R Markdown: Links between personality, reproductive success and re-pairing patterns in a long-lived seabird

Table of Contents

[Kiitiwake re-pairing model script 1](#_Toc141259521)

[Data Setup, Descriptive Statistics and Variable Refinement 1](#_Toc141259522)

[M1: male boldness\_female boldness- Assortative Mating 2](#_Toc141259523)

[M2: boldness\_breeding outcome- Boldness and Reproductive Outcome 6](#_Toc141259524)

[M3: breeding outcome\_re-pairing outcome- Reproductive outcome and re-pairing probability 13](#_Toc141259525)

[M4: re-pairing outcome\_breeding outcome- Re-pairing and future reproductive outcomes 17](#_Toc141259526)

[M5: Boldness\_re-pairing outcome- boldness and re-pairing probability 19](#_Toc141259527)

[Permutation tests 24](#_Toc141259528)

## Kiitiwake re-pairing model script

This R Markdown document details the code used to generate the results for a study on the links between personality (boldness), breeding outcome and re-pairing in black-legged kittiwakes. It covers the selection criteria used to create multiple datasets suitable for analysis in various different models used to analyse the data.

## Data Setup, Descriptive Statistics and Variable Refinement

The following packages are used in this script. The data are uploaded to r studio. The categorical variables are defined as factors.

library(dplyr)  
library(ggplot2)  
library(lme4)  
library(MuMIn)  
library(sjPlot)  
library(car)  
library(knitr)  
library(lmerTest)  
  
  
absol\_hist<-read.csv ("absol\_hist.csv") # this dataset is used to test for assortative mating (model 1)  
all\_kitt<-read.csv("all\_kitt\_data3.csv") #this dataset is used for models 2, 3 and 5  
next\_season\_repro<-read.csv("next\_season\_repro.csv") # this dataset is used for model 4  
pers\_colony<-read.csv("pers\_scores\_with\_colony.csv") # this dataset and the one below are used to conduct the randomisations  
absol\_ran<-read.csv("absol\_random.csv")  
  
  
absol\_hist$colony<-as.factor(absol\_hist$colony)  
absol\_hist$male\_ring<-as.factor(absol\_hist$male\_ring)  
absol\_hist$female\_ring.x<-as.factor(absol\_hist$female\_ring)  
  
  
all\_kitt$colony<-as.factor(all\_kitt$colony)  
all\_kitt$male\_ring<-as.factor(all\_kitt$male\_ring)  
all\_kitt$female\_ring<-as.factor(all\_kitt$female\_ring)  
all\_kitt$year<-as.factor(all\_kitt$year)  
all\_kitt$chick\_sur<-as.factor(all\_kitt$chick\_sur)  
all\_kitt$season\_outcome<-as.factor(all\_kitt$season\_outcome)  
all\_kitt$ulti\_outcome<-as.factor(all\_kitt$ulti\_outcome)

## M1: male boldness\_female boldness- Assortative Mating

This code shows we tested for assortative mating by boldness at the Grumant and Pyramiden colonies. First the model is set up by transforming the male boldness scores with a square root transformation (following reflection) to correct for negative skew (and then returning the scores to the original direction) and scaling the female boldness scores. We also check for colinearity between the continuous variables.

#M1: female boldness\_male boldness- assortative mating  
#create transformed response variable and scale the female boldness scores  
  
absol\_hist$male\_sqrt<-sqrt(max(absol\_hist$male\_pers.x +1 )-absol\_hist$male\_pers.x) # correct for negative skew  
absol\_hist$male\_sqrt2<- max(absol\_hist$male\_sqrt+1)-absol\_hist$male\_sqrt # return figures to correct direction  
  
absol\_hist\_g<-ggplot(absol\_hist, aes (x = male\_sqrt2))+ # check success  
 geom\_histogram()  
  
absol\_hist$female\_pers\_sc<-scale(absol\_hist$female\_pers.x) # scale female pers variable  
  
#check for colinearity  
colin\_model <- glm(male\_sqrt2~ colony +female\_pers\_sc, data = absol\_hist)  
summary(colin\_model)

##   
## Call:  
## glm(formula = male\_sqrt2 ~ colony + female\_pers\_sc, data = absol\_hist)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.48977 -0.22978 0.07844 0.20550 0.48036   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.712065 0.043864 39.031 < 2e-16 \*\*\*  
## colonypyramiden -0.229438 0.052359 -4.382 2.38e-05 \*\*\*  
## female\_pers\_sc 0.009841 0.023634 0.416 0.678   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.06868394)  
##   
## Null deviance: 10.5950 on 134 degrees of freedom  
## Residual deviance: 9.0663 on 132 degrees of freedom  
## AIC: 26.517  
##   
## Number of Fisher Scoring iterations: 2

vif(colin\_model) # vif scores < 5 acceptable

## colony female\_pers\_sc   
## 1.089787 1.089787

We then build a linear mixed model testing the relationships between male boldness score and female boldness score within pairs. We use a random effect of female ID to account for repeated measures.Colony and the interaction between colony and female boldness were included as fixed effects. The R squared values for the model are also produced.

#linear mixed model testing relationship between male and female boldness scores, interacting with colony  
mating\_model<-lmer(male\_sqrt2 ~ female\_pers\_sc+  
 colony+  
 colony:female\_pers\_sc+  
 (1|female\_ring.x), #female id random effect  
 data = absol\_hist, na.action=na.fail, REML = FALSE)  
summary(mating\_model)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's  
## method [lmerModLmerTest]  
## Formula: male\_sqrt2 ~ female\_pers\_sc + colony + colony:female\_pers\_sc +   
## (1 | female\_ring.x)  
## Data: absol\_hist  
##   
## AIC BIC logLik deviance df.resid   
## 23.4 40.9 -5.7 11.4 129   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.9632 -0.8776 0.2672 0.6578 2.0490   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## female\_ring.x (Intercept) 0.01455 0.1206   
## Residual 0.04981 0.2232   
## Number of obs: 135, groups: female\_ring.x, 112  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)  
## (Intercept) 1.67263 0.04712 116.13846 35.494 < 2e-16  
## female\_pers\_sc 0.09082 0.04467 117.42115 2.033 0.044318  
## colonypyramiden -0.19372 0.05440 115.08440 -3.561 0.000539  
## female\_pers\_sc:colonypyramiden -0.10877 0.05251 118.33683 -2.071 0.040498  
##   
## (Intercept) \*\*\*  
## female\_pers\_sc \*   
## colonypyramiden \*\*\*  
## female\_pers\_sc:colonypyramiden \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) fml\_p\_ clnypy  
## feml\_prs\_sc -0.405   
## colnypyrmdn -0.866 0.351   
## fml\_prs\_sc: 0.344 -0.851 -0.243

r.squaredGLMM(mating\_model)

## R2m R2c  
## [1,] 0.1710973 0.3585054

The model selection criteria (see paper) are applied. This involves building every possible model and ranking them according to AICc. We remove models where delta is < 2 and nested models.

mating\_mod<-dredge(mating\_model, extra = alist(deviance))  
mat\_sb<-subset(mating\_mod, delta<2) #subset all models with delta < 2  
mating\_mod\_sub<- subset(mat\_sb, !nested(.))# remove nested models  
mating\_mod\_coeffs<-coefTable(mating\_mod\_sub, full=TRUE) # create final table of coefficients  
mating\_mod\_coeffs

## $`8`  
## Estimate Std. Error df  
## (Intercept) 1.672625 0.047124 129  
## colonypyramiden -0.193720 0.054402 129  
## female\_pers\_sc 0.090817 0.044674 129  
## colonypyramiden:female\_pers\_sc -0.108770 0.052511 129  
##   
## $`2`  
## Estimate Std. Error df  
## (Intercept) 1.710744 0.044086 131  
## colonypyramiden -0.228015 0.051779 131

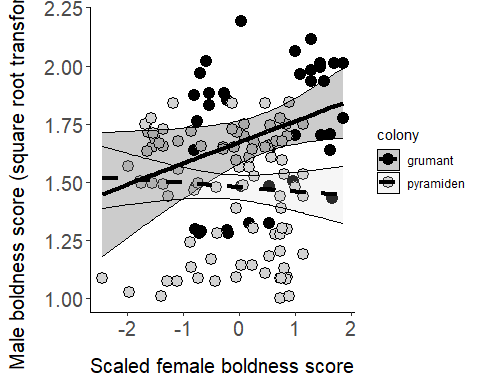
Two models (with all fixed effects) is retained. This suggests there is evidence of assortative mating at Grumant, while mating at Pyramiden is weakly disassortative.

The data are extracted from the model for visualisation.

# reconfiguring the data produced by the model for visualisation  
mating\_model\_g<-plot\_model(mating\_model, type = "pred", terms=c ("female\_pers\_sc[all]", "colony")) # extract the data from the model  
  
mating\_model\_g<-as.data.frame(mating\_model\_g[["data"]]) # convert to a data frame  
mating\_model\_g$female\_pers\_sc<-mating\_model\_g$x # rename variables  
mating\_model\_g$male\_pers\_sc<-mating\_model\_g$predicted  
mating\_model\_g$colony<-mating\_model\_g$group

A graph (Figure 2) is then constructed which shows how mating patterns differ between colonies, using original data and estimates from the best fitting model.

#plot graph  
mating\_plot2<-ggplot()+  
 geom\_point(data=absol\_hist, aes(x = female\_pers\_sc, y = male\_sqrt2, fill = colony), colour ="black",pch=21, size = 4)+  
 geom\_ribbon(data=mating\_model\_g, aes(x = female\_pers\_sc, ymin = conf.low, ymax = conf.high, colour = colony,fill = colony), alpha= 0.2)+  
 geom\_line(data=mating\_model\_g, aes(x = female\_pers\_sc, y = male\_pers\_sc, colour = colony, linetype = colony), linewidth = 1.5)+  
 scale\_linetype\_manual(values = c("solid","dashed"))+  
 scale\_colour\_manual(values=c("black", "black"))+  
 scale\_fill\_manual(values=c("black", "lightgrey"))+  
 theme\_classic()+  
 xlab("Scaled female boldness score")+  
 ylab("Male boldness score (square root transformed)")+  
 theme(axis.text = element\_text(size = 15))+  
 theme(axis.title= element\_text(size=15))+  
 theme (axis.title.y = element\_text (margin = margin (t = 0, r = 13, b = 0, l = 0)))+  
 theme (axis.title.x = element\_text (margin = margin (t = 13, r = 0, b = 0, l = 0)))  
  
mating\_plot2



## M2: boldness\_breeding outcome- Boldness and Reproductive Outcome

In this model the influence of the boldness metrics on reproductive success in year t are evaluated.To begin, the data are filtered so that any missing data from the variables required for this model are removed and the continuous variables are scaled.

male\_pers\_sc = the scaled male’s boldness score female\_pers\_sc = the scaled female’s boldness score absol\_pers\_sc = the scaled absolute difference between pair members’ boldness scores

#M2: boldness\_breeding outcome- Boldness and Reproductive Outcome  
  
# exclusions  
repro\_pers<-all\_kitt[!is.na(all\_kitt$absol),] # pairs where required pers data are missing removed  
repro\_pers<-repro\_pers[!is.na(repro\_pers$chick\_sur),]# pairs where repro data not available  
  
#scaling  
repro\_pers$male\_pers\_sc<-scale(repro\_pers$male\_pers)   
repro\_pers$female\_pers\_sc<-scale(repro\_pers$female\_pers)   
repro\_pers$absol\_pers\_sc<-scale(repro\_pers$absol)

Exploring the data gives us information on the boldness variables within and between sexes and colonies. We also retrieve details of the sample sizes included in the model.

#males slightly bolder than females overall  
repro\_pers%>%summarise(average=mean(male\_pers),med=median(male\_pers),   
 sd=sd(male\_pers))

## average med sd  
## 1 0.9998323 1.236015 0.9142968

repro\_pers%>%summarise(average=mean(female\_pers),med=median(female\_pers),   
 sd=sd(female\_pers))

## average med sd  
## 1 0.8349739 0.8927347 0.9401638

#both males and females bolder at Grumant  
repro\_pers%>%group\_by(colony)%>%summarise(average=mean(male\_pers),med=median(male\_pers),   
 sd=sd(male\_pers))

## # A tibble: 2 × 4  
## colony average med sd  
## <fct> <dbl> <dbl> <dbl>  
## 1 grumant 1.50 1.68 0.734  
## 2 pyramiden 0.617 0.962 0.852

repro\_pers%>%group\_by(colony)%>%summarise(average=mean(female\_pers),med=median(female\_pers),   
 sd=sd(female\_pers))

## # A tibble: 2 × 4  
## colony average med sd  
## <fct> <dbl> <dbl> <dbl>  
## 1 grumant 1.24 1.64 0.900  
## 2 pyramiden 0.524 0.722 0.850

unique(repro\_pers$pair\_ID)#124 pairs  
unique(repro\_pers$cycle\_pair)#213 breeding attempts  
  
unique(repro\_pers$male\_ring)#104 male birds  
unique(repro\_pers$female\_ring)#107 female birds  
  
unique(repro\_pers$year) #11 breeding seasons  
# sample size per year  
with(repro\_pers, tapply(pair\_ID, year, FUN = function(x) length(unique(x))))   
#pairs per colony  
with(repro\_pers, tapply(pair\_ID, colony, FUN = function(x) length(unique(x))))

We check for multi-colinearity again. Any VIF scores greater than 5 would be a concern. These are all below that threshold.

#assumption of multicolinearity  
  
colin\_model <- glm(chick\_sur ~ male\_pers\_sc +female\_pers\_sc, data = repro\_pers, family = "binomial")  
summary(colin\_model)

##   
## Call:  
## glm(formula = chick\_sur ~ male\_pers\_sc + female\_pers\_sc, family = "binomial",   
## data = repro\_pers)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.439 -1.152 -1.021 1.183 1.365   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.008873 0.137951 -0.064 0.9487   
## male\_pers\_sc -0.235043 0.142358 -1.651 0.0987 .  
## female\_pers\_sc 0.071904 0.140955 0.510 0.6100   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 295.28 on 212 degrees of freedom  
## Residual deviance: 292.47 on 210 degrees of freedom  
## AIC: 298.47  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model)

## male\_pers\_sc female\_pers\_sc   
## 1.040712 1.040712

colin\_model <- glm(chick\_sur ~ male\_pers\_sc +absol\_pers\_sc, data = repro\_pers, family = "binomial")  
summary(colin\_model)

##   
## Call:  
## glm(formula = chick\_sur ~ male\_pers\_sc + absol\_pers\_sc, family = "binomial",   
## data = repro\_pers)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4933 -1.1568 -0.8797 1.1418 1.5225   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.01023 0.13892 -0.074 0.9413   
## male\_pers\_sc -0.25497 0.14186 -1.797 0.0723 .  
## absol\_pers\_sc -0.25248 0.14176 -1.781 0.0749 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 295.28 on 212 degrees of freedom  
## Residual deviance: 289.50 on 210 degrees of freedom  
## AIC: 295.5  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model)

## male\_pers\_sc absol\_pers\_sc   
## 1.02378 1.02378

colin\_model <- glm(chick\_sur ~ female\_pers\_sc +absol\_pers\_sc, data = repro\_pers, family = "binomial")  
summary(colin\_model)

##   
## Call:  
## glm(formula = chick\_sur ~ female\_pers\_sc + absol\_pers\_sc, family = "binomial",   
## data = repro\_pers)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3262 -1.1631 -0.9363 1.1500 1.4583   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.009909 0.137903 -0.072 0.943  
## female\_pers\_sc -0.063100 0.149113 -0.423 0.672  
## absol\_pers\_sc -0.241363 0.150073 -1.608 0.108  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 295.28 on 212 degrees of freedom  
## Residual deviance: 292.61 on 210 degrees of freedom  
## AIC: 298.61  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model)

## female\_pers\_sc absol\_pers\_sc   
## 1.16371 1.16371

We then build the global model. Here, chick\_sur represents the binary breeding outcome for year t. We also generate R squared values for the model.

repro\_pers\_mod<- glmer(chick\_sur ~ female\_pers\_sc+  
 male\_pers\_sc+  
 absol\_pers\_sc+  
 colony+   
 (1|year)+  
 (1|pair\_ID)+  
 (1|male\_ring)+  
 (1|female\_ring),   
 data = repro\_pers, na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))  
  
summary(repro\_pers\_mod)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: chick\_sur ~ female\_pers\_sc + male\_pers\_sc + absol\_pers\_sc + colony +   
## (1 | year) + (1 | pair\_ID) + (1 | male\_ring) + (1 | female\_ring)  
## Data: repro\_pers  
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))  
##   
## AIC BIC logLik deviance df.resid   
## 234.4 264.6 -108.2 216.4 204   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.6234 -0.5191 -0.1210 0.5091 4.7886   
##   
## Random effects:  
## Groups Name Variance Std.Dev.   
## pair\_ID (Intercept) 9.400e-16 3.066e-08  
## female\_ring (Intercept) 0.000e+00 0.000e+00  
## male\_ring (Intercept) 2.532e-01 5.032e-01  
## year (Intercept) 4.958e+00 2.227e+00  
## Number of obs: 213, groups:   
## pair\_ID, 124; female\_ring, 107; male\_ring, 104; year, 12  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.34760 0.91338 -1.475 0.140  
## female\_pers\_sc 0.22849 0.21869 1.045 0.296  
## male\_pers\_sc 0.03465 0.21385 0.162 0.871  
## absol\_pers\_sc -0.28292 0.19802 -1.429 0.153  
## colonypyramiden 1.92843 1.61945 1.191 0.234  
##   
## Correlation of Fixed Effects:  
## (Intr) fml\_p\_ ml\_pr\_ absl\_\_  
## feml\_prs\_sc -0.158   
## male\_prs\_sc -0.180 0.144   
## absl\_prs\_sc 0.001 0.325 0.155   
## colnypyrmdn -0.586 0.146 0.168 0.009  
## optimizer (bobyqa) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

r.squaredGLMM(repro\_pers\_mod)

## R2m R2c  
## theoretical 0.09189800 0.6485520  
## delta 0.08465468 0.5974337

We then employ model selection. Three models are retained. The first highlights absolute difference in boldness score, the second highlights female boldness score and the third is the null model.

*HOWEVER as the second model (highlighting female boldness) has an AICc equal to the null model, it can be labelled as nested and so we do not take it into consideration.*

sur\_mol\_ab<-dredge(repro\_pers\_mod, extra = alist(deviance)) # create all possible models  
sur\_mol\_absb<-subset(sur\_mol\_ab, delta<2) ##subsets models delta <2  
nest\_repro\_pers <- subset(sur\_mol\_absb, !nested(.)) # remove nested models  
  
#An additional model featuring female boldness is retained alongside the model featuring only absolute difference and the null model.  
#This model has an AICc equal to that of the null model and so was ultimately exclude, however it continues to appear  
#in the results produced by this code.   
  
repro\_pers\_coeffs<-coefTable(nest\_repro\_pers, full=TRUE) # create final coefficient tables  
repro\_pers\_coeffs

## $`2`  
## Estimate Std. Error df  
## (Intercept) -0.69535 0.77765 207  
## absol\_pers\_sc -0.34684 0.18480 207  
##   
## $`5`  
## Estimate Std. Error df  
## (Intercept) -0.74120 0.78798 207  
## female\_pers\_sc 0.30398 0.20405 207  
##   
## $`1`  
## Estimate Std. Error df  
## (Intercept) -0.65866 0.77345 208

Simplified models (based on the model selection criteria) are built to generate the data needed for visualisation.

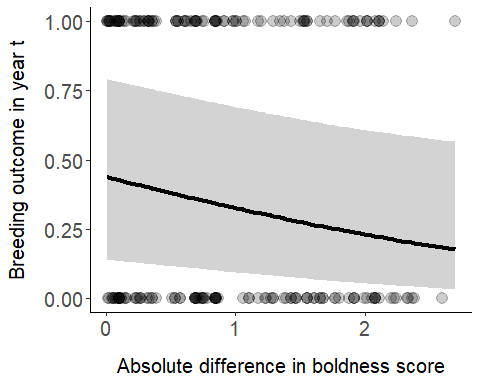
repro\_pers\_mod\_ab<- glmer(chick\_sur ~ #female\_pers\_sc+  
 #male\_pers\_sc+  
 absol+ #changed to original variable so that the graph axis is not scaled i.e positive values only  
 #colony+   
 (1|year)+  
 (1|pair\_ID)+  
 (1|male\_ring)+  
 (1|female\_ring),   
 data = repro\_pers, na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))

To create the prediction graphs (Figure 3) the relevant data are extracted using sjPlot and used in tandem with the original data to show the trends.

repro\_pers$chick\_sur\_num <- as.numeric(repro\_pers$chick\_sur) - 1 #creates scale for chick survival probability  
  
pers\_chick\_sur<-plot\_model(repro\_pers\_mod\_ab, type = "pred", terms=c ("absol[all]"))  
repro\_pers\_plot1<-as.data.frame(pers\_chick\_sur[["data"]]) #data turned into data frame  
repro\_pers\_plot1$absol<-repro\_pers\_plot1$x # variables are renamed so they can be used in the same graph   
repro\_pers\_plot1$chick\_sur<-repro\_pers\_plot1$predicted

The model data and the original data and then combined to make the plot.

repro\_pers\_plot1<-ggplot()+ geom\_point(data= repro\_pers, aes(x = absol, y = chick\_sur\_num), alpha = 0.2, size = 4)+  
 geom\_ribbon(data=repro\_pers\_plot1, aes(x = absol, ymin = conf.low, ymax = conf.high), fill = "lightgrey")+  
 geom\_line(data=repro\_pers\_plot1, aes(x = absol, y = chick\_sur), size = 1.5)+  
 theme\_classic()+  
 xlab("Absolute difference in boldness score")+  
 ylab("Breeding outcome in year t")+  
theme(axis.text = element\_text(size = 15))+  
theme(axis.title= element\_text(size=15))+  
theme (axis.title.y = element\_text (margin = margin (t = 0, r = 13, b = 0, l = 0)))+  
theme (axis.title.x = element\_text (margin = margin (t = 13, r = 0, b = 0, l = 0)))  
  
repro\_pers\_plot1



## M3: breeding outcome\_re-pairing outcome- Reproductive outcome and re-pairing probability

In this model the influence of reproductive success in year t on re-pairing likelihood in year t + 1 is evaluated.To begin, the data are filtered so that any missing data from the variables required for this model are removed.

#M3: breeding outcome\_re-pairing outcome- Reproductive outcome and re-pairing probability   
  
#exclusions  
suc\_rep<-all\_kitt[!is.na(all\_kitt$chick\_sur),]# remove nests with no breeding success info  
suc\_rep<-suc\_rep[!is.na(suc\_rep$season\_outcome),] # remove nests with no next season repairing outcome info

Exploring the data gives us details of the sample sizes included in the model.

unique(suc\_rep$pair\_ID)#139 pairs  
unique(suc\_rep$cycle\_pair)#218 breeding attempts  
  
unique(suc\_rep$male\_ring)#108 male birds  
unique(suc\_rep$female\_ring)#111 female birds  
  
unique(suc\_rep$year) #11 breeding seasons  
with(suc\_rep, tapply(pair\_ID, year, FUN = function(x) length(unique(x)))) # sample size per year

We then build the global model. Here, chick\_sur represents the binary breeding outcome for year t. We also generate R squared values for the model.

repro\_repair <- glmer(season\_outcome ~ chick\_sur+  
 colony+   
 (1|year)+  
 (1|pair\_ID)+  
 (1|male\_ring)+  
 (1|female\_ring),   
 data = suc\_rep, na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))  
summary(repro\_repair)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: season\_outcome ~ chick\_sur + colony + (1 | year) + (1 | pair\_ID) +   
## (1 | male\_ring) + (1 | female\_ring)  
## Data: suc\_rep  
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))  
##   
## AIC BIC logLik deviance df.resid   
## 272.8 296.4 -129.4 258.8 211   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.1365 -0.5716 -0.4471 0.8315 1.9832   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## pair\_ID (Intercept) 0.0000 0.0000   
## female\_ring (Intercept) 0.6969 0.8348   
## male\_ring (Intercept) 0.0000 0.0000   
## year (Intercept) 0.4021 0.6341   
## Number of obs: 218, groups:   
## pair\_ID, 139; female\_ring, 111; male\_ring, 108; year, 11  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.1152 0.3885 -0.297 0.76683   
## chick\_sur1 -1.1361 0.4341 -2.617 0.00887 \*\*  
## colonypyramiden -0.1745 0.6507 -0.268 0.78854   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) chck\_1  
## chick\_sur1 -0.552   
## colnypyrmdn -0.387 -0.057  
## optimizer (bobyqa) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

r.squaredGLMM(repro\_repair)

## R2m R2c  
## theoretical 0.07134995 0.3038784  
## delta 0.05746076 0.2447245

We then employ model selection.

repair\_repro<-dredge(repro\_repair, extra = alist(deviance)) # create all possible models  
rep\_repro\_sb<-subset(repair\_repro, delta<2) ##subsets models delta <2  
repair\_repro\_coeffs<-coefTable(rep\_repro\_sb, full=TRUE) # final coefficient table   
repair\_repro\_coeffs

## $`2`  
## Estimate Std. Error df  
## (Intercept) -0.15332 0.35910 212  
## chick\_sur1 -1.14650 0.43176 212

The best fitting model retains only chick survival as an influence on re-pairing probability.

We build a simplified model which reflect the results of our model selection criteria.This will be used for visualisation.

repro\_repair\_g <- glmer(season\_outcome ~ chick\_sur+  
 (1|year)+  
 (1|pair\_ID)+  
 (1|male\_ring)+  
 (1|female\_ring),   
 data = suc\_rep, na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))

We use sjPlot to extract the data used in the graph (Figure 4).

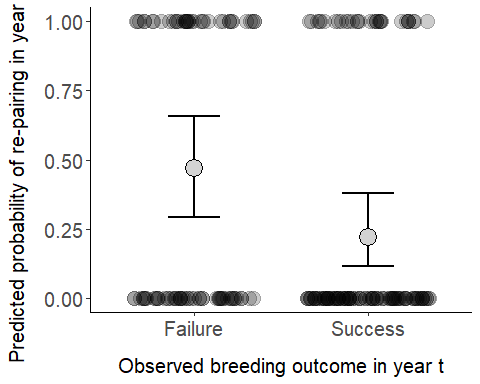
rep\_repair\_chick\_sur<-plot\_model(repro\_repair, type = "pred", terms=c ("chick\_sur"))  
repro\_plot2<-as.data.frame(rep\_repair\_chick\_sur[["data"]])  
repro\_plot2$season\_outcome2<-repro\_plot2$predicted # variables are renamed so they can be used in the same graph

Variables are renamed to make the final graph clearer.A new scale is created for the response variable so it is clear on the final graph.

for (a in 1:(nrow(repro\_plot2))){  
   
 if (repro\_plot2$x[a] == "0")   
   
 {repro\_plot2$chick\_sur2[a]<-"Failure"}   
 else {repro\_plot2$chick\_sur2[a]<-"Success"}  
}  
  
for (a in 1:(nrow(suc\_rep))){  
   
 if (suc\_rep$chick\_sur[a] == "0")   
   
 {suc\_rep$chick\_sur2[a]<-"Failure"}   
 else {suc\_rep$chick\_sur2[a]<-"Success"}  
}  
  
#new scale created for season outcome to make graph more logical  
suc\_rep$season\_outcome2<- as.numeric(suc\_rep$season\_outcome) - 1

The final graph is produced.

repro\_plotg2<-ggplot()+   
 geom\_jitter(data = suc\_rep, aes(x =chick\_sur2, y = season\_outcome2), width =0.35, height = 0.0, alpha = 0.2, size = 5)+   
 geom\_errorbar(data= repro\_plot2, aes(x = chick\_sur2, ymin = conf.low, ymax = conf.high), width = 0.3, size = 0.8)+  
 geom\_point(data = repro\_plot2, aes( x = chick\_sur2, y = season\_outcome2), colour ="black", fill = "lightgrey", pch=21, size = 6)+  
 theme\_classic()+  
 xlab("Observed breeding outcome in year t")+  
 ylab("Predicted probability of re-pairing in year t + 1")+  
 theme(axis.text = element\_text(size = 15))+  
 theme(axis.title= element\_text(size=15))+  
 theme (axis.title.y = element\_text (margin = margin (t = 0, r = 13, b = 0, l = 0)))+  
 theme (axis.title.x = element\_text (margin = margin (t = 13, r = 0, b = 0, l = 0)))  
  
repro\_plotg2



## M4: re-pairing outcome\_breeding outcome- Re-pairing and future reproductive outcomes

In this model the influence of re-pairing outcome on reproductive success in year t + 1 is evaluated.To begin, the data are filtered so that any missing data from the variables required for this model are removed. the categorical variables are turned into factors.

#M4: re-pairing outcome\_breeding outcome- Re-pairing and future reproductive outcomes  
  
ns\_repro<-next\_season\_repro[!is.na(next\_season\_repro$season\_outcome),]  
ns\_repro<-ns\_repro[!is.na(ns\_repro$next\_season),]  
  
ns\_repro$colony<-as.factor(ns\_repro$colony)  
ns\_repro$sex<-as.factor(ns\_repro$sex)  
ns\_repro$ring<-as.factor(ns\_repro$ring)  
ns\_repro$pair\_ID<-as.factor(ns\_repro$pair\_ID)  
ns\_repro$year\_t<-as.factor(ns\_repro$year\_t)  
ns\_repro$season\_outcome<-as.factor(ns\_repro$season\_outcome)  
ns\_repro$next\_season<-as.factor(ns\_repro$next\_season)  
ns\_repro$cycle\_pair<-as.factor(ns\_repro$cycle\_pair)

The sample sizes may be determined using the below code.

unique(ns\_repro$cycle\_pair) #175 breeding attempts  
unique(ns\_repro$pair\_ID) # 107 pairs  
with(ns\_repro, tapply(cycle\_pair, year\_t, FUN = function(x) length(unique(x)))) # sample per year  
with(ns\_repro, tapply(ring, sex, FUN = function(x) length(unique(x)))) # 75 females, 70 males

The global M4: re-pairing outcome\_ breeding outcome is built. In the this model next\_season is the breeding success in year t + 1 and season\_outcome is the re-pairing outcome at the start of year t + 1.

next\_season\_mol <- glmer(next\_season ~ colony+  
 season\_outcome+  
 season\_outcome:sex+  
 sex+  
 (1|year\_t)+  
 (1|pair\_ID)+  
 (1|ring),   
 data = ns\_repro,na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))  
summary(next\_season\_mol)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: next\_season ~ colony + season\_outcome + season\_outcome:sex +   
## sex + (1 | year\_t) + (1 | pair\_ID) + (1 | ring)  
## Data: ns\_repro  
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))  
##   
## AIC BIC logLik deviance df.resid   
## 240.6 270.4 -112.3 224.6 296   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.71578 -0.02215 -0.00389 0.22051 2.49521   
##   
## Random effects:  
## Groups Name Variance Std.Dev.   
## ring (Intercept) 2.552e-16 1.597e-08  
## pair\_ID (Intercept) 1.209e+01 3.477e+00  
## year\_t (Intercept) 1.477e+02 1.215e+01  
## Number of obs: 304, groups: ring, 145; pair\_ID, 107; year\_t, 10  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.339e+00 4.192e+00 -2.228 0.0259 \*  
## colonypyramiden 8.204e+00 1.286e+01 0.638 0.5236   
## season\_outcome1 -1.439e+00 1.591e+00 -0.904 0.3658   
## sexM -2.009e-07 5.297e-01 0.000 1.0000   
## season\_outcome1:sexM 1.015e+00 1.816e+00 0.559 0.5761   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) clnypy ssn\_t1 sexM   
## colnypyrmdn -0.319   
## season\_tcm1 0.064 -0.028   
## sexM -0.063 0.000 0.166   
## ssn\_tcm1:sM -0.057 0.020 -0.734 -0.292  
## optimizer (bobyqa) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

r.squaredGLMM(next\_season\_mol)

## R2m R2c  
## theoretical 6.500283e-02 9.811348e-01  
## delta 3.175692e-06 4.793301e-05

Model selection takes place as before.

next\_season\_dre<-dredge(next\_season\_mol, extra = alist(deviance))  
next\_season\_dre2<-subset(next\_season\_dre, delta<2) ##subsets models delta <2  
next\_season\_dre3<- subset(next\_season\_dre2, !nested(.))# remove nested models  
next\_season\_dre\_coeff<-coefTable(next\_season\_dre3, full=TRUE)  
next\_season\_dre\_coeff

## $`1`  
## Estimate Std. Error df  
## (Intercept) -3.7944 3.2986 300

The null model is retained as the best fitting model. This suggests there is little evidence to suggest re-pairing at the start of year t impacts breeding success in that year.

## M5: Boldness\_re-pairing outcome- boldness and re-pairing probability

In this model the influence of pair members’ boldness and absolute difference in boldness within pairs on re-pairing probability is evaluated. To begin, the data are filtered so that any missing data from the variables required for this model are removed. M5: boldness\_re-pairing outcome is independent of breeding success in year t and thus the variable ulti\_outcome can be used (i.e. did the pair dissolve at any point in the data collection period? Y/N). The missed\_seasons variable accounts for the extra opportunities these birds had to re-pair in between sightings (see paper). Boldness metrics are defined as before.

pers\_repair<-all\_kitt[!is.na(all\_kitt$ulti\_outcome),]# where ultimate outcome of the pair is unknown- exclude  
pers\_repair<-pers\_repair[!is.na(pers\_repair$absol),]# exclude pairs where both personalities are not available  
  
#scaling  
  
pers\_repair$male\_pers\_sc<-scale(pers\_repair$male\_pers)   
pers\_repair$female\_pers\_sc<-scale(pers\_repair$female\_pers)   
pers\_repair$absol\_pers\_sc<-scale(pers\_repair$absol)

Descriptive reveal similar patterns to those reported under M2: boldness\_breeding outcome. Males are bolder than females and birds are bolder at Grumant than Pyramiden.

pers\_repair%>%summarise(average=mean(male\_pers),med=median(male\_pers), #males slightly bolder than females overall  
 sd=sd(male\_pers))

## average med sd  
## 1 0.980684 1.217355 0.8817706

pers\_repair%>%summarise(average=mean(female\_pers),med=median(female\_pers),   
 sd=sd(female\_pers))

## average med sd  
## 1 0.8573766 0.9132636 0.9095447

pers\_repair%>%group\_by(colony)%>%summarise(average=mean(male\_pers),med=median(male\_pers), #both males and females bolder at grumant  
 sd=sd(male\_pers))

## # A tibble: 2 × 4  
## colony average med sd  
## <fct> <dbl> <dbl> <dbl>  
## 1 grumant 1.57 1.82 0.633  
## 2 pyramiden 0.619 0.870 0.816

pers\_repair%>%group\_by(colony)%>%summarise(average=mean(female\_pers),med=median(female\_pers),   
 sd=sd(female\_pers))

## # A tibble: 2 × 4  
## colony average med sd  
## <fct> <dbl> <dbl> <dbl>  
## 1 grumant 1.30 1.66 0.887  
## 2 pyramiden 0.586 0.722 0.815

Sample sizes can be determined using the below code.

#sample sizes  
  
unique(pers\_repair$pair\_ID)#98 pairs  
unique(pers\_repair$cycle\_pair)#166 breeding attempts  
  
unique(pers\_repair$male\_ring)#81 male birds  
unique(pers\_repair$female\_ring)#87 female birds  
  
unique(pers\_repair$year) #11 breeding seasons  
with(pers\_repair, tapply(pair\_ID, year, FUN = function(x) length(unique(x))))# sample size per year  
with(pers\_repair, tapply(pair\_ID, colony, FUN = function(x) length(unique(x))))# sample size per colony

We test for colinearity between continuous variables. And VIF scores > 5 are problematic. There are no problems highlighted here.

#assumption multicolinearity  
  
colin\_model11 <- glm(ulti\_outcome ~ male\_pers\_sc +female\_pers\_sc, data = pers\_repair, family = "binomial")  
summary(colin\_model11)

##   
## Call:  
## glm(formula = ulti\_outcome ~ male\_pers\_sc + female\_pers\_sc, family = "binomial",   
## data = pers\_repair)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0321 -0.8007 -0.7209 1.3394 1.8515   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.043802 0.179538 -5.814 6.11e-09 \*\*\*  
## male\_pers\_sc -0.324950 0.175280 -1.854 0.0638 .   
## female\_pers\_sc -0.008388 0.182204 -0.046 0.9633   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 191.99 on 165 degrees of freedom  
## Residual deviance: 188.47 on 163 degrees of freedom  
## AIC: 194.47  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model11)

## male\_pers\_sc female\_pers\_sc   
## 1.01514 1.01514

colin\_model22 <- glm(ulti\_outcome ~ male\_pers\_sc +absol\_pers\_sc, data = pers\_repair, family = "binomial")  
summary(colin\_model22)

##   
## Call:  
## glm(formula = ulti\_outcome ~ male\_pers\_sc + absol\_pers\_sc, family = "binomial",   
## data = pers\_repair)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0591 -0.7817 -0.7210 1.3004 1.8678   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.04439 0.17962 -5.815 6.08e-09 \*\*\*  
## male\_pers\_sc -0.31331 0.17668 -1.773 0.0762 .   
## absol\_pers\_sc 0.06792 0.18065 0.376 0.7069   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 191.99 on 165 degrees of freedom  
## Residual deviance: 188.33 on 163 degrees of freedom  
## AIC: 194.33  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model22)

## male\_pers\_sc absol\_pers\_sc   
## 1.035276 1.035276

colin\_model33 <- glm(ulti\_outcome ~ female\_pers\_sc +absol\_pers\_sc, data = pers\_repair, family = "binomial")  
summary(colin\_model33)

##   
## Call:  
## glm(formula = ulti\_outcome ~ female\_pers\_sc + absol\_pers\_sc,   
## family = "binomial", data = pers\_repair)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.8622 -0.7986 -0.7502 1.5262 1.7040   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.02337 0.17640 -5.801 6.58e-09 \*\*\*  
## female\_pers\_sc -0.00545 0.18920 -0.029 0.977   
## absol\_pers\_sc 0.12134 0.18889 0.642 0.521   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 191.99 on 165 degrees of freedom  
## Residual deviance: 191.49 on 163 degrees of freedom  
## AIC: 197.49  
##   
## Number of Fisher Scoring iterations: 4

vif(colin\_model33)

## female\_pers\_sc absol\_pers\_sc   
## 1.16543 1.16543

The global M5: boldness\_re-pairing outcome is built.

repair\_mol\_abs <- glmer(ulti\_outcome ~ female\_pers\_sc+  
 male\_pers\_sc+  
 absol\_pers\_sc+  
 colony+  
 missed\_seasons+ # to account for pairs which had multiple opportunities to re-pair without being recorded  
 (1|year)+  
 (1|pair\_ID)+  
 (1|male\_ring)+  
 (1|female\_ring),   
 data = pers\_repair,na.action=na.fail, family = binomial(link = "logit"),  
 control=glmerControl(optimizer="bobyqa",  
 optCtrl=list(maxfun=2e5)))  
  
summary(repair\_mol\_abs)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: ulti\_outcome ~ female\_pers\_sc + male\_pers\_sc + absol\_pers\_sc +   
## colony + missed\_seasons + (1 | year) + (1 | pair\_ID) + (1 |   
## male\_ring) + (1 | female\_ring)  
## Data: pers\_repair  
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))  
##   
## AIC BIC logLik deviance df.resid   
## 194.8 225.9 -87.4 174.8 156   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.1849 -0.5330 -0.3686 0.5186 2.5758   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## pair\_ID (Intercept) 0.01039 0.1019   
## female\_ring (Intercept) 0.77634 0.8811   
## male\_ring (Intercept) 0.00000 0.0000   
## year (Intercept) 0.21665 0.4655   
## Number of obs: 166, groups:   
## pair\_ID, 98; female\_ring, 87; male\_ring, 81; year, 11  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.20626 0.59940 -3.681 0.000233 \*\*\*  
## female\_pers\_sc 0.26074 0.27637 0.943 0.345446   
## male\_pers\_sc -0.08076 0.26370 -0.306 0.759405   
## absol\_pers\_sc 0.14422 0.25155 0.573 0.566412   
## colonypyramiden 0.94049 0.73307 1.283 0.199511   
## missed\_seasons 0.88243 0.44094 2.001 0.045364 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) fml\_p\_ ml\_pr\_ absl\_\_ clnypy  
## feml\_prs\_sc -0.290   
## male\_prs\_sc -0.237 0.185   
## absl\_prs\_sc -0.108 0.376 0.257   
## colnypyrmdn -0.742 0.311 0.371 0.111   
## missed\_ssns -0.261 0.122 -0.017 0.101 -0.145  
## optimizer (bobyqa) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

r.squaredGLMM(repair\_mol\_abs)

## R2m R2c  
## theoretical 0.13540832 0.3374718  
## delta 0.09336083 0.2326789

Model selection occurs as before.

Only missed\_seasons is retained and therefore there is little veidnece that the boldness metrics have an impact on re-pairing outcome.

## Permutation tests

These permutation tests assess if re-pairing birds choose a new partner which is more similar to themselves compared to their old partner. First, to determine the observed or ‘true’ difference in boldness scores between new partnerships and original partnerships, the absolute difference in boldness within each new partnership was subtracted from the absolute difference in boldness within the original partnership. A positive value indicated that the new partner was more similar in boldness to the focal bird than its original partner.

absol\_ran$trueDiff <- absol\_ran$orig\_absol - absol\_ran$new\_absol

The colonies are divided to account for the different personality distributions of potential mates at the two sites.

#colonies separately   
absol\_ran\_g<-subset(absol\_ran, colony.x == "grumant")  
absol\_ran\_p<-subset(absol\_ran, colony.x == "pyramiden")  
  
pers\_colony\_g<-subset(pers\_colony, colony == "Grumant")  
pers\_colony\_p<-subset(pers\_colony, colony == "Pyramiden")

The loop parameters are set. This includes setting the number of permutations, creating new vectors and calculating the mean true difference.

#randomise personality of new partner across the whole population  
## sampling randomly from the true personality distribution  
  
# grumant  
set.seed(100)# to ensure randomisation results are repeatable  
n\_perms <- 10000 # typical minimum for randomisation/permutation  
perms\_output <- list()  
mean\_trueDiff <- mean(absol\_ran\_g$trueDiff) # this is the observed mean difference  
pers\_cg<-pers\_colony\_g$pers\_scores

The loop is built. Boldness differences are calculated within randomised pairings. Within each permutation, every focal bird in the population was paired with a new partner boldness score, which is randomly sampled from the distribution of boldness scores available within each population. The absolute difference between the focal birds’ boldness and the randomly sampled personalities is calculated and subtracted from the original within pair difference in boldness. This value (within original pair difference – within random pair difference) is then averaged across the population to give a randomised mean difference (difference between the original and randomised pair’s absolute difference). This process is repeated 10,000 times per colony. The code set.seed(100) is used to ensure that the randomsiations return the same result every time for the purposes of repeatability.

for (i in 1:n\_perms ) {  
   
 # Randomise new partner personalities  
 perm\_dat <- data.frame(F\_pers = absol\_ran\_g$F\_pers,   
 orig\_absol = absol\_ran\_g$orig\_absol,  
 shuffled\_part\_pers = sample(pers\_cg, nrow(absol\_ran\_g)))   
   
 # Calculate differences  
 perm\_dat$shuffled\_absol <- abs(perm\_dat$F\_pers - perm\_dat$shuffled\_part\_pers)  
 perm\_dat$shuffled\_orig\_diff <- perm\_dat$orig\_absol - perm\_dat$shuffled\_absol  
 perms\_output[[i]] <- mean(perm\_dat$shuffled\_orig\_diff)  
   
}

By calculating the proportion of cases where randomised personality values are equal or closer to focal birds’ personality than true partners’ personality (i.e. the null hypothesis)

# Null hypothesis: randomised personality values are equal or closer to focal birds'  
# personality than true partners' personality  
  
# Calculate proportion of cases that meet the null hypothesis  
length(perms\_output[perms\_output >= mean\_trueDiff])/length(perms\_output) # less than #0.05

## [1] 0.0308

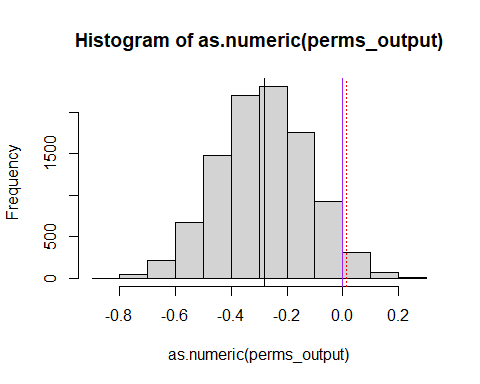
# If p < 0.05,   
# we can infer that focal birds are choosing partners who are closer to their  
# personality than we would expect by chance

These results can then be visualised and edited outside r for clarity (Figure 5).

# visualise data  
hist(as.numeric(perms\_output)) # lots of negative differences = original partner is closer to Focal than randomised  
abline(v = mean\_trueDiff, col = "red", lty = "dotted") # represents observed mean difference or new partners  
abline(v = 0, col = "purple") # represents original mean difference or old partners   
  
mean(as.numeric(perms\_output))

## [1] -0.2849031

abline(v = -0.28, col = "black") # observed mean difference which would be expected if choosing partners at random



This result suggests that birds at Grumant choose new partners that were only slightly more similar to themselves than their previous mates, but that this was still significantly different from random, because the birds were mated assortatively to begin with.

This process is then repeated using data for the Pyramiden birds.

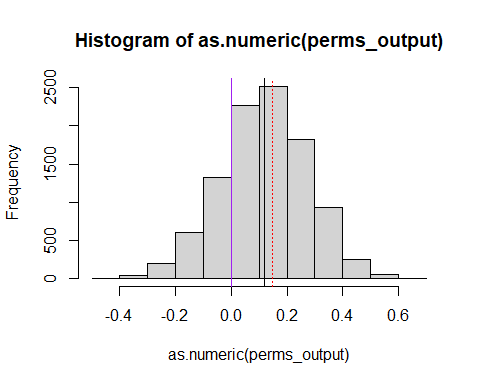
#pyramiden  
set.seed(100) # to ensure randomisation results are repeatable  
n\_perms <- 10000 # typical minimum for randomisation/permutation  
perms\_output <- list()  
mean\_trueDiff <- mean(absol\_ran\_p$trueDiff) # this is your observed mean difference  
pers\_cp<-pers\_colony\_p$pers\_scores  
  
for (i in 1:n\_perms ) {  
   
 # Randomise new partner personalities  
 perm\_dat <- data.frame(F\_pers = absol\_ran\_p$F\_pers,   
 orig\_absol = absol\_ran\_p$orig\_absol,  
 shuffled\_part\_pers = sample(pers\_cp, nrow(absol\_ran\_p)))   
 # x is the simulated personality distribution - you should sample  
 # from the real personality distribution.  
   
 # Calculate differences  
 perm\_dat$shuffled\_absol <- abs(perm\_dat$F\_pers - perm\_dat$shuffled\_part\_pers)  
 perm\_dat$shuffled\_orig\_diff <- perm\_dat$orig\_absol - perm\_dat$shuffled\_absol  
 perms\_output[[i]] <- mean(perm\_dat$shuffled\_orig\_diff)  
   
}  
  
# Null hypothesis: randomised personality values are equal or closer to focal birds'  
# personality than true partners' personality  
  
# Calculate proportion of cases that meet the null hypothesis  
length(perms\_output[perms\_output >= mean\_trueDiff])/length(perms\_output) # greater than 0.05

## [1] 0.4312

# visualise data  
hist(as.numeric(perms\_output)) # lots of negative differences = original partner is closer to Focal than randomised  
abline(v = mean\_trueDiff, col = "red", lty = "dotted") # represents observed mean difference or new partners  
abline(v = 0, col = "purple") # represents original mean difference or old partners  
  
mean(as.numeric(perms\_output))

## [1] 0.1196162

abline(v = 0.12, col = "black") # observed mean difference which would be expected if choosing partners at random



This result suggests that birds at Pyramiden may have re-paired at random.