Intermediate Biostatistics for Medical Researchers

Robert Goldman
Professor of Statistics
Simmons College

Thursday, April 19, 2018

Introduction to Logistic Regression

Thus far we have looked at regression models in which the response variable is *quantitative* and the explanatory variables are a mixture of quantitative and qualitative.

Now we look at models in which the response variable is *qualitative* and binary and the explanatory variables are, again, a mixture of quantitative and qualitative.

In this context, the response variable, Y might be (i) whether or not a patient survives a procedure, (ii) Whether an infant is low birth-weight or not, or (iii) whether or not a patient can return home or go on to long-term care following rehabilitation.

When the response variable is qualitative with just two categories a frequently used technique is called **logistic regression**.

Uses for Logistic Regression

Logistic regression can be used:

- to create a prediction rule for assigning individuals to one of two groups.
- and to identify ‘risk’ factors that affect the likelihood of an outcome.
- to remove the effect of confounding variables in observational studies in which the response is binary.
- to create *propensity scores*. These scores are used in observational studies as estimates of the probabilities that each participant would choose/receive the experimental treatment.

The Burn data

SOURCE: Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X. (2013) Applied Logistic Regression: Third Edition. These data are copyrighted by John Wiley & Sons Inc.

Hospital Discharge Status	0 = Alive 1 = Dead	Death
Age at admission	Years	Age
Gender	0 = Female 1 = Male	Gender
Race	0 = Non-White 1 = White	Race
Total burn surface area	0 - 100%	TBSA
Burn involved inhalation injury	1 = Yes 0 = No	INH
Flame involved in burn injury	1 = Yes 0 = No	Flame

```
head(burn)

  Death   Age Gender  Race  TBSA INH_INJ Flame
1     0 26.6      1     1 25.3       0     1
2     0  2.00     0     0  5.00      0     0
3     0 22.0      0     0  2.00      0     0
4     0 37.3      1     1  2.00      0     0
5     0 52.1      1     1  6.00      0     1
6     0 50.2      1     1  7.00      0     0


tail(burn)

  Death   Age Gender  Race  TBSA INH_INJ Flame
1     1  83.7     0     1 50.5       0     0
2     1  34.2     1     1 91.0       1     1
3     1  59.0     1     1 37.5       1     1
4     1  85.5     1     1  4.60      1     1
5     1  46.8     1     0 47.0       1     1
6     1  40.8     1     1  1.20      1     1
```

In this case we shall construct models that relate whether or not a person will die to (i) Flame, (ii) TBSA, and (iii) Flame and TBSA, and finally, to all the available predictors.

In this case, the response variable (Y) can take two values (1 or 0)

Why does linear regression not work in this case?

```
model <- lm(Death ~ TBSA, burn)
model

Call:
lm(formula = Death ~ TBSA, data = burn)

Coefficients:
(Intercept)         TBSA
  -0.009719     0.011792
```

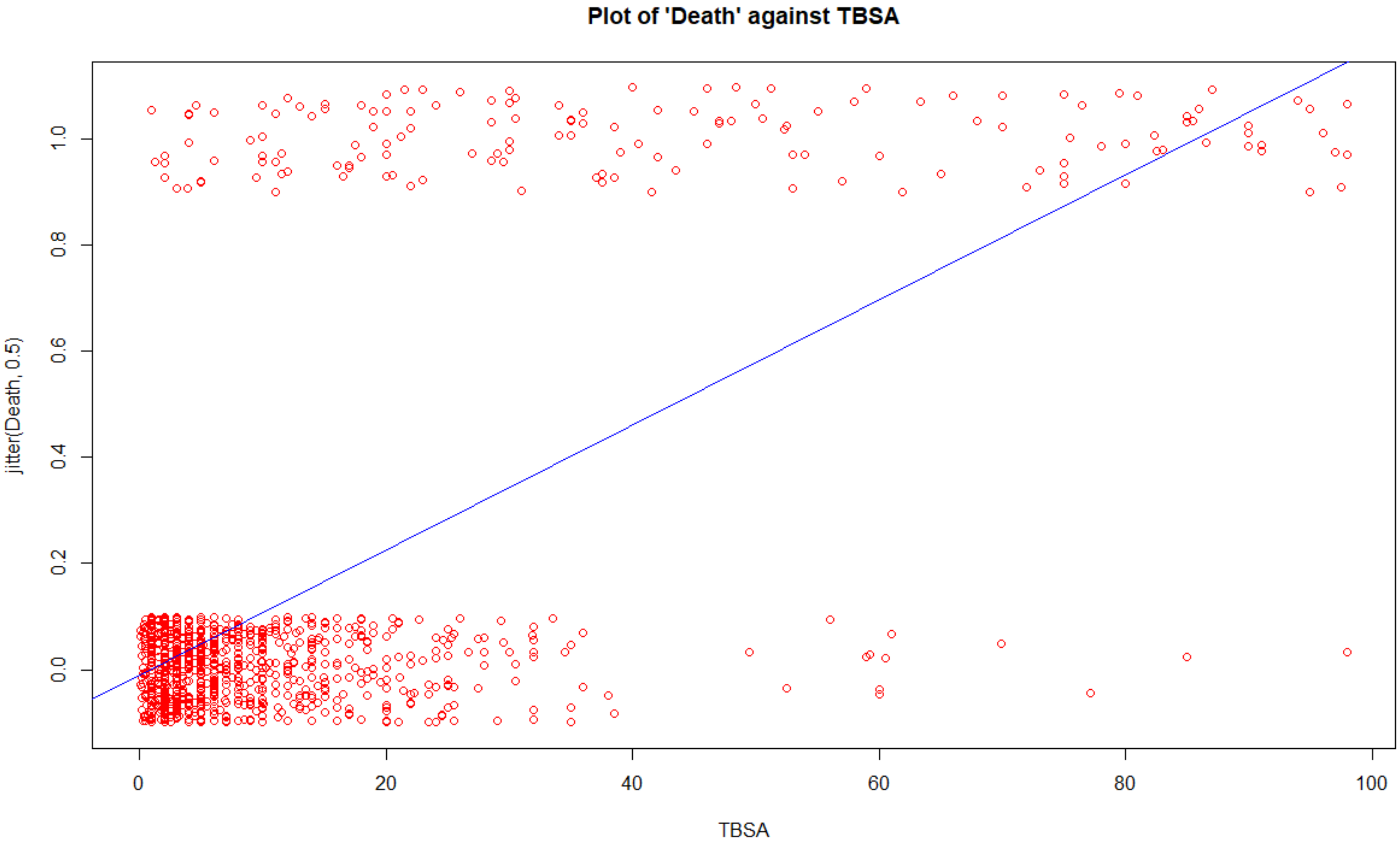
Death = -0.00972 + 0.01179TBSA

When TBSA = 50% Predicted Death = 0.5798

When TBSA = 0.1% Predicted Death = -0.0085

When TBSA = 99% Predicted Death = 1.157

```
plot(jitter(Death, 0.5) ~ TBSA, data = burn,
  col = "red",
  main = "Plot of 'Death' against TBSA")
abline(model, col = "blue")
```



Some Preliminary Analyses

```
tally(~Death, data = burn)
Death
  0   1
850 150
```

```
tally(Death ~ Gender, data = burn)
     Gender
Death   0   1
    0 246 604
    1  49 101
tally(Death ~ Gender, data = burn,
 format = "percent")
     Gender
Death        0        1
    0 83.38983 85.67376
    1 16.61017 14.32624
```

		Female	Male	All
Death	No	246	604	850
	Yes	49 (16.6%)	101 (14.3%)	150 (15%)
	All	295	705	1000

Race

		Non-White	White	All
Death	No	356	494	850
	Yes	55 (13.4%)	95 (16.1%)	150 (15%)
	All	411	589	1000

INH_INJ

		No	Yes	All
Death	No	800	50	850
	Yes	78 (8.9%)	72 (59.0%)	150 (15%)
	All	878	122	1000

Flame

		No	Yes	All
Death	No	451	399	850
	Yes	20 (4.2%)	130 (24.6%)	150 (15%)
	All	471	529	1000

Flame

		No	Yes	All
	No	451	399	850
Death				
	Yes	20 (4.2%)	130 (24.6%)	150 (15%)
	All	471	529	1000

$$\hat{p}_N = \frac{20}{471} = 0.04246$$

$$\hat{p}_Y = \frac{130}{529} = 0.24575$$

$$\hat{O}_N = \frac{20}{451} = 0.04435 \qquad \hat{O}_Y = \frac{130}{399} = 0.32581$$

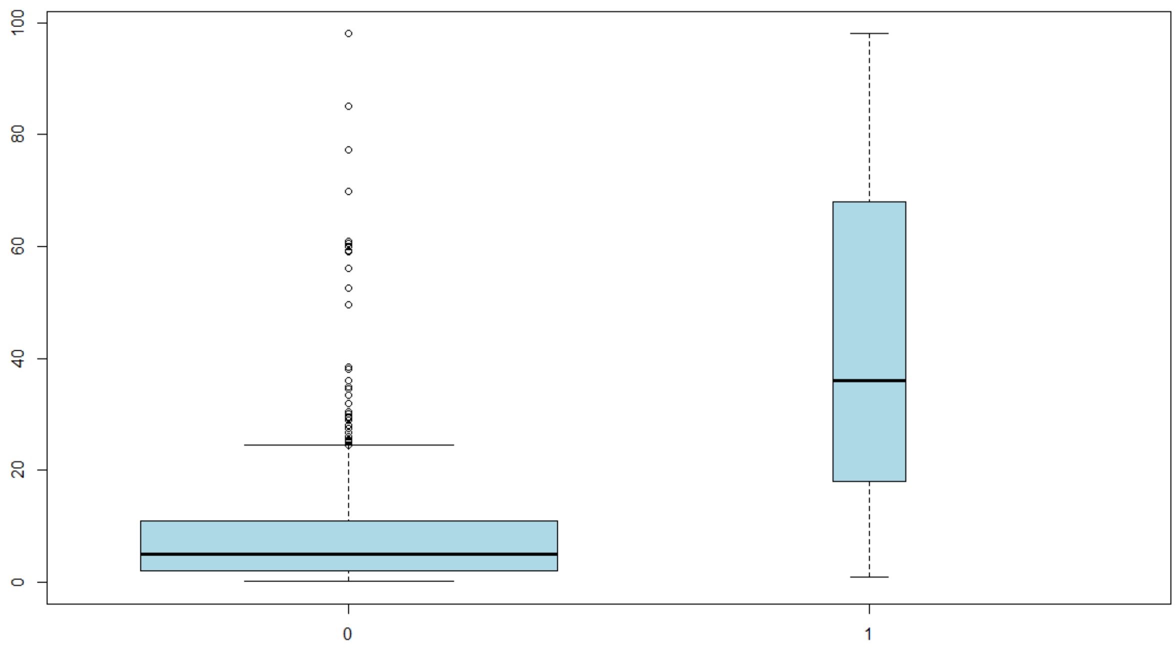
$$\widehat{OR} = 0.32581/0.04435 = 7.346.$$

Where a flame is involved, the burn victim's odds of death is 7.3 times the odds when a flame is not involved.

```
mean(TBSA ~ Death, data = burn)
        0         1
 8.504588 42.106000

median(TBSA ~ Death, data = burn)
 0  1
 5 36
```

```
proportion <- tally(~Death, data = burn)/1000
boxplot(TBSA ~ Death, data = burn,
 width = proportion,
 col = "lightblue")
```



1. Descriptive Aspects of Logistic Regression

The Simple Logistic Regression Model

Logistic regression models enable us to predict not Y but rather, the quantity p = P(Y = 1), the probability that a person will take the value Y = 1, as a function of the X variable(s). The simple logistic regression model is

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Here, e = 2.718… is the base of natural logarithms.

The quantity

$$e^{\beta_0 + \beta_1 X}$$

must always be positive and can vary from 0 up to infinity. As a consequence

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

must always lie between 0 and 1.

In simple linear regression (and multiple linear regression), statistical software uses the procedure called least squares to obtain, from the data,the ‘best’ values for the regression coefficients.

In the context of logistic regression, the software uses, not least squares, but a procedure called Maximum Likelihood Estimation to find the 'best' values for b_0 and b_1 from our data. The method seeks to find the values

$b_0 = \hat{\beta}_0$ and $b_1 = \hat{\beta}_1$

which are ‘most likely’ to have generated the sample of zeros or ones.

There are three ways to write the fitted model:

1. $\widehat{P(Y=1)} = \hat{p} = \dfrac{e^{b0 + b1X}}{1 + e^{b0 + b1X}}$

This is an expression for the predicted probability that Y = 1.

2. $\dfrac{\hat{p}}{1 - \hat{p}} = \hat{O} = e^{b0 + b1X} = \text{Exp}(b_0 + b_1X)$

This is an expression for the predicted odds that Y = 1.

3. $\hat{L} = \ln\left(\dfrac{\hat{p}}{1 - \hat{p}}\right) = b_0 + b_1X$

This is an expression for the predicted log odds that Y = 1.

Logistic Regression when X is also 0/1

Here is the 'coefficients' output for a logistic regression when Flame is the explanatory variable.

```
model <- glm(Death ~ Flame,
 family = binomial,
 data = burn)
model


Coefficients:
(Intercept)        Flame
     -3.116        1.994
```

$$\widehat{P(Y=1)} = \hat{p} = \frac{e^{-3.116 + 1.994\text{Flame}}}{1 + e^{-3.116 + 1.994\text{Flame}}}$$

$$\hat{O} = e^{-3.116 + 1.994\text{Flame}}$$

$$\hat{L} = -3.116 + 1.994\ \text{Flame}$$

"No Flame" $\widehat{P(Y=1)} = \frac{e^{-3.116 + 1.994(0)}}{1+e^{-3.116 + 1.994(0)}} = 0.04245$

"Flame" $\widehat{P(Y=1)} = \frac{e^{-3.116 + 1.994(1)}}{1+e^{-3.116 + 1.994(1)}} = 0.24575$

These are the sample proportions we found earlier.

"No Flame" $\hat{O} = e^{-3.116 + 1.994(0)} = 0.04435$

"Flame" $\hat{O} = e^{-3.116 + 1.994(1)} = 0.32581$

These are the sample odds we found earlier.

When we have a 0/1 variable as the only explanatory variable, logistic regression returns predictions equal to the sample proportions and odds.

An important result!

X is a variable that takes values 0 or 1

The odds that Y = 1 $= e^{b0 + b1X}$

The odds ratio, $\widehat{\text{OR}} = \dfrac{\text{odds that } Y = 1 \text{ when } X = 1}{\text{odds that } Y = 1 \text{ when } X = 0}$

$$= \frac{e^{b0 + b1(1)}}{e^{b0 + b1(0)}}$$

$$= e^{b0 + b1 - b0} = e^{b1}$$

For our example $\widehat{\text{OR}} = e^{b1} = e^{1.994} = 7.346$

Logistic Regression When the Explanatory Variable is Quantitative (TBSA)

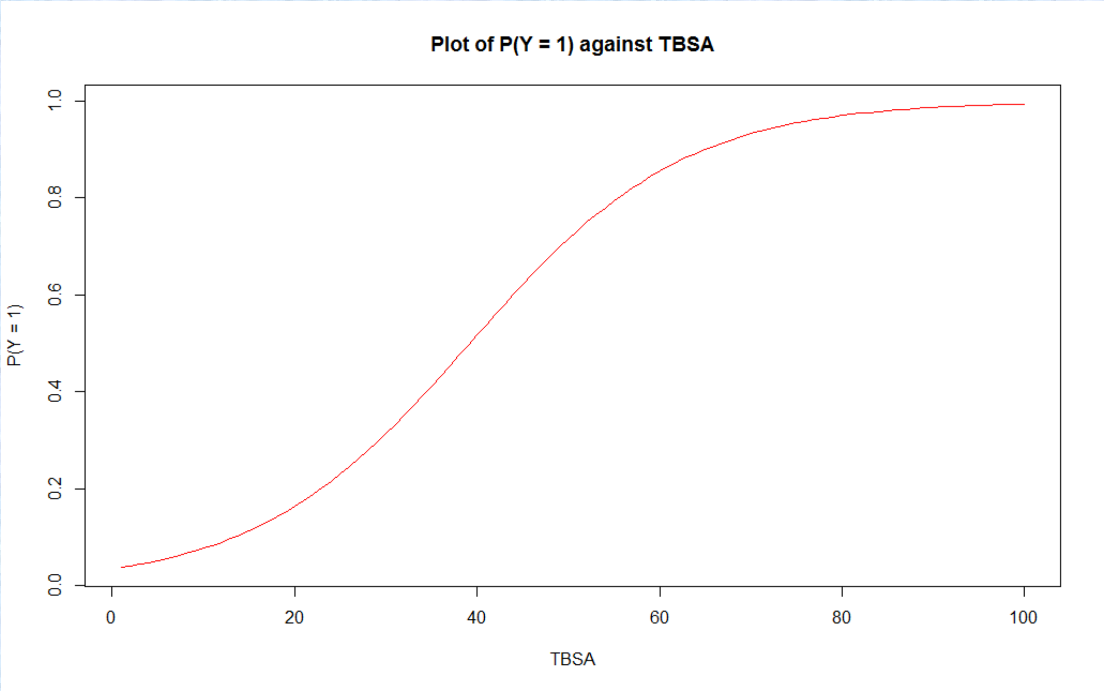
```
model <- glm(Death ~ TBSA, binomial, burn)
model
```

$$P(\widehat{Y = 1}) = \hat{p} = \frac{e^{-3.34511 + 0.08537TBSA}}{1 + e^{-3.34511 + 0.08537TBSA}}$$

TBSA	$P(\widehat{Y = 1})$
1%	0.036978
20%	0.162777
50%	0.715732
80%	0.970243
99%	0.993979

```
x <- seq(1,100)
z <- exp(-3.34511 + 0.08537*x)
y <- z/(1 + z)
plot(y ~ x, col = "red", type = "l",
     main = "Plot of P(Y = 1) against TBSA",
     xlab = "TBSA",
     ylab = "P(Y = 1)")
```

(The notation type = “l” connects the dots and omits the symbols.)



$\hat{O}$ = odds that Y= 1 = $e^{-3.34511 - 0.08537TBSA}$

The predicted odds that a patient with TBSA of 20% will die is

$e^{-3.34511 - 0.08537(20)}$ = 0.162777

The predicted odds that a patient with TBSA of 80% will die is

$e^{-3.34511 - 0.08537(80)}$ = 0.970243

Earlier, we noted that when X is a 0/1 variable

$$\widehat{OR} = \frac{\text{odds that } Y=1 \text{ when } X=1}{\text{odds that } Y=1 \text{ when } X=0} = e^{b1}$$

Does e^{b1} have any similar interpretation when X is quantitative?

Yes!

$$e^{b1} = \frac{\text{odds that } Y=1 \text{ for } X}{\text{odds that } Y=1 \text{ for } X-1}$$

For our example, $b_1 = 0.08537$

So $e^{b1} = e^{-0.08537} = 1.08912$

For each additional 1% in TBSA, the predicted odds of dying change by a factor of 1.09.

In logistic regression where X is quantitative, e^{b1} is the factor by which the odds of Y = 1 change as X increases by one unit. In other words, e^{b1} is the odds (that Y = 1) ratio associated with being X as opposed to X - 1.

The odds of a patient with a TBSA of 21 dying is 1.08912 times the corresponding odds for a patient with a TBSA of 20.

The odds of a patient with a TBSA of 81 dying is 1.08912 times the corresponding odds for a patient with a TBSA of 80.

$$\frac{\text{Odds of dying with TBSA of 36}}{\text{Odds of dying with TBSA of 26}} =$$

$$\frac{\text{Odds of dying with TBSA of 26}}{\text{Odds of dying with TBSA of 36}} =$$

Classification Tables

The following code will assign a 1 if P(Y = 1) > 0.5 and a 0 if P(Y = 1) < 0.5 to preddeath.

```
model <- glm(Death ~ TBSA, binomial,
         burn)
fit <- fitted(model)
# gives predicted probabilities

preddeath <- rep(0, 1000)
preddeath[fit >= 0.5] <- 1

tally(preddeath ~ burn$Death,
 format = "percent")

        burn$Death
preddeath         0         1
        0 98.470588 54.666667
        1  1.529412 45.333333
```

		Actual Death		
		No	Yes	All
Predicted Death	No	837 (98.5%)	82	919
	Yes	13	68 (45.3%)	81
	All	850	150	1000

Death: p >0.5

		Predicted No	Death Yes	All
Death?	No	837 (98.5%)	13	850
	Yes	82	68 (45.3%)	150
	All	919	81	1000

Death: p > 0.4

		Predicted No	Death Yes	All
Death?	No	829 (97.5%)	21	850
	Yes	71	79 (52.7.3%)	150
	All	900	100	1000

TBSA + Flame

```
model2 <- glm(Death ~ TBSA + Flame,
          binomial, burn)

model2

Coefficients:
(Intercept)         TBSA        Flame
   -4.10581      0.07812      1.26716
```

$$\widehat{P(Y=1)} = \hat{p} = \frac{e^{-4.105814 + 0.078119\text{TBSA} + 1.267158\text{Flame}}}{1 + e^{-4.105814 + 0.078119\text{TBSA} + 1.267158\text{Flame}}}$$

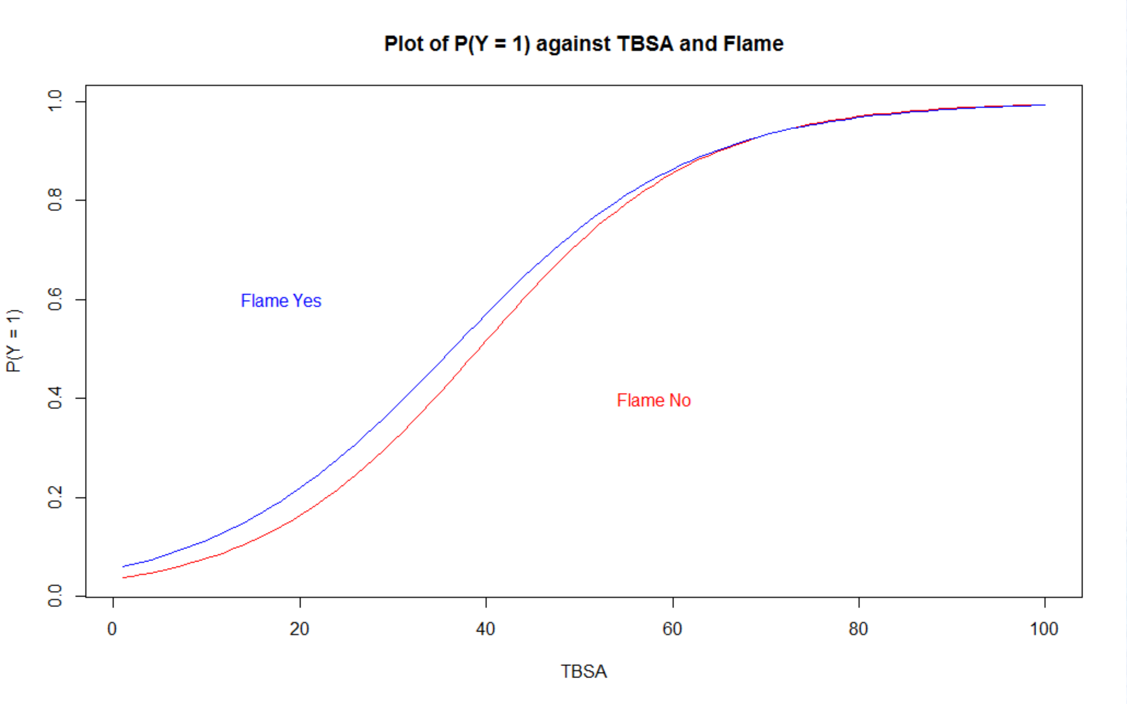
$b_1 = 0.07812 \quad e^{b1} = e^{0.07812} = 1.0813$

Adj_OR for TBSA = 1.0813

$b_2 = 1.26716 \quad e^{b2} = e^{1.26716} = 3.5508$

Adj_OR for Flame = 3.5508

```
x <- seq(1,100)
z1 <-  exp(-4.105814 + 0.078119*x)
y1 <- z1/(z1  + z1)
z2 <- exp(-2.838664 + 0.078119*x)
y2 <- z2/(1 + z2)
plot(y ~ x, col = "red", type = "l",
      main = "Plot of P(Y = 1) against TBSA and Flame",
      xlab = "TBSA",
      ylab = "P(Y = 1)")
lines(x, y2, col = "blue")
text(18, 0.6, "Flame Yes", col = "blue")
text(58, 0.4, "Flame No", col = "red")
```



For the burn data, this is the ‘best’ model

```
model11 <- glm(Death ~ Age  + Race + TBSA + INH_INJ +
Age:INH_INJ, binomial, burn)
```

Classification Table

		Actual Death		
		No	Yes	All
Predicted Death?	No	824 (96.9%)	47	871
	Yes	26	103 (68.7%)	129
	All	850	150	1000

$$\text{sensitivity} = P(\hat{Y} = 1 \mid Y = 1) = 0.687$$

= proportion of deaths that are correctly identified as deaths.

$$\text{specificity} = P(\hat{Y} = 0 \mid Y = 0) = 0.969$$

= proportion of survives that are correctly identified as survives.

For the burn data, this is the ‘best’ model

```
model11 <- glm(Death ~ Age  + Race + TBSA + INH_INJ +
Age:INH_INJ, binomial, burn)
```

Classification Table

		Actual Death: No	Actual Death: Yes	All
Predicted Death?	No	824 (96.9%)	47	871
	Yes	26	103 (68.7%)	129
	All	850	150	1000

$$\text{sensitivity} = P(\hat{Y} = 1 \mid Y = 1) = 0.687$$

= proportion of deaths that are correctly identified as deaths.

$$\text{specificity} = P(\hat{Y} = 0 \mid Y = 0) = 0.969$$

= proportion of survives that are correctly identified as survives.

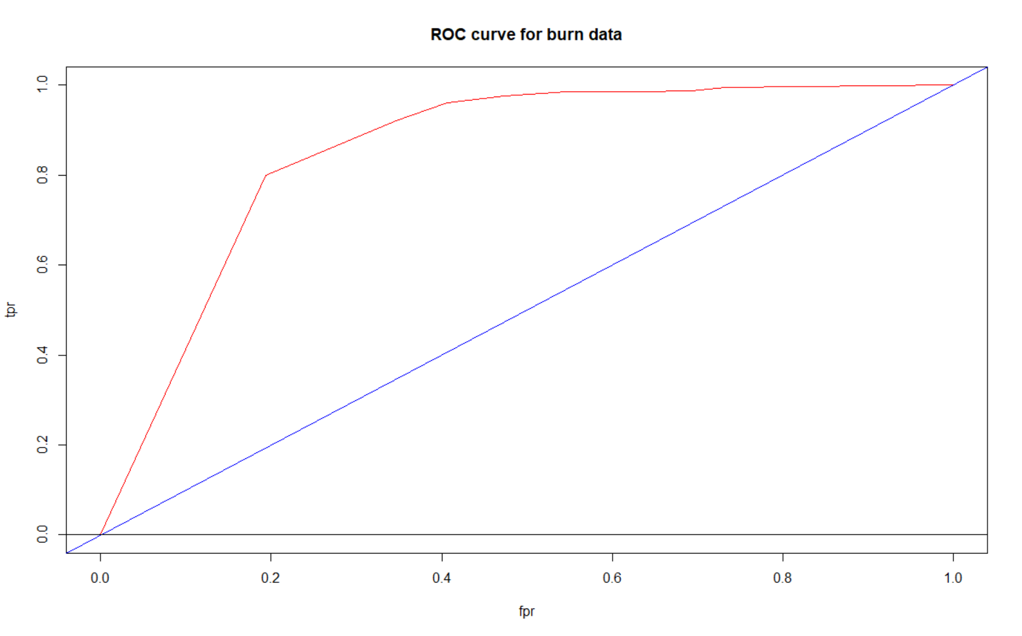
```
burnss
                    tpr             fpr
   threshhold sensitivity  specificity
 1      0           0            1.00
 2      0.100       0.799        0.807
 3      0.200       0.921        0.653
 4      0.300       0.960        0.593
 5      0.400       0.975        0.527
 6      0.500       0.985        0.453
 7      0.600       0.985        0.423
 8      0.700       0.985        0.367
 9      0.800       0.987        0.307
10      0.900       0.995        0.267
11      1.00        1.00         0
```

It is common to construct what we call an ROC curve with this type of data. ROC stands for Receiver Operator Characteristic. The curve is simply a plot of the sensitivity values against 1 – specificity. Sensitivity is the true positive rate (tpr) and 1 – specificity is the false positive rate (fpr).

```
tpr <- burnss$sensitivity
fpr <- 1 - burnss$specificity

plot(tpr ~ fpr, type = "l", col = "red",
 main = "ROC curve for burn data")

abline(0, 1, col = "blue")
abline(h = 0, lty = 1)
```



The closer the plot is to the upper top left-hand corner the more accurate the procedure. The point that lies closest to the upper left-hand corner is usually chosen as the cutoff point that maximizes both sensitivity and specificity simultaneously. The blue line corresponds to a procedure that gives negative and positive results by chance alone; such a test has no inherent value.

The area under the ROC curve (c = 0.852) has a nice interpretation. Suppose we randomly select one patient known to have died and randomly select one patient known to have survived. The area under the ROC curve (c = 0.852) is the probability that the model correctly identifies the two patients.

The area under the blue line is 0.5.

There are several methods for computing the area under the curve (c = 0.852) . The code below will do the job.

```
t <- tpr; f <- fpr

k <- nrow(s) -1

x <- numeric(k)
for (i in 1:k)
{
   x[i] <- .5*(t[i] + t[i+1])*(f[i + 1] - f[i])
}
Area <- sum(x)
Area
[1] 0.8520811
```

2. Inferential Aspects of Logistic Regression

	Model	Odds ratio
Population	$P(Y = 1) = \dfrac{e^{\beta_0 + \beta_1 X}}{1+e^{\beta_0 + \beta_1 X}}$	$OR = e^{\beta_1}$
Sample	$P(\widehat{Y = 1}) = \dfrac{e^{b_0 + b_1 X}}{1+e^{b_0 + b_1 X}}$	$\widehat{OR} = e^{b_1}$

For our example X is Flame or TBSA

- b_0 is an estimate for β_0
- b_1 is an estimate for β_1
- $\widehat{OR} = e^{b_1}$ is an estimate for $OR = e^{\beta_1}$

Inferential Tasks in Logistic Regression

1. Confidence interval for OR = $e^{\beta 1}$ in the case of a single predictor and for $adjOR_1$, $adjOR_2$, … in the case of multiple predictors.

2. Test H_0:OR = $e^{\beta 1}$ = 1 against H_A: OR = $e^{\beta 1}$ ≠ 1

3. With multiple predictors, we need methods that allow us to test for the benefit of adding a variable or a block of variables to an existing model.

In logistic regression inferences can be based on either of two processes:

1. For large n, in repeated samples, the distribution of b_1 is approximately Normal with a mean of β_1.

2. Inferences can more reliably be based on the likelihood function—the probability of getting our sample.

Using the Approximate Normality of b1, b2, …

```
model <- glm(Death ~ TBSA, data = burn,
 family = binomial)
model

 (Intercept)         TBSA
   -3.34511      0.08537
```

```
summary(model)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.345107   0.175648  -19.04   <2e-16
TBSA         0.085367   0.006956   12.27   <2e-16
```

A 95% confidence interval for β_1 is

$0.08537 \pm 1.96*0.006956 \rightarrow 0.0717$ to 0.0990

A 95% confidence interval for OR = $e^{\beta 1}$ is

$e^{0.07174}$ to $e^{0.0990} \rightarrow 1.0743$ to 1.104

```
confint.default(model)
                 2.5 %     97.5 %
(Intercept) -3.68937118 -3.0008438
TBSA         0.07173324  0.0990003
```

The 95% confidence interval,1.0743 to 1.104 is entirely above 1 and so we can reject the null hypothesis (H_0: OR = 1) at the 5% level of significance. The data suggest that the OR > 1.

If you prefer to get a p-value, you can use the summary output again.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.345107   0.175648  -19.04   <2e-16
TBSA         0.085367   0.006956   12.27   <2e-16
```

$$Z = \frac{b_1 - 0}{SE(b_1)} = \frac{0.085367}{0.006956} = 12.27$$

p-value = 2*P(Z > 12.27) = 0

```
model <- glm(Death ~ TBSA + Flame, data = burn,
  family = binomial)
summary(model)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.105814   0.280726 -14.626  < 2e-16
TBSA         0.078119   0.006928  11.276  < 2e-16
Flame        1.267158   0.289756   4.373 1.22e-05

confint.default(model)
                  2.5 %      97.5 %
(Intercept) -4.65602741 -3.55559976
TBSA         0.06454081  0.09169658
Flame        0.69924753  1.83506824
```

A 95% confidence interval for adj_OR_{TBSA} is:

$e^{0.06454081}$ to $e^{0.09169658}$ $\rightarrow$ 1.067 to 1.096

A 95% confidence interval for adj_OR_{Flame} is:

$e^{0.69924753}$ to $e^{1.83506824}$ $\rightarrow$ 2.01 to 6.27

In the case of multiple predictors, the Z-test can be used to test for the benefit of adding a new variable to an existing model.

Is it worth adding the variable Flame to a model predicting the probability of death from only TBSA?

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.105814   0.280726 -14.626  < 2e-16
TBSA         0.078119   0.006928  11.276  < 2e-16
Flame        1.267158   0.289756   4.373 1.22e-05
```

The p-value is the probability of getting a sample slope for Flame at least as large as 1.267 (in either direction) if $\beta_{Flame} = 0$ in a model with TBSA.

p-value = $2 * P(b_{Flame} > 1.267)$

= $2 * P(Z > 4.373)$ = 0.0000122.

Inferences using the Likelihood Function

In logistic regression we estimate the coefficients β_0 and β_1 using a method called Maximum Likelihood Estimation (MLE). A likelihood function expresses the probability of obtaining the observed sample as a function of β_0 and β_1. The method of MLE asks: what values for β_0 and β_1 make our sample most likely?

The simplest situation to illustrate MLE is for the null case where p = P(Y = 1) is independent of X. That is

$$p = P(Y = 1) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$

$$1 - p = P(Y = 0) = \frac{1}{1 + e^{\beta_0}}$$

Then, assuming independent observations

$$L(\beta_0) = \left(\frac{e^{\beta_0}}{1 + e^{\beta_0}}\right)^{150} \left(\frac{1}{1 + e^{\beta_0}}\right)^{850} \Leftarrow \text{Likelihood}$$

$$L_0(\beta_0) = \log_e(L(\beta_0)) = 150*\beta_0 \; -1000*\log_e(1 + e^{\beta_0})$$

⇧

Log Likelihood

We seek the value for β_0 that maximizes $L_0(\beta_0)$:

$$\frac{dL_0}{d\beta_0} = 150 - 1000\frac{e^{\beta_0}}{1 + e^{\beta_0}} = 0 \quad \text{(Calculus)}$$

$$\hat{\beta}_0 = b_0 = -1.7346$$

$$e^{b_0} = e^{-1.7346} = 0.17647 = \frac{150}{850} = \hat{O}$$

$$P(\widehat{Y = 1}) = \frac{e^{-1.7346}}{1 + e^{-1.7346}} = 0.15 = \frac{150}{1000} = \hat{p}$$

For confidence intervals for the population odds ratio(s) we can use the confint command. This yields the *profile-likelihood* intervals.

```
confint(model)
Waiting for profiling to be done...
                  2.5 %     97.5 %
(Intercept) -4.69616358 -3.5907614
TBSA         0.06518979  0.0923746
Flame        0.71873320  1.8604831
```

A 95% confidence interval for adj_OR_{TBSA} is:

$e^{0.06518979}$ to $e^{0.0923746}$ $\rightarrow$ 1.067 to 1.097

(normal case, 1.067 to 1.096)

A 95% confidence interval for adj_OR_{Flame} is:

$e^{0.71873320}$ to $e^{1.8604831}$ $\rightarrow$ 2.05 to 6.42

(normal case, 2.01 to 6.27)

The Deviance and the Drop-in-Deviance Test

In logistic regression the **deviance** plays roughly the same role as the residual sum of squares in linear regression.

The **deviance** associated with a logistic regression model is

$$D = -2 * \log_e(\text{likelihood of the fitted model})$$

For our null model

$$\text{Likelihood} = L(b_0) = \left(\frac{e^{b_0}}{1 + e^{b_0}}\right)^{150} \left(\frac{1}{1 + e^{b_0}}\right)^{850}$$

$$= (0.15^{150})(0.85^{850})$$

$$D = -2*\log_e [(0.15^{150})(0.85^{850})]$$

$$= -2*[\, 150 \log_e(0.15) + 850 \log_e(0.85) \,]$$

$$= 845.42$$

Null deviance

```
model <- glm(Death ~ TBSA, binomial, burn)
summary(model)

Null deviance: 845.42  on 999 degrees of freedom
Residual deviance: 538.65 on 998  degrees of freedom

AIC: 542.65

Number of Fisher Scoring iterations: 5


anova(model, test = "Chisq")

     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                   999     845.42
TBSA  1   306.76       998     538.65 < 2.2e-16
```

Linear Regression		Logistic Regression		
SS	df	Deviance	df	p-value
SSReg	1	306.76	1	0.0000
SSRes	n - 2	538.65	998	
SSTot	n - 1	845.42	999	

$\sum (Y - \bar{Y})^2$ null deviance

$H_0: \beta_{TBSA} = 0 \quad H_A: \beta_{TBSA} \neq 0$

p-value = $P(\chi_1^2 > 306.76) = 0$

The Drop-in-deviance Chi-Square test can be used to compare two models so long as one is *nested* within the other. Model 1 is nested within model 2 if the predictor variables in model 1 are a subset of those in Model 2.

Here are several examples.

Example 1: Is it worth adding the variable Flame to a model predicting P(Y = 1) from TBSA?

1. Z test

```
model2 <- glm(Death ~ TBSA + Flame, binomial,
          burn)
summary(model2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.105814   0.280726 -14.626  < 2e-16
TBSA         0.078119   0.006928  11.276  < 2e-16
Flame        1.267158   0.289756   4.373 1.22e-05
```

2. Drop-in-deviance Chi-Square test

```
model1 <- glm(Death ~ TBSA, binomial, burn)
anova(model1, model2, test = "Chisq")
Analysis of Deviance Table

Model 1: Death ~ TBSA
Model 2: Death ~ TBSA + Flame
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       998     538.65
2       997     516.68  1   21.978 2.758e-06
```

Example 2: Our current model (2) predicts P(Y = 1) from TBSA and Flame. Is it worth adding the remaining four potential predictors Age, Gender, Race, and INH_INJ?

```
model3 <- glm(Death ~ TBSA + Flame + Age +
      Gender + Race + INH_INJ, binomial, burn)
```

```
anova(model2, model3, test = "Chisq")

Analysis of Deviance Table

Model 1: Death ~ TBSA + Flame
Model 2: Death ~ TBSA + Flame + Age + Gender + Race
+ INH_INJ
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       997     516.68
2       993     336.46  4   180.21 < 2.2e-16 ***
```

Building a Logistic Regression Model

```
modelAge <- glm(Death ~Age, binomial, burn)
AIC(modelAge)
[1] 674.2585

modelGender <- glm(Death ~Gender, binomial,
burn)
AIC(modelGender)
[1] 848.5809

:    :    :    :    :    :    :    :    :    :

modelflame <- glm(Death ~ Flame, binomial, burn)
AIC(modelflame)
[1] 759.4591
```

	Variable	AIC
	Age	674.3
	Gender	848.6
	Race	848.0
√	TBSA	542.7
	INH_INJ	695.5
	Flame	759.5

Now consider the performance (using AIC) of all pairs of variables including TBSA. …..

The Complete Model

TBSA_Group = 1 if TBSA ≥ 50

= 0 otherwise

Age_Group = 1 if Age ≥ 32 [= median Age]

= 0 otherwise

```
m <- glm(Death ~ Gender + Race + INH_INJ +
 Flame + TBSA_Group + Age_Group, binomial, burn)

options(digits = 2)
```

Here are the sample slopes:

```
b <- coef(m)

b
(Intercept) Gender  Race  INH_INJ   Flame  TBSA_Group   Age_Group
      -4.51  -0.47 -0.18     1.76    1.04        3.13        2.44
```

Here are the sample adjusted odds ratios:

```
OR <- exp(b)
OR
(Intercept) Gender  Race  INH_INJ   Flame  TBSA_Group   Age_Group
      0.011  0.626 0.836    5.805   2.838      22.872      11.444
```

Here are the 95% CI's for the β's

```
c <- confint(m)
Waiting for profiling to be done...
c
               2.5 % 97.5 %
(Intercept) -5.40 -3.726
Gender      -0.95  0.019
Race        -0.65  0.301
INH_INJ      1.21  2.315
Flame        0.48  1.645
TBSA_Group   2.37  3.978
Age_Group    1.79  3.172
```

Here are the 95% CI's for the adjusted OR's

```
CI <- exp(c)
CI
               2.5 % 97.5 %
(Intercept)  0.0045  0.024
Gender       0.3869  1.019
Race         0.5196  1.351
INH_INJ      3.3562 10.128
Flame        1.6172  5.181
TBSA_Group  10.6780 53.436
Age_Group    6.0115 23.846
```

```
Null deviance: 845.42  on 999  degrees of freedom
Residual deviance: 504.49 on 993 degrees of freedom
AIC: 518.5
```

$D_0 - D_6 = 845.42 - 504.49 = 340.93$

This value can be compared to the Chi-Square distribution with 6 degrees of freedom.

Variable	Slope	Adj_OR	95% CI
Gender	- 0.468	0.626	0.387 - 1.019
Race	- 0.179	0.836	0.520 - 1.351
INH_INJ	1.759	5.807	3.356 -10.128
Flame	1.043	2.838	1.617 - 5.181
TBSA_Group	3.130	22.874	10.678 - 53.436
Age_Group	2.438	11.450	6.012 - 23.846

Conditions for Inference in Logistic Regression

(a) Conditions we don't need

- No more condition that the Y values are approximately normal. Why not?

- No more condition that the standard deviation of the Ys not vary with the Xs.

(b) Conditions we do need

- We assume a linear relationship between the X variables and **logit** of Y

  $$L = \log_e\left(\frac{\hat{p}}{1-\hat{p}}\right) = b_0 + b_1X_1 + b_2X_2 + \ldots$$

  It is hard to check unless n is very large.

- We assume that the observations represent a random sample from some well-defined population.

Sample Size and Model Complexity in Logistic Regression

Here is a popular guideline for sample size in logistic regression

Suppose p_0 is the proportion of 0's in our sample and p_1 is the proportion of 1's.

Call p the smaller of p_0 and p_1.

Call K the number of predictors (explanatory variables) in our model

Then the minimum sample size needed is

n = 10*K/p

For the burn data, $p_0 = 0.85$ and $p_1 = 0.15$, so p = 0.15.

With K = 6, n = 10*6/0.15 = 400