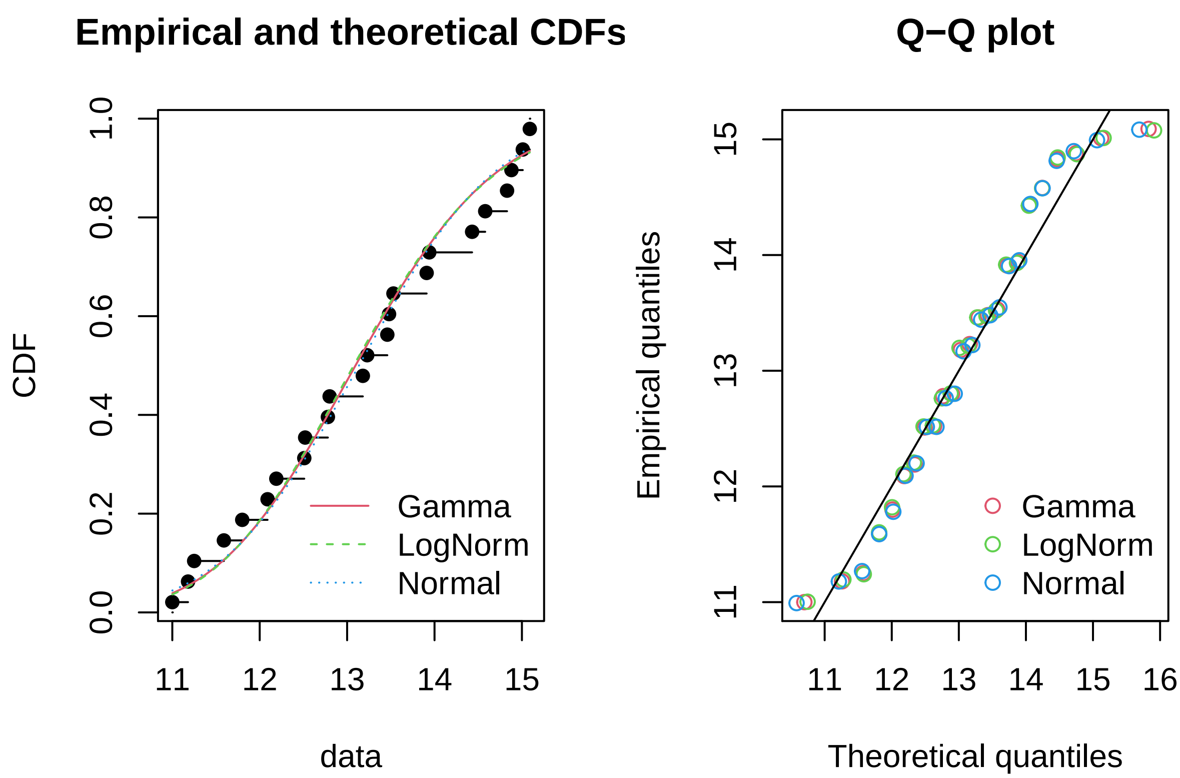
Ramírez-Mejía et al. 2020. Functional diversity of phyllostomid bats in an urban-rural landscape: A scale-dependent analysis

Diversidad funcional de murcielagos filostómidos en un paisaje urbano-rural: un análisis escala-dependiente

Andrés F. Ramírez-Mejía

This document's primary purpose is to facilitate access to the code from different devices, so I did not include a detailed explanation. I hope it is clear enough to understand the way I conducted GLMs analyses. The script is not fully edited, so please feel free to contact me if you have any questions or suggestions.

```
#Data analisis

#bat functional diversity


#CWM WL----

DF <- read.csv("data.actualizado.csv", header = T, sep = ";", dec = ",")


for (i in seq_along(DF)) {
  if (grepl("^f(.*)*\\.05$", colnames(DF)[i]) ||
      grepl("^g(.*)*\\.05$", colnames(DF[i])) ||
      grepl("^f(.*)*\\.125$", colnames(DF[i])) ||
      grepl("^g(.*)*\\.2$", colnames(DF[i]))) {
    DF[i] <- as.vector(scale(DF[[i]], center = T, scale = T))
  }
}


library(fitdistrplus)
DF5 <- na.omit(DF)
gamma=fitdist(DF5$CWM.wing_loading,"gamma")
lognormal=fitdist(DF5$CWM.wing_loading,"lnorm")
normal=fitdist(DF5$CWM.wing_loading, "norm")#mejor ajuste
par(mfrow= c(1, 2))
cdfcomp(list(gamma, lognormal,normal),addlegend=T,
        legendtext = c("Gamma","LogNorm","Normal"))
qqcomp(list(gamma, lognormal,normal),addlegend=T,
       legendtext = c("Gamma","LogNorm","Normal"))
```




```
gofstat(list(gamma,lognormal,normal))
```

```
## Goodness-of-fit statistics
##                              1-mle-gamma 2-mle-lnorm 3-mle-norm
## Kolmogorov-Smirnov statistic  0.09763015  0.09690194 0.09892122
## Cramer-von Mises statistic    0.03641414  0.03753757 0.03604458
## Anderson-Darling statistic    0.30976499  0.31625363 0.30804751
##
## Goodness-of-fit criteria
##                                1-mle-gamma 2-mle-lnorm 3-mle-norm
## Akaike's Information Criterion    83.07398    83.18944   82.98321
## Bayesian Information Criterion    85.43009    85.54555   85.33932
```

```
par(mfrow = c(1, 1))



CWMm1 <- glm(CWM.wing_loading ~ forest.perc.05 + grass.perc.05 +
            forest.perc.125 + grass.perc.2, data = DF)

summary(CWMm1)
```

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

```
## glm(formula = CWM.wing_loading ~ forest.perc.05 + grass.perc.05 +
##     forest.perc.125 + grass.perc.2, data = DF)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.494  -0.262  -0.006   0.224   1.172
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       13.1279     0.1226 107.040  < 2e-16 ***
## forest.perc.05     0.7283     0.3870   1.882  0.07452 .
## grass.perc.05     -0.2793     0.3533  -0.791  0.43844
## forest.perc.125   -0.8525     0.1869  -4.561  0.00019 ***
## grass.perc.2      -1.0154     0.3380  -3.004  0.00701 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.373834)
##
##     Null deviance: 37.8458  on 24  degrees of freedom
## Residual deviance:  7.4767  on 20  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 52.77
##
## Number of Fisher Scoring iterations: 2
```

```
anova(CWMm1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: CWM.wing_loading
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                               24     37.846
## forest.perc.05   1   0.0389        23     37.807 0.7468937
## grass.perc.05    1  22.3795        22     15.427 1.016e-14 ***
## forest.perc.125  1   4.5762        21     10.851 0.0004674 ***
## grass.perc.2     1   3.3744        20      7.477 0.0026610 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CWMm2 <- update(CWMm1, ~.-forest.perc.05)
anova(CWMm1,CWMm2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: CWM.wing_loading ~ forest.perc.05 + grass.perc.05 + forest.perc.125 +
##     grass.perc.2
```

```
## Model 2: CWM.wing_loading ~ grass.perc.05 + forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        20     7.4767
## 2        21     8.8002 -1  -1.3236  0.05989 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CWMm3 <- update(CWMm2, ~. -grass.perc.05)
anova(CWMm2, CWMm3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: CWM.wing_loading ~ grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: CWM.wing_loading ~ forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        21     8.8002
## 2        22    18.7609 -1  -9.9607 1.086e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CWMm4 <- update(CWMm2, ~. -forest.perc.125)
anova(CWMm2, CWMm4, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: CWM.wing_loading ~ grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: CWM.wing_loading ~ grass.perc.05 + grass.perc.2
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        21     8.8002
## 2        22    15.9643 -1  -7.1641 3.554e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CWMm5 <- update(CWMm2, ~. -grass.perc.2)
anova(CWMm2, CWMm5, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: CWM.wing_loading ~ grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: CWM.wing_loading ~ grass.perc.05 + forest.perc.125
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        21     8.8002
## 2        22    12.1039 -1  -3.3036 0.004989 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(CWMm2)#varianza explicada por el modelo 78.73
```

```
##
## Call:
## glm(formula = CWM.wing_loading ~ grass.perc.05 + forest.perc.125 +
```

```
##     grass.perc.2, data = DF)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.23946  -0.33385  -0.08385   0.29781   1.39054
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.1341     0.1298 101.184  < 2e-16 ***
## grass.perc.05    -0.8645     0.1773  -4.875 8.05e-05 ***
## forest.perc.125  -0.6261     0.1514  -4.135 0.000471 ***
## grass.perc.2     -0.4454     0.1586  -2.808 0.010545 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.4190593)
##
##     Null deviance: 37.8458  on 24  degrees of freedom
## Residual deviance:  8.8002  on 21  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 54.845
##
## Number of Fisher Scoring iterations: 2
```

```
anova(CWMm2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: CWM.wing_loading
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                               24     37.846
## grass.perc.05    1  17.0960        23     20.750 40.7961 2.482e-06 ***
## forest.perc.125  1   8.6459        22     12.104 20.6318 0.0001778 ***
## grass.perc.2     1   3.3036        21      8.800  7.8835 0.0105447 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

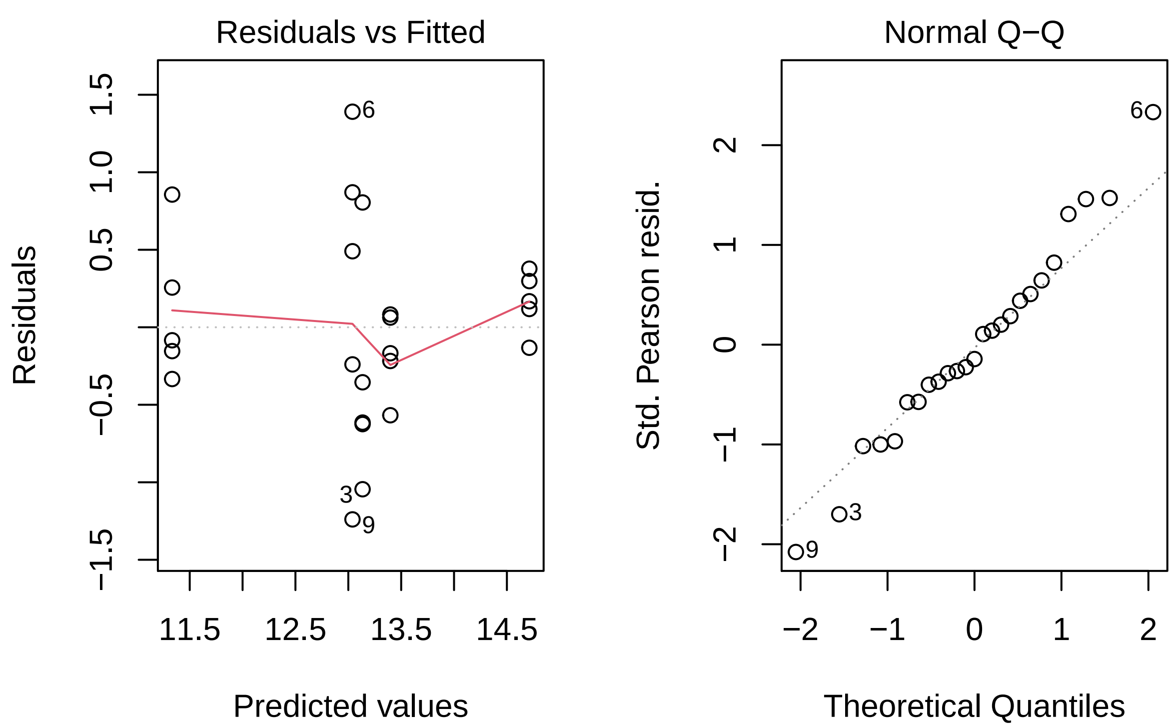
```
confint(CWMm2)
```

```
##                      2.5 %     97.5 %
## (Intercept)     12.8797252 13.3885512
## grass.perc.05   -1.2120871 -0.5169779
## forest.perc.125 -0.9229093 -0.3293174
## grass.perc.2    -0.7562698 -0.1344786
```

```r
#funciones para el c?lculo de effect size
r.se<-function(t.val,df,n){
  # Calcula los valores de r y su error standard en la escala arctanh-1 de Fisher.
  # INPUTS DE LA FUNCION:
  # t.val: valor de t para la var. explciativa numerica obtenido de la tabla summary del modelo
  # df: grados de libertad residuales del modelo obtenido del summary o la tabla anova del modelo
  # n: número de datos obtenido del # de filas del DF que no sean NA.
  # OUTPUTS DE LA FUNCION: los valores de r y su error standard en la escala transformada
  r<-0.5*log((1+(t.val/sqrt((t.val)^2+df)))/(1-(t.val/sqrt((t.val)^2+df))))
  se<-(1/sqrt(n-3))
  names(r)<-"r transf"
  names(se)<-"SE(r transf)"
  c(r,se)
}

#c?lculo de los cuantiles para calcular los IC
conf.limits.nct <- function(tval.1,df,conf){
  # tval.1: t valor de la tabla summary
  # df: grados de libertad del modelo
  # conf: nivel de confianza deseado. Generalmente 0.95
  # Esta función emplea la función ptnoncent que tambien debe ser cargada.
  Result <- matrix(NA,1,4)
  tval <- abs(tval.1)
  ############################ Estima el cuantil inferior ###########################
  ulim <- 1 - (1-conf)/2
  lc <- c(-tval,tval/2,tval)
  while(ptnoncent(tval,df,lc[1])<ulim)    {
    lc <- c(lc[1]-tval,lc[1],lc[3])  }
  #  Estima el cuantil inferior
  diff <- 1
  while(diff > .00000001)     {
    if(ptnoncent(tval,df,lc[2])<ulim)
      lc <- c(lc[1],(lc[1]+lc[2])/2,lc[2])
    else lc <- c(lc[2],(lc[2]+lc[3])/2,lc[3])
    diff <- abs(ptnoncent(tval,df,lc[2]) - ulim)
    ucdf <- ptnoncent(tval,df,lc[2])}
  res.1 <- ifelse(tval.1 >= 0,lc[2],-lc[2])
  ############################ Estima el cuantil superior ###########################
  llim <- (1-conf)/2
  uc <- c(tval,1.5*tval,2*tval)
  while(ptnoncent(tval,df,uc[3])>llim)   {
    uc <- c(uc[1],uc[3],uc[3]+tval)
  }
  diff <- 1
  while(diff > .00000001)        {
    if(ptnoncent(tval,df,uc[2])<llim)
      uc <- c(uc[1],(uc[1]+uc[2])/2,uc[2])
    else uc <- c(uc[2],(uc[2]+uc[3])/2,uc[3])
    diff <- abs(ptnoncent(tval,df,uc[2]) - llim)
    lcdf <- ptnoncent(tval,df,uc[2])
  }
  res <- ifelse(tval.1 >= 0,uc[2],-uc[2])
  ################################ Pone los cuantiles y sus niveles confianza en una matriz ############
```

```
  Result[1,1] <- min(res,res.1)
  Result[1,2] <- lcdf
  Result[1,3] <- max(res,res.1)
  Result[1,4] <- ucdf
  dimnames(Result) <- list("Valores", c("Cuantil Inf", "Prob.Limite.Inf", "Cuantil.Sup", "Prob.Limite.Su
  Result
}

#intervalos de confianza del effect size
IC_r<-function(r,se.r,cuan.inf,cuan.sup){
  # La funcion IC_r calcula los limites inferior y superior del IC de r.
  # INPUTS DE LA FUNCION:
  # Los valores de r y su error standard se(r) calculados en la función r.se
  # Los cuantiles de la distr t no centrada calculados en la función conf.limits.nct
  # Luego de calcular el los limites del IC, aplica la transformacion inversa de Fisher
  # para expresar el r y su IC en la escala correcta.
  # OUTPUTS DE LA FUNCION: los valores de r y los limites de su intervalo de confianza
  r<-(exp(2*r)-1)/(exp(2*r)+1)
  l.inf=(exp(2*(r-(cuan.inf*se.r)))-1)/(exp(2*(r-(cuan.inf*se.r)))+1)
  l.sup=(exp(2*(r+(cuan.sup*se.r)))-1)/(exp(2*(r+(cuan.sup*se.r)))+1)
  names(r)<-"r"
  names(l.inf)<-"Limite.Inf IC"
  names(l.sup)<-"Limite.Sup IC"
  return(c(r,l.inf,l.sup))
}

ptnoncent <- function(tx, df, nonc = 0, itrmax = 1000, errmax= 1E-6)
{if(min(df) <= 0)
  stop("All df must be > 0")
  lengths <- c(length(tx), length(df), length(nonc))
  if(any(lengths < (ltx <- max(lengths)))) {
    tx <- rep(tx, length.out = ltx)
    df <- rep(df, length.out = ltx)
    nonc <- rep(nonc, length.out = ltx)}
  tnc <- numeric(ltx)
  del <- nonc
  negdel <- (tx < 0)
  del <- ifelse(negdel,  - del, del)
  xx <- (tx * tx)/(tx * tx + df)
  lambda <- del * del
  p <- 0.5 * exp(-0.5 * lambda)
  q <- 0.79788456080286496 * p * del
  ss <- 0.5 - p
  a <- rep(0.5, ltx)
  b <- 0.5 * df
  rxb <- (1 - xx)^b
  albeta <- 0.57236494292469997 + lgamma(b) - lgamma(a + b)
  xodd <- pbeta(xx, a, b)
  godd <- 2 * rxb * exp(a * log(xx) - albeta)
  xeven <- 1 - rxb
  geven <- b * xx * rxb
  tnc <- p * xodd + q * xeven
  itr <- 0
```

```r
  err <- rep(1, ltx)
  while((itr <- itr + 1) <= itrmax && max(err) > errmax) {
    a <- a + 1
    xodd <- xodd - godd
    xeven <- xeven - geven
    godd <- (godd * xx * (a + b - 1))/a
    geven <- (geven * xx * (a + b - 0.5))/(a +0.5)
    p <- (p * lambda)/(2 * itr)
    q <- (q * lambda)/(2 * itr + 1)
    ss <- ss - p
    tnc <- tnc + p * xodd + q * xeven
    err <- 2 * ss * (xodd - godd)}
  if(itr > itrmax)
    warning("maximum number of iteration reached")
  tnc <- tnc + 1 - pnorm(del)
  ifelse(negdel, 1 - tnc, tnc)}


#verificaci?n del ajuste del modelo con un an?lisis de residuales

par(mfrow=c(1,2))
plot(CWMm2, which=c(1,2))
```




```r
par(mfrow=c(1,1))
```

```r
#visualizaci?n de resultados

library(ggplot2)
library(visreg);library(arm)


#tama?o del efecto e IC grass05
r.se(4.85,21,25)
```

```
##     r transf SE(r transf)
##    0.9220420    0.2132007
```

```r
conf.limits.nct(-4.85,21,0.95)
```

```
##         Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores   -7.255092           0.025   -2.370111           0.975
```

```r
IC_r(0.92,0.21,2.37,7.25)#ES=0.72; IC.i=0.22; IC.s=0.97
```

```
##             r Limite.Inf IC Limite.Sup IC
##     0.7258974     0.2243172     0.9779563
```

```r
p1 <- visreg(CWMm2, xvar="grass.perc.05",rug=1,type="conditional",
       scale="response",xlab="grass.per.05.c", cex=1.7,
       ylab="CWM.wing_loading", gg=T)+geom_jitter() + labs(x="Grass area 0.5 km", y="CWM wing loading")
  theme_classic() + theme(axis.text = element_text(size = 11, colour = "black"), axis.title = element_te
                         plot.margin = unit(c(0,0,0,0.2), "cm"))+
  ylim(11,15) + annotate("text", x = -1.4, y = 15, label = expression(paste(italic("r"),"= 0.72 (0.22 -
  #theme(legend.position = "none", axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5, face =
#ggsave("CWM_WL_grass05c.tiff", width = 15, height = 15, units = "cm", dpi = 300)


#tama?o del efecto e IC forest125
r.se(4.13,21,25)
```

```
##     r transf SE(r transf)
##    0.8097883    0.2132007
```

```r
conf.limits.nct(4.13,21,0.95)
```

```
##         Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores    1.774874      0.02499999    6.414747           0.975
```

```r
IC_r(0.80,0.21,1.77,6.41)#ES=0.66; IC.i=0.28; IC.s=0.96
```

```
##             r Limite.Inf IC Limite.Sup IC
##     0.6640368     0.2842842     0.9647368
```

```r
p2 <- visreg(CWMm2, xvar="forest.perc.125",rug=1,type="conditional",
       scale="response",xlab="forest.perc.125", cex=1.7,
       ylab="CWM.wing_loading", gg=T)+geom_jitter() + labs(x="Forest area 1.25 km", y="CWM wing loading
  theme_classic() + theme(axis.text.y = element_blank(), axis.text.x = element_text(size = 11, colour =
                          axis.title.y = element_blank(), axis.title.x = element_text(size = 12), plot.r
  ylim(11,15)+ annotate("text", x = -0.95, y = 15, label = expression(paste(italic("r"), "= 0.66 (0.28 -
  scale_x_continuous(breaks = c(-1,0,1))# +

#ggsave("CWM_WL_forest125c.tiff", width = 15, height = 15, units = "cm", dpi = 300)

#tama?o del efecto e IC grass2
r.se(2.8,21,25)
```

```
##     r transf SE(r transf)
##    0.5782430    0.2132007
```

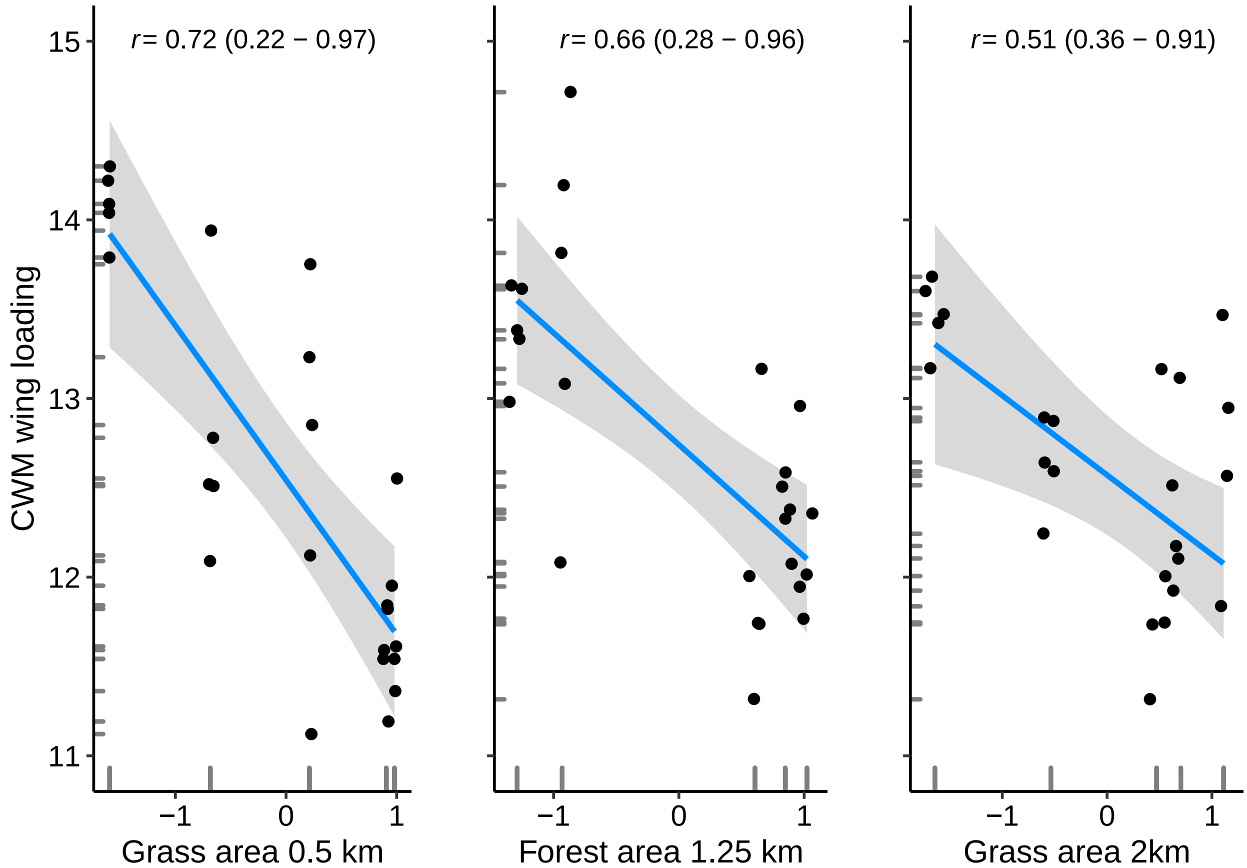
```r
conf.limits.nct(2.8,21,0.95)
```

```
##         Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores   0.6388913      0.02499999    4.904924           0.975
```

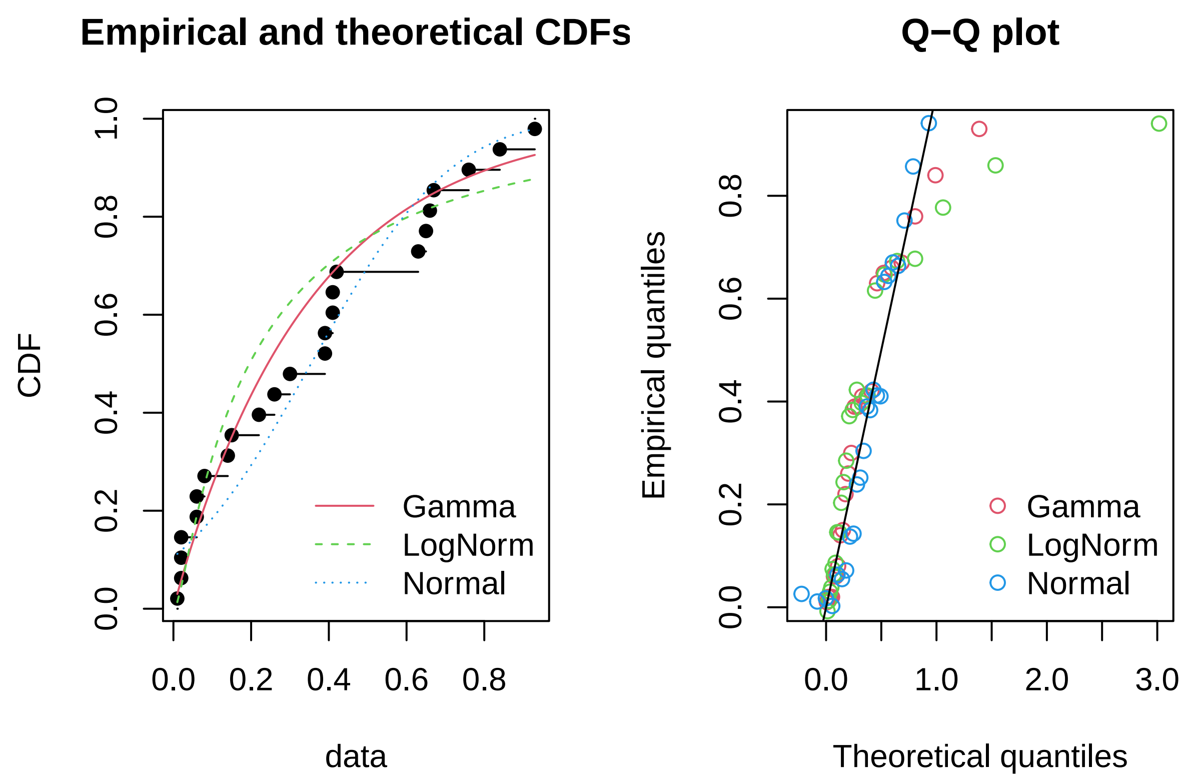
```r
IC_r(0.57,0.21,0.63,4.90)#ES=0.51; IC.i=0.36; IC.s=0.91
```

```
##             r Limite.Inf IC Limite.Sup IC
##     0.5153593     0.3653613     0.9128500
```

```r
p3 <- visreg(CWMm2, xvar="grass.perc.2",rug= 1, type="conditional",
       scale="response",xlab="grass.per.2.c", cex=1.7,
       ylab="CWM.wing_loading", gg=T)+geom_jitter() + labs(x="Grass area 2km", y="CWM wing loading") +
  theme_classic() + theme(axis.text.x = element_text(size = 11, colour = "black"), axis.text.y = element
                          axis.title.y = element_blank(), axis.title.x = element_text(size = 12), plot.r
  ylim(11,15)+annotate("text", x = -1.3, y = 15, label = expression(paste(italic("r"), "= 0.51 (0.36 - 0

#ggsave("CWM_WL_grass2c.tiff", width = 15, height = 15, units = "cm", dpi = 300)


library(cowplot)

plot_grid(p1, p2, p3, ncol = 3, align = "h")
```




```
#ggsave("model_cwm.tiff", width= 15, height = 14,units = "cm", dpi = 300)


#FRic----


DF5 <- na.omit(DF)
gamma=fitdist(DF5$FRic,"gamma")#mejor
lognormal=fitdist(DF5$FRic,"lnorm")
normal=fitdist(DF5$FRic, "norm")
par(mfrow = c(1, 2))
cdfcomp(list(gamma, lognormal,normal),addlegend=T,
        legendtext = c("Gamma","LogNorm","Normal"))
qqcomp(list(gamma, lognormal,normal),addlegend=T,
       legendtext = c("Gamma","LogNorm","Normal"))
```




```
gofstat(list(gamma,lognormal,normal))
```

```
## Goodness-of-fit statistics
##                              1-mle-gamma 2-mle-lnorm 3-mle-norm
## Kolmogorov-Smirnov statistic   0.1687054   0.1965126  0.1397231
## Cramer-von Mises statistic     0.1059986   0.1872270  0.1071601
## Anderson-Darling statistic     0.6742318   1.1473109  0.7187851
##
## Goodness-of-fit criteria
##                                1-mle-gamma 2-mle-lnorm 3-mle-norm
## Akaike's Information Criterion    2.165341    7.914117   11.50885
## Bayesian Information Criterion    4.521449   10.270225   13.86496
```

```
par(mfrow = c(1, 1))




library(lme4)

fricM <- glm(FRic ~ forest.perc.05 + grass.perc.05 +
               forest.perc.125 + grass.perc.2, family = Gamma(link = "log"), data = DF)
summary(fricM)
```

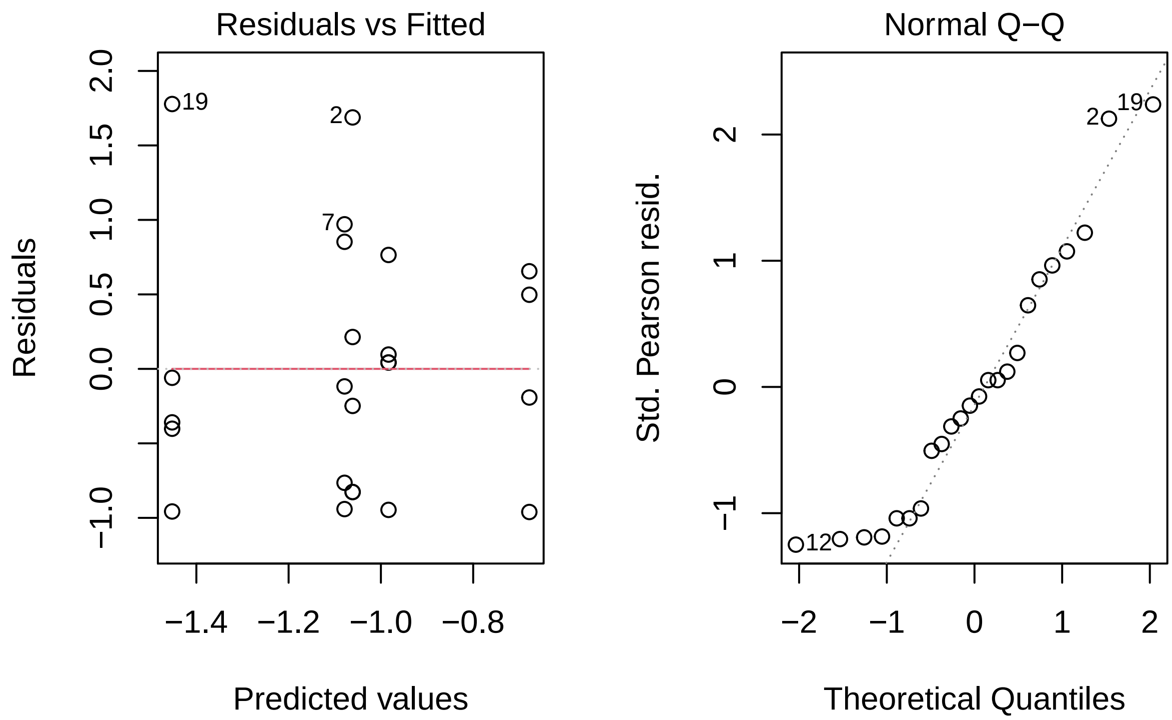
```
##
```

```
## Call:
## glm(formula = FRic ~ forest.perc.05 + grass.perc.05 + forest.perc.125 +
##     grass.perc.2, family = Gamma(link = "log"), data = DF)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.13222  -1.21619  -0.09184   0.46221   1.22974
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.03653    0.18389  -5.637 1.96e-05 ***
## forest.perc.05  -0.26342    0.56892  -0.463    0.649
## grass.perc.05   -0.27040    0.51328  -0.527    0.604
## forest.perc.125 -0.14022    0.27179  -0.516    0.612
## grass.perc.2     0.07819    0.49091   0.159    0.875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.788294)
##
##     Null deviance: 28.549  on 23  degrees of freedom
## Residual deviance: 27.186  on 19  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 9.1069
##
## Number of Fisher Scoring iterations: 6
```

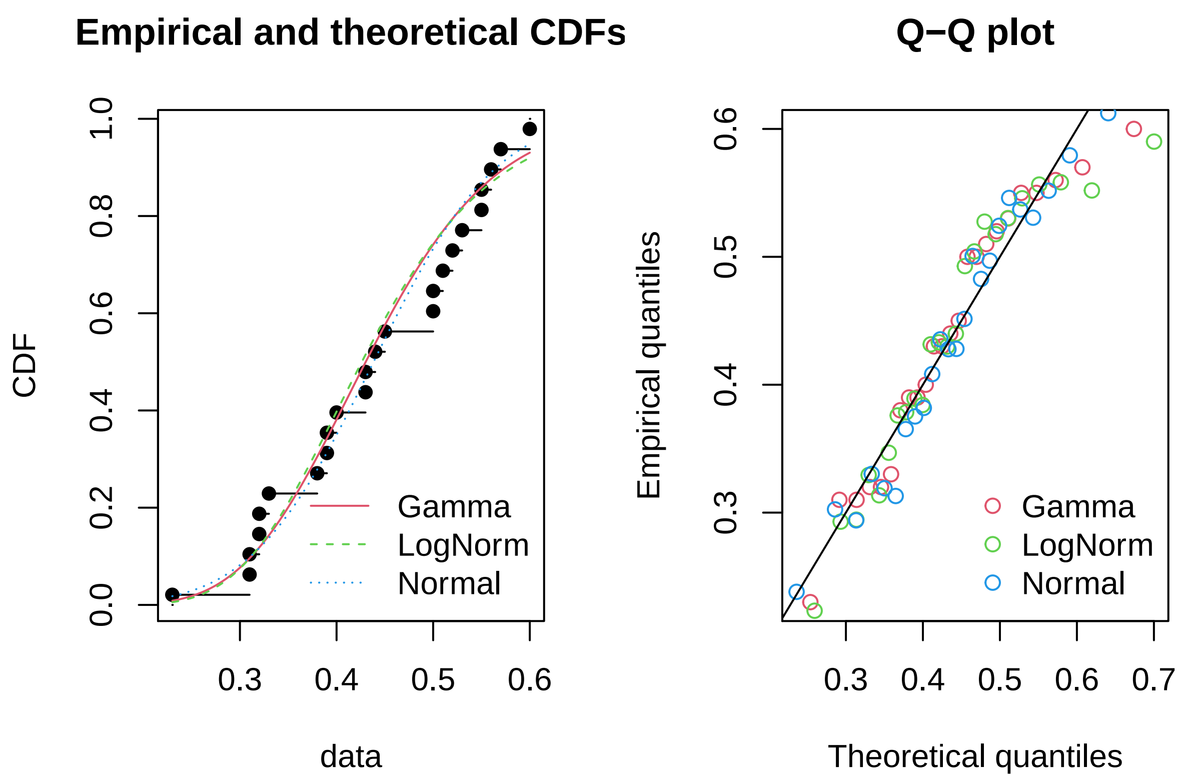
```
anova(fricM, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model: Gamma, link: log
##
## Response: FRic
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               23     28.549
## forest.perc.05   1  0.55761        22     27.991   0.4003
## grass.perc.05    1  0.39202        21     27.599   0.4807
## forest.perc.125  1  0.39344        20     27.206   0.4799
## grass.perc.2     1  0.02018        19     27.186   0.8729
```

```
#an?lisis de residuales

par(mfrow=c(1,2))
plot(fricM, which=c(1,2))
```




```
par(mfrow=c(1,1))

#Feve----


gamma=fitdist(DF5$FEve,"gamma")
lognormal=fitdist(DF5$FEve,"lnorm")
normal=fitdist(DF5$FEve, "norm")#mejor ajuste
par(mfrow=c(1,2))
cdfcomp(list(gamma, lognormal,normal),addlegend=T,
        legendtext = c("Gamma","LogNorm","Normal"))
qqcomp(list(gamma, lognormal,normal),addlegend=T,
       legendtext = c("Gamma","LogNorm","Normal"))
```




```
par(mfrow=c(1,1),cex.lab=1, cex.axis=1, cex=1)
gofstat(list(gamma,lognormal,normal))
```

```
## Goodness-of-fit statistics
##                              1-mle-gamma 2-mle-lnorm 3-mle-norm
## Kolmogorov-Smirnov statistic  0.15796992   0.1606056 0.14924994
## Cramer-von Mises statistic    0.07283643   0.0807354 0.06649415
## Anderson-Darling statistic    0.49687757   0.5531113 0.43891383
##
## Goodness-of-fit criteria
##                                1-mle-gamma 2-mle-lnorm 3-mle-norm
## Akaike's Information Criterion   -37.63102   -36.58339  -38.72368
## Bayesian Information Criterion   -35.27491   -34.22728  -36.36757
```

```
par(1, 1)
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
```

```
feveM <- glm(FEve~forest.perc.05 + grass.perc.05 +
               forest.perc.125 + grass.perc.2, data = DF)
summary(feveM)
```

```
##
## Call:
## glm(formula = FEve ~ forest.perc.05 + grass.perc.05 + forest.perc.125 +
##     grass.perc.2, data = DF)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.1580  -0.0385  -0.0210   0.0390   0.1720
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.43308    0.01737  24.935 5.59e-16 ***
## forest.perc.05  -0.09066    0.05373  -1.687  0.10791
## grass.perc.05   -0.09282    0.04848  -1.915  0.07071 .
## forest.perc.125  0.05139    0.02567   2.002  0.05979 .
## grass.perc.2     0.13499    0.04636   2.911  0.00895 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.007031579)
##
##     Null deviance: 0.23693  on 23  degrees of freedom
## Residual deviance: 0.13360  on 19  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: -44.474
##
## Number of Fisher Scoring iterations: 2
```

```
anova(feveM,test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: FEve
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               23    0.23693
## forest.perc.05   1 0.022941        22    0.21399 0.070877 .
## grass.perc.05    1 0.018739        21    0.19525 0.102583
## forest.perc.125  1 0.002048        20    0.19320 0.589378
## grass.perc.2     1 0.059605        19    0.13360 0.003597 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#LRT
feveM1 <- update(feveM, ~.-forest.perc.05)
anova(feveM,feveM1, test = "Chi")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: FEve ~ forest.perc.05 + grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: FEve ~ grass.perc.05 + forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        19    0.13360
## 2        20    0.15362 -1 -0.020016  0.09157 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
feveM2 <- update(feveM1, ~.-grass.perc.05)
anova(feveM1, feveM2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FEve ~ grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: FEve ~ forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1        20    0.15362
## 2        21    0.15938 -1 -0.0057668   0.3862
```

```
feveM3 <- update(feveM2,~.-forest.perc.125)
anova(feveM2,feveM3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FEve ~ forest.perc.125 + grass.perc.2
## Model 2: FEve ~ grass.perc.2
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        21    0.15938
## 2        22    0.17938 -1 -0.019995   0.1046
```

```
feveM4 <- update(feveM3, ~.-grass.perc.2)
anova(feveM4, feveM3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FEve ~ 1
## Model 2: FEve ~ grass.perc.2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        23    0.23693
## 2        22    0.17938  1 0.057555 0.007887 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(feveM3)
```

```
##
## Call:
## glm(formula = FEve ~ grass.perc.2, data = DF)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -0.16032  -0.06516  -0.00513   0.06058   0.17369
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.43616    0.01845  23.640   <2e-16 ***
## grass.perc.2  0.04859    0.01829   2.657   0.0144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.008153549)
##
##     Null deviance: 0.23693  on 23  degrees of freedom
## Residual deviance: 0.17938  on 22  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: -43.402
##
## Number of Fisher Scoring iterations: 2
```

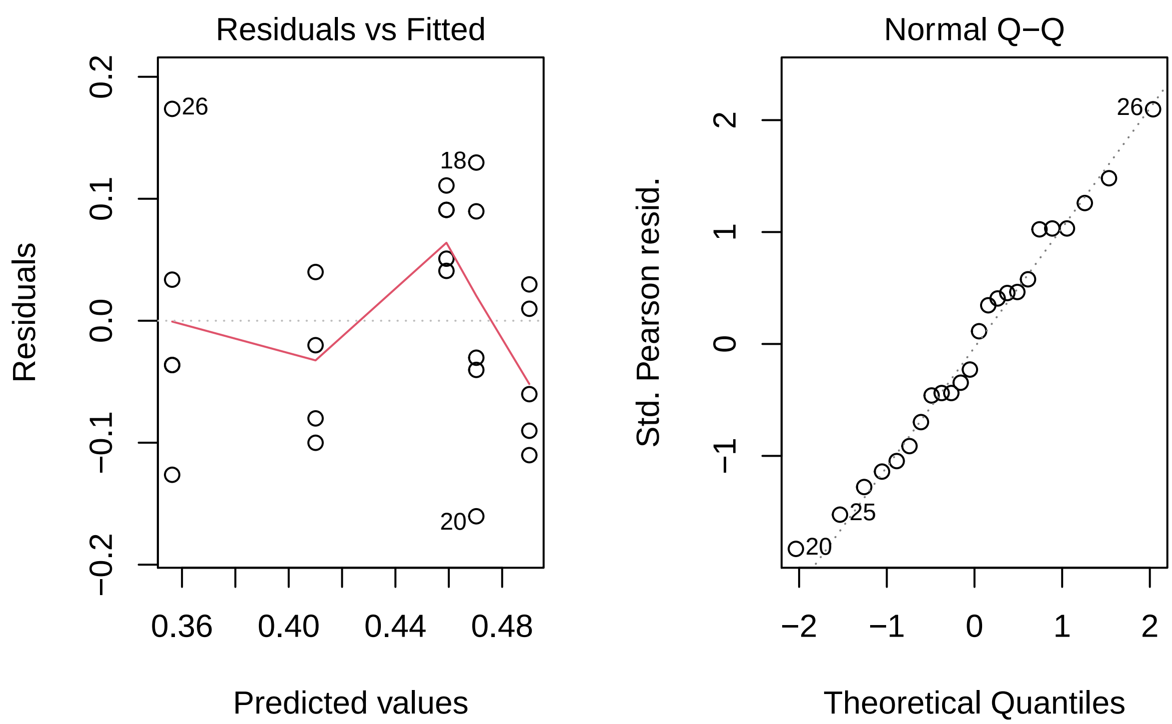
```
anova(feveM3, test = "F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: FEve
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev      F Pr(>F)
## NULL                            23    0.23693
## grass.perc.2  1 0.057555        22    0.17938 7.0589 0.0144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(feveM3)
```

```
##                   2.5 %     97.5 %
## (Intercept)  0.39999770 0.47232021
## grass.perc.2 0.01274584 0.08443882
```

```
#an?lsiis de residuales
par(mfrow=c(1,2))
plot(feveM3, which=c(1,2))
```




```
par(mfrow=c(1,1), mar=c(1,1,1,1))

#visualizaci?n del modelo
#tama?o del efecto e intervalos de confianza
r.se(2.65,22,25)
```

```
##     r transf SE(r transf)
##    0.5385662    0.2132007
```

```
conf.limits.nct(2.65,22,0.95)
```

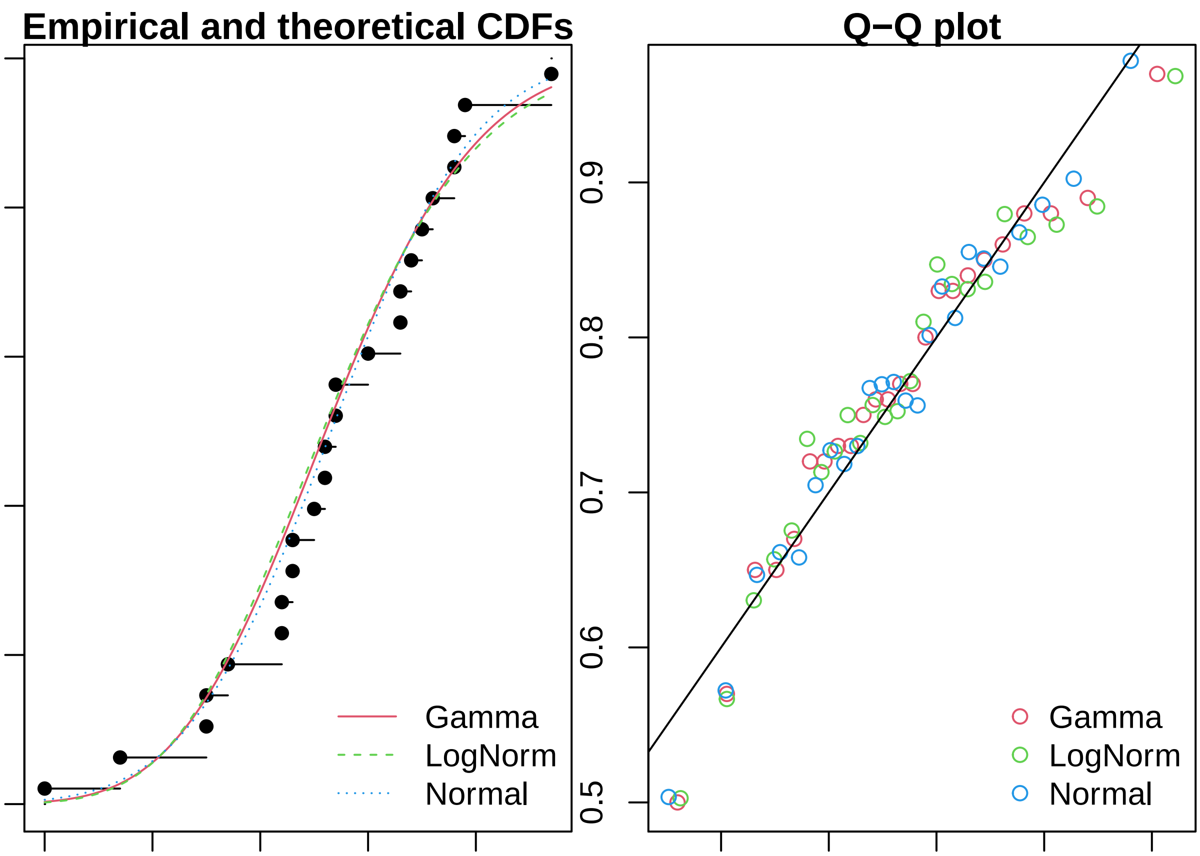
```
##         Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores   0.5151539           0.025    4.732927           0.975
```

```
IC_r(0.53,0.21,0.51,4.73)#ES=0.48; IC.i=0.36; IC.s=0.90
```

```
##             r Limite.Inf IC Limite.Sup IC
##     0.4853811     0.3612138     0.9012206
```

```
i1 <- visreg(feveM3, xvar="grass.perc.2",type="conditional",
       scale="response",xlab="grass.perc.2",rug=1, cex=1.7,
       ylab="FEve", gg=T)+geom_jitter() + labs(x="Grass area 2 km", y="FEve") +
  theme_classic() + theme(axis.title = element_text(size = 15), axis.text = element_text(colour = "blac
```

```
  ylim(0.2,0.7) +annotate("text", x = -1, y = 0.7, label = expression(paste(italic("r"), "= 0.48 (0.36
#ggsave("feve_grass2c.tiff", width = 15, height = 15, units = "cm", dpi = 300)




#FDiv----


gamma=fitdist(DF5$FDiv,"gamma")
lognormal=fitdist(DF5$FDiv,"lnorm")
normal=fitdist(DF5$FDiv, "norm")#mejor ajuste
par(mfrow = c(1, 2))
cdfcomp(list(gamma, lognormal,normal),addlegend=T,
        legendtext = c("Gamma","LogNorm","Normal"))
qqcomp(list(gamma, lognormal,normal),addlegend=T,
       legendtext = c("Gamma","LogNorm","Normal"))
```




```
par(mfrow = c(1, 1))

gofstat(list(lognormal,gamma, normal))
```

```
## Goodness-of-fit statistics
##                              1-mle-lnorm 2-mle-gamma 3-mle-norm
## Kolmogorov-Smirnov statistic  0.15445563  0.14388153 0.12337248
```

```
## Cramer-von Mises statistic    0.07575846  0.06285575 0.04379054
## Anderson-Darling statistic    0.51429916  0.43011403 0.30059867
##
## Goodness-of-fit criteria
##                                 1-mle-lnorm 2-mle-gamma 3-mle-norm
## Akaike's Information Criterion   -33.35516   -34.38333  -35.92992
## Bayesian Information Criterion   -30.99905   -32.02722  -33.57381
```

```
mfdiv <- glm(FDiv~forest.perc.05 + grass.perc.05 +
               forest.perc.125 + grass.perc.2, data = DF)
summary(mfdiv)
```

```
##
## Call:
## glm(formula = FDiv ~ forest.perc.05 + grass.perc.05 + forest.perc.125 +
##     grass.perc.2, data = DF)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.22800  -0.06075   0.01550   0.03800   0.15200
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.77269    0.01964  39.340   <2e-16 ***
## forest.perc.05  -0.01461    0.06076  -0.240   0.8126
## grass.perc.05   -0.00129    0.05482  -0.024   0.9815
## forest.perc.125 -0.05339    0.02903  -1.839   0.0816 .
## grass.perc.2     0.02079    0.05243   0.396   0.6962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.008992632)
##
##     Null deviance: 0.26618  on 23  degrees of freedom
## Residual deviance: 0.17086  on 19  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: -38.57
##
## Number of Fisher Scoring iterations: 2
```

```
anova(mfdiv,test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: FDiv
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               23    0.26618
```

```
## forest.perc.05   1 0.025106        22    0.24108  0.09475 .
## grass.perc.05    1 0.016248        21    0.22483  0.17889
## forest.perc.125  1 0.052556        20    0.17227  0.01563 *
## grass.perc.2     1 0.001414        19    0.17086  0.69175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#LRT

mfdiv1 <- update(mfdiv, ~.-forest.perc.05)
anova(mfdiv,mfdiv1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FDiv ~ forest.perc.05 + grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: FDiv ~ grass.perc.05 + forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df    Deviance Pr(>Chi)
## 1        19    0.17086
## 2        20    0.17138 -1 -0.00051972     0.81
```

```
mfdiv2 <- update(mfdiv1, ~.-grass.perc.05)
anova(mfdiv1,mfdiv2, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FDiv ~ grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: FDiv ~ forest.perc.125 + grass.perc.2
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1        20    0.17138
## 2        21    0.17271 -1 -0.0013345   0.6931
```

```
mfdiv3 <- update(mfdiv2,~.-grass.perc.2)
anova(mfdiv2,mfdiv3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FDiv ~ forest.perc.125 + grass.perc.2
## Model 2: FDiv ~ forest.perc.125
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1        21    0.17271
## 2        22    0.17795 -1 -0.0052401   0.4248
```

```
mfdiv4 <- update(mfdiv3, ~.-forest.perc.125)
anova(mfdiv4,mfdiv3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: FDiv ~ 1
## Model 2: FDiv ~ forest.perc.125
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        23    0.26618
```

```
## 2        22    0.17795  1 0.088229 0.0009577 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mfdiv3)
```

```
##
## Call:
## glm(formula = FDiv ~ forest.perc.125, data = DF)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.207096  -0.055971   0.001819   0.044590   0.172904
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.77274    0.01848  41.821  < 2e-16 ***
## forest.perc.125 -0.06422    0.01944  -3.303  0.00324 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.008088831)
##
##     Null deviance: 0.26618  on 23  degrees of freedom
## Residual deviance: 0.17795  on 22  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: -43.594
##
## Number of Fisher Scoring iterations: 2
```

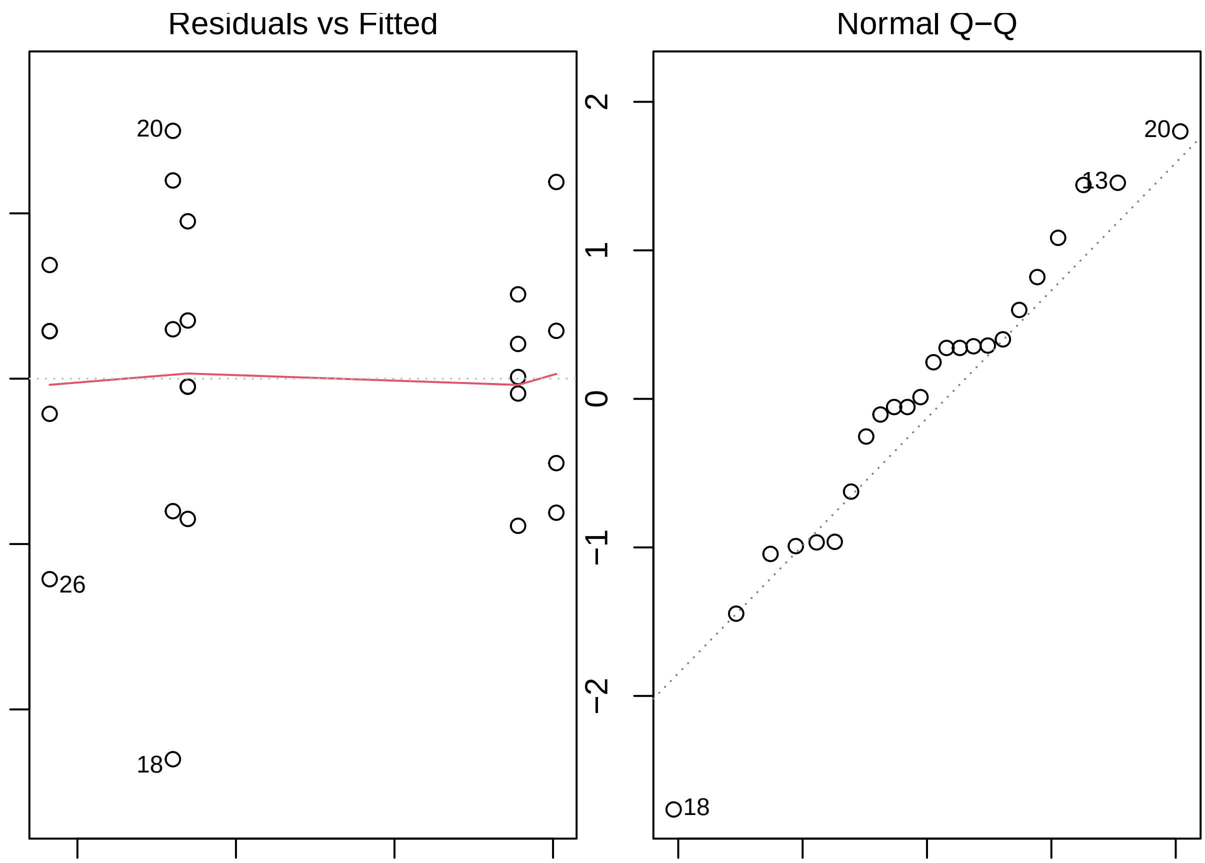
```
anova(mfdiv3, test = "F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: FDiv
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev      F   Pr(>F)
## NULL                               23    0.26618
## forest.perc.125  1 0.088229        22    0.17795 10.908 0.003242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(mfdiv3)
```

```
##                      2.5 %      97.5 %
## (Intercept)      0.7365236  0.80895275
## forest.perc.125 -0.1023309 -0.02610842
```

```
#residuales
par(mfrow=c(1,2))
plot(mfdiv1, which = c(1,2))
```




```
par(mfrow=c(1,1))


#visualizaci?n de resultados
#tama?o del efecto e intervalos de confianza
r.se(3.3,22,25)
```

```
##     r transf SE(r transf)
##    0.6555825    0.2132007
```

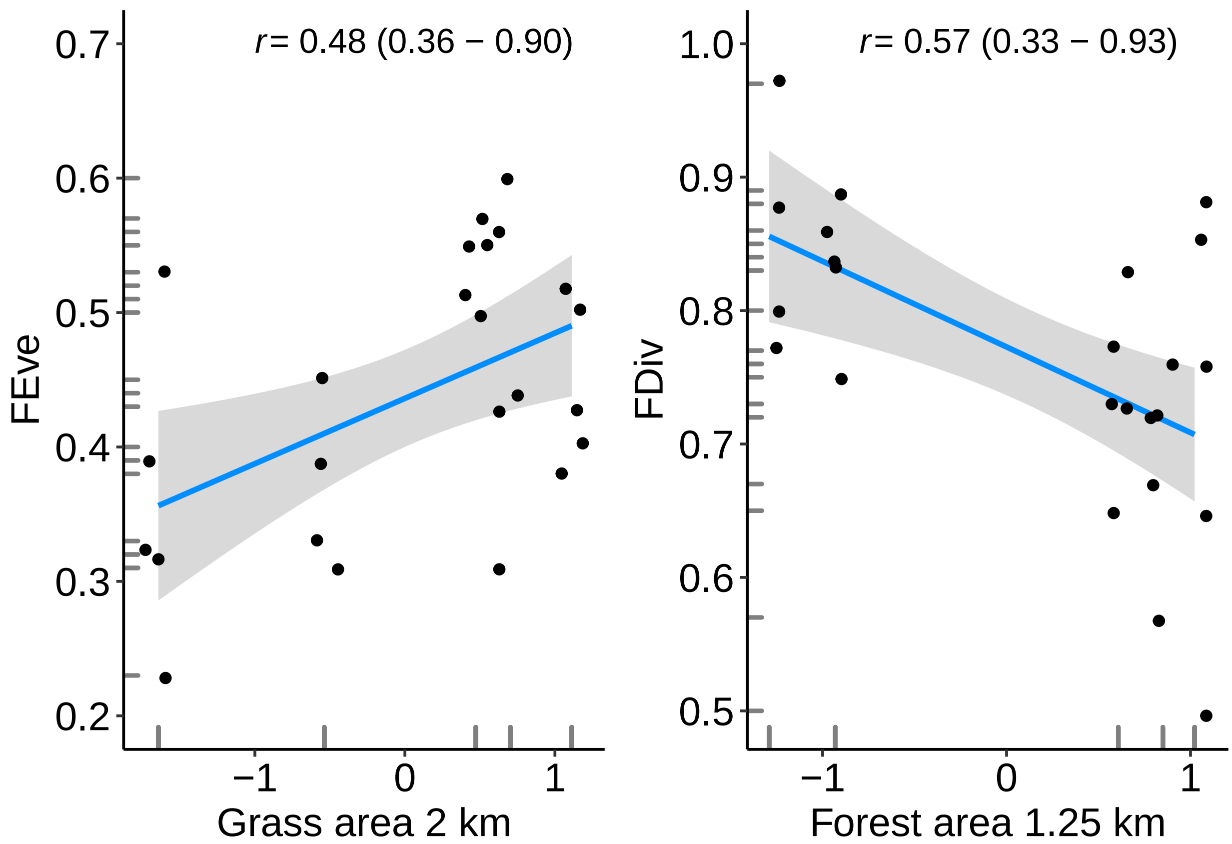
```
conf.limits.nct(3.3,22,0.95)
```

```
##           Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores     1.083297           0.025     5.456433           0.975
```

```
IC_r(0.65,0.21,1.08,5.45)#ES=0.57; IC.i=0.33; IC.s=0.93
```

```
##               r Limite.Inf IC Limite.Sup IC
##       0.5716700      0.3318181      0.9374002
```

```
i2 <- visreg(mfdiv3, xvar="forest.perc.125",type="conditional",
       scale="response",xlab="forest.perc.125",rug=1, cex=1.7,
       ylab="FDiv", gg=T)+geom_jitter() + labs(x="Forest area 1.25 km", y="FDiv") +
  theme_classic() + theme(axis.title = element_text(size = 15, colour = "black"),axis.text = element_te
  annotate("text", x = -0.8, y = 1, label = expression(paste(italic("r"), "= 0.57 (0.33 - 0.93)")), size
  scale_x_continuous(breaks = c(-1,0,1))
#ggsave("fdiv_forest125c.tiff", width = 15, height = 15, units = "cm", dpi = 300)

library(cowplot)
plot_grid(i1,i2, ncol=2, align = "h")
```




```
#ggsave("models_fdiv-feve.tiff", height = 15, width = 15, units = "cm", dpi= 300)


#number of functional groups----

noGF <- read.csv("data.FR.csv", header = T, sep = ";", dec = ",")
str(noGF)
```

```
## 'data.frame':    26 obs. of  52 variables:
##  $ Fecha                    : chr  "1/21/2016" "12-Mar" "20-Apr" "30-Jun" ...
##  $ no.FG                    : int  4 7 4 5 4 5 5 5 6 4 ...
##  $ site                     : chr  "Balmoral" "Balmoral" "Balmoral" "Balmoral" ...
##  $ Carollia.perspicillata   : int  11 12 6 4 4 5 6 0 7 6 ...
##  $ Carollia.castanea        : int  1 3 1 1 0 0 0 3 0 0 ...
```

```
##  $ Artibeus.lituratus       : int  1 3 0 0 0 0 4 2 0 2 ...
##  $ Artibeus.planirostris    : int  2 4 3 9 3 12 11 34 9 14 ...
##  $ Sturnira.lilium          : int  0 0 0 3 1 7 13 4 2 9 ...
##  $ Platyrrhinus.sp.1        : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Platyrrhinus.sp.2        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Uroderma.bilobatum       : int  0 2 1 0 0 1 1 0 0 2 ...
##  $ Mesophylla.maconnelli    : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Desmodus.rotundus        : int  3 9 0 4 0 0 0 0 0 0 ...
##  $ Glossophaga.soricina     : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ Anoura.sp.               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Phyllostomus.discolor    : int  0 0 0 0 0 3 0 1 2 0 ...
##  $ Phyllostomus.elongatus   : int  0 0 2 2 3 0 0 0 1 0 ...
##  $ Phyllostomus.hastatus    : int  0 0 0 0 0 0 0 0 2 0 ...
##  $ Gardnerycteris.crenulatum: int  0 1 0 1 0 3 9 0 2 0 ...
##  $ Tonatia.saurophila       : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Micronicteris.sp.        : int  0 1 0 1 1 0 0 0 0 0 ...
##  $ Lophostoma.brasiliense   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Rhinophylla.sp           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ S                        : int  5 10 5 9 5 6 6 6 8 5 ...
##  $ forest.perc.05           : num  40.8 40.8 40.8 40.8 40.8 ...
##  $ grass.perc.05            : num  25.4 25.4 25.4 25.4 25.4 ...
##  $ ba.perce.05              : num  33.8 33.8 33.8 33.8 33.8 ...
##  $ mps.05                   : num  1.68 1.68 1.68 1.68 1.68 0.98 0.98 0.98 0.98 0.98 ...
##  $ pssd.05                  : num  6.6 6.6 6.6 6.6 6.6 4.13 4.13 4.13 4.13 4.13 ...
##  $ p.no.05                  : int  19 19 19 19 19 32 32 32 32 32 ...
##  $ patchD.05                : num  0.24 0.24 0.24 0.24 0.24 0.41 0.41 0.41 0.41 0.41 ...
##  $ forest.perc.125          : num  37.1 37.1 37.1 37.1 37.1 ...
##  $ grass.perc.125           : num  43.1 43.1 43.1 43.1 43.1 ...
##  $ ba.perce.125             : num  19.9 19.9 19.9 19.9 19.9 ...
##  $ p.no.125                 : int  117 117 117 117 117 322 322 322 322 322 ...
##  $ mps.125                  : num  1.53 1.53 1.53 1.53 1.53 0.26 0.26 0.26 0.26 0.26 ...
##  $ pssd.125                 : num  8.17 8.17 8.17 8.17 8.17 1.6 1.6 1.6 1.6 1.6 ...
##  $ patchD.125               : num  0.24 0.24 0.24 0.24 0.24 0.66 0.66 0.66 0.66 0.66 ...
##  $ forest.perc.2            : num  30 30 30 30 30 ...
##  $ grass.perc.2             : num  57.9 57.9 57.9 57.9 57.9 ...
##  $ ba.perce.2               : num  12.1 12.1 12.1 12.1 12.1 9.16 9.16 9.16 9.16 9.16 ...
##  $ p.no.2                   : int  293 293 293 293 293 681 681 681 681 681 ...
##  $ mps.2                    : num  1.25 1.25 1.25 1.25 1.25 0.35 0.35 0.35 0.35 0.35 ...
##  $ pssd.2                   : num  9.5 9.5 9.5 9.5 9.5 3.16 3.16 3.16 3.16 3.16 ...
##  $ patchD.2                 : num  0.23 0.23 0.23 0.23 0.23 0.54 0.54 0.54 0.54 0.54 ...
##  $ CWM.body_mass            : num  27.4 28.9 26.1 31.4 31.3 ...
##  $ CWM.wing_span            : num  35.5 36.1 35 37.1 37.4 ...
##  $ CWM.wing_loading         : num  12.5 12.5 12.1 13.9 12.8 ...
##  $ CWM.aspect_ratio         : num  6.51 6.48 6.37 6.65 6.44 6.6 6.53 6.55 6.46 6.43 ...
##  $ FRic                     : num  0.42 0.93 0.26 0.06 0.06 0.3 0.67 0.63 0.08 0.02 ...
##  $ FEve                     : num  0.5 0.55 0.55 0.57 0.51 0.38 0.52 0.4 0.5 0.43 ...
##  $ FDiv                     : num  0.73 0.77 0.65 0.73 0.83 0.84 0.83 0.75 0.86 0.89 ...
```

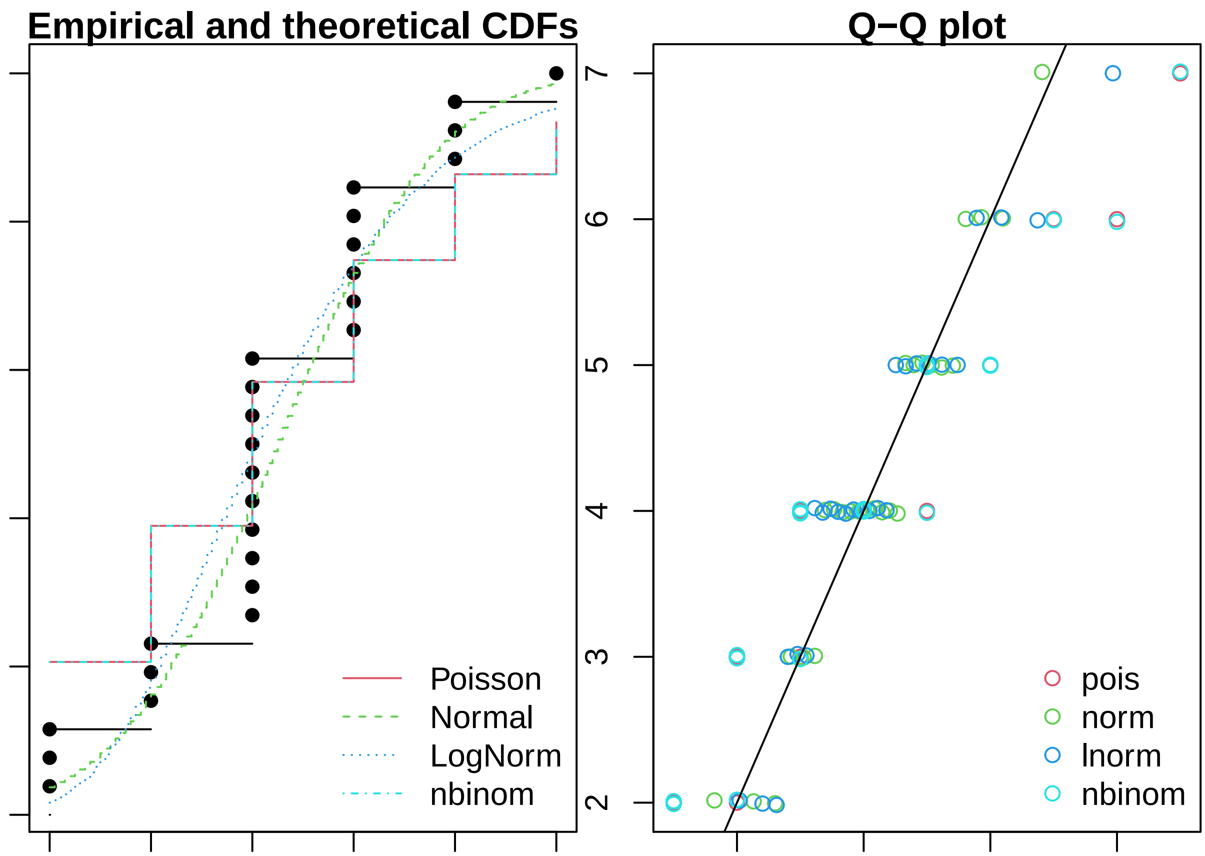
```
str(DF)
```

```
## 'data.frame':    26 obs. of  50 variables:
##  $ site                     : chr  "Balmoral" "Balmoral" "Balmoral" "Balmoral" ...
##  $ Carollia.perspicillata   : num  11 12 6 4 4 5 6 0 7 6 ...
##  $ Carollia.castanea        : num  1 3 1 1 0 0 0 3 0 0 ...
```

```
##  $ Artibeus.lituratus       : num  1 3 0 0 0 0 4 2 0 2 ...
##  $ Artibeus.planirostris    : num  2 4 3 9 3 12 11 34 9 14 ...
##  $ Sturnira.lilium          : num  0 0 0 3 1 7 13 4 2 9 ...
##  $ Platyrrhinus.sp1         : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Platyrrhinus.sp2         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Uroderma.bilobatum       : num  0 2 1 0 0 1 1 0 0 2 ...
##  $ Mesophylla.maconnelli    : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Desmodus.rotundus        : num  3 9 0 4 0 0 0 0 0 0 ...
##  $ Glossophaga.soricina     : num  0 0 0 1 0 0 0 1 0 0 ...
##  $ Anoura.sp                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Phyllostomus.discolor    : num  0 0 0 0 0 3 0 1 2 0 ...
##  $ Phyllostomus.elongatus   : num  0 0 2 2 3 0 0 0 1 0 ...
##  $ Phyllostomus.hastatus    : num  0 0 0 0 0 0 0 0 2 0 ...
##  $ Gardnerycteris.crenulatum: num  0 1 0 1 0 3 9 0 2 0 ...
##  $ Tonatia.saurophila       : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Micronicteris.sp         : num  0 1 0 1 1 0 0 0 0 0 ...
##  $ Lophostoma.brasiliense   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Rhinophylla.sp           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ S                        : num  5 10 5 9 5 6 6 6 8 5 ...
##  $ forest.perc.05           : num  0.613 0.613 0.613 0.613 0.613 ...
##  $ grass.perc.05            : num  -0.684 -0.684 -0.684 -0.684 -0.684 ...
##  $ ba.perce.05              : num  33.8 33.8 33.8 33.8 33.8 ...
##  $ mps.05                   : num  1.68 1.68 1.68 1.68 1.68 0.98 0.98 0.98 0.98 0.98 ...
##  $ pssd.05                  : num  6.6 6.6 6.6 6.6 6.6 4.13 4.13 4.13 4.13 4.13 ...
##  $ p.no.05                  : num  19 19 19 19 19 32 32 32 32 32 ...
##  $ patchD.05                : num  0.24 0.24 0.24 0.24 0.24 0.41 0.41 0.41 0.41 0.41 ...
##  $ forest.perc.125          : num  0.608 0.608 0.608 0.608 0.608 ...
##  $ grass.perc.125           : num  43.1 43.1 43.1 43.1 43.1 ...
##  $ ba.perce.125             : num  19.9 19.9 19.9 19.9 19.9 ...
##  $ p.no.125                 : num  117 117 117 117 117 322 322 322 322 322 ...
##  $ mps.125                  : num  1.53 1.53 1.53 1.53 1.53 0.26 0.26 0.26 0.26 0.26 ...
##  $ pssd.125                 : num  8.17 8.17 8.17 8.17 8.17 1.6 1.6 1.6 1.6 1.6 ...
##  $ patchD.125               : num  0.24 0.24 0.24 0.24 0.24 0.66 0.66 0.66 0.66 0.66 ...
##  $ forest.perc.2            : num  30 30 30 30 30 ...
##  $ grass.perc.2             : num  0.473 0.473 0.473 0.473 0.473 ...
##  $ ba.perce.2               : num  12.1 12.1 12.1 12.1 12.1 9.16 9.16 9.16 9.16 9.16 ...
##  $ p.no.2                   : num  293 293 293 293 293 681 681 681 681 681 ...
##  $ mps.2                    : num  1.25 1.25 1.25 1.25 1.25 0.35 0.35 0.35 0.35 0.35 ...
##  $ pssd.2                   : num  9.5 9.5 9.5 9.5 9.5 3.16 3.16 3.16 3.16 3.16 ...
##  $ patchD.2                 : num  0.23 0.23 0.23 0.23 0.23 0.54 0.54 0.54 0.54 0.54 ...
##  $ CWM.body_mass            : num  27.4 28.9 26.1 31.4 31.3 ...
##  $ CWM.wing_span            : num  35.5 36.1 35 37.1 37.4 ...
##  $ CWM.wing_loading         : num  12.5 12.5 12.1 13.9 12.8 ...
##  $ CWM.aspect_ratio         : num  6.51 6.48 6.37 6.65 6.44 6.6 6.53 6.55 6.46 6.43 ...
##  $ FRic                     : num  0.42 0.93 0.26 0.06 0.06 0.3 0.67 0.63 0.08 0.02 ...
##  $ FEve                     : num  0.5 0.55 0.55 0.57 0.51 0.38 0.52 0.4 0.5 0.43 ...
##  $ FDiv                     : num  0.73 0.77 0.65 0.73 0.83 0.84 0.83 0.75 0.86 0.89 ...
```

```r
DF$GF <- noGF$no.FG

library(fitdistrplus)

pois <- fitdist(DF$GF, "pois")
norm <- fitdist(DF$GF, "norm")
```

```r
lnorm <- fitdist(DF$GF, "lnorm")
nbino <- fitdist(DF$GF, "nbinom")
par(mfrow = c(1, 2))
cdfcomp(list(pois,norm,lnorm,nbino), addlegend = T,
        legendtext = c("Poisson", "Normal","LogNorm","nbinom"))
qqcomp(list(pois,norm,lnorm,nbino))
```




```r
gofstat(list(pois,norm,lnorm,nbino))
```

```
## Chi-squared statistic:  8.238933 3.914509 1.893356 8.238865
## Degree of freedom of the Chi-squared distribution:  2 1 1 1
## Chi-squared p-value:  0.01625319 0.04787098 0.1688239 0.004100276
##    the p-value may be wrong with some theoretical counts < 5
## Chi-squared table:
##      obscounts theo 1-mle-pois theo 2-mle-norm theo 3-mle-lnorm
## <= 3         6       10.133008        4.221825         4.800358
## <= 4        10        5.047075        6.873852         7.989854
## <= 5         6        4.270602        7.907039         6.593835
## > 5          4        6.549315        6.997285         6.615953
##      theo 4-mle-nbinom
## <= 3         10.132131
## <= 4          5.047024
## <= 5          4.270744
## > 5           6.550101
##
```

```
## Goodness-of-fit criteria
##                                 1-mle-pois 2-mle-norm 3-mle-lnorm 4-mle-nbinom
## Akaike's Information Criterion   97.12181   89.38211    92.21729     99.12181
## Bayesian Information Criterion   98.37990   91.89831    94.73348    101.63800
```

```r
par(mfrow = c(1, 1))

FGm1 <- glm(GF~forest.perc.05 + grass.perc.05 +
            forest.perc.125 + grass.perc.2, data=DF)
summary(FGm1)
```

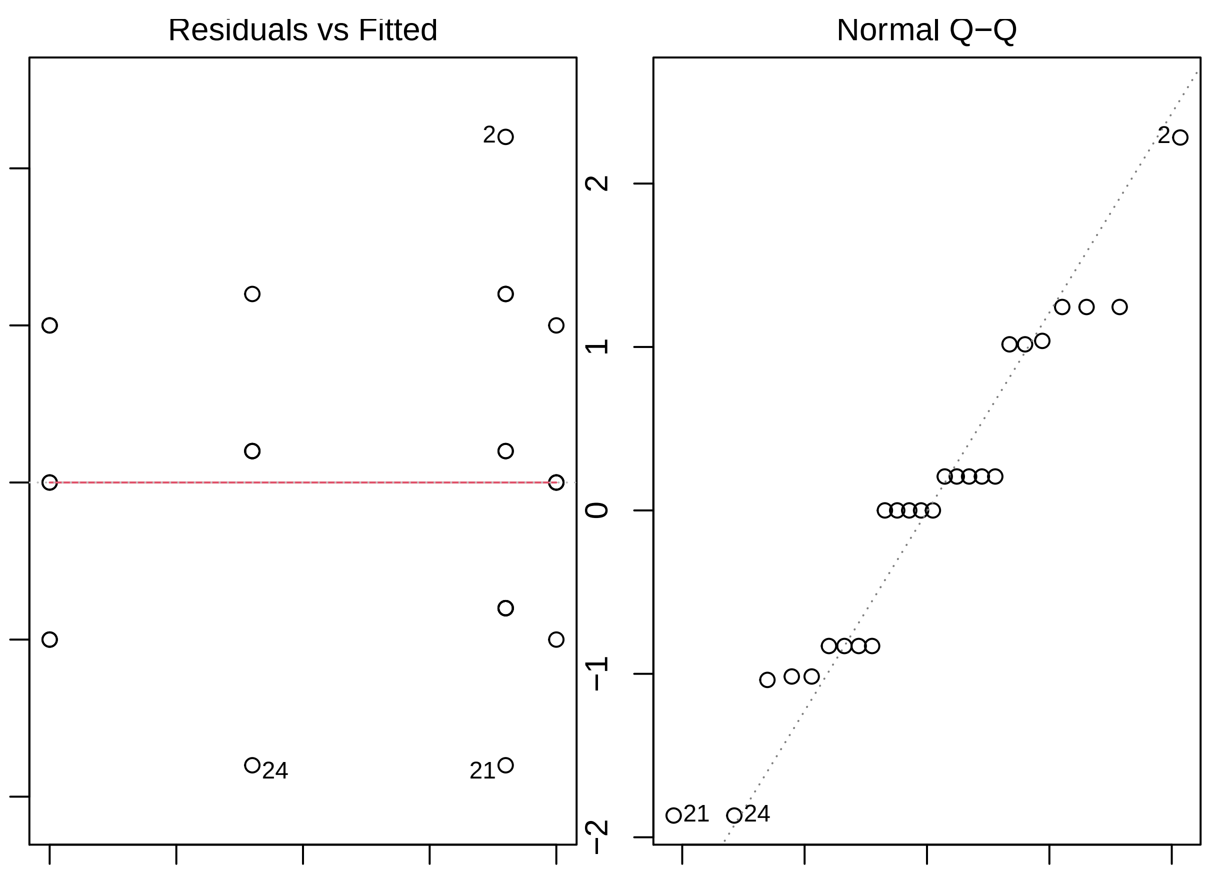
```
##
## Call:
## glm(formula = GF ~ forest.perc.05 + grass.perc.05 + forest.perc.125 +
##     grass.perc.2, data = DF)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##   -1.8    -0.8     0.0     0.8     2.2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.23077    0.21140  20.013  3.7e-15 ***
## forest.perc.05     0.48412    0.67671   0.715    0.482
## grass.perc.05      0.01776    0.62253   0.029    0.978
## forest.perc.125    0.11705    0.32916   0.356    0.726
## grass.perc.2       0.45131    0.59575   0.758    0.457
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.161905)
##
##     Null deviance: 40.615  on 25  degrees of freedom
## Residual deviance: 24.400  on 21  degrees of freedom
## AIC: 84.133
##
## Number of Fisher Scoring iterations: 2
```

```r
anova(FGm1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: GF
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               25     40.615
## forest.perc.05   1  12.0197        24     28.596 0.001298 **
## grass.perc.05    1   3.5250        23     25.071 0.081547 .
```

```
## forest.perc.125  1   0.0039         22      25.067 0.953895
## grass.perc.2     1   0.6668         21      24.400 0.448725
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#residual analysis
par(mfrow=c(1,2))
plot(FGm1, which=c(1,2))
```




```
par(mfrow=c(1,1))

#likelihood ratio test
FGm2 <- update(FGm1,~.-grass.perc.2)
anova(FGm1,FGm2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GF ~ forest.perc.05 + grass.perc.05 + forest.perc.125 + grass.perc.2
## Model 2: GF ~ forest.perc.05 + grass.perc.05 + forest.perc.125
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        21     24.400
## 2        22     25.067 -1 -0.66678   0.4487
```

```r
FGm3 <- update(FGm2, ~.-forest.perc.125)
anova(FGm2,FGm3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GF ~ forest.perc.05 + grass.perc.05 + forest.perc.125
## Model 2: GF ~ forest.perc.05 + grass.perc.05
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1        22     25.067
## 2        23     25.071 -1 -0.0038839   0.9534
```

```r
FGm4 <- update(FGm3,~.-grass.perc.05)
anova(FGm3,FGm4, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GF ~ forest.perc.05 + grass.perc.05
## Model 2: GF ~ forest.perc.05
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        23     25.071
## 2        24     28.596 -1   -3.525  0.07213 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
FGm5 <- update(FGm4, ~.-forest.perc.05)
anova(FGm4,FGm5,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GF ~ forest.perc.05
## Model 2: GF ~ 1
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        24     28.596
## 2        25     40.615 -1   -12.02 0.001492 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(FGm4)
```

```
##
## Call:
## glm(formula = GF ~ forest.perc.05, data = DF)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.57405  -0.65613  -0.00832   0.55993   2.34387
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.2308     0.2141  19.763 2.35e-16 ***
## forest.perc.05    0.6934     0.2183   3.176  0.00407 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.191485)
##
##     Null deviance: 40.615  on 25  degrees of freedom
## Residual deviance: 28.596  on 24  degrees of freedom
## AIC: 82.259
##
## Number of Fisher Scoring iterations: 2
```

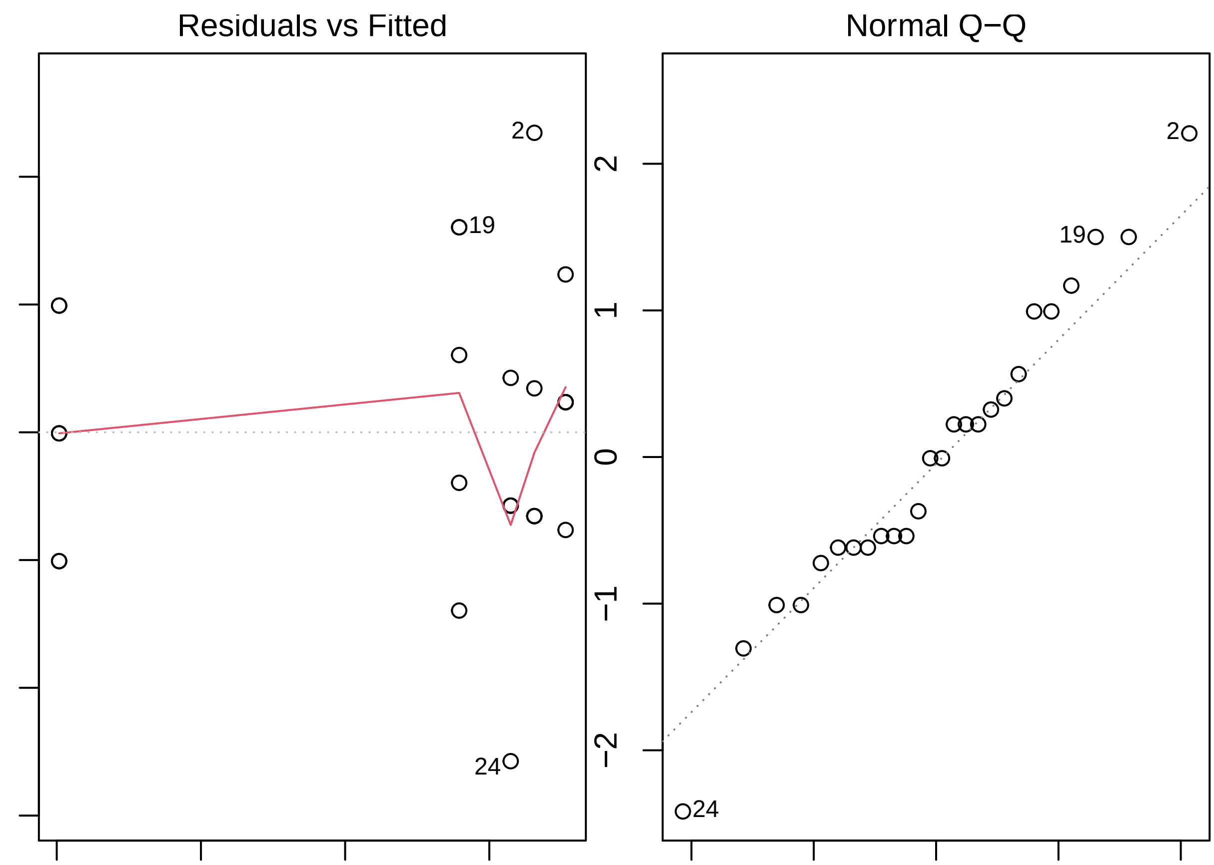
```
confint(FGm4)
```

```
##                    2.5 %   97.5 %
## (Intercept)    3.8111979 4.650341
## forest.perc.05 0.2655094 1.121270
```

```
anova(FGm4, test = "F")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: GF
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev      F   Pr(>F)
## NULL                              25     40.615
## forest.perc.05  1    12.02        24     28.596 10.088 0.004069 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#residual analysis
par(mfrow=c(1,2))
plot(FGm4,which=c(1,2))
```




```
par(mfrow=c(1,1))


#size effect

r.se(3.176,24,25)
```

```
##     r transf SE(r transf)
##    0.6097970    0.2132007
```

```
conf.limits.nct(3.176,25,0.95)
```

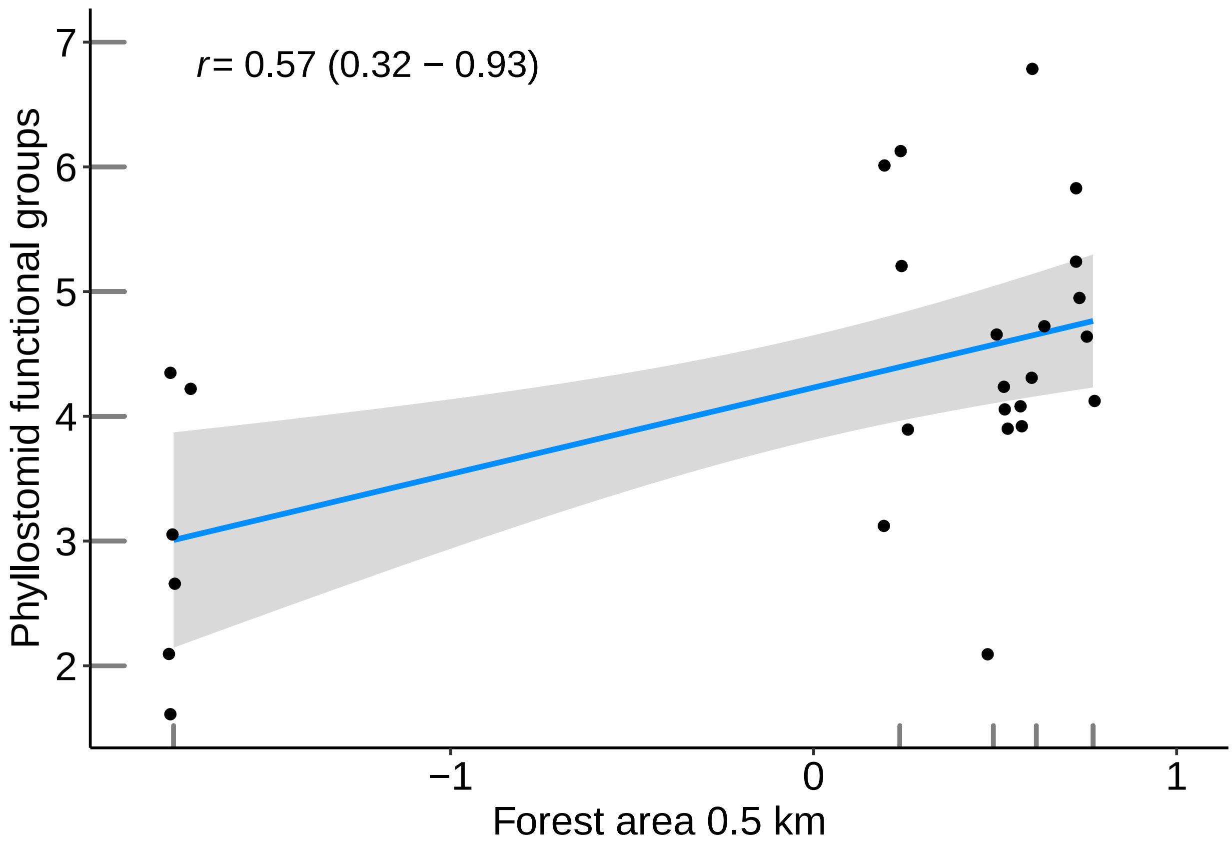
```
##         Cuantil Inf Prob.Limite.Inf Cuantil.Sup Prob.Limite.Sup
## Valores     1.002778      0.02499999    5.296248           0.975
```

```
IC_r(0.609,0.213,1.002,5.29)# r=0.57(0.32-0.93)
```

```
##              r Limite.Inf IC Limite.Sup IC
##     0.5434228     0.3185179     0.9315772
```

```
library(visreg)

visreg(FGm4, xvar = "forest.perc.05", scale = "response", type = "conditional",
```

```
      rug=1, gg=T)+geom_jitter()+theme_classic()+labs(x="Forest area 0.5 km", y="Phyllostomid functiona
  theme(axis.title = element_text(size=15,colour = "black"), axis.text = element_text(size = 15, colour
  annotate("text", x=-1.7, y=6.8, label= expression(paste(italic("r"),"= 0.57 (0.32 - 0.93)")), size = 5
  scale_x_continuous(breaks = c(-1,0,1.5)) + xlim(-1.85,1)
```




```
#ggsave("noFG.tiff", width = 10, height = 15, units = "cm", dpi = 300)

#plot residual analyses all models
#par(mfrow=c(4,2), cex.lab=1.8, mar=c(6,5,3,3))
#plot(CWMm2, which = c(1,2))
#plot(feveM3, which=c(1,2))
#plot(mfdiv3, which=c(1,2))
#plot(FGm4, which = c(1,2))
#par(mfrow=c(1,1))
#dev.off()
```