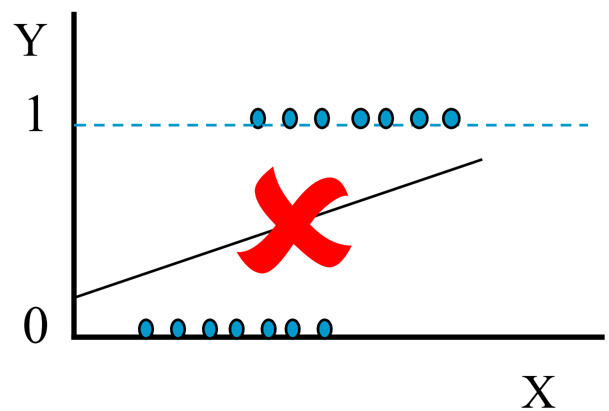
Binomial Logistic Models

KOGOD SCHOOL *of* BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC

Why not Linear Regression?

- Dependent variable is **categorical**, transformed to **binary** (e.g., 0 – no sale; 1 – sale)
- **OLS** will fit a line that **minimizes** the **SSE**. In the example below we see that higher values of X are more closely associated with a 1 than a 0, and vice versa.
- But the predicted values will be some **fraction** between 0 and 1, which is not very useful because the response can only be 0 or 1
- It is more useful to **transform** (using the **Logistic** function) the outcome variable into a **continuous** value and then use OLS

Can't fit an OLS regression line on this data without some transformation !!



What Are the Odds? Illustration

	Marketing Campaign		
Outcome	A	B	Total
No Sale (0)	2,820 (2,800)	4,180 (4,200)	**7,000**
Sale (1)	1,180 (1,200)	1,820 (1,800)	**3,000**
Total	**4,000**	**6,000**	**10,000**

Success/Failure	0.4 / 1	0.4 / 1	**3 / 7**
Chi^2			0.79
p-value			0.794

Campaign has no effect

	Marketing Campaign		
Outcome	A	B	Total
No Sale (0)	2,100 (2,800)	4,900 (4,200)	**7,000**
Sale (1)	1,900 (1,200)	1,100 (1,800)	**3,000**
Total	**4,000**	**6,000**	**10,000**

Success/Failure	0.9 / 1	0.2 / 1	**3 / 7**
Chi^2			972.22
p-value			<0.001

Campaign has effect

(Expected by chance if sales are totally independent from the campaign)

- What if you have **predictors other** than marketing campaign?
- What if you have **continuous** variable predictors?

→ Need a Binary **Logistic Regression**

Link Functions and the GLM

- A **link function** is a transformation applied to the outcome variable
- Link functions are important in predictive models to transform the outcome variable when it does not follow a normal distribution.
- For example, in a model $\mathbf{Log}(\boldsymbol{y}) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1}\mathbf{X} + \boldsymbol{\beta_1}\mathbf{X} + \cdots + \boldsymbol{\varepsilon}$ the link function is **"Log"**
- In **binomial logistic** regression, the link function is the **"Logit"** function, which we discuss in the next slide
- We fit **Logit** models with the **"Generalized Linear Model"** (GLM)
- The **GLM** is like OLS, but it can estimate any linear model:
  - Y can have **any distribution** other than **normal** (e.g., binomial, poisson, etc.)
  - Y can be transformed with various **link** functions (e.g., log, **logit**, probit)

Odds, Log Odds and the Logistic Function

- P_s = Probability of an event **success** (e.g., approving a loan)
- P_f = Probability of an event **failure** = $1 - P_s$ (e.g., declining a loan)
- **Odds** of success $= \frac{P_s}{P_f} = \frac{P_s}{1-P_s}$
- **Example:**
  - If the **probability** the Redskins **win** a game is **0.25** or 25%
  - Then the **probability** of **losing** is 1 – 0.25 = **0.75** = 75%
  - The **odds** of winning are 0.25/0.75 = 0.33 = 1/3 or **1 to 3**
- The "odds" function is a **curve** and **doesn't** generally yield a **normal distribution**, so it is customary to **log-transform** the odds
- The **log-odds** function is more linear, thus more suitable for linear regression methods
- This function is called the **"Logistic"** or **"Logit"** function:

$$\boldsymbol{Logit}(P_s) = Log\ Odds\ of\ Success = \boldsymbol{Log}\left(\frac{P_s}{1-P_s}\right)$$

The Logistic Regression Model

- The dependent variable Y can only be 0 or 1
- The regression model does **NOT** predict Y, like in **OLS** regressions
- Instead, it predicts the **Log-Odds** of Y being 1 (of **success**)
- Although you can also model the Log-Odds of Y being 0 (of **failure**)

$$Logit(Y) = \beta_0 + \beta_1(X_1) + \beta_2(X_2) + \beta_3(X_3) + \ldots\ldots + \varepsilon$$

- β_1 is the change in the **log odds** of success of Y (i.e., $= 1$) when X_1, increases by 1, keeping X_2, X_3, etc. constant.
- Same interpretation for X_2, X_3, etc.
- If X_1 is binary, β_1 is the log odds “change” when X_1 goes from 0 to 1
- If X_1 is continuous, β_1 is the log odds “change” when X_1 increases by 1

Interpretation

For a model: $Logit(Y) = \beta_0 + \beta_1(X_1) + \beta_2(X_2) + \beta_3(X_3) + \ldots\ldots + \varepsilon$

- Y is a **dummy** variable (e.g., non-smoker = 0; smoker =1)

- If X_1 is a **dummy** variable (e.g., male = 0, female = 1)
  - β_0 is the **log odds** of $Y = 1$ when $X_1 = 0$ (a male smokes)
  - e^{β_0} is the **odds** of $Y = 1$ when $X_1 = 0$ (a male smokes)
  - β_1 is the **increase** or decrease in log odds of $Y = 1$ when $X_1 = 1$ (female)
  - e^{β_1} is how much the odds of $Y = 1$ **multiply** when $X_1 = 1$ (↑ if >1; ↓ if <1)
  - $\beta_0 + \beta_1$ is the **log odds** of $Y = 1$ when $X_1 = 1$ (a female smokes)
  - $e^{(\beta_1 + \beta_2)}$ is the **odds** of $Y = 1$ when $X_1 = 1$ (a female smokes)

- If X_2 is a **continuous** variable (e.g. age)
  - β_0 is the **log odds** of $Y = 1$ when $X_2 = 0$ (just born smoker)
  - e^{β_0} is the **odds** of $Y = 1$ when $X_2 = 0$ (just born smoker)
  - β_2 is the **increase** in log odds of $Y = 1$ when $X_2 \uparrow 1$ (age increases by 1)
  - e^{β_2} is how much the odds of $Y = 1$ **multiply** when $X_2 \uparrow 1$ (↑ if >1; ↓ if <1)

Useful Pointers

- A probability of success (***Ps***) of ***50%*** is equivalent to ***1 to 1* odds** or a **log odds** of ***0*** → chance (**flip a coin**)
- If $\beta = 0$ (i.e., not significant) the predictor doesn't change the odds
- Odds < 1 → Log Odds *(-)*
  → A *(-)* β in a logistic regression
  → The **log odds** go **down**
  → The **odds change** by **less** than **1 to 1**
- Odds > 1 → Log Odds *(+)*
  → A *(+)* β in a logistic regression
  → The **log odds** go **up**
  → The **odds change** by **more** than **1 to 1**

Ps	Ps%	Odds	Log Odds
	$100 * P_s$	$\frac{P_s}{1 - P_s}$	$Ln(Odds)$
0.001	0.1%	0.001	-6.907
0.01	1%	0.010	-4.595
0.1	10%	0.111	-2.197
0.2	20%	0.250	-1.386
0.3	30%	0.429	-0.847
0.4	40%	0.667	-0.405
0.5	**50%**	**1.000**	**0.000**
0.6	60%	1.500	0.405
0.7	70%	2.333	0.847
0.8	80%	4.000	1.386
0.9	90%	9.000	2.197
1.00	100%		

Binomial Logit: Fit Statistics

- The **glm()** function in **R** reports two important statistics:
  - **Deviance** (2LL) → $-2 * \mathit{Log\ Likelihood}$
  - **AIC** → $-2 * \mathit{Log\ Likelihood} + 2 * \mathit{Number\ of\ Variables}$
- **Confusion Matrix** – Example: predicting stock price change:

$$\mathit{Error\ Rate} = \frac{\mathit{Incorrect}}{\mathit{Total}} = \frac{457 + 141}{1{,}250} = 0.48$$

$$\mathit{Sensitivity} = \frac{\mathit{True\ Positives}}{\mathit{All\ Positives}} = \frac{145}{602} = 0.24$$

$$\mathit{Specificity} = \frac{\mathit{True\ Negatives}}{\mathit{All\ Negatives}} = \frac{507}{648} = 0.78$$

Predicted	Actual		Total
	Down	Up	
Down	145	141	**286**
Up	457	507	**964**
Total	**602**	**648**	**1,250**

- Models are often evaluated based on the **analysis goals**: the model above has almost the same **error rate** as flipping a coin (no good); it's very **imprecise** about predicting **positives** (Down) but it does much **better** at predicting **negatives** (Up)

R Tips

`glm(){stats}` → "Generalized Linear Model" function in the `{stats}` package to fit various types of linear models, including binary logistic

`logit.fit=glm(y~x1+x2+etc., data=dataName, family=binomial(link="logit"))` → The `family=binomial` attribute fits a binomial model and the `link="logit"` attribute specifies the link transformation function for the dependent variable

`logLik(logit.fit)` → Get the log-likelihood

`-2*logLik(logit.fit)` → Get the residual deviance

`deviance(logit.fit)` → Should yield the same residual deviance

`AIC(logit.fit)` → Akaike Info Criterion = deviance + dimension penalty

`log.odds=(logit.fit)` → Log odd coefficients

`odds=exp(coef(logit.fit))` → Multiplicative change in odds

`prob = odds/(1+odds)` → Probabilities

`confint(logit.fit)` → 95% confidence intervals of Log-Odds coefficients

`exp(confint(logit.fit))` → 95% confidence interval of odds

Kogod School
of
Business