Ch 5 Homework

Julia Van Dyke

January 19, 2015

5

a

```
set.seed(22)
library(ISLR)
attach(Default)
glm.default <- glm(default~income+balance, family=binomial)
summary(glm.default)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 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: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

b

```
train=sample(10000,5000)
lm.fit=glm(default~income+balance, family=binomial, subset=train)
summary(lm.fit)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = binomial,
##     subset = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2138  -0.1552  -0.0640  -0.0237   3.6744
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.140e+01  5.931e-01 -19.221  < 2e-16 ***
## income       2.421e-05  6.925e-06   3.496 0.000473 ***
## balance      5.557e-03  3.088e-04  17.995  < 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: 1530.39  on 4999  degrees of freedom
## Residual deviance:  847.44  on 4997  degrees of freedom
## AIC: 853.44
##
## Number of Fisher Scoring iterations: 8
```

```
mean((default-predict(lm.fit,Default))[-train]^2)
```

```
## Warning in Ops.factor(default, predict(lm.fit, Default)): '-' not
## meaningful for factors

## Warning in Ops.factor(default, predict(lm.fit, Default)): '-' not
## meaningful for factors

## Warning in Ops.factor(default, predict(lm.fit, Default)): '-' not
## meaningful for factors

## [1] NA
```

c

```
train=sample(10000,5000)
lm.fit=glm(default~income+balance, family=binomial, subset=train)
mean((predict(lm.fit,Default))[-train]^2)
```

```
## [1] 40.46199
```

c

```r
train2=sample(10000,5000)
lm.fit=glm(default~income+balance, family=binomial, subset=train2)
mean((predict(lm.fit,Default))[-train2]^2)
```

```
## [1] 41.18329
```

```r
train3=sample(10000,5000)
lm.fit=glm(default~income+balance, family=binomial, subset=train3)
mean((predict(lm.fit,Default))[-train3]^2)
```

```
## [1] 50.95137
```

```r
train4=sample(10000,5000)
lm.fit=glm(default~income+balance, family=binomial, subset=train4)
mean((predict(lm.fit,Default))[-train4]^2)
```

```
## [1] 44.21473
```

d

```r
glm.default2 <- glm(default~income+balance+student,family=binomial)
summary(glm.default2)
```

```
##
## Call:
## glm(formula = default ~ income + balance + student, family = binomial)
##
## 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 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## 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
```

```
train=sample(10000,5000)
lm.fit2=glm(default~income+balance+student, family=binomial, subset=train)
mean((predict(lm.fit2,Default))[-train]^2)
```

```
## [1] 44.57152
```

the added variable doesn't seem to make much difference in the test error rate.

8

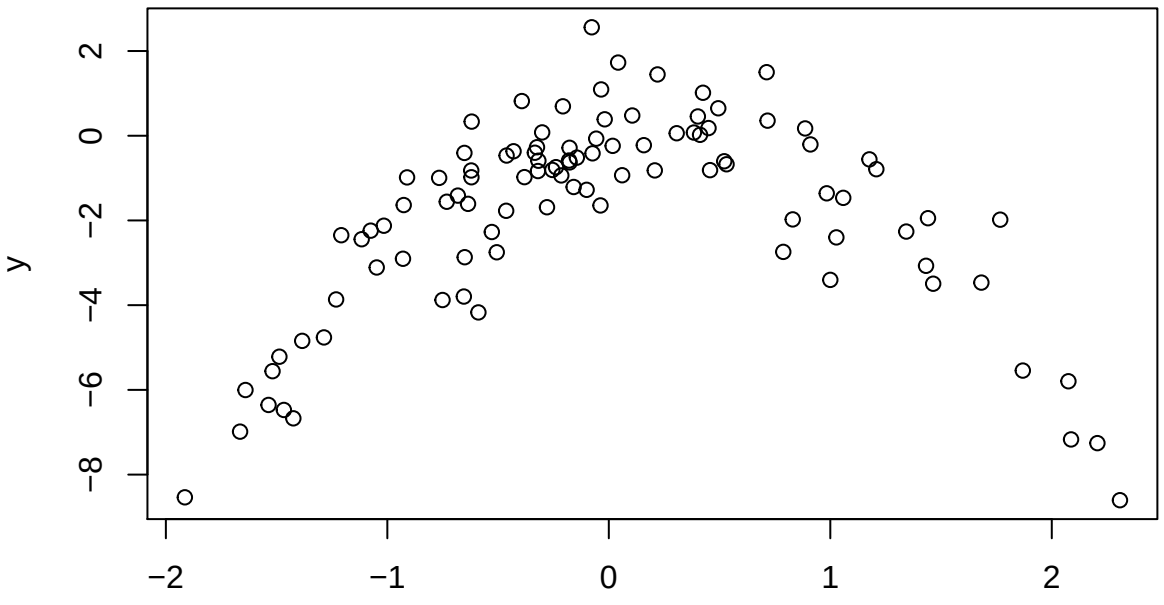
a

```
set.seed(1)
y=rnorm(100)
x=rnorm(100)
y=x-2*x^2+rnorm(100)
```

n=100, p=2

b

```
plot(x,y)
```



#The data seems to follow a curve. y is highest when x is around 0, and lowest when x is around -2 or 2.