Regressions

MKT 566

Instructor: Davide Proserpio

What we will learn

- We continue to talk about **covariation** and learn how to model it using regressions

- We are going to cover regressions for binary outcomes, i.e., logistic regressions

- Chapter 3.6 of R for Marketing Students

Binary outcomes: Logistic regression

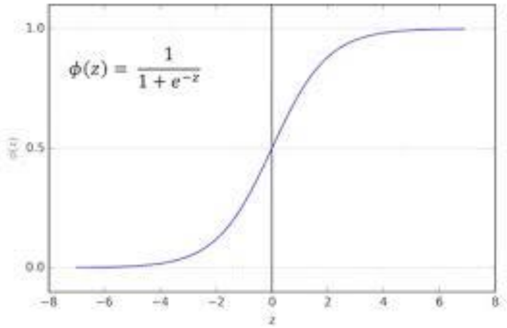
- Let's assume Y is binary, e.g.: 1 if consumer *i* buys a product, 0 otherwise
  - Linear regression can return predictions outside [0,1]

- We need a different functional form that can force predictions between [0,1]

Binary outcomes: Logistic regression

Solution: use the **logistic (sigmoid) function**, which maps any real number into [0,1]:

$$p = P(Y = 1 \mid X) = \frac{1}{1 + e^{-z}}$$

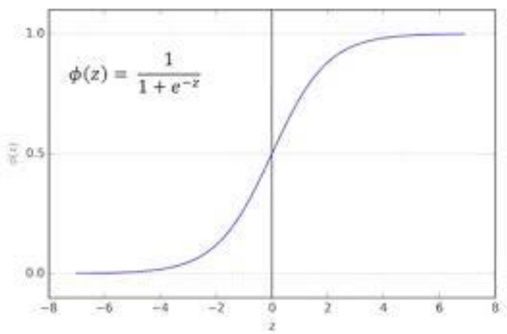


Where: $z = \beta_0 + \beta_1 X$

Binary outcomes: Logistic regression

Solution: use the **logistic (sigmoid) function**, which maps any real number into [0,1]:

$$p = P(Y = 1 \mid X) = \frac{1}{1 + e^{-z}}$$



Where: $z = \beta_0 + \beta_1 X$

Odds and Log-Odds:

- The **odds** of success are: $\frac{p}{1-p} = e^z$
- Taking the natural log gives the **log-odds (logit)**: $\log\left(\frac{p}{1-p}\right) = z = \beta_0 + \beta_1 X$
- So, the logistic regression models the **log-odds** of success as a linear function of the predictors.

Proof

1. **Write out the odds**

$$\frac{p}{1-p}=\frac{\dfrac{1}{1+e^{-x}}}{1-\dfrac{1}{1+e^{-x}}}=\frac{1}{1+e^{-x}}\Bigg/\frac{(1+e^{-x})-1}{1+e^{-x}}=\frac{1}{1+e^{-x}}\Bigg/\frac{e^{-x}}{1+e^{-x}}=\frac{1}{e^{-x}}=e^{x}.$$

2. **Take the natural log**

$$\ln\!\left(\tfrac{p}{1-p}\right)=\ln\!\left(e^{x}\right)=x.$$

So when $p=1/(1+e^{-x})$, its log-odds $\ln\big(p/(1-p)\big)$ simplifies exactly to x.

Binary outcomes: Logistic regression interpretation

- Logistic regression estimates changes in log (odds)
- To get changes in odds we can exponentiate both sides

- Given $y_i = \beta_0 + \beta_1 X_{1,i} + \epsilon_i$
  - A one unit increase in X_1 **multiplies** the odds of Y by $\boldsymbol{e^{\beta_1}}$, $i.e.$, odds change by $(e^{\beta_1} - 1) * 100\%$
  - (again, **for small $\boldsymbol{\beta_1}$ we can approximate it with $\boldsymbol{\beta_1}$*100%)**

Estimation

Unlike linear regression (which minimizes squared errors), logistic regression **maximizes the likelihood** of observing the actual outcomes, given the model.

Model fit

- R^2 does not work well with binary outcomes.
- Binary outcomes are about **yes/no decisions**, not “how much” → the **usual R^2 just doesn’t work well.**

- There are alternative measures of fit, e.g.,
  - **Pseudo- R^2**: How much better is this model at predicting 0s and 1s compared to guessing the average?
    - It's useful for checking if your logistic model is doing better than chance.
  - **Log Likelihood**: is a measure of how well the model’s predicted probabilities match the actual 0/1 outcomes.
    - The higher the better.

Logistic regression in R

```
# Crate the variable gem which identifies very good listings
airbnb[, gem:=as.integer(star_rating>=4.5 &
reviews_count>20)]
# Predict probability of being a gem using logistic
regression
m_logit = glm(gem ~ price + guests_included + city +
room_type, data = airbnb, family = binomial)
# Print results
stargazer(m_logit, type = "text",
          omit.stat = c("f", "ser", "aic", "bic"))
```

Logistic regression in R

```
# Crate the variable gem which identifies very good listings

airbnb[, gem:=as.integer(star_rating>=4.5 & reviews_count>20)]

# Predict probability of being a gem using logistic regression

m_logit = glm(gem ~ price + guests_included + city + room_type, data = airbnb, family = binomial)

# Print results

stargazer(m_logit, type = "text",
          omit.stat = c("f", "ser", "aic", "bic"))
```

```
==================================================
                          Dependent variable:
                      ----------------------------
                                  gem
--------------------------------------------------
price                          -0.002***
                               (0.0001)

guests_included                 0.112***
                                (0.008)

cityBoston                      0.167***
                                (0.040)

cityLos Angeles                 0.162***
                                (0.029)

cityMiami                      -0.195***
                                (0.042)

cityNew York City                0.188
                                (0.569)

room_typePrivate room           -0.008
                                (0.026)

room_typeShared room           -0.707***
                                (0.064)

Constant                       -1.393***
                                (0.035)

--------------------------------------------------
Observations                    50,836
Log Likelihood                -25,430.070
==================================================
```

Logistic regression in R

```r
# Crate the variable gem which identifies very good listings
airbnb[, gem:=as.integer(star_rating>=4.5 & reviews_count>20)]

# Predict probability of being a gem using logistic regression
m_logit = glm(gem ~ price + guests_included + city + room_type, data = airbnb, family = binomial)

# Print results
stargazer(m_logit, type = "text",
          omit.stat = c("f", "ser", "aic", "bic"))
```

	Dependent variable:
	gem
price	-0.002***
	(0.0001)
guests_included	0.112***
	(0.008)
cityBoston	0.167***
	(0.040)
cityLos Angeles	0.162***
	(0.029)
cityMiami	-0.195***
	(0.042)
cityNew York City	0.188
	(0.569)
room_typePrivate room	-0.008
	(0.026)
room_typeShared room	-0.707***
	(0.064)
Constant	-1.393***
	(0.035)
Observations	50,836
Log Likelihood	-25,430.070

The coefficient of **–0.002** means each **$1 increase in price decreases the odds of being a gem by about 0.2%**.