STA303/1002 - Methods of Data Analysis II

(Week 09 lecture note)

Wei (Becky) Lin

Mar 07/09, 2017

Notes

- Assignment 3 (last assignment) will be available this week.
  - Due: April 5. (Roughly, you will have 3 weeks to try)
- Midterm
  - Result will be available this week before weekend.
- Topics for the last 4 lectures
  - Logit regression and Poisson regression.
  - Linear mixed effect model.

Review on logistic regression with binary response

- A binary dependent variable. Let Y be the response for the i-th observation where

$$Y_i|x_i = \begin{cases} 1, & \text{with prob}=\pi(x_i) \\ 0, & \text{with prob}=1-\pi(x_i) \end{cases}$$

- The probability density of Y_i

$$f(Y_i) = \pi_i^{Y_i}(1-\pi_i)^{1-Y_i} = \begin{cases} \pi_i, & Y_i=1 \\ 1-\pi_i, & Y_i=0 \end{cases}$$

  - $E(Y_i) = \pi_k, V(Y_i) = \pi_i(1-\pi_i)$

- To keep the linear type predictor, we need a link function g$(\cdot)$ to transform the mean to a linear function, let $g = logit$

$$g(\pi) = g(\mu) = logit(\pi) = \log\frac{\pi}{1-\pi} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

This leads

$$\frac{\pi}{1-\pi} = \exp\{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\} = e^{X\beta}$$

$$\pi = E(Y) = P(Y=1) = \frac{\exp\{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\}}{1+\exp\{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\}} = \frac{e^{X\beta}}{1+e^{X\beta}}$$

Review on logistic regression with binary response

- β estimation:
  - Least square estimates are inappropriate
  - use maximum likelihood estimate: IRLS algorithm, in Week 7, slides 32-35.
  - fitted value for the individual i is a probability in the interval of (0,1)

$$\hat{\pi}_i = \hat{E}(Y_i) = \frac{\exp\{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p\}}{1 + \exp\{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p\}} = \frac{e^{x\hat{\beta}}}{1 + e^{x\hat{\beta}}}$$

- Coefficient interpretation in logistic regression

$$g(\mu) = logit(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - $\beta_0 = \log\left(\frac{\pi}{1-\pi}\right)$ is a logit function of the marginal probability that $Y = 1|X = 0$.
  - β_k: Similar to LR model, holding all the other predictors fixed, changing X_k by one unit leads to a change in log odds, η, of β_k. Or, holding all the other predictors fixed, changing X_k by one unit leads to a change in the odds ratio of β_k.

(*) For categorical variable, $OR_i = \frac{odds_i}{odds_0} = e^{\beta_i}$, reference level

$$ln(odds(X_k+1)) - ln(odds(X_k)) = \log\left(\frac{odds(X_k+1)}{odds(X_k)}\right) = \beta_k \rightarrow \frac{\omega(X_k+1)}{\omega(X_k)} = e^{\beta_k}$$

  - The sign of β_k indicates whether η or ω increases ($\beta_k > 0$) or decreases ($\beta_k < 0$) as X_k increases.

Equivalent specification: Binomial distribution

- Change in notation
  - Data: (Y_{ij}, n_i, X_i), i=1,2,..., c
  - X_i: predictor for observation i
  - n_i: # of Bernoulli trials in observation i
  - $Y_{i\cdot} = \sum_{j=1}^{n_i} Y_{ij}$, $Y_{ij} \sim Bern(\pi_i)$
  - Model

$$Y_{i\cdot} \sim_{iid} Bin(n_i, \pi_i), \ \pi_i = \frac{\exp\{\beta_0 + \beta_1 X_i\}}{1 + \exp\{\beta_0 + \beta_1 X_i\}}$$

- Log-likelihood for the Binomial model

$$\ell(\beta) = \log_e \prod_{i=1}^{c} \left\{ \binom{n_i}{Y_{i\cdot}} \pi_i^{Y_{i\cdot}} (1-\pi_i)^{n_i - Y_{i\cdot}} \right\} \quad (1)$$

$$= \sum_{i=1}^{c} \left\{ Y_{i\cdot} \log(\pi_i) + (n_i - Y_{i\cdot}) \log(1-\pi_i) + \log \binom{n_i}{Y_{i\cdot}} \right\} \quad (2)$$

$$= \sum_{i=1}^{c} \left\{ Y_{i\cdot} \log\left(\frac{\pi_i}{1-\pi_i}\right) + n_i \log(1-\pi_i) + \log \binom{n_i}{Y_{i\cdot}} \right\} \quad (3)$$

Equivalent specification: Binomial distribution

- Show that the Binomial log-likelihood equals Bernoulli log-likelihood up to a constant

$Y_{i\cdot} = \sum_{j=1}^{n_i} Y_{ij}$, $Y_{ij} \sim iid\ Bern(\pi_i)$

$$\ell(\beta)^{Bino} = \sum_{i=1}^{c} \left\{ Y_{i\cdot} \log\left(\frac{\pi_i}{1-\pi_i}\right) + n_i \log(1-\pi_i) + \text{constant} \right\} \tag{4}$$

$$= \sum_{i=1}^{c} \left\{ \sum_{j=1}^{n_i} Y_{ij} \log\left(\frac{\pi_i}{1-\pi_i}\right) + \sum_{j=1}^{n_i} \log(1-\pi_i) + \text{constant} \right\} \tag{5}$$

$$= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left\{ Y_{ij} \log\left(\frac{\pi_i}{1-\pi_i}\right) + \log(1-\pi_i) + \text{constant} \right\} \tag{6}$$

$$= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left\{ \ell(\beta)^{Bern} + \text{constant} \right\} \tag{7}$$

- Both models lead to same parameter estimates and inferences, but have different deviances.

Deviance

- In standard linear models, we estimate unknown parameter (β) by minimizing the sum of the squared residuals $\sum_i e_i^2$. Equivalent to finding the parameters that maximize the likelihood.
- In a GLM, we also fit parameters by maximizing the likelihood. The deviance is negative two times the maximum log likelihood up to an additive constant.

$$D = -2[\ell(\hat{\beta}) - \ell_S(\tilde{\beta})]$$

maximize $\ell(\hat{\beta})$ is the same to minimize $-2\ell(\hat{\beta})$ since $\ell_S(\tilde{\beta})$ is viewed as constant given data.

- Estimation is equivalent to finding parameter value that minimize the deviance.

Deviance for Bernoulli model

$$Y_i \sim Bern(\pi_i)$$

$$D = -2\{\ell(\hat{\beta}) - \ell_S(\tilde{\beta})\}$$

$$= -2\ell(\hat{\beta}) = -2\sum_{i=1}^{n}[Y_i \log(\hat{\pi}_i) + (1 - Y_i)\log(1 - \hat{\pi}_i)]$$

$$= 2\sum_{i=1}^{n}[Y_i \log\frac{Y_i}{\hat{\pi}_i} + (1 - Y_i)\log\frac{1 - Y_i}{1 - \hat{\pi}_i}]$$

Proof:

Saturated model. # of Y_i = # of β_i $\Rightarrow$ $Y_i = \hat{\pi}_i$ ($x\hat{\beta}$)

$$\ell_S(\tilde{\beta}) = \sum_{i=1}^{n}[Y_i \log(\tilde{\pi}_i) + (1 - Y_i)\log(1 - \tilde{\pi}_i)], \hat{\pi}_i = Y_i$$

$$= \sum_{i=1}^{n}[\underbrace{Y_i \log(Y_i)}_{A_1} + \underbrace{(1 - Y_i)\log(1 - Y_i)}_{A_2}]$$

$= 0$ since Y_i is either 0 or 1

$Y_i=1$, $A_1=0$, $A_2=0$

$Y_i=0$ $A_1=0$, $A_2=0$

Deviance for Binomial model

$$Y_i \sim Bino(n_i, \pi_i), P(Y_i = y_i) = \binom{n_i}{y_i} \pi_i^{y_i} (1-\pi_i)^{n_i - y_i}$$

(*)
$$D = 2 \sum_{i=1}^{c} \left\{ y_i \log \frac{y_i}{\hat{y}_i} + (n_i - y_i) \log \frac{n_i - y_i}{n_i - \hat{y}_i} \right\}$$

proof:

Current M ↙ $\hat{\pi}_i(X\hat{\beta})$, $\hat{p}_i(X\tilde{\beta})$ ↙ saturated M

$$\begin{aligned}
D &= -2[\ell(\hat{\beta}) - \ell_S(\tilde{\beta})] \\
&= -2 \sum_{i=1}^{c} \left[y_i \log(\hat{\pi}_i) + (n_i - y_i) \log(1 - \hat{\pi}_i) \right] \\
&\quad + 2 \sum_{i=1}^{c} \left[y_i \log(\hat{p}_i) + (n_i - y_i) \log(1 - \hat{p}_i) \right] \quad \hat{p}_i = y_i / n_i \\
&= 2 \sum_{i=1}^{c} \left\{ y_i \log \frac{\hat{p}_i n_i}{\hat{\pi}_i n_i} + (n_i - y_i) \log \frac{(1 - \hat{p}_i)}{(1 - \hat{\pi}_i)} \right\}^{n_i} \\
&= 2 \sum_{i=1}^{c} \left\{ y_i \log \frac{y_i}{\hat{y}_i} + (n_i - y_i) \log \frac{n_i - y_i}{n_i - \hat{y}_i} \right\}, \quad \hat{y}_i = n_i \hat{\pi}_i
\end{aligned}$$

Null and Residual deviance in Summary output

From Summary Output:

null deviance: $\text{logit}(\pi) = \beta_0$

residual deviance: $\text{logit}(\pi) = \beta_0 + \beta_1 x + \ldots \beta_p x_p$

- The null deviance shows how well the response variable is predicted by a model that includes only the intercept (grand mean) where as residual with inclusion of independent variables.
- Residual deviance shows how well the response variable is predicted by a model that includes p-covariates in the model.

Smoking data set

```
library(SMPracticals)
data(smoking)
str(smoking)
```

```
## 'data.frame':    14 obs. of  4 variables:
##  $ age   : Factor w/ 7 levels "18-24","25-34",..: 1 1 2 2 3 3 4 4 5 5 ...
##  $ smoker: num  1 0 1 0 1 0 1 0 1 0 ...
##  $ alive : num  53 61 121 152 95 114 103 66 64 81 ...
##  $ dead  : num  2 1 3 5 14 7 27 12 51 40 ...
```

```
dim(smoking)
```

c=n=# of Y_i = 14

```
## [1] 14  4
```

```
head(smoking,6)
```

X_1 X_2 $n-Y$ Y

```
##     age smoker alive dead
## 1 18-24      1    53    2
## 2 18-24      0    61    1
## 3 25-34      1   121    3
## 4 25-34      0   152    5
## 5 35-44      1    95   14
## 6 35-44      0   114    7
```

Smoking data set

cbind(Y, n-Y)

```
mod = glm(cbind(dead,alive)~factor(smoker)+age,data=smoking, family=binomial)
summary(mod)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ factor(smoker) + age, family = binomial,
##     data = smoking)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72545  -0.22836   0.00005   0.19146   0.68162
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -3.8601     0.5939  -6.500 8.05e-11 ***
## factor(smoker)1    0.4274     0.1770   2.414 0.015762 *
## age25-34           0.1201     0.6865   0.175 0.861178
## age35-44           1.3411     0.6286   2.134 0.032874 *
## age45-54           2.1134     0.6121   3.453 0.000555 ***
## age55-64           3.1808     0.6006   5.296 1.18e-07 ***
## age65-74           5.0880     0.6195   8.213  < 2e-16 ***
## age75+            27.8073 11293.1430   0.002 0.998035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 641.4963  on 13  degrees of freedom
## Residual deviance:   2.3809  on  6  degrees of freedom
## AIC: 65.377
```

β_0 β_1 β_2 β_3 β_4 β_5 β_6 β_7

n-1 = 14-1 = 13

$Y \sim \beta_0$

$Y \sim X\beta$

C-D = 14-8 = 6 = 14 - # of β

Inference about individual β_k: Wald test

- From large sample theory, we have

$$Z = \frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \sim_{approx} N(0,1)$$

- An approximate variance for $S^2(\hat{\beta})$

$$S^2(\hat{\beta}) = \left(- \frac{\partial^2 \ell(\beta)}{\partial \beta_i \partial \beta_j} \Big|_{\hat{\beta}} \right)^{-1}$$

p×p var-covariance matrix. Diagonal elements = $\widehat{var}(\beta_j)$

- An approximate confidence interval for β_j

$$\hat{\beta}_j \pm Z_{\alpha/2}\, se(\hat{\beta}_j)$$

  - You can also exponentiate the coefficients (get OR) and the corresponding CI for the OR.

- Two sided test $H_0 : \beta_j = 0$ versus $H_a : \beta_j \neq 0$: reject H_0 if

$$\left| \frac{\hat{\beta}_j - 0}{se(\hat{\beta}_j)} \right| > Z_{\alpha/2}$$

Simultaneous Inference about several $\beta_k = 0$: Likelihood ratio Test

(or called: χ^2 test for nested models)

- Multivariate logistic regression

$$\log\left(\frac{E(Y_i)}{1-E(Y_i)}\right) = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_p X_{ip}$$

- Test:

$$H_0 : \beta_1 = \ldots = \beta_q = 0; \quad H_a : \text{not all } \beta_1, \ldots, \beta_q = 0$$

- Test statistic

$G^2 = -2(\ell_R - \ell_F) = D_R - D_F$

LRT $=$ $G^2 = -2\,(\log \mathrm{L}(\text{reduced model}) - \log \mathrm{L}(\text{full model}))$

$= -2(\ell_R - \ell_F)$

$= -2(\log \mathrm{L}(\text{reduced model}) - \log \mathrm{L}(\text{saturated model}))$ $\rightarrow D_R$

$\quad -\{\log \mathrm{L}(\text{full model}) - \log \mathrm{L}(\text{saturated model})\})$ $\rightarrow D_F$

$= \text{Deviance}_R - \text{Deviance}_F$

- Approximate distribution of G^2 for large n
  - Reject H_0 if $G^2 > \chi^2_{1-\alpha,q}$ $\leftarrow$ $q = df_R - df_F = (n-(p-q)) - (n-p)$
- R function: anova(mod_R, mod_F)

Inference about several $\beta_k = 0$ by constraint

- Test of constraint $H_0 : C_{j\times p}\beta_{p\times 1} = h_{j\times 1}$
  - reject if

$$(C\hat{\beta} - h)'\{C'\,avar(\hat{\beta})C\}^{-1}(C\hat{\beta} - h)$$

  is larger than $\chi^2_{j,1-\alpha}$ where ~~$z_{\alpha/2}$~~ is the usual ~~normal~~ critical value.
- Example :

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$$

Testing $H_0 : \beta_1 = \beta_3 = 0$

$$C_{2\times 4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, h = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$C\beta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Revisit the Graduate school admission data

```
mydata <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
modF <- glm(admit~gpa+gre+factor(rank), family=binomial,data=mydata)
summary(modF)
```

M_F: $\text{logit}(\text{Admit}) \sim \text{GPA} + \text{GRE} + \text{rank}$

```
##
## Call:
## glm(formula = admit ~ gpa + gre + factor(rank), family = binomial,
##     data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.989979   1.139951  -3.500 0.000465 ***
## gpa            0.804038   0.331819   2.423 0.015388 *
## gre            0.002264   0.001094   2.070 0.038465 *
## factor(rank)2 -0.675443   0.316490  -2.134 0.032829 *
## factor(rank)3 -1.340204   0.345306  -3.881 0.000104 ***
## factor(rank)4 -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
```

Revisit the Graduate school admission data

```
mydata <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
modR <- glm(admit~gre, family=binomial,data=mydata)
summary(modR)
```

M_R: logit(Admit) ~ GRE

```
##
## Call:
## glm(formula = admit ~ gre, family = binomial, data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1623  -0.9052  -0.7547   1.3486   1.9879
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.901344   0.606038  -4.787 1.69e-06 ***
## gre          0.003582   0.000986   3.633  0.00028 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 486.06  on 398  degrees of freedom
## AIC: 490.06
##
## Number of Fisher Scoring iterations: 4
```

Revisit the Graduate school admission data

① use LRT

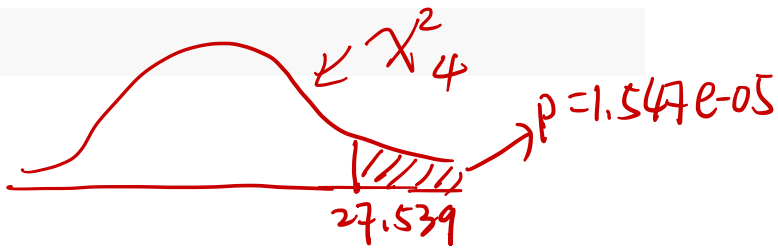
$$Mod_F : logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 GRE + \beta_3\ Rank_2 + \beta_4\ Rank_3 + \beta_5\ Rank_4$$

$$Mod_R : logit(\pi) = \beta_0 + \beta_1 GRE$$

Want to test

$$H_0 : \beta_1 = \beta_3 = \beta_4 = \beta_6 = 0, \quad H_a : \text{ at least one of them is nonzero}$$

```
anova(modR,modF,test="Chisq")
```




```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre
## Model 2: admit ~ gpa + gre + factor(rank)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       398     486.06 ← DR
## 2       394     458.52  4   27.539 1.547e-05 ***
##                  ↑ DF    DR-DF
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

df for $D_R - D_F$
$= df_R - df_F = 398 - 394$
$= 4$

② use method by constraint.

```
library(aod)
wald.test(b = coef(mod), Sigma = vcov(mod), Terms =c(2,4:6))
```

model: $\beta_0, \beta_1 \cdots \beta_5$
in R: $\beta_1\ \beta_2 \cdots \beta_6$

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 166.1, df = 4, P(> X2) = 0.0
```

Goodness of fit
(quality of fit)

Hosmer-Lemeshow Test

$$H_0 : logit(E(Y_i)) = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1}$$
$$H_a : logit(E(Y_i)) \neq \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1}$$

- Group cases based on values of estimated probabilities $\hat{\pi}$ into c groups
  - e.g. find c=6 to 9 groups based on percentiles
- Apply pearson χ^2 test to the groups
- Reject H_0 if $X^2 > \chi^2_{1-\alpha,c-2}$
- Week 7, slides 55, 56.

Goodness-of-fit for Bernoulli data

Pearson χ^2 (But not use for Tests)

- Test statistic:

$$X^2 = \sum_{i=1}^{c} \sum_{k=0}^{1} \frac{(O_{jk} - \hat{E}_{jk})^2}{\hat{E}_{jk}}$$

- The corresponding quantities (KNNL p.591)
  - c=n, $n_i = 1$
  - $O_{i0} = 1, O_{i1} = 1$
  - $E_{i0} = 1 - \hat{\pi}_i, E_{i1} = \hat{\pi}_i$

- Test statistic

$$X^2 = \sum_{i=1}^{n} \frac{[(1 - y_i) - (1 - \hat{\pi}_i)]^2}{1 - \hat{\pi}_i} + \sum_{i=1}^{n} \frac{(y_i - \hat{\pi}_i)^2}{\hat{\pi}_i}$$

$$= \sum_{i=1}^{n} \frac{(y_i - \hat{\pi}_i)^2}{1 - \hat{\pi}_i} + \sum_{i=1}^{n} \frac{(y_i - \hat{\pi}_i)^2}{\hat{\pi}_i} = \sum_{i=1}^{n} (y_i - \hat{\pi}_i)^2 \left[\frac{1}{1 - \hat{\pi}_i} + \frac{1}{\hat{\pi}_i}\right]$$

$$= \sum_{i=1}^{n} \frac{(y_i - \hat{\pi}_i)^2}{(1 - \hat{\pi}_i)\hat{\pi}_i}$$

Goodness-of-fit for Bernoulli data

Deviance (But not use for Tests)

- Test statistic: $G^2 = Dev(X_0, X_1, \ldots, X_{p-1})$ = Deviance($X_0, X_1, \cdots X_{p-1}$)

p covariates

$$G^2 = -2\sum_{i=1}^{c}[\sum_{j=1}^{n_i} Y_{ij}\log(\frac{\hat{\pi}_i}{\hat{\pi}_i}) + (n_i - \sum_{j=1}^{n_i} Y_{ij})\log(\frac{1-\hat{\pi}_i}{1-\hat{\pi}_i})]$$

- The corresponding quantities (KNNL. P592)
  - $c = n, n_i = 1, \sum_{j=1}^{n_i} Y_{ij} = Y_i, \hat{p}_i = \sum_{j=1}^{n_i} Y_{ij}/n_i = Y_i/n_i$
- Test statistic

$= -2[\ell(\hat{\beta}) - \ell_S(\hat{\beta})] = -2\ell(\hat{\beta})$

$$G^2 = -2\sum_{i=1}^{n}[\underbrace{Y_i\log(\frac{\hat{\pi}_i}{Y_i})}_{A_1} + \underbrace{(1-Y_i)\log(\frac{1-\hat{\pi}_i}{1-Y_i})}_{A_2}]$$

$$= -2\sum_{i=1}^{n}[Y_i\log\hat{\pi}_i + (1-Y_i)\log(1-\hat{\pi}_i)]$$

$Y_i = 1$, $A_1 = \log\hat{\pi}_i$, $A_2 = 0$

$Y_i = 0$, $A_1 = 0$, $A_2 = \log(1-\hat{\pi}_i)$

Goodness-of-fit for Binomial data

Pearson χ^2

$$H_0 : logit(E(Y_{ij})) = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1} \rightarrow M_1$$

$$H_a : logit(E(Y_{ij})) = \beta_0 + X\beta = \text{Saturated model} \rightarrow M_s$$

- In total, c groups, each with n_j cases
- Observed counts
  - O_{i0}: observed # of 0's in group i
  - O_{i1}: observed # of 1's in group i
- Expected counts
  - $E_{i0} = n_i(1 - \hat{\pi}_i)$: expected # of 0's in group i
  - $E_{i1} = n_i\hat{\pi}_i$: expected # of 1's in group i
- Test statistic: reject H_0 if

$$X^2 = \sum_{i=1}^{c}\sum_{k=0}^{1} \frac{(O_{jk} - E_{jk})^2}{E_{jk}} > \chi^2_{1-\alpha, c-p}$$

$df_1 - df_s$
$= (c-p) - (c-c)$
$= c-p$

Goodness-of-fit for Binomial data

Drop in Deviance

$$H_0 : logit(E(Y_{ij})) = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1} \quad \rightarrow \quad M_1$$

$$H_a : logit(E(Y_{ij})) = \beta_0 + X\beta \rightarrow \text{saturated model} \rightarrow M_S$$

- In total, c groups, each with n_j cases
- H_0: model of interest; H_a : saturated model
- model of interest: $E(Y_{ij}) = \pi_i$, $\widehat{E(Y_{ij})} = \hat{\pi}_i$
- Saturated model
  - $E(Y_{ij}) = p_i$, $\widehat{E(Y_{ij})} = \hat{p}_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i}$
- Test statistics: reject H_0 if

$$G^2 = 2 \sum_{i=1}^{c} \left\{ y_i \log \frac{y_i}{\hat{y}_i} + (n_i - y_i) \log \frac{n_i - y_i}{n_i - \hat{y}_i} \right\}, \ \hat{y}_i = n_i \hat{\pi}_i$$

$D_1 - D_S = -2(\ell(\hat{\beta}) - \ell_S(\hat{\beta}_1))$

- Test statistic: LR, also called deviance (D of saturated model always=0) **
  - Reject H_0 if $G^2 > \chi^2_{1-\alpha, c-p}$ $= df_1 - df_S$
  - Approximation $\chi^2_{1-\alpha, c-p}$ can be poor: unlike with the LR test, the quality of approximation does not improve with the sample size.

Diagnostics: residual

- Logistic Regression Residuals

① response residual

$$e_i = \begin{cases} 1 - \hat{\pi}_i, & Y_i = 1 \\ 0 - \hat{\pi}_i, & Y_i = 0 \end{cases}$$

② • Pearson residual (or standardized Pearson)

$$r_{p_i} = \frac{Y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)}}$$

  - e_i divided by the standard error of Y_i
  - $\sum_i^n r_{p_i}^2$ equals pearson X^2 (benoulli case)

- Studentized Pearson Residual

$$r_{P_i} = \frac{Y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)(1 - h_{ii})}}$$

  - e_i divided by the SE of $e_i \rightarrow$ unit variance
  - h_{ii} is the diagonal element of the hat matrix

$$H = \hat{W}^{1/2} X (X' \hat{W} X)^{-1} X' \hat{W}^{1/2}$$

$$\hat{W} = \text{diag}(\hat{\pi}_i(1 - \hat{\pi}_i))$$

Diagnostics: residual

- Deviance Residual

$$d_i = \text{sign}(Y_i - \hat{\pi}_i)\sqrt{-2[Y_i \log(\hat{\pi}_i) + (1 - Y_i)\log(1 - \hat{\pi}_i)]}$$

  - the signed square root of the contribution of Y_i to the model deviance

- Analysis of residuals
  - unknown distribution of residuals under true model
  - plot residual by predicted value. A flat lowess smooth to this plot suggests good model

- Other summaries as in linear regression
  - DFFITS, DFBETAs
  - $\Delta_{\chi^2}, \Delta_{dev}$, cook's distance (See KNNL P598)

R^2 statistics for logistic regression

- There are many different ways to calculate R^2 for logistic regression and, unfortunately, no consensus on which one is best.
  - Mittlbock and Schemper (1996) reviewed 12 different measures;
  - Menard (2000) considered several others.
- The two methods that are most often reported in statistical software appear to be one proposed by McFadden (1974) and another that is usually attributed to Cox and Snell (1989) along with its "corrected" version.
  - L_0 be the value of likelihood without predictors. L_M be the value of likelihood for the model being estimated.
  - McFadden's R^2

$$R^2_{McF} = 1 - \frac{ln(L_m)}{ln(L_0)}$$

  - The Cox and Snell R^2 is

$$R^2_{CS} = 1 - (L_0/L_m)^{2/n}$$

Graphical check of the fit

- Partition of the observation into groups of covariates pattern X_i
- Haldane (1956) recommend to plot $\hat{\eta}$ against x

$$\hat{\eta}_i = log\frac{y_i + 0.5}{n_i - y_i + 0.5}$$

+0.5: to fix zero y_i cases.

- the plot should be roughly linear if the model is appropriate for the data
- when all $n_i = 1$ or all n_i are small, one can group the data with nearby x values to make this plot.

$\hat{\eta} = x_i\hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_{p-1} x_p$

$\rightarrow$ η and x_i : linear associated.

Take a break, and see you on Thursday

Revisit: GLM - conditional distribution

- In GLM, it is assumed that $Y|\vec{X}$ belongs to a canonical exponential family which p.d.f has the following form

pdf of GLM :

$$f(y|x) = \exp\left\{\frac{\theta(x)y - b\big(\theta(x)\big)}{a(\phi)} + c(y;\phi)\right\}$$

for some known function $a(\cdot), b(\cdot), c(\cdot,\cdot)$.

  - $\theta(x)$ is called a canonical parameter
  - ϕ is called a dispersion parameter

- Binomial distribution: $Y|x \sim B(m, p(x)), 0 < p(x) < 1$

$$P(Y = y|X) = \binom{m}{y} p^y (1-p)^y = \exp\{y \log(\frac{p}{1-p}) + m \log(1-p) + \log \binom{m}{y}\};$$

  - $\theta = \log(\frac{p}{1-p}), b(\theta) = m \log(1-p), c(y;\phi) = \log \binom{m}{y}$ with $\phi = 1$
  - $g(\mu) = g(p) = \log(\frac{p}{1-p})$

Overdispersion

- Logistic model:

$$Y_i \sim Bin(n_i, \pi_i) \text{ independent }, \pi_i = e^{X\beta}/(1+e^{X\beta})$$

$$E(Y_i) = n_i p_i,\ var(Y_i) = n_i p_i(1-p_i)$$

  - If one assumes that p_i is correctly modelled, but the observed variance is larger or smaller than the expected from the logistic model give by $n_i p_i(1-p_i)$, one speaks of under or overdispersion.
  - In application one often observes only overdispersion, so we concentrate on modelling overdispersion.

- How to detect overdispersion?
  - If the logistic model is correct, then $D \sim \chi^2_{n-p}$, therefore $D > n-p = E(\chi^2_{n-p})$ can indicate overdispersion.

→ use fact: $D \sim \chi^2_{n-p} \rightarrow \hat{\phi} = \frac{D}{n-p}$

→ use fact: $\chi^2 \sim \chi^2_{n-p} \rightarrow \hat{\phi} = \frac{\chi^2}{n-p}$

In R, quasi-binomial or quasipoisson

- Reasons for overdispersion
  - variation among the success probabilities or
  - correlation between the binary responses.
  - Both reasons are the same, since variation leads to correlation and vice versa. But for interpretative reasons one explanation might be more reasonable than the other.

Overdispersion from correlation

correlated data

- Suppose $Y_1, \ldots, Y_n$ are not independent Bernoulli RV.
- suppose all pairs (Y_i, Y_j) have a same correlation ρ

$$
\begin{aligned}
Var(Y) &= Var(\sum_i Y_i), \; Y = \sum_{i=1}^{n} Y_i \\
&= \sum_{i}^{n} V(Y_i) + \sum_{i \neq j} cov(Y_i, Y_j) \\
&= \sum_{i}^{n} V(Y_i) + \sum_{i \neq j} \rho \sqrt{V(Y_i) V(Y_j)} \\
&= n\pi(1-\pi) + n(n-1)\rho\pi(1-\pi), \text{ assume } \rho > 0 \\
&> n\pi(1-\pi)
\end{aligned}
$$

$\rho \sqrt{\pi(1-\pi)\pi(1-\pi)}$

$var(Y_1 + Y_2) = V(Y_1) + V(Y_2) + 2\,cov(Y_1, Y_2)$

$var(\sum_1^n Y_i) = \sum_1^n V(Y_i) + \sum_{i \neq j} C_{ij}$

$cor(X, Y) = \frac{cov(X,Y)}{\sqrt{V(Y)V(X)}}$

$cov(X, Y) = \rho \sqrt{V(Y)V(X)}$

- The variance exceeds the variance of the binomial distribution.

Overdispersion from clustered data

- suppose $Y = \sum_i Y_i | \pi \sim Bin(\pi)$, π is different, then Y is from different Bin(π)
- suppose π is RV. $E(\pi) = p, V(\pi) = p(1-p)$

$$
\begin{aligned}
E(Y) &= E(E(Y|\pi)) = p \\
V(Y) &= V(E(Y|\pi)) + E(V(Y|\pi)) \\
&= V(n\pi) + E(n\pi(1-\pi)) \\
&= np(1-p) + n(n-1)V(\pi) \\
&> n\pi(1-\pi)
\end{aligned}
$$

- The variance exceeds the variance of the binomial distribution.

Overdispersion

- Estimating the dispersion parameter for $Y_i \sim Bin(n_i, \pi_i)$

$$X_S^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{\mu}_i)^2}{V(\mu_i)\phi}$$

  - The scaled Pearson chi-squared statistic is defined as

$$X_s^2 = X^2/\phi \quad \Rightarrow \quad E(X_S^2) = \frac{E(X^2)}{\phi} = n-p$$

$$\Rightarrow \phi = \frac{E(X^2)}{n-p}$$

$$\Rightarrow \hat{\phi} = \frac{X^2}{n-p}$$

  - it turns out that, if the model is specified correctly,

$$X_s^2 \sim \chi^2_{n-p}, E(\chi^2_{n-p}) = n - p$$

  - Estimate of ϕ

Week 7. Slide 16

$E(Y) = b'(\theta)$

$V(Y) = a(\phi)\, b''(\theta)$

$$\hat{\phi} = \frac{X^2}{n-p} = \frac{\text{sum of squared Pearson residual}}{d.f.}$$

- $\hat{\phi} >> 1$ indicates evidence of overdispersion
- since ϕ does not affect $E(Y_i)$, modelling overdispersed does not change $\hat{\beta}$.
- $SE(\hat{\beta})$ is multiplied by $\sqrt{\hat{\phi}}$ $\quad SE_\phi(\hat{\beta}) = \sqrt{\hat{\phi}}\, SE(\hat{\beta})$

- Overdispersion as such doesn't apply to Bernoulli data. McCullagh and Nelder (1989) point out that overdispersion is not possible if $n_i = 1$. If y_i only takes values 0 and 1, the it must be distributed as Bernoulli(π_i) and its variance must be $\pi_i(1 - \pi_i)$.

Comparing nested models in presence of overdispersion

- regular likelihood-based approaches (LRT, AIC) are not applicable
- F test approximates deviance-based LR test

$$F = \frac{D_R - D_F}{df_R - df_F} / \hat{\phi}_F \sim_{H_0} F_{df_R - df_F, df_F}$$

- model strategy
  - fit the full model (with all predictors)
  - estimate $\hat{\phi}_F$
  - compare nested models with F test to reduce the number of predictors.

Revisit the smoking data

```
mod = glm(cbind(dead,alive)~factor(smoker)+age,data=smoking, family=binomial)
summary(mod)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ factor(smoker) + age, family = binomial,
##     data = smoking)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72545  -0.22836   0.00005   0.19146   0.68162
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.8601     0.5939  -6.500 8.05e-11 ***
## factor(smoker)1     0.4274     0.1770   2.414 0.015762 *
## age25-34            0.1201     0.6865   0.175 0.861178
## age35-44            1.3411     0.6286   2.134 0.032874 *
## age45-54            2.1134     0.6121   3.453 0.000555 ***
## age55-64            3.1808     0.6006   5.296 1.18e-07 ***
## age65-74            5.0880     0.6195   8.213  < 2e-16 ***
## age75+             27.8073 11293.1430   0.002 0.998035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 641.4963  on 13  degrees of freedom
## Residual deviance:   2.3809  on  6  degrees of freedom
## AIC: 65.377
```

Revisit the smoking data

two methods < (A) (B)

```
(phi=sum(residuals(mod, type = "pearson")^2)/mod$df.residual)
```

$\phi = \frac{X^2}{n-p}$

```
## [1] 0.395663
```

= ϕ

(A)

```
summary(mod,dispersion=phi)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ factor(smoker) + age, family = binomial,
##     data = smoking)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72545  -0.22836   0.00005   0.19146   0.68162
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.8601     0.3736 -10.333  < 2e-16 ***
## factor(smoker)1     0.4274     0.1114   3.838 0.000124 ***
## age25-34            0.1201     0.4319   0.278 0.781002
## age35-44            1.3411     0.3954   3.392 0.000694 ***
## age45-54            2.1134     0.3850   5.489 4.04e-08 ***
## age55-64            3.1808     0.3778   8.420  < 2e-16 ***
## age65-74            5.0880     0.3897  13.057  < 2e-16 ***
## age75+             27.8073  7103.5849   0.004 0.996877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 0.395663)
##
```

Revisit the smoking data

(b)

```
mod2 = glm(cbind(dead,alive)~factor(smoker)+age,data=smoking, family=quasibinomial())
summary(mod2)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ factor(smoker) + age, family = quasibinomial(),
##     data = smoking)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.72545  -0.22836   0.00005   0.19146   0.68162
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.8601     0.3736 -10.333 4.80e-05 ***
## factor(smoker)1    0.4274     0.1114   3.838 0.008576 **
## age25-34           0.1201     0.4319   0.278 0.790334
## age35-44           1.3411     0.3954   3.392 0.014640 *
## age45-54           2.1134     0.3850   5.489 0.001531 **
## age55-64           3.1808     0.3778   8.420 0.000153 ***
## age65-74           5.0880     0.3897  13.057 1.24e-05 ***
## age75+            27.8073  7103.5849   0.004 0.997004
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.395663)
##
##     Null deviance: 641.4963  on 13  degrees of freedom
## Residual deviance:   2.3809  on  6  degrees of freedom
## AIC: NA
```

compare with slide 37 : same output.

Classification with Logistic Regression

Classification

- Have a set of predictors $\{X_1^*, X_2^*, \ldots, X_p^*\}$ to predict Y^*
- Fit a logistic model and pick a cut point
  - Like 0.5
- If $\hat{\pi} > 0.5$, predict $Y^* = 1$
- If $\hat{\pi} < 0.5$, predict $Y^* = 0$
  - you can use a cut point other than 0.5 too.
- There are many other classification in use

or use cross-validation method to find the cut pt.

Confusion Matrix

Truth\ Prediction	Postive	Negative	prop (row)
$D = 1$	TP	FN	Sensitivity=TPR= $\frac{TP}{TP+FN}$
$D^c = 0$	FP	TN	Specificity=TNR= $\frac{TN}{TN+FP}$
prop (col)	PPV= $\frac{TP}{TP+FP}$	NPV= $\frac{TN}{TN+FN}$	

- TP: true positive; TN: true negative
- FP: false positive (type I error); FN: false negative (type II error)
- TPR (sensitivity): True Positive Rate = TP/(TP+FN)
- TNR (specificity): True Negative Rate = TN/(TN+FP), 1-specificity=1-TNR=FPR
- PPV: Positive Predictive Value = TP/(TP+FP)
- FDR= 1- PPV: False Discovery Rate (the complement of the PPV)
- NPV: Negative Predictive Value= TN/(FN+TN)
- FOR = 1- NPV: False Omission Rate (the complement of the NPV)
- Mean of error (misclassification) rate = (FN+FP)/(TP+FP+FN+TN)

Confusion Matrix Example

- Truth: $Y = c(1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0)$
- Prediction: $Y^* = c(1,1,0,1,0,1,0,0,1,0,0,0,1,1,0,0)$

Truth\ Prediction	Postive	Negative	prop (row)
$D = 1$ (6)	TP=4	FN=2	Sensitivity=TPR= 4/6
$D^c = 0$ (10)	FP=3	TN=7	Specificity=TNR= 7/10
prop (col)	PPV=4/7	NPV=7/9	

Mean of misclassification rate $= \frac{2+3}{6+10} = 5/16$

Confusion Matrix

- From Wikipedia

		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

https://en.wikipedia.org/wiki/Positive_and_negative_predictive_values

Diagnostic Accuracy

- High sensitivity (TPR) makes a good screening test
- High specificity (TNR) makes a good confirmatory test
- A screening test followed by a confirmatory test is a good (albeit expensive) diagnostic procedure.

ROC curves and AUC

- ROC = Receiver Operating Characteristic
  - A plot of sensitivity (TPR) vs 1-specificity (FPR=1- TNR)
- ROC: originally designed to grade radar detection methods for German planes
- Decades later, their usefulness in classification problems was realized
  - But the name stuck
- AUC: = Area Under the Curve. Most of the time, AUC mean AUROC (area under ROC)

Classification for diabetes data

- Read in data

```
fdata = read.table("/Users/Wei/TA/Teaching/STA303-2017S/303Week9/diabetes.data",sep=",")
str(fdata)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ V1: int  6 1 8 1 0 5 3 10 2 8 ...
##  $ V2: int  148 85 183 89 137 116 78 115 197 125 ...
##  $ V3: int  72 66 64 66 40 74 50 0 70 96 ...
##  $ V4: int  35 29 0 23 35 0 32 0 45 0 ...
##  $ V5: int  0 0 0 94 168 0 88 0 543 0 ...
##  $ V6: num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ V7: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ V8: int  50 31 32 21 33 30 26 29 53 54 ...
##  $ V9: int  1 0 1 0 1 0 1 0 1 1 ...
```

X's

Y

```
dim(fdata)
```

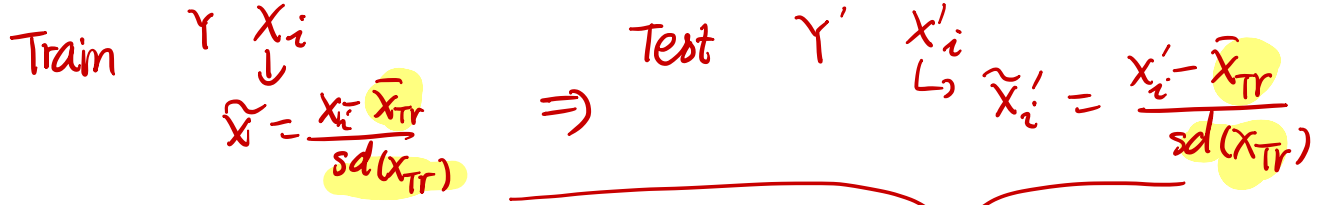
```
## [1] 768   9
```

```
#summary(fdata)
train= fdata[1:400,]
test= fdata[-c(1:400),]
```

← get first 400 data lines as training data

← rest 368 data lines serve as testing data.

Classification for diabetes data



- Data prepossessing of all the predictors

Method A • standardized the training data set, then using the mean and sd in training data to standardize the corresponding predictor in the testing data.

method B • log-transformation to all the predictors

```
xtrain_std = scale(train[,1:8]) # (x-mean(x))/sd(x)
ytrain_std = as.factor(train[,9])
trmu =apply(train[,1:8],2,mean) # mean of each column of train data
trsd =apply(train[,1:8],2,sd)   # sd of each column of train data
xtest_std = scale(test[,1:8], center=trmu, scale=trsd)
ytest_std = as.factor(test[,9])

## log transform of X in training data set
xtrain_log = as.matrix(log(train[,1:8]+0.5))
ytrain_log = as.factor(train[,9])
xtest_log = as.matrix( log(test[,1:8]+0.5) )
ytest_log = as.factor(test[,9])
```

(A) (B)

scale( ): could standarize multi-columns!

① Don't standarize all data then split it into Train/Test

- If so, for Train data: $E(\tilde{X}) \neq 0$ $sd(\tilde{X}) \neq 1$
- If so: we use partial information from "future" ∈ testing data

Classification for diabetes data

must be matrix

$\lambda|\beta|^2 \rightarrow \alpha = 0$: Ridge penalty

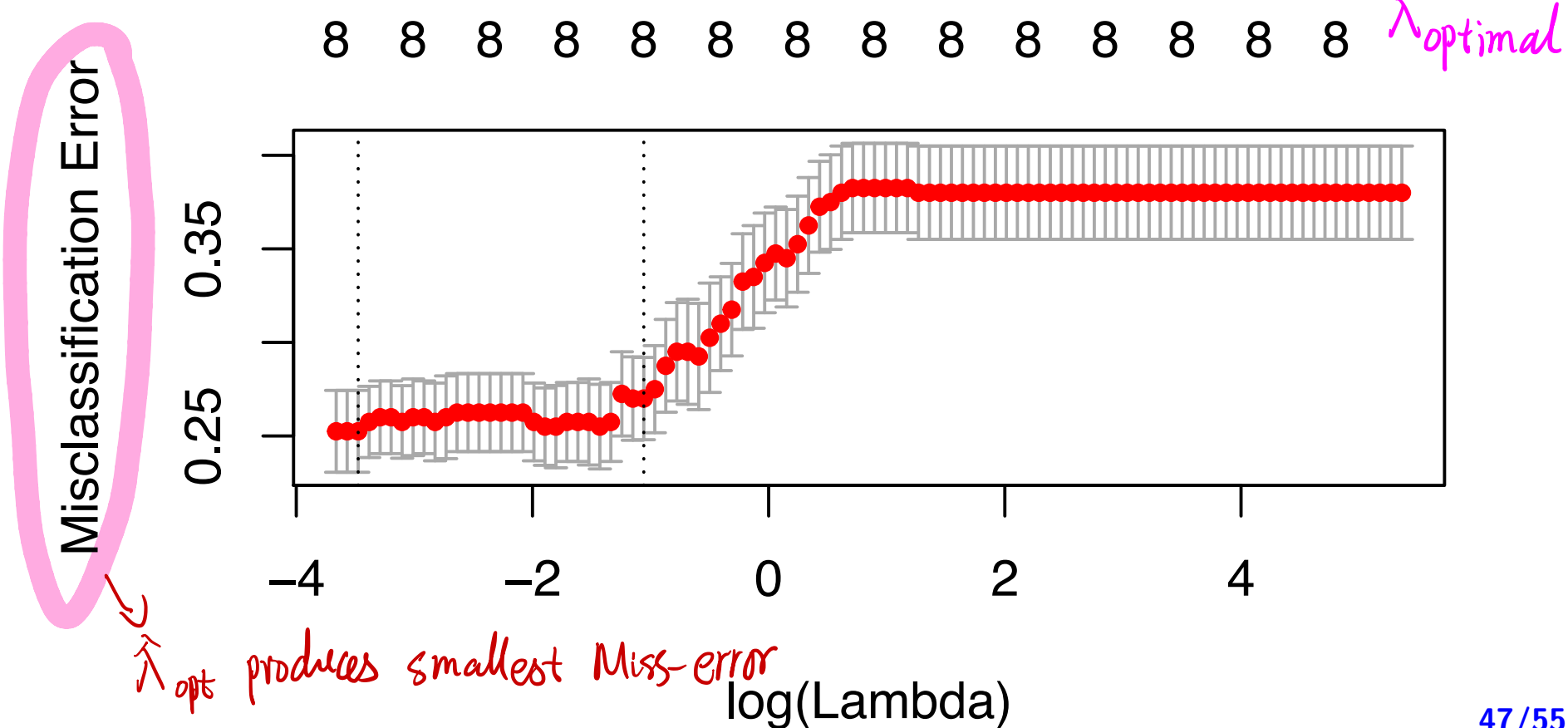
$\lambda|\beta| \rightarrow \alpha = 1$: LASSO penalty

```
library(glmnet)
cvstd <- cv.glmnet(xtrain_std,ytrain_std,family = "binomial",
                   alpha = 0, nfolds = 10, type.measure = "class")
# optimal lambda from 10-fold cross-validation by ridge regression (alpha=0)
(lamstd = cvstd$lambda.min)
```

```
## [1] 0.03092815
```

```
plot(cvstd)
```

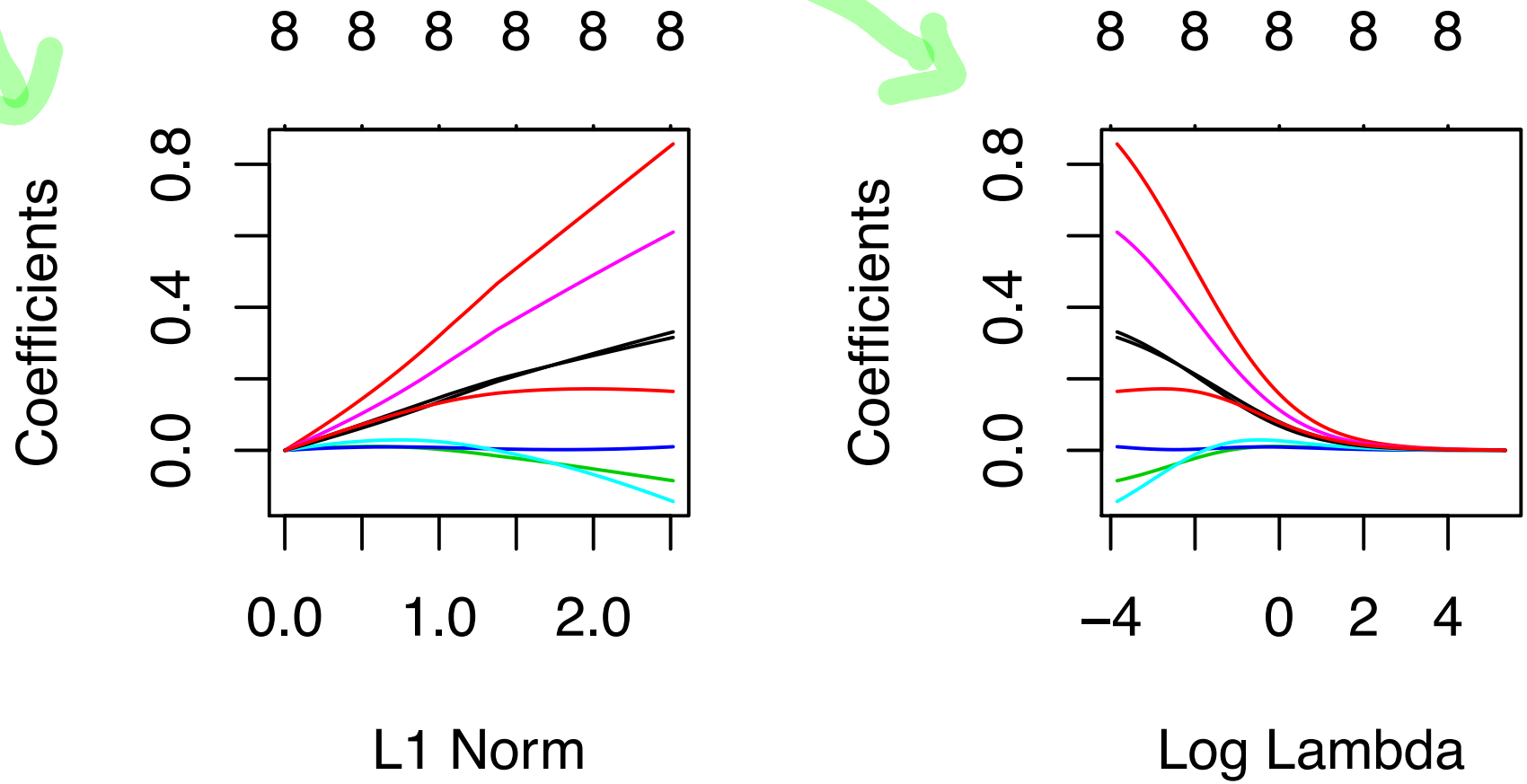
use c-v to find $\hat{\lambda}_{optimal}$



$\hat{\lambda}_{opt}$ produces smallest Miss-error

Classification for diabetes data

```r
# fit the model
# alpha=0: Ridge regression, alpha=1: LASSO regression
std_rlogit <- glmnet(xtrain_std, ytrain_std, family="binomial", alpha=0)
par(mfrow=c(1, 2))
plot(std_rlogit, "norm",   label=TRUE)
plot(std_rlogit, "lambda", label=TRUE)
```



Classification for diabetes data

```
# predicting with the test set
std_predtest <- predict(std_rlogit, xtest_std, type="class", s=lamstd)
cfm_test <- table(std_predtest,ytest_std)
cfm_test #   confusion matrix
```

on test data

```
##              ytest_std
## std_predtest   0   1
##            0 228  51
##            1  24  65
```

```
(prederror_train = (cfm_test[1,2]+cfm_test[2,1])/length(ytest_std) )
```

```
## [1] 0.2038043
```

$= (51+24)/(228+51+24+65)$

Classification for diabetes data

To produce ROC with AUC value

```
library(ROCR)
library(ggplot2)
# ROC curve for standardized data
std_probtest <- predict(std_rlogit, xtest_std, type="response", s=lamstd)
std_predresp <- prediction(std_probtest, ytest_std)
perf_std <- performance(std_predresp, measure = "tpr", x.measure = "fpr")

# TPR = true positive rate
tpr.ptstd <- attr(perf_std, "y.values")[[1]]
# FPR = false positive rate
fpr.ptstd <- attr(perf_std,"x.values")[[1]]

# AUC = area under the curve
auc.std <- attr(performance(std_predresp, "auc"), "y.values")[[1]]
auc1<- signif(auc.std, digits=3)

roc.std <- data.frame(FPR=fpr.ptstd, TPR=tpr.ptstd, model="GLM-Ridge")

ggplot(roc.std, aes(x=FPR, y=TPR, ymin=0, ymax=TPR)) +geom_point()+
  geom_ribbon(alpha=0.2)+
  geom_abline(intercept =0, slope =1,lty=2)+
  ggtitle(paste0("ROC Curve for Standardized Data with AUC=", auc1))+
  theme(plot.title = element_text(hjust = 0.5))
```

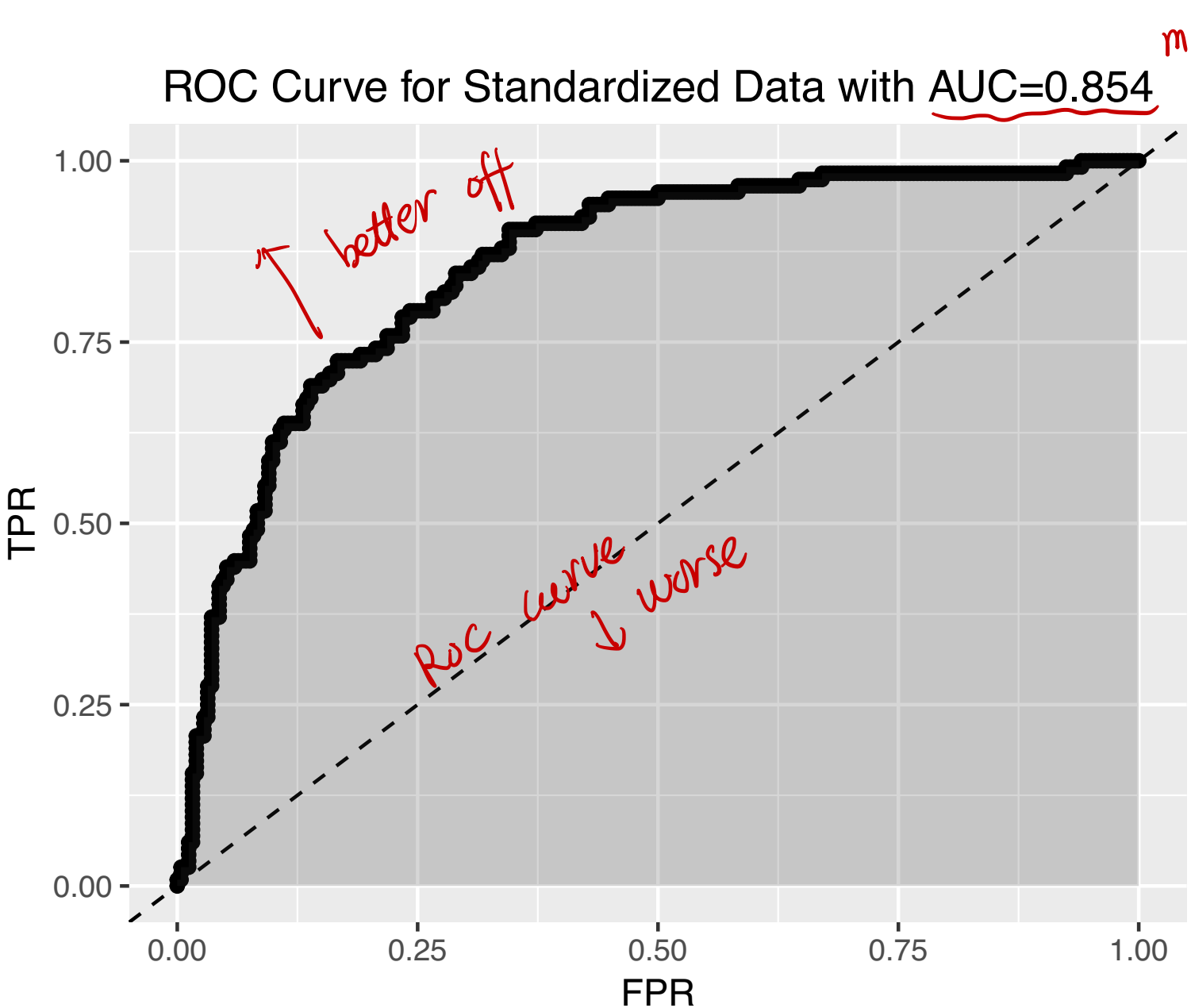
ROC: plot of TPR~FPR

ROC plot

adding line Y~X

centering title

Classification for diabetes data



measure how good of a classification

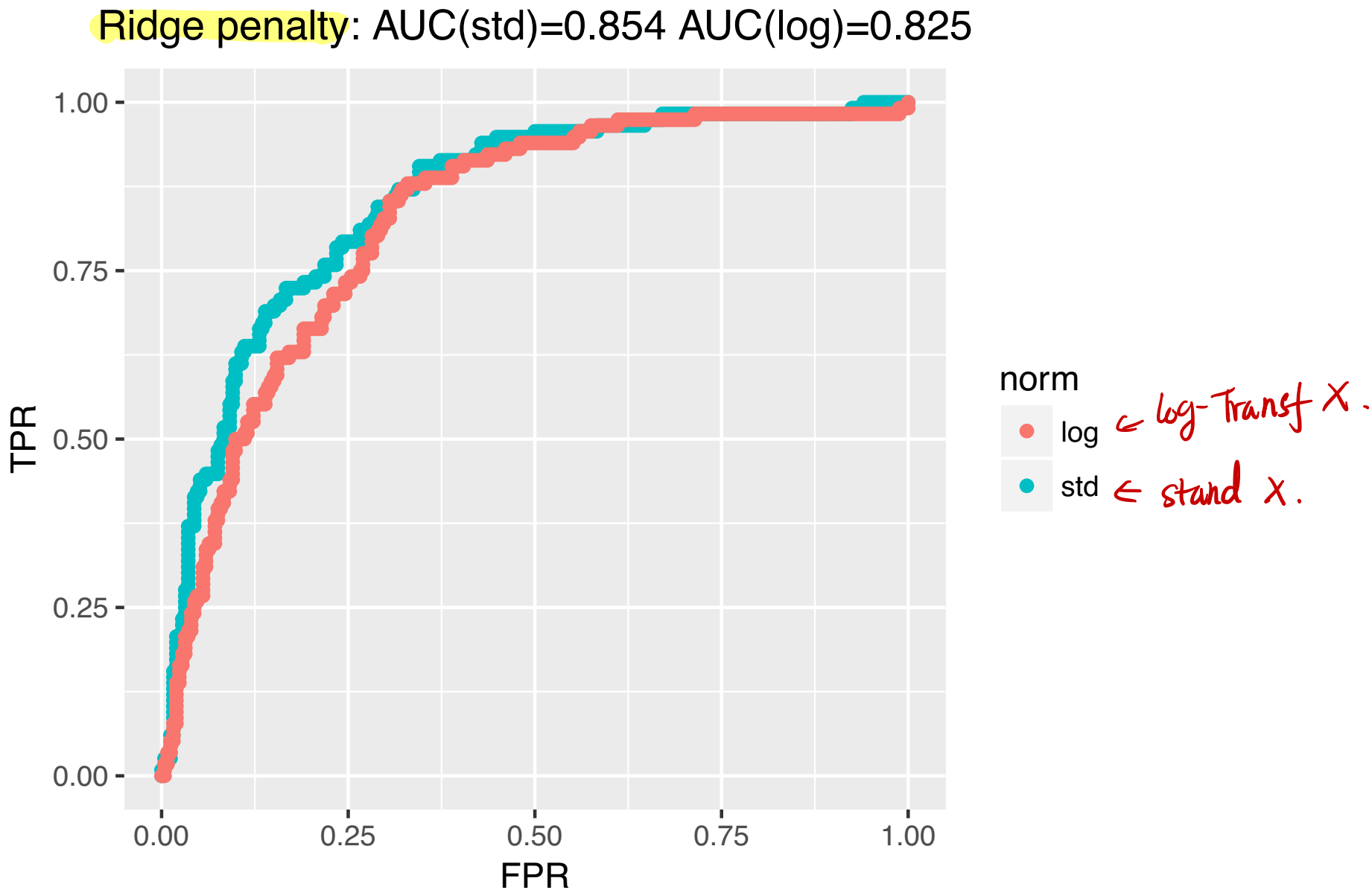
① far above 0.5 ⇒ good

② ≈ 0.5 : random

③ < 0.5 : Let go of this method

Classification for diabetes data

std better off.

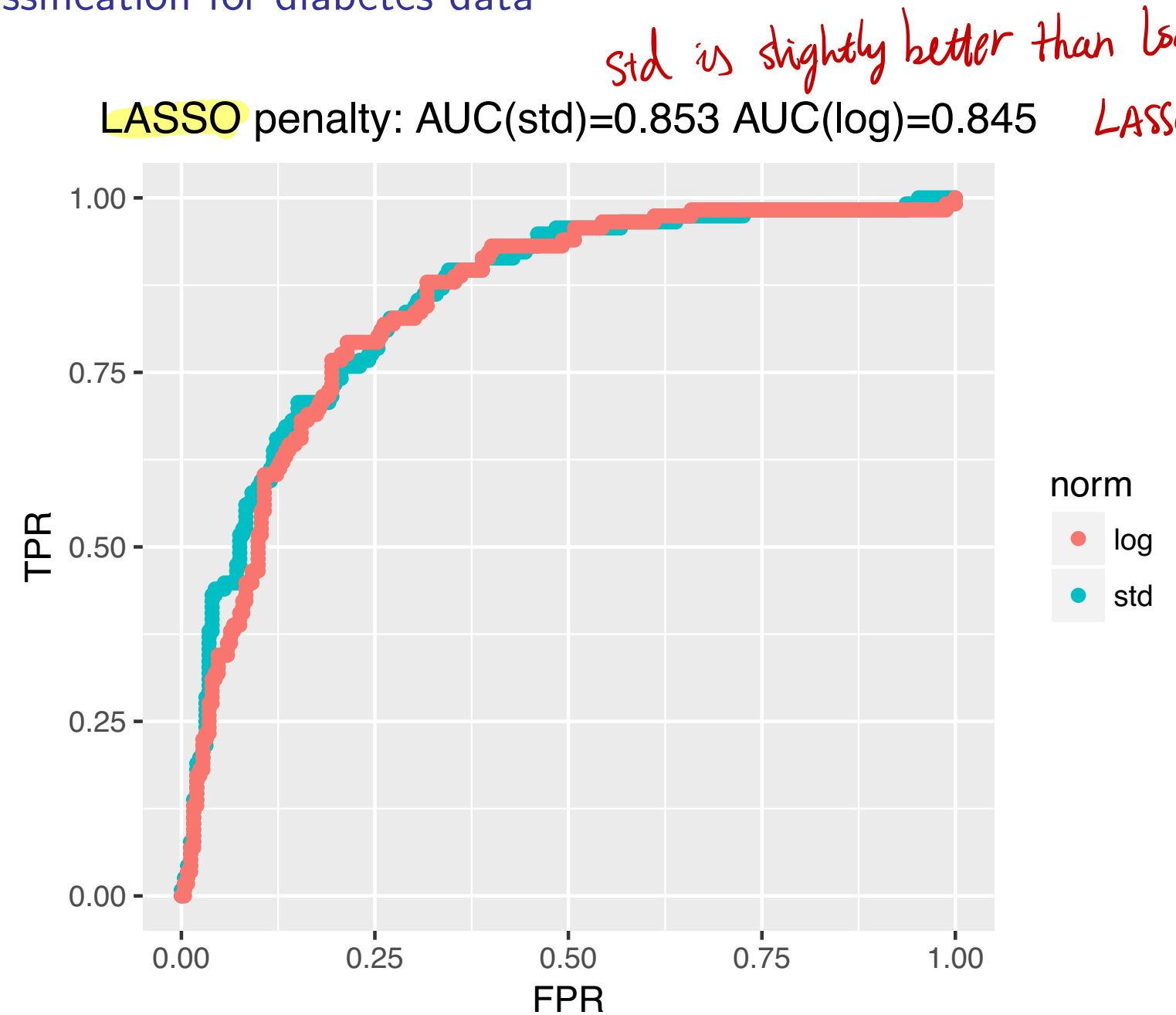


log-Transf X.

stand X.

Classification for diabetes data

std is slightly better than log using Lasso penalty.



Classification for diabetes data

R code: how to put two ROC curves in one plot using ggplot2.

```
##Put two ROCs in one plot
tworoc <- rbind(roc.std[,1:2], roc.log[,1:2])
tworoc$norm<- c(rep('std', dim(roc.std)[1]), rep('log', dim(roc.log)[1]))
ggplot(tworoc, aes(FPR, TPR, group = norm, col = norm)) + geom_point()+
  ggtitle(paste0("LASSO: AUC(std)=", auc1,"  AUC(log)=",auc2))+
  theme(plot.title = element_text(hjust = 0.5))
```

After Lecture This Week

Practice problems

- Review all the slides
- Try all the R example in slides.

Topics for next week:

- Topic in the following week after midterm
  - Poisson regression
  - Introduction to linear mixed effect model