



BIAŁOWIESKA SZKOŁA STATYSTYKI

Regresja – uogólnione modele liniowe (GLM i GLMM)

Krótkie przypomnienie

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

modele mogą mieć wiele czynników

modele mogą być funkcjami liniowymi innymi niż $y=ax+b$, 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='rozkład')`

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

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

`glmmTMB::glmTMB(formula, data, family=rozkład)` do GLMMów i niektórych rozkładów

`family=beta_family()` – rozkład beta (proporcja w zakresie 0-1 bez 1, np. pokrycie, udział, itp.)

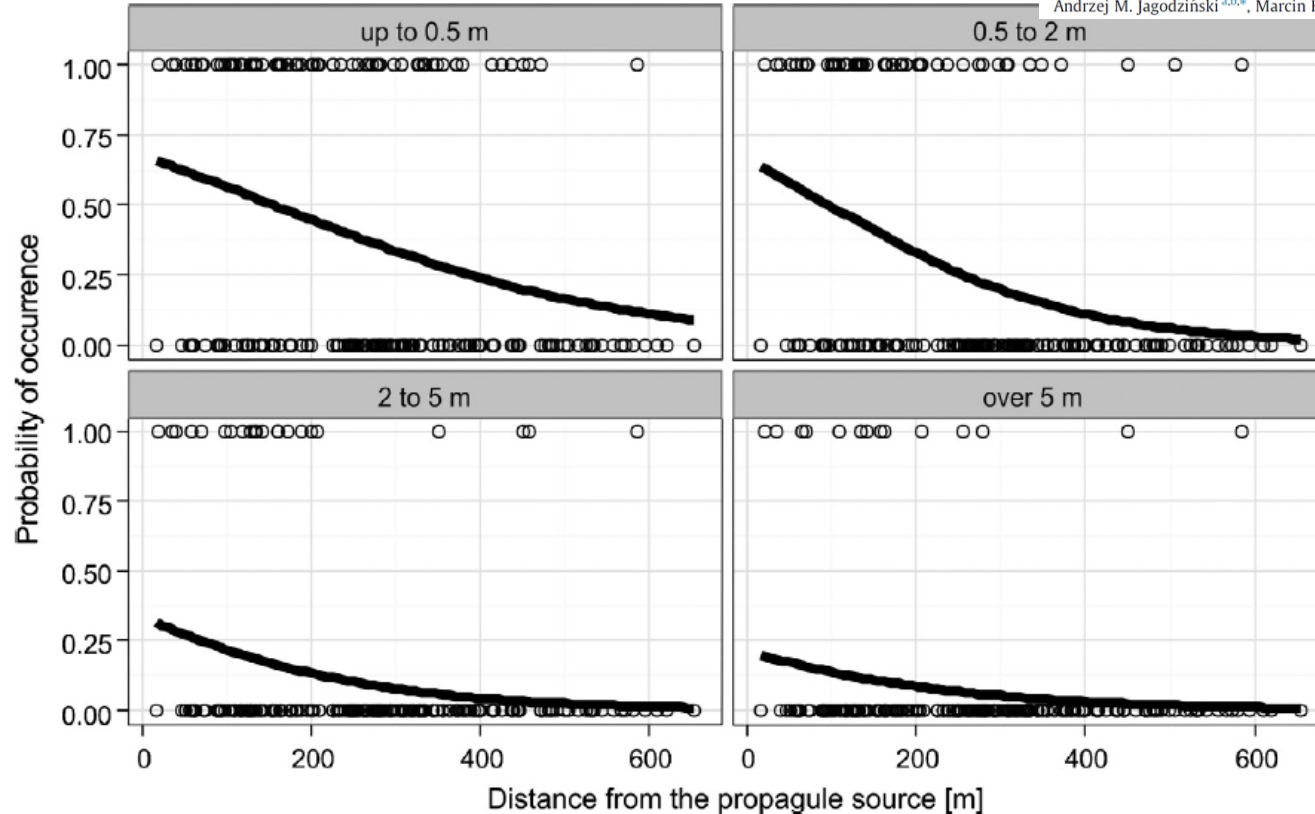
Rozkład dwumianowy (binomial)

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

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 Banaszczyk^e



GLM binomial (regresja logistyczna)

```
model<-glm(szens~odl, family=binomial, dane)
```

```
summary(model)
```

```
call:
glm(formula = szens ~ 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

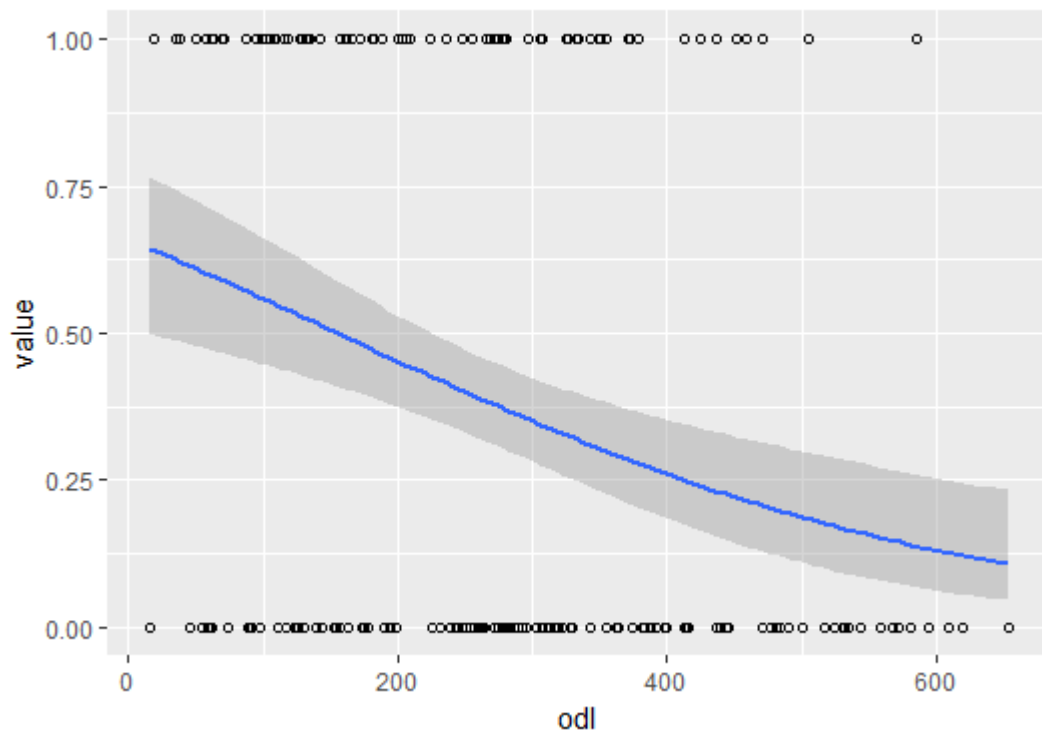
(Dispersion parameter for binomial family taken to be 1)

    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, dane)`

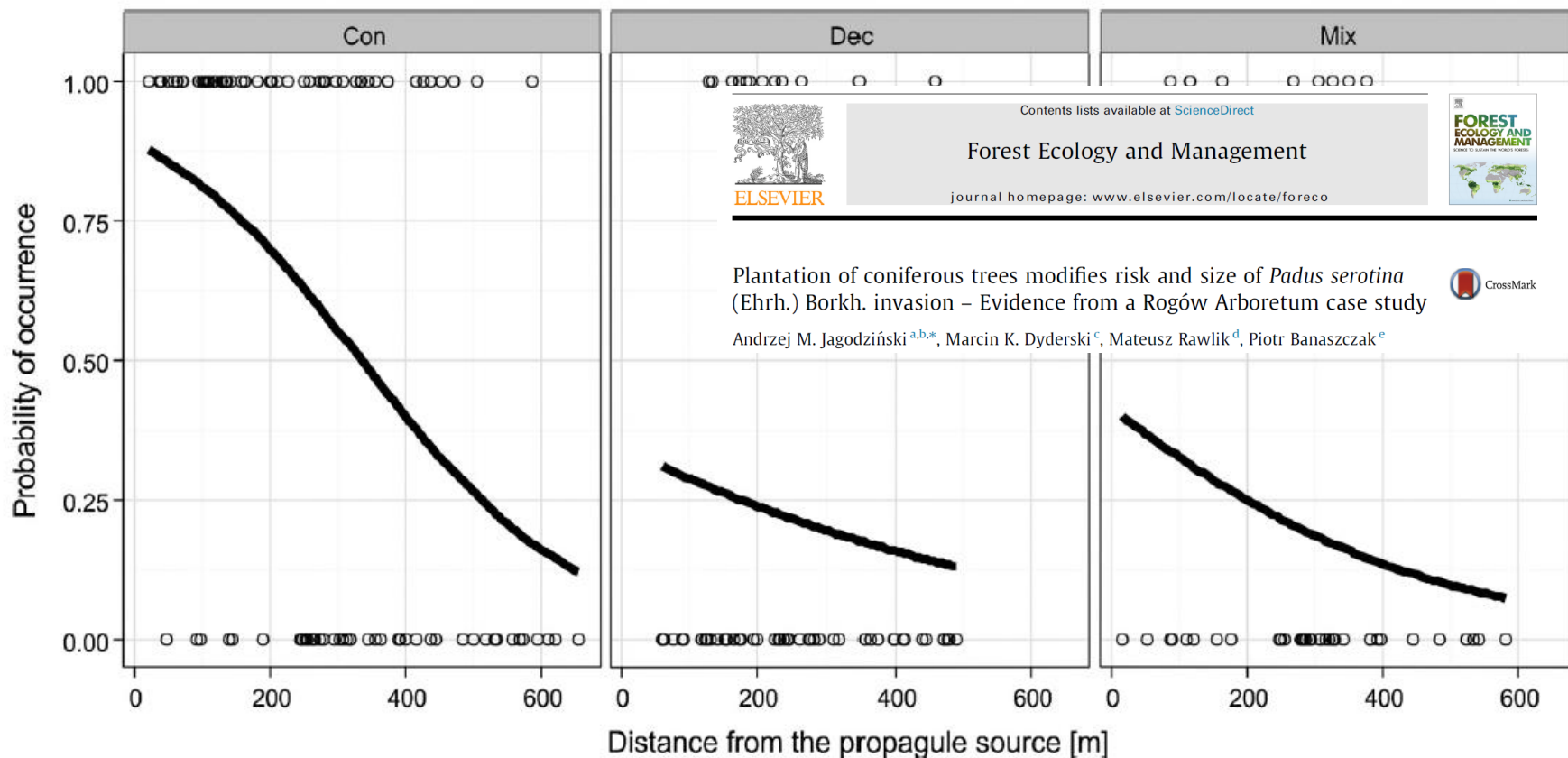


Fig. 4. Effect of coniferous (Con), deciduous (Dec) and mixed (Mix) tree stands on the probability of colonisation by black cherry ($p < 0.001$).

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,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
```

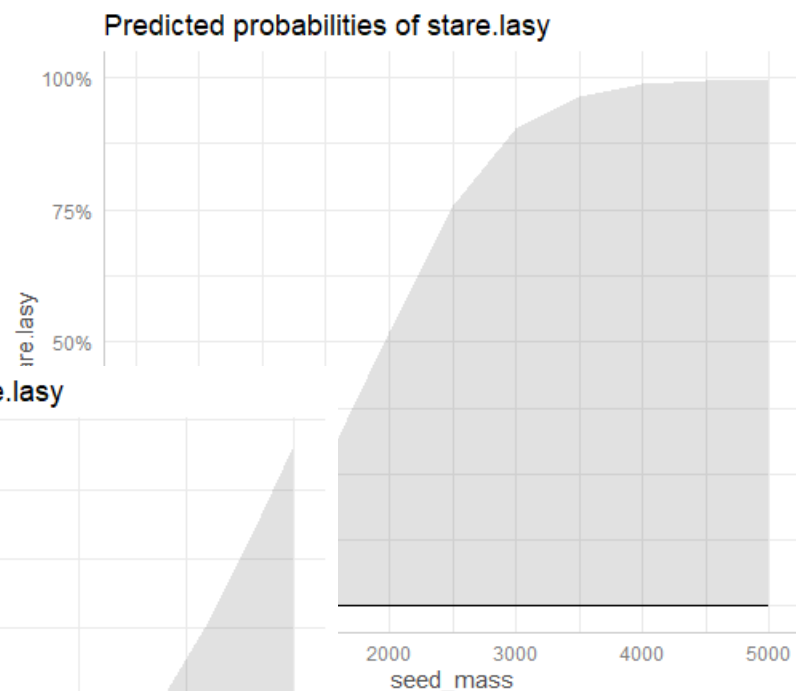
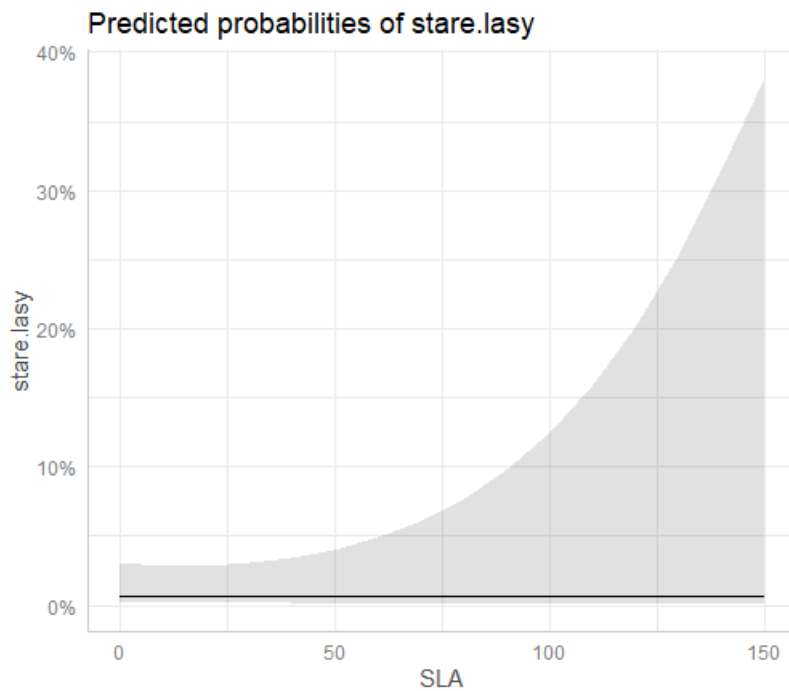
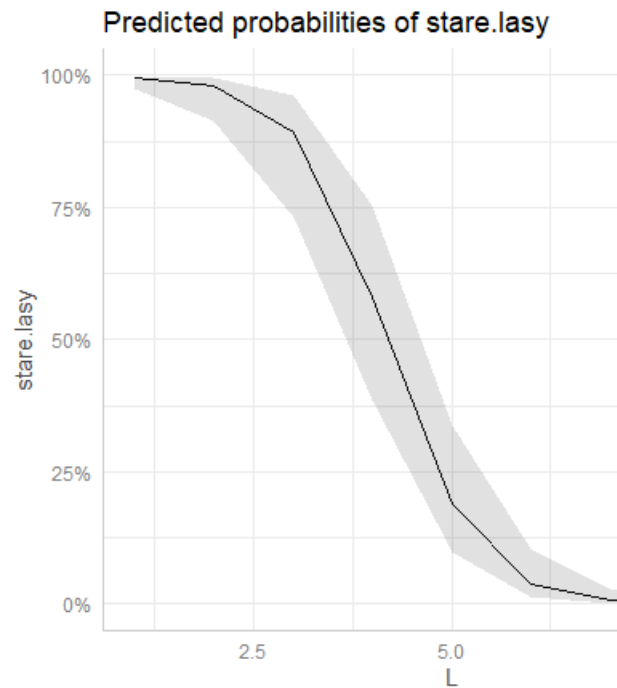
Coefficients:

```
      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
 - nie działa przy różnych jednostkach
- Bezwzględnie – trzeba przeliczyć w oparciu o funkcję łączącą
- Najłatwiej – `ggpredict()` albo `emmeans(model, formula, type='response')`
 - podaje od razu w jednostkach rzeczywistych

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 naddispersji (overdispersion) i poddispersji/niedodispersji (underdispersion)
 - Rozkład dwumianowy ujemny – przy naddispersji
 - Rozkład COM Poisson – przy poddispersji

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

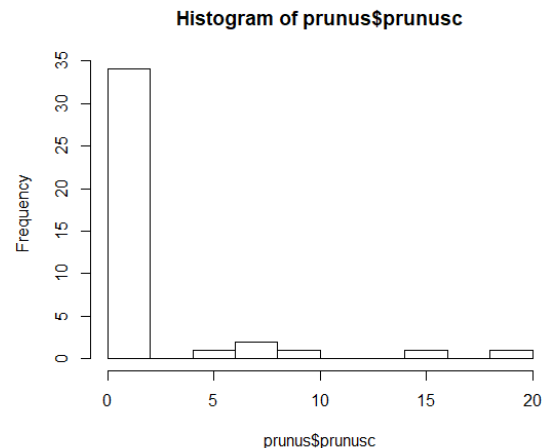
Number of Fisher Scoring iterations: 7

zwiększyło się AIC

teoretycznie jest to gorszy model, ale!

do wykonania LM jesteśmy uprawnieni mając dane o rozkładzie normalnym

logika i założenia modelu > cyferki



Interpretacja?

```
ggpredict(model)
```

```
$richness
```

```
# Predicted counts of prunusc
```

```
richness | Predicted | 95% CI
```

```
-----
```

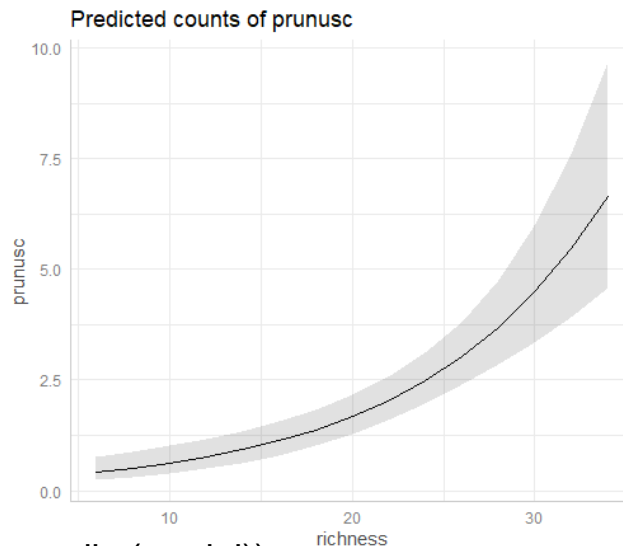
6	0.42 [0.23, 0.76]
10	0.62 [0.38, 1.01]
12	0.76 [0.49, 1.16]
16	1.12 [0.80, 1.57]
20	1.67 [1.29, 2.16]
24	2.48 [1.97, 3.11]
28	3.68 [2.85, 4.76]
34	6.66 [4.57, 9.70]

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)
```

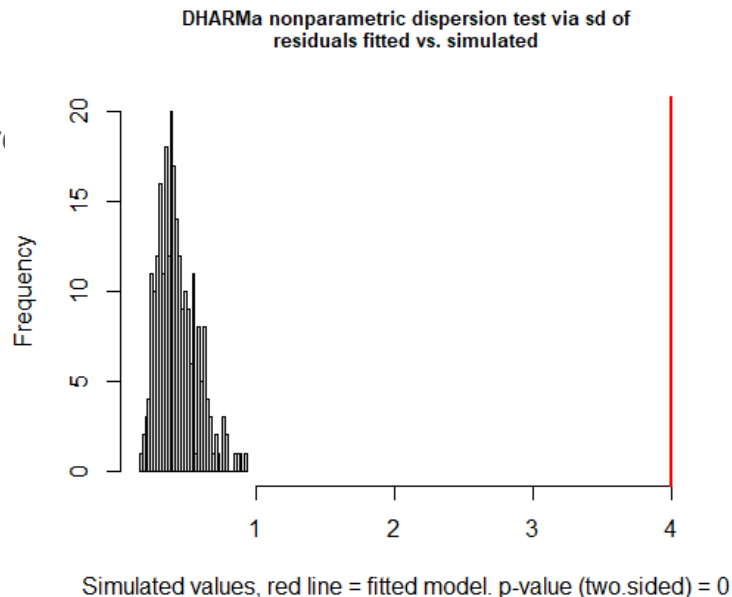
DHARMa nonparametric dispersion test via sd of residuals fitted vs. simulated

data: simulationOutput

dispersion = 9.2819, p-value < 2.2e-16

alternative hypothesis: two.sided

Mamy problem - naddyseprsja



Sprawdzamy – pakiet DHARMA

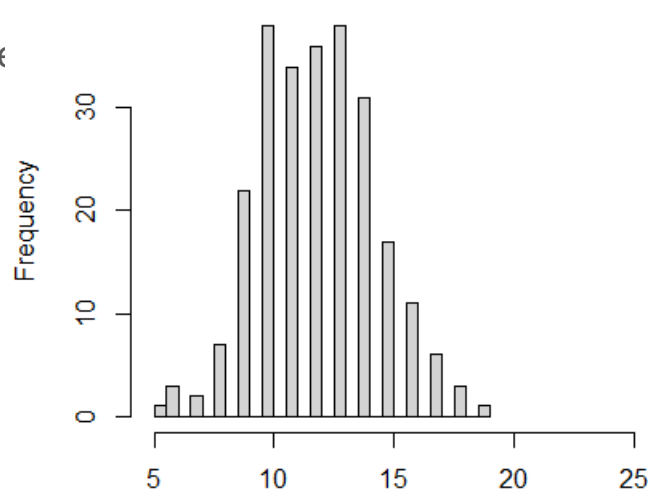
```
>library(DHARMA)
> model<-glm(prunusc~richness,data=prunus, family=poisson)
> simres<-simulateResiduals(model)
> testZeroInflation(simres)
```

DHARMA zero-inflation test via comparison to expected simulation under H_0 = fitted model

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

Mamy problem – za dużo zer!

DHARMA zero-inflation test via comparison to expected zeros with simulation under H_0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0

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)
```

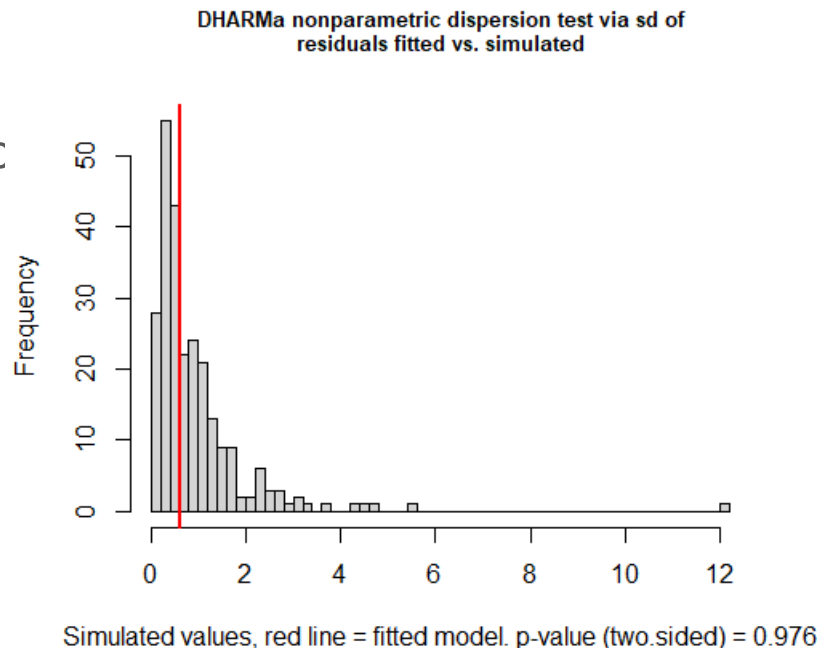
DHARMa nonparametric dispersion test via sd of residuals fitted vs. simulated

data: simulationOutput

dispersion = 0.67144, p-value = 0.976

alternative hypothesis: two.sided

Trochę poddyseprsj, ale wg testu jest dobrze, nie ma problemu

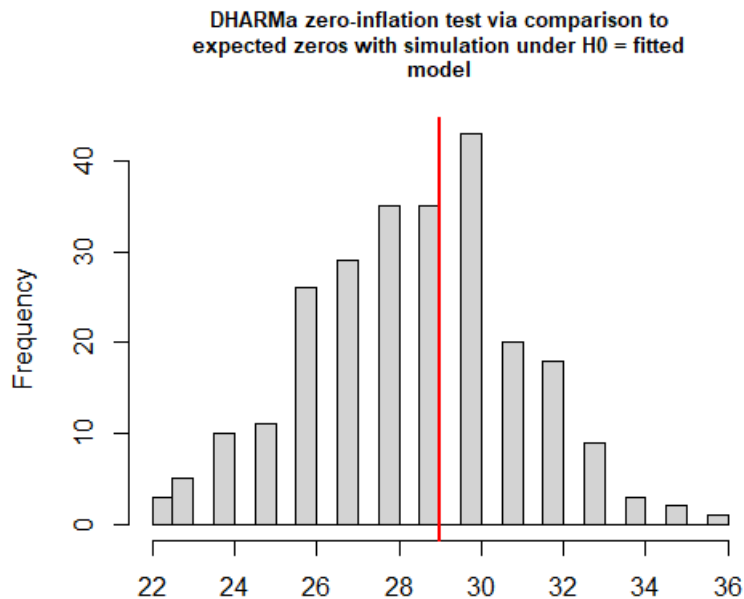


testZeroInflation(simres)

DHARMA zero-inflation test via comparison
to expected zeros with
simulation under H_0 = 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!



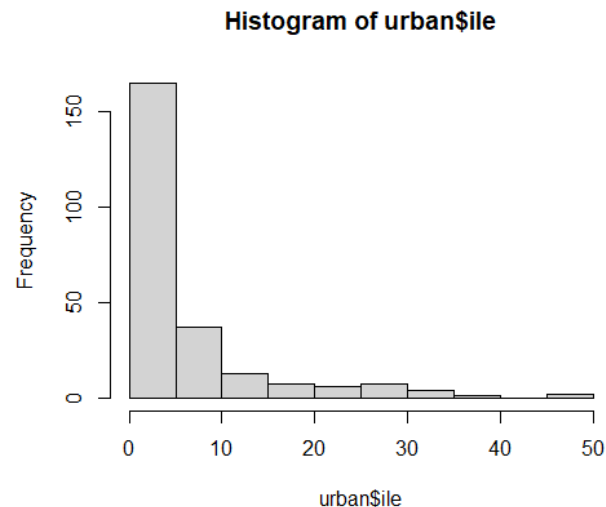
Simulated values, red line = fitted model. p-value (two.sided) = 1

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

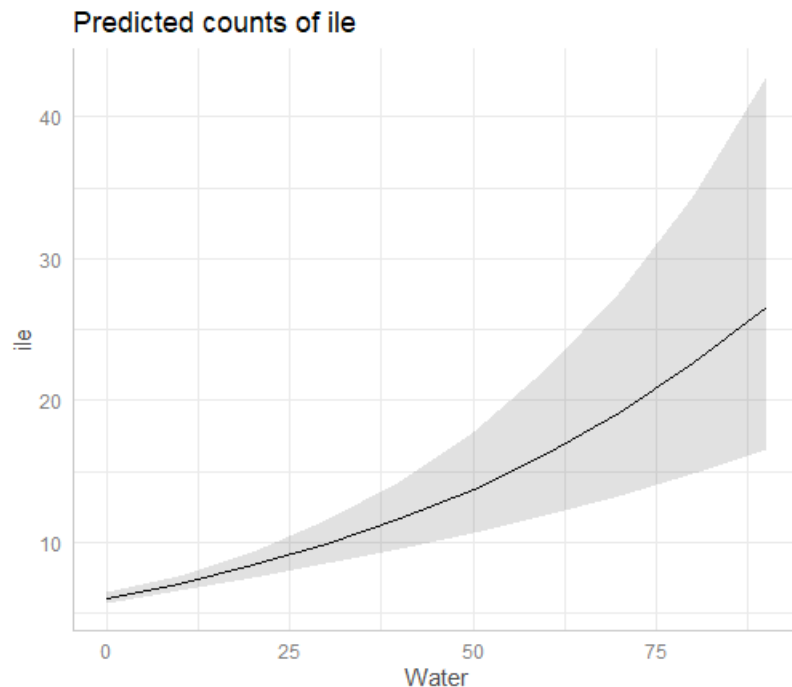
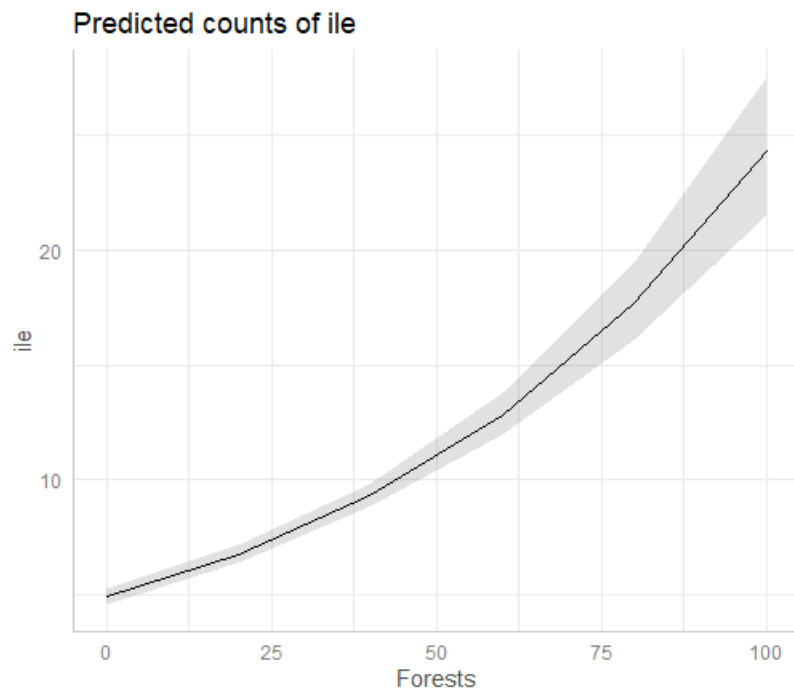
Table 2

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,
	(Intercept)	4.1021	0.5673	7.231	<0.0001	Random effect SD = 0.4503
Density	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, AICc₀ – AICc of null model (intercept-only), AICc_g – AICc of global model (covering all hypothesized variables after excluding intercorrelated variables), SD – standard deviation.

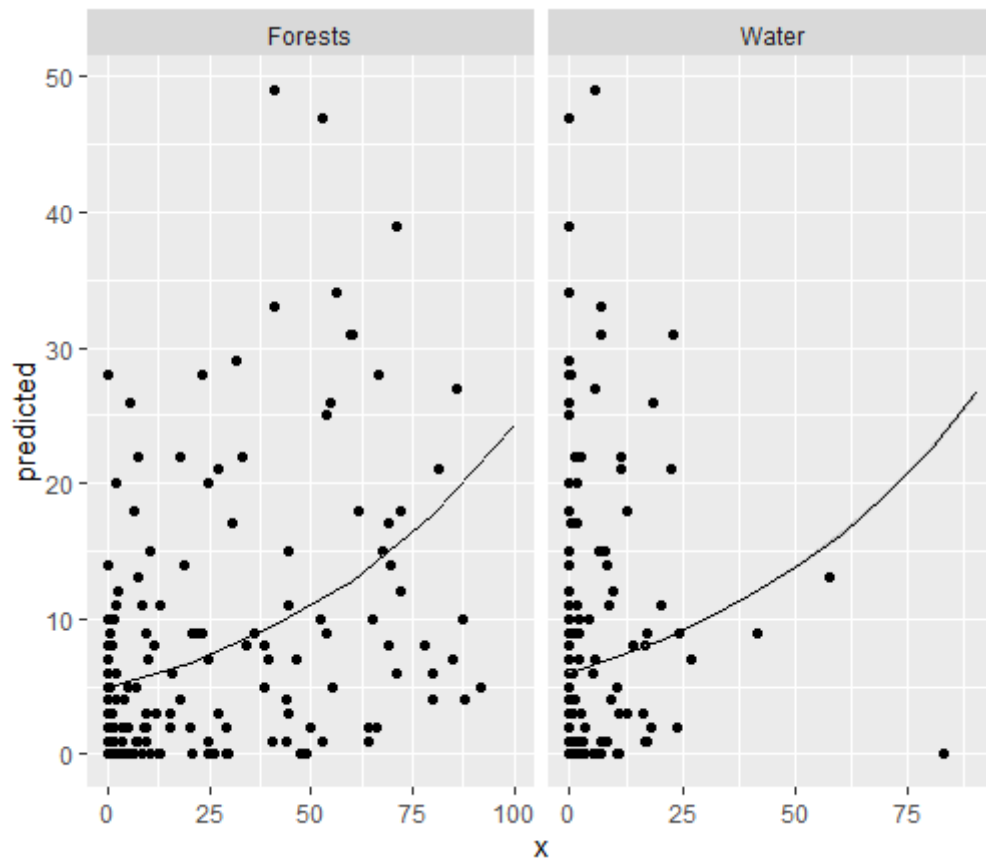
Efekty brzegowe `plot(ggpredict(mod.u))`



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

Wykres z punktami

```
plot(ggpredict(mod.u), add.data=T)
```

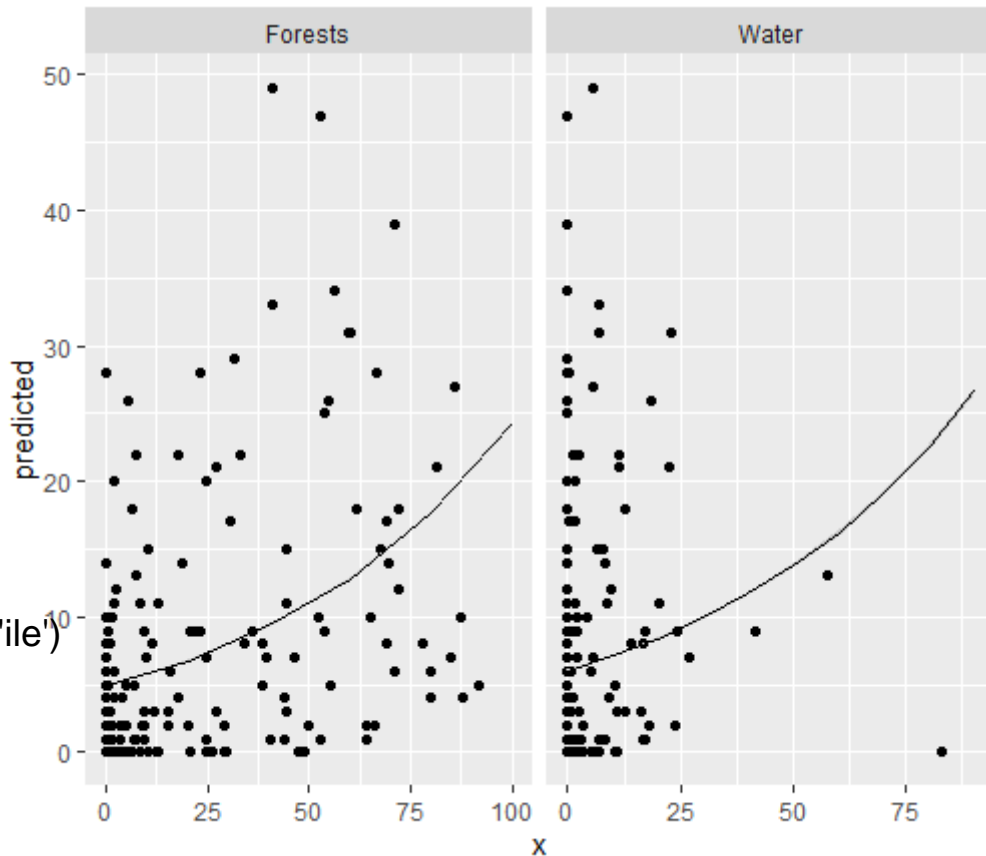


Wykres z punktami

```
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')
```

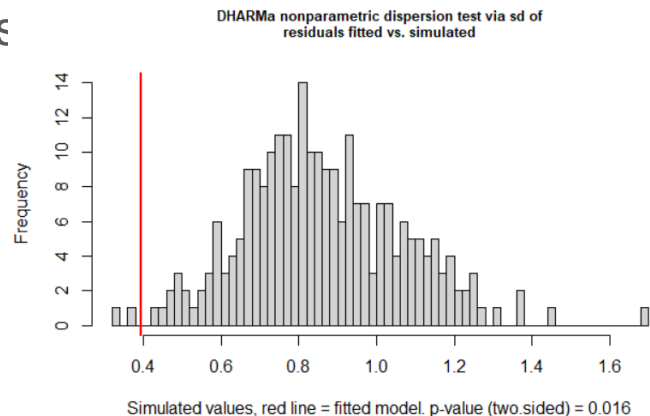


A co jak mamy niedodyspersję?

```
mod.grzyrich<-glmmTMB(grzy.rich~Sol_cover+Sol_cover*SLAsoli+  
Sol_cover*Hsoli,data=soli,na.action = na.fail, family=poisson())  
DHARMA::testDispersion(DHARMA::simulateResiduals(mod.grzyrich))
```

DHARMA nonparametric dispersion test
residuals fitted vs. simulated

data: simulationOutput
dispersion = 0.45901, p-value = 0.016
alternative hypothesis: two.sided



*tu mamy model mieszany i pakiet glmmTMB – o tym dalej będzie

Rozkład COM Poisson

- `mod.grzyrich<-glmmTMB(grzy.rich~Sol_cover+Sol_cover*SLAsoli+Sol_cover*Hsoli,data=soli,na.action = na.fail, family=compois())`

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

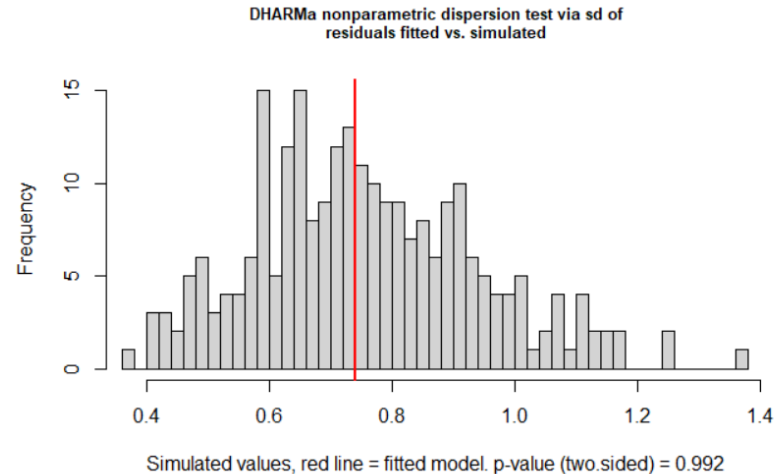
data: simulationOutput

dispersion = 0.97608, p-value = 0.992

alternative hypothesis: two.sided

O rozkładzie:

<https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/13-1912.1>



dredge(mod.grzyrich)

dla zmiennych i interakcji

model finalny:

```
mod.grzyrich<-glmmTMB(  
  grzy.rich~  
  Sol_cover+SLAsoli,  
  data=soli, na.action =  
  na.fail, family=compois())
```

```
R 4.2.2 · E:/Nauka/Projekty badawcze/Solidago/
Model selection table
```

	cnd((Int))	dsp((Int))	cnd(Hs1)	cnd(SLA)	cnd(Sol_cvr)	cnd(Hs1:Sol_cvr)
23	2.225	+		0.0012350	-0.012140	
7	1.931	+		0.0027540	0.003222	
6	3.150	+	-0.52160		0.006685	
14	2.887	+	-0.33000		0.015960	-0.005663
24	2.495	+	-0.16430	0.0008845	-0.009918	
1	2.606	+				
5	2.554	+			0.001648	
8	2.446	+	-0.29580	0.0019730	0.005627	
4	1.643	+	0.26370	0.0027960		
16	2.353	+	-0.17250	0.0015930	0.013920	-0.004937
3	2.428	+		0.0008602		
2	2.492	+	0.07849			
32	2.597	+	-0.19710	0.0006308	-0.025310	0.004178

	cnd(SLA:Sol_cvr)	df	logLik	AICc	delta	weight
23	8.357e-05	6	-93.868	202.3	0.00	0.270
7		5	-95.859	203.5	1.20	0.148
6		5	-96.246	204.3	1.97	0.101
14		6	-95.068	204.7	2.40	0.081
24	7.871e-05	7	-93.729	205.0	2.68	0.071
1		3	-99.159	205.0	2.70	0.070
5		4	-98.012	205.2	2.88	0.064
8		6	-95.434	205.4	3.13	0.056
4		5	-97.257	206.3	4.00	0.037
16		7	-94.539	206.6	4.30	0.032

```
summary(mod.grzyrich)
Family: comppois ( log )
Formula:      grzy.rich ~ Sol_cover + SLAsoli Data: soli
```

AIC	BIC	logLik	deviance	df.resid
201.7	210.2	-95.9	191.7	35

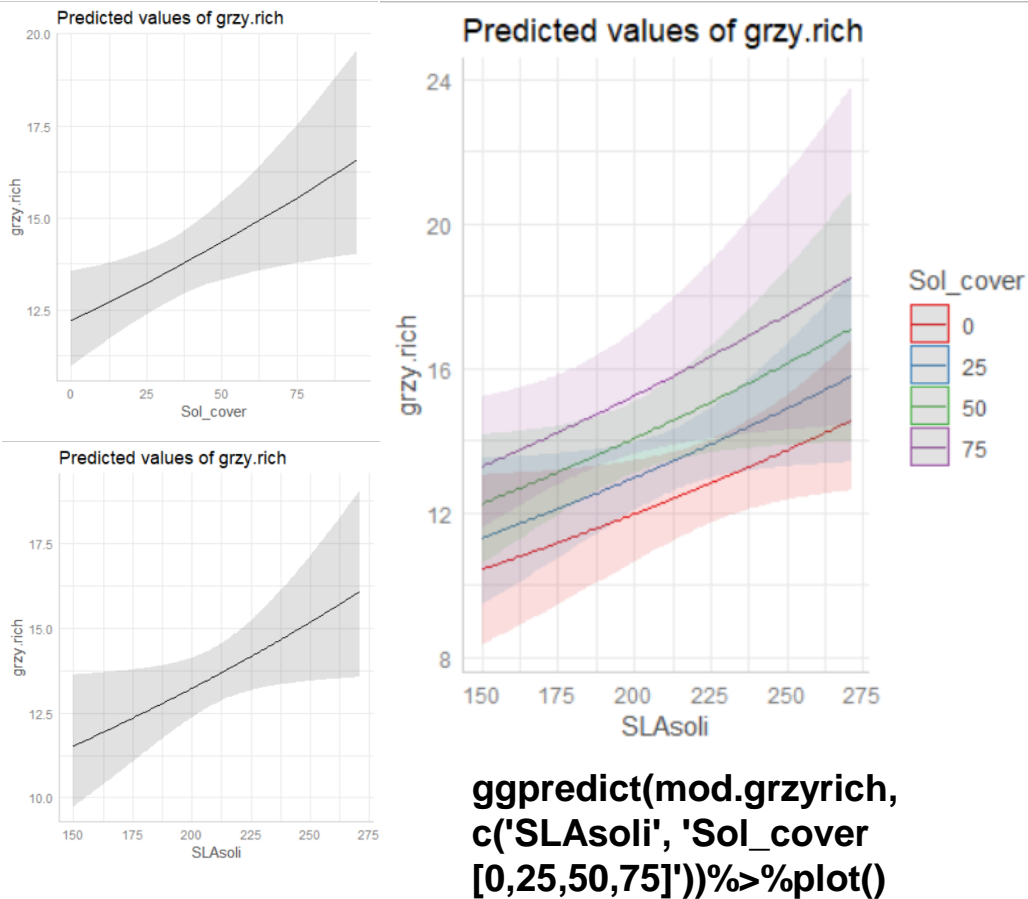
Dispersion parameter for comppois family (): 0.517

Conditional model:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.931034	0.307716	6.275	3.49e-10 ***
Sol_cover	0.003222	0.001290	2.498	0.0125 *
SLAsoli	0.002754	0.001332	2.068	0.0387 *

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.' 0.1 ' ' 1

```
ggpredict(mod.grzyrich, 'Sol_cover')%>%plot()
ggpredict(mod.grzyrich, 'SLAsoli')%>%plot
```



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
- $0 < \text{zmienna zależna} < 1$, 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
```

	cond((Int))	zi((Int))	dsp((Int))	cond(b)	cond(c)	cond(CN)	cond(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

cond – conditional (liczebność), zi – zero-inflated

Najlepszy model: c jako cond 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

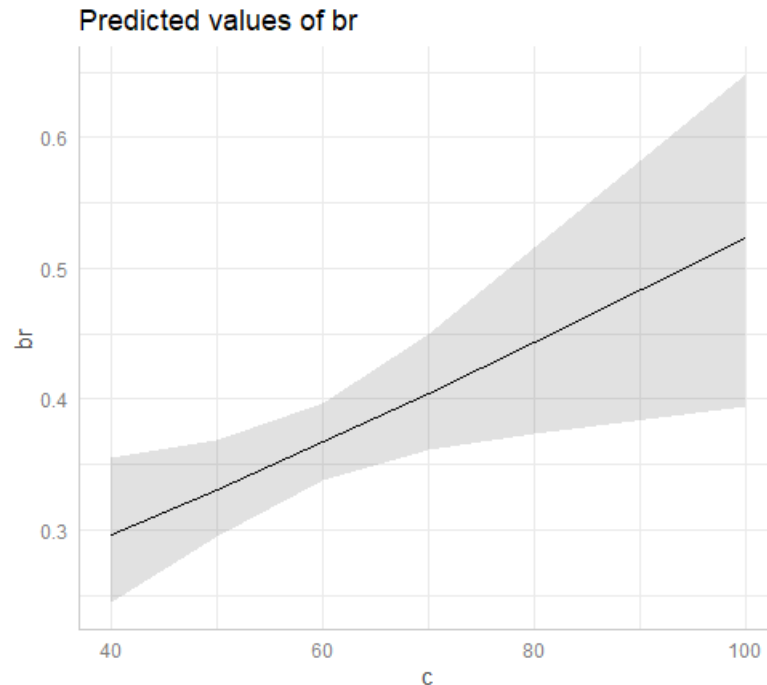
Wielkość efektu?

```
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]



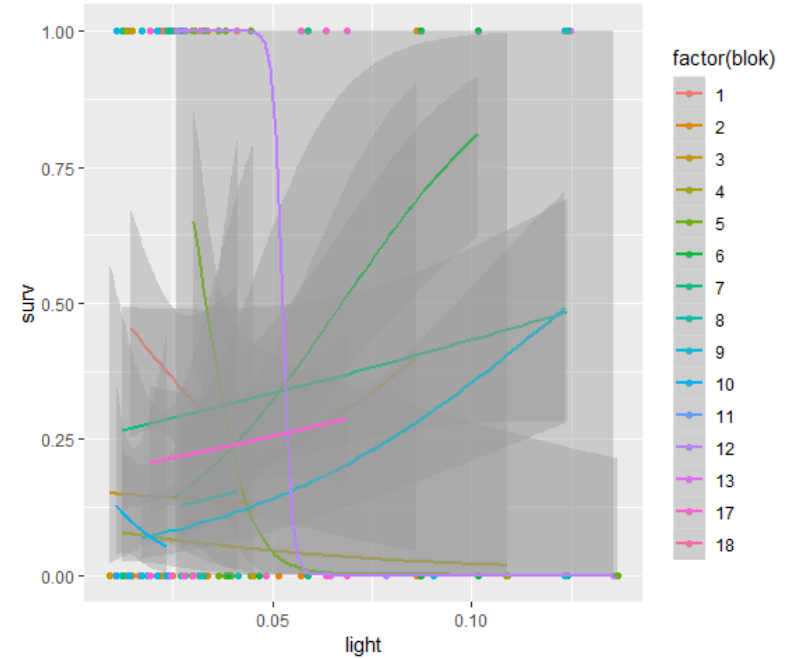
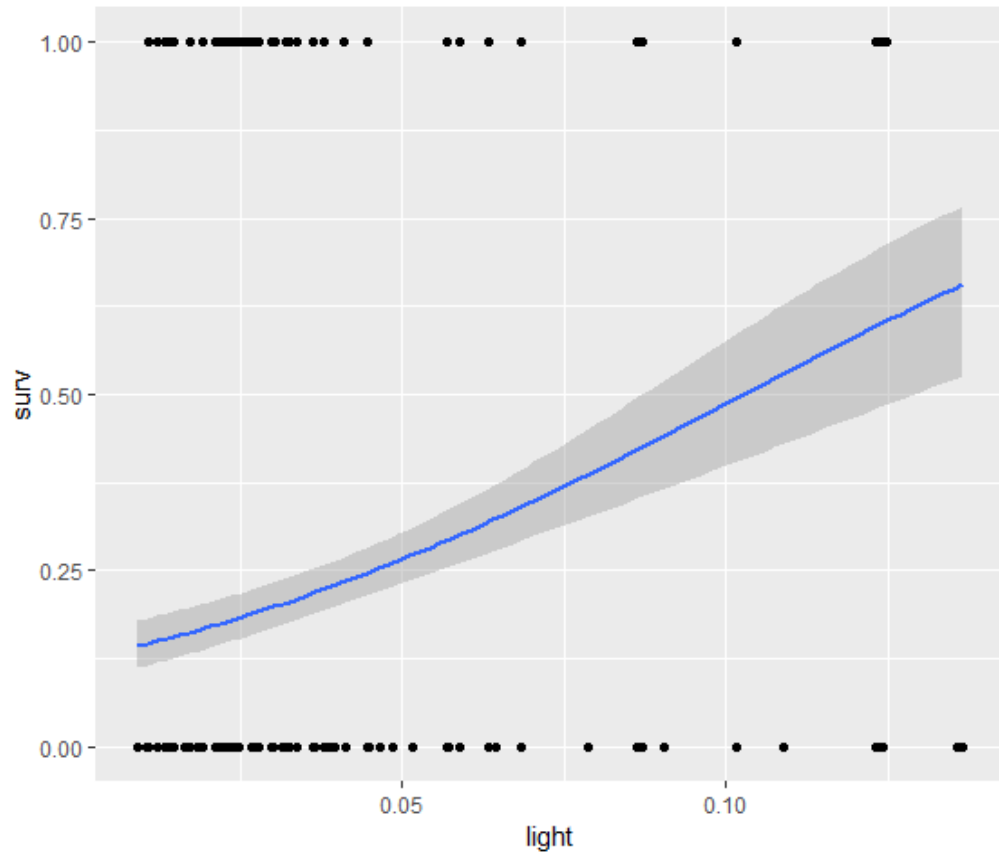
Zwiększenie pokrycia c z 40 do 50% zwiększa proporcję zgryzionych drzewek o 3%,
zwiększenie z 80 na 100% o 8%

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

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

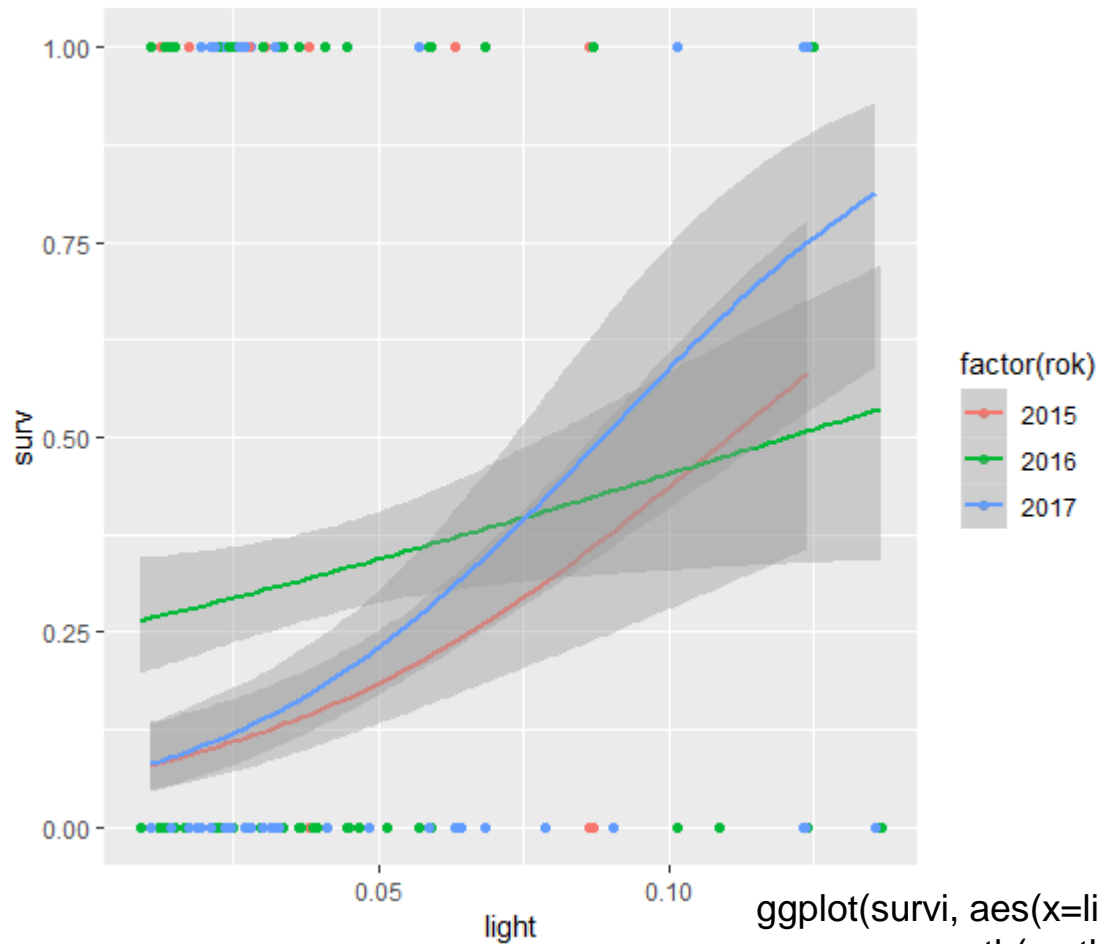
Przykład - przeżywalność siewek dębu czerwonego

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'))
```


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

r.squaredGLMM(mod4)

	R2m	R2c
theoretical	0.07901921	0.2946918
delta	0.06523120	0.2432712

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

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 .
parhind 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:parhind 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
parhind 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:parhind leg	0.04727	1.35836	0.035	0.97224
habforest:partneck	10.40834	444.83724	0.023	0.98133

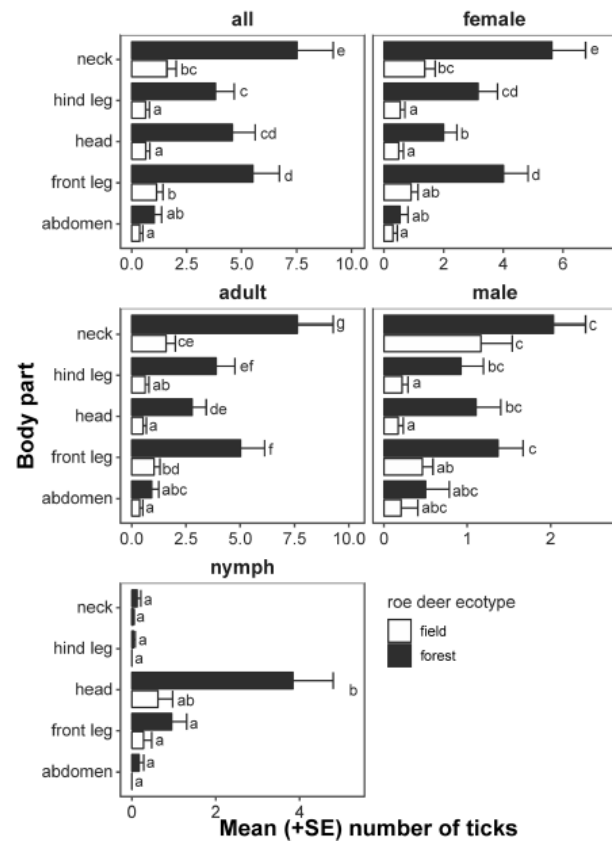
Jak wyciągnąć wartości? cld i emmeans

```
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

Czas liczenia – czasem do kilku godzin dla modelu na $n=40$ (zero-inflation, COM-Poisson)

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 optymalizacją 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>

Przerwa

Zagęszczenie siewek buka w PNGS – przykład

- $n=32$ poletka, każde 100 m²
- 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+de
adAB+(1|riv),data=od.df,family=pois
son(),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

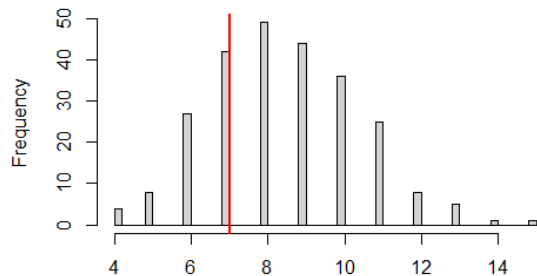
dispersion = 36.199, p-value < 2.2e-16

alternative hypothesis: two.sided

Problem jest raczej z rozkładem niż z zerami

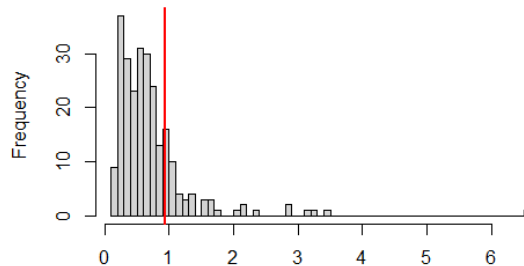
```
mod.fsy1.glob<-glmmTMB(`Fagus sylvatica  
seedlings`~ekto+DIFN+ph+PC1+deadAB  
+(1|riv),data=od.df,family=nbinom1(),na.action=na.fail,  
control=glmmTMBControl(optimizer=optim,optArgs=list(method="BFGS")))
```

DHARMA zero-inflation test via comparison to
expected zeros with simulation under H_0 = fitted
model



Simulated values, red line = fitted model. p-value (two.sided) = 0.648

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



Simulated values, red line = fitted model. p-value (two.sided) = 0.392

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*)

	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

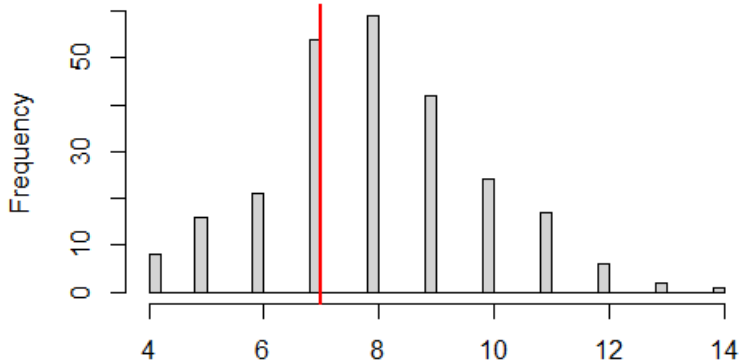
Showing 1 to 14 of 32 entries

Console: E:\Nauka\Projekty badawcze\pnps\pnps2021Banal/

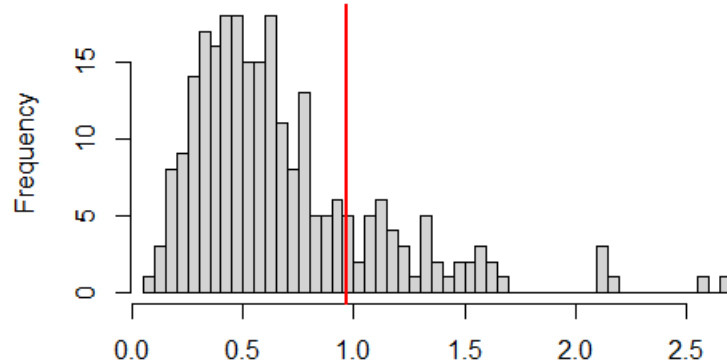
model finalny

```
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")))
```

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



**DHARMA nonparametric dispersion test via sd of
residuals fitted vs. simulated**



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

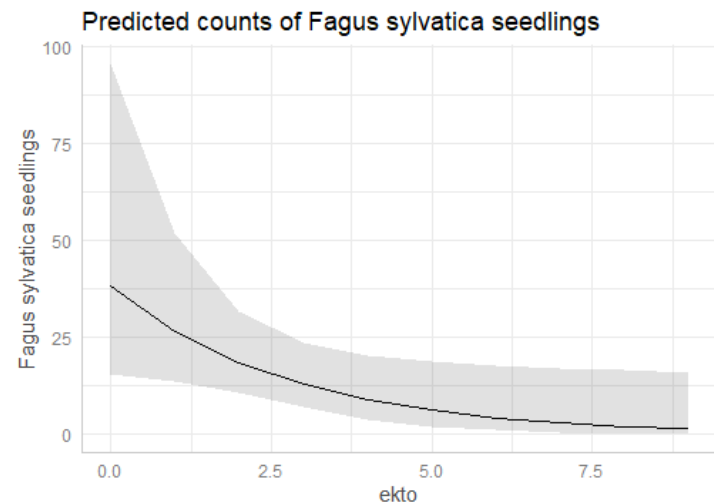
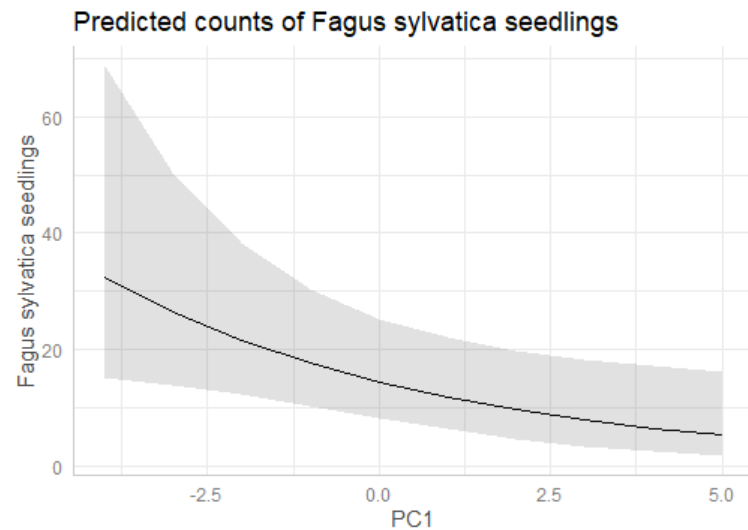
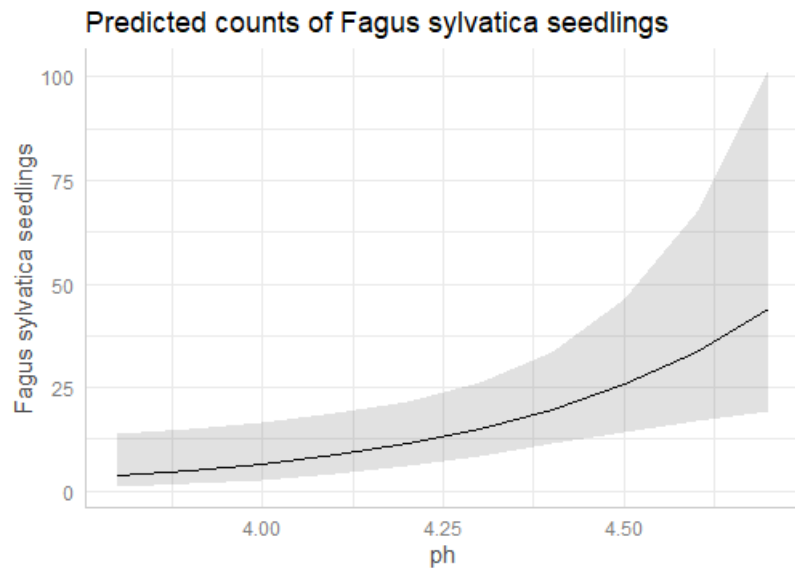
Dispersion parameter for nbinom1 family (): 35.6

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))
```



Przykład – wpływ na krzewy

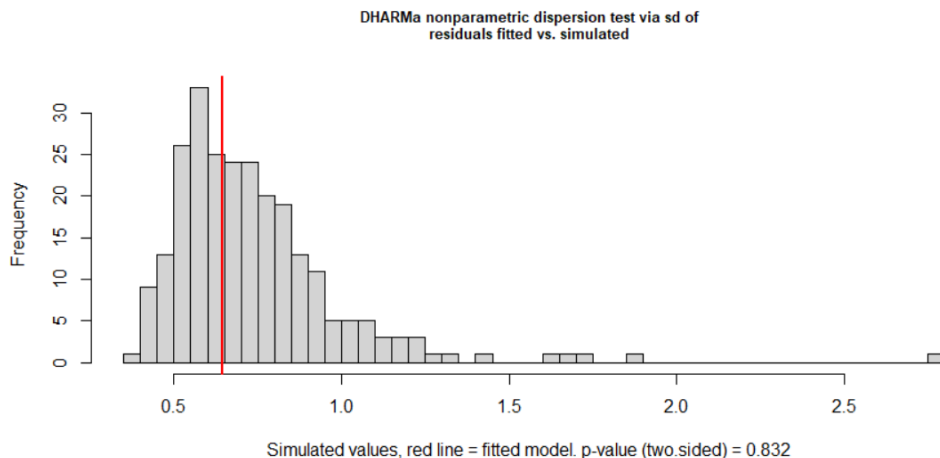
How do invasive trees impact shrub layer diversity and productivity in temperate forests?

Marcin K. Dyderski¹ · Andrzej M. Jagodziński^{1,2}

```
m.rich<-glmer(rich~type+overAB+age+pow+soil+soilpH+soilCN+slope +(1|blok),  
data=gg.ord, family=poisson, glmerControl(optimizer="bobyqa",  
optCtrl=list(maxfun=2e6)), na.action = na.fail)  
#optimizer bo dawał warning  
car::vif(m.rich)
```

	GVIF	Df	GVIF ^{1/(2*Df)}
type	3.888060	8	1.088575
l(overAB/1000)	1.562210	1	1.249884
age	1.463979	1	1.209950
pow	1.258263	1	1.121723
soil	4.992003	6	1.143377
soilpH	2.558514	1	1.599536
soilCN	1.802845	1	1.342701
slope	1.369857	1	1.170409

testDispersion(simulateResiduals(m.rich))



dredge(m.rich)

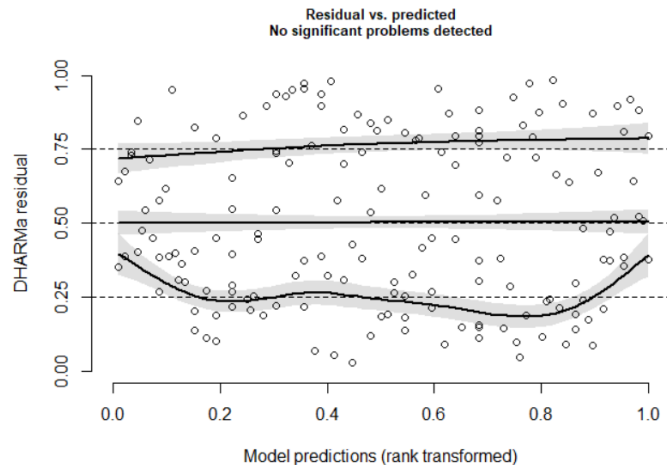
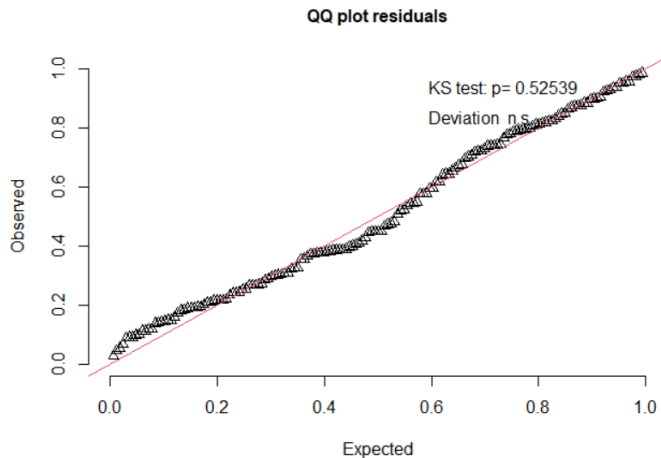
shRubs_hab - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help																
Go to file/function Addins																
revised.R* gg.spe plots dr.abha dredge(m.ABha) dr.Fdis dr.rich dr.FR																
Filter																
	(Intercept)	age	l(overAB/1000)	pow	slope	soil	soilCN	soilpH	type	df	logLik	AICc	delta	weight		
133	0.299174955	NA	NA	2.236406	NA	NA	NA	NA	+	11	-355.6620	735.0495	0.0000000	0.1025738453		
141	0.236121599	NA	NA	2.560151	0.008818338	NA	NA	NA	+	12	-354.7045	735.4616	0.4121247	0.0834727853		
129	0.446852783	NA	NA	NA	NA	NA	NA	NA	+	10	-357.5881	736.6048	1.5552986	0.0471311287		
165	0.495784162	NA	NA	2.255501	NA	NA	-0.007990353	NA	+	12	-355.4491	736.9509	1.9013870	0.0396420121		
135	0.452063165	NA	-3.914611e-04	2.325064	NA	NA	NA	NA	+	12	-355.4522	736.9569	1.9074557	0.0395219070		
173	0.486767166	NA	NA	2.612443	0.009503097	NA	-0.010384005	NA	+	13	-354.3472	737.1050	2.0555294	0.0367015240		
134	0.203684320	0.001078850	NA	2.102653	NA	NA	NA	NA	+	12	-355.5575	737.1676	2.1180995	0.0355710892		
197	0.171626439	NA	NA	2.315079	NA	NA	NA	0.028964288	+	12	-355.5705	737.1937	2.1442095	0.0351097280		
143	0.410038113	NA	-4.514076e-04	2.672586	0.009193715	NA	NA	NA	+	13	-354.4288	737.2682	2.2186909	0.0338262632		
142	0.129370710	0.001198394	NA	2.415666	0.008952063	NA	NA	NA	+	13	-354.5748	737.5601	2.5106646	0.0292316102		
205	0.207515937	NA	NA	2.571967	0.008660209	NA	NA	0.006757537	+	13	-354.6998	737.8101	2.7606611	0.0257968506		
137	0.417827491	NA	NA	NA	0.006005593	NA	NA	NA	+	11	-357.1307	737.9869	2.9374020	0.0236149968		
130	0.245904678	0.002109720	NA	NA	NA	NA	NA	NA	+	11	-357.1521	738.0297	2.9802121	0.0231148880		
161	0.630885789	NA	NA	NA	NA	NA	-0.007418297	NA	+	11	-357.4045	738.5345	3.4850493	0.0179584065		

po dredge() – model finalny

```
m.rich<-glmer(rich~type+pow+(1|blok),  
gg.ord,family=poisson,glmerControl(optimizer="bobyqa",  
optCtrl=list(maxfun=2e6)),na.action = na.fail)  
plot(simulateResiduals(m.rich))
```

DHARMa residual



summary(m.rich)

Generalized linear mixed model fit by maximum likelihood
(Laplace Approximation) [
glmerMod]

Family: poisson (log)

Formula: rich ~ type + pow + (1 | blok)

Data: gg.ord

Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun
= 2e+06))

AIC	BIC	logLik	deviance	df.resid
733.3	767.5	-355.7	711.3	154

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.80538	-0.62615	-0.05085	0.45591	2.31642

Random effects:

Groups Name	Variance	Std.Dev.
blok (Intercept)	0.2151	0.4637

Number of obs: 165, groups: blok, 21

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.2992	0.2806	1.066	0.286332
typePinus poor	1.1945	0.3506	3.407	0.000656 ***
typePinus poor - Ps	1.0788	0.2885	3.739	0.000185 ***
typePinus rich	1.1043	0.2562	4.311	1.63e-05 ***
typePinus rich - Ps	1.3298	0.2789	4.769	1.85e-06 ***
typeQ. petraea	0.8720	0.3016	2.891	0.003838 **
typeQ. rubra	0.5930	0.3084	1.923	0.054491 .
typeQuercus-Acer-Tilia	1.2184	0.2769	4.399	1.09e-05 ***
typeRobinia	1.0889	0.2878	3.783	0.000155 ***
pow	2.2364	1.1208	1.995	0.045996 *

```
cld(emmeans(m.rich, ~type,type='resp'))
```

type	rate	SE	df	asympt.LCL	asympt.UCL	.group
Fagus	1.56	0.422	Inf	0.922	2.65	1
Q. rubra	2.83	0.607	Inf	1.858	4.31	12
Q. petraea	3.74	0.745	Inf	2.532	5.53	123
Pinus poor - Ps	4.60	0.752	Inf	3.339	6.34	23
Robinia	4.65	0.826	Inf	3.279	6.58	23
Pinus rich	4.72	0.604	Inf	3.672	6.06	23
Pinus poor	5.16	1.318	Inf	3.131	8.52	23
Quercus-Acer-Tilia	5.29	0.843	Inf	3.870	7.23	23
Pinus rich - Ps	5.91	0.917	Inf	4.362	8.01	3

Species richness

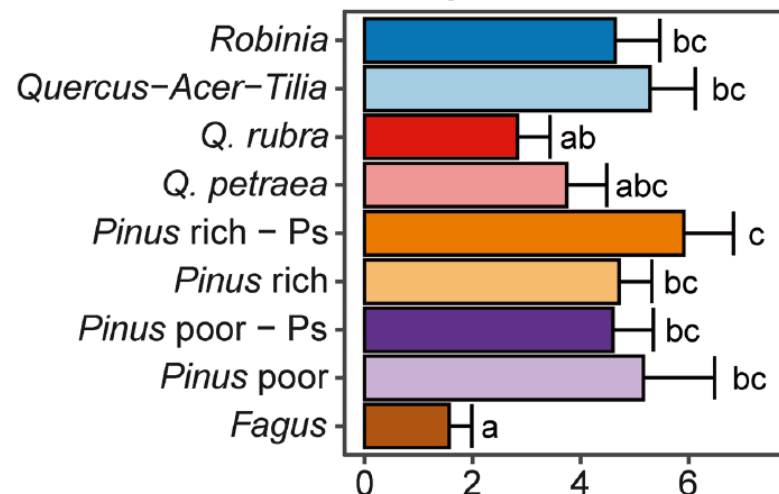


Table 4 Models of shrub layer alpha diversity, estimated using Gaussian distribution GLMMs (for species richness – Poisson distribution and z statistic instead of t). Significant variables were bolded. AICc – Akaike Information Criterion, AICc₀ – AIC of null model (intercept-only), AICc_{full} – AIC of full model (including all hypothesized

variables), R²_m – conditional coefficient of determination, AICc₀ = 199.5, R²_m = 0.

Response	Variable	Estimate	SE	t value	Pr(> t)
Species richness	(Intercept)	0.2992	0.2806	1.0660	0.2863
Block SD = 0.464	type = <i>Pinus</i> poor	1.1945	0.3506	3.4070	0.0007
R ² _m = 0.185	type = <i>Pinus</i> poor—Ps	1.0788	0.2885	3.7390	0.0002
R ² _c = 0.611	type = <i>Pinus</i> rich	1.1043	0.2562	4.3110	<0.0001
AICc = 735.0	type = <i>Pinus</i> rich—Ps	1.3298	0.2789	4.7690	<0.0001
AICc _{full} = 749.7	type = <i>Q. petraea</i>	0.8720	0.3016	2.8910	0.0038
AICc ₀ = 757.4	type = <i>Q. rubra</i>	0.5930	0.3084	1.9230	0.0545
	type = <i>Quercus-Acer-Tilia</i>	1.2184	0.2769	4.3990	<0.0001
	type = <i>Robinia</i>	1.0889	0.2878	3.7830	0.0002
	plot area	2.2364	1.1208	1.9950	0.0460

Przykład – proporcja grzybów patogenicznych

```
mod.grzypato<-glmmTMB(pato~Sol_cover+Sol_cover*SLAsoli+  
Sol_cover*Hsoli+(1|locality),data=soli,na.action = na.fail, family=beta_family())
```

```
#wcześniej sprawdzone VIFy
```

```
dredge(mod.grzypato)
```

```
#model końcowy
```

```
mod.grzypato<-glmmTMB(pato~Sol_cover+(1|locality),  
data=soli,na.action = na.fail, family=beta_family())
```

Conditional model:

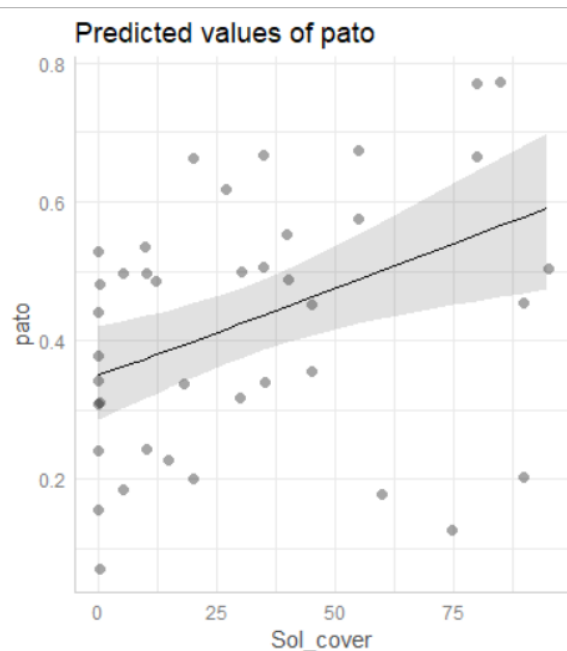
Groups	Name	Variance	Std.Dev.
locality	(Intercept)	5.22e-11	7.225e-06

Number of obs: 40, groups: locality, 2

Dispersion parameter for beta family (:): 8.32

Conditional model:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.616894	0.153495	-4.019	5.84e-05 ***
Sol_cover	0.010379	0.003471	2.990	0.00279 **



```
ggpredict(mod.grzypato, 'Sol_cover')%>%plot(add.data=T)
```

Przykład – jaka część żołądźzi opada w roku?

```
btr.zoib<-glmmTMB(csumrel.tr~type+month+(1|trapID)+(1|year), ziformula =  
~month+(1|year)+(1|trapID),family=beta_family(),data=btr2,na.action = na.fail)
```

dużo zer
zero-inflation

Table 4

Generalized linear mixed-effects model assuming the zero-inflated Beta distribution of the cumulative proportion of acorn fall mass.

Model part	Predictor	Estimate	SE	Z	p	Random effect/AIC	SD
Count	(Intercept)	-6.8933	0.4359	-15.814	<0.0001	trap ID	0.3572
	TFS=DQ	-0.4759	0.1758	-2.708	0.0068	year	0.5431
	TFS=MQ	-0.2609	0.1719	-1.518	0.1290	AIC	-1291.9
	Month	0.8189	0.0340	24.054	<0.0001	AIC ₀	488.9
zero-inflation	(Intercept)	8.7963	0.6886	12.780	<0.0001	year	0.4709
	Month	-1.3041	0.0839	-15.540	<0.0001	trap ID	1.1870

Abbreviations: TFS – the type of the forest site, SE – standard error, Z – test statistic, AIC – Akaike Information Criterion, AIC₀ – AIC of the null model, SD – standard deviation determination.

Agricultural and Forest Meteorology 327 (2022) 109228



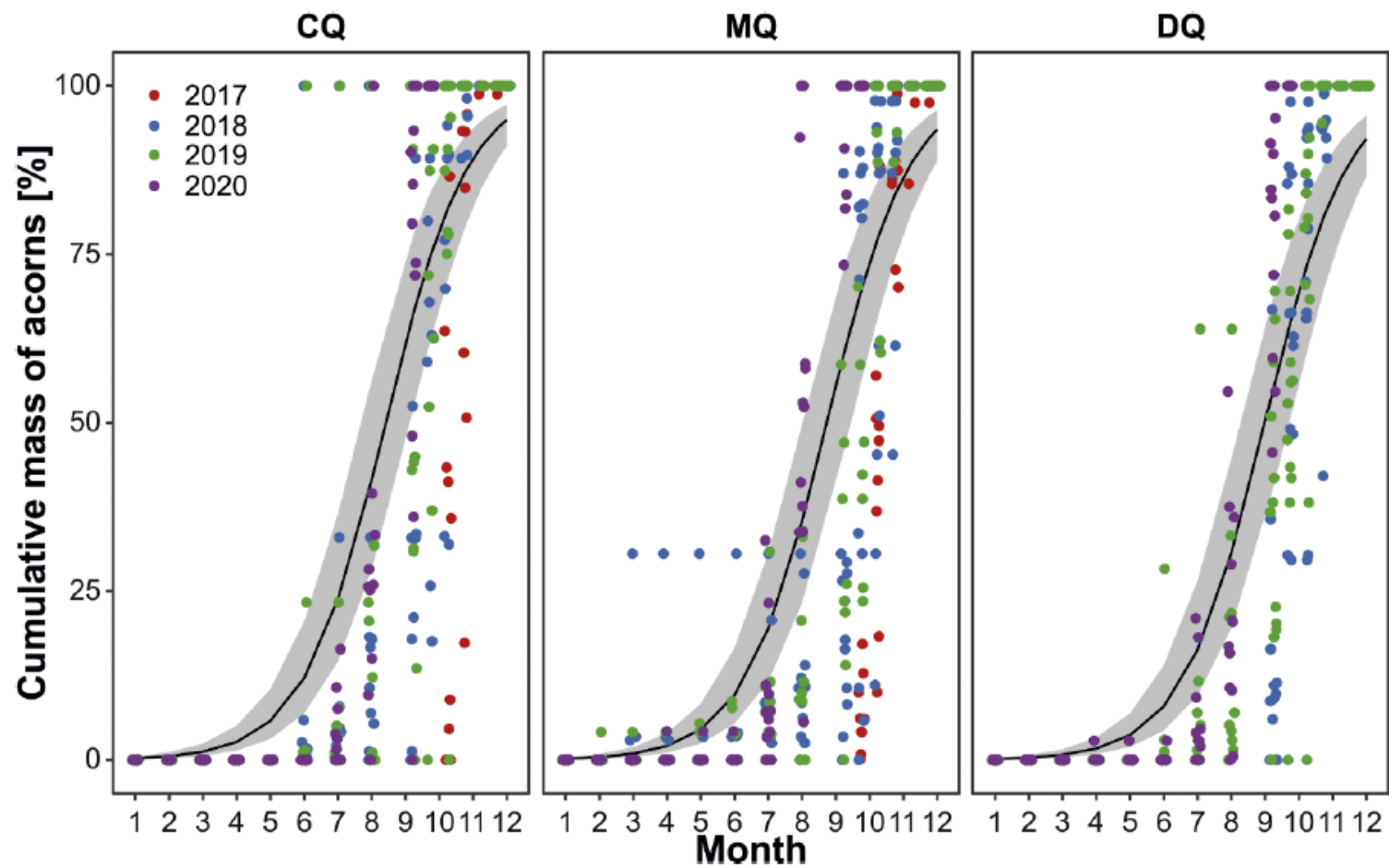
Contents lists available at ScienceDirect

Agricultural and Forest Meteorology

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

Acorn production of introduced *Quercus rubra* is more strongly impacted by the weather than by the forest site (a case study from Poland)

Anastazja Greda^a, Beata Woźniowa^{a,*}, Marcin K. Dyderski^b, Andrzej M. Jagodziński^b, Lee E. Frelich^c



Uogólnione modele addytywne - GAM

Generalized Additive Models

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

Co zamiast funkcji liniowych? Funkcje sklepane (splines)

Zamiast składników liniowych - składniki nieliniowe

Wada: słabsza możliwość wykorzystania, trudniejsze interpretacyjnie

Funkcje sklejane (splines)

dla $x \in (-\infty, -10)$ $y = 12x + 2$

dla $x \in [-10, 2)$ $y = 17^x$

dla $x \in [2, 7)$ $y = 2 - x^3$

dla $x \in [7, \infty)$ $y = 2x^2 - 3.122$

mamy cztery funkcje, sklejone w jedną (wygładzoną)

Mamy trzy punkty przegięcia - w których funkcja zmienia bieg

Przykład zastosowania

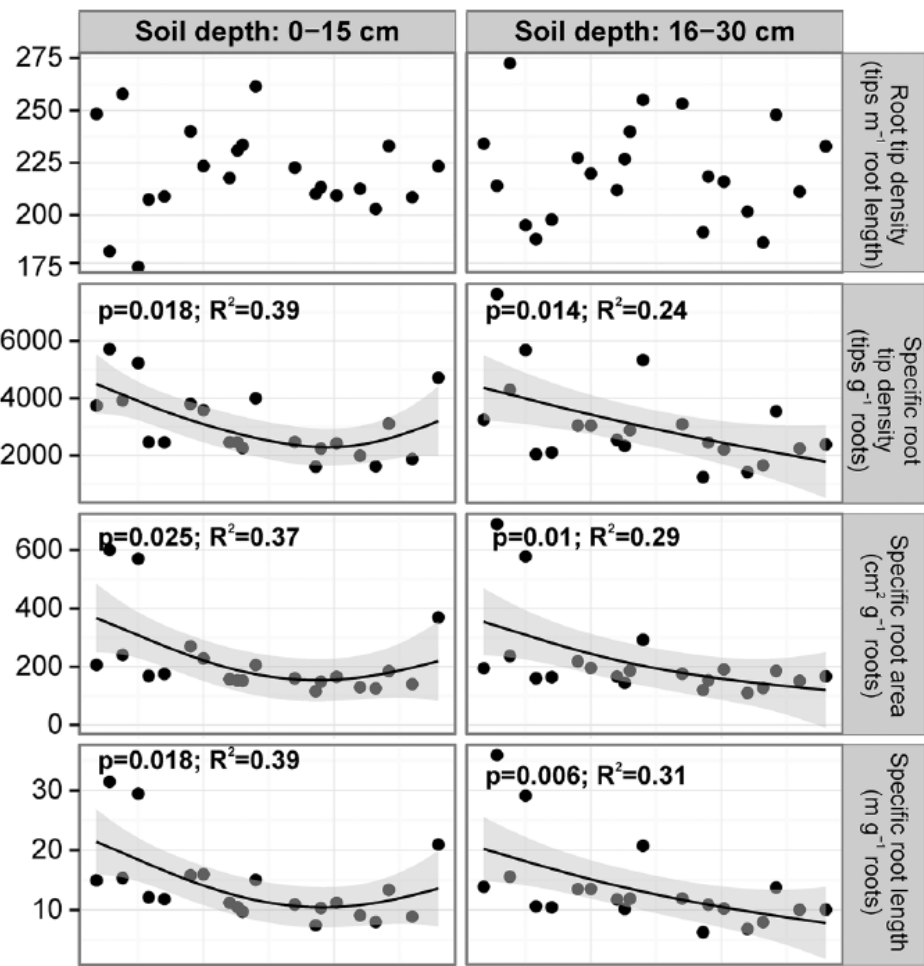
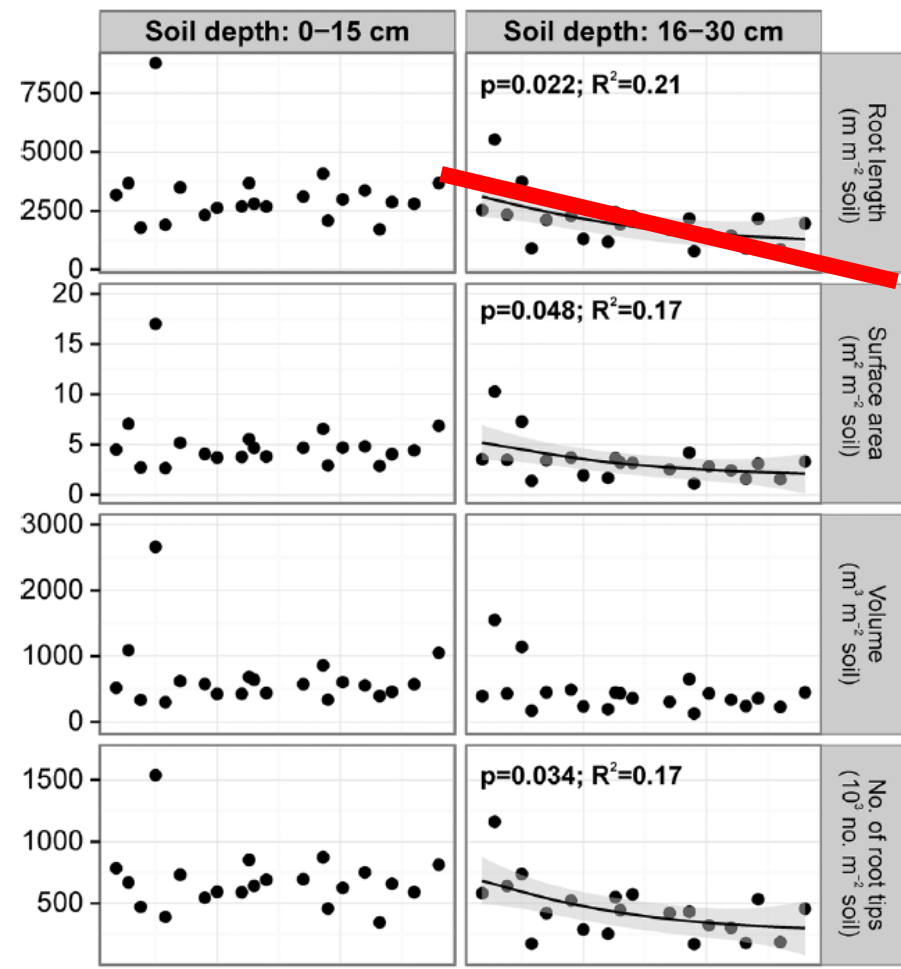
wzorce rozwoju korzeni drobnych

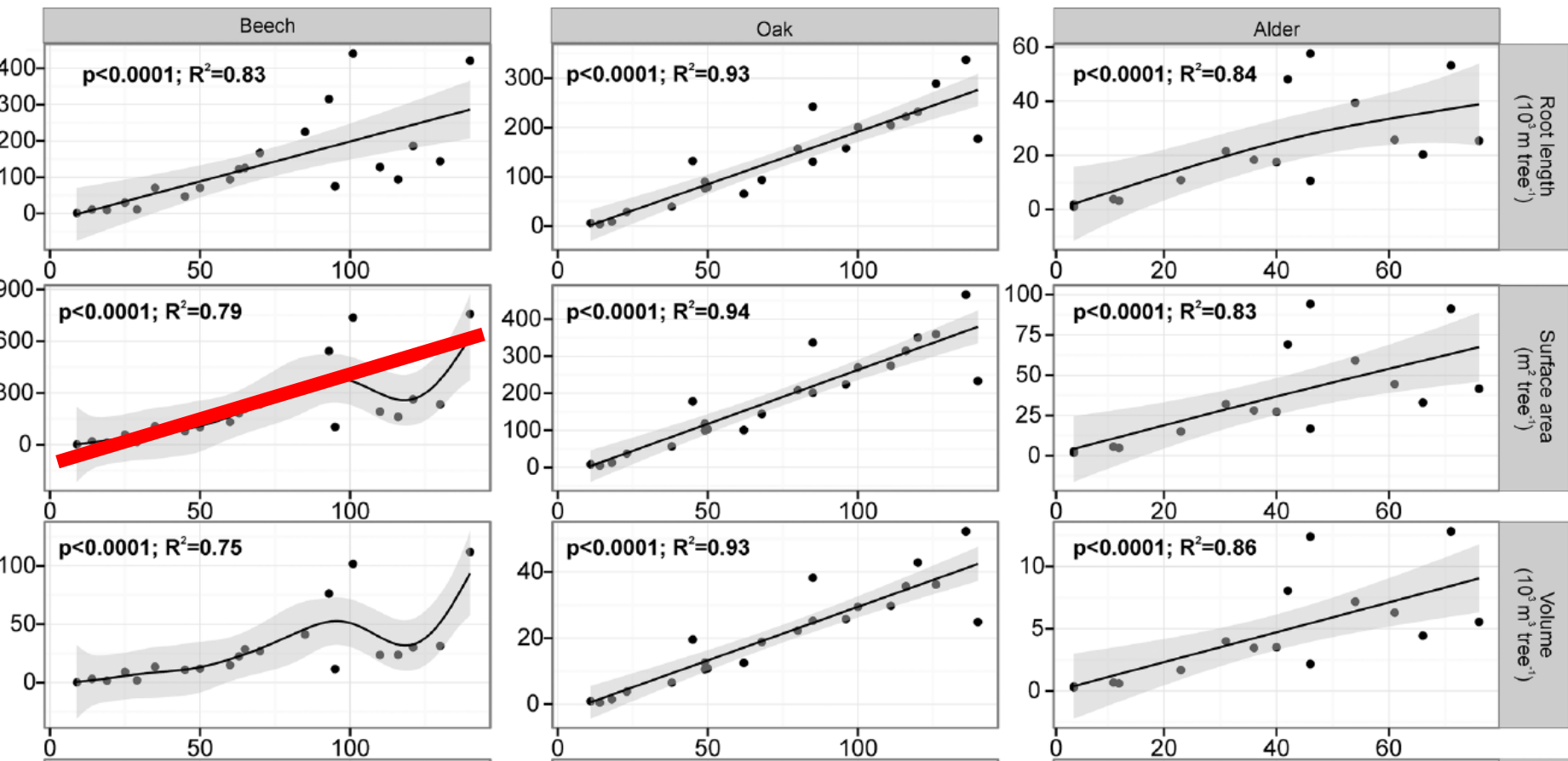


RESEARCH ARTICLE

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 Ziółkowski², Aleksandra Warnkowska², Hubert Prais²






```

> 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

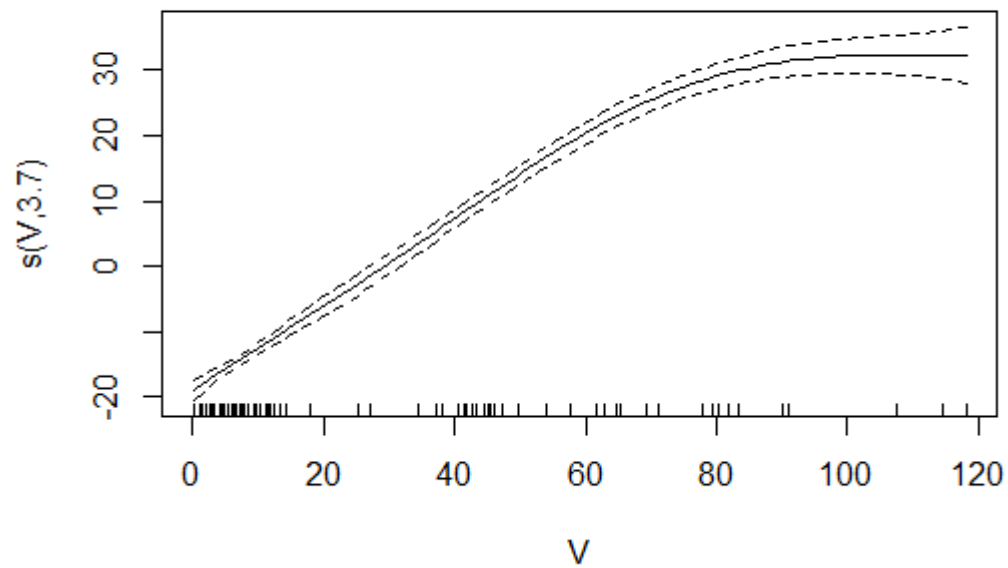
jest to wielomian
stopnia 3,7
(patrzymy na edf)

edf - estimated degrees
of freedom

Ref.df - reference
degrees of freedom

F - statystyka testowa

plot(model1)



```
> 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

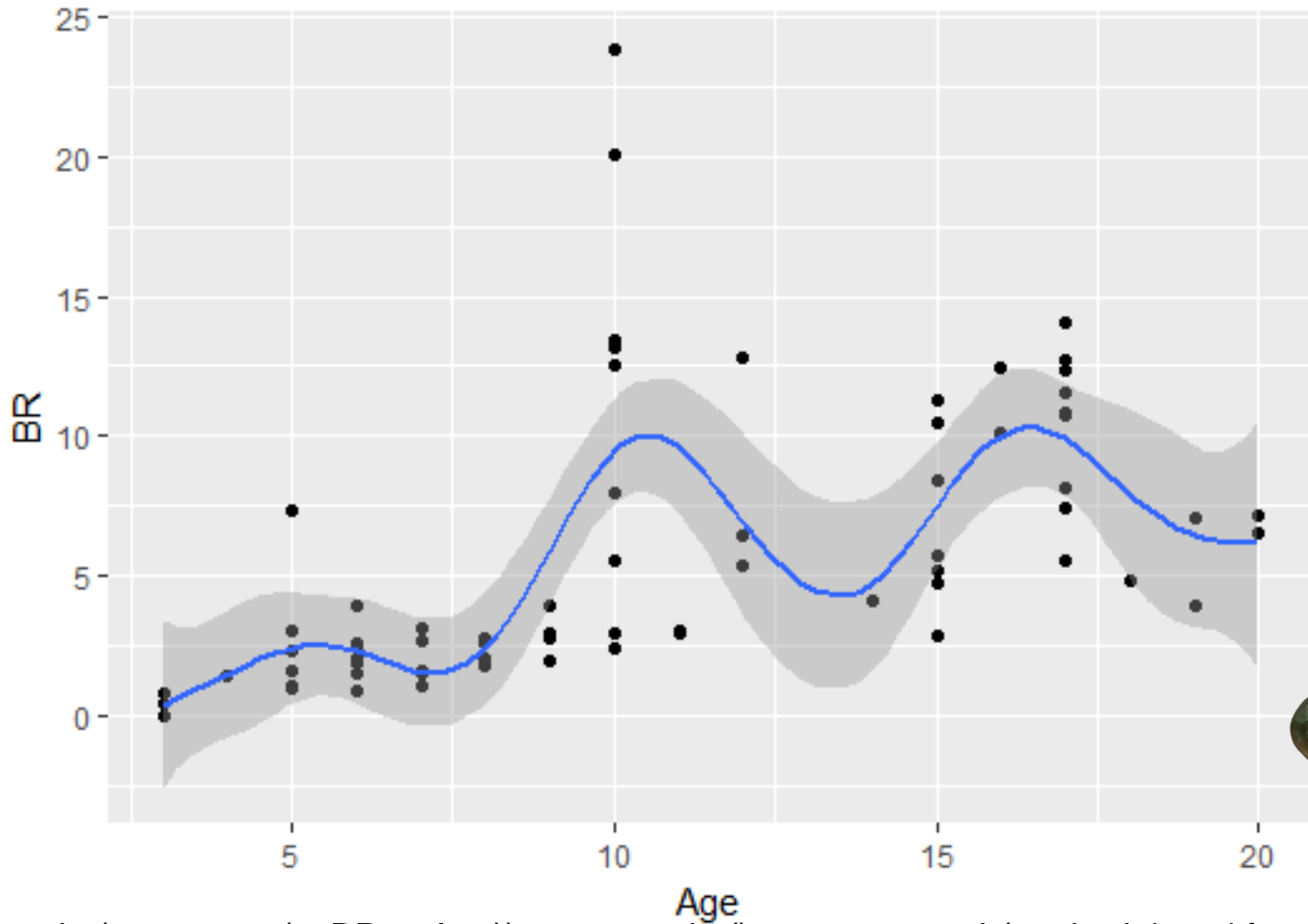
Approximate significance of smooth terms:

	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

```
> |
```



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.

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

Jonathan 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: "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 parameterize 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 separately² 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 k th 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. Székely *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. Wei¹ 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

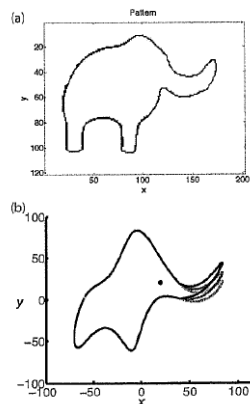


Fig. 1. (a) Outline of an elephant. (b) Three snapshots of the wagging trunk.

~John vonNeumann

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

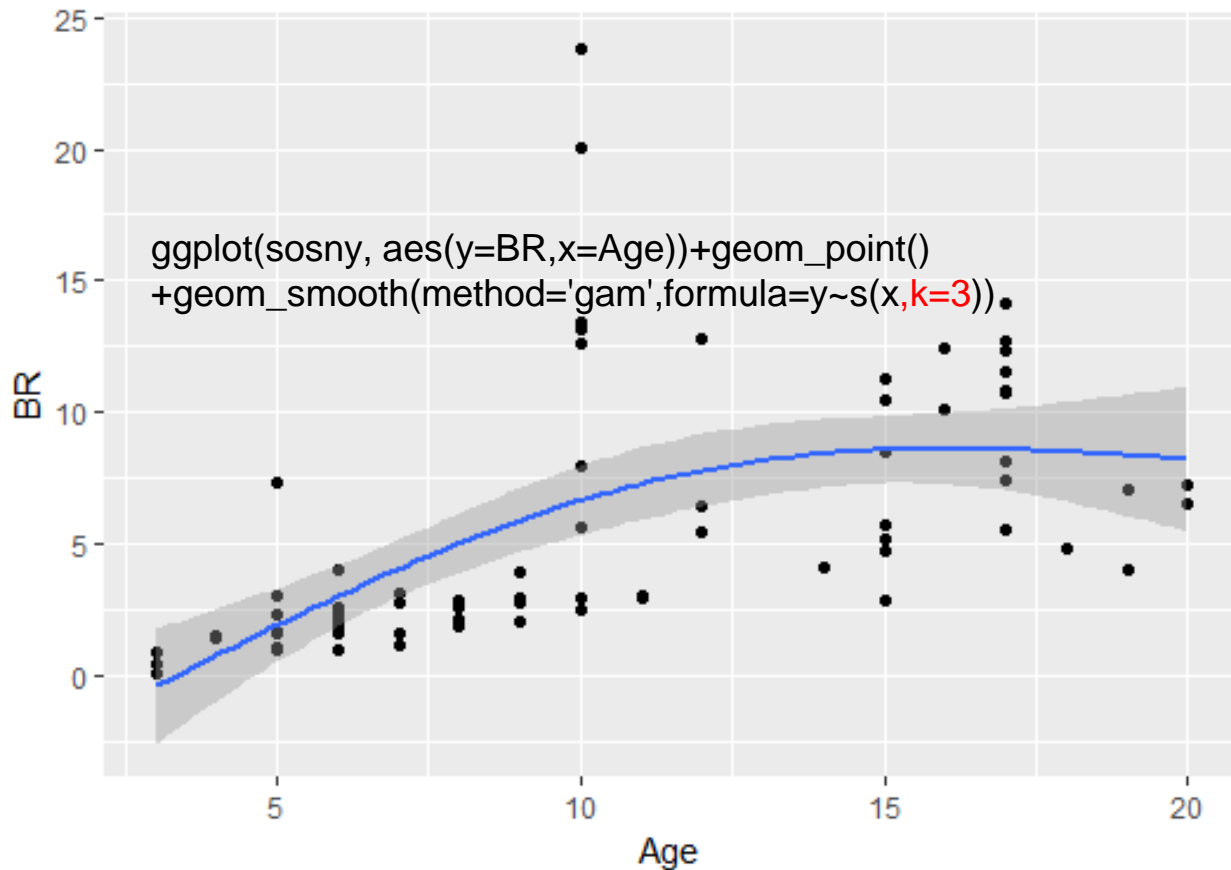
overfitting - model dobrze działa na zbiorze treningowym, ale źle na testowym
brak możliwości ekstrapolacji

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:

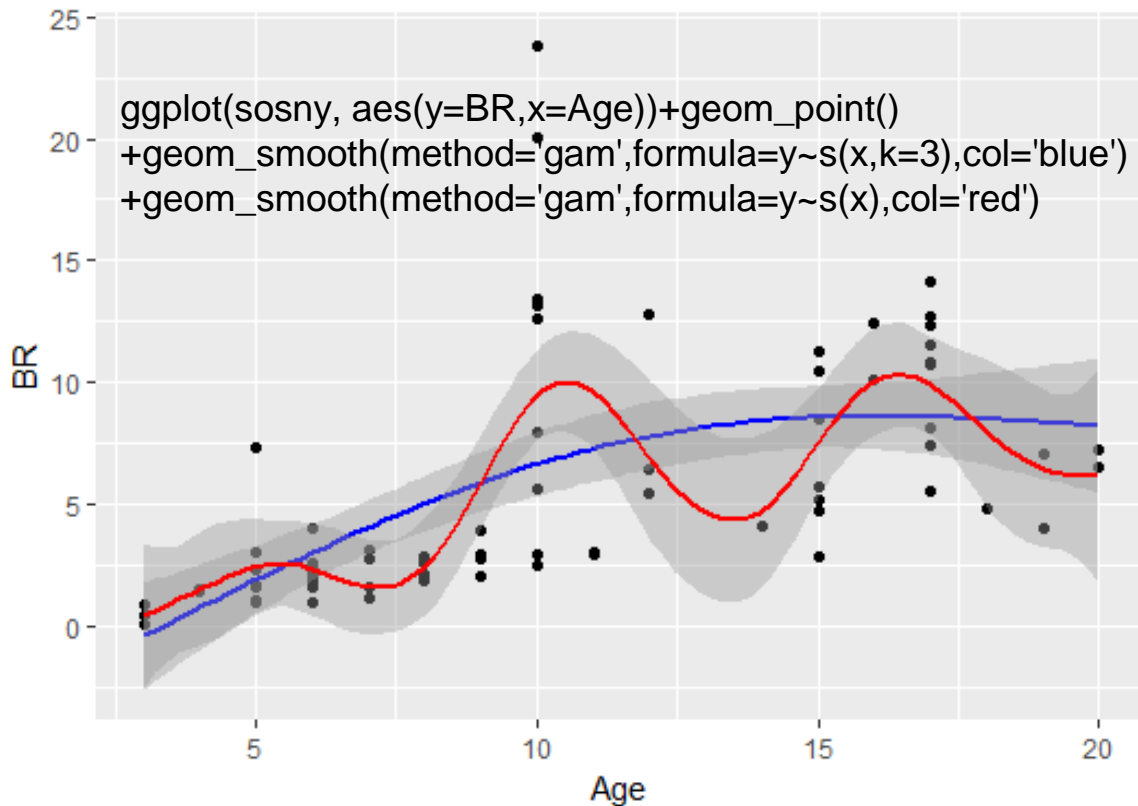
	edf	Ref.df	F	p-value
s(Age)	1.87	1.983	22.44	6.6e-08 ***

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 jakości ale model jest sensowniejszy



Przykład – udział gatunków *Molinia*

Received: 31 March 2020 | Revised: 9 December 2020 | Accepted: 16 December 2020
DOI: 10.1111/jvs.12983

RESEARCH ARTICLE

Journal of Vegetation Science
IAVS

TABLE 1 Generalised Additive Model describing the proportion of species typical of *Molinia* meadows as a function of SLA CWM, and standardised effect size (SES) of functional diversity components

Parametric co-efficients	Estimate	SE
(Intercept)	-1.139	0.053
Approximate significance of smooth terms:		
	Estimated <i>df</i>	Reference <i>df</i>
SES functional dispersion	1.577	1.821
SES functional divergence	1.828	1.970
CWM of SLA	1.000	1.000

The best model was chosen according to the Akaike Information Criterion corrected for small sample sizes (AICc; full model AICc = -121.1, null model [intercept only] AICc = -119.3, final model AICc = -133.6). CWM, community-weighted mean; SLA, specific leaf area; *df*, degree of freedom.

Niche differentiation, competition or habitat filtering? Mechanisms explaining co-occurrence of plant species on wet meadows of high conservation value

CZORTEK ET AL. | Patryk Czortek¹ | Anna Orczewska² | Marcin K. Dyderski³

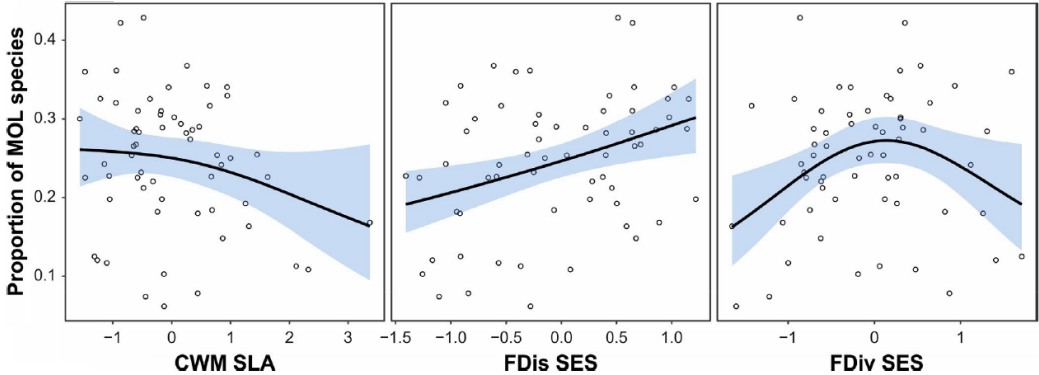


FIGURE 2 Partial regressions of the final model with community-weighted mean value (CWM) of specific leaf area (SLA) and standardised effect sizes (SES) of functional diversity components (FDiv, functional divergence; FDis, functional dispersion) as predictors influencing the proportion of species typical of *Molinia* meadows in vegetation plots. For model parameters see Table 1

Podsumowanie - algorytm działania

1. Jaki jest rozkład tego co modelujesz? W zależności zbuduj model globalny:
 1. Ciągły, zbliżony do normalnego – zrób LM/LMM (Gaussian)
 2. 0/1 – binomial (glm, glmer, glmmTMB)
 3. 0.00-0.9(9) proporcja – beta_family (glmmTMB)
 1. Czy masz wartości 1? jeśli tak to zamień na 0.99
 2. Czy masz wartości 0? jeśli tak to dodaj parametr ziformula
 4. dane zliczeniowe (1,2,3...) – Poisson (glm, glmer, glmmTMB)
 1. czy są zera? jeśli tak to sprawdź czy jest problem z zero-inflation używając `DHARMA::testZeroInflation()`
 2. jak wygląda kwestia dyspersji? sprawdź `DHARMA::testDispersion()`
 1. wartość bliska 1.0 i $p > 0.05$ – zostań przy Poissonie
 2. wartość < 1.0 i $p < 0.05$ – COM-Poisson (glmmTMB, family=compois)
 3. wartość > 1.0 i $p < 0.05$ – negative binomial (glmmTMB, family=nbinom1)
5. inny, dziwny rozkład – zrób GAM, ale upewnij się że wady (brak effect size i nieparametryczność) nie przysłonią zalet

Podsumowanie - algorytm działania

2. Dla modelu globalnego (ze wszystkimi potencjalnymi predyktorami) sprawdź vify. Uwaga: `car::vif()` zadziała na `glmer()`, nie zadziała z `glmmTMB`. Sprawdź na jakimkolwiek modelu `glmer()` ten sam zestaw predyktorów
3. Sprawdź `warnings()` – jeśli są problemy dodaj optyimizer lub zmniejsz liczbę zmiennych/stopień skomplikowania modeli
4. Przeprowadź selekcję modeli za pomocą `MuMIn::dredge()`
5. Wybierz model finalny, obejrzyj `summary` i `ggpredicty`
6. Jakie predyktory chcesz pokazać?
 2. kategoryczne – zrób `emmeans()` i pokaż je na barplotach
 3. numeryczne – użyj `ggpredict()` i pokaż je
 4. interakcję – użyj `ggpredict()` do pokazania interakcji między dwiema zmiennymi, pamiętaj o zdefiniowaniu poziomów drugiej zmiennej

... 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

