

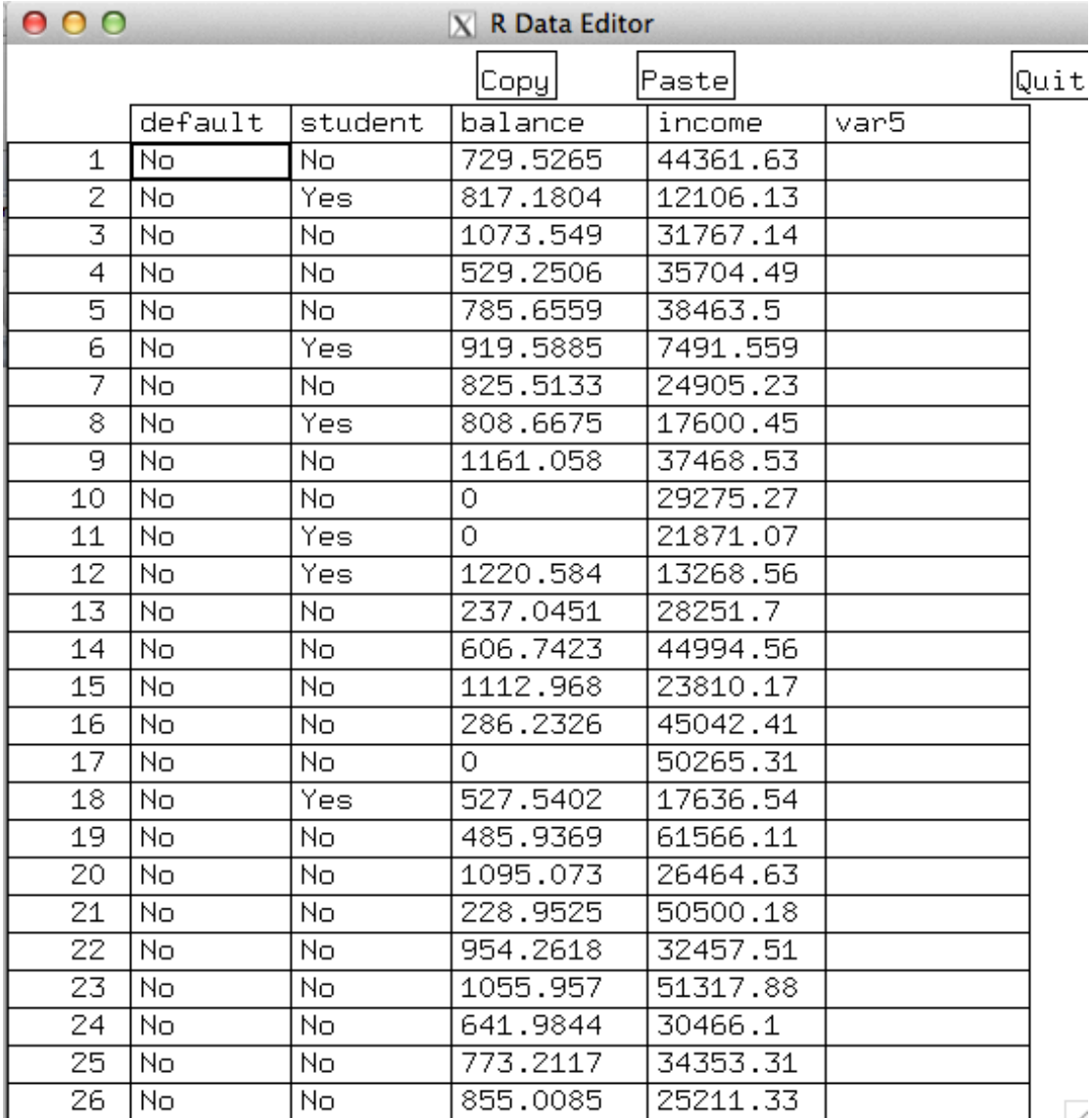
#load the libraries

> library(MASS)

> library(ISLR)

#view data set

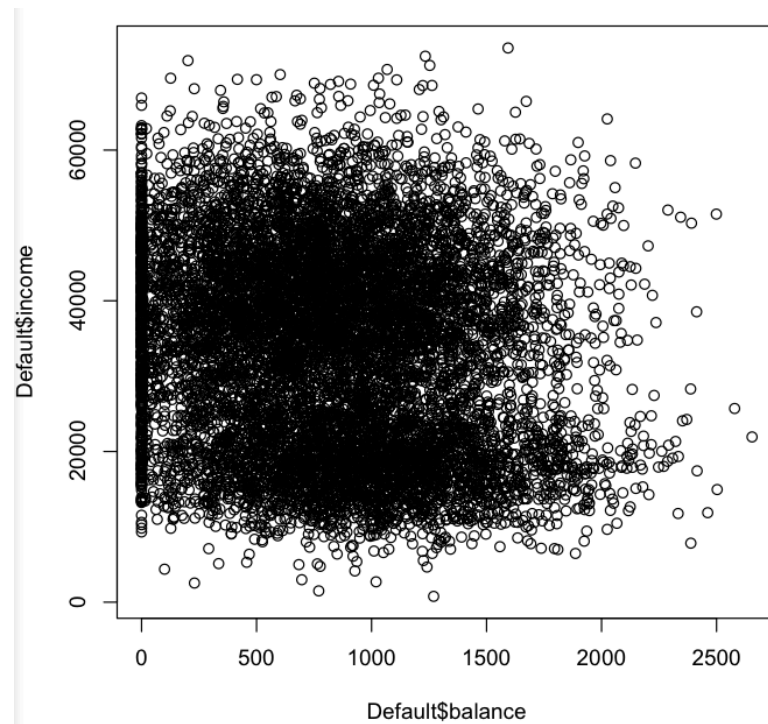
> fix(Default)



	default	student	balance	income	var5
1	No	No	729.5265	44361.63	
2	No	Yes	817.1804	12106.13	
3	No	No	1073.549	31767.14	
4	No	No	529.2506	35704.49	
5	No	No	785.6559	38463.5	
6	No	Yes	919.5885	7491.559	
7	No	No	825.5133	24905.23	
8	No	Yes	808.6675	17600.45	
9	No	No	1161.058	37468.53	
10	No	No	0	29275.27	
11	No	Yes	0	21871.07	
12	No	Yes	1220.584	13268.56	
13	No	No	237.0451	28251.7	
14	No	No	606.7423	44994.56	
15	No	No	1112.968	23810.17	
16	No	No	286.2326	45042.41	
17	No	No	0	50265.31	
18	No	Yes	527.5402	17636.54	
19	No	No	485.9369	61566.11	
20	No	No	1095.073	26464.63	
21	No	No	228.9525	50500.18	
22	No	No	954.2618	32457.51	
23	No	No	1055.957	51317.88	
24	No	No	641.9844	30466.1	
25	No	No	773.2117	34353.31	
26	No	No	855.0085	25211.33	

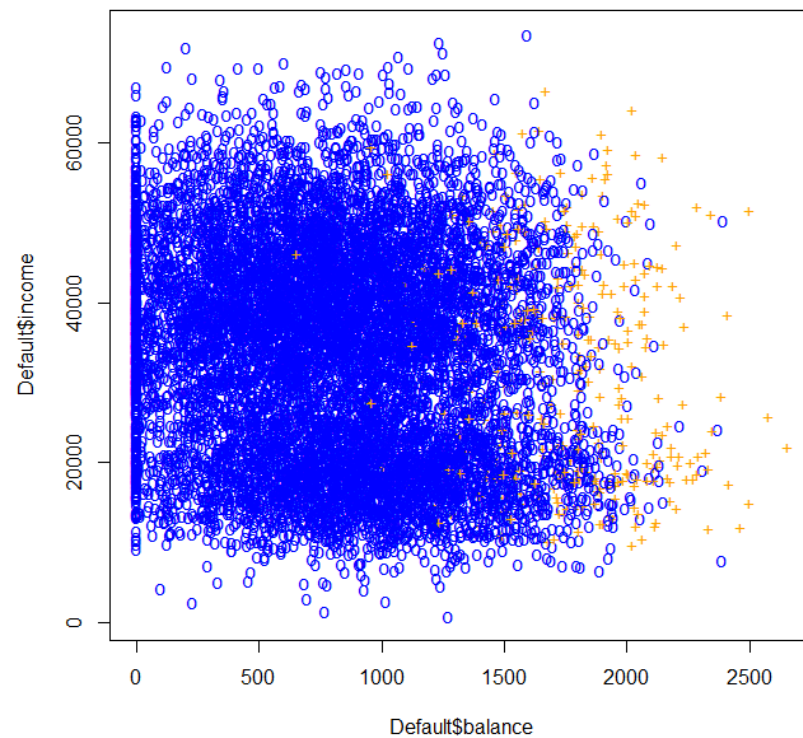
#first trial of plot the relationship between balance and #income

```
> plot(Default$balance,Default$income)
```



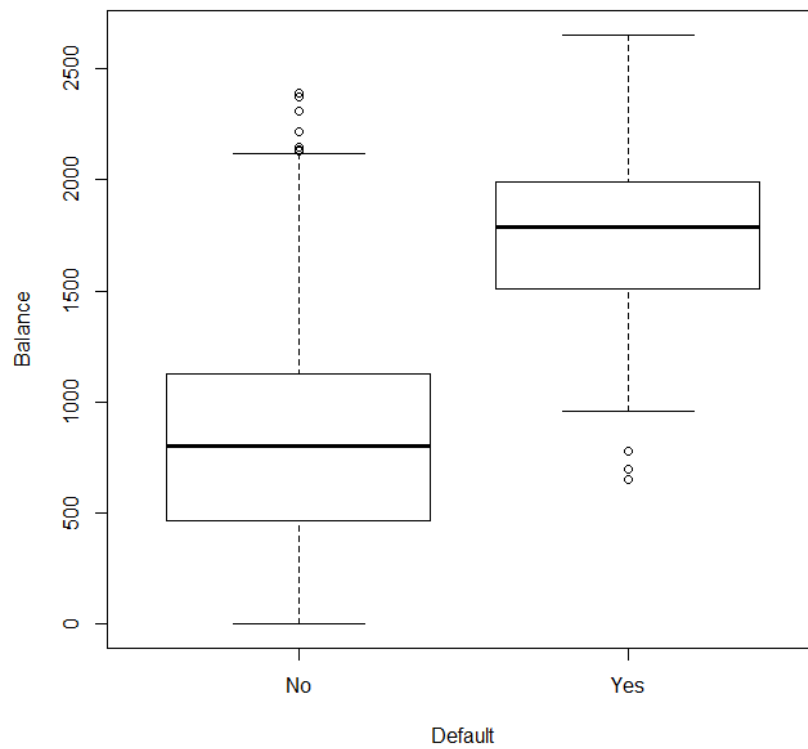
# adding colors and legends. Blue circles are for not default and orange + are for default.

```
> plot(Default$balance,Default$income,col=ifelse(Default$default=='No',"blue","orange"),pch=ifelse(
Default$default=='No',"o","+"))
```



# Plot the relationship between default and balance.

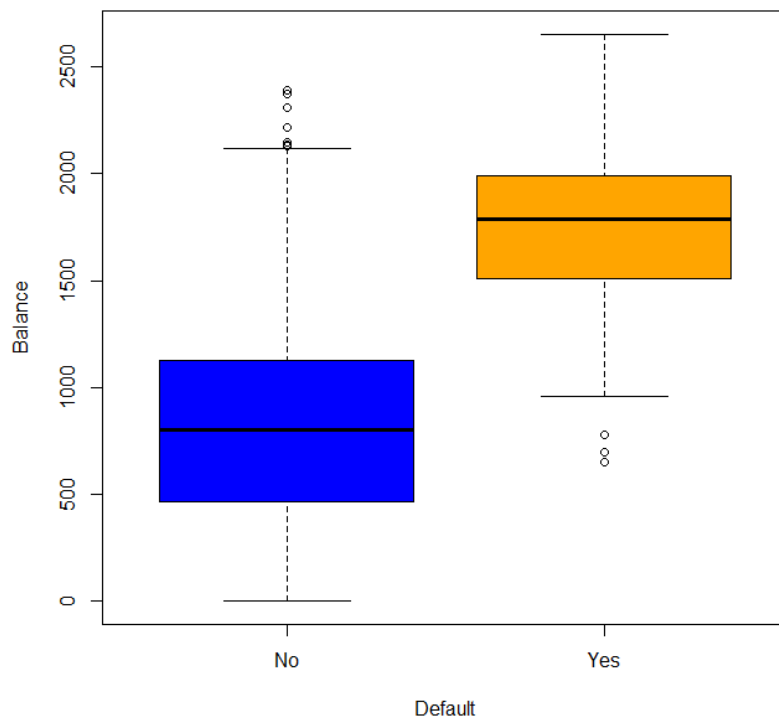
```
> plot(Default$default,Default$balance,xlab="Default",ylab="Balance")
```



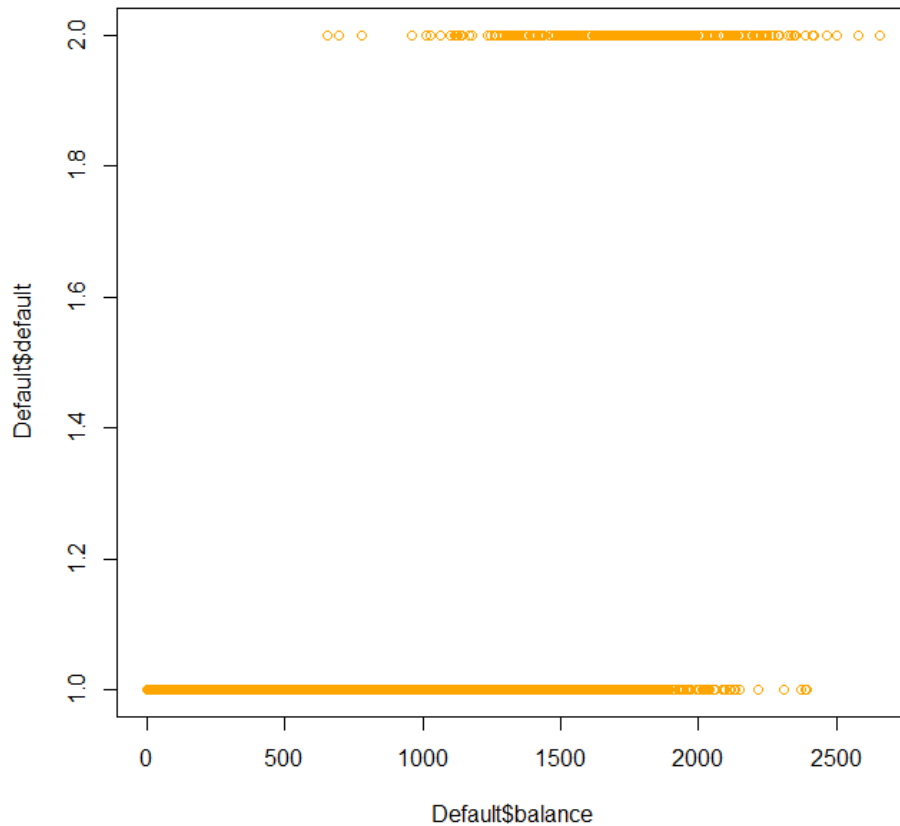
#adding colors by predefining a sequence of colors.

```
> colors=c("blue","orange")
```

```
> plot(Default$default,Default$balance,xlab="Default",ylab="Balance",col=colors)
```



```
> plot(Default$balance,Default$default,col="orange")
```



#as.numeric turns a qualitative variable to a quantitative variable. However, R will automatically assign 1 to “No” and 2 to “Yes”.

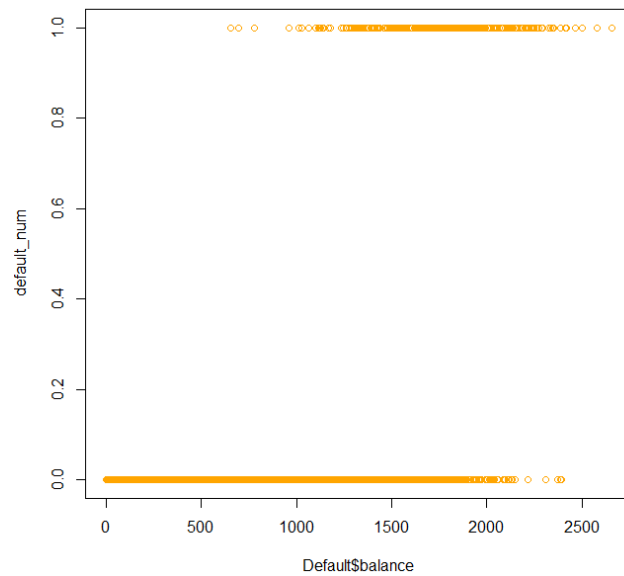
...

#So we subtract 1 from all the default value and make it a new vector default num.

```
[1]00000000000000000000000000000000000000000000000000000  
[37]00000000000000000000000000000000000000000000000000000  
[73]00000000000000000000000000000000000000000000000000000  
[109]000000000000000000000000000000000000000001000000000  
[145]00000000000000000000000000000000000000000100
```

# Now the following plot shows 0's and 1's

```
> plot(Default$balance,default_num,col="orange")
```



#Next we will plot the linear regression line. We need to build the linear regression model first.

```
> lm(default_num~ balance,data=Default)
```

Call:

```
lm(formula = default_num ~ balance, data = Default)
```

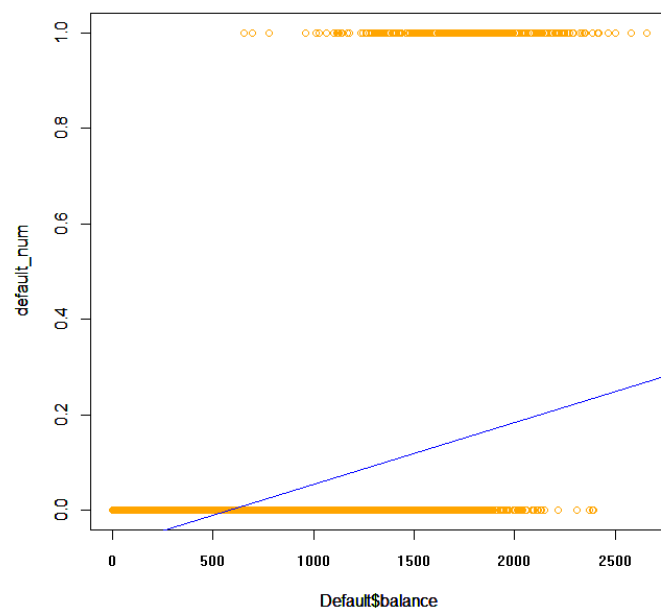
Coefficients:

```
(Intercept) balance  
-0.0751920    0.0001299
```

#the linear regression model is called fit\_linear and we plot the line by abline

```
> fit_linear<-lm(default_num~ balance,data=Default)
```

```
> abline(fit_linear,col="blue")
```



# Next, we build the logistic regression model

```
> glm.fit=glm(Default$default~Default$balance,data=Default,family=binomial)
# abline(glm.fit)
> summary(glm.fit)
```

Call:

```
glm(formula = Default$default ~ Default$balance, family = binomial,
    data = Default)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.2697	-0.1465	-0.0589	-0.0221	3.7589

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-1.065e+01	3.612e-01	-29.49	<2e-16 ***
Default\$balance	5.499e-03	2.204e-04	24.95	<2e-16 ***

---

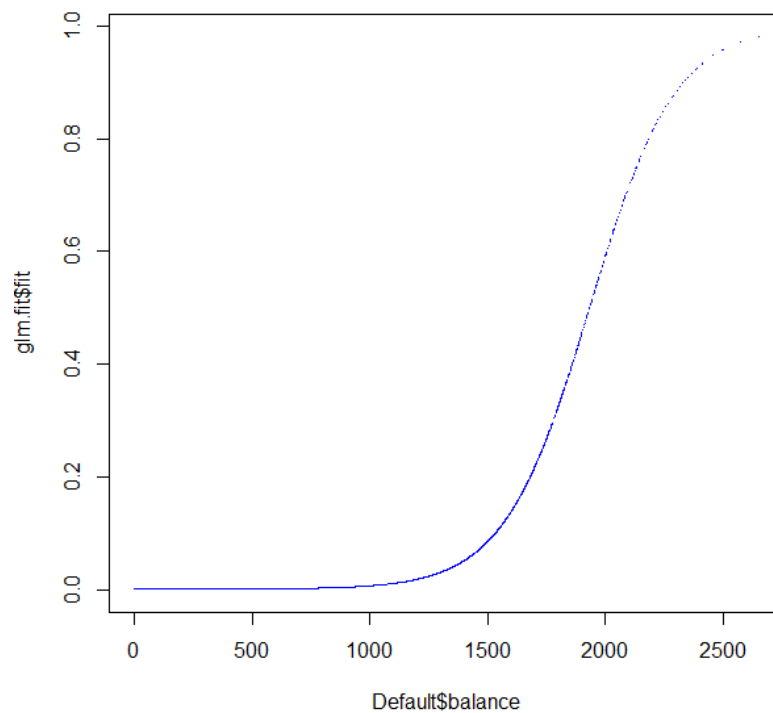
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2920.6 on 9999 degrees of freedom  
Residual deviance: 1596.5 on 9998 degrees of freedom  
AIC: 1600.5

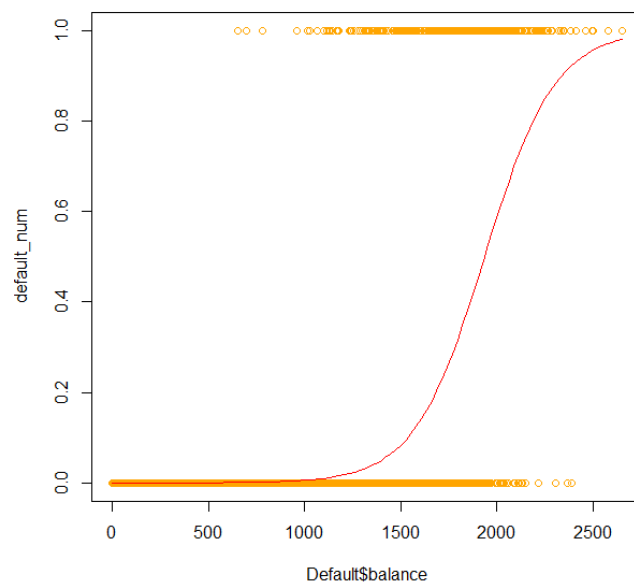
Number of Fisher Scoring iterations: 8

```
# we will plot the result of the logistic regression. 3 ways to do it.
#The first way is a bit 'cheating' as instead of a smooth curve, we plot the dots.
> plot(Default$balance,glm.fit$fit,col="blue",pch=".")
```



#The second way is by using the inverse function of logit. This function will take  $\hat{\beta}_0 + \hat{\beta}_1 X$  as input and return  $p(X)$

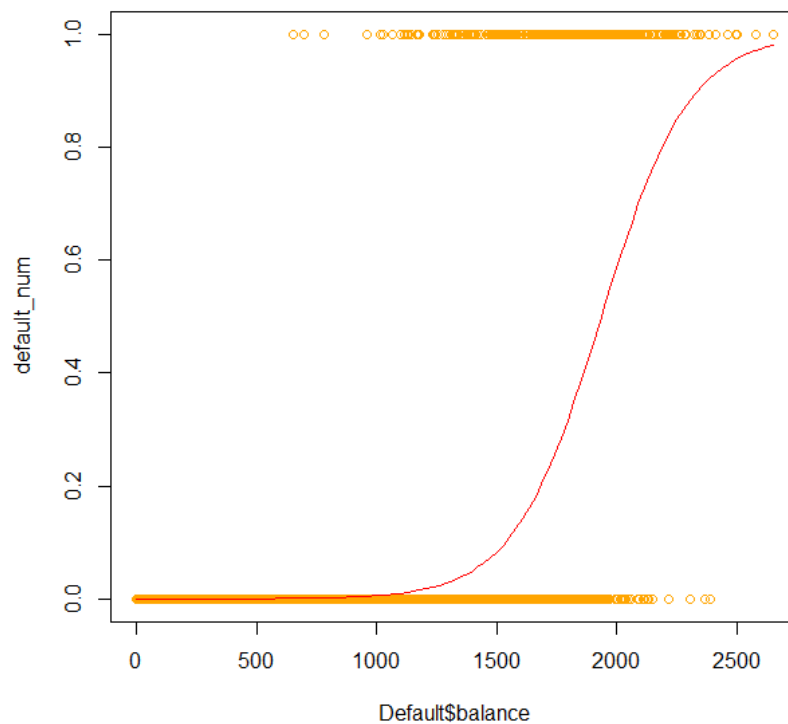
```
> plot(Default$balance,default_num,col="orange")
#we take sample numbers from balance
> xrange=seq(min(Default$balance),max(Default$balance),length.out=100)
> library(boot)
> lines(xrange,inv.logit(glm.fit$coef[1]+glm.fit$coef[2]*xrange),col="red")
```



#The third way is making prediction first and plot the predicted value as a smooth line

```
> y<-Default$default
> x<-Default$balance
> glm.fit_1=glm(y~x,family=binomial)
#making predictions
> yrange<-predict(glm.fit_1,data.frame(x=xrange),type="response")
> lines(xrange,yrange,col="red")
```

#The third way is the preferred way.



#We will show how to predict using the logistic model.

# We predict the prob of default when an individual has an average balance of 1000 and 2000.

```
>glm.fit=glm(default~balance, data=Default, family=binomial)
>newy=predict(glm.fit, data.frame(balance=c(1000,2000)), type="response")
> newy
      1      2
0.005752145 0.585769370
```



```
#The following code builds a logistic regression model between two qualitative variables.  
> glm.fit_student<-glm(default~ student,data=Default,family=binomial)  
> summary(glm.fit_student)
```

Call:

```
glm(formula = default ~ student, family = binomial,  
    data = Default)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.2970	-0.2970	-0.2434	-0.2434	2.6585

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-3.50413	0.07071	-49.55	< 2e-16 ***
Default\$studentYes	0.40489	0.11502	3.52	0.000431 ***

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2920.6 on 9999 degrees of freedom  
Residual deviance: 2908.7 on 9998 degrees of freedom  
AIC: 2912.7

Number of Fisher Scoring iterations: 6

```
#The following code predicts the prob of default given student or non-student
```

```
#There are two ways of doing this:
```

```
#The first is to use the student model directly (the student model is built from two qualitative variables)
```

```
glm.fit.student=glm(default~student, data=Default, family=binomial)  
newy=predict(glm.fit.student, data.frame(student=c("No","Yes")), type="response")
```

```
#The second is to use make student as a numeric value and predict using the student_num model.
```

```
student_num=as.numeric(Default$student)-1  
glm.fit.student.num=glm(default~student_num, data=Default, family=binomial)  
newy=predict(glm.fit.student.num, data.frame(student_num_01=c(0,1)), type="response")
```

```
#The following code builds a multiple logistic regression model
> glm.fit_multi<-glm(default~balance+income+student,data=Default,family=binomial)
> summary(glm.fit_multi)
```

Call:

```
glm(formula = default ~ balance + income + student, family = binomial,
    data = Default)
```

Deviance Residuals:

```
    Min      1Q  Median      3Q     Max
-2.4691 -0.1418 -0.0557 -0.0203  3.7383
```

Coefficients:

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080 < 2e-16 ***
balance      5.737e-03  2.319e-04  24.738 < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 2920.6 on 9999 degrees of freedom
Residual deviance: 1571.5 on 9996 degrees of freedom
AIC: 1579.5
```

Number of Fisher Scoring iterations: 8

#The following code predicts the prob of default given a student with balance of 1500 and an income of 40,000

```
> predict(glm.fit_multi,data.frame(student="Yes",balance=1500,income=40000), type="response")
1
0.05788194
```

# This is when we input 40 as income (in k-pounds).

```
> predict(glm.fit_multi,data.frame(student="Yes",balance=1500,income=40), type="response")
1
0.05161531
```

Therefore, the figure on the slides is not correct. It should be 0.052 instead of 0.058.

$$\hat{p}(X) = \frac{e^{-10.869+0.00574 \times 1500+0.003 \times 40-0.6468 \times 1}}{1 + e^{-10.869+0.00574 \times 1500+0.003 \times 40-0.6468 \times 1}} = 0.058 \quad \times$$