**Appendix A**

%% import data and index different types of patches

Results = importdata('Pooled Results.mat');

Results(:,9) = Results(:,8).\*Results(:,9);

Find\_N = Results(:,5) ==1;

% present patch trials -- signal present for hypo 1

Find\_CAP = Results(:,4) == 0 & Results(:,5) == 2;

Find\_IAP = Results(:,4) == 1 & Results(:,5) == 3;

%Congruent trial with Congruent object, and incongruent trial with

%incongruent object -- signal present for hypo 2

Find\_Congruent\_CP = Results(:,4) == 0 & Results(:,5) == 2 & Results(:,6) == 3;

Find\_Incongruent\_IP = Results(:,4) == 1 & Results(:,5) == 3 & Results(:,6) == 3;

%Incongruent trial with congruent object, congruent trial with incongruent

%object -- signal absent for hypo 2

Find\_Congruent\_IP = Results(:,4) == 0 & Results(:,5) == 3 & Results(:,6) == 3;

Find\_Incongruent\_CP = Results(:,4) == 1 & Results(:,5) == 2 & Results(:,6) == 3;

**%% hypothesis 1 analysis**

% hit and CR

matrix1 = zeros(15,2);

for sub = 1:15

R\_indv = Results(Results(:,1)==sub &(Find\_IAP|Find\_CAP|Find\_N),:);

for condition = 1:2

if condition == 1

Find\_patch = R\_indv(:,5)~=1;

Results\_P = R\_indv(Find\_patch,:);

accuracy = sum(Results\_P(:,8)==1)/size(Results\_P,1);

else

Find\_patch = R\_indv(:,5)==1;

Results\_P = R\_indv(Find\_patch,:);

accuracy = sum(Results\_P(:,8)==-1)/size(Results\_P,1);

end

matrix1(sub,condition) = accuracy;

clear Results\_P

clear condition\_mean

clear accuracy

end

end

[h,p,ci] = ttest(matrix1(:,1),matrix1(:,2)) % compare accuracy for present and absent patches

% **output for hypothesis 1 hit and CR comparison**

**h =**

**1**

**p =**

**3.7639e-04**

**ci =**

**-0.2853**

**-0.1051**

**stats =**

**struct with fields:**

**tstat: -4.6482**

**df: 14**

**sd: 0.1627**

% AUC analysis

Results\_NC = Results(Find\_N,:); % find absent patchese

Results\_APC = Results(Find\_IAP|Find\_CAP,:); % find present patches

matrix1 = zeros(15,1);

for sub = 1:15

Results\_NC = Results(Find\_N,:);

Confidence\_N = Results\_NC(Results\_NC(:,1)==sub,9);

Confidence\_AP = Results\_APC(Results\_APC(:,1)==sub,9);

for i = -4:4

Confidence\_APCounts(i+5) = sum(Confidence\_AP == -i);

Confidence\_NCounts(i+5) = sum(Confidence\_N == -i);

end

for i = 1:9

if i == 1

Cumulative\_NCounts(i) = Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i) = Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

else

Cumulative\_NCounts(i) = Cumulative\_NCounts(i-1) + Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i)= Cumulative\_APCounts(i-1)+ Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

end

end

Cumulative\_Hit = [0 Cumulative\_Hit(1:4) Cumulative\_Hit(6:9)];

Cumulative\_FA = [0 Cumulative\_FA(1:4) Cumulative\_FA(6:9)];

AUC = round(AreaUnderROC([Cumulative\_Hit; Cumulative\_FA]'),2);

matrix1(sub,:)= AUC;

end

[h,p,ci]= ttest(matrix1(:,1),0.5) % test AUC against chance

% **output for hypothesis 1 AUC analysis**

**h =**

**1**

**p =**

**3.8344e-12**

**ci =**

**0.7942**

**0.8591**

**stats =**

**struct with fields:**

**tstat: 21.5770**

**df: 14**

**sd: 0.0586**

**%% hypothesis 2**

% AUC calculation

grandmatrix = zeros(15,2);

for condition = 1:2

for sub = 1:15

if condition ==1

Confidence\_P = Results(Find\_Congruent\_CP & Results(:,1)==sub,9);

Confidence\_A = Results(Find\_Congruent\_IP & Results(:,1)==sub,9);

else

Confidence\_P = Results(Find\_Incongruent\_IP & Results(:,1)==sub,9);

Confidence\_A = Results(Find\_Incongruent\_CP & Results(:,1)==sub,9);

end

for i = -4:4

Confidence\_APCounts(i+5) = sum(Confidence\_P == -i);

Confidence\_NCounts(i+5) = sum(Confidence\_A == -i);

end

for i = 1:9

if i == 1

Cumulative\_NCounts(i) = Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_A);

Cumulative\_APCounts(i) = Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_P);

else

Cumulative\_NCounts(i) = Cumulative\_NCounts(i-1) + Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_A);

Cumulative\_APCounts(i)= Cumulative\_APCounts(i-1)+ Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_P);

end

end

Cumulative\_Hit = [0 Cumulative\_Hit(1:4) Cumulative\_Hit(6:9)];

Cumulative\_FA = [0 Cumulative\_FA(1:4) Cumulative\_FA(6:9)];

AUC = round(AreaUnderROC([Cumulative\_Hit; Cumulative\_FA]'),2);

grandmatrix(sub,condition)= AUC;

clear Confidence\_P

clear Confidence\_A

clear Cumulative\_NCounts;

clear Cumulative\_APCounts;

clear Cumulative\_Hit;

clear Cumulative\_FA;

clear AUC;

end

end

% t test

[h,p,ci,stats] = ttest(grandmatrix(:,1),0.5,'Alpha',0.025) % congruent AUC against chance, alpha corrected to 0.025

% **output**

**h =**

**1**

**p =**

**3.0499e-06**

**ci =**

**0.5774**

**0.6559**

**stats =**

**struct with fields:**

**tstat: 7.4620**

**df: 14**

**sd: 0.0606**

[h,p,ci,stats] = ttest(grandmatrix(:,2),0.5,'Alpha',0.025) % incongruent AUC against chance, alpha corrected to 0.025

% **output**

**h =**

**1**

**p =**

**0.0066**

**ci =**

**0.5111**

**0.5929**

**stats =**

**struct with fields:**

**tstat: 3.1889**

**df: 14**

**sd: 0.0632**

[h,p,ci,stats] = ttest(grandmatrix(:,1),matrix1,'Alpha',0.025) % congruent AUC compared with AUC for hypothesis 1, alpha = 0.025

% **output**

**h =**

**1**

**p =**

**2.7492e-11**

**ci =**

**-0.2382**

**-0.1818**

**stats =**

**struct with fields:**

**tstat: -18.6590**

**df: 14**

**sd: 0.0436**

[h,p,ci,stats] = ttest(grandmatrix(:,2),matrix1,'Alpha',0.025)

% **output**

**h =**

**1**

**p =**

**2.3754e-10**

**ci =**

**-0.3181**

**-0.2313**

**stats =**

**struct with fields:**

**tstat: -15.8865**

**df: 14**

**sd: 0.0670**

[h,p,ci,stats] = ttest(grandmatrix(:,1),grandmatrix(:,2)) % congruent v.s. incongruent AUC, alpha = 0.05

%**output**

**h =**

**1**

**p =**

**0.0046**

**ci =**

**0.0234**

**0.1059**

**stats =**

**struct with fields:**

**tstat: 3.3646**

**df: 14**

**sd: 0.0744**

**%% hypothesis 3 analysis**

% create column showing eccentricity of each patch

location1 = (Results(:,7)==2 | Results(:,7)==4| Results(:,7)==6|Results(:,7)== 8) .\* 6.48;

location2 = (Results(:,7)==1 | Results(:,7)==3| Results(:,7)==7|Results(:,7)== 9) .\* 9.16;

eccentricity = zeros(length(Results),1)+ location1 + location2;

% hit and CR analysis

matrix1 = zeros(90,4);

location = [0 6.48 9.16];

for loc = 1:3

for sub = 1:15

R\_indv = Results(Results(:,1)==sub &(Find\_IAP|Find\_CAP|Find\_N)& Results(:,end)==location(loc),:);

for condition = 1:2

if condition == 1

Find\_patch = R\_indv(:,5)~=1;

Results\_P = R\_indv(Find\_patch,:);

accuracy = sum(Results\_P(:,8)==1)/size(Results\_P,1);

else

Find\_patch = R\_indv(:,5)==1;

Results\_P = R\_indv(Find\_patch,:);

accuracy = sum(Results\_P(:,8)==-1)/size(Results\_P,1);

end

rowindex = size(matrix1,1)-sum(matrix1(:,1)==0)+1;

matrix1(rowindex,1) = sub;

matrix1(rowindex,2) = condition - 1;

matrix1(rowindex,3) = location(loc);

matrix1(rowindex,4) = accuracy;

clear Results\_P

clear condition\_mean

clear accuracy

end

end

end

% fit lme, for hit

data1 = table(matrix1(matrix1(:,2)==0,1),matrix1(matrix1(:,2)==0,3),matrix1(matrix1(:,2)==0,4),...

'VariableName',{'Subjects','Location','Accuracy'});

lm1 = fitlme(data1,'Accuracy ~ Location + (1|Subjects) + (Location|Subjects) ');

lm2 = fitlme(data1,'Accuracy ~ 1+ (1|Subjects) + (Location|Subjects)');

compare(lm2,lm1)

% assumption checks for the full model

subplot(1,2,1), plotResiduals(lm1, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

subplot(1,2,2), plotResiduals(lm1,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

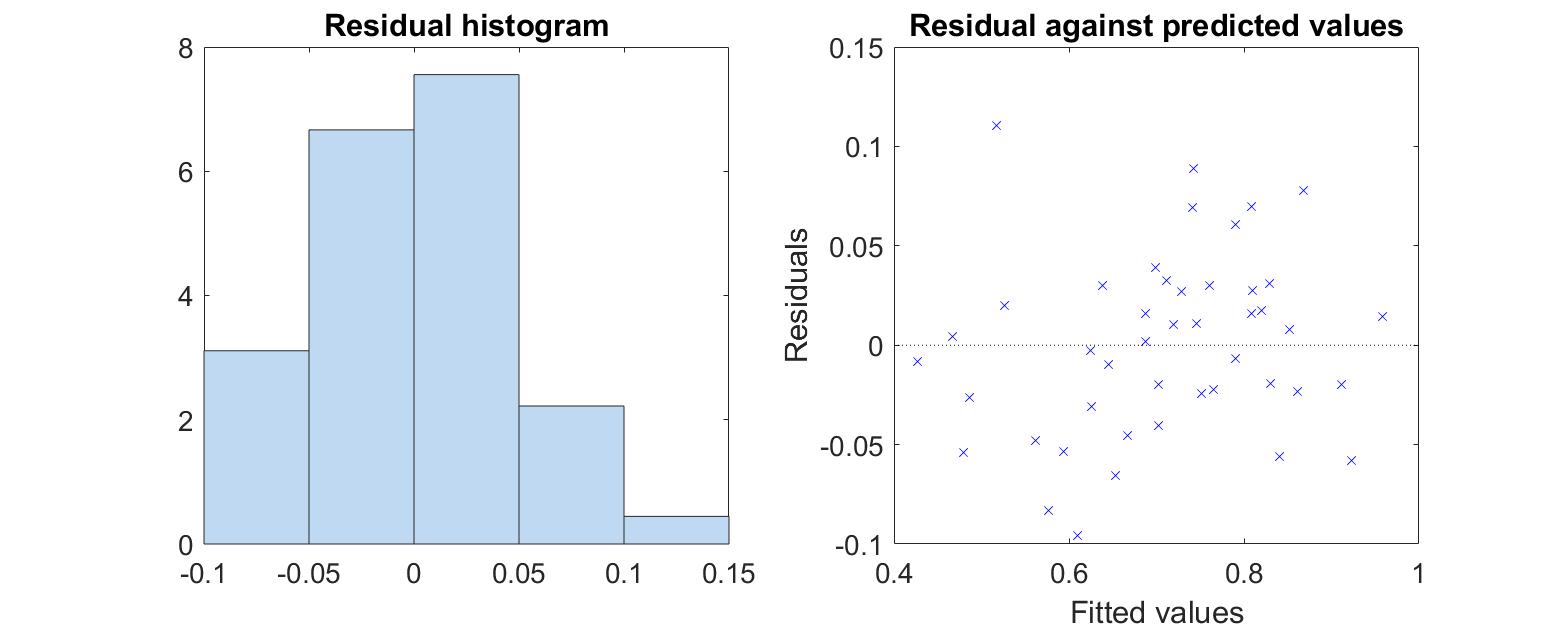
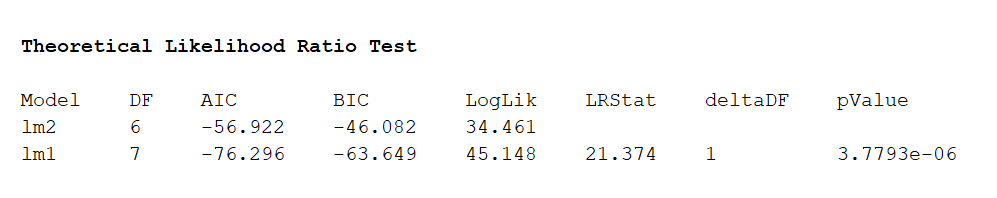


Figure 1. Residual distribution (left), and residual plotted against predicted values (right) for the LME model built for hit rate estimated in hypothesis 3.

% **output**



% fit lme, for CR

data2 = table(matrix1(matrix1(:,2)==1,1),matrix1(matrix1(:,2)==1,3),matrix1(matrix1(:,2)==1,4),...

'VariableName',{'Subjects','Location','Accuracy'});

lm3 = fitlme(data2,'Accuracy ~ Location + (1|Subjects) + (Location|Subjects) ');

lm4 = fitlme(data2,'Accuracy ~ 1+ (1|Subjects) + (Location|Subjects)');

compare(lm4,lm3)

%assumption checks for lm3

subplot(1,2,1), plotResiduals(lm3, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

subplot(1,2,2), plotResiduals(lm3,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

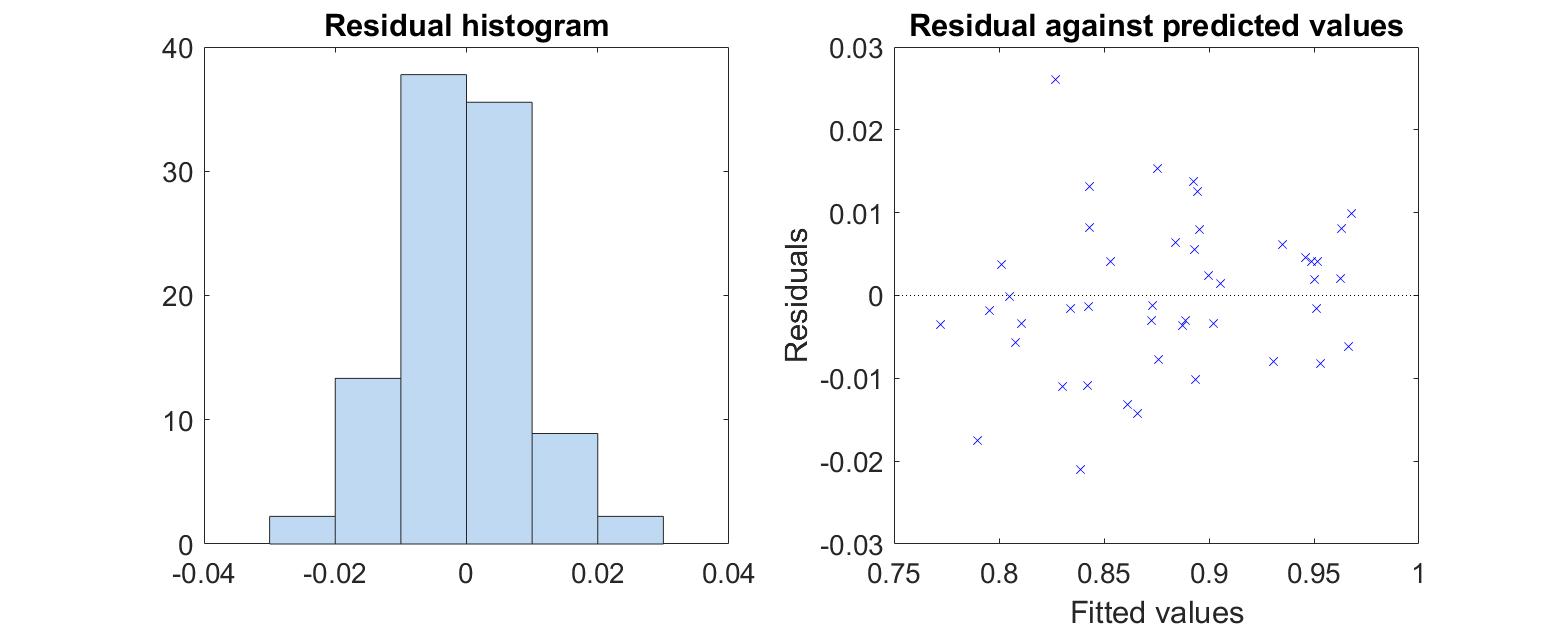
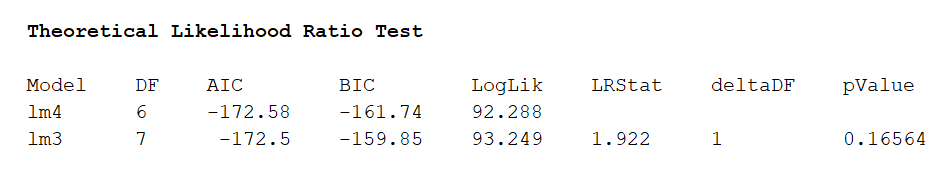


Figure 2. Residual distribution (left), and residual plotted against predicted values (right) for the LME model built for correct rejection rate estimated in hypothesis 3.

%**output**



% AUC calculation, on each eccentricity levels

matrix3 = zeros(15, 3);

for sub = 1:15

indvN = Results(Results(:,1)==sub & Find\_N,:);

indvP = Results(Results(:,1)==sub & (Find\_CAP|Find\_IAP),:); % trial classification

for a = 1:3

indvN\_loc = indvN(indvN(:,end)== location(a),:); % select trials on that location

indvP\_loc = indvP(indvP(:,end) == location(a),:);

Confidence\_N = indvN\_loc(:,9);

Confidence\_AP = indvP\_loc(:,9);

for i = -4:4

Confidence\_APCounts(i+5) = sum(Confidence\_AP == -i); % AUC calculation

Confidence\_NCounts(i+5) = sum(Confidence\_N == -i);

end

for i = 1:9

if i == 1

Cumulative\_NCounts(i) = Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i) = Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

else

Cumulative\_NCounts(i) = Cumulative\_NCounts(i-1) + Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i)= Cumulative\_APCounts(i-1)+ Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

end

end

Cumulative\_Hit = [0 Cumulative\_Hit(1:4) Cumulative\_Hit(6:9)];

Cumulative\_FA = [0 Cumulative\_FA(1:4) Cumulative\_FA(6:9)];

AUC = round(AreaUnderROC([Cumulative\_Hit; Cumulative\_FA]'),2);

matrix3(sub,a)= AUC;

end

end

% sort AUC, location and individual ID into a table, then fit lme model

AUC = reshape(matrix3,[45,1]);

subjects(1:15,1)= 1:1:15;

subjects = repmat(subjects,[3,1]);

location = [zeros(15,1); 6.48 + zeros(15,1); 9.16 + zeros(15,1)];

data = table(AUC, subjects, location,'VariableName',{'AUC','subject','location'});

lm1 = fitlme(data, 'AUC ~ location + (location|subject) + (1|subject)');

lm2 = fitlme(data, 'AUC ~ 1 +(location|subject) + (1|subject) ');

compare(lm2,lm1)

%assumption checks for the full model

subplot(1,2,1), plotResiduals(lm1, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

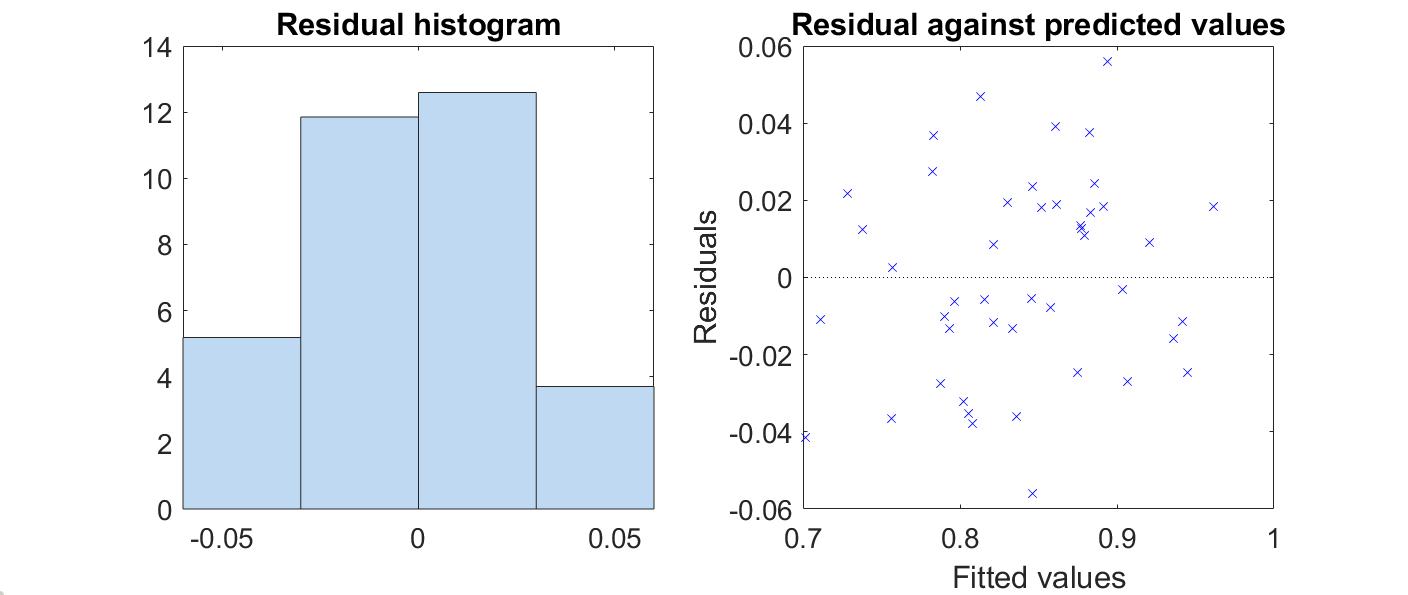
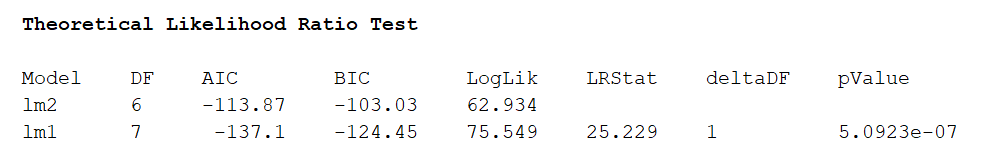


Figure 3. Residual distribution (left), and residual plotted against predicted values (right) for the LME model built for AUC estimated in hypothesis 3.

subplot(1,2,2), plotResiduals(lm1,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

% **output**



**%% hypothesis 4**

%congruent

clear indvN

clear indvP

clear indvN\_loc

clear indvP\_loc

matrix4 = zeros(15, 3);

location = [0 6.48 9.16];

for sub = 1:15

indvN = Results(Results(:,1)==sub & Find\_Congruent\_IP,:);

indvP = Results(Results(:,1)==sub & Find\_Congruent\_CP,:); % trial classification

for a = 1:3

indvN\_loc = indvN(indvN(:,end)== location(a),:); % select trials on that location

indvP\_loc = indvP(indvP(:,end) == location(a),:);

Confidence\_N = indvN\_loc(:,9);

Confidence\_AP = indvP\_loc(:,9);

for i = -4:4

Confidence\_APCounts(i+5) = sum(Confidence\_AP == -i); % AUC calculation

Confidence\_NCounts(i+5) = sum(Confidence\_N == -i);

end

for i = 1:9

if i == 1

Cumulative\_NCounts(i) = Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i) = Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

else

Cumulative\_NCounts(i) = Cumulative\_NCounts(i-1) + Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i)= Cumulative\_APCounts(i-1)+ Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

end

end

Cumulative\_Hit = [0 Cumulative\_Hit(1:4) Cumulative\_Hit(6:9)];

Cumulative\_FA = [0 Cumulative\_FA(1:4) Cumulative\_FA(6:9)];

AUC = round(AreaUnderROC([Cumulative\_Hit; Cumulative\_FA]'),2);

matrix4(sub,a)= AUC;

clear Confidence\_P

clear Confidence\_A

clear Cumulative\_NCounts;

clear Cumulative\_APCounts;

clear Cumulative\_Hit;

clear Cumulative\_FA;

clear AUC;

end

end

%incongruent condition

clear indvN

clear indvP

clear indvN\_loc

clear indvP\_loc

clear population\_m

clear indv\_mean

clear indv\_dff

matrix5 = zeros(15, 3);

for sub = 1:15

indvN = Results(Results(:,1)==sub & Find\_Incongruent\_CP,:);

indvP = Results(Results(:,1)==sub & Find\_Incongruent\_IP,:); % trial classification

for a = 1:3

indvN\_loc = indvN(indvN(:,end)== location(a),:); % select trials on that location

indvP\_loc = indvP(indvP(:,end) == location(a),:);

Confidence\_N = indvN\_loc(:,9);

Confidence\_AP = indvP\_loc(:,9);

for i = -4:4

Confidence\_APCounts(i+5) = sum(Confidence\_AP == -i); % AUC calculation

Confidence\_NCounts(i+5) = sum(Confidence\_N == -i);

end

for i = 1:9

if i == 1

Cumulative\_NCounts(i) = Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i) = Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

else

Cumulative\_NCounts(i) = Cumulative\_NCounts(i-1) + Confidence\_NCounts(i);

Cumulative\_FA(i) = Cumulative\_NCounts(i)/length(Confidence\_N);

Cumulative\_APCounts(i)= Cumulative\_APCounts(i-1)+ Confidence\_APCounts(i);

Cumulative\_Hit(i) = Cumulative\_APCounts(i)/length(Confidence\_AP);

end

end

Cumulative\_Hit = [0 Cumulative\_Hit(1:4) Cumulative\_Hit(6:9)];

Cumulative\_FA = [0 Cumulative\_FA(1:4) Cumulative\_FA(6:9)];

AUC = round(AreaUnderROC([Cumulative\_Hit; Cumulative\_FA]'),2);

matrix5(sub,a)= AUC;

clear Confidence\_P

clear Confidence\_A

clear Cumulative\_NCounts;

clear Cumulative\_APCounts;

clear Cumulative\_Hit;

clear Cumulative\_FA;

clear AUC;

end

end

AUC = [reshape(matrix4,[45,1]); reshape(matrix5,[45,1])];

clear subjects

subjects(1:15,1)= 1:1:15;

subjects = repmat(subjects,[6,1]);

eccentricity = [zeros(15,1); 6.48 + zeros(15,1); 9.16 + zeros(15,1)];

eccentricity = [eccentricity; eccentricity];

condition = [zeros(45,1);ones(45,1)];

data = table(AUC, subjects, eccentricity,condition,'VariableName',{'AUC','subject','eccentricity','condition'});

lm4 = fitlme(data, 'AUC ~ eccentricity\*condition + (eccentricity|subject) + (condition|subject)+(1|subject)');

lml1 = fitlme(data, 'AUC ~ eccentricity:condition + condition + (eccentricity|subject) + (condition|subject)+(1|subject)');

lml2 = fitlme(data, 'AUC ~ eccentricity + condition + (eccentricity|subject) + (condition|subject)+(1|subject)');

lml3 = fitlme(data, 'AUC ~ eccentricity:condition + eccentricity + (eccentricity|subject) + (condition|subject)+(1|subject)');

c\_ecc = compare(lml1,lm4)

c\_interaction = compare(lml2,lm4)

c\_condition = compare(lml3,lm4)

% assumption check for full model

subplot(1,2,1), plotResiduals(lm4, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

subplot(1,2,2), plotResiduals(lm4,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

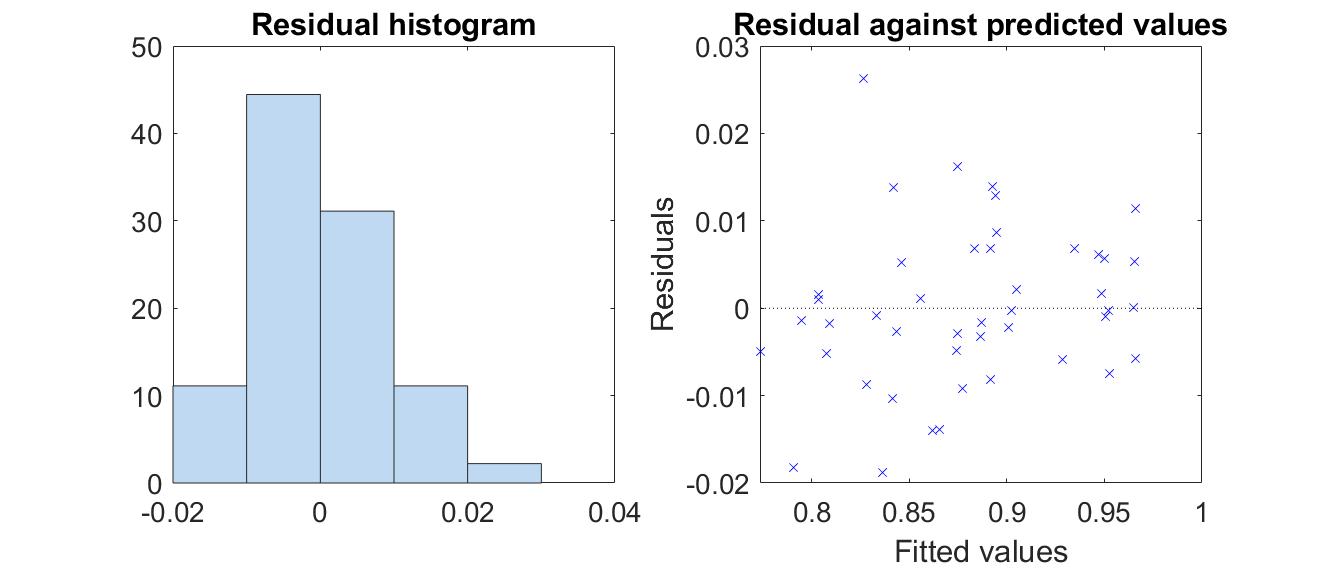
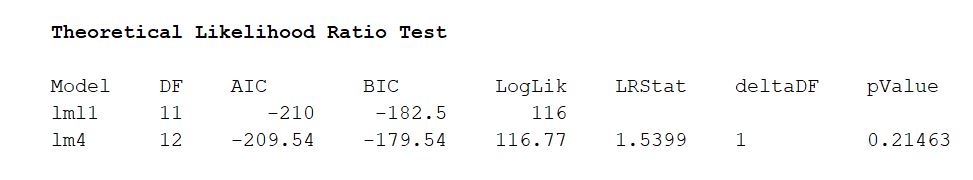
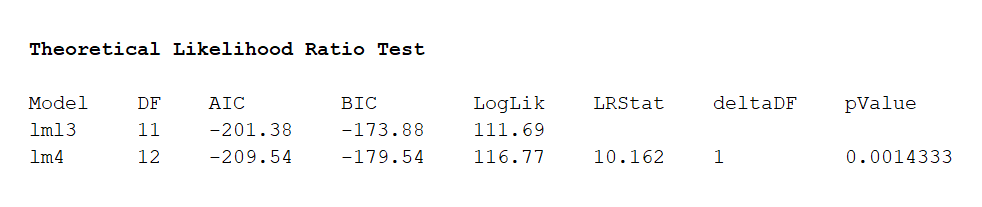


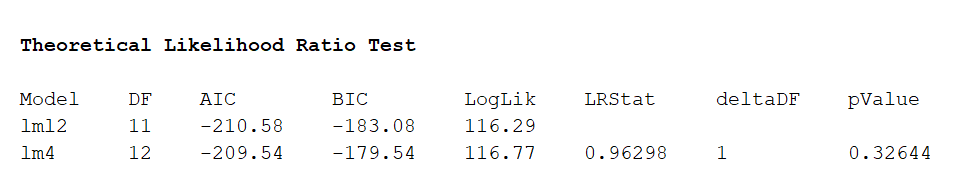
Figure 4. Residual distribution (left), and residual plotted against predicted values (right) for the full LME model built for AUC estimated in hypothesis 4.

**%output for eccentricity**



% **output for condition**



% **output for interaction**

%post-hoc tests, estimating lme model on congruent condition

data2 = table(AUC(1:45,:),subjects(1:45,:),eccentricity(1:45,:), condition(1:45,:),'VariableName',{'AUC','subject','eccentricity','condition'});

lm5 = fitlme(data2, 'AUC ~ eccentricity + (eccentricity|subject)+(1|subject)')

lml4 = fitlme(data2, 'AUC ~ 1 + (eccentricity|subject)+(1|subject)')

compare(lml4,lm5)

% assumption check

subplot(1,2,1), plotResiduals(lm5, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

subplot(1,2,2), plotResiduals(lm5,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

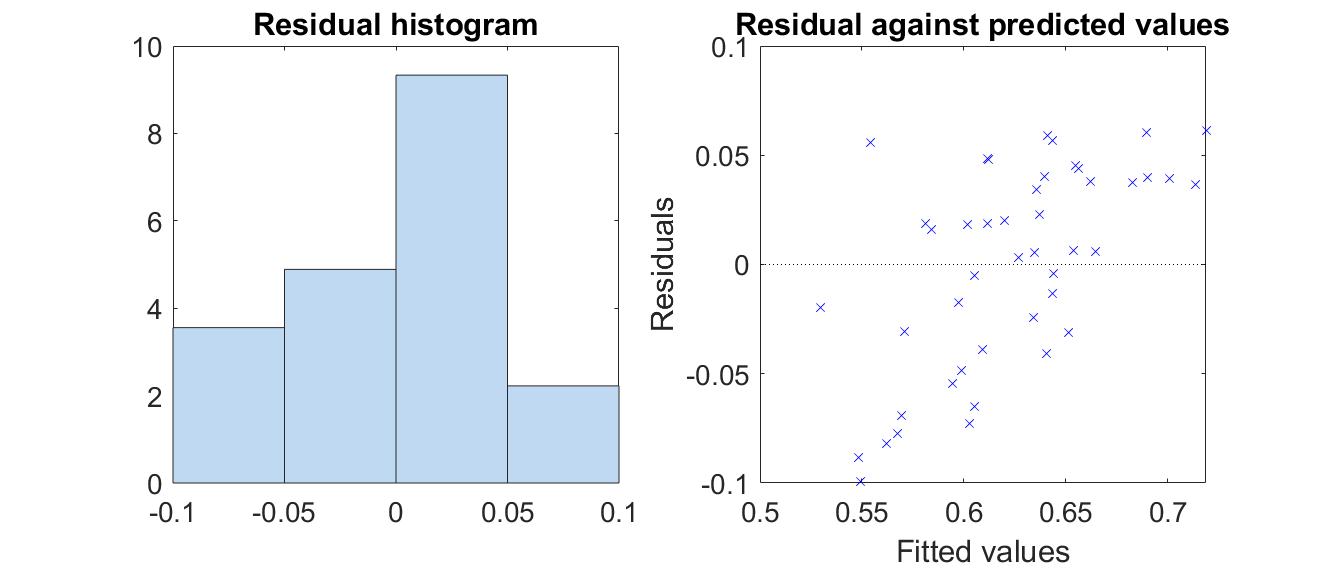
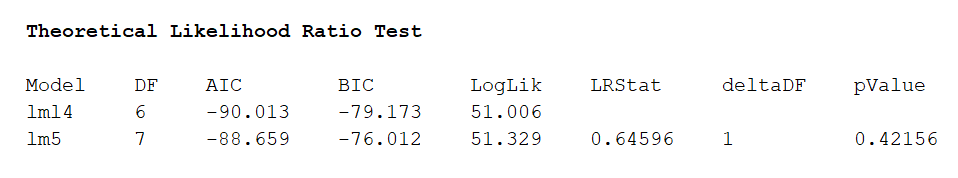


Figure 5. Residual distribution (left), and residual plotted against predicted values (right) for the LME model built for AUC estimated in congruent condition for hypothesis 4.

**% output**

****

data3 = table(AUC(46:90,:),subjects(46:90,:),eccentricity(46:90,:), condition(46:90,:),'VariableName',{'AUC','subject','eccentricity','condition'});

lm6 = fitlme(data3, 'AUC ~ eccentricity + (eccentricity|subject) + (1|subject)')

lml5 = fitlme(data3, 'AUC ~ 1 + (eccentricity|subject) + (1|subject)')

compare(lml5,lm6)

% assumption checks

subplot(1,2,1), plotResiduals(lm6, 'histogram'),title('Residual histogram'),set(gca,'FontSize',14);

subplot(1,2,2), plotResiduals(lm6,'fitted'), title('Residual against predicted values'),set(gca,'FontSize',14);

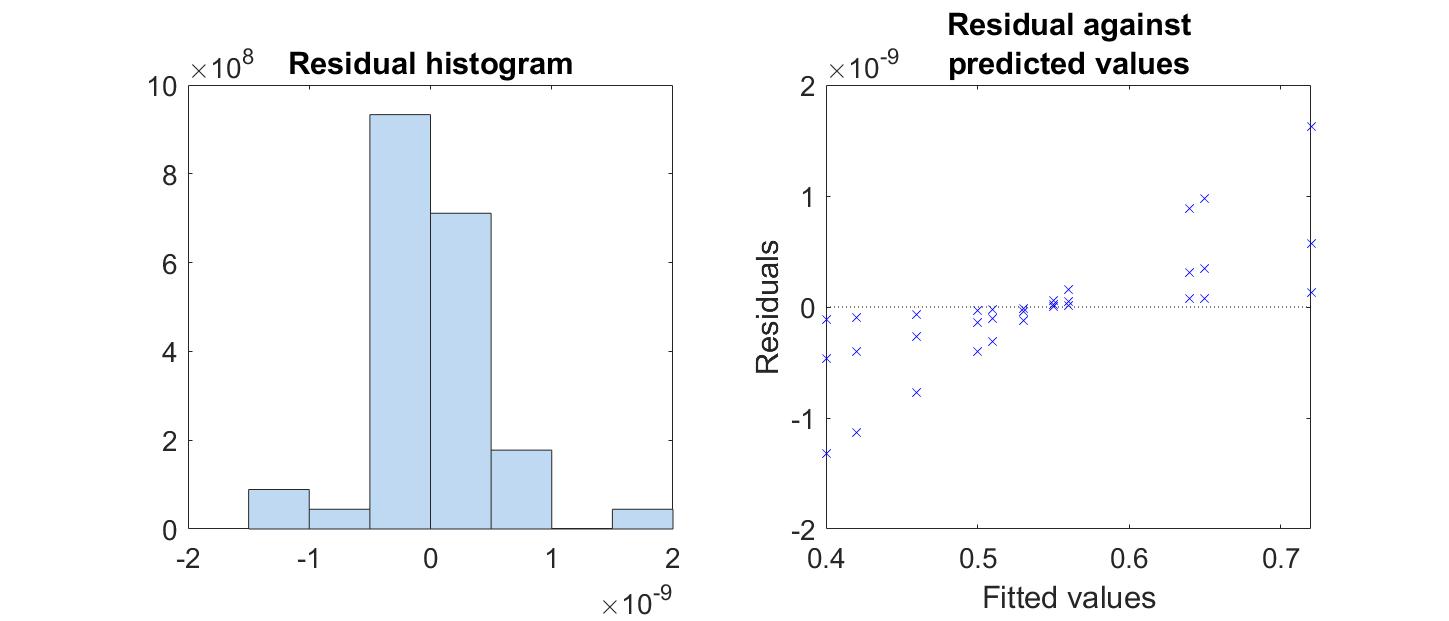
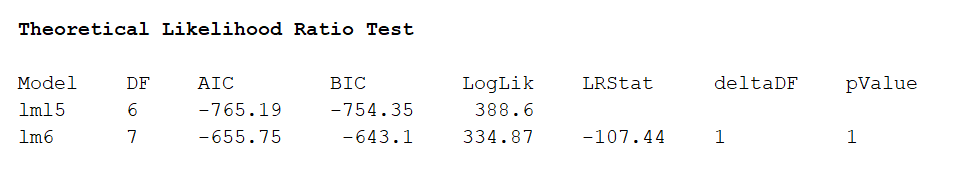


Figure 6. Residual distribution (left), and residual plotted against predicted values (right) for the LME model built for AUC estimated in incongruent condition for hypothesis 4.

**% output**

****