Multivariate models for analyzing data
(with categorical response)

Natalia Levshina ©

Digital Methods in Humanities and Social Sciences

University of Tartu, August 2018

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

Main concepts

- Dependent/response variable
- Independent variable/predictor
- Intercept
- Slope
- Residuals
- Deviance
- Fitted values
- Interaction
- Logit, log odds (ratio)
- Random intercept
- Random slope
- Smooth
- Mixed-effects models
- Generalized Additive Models
- Likelihood ratio test
- AIC
- *C*-index
- R^2
- Independence of observations
- Non-linearity
- Multicollinearity
- Priors
- Posterior probability
- MCMC chain

R packages

- rms
- visreg
- lme4
- Hmisc
- mgcv
- MuMIn
- brms

- To check and load:

```
> library(rms)
```

- To install, go to

Packages > Install

Type in the name of the package and press “Install”.

Rtools

- A package with a C++ compiler for Bayesian modelling
- https://cran.r-project.org/bin/windows/Rtools/

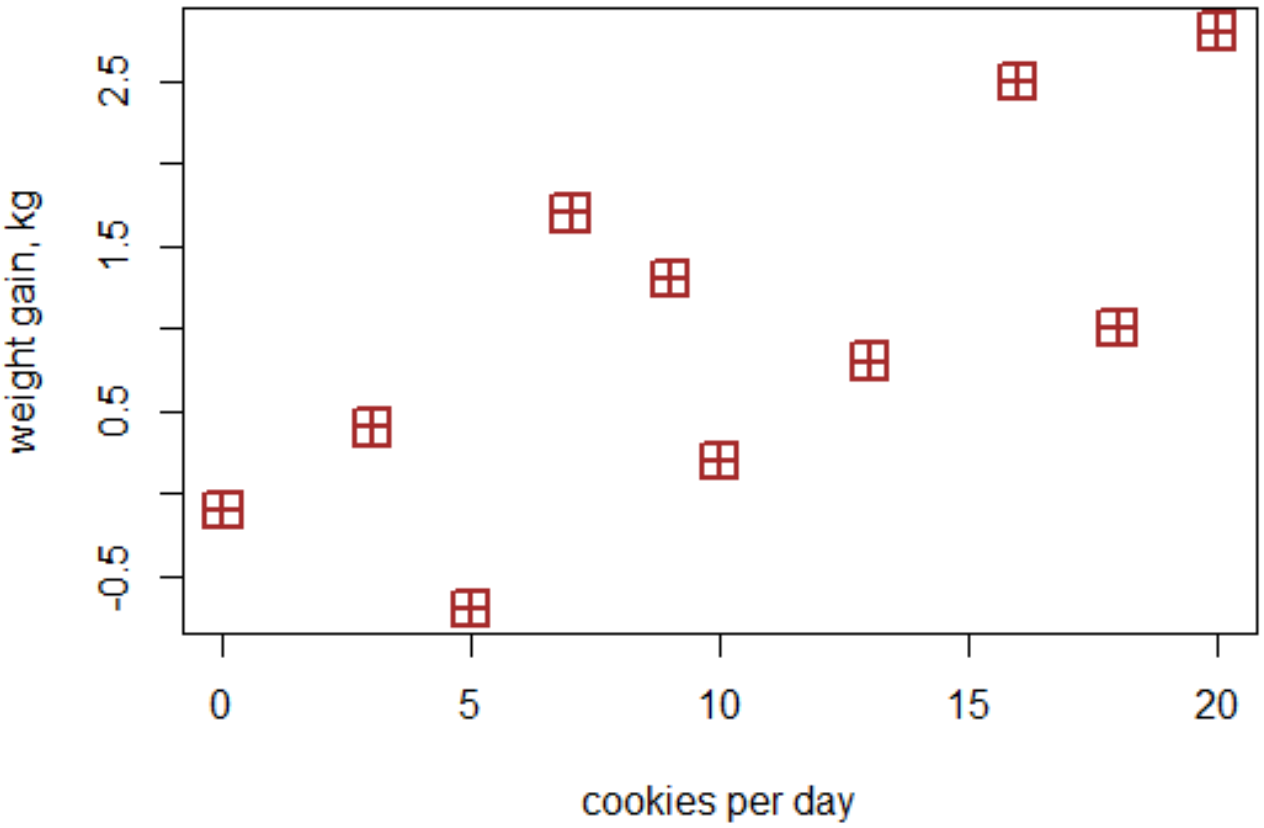
(the most recent version)

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

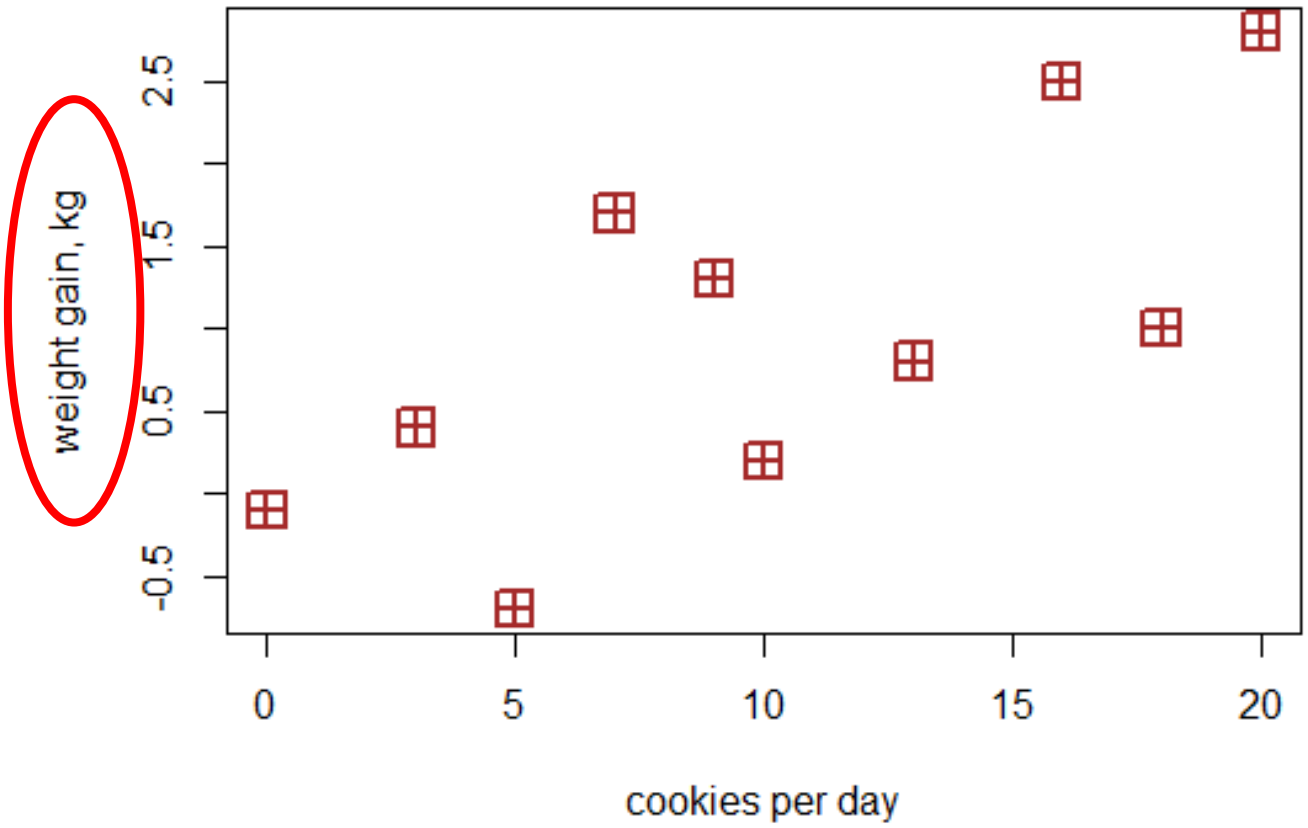
A cookie diet

Name	Cookies eaten per day	Kilos gained
John	0	-0.1
Mary	3	0.4
Bill	5	-0.7
Jane	7	1.7
Laura	9	1.3
Ann	10	0.2
Chris	13	0.8
Eve	16	2.5
Peter	18	1.0
Steve	20	2.8


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

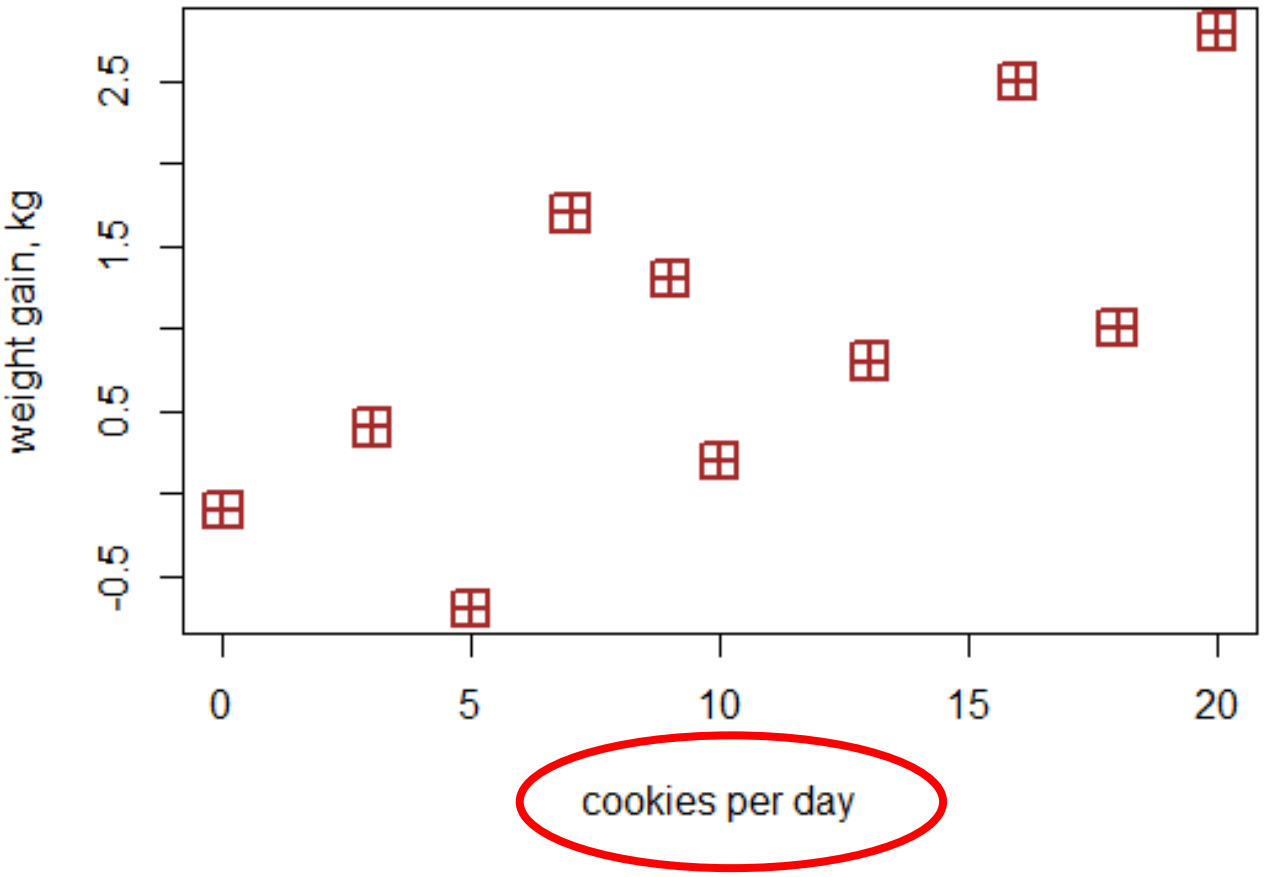
Fundamental concepts of regression

- Dependent variable (response): weight gain


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

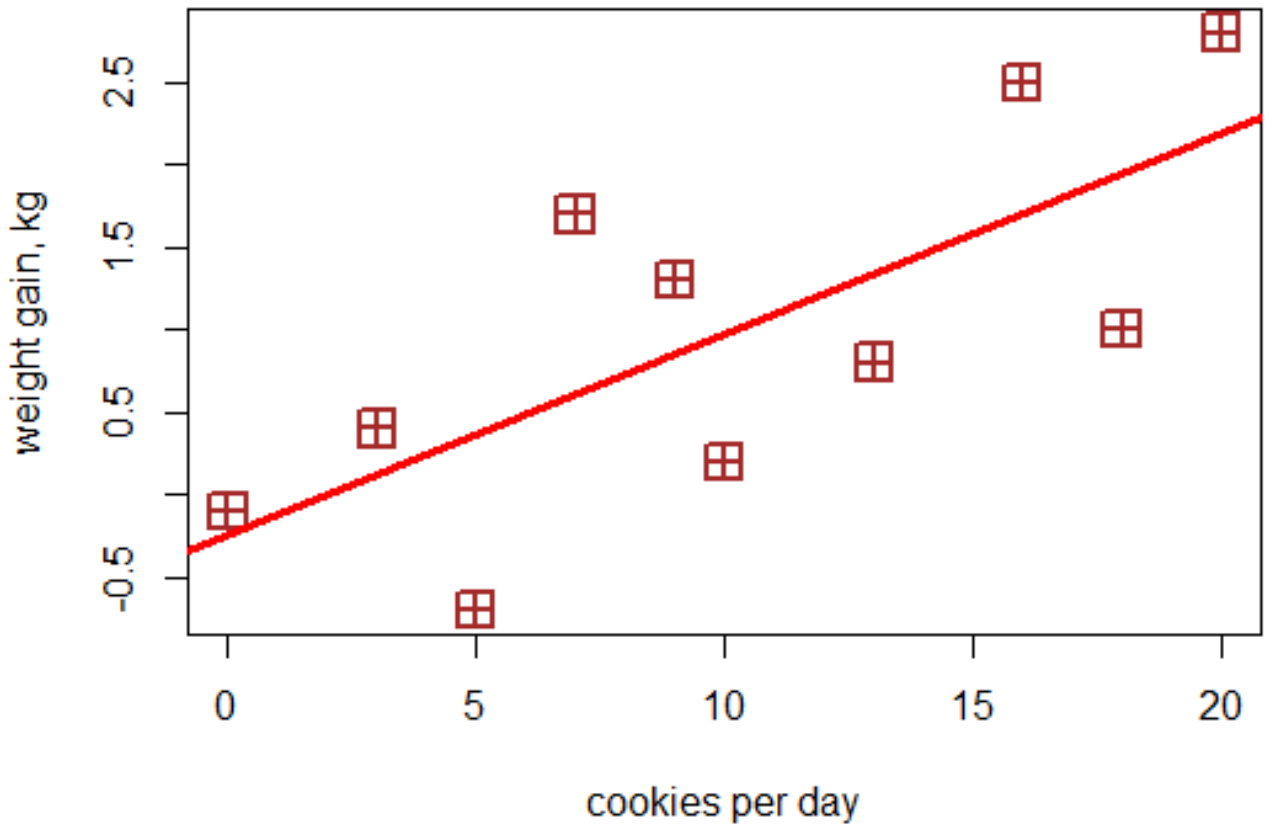
Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

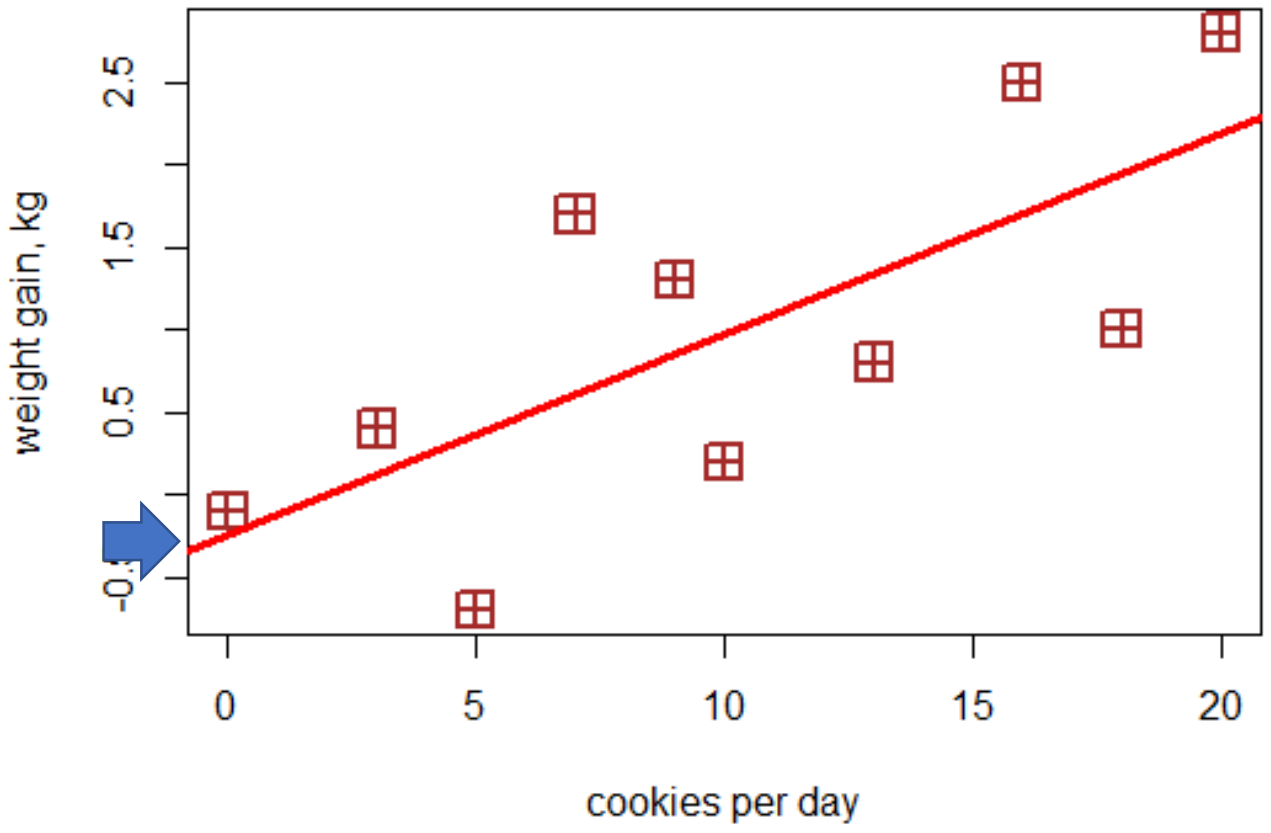
Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies
- Regression line: line that can be drawn through the cloud of points such as the distances between the line and all points are as small as possible


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

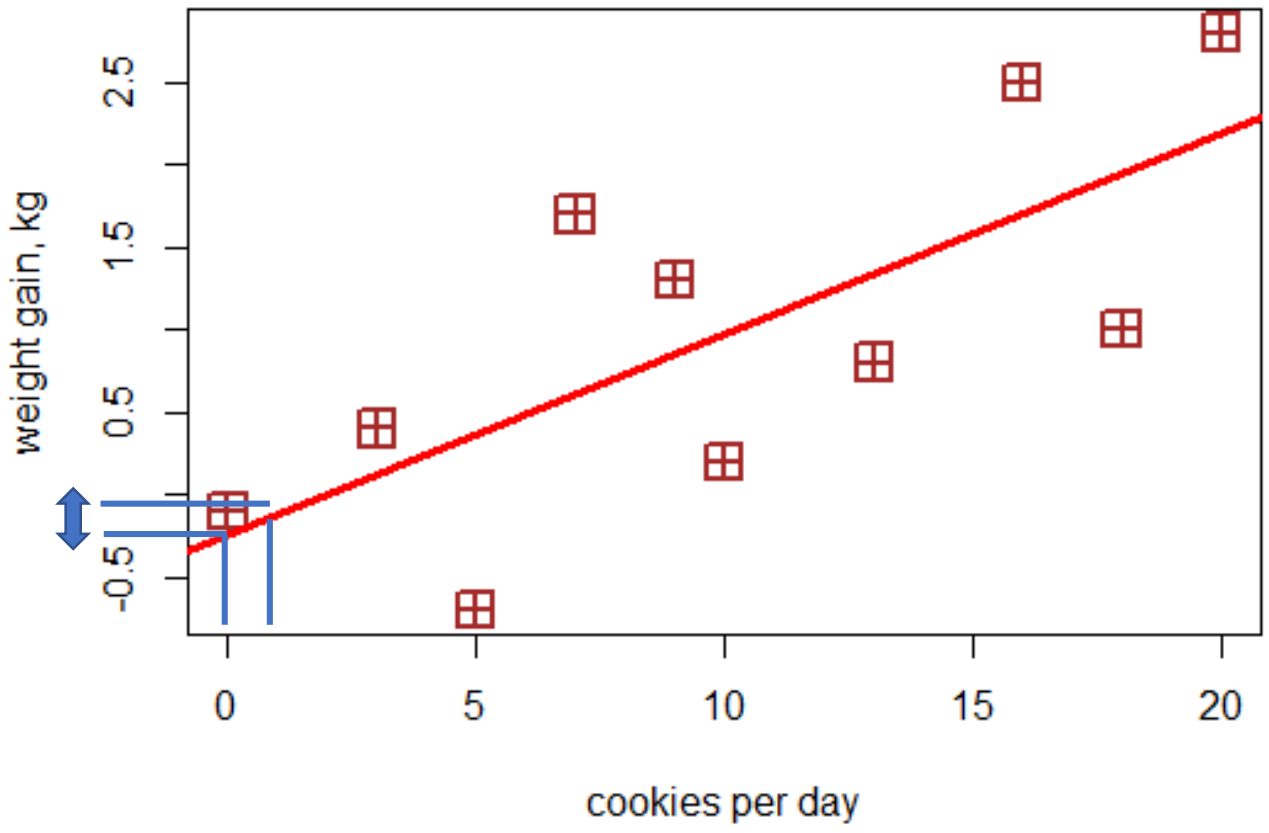
Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies
- Regression line: line that can be drawn through the cloud of points such as the distances between the line and all points are as small as possible
- Intercept: value of y where the line crosses the y-axis


weight gain, kg
2.5
1.5
0.5
0
5
10
15
20
cookies per day

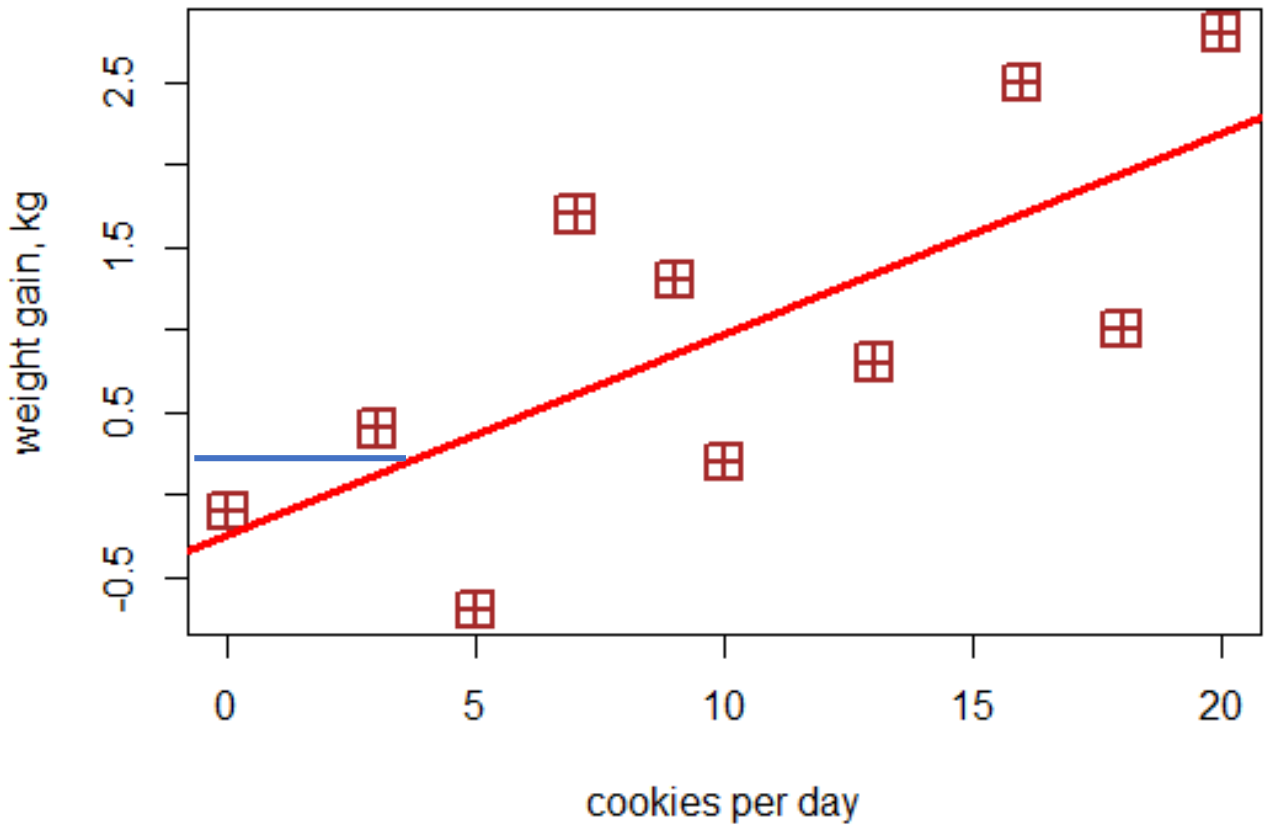
Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies
- Regression line: line that can be drawn through the cloud of points such as the distances between the line and all points are as small as possible
- Intercept: value of y where the line crosses the y-axis
- Slope: increase of y per unit of x on the line


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

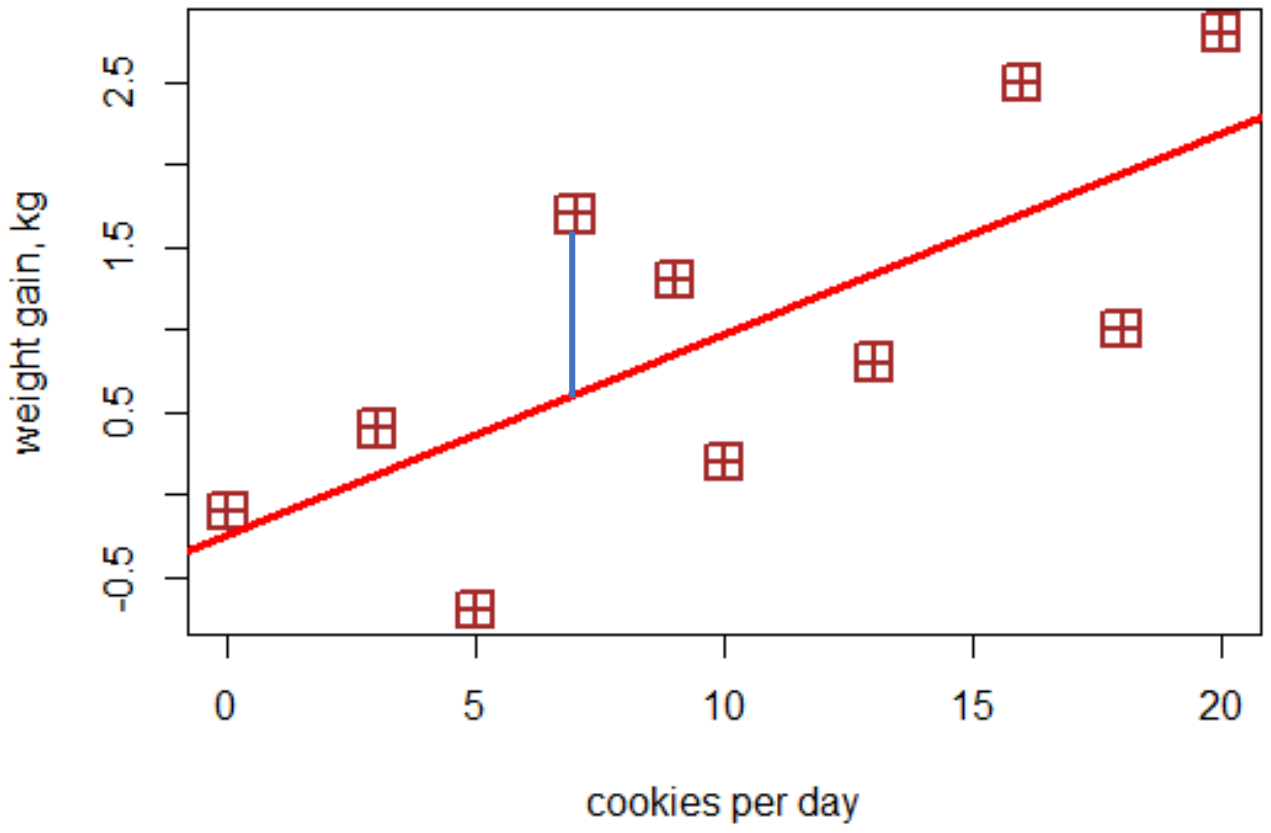
Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies
- Regression line: line that can be drawn through the cloud of points such as the distances between the line and all points are as small as possible
- Intercept: value of y where the line crosses the y-axis
- Slope: increase of y per unit of x
- Fitted values: the y-coordinates of the projections of the points on the line


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

Fundamental concepts of regression

- Dependent variable (response): weight gain
- Independent variable (predictor): cookies
- Regression line: line that can be drawn through the cloud of points such as the distances between the line and all points are as small as possible
- Intercept: value of y where the line crosses the y-axis
- Slope: increase of y per unit of x
- Fitted values: the y-coordinates of the projections of the points on the line
- Residuals: the differences between the observed and fitted response values (the distances between the points and their projections on the line)


weight gain, kg
-0.5
0.5
1.5
2.5
0
5
10
15
20
cookies per day

The magic of linear regression

Observed value of **y** =

Intercept **α**

\+

Slope **β***value of **x**

} Fitted value **ȳ**

\+

Residual **ε**

$$y = \alpha + \beta x + \varepsilon = \bar{y} + \varepsilon$$

Exercise

- Your colleague fitted a regression model which shows that happiness measured on a scale from 0 to 100 depends on Kalev chocolate (in grams).
- The formula looks as follows:

$$\text{Happiness} = 38 + 0.5*\text{Chocolate} + \text{Error}$$

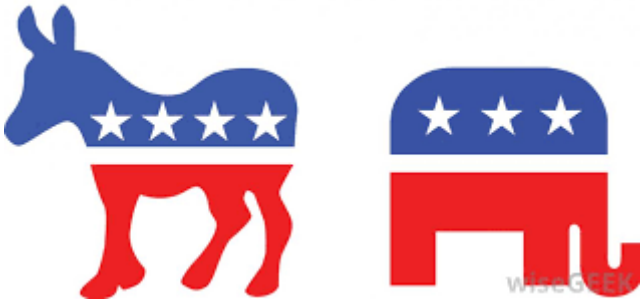
- How can you interpret these numbers?
- How happy will you be if you eat 50 grams, as predicted by the model? If you eat 100 grams?

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

Logistic regression

- Situations with two or more categorical outcomes:



Various applications

- Banking: will Mr Smith pay back the mortgage, depending on his income, education, previous credits, family status and health condition?
- Health: will a patient live or die, depending on her desease, general health condition, age and gender?
- Language use: will a speaker use gonna or going to, depending on the variety of English (e.g. AmE vs. BrE) and formality of communication?

Probabilities, odds and log-odds (logit)

	PROBABILITIES	ODDS	LOG ODDS
Outcome 1 is as probable as Outcome 2	0.5 (or 50%)	1	Log (1) = 0
Outcome 1 is more probable than Outcome 2	> 0.5 (or 50%)	> 1	Positive
Outcome 1 is less probable than Outcome 2	< 0.5 (or 50%)	< 1	Negative

Examples of odds and log odds

- In town X live 500 liars and 500 truth tellers. If you meet someone from X by chance, what is the probability that this person is a liar? The odds? The log-odds?
- The demographic situation has changed. There is only 200 liars and 800 truth tellers. What is the probability and the odds and log-odds of meeting a liar?
- Now there are 800 liars and 200 truth tellers. What are the odds and probability of meeting a liar now?

The basic idea of a logistic model

- Linear model:

$$y = \alpha + \beta x + \varepsilon$$

- Logistic generalized linear model:

$$\log\left(\frac{P\,(outcome\ 1)}{1 - P\,(outcome\ 1)}\right) = \alpha + \beta x + \varepsilon$$

Disclaimer!

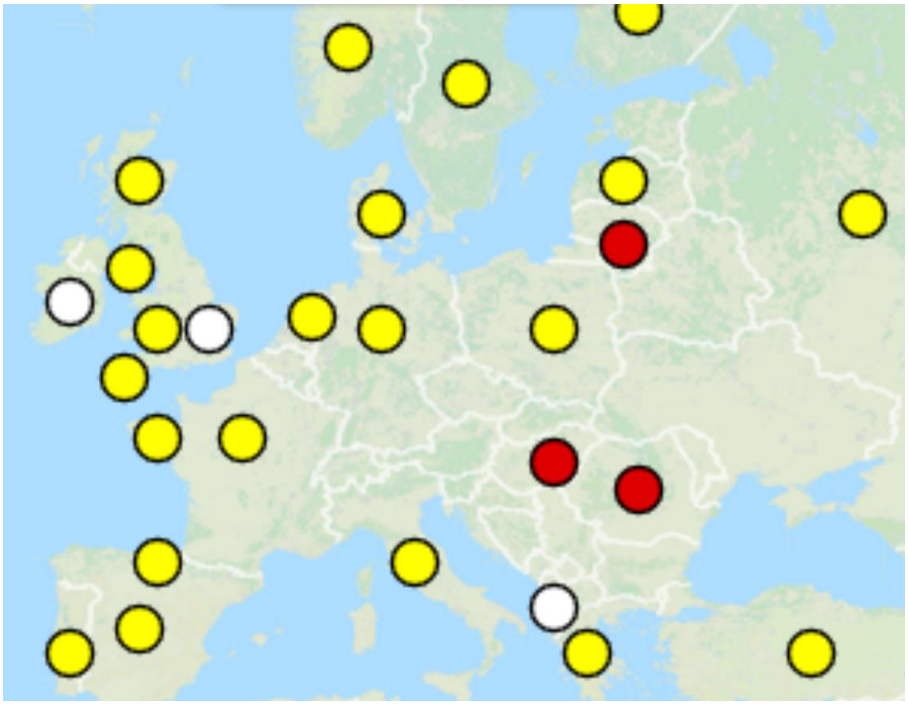
- We will discuss how to fit logistic models step by step. The intermediary models we'll be fitting are not 'correct', and should not be reported! Their purpose is to illustrate the basic concepts and techniques only.

T or V?

- The distinction is present in most European languages
  - T forms: informal, familiar, e.g. French *tu*, German *du*, Russian *ty* + Verb 2nd SG
  - V forms: formal, polite, e.g. French *vous*, German *Sie*, Russian *vy* + Verb 2nd PL or 3rd SG/PL

Cross-linguistic research

- WALS Chapter 45, Helmbrecht 2013



Values

○	No politeness distinction	136
●	Binary politeness distinction	49
●	Multiple politeness distinctions	15
●	Pronouns avoided for politeness	7

Power and solidarity (Brown and Gilman 1960)

- Power dimension:
  - Based on “older than”, “richer than”, “parent of”, etc.
  - T = lower status of the Hearer, V = higher status of the Hearer
  - Systematic distinction already in the late Middle Ages. Everyone had his/her fixed place in the society.

- Solidarity dimension:
  - Based on “the same age/family/class as”.
  - T = closeness, V = distance
  - Emerged with social mobility and egalitarian ideology. Starting from the French revolution (*Citoyen*, *tu*).
  - Currently dominates in major European languages, but there are subtle cross-linguistic differences.

Survey

- See supplementary materials

Data for the case study

- T/V forms
- 50 subjects, 18 questions
- Communicative situations:
  - Q_ID (ID of the question in the questionnaire)
  - Familiarity (Close, Middle and Far)
  - H_Age (Younger, Older and Same)
- Subject's characteristics
  - S_ID (Subject's ID)
  - S_Extrav (index of extraversion)
  - S_Age (age)

Data in R

```
> str(tvdata)
'data.frame': 900 obs. of  9 variables:
$ S_ID        : Factor w/ 50 levels "1","2","3","4",..:
$ S_Year      : int  1934 1934 1934 1934 1934 1934 1934
$ S_Extrav    : int  74 74 74 74 74 74 74 74 74 74 ...
$ S_Age       : num  84 84 84 84 84 84 84 84 84 84 ...
$ Q_ID        : int  1 2 3 4 5 6 7 8 9 10 ...
$ Familiarity: Factor w/ 3 levels "Close","Far",..: 1 1
$ H_Age       : Factor w/ 3 levels "Older","Same",..: 3 2
$ Form        : Factor w/ 2 levels "T","V": 2 1 2 2 2 2
$ Name        : Factor w/ 3 levels "Short","First"
```

Logistic regression in R

```
> mymodel <- glm(Response ~ Predictor1 + Predictor 2…, data = tvdata, family = binomial)
```

```
> summary(glm)
```

…

Simple model: 1 num. predictor

```
> mymodel <- glm(Form ~ S_Extrav, data = tvdata,
family = binomial)
> summary(mymodel)

Call:
glm(formula = Form ~ S_Extrav, family = binomial, data =
tvdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3532  -1.1535  -0.9118   1.1552   1.4687

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.501240   0.147543   3.397 0.000681 ***
S_Extrav    -0.012126   0.003028  -4.005  6.2e-05 ***
```

Slope of a numeric predictor

- The beta (slope) of a numeric predictor shows how much the log odds of V vs. T increase if one increases the unit of the predictor by 1.
- Here, we see the change in the log odds for 1 point on the extraversion scale.
- The slope of -0.01 means that for every additional point on the extraversion scale, the log-odds of V/T gain -0.01 (or decrease by 0.01).
- This number represent factor (as “by factor of”). To extract this factor, one should exponentiate (i.e. to apply a function opposite to log).

```
> exp(-0.012126)
[1] 0.9879472
```

That is, with every point of extraversion scale, the odds of V against T change by factor of 0.99.

The odds of what to what?

- How to check what kind of odds we are measuring, odds of V vs. T or T vs. V?
- It's easy:

```
> levels(tvdata$Form)
[1] "T" "V"
```

The model shows the log-odds for the second outcome, i.e. V, vs. the first outcome, i.e. T!

- How to change the order:

```
> tvdata$Form_rev <- relevel(tv_data$Form,
ref = "V")
```

Exercise

- Fit a simple logistic model with S_Age as the predictor. Interpret the result.

Simple model: 1 cat. predictor

```
> mymodel <- glm(Form ~ Familiarity, data = tvdata, family = binomial)
> summary(mymodel)
Call:
glm(formula = Form ~ Familiarity, family = binomial, data = tvdata)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6031  -0.5701  -0.5701   0.9998   1.9479
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.7346     0.1617  -10.73   <2e-16 ***
FamiliarityFar      2.6957     0.2069   13.03   <2e-16 ***
FamiliarityMiddle   2.1679     0.2003   10.82   <2e-16 ***
…
```

Meaning of the slope for a categorical predictor

- By default (so-called treatment coding), the slope coefficient shows the difference between the log odds of V vs. T (log odds ratio) for the category specified in the output in comparison with the reference level (not shown in the output).
- Familiarity = Far increases the log odds of V against T in comparison with Familiarity = Close (the reference level, not shown). The difference between the log odds is 2.7 (log-odds ratio).
- To get the simple odds ratio, use **`exp()`**:

```
> exp(2.69)
[1] 14.73168
```

The odds of V against T increase by factor of 14 (i.e. are 14 times greater) in comparison with Familiarity = Close!

For a deeper understanding

```
> table(tvdata$Form, tvdata$Familiarity)

   Close Far Middle
T   255  83    118
V    45 217    182
```

Some nuances

- Odds:

```
> odds_close <- 45/255
> odds_far <- 217/83
> odds_far/odds_close
[1] 14.81526
```

The slope coefficient as a simple odds ratio shows the **factor** of the odds (i.e. by how many times the odds are greater or smaller)!

- Log-odds

```
> log_odds_close <- log(45/255)
> log_odds_far <- log(217/83)
> log_odds_far - log_odds_close
[1] 2.695658
```

Beta as a log-odds ratio shows the **difference** between the log-odds!

Exercise

- Fit a simple logistic model with H_Age as the predictor. Interpret the result.
- Change the reference level to “Younger”. How can you interpret the difference?

Combining several predictors

```
> mymodel <- glm(Form ~ S_Extrav + S_Age + H_Age + Familiarity, data = tvdata, family = binomial)
> summary(mymodel)
[…]Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.030671   0.339329   0.090    0.928
S_Extrav           -0.018570   0.003779  -4.914 8.92e-07 ***
S_Age              -0.003622   0.003957  -0.915    0.360
H_AgeSame          -1.088334   0.212578  -5.120 3.06e-07 ***
H_AgeYounger       -2.019630   0.220811  -9.146  < 2e-16 ***
FamiliarityFar      3.156725   0.236553  13.345  < 2e-16 ***
FamiliarityMiddle   2.537195   0.225126  11.270  < 2e-16 ***
```

What you need to know for regression diagnostics

- (Log) likelihood is a measure of how close the fitted values are to the observed values of the response variable. The higher, the better (note that Maximal Likelihood is the estimation method for logistic regression).
- Deviance represents the (minus doubled) difference between the log-likelihood of the saturated model, i.e. with the perfect fit, and the log-likelihood of the current model. The smaller the deviance, the better the model fits the data.
- One can also compute the difference between likelihoods of two possible models. This difference is used in Likelihood ratio test (LRT), a very useful tool for comparison of two possible models.
- Another useful statistic is AIC (Akaike's Information Criterion), which helps us to identify the most parsimonious model. It increases with the number of parameters and decreases with likelihood. The smaller AIC, the better.

How to compare models

- **Very important: One cannot use these statistics to compare models based on different data!**

MEASURE	BETTER MODEL	WORSE MODEL
Deviance	Low	High
(log) Likelihood	High	Low
AIC	Low	High

Which predictors are useful?

- Do we need S_Age? Let's fit a model without this predictor:

```
> mymodel1 <- glm(Form ~ S_Extrav + H_Age + Familiarity, data = tvdata, family = binomial)
> anova(mymodel, mymodel1, test = "Chisq")
Analysis of Deviance Table
Model 1: Form ~ S_Extrav + S_Age + H_Age + Familiarity
Model 2: Form ~ S_Extrav + H_Age + Familiarity
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       893     890.10
2       894     890.94 -1 -0.83907   0.3597
```

Dropping terms one by one

```
> drop1(mymodel, test = "Chisq")
Single term deletions

Model:
Form ~ S_Extrav + S_Age + H_Age + Familiarity
            Df Deviance     AIC     LRT  Pr(>Chi)
<none>           890.10  904.10
S_Extrav     1   915.28  927.28  25.176 5.234e-07 ***
S_Age        1   890.94  902.94   0.839    0.3597
H_Age        2   987.16  997.16  97.059 < 2.2e-16 ***
Familiarity  2  1159.76 1169.76 269.658 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
```

Multicollinearity

- M. is a condition when predictors are strongly intercorrelated. Regression algorithm then has problems estimating their effects. As a result, the p-values are usually high.
- Testing M. with Variance Inflation Factors (should be less than 5):

```
> library(rms)
> vif(mymodel) #note that there's a similar function in package car
          S_Extrav             S_Age
          1.051529          1.016670
       H_AgeSame     H_AgeYounger
          1.512983          1.630605
   FamiliarityFar FamiliarityMiddle
          1.877028          1.799683
```

Testing Interactions

```
> mymodel2 <- glm(Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity, data = tvdata, family = binomial)
> anova(mymodel1, mymodel2, test = "Chisq")
Analysis of Deviance Table

Model 1: Form ~ S_Extrav + H_Age + Familiarity
Model 2: Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       894     890.94
2       890     855.08  4   35.863 3.088e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interaction in the table of coefficients

```
> summary(mymodel2)
```

…

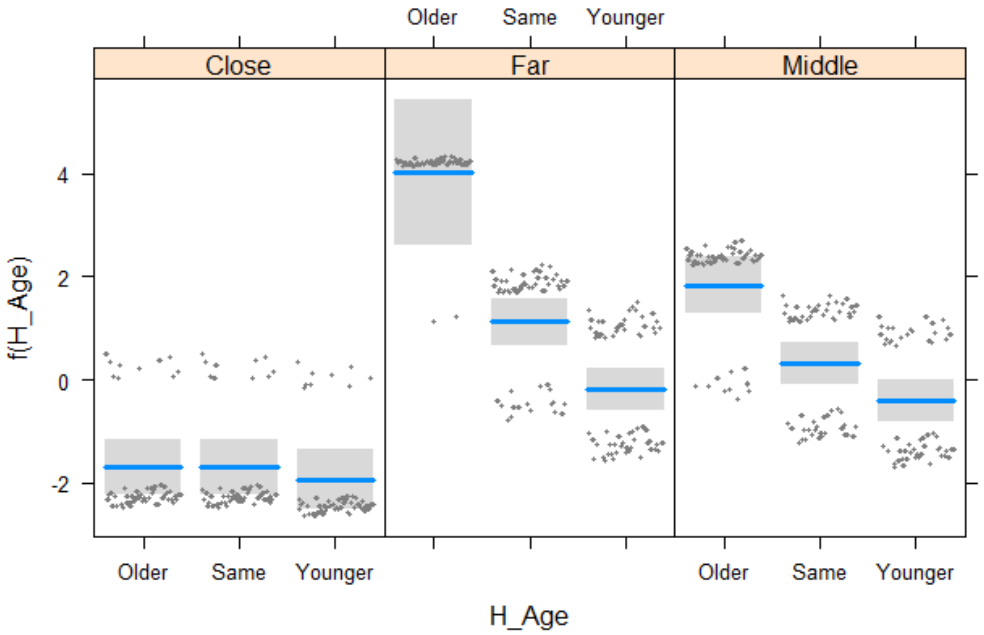
```
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -8.813e-01  3.118e-01  -2.827 0.004706 **
S_Extrav                        -1.921e-02  3.889e-03  -4.939 7.85e-07 ***
H_AgeSame                       -1.669e-14  3.899e-01   0.000 1.000000
H_AgeYounger                    -2.476e-01  4.075e-01  -0.608 0.543485
FamiliarityFar                   5.706e+00  7.697e-01   7.413 1.23e-13 ***
FamiliarityMiddle                3.520e+00  3.989e-01   8.823  < 2e-16 ***
H_AgeSame:FamiliarityFar        -2.897e+00  8.479e-01  -3.417 0.000633 ***
H_AgeYounger:FamiliarityFar     -3.951e+00  8.500e-01  -4.649 3.34e-06 ***
H_AgeSame:FamiliarityMiddle     -1.509e+00  5.252e-01  -2.873 0.004071 **
H_AgeYounger:FamiliarityMiddle -1.979e+00  5.396e-01  -3.668 0.000244 ***
```

Meaning of interaction terms

- Interaction is observed when the effect of two or more predictors on the response is not additive.
- The interaction term in R (by default) shows how much should be added to the fitted log-odds for the levels specified in the interaction.
- The best way of interpreting an interaction is visualization.

Interpreting interactions

```
> library(visreg)
> visreg(mymodel2, xvar = "H_Age", by = "Familiarity")
```



Two equivalent ways of specifying an interaction in R

- In the model formula, you can specify an interaction between predictor X1 and X2 in two equivalent ways:

$$Y \sim X1*X2$$

or

$$Y \sim X1 + X2 + X1:X2$$

Exercise

- Add to the model and test an interaction between S_Extrav and H_Age. Create an interaction plot.

Goodness of fit

- How well does the model fit the data?
- Two most popular measures:
  - Pseudo-R^2 (based on likelihood), from 0 (useless) to 1 (perfect).
  - *C*-index (or concordance index), from 0.5 (random) to 1 (perfect), at least 0.7 for a model to be reported.
- For fixed-effects logistic models, it's convenient to use function `lrm()` from package rms.

A lrm() model

```
> library(rms)
> mymodel_lrm <- lrm(Form ~ S_Extrav + H_Age*Familiarity, data = tvdata) #you don't need family = binomial!
> mymodel_lrm #you don't need summary()!
```

…

Useful statistics

Model Likelihood Ratio Test		Discrimination Indexes		Rank Discrim. Indexes	
LR chi2	392.43	R2	0.471	C	0.850
d.f.	9	g	2.110	Dxy	0.701
Pr(> chi2)	<0.0001	gr	8.247	gamma	0.702
		gp	0.352	tau-a	0.351
		Brier	0.156		

…

Exercise

- Fit a model with an interaction between S_Extrav and H_Age. Does the goodness of fit of the model improve?

Exercise: data

- Do you like cats or dogs?
- The dataset catdog contains observations on 200 subjects who either prefer dogs or cats (variable 'Animal').
- Other variables:
  - Gender (M or F)
  - Extrav (extraversion, on a scale from 0 to 100)
  - Liter (preference for poetry or prose)

Exercise: task

Perform logistic regression modelling in order to test if the preference for cats or dogs depends on the other variables:

1. Select the best model with all relevant predictors.
2. Test all interactions and interpret the relevant ones.
3. Check the goodness of fit.

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

Assumption of independence

- Statistical independence: occurrence of one event does not influence the probability of occurrence of another event.
- In our example: the choice for T or V in one cell in the data frame does not influence the choice for T or V in another.
- This assumption is obviously violated: the dataset contains observations from 50 individuals (18 from each person).

Random intercepts and slopes

- Some subjects may prefer T more than others (due to some personality characteristics, life experience that we cannot capture). -> Random intercepts

- The effect of some variables on the choice for T or V may be different across the subjects. -> Random slopes

Random intercepts in R

```
> library(lme4)
> mymodel3 <- glmer(Form ~ S_Extrav + H_Age*Familiarity + (1|S_ID), data = tvdata, family = binomial)
```

```
Warning message:
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with max|grad| = 0.073256 (tol = 0.001, component 1)
```

Fix the convergence issue

- Add some controls:

```
> mymodel3 <- glmer(Form ~ S_Extrav +
H_Age*Familiarity + (1|S_ID), data = tvdata,
family = binomial, control =
glmerControl(optimizer="bobyqa", optCtrl =
list(maxfun = 100000))
```

Interpret glmer output

```
> summary(mymodel3)
```

…

What has changed?

Use visreg to study the interaction

```
> visreg(mymodel3, xvar = "H_Age", by = "Familiarity")
```

Note: no confidence bands!

Where are the random effects?

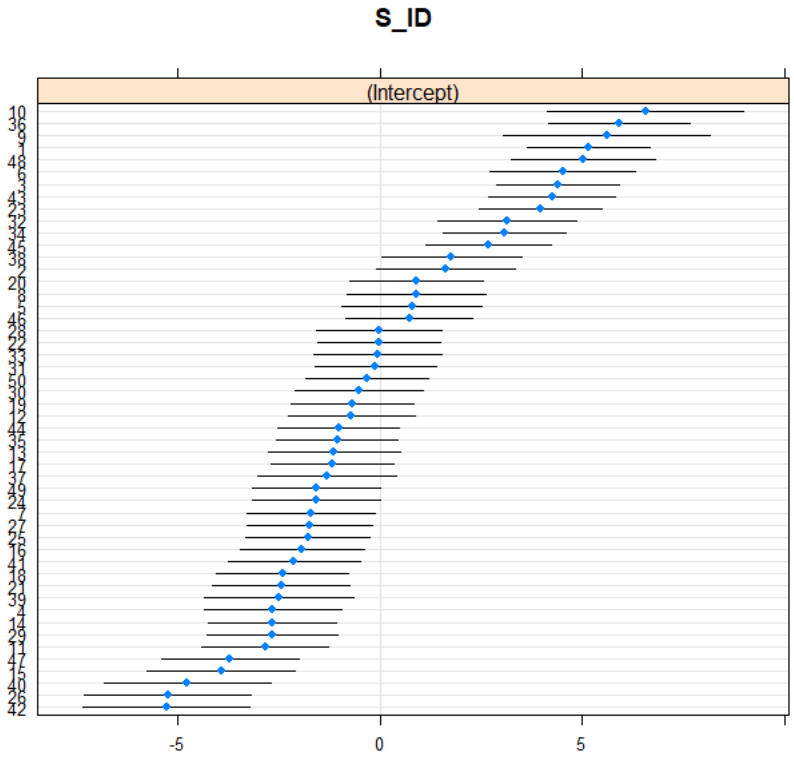
```
> ranef(mymodel3)
$S_ID
   (Intercept)
1    5.16192375
2    1.62200270
3    4.39230429
4   -2.64867196
5    0.79411251
6    4.51858577
7   -1.70616250
…
```

Variance of random effects

```
…
Random effects:
 Groups Name        Variance Std.Dev.
 S_ID   (Intercept) 10.33    3.215
Number of obs: 900, groups:  S_ID, 50
```

Plotting random intercepts

```
> lattice::dotplot(ranef(mymodel3, condVar = TRUE))
```



How to decide on random effects?

- In other words, is their variance different from 0? How to compare glmer vs. glm.
- Unfortunately, the LRT does not work directly (problems with comparing different log-likelihoods – computed differently)
- However, AIC can be compared:

```
> AIC(mymodel2)
[1] 875.0795
> AIC(mymodel3)
[1] 572.4117
```

A little trick

Make a mixed model where all observations belong to the same group 1 and fit a mixed model.

```
> idconst <- factor(rep(1, nrow(tvdata))
> summary(idconst)
   1
900
> trick <- glmer(Form ~ S_Extrav + H_Age*
Familiarity +(1|idconst), data = tvdata,
family = binomial, control =
glmerControl(optimizer="bobyqa",
check.nlev.gtr.1="ignore", optCtrl =
list(maxfun = 10000000)))
```

https://stats.stackexchange.com/questions/56150/how-can-i-test-whether-a-random-effect-is-significant , based on N. W. Galwey Introduction to Mixed Modelling: Beyond Regression and Analysis of Variance', 213-214.

Perform LRT

```
> anova(trick, mymodel3, test = "Chisq")
Data: tvdata
Models:
trick: Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity + (1 |
test:      idconst)
mymodel3: Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity + (1
|
mymodel3:      S_ID)
          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
trick     11 877.08 929.91 -427.54   855.08
mymodel3 11 572.41 625.24 -275.21   550.41 304.67      0  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Adding random slopes

What if the subjects have different views about how they should address people of different age?

```
> mymodel4 <- glmer(Form ~ S_Extrav + H_Age*
Familiarity + (1 + H_Age|S_ID), data =
tvdata, family = binomial, control =
glmerControl(optimizer="bobyqa", optCtrl =
list(maxfun = 10000000)))
```

Or simply `(H_Age|S_ID)`

Do we need the random slopes?

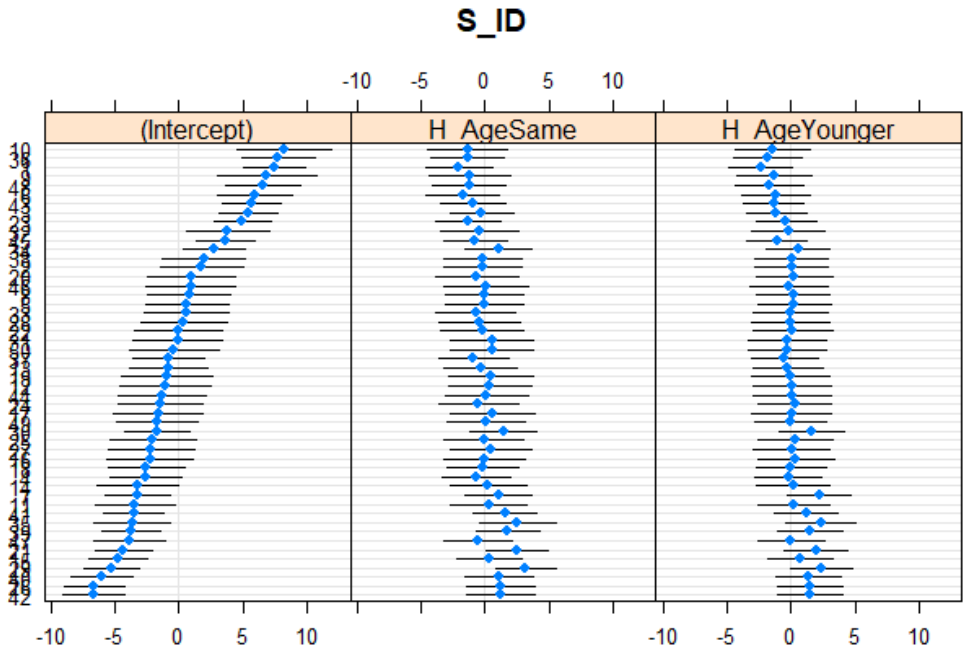
```
> anova(mymodel3, mymodel4, test = "Chisq")
Data: tvdata
Models:
mymodel3: Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity
+ (1 | S_ID)
mymodel4: Form ~ S_Extrav + H_Age + Familiarity + H_Age:Familiarity
+ (1 + H_Age | S_ID)
          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mymodel3 11 572.41 625.24 -275.21   550.41
mymodel4 16 578.55 655.39 -273.28   546.55 3.8599      5     0.5698
```

Where to find the random effects

```
> ranef(mymodel4)
$S_ID
   (Intercept)    H_AgeSame H_AgeYounger
1   7.47175239 -2.040599146 -2.381343033
2   1.75050940 -0.154422635  0.064950956
3   5.43291054 -0.258438518 -1.142634471
4  -2.55714510 -0.694613032 -0.218784795
5   0.62683719 -0.004011498  0.223731359
6   5.92391467 -1.753830152 -1.227288911
7  -3.23246413  1.082247382  2.212249908
```

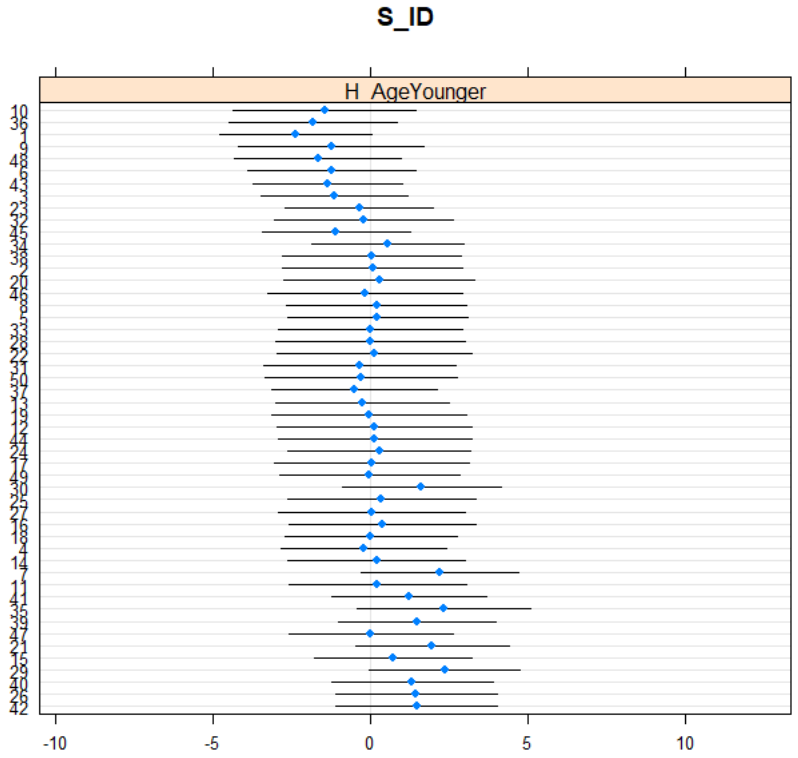
Plotting the random effects

```
> lattice::dotplot(ranef(mymodel4, condVar = TRUE))
```



Plotting only some columns

```
> lattice::dotplot(ranef(mymodel4, condVar = TRUE))$S_ID[3]
```



Exercise

- Test if we need to add random slopes for Familiarity.

Correlations between random effects

- When there is not enough data, one can find correlations between random intercepts and slopes to be equal to -1 or +1. The correlations can be found here:

```
> summary(mymodel4)
```

…

```
Random effects:
 Groups Name          Variance Std.Dev. Corr
 S_ID   (Intercept)   18.434   4.293
        H_AgeSame      3.423   1.850    -0.61
        H_AgeYounger   3.084   1.756    -0.66  0.81
```

- If any correlation is equal to +1 or -1, this means basically that the model has reached its limits and hasn't converged.

What to do?

- You can tell R that the random effects should not be correlated.
- Two solutions:

```
… + (1 + H_Age||S_ID)
```

or

```
… + (1|S_ID) + (0 + H_Age|S_ID)
```

Some notes about mixed models

- Some authors suggest using maximal random effect structure (all possible random effects and slopes). However, there is evidence that this approach leads to loss of statistical power (e.g. Bates et al. 2015).
- For comparison of fixed effects, linear mixed models should be fitted with ML, for better estimation of random effects with REML (restricted maximum likelihood).
- Currently, lme4 does not have REML for glmer (instead, Laplace approximation of ML), so we don’t need to worry about that now.

Goodness of fit for glmer

Pseudo R^2:

```
> library(MuMIn)
> r.squaredGLMM(mymodel3)
The result is correct only if all data used by the model has not changed since model was fitted.
      R2m       R2c
0.5017074 0.8728366
```

R2m: marginal (only fixed effects)

R2c: conditional (both fixed and random effects)

C-index from scratch

```
> pred_probs <- predict(mymodel3, type = "response")
> head(pred_probs)
        1         2         3         4         5         6
0.3462874 0.4837504 0.4837508 0.9653662 0.9934337 0.9996078

> library(Hmisc)
> somers2(pred_probs, as.numeric(tvdata$Form) - 1)
          C         Dxy           n     Missing
  0.9741682   0.9483365 900.0000000   0.0000000
```

When to use random effects?

- If the number of groups (individuals, texts from corpora) is greater than 5, it makes more sense to treat them as random effect. The estimation of their coefficients as fixed effects may be unreliable (too few observations per group).
- If the number of groups is 5 or less, it may be more useful to include them as fixed effects, especially if you are interested in their estimation (and p-values).

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

Assumption of linearity

- The model assumes that the relationships between the logit (log-odds) and the numeric predictors are linear.
- What are non-linear relationships like?
  - IQ and income
  - Age and height
  - IQ and popularity
  - Money and happiness

Generalized Additive Models

- An attractive method for testing and incorporating non-linearity
- Done with the help of smooths of different kinds and degrees of wiggliness
- Package mgcv, function bam() for large datasets and fast computation

GAM with R

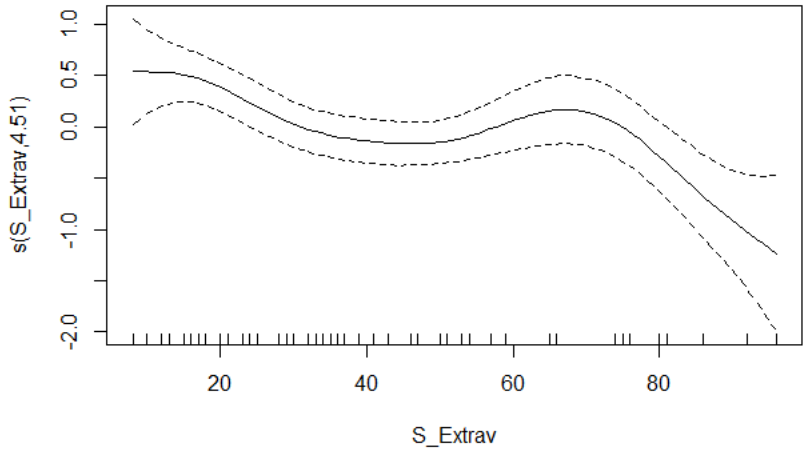
Let's test S_Extrav only:

```
> library(mgcv)
> gam1 <- bam(Form ~ s(S_Extrav), data = tvdata, family = binomial)
> plot(gam1)
> summary(gam1)
```

Smooth term in gam

…

```
Approximate significance of smooth terms:
                edf Ref.df Chi.sq  p-value
s(S_Extrav) 4.506  5.534   25.4 0.000214 ***
```



Some parameters of smooths

- k (knots) – how much wiggliness do we expect? 10 by default, e.g. s(X, k = 10). The larger k, the more wiggliness we expect.
- Gamma: another way to control wiggliness. The larger gamma, the less wiggliness there will be, as a rule. It can be useful when you have many smooths and want to specify it for the whole model. 1 by default, one may use 1.4 (recommended due to slight overfitting, or undersmoothing of GAMs), e.g. bam(… , gamma = 1.4).
- Different types of smooth bases (e.g. cubic regression splines, thin plate regression splines – the default), e.g. s(X, bs = “cr”) or s(X, bs = “tp”) – usually the choice doesn’t matter much, but cr may be faster to compute.

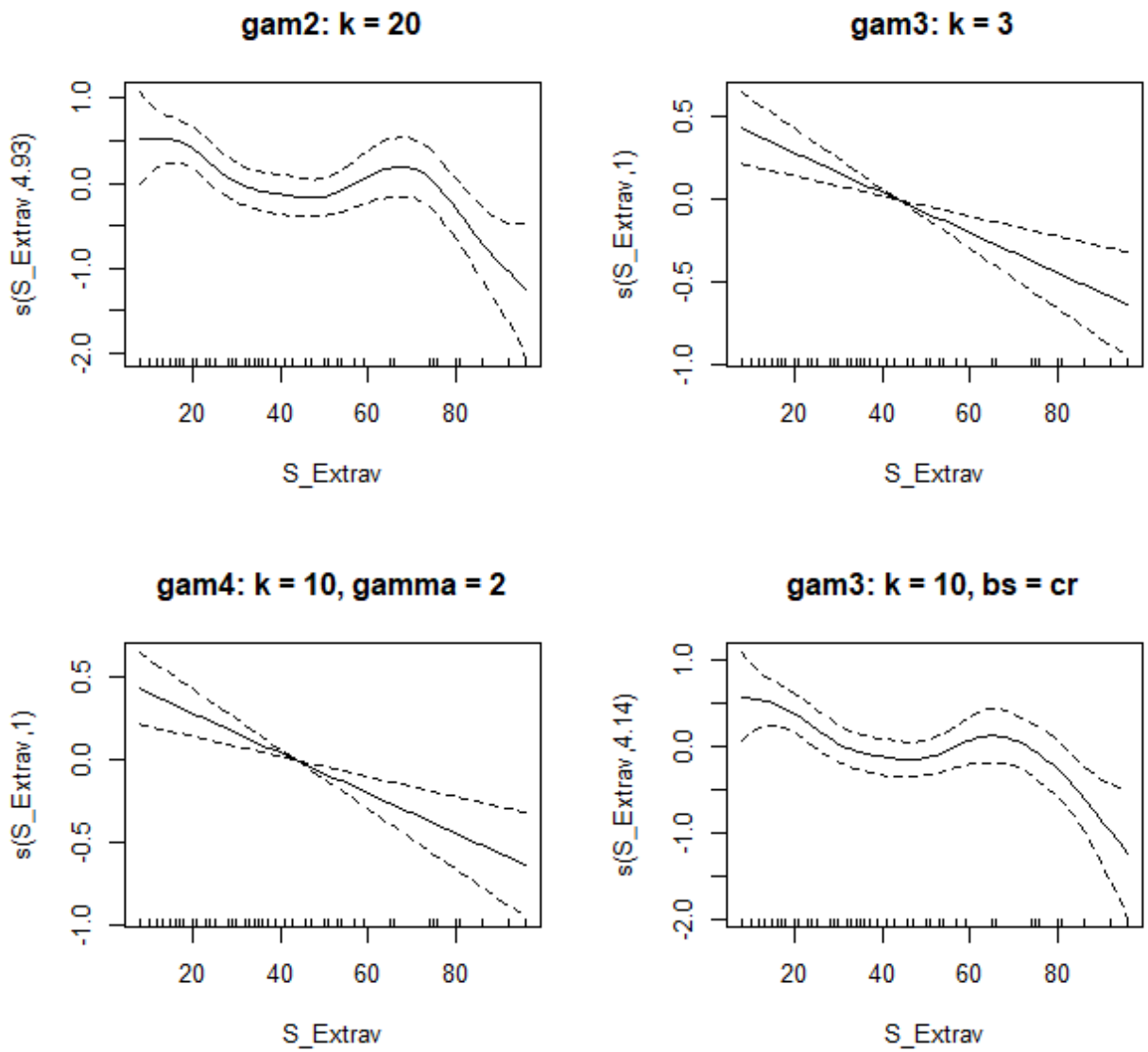
Try different options

```
> gam2 <- bam(Form ~ s(S_Extrav, k = 20), data = tvdata, family = binomial)

> gam3 <- bam(Form ~ s(S_Extrav, k = 3), data = tvdata, family = binomial)

> gam4 <- bam(Form ~ s(S_Extrav, k = 10), data = tvdata, family = binomial, gamma = 2)

> gam5 <- bam(Form ~ s(S_Extrav, k = 10, bs = "cr"), data = tvdata, family = binomial)
```


gam2: k = 20
s(S_Extrav,4.93)
S_Extrav
gam3: k = 3
s(S_Extrav,1)
S_Extrav
gam4: k = 10, gamma = 2
s(S_Extrav,1)
S_Extrav
gam3: k = 10, bs = cr
s(S_Extrav,4.14)
S_Extrav

How to decide?

```
> AIC(gam1, gam2, gam3, gam4, gam5)
            df      AIC
gam1 6.534205 1228.073
gam2 7.140967 1227.555
gam3 1.999750 1235.091
gam4 1.999762 1235.091
gam5 5.974887 1228.849
```

The oversmoothed versions with higher AIC are worse.

Exercise

- Perform similar analyses with S_Age. What is your conclusion?
- How to interpret the result theoretically?

Interactions between smooths and categorical variables

- For example, with H_Age:

```
> gam_int <- bam(Form ~ H_Age +
s(S_Extrav, by = H_Age), data = tvdata,
family = binomial)
```

```
> plot(gam_int)
```

Exercise

- Test if there is non-linearity in Extrav in the dataset catdog, alone and in the interaction with Gender.

Two univariate smooths

```
> gam6 <- bam(Form ~ s(S_Age, k = 20) +
s(S_Extrav, k = 20), data = tvdata, family =
binomial)
```

```
> summary(gam6)
```

…

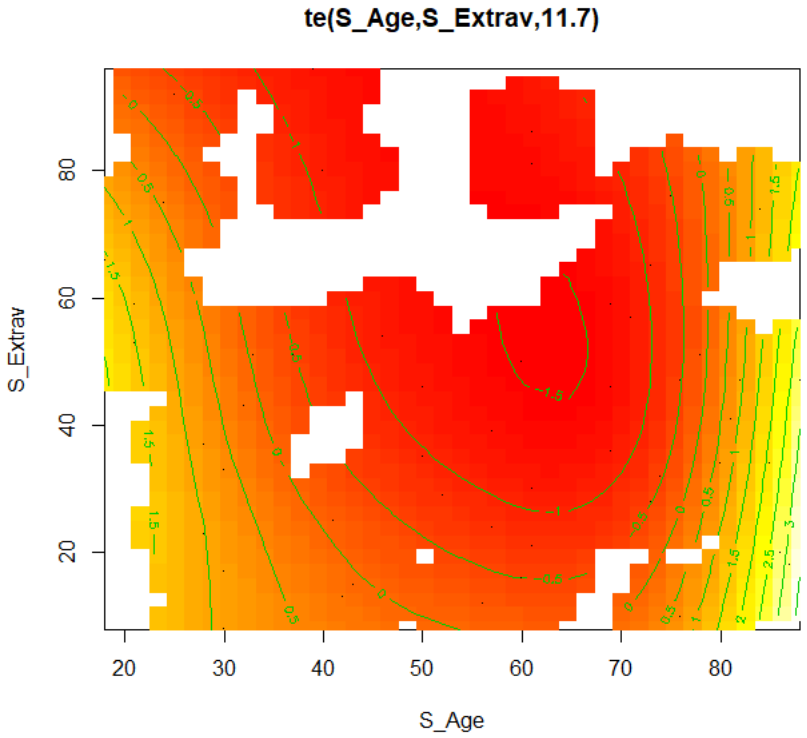
```
Approximate significance of smooth terms:
              edf Ref.df Chi.sq  p-value
s(S_Age)    5.695  7.029 146.56  < 2e-16 ***
s(S_Extrav) 2.538  3.154  27.73 4.52e-06 ***
```

Tensor product smooth

```
> gam7 <- bam(Form ~ te(S_Age, S_Extrav), data = tvdata, family = binomial)
> summary(gam7)
…
Approximate significance of smooth terms:
                        edf Ref.df Chi.sq p-value
te(S_Age,S_Extrav) 11.7  14.08  165.4  <2e-16 ***
```

Tensor product smooth

```
> plot(gam7, scheme = 2)
```



Univariate or bivariate?

- We can add an interaction term and use a LRT:

```
> gam8 <- bam(Form ~ s(S_Age, k = 20) +
s(S_Extrav, k = 20) + ti(S_Age, S_Extrav),
data = tvdata, family = binomial)
> anova(gam6, gam8, test = "Chisq")
Analysis of Deviance Table

Model 1: Form ~ s(S_Age, k = 20) + s(S_Extrav, k = 20)
Model 2: Form ~ s(S_Age, k = 20) + s(S_Extrav, k = 20) +
ti(S_Age, S_Extrav)
  Resid. Df Resid. Dev          Df   Deviance  Pr(>Chi)
1    886.87     1017.3
2    886.87     1017.3 5.5012e-05 2.0227e-05 0.0003004
***
```

ML vs. REML

- By default, the bam() algorithm uses fast Restricted Maximum Likelihood (fREML).
- For model comparison (fixed effects), it is recommended to use ML (here: Marginal Likelihood)
- To change that, refit the model, adding `method = "ML"` and perform the likelihood ratio test again.
- Usually, the differences are subtle. But the method may matter when the p-value is close to the cut-off point.

Some remarks

- You can also use a bivariate smooth s(X1, X2) – if the variables are on the same scale!
  - E.g. cm in width and height, e.g. when analysing a brain scan, latitude and longitude, when analysing distributions of biological organisms, etc.

Fixed-effect model

```
> gam9<- bam(Form ~ te(S_Age, S_Extrav) +
H_Age*Familiarity, data = tvdata, family =
binomial, gamma = 1.4)
```

```
> plot(gam9, scheme = 2)
```

Again, use visgam to visualize the interaction:

```
> visreg(gam9, xvar = "H_Age", by =
"Familiarity")
```

Random effects in GAM

- The most efficient way is to use a smoothing term s(ID_var, bs = “re”) for random intercepts, and by = X, bs = “re”) for slopes.
- This works, however, only if you have a relatively small number of subjects and a reasonable number of observations per subjects.
- Alternatively, use the following:
  - gamm(…, random = list(ID_var=~1)) for random intercepts
  - gamm(…, random = list(ID_var = ~ 1 + X)) for correlated intercepts and slopes
  - gamm(…, random = list(ID_var =~ 1, ID_var =~0 + X) for uncorrelated intercepts and slopes

Adding random intercepts

```
> gam10<- bam(Form ~ te(S_Age, S_Extrav) +
H_Age*Familiarity + s(S_ID, bs = "re"), data
= tvdata, family = binomial, gamma = 1.4)
```

```
> summary(gam10)
```

…

```
Approximate significance of smooth terms:
                       edf Ref.df Chi.sq  p-value
te(S_Age,S_Extrav) 12.25  14.11 152.72  < 2e-16 ***
s(S_ID)            10.12  46.00  19.24 0.000232 ***
```

Exercise

Test if it makes sense to add random intercepts for the survey questions.

Trying random slopes

```
> gam12<- bam(Form ~ te(S_Age, S_Extrav) +
H_Age*Familiarity + s(S_ID, bs = "re") + s(H_Age,
S_ID, bs = "re"), data = tvdata, family = binomial,
gamma = 1.4)
> summary(gam12)
…
Approximate significance of smooth terms:
                          edf Ref.df Chi.sq  p-value
te(S_Age,S_Extrav) 1.225e+01  14.11 152.72  < 2e-16
***
s(S_ID)            1.012e+01  46.00  19.24 0.000232
***
s(H_Age,S_ID)      9.896e-05 144.00   0.00 0.018246 *
```

Goodness of fit

- Adjusted pseudo R^2
- Explained deviance

Check the summary:

…

```
R-sq.(adj) =  0.705   Deviance explained = 66.9%
```

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
   - with fixed-effects
   - with mixed effects
   - Generalized Additive Models
   - Bayesian regression
3. More than two competitors: Multinomial logistic regression

Why Bayesian?

- You can the research hypothesis directly, instead of trying to reject the null hypothesis.
  - e.g. How confident can I be that the odds of V are greater when the addressee is a stranger than when the addressee is a friend?
  - Compare with the frequentist approach (what we've tested before): how likely is it that the odds of V when the addressee is a stranger are different from the odds of V when the addressee is a friend
- No *p*-values and hence no *p*-value hacking!
- Every result is interpretable and useful, not only the 'significant' ones. Good for scientific progress.

Types of probabilities

- p(x) is the probability of event x
  - The probability of getting the bullet when playing Russian roulette
  - The probability of a random person in this room being a linguist
- p(x, y) is the probability that events x and y will happen together
  - E.g. the probability that a random person in this room is a linguist and loves ice-cream
- p(x|y) is conditional probability of event x given event y, i.e. that event x will happen if y happens
  - E.g. The probability of finding an ice-cream fan if one picks a linguist.
  - Can be computed as $p(x|y) = p(x,y)/p(y)$

Bayes' rule

$$P(y|x) = \frac{P(x|y)\,P(y)}{P(x)}$$

The mysterious
Thomas Bayes (1702 – 1761)

Why would you care?

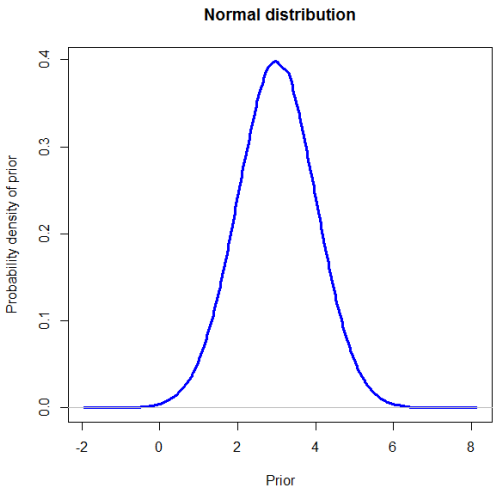
p(beliefs|data) = p(data|beliefs) p(beliefs)/p(data)

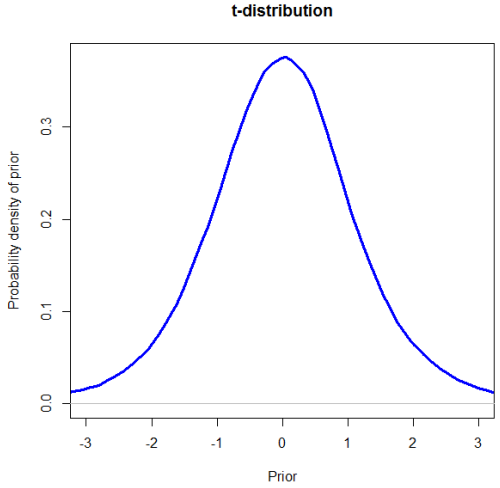
- p(beliefs) is called prior probability, or just prior
- p(beliefs|data) is called posterior probability, or posterior
- p(data|beliefs) is called likelihood
- p(data), aka evidence, or prior predictive, or marginal likelihood

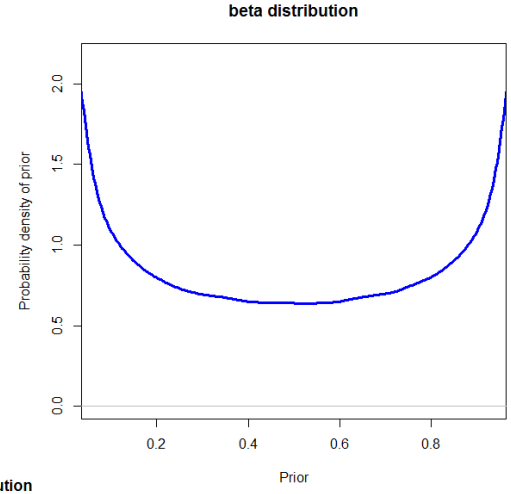
The posteriors express directly the probability of our beliefs given the data!

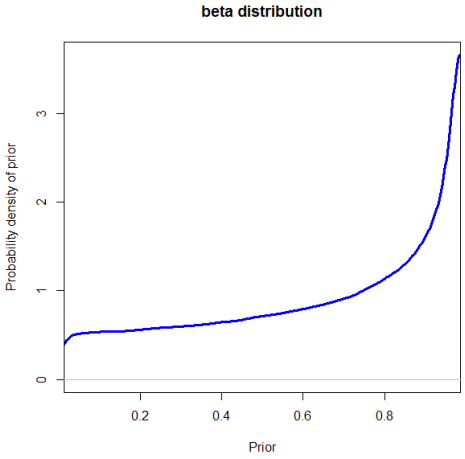
A challenge

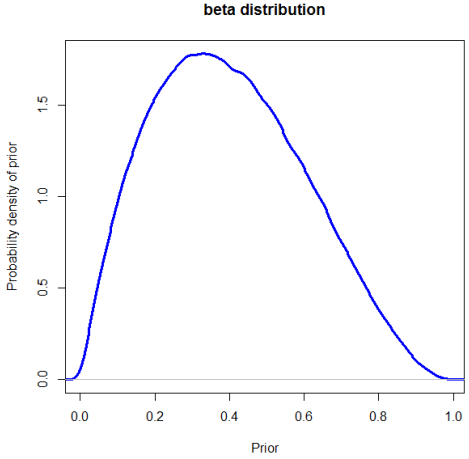
- Beliefs are usually expressed as prob. distributions:











Markov Chain Monte Carlo

- This means that the posterior is also a probability distribution. But how are we supposed to know what it looks like?!
- One can get the posterior distribution by sampling a large number of representative points from the posterior with the help of a Markov Chain Monte Carlo algorithm
  - Monte Carlo simulation: any simulation that draws random values from a distribution, e.g. `rnorm()`, `rt()` in R.
  - Markov Chain process: a random walk when the next step does not depend on the steps before the current position.

Metropolis algorithm

- President X of country Y (no names!) wants to sell weapons in order to promote peace in the world.
- Goes to a rich country Z to negotiate the deals with several sheikhs.
- Some of them have more money, some have less.
- Obviously, the more money, the more X should be interested!
- But X doesn’t know how much money each has (he doesn’t know much, in general).

The Sheikhs’palaces

This consumption is very conspicuous: the more money, the more splendid the palace.

10bn

20bn

30bn

40bn

50bn

60bn

The random walk

- Imagine the president arrives first in Sheikh 3's palace. Let's call him the Current Sheikh (CS). Flips a coin if he should go to the sheikh on the left or to the one on the right. The Sheikh selected in this process is called the Proposed Sheikh (PS).
- X doesn't know how much money each sheikh has, but he is not totally stupid. He can look from the window of the CS's palace and compare the sizes of the palaces, computing the ratio PS/CS with the help of a top secret supercomputer.
- If the PS's palace is larger and more lavish, PS/CS > 1 and the president goes to the PS.

The random walk (continued)

- If the PS's palace is smaller and more modest, 0 < PS/CS < 1, then X generates randomly the probability from 0 to 1 using a top secret random probability generator.
- If the randomly generated probability is less than the PS/CS ratio, then he moves to the PS. If the probability is greater, then he stays, and the steps are repeated again.
- In the long run, the frequency of stays at each palace will approximate the sheikhs' relative wealth!

Some implications for Bayesians

- If some value (e.g. the wealth of the current sheikh) is very large, the algorithm may get stuck at it and not traverse the space quickly enough. One should use diagnostic plots for that purpose.
- The results of the first walks are usually excluded (aka burn-in, or warm-up period) because the initial initial position can introduce substantial bias.
- It’s recommended to have several Markov chains and check if they behave similarly.

Why Bayesian regression?

- Bayesian regression can be used with tricky data, where frequentist regression encounters problems and one needs complex solutions:
  - small samples
  - multicollinearity
  - outliers and influential observations
  - complete and quasi-complete separation
  - complex random effects structure in mixed models
  - etc.

Bayesian regression in R: some examples

- brms (a wrapper for Stan)
- rstan (for advanced, requires programming in Stan)
- MCMCglmm (logistic regression was a bit tricky, in my experience)
- arm
- blme

Simple logistic regression: R code

```
> library(brms)
> mybrm1 <- brm(Form ~ Familiarity, data = tvdata, chains = 2, iter = 500, warmup = 200, family = bernoulli)
```

Important! Use bernoulli instead of binomial.

Markov chains and burn-in

```
chains = 2 #to speed up the things a little bit, 4 Markov chains by default

iter = 500 #number of iterations in the random walk, 2000 by default. Again, we just want to speed the things up.

warmup = 200 #default first half, i.e. 250 with 500 iterations
```

brm summary (fragment)

```
> summary(mybrm)
Population-Level Effects:
                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
Intercept           -1.75      0.18    -2.11    -1.43        284    1
FamiliarityFar       2.71      0.21     2.28     3.14        345    1
FamiliarityMiddle    2.19      0.21     1.80     2.58        342    1
```

Interpreting the numbers

- Estimate: the mean of the posterior distribution
- Est.error: standard error of the posterior distribution
- l-95% CI: the lower boundary of the 95% credible interval
- u-95% CI: the upper boundary of the 95% credible interval

Posterior distribution

```
> ps_beta <- posterior_samples(mybrm1, pars = "Familiarity")
> dim(ps_beta)
[1] 600   2
> summary(ps_beta)
b_FamiliarityFar b_FamiliarityMiddle
 Min.   :2.158    Min.   :1.626
 1st Qu.:2.568    1st Qu.:2.039
 Median :2.717    Median :2.177
 Mean   :2.713    Mean   :2.188
 3rd Qu.:2.861    3rd Qu.:2.338
 Max.   :3.382    Max.   :2.861
```

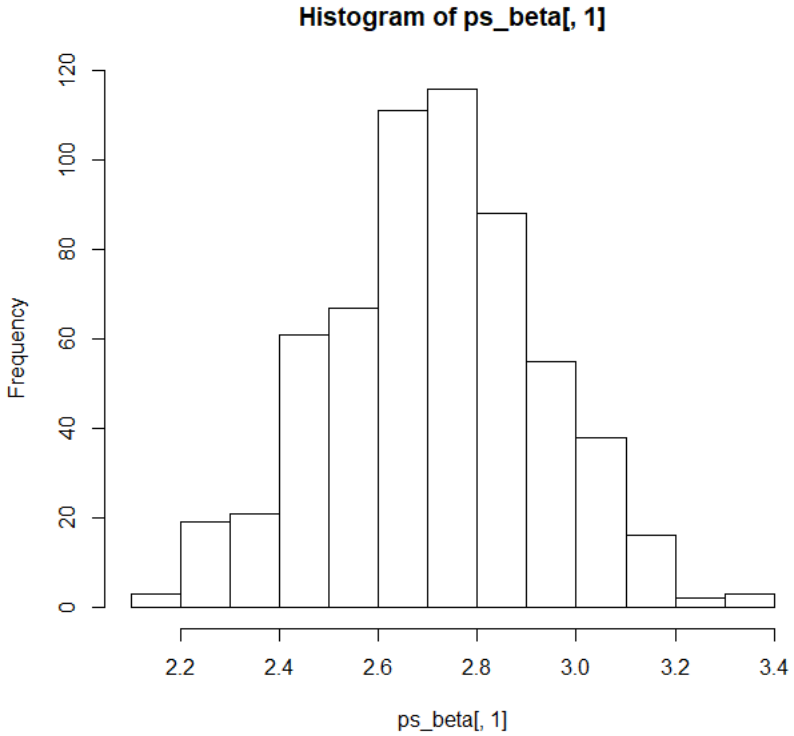
compare with the table!

```
> sd(ps_beta[, 1])
[1] 0.2149344
```

compare with the table!

Posterior distribution

```
> hist(ps_beta[, 1])
```



Mean or median posteriors

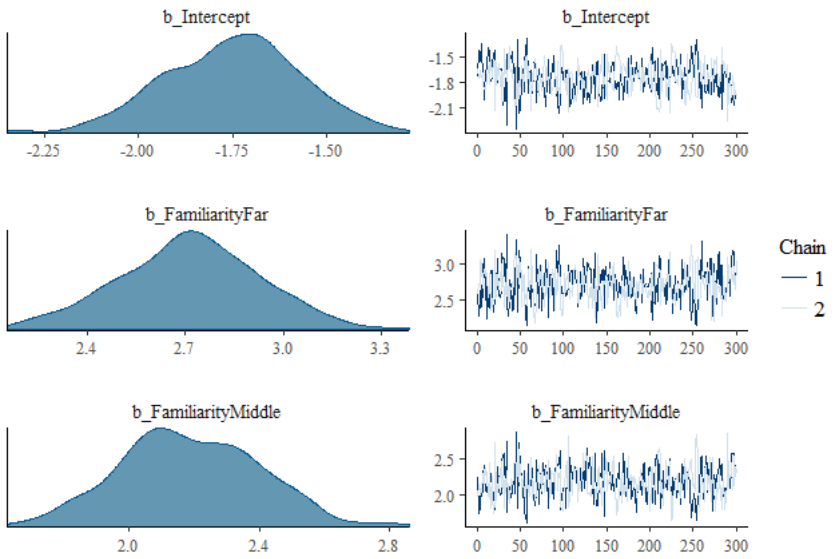
```
> fixef(mybrm) #by default, the mean
                       mean
Intercept          -1.750527
FamiliarityFar      2.713482
FamiliarityMiddle   2.187722
> fixef(mybrm, estimate = "median")
                     median
Intercept          -1.741829
FamiliarityFar      2.716772
FamiliarityMiddle   2.176549
```

Density and trace plots

```
> plot(mybrm)
```

The density plots should not be bimodal (with two humps).

The trace plots should look like fat hairy caterpillars, not bending anywhere.



Effective sample size and r-hat

- Effective sample size: the number of posteriors in Markov chains discounted for autocorrelation between them (when the chain gets stuck). The greater effective sample size, the more reliable the results.
- R-hat metric: the ratio of the between-chain and the within-chain variability of posteriors. If the chains have converged, these measures will be similar; otherwise, the between-chain variability will be larger. R-hat should be 1.

Fitted values

```
> brm_fit <- fitted(mybrm1) #gets the fitted values for each observation (i.e. the subject)
> head(brm_fit) #the first six observations
  Estimate  Est.Error   2.5%ile  97.5%ile
1 0.1493673 0.02229118 0.1083592 0.1934067
2 0.1493673 0.02229118 0.1083592 0.1934067
3 0.1493673 0.02229118 0.1083592 0.1934067
4 0.6072164 0.02859605 0.5543798 0.6623144
5 0.6072164 0.02859605 0.5543798 0.6623144
6 0.6072164 0.02859605 0.5543798 0.6623144
```

Goodness of fit

- To check how well the fitted values correspond to the observed forms, we can use the *C*-index.
- You already know how to compute it!

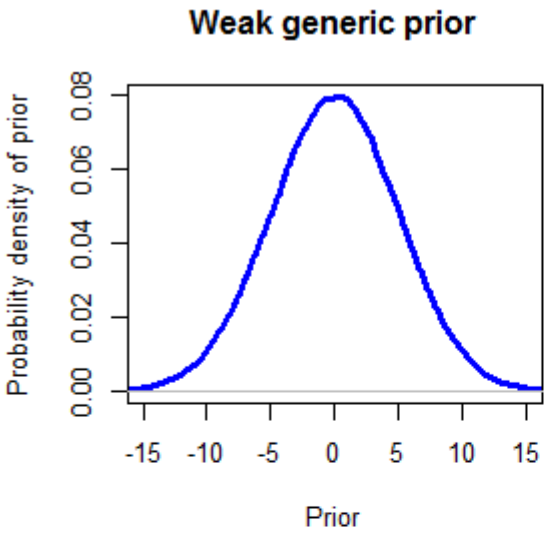
Measures for model comparison

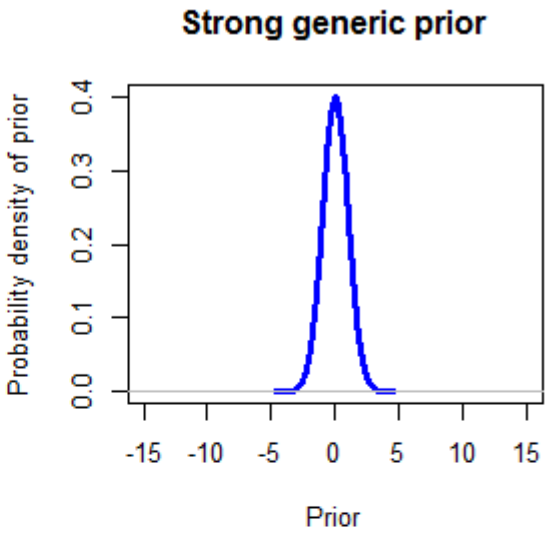
- In frequentist regression, one often uses AIC for model comparison. It favours parsimonious models.
- Its Bayesian 'colleagues' are Widely Applicable Information Criterion and Leave-One-Out Information Criterion. The smaller WAIC & LOOIC, the better the model. These criteria should only be applied to models based on the same data!

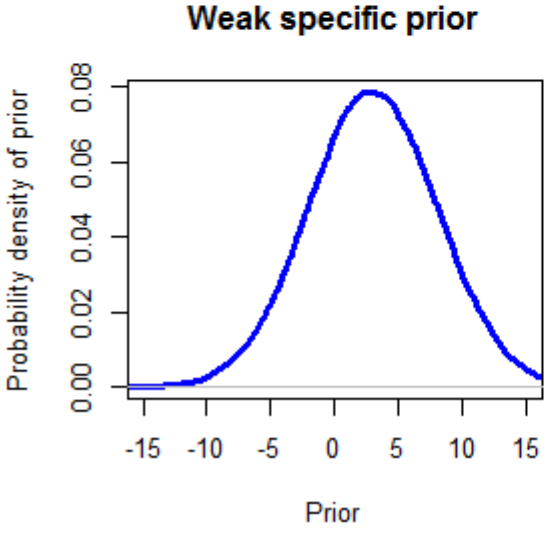
```
> WAIC(mybrm1) #Widely Applicable IC
   WAIC     SE
 1016.06 28.17
> LOO(mybrm1) #Leave-one-out IC
   LOOIC     SE
 1016.05 28.17
```

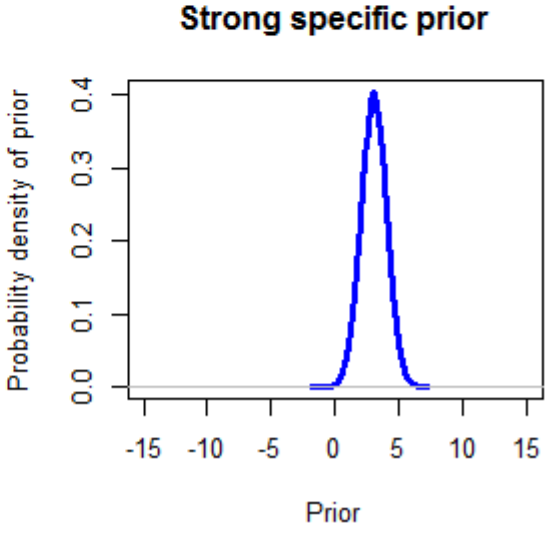
Types of prior distributions

- Flat, or uniform non-informative priors have no impact on the posteriors. All values have the same likelihood. The results are nearly identical to the ones obtained with the help of frequentist methods.
- Informative priors have impact on the posteriors:
  - weak vs. strong (e.g. normal distribution with sd = 10 vs. normal distribution with sd = 1)
  - generic vs. specific (e.g. normal distribution with sd = 5 and mean = 0 vs. normal distribution with sd = 5 and mean = 3).


Weak generic prior
Probability density of prior
0.00
0.02
0.04
0.06
0.08
-15
-10
-5
0
5
10
15
Prior


Strong generic prior
Probability density of prior
0.0
0.1
0.2
0.3
0.4
-15
-10
-5
0
5
10
15
Prior


Weak specific prior
Probability density of prior
0.00
0.02
0.04
0.06
0.08
-15
-10
-5
0
5
10
15
Prior


Strong specific prior
Probability density of prior
0.0
0.1
0.2
0.3
0.4
-15
-10
-5
0
5
10
15
Prior

Effect of priors

- By default, the priors are weak and generic. When such priors are used, the results are very similar to the ones from the corresponding frequentist model:

```
> fixef(mybrm1) #Bayesian
                    mean
Intercept          -1.750527
FamiliarityFar      2.713482
FamiliarityMiddle   2.187722
```

```
> coef(glm(Form ~ Familiarity, data = tvdata,
family = binomial)) #frequentist
 (Intercept)    FamiliarityFar    FamiliarityMiddle
 -1.734601          2.695658          2.167923
```

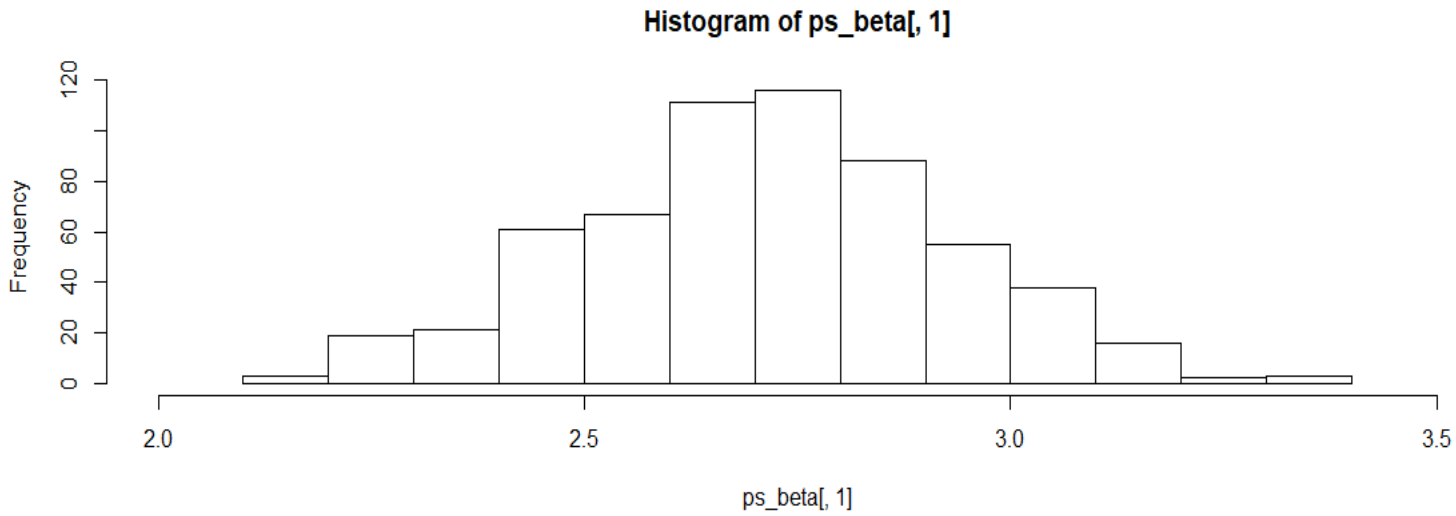
Small informative priors

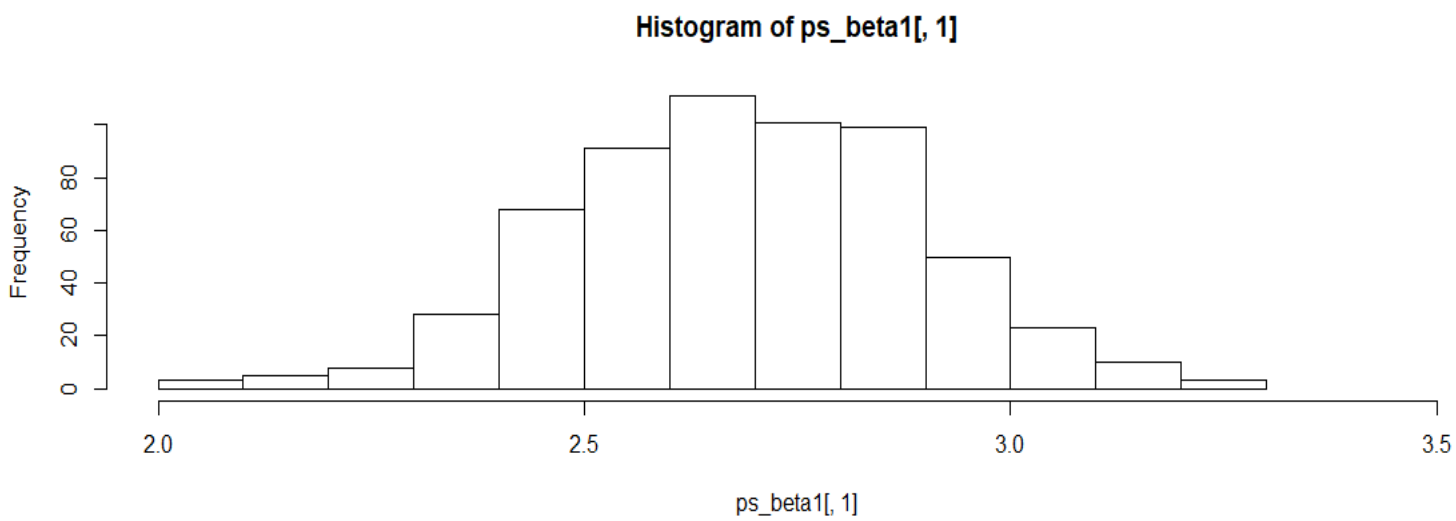
- Purely hypothetically, imagine that there is some previous research that suggests that the estimate of Familiarity = Far is -2 (i.e. people use T to talk to strangers more often than to friends)
- You want to use this information as your priors.

Adding new priors

```
> mybrm1 <- brm(Form ~ Familiarity, data = tvdata, chains = 2, iter = 500, warmup = 200, family = bernoulli, prior = prior(normal(-2, 1), class = b))
```

- Compare the old and new posterior distributions. What is the difference?

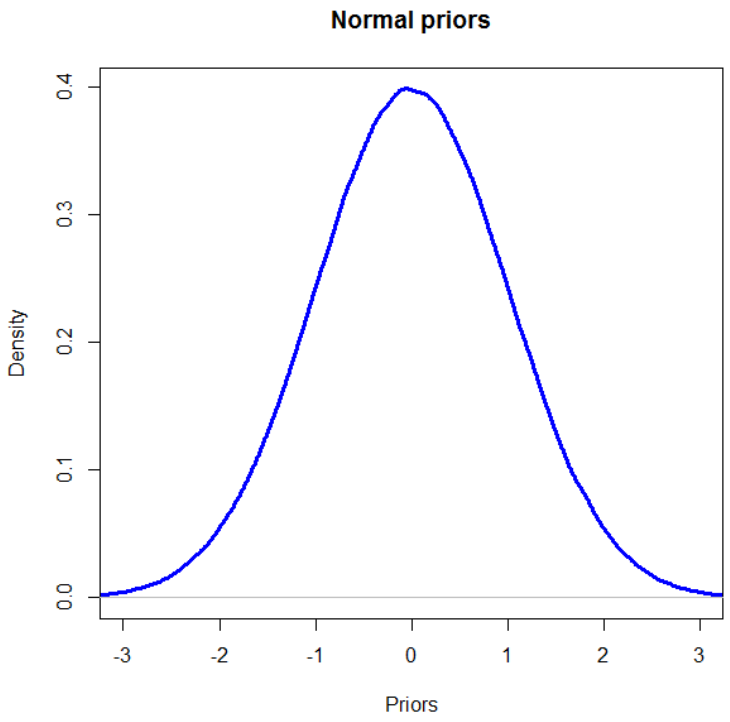

Histogram of ps_beta[, 1]
Frequency
0
20
40
60
80
120
2.0
2.5
3.0
3.5
ps_beta[, 1]

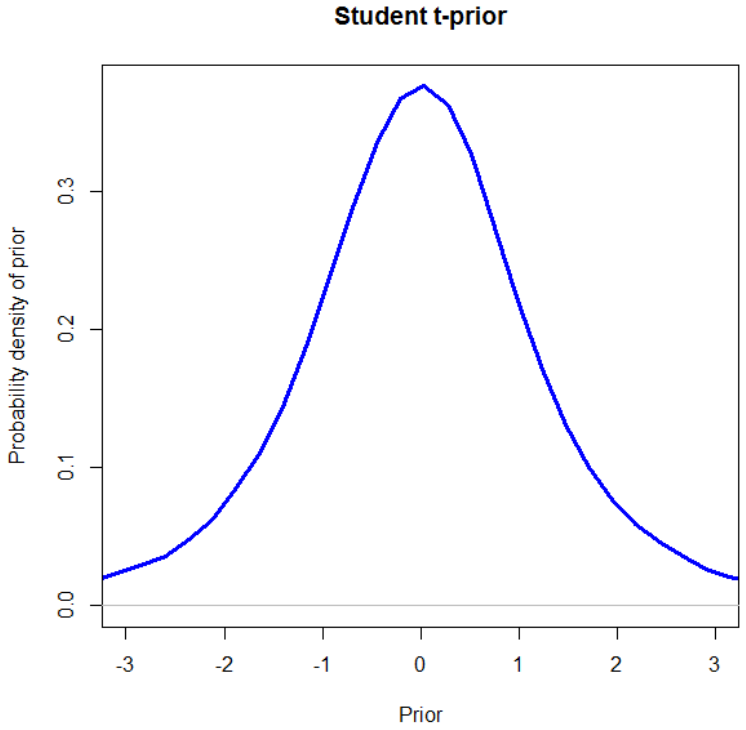

Histogram of ps_beta1[, 1]
Frequency
0
20
40
60
80
2.0
2.5
3.0
3.5
ps_beta1[, 1]

Popular informative priors

- normal(mean, scale)
- student_t(df, mean, scale)
- cauchy(mean, scale)
- See examples in `?set_prior` and `?brm`

Watch the tails





Student t-priors allow for outliers.

Tasks

1. Make a subset of data (take only 100 observations).

```
> data_sample <- tvdata[sample(900, 100, replace = F), ]
```

Fit a model with the same informative priors as above. Compare the posteriors with the posteriors of mybrm1.

2. Use the same subset and fit a model with the default priors. Compare the posteriors with the posteriors of mybrm.

3. Try different burn-in periods (warmup). Compare the results (histograms and trace plots).

4. Try different number of iterations (iter). Compare the results (histograms and trace plots).

What are your conclusions?

Testing your hypothesis directly

```
> mean(ps_beta[, 1] > 0)
[1] 1
```

- That is, 100% of the posterior distribution is positive. In other words, our confidence that there is an increase in the chances of V when one addresses a stranger, given the data, is 100%.
- If you still want to do pointwise NHST (i.e. show that the effect is not zero), you can use the 95% credible intervals for for hypothesis testing (don't mix with 95% confidence intervals in frequentist statistics!). If they do not include zero, then we can be 95% sure that the effect is different from zero.
- So, you can both say how much your hypothesis is supported by the data and do NHST, all in one.

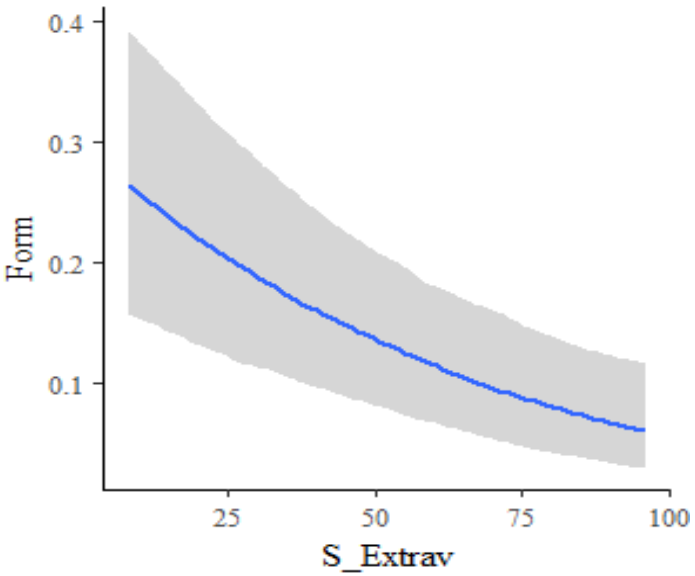
Exercise: Multivariate model

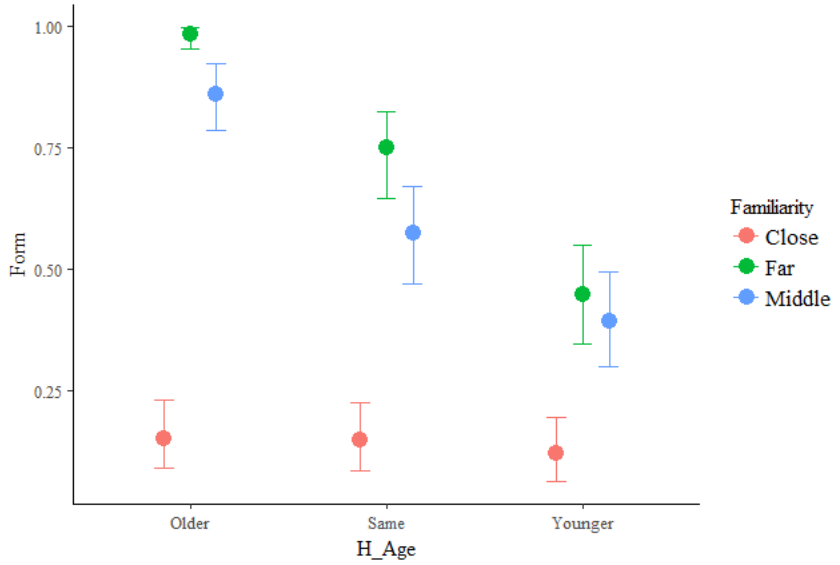
- Fit a Bayesian model with the fixed effects. Use the default flat priors and the same settings as in the first model. You should only change the formula.
- Does the model fit the data better than the first model with only one predictor? Compare the WAIC, LOO and R^2 scores. Is the new model better?

Visualization of effects

```
> marginal_effects(mybrm5)
```

Selected plots:





Enriching the model

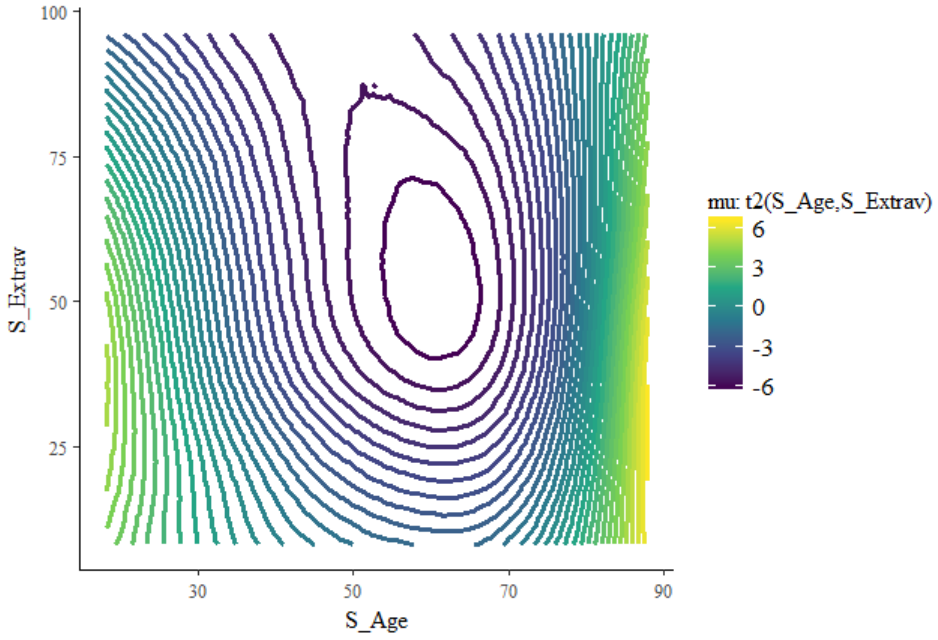
- Random effects are specified as in lme4.
- For smooths, one can currently use either s() or t2():

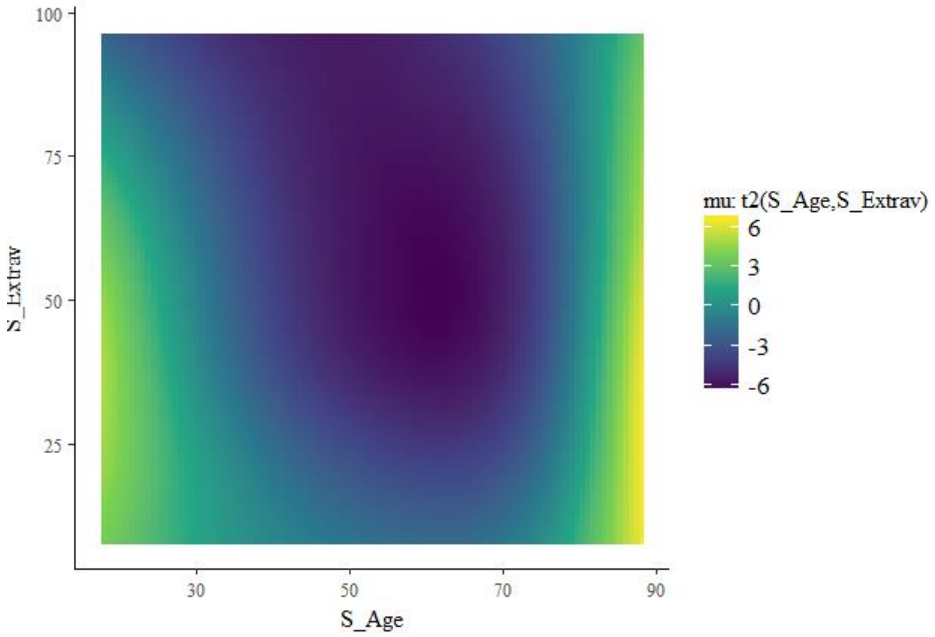
```
> mybrm6 <- brm(Form ~ t2(S_Age, S_Extrav)
+ H_Age*Familiarity + (1|S_ID), data =
tvdata, family = bernoulli, chains = 2, iter
= 1000, warmup =  200)
```

Plotting the smooths

```
> ms <- marginal_smooths(mybrm6) #to save time, because it's computationally intensive
> plot(ms, stype = "contour")
> plot(ms, stype = "raster")
```

Marginal smooth plots





Adjustments per subject

```
> ranef(mybrm6)
$S_ID
     Intercept
1   0.67283479
2   0.05976521
3  -0.03645735
4   0.07923777
5   0.16867203
…
```

Priors for fixed and random effects

- for fixed effects:
  - `prior(…, class = b)`
- for random effects:
  - `prior(…, class = sd)`
- for the intercept:
  - `prior(…, class = Intercept)`
- for correlations between intercepts and slopes:
  - `prior(lkj(eta = 2), class = cor)` #the larger eta, the less likely extreme correlations (-1 or +1).

Big exercise

- Use the data from the survey and the techniques from the course to investigate how you and your colleagues use politeness forms.

Course outline

1. Basic concepts of regression analysis
2. Two rivals: Binomial logistic regression
    - with fixed-effects
    - with mixed effects
    - Generalized Additive Models
    - Bayesian regression
3. More than two competitors: Multinomial logistic regression

More than two possible outcomes

Multinomial model

- The most common approach: the reference level is compared with each of the other levels.
- Why not fit two or more separate binomial logistic models?
  - This solution is less efficient, leading to higher standard errors.
  - But the results are likely to be very similar, especially if the reference level is the most frequent category.

Multinomial Bayesian model in brms

```
> mymulti <- brm(Name ~ S_Extrav + H_Age*Familiarity, data = tvdata, family = categorical, chains = 2, iter = 500, warmup = 200)
```

All other effects (interactions, random, smooths) can be tested, as well, as shown above.

Warnings:

- this is a very poor model (more chains and iterations are needed), but we don't have time right now… Please try this at home.
- So far, there are no visualization tools for the effects.

Two sets of estimates

```
> summary(mymulti)
```

…

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI
First_Intercept	0.33	0.28	-0.22	0.88
Full_Intercept	-0.92	0.57	-2.19	0.12
First_S_Extrav	-0.03	0.00	-0.04	-0.02
First_H_AgeSame	-0.06	0.32	-0.69	0.54
First_H_AgeYounger	-0.17	0.33	-0.86	0.45

Snow in English

- Sposh (soft slushy mud or snow)
- Blizzard (a long severe snowstorm)
- Onding (a heavy fall or rain or snow)
- Skift (a light fall of snow or rain)
- Graupel (granular snow pellets)
- Sleet (frozen or partly frozen rain)
- Névé (a snowfield at the head of the glacier)
- Grue (thin floating ice)
- Corn snow (granular snow formed by alternate thawing and freezing, dry and crunchy)

https://www.merriam-webster.com/words-at-play/10-more-words-for-snow/onding

When to use?

- Modelling more than 5 categories already becomes cumbersome. Also, one needs a lot of data. You can use classification methods instead (e.g. Conditional Inference Trees and Random Forests).