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")
```

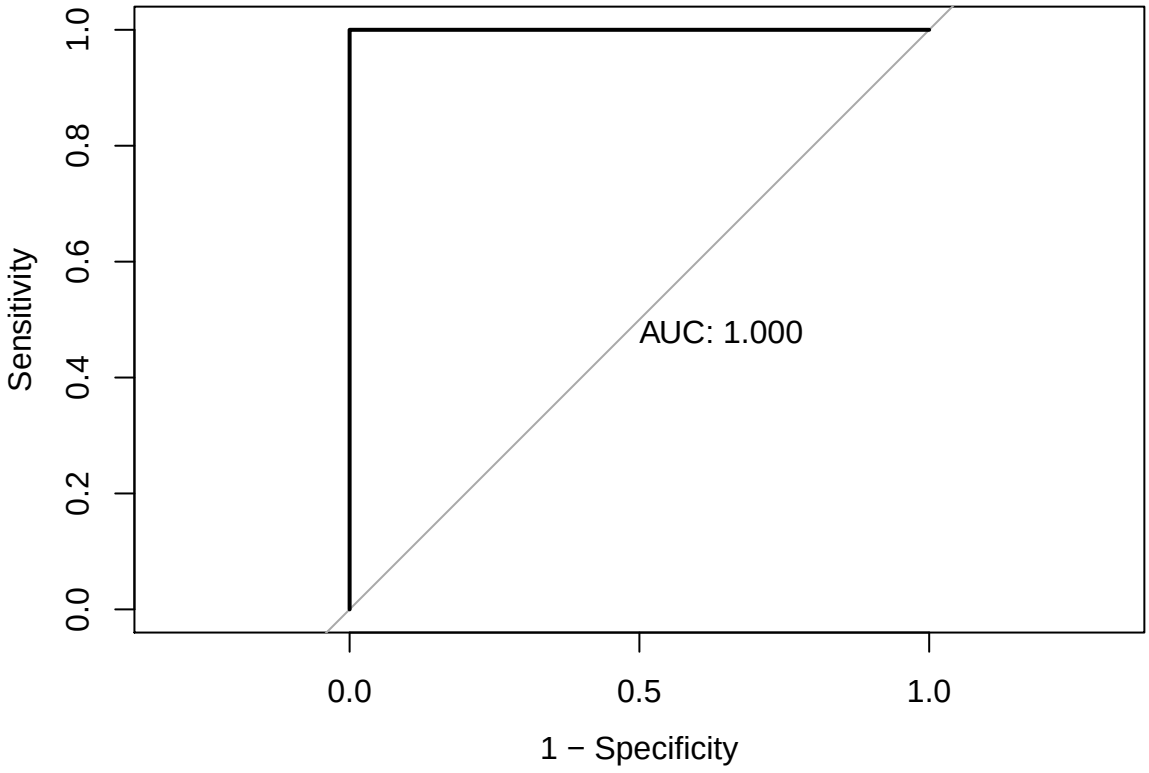
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
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")
```

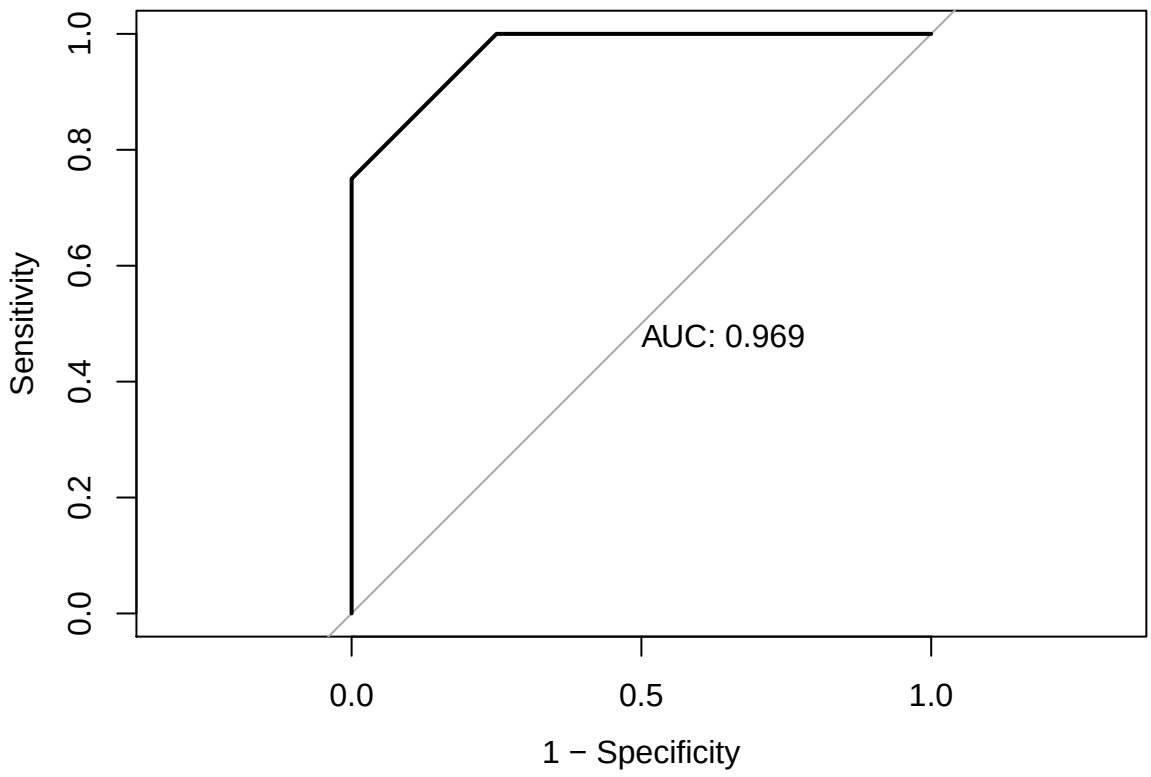
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
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.

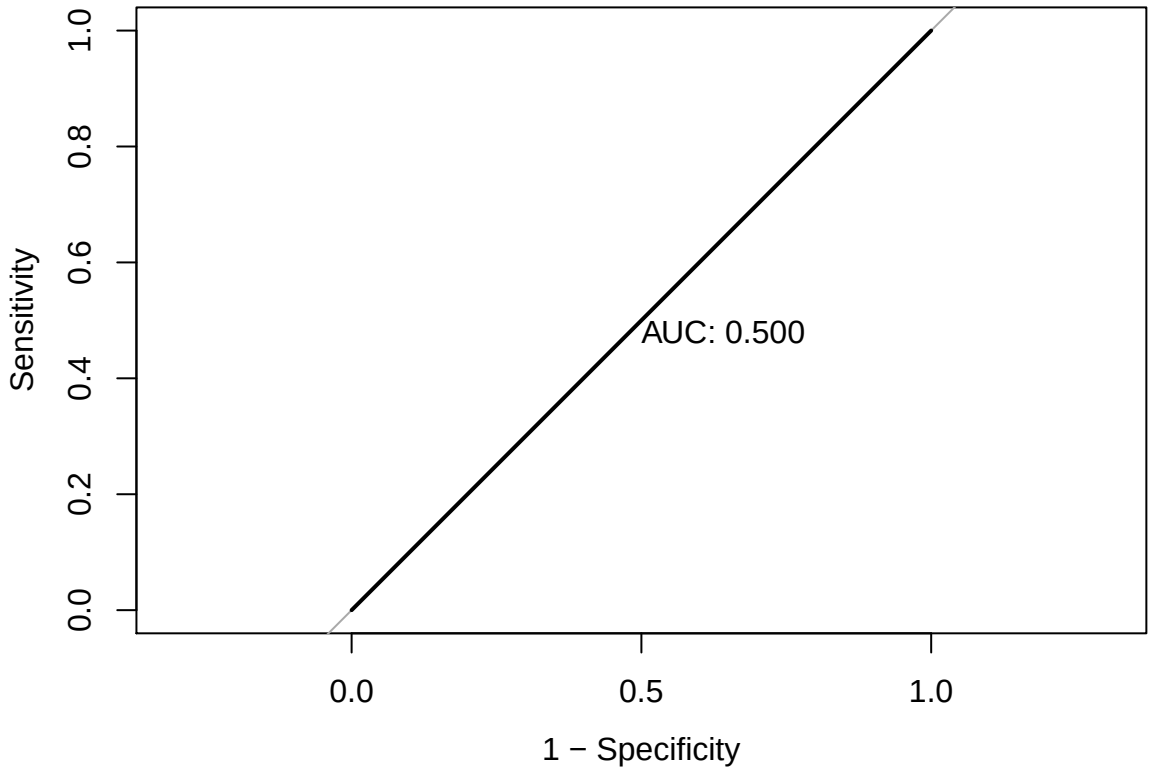
```r
# construct new dataset
x.new = c(1, 1, 2, 2, 3, 3, 4, 4)
y.new = c(1, 0, 1, 0, 1, 0, 1, 0)

# 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

From the example, we know that $\sum_{i=1}^{6} y_i$ is number of 1's and $\sum_{i=1}^{6} x_i y_i$ is the sum of x_is when y_i is 1. Therefore, in this example, we already know that $\sum_i y_i = 3$. Then set $\sum_{i=1}^{6} x_i y_i = t$. The distribution of $\sum_{i=1}^{6} x_i y_i = t$ can be written. t can take values of $6, 7, 8, \cdots, 15$ with probability $\frac{1}{20}, \frac{1}{20}, \frac{2}{20}, \frac{3}{20}, \frac{3}{20}, \frac{3}{20}, \frac{3}{20}, \frac{2}{20}, \frac{1}{20}, \frac{1}{20}$ respectively. Hence, the exact p-value for $P(\sum_{i=1}^{6} x_i y_i \leq 6 | \sum_{i=1}^{6} y_i = 3) = \frac{1}{20} = 0.05$.

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

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

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

- If we treat the data as N Binomial observations, there are N parameters $(\pi_1, \cdots, \pi_N)$.
- If we treat the data as n Bernoulli observations, there are n parameters $(\pi_{11}, \cdots, \pi_{Nn_i})$.

(c) Explain why the difference between deviances for two unsaturated models does not depend on the form of data entry.

They do not depend on the form of data entry because when subtracting, the log likelihood of saturated models cancel out, so the result only depends on the log likelihoods of unsaturated models. We aready know from (a) part that the log likelihood of unsaturated models do not depend on the form of data entry.

5.17

Create a data file in two ways.

```
# Ungrouped data
x.ungroup = c(rep(0, 4), rep(1, 4), rep(2, 4))
y.ungroup = c(1, 0, 0, 0, 1, 1, 0, 0, rep(1, 4))

# Grouped data
x.group = c(0, 1, 2)
n.trials = c(4, 4, 4)
n.successes = c(1, 2, 4)
resp = cbind(n.successes, n.trials - n.successes)

# Fit models
model.ungrouped = glm(y.ungroup ~ x.ungroup, family = "binomial")
summary(model.ungrouped)
```

(a)

```
##
## Call:
## glm(formula = y.ungroup ~ x.ungroup, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4216  -0.6339   0.3752   0.5193   1.8459
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.503      1.181  -1.272   0.2033
## x.ungroup      2.060      1.130   1.823   0.0682 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16.301  on 11  degrees of freedom
## Residual deviance: 11.028  on 10  degrees of freedom
## AIC: 15.028
##
## Number of Fisher Scoring iterations: 4
```

```
model.grouped = glm(resp~x.group, family = "binomial")
summary(model.grouped)
```

```
##
## Call:
## glm(formula = resp ~ x.group, family = "binomial")
##
## Deviance Residuals:
##       1        2        3
##  0.3377  -0.5543   0.7504
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.503      1.181  -1.272   0.2034
## x.group        2.060      1.130   1.823   0.0683 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.2568  on 2  degrees of freedom
## Residual deviance: 0.9844  on 1  degrees of freedom
## AIC: 8.6722
##
## Number of Fisher Scoring iterations: 4
```

From the summaries, we could know that:

- For ungrouped data, the deviance for M_0 us 16.3 and the deviance for M_1 is 11.0.
- For grouped data, the deviance for M_0 us 6.3 and the deviance for M_1 is 1.0.

The saturated model in the ungrouped case has 12 parameters (df = 11) and the saturated model in the grouped case only has three parameters.

(b) The differences between the deviances are the same. 16.3 - 11.0 = 6.3 - 1.0 = 5.3

Explanation: the only difference between log likelihoods of these two data entry forms is the binomial coefficients. However, it cancels out when we do the subtraction. Therefore, the differences of deviances are the same.

5.19

Suppose $\pi_{ab} + \pi_{ba} = 1$ and we are using the model

$$log(\pi_{ab}/\pi_{ba}) = \beta_a - \beta_b.$$

For $a < b$, let N_{ab} denote the number of matches between teams a and b, with team a winning n_{ab} times and team b winning n_{ba} times.

(a) Find the log-likelihood, treating n_{ab} as a binomial variate for N_{ab} trials. Show that sufficient statistics are $\{n_{a+}\}$, so that "victory totals" determine the estimated ranking of teams. Since $\pi_{ab} + \pi_{ba} = 1$, we have $\pi_{ba} = 1 - \pi_{ab}$. Then the model could be re-written as:

$$log(\frac{\pi_{ab}}{1-\pi_{ab}}) = \beta_a - \beta_b$$

The likelihood function would be:

$$\begin{aligned} l &= \binom{N_{ab}}{n_{ab}} \cdot \pi_{ab}^{n_{ab}} \cdot (1-\pi_{ab})^{N_{ab}-n_{ab}} \\ &= \binom{N_{ab}}{n_{ab}} \cdot (1-\pi_{ab})^{N_{ab}} \cdot [\frac{\pi_{ab}}{(1-\pi_{ab})}]^{n_{ab}} \end{aligned}$$

So the first two terms are the function related to n_{ab} and the later two terms are the function related to π_{ab} and it interact with π_{ab} only through n_{ab}.

(b) Generalize the model to allow a "home-team advantage", with a team's chance of winning possibly increasing when it plays at its home city. Interpret parameters.

The new model would be:

$$log(\frac{\pi_{ab}}{1-\pi_{ab}}) = \beta_a - \beta_b + \beta_{hc} \cdot X_{hc}$$

Interpretation: The X_{hc} is the indicator variable that implies whether the city is the home city for this team. If the city is the home city, then the log odds will increase by the value of β_{hc}

5.20

Let $y_i, i = 1, \cdots, N$ denote N independent binary random variables.

(a) Derive the log likelihood for the probit model $\Phi^{-1}[\pi(\mathbf{x_i})] = \sum_j \beta_j x_{ij}$.

The likelihood function is:

$$l(\pi(\mathbf{x_i})) = \Pi_{i=1}^{N} \pi(\mathbf{x_i})^{y_i} (1-\pi(\mathbf{x_i}))^{1-y_i}$$

So the log likelihood can be written as:

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

Then, since we are using probit link function, we have:

$$\pi(\mathbf{x_i}) = \Phi(\sum_j \beta_j x_{ij})$$

Plug in the linkage, we will obtain that:

$$L(\pi(\mathbf{x_i})) = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{y_i log}(\mathbf{\Phi}(\sum_{\mathbf{j}} \beta_{\mathbf{j}} \mathbf{x_{ij}})) + (\mathbf{1}-\mathbf{y_i})\mathbf{log}(\mathbf{1}-\mathbf{\Phi}(\sum_{\mathbf{j}} \beta_{\mathbf{j}} \mathbf{x_{ij}}))$$

(b) Show that the likelihood equations for the logistic and probit regression models are

$$\sum_{i=1}^{N}(y_i - \hat{\pi}_i)z_i x_{ij} = 0, j = 1, \cdots, p,$$

where $z_i = 1$ for the logistic case and $z_i = \phi(\sum_j \hat{\beta}_j x_{ij})/\hat{\pi}_i(1-\hat{\pi}_i)$ for the probit case.

In this case, we know that $\mu_i = \hat{\pi}_i$ and $var(y_i) = \hat{\pi}_i(1-\hat{\pi}_i)$.

The likelihood equations for the logistic and probit regression models are

$$\begin{aligned}\frac{\partial L(\beta)}{\partial \beta_j} &= \sum_{i=1}^{N} \frac{(y_i - \mu_i)x_{ij}}{var(y_i)} \cdot \frac{\partial \mu_i}{\partial \eta_i} \\ &= \sum_{i=1}^{N}(y_i - \hat{\pi}_i) \cdot \phi(\sum_j \hat{\beta}_j x_{ij}) x_{ij}/\hat{\pi}_i(1-\hat{\pi}_i) \\ &= \sum_{i=1}^{N}(y_i - \hat{\pi}_i)z_i x_{ij} = 0\end{aligned}$$

5.30

We have 709 cases and 709 controls. This is a retrospective case-control study. In this study, we can estimate the odds ratio of having lung cancer for cases versus controls. However, we cannot estimate the intercept (the log odds of having lung cancer for controls). The odds ratio would be $\frac{59\times688}{21\times650} = 2.974$

Here we also want to calculate the odds ratio by fitting a model.

```
# Create variables
lung.cancer = c(rep(0, 709), rep(1, 709))
smoking = c(rep(1, 650), rep(0, 709-650), rep(1, 688), rep(0, 709-688))
strata = c(rep(0, 709), rep(1, 709))
data = data.frame(lung.cancer, smoking, strata)

model.clog = glm(lung.cancer~smoking,family = "binomial", data = data)
summary(model.clog)
```

```
##
## Call:
## glm(formula = lung.cancer ~ smoking, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2016  -1.2016   0.1865   1.1534   1.6356
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0330     0.2541  -4.065 4.80e-05 ***
## smoking       1.0898     0.2599   4.193 2.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1965.8  on 1417  degrees of freedom
## Residual deviance: 1945.9  on 1416  degrees of freedom
## AIC: 1949.9
##
## Number of Fisher Scoring iterations: 4
```

The odds ratio calculated by the model is $e^{1.0898} = 2.974$, which is the same as what we have calculated by hand.