1. Ngene script for code-MNL

Design

? Anything following a ? is a dead code (comment).

? \*\*\*\*\*\*\*\*\*\*\* General Information about this script \*\*\*\*\*\*\*\*\*\*\*\*\*

? This will generate a homogeneous pivot design.

? We are considering 6 segments for now namely Cat1, Cat2, Cat3, Cat4, Cat5 and Cat6.

? The weight parameters for Fisher information matrix are kept constant for each category. The weights must sum up to 1.

? Each segment has a different reference alternative.

? Ngene cannot accommodate 4 decimal digits for Fisher Weights. So, we adjusted them up to 3 decimal places which can be taken up by Ngene.

? Adding \* to the alts so as to avoid dominant alternative, repetition of bundle of attributes.

? Prior estimates obtained from MNL estimation.

? Run this script for 10 rows twice. You are likely to find 2 unique set of designs.

;alts(Cat1) = alt1\*, alt2\*, alt3\*

;alts(Cat2) = alt1\*, alt2\*, alt3\*

;alts(Cat3) = alt1\*, alt2\*, alt3\*

;alts(Cat4) = alt1\*, alt2\*, alt3\*

;alts(Cat5) = alt1\*, alt2\*, alt3\*

;alts(Cat6) = alt1\*, alt2\*, alt3\*

;rows = 10

;eff = fish1(mnl,d,mean)

;fisher(fish1) = design1(Cat1[0.191], Cat2[0.163], Cat3[0.207], Cat4[0.289], Cat5[0.134], Cat6[0.016])

;model(Cat1):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[7.5] +

b3[(n,-0.0410,0.0139)] \* tts.ref[2] +

b4[(n,-0.0236,0.0157)] \* sn.ref[5] +

b5[(n,-0.694,0.189)] \* vr.ref[0.85]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] + b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat2):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[22.5] +

b3[(n,-0.0410,0.0139)] \* tts.ref[6] +

b4[(n,-0.0236,0.0157)] \* sn.ref[12] +

b5[(n,-0.694,0.189)] \* vr.ref[2.45]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat3):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[45] +

b3[(n,-0.0410,0.0139)] \* tts.ref[11] +

b4[(n,-0.0236,0.0157)] \* sn.ref[20] +

b5[(n,-0.694,0.189)] \* vr.ref[5.3]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat4):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[75] +

b3[(n,-0.0410,0.0139)] \* tts.ref[19] +

b4[(n,-0.0236,0.0157)] \* sn.ref[30] +

b5[(n,-0.694,0.189)] \* vr.ref[12.0]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat5):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[105] +

b3[(n,-0.0410,0.0139)] \* tts.ref[26] +

b4[(n,-0.0236,0.0157)] \* sn.ref[42] +

b5[(n,-0.694,0.189)] \* vr.ref[19.3]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat6):

U(alt1) = b2[(n,-0.136,0.0222)] \* tt.ref[135] +

b3[(n,-0.0410,0.0139)] \* tts.ref[34] +

b4[(n,-0.0236,0.0157)] \* sn.ref[50] +

b5[(n,-0.694,0.189)] \* vr.ref[21.0]/

U(alt2) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.136,0.0222)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0410,0.0139)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0236,0.0157)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.694,0.189)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]$

2. Ngene script for code-RPECL

Design

? Anything following a ? is a dead code (comment).

? \*\*\*\*\*\*\*\*\*\*\* General Information about this script \*\*\*\*\*\*\*\*\*\*\*\*\*

? This will generate a homogeneous pivot design.

? Only one design will be generated for all the six segments.

? The weight parameters for Fisher information matrix are kept constant for each category. The weights must sum up to 1.

? Each segment has a different reference alternative.

? Ngene cannot accommodate 4 decimal digits for Fisher Weights. So, we adjusted them up to 3 decimal places which can be taken up by Ngene.

? Adding \* to the alts so as to avoid dominant alternative, repetition of bundle of attributes.

? The script will only evaluate the performance of previously obtained design under RPEC paradigm. Therefore, the structure of design remains the same.

? Prior estimates obtained from the mixed logit formulation.

? Run this script for 10 rows twice. You are likely to find 2 unique set of designs.

;alts(Cat7) = alt1\*, alt2\*, alt3\*

;alts(Cat8) = alt1\*, alt2\*, alt3\*

;alts(Cat9) = alt1\*, alt2\*, alt3\*

;alts(Cat10) = alt1\*, alt2\*, alt3\*

;alts(Cat11) = alt1\*, alt2\*, alt3\*

;alts(Cat12) = alt1\*, alt2\*, alt3\*

? ////////// Change the saved design file name here /////////

;alg = eval(block1.ngd)

;rows = 10

;eff = fish2(rpecpanel,d,mean)

;rdraws = gauss(2)

;bdraws = gauss(2)

;rep = 1000

;fisher(fish2) = design1(Cat7[0.191], Cat8[0.163], Cat9[0.207], Cat10[0.289], Cat11[0.134], Cat12[0.016])

;model(Cat7):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[7.5] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[2] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[5] +

b93[(n,-1.5784,0.3706)] \* vr.ref[0.85] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat8):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[22.5] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[6] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[12] +

b93[(n,-1.5784,0.3706)] \* vr.ref[2.45] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat9):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[45] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[11] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[20] +

b93[(n,-1.5784,0.3706)] \* vr.ref[5.3] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat10):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[75] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[19] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[30] +

b93[(n,-1.5784,0.3706)] \* vr.ref[12.0] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat11):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[105] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[26] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[42] +

b93[(n,-1.5784,0.3706)] \* vr.ref[19.3] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat12):

U(alt1) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.ref[135] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.ref[34] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.ref[50] +

b93[(n,-1.5784,0.3706)] \* vr.ref[21.0] + s1[ec,0.2]/

U(alt2) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.5]/

U(alt3) = b90[n,(n,-0.2746,0.0544),(n,0.1923,0.0526)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1228,0.0379),(n,0.1843,0.0449)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.1382,0.0654),(n,0.2310,0.0770)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.5784,0.3706)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2 $