R Estimate Logistic Choice Model with Aggregate Shares

Fan Wang

2021-06-08

Contents

1 Logistic Choice Model with Aggregate Shares

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

1.1 Logistic Choices, Wages and Market Shares

1.1.1 Wage and Aggregate Share of Workers

Individual i can choose among $M+1$ options, the M options, are for example, alternative job possibilities. The $+1$ refers to a leisure category. The value associated with each one of the choice alternatives is:

$$V_{itm} = \widehat{V}_{tm} + \epsilon_{itm} = \alpha_m + \beta \cdot \text{WAGE}_{tm} + \epsilon_{itm}$$

Note that β captures the effect of occupation-specific wage, β is the same across occupational groups. Each occupational group has its own intercept α_m, and ϵ_{im} is the individual and occupation specific extreme value error. The non-error component of the value of leisure is normalized to 0 ($\exp(0) = 1$).

The discrete choice problem is solved by a comparison across alternatives. o_i is the individual optimal choice $o_i = \arg\max_m \left(V_{i,m}\right)$

Choice probabilities are functions of wages. The probability that individual i chooses occupation m is (ignoring the t-subscripts):

$$P(o=m) = \frac{\exp\left(\widehat{V}_m\right)}{1+\sum_{\hat{m}=1}^{M}\exp\left(\widehat{V}_{\hat{m}}\right)}$$

The log ratio of probability of choosing any of the M occupation alternatives and leisure is:

$$\log P\left(o=m\right) - \log P\left(o=0\right) = \log\left(\frac{P\left(o=m\right)}{P\left(o=0\right)}\right) = \log\left(\frac{\exp\left(\widehat{V}_m\right)}{1+\sum_{\hat{m}=1}^{M}\exp\left(\widehat{V}_{\hat{m}}\right)} \cdot \frac{1+\sum_{\hat{m}=1}^{M}\exp\left(\widehat{V}_{\hat{m}}\right)}{1}\right)$$

This means that the log of the probability ratio is linear in wages:

$$\log\left(\frac{P\left(o=m\right)}{P\left(o=0\right)}\right) = \alpha_m + \beta\cdot\text{WAGE}_m$$

Supose we have $M-1$, either work or leisure. Suppose we have aggregate data from 1 time period, with relative work to leisure share (aggregate-share data, "market-share" data) and a single measure of wage, then we have 1 equation with two parameters, α and β can not be jointly identified. α and β, neither of which is time-varying, are exactly identified if we have data from two periods with variations in wage across the periods.

1.1.2 Simulate Market Share

In this section, we now simulate the above model, with $M=2$ and data over three periods, and estimate α and β via OLS. Note that $m=0$ is leisure.

First, simulate the data.

```r
# set seed
set.seed(123)
# T periods, and M occupations (+1 leisure)
it_T <- 3
it_M <- 2
# define alpha and beta parameters
ar_alpha <- runif(it_M) + 0
fl_beta <- runif(1) + 0
# wage matrix, no wage for leisure
mt_wages <- matrix(runif(it_T*it_M) + 1, nrow=it_T, ncol=it_M)
colnames(mt_wages) <- paste0('m', seq(1,it_M))
rownames(mt_wages) <- paste0('t', seq(1,it_T))
# Define a probability assignment function
ffi_logit_prob_alpha_beta <- function(ar_alpha, fl_beta, mt_wages) {
  # Dimensions
  it_T <- dim(mt_wages)[1]
  it_M <- dim(mt_wages)[2]
  # Aggregate probabilities
  mt_prob_o <- matrix(data=NA, nrow=it_T, ncol=it_M+1)
  colnames(mt_prob_o) <- paste0('m', seq(0,it_M))
  rownames(mt_prob_o) <- paste0('t', seq(1,it_T))
  # Generate Probabilities
  for (it_t in seq(1, it_T)) {
    # get current period wages
    ar_wage_at_t <- mt_wages[it_t, ]
    # Value without shocks/errors, M+1
    ar_V_hat_at_t <- c(0, ar_alpha + fl_beta*ar_wage_at_t)
    # Probabilities across M+1
    fl_prob_denominator <- sum(exp(ar_V_hat_at_t))
    ar_prob_at_t <- exp(ar_V_hat_at_t)/fl_prob_denominator
    # Fill in
    mt_prob_o[it_t,] <- ar_prob_at_t
  }
  return(mt_prob_o)
}
# Show probabilities
mt_prob_o <- ffi_logit_prob_alpha_beta(ar_alpha, fl_beta, mt_wages)
st_caption <- 'Occupation aggregate participation probabilities across time'
kable(mt_prob_o, caption=st_caption) %>% kable_styling_fc()
```

Occupation aggregate participation probabilities across time

	m0	m1	m2
t1	0.1251712	0.3604563	0.5143725
t2	0.1147084	0.3381795	0.5471121
t3	0.1390180	0.2842316	0.5767504

```
# mt_prob_o
```

1.1.3 Create Regression Data Inputs

Second, generate relative market shares of various work occupations, columns 2 through $M + 1$, with respect to leisure, column 1. If there are M categories and T time periods, there are $M \times T$ observations (rows). Flatten the structure so that row-groups are the M categories, and we have time-specific information within row groups.

```
# A Matrix with share from 1:M columns
mt_prob_rela_m2leisure <-
  matrix(data=mt_prob_o[1:it_T, 2:(it_M+1)], nrow=it_T, ncol=it_M)
colnames(mt_prob_rela_m2leisure) <- paste0('m', seq(1,it_M))
rownames(mt_prob_rela_m2leisure) <- paste0('t', seq(1,it_T))
# Divide 1:M by leisure
mt_prob_rela_m2leisure <- mt_prob_rela_m2leisure/mt_prob_o[1:it_T, 1]
# Take Logs, log(Pm/Po)
mt_prob_rela_m2leisure_log <- log(mt_prob_rela_m2leisure)
# Flatten to single column
ar_prob_ols_output <- matrix(data=mt_prob_rela_m2leisure_log, nrow=it_T*it_M, ncol=1)
# Show probabilities
st_caption <- 'Log of relative participation probabilities against leisure'
kable(mt_prob_rela_m2leisure_log, caption=st_caption) %>% kable_styling_fc()
```

Log of relative participation probabilities against leisure

	m1	m2
t1	1.057688	1.413265
t2	1.081184	1.562261
t3	0.715186	1.422806

Third, construct the estimation input matrices. If there are M categories and T time periods, there are

$M \times T$ observations (rows). Flatten in the same way as for outcomes. There are M indicator vectors and 1 wage vector for data inputs:

- α is a fixed effect that is m-specific, so we have as Data, M indicator vectors.
- The wage variable is shared by all, so a single data column

```
# Regression input matrix
mt_prob_ols_input <- matrix(data=NA, nrow=it_T*it_M, ncol=it_M+1)
colnames(mt_prob_ols_input) <- paste0('m', seq(1,dim(mt_prob_ols_input)[2]))
rownames(mt_prob_ols_input) <- paste0('rowMcrsT', seq(1,dim(mt_prob_ols_input)[1]))
# Generate index position in ols input matrix for M indicators
mt_m_indicators <- matrix(data=NA, nrow=it_T*it_M, ncol=it_M)
colnames(mt_prob_ols_input) <- paste0('m', seq(1,dim(mt_prob_ols_input)[2]))
rownames(mt_prob_ols_input) <- paste0('rowMcrsT', seq(1,dim(mt_prob_ols_input)[1]))
# loop over columns
for (it_indix in seq(1, it_M)) {
  # loop over rows
  for (it_rowMcrsT in seq(1, it_T*it_M)) {
    if ((it_rowMcrsT >= (it_indix-1)*(it_T) + 1) && it_rowMcrsT <= it_indix*(it_T)){
      mt_m_indicators[it_rowMcrsT, it_indix] <- 1
    } else {
      mt_m_indicators[it_rowMcrsT, it_indix] <- 0
    }
  }
}
# Indicators are the earlier columns
mt_prob_ols_input[, 1:it_M] <- mt_m_indicators
# Wage is the last column
mt_prob_ols_input[, it_M+1] <- matrix(data=mt_wages, nrow=it_T*it_M, ncol=1)
```

Fourth, combine left-hand-side and right-hand-side regression input data structures/

```
# Construct data structure
mt_all_inputs <- cbind(ar_prob_ols_output, mt_prob_ols_input)
colnames(mt_all_inputs)[1] <- 'log_pm_over_po'
colnames(mt_all_inputs)[dim(mt_all_inputs)[2]] <- 'wages'
tb_all_inputs <- as_tibble(mt_all_inputs)
# Show data
st_caption <- 'LHS=Log Probability Ratios (column 1); RHS=Indicators of occupations and wages, OLS inpu
kable(tb_all_inputs, caption=st_caption) %>% kable_styling_fc()
```

LHS=Log Probability Ratios (column 1); RHS=Indicators of occupations and wages, OLS inputs (other columns)

log_pm_over_po	m1	m2	wages
1.057688	1	0	1.883017
1.081184	1	0	1.940467
0.715186	1	0	1.045556
1.413265	0	1	1.528105
1.562261	0	1	1.892419
1.422806	0	1	1.551435

1.1.4 Estimate Wage Coefficients

Fifth, estimate the OLS equation, and compare estimate to true parameters.

```r
# Regression
fit_ols_agg_prob <- lm(log_pm_over_po ~ . -1, data = tb_all_inputs)
summary(fit_ols_agg_prob)
```

```
## 
## Call:
## lm(formula = log_pm_over_po ~ . - 1, data = tb_all_inputs)
## 
## Residuals:
##          1          2          3          4          5          6 
##  4.027e-17  2.276e-17 -6.303e-17 -6.481e-17 -1.631e-16  2.279e-16 
## 
## Coefficients:
##        Estimate Std. Error   t value Pr(>|t|)    
## m1    2.876e-01  3.784e-16 7.599e+14   <2e-16 ***
## m2    7.883e-01  3.859e-16 2.043e+15   <2e-16 ***
## wages 4.090e-01  2.250e-16 1.818e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.721e-16 on 3 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1 
## F-statistic: 1.043e+32 on 3 and 3 DF,  p-value: < 2.2e-16
```

```r
# alpha estimates
ar_coefficients <- fit_ols_agg_prob$coefficients
ar_alpha_esti <- ar_coefficients[1:(length(ar_coefficients)-1)]
fl_beta_esti <- ar_coefficients[length(ar_coefficients)]
# Compare estimates and true
print(paste0('ar_alpha_esti=', ar_alpha_esti))
```

```
## [1] "ar_alpha_esti=0.287577520124614" "ar_alpha_esti=0.788305135443806"
```

```r
print(paste0('ar_alpha=', ar_alpha))
```

```
## [1] "ar_alpha=0.287577520124614" "ar_alpha=0.788305135443807"
```

```r
print(paste0('fl_beta_esti=', fl_beta_esti))
```

```
## [1] "fl_beta_esti=0.4089769218117"
```

```r
print(paste0('fl_beta=', fl_beta))
```

```
## [1] "fl_beta=0.4089769218117"
```

```r
# Simulate given estimated parameters using earlier function
mt_prob_o_esti <- ffi_logit_prob_alpha_beta(ar_alpha_esti, fl_beta_esti, mt_wages)
# Results
st_caption <- 'Predicted probabilities based on estimates'
kable(mt_prob_o_esti, caption=st_caption) %>% kable_styling_fc()
```

Predicted probabilities based on estimates

	m0	m1	m2
t1	0.1251712	0.3604563	0.5143725
t2	0.1147084	0.3381795	0.5471121
t3	0.1390180	0.2842316	0.5767504

```
# Results
st_caption <- 'Compare differences in probability predictions based on estimates and true probabilities
kable(mt_prob_o_esti-mt_prob_o, caption=st_caption) %>% kable_styling_fc()
```

Compare differences in probability predictions based on estimates and true probabilities

	m0	m1	m2
t1	0	0	0
t2	0	0	0
t3	0	0	0

1.2 Logistic Choices with Time-specific Observables

In our example here, there are choice alternatives (m) and time periods (t). In terms of data inputs, in the prior example, we had on the Right

In the example in the prior section, the data structure included:

1. m-specific variables: indicators
2. m- and t-specific variables: wages

In the new data structure for this section, we have:

1. m-specific variables: indicators
2. m- and t-specific variables: wages
3. t-specific variables: characteristics that are homogeneous across groups, but varying over time. This will include both a trend variable capturing time patterns, as well as an observable that is different over time.

Suppose there is some information that impacts individual's willingness to stay at home. Note that leisure is the same as work from home. For example, there could be technological improvements that makes home-production less time-consuming, and we might have a variable that captures the efficiency of home-technology. Captured by a time trend, there might also be changes in social attitudes, which could be harder to measure.

Suppose we still have:

$$V_{itm} = \widehat{V}_{tm} + \epsilon_{itm} = \alpha_m + \beta \cdot \text{WAGE}_{tm} + \epsilon_{itm}$$

But now, for the leisure category we have:

$$V_{it0} = \widehat{V}_{t0} + \epsilon_{it0} = \alpha_0 + \phi \cdot t + \theta \cdot \text{HOMETECH}_t + \epsilon_{it0}$$

Because only the differences in value across choices matter, so we can continue with the prior normalization, in difference, we have:

$$V_{itm} - V_{it0} = (\alpha_m - \alpha_0) + (\beta \cdot \text{WAGE}_{tm} - \phi \cdot t - \theta \cdot \text{HOMETECH}_t) + (\epsilon_{itm} - \epsilon_{it0})$$

Note that:

- α_0 is not identifiable.
- If the HOMETECH or time variables are multiplied by negative one, the signs changes.
- Since only the differences in value matter, the coefficients for these variables represent the net changes on alternative value difference due to changes in time-trend or the HOMETECH variable.

1.2.1 Simulate Market Share

We simulate market share data now, with information over several more periods, and the HOMETECH variable, along with the time trend.

First, simulate the data.

```
# set seed
set.seed(123)
# T periods, and M occupations (+1 leisure/home)
it_T <- 5
it_M <- 4
# define alpha and beta parameters
ar_alpha <- runif(it_M)*0.5
fl_beta <- runif(1)*0.25
# also negative time-trend from home category perspective, culturally increasingly accepting of work
fl_phi <- -runif(1)*0.50
# home-tech is negative from home category perspective, higher home-tech, more chance to work
fl_theta <- -runif(1)*1
# wage matrix, no wage for leisure
mt_wages <- matrix(runif(it_T*it_M) + 1, nrow=it_T, ncol=it_M)
# HOMETECH changes, random and sorted in ascending order, increasing over time
ar_hometech <- sort(runif(it_T))
colnames(mt_wages) <- paste0('m', seq(1,it_M))
rownames(mt_wages) <- paste0('t', seq(1,it_T))
# Define a probability assignment function
ffi_logit_prob2_alpha_beta <- function(ar_alpha, fl_beta, fl_phi, fl_theta, ar_hometech, mt_wages) {
  # Dimensions
  it_T <- dim(mt_wages)[1]
  it_M <- dim(mt_wages)[2]
  # Aggregate probabilities
  mt_prob_o <- matrix(data=NA, nrow=it_T, ncol=it_M+1)
  colnames(mt_prob_o) <- paste0('m', seq(0,it_M))
  rownames(mt_prob_o) <- paste0('t', seq(1,it_T))
  # Generate Probabilities
  for (it_t in seq(1, it_T)) {
    # get current period wages
    ar_wage_at_t <- mt_wages[it_t, ]
    # Value without shocks/errors, M+1
    ar_V_hat_at_t <- c(0, ar_alpha + fl_beta*ar_wage_at_t - fl_phi*it_t - fl_theta*ar_hometech[it_t] )
    # Probabilities across M+1
    fl_prob_denominator <- sum(exp(ar_V_hat_at_t))
    ar_prob_at_t <- exp(ar_V_hat_at_t)/fl_prob_denominator
    # Fill in
    mt_prob_o[it_t,] <- ar_prob_at_t
  }
  return(mt_prob_o)
}
# Show probabilities
mt_prob_o2 <- ffi_logit_prob2_alpha_beta(ar_alpha, fl_beta, fl_phi, fl_theta, ar_hometech, mt_wages)
st_caption <- 'Occupation aggregate participation probabilities across time: time-varying observables, 
kable(mt_prob_o2, caption=st_caption) %>% kable_styling_fc()
```

Occupation aggregate participation probabilities across time: time-varying observables, time- and occupation-specific wages, occupation-specific intercepts; note reduction in m0, home-activity share over time

	m0	m1	m2	m3	m4
t1	0.1032335	0.2056532	0.2511476	0.1789233	0.2610423
t2	0.0932896	0.1891478	0.2441729	0.1906952	0.2826945
t3	0.0805830	0.1920307	0.2269798	0.2293885	0.2710180
t4	0.0633282	0.2043484	0.2589892	0.2137280	0.2596062
t5	0.0653460	0.1978795	0.2420864	0.2224408	0.2722473

```
# mt_prob_o
```

1.2.2 Create Regression Data Inputs

Second, generate relative market shares of various work occupations as prior

```
mt_prob_rela_m2leisure2 <- matrix(data=mt_prob_o2[1:it_T, 2:(it_M+1)], nrow=it_T, ncol=it_M)
ar_prob_ols_output2 <- matrix(data=log(mt_prob_rela_m2leisure2/mt_prob_o2[1:it_T, 1]), nrow=it_T*it_M, n
```

Third, construct the estimation input matrices as before. There are M indicator vectors, a time variable, a HOMETECH variable, and 1 wage vector for data inputs:

```
# Regression input matrix
mt_prob_ols_input2 <- matrix(data=NA, nrow=it_T*it_M, ncol=it_M+2+1)
colnames(mt_prob_ols_input2) <- paste0('m', seq(1,dim(mt_prob_ols_input2)[2]))
rownames(mt_prob_ols_input2) <- paste0('rowMcrsT', seq(1,dim(mt_prob_ols_input2)[1]))
# Generate index position in ols input matrix for M indicators
mt_m_indicators <- matrix(data=NA, nrow=it_T*it_M, ncol=it_M)
for (it_indix in seq(1, it_M)) {
  for (it_rowMcrsT in seq(1, it_T*it_M)) {
    if ((it_rowMcrsT >= (it_indix-1)*(it_T) + 1) && it_rowMcrsT <= it_indix*(it_T)){
      mt_m_indicators[it_rowMcrsT, it_indix] <- 1
    } else {
      mt_m_indicators[it_rowMcrsT, it_indix] <- 0
    }
  }
}
# Indicators are the earlier columns
mt_prob_ols_input2[, 1:it_M] <- mt_m_indicators
# Time variable column
ar_time <- matrix(seq(1,it_T), nrow=it_T*it_M, ncol=1)
mt_prob_ols_input2[, it_M+1] <- ar_time
# Home Tech Column
ar_hometech_mesh <- matrix(ar_hometech, nrow=it_T*it_M, ncol=1)
mt_prob_ols_input2[, it_M+2] <- ar_hometech_mesh
# Wage is the last column
mt_prob_ols_input2[, it_M+3] <- matrix(data=mt_wages, nrow=it_T*it_M, ncol=1)
```

Fourth, combine left-hand-side and right-hand-side regression input data structures/

```
# Construct data structure
mt_all_inputs2 <- cbind(ar_prob_ols_output2, mt_prob_ols_input2)
colnames(mt_all_inputs2)[1] <- 'log_pm_over_po'
colnames(mt_all_inputs2)[dim(mt_all_inputs2)[2]-2] <- 'time'
colnames(mt_all_inputs2)[dim(mt_all_inputs2)[2]-1] <- 'hometech'
colnames(mt_all_inputs2)[dim(mt_all_inputs2)[2]] <- 'wages'
tb_all_inputs2 <- as_tibble(mt_all_inputs2)
# Show data
st_caption <- 'LHS=Log Probability Ratios (column 1); RHS=Indicators of occupations and other variables
kable(tb_all_inputs2, caption=st_caption) %>% kable_styling_fc()
```

LHS=Log Probability Ratios (column 1); RHS=Indicators of occupations and other variables, OLS inputs (other columns)

log_pm_over_po	m1	m2	m3	m4	time	hometech	wages
0.6891981	1	0	0	0	1	0.1471136	1.892419
0.7068206	1	0	0	0	2	0.2891597	1.551435
0.8683678	1	0	0	0	3	0.5941420	1.456615
1.1714953	1	0	0	0	4	0.9022990	1.956833
1.1079617	1	0	0	0	5	0.9630242	1.453334
0.8890474	0	1	0	0	1	0.1471136	1.677571
0.9621685	0	1	0	0	2	0.2891597	1.572633
1.0355731	0	1	0	0	3	0.5941420	1.102925
1.4084555	0	1	0	0	4	0.9022990	1.899825
1.3095984	0	1	0	0	5	0.9630242	1.246088
0.5499640	0	0	1	0	1	0.1471136	1.042059
0.7149683	0	0	1	0	2	0.2891597	1.327921
1.0461296	0	0	1	0	3	0.5941420	1.954504
1.2163730	0	0	1	0	4	0.9022990	1.889539
1.2249646	0	0	1	0	5	0.9630242	1.692803
0.9276892	0	0	0	1	1	0.1471136	1.640507
1.1086584	0	0	0	1	2	0.2891597	1.994270
1.2128974	0	0	0	1	3	0.5941420	1.655706
1.4108350	0	0	0	1	4	0.9022990	1.708530
1.4270142	0	0	0	1	5	0.9630242	1.544066

1.2.3 Estimate Wage Coefficients, HOMETECH Effects and Time Trend

Similar to before, estimate with OLS.

```
# Regression
fit_ols_agg_prob2 <- lm(log_pm_over_po ~ . -1, data = tb_all_inputs2)
summary(fit_ols_agg_prob2)
```

```
##
## Call:
## lm(formula = log_pm_over_po ~ . - 1, data = tb_all_inputs2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.670e-16 -8.439e-17 -2.664e-17  9.403e-17  2.704e-16
##
## Coefficients:
##           Estimate Std. Error   t value Pr(>|t|)
## m1       1.438e-01  3.165e-16 4.543e+14   <2e-16 ***
## m2       3.942e-01  2.927e-16 1.347e+15   <2e-16 ***
## m3       2.045e-01  3.046e-16 6.713e+14   <2e-16 ***
## m4       4.415e-01  3.234e-16 1.365e+15   <2e-16 ***
## time     2.278e-02  1.568e-16 1.453e+14   <2e-16 ***
## hometech 5.281e-01  6.900e-16 7.653e+14   <2e-16 ***
## wages    2.351e-01  1.615e-16 1.456e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.741e-16 on 13 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.098e+32 on 7 and 13 DF,  p-value: < 2.2e-16
```

```
# alpha estimates
ar_coefficients2 <- fit_ols_agg_prob2$coefficients
ar_alpha_esti2 <- ar_coefficients2[1:(length(ar_coefficients2)-3)]
# time -1 because of the sign structure
fl_phi_esti2 <- -1*ar_coefficients2[length(ar_coefficients2)-2]
# time -1 because of the sign structure
fl_theta_esti2 <- -1*ar_coefficients2[length(ar_coefficients2)-1]
fl_beta_esti2 <- ar_coefficients2[length(ar_coefficients2)]
# Compare estimates and true
print(paste0('ar_alpha_esti2=', ar_alpha_esti2))
```

```
## [1] "ar_alpha_esti2=0.143788760062308" "ar_alpha_esti2=0.394152567721904" "ar_alpha_esti2=0.204488460
```

```
print(paste0('ar_alpha=', ar_alpha))
```

```
## [1] "ar_alpha=0.143788760062307" "ar_alpha=0.394152567721903" "ar_alpha=0.20448846090585"  "ar_alpha=
```

```
print(paste0('fl_theta_esti2=', fl_theta_esti2))
```

```
## [1] "fl_theta_esti2=-0.528105488047005"
```

```
print(paste0('fl_theta=', fl_theta))
```

```
## [1] "fl_theta=-0.528105488047004"
```

```
print(paste0('fl_phi_esti2=', fl_phi_esti2))
```

```
## [1] "fl_phi_esti2=-0.0227782496949655"
```

```
print(paste0('fl_phi=', fl_phi))
```

```
## [1] "fl_phi=-0.0227782496949658"
```

```
print(paste0('fl_beta_esti2=', fl_beta_esti2))
```

```
## [1] "fl_beta_esti2=0.235116821073461"
```

```
print(paste0('fl_beta=', fl_beta))
```

```
## [1] "fl_beta=0.235116821073461"
```

```
# Simulate given estimated parameters using earlier function
mt_prob_o2_esti <- ffi_logit_prob2_alpha_beta(ar_alpha_esti2, fl_beta_esti2, fl_phi_esti2, fl_theta_esti
# Results
st_caption <- 'Predicted probabilities based on estimates'
kable(mt_prob_o2_esti, caption=st_caption) %>% kable_styling_fc()
```

```
# Results
st_caption <- 'Compare differences in probability predictions based on estimates and true probabilities
kable(mt_prob_o2_esti-mt_prob_o2, caption=st_caption) %>% kable_styling_fc()
```

Predicted probabilities based on estimates

	m0	m1	m2	m3	m4
t1	0.1032335	0.2056532	0.2511476	0.1789233	0.2610423
t2	0.0932896	0.1891478	0.2441729	0.1906952	0.2826945
t3	0.0805830	0.1920307	0.2269798	0.2293885	0.2710180
t4	0.0633282	0.2043484	0.2589892	0.2137280	0.2596062
t5	0.0653460	0.1978795	0.2420864	0.2224408	0.2722473

Compare differences in probability predictions based on estimates and true probabilities

	m0	m1	m2	m3	m4
t1	0	0	0	0	0
t2	0	0	0	0	0
t3	0	0	0	0	0
t4	0	0	0	0	0
t5	0	0	0	0	0

1.3 Logistic Choices and Multiple Types of Workers

In the prior two examples, we had the same type of worker, choosing among multiple occupations. There could be different types of workers, unskilled and skilled workers, both of whom can choose among the same set of job possibilities. If the two types of workers do not share any parameters in the occupation-choice problems, then we can estimate their problems separately. However, perhaps on the stay-at-home front, they share the same time trend, which captures overall cultural changes to work vs not-work. And perhaps the wage parameter is the same. On other fronts, they are different with different parameters as well as data inputs.

So we have:

$$V_{istm} - V_{ist0} = (\alpha_{sm} - \alpha_{s0}) + (\beta \cdot \text{WAGE}_{stm} - \phi \cdot t - \theta_s \cdot \text{HOMETECH}_{st}) + (\epsilon_{istm} - \epsilon_{ist0})$$

In the new data structure for this section, we have:

1. k- and m-specific variables: indicators
2. k- and m- and t-specific variable: wages
3. t-specific variable: dates
4. k- and t-specific variable: hometech

1.3.1 Simulate Market Share

In the simulation example, store all K types and T time periods as rows, and the M occupational types as columns.

```r
# set seed
set.seed(123)
# K skill levels, T periods, and M occupations (+1 leisure/home)
it_K <- 2
it_T <- 3
it_M <- 4
# define alpha and beta parameters
mt_alpha <- matrix(runif(it_K*it_M)*0.5, nrow=it_K, ncol=it_M)
colnames(mt_alpha) <- paste0('m', seq(1,dim(mt_alpha)[2]))
rownames(mt_alpha) <- paste0('k', seq(1,dim(mt_alpha)[1]))
fl_beta <- runif(1)*0.25
```

```r
# also negative time-trend from home category perspective, culturally increasingly accepting of work
fl_phi <- -runif(1)*0.50
# home-tech is negative from home category perspective, higher home-tech, more chance to work
ar_theta <- -runif(it_K)*1
# wage matrix, no wage for leisure
mt_wages <- matrix(runif(it_K*it_T*it_M) + 1, nrow=it_K*it_T, ncol=it_M)
colnames(mt_wages) <- paste0('m', seq(1,it_M))
rownames(mt_wages) <- paste0('kxt', seq(1,it_K*it_T))
# HOMETECH changes, random and sorted in ascending order, increasing over time, specific to each k
mt_hometech <- sapply(seq(1,it_K), function(i){sort(runif(it_T))})
colnames(mt_hometech) <- paste0('k', seq(1,dim(mt_hometech)[2]))
rownames(mt_hometech) <- paste0('t', seq(1,dim(mt_hometech)[1]))
# Define a probability assignment function
ffi_logit_prob3_alpha_beta <- function(it_K, it_T, it_M,
  mt_alpha, fl_beta, fl_phi, ar_theta, mt_hometech, mt_wages) {

  # Aggregate probabilities
  mt_prob_o <- matrix(data=NA, nrow=it_K*it_T, ncol=it_M+3)
  colnames(mt_prob_o) <- c('skillgroup', 'time', paste0('m', seq(0,it_M)))
  rownames(mt_prob_o) <- paste0('kxt', seq(1,it_K*it_T))
  # Generate Probabilities
  it_kxt_ctr <- 0
  for (it_k in seq(1, it_K)) {
    for (it_t in seq(1, it_T)) {
      # Counter updating
      it_kxt_ctr <- it_kxt_ctr + 1
      # get current period wages
      ar_wage_at_t <- mt_wages[it_kxt_ctr, ]
      # Value without shocks/errors, M+1
      ar_V_hat_at_t <- c(0, mt_alpha[it_k,] + fl_beta*ar_wage_at_t - fl_phi*it_t - ar_theta[it_k]*mt_hon
      # Probabilities across M+1
      fl_prob_denominator <- sum(exp(ar_V_hat_at_t))
      ar_prob_at_t <- exp(ar_V_hat_at_t)/fl_prob_denominator
      # Fill in
      mt_prob_o[it_kxt_ctr,1] <- it_k
      mt_prob_o[it_kxt_ctr,2] <- it_t
      mt_prob_o[it_kxt_ctr,3:dim(mt_prob_o)[2]] <- ar_prob_at_t
      # row name
      rownames(mt_prob_o)[it_kxt_ctr] <- paste0('rk', it_k, 't', it_t)
    }
  }
  return(mt_prob_o)

}
# Show probabilities
mt_prob_o3_full <- ffi_logit_prob3_alpha_beta(it_K, it_T, it_M,
  mt_alpha, fl_beta, fl_phi, ar_theta, mt_hometech, mt_wages)
# Selected only probability columns
mt_prob_o3 <- mt_prob_o3_full[, 3:dim(mt_prob_o3_full)[2]]
st_caption <- 'Occupation aggregate participation probabilities across time: skill- and time-varying obs
kable(mt_prob_o3_full, caption=st_caption) %>% kable_styling_fc()
```

```r
# mt_prob_o
```

Occupation aggregate participation probabilities across time: skill- and time-varying observables, skill and time- and occupation-specific wages, skill and occupation-specific intercepts; common wage coefficient and time coefficient; note reduction in m0, home-activity share over time

	skillgroup	time	m0	m1	m2	m3	m4
rk1t1	1	1	0.0887175	0.1995146	0.2020237	0.2757014	0.2340428
rk1t2	1	2	0.0646539	0.1984822	0.2223035	0.2803052	0.2342551
rk1t3	1	3	0.0358582	0.1975615	0.2339696	0.2909961	0.2416147
rk2t1	2	1	0.0942785	0.2435583	0.2481846	0.1610703	0.2529083
rk2t2	2	2	0.0780946	0.2411637	0.2669887	0.1673420	0.2464110
rk2t3	2	3	0.0572989	0.2348487	0.2807791	0.1643585	0.2627147

1.3.2 Create Regression Data Inputs

Second, generate relative market shares of various work occupations as prior.

```
mt_prob_rela_m2leisure3 <- mt_prob_o3[, 2:(it_M+1)]/mt_prob_o3[, 1]
# mt_log_prob_rela_m2leisure3 <- log(mt_prob_rela_m2leisure3/mt_prob_o3[, 1])
# kable(log(mt_prob_rela_m2leisure3/mt_prob_o3[, 1]), caption=st_caption) %>% kable_styling_fc()
```

Third, construct the estimation input matrices as before. There are $K \times M$ indicator vectors, a time variable, a HOMETECH variable, and 1 wage vector for data inputs. The indicator vectors are specific interecept for each type of worker (skilled or not) and for each occupation. Note:

1. k and m specific indicators
2. k specific hometech coefficients
3. common time coefficient
4. common coefficient for wage

```
# Regression input matrix
# it_K*it_M+it_K+1+1: 1. it_K*it_M for all indicators; 2. it_k for HOMETECH; 3. 1 for time; 4. 1 for wa
mt_prob_ols_input3 <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=it_K*it_M+it_K+1+1)
colnames(mt_prob_ols_input3) <- paste0('m', seq(1,dim(mt_prob_ols_input3)[2]))
rownames(mt_prob_ols_input3) <- paste0('rowkxMxT', seq(1,dim(mt_prob_ols_input3)[1]))

# LHS variable meshed store
ar_prob_ols_mesh_kmt <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=1)
# RHS variables meshed store
mt_indi_mesh_kmt <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=it_K*it_M)
ar_time_mesh_kmt <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=1)
ar_wage_mesh_kmt <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=1)
mt_hometech_mesh_kmt <- matrix(data=NA, nrow=it_K*it_T*it_M, ncol=it_K)

# Loop over columns
it_kxm_ctr <- 0
for (it_r_k in seq(1, it_K)) {
  for (it_r_m in seq(1, it_M)) {
    # Column counter
    it_kxm_ctr <- it_kxm_ctr + 1

    # Update name of indicator column
    colnames(mt_prob_ols_input3)[it_kxm_ctr] <- paste0('i_k', it_r_k, 'm', it_r_m)
    # Start and end row for the indictor function mt_indi_mesh_kmt
    it_indi_str_row <- (it_kxm_ctr-1)*it_T
    it_indi_end_row <- (it_kxm_ctr+0)*it_T
    # Update names of the hometech column
    colnames(mt_prob_ols_input3)[it_K*it_M + it_r_k] <- paste0('hometech_k', it_r_k)
    # Start and end row for the indictor function mt_hometech_mesh_kmt
    it_hometech_str_row <- (it_r_k-1)*it_M*it_T
    it_hometech_end_row <- (it_r_k+0)*it_M*it_T

    # Loop over rows
    it_rowKxT <- 0
    it_rowKxTxM <- 0
    for (it_k in seq(1, it_K)) {
      for (it_m in seq(1, it_M)) {
        for (it_t in seq(1, it_T)) {

          # KxT group counter
          it_rowKxT <- it_T*(it_k-1) + it_t

          # Row counter
          it_rowKxTxM <- it_rowKxTxM + 1

          # Indicator matrix, heterogeneous by K and M
          if ((it_rowKxTxM > it_indi_str_row) && (it_rowKxTxM <= it_indi_end_row)){
            mt_indi_mesh_kmt[it_rowKxTxM, it_kxm_ctr] <- 1
          } else {
            mt_indi_mesh_kmt[it_rowKxTxM, it_kxm_ctr] <- 0
          }

          # HOMETECH specific matrix, heterogeneous by K, homogeneous by M
          if (it_r_m == 1) {
            if ((it_rowKxTxM > it_hometech_str_row) && (it_rowKxTxM <= it_hometech_end_row)){
              mt_hometech_mesh_kmt[it_rowKxTxM, it_r_k] <- mt_hometech[it_t, it_k]
            } else {
              mt_hometech_mesh_kmt[it_rowKxTxM, it_r_k] <- 0
            }
          }

          # Only need to do once, homogeneous across K, M and T
          if (it_kxm_ctr == 1) {
            rownames(mt_prob_ols_input3)[it_rowKxTxM] <- paste0('ik', it_k, 'm', it_m, 't', it_t)
            # RHS
            ar_time_mesh_kmt[it_rowKxTxM] <- it_t
            ar_wage_mesh_kmt[it_rowKxTxM] <- mt_wages[it_rowKxT, it_m]
            # LHS, log of probability
            ar_prob_ols_mesh_kmt[it_rowKxTxM] <- log(mt_prob_rela_m2leisure3[it_rowKxT, it_m])
          }
        }
      }
    }
  }
}
# Indicators are the earlier columns
mt_prob_ols_input3[, 1:(it_K*it_M)] <- mt_indi_mesh_kmt
# Time variable column
it_col_start <- (it_K*it_M) + 1
```

```r
it_col_end <- it_col_start + it_K - 1
mt_prob_ols_input3[, it_col_start:it_col_end] <- mt_hometech_mesh_kmt
# Home Tech Column
mt_prob_ols_input3[, it_col_end+1] <- ar_time_mesh_kmt
# Wage is the last column
mt_prob_ols_input3[, it_col_end+2] <- ar_wage_mesh_kmt
# kable out
# kable(mt_prob_ols_input3) %>% kable_styling_fc()
```

Fourth, combine left-hand-side and right-hand-side regression input data structures/

```r
# Construct data structure
mt_all_inputs3 <- cbind(ar_prob_ols_mesh_kmt, mt_prob_ols_input3)
colnames(mt_all_inputs3)[1] <- 'log_pm_over_po'
colnames(mt_all_inputs3)[dim(mt_all_inputs3)[2]-1] <- 'time'
colnames(mt_all_inputs3)[dim(mt_all_inputs3)[2]] <- 'wages'
tb_all_inputs3 <- as_tibble(mt_all_inputs3)
# Show data
st_caption <- 'LHS=Log Probability Ratios (column 1); RHS=Indicator, time-trends and data with differen
kable(tb_all_inputs3, caption=st_caption) %>% kable_styling_fc_wide()
```

LHS=Log Probability Ratios (column 1); RHS=Indicator, time-trends and data with different assumptions on whether coefficients with be skill specific (hometech), occuption and skill specific (indictor), or homogeneous across skills and occupations (time and wage), OLS inputs (other columns)

log_pm_over_po	i_k1m1	i_k1m2	i_k1m3	i_k1m4	i_k2m1	i_k2m2	i_k2m3	i_k2m4	hometech_k1	hometech_k2	time	wages
0.8104303	1	0	0	0	0	0	0	0	0.2164079	0.0000000	1	1.677571
1.1216510	1	0	0	0	0	0	0	0	0.3181810	0.0000000	2	1.572633
1.7064781	1	0	0	0	0	0	0	0	0.7584595	0.0000000	3	1.102925
0.8229277	0	1	0	0	0	0	0	0	0.2164079	0.0000000	1	1.327921
1.2349948	0	1	0	0	0	0	0	0	0.3181810	0.0000000	2	1.954504
1.8756195	0	1	0	0	0	0	0	0	0.7584595	0.0000000	3	1.889539
1.1338609	0	0	1	0	0	0	0	0	0.2164079	0.0000000	1	1.655706
1.4668305	0	0	1	0	0	0	0	0	0.3181810	0.0000000	2	1.708530
2.0937381	0	0	1	0	0	0	0	0	0.7584595	0.0000000	3	1.544066
0.9700465	0	0	0	1	0	0	0	0	0.2164079	0.0000000	1	1.963024
1.2873623	0	0	0	1	0	0	0	0	0.3181810	0.0000000	2	1.902299
1.9077727	0	0	0	1	0	0	0	0	0.7584595	0.0000000	3	1.690705
0.9491036	0	0	0	0	1	0	0	0	0.0000000	0.1428000	1	1.899825
1.1275553	0	0	0	0	1	0	0	0	0.0000000	0.2316258	2	1.246088
1.4106597	0	0	0	0	1	0	0	0	0.0000000	0.4145463	3	1.042059
0.9679200	0	0	0	0	0	1	0	0	0.0000000	0.1428000	1	1.692803
1.2292855	0	0	0	0	0	1	0	0	0.0000000	0.2316258	2	1.640507
1.5892864	0	0	0	0	0	1	0	0	0.0000000	0.4145463	3	1.994270
0.5355882	0	0	0	0	0	0	1	0	0.0000000	0.1428000	1	1.594142
0.7621188	0	0	0	0	0	0	1	0	0.0000000	0.2316258	2	1.289160
1.0537680	0	0	0	0	0	0	1	0	0.0000000	0.4145463	3	1.147114
0.9867739	0	0	0	0	0	0	0	1	0.0000000	0.1428000	1	1.795467
1.1490801	0	0	0	0	0	0	0	1	0.0000000	0.2316258	2	1.024614
1.5227867	0	0	0	0	0	0	0	1	0.0000000	0.4145463	3	1.477796

1.3.3 Estimate Wage and Time Coefficients, Skill Specific Fixed Effects and HOMETECH Coefficient

Similar to before, estimate with OLS, and show that predictions based on OLS estimates match with the true parameters. Predicted probabilities are the same as observed probabilities.

```r
# Regression
fit_ols_agg_prob3 <- lm(log_pm_over_po ~ . -1, data = tb_all_inputs3)
summary(fit_ols_agg_prob3)
```

```
##
```

```
## Call:
## lm(formula = log_pm_over_po ~ . - 1, data = tb_all_inputs3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.978e-16 -1.212e-16  2.245e-17  6.724e-17  5.494e-16
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## i_k1m1      1.438e-01  4.270e-16 3.368e+14   <2e-16 ***
## i_k1m2      2.045e-01  4.796e-16 4.264e+14   <2e-16 ***
## i_k1m3      4.702e-01  4.624e-16 1.017e+15   <2e-16 ***
## i_k1m4      2.641e-01  5.049e-16 5.229e+14   <2e-16 ***
## i_k2m1      3.942e-01  4.277e-16 9.216e+14   <2e-16 ***
## i_k2m2      4.415e-01  5.028e-16 8.780e+14   <2e-16 ***
## i_k2m3      2.278e-02  4.176e-16 5.455e+13   <2e-16 ***
## i_k2m4      4.462e-01  4.348e-16 1.026e+15   <2e-16 ***
## hometech_k1 9.568e-01  7.700e-16 1.243e+15   <2e-16 ***
## hometech_k2 4.533e-01  1.662e-15 2.728e+14   <2e-16 ***
## time        2.283e-01  2.182e-16 1.046e+15   <2e-16 ***
## wages       1.379e-01  2.235e-16 6.168e+14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.412e-16 on 12 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.778e+31 on 12 and 12 DF,  p-value: < 2.2e-16
```

```
# Parse coefficients
ls_coefficients_esti <- vector(mode = "list", length = 0)
ls_coefficients_true <- vector(mode = "list", length = 0)
ar_coefficients3 <- fit_ols_agg_prob3$coefficients
it_col_end <- 0
for (it_coef_grp in c(1,2,3,4)) {
  it_col_str <- it_col_end + 1
  if (it_coef_grp == 1) {
    it_grp_coef_cnt <- (it_K*it_M)
    st_coef_name <- 'indi_km'
    ar_esti_true <- mt_alpha
    it_sign <- +1
  } else if (it_coef_grp == 2) {
    it_grp_coef_cnt <- it_K
    st_coef_name <- 'hometech_k'
    ar_esti_true <- ar_theta
    it_sign <- -1
  } else if (it_coef_grp == 3) {
    it_grp_coef_cnt <- 1
    st_coef_name <- 'time'
    ar_esti_true <- fl_phi
    it_sign <- -1
  } else if (it_coef_grp == 4) {
    it_grp_coef_cnt <- 1
    st_coef_name <- 'wage'
    ar_esti_true <- fl_beta
    it_sign <- +1
  }

  # select
  it_col_end <- it_col_end + it_grp_coef_cnt
  ar_esti_curgroup <- ar_coefficients3[it_col_str:it_col_end]
  if (it_coef_grp == 1) {
      ar_esti_curgroup <- t(matrix(data=ar_esti_curgroup, nrow=it_M, ncol=it_K))
  }
  # store
  ls_coefficients_esti[[st_coef_name]] <- it_sign*ar_esti_curgroup
  ls_coefficients_true[[st_coef_name]] <- ar_esti_true
}

# Compare estimates and true
print(ls_coefficients_esti)
```

```
## $indi_km
##           [,1]      [,2]       [,3]      [,4]
## [1,] 0.1437888 0.2044885 0.47023364 0.2640527
## [2,] 0.3941526 0.4415087 0.02277825 0.4462095
##
## $hometech_k
## hometech_k1 hometech_k2
##  -0.9568333  -0.4533342
##
## $time
##       time
## -0.2283074
##
## $wage
##     wages
## 0.1378588
```

```
print(ls_coefficients_true)
```

```
## $indi_km
##           m1        m2         m3        m4
## k1 0.1437888 0.2044885 0.47023364 0.2640527
## k2 0.3941526 0.4415087 0.02277825 0.4462095
##
## $hometech_k
## [1] -0.9568333 -0.4533342
##
## $time
## [1] -0.2283074
##
## $wage
## [1] 0.1378588
```

```
# Simulate given estimated parameters using earlier function
mt_prob_o3_full_esti <- ffi_logit_prob3_alpha_beta(it_K, it_T, it_M,
  ls_coefficients_esti[['indi_km']],
  ls_coefficients_esti[['wage']],
  ls_coefficients_esti[['time']],
  ls_coefficients_esti[['hometech_k']],
  mt_hometech, mt_wages)
# Results
st_caption <- 'Predicted probabilities based on estimates'
kable(mt_prob_o3_full_esti, caption=st_caption) %>% kable_styling_fc()
```

Predicted probabilities based on estimates

	skillgroup	time	m0	m1	m2	m3	m4
rk1t1	1	1	0.0887175	0.1995146	0.2020237	0.2757014	0.2340428
rk1t2	1	2	0.0646539	0.1984822	0.2223035	0.2803052	0.2342551
rk1t3	1	3	0.0358582	0.1975615	0.2339696	0.2909961	0.2416147
rk2t1	2	1	0.0942785	0.2435583	0.2481846	0.1610703	0.2529083
rk2t2	2	2	0.0780946	0.2411637	0.2669887	0.1673420	0.2464110
rk2t3	2	3	0.0572989	0.2348487	0.2807791	0.1643585	0.2627147

```
# Results
st_caption <- 'Compare differences in probability predictions based on estimates and true probabilities
kable(mt_prob_o3_full_esti-mt_prob_o3_full, caption=st_caption) %>% kable_styling_fc()
```

Compare differences in probability predictions based on estimates and true probabilities

	skillgroup	time	m0	m1	m2	m3	m4
rk1t1	0	0	0	0	0	0	0
rk1t2	0	0	0	0	0	0	0
rk1t3	0	0	0	0	0	0	0
rk2t1	0	0	0	0	0	0	0
rk2t2	0	0	0	0	0	0	0
rk2t3	0	0	0	0	0	0	0