CS6220: Data mining techniques
Categorical response

Olga Vitek

October 22, 2015

Outline

Two-way tables

Read data

```
> X <- data.frame(y=c(178, 138, 108, 570, 648,
+       442, 138, 252, 252),
+       belief=rep(c("1-Fundam", "2-Moder", "3-Liber"), 3),
+       degree=rep(c("1-<HS", "2-HS", "3-BS/grad"), 1, each=3)
+       )
> X
```

```
    y   belief    degree
1 178 1-Fundam     1-<HS
2 138  2-Moder     1-<HS
3 108  3-Liber     1-<HS
4 570 1-Fundam      2-HS
5 648  2-Moder      2-HS
6 442  3-Liber      2-HS
7 138 1-Fundam 3-BS/grad
8 252  2-Moder 3-BS/grad
9 252  3-Liber 3-BS/grad
```

Reformat data

```
> ov <- xtabs(y ~ degree+belief, data=X)
> ov
```

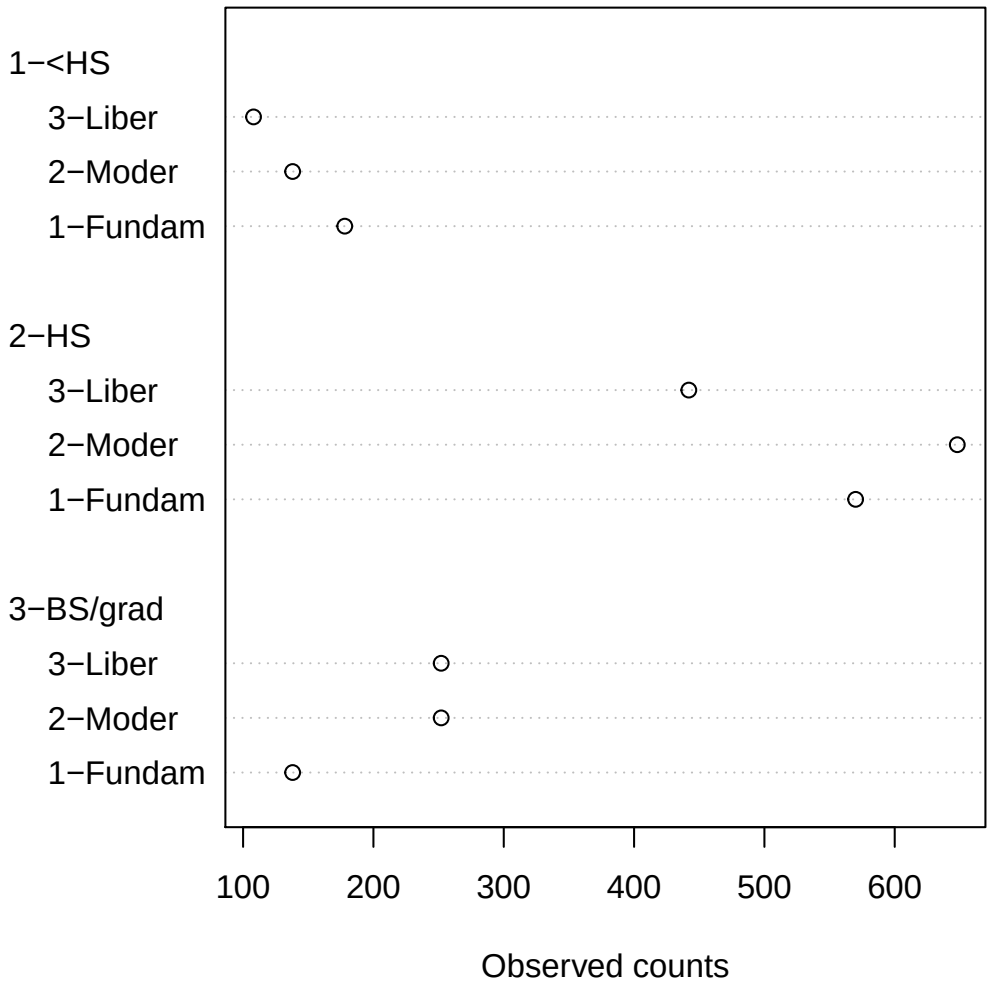
```
            belief
degree       1-Fundam 2-Moder 3-Liber
  1-<HS           178     138      108
  2-HS            570     648      442
  3-BS/grad       138     252      252
```

Export data in latex

```
> library(xtable)
> xtable(ov)

% latex table generated in R 3.2.2 by xtable 1.7-4 package
% Thu Oct 22 14:15:54 2015
\begin{table}[ht]
\centering
\begin{tabular}{rrrr}
  \hline
 & 1-Fundam & 2-Moder & 3-Liber \\
  \hline
1-$<$HS & 178.00 & 138.00 & 108.00 \\
  2-HS & 570.00 & 648.00 & 442.00 \\
  3-BS/grad & 138.00 & 252.00 & 252.00 \\
   \hline
\end{tabular}
\end{table}
```

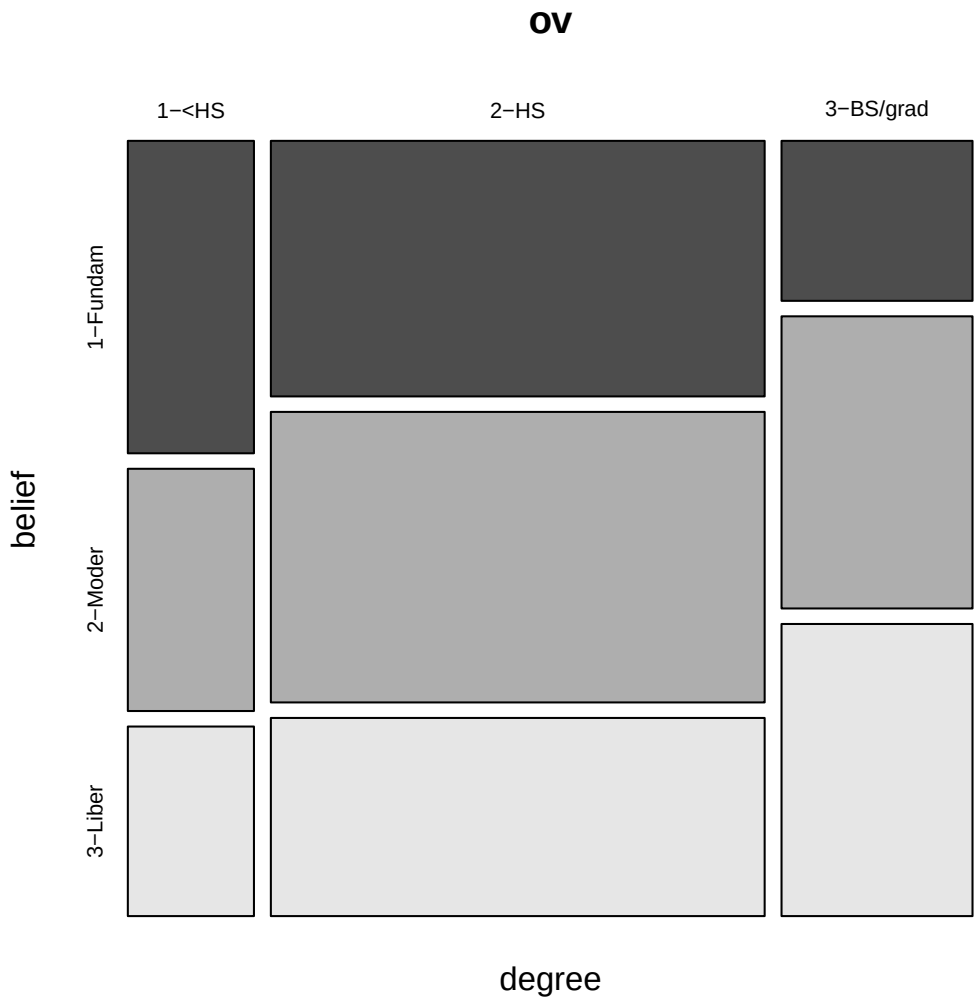
Data visualization

```
> dotchart(t(ov), xlab="Observed counts")
```



Data visualization

```
> mosaicplot(ov, color=TRUE)
```



Compare proportions

```
> prop.test(ov[1:2,1:2])

        2-sample test for equality of proportions with continuity correction

data:  ov[1:2, 1:2]
X-squared = 8.7451, df = 1, p-value = 0.003104
alternative hypothesis: two.sided
95 percent confidence interval:
 0.03187153 0.15875016
sample estimates:
   prop 1    prop 2
0.5632911 0.4679803

> # ---Double-check the proportions---
> 178/(178+138)

[1] 0.5632911

> 570/(570+648)

[1] 0.4679803
```

Pearson χ^2

```
> summary(ov)

Call: xtabs(formula = y ~ degree + belief, data = X)
Number of cases in table: 2726
Number of factors: 2
Test for independence of all factors:
        Chisq = 69.16, df = 4, p-value = 3.42e-14
```

Grouped logistic regression

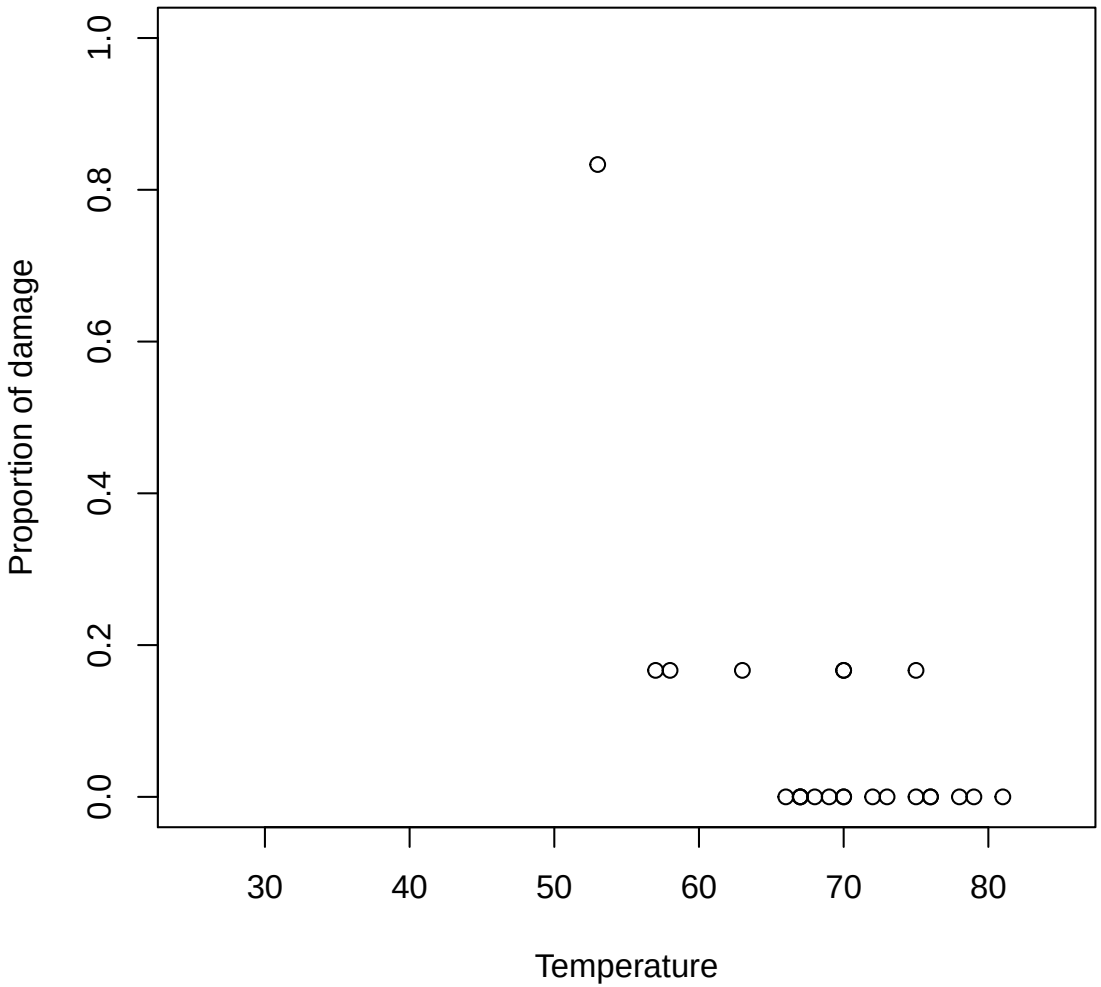
Read the data

```
> library(faraway)
> data(orings)
> ?orings
> head(orings)

  temp damage
1   53      5
2   57      1
3   58      1
4   63      1
5   66      0
6   67      0
```

Explore graphically

Specify 2 responses: 1s and 0s

```
> plot(damage/6 ~ temp, orings, xlim=c(25,85),ylim=c(0,1),
+     xlab="Temperature",ylab="Proportion of damage")
```



Fit simple logistic regression

Specify 2 responses: 1s and 0s

```
> library(MASS)
> fit <- glm(cbind(damage, 6-damage) ~ temp, family=binomial, data=orings)
> summary(fit)

Call:
glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial,
    data = orings)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9529  -0.7345  -0.4393  -0.2079   1.9565

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.66299    3.29626   3.538 0.000403 ***
temp        -0.21623    0.05318  -4.066 4.78e-05 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 16.912  on 21  degrees of freedom
AIC: 33.675

Number of Fisher Scoring iterations: 6
```

Inference

Specify 2 responses: 1s and 0s

```
> # Confidence intervals for parameters
> library(MASS)
> confint(fit)
```

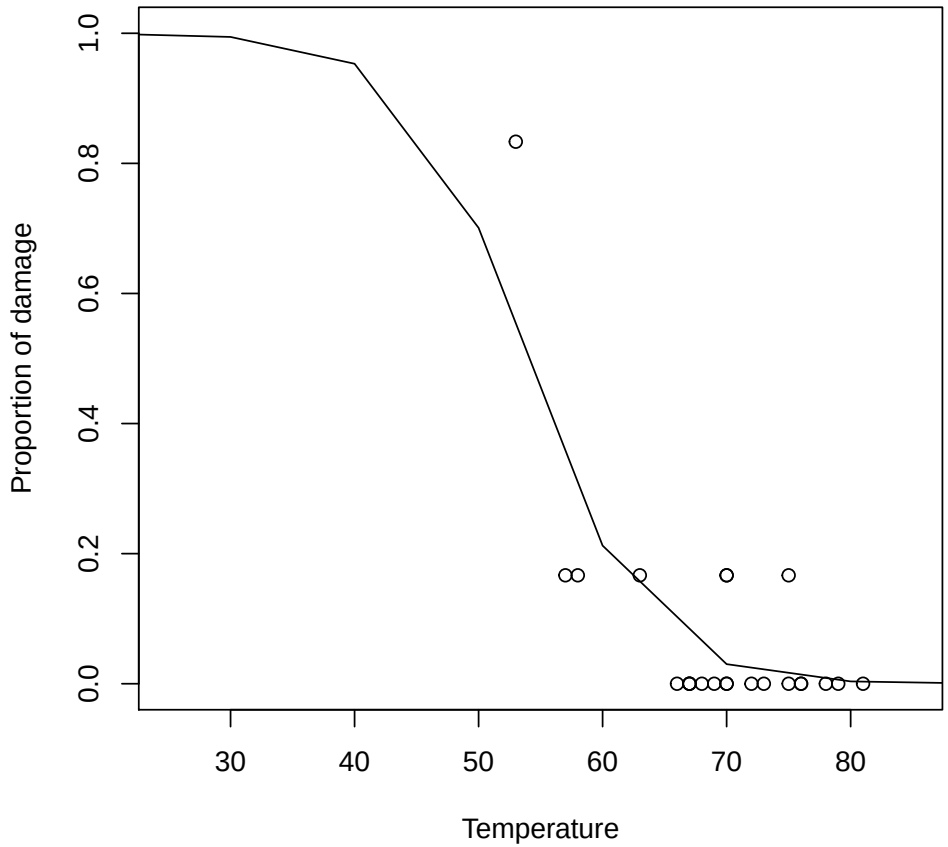
```
                2.5 %    97.5 %
(Intercept)  5.575195 18.737598
temp        -0.332657 -0.120179
```

```
> # Prediction
> newOrings <- data.frame(temp=seq(from=10, to=100, length=10))
> head(predict(fit, newdata=newOrings, se.fit=T, type="response"))
```

```
$fit
           1            2            3            4            5            6
9.999252e-01 9.993503e-01 9.943811e-01 9.531867e-01 7.008411e-01 2.123145e-01
           7            8            9           10
3.007960e-02 3.555480e-03 4.103703e-04 4.723271e-05

$se.fit
           1            2            3            4            5            6
2.070437e-04 1.455768e-03 9.606204e-03 5.374194e-02 1.498385e-01 6.178761e-02
           7            8            9           10
1.670415e-02 3.689817e-03 6.364251e-04 9.788363e-05

$residual.scale
[1] 1
```

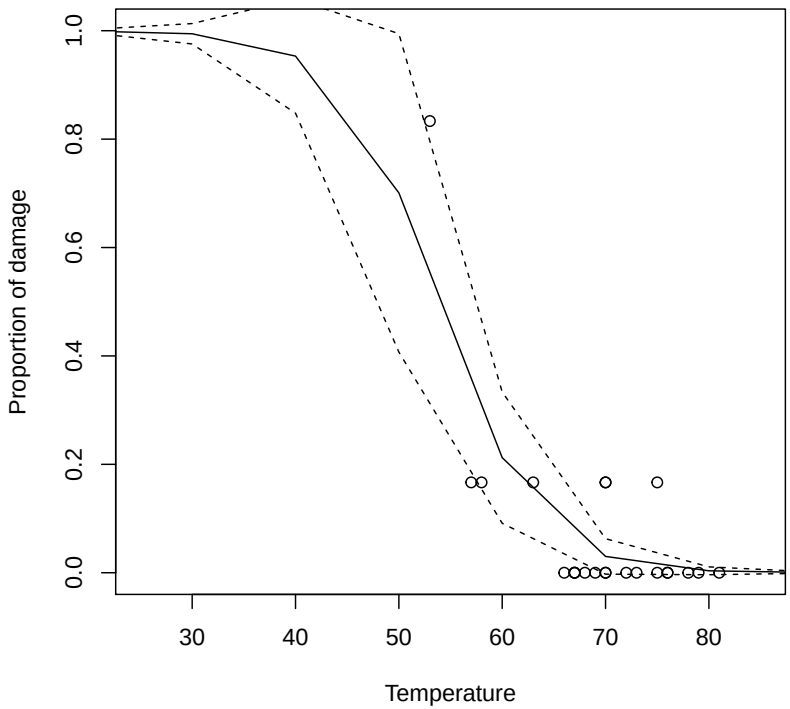
Overlay predicted values

```
> plot(damage/6 ~ temp, orings, xlim=c(25,85),ylim=c(0,1),
+     xlab="Temperature",ylab="Proportion of damage")
> newOrings.predict <- predict(fit, newdata=newOrings, se.fit=T,type="response")
> lines(newOrings$temp, newOrings.predict$fit)
```



Overlay CI for the predicted values

```
> plot(damage/6 ~ temp, orings, xlim=c(25,85),ylim=c(0,1),
+     xlab="Temperature",ylab="Proportion of damage")
> newOrings.predict <- predict(fit, newdata=newOrings, se.fit=T,type="response")
> lines(newOrings$temp, newOrings.predict$fit)
> lines(newOrings$temp,
+       newOrings.predict$fit-qnorm(1-0.05/2)*newOrings.predict$se.fit, lty=2)
> lines(newOrings$temp,
+       newOrings.predict$fit+qnorm(1-0.05/2)*newOrings.predict$se.fit, lty=2)
```



Per-subject logistic regression

Read Individual data

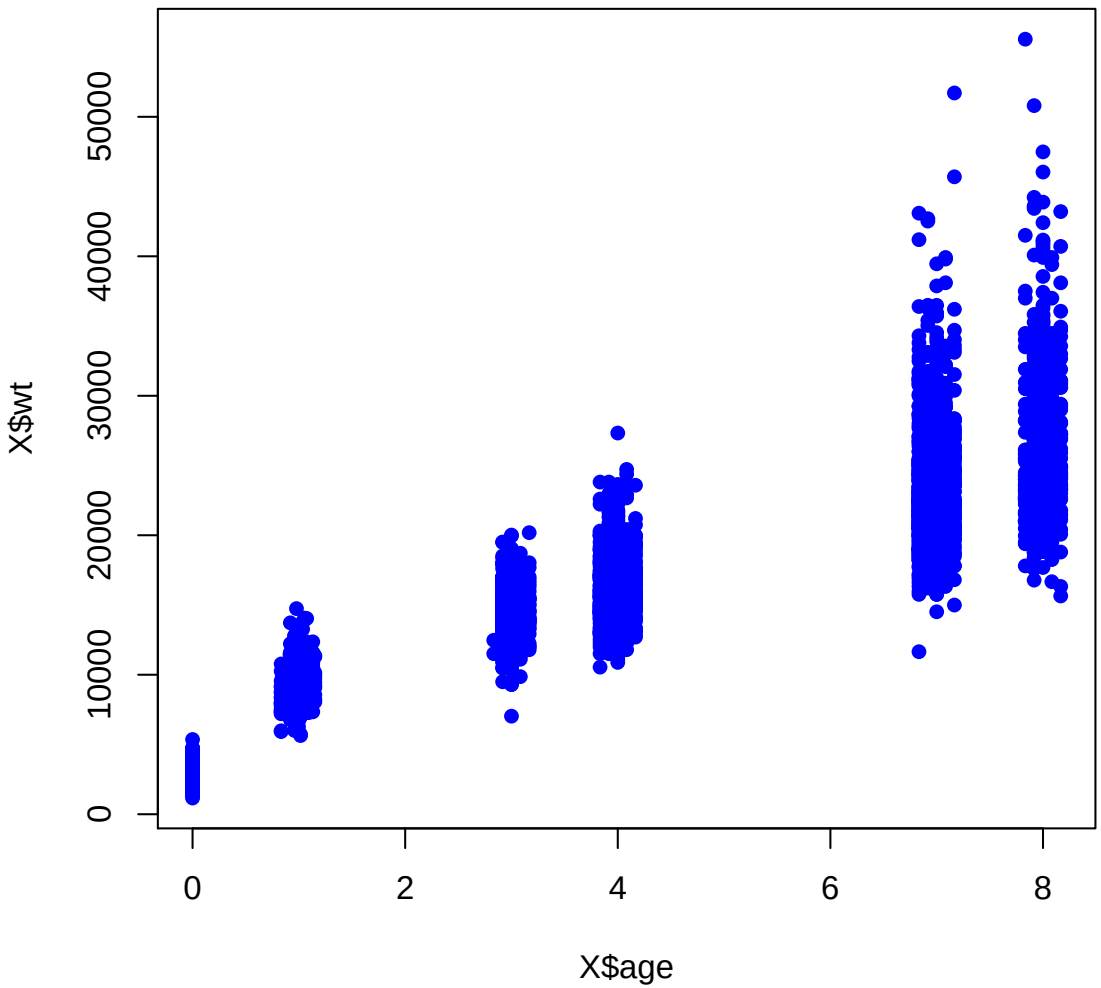
```
> setwd('/Users/ovitek/Dropbox/Olga/Teaching/CS6220/Fall15/LectureNotes/4-logist
> X <- read.table("smokingAndObesity.txt", sep=" ", as.is=TRUE, header=TRUE)
> X <- X[order(X$age),]
> # factor for 'smoking status'
> X$smokeF <- factor(X$smoke)
> head(X)

   personid   wt age smoke over_wt smokeF
1  82109491 3402   0     1    -999      1
3   5115721 2523   0     3    -999      3
6  15123981 3799   0     1    -999      1
11 10110381 2637   0     3    -999      3
17 45115281 3090   0     3    -999      3
23 10110071 3118   0     2    -999      2
```

Format data

```
> # Create a proper binary response for 'overweight'
> table(X$over_wt)

-999    1    2
3298  201 3674

> X$over_wtF <- factor(abs(X$over_wt - 2), levels=c(0,1))
> table(X$over_wtF)

   0    1
3674  201

> head(X)

   personid   wt age smoke over_wt smokeF over_wtF
1  82109491 3402   0     1    -999      1     <NA>
3   5115721 2523   0     3    -999      3     <NA>
6  15123981 3799   0     1    -999      1     <NA>
11 10110381 2637   0     3    -999      3     <NA>
17 45115281 3090   0     3    -999      3     <NA>
23 10110071 3118   0     2    -999      2     <NA>
```

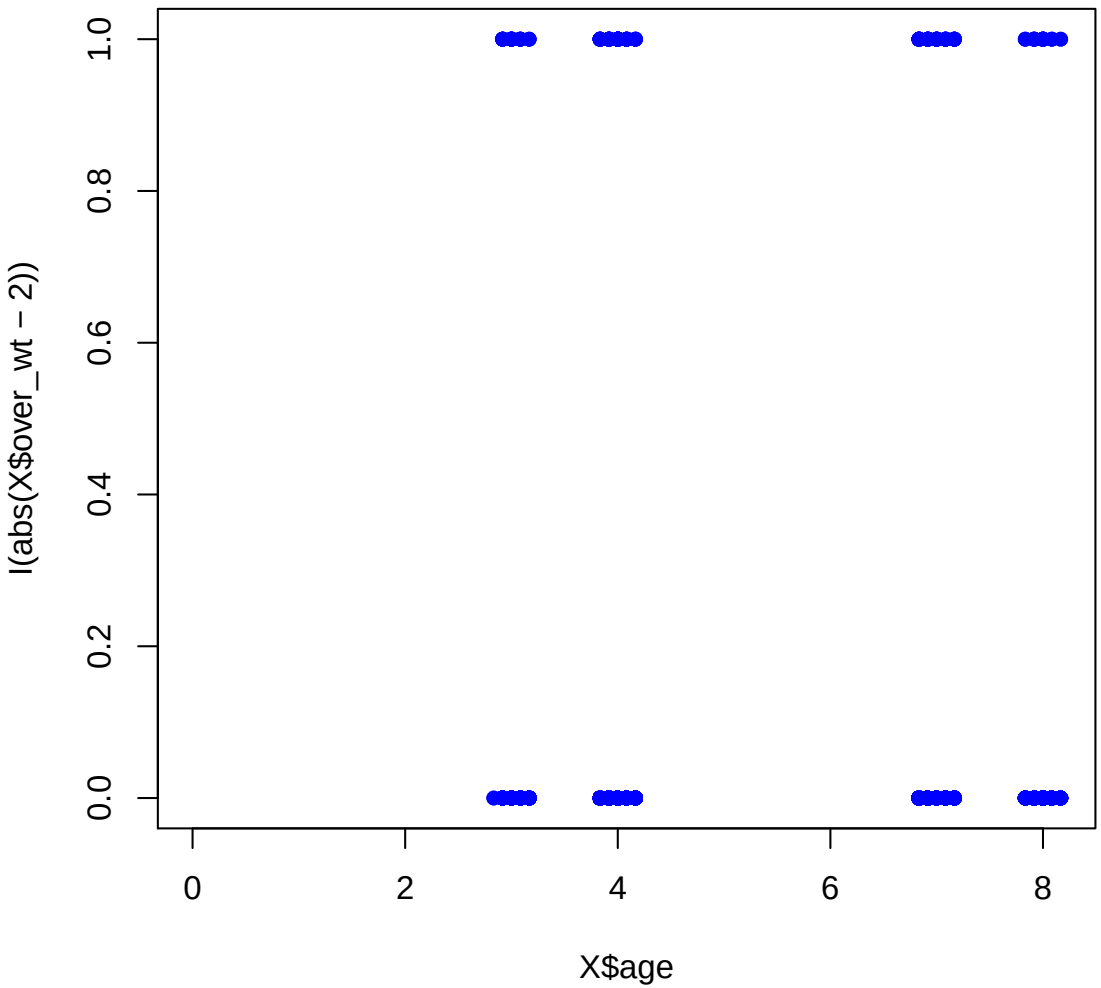
Display continuous response

```
> plot(X$wt~X$age, pch=16, sex=.5, col='blue')
```



Display binary response

```
> plot(I(abs(X$over_wt - 2))~X$age, pch=16, sex=.5, col='blue', ylim=c(0,1))
```

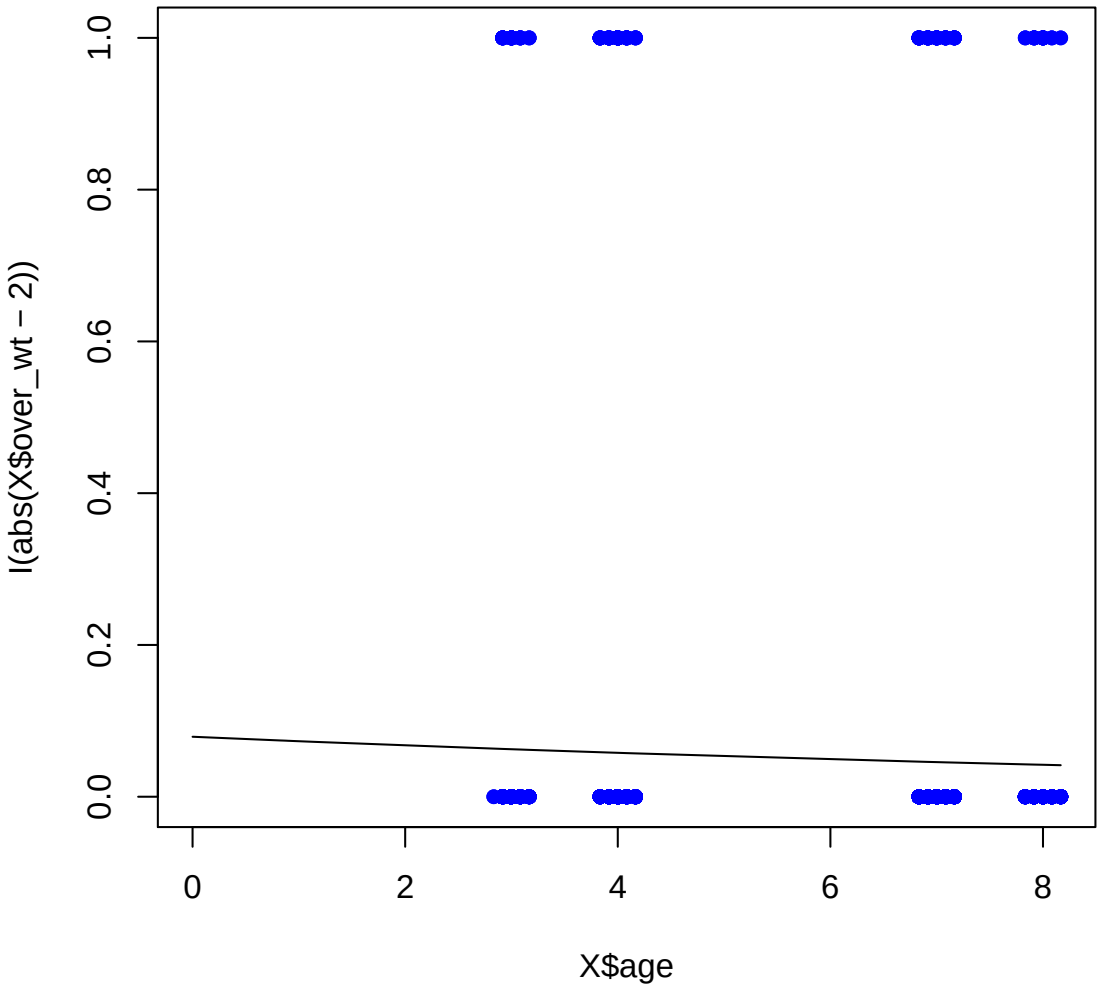


Fit simple logistic regression

```
> fit<- glm(over_wtF ~ age, family=binomial, data=X)
> summary(fit)

Call:
glm(formula = over_wtF ~ age, family = binomial, data = X)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3619  -0.3464  -0.3076  -0.3045   2.5224

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.45573    0.21145 -11.614   <2e-16 ***
age         -0.08366    0.03790  -2.207   0.0273 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1580.9  on 3874  degrees of freedom
Residual deviance: 1576.0  on 3873  degrees of freedom
  (3298 observations deleted due to missingness)
AIC: 1580

Number of Fisher Scoring iterations: 5
```

Display binary response

```
> plot(I(abs(X$over_wt - 2))~X$age, pch=16, sex=.5, col='blue', ylim=c(0,1))
> lines(X$age, predict(fit, newdata=data.frame(age=X$age), type='response'))
```



Fit logistic regression

```
> fit<- glm(over_wtF ~ age + smokeF + age*smokeF, family=binomial, data=X)
> summary(fit)

Call:
glm(formula = over_wtF ~ age + smokeF + age * smokeF, family = binomial,
    data = X)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3780  -0.3470  -0.3240  -0.2924   2.5582

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.43131    0.32271  -7.534 4.92e-14 ***
age         -0.07075    0.05750  -1.230    0.219
smokeF2      0.16174    0.71087   0.228    0.820
smokeF3     -0.08464    0.44866  -0.189    0.850
age:smokeF2 -0.04348    0.12874  -0.338    0.736
age:smokeF3 -0.01712    0.08024  -0.213    0.831
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1580.9  on 3874  degrees of freedom
Residual deviance: 1574.6  on 3869  degrees of freedom
  (3298 observations deleted due to missingness)
```

Prediction

Summaries of classification

Hypothesized class \ True class	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives
Column totals:	P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

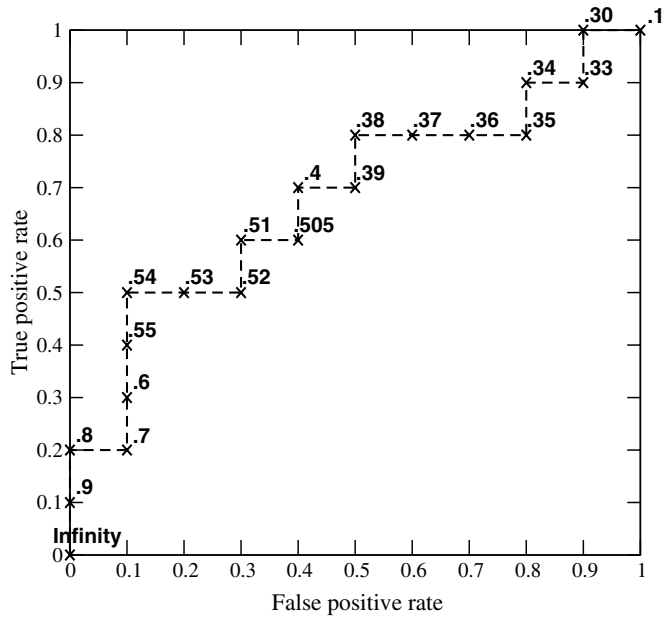
$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

- Results over multiple score cutoffs are summarized in a Receiver Operating Characteristic (ROC) curve
- Vary the cut-off $c \in (0, 1)$, and choose c to optimize sensitivity and specificity.
- Vary c, and for all c plot sensitivity vs 1-specificity.
  Evaluate models by area under the curve.

Fawcett, "An introduction to ROC analysis". *Pattern Recognition Letters*, 2005

ROC curve

Inst#	Class	Score	Inst#	Class	Score
1	**p**	.9	11	**p**	.4
2	**p**	.8	12	**n**	.39
3	**n**	.7	13	**p**	.38
4	**p**	.6	14	**n**	.37
5	**p**	.55	15	**n**	.36
6	**p**	.54	16	**n**	.35
7	**n**	.53	17	**p**	.34
8	**n**	.52	18	**n**	.33
9	**p**	.51	19	**p**	.30
10	**n**	.505	20	**n**	.1



Fawcett, "An introduction to ROC analysis". *Pattern Recognition Letters*, 2005

Example

```
> library(faraway)
> data(pima)
> ?pima
> head(pima)
```

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0

Fit full model on the training set

```
> library(ROCR)
> # as example, here use 1/4 of the data to build the model
> train <- sample(x=1:nrow(pima), size=nrow(pima)/4)
> # fit the full model on the training dataset
> fit.train <- glm(test ~., family=binomial, data=pima[train,])
> summary(fit.train)

Call:
glm(formula = test ~ ., family = binomial, data = pima[train,
    ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1120  -0.6944  -0.4096   0.7077   2.3549

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.059396   1.185801  -5.953 2.63e-09 ***
pregnant     0.128331   0.066353   1.934   0.0531 .
glucose      0.042004   0.008042   5.223 1.76e-07 ***
diastolic   -0.015280   0.009087  -1.682   0.0927 .
triceps      0.020472   0.014155   1.446   0.1481
insulin     -0.003293   0.001988  -1.656   0.0977 .
bmi          0.052463   0.024665   2.127   0.0334 *
diabetes     0.313715   0.594159   0.528   0.5975
age         -0.009847   0.021957  -0.448   0.6538
---
```

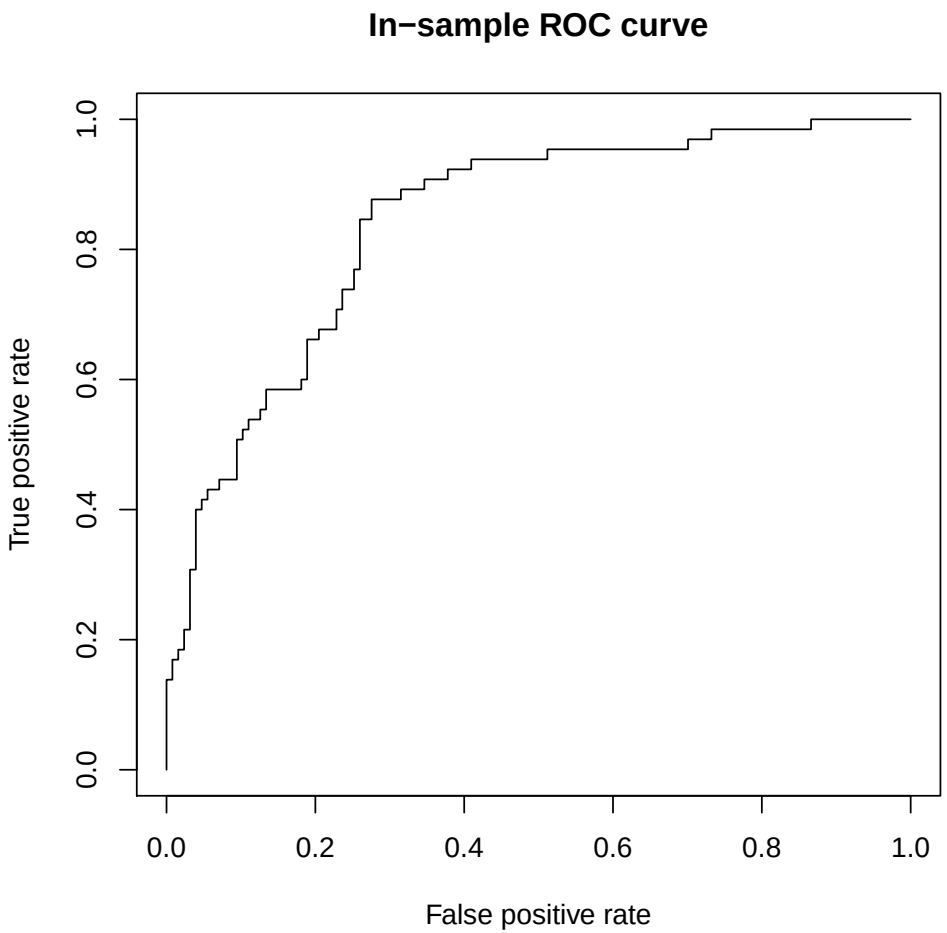
Predicted probabilities on the same training set

```
> scores <- predict(fit.train, newdata=pima[train,], type="response")
> # compare predicted probabilities to labels, for varying probability cutoffs
> pred <- prediction(scores, labels=pima[train,]$test )
> perfTrain <- performance(pred, "tpr", "fpr")
```

ROC curve

```
> # plot the ROC curve
> plot(perfTrain, colorize=F, main="In-sample ROC curve")
> # print out the area under the curve
> unlist(attributes(performance(pred, "auc"))$y.values)
```

```
[1] 0.8408237
```

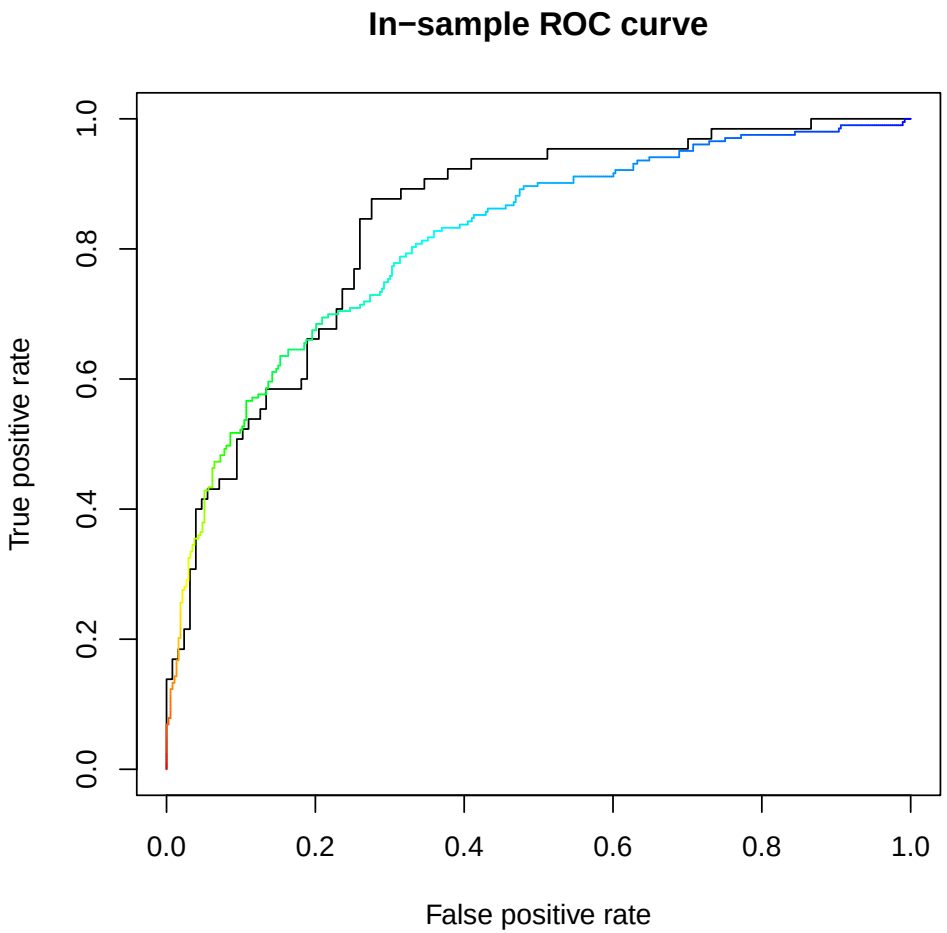


Evaluate on the validation set

```
> scores <- predict(fit.train, newdata=pima[-train,], type="response")
> pred <- prediction( scores, labels=pima[-train,]$test )
> perfValid <- performance(pred, "tpr", "fpr")
```

ROC curve

```
> # overlay the line for the ROC curve
> plot(perfTrain, colorize=F, main="In-sample ROC curve")
> plot(perfValid, colorize=T, add=TRUE)
> # print out the area under the curve
> unlist(attributes(performance(pred, "auc"))$y.values)
[1] 0.8145908
```



Visualizing prediction

Simulate data

Example from http://www.r-bloggers.com/choosing-a-classifier/

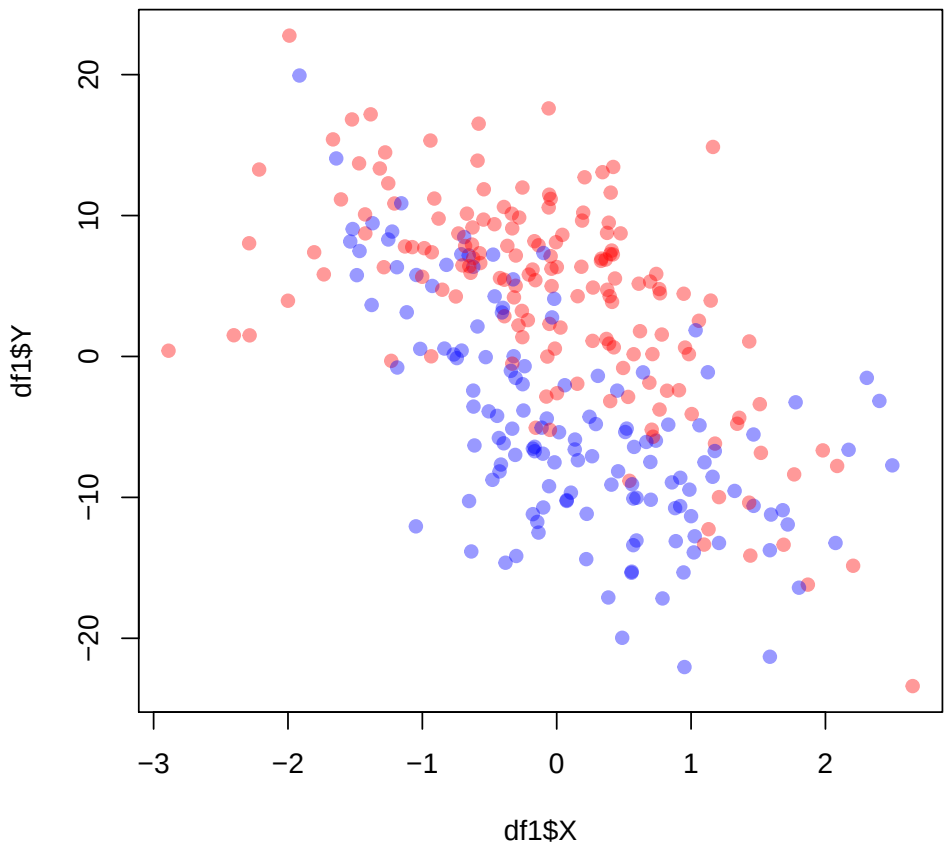
```
> n = 500
> set.seed(1)
> X = rnorm(n)
> ma = 10-(X+1.5)^2*2
> mb = -10+(X-1.5)^2*2
> M = cbind(ma,mb)
> set.seed(1)
> Z = sample(1:2,size=n,replace=TRUE)
> # define value of Y according to the class of Z, and add noise
> Y = ma*(Z==1)+mb*(Z==2)+rnorm(n)*5
> df = data.frame(Z=as.factor(Z),X,Y)
```

Split into training and validation set

```
> df1 = training = df[1:300,]
> df2 = testing  = df[301:500,]
```

Visualize training set

```
> plot(df1$X,df1$Y,pch=19,col=c(rgb(1,0,0,.4),
+    rgb(0,0,1,.4))[df1$Z])
```

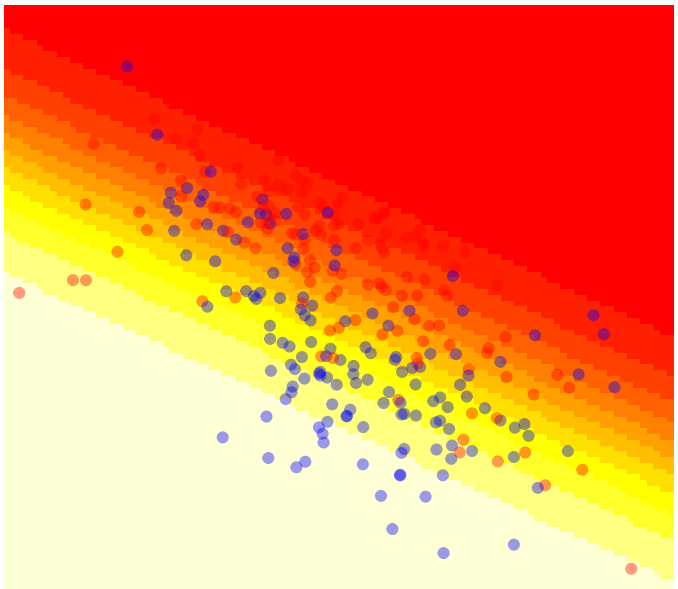


Fit logistic regression

```
> fit=glm(Z~X+Y,data=df1,family=binomial)
>  pred=function(x,y)
+   predict(fit,newdata=data.frame(X=x,Y=y),
+   type="response")
> summary(fit)

Call:
glm(formula = Z ~ X + Y, family = binomial, data = df1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3373  -0.7906  -0.3616   0.7792   2.3781

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.13626    0.14311  -0.952    0.341
X           -1.00156    0.20281  -4.938 7.88e-07 ***
Y           -0.22813    0.02706  -8.431  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂ

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 414.55  on 299  degrees of freedom
Residual deviance: 297.17  on 297  degrees of freedom
AIC: 303.17
```

Visualize prediction

```
> vx=seq(-3,3,length=101)
> vy=seq(-25,25,length=101)
> z=matrix(NA,length(vx),length(vy))
> for(i in 1:length(vx)){
+    for(j in 1:length(vy))
+    {z[i,j]=pred(vx[i],vy[j])}
+ }
> image(vx,vy,z,axes=FALSE,xlab="",ylab="")
> points(df1$X,df1$Y,pch=19,col=c(rgb(1,0,0,.4),
+ rgb(0,0,1,.4))[df1$Z])
```

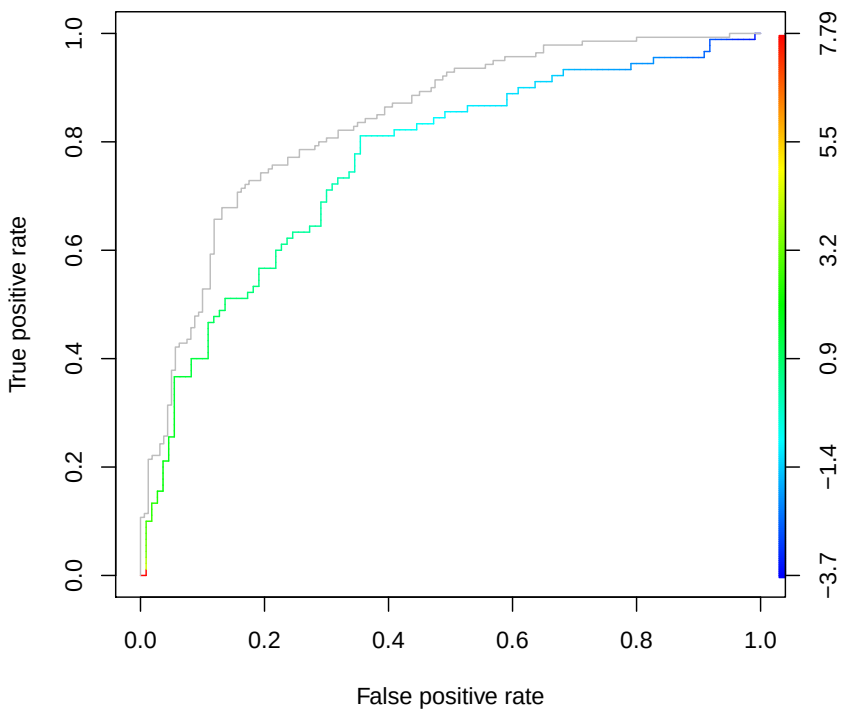


Evaluate the predictive ability

```
> Y1=as.numeric(df1$Z)-1
> Y2=as.numeric(df2$Z)-1
> library(ROCR)
> S1 = predict(fit,newdata=df1)
> S2 = predict(fit,newdata=df2)
> pred <- prediction( S2, Y2 )
> perfValid <- performance( pred, "tpr", "fpr" )
```

Evaluate the predictive ability

```
> pred <- prediction( S1, Y1 )
> perfTrain <- performance( pred, "tpr", "fpr" )
> plot( perfValid, colorize=TRUE )
> plot( perfTrain ,add=TRUE,col="grey")
```



Variable selection

Automatic Variable Selection

- Exhaustive search. Minimize:

$$\begin{aligned} & -2\log_e L(\mathbf{b}) \\ AIC_p &= -2\log_e L(\mathbf{b}) + 2p \\ BIC_p &= -2\log_e L(\mathbf{b}) + p\log_e(n) \end{aligned}$$

- Heuristic search
  - forward selection; backward elimination; stepwise selection
  - based on Wald statistic and Normal distribution

Stepwise variable selection based on AIC

```
> # 'k' distinguishes AIC and BIC
> fit <- glm(test ~., family=binomial, data=pima)
> step.aic <- step(fit, k=2, trace=F)
> step.aic$anova
```

	Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1		NA	NA	759	723.4454	741.4454
2	- triceps	1	0.008051802	760	723.4534	739.4534

Stepwise variable selection based on BIC

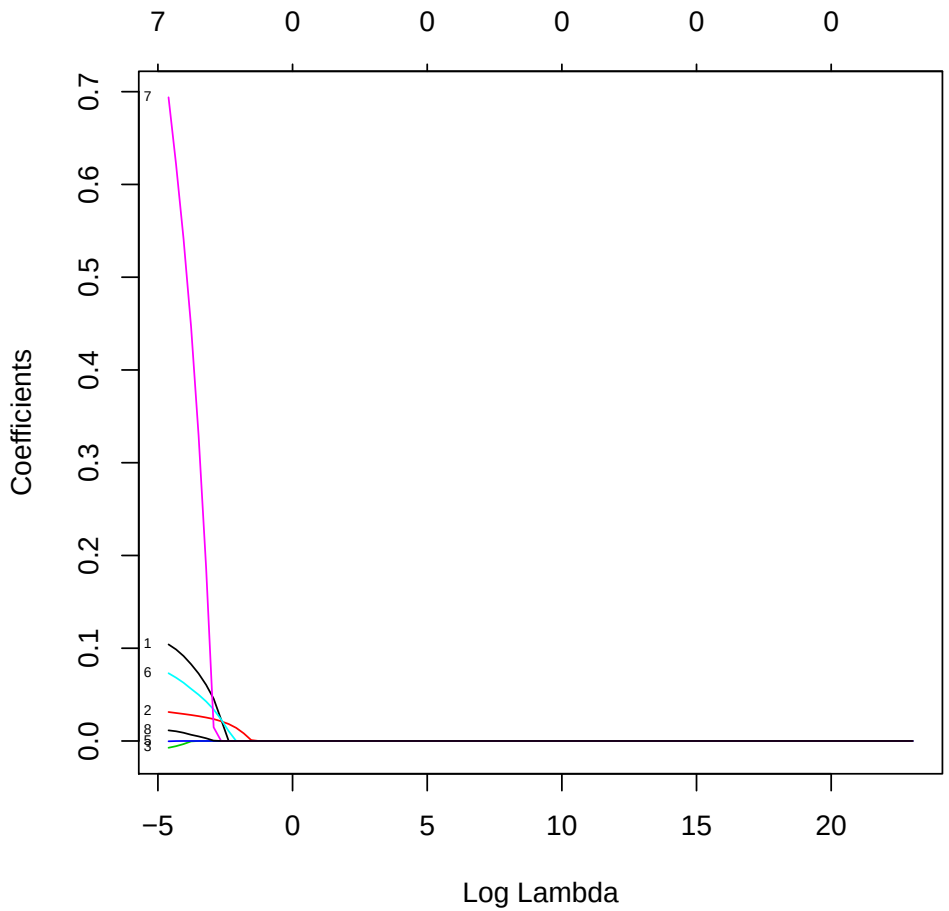
```
> # 'k' distinguishes AIC and BIC
> step.bic <- step(fit, k=log(nrow(pima)), trace=F)
> step.bic$anova
          Step Df    Deviance Resid. Df Resid. Dev      AIC
1             NA          NA       759   723.4454 783.2395
2   - triceps  1 0.008051802       760   723.4534 776.6037
3   - insulin  1 2.008267852       761   725.4617 771.9682
4       - age  1 3.097908342       762   728.5596 768.4223
5 - diastolic  1 5.746278627       763   734.3059 767.5248
```

Variable selection based on lasso

```
> library(glmnet)
> lasso.mod  <- glmnet(x=as.matrix(pima[,-9]), y=pima[,9],
+   family='binomial', alpha=1, lambda=10^seq(10,-2,length=100))
> names(lasso.mod)
 [1] "a0"         "beta"       "df"         "dim"        "lambda"
 [6] "dev.ratio"  "nulldev"    "npasses"    "jerr"       "offset"
[11] "classnames" "call"       "nobs"

> lasso.mod$lambda[40]

[1] 187381.7

> coef(lasso.mod)[,40]

(Intercept)    pregnant     glucose   diastolic     triceps     insulin
 -0.6236211   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
        bmi    diabetes         age
  0.0000000   0.0000000   0.0000000

> lasso.mod$lambda[95]

[1] 0.04037017

> coef(lasso.mod)[,95]

(Intercept)    pregnant     glucose   diastolic     triceps     insulin
-5.58534068  0.06052017  0.02531610  0.00000000  0.00000000  0.00000000
        bmi    diabetes         age
 0.04261398  0.18673245  0.00291996
```

Variable selection based on lasso

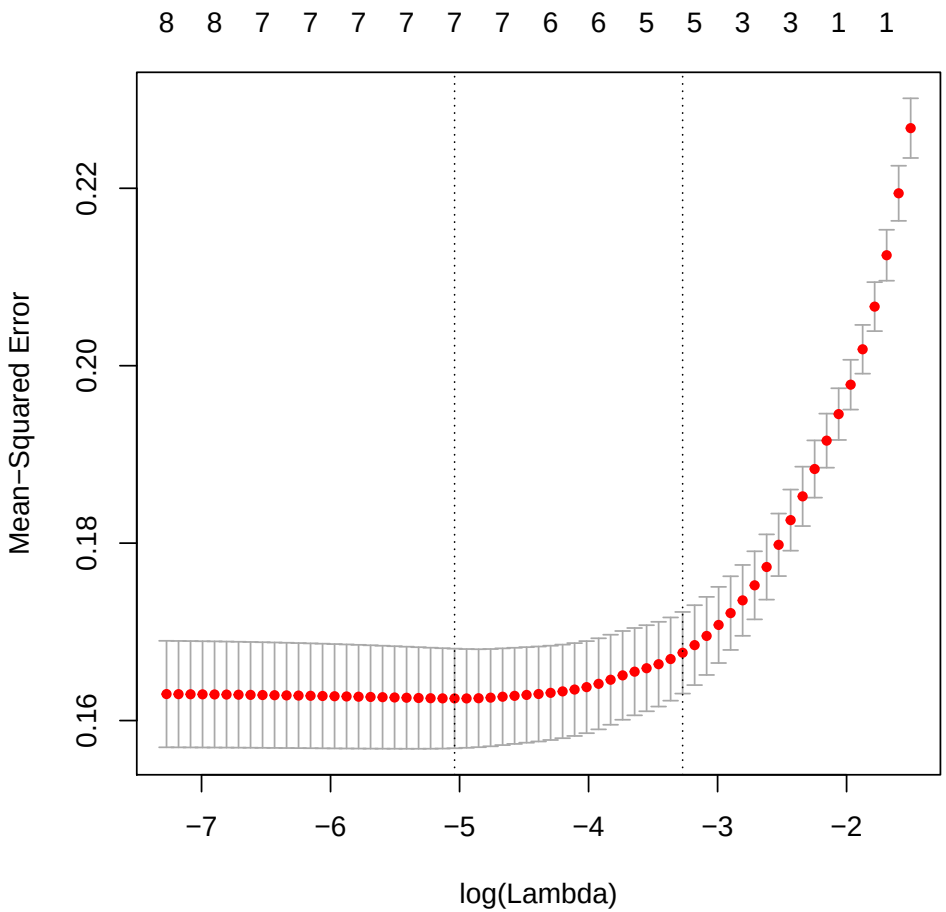
```
> plot(lasso.mod, label=TRUE, xvar='lambda')
```



Variable selection based on lasso

```
> cv.out <- cv.glmnet(x=as.matrix(pima[,-c(9)]), y=pima[,9], alpha=1)
> plot(cv.out)
> bestlam <- cv.out$lambda.min
> bestlam
```

```
[1] 0.006482836
```

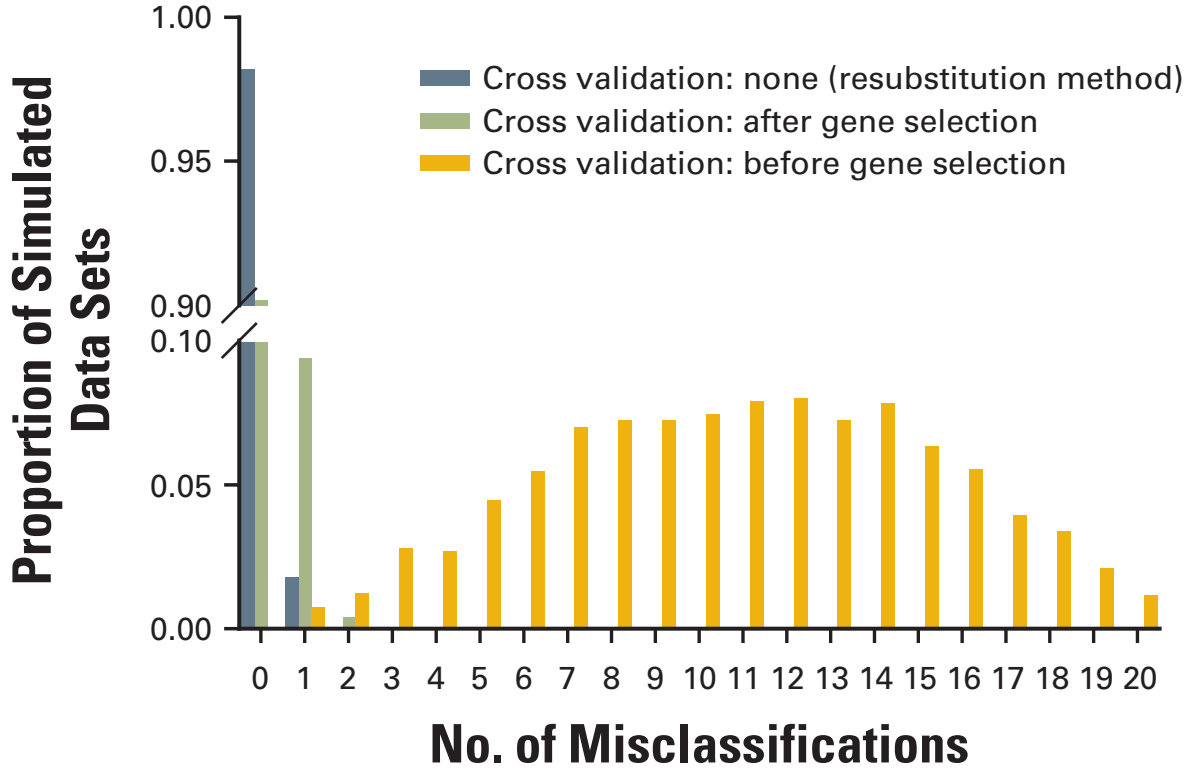


Variable Selection Should be Done as Part of Cross-Validation

- Example from Simon *et al.*, JNCI, 2003.
- Simulated data with no structure
  - 20 observations with random labels
  - 6,000 possible but unrelated predictors
  - Repeated 200 times
- Estimated predictive accuracy using
  - no cross-validation
  - selecting features on full dataset,
    then using cross-validation
  - selecting features at each step of cross-validation

Variable Selection Should be Done as Part of Cross-Validation

Example from Simon *et al.*, JNCI, 2003.



- Conclusion
  - Incorporating selection of predictors within the cross-validation procedure is key