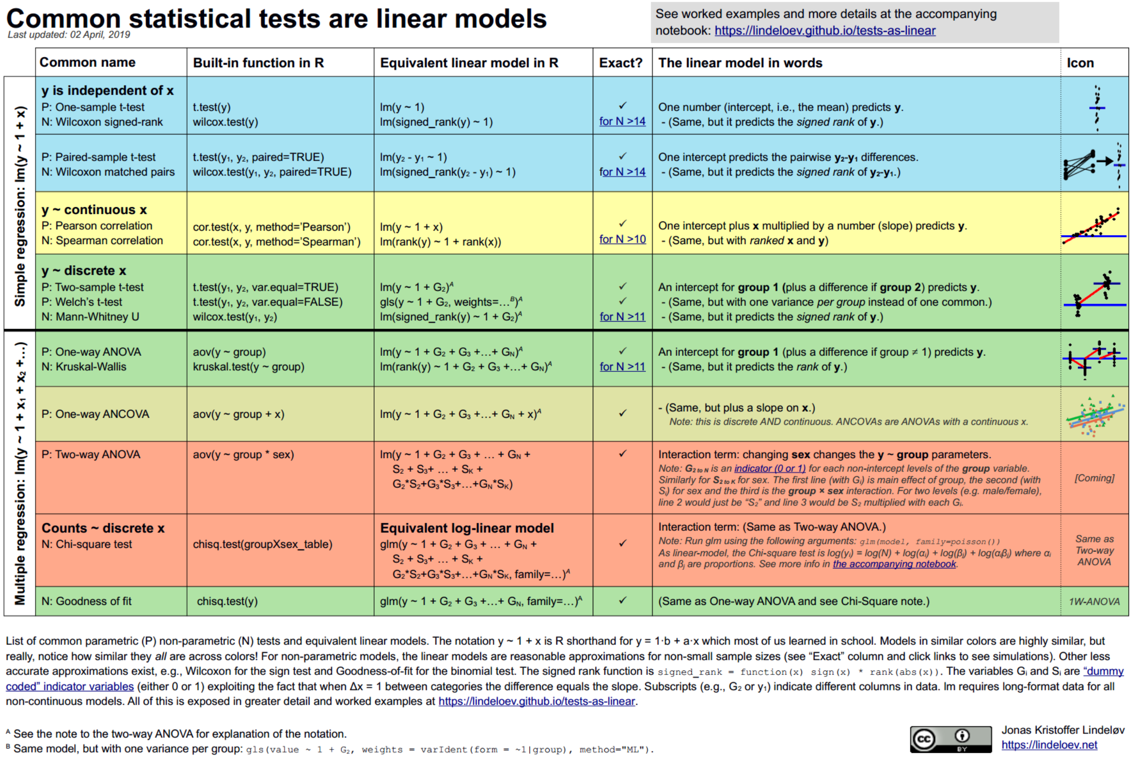
Review of linear models

12 Sep 2024

```r
## it's nice to include packages at the top
## (and NOT automatically install them)
## try not to carry over packages you don't use
library(faux)
library(brglm2) ## for lizards data
library(ggplot2); theme_set(theme_bw())
## diagnostics
library(performance)
library(DHARMa)
## downstream model evaluation
library(broom)
library(dotwhisker)
library(emmeans)
library(effects)
library(marginaleffects)
library(parameters)
## library(ggeffects)
```

Why linear models? (Lindeløv 2019)



Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon

Simple regression: lm(y ~ 1 + x)	**y is independent of x**
	P: One-sample t-test
	N: Wilcoxon signed-rank
		t.test(y)
		wilcox.test(y)
			lm(y ~ 1)
			lm(signed_rank(y) ~ 1)
				✓
				for N >14
					One number (intercept, i.e., the mean) predicts **y**.
					- (Same, but it predicts the *signed rank* of **y**.)	

	P: Paired-sample t-test
	N: Wilcoxon matched pairs
		t.test(y_1, y_2, paired=TRUE)
		wilcox.test(y_1, y_2, paired=TRUE)
			lm(y_2 - y_1 ~ 1)
			lm(signed_rank(y_2 - y_1) ~ 1)
				✓
				for N >14
					One intercept predicts the pairwise **y_2-y_1** differences.
					- (Same, but it predicts the *signed rank* of **y_2-y_1**.)	

	y ~ continuous x
	P: Pearson correlation
	N: Spearman correlation
		cor.test(x, y, method='Pearson')
		cor.test(x, y, method='Spearman')
			lm(y ~ 1 + x)
			lm(rank(y) ~ 1 + rank(x))
				✓
				for N >10
					One intercept plus **x** multiplied by a number (slope) predicts **y**.
					- (Same, but with *ranked* **x** and **y**)	

	y ~ discrete x
	P: Two-sample t-test
	P: Welch's t-test
	N: Mann-Whitney U
		t.test(y_1, y_2, var.equal=TRUE)
		t.test(y_1, y_2, var.equal=FALSE)
		wilcox.test(y_1, y_2)
			lm(y ~ 1 + G_2)^A
			gls(y ~ 1 + G_2, weights=…^{B})^A
			lm(signed_rank(y) ~ 1 + G_2)^A
				✓
				✓
				for N >11
					An intercept for **group 1** (plus a difference if **group 2**) predicts **y**.
					- (Same, but with one variance *per group* instead of one common.)
					- (Same, but it predicts the *signed rank* of **y**.)	

Multiple regression: lm(y ~ 1 + x_1 + x_2 +…)	P: One-way ANOVA
	N: Kruskal-Wallis
		aov(y ~ group)
		kruskal.test(y ~ group)
			lm(y ~ 1 + G_2 + G_3 +…+ G_N)^A
			lm(rank(y) ~ 1 + G_2 + G_3 +…+ G_N)^A
				✓
				for N >11
					An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**.
					- (Same, but it predicts the *rank* of **y**.)	

	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +…+ G_N + x)^A	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: **$G_{2 \text{ to } N}$** is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for **$S_{2 \text{ to } K}$** for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K, family=…)^A	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments:* `glm(model, family=poisson())` *As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +…+ G_N, family=…)^A	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

^{A} See the note to the two-way ANOVA for explanation of the notation.
^{B} Same model, but with one variance per group: `gls(value ~ 1 + G₂, weights = varIdent(form = ~1|group), method="ML").`

Jonas Kristoffer Lindeløv
https://lindeloev.net

Basics

- assume $\mathbf{y} \sim \text{Normal}(\mathbf{X}\beta, \sigma)$[1]
- $\mathbf{X}$ is the *model matrix*, can be anything we want it to be
- the *Gauss-Markov theorem* (Wikipedia) makes weaker assumptions: $\mathbf{y} = \mathbf{X}\beta + \epsilon$; as long as ϵ is mean-zero, homoscedastic with finite variance, and uncorrelated … then the OLS solution
$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$
is the BLUE (or MVUE).
- we'll embrace the assumptions (which are needed for inference!)

Computation

- matrix decompositions (QR with pivoting; see here)
- big problems: `biglm`, `speedglm`, `RcppEigen::fastLm`

[1] Notation-abuse warning …

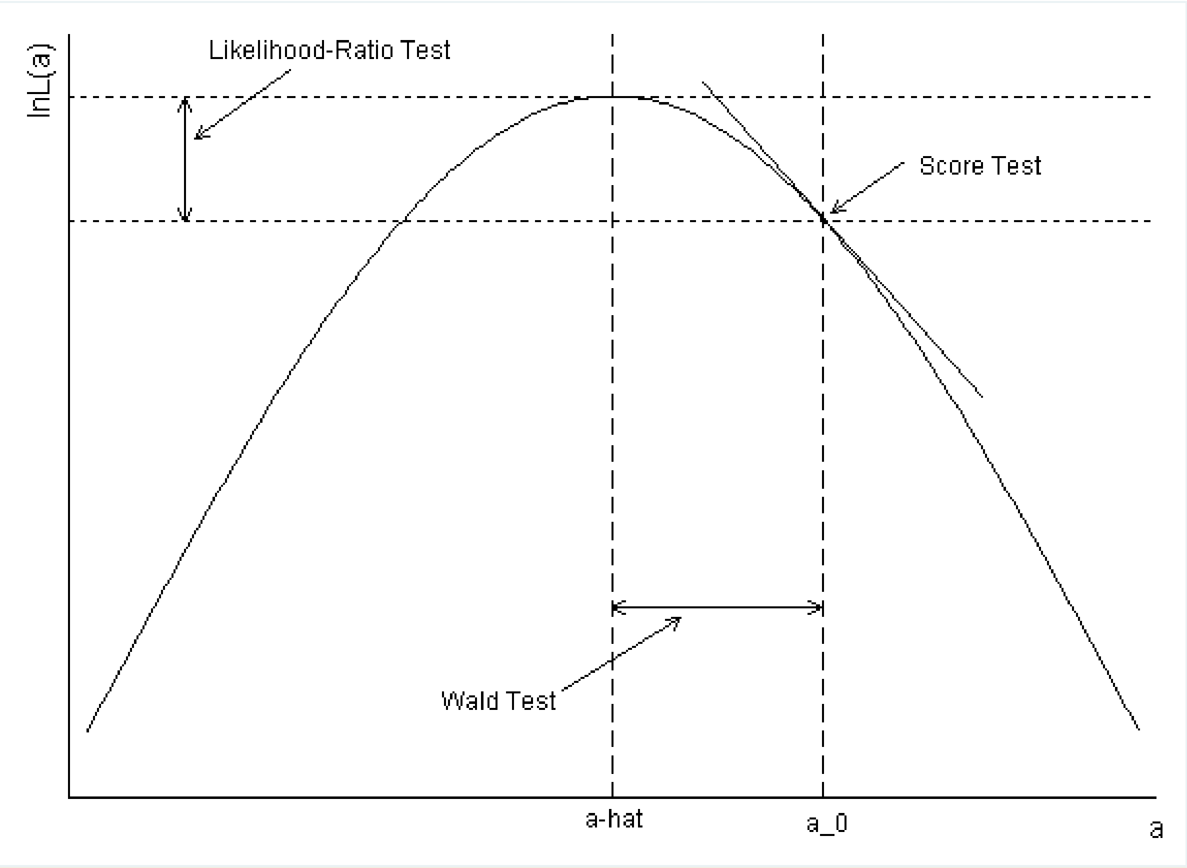
    - optimized BLAS, kernel trick, etc.
    - memory vs speed vs robustness ...
    - p vs. n vs. many-small-regressions vs. ...

Inference

- σ^2 (residual variance) is $\text{RSS}/(n-p)$
- The covariance matrix is $\Sigma = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$.
- Individual coefficients are t-distributed
- Linear combinations of coefficients (contrasts or predictions) are t-distributed with covariance matrix $\mathbf{C}^\top\Sigma^{-1}\mathbf{C}$
- Joint hypotheses on coefficients are F-distributed
- Wald and likelihood ratio test comparisons are equivalent
  (but need to be careful about marginality)

Wald vs LRT

- either approach can be used for tests of *single* parameters, *joint* tests of parameters, or confidence intervals
- Wald makes an additional assumption/approximation (quadratic log-likelihood surface)
    - this assumption is exact for linear models
- Wald may assume either known σ^2 (Z, χ^2 tests) or estimated $\hat{\sigma}^2$ (t, F tests)



From OARC stats, based on Fox 1977 *Applied regression analysis*

Model matrices

- model definition converted to **X** before we start
- **input variables** vs **predictor variables** (Schielzeth (2010), Gelman and Hill (2006), CV)
  - transformations
  - encoding of categorical variables: **contrasts**
  - interactions
  - basis expansions (e.g. polynomials)

Wilkinson-Rogers formulas

- Wilkinson and Rogers (1973), updated by Chambers and Hastie (1991, ch. 2)

- operators: `+`, `*`, `:`, `/`, `-`, `^`
- `I()`

Contrasts

treatment contrasts

- intercept = baseline, subsequent values are differences
- $\{\beta_0 = \mu_0, \beta_i = \mu_i - \mu_0 \text{ for } i > 0\}$
- equivalently: $\{\mu_0 = \beta_0, \mu_i = \beta_0 + \beta_i \text{ for } i > 0\}$
- **contrast matrix**:

$$\mathbf{C}\beta = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 1 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \dots \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \dots \end{pmatrix} \quad .$$

Maybe easier to start from the **inverse** contrast matrix: $\beta = \mathbf{C}^{-1}\mu$.

```
C <- cbind(1, contr.treatment(3))   ## R omits the intercept by default
solve(C)
```

```
   1  2  3
   1  0  0
2 -1  1  0
3 -1  0  1
```

We have to specify the baseline level (`contr.treatment` uses first level of a factor; `contr.SAS()` uses the last level).

It's nice when contrasts are *orthogonal*, i.e. all rows are independent → $\mathbf{C}^\top\mathbf{C}$ is diagonal.

Sum-to-zero contrasts

- intercept is the (unweighted!) average rather than baseline value ($\sum \mu_i/n$)
- other parameters are differences between mean of level i and intercept ($\mu_i - \sum_j \mu_j/n$)
- **last** level is dropped

```r
mfun <- function(C) MASS::fractions(solve(C))
(C <- cbind(1,contr.sum(3)))
```

```
  [,1] [,2] [,3]
1    1    1    0
2    1    0    1
3    1   -1   -1
```

```r
mfun(C)
```

```
     1    2    3   
[1,]  1/3  1/3  1/3
[2,]  2/3 -1/3 -1/3
[3,] -1/3  2/3 -1/3
```

Helmert contrasts

- Weird but orthogonal
- intercept, diff of first two levels, diff of level 3 from 1 & 2, …

```r
(C <- cbind(1,contr.helmert(3)))
```

```
  [,1] [,2] [,3]
1    1   -1   -1
2    1    1   -1
3    1    0    2
```

```r
mfun(C)
```

```
     1    2    3   
[1,]  1/3  1/3  1/3
[2,] -1/2  1/2    0
[3,] -1/6 -1/6  1/3
```

others

- `MASS::contr.sdif()` (successive-differences)
- `contr.poly()` (orthogonal polynomial contrasts)
- custom (e.g., “none” vs “symbiont effect” vs “crabs vs shrimp” vs “two-symbiont effect”) (McKeon et al. (2012); data here)

```
cc_inv <- matrix(c(1/4,1/4,1/4,1/4,
               1,-1/3,-1/3,-1/3,
               0,1,-1,0,
               0,1/2,1/2,-1),
             byrow=TRUE,
             nrow=4,
             dimnames=list(c("intercept","avg_symb","C.vs.S","twosymb"),
                           c("none","C","S","CS")))
## inverse contrast matrix
MASS::fractions(cc_inv)
```

```
          none C    S    CS  
intercept  1/4  1/4  1/4  1/4
avg_symb     1 -1/3 -1/3 -1/3
C.vs.S       0    1   -1    0
twosymb      0  1/2  1/2   -1
```

```
## contrast matrix
mfun(cc_inv)
```

```
     intercept avg_symb C.vs.S twosymb
none    1       3/4        0      0
C       1      -1/4      1/2    1/3
S       1      -1/4     -1/2    1/3
CS      1      -1/4        0   -2/3
```

practical issues

- too many ways to set contrasts (`options()`, `contrasts(f) <-`, `lm(..., contrasts = list(...))`)
- terrible naming conventions: you can get used to it or use the `faux` package
- OK to fit models and later use `emmeans` to recover desired contrasts (switching linear bases)

```
mtcars$fcyl <- factor(mtcars$cyl)
lm0 <- lm(mpg ~ fcyl, mtcars)
cn <- function(x) names(coef(x))
cn(lm0)
```

```
[1] "(Intercept)" "fcyl6"       "fcyl8"
```

```
update(lm0, contrasts = list(fcyl = contr.sum(3))) |> cn()
```

```
[1] "(Intercept)" "fcyl1"       "fcyl2"
```

```
update(lm0, contrasts = list(fcyl = contr.helmert(3))) |> cn()
```

```
[1] "(Intercept)" "fcyl1"       "fcyl2"
```

using `faux`

```
update(lm0, data = transform(mtcars, fcyl = contr_code_sum(fcyl))) |> cn()
```

```
[1] "(Intercept)"      "fcyl.4-intercept" "fcyl.6-intercept"
```

```
update(lm0, data = transform(mtcars, fcyl = contr_code_helmert(fcyl))) |> cn()
```

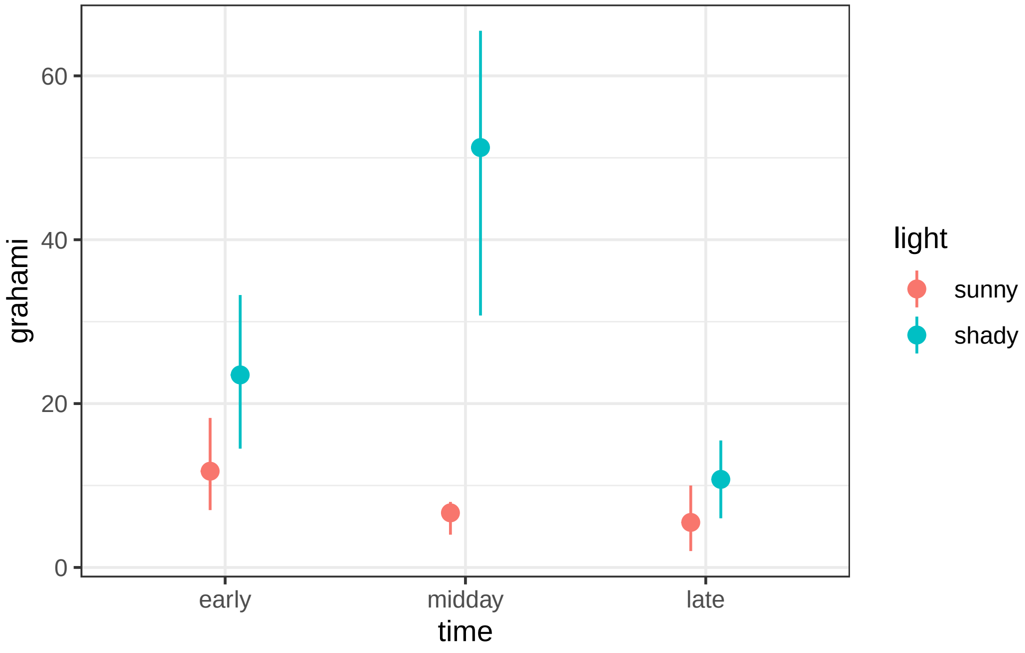
```
[1] "(Intercept)" "fcyl.6-4"    "fcyl.8-4.6"
```

Interactions

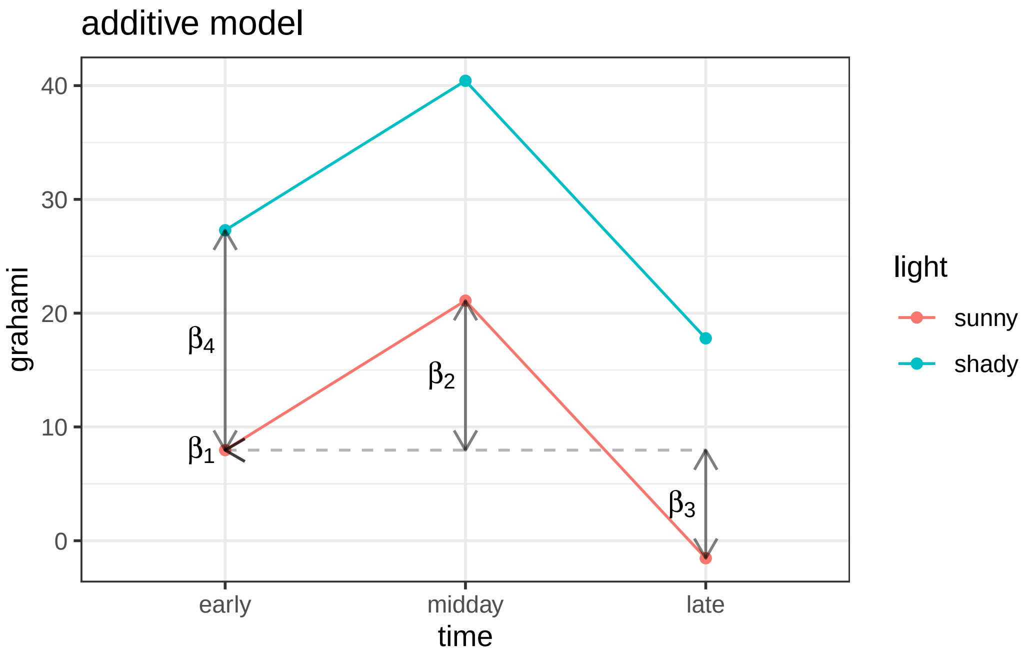
- differences in differences
- parameter values of main effects (and p values etc.) depend on contrasts/centering!
- overall model fit (R^2, predictions, etc.) is invariant

Lizard data (Schoener (1970), from the `brglm2` package):

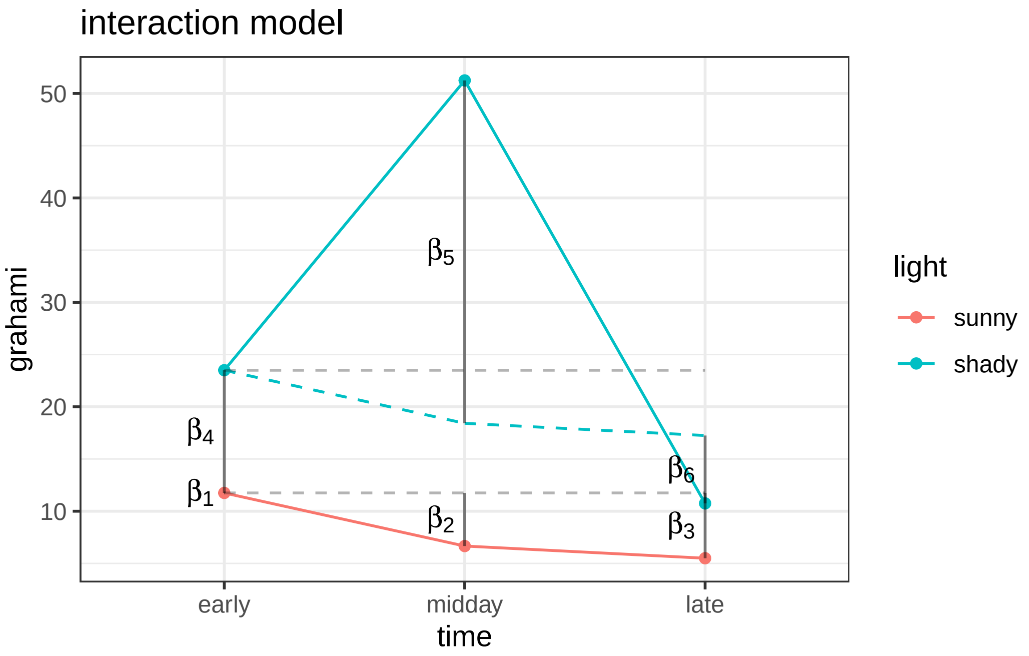
```
data("lizards", package = "brglm2")
ggplot(lizards, aes(time, grahami, colour = light)) +
    stat_summary(fun.data = mean_cl_boot,
                 position = position_dodge(width = 0.25))
```



```
data("lizards", package = "brglm2")
lmTL1 <- lm(grahami~time+light,data=lizards)
pp <- with(lizards,expand.grid(time=levels(time),light=levels(light)))
pp$grahami <- predict(lmTL1,newdata=pp)
cc <- as.list(plyr::rename(coef(lmTL1),c(`(Intercept)`="int")))
labelpos <- with(cc,
  list(x=c(1,2,3,1),xend=c(1,2,3,1),
      y=c(int,int,int,int),
      yend=c(int,int+timemidday,int+timelate,int+lightshady)))
xpos <- -0.1
ggplot(pp,aes(x=time,y=grahami,colour=light))+geom_point()+
  geom_line(aes(group=light))+
  annotate("segment",x=labelpos$x,xend=labelpos$xend,y=labelpos$y,
           yend=labelpos$yend,alpha=0.5,
           arrow=arrow(length = unit(0.3,"cm"),ends="both"))+
  annotate("text",x=with(labelpos,(x+xend)/2)+xpos,y=with(labelpos,(y+yend)/2),
label=paste0("beta[",1:4,"]"),parse=TRUE)+
  annotate("segment",x=labelpos$x[1],xend=labelpos$x[3],y=labelpos$y[1],
           yend=labelpos$y[1],alpha=0.3,lty=2) +
    labs(title = "additive model")
```



```
lmTL2 <- lm(grahami~time*light,data=lizards)
gg_color_hue <- function(n) {
  hues = seq(15, 375, length=n+1)
  hcl(h=hues, l=65, c=100)[1:n]
}
pp2 <- pp
pp2$grahami <- predict(lmTL2,newdata=pp)
cc <- as.list(plyr::rename(coef(lmTL2),c(`(Intercept)`="int",
         `timemidday:lightshady`="midshady",`timelate:lightshady`="lateshady")))
labelpos <- with(cc,
  list(x=c(1,2,3,1,2,3),xend=c(1,2,3,1,2,3),
      y=c(int,int,int,int,int+lightshady+timemidday,int+lightshady+timelate),
      yend=c(int,int+timemidday,int+timelate,int+lightshady,
             int+timemidday+lightshady+midshady,int+timelate+lightshady+lateshady)))
xpos <- -0.1
ggplot(pp2,aes(x=time,y=grahami,colour=light))+geom_point()+
  geom_line(aes(group=light))+
  annotate("segment",x=1:2,xend=2:3,
           y=with(cc,c(int+lightshady,int+timemidday+lightshady)),
           yend=with(cc,c(int+timemidday+lightshady,int+timelate+lightshady)),
           colour=gg_color_hue(2)[2],lty=2)+
  annotate("segment",x=labelpos$x,xend=labelpos$xend,y=labelpos$y,
           yend=labelpos$yend,alpha=0.5) +
           ## arrow=arrow(length = unit(0.3,"cm"),ends="both"))+
  annotate("text",x=with(labelpos,(x+xend)/2)+xpos,y=with(labelpos,(y+yend)/2),
label=paste0("beta[",1:6,"]"),parse=TRUE)+
  annotate("segment",x=rep(labelpos$x[1],2),
                     xend=rep(labelpos$x[3],2),
                     y=labelpos$yend[c(1,4)],
          yend=labelpos$yend[c(1,4)],alpha=0.3,lty=2) +
    labs(title = "interaction model")
```



Marginality

- Venables (1998)
- ‘type (X) sums of squares’
- scaling and centering (Schielzeth 2010) alleviates many problems; sum-to-zero contrasts (weighted or unweighted?)

Making life easier with `emmeans`

- important to understand contrasts
- **but** easier to compute *expected marginal means*, contrasts via `emmeans` package
- contrasts expressed in terms of *group means*
- e.g.
- good documentation, vignettes

Model interpretation, visualization, testing

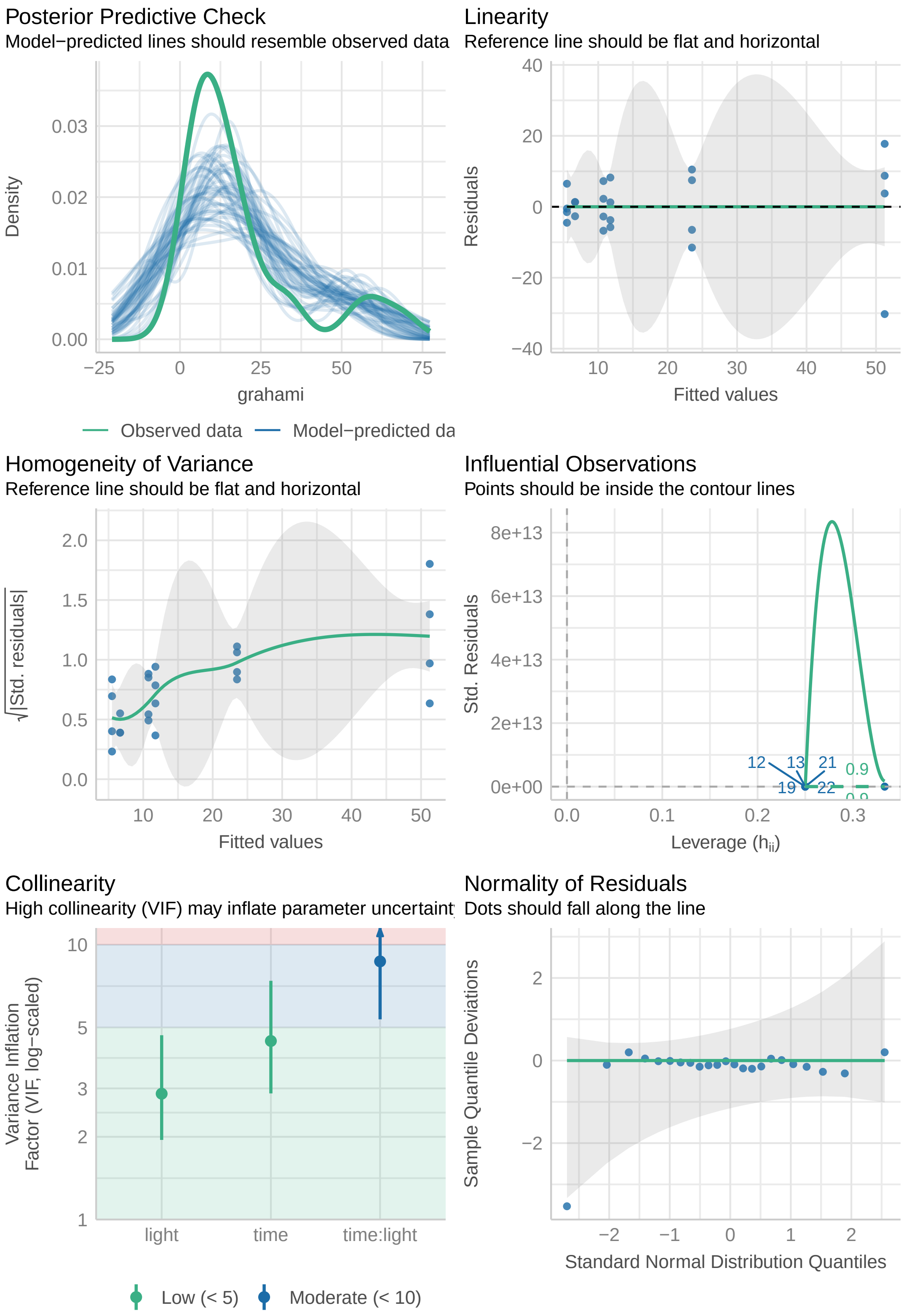
Diagnostics

- linearity > heteroscedasticity, outliers > normality
- upstream problems can induce downstream problems first
- universal plots are universal, but less interpretable than problem-specific exploration (try to identify problematic predictors/groups/etc.)

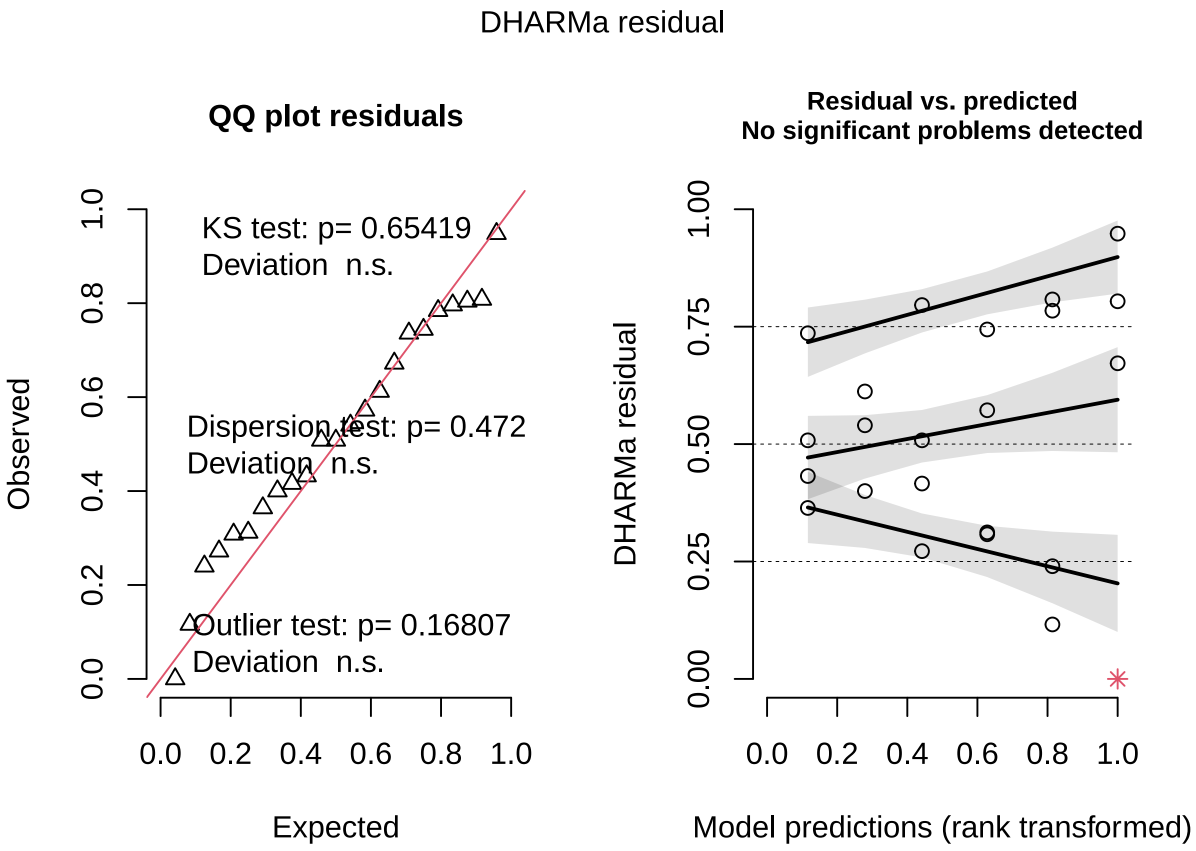
Graphical diagnostics

- base R: `stats::plot.lm()`
- `performance::check_model()`
- `DHARMa` (`simulateResiduals(., plot = TRUE)`)
  - (`plotResiduals(simout, form = pred_var)`)
- `broom::augment()` + plot-your-own (`ggplot2`)

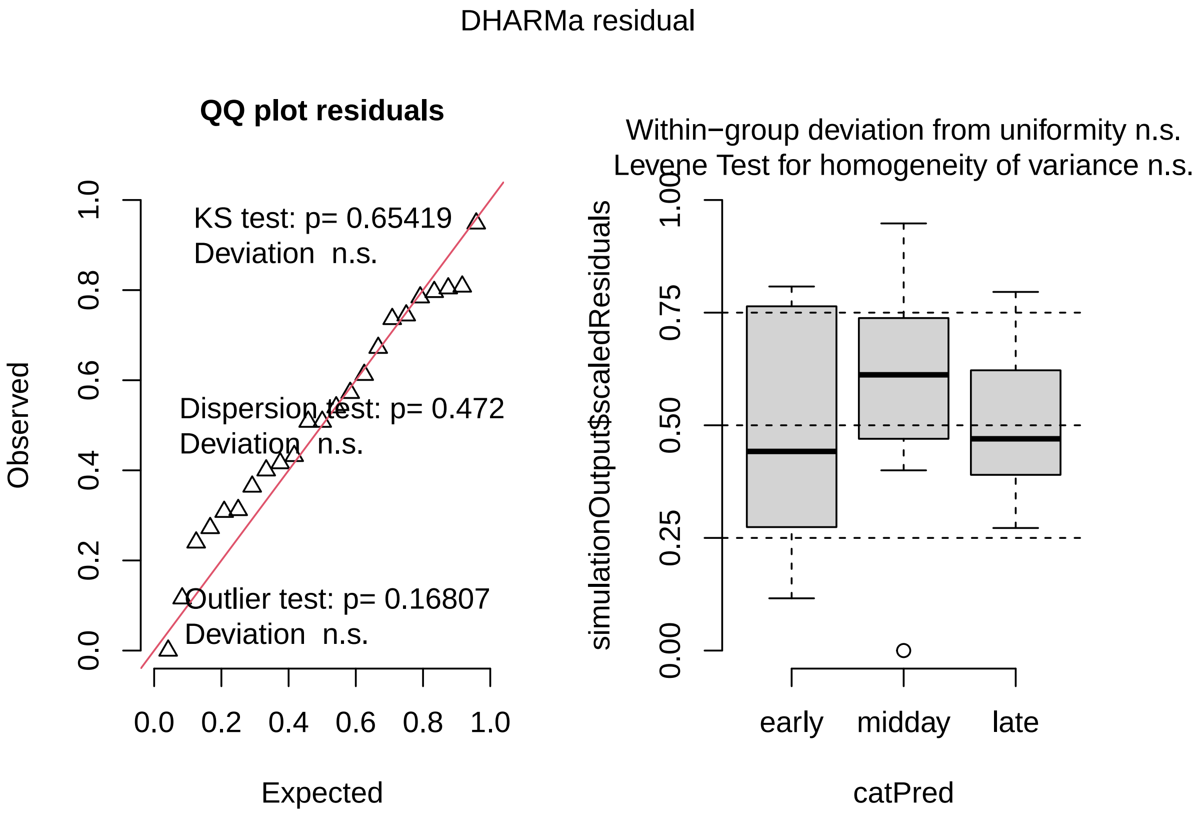
```
performance::check_model(lmTL2)
```


Posterior Predictive Check
Model−predicted lines should resemble observed data
Density
0.03
0.02
0.01
0.00
−25
0
25
50
75
grahami
Observed data
Model−predicted da
Linearity
Reference line should be flat and horizontal
Residuals
40
20
0
−20
−40
10
20
30
40
50
Fitted values
Homogeneity of Variance
Reference line should be flat and horizontal
√|Std. residuals|
2.0
1.5
1.0
0.5
0.0
10
20
30
40
50
Fitted values
Influential Observations
Points should be inside the contour lines
Std. Residuals
8e+13
6e+13
4e+13
2e+13
0e+00
12
13
21
19
22
0.9
0.9
0.0
0.1
0.2
0.3
Leverage (h_ii)
Collinearity
High collinearity (VIF) may inflate parameter uncertaint
Variance Inflation Factor (VIF, log−scaled)
10
5
3
2
1
light
time
time:light
Low (< 5)
Moderate (< 10)
Normality of Residuals
Dots should fall along the line
Sample Quantile Deviations
2
0
−2
−2
−1
0
1
2
Standard Normal Distribution Quantiles

```
ss <- simulateResiduals(lmTL2)
plot(ss)
```



```
plot(ss, form = lizards$time)
```



Solutions to problems

- **nonlinearity**: transformation, add covariates (??), add interactions, add polynomial terms etc.
- **outliers**: drop values (report both!), use robust regression
- **heteroscedasticity**: transformation, model dispersion explicitly, GLMs
- **non-Normality**: transformation, GLMs

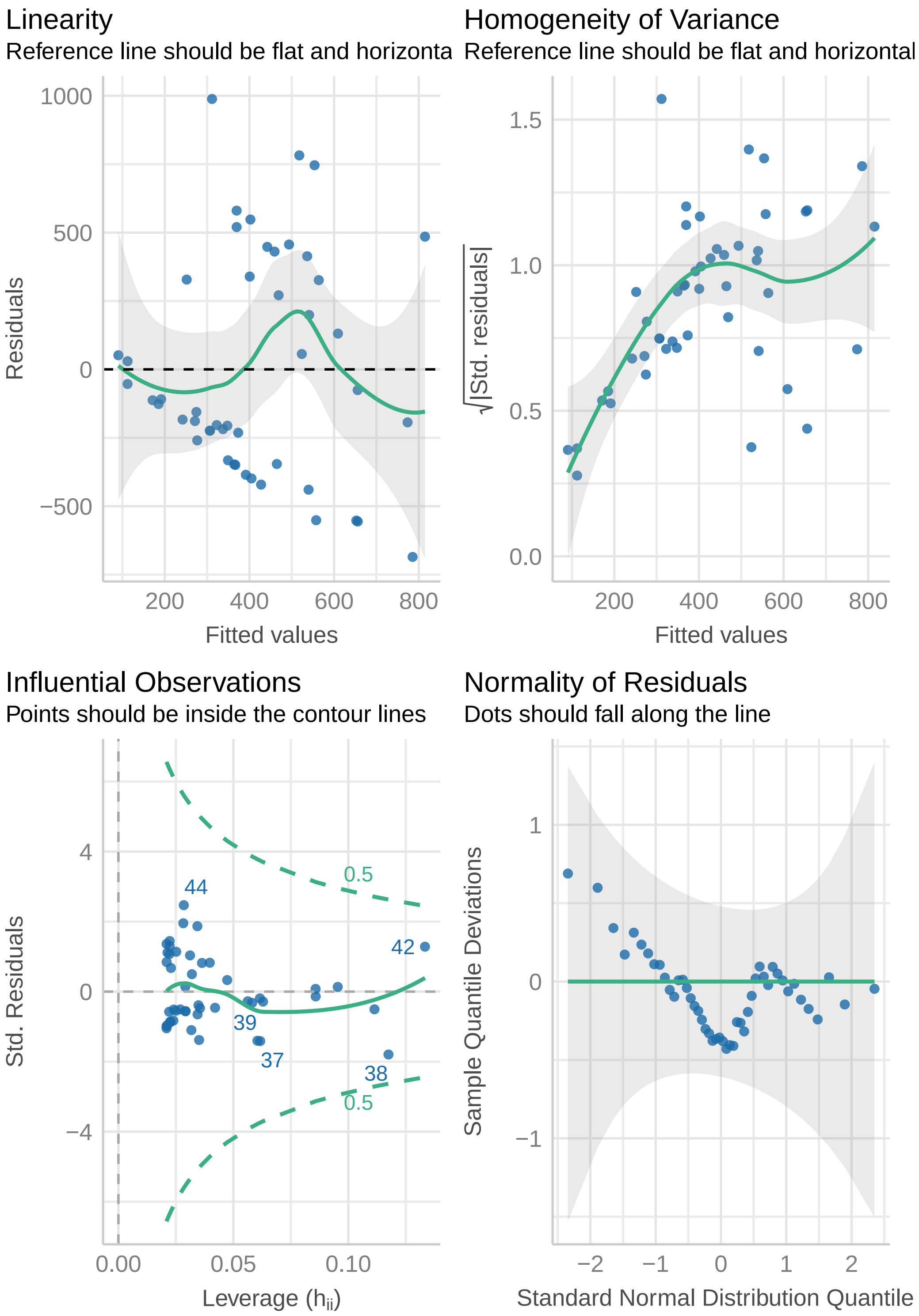
Transformation

- May do too much at once (GLMs and GAMs allow more flexibility)
- Log-transformation is often interpretable and solves problems
- Transforming boundary values (e.g. log(0)) is problematic
- **Box-Cox transformations**: $y \to \frac{y^{\lambda-1}}{\lambda}$ (include Jacobian term GM$^{\lambda-1}$ in denominator to keep log-likelihood comparable)
  - flexible
  - in practice people often use 'round numbers': $\lambda = 0$ (log), 1/2 (square root), etc.

  - `MASS::boxcox()`
  - hard to interpret!

Example

```r
library(faraway)
m1 <- lm(perm ~ area, rock)
performance::check_model(m1, check = c("linearity", "homogeneity", "outliers", "qq"))
```



Linearity

Reference line should be flat and horizonta

Homogeneity of Variance

Reference line should be flat and horizontal

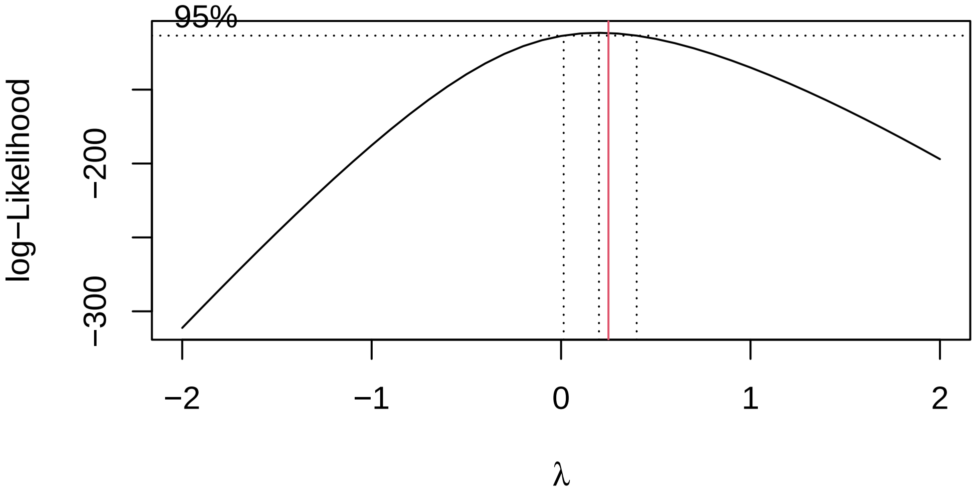
Influential Observations

Points should be inside the contour lines

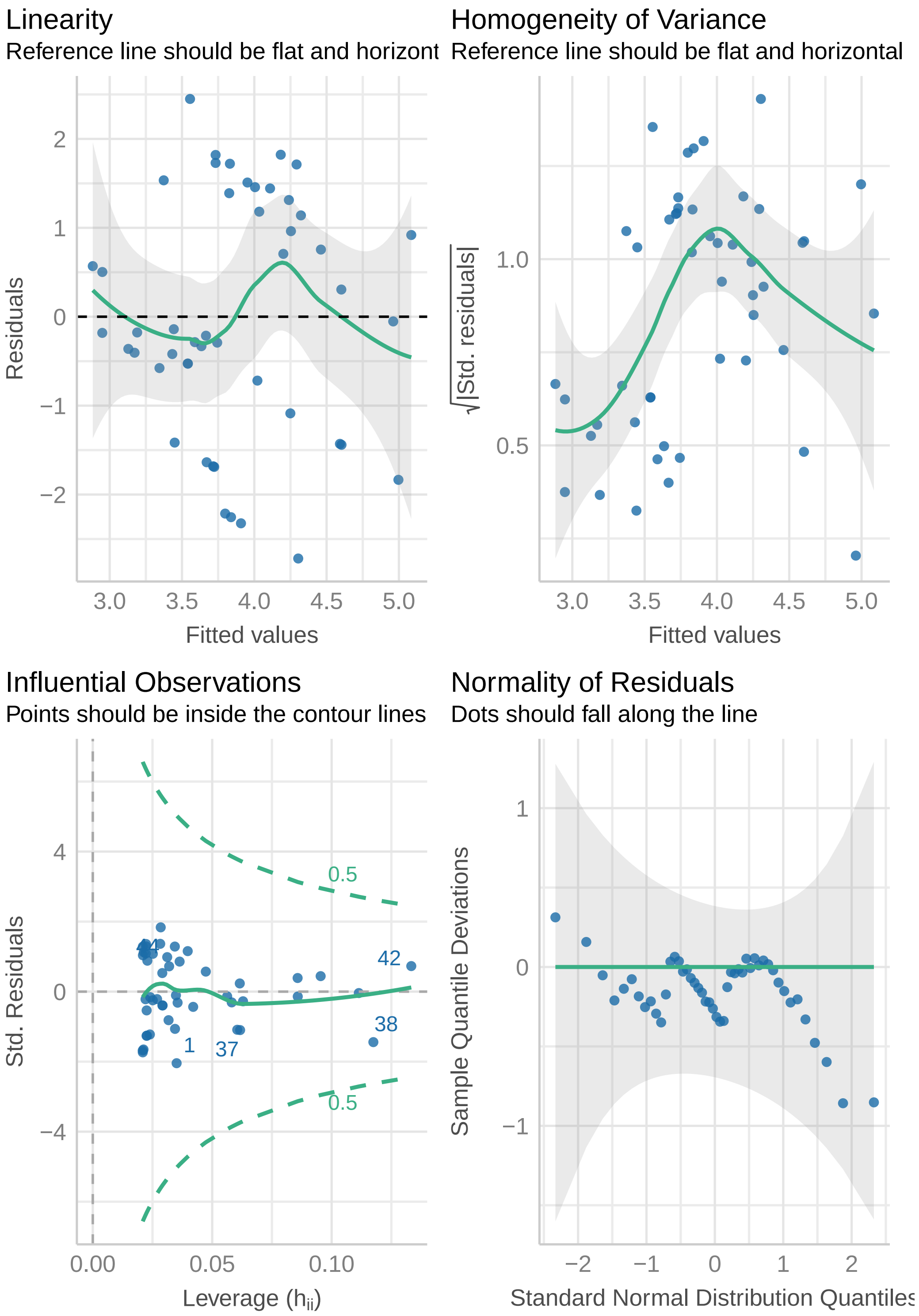
Normality of Residuals

Dots should fall along the line

```
m <- MASS::boxcox(m1, interp = FALSE)
abline(v=1/4, col = 2)
```



```
m2 <- update(m1, perm^(1/4) ~ .)
performance::check_model(m2, check = c("linearity", "homogeneity", "outliers", "qq"))
```



When we transform we have to add the logs of the *Jacobian* of the transform to the log-likelihood, or subtract $\sum \log(J_i)$ from the negative log-likelihood (Cranmer n.d.). For example, if we log-transform,

$$\begin{aligned}\sum \log(J(y_i)) &= \sum \log\left(\partial(\log(y_i))/\partial y\right) \\ &= \sum \log(1/y_i) \\ &= -\sum \log y_i.\end{aligned}$$

In this case $\sum \log(J(y_i)) = \sum \log\left((1/4)P^{-3/4}\right) = -N\log 4 - 3/4 \sum \log(P)$:

Check:

```
logJ <- -(nrow(rock)*log(4)  + 3/4*sum(log(rock$perm)))
lm2 <- logLik(m2) + logJ
lm1 <- logLik(m1)
lm2-lm1
```

```
'log Lik.' 23.3844 (df=3)
```

```
## difference from Box-Cox sequence
max(m$y)-m$y[m$x==1]
```

```
[1] 23.52213
```

Reminder about parameter scaling

- Schielzeth (2010), Gelman (2008)
- centering ensures no problems with marginality (analogous to sum-to-zero contrasts)
- scaling predictors gives coefficients equal units (analogous to Cohen's d)
- scaling response makes coefficients *standardized partial regression coefficients* (equal to partial correlation coefficients if predictors are independent)
- scaling by 2 SD makes coefficients equivalent to those from balanced binary predictors
- scaling inputs or scaling predictors?

What about correlated predictors?

- Can compute *variance inflation factors* (VIFs)
- Dropping correlated factors is dubious: Graham (2003), Dormann et al. (2012), Morrissey and Ruxton (2018), Vanhove (2021)

- perfect collinearity gets handled automatically by R's pivoting, but may want to change contrasts/model setup

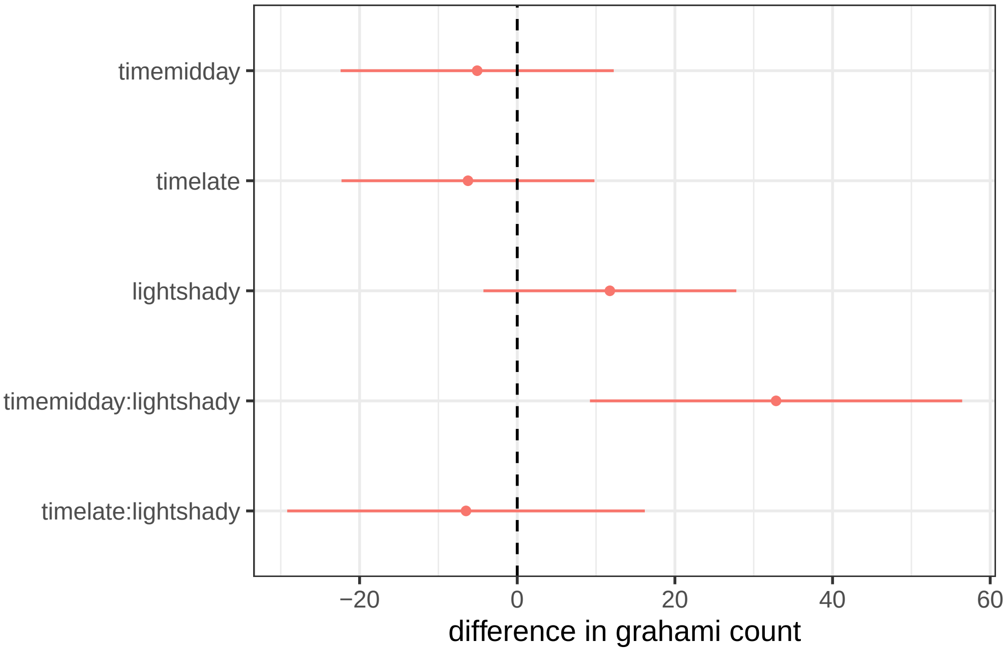
```
summary(lmTL2)
```

```

Call:
lm(formula = grahami ~ time * light, data = lizards)

Residuals:
    Min      1Q  Median      3Q     Max 
-30.250  -4.125   1.250   6.875  17.750 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)   
(Intercept)             11.750      5.375   2.186  0.04311 * 
timemidday              -5.083      8.211  -0.619  0.54408   
timelate                -6.250      7.602  -0.822  0.42238   
lightshady              11.750      7.602   1.546  0.14061   
timemidday:lightshady   32.833     11.190   2.934  0.00927 **
timelate:lightshady     -6.500     10.751  -0.605  0.55343   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.75 on 17 degrees of freedom
Multiple R-squared:  0.7504,    Adjusted R-squared:  0.677 
F-statistic: 10.22 on 5 and 17 DF,  p-value: 0.0001179
```

```
broom::tidy(lmTL2)
```

```
# A tibble: 6 x 5
  term                  estimate std.error statistic p.value
  <chr>                    <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)              11.8       5.38     2.19  0.0431 
2 timemidday               -5.08      8.21    -0.619 0.544  
3 timelate                 -6.25      7.60    -0.822 0.422  
4 lightshady               11.7       7.60     1.55  0.141  
5 timemidday:lightshady    32.8      11.2      2.93  0.00927
6 timelate:lightshady      -6.5      10.8     -0.605 0.553  
```

```
## automatically drop intercept; optional by_2sd argument
dotwhisker::dwplot(lmTL2) + geom_vline(xintercept = 0, lty = 2) +
    labs(x="difference in grahami count")
```



Interpretation and testing

- Look at coefficient tables: `summary()` or `coef(summary())`
- model comparison: `drop1()`, `anova()`, `car::Anova()`
- coefficient *plots*: `broom` + `ggplot2`, `dotwhisker`

Downstream methods

- plot predictions **with data**
- partial residuals plots (e.g. remef package)
- prediction, effects plots
- uncertainty of predictions
- `emmeans`, `marginaleffects`, `effects`, `sjPlot` ...

References

Chambers, J. M., and T. J. Hastie, eds. 1991. *Statistical Models in S*. 1st ed. Chapman & Hall/CRC.

Cranmer, Kyle. n.d. "Transformation Properties of the Likelihood and Posterior." Accessed September 12, 2024. https://cranmer.github.io/stats-ds-book/distributions/invariance-of-likelihood-to-reparameterizaton.html.

Dormann, Carsten F., Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel Carré, Jaime R. García Marquéz, et al. 2012. "Collinearity: A Review of Methods to Deal with It and a Simulation Study Evaluating Their Performance." *Ecography*, no–. https://doi.org/10.1111/j.1600-0587.2012.07348.x.

Gelman, Andrew. 2008. "Scaling Regression Inputs by Dividing by Two Standard Deviations." *Statistics in Medicine* 27 (15): 2865–73. https://doi.org/10.1002/sim.3107.

Gelman, Andrew, and Jennifer Hill. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge, England: Cambridge University Press.

Graham, Michael H. 2003. "Confronting Multicollinearity in Ecological Multiple Regression." *Ecology* 84 (11): 2809–15. https://doi.org/10.1890/02-3114.

Lindeløv, Jonas Kristoffer. 2019. "Common Statistical Tests Are Linear Models (or: How to Teach Stats)." https://lindeloev.github.io/tests-as-linear/.

McKeon, C., Adrian Stier, Shelby McIlroy, and Benjamin Bolker. 2012. "Multiple Defender Effects: Synergistic Coral Defense by Mutualist Crustaceans." *Oecologia* 169 (4): 1095–1103. https://doi.org/10.1007/s00442-012-2275-2.

Morrissey, Michael B., and Graeme D. Ruxton. 2018. "Multiple Regression Is Not Multiple Regressions: The Meaning of Multiple Regression and the Non-Problem of Collinearity." *Philosophy, Theory, and Practice in Biology* 10 (3).

Schielzeth, Holger. 2010. "Simple Means to Improve the Interpretability of Regression Coefficients: Interpretation of Regression Coefficients." *Methods in Ecology and Evolution* 1 (2): 103–13. https://doi.org/10.1111/j.2041-210X.2010.00012.x.

Schoener, Thomas W. 1970. "Nonsynchronous Spatial Overlap of Lizards in Patchy Habitats." *Ecology* 51 (3): 408–18. https://doi.org/10.2307/1935376.

Vanhove, Jan. 2021. "Collinearity Isn't a Disease That Needs Curing." *Meta-Psychology* 5 (April). https://doi.org/10.15626/MP.2021.2548.

Venables, W. N. 1998. "Exegeses on Linear Models." In. 1998 International S-PLUS User Conference. Washington, DC. http://www.stats.ox.ac.uk/pub/MASS3/Exegeses.pdf.

Wilkinson, G. N., and C. E. Rogers. 1973. "Symbolic Description of Factorial Models for Analysis of Variance." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 22 (3): 392–99. https://doi.org/10.2307/2346786.