

# 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")
```

# 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.



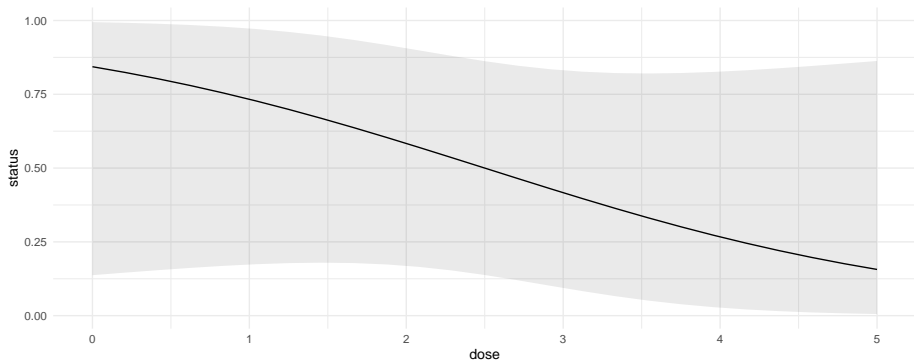
## Output part 2: predicted survival probs

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

rowid	type	predicted	std.error	conf.low	conf.high	status	dose
1	re- sponse	0.8434490	0.2373945	0.1370958	0.9945564	lived	0
2	re- sponse	0.7331122	0.2569246	0.1731865	0.9729896	died	1
3	re- sponse	0.5834187	0.2394051	0.1688476	0.9061463	lived	2
4	re- sponse	0.4165813	0.2394051	0.0938537	0.8311524	lived	3
5	re- sponse	0.2668878	0.2569246	0.0270104	0.8268135	died	4
6	re- sponse	0.1565510	0.2373945	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
##
```

## Predicted survival probs

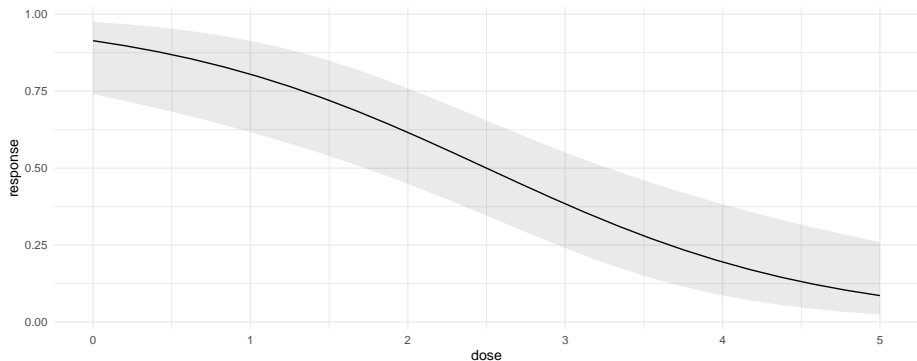
```
# p <- predict(rat2.1, type = "response")  
# cbind(rat2, p)  
predictions(rat2.1)
```

rowid	type	predicted	std.error	conf.low	conf.high	dose
1	response	0.9138762	0.0528798	0.7398304	0.9753671	0
2	response	0.8048905	0.0753564	0.6169584	0.9135390	1
3	response	0.6159474	0.0818379	0.4487610	0.7595916	2
4	response	0.3840526	0.0818379	0.2404084	0.5512390	3
5	response	0.1951095	0.0753564	0.0864609	0.3830417	4
6	response	0.0861238	0.0528798	0.0246329	0.2601697	5



# On a picture

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



# 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 for
```

```
## i Specify the column types or set `show_col_types = FALSE`
```

## Make sure categoricals really are

```
sepsis %>%  
  mutate(across(-age, ~factor(.))) -> 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 xxx

```
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	$(x + 1) - x$	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

# xxx 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	death	shock	malnut	alcohol	age	bowelinf
1	response	0.0014153	0.0022471	0	0	0	0	26	0
2	response	0.0205524	0.0166577	0	0	0	0	56	0
3	response	0.1534168	0.0740538	0	0	0	0	80	0
4	response	0.9312901	0.0785121	1	0	0	1	66	1
5	response	0.2130010	0.1012365	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 xxx

of age:

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

rowid	type	predicted	std.error	death	shock	alcohol	bowelinf	age
1	re-sponse	0.0006311	0.0011597	0	0	0	0	17.0
2	re-sponse	0.0026511	0.0037014	0	0	0	0	33.0
3	re-sponse	0.0150914	0.0134607	0	0	0	0	52.5
4	re-sponse	0.0631980	0.0358615	0	0	0	0	69.0
5	re-sponse	0.3892644	0.1790522	0	0	0	0	94.0

## Assessing effects 2/2 xxx

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

rowid	type	pre- dicted	std.error	death	alcohol	age	bow- elinf	shock
1	re- sponse	0.0135497	0.0124824	0	0	51.28302	0	0
2	re- sponse	0.3574261	0.2065571	0	0	51.28302	0	1

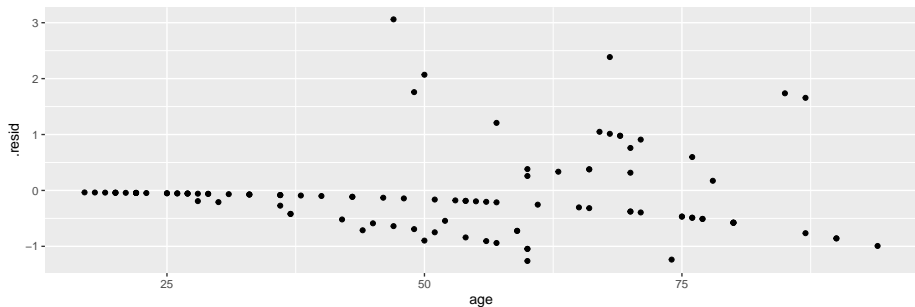
# Assessing proportionality of odds for age

- 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.

## xxx 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.

## xxx 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 = "miners")  
  mutate(Severity = fct_inorder(Severity)) -> miners
```

# xxx Result

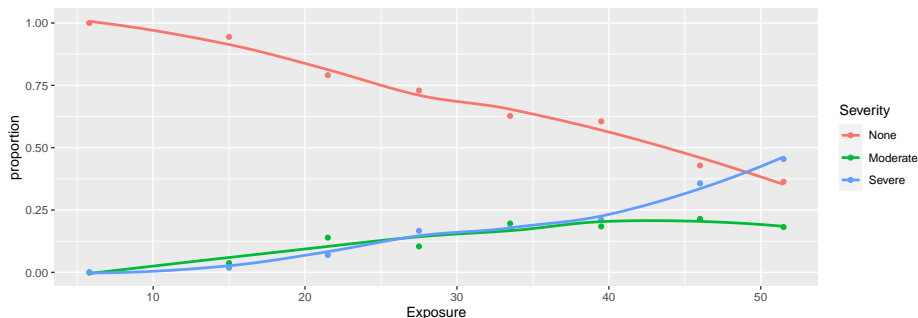
miners

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



# xxx 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)
```



## xxx 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

Logistic regression

## xxx Fitting ordered logistic model

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

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

## xxx Output: not very illuminating

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

```
##  
## Re-fitting to get Hessian  
  
## Call:  
## polr(formula = Severity ~ Exposure, data = miners, weights = Freq)  
##  
## 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
```

## xxx 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.

## xxx 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.

## xxx Predicted probabilities

```
freqs %>% select(Exposure) -> new  
new
```

Exposure
5.8
15.0
21.5
27.5
33.5
39.5
46.0
51.5

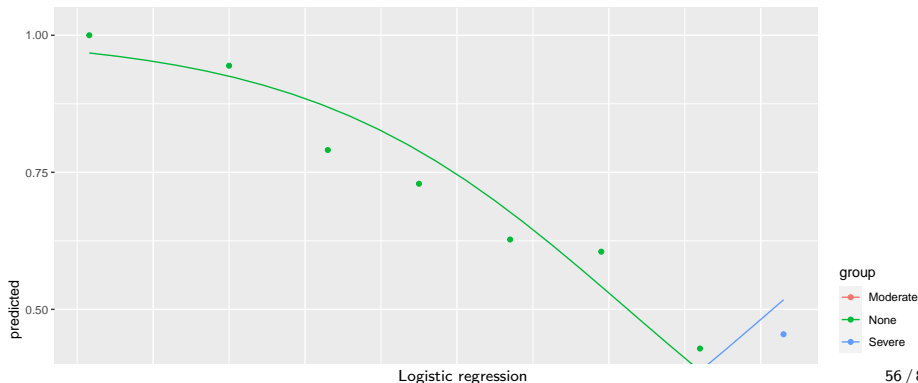
```
predictions(sev.1, newdata = new, type = "probs") %>%  
  select(group, predicted, Exposure = Exposure.x) %>%  
  pivot_wider(names_from = group, values_from = predicted)
```

# Plot of predicted probabilities

```
plot_cap(model = sev.1, condition = "Exposure", type = "probs") +  
  ggplot(aes(x = Exposure.x, y = predicted, colour = group)) +  
  geom_point(data = prop, aes(x = Exposure, y = proportion, co
```

##

## Re-fitting to get Hessian





## xxx Comments

- Model appears to match data: as exposure goes up, prob of None goes down, Severe goes up (sharply for high exposure).
- add

## xxx 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 for
## i Specify the column types or set `show_col_types = FALSE`
```

## xxx The data

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

## xxx Bashing into shape, and fitting model

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

```
brandpref <- brandpref %>%  
  mutate(sex = factor(sex)) %>%  
  mutate(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 702.976983  
## final value 702.970704  
## converged
```

## xxx 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 logical
```

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

## xxx Do age/sex help predict brand? 1/2

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
```

## xxx Do age/sex help predict brand? 2/2

```
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

## xxx Do age/sex help predict brand? 3/3

- age definitely significant (second anova)
- sex seems significant also (first anova)
- Keep both.



## xxx 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, data = brandpref)
```

```
##
```

```
## Coefficients:
```

```
## (Intercept)      age      sex1
```

```
## 2    -11.77469 0.3682075 0.5238197
```

```
## 3    -22.72141 0.6859087 0.4659488
```

```
##
```

```
## Residual Deviance: 1405.941
```

```
## AIC: 1417.941
```

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

## xxx Predictions: all possible combinations

Create data frame with various age and sex:

```
ages <- c(24, 28, 32, 35, 38)
sexes <- factor(0:1)
new <- crossing(age = ages, sex = sexes)
new
```

age	sex
24	0
24	1
28	0
28	1
32	0
32	1
35	0
35	1
38	0
38	1

## xxx Making predictions

```
p <- predict(brands.1, new, type = "probs")  
probs <- cbind(new, p)
```

or

```
p %>% as_tibble() %>%  
  bind_cols(new) -> probs
```

## xxx The predictions

probs

	1	2	3	age	sex
	0.9479582	0.0502293	0.0018125	24	0
	0.9153208	0.0818904	0.0027888	24	1
	0.7931320	0.1832969	0.0235711	28	0
	0.6956179	0.2714391	0.0329430	28	1
	0.4048727	0.4081032	0.1870241	32	0
	0.2908635	0.4950314	0.2141052	32	1
	0.1305782	0.3972405	0.4721813	35	0
	0.0840413	0.4316859	0.4842727	35	1
	0.0259816	0.2385507	0.7354677	38	0
	0.0162309	0.2516220	0.7321471	38	1

- Young males (sex=0) prefer brand 1, but older males prefer brand 3.
- Females similar, but like brand 1 less and brand 2 more.

## xxx Making a plot

- Plot fitted probability against age, distinguishing brand by colour and gender by plotting symbol.
- Also join points by lines, and distinguish lines by gender.
- I thought about facetting, but this seems to come out clearer.
- First need tidy data frame, by familiar process:

```
probs %>%  
  gather(brand, probability, -(age:sex)) -> probs.long
```

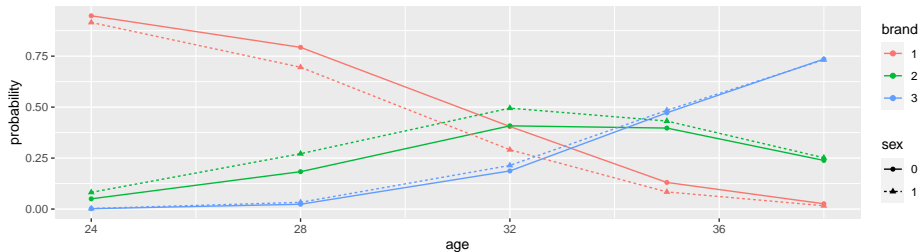
## xxx The tidy data (random sample of rows)

```
probs.long %>% sample_n(10)
```

age	sex	brand	probability
32	0	1	0.4048727
38	0	3	0.7354677
28	1	1	0.6956179
28	0	3	0.0235711
38	1	3	0.7321471
28	0	1	0.7931320
35	1	1	0.0840413
24	0	1	0.9479582
32	0	3	0.1870241
35	0	1	0.1305782

## xxx The plot

```
ggplot(probs.long, aes(  
  x = age, y = probability,  
  colour = brand, shape = sex  
)) +  
  geom_point() + geom_line(aes(linetype = sex))
```



## xxx 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 (dashed) like brand 1 less than males (solid), 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.



## xxx 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?

## xxx Getting alternative data format

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

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

```
b %>% slice(1:6)
```

age	sex	brand	Freq
24	0	1	1
26	0	1	2
27	0	1	4
27	1	1	4
27	1	3	1
28	0	1	4

## xxx 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)
```

## xxx 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.

## xxx 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

## xxx 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
```

## xxx Checking proportions for age 32

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

age	sex	brand	Freq	proportion
32	0	1	48	0.4067797
32	0	2	51	0.4322034
32	0	3	19	0.1610169
32	1	1	62	0.2883721
32	1	2	117	0.5441860
32	1	3	36	0.1674419

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

## xxx Attempting plot

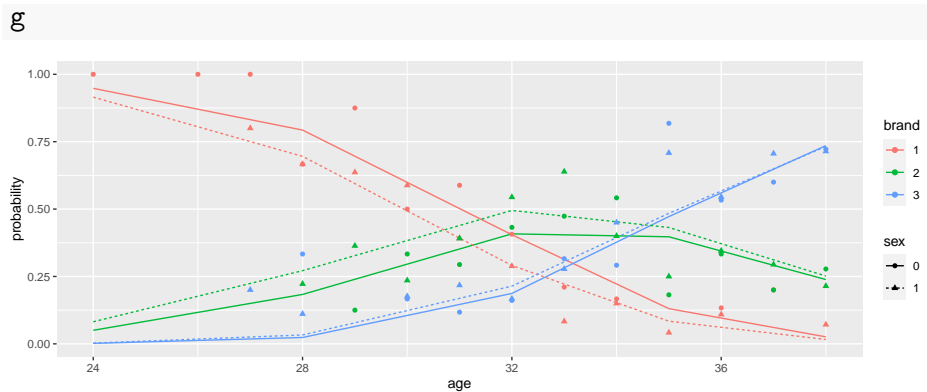
- Take code from previous plot and:
- remove `geom_point` for fitted values
- add `geom_point` with correct `data=` and `aes` to plot data.

```
g <- ggplot(probs.long, aes(  
  x = age, y = probability,  
  colour = brand, shape = sex  
)) +  
  geom_line(aes(linetype = sex)) +  
  geom_point(data = brands, aes(y = proportion))
```

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



# xxx The plot

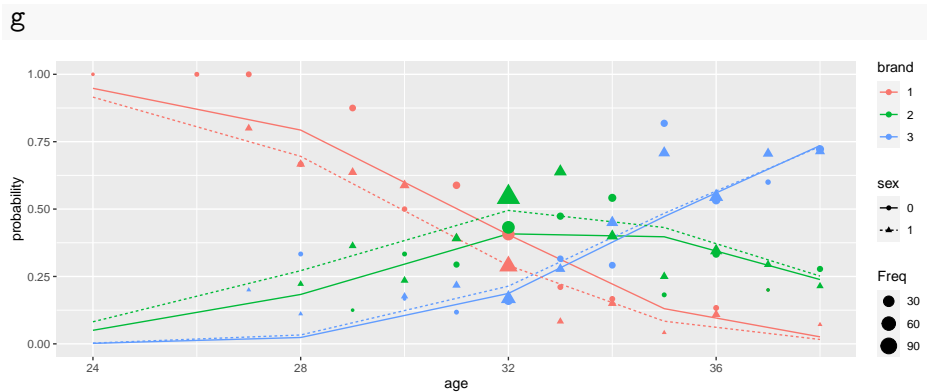


## xxx But...

- 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 <- ggplot(probs.long, aes(  
  x = age, y = probability,  
  colour = brand, shape = sex  
)) +  
  geom_line(aes(linetype = sex)) +  
  geom_point(  
    data = brands,  
    aes(y = proportion, size = Freq)  
  )
```

# xxx The plot



## xxx Trying interaction between age and gender

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

```
## # weights: 15 (8 variable)
## initial value 807.480032
## iter 10 value 704.811229
## iter 20 value 702.582802
## final value 702.582761
## converged
```

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

Model	Resid. df	Resid. Dev	Test	Df	LR stat.	Pr(Chi)
age + sex	124	1405.941		NA	NA	NA
age + sex + age:sex	122	1405.166	1 vs 2	2	0.7758861	0.678451

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