

Cofeeding at rich clumped food patches in free-ranging dogs: Social tolerance or scramble competition?

Behavioural ecology and sociobiology

Andreas Berghänel^{1#}, Martina Lazzaroni², Malgorzata Ferenc¹, Malgorzata Pilot³, Ikhlass el Berbri⁴, Sarah Marshall-Pescini^{1*}, Friederike Range^{1*}

1 Domestication Lab, Konrad Lorenz Institute of Ethology, University of Veterinary Medicine Vienna, Vienna, Austria

2 Department of Chemistry, Life Science and Environmental Sustainability, University of Parma, Italy

3 Faculty of Biology, University of Gdańsk, Gdańsk, Poland

4 Department of Pathology and Veterinary Public Health, Agronomic and Veterinary Institute Hassan II, Rabat, Morocco

* Authors contributed equally

Corresponding author: Andreas.Berghaenel@vetmeduni.ac.at

R-Code

Model 1: Pattern of cofeeding

```
#
library(tidyverse)
library(mgcv)
library(itsadug)
library(lmerTest)
library(ggplot2)
library(DHARMA)
library(sjPlot)
xdata = read.table("clipboard", dec=".", header=T, na.strings=c("x", "xxx"))
xdata$rankF=as.factor(xdata$rankpresent)
xdata$dateF=as.factor(xdata$date)
attach(xdata, warn.conflicts = FALSE)
ndata=xdata %>% filter(activity=="ff")
boxplot(ndata$NrCofeeders~ ndata$rankF)
boxplot(ndata$NrLowerCofeeders~ ndata$rankF)
```

0.8m cofeeding distance:

```
ggplot(ndata, aes(y= NrCofeeders, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() + theme(axis.text.y=element_text(angle=90,
hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7, shape=21, stroke = 1.5, fill="grey50") +
theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) + theme(legend.title = element_text(size = 0)) #+
scale_y_continuous(lim=c(0,7))
ggplot(ndata, aes(y= NrLowerCofeeders, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() + theme(axis.text.y=element_text(angle=90,
hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7, shape=21, stroke = 1.5, fill="grey50") +
theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) + theme(legend.title = element_text(size = 0)) #+
scale_y_continuous(lim=c(0,4))
ggplot(ndata, aes(y= NrLowerCofeedersUnrelated, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) #+ scale_y_continuous(lim=c(0,4))
ggplot(ndata, aes(y= NrLowerCofeedersRelated, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) #+ scale_y_continuous(lim=c(0,4))
```

Test whether Nr of Lower-ranking Cofeeders is significantly different from zero (0) (significant in all 3 cases) (Prediction under Priority of Access model = complete monopolization within defensible range of 0.8m = 1BL: Nr of Lower-ranking Cofeeders should always be =0):

#Across all individuals

```
full <- glmer(NrLowerCofeeders ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=ndata)#, control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e10))) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(NrLowerCofeeders ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=ndata)#, control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e10)))
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

#For the Alpha only:

```
mdata=ndata %>% filter(rankpresent<=1)
```

```
full <- glmer(NrLowerCofeeders ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=mdata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(NrLowerCofeeders ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=mdata)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

Test whether Nr of Lower-ranking unrelated Cofeeders is significantly different from zero (0) (significant in all 3 cases) (Prediction under Priority of Access model = complete monopolization within defensible range of 0.8m = 1BL: Nr of Lower-ranking Cofeeders should always be =0):

#Across all individuals

```
full <- glmer(NrLowerCofeedersUnrelated ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=ndata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(NrLowerCofeedersUnrelated ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=ndata)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

#For the Alpha only:

```
mdata=ndata %>% filter(rankpresent<=1)
```

```
full <- glmer(NrLowerCofeedersUnrelated ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=mdata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(NrLowerCofeedersUnrelated ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=mdata)
```

```
testDispersion(full)
testZeroInflation(full)
anova(null, full, test="Chisq")
```

```
#####
```

0.4m cofeeding distance:

```
ggplot(ndata, aes(y= X0.4_NrCofeeders, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() + theme(axis.text.y=element_text(angle=90,
hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7, shape=21, stroke = 1.5, fill="grey50") +
theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) + theme(legend.title = element_text(size = 0)) #+
scale_y_continuous(lim=c(0,7))
ggplot(ndata, aes(y= X0.4_NrLowerCofeeders, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) #+ scale_y_continuous(lim=c(0,4))
ggplot(ndata, aes(y= X0.4_NrLowerCofeedersUnrelated, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) #+ scale_y_continuous(lim=c(0,4))
ggplot(ndata, aes(y= X0.4_NrLowerCofeedersRelated, x=rankF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.24, position=position_jitter(height=.1, width=.3), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) #+ scale_y_continuous(lim=c(0,4))
```

Test whether Nr of Lower-ranking Cofeeders is significantly different from zero (0) (significant in all 3 cases) (Prediction under Priority of Access model = complete monopolization within defendable range of 0.4m = 0.5BL: Nr of Lower-ranking Cofeeders should always be =0):

#Across all individuals

```
full <- glmer(X0.4_NrLowerCofeeders ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=ndata)#, control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e10))) #model
without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(X0.4_NrLowerCofeeders ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=ndata)#, control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e10)))
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

#For the Alpha only:

```
mdata=ndata %>% filter(rankpresent<=1)
```

```
full <- glmer(X0.4_NrLowerCofeeders ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=mdata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(X0.4_NrLowerCofeeders ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=mdata)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

Test whether Nr of Lower-ranking unrelated Cofeeders is significantly different from zero (0) (significant in all 3 cases) (Prediction under Priority of Access model = complete monopolization within defendable range of 0.4m = 0.5BL: Nr of Lower-ranking Cofeeders should always be =0):

#Across all individuals

```
full <- glmer(X0.4_NrLowerCofeedersUnrelated ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=ndata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(X0.4_NrLowerCofeedersUnrelated ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=ndata)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

#For the Alpha only:

```
mdata=ndata %>% filter(rankpresent<=1)
```

```
full <- glmer(X0.4_NrLowerCofeedersUnrelated ~ 1 + (1|dateF) + (1|individual), family = "poisson", data=mdata) #model without fixed effects except for the intercept
```

```
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. #
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- glmer(X0.4_NrLowerCofeedersUnrelated ~ 0 + (1|dateF) + (1|individual), family = "poisson", data=mdata)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq")
```

PLOT: Rank distribution mainpatch depending on feeding groups size

```
xdata$feeding_groupsize_mainpatchF=as.factor(xdata$feeding_groupsize_mainpatch)
ndata=xdata %>% filter(distance_to_mainfeedpatch<=0.1)
cor.test(ndata$feeding_groupsize_mainpatch,ndata$rankpresent)
plot(jitter(ndata$feeding_groupsize_mainpatch,0.2),jitter(ndata$rankpresent,0.2))
abline(lm(feeding_groupsize_mainpatch~rankpresent,data=ndata))
plot(jitter(ndata$feeding_groupsize_mainpatch,0.5),jitter(ndata$rankpresent,0.5), xlab="Feeding group size at main patch", ylab="Individual ranks present at main patch*", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n")
#lines(lowess(ndata$feeding_groupsize_mainpatch,ndata$rankpresent))
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
axis(2, at=c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
ggplot(ndata, aes(x=feeding_groupsize_mainpatchF, y=rankpresent)) + geom_violin() + theme_classic() + theme(axis.text.y=element_text(angle=90, hjust=0.5)) +
theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.1, position=position_jitter(height=.0, width=.3))#+ scale_y_continuous(lim=c(-50,400))

n1data=ndata %>% filter(scan<=10)
cor.test(n1data$feeding_groupsize_mainpatch,n1data$rankpresent)
plot(jitter(n1data$feeding_groupsize_mainpatch,0.2),jitter(n1data$rankpresent,0.2))
abline(lm(feeding_groupsize_mainpatch~rankpresent,data=n1data))
plot(jitter(n1data$feeding_groupsize_mainpatch,0.5),jitter(n1data$rankpresent,0.5), xlab="Feeding group size at main patch", ylab="Individual ranks present at main patch*", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n")
#lines(lowess(n1data$feeding_groupsize_mainpatch,n1data$rankpresent))
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
axis(2, at=c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
ggplot(n1data, aes(x=feeding_groupsize_mainpatchF, y=rankpresent)) + geom_violin() + theme_classic() + theme(axis.text.y=element_text(angle=90, hjust=0.5)) +
theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha = 0.1, position=position_jitter(height=.0, width=.3))#+ scale_y_continuous(lim=c(-50,400))
```

Model 2: Compare waiting and feeding individuals

```
library(tidyverse)
library(mgcv)
library(itsadug)
library(lmerTest)
library(ggplot2)
library(DHARMA)
library(sjPlot)
#xdata = read.table("clipboard", dec=".",header=T, na.strings=c("x","xxx"))
xdata$individual=as.factor(xdata$individual)
xdata$dateF=as.factor(xdata$date)
attach(xdata ,warn.conflicts = FALSE)
#Comparing waiting individuals (=1) with feeding individuals (=0)
wtffdata=xdata %>% filter(!wt_ff=="x") %>% drop_na(CS$select_IndivAverageDiff) %>% drop_na(AggDoBefore) #Adding AggDoBefore is not significant but reduces N

# Response variable: "Feeding or waiting/ queueing"
res1=gam(wt_ff~ s(scan, bs="cr", k=3) + s(groupsize, bs="cr", k=3) + s(feeding_groupsize_mainpatch, bs="cr", k=3) + number_additional_foodpatches + s(DS, bs="cr", k=3) +
s(CS$select_IndivAverageDiff, bs="cr", k=3) + s(Familiarity_IndivAverageDiff, bs="cr", k=3) + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= wtffdata, family=binomial, method="REML")
null1=gam(wt_ff~ s(scan, bs="cr", k=3) + s(individual, bs="re")+ s(dateF, bs="re"), data= wtffdata, family=binomial, method="REML")
testDispersion(res1)
testZeroInflation(res1)
compareML(null1, res1)
summary(res1)

plot(wtffdata$groupsize, wtffdata$wt_ff,type="p",col="white", xlab="Overall groupsize", ylab="Likelihood of waiting instead of feeding", cex.lab=2.6, cex.axis=2,
xaxt="n",yaxt="n",ylim=c(0,1))
points(jitter(wtffdata$groupsize,0), jitter(wtffdata$wt_ff,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(wtffdata$N_GS_wtff))
plot_smooth(res1, view="groupsize", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)

plot(wtffdata$scan, wtffdata$wt_ff,type="p",col="white", xlab="Scan", ylab="Likelihood of waiting instead of feeding", cex.lab=2.6, cex.axis=2, yaxt="n",ylim=c(0,1))
points(jitter(wtffdata$scan,1), jitter(wtffdata$wt_ff,0.15),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res1, view="scan", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
```

```
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
plot(wtffdata$number_additional_foodpatches, wtffdata$wt_ff,type="p",col="white", xlab="Number additional food patches", ylab="Likelihood of waiting instead of feeding", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(wtffdata$number_additional_foodpatches,0), jitter(wtffdata$wt_ff,0),type="p",col= alpha("black", 1), pch=1, lwd=2,
cex=sqrt(wtffdata$N_number_additional_foodpatches_wtff))
plot_smooth(res1, view="number_additional_foodpatches", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
plot(wtffdata$Familiarity_IndivAverageDiff, wtffdata$wt_ff,type="p",col="white", xlab="Proportion of scans seen together (average)", ylab="Likelihood of waiting instead of feeding",
cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(wtffdata$Familiarity_IndivAverageDiff,0), jitter(wtffdata$wt_ff,0.1),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res1, view="Familiarity_IndivAverageDiff", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(0,50,100,150,200,250,300,350,400),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

#Single effects plots

```
res1=gam(wt_ff~ s(groupsize, bs="cr", k=3) + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= wtffdata, family=binomial, method="REML")
plot(wtffdata$groupsize, wtffdata$wt_ff,type="p",col="white", xlab="Overall groupsize", ylab="Likelihood of waiting instead of feeding", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(wtffdata$groupsize,0), jitter(wtffdata$wt_ff,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(wtffdata$N_GS_wtff))
plot_smooth(res1, view="groupsize", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
res1=gam(wt_ff~ number_additional_foodpatches + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= wtffdata, family=binomial, method="REML")
plot(wtffdata$number_additional_foodpatches, wtffdata$wt_ff,type="p",col="white", xlab="Number additional food patches", ylab="Likelihood of waiting instead of feeding", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(wtffdata$number_additional_foodpatches,0), jitter(wtffdata$wt_ff,0),type="p",col= alpha("black", 1), pch=1, lwd=2,
cex=sqrt(wtffdata$N_number_additional_foodpatches_wtff))
plot_smooth(res1, view="number_additional_foodpatches", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
res1=gam(wt_ff~ s(Familiarity_IndivAverageDiff, bs="cr", k=3) + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= wtffdata, family=binomial, method="REML")
```

```
plot(wtffdata$Familiarity_IndivAverageDiff, wtffdata$wt_ff,type="p",col="white", xlab="Proportion of scans seen together (average)", ylab="Likelihood of waiting instead of feeding",
cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(wtffdata$Familiarity_IndivAverageDiff,0), jitter(wtffdata$wt_ff,0.1),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res1, view="Familiarity_IndivAverageDiff", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(0,50,100,150,200,250,300,350,400),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
#####
```

Model 3: Compare mainpatch-feeding and other feeding individuals

```
#
library(tidyverse)
library(mgcv)
library(itsadug)
library(lmerTest)
library(ggplot2)
library(DHARMA)
library(sjPlot)

#xdata = read.table("clipboard", dec=".", header=T, na.strings=c("x", "xxx"))
xdata$dateF=as.factor(xdata$date)
xdata$individual=as.factor(xdata$individual)
attach(xdata, warn.conflicts = FALSE)

ffdata=xdata %>% filter(activity=="ff") %>% drop_na(CSIselect_IndivAverageDiff) %>% drop_na(AggDoBefore) #Adding AggDoBefore is not significant but reduces N
# Response variable: "Feeding or waiting/ queueing"
res2=gam(feeding_at_mainfeedpatchF~ s(scan, bs="cr", k=3) + s(groupsize, bs="cr", k=3) + s(feeding_groupsize_mainpatch, bs="cr", k=3) + number_additional_foodpatches + s(DS, bs="cr", k=3) + s(CSIselect_IndivAverageDiff, bs="cr", k=3) + s(Familiarity_IndivAverageDiff, bs="cr", k=3) + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= ffdata, family=binomial, method="REML")
null2=gam(feeding_at_mainfeedpatchF~ s(scan, bs="cr", k=3) + s(individual, bs="re")+ s(scan,dateF, bs="re"), data= ffdata, family=binomial, method="REML")
testDispersion(res2)
testZeroInflation(res2)
acf_resid(res2)
compareML(null2, res2)
summary(res2)

plot(ffdata$groupsize, ffdata$feeding_at_mainfeedpatchF, type="p", col="white", xlab="Overall groupsize", ylab="Likelihood of feeding at main patch", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
points(jitter(ffdata$groupsize,0), jitter(ffdata$feeding_at_mainfeedpatchF,0), type="p", col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_GS)*2)
plot_smooth(res2, view="groupsize", rm.ranef=TRUE, h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20), cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16), cex.axis=2)

plot(ffdata$feeding_groupsize_mainpatch, ffdata$feeding_at_mainfeedpatchF, type="p", col="white", xlab="Number individuals feeding at mainpatch", ylab="Likelihood of feeding at main patch", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n", ylim=c(0,1))
```

```

points(jitter(ffdata$feeding_groupsize_mainpatch,0), jitter(ffdata$feeding_at_mainfeedpatchF,0),type="p",col= alpha("black", 1), pch=1, lwd=2,
cex=sqrt(ffdata$N_feeding_groupsize_mainpatch))
plot_smooth(res2, view="feeding_groupsize_mainpatch", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)

```

```

plot(ffdata$scan, ffdata$feeding_at_mainfeedpatchF,type="p",col="white", xlab="Scan", ylab="Likelihood of feeding at main patch", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$scan,0), jitter(ffdata$feeding_at_mainfeedpatchF,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_scan))
plot_smooth(res2, view="scan", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)

```

```

plot(ffdata$number_additional_foodpatches, ffdata$feeding_at_mainfeedpatchF,type="p",col="white", xlab="Number additional food patches", ylab="Likelihood of feeding at main patch",
cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$number_additional_foodpatches,0), jitter(ffdata$feeding_at_mainfeedpatchF,0),type="p",col= alpha("black", 1), pch=1, lwd=2,
cex=sqrt(ffdata$N_number_additional_foodpatches))
plot_smooth(res2, view="number_additional_foodpatches", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)

```

```

plot(ffdata$DS, ffdata$feeding_at_mainfeedpatchF,type="p",col="white", xlab="David's score (norm)", ylab="Likelihood of feeding at main patch", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$DS,0), jitter(ffdata$feeding_at_mainfeedpatchF,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_DS))
plot_smooth(res2, view="DS", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(12,12.5,13,13.5,14,14.5,15,15.5),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)

```

```

plot(ffdata$CSIsselect_IndivAverageDiff, ffdata$feeding_at_mainfeedpatchF,type="p",col="white", xlab="CSI (average)", ylab="Likelihood of feeding at main patch", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$CSIsselect_IndivAverageDiff,0.01), jitter(ffdata$feeding_at_mainfeedpatchF,0.15),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res2, view="CSIsselect_IndivAverageDiff", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
abline( v=c(1),col="red3",lty=5,lwd=2)

```

#####

Model 4: cofeeding

```
library(tidyverse)
library(mgcv)
library(itsadug)
library(lmerTest)
library(MuMIn)
library(sjPlot)
library(sjmisc)
library(ggplot2)
library(DHARMa)
```

```
xdata = read.table("clipboard", dec=".", header=T, na.strings=c("x", "xxx"))
xdata$wt_ff=as.factor(xdata$wt_ff)
xdata$dateF=as.factor(xdata$date)
xdata$individual1=factor(xdata$individual1)
xdata$individual2=factor(xdata$individual2)
xdata$dyad=factor(xdata$dyad)
attach(xdata ,warn.conflicts = FALSE)
```

#Single boxplot rankdistance (directed) for waiting individuals compared to feeding individuals (dyadic values) + statistical test If rankdistance is significantly different from zero (0)
(Prediction: All values should be > 0 meaning all feeding dogs should rank above the waiting/queuing dog)

```
wtdata=xdata %>% filter(wt_ff=="wt_ff")
```

```
boxplot(wtdata$rankdistance)
```

```
full <- lmer(DSdistance ~ (1|individual1)+(1|individual2)+(1|dyad)+(1|dateF), data=wtdata, REML=F) #model without fixed effects except for the intercept
summary(full)$coefficients #you will already notice if the value for the intercept is not 0. # Result-> means DSdistance not different from zero
```

#But if you want to properly test for this you can compare it to a null-model with the intercept suppressed:

```
null <- lmer(DSdistance ~ 0 + (1|individual1)+(1|individual2)+(1|dyad)+(1|dateF), data=wtdata, REML=F)
```

```
testDispersion(full)
```

```
testZeroInflation(full)
```

```
anova(null, full, test="Chisq") # Result identical-> means DSdistance not different from zero
```

```
#####  
#####  
# COFEEDING  
#####
```

```
# Model 4: Compare Cofeeding-feeding dyads (=1) with non-cofeeding-feeding dyads (=0) -> which characteristics of the dyadic relationship increase probability of cofeeding?  
ffdata=xdata %>% filter(wt_ff=="ff_ff")
```

```
# Response variable: "Co-Feeding OR just feeding"
```

```
res3=gam(cofeeding ~ s(scan, bs="cr", k=3) + s(groupsize, bs="cr", k=3) + s(feeding_groupsize_mainpatch, bs="cr", k=3) + number_additional_foodpatches + s(AggDoBefore, bs="cr", k=3) +  
s(topds, bs="cr", k=3) + s(ABS_DSdistance, bs="cr", k=3) + s(CSIselect, bs="cr", k=3) + s(Familiarity, bs="cr", k=3) + s(individual1, bs="re") + s(individual2, bs="re") + s(dyad, bs="re") +  
s(scan,dateF, bs="fs", xt="cr", m=1, k=3), data= ffdata, family=binomial, method="REML")  
null3=gam(cofeeding ~ s(scan, bs="cr", k=3) + s(individual1, bs="re") + s(individual2, bs="re") + s(dyad, bs="re") + s(scan,dateF, bs="re"), data= ffdata, family=binomial, method="REML")  
res3_relatedness=gam(cofeeding ~ s(scan, bs="cr", k=3) + s(groupsize, bs="cr", k=3) + s(feeding_groupsize_mainpatch, bs="cr", k=3) + s(relatedness, bs="cr", k=3) + s(topds, bs="cr", k=3) +  
s(ABS_DSdistance, bs="cr", k=3) + s(individual1, bs="re") + s(individual2, bs="re") + s(dyad, bs="re") + s(scan,dateF, bs="fs", xt="cr", m=1, k=3), data= ffdata, family=binomial,  
method="REML")
```

```
testDispersion(res3)  
testZeroInflation(res3)  
compareML(null3, res3)  
summary(res3)  
summary(res3_relatedness)
```

```
plot_model(res3, type = "pred", terms = c("topds [all]"), color="black")+  
  scale_x_continuous(expand = c(0, 0)) +  
  #scale_y_continuous(labels = percent, limits = c(0, 1)) +  
  labs(x = "topds", y = "Predicted probability of cofeeding", title = NULL) +  
  theme_light()  
plot_model(res3, type = "pred", terms = c("ABS_DSdistance [all]"), color="black")+  
  scale_x_continuous(expand = c(0, 0)) +  
  #scale_y_continuous(labels = percent, limits = c(0, 1)) +  
  labs(x = "Rank distance in normDS (absolute distance)", y = "Predicted probability of cofeeding", title = NULL) +  
  theme_light()
```

```
plot(ffdata$scan, ffdata$cofeeding,type="p",col="white", xlab="Scan", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n",ylim=c(0,1))  
points(jitter(ffdata$scan,0), jitter(ffdata$cofeeding,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_scan_cofeed))  
plot_smooth(res3, view="scan", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
```

```
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
plot(ffdata$groupsize, ffdata$cofeeding,type="p",col="white", xlab="Overall groupsize", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$groupsize,0), jitter(ffdata$cofeeding,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_GS_cofeed))
plot_smooth(res3, view="groupsize", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
plot(ffdata$feeding_groupsize_mainpatch, ffdata$cofeeding,type="p",col="white", xlab="Number individuals feeding at mainpatch", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$feeding_groupsize_mainpatch,0), jitter(ffdata$cofeeding,0), type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_feeding_groupsize_mainpatch_cofeed))
plot_smooth(res3, view="feeding_groupsize_mainpatch", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
plot(ffdata$number_additional_foodpatches, ffdata$cofeeding,type="p",col="white", xlab="Number additional food patches", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$number_additional_foodpatches,0), jitter(ffdata$cofeeding,0),type="p",col= alpha("black", 1), pch=1, lwd=2,
cex=sqrt(ffdata$N_number_additional_foodpatches_cofeed))
plot_smooth(res3, view="number_additional_foodpatches", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
plot(ffdata$topds, ffdata$cofeeding,type="p",col="white", xlab="Top David's score (norm)", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2, yaxt="n",ylim=c(0,1))
points(jitter(ffdata$topds,0), jitter(ffdata$cofeeding,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ffdata$N_topds_cofeed))
plot_smooth(res3, view="topds", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
#axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
plot(ffdata$ABS_DSdistance, ffdata$cofeeding,type="p",col="white", xlab="Absolut rank distance (David's score (norm))", ylab="Likelihood of cofeeding", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",ylim=c(0,1))
points(jitter(ffdata$ABS_DSdistance,0.0), jitter(ffdata$cofeeding,0),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res3, view="ABS_DSdistance", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, transform=plogis, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
#####
```

Model 5: Determinants of Feeding group size:

```
library(tidyverse)
library(lmerTest)
library(ggplot2)
library(mgcv)
library(DHARMA)
library(sjPlot)
library(itsadug)
```

```
xdata = read.table("clipboard", dec=".",header=T, na.strings=c("x","xxx"))
xdata$dateF=as.factor(xdata$date)
xdata$feeding_groupsize_mainpatch_Cat2=as.factor(xdata$feeding_groupsize_mainpatch_Cat2)
attach(xdata ,warn.conflicts = FALSE)
```

```
hist(feeding_groupsize_mainpatch)
ndata=xdata %>% filter(feeding_groupsize_mainpatch>=1 & scan<=60) %>% drop_na(AggDoBefore) %>% drop_na(CSIselect_Average) #>% filter(AggDoAfter<=5)
n1data=xdata %>% filter(feeding_groupsize_mainpatch>=3 & scan<=60) %>% drop_na(AggDoAfter, AggDoBefore) %>% drop_na(CSIselect_AverageAlpha) #>% filter(AggDoAfter<=5)
```

```
res=gam(feeding_groupsize_mainpatch~ scan + AggDoBefore + groupsize
+ number_additional_foodpatches + rankpresent_mainfeedingpatch_Average
+ CSIselect_Average + Familiarity_Average
+ s(scan,dateF, bs="re"), method = "REML", data=ndata)
```

```
null=gam(feeding_groupsize_mainpatch~ scan
+ s(scan,dateF, bs="re"), method = "REML", data=ndata)
```

```
summary(res)
hist(residuals(res))
```

```
#simulationOutput <- simulateResiduals(fittedModel = res)
#plot(simulationOutput)
#testDispersion(simulationOutput)
compareML(res,null)
```

```
plot(ndata$groupsize, ndata$feeding_groupsize_mainpatch, type="p",col="white", xlab="Overall groupsize", ylab="Feeding group size at main patch", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n")
points(jitter(ndata$groupsize,0.5), jitter(ndata$feeding_groupsize_mainpatch,0.5),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res, view="groupsize", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, add=T)
```

```
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
plot(ndata$rankpresent_mainfeedingpatch_Average, ndata$feeding_groupsize_mainpatch, type="p",col="white", xlab="Average rank of individuals at main patch*", ylab="Feeding group
size at main patch", cex.lab=2.6, cex.axis=2, xaxt="n", yaxt="n")
points(jitter(ndata$rankpresent_mainfeedingpatch_Average,0.5), jitter(ndata$feeding_groupsize_mainpatch,0.5),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res, view="rankpresent_mainfeedingpatch_Average", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
```

```
plot(ndata$Familiarity_Average, ndata$feeding_groupsize_mainpatch,type="p",col="white", xlab="Scans seen together", ylab="Feeding group size at main patch ", cex.lab=2.6, cex.axis=2,
xaxt="n",yaxt="n")
points(jitter(ndata$Familiarity_Average,0), jitter(ndata$feeding_groupsize_mainpatch,0),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res, view="Familiarity_Average", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, add=T)
axis(1, at=c(0,50,100,150,200,250,300,350,400),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

```
plot(ndata$CSIsselect_Average, ndata$feeding_groupsize_mainpatch,type="p",col="white", xlab="CSI (average)", ylab="Feeding group size at main patch ", cex.lab=2.6, cex.axis=2,
xaxt="n",yaxt="n")
points(jitter(ndata$CSIsselect_Average,0), jitter(ndata$feeding_groupsize_mainpatch,0),type="p",col= alpha("black", 0.25), pch=19, lwd=2, cex=3)
plot_smooth(res, view="CSIsselect_Average", rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3, add=T)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
axis(2, at=c(0,0.5,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),cex.axis=2)
abline( v=c(1),col="red3",lty=5,lwd=2)
```

```
#####
```

Model 6: Feeding group size and aggression

```
library(tidyverse)
library(itsadug)
library(lmerTest)
library(ggplot2)
library(mgcv)
library(DHARMA)

#xdata = read.table("clipboard", dec=".",header=T, na.strings=c("x","xxx"))
xdata$dateF=as.factor(xdata$date)
xdata$feeding_groupsize_mainpatch_Cat2=as.factor(xdata$feeding_groupsize_mainpatch_Cat2)
attach(xdata ,warn.conflicts = FALSE)
boxplot(feeding_groupsize_mainpatch~scan)
ndata=xdata %>% filter(feeding_groupsize_mainpatch>=3)

cor.test(xdata$feeding_groupsize_mainpatch,xdata$rankpresent_mainfeedingpatch_Average)
plot(jitter(xdata$feeding_groupsize_mainpatch,0.2),jitter(xdata$rankpresent_mainfeedingpatch_Average,0.2))
abline(lm(feeding_groupsize_mainpatch~rankpresent_mainfeedingpatch_Average,data=xdata))
plot(jitter(xdata$feeding_groupsize_mainpatch,0.2),jitter(xdata$rankpresent_mainfeedingpatch_Average,0.2))
lines(lowess(xdata$feeding_groupsize_mainpatch,xdata$rankpresent_mainfeedingpatch_Average))
ndata=xdata %>% filter(scan<=10)
cor.test(ndata$feeding_groupsize_mainpatch,ndata$rankpresent_mainfeedingpatch_Average)
plot(jitter(ndata$feeding_groupsize_mainpatch,0.2),jitter(ndata$rankpresent_mainfeedingpatch_Average,0.2))
abline(lm(feeding_groupsize_mainpatch~rankpresent_mainfeedingpatch_Average,data=ndata))
plot(jitter(ndata$feeding_groupsize_mainpatch,0.2),jitter(ndata$rankpresent_mainfeedingpatch_Average,0.2))
lines(lowess(ndata$feeding_groupsize_mainpatch,ndata$rankpresent_mainfeedingpatch_Average))

#AGGDOAfter (all individuals at mainpatch):
ndata=xdata %>% filter(feeding_groupsize_mainpatch>=1 & scan<=60) %>% drop_na(AggDoAfter, AggDoBefore) %>% drop_na(CSIselect_Average) %>% filter(AggDoAfter<=5)

res=glmer(AggDoAfter~ feeding_groupsize_mainpatch_Cat2 + feeding_groupsize_mainpatch + scan + AggDoBefore + number_additional_foodpatches + groupsize + (1+scan|dateF),
data=ndata, family = "poisson", control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))

resG=gam(AggDoAfter~ feeding_groupsize_mainpatch_Cat2 + feeding_groupsize_mainpatch + scan + AggDoBefore
+ number_additional_foodpatches
+ CSIselect_Average + Familiarity_Average
+ s(scan,dateF, bs="re"), select = TRUE, data=ndata, method = "REML", family = "poisson")

resGL=gam(AggDoAfter~ feeding_groupsize_mainpatch + scan + AggDoBefore
+ number_additional_foodpatches
```

```
+ CSiselect_Average + Familiarity_Average
+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")
```

```
resGS=gam(AggDoAfter~ groupsize + scan + AggDoBefore
+ number_additional_foodpatches
+ CSiselect_Average + Familiarity_Average
+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")
```

```
nullG=gam(AggDoAfter~ 1+ scan + AggDoBefore
+ number_additional_foodpatches
+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")
```

```
testDispersion(resG)
testZeroInflation(resG)
summary(res)
summary(resGS)
summary(resG)
#plot(res)
simulationOutput <- simulateResiduals(fittedModel = resG)
#plot(simulationOutput)
#testDispersion(simulationOutput)
compareML(resG,nullG)
compareML(resG,resGL)
compareML(resG,resGS)
```

```
ndata$feeding_groupsize_mainpatchF=as.factor(ndata$feeding_groupsize_mainpatch)
ggplot(ndata, aes(y= AggDoAfter, x=feeding_groupsize_mainpatchF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha= 0.24, position=position_jitter(height=.1, width=.1), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0))+ labs(x = "Feeding group size at scan", y = "Number of aggressions after scan", title = NULL) #+ scale_y_continuous(lim=c(0,4))
```

#PLOTS of GAMM

Aggression by feedinggroupsize

```
plot(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfter,type="p",col="white", xlab="Feeding group size at scan", ylab="Number of aggressions after scan", cex.lab=2.6, cex.axis=2,
xaxt="n", yaxt="n",xlim=c(1,9.5))
points(jitter(ndata$feeding_groupsize_mainpatch,0), jitter(ndata$AggDoAfter,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ndata$N_AGG*10))
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="2"),xlim=c(1,2.5), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="3"),xlim=c(2.5,9.2), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
abline(h=c(0,1,2,3,4,5,6,7,8,9), v=c(2,3,4,5,6,7,8,9),col="grey40",lty=3)
```

```

abline( v=c(2.5),col="red3",lty=5,lwd=2)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,1,2,3,4,5,6,7,8,9),cex.axis=2)
# Aggression by feedinggroupsize ---- ZOOM
plot(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfter,type="p",col="white", xlab="Feeding group size at scan", ylab="Number of aggressions after scan", cex.lab=2.6, cex.axis=2,
xaxt="n",yaxt="n",xlim=c(1,9.5),ylim=c(0,0.5))
#points(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfter,type="p",col= alpha("black", 0.5), pch=1, lwd=2, cex=sqrt(ndata$N_AGG*10))
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="2") ,xlim=c(1,2.5), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black",
lwd=3,add=TRUE, transform=plogis)
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="3") ,xlim=c(2.5,9.2), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
#abline(h=c(0,1,2,3,4,5,6,7,8,9), v=c(2,3,4,5,6,7,8,9),col="grey40",lty=3)
abline( v=c(2.5),col="red3",lty=5,lwd=2)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)

```

#Only Alpha & Beta:

```

ndata=xdata %>% filter(feeding_groupsize_mainpatch>=1 & scan<=60) %>% drop_na(AggDoAfter, AggDoBefore) %>% drop_na(CSIselect_AverageAlpha) #%>% filter(AggDoAfter<=5)
res=glmer(AggDoAfterAlpha~ feeding_groupsize_mainpatch_Cat2 + feeding_groupsize_mainpatch + scan + AggDoBeforeAlpha + number_additional_foodpatches + groupsize +
(1|scan|dateF), data=ndata, family = "poisson", control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))

```

```

resG=gam(AggDoAfterAlpha~ feeding_groupsize_mainpatch_Cat2 + feeding_groupsize_mainpatch + scan + AggDoBeforeAlpha
+ number_additional_foodpatches
+ CSIselect_AverageAlpha + Familiarity_AverageAlpha
+ s(scan,dateF, bs="re"), select = TRUE, data=ndata, method = "REML", family = "poisson")

```

```

resGL=gam(AggDoAfterAlpha~ feeding_groupsize_mainpatch + scan + AggDoBeforeAlpha
+ number_additional_foodpatches
+ CSIselect_AverageAlpha + Familiarity_AverageAlpha
+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")

```

```

resGS=gam(AggDoAfterAlpha~ groupsize + scan + AggDoBeforeAlpha
+ number_additional_foodpatches
+ CSIselect_AverageAlpha + Familiarity_AverageAlpha
+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")

```

```

nullG=gam(AggDoAfterAlpha~ 1+ scan + AggDoBeforeAlpha
+ number_additional_foodpatches

+ s(scan,dateF, bs="re"), data=ndata, method = "REML", family = "poisson")

```

```

testDispersion(resG)
testZeroInflation(resG)

```

```
summary(res)
summary(resGS)
summary(resG)
#plot(res)
simulationOutput <- simulateResiduals(fittedModel = resG)
#plot(simulationOutput)
#testDispersion(simulationOutput)
compareML(resG,nullG)
compareML(resG,resGL)
compareML(resG,resGS)
```

```
ndata$feeding_groupsize_mainpatchF=as.factor(ndata$feeding_groupsize_mainpatch)
ggplot(ndata, aes(y= AggDoAfterAlpha, x=feeding_groupsize_mainpatchF)) + geom_violin(lwd=1) + scale_y_continuous(breaks = seq(0, 100, by = 1)) + theme_classic() +
theme(axis.text.y=element_text(angle=90, hjust=0.5)) + theme(plot.margin=unit(c(40,36,10,59), "pt")) + geom_point(alpha= 0.24, position=position_jitter(height=.1, width=.1), size=7,
shape=21, stroke = 1.5, fill="grey50") + theme(axis.title = element_text(size = 40)) + theme(axis.text = element_text(size = 34)) + theme(legend.text = element_text(size = 34)) +
theme(legend.title = element_text(size = 0)) + labs(x = "Feeding group size at scan", y = "Number of aggressions after scan", title = NULL) #+ scale_y_continuous(lim=c(0,4))
```

#PLOTS of GAMM

Aggression by feedinggroupsize

```
plot(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfterAlpha,type="p",col="white", xlab="Feeding group size at scan", ylab="Number of aggressions after scan", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n",xlim=c(1,9.5))
points(jitter(ndata$feeding_groupsize_mainpatch,0), jitter(ndata$AggDoAfterAlpha,0),type="p",col= alpha("black", 1), pch=1, lwd=2, cex=sqrt(ndata$N_AGGA*10))
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="2"),xlim=c(1,2.5), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="3"),xlim=c(2.5,9.2), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
abline(h=c(0,1,2,3,4,5,6,7,8,9), v=c(2,3,4,5,6,7,8,9),col="grey40",lty=3)
abline( v=c(2.5),col="red3",lty=5,lwd=2)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
axis(2, at=c(0,1,2,3,4,5,6,7,8,9),cex.axis=2)
```

Aggression by feedinggroupsize ---- ZOOM

```
plot(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfterAlpha,type="p",col="white", xlab="Feeding group size at scan", ylab="Number of aggressions after scan", cex.lab=2.6,
cex.axis=2, xaxt="n", yaxt="n",xlim=c(1,9.5),ylim=c(0,0.5))
#points(ndata$feeding_groupsize_mainpatch, ndata$AggDoAfter,type="p",col= alpha("black", 0.5), pch=1, lwd=2, cex=sqrt(ndata$N_AGG*10))
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="2"),xlim=c(1,2.5), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black",
lwd=3,add=TRUE, transform=plogis)
plot_smooth(resG, view="feeding_groupsize_mainpatch", cond=list(feeding_groupsize_mainpatch_Cat2 ="3"),xlim=c(2.5,9.2), rm.ranef=TRUE,h0=NULL, rug= FALSE, col="black", lwd=3,
add=TRUE, transform=plogis)
#abline(h=c(0,1,2,3,4,5,6,7,8,9), v=c(2,3,4,5,6,7,8,9),col="grey40",lty=3)
abline( v=c(2.5),col="red3",lty=5,lwd=2)
axis(1, at=c(-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9),cex.axis=2)
```