IO III: Entry Models

0. Libraries and set-up

First, the libraries used:

```
# install.packages("haven")
library("haven") # to charge dta files
# install.packages("VGAM")
library("VGAM") # to get the bivariate normal
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
# install.packages("mlogit")
library(mlogit)
```

```
## Loading required package: dfidx
```

```
##
## Attaching package: 'dfidx'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
##
## Attaching package: 'mlogit'
```

```
## The following objects are masked from 'package:VGAM':
##
##     has.intercept, lrtest
```

Now we remove everything from the environment and charge the dataset.

```
rm(list = ls())
data = read_dta("jiadata.dta")
head(data)
```

```
## # A tibble: 6 x 19
##   county population   spc  urban midwest dbenton southern kmart walmart smalls~1
##    <dbl>      <dbl> <dbl>  <dbl>   <dbl>   <dbl>    <dbl> <dbl>   <dbl>    <dbl>
## 1      1       3.72  8.62 0.550        0    6.23        1     1       1        5
## 2      3       3.29  8.24 0.290        0    6.39        1     0       1        6
## 3      4       2.93  7.86 0.190        0    6.15        1     0       1        1
## 4      5       3.80  7.94 0.0900       0    6.14        1     0       1        6
## 5      6       2.43  7.71 0.350        0    6.34        1     0       0        3
## 6      7       3.08  8.33 0.25         0    6.29        1     0       1        6
## # ... with 9 more variables: dkmart <dbl>, dwmart <dbl>, x1 <dbl>, x2 <dbl>,
## #   x3 <dbl>, count0 <dbl>, count1 <dbl>, count2 <dbl>, nfirms <dbl>, and
## #   abbreviated variable name 1: smallstores
```

3. Implementation

3.1. Two Types and Sequential Entry (Berry Model)

Mostly, translating Stata code to R.

Create market configurations variable and a constant:

```
data["y"] = numeric(dim(data)[1]) + as.numeric((data$walmart==1)&(data$kmart==0)) + 2*as.numeric((data$
data["comp"] = numeric(dim(data)[1]) + 1
```

3.1.1. Walmart moves first, binormal

rho = 0.5, walmartbi_ext_pos

```
walmartbi_ext_pos = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1]*data$population + par[2]*data$spc + par[3]*data$urban
  zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  zk <- par[7] + par[8]*data$midwest
  com <- par[9]*data$comp

  pik <- xb + zk
  piw <- xb + zw
  piduow <- xb + zw + com
  piduok <- xb + zk + com
  athrho = 0.5

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_duo - prob_w - prob_nofirm
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$y == 0, log(prob_nofirm), ifelse(data$y == 1, log(prob_w), ifelse(data$y == 2, log
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -11.8, -1.1, 0.72, -19.6, 0.37, -0.20), walmartbi_ext_pos, data_in=data, con
```

```
## $par
## [1]   1.7094151   1.5157002   1.2959771 -12.3036877  -0.9743044   0.6418977
## [7] -19.1729452   0.3097677  -0.9120202
##
## $value
## [1] 1277.07
##
## $counts
```

```
## function gradient
##     1512       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

rho = -0.5, walmartbi_ext_neg

```
walmartbi_ext_neg = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1]*data$population + par[2]*data$spc + par[3]*data$urban
  zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  zk <- par[7] + par[8]*data$midwest
  com <- par[9]*data$comp

  pik <- xb + zk
  piw <- xb + zw
  piduow <- xb + zw + com
  piduok <- xb + zk + com
  athrho = -0.5

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_duo - prob_w - prob_nofirm
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$y == 0, log(prob_nofirm), ifelse(data$y == 1, log(prob_w), ifelse(data$y == 2, log
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -11.8, -1.1, 0.72, -19.6, 0.37, -0.20), walmartbi_ext_pos, data_in=data, con
```

```
## $par
## [1]   1.7094151   1.5157002   1.2959771 -12.3036877  -0.9743044   0.6418977
## [7] -19.1729452   0.3097677  -0.9120202
##
## $value
## [1] 1277.07
##
## $counts
## function gradient
##     1512       NA
##
## $convergence
```

```
## [1] 0
##
## $message
## NULL
```

Estimating Rho, walmartbi_ext_est

```
walmartbi_ext_est = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1]*data$population + par[2]*data$spc + par[3]*data$urban
  zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  zk <- par[7] + par[8]*data$midwest
  com <- par[9]*data$comp

  pik <- xb + zk
  piw <- xb + zw
  piduow <- xb + zw + com
  piduok <- xb + zk + com
  athrho = par[10]
  if (athrho>1){athrho=0.99}
  if (athrho<-1){athrho=-0.99}

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_duo - prob_w - prob_nofirm
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$y == 0, log(prob_nofirm), ifelse(data$y == 1, log(prob_w), ifelse(data$y == 2, log
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -11.8, -1.1, 0.72, -19.6, 0.37, -0.73, 0.32), walmartbi_ext_est, data_in=data
```

```
## $par
##  [1]   1.1021290   1.4939046   0.9310503 -11.3225283  -0.8086709   0.6218461
##  [7] -19.3647554   0.3289755   2.0933666  -0.0803939
##
## $value
## [1] 934.0785
##
## $counts
## function gradient
##     2599       NA
##
## $convergence
## [1] 0
##
```

```
## $message
## NULL
```

kmart moves first, estimate rho:

we need to change the probability areas:

```
kmart_est = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1]*data$population + par[2]*data$spc + par[3]*data$urban
  zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  zk <- par[7] + par[8]*data$midwest
  com <- par[9]*data$comp

  pik <- xb + zk
  piw <- xb + zw
  piduow <- xb + zw + com
  piduok <- xb + zk + com
  athrho = par[10]
  if (athrho>1){athrho=0.9}
  if (athrho<-1){athrho=-0.9}

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_k <- pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -pik, cov12 = athrho)
  prob_w <- 1 - prob_nofirm - prob_k - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$y == 0, log(prob_nofirm), ifelse(data$y == 1, log(prob_w), ifelse(data$y == 2, log
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -11.8, -1.1, 0.72, -19.6, 0.37, -0.7, 0.3), kmart_est, data_in=data, control
```

```
## $par
##  [1]   1.5042370   1.4931971   1.4510732 -11.5710503  -1.0013493   0.9442870
##  [7] -19.5143402   0.3046122   0.4920660   0.1554741
##
## $value
## [1] 1363.597
##
## $counts
## function gradient
##     1241       NA
##
## $convergence
## [1] 0
##
```

```
## $message
## NULL
```

estimate firm specific delta 1 y delta 2

```
delta_esp_est = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1]*data$population + par[2]*data$spc + par[3]*data$urban
  zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  zk <- par[7] + par[8]*data$midwest
  com1 <- par[10]*data$comp
  com2 <- par[11]*data$comp

  pik <- xb + zk
  piw <- xb + zw
  piduow <- xb + zw + com1
  piduok <- xb + zk + com2
  athrho = par[9]
  if (athrho>1){athrho=0.9}
  if (athrho<-1){athrho=-0.9}

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$y == 0, log(prob_nofirm), ifelse(data$y == 1, log(prob_w), ifelse(data$y == 2, log
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -11.8, -1.1, 0.72, -19.6, 0.37, -0.7, 0.3,-0.7,-0.7), delta_esp_est, data_in=
```

```
## $par
##  [1]   1.3197357   1.5873822   0.6516524 -11.5002063  -1.0169652   0.7868933
##  [7] -19.2435600   0.4552301   0.3917223   0.4594924   0.4234990  -0.2941245
##
## $value
## [1] 1358.839
##
## $counts
## function gradient
##     5827       NA
##
## $convergence
## [1] 0
##
```

```
## $message
## NULL
```

4 Nearly identical firms

rho=1

```
agg_noesp_rho1 = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1] + par[2]*data$population + par[3]*data$spc + par[4]*data$urban # this relates to entry (
  #zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  #zk <- par[7] + par[8]*data$midwest
  com <- par[5]*data$comp # this to y...

  pik <- xb #+ zk
  piw <- xb #+ zw
  piduow <- xb + com #+ zw + com
  piduok <- xb + com #+ zk
  athrho = 0.99999 #par[6]

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$nfirms == 0, log(prob_nofirm), ifelse(data$nfirms == 1, log(prob_w) + log(prob_k),
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1,1.5, 1.5, 1.2, -.20), agg_noesp_rho1, data_in=data, control = list("maxit"=10000))
```

```
## $par
## [1] -153.08290   93.36311  -32.88775   41.09768  184.36868
##
## $value
## [1] 2.985434e-11
##
## $counts
## function gradient
##      258       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

rho = 0.5

```
agg_noesp_rho05 = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1] + par[2]*data$population + par[3]*data$spc + par[4]*data$urban # this relates to entry (
  #zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  #zk <- par[7] + par[8]*data$midwest
  com <- par[5]*data$comp # this to y...

  pik <- xb #+ zk
  piw <- xb #+ zw
  piduow <- xb + com #+ zw + com
  piduok <- xb + com #+ zk
  athrho = 0.5 #par[6]

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$nfirms == 0, log(prob_nofirm), ifelse(data$nfirms == 1, log(prob_w) + log(prob_k),
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(0, 1.5, 1.5, 1.2, -.20), agg_noesp_rho05, data_in=data, control = list("maxit"=10000))
```

```
## $par
## [1]  0.0268959  1.5147711  1.5317626  1.2300520 -0.1642291
##
## $value
## [1] 16888.21
##
## $counts
## function gradient
##      236       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

estimating rho nearly identical case

```
agg_noesp_est_rho = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1] + par[2]*data$population + par[3]*data$spc + par[4]*data$urban # this relates to entry (
  #zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  #zk <- par[7] + par[8]*data$midwest
```

```
  com <- par[5]*data$comp # this to y...

  pik <- xb #+ zk
  piw <- xb #+ zw
  piduow <- xb + com #+ zw + com
  piduok <- xb + com #+ zk
  athrho = par[6]

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$nfirms == 0, log(prob_nofirm), ifelse(data$nfirms == 1, log(prob_w) + log(prob_k),
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(0, 1.5, 1.5, 1.2, -.20, 0.5), agg_noesp_est_rho, data_in=data, control = list("maxit"=10000)
```

```
## $par
## [1]  0.04837254  1.54980325  1.55095672  1.25091735 -0.14940389  0.54996313
##
## $value
## [1] 16200.67
##
## $counts
## function gradient
##      571       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

including Z

```
agg_estZ = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1] + par[2]*data$population + par[3]*data$spc + par[4]*data$urban # this relates to entry (
  zw <- par[5] +par[6]*data$dbenton + par[7]*data$southern
  zk <- par[8] + par[9]*data$midwest
  com <- par[10]*data$comp # this to y...

  pik <- xb #+ zk
  piw <- xb #+ zw
```

```
  piduow <- xb + com #+ zw + com
  piduok <- xb + com #+ zk
  athrho = par[11]
  if (athrho>1){athrho=0.9}
  if (athrho<-1){athrho=-0.9}

  prob_duo <- 1 - (pbinorm(-piduow, 100, cov12 = athrho) - pbinorm(-piduow, -piduok, cov12 = athrho)) -

  prob_nofirm <- pbinorm(-piw, -pik, cov12 = athrho)
  prob_w <- pbinorm(100, -piduok, cov12 = athrho) - pbinorm(-piw, -piduok, cov12 = athrho)
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$nfirms == 0, log(prob_nofirm), ifelse(data$nfirms == 1, log(prob_w) + log(prob_k),
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.5, 1.5, 1.2, -1.1, .72, -11.8, .37, -19.6, -0.2,-0.2, 0.5), agg_estZ, data_in=data, contr
```

```
## $par
##  [1]   3.39734007   0.83701954   1.61816834  -1.54050695   1.23842201
##  [6] -10.98372829   0.04366567 -19.75393426   0.50274592  -1.10809891
## [11]   0.29191474
##
## $value
## [1] 1396.792
##
## $counts
## function gradient
##      539       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Univariate Case

```
agg_noesp_univariate = function(par,data_in=data){
  # par -> c(a1,a2,a3,b0,b1,b2,c0,c1,d1)
  xb <- par[1] + par[2]*data$population + par[3]*data$spc + par[4]*data$urban # this relates to entry (
  #zw <- par[4] +par[5]*data$dbenton + par[6]*data$southern
  #zk <- par[7] + par[8]*data$midwest
  com <- par[5]*data$comp # this to y...

  pik <- xb #+ zk
  piw <- xb #+ zw
  piduow <- xb + com #+ zw + com
```

```
  piduok <- xb + com #+ zk

  prob_duo <- 1 - (dnorm(-piduow) - dnorm(max(-piduow, -piduok))) - (dnorm(-piduok) - dnorm(max(-piduow
  prob_nofirm <- dnorm(max(-piw, -pik))
  prob_w <- dnorm(-piduok) - dnorm(max(-piw, -piduok))
  prob_k <- 1 - prob_nofirm - prob_w - prob_duo
  prob_duo[prob_duo<0] <- 0
  prob_nofirm[prob_nofirm<0] <- 0
  prob_w[prob_w<0] <- 0
  prob_k[prob_k<0] <- 0

  lnf <- ifelse(data$nfirms == 0, log(prob_nofirm), ifelse(data$nfirms == 1, log(prob_w) + log(prob_k),
  lnf[is.infinite(lnf)] <- 0
  return(-sum(lnf))
}
```

```
optim(par=c(1.6417959,1.5124064, -0.9862969, 1.689094, 0.9191), agg_noesp_univariate, data_in=data, con
```

```
## $par
## [1]  1.8696706921  1.8355454052 -0.5594583222  2.0480723325 -0.0001081403
##
## $value
## [1] 802.0832
##
## $counts
## function gradient
##      164       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```