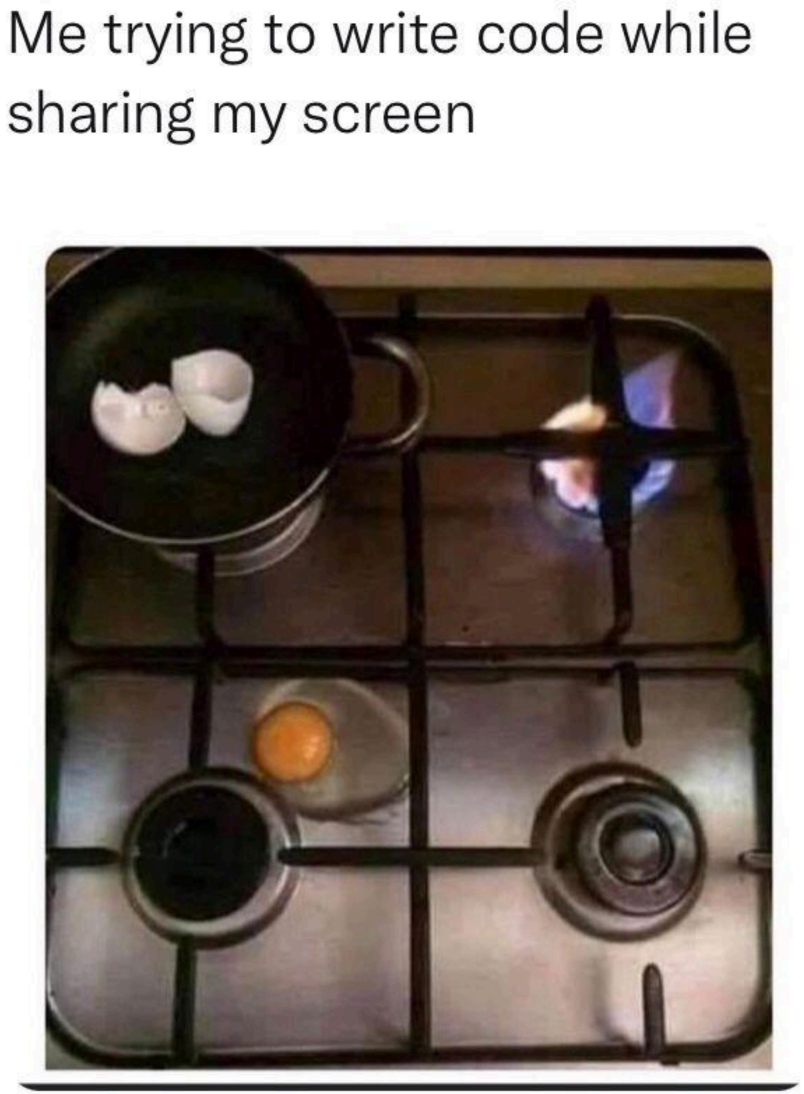
Logistic vs Binomial regressions and other alternatives to model binary outcomes

Mabel Carabali

EBOH, McGill University

Updated: 2024-10-08



Me trying to write code while sharing my screen

Objective: To review core concepts of logistic regression and identify opportunities to estimate absolute and relative measures of association when the outcome is binary

Outline

1) (mini) Review of concepts (Slides 4-17)

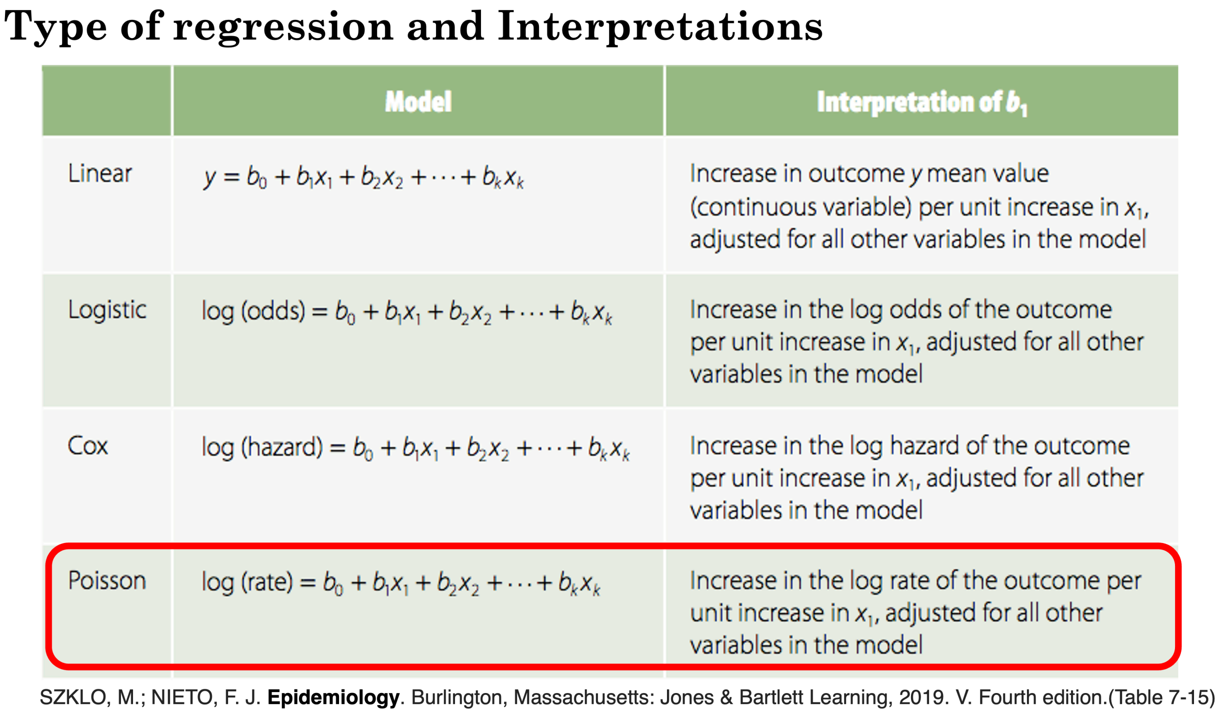
- Outcome’s distribution and study designs
- Regression adjustment (Why and When)

2) Alternatives to obtain absolute measures when the outcome is binary (Slides 18-85)

- Prediction (at the modes, means)
- Marginal standardization
- Log binomial regression

Extra slides: For own review (86- 95)

What do we know so far?



Type of regression and Interpretations

	Model	Interpretation of b_1
Linear	$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$	Increase in outcome y mean value (continuous variable) per unit increase in x_1, adjusted for all other variables in the model
Logistic	$\log(\text{odds}) = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$	Increase in the log odds of the outcome per unit increase in x_1, adjusted for all other variables in the model
Cox	$\log(\text{hazard}) = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$	Increase in the log hazard of the outcome per unit increase in x_1, adjusted for all other variables in the model
Poisson	$\log(\text{rate}) = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$	Increase in the log rate of the outcome per unit increase in x_1, adjusted for all other variables in the model

SZKLO, M.; NIETO, F. J. **Epidemiology**. Burlington, Massachusetts: Jones & Bartlett Learning, 2019. V. Fourth edition.(Table 7-15)

SZKLO, M.; NIETO, F. J.Epidemiology. Burlington, Massachusetts: Jones & Bartlett Learning, 2019. V. Fourth edition.(Table 7-15)

What do we know so far?

Method	Characteristics	Outcome	Measure
Standardization	Weight-based adjustment; Depends on the standard pop. selected; No homogeneity needed	Binary or categorical	SMR
Mantel- Haenszel Adjustment	Requires homogeneity; Do not handle clusters	Binary or categorical	RD, RR, OR
Regression Adjustment	Efficient, Useful for prediction, adjust for several covariates, *require assumptions*	Any type	RD, RR, OR; AME/ATE
IPTW [1]	Regression + Weights: 1/Pr(X=1, covars); Ensure Exchangeability; Only for measured Confounders	Any type	**Causal** RD, RR, OR; AME/ATE

[1] More on this later, hopefully with the help of this lecture

But wait... When and Why?

Model Assumptions and Considerations

- What is the distribution of the data (for a fixed pattern of covariates)?
- Are the model-specific assumptions met?
- What function will be used to *link* the mean of the data to the covariates?
- Which covariates should be included in the model?

Remember this table?

Sample	Risk Among Exposed	Risk Among Non-Exposed	Risk Difference	Risk Ratio	Odds Ratio
63	0.25	0.22	0.03	1.12	1.17
63	0.17	0.15	0.02	1.12	1.15
630	0.017	0.015	0.002	1.12	1.15
630	0.25	0.22	0.03	1.12	1.17

But we want *meaningful* and complementary measures (e.g., RD, RR)!!

“But my outcome is dichotomous!”

It doesn’t matter!!!

"we are not chained to our output" let's not fell for the "Risk relativism" By Poole

We also have options and a number of tools at our disposal to directly estimate risks, RRs and RDs

Different Approaches

Parametric: Data/Outcome dependent (Assumptions!)

1) Frequentist approach

- Deductive $\rightarrow P(Data|H_0)$
- Uncertainty is given by the 95%Confidence Interval
- Maximum likelihood Estimates
- Consistent, efficient, asymptotically normal

2) Bayesian approach

- Inductive $P(\theta|Data)$
- MCMC, priors!

Be aware of multiplicative models, sample size and number of parameters!

Data cleaning: variables' coding and missing data

Simple models for generating absolute & relative estimates

Type	Model	Estimate
Continuous	Linear Regression	RD
Binary	Logistic Regression, Binomial Regression	OR, **RR**, **RD**
Counts	Poisson, Negative Binomial [1]	IR, IRR, **RD**

Assuming:

- Simple random sampling from a target population
- Adequate sample size

[1] More on this on Poisson regression's lecture.

Simulations for the impact of priors in data analysis

- Consider several data scenarios, each time assuming that the true parameter values are a = -2 and b = 0.8 and that the values of x are drawn from a uniform distribution between -1 and 1.
- To repeat the same analyses (Bayesian & frequentist) with different sample sizes, we write a function.
- `bayes_sim()` enables the analysis to be sequentially performed as a both standard (maximum likelihood, `glm`) and Bayesian (`stan_glm`) logistic regression with varying sample sizes.

```r
library("arm", "rstanarm")
set.seed(1234)
bayes_sim <- function(n, a = -2, b = 0.8) {
  data <- tibble(x = runif(n, min = -1, max = 1),
      y = if_else(0 < rlogis(n, location = a + b * x, scale = 1), 1, 0))
  fit_glm <- glm(y ~ x, family = binomial(link = "logit"), data = data)
  fit_stan <- stan_glm(y ~ x, family = binomial(link = "logit"),
      data = data, refresh = 0,
      prior = normal(location = 0.5, scale = 0.5)) #<
  arm::display(fit_glm, digits = 1)
  cat("\n")
  print(fit_stan, digits = 1)
}
```

Simulation study n = 10

Focus on inference about b, which was assigned a value = 0.8 when generating the data

```
set.seed(1234); bayes_sim(n=10) #small sample size of only 10 obervations
```

```
## glm(formula = y ~ x, family = binomial(link = "logit"), data = data)
##             coef.est coef.se
## (Intercept) -2.4      1.2   
## x            1.4      2.8   
## ---
##   n = 10, k = 2
##   residual deviance = 6.2, null deviance = 6.5 (difference = 0.3)
## 
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 10
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -2.1    0.9  
## x            0.5    0.5  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Simulation study n = 10 interpretation

With only 10 observations, the maximum likelihood estimate is noisy,and in this simulation, `glm` gives a maximum likelihood estimate of 1.4, with a large se = 2.8, confirming that the likelihood provides little precision (information)

As expected with little data the Bayesian posterior will be influenced the prior.

- Inference from `stan_glm` relies heavily on the prior distribution: the Bayes estimate of the coefficient = 0.6 is close to the prior mean of 0.5, being pulled away by the data only slightly.

```
## Priors for model 'fit_stan'
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 0, scale = 2.5)
##
## Coefficients
##  ~ normal(location = 0.5, scale = 0.5)
## ------
## See help('prior_summary.stanreg') for more details
```

Simulation study n = 100

```r
set.seed(1234); bayes_sim(n=100)
```

```
## glm(formula = y ~ x, family = binomial(link = "logit"), data = data)
##             coef.est coef.se
## (Intercept) -1.7      0.3   
## x            0.1      0.5   
## ---
##   n = 100, k = 2
##   residual deviance = 84.5, null deviance = 84.5 (difference = 0.0)
## 
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -1.7    0.3  
## x            0.3    0.4  
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Simulation study n = 100 interpretation

With 100 observations, the maximum likelihood estimate has now excluded more extreme values and provides a more precise estimate, and in this simulation, `glm` gives a MLE = 0.1, with a smaller se = 0.5 (was 2.4), confirming that the likelihood (data) provides modest precision (information) and the CI includes the true paramter value (0.8)

As expected with more data the Bayesian posterior will be less influenced the prior.

- Nevertheless, the inference from `stan_glm`, parameter = 0.3 has still seen the data (0.1) pulled towards the prior (0.5) but less than with the previous smaller sample size

Simulation study n = 1000

```
set.seed(1234); bayes_sim(n=1000)
```

```
## glm(formula = y ~ x, family = binomial(link = "logit"), data = data)
##             coef.est coef.se
## (Intercept) -2.3      0.1   
## x            0.9      0.2   
## ---
##   n = 1000, k = 2
##   residual deviance = 639.3, null deviance = 663.3 (difference = 23.9)
##
## stan_glm
##  family:       binomial [logit]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -2.3    0.1  
## x            0.9    0.2  
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Simulation study n = 1000 interpretation

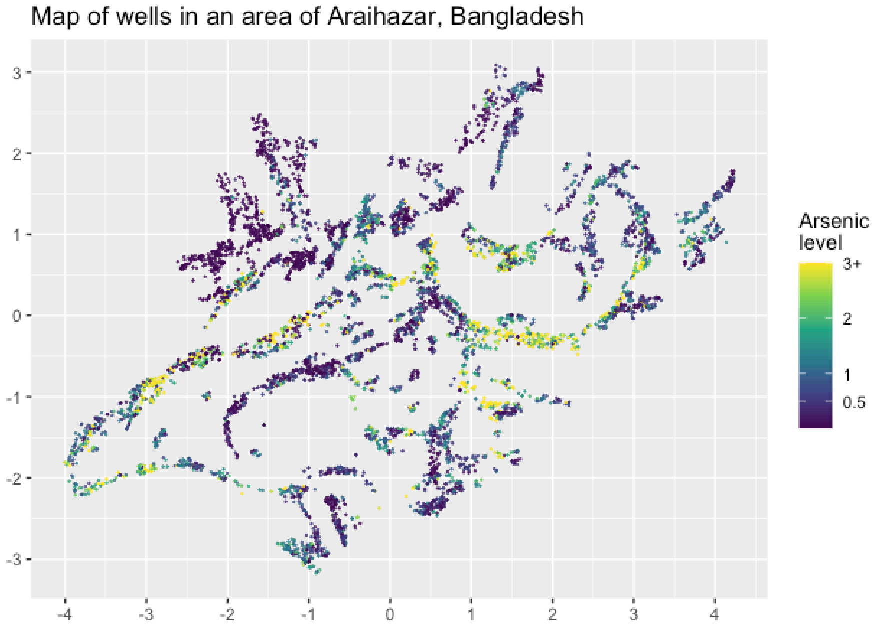
With 1000 observations, the maximum likelihood estimate now provides an accurate and precise estimate (0.9, se = 0.2) of the known parameter, $\beta_1 = 0.8$

The Bayes estimate is now also dominated by the data with an almost negligible effect of the prior.
Once $n $is as large as 1000, a weak or even a modest prior distribution doesn't really make a difference and the two approaches produce essentially identical results.

Building a Bayesian logistic regression model - A public health example

Wells in Bangladesh

Example from **Regression and other Stories - Chapters 13-14**



Background:

Research teams from the US and Bangladesh measured all the wells and labeled them with their arsenic level as well as a characterization as “safe” (<0.5 micrograms per liter)

People with unsafe wells were encouraged to switch to nearby private or community wells or to new wells of their own construction. A few years later, the researchers returned to find out who had switched wells. We shall perform a logistic regression analysis to understand the factors **predictive** of well switching among the users of unsafe wells.

Variables: Outcome: y_i = 1 if household i switched or = 0 if household i continued using its own well.
Potential independent (predictor) variables are

- distance (in meters) to the closest known safe well
- arsenic level of respondent’s well
- any household members active in community organizations
- education level of the head of household

We shall first fit the model just using distance to the nearest well and then put in arsenic concentration, organizational membership, and education.

Read in the data

```r
# Data on arsenic in unsafe wells in Bangladesh
# remotes::install_github("avehtari/ROS-Examples",subdir = "rpackage")
library(rosdata)
data(wells)
file_common <- here::here("_common.R")  # Specific formating and functions
source(file_common) # Run common code
str(wells)
```

```
## 'data.frame':    3020 obs. of  7 variables:
##  $ switch : int  1 1 0 1 1 1 1 1 1 1 ...
##  $ arsenic: num  2.36 0.71 2.07 1.15 1.1 3.9 2.97 3.24 3.28 2.52 ...
##  $ dist   : num  16.8 47.3 21 21.5 40.9 ...
##  $ dist100: num  0.168 0.473 0.21 0.215 0.409 ...
##  $ assoc  : int  0 0 0 0 1 1 1 0 1 1 ...
##  $ educ   : int  0 0 10 12 14 9 4 10 0 0 ...
##  $ educ4  : num  0 0 2.5 3 3.5 2.25 1 2.5 0 0 ...
```

Data overview

```
#summary(wells) # details can found with ?wells
wells %>%  tbl_summary()
```

Characteristic	N = 3,020[1]
switch	1,737 (58%)
arsenic	1.30 (0.82, 2.20)
dist	37 (21, 64)
dist100	0.37 (0.21, 0.64)
assoc	1,277 (42%)
educ	5 (0, 8)
educ4	1.25 (0.00, 2.00)

[1] n (%); Median (Q1, Q3)

Data overview

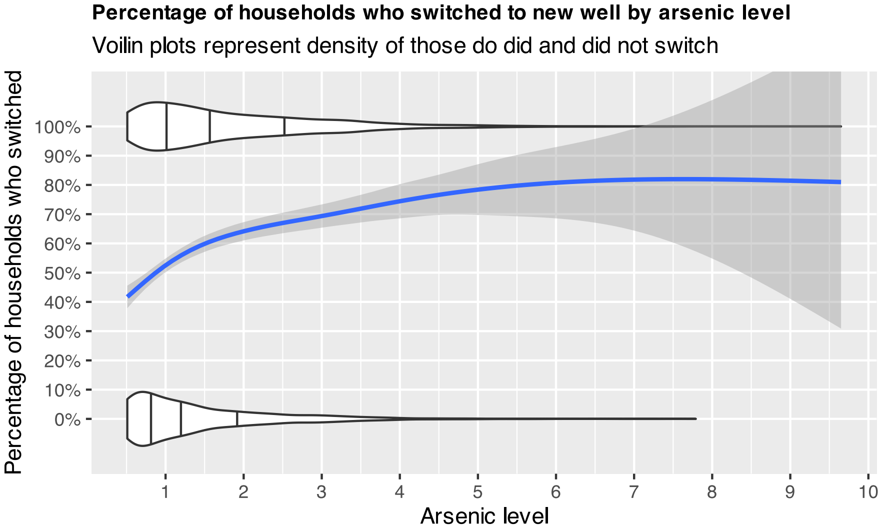
How many homes with unsafe wells switched?

```
wells %>% count(switch) %>%
  mutate(prop = n / sum(n))
```

```
#>   switch    n  prop
#> 1      0 1283 0.425
#> 2      1 1737 0.575
```

visualization

A helpful visualization



As expected, the % of households increases with the arsenic level in their well, from about 40% for wells that are just over the safety threshold to perhaps 80% for very high levels. The sparse data for high arsenic levels results in a large uncertainty.
What is the blue line?

- The blue line is a non-parametric approach to draw a smooth curve through a scatter plot, known as LOWESS (Locally Weighted Scatterplot Smoothing),or sometimes called LOESS (locally weighted smoothing)

Logistic regression with just one predictor

Fit the logistic regression

```
fit_0 <- stan_glm(switch ~ dist, family=binomial(link="logit"), refresh=0, data=wells, seed=
print(fit_0, digits = 4)
```

```
#> stan_glm
#>  family:       binomial [logit]
#>  formula:      switch ~ dist
#>  observations: 3020
#>  predictors:   2
#> ------
#>             Median  MAD_SD
#> (Intercept)  0.6073  0.0599
#> dist        -0.0062  0.0009
#>
#> ------
#> * For help interpreting the printed output see ?print.stanreg
#> * For info on the priors used see ?prior_summary.stanreg
```

What happens to the β coefficient if we change distance in meters to 100 meter units?

Logistic regression with just one predictor

Fit the logistic regression

```
wells$dist100 <- wells$dist/100  #change distance from meters to 100 meter units
fit_1 <- stan_glm(switch ~ dist100, family=binomial(link="logit"), refresh=0, data=wells, se
print(fit_1, digits = 4)
```

```
#> stan_glm
#>  family:       binomial [logit]
#>  formula:      switch ~ dist100
#>  observations: 3020
#>  predictors:   2
#> ------
#>             Median  MAD_SD
#> (Intercept)  0.6073  0.0599
#> dist100     -0.6234  0.0946
#>
#> ------
#> * For help interpreting the printed output see ?print.stanreg
#> * For info on the priors used see ?prior_summary.stanreg
```

If divide (or multiply) units by X then coefficient for X is multiplied (or divided) by X

Interpreting coefficients - Three different scales

We can interpret the coefficient estimates on 3 different scales

$$log\,odds(switch) = logit\frac{p}{1-p} = 0.61 - 0.62 * dist100$$

$$odds = exp^{logit} = exp^{0.61-0.62*dist100}$$

$$Pr\,(switch) = logit^{-1}(0.61 - 0.62 * dist100) = \frac{1}{1 + exp^{-(0.61-0.62*dist100)}}$$

Interpreting the logistic regression intercept

$$log\, odds(switch) = logit \frac{p}{1-p} = 0.61 - 0.62 * dist100$$

The constant term (intercept) is value when `dist100` = 0

i) log odds (switching) = 0.61

ii) odds (switching) = $e^{0.61}$ = 1.84

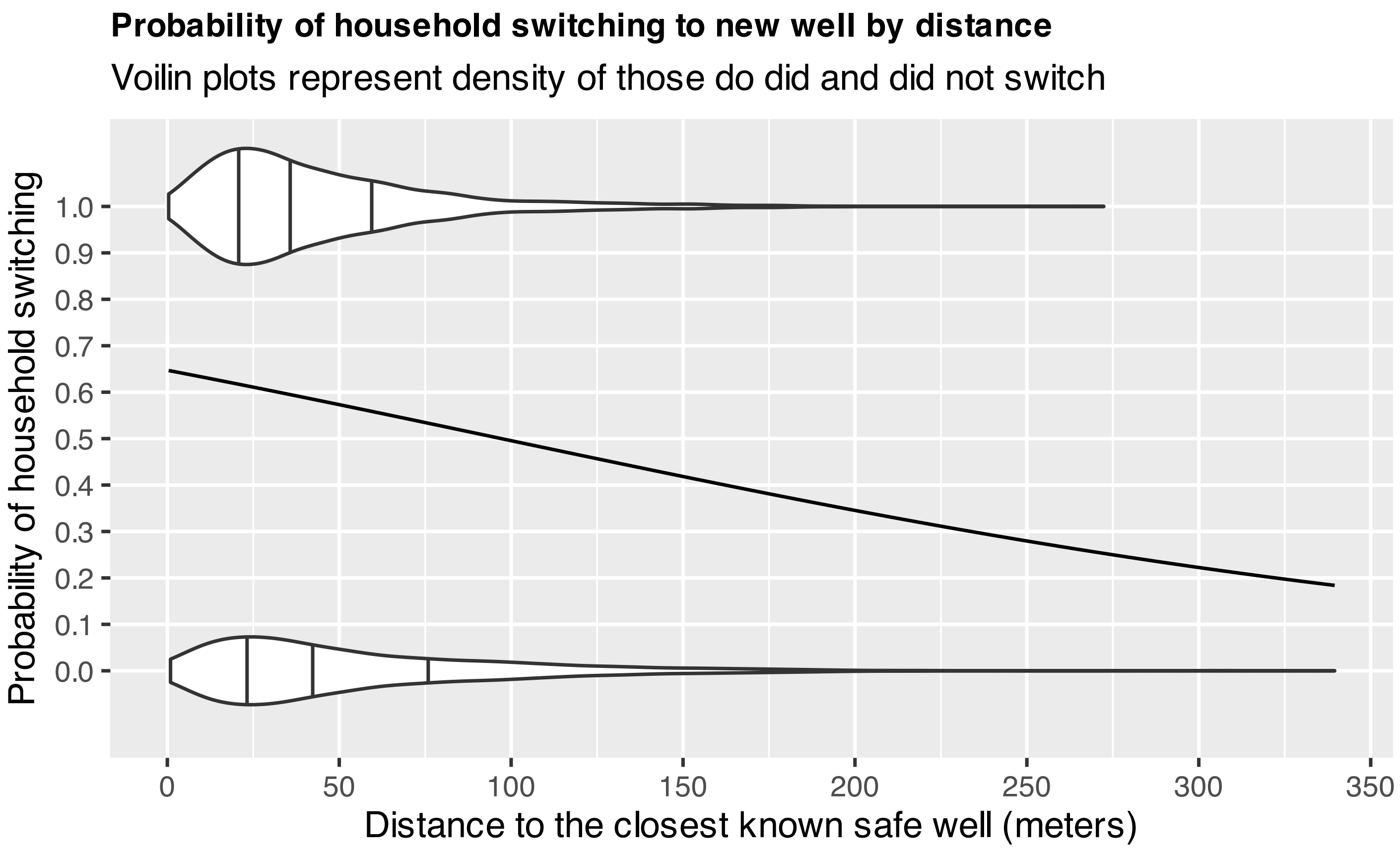
iii) P(switching) $p = \dfrac{odds}{odds+1} = \dfrac{1.84}{2.84} = .65$, by rearranging $odds = \dfrac{p}{1-p}$

More directly, constant term = probability(switching) when `dist100` = 0

$logit^{-1}(0.61) = \dfrac{1}{1+e^{-0.61}} = 0.65$ or with `R` code `invlogit(0.61) = 0.65`

Thus, model estimates a 65% probability of switching if live right next to an existing safe well

Logistic regression with just one predictor



The probability of switching is about 65% `invlogit(.61)` for people who live near a safe well, declining to about 20% for people who live more than 300 meters from any safe well. This makes sense: the probability of switching is higher for people who live closer to a safe well.

Interpreting the β_1 regression coefficient

Remember this interpretation can also be made on 3 different scales

$$log\,odds(switch) = logit\frac{p}{1-p} = 0.61 - 0.62 * dist100$$

i) Change in log odds for unit change in `dist100` = -0.62

ii) Δ odds = `exp(-0.62)` = 0.54 (< likely to switch for each 100m increase in distance)

iii) Since probability scale is nonlinear, $\frac{1}{1+e^{-0.61-0.62*dist100}}$, must choose where to evaluate the effect of a 1 unit change in the `dist100` variable

Interpreting the β_1 regression coefficient

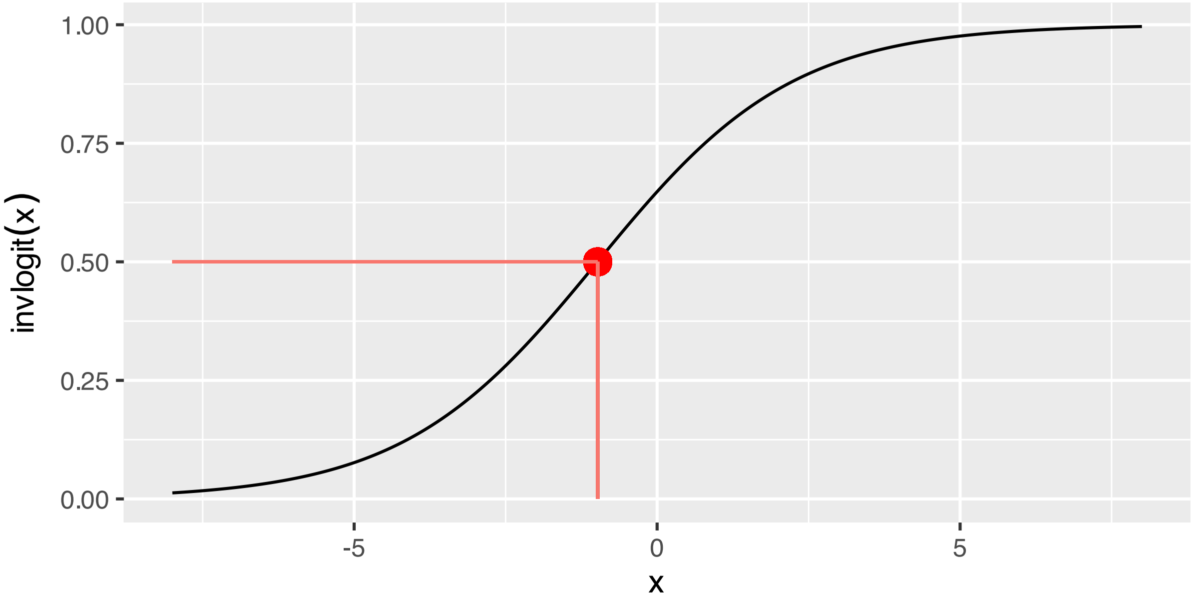
- Often choose predictor mean (steepest part of logistic curve) `mean(wells$dist100)` = 0.48
- Linear predictor for logit function = 0.61 - 0.62 $*$ 0.48 = 0.31
- P(switching) = `invlogit(.31)` = 0.57 (reverts back to (0,1) probability scale)
- Logit linear predictor for a 1 unit increment from the mean value = 0.61 - 0.62 $*$ 1.48 = -0.31
- P(switching) = `invlogit(-0.31)` = 0.42

Thus, adding 100 meters to the distance to the nearest safe well (from the mean distance), decreases the probability of switching by about 15% (57%-42%).

Interpreting the logistic regression intercept

$$log\,odds(switch) = logit\frac{p}{1-p} = 0.61 - 0.62 * dist100$$

invlogit (0.61 - 0.62*x) = invlogit(0) = 0.5, when x = - 0.61/0.62 = -0.98



Interpreting the β_1 regression coefficient

Divide by 4 rule The slope (1st derivative) of the inverse logistic function = $\frac{d(1/(1+exp(x)^{-1})}{dx}$

$$\frac{e^x}{(e^x+1)^2}$$

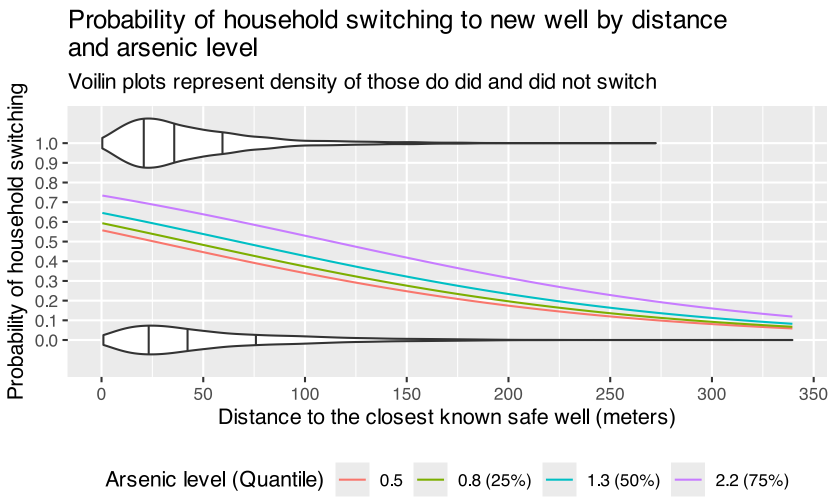
(if you forgot how to take derivatives use `D(expression(1/(1+exp(x)^-1)), "x")`
The logistic curve is steepest at its center, which occurs when the linear predictors $\alpha + \beta x$ = 0 so that `invlogit(0)` = 0.5. Substituting x=0 into the 1st derivative equation maximizes the slope and equals

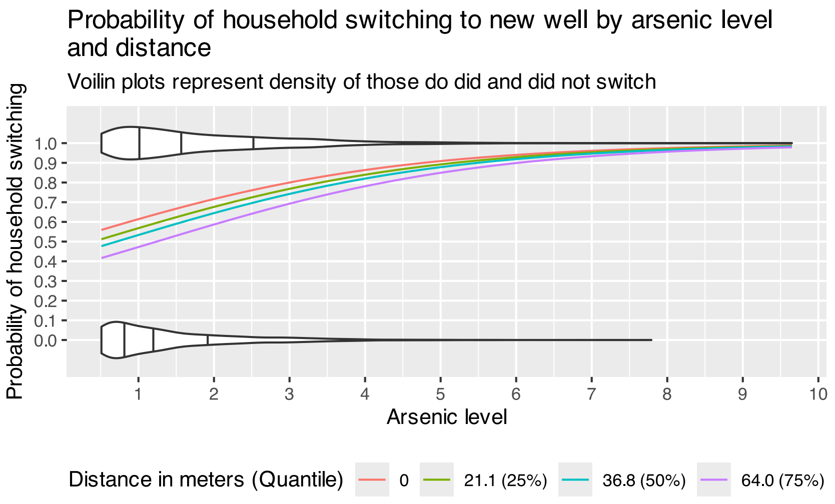
$$\beta \frac{e^0}{(e^0+1)^2} = \frac{\beta}{4}$$

Thus, $\frac{\beta}{4} = -0.62/4 = -0.15$ is the maximum Δ in Pr(y = 1) corresponding to a unit difference in x.

Graphing the fitted model with two predictors

Probability switching to new well by distance and arsenic level

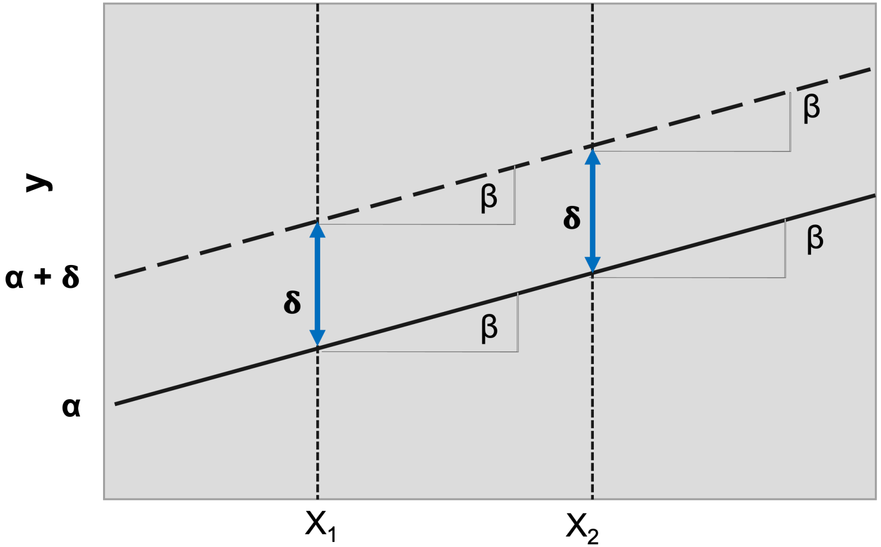




Recall: Linear Regression Model

Review: *“The effect of a given change in an independent variable is the same regardless the value of that variable at the start of it changes and regardless of the level of the other variables in the model.”*

$$y = \alpha + \beta x + \boldsymbol{\delta} d$$



Adapted from: Long, J., & Freese, J. (2006).Regression models for categorical dependent variables using stata(Second ed.). College Station, Texas: StataCorp LP.

Linear Probability Model for RDs when the outcome is binary:

Observations	3020
Dependent variable	switch
Type	OLS linear regression

F(1,3018)	42.57
R^2	0.01
Adj. R^2	0.01

	Est.	**2.5%**	**97.5%**	**t val.**	**p**
(Intercept)	0.65	0.62	0.68	45.19	0.00
dist100	-0.15	-0.20	-0.11	-6.52	0.00
Standard errors: OLS					

Observations	3020
Dependent variable	switch
Type	OLS linear regression

F(1,3018)	3.90
R^2	0.00
Adj. R^2	0.00

	Est.	**2.5%**	**97.5%**	**t val.**	**p**
(Intercept)	0.59	0.57	0.61	49.88	0.00
assoc	-0.04	-0.07	-0.00	-1.97	0.05
Standard errors: OLS					

Remember the divide by 4 rule? $\frac{\beta}{4} = -0.62/4 = -0.15$ is the maximum difference in Pr(y = 1) corresponding to a unit difference in x. **Would it work here too?**

(I) Linear Probability Model for RDs when the outcome is binary:

Advantages:

- Very easy to fit
- Single uniform estimate
- *Economists will love you*

Disadvantages:

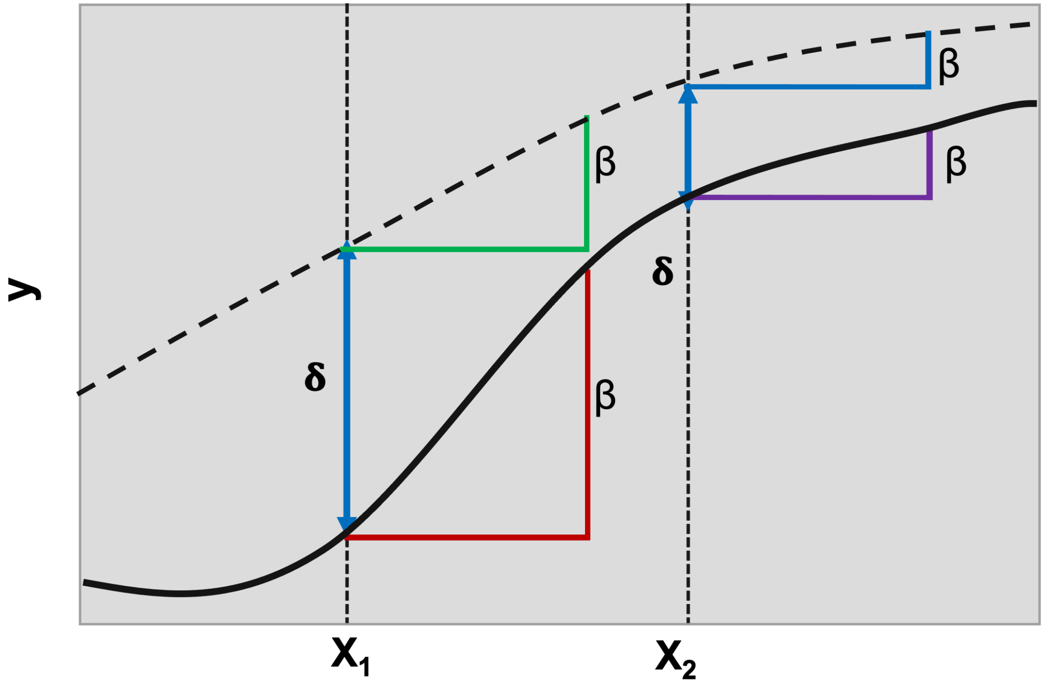
- Possible to get *impossible* estimates
- Biostatisticians will hate you

Fit an OLS linear regression on the binary outcome variable: $Pr(Y = 1|X = x) = \beta 0 + \beta 1X$

Note: Homoskedasticity assumption cannot be met, since variance is a function of p. Therefore, use robust variance.

Multiplicative Models

Review: *"The effect of a change in a variable* ***depends*** *on the values of all variables in the model and it's no longer simply equal to one of the parameters in the model"*



Adapted from: Long, J., & Freese, J. (2006).Regression models for categorical dependent variables using Stata (Second ed.). College Station, Texas: StataCorp LP.

Multiplicative Models - Review

- Binary outcomes
- Expressed on a transformed scale, it prescribes a linear relationship between the log-odds of Y and X.

$$P(X=1)=\frac{exp(\alpha+\beta_x)}{1+exp(\alpha+\beta_x)}$$

- The log odds of the outcome is linearly related to x, with intercept coefficient α and slope coefficient β
    - i.e., the logistic model is an **additive model** when expressed on the log odds scale.

Logistic Regression: Log Odds vs Probabilities

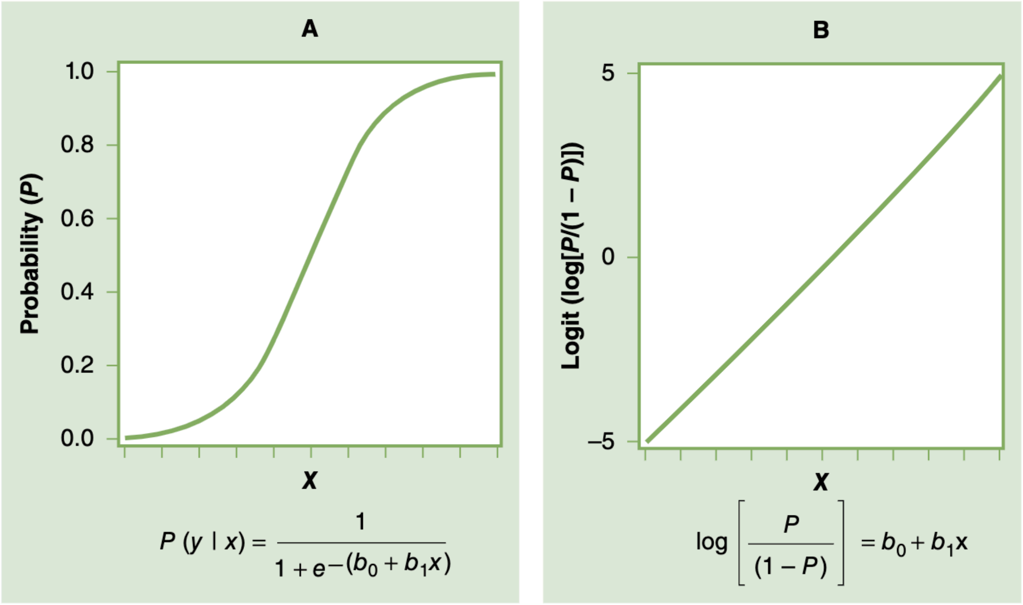


FIGURE 7-7 Two mathematically equivalent formulations of the logistic regression function.

(II) Logistic Models for RDs and RRs

Disadvantages:

- Does not give a **single** uniform estimate
- Choose between different formulations

Advantages:

- Always fits easily
- Can never get impossible estimates
- Epidemiologists will love you

Logistic Models for RDs and RRs

Several approaches:

- Transformation of ORs, estimation of RRs/RDs with different models

Simple as:

(1) Fit a logistic regression

(2) Predict probabilities based on the regression parameters (several options to do this!)

(3) Use these probabilities to calculate risk ratios/risk differences

(1) Fit a standard logistic regression model:

The Frequentist way

```r
mod2a <- glm(switch ~ assoc, data=wells, family = "binomial")
round(summ(mod2a, confint = T)$"coeftable", 2) #;summ(mod2a)
```

```
#>               Est.  2.5% 97.5% z val.    p
#> (Intercept)  0.37  0.27  0.46   7.50 0.00
#> assoc       -0.15 -0.29  0.00  -1.97 0.05
```

The Bayesian way:

```r
mod2b <- stan_glm(switch ~ assoc, family=binomial(link="logit"), data=wells, refresh=0)
round(mod2b$stan_summary[1:2,1:5-10], 2) #; print(mod2b, digits = 2) #; summary(mod2b)
```

```
#>              mean se_mean   sd  2.5% 97.5% n_eff Rhat
#> (Intercept)  0.36       0 0.05  0.27  0.45  3076    1
#> assoc       -0.15       0 0.07 -0.29  0.00  3013    1
```

Next: obtain the predicted probabilities

(2) Estimate the probabilities based on/using the regression parameters and the observed data (i)

```r
p1<- predict(mod2a, newdata = transform(wells, assoc=1), type="response")
summary(p1) # probabilities among "Exposed"
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
#>   0.554   0.554   0.554   0.554   0.554   0.554 
```

```r
p0<- predict(mod2a, newdata = transform(wells, assoc=0), type="response")
summary(p0) # probabilities among "Unexposed"
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
#>    0.59    0.59    0.59    0.59    0.59    0.59 
```

Next: contrast the predicted probabilities

(3) Use these probabilities to calculate average "predictive" risk ratios/risk and differences

```r
RR<- p1/p0 # Calculate the ratio contrast
#summary(RR)
round(mean(RR), 2)
```

```
#> [1] 0.94
```

```r
RD<- p1-p0 # Calculate the difference contrast
#summary(RD)
round(mean(RD),2)
```

```
#> [1] -0.04
```

Then we just obtained our RDs and RRs...

(2) Estimate the probabilities based on/using the regression parameters and the observed data (ii)

Inverse logit **option:**

The syntax is simple, using the "linear combination" for each group "Exposed" and "Unexposed" the functions are `inv.logit`, `invlogit` or any other `function` that estimates the inverse logit

$$logit^{-1}(x) = \frac{1}{1+e^{x}}$$

RR <- inv.logit(Linear Combination for Exposed) / inv.logit(Linear Combination Unexposed)

RD <- inv.logit(Linear Combination for Exposed) — inv.logit(Linear Combination Unexposed)

Using the `inv.logit` function: **The Frequentist way**

Risk Ratio

```r
#Using the inv.logit
rr.i<-inv.logit(mod2a$coefficients["(Intercept)"] + mod2a$coefficients["assoc"]) /
  inv.logit(mod2a$coefficients["(Intercept)"] )
round(rr.i, 2)
```

```
#> (Intercept)
#>        0.94
```

Risk Difference

```r
rd.i<-inv.logit(mod2a$coefficients["(Intercept)"] + mod2a$coefficients["assoc"]) -
  inv.logit(mod2a$coefficients["(Intercept)"] )
round(rd.i, 2)
```

```
#> (Intercept)
#>       -0.04
```

Contrasting the predicted probabilities in a single step

Using the `invlogit` function: **The Bayesian way**

```
beta <- coef(mod2b) # to extract the beta coefficients from the model
beta
```

```
#> (Intercept)       assoc
#>       0.365      -0.145
```

```
yes <- 1 #to assign exposed, assoc= 1
no <- 0 #to assign unexposed, assoc= 0
```

Risk Ratio

```
ratio<- invlogit (beta[1] + beta[2]*yes) /
         invlogit (beta[1] + beta[2]*no)
round(mean(ratio), 2)
```

```
#> [1] 0.94
```

Risk Difference

```
diff <- invlogit (beta[1] + beta[2]yes) -
         invlogit (beta[1] + beta[2]no)
round(mean(diff), 2)
```

```
#> [1] -0.04
```

Contrasting the predicted probabilities in a single step!

How do I interpret these??

Average **predictive** ratios and differences effects

Average **conditional** ratios and differences effects

Average Marginal Effects

Where/how you fix your covariates has an impact on your estimated probabilities (and your population of inference). Muller and MacLehose present us three strategies:

- Prediction at the modes : conditional predicted probabilities are calculated for each exposure level with every covariate/confounder fixed at its most common value
- Prediction at the means : conditional predicted probabilities are calculated for each exposure level with every covariate/confounder fixed at its mean value
- Marginal standardization : predicted probabilities of the outcome are calculated for every observed confounder value and then combined as a weighted average separately for each exposure level

Estimating predicted probabilities from logistic regression: different methods correspond to different target populations, by Clemma J Muller & Richard F MacLehose

Margins at the Modes

Assumes everyone had most common values of the confounders:

$$Pr(Y = 1|Set[E = e], Z = zm)$$

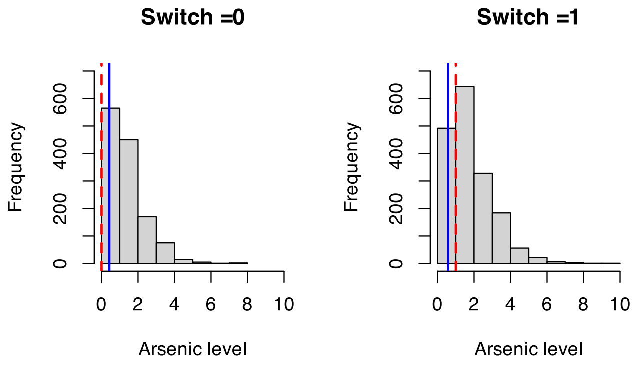
- In the case of a single dichotomous confounder, we'd calculate predicted probabilities of the outcome only for the most frequently observed value of the confounder (0 or 1) in the population
- As the number of confounders increases, the number of observations in our cells will decrease (all are set to their modal value)
- Could also predict at most common *JOINT* covariate pattern, but you're still standardizing estimates to the population of those w/the modal distribution of Z
- Might be okay if the modal population is of interest, but likely misleading if you want to say something about the full population

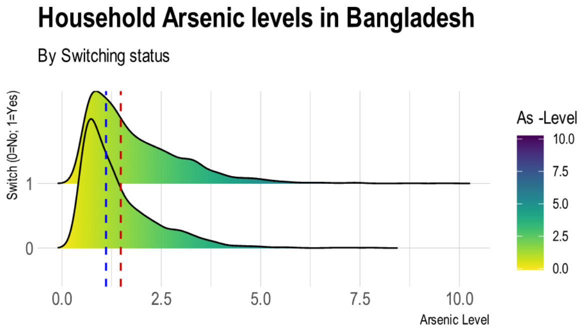
Let's go back to our example of Wells in Bangladesh

Characteristic	N = 3,020[1]
switch	1,737 (58%)
arsenic	
Median (Q1, Q3)	1.30 (0.82, 2.20)
Min, Max	0.51, 9.65
dist100	
Median (Q1, Q3)	0.37 (0.21, 0.64)
Min, Max	0.00, 3.40
assoc	1,277 (42%)

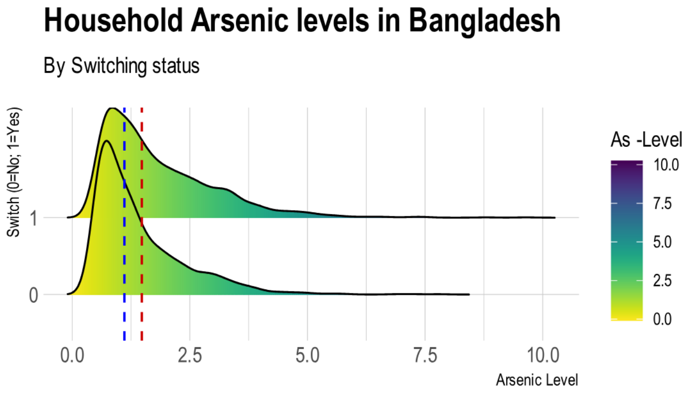
[1] n (%)

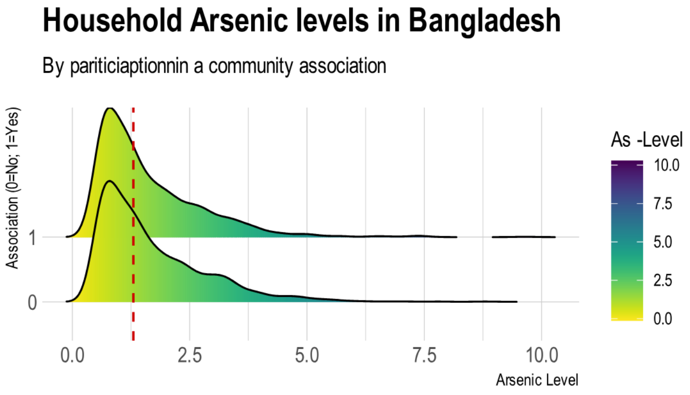
Arsenic Levels in the Wells





Let's model the "switch" outcome as a function of the As level and whether the owner belongs to a community association.





Let's model the "switch" outcome as a function of the As. level and whether the owner belongs to a community association.

Frequentist

```r
mod3a <- glm(switch ~ assoc + arsenic, data=wells, family = "binomial")
round(summ(mod3a, confint = T)$"coeftable", 2) #;summ(mod3a)
```

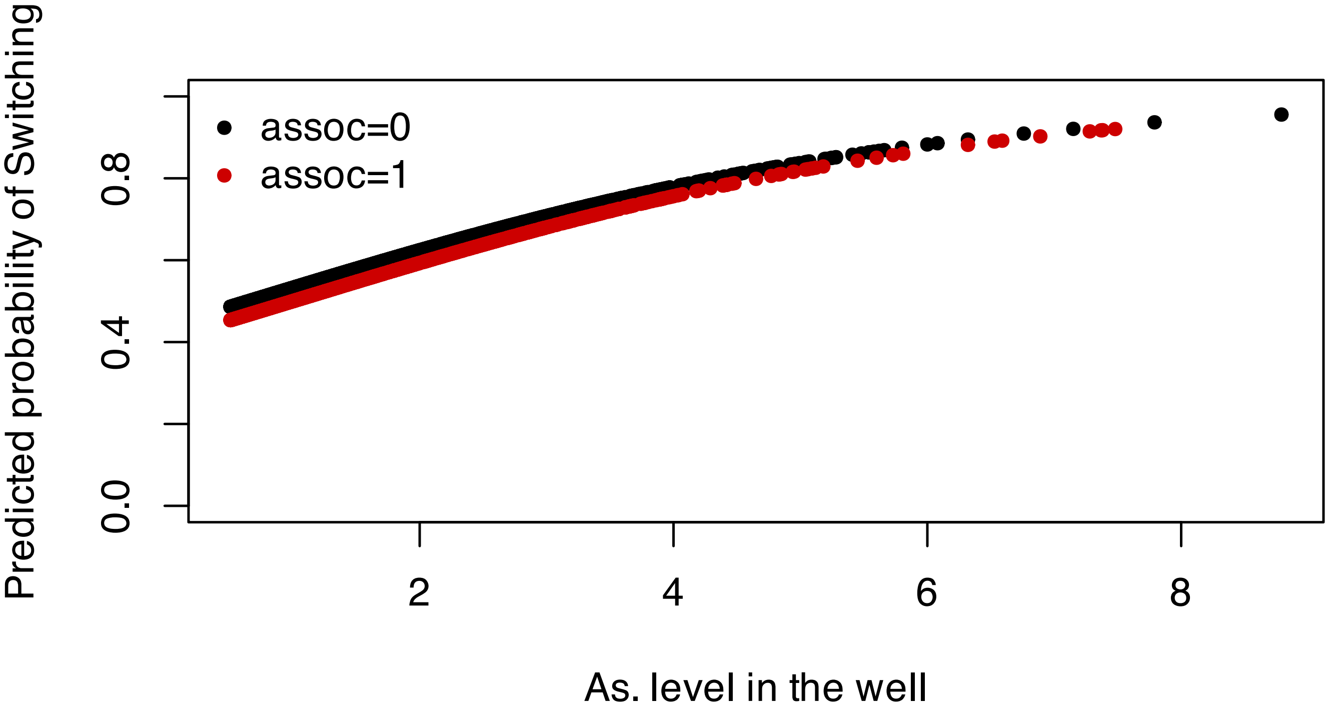
```
#>              Est.  2.5% 97.5% z val.    p
#> (Intercept) -0.25 -0.40 -0.10  -3.18 0.00
#> assoc       -0.13 -0.28  0.02  -1.72 0.09
#> arsenic      0.38  0.30  0.45   9.80 0.00
```

Bayesian

```r
mod3b <- stan_glm(switch ~ assoc + arsenic, family=binomial(link="logit"),
                  data=wells, refresh=0)
round(mod3b$stan_summary[1:3,1:5-10], 2) #; print(mod3b, digits = 2) #; summary(mod3b)
```

```
#>              mean se_mean   sd  2.5% 97.5% n_eff Rhat
#> (Intercept) -0.25       0 0.08 -0.41 -0.10  4109    1
#> assoc       -0.13       0 0.08 -0.28  0.01  3997    1
#> arsenic      0.38       0 0.04  0.31  0.45  3958    1
```

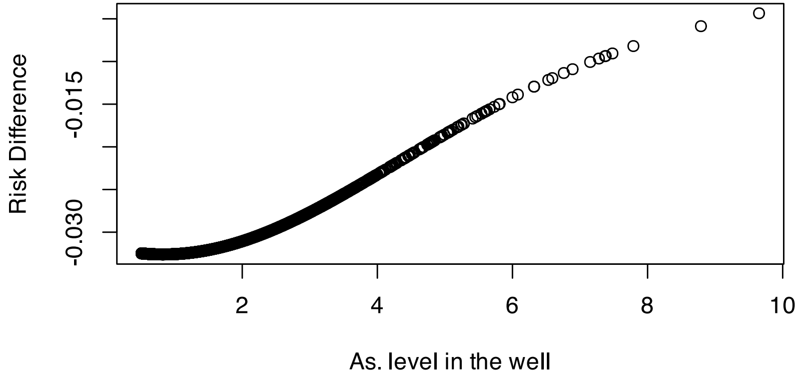
Predicted probabilities of "switch" to a safer well by whether the owner is member of a community organization througout the observed range of As level in the household wells.



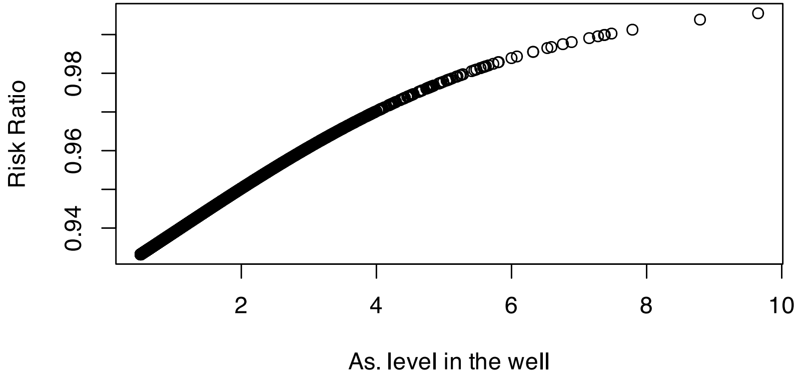
What would be the RD and RR?

Estimated RR and RD using the probabilities from the logistic model

Risk Difference



Risk Ratio



```r
wells$pp0 <- predict(mod3a, newdata = transform(wells, assoc=0), type="response")
wells$pp1 <- predict(mod3a, newdata = transform(wells, assoc=1), type="response")
```

Risk Difference

```r
wells$pp.rd <- wells$pp1 - wells$pp0
plot(wells$arsenic, wells$pp.rd, ylab="Risk
```

Risk Ratio

```r
wells$pp.rr <- wells$pp1 / wells$pp0
plot(wells$arsenic, wells$pp.rr, ylab="Risk
```

Using the `inv.logit` function to obtain the **"overall" RR and RD**

```r
#Using the inv.logit
rr.i2<-inv.logit(mod3a$coefficients["(Intercept)"] +
                   mod3a$coefficients["arsenic"] + mod3a$coefficients["assoc"]) /
  inv.logit(mod3a$coefficients["(Intercept)"]+ mod3a$coefficients["arsenic"] )
round(rr.i2, 2)
```

```
#> (Intercept)
#>        0.94
```

```r
rd.i2<-inv.logit(mod3a$coefficients["(Intercept)"] +
                   mod3a$coefficients["arsenic"]+  mod3a$coefficients["assoc"]) -
  inv.logit(mod3a$coefficients["(Intercept)"]+ mod3a$coefficients["arsenic"] )
round(rd.i2, 2)
```

```
#> (Intercept)
#>       -0.03
```

Margins at the Means

Assumes everyone in the dataset has the mean value of the confounder(s)

$$Pr(Y = 1|Set[E = e], Z = z)$$

- If a confounder is continuous/ordinal, the mean might not reflect any particular individual in the data

- If a confounder is binary, the mean reflects the proportion of subjects with that confounder, *but the mean will (probably) not correspond to any observations in the data* - predictions will not be true for any individual subject

- Sometimes wrongly interpreted as the average/marginal probability in the population

Margins at the Means

As per Muller (2014):

"In the presence of binary covariates, prediction at the means yields results that are not meaningful to any real-world group of individuals."

- It estimates *associations at the mean of each confounder* in the regression model
- No one can be half hypertensive or 10% diabetic
- *When calculating predicted probabilities, the inverse logit of the averages (prediction at the means) is not equal to the average of the inverse logits (marginal standardization)*
- This is why these approaches differ, and it's why they can diverge significantly in certain situations

Estimating predicted probabilities from logistic regression: different methods correspond to different target populations, by Clemma J Muller & Richard F MacLehose

Margins at the Means

R does not have the *‘atmeans’* option (because it is *often* Meaningless!)

```
mean(wells$arsenic); mean(wells$assoc)
```

```
#> [1] 1.66

#> [1] 0.423
```

```
mar.means<-margins(mod3a,
          at = list(assoc=mean(wells$assoc)
       type = "response")
mar.means
```

```
#>   at(assoc)    assoc arsenic
#>      0.4228 -0.03079 0.08913
```

Similarly:

```
margins(mod3a, at = list(assoc= mean(wells$
        arsenic=mean(wells$arsenic)),
        type = "response")
```

```
#>   at(assoc) at(arsenic)    assoc arsenic
#>      0.4228       1.657 -0.03178   0.092
```

Marginal Standardization

- Prediction at the means involves hypothetical people (with likely impossible covariate patterns), whereas marginal standardization involves hypothetical populations

- Under this approach, you’re comparing two hypothetical populations that are identical except for exposure status

- Since the only difference between these two populations is their **exposure status**, we can attribute any differences in the probability of the outcome to the exposure

- The resulting estimates are essentially weighted to the distribution of confounders in the sample = **Average Marginal Effect (AME)**

Marginal Standardization

Basic steps for the AME (assuming a binary exposure):

- Start at subject #1. Treat that person as though they were exposed (i.e., change exposure to “1”), but don’t change anything else. Compute this subject’s probability of the outcome.

- Now switch the same subject’s exposure status to “0” (unexposed) and repeat the probability calculation.

- Take the individual-level difference in the two probabilities
    - This is the marginal effect for that subject
    - Repeat the process for every subject in the sample
    - Compute the average of all the subject-specific marginal effects.
- This gives you the **AME of your exposure!**.

Marginal Standardization

$Pr(Y)$ will be determined by confounder (Z) pattern:

Assuming a binary exposure, the marginal RR/RD are simply:

$RR = Pr(Y = 1|Set[E = 1])/Pr(Y = 1|Set[E = 0])$

$RD = Pr(Y = 1|Set[E = 1])– Pr(Y = 0|Set[E = 0])$

(Z drops out of these equations as we’ve predicted probabilities under the same distribution of Z for both groups)

Sounds familiar? … We have a name for this…

Marginal Standardization

The steps:

- Generate one new variable representing the predicted risk of the outcome if everyone had been ***exposed***
- Generate another new variable representing the predicted risk if everyone had been ***unexposed***

This will often give you a different quantity than you'd get if you ran the usual postestimation commands (i.e., run the logistic, predict the probabilities, and calculate the ratio/difference)

This is marginal standardization!

Marginal Standardization (i)

```r
#to obtain the conditional probability
wells$predprob<-predict(mod3a, wells=transform(wells), type="response")
#to change the exposure statures with the predicted probabilities
wells$unex<- wells$predprob
wells$unex[wells$assoc==1]<- inv.logit(mod3a$coefficients["(Intercept)"]+
                            (mod3a$coefficients["arsenic"]*wells$arsenic))
wells$exp<-wells$predprob
wells$exp[wells$assoc==0]<- inv.logit(mod3a$coefficients["(Intercept)"] +
                           (mod3a$coefficients["arsenic"]*wells$arsenic) +
                            (mod3a$coefficients["assoc"]*wells$assoc))
#summary(wells$exp); summary(wells$unex)
```

RD

```r
mar.RD <-mean(wells$exp) - (mean(wells$unex
wells$m.diff <- (wells$exp-wells$unex)
mar.RD
```

```
#> [1] -0.0208
```

```r
round(summary(wells$m.diff), 2)
```

```
#>     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>    -0.46   -0.10   -0.02   -0.02    0.06    0.43
```

RR

```r
mar.RR <-mean(wells$exp) /(mean(wells$unex)
wells$m.rr <- (wells$exp/wells$unex)
mar.RR
```

```
#> [1] 0.965
```

```r
round(summary(wells$m.rr), 2)
```

```
#>     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>     0.50    0.84    0.96    0.98    1.10    1.88
```

Marginal Standardization (ii)

Using Standardization `stdReg` package

```
#install.packages("stdReg")
require(stdReg)
condprob<- stdGlm(fit=mod3a,data = wells, X="assoc", x=c(0:1)); summary(condprob)$est.table
```

```
#>   Estimate Std. Error lower 0.95 upper 0.95
#> 0    0.588     0.0117      0.565      0.611
#> 1    0.557     0.0138      0.530      0.584
```

```
summary(condprob, contrast="difference", reference=0)$est.table #std RD
```

```
#>   Estimate Std. Error lower 0.95 upper 0.95
#> 0   0.0000     0.0000      0.000    0.00000
#> 1  -0.0308     0.0179     -0.066    0.00434
```

```
summary(condprob, contrast="ratio", reference=0)$est.table #std RR; #summary(condprob, trans
```

```
#>   Estimate Std. Error lower 0.95 upper 0.95
#> 0    1.000     0.0000      1.000       1.00
#> 1    0.948     0.0298      0.889       1.01
```

Nice walk through here: SjölanderA. Regression standardization with the R package stdReg. European journal of epidemiology. 2016 May 14:1-2.

Marginal Standardization (ii)

Using `Margins` package

```
require(margins)
margins_summary(mod3a) #; margins(mod3a, at = list(assoc=1), type = "response")
```

```
#>   factor     AME     SE       z      p   lower  upper
#>  arsenic  0.0890 0.0085 10.4179 0.0000  0.0723 0.1058
#>    assoc -0.0308 0.0178 -1.7253 0.0845 -0.0657 0.0042
```

```
avdiff<-dydx(wells, mod3a, "assoc", change = "minmax") #; mean(avdiff$dydx_assoc, na.rm = T)
```

- On the scale of the linear predictor

```
margins(mod3a, at = list(assoc=1), type = "link"); exp(-0.1305)
```

```
#>  at(assoc)   assoc arsenic
#>          1 -0.1305  0.3777
```

```
#> [1] 0.878
```

What's this?

Comparing results

Model	Frequentist Estimates	Bayesian Estimates
Linear Model [1]	-0.036	-0.035
Logit Model (Predicted probs.) [1]	RD= -0.036; RR= 0.939	RD = -0.035; RR= 0.94
Logit Model (Predicted probs.) [2]	RD= -0.033; RR= 0.939	RD= -0.031; RR= 0.945
AME (prediction `StdGLM`) [2]	RD= -0.031; RR= 0.948	-
AME (`margins` at means) [2]	RD= -0.031	-
AME (marginal standardization) [2]	RD= -0.021 ; RR= 0.965	RD= -0.032 ; RR= 0.946

[1] Single predictor; [2] Two predictors

Estimated RR and RD using the probabilities from the logistic model

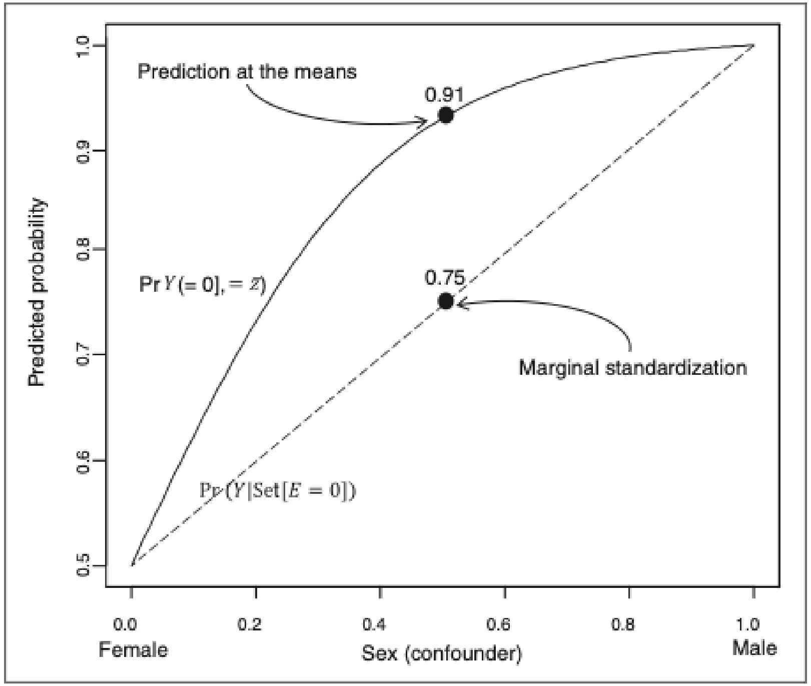


Figure 1. Half of a sigmoid curve depicting calculation of predicted probabilities following logistic regression using marginal standardization (dashed straight line) and prediction at the means (solid curved line) in unexposed people from a hypothetical population.

Estimating predicted probabilities from logistic regression: different methods correspond to different target populations, by Clemma J Muller & Richard F MacLehose

(III) Other Generalized Linear Models:

Log binomial: The log-link function maps the probability of disease

- Attempts to find a MLE if it exists.
- Useful for RDs and RRs
- Software/packages: SAS's GENMOD, R's GLM or STATA's GLM/binreg.

- The MLE can be on the boundary of the parameter space, leading to the difficulty of finding the MLE.

- The log-link function maps the probability of disease onto the negative real line, requiring the constraint that a linear predictor must be negative.

Log-Binomial

Advantages:

- Single uniform estimate
- Biostatisticians will love you

Disadvantages:

- Very difficult to fit
- Still possible to get impossible values

For RDs, fit a GLM with a binomial variance and an identity link

$$g[Pr(Y=1|X=x)] = \beta_0 + \beta_1 X$$

Wacholder S.Binomial regression in GLIM: estimating risk ratios and risk differences.
Am J Epidemiol 1986. Jan;123(1):174-84.62

Spiegelman D, Hertzmark E. Easy SAS calculations for risk or prevalence ratios and differences.
Am J Epidemiol 2005 Aug 1;162(3):199-200.

Logistic vs Log-Binomial

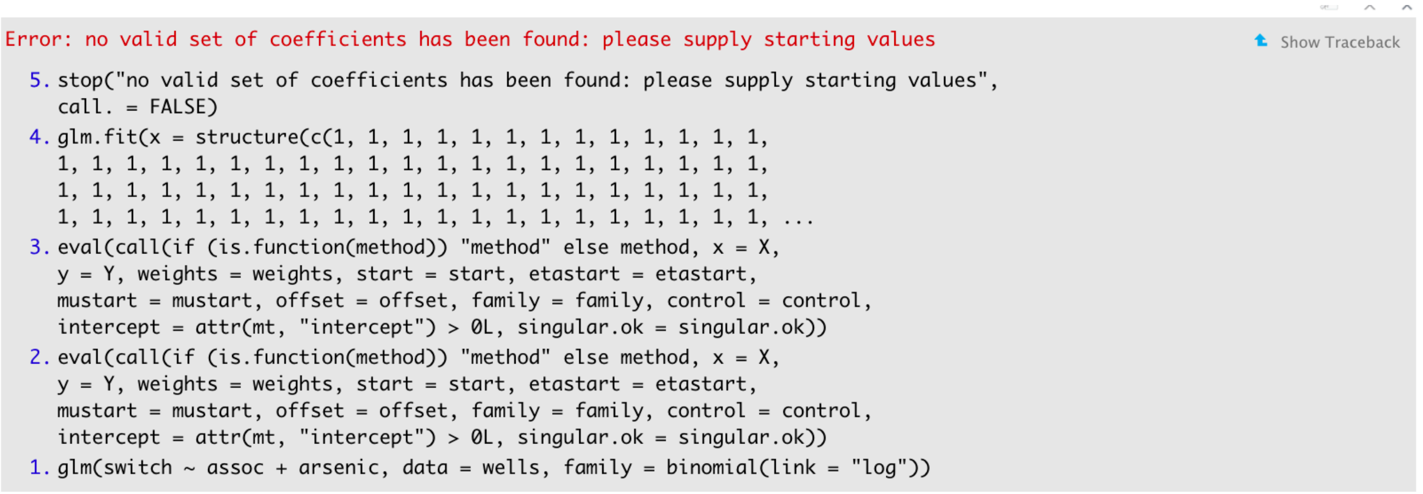
- Both model $Pr(Y|X, c)$
- Both assume that the error terms have a binomial distribution.
- Different links between the X and the Pr(Y):
  - Logistic regression, the *logit* function
  - Log-binomial model, the *log* function
- In general, the log-binomial model produces an unbiased estimate of the adjusted relative risk.
- Minimal restriction unless adjustment for many confounders is needed.
- The CIs for the adjusted RR may be narrower than is true

Log-Binomial

Let's estimate our RD and RR Using `glm` or `glm2` package

```
mod4a <- glm(switch ~ assoc + arsenic, data = wells,
             family=binomial(link="identity"))
summ(mod4a, confint=T)
mod4b <-glm(switch ~ assoc + arsenic,  data = wells,
            family = binomial(link = "log"))
summ(mod4b, confint=T)
```

Error!

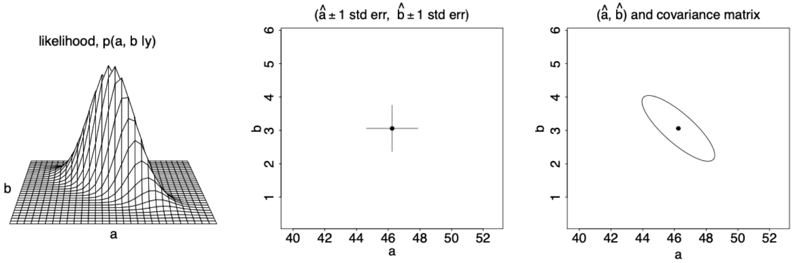


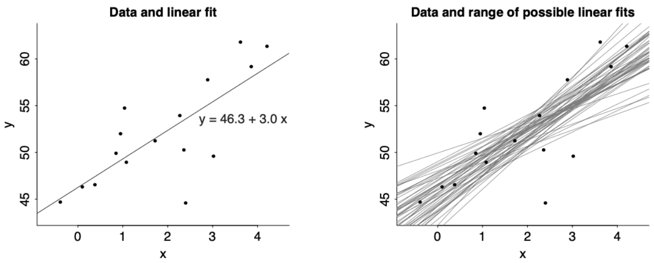
```
Error: no valid set of coefficients has been found: please supply starting values

 5. stop("no valid set of coefficients has been found: please supply starting values",
    call. = FALSE)
 4. glm.fit(x = structure(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
 3. eval(call(if (is.function(method)) "method" else method, x = X,
    y = Y, weights = weights, start = start, etastart = etastart,
    mustart = mustart, offset = offset, family = family, control = control,
    intercept = attr(mt, "intercept") > 0L, singular.ok = singular.ok))
 2. eval(call(if (is.function(method)) "method" else method, x = X,
    y = Y, weights = weights, start = start, etastart = etastart,
    mustart = mustart, offset = offset, family = family, control = control,
    intercept = attr(mt, "intercept") > 0L, singular.ok = singular.ok))
 1. glm(switch ~ assoc + arsenic, data = wells, family = binomial(link = "log"))
```

To understand convergency issues we need to talk about:

The Likelihood function, defined (in a regression model) as the probability density of the data given the parameters and predictors.

- Maximizing the likelihood requires minimizing the sum of squared residuals;
  - Hence the least squares estimate can be viewed as a maximum likelihood estimate under the normal model (OLS).





ROS-Gelman, Hill & Vehtari

Likelihood for GLM - Logistic Regression

For binary logistic regression with data $y_i = 0$ or 1, the likelihood is:

$$p(y|\beta, X) = \prod_{i=1}^{n} (logit^{-1}(X_i\beta))^{y_i}(1 - logit^{-1}(X_i\beta))^{1-y_i}$$

To find the β that maximizes this expression:

- Compute the derivative of the logarithm of the likelihood.
- Set this derivative equal to 0, and solve for β.

There is **no closed-form solution**, but the maximum likelihood estimate can be found using an iterative optimization algorithm that converges to a point of *zero* derivative and

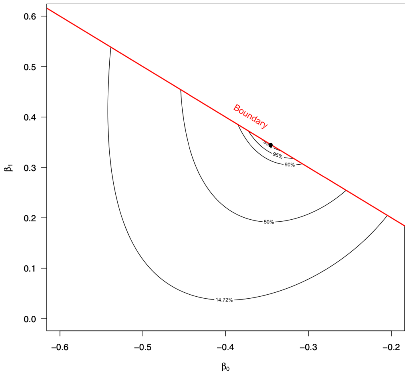
- Thus the vector of coefficients β that maximizes the likelihood, when such a maximum exists.

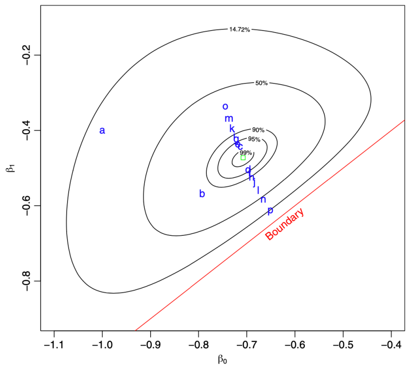
ROS-Gelman, Hill & Vehtari

Convergency Log-Binomial

- Failed convergence occurs whenever the maximizing process fails to find the MLE.
- Estimation challenges can be grouped based on the location of the true maximum of the log-likelihood function, relative to the parameter space.
    - On the boundary of the parameter space (i.e., where the linear predictor equals 0);
    - In the limit (i.e., as the linear predictor heads towards -∞);
    - Inside the parameter space.

These three regions span the entire parameter space and are mutually exclusive.





Williamson, T.Log-binomial models: exploring failed convergence

Convergency Log-Binomial

- Software utilizes iterative weighted least squares (IWLS*) approach or variations of IWLS to find MLEs for generalized linear models.

- For log-binomial models, the weights used by the IWLS approach contain the term $1/(1-p)$, where $p = exp(X^T\beta)$ with a range from 0 to 1.

- The MLE of a log-binomial model is likely to be too sensitive to outliers because a very large p has a large influence on the weights.

- MLE and pseudolikelihood estimators are deteriorated in presence of outliers.

- The level of deterioration differed when the relationships between the confounder and the outcome was not in a simple form.

Chen et al.: BMC Medical Research Methodology 2014 14:82.

Convergence Issues

… Requirement that the linear predictor be constrained to be negative … when the issue is the boundary of the parameter space (i.e., where the linear predictor equals 0) the solution resides on the boundary

```
wells$assoc1 <- 1- wells$assoc
mod4b <-glm(switch ~ -1 + assoc1 + arsenic,  data = wells,
            family = binomial(link = "log"))
round(summ(mod4b, confint=T, exp=T)$"coeftable", 2)
```

```
#>         exp(Est.) 2.5% 97.5% z val. p
#> assoc1       0.77 0.74  0.81  -10.2 0
#> arsenic      0.79 0.77  0.81  -21.1 0
```

Interpretation???, setting intercept to 1; log(1) =0; Pr(Y=1) = 0.5 ??

Williamson, T. Log-binomial models: exploring failed convergence

Convergence Issues

Requirement that the linear predictor be constrained to be negative (ii)

```r
library(glm2)
mod4b1 <-glm2(switch ~ assoc + arsenic,
      data = wells,
      family = binomial(link = "log"),
      start = c(-1, -1,-1))
round(summ(mod4b1, confint = T, exp = T)$"c
```

```
#>              exp(Est.) 2.5% 97.5% z val.    p
#> (Intercept)       0.42 0.40  0.44 -35.87 0.00
#> assoc             0.92 0.87  0.98  -2.69 0.01
#> arsenic           1.10 1.10  1.11  28.72 0.00
```

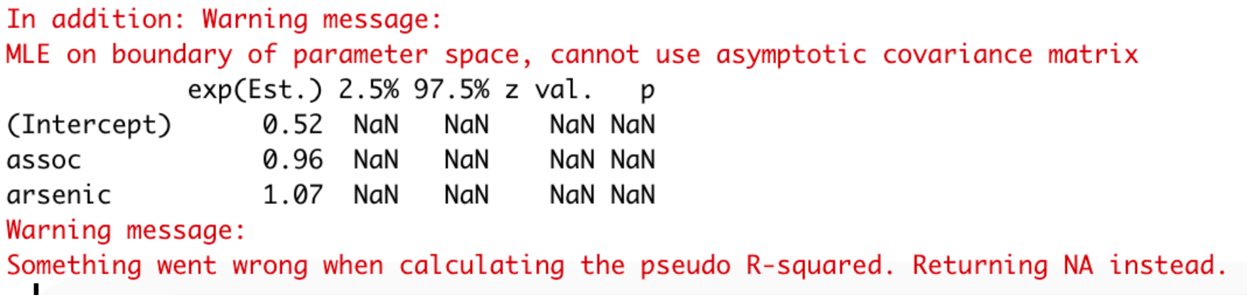
```r
#
mod4b2 <-glm2(switch ~ assoc + arsenic,
        data = wells,
        family = binomial(link = "log"),
        start = c(-0.5, -0.5,-0.5))
round(summ(mod4b2, confint = T, exp = T)$"c
```

```
#>              exp(Est.) 2.5% 97.5% z val.    p
#> (Intercept)       0.48 0.46  0.51 -29.35 0.00
#> assoc             0.93 0.88  0.98  -2.57 0.01
#> arsenic           1.09 1.08  1.09  29.35 0.00
```

Log-Binomial

Using the `logbin` package

```
library(logbin)
mod4c<- logbin(switch ~ assoc + arsenic, data = wells)
round(summ(mod4c, confint = T, exp = T)$"coeftable", 2)
mod4d<- logbin(switch ~ assoc + arsenic, data = wells, trace = 1, maxit = 100000)
round(summ(mod4d, confint = T, exp = T)$"coeftable", 2)
```



```
In addition: Warning message:
MLE on boundary of parameter space, cannot use asymptotic covariance matrix
            exp(Est.) 2.5% 97.5% z val.   p
(Intercept)      0.52  NaN   NaN    NaN NaN
assoc            0.96  NaN   NaN    NaN NaN
arsenic          1.07  NaN   NaN    NaN NaN
Warning message:
Something went wrong when calculating the pseudo R-squared. Returning NA instead.
```

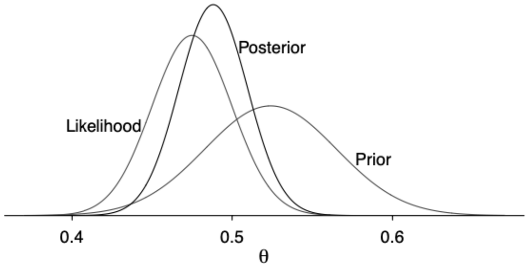
The Bayesian way

Estimating **RD**

```
mod5a <- stan_glm(switch ~ assoc + arsenic, data = wells, family="gaussian", refresh=0)
round(mod5a$stan_summary[1:3,1:5-10], 3) #;print(mod5a, digits=3)
```

```
#>               mean se_mean    sd   2.5% 97.5% n_eff Rhat
#> (Intercept)  0.453       0 0.018  0.418 0.489  4972    1
#> assoc       -0.031       0 0.018 -0.066 0.003  5077    1
#> arsenic      0.082       0 0.008  0.066 0.098  5004    1
```

> *"In Bayesian inference, the uncertainty for each parameter in the model automatically accounts for the uncertainty in the other parameters. This property of Bayesian inference is particularly relevant for models with many predictors, and for advanced and hierarchical models."*



The Bayesian way

Estimating **RR**

```
mod5b <-stan_glm(switch ~ assoc + arsenic, data = wells,
                 family = binomial(link = "log"), refresh=0)
round(mod5b$stan_summary[1:3,1:5-10], 3)#; print(mod5b, digits=3)
```

```
#>               mean se_mean    sd   2.5%  97.5% n_eff Rhat
#> (Intercept) -0.650   0.000 0.022 -0.691 -0.607  2876    1
#> assoc       -0.067   0.001 0.025 -0.121 -0.018  1650    1
#> arsenic      0.071   0.000 0.004  0.062  0.077  1758    1
```

```
exp(mod5b$coefficients["assoc"] )
```

```
#> assoc
#> 0.936
```

The Bayesian way

> *If the prior distribution on the parameters is uniform, then the posterior density is proportional to the likelihood function, and the posterior mode—the vector of coefficients β that maximizes the posterior density is the same as the maximum likelihood estimate.*
>
> *The benefit of Bayesian inference with a non-informative prior is that we can use simulations from the entire posterior distribution*
>
> - *Not just a maximum or any other point estimate—to summarize uncertainty, and we can also use these simulations to make probabilistic predictions.*

ROS-Gelman, Hill & Vehtari

Comparing results

Model	Frequentist Estimates	Bayesian Estimates
Logit Model (Predicted probs.) [2]	RD= -0.033; RR= 0.939	RD= -0.031; RR= 0.945
AME (prediction `StdGLM`) [2]	RD= -0.031; RR= 0.948	-
AME (`margins` at means) [2]	RD= -0.031	-
AME (marginal standardization) [2]	RD= -0.021 ; RR= 0.965	RD= -0.032 ; RR= 0.946
GLM: Log-Binomial [2]	RD= ?; RR= 0.928 or RRa= 0.773	RD= -0.031; RR= 0.936

[1] Single predictor; [2] Two predictors

What about Confidence Intervals??

1) **Bootstrap!!!**

```
RR <- function(data,d) {
 dta <- data[d,]
 mod <- glm(switch ~ assoc + arsenic, data=
 pp0 <- predict(mod, newdata=transform(dta,
 pp1 <- predict(mod, newdata=transform(dta,
 return(RR=pp1/pp0)
 }

 RD <- function(data,d) {
 dta <- data[d,]
 modd <- glm(switch ~ assoc + arsenic, data
 pp0 <- predict(modd, newdata=transform(dta
 pp1 <- predict(modd, newdata=transform(dta
 return(RD=pp1-pp0)
}

library(boot)
boot.RR <- boot(data=wells, statistic=RR, R
boot.RD <- boot(data=wells, statistic=RD, R

RR <- quantile(boot.RR$t, probs=c(0.5,0.025
RD <- quantile(boot.RD$t, probs=c(0.5,0.025
```

```
RR
```

```
#>    50%  2.5% 97.5%
#> 0.947 0.878 1.004
```

```
RD
```

```
#>       50%     2.5%    97.5%
#> -0.03027 -0.07100  0.00528
```

--

2) Alternatively, use prudently the `margins` or `StdReg` and other resources

3) **Use Bayesian Inference!!!**

Conclusions:

1) You don’t ever have to report another OR again, (unless you have a cumulative case-control study with an unknown sampling fraction)

- The popularity of the OR was based largely on statistical convenience, but modern software has largely overcome those early limitations.

2) The interpretation depends on the method and the assumptions required for each estimation!!!

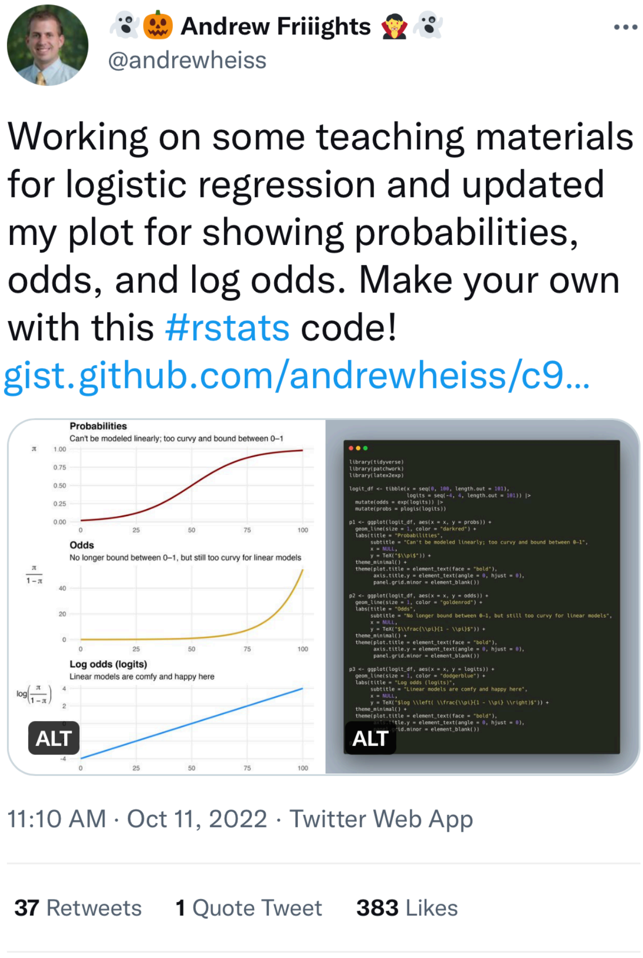
3) Take a pledge, join a support group, and kick the habit.

A kind message from Dr. Jay Kaufman!

QUESTIONS?

COMMENTS?

RECOMMENDATIONS?



👻🎃 Andrew Friiights 🧛 👻
@andrewheiss

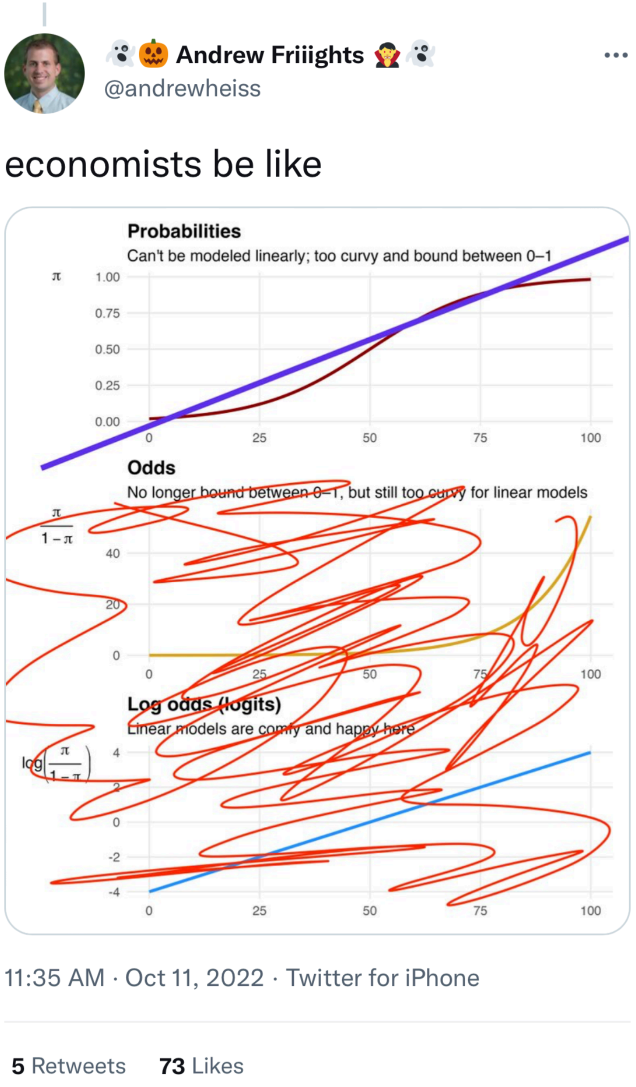
Working on some teaching materials for logistic regression and updated my plot for showing probabilities, odds, and log odds. Make your own with this #rstats code! gist.github.com/andrewheiss/c9…

Probabilities
Can't be modeled linearly; too curvy and bound between 0–1
π
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
Odds
No longer bound between 0–1, but still too curvy for linear models
$\frac{\pi}{1-\pi}$
40
20
0
0
25
50
75
100
Log odds (logits)
Linear models are comfy and happy here
$\log\left(\frac{\pi}{1-\pi}\right)$
4
2
-4
0
25
50
75
100
ALT
ALT

```r
library(tidyverse)
library(patchwork)
library(latex2exp)

logit_df <- tibble(x = seq(0, 100, length.out = 101),
                   logits = seq(-4, 4, length.out = 101)) |>
  mutate(odds = exp(logits)) |>
  mutate(probs = plogis(logits))

p1 <- ggplot(logit_df, aes(x = x, y = probs)) +
  geom_line(size = 1, color = "darkred") +
  labs(title = "Probabilities",
       subtitle = "Can't be modeled linearly; too curvy and bound between 0-1",
       x = NULL,
       y = TeX("$\\pi$")) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"),
        axis.title.y = element_text(angle = 0, hjust = 0),
        panel.grid.minor = element_blank())

p2 <- ggplot(logit_df, aes(x = x, y = odds)) +
  geom_line(size = 1, color = "goldenrod") +
  labs(title = "Odds",
       subtitle = "No longer bound between 0-1, but still too curvy for linear models",
       x = NULL,
       y = TeX("$\\frac{\\pi}{1 - \\pi}$")) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"),
        axis.title.y = element_text(angle = 0, hjust = 0),
        panel.grid.minor = element_blank())

p3 <- ggplot(logit_df, aes(x = x, y = logits)) +
  geom_line(size = 1, color = "dodgerblue") +
  labs(title = "Log odds (logits)",
       subtitle = "Linear models are comfy and happy here",
       x = NULL,
       y = TeX("$log \\left( \\frac{\\pi}{1 - \\pi} \\right)$")) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"),
        axis.title.y = element_text(angle = 0, hjust = 0),
        panel.grid.minor = element_blank())
```

11:10 AM · Oct 11, 2022 · Twitter Web App

37 Retweets **1** Quote Tweet **383** Likes



👻🎃 Andrew Friiights 🧛 👻
@andrewheiss

economists be like

Probabilities
Can't be modeled linearly; too curvy and bound between 0–1
π
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
Odds
No longer bound between 0–1, but still too curvy for linear models
$\frac{\pi}{1-\pi}$
40
20
0
0
25
50
75
100
Log odds (logits)
Linear models are comfy and happy here
$\log\left(\frac{\pi}{1-\pi}\right)$
4
2
0
-2
-4
0
25
50
75
100

11:35 AM · Oct 11, 2022 · Twitter for iPhone

5 Retweets **73** Likes

Extra slides

Summary table of What, When and Why

Method	Characteristics	Outcome	Measure
Standardization	Weight-based adjustment; Depends on the standard pop. selected; No homogeneity needed	Binary or categorical	SMR
Mantel- Haenszel Adjustment	Requires homogeneity; Do not handle clusters	Binary or categorical	RD, RR, OR
Regression Adjustment	Efficient, Useful for prediction, adjust for several covariates, *require assumptions*	Any type	RD, RR, OR; AME/ATE
IPTW [1]	Regression + Weights: 1/Pr(X=1, covars); Ensure Exchangeability; Only for measured Confounders	Any type	**Causal** RD, RR, OR; AME/ATE

Standardization & M-H Adjustment

- Non-parametric
- Adjustment based on weights
- Useful of a small number of covariates
- Basic arithmetic calculations
- Homogeneity assumption for M-H Adjustment
- Useful for few categorical covariate, limited for continuous variables

Regression Adjustment

- Parametric*
- Efficient (could provide measures of association for different covariates)
- Adjustment for more than one covariate at a time
- Handles different types of covariates (continuous, binary, counts, etc.)
- Control/Adjust for confounding
- Helpful for prediction

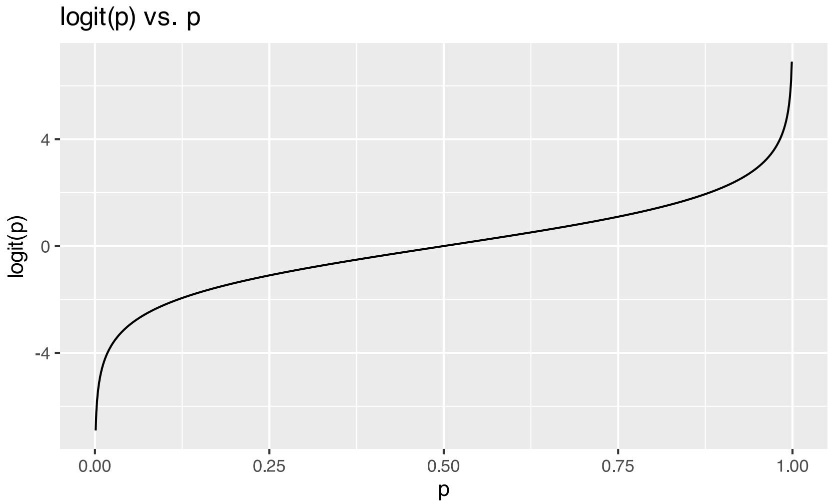
But it has a cost!

Outcome’s Distribution

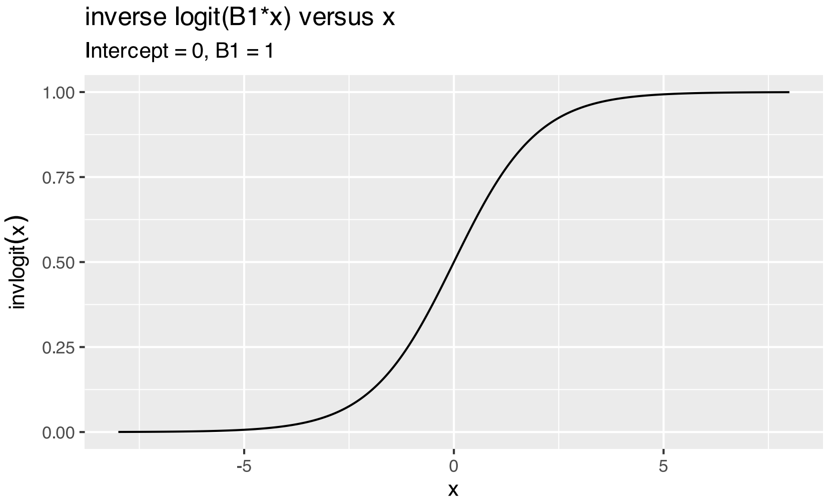
Type	Model	Estimate
Continuous	Linear Regression	RD
Binary	Logistic Regression	OR $\cong$ RR; **RD**
Categorical	Multinomial /Polytomous Logistic Regression	OR ; **RD**
Ordinal	Ordinal Logistic Regression	OR; **RD**
Counts	Poisson, Negative Binomial	IR, IRR

Graphing functions

logit maps the range (0, 1) to (-∞, ∞) useful to model binary outcomes



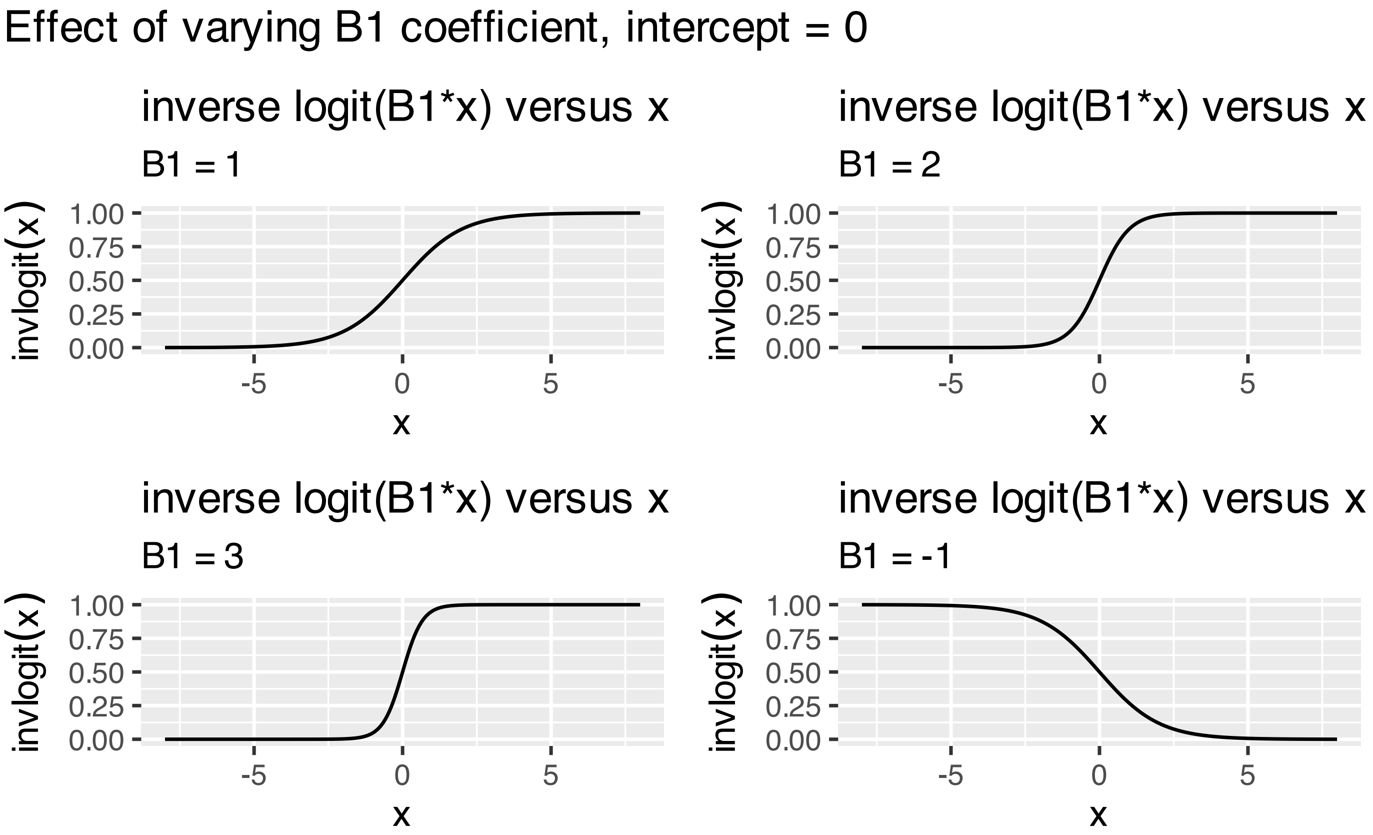
Inverse logit (logistic) maps back to the probability scale



Remember invlogit = $\frac{e^{\beta x}}{1+e^{\beta x}}$ how does graph change i) as coefficient of x varies? ii) with addition of intercept?

If x = 0, invlogit = $\frac{e^{0}}{1+e^{0}} = \frac{1}{1+1} = 0.5$ as shown on the graph

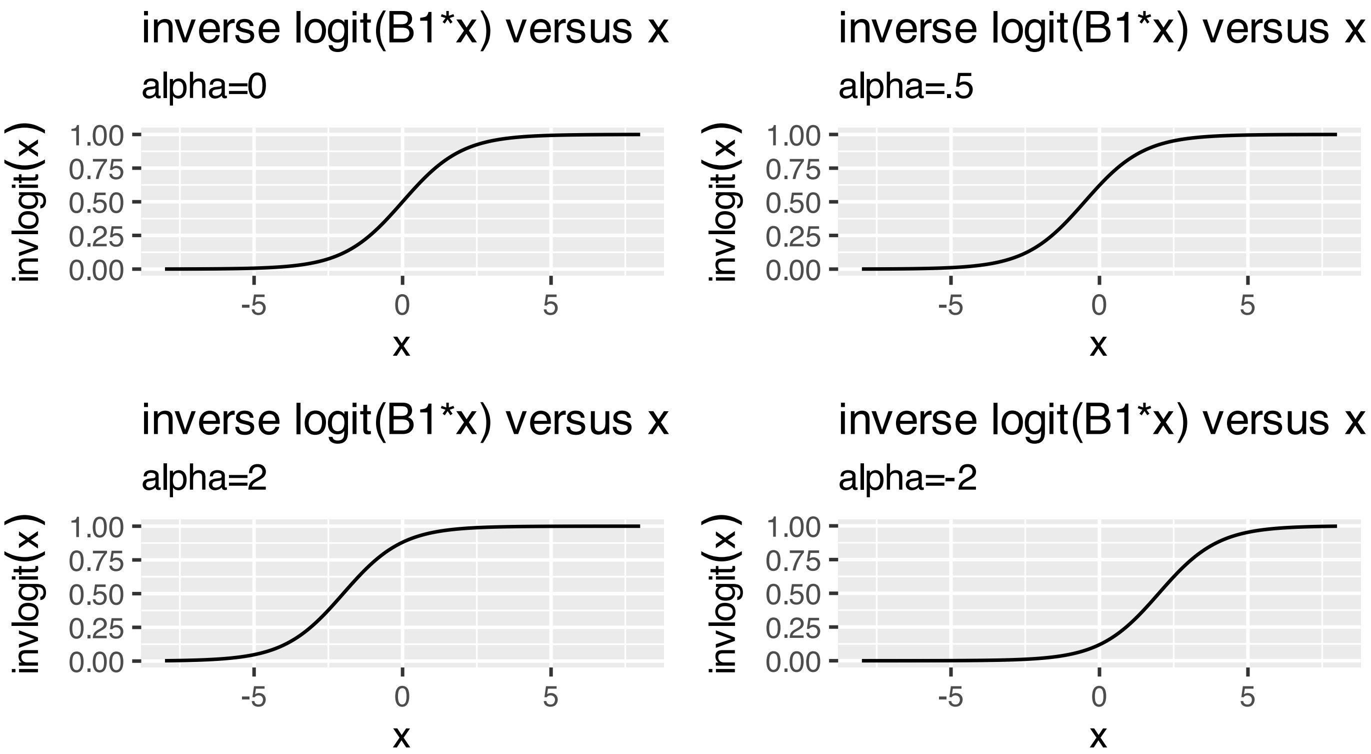
Inverse logit graphs



Notice maximum slope remains at invlogit = 0.5 or x = 0
As β_1 changes, slope becomes steeper ($\beta*{1}$ increases) or shallower ($\beta*{1}$ decreases)
but the curve doesn't shift position.

Inverse logit graphs

Effect of varying intercept with B1 = 1



Shifts the curves **horizontally** but slope remains constant and maximum remains at invlogit(x) = 0.5 or x = - intercept

R - extension of ‘margins’ and more

- Marginal effects of the adjusted estimates
- Provides estimates for all covariates
- R-margins does not include the “over” option but is replaced by the “at=list” option

Useful resources:

Margins

Margins, blog

Estimating Risk Ratios and Risk Differences Alternatives to Odds Ratios