THE UNIVERSITY of EDINBURGH

Week 10: The Generalized Linear Model

Univariate Statistics and Methodology using R

Department of Psychology
The University of Edinburgh

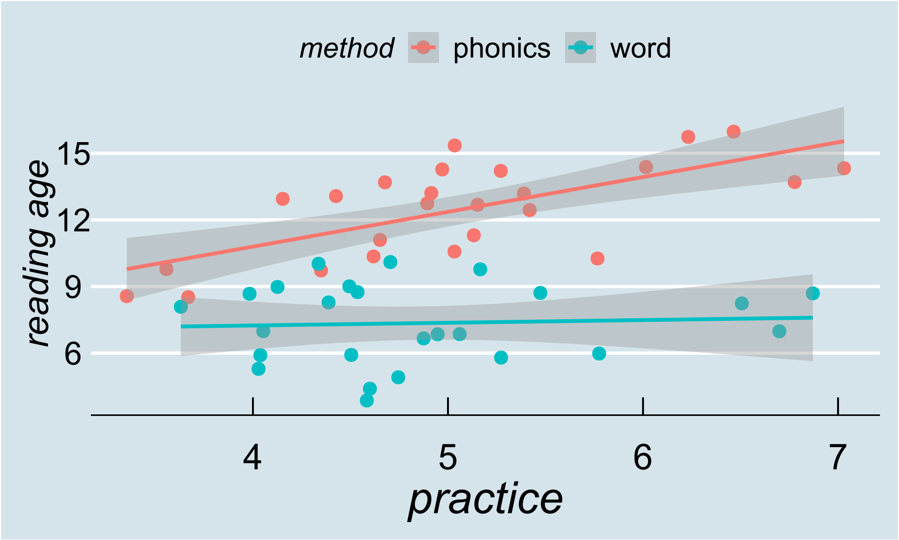
Part 1

Bigger and Better

Learning to Read

age	hrs_wk	method	R_AGE
10.115	4.971	phonics	14.272
9.940	4.677	phonics	13.692
6.060	4.619	phonics	10.353
9.269	4.894	phonics	12.744
10.991	5.035	phonics	15.353
6.535	5.272	word	5.798
8.150	6.871	word	8.691
7.941	4.053	word	6.988
8.233	5.474	word	8.713
6.219	4.038	word	5.908

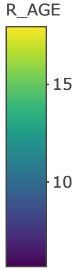
Learning to Read



Learning to Read

age	hrs_wk	method	R_AGE
10.115	4.971	phonics	14.272
9.940	4.677	phonics	13.692
6.060	4.619	phonics	10.353
9.269	4.894	phonics	12.744
10.991	5.035	phonics	15.353
6.535	5.272	word	5.798
8.150	6.871	word	8.691
7.941	4.053	word	6.988
8.233	5.474	word	8.713
6.219	4.038	word	5.908

Learning to Read



Bigger and Better

- easy to build models including more predictors

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_k x_{ki} + \ldots + b_m x_{1i} x_{2i} + b_{m+1} x_{2i} x_{3i} + \ldots$$

- for example

```
mod.mm <- lm(R_AGE ~ age + hrs_wk + method + hrs_wk:method + age:hrs_wk, data=reading)
```

Bigger and Better

- easy to build models including more predictors

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_k x_{ki} + \ldots + b_m x_{1i} x_{2i} + b_{m+1} x_{2i} x_{3i} + \ldots$$

- for example

```
mod.mm <- lm(R_AGE ~ age + hrs_wk + method + hrs_wk:method + age:hrs_wk, data=reading)
```

- NB., order of predictors **can** matter (judgement is important)
    - if we conduct `anova(mod.mm)` we test *incremental addition* of each predictor
- first question: is it worth it building such a complex model?

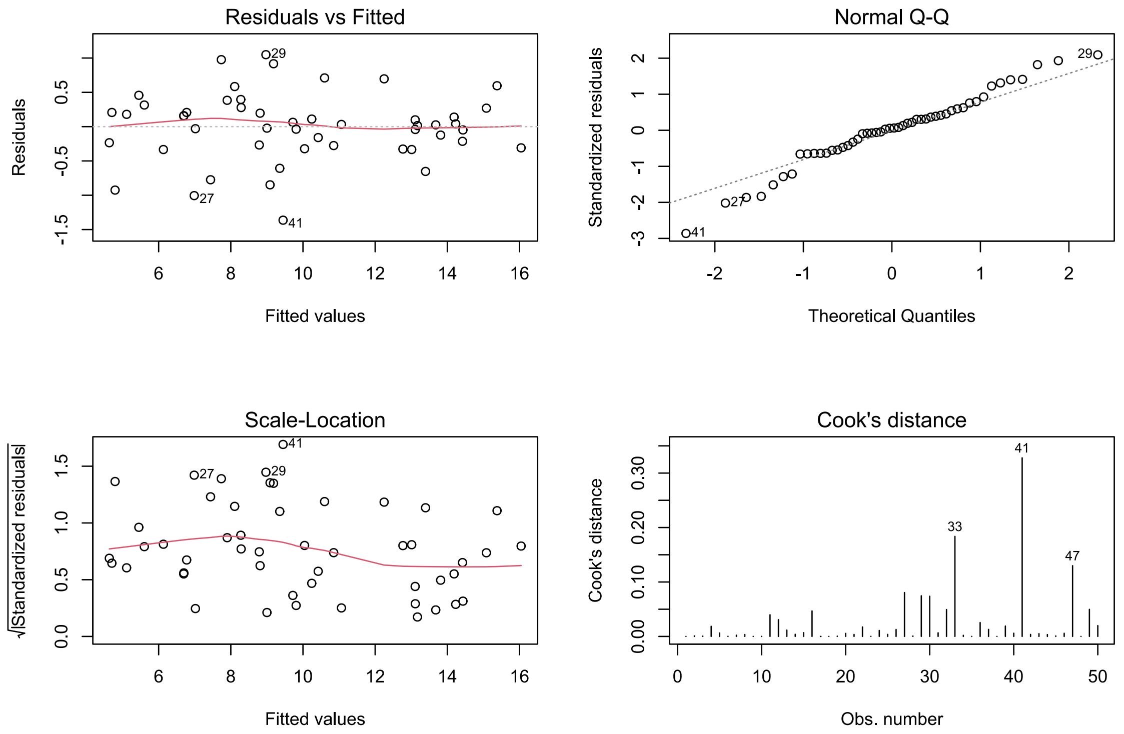
Does Each New Predictor Improve Fit?

```
anova(mod.mm)
```

```
## Analysis of Variance Table
##
## Response: R_AGE
##                Df Sum Sq Mean Sq F value  Pr(>F)
## age             1  166.0   166.0  599.36 < 2e-16 ***
## hrs_wk          1   35.7    35.7  128.99 1.1e-14 ***
## method          1  300.2   300.2 1083.79 < 2e-16 ***
## hrs_wk:method   1    2.8     2.8   10.25  0.0025 **
## age:hrs_wk      1    0.1     0.1    0.27  0.6078
## Residuals      44   12.2     0.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- adding `age:hrs_wk` doesn't improve the model any further over a model without it

```
mod.mm <- update(mod.mm, ~ . -age:hrs_wk)
# equivalent
# mod.mm <- lm(R_AGE ~ age + hrs_wk + method + hrs_wk:method, data=reading)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Cook's distance
Cook's distance
Obs. number

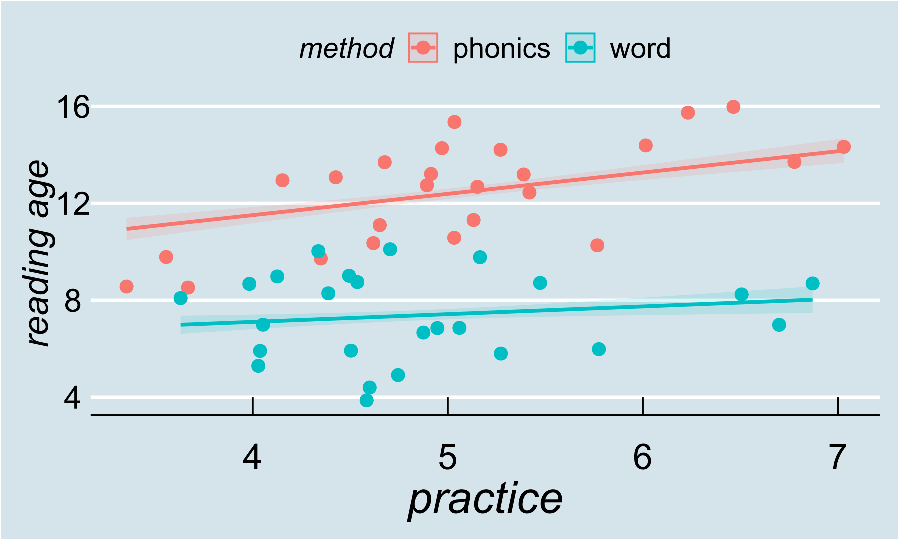
The Model

```
##
## Call:
## lm(formula = R_AGE ~ age + hrs_wk + method + hrs_wk:method, data = reading)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3637 -0.2737  0.0288  0.2538  1.0491
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.6849     0.6114    1.12   0.2686
## age                   0.9076     0.0428   21.22  < 2e-16 ***
## hrs_wk                0.8785     0.1177    7.46  2.1e-09 ***
## methodword           -2.1643     0.8724   -2.48   0.0169 *
## hrs_wk:methodword    -0.5599     0.1734   -3.23   0.0023 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.522 on 45 degrees of freedom
## Multiple R-squared:  0.976,	Adjusted R-squared:  0.974
## F-statistic:  463 on 4 and 45 DF,  p-value: <2e-16
```

- coef_as_pred.R

The Model

- not always convenient to draw 3d models!
- graphs can show "interesting" results
- here, age doesn't interact with anything
- so show plot for *mean age* (or some other meaningful value)



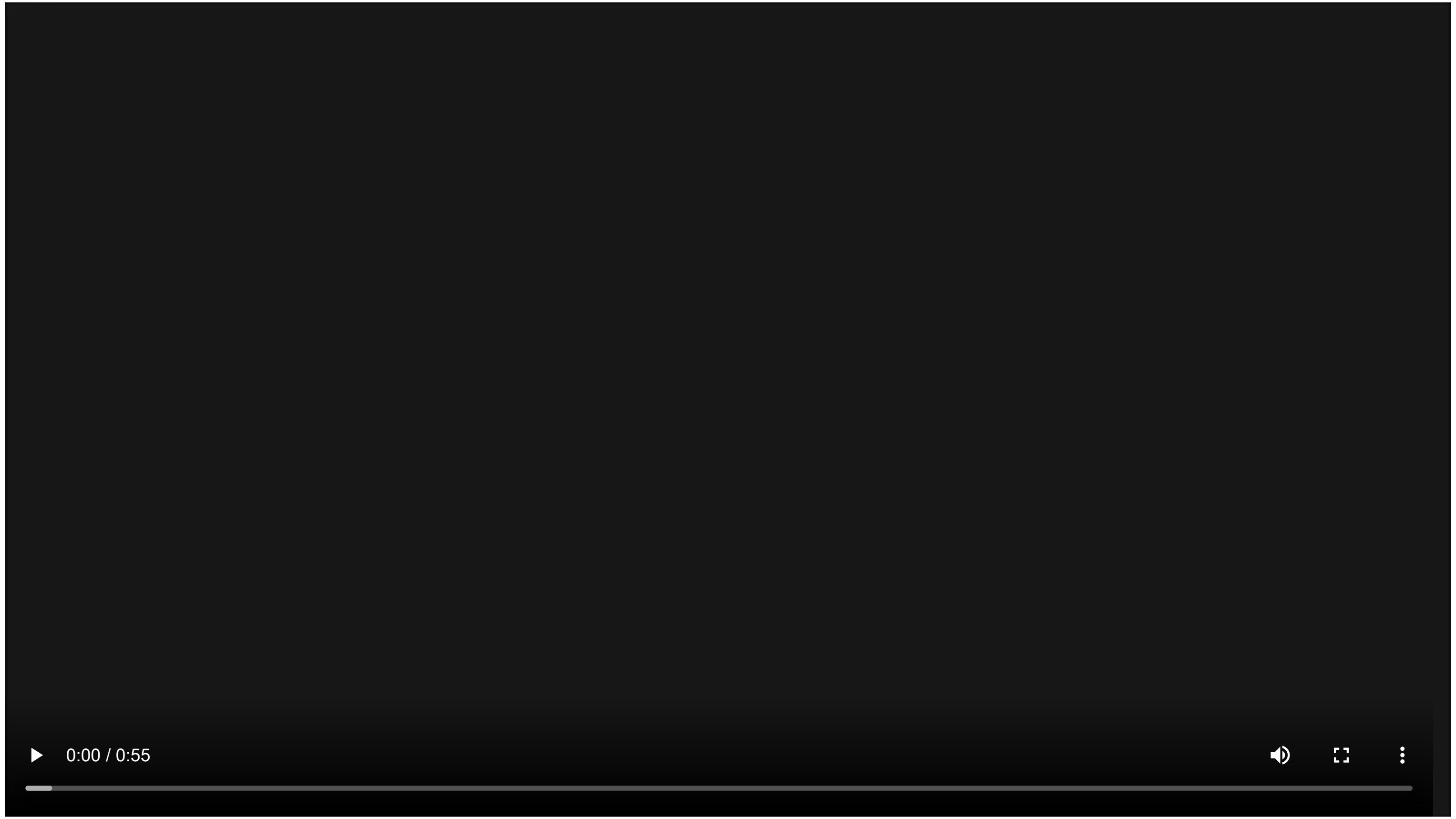
reading age predicted by practice hours per week for children of average age

End of Part 1

Part 2

Probability, Odds, Log-Odds

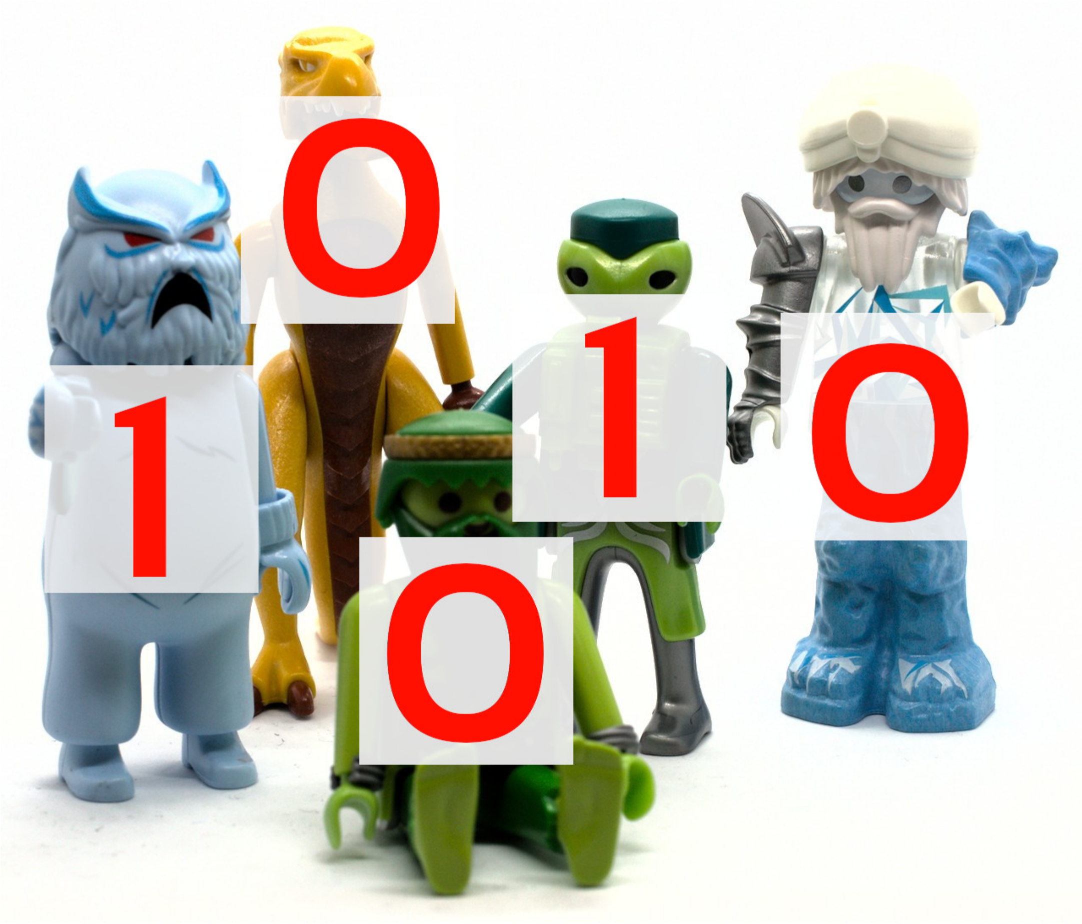
Aliens



A Binary World

A Binary World

A Binary World

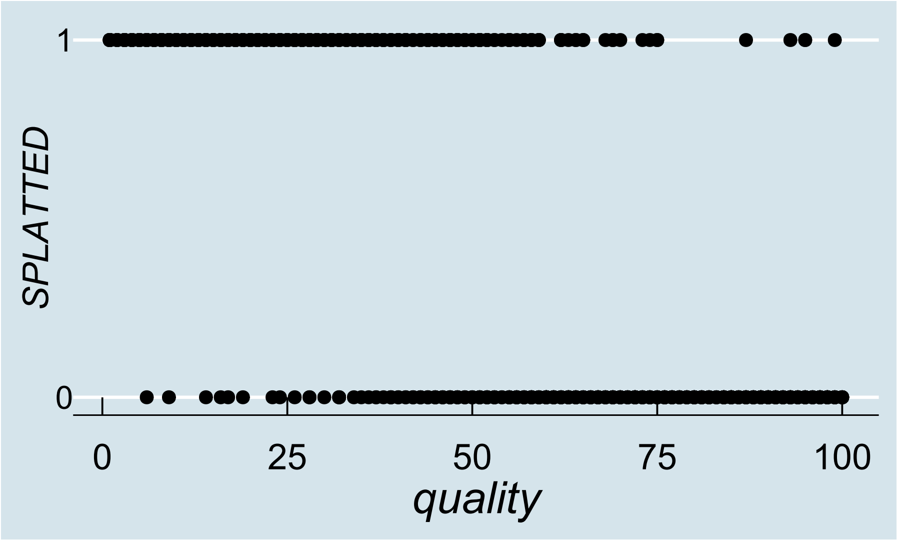


1,000 Aliens

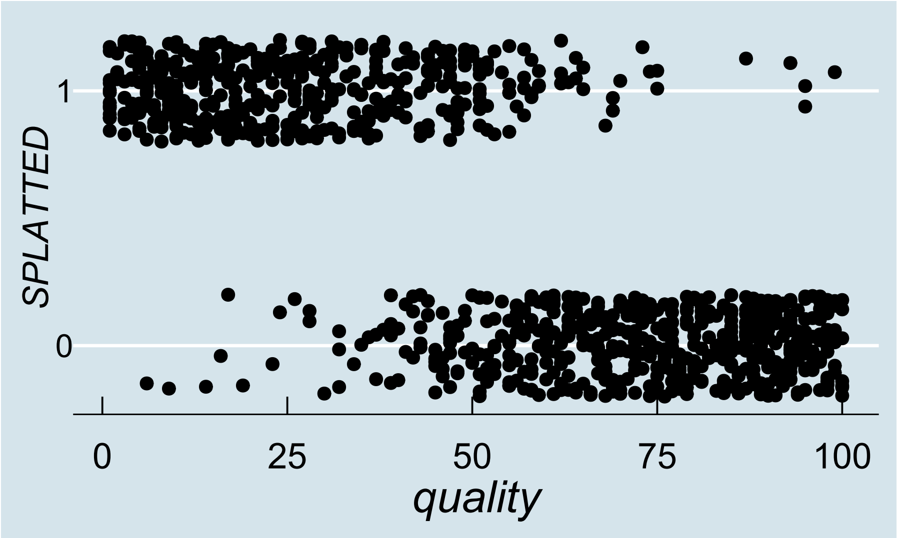
id	quality	SPLATTED
The Great Odorjan of Erpod	84	0
Hapetox Bron	34	1
Loorn Molzeks	92	0
Ba'lite Adrflen	49	1
Tedlambo Garilltet	93	0
Goraveola Grellorm	5	1
Colonel Garqun	55	1
Bosgogo Lurcat	64	1
Osajed Voplily	45	0
Subcommander Edorop	90	0

- `quality` = quality of singing
- `SPLATTED` = whether splatted (1 or 0)

1,000 Aliens



1,000 Aliens



- using `geom_jitter()`

Binomial Regression, Conceptually

- each alien either gets splatted or doesn't
  - each observation is either a 1 or a 0
- underlyingly, there's a **binomial** distribution
- for each value of "quality of singing" there's a *probability* of getting splatted

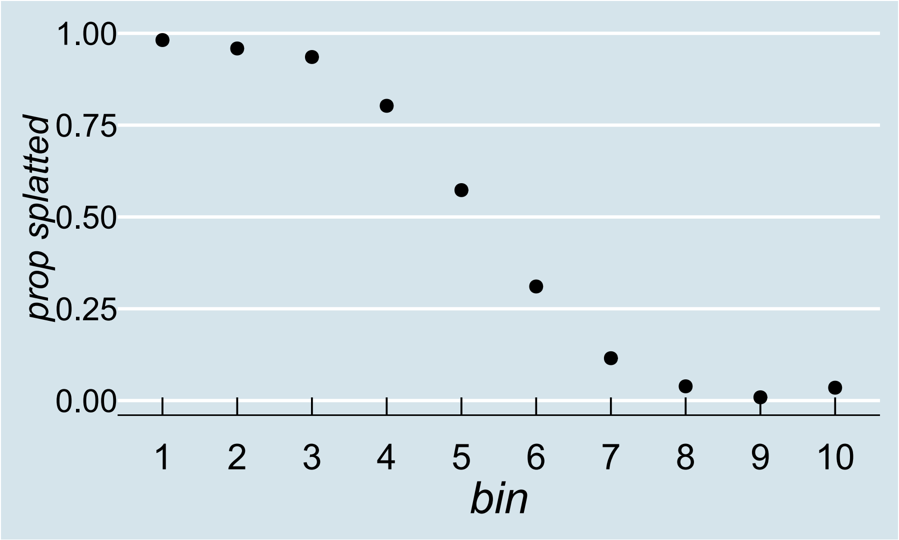
Binomial Regression, Conceptually

- each alien either gets splatted or doesn't
  - each observation is either a 1 or a 0
- underlyingly, there's a **binomial** distribution
- for each value of "quality of singing" there's a *probability* of getting splatted

- for each alien, the outcome is deterministic
- but it's the *probability* we are ultimately interested in
- we can approximate it by binning our data...

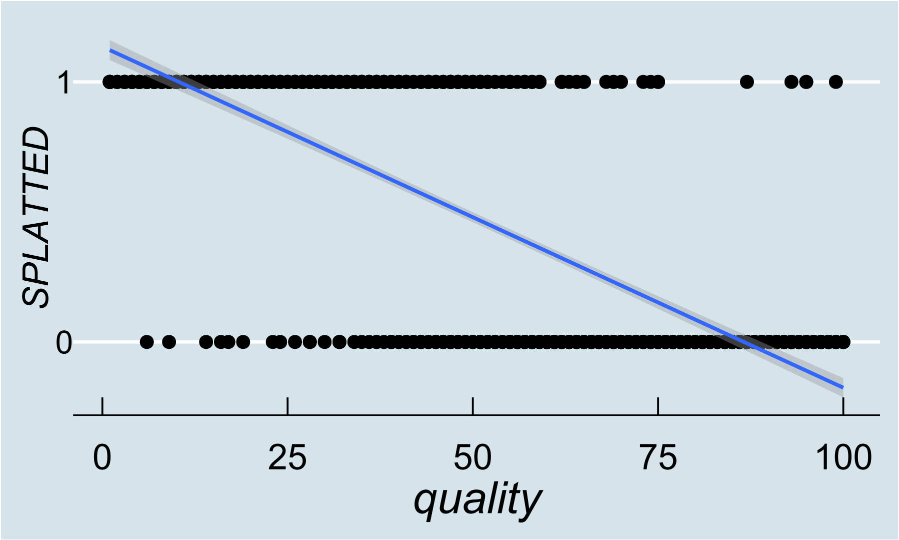
Binned Data

```
singers <- singers %>%
  mutate(bin=cut_interval(quality,10))

dat <- singers %>% group_by(bin) %>%
  summarise(prop=mean(SPLATTED))

dat %>% ggplot(aes(x=bin,y=prop)) +
  xlab("bin") + ylab("prop splatted") +
  geom_point(size=3) +
  scale_x_discrete(label=1:10)
```

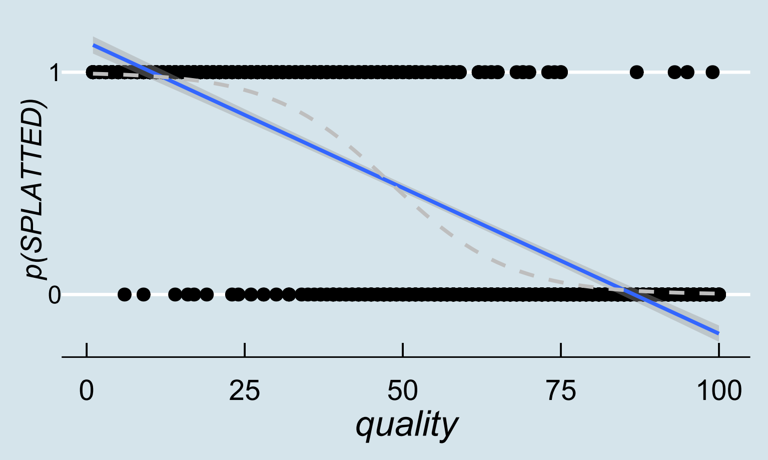


Best Fit Lines

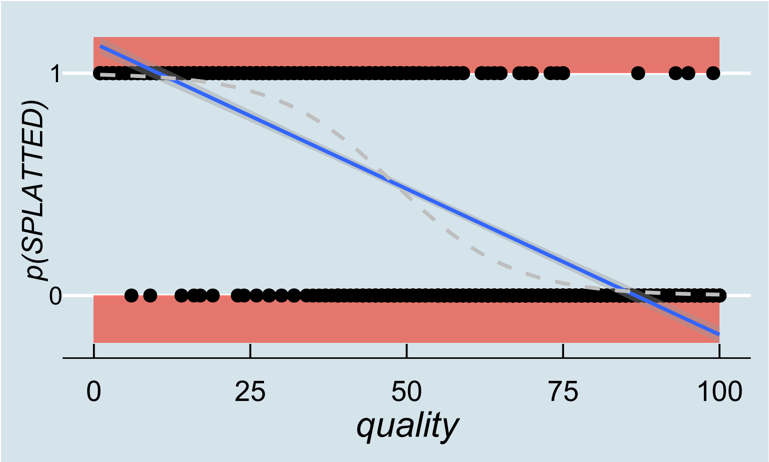
- we can fit our data using a standard linear model
- but there's something very wrong...



The Problem with Probability

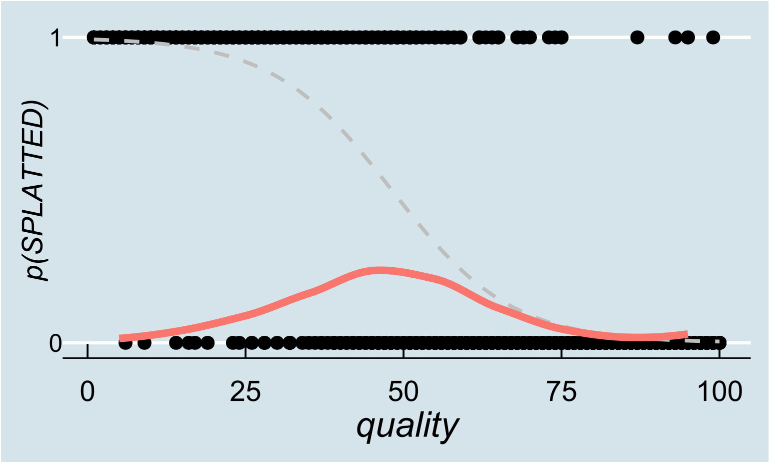


The Problem with Probability



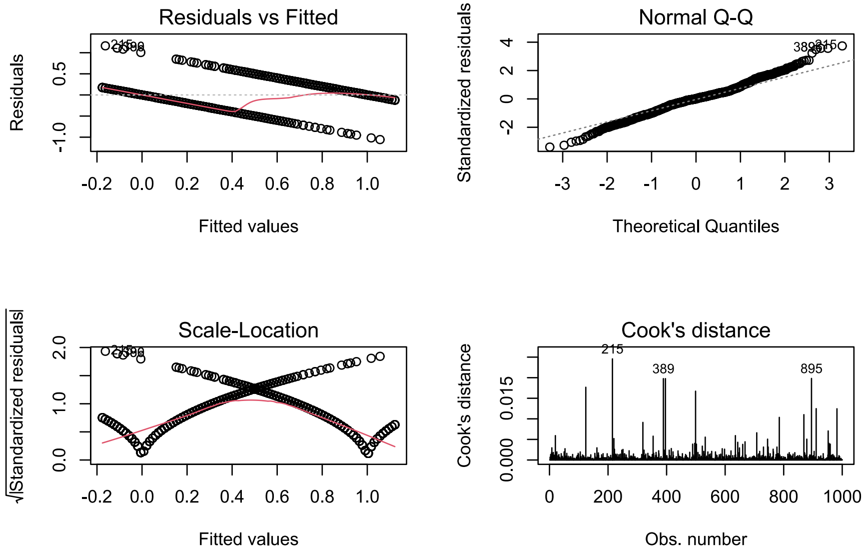
- a *linear* model predicts impossible values because probability isn't linear; it's **asymptotic**

The Problem with Probability



- variance *necessarily* covaries with probability

Assumptions



Probability and Odds

$$\text{odds}(y) = \frac{p(y)}{1 - p(y)}$$

$$0 < p < 1$$

$$0 < \text{odds} < \infty$$

Probability and Odds

$$\mathrm{odds}(y) = \frac{p(y)}{1 - p(y)}$$

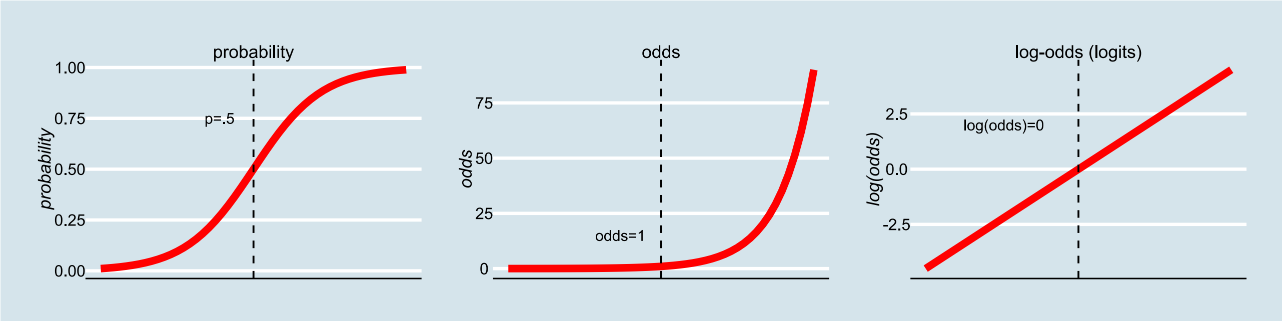
$$0 < p < 1$$

$$0 < \mathrm{odds} < \infty$$

	$p(y)$	$\mathrm{odds}(y)$
throw heads	$\frac{1}{2}$	$\frac{1}{1}$
throw 8 from two dice	$\frac{5}{36}$	$\frac{5}{31}$
get splatted	$\frac{99}{100}$	$\frac{99}{1}$

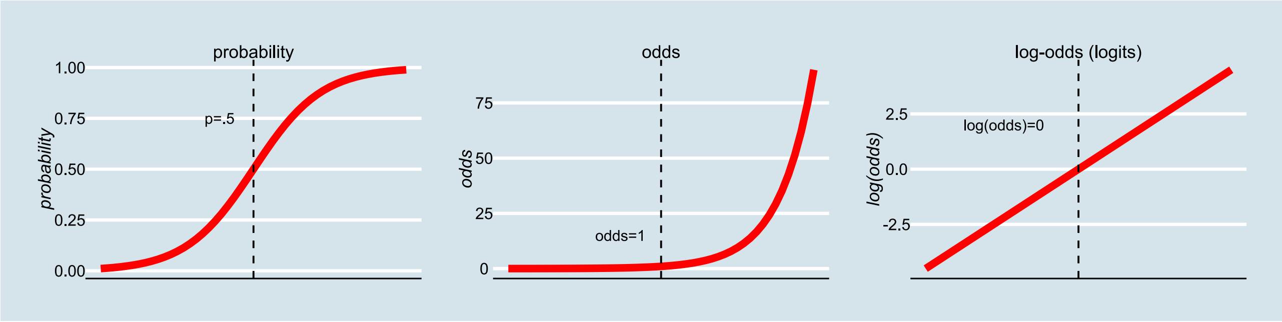
Probability and Log-Odds

- $\log(0) = -\infty; \log(\infty) = +\infty$
- $\log(1) = 0$ where odds of 1 are exactly 50:50 ($p = 0.5$)



Probability and Log-Odds

- $\log(0) = -\infty; \log(\infty) = +\infty$
- $\log(1) = 0$ where odds of 1 are exactly 50:50 ($p = 0.5$)



- if log-odds are *less than zero*, the odds go down (multiply by <1)
- if log-odds are *more than zero*, the odds go up (multiply by >1)
- high odds = high probability

End of Part 2

Part 3

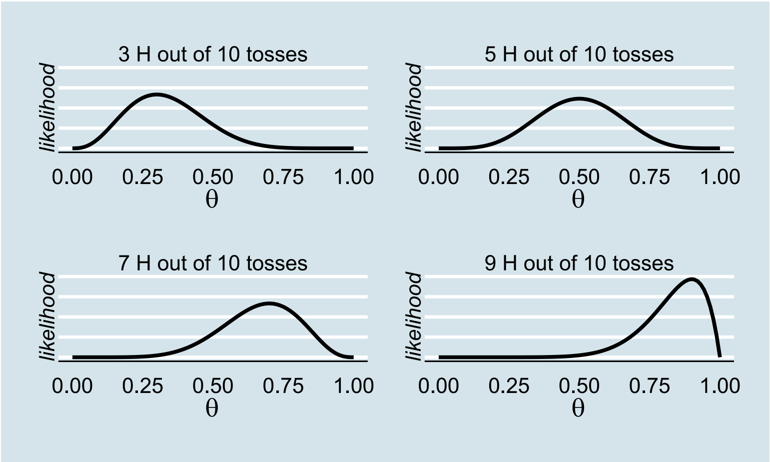
The Generalized Linear Model

The Generalized Linear Model

- generalises the linear model using mapping functions
- coefficients are in **logit** (log-odds) units

- fit using **maximum likelihood**
- coefficients use **Wald's** z instead of t

Likelihood



- extent to which a sample provides support for a model (MLE_bend_in_the_road.R).

The Generalized Linear Model

- generalises the linear model using mapping functions
- coefficients are in **logit** (log-odds) units

- fit using **maximum likelihood**
- coefficients use **Wald's** z instead of t
- but actually it's all quite straightforward...

Alien Singer Splat Probability

id	quality	SPLATTED
The Great Odorjan of Erpod	84	0
Hapetox Bron	34	1
Loorn Molzeks	92	0
Ba'lite Adrflen	49	1
Tedlambo Garilltet	93	0
Goraveola Grellorm	5	1
Colonel Garqun	55	1
Bosgogo Lurcat	64	1
Osajed Voplily	45	0
Subcommander Edorop	90	0

- use `glm()` * instead of `lm()`
- specify **link function** with `family = binomial` **

```
mod.b <- glm(SPLATTED ~ quality,
             family = binomial,
             data=singers)
```

* can take a 2-level factor DV
** `family="binomial"` and `family=binomial(link="logit")` also work

Evaluating the Model

- NB., no statistical test done by default
- **deviance** compares the likelihood of the new model to that of the previous model
  - a generalisation of sums of squares
  - *lower* "residual deviance" is good
    (*a bit like Residual Sums of Squares*)

```
summary(mod.b)
```

```
##
## Call:
## glm(formula = SPLATTED ~ quality, family = binomial, data = singers)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.987  -0.374  -0.113   0.333   3.279
## ...
## ...
## ...
##     Null deviance: 1377.06  on 999  degrees of freedom
## Residual deviance:  577.29  on 998  degrees of freedom
```

Evaluating the Model

```
##     Null deviance: 1377.06  on 999  degrees of freedom
## Residual deviance:  577.29  on 998  degrees of freedom
```

- deviance is $-2\times$ the log-likelihood ratio of the reduced compared to the full model
- *higher* "deviance" is good (*a bit like F*)

```
mod.n <- glm(SPLATTED~1,family=binomial,data=singers)
```

```
logLik(mod.n)
```

```
## 'log Lik.' -688.5 (df=1)
```

```
logLik(mod.b)
```

```
## 'log Lik.' -288.6 (df=2)
```

```
-2*logLik(mod.n)
```

```
## 'log Lik.' 1377 (df=1)
```

```
-2*logLik(mod.b)
```

```
## 'log Lik.' 577.3 (df=2)
```

```
-2 * (logLik(mod.n)-logLik(mod.b))
```

```
## 'log Lik.' 799.8 (df=1)
```

Evaluating the Model

- model deviance maps to the χ^2 distribution
- can specify a χ^2 test to statistically evaluate model in a similar way to F ratio

```
anova(mod.b, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: SPLATTED
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                      999       1377
## quality  1      800       998        577   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Coefficients

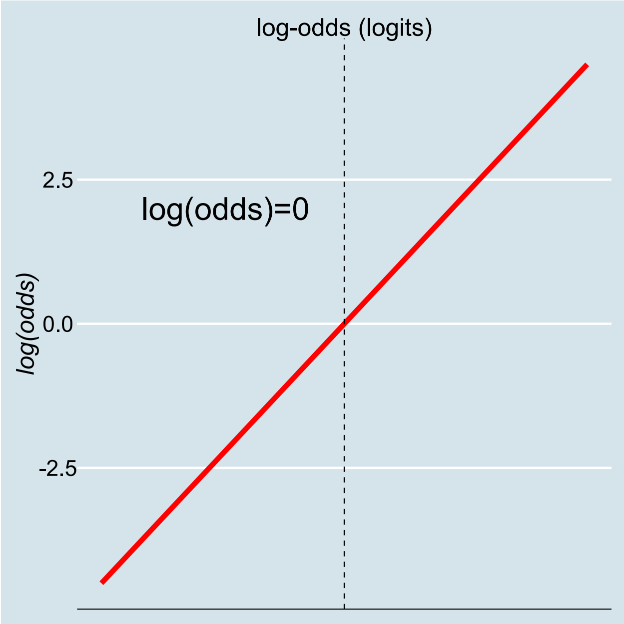
```
##
## Call:
## glm(formula = SPLATTED ~ quality, family = binomial, data = singers)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.987  -0.374  -0.113   0.333   3.279
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08191    0.33410    15.2   <2e-16 ***
## quality     -0.10557    0.00642   -16.5   <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: 1377.06  on 999  degrees of freedom
## Residual deviance:  577.29  on 998  degrees of freedom
## AIC: 581.3
##
## Number of Fisher Scoring iterations: 6
```

Model Coefficients

coefficients are in logits (= log-odds)

```
## ...
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08191    0.33410    15.2   <2e-16 ***
## quality     -0.10557    0.00642   -16.5   <2e-16 ***
## ...
```

- zero = "50/50" (odds of 1)
- value below zero: probability of being splatted *decreases* as quality increases



Log-Odds, Odds, and Probability

```
## ...
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08191    0.33410    15.2   <2e-16 ***
## quality     -0.10557    0.00642   -16.5   <2e-16 ***
## ...
```

quality = 50

- *log-odds*: $5.08 + -0.11 \cdot 50 = -0.42$
- *odds*: $e^{-0.42} = 0.657$
- *probability*: $\frac{0.657}{1+0.657} = 0.3965$

$$\hat{y}_i = b_0 + b_1 x_i$$

$$\text{odds} = e^{\hat{y}_i}$$

$$p = \frac{\text{odds}}{1 + \text{odds}}$$

A Useful Function

- intuitive to think in probability
- useful to write a function which takes a value in logits `l` and converts it to a probability `p`

```
l2p <- function(logits) {
  odds = exp(logits)
  prob = odds/(1+odds)
  return(prob)
}
```

- singing qualities 50 and 51

```
l2p(5.08+-0.11*50)
```

```
## [1] 0.3965
```

```
l2p(5.08+-0.11*51)
```

```
## [1] 0.3705
```

- singing qualities 10 and 11

```
l2p(5.08+-0.11*10)
```

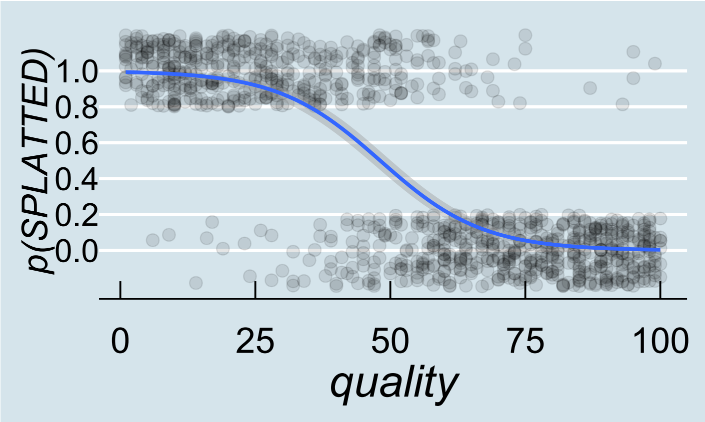
```
## [1] 0.9817
```

```
l2p(5.08+-0.11*11)
```

```
## [1] 0.9796
```

Representing the Model Graphically

```
singers %>% ggplot(aes(x=quality,y=SPLATTED)) +
  ylab("p(SPLATTED)") +
  geom_jitter(size=3,width=0,height=.2,alpha=.1) +
  geom_smooth(method="glm",method.args=list(family=binomial)) +
  scale_y_continuous(breaks=seq(0,1,by=.2))
```



One Last Trick

- so far we've looked at
  - model *deviance* and χ^2 (similar to sums of squares and F)
  - model *coefficients* and how to map them to probability
- what about "explained variance" (similar to R^2)?
- no really good way of doing this, many proposals
- SPSS uses something called "accuracy" (how well does the model predict actual data?)
- not very informative, but good for learning R

Accuracy

- first, what does the model predict (in logit units)?

```
guess <- predict(mod.b) # in logit units
```

- if the chance of being splatted is more than .5 (logit > 0) call it a "splat"

```
guess <- ifelse(guess>0,1,0)
```

- how well do predicted splats match actual splats?

```
hits <- sum(guess == singers$SPLATTED)
hits/length(singers$SPLATTED)
```

```
## [1] 0.879
```

- present model "correctly predicts" 87.9% of the observations

Other Types of Data

- logit regression is *one type* of GLM
- others make use of different **link functions** (through `family=...`)

- **poisson**: number of events in a time period
- **inverse gaussian**: time to reach some criterion
- ...

GLMs

Predictor Variables

- linear
- convertible to linear (use `log()` etc.)
- non-convertible (use `contrasts()` etc. to map)
- **don't affect the choice of model**

Dependent Variables

- linear
- convertible to linear (use `log()` etc.)
- non-convertible (use `glm()` with `family=...`)
- **directly affect the choice of model**

End

Acknowledgements

- icons by Diego Lavecchia from the Noun Project