim()`. Your call of the `optim()` function may look something like:

```
optim(par = your_starting_guesses, fn = your_function, data = your_data,
      method = 'BFGS', hessian = TRUE)
```

Report your parameter estimates, standard errors, z-stats, and p-values. Briefly interpret these results. For example, what does each parameter mean?

```r
## Function to calculate log-likelihood for heating choice
ll_fn_1a <- function(params, data){
  ## Extract individual parameters with descriptive names
  beta_1 <- params[1]
  beta_2 <- params[2]
  ## Calculate representative utility for each alternative given the parameters
  model_data <- data %>%
    mutate(utility_bike = beta_1 * cost.bike + beta_2 * time.bike,
           utility_bus = beta_1 * cost.bus + beta_2 * time.bus,
           utility_car = beta_1 * cost.car + beta_2 * time.car,
           utility_walk = beta_1 * cost.walk + beta_2 * time.walk)
  ## Calculate logit choice probability denominator given the parameters
  model_data <- model_data %>%
    mutate(prob_denom = exp(utility_bike) + exp(utility_bus) +
             exp(utility_car) + exp(utility_walk))
  ## Calculate logit choice probability for each alt given the parameters
  model_data <- model_data %>%
    mutate(prob_bike = exp(utility_bike) / prob_denom,
           prob_bus = exp(utility_bus) / prob_denom,
           prob_car = exp(utility_car) / prob_denom,
           prob_walk = exp(utility_walk) / prob_denom)
  ## Calculate logit choice probability for chosen alt given the parameters
```

```r
  model_data <- model_data %>%
    mutate(prob_choice = prob_bike * (mode == 'bike') +
             prob_bus * (mode == 'bus') + prob_car * (mode == 'car') +
             prob_walk * (mode == 'walk'))
  ## Calculate log of logit choice probability for chosen alt given the params
  model_data <- model_data %>%
    mutate(log_prob = log(prob_choice))
  ## Calculate the log-likelihood for these parameters
  ll <- sum(model_data$log_prob)
  return(-ll)
}
## Maximize the log-likelihood function
model_1a <- optim(par = rep(0, 2), fn = ll_fn_1a, data = data_multi,
                  method = 'BFGS', hessian = TRUE)
## Summarize model results
model_1a %>%
  summarize_mle(c('cost', 'time'))
```

```
## # A tibble: 2 x 5
##   names parameters std_errors z_stats p_values
##   <chr>      <dbl>      <dbl>   <dbl>    <dbl>
## 1 cost      -1.00      0.175    -5.72 1.07e- 8
## 2 time      -0.126     0.0100  -12.6  3.31e-36
```

Both parameters are statistically significant and are interpreted as marginal utilities. The cost of driving decreases the utility of driving, and the time spent commuting by a particular travel mode decreases the utility of taking that mode. This result is intuitive since people like both money and leisure time.

b. Again model the commute choice during spring as a multinomial logit model, but now add alternative-specific intercepts for all but one alternative. That is, the representative utility to student n from alternative j is

$$V_{nj} = \alpha_j + \beta C_{nj} + \gamma T_{nj}$$

where C_{nj} is the cost to student n of alternative j, T_{nj} is the time for student n of alternative j, and the α, β, and γ parameters are to be estimated. Estimate the parameters of this model by maximum likelihood estimation. The steps to creating your own maximum likelihood estimator are the same as in part (a), but some of the calculations will be different.

```r
## Function to calculate log-likelihood for heating choice
ll_fn_1b <- function(params, data){
  ## Extract individual parameters with descriptive names
  alpha_bus <- params[1]
  alpha_car <- params[2]
  alpha_walk <- params[3]
  beta_1 <- params[4]
  beta_2 <- params[5]
  ## Calculate representative utility for each alternative given the parameters
```

```r
  model_data <- data %>%
    mutate(utility_bike = beta_1 * cost.bike + beta_2 * time.bike,
           utility_bus = alpha_bus + beta_1 * cost.bus + beta_2 * time.bus,
           utility_car = alpha_car + beta_1 * cost.car + beta_2 * time.car,
           utility_walk = alpha_walk + beta_1 * cost.walk + beta_2 * time.walk)
  ## Calculate logit choice probability denominator given the parameters
  model_data <- model_data %>%
    mutate(prob_denom = exp(utility_bike) + exp(utility_bus) +
             exp(utility_car) + exp(utility_walk))
  ## Calculate logit choice probability for each alt given the parameters
  model_data <- model_data %>%
    mutate(prob_bike = exp(utility_bike) / prob_denom,
           prob_bus = exp(utility_bus) / prob_denom,
           prob_car = exp(utility_car) / prob_denom,
           prob_walk = exp(utility_walk) / prob_denom)
  ## Calculate logit choice probability for chosen alt given the parameters
  model_data <- model_data %>%
    mutate(prob_choice = prob_bike * (mode == 'bike') +
             prob_bus * (mode == 'bus') + prob_car * (mode == 'car') +
             prob_walk * (mode == 'walk'))
  ## Calculate log of logit choice probability for chosen alt given the params
  model_data <- model_data %>%
    mutate(log_prob = log(prob_choice))
  ## Calculate the log-likelihood for these parameters
  ll <- sum(model_data$log_prob)
  return(-ll)
}
## Maximize the log-likelihood function
model_1b <- optim(par = rep(0, 5), fn = ll_fn_1b, data = data_multi,
                  method = 'BFGS', hessian = TRUE)
```

i. Report your parameter estimates, standard errors, z-stats, and p-values. Briefly interpret these results. For example, what does each parameter mean?

```r
## Summarize model results
model_1b %>%
  summarize_mle(c('bus_int', 'car_int', 'walk_int', 'cost', 'time'))

## # A tibble: 5 x 5
##   names    parameters std_errors z_stats p_values
##   <chr>         <dbl>      <dbl>   <dbl>    <dbl>
## 1 bus_int       1.76      0.113     15.6 4.69e-55
## 2 car_int       2.92      0.200     14.6 2.39e-48
## 3 walk_int      3.17      0.306     10.4 3.46e-25
## 4 cost         -6.05      0.511    -11.9 2.11e-32
## 5 time         -0.296     0.0246   -12.0 2.08e-33
```

As in the previous model, the cost parameter and the time parameter are both negative, indicating that the cost and the time spent commuting by a particular travel mode decrease the utility of taking that mode. Additionally, all three alternative-specific intercepts are positive and significant, suggesting that, *ceteris paribus*, all other modes would be preferred to biking.

ii. Conduct a likelihood ratio test on this model to test the joint significance of the alternative-specific intercepts. That is, test the null hypothesis:

$$H_0\colon\ \alpha_{bus} = \alpha_{car} = \alpha_{walk} = 0$$

Your null hypothesis may be slightly different, depending on what you consider your "reference alternative." Do you reject this null hypothesis? What is the p-value of the test? Briefly interpret the result of this test. (Reminder: to conduct this likelihood ratio test, you need the log-likelihood value of the model in part (b) and the log-likelihood value of the restricted model that is obtained when the hypothesized restrictions are imposed.)

```r
## Conduct likelihood ratio test of models 1a and 1b
test_1b <- test_likelihood_ratio(model_1a, model_1b)
## Display test results
test_1b
```

```
## $reject
## [1] TRUE
##
## $p_value
## [1] 0
```

We reject this null hypothesis and conclude that the alternative-specific intercepts are jointly significant. That is, this model provides a better fit than the model in part (a), which restricted these parameters to all be zero.

c. Again model the commute choice during spring as a multinomial logit model, but now allow the parameter on time to be alternative-specific. That is, the representative utility to student n from alternative j is

$$V_{nj} = \alpha_j + \beta C_{nj} + \gamma_j T_{nj}$$

where C_{nj} is the cost to student n of alternative j, T_{nj} is the time for student n of alternative j, and the α, β, and γ parameters are to be estimated. Estimate the parameters of this model by maximum likelihood estimation. The steps to creating your own maximum likelihood estimator are the same as in part (a), but some of the calculations will be different.

```r
## Function to calculate log-likelihood for heating choice
ll_fn_1c <- function(params, data){
  ## Extract individual parameters with descriptive names
  alpha_bus <- params[1]
  alpha_car <- params[2]
  alpha_walk <- params[3]
  beta <- params[4]
  gamma_bike <- params[5]
  gamma_bus <- params[6]
  gamma_car <- params[7]
```

```r
  gamma_walk <- params[8]
  ## Calculate representative utility for each alternative given the parameters
  model_data <- data %>%
    mutate(utility_bike = beta * cost.bike + gamma_bike * time.bike,
           utility_bus = alpha_bus + beta * cost.bus + gamma_bus * time.bus,
           utility_car = alpha_car + beta * cost.car + gamma_car * time.car,
           utility_walk = alpha_walk + beta * cost.walk +
             gamma_walk * time.walk)
  ## Calculate logit choice probability denominator given the parameters
  model_data <- model_data %>%
    mutate(prob_denom = exp(utility_bike) + exp(utility_bus) +
             exp(utility_car) + exp(utility_walk))
  ## Calculate logit choice probability for each alt given the parameters
  model_data <- model_data %>%
    mutate(prob_bike = exp(utility_bike) / prob_denom,
           prob_bus = exp(utility_bus) / prob_denom,
           prob_car = exp(utility_car) / prob_denom,
           prob_walk = exp(utility_walk) / prob_denom)
  ## Calculate logit choice probability for chosen alt given the parameters
  model_data <- model_data %>%
    mutate(prob_choice = prob_bike * (mode == 'bike') +
             prob_bus * (mode == 'bus') + prob_car * (mode == 'car') +
             prob_walk * (mode == 'walk'))
  ## Calculate log of logit choice probability for chosen alt given the params
  model_data <- model_data %>%
    mutate(log_prob = log(prob_choice))
  ## Calculate the log-likelihood for these parameters
  ll <- sum(model_data$log_prob)
  return(-ll)
}
## Maximize the log-likelihood function
model_1c <- optim(par = rep(0, 8), fn = ll_fn_1c, data = data_multi,
                  method = 'BFGS', hessian = TRUE)
```

i. Report your parameter estimates, standard errors, z-stats, and p-values. Briefly interpret these results. For example, what does each parameter mean?

```r
## Summarize model results
model_1c %>%
  summarize_mle(c('bus_int', 'car_int', 'walk_int', 'cost',
                  'time_bike', 'time_bus', 'time_car', 'time_walk'))

## # A tibble: 8 x 5
##   names     parameters std_errors z_stats p_values
##   <chr>          <dbl>      <dbl>   <dbl>    <dbl>
## 1 bus_int       -0.219      0.386  -0.568 5.70e- 1
## 2 car_int        2.75       0.443   6.20  5.52e-10
```

```
## 3 walk_int       2.98      0.783    3.80  1.44e- 4
## 4 cost          -2.60      0.824   -3.16  1.56e- 3
## 5 time_bike     -0.289     0.0386  -7.51  6.12e-14
## 6 time_bus      -0.143     0.0351  -4.08  4.53e- 5
## 7 time_car      -0.405     0.0464  -8.73  2.63e-18
## 8 time_walk     -0.297     0.0384  -7.72  1.14e-14
```

As in the previous models, the cost parameter is negative, indicating that the cost of driving decreases the utility of driving. The alternative-specific parameters on time are all negative but tend to be different from one another—the bike and walk parameters are not statistically different from one another, but all other pairwise combinations of parameters are. These parameters indicate that the time spent commuting by a particular travel mode always decreases the utility of taking that mode, but that these marginal utilities of time differ by travel mode. Additionally, the car and walk intercepts are positive and significant, while the bus intercept is not statistically significant. These results suggest that, *ceteris paribus*, driving or walking would be preferred to taking the bus or biking.

ii. Conduct a likelihood ratio test on this model to test if the alternative-specific parameters on time are equal to one another. That is, test the null hypothesis:

$$H_0\colon \gamma_{bike} = \gamma_{bus} = \gamma_{car} = \gamma_{walk}$$

Do you reject this null hypothesis? What is the p-value of the test? Briefly interpret the result of this test.

```r
## Conduct likelihood ratio test of models 1a and 1b
test_1c <- test_likelihood_ratio(model_1b, model_1c)
## Display test results
test_1c
```

```
## $reject
## [1] TRUE
##
## $p_value
## [1] 5.179395e-10
```

We reject this null hypothesis and conclude that the alternative-specific marginal utilities of time are not all equal to one another. That is, this model provides a better fit than the model in part (b), which restricted these parameters to be equal.

Problem 2: Generalized Method of Moments

We are again studying how UMass graduate students choose how to commute to campus in the winter when riding a bike and walking are infeasible—as in problem 2 of problem set 1—but we are now estimating the model "by hand" to better understand the generalized method of moments estimation method. Use the `commute_binary.csv` dataset for this question.

```r
## Load dataset
data_binary <- read_csv('commute_binary.csv')
```

```
## Rows:  1000 Columns:  13
## - Column specification --------------------------
## Delimiter:  ","
## chr  (2):  mode, marital_status
## dbl (11):  id, time.car, cost.car, time.bus, cost.bus, price_gas, snowfa...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a. Model the choice to drive to campus during winter as a binary logit model. Express the representative utility of each alternative as a linear function of its cost and time. Include an alternative-specific intercept and allow the parameter on time to be alternative-specific. That is, the representative utility to student n from driving and taking the bus, respectively, are

$$V_{nc} = \alpha + \beta C_{nc} + \gamma_{car} T_{nc}$$
$$V_{nb} = \gamma_{bus} T_{nb}$$

where C_{nj} is the cost to student n of alternative j, T_{nj} is the time for student n of alternative j, and the α, β, and γ parameters are to be estimated. We exclude a bus-specific intercept term because only one intercept term is identified in this model, and we exclude the bus cost because it is free for all students. It may be easier to think about the difference in representative utility between driving and taking the bus for student n:

$$V_{nc} - V_{nb} = \alpha + \beta C_{nc} + \gamma_{car} T_{nc} - \gamma_{bus} T_{nb}$$

Because this is a binary logit model, we can express the choice probability of driving as a function of $V_{nc} - V_{nb}$:

$$P_{nc} = \frac{1}{1 + e^{-(V_{nc} - V_{nb})}}$$

Estimate the parameters of this model by method of moments. The following steps can provide a rough guide to creating your own method of moments estimator:

I. Write down moment conditions for this model. You should have four moment conditions for this model.

II. Create a function that takes a set of parameters and a *matrix* of data as inputs: `function(parameters, data_matrix)`.

III. Within that function, make the following calculations:

   i. Calculate the difference in representative utility for each decision maker.

   ii. Calculate the choice probability of driving for each decision maker.

   iii. Calculate the econometric residual, or the difference between the outcome and the probability, for each decision maker.

   iv. Calculate each of the L moments for each decision maker.

   v. Return the $N \times L$ matrix of individual moments.

IV. Find the MM estimator using `gmm()`. Your call of the `gmm()` function may look something like:

```
gmm(g = your_function, x = your_data_matrix, t0 = your_starting_guesses,
    vcov = 'iid', method = 'Nelder-Mead'
    control = list(reltol = 1e-25, maxit = 10000))
```

Report your parameter estimates, standard errors, t-stats, and p-values. Briefly interpret these results. For example, what does each parameter mean?

```
## Create dataset for use in MM moment function
data_2a <- data_binary %>%
  mutate(choice = 1 * (mode == 'car'),
         constant = 1,
         time.bus = -time.bus) %>%
  select(choice, constant, cost.car, time.car, time.bus) %>%
  as.matrix()
## Function to calculate moments for commute choice
mm_fn_2a <- function(params, data){
  ## Select data for X [N x K]
  X <- data[, -1]
  ## Select data for y [N x 1]
  y <- data[, 1]
  ## Calculate representative utility of driving [N x 1]
  utility <- X %*% params
  ## Calculate logit choice probability of driving [N x 1]
  prob <- 1 / (1 + exp(-utility))
  ## Calculate econometric residuals [N x 1]
  residuals <- y - prob
  ## Create moment matrix [N x K]
  moments <- c(residuals) * X
  return(moments)
}
## Use GMM to estimate model
model_2a <- gmm(g = mm_fn_2a, x = data_2a, t0 = rep(0, 4),
                vcov = 'iid', method = 'Nelder-Mead',
                control = list(reltol = 1e-25, maxit = 10000))
## Summarize model results
summary(model_2a)
```

```
##
## Call:
## gmm(g = mm_fn_2a, x = data_2a, t0 = rep(0, 4), vcov = "iid",
##     method = "Nelder-Mead", control = list(reltol = 1e-25, maxit = 10000))
##
##
## Method:  twoStep
##
## Coefficients:
##            Estimate     Std. Error   t value      Pr(>|t|)
## Theta[1]    2.2333e+00   3.7522e-01   5.9518e+00   2.6516e-09
## Theta[2]   -2.0772e+00   7.1798e-01  -2.8930e+00   3.8153e-03
## Theta[3]   -3.3222e-01   3.7929e-02  -8.7590e+00   1.9707e-18
## Theta[4]   -1.3257e-01   3.1928e-02  -4.1523e+00   3.2919e-05
##
```

```
## J-Test: degrees of freedom is 0
##                  J-test               P-value
## Test E(g)=0:    9.48387867832971e-23  *******
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  1703
## Gradian eval. =  NA
```

The parameters on cost and time are statistically significant and are interpreted as marginal utilities. The cost of driving decreases the utility of driving, and the time spent commuting by a particular travel mode decreases the utility of taking that mode. This result is intuitive since people like both money and leisure time. Notably, the marginal utility of time spent driving and the marginal utility of time spent on the bus are different, indicating that time on the bus is preferred to time driving.

b. You might be concerned that the cost and time data are exogenous; for example, a student who enjoys driving is more likely to live farther from campus because they do not mind the extra cost and time spent driving, and a student who enjoys taking the bus is more likely to live close to a bus stop so the bus commute time is less. The `commute_binary.csv` dataset includes four possible instruments that could be correlated with the cost or time of commuting: `price_gas`, `snowfall`, `construction`, and `bus_detour`. Again model the choice to drive to campus during winter as in (a). That is, the representative utility to student n from driving and taking the bus, respectively, are

$$V_{nc} = \alpha + \beta C_{nc} + \gamma_{car} T_{nc}$$
$$V_{nb} = \gamma_{bus} T_{nb}$$

where C_{nj} is the cost to student n of alternative j, T_{nj} is the time for student n of alternative j, and the α, β, and γ parameters are to be estimated. Estimate the parameters of this model by generalized method of moments, constructing moment conditions using the instruments described above. Thus, you will have five instruments: constant term, `price_gas`, `snowfall`, `construction`, and `bus_detour`. The steps to creating your own generalized method of moments estimator are the same as in part (a), but now you have four parameters and five moment conditions. (Note: this model may have challenges converging, but you can help it along by giving it different starting values. I got it to converge by using `t0 = c(0, 0, -0.3, -0.1)`.)

```r
## Create dataset for use in MM moment function
data_2b <- data_binary %>%
  mutate(choice = 1 * (mode == 'car'),
         constant = 1,
         time.bus = -time.bus) %>%
  select(choice, constant, cost.car, time.car, time.bus,
         price_gas, snowfall, construction, bus_detour) %>%
  as.matrix()
## Function to calculate moments for commute choice
gmm_fn_2b <- function(params, data){
  ## Select data for X [N x K]
```

```r
  X <- data[, 2:5]
  ## Select data for y [N x 1]
  y <- data[, 1]
  ## select data for Z [N x L]
  Z <- data[, c(2, 6:9)]
  ## Calculate representative utility of driving [N x 1]
  utility <- X %*% params
  ## Calculate logit choice probability of driving [N x 1]
  prob <- 1 / (1 + exp(-utility))
  ## Calculate econometric residuals [N x 1]
  residuals <- y - prob
  ## Create moment matrix [N x K]
  moments <- c(residuals) * Z
  return(moments)
}
## Use GMM to estimate model
model_2b <- gmm(g = gmm_fn_2b, x = data_2b, t0 = c(0, 0, -0.3, -0.1),
                vcov = 'iid', method = 'Nelder-Mead',
                control = list(reltol = 1e-25, maxit = 10000))
```

i. Report your parameter estimates, standard errors, t-stats, and p-values. Briefly interpret these results. For example, what does each parameter mean?

```r
## Summarize model results
summary(model_2b)
```

```
##
## Call:
## gmm(g = gmm_fn_2b, x = data_2b, t0 = c(0, 0, -0.3, -0.1), vcov = "iid",
##     method = "Nelder-Mead", control = list(reltol = 1e-25, maxit = 10000))
##
##
## Method:  twoStep
##
## Coefficients:
##             Estimate    Std. Error  t value     Pr(>|t|)
## Theta[1]     2.9119906   3.8112198   0.7640574   0.4448330
## Theta[2]    -3.9860509   8.0529192  -0.4949821   0.6206128
## Theta[3]    -0.3509452   0.1228629  -2.8563968   0.0042848
## Theta[4]    -0.1502996   0.0525380  -2.8607774   0.0042260
##
## J-Test: degrees of freedom is 1
##                  J-test     P-value
## Test E(g)=0:     0.0060018  0.9382488
##
## Initial values of the coefficients
##   Theta[1]   Theta[2]   Theta[3]   Theta[4]
```

```
##  2.0681542 -2.1903611 -0.3290997 -0.1453720
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  1175
## Gradian eval. =  NA
```

The parameter estimates are roughly the same as those in the previous model. However, the intercept and cost parameters now have much larger standard errors, rendering those parameters not statistically significant. Using instruments can reduce the precision of our parameter estimates, especially if they are not sufficiently correlated with the relevant variables, which may be the case here.

ii. Test if your model is correctly specified by performing an overidentifying restrictions test. Report the results of this test and briefly interpret these results. (Reminder: the `specTest()` function from the `gmm` package conducts this specification test.)

```
## Test overidentifying restrictions
specTest(model_2b)
```

```
##
##  ##  J-Test: degrees of freedom is 1  ##
##
##                  J-test     P-value
## Test E(g)=0:    0.0060018  0.9382488
```

The overidentifying restrictions test fails to reject the null hypothesis, so we conclude that all empirical moments are sufficiently close to zero. This result implies that this model is correctly specified and provides a good fit for our data.