

Binary Response and Logistic Regression

Marvin Schmitt

May 11, 2021

- 1 Model
- 2 Implementation in R
- 3 Model Evaluation
- 4 Outlook

Model

Binary Response Data

- **General setting:** $x_i \in \mathbb{R}^n, y_i \in \{0, 1\}$
- **Examples:**
 - Stimulus discrimination tasks
 - Shooter paradigm
 - Just noticeable difference in color perception
 - Mortality
 - *Is regular physical exercise life extending?*
 - *What are risk factors for severe COVID-19 symptoms?*
 - Performance assessment
 - Student assessment tests
 - Exam design: modeling task difficulty

Estimation

Basic idea: predict probability for class 1 as

$$P(Y = 1) = \frac{\exp(\eta)}{1 + \exp(\eta)} \text{ with } \eta = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

- Underlying linear model $\eta_i = \beta_0 + \sum_{j=1}^k \beta_j x_j$ that is plugged into the logistic function to obtain $P(Y = 1) \in [0, 1]$ and $P(Y = 1) + P(Y \neq 1) = 1$.

Likelihood

Likelihood $\mathcal{L}(\beta|x_1, \dots, x_n) = \prod_{i=1}^n \mathcal{L}_i(\beta|x_i)$

$$\begin{aligned} l(\beta|x_1, \dots, x_n) &= \sum_{i=1}^n l_i(\beta|x_i) \\ &= \sum (1 - y_i) \log(1 - p(x_i; \beta)) + y_i \log p(x_i; \beta) \\ &= \sum y_i \log \frac{p(x_i; \beta)}{1 - p(x_i; \beta)} + \log(1 - p(x_i; \beta)) \\ &= \sum y_i x_i \beta - \log(1 + \exp(x_i \beta)) \\ &= \sum_{i=1}^n y_i \eta_i - \log(1 + \exp(\eta)) \end{aligned}$$

- Maximize log-likelihood $l = \sum_{i=1}^n y_i \eta_i - \log(1 + \exp(\eta)) =: \text{LL}$

Inference

■ Data format

- The criterion must be coded as 0/1 or as factor.
- Predictors must be metric or dummy-coded

■ Interpretation

- Odds ratio for predictor x_j is equal to $\exp(\beta_j)$
- Odds ratio OR_j quantifies how the odds $\frac{P(Y=1)}{P(Y=0)}$ change when x_j increases by 1 unit.

Implementation in R

Syntax

General syntax

- Command `glm()` (generalized linear model)
- Formula syntax:
 - `y~x1+x2+x3` (no interactions)
 - `y~x1*x2` (interactions)
 - `y~x1+x2+x3+x1:x2` (selected interactions)
- Must provide a link function to `glm()` through the `family` parameter (cf. section Other link functions)
 - For logistic regression, we use `family=binomial('logit')`.

Examples

```
m1 = glm(correct_response ~ iq + math_skill, data=df, family=binomial('logit'))
m2 = glm(fatal_accident ~ bmi + risk_seeking + gender, family=binomial('logit'))
m3 = glm(vaccination_skeptic ~ iq * income, family=binomial('logit'))
```

Toy Example

The dataset `df`¹ contains the yearly sick leave hours, number of tweets on Twitter, IQ, and hair length of $N = 100$ employees along with their gender (binary: male/female) and whether they have ever suffered from depression (binary: yes/no):

```
df %>% slice(sample(nrow(df))) %>% head(5)
```

```
##   ID depr gender avg_sickhours n_tweets iq hairlength
## 1 75  no   male          13.0         3 90          19.5
## 2 91  no   male          12.7         3 84           4.8
## 3 23  yes  female         11.5        12 90          46.3
## 4 33  no   female          5.4         3 97          40.4
## 5 86  no   male           8.6         1 83           4.4
```

```
table(df$gender)
```

```
##
## female    male
##       50      50
```

¹www.github.com/marvinschmitt/talk-binary-response

We define gender as criterion and avg_sickhours as predictor. The logit link function leads to a logistic regression. The output's coefficients correspond to β_0, \dots, β_k .

```
m = glm(gender ~ avg_sickhours, data = df, family = binomial('logit'))  
m$coefficients
```

```
## (Intercept) avg_sickhours  
## -4.2693393 0.4440937
```

- We can calculate η from the underlying linear model:

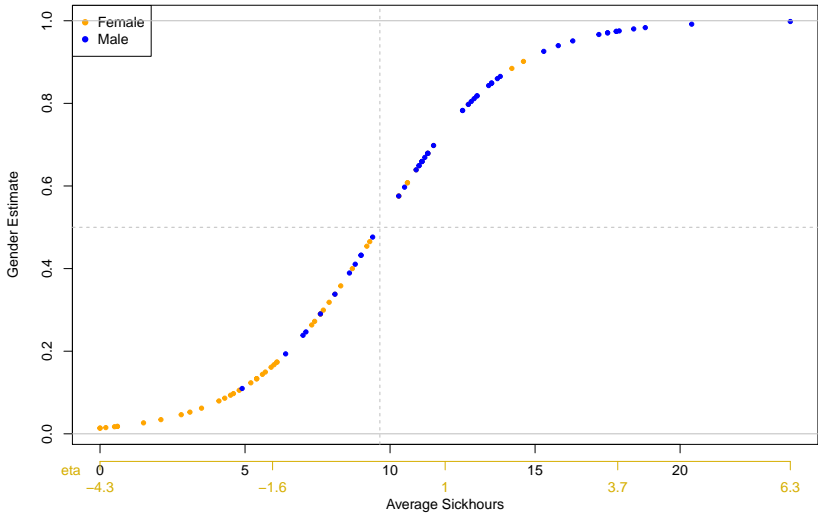
$$\eta = -4.27 + 0.44x_1$$

- Thus, the criterion estimate is:

$$\hat{P}(Y = \text{male}) = \frac{\exp(\overbrace{-4.27 + 0.44x_1}^{\eta})}{1 + \exp(\underbrace{-4.27 + 0.44x_1}_{\eta})}$$

- The odds for gender=m are increased by the factor $\exp(0.44) = 1.55$ per additional sick leave hour.

Gender Estimate by Average Sickhours



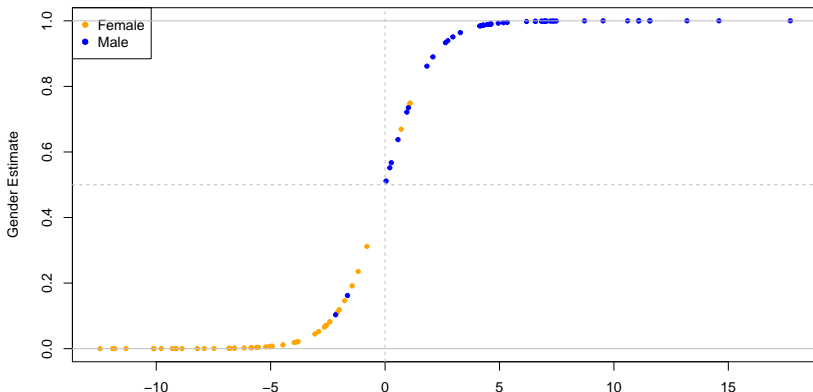
Predictors: Avg. sick hours (x_1), Number of tweets (x_2)

```
m = glm(gender ~ avg_sickhours + n_tweets, data = df, family = binomial('logit'))  
m$coefficients
```

```
## (Intercept) avg_sickhours      n_tweets  
## -10.109104      1.621815      -1.197958
```

$$\blacksquare \eta = -10.11 + 1.62x_1 - 1.2x_2$$

Gender Estimate by Average Sickhours and number of tweets



Predictors: Avg. sick hours (x_1), Number of tweets (x_2), IQ (x_3)

```
m = glm(gender ~ avg_sickhours + n_tweets + iq, data = df, family = binomial('logit'))
```

```
## Warning: glm.fit: algorithm did not converge
```

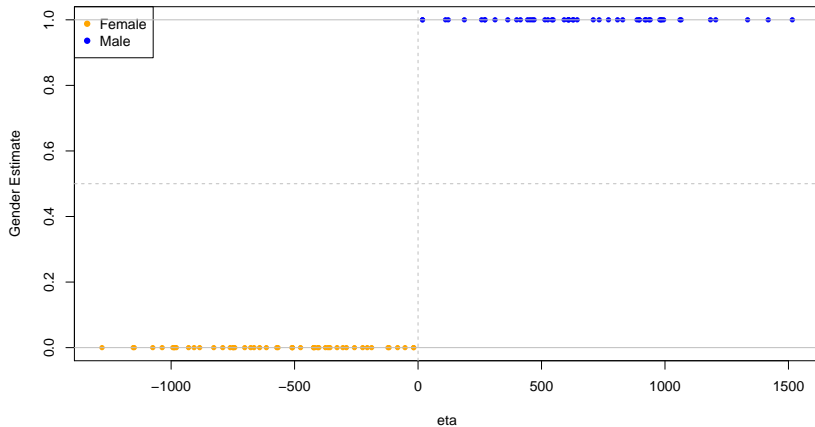
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
m$coefficients
```

```
## (Intercept) avg_sickhours      n_tweets          iq  
## 1998.03388      150.38876     -79.29431     -33.13080
```

- $\eta = 1998.03 + 150.39x_1 + -79.29x_2 + -33.13x_3$
- Note the output Warning: glm.fit: algorithm did not converge
 - Issue: Data is linearly separable (cf. section Issue: Linear Separability)
 - See the plot (next slide)

Gender Estimate by Average Sickhours, number of tweets, and IQ

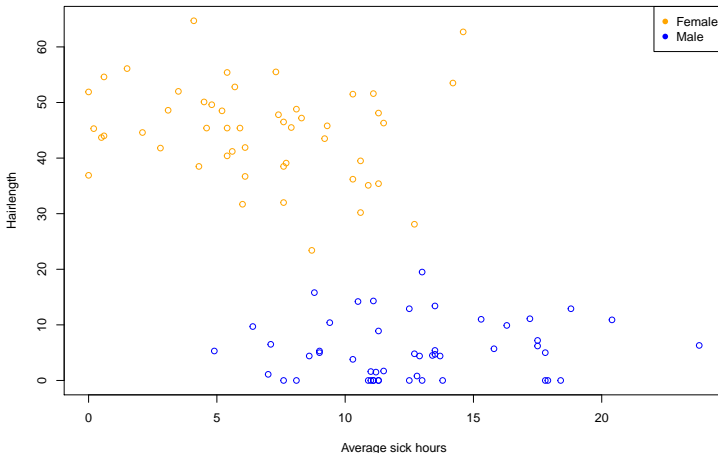


Predictors: Avg. sick hours (x_1), Hairlength (x_2)

```
m = glm(gender ~ avg_sickhours + hairlength, data = df, family = binomial('logit'))
```

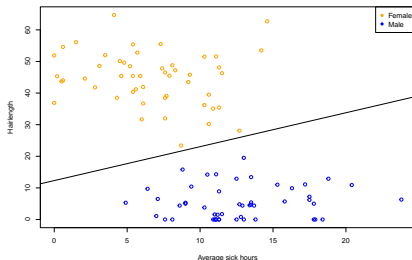
```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



Issue: Linear Separability

- Linear separability of the data causes convergence issues.



- Unstable estimates of the parameters and their standard errors.
- Alternative: *Exact logistic regression*

Model Evaluation

Logarithmic Scoring

- Basic evaluation of predicted probabilities:
 - For $Y_{true} = 1$, the predicted probability $\hat{P}(Y = 1)$ should be close to 1.
 - For $Y_{true} = 0$, the predicted probability $\hat{P}(Y = 1)$ should be close to 0.
- Compute $\text{logScore} = Y_i \ln(\hat{Y}_i) + (1 - Y_i) \ln(1 - \hat{Y}_i)$
 - $= \ln(\hat{Y}_i)$ if $Y_i = 1$
 - $= \ln(1 - \hat{Y}_i)$ if $Y_i = 0$

Model Selection

- Model selection aims at selecting a model M_i from a set of candidate models M_1, \dots, M_m .
- The choice depends on the selection criterion and the
- For a model M with k parameters, the **Akaike-Information-Criterion** is defined as

$$AIC_M = -2 \log L_M + 2k$$

- R provides the `step()` function for stepwise selection from a set of models².
- The `step()` function uses the AIC as selection criterion for logistic regression models.

²stepwise selection is not an ideal selection technique. Contemporary alternative: regularized methods

```
m0 = glm(chd~1, data=chd_data, family=binomial('logit'))
m1 = step(m0, direction='both', trace=0,
          scope='~cigs+chol+weight+age')
summary(m1)
```

```
##
## Call:
## glm(formula = chd ~ chol + age + cigs, family = binomial("logit"),
##      data = chd_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1307  -0.4596  -0.3550  -0.2587   2.6391
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.655131   1.660169  -5.213 1.85e-07 ***
## chol         0.014092   0.003422   4.118 3.82e-05 ***
## age          0.056783   0.029257   1.941  0.0523 .
## cigs         0.017611   0.010368   1.699  0.0894 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 302.35  on 498  degrees of freedom
## Residual deviance: 276.65  on 495  degrees of freedom
## AIC: 284.65
##
## Number of Fisher Scoring iterations: 5
```

Likelihood Ratio Test

- We can test competing nested models with the **Likelihood Ratio (LR) Test**
- Small model M_1 with k_1 parameters, Large model M_2 with k_2 parameters
- The difference of log-likelihoods is χ^2 distributed:
 - Test statistic $G^2 = -2LL_{M_1} - (-2LL_{M_2}) \sim \chi^2(df = k_2 - k_1)$
 - If $p < .05$, the larger model improves model fit.
- Implementation in R for example with `anova([models], test='Chisq')`

```
anova(m0, m1, m2, m3, m4, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: chd ~ 1
## Model 2: chd ~ age
## Model 3: chd ~ age + cigs
## Model 4: chd ~ age + cigs + chol
## Model 5: chd ~ age + cigs + chol + height
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         498      302.35
## 2         497      296.76  1    5.5866  0.01810 *
## 3         496      293.78  1    2.9815  0.08422 .
## 4         495      276.65  1   17.1342 3.483e-05 ***
## 5         494      275.52  1    1.1310  0.28755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hosmer and Lemeshow Goodness-of-fit test

- Approach of the HL test:
 - Partition the model population space into bins (*risk decentiles*)
 - Compare observed relative bin counts with predicted relative bin counts
 - Test statistic follows a χ^2 distribution
- Interpretation
 - $p < .05$ indicates a systematic deviance between observed and predicted bin counts.
- Reasons for a bad fit
 - Nonlinear influence of predictors on η
 - Solution: Polynomial logistic regression
 - e.g. $\eta = \beta_0 + \beta_{11}x_1 + \beta_{12}x_1^2 + \dots + \beta_{1p}x_1^p + \beta_{21}x_2 + \dots + \beta_{kp}x_k^p$
 - Interaction between predictors
 - Solution: Allow and analyze interactions ($y \sim x_1 * x_2$).


```
df$gender01 = as.numeric(df$gender)-1 # recode to 0/1 for HL-test
m = glm(gender01 ~ avg_sickhours, data=df, family=binomial('logit'))
y_hat = predict(m, type="response")
hoslem.test(df$gender01, y_hat) # default: g=10 bins
```

```
##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data: df$gender01, y_hat
## X-squared = 3.6825, df = 8, p-value = 0.8846
hoslem.test(df$gender01, y_hat, g=5) # g=5 bins
```

```
##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data: df$gender01, y_hat
## X-squared = 0.57315, df = 3, p-value = 0.9026
```

Effect size

McFadden's ρ^2

- **Problem:** We cannot calculate the explained variance R^2
- **Approach:** Calculate McFadden's ρ^2 with the log-likelihood of the full model `m1` and the log-likelihood of the null model `m0` without predictors.
- **Optional:** Include a correction for the number of predictors k .

- **Formula:**
$$\rho^2 = 1 - \frac{\ln(L_1) \overset{\text{correction}}{-k}}{\ln(L_0)}$$

```
m0 = glm(gender ~ 1, data=df, family=binomial('logit'))
m1 = glm(gender ~ avg_sickhours, data=df, family=binomial('logit'))
K = length(m1$coefficients) - 1 # number of predictors
as.numeric(1 - logLik(m1) / logLik(m0)) %>% round(3)
```

```
## [1] 0.36
```

```
as.numeric(1 - (logLik(m1)-K)/logLik(m0)) %>% round(3)
```

```
## [1] 0.346
```

Nagelkerke's R^2

- Another approach to compute an analogon to R^2 is Nagelkerke's (pseudo-) R^2 :

```
m = glm(gender ~ avg_sickhours, data=df, family=binomial('logit'))
N = nrow(df)
(1-exp((m$dev-m$null)/N))/(1-exp(-m$null/N))
```

```
## [1] 0.5238511
```

Outlook

Predictions

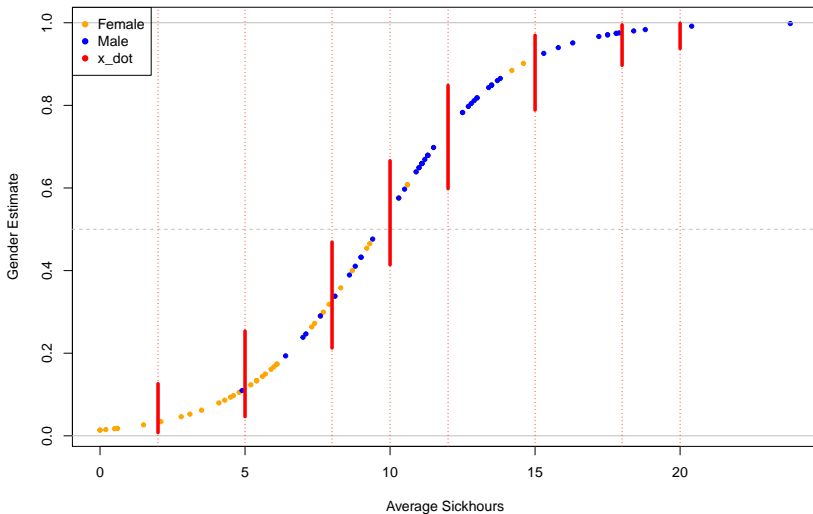
- Given a new input \dot{x} , the prediction on the linear predictor is $\hat{\eta} = \dot{x}\hat{\beta}$.
- This prediction η can be equipped with a confidence interval.
- To obtain a probability confidence interval, $\hat{\eta}$ can be transformed with the well-known inverse link function:

$$\hat{p} = \frac{\exp(\eta)}{1 + \exp(\eta)}$$

```
m = glm(gender~avg_sickhours, data=df, family=binomial('logit'))
pred = predict(m,newdata=data.frame(avg_sickhours=10),se=T)
(pred_ci = c(pred$fit-1.96*pred$se.fit, pred$fit+1.96*pred$se.fit) %>% ilogit())
```

```
##           1           1
## 0.4141107 0.6660099
```

Gender Estimate by Average Sickhours



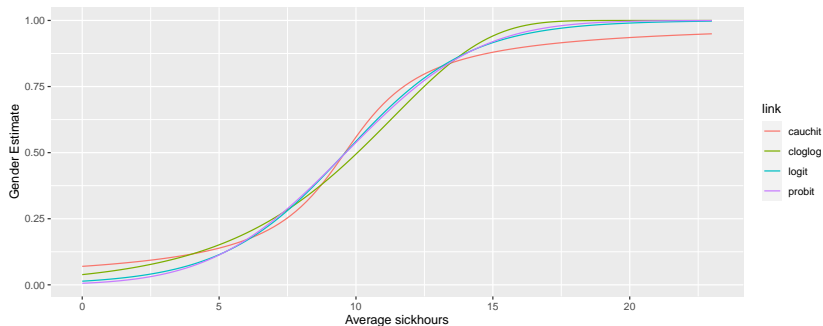
- **Alternative:** Equip the odds ratio of β with a confidence interval:
 - 95% CI: $[\exp(\hat{\beta} - 1.96\hat{\sigma}_{\beta}), \exp(\hat{\beta} + 1.96\hat{\sigma}_{\beta})]$
 - Invariant to the value of \dot{x}
 - Typically reported in clinical research papers.

```
m = glm(gender~avg_sickhours, data=df, family=binomial('logit'))
pred = predict(m,newdata=data.frame(avg_sickhours=10),se=T)
c(exp(pred$fit-1.96*pred$se.fit), exp(pred$fit+1.96*pred$se.fit))
```

```
##           1           1
## 0.7068071 1.9941010
```

Other link functions

```
mlogit    = glm(gender ~ avg_sickhours, data = df, family = binomial(link='logit'))  
mprobit   = glm(gender ~ avg_sickhours, data = df, family = binomial(link='probit'))  
mcloglog  = glm(gender ~ avg_sickhours, data = df, family = binomial(link='cloglog'))  
mcauchit  = glm(gender ~ avg_sickhours, data = df, family = binomial(link='cauchit'))
```



How to choose an appropriate link function?

- Usually, most observed data lies in the center of the distribution.
- Different link functions are typically similar in the center but differ in the **tails**.
- **Approach:** Select the link function based on *theoretical assumptions, experience, and domain expertise*.

Questions?