Generalized linear model

02/26/2020

Logistics

Application section

Thursday, February 27th, 4:30pm - 5:20pm in 160-322

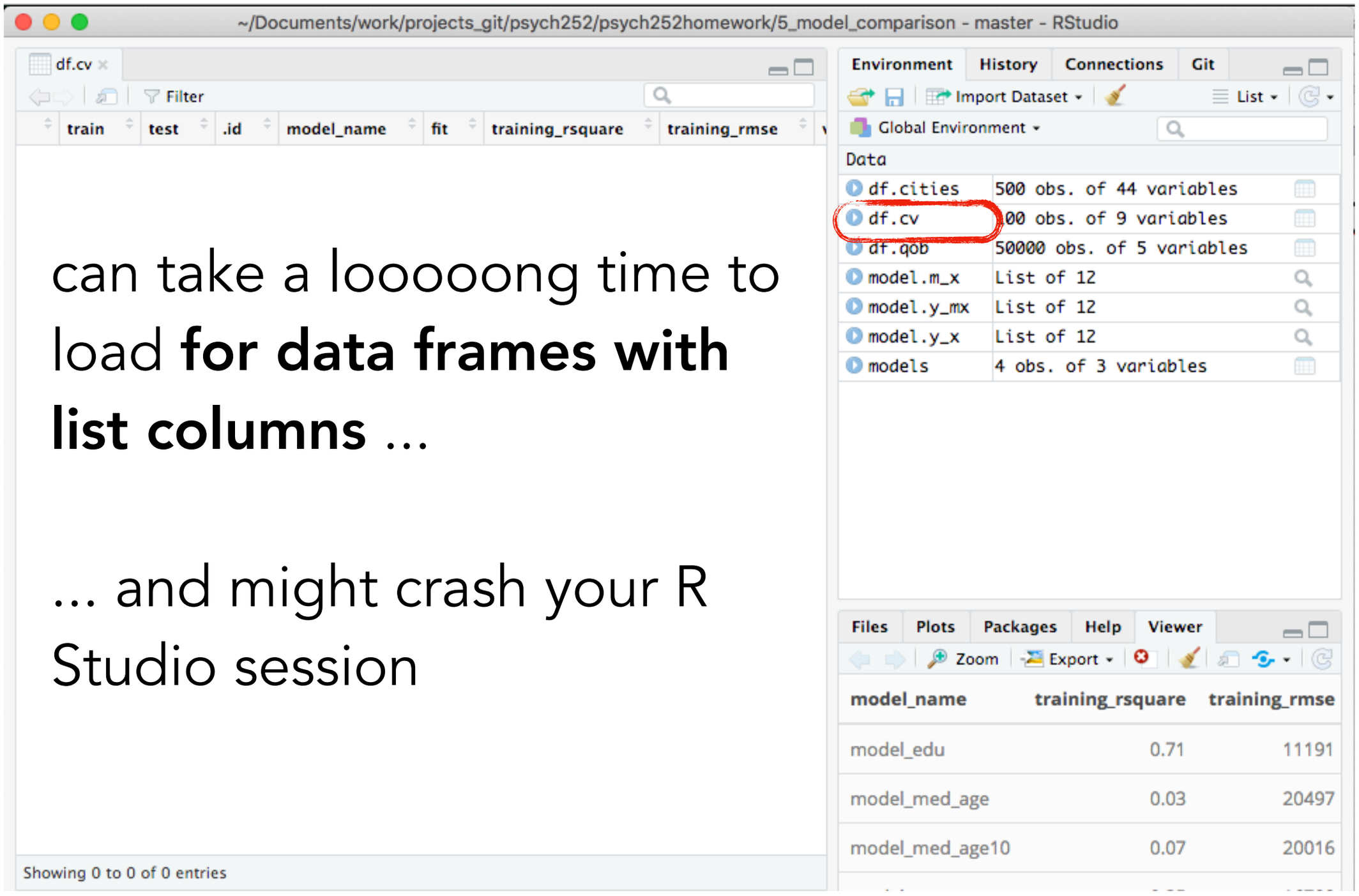
Applied linear mixed effects models

Midterm

- will release grades later today
- check with us if it looks like we made any mistakes grading your midterm
- it's possible to re-grade your midterm

Things that came up

Data viewer can be slooooooooooow



can take a looooong time to load **for data frames with list columns** …

… and might crash your R Studio session

Data viewer can be slooooooooooow

- if you want to check out a data frame with list columns:
  - make a data frame that only has one or two rows (and not too many columns)
  - then you can inspect this data frame using the data viewer
  - otherwise, use indexing in the R console:
  - `df.cv$fit[[1]]` will show the first entry in the fit list column

Getting tables to fit in latex

```
df.cv %>%
  kable(digits = 2) %>%
  kable_styling(latex_options = c("scale_down", "striped"))
```

scales to make it fit into
the page margins

h/t Rebecca Hinds

means hat tip, or tip of the hat—a way of recognizing the original source of a meme, expression, image, or idea on social media

Question 3.1

Since there are so many different models we could make with all those variables, we would like to use some measure to compare how well they fit the data. Some methods are AIC/BIC and cross-validation.

Create a data frame with 4 rows and the following 5 columns: `model_name`, `rsquared`, `logLik`, `AIC`, `BIC`. Each row should represent the following 4 models

1. `model_med_age`: mean_income ~ median_age,
2. `model_med_age10`: mean_income ~ median_age + median_age^2 + median_age^3 + ... + median_age^10
3. `model_edu`: mean_income ~ less9thgrade + grade9to12 + highschool + somecollege + assoc + bachelors + grad
4. `model_race`: mean_income ~ percent_white + percent_black + percent_amindian_alaskan + percent_asian + percent_nativeandother + percent_other_nativeandother + percent_hispanicorlatino + percent_race_other

To create the data frame, make sure to use `map()` for fitting the linear models, and you can use `glance()` from the `broom` package to get the model summaries in tidy format. For defining model 2. the `poly()` function is helpful.

Which model is the best model using the different measures? Are the different model comparison measures consistent?

Question 3.1

To create the data frame, make sure to use `map()` for fitting the linear models, and you can use `glance()` from the `broom` package to get the model summaries in tidy format. For defining model 2. the `poly()` function is helpful.

Which model is the best model using the different measures? Are the different model comparison measures consistent?

apply the function to each element of the list

```
map(.x = list, .f = ~ function(.x))
```

- make a data frame that has as one column the formulas (as strings) for the different models, e.g. `"mean_income ~ median_age"`
- remember the syntax for `lm(formula, data)`
- `lm(formula = "mean_income ~ median_age", data = df.cities)`
- use `map()` to iterate over the different formulas, and apply them to the same data set

Question 3.3

Your RA comes running to you and tells you that a few cities were missing in the file they previously sent you and hands you another file with cities missing from the first dataset. You realize that this is a good opportunity to test for the robustness of your models by checking against this new dataset. Using the same 4 models regressed using the entire training data, **create a dataframe with 4 rows (for each model) and the following 7 columns: `model_name`, `training_rsquare`, `training_rmse`, `validation_rsquare`, `validation_rmse`, `test_rsquare`, `test_rmse`**, where `test` refers to the new dataset that was **not** used during training.

What do you notice about the rsquare of this test set? Explain why this may be happening (you may need to dig in the data a bit to find the answer; you may include a supporting figure if it's helpful).

"Are we supposed to rerun crossv_mc on the entire data? Is there a similar example that we can look at from class? Also part of the question is cut off. Thank you!"

- Use the model fits from Question 3.1 and apply those fitted models to the new test data.
- Combine these results with the cross-validation results from Question 3.2

Plan for today

- Linear mixed effects model
  - very quick reminder
  - 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

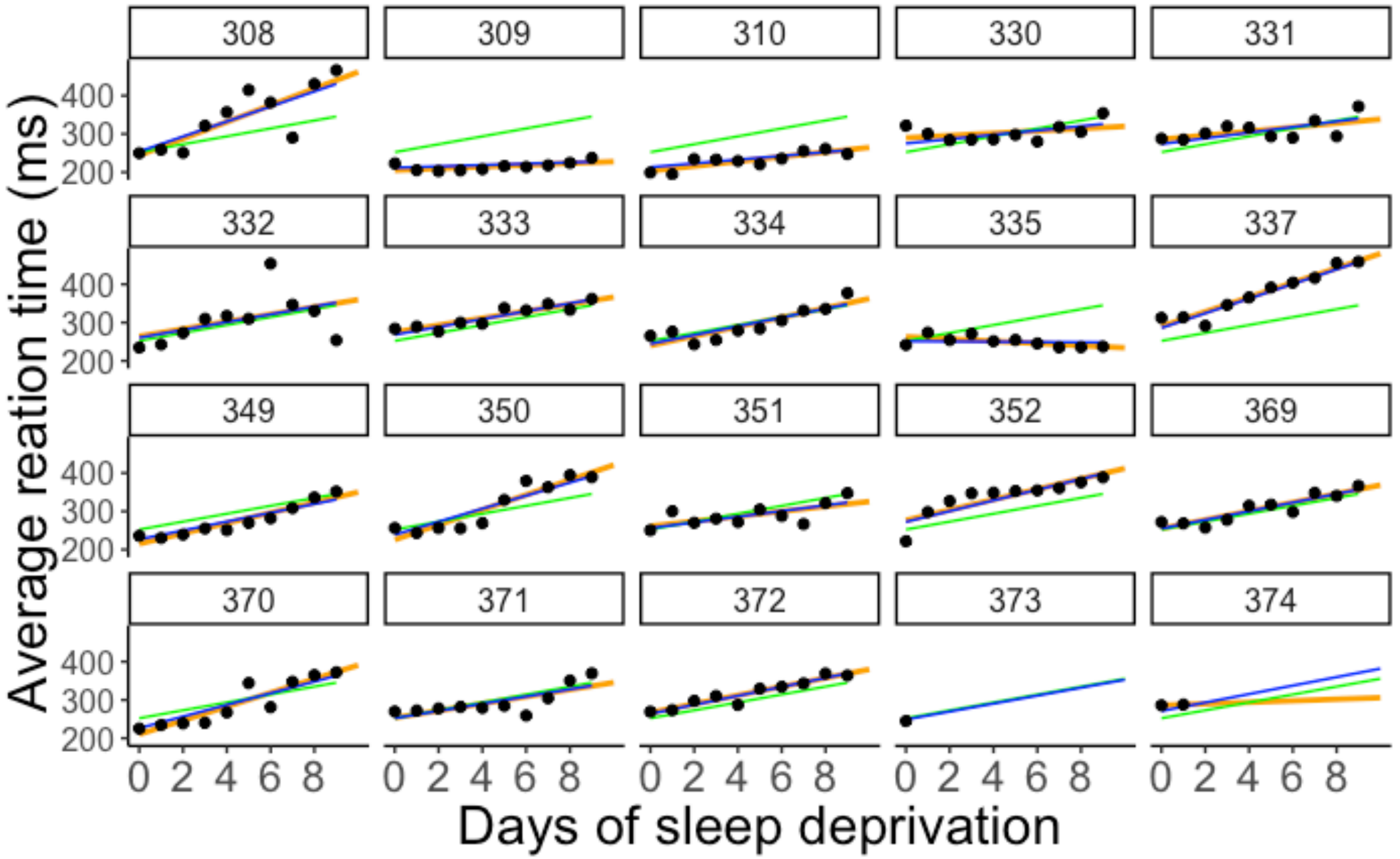
very quick reminder

Pooling and shrinkage

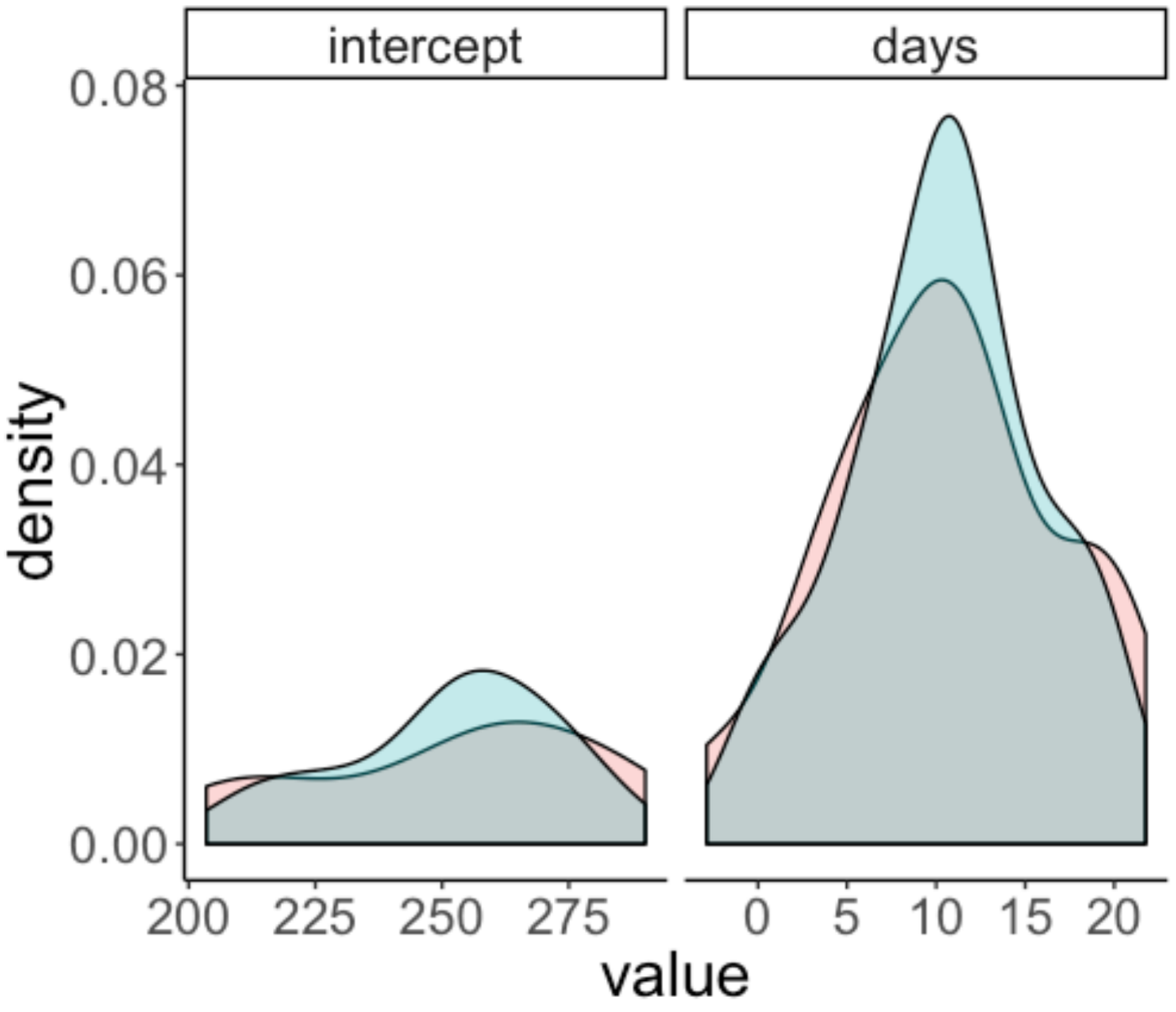
complete pooling

no pooling

partial pooling



Pooling and shrinkage

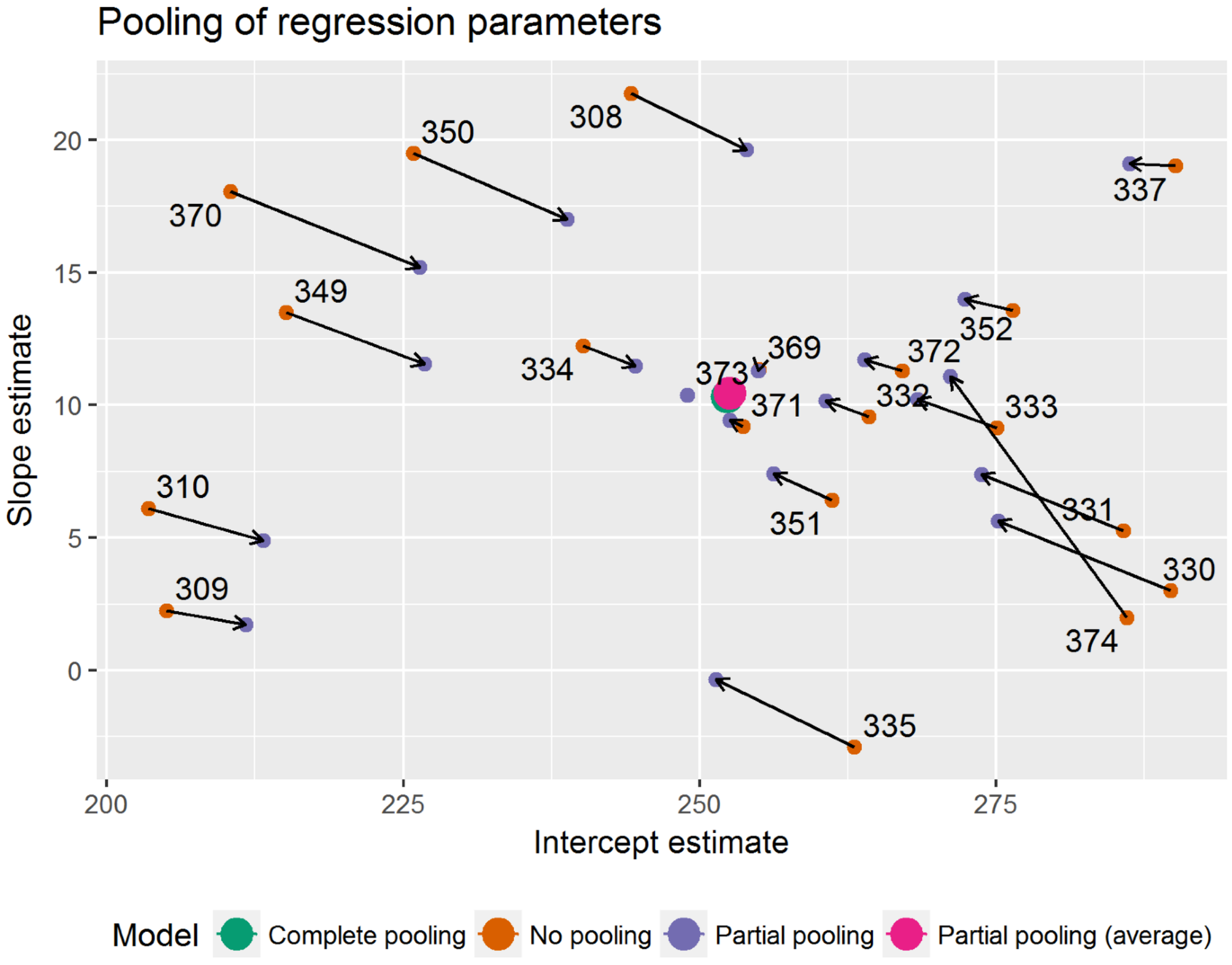


standard deviation

method	intercept	days
no pooling	28.95	6.56
partial pooling	21.59	5.46

variance "shrinks"

Pooling and shrinkage



Pitfalls in fitting `lmer()`s

My model does not converge ...

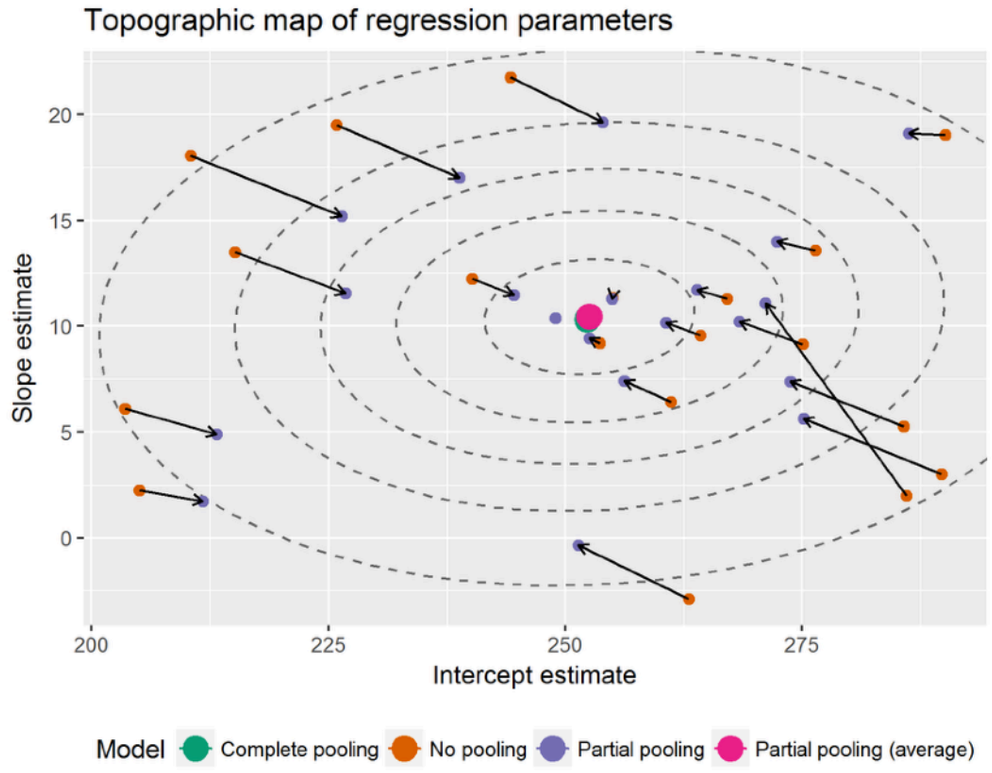
- **`lmer`**`()`s are solved through a (complicated) process of iterative optimization
- only interpret the results of models that actually converged!
- here are some tricks that might help:
  - *continuous predictors*: center and scale
  - *categorical predictors*: choose a factor that has more data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

Remove the correlation component from your model

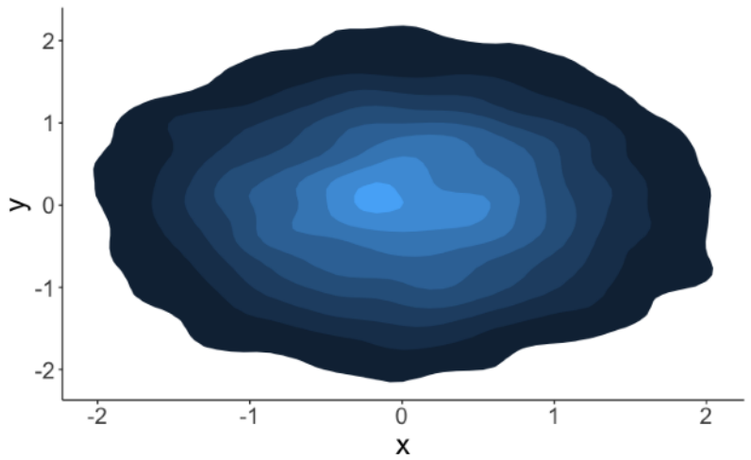
```
# fit the model
fit.lmer = lmer(formula = reaction ~ 1 + days + (1 + days | subject),
                data = df.sleep)
# model summary
fit.lmer %>%
  summary()
```

```
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
```

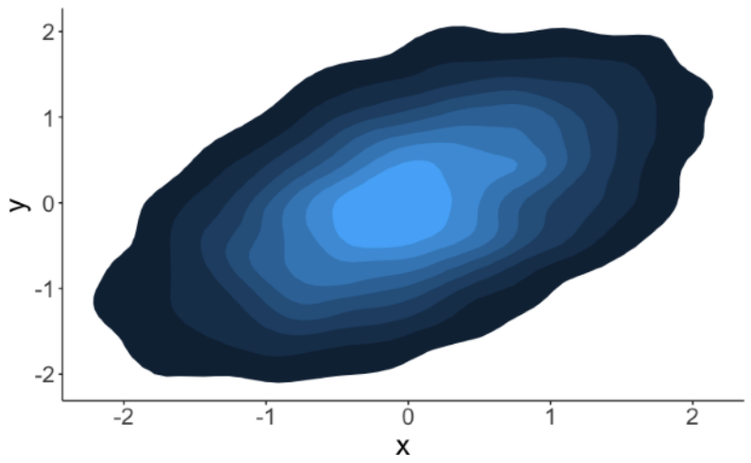
multivariate Gaussian



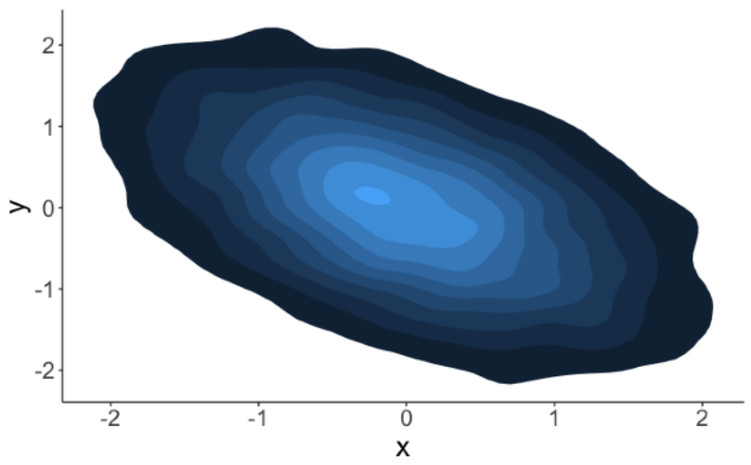
Remove the correlation component from your model



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()
```

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 ...

Description

`[g]lmer` fits may produce convergence warnings; these do **not** necessarily mean the fit is incorrect (see "Theoretical details" below). The following steps are recommended assessing and resolving convergence warnings (also see examples below):

- double-check the model specification and the data
- adjust stopping (convergence) tolerances for the nonlinear optimizer, using the `optCtrl` argument to `[g]lmerControl` (see "Convergence controls" below)
- center and scale continuous predictor variables (e.g. with `scale`)
- double-check the Hessian calculation with the more expensive Richardson extrapolation method (see examples)
- restart the fit from the reported optimum, or from a point perturbed slightly away from the reported optimum
- use `allFit` to try the fit with all available optimizers (e.g. several different implementations of BOBYQA and Nelder-Mead, L-BFGS-B from `optim`, `nlminb`, ...). While this will of course be slow for large fits, we consider it the gold standard; if all optimizers converge to values that are practically equivalent, then we would consider the convergence warnings to be false positives.

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

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
```

Plan for today

- Linear mixed effects model
  - very quick reminder
  - 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

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

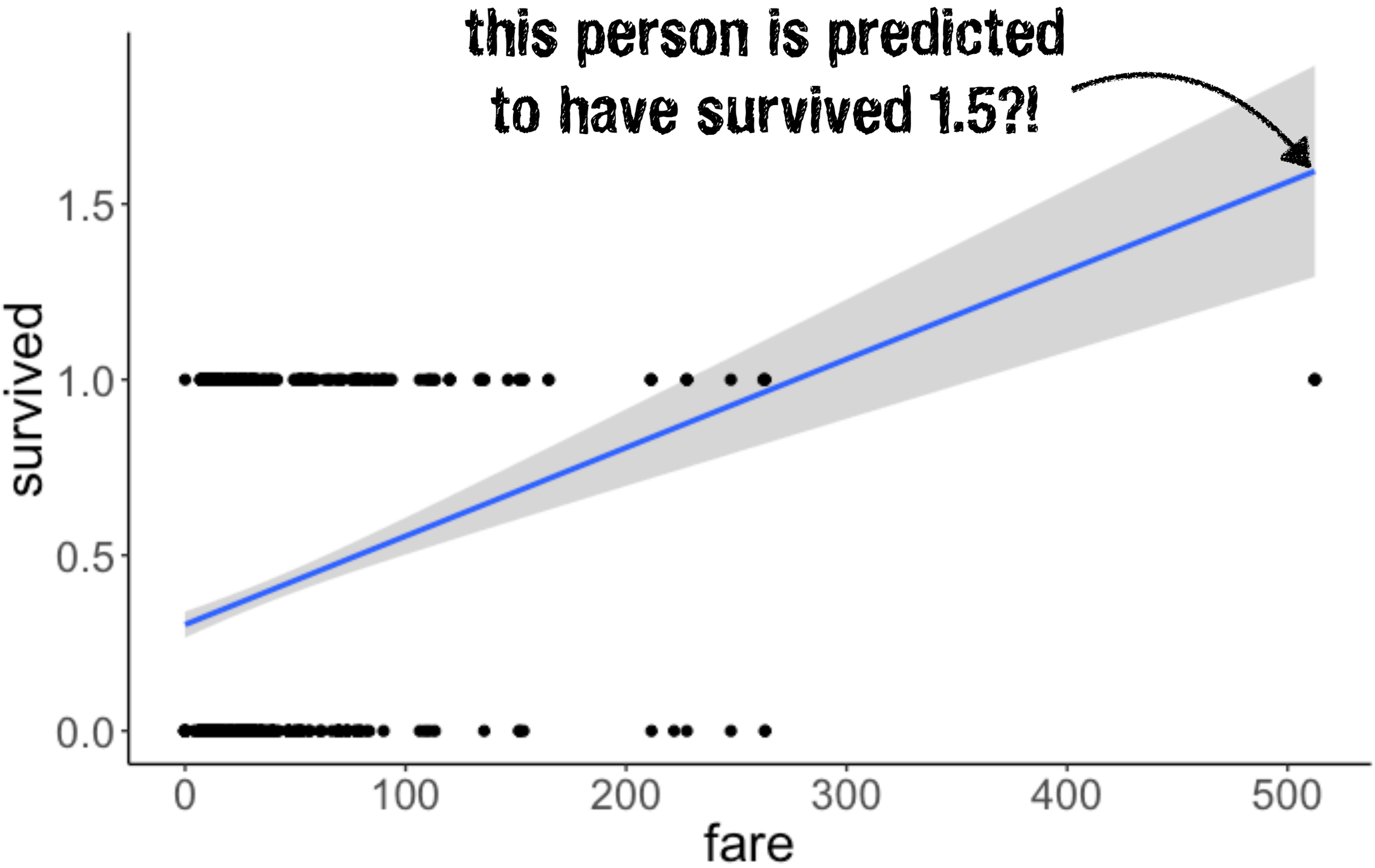
Is there a relationship between `fare` and `survived`?

```
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/

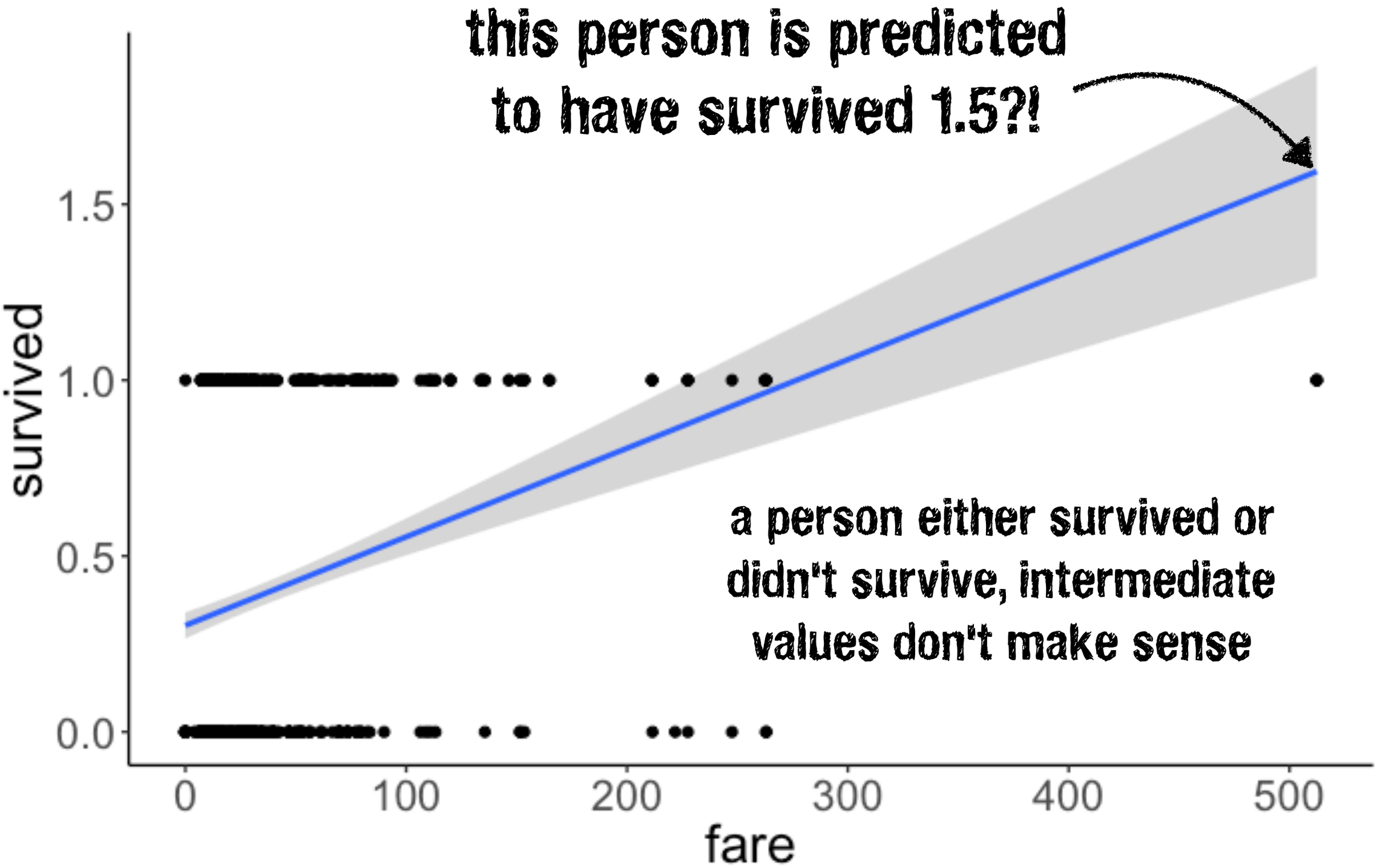
Is there a relationship between `fare` and `survived`?

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

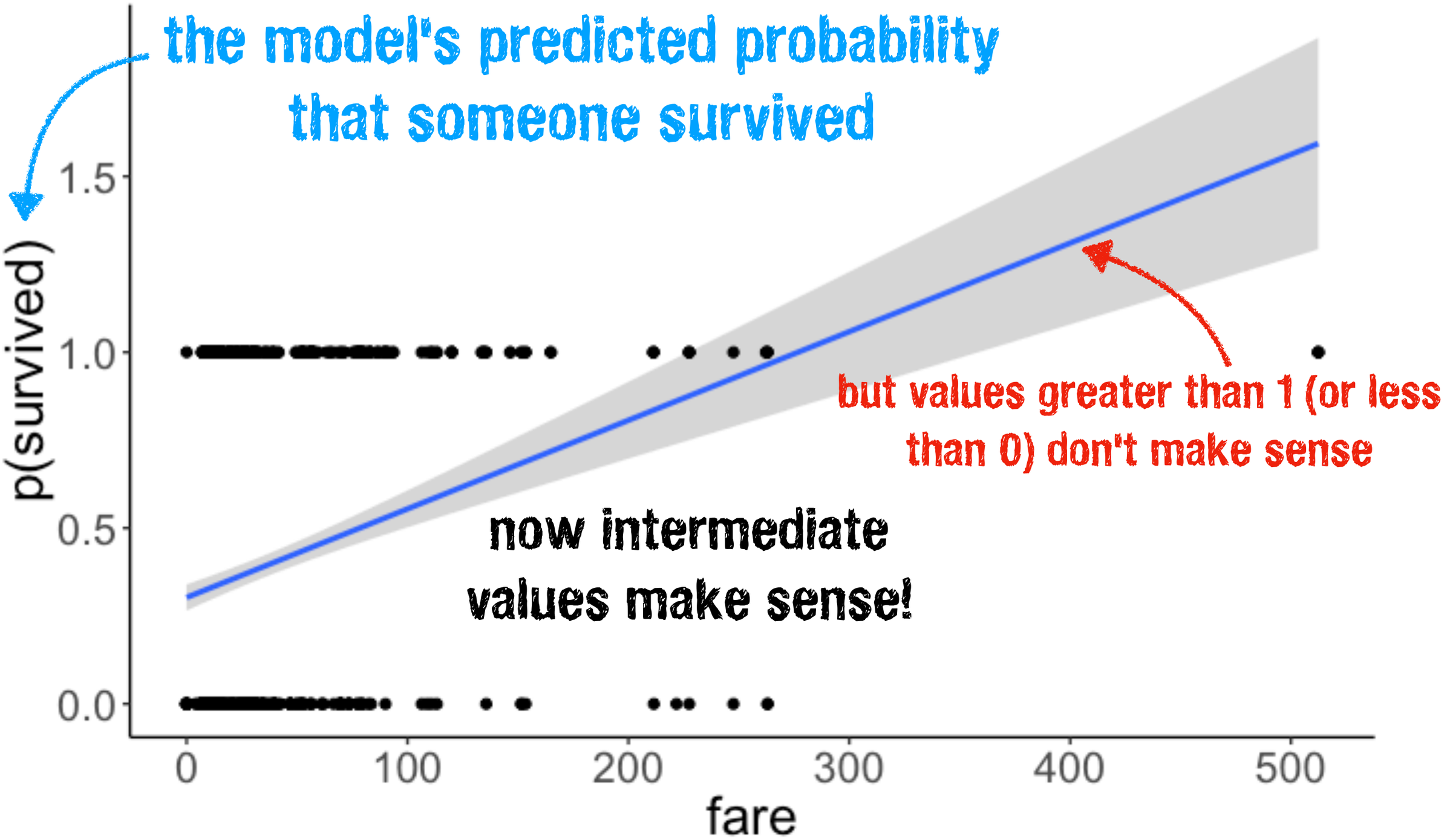
The fact that this class is called "**Generalized linear model**" suggests we can!

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

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

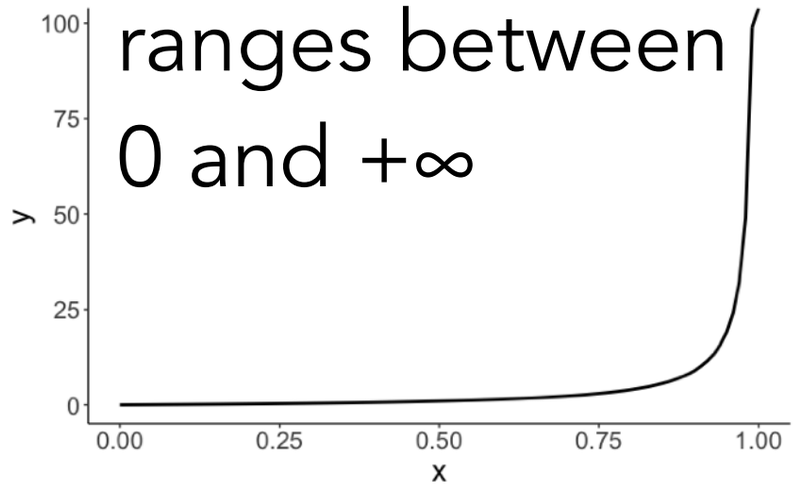
predict the probability of Y

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

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)

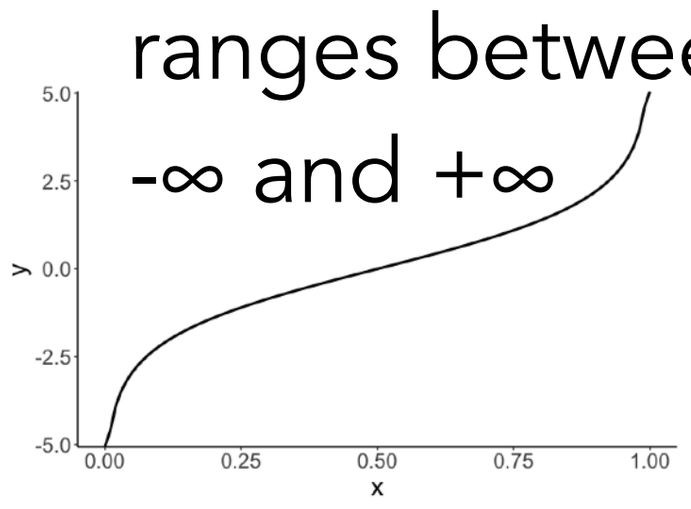
Step 1: Calculate the "odds"

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



Step 2: Take the (natural) log

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



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

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

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

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: $P(Y_i = 1) < P(Y_i = 0)$

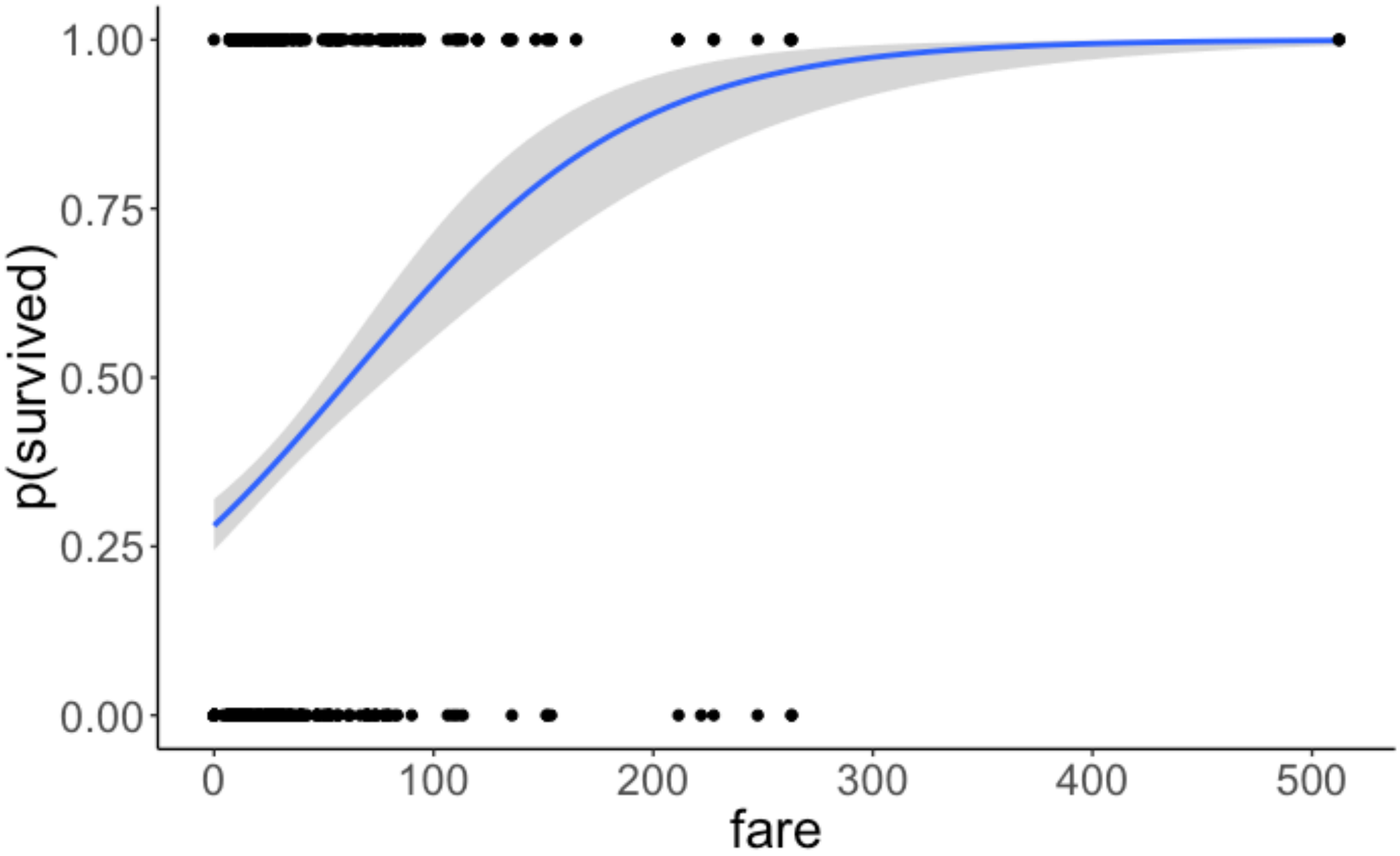
Fitting a logistic regression in R

```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

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

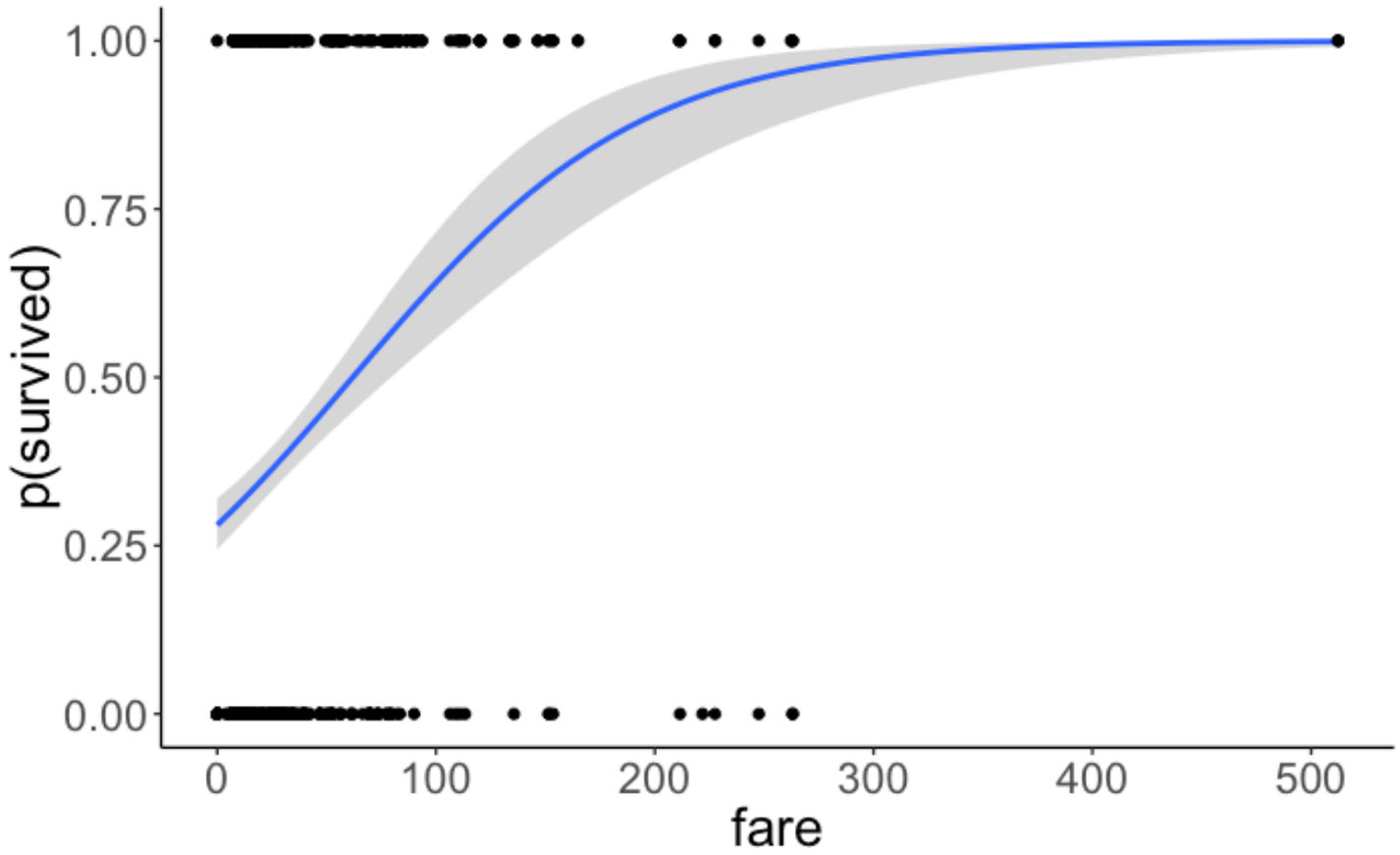


Plan for today

- Linear mixed effects model
  - very quick reminder
  - 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

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)

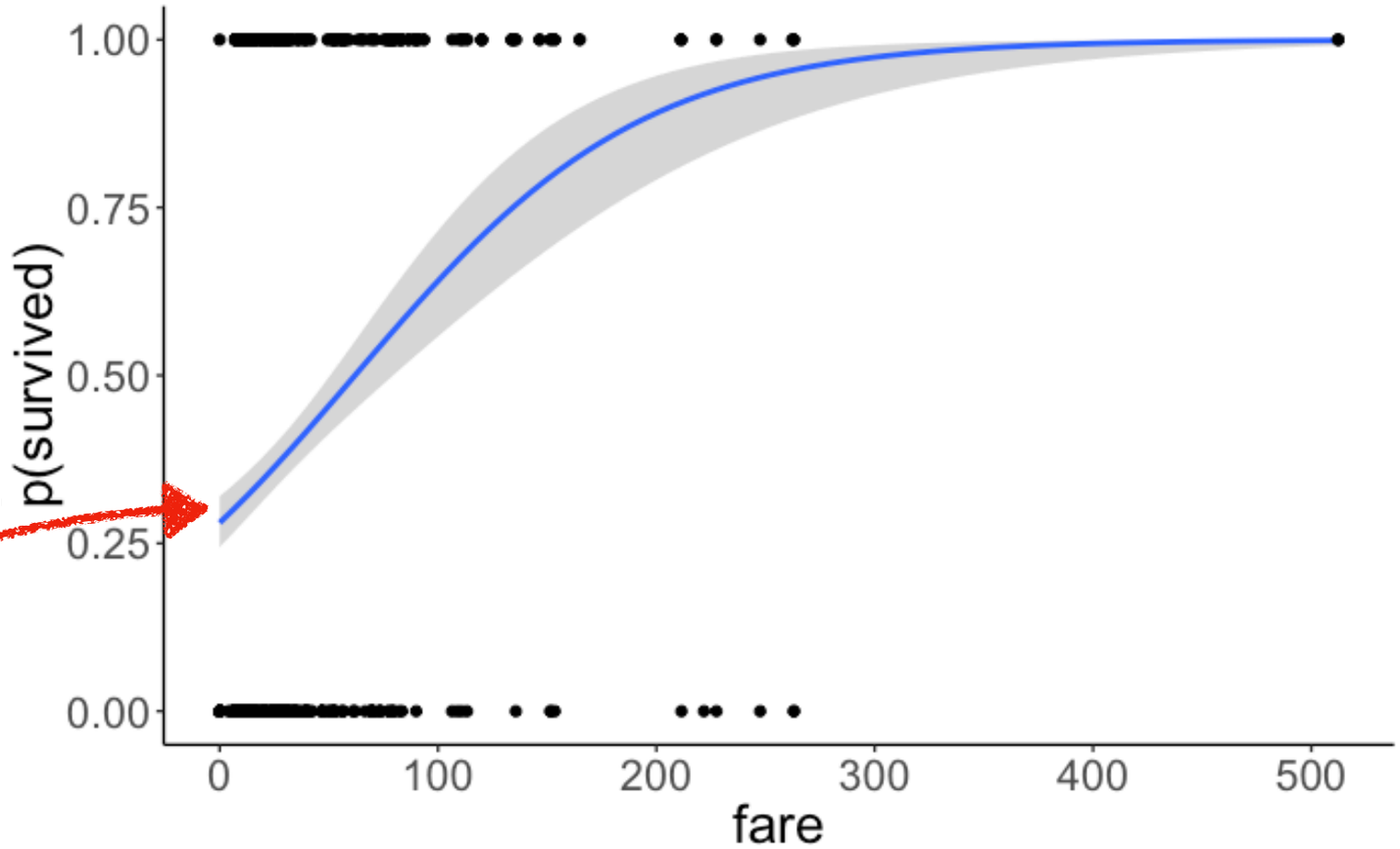
$$\pi_i = \frac{e^{b_0 + b_1 \cdot X_i + e_i}}{1 + e^{b_0 + b_1 \cdot X_i + e_i}}$$

another way to
specify the model

Interpreting the model output

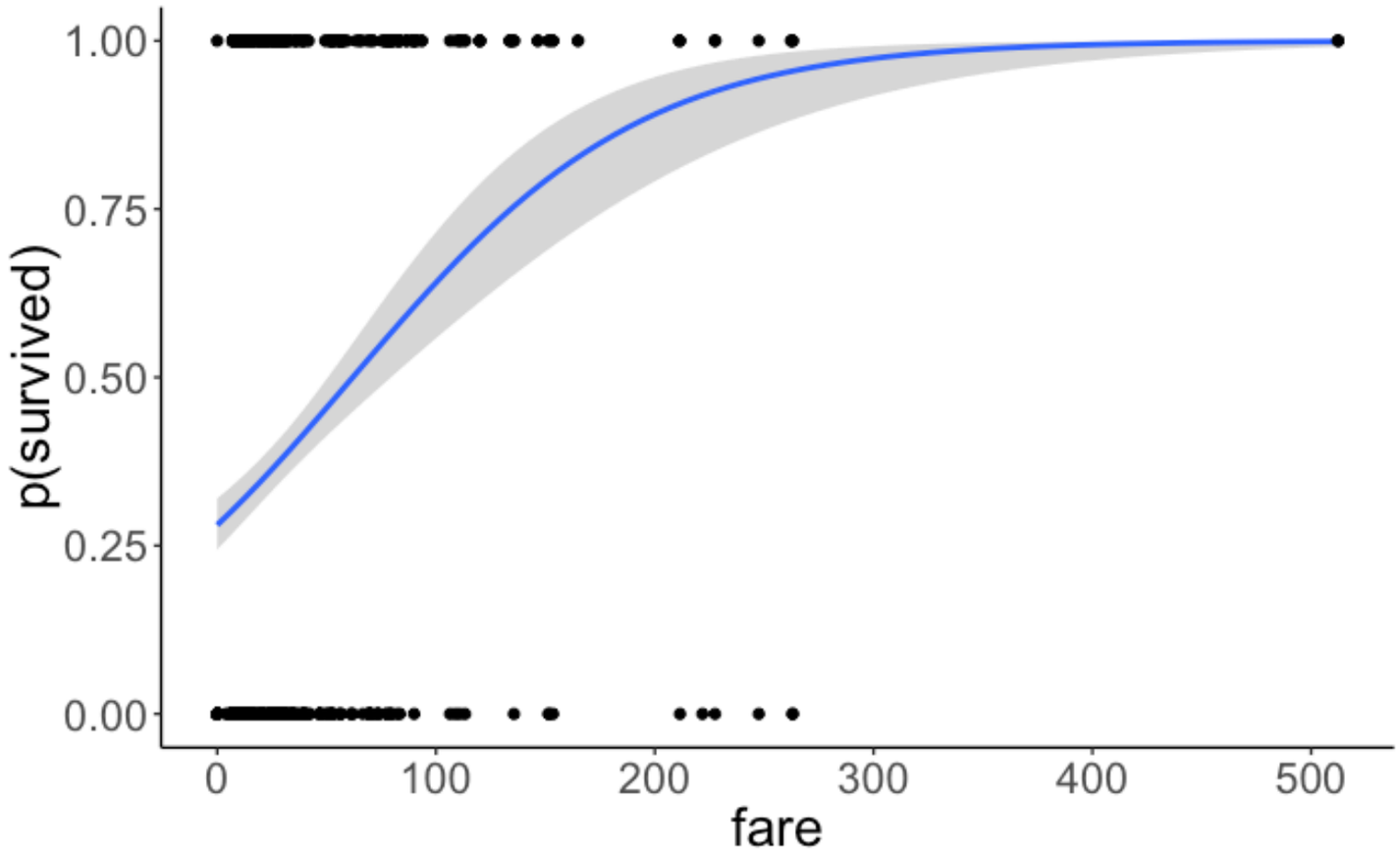
inverse logit

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




```
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
```

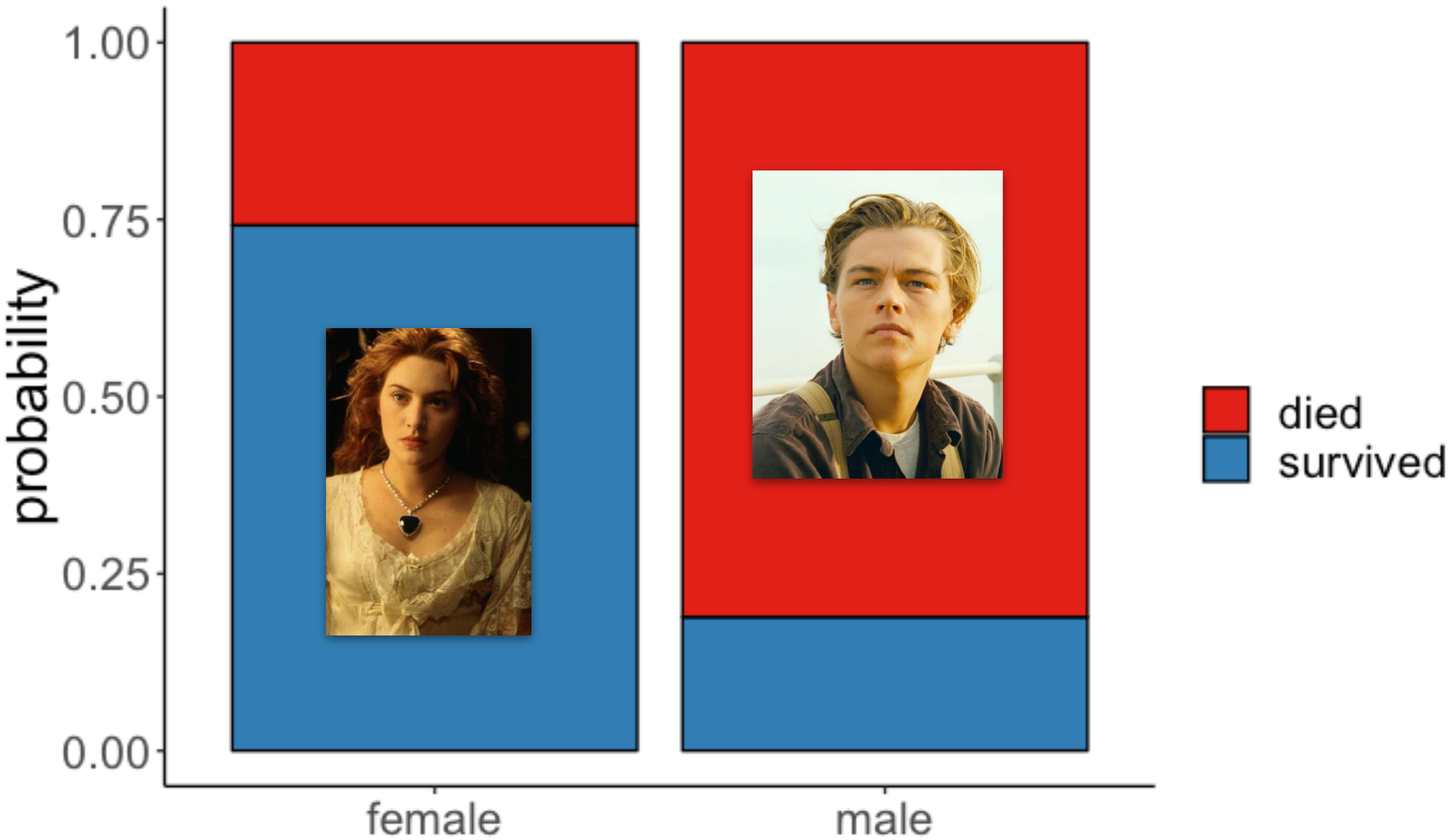
Interpreting the model output




```
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

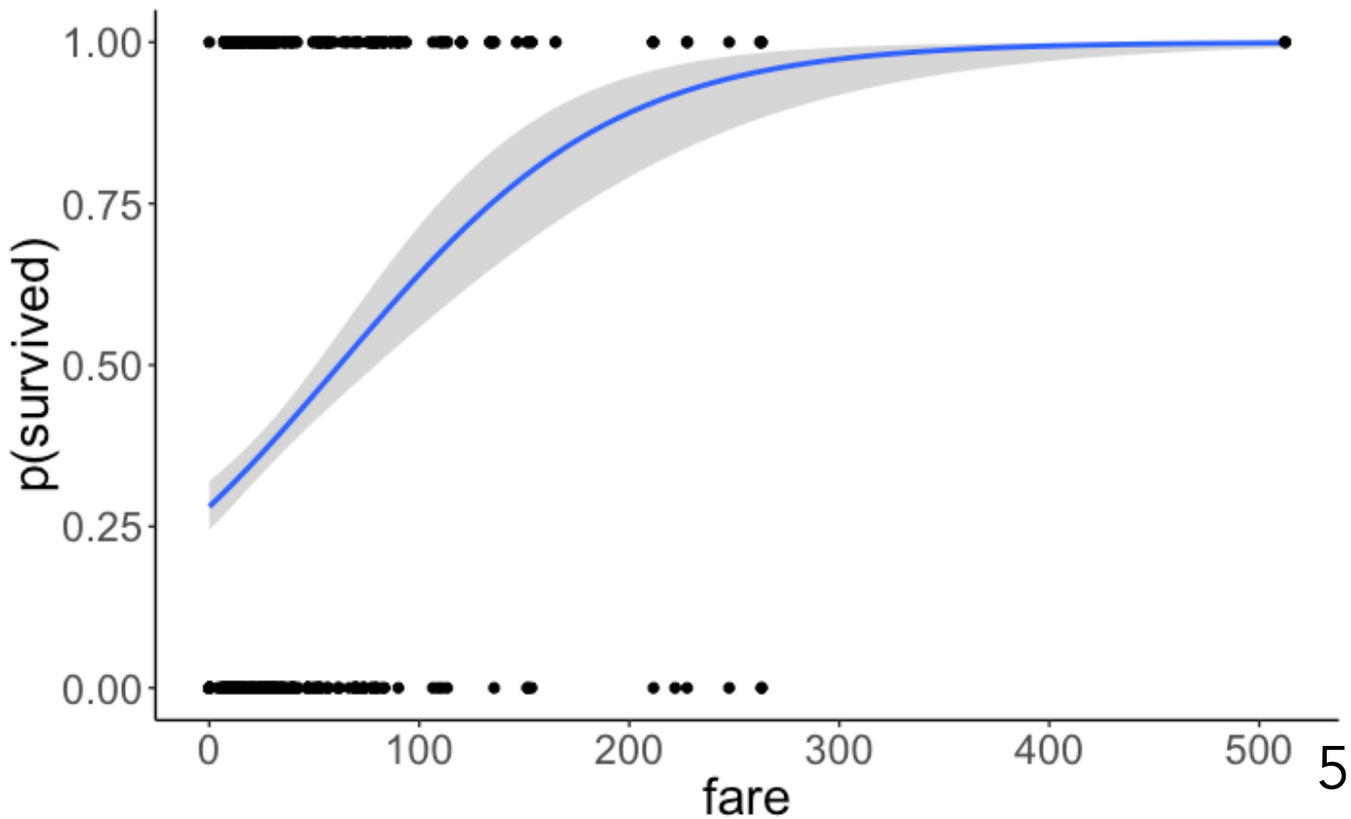
$$\ln\left(\frac{\text{p(survived)}_i}{1-\text{p(survived)}_i}\right) = b_0 + b_1 \cdot \text{fare}_i + e_i$$

```
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

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

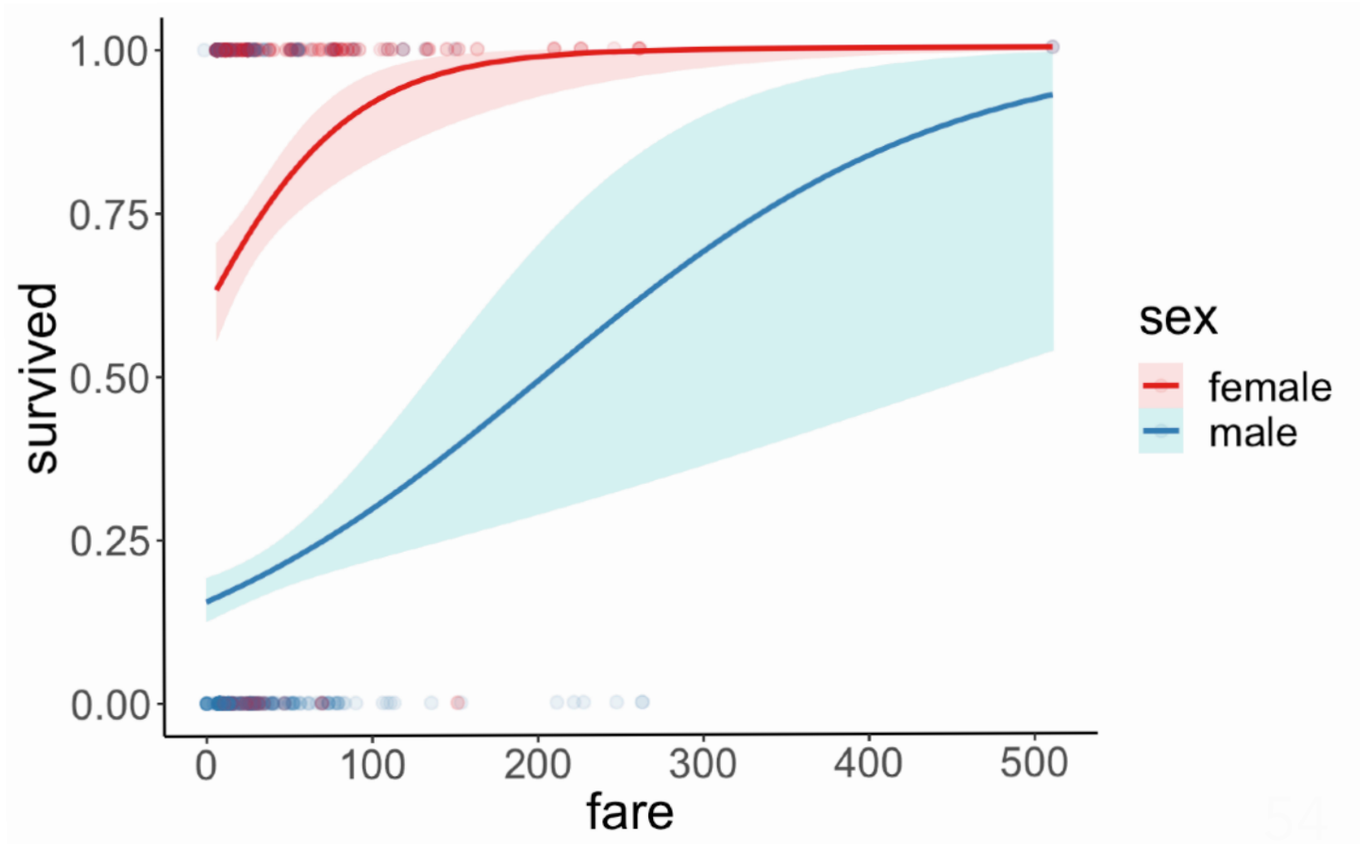
```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.647100   0.148502   4.358 1.32e-05 ***
sexmale     -2.422760   0.170515 -14.208  < 2e-16 ***
fare         0.011214   0.002295   4.886 1.03e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

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

```
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")
```

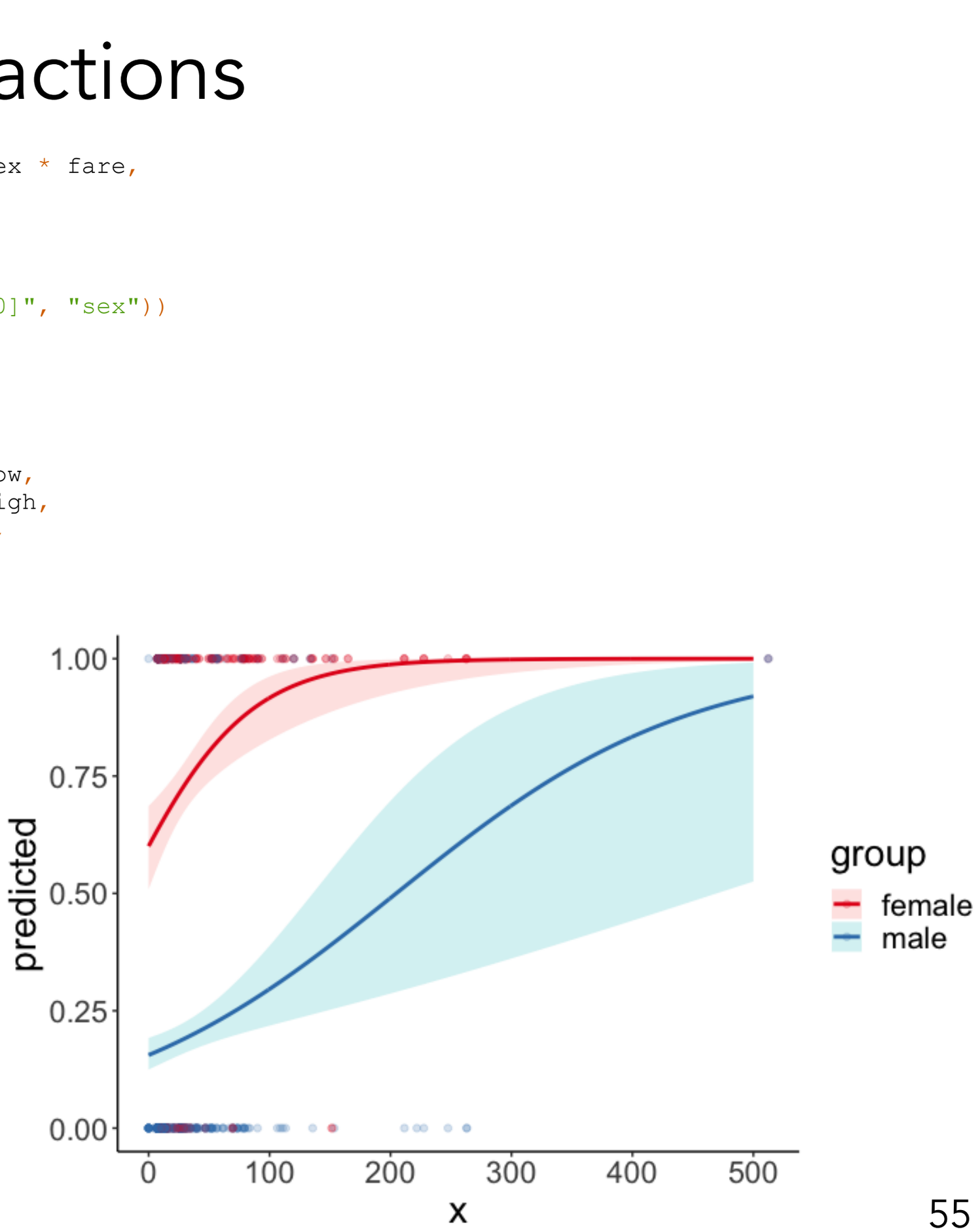
```
1 ggpredict(fit.glm,
2           terms = c("sex"))
```

```
# 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
```



Models with interactions

```r
fit.glm3 = glm(formula = survived ~ 1 + sex * fare,
               family = "binomial",
               data = df.titanic)

df.data = ggpredict(fit.glm3,
                    terms = c("fare [0:500]", "sex"))

ggplot(data = df.data,
       mapping = aes(x = x,
                     y = predicted,
                     color = group)) +
  geom_ribbon(mapping = aes(ymin = conf.low,
                            ymax = conf.high,
                            fill = group),
              alpha = 0.2,
              color = NA) +
  geom_line(size = 1) +
  geom_point(data = df.titanic,
             mapping = aes(x = fare,
                           y = survived,
                           color = sex),
             alpha = 0.2) +
  scale_color_brewer(palette = "Set1")
```



Plan for today

- Linear mixed effects model
  - very quick reminder
  - 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

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

predicted value

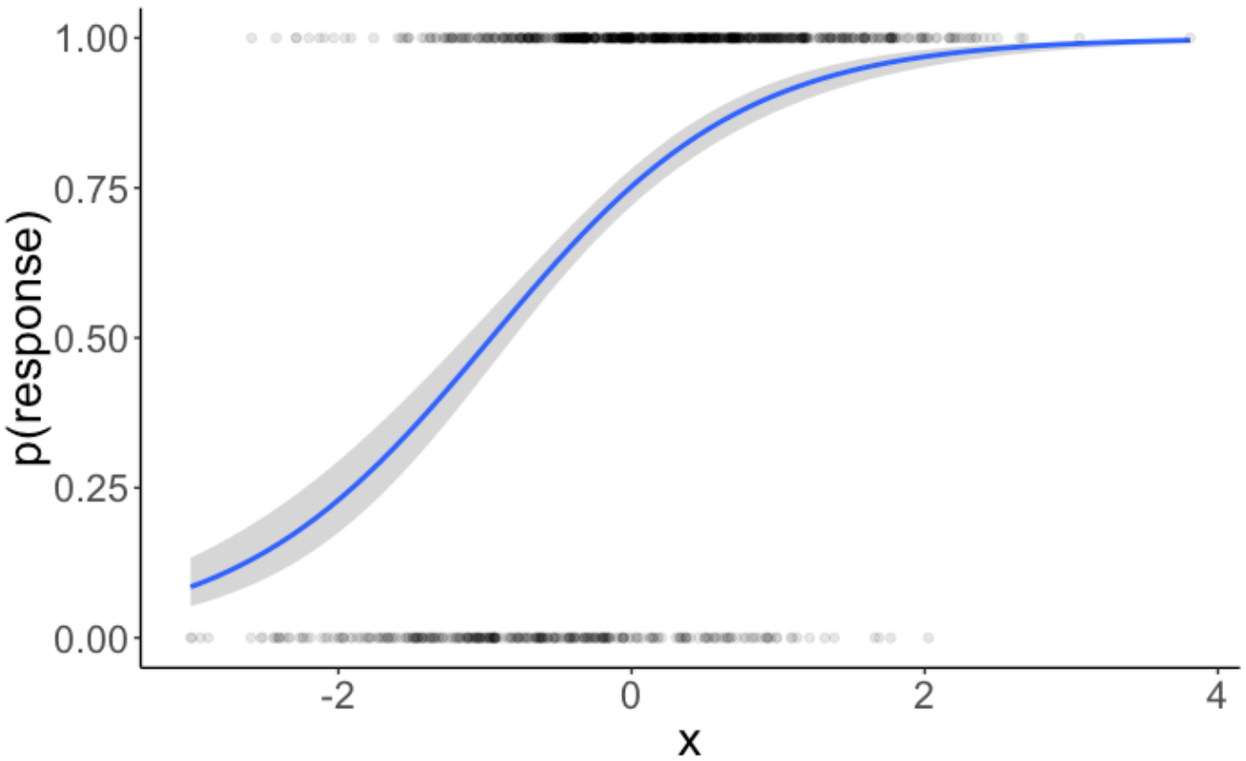
$$\text{log-likelihood} = \sum_{i=1}^{n} [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

- 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

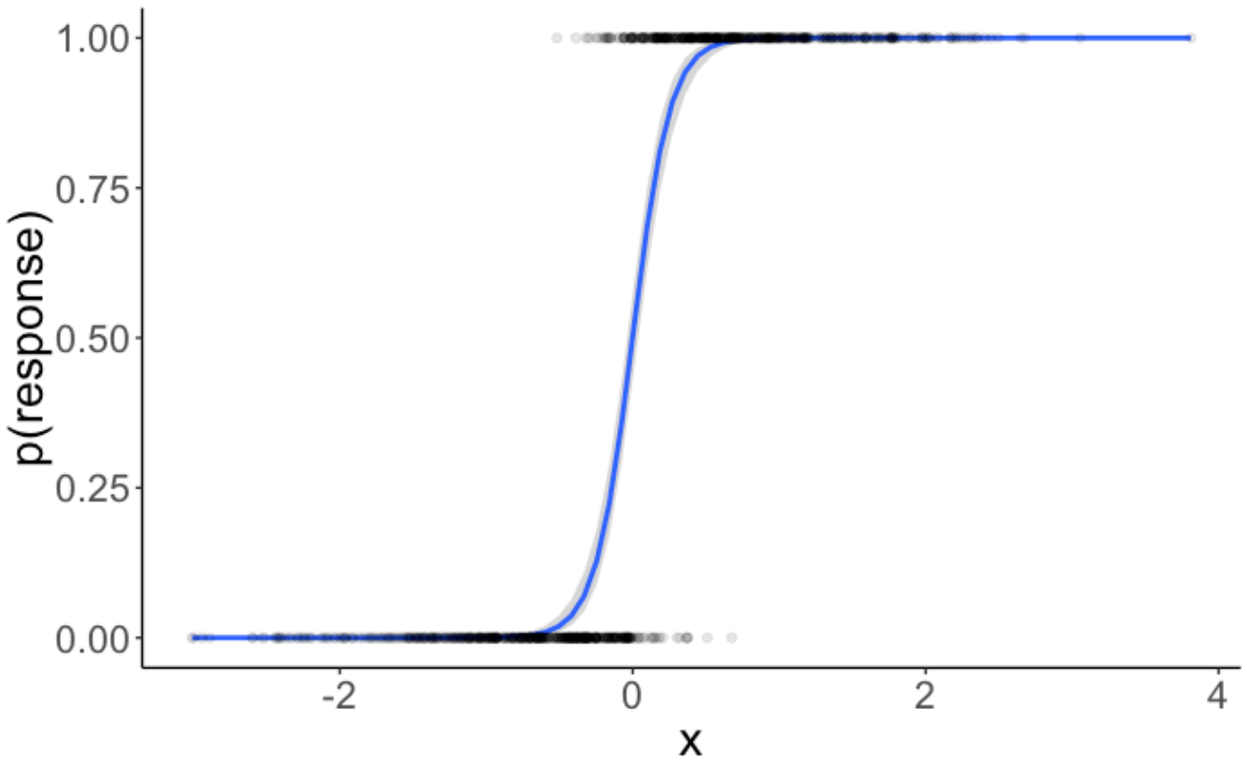
Assessing the model fit

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**

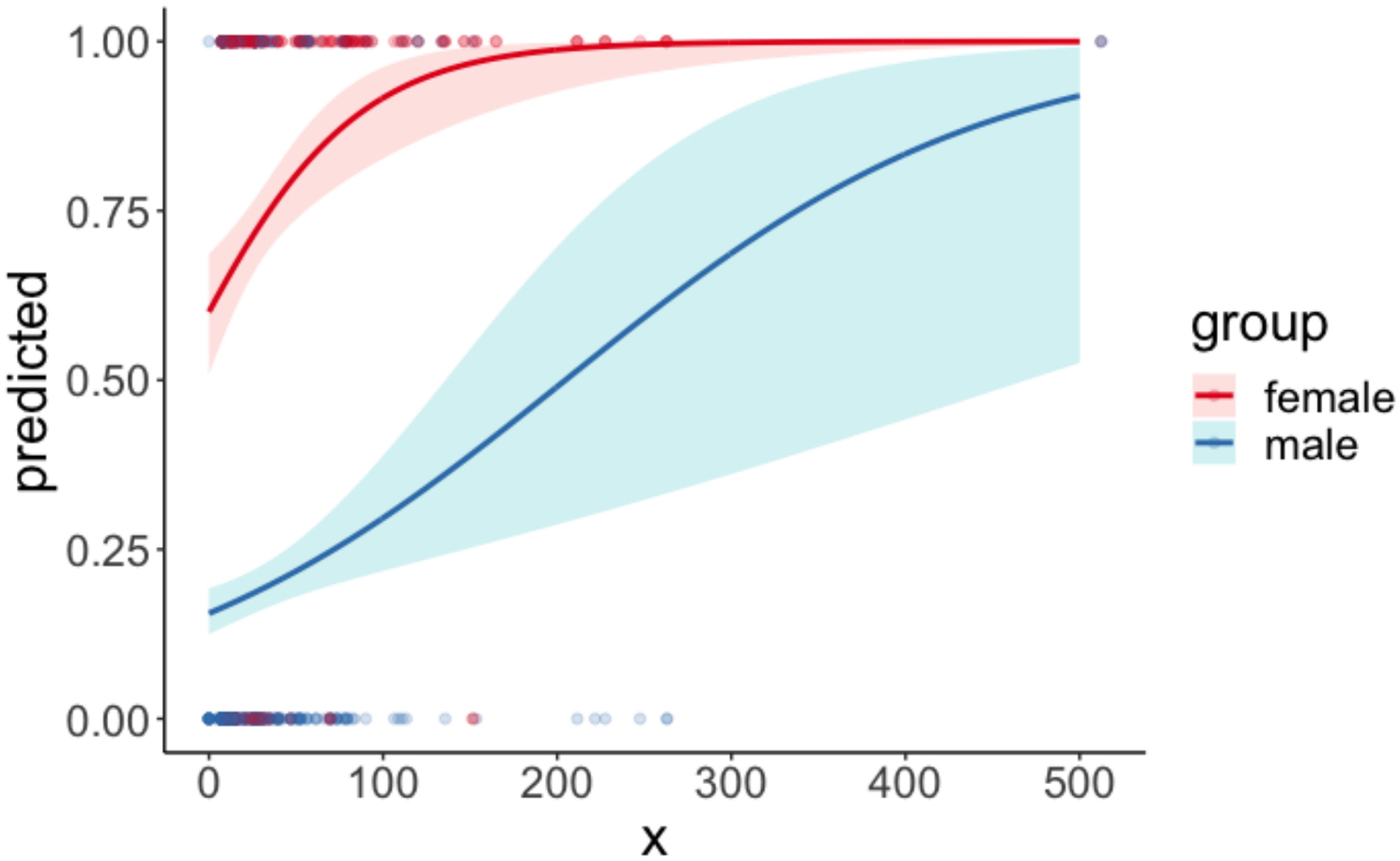
```
1  # fit compact model
2  fit.compact = glm(formula = survived ~ 1 + fare,
3                    family = "binomial",
4                    data = df.titanic)
5
6  # fit augmented model
7  fit.augmented = glm(formula = survived ~ 1 + sex + fare,
8                      family = "binomial",
9       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

```
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

Plan for today

- Linear mixed effects model
  - very quick reminder
  - 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**

Mixed effects logistic regression

Mixed effects logistic regression

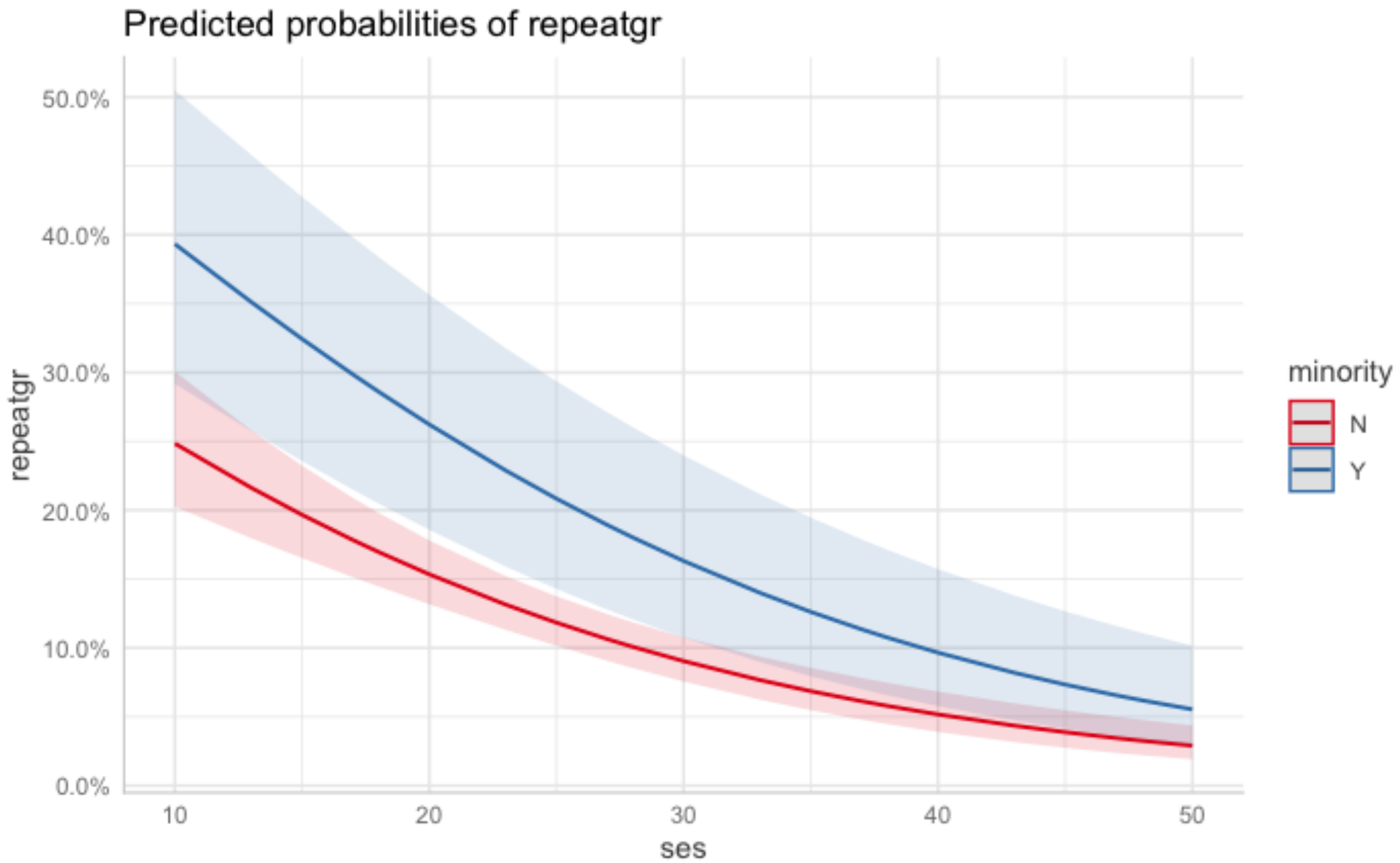
repeated a grade: yes / no

```
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()
```

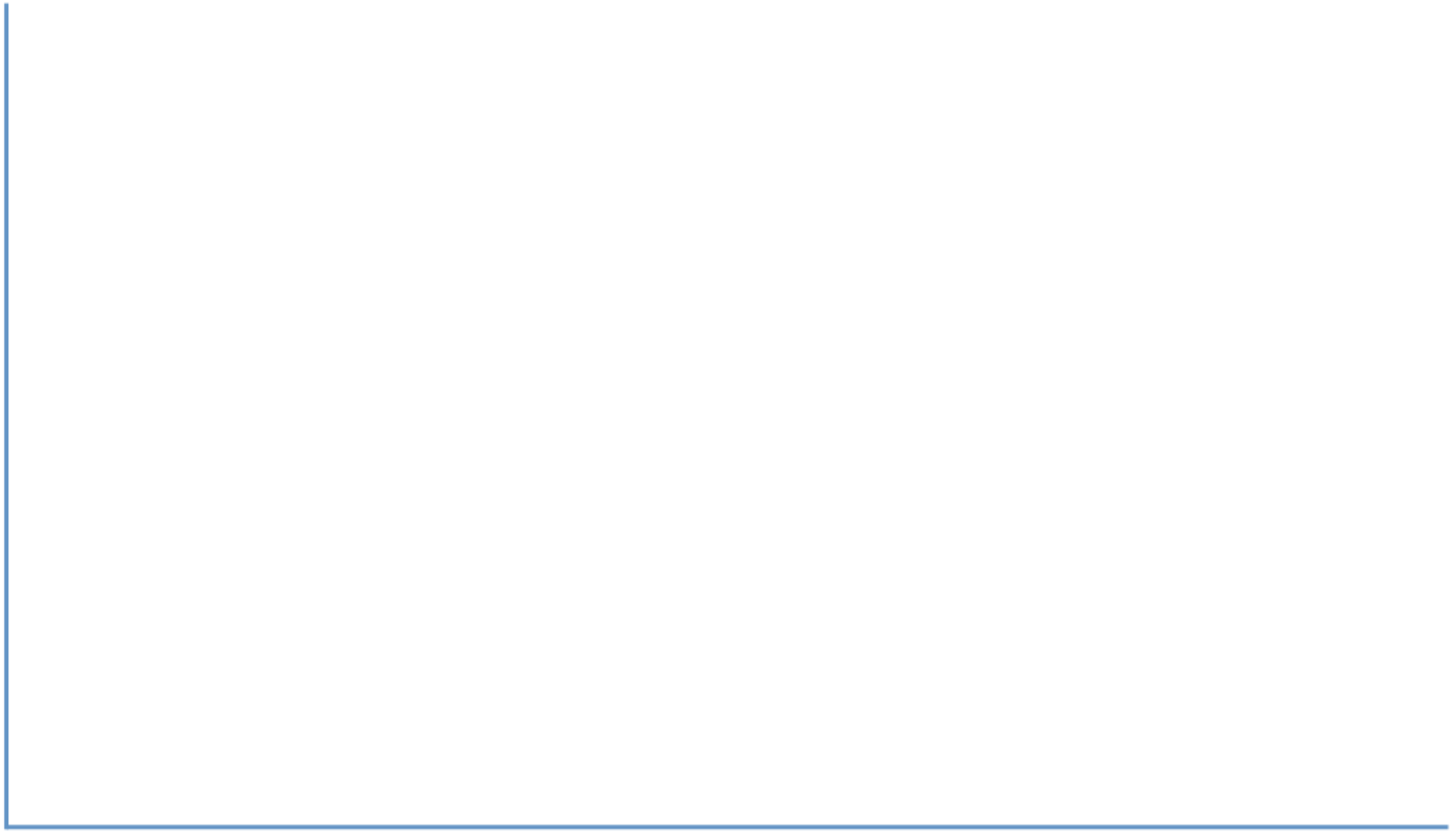


Summary

- 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!