1. Ngene script for code\_full\_MNL

Design

? This will generate a homogeneous pivot design for Sydney population.

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

? Giving a RANGE of weights in the Fisher matrix instead of specifying a POINT value.

? The sum of upper bounds for the ranges should sum up to 1.

? If it doesn't work well, I am going to resort to the original syntax.

? The weights were obtained from BTS HTS Report 2014/15 and Kiran Shakeel's survey.

? Each segment has a different reference alternative.

? Ngene cannot accommodate 4 decimal digits for Fisher Weights. So, adjusting 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 model on a dataset of 145 participants.

;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)

? /// Specifying weight range in the Fisher matrix ///

;fisher(fish1) = design1(Cat1[0.30:0.45], Cat2[0.20:0.35], Cat3[0.1:0.2], Cat4[0.02:0.1], Cat5[0.05:0.1], Cat6[0.01:0.05])

;model(Cat1):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[5] +

b3[(n,-0.0679,0.00604)] \* tts.ref[2] +

b4[(n,-0.0256,0.00449)] \* sn.ref[4] +

b5[(n,-0.779,0.0811)] \* vr.ref[0.35]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat2):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[15] +

b3[(n,-0.0679,0.00604)] \* tts.ref[5] +

b4[(n,-0.0256,0.00449)] \* sn.ref[7] +

b5[(n,-0.779,0.0811)] \* vr.ref[1.00]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat3):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[25] +

b3[(n,-0.0679,0.00604)] \* tts.ref[8] +

b4[(n,-0.0256,0.00449)] \* sn.ref[10] +

b5[(n,-0.779,0.0811)] \* vr.ref[2.6]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat4):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[35] +

b3[(n,-0.0679,0.00604)] \* tts.ref[10] +

b4[(n,-0.0256,0.00449)] \* sn.ref[14] +

b5[(n,-0.779,0.0811)] \* vr.ref[3.05]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat5):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[50] +

b3[(n,-0.0679,0.00604)] \* tts.ref[15] +

b4[(n,-0.0256,0.00449)] \* sn.ref[20] +

b5[(n,-0.779,0.0811)] \* vr.ref[3.95]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]

;model(Cat6):

U(alt1) = b2[(n,-0.175,0.00925)] \* tt.ref[75] +

b3[(n,-0.0679,0.00604)] \* tts.ref[20] +

b4[(n,-0.0256,0.00449)] \* sn.ref[30] +

b5[(n,-0.779,0.0811)] \* vr.ref[4.5]/

U(alt2) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%]/

U(alt3) = b2[(n,-0.175,0.00925)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b3[(n,-0.0679,0.00604)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b4[(n,-0.0256,0.00449)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b5[(n,-0.779,0.0811)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] $

2. Ngene script for code\_full\_RPECL

Design

? This will generate a homogeneous pivot design for Sydney population.

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

? Giving a RANGE of weights in the Fisher matrix instead of specifying a POINT value.

? The sum of upper bounds for the ranges should sum up to 1.

? If it doesn't work well, I am going to resort to the original syntax.

? The weights were obtained from BTS HTS Report 2014/15 and Kiran Shakeel's survey.

? Each segment has a different reference alternative.

? Ngene cannot accommodate 4 decimal digits for Fisher Weights. So, adjusting 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 model on a dataset of 145 participants.

;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 file name here /////////

;alg = eval(Sydblock0.ngd)

;rows = 10

;eff = fish2(rpecpanel,d,mean)

;rdraws = gauss(2)

;bdraws = gauss(2)

;rep = 1000

? //// The weights correspond to optimal values obtained from MNL script

;fisher(fish2) = design1(Cat7[0.384], Cat8[0.284], Cat9[0.157], Cat10[0.065], Cat11[0.078], Cat12[0.032])

;model(Cat7):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[5] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[2] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[4] +

b93[(n,-1.1653,0.1148)] \* vr.ref[0.35] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat8):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[15] + b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[5] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[7] +

b93[(n,-1.1653,0.1148)] \* vr.ref[1.00] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat9):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[25] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[8] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[10] +

b93[(n,-1.1653,0.1148)] \* vr.ref[2.6] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,- 25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat10):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[35] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[10] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[14] +

b93[(n,-1.1653,0.1148)] \* vr.ref[3.05] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat11):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[50] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[15] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[20] +

b93[(n,-1.1653,0.1148)] \* vr.ref[3.95] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2

;model(Cat12):

U(alt1) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.ref[75] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.ref[20] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.ref[30] +

b93[(n,-1.1653,0.1148)] \* vr.ref[4.5] + s1[ec,0.1]/

U(alt2) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s1 + s2[ec,0.15]/

U(alt3) = b90[n,(n,-0.3019,0.0238),(n,0.1905,0.0207)] \* tt.piv[-20%,-10%,0%,10%,20%] +

b91[n,(n,-0.1271,0.0149),(n,0.1060,0.0134)] \* tts.piv[-50%,-25%,0%,25%,50%] +

b92[n,(n,-0.0813,0.0110),(n,0.0556,0.0107)] \* sn.piv[-50%,-25%,0%,25%,50%] +

b93[(n,-1.1653,0.1148)] \* vr.piv[-25%,-12.5%,0%,12.5%,25%] + s2 $