Logistic regression

Logistic regression

- When response variable is measured/counted, regression can work well.
- But what if response is yes/no, lived/died, success/failure?
- Model *probability* of success.
- Probability must be between 0 and 1; need method that ensures this.
- *Logistic regression* does this. In R, is a *generalized linear model* with binomial "family":

```
glm(y ~ x, family="binomial")
```

- Begin with simplest case.

Packages

```
library(MASS)
library(tidyverse)
library(marginaleffects)
library(broom)
library(nnet)
library(conflicted)
conflict_prefer("select", "dplyr")
conflict_prefer("filter", "dplyr")
conflict_prefer("rename", "dplyr")
conflict_prefer("summarize", "dplyr")
```

The rats, part 1

- Rats given dose of some poison; either live or die:

```
dose status
0 lived
1 died
2 lived
3 lived
4 died
5 died
```

Read in:

```
my_url <- "http://ritsokiguess.site/datafiles/rat.txt"
rats <- read_delim(my_url, " ")
rats
```

dose	status
0	lived
1	died
2	lived
3	lived
4	died
5	died

Basic logistic regression

- Make response into a factor first:

```
rats2 <- rats %>% mutate(status = factor(status))
```

- then fit model:

```
status.1 <- glm(status ~ dose, family = "binomial", data = rats2)
```

Output

```
summary(status.1)
```

```
##
## Call:
## glm(formula = status ~ dose, family = "binomial", data = rats2)
##
## Deviance Residuals:
##       1        2        3        4        5        6
##  0.5835  -1.6254   1.0381   1.3234  -0.7880  -0.5835
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6841     1.7979   0.937    0.349
## dose         -0.6736     0.6140  -1.097    0.273
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.3178  on 5  degrees of freedom
## Residual deviance: 6.7728  on 4  degrees of freedom
## AIC: 10.773
##
## Number of Fisher Scoring iterations: 4
```

Interpreting the output

- Like (multiple) regression, get tests of significance of individual x's
- Here not significant (only 6 observations).
- "Slope" for dose is negative, meaning that as dose increases, probability of event modelled (survival) decreases.

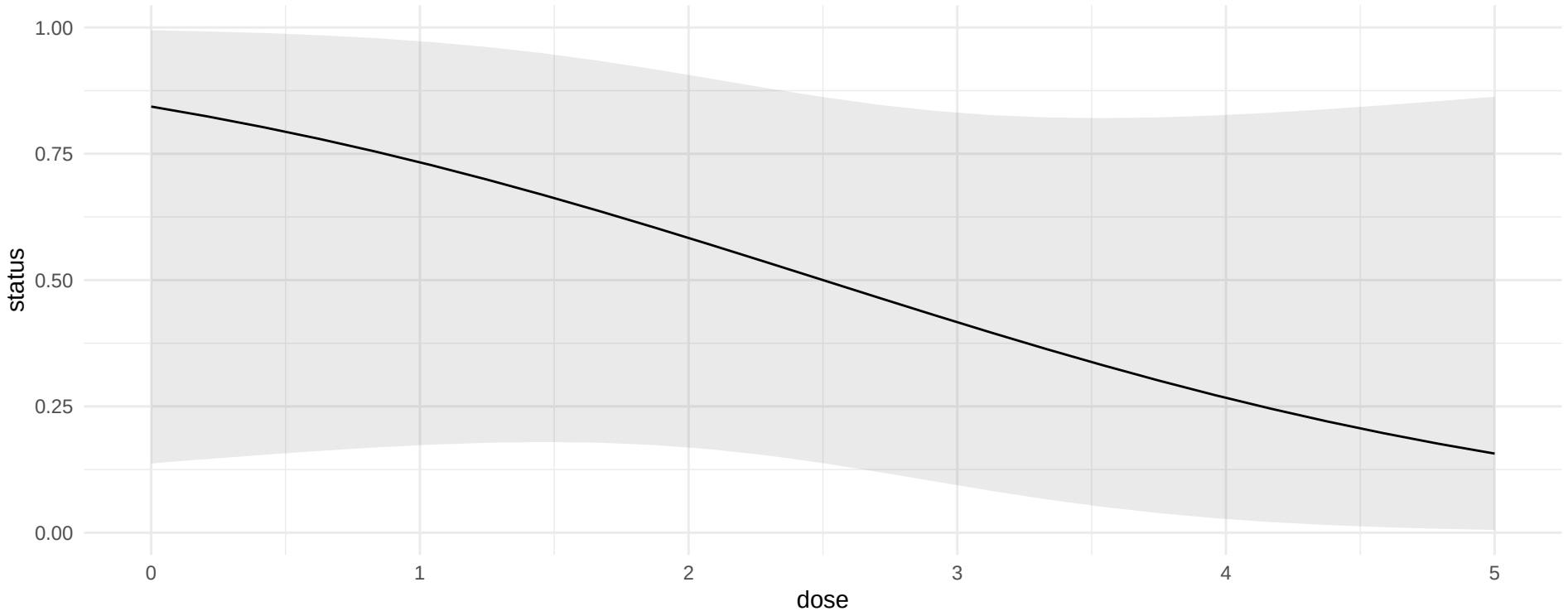
Output part 2: predicted survival probs

```
predictions(status.1)
```

rowid	type	predicted	std.error	statistic	p.value	conf.low	conf.high	status	dose
1	response	0.8434490	0.2373945	3.5529421	0.0003809	0.1370958	0.9945564	lived	0
2	response	0.7331122	0.2569246	2.8534134	0.0043252	0.1731865	0.9729896	died	1
3	response	0.5834187	0.2394051	2.4369518	0.0148117	0.1688476	0.9061463	lived	2
4	response	0.4165813	0.2394051	1.7400686	0.0818470	0.0938537	0.8311524	lived	3
5	response	0.2668878	0.2569246	1.0387786	0.2989077	0.0270104	0.8268135	died	4
6	response	0.1565510	0.2373945	0.6594552	0.5096035	0.0054436	0.8629042	died	5

On a graph

```
plot_cap(status.1, condition = "dose")
```



The rats, more

- More realistic: more rats at each dose (say 10).
- Listing each rat on one line makes a big data file.
- Use format below: dose, number of survivals, number of deaths.

```
dose lived died
0    10    0
1     7    3
2     6    4
3     4    6
4     2    8
5     1    9
```

- 6 lines of data correspond to 60 actual rats.
- Saved in `rat2.txt`.

These data

```
my_url <- "http://ritsokiguess.site/datafiles/rat2.txt"
rat2 <- read_delim(my_url, " ")
```

```
## Rows: 6 Columns: 3
## -- Column specification --------------------------------------
## Delimiter: " "
## dbl (3): dose, lived, died
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
```

```
rat2
```

dose	lived	died
0	10	0
1	7	3
2	6	4
3	4	6
4	2	8
5	1	9

Create response matrix:

- Each row contains *multiple* observations.
- Create *two-column* response:
  - #survivals in first column,
  - #deaths in second.

```
response <- with(rat2, cbind(lived, died))
response
```

```
##       lived died
## [1,]    10    0
## [2,]     7    3
## [3,]     6    4
## [4,]     4    6
## [5,]     2    8
## [6,]     1    9
```

- Response is R `matrix`:

```
class(response)
```

```
## [1] "matrix" "array"
```

Fit logistic regression

- using response you just made:

```
rat2.1 <- glm(response ~ dose,
  family = "binomial",
  data = rat2
)
```

Output

```
summary(rat2.1)
```

```
##
## Call:
## glm(formula = response ~ dose, family = "binomial", data = rat2)
##
## Deviance Residuals:
##       1        2        3        4        5        6
##  1.3421  -0.7916  -0.1034   0.1034   0.0389   0.1529
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3619     0.6719   3.515 0.000439 ***
## dose         -0.9448     0.2351  -4.018 5.87e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.530  on 5  degrees of freedom
## Residual deviance:  2.474  on 4  degrees of freedom
## AIC: 18.94
##
## Number of Fisher Scoring iterations: 4
```

Predicted survival probs

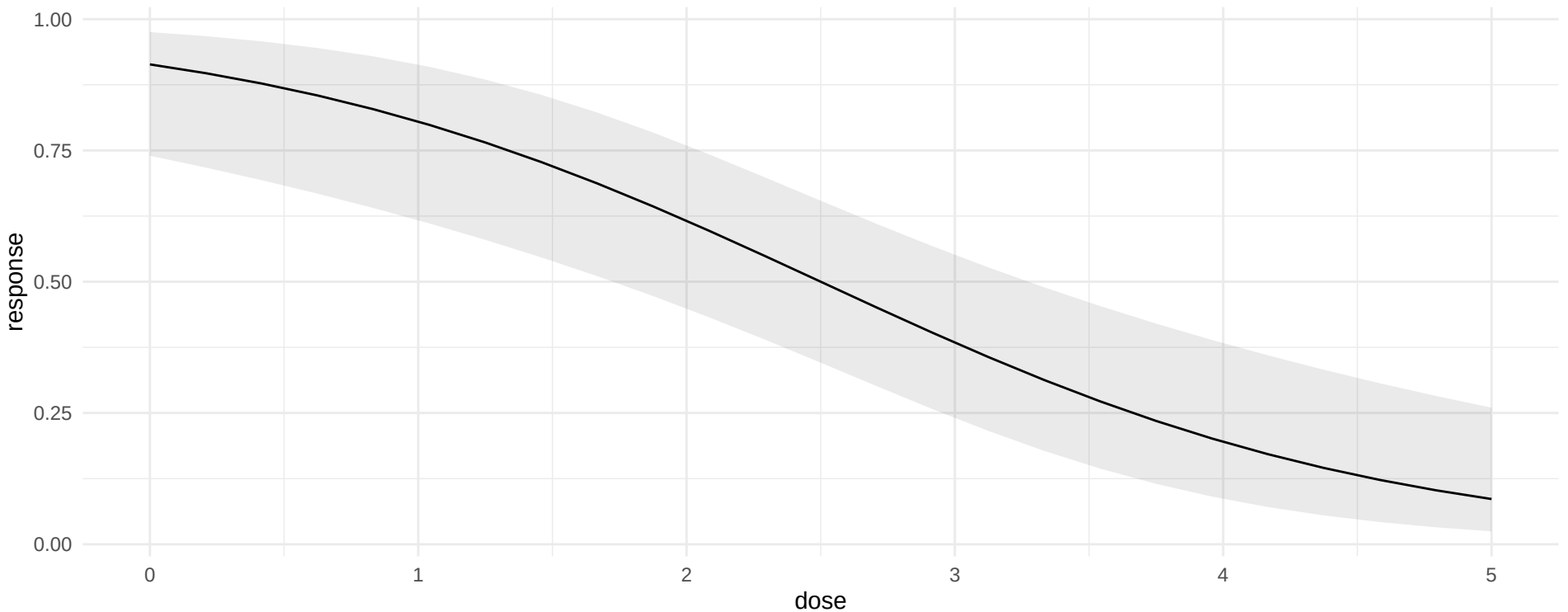
```
# p <- predict(rat2.1, type = "response")
# cbind(rat2, p)
predictions(rat2.1)
```

rowid	type	predicted	std.error	statistic	p.value	conf.low	conf.high	dose
1	response	0.9138762	0.0528798	17.282153	0.0000000	0.7398304	0.9753671	0
2	response	0.8048905	0.0753564	10.681118	0.0000000	0.6169584	0.9135390	1
3	response	0.6159474	0.0818379	7.526434	0.0000000	0.4487610	0.7595916	2
4	response	0.3840526	0.0818379	4.692846	0.0000027	0.2404084	0.5512390	3
5	response	0.1951095	0.0753564	2.589155	0.0096212	0.0864609	0.3830417	4

On a picture

```
plot_cap(rat2.1, condition = "dose")
```

```
## Warning:
## Matrix columns are not supported and are omitted. This may
## of the quantities of interest. You can construct your own p
## supply it explicitly to the `newdata` argument.
```



Comments

- Significant effect of dose.
- Effect of larger dose is to *decrease* survival probability ("slope" negative; also see in decreasing predictions.)
- Confidence intervals around prediction narrower (more data).

Multiple logistic regression

- With more than one x, works much like multiple regression.
- Example: study of patients with blood poisoning severe enough to warrant surgery. Relate survival to other potential risk factors.
- Variables, 1=present, 0=absent:
  - survival (death from sepsis=1), response
  - shock
  - malnutrition
  - alcoholism
  - age (as numerical variable)
  - bowel infarction
- See what relates to death.

Read in data

```
my_url <-
  "http://ritsokiguess.site/datafiles/sepsis.txt"
sepsis <- read_delim(my_url, " ")
```

```
## Rows: 106 Columns: 6
## -- Column specification ------------------------------
## Delimiter: " "
## dbl (6): death, shock, malnut, alcohol, age, bowelinf
##
## i Use `spec()` to retrieve the full column specification fo
## i Specify the column types or set `show_col_types = FALSE`
```

Make sure categoricals really are

```
sepsis %>%
  mutate(across(-age, \(x) factor(x))) -> sepsis
```

The data (some)

```
sepsis
```

death	shock	malnut	alcohol	age	bowelinf
0	0	0	0	56	0
0	0	0	0	80	0
0	0	0	0	61	0
0	0	0	0	26	0
0	0	0	0	53	0
1	0	1	0	87	0
0	0	0	0	21	0
1	0	0	1	69	0
0	0	0	0	57	0
0	0	1	0	76	0
1	0	0	1	66	1
0	0	0	0	48	0
0	0	0	0	18	0

Fit model

```
sepsis.1 <- glm(death ~ shock + malnut + alcohol + age +
  bowelinf,
family = "binomial",
data = sepsis
)
```

Output part 1

```
tidy(sepsis.1)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-9.7539056	2.5416952	-3.837559	0.0001243
shock1	3.6738658	1.1648114	3.154044	0.0016103
malnut1	1.2165811	0.7282236	1.670615	0.0947978
alcohol1	3.3548846	0.9821026	3.416022	0.0006354
age	0.0921527	0.0303237	3.038968	0.0023739
bowelinf1	2.7975864	1.1639717	2.403483	0.0162397

- All P-values fairly small
- but `malnut` not significant: remove.

Removing malnut

```
sepsis.2 <- update(sepsis.1, . ~ . - malnut)
tidy(sepsis.2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-8.8945899	2.3168948	-3.839013	0.0001235
shock1	3.7011932	1.1035347	3.353944	0.0007967
alcohol1	3.1859040	0.9172457	3.473338	0.0005140
age	0.0898318	0.0292153	3.074821	0.0021063
bowelinf1	2.3864685	1.0722662	2.225631	0.0260389

- Everything significant now.

Comments

- Most of the original x's helped predict death. Only `malnut` seemed not to add anything.
- Removed `malnut` and tried again.
- Everything remaining is significant (though `bowelinf` actually became *less* significant).
- All coefficients are *positive*, so having any of the risk factors (or being older) *increases* risk of death.

Another way to see

```
comparisons(sepsis.2) %>% summary()
```

type	term	contrast	estimate	std.error	statistic	p.value	conf.low	conf.high
response	shock	1 - 0	0.3985100	0.0963208	4.137321	0.0000351	0.2097247	0.5872952
response	alcohol	1 - 0	0.3059301	0.0679554	4.501923	0.0000067	0.1727399	0.4391203
response	age	+1	0.0074386	0.0019647	3.786060	0.0001531	0.0035878	0.0112894
response	bowelinf	1 - 0	0.2415562	0.1009452	2.392944	0.0167138	0.0437072	0.4394051

- An additional year of age, all else equal, increases P(death) by 0.007 on average
- Having `shock` (vs. not), all else equal, increases P(death) by 0.399 on average
- The actual size of the effects depends on values of other variables (non-linear model)

Predictions from model without “malnut”

- A few (rows of original dataframe) chosen “at random”:

```
sepsis %>% slice(c(4, 1, 2, 11, 32)) -> new
predictions(sepsis.2, newdata = new)
```

rowid	type	predicted	std.error	statistic	p.value	conf.low	conf.high	death	shock	malnut	alcohol	age	bowelinf
1	response	0.0014153	0.0022482	0.6295479	0.5289904	0.0000627	0.0310305	0	0	0	0	26	0
2	response	0.0205524	0.0167209	1.2291450	0.2190174	0.0041025	0.0965660	0	0	0	0	56	0
3	response	0.1534168	0.0739159	2.0755605	0.0379346	0.0560684	0.3560344	0	0	0	0	80	0
4	response	0.9312901	0.0786701	11.8379155	0.0000000	0.5490986	0.9934148	1	0	0	1	66	1
5	response	0.2130010	0.1013932	2.1007420	0.0356636	0.0763906	0.4696795	1	0	0	1	49	0

Comments

- Survival chances pretty good if no risk factors, though decreasing with age.
- Having more than one risk factor reduces survival chances dramatically.
- Usually good job of predicting survival; sometimes death predicted to survive.

Another way to assess effects 1/2

of age:

```
predictions(sepsis.2, variables = "age")
```

rowid	rowidcf	type	predicted	std.error	statistic	p.value	conf.low	conf.high	death	shock	alcohol	bowelinf	age
1	1	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
2	2	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
3	3	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
4	4	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
5	5	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
6	6	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0
7	7	response	0.000631	0.00115	0.544111	0.586364	0.000017	0.022686	0	0	0	0	17.0

Assessing effects 2/2

Effect of `shock`:

```
predictions(sepsis.2, variables = "shock")
```

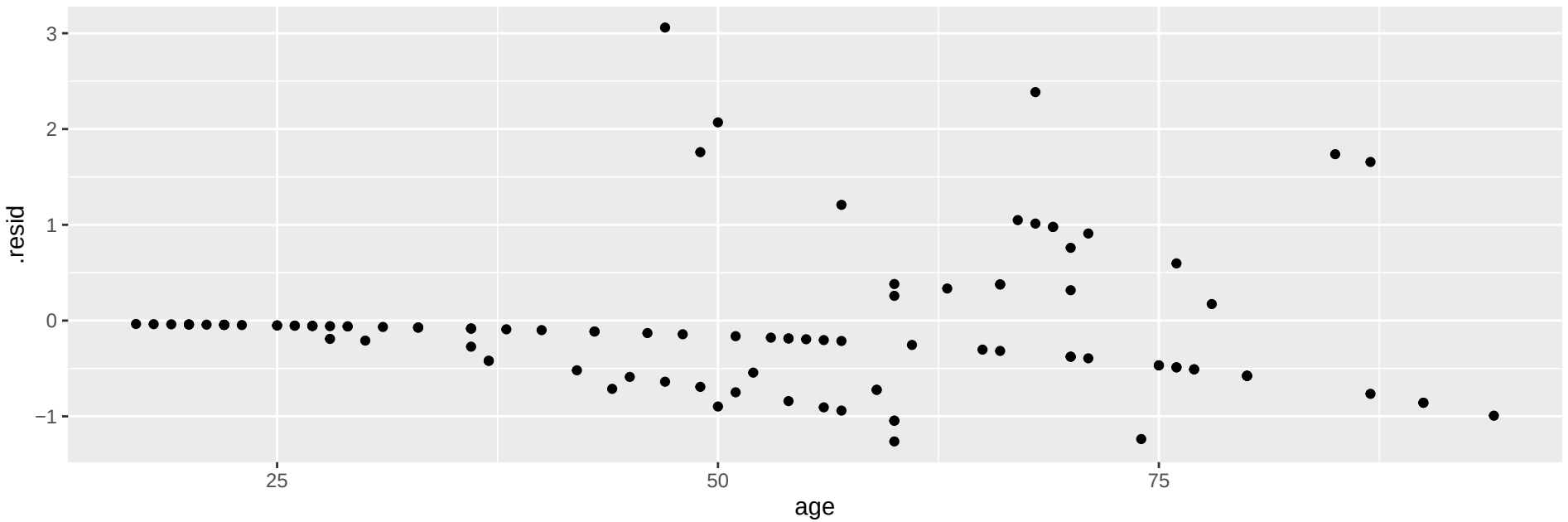
rowid	rowidcf	type	predicted	std.error	statistic	p.value	conf.low	conf.high	death	alcohol	age	bowelinf	shock
1	1	response	0.020552	0.016721	1.229145	0.219017	-0.012	0.053266	0	0	56	0	0
2	2	response	0.153416	0.073912	2.075560	0.037934	0.008	0.298034	0	0	80	0	0
3	3	response	0.031834	0.022491	1.415280	0.156986	-0.013	0.075844	0	0	61	0	0
4	4	response	0.001415	0.002248	0.629547	0.528990	-0.003	0.005831	0	0	26	0	0
5	5	response	0.015773	0.013925	1.132511	0.257419	-0.011	0.043085	0	0	53	0	0
6	6	response	0.253652	0.120072	2.112370	0.034654	0.018	0.489231	0	0	87	0	0
7	7	response	0.000903	0.001560	0.579187	0.562462	-0.002	0.003968	0	0	21	0	0

Assessing proportionality of odds for age

- An assumption we made is that log-odds of survival depends linearly on age.
- Hard to get your head around, but basic idea is that survival chances go continuously up (or down) with age, instead of (for example) going up and then down.
- In this case, seems reasonable, but should check:

Residuals vs. age

```
sepsis.2 %>% augment(sepsis) %>%
  ggplot(aes(x = age, y = .resid)) +
  geom_point()
```



Comments

- No apparent problems overall.
- Confusing "line" across: no risk factors, survived.

Probability and odds

- For probability p, odds is $p/(1-p)$:

Prob.	Odds	log-odds	in words
0.5	$0.5/0.5 = 1/1 = 1.00$	0.00	"even money"
0.1	$0.1/0.9 = 1/9 = 0.11$	-2.20	"9 to 1"
0.4	$0.4/0.6 = 1/1.5 = 0.67$	-0.41	"1.5 to 1"
0.8	$0.8/0.2 = 4/1 = 4.00$	1.39	"4 to 1 on"

- Gamblers use odds: if you win at 9 to 1 odds, get original stake back plus 9 times the stake.
- Probability has to be between 0 and 1
- Odds between 0 and infinity
- *Log*-odds can be anything: any log-odds corresponds to valid probability.

Odds ratio

- Suppose 90 of 100 men drank wine last week, but only 20 of 100 women.
- Prob of man drinking wine $90/100 = 0.9$, woman $20/100 = 0.2$.
- Odds of man drinking wine $0.9/0.1 = 9$, woman $0.2/0.8 = 0.25$.
- Ratio of odds is $9/0.25 = 36$.
- Way of quantifying difference between men and women: "odds of drinking wine 36 times larger for males than females' '.

Sepsis data again

- Recall prediction of probability of death from risk factors:

```
sepsis.2.tidy <- tidy(sepsis.2)
sepsis.2.tidy
```

term	estimate	std.error	statistic	p.value
(Intercept)	-8.8945899	2.3168948	-3.839013	0.0001235
shock1	3.7011932	1.1035347	3.353944	0.0007967
alcohol1	3.1859040	0.9172457	3.473338	0.0005140
age	0.0898318	0.0292153	3.074821	0.0021063
bowelinf1	2.3864685	1.0722662	2.225631	0.0260389

- Slopes in column `estimate`.

Multiplying the odds

- Can interpret slopes by taking "exp" of them. We ignore intercept.

```
sepsis.2.tidy %>%
  mutate(exp_coeff=exp(estimate)) %>%
  select(term, exp_coeff)
```

term	exp_coeff
(Intercept)	0.0001371
shock1	40.4955951
alcohol1	24.1891449
age	1.0939902
bowelinf1	10.8750206

Interpretation

term	exp_coeff
(Intercept)	0.0001371
shock1	40.4955951
alcohol1	24.1891449
age	1.0939902
bowelinf1	10.8750206

- These say "how much do you *multiply* odds of death by for increase of 1 in corresponding risk factor?" Or, what is odds ratio for that factor being 1 (present) vs. 0 (absent)?
- Eg. being alcoholic vs. not increases odds of death by 24 times
- One year older multiplies odds by about 1.1 times. Over 40 years, about $1.09^{40} = 31$ times.

Odds ratio and relative risk

- **Relative risk** is ratio of probabilities.
- Above: 90 of 100 men (0.9) drank wine, 20 of 100 women (0.2).
- Relative risk 0.9/0.2=4.5. (odds ratio was 36).
- When probabilities small, relative risk and odds ratio similar.
- Eg. prob of man having disease 0.02, woman 0.01.
- Relative risk $0.02/0.01 = 2$.

Odds ratio vs. relative risk

- Odds for men and for women:

```
(od1 <- 0.02 / 0.98) # men
```

```
## [1] 0.02040816
```

```
(od2 <- 0.01 / 0.99) # women
```

```
## [1] 0.01010101
```

- Odds ratio

```
od1 / od2
```

```
## [1] 2.020408
```

- Very close to relative risk of 2.

More than 2 response categories

- With 2 response categories, model the probability of one, and prob of other is one minus that. So doesn't matter which category you model.
- With more than 2 categories, have to think more carefully about the categories: are they
- *ordered*: you can put them in a natural order (like low, medium, high)
- *nominal*: ordering the categories doesn't make sense (like red, green, blue).
- R handles both kinds of response; learn how.

Ordinal response: the miners

- Model probability of being in given category *or lower*.
- Example: coal-miners often suffer disease pneumoconiosis. Likelihood of disease believed to be greater among miners who have worked longer.
- Severity of disease measured on categorical scale: none, moderate, severe.

Miners data

- Data are frequencies:

Exposure	None	Moderate	Severe
5.8	98	0	0
15.0	51	2	1
21.5	34	6	3
27.5	35	5	8
33.5	32	10	9
39.5	23	7	8
46.0	12	6	10
51.5	4	2	5

Reading the data

Data in aligned columns with more than one space between, so:

```
my_url <- "http://ritsokiguess.site/datafiles/miners-tab.txt"
freqs <- read_table(my_url)
```

```
##
## -- Column specification --------------------------------------
## cols(
##   Exposure = col_double(),
##   None = col_double(),
##   Moderate = col_double(),
##   Severe = col_double()
## )
```

The data

```
freqs
```

Exposure	None	Moderate	Severe
5.8	98	0	0
15.0	51	2	1
21.5	34	6	3
27.5	35	5	8
33.5	32	10	9
39.5	23	7	8
46.0	12	6	10
51.5	4	2	5

Tidying

```
freqs %>%
  pivot_longer(-Exposure, names_to = "Severity", values_to = "
  mutate(Severity = fct_inorder(Severity)) -> miners
```

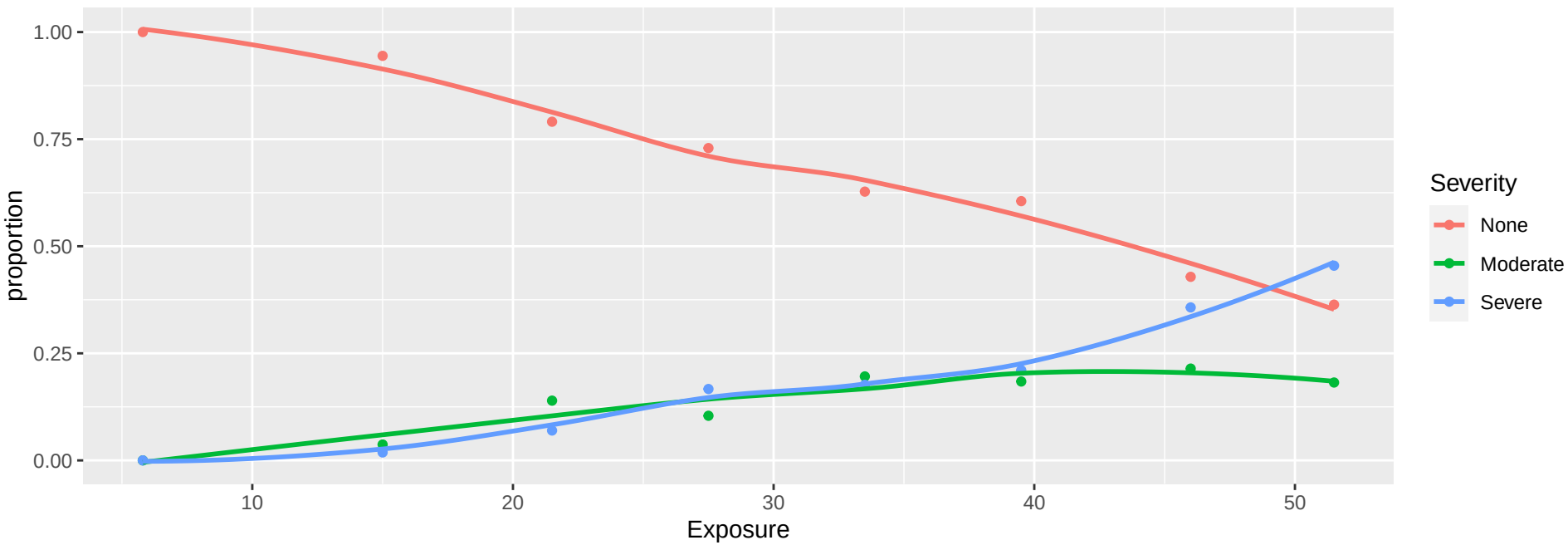
Result

```
miners
```

Exposure	Severity	Freq
5.8	None	98
5.8	Moderate	0
5.8	Severe	0
15.0	None	51
15.0	Moderate	2
15.0	Severe	1
21.5	None	34
21.5	Moderate	6
21.5	Severe	3
27.5	None	35
27.5	Moderate	5
27.5	Severe	8
33.5	None	32
33.5	Moderate	10
33.5	Severe	9
39.5	None	23
39.5	Moderate	7
39.5	Severe	8
46.0	None	12
46.0	Moderate	6
46.0	Severe	10
51.5	None	4
51.5	Moderate	2
51.5	Severe	5

Plot proportions against exposure

```
miners %>%
  group_by(Exposure) %>%
  mutate(proportion = Freq / sum(Freq)) -> prop
  ggplot(prop, aes(x = Exposure, y = proportion,
                   colour = Severity)) +
  geom_point() + geom_smooth(se = F)
```



Reminder of data setup

```
miners
```

Exposure	Severity	Freq
5.8	None	98
5.8	Moderate	0
5.8	Severe	0
15.0	None	51
15.0	Moderate	2
15.0	Severe	1
21.5	None	34
21.5	Moderate	6
21.5	Severe	3
27.5	None	35
27.5	Moderate	5
27.5	Severe	8
33.5	None	32
33.5	Moderate	10
33.5	Severe	9
39.5	None	23
39.5	Moderate	7

Fitting ordered logistic model

Use function `polr` from package `MASS`. Like `glm`.

```
sev.1 <- polr(Severity ~ Exposure,
  weights = Freq,
  data = miners
)
```

Output: not very illuminating

```
sev.1 <- polr(Severity ~ Exposure,
  weights = Freq,
  data = miners,
  Hess = TRUE
)
```

```
summary(sev.1)
```

```
## Call:
## polr(formula = Severity ~ Exposure, data = miners, weights = Freq,
##     Hess = TRUE)
##
## Coefficients:
##           Value Std. Error t value
## Exposure 0.0959    0.01194   8.034
##
## Intercepts:
##                 Value   Std. Error t value
## None|Moderate    3.9558  0.4097     9.6558
## Moderate|Severe  4.8690  0.4411    11.0383
##
## Residual Deviance: 416.9188
## AIC: 422.9188
```

Does exposure have an effect?

Fit model without `Exposure`, and compare using `anova`. Note `1` for model with just intercept:

```
sev.0 <- polr(Severity ~ 1, weights = Freq, data = miners)
anova(sev.0, sev.1)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
1	369	505.1621		NA	NA	NA
Exposure	368	416.9188	1 vs 2	1	88.24324	0

Exposure definitely has effect on severity of disease.

Another way

- What (if anything) can we drop from model with `exposure`?

```
drop1(sev.1, test = "Chisq")
```

	Df	AIC	LRT	Pr(>Chi)
	NA	422.9188	NA	NA
Exposure	1	509.1621	88.24324	0

- Nothing. Exposure definitely has effect.

Predicted probabilities

```
freqs %>% select(Exposure) -> new
predictions(sev.1, newdata = new, type = "probs") %>%
  select(group, predicted, Exposure) %>%
  pivot_wider(names_from = group, values_from = predicted)
```

Exposure	None	Moderate	Severe
5.8	0.9676920	0.0190891	0.0132189
15.0	0.9253445	0.0432993	0.0313561
21.5	0.8692003	0.0738586	0.0569411
27.5	0.7889290	0.1141300	0.0969409
33.5	0.6776641	0.1620715	0.1602644
39.5	0.5418105	0.2048420	0.2533476
46.0	0.3879962	0.2244155	0.3875883
51.5	0.2722543	0.2102501	0.5174956

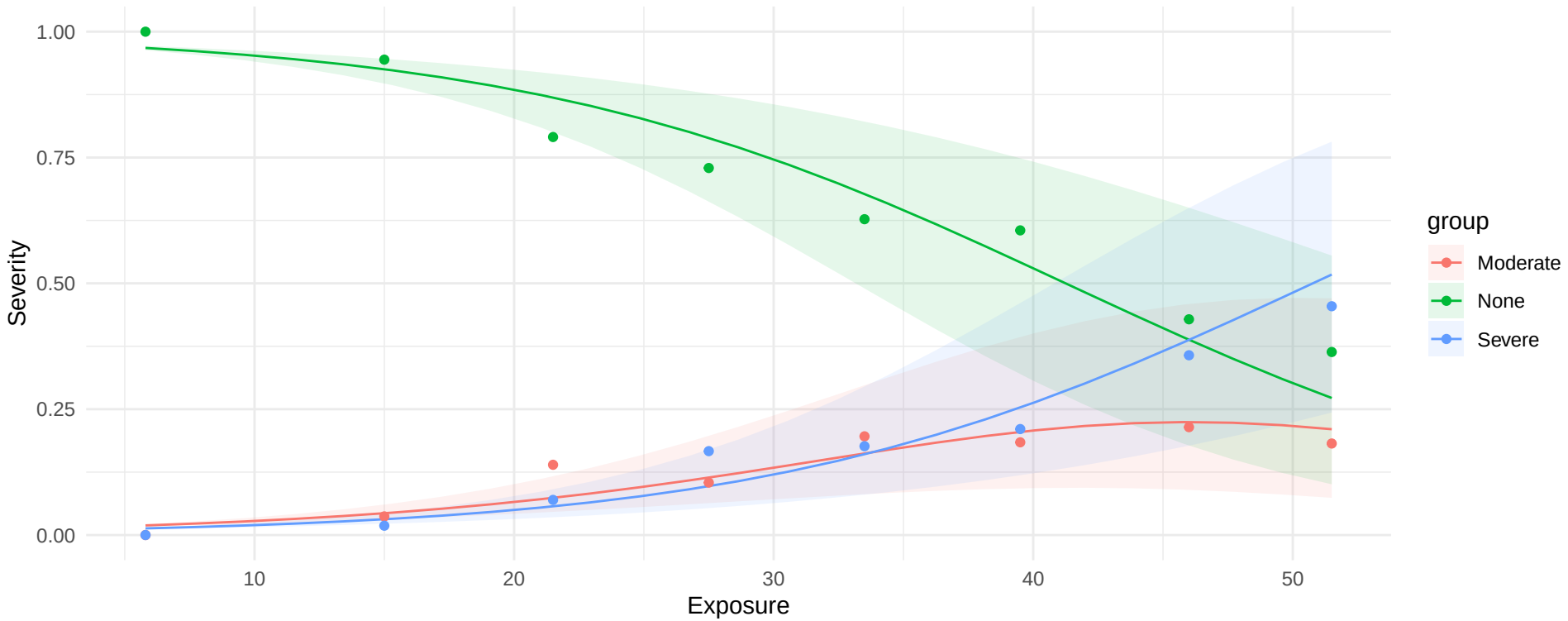
Plot of predicted probabilities

```
summary(sev.1)
```

```
## Call:
## polr(formula = Severity ~ Exposure, data = miners, weights
##     Hess = TRUE)
##
## Coefficients:
##           Value Std. Error t value
## Exposure 0.0959    0.01194   8.034
##
## Intercepts:
##                  Value   Std. Error t value
## None|Moderate     3.9558  0.4097      9.6558
## Moderate|Severe   4.8690  0.4411     11.0383
##
## Residual Deviance: 416.9188
## AIC: 422.9188
```

The graph

ggg



Comments

- Model appears to match data well enough.
- As exposure goes up, prob of None goes down, Severe goes up (sharply for high exposure).
- So more exposure means worse disease.

Unordered responses

- With unordered (nominal) responses, can use *generalized logit*.
- Example: 735 people, record age and sex (male 0, female 1), which of 3 brands of some product preferred.
- Data in `mlogit.csv` separated by commas (so `read_csv` will work):

```
my_url <- "http://ritsokiguess.site/datafiles/mlogit.csv"
brandpref <- read_csv(my_url)
```

```
## Rows: 735 Columns: 3
## -- Column specification --------------------------------
## Delimiter: ","
## dbl (3): brand, sex, age
##
## i Use `spec()` to retrieve the full column specification fo
## i Specify the column types or set `show_col_types = FALSE`
```

The data (some)

`brandpref`

brand	sex	age
1	0	24
1	0	26
1	0	26
1	1	27
1	1	27
3	1	27
1	0	27
1	0	27
1	1	27
1	0	27
1	0	27
1	1	27
2	1	28

Bashing into shape, and fitting model

- sex and brand not meaningful as numbers, so turn into factors:

```
brandpref <- brandpref %>%
  mutate(sex = ifelse(sex == 1, "female", "male"),
         sex = factor(sex),
         brand = factor(brand)
         )
```

- We use multinom from package nnet. Works like polr.

```
brands.1 <- multinom(brand ~ age + sex, data = brandpref)
```

```
## # weights:  12 (6 variable)
## initial  value 807.480032
## iter  10 value 702.990572
## final  value 702.970704
## converged
```

Can we drop anything?

- Unfortunately `drop1` seems not to work:

```
drop1(brands.1, test = "Chisq", trace = 0)
```

```
## trying - age
```

```
## Error in if (trace) {: argument is not interpretable as log
```

- So, fall back on fitting model without what you want to test, and comparing using `anova`.

Do age/sex help predict brand? 1/3

Fit models without each of age and sex:

```
brands.2 <- multinom(brand ~ age, data = brandpref)
```

```
## # weights:  9 (4 variable)
## initial  value 807.480032
## iter  10 value 706.796323
## iter  10 value 706.796322
## final  value 706.796322
## converged
```

```
brands.3 <- multinom(brand ~ sex, data = brandpref)
```

```
## # weights:  9 (4 variable)
## initial  value 807.480032
## final  value 791.861266
## converged
```

Do age/sex help predict brand? 2/3

```
anova(brands.2, brands.1)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
age	1466	1413.593		NA	NA	NA
age + sex	1464	1405.941	1 vs 2	2	7.651236	0.021805

```
anova(brands.3, brands.1)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
sex	1466	1583.723		NA	NA	NA
age + sex	1464	1405.941	1 vs 2	2	177.7811	0

Do age/sex help predict brand? 3/3

- age definitely significant (second `anova`)
- `sex` significant also (first `anova`), though P-value less dramatic
- Keep both.
- Expect to see a large effect of age, and a smaller one of `sex`.

Another way to build model

- Start from model with everything and run `step`:

```
step(brands.1, trace = 0)
```

```
## trying - age
## trying - sex
```

```
## Call:
## multinom(formula = brand ~ age + sex)
##
## Coefficients:
##   (Intercept)       age    sexmale
## 2   -11.25127 0.3682202 -0.5237736
## 3   -22.25571 0.6859149 -0.4658215
##
## Residual Deviance: 1405.941
## AIC: 1417.941
```

- Final model contains both age and sex so neither could be removed.

Making predictions

```
predictions(brands.1, variables = c("age", "sex"),
            type = "probs") %>%
  select(group, predicted, age, sex) %>%
  group_by(group, age, sex) %>%
  summarize(pred = mean(predicted)) %>%
  pivot_wider(names_from = group, values_from = pred)
```

```
## `summarise()` has grouped output by 'group', 'age'. You can
## override using the `.groups` argument.
```

age	sex	1	2	3
24	female	0.9153281	0.0818834	0.0027886
24	male	0.9479605	0.0502270	0.0018126
32	female	0.2908654	0.4950385	0.2140961
32	male	0.4048579	0.4081103	0.1870318
34	female	0.1341114	0.4766996	0.3891890

Comments

- Young males prefer brand 1, but older males prefer brand 3.
- Females similar, but like brand 1 less and brand 2 more.
- A clear `brand` effect, but the `sex` effect is less clear.

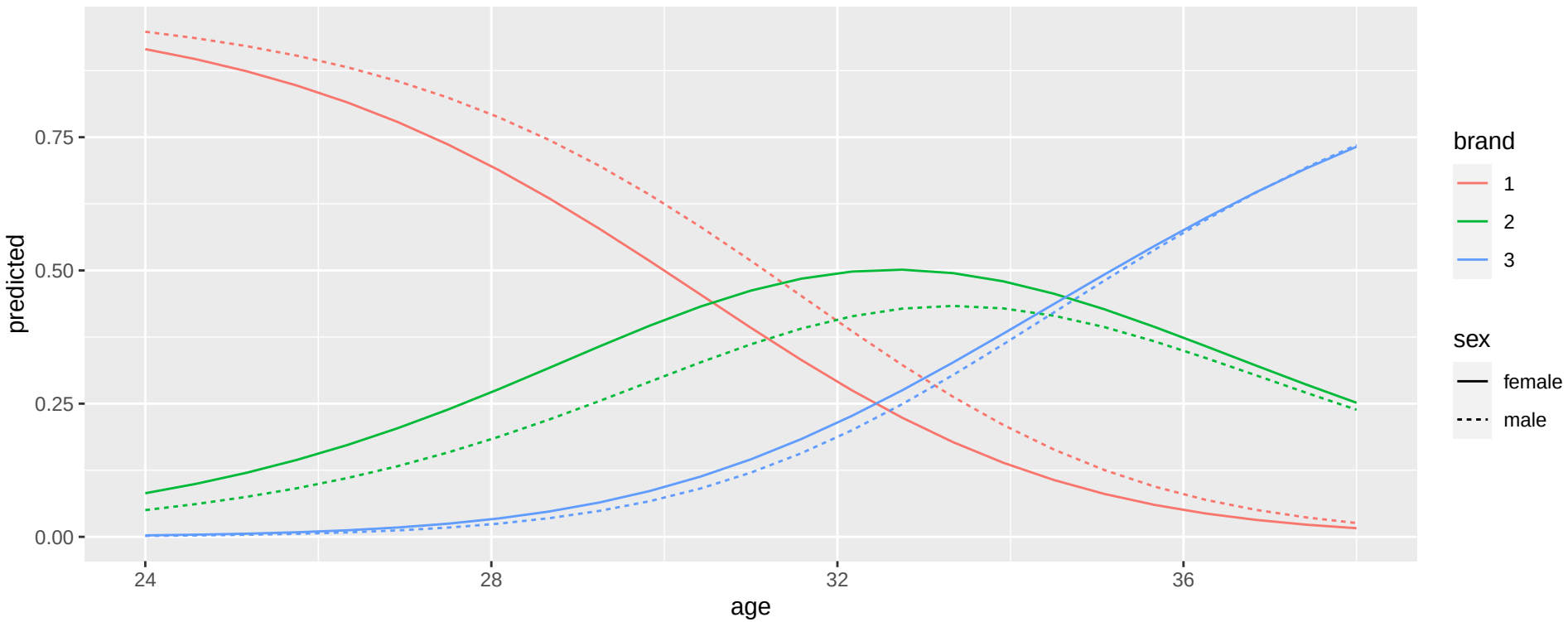
Making a plot

- `plot_cap` doesn't quite work
- so don't draw, edit, *then* make graph:

```
plot_cap(brands.1, condition = c("age", "brand", "sex"),
         type = "probs", draw = FALSE)  %>%
  rename(age = condition1, sex = condition3,
         brand = group) %>%
  ggplot(aes(x = age, y = predicted, colour = brand,
             linetype = sex)) +
  geom_line() -> g
```

The graph

```
gg
```


0.75
0.50
0.25
0.00
predicted
24
28
32
36
age
brand
1
2
3
sex
female
male

Digesting the plot

- Brand vs. age: younger people (of both genders) prefer brand 1, but older people (of both genders) prefer brand 3. (Explains significant age effect.)
- Brand vs. sex: females (solid) like brand 1 less than males (dashed), like brand 2 more (for all ages).
- Not much brand difference between genders (solid and dashed lines of same colours close), but enough to be significant.
- Model didn't include interaction, so modelled effect of gender on brand same for each age, modelled effect of age same for each gender. (See also later.)

Alternative data format

Summarize all people of same brand preference, same sex, same age on one line of data file with frequency on end:

```
1 0 24 1
1 0 26 2
1 0 27 4
1 0 28 4
1 0 29 7
1 0 30 3
...
```

Whole data set in 65 lines not 735! But how?

Getting alternative data format

```
brandpref %>%
  group_by(age, sex, brand) %>%
  summarize(Freq = n()) %>%
  ungroup() -> b
b %>% slice(1:6)
```

age	sex	brand	Freq
24	male	1	1
26	male	1	2
27	female	1	4
27	female	3	1
27	male	1	4
28	female	1	6

Fitting models, almost the same

- Just have to remember `weights` to incorporate frequencies.
- Otherwise `multinom` assumes you have just 1 obs on each line!
- Again turn (numerical) `sex` and `brand` into factors:

```
b %>%
  mutate(sex = factor(sex)) %>%
  mutate(brand = factor(brand)) -> bf
b.1 <- multinom(brand ~ age + sex, data = bf, weights = Freq)
b.2 <- multinom(brand ~ age, data = bf, weights = Freq)
```

P-value for sex identical

```
anova(b.2, b.1)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
age	126	1413.593		NA	NA	NA
age + sex	124	1405.941	1 vs 2	2	7.651236	0.021805

Same P-value as before, so we haven't changed anything important.

Including data on plot

- Everyone's age given as whole number, so maybe not too many different ages with sensible amount of data at each:

```
b %>%
  group_by(age) %>%
  summarize(total = sum(Freq))
```

age	total
24	1
26	2
27	9
28	15
29	19
30	23
31	40
32	333
33	55
34	64
35	35
36	85
37	22
38	32

Comments and next

- Not great (especially at low end), but live with it.
- Need proportions of frequencies in each brand for each age-gender combination. Mimic what we did for miners:

```
b %>%
  group_by(age, sex) %>%
  mutate(proportion = Freq / sum(Freq)) -> brands
```

Checking proportions for age 32

```
brands %>% filter(age == 32)
```

age	sex	brand	Freq	proportion
32	female	1	62	0.2883721
32	female	2	117	0.5441860
32	female	3	36	0.1674419
32	male	1	48	0.4067797
32	male	2	51	0.4322034
32	male	3	19	0.1610169

- First three proportions (females) add up to 1.
- Last three proportions (males) add up to 1.
- So looks like proportions of right thing.

Attempting plot

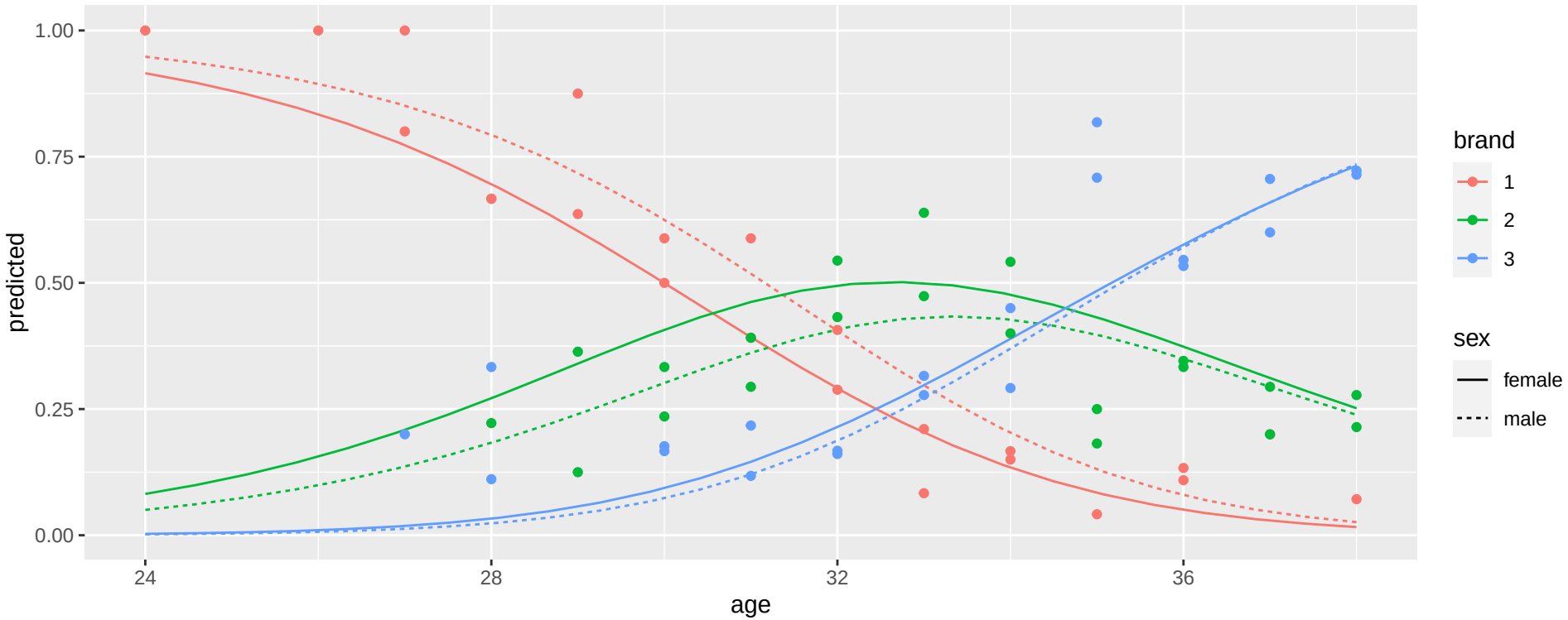
- Take code from previous plot and add `geom_point` with correct `data=` and `aes` to plot data.

```
g + geom_point(data = brands, aes(y = proportion)) -> g1
```

- Data seem to correspond more or less to fitted curves:

The plot

```
g1
```


1.00
0.75
0.50
0.25
0.00
predicted
24
28
32
36
age
brand
1
2
3
sex
female
male

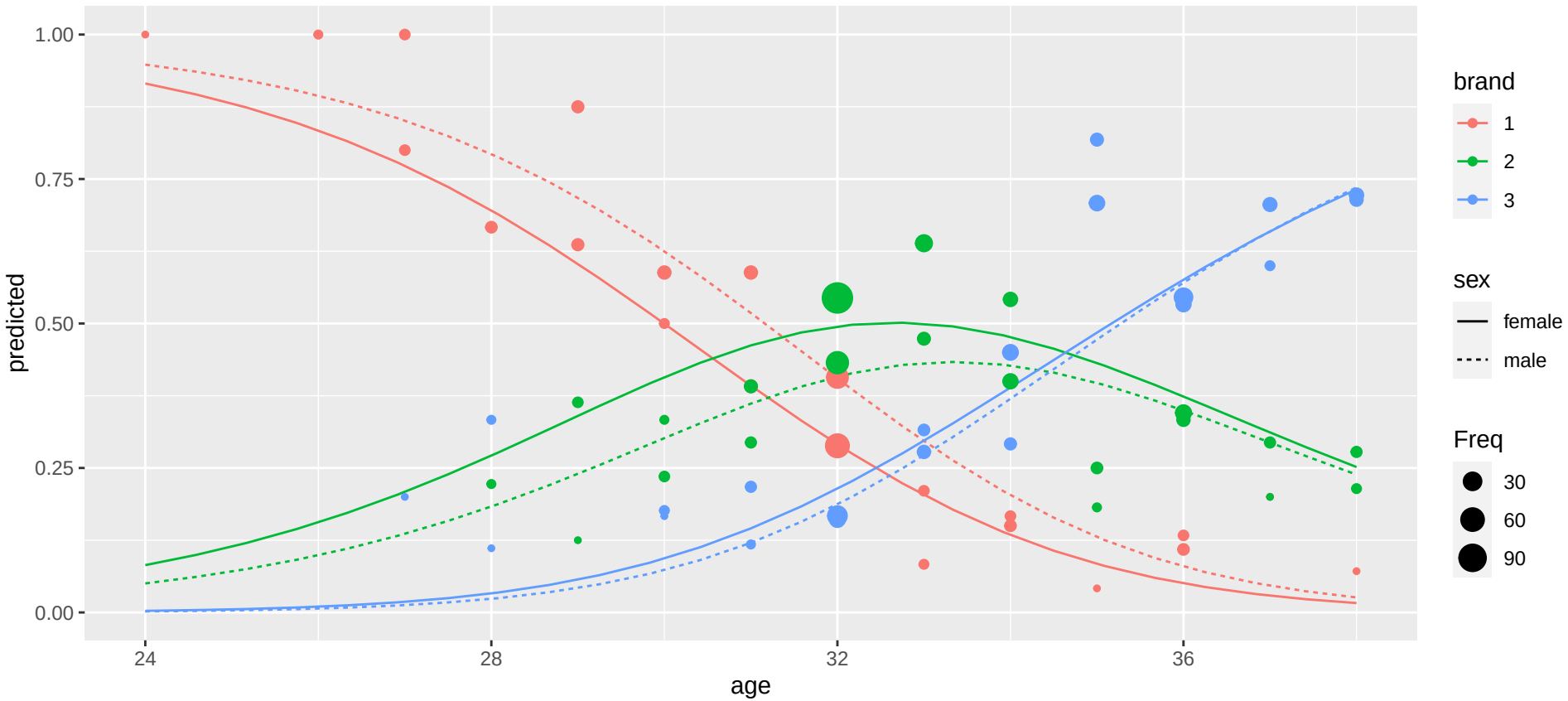
But...

- Some of the plotted points based on a lot of people, and some only a few.
- Idea: make the *size* of plotted point bigger if point based on a lot of people (in `Freq`).
- Hope that larger points then closer to predictions.
- Code:

```
g + geom_point(
    data = brands,
    aes(y = proportion, size = Freq)
  ) -> g2
```

The plot

g2


1.00
0.75
0.50
0.25
0.00
predicted
24
28
32
36
age
brand
1
2
3
sex
female
male
Freq
30
60
90

Trying interaction between age and gender

```
brands.4 <- update(brands.1, . ~ . + age:sex)
```

```
## # weights:  15 (8 variable)
## initial  value 807.480032
## iter  10 value 703.191146
## iter  20 value 702.572260
## iter  30 value 702.570900
## iter  30 value 702.570893
## iter  30 value 702.570893
## final  value 702.570893
## converged
```

```
anova(brands.1, brands.4)
```

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
age + sex	1464	1405.941		NA	NA	NA
age + sex + age:sex	1462	1405.142	1 vs 2	2	0.7996223	0.6704466

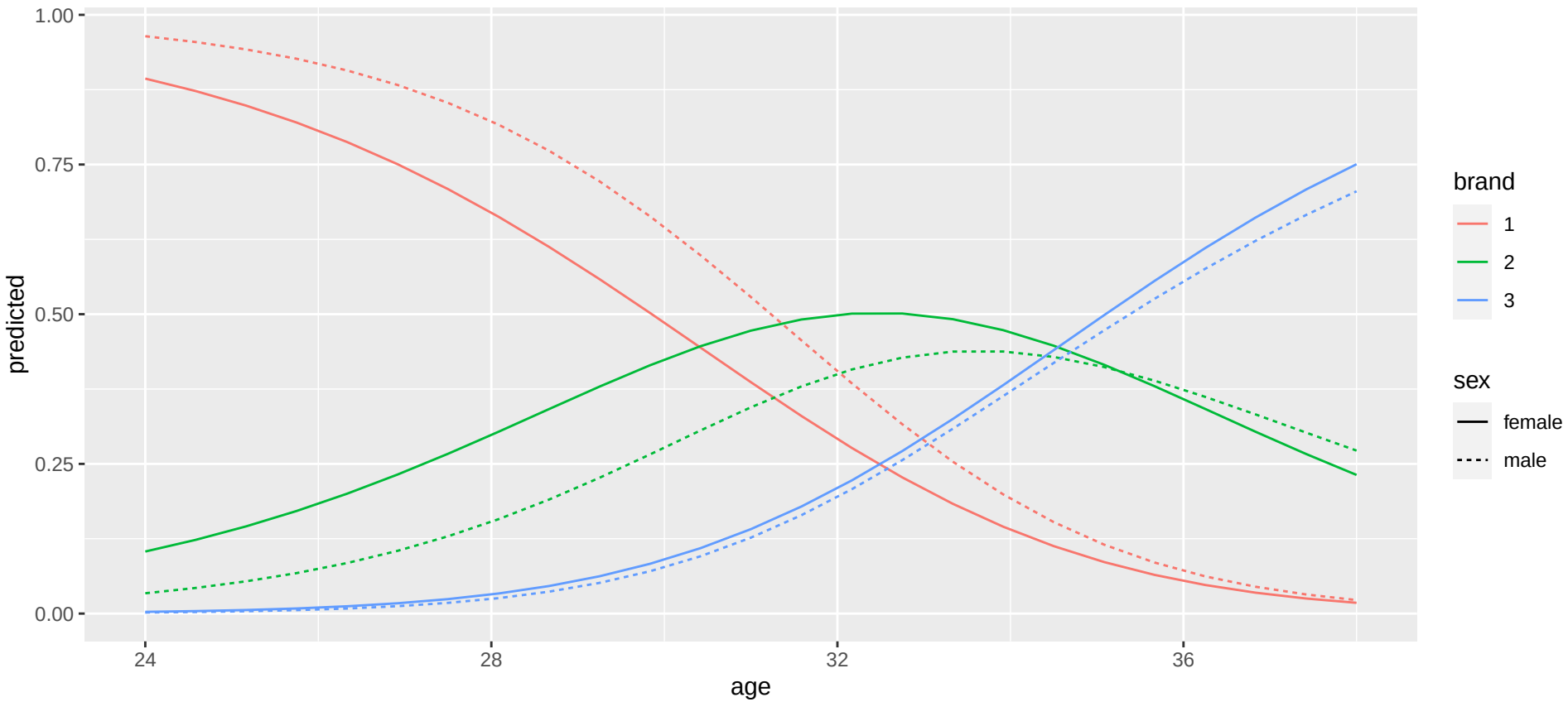
- No evidence that effect of age on brand preference differs for the two genders.

Make graph again

```
plot_cap(brands.4, condition = c("age", "brand", "sex"),
         type = "probs", draw = FALSE)  %>%
  rename(age = condition1, sex = condition3,
         brand = group) %>%
  ggplot(aes(x = age, y = predicted, colour = brand,
             linetype = sex)) +
  geom_line() -> g4
```

Not much difference in the graph

g4



Compare model without interaction

```
gg
```