Homework 3

Jieqi Tu

3/30/2021

5.3

(a) Construct the ROC curve for the toy example in Section 5.4.2. with complete separation.

```
# data import
x = c(1, 2, 3, 4, 5, 6)
y = c(1, 1, 1, 0, 0, 0)

# fit model
toy.model = glm(y~x, family = "binomial")
```

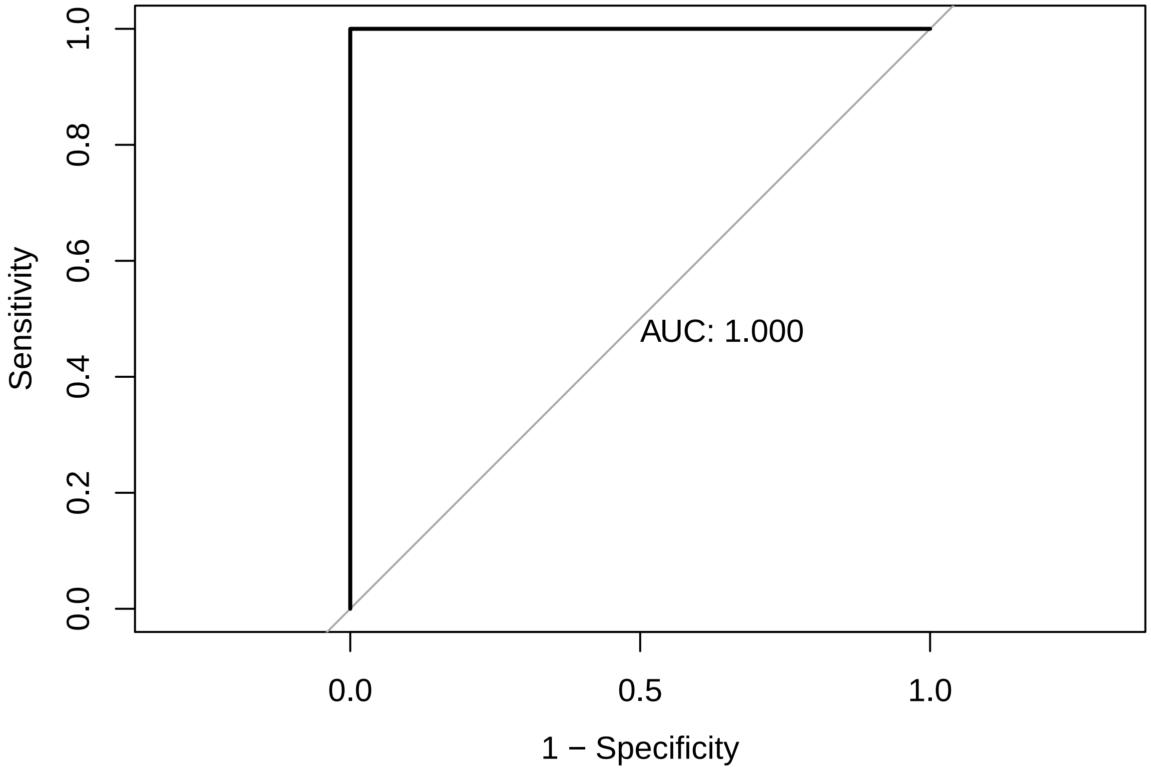
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
toy.pred = predict(toy.model, newdata = data.frame(x))
toy.roc = roc(y, toy.pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(toy.roc, legacy.axes = T, print.auc = T)
```



```
summary(toy.model)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##          1           2           3           4           5           6
##  2.110e-08   2.110e-08   1.052e-05  -1.052e-05  -2.110e-08  -2.110e-08
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    165.32  407521.43       0        1
## x              -47.23  115264.41       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.3178e+00  on 5  degrees of freedom
## Residual deviance: 2.2152e-10  on 4  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 25
```

From the result we could see that, when fitting the GLM, we have a warning message saying that fitted probabilities numerically 0 or 1 occurred. This implies complete/semi-complete separation occurred. We could see that the AUC = 1. Therefore, in this case, we have complete separation.The standard error of the estimated coefficient for x is very large.

(b) Add two observations at x = 0.5, one with y = 1 and one with 0.

```
x = c(1, 2, 3, 3.5, 3.5, 4, 5, 6)
y = c(1, 1, 1, 1, 0, 0, 0, 0)

toy.model2 = glm(y~x, family = "binomial")
```

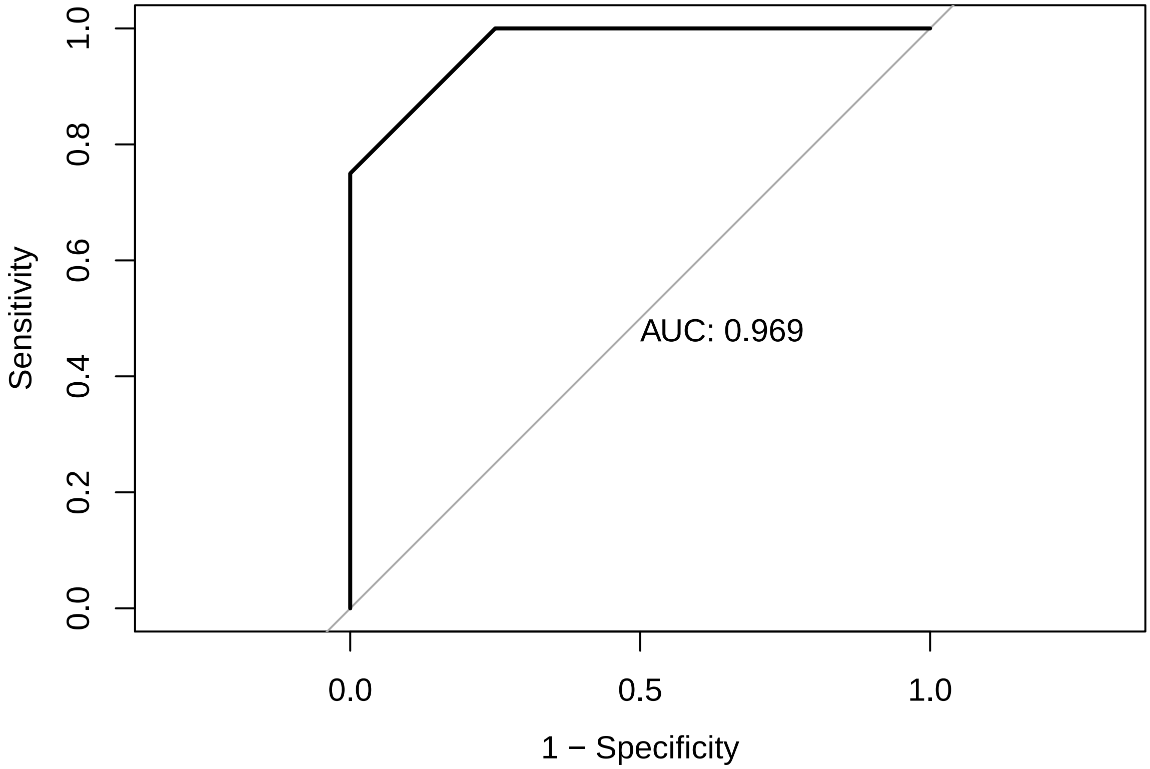
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
toy.pred = predict(toy.model2, newdata = data.frame(x))
toy.roc = roc(y, toy.pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(toy.roc, legacy.axes = T, print.auc = T)
```



```r
summary(toy.model2)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.17741  -0.00002   0.00000   0.00002   1.17741
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   137.32   54599.64   0.003    0.998
## x             -39.23   15599.90  -0.003    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11.0904  on 7  degrees of freedom
## Residual deviance:  2.7726  on 6  degrees of freedom
## AIC: 6.7726
##
## Number of Fisher Scoring iterations: 21
```

In this case, we have AUC $= 0.969$.But we also have the warning message. So we are facing the semi-complete separation. The standard error for estimated coefficient for x is still very large.

Then we want to construct a toy data set with n = 8 and the area under the ROC curve equals 0.5.

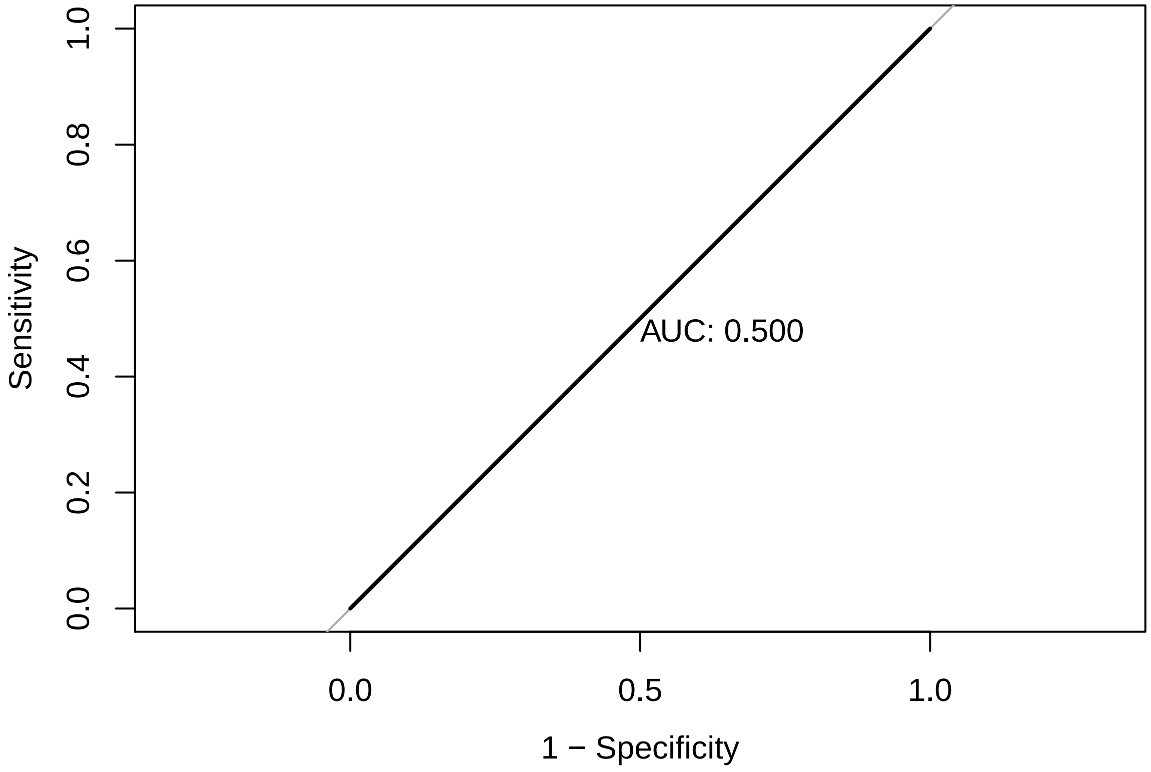
```r
# construct new dataset
x.new = c(1, 1, 2, 2, 3, 3, 4, 4)
y.new = c(1, 0, 1, 0, 1, 0, 1, 0)

# fit new model for new dataset
toy.model3 = glm(y.new~x.new, family = "binomial")
toy.pred = predict(toy.model3, newdata = data.frame(x.new))
toy.roc = roc(y.new, toy.pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(toy.roc, legacy.axes = T, print.auc = T)
```



```r
summary(toy.model3)
```

```
##
## Call:
## glm(formula = y.new ~ x.new, family = "binomial")
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.177  -1.177   0.000   1.177   1.177
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.022e-16  1.732e+00       0        1
## x.new       -2.809e-16  6.325e-01       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11.09  on 7  degrees of freedom
## Residual deviance: 11.09  on 6  degrees of freedom
## AIC: 15.09
##
## Number of Fisher Scoring iterations: 2
```

In this case, we have AUC = 0.5 exactly and the ROC looks like a straight line. In this case, the standard error of the estimated coefficient for x is reasonable.

5.5

We now know that $n_i y_i \sim Binomial(n_i, \pi_i)$ and $\mu_i = E(y_i) = \pi_i$. In addition, since $\pi_i = F(\sum_j \beta_j x_{ij})$, we also have $n_i = F^{-1}(\pi_i)$ (as F is absolutely continuous).Another equation is $n_i = \sum_j \beta_j x_{ij}$.

According to the likelihood equation, we have:

$$\begin{aligned}\frac{\partial L(\beta)}{\partial \beta_j} &= \sum_{i=1}^{N} \frac{n_i(y_i - \mu_i)}{var(y_i)} \cdot \frac{\partial \mu_i}{\partial n_i} \\ &= \sum_{i=1}^{N} \frac{n_i(y_i - \mu_i)}{var(y_i)} \cdot f(n_i)[f(\cdot) is the pdf corresponding to F(\cdot)]\end{aligned}$$

Therefore, we obtain:

$$w_i = (\frac{\partial \mu_i}{\partial n_i})^2 / var(y_i) = f^2(n_i)/var(y_i)$$

Then, let W be the diagonal matrix with w_i as the main diagonal elements. Hence the information matrix for MLE $\hat{\boldsymbol{\beta}}$ $J = X'WX$ and $var(\hat{\boldsymbol{\beta}}) = J^{-1}$.

5.9

Use conditional logistic regression to test $H_0 : \beta_1 = 0$ against $H_1 : \beta_1 < 0$ for the toy example in Section 5.4.2. According to the textbook page #175-176, we could know that„ The p-value is $1 - P(s_1 \geq t | s_1 + s_2)$, for observed value t for s_1.

```r
library(survival)
# data import
x = c(1, 2, 3, 4, 5, 6)
y = c(1, 1, 1, 0, 0, 0)
clogistic = clogit(y~x)
```

```
## Warning in coxexact.fit(X, Y, istrat, offset, init, control, weights =
## weights, : Ran out of iterations and did not converge
```

```r
summary(clogistic)
```

```
## Call:
## coxph(formula = Surv(rep(1, 6L), y) ~ x, method = "exact")
##
##   n= 6, number of events= 3
##
##         coef  exp(coef)   se(coef)      z Pr(>|z|)
## x -1.856e+01  8.687e-09  1.073e+04 -0.002    0.999
##
##   exp(coef) exp(-coef) lower .95 upper .95
## x 8.687e-09  115110695         0       Inf
##
## Concordance= 1  (se = 0 )
## Likelihood ratio test= 5.99  on 1 df,   p=0.01
## Wald test            = 0  on 1 df,   p=1
## Score (logrank) test = 3.86  on 1 df,   p=0.05
```

5.14

Assuming $\pi_1 = \pi_2 = \cdots = \pi_N = \pi$, then the log likelihood would be

$$L(\pi) = \sum_{i=1}^{N} y_i log(\pi) + (n_i - y_i) log(1-\pi)$$

Take the first derivative, we can get

$$L'(\pi) = \frac{\sum y_i}{\pi} - \frac{\sum n_i - y_i}{1-\pi}$$

Set it equal to 0, we can get

$$\hat{\pi} = (\sum y_i)/(\sum n_i)$$

And the second derivative of $L(\pi)$ also confirms that $\hat{\pi}$ maximizes the likelihood function. Then the Pearson statistic for ungrouped data (when $ n_i=1$) is:

$$\begin{aligned}
\chi^2 &= \sum \frac{(observed - fitted)^2}{fitted} \\
&= \sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{(y_{ij} - \hat{\pi})^2}{\hat{\pi}} + \frac{[1 - y_{ij} - (1-\hat{\pi})]^2}{1-\hat{\pi}} \\
&= \sum_{i=1}^{N} \sum_{j=1}^{n_i} \frac{(y_{ij} - \hat{\pi})^2}{\hat{\pi}(1-\hat{\pi})} \\
&= \frac{N\hat{\pi}(1-\hat{\pi})}{\hat{\pi}(1-\hat{\pi})} = N
\end{aligned}$$

Since the Pearson statistic $\chi^2 = N$, the statistic is not informative for us to test the goodness-of-fit of the null model.

5.15

The log likelihood is $\sum_i [y_i log \pi_i + (1-y_i) log(1-\pi_i)]$. For the saturated model, we have $\hat{\pi}_i = y_i$ and the value of the log likelihood of saturated model equals 0 (because y_i can only take value of 0 and 1).

$$\begin{aligned}
D(y; \hat{\boldsymbol{\mu}}) &= -2 \sum observed \times log(observed/fitted) \\
&= -2(\sum_i y_{ij} log(\frac{y_i}{\hat{\pi}_i}) + \sum_i (1-y_i) log(\frac{1-y_i}{1-\hat{\pi}})) \\
&= -2 \sum_i [y_i log(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}) + log(1-\hat{\pi}_i)] \\
&= -2 \sum_i [y_i(\hat{\beta}_0 + \hat{\beta}_1 x_i) + log(1-\hat{\pi}_i)]
\end{aligned}$$

From 5.14, we could know that $\sum_i y_i = \sum_i \hat{\pi}_i$ and so $\sum_i x_i = \sum_i x_i\hat{\pi}_i$. So the deviance would be:

$$\begin{aligned} D &= -2[\hat{\beta_0}\sum_i \hat{\pi}_i + \hat{\beta_1}\sum_i x_i\hat{\pi}_i + \sum_i log(1-\hat{\pi}_i)] \\ &= -2[\sum_i \hat{\pi}_i(\hat{\beta_0} + \hat{\beta_1}x_i) + \sum_i log(1-\hat{\pi}_i)] \\ &= -2\sum_i \hat{\pi}_i log(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}) - 2\sum_i log(1-\hat{\pi}_i) \end{aligned}$$

Therefore, the deviance only depends on $\hat{\pi}_i$, and it is uninformative for checking model fit.

5.16

(a) If we treat the data as N binomial observations and let $s_i = \sum_{j=1}^{n_i}$, the kernel of the log likelihood is:

$$L(\pi) = \sum_{i=1}^{N} = s_i log(\pi_i) + (n_i - s_i)log(1-\pi_i)$$

If we treat the data as n Bernoulli observations, the kernel of the log likelihood is

$$\begin{aligned} L(\pi) &= \sum_{i=1}^{N}\sum_{j=1}^{n_i} y_{ij} log(\pi_i) + (1-y_{ij})log(1-\pi_i) \\ &= \sum_{i=1}^{N} s_i log(\pi_i) + (n_i - s_i)log(1-\pi_i) \end{aligned}$$

(b) For saturated model, explain why the likelihood function is different for these two data forms.

-