

LOGISTIC REGRESSION

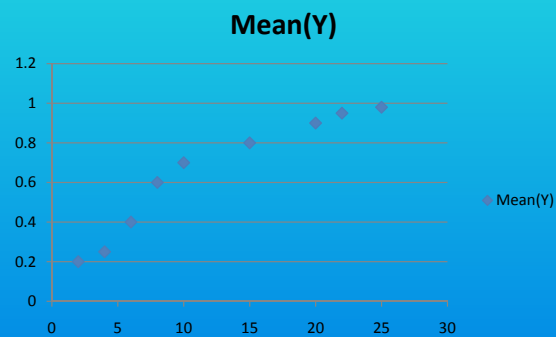
Consider a simple example:

$Y = 1$ (restaurant is a success, and 0 if restaurant fails).

Experience	N	#(Successes)	#(Failures)	Mean(Y)
2	10	2	8	0.2
4	40	10	30	0.25
6	50	20	30	0.4
8	50	30	20	0.6
10	50	35	15	0.7
15	50	40	10	0.8
20	50	45	5	0.9
22	100	95	5	0.95
25	100	98	2	0.98

NOTE:

$\text{mean}(Y) = \#(\text{successes})/N = \text{estimated probability of success } p$



$$\hat{p} = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}, 1 - \hat{p} = \frac{1}{1 + e^{\beta_0 + \beta_1 X}}$$

$$\frac{\hat{p}}{1 - \hat{p}} = e^{\beta_0 + \beta_1 X}$$

$$\ln\left(\frac{\hat{p}}{1 - \hat{p}}\right) = \beta_0 + \beta_1 X$$

i.e., log of odds ratio is linearly related to the predictor X.

In general, when there are K predictors,

$$\hat{p} = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}}, 1 - \hat{p} = \frac{1}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}}$$

$$\frac{\hat{p}}{1 - \hat{p}} = e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}$$

$$\ln\left(\frac{\hat{p}}{1 - \hat{p}}\right) = \beta_0 + \beta_1 X_1 + \dots + \beta_K X_K$$

i.e., log of odds ratio is linearly related to the predictors X_1, X_2, \dots, X_K

How are $\beta_0, \beta_1, \dots, \beta_K$ estimated in logistic regression?

Y	X1	X2	X3	X4	X5
0	1.5	0	4.1
1	2.5	0	3.8
0	4.1	1	4.1
0	7.5	0	8.7
1	2.3	1	1.4
1	1.9	1	3.2
0	8.1	0	11.3
1	1.1	0	0.4
...
...
0	10.2	1	13.7
0	11.6	0	7.9
1	7.3	1	5.4

Instead of the LS method, the method of Maximum Likelihood is used.

The probability of observing the sample (Y_1, Y_2, \dots, Y_N)

can be expressed as:

$$P(Y=0) P(Y=1) P(Y=1) \dots P(Y=0)P(Y=1)$$

$$= \prod [p^Y (1-p)^{1-Y}]$$

$$= \text{Likelihood Function } L(p; Y_1, \dots, Y_N)$$

$$\log L(p; Y_1, \dots, Y_N) = H(\beta_0, \beta_1, \dots, \beta_K; L(p; Y_1, \dots, Y_N))$$

The method of maximum likelihood finds the values of $\beta_0, \beta_1, \dots, \beta_K$ which maximize the log-likelihood (computer search method is used to find the maximum).

Output from Logistic Regression:

Deviance = $-2 * \log\text{-likelihood}$

Akaike's Information Criterion (AIC) = $-2 * \log\text{-likelihood} + 2 M$

$M = K + 1 = \#(\text{parameters estimated})$

Bayesian Information Criterion (BIC) = $-2 * \log\text{-likelihood} + M \log(N)$

A model with small AIC or BIC is preferred.

$$p = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_K X_K)}}$$

If $\beta_1 > 0$ and X_1 large, then the denominator will be small, which will make p large. If $\beta_1 < 0$ and X_1 large, then the denominator will be large, which will make p small. Hence a negative β_j implies that p is negatively associated with X_j , and a positive β_j implies that p is positively associated with X_j .

Example:

```
book <- read.csv("M:/DataMining/Data/Charles_BookClub.csv",
header=TRUE)
```

```
book.out <- glm(Florence ~
Gender+M+R+F+FirstPurch+ChildBks+YouthBks+CookBks+DoltYBks+RefBks+ArtBks
+GeogBks+ItalCook+ItalHAtlas+ItalArt, family=binomial("logit"))
```

```
summary(book.out)
```

```
glm(formula = Florence ~ Gender + M + R + F + FirstPurch +
ChildBks +
YouthBks + CookBks + DoltYBks + RefBks + ArtBks + GeogBks +
ItalCook + ItalHAtlas + ItalArt, family = binomial("logit"))
```

Deviance Residuals:

```
Min    1Q  Median    3Q    Max
-1.8759 -0.4774 -0.3311 -0.1971  2.9345
```

Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.3524835	0.3096589	-4.368 1.26e-05 ***
Gender	-0.8833497	0.1613001	-5.476 4.34e-08 ***
M	0.0003501	0.0009195	0.381 0.703404
R	-0.0892098	0.0187102	-4.768 1.86e-06 ***
F	0.2711517	0.0921058	2.944 0.003241 **
FirstPurch	0.0138329	0.0116336	1.189 0.234419
ChildBks	-0.5059118	0.1367719	-3.699 0.000216 ***
YouthBks	-0.6457685	0.1889838	-3.417 0.000633 ***
CookBks	-0.6442635	0.1468503	-4.387 1.15e-05 ***
DoltYBks	-0.7815103	0.1742908	-4.484 7.33e-06 ***
RefBks	0.0171449	0.1790575	0.096 0.923719
ArtBks	0.6449422	0.1545865	4.172 3.02e-05 ***
GeogBks	0.1943929	0.1428007	1.361 0.173423
ItalCook	0.3067418	0.2598205	1.181 0.237765
ItalHAtlas	0.0909080	0.3998221	0.227 0.820135
ItalArt	0.3923821	0.3372949	1.163 0.244699

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1373.5 on 1999 degrees of freedom
Residual deviance: 1120.9 on 1984 degrees of freedom
(2 observations deleted due to missingness)
AIC: 1152.9 = Residual Deviance + 2 M = 1120.9 + 32

Number of Fisher Scoring iterations: 6

Null deviance = deviance with ONLY the intercept term in the logistic regression model

Null deviance - Residual deviance $\sim \chi^2$ distribution with $df = (K+1)$

```
> book1.out <- glm(Florence ~  
Gender+R+F+ChildBks+YouthBks+CookBks+DoltYBks+ArtBks,  
family=binomial("logit"))  
> summary(book1.out)
```

Call:

```
glm(formula = Florence ~ Gender + R + F + ChildBks + YouthBks +  
CookBks + DoltYBks + ArtBks, family = binomial("logit"))
```

Deviance Residuals:

```
Min    1Q  Median    3Q    Max  
-1.7523 -0.4749 -0.3344 -0.2028  2.9177
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.52130	0.22876	-6.650	2.93e-11	***
Gender	-0.88362	0.16032	-5.512	3.56e-08	***
R	-0.06604	0.01414	-4.671	2.99e-06	***
F	0.40397	0.05851	6.904	5.04e-12	***
ChildBks	-0.54516	0.12149	-4.487	7.21e-06	***
YouthBks	-0.68522	0.17583	-3.897	9.74e-05	***
CookBks	-0.62355	0.12153	-5.131	2.88e-07	***
DoltYBks	-0.81585	0.15806	-5.162	2.45e-07	***
ArtBks	0.63083	0.13085	4.821	1.43e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1373.5 on 1999 degrees of freedom
Residual deviance: 1131.0 on 1991 degrees of freedom
(2 observations deleted due to missingness)
AIC: 1149.0

Number of Fisher Scoring iterations: 6

More on Logistic Regression

In this lecture –

- We will compute the confusion matrix for logistic regression example from last lecture
- Learn how to split a data set into training and test sets, then build a logistic model on the training set using variable selection via AIC, and predict $P(Y=1)$ for the test set
- Compute confusion matrix and classification rates for both training and test sets
- Do another logistic regression example.

R-libraries/R-functions used in this lecture

1. xtabs for computing confusion matrix (slide 7)
2. Split data set into a training set and a test set (slide 9)
3. Variable selection in logistic regression using AIC using the R-function stepAIC of library MASS (slide 10)
4. Calculating classification results for training and test sets (slides 13 – 14)

One way to assess the performance of logistic regression on a data set is to look at the CONFUSION MATRIX, which contains information about actual and predicted values.

Logistic regression outputs values of estimated $P(Y=1)$, and the following rule is used to predict classification into 0 or 1:

Estimated $P(Y=1) > 0.5$, predict $Y = 1$, else predict $Y = 0$.

		PREDICTED	
		0	1
OBSERVED	0	a	b
	1	c	d

a = # of correct predictions of response 0

d = # of correct predictions of response 1

b = # of incorrect predictions of response 0

c = # of incorrect predictions of response 1

Missclassification Rate = $(b+c)/(a+b+c+d)$

Accuracy of prediction = AC = $(a+d)/(a+b+c+d)$

Recall or true positive rate TP = $d/(c+d)$ = correctly classified 1's

False positive rate FP = $b/(a+b)$

True negative rate TN = $a/(a+b)$

False negative rate FN = $c/(c+d)$

Precision P = $d/(b+d)$ = proportion of predicted 1's that were correct

We will next compute the confusion matrix of a logistic regression result.

Example 1 (Charles Book Club Example – revisited)

(continued from – Advanced Regression with R.pptx)

Read the data file Charles_BookClub.csv, and attach the data frame, then run the following logistic regression model.

```
book.out <- glm(Florence ~
Gender+M+R+F+FirstPurch+ChildBks+YouthBks+CookBks+DoItYBks
+RefBks+ArtBks+GeogBks+ItalCook+ItalHAtlas+ItalArt,
family=binomial("logit"))

summary(book.out)
```

```
names(book.out)
[1] "coefficients" "residuals" "fitted.values" "effects"
"R" "rank"
[7] "qr" "family" "linear.predictors" "deviance"
"aic" "null.deviance"
[13] "iter" "weights" "prior.weights" "df.residual"
"df.null" "y"
[19] "converged" "boundary" "model" "na.action"
"call" "formula"
[25] "terms" "data" "offset" "control"
"method" "contrasts"
[31] "xlevels"
```

```
# there are 2 NA values in Y=Florence column (cases 2001 and 2002)
# delete these two NA values
```

```
observed <- Florence[1:2000]
fitted <- round(book.out$fitted.values)
xtabs(~observed+fitted)
```

	fitted		
observed	0	1	
0	1764	19	FP = $19/(1764+19) = .01$
1	189	28	FN = $189/(189+28) = .87$

Overall Misclassification Rate = $(189+19)/(1764+189+19+28)$
= $208/2000 = .10 = 1 - AC$

It is good practice to split a data set into a training set and a test set, then

Build model on the training set
Predict the test set

If the misclassification rate is low on both training and test sets, then the fitted model can be considered satisfactory.

We will use the Charles_BookClub.csv data set to illustrate the above method.

Continued from R session (slide #7) – find # of cases in the data file Charles_BokokClub.csv

```
length(Florence)
[1] 2002
```

We will split the data file so that training set has 75% of the data and test set has 25% of the data. Getting back to R:

```
set.seed(10245)
holdout <- sample(1:nrow(book), 500, replace = FALSE)
# Now split into training and test sets
book.train <- book[-holdout, ]
book.test <- book[holdout, ]
```

```
# Fit logistic regression model to training data, predict on test data
lrm2 <- glm(Florence ~
Gender+M+R+F+FirstPurch+ChildBks+YouthBks+CookBks+DoltYBks+
RefBks+ArtBks+GeogBks+ItalCook+ItalHAtlas+ItalArt,data =
book.train, family=binomial("logit"))
```

```
# use library MASS to perform variable selection using AIC criterion
library(MASS)
stepAIC(lrm2)
# voluminous output
```

```
Call: glm(formula = Florence ~ Gender + R + F + FirstPurch +
ChildBks + YouthBks + CookBks + DoltYBks + ArtBks +
ItalCook, family = binomial("logit"), data = book.train)
```

Coefficients:

(Intercept)	Gender	R	F	FirstPurch	ChildBks
-1.04478	-1.06197	-0.11639	0.19435	0.02972	-
0.45224	-0.67728				
CookBks	DoltYBks	ArtBks	ItalCook		
-0.64874	-0.77926	0.71032	0.62623		

Degrees of Freedom: 1499 Total (i.e. Null); 1489 Residual
(2 observations deleted due to missingness)

Null Deviance: 984

Residual Deviance: 799.2 AIC: 821.2

```
# Final model
```

```
lrm2a <- glm(formula = Florence ~ Gender + R + F + FirstPurch +
ChildBks + YouthBks + CookBks + DoltYBks + ArtBks + ItalCook,
family = binomial("logit"), data = book.train)
```

```
# use the final model built on training data to predict test data
```

```
lrm3tst <- predict(lrm2a, book.test, type = 'response')
```

```
# calculate correct confusion matrix and classification percentages
```

```
# training set
```

```
observed.train <- book.train$Florence[1:1500]
```

```
fitted.train <- lrm2a$fitted.values
```

```
xtabs(~observed.train+fitted.train)
```

	fitted.train	
observed.train	0	1
0	1340	8
1	136	16

```
correct_classification.train <- (1340+16)/(1340+136+8+16)
```

```
correct_classification.train
```

```
[1] 0.904
```

```
observed.test <- book.test$Florence
```

```
fitted.test <- round(lrm3tst)
```

```
xtabs(~observed.test + fitted.test)
```

	fitted.test	
observed.test	0	1
0	433	2
1	56	9

```
correct_classification.test <- (433+9)/(433+2+56+9)
```

```
correct_classification.test
```

```
[1] 0.884
```

Example 2: Wetland Classification

Data set has 26 variables, with

DV = V26 = binary (wetland classification)

V1, V2 = spatial coordinates of sample location

Data set had over 6 million lines – 2 random samples, each of size 10000, was taken from this large data set.

```
wetland <- read.csv("G:/hc1a.csv",header=TRUE)
names(wetland)
attach(wetland)
```

```
wetland.out <- glm(V26 ~
V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16
+V17+V18+V19+V20+V21+V22+V23+V24,
family=binomial("logit"))
```

```
summary(wetland.out)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.4971485	0.3698332	-17.568	< 2e-16 ***
V3	0.9040565	0.3508822	2.577	0.009980 **
V4	-0.0065268	0.0018371	-3.553	0.000381 ***
V5	0.7048252	0.0754529	9.341	< 2e-16 ***
V6	0.0040051	0.0006381	6.276	3.47e-10 ***
V7	-0.0358227	0.1856778	-0.193	0.847014
V8	0.1506237	0.0217626	6.921	4.48e-12 ***
V9	0.5306576	0.3440593	1.542	0.122990
V10	0.2438647	0.1830803	1.332	0.182857
V11	0.0429113	0.0438971	0.978	0.328300
V12	0.0967733	0.5052403	0.192	0.848103
V13	0.3521818	0.3441767	1.023	0.306186
V14	-0.0804089	0.0421495	-1.908	0.056429 .
V15	0.0029327	0.0014124	2.076	0.037860 *
V16	0.0015271	0.0002880	5.302	1.15e-07 ***
V17	-0.0045502	0.0009866	-4.612	3.99e-06 ***
V18	-0.0046850	0.0014881	-3.148	0.001642 **
V19	-0.0582260	0.0273372	-2.130	0.033178 *
V20	0.2076367	0.0488683	4.249	2.15e-05 ***
V21	-0.1248815	0.0358366	-3.485	0.000493 ***
V22	-0.0021921	0.0052871	-0.415	0.678430
V23	-0.0124892	0.0127656	-0.978	0.327899
V24	0.0286177	0.0219352	1.305	0.192013

Signif. codes: 0 '***' 0.001
'**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1624.2 on 9999 degrees of freedom

Residual deviance: 1202.0 on 9977 degrees of freedom

AIC: 1248.0

```
observed <- V26
fitted <- round(wet.out$fitted.values)
```

```
xtabs(~observed+fitted)
      fitted
observed 0    1
      0  9829 13
      1   143 15
```

```
FP <- 13/(13+9829)
FN <- 143/(143+15)
```

```
FP
0.001320870
FN
0.9050633
```

Overall
misclassification =
(143+13)/10000
= 0.0156

NOTE: this data set has very small % of 1's:

```
#(1) = 158, n = 10000
P(Y=1) = .0158
```

Coefficients:

```
Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.9816131 0.2855980 -20.944 < 2e-16 ***
V3 0.8378952 0.3267936 2.564 0.010348 *
V4 -0.0054307 0.0015450 -3.515 0.000440 ***
V5 0.6248313 0.0765550 8.162 3.30e-16 ***
V6 0.0052916 0.0005910 8.953 < 2e-16 ***
V7 0.2450584 0.1751301 1.399 0.161725
V8 0.1267578 0.0199955 6.339 2.31e-10 ***
V9 -0.0885989 0.3772810 -0.235 0.814337
V10 0.5496452 0.1747053 3.146 0.001654 **
V11 0.0139899 0.0525293 0.266 0.789988
V12 0.1472172 0.4457888 0.330 0.741219
V13 0.9272693 0.3180951 2.915 0.003556 **
V14 -0.1695692 0.0469682 -3.610 0.000306 ***
V15 0.0030291 0.0012677 2.390 0.016870 *
V16 0.0015513 0.0003020 5.137 2.79e-07 ***
V17 -0.0041240 0.0009928 -4.154 3.27e-05 ***
V18 -0.0081692 0.0014183 -5.760 8.42e-09 ***
V19 -0.0262453 0.0300195 -0.874 0.381967
V20 0.2030188 0.0485512 4.182 2.90e-05 ***
V21 -0.1545541 0.0365230 -4.232 2.32e-05 ***
V22 -0.0084037 0.0050864 -1.652 0.098498 .
V23 0.0082330 0.0127746 0.644 0.519265
V24 0.0027807 0.0227944 0.122 0.902905
```

```
observed <- V26
fitted <-
round(wetlandb.out$fitted.values)
xtabs(~observed+fitted)
```

```
      fitted
observed 0  1
      0 9809 12
      1  166 13
```

```
overall_misclassification
<- (166+12)/10000
>
overall_misclassification
[1] 0.0178
```