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 all group members can submit the same code and output; indicate in your writeup who you worked with. You must submit a unique writeup that answers the problems below. You can discuss answers with your fellow group members, but your writeup must be in your own words. This problem set requires you to code your own estimators, rather than using R's "canned routines."

Data

Download the file travel_datasets.zip from the course website (github.com/woerman/ResEcon703). This zipped file contains two datasets, travel_binary.csv and travel_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 commuting to campus. The travel_binary.csv dataset corresponds to commuting in the middle of winter when only driving a car or taking a bus are feasible options (assume the weather is too severe for even the heartiest graduate students to ride a bike or walk). The travel_multinomial.csv dataset corresponds to commuting in spring when riding a bike and walking are feasible alternatives. See the file travel_descriptions.txt for descriptions of the variables in each dataset.

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

## - Attaching packages ------------------ tidyverse 1.2.1 -
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## - Conflicts -------------------- tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Problem 1: Maximum Likelihood Estimation

Use the `travel_multinomial.csv` dataset for this question.

a. Model the travel mode choice to commute to campus during spring as a multinomial logit model. Include the cost and the time of each alternative as explanatory variables with common coefficients; do not include alternative-specific intercepts. That is, the representative utility for alternative j is simply

$$V_{nj} = \beta_1 C_{nj} + \beta_2 T_{nj}$$

where C_{nj} is the cost of alternative j and T_{nj} is the time of alternative j. 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 for each alternative and 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 (or minimize 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.

```
### Create functions to calculate log likelihood, summarize MLE models, and
### conduct likelihood ratio tests
## Function to calculate logit log-likelihood
calculate_log_likelihood <- function(parameters, data_x, data_y){
  ## Calculate utility for each alternative given the parameters
  utility <- data_x %>%
    map(~ .x %*% parameters)
  ## Calculate logit probability denominator given the parameters
  probability_denominator <- utility %>%
    map(~ sum(exp(.x))) %>%
    unlist()
  ## Calculate logit probability numerator given the parameters
  probability_numerator <- utility %>%
    map2(data_y, ~ exp(sum(.x * .y))) %>%
    unlist()
  ## Calculate logit choice probability given the parameters
  probability_choice <-  probability_numerator / probability_denominator
  ## Calculate log of logit choice probability given the parameters
```

```r
  log_probability_choice <- log(probability_choice)
  ## Calculate the log-likelihood for these parameters
  log_likelihood <- sum(log_probability_choice)
  return(-log_likelihood)
}
## Function to summarize MLE model results
summarize_mle <- function(model, names){
  ## Extract model parameter estimates
  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 <- list(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_restricted, model_unrestricted){
  ## Calculate likelihood ratio test statistic
  test_statistic <- 2 * (model_restricted$value - model_unrestricted$value)
  ## Calculate the number of restrictions
  df <- length(model_unrestricted$par) - length(model_restricted$par)
  ## Test if likelihood ratio test statitsic is greater than critical value
  test <- test_statistic > qchisq(0.95, df)
  ## Calculate p-value of test
  p_value <- 1 - pchisq(test_statistic, df)
  ## Return test result and p-value
  return(list(reject = test, p_value = p_value))
}

### Part a
## Load dataset
data_multinomial <- read_csv('travel_multinomial.csv')
```

```
## Parsed with column specification:
## cols(
```

```r
##   mode = col_character(),
##   time_car = col_double(),
##   cost_car = col_double(),
##   time_bus = col_double(),
##   cost_bus = col_double(),
##   time_bike = col_double(),
##   cost_bike = col_double(),
##   time_walk = col_double(),
##   cost_walk = col_double(),
##   age = col_double(),
##   income = col_double(),
##   marital_status = col_character()
## )

## Gather wide dataset into a long dataset
data_multinomial_long <- data_multinomial %>%
  mutate(id = 1:n()) %>%
  gather(key, value, starts_with('cost_'), starts_with('time_')) %>%
  separate(key, c('key', 'alt')) %>%
  spread(key, value) %>%
  mutate(choice = (mode == alt)) %>%
  select(-mode)
## Split heating dataset into list of household data frames
data_multinomial_split <- data_multinomial_long %>%
  group_by(id) %>%
  arrange(id, alt) %>%
  group_split()
## Create matrices for choice outcomes
data_choice_1 <- data_multinomial_split %>%
  map(~ .x %>%
        select(choice) %>%
        mutate(choice = 1 * choice) %>%
        pull(choice))
## Create matrices of explanatory variables
data_explanatory_1a <- data_multinomial_split %>%
  map(~.x %>%
        select(cost, time) %>%
        as.matrix())
## Maximize the log-likelihood function
model_1a <- optim(rep(0, 2), calculate_log_likelihood,
                  data_x = data_explanatory_1a, data_y = data_choice_1,
                  method = 'BFGS', hessian = TRUE)
## Summarize model results
model_1a %>%
  summarize_mle(c('cost', 'time'))
```

```
## [1] "cost" "time"
##
## $parameters
## [1]  0.005921936 -0.077191067
##
## $std_errors
## [1] 0.048334999 0.006541943
##
## $z_stats
## [1]   0.1225186 -11.7994092
##
## $p_values
## [1] 9.024883e-01 3.930612e-32
```

The cost parameter is positive but not statistically significant, indicating that cost has no effect on the choice of travel mode. The time parameter is negative and statistically significant, indicating that time spent commuting reduces utility. The time result is intuitive, but the cost result does not have an economic explanation, suggesting that the model may be incorrect.

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

$$V_{nj} = \alpha_j + \beta_1 C_{nj} + \beta_2 T_{nj}$$

where α_j is an alternative-specific intercept, C_{nj} is the cost of alternative j and T_{nj} is the time of alternative j. Estimate the parameters of this model by maximum likelihood estimation. Follow the same set of steps as in (a), but now you have five parameters and the representative utility calculation has an additional component. You can use matrix multiplication in your log-likelihood function to flexibly accommodate different models, or you can create a different function for each model. Report your parameter estimates, standard errors, z-stats, and p-values. Briefly interpret these results.

```r
### Part b
## Create matrices of explanatory variables
data_explanatory_1b <- data_multinomial_split %>%
  map(~ rep(0, 3) %>%
        rbind(diag(3)) %>%
        cbind(.x %>%
                select(cost, time)) %>%
        as.matrix())
## Maximize the log-likelihood function
model_1b <- optim(rep(0, 5), calculate_log_likelihood,
                  data_x = data_explanatory_1b, data_y = data_choice_1,
                  method = 'BFGS', hessian = TRUE)
## Summarize model results
model_1b %>%
  summarize_mle(c('bus', 'car', 'walk', 'cost', 'time'))
```

```
## $names
## [1] "bus"  "car"  "walk" "cost" "time"
##
## $parameters
## [1]  1.6773063  1.7915827  1.8117712 -0.7082199 -0.1129539
##
## $std_errors
## [1] 0.267522222 0.373361834 0.153299220 0.164312061 0.008254052
##
## $z_stats
## [1]   6.269783   4.798516  11.818528  -4.310212 -13.684659
##
## $p_values
## [1] 3.615517e-10 1.598455e-06 3.131223e-32 1.630979e-05 1.253941e-42
```

All three alternative intercepts are positive and significant, indicating that, *ceteris paribus*, all other modes are preferred to biking; these three modes do not appear to be statistically different from one another. The cost parameter is now negative and statistically significant, indicating that the cost of commuting reduces utility. The time parameter is again negative and statistically significant, indicating that time spent commuting reduces utility. The cost and time results are now intuitive, implying that people like both money and time.

c. Conduct a likelihood ratio test on the model in part (b) to test the joint significance of the alternative-specific intercepts. That is, test the null hypothesis:

$$H_0\colon\ \alpha_b = \alpha_c = \alpha_w = 0$$

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?

```
### Part c
## Conduct likelihood ratio test of models 1a and 1b
test_1c <- test_likelihood_ratio(model_1a, model_1b)
## Display test results
test_1c

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

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

d. Again model the travel mode choice to commute to campus during spring as a multinomial logit model, but now add alternative-specific coefficients on the time variable. That is, the representative utility for alternative j is

$$V_{nj} = \alpha_j + \beta_1 C_{nj} + \beta_j T_{nj}$$

where α_j is an alternative-specific intercept, C_{nj} is the cost of alternative j, T_{nj} is the time of alternative j, and β_j varies for each alternative. Estimate the parameters of this model by maximum likelihood estimation. Follow the same set of steps as in (a), but now you have eight parameters and the representative utility calculation is different. You can use matrix multiplication in your log-likelihood function to flexibly accommodate different models, or you can create a different function for each model. Report your parameter estimates, standard errors, z-stats, and p-values. Briefly interpret these results.

```
### Part d
## Create matrices of explanatory variables
data_explanatory_1d <- data_multinomial_split %>%
  map(~ rep(0, 3) %>%
        rbind(diag(3)) %>%
        cbind(.x$cost) %>%
        cbind(diag(.x$time)) %>%
        as.matrix())
## Maximize the log-likelihood function
model_1d <- optim(rep(0, 8), calculate_log_likelihood,
                  data_x = data_explanatory_1d, data_y = data_choice_1,
                  method = 'BFGS', hessian = TRUE)
## Summarize model results
model_1d %>%
  summarize_mle(c('bus', 'car', 'walk', 'cost',
                  'time_bike', 'time_bus', 'time_car', 'time_walk'))
```

```
## $names
## [1] "bus"       "car"       "walk"      "cost"      "time_bike" "time_bus"
## [7] "time_car"  "time_walk"
##
## $parameters
## [1]  3.0835866  6.1870964  3.6096535 -2.8727780 -0.3222726 -0.2248886
## [7] -0.4602647 -0.3878172
##
## $std_errors
## [1] 0.31611786 0.57498330 0.32667525 0.28725797 0.02515921 0.02322886
## [7] 0.04488185 0.03363322
##
## $z_stats
## [1]   9.754547  10.760480  11.049669 -10.000691 -12.809328  -9.681429
## [7] -10.255030 -11.530777
##
## $p_values
## [1] 1.763860e-22 5.289383e-27 2.200235e-28 1.513379e-23 1.453890e-37
## [6] 3.616258e-22 1.123409e-24 9.230635e-31
```

All three alternative intercepts are again positive and significant, indicating that, *ceteris paribus*, all other modes are preferred to biking; it now appears that the intercept for driving is statistically larger than the others, indicating it is the most preferred alternative, *ceteris paribus*. The cost parameter is again negative and statistically significant, indicating that the cost of commuting reduces utility; this parameter is even larger than it was in previous models. All four time parameters are negative and statistically significant, indicating the time spent commuting by each mode reduces utility; the time parameters appear to be statistically different for most alternatives, with each minute driving a car yielding the greatest disutility and each minute riding the bus yielding the least disutility.

e. Conduct a likelihood ratio test on the model in part (d) to test that the alternative-specific coefficients on time are significantly different from one another. That is, test the null hypothesis:

$$H_0\colon\ \beta_k = \beta_b = \beta_c = \beta_w$$

Do you reject this null hypothesis? What is the p-value of the test?

```
### Part e
## Conduct likelihood ratio test of models 1a and 1b
test_1e <- test_likelihood_ratio(model_1b, model_1d)
## Display test results
test_1e

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

We reject this null hypothesis; the four alternative-specific time parameters are not equal. That is, the model in part (d) provides a better fit than the model in part (b), which restricted these parameters to be equal.

Problem 2: Nonlinear Least Squares

Use the `travel_binary.csv` dataset for this question.

a. Model the choice to drive to campus during winter as a binary logit model. Include the cost and the time of each alternative as explanatory variables with common coefficients; do not include an intercept. That is, the representative utility for alternative j is simply

$$V_{nj} = \beta_1 C_{nj} + \beta_2 T_{nj}$$

where C_j is the cost of alternative j and T_j is the time of alternative j. Estimate the parameters of this model by nonlinear least squares. The following steps can provide a rough guide to creating your own nonlinear least squares 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 for each alternative and 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. Sum the square of these residuals.
v. Return the sum of squares.

III. Minimize the sum of squares 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')
```

Report your parameter estimates and briefly interpret them. You do not have to estimate standard errors yet.

Hint: For a binary logit model, characterizing one choice probability is sufficient because the two probabilities must sum to 100%. The probability of driving is

$$\begin{aligned} P_{nc} &= \frac{e^{V_{nc}}}{e^{V_{nc}} + e^{V_{nb}}} \\ &= \frac{1}{1 + e^{-V_{nc}+V_{nb}}} \end{aligned}$$

Both of these expressions for the probability of driving may be useful as you solve this problem.

```
### Create function to calculate sum of squares
## Function to calculate binary logit sum of squares
calculate_sum_of_squares <- function(parameters, data_x, data_y){
  ## Calculate net utility of alternative given the parameters
  utility <- data_x %*% parameters
  ## Caclculate logit probability of alternative given the parameters
  probability_choice <- 1 / (1 + exp(-utility))
  ## Calculate sum of squares
  sum_of_squares <- sum((data_y - probability_choice)^2)
  return(sum_of_squares)
}

### Part a
## Load dataset
data_binary <- read_csv('travel_binary.csv')

## Parsed with column specification:
## cols(
##   mode = col_character(),
##   time_car = col_double(),
##   cost_car = col_double(),
##   time_bus = col_double(),
##   cost_bus = col_double(),
```

```
##   age = col_double(),
##   income = col_double(),
##   marital_status = col_character()
## )

## Create vector for choice outcomes
data_choice_2 <- data_binary %>%
  mutate(choice = 1 * (mode == 'car')) %>%
  pull(choice)
## Create matrix of explanatory variables
data_explanatory_2a <- data_binary %>%
  mutate(cost_difference = cost_car - cost_bus,
         time_difference = time_car - time_bus) %>%
  select(cost_difference, time_difference) %>%
  as.matrix()
## Minimize the sum of squares
model_2a <- optim(rep(0, 2), calculate_sum_of_squares,
                  data_x = data_explanatory_2a, data_y = data_choice_2,
                  method = 'BFGS')
## Show parameter estimates
list(names = c('cost', 'time'),
     parameters = model_2a$par)
```

Both the cost parameter and the time parameter are negative, indicating that the cost and time of commuting reduce utility.

b. Again model the choice to drive to campus during winter as a binary logit model, but now add an intercept term and alternative-specific coefficients. That is, the representative utilities for the alternatives are

$$V_{nc} = \beta_0 + \beta_1 C_{nc} + \beta_2 T_{nc}$$
$$V_{nb} = \beta_3 C_{nb} + \beta_4 T_{nb}$$

where C_{nj} is the cost of alternative j and T_{nj} is the time of alternative j. Estimate the parameters of this model by nonlinear least squares. Follow the same set of steps as in (a), but now you have five parameters and the representative utility calculation is different. You can use matrix multiplication in your sum of squares function to flexibly accommodate different models, or you can create a different function for each model. Report your parameter estimates and briefly interpret them. You do not have to estimate standard errors yet.

```r
### Part b
## Create matrix of explanatory variables
data_explanatory_2b <- data_binary %>%
  mutate(constant = 1,
         cost_bus = -cost_bus,
         time_bus = -time_bus) %>%
  select(constant, cost_car, time_car, cost_bus, time_bus) %>%
  as.matrix()
## Minimize the sum of squares
model_2b <- optim(rep(0, 5), calculate_sum_of_squares,
                  data_x = data_explanatory_2b, data_y = data_choice_2,
                  method = 'BFGS')
## Show parameter estimates
list(names = c('car', 'cost_car', 'time_car', 'cost_bus', 'time_bus'),
     parameters = model_2b$par)
```

```
## $names
## [1] "car"      "cost_car" "time_car" "cost_bus" "time_bus"
##
## $parameters
## [1]  2.1204264 -2.4413778 -0.5353962 -2.8834812 -0.2536593
```

The car intercept is positive, indicating that, *ceteris paribus*, driving a car is preferred to taking the bus. The alternative-specific parameters for cost and time are all negative, indicating that the cost and time of commuting by either alternative reduce utility.

c. Conduct a Wald test on the model in part (b) to test that the alternative-specific coefficients on cost are significantly different from one another. That is, test the null hypothesis:

$$H_0\colon\ \beta_1 = \beta_3$$

To conduct a Wald test, you need the variance-covariance matrix of your parameters estimates. For now, use the following parameter variances:

$$\begin{aligned}
Var(\beta_0) &= 1.61 \\
Var(\beta_1) &= 0.86 \\
Var(\beta_2) &= 0.09 \\
Var(\beta_3) &= 0.31 \\
Var(\beta_4) &= 0.03
\end{aligned}$$

and assume no covariances between parameters. The following steps can provide a rough guide to performing a Wald test:

I. Create a vector of J parameter restrictions, $r(\theta) = q$.

II. Calculate the $J \times K$ Jacobian matrix by differentiating each restriction with respect to each of the K parameters.

III. Calculate the Wald test statistic, which is a function of the vector of restrictions, the Jacobian matrix, and the variance-covariance matrix.

IV. Conduct the Wald test using this test statistic, which is distributed χ^2.

Do you reject the null hypothesis? What is the p-value of the test?

```
### Part c
## Create the variance-covariance matrix from the given standard erros
variance_covariance_2c <- c(1.61, 0.86, 0.09, 0.31, 0.03) %>%
  diag()
## Calculate the restriction vector
restriction_vector_2c <- model_2b$par[2] - model_2b$par[4]
## Construct the restriction Jacobian
restriction_jacobian_2c <- c(0, 1, 0, -1, 0) %>%
  t()
## Calculate the Wald test statistic
wald_test_stat_2c <- t(restriction_vector_2c) %*%
  solve(restriction_jacobian_2c %*%
          variance_covariance_2c %*%
          t(restriction_jacobian_2c)) %*%
  restriction_vector_2c %>%
  c()
## Test if Wald test statitsic is greater than critical value
reject_2c <- wald_test_stat_2c > qchisq(0.95, 1)
## Calculate p-value of Wald test
p_value_2c <- 1 - pchisq(wald_test_stat_2c, 1)
## Report Wald test results
list(reject = reject_2c,
     p_value = p_value_2c)

## $reject
## [1] FALSE
##
## $p_value
## [1] 0.6827417
```

We cannot reject this null hypothesis that the cost parameters are equal. This result is intuitive; a dollar spent on driving and a dollar spent on riding the bus are identical and there is no reason why the marginal utility of those dollars should differ. Thus, we could simplify the model by using a common cost parameter.

d. Estimate the variance-covariance matrix for the model in part (a). The following steps can provide a rough guide to estimating the variance-covariance matrix for nonlinear least squares estimators:

   I. Write down the derivative of the nonlinear regression model (in this case, the logit choice probability for driving) with respect to each of the K parameters.
   II. Calculate this $K \times 1$ vector of derivatives, at the estimated parameters, for each decision maker.
   III. Calculate the $K \times K$ matrix that is the product of the above vector and its transpose for each decision maker.
   IV. Sum these matrices for all decision makers.

V. Estimate the variance of the econometric error as the mean sum of squares at the estimated parameters.
VI. Calculate the variance-covariance matrix, which is a function of the above $K \times K$ matrix and the estimated error variance.

Report your parameter estimates (from (a)), standard errors, t-stats, and p-values. Briefly interpret these results.

```
### Part d
## Calculate net utility of driving at NLS parameters
utility_2d <- data_explanatory_2a %*% model_2a$par
## Caclculate logit probability of driving at NLS parameters
probability_2d <- 1 / (1 + exp(-utility_2d))
## Create list of individual derivative vectors
derivative_vector_list_2d <- data_explanatory_2a %>%
  split(row(.)) %>%
  map2(as.list(probability_2d), ~ .x * .y * (1 - .y))
## Create list of individual derivative product matrices
derivative_matrix_list_2d <- derivative_vector_list_2d %>%
  map(~ .x %*% t(.x))
## Sum individual derivative product matrices
derivative_matrix_2d <- derivative_matrix_list_2d %>%
  reduce(`+`)
## Calculate error variance at NLS parameters
error_variance_2d <- calculate_sum_of_squares(model_2a$par,
                                              data_explanatory_2a,
                                              data_choice_2) /
  length(data_choice_2)
## Calculate variance-covariance matrix
variance_covariance_2d <- error_variance_2d * solve(derivative_matrix_2d)
## Calculate parameter standard erros
standard_errors_2d <- variance_covariance_2d %>%
  diag() %>%
  sqrt()
## Calculate parameter t-stats
t_stats_2d <- model_2a$par / standard_errors_2d
## Calculate parameter p-values
p_values_2d <- 2 * pt(-abs(t_stats_2d),
                      length(data_choice_2) - length(model_2a$par))
## Report summary of model results
list(names = c('cost', 'time'),
     parameters = model_2a$par,
     std_errors = standard_errors_2d,
     t_stats = t_stats_2d,
     p_values = p_values_2d)


## $names
## [1] "cost" "time"
```

```
##
## $parameters
## [1] -0.41262415 -0.06482335
##
## $std_errors
## [1] 0.086585198 0.008764028
##
## $t_stats
## [1] -4.765528 -7.396525
##
## $p_values
## [1] 2.163055e-06 2.959150e-13
```

As before, the cost parameter and the time parameter are negative. We now see both parameters are statistically significant, so the initial interpretation remains correct.

e. Estimate the variance-covariance matrix for the model in part (b). Follow the same set of steps as in (d), but now you have five parameters to consider. Report your parameter estimates (from (b)), standard errors, t-stats, and p-values. Briefly interpret these results. Additionally, perform the Wald test from part (c) using your estimated variance-covariance matrix. Do you reject the null hypothesis? What is the p-value of the test?

```
### Part e
## Calculate net utility of driving at NLS parameters
utility_2e <- data_explanatory_2b %*% model_2b$par
## Caclculate logit probability of driving at NLS parameters
probability_2e <- 1 / (1 + exp(-utility_2e))
## Create list of individual derivative vectors
derivative_vector_list_2e <- data_explanatory_2b %>%
  split(row(.)) %>%
  map2(as.list(probability_2e), ~ .x * .y * (1 - .y) * c(1, 1, 1, -1, -1))
## Create list of individual derivative product matrices
derivative_matrix_list_2e <- derivative_vector_list_2e %>%
  map(~ .x %*% t(.x))
## Sum individual derivative product matrices
derivative_matrix_2e <- derivative_matrix_list_2e %>%
  reduce(`+`)
## Calculate error variance at NLS parameters
error_variance_2e <- calculate_sum_of_squares(model_2b$par,
                                              data_explanatory_2b,
                                              data_choice_2) /
  length(data_choice_2)
## Calculate variance-covariance matrix
variance_covariance_2e <- error_variance_2e * solve(derivative_matrix_2e)
## Calculate parameter standard erros
standard_errors_2e <- variance_covariance_2e %>%
  diag() %>%
  sqrt()
```

```r
## Calculate parameter t-stats
t_stats_2e <- model_2b$par / standard_errors_2e
## Calculate parameter p-values
p_values_2e <- 2 * pt(-abs(t_stats_2e),
                      length(data_choice_2) - length(model_2b$par))
## Report summary of model results
list(names = c('car', 'cost_car', 'time_car', 'cost_bus', 'time_bus'),
     parameters = model_2b$par,
     std_errors = standard_errors_2e,
     t_stats = t_stats_2e,
     p_values = p_values_2e)

## $names
## [1] "car"      "cost_car" "time_car" "cost_bus" "time_bus"
##
## $parameters
## [1]  2.1204264 -2.4413778 -0.5353962 -2.8834812 -0.2536593
##
## $std_errors
## [1] 1.60537805 0.86494518 0.08746187 0.31250299 0.03247035
##
## $t_stats
## [1]  1.320827 -2.822581 -6.121482 -9.227052 -7.812028
##
## $p_values
## [1] 1.868628e-01 4.858813e-03 1.332118e-09 1.624083e-19 1.424491e-14

## Use the restriction vector from 2c
restriction_vector_2e <- restriction_vector_2c
## Use the restriction Jacobian from 2c
restriction_jacobian_2e <- restriction_jacobian_2c
## Calculate the Wald test statistic
wald_test_stat_2e <- t(restriction_vector_2e) %*%
  solve(restriction_jacobian_2e %*%
          variance_covariance_2e %*%
          t(restriction_jacobian_2e)) %*%
  restriction_vector_2e %>%
  c()
## Test if Wald test statitsic is greater than critical value
reject_2e <- wald_test_stat_2e > qchisq(0.95, 1)
## Calculate p-value of Wald test
p_value_2e <- 1 - pchisq(wald_test_stat_2e, 1)
## Report Wald test results
list(reject = reject_2e,
     p_value = p_value_2e)

## $reject
```

```
## [1] FALSE
##
## $p_value
## [1] 0.6681799
```

As before, the car intercept is positive and the alternative-specific parameters for cost and time are all negative. We now see all parameters are statistically significant, so the initial interpretation remains correct. We again fail to reject the null hypothesis that the cost parameters are equal.