teleuniv
innovative interactive immersive

Logistic Regression

Ajeet K. Jain

An Introduction to Logistic Regression

Logistic regression is one of the most popular machine learning algorithms for binary classification. This is because it is a simple algorithm that performs very well on a wide range of problems.

Logistic regression extends the ideas of linear regression to the situation where the dependent variable, **Y , is categorical**.

Frog Jumps and not generally Walk !!!

Logistic regression is a predictive modelling algorithm that is used when the Y variable is binary categorical. That is, it can take only two values like 1 or 0. The goal is to determine a mathematical equation that can be used to predict the probability of event 1. Once the equation is established, it

- A categorical variable as divides the observations into classes.
  - If Y denotes a recommendation on holding /selling / buying a stock, then we have a categorical variable with 3 categories.
  - Each of the stocks in the dataset (the observations) as belonging to one of three classes: the “hold" class, the “sell" class, and the “buy” class.

- Logistic regression can be used for classifying a new observation into one of the classes, based on the values of its predictor variables (called “classification").

- It can also be used in data (where the class is known) to find similarities between observations within each class in terms of the predictor variables (called

Some real world examples of binary classification problems

You might wonder what kind of problems you can use logistic regression for.

Here are some examples of binary classification problems:

Spam Detection : Predicting if an email is Spam or not

Credit Card Fraud : Predicting if a given credit card transaction is fraud or not

Health : Predicting if a given mass of tissue is benign or malignant

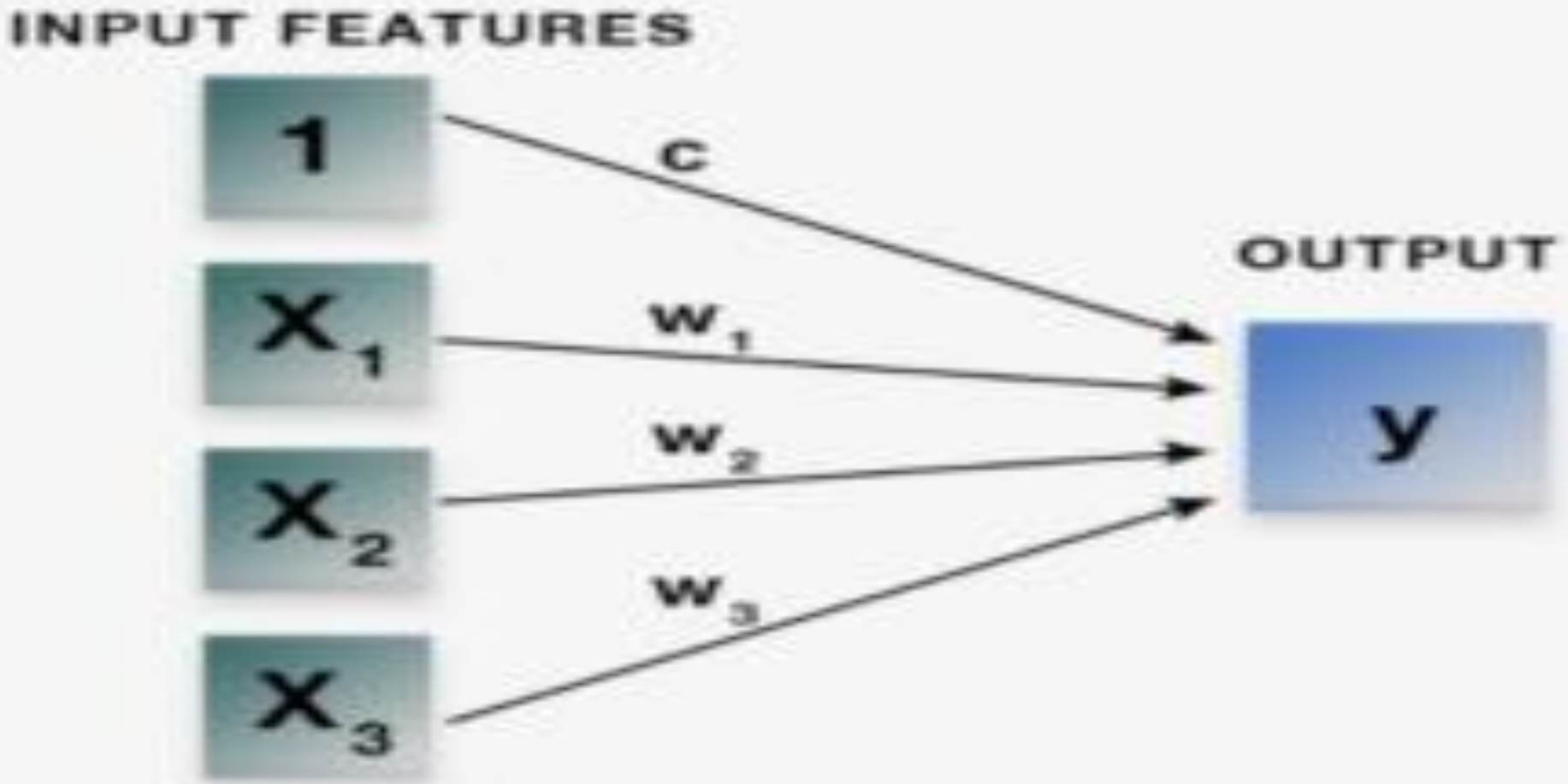
Marketing : Predicting if a given user will buy an insurance product or not

Banking : Predicting if a customer will default on a loan.

Recap : Regression

- A form of statistical modeling that attempts to evaluate the relationship between one variable (termed the dependent variable) and one or more other variables (termed the independent variables). It is a form of global analysis as it only produces a single equation for the relationship.
- A model for predicting one variable from another.

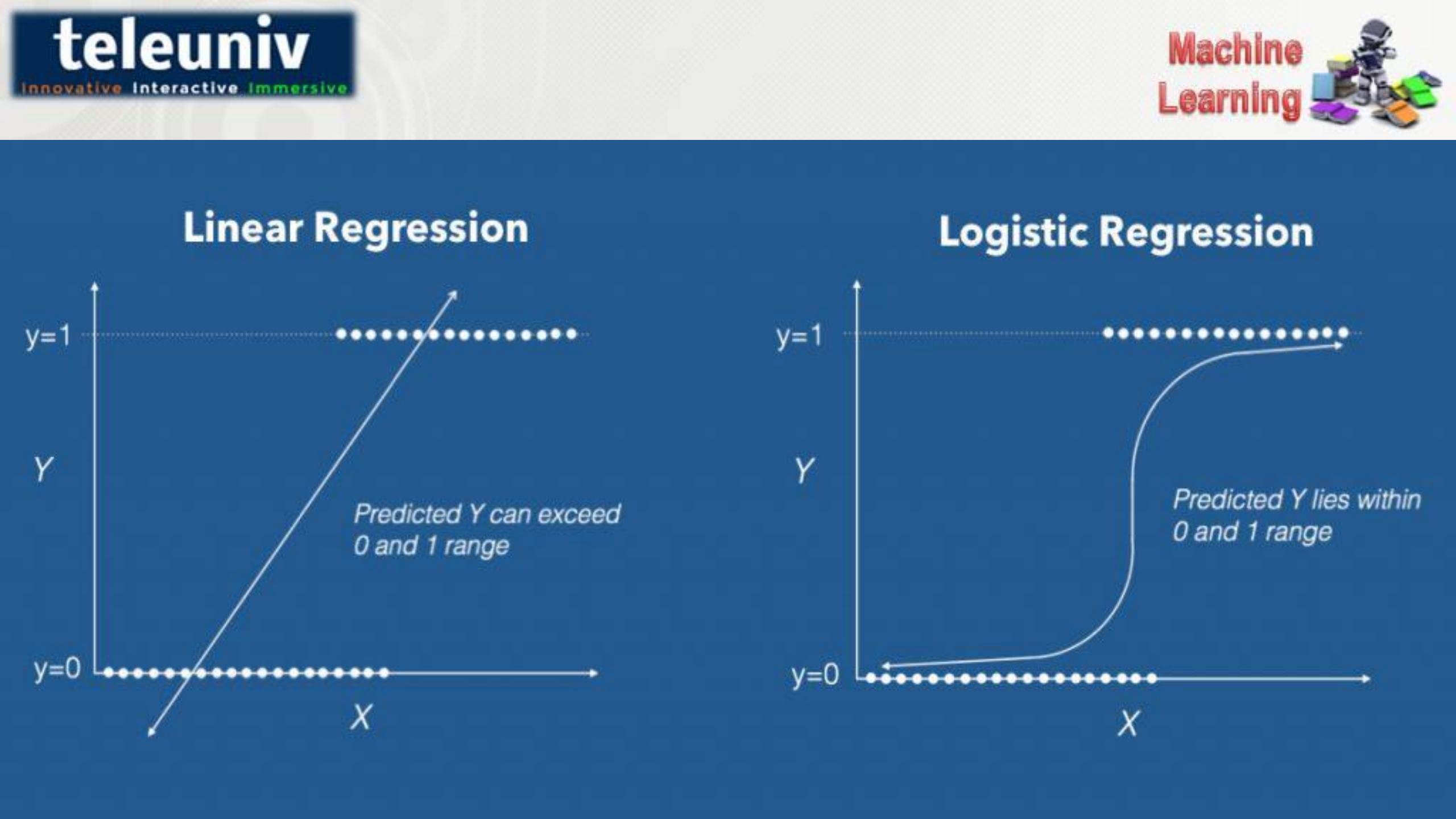
LINEAR REGRESSION



$$y = c + x_1*w_1 + x_2*w_2 + x_3*w_3 + \ldots\ldots\ldots + x_n*w_n$$

Why not linear regression?

When the response variable has only 2 possible values, it is desirable to have a model that predicts the value either as 0 or 1 or as a probability score that ranges between 0 and 1. Linear regression does *not* have this capability. Because, If you use linear regression to model a binary response variable, the resulting model may not restrict the predicted Y values within 0 and 1.


Linear Regression
y=1
Y
y=0
X
Predicted Y can exceed
0 and 1 range
Logistic Regression
y=1
Y
y=0
X
Predicted Y lies within
0 and 1 range

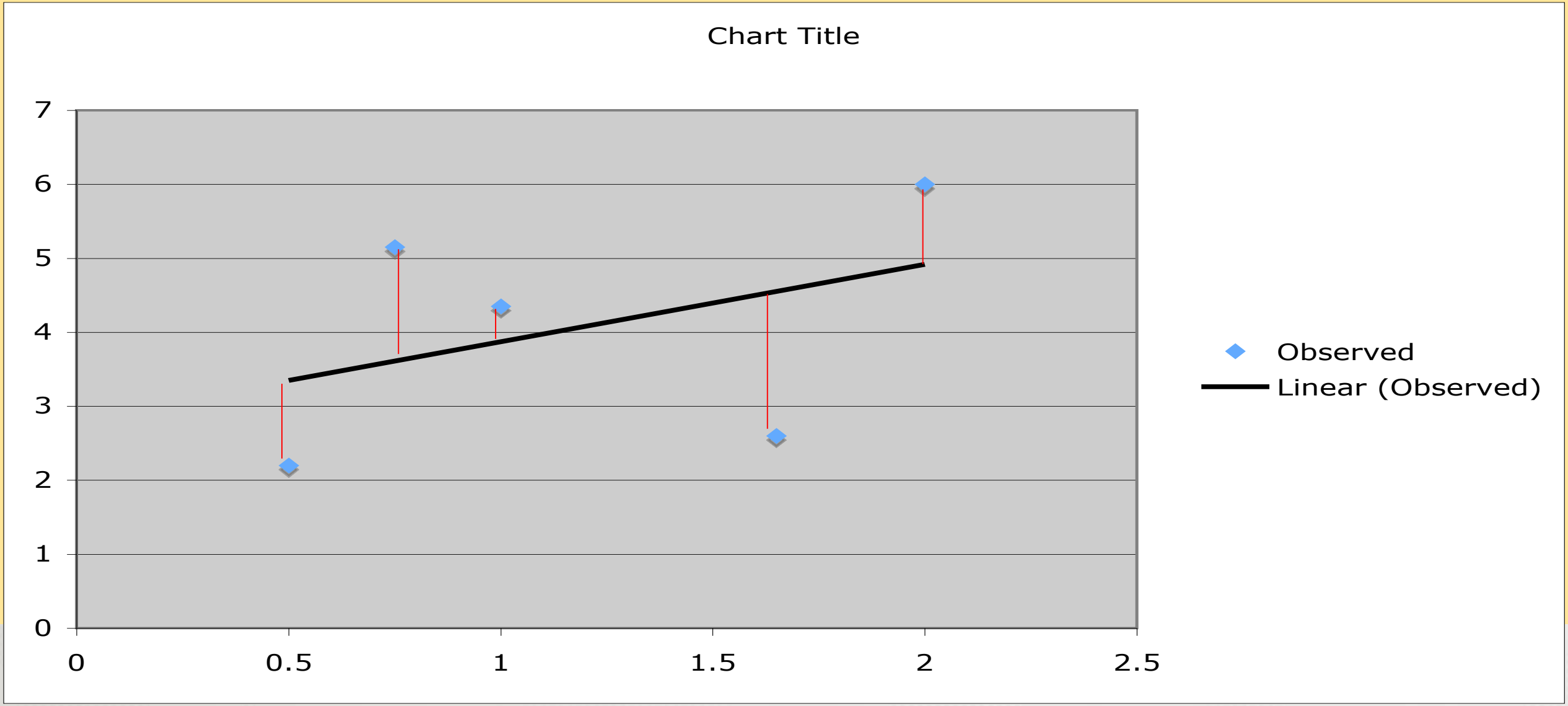
Linear Regression (contd…)

- Regression used to fit a linear model to data where the dependent variable is continuous:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \square + b_n X_n + \epsilon$$

- Given a set of points (Xi,Yi), we wish to find a linear function (or line in 2 dimensions) that "goes through" these points.
- **In general, the points are not exactly aligned**:
  - **Find line that best fits the points**

- **Error or Residue:**
  - **Observed value - Predicted value**

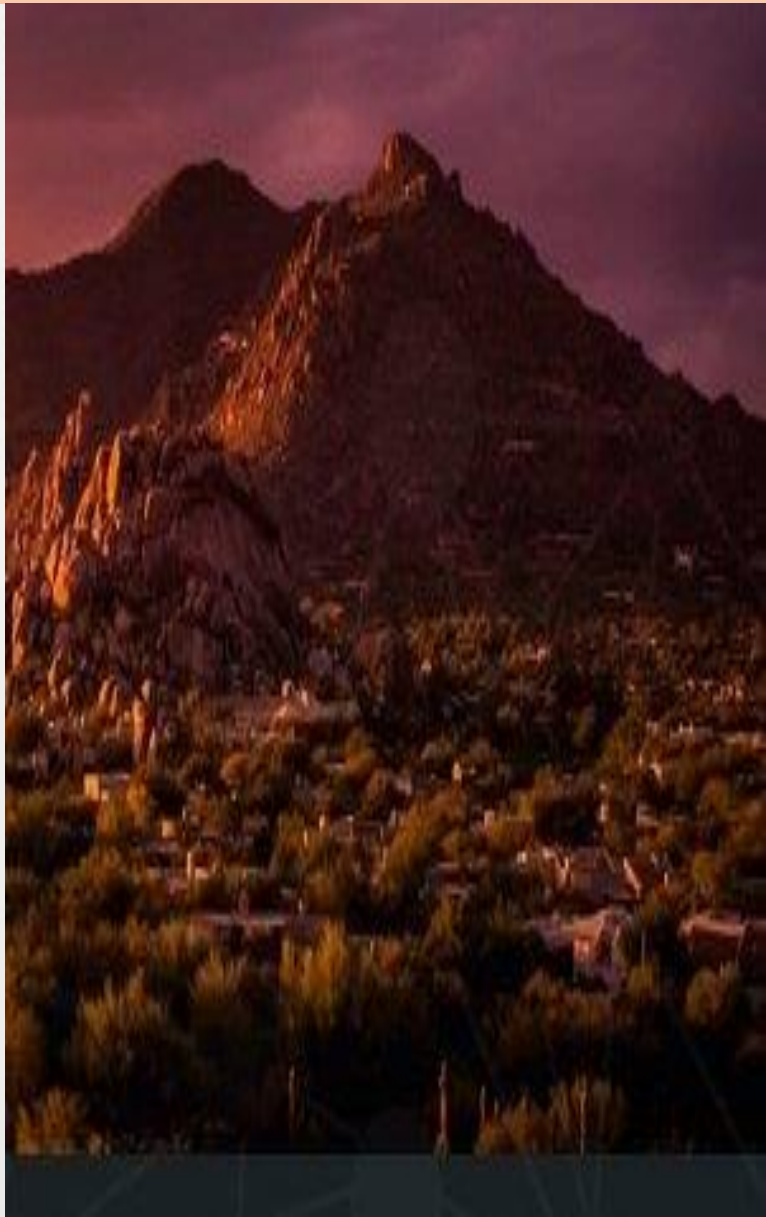


Sum-Squared Error (SSE) / Total Sum of Squares (TSS)

$$SSE = \sum_{y} (y_{observed} - y_{predicted})^2$$

$$TSS = \sum_{y} (y_{observed} - \bar{y}_{observed})^2$$

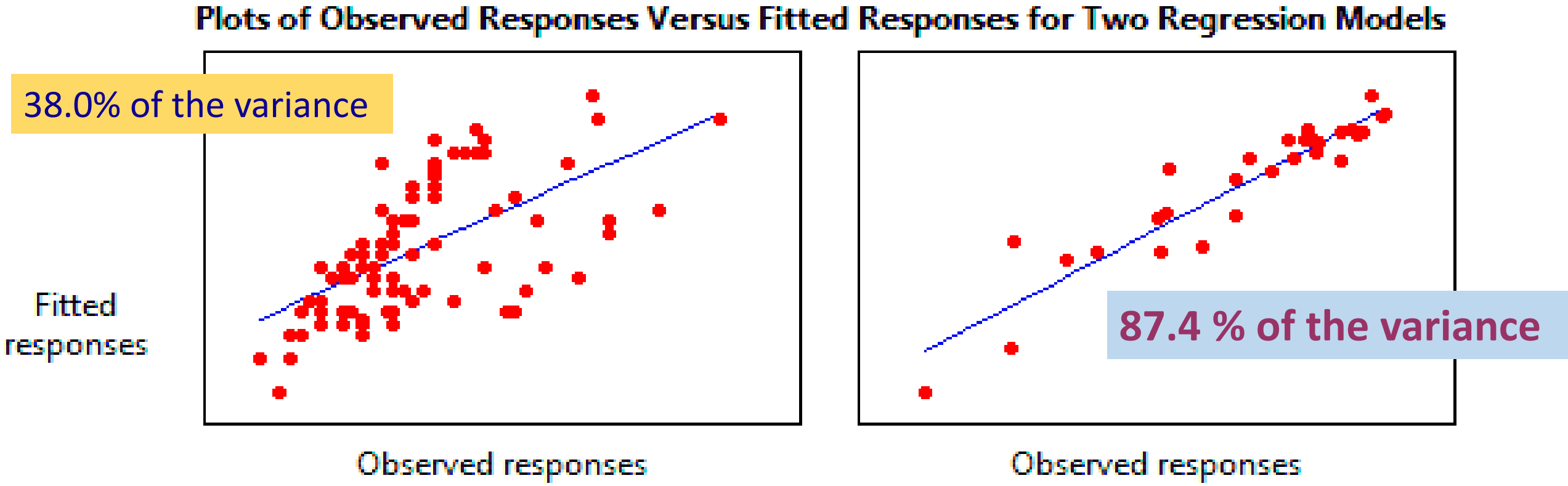
$$R^2 = 1 - \frac{SSE}{TSS}$$



What Is R-squared?

- R-squared is a statistical measure of how close the data are to the fitted regression line.
- It is also known as coefficient of determination, or the coefficient of multiple determination for multiple regression.
- R-squared is the percentage of the response variable variation that is explained by a linear model.

Graphical Representation of R-squared



The more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line. Theoretically, if a model could explain 100% of the variance, the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line.

What Is R-squared?(contd…)

R-squared(R^2) = Explained variation / Total variation

R-squared is always between 0 and 100% :

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

What is Best Fit?

- The smaller the SSE, the better the fit
- Hence,
  - Linear regression attempts to minimize SSE (or similarly to maximize R2)
- Assume 2 dimensions

$$Y = b_0 + b_1 X$$

Analytical Solution

$$b_0 = \frac{\sum y - b_1 \sum x}{n}$$

$$b_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - \left(\sum x\right)^2}$$

Example (I)

x	y	x^2	xy
1.20	4.00	1.44	4.80
2.30	5.60	5.29	12.88
3.10	7.90	9.61	24.49
3.40	8.00	11.56	27.20
4.00	10.10	16.00	40.40
4.60	10.40	21.16	47.84
5.50	12.00	30.25	66.00
24.10	**58.00**	**95.31**	**223.61**

Target: $y=2x+1.5$

$$b_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - \left(\sum x\right)^2}$$

$$= \frac{7 \times 223.61 - 24.10 \times 58.00}{7 \times 95.31 - 24.10^2}$$

$$= \frac{1565.27 - 1397.80}{667.17 - 580.81}$$

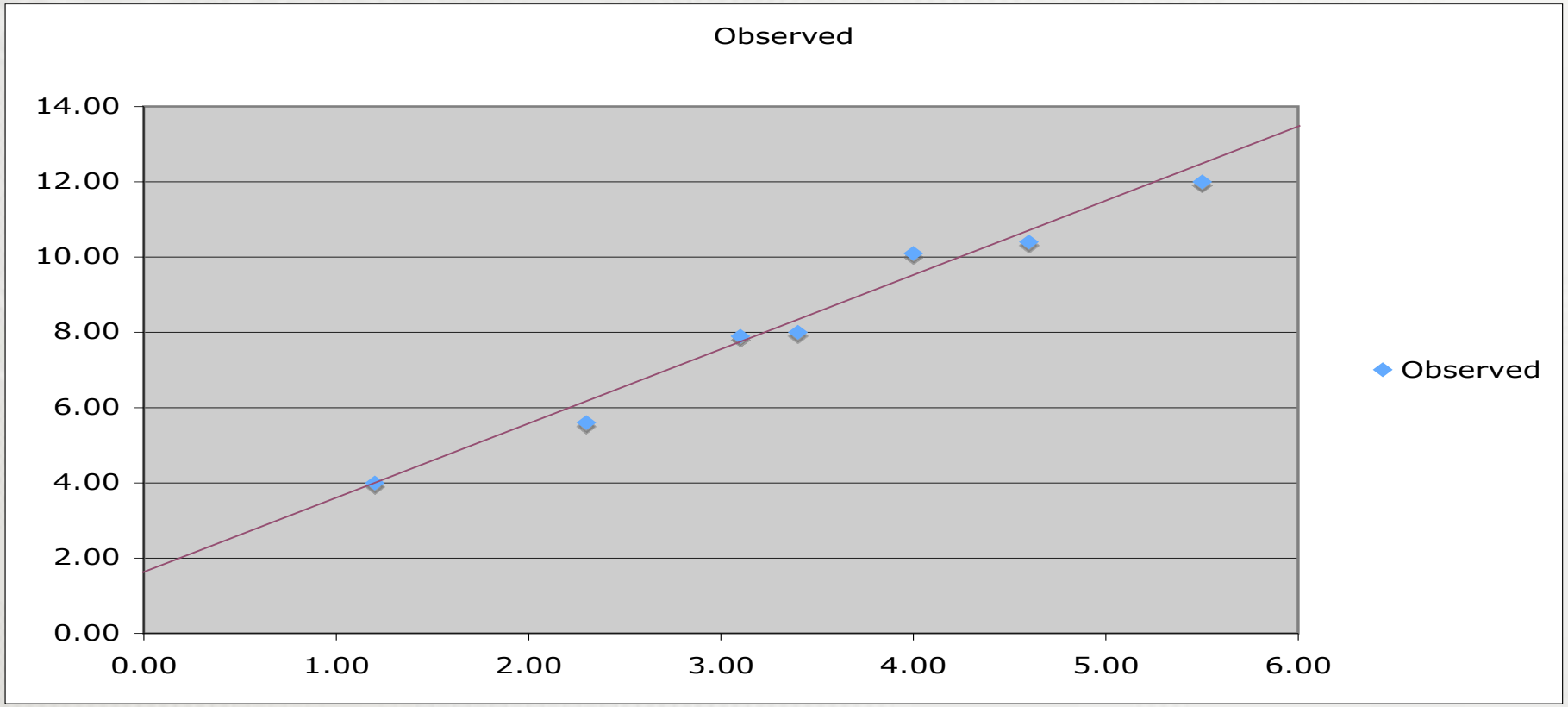
$$= \frac{167.47}{86.36} = \underline{\underline{1.94}}$$

$$b_0 = \frac{\sum y - b_1 \sum x}{n}$$

$$= \frac{58.00 - 1.94 \times 24.10}{7}$$

$$= \frac{11.27}{7} = \underline{\underline{1.61}}$$

Example (II)


Observed
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
Observed

Example (III)

x	y (obs)	y (pred)	SSE	TSS
1.20	4.00	3.94	0.004	18.367
2.30	5.60	6.07	0.221	7.213
3.10	7.90	7.62	0.078	0.149
3.40	8.00	8.21	0.044	0.082
4.00	10.10	9.37	0.533	3.292
4.60	10.40	10.53	0.017	4.470
5.50	12.00	12.28	0.078	13.796
			0.975	**47.369**

$$R^2 = 1 - \frac{SSE}{TSS} = 1 - \frac{0.975}{47.369} = 0.98$$

Logistic Regression

- Regression used to fit a curve to data in which the dependent variable is binary, or dichotomous

- Typical application: Medicine
  - We might want to predict response to treatment, where we might code survivors as 1 and those who don't survive as 0

Example

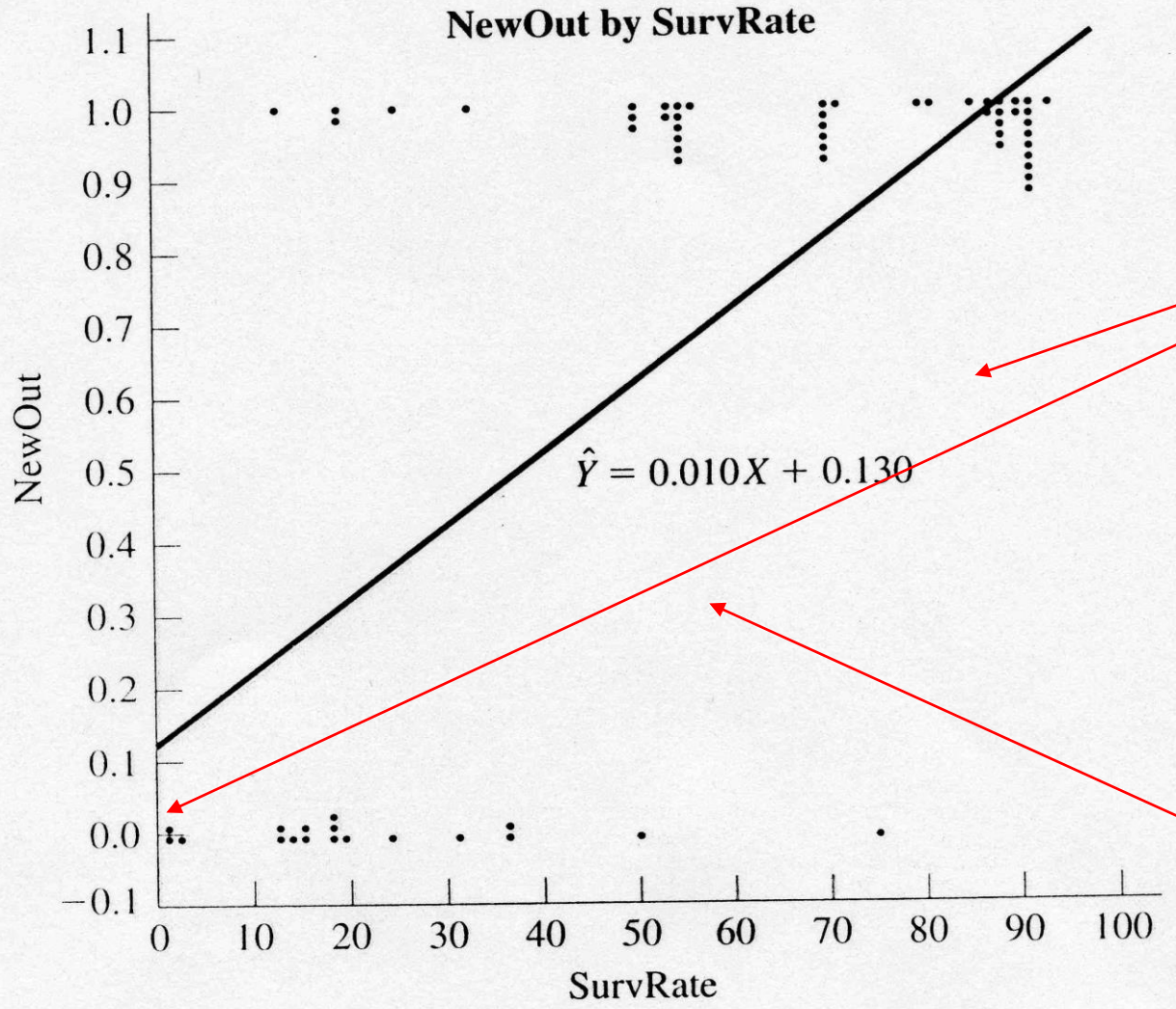


FIGURE 15.7 *Outcome as a function of SurvRate*

Observations:
For each value of SurvRate, the number of dots is the number of patients with that value of NewOut

Regression:
Standard linear regression

Problem: extending the regression line a few units left or right along the X axis produces predicted probabilities that fall outside of [0,1]

A Better Solution

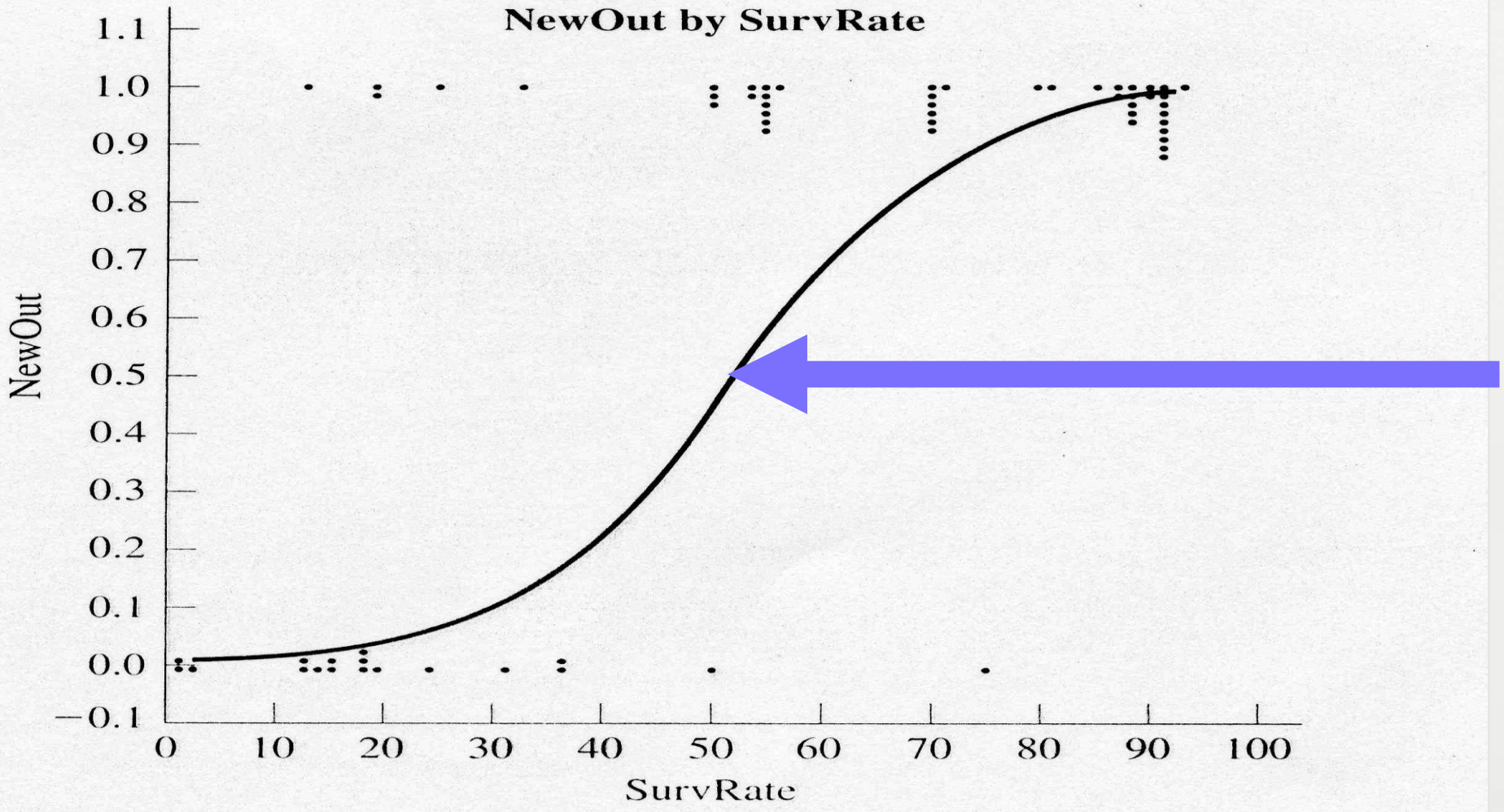


FIGURE 15.8 *More appropriate regression line for predicting outcome*

Regression Curve: Sigmoid function!

(bounded by asymptotes $y=0$ and $y=1$)

Odds

- Given some event with probability *p* of being 1, the odds of that event are given by: **odds = *p* / (1–*p*)**

- Consider the following data

Delinquent

Testosterone	Yes	No	Total
Normal	402	3614	4016
High	101	345	446
	503	3959	4462

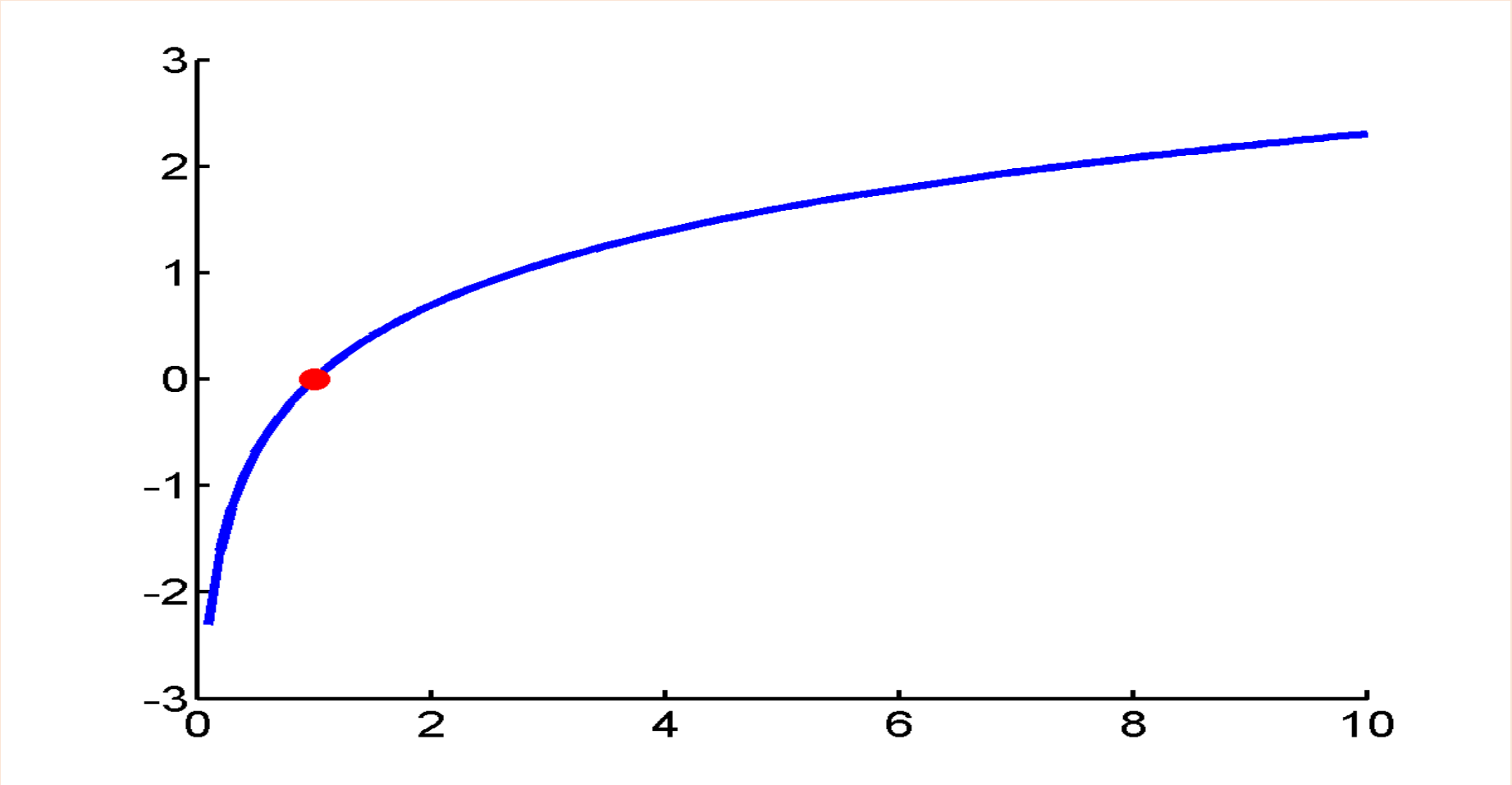
- The odds of being delinquent if you are in the Normal group are:

pdelinquent/(1–pdelinquent) = (402/4016) / (1 - (402/4016)) = 0.1001 / 0.8889 = 0.111

Odds Ratio

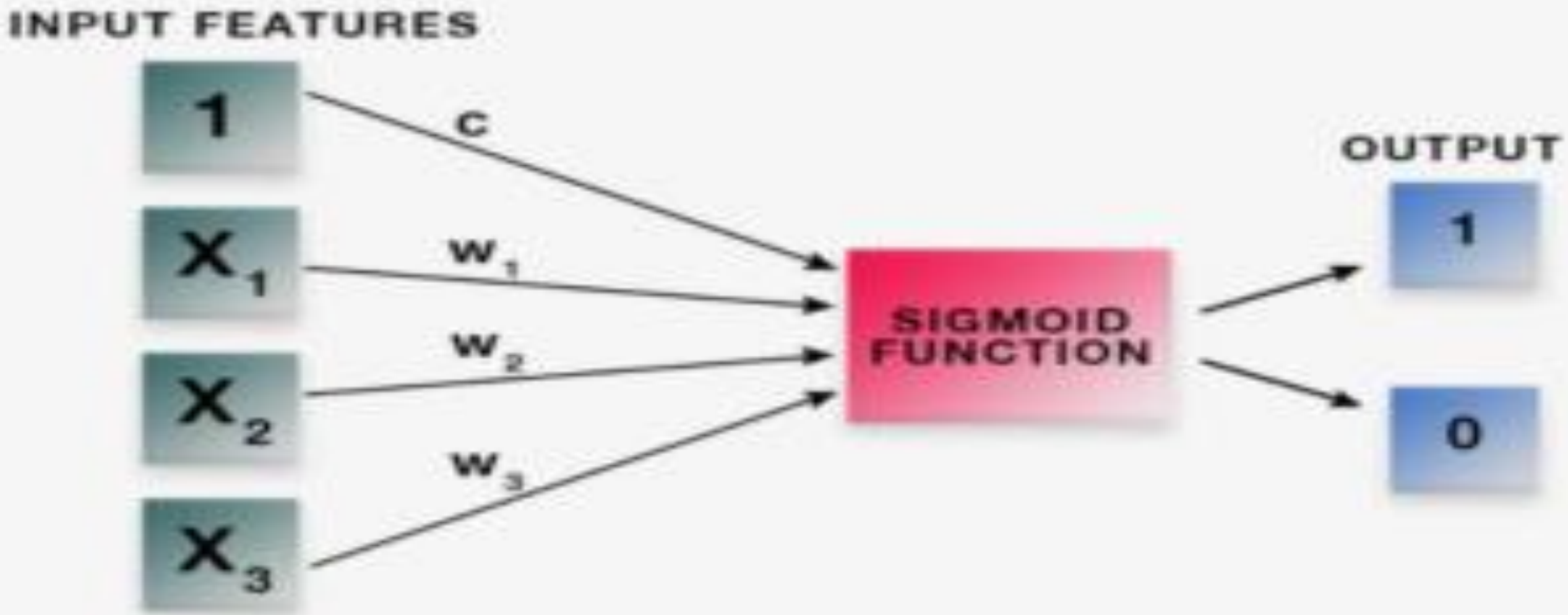
- The odds of being not delinquent in the Normal group is the reciprocal of this:
  - 0.8999/0.1001 = 8.99
- Now, for the High testosterone group
  - odds(delinquent) = 101/345 = 0.293
  - odds(not delinquent) = 345/101 = 3.416
- When we go from Normal to High, the odds of being delinquent nearly triple:
  - Odds ratio: 0.293/0.111 = 2.64
  - 2.64 times more likely to be delinquent with high testosterone levels

- The logit is the natural log of the odds



- logit(p) = ln(odds) = ln (p/(1-p))

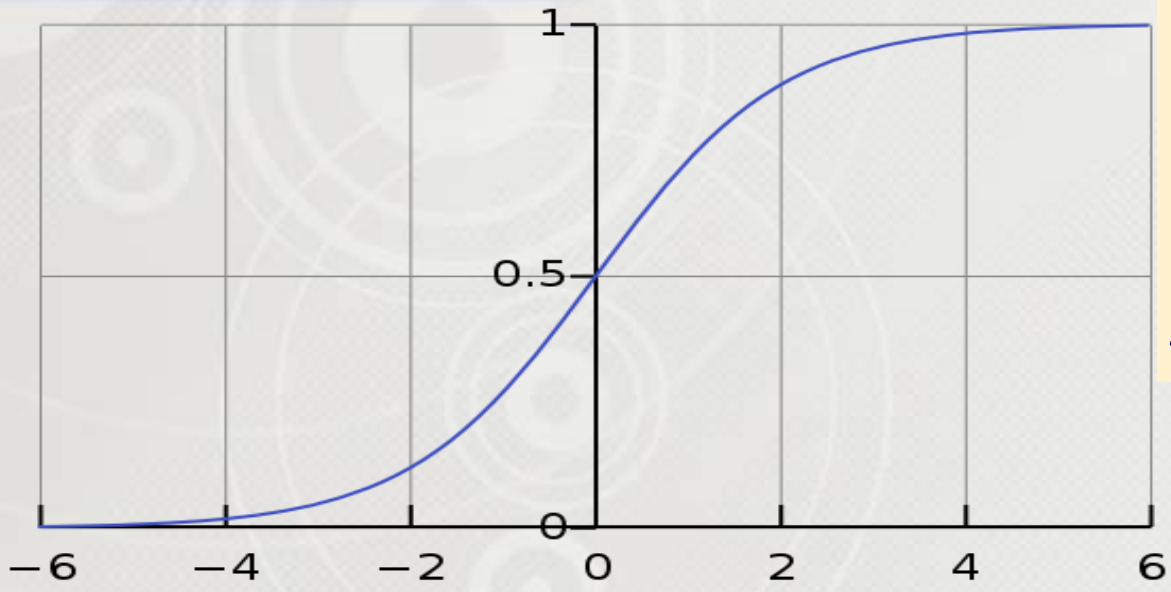
LOGISTIC REGRESSION



$$y = \text{logistic}\ (c + x_1{*}w_1 + x_2{*}w_2 + x_3{*}w_3 + \ldots\ldots\ldots + x_n{*}w_n)$$

$$y = 1\ /\ 1 + e\ [-\ (c + x_1{*}w_1 + x_2{*}w_2 + x_3{*}w_3 + \ldots\ldots\ldots + x_n{*}w_n)]$$

Sigmoid Function



A **sigmoid function** is a mathematical function having a characteristic "S"-shaped curve or **sigmoid curve**. *Sigmoid function* is defined by the formula :

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{1 + e^{x}}$$

Sigmoid functions have domain of all **real numbers**, with return value **monotonically increasing** most often from 0 to 1 or alternatively from −1 to 1, depending on convention.

Are differentiable having a non-negative first derivative which is bell shaped. A sigmoid function is constrained by a pair of horizontal asymptotes as : $x \to \pm\infty$

In [10]:
```
import math
def sigmoid(x):
  return 1 / (1 + math.exp(-x))
print(sigmoid(0.458))   # calculate for positive value of argument
print(sigmoid(-0.458))  # calculate for negaqtive value of argument
%timeit -r 1 sigmoid(0.458)
```

```
0.6125396134409151
0.3874603865590849
258 ns ± 0 ns per loop (mean ± std. dev. of 1 run, 1000000 loops each)
```

In [12]:
```
from scipy.special import expit
x = expit(0.458)
y = expit(-0.458)
print(x)
print(y)
%timeit -r 1 expit(0.458)
```

```
0.6125396134409151
0.3874603865590849
565 ns ± 0 ns per loop (mean ± std. dev. of 1 run, 1000000 loops each)
```

In [13]:
```
import numpy as np
def sigmoid(x):
  return 1 / (1 + np.exp(-x))
print(sigmoid(0.458))
print(sigmoid(-0.458))
%timeit -r 1 np.exp(0.458)
```

```
0.6125396134409151
0.387460386559084 9
627 ns ± 0 ns per loop (mean ± std. dev. of 1 run, 1000000 loops each)
```

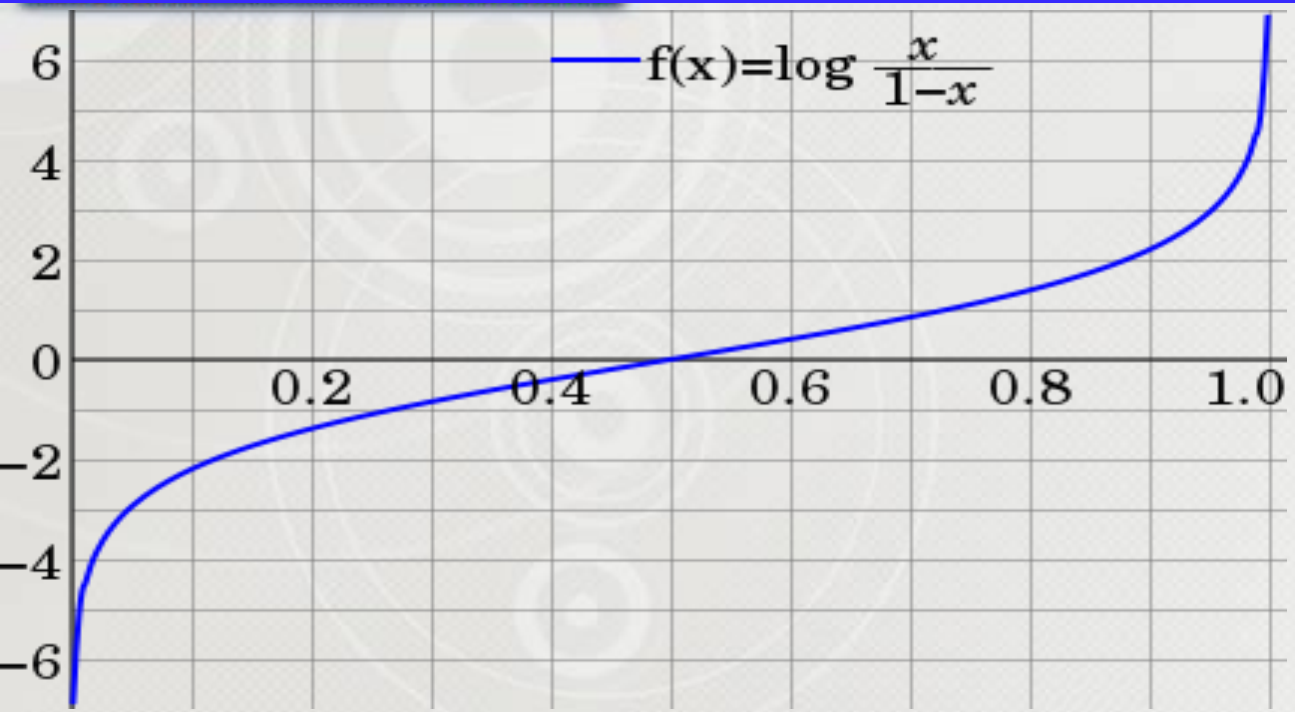
In [14]:
```
import numpy as np

x = np.random.random(1000000)

def sigmoid_array(x):
    return 1 / (1 + np.exp(-x))
%timeit -r 1 -n 100 sigmoid_array(x)
%timeit -r 1 -n 100 expit(x)
```

```
14.8 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 100 loops each)
9.94 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 100 loops each)
```

Logit Function



Plot of logit(*p*) in the domain of 0 to 1,
Where the base of logarithm is *e*

- The **logit** function is the inverse of the sigmoidal "logistic" function or logistic transform used in mathematics, especially in statistics. When the function's variable represents a probability *p*, the logit function gives the **log-odds**, or the logarithm of the odds $p/(1-p)$.

The **logit** of a number *p* between 0 and 1 is given by the formula:

$$logit(x) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p) = -\log\left(\frac{1}{p} - 1\right)$$

$$logit(x) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p) = -\log\left(\frac{1}{p} - 1\right)$$

The base of the logarithm function used is of little importance : as long as it is greater than 1, but the natural logarithm with base **e** is the one most often used. The choice of base corresponds to the choice of logarithmic unit (logit) for the value:

base 2 corresponds to a Shannon, base e to a nat, and base 10 to a Hartley; these units are particularly used in information-theoretic interpretations. For each choice of base, the logit function takes values between negative and positive infinity.

The "logistic" function of any number α is given by the inverse-logit:

$$\operatorname{logit}^{-1}(\alpha) = \operatorname{logistic}(\alpha) = \frac{1}{1+\exp(-\alpha)} = \frac{\exp(\alpha)}{\exp(\alpha)+1}$$

If *p* is a probability, then *p*/(1 − *p*) is the corresponding odds;
the logit of the probability is the logarithm of the odds.

Similarly, the difference between the logits of two probabilities is the logarithm of the **odds ratio (*R*),** thus providing a shorthand for writing the correct combination of odds ratios **only by adding and subtracting:**

$$\log(R) = \log\left(\frac{p_1/(1-p_1)}{p_2/(1-p_2)}\right) = \log\left(\frac{p_1}{1-p_1}\right) - \log\left(\frac{p_2}{1-p_2}\right) = \operatorname{logit}(p_1) - \operatorname{logit}(p_2).$$