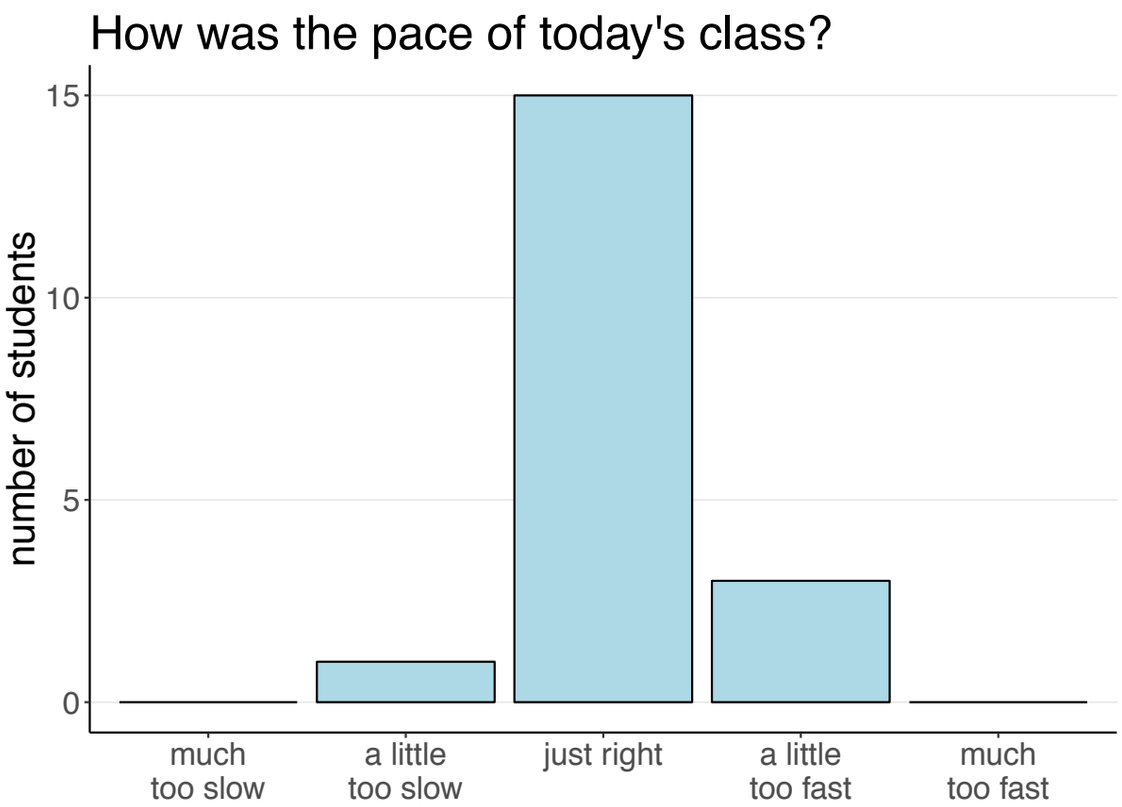
Generalized linear model

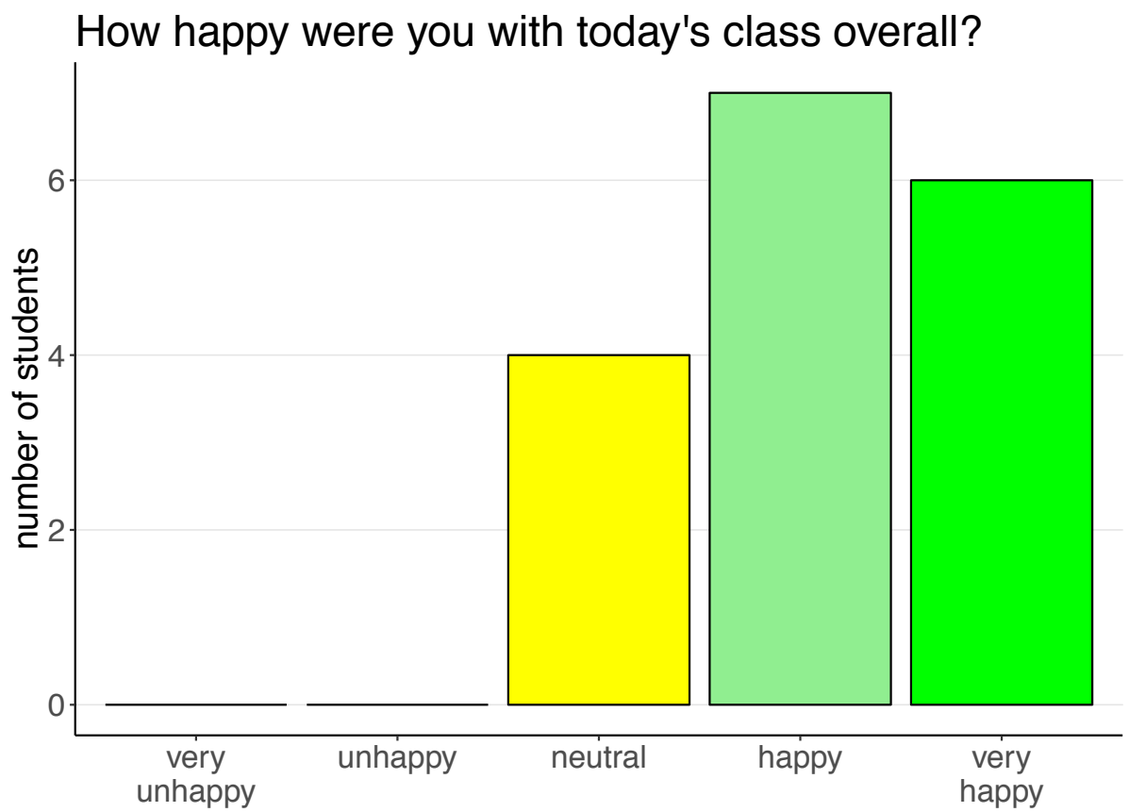
03/01/2019

Logistics

Your feedback

Your feedback





Things that came up

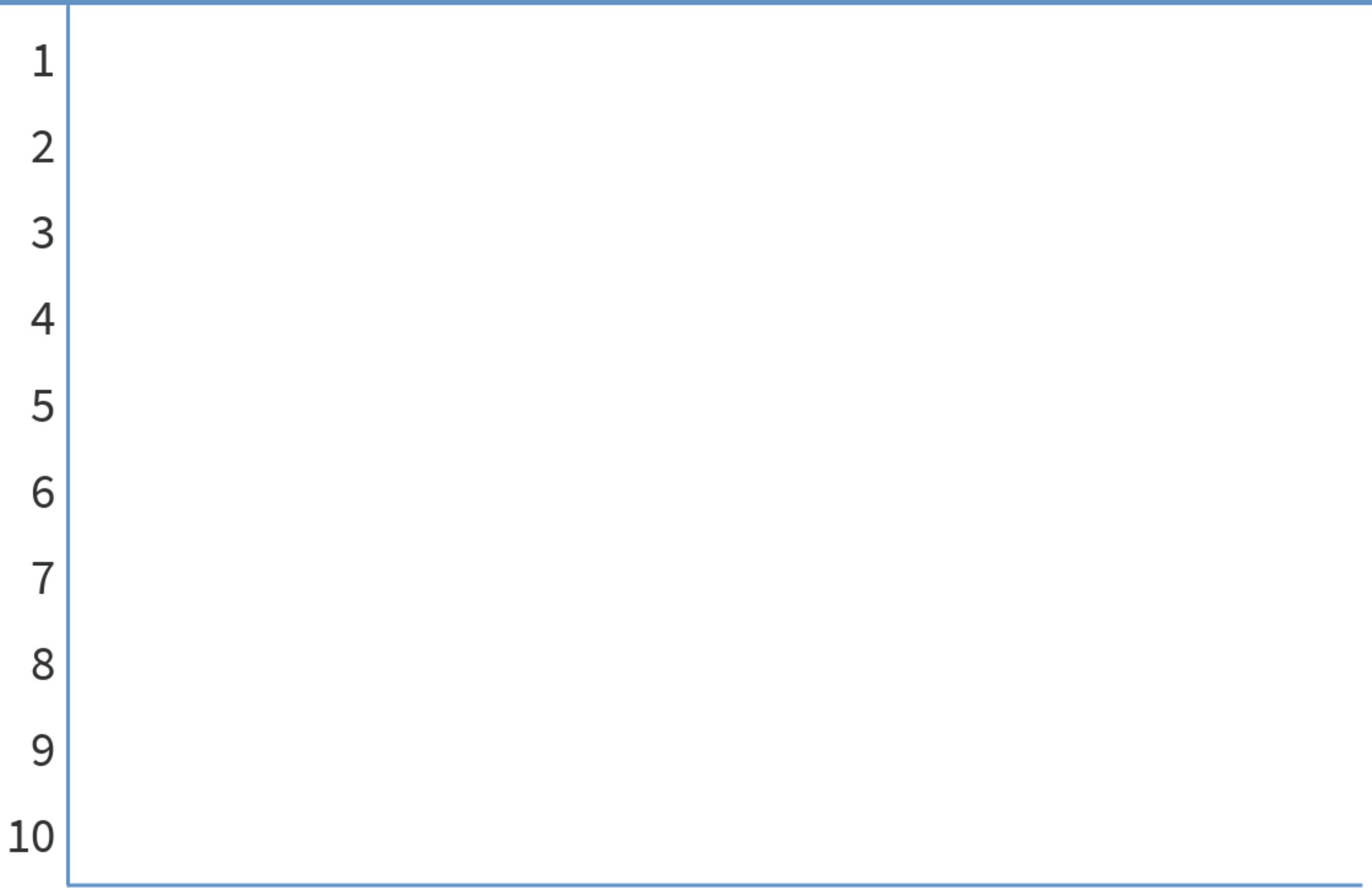
Things that came up

I am uncertain why one would choose no slope or no intercept for an lmer. Can you go over the reasons for choosing (1 + var | other_var), (1 | other_var), and (0 + var | other_var)?

yes, we'll discuss this in this lecture

Homework 5

How many hours did it take you to complete Homework 5?



Plan for today

- Linear mixed effects model
  - Getting p-values
  - Pitfalls in fitting **`lmer`**`()`s (and what to do about it)
  - Understanding **`lmer`**`()` syntax
- Generalized linear model
  - logistic regression
  - mixed effects logistic regression

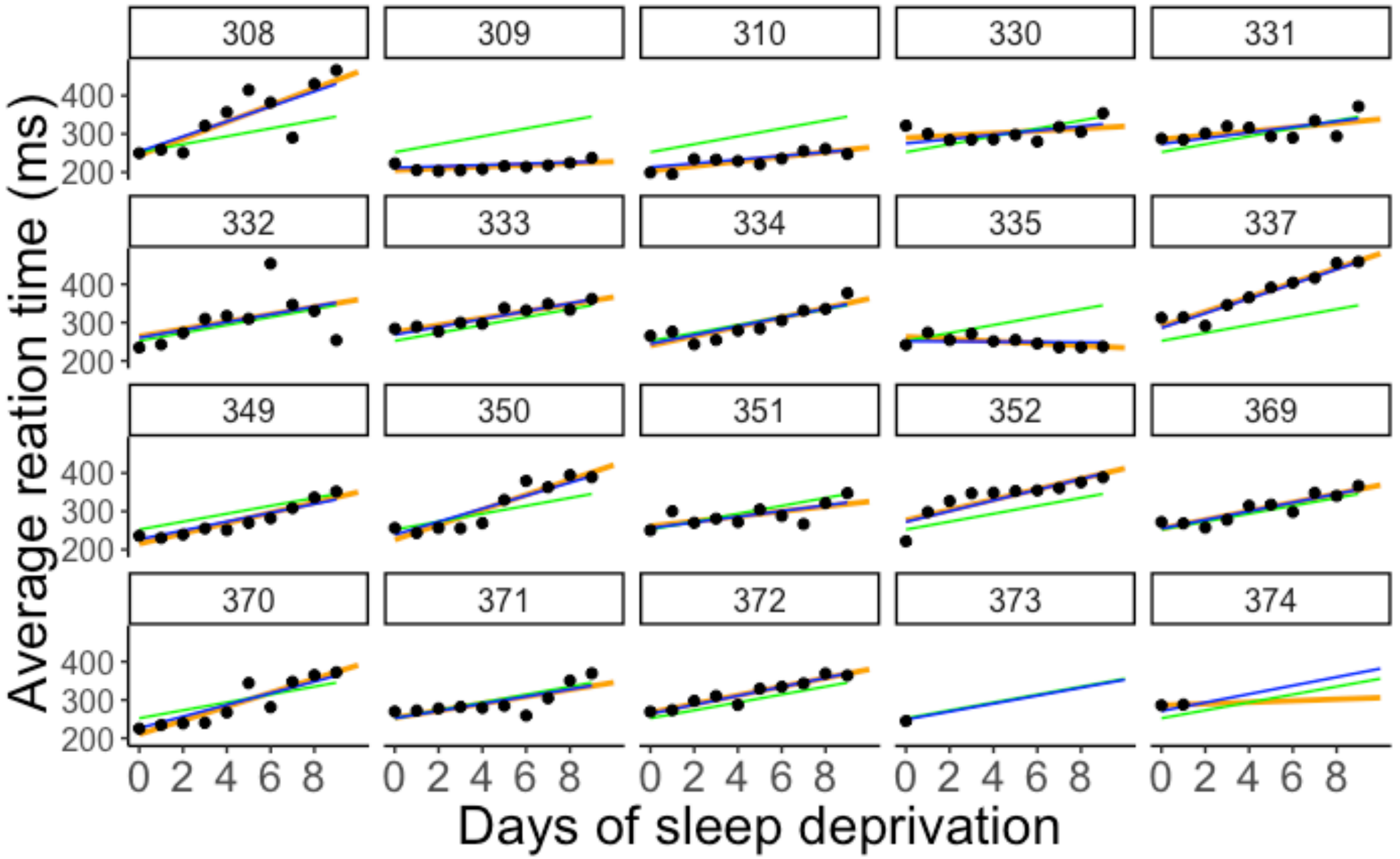
Quick reminder

Pooling and shrinkage

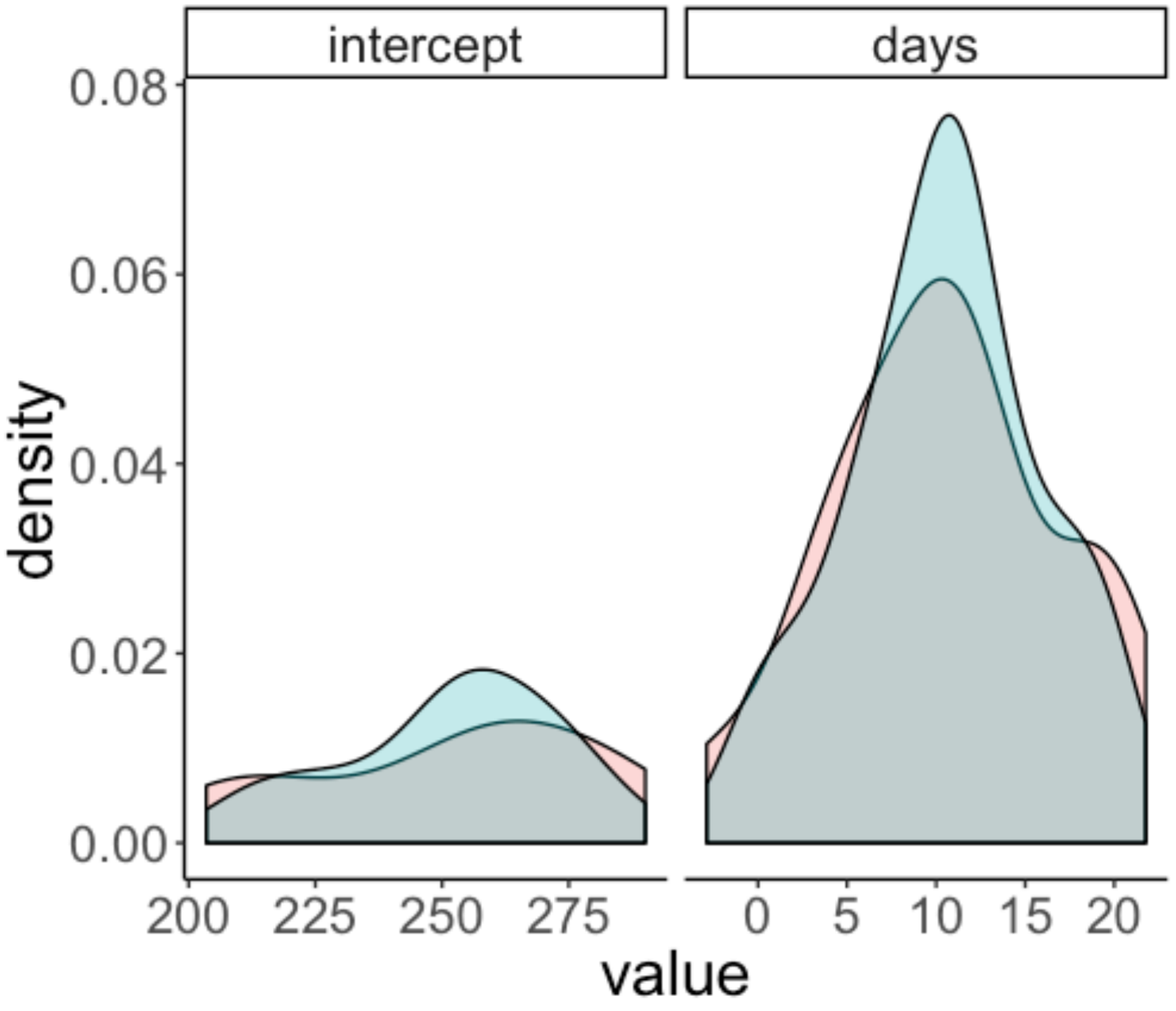
complete pooling

no pooling

partial pooling



Pooling and shrinkage

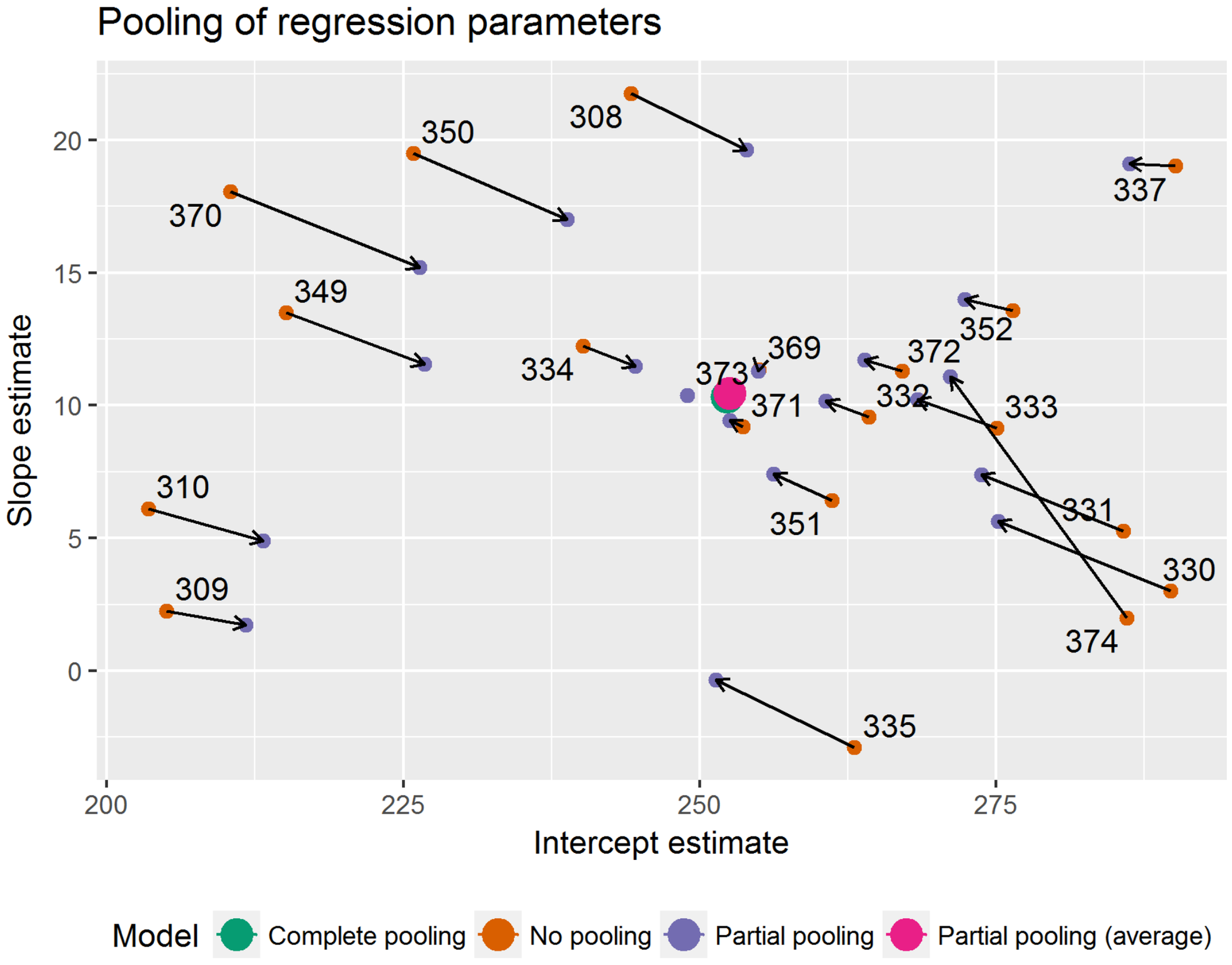


standard deviation

method	intercept	days
no pooling	28.95	6.56
partial pooling	21.59	5.46

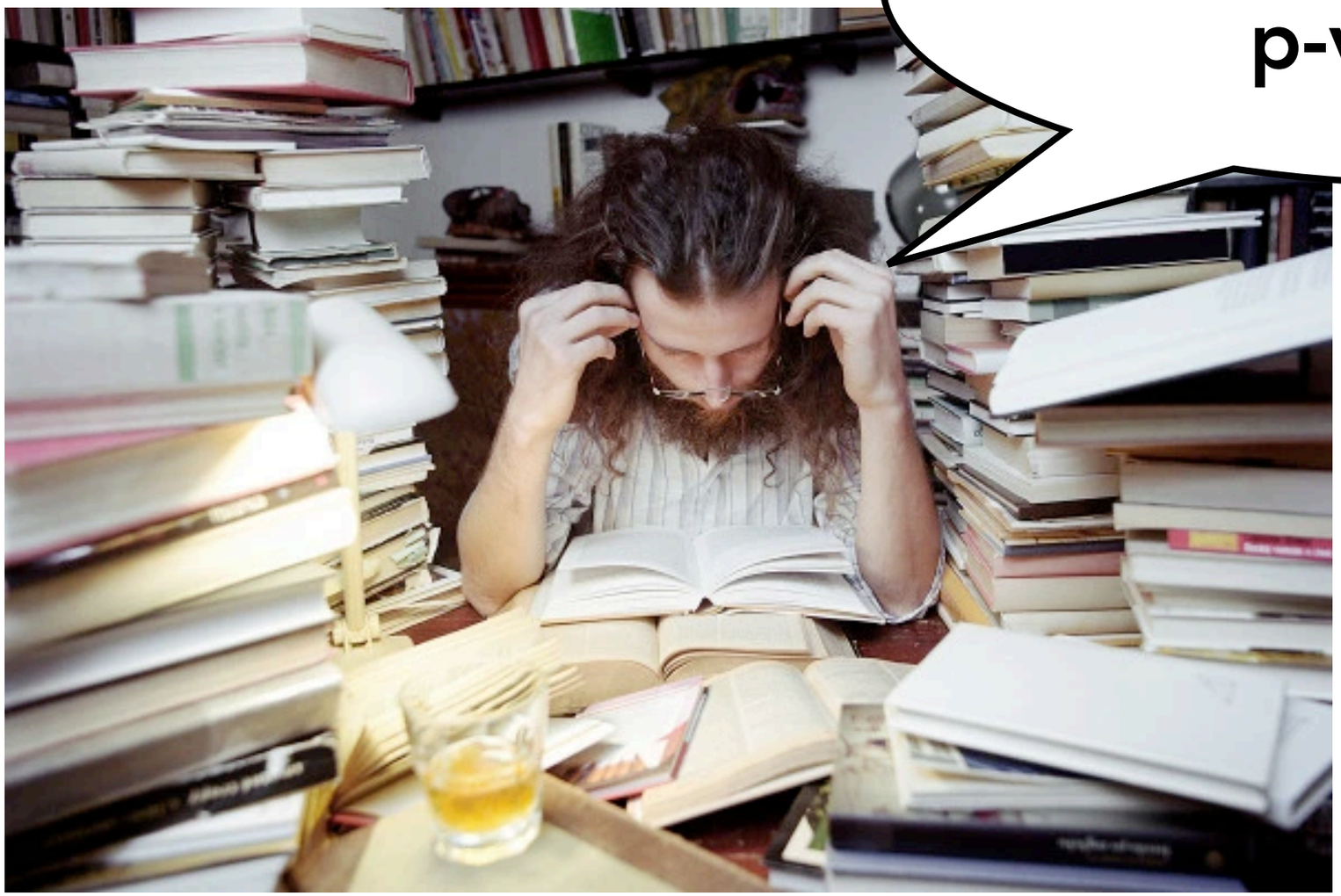
variance "shrinks"

Pooling and shrinkage



Getting p-values

Reviewer #2



Model comparison

we can still do our good ol model comparison trick

```
1 # fit models
2 fit.compact = lmer(formula = value ~ 1 + (1 | participant),
3                    data = df.original)
4 fit.augmented = lmer(formula = value ~ 1 + condition + (1 | participant),
5                    data = df.original)
6
7 # compare via Chisq-test
8 anova(fit.compact, fit.augmented)
```

```
refitting model(s) with ML (instead of REML)
Data: df.original
Models:
fit.compact: value ~ 1 + (1 | participant)
fit.augmented: value ~ 1 + condition + (1 | participant)
              Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
fit.compact    3 53.315 58.382 -23.6575   47.315
fit.augmented  4 17.849 24.605  -4.9247    9.849 37.466      1  9.304e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

`"lmerTest"` package

- I recommend **not** to load the package as it doesn't play nicely with the "broom" package at the moment (this will presumably be fixed soon though ...)

"lmerTest" package

```
1 lmerTest::lmer(formula = reaction ~ 1 + days + (1 + days | subject),
2                data = df.sleep) %>%
3   summary()
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
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      df t value Pr(>|t|)
(Intercept)  252.543      6.433  19.294  39.256  < 2e-16 ***
days          10.452      1.542  17.163   6.778 3.06e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

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

Pitfalls in fitting `lmer()`s

My model does not converge ...

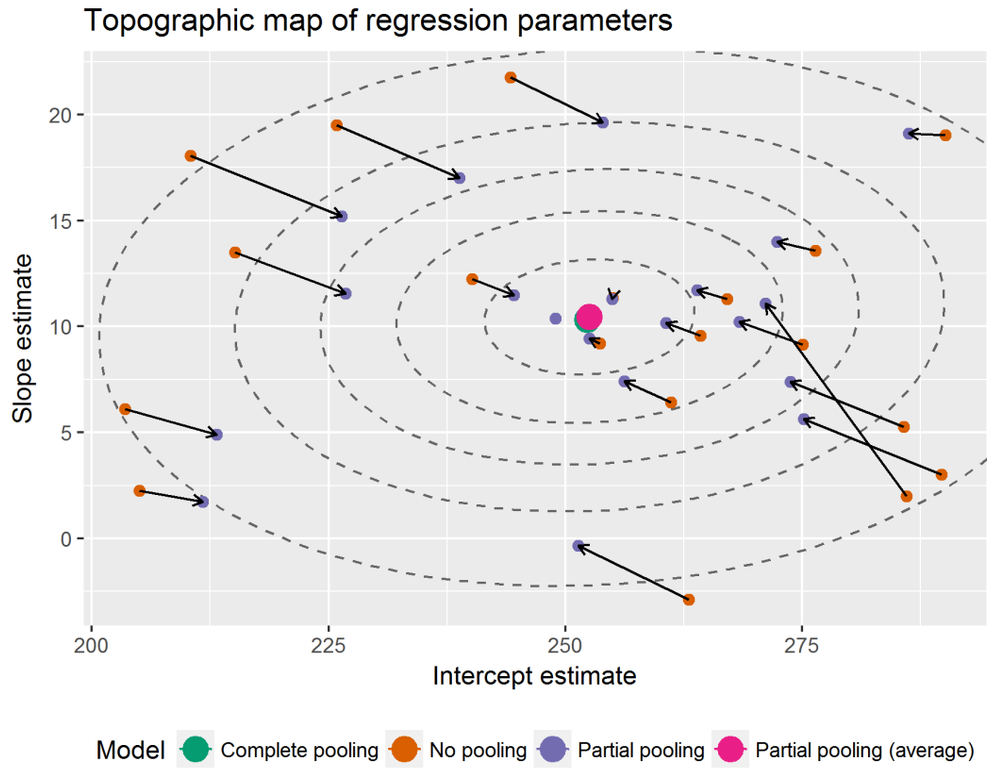
- **`lmer`**`()`s are solved through a (complicated) process of iterative optimization
- only interpret the results of models that actually converged!
- here are some tricks that might help:
  - *continuous predictors*: center and scale
  - *categorical predictors*: choose a factor that has more data as your reference level
  - remove the correlation component from your model

Remove the correlation component from your model

```
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  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 -0.137
```

multivariate Gaussian



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()
```

random slopes

random intercepts

```
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 -0.184
```

independent Gaussians

Remove the correlation component from your model

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

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

```
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 -0.184
```

independent Gaussians

My model does not 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

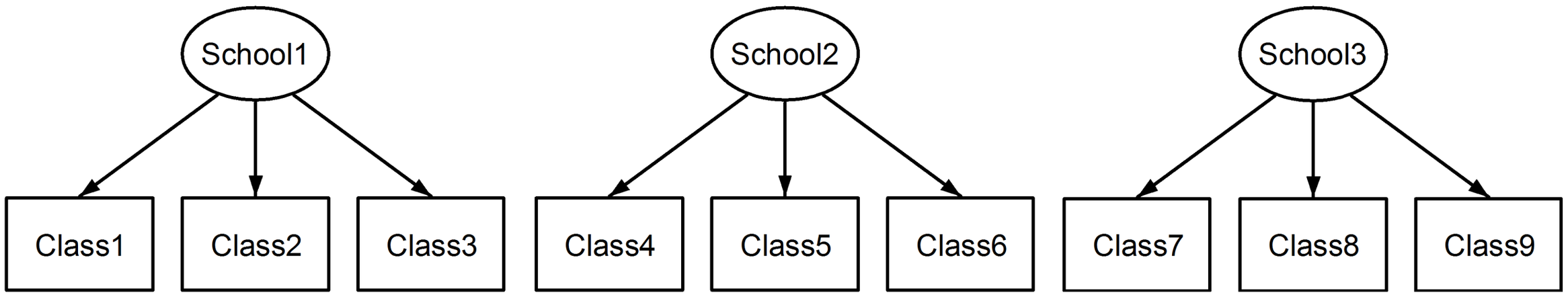
```
$fixef
                               (Intercept)     days
bobyqa                            252.5426 10.45212
Nelder_Mead                       252.5426 10.45212
nlminbwrap                        252.5426 10.45212
nloptwrap.NLOPT_LN_NELDERMEAD     252.5426 10.45212
nloptwrap.NLOPT_LN_BOBYQA    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

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

$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

Understanding `lmer()` syntax

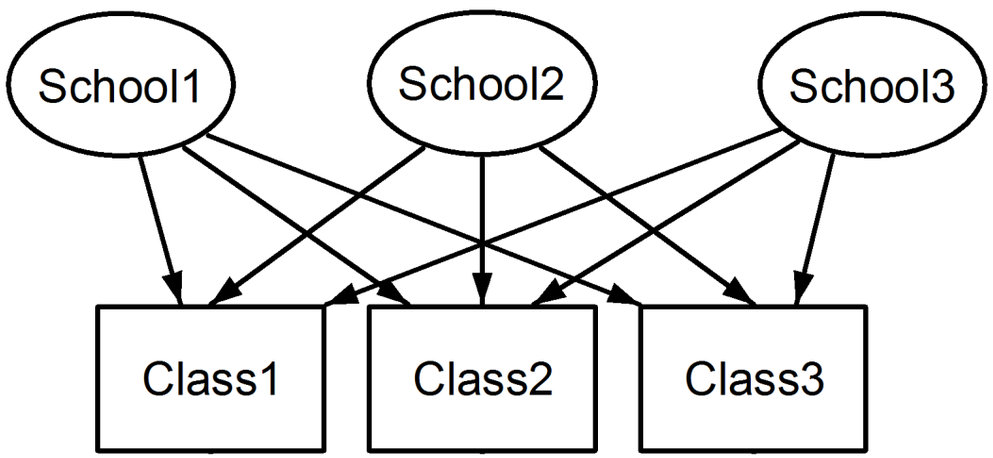
Multi-level models

nested `(1|School/Class)`



each class only appears within one school

crossed `(1|School) + (1|Class)`

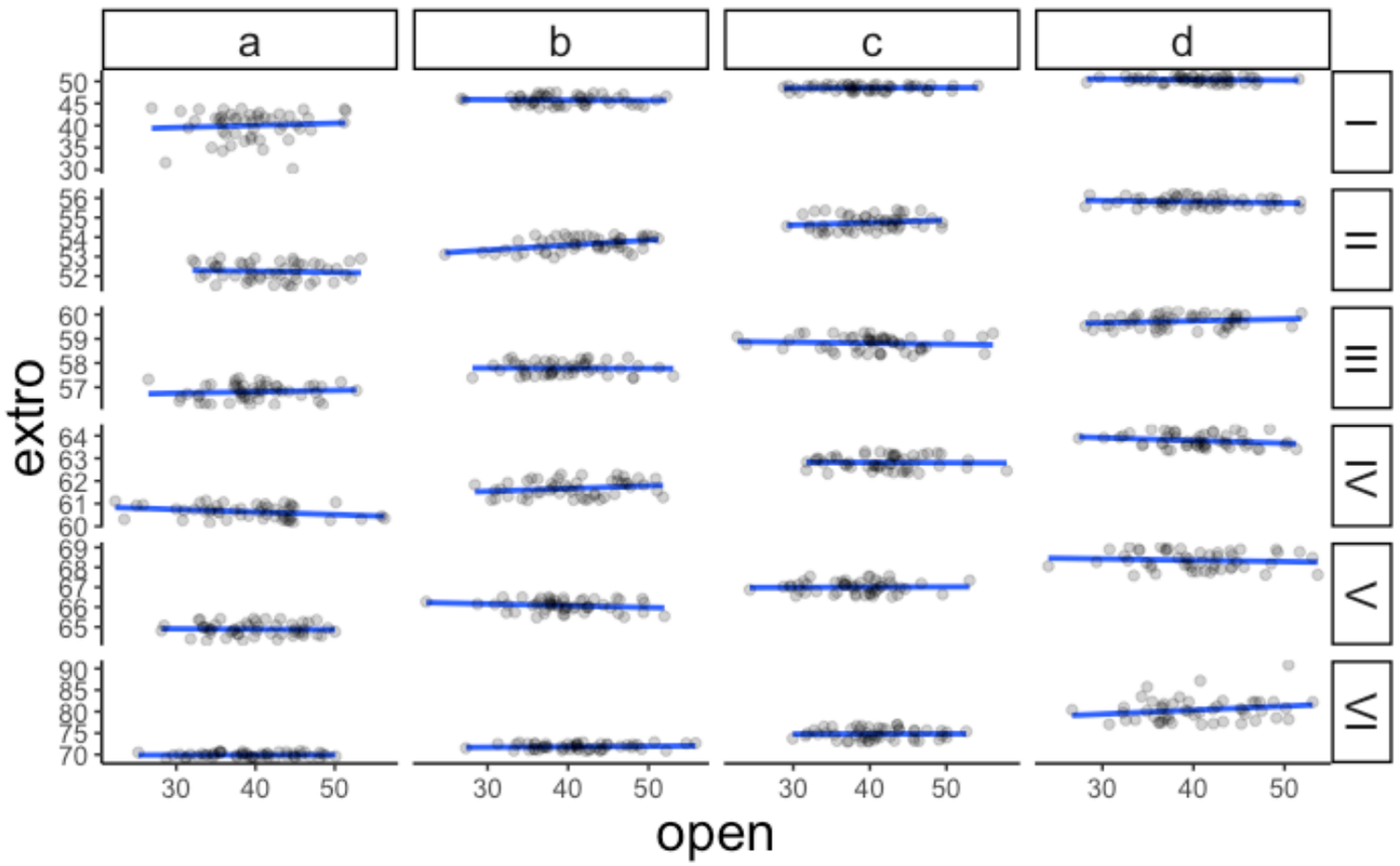


each class appears in each of the schools

Multi-level models

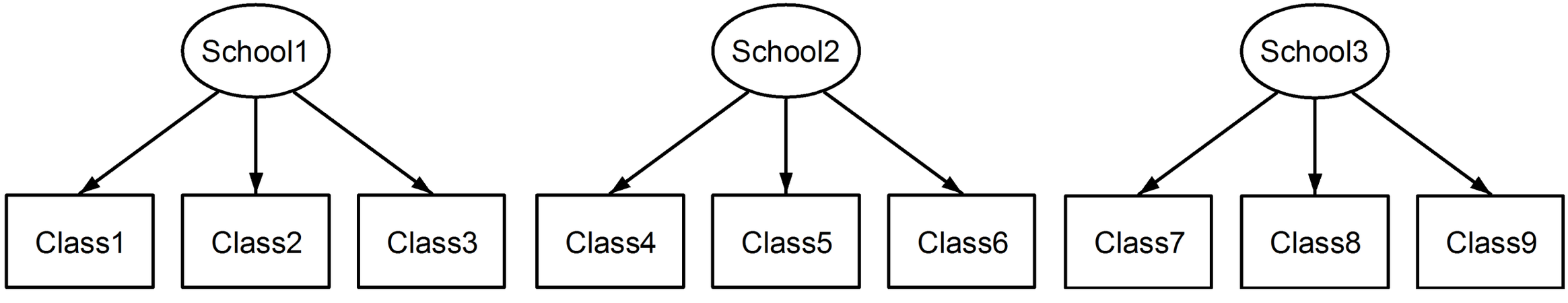
	id	extro	open	agree	social	class	school
1	1	63.69356	43.43306	38.02668	75.05811	d	IV
2	2	69.48244	46.86979	31.48957	98.12560	a	VI
3	3	79.74006	32.27013	40.20866	116.33897	d	VI
4	4	62.96674	44.40790	30.50866	90.46888	c	IV
5	5	64.24582	36.86337	37.43949	98.51873	d	IV
6	6	50.97107	46.25627	38.83196	75.21992	d	I
7	7	60.14740	37.04243	38.55959	95.91299	d	III
8	8	64.17886	42.16530	34.88235	91.45257	d	IV
9	9	56.67670	32.84933	31.68027	115.25167	a	III
10	10	47.23914	44.25764	24.99970	122.70848	b	I

relationship between openness and extraversion



Multi-level models

nested `(1|School/Class)`

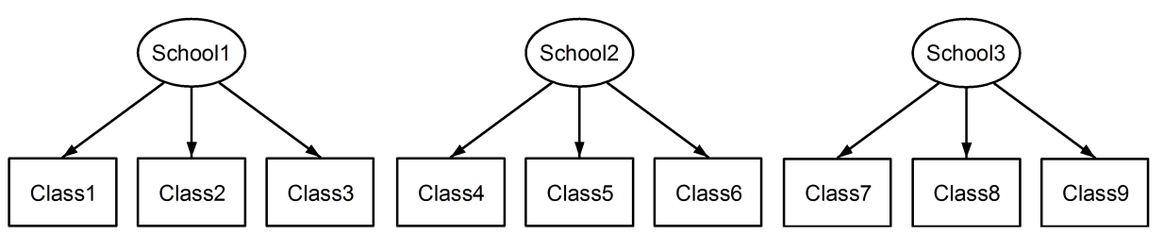



```
1 # fit nested model
2 fit.nested = lmer(extro ~ open + agree + social + (1 | school/class), data = df.school)
3
4 # print model summary
5 fit.nested %>% summary()
6
7 # model coefficients
8 fit.nested %>% coef()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: extro ~ open + agree + social + (1 | school/class)
   Data: df.school

REML criterion at convergence: 3554.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-9.9949 -0.3348  0.0057  0.3394 10.6476

Random effects:
 Groups       Name        Variance Std.Dev.
 class:school (Intercept)  8.2046  2.8644
 school       (Intercept) 93.8433  9.6873
 Residual                  0.9684  0.9841
Number of obs: 1200, groups:  class:school, 24; school, 6
```

Multi-level models

nested `(1|School/Class)`



random intercepts of class within each school

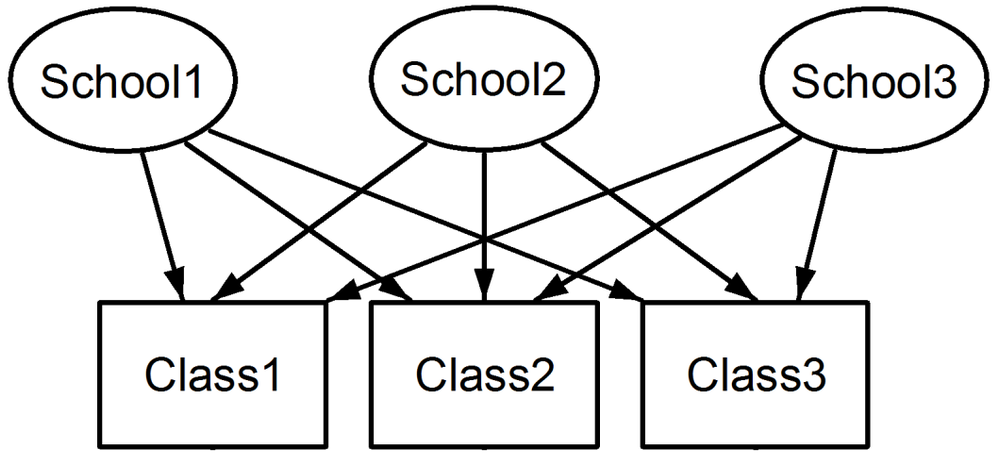
random intercepts of schools

random intercepts

```
$`class:school`
      (Intercept)        open       agree       social
a:I      53.77106 0.006106514 -0.007665927 0.0005404069
a:II     58.23842 0.006106514 -0.007665927 0.0005404069
a:III    58.71819 0.006106514 -0.007665927 0.0005404069
a:IV     58.68813 0.006106514 -0.007665927 0.0005404069
a:V      58.68268 0.006106514 -0.007665927 0.0005404069
a:VI     56.23088 0.006106514 -0.007665927 0.0005404069
b:I      59.54852 0.006106514 -0.007665927 0.0005404069
b:II     59.62643 0.006106514 -0.007665927 0.0005404069
b:III    59.70219 0.006106514 -0.007665927 0.0005404069
b:IV     59.73276 0.006106514 -0.007665927 0.0005404069
b:V      59.87036 0.006106514 -0.007665927 0.0005404069
b:VI     58.14865 0.006106514 -0.007665927 0.0005404069
c:I      62.28460 0.006106514 -0.007665927 0.0005404069
c:II     60.74743 0.006106514 -0.007665927 0.0005404069
c:III    60.70970 0.006106514 -0.007665927 0.0005404069
c:IV     60.86062 0.006106514 -0.007665927 0.0005404069
c:V      60.80225 0.006106514 -0.007665927 0.0005404069
c:VI     61.10164 0.006106514 -0.007665927 0.0005404069
d:I      64.14113 0.006106514 -0.007665927 0.0005404069
d:II     61.81189 0.006106514 -0.007665927 0.0005404069
d:III    61.65165 0.006106514 -0.007665927 0.0005404069
d:IV     61.83703 0.006106514 -0.007665927 0.0005404069
d:V      62.13593 0.006106514 -0.007665927 0.0005404069
d:VI     66.66561 0.006106514 -0.007665927 0.0005404069

$school
    (Intercept)        open       agree       social
I      46.44407 0.006106514 -0.007665927 0.0005404069
II     54.20862 0.006106514 -0.007665927 0.0005404069
III    58.29847 0.006106514 -0.007665927 0.0005404069
IV     62.15074 0.006106514 -0.007665927 0.0005404069
V      66.41348 0.006106514 -0.007665927 0.0005404069
VI     73.91156 0.006106514 -0.007665927 0.0005404069
```

Multi-level models

crossed `(1|School) + (1|Class)`



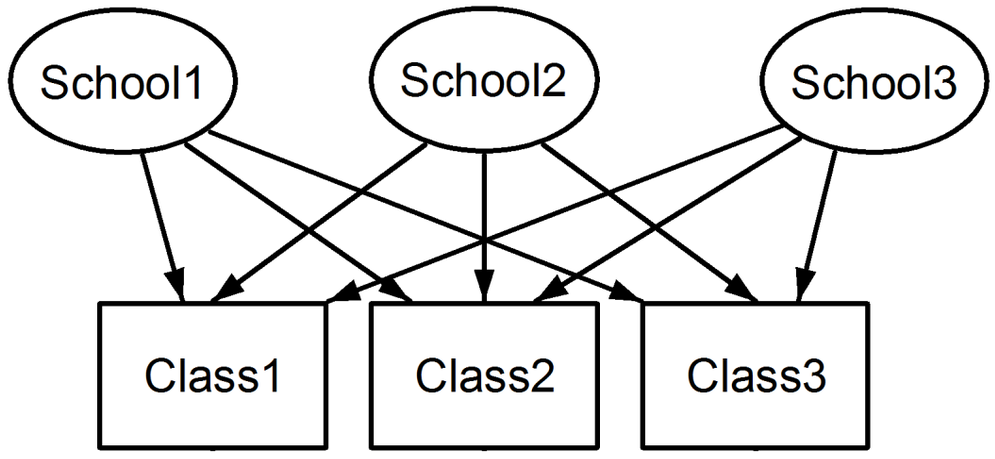
each class appears in each of the schools

```
1 # fit crossed model
2 fit.crossed = lmer(extro ~ open + agree + social + (1 | school) + (1 | class), data = df.school)
3
4 # print model summary
5 fit.crossed %>% summary()
6
7 # model coefficients
8 fit.crossed %>% coef()
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: extro ~ open + agree + social + (1 | school) + (1 | class)
   Data: df.school

REML criterion at convergence: 4723.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-7.8677 -0.5421  0.0101  0.5218  8.2282

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 95.914   9.794
 class    (Intercept)  5.787   2.406
 Residual              2.787   1.669
Number of obs: 1200, groups:  school, 6; class, 4
```

Multi-level models

crossed `(1|School) + (1|Class)`



each class is in
each of the schools

random intercepts

random intercepts of school →

random intercepts of class →

```
$school
    (Intercept)        open         agree        social
I      46.10663 0.01083374 -0.005420032 -0.001761963
II     54.02956 0.01083374 -0.005420032 -0.001761963
III    58.22277 0.01083374 -0.005420032 -0.001761963
IV     62.15508 0.01083374 -0.005420032 -0.001761963
V      66.51062 0.01083374 -0.005420032 -0.001761963
VI     74.16838 0.01083374 -0.005420032 -0.001761963

$class
  (Intercept)        open         agree        social
a    57.35175 0.01083374 -0.005420032 -0.001761963
b    59.39261 0.01083374 -0.005420032 -0.001761963
c    61.04758 0.01083374 -0.005420032 -0.001761963
d    63.00342 0.01083374 -0.005420032 -0.001761963
```

`lmer()` syntax summary

formula	description
`dv ~ x1 + (1 | g)`	Random **intercept** for each level of `g`
`dv ~ x1 + (0 + x1 | g)`	Random **slope** for each level of `g`
`dv ~ x1 + (x1 | g)`	**Correlated** random slope and intercept for each level of `g`
`dv ~ x1 + (x1 || g)`	**Uncorrelated** random slope and intercept for each level of `g`
`dv ~ x1 + (1 | sch) + (1 | tch)`	Random intercept for each level of `sch` and for each level of `tch` (**crossed**)
`dv ~ x1 + (1 | sch/tch)`	Random intercept for each level of `sch` and for each level of `tch` in `sch` (**nested**)

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!

Journal of Memory and Language 68 (2013) 255–278

Contents lists available at SciVerse ScienceDirect

Journal of Memory and Language

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

Random effects structure for confirmatory hypothesis testing: Keep it maximal

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

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

"Through theoretical arguments and Monte Carlo simulation, we show that LMEMs generalize best when they include the maximal random effects structure justified by the design. ...

Maximal LMEMs should be the 'gold standard' for confirmatory hypothesis testing in psycholinguistics and beyond."

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3), 255--278.

What shall I include as random effects?

- Failure to include maximal random-effect structures in LMEMs (when such random effects are present in the underlying populations) **inflates Type I error rates**.
- For designs including within-subjects (or within-items) manipulations, random-intercepts-only LMEMs **can have catastrophically high Type I error rates**, regardless of how p-values are computed from them.
- The performance of a data-driven approach to determining random effects (i.e., model selection) depends strongly on the specific algorithm, size of the sample, and criteria used; **moreover, the power advantage of this approach over maximal models is typically negligible**.
- In terms of power, maximal models perform surprisingly well even in a "worst case" scenario where they assume random slope variation that is actually not present in the population.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3), 255--278.

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

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 (Lily	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

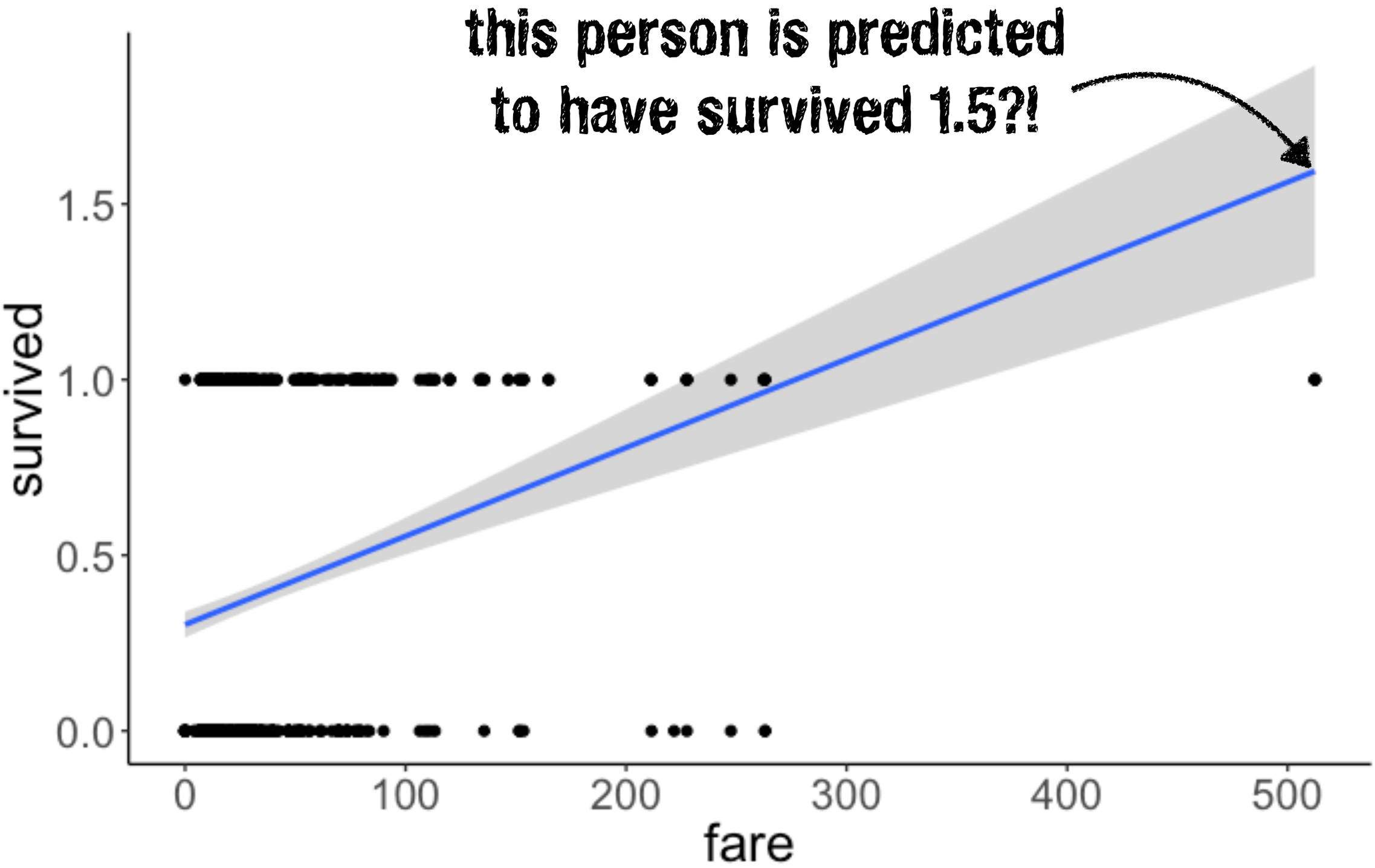
Is there a relationship between `fare` and `survived`?

```
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. malignent
  - yes vs. no
  - ...

Logistic regression

Demo

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

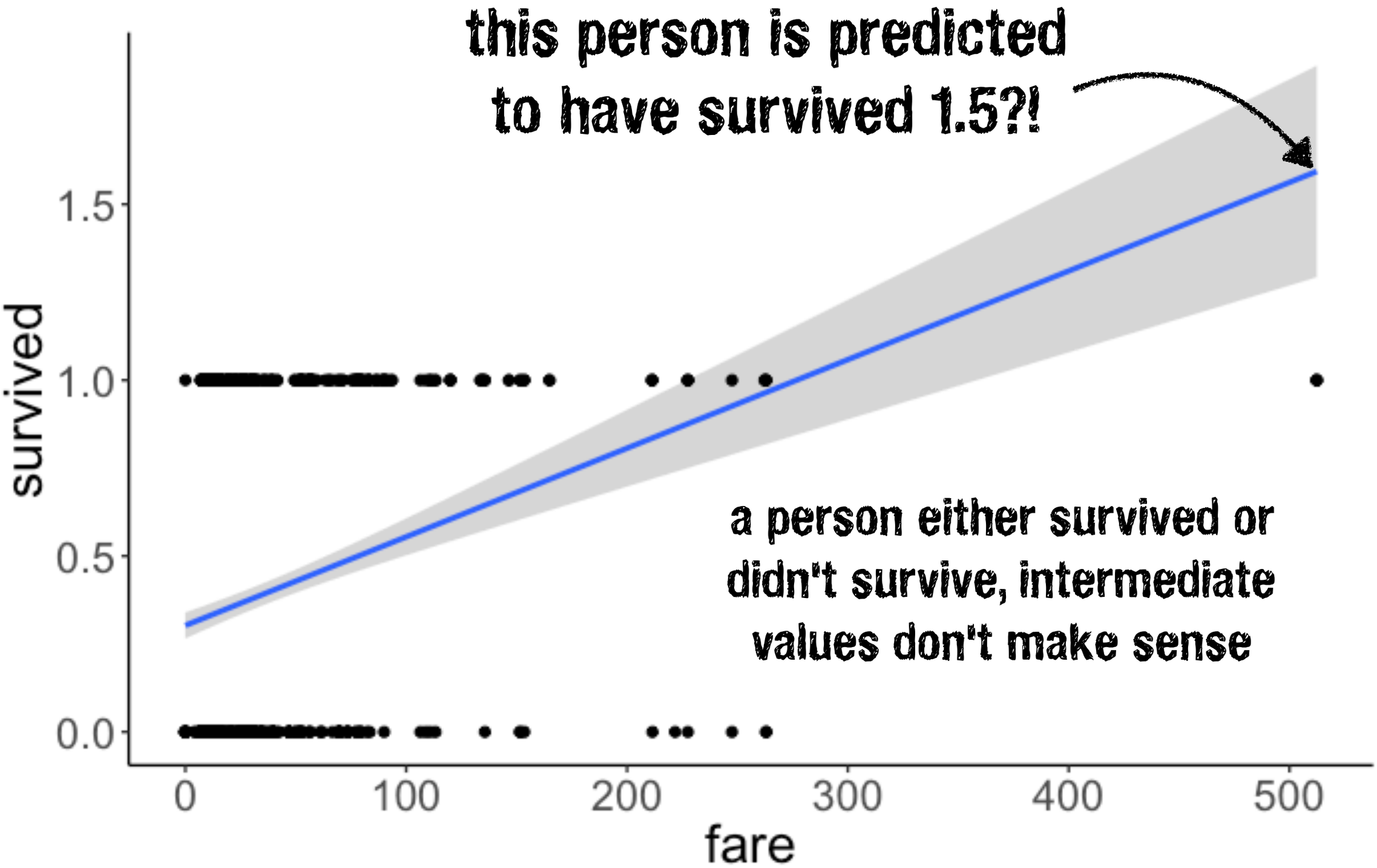
Is there a relationship between `fare` and `survived`?

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

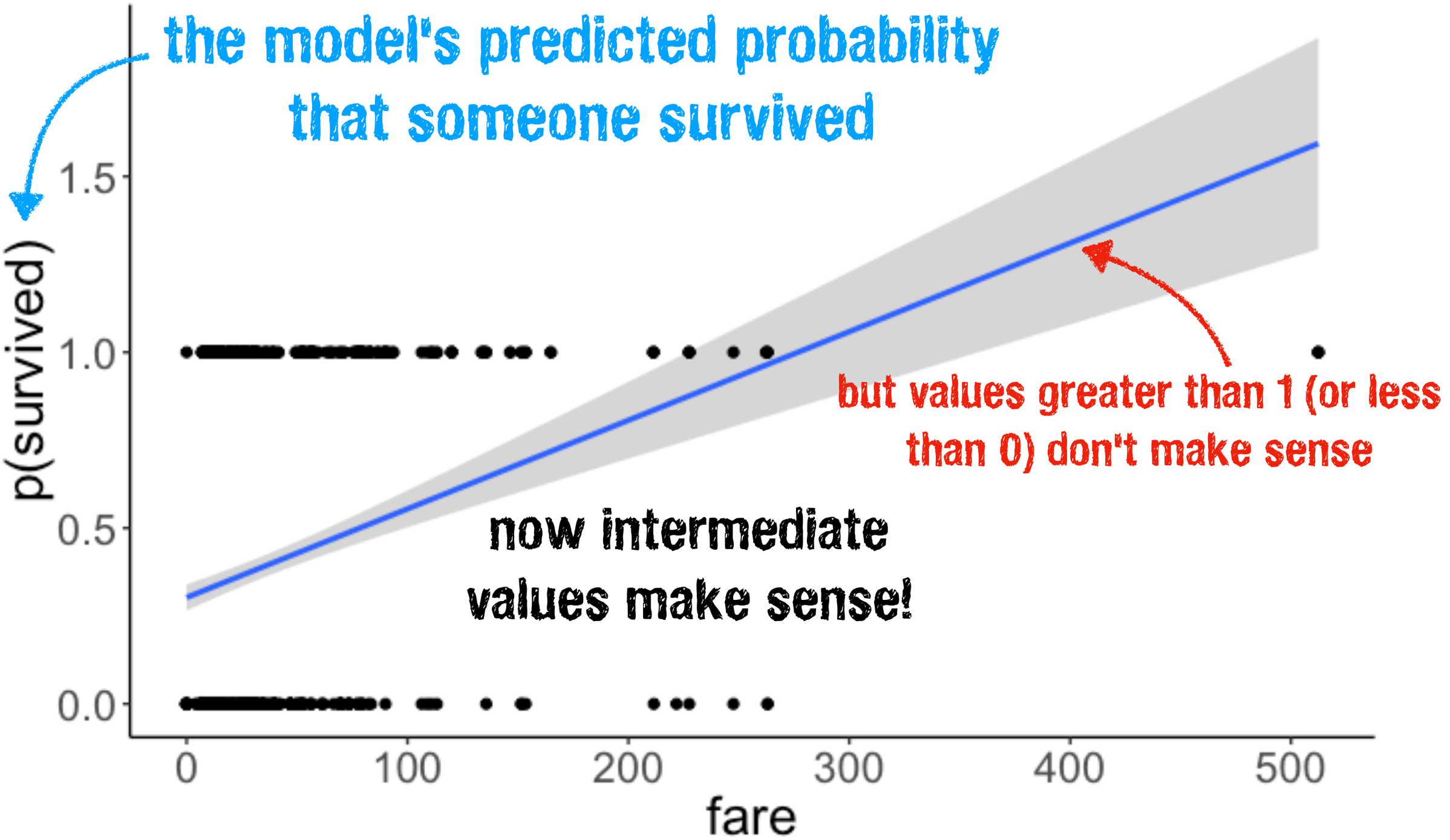
The fact that this class is called "**Generalized linear model**" suggests we can!

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

$$Y_i = b_0 + b_1 \cdot X_i + e_i$$

predict the value of Y

$$\pi_i = b_0 + b_1 \cdot X_i + e_i$$

predict the probability of Y

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

let's just do a **logit** transform

we need to map from **[-∞, +∞]** to **[0, 1]**

Logit transform

$$\pi_i = b_0 + b_1 \cdot X_i + e_i$$

predict the probability of Y

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

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

Step 1: Calculate the "odds"

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

ranges between 0 and +∞

Step 2: Take the (natural) log

ranges between -∞ and +∞

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

Logit transform

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

log odds

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

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

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

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

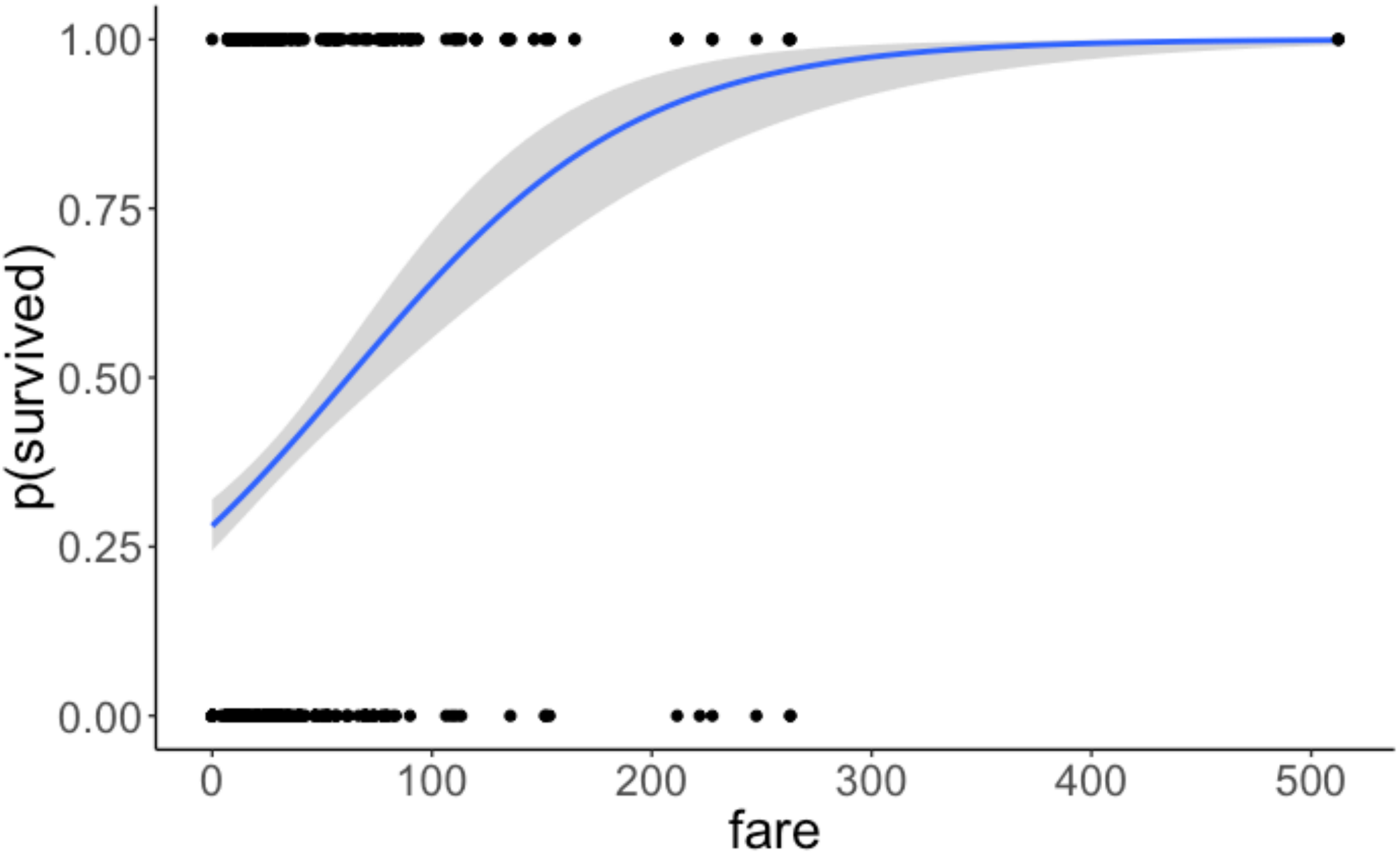
if log odds < 0: $P(Y_i = 1) < P(Y_i = 0)$

Fitting a logistic regression in R

```r
fit.glm = glm(formula = survived ~ 1 + fare,
              family = "binomial",
              data = df.titanic)

fit.glm %>% summary()
```

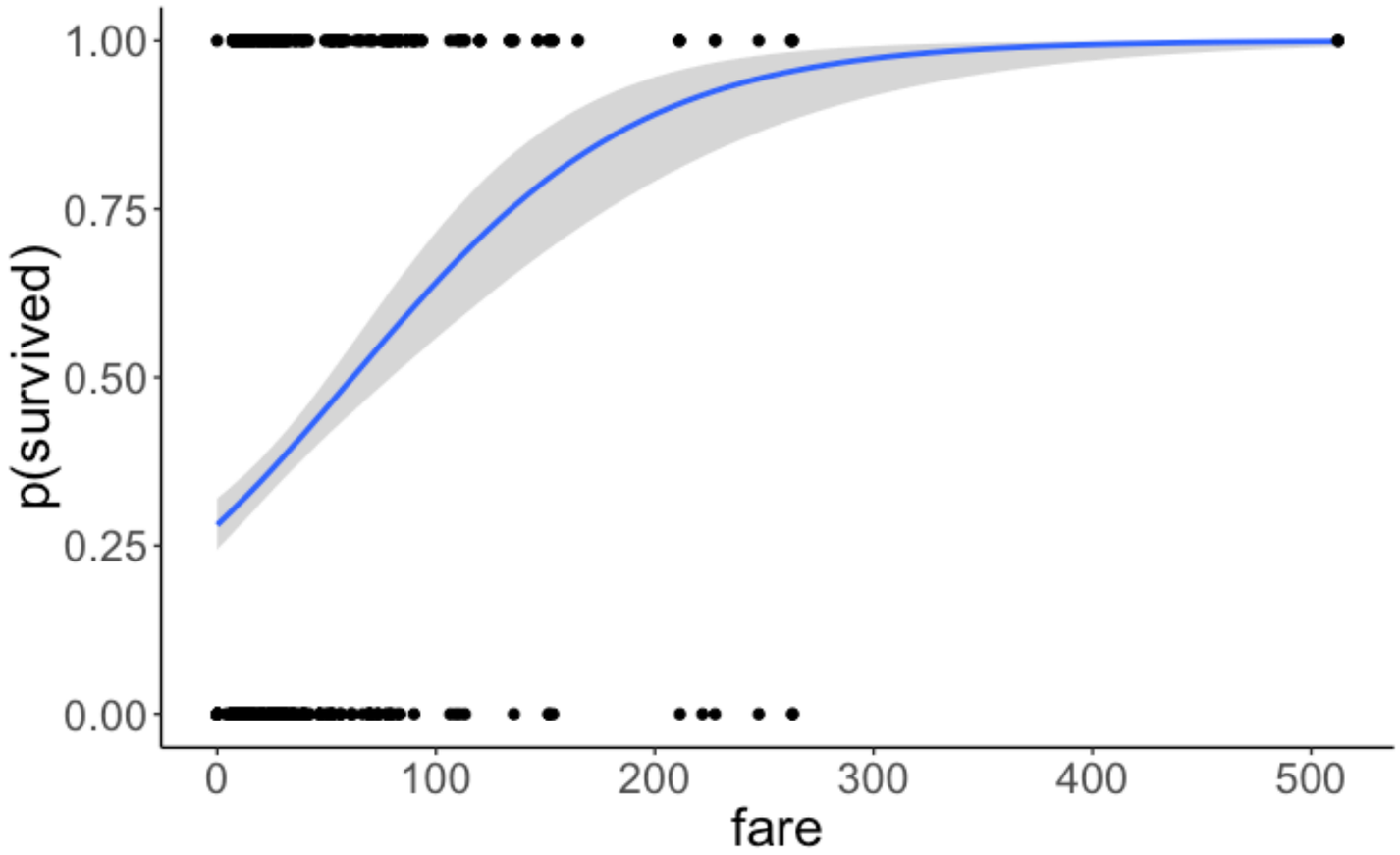
```
Call:
glm(formula = survived ~ 1 + fare, family = "binomial", data = df.titanic)

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



Interpreting the model output

Interpreting the model output




```
Call:
glm(formula = survived ~ 1 + fare,

Deviance Residuals:
    Min       1Q   Median       3Q
-2.4906  -0.8878  -0.8571   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
```

log odds ?

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)

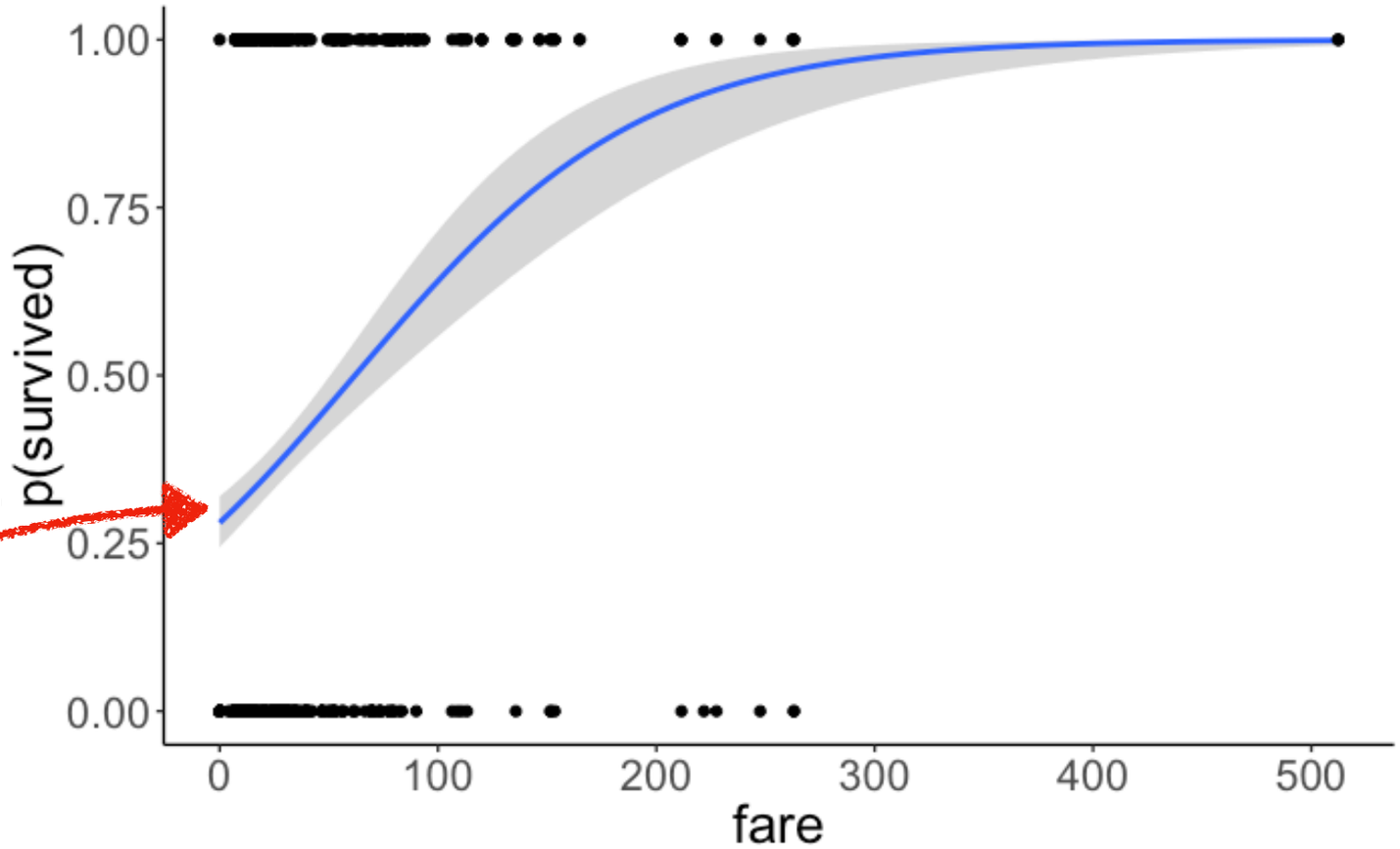
$$\pi_i = \frac{e^{b_0 + b_1 \cdot X_i + e_i}}{1 + e^{b_0 + b_1 \cdot X_i + e_i}}$$

another way to
specify the model

Interpreting the model output

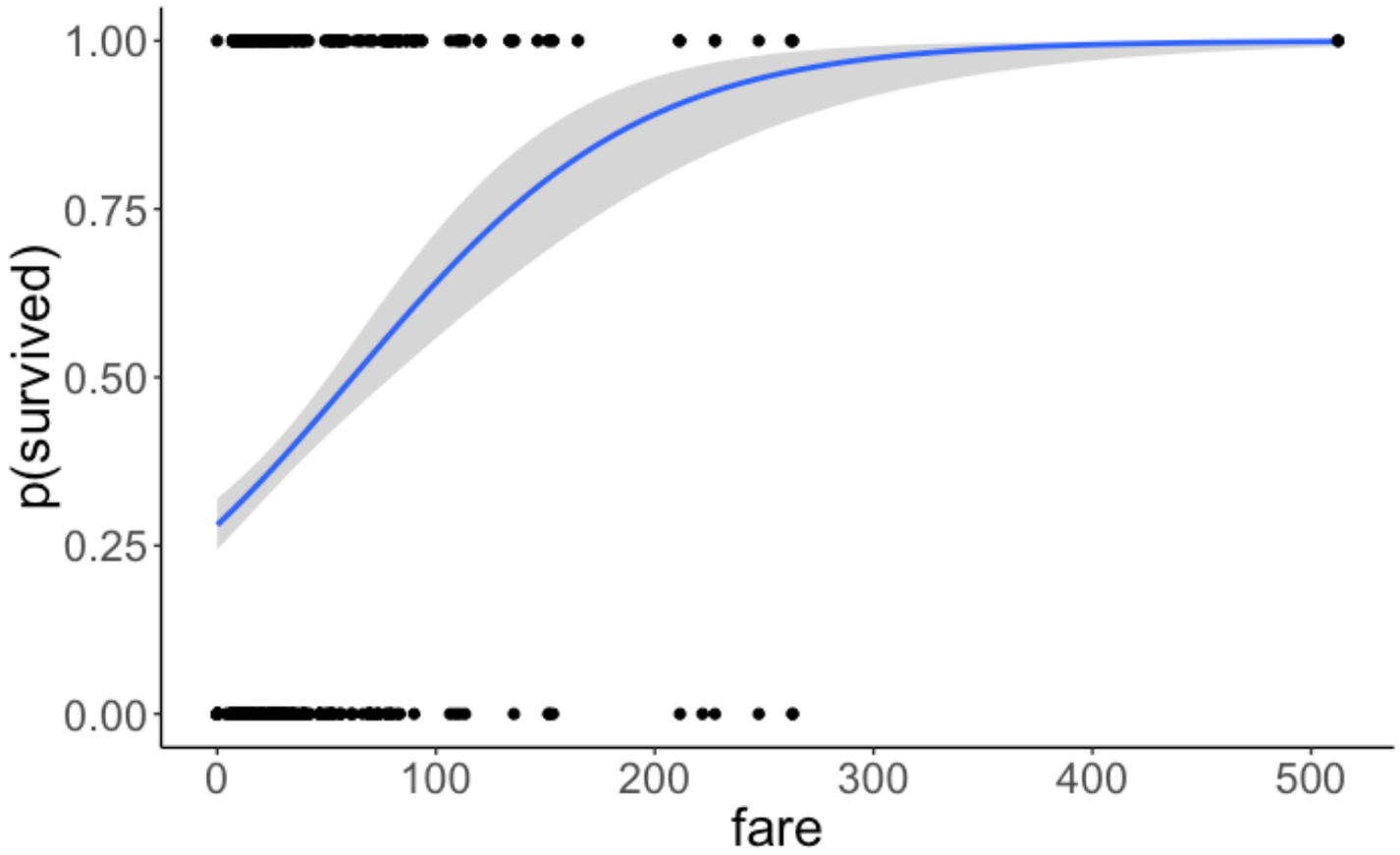
inverse logit

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




```
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
```

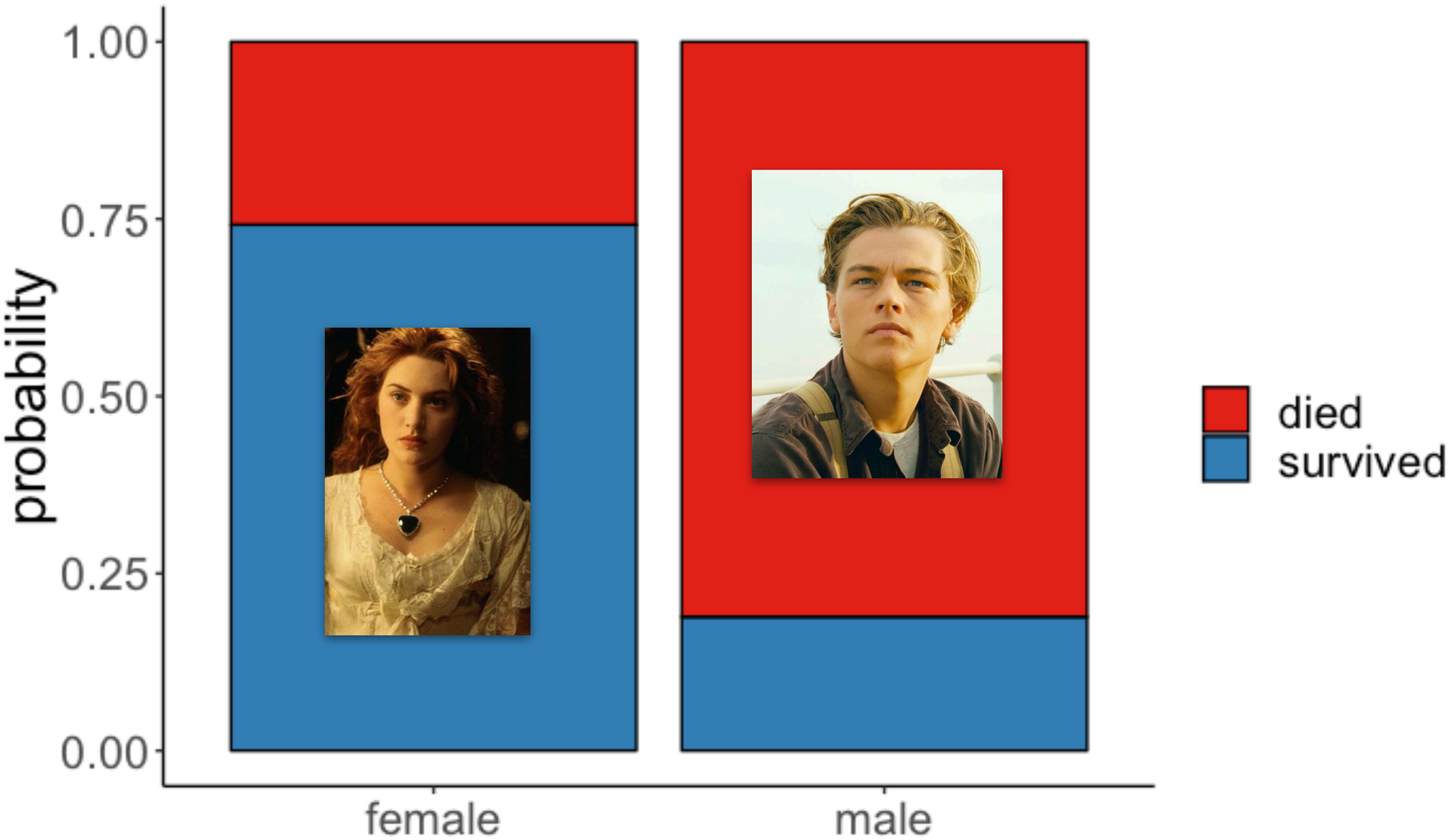
Interpreting the model output




```
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

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
```

sex was significantly associated with survival

Let's consider a binary predictor

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

```
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:

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

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

Let's consider a binary predictor

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

```
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 == 1:

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

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

Now let's go back to a continuous predictor

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

```
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

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

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

Now let's go back to a continuous predictor

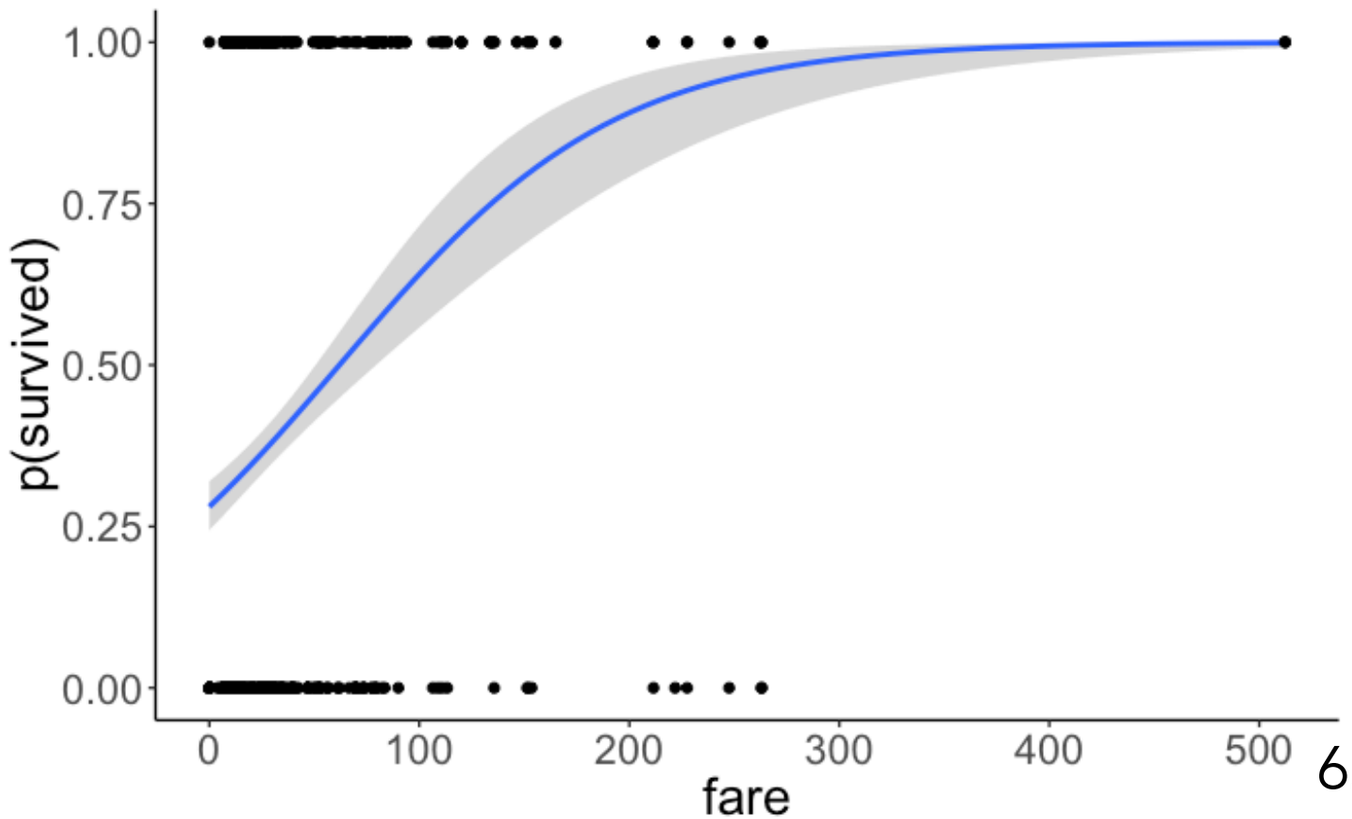
$$\ln\left(\frac{\text{p(survived)}_i}{1-\text{p(survived)}_i}\right) = b_0 + b_1 \cdot \text{fare}_i + e_i$$

```
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%.

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

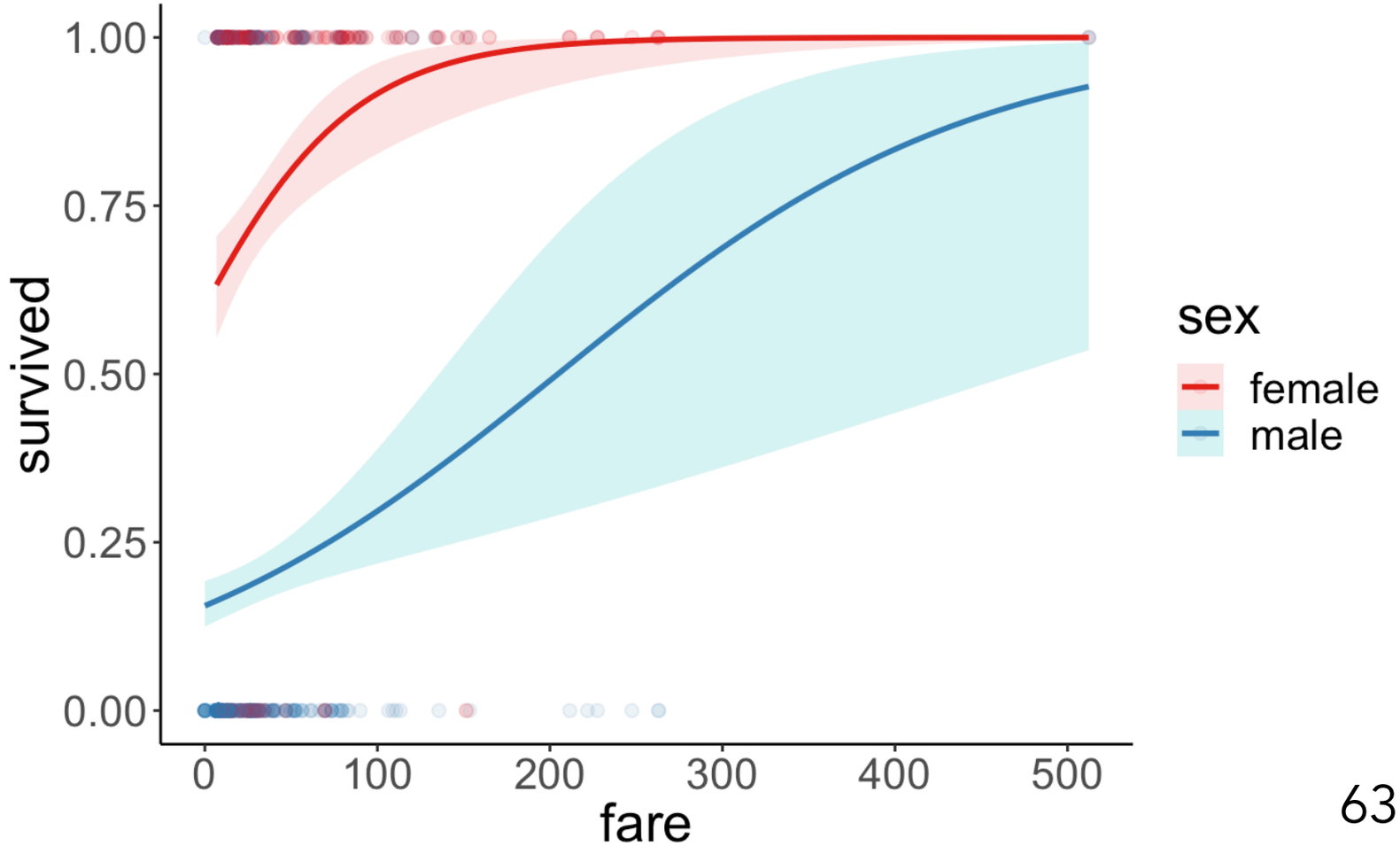
fare	prediction	p(survival)
0	-0.94	0.28
10	-0.79	0.31
50	-0.18	0.45
100	0.58	0.64
500	6.66	1.00



Models with several predictors

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

```
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
```



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)) %>%
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

```r
# make example reproducible
set.seed(1)

# set parameters
sample_size = 1000
b0 = 0
b1 = 1

# generate data
df.data = tibble(
  x = rnorm(n = sample_size),
  y = b0 + b1 * x,
  p = inv.logit(y)) %>%
  mutate(response = rbinom(n(), size = 1, p = p))

# fit model
fit = glm(formula = response ~ 1 + x,
          family = "binomial",
          data = df.data)

# model summary
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

actual value → Y_i ; predicted value → $P(Y_i)$

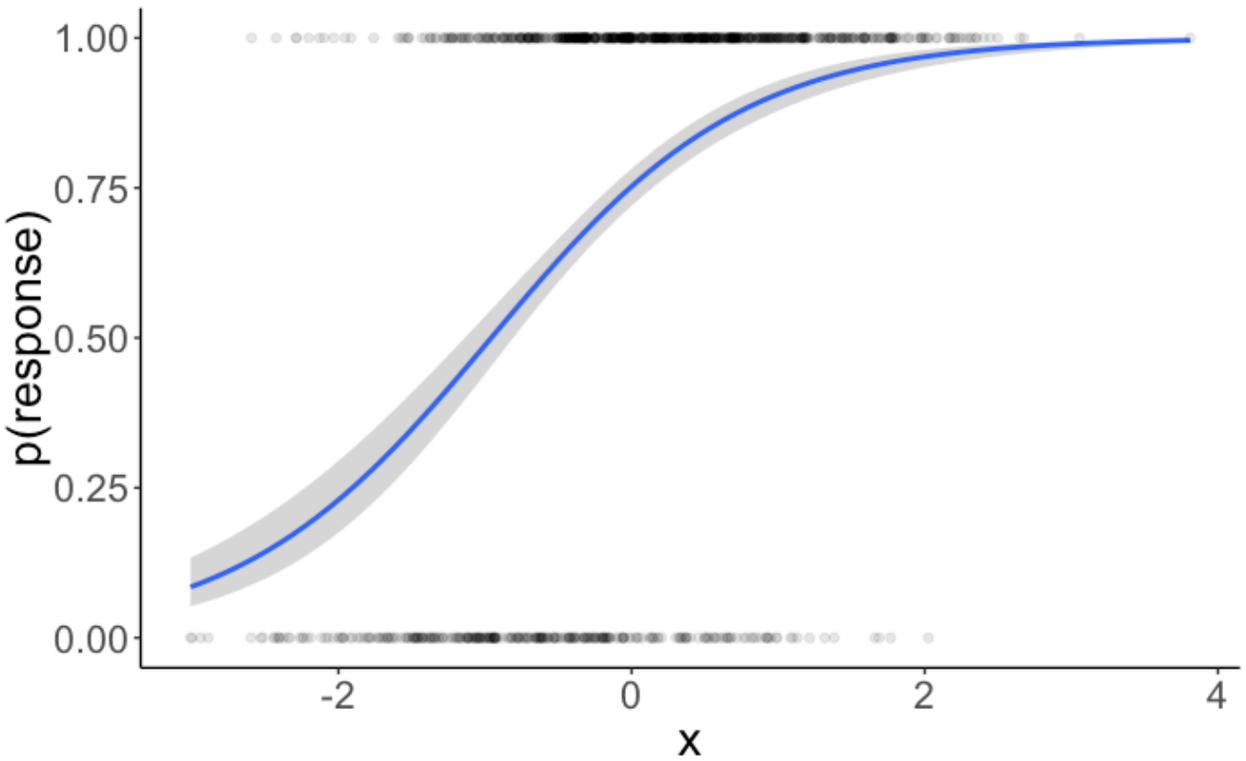
$$\text{log-likelihood} = \sum_{i=1}^{n} [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

- 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

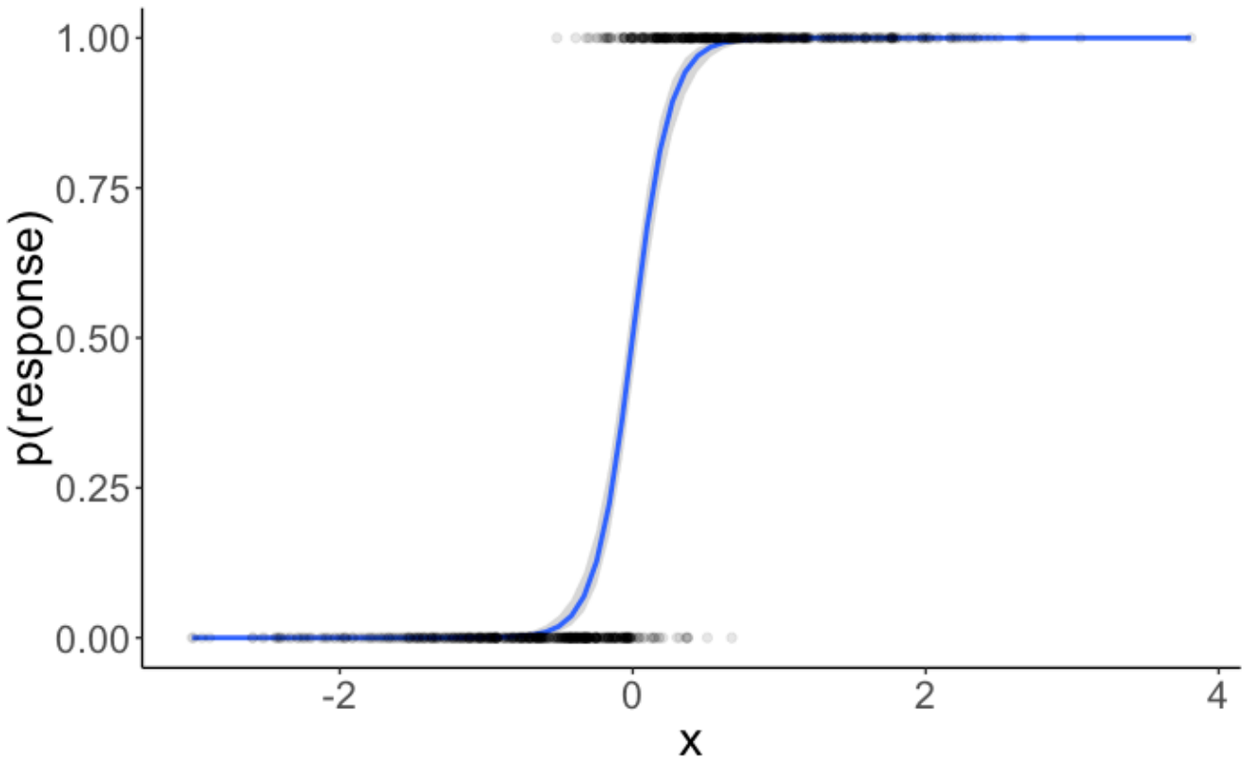
Assessing the model fit

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

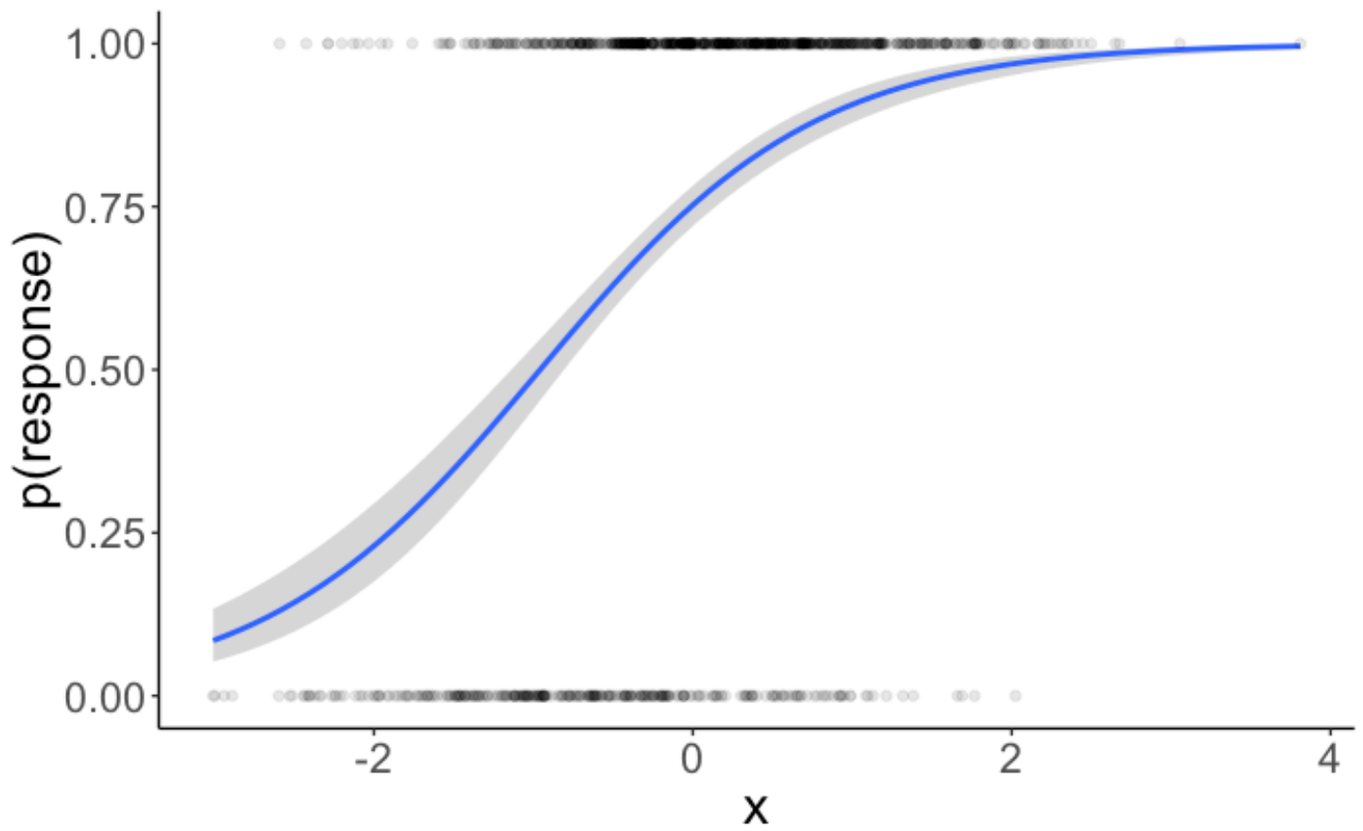
aka checking whether it's **worth it**

```
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

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



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

```r
fit =  glmer(repeatgr ~ 1 + ses * Minority + (1 | schoolNR),
             data = df.language,
             family = "binomial")

fit %>% summary()
```

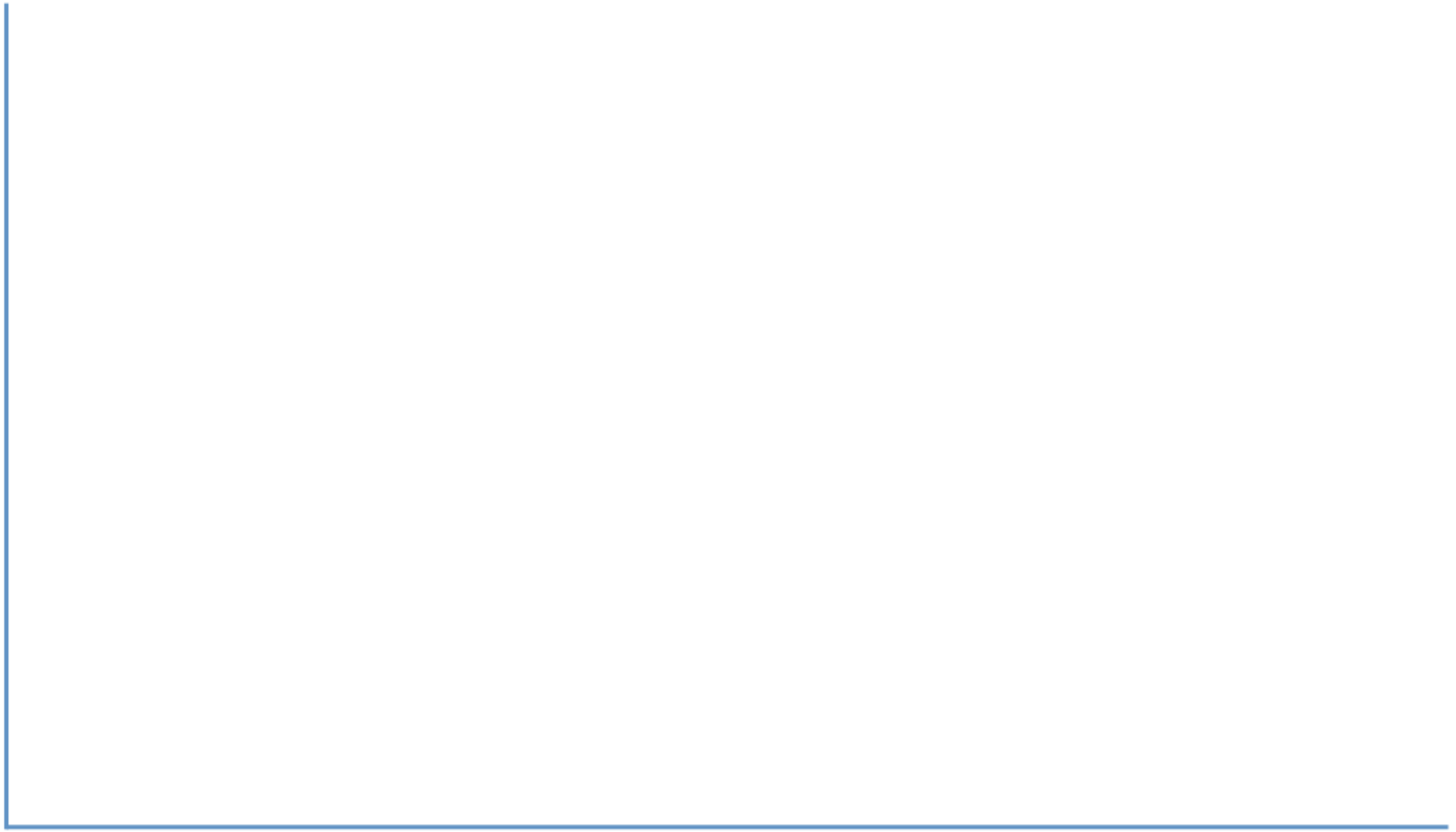
```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: repeatgr ~ 1 + ses * Minority + (1 | schoolNR)
   Data: bdf

     AIC      BIC   logLik deviance df.resid
  1672.8   1701.5   -831.4   1662.8     2282

Scaled residuals:
    Min      1Q  Median      3Q     Max
-0.9602 -0.4071 -0.3155 -0.2219  5.9500

Random effects:
 Groups   Name        Variance Std.Dev.
 schoolNR (Intercept) 0.2583   0.5083
Number of obs: 2287, groups:  schoolNR, 131

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.454556   0.206103  -2.205   0.0274 *
ses           -0.061913   0.007908  -7.829 4.93e-15 ***
MinorityY      0.480047   0.471208   1.019   0.3083
ses:MinorityY  0.011938   0.022737   0.525   0.5996
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) ses    MnrtyY
ses         -0.906
MinorityY   -0.400  0.369
ses:MinrtyY  0.299 -0.321 -0.866
```

Summary

- Linear mixed effects model
  - Getting p-values
  - Pitfalls in fitting **`lmer`**`()`s (and what to do about it)
  - Understanding **`lmer`**`()` syntax
- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression

Feedback

How was the pace of today's class?



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!