Big Data Analytics

Session 4

Logistic Regression

Classification

- Regression vs Classification
  - Response variable: quantitative vs qualitative
- Classification: predicting a qualitative/categorical response
  - Given an observation, classify it (assign it to a category or class)
- Classification: in some cases making decisions based on
  - Predicting the probability of each of the categories
  - In this sense classification behaves like regression methods
- Widely-used classifiers (classification techniques)
  - Logistic regression, linear discriminant analysis, K-nearest neighbors (IRO), Generalised additive models, tree-based methods, support vector machines

Logistic Regression Outline

- Case: Orange Juice Brand Preference
- Why Not Linear Regression?
- Simple Logistic Regression
    - Logistic Function
    - Interpreting the coefficients
    - Making Predictions
- Case: Credit Card Default Data (A whole running example)
- Adding Qualitative Predictors
- Multiple Logistic Regression
- This session is based on Chapter 4.3 in ISLR.

Case 1: Brand Preference for Orange Juice

- We would like to predict what customers prefer to buy: Innocent or Tropicana orange juice?

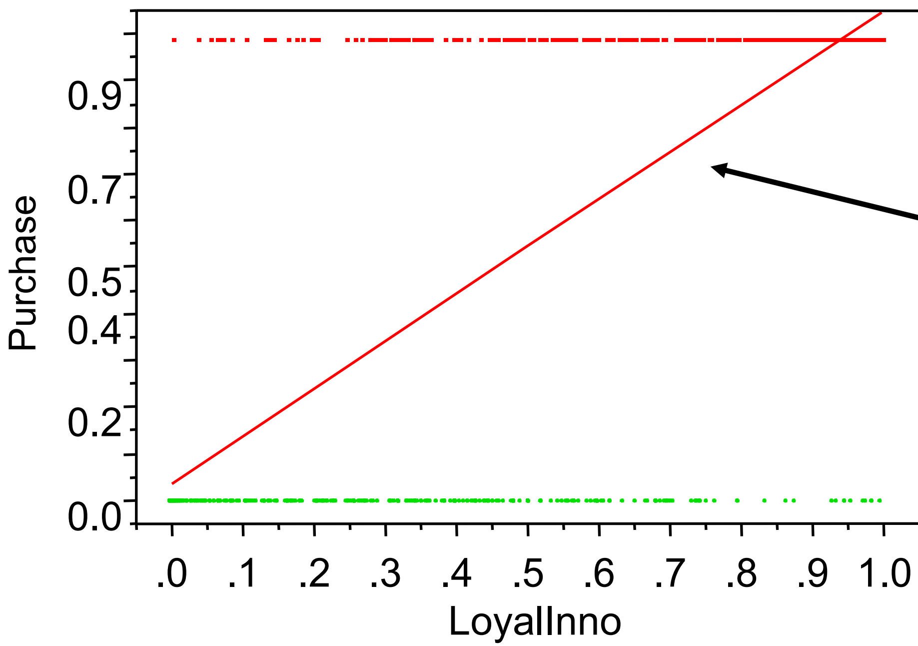
- The Y (Purchase) variable is categorical: 0 or 1

- The X (LoyalInno) variable is a numerical value (between 0 and 1) which specifies the how much the customers are loyal to the Innocent (Inno) orange juice.

- Can we use Linear Regression when Y is categorical?

Why not Linear Regression?

- When Y only takes on values of 0 and 1, why is standard linear regression inappropriate?
  - The red and green ticks indicate the 1/0 values coded for Purchase of Innocent and or not



How do we interpret values of Y between 0 and 1?

Problems

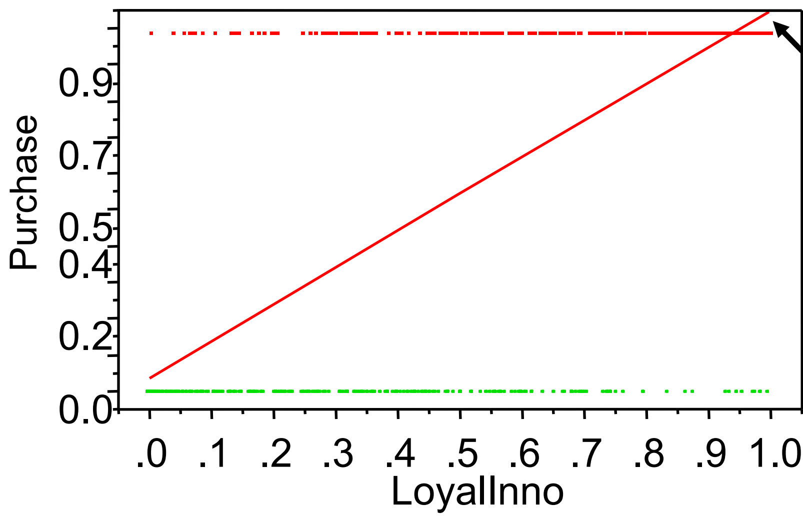
- The regression line $\beta_0+\beta_1 X$ can take on any value between negative and positive infinity

- BUT, in the orange juice classification problem, Y can only take on two possible values: 0 or 1.

- Therefore the regression line almost always predicts the wrong value for Y in classification problems

More Problems

- Solution:
  - Instead of trying to predict Y, let's try to predict P(Y = 1), i.e., the probability a customer buys Innocent juice.
  - Thus, P(Y = 1) gives outputs between 0 and 1.
- Again, can we use linear regression for P(Y=1)?



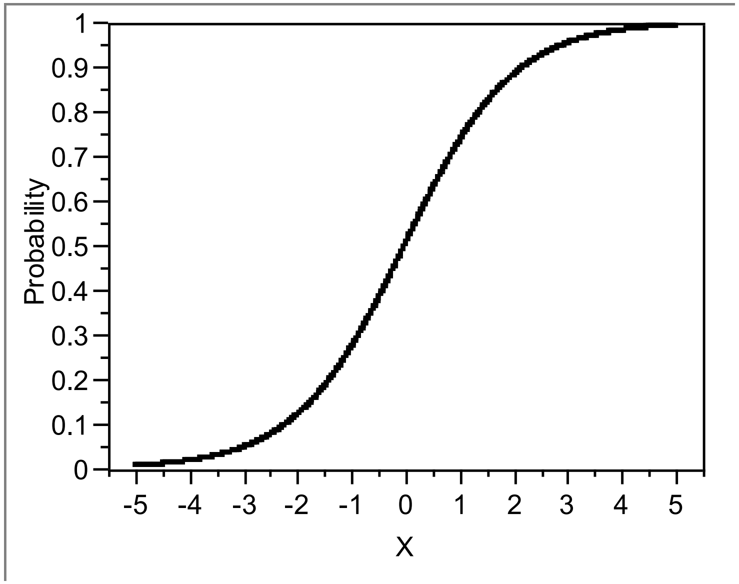
for high loyalty we predict a probability above 1

for very low loyalty we predict a negative probability

Solution: Use Logistic Function

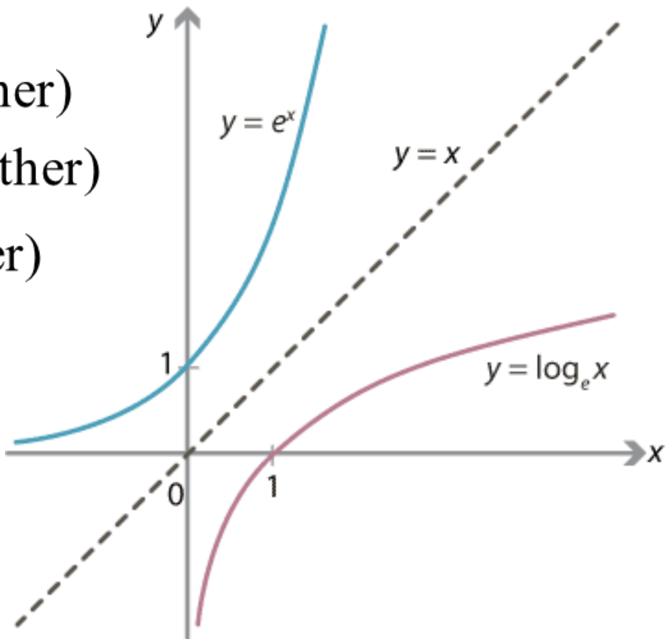
- Goal: we need to model P(Y = 1) using a function that gives outputs between 0 and 1.
- We can use the logistic function ➔ Logistic Regression!
  - Always produce an S-shaped curve
  - Always between 0 and 1
  - Regardless of the value of X, we always obtain a sensible prediction

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



A Primer on *e* and log

- Exponentials
  - $3^2 = 3 * 3$ (Multiply two 3 together)
  - $a^5 = a * a * a * a * a$ (Multiply five as together)
  - $b^n = \underbrace{b * b * \cdots * b}_{n\ b\text{s}}$ (Multiply n bs together)
- Logarithms
  - Inverse of exponentials
  - $\log_{10} 1000 = 3$ because $10^3 = 1000$
  - $\log_2 16 = 4$ because $2^4 = 16$



- e – a constant, approximately equal to 2.71828
  - $y = e^x$ is the natural exponential function
  - $y = \log_e x$ is the natural logarithm function, it is the inverse function of $y = e^x$
  - $\ln x = \log_e x, \qquad \log x = \log_{10} x$

More on Logistic Functions

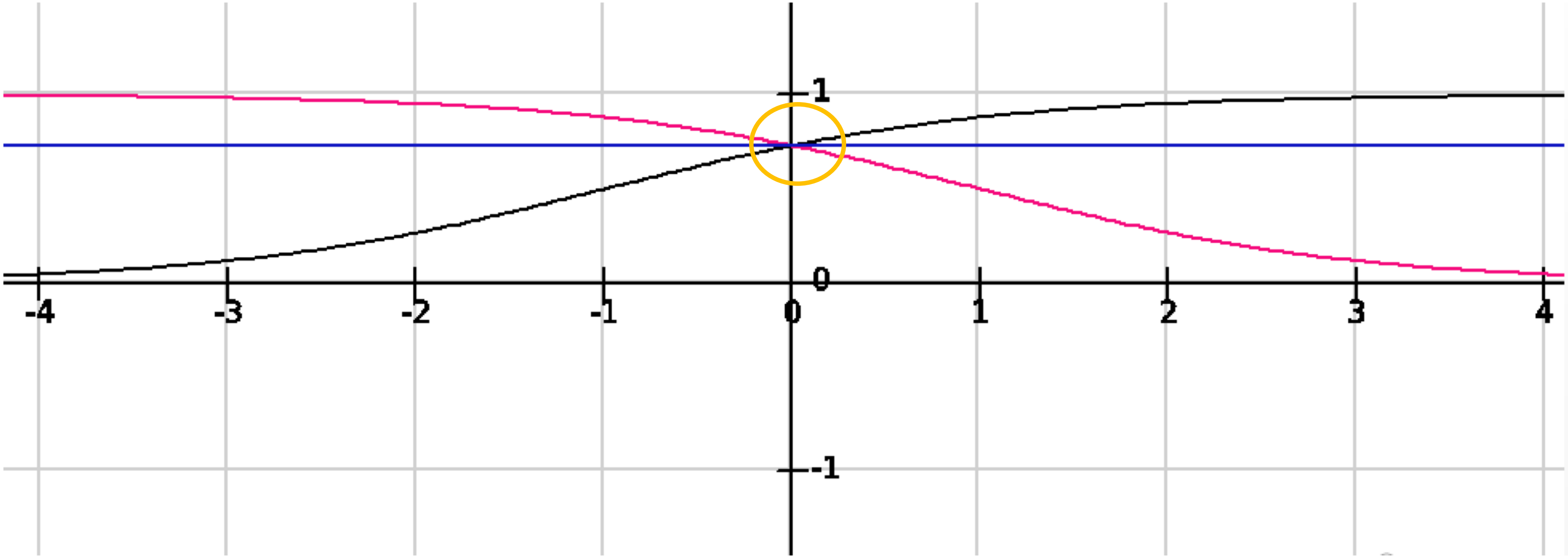
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

Intercept point: $\frac{e^{\beta_0}}{1+e^{\beta_0}}$

blue line: $\beta_1 = 0$

black curve: $\beta_1 > 0$

red curve: $\beta_1 < 0$



More on Logistic Function

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

Conditional probability

Given the condition that one is 0.7 loyal to Innocent ($X = 0.7$), what is the probability that this person buys Innocent juice ($Y = 1$)?

Odds

$$\frac{\text{chance for}}{\text{chance against}}$$

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

Odds ←

$$\boxed{\frac{p(X)}{1 - p(X)}} = e^{\beta_0 + \beta_1 X}$$

The odds of a month being August is ? 1/11

The odds of a coin landing heads up is ? 1/1

- Odds can take on any value between 0 and infinity
  - 1 in 5 people will buy Innocent Juice with an odds of ¼
    p(X) = 0.2, odds = 0.2/(1-0.2) = 1/4
  - 9 out of 10 people will buy Innocent Juice with an odds of 9
    p(X) = 0.9, odds = 0.9/(1-0.9) = 9
- Odds are traditionally used instead of probabilities in horse-racing, since they relate more naturally to the correct betting strategy. See more http://www.racingexplained.co.uk/betting/understanding-odds/

Odds


LIFELONG SMOKERS HAVE A ONE-IN-TWO CHANCE OF DYING FROM SMOKING-RELATED DISEASE.
IT'LL NEVER HAPPEN TO ME.
THE ODDS OF WINNING THE POWERBALL LOTTERY ARE 80 MILLION TO ONE.
THIS COULD BE MY LUCKY DAY!

Log-odds/Logit

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

Odds

$$\boxed{\frac{p(X)}{1-p(X)}} = e^{\beta_0+\beta_1 X}$$

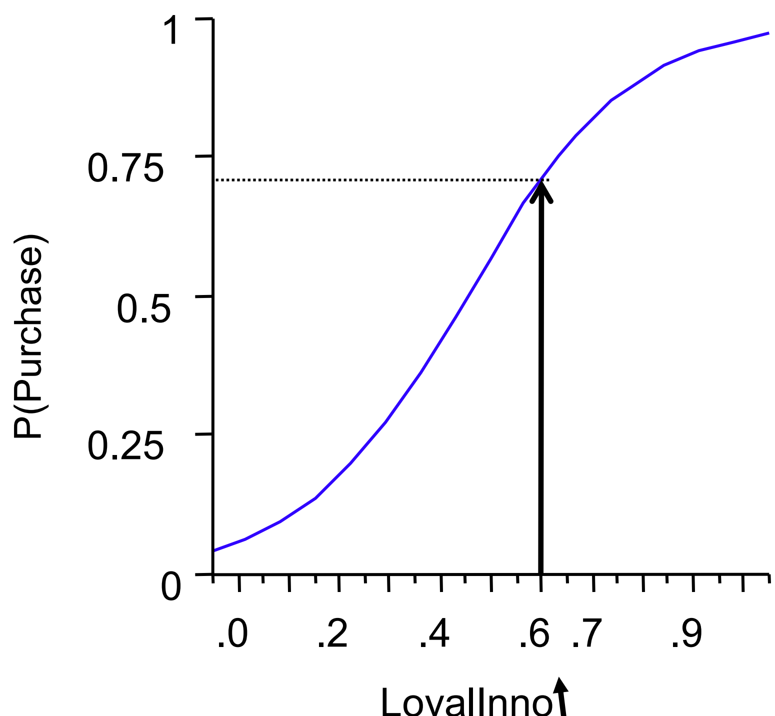
Log-odds or logit

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

- The logistic regression model has a logit that is linear in X.

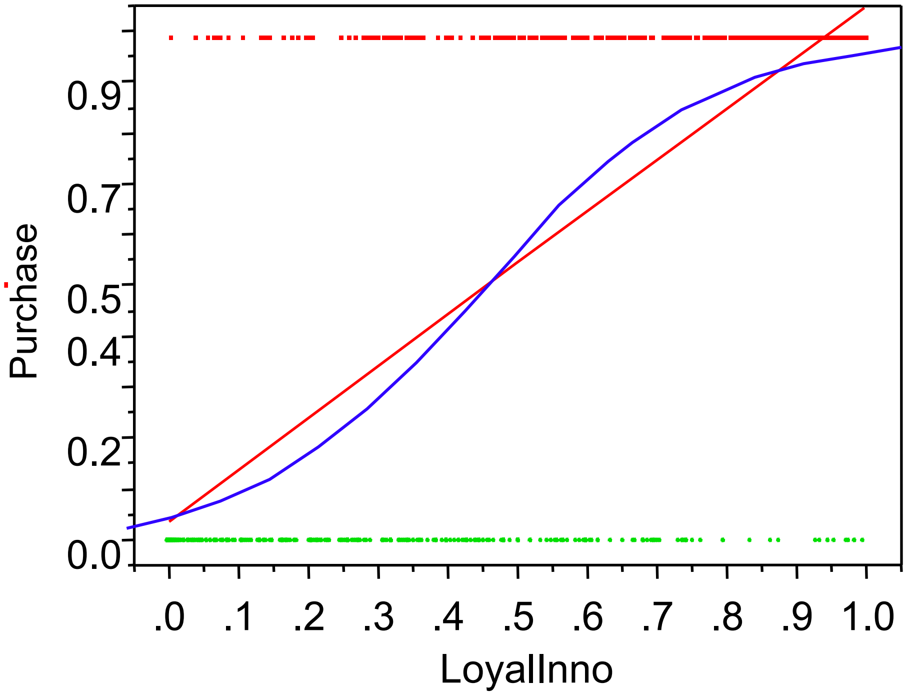
Logistic Regression

- Logistic regression is very similar to linear regression
- We come up with $\hat{\beta}_0$ and $\hat{\beta}_1$ to estimate β_0 and β_1
  - How? (Q1)
- We have similar problems and questions as in linear regression, e.g. (Q2)
  - Is β_1 equal to 0?
  - How sure are we about our guesses for β_0 and β_1?
- How to make predictions? (Q3)



If LoyalInno is about .6 then Pr(Inno=1) ≈ .7

Linear vs Logistic Regression



Q1: How to estimate coefficients?

- Recall in linear regression, we use least squares coefficient estimates
- Here, we use a method called maximum likelihood

- Intuition: we try to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that plugging these estimates into

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

yields

a number close to 1 for all individuals who chose Innocent, and

a number close to 0 for all individuals who did not choose Innocent.

- Formally, to maximise the following likelihood function

$$\boxed{l(\beta_0,\beta_1) = \boxed{\prod_{i:y_i=1} p(x_i)} \boxed{\prod_{j:y_j=0} (1-p(x_j))}}$$

Notations:

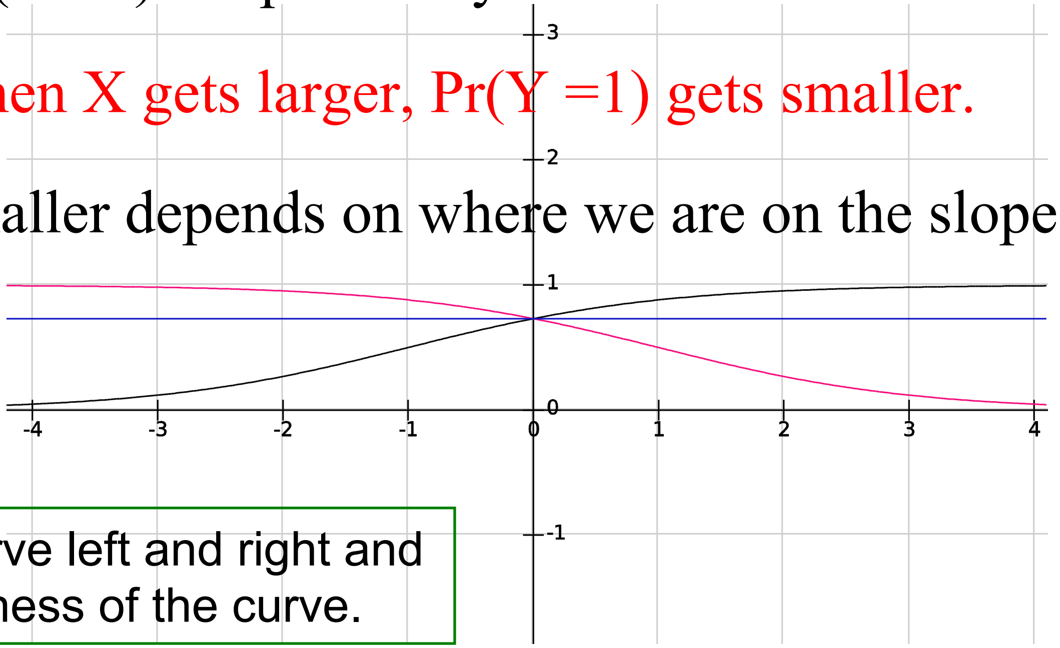
$\sum x_i$ add all x_i

$\prod x_i$ multiply all x_i

Interpreting β_1

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting Pr(Y=1) and not Y.
- If $\beta_1 = 0$, this means that there is no relationship between Y and X.
- If $\beta_1 > 0$, this means that when X gets larger so does the probability that Y = 1, that is, X and Pr(Y =1) are positively correlated.
- If $\beta_1 < 0$, this means that when X gets larger, Pr(Y =1) gets smaller.
- But how much bigger or smaller depends on where we are on the slope.

Blue: $\beta_0=1,\ \beta_1=0$
Black: $\beta_0=1,\ \beta_1=1$
Red: $\beta_0=1,\ \beta_1=-1$



The constant (β_0) moves the curve left and right and the slope (β_1) defines the steepness of the curve.

Q2: Are the coefficients significant?

- We still want to perform a hypothesis test to see whether we can be sure that β_1 is significantly different from zero.
- H_0: $\beta_1 = 0$? --- the null hypothesis
- We use a z test instead of a t test
  - z-statistics associated with β_1 is $\dfrac{\widehat{\beta}_1}{SE(\widehat{\beta}_1)}$

$$\boxed{t = \frac{\widehat{\beta}_1}{SE(\widehat{\beta}_1)}}$$

- This doesn't change the way we interpret the p-value.
  - If p-value is tiny, reject H_0 → there is relationship between X and Pr(Y=1)
  - Otherwise, accept H_0 → there is no relationship between X and Pr(Y=1)

Q3: Making Prediction

- Suppose an individual has a loyalty of 0.1. What is the probability of buying Innocent?

- $\hat{\beta}_0$= - 2.91 and $\hat{\beta}_1$= 6.26

- The predicted probability of purchasing Innocent juice for an individual with the loyalty of 0.1 is

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-2.91+6.26*0.1}}{1+e^{-2.91+6.26*0.1}} = 0.0924$$

Logistic Regression Outline

- Case: Orange Juice Brand Preference

- Why Not Linear Regression?
- Simple Logistic Regression
  - Logistic Function
  - Interpreting the coefficients
  - Making Predictions

- Case: Credit Card Default Data (A whole running example)
  - Adding Qualitative Predictors
- Multiple Logistic Regression

Case: Credit Card Default Data

- We'd like to be able to predict customers that are likely to default
  - default: fail to adhere to the payment policy on credit card agreement

- Possible X variables are:
  - Annual income
  - Monthly credit card balance
    - 0 (nothing is owed)
    - positive (something is owed)
    - negative (a payment is made over what is owed)

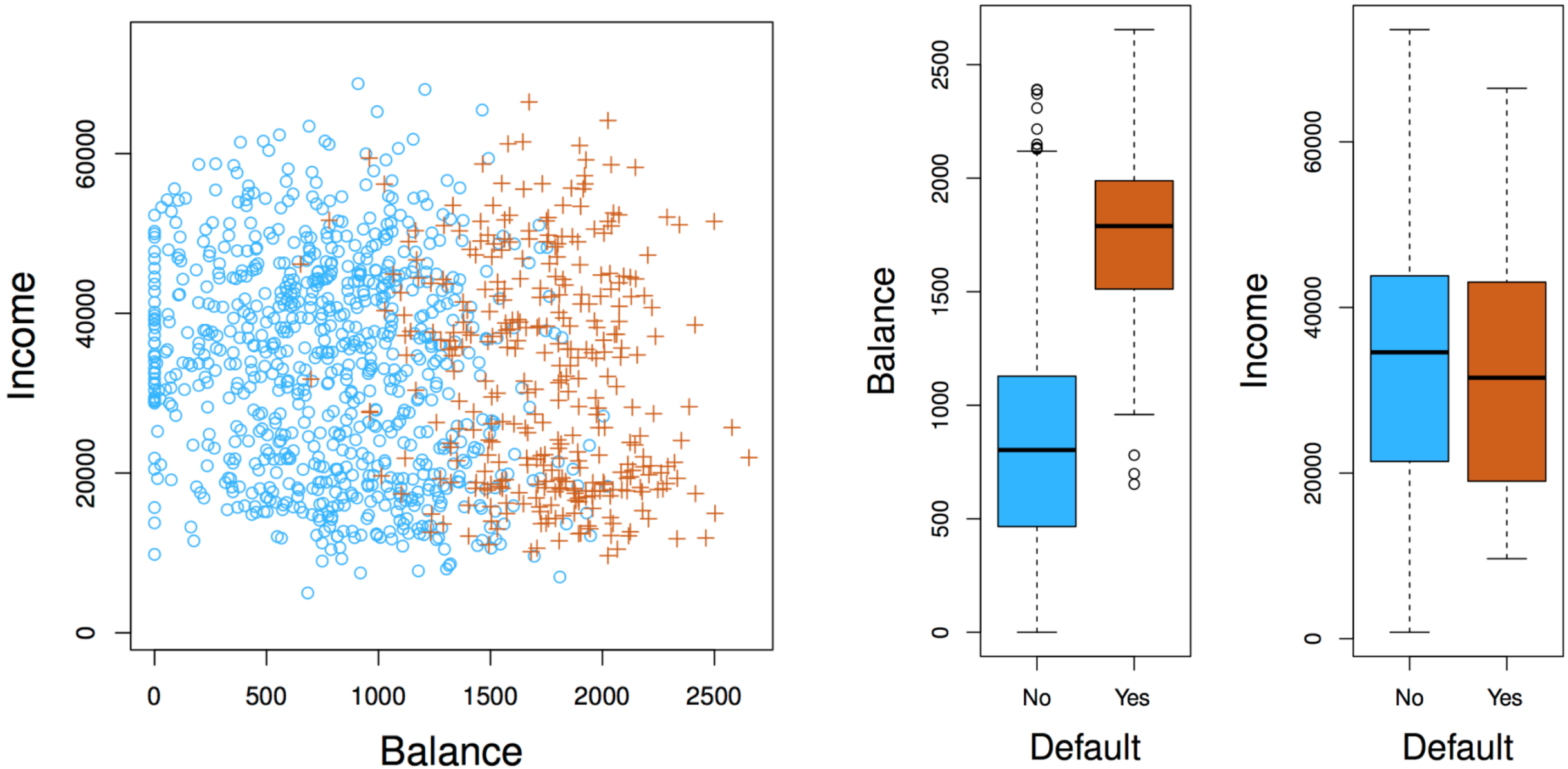
- The Y variable (default) is categorical: Yes or No

- How do we check the relationship between Y and X?

The Default Dataset

Birkbeck
UNIVERSITY OF LONDON

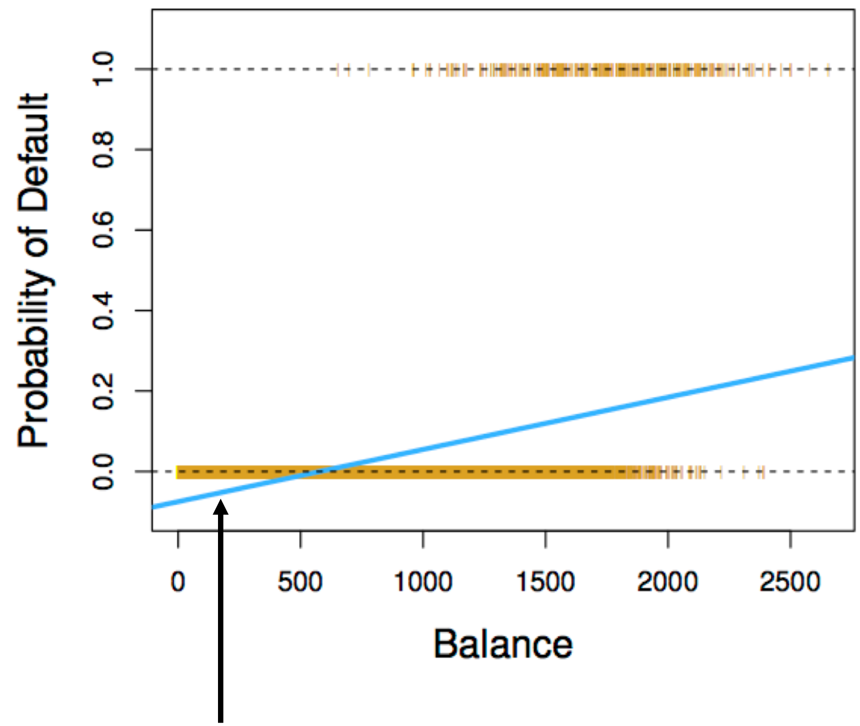
- ?Default to see details




```
install.packages("ISLR")
library(ISLR)
?Default
plot(Default$balance,Default$income,col=ifelse(Default$default=='No',"blue","orange"),pch=ifelse(Default$default=='No',"o","+"))
plot(Default$default,Default$balance,xlab="Default",ylab="Balance")
```

Why not Linear Regression?

- If we fit a linear regression to the Default data, then
  - for very low balances we predict a negative probability
  - for high balances we predict a probability above 1

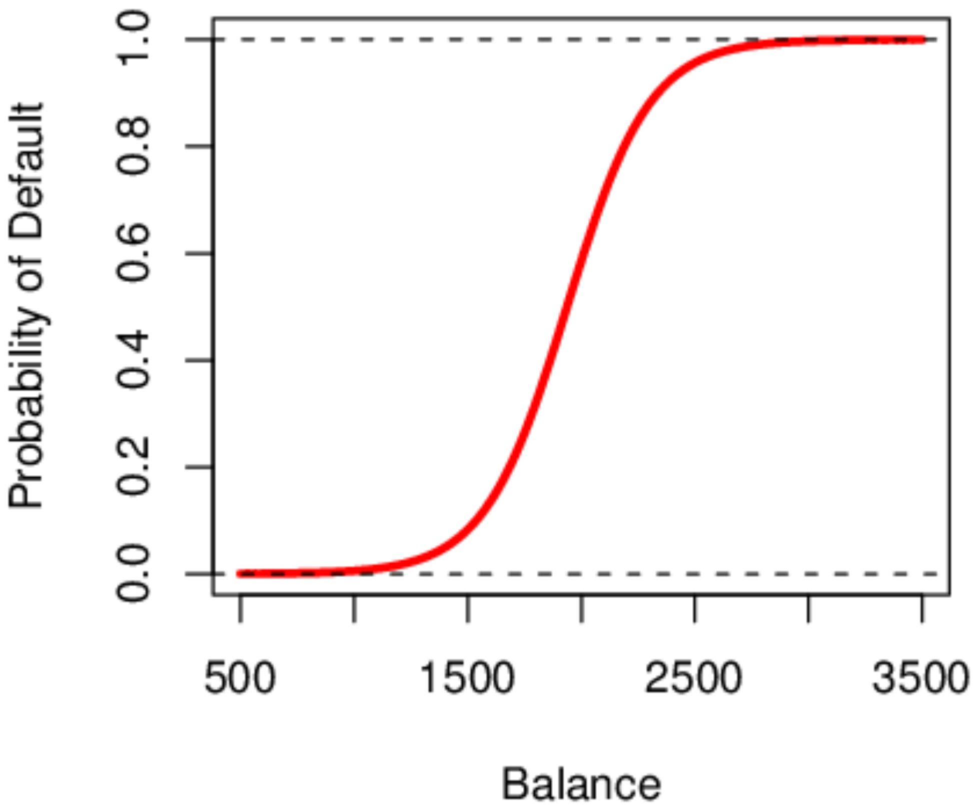


When Balance < 500,

Pr(default) is negative!

Logistic Function on Default Data

- Now the probability of default is
  - close to, but not less than zero for low balances
  - close to, but not above 1 for high balances



Are the coefficients significant?

- The results

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	$\boxed{0.0055}$	0.0002	24.9	$\boxed{< 0.0001}$

- How to read the results from logistic regression?

- Here the p-value for balance is very small, and $\hat{\beta}_1$ is positive, so we are sure that if the balance increase, then the probability of default will increase as well.

Making Prediction

- Suppose an individual has an average balance of $1000. What is their probability of default?
- Write down the expression.
- Recall $\hat{\beta}_0 = -10.6513$ and $\hat{\beta}_1 = 0.0055$

$$\boxed{p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}$$

Making Prediction

- Suppose an individual has an average balance of \$1000. What is their probability of default?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.00576$$

- The predicted probability of default for an individual with a balance of \$1000 is less than 1%.

- For a balance of \$2000, the probability is much higher, and equals to 0.586 (58.6%).

Logistic Regression Outline

- Case: Orange Juice Brand Preference

- Why Not Linear Regression?
- Simple Logistic Regression
  - Logistic Function
  - Interpreting the coefficients
  - Making Predictions

- Case: Credit Card Default Data (A whole running example)
  - Adding Qualitative Predictors
- Multiple Logistic Regression

Qualitative Predictors in Logistic Regression

- We can predict if an individual default by checking if s/he is a student or not. Thus we can use a qualitative variable "student" coded as (Student = 1, Non-student =0). How?

student is a categorical variable. Yes/No

Step 1: Turn it into a numerical variable by as.numeric()

```
student_num <- as.numeric(student)
```

Yes: 2, No: 1

Step 2: Turn it into 1/0

```
student_num_01 <- as.numeric(student) - 1
```

Step 3: Fit a logistic model

```
glm.fit.01<- glm(default~student_num_01, data=Default, family=binomial)
#the same effect as
glm.fit<- glm(default~student, data=Default, family=binomial)
```

Qualitative Predictors in Logistic Regression

- We can predict if an individual default by checking if s/he is a student or not. Thus we can use a qualitative variable “student” coded as (Student = 1, Non-student =0). How?
- $\hat{\beta}_1$ is positive: This indicates students tend to have higher default probabilities than non-students

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

1 means is student

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

0 means is non-student

Logistic Regression Outline

- Case: Orange Juice Brand Preference

- Why Not Linear Regression?
- Simple Logistic Regression
  - Logistic Function
  - Interpreting the coefficients
  - Making Predictions

- Case: Credit Card Default Data (A whole running example)
  - Adding Qualitative Predictors
- Multiple Logistic Regression

Multiple Logistic Regression

- We can fit multiple logistic just like linear regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

Multiple Logistic Regression- Default Data

- Predict Default using:
  - balance (quantitative)
  - income (quantitative)   in thousand pounds
  - student (qualitative)

glm.fit<- glm(default~student+income+balance, data=Default, family=binomial)
summary(glm.fit)

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	$\boxed{0.7115}$
student[Yes]	$\boxed{-0.6468}$	0.2362	-2.74	0.0062

- Which predictors are associated with the probability of default?

Predictions

- A student with a credit card balance of £1,500 and an income of £40K has an estimated probability of default

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times1500+0.003\times40-0.6468\times1}}{1+e^{-10.869+0.00574\times1500+0.003\times40-0.6468\times1}} = 0.058.$$

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

An Apparent Contradiction!

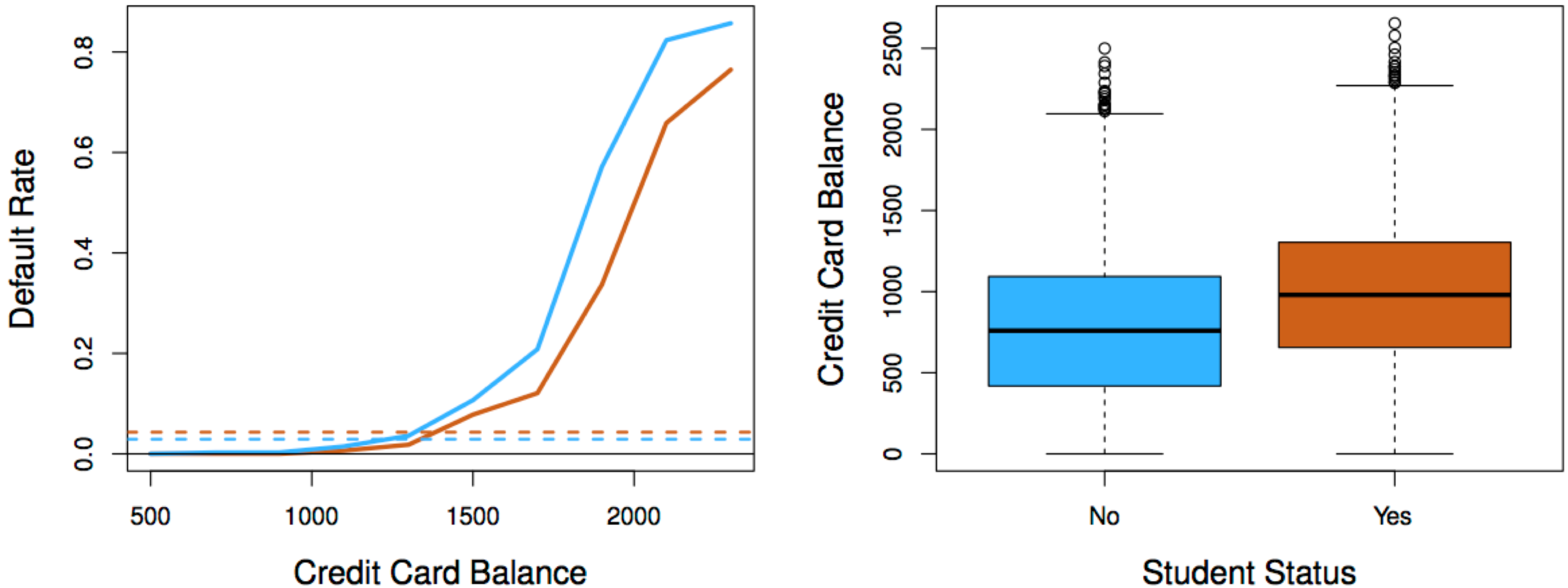
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Positive

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Negative

Non-students (Blue) vs. Students (Orange)



- Left solid lines: Given a balance, a student is less likely to default
- Left broken lines: the default rates (DR) over all values of balances
  - The overall student default rate is higher
- Right: student and balance are correlated!
  - Students are more likely to have large credit card balances (left) → higher DR

To whom should credit be offered?

Birkbeck UNIVERSITY OF LONDON

- A student is riskier than non students if no information about the credit card balance is available

- However, that student is less risky than a non student with the same credit card balance!

➢ The results obtained using one predictor may be quite different from those obtained using multiple predictors, especially when there is correlation among the predictors.

LAB

Logistic Regression

The Stock Market Data

- The data set consists of percentage returns for the S&P 500 stock index over 1,250 days, from the beginning of 2001 until the end of 2005.

- For each day, the following are recorded:
  - `Year`: in which year the day is
  - `Lag1,…,Lag5`: The % returns for each of the 5 previous trading days
  - `Volumn`: the number of shares traded on the previous day (in billion)
  - `Today`: the percentage return on the date in question
  - `Direction`: whether the market was Up or Down on this day

The Stock Market Data

- The `cor()` function produces a matrix that contains all the pairwise correlations among the predicators in a data set

```
> cor(Smarket[,-9])
```

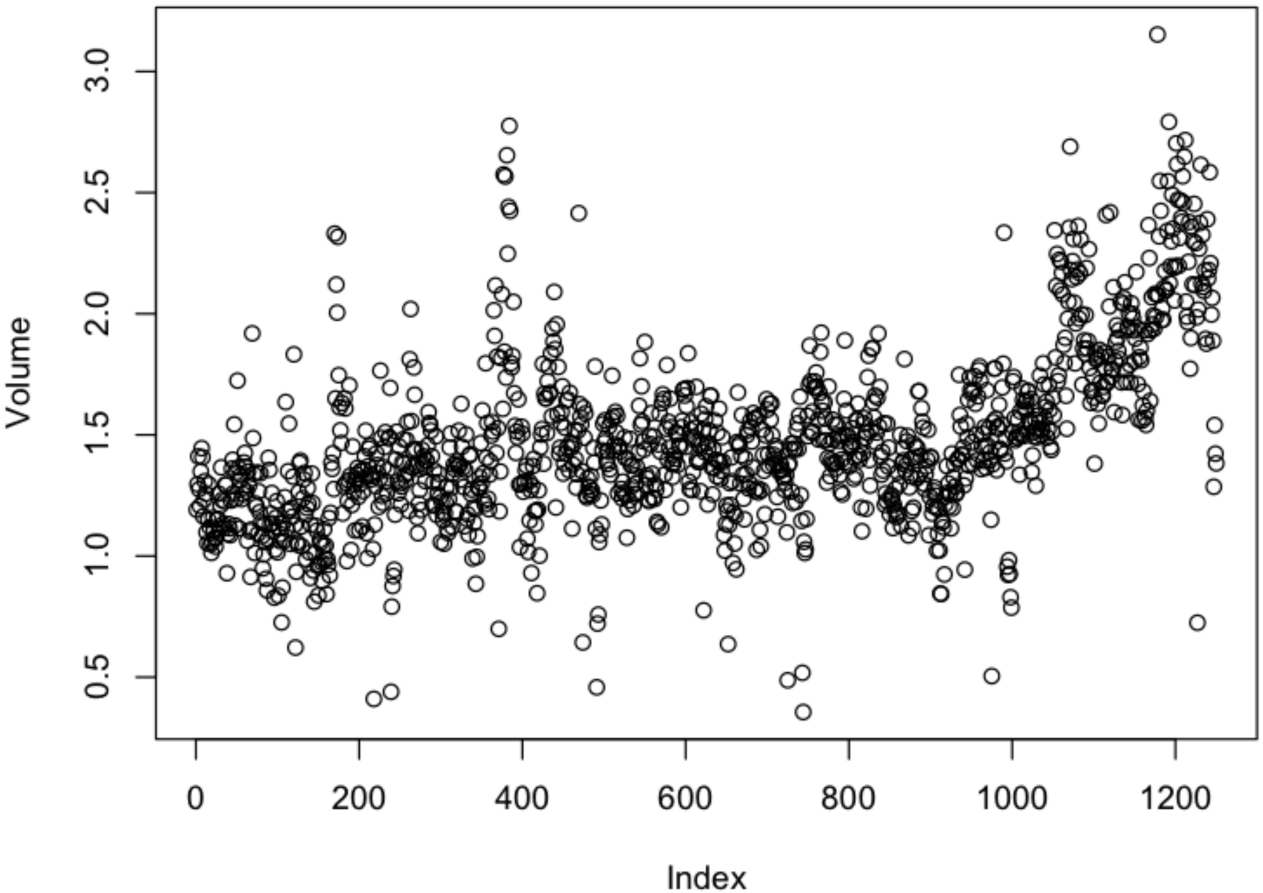
	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year	1.00000000	0.029699649	0.030596422	0.033194581	0.035688718	0.029787995	0.53900647	0.030095229
Lag1	0.02969965	1.000000000	-0.026294328	-0.010803402	-0.002985911	-0.005674606	0.04090991	-0.026155045
Lag2	0.03059642	-0.026294328	1.000000000	-0.025896670	-0.010853533	-0.003557949	-0.04338321	-0.010250033
Lag3	0.03319458	-0.010803402	-0.025896670	1.000000000	-0.024051036	-0.018808338	-0.04182369	-0.002447647
Lag4	0.03568872	-0.002985911	-0.010853533	-0.024051036	1.000000000	-0.027083641	-0.04841425	-0.006899527
Lag5	0.02978799	-0.005674606	-0.003557949	-0.018808338	-0.027083641	1.000000000	-0.02200231	-0.034860083
Volume	**0.53900647**	0.040909908	-0.043383215	-0.041823686	-0.048414246	-0.022002315	1.00000000	0.014591823
Today	0.03009523	-0.026155045	-0.010250033	-0.002447647	-0.006899527	-0.034860083	0.01459182	1.000000000

- The correlations between `Today` and `Lag`*n* are close to zero
  - little correlation between today's returns and previous days' returns
- The correlations between Year and Volume are substantial
  - How are they correlated?

The Stock Market Data

- Plot the data

```
> plot(Smarket$Volume)
```



Logistic Regression

- Goal: fit a logistic regression model in order to predict `Direction` using `Lag1, …, Lag5` and `Volume`.
- The `glm()` function fits generalised linear models (including logistic regression)
  - Similar to `lm()`, but must pass in the argument `family=binomial` to tell R to run a logistic regression

Logistic Regression

```
> glm.fit <- glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
                 data=Smarket,family=binomial)
> summary(glm.fit)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.446   -1.203   1.065    1.145    1.326

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392
```

- The smallest p-value: `Lag1`
- Its negative coefficient suggests that
  - if the market had a positive return yesterday
  - then it is less likely to go up today
- 0.145 is still relatively large, so no clear evidence of a real association between `Lag1` and `Direction`.

Make A Prediction

```
> glm.probs <- predict(glm.fit,type="response")
> glm.probs[1:10]  % only print out first 10 probabilities
        1         2         3         4         5         6         7         8         9        10 
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 0.5092292 0.5176135 0.4888378
> contrasts(Smarket$Direction)
     Up
Down  0
Up    1
```

- `type="response"` tells R to output the probabilities of the form P(Y=1|X)

- If no data is provided to the `predict()` function, then it will predict the training data

- These probabilities correspond to the probability of market going up
  - `contrasts()` function indicates that R has created a dummy variable with 1 for `Up`

Make A Prediction

- Now predict whether the market will go up or down on a particular day
  - Need to convert the predicted probabilities into class labels: `Up` or `Down`
  - `> glm.pred <- rep("Down",1250)` → create a vector of 1250 Down elements
  - `> glm.pred[glm.probs>.5] <- "Up"` → transform those with prob >.5 to Up

- Given these predictions, use `table()` function
  - to produce a confusion matrix
  - to determine how many observations were in/correctly classified

```
> table(glm.pred,Smarket$Direction)
        Direction
glm.pred Down  Up
    Down  145 141
    Up    457 507
> (507+145)/1250
[1] 0.5216
> mean(glm.pred==Smarket$Direction)
[1] 0.5216
```

LR correctly predicted the movement of the market 52.2% of the time

Improve the Performance

- The above result train and test the same data set, this might
  - Overestimate the training error
  - Underestimate the test error

- Train the model corresponding to 01-04 data & test it on 05 data
  - To get a more realistic error rate

Step 1: Create vectors for training & test data

```
> train <- (Year < 2005)
> Smarket.2005 <- Smarket[!train,]
> dim(Smarket.2005)
[1] 252   9
> Direction.2005 <- Smarket$Direction[!train]
```

Improve the Performance

Step 2: fit the model using logistic regression

```
> glm.fit <- glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
                 data=Smarket,   family=binomial, subset=train)
> glm.probs <- predict(glm.fit, Smarket.2005, type="response")
```

Step3: compute the predictions for 2005

```
> glm.pred <- rep("Down", 252)
> glm.pred[glm.probs>.5] <- "Up"
```

Step 4: compare the predictions to the actual movements for 2005

```
> table(glm.pred, Direction.2005)
        Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
> mean(glm.pred == Direction.2005)
[1] 0.4801587
> mean(glm.pred != Direction.2005)
[1] 0.5198413
```

Remove Irrelevant Predictors

- Large p-value means irrelevancy
  - Irrelevant predictors will deteriorate the test error rate
  - Remove them to improve the model

```
> glm.fit <- glm(Direction~Lag1+Lag2, data=Smarket, family=binomial, subset=train)
> glm.probs <- predict(glm.fit,Smarket.2005,type="response")
> glm.pred <- rep("Down",252)
> glm.pred[glm.probs>.5] <- "Up"
> table(glm.pred,Direction.2005)
        Direction.2005
glm.pred Down  Up
    Down   35  35
    Up     76 106
> mean(glm.pred==Direction.2005)
[1] 0.5595238

> 35/(35+35)
[1] 0.5

> 106/(106+76)
[1] 0.5824176
```

```
Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392
```

Predicting Particular Values

- We can predict the returns associated with particular values of `Lag1` and `Lag2`
  - Predict Direction on a day when
    - `Lag1=1.2` and `Lag2=1.1`
    - `Lag1=1.5` and `Lag2=-0.8`

```
predict(glm.fit,
        newdata = data.frame(Lag1=c(1.2,1.5), Lag2=c(1.1,-0.8)),
        type = "response")
            1         2
    0.4791462 0.4960939
```