# Appendix S2: Ecological drivers of parasite aggregation: a meta-analysis of adherence to the 20/80 rule

*E. Sherrard-Smith, J. F. Stephenson, E. Gillingham, R. Brackley, D. Shaw, S. E. Perkins*

*26 March 2017*

## R Markdown

This supplement provides the script for the data preparation and calculation of $k$, Gini coefficient, Poulin's D and *t20*. The script and output of the analyses of the ecological factors potentially important in determining parasite aggregation in host populations. The data used for this analysis are in the supplementary file 'SherrardSmithetalDATA.csv' (Appendix S1; Table S1).

## A step by step run through of the data analysis for 20:80

We explore whether parasite populations fit the '20:80' distribution where 80% of the parasites are contained within 20% of the hosts. First, we explore real data on this relationship and then compare these distributions to those that are generated using the mean, N of hosts, and a measure of variance.

### Step 1

```
library(MASS)
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.3.3
```

### Example Using Real Data; supplied by Darren Shaw September 2013

```
CnodosMarWilliams63 <- rep(seq(0,5,1),c(496,58,18,7,2,4))
```

First data is summarised to give the appropriate information on mean, variance, host and parasite number for the comparison of these real distributions with those that are estimated.

```
data<-sort(CnodosMarWilliams63,decreasing=TRUE)
    data[1:200]
```

```
##   [1] 5 5 5 5 4 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    sampmean<-mean(data)
    sampmean              ###3.446
```

```
## [1] 0.2444444
```

```
        a<-numeric(10000)
        for(i in 1:10000) a[i]<-mean(sample(data,replace=T),na.rm=T)
        quantile(a,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1897436 0.3042735
```

```
    sumNpar<-sum(data)
    sumNpar
```

```
## [1] 143
```

```
    N<-length(data)
    N
```

```
## [1] 585
```

```
    var<-var(data)
    var
```

```
## [1] 0.4966514
```

```
    k<-(sampmean^2-(var/N))/(var-sampmean)
    k
```

```
## [1] 0.2335546
```

Generate the distribution of parasites in a given proportion of the hosts Using the real data distributions

```
real2<-data.frame(data)

##distribution of parasites in the sample population
real3<-data.frame(sort(real2[,1],decreasing=TRUE))

samp<-length(real3[,1])
nmk<-numeric(samp)
mk<-matrix(data = NA, nrow = samp, ncol = 1)

for (j in 1:samp) ## a nested loop which counts through the number of hosts
  {
            addup<-sum(real3[1:j,1])              ## cumulative sum of parasites
     if (sum(real3)>0){nmk[j]<-addup/sum(real3)} ## if there are ANY parasites
                                                  ## this gives the cumulative
                                                  ## proportion transmission per host
            else {nmk[j]<-0 }                     ## if there are no parasites pop
                                                  ## stops a problem due to division
                                                  ## by 0
  }

mk[,1]<-nmk                                       ## puts nmk into matrix mk in column i
raw<-data.frame(mk)
props<-c(0.01,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,
         0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.99)
#the proportions we will investigate
```

```
names<-paste("t",props,sep="")#names for each list of proportions
tvals<-matrix(NA,nrow=dim(raw)[2],ncol=length(props))# a matrix to store the tx values
                                                     # rows = studies,
```

```
                                                    # cols = tx values


for (j in 1:length(props))# first loop cycles through desired tx values
                          # contained in props above
{
  for (i in 1:dim(raw)[2]) # second loop cycles through each column of the data
                           # (which is a vector of proportions of parasites in each
                           # host from most to least infected)
  {
    vec<-raw[,i]             # make a vector from the column
    vec<-na.omit(vec)        # take out NAs after values reach 1 (it's a ragged array)
    N<-length(vec)           # number of hosts = number of rows
    Nx<-round(props[j]*N)    # gets the row which is x of the way down the column
    if (Nx==0) tvals[i,j]<-0 # if it rounds Nx to 0, there is no such thing as
                             # a zeroth row, so make tx 0
    else {tx<-vec[Nx]        # gets the value in that column - the proportion of
                             # parasites captured in the x most infected hosts
    tvals[i,j]<-tx           # rowi=study index colj=tx index

    }
  }

}
t.vals_Observed<-t(tvals)
```

**Step 2**

**Generate the estimated distributions from the real data summary information; mean, N, var**

```
dmean = 0.2444444                   #This is the sample mean in study 'Aa'
samp = 585                          #This is the number of hosts (N) in study 'Aa'
k = 0.2335546                       #This is the k estimate from the mean var and N
                                    #in study 'Aa'

mk<-matrix(data = NA, nrow = samp, ncol = 100)  # creates a matrix with
                                                # rows equal to sample size and
                                                # columns equal to iteration number
nmk<-numeric(samp)                              # storage vector also equal to sample size

for( i in 1:100) # bootstrap starts (runs for 10000) iterations
    {

    dist1<-rnegbin(samp, dmean, k)      # creates a negbin distribution with your data
    dizzy<-sort(dist1,decreasing = TRUE) #sorts distribution highest parasite count first

        for (j in 1:samp) # a nested loop which counts through the number of hosts

            {

                    addup<-sum(dizzy[1:j])                      # cumulative sum of parasites
                    if (sum(dizzy)>0){nmk[j]<-addup/sum(dizzy)} # if there are ANY parasites
```

```
                                                        # this gives the cumulative
                                                        # proportion transmission per
                                                        # host
                        else {nmk[j]<-0 }  # if there are no parasites pop stops
                                           # a problem due to division by 0

            }

    mk[,i]<-nmk # puts nmk into matrix mk in column i

    }

fk<-numeric(samp)

for (i in 1:samp)

    {fk[i]<-mean(mk[i,])} # take average of 10000 iterations for each row (each sample)

raw<-data.frame(fk)
props<-c(0.01,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,
         0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.99)
###The proportion of hosts to which we want to fit parasite populations

tvals<-matrix(NA,nrow=dim(raw)[2],ncol=length(props))

for (j in 1:length(props))
    {
    for (i in 1:dim(raw)[2])
        {
        vec<-raw[,i]
        vec<-na.omit(vec)
        N<-length(vec)
        Nx<-round(props[j]*N)

        if (Nx==0) tvals[i,j]<-0
        else {tx<-vec[Nx]
        tvals[i,j]<-tx
            }
        }
    }

t.vals_Estimated<-data.frame(tvals)
```

### Data comparison

These distributions can be put together for all the data with complete raw parasite counts. These can then be compared statistically. The data are provided in Appendix S1; Table S2.

```
dat<-read.csv("C:\\Malaria\\SherrardSmithetalGROUNDTRUTH.csv")
stackT20<-c(dat$estimated0.2,dat$real0.2)
stackT40<-c(dat$estimated0.4,dat$real0.4)
stackT60<-c(dat$estimated0.6,dat$real0.6)
stackT80<-c(dat$estimated0.8,dat$real0.8)
```

```r
##create a data frame with the stacked data
datatype<-c(rep("estimate",27),rep("real",27))
mat<-data.frame(datatype,stackT20,stackT40,
    stackT60,stackT80)

###Compare observed and estimated data
modt20<-glm(log(mat$stackT20)~mat$datatype,family=gaussian)
summary.lm(modt20)
```

```
##
## Call:
## glm(formula = log(mat$stackT20) ~ mat$datatype, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0275 -1.3108 -0.4036  0.6438  5.7576
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.13618    0.36020  17.036   <2e-16 ***
## mat$datatypereal  0.02678    0.50940   0.053    0.958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.872 on 52 degrees of freedom
## Multiple R-squared:  5.315e-05,  Adjusted R-squared:  -0.01918
## F-statistic: 0.002764 on 1 and 52 DF,  p-value: 0.9583
```

```r
modt40<-glm(log(mat$stackT40)~mat$datatype,family=gaussian)
summary.lm(modt40)
```

```
##
## Call:
## glm(formula = log(mat$stackT40) ~ mat$datatype, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2155 -1.3408 -0.3663  0.5986  5.5696
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.341620   0.367099  17.275   <2e-16 ***
## mat$datatypereal 0.009364   0.519157   0.018    0.986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.908 on 52 degrees of freedom
## Multiple R-squared:  6.256e-06,  Adjusted R-squared:  -0.01922
## F-statistic: 0.0003253 on 1 and 52 DF,  p-value: 0.9857
```

```r
modt60<-glm(log(mat$stackT60)~mat$datatype,family=gaussian)
summary.lm(modt60)
```

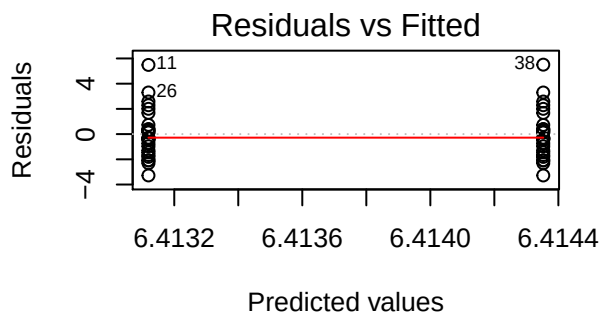
```
##
## Call:
```

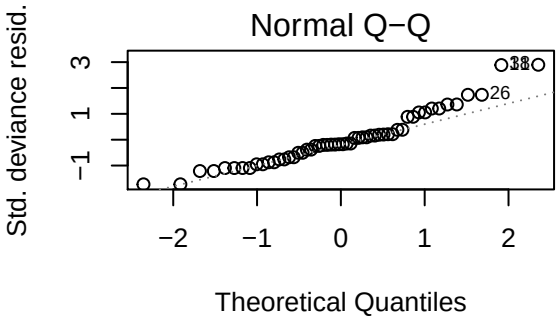
```
## glm(formula = log(mat$stackT60) ~ mat$datatype, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2697 -1.3981 -0.3226  0.6269  5.5154
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.399939   0.371093   17.25   <2e-16 ***
## mat$datatypereal 0.005259   0.524805    0.01    0.992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.928 on 52 degrees of freedom
## Multiple R-squared:  1.931e-06,  Adjusted R-squared:  -0.01923
## F-statistic: 0.0001004 on 1 and 52 DF,  p-value: 0.992
```

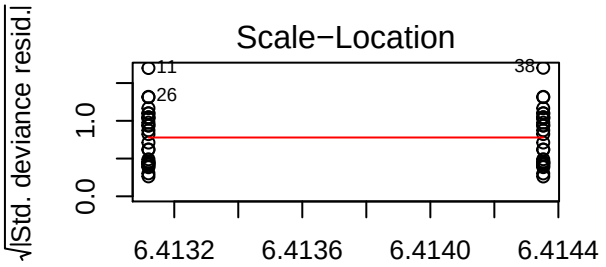
```
modt80<-glm(log(mat$stackT80)~mat$datatype,family=gaussian)
summary.lm(modt80)
```

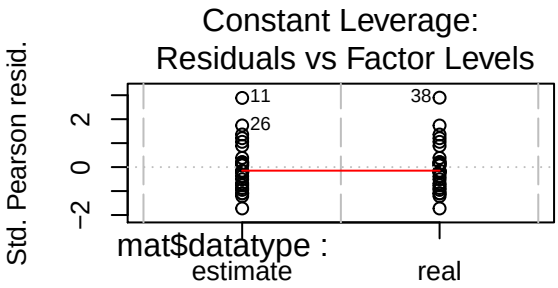
```
##
## Call:
## glm(formula = log(mat$stackT80) ~ mat$datatype, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2789 -1.4103 -0.3338  0.6440  5.5062
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.413119   0.372688  17.208   <2e-16 ***
## mat$datatypereal 0.001233   0.527061   0.002    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.937 on 52 degrees of freedom
## Multiple R-squared:  1.053e-07,  Adjusted R-squared:  -0.01923
## F-statistic: 5.476e-06 on 1 and 52 DF,  p-value: 0.9981
```

```
par(mfrow=c(2,2))
plot(modt80)
```

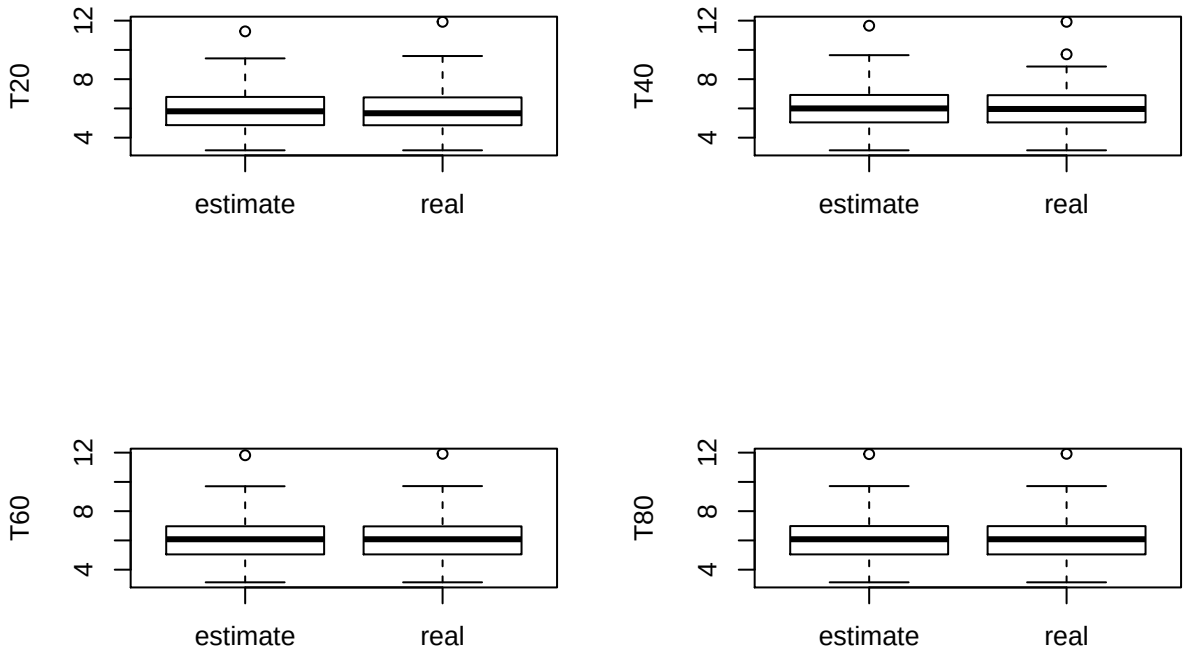



```
boxplot(log(mat$stackT20)~mat$datatype,ylab="T20")
boxplot(log(mat$stackT40)~mat$datatype,ylab="T40")
boxplot(log(mat$stackT60)~mat$datatype,ylab="T60")
boxplot(log(mat$stackT80)~mat$datatype,ylab="T80")
```




**Step 3**

**Calculating Poulin's D and Gini Coefficient**

```
################################################
##
## Poulin's D function (Sherrard-Smith, R version 3.3.3)
##
################################################

poulD <- function(dat){
        N=length(dat)
x=sumx=numeric(N)

    for(j in 1:N){
            x[j]=mean(dat[1:j])
         }
    for(i in 1:N){
            sumx[i]=sum(x[1:i])
        }
PoulinsD = 1 - (2*sum(sumx)/ mean(dat)*N*(N+1))
    return(list(PoulinsD))
}
################################################
```

```
##
## Gini coefficient function (Sherrard-Smith, R version 3.3.3)
##
################################################
Gini <- function(dat){
        N=length(dat)
    abs_diff = array(dim=c(N,N))
x=sumx=ABS_3=numeric(N)

    for(j in 1:N){
            x[j]=mean(dat[1:j])
         }
    for(i in 1:N){
    for(j in 1:N){
            abs_diff[j,i] = (dat[i] - dat[j])
         }
         }
abs_diff2 = ifelse(abs_diff < 0, abs_diff*-1,abs_diff)
GiniCoef = sum(abs_diff2) / (2 * N * sum(dat))
return(list(GiniCoef))
}
```

**Step 4**

**Ecological Analysis**

This section provides the script and output of the analyses of the ecological factors potentially important in determining parasite aggregation in host populations. The data used for this analysis are in Appendix S1; Table S1. This file contains the following columns (for more details please see the main text):

- **label**: Identifies individual record - these match up with those given in the summary of the references
- **lab.field**: Whether the study was conducted in a lab or field setting
- **parasite.taxa**; parasite.type, parasite.species: Details about the parasites included in the database
- **host.taxa**; host.common.name; host.species: Details about the hosts included in the database
- **transmission**: The parasite transmission mode
- **host.management**: Whether and how the host population is managed
- **habitat**: The host's habitat
- **eat.meat**: Whether or not the host eats other animals
- **social**: Whether or not the host is social
- **n**: The number of hosts sampled
- **k**: The negative binomial shape parameter
- **mean**: The mean abundance (i.e. total.parasites divided by n)
- **total.parasites**: The total number of parasites recorded in the study
- **gini**: The gini coefficient
- **poulinD**: Poulin's index of dispersion, D
- **t1-t99**: The proportion of parasites in x proportion of hosts

```
rm(list=ls()) #remove previous objects and packages
dframe1<-read.csv('C://Malaria//SherrardSmithetalDATA.csv')
require(lme4)
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:reshape':
##
##     expand
```

```
require(car)
```

```
## Loading required package: car
```

```
require(MASS)
require(MuMIn)
```

```
## Loading required package: MuMIn
```

```
## Warning: package 'MuMIn' was built under R version 3.3.3
```

```
require(arm)
```

```
## Loading required package: arm
```

```
## Warning: package 'arm' was built under R version 3.3.3
```

```
##
## arm (Version 1.9-3, built: 2016-11-21)
```

```
## Working directory is C:/Users/Ellie/Documents/RProjects/2080Model
```

```
##
## Attaching package: 'arm'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
require(multcomp)
```

```
## Loading required package: multcomp
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
require(visreg)
```

```
## Loading required package: visreg
```

```
## Warning: package 'visreg' was built under R version 3.3.3
```

```
require(glmmADMB)
```

```
## Loading required package: glmmADMB
```

```
##
## Attaching package: 'glmmADMB'
```

```
## The following object is masked from 'package:MASS':
##
##     stepAIC
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
require(ResourceSelection)
```

```
## Loading required package: ResourceSelection
```

```
## Warning: package 'ResourceSelection' was built under R version 3.3.3
```

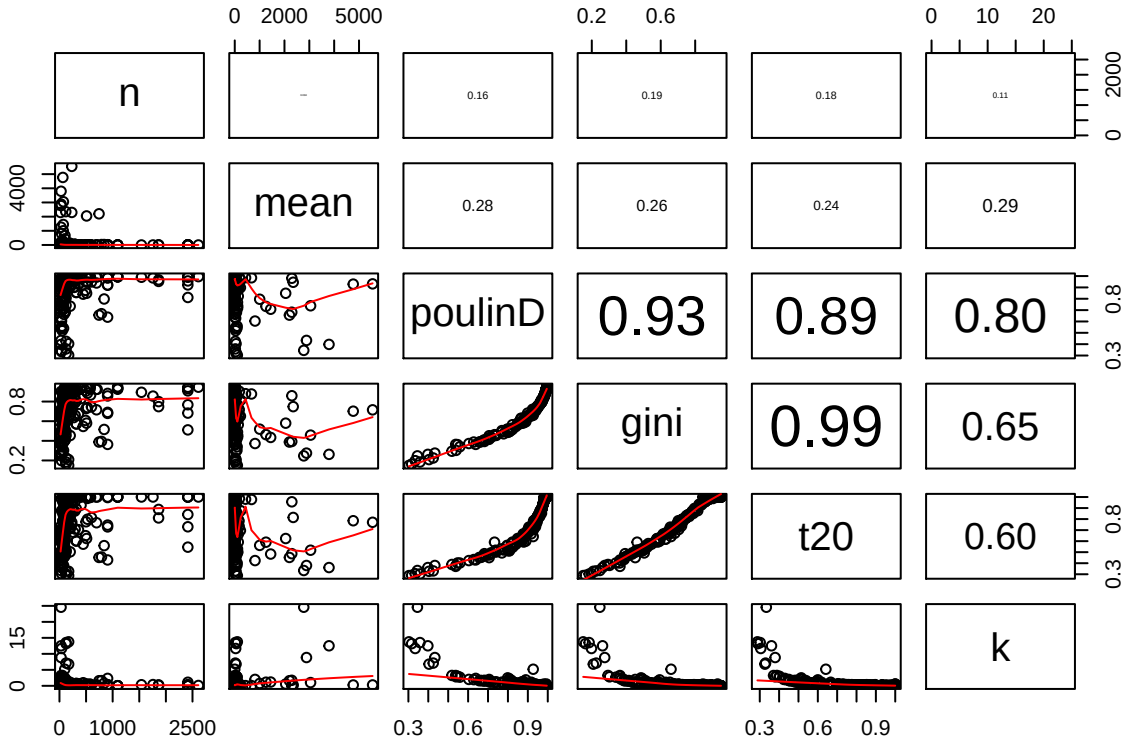
```
## ResourceSelection 0.3-2   2017-02-28
```

```
# Remove one study with outlying mean abundance
df2 <- subset(dframe1, label != "227")
```

First, we assessed the correlations between our putative explanatory variables to check their suitability for inclusion in the model.

```
# This function is from 'Mixed effects models and extensions in
# ecology with R'. (2009).Zuur, AF, Ieno, EN, Walker, N, Saveliev,
# AA, and Smith, GM. Springer.

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
usr <- par("usr")
on.exit(par(usr))
par(usr = c(0, 1, 0, 1))
r <- abs(cor(x, y))
txt <- format(c(r, 0.123456789), digits = digits)[1]
txt <- paste(prefix, txt, sep = "")
if (missing(cex.cor))
cex.cor <- 0.8/strwidth(txt)
text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(~n + mean + poulinD + gini + t20 + k, data = df2, lower.panel = panel.smooth,
upper.panel = panel.cor, na.action = na.omit)
```




```
mod <- glmmadmb(gini ~ n + log(mean) + transmission + social +
habitat + eat.meat + host.management + parasite.type +
(1 | parasite.taxa/parasite.species) + (1 | host.taxa/host.species),
data = df2, family = "beta")
Anova(mod)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: gini
##                  Df   Chisq Pr(>Chisq)
## n                 1  3.8503    0.04974 *
## log(mean)         1 48.2037  3.842e-12 ***
## transmission      2  0.5198    0.77114
## social            1  4.8811    0.02715 *
## habitat           2 20.1207  4.274e-05 ***
## eat.meat          1  0.0745    0.78491
## host.management   3  1.1803    0.75774
## parasite.type     1  0.1205    0.72846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

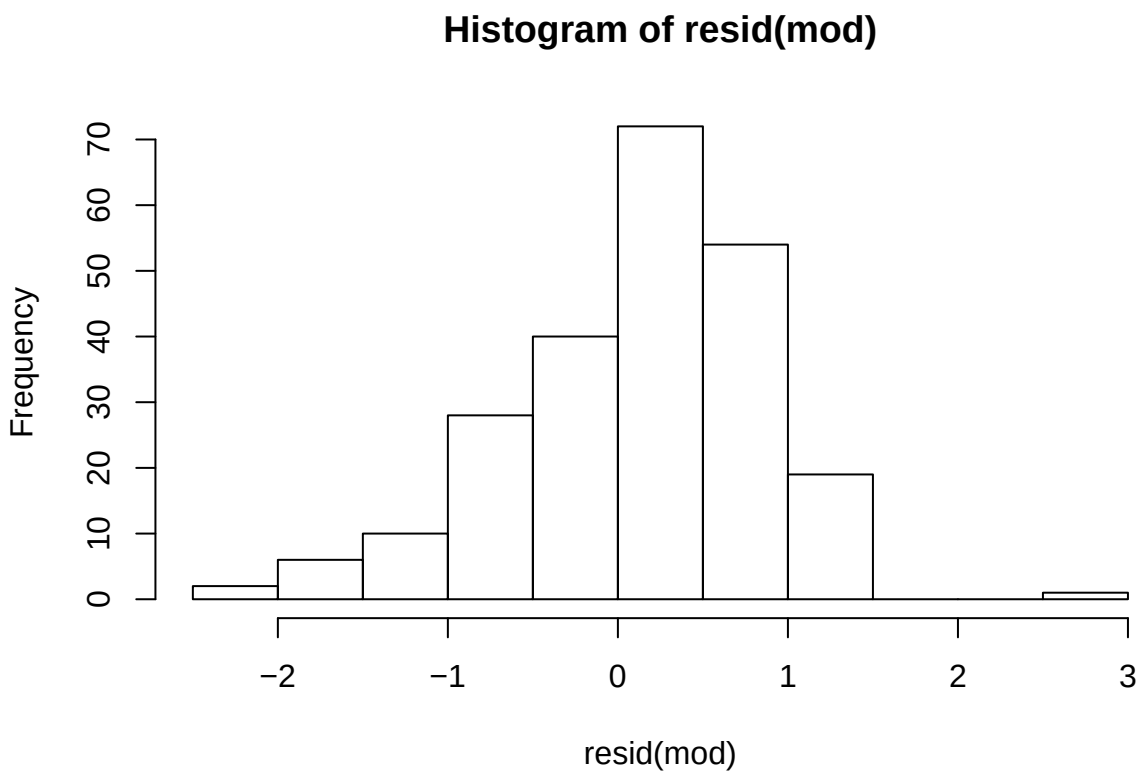
```
##
## Call:
## glmmadmb(formula = gini ~ n + log(mean) + transmission + social +
##     habitat + eat.meat + host.management + parasite.type + (1 |
```

```
##     parasite.taxa/parasite.species) + (1 | host.taxa/host.species),
##     data = df2, family = "beta")
##
## AIC: -272.4
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.086824   0.399260    2.72   0.0065 **
## n                          0.000258   0.000131    1.96   0.0497 *
## log(mean)                 -0.201198   0.028979   -6.94  3.8e-12 ***
## transmissioneat.parasite -0.084682   0.251550   -0.34   0.7364
## transmissioneat.prey     -0.163466   0.249090   -0.66   0.5117
## socialyes                 -0.314321   0.142270   -2.21   0.0272 *
## habitatSemi-aquatic        0.256892   0.261460    0.98   0.3258
## habitatTerrestrial         0.794798   0.185250    4.29  1.8e-05 ***
## eat.meatyes                0.044508   0.163080    0.27   0.7849
## host.managementexp        -0.342808   0.436900   -0.78   0.4327
## host.managementseminat    -0.186634   0.306860   -0.61   0.5431
## host.managementwild       -0.070116   0.195060   -0.36   0.7193
## parasite.typeworm          0.120311   0.346540    0.35   0.7285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of observations: total=232, parasite.taxa=4, parasite.taxa:parasite.species=180, host.taxa=6,
## Random effect variance(s):

## Warning in .local(x, sigma, ...): 'sigma' and 'rdig' arguments are present
## for compatibility only: ignored

## Group=parasite.taxa
##             Variance StdDev
## (Intercept)   0.0289   0.17
## Group=parasite.taxa:parasite.species
##             Variance StdDev
## (Intercept)    0.305 0.5523
## Group=host.taxa
##              Variance    StdDev
## (Intercept) 1.225e-07 0.0003499
## Group=host.taxa:host.species
##             Variance StdDev
## (Intercept)  0.02308 0.1519
##
## Beta dispersion parameter: 19.134 (std. err.: 3.9972)
##
## Log-likelihood: 154.215
```

```
hist(resid(mod))
```




```
# this function tests for overdispersion. It's from
# http://glmm.wikidot.com/faq
overdisp_fun <- function(model) {
## number of variance parameters in an n-by-n
## variance-covariance matrix
vpars <- function(m) {
nrow(m) * (nrow(m) + 1)/2
}
model.df <- sum(sapply(VarCorr(model), vpars)) + length(fixef(model))
rdf <- nrow(model.frame(model)) - model.df
rp <- residuals(model, type = "pearson")
Pearson.chisq <- sum(rp^2)
prat <- Pearson.chisq/rdf
pval <- pchisq(Pearson.chisq, df = rdf, lower.tail = FALSE)
c(chisq = Pearson.chisq, ratio = prat, rdf = rdf, p = pval)
}
overdisp_fun(mod)
```

```
## Warning in .local(x, sigma, ...): 'sigma' and 'rdig' arguments are present
## for compatibility only: ignored
```

```
##       chisq       ratio         rdf           p
## 127.8743279   0.5947643 215.0000000   0.9999996
```

Hosmer-Lemeshow goodness of fit test with

```
# ResourceSelection package
hoslem.test(df2$gini, y = fitted(mod))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  df2$gini, fitted(mod)
## X-squared = 1.774, df = 8, p-value = 0.9872
```

### Post-hoc tests of significant effects

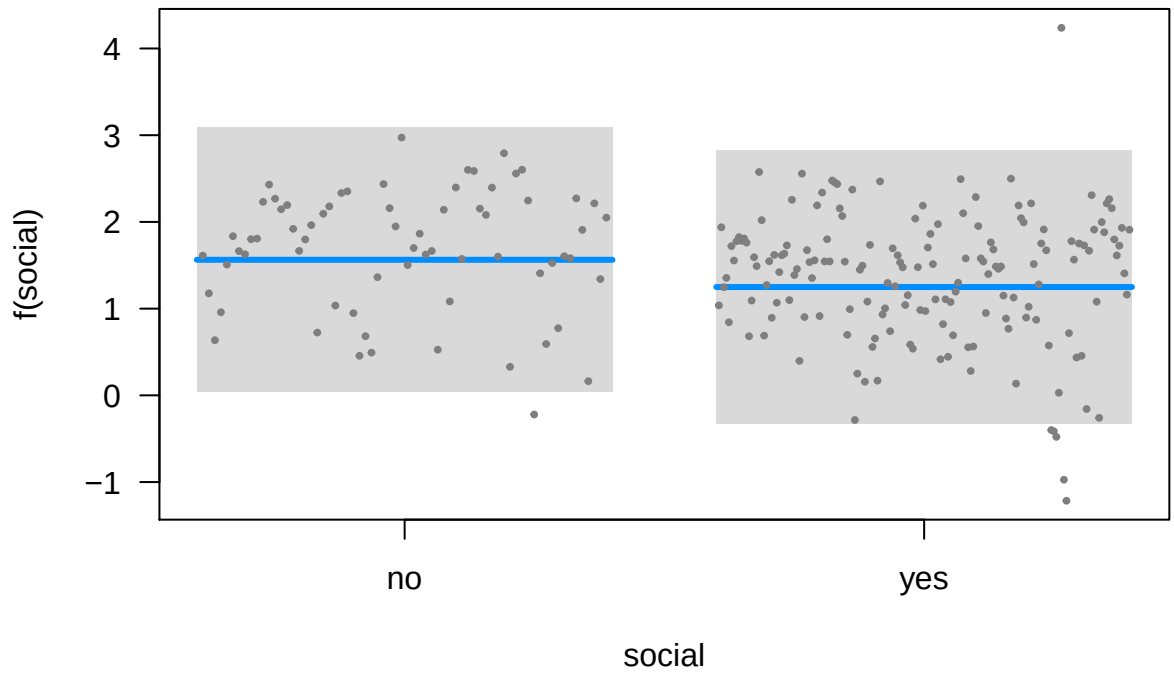
#### Social structure

Solitary animals have more highly aggregated parasites than social animals.

```
summary(df2$social)
```

```
##  no yes
##  68 164
```

```
visreg(mod, 'social') # Gives the partial residuals used to plot
```




```
# Fig 3.
summary(glht(mod, mcp(social = "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
```

```
## Fit: glmmadmb(formula = gini ~ n + log(mean) + transmission + social +
##     habitat + eat.meat + host.management + parasite.type + (1 |
##     parasite.taxa/parasite.species) + (1 | host.taxa/host.species),
##     data = df2, family = "beta")
##
## Linear Hypotheses:
##               Estimate Std. Error z value Pr(>|z|)
## yes - no == 0  -0.3143     0.1423  -2.209   0.0272 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

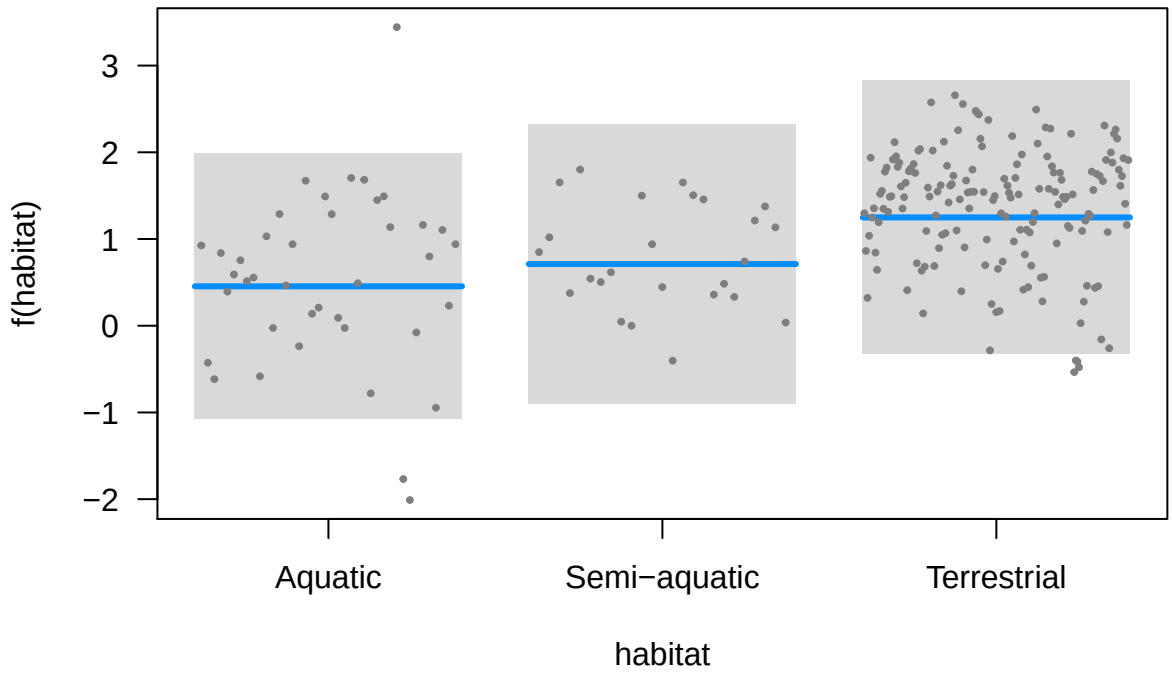
### Host habitat

Aquatic parasites are significantly less aggregated than terrestrial ones.

```
summary(df2$habitat)
```

```
##      Aquatic Semi-aquatic  Terrestrial
##           40           25          167
```

```
visreg(mod, 'habitat') #Gives the partial residuals used to plot
```




```
# Fig 3.
summary(glht(mod, mcp(habitat = "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
```

```
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: glmmadmb(formula = gini ~ n + log(mean) + transmission + social +
##     habitat + eat.meat + host.management + parasite.type + (1 |
##     parasite.taxa/parasite.species) + (1 | host.taxa/host.species),
##     data = df2, family = "beta")
##
## Linear Hypotheses:
##                                 Estimate Std. Error z value Pr(>|z|)
## Semi-aquatic - Aquatic == 0       0.2569     0.2615   0.983    0.567
## Terrestrial - Aquatic == 0        0.7948     0.1852   4.290   <0.001 ***
## Terrestrial - Semi-aquatic == 0   0.5379     0.3316   1.622    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

### Sample size, n

Aggregation increases as you sample more hosts.

```
visreg(mod, "n")
```