data = read.csv(file = "M:/A Master of Science in Marketing Sciences/Mathematical Models in Marketing (Kohli)/latent/data\_ipad.csv",head = TRUE)

attach(data)

summary(data)

## Modelling Step 1: Set up Training Set and Dev Set

random\_factor = sample(1:15,137,replace = TRUE)

determinant\_factor = (1:137)\*15 - 15

## The index for Dev Set is the summation of Random and Deterministic.

index = random\_factor + determinant\_factor

dev = data[which(choice\_set\_id %in% index) ,]

train = data[which(!(choice\_set\_id %in% index)) ,]

## Model 1: Fit an aggregate model without any segment

## In the training set,

X1\_train = subset(train, alternative\_id\_in\_set == 1)

X2\_train = subset(train, alternative\_id\_in\_set == 2)

X3\_train = subset(train, alternative\_id\_in\_set == 3)

## In the test set,``

X1\_test = subset(dev, alternative\_id\_in\_set == 1)

X2\_test = subset(dev, alternative\_id\_in\_set == 2)

X3\_test = subset(dev, alternative\_id\_in\_set == 3)

## The dependent variable

train\_choice = X1\_train$choice

test\_choice = X1\_test$choice

train\_set = cbind(X1\_train[,5:22],X2\_train[,5:22],X3\_train[,5:22])

test\_set = cbind(X1\_test[,5:22],X2\_test[,5:22],X3\_test[,5:22])

## The parameters for the first segment

par1 = rnorm(18)

## The parameters for the second segment

par2 = rnorm(18)

## The parameters for the third segment

par3 = rnorm(18)

## The parameters for the fourth segment

par4 = rnorm(18)

## The parameter for the likelihood that a randomly chosen individual belongs to segment 1

p4 = exp(rnorm(4))

p4 = p4/sum(p4)

## The combined parameters are

par\_4 = t(cbind(t(par1),t(par2),t(par3),t(par4),t(p4[1:3])))

N = 1918

ll\_4 <- function(beta)

{

res = 0

## To calculate the likelohood for the first segment

M11 = as.matrix(train\_set)[,1:18] %\*% beta[1:18]

M12 = as.matrix(train\_set)[,19:36] %\*% beta[1:18]

M13 = as.matrix(train\_set)[,37:54] %\*% beta[1:18]

M1 = cbind(exp(M11),exp(M12),exp(M13))

## The likelihood for each of the three alternatives in the first segment

M1 = M1 / rowSums(M1)

M21 = as.matrix(train\_set)[,1:18] %\*% beta[19:36]

M22 = as.matrix(train\_set)[,19:36] %\*% beta[19:36]

M23 = as.matrix(train\_set)[,37:54] %\*% beta[19:36]

M2 = cbind(exp(M21),exp(M22),exp(M23))

## The likelihood for each of the three alternatives in the second segment

M2 = M2 / rowSums(M2)

M31 = as.matrix(train\_set)[,1:18] %\*% beta[37:54]

M32 = as.matrix(train\_set)[,19:36] %\*% beta[37:54]

M33 = as.matrix(train\_set)[,37:54] %\*% beta[37:54]

M3 = cbind(exp(M31),exp(M32),exp(M33))

## The likelihood for each of the three alternatives in the third segment

M3 = M3 / rowSums(M3)

M41 = as.matrix(train\_set)[,1:18] %\*% beta[55:72]

M42 = as.matrix(train\_set)[,19:36] %\*% beta[55:72]

M43 = as.matrix(train\_set)[,37:54] %\*% beta[55:72]

M4 = cbind(exp(M41),exp(M42),exp(M43))

## The likelihood for each of the three alternatives in the fourth segment

M4 = M4 / rowSums(M4)

## Given that the consumer belongs to Segment 1, what is the probability for him to choose Alternative i?

prob\_1 = M1[cbind(seq(1,length(train\_choice)),train\_choice)]

## Given that the consumer belongs to Segment 2, what is the probability for him to choose Alternative i?

prob\_2 = M2[cbind(seq(1,length(train\_choice)),train\_choice)]

## Given that the consumer belongs to Segment 3, what is the probability for him to choose Alternative i?

prob\_3 = M3[cbind(seq(1,length(train\_choice)),train\_choice)]

## Given that the consumer belongs to Segment 3, what is the probability for him to choose Alternative i?

prob\_4 = M4[cbind(seq(1,length(train\_choice)),train\_choice)]

## This transformation guarantees that the likelihood for each segment is within the range of (0,1)

probability\_1 = beta[73]

probability\_2 = beta[74]

probability\_3 = beta[75]

for (i in 1:137)

{

MP = c()

MP[i] = probability\_1 \* cumprod(prob\_1[(14\*(i-1)+1):(14\*i)])[14] + probability\_2 \* cumprod(prob\_2[(14\*(i-1)+1):(14\*i)])[14] + probability\_3 \* cumprod(prob\_3[(14\*(i-1)+1):(14\*i)])[14] + (1 - probability\_1 - probability\_2 - probability\_3) \* cumprod(prob\_4[(14\*(i-1)+1):(14\*i)])[14]

res = res -log(MP[i])

}

return (res)

}

ML\_4 = nlm(ll\_4,par\_4,hessian = TRUE)

mode = ML\_4$estimate

SE = sqrt(diag(solve(ML\_4$hessian)))

Tvalue = mode/SE

ll = 2\*ML\_4$minimum

Result\_4 = cbind(Estimate = mode, SE= SE, Tvalue = Tvalue, minusll = ll)

round(Result\_4,2)

## This is the estimates for pi\_i(the market share of the first segment)

pi\_1 = ML\_4$estimate[73]

pi\_2 = ML\_4$estimate[74]

pi\_3 = ML\_4$estimate[75]

## Concise Expression

## The likelihood for each of the three alternatives in the first segment

M11 = as.matrix(train\_set)[,1:18] %\*% ML\_4$estimate[1:18]

M12 = as.matrix(train\_set)[,19:36] %\*% ML\_4$estimate[1:18]

M13 = as.matrix(train\_set)[,37:54] %\*% ML\_4$estimate[1:18]

M1 = cbind(exp(M11),exp(M12),exp(M13))

M1 = M1 / rowSums(M1)

## M11 is the probability for individual i's actual choice in Segment 1

M11 = M1[cbind(1:(137\*14),train\_choice[1:(137\*14)])]

## The likelihood for each of the three alternatives in the second segment

M21 = as.matrix(train\_set)[,1:18] %\*% ML\_4$estimate[19:36]

M22 = as.matrix(train\_set)[,19:36] %\*% ML\_4$estimate[19:36]

M23 = as.matrix(train\_set)[,37:54] %\*% ML\_4$estimate[19:36]

M2 = cbind(exp(M21),exp(M22),exp(M23))

M2 = M2 / rowSums(M2)

## M11 is the probability for individual i's actual choice in Segment 2

M12 = M2[cbind(1:(14\*137),train\_choice[1:(14\*137)])]

## The likelihood for each of the three alternatives in the third segment

M31 = as.matrix(train\_set)[,1:18] %\*% ML\_4$estimate[37:54]

M32 = as.matrix(train\_set)[,19:36] %\*% ML\_4$estimate[37:54]

M33 = as.matrix(train\_set)[,37:54] %\*% ML\_4$estimate[37:54]

M3 = cbind(exp(M31),exp(M32),exp(M33))

M3 = M3 / rowSums(M3)

## M11 is the probability for individual i's actual choice in Segment 2

M13 = M3[cbind(1:(14\*137),train\_choice[1:(14\*137)])]

## The likelihood for each of the three alternatives in the fourth segment

M41 = as.matrix(train\_set)[,1:18] %\*% ML\_4$estimate[55:72]

M42 = as.matrix(train\_set)[,19:36] %\*% ML\_4$estimate[55:72]

M43 = as.matrix(train\_set)[,37:54] %\*% ML\_4$estimate[55:72]

M4 = cbind(exp(M41),exp(M42),exp(M43))

M4 = M4 / rowSums(M4)

## M11 is the probability for individual i's actual choice in Segment 2

M14 = M4[cbind(1:(14\*137),train\_choice[1:(14\*137)])]

## The individual-level estimates of segment membership using Bayes Rule

## What is the orobability for each individual to belong to Segment 1?

prob <- rep(0,137\*3)

prob = matrix(prob, 137,3)

for (i in 1:137)

{

prob[i,1] = (pi\_1 \* (cumprod(M11[(1+(i-1)\*14):(14 \* i)])[14]))/((pi\_1 \* cumprod(M11[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_2 \* cumprod(M12[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_3 \* cumprod(M13[(1+(i-1)\*14):(14 \* i)])[14]) + ((1 - pi\_2 - pi\_1 - pi\_3) \* cumprod(M14[(1+(i-1)\*14):(14 \* i)])[14]))

prob[i,2] = (pi\_2 \* (cumprod(M12[(1+(i-1)\*14):(14 \* i)])[14]))/((pi\_1 \* cumprod(M11[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_2 \* cumprod(M12[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_3 \* cumprod(M13[(1+(i-1)\*14):(14 \* i)])[14]) + ((1 - pi\_2 - pi\_1 - pi\_3) \* cumprod(M14[(1+(i-1)\*14):(14 \* i)])[14]))

prob[i,3] = (pi\_3 \* (cumprod(M13[(1+(i-1)\*14):(14 \* i)])[14]))/((pi\_1 \* cumprod(M11[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_2 \* cumprod(M12[(1+(i-1)\*14):(14 \* i)])[14]) + (pi\_3 \* cumprod(M13[(1+(i-1)\*14):(14 \* i)])[14]) + ((1 - pi\_2 - pi\_1 - pi\_3) \* cumprod(M14[(1+(i-1)\*14):(14 \* i)])[14]))

}

## The vector prob is the probability for each individual to choose Segment 1

##########################################

# ## Another Method which is theoretical correct but practically infeasible

# cumprod(M11[(1+(2-1)\*14):(14+(2-1)\*14)])[14]

# cum\_M11 = cumprod(M11)

# cum\_M12 = cumprod(M12)

# cumprod(M11)[14]/(cumprod(M11)[14]+cumprod(M12)[14])

#

# a <- 1:(137\*14)

# b\_11 <- cum\_M11[seq(14, 137\*14, 14)]

# b\_11\_tmp <- t(cbind(1,t(b\_11)))

# b\_11\_tem = b\_11\_tmp[1:137]

# b\_11/b\_11\_tem

##########################################

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Then I am going to do the Cross-Validation

## The expected probability for each alternative in Segment 1

## Segment 1 -- Choice 1

X1\_predict\_11 = as.matrix(X1\_test)[,5:22] %\*% ML\_4$estimate[1:18]

## Segment 1 -- Choice 2

X2\_predict\_12 = as.matrix(X2\_test)[,5:22] %\*% ML\_4$estimate[1:18]

## Segment 1 -- Choice 3

X3\_predict\_13 = as.matrix(X3\_test)[,5:22] %\*% ML\_4$estimate[1:18]

## The expected probability for each alternative in Segment 2

## Segment 2 -- Choice 1

X1\_predict\_21 = as.matrix(X1\_test)[,5:22] %\*% ML\_4$estimate[19:36]

## Segment 2 -- Choice 2

X2\_predict\_22 = as.matrix(X2\_test)[,5:22] %\*% ML\_4$estimate[19:36]

## Segment 2 -- Choice 3

X3\_predict\_23 = as.matrix(X3\_test)[,5:22] %\*% ML\_4$estimate[19:36]

## The expected probability for each alternative in Segment 3

## Segment 3 -- Choice 1

X1\_predict\_31 = as.matrix(X1\_test)[,5:22] %\*% ML\_4$estimate[37:54]

## Segment 3 -- Choice 2

X2\_predict\_32 = as.matrix(X2\_test)[,5:22] %\*% ML\_4$estimate[37:54]

## Segment 3 -- Choice 3

X3\_predict\_33 = as.matrix(X3\_test)[,5:22] %\*% ML\_4$estimate[37:54]

## The expected probability for each alternative in Segment 4

## Segment 4 -- Choice 1

X1\_predict\_41 = as.matrix(X1\_test)[,5:22] %\*% ML\_4$estimate[55:72]

## Segment 4 -- Choice 2

X2\_predict\_42 = as.matrix(X2\_test)[,5:22] %\*% ML\_4$estimate[55:72]

## Segment 4 -- Choice 3

X3\_predict\_43 = as.matrix(X3\_test)[,5:22] %\*% ML\_4$estimate[55:72]

predict\_41 = cbind(exp(X1\_predict\_11),exp(X2\_predict\_12),exp(X3\_predict\_13))

predict\_42 = cbind(exp(X1\_predict\_21),exp(X2\_predict\_22),exp(X3\_predict\_23))

predict\_43 = cbind(exp(X1\_predict\_31),exp(X2\_predict\_32),exp(X3\_predict\_33))

predict\_44 = cbind(exp(X1\_predict\_41),exp(X2\_predict\_42),exp(X3\_predict\_43))

## predict\_41 is the predicted probability for each individual to choose each alternative given that he belongs to Segment 1

predict\_41 = predict\_41/rowSums(predict\_41)

## predict\_42 is the predicted probability for each individual to choose each alternative given that he belongs to Segment 2

predict\_42 = predict\_42/rowSums(predict\_42)

predict\_43 = predict\_43/rowSums(predict\_43)

predict\_44 = predict\_44/rowSums(predict\_44)

## The vector prob is the probability for each individual to choose Segment 1

## We can use the posterior segment membership probability to estimate the probability that individual i chooses alternative j

predict\_4 = prob[,1] \* predict\_41 + prob[,2] \* predict\_42 + prob[,3] \* predict\_43 + (1 - prob[,1] - prob[,2] - prob[,3]) \* predict\_44

prediction\_4 = max.col(predict\_4)

## The performance of the model without segments is evaluated by the proportion of correct predictions.

performance\_4 = sum(prediction\_4 == test\_choice)/length(test\_choice)

performance\_4

Result\_4

> performance\_4

[1] 0.6058394

> Result\_4

Estimate SE Tvalue minusll

[1,] -0.90802887 0.33333513 -2.72407196 3110.11

[2,] -0.58179173 0.28112494 -2.06951307 3110.11

[3,] 0.12552095 0.29046030 0.43214494 3110.11

[4,] 0.17273327 0.26961978 0.64065503 3110.11

[5,] 0.47279494 0.24480803 1.93128855 3110.11

[6,] 0.31310241 0.25099407 1.24744943 3110.11

[7,] 0.34375867 0.26987036 1.27379190 3110.11

[8,] 0.59843206 0.26539303 2.25488993 3110.11

[9,] 0.96119064 0.26389818 3.64227834 3110.11

[10,] 0.65977216 0.29517229 2.23521036 3110.11

[11,] 1.30528624 0.26427174 4.93918217 3110.11

[12,] 1.77393380 0.28825452 6.15405366 3110.11

[13,] -0.28209838 0.20855811 -1.35261285 3110.11

[14,] 0.05088475 0.19419996 0.26202245 3110.11

[15,] -0.04699443 0.28268952 -0.16624046 3110.11

[16,] -0.02719406 0.30917398 -0.08795714 3110.11

[17,] -0.71958465 0.36917248 -1.94918282 3110.11

[18,] -1.56441378 0.37365052 -4.18683683 3110.11

[19,] 0.93890108 0.24321612 3.86035711 3110.11

[20,] 0.63492037 0.21884563 2.90122483 3110.11

[21,] 0.58741764 0.20837377 2.81905745 3110.11

[22,] 0.20085680 0.20409043 0.98415591 3110.11

[23,] 0.34784602 0.18441278 1.88623592 3110.11

[24,] 0.69900704 0.18251653 3.82982873 3110.11

[25,] 0.73053388 0.19518365 3.74280271 3110.11

[26,] 0.76186261 0.18322297 4.15811742 3110.11

[27,] 0.80190480 0.18374705 4.36417778 3110.11

[28,] 1.16879950 0.23287457 5.01900868 3110.11

[29,] 0.53333864 0.16157894 3.30079306 3110.11

[30,] 0.91055907 0.15995789 5.69249250 3110.11

[31,] 0.24012336 0.14434503 1.66353738 3110.11

[32,] 0.16105015 0.13665749 1.17849487 3110.11

[33,] -0.64860071 0.16032234 -4.04560413 3110.11

[34,] -2.07975254 0.22743728 -9.14429041 3110.11

[35,] -3.81062839 0.33547464 -11.35891649 3110.11

[36,] -5.68112451 0.57002323 -9.96647900 3110.11

[37,] 2.71852161 16.94864812 0.16039755 3110.11

[38,] -0.86660267 25.72392796 -0.03368858 3110.11

[39,] 1.96765521 19.33409587 0.10177126 3110.11

[40,] 6.02336926 12.64717294 0.47626211 3110.11

[41,] -7.05351769 19.04662491 -0.37032901 3110.11

[42,] 6.00745634 21.51030098 0.27928277 3110.11

[43,] 7.63069887 15.05008729 0.50702024 3110.11

[44,] -0.53349517 10.67646798 -0.04996926 3110.11

[45,] -2.55419152 20.82121272 -0.12267256 3110.11

[46,] 2.28311901 13.68033469 0.16689058 3110.11

[47,] 4.06721763 11.03111695 0.36870406 3110.11

[48,] 1.61041184 13.96639489 0.11530619 3110.11

[49,] 0.96407225 12.17541110 0.07918191 3110.11

[50,] 3.78966531 12.18202655 0.31108661 3110.11

[51,] 1.05643062 18.98958336 0.05563211 3110.11

[52,] -3.45715531 23.14609968 -0.14936233 3110.11

[53,] -0.45795500 24.12905345 -0.01897940 3110.11

[54,] -0.80187470 29.21902835 -0.02744358 3110.11

[55,] 0.36144351 0.19164231 1.88603191 3110.11

[56,] 2.05855371 0.19182949 10.73116378 3110.11

[57,] 0.57038736 0.17294420 3.29810052 3110.11

[58,] 0.14921973 0.18203987 0.81970906 3110.11

[59,] 0.16394512 0.14748873 1.11157734 3110.11

[60,] 0.46497116 0.14466237 3.21418189 3110.11

[61,] 0.33465927 0.15141837 2.21016297 3110.11

[62,] 0.02464678 0.15174485 0.16242252 3110.11

[63,] 0.67936363 0.14653628 4.63614635 3110.11

[64,] 0.71748689 0.15316276 4.68447350 3110.11

[65,] 0.10188675 0.11402943 0.89351277 3110.11

[66,] 0.34970825 0.11271892 3.10248052 3110.11

[67,] 0.18055917 0.11850794 1.52360393 3110.11

[68,] 0.14251957 0.11389652 1.25130745 3110.11

[69,] -0.13564654 0.17390820 -0.77998932 3110.11

[70,] 0.01242294 0.17322787 0.07171443 3110.11

[71,] 0.01470669 0.18491841 0.07953071 3110.11

[72,] -0.33175646 0.18592449 -1.78436125 3110.11

[73,] 0.16010576 0.03751567 4.26770350 3110.11

[74,] 0.43477562 0.04918881 8.83891230 3110.11

[75,] 0.01420655 0.01004568 1.41419488 3110.11

> performance\_4

[1] 0.5839416

> Result\_4

Estimate SE Tvalue minusll

[1,] -0.54471465 0.29776468 -1.8293461 3010.283

[2,] -0.13788050 0.25841512 -0.5335620 3010.283

[3,] 0.42731496 0.25064769 1.7048430 3010.283

[4,] 0.17386938 0.25974755 0.6693783 3010.283

[5,] 0.03881432 0.23616391 0.1643533 3010.283

[6,] 0.37665694 0.25416357 1.4819470 3010.283

[7,] 0.47923838 0.28297212 1.6935887 3010.283

[8,] 0.58307760 0.20898901 2.7899917 3010.283

[9,] 0.77584535 0.24144444 3.2133493 3010.283

[10,] 0.35675487 0.25195483 1.4159477 3010.283

[11,] 1.50987188 0.26703237 5.6542653 3010.283

[12,] 2.22287343 0.30346268 7.3250307 3010.283

[13,] -0.21705734 0.27115727 -0.8004850 3010.283

[14,] 0.14664369 0.17154370 0.8548474 3010.283

[15,] -0.39681109 0.26669523 -1.4878822 3010.283

[16,] -0.71153271 0.24489839 -2.9054201 3010.283

[17,] -1.61239011 0.29438182 -5.4772069 3010.283

[18,] -2.54160710 0.38370606 -6.6238388 3010.283

[19,] 0.82314420 0.31986185 2.5734367 3010.283

[20,] 0.49419198 0.30363573 1.6275818 3010.283

[21,] 0.68008547 0.27486834 2.4742226 3010.283

[22,] 0.08140519 0.30433329 0.2674870 3010.283

[23,] 0.38907818 0.25879874 1.5034006 3010.283

[24,] 0.42033568 0.26454640 1.5888921 3010.283

[25,] 0.76583202 0.28174119 2.7182111 3010.283

[26,] 1.02496816 0.26651359 3.8458383 3010.283

[27,] 1.04369592 0.25382759 4.1118301 3010.283

[28,] 1.01698288 0.29811706 3.4113541 3010.283

[29,] 0.43055886 0.22652000 1.9007543 3010.283

[30,] 0.61625473 0.22189749 2.7772046 3010.283

[31,] 0.29312520 0.20022836 1.4639544 3010.283

[32,] 0.20457128 0.18833838 1.0861901 3010.283

[33,] -0.90137598 0.19947486 -4.5187448 3010.283

[34,] -3.10111817 0.34487512 -8.9920033 3010.283

[35,] -5.66211453 0.58101620 -9.7451921 3010.283

[36,] -8.55145261 0.90545433 -9.4443776 3010.283

[37,] 0.23757118 0.25597599 0.9280995 3010.283

[38,] 2.78281677 0.24755117 11.2413798 3010.283

[39,] 0.99249071 0.21133588 4.6962717 3010.283

[40,] 0.15528625 0.22701742 0.6840279 3010.283

[41,] -0.05204298 0.18757326 -0.2774541 3010.283

[42,] 0.40557594 0.17823631 2.2754956 3010.283

[43,] 0.42787055 0.19088349 2.2415273 3010.283

[44,] -0.35437182 0.18822560 -1.8826973 3010.283

[45,] 0.57219472 0.18147764 3.1529764 3010.283

[46,] 0.28066876 0.19836594 1.4149040 3010.283

[47,] 0.14563701 0.14902391 0.9772728 3010.283

[48,] 0.59802796 0.14559623 4.1074412 3010.283

[49,] 0.03941224 0.15026775 0.2622801 3010.283

[50,] 0.21113272 0.14987185 1.4087550 3010.283

[51,] -0.14071562 0.24402531 -0.5766435 3010.283

[52,] -0.13882963 0.22805644 -0.6087512 3010.283

[53,] 0.14330873 0.25038072 0.5723633 3010.283

[54,] -0.14335134 0.26082634 -0.5496045 3010.283

[55,] 1.19095484 0.29980192 3.9724724 3010.283

[56,] 0.90456579 0.32999814 2.7411239 3010.283

[57,] -0.04377282 0.27045525 -0.1618486 3010.283

[58,] 0.28333190 0.31196793 0.9082084 3010.283

[59,] 0.10788316 0.24294958 0.4440558 3010.283

[60,] 0.51942774 0.24686165 2.1041248 3010.283

[61,] 0.33618307 0.30806417 1.0912761 3010.283

[62,] 0.64969511 0.24770396 2.6228693 3010.283

[63,] 1.00223002 0.25583097 3.9175477 3010.283

[64,] 1.85386372 0.30600796 6.0582205 3010.283

[65,] -0.02762419 0.16641194 -0.1659989 3010.283

[66,] 0.09803065 0.22183218 0.4419136 3010.283

[67,] 0.60097367 0.18077530 3.3244236 3010.283

[68,] 0.18205069 0.20817804 0.8744952 3010.283

[69,] 0.37330588 0.24756631 1.5079026 3010.283

[70,] 0.24854183 0.28272355 0.8790984 3010.283

[71,] -0.48752783 0.25806885 -1.8891386 3010.283

[72,] -1.15696166 0.28244150 -4.0962878 3010.283

[73,] 0.20833086 0.04468928 4.6617634 3010.283

[74,] 0.29977071 0.04127790 7.2622577 3010.283

[75,] 0.28066679 0.04219556 6.6515721 3010.283

> performance\_4

[1] 0.5474453

> Result\_4

Estimate SE Tvalue minusll

[1,] 8.79730725 0.29776468 2.954450e+01 1941.754

[2,] 35.00549025 0.25841512 1.354622e+02 1941.754

[3,] -27.51306185 0.25064769 -1.097679e+02 1941.754

[4,] 33.41782634 0.25974755 1.286550e+02 1941.754

[5,] -68.42673149 0.23616391 -2.897425e+02 1941.754

[6,] 78.27359901 0.25416357 3.079655e+02 1941.754

[7,] -18.70435485 0.28297212 -6.609964e+01 1941.754

[8,] 10.70245639 0.20898901 5.121062e+01 1941.754

[9,] 35.19109140 0.24144444 1.457523e+02 1941.754

[10,] -22.48671094 0.25195483 -8.924898e+01 1941.754

[11,] -17.19096584 0.26703237 -6.437783e+01 1941.754

[12,] 10.57444955 0.30346268 3.484596e+01 1941.754

[13,] 15.56904130 0.27115727 5.741702e+01 1941.754

[14,] -16.27042910 0.17154370 -9.484714e+01 1941.754

[15,] 11.77312136 0.26669523 4.414448e+01 1941.754

[16,] 30.41344211 0.24489839 1.241880e+02 1941.754

[17,] -36.97749496 0.29438182 -1.256107e+02 1941.754

[18,] -25.59329646 0.38370606 -6.670027e+01 1941.754

[19,] 47.78793002 0.31986185 1.494018e+02 1941.754

[20,] 84.66655545 0.30363573 2.788425e+02 1941.754

[21,] -80.92177892 0.27486834 -2.944020e+02 1941.754

[22,] 50.00166893 0.30433329 1.642990e+02 1941.754

[23,] 53.27034698 0.25879874 2.058370e+02 1941.754

[24,] 69.50565427 0.26454640 2.627352e+02 1941.754

[25,] -147.44392878 0.28174119 -5.233311e+02 1941.754

[26,] 244.76050885 0.26651359 9.183791e+02 1941.754

[27,] -207.04727394 0.25382759 -8.157004e+02 1941.754

[28,] -39.78565272 0.29811706 -1.334565e+02 1941.754

[29,] 14.19385992 0.22652000 6.266052e+01 1941.754

[30,] 45.46144190 0.22189749 2.048759e+02 1941.754

[31,] -29.55616763 0.20022836 -1.476123e+02 1941.754

[32,] 150.93760973 0.18833838 8.014172e+02 1941.754

[33,] 9.16334211 0.19947486 4.593733e+01 1941.754

[34,] 3.01888940 0.34487512 8.753573e+00 1941.754

[35,] 114.44923225 0.58101620 1.969811e+02 1941.754

[36,] -91.78746437 0.90545433 -1.013717e+02 1941.754

[37,] -10.33992390 0.25597599 -4.039412e+01 1941.754

[38,] 39.50610491 0.24755117 1.595876e+02 1941.754

[39,] -52.97184268 0.21133588 -2.506524e+02 1941.754

[40,] -7.31416798 0.22701742 -3.221853e+01 1941.754

[41,] -193.75680797 0.18757326 -1.032966e+03 1941.754

[42,] 90.72491041 0.17823631 5.090147e+02 1941.754

[43,] 119.46145731 0.19088349 6.258344e+02 1941.754

[44,] 76.56089766 0.18822560 4.067507e+02 1941.754

[45,] 27.48748670 0.18147764 1.514649e+02 1941.754

[46,] -55.85718766 0.19836594 -2.815866e+02 1941.754

[47,] 135.95324614 0.14902391 9.122915e+02 1941.754

[48,] -13.77414408 0.14559623 -9.460509e+01 1941.754

[49,] -6.30691341 0.15026775 -4.197117e+01 1941.754

[50,] 42.62065845 0.14987185 2.843807e+02 1941.754

[51,] -53.03894836 0.24402531 -2.173502e+02 1941.754

[52,] 45.35212515 0.22805644 1.988636e+02 1941.754

[53,] 68.68157405 0.25038072 2.743086e+02 1941.754

[54,] -21.55658988 0.26082634 -8.264729e+01 1941.754

[55,] -0.63134169 0.29980192 -2.105863e+00 1941.754

[56,] 0.51952603 0.32999814 1.574330e+00 1941.754

[57,] -0.13631262 0.27045525 -5.040117e-01 1941.754

[58,] -0.41622746 0.31196793 -1.334200e+00 1941.754

[59,] 0.39347906 0.24294958 1.619591e+00 1941.754

[60,] 0.59771593 0.24686165 2.421259e+00 1941.754

[61,] 0.42876509 0.30806417 1.391804e+00 1941.754

[62,] 0.01792433 0.24770396 7.236189e-02 1941.754

[63,] 0.41704153 0.25583097 1.630145e+00 1941.754

[64,] 0.37239502 0.30600796 1.216946e+00 1941.754

[65,] 0.09011512 0.16641194 5.415184e-01 1941.754

[66,] 0.40230136 0.22183218 1.813539e+00 1941.754

[67,] 0.17480992 0.18077530 9.670012e-01 1941.754

[68,] 0.17298509 0.20817804 8.309478e-01 1941.754

[69,] -0.87951238 0.24756631 -3.552633e+00 1941.754

[70,] -1.26794409 0.28272355 -4.484749e+00 1941.754

[71,] -1.78522821 0.25806885 -6.917643e+00 1941.754

[72,] -2.27998635 0.28244150 -8.072420e+00 1941.754

[73,] 6.19016362 0.04468928 1.385156e+02 1941.754

[74,] -834.17983379 0.04127790 -2.020888e+04 1941.754

[75,] 72.53617451 0.04219556 1.719048e+03 1941.754

> performance\_4

[1] 0.6350365

> Result\_4

Estimate SE Tvalue minusll

[1,] 0.92895643 0.21803975 4.2604912842 2816.749

[2,] 1.93416180 0.40038915 4.8307048891 2816.749

[3,] 0.98456125 0.23018437 4.2772723560 2816.749

[4,] 0.70373189 0.15369405 4.5787842146 2816.749

[5,] 0.13606696 0.12022525 1.1317669433 2816.749

[6,] 0.52608730 0.13669719 3.8485596421 2816.749

[7,] 0.23001149 0.11924760 1.9288563275 2816.749

[8,] -0.02685910 0.13814423 -0.1944279454 2816.749

[9,] 0.51227125 0.11745612 4.3613840639 2816.749

[10,] 0.71050857 0.13418354 5.2950498952 2816.749

[11,] 0.33967688 0.09383079 3.6201003739 2816.749

[12,] 0.62904168 0.10064235 6.2502684718 2816.749

[13,] 0.14882849 0.09220425 1.6141174596 2816.749

[14,] 0.18630633 0.09364985 1.9893926878 2816.749

[15,] -0.14731942 0.12927171 -1.1396106724 2816.749

[16,] -0.16907233 0.16793378 -1.0067797892 2816.749

[17,] -0.38945216 0.21064037 -1.8488960894 2816.749

[18,] -0.73478776 0.27389412 -2.6827438819 2816.749

[19,] -2.18619069 1.82698339 -1.1966122434 2816.749

[20,] -2.32997500 2.37774746 -0.9799085192 2816.749

[21,] 3.62953829 2.61829760 1.3862206821 2816.749

[22,] -1.35500072 1.72797703 -0.7841543596 2816.749

[23,] 0.96179714 1.31827937 0.7295852128 2816.749

[24,] 0.74464244 1.37091833 0.5431705319 2816.749

[25,] 1.01753524 2.01558501 0.5048336993 2816.749

[26,] 0.02920143 1.50788368 0.0193658382 2816.749

[27,] 0.66885026 1.64822751 0.4057997202 2816.749

[28,] 1.08612656 1.66106806 0.6538724008 2816.749

[29,] -0.04762588 1.56371957 -0.0304567891 2816.749

[30,] 0.74710715 1.56508234 0.4773596426 2816.749

[31,] 0.26531082 1.64113107 0.1616633944 2816.749

[32,] -0.41125065 1.33360886 -0.3083742605 2816.749

[33,] 0.68100877 2.04915070 0.3323370829 2816.749

[34,] 0.76777149 2.10179247 0.3652936729 2816.749

[35,] -0.30657292 2.28669085 -0.1340683728 2816.749

[36,] -0.04952667 2.26517955 -0.0218643484 2816.749

[37,] -0.31278322 NaN NaN 2816.749

[38,] -3.89769642 NaN NaN 2816.749

[39,] 2.24966730 237.12112645 0.0094874182 2816.749

[40,] 2.16437679 78.56270066 0.0275496739 2816.749

[41,] 3.66987654 54.09550587 0.0678406918 2816.749

[42,] -2.47800495 280.84043246 -0.0088235334 2816.749

[43,] 1.61402782 60.33621831 0.0267505631 2816.749

[44,] -3.40200857 126.72584812 -0.0268454196 2816.749

[45,] -0.67972773 218.36907669 -0.0031127472 2816.749

[46,] 2.91232975 391.16763188 0.0074452217 2816.749

[47,] -0.61280026 104.81625770 -0.0058464238 2816.749

[48,] -1.40436716 NaN NaN 2816.749

[49,] -2.12275161 NaN NaN 2816.749

[50,] -1.91492995 NaN NaN 2816.749

[51,] -0.19386224 214.02997350 -0.0009057715 2816.749

[52,] -1.28832446 82.69348858 -0.0155795152 2816.749

[53,] -0.76320588 NaN NaN 2816.749

[54,] 2.15035192 NaN NaN 2816.749

[55,] 0.69517290 0.36018131 1.9300638042 2816.749

[56,] 0.43303218 0.22133869 1.9564233250 2816.749

[57,] 0.87928496 0.46189062 1.9036648991 2816.749

[58,] 0.40013692 0.25660465 1.5593517758 2816.749

[59,] 0.36418913 0.18908943 1.9260152942 2816.749

[60,] 0.59193236 0.23429015 2.5264926592 2816.749

[61,] 0.56522598 0.27417968 2.0615166846 2816.749

[62,] 1.09651020 0.42985785 2.5508669559 2816.749

[63,] 1.11234016 0.30543815 3.6417852922 2816.749

[64,] 1.07104290 0.32107101 3.3358443202 2816.749

[65,] 0.62434507 0.26779115 2.3314626404 2816.749

[66,] 1.34341892 0.34419506 3.9030743857 2816.749

[67,] -0.01830900 0.22396298 -0.0817501161 2816.749

[68,] 0.11095549 0.14462219 0.7672092965 2816.749

[69,] -0.71786799 0.28785780 -2.4938285272 2816.749

[70,] -2.18741083 0.80161621 -2.7287507526 2816.749

[71,] -3.84965710 1.44118449 -2.6711757664 2816.749

[72,] -7.60639403 4.87447378 -1.5604543943 2816.749

[73,] 3.21776271 0.71375077 4.5082441226 2816.749

[74,] 0.03920562 0.02799661 1.4003706159 2816.749

[75,] -5.08752329 0.53280864 -9.5485000306 2816.749

> performance\_4

[1] 0.6642336

> Result\_4

Estimate SE Tvalue minusll

[1,] -2.159860921 0.882652339 -2.44701206 3127.03

[2,] -0.960751290 NaN NaN 3127.03

[3,] 1.584381134 0.376374860 4.20958279 3127.03

[4,] -0.904376712 0.327639630 -2.76027876 3127.03

[5,] -0.604235249 NaN NaN 3127.03

[6,] 0.227931705 0.339629131 0.67111942 3127.03

[7,] -0.320428009 NaN NaN 3127.03

[8,] 0.935553615 0.599885326 1.55955409 3127.03

[9,] 0.604006913 0.283812065 2.12819322 3127.03

[10,] -0.885594351 0.086735890 -10.21024108 3127.03

[11,] -0.235230525 NaN NaN 3127.03

[12,] 0.365371080 NaN NaN 3127.03

[13,] -0.398579688 0.533827361 -0.74664530 3127.03

[14,] -1.346814044 NaN NaN 3127.03

[15,] 0.160994165 0.502255219 0.32054254 3127.03

[16,] 0.512640166 0.724575230 0.70750440 3127.03

[17,] -0.696689612 0.715489843 -0.97372397 3127.03

[18,] -1.382628504 0.826021583 -1.67384065 3127.03

[19,] -0.026591157 0.233008753 -0.11412085 3127.03

[20,] 0.277386250 0.205132711 1.35222827 3127.03

[21,] -0.010565301 0.184371083 -0.05730454 3127.03

[22,] 0.038399594 0.196886215 0.19503445 3127.03

[23,] 0.210226453 0.171842293 1.22336853 3127.03

[24,] 0.628616220 0.163849693 3.83654194 3127.03

[25,] 0.412804776 0.167286873 2.46764596 3127.03

[26,] 0.323941870 0.173678374 1.86518253 3127.03

[27,] 1.017663601 0.172175254 5.91062639 3127.03

[28,] 1.346988109 0.184340446 7.30706764 3127.03

[29,] 0.486860782 0.138062820 3.52637143 3127.03

[30,] 0.750933173 0.132098639 5.68463974 3127.03

[31,] 0.054827223 0.145056719 0.37797093 3127.03

[32,] -0.009131395 0.136196367 -0.06704580 3127.03

[33,] -0.020449757 0.185819094 -0.11005197 3127.03

[34,] 0.182621022 0.187585348 0.97353564 3127.03

[35,] 0.180252343 0.199079243 0.90543012 3127.03

[36,] -0.453225910 0.220578722 -2.05471274 3127.03

[37,] 0.811534722 0.192162077 4.22317835 3127.03

[38,] 0.682117977 0.194547523 3.50617662 3127.03

[39,] 0.511560019 0.177121451 2.88818782 3127.03

[40,] 0.212351536 0.174865462 1.21437094 3127.03

[41,] 0.372061977 0.152924231 2.43298250 3127.03

[42,] 0.617435875 0.156093826 3.95554323 3127.03

[43,] 0.712870584 0.168938398 4.21970725 3127.03

[44,] 0.563268474 0.151920293 3.70765789 3127.03

[45,] 0.549709963 0.151072665 3.63871229 3127.03

[46,] 0.633034865 0.176023819 3.59630229 3127.03

[47,] 0.578754762 0.141713466 4.08397861 3127.03

[48,] 0.971357976 0.139109656 6.98267832 3127.03

[49,] 0.349774013 0.118541705 2.95064098 3127.03

[50,] 0.200753316 0.123031356 1.63172481 3127.03

[51,] -0.471222355 0.140808349 -3.34655124 3127.03

[52,] -1.678985824 0.203357706 -8.25631766 3127.03

[53,] -2.977650611 0.274597486 -10.84369216 3127.03

[54,] -4.292830306 0.384220417 -11.17283234 3127.03

[55,] 0.205846621 0.344430239 0.59764387 3127.03

[56,] 4.492703188 0.570131926 7.88011158 3127.03

[57,] 0.913912405 0.303818503 3.00808672 3127.03

[58,] 0.356259316 0.313536644 1.13626054 3127.03

[59,] -0.072907139 0.285840020 -0.25506274 3127.03

[60,] 0.338474653 0.269390448 1.25644638 3127.03

[61,] 0.222725773 0.281441996 0.79137363 3127.03

[62,] -0.575398249 0.287844969 -1.99898665 3127.03

[63,] 0.571005648 0.267939894 2.13109604 3127.03

[64,] 0.372110668 0.288978050 1.28767797 3127.03

[65,] 0.204890746 0.232077864 0.88285346 3127.03

[66,] 0.876137723 0.228483365 3.83457991 3127.03

[67,] 0.173170946 0.229393029 0.75490936 3127.03

[68,] 0.247725665 0.243914907 1.01562331 3127.03

[69,] -0.161048092 0.377328582 -0.42681127 3127.03

[70,] 0.010812820 0.348333228 0.03104160 3127.03

[71,] 0.127232521 0.354759223 0.35864472 3127.03

[72,] -0.379660894 0.366228422 -1.03667785 3127.03

[73,] 0.042689113 0.004773436 8.94305669 3127.03

[74,] 0.287468832 0.046318714 6.20632156 3127.03

[75,] 0.471633439 0.050962665 9.25448933 3127.03