Methods 3: Multilevel Statistical Modeling and Machine Learning

Week 04: *Evaluating Generalised linear mixed models*:
September 24, 2024

The course plan

Week 1: *Introduction*

Instructor sessions: *Setting up* R *and* Python *and recollection of the general linear model*

Week 2: *Multilevel linear regression*

Instructor sessions: *Modelling subject level effects – and how do they differ from group level effects?*

Week 3: *Link functions and fitting generalised linear multilevel models*

Instructor sessions: *What to do when the response variable is not continuous?*

Week 4: *Evaluating Generalised linear mixed models*

Instructor sessions: *How do we assess how models compare to one another?*

Week 5: *Explanation and Prediction*

Instructor sessions: *Code review*

Week 6: *Mid-way evaluation and Machine Learning Intro*

Instructor sessions: *Getting Python Running*

Week 7: *Linear regression revisited (machine learning)*

Instructor sessions: *How to constrain our models to make them more predictive*

Week 8: *Logistic regression revisited (machine learning)*

Instructor sessions: *Categorizing responses based on informed guesses*

Week 9: *Dimensionality Reduction, Principled Component Analysis (PCA)*

Instructor sessions: *What to do with very rich data?*

Week 10: *Outlook, unsupervised classification and neural networks*

Instructor sessions: *Data with no labels and networks*

Week 11: *Organising and preprocessing messy data*

Instructor sessions: *Code review*

Week 12: Final evaluation and wrap-up of course

Instructor sessions: *Ask anything!*

Recap

- Lydia's question – what about log 0
  - it's the linear predictor, $X\beta$, that is transformed
- 0^0:

```
> 0^0
[1] 1
```

  - https://en.wikipedia.org/wiki/Zero_to_the_power_of_zero
  - "*The choice whether to define 0^0 is based on convenience, not on correctness. If we refrain from defining 0^0, then certain assertions become unnecessarily awkward. ... The consensus is to use the definition $0^0 = 1$, although there are textbooks that refrain from defining 0^0*"

Benson DC (1999) The Moment of Proof: Mathematical Epiphanies. Oxford University Press

Recap – $0^0 = 1$

$$PMF_{Bernoulli} = p^k (1-p)^{1-k}$$

$$p=1;\ k=1;\ 1^1 * (1-1)^{1-1} = 1^1 * \mathbf{0^0} = 1*1 = 1$$

$$p=0;\ k=0;\ \mathbf{0^0} * (1-0)^{1-0} = \mathbf{0^0} * 1^1 = 1*1 = 1$$

$$p=1;\ k=0;\ 1^0 * (1-1)^{1-0} = 1^0 * 0^1 = 1*0 = 0$$

$$p=0;\ k=1;\ 0^1 * (1-0)^{1-1} = 0^1 * 1^0 = 0*1 = 0$$

Recap – generalised modelling

- At <u>least</u> four ingredients needed

1) A data vector: $y = (y_1, \ldots, y_n)$
2) Predictors: X and coefficients β, forming a linear predictor $X\beta$
3) A *link function* g: yielding a vector of transformed data $\hat{y} = g^{-1}(X\beta)$ that are used to model the data
4) A data distribution: $p(y \mid \hat{y})$

$$\left(X\beta = \beta_0 + X_1\beta_1 + \ldots + X_k\beta_k\right)$$

Gelman A, Hill J (2006) Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press: Chapter 6

Learning goals and outline

Evaluating Generalised linear mixed models

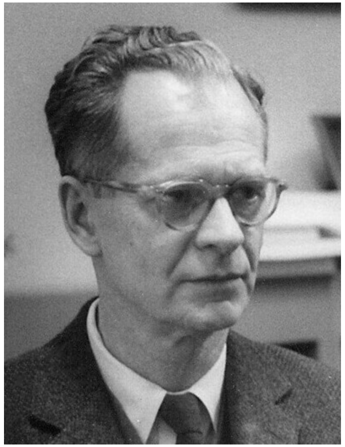
1) Learning tools for comparing models
    1) Variance explained
    2) Likelihood ratio tests
    3) Information criteria
2) Bridging to out-of-sample
    1) Introducing regularisation

Why are we modelling?

To be able to understand the world

EXPLANATION

To be able to predict and control the world



By Silly rabbit - self-made (by User:Silly rabbit). Updated in the Gimp by User:Michaelrayw2., CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=115478248

PREDICTION/CONTROL

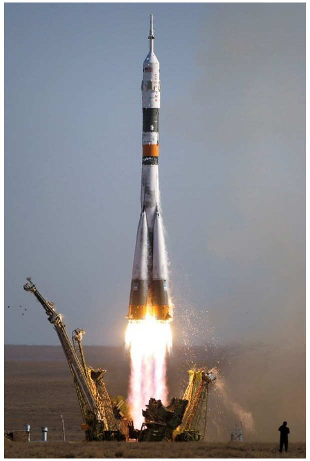
Why are we modelling?

To be able to understand the world

$$F = G\frac{m_1 m_2}{r^2}$$

EXPLANATION

To be able to predict and control the world



PREDICTION /CONTROL

What constitutes a good model?

Accurate estimation of the underlying parameters of the population distribution

Generalisation to new data

EXPLANATION

PREDICTION/CONTROL

Within an **explanatory** framework, how can we assess whether we have done a good job?

1) Variance explained

R summary

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

R^2 (coefficient of determination)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \mu_y\right)^2}$$

R^2; adjusted

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2}{\sum_{i=1}^{n} \left(y_i - \mu_y \right)^2}$$

$$\bar{R}^2 = 1 - \left(1 - R^2 \right) \frac{n-1}{n-p-1}$$

$n:$ number of observations
$p:$ number of predictors beyond the constant term

```
lm(formula = mpg ~ wt, data = mtcars)

Coefficients:
(Intercept)           wt
     37.285       -5.344
```

```
"R-squared: 0.753"
"R-squared, adjusted: 0.745"
```

```
lm(formula = mpg ~ I(wt^2) + wt, data = mtcars)

Coefficients:
(Intercept)      I(wt^2)           wt
     49.931        1.171      -13.380
```

```
"R-squared: 0.819"
"R-squared, adjusted: 0.807"
```

```
lm(formula = mpg ~ I(wt^3) + I(wt^2) + wt, data = mtcars)

Coefficients:
(Intercept)      I(wt^3)      I(wt^2)           wt
   48.40370      0.04594      0.68938    -11.82598
```

```
"R-squared: 0.819"
"R-squared, adjusted: 0.8"
```

```
lm(formula = mpg ~ I(wt^4) + I(wt^3) + I(wt^2) + wt, data = mtcars)

Coefficients:
(Intercept)      I(wt^4)      I(wt^3)      I(wt^2)           wt
    14.4558      -0.3863       5.2004     -23.7018      36.6195
```

```
"R-squared: 0.822"
"R-squared, adjusted: 0.796"
```

R^2 – multilevel models

$$R_{GLMM}(m)^2 = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_\alpha^2 + \sigma_\epsilon^2} \qquad R_{GLMM}(c)^2 = \frac{\sigma_f^2 + \sigma_\alpha^2}{\sigma_f^2 + \sigma_\alpha^2 + \sigma_\epsilon^2}$$

$\sigma_f^2:$ variance of the first-level effects
$\sigma_\alpha^2:$ variance of the second-level effects
$\sigma_\epsilon^2:$ unexplained variance
$m:$ marginal variance
$c:$ conditional variance

Nakagawa S, Schielzeth H (2013) A general and simple method for obtaining R2 from generalized linear mixed-effects models. Methods in Ecology and Evolution 4:133–142. https://doi.org/10.1111/j.2041-210x.2012.00261.x

R^2 – multilevel models

```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (1 | Subject)
   Data: sleepstudy
REML criterion at convergence: 1786.465
Random effects:
 Groups   Name        Std.Dev.
 Subject  (Intercept) 37.12
 Residual             30.99
Number of obs: 180, groups:  Subject, 18
Fixed Effects:
(Intercept)         Days
     251.41        10.47
```

$\hat{\sigma}_\alpha$

$\hat{\sigma}_\epsilon$

$$\hat{\sigma}_f = \mathrm{sd}(X\hat{\beta})$$

```r
library(lme4)
mm <- lmer(Reaction ~ Days + (1 | Subject), data=sleepstudy)
var.comp      <- as.data.frame(VarCorr(mm))

X <- model.matrix(mm)
beta.hat <- fixef(mm)

sigma.f       <- sd(X %*% beta.hat)
sigma.alpha   <- var.comp$sdcor[1]
sigma.epsilon <- var.comp$sdcor[2]

R.squared.m <- sigma.f^2 / (sigma.f^2 + sigma.alpha^2 + sigma.epsilon^2)
R.squared.c <- (sigma.f^2 + sigma.alpha^2) /
    (sigma.f^2 + sigma.alpha^2 + sigma.epsilon^2)
```

```r
print(R.squared.m)
```

```
## [1] 0.2798856
```

```r
print(R.squared.c)
```

```
## [1] 0.7042555
```

With slopes...

```
mm <- lmer(Reaction ~ Days + (Days | Subject), data=sleepstudy)

unscaled.SIGMA <- getME(mm, 'Lambda') %*% getME(mm, 'Lambdat')
SIGMA <- sigma(mm)^2 * unscaled.SIGMA
Z <- getME(mm, 'Z')
n <- length(sleepstudy$Reaction)
X <- model.matrix(mm)
beta.hat <- fixef(mm)

var.comp <- as.data.frame(VarCorr(mm))

obs.var <- Z %*% SIGMA %*% t(Z)

sigma.f       <- sd(X %*% beta.hat)
sigma.alpha <- sqrt(sum(diag(obs.var)) / n)
sigma.epsilon <- var.comp$sdcor[4]

R.squared.m <- sigma.f^2 / (sigma.f^2 + sigma.alpha^2 + sigma.epsilon^2)
R.squared.c <- (sigma.f^2 + sigma.alpha^2) /
    (sigma.f^2 + sigma.alpha^2 + sigma.epsilon^2)

print(R.squared.m)
```

Johnson PCD (2014) Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models. Methods in Ecology and Evolution 5:944–946.
https://doi.org/10.1111/2041-210X.12225

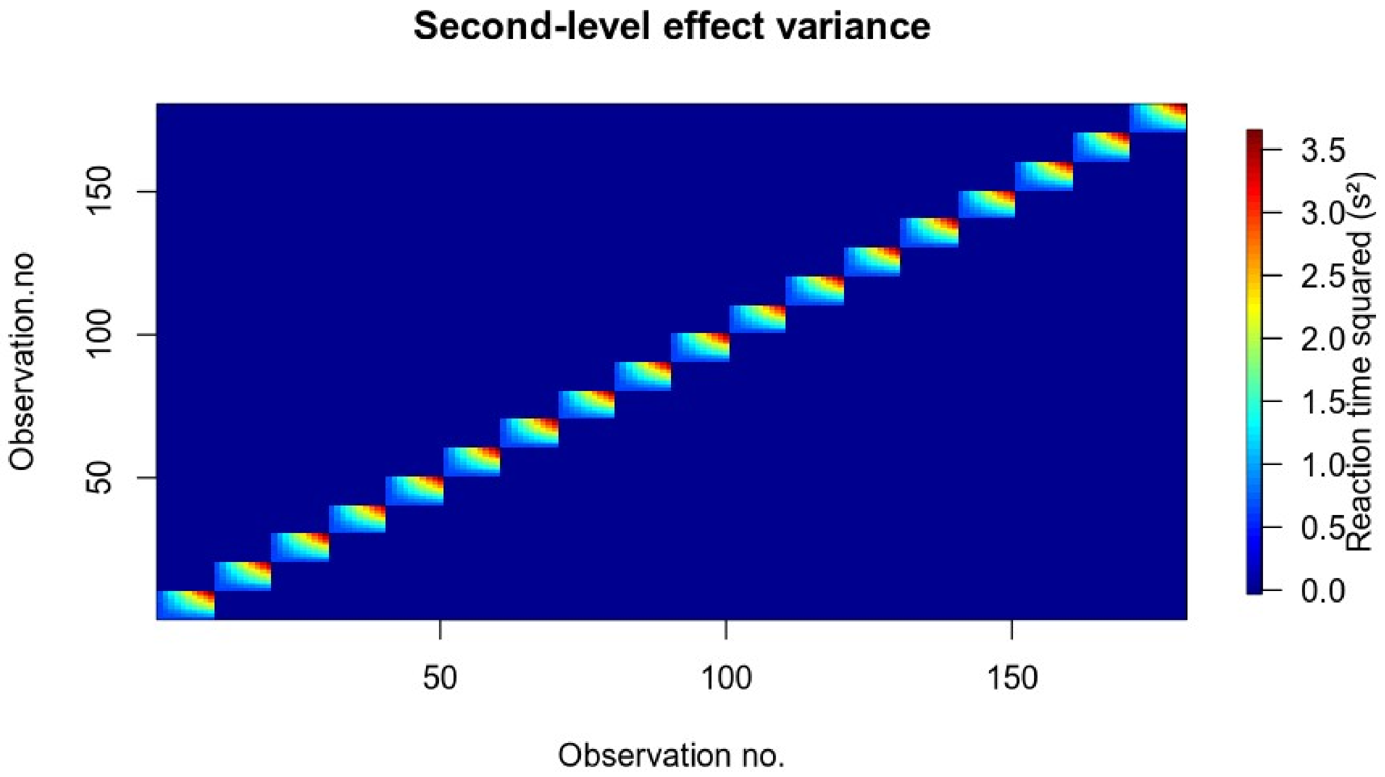
```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy
REML criterion at convergence: 1743.628
Random effects:
 Groups   Name        Std.Dev. Corr
 Subject  (Intercept) 24.741
          Days         5.922   0.07
 Residual             25.592
Number of obs: 180, groups:  Subject, 18
Fixed Effects:
(Intercept)         Days
     251.41        10.47
```

$$B \sim N(0, \boldsymbol{\Sigma})$$

$$\hat{\boldsymbol{\Sigma}} = \begin{matrix} 24.741^2 & 9.61 \\ 9.61 & 5.922^2 \end{matrix}$$

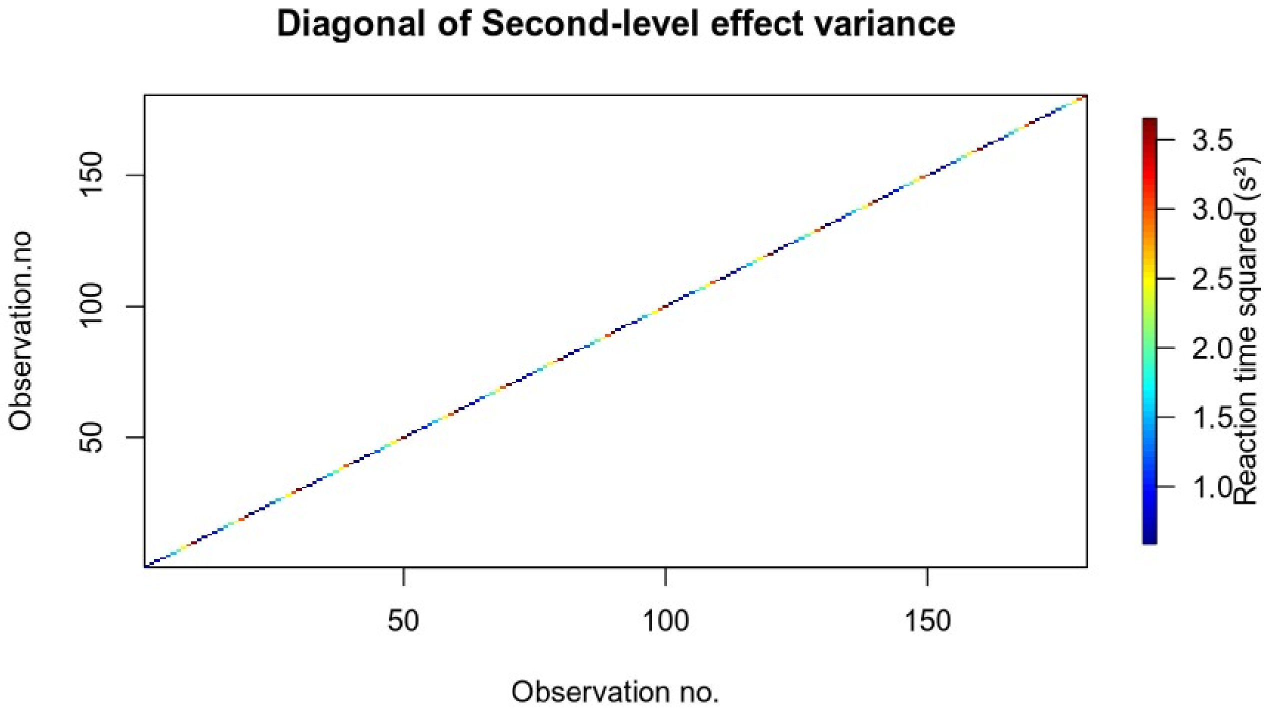
$$\text{obs.var} = \boldsymbol{Z}\boldsymbol{\Sigma}\boldsymbol{Z}^T$$

$$\sigma_{\alpha}{}^{2} = tr(\text{obs.var})/n$$

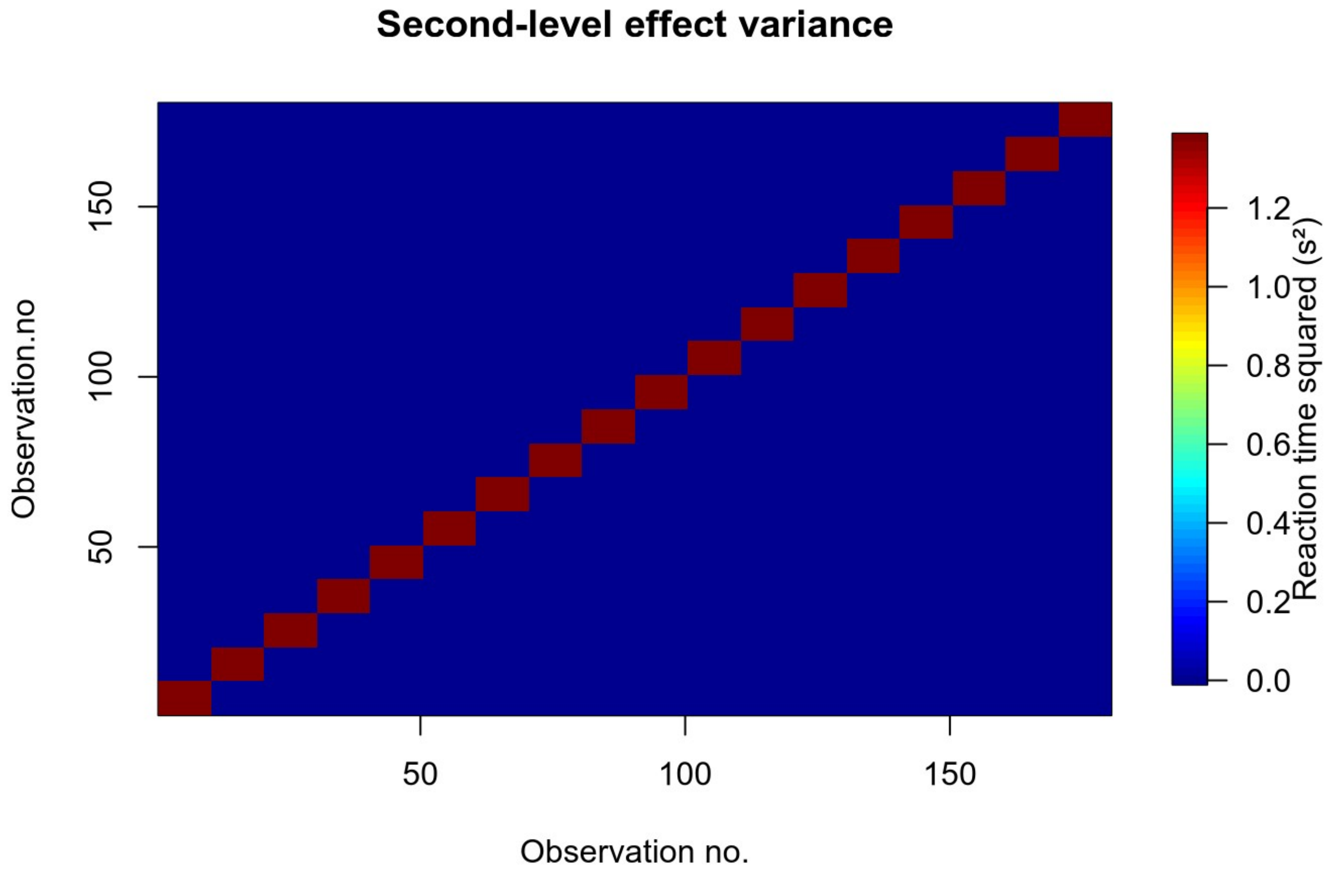


Johnson PCD (2014) Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models. Methods in Ecology and Evolution 5:944–946.
https://doi.org/10.1111/2041-210X.12225

$$\sigma_{\alpha}{}^{2} = tr(\text{obs.var})/n$$



Johnson PCD (2014) Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models. Methods in Ecology and Evolution 5:944–946.
https://doi.org/10.1111/2041-210X.12225



Discuss: what *Z* is behind this matrix?

MuMIn

MULTI-MODEL INFERENCE

```r
library(MuMIn)
print(r.squaredGLMM(mm))
```

```
##                R2m       R2c
## [1,] 0.2798856 0.7042555
```

```
conda install r-MuMIn # in your conda environment (terminal)
```

Variance explained

- Pros
  - R^2 is intuitive
- Cons
  - More complex models will always explain more variance
  - Hard to interpret in the case of collinearity
  - R^2 may not give us what we want

Yarkoni T, Westfall J (2017) Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. Perspect Psychol Sci 12:1100–1122. https://doi.org/10.1177/1745691617693393

$$y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i. \qquad (1)$$

$$\downarrow$$

$$\hat{y}_i = 1.6 + 0.35 X_{1i} + 0.62 X_{2i}.$$

$$\downarrow$$

$$R^2 = 0.750$$

$$\downarrow$$

Tempting interpretation: the parameter estimates are likely to be true, because R^2 is large

Correct interpretation:

“if one fits a model with the form of equation 1 in each new sample – *each time estimating new values of* b_0, b_1, and b_2 – what will be the average proportional reduction in the sum of squared errors?” (Yarkoni and Westfall 2017)

2) Likelihood ratios

Log-likelihoods

$$l(\hat{\beta}, \hat{\sigma}^2|y) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\hat{\sigma}^2) - \frac{n}{2}$$

$$l(\hat{\beta}|y) = \sum_{i=1}^{n} y_i \log(\hat{y}_i) + \log(1 - y_i)(1 - \hat{y}_i)$$

$$l(\hat{\beta}|y) = \sum_{i=1}^{n} y_i \log(\hat{y}_i) - \hat{y}_i - \log(y_i!)$$

```r
model.ranint   <- lmer(Reaction ~ Days + (1 | Subject), data=sleepstudy)
model.ranslope <- lmer(Reaction ~ Days + (Days | Subject), data=sleepstudy)
```

```r
print(ll.i <- logLik(model.ranint))
```

```
## 'log Lik.' -893.2325 (df=4)
```

```r
print(ll.s <- logLik(model.ranslope))
```

```
## 'log Lik.' -871.8141 (df=6)
```

Is the gain in log-likelihood worth the parameters we spend, and why is df increased by 2?

Likelihood-ratio test

$$LR = -2\left(l(\theta_2) - l(\theta_1)\right)$$

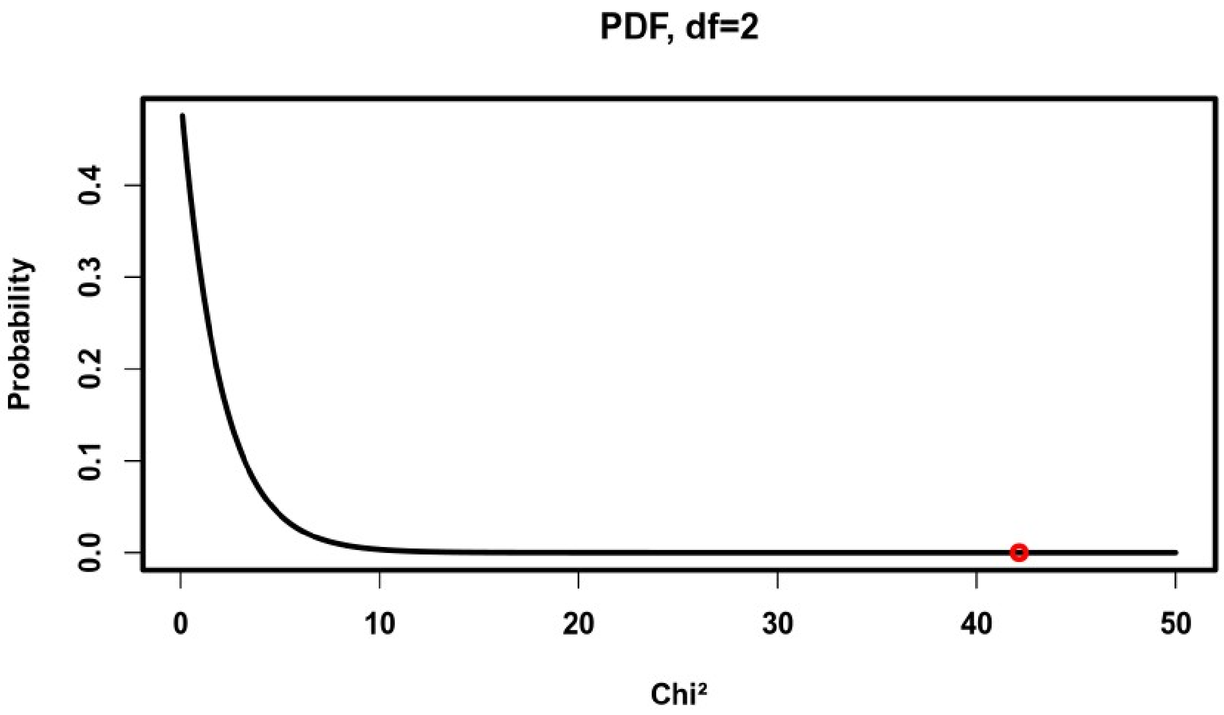
```
print(anova(model.ranint, model.ranslope))
```

```
refitting model(s) with ML (instead of REML)
Data: sleepstudy
Models:
model.ranint: Reaction ~ Days + (1 | Subject)
model.ranslope: Reaction ~ Days + (Days | Subject)
               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
model.ranint      4 1802.1 1814.8 -897.04   1794.1
model.ranslope    6 1763.9 1783.1 -875.97   1751.9 42.139  2  7.072e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

$$\chi^2(2)$$

Pr(>Chisq)

7.072e-10



When is it valid?

"The LR test is only adequate for testing fixed effects when both the ratio of the total sample size to the number of fixed-effect levels being tested and the number of random-effect levels (blocks) are large. *We have found little guidance and no concrete rules of thumb in the literature* [...]" (my highlights, Bolker et al., 2009)

Bolker BM, Brooks ME, Clark CJ, et al (2009) Generalized linear mixed models: a practical guide for ecology and evolution. Trends in Ecology & Evolution 24:127–135. https://doi.org/10.1016/j.tree.2008.10.008

Contents lists available at SciVerse ScienceDirect

Journal of Memory and Language

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

Random effects structure for confirmatory hypothesis testing: Keep it maximal

Dale J. Barr[a,*], Roger Levy[b], Christoph Scheepers[a], Harry J. Tily[c]

[a] *Institute of Neuroscience and Psychology, University of Glasgow, 58 Hillhead St., Glasgow G12 8QB, United Kingdom*
[b] *Department of Linguistics, University of California at San Diego, La Jolla, CA 92093-0108, USA*
[c] *Department of Brain and Cognitive Sciences, Massachussetts Institute of Technology, Cambridge, MA 02139, USA*

“Through theoretical arguments and Monte Carlo simulation, we show that LMEMs generalize best when they include the **maximal random effects structure justified by the design**. [...] Maximal LMEMs should be the ‘gold standard’ for confirmatory hypothesis testing in psycholinguistics and beyond. (my highlight)”

Barr DJ, Levy R, Scheepers C, Tily HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal. Journal of Memory and Language 68:255–278. https://doi.org/10.1016/j.jml.2012.11.001

Discuss: what does this mean?

Likelihood ratio

- Pros
  - Models can be compared in a principled way by reference to a theoretical distribution, χ^2. (In the single level case, *F* can be calculated)
- Cons
  - Models have to be nested in one another
  - Maximum likelihood fitting may be biased for complex models
  - Requires large sample sizes
  - Be careful if collinearity is high

N_{items}	Type I		Power	
	12	24	12	24
Type I: Error at or near α = .05				
min-F'	.027	.031	.327	.512
LMEM, maximal, χ^2_{LR}	.059	.056	.460	.610
LMEM, no random correlations, χ^2_{LR} [a]	.059	.056	.461	.610
LMEM, no within-unit intercepts, χ^2_{LR} [a]	.056	.055	.437	.596
LMEM, maximal, t	.072	.063	.496	.629
LMEM, no random correlations, t	.072	.062	.497	.629
LMEM, no within-unit intercepts, t [a]	.070	.064	.477	.620
$F_1 \times F_2$	.057	.072	.440	.643
Type I: Error far exceeding α = .05				
F_1	.176	.139	.640	.724
LMEM, no random correlations, MCMC[a]	.187	.198	.682	.812
LMEM, random intercepts only, MCMC	.415	.483	.844	.933
LMEM, random intercepts only, χ^2_{LR}	.440	.498	.853	.935
LMEM, random intercepts only, t	.441	.499	.854	.935

Barr DJ, Levy R, Scheepers C, Tily HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal. Journal of Memory and Language 68:255–278. https://doi.org/10.1016/j.jml.2012.11.001

3) Information criteria

Information criteria

$$deviance = -2l(\hat{\theta})$$
$$AIC = deviance + \boxed{2}k$$

$$k: \text{number of predictors}$$

When we add *k* predictors that are pure noise, deviance is reduced by an amount corresponding to a χ^2 distribution with *k* degrees of freedom.

(Gelman and Hill, 2006)

"On average, a predictor needs to reduce the deviance by 2 to in order to improve the fit to new data"

(Gelman and Hill, 2006, p. 525)

```
## Data: sleepstudy
## Models:
## model.ranint: Reaction ~ days_deprived + (1 | Subject)
## model.ranslope.and.int: Reaction ~ days_deprived + (days_deprived | Subject)
##                        Df    AIC    BIC  logLik deviance  Chisq Chi Df
## model.ranint            4 1446.5 1458.4 -719.25   1438.5
## model.ranslope.and.int  6 1425.2 1443.0 -706.58   1413.2 25.332      2
##                        Pr(>Chisq)
## model.ranint
## model.ranslope.and.int  3.156e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How to estimate *k*?

With *J* parameters

- Complete pooling
  - all *J* parameters collapsed into 1 parameter, (k = 1)
- No pooling
  - each of the *J* parameters is estimated, ($k = J$)
- Partial pooling
  - the number of effective parameters may be estimated by sampling and the *Deviance Information Criterion* can be estimated (beyond this course)
  - the number of parameters, loosely speaking, depend on whether the parameters are estimated mostly by the group average or the individual's own average (k = ?)

Information criteria

- Pros
  - Models can be compared even though one is not nested within the other (response data has to be the same though)
- Cons
  - Number of effective parameters not well defined for multilevel models
  - Maximum likelihood fitting may be biased for complex models
  - Be careful if collinearity is high

Interim summary

- Our different ways of comparing models seem to rely on estimating the number of effective parameters, on which there is no clear consensus
- Collinearity is a problem for all methods we have covered
  - Collinearity leads to unstable models
    - What can we do about it?

Collinearity

```
##
## Call:
## lm(formula = hp ~ mpg + wt + drat + qsec, data = mtcars)
##
## Coefficients:
## (Intercept)          mpg           wt         drat         qsec
##     473.779       -2.877       26.037        4.819      -20.751
```

Assuming no collinearity, what is the interpretation of the coefficients?

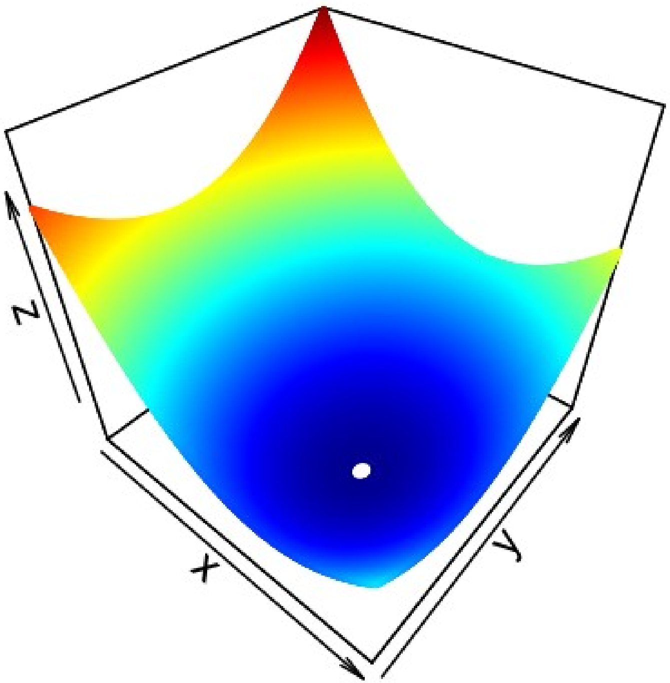
With collinearity, is that interpretation possible?

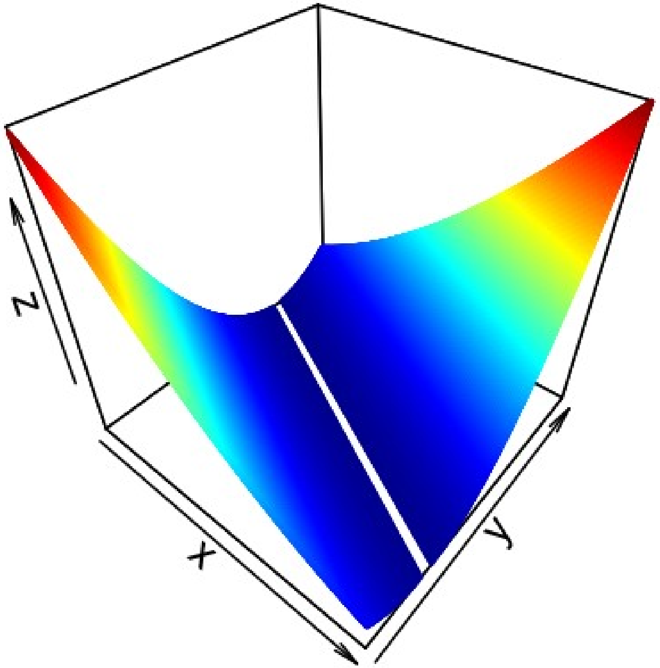
Stability of a model:

Feeding new data or adding new predictor variables will not change the parameter estimates a lot

Seems to be a good ingredient for a model that generalises well

Stability of a model



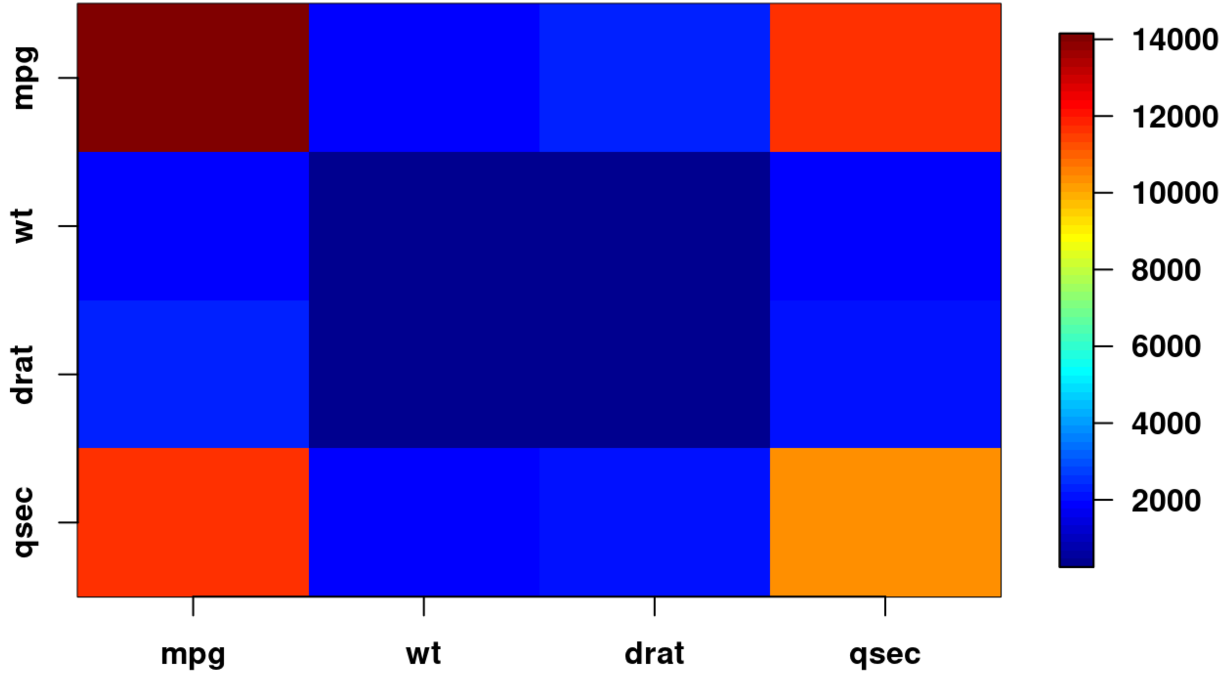


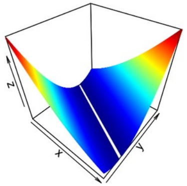
Here you see the gradient descent profiles for two models.
Which of these models is the most stable?

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

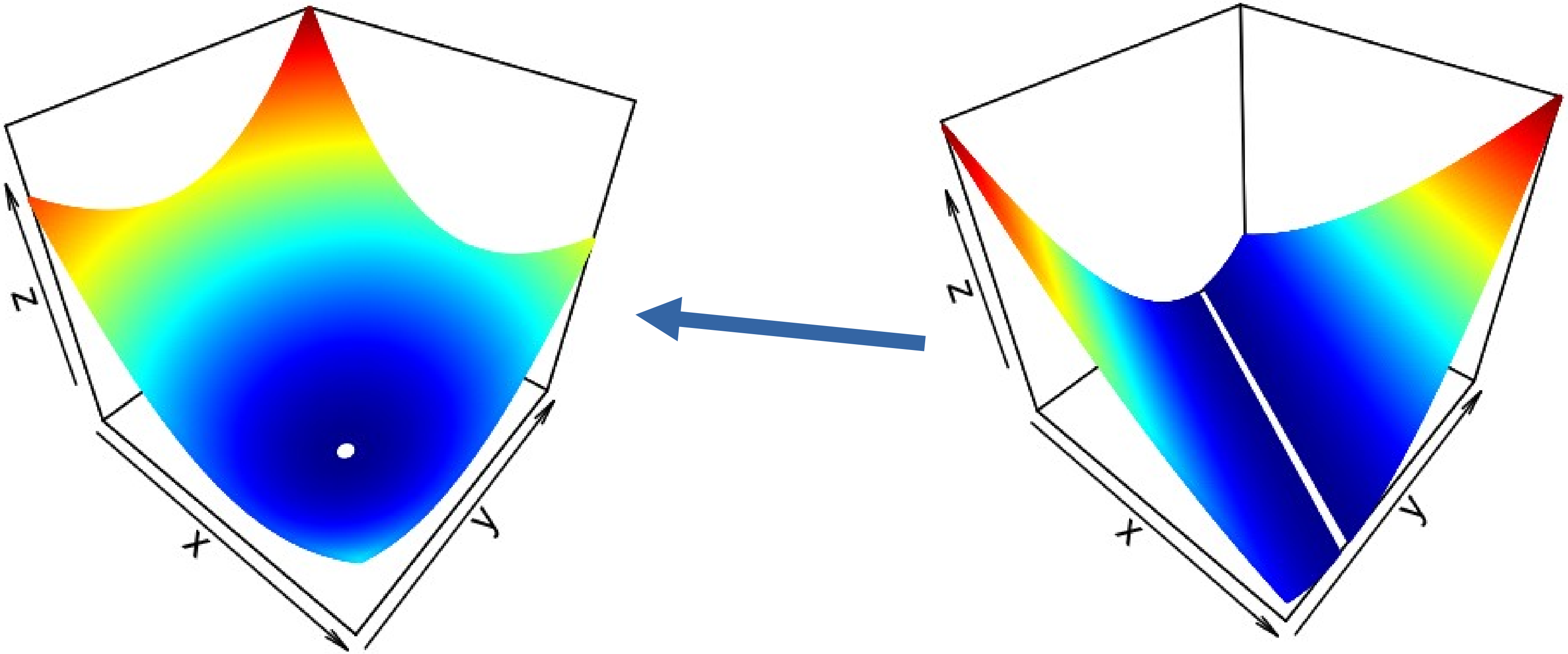
$$X^T X$$

The fact that the off-diagonal > 0, indicates that there is **collinearity**





How can we make a model stable?



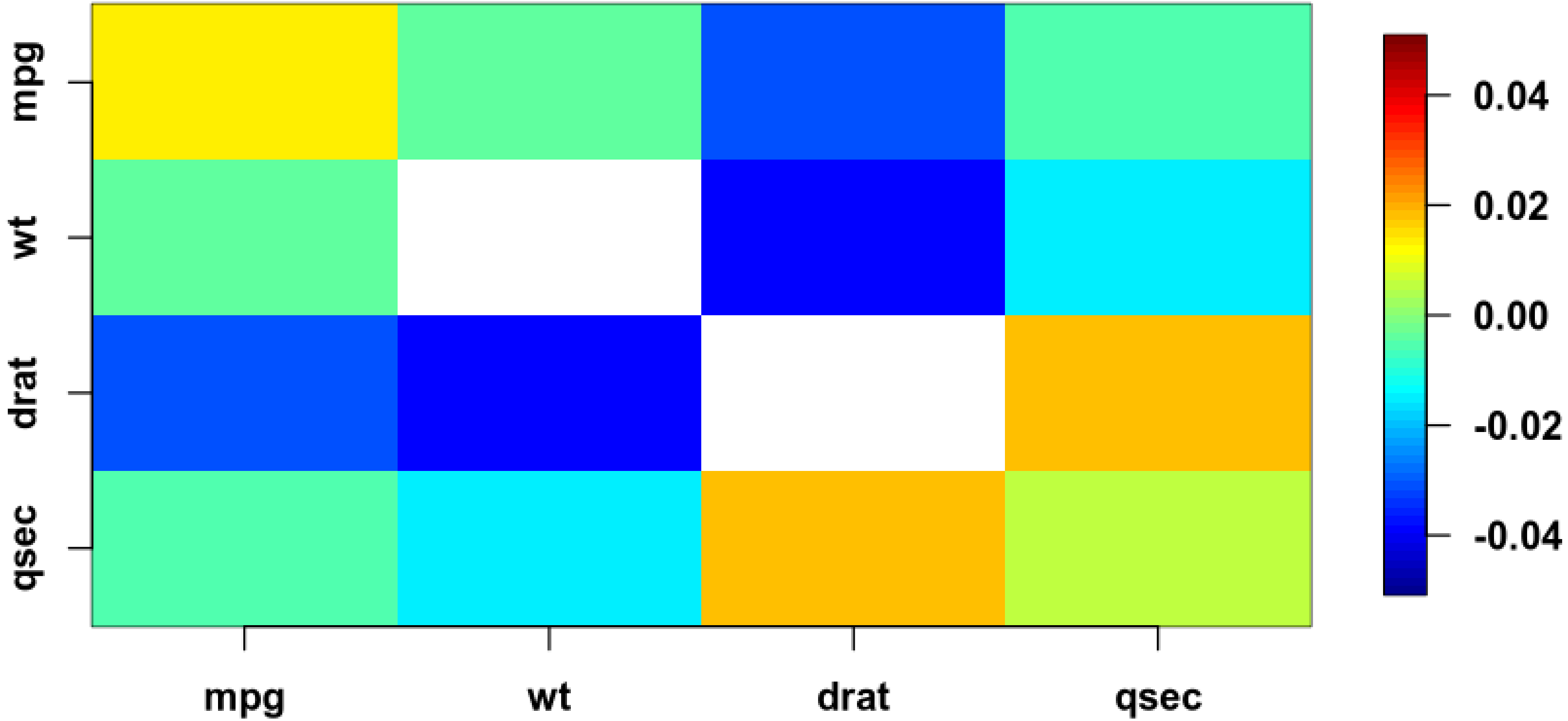
Regularisation

$$\hat{\beta}_{reg} = (X^T X + \lambda I)^{-1} X^T y$$

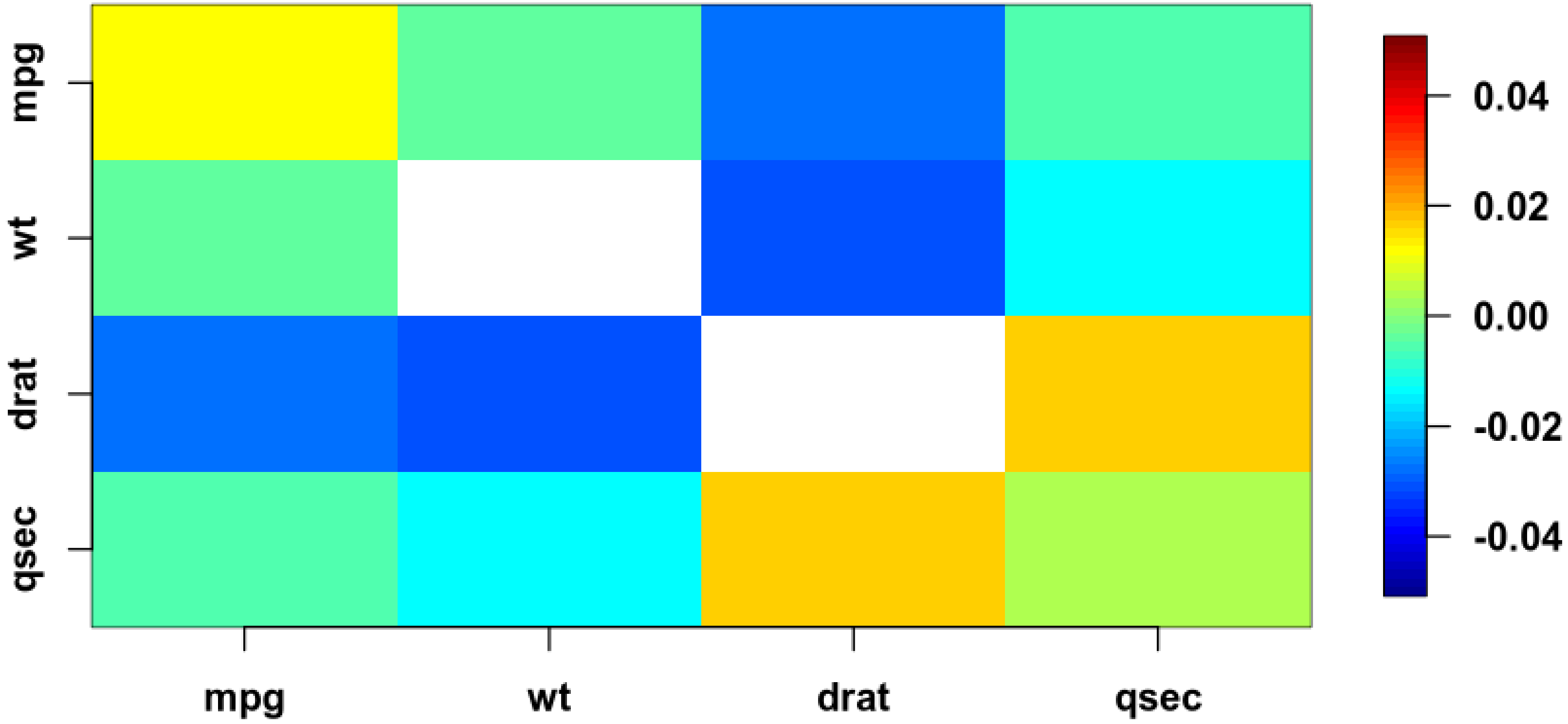
I : an identity matrix with p predictor variables

λ : a constant

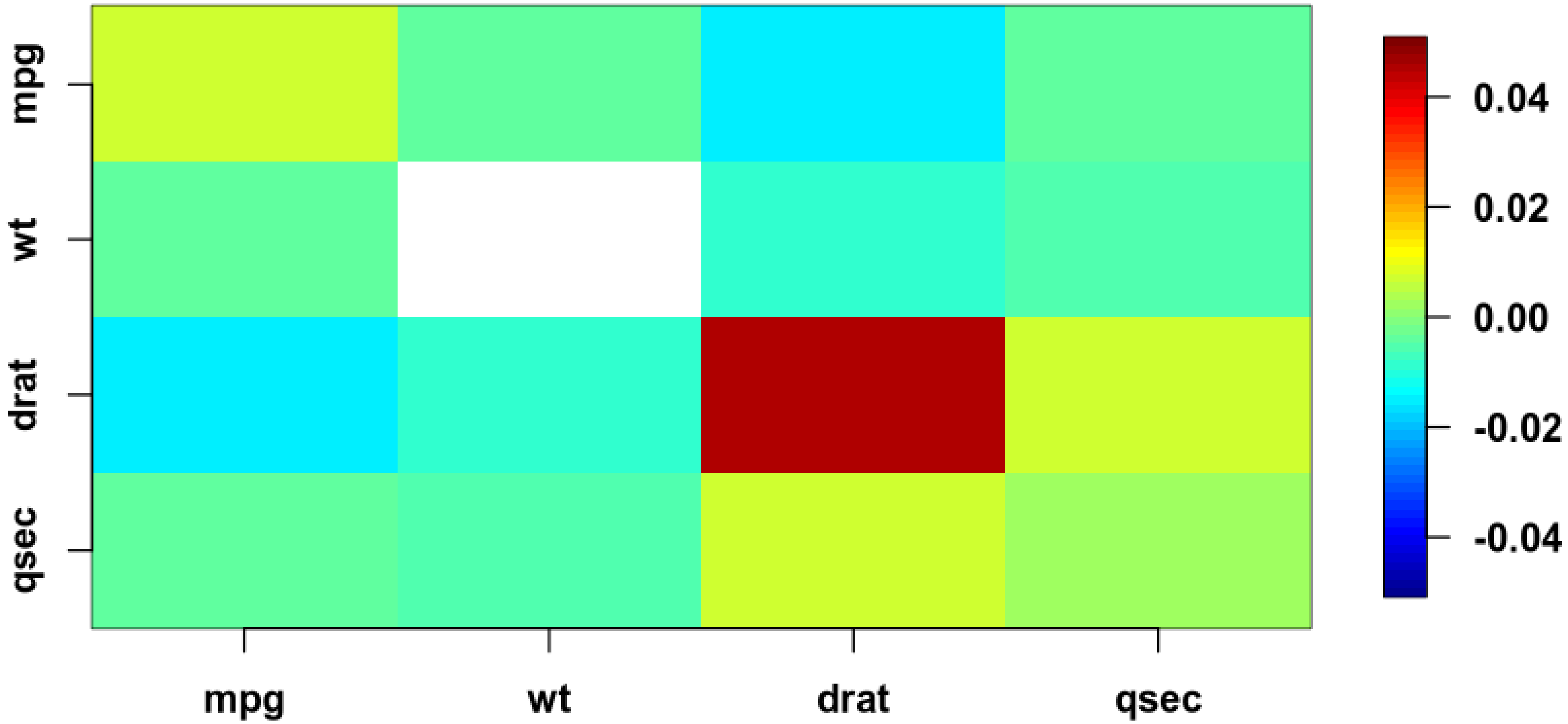
$$\left(X^T X + \lambda I\right)^{-1}; (\lambda = 0)$$



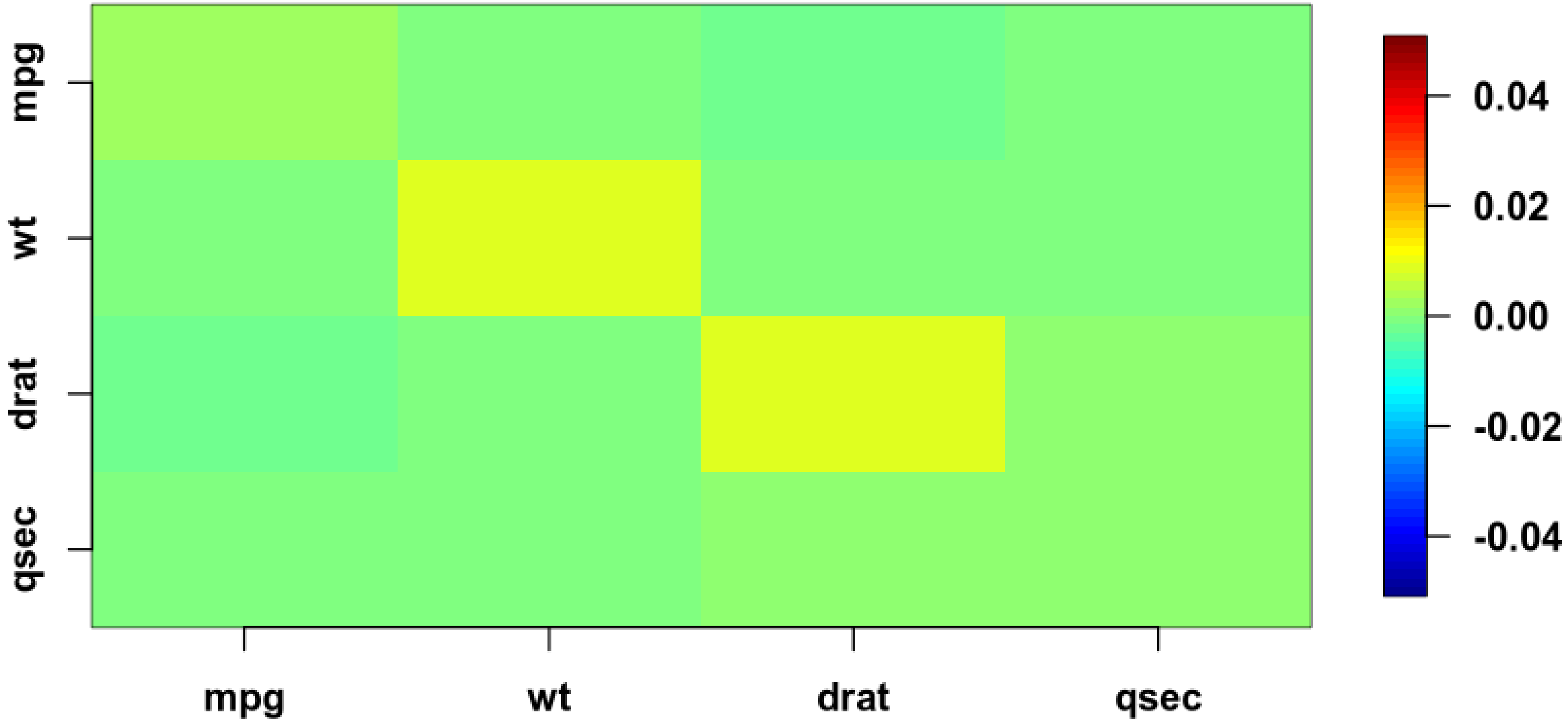
$$(X^T X + \lambda I)^{-1}; (\lambda = 1)$$



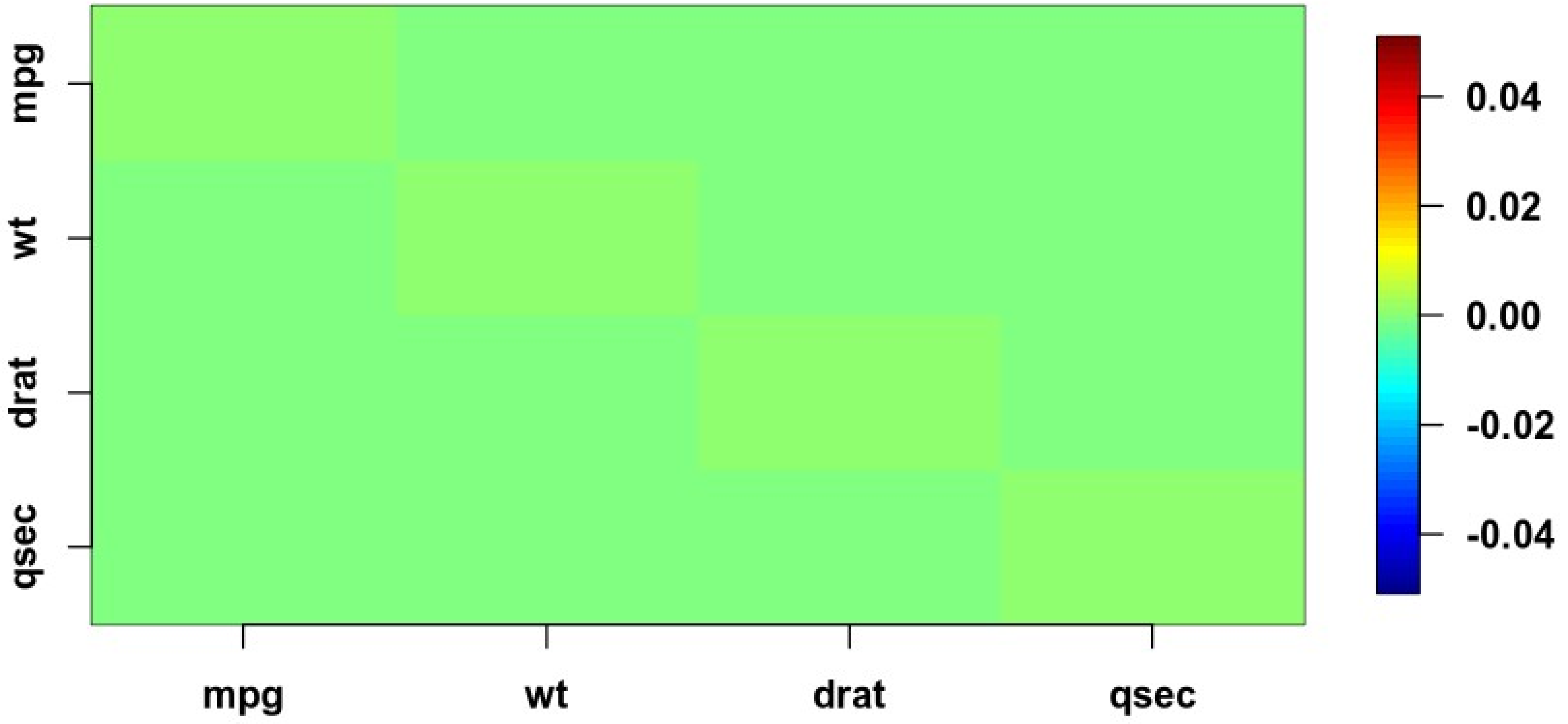
$$(X^T X + \lambda I)^{-1}; (\lambda = 10)$$



$$(X^T X + \lambda I)^{-1}; (\lambda = 100)$$



$$(X^T X + \lambda I)^{-1};(\lambda = 1000)$$



Regularisation decreases collinearity, but at what price?

I.e. what increases instead?

(Yarkoni and Westfall, 2017)

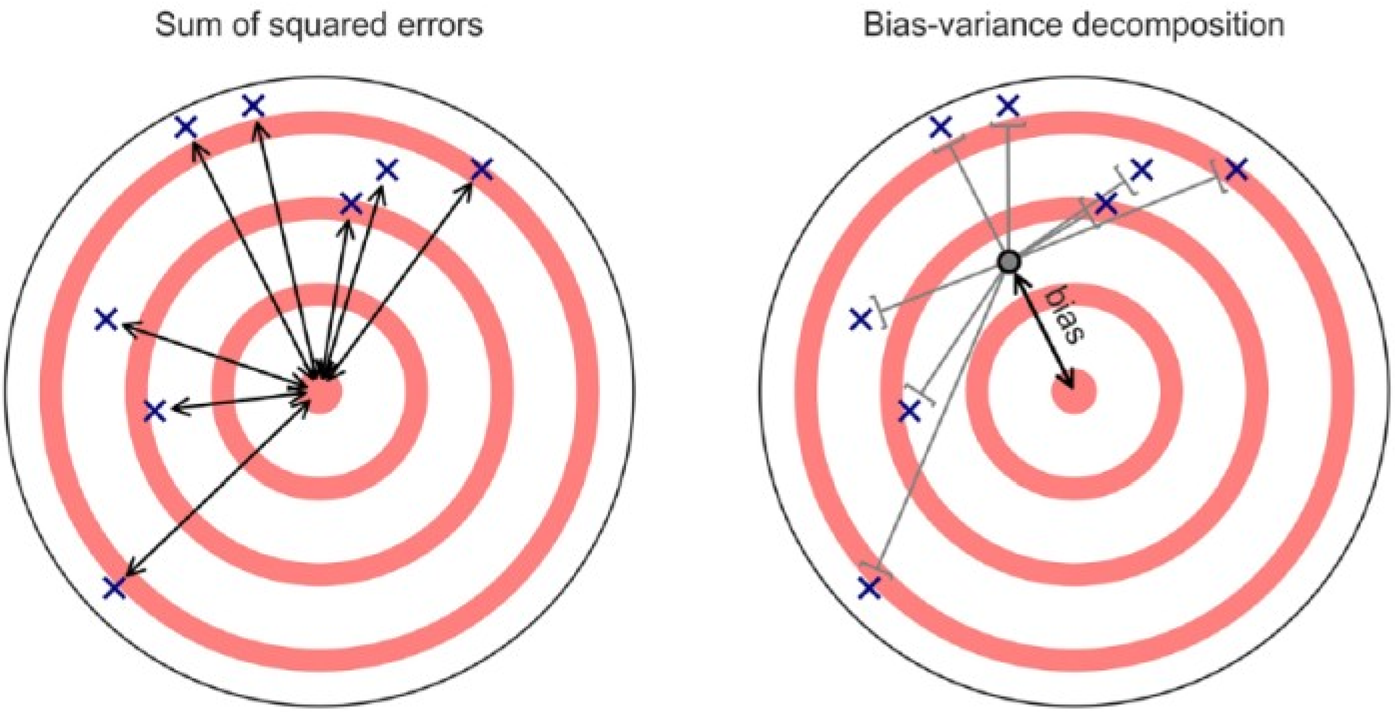


Fig. 3. Schematic illustration of the bias-variance decomposition. (Left) Under the classical error model, prediction error is defined as the sum of squared differences between true scores and observed scores (black lines). (Right) The bias-variance decomposition partitions the total sum of squared errors into two separate components: a bias term that captures a model's systematic tendency to deviate from the true scores in a predictable way (black line) and a variance term that represents the deviations of the individual observations from the model's expected prediction (gray lines).

Summary

- *R^2*, AIC and log-likelihood tests can all be adapted to multilevel models
  - However, some debate exists around what are the number of effective parameters
  - And none of them handle collinearity explicitly
- Regularisation gives a handle on collinearity
  - at the cost of introducing bias

Learning goals and outline

Evaluating Generalised linear mixed models

1) Learning tools for comparing models
    1) Variance explained
    2) Likelihood ratio tests
    3) Information criteria
2) Bridging to out-of-sample
    1) Introducing regularisation

The course plan

Week 1: *Introduction*

Instructor sessions: *Setting up* R *and* Python *and recollection of the general linear model*

Week 2: *Multilevel linear regression*

Instructor sessions: *Modelling subject level effects – and how do they differ from group level effects?*

Week 3: *Link functions and fitting generalised linear multilevel models*

Instructor sessions: *What to do when the response variable is not continuous?*

Week 4: *Evaluating Generalised linear mixed models*

Instructor sessions: *How do we assess how models compare to one another?*

Week 5: *Explanation and Prediction*

Instructor sessions: *Code review*

Week 6: *Mid-way evaluation and Machine Learning Intro*

Instructor sessions: *Getting Python Running*

Week 7: *Linear regression revisited (machine learning)*

Instructor sessions: *How to constrain our models to make them more predictive*

Week 8: *Logistic regression revisited (machine learning)*

Instructor sessions: *Categorizing responses based on informed guesses*

Week 9: *Dimensionality Reduction, Principled Component Analysis (PCA)*

Instructor sessions: *What to do with very rich data?*

Week 10: *Outlook, unsupervised classification and neural networks*

Instructor sessions: *Data with no labels and networks*

Week 11: *Organising and preprocessing messy data*

Instructor sessions: *Code review*

Week 12: Final evaluation and wrap-up of course

Instructor sessions: *Ask anything!*

Next time

- Penalised regression
  - Ridge and lasso regression
- Doing out-of-sample testing
  - Finding optimal lambda
- Bias variance decomposition
  - Increasing one, reduces the other

Reading questions

- Yarkoni and Westfall
  - What is overfitting?
  - What are training and test errors?
  - How is *p*-hacking and overfitting related?
  - How may *big data* guard against overfitting?
- Breimann
  - What are the *Data* and *Algorithmic Modeling Cultures*?
  - What is the Rashomon effect?
  - What is the Occam dilemma?