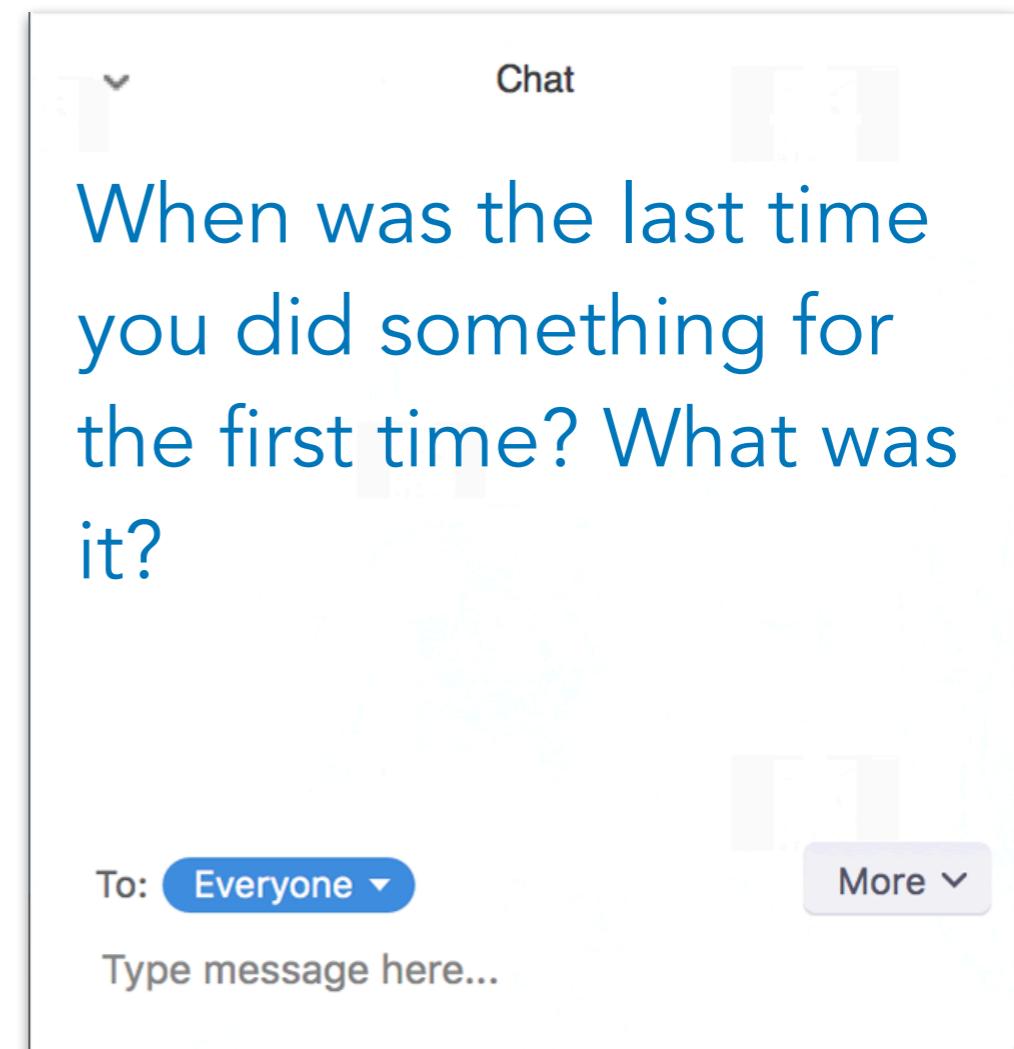
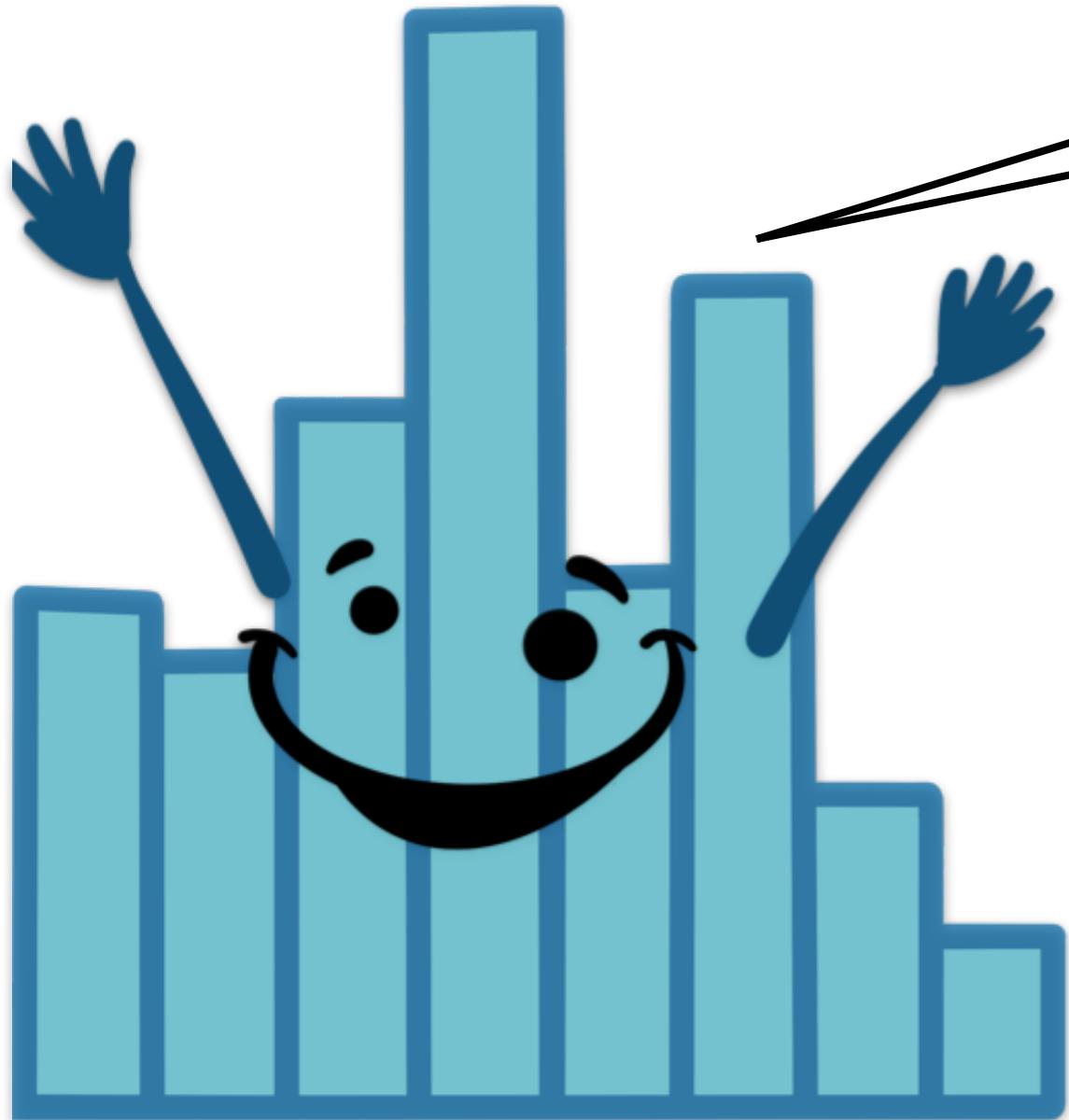


Linear mixed effects models 3

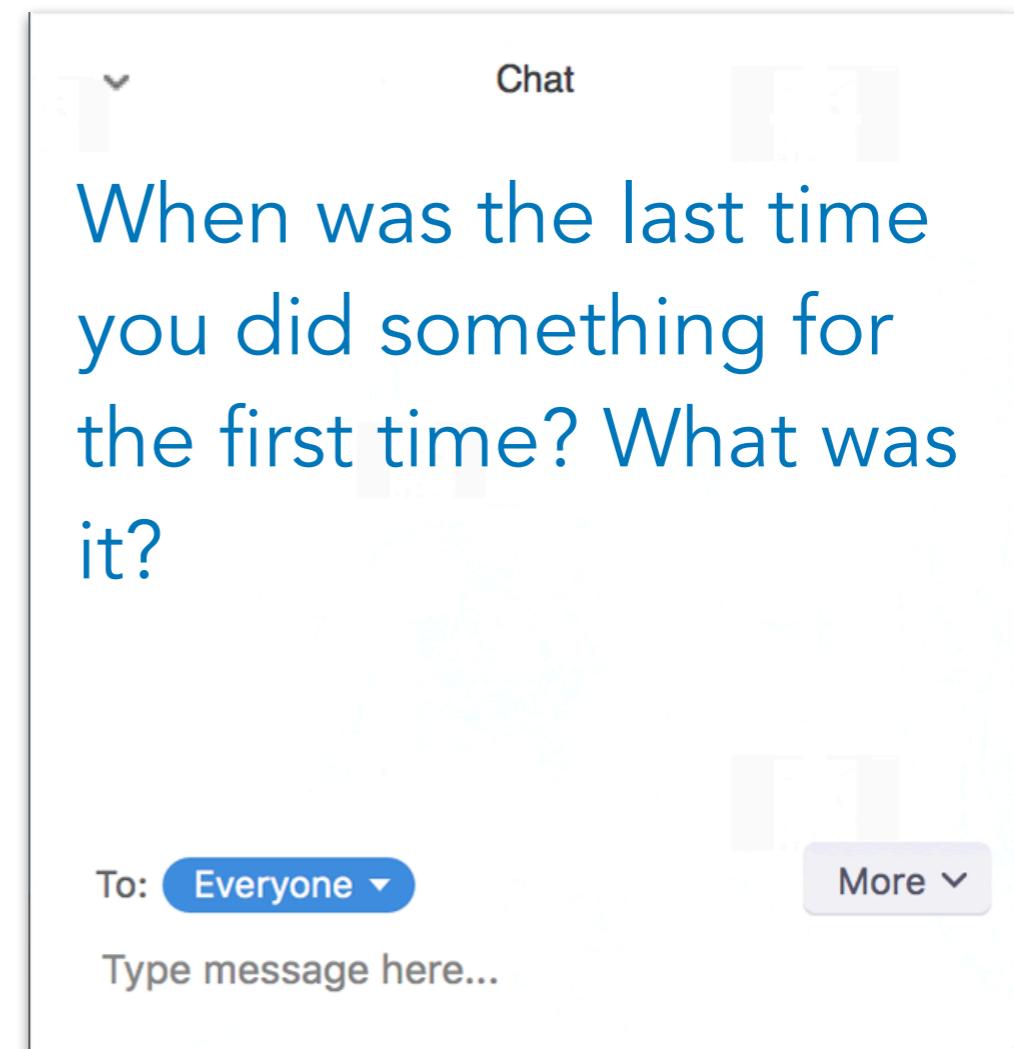


01/01/2021

Remember to
record the
lecture!



Linear mixed effects models 3



01/01/2021

Plan for today

- Quick LMM recap
- Understanding `lmer()` syntax
- Pitfalls in fitting `lmer()`s (and what to do about it)
- Generalized linear model
 - Logistic regression
 - interpreting the model output
 - fitting and reporting models
 - mixed effects logistic regression

Things that came up



Brenton Wiernik @bmwiernik

...

A student in my R programming class shared this and I
can't stop laughing about it

```
sandwich %>%  
  pivot_longer()
```



8:34 AM · Mar 1, 2021 · Twitter for iPhone

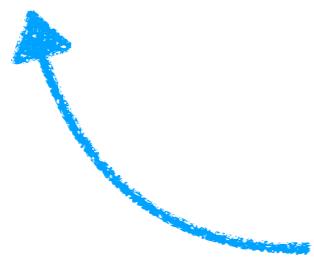
273 Retweets 29 Quote Tweets 2,012 Likes

Feedback

Feedback

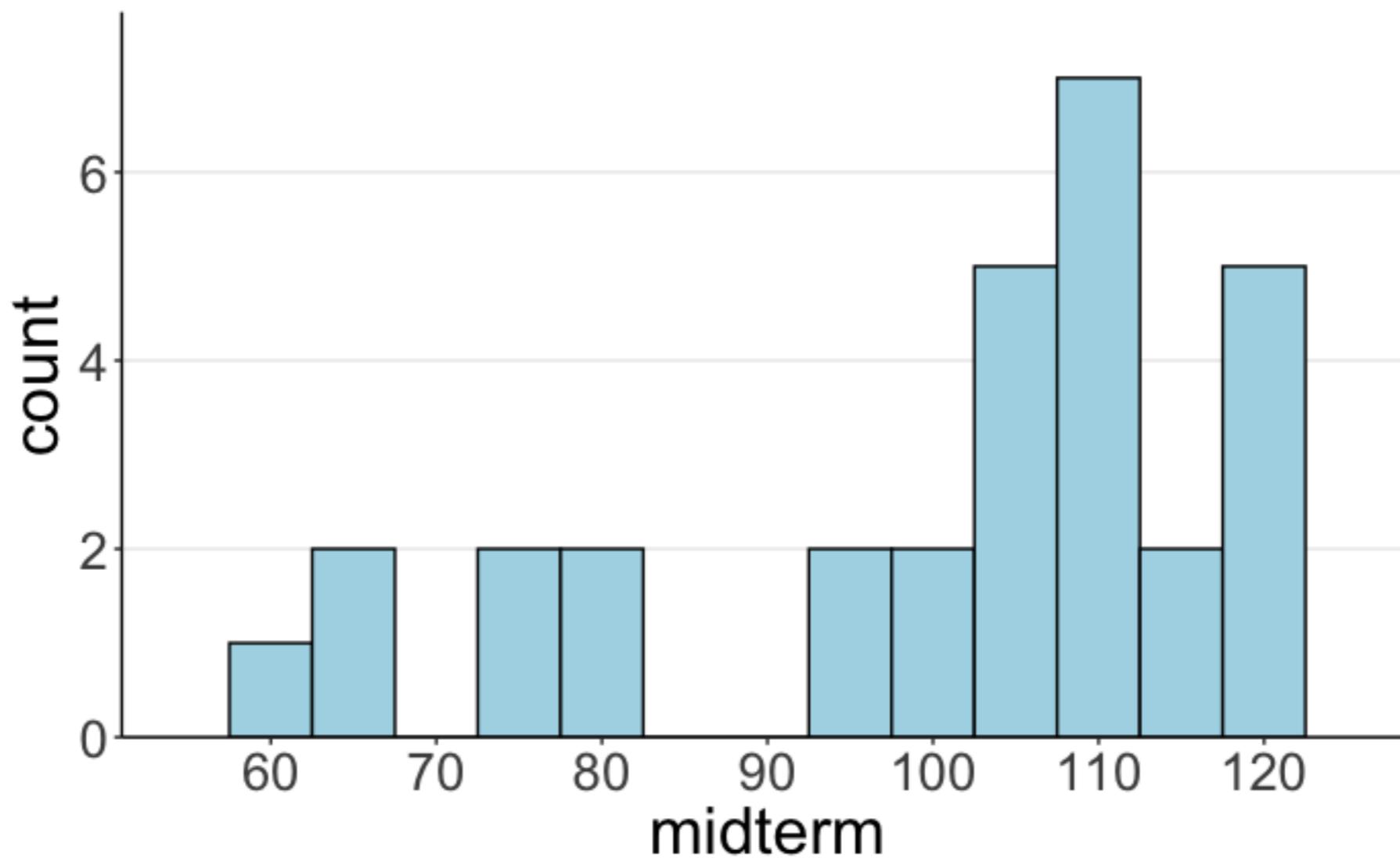
I think the pacing and everything about the class is great! I just wanted to submit a bit of feedback about the structure of lecture. I find that **sometimes lecture is very 'bottom up'** --- usually, they'd start with a lot of examples and build-up, and then in the ***very end***, there's an 'aha!' with the key definitions. A good example of this is the Mediation and Moderation lecture; there was a ton of build-up, but the definition of mediation / moderation didn't really come until the second half, or even the very end. **I think that it might be better to have a more 'top-down' model, where basically key terms are defined from the top, but referenced again when there is more context.** I feel like I'm lost, sometimes, just listening to the winding story, but don't have enough fenceposts guiding me in the lecture.

<https://tinyurl.com/psych252survey21>



so that students who can't be here synchronously can give feedback, too

Midterm



midterm only counts for 20% of the final grade
homeworks = 40%, final project = 40%

**you can still get a very good grade even you didn't do
as well as you had hoped on the midterm**

Quick LMM recap

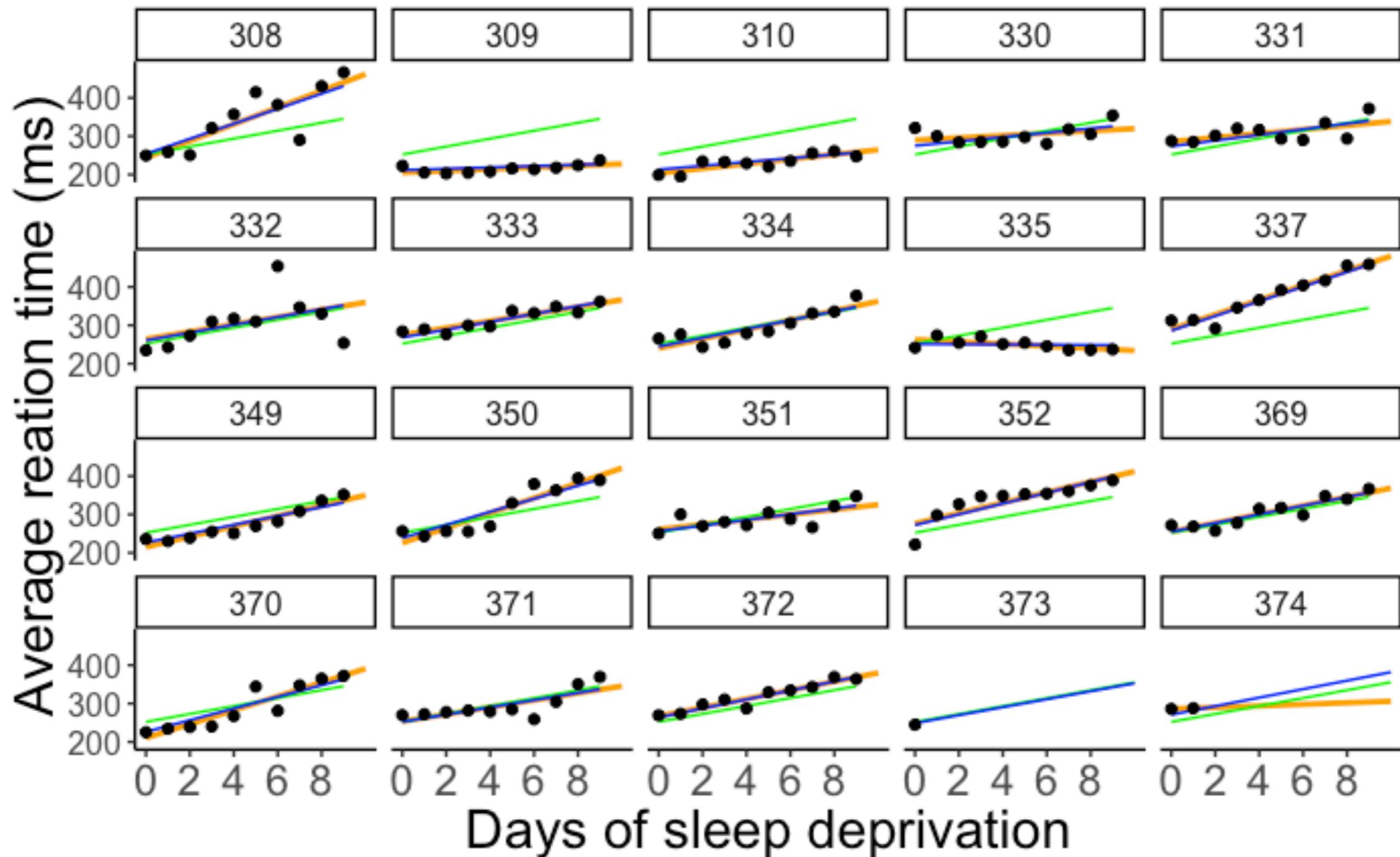
general points about `lmer()`



- **fixed effects:**
 - often: factors that we manipulate experimentally
 - parameters are estimated --> we are interested in characterizing the relationship between this variable and the outcome
- **random effects:**
 - variation we want to control for
 - often: differences between participants or items in our experiment

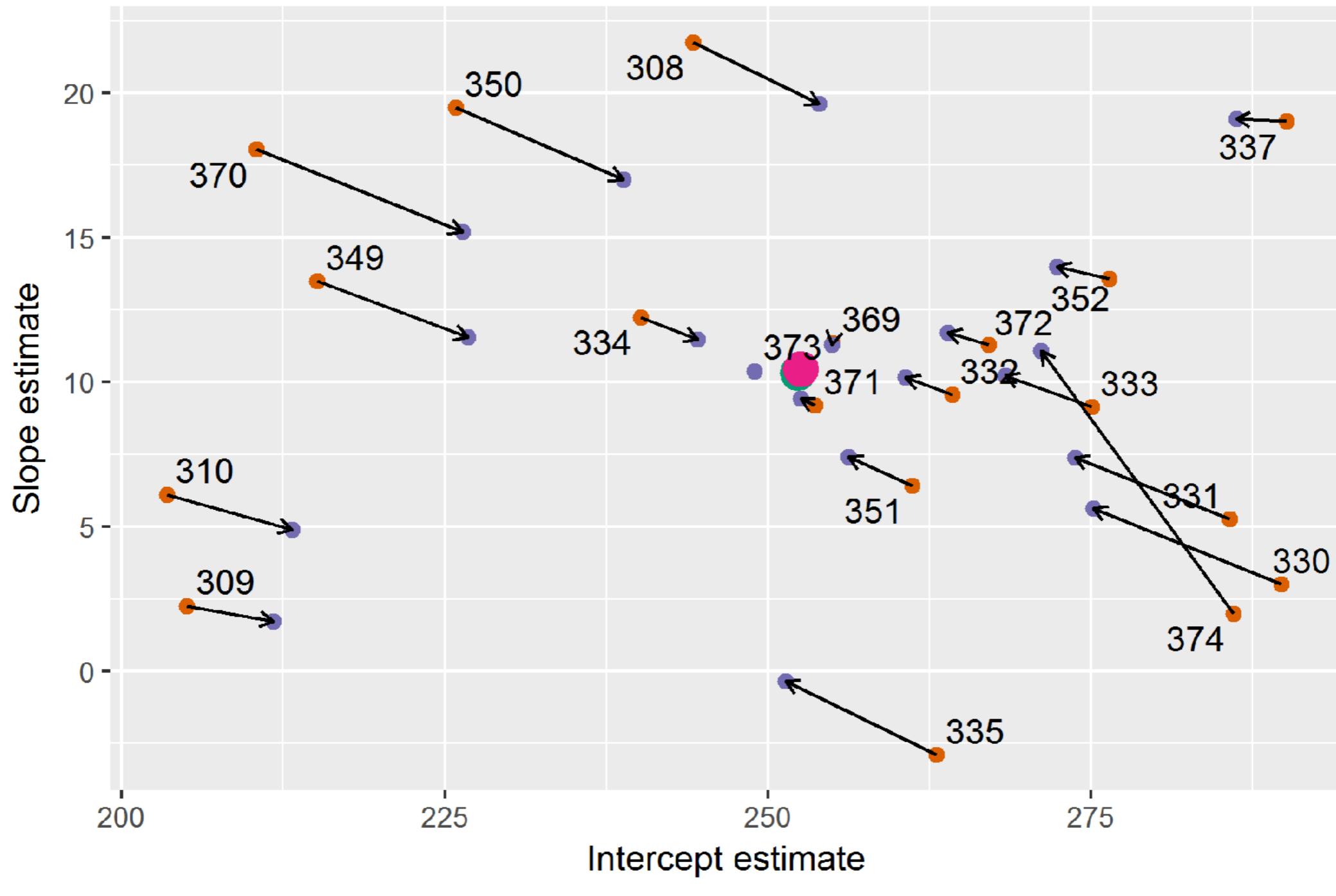
Comparison

complete pooling
no pooling
partial pooling



Shrinkage

Pooling of regression parameters



Model Complete pooling No pooling Partial pooling Partial pooling (average)

Understanding lmer() syntax

`lmer()` syntax summary

formula	description
<code>dv ~ x1 + (1 g)</code>	Random intercept for each level of `g`
<code>dv ~ x1 + (0 + x1 g)</code>	Random slope for each level of `g`
<code>dv ~ x1 + (x1 g)</code>	Correlated random slope and intercept for each level of `g`
<code>dv ~ x1 + (x1 g)</code>	Uncorrelated random slope and intercept for each level of `g`
<code>dv ~ x1 + (1 part) + (1 item)</code>	Random intercept for each level of `participant` and for each level of `item` (crossed)
<code>dv ~ x1 + (1 school/class)</code>	Random intercept for each level of `school` and for each level of `class` in `school` (nested)

Coefficients

```
lmer (formula = reaction ~ 1 + days +  
      data = df.sleep)
```

$(1 \mid \text{subject})$

random intercepts

\$subject	(Intercept)	days
308	292.2749	10.43191
309	174.0559	10.43191
310	188.7454	10.43191
330	256.0247	10.43191
331	261.8141	10.43191
332	259.8262	10.43191
333	268.0765	10.43191
334	248.6471	10.43191
335	206.5096	10.43191
337	323.5643	10.43191
349	230.5114	10.43191
350	265.6957	10.43191
351	243.7988	10.43191
352	287.8850	10.43191
369	258.6454	10.43191
370	245.2931	10.43191
371	248.3508	10.43191
372	269.6861	10.43191
373	248.2086	10.43191
374	273.9400	10.43191

$(0 + \text{days} \mid \text{subject})$

random slopes

\$subject	(Intercept)	days
308	252.2965	19.9526801
309	252.2965	-4.3719650
310	252.2965	-0.9574726
330	252.2965	8.9909957
331	252.2965	10.5394285
332	252.2965	11.3994289
333	252.2965	12.6074020
334	252.2965	10.3413879
335	252.2965	-0.5722073
337	252.2965	24.2246485
349	252.2965	7.7702676
350	252.2965	15.0661415
351	252.2965	7.9675415
352	252.2965	17.0002999
369	252.2965	11.6982767
370	252.2965	11.3939807
371	252.2965	9.4535879
372	252.2965	13.4569059
373	252.2965	10.4142695
374	252.2965	11.9097917

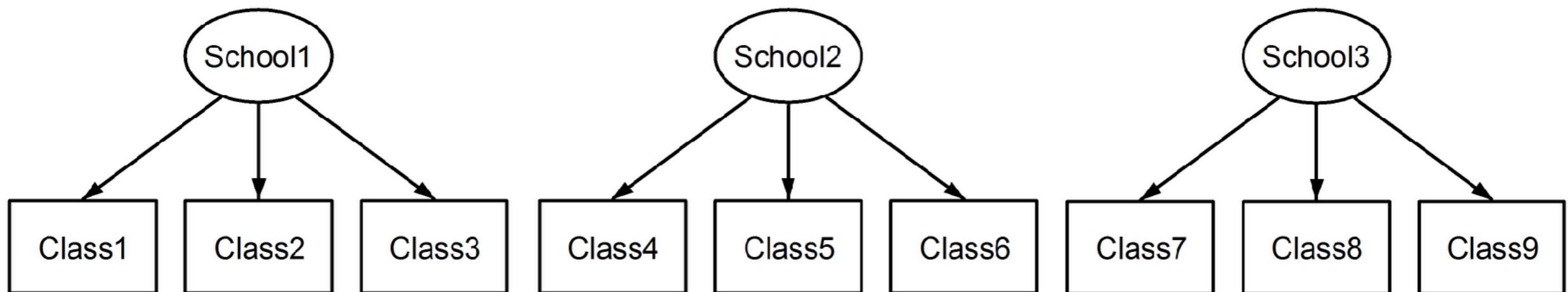
\dots
 $(1 + \text{days} \mid \text{subject})$

random intercepts and
slopes (+ correlation)

\$subject	(Intercept)	days
308	253.9479	19.6264139
309	211.7328	1.7319567
310	213.1579	4.9061843
330	275.1425	5.6435987
331	273.7286	7.3862680
332	260.6504	10.1632535
333	268.3684	10.2245979
334	244.5523	11.4837825
335	251.3700	-0.3355554
337	286.2321	19.1090061
349	226.7662	11.5531963
350	238.7807	17.0156766
351	256.2344	7.4119501
352	272.3512	13.9920698
369	254.9484	11.2985741
370	226.3701	15.2027922
371	252.5051	9.4335432
372	263.8916	11.7253342
373	248.9752	10.3915245
374	271.1451	11.0782697

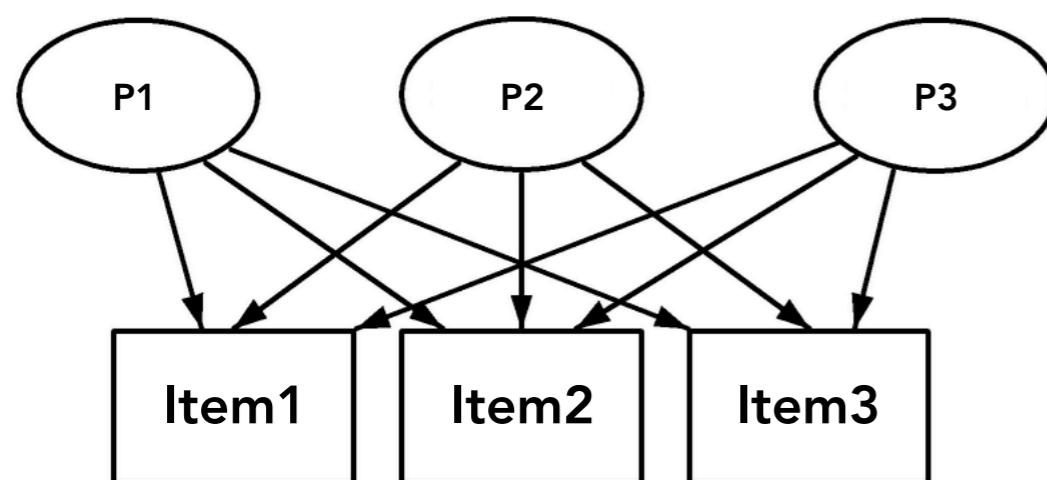
Multi-level models

nested $(1 | \text{School}/\text{Class})$



each class only appears within one school

crossed $(1 | \text{participant}) + (1 | \text{item})$



**each participant
rates each item**

Example

If a person is wet, then the person fell into a swimming pool.
A person fell into a swimming pool.

How valid is the conclusion/How likely is it that the person is wet?

id	instruction	plausibility	inference	validity	what	type	response	content
10	probabilistic	plausible	DA	invalid	denial	reversed	100	C1
10	probabilistic	implausible	MT	valid	denial	reversed	60	C1
10	probabilistic	implausible	MP	valid	affirmation	reversed	94	C1
10	probabilistic	plausible	AC	invalid	affirmation	reversed	70	C1
10	probabilistic	implausible	DA	invalid	denial	original	100	C2

Between subjects ANOVA

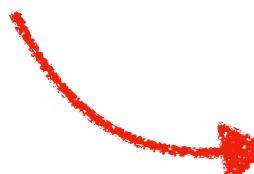
we are in lm() world

Was there an effect of instruction on responses?

set this whenever you
want to run ANOVA models

was manipulated
between participants

```
1 # set contrasts to using sum contrasts  
2 options(contrasts = c("contr.sum","contr.poly"))
```



Between subjects ANOVA

we are in lm() world

Was there an effect of instruction on responses?

Option 1: use aov_ez() from the afex package

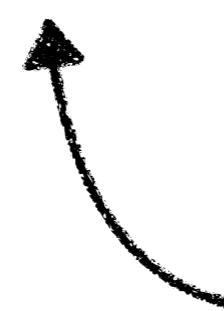
```
1 aov_ez(id = "id",
2         dv = "response",
3         between = "instruction",
4         data = df.reasoning)
```

```
More than one observation per cell, aggregating the data using
mean (i.e, fun_aggregate = mean)!Anova Table (Type 3 tests)

Response: response
Effect      df     MSE      F ges p.value
1 instruction 1, 38 253.43 0.31 .008   .583
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Option 2: use lm() and Anova() from the car package

```
1 lm(formula = response ~ instruction,
2      data = df.reasoning %>%
3        group_by(id, instruction) %>%
4        summarize(response = mean(response))) %>%
5      Anova(type = 3,
6             method = "F")
```



```
Anova Table (Type III tests)

Response: response
          Sum Sq Df  F value Pr(>F)
(Intercept) 250530  1 988.5637 <2e-16 ***
instruction    78  1  0.3066  0.583
Residuals     9630 38

```

I took means by hand here

Repeated measures ANOVA



Repeated measures ANOVA

Was there an effect of validity & plausibility on responses?

```
1 df.reasoning_within = sk2011.1 %>%
2   filter(instruction == "probabilistic")
3
4 aov_ez(id = "id",
5         dv = "response",
6         within = c("validity", "plausibility"),
7         data = df.reasoning_within)
```

Repeated measures ANOVA

Was there an effect of validity & plausibility on responses?

```
1 df.reasoning_within = sk2011.1 %>%
2   filter(instruction == "probabilistic")
3
4 aov_ez(id = "id",
5   dv = "response",
6   within = c("validity", "plausibility"),
7   data = df.reasoning_within)
8
9 mixed(formula = response ~ validity * plausibility + (1 | id) + (1 | validity:id) + (1 | plausibility:id),
10   data = df.reasoning_within %>%
11     group_by(id, validity, plausibility) %>%
12     summarize(response = mean(response)))
```

random effects structure of a
repeated measures ANOVA

the culprit

More than one observation per cell, aggregating the data
using mean (i.e., fun_aggregate = mean)!

Anova Table (Type 3 tests)

Response: response

	Effect	df	MSE	F	ges	p.value
1	validity	1, 19	183.01	0.01	<.001	.904
2	plausibility	1, 19	321.44	30.30	***	.366 <.001
3	validity:plausibility	1, 19	65.83	9.21	**	.035 .007

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

Fitting one lmer() model. ~~boundary (singular) fit: see ?isSingular~~
Mixed Model Anova Table (Type 3 tests, KR method)

Model: response ~ validity * plausibility + (1 | id) + (1 |
validity:id) + (1 | plausibility:id)

	Effect	df	F	p.value
1	validity	1, 19	0.02	.901
2	plausibility	1, 19	34.21	*** <.001
3	validity:plausibility	1, 19	8.93	** .008

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

close but not the same

Repeated measures ANOVA

Was there an effect of validity & plausibility on responses?

```
1 lmer(formula = response ~ validity * plausibility +  
       (1 | id) + (1 | validity:id) + (1 | plausibility:id),  
2   data = df.reasoning_within %>%  
3   group_by(id, validity, plausibility) %>%  
4   summarize(response = mean(response))) %>%  
5 summary()
```

```
boundary (singular) fit: see ?isSingular  
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']  
  
REML criterion at convergence: 631.5  
  
Scaled residuals:  
    Min      1Q  Median      3Q     Max  
-1.38401 -0.51395  0.09847  0.42796  1.77284  
  
Random effects:  
Groups           Name        Variance Std.Dev.  
plausibility:id (Intercept) 1.084e+02 1.041e+01  
validity:id      (Intercept) 5.100e+01 7.142e+00  
id               (Intercept) 1.597e-08 1.264e-04  
Residual          6.792e+01 8.242e+00  
Number of obs: 80, groups: plausibility:id, 40; validity:id, 40; id, 20  
  
Fixed effects:  
              Estimate Std. Error    df t value Pr(>|t|)  
(Intercept)  80.5344    2.1987 42.4507 36.629 < 2e-16 ***  
validity1    -0.1844    1.4574 23.7558 -0.127  0.90040  
plausibility1 11.0344    1.8866 29.8209  5.849 2.18e-06 ***  
validity1:plausibility1 2.7531    0.9214 18.1054  2.988  0.00785 **  
---  
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

very small value

Repeated measures ANOVA

This procedure does not guarantee that LMM results match RM-ANOVA results as the latter can accommodate data that would imply negative variance terms for the LMM models (which it cannot handle).



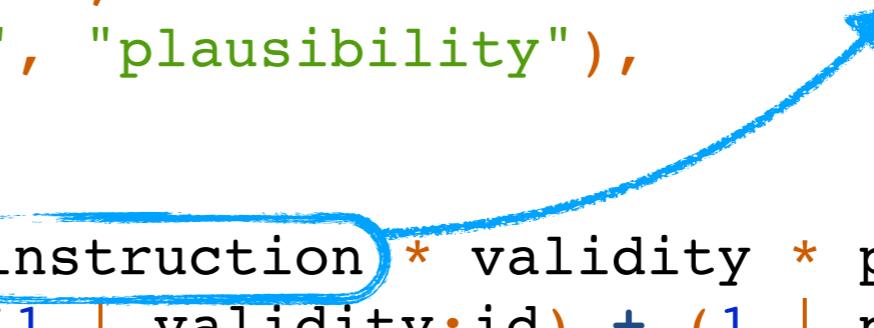
<https://stats.stackexchange.com/questions/117660/what-is-the-lme4lmer-equivalent-of-a-three-way-repeated-measures-anova>

Mixed ANOVA

Was there an effect of instruction, validity or plausibility on responses?

```
1 aov_ez(id = "id",
2         dv = "response",
3         between = "instruction",
4         within = c("validity", "plausibility"),
5         data = df.reasoning)
6
7 mixed(formula = response ~ instruction * validity * plausibility +
8        (1 | id) + (1 | validity:id) + (1 | plausibility:id),
9        data = df.reasoning %>%
10       group_by(id, validity, plausibility, instruction) %>%
11       summarize(response = mean(response)))
```

doesn't appear in the random effects structure



**here both solutions yield the same result,
because the `lmer()` is happy**

TLDR

- If you want ANOVAs, then I recommend using the `aov_ez()` function

```
1 aov_ez(id = "id",
2         dv = "response",
3         between = "instruction",
4         within = c("validity", "plausibility"),
5         data = df.reasoning)
```

- assumes that there is **only one observation** in each cell
- will calculate means first if that's not the case

TLDR

- if you have repeated observations that you want to model (rather than just taking the mean), then I recommend using the `mixed()` function

```
1 mixed(formula = rt ~ noise * angle +
         (noise + angle | id),
2       data = df.reaction)
3
4 lmer(formula = rt ~ noise * angle +
         (noise + angle | id),
5       data = df.reaction) %>%
6   Anova(type = 3,
7         test.statistic = "F")
```

```
Fitting one lmer() model. [DONE]
Calculating p-values. [DONE]
Mixed Model Anova Table (Type 3 tests, KR-method)

Model: rt ~ noise * angle + (noise + angle | id)
Data: df.reaction
Effect      df        F p.value
1    noise  1, 9 33.77 ***  <.001
2    angle  2, 8 31.59 ***  <.001
3 noise:angle 2, 18 45.31 ***  <.001
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger df)

Response: rt
              F Df Df.res Pr(>F)
(Intercept) 598.419  1     9 1.527e-09 ***
noise        33.765  1     9 0.0002560 ***
angle        31.588  2     8 0.0001596 ***
noise:angle  45.309  2    18 9.426e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MOVE NEXT
TIME

lmer() standard operating procedures

Standard Operating Procedures For Using Mixed-Effects Models

A Principled Workflow from the Decision, Development, and Psychopathology (D2P2) Lab
document version 1.0.0 – 28 June 2020

[This document will be continuously updated and expanded; it may contain typos and other errors--both unintentional errors and errors based on incorrect or outdated knowledge--we will try to improve these things in future versions. Feel free to let us know if you spotted such things, how to further improve this document!]

Authors (in alphabetical order except that the youngsters were so kind to put the oldest guy in the lab first; BF)

Bernd Figner, Johannes Algermissen, Floor Burghoorn, Leslie Held, Afreen Khalid, Felix Klaassen, Farnaz Mosannenzadeh, Julian Quandt

Content/Analysis Steps

Content/Analysis Steps	1
1. Before data collection:	
Power/ design/ planning/ sample size	3
1.1. Power analysis	3
1.2. Sensitivity analysis	4
1.3. Sequential sampling with stopping rules	5
1.4. More readings	5
2. Preparing data	6
2.1. Categorical variables	6
2.2. Continuous variables	6
3. Running the model	7
3.1. Model specification and random effects	7
3.2. Addressing convergence warnings	7
3.2.1. Convergence warnings in R's lme4	7
3.2.2. Or we choose the Bayesian approach	9
3.2.3. MixedModels in Julia	9

also available
in Files >
papers on
Canvas

[http://decision-lab.org/wp-content/uploads/2020/07/
SOP_Mixed_Models_D2P2_v1_0_0.pdf](http://decision-lab.org/wp-content/uploads/2020/07/SOP_Mixed_Models_D2P2_v1_0_0.pdf)

2. Preparing data

2.1. Categorical variables

We most commonly use sum-to-zero coding for categorical predictors (via the options (contrasts = c("contr.sum", "contr.poly")) for factors. We use this coding scheme because we are typically interested in main effects and main interactions rather than simple effects or simple interactions (see also [this blog post by Dale Barr](#)). One option is also to use the command `mixed()` from the package **afex**, as it will automatically set all contrasts to sum-to-zero.

Reasons to deviate might include the use of *custom contrasts* to test specific hypotheses.

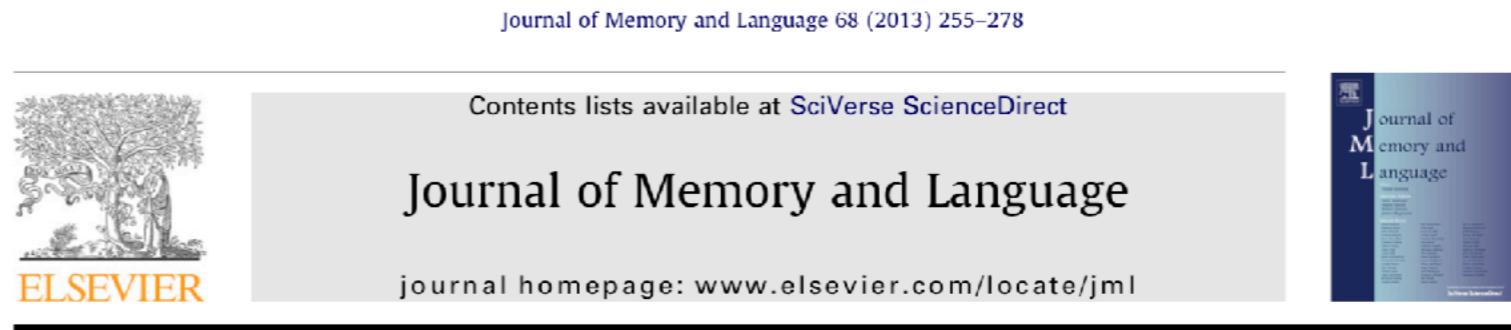
We usually will *follow-up* on significant effects involving factors with more than two levels by either restricting analyses to only two levels in the form of follow-up models (i.e. analyzing a subset of the data comprising only two levels of the given factor) or, alternatively, we use some post-hoc procedures, e.g., using the package **emmeans** (for more details on both, see the section on post-hocs and follow-ups below).

2.2. Continuous variables

As a default, we typically use z-standardization for the continuous predictors (to help with model estimation), unless there are specific reasons not to do so (e.g. if we want to interpret effects on the original scale; in these cases, we typically center the predictor(s)) Centering is essential to make interactions interpretable and avoid so-called nonessential multicollinearity ([Dunlap and Kemery, 1987](#); [Marquardt, 1980](#); also see [this blog post by Philipp Masur](#)).

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!



"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."

What shall I include as random effects?

- general advice:
 - start maximal (as supported by the design)
 - random intercepts for different participants
 - random slopes when participants are tested multiple times
 - random intercepts for items
 - reduce complexity of the random effects structure step by step
 - remove the correlation between random effects first

Remove the correlation component from your model

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
3                   data = df.sleep)
4 # model summary
5 fit.lmer %>%
6   summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (1 + days | subject)
Data: df.sleep

REML criterion at convergence: 1771.4

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-3.9707 -0.4703  0.0276  0.4594  5.2009 

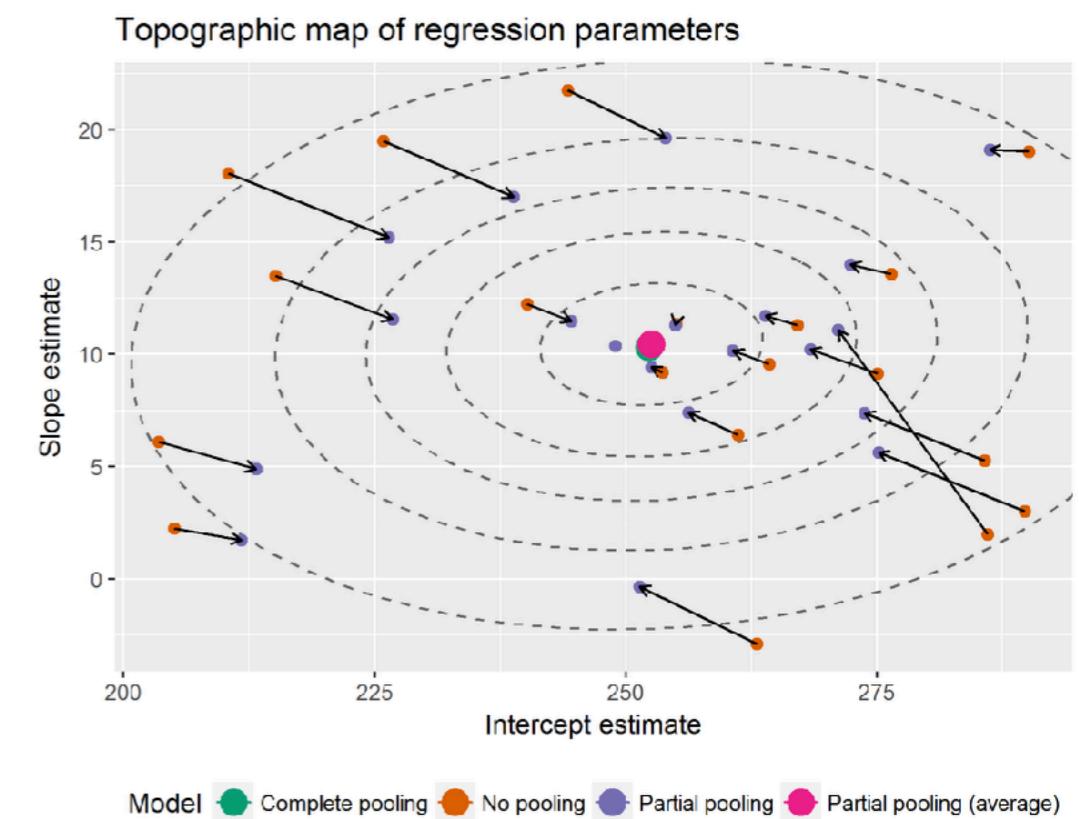
Random effects:
Groups   Name        Variance Std.Dev. Corr
subject (Intercept) 582.73   24.140
          days       35.03   5.919   0.07
Residual            649.36   25.483

Number of obs: 183, groups: subject, 20

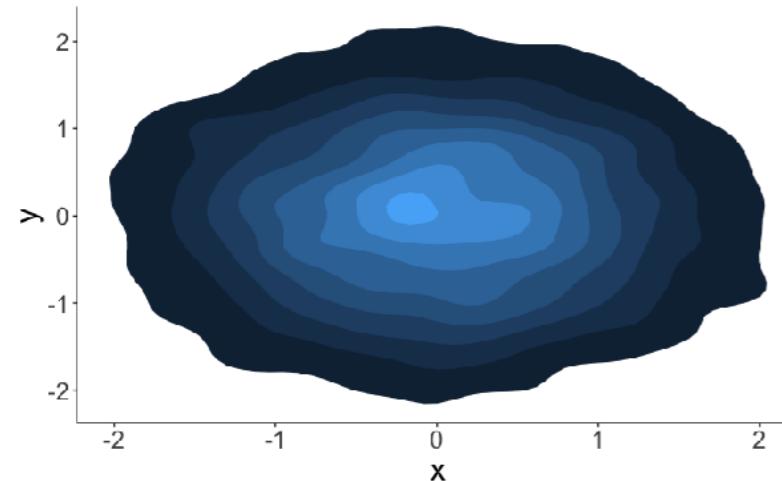
Fixed effects:
            Estimate Std. Error t value
(Intercept) 252.543    6.433  39.256
days         10.452    1.542   6.778

Correlation of Fixed Effects:
  (Intr) days  
days -0.137
```

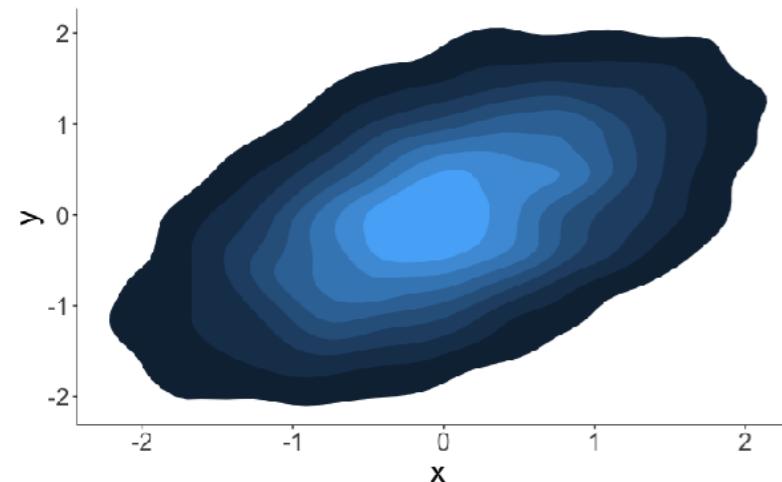
multivariate Gaussian



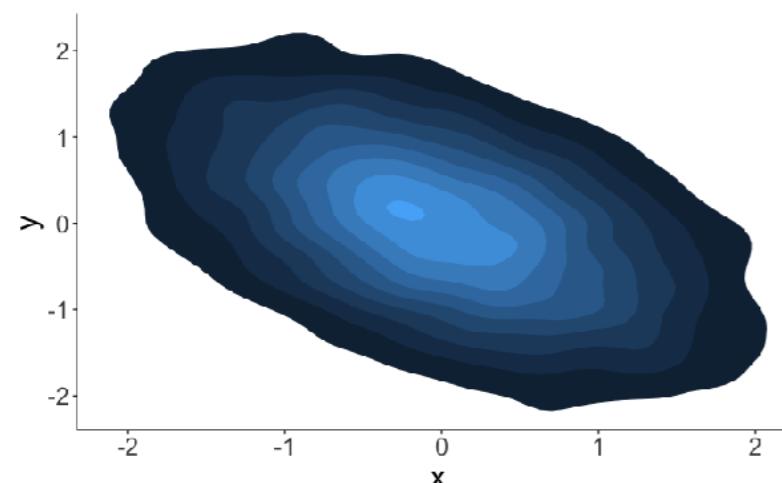
Remove the correlation component from your model



uncorrelated



positively correlated



negatively correlated

Remove the correlation component from your model

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (0 + days | subject) + (1 | subject),
3                  data = df.sleep)
4 # model summary
5 fit.lmer %>%
6   summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)
Data: df.sleep

REML criterion at convergence: 1771.5

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-3.9805 -0.4673  0.0250  0.4589  5.2083 

Random effects:
 Groups   Name        Variance Std.Dev.    
subject  days       35.88    5.99      
subject.1 (Intercept) 598.11   24.46    
Residual           647.90   25.45    
Number of obs: 183, groups: subject, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept) 252.550    6.491  38.907
days         10.439    1.556   6.708

Correlation of Fixed Effects:
  (Intr) days  
days -0.184
```

↑
random slopes
↑
random intercepts

independent Gaussians

Remove the correlation component from your model

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days || subject),
3                  data = df.sleep)
4 # model summary
5 fit.lmer %>%
6   summary()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: reaction ~ 1 + days + (0 + days | subject) + (1 | subject)
Data: df.sleep

REML criterion at convergence: 1771.5

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-3.9805 -0.4673  0.0250  0.4589  5.2083 

Random effects:
 Groups   Name        Variance Std.Dev.    
subject  days       35.88    5.99      
subject.1 (Intercept) 598.11   24.46    
Residual           647.90   25.45    
Number of obs: 183, groups: subject, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept) 252.550    6.491  38.907
days         10.439    1.556   6.708

Correlation of Fixed Effects:
  (Intr) days  
days -0.184
```

alternative syntax (doesn't model correlation between random effects)

independent Gaussians

What if lmer() fails to converge?

3.2.1. Convergence warnings in R's lme4

In case of convergence warnings, we attempt the listed approaches, typically in the order in which they are listed (these steps are based mainly on recommendations by Ben Bolker/the lme4 team and [Barr et al., 2013](#)). For each step, we check whether it resolves the convergence issues.

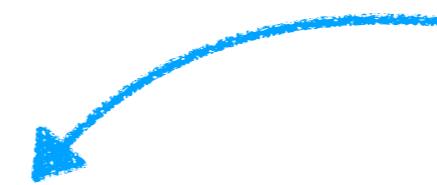
Before going through the listed steps, some of us would always set the optimizer to bobyqa as default via `optimizer = c("bobyqa")` (since it has been suggested that it might work better for the kind of data that we typically have in psychology) and/or switch off the calculation of the gradient and Hessian via `control = [g]lmerControl(calc.derivs = FALSE)`; these settings might already resolve the convergence issues .

1. We increase the number of iterations to the maximum.
2. We use the estimates from the previous (non-converged) fit as our new starting values.
3. We compare the estimates of different optimizers (e.g., using `allFit()`); if different optimizers give highly similar estimates (even if they give convergence warnings), the convergence warnings can be considered false positives.
4. We follow the steps suggested [in Ben Bolker's blog post](#):
 - a. Center independent (and dependent) variables instead of scaling; multiply the independent variables by 10 (or 100) to increase the variance
 - b. Robustness check: Check whether certain random correlations are close to +/-1 and/or certain random slope variances are close to 0. If yes, remove those; afterwards check whether the estimates are still the same
 - c. Double check gradient calculations: Check the (parallel) minimum of the absolute and relative gradients. If those gradients are > 0.001, gradient calculation is likely not a problem.

What if lmer() fails to converge?

```
1 # fit the model
2 fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
3   data = df.sleep)
4
5 # explore different optimization algorithms
6 fit.all = allFit(fit.lmer)
7
8 # summarize result
9 fit.all %>% summary()
```

comparison of the different optimization algorithms



\$fixef	(Intercept)	days
bobyqa	252.5426	10.45212
Nelder_Mead	252.5426	10.45212
nlminbwrap	252.5426	10.45212
nloptwrap.NLOPT_LN_NELDERMEAD	252.5426	10.45212
nloptwrap.NLOPT_LN_BOBYQA	252.5426	10.45212

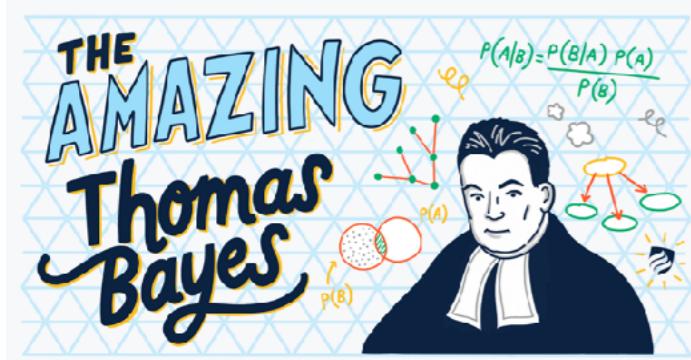
\$llik	bobyqa	Nelder_Mead	nlminbwrap
	-885.7239	-885.7239	-885.7239
	nloptwrap.NLOPT_LN_NELDERMEAD	nloptwrap.NLOPT_LN_BOBYQA	

\$sdcor	subject.(Intercept)	subject.days.(Intercept)	subject.days	sigma
bobyqa	24.13911		5.918866	0.06927657 25.48261
Nelder_Mead	24.13900		5.918891	0.06928125 25.48261
nlminbwrap	24.13911		5.918867	0.06927628 25.48261
nloptwrap.NLOPT_LN_NELDERMEAD	24.13979		5.918851	0.06927975 25.48255
nloptwrap.NLOPT_LN_BOBYQA	24.13979		5.918851	0.06927975 25.48255

<https://rdrr.io/cran/lme4/man/convergence.html>

What if lmer() fails to converge?

1. We drop random effects in the following order: random correlations, random slopes of covariates (where significance is of no interest), random intercepts ("0+" instead "1+") (following [Barr et al., 2013](#)). We never remove the random slopes of the variables of interest (i.e., the ones for which we want to conduct significance tests).
Please note that removing random correlation terms can be tricky if random slopes are estimated for factors with 3 or more levels. In that case, it is probably easiest to use `afex::mixed()` with `expand_re = TRUE` (an alternative option is to create manually the relevant contrasts yourself and add them as predictors to your model, which allows you to suppress the random corrections using the double pipe symbol `||`).
2. We try to run separate analyses: For example, one model to only test the fixed and random effect of A (with fixed effect of B present); then one model to only test the effect of B. If we really have to drop random slopes, we follow the next step:
3. We follow the PCA approach suggested by [rePsychLing](#) (see [Bates et al., 2015](#)) that is performing a PCA on the random effects and following the guidelines described in the paper.
 - a. We use a likelihood ratio test to test whether the model fit becomes significantly worse. As we prefer a more conservative approach here (i.e., rather err on the side of keeping too many random effects; we prioritize avoiding inflated Type 2 errors for this kind of decision), we use larger alpha-level of .2 ([Matuschek et al., 2017](#)).
 - b. Alternatively, we suggest an Information criterion approach to avoid using a *p* value for our inclusion/exclusion decision, but choose the best model based on *BIC* or *AIC*.

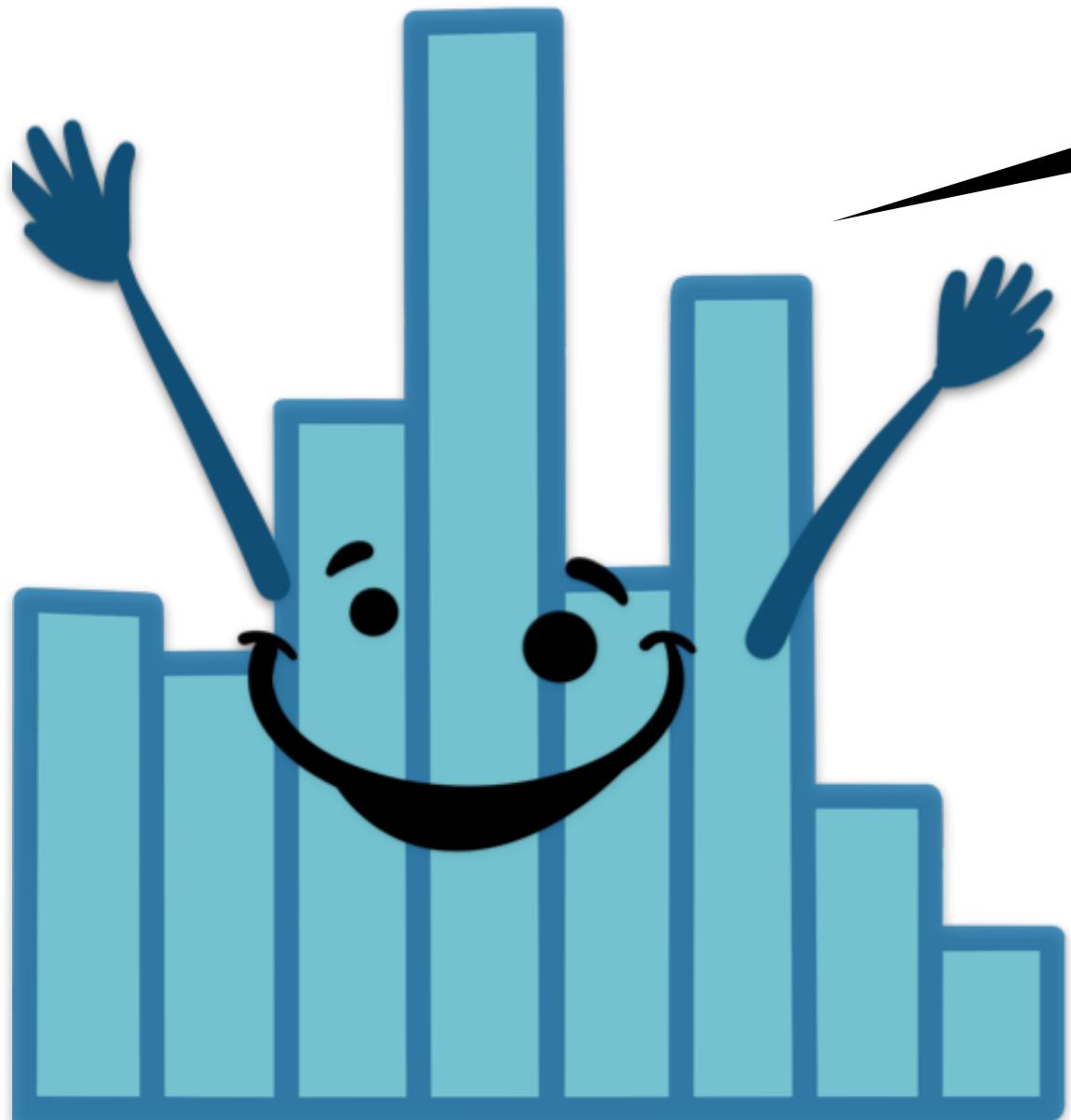


3.2.2. Or we choose a Bayesian approach

As an alternative to targeting convergence issues within **lme4**, we suggest fitting the same model with **brms** and comparing it to the **lme4** fit. We assume that both provide similar results when

01:00

stretch break!



Generalized linear model

Titanic dataset



Titanic data set

891 passengers

passenger_id	survived	pclass	name	sex	age	sib_sp	parch	ticket	fare	cabin	embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley	female	38	1	0	PC 17599	71.28	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.92		S
4	1	1	Futrelle, Mrs. Jacques Heath (Iluv)	female	35	1	0	113803	53.10	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.46		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.86	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.07		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth)	female	27	0	2	347742	11.13		S
10	1	2	Nasser, Mrs. Nicholas (Adele)	female	14	1	0	237736	30.07		C

Is there a relationship between fare and survived?

```
1 fit.lm = lm(formula = survived ~ 1 + fare,  
2               data = df.titanic)  
3  
4 fit.lm %>% summary()
```

Call:

```
lm(formula = survived ~ 1 + fare, data = df.titanic)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.9653	-0.3391	-0.3222	0.6044	0.6973

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.3026994	0.0187849	16.114	< 2e-16	***
fare	0.0025195	0.0003174	7.939	6.12e-15	***

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

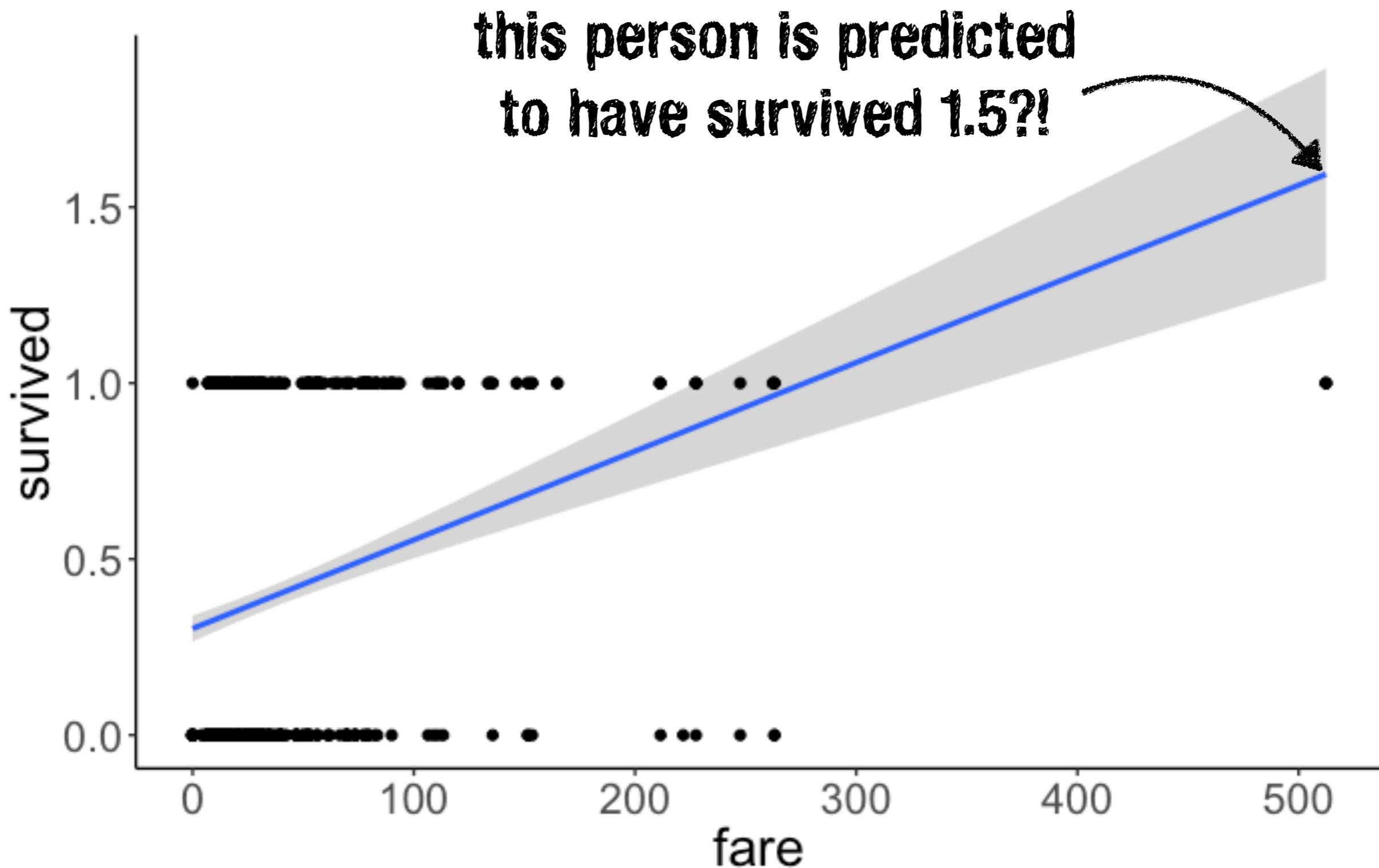
Residual standard error: 0.4705 on 889 degrees of freedom

Multiple R-squared: 0.06621, Adjusted R-squared: 0.06516

F-statistic: 63.03 on 1 and 889 DF, p-value: 6.12e-15

How should we interpret this parameter?

Is there a relationship between fare and survived?



Generalized linear model

- so far, we have only looked at situations where our dependent variable was continuous
- what about situations in which we have a binary dependent variable?
 - survived vs. died
 - correct vs. incorrect
 - benign vs. malignant
 - yes vs. no
 - ...



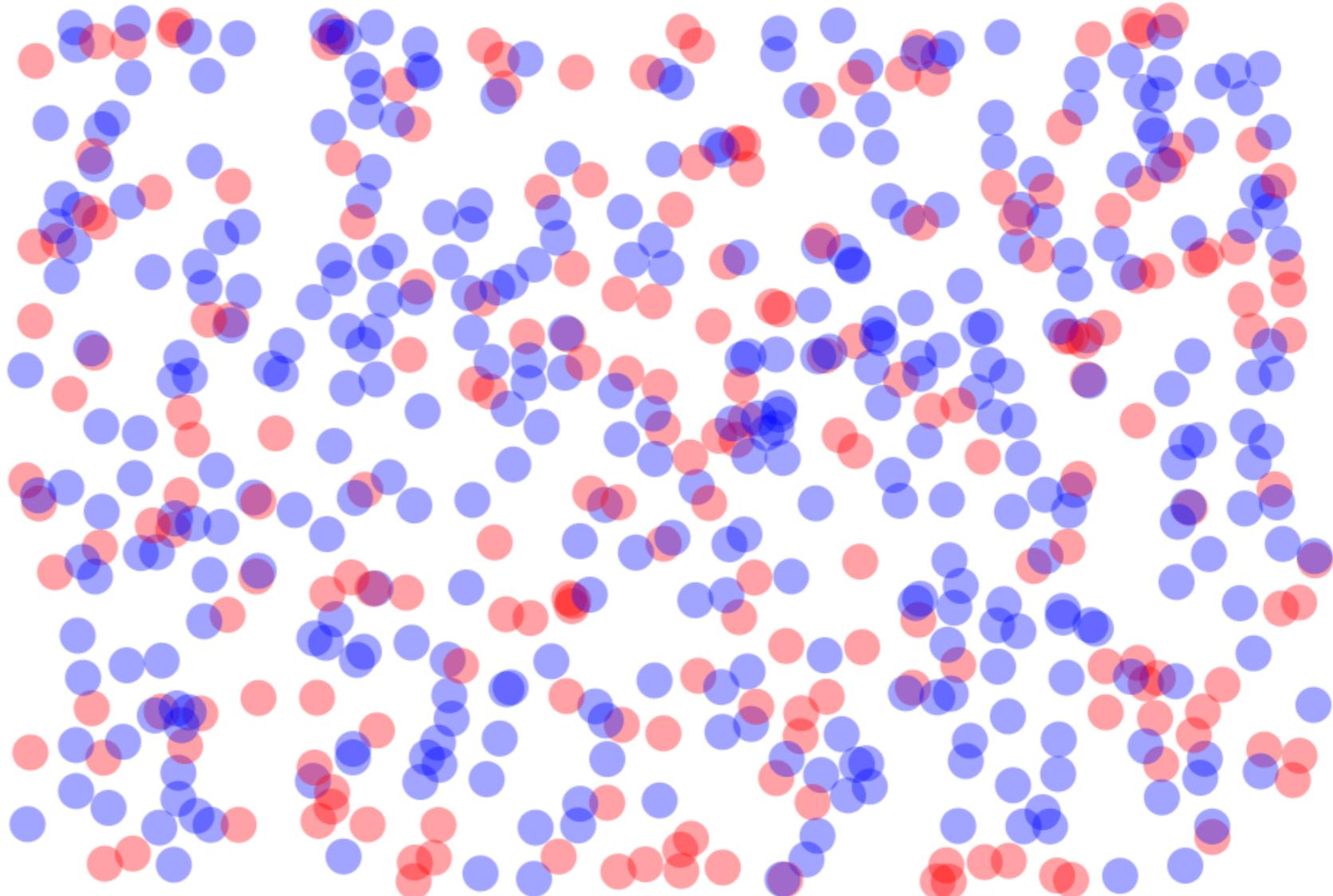
Logistic regression

Demo

[Introduction](#) [Data](#) [Modeling](#) [Predictions](#) [Thresholds](#) [Accuracy](#) [Vocab](#) [Sensitivity](#) [Specificity](#) [ROC](#) [About](#)

Binary Predictions Metrics

This visual explanation introduces the metrics of model fit used when predicting of **binary outcomes**. It uses the challenge of classifying tumors as **benign** or **malignant** to explore the importance of these metrics.



<http://mfviz.com/binary-predictions/>

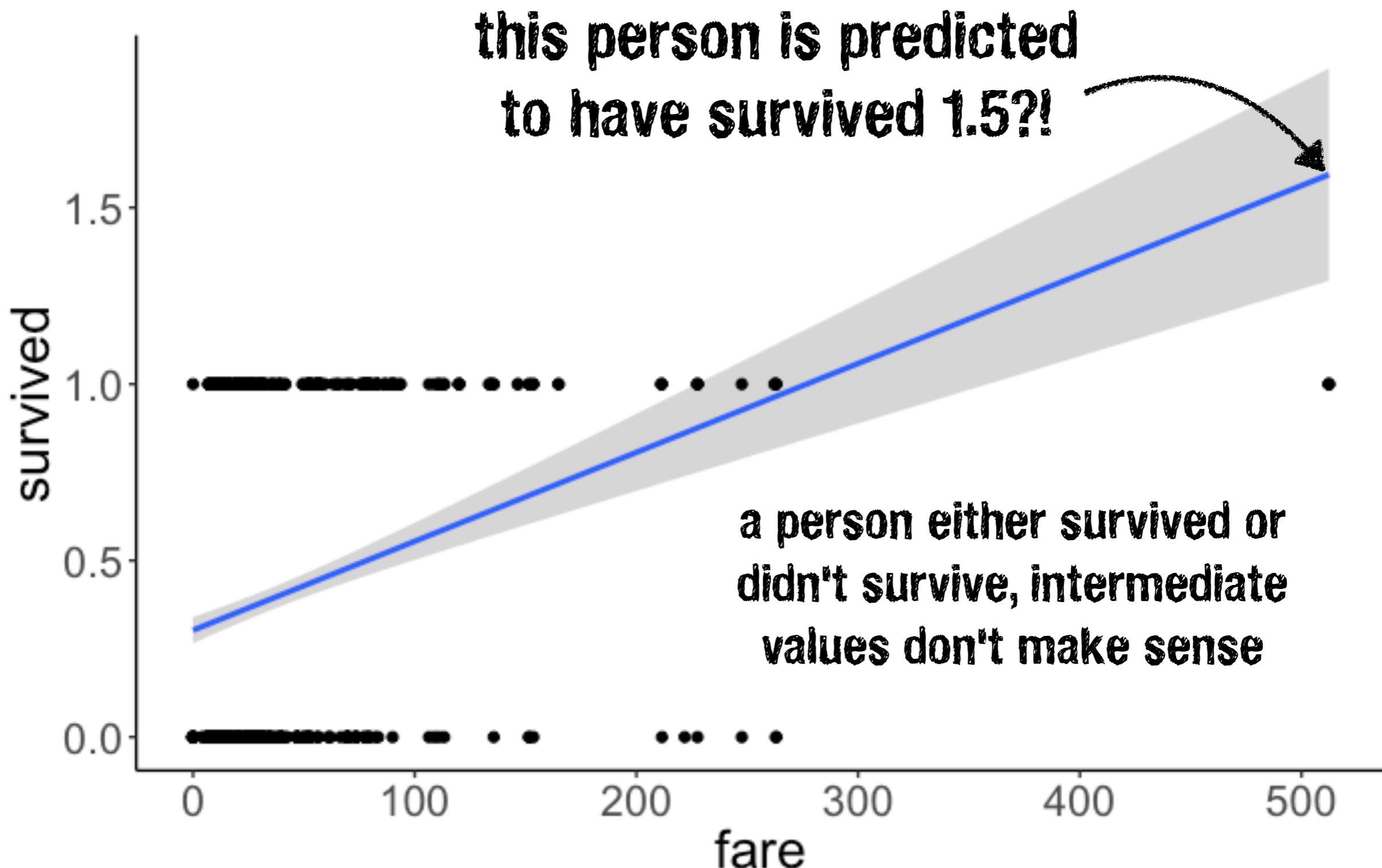
Is there a relationship between fare and survived?

Can we still use a linear model to make predictions about a binary outcome variable?

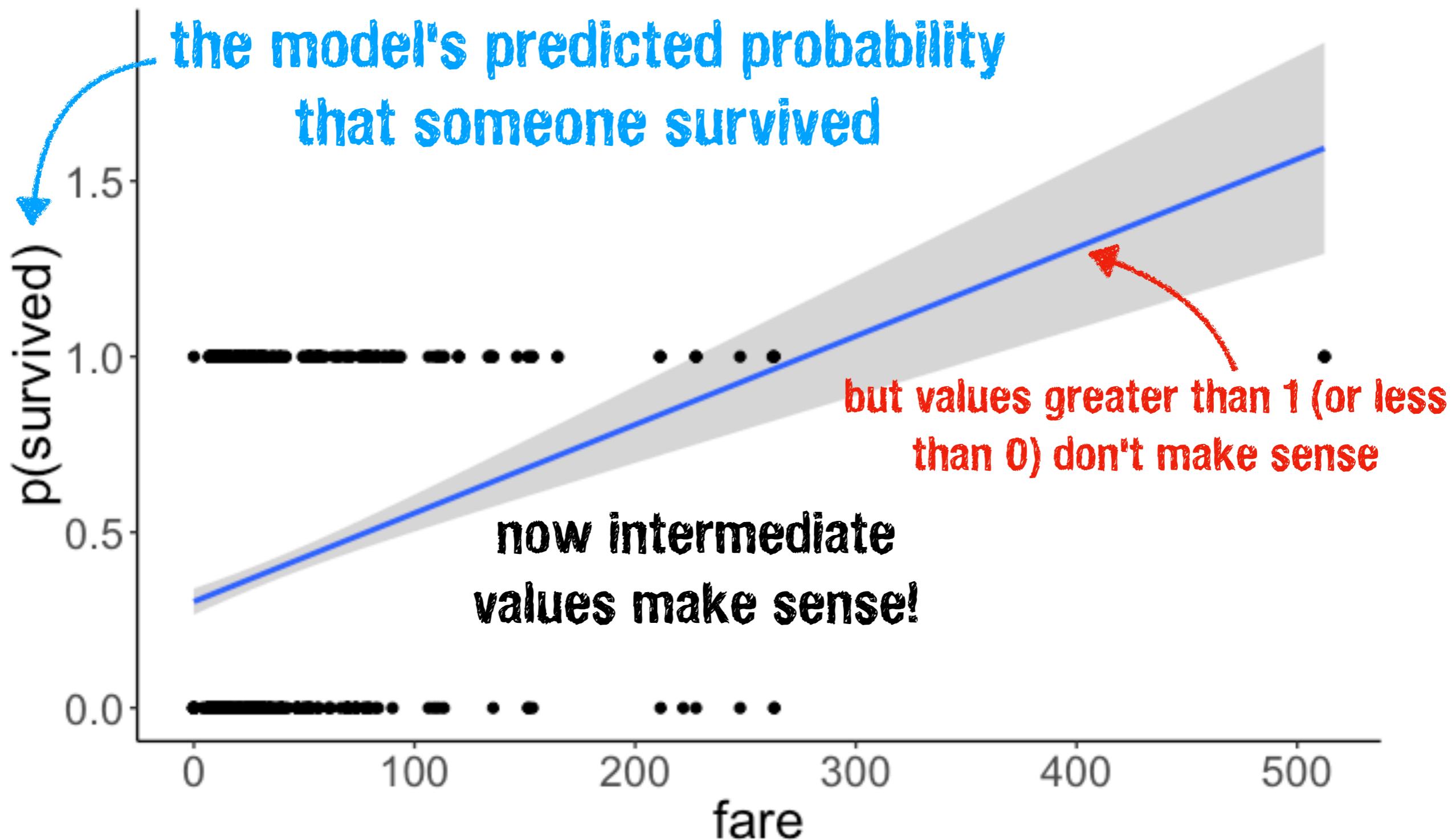
The fact that this class is called "**Generalized linear model**" suggests we can!

Is there a relationship between fare and survived?

```
fit.lm = lm(formula = survived ~ 1 + fare, data = df.titanic)
```



Is there a relationship between fare and survived?



From linear regression to logistic regression

$$Y_i = b_0 + b_1 \cdot X_i + e_i \quad \text{predict the value of Y}$$

$$\pi_i = b_0 + b_1 \cdot X_i + e_i \quad \text{predict the probability of Y}$$

$$\pi_i = P(Y_i = 1)$$

let's just do a
logit transform

we need to map from $[-\infty, +\infty]$ to $[0, 1]$

Logit transform

$$\pi_i = b_0 + b_1 \cdot X_i + e_i \quad \text{predict the probability of Y}$$

$$\pi_i = P(Y_i = 1)$$

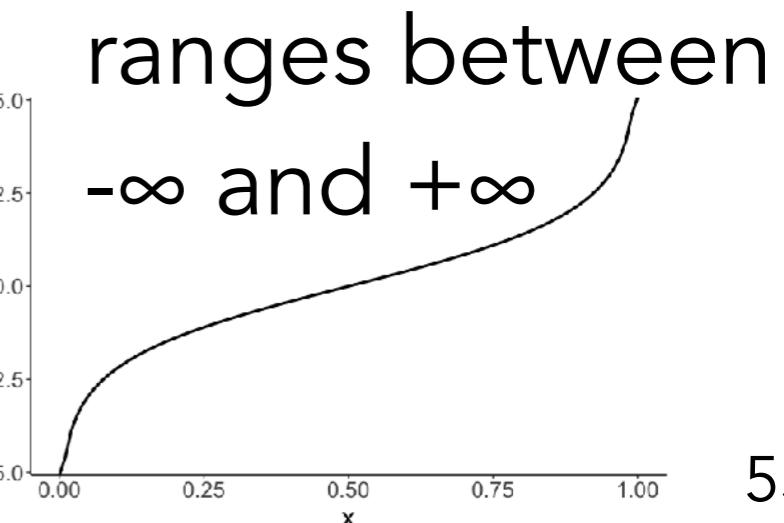
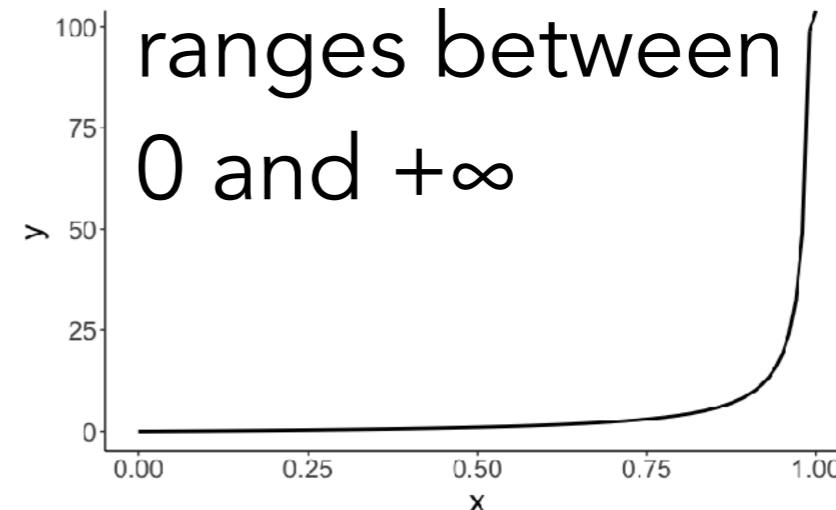
Step 1: Calculate the "odds"

$$\frac{P(Y_i = 1)}{P(Y_i = 0)} = \frac{\pi_i}{1 - \pi_i}$$

Step 2: Take the (natural) log

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = b_0 + b_1 \cdot X_i + e_i$$

we need to transform the dependent variable so that it can take any value between $-\infty$ and $+\infty$ (we can then transform it back into a probability later)



Logit transform

log odds

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = b_0 + b_1 \cdot X_i + e_i$$

$$\pi_i = P(Y_i = 1)$$

if log odds == 0: $P(Y_i = 1) = P(Y_i = 0)$

if log odds > 0: $P(Y_i = 1) > P(Y_i = 0)$

if log odds < 0: $P(Y_i = 1) < P(Y_i = 0)$

after transforming from a binary variable, to a probability, to odds, to log odds, the model looks like a normal linear model



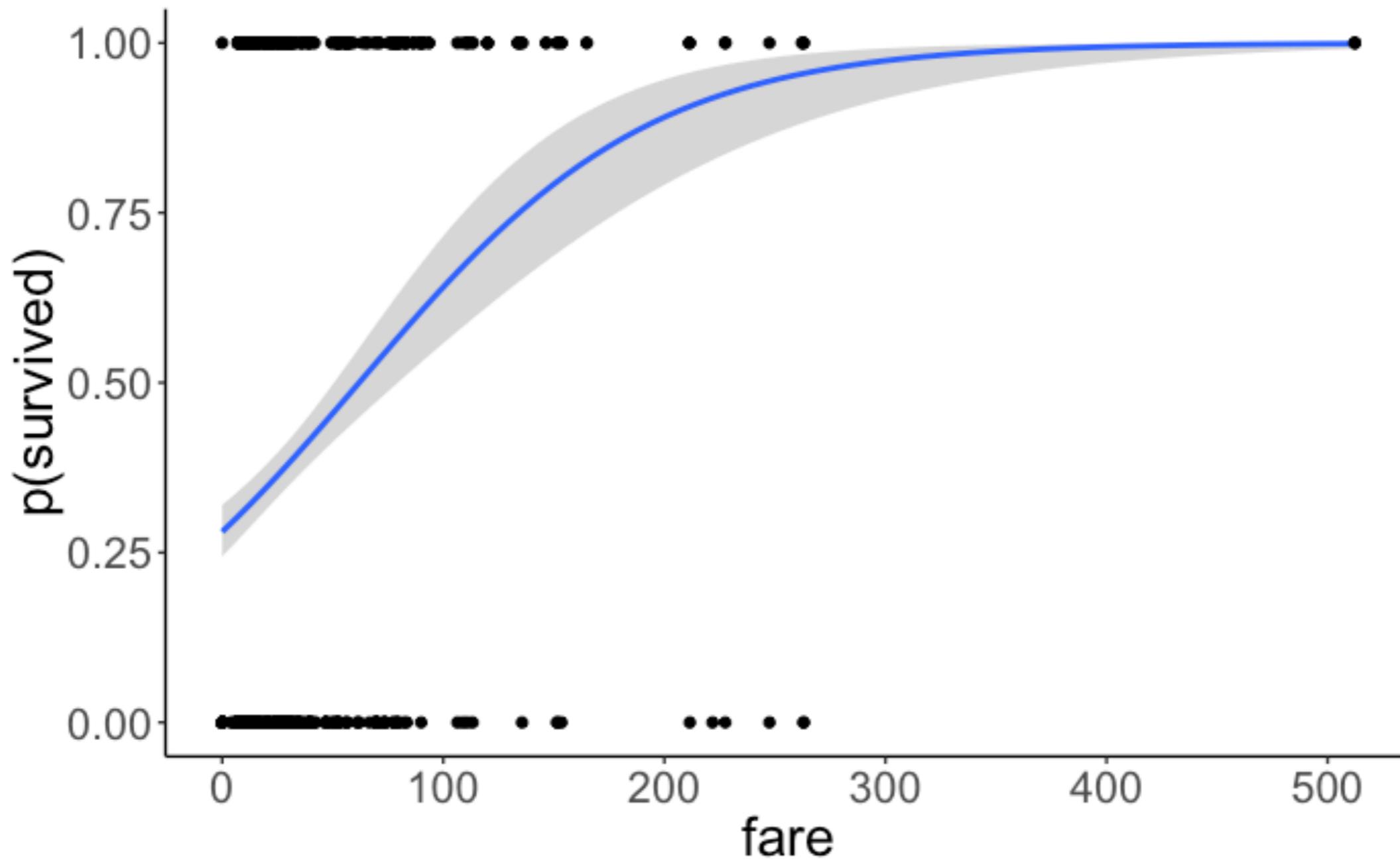
Fitting a logistic regression in R

```
1 fit.glm = glm(formula = survived ~ 1 + fare,  
2 family = "binomial",  
3 data = df.titanic)  
4  
5 fit.glm %>% summary()
```

```
Call:  
glm(formula = survived ~ 1 + fare, family = "binomial", data = df.titanic)  
  
Deviance Residuals:  
    Min      1Q  Median      3Q     Max  
-2.4906 -0.8878 -0.8531  1.3429  1.5942  
  
Coefficients:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***  
fare         0.015197  0.002232  6.810 9.79e-12 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
Null deviance: 1186.7 on 890 degrees of freedom  
Residual deviance: 1117.6 on 889 degrees of freedom  
AIC: 1121.6  
  
Number of Fisher Scoring iterations: 4
```

Visualize the model's predictions

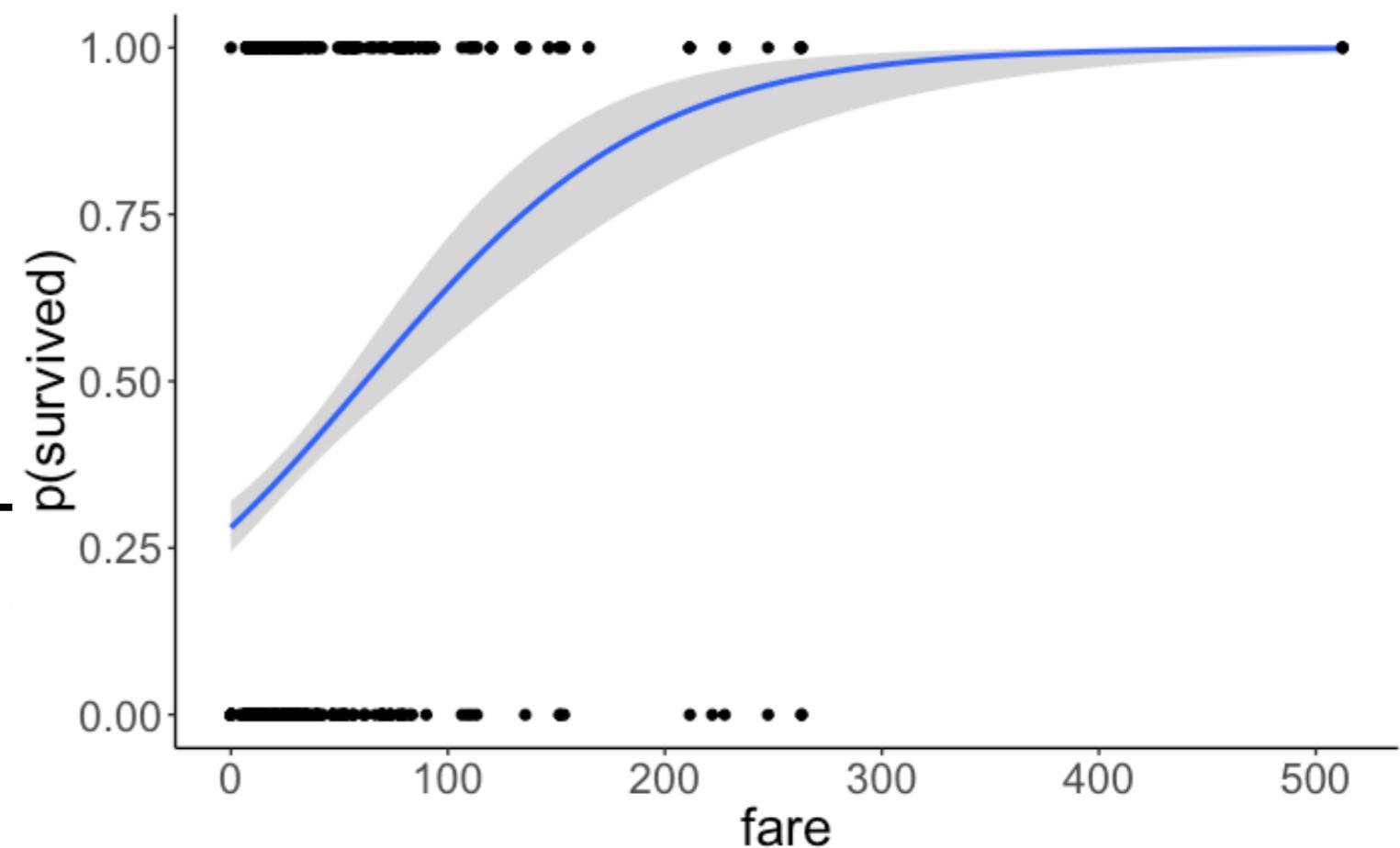
```
1 ggplot(data = df.titanic,  
2         mapping = aes(x = fare,  
3                             y = survived)) +  
4     geom_smooth(method = "glm",  
5                  method.args = list(family = "binomial")) +  
6     geom_point() +  
7     labs(y = "p(survived)")
```



Interpreting the model output

Interpreting the model output

```
Call:  
glm(formula = survived ~ 1 + fare,  
  
Deviance Residuals:  
    Min      1Q      Median      3Q  
-2.4906 -0.8878 -0.8551  1.3429  
log odds?  
Coefficients:  
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***  
fare         0.015197  0.002232   6.810 9.79e-12 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
Null deviance: 1186.7 on 890 degrees of freedom  
Residual deviance: 1117.6 on 889 degrees of freedom  
AIC: 1121.6  
  
Number of Fisher Scoring iterations: 4
```



Transform log odds into probability

$$\pi = P(Y = 1)$$

just a placeholder

$$\ln\left(\frac{\pi}{1 - \pi}\right) = V$$

logit transformation

$$\pi = \frac{e^V}{1 + e^V}$$

inverse logit

gives us back the probability
(which is much easier to interpret)

$$\pi_i = \frac{e^{b_0 + b_1 \cdot X_i + e_i}}{1 + e^{b_0 + b_1 \cdot X_i + e_i}}$$

another way to
specify the model

Interpreting the model output

inverse logit

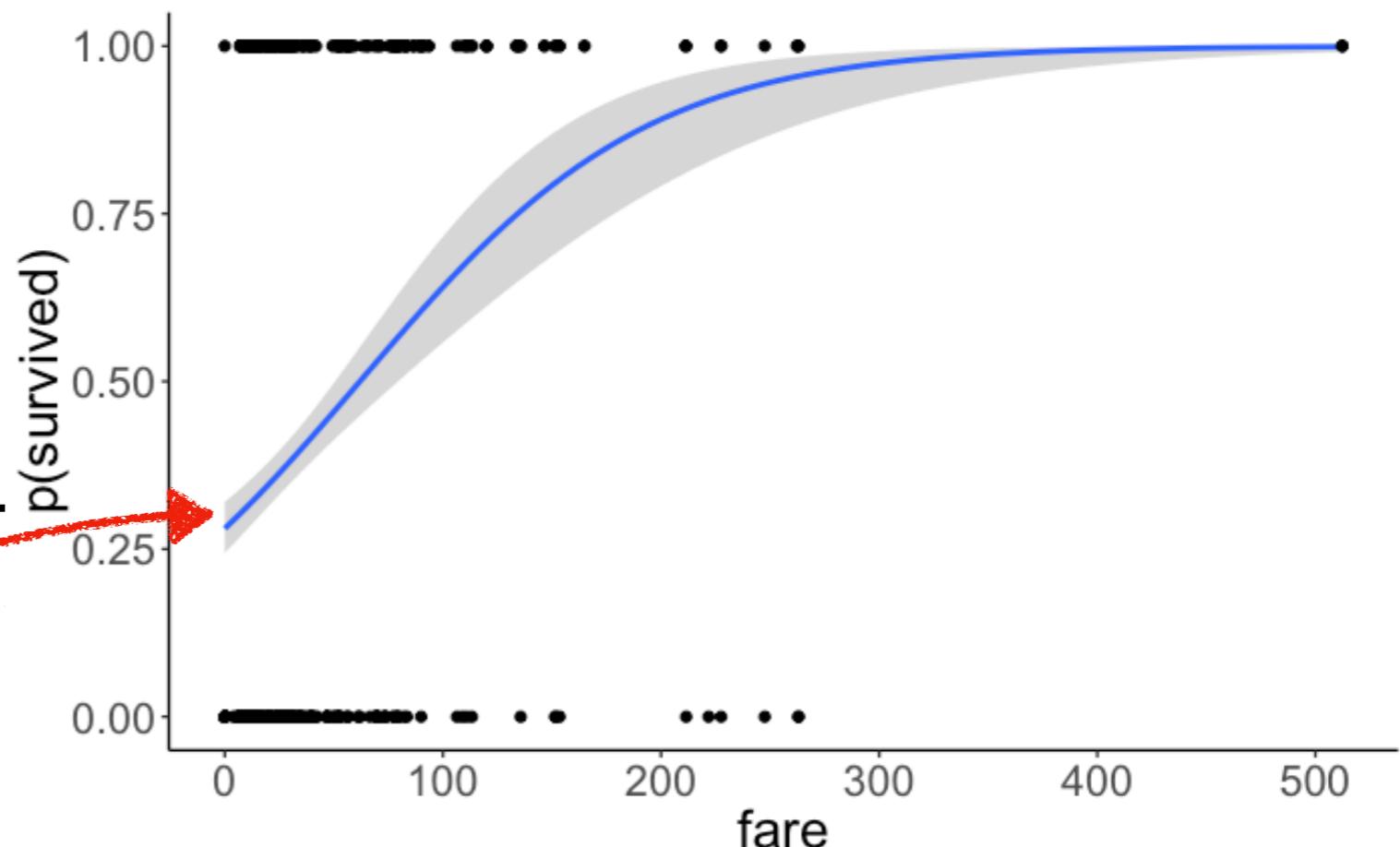
$$\pi = \frac{e^{-0.94}}{1 + e^{-0.94}} \approx 0.28$$

```
Call:  
glm(formula = survived ~ 1 + fare,  
  
Deviance Residuals:  
    Min      1Q  Median      3Q  
-2.4906 -0.8878 -0.8531  1.3429  
  
Coefficients:  
             Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***  
fare         0.015197  0.002232  6.810 9.79e-12 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

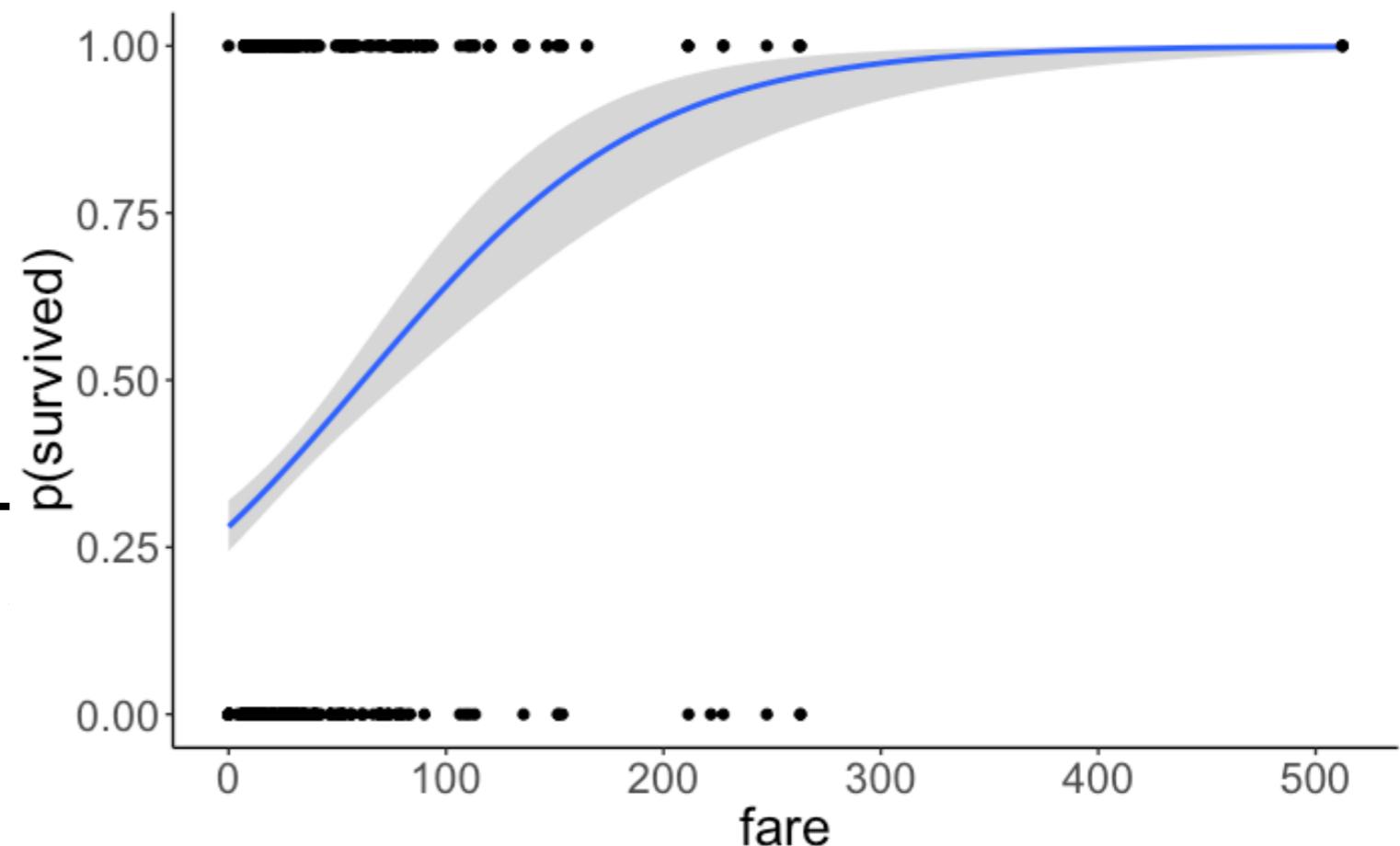
Null deviance: 1186.7 on 890 degrees of freedom
Residual deviance: 1117.6 on 889 degrees of freedom
AIC: 1121.6

Number of Fisher Scoring iterations: 4



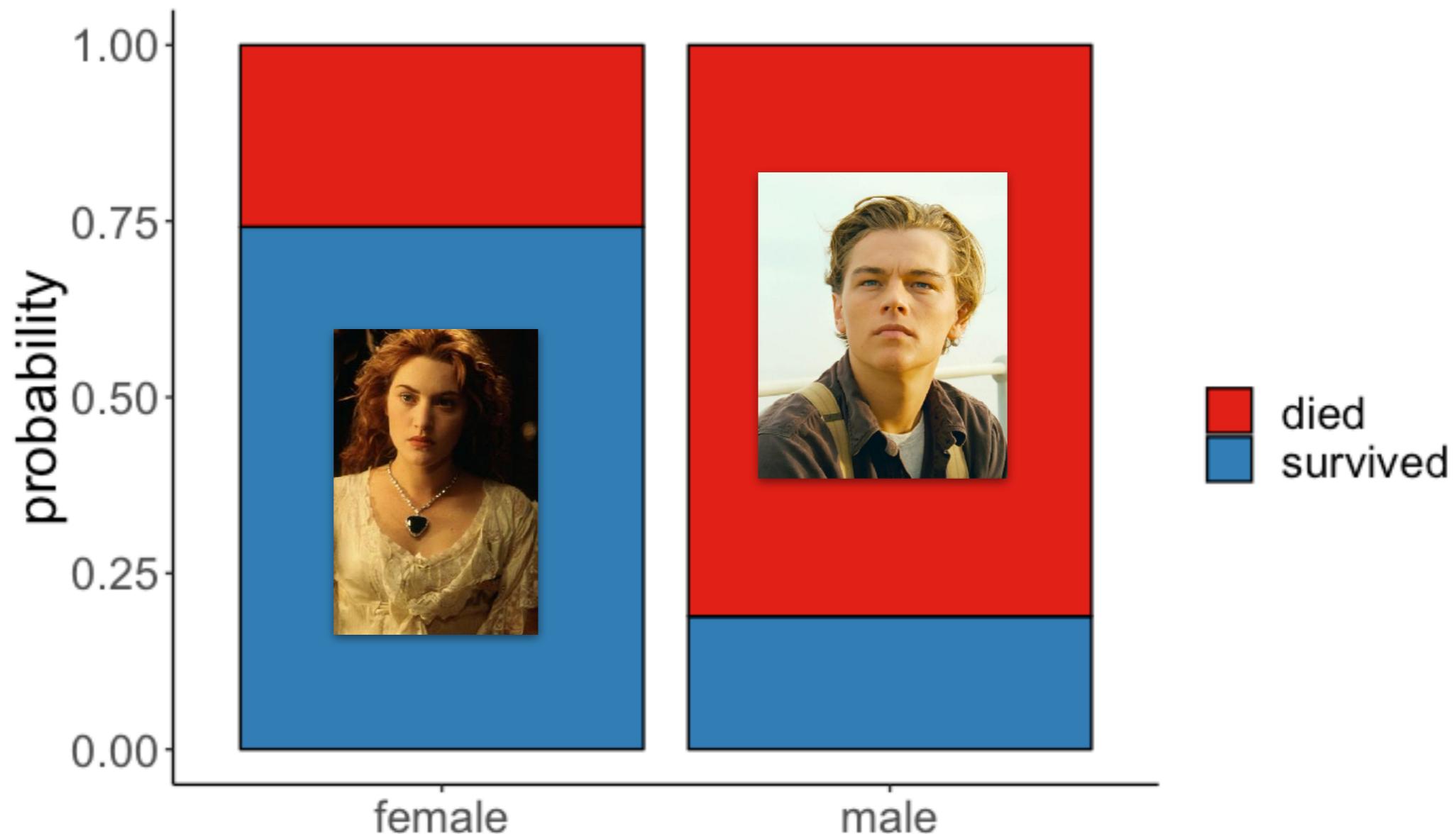
Interpreting the model output

```
Call:  
glm(formula = survived ~ 1 + fare,  
  
Deviance Residuals:  
    Min      1Q  Median      3Q  
-2.4906 -0.8878 -0.8531  1.3429  
  
Coefficients:  
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.941330  0.095129 -9.895 < 2e-16 ***  
fare         0.015197  0.002232   6.810 9.79e-12 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
Null deviance: 1186.7 on 890 degrees of freedom  
Residual deviance: 1117.6 on 889 degrees of freedom  
AIC: 1121.6  
  
Number of Fisher Scoring iterations: 4
```



Let's consider a binary predictor

Was the probability of survival different between female and male passengers on the Titanic?



Let's consider a binary predictor

```
1 fit.glm2 = glm(formula = survived ~ sex,  
2 family = "binomial",  
3 data = df.titanic)  
4  
5 fit.glm2 %>% summary()
```

```
Call:  
glm(formula = survived ~ sex, family = "binomial", data = df.titanic)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6462	-0.6471	-0.6471	0.7725	1.8256

sex was significantly associated with survival

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.0566	0.1290	8.191	2.58e-16 ***
sexmale	-2.5137	0.1672	-15.036	< 2e-16 ***

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1186.7 on 890 degrees of freedom
Residual deviance: 917.8 on 889 degrees of freedom
AIC: 921.8

Number of Fisher Scoring iterations: 4

Let's consider a binary predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{sex}_i + e_i$$

Coefficients:						
	Estimate	Std. Error	z value	Pr(> z)		
(Intercept)	1.0566	0.1290	8.191	2.58e-16	***	
sexmale	-2.5137	0.1672	-15.036	< 2e-16	***	

sex	survived	n	p	p(survived sex)
female	0	81	0.09	0.26
female	1	233	0.26	0.74
male	0	468	0.53	0.81
male	1	109	0.12	0.19

if sex == 0:

$$\ln\left(\frac{\widehat{p(\text{survived})}_i}{1 - \widehat{p(\text{survived})}_i}\right) = b_0$$

$$p(\text{survived})_i = \frac{e^{b_0}}{1 + e^{b_0}} = 0.74$$

Let's consider a binary predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{sex}_i + e_i$$

Coefficients:						
	Estimate	Std. Error	z value	Pr(> z)		
(Intercept)	1.0566	0.1290	8.191	2.58e-16	***	
sexmale	-2.5137	0.1672	-15.036	< 2e-16	***	

sex	survived	n	p	p(survived sex)
female	0	81	0.09	0.26
female	1	233	0.26	0.74
male	0	468	0.53	0.81
male	1	109	0.12	0.19

if $\text{sex} \equiv 1$:

$$\ln\left(\frac{\widehat{p(\text{survived})}_i}{1 - \widehat{p(\text{survived})}_i}\right) = b_0 + b_1$$

$$p(\text{survived})_i = \frac{e^{b_0+b_1}}{1 + e^{b_0+b_1}} = 0.19$$

Now let's go back to a continuous predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{fare}_i + e_i$$

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.941330	0.095129	-9.895	< 2e-16	***
fare	0.015197	0.002232	6.810	9.79e-12	***

fare	prediction	p(survival)
0	-0.94	0.28
10	-0.79	0.31
50	-0.18	0.45
100	0.58	0.64
500	6.66	1.00

$$\ln\left(\frac{\widehat{p(\text{survived})}}{1 - p(\text{survived})}\right) = -0.94 + 0.015 \cdot 10$$

$$p(\text{survived})_i = \frac{e^{-0.94+0.015 \cdot 10}}{1 + e^{-0.94+0.015 \cdot 10}} = 0.31$$

Now let's go back to a continuous predictor

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{fare}_i + e_i$$

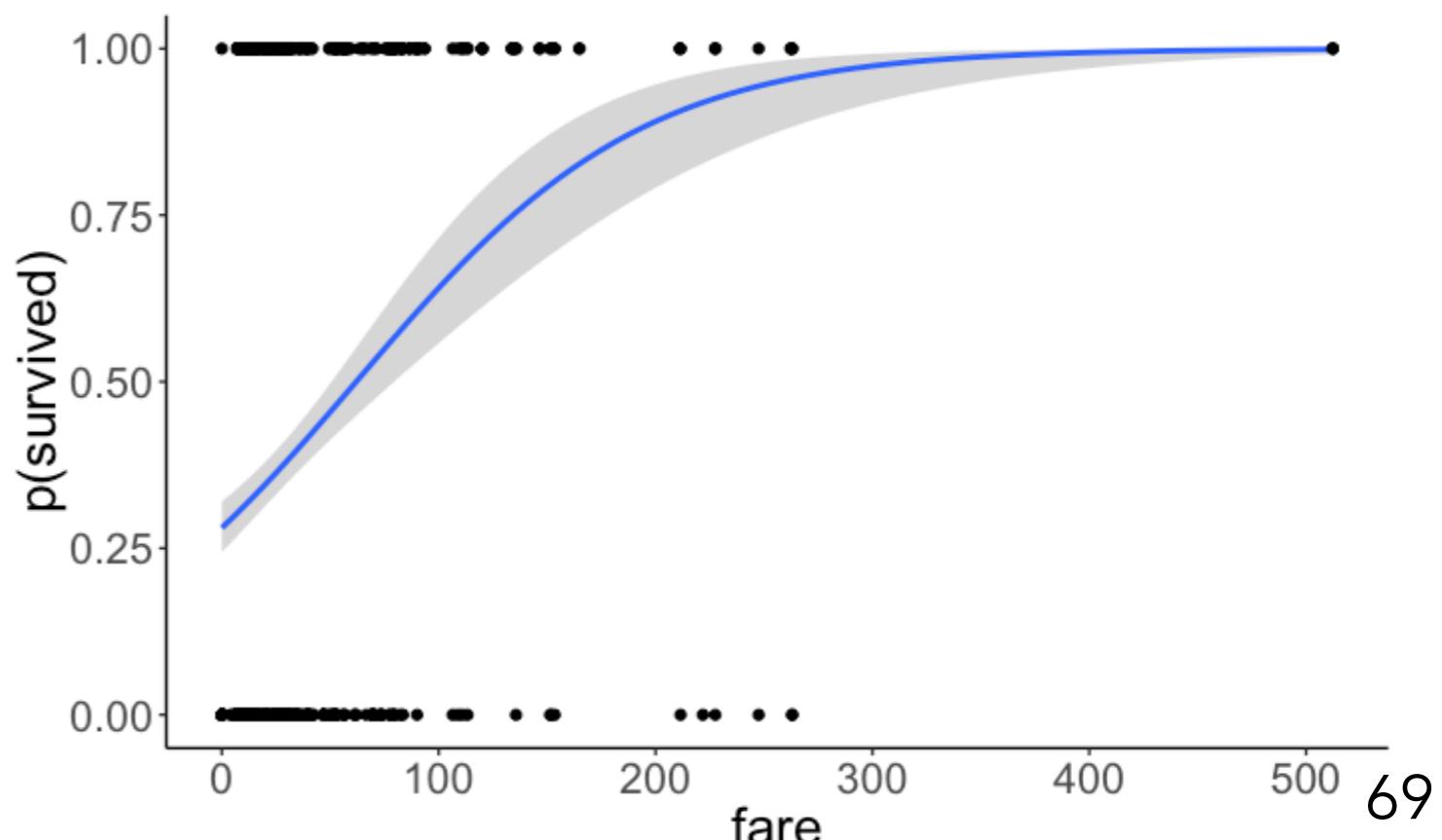
Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.941330	0.095129	-9.895	< 2e-16	***
fare	0.015197	0.002232	6.810	9.79e-12	***

For a one-unit increase in the fare, the expected increase in the odds of survival is 16%.

$$e^{0.015} \approx 1.16$$

fare	prediction	p(survival)
0	-0.94	0.28
10	-0.79	0.31
50	-0.18	0.45
100	0.58	0.64
500	6.66	1.00



Models with several predictors

$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{sex}_i + b_2 \cdot \text{fare}_i + e_i$$

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.647100  0.148502  4.358 1.32e-05 ***
sexmale     -2.422760  0.170515 -14.208 < 2e-16 ***
fare        0.011214  0.002295  4.886 1.03e-06 ***
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
```

controlling for "fare" there is still a significant difference between female and male passengers

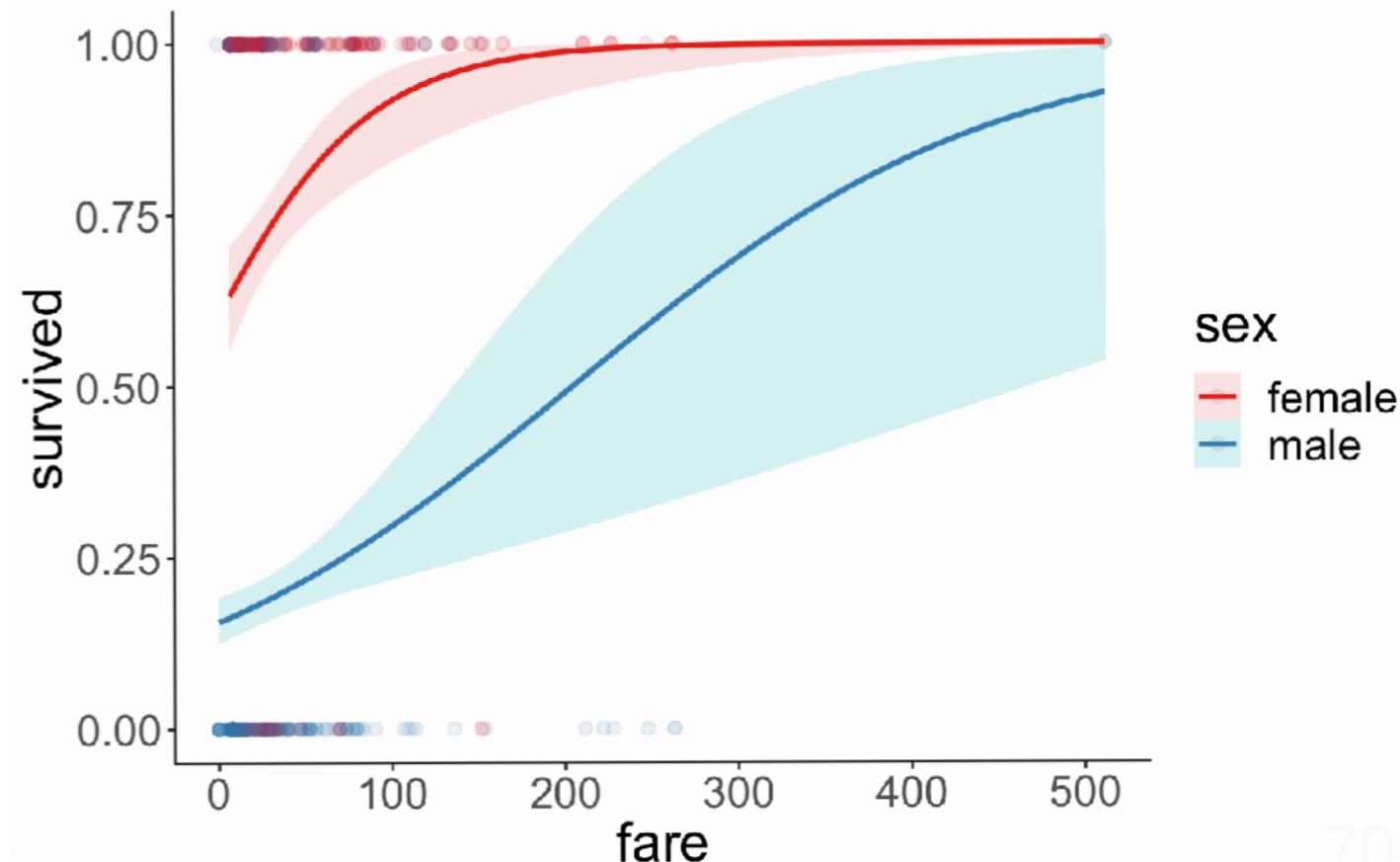
```
1 ggpredict(fit.glm,
2   terms = c("sex"))
```

```
# Predicted values of survived
# x = sex

x | Predicted | SE | 95% CI
---|---|---|---
female | 0.73 | 0.13 | [0.68, 0.78]
male | 0.20 | 0.11 | [0.16, 0.23]

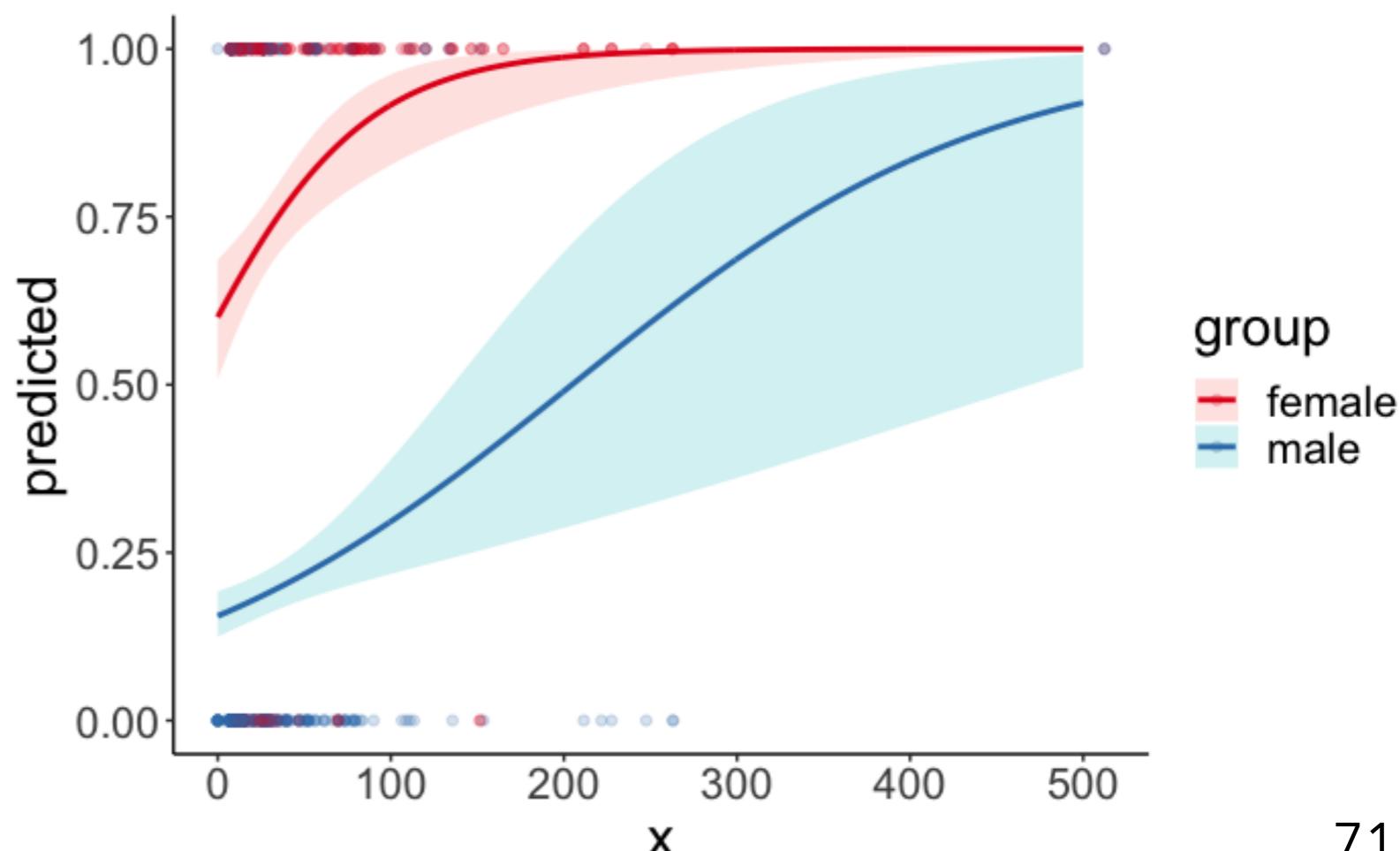
Adjusted for:
* fare = 32.20
```

```
1 df.titanic %>%
2   mutate(sex = as.factor(sex)) %>%
3   ggplot(data = .,
4         mapping = aes(x = fare,
5                         y = survived,
6                         color = sex)) +
7   geom_point(alpha = 0.1, size = 2) +
8   geom_smooth(method = "glm",
9               method.args = list(family = "binomial"),
10              alpha = 0.2,
11              aes(fill = sex)) +
12   scale_color_brewer(palette = "Set1")
```



Models with interactions

```
1 fit.glm3 = glm(formula = survived ~ 1 + sex * fare,  
2                      family = "binomial",  
3                      data = df.titanic)  
4  
5 df.data = ggpredict(fit.glm3,  
6                      terms = c("fare [0:500]", "sex"))  
7  
8 ggplot(data = df.data,  
9                      mapping = aes(x = x,  
10                         y = predicted,  
11                         color = group)) +  
12 geom_ribbon(mapping = aes(ymin = conf.low,  
13                         ymax = conf.high,  
14                         fill = group),  
15                         alpha = 0.2,  
16                         color = NA) +  
17 geom_line(size = 1) +  
18 geom_point(data = df.titanic,  
19                      mapping = aes(x = fare,  
20                         y = survived,  
21                         color = sex),  
22                         alpha = 0.2) +  
23 scale_color_brewer(palette = "Set1")
```



Fitting and reporting models

Simulating a logistic regression

```
1 # make example reproducible
2 set.seed(1)
3
4 # set parameters
5 sample_size = 1000
6 b0 = 0
7 b1 = 1
8
9 # generate data
10 df.data = tibble(
11   x = rnorm(n = sample_size),
12   y = b0 + b1 * x,
13   p = inv.logit(y) >= 0
14   mutate(response = rbinom(n(), size = 1, p = p))
15
16 # fit model
17 fit = glm(formula = response ~ 1 + x,
18            family = "binomial",
19            data = df.data)
20
21 # model summary
22 fit %>% summary()
```

set some parameters

linear model (y is in log odds)

transform into probability

randomly draw response

fit a logistic regression

summarize the result

Simulating a logistic regression

```
1 # make example reproducible
2 set.seed(1)
3
4 # set parameters
5 sample_size = 1000
6 b0 = 0
7 b1 = 1
8
9 # generate data
10 df.data = tibble(
11   x = rnorm(n = sample_size),
12   y = b0 + b1 * x,
13   p = inv.logit(y)) %>%
14   mutate(response = rbinom(n(), size = 1, p = p))
15
16 # fit model
17 fit = glm(formula = response ~ 1 + x,
18           family = "binomial",
19           data = df.data)
20
21 # model summary
22 fit %>% summary()
```

```
Call:
glm(formula = response ~ 1 + x, family = "binomial", data = df.data)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.1137 -1.0118 -0.4591  1.0287  2.2591 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.06214    0.06918 -0.898   0.369    
x             0.92905    0.07937 11.705 <2e-16 ***  
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1385.4 on 999 degrees of freedom
Residual deviance: 1209.6 on 998 degrees of freedom
AIC: 1213.6

Number of Fisher Scoring iterations: 3
```

Assessing the model fit

$$\text{log-likelihood} = \sum_{i=1}^n [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

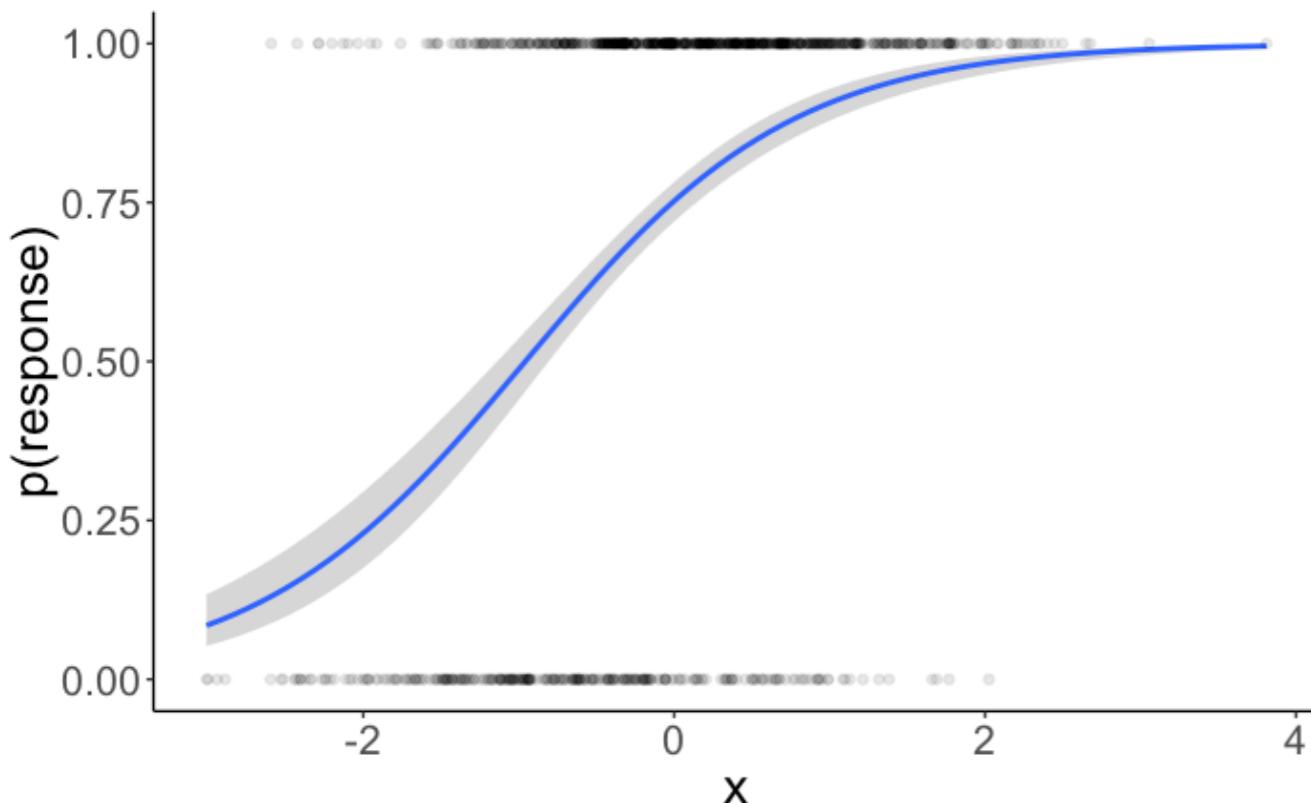
actual value → predicted value →

- calculate the probability of the observed response
- take the log of these probabilities
- sum them up to get the log-likelihood of the data (given the model)

response	p(Y = 1)	p(Y = response)	log(p(Y = response))
1	0.34	0.34	-1.07
0	0.53	0.47	-0.75
1	0.30	0.30	-1.20
1	0.81	0.81	-0.22
1	0.56	0.56	-0.58
0	0.30	0.70	-0.36
1	0.60	0.60	-0.52
1	0.65	0.65	-0.43
1	0.62	0.62	-0.48
0	0.41	0.59	-0.54

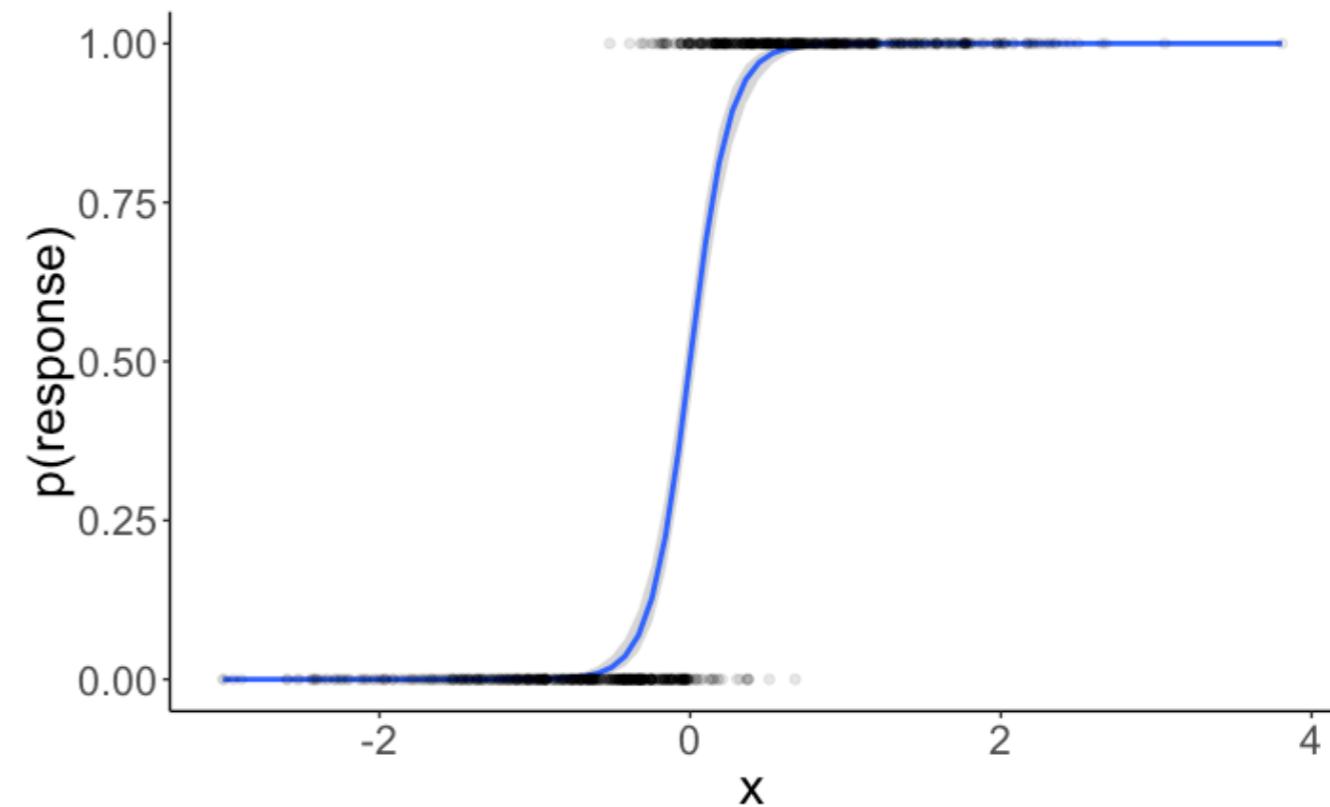
Assessing the model fit

doesn't predict the response very well



logLik	AIC	BIC
-501.65	1007.3	1017.12

predicts the response much better



logLik	AIC	BIC
-156.37	316.74	326.55

Testing hypotheses

aka checking
whether it's **worth it**

```
1 # fit compact model
2 fit.compact = glm(formula = survived ~ 1 + fare,
3                         family = "binomial",
4                         data = df.titanic)
5
6 # fit augmented model
7 fit.augmented = glm(formula = survived ~ 1 + sex + fare,
8                         family = "binomial",
9                         data = df.titanic)
10
11 # likelihood ratio test
12 anova(fit.compact, fit.augmented, test = "LRT")
```

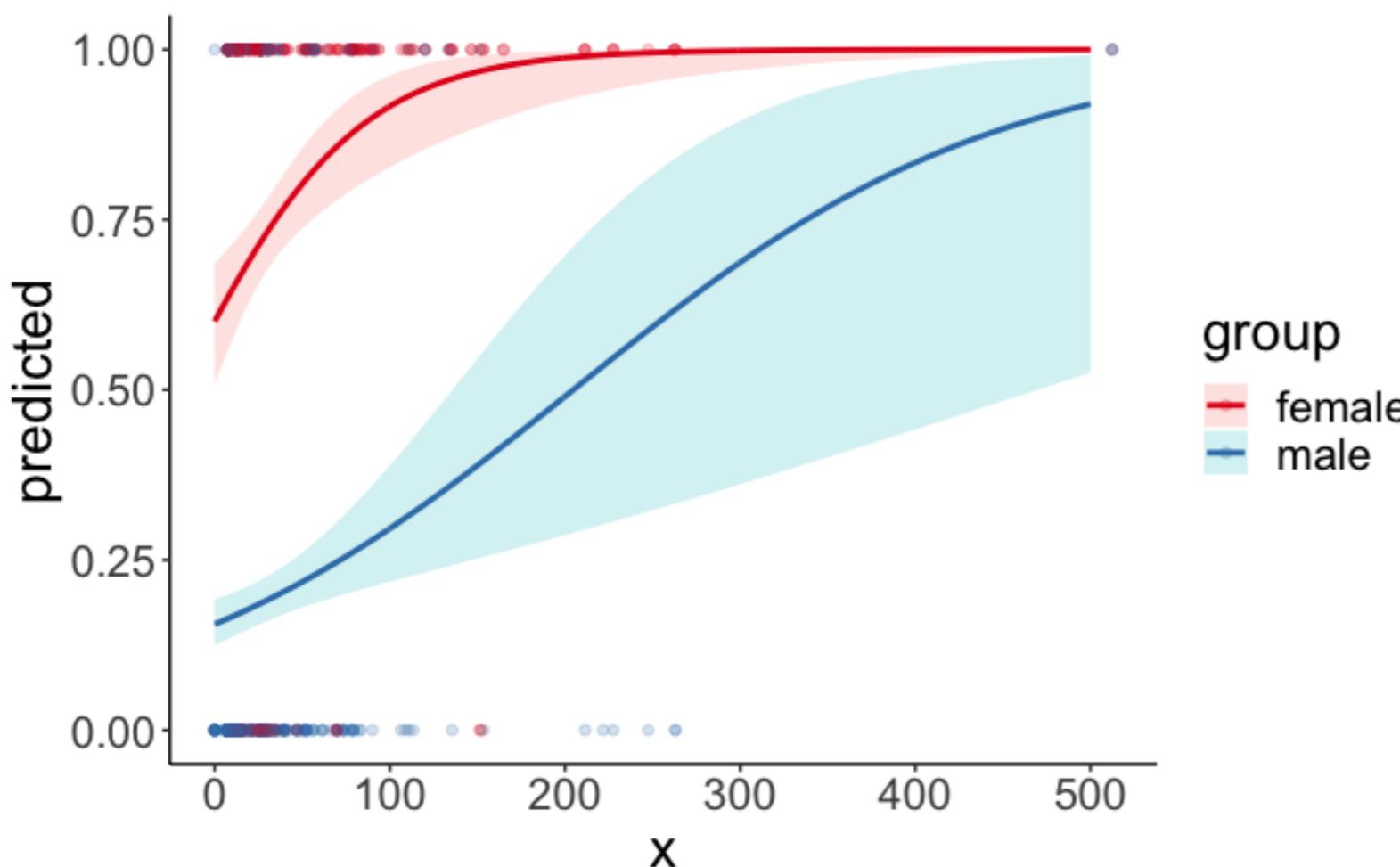
we need to specify that we
want a likelihood ratio test

Analysis of Deviance Table						
Model 1: survived ~ 1 + fare						
Model 2: survived ~ 1 + sex + fare						
Resid.	Df	Resid.	Dev	Df	Deviance	Pr(>Chi)
1	889	1117.57				
2	888	884.31	1	233.26	< 2.2e-16	***

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

Reporting results

- Visualize the data
- Show a table with the regression results
- Report significance of different factors
- Interpreting parameter estimates is tricky -- probably best to report probabilities for a few example cases



# Predicted values of survived				
# x = fare				
# sex = female				
x	Predicted	SE	95% CI	
0	0.60	0.19	[0.51,	0.69]
100	0.92	0.42	[0.83,	0.96]
200	0.99	0.95	[0.93,	1.00]
300	1.00	1.48	[0.97,	1.00]
400	1.00	2.02	[0.99,	1.00]
500	1.00	2.55	[1.00,	1.00]
# sex = male				
x	Predicted	SE	95% CI	
0	0.16	0.13	[0.12,	0.19]
100	0.30	0.21	[0.22,	0.39]
200	0.49	0.44	[0.29,	0.70]
300	0.69	0.69	[0.36,	0.90]
400	0.83	0.94	[0.44,	0.97]
500	0.92	1.19	[0.53,	0.99]

Assumptions

- linearity (between predictors and log odds)
- independence
- no multi-collinearity
- model fails to converge when there is **complete separation**:
 - if outcome variable can be perfectly predicted by a (combination of) predictor(s)

Different kinds of generalized models

Different linking functions

```
binomial(link = "logit")  
  
gaussian(link = "identity")  
  
Gamma(link = "inverse")  
  
inverse.gaussian(link = "1/mu^2")  
  
poisson(link = "log")  
  
quasi(link = "identity", variance = "constant")  
  
quasibinomial(link = "logit")  
  
quasipoisson(link = "log")
```

**apply different transformations to the
dependent variable**

Mixed effects logistic regression

Mixed effects logistic regression

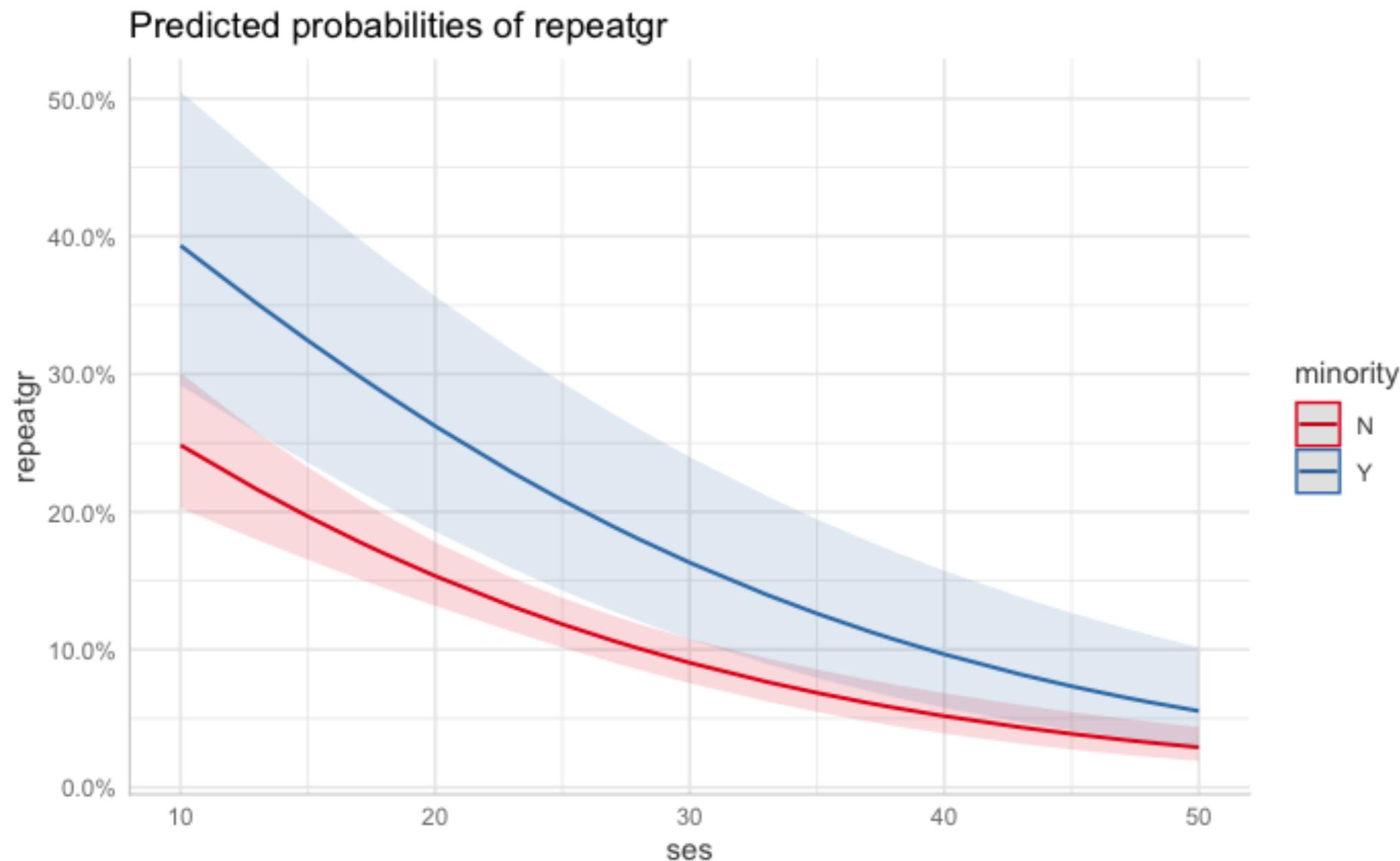
repeated a grade: yes / no

```
1 fit = glmer(repeatgr ~ 1 + ses * Minority + (1 | schoolNR),  
2               data = df.language,  
3               family = "binomial")  
4  
5 fit %>% summary()
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']  
Family: binomial ( logit )  
Formula: repeatgr ~ 1 + ses + minority + (1 | school_nr)  
Data: df.language  
  
AIC      BIC      logLik deviance df.resid  
1659.1  1682.1   -825.6    1651.1     2279  
  
Scaled residuals:  
    Min      1Q  Median      3Q      Max  
-0.9235 -0.4045 -0.3150 -0.2249  5.8372  
  
Random effects:  
 Groups   Name        Variance Std.Dev.  
 school_nr (Intercept) 0.2489   0.4989  
Number of obs: 2283, groups: school_nr, 131  
  
Fixed effects:  
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.506291  0.197570 -2.563  0.01039 *  
ses         -0.060086  0.007524 -7.986 1.39e-15 ***  
minorityY    0.673612  0.238660  2.822  0.00477 **  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Correlation of Fixed Effects:  
          (Intr) ses  
ses       -0.898  
minorityY -0.308  0.208
```

Mixed effects logistic regression

```
1 ggpredict(model = fit,  
2            terms = c("ses [all]", "minority")) %>%  
3   plot()
```



Feedback

How was the pace of today's class?

much a little just a little much
too too right too too
slow slow

How happy were you with today's class overall?



What did you like about today's class? What could be improved next time?

Thank you!