

Generalized linear model



O COLLABORATIVE PLAYLIST
psych252
<https://tinyurl.com/psych252spotify24>

PLAY ...

A screenshot of a Spotify interface. On the left, there's a decorative graphic of blue bars with black outlines and hands reaching up from behind them. The main area has a dark background. At the top, it says "O COLLABORATIVE PLAYLIST". Below that is the title "psych252" in a large, bold, white font. Underneath the title is a URL: "<https://tinyurl.com/psych252spotify24>". At the bottom, there's a green "PLAY" button and a small "..." icon.

03/01/2024

Logistics

Homework 6

Linear mixed effects models

My name and the names of the people I have worked with go here

2024-02-28 14:59:22.573914

Instructions

This homework is due by **Thursday, March 7th, 8:00pm.**

Note:

- When asked to report results, please do so like you would in a scientific article (see examples from lectures, as well as in ‘Reporting Results.pdf’ on Canvas under Files > papers).
- Some code chunks contain some skeleton code. The code chunk option for these chunks is set to `eval=F` so that knitting the RMarkdown document doesn’t throw any errors. Make sure to set these chunks to `eval=T` when you knit your homework, so that your calculations are shown in the pdf.
- Make sure to show the results of your calculations in the knitted pdf, for example, by using the `print()` function at the end of a code chunk.
- Some questions ask for a short written response as indicated by this prompt: **Your answer:**

**submit both the rendered pdf and the
RMarkdown document**

Reporting results

- When we ask you to report results in the homework, try and do this like you would do in a scientific paper (reporting the statistical test, test-statistic, p-value, effect size).

The screenshot shows a course page from Stanford University's website. The title bar indicates the course is 'W21-PSYCH-252-01'. The main content is a PDF titled 'Reporting Results.pdf' from the University of Washington Psychology Writing Center. The PDF is page 1 of 3. It contains information about reporting statistical results in APA format, including a section on the goal of the results section and instructions for presenting multiple numerical results.

Stanford

Reporting Results.pdf

W21-PSYCH-252-01 > Files > papers

Download Info Close

Home

Account

Dashboard

Courses

Calendar

Inbox

Page 1 of 3

University of Washington
Psychology Writing Center
<http://www.psych.uw.edu/psych.php?p=339>

Box 351525
psywc@uw.edu
(206) 685-8278

Reporting Results of Common Statistical Tests in APA Format

The goal of the results section in an empirical paper is to report the results of the data analysis used to test a hypothesis. The results section should be in condensed format and lacking interpretation. Avoid discussing why or how the experiment was performed or alluding to whether your results are good or bad, expected or unexpected, interesting or uninteresting. This document is specifically about how to report statistical results. Refer to our handout "Writing an APA Empirical (lab) Report" for details on writing a results section.

Every statistical test that you report should relate directly to a hypothesis. Begin the results section by restating each hypothesis, then state whether your results supported it, then give the data and statistics that allowed you to draw this conclusion.

If you have multiple numerical results to report, it's often a good idea to present them in a figure (graph) or a table (see our handout on APA table guidelines).

Files > papers > Reporting results.pdf

Plan for today

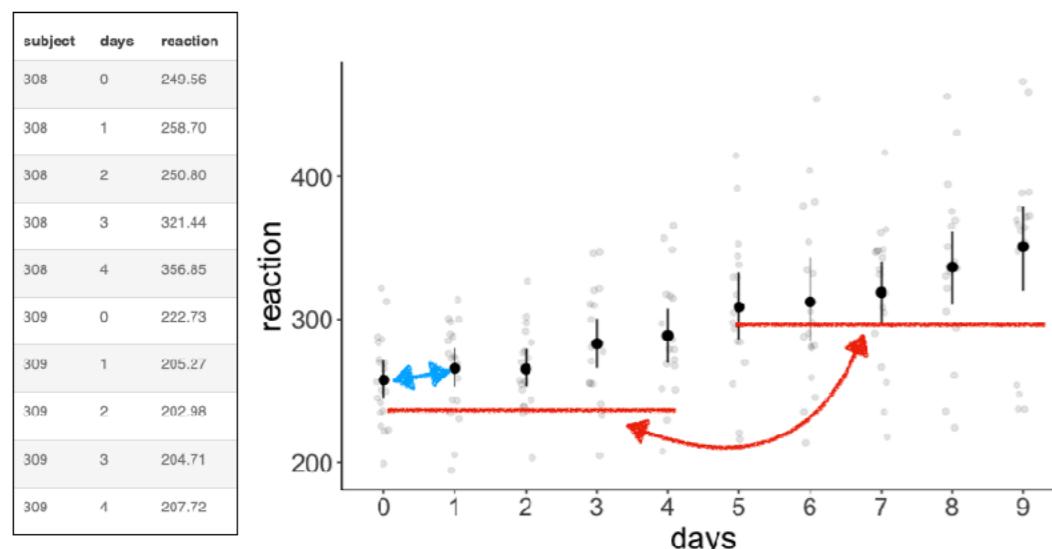
- Quick Recap
- `lmer()` standard operating procedures
- Generalized linear model
 - Logistic regression
 - interpreting the model output
 - fitting and reporting models
 - mixed effects logistic regression

Quick recap

Quick recap: Some more examples

Testing specific hypotheses with linear contrasts

1. Is there a significant difference between day 0 and day 1?
2. Is there a significant difference between the days 0-4 and days 5-9?



14

Sleep data

```
fit = lmer(formula = reaction ~ 1 + days + (1 | subject),
           data = df.sleep %>%
             mutate(days = as.factor(days)))
contrast = list(first_vs_second = c(-1, 1, rep(0, 8)),
               early_vs_late = c(rep(-1, 5)/5, rep(1, 5)/5))
fit %>%
  emmeans(specs = "days",
          contr = contrast) %>%
  pluck("contrasts")
```

contrast estimate SE df t.ratio p.value
first_vs_second 7.82 10.10 156 0.775 0.4398
early_vs_late 53.66 4.65 155 11.534 <.0001

Degrees-of-freedom method: kenward-roger.

<https://aosmith.rbind.io/2019/03/25/getting-started-with-emmeans/>

fit the model

define the contrasts

test the contrasts

days	reaction
0	257.54
1	255.73

index	reaction
early	271.67
late	325.39

16

Quick recap: Some more examples

Weight loss data

```
1 fit = aov_ez(id = "id",
2                 dv = "score",
3                 between = "exercises",
4                 within = c("diet", "timepoint"),
5                 data = df.weightloss)
```

Anova Table (Type III tests)

	Effect	df	MSE	F ges	p.value
1	exercises	1, 22	1.84	38.77 ***	.284 <.001
2	diet	1, 22	0.65	7.91 *	.028 .010
3	exercises:diet	1, 22	0.65	51.70 ***	.157 <.001
4	timepoint	1.74, 38.26	1.48 82.20 ***	.541 <.001	
5	exercises:Timepoint	1.74, 38.26	1.48 26.22 ***	.274 <.001	
6	diet:timepoint	1.61, 35.44	1.92 0.78	.013 .439	
7	exercises:diet:timepoint	1.61, 35.44	1.92 9.97 ***	.147 <.001	

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

main effects and interactions

df.weightloss				
id	diet	exercises	timepoint	score
1	no	no	t1	10.43
1	no	no	t2	13.21
1	no	no	t3	11.59
1	yes	no	t1	10.20
1	yes	no	t2	12.51
1	yes	no	t3	14.60
2	no	no	t1	11.59

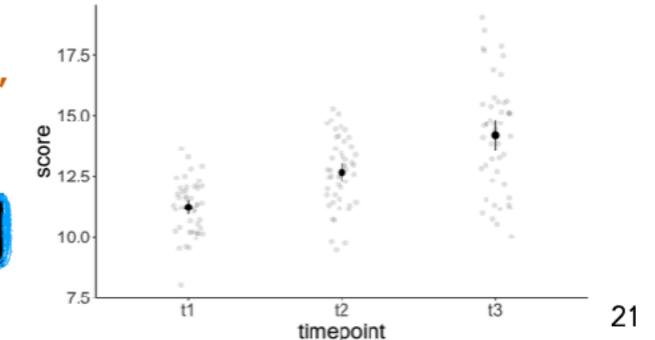
Weight loss data

1. Is the score at the third time point different from the other two time points?
2. Is there a linear increase across time points?

```
1 fit = aov_ez(id = "id",
2                 dv = "score",
3                 between = "exercises",
4                 within = c("diet", "timepoint"),
5                 data = df.weightloss)
6
7 contrasts = list(first_two_vs_last = c(-0.5, -0.5, 1),
8                   linear_increase = c(-1, 0, 1))
9
10 fit %>%
11   emmeans(spec = "timepoint",
12             contr = contrasts)
```

contrast	estimate	SE	df	t.ratio	p.value
first two vs last	2.24	0.200	4	11.194	<.0001
linear_increase	2.97	0.231	4	12.820	<.0001

df.weightloss				
id	diet	exercises	timepoint	score
1	no	no	t1	10.43
1	no	no	t2	13.21
1	no	no	t3	11.59
1	yes	no	t1	10.20
1	yes	no	t2	12.51
1	yes	no	t3	14.60
2	no	no	t1	11.59



20

21

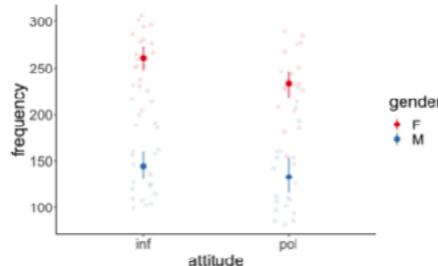
Quick recap: Some more examples

Politeness

Was there an effect of gender and attitude on pitch?

```
1 lmer(formula = frequency ~ 1 + attitude * gender + (1 | subject) + (1 | scenario),  
2       data = df.politeness) %>%  
3   joint_tests()
```

model term	df1	df2	F.ratio	p.value
attitude	1	69.04	12.497	0.0007
gender	1	4.00	26.578	0.0067
attitude:gender	1	69.04	1.969	0.1650



main effect of attitude, main effect of gender, no significant interaction effect

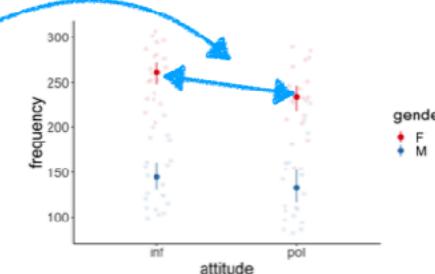
Politeness

Was there a difference between informal and polite speech for female participants?

```
1 fit = lmer(formula = frequency ~ 1 + attitude * gender + (1 | subject) + (1 | scenario),  
2             data = df.politeness)  
3  
4 fit %>%  
5   emmeans(specs = pairwise ~ attitude + gender,  
6             adjust = "none")
```

contrast	estimate	SE	df	t.ratio	p.value
inf F - pol F	27.4	7.79	69.00	3.517	0.0008
inf F - inf M	116.2	21.73	4.56	5.348	0.0040
inf F - pol M	128.0	21.77	4.59	5.881	0.0027
pol F - inf M	88.8	21.73	4.56	4.087	0.0115
pol F - pol M	100.6	21.77	4.59	4.623	0.0071
inf M - pol M	11.8	7.90	69.08	1.497	0.1390

Degrees-of-freedom method: kenward-roger



yes, there was significant difference in pitch for women between informal and formal speech

25

26

Politeness

Was there an effect of gender and attitude on pitch?

ANOVA

```
1 aov_ez(id = "subject",  
2         dv = "frequency",  
3         between = "gender",  
4         within = "attitude",  
5         data = df.politeness)
```

```
More than one observation per cell, aggregating the data using  
mean (i.e., fun_aggregate = mean). Missing values for following  
ID(s):  
M4  
Removing those cases from the analysis.Anova Table (Type 3 tests)  
Response: frequency  
Effect          df      MSE      F ges p.value  
1   gender 1, 3 1729.42 17.22 * .851 .025  
2   attitude 1, 3  3.65 22.30 ** .730 < .001  
3 gender:attitude 1, 3  3.65 22.30 ** .051 .019  
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ignores variation between scenarios, and just takes the mean

interaction effect

LMER

```
1 lmer(formula = frequency ~ 1 + attitude * gender +  
2       (1 | subject) + (1 | scenario),  
3       data = df.politeness) %>%  
4   joint_tests()
```

model term	df1	df2	F.ratio	p.value
attitude	1	69.04	12.497	0.0007
gender	1	4.00	26.578	0.0067
attitude:gender	1	69.04	1.969	0.1650

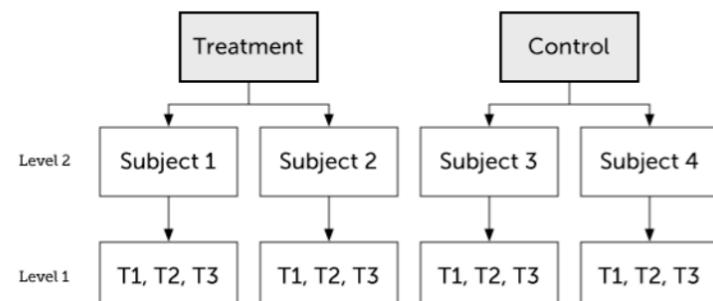
no interaction effect

27

9

Quick recap: different random effects

Graphical representation

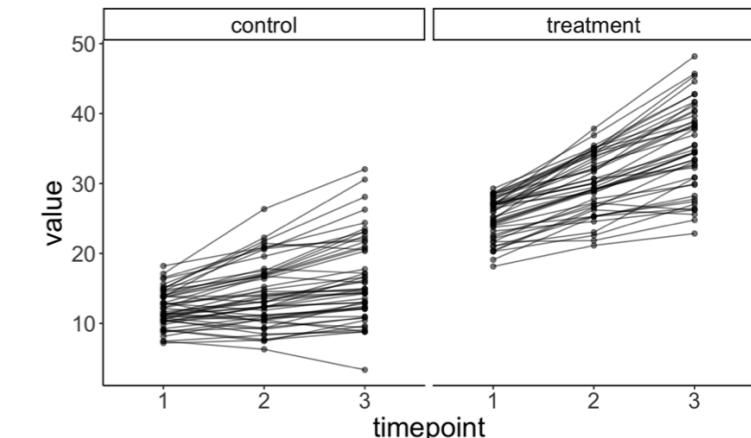


Simulate data

```

1 set.seed(1)
2 n_participants = 100
3 n_timepoints = 3
4 n_conditions = 2
5 p_condition = 0.5
6 b0 = 10 # intercept
7 b1 = 2 # condition
8 b2 = 2 # time
9 b3 = 3 # interaction
10 sd_participant = 2
11 sd_time = 2
12 sd_residual = 1
13
14 df.data = tibble(participant = rep(1:n_participants, each = n_timepoints),
15                   timepoint = rep(1:n_timepoints, times = n_participants),
16                   intercept_participant = rep(rnorm(n_participants, sd = sd_participant),
17                                         each = n_timepoints),
18                   time_participant = rep(rnorm(n_participants, sd = sd_time),
19                                         each = n_timepoints)) %>%
20
21 group_by(participant) %>%
22 mutate(condition = rbinom(n = 1, size = 1, prob = p_condition)) %>%
23 ungroup() %>%
24 mutate(value = b0 + intercept_participant +
25       b1 * condition +
26       b2 * time_participant * timepoint +
27       b3 * condition * timepoint +
28       rnorm(n_participants * n_timepoints, sd = sd_residual))
  
```

Plot data



40

41

Fit the model

```

1 fit = lmer(formula = value ~ 1 + condition * timepoint + (1 + timepoint | participant),
2             data = df.data)
  
```

```

Linear mixed model fit by REML ['lmerMod']
Formula: value ~ 1 + condition * timepoint + (1 + timepoint | participant)
Data: df.data

REML criterion at convergence: 1360.3

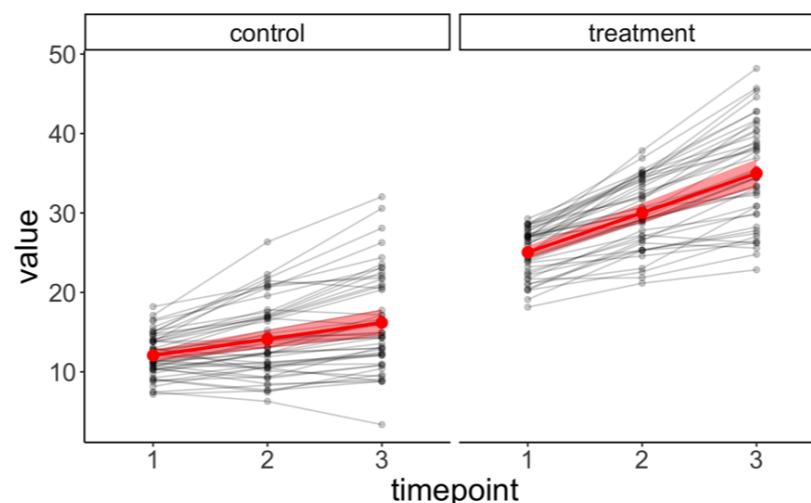
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-2.14633 -0.46360  0.03902  0.42302  2.82945 

Random effects:
 Groups   Name        Variance Std.Dev. Corr  
 participant (Intercept) 3.190   1.786    
          timepoint   3.831   1.957   -0.06  
 Residual           1.149   1.072    
Number of obs: 300, groups: participant, 100

Fixed effects:
            Estimate Std. Error t value
(Intercept) 10.0101   0.3328 30.079
condition    10.0684   0.4854 20.741
timepoint    2.0595   0.2883  7.143
condition:timepoint 2.9090   0.4205  6.917

Correlation of Fixed Effects:
 (Int) condtn timptn
condtn -0.686
timptn -0.266  0.182
cndtn:timptn 0.182 -0.266 -0.686
  
```

Visualize model predictions (overall)

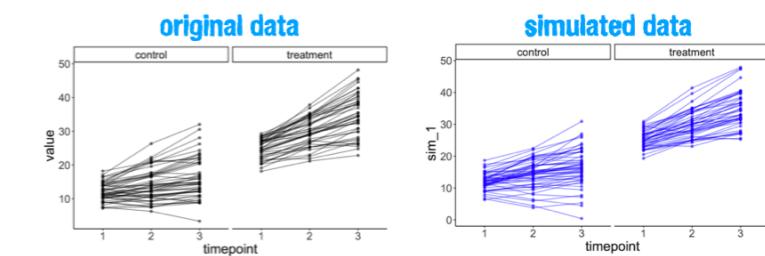


42

Simulate new data from the model

```

1 set.seed(1)
2 fit %>%
3   simulate() %>%
4   bind_cols(df.data) %>%
5   mutate(condition = factor(condition,
6                             levels = c(0, 1),
7                             labels = c("control", "treatment")),
8         timepoint = as.factor(timepoint)) %>%
9
10 ggplot(data = .,
11         mapping = aes(x = timepoint,
12                         y = sim_1,
13                         group = participant)) +
14   geom_point(alpha = 0.5,
15               color = "blue") +
16   geom_line(alpha = 0.5,
17               color = "blue") +
18   facet_grid(~ condition) +
19   labs(x = "timepoint")
  
```



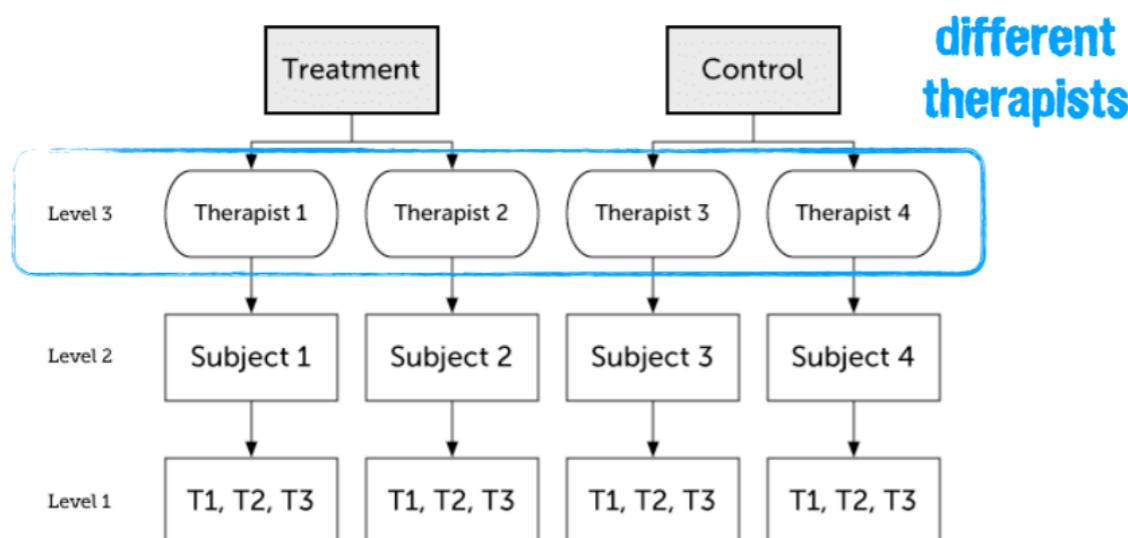
44

45

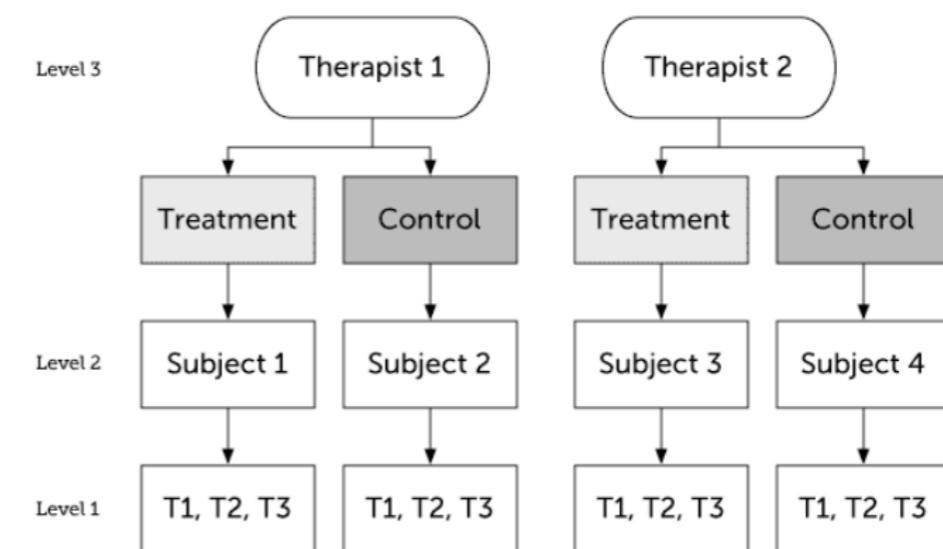
10

Quick recap: different random effects

Graphical representation



Other possible model structures



```
fit = lmer(formula = value ~ 1 + condition * timepoint +
           (1 + condition * timepoint | therapist) +
           (1 + timepoint | therapist:participant),
           data = df.data)
```

47

50

11

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

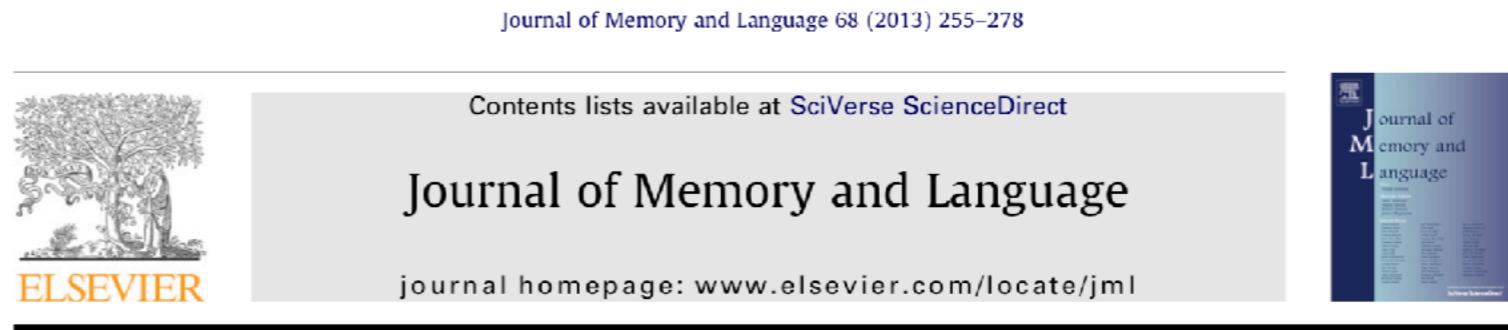
Content/Analysis Steps

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

[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)

What shall I include as random effects?

- mixed opinions on the topic
- go maximal!



Random effects structure for confirmatory hypothesis testing:
Keep it maximal



Dale J. Barr ^{a,*}, Roger Levy ^b, Christoph Scheepers ^a, Harry J. Tily ^c

^aInstitute of Neuroscience and Psychology, University of Glasgow, 58 Hillhead St., Glasgow G12 8QB, United Kingdom

^bDepartment of Linguistics, University of California at San Diego, La Jolla, CA 92093-0108, USA

^cDepartment of Brain and Cognitive Sciences, Massachusetts 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."

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 

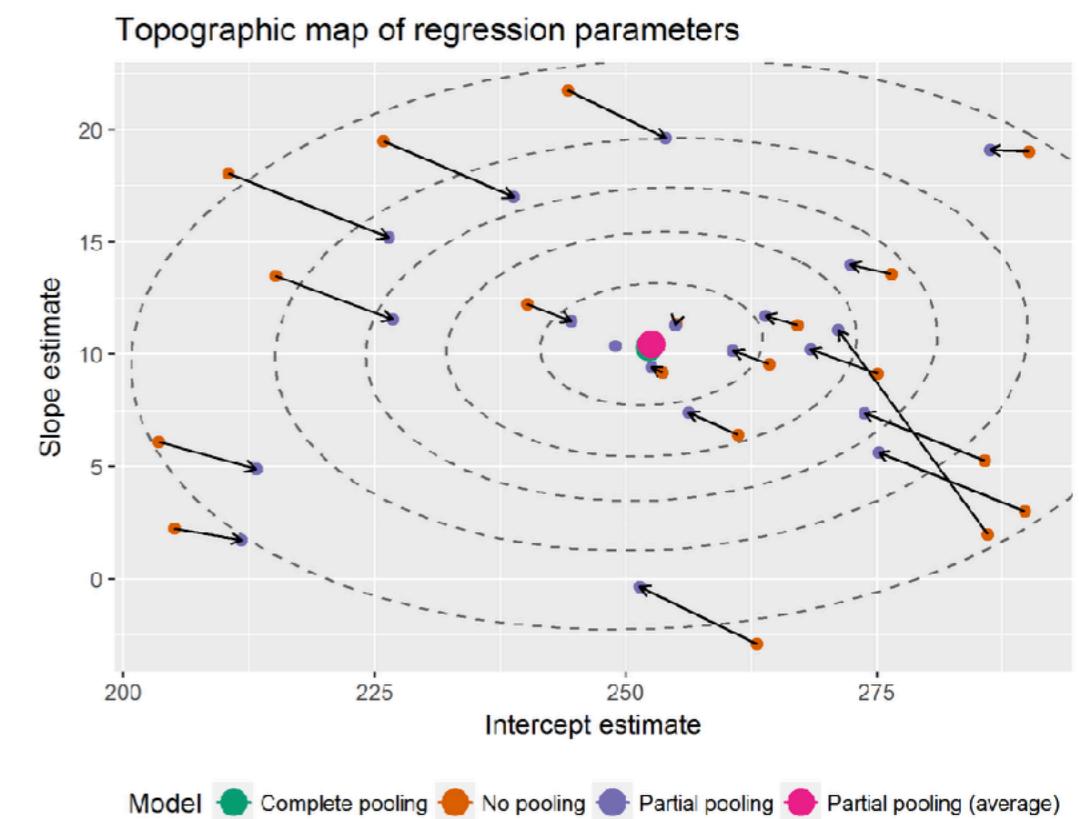
Random effects:
Groups   Name        Variance Std.Dev. Corr
subject (Intercept) 582.73   24.140
          days       35.03   5.919   0.07
Residual            649.36   25.483

Number of obs: 183, groups: subject, 20

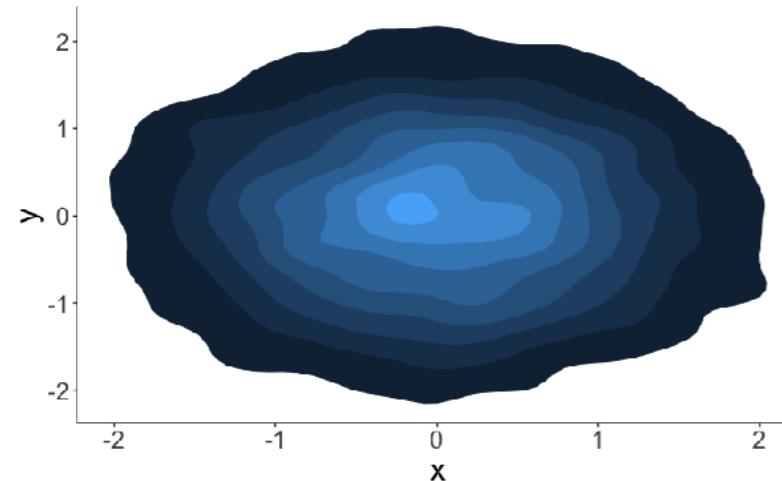
Fixed effects:
            Estimate Std. Error t value
(Intercept) 252.543    6.433  39.256
days         10.452    1.542   6.778

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

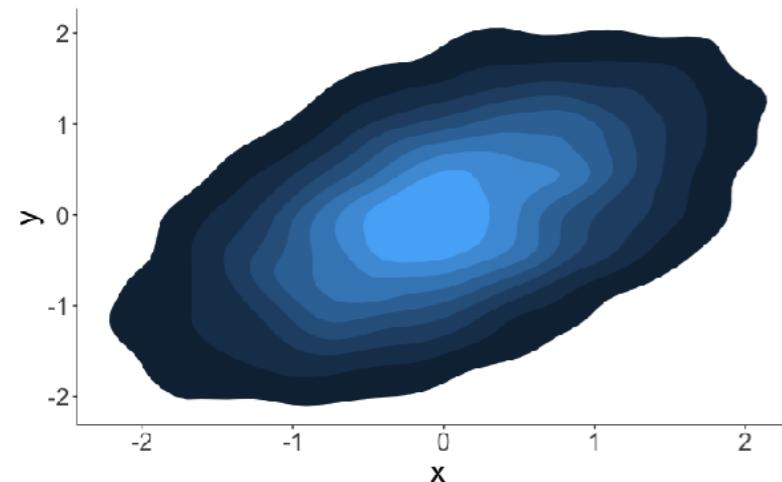
multivariate Gaussian



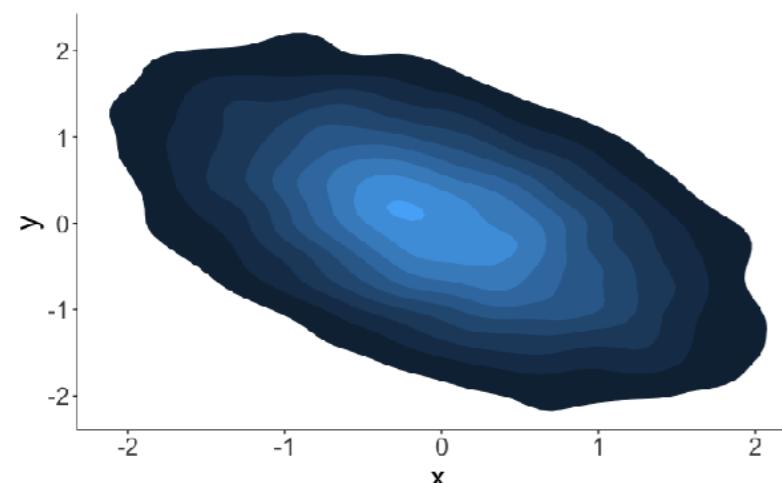
Remove the correlation component from your model



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 

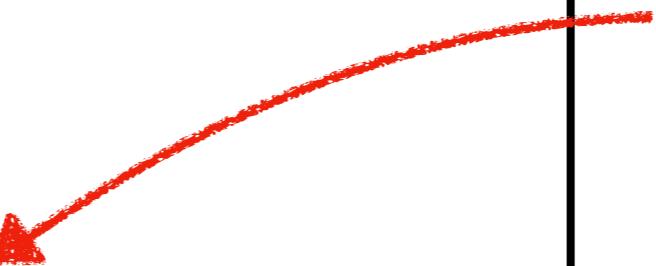
Random effects:
 Groups   Name        Variance Std.Dev.    
subject  days       35.88    5.99      
subject.1 (Intercept) 598.11   24.46    
Residual           647.90   25.45    
Number of obs: 183, groups: subject, 20

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
```

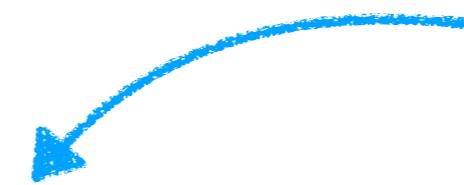
alternative syntax (doesn't model correlation between random effects)

independent Gaussians

What if lmer() fails to converge?

```
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

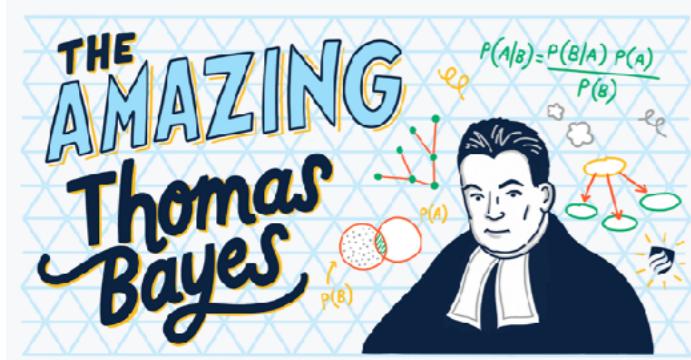
\$llik	bobyqa	Nelder_Mead	nlminbwrap
	-885.7239	-885.7239	-885.7239
	nloptwrap.NLOPT_LN_NELDERMEAD	nloptwrap.NLOPT_LN_BOBYQA	

\$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

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 (Florence	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 May Peel)	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 Achem)	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

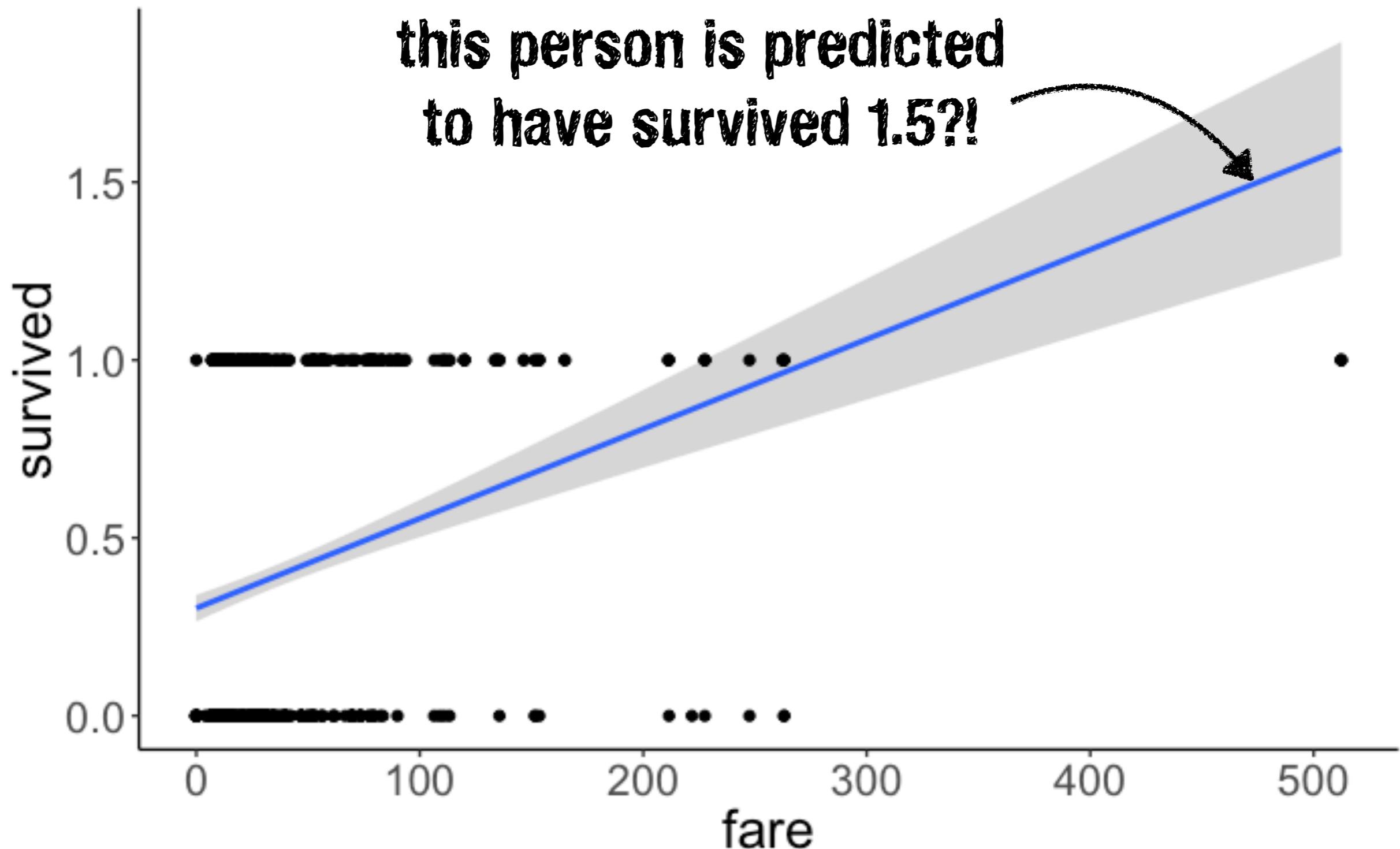
Residual standard error: 0.4705 on 889 degrees of freedom

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
 - ...



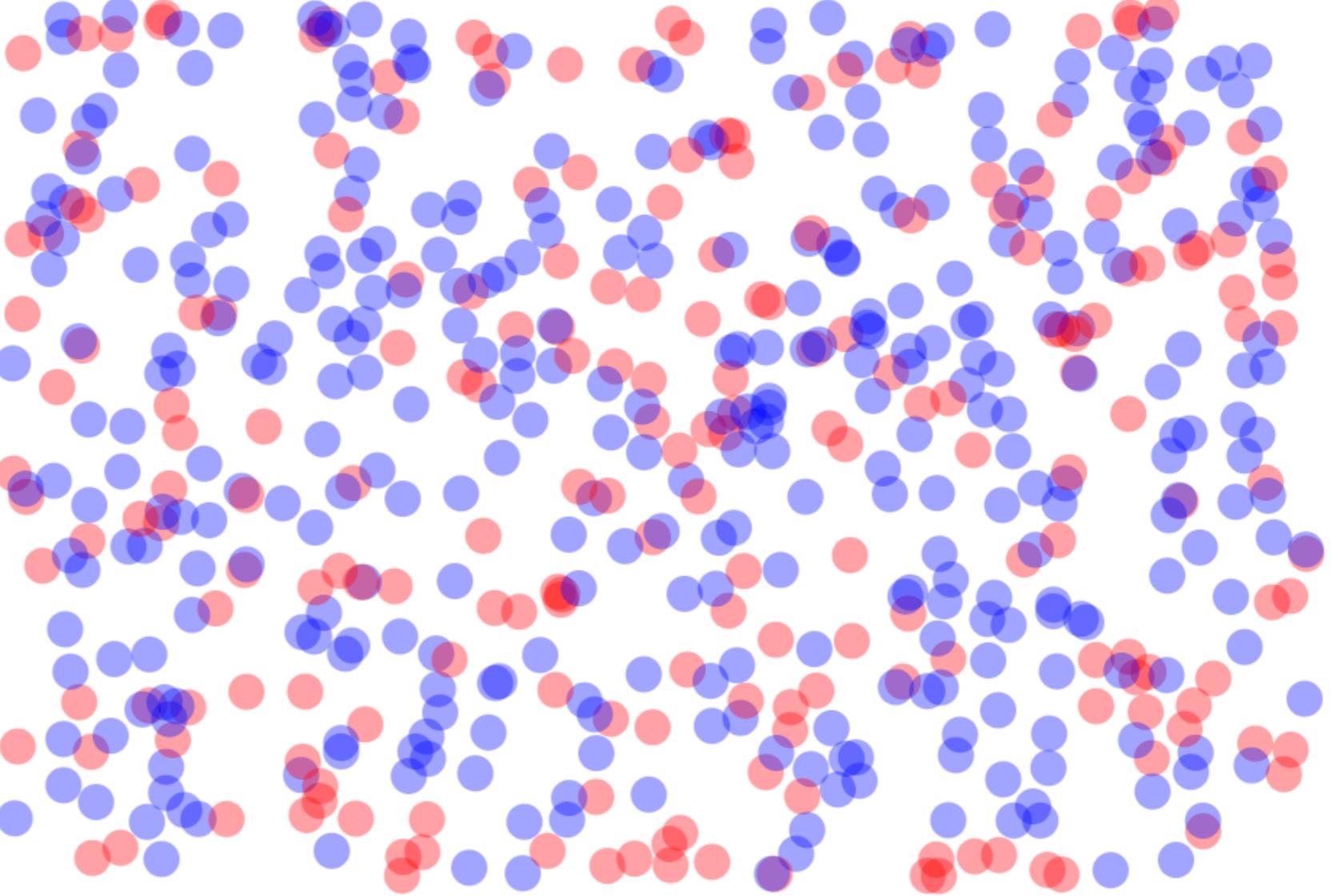
Logistic regression

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.



A scatter plot showing two classes of data points, represented by blue and red circles, forming a roughly circular pattern. The plot is used to illustrate binary classification metrics.

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

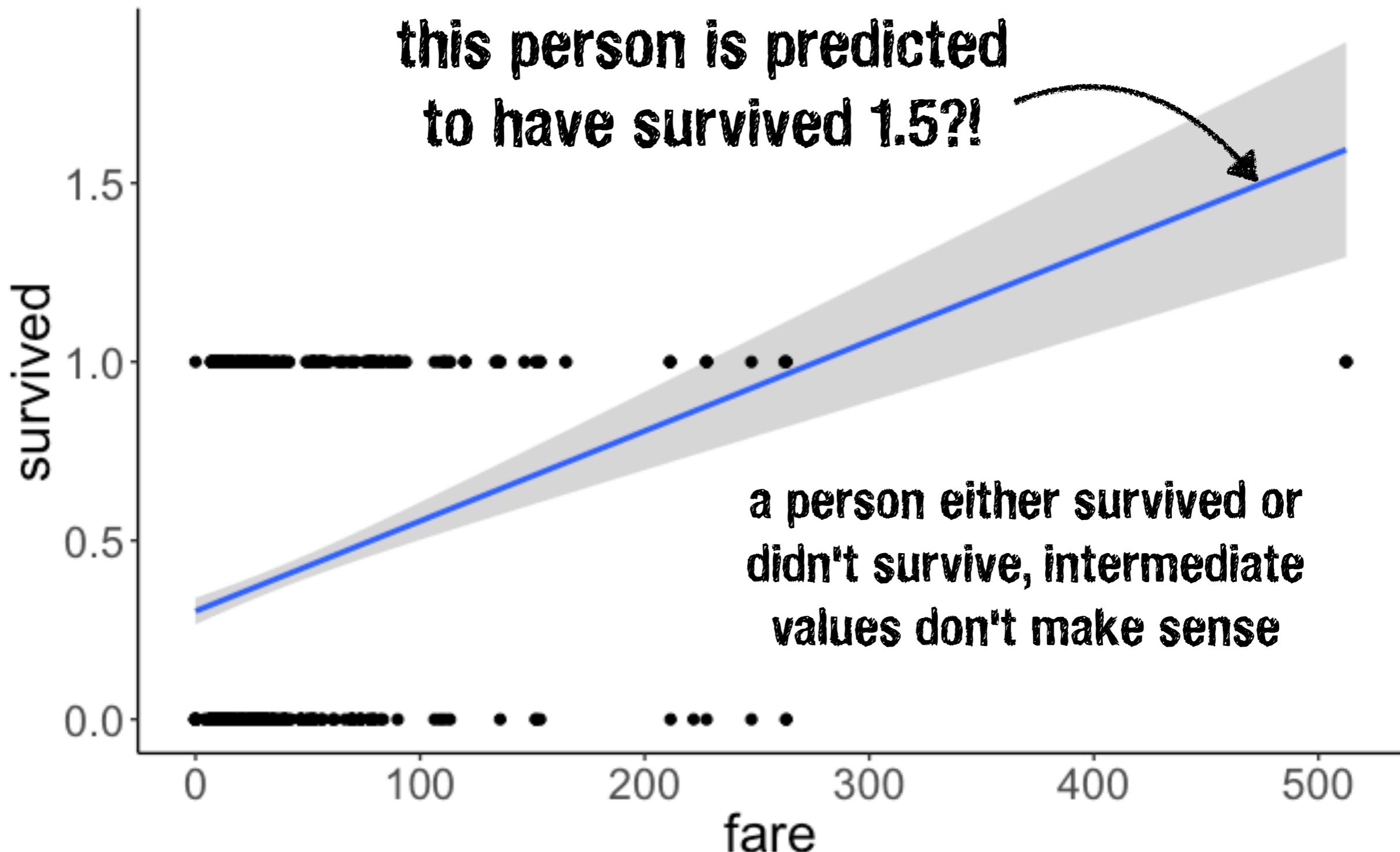
Is there a relationship between fare and survived?

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

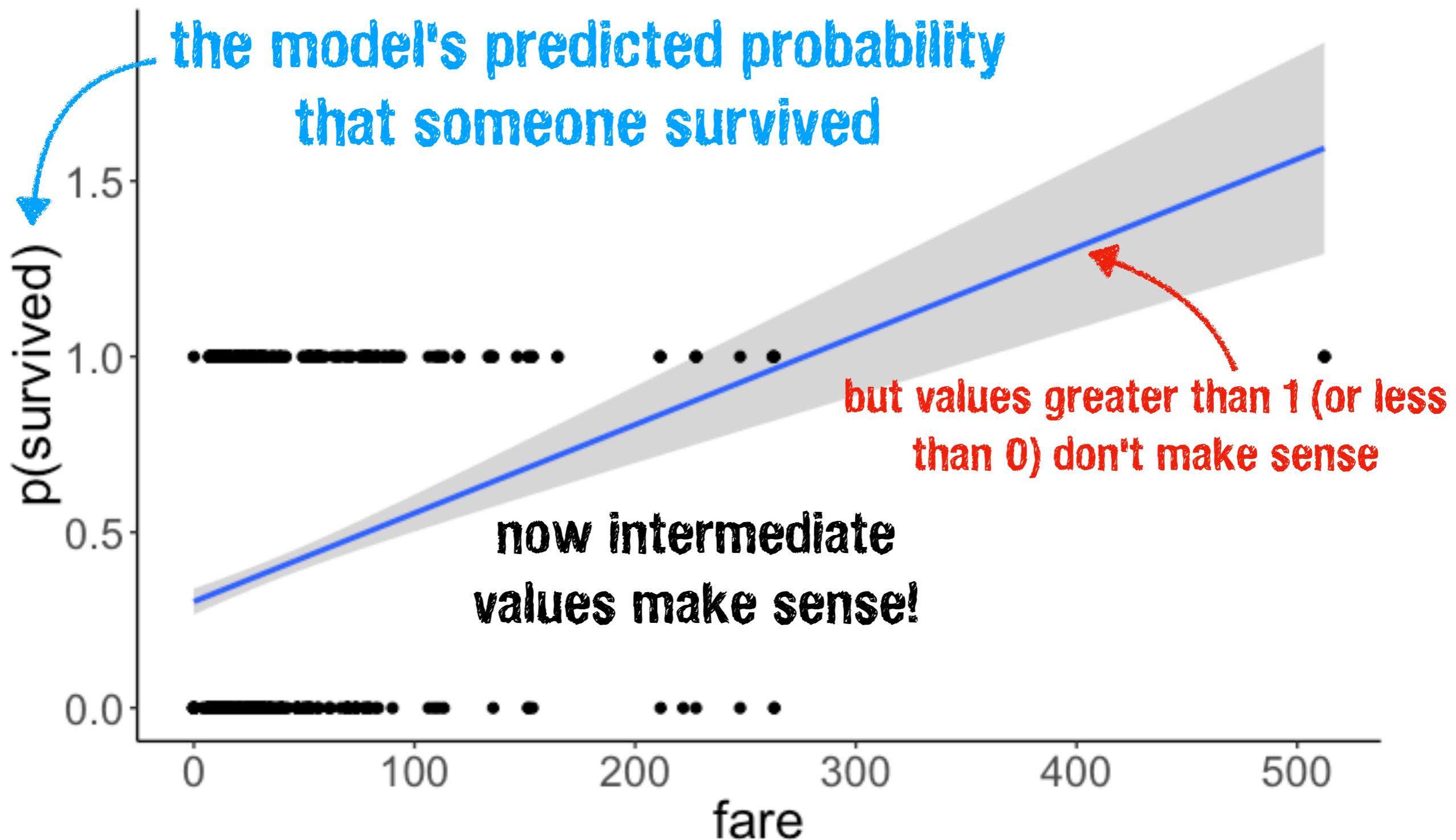
The fact that this class is called "**Generalized linear model**" suggests we can!

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

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

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

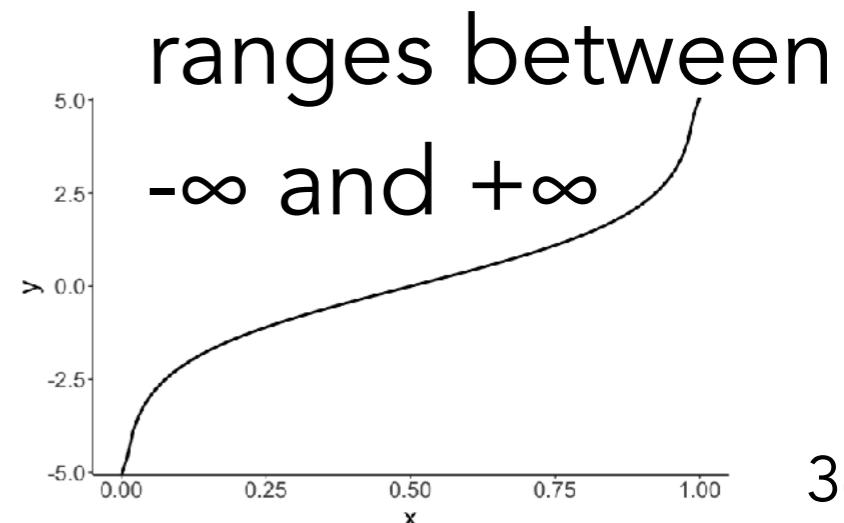
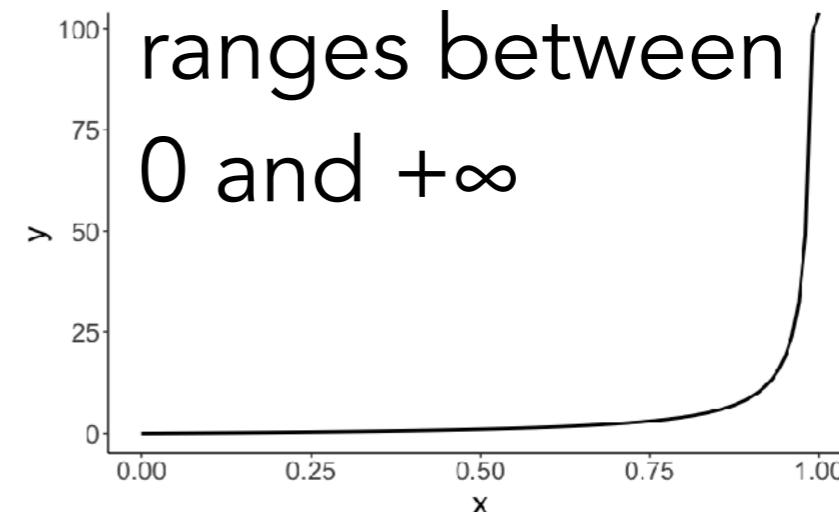
Step 1: Calculate the "odds"

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

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

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

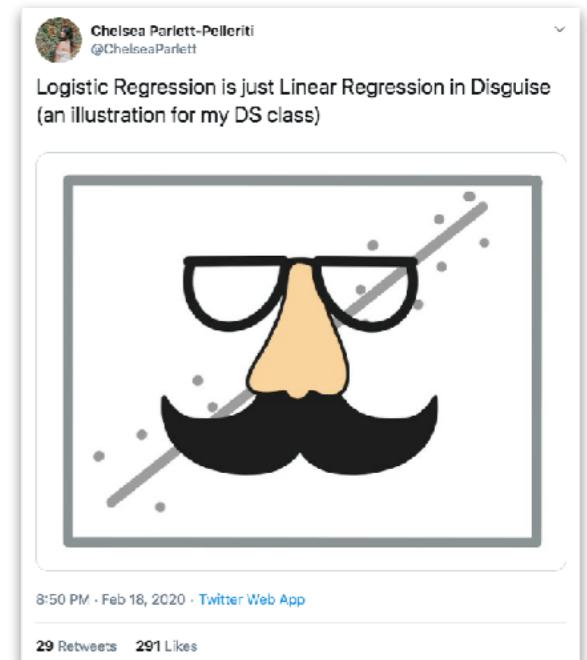
$$\pi_i = P(Y_i = 1)$$

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



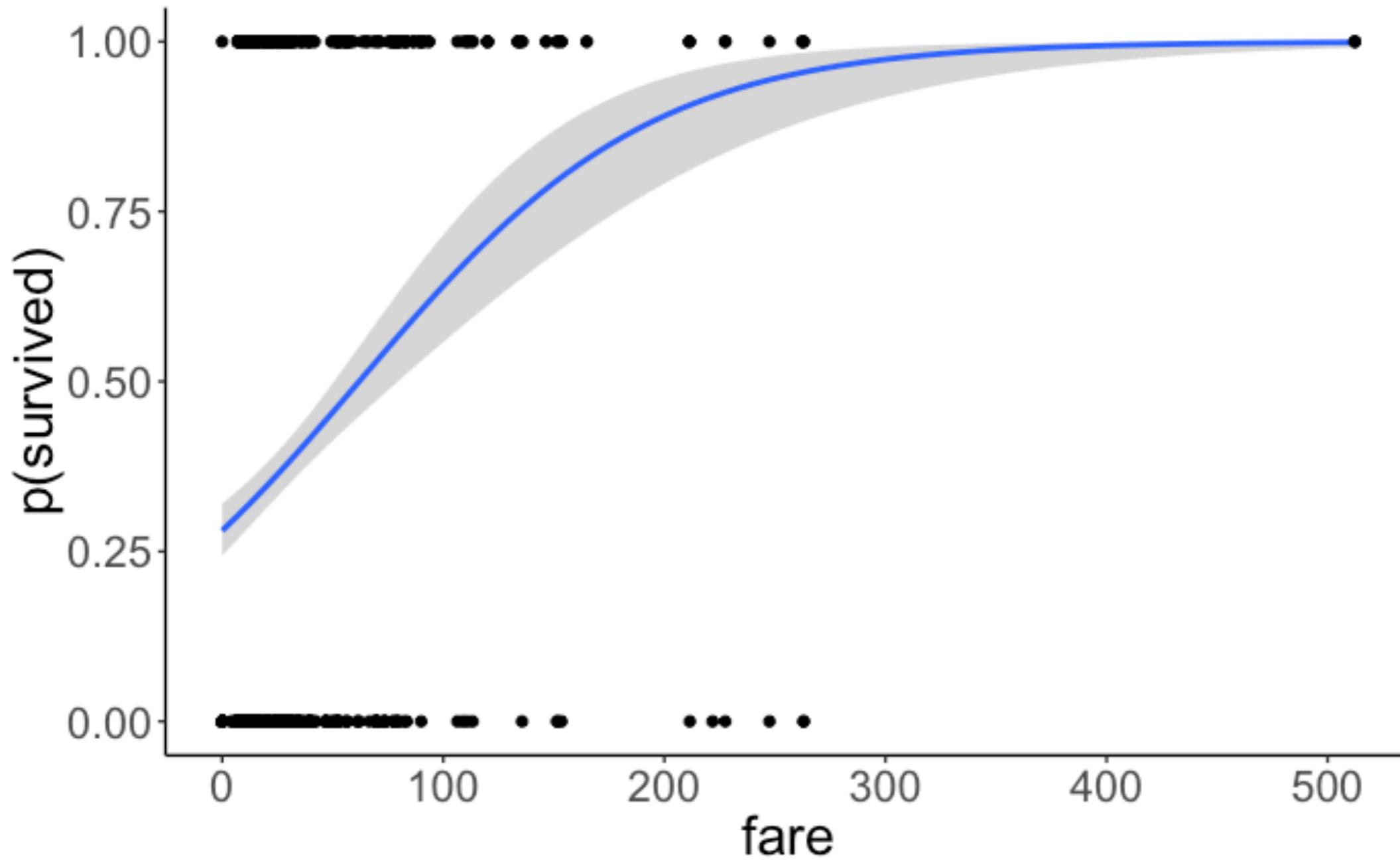
Fitting a logistic regression in R

```
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
```

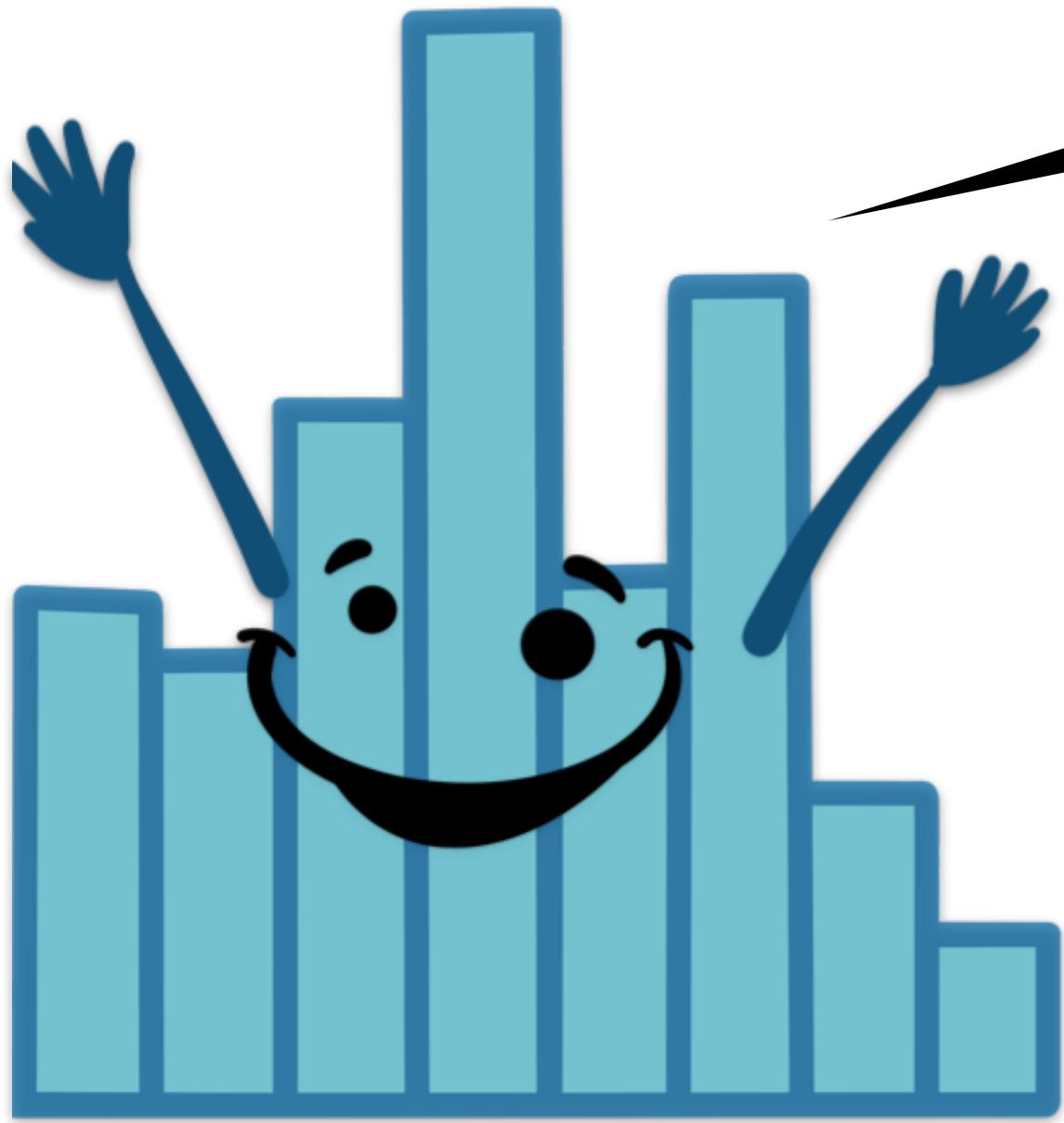
Visualize the model's predictions

```
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)")
```



02:00

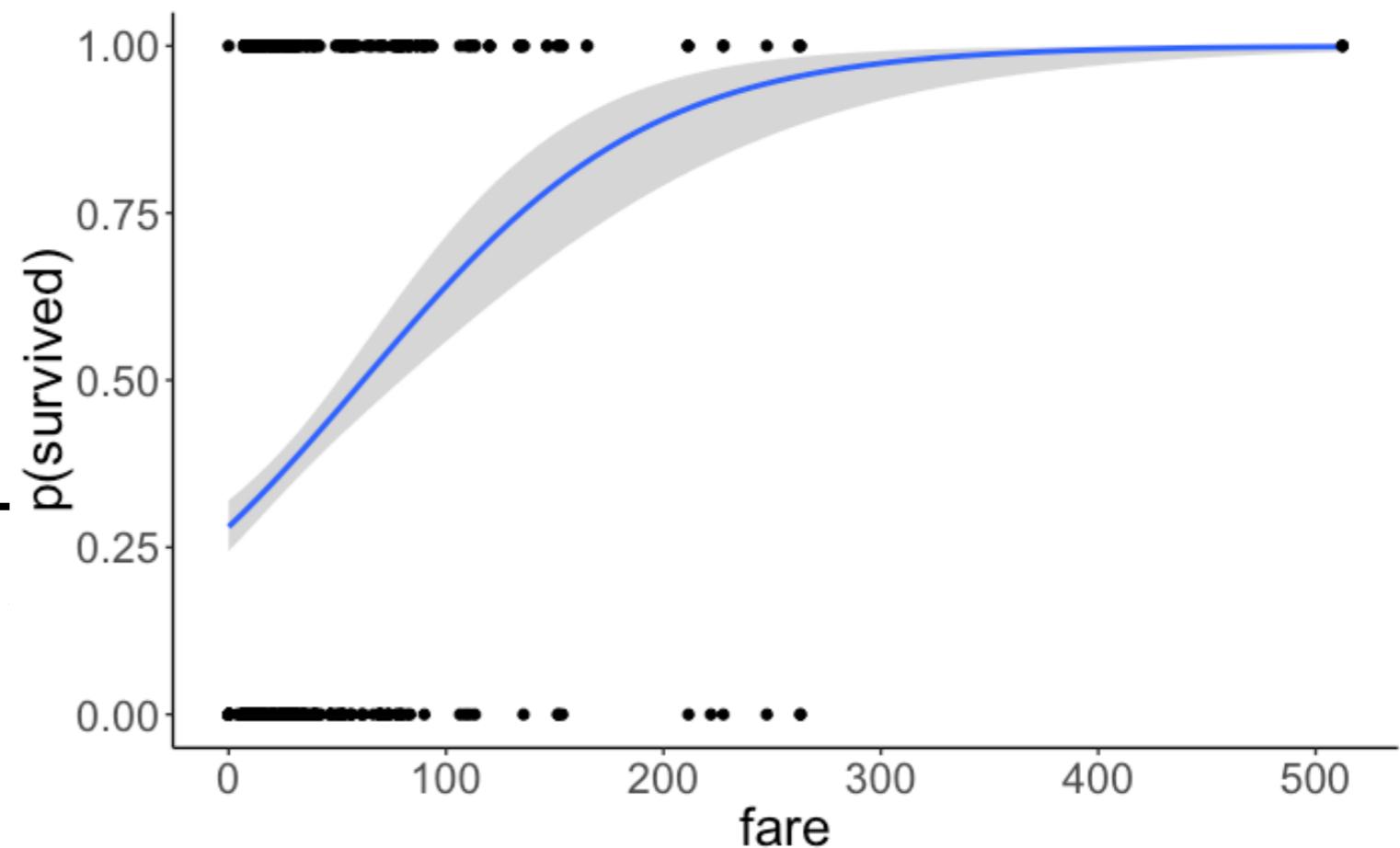
stretch break!



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)

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

another way to
specify the model

Interpreting the model output

inverse logit

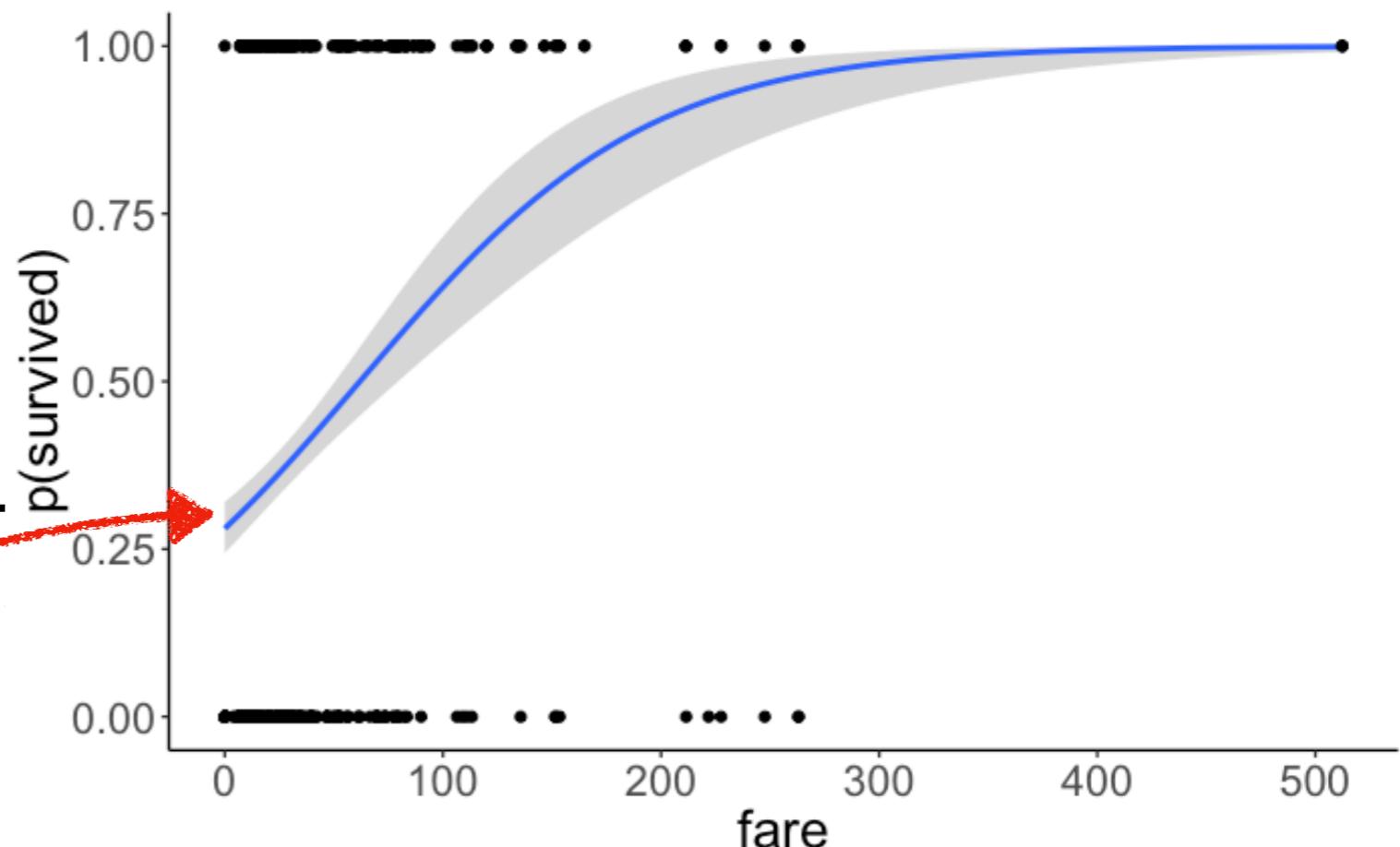
$$\pi = \frac{e^{-0.94}}{1 + e^{-0.94}} \approx 0.28$$

```
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)

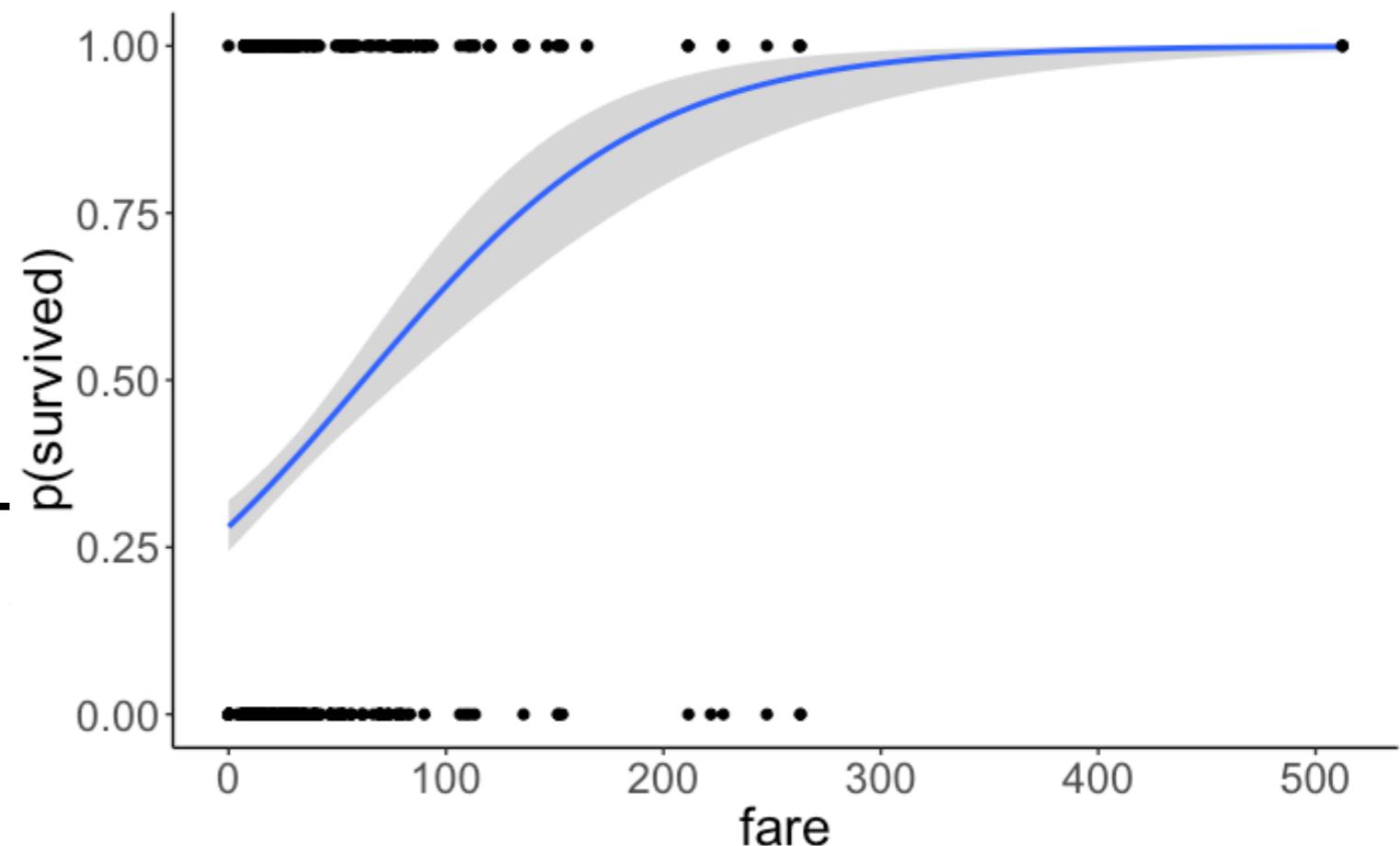
Null deviance: 1186.7 on 890 degrees of freedom
Residual deviance: 1117.6 on 889 degrees of freedom
AIC: 1121.6

Number of Fisher Scoring iterations: 4



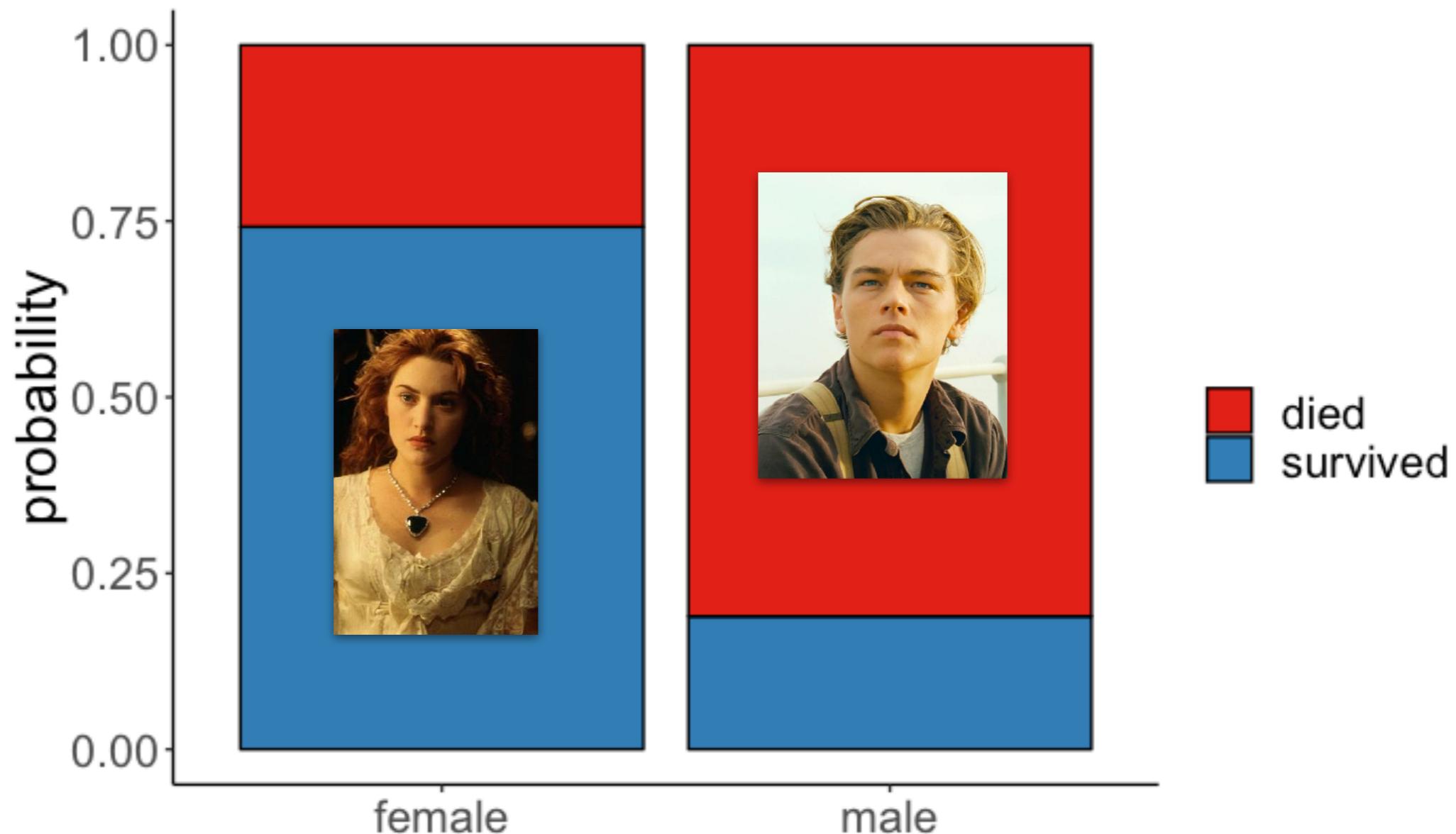
Interpreting the model output

```
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

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

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 $\text{sex} \equiv 0$:

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

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

Let's consider a binary predictor

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

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

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

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

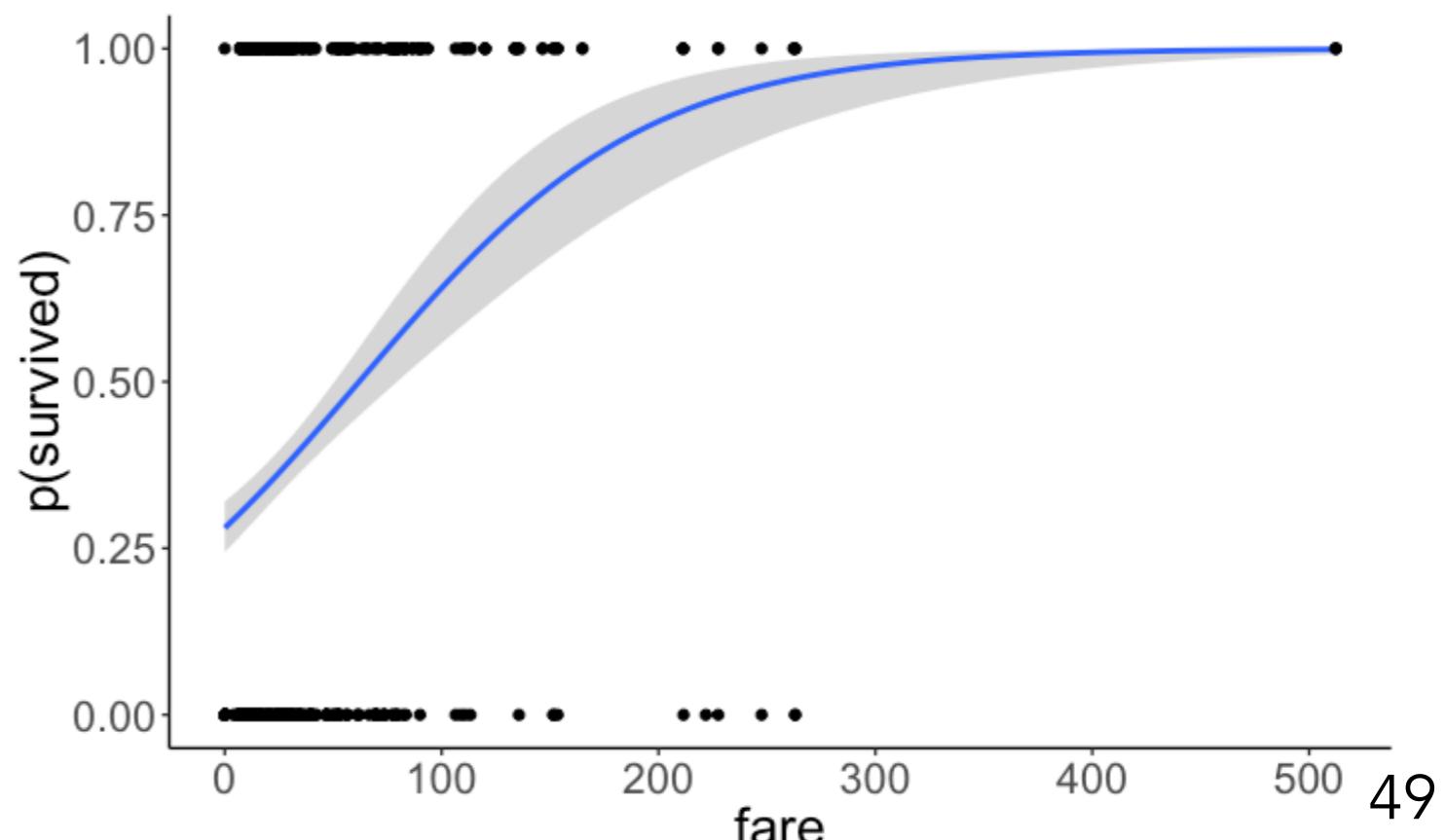
$$\ln\left(\frac{p(\text{survived})_i}{1 - p(\text{survived})_i}\right) = b_0 + b_1 \cdot \text{fare}_i$$

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



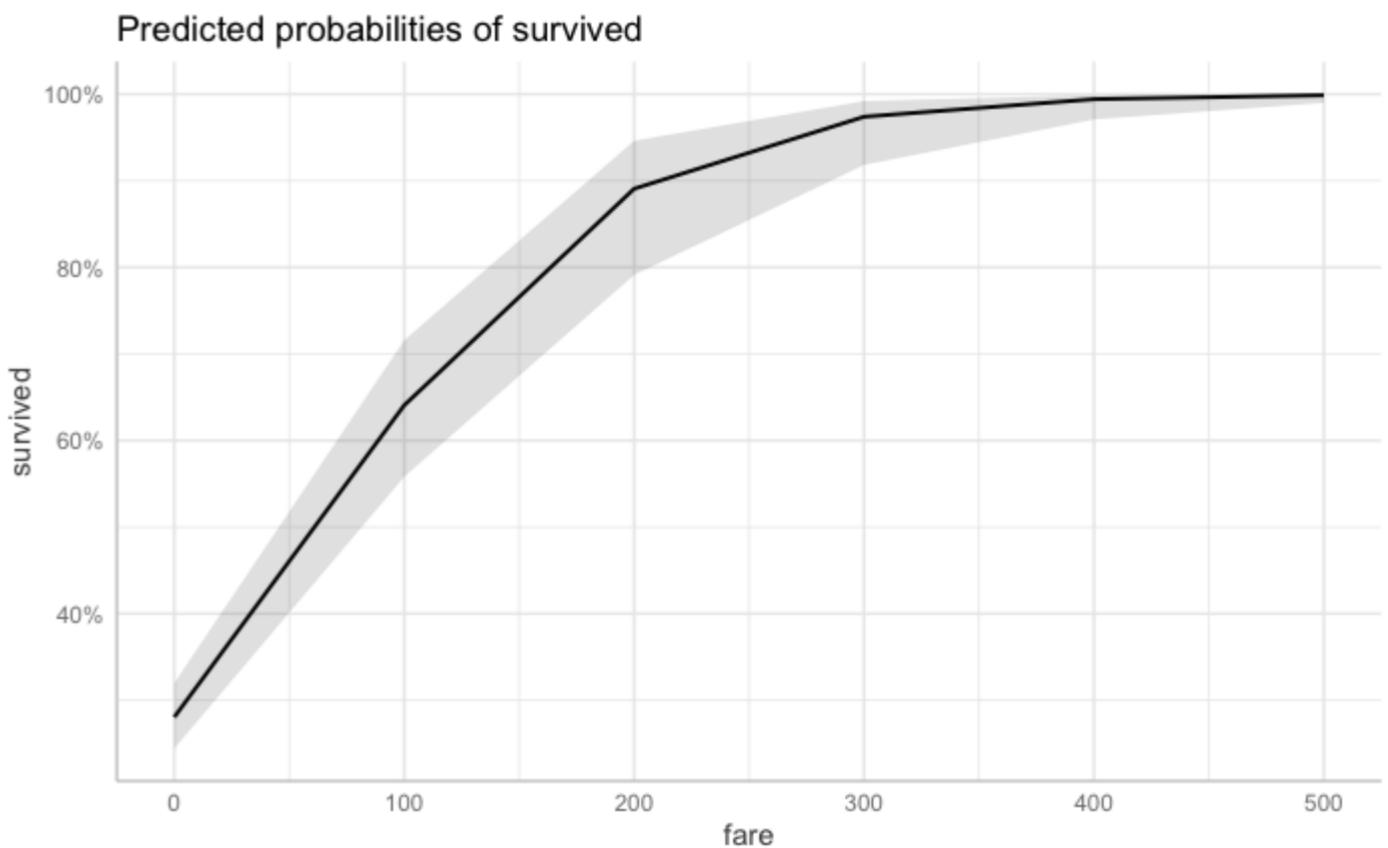
Do we have to do this by hand?



```
1 ggpredict(model = fit.glm,  
2           terms = "fare [0, 100, 200, 300, 400, 500]")
```

```
# Predicted probabilities of survived  
# x = fare
```

x	Predicted	95% CI
<hr/>		
0	0.28	[0.24, 0.32]
100	0.64	[0.56, 0.72]
200	0.89	[0.79, 0.95]
300	0.97	[0.92, 0.99]
400	0.99	[0.97, 1.00]
500	1.00	[0.99, 1.00]



Models with several predictors

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

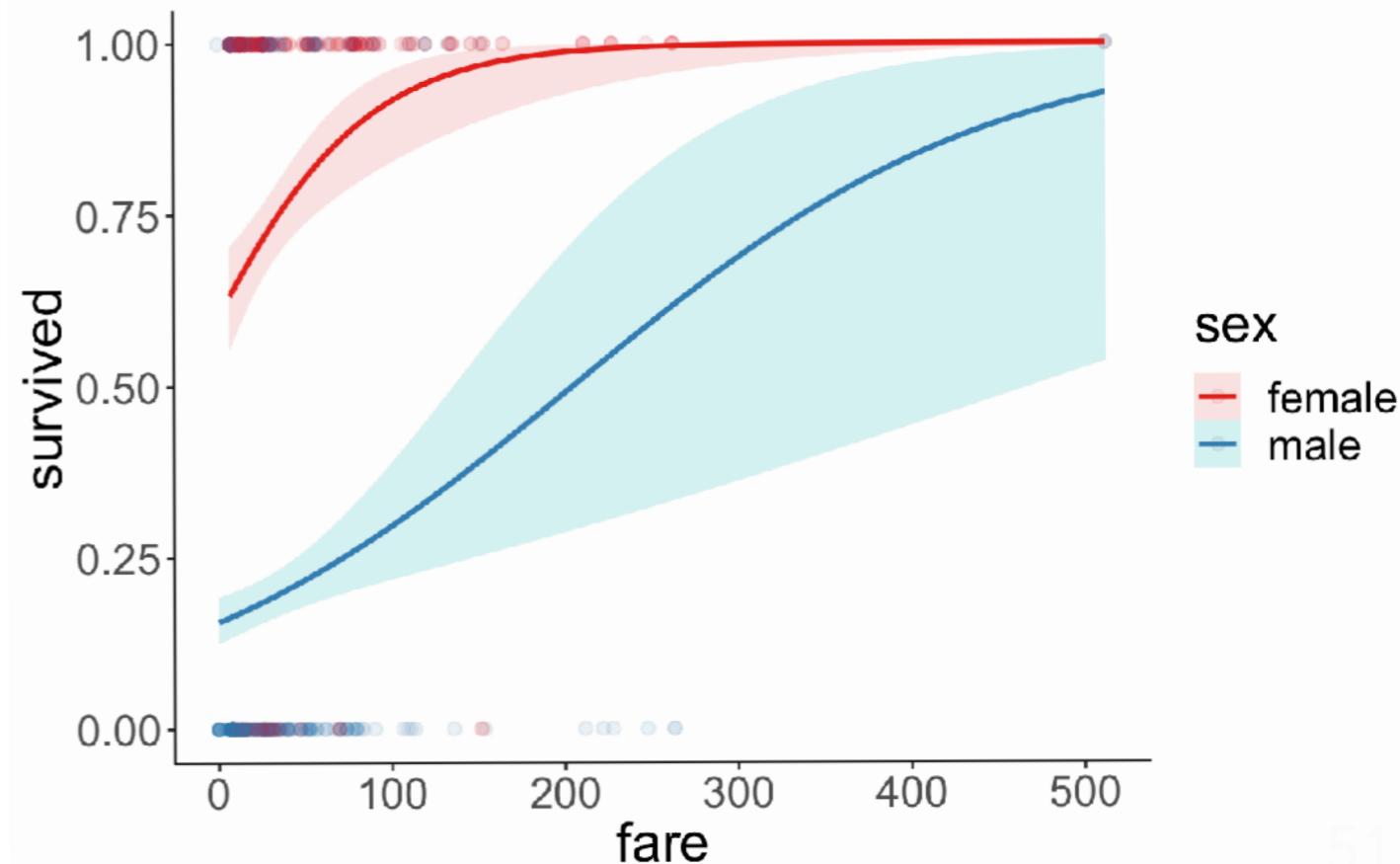
```
Coefficients:  
Estimate Std. Error z value Pr(>|z|)  
(Intercept) 0.647100 0.148502 4.358 1.32e-05 ***  
sexmale -2.422760 0.170515 -14.208 < 2e-16 ***  
fare 0.011214 0.002295 4.886 1.03e-06 ***  
---  
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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")
```



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

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

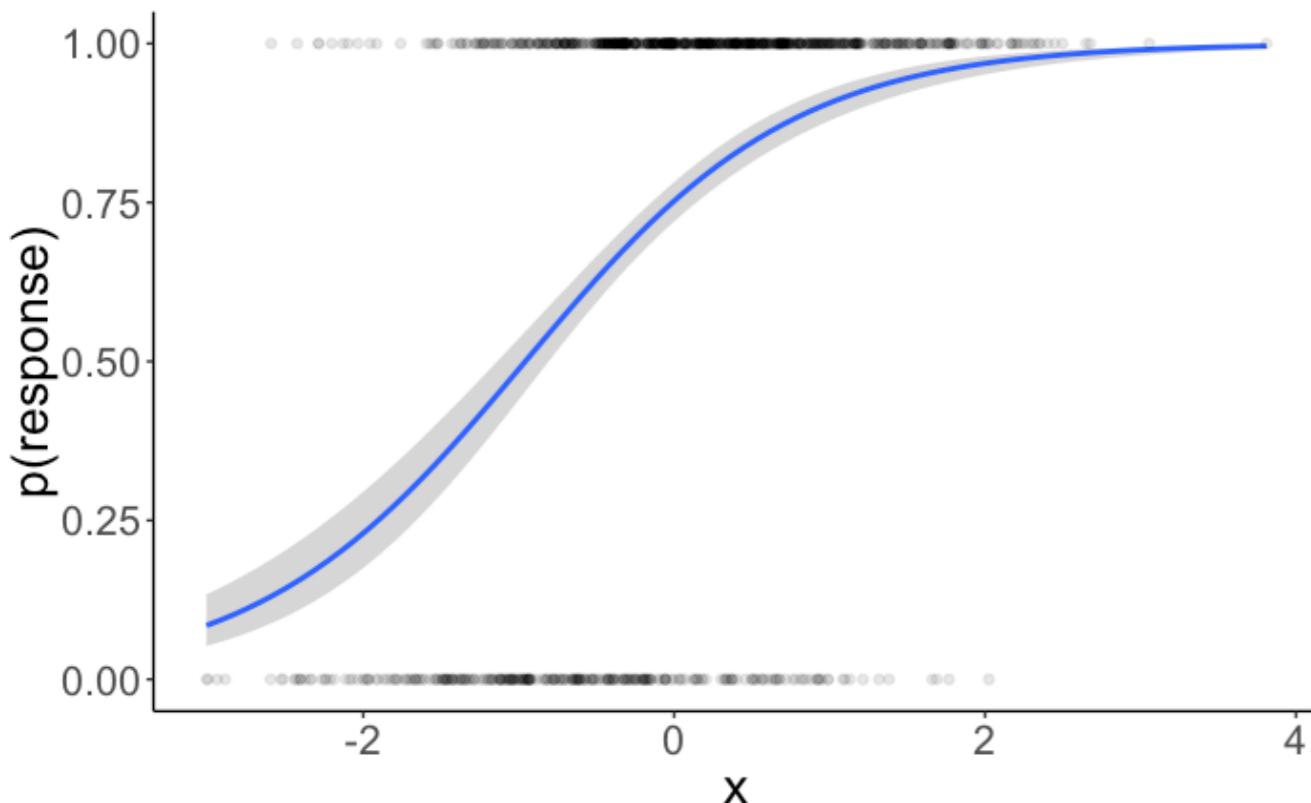
actual value → predicted value →

- 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

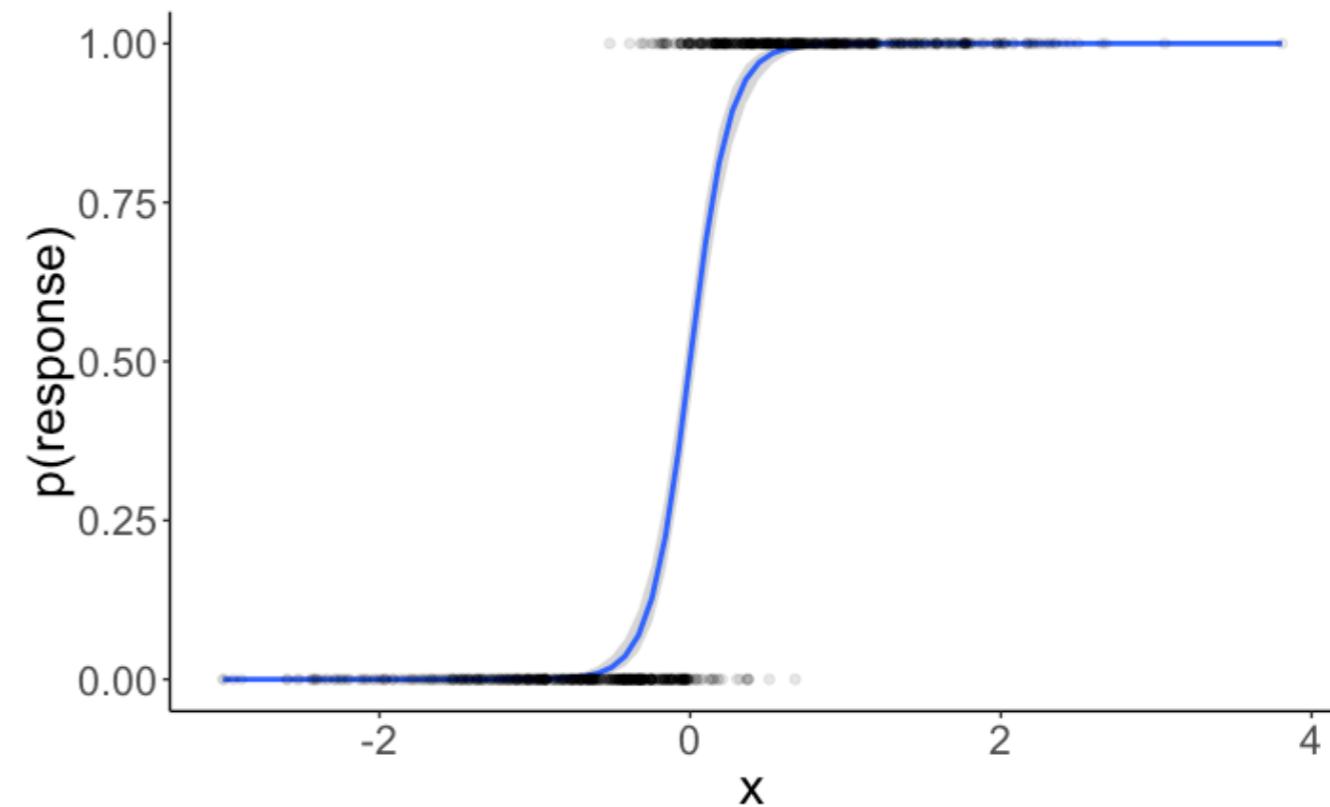
Assessing the model fit

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

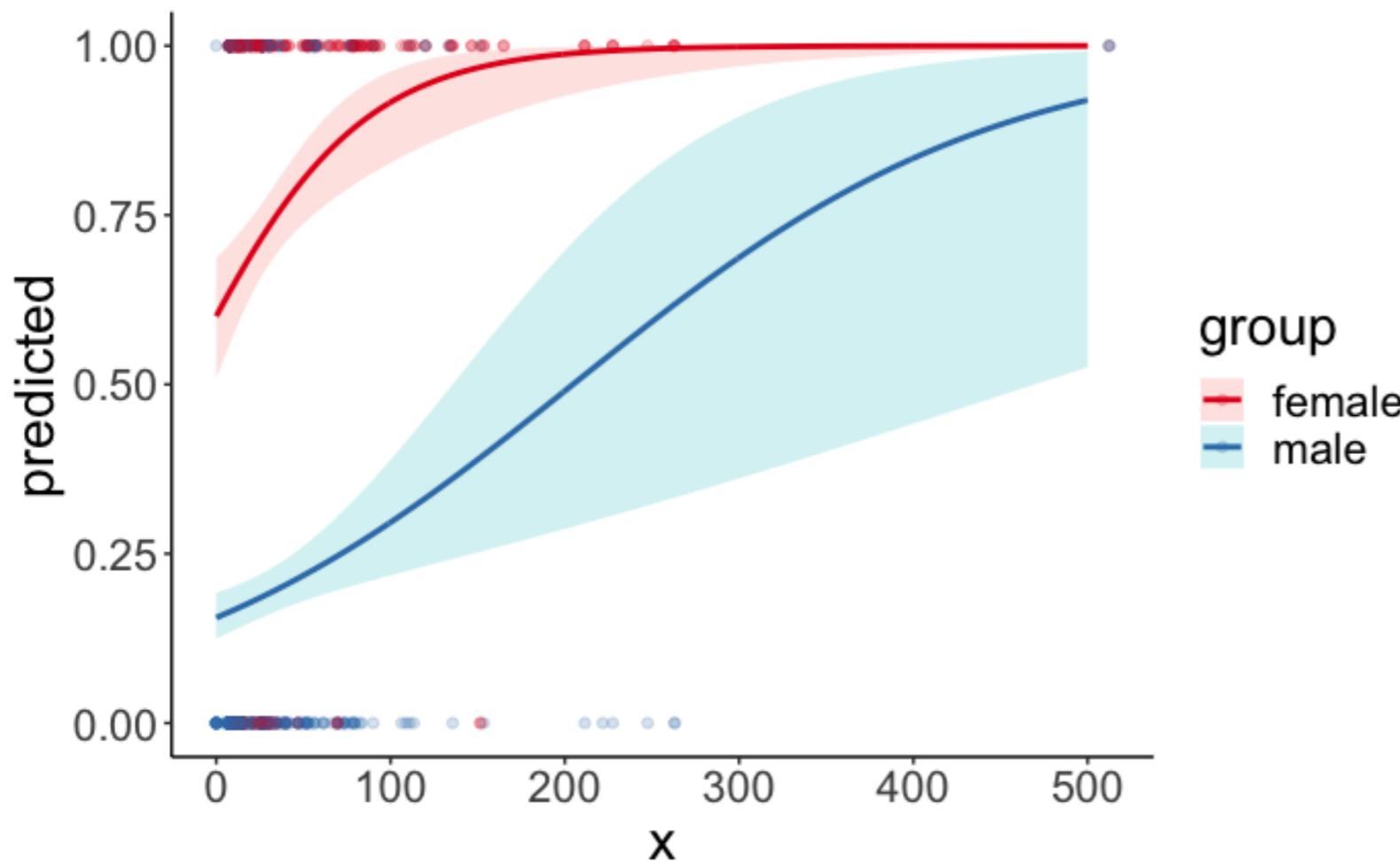
```
Analysis of Deviance Table

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

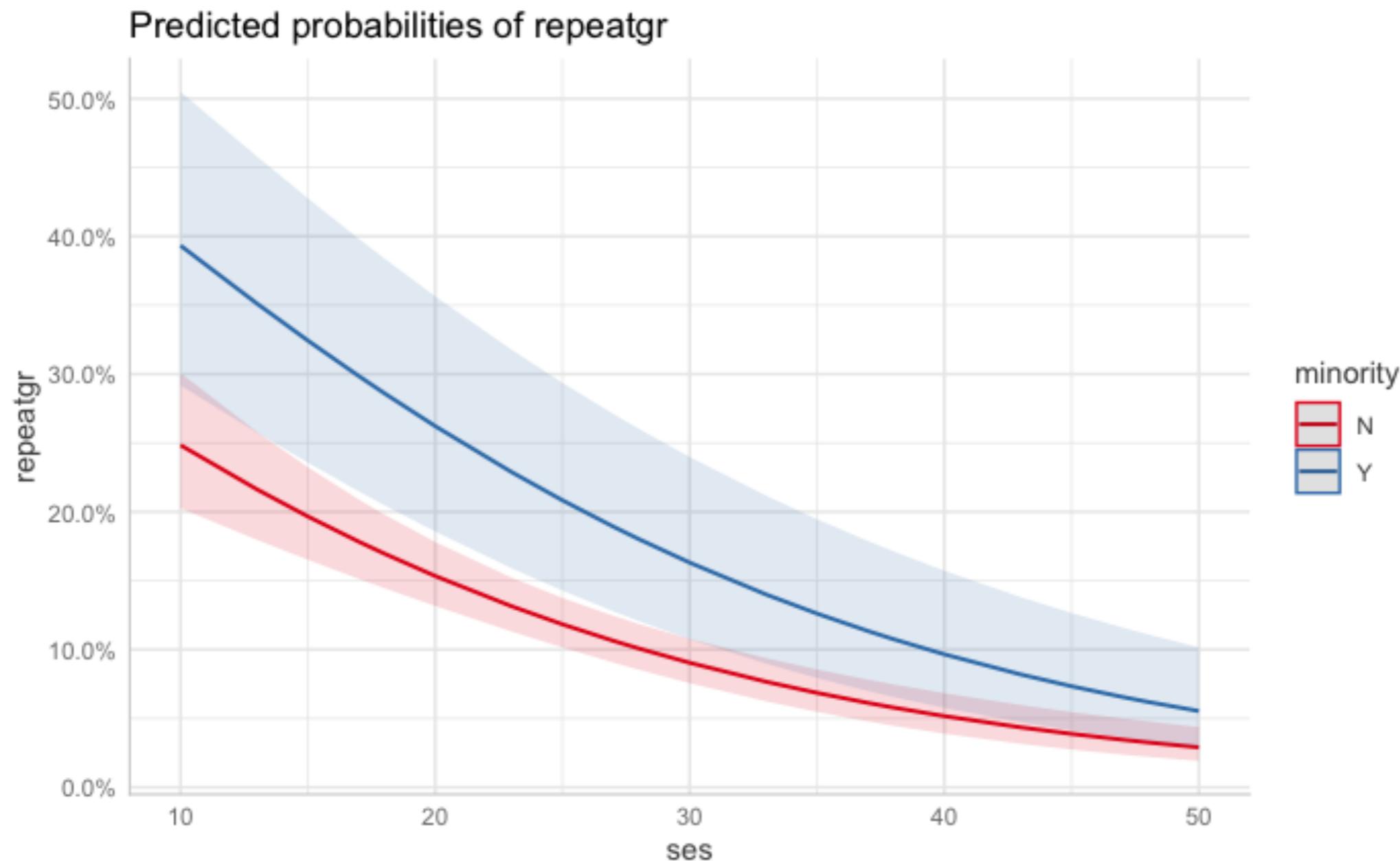
repeated a grade: yes / no

```
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()
```



Hypothesis test: anova (test = "LRT")

```
1 fit_a = glmer(repeatgr ~ 1 + ses + minority + (1 | school_nr),  
2 data = df.language,  
3 family = "binomial")  
4  
5 # dropping ses as a predictor  
6 fit_c = glmer(repeatgr ~ 1 + minority + (1 | school_nr),  
7 data = df.language,  
8 family = "binomial")  
9  
10 anova(fit_a, fit_c, test = "LRT")
```

```
Data: df.language  
Models:  
fit_c: repeatgr ~ 1 + minority + (1 | school_nr)  
fit_a: repeatgr ~ 1 + ses + minority + (1 | school_nr)  
      npar   AIC   BIC  logLik deviance    Chisq Df Pr(>Chisq)  
fit_c     3 1732.5 1749.7 -863.27    1726.5  
fit_a     4 1659.1 1682.1 -825.57    1651.1 75.395  1 < 2.2e-16 ***  
---  
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plan for today

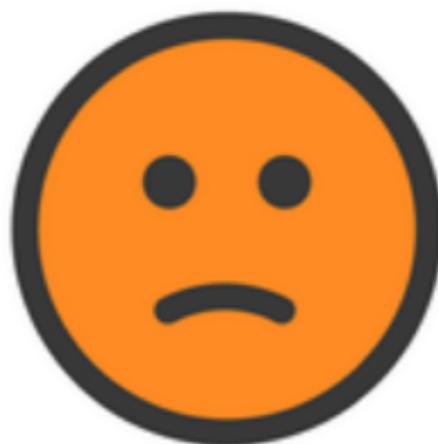
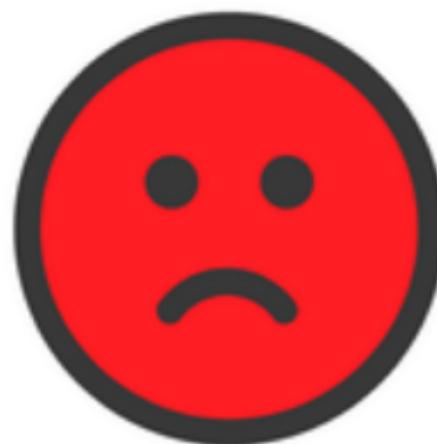
- Quick Recap
- `lmer()` standard operating procedures
- Generalized linear model
 - Logistic regression
 - interpreting the model output
 - fitting and reporting models
 - mixed effects logistic regression

Feedback

How was the pace of today's class?

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

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!