
BSS
BIAŁOWIESKA SZKOŁA STATYSTYKI

Regresja – uogólnione modele liniowe i addytywne

Krótkie przypomnienie

modele liniowe y=ax+b *lm(y~x, data=dataset)*

modele mogą mieć wiele czynników

modele mogą być funkcjami liniowymi innymi wielomian 1. stopnia, np. parabole:

y=ax^2+bx+c *lm(y~poly(x,2),dataset)*, y=a*log(x)+b *lm(y~log(x), dataset)*

modele mogą mieć wiele zmiennych

y=ax+by+cz+d *lm(y~a+b+c, dataset)*

GLM

uogólniony model liniowy - generalized linear model

dlaczego uogólniony?

model liniowy zakłada rozkład normalny, GLM uogólnia metodę na inne rozkłady

po co? do innych rozkładów

(przypominamy sobie przykłady)

GLM z innymi rozkładami

glm(formula, data, family=**’rozklad’**)

family=’poisson’ - rozkład Poissona (dyskretny - liczby naturalne - liczba osobników, liczba gatunków)

family=binomial(link=’logit’) - regresja logistyczna (0/1 - np. przeżycie, występowanie lub brak - klasyfikacja binarna)

Rozkład dwumianowy

- Dane zero-jedynkowe
- Dwa stany – prawdopodobieństwo osiągnięcia jednego
- Model rozmieszczenia gatunku <=> presence i true absence!
- Przeżywalność, sukces lęgowy, obecność

Contents lists available at ScienceDirect

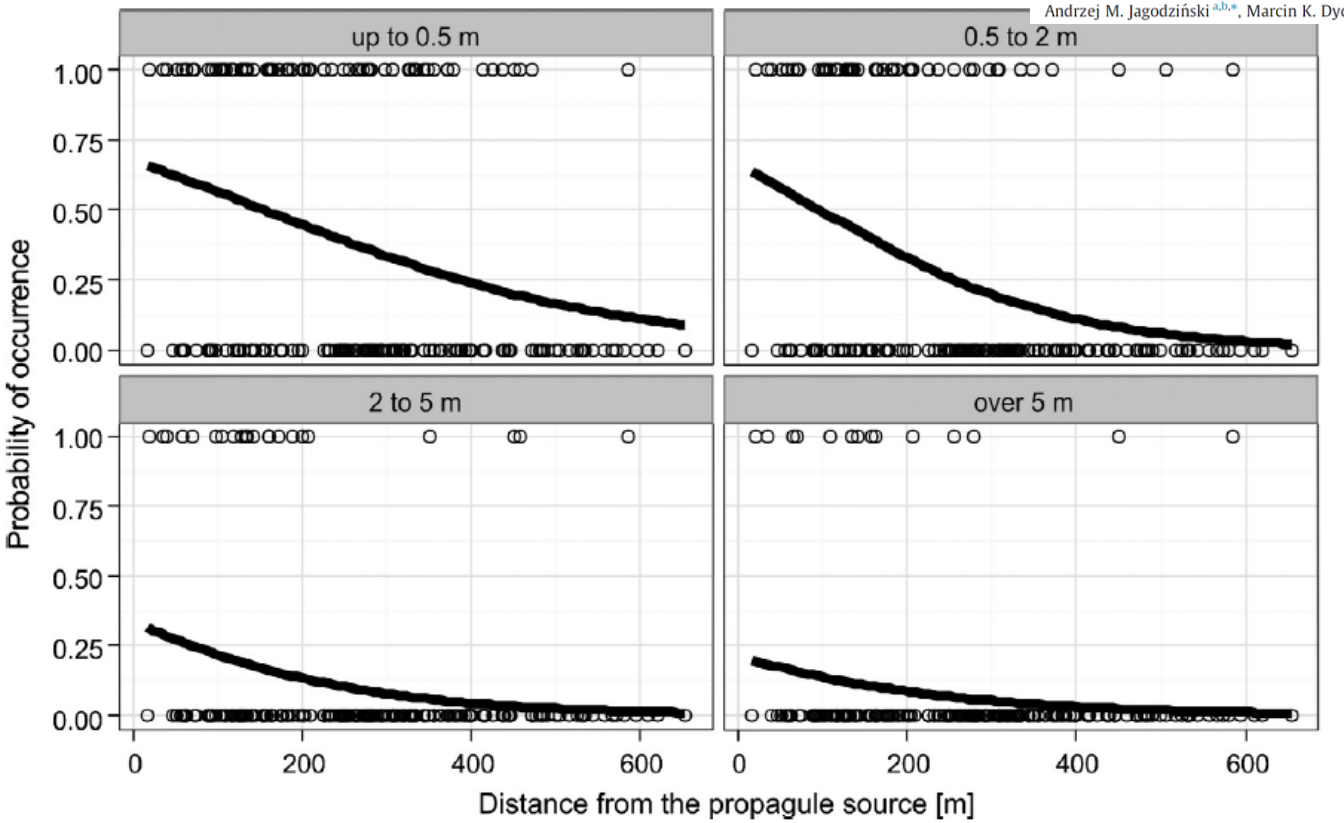
Forest Ecology and Management

journal homepage: www.elsevier.com/locate/foreco

regresja logistyczna

Plantation of coniferous trees modifies risk and size of *Padus serotina* (Ehrh.) Borkh. invasion – Evidence from a Rogów Arboretum case study

Andrzej M. Jagodziński [a,b,*], Marcin K. Dyderski [c], Mateusz Rawlik [d], Piotr Banaszczak [e]



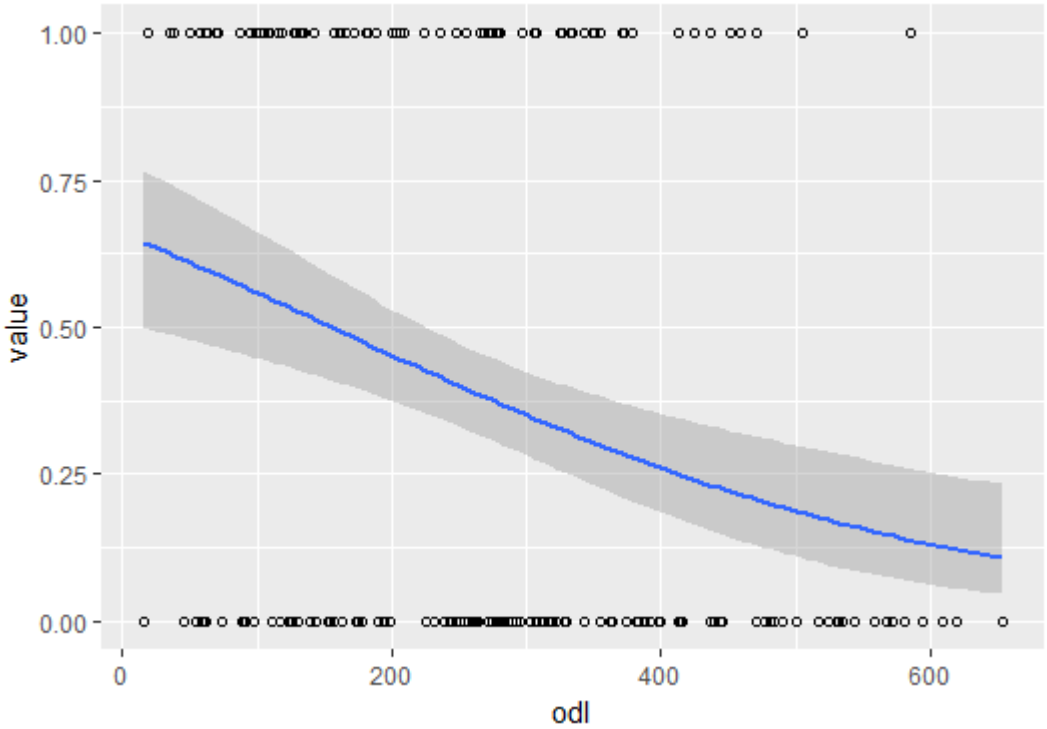
GLM binomial (regresja logistyczna)

model<-glm(szans~odl, family=binomial(link=‚logit’), dane)

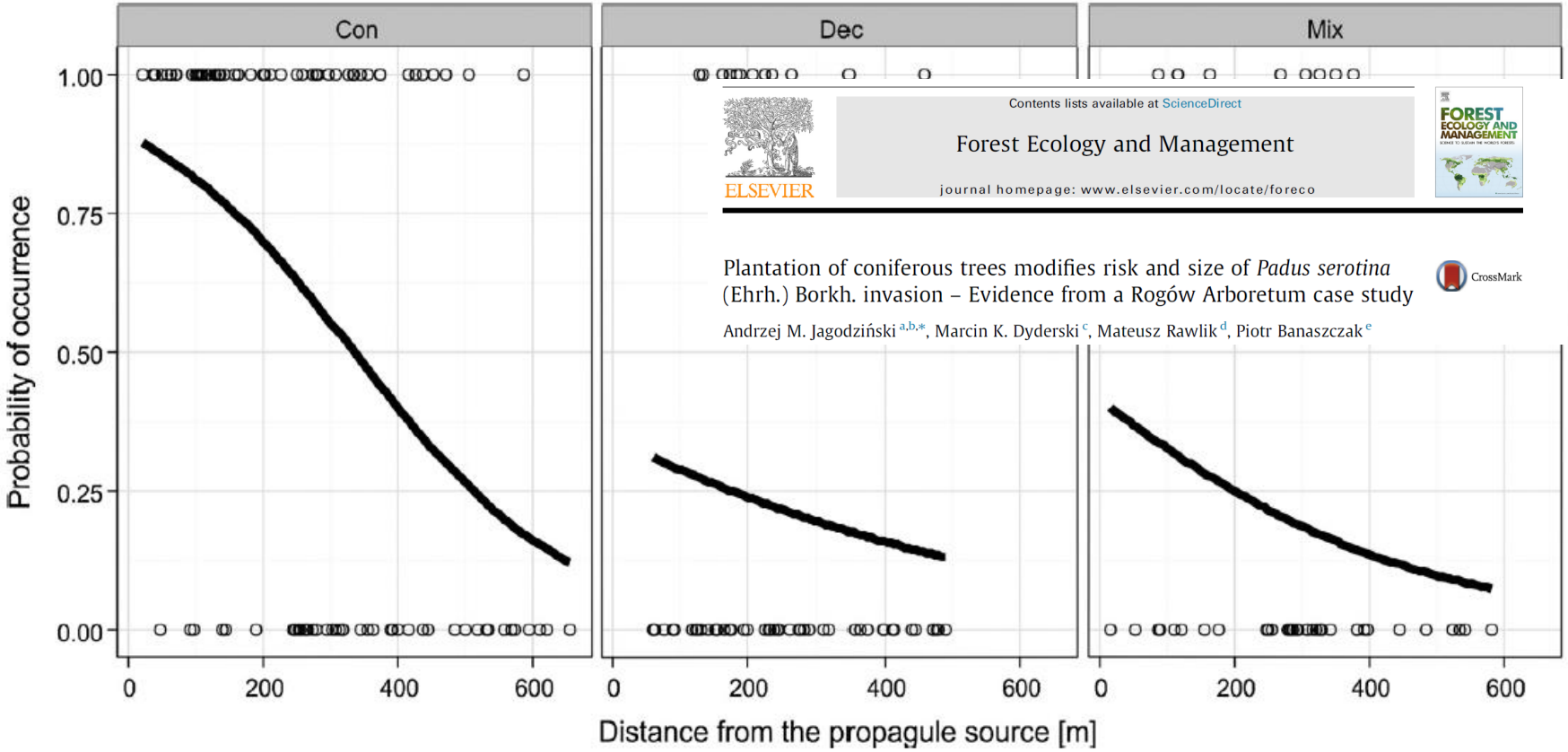
summary (model)

```
Call:
glm(formula = szans ~ odl, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4374  -0.9748  -0.7137   1.1419   1.9901

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.661478   0.318846   2.075 0.038024 *
odl         -0.004263   0.001120  -3.806 0.000141 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 268.48  on 200  degrees of freedom
Residual deviance: 252.23  on 199  degrees of freedom
AIC: 256.23

Number of Fisher Scoring iterations: 4
```

ggplot(data=fig4df, aes(x=odl, y=value))+geom_point(shape=1)
+geom_smooth(method="**glm", method.args=c(family="binomial")**)



Dwie zmienne: typ lasu i odległość glm(szans~odl+typ, family=binomial(link=,logit'), dane)



Contents lists available at ScienceDirect
Forest Ecology and Management
journal homepage: www.elsevier.com/locate/foreco
ELSEVIER

Plantation of coniferous trees modifies risk and size of *Padus serotina* (Ehrh.) Borkh. invasion – Evidence from a Rogów Arboretum case study

Andrzej M. Jagodziński [a,b,*], Marcin K. Dyderski [c], Mateusz Rawlik [d], Piotr Banaszczak [e]

Fig. 4. Effect of coniferous (Con), deciduous (Dec) and mixed (Mix) tree stands on the probability of colonisation by black cherry ($p < 0.001$).

Inny przykład – gatunki starych lasów

- Zbiór danych z poniedziałku (eks)

```
table(eks$stare.lasy)

  0   1
263  49
```

Czy gatunek jest wskaźnikowy dla starych lasów?

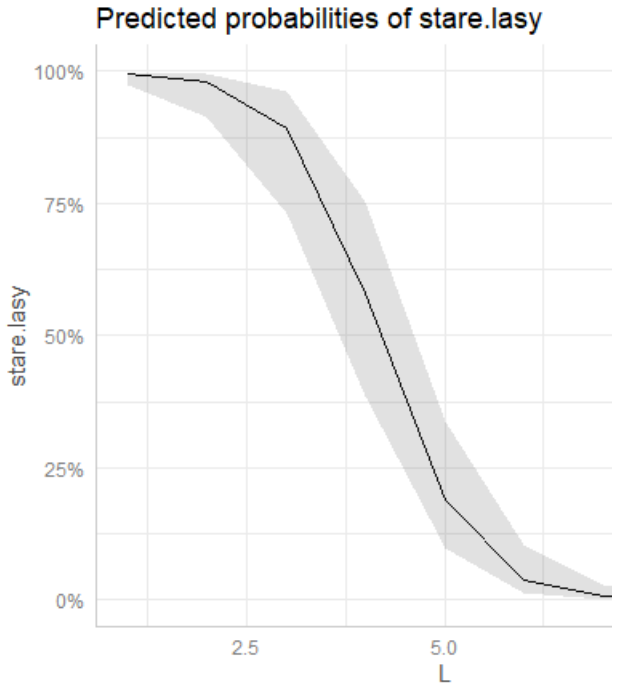
Budujemy model globalny

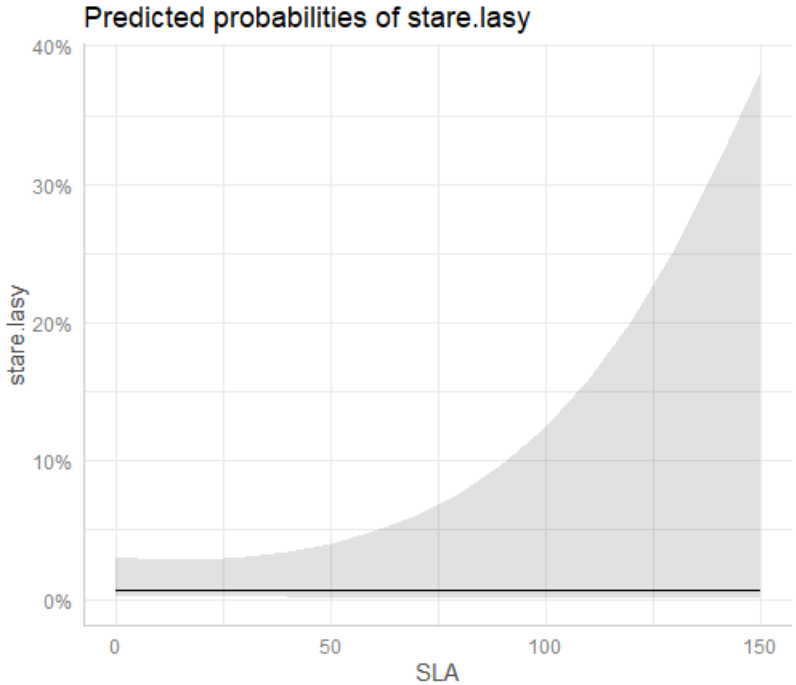
```
mod.sl<-glm(stare.lasy~SLA+seed_mass+L, family=binomial(link='logit'),eks)
car::vif(mod.sl)
      SLA seed_mass         L
 1.108886  1.151267  1.111789

summary(mod.sl)
Call:
glm(formula = stare.lasy ~ SLA + seed_mass + L, family = binomial(link = "logit"),
    data = eks)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6239  -0.2989  -0.1587  -0.0660   3.7668

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.1168346  1.5525297   5.228 1.71e-07 ***
SLA          0.0002634  0.0161591   0.016    0.987
seed_mass   -0.0104889  0.0064559  -1.625    0.104
L           -1.7781285  0.2723928  -6.528 6.67e-11 ***
```

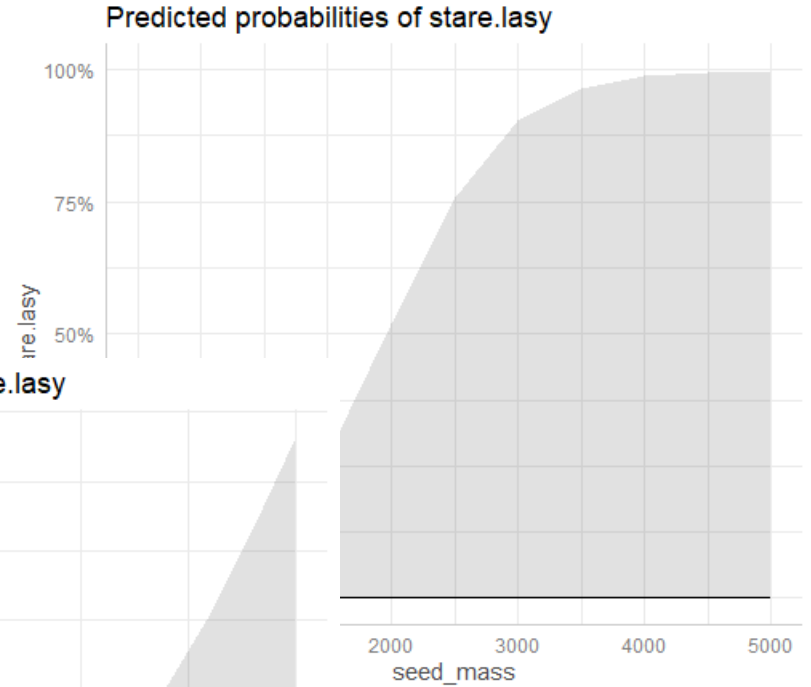
Jak rozumieć effect sizes?

- Względnie – im większy estimate tym większy effect size
- Bezwględnie – trzeba przeliczyć w oparciu o funkcję łączącą
- Najłatwiej – ggpredict()

ggpredict(mod.sl)







Dane zliczeniowe

- Liczby naturalne (>0, całkowite)
- Bogactwo gatunkowe na powierzchni badawczej
- Zagęszczenie odnowienia naturalnego
- Liczba saren na polanie
- Liczba piskląt w gnieździe

- Rozkład Poissona – pierwszy wybór – założenie braku nadyspersji (overdispersion)
- Rozkład dwumianowy ujemny – przy naddyspersji

GLM z rozkładem Poissona –zobaczmy najpierw lm

```
model<-lm(prunusc~richness,data=prunus)
summary(model)
```

```
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-4.5654 -1.7546 -0.7967  0.2663 15.6279

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.61911    1.70456  -0.950   0.3482
richness     0.19327    0.08731   2.214   0.0329 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 17.97445)

    Null deviance: 771.10  on 39  degrees of freedom
Residual deviance: 683.03  on 38  degrees of freedom
AIC: 233.02
```

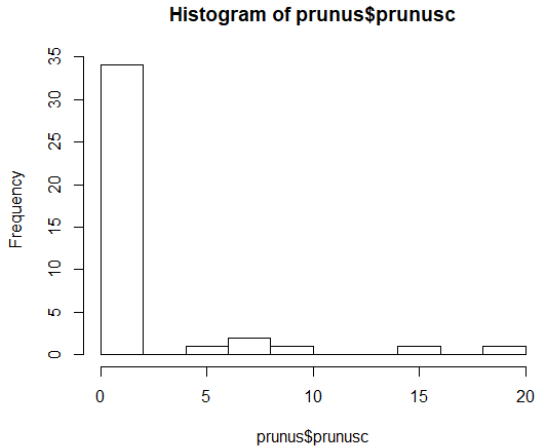
```
model<-glm(prunusc~richness,data=prunus, family=poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3062  -1.5175  -1.1999  -0.7244   8.0114

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.46686    0.38989  -3.762 0.000168 ***
richness     0.09892    0.01544   6.408 1.48e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 248.48  on 39  degrees of freedom
Residual deviance: 203.69  on 38  degrees of freedom
AIC: 244.38

Number of Fisher Scoring iterations: 7
```

zwiększyło się AIC

teoretycznie jest to gorszy model, ale!

do wykonania GLM jesteśmy uprawnieni mając dane o rozkładzie normalnym

logika i założenia modelu> cyferki



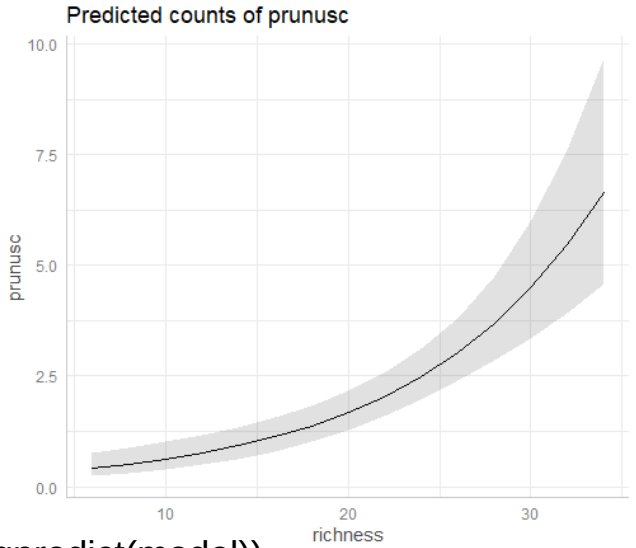
Interpretacja?

ggpredict(model)
$richness
Predicted counts of prunusc

richness	Predicted	95% CI
6	0.42	[0.23, 0.76]
10	0.62	[0.38, 1.01]
12	0.76	[0.49, 1.16]
16	1.12	[0.80, 1.57]
20	1.67	[1.29, 2.16]
24	2.48	[1.97, 3.11]
28	3.68	[2.85, 4.76]
34	6.66	[4.57, 9.70]

Co to znaczy?
W d-stanach z 6 gatunkami w runie oczekiwane zagęszczenie czeremchy to 0,42 szt.
W d-stanach z 28 oczekujemy 3,68 szt. na poletko

przydaje się do opisu wyników



plot(ggpredict(model))

Czy model jest poprawny?

Przy danych zliczeniowych mamy dwa potencjalne problemy:

- dyspersja (stosunek zmienności w danych do zmienności oczekiwanej w modelu)
- wartości zerowe

Rozkład Poissona - rozkład teoretyczny

Rzeczywistość - puste próby, brak występowania

Usunąć? informacja biologiczna - jak wnioskować o korniku nie badając miejsc gdzie go nie ma?

Zostawić - inflacja zer - model mając wiele zer “idzie na łatwiznę” - częściej przewiduje zero niż by wynikało z rzeczywistości

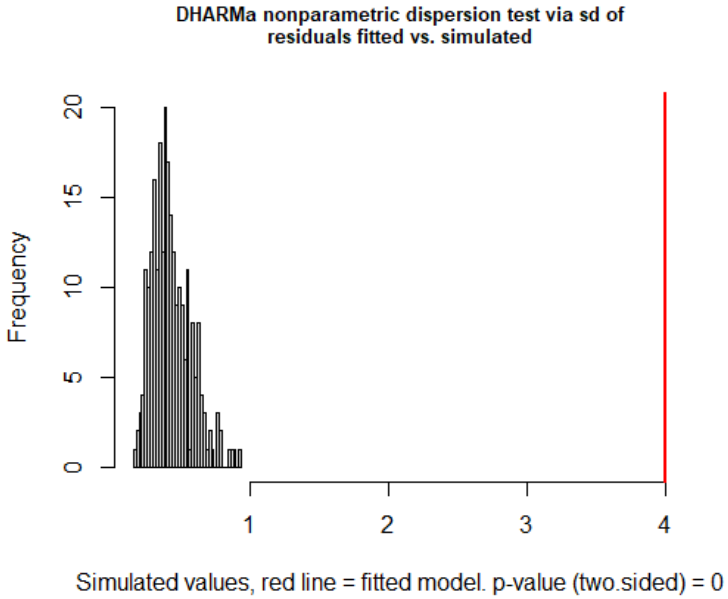
Sprawdzamy – pakiet DHARMa

```
>library(DHARMa)
> model<-glm(prunusc~richness,data=prunus, family=poisson)
> simres<-simulateResiduals(model)
> testDispersion(simres)
```

DHARMa nonparametric dispersion test via sd of r
vs. simulated

data: simulationOutput
dispersion = 9.2819, p-value < 2.2e-16
alternative hypothesis: two.sided

Mamy problem - naddyseprsja

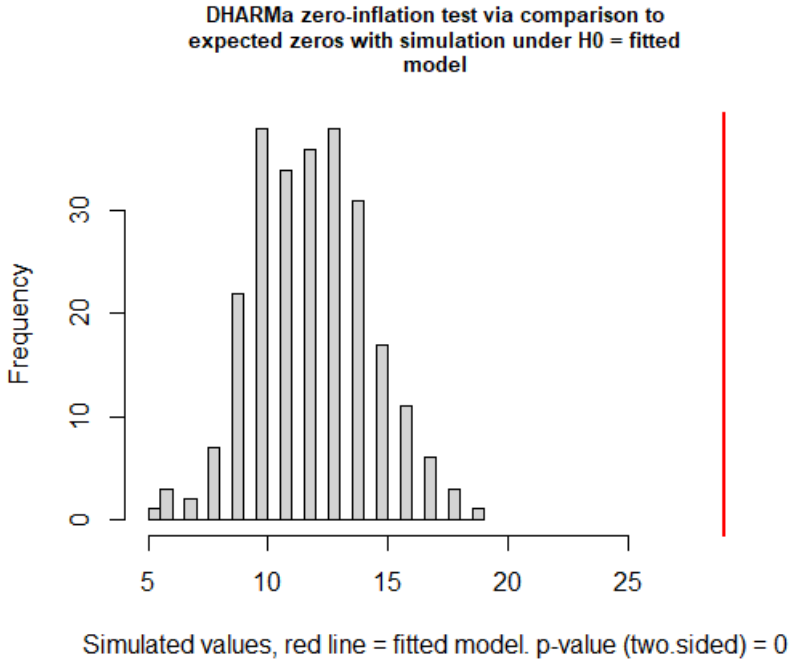


Sprawdzamy – pakiet DHARMa

```
>library(DHARMa)
> model<-glm(prunusc~richness,data=prunus, family=poisson)
> simres<-simulateResiduals(model)
> testZeroInflation(simres)
```

```
        DHARMa zero-inflation test via comparison to expe
        simulation under H0 = fitted model

data:  simulationOutput
ratioObsSim = 2.4078, p-value < 2.2e-16
alternative hypothesis: two.sided
```

Mamy problem – za dużo zer!



Problem nr 1 - naddyspersja

- Zamiast Poissona wybieramy rozkład negative binomial, lepiej przejść na :

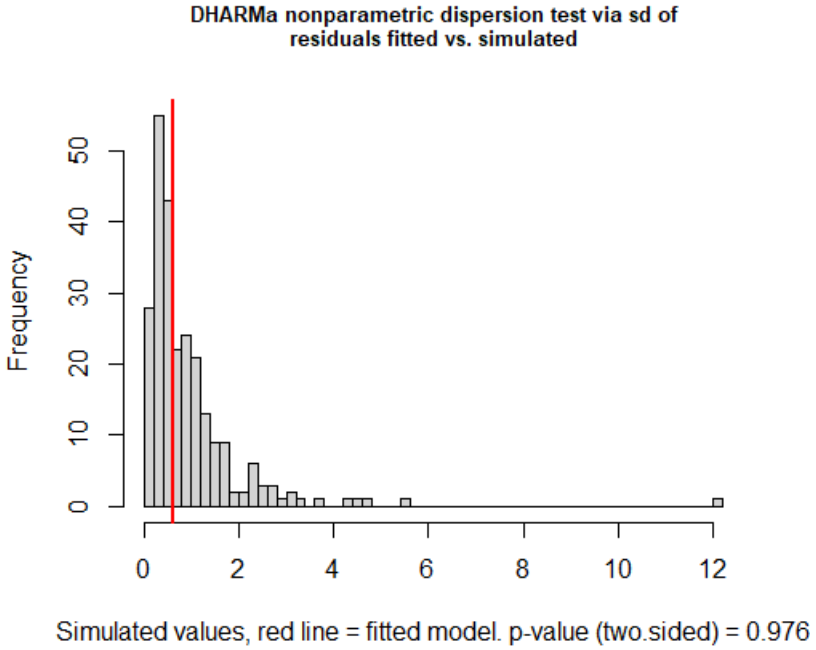
```
library(glmmTMB)
> model.nb<-glmmTMB(prunusc~richness,data=prunus, family=nbinom1())
> summary(model.nb)
 Family: nbinom1  ( log )
Formula:          prunusc ~ richness
Data: prunus
     AIC      BIC   logLik deviance df.resid
   111.2    116.3    -52.6    105.2       37
Dispersion parameter for nbinom1 family (): 13.2
Conditional model:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.11077    1.00707  -1.103   0.2700
richness     0.08387    0.03905   2.148   0.0317 *
```

Sprawdzamy problem z dyspersją

```
model.nb<-glmmTMB(prunusc~richness,data=prunus, family=nbinom1())
simres<-simulateResiduals(model.nb)
testDispersion(simres)
```

```
        DHARMa nonparametric dispersic
        vs. simulated

data:  simulationOutput
dispersion = 0.67144, p-value = 0.976
alternative hypothesis: two.sided
```

Jest dobrze, nie ma problemu

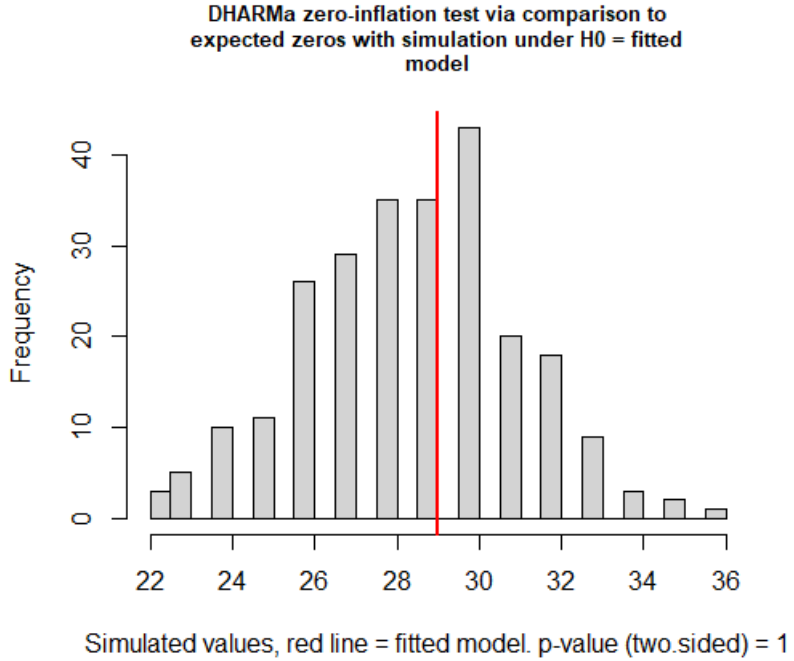


testZeroInflation(simres)

DHARMa zero-inflation test via comparison to expected zeros with
simulation under H0 = fitted model

data: simulationOutput
ratioObsSim = 1.0153, p-value = 1
alternative hypothesis: two.sided

Mimo iż nic nie zrobiliśmy z zerami, to inny rozkład pozwolił też ominąć ten problem!

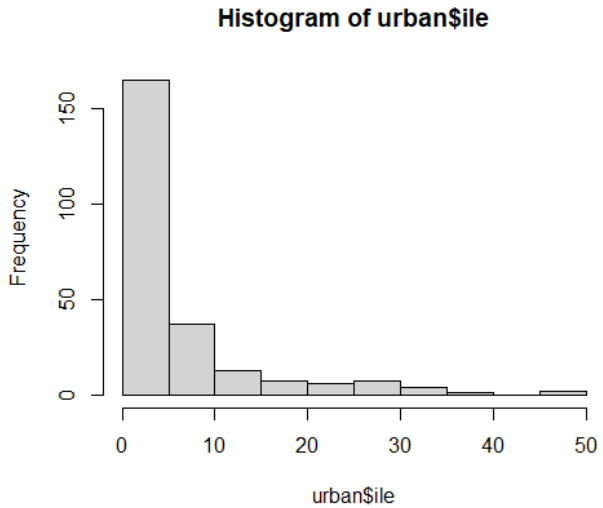


Problem z zerami – potrzeba modelu zero-inflated

Tzw. modele złożone (hurdle models) - składają się z dwóch części:

1. modeluje prawdopodobieństwo wystąpienia (binomial) - czy w ogóle coś będzie
2. modeluje liczebność - czyli właściwy Poisson (lub negative binomial albo beta)

```
hist(urban$ile)
length(which(urban$ile>0))
187 #na 242
```



```
mod.u<-glmmTMB(ile~Forests+Water, ziformula = ~Forests, urban, family=poisson())
> summary(mod.u)
 Family: poisson  ( log )
Formula:          ile ~ Forests + Water
Zero inflation:       ~Forests
Data: urban

     AIC      BIC   logLik deviance df.resid
  2053.4   2070.8  -1021.7   2043.4      237


Conditional model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.5408407  0.0411004   37.49  < 2e-16 ***
Forests     0.0160141  0.0008578   18.67  < 2e-16 ***
Water       0.0164942  0.0028167    5.86 4.74e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Zero-inflation model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.86517    0.17745  -4.876 1.08e-06 ***
Forests     -0.03498    0.01099  -3.182  0.00146 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

ziformula – definiujemy zmienne wpływające na 0/1
działa na Poissona, beta, negative binomial

ggpredict(mod.u)
$Forests
Predicted counts of ile

Forests	Predicted	95% CI
0	4.91	[4.54, 5.30]
20	6.76	[6.38, 7.16]
40	9.31	[8.82, 9.83]
60	12.83	[11.96, 13.75]
80	17.67	[16.07, 19.43]
100	24.34	[21.50, 27.56]

Adjusted for:
* Water = 3.03

$Water
Predicted counts of ile

Water	Predicted	95% CI
0	6.04	[5.66, 6.45]
10	7.12	[6.66, 7.62]
20	8.40	[7.56, 9.33]
30	9.90	[8.50, 11.55]
50	13.78	[10.64, 17.84]
60	16.25	[11.89, 22.20]
70	19.16	[13.28, 27.65]
90	26.65	[16.56, 42.89]

Adjusted for:
* Forests = 16.07

attr(,"class")
[1] "ggalleffects" "list"
attr(,"model.name")
[1] "mod.u"

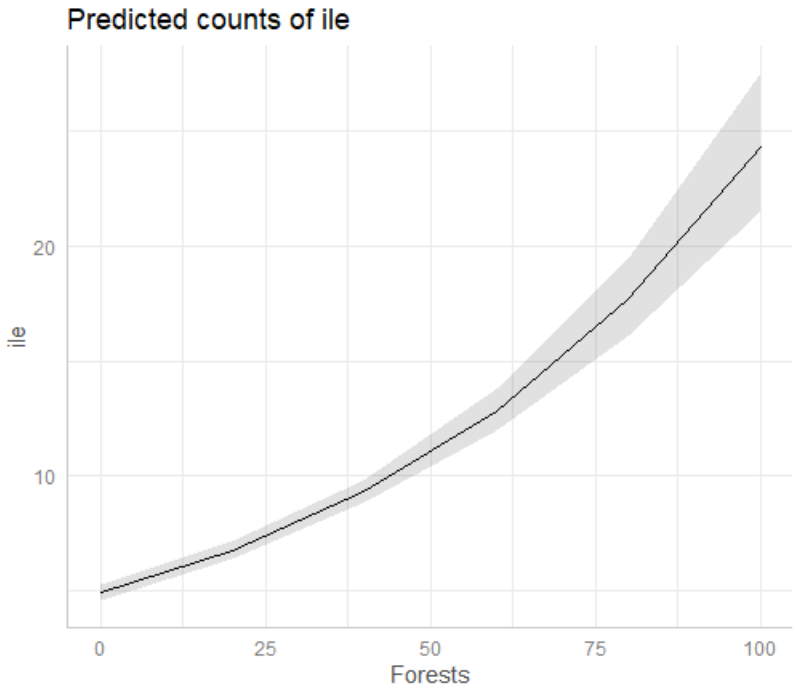
Raportowanie w publikacji

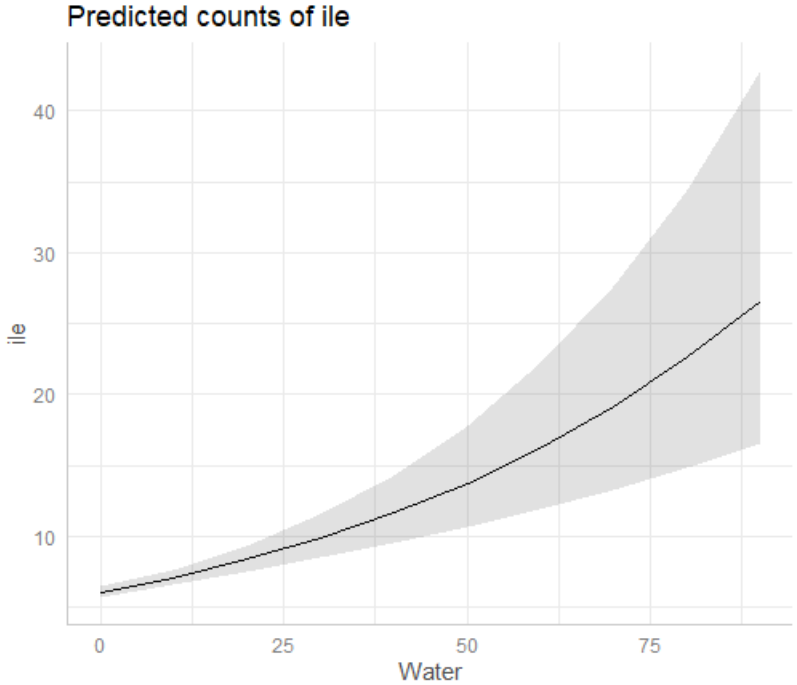
Table 2
Generalized Linear Mixed-Effects Models explaining proportion of browsed and damaged trees proportion and density of *F. excelsior* natural regeneration within study plots ($n = 32$).

Response	Variable	Estimate	SE	z	Pr(>|z|)	Random effect and AICc
Proportion of browsed trees	(Intercept)	−0.2723	0.1315	−2.071	0.0384	Random effect SD = 0.0452
	Density	0.0038	0.0013	2.907	0.0037	AICc = −73.2
	Shrub cover	−0.5302	0.2453	−2.161	0.0307	$AICc_0 = -50.1$, $AICc_g = -61.7$
	$CaCO_3$ content	−0.0251	0.0100	−2.516	0.0119	
Proportion of damaged trees	Count: (Intercept)	−9.0370	1.9436	−4.650	<0.0001	Random effect (count) SD = 0.2644
	Count: Summer groundwater table level	−0.3997	0.2160	−1.850	0.0643	Random effect (zero-inflation) SD = 0.0512
	Count: Soil pH	1.0719	0.2356	4.550	<0.0001	AICc = −96.9
	Zero-inflation: (Intercept)	−3.4350	1.0350	−3.318	0.0009	$AICc_0 = -52.0$, $AICc_g = -83.7$
Proportion of damaged trees	Count: (Intercept)	−2.8985	0.2811	−10.310	<0.0001	Random effect (count) SD = 0.2502
Alternative model	Count: Defoliation	0.0286	0.0072	4.000	0.0001	Random effect (zero-inflation) SD = 7.063
	Zero-inflation: (Intercept)	−8.431	4.503	−1.872	0.0612	AICc = −96.9, $AICc_0 = -52.0$,
Density	(Intercept)	4.1021	0.5673	7.231	<0.0001	Random effect SD = 0.4503
	Proportion of damaged trees	−4.4382	1.5805	−2.808	0.0050	AICc = 237.8
	Canopy cover	−1.8231	0.2834	−6.432	<0.0001	$AICc_0 = 352.9$, $AICc_g = 250.4$
	Summer groundwater table level	0.5399	0.2058	2.623	0.0087	

SE – standard error, z – test statistic, Pr(>|z|) – *p*-value, AICc – Akaike's Information Criterion, with correction for small sample size, $AICc_0$ – AICc of null model (intercept-only), $AICc_g$ – AICc of global model (covering all hypothesized variables after excluding intercorrelated variables), SD – standard deviation.

Efekty brzegowe plot(ggpredict(mod.u))

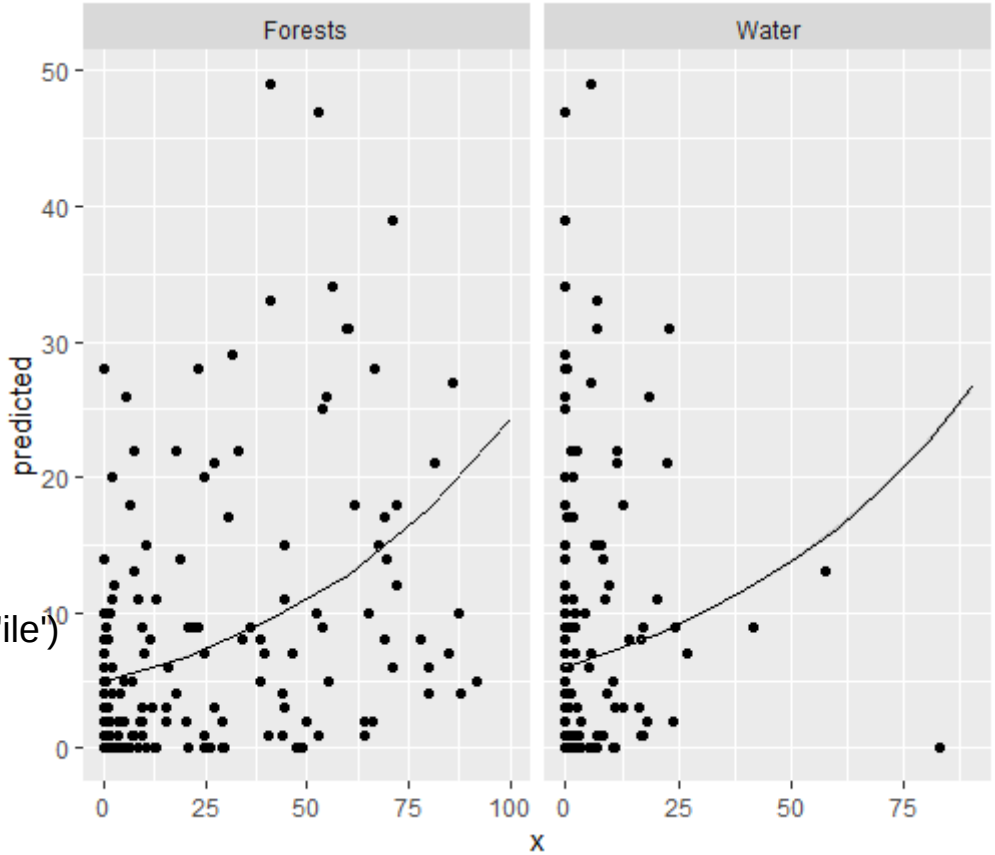




Tylko część zliczeniowa (count), bez zero-inflated!

Wykres z punktami

```
ggp1<-as.data.frame(ggpredict(mod.u, 'Forests'))
ggp1$group<-'Forests'
ggp2<-as.data.frame(ggpredict(mod.u, 'Water'))
ggp2$group<-'Water'
ggp<-bind_rows(list(ggp1, ggp2))

gg.points<-reshape2::melt(urban[,c(6,14,16)], id.vars='ile')
colnames(gg.points)<-c('predicted','group','x')

ggplot(ggp, aes(x=x, y=predicted))
+geom_point(data=gg.points)
+geom_ribbon(aes(ymin=predicted-std.error
,ymax=predicted+std.error ),fill='gray80’)
+geom_line()+facet_wrap(~group, scales='free_x')
```



Dane zliczeniowe – kolejność wykonywania działań

- Start: wiemy że mamy dane zliczeniowe
- Sprawdzamy model globalny pod kątem VIF
- Po ewentualnej redukcji sprawdzamy zero inflation i overdispersion funkcjami *DHARMa::testDisepersion()* i *DHARMa::testZeroInflation()*
- selekcja zmiennych: *step()* lub *dredge()* – działają z *glmmTMB()*
- summary, wykresy, decyzja

Proporcje – rozkład Beta

- Do danych dotyczących udziału lub procentów
- 0<zmienna zależna<1, choć możliwy rozkład zero inflated beta
- przykłady:
    - udział gatunków wskaźnikowych starych lasów
    - pokrycie koron drzew
    - udział zgryzionych pędów
    - proporcja piskląt które przeżyły pierwszy rok

Co zrobić jak są wartości 1? Jak dużo – iść w kierunku binomial (stracić informację), jak niedużo – zmienić 1 na 0.99 – model wtedy to przepuści

Przykład – procent zgryzanych drzewek

br to liczba zgryzanych drzewek/liczba wszystkich drzewek

```
summary(prop$br)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.2935  0.3649  0.3454  0.4223  0.5556 
```

Model 1

- mod.br<-glmmTMB(br~pH+CN+b+c, data = prop, family=beta_family())
- Error in eval(family$initialize) : y values must be 0 < y < 1

- Zero-inflation – trzeba dać ziformula
- mod.br<-glmmTMB(br~pH+CN+b+c, ziformula = ~b+c, data = prop, family=beta_family() , na.action = na.fail) #ostatnie dla dredge

- MuMIn::dredge(mod.br)

```
> MuMIn::dredge(mod.br)
Fixed terms are "cond((Int))", "zi((Int))" and "disp((Int))"
Global model call: glmmTMB(formula = br ~ pH + CN + b + c, data = prop, family = beta_family(),
    ziformula = ~b + c, na.action = na.fail, dispformula = ~1)
---
Model selection table
   cnd((Int)) zi((Int)) dsp((Int))    cnd(b)  cnd(c)    cnd(CN)   cnd(pH)   zi(b)   zi(c) df logLik  AICc
35    -1.5010    -8.404          +           0.01593                                0.08317  5 31.119 -50.5
51    -1.5010   -13.330          +           0.01593                      0.09288 0.11550  6 32.453 -50.4
3     -1.5010    -2.944          +           0.01593                                       4 29.744 -50.3
4     -1.2950    -2.944          + -0.005876 0.01459                                       5 30.665 -49.6
```

cnd – conditional (liczebność), zi – zero-inflated
Najlepszy model: c jako cnd i zi

mod.br<-glmmTMB(br~c, ziformula = ~c, data = prop, family=beta_family(), na.action = na.fail)
> summary(mod.br)
 Family: beta (logit)
Formula: br ~ c
Zero inflation: ~c
Data: prop

AIC	BIC	logLik	deviance	df.resid
-52.2	-43.8	31.1	-62.2	35

Dispersion parameter for beta family (): 27.2

Conditional model:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-1.50121	0.37774	-3.974	7.06e-05	***
c	0.01593	0.00631	2.525	0.0116	*

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

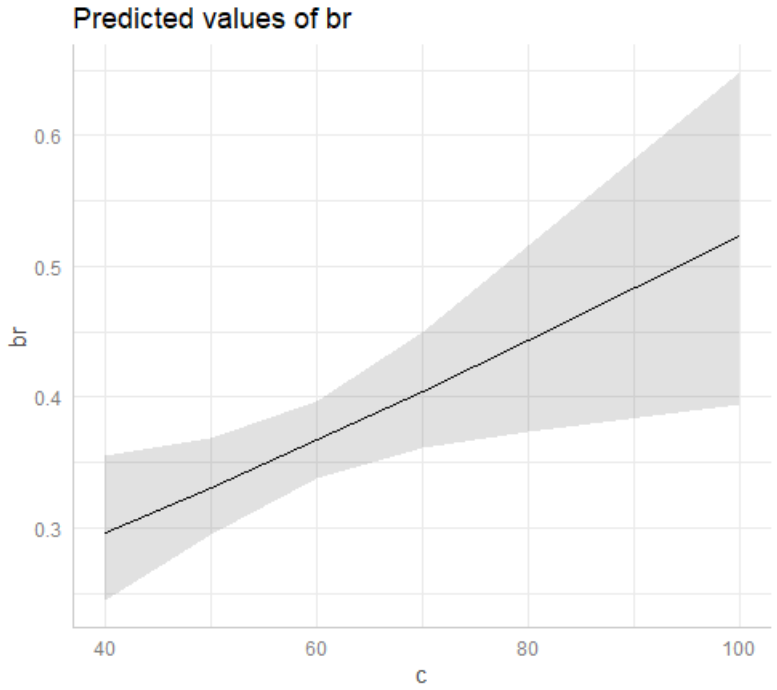
Zero-inflation model:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-8.40446	3.71108	-2.265	0.0235	*
c	0.08317	0.05067	1.642	0.1007	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Wielkość efektu?

```
ggpredict(mod.br)
$c
# Predicted values of br

  c | Predicted |       95% CI
------------------------------
 40 |      0.30 | [0.24, 0.35]
 50 |      0.33 | [0.29, 0.37]
 60 |      0.37 | [0.34, 0.40]
 70 |      0.40 | [0.36, 0.45]
 80 |      0.44 | [0.37, 0.52]
100 |      0.52 | [0.39, 0.65]
```



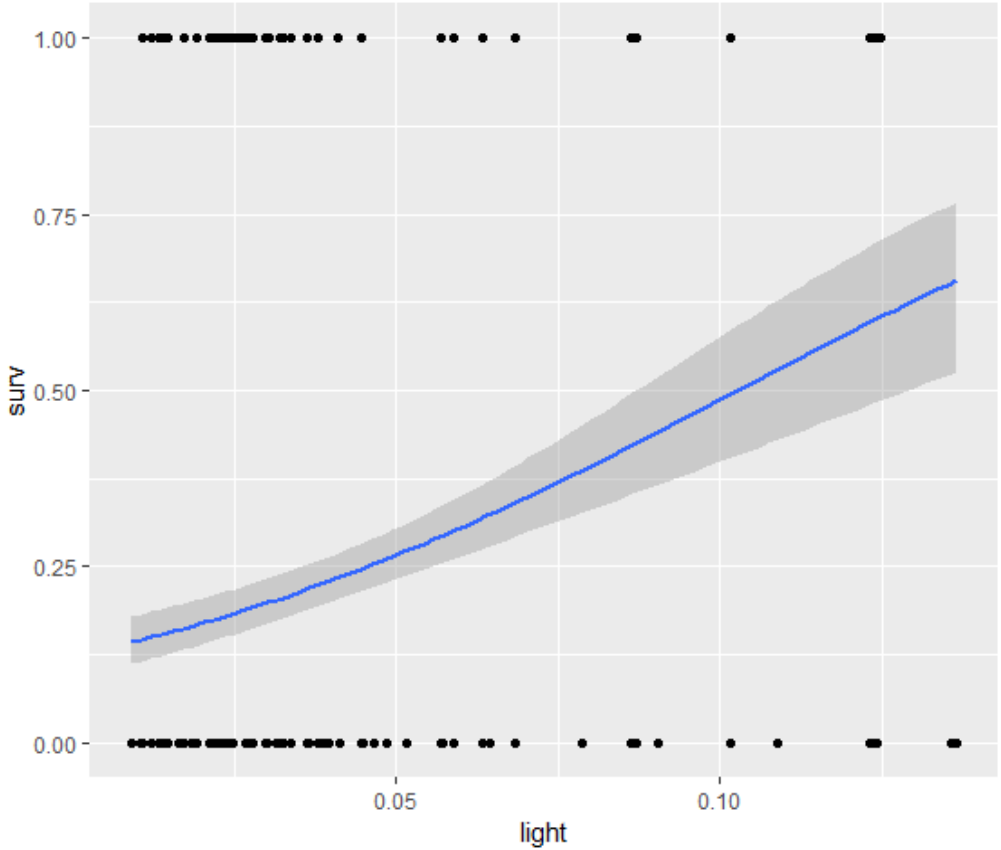
Zwiększenie pokrycia c z 40 do 50% zwiększa proporcję zgryzionych drzewek o 3%, zwiększenie z 80 na 100% o 8%

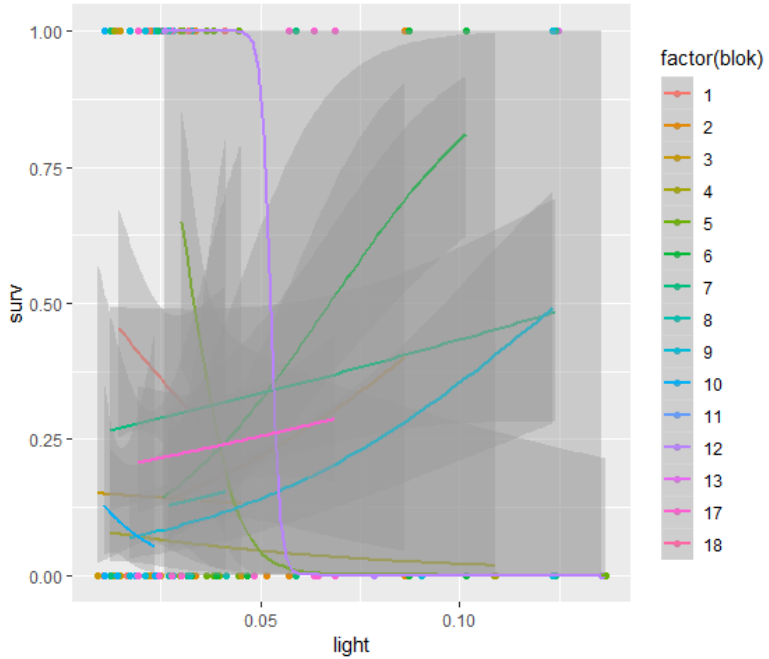
Uogólnione LMM - Generalized Linear Mixed-effects Models (GLMM)

Ta sama zasada co w przypadku GLM - możemy uogólnić LMM na inne rozkłady

Przykład - przeżywalność siewek dębu czerwonego

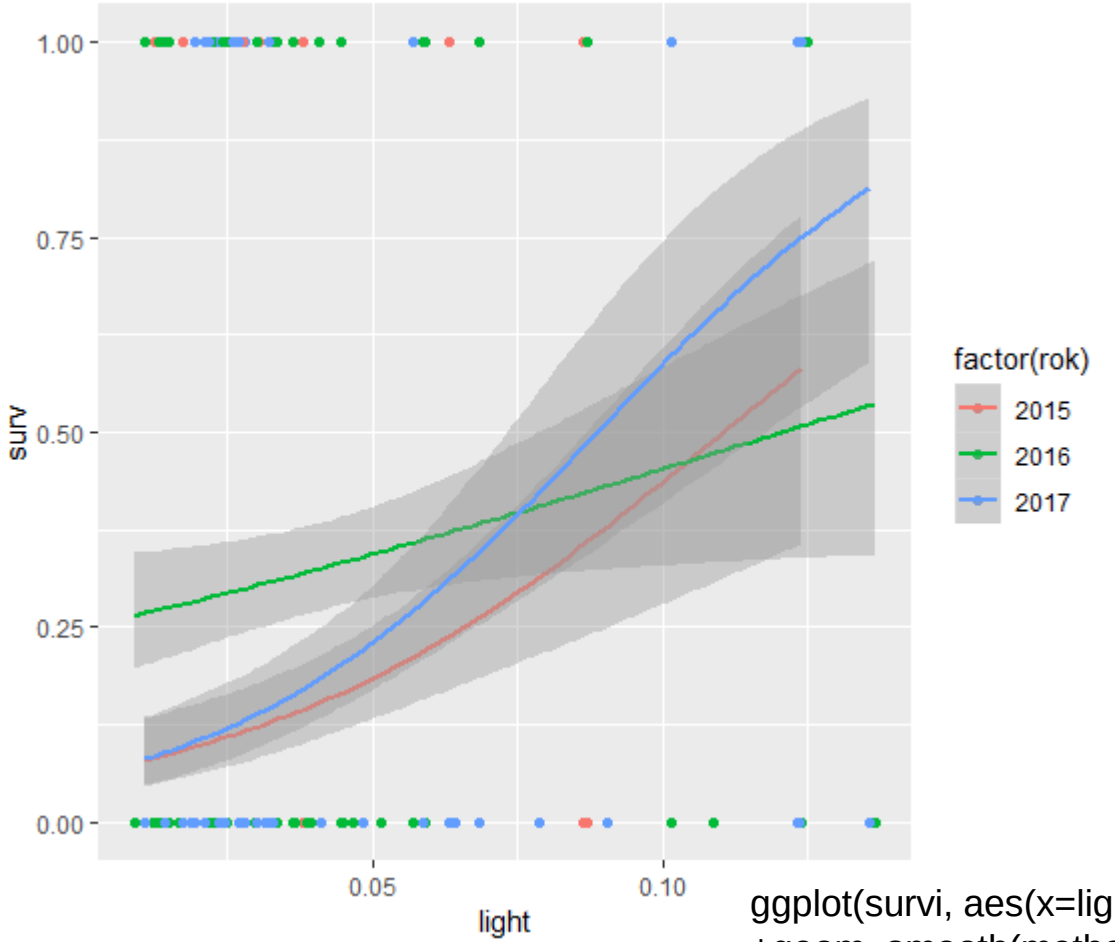
Trzy lata, poletka w ramach bloków - efekty losowe





ggplot(survi, aes(x=light, y=surv, col=factor(blok)))+geom_point()
+geom_smooth(method='glm',method.args=list(family='binomial'))

ggplot(survi, aes(x=light, y=surv))+geom_point()
+geom_smooth(method='glm',method.args=list(family='binomial'))



Co się działo w tych latach?
rok oznaczenia tasiemką,
przeżywalność oceniona po roku

2015 - pierwsze tasiemkowanie
2016 - dobra pomoc w terenie
2017 - susza w 2018

```
ggplot(survi, aes(x=light, y=surv, col=factor(rok)))+geom_point()
+geom_smooth(method='glm',method.args=list(family='binomial'))
```

Call:
glm(formula = surv ~ light, family = binomial(link = "logit"),
    data = survi)

Deviance Residuals:
    Min   1Q   Median   3Q   Max
-1.4627 -0.6787 -0.6265 -0.5605   1.9640

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.978   0.159 -12.444 < 2e-16 ***
light   19.233   2.859   6.727 1.73e-11 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 750.10 on 681 degrees of freedom
Residual deviance: 704.07 on 680 degrees of freedom
AIC: **708.07**

Number of Fisher Scoring iterations: 4

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial (logit)
Formula: surv ~ light + (1 | plot:blok) + (1 | rok)
   Data: survi
     AIC   BIC   logLik deviance df.resid
   678.8   696.9   -335.4   670.8   678

Scaled residuals:
    Min   1Q Median   3Q   Max
-2.0560 -0.4657 -0.3609 -0.2568 2.9111

Random effects:
 Groups   Name   Variance Std.Dev.
 plot:blok (Intercept) 0.89750 0.9474
 rok   (Intercept) 0.09808 0.3132
Number of obs: 682, groups: plot:blok, 127; rok, 3
Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.243788   0.001140   -1968   <2e-16 ***
light   19.003078   0.001139   16677   <2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Correlation of Fixed Effects:
      (Intr)
light 0.000
convergence code: 0

r.squaredGLMM(mod)

	R2m	R2c
theoretical	0.06727710	0.2839641
delta	0.04502403	0.1900380

```
mod2<-glmer(surv~light+(light|plot:blok)+(1|rok),family=binomial(link='logit'),survi)
mod3<-glmer(surv~light+(light|plot:blok)+(light|rok),family=binomial(link='logit'),survi)
mod4<-glmer(surv~light+(1|plot:blok)+(light|rok),family=binomial(link='logit'),survi)

AIC(mod.lm, mod,mod2,mod3,mod4)
          df      AIC
mod.lm  2 708.0652
mod     4 678.8019
mod2    6 682.7923
mod3    8 681.2299
mod4    6 677.2307
```

Wniosek - random slop zależny od bloku i plotu - nie za bardzo, intercept - tak, random slop zależny od roku - tak

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
Family: binomial (logit)
Formula: surv ~ light + (1 | plot:blok) + (light | rok)
Data: survi

AIC	BIC	logLik	deviance	df.resid
677.2	704.4	-332.6	665.2	676

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.6123	-0.4963	-0.3524	-0.2451	3.0508

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
plot:blok	(Intercept)	0.8019	0.8955	
rok	(Intercept)	0.4788	0.6919	
	light	58.6229	7.6566	-1.00

Number of obs: 682, groups: plot:blok, 127; rok, 3

Fixed effects:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.3249	0.4699	-4.948	7.51e-07	***
light	20.7508	6.2316	3.330	0.000869	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

	(Intr)
light	-0.891

convergence code: 0
boundary (singular) fit: see ?isSingular

r.squaredGLMM(mod4)

	R2m	R2c
theoretical	0.07901921	0.2946918
delta	0.06523120	0.2432712

Inny przykład – rozszerzenie modelu zgryzania

```
> mod.br<-glmmTMB(br~c+(1|set), ziformula = ~c+(1|set), data = prop,
family=beta_family(), na.action = na.fail)
> summary(mod.br)
 Family: beta  ( logit )
Formula:          br ~ c + (1 | set)
Zero inflation:      ~c + (1 | set)
Data: prop

     AIC      BIC   logLik deviance df.resid
   -49.3    -37.4     31.6    -63.3       33

Random effects:

Conditional model:
 Groups Name        Variance Std.Dev.
 set    (Intercept) 0.02431  0.1559
Number of obs: 40, groups:  set, 10

Zero-inflation model:
 Groups Name        Variance  Std.Dev.
 set    (Intercept) 1.913e-08 0.0001383
Number of obs: 40, groups:  set, 10

Dispersion parameter for beta family (): 32.2

Conditional model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.456134   0.386194  -3.770 0.000163 ***
c            0.015184   0.006423   2.364 0.018071 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Zero-inflation model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.40446    3.71108  -2.265   0.0235 *
c            0.08317    0.05067   1.642   0.1007
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

GLMM Zero-Inflated Poisson

hab – siedlisko (las/pole)
part – część ciała
osobnik – ID sarny

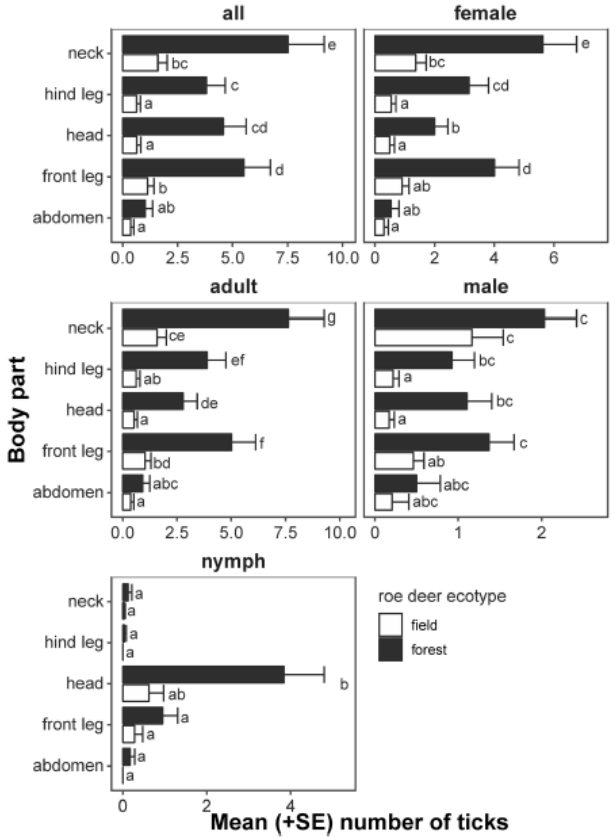
```
mod.total<-glmmTMB(all~hab*part+(1|osobnik),ziformula =
~hab*part,family = poisson,kl2)
Family: poisson  ( log )
Formula:          all ~ hab * part + (1 | osobnik)
Zero inflation:       ~hab * part
Data: kl2

     AIC      BIC   logLik deviance df.resid
  1813.4   1897.2   -885.7   1771.4      379

Random effects:

Conditional model:
 Groups  Name        Variance Std.Dev.
 osobnik (Intercept) 1.769    1.33
Number of obs: 400, groups:  osobnik, 80

Conditional model:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.0045     0.3973  -2.528  0.01147 *
habforest             1.0377     0.5141   2.018  0.04356 *
partfront leg         1.1313     0.3202   3.534  0.00041 ***
parthead              0.5646     0.3344   1.689  0.09131 .
parthind leg          0.5445     0.3348   1.626  0.10389
partneck              1.4791     0.3165   4.674 2.95e-06 ***
habforest:partfront leg   0.5420     0.4102   1.321  0.18640
habforest:parthead        0.9245     0.4230   2.186  0.02885 *
habforest:parthind leg    0.7593     0.4241   1.790  0.07339 .
habforest:partneck        0.5047     0.4063   1.242  0.21412
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Zero-inflation model:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.89642    0.59288   1.512  0.13054
habforest           -0.47429    0.75932  -0.625  0.53222
partfront leg      -16.33248  549.24286  -0.030  0.97628
parthead            -4.66956    5.47047  -0.854  0.39333
parthind leg        -2.84490    1.06626  -2.668  0.00763 **
partneck           -14.16956  444.83585  -0.032  0.97459
habforest:partfront leg   -2.41148 2253.92771  -0.001  0.99915
habforest:parthead         2.61471    5.51341   0.474  0.63533
habforest:parthind leg     0.04727    1.35836   0.035  0.97224
habforest:partneck        10.40834  444.83724   0.023  0.98133
```

Jak wyciągnąć wartości? cld i emmeans

```
cld(emmeans(mod.total, ~hab*part,type='response'))
 hab    part       rate    SE  df lower.CL upper.CL .group
 field  abdomen   0.366 0.146 379    0.168     0.80  1
 field  hind leg  0.631 0.179 379    0.361     1.10  1
 field  head      0.644 0.184 379    0.368     1.13  1
 forest abdomen   1.034 0.340 379    0.542     1.97  12
 field  front leg 1.135 0.298 379    0.678     1.90   2
 field  neck      1.607 0.414 379    0.968     2.67   23
 forest hind leg  3.807 0.859 379    2.443     5.93    3
 forest head      4.582 1.029 379    2.947     7.13    34
 forest front leg 5.510 1.215 379    3.572     8.50     4
 forest neck      7.516 1.645 379    4.888    11.56      5
```

```
library(multcomp)
library(emmeans)
```

.group – test Tukeya



Problemy z glmmTMB

Są – trzeba się na to nastawić że nie zawsze model wychodzi
Najczęściej – za dużo zmiennych w modelu – brak konwergencji – trzeba zmniejszyć liczbę zmiennych i efektów losowych
Nieraz pomaga zamiana zakresu jednostek (np. 5%=0,05) lub skalowanie predyktorów
MuMIn::dredge() liczy się zwykle kilkadziesiąt razy dłużej

czasem problem z optimalizacją kodu – warto dodać w funkcji glmmTMB argument *control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS"))*

więcej
https://cran.r-project.org/web/packages/glmmTMB/vignettes/troubleshooting.html

Zagęszczenie siewek buka w PNGS – przykład

- n=32 poletka, każde 100 m^2
- zależne przestrzennie – wzdłuż trzech rzek (random intercept riv)
- zaczynamy od modelu globalnego

```
mod.fsy1.glob<-glmmTMB(`Fagus sylvatica seedlings`~ekto+DIFN+ph+PC1
+deadAB+(1|riv),data=od.df,family=poisson(),na.action=na.fail)#,
control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS")))
Warning message:
In (function (start, objective, gradient = NULL, hessian = NULL,  :
  NA/NaN function evaluation
```

warning mnie niepokoi, wrzucam control

- mod.fsy1.glob<-glmmTMB(`Fagus sylvatica seedlings`~ekto+DIFN+ph+PC1+deadAB+(1|riv),data=od.df,family=poisson(),na.action=na.fail, control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS")))
- DHARMa:

testZeroInflation(simulateResiduals(mod.fsy1.glob))

DHARMa zero-inflation test via comparison to expected zeros with

simulation under H0 = fitted model

data: simulationOutput
ratioObsSim = 11.218, p-value < 2.2e-16
alternative hypothesis: two.sided

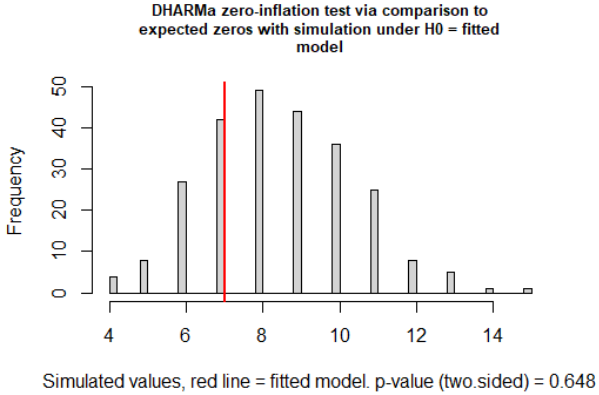
> testDispersion(simulateResiduals(mod.fsy1.glob))

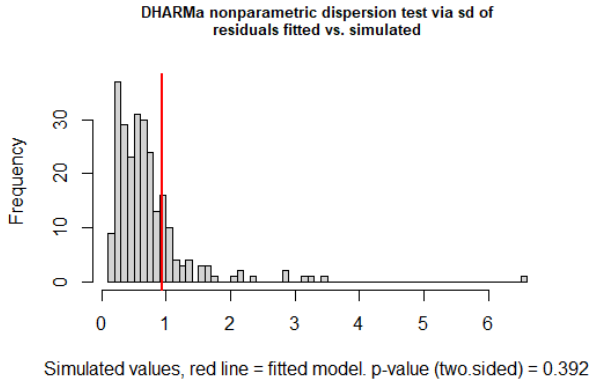
DHARMa nonparametric dispersion test via sd of residuals fitted
vs. simulated

data: simulationOutput
dispersion = 36.199, p-value < 2.2e-16
alternative hypothesis: two.sided

Problem jest raczej z rozkładem niż z zerami

```
mod.fsy1.glob<-glmmTMB(`Fagus sylvatica
seedlings`~ekto+DIFN+ph+PC1+deadAB
+(1|riv),data=od.df,family=nbinom1(),na.action=na.fail,
control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS")))
```

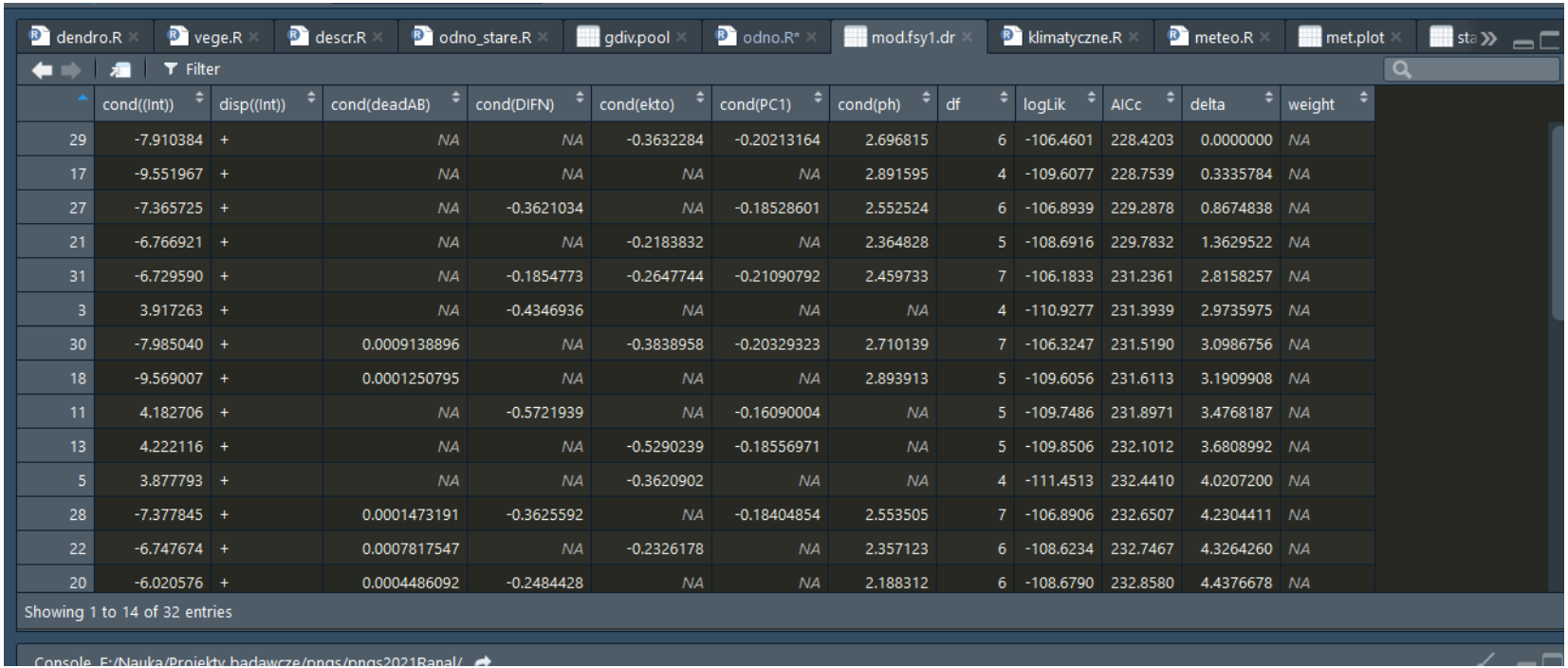




VIF

```
car::vif(mod.fsy1.glob)
Error in cov2cor(v) : 'V' is not a square numeric matrix
In addition: Warning message:
In vif.default(mod.fsy1.glob) : No intercept: vifs may not be sensible.
#Trzeba inaczej –sprawdzić vify na glmer lub glm:
car::vif(glmer(`Fagus sylvatica
seedlings`~ekto+DIFN+ph+PC1+deadAB+(1|riv),data=od.df, family='poisson'))

    ekto     DIFN       ph      PC1   deadAB
2.104512 2.995334 1.721692 1.271933 1.179923
```

#selekcja *mod.fsy1.dr<-dredge(mod.fsy1.glob)*



dendro.R | vege.R | descr.R | odno_stare.R | gdiv.pool | odno.R* | mod.fsy1.dr | klimatyczne.R | meteo.R | met.plot | sta

Filter

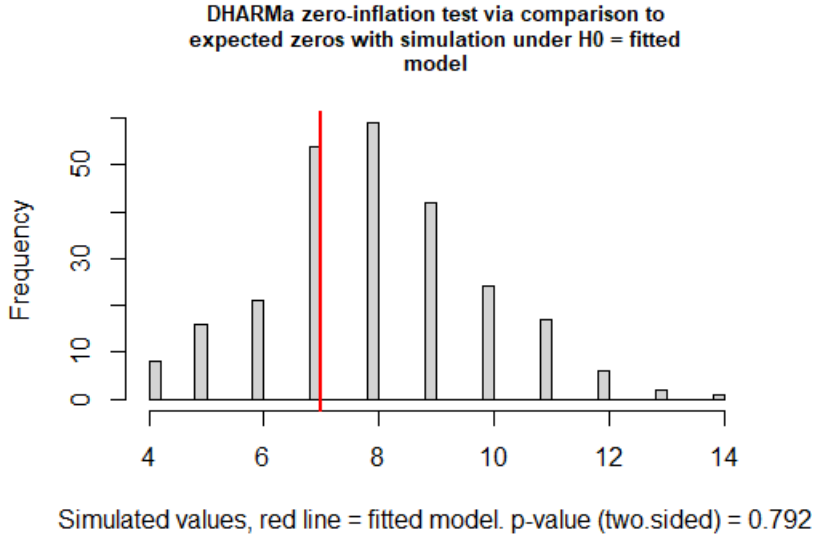
	cond((Int))	disp((Int))	cond(deadAB)	cond(DIFN)	cond(ekto)	cond(PC1)	cond(ph)	df	logLik	AICc	delta	weight
29	-7.910384	+	NA	NA	-0.3632284	-0.20213164	2.696815	6	-106.4601	228.4203	0.0000000	NA
17	-9.551967	+	NA	NA	NA	NA	2.891595	4	-109.6077	228.7539	0.3335784	NA
27	-7.365725	+	NA	-0.3621034	NA	-0.18528601	2.552524	6	-106.8939	229.2878	0.8674838	NA
21	-6.766921	+	NA	NA	-0.2183832	NA	2.364828	5	-108.6916	229.7832	1.3629522	NA
31	-6.729590	+	NA	-0.1854773	-0.2647744	-0.21090792	2.459733	7	-106.1833	231.2361	2.8158257	NA
3	3.917263	+	NA	-0.4346936	NA	NA	NA	4	-110.9277	231.3939	2.9735975	NA
30	-7.985040	+	0.0009138896	NA	-0.3838958	-0.20329323	2.710139	7	-106.3247	231.5190	3.0986756	NA
18	-9.569007	+	0.0001250795	NA	NA	NA	2.893913	5	-109.6056	231.6113	3.1909908	NA
11	4.182706	+	NA	-0.5721939	NA	-0.16090004	NA	5	-109.7486	231.8971	3.4768187	NA
13	4.222116	+	NA	NA	-0.5290239	-0.18556971	NA	5	-109.8506	232.1012	3.6808992	NA
5	3.877793	+	NA	NA	-0.3620902	NA	NA	4	-111.4513	232.4410	4.0207200	NA
28	-7.377845	+	0.0001473191	-0.3625592	NA	-0.18404854	2.553505	7	-106.8906	232.6507	4.2304411	NA
22	-6.747674	+	0.0007817547	NA	-0.2326178	NA	2.357123	6	-108.6234	232.7467	4.3264260	NA
20	-6.020576	+	0.0004486092	-0.2484428	NA	NA	2.188312	6	-108.6790	232.8580	4.4376678	NA

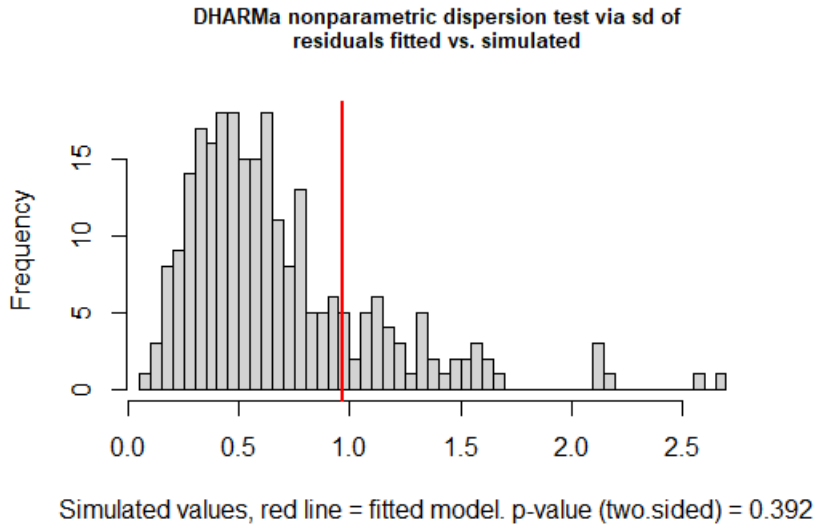
Showing 1 to 14 of 32 entries

Console F:/Nauka/Projekty badawcze/pngs/pngs2021Banal/

model finalny

mod.fsy1.fin<-glmmTMB(`Fagus sylvatica seedlings`~ekto+PC1+ph+(1|riv),data=od.df,family=nbinom1(),na.action=na.fail, control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS")))





summary(mod.fsy1.fin)

Family: nbinom1 (log)
Formula: `Fagus sylvatica seedlings` ~ ekto + PC1 + ph + (1 | riv)
Data: od.df

AIC	BIC	logLik	deviance	df.resid
224.9	233.5	-106.5	212.9	25

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.
riv	(Intercept)	3.427e-05	0.005854

Number of obs: 31, groups: riv, 3

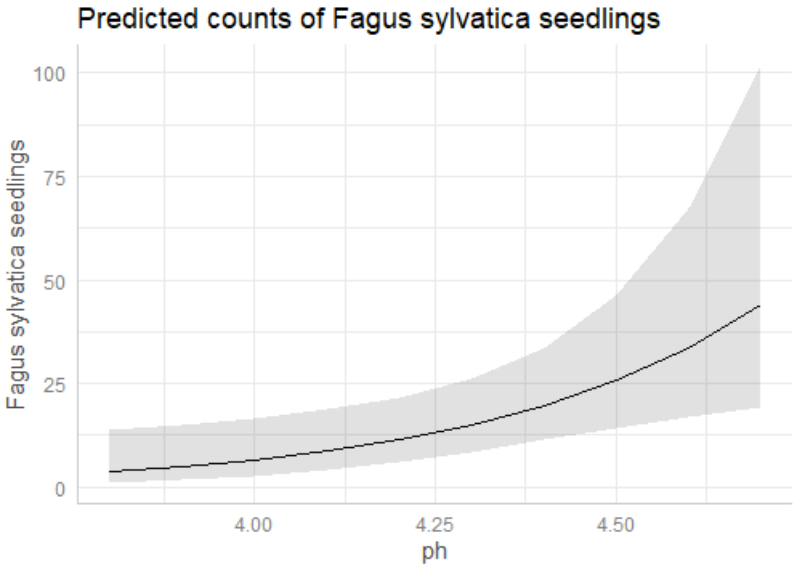
Dispersion parameter for nbinom1 family (): 35.6

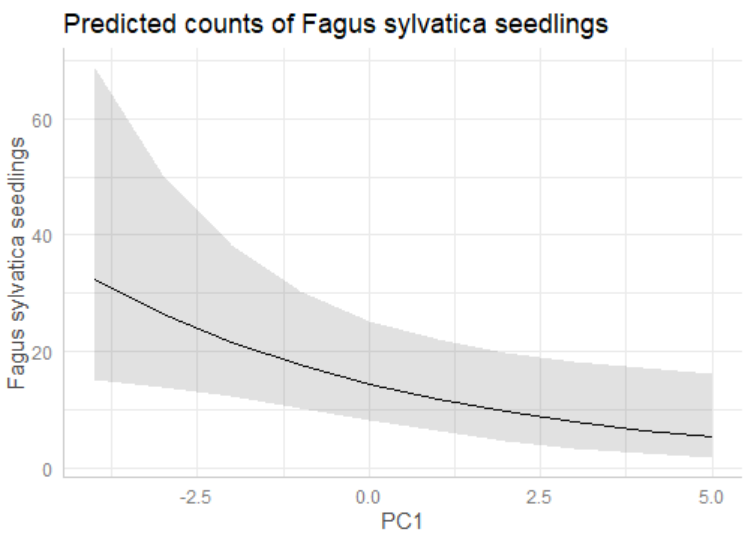
Conditional model:

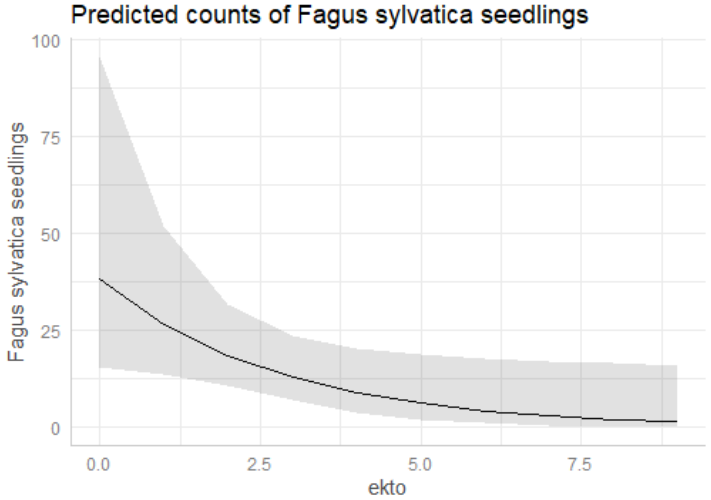
	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-7.91038	4.61592	-1.714	0.08658	.
ekto	-0.36323	0.17454	-2.081	0.03743	*
PC1	-0.20213	0.08532	-2.369	0.01783	*
ph	2.69681	1.01302	2.662	0.00776	**

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

plot(ggpredict(mod.fsy1.fin))







Uogólnione modele addytywne - GAM

Generalized Additive Models

Po co? nieparametryczne - nie zawsze rozkłady odpowiadają rzeczywistym danym i założeniom modeli

Co zamiast funkcji liniowych? Funkcje sklejane (splines)

Zamiast składników liniowych - składniki nieliniowe

Wada: słabsza możliwość wykorzystania, trudniejsze interpretacyjnie

Funkcje sklejane (splines)

dla x (-nieskończoność, -10) y=12*x+2

dla x <-10, 2) y=17^x

dla x<2,7) y=2-x^3

dla x<7,nieskończoność) y=2*x^2-3.122

mamy cztery funkcje, sklejone w jedną (wygładzoną)

Mamy trzy punkty przegięcia - w których funkcja zmienia bieg

Przykład zastosowania

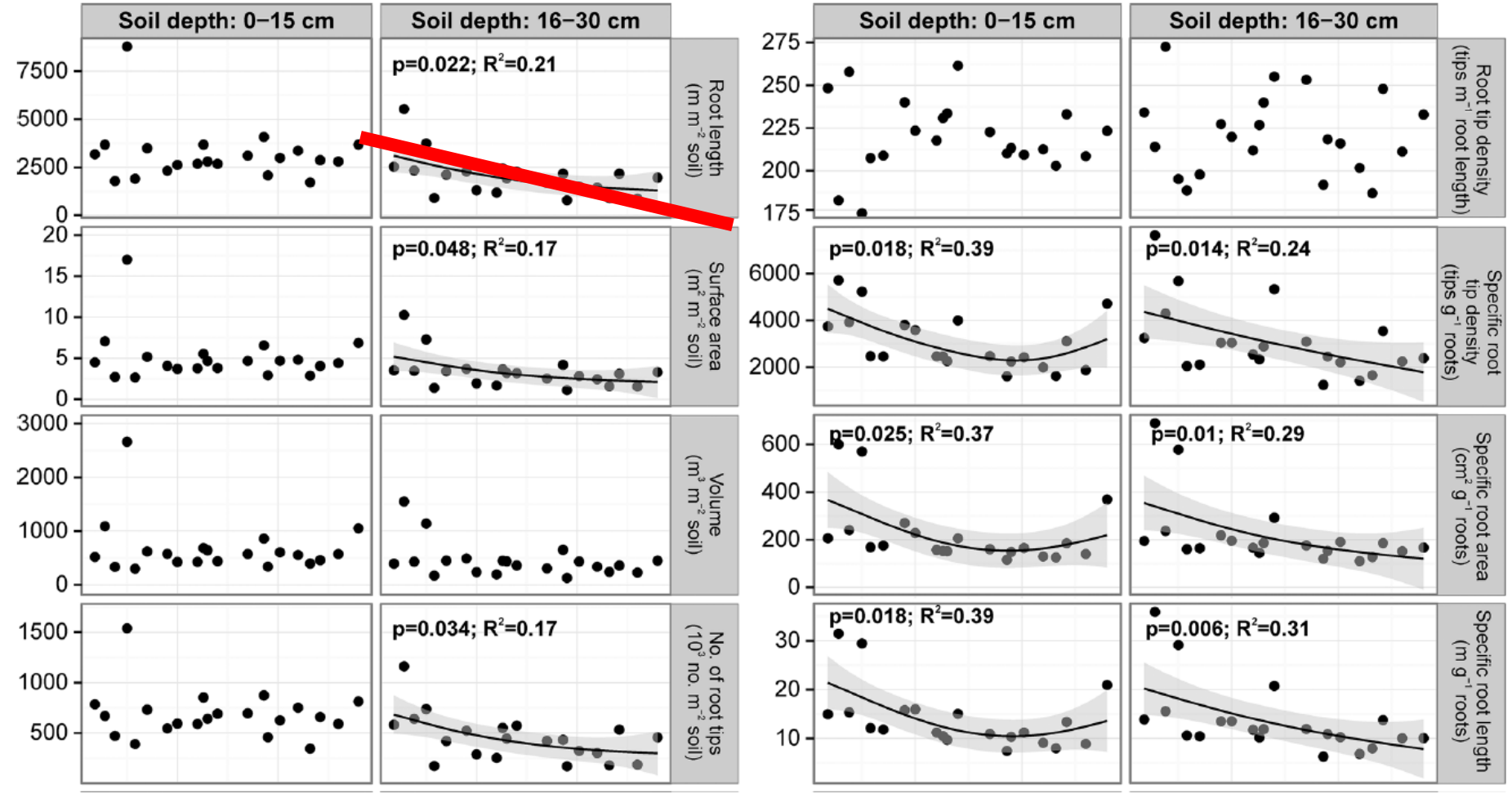
wzorce rozwoju korzeni drobnych

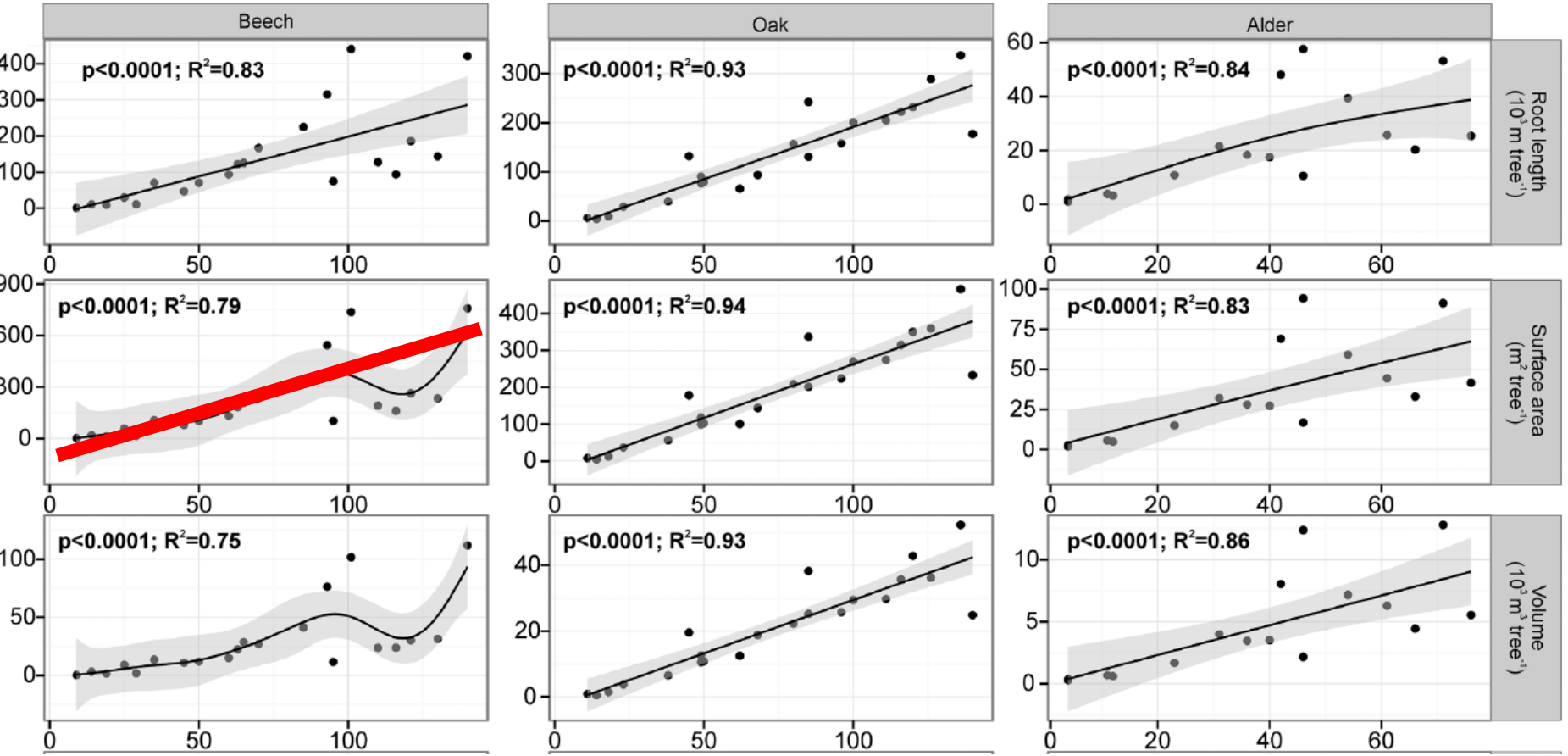
PLOS ONE

RESEARCH ARTICLE

Tree Age Effects on Fine Root Biomass and Morphology over Chronosequences of *Fagus sylvatica*, *Quercus robur* and *Alnus glutinosa* Stands

Andrzej M. Jagodzinski[1,2]*, Jędrzej Ziółkowski[2], Aleksandra Warnkowska[2], Hubert Prais[2]


Soil depth: 0–15 cm
Soil depth: 16–30 cm
Soil depth: 0–15 cm
Soil depth: 16–30 cm
Root length (m m⁻² soil)
p=0.022; R²=0.21
Root tip density (tips m⁻¹ root length)
Surface area (m² m⁻² soil)
p=0.048; R²=0.17
p=0.018; R²=0.39
p=0.014; R²=0.24
Specific root tip density (tips g⁻¹ roots)
Volume (m³ m⁻² soil)
p=0.025; R²=0.37
p=0.01; R²=0.29
Specific root area (cm² g⁻¹ roots)
No. of root tips (10³ no. m⁻² soil)
p=0.034; R²=0.17
p=0.018; R²=0.39
p=0.006; R²=0.31
Specific root length (m g⁻¹ roots)


Beech
Oak
Alder
p<0.0001; R²=0.83
p<0.0001; R²=0.93
p<0.0001; R²=0.84
p<0.0001; R²=0.79
p<0.0001; R²=0.94
p<0.0001; R²=0.83
p<0.0001; R²=0.75
p<0.0001; R²=0.93
p<0.0001; R²=0.86
Root length (10³ m tree⁻¹)
Surface area (m² tree⁻¹)
Volume (10³ m³ tree⁻¹)

```
> library(mgcv)
> model1<-gam(AB~s(V),data=sosny)
> summary(model1)

Family: gaussian
Link function: identity

Formula:
AB ~ s(V)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.5752     0.3845   56.11   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
       edf Ref.df     F p-value
s(V) 3.701  4.546 460.4  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.965   Deviance explained = 96.7%
GCV = 12.125  Scale est. = 11.385    n = 77
>
```

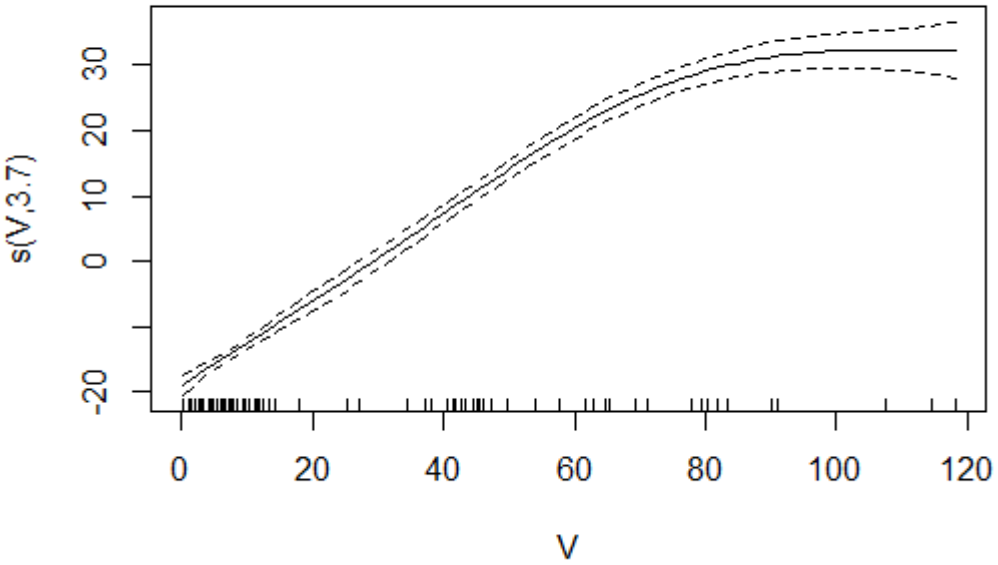
w modelu zamiast V mamy s(V), czyli funkcję sklejaną

jest to wielomian stopnia 3,7 (patrzymy na edf)

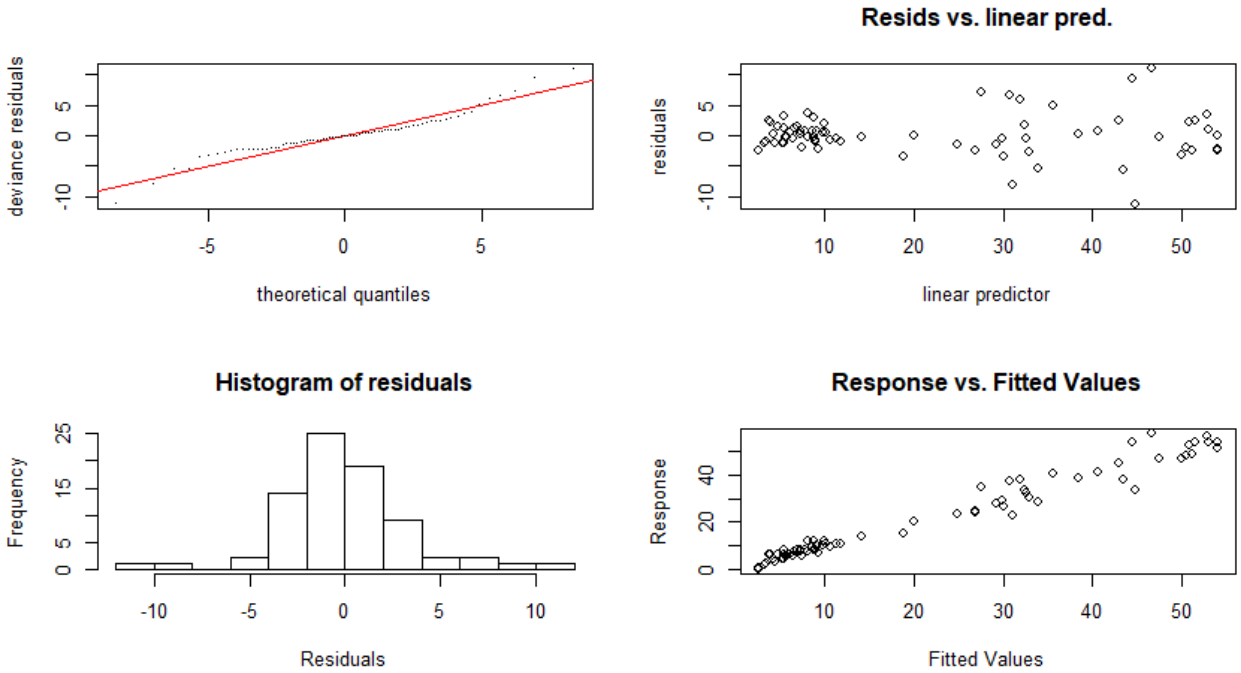
edf - estimated degrees of freedom
Ref.df - reference degrees of freedom

F - statystyka testowa

plot(model1)



```
par(mfrow=c(2,2))
gam.check(model1)
```



```
> AIC(model1)

[1] 412.3526

> AIC(model0)

[1] 666.9126
```

poprzednie modele

> AIC(lm(AB~1,data=sosny)) #model zerowy
[1] 666.9126
> AIC(lm(AB~V,data=sosny)) #model liniowy
[1] 467.6902
> AIC(lm(AB~poly(V,2),data=sosny)) #model kwadratowy
[1] 422.4431
> AIC(nls(AB~a*V^b,data=sosny,start=list(a=1,b=-2))) #model potęgowy
[1] 441.9817

>AIC(gam(AB~s(V),data=sosny)) #GAM
[1] 412.3526

```
> plot(modelbr)
> modelbr<-gam(BR~s(Age),data=sosny)
> summary(modelbr)

Family: gaussian
Link function: identity

Formula:
BR ~ s(Age)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.4899     0.3988   13.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
          edf Ref.df     F  p-value
s(Age) 7.751  8.566 9.005 3.45e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.486   Deviance explained = 53.9%
GCV = 13.814  Scale est. = 12.244    n = 77
> |
```

albo Nagini

https://harrypotter.fandom.com/pl/wiki/Nagini

```
ggplot(sosny, aes(y=BR,x=Age))+geom_point()+geom_smooth(method='gam',formula=y~s(x))
```

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

~John vonNeumann

Drawing an elephant with four complex parameters

Jürgen Mayer
Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

Khaled Khairy
European Molecular Biology Laboratory, Meyerhofstraße. 1, 69117 Heidelberg, Germany

Jonathon Howard
Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

(Received 20 August 2008; accepted 5 October 2009)

We define four complex numbers representing the parameters needed to specify an elephantine shape. The real and imaginary parts of these complex numbers are the coefficients of a Fourier coordinate expansion, a powerful tool for reducing the data required to define shapes. © *2010 American Association of Physics Teachers.*

[DOI: 10.1119/1.3254017]

A turning point in Freeman Dyson's life occurred during a meeting in the Spring of 1953 when Enrico Fermi criticized the complexity of Dyson's model by quoting Johnny von Neumann:[1] "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." Since then it has become a well-known saying among physicists, but nobody has successfully implemented it.

To parametrize an elephant, we note that its perimeter can be described as a set of points $(x(t), y(t))$, where t is a parameter that can be interpreted as the elapsed time while going along the path of the contour. If the speed is uniform, t becomes the arc length. We expand x and y separately[2] as a Fourier series

$$x(t) = \sum_{k=0}^{\infty} (A_k^x \cos(kt) + B_k^x \sin(kt)), \quad (1)$$

$$y(t) = \sum_{k=0}^{\infty} (A_k^y \cos(kt) + B_k^y \sin(kt)), \quad (2)$$

where A_k^x, B_k^x, A_k^y, and B_k^y are the expansion coefficients. The lower indices k apply to the kth term in the expansion, and the upper indices denote the x or y expansion, respectively.

Using this expansion of the x and y coordinates, we can analyze shapes by tracing the boundary and calculating the coefficients in the expansions (using standard methods from Fourier analysis). By truncating the expansion, the shape is smoothed. Truncation leads to a huge reduction in the information necessary to express a certain shape compared to a pixelated image, for example. Szekely *et al.*[3] used this approach to segment magnetic resonance imaging data. A similar approach was used to analyze the shapes of red blood cells,[4] with a spherical harmonics expansion serving as a 3D generalization of the Fourier coordinate expansion.

The coefficients represent the best fit to the given shape in the following sense. The $k=0$ component corresponds to the center of mass of the perimeter. The $k=1$ component corresponds to the best fit ellipse. The higher order components trace out elliptical corrections analogous to Ptolemy's epicycles.[5] Visualization of the corresponding ellipses can be found at Ref. 6.

We now use this tool to fit an elephant with four parameters. Wei[7] tried this task in 1975 using a least-squares Fourier sine series but required about 30 terms. By analyzing the picture in Fig. 1(a) and eliminating components with amplitudes less than 10% of the maximum amplitude, we obtained an approximate spectrum. The remaining amplitudes were

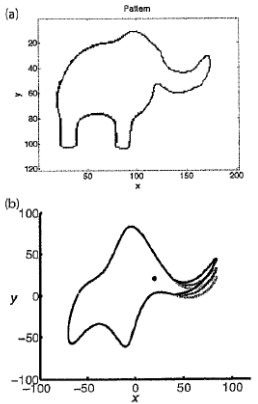


Fig. 1. (a) Outline of an elephant. (b) Three snapshots of the wiggling trunk.

Czy jest to biologicznie uzasadnione?

Co się dzieje w wieku 10 lat?

Czy coś takiego przejdzie?

- na obronie doktoratu - zależy od audytorium
- w dobrym czasopiśmie - nie bardzo

im większy stopień wielomianu/głębokość splinu tym lepsze dopasowanie

na czym nam zależy?

overfitting

Overfitting i ekstrapolacja

overfitting - model dobrze działa na zbiorze treningowym, ale źle na testowym

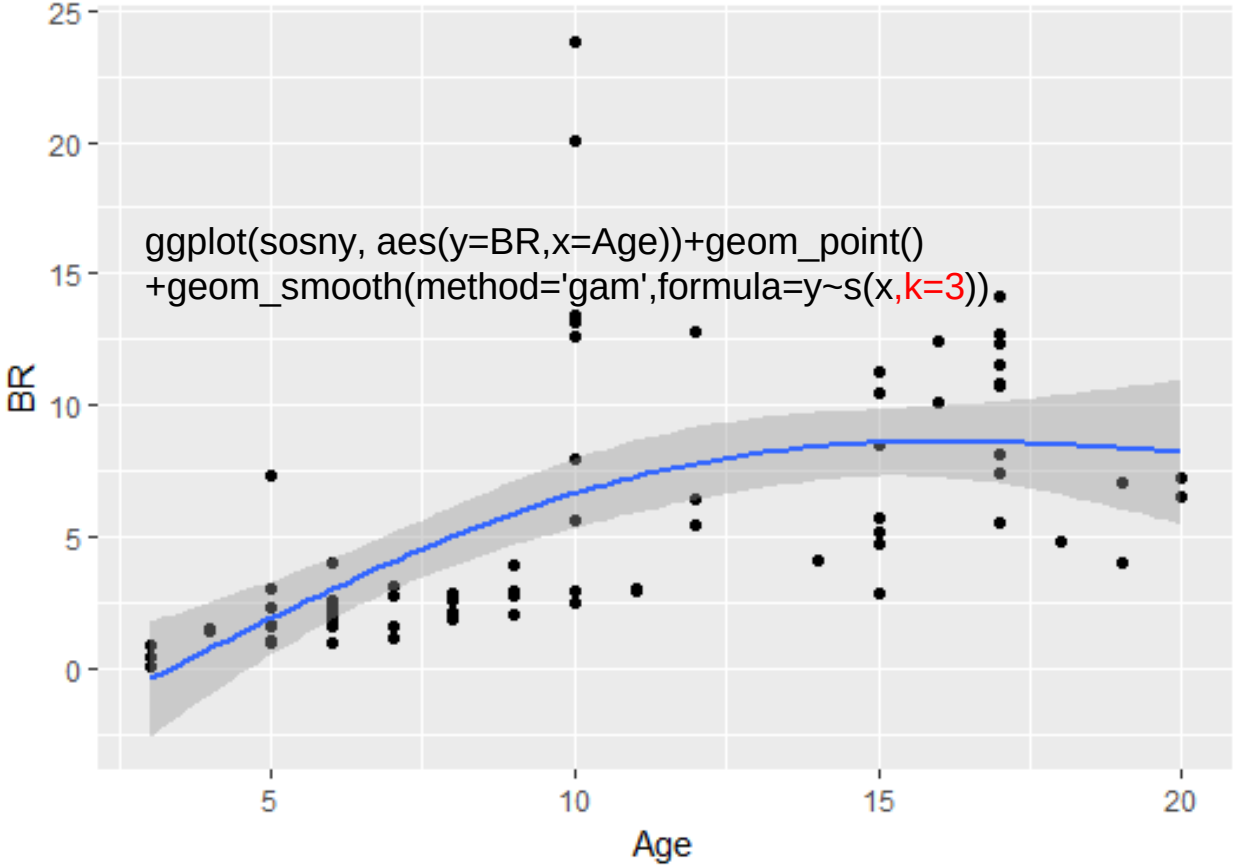
brak możliwości ekstrapolacji

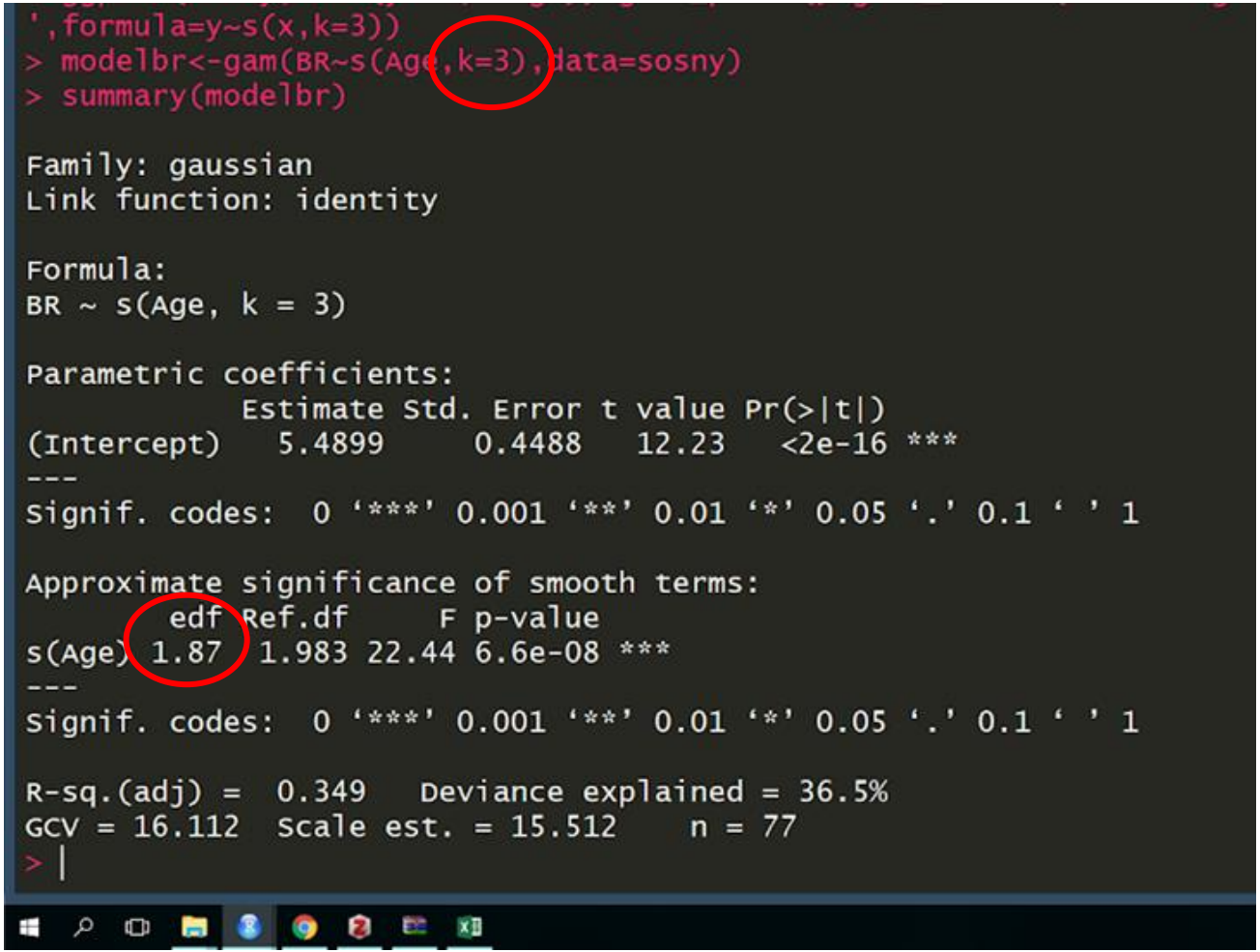
Po co robimy modele? Aby coś uogólnić, wyciągnąć trend

Oddzielić ziarno od plew - wiedzę biologiczną od elementów lokalnych i szumu

Stąd założenia o niezależności prób, wielkość próby, powtórzenia ...

rozwiązanie - ograniczyć spline




',formula=y~s(x,k=3))
> modelbr<-gam(BR~s(Age,k=3),data=sosny)
> summary(modelbr)

Family: gaussian
Link function: identity

Formula:
BR ~ s(Age, k = 3)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.4899   0.4488   12.23   <2e-16 ***

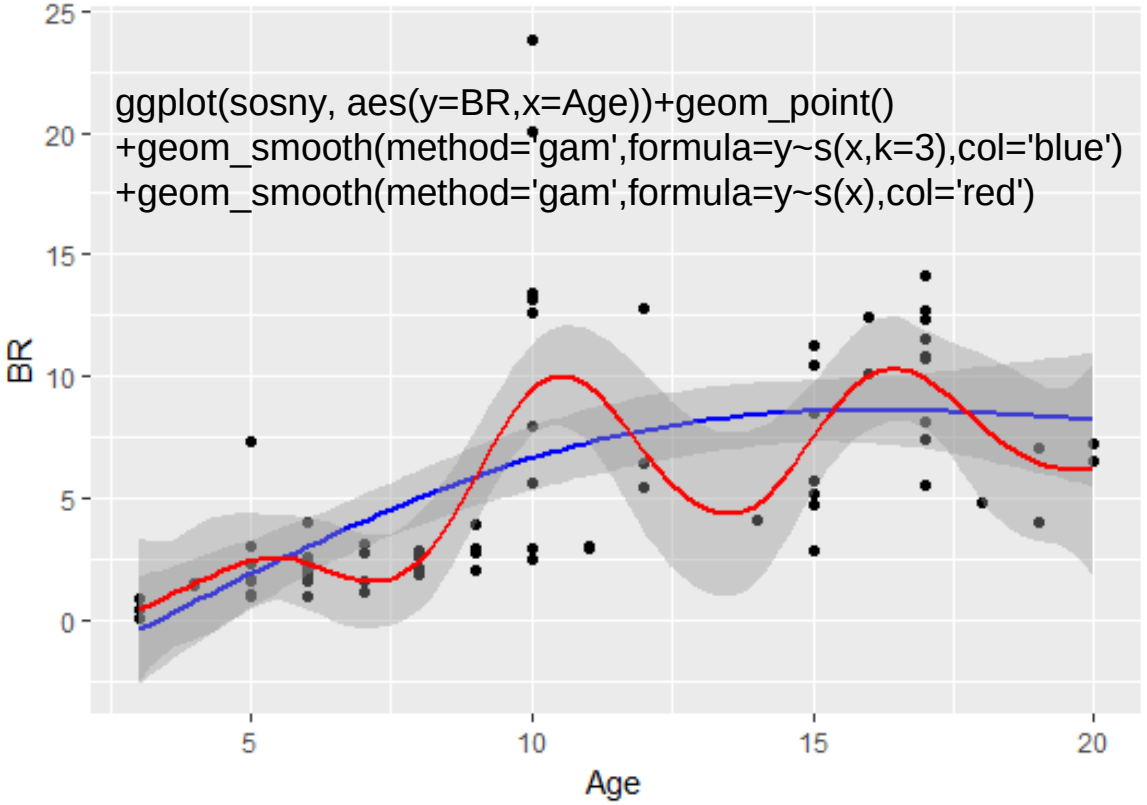
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
       edf Ref.df   F p-value
s(Age) 1.87 1.983 22.44 6.6e-08 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) = 0.349   Deviance explained = 36.5%
GCV = 16.112 Scale est. = 15.512   n = 77
>

Tracimy trochę R2 ale model jest bardziej sensowny



... all models are approximations. Essentially, **all models are wrong, but some are useful**. However, the approximate nature of the model must always be borne in mind....

~George E. P. Box


BSS
BIAŁOWIESKA SZKOŁA STATYSTYKI