Stat 230. HW7. Yifan Zheng. 30352/2865

1. Lecture 18. Q2.

**2 Hessian matrix in the multinomial logit model** Prove that the Hessian matrix of the log-likelihood function of the multinomial logit model is negative semi-definite.

By proposition 1. the covariance matrix:

$$\begin{bmatrix} \pi_1(1-\pi_1) & -\pi_1\pi_2 & \cdots & -\pi_1\pi_{k-1} \\ -\pi_1\pi_2 & \pi_2(1-\pi_2) & \cdots & -\pi_2\pi_{k-1} \\ \vdots & \vdots & & \\ -\pi_1\pi_{k-1} & -\pi_2\pi_{k-1} & \cdots & \pi_{k-1}(1-\pi_{k-1}) \end{bmatrix}$$ is positive semi-definite

Suppose this matrix is $A$

Then Hessian is $-\sum_{i=1}^{n} A_i \otimes (X_i X_i^T)$

Suppose eigenvalues of $A_i$ is $\lambda_i = \begin{pmatrix} \lambda_{i1} \\ \vdots \\ \lambda_{ik-1} \end{pmatrix}$

and eigenvalues of $X_i X_i^T$ is $\mu_i = \begin{pmatrix} \mu_{i1} \\ \vdots \\ \mu_{ik-1} \end{pmatrix}$

Because $A_i$ and $X_i X_i^T$ are positive semi-definite

Hence $\lambda_{ij} \geq 0$, $\mu_{ij'} \geq 0$ $(i=1,\cdots,n,\ j,j'=1,\cdots,k-1)$

Hence $\lambda_{ij} \cdot \mu_{ij'} \geq 0$, which is the eigenvalues of $A_i \otimes X_i X_i^T \geq 0$, that is, $A_i \otimes X_i X_i^T$ is positive semi-definite, Hence $\left[-\sum_{i=1}^{n} A_i \otimes X_i X_i^T\right]$ is negative semi-definite

2. Lecture 18 Q3.

**3 Iteratively reweighted least squares algorithm for the multinomial logit model** Similar to the binary logistic model, Newton's method for computing the MLE for the multinomial logit model can be written as iteratively reweighted least squares. Give the details.

① Start from $\beta^{old}$

② Approximate the score equation as:

$$0 \cong A + B \cdot (\beta - \beta^{old})$$

where $A = \frac{\partial \log L(\beta)}{\partial \beta}$ $\quad B = \frac{\partial^2 \log L(\beta)}{\partial \beta \, \partial \beta^T}$

③ Using matrix form. Suppose $k=1, \cdots, k-1$

Define: $Y_k = \begin{bmatrix} I(y_1 = k) \\ \vdots \\ I(y_n = k) \end{bmatrix}$ $\quad \pi_k^{old} = \begin{bmatrix} \pi_k(x_1, \beta^{old}) \\ \vdots \\ \pi_k(x_n, \beta^{old}) \end{bmatrix}$

$$W_k^{old} = diag\left[\pi_k(x_i, \beta^{old})\{1 - \pi_k(x_i, \beta^{old})\}\right]_{i=1}^{n}$$

$$V_{k,k'}^{old} = diag\left[\pi_k(x_i, \beta^{old})\,\pi_{k'}(x_i, \beta^{old})\right]_{i=1}^{n} \quad \text{where } k \neq k'$$

Hence $A^{old} = \begin{bmatrix} X^T(Y_1 - \pi_1^{old}) \\ \vdots \\ X^T(Y_{k-1} - \pi_{k-1}^{old}) \end{bmatrix}$

$$B^{old} = \begin{bmatrix} -X^T W_1^{old} X, & X^T V_{12}^{old} X, & \cdots & X^T V_{1,k-1}^{old} X \\ X^T V_{2,1}^{old} X, & -X^T W_2^{old} X, & \cdots & X^T V_{2,k-1}^{old} X \\ \vdots & \vdots & & \vdots \\ X^T V_{k-1,1}^{old} X, & X^T V_{k-1,2}^{old} X, & \cdots & -X^T W_{k-1}^{old} X \end{bmatrix}$$

④ Update $\beta^{new} = \beta^{old} + B^{old\,-1} A^{old}$ (Next Page).

Actually, B can be decompesed as:

$$B = -\begin{bmatrix} X^T & 0 & \cdots & 0 \\ 0 & X^T & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & X^T \end{bmatrix} \cdot \begin{bmatrix} W_1 & -V_{12} & \cdots & -V_{1,k-1} \\ -V_{21} & W_2 & \cdots & -V_{2,k-1} \\ \vdots & \vdots & & \\ V_{k-1,1} & -V_{k-1,2} & \cdots & W_{k-1} \end{bmatrix} \cdot \begin{bmatrix} X & 0 & \cdots & 0 \\ 0 & X & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & X \end{bmatrix}$$

$$= -\tilde{X}^T \tilde{W} \tilde{X}$$

A can be decomposed as:

$$A = \begin{bmatrix} X^T & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X^T \end{bmatrix} \cdot \left\{ \begin{bmatrix} Y_1 \\ \vdots \\ Y_{k-1} \end{bmatrix} - \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_{k-1} \end{bmatrix} \right\}$$

$$= \tilde{X}^T \cdot (\tilde{Y} - \tilde{\pi})$$

Hence $\beta^{new} = \beta^{old} + (\tilde{X}^T \tilde{W}^{old} \tilde{X})^{-1} \cdot \tilde{X}^T (\tilde{Y} - \tilde{\pi}^{old})$

$$= (\tilde{X}^T \tilde{W}^{old} \tilde{X})^{-1} \tilde{X}^{-1} \tilde{W}^{old} \tilde{Z}^{old}$$

where $\tilde{Z}^{old} = \tilde{X} \beta^{old} + (\tilde{W}^{old})^{-1} (\tilde{Y} - \tilde{\pi}^{old})$

That is the iteratively reweighted least squares

3. Lecture 19. Q7.

**7 Likelihood for the Zero-inflated Poisson regression** Write down the likelihood function for the Zero-inflated Poisson model, and derive the steps for Newton's method.

$$y_i | x_i \sim \begin{cases} 0 & \text{with prob } p_i \\ \text{poisson}(\lambda_i) & \text{with prob } 1-p_i \end{cases}$$

$$\text{with } p_i = \frac{e^{x_i^T \gamma}}{1+e^{x_i^T \gamma}} \qquad \lambda_i = e^{x_i^T \beta}$$

Hence, $y_i | x_i = 0$, with prob $p_i + (1-p_i) \cdot e^{-\lambda_i}$

$$y_i | x_i = k, \text{ with prob } (1-p_i) \cdot e^{-\lambda_i} \cdot \frac{\lambda_i^k}{k!}, \quad k = 1, 2, \ldots$$

$$L(\gamma, \beta; y) = \prod_{i=1}^{n} \left[ \left( \frac{e^{x_i^T \gamma}}{1+e^{x_i^T \gamma}} + \frac{\exp(-e^{x_i^T \beta})}{1+e^{x_i^T \gamma}} \right)^{1(y_i=0)} \cdot \left( \frac{\exp(-e^{x_i^T \beta}) \cdot e^{x_i^T \beta y_i}}{(1+e^{x_i^T \gamma}) \cdot (y_i!)} \right)^{1(y_i>0)} \right]$$

$$\log L(\gamma, \beta; y) = \sum_{y_i=0} \log\left(e^{x_i^T \gamma} + \exp(-e^{x_i^T \beta})\right) \quad \text{---} ①$$

$$+ \sum_{y_i>0} \left(y_i x_i^T \beta - \exp(x_i^T \beta)\right) \quad \text{---} ②$$

$$- \sum_{i=1}^{n} \log(1+e^{x_i^T \gamma}) \quad \text{---} ③$$

$$- \sum_{y_i>0} \log(y_i!) \quad \text{---} ④$$

① involves $\gamma$, $\beta$. ② involves $\beta$. ③ involves $\gamma$.

Suppose $e^{X_i^T\gamma} = \Gamma_i \quad e^{X_i^T\beta} = B_i$

**FOC**

$$\frac{\partial \log L(\beta,\gamma)}{\partial \beta} = \frac{\partial ①}{\partial \beta} + \frac{\partial ②}{\partial \beta}$$

$$= \sum_{y_i=0} \frac{-X_i \cdot e^{X_i^T\beta} \cdot \exp(-e^{X_i^T\beta})}{e^{X_i^T\gamma} + \exp(-e^{X_i^T\beta})} + \sum_{y_i>0} \left[ y_i X_i - X_i \exp(X_i^T\beta) \right]$$

$$= -\sum_{y_i=0} \frac{X_i B_i}{\Gamma_i \cdot \exp(B_i) + 1} + \sum_{y_i>0} \left[ y_i X_i - X_i B_i \right]$$

$$\frac{\partial \log L(\beta,\gamma)}{\partial \gamma} = \frac{\partial ①}{\partial \gamma} + \frac{\partial ②}{\partial \gamma}$$

$$= \sum_{y_i=0} \frac{e^{X_i^T\gamma} \cdot X_i}{e^{X_i^T\gamma} + \exp(-e^{X_i^T\beta})} - \sum_{i=1}^{n} \frac{e^{X_i^T\gamma} \cdot X_i}{1 + e^{X_i^T\gamma}}$$

$$= \sum_{y_i=0} \frac{\Gamma_i \cdot X_i}{\Gamma_i + \exp(-B_i)} - \sum_{i=1}^{n} \frac{\Gamma_i \cdot X_i}{1 + \Gamma_i}$$

**Hessian**

$$H_{11} = \partial \left( \frac{\partial \log L(\beta,\gamma)}{\partial \beta} \right) / \partial \beta^T$$

$$= -\sum_{y_i=0} \frac{(\Gamma_i \exp(B_i)+1) \cdot X_i \cdot \frac{\partial B_i}{\partial \beta} - X_i B_i \cdot \Gamma_i \exp(B_i) \cdot \frac{\partial B_i}{\partial \beta}}{(\Gamma_i \exp(B_i)+1)^2} - \sum_{y_i>0} X_i \frac{\partial B_i}{\partial \beta}$$

$$= -\sum_{y_i=0} X_i X_i^T \frac{B_i(\Gamma_i \exp(B_i)+1) + B_i^2 \Gamma_i \exp(B_i)}{(\Gamma_i \exp(B_i)+1)^2} - \sum_{y_i>0} X_i X_i^T B_i$$

$$H_{12} = \partial \left( \frac{\partial \log L(\beta,\gamma)}{\partial \beta} \right) / \partial \gamma^T = -\sum_{y_i=0} \frac{\exp(B_i) \cdot X_i B_i \frac{\partial \Gamma_i}{\partial \gamma}}{(\Gamma_i \exp(B_i)+1)^2}$$

$$= -\sum_{y_i=0} X_i X_i^T \frac{\exp(B_i) B_i \Gamma_i}{(\Gamma_i \exp(B_i)+1)^2}$$

$$H_{21} = \partial\left(\frac{\partial \log L(\beta,\gamma)}{\partial \gamma}\right)/\partial \beta^T = \sum_{y_i=0} \frac{-\pi_i \cdot \exp(-B_i)\cdot \frac{\partial B_i}{\partial \beta}}{[\pi_i + \exp(-B_i)]^2}$$

$$= \sum_{y_i=0} \frac{-\pi_i B_i \exp(-B_i)}{[\pi_i + \exp(-B_i)]^2}$$

$$H_{22} = \partial\left(\frac{\partial \log L(\beta,\gamma)}{\partial \gamma}\right)/\partial \gamma^T$$

$$= \sum_{y_i=0} \frac{x_i \frac{\partial \pi_i}{\partial \gamma}\cdot \exp(-B_i)}{(\pi_i + \exp(-B_i))^2} - \sum_{i=1}^{n} \frac{x_i \frac{\partial \pi_i}{\partial \gamma}}{(1+\pi_i)^2}$$

$$= \sum_{y_i=0} x_i x_i^T \cdot \frac{\pi_i \exp(-B_i)}{(\pi_i + \exp(-B_i))^2} - \sum_{i=1}^{n} x_i x_i^T \frac{\pi_i}{(1+\pi_i)^2}$$

Let $\begin{pmatrix} \frac{\partial \log L(\beta,\gamma)}{\partial \beta} \\ \frac{\partial \log L(\beta,\gamma)}{\partial \gamma} \end{pmatrix} = S$ $\begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} = H$

Hence $\begin{pmatrix} \beta^{new} \\ \gamma^{new} \end{pmatrix} = \begin{pmatrix} \beta^{old} \\ \gamma^{old} \end{pmatrix} - (H^{old})^{-1} \cdot S^{old}$

# 5. Lecture 20 Q1.

**1 MLE in GLMs with binary regressors** The MLEs of Models (1)–(3) does not have explicit formulas in general. But in the special case with $x_i$ containing an intercept and a binary covariate, the MLEs do have simple formulas. Find them in terms of sample means of the outcomes. Does the MLE of Model (4) have explicit formulas with $x_i$ containing an intercept and a binary covariate? If so, find it; if not, propose an estimator with explicit formula.

Suppose $x_i = (1, z_i)$ where $z_i$ is 0 or 1.

$\sum_{i=1}^n 1(z_i=0) = n_0$, $\sum_{i=1}^n 1(z_i=1) = n_1$.

$\frac{\sum_{i=1}^n y_i \cdot 1(z_i=0)}{n_0} = \bar{y}_0$, $\frac{\sum_{i=1}^n y_i \cdot 1(z_i=1)}{n_1} = \bar{y}_1$

**Model 1** $y_i | x_i \sim N(\mu_i, \sigma^2)$ $\mu_i = x_i^T \beta = (1, z_i) \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$

$S(\beta_0) = \frac{\partial \log \prod_{i=1}^n f(y_i|x_i; \mu_i, \sigma^2)}{\partial \beta_0} = \sum_{i=1}^n \left( \frac{y_i}{\sigma^2} - \frac{\beta_0}{\sigma^2} - \frac{z_i \beta_1}{\sigma^2} \right) = 0$

Check $\frac{\partial S(\beta_1)}{\partial \beta_1} = -\frac{1}{\sigma^2} < 0$.

Hence $\bar{y} - \hat{\beta}_0^{MLE} - \bar{z} \hat{\beta}_1^{MLE} = 0 \Rightarrow \bar{y} - \hat{\beta}_0^{MLE} - \frac{n_1}{n} \hat{\beta}_1^{MLE} = 0$ —— (1)

$S(\beta_1) = \frac{\partial \log \prod_{i=1}^n f(y_i|x_i; \mu_i, \sigma^2)}{\partial \beta_1} = \sum_{i=1}^n \left( \frac{y_i z_i}{\sigma^2} - \frac{z_i \beta_0}{\sigma^2} - \frac{z_i^2 \cdot \beta_1}{\sigma^2} \right) = 0$

Check $\frac{\partial S(\beta_1)}{\partial \beta_1} = -\frac{\sum_{i=1}^n z_i^2}{\sigma^2} < 0$

Hence $\bar{y}_1 \cdot \frac{n_1}{n} - \frac{n_1}{n} \hat{\beta}_0^{MLE} - \frac{n_1}{n} \hat{\beta}_1^{MLE} = 0$ —— (2)

Combining (1) & (2). $\hat{\beta}_1^{MLE} = (\bar{y}_1 - \bar{y}) \cdot \frac{n}{n_0} = \bar{y}_1 - \bar{y}_0$

$\hat{\beta}_0^{MLE} = \bar{y}_1 - \frac{n}{n_0} \bar{y}_1 + \frac{n}{n_0} \bar{y} = -\frac{n_1}{n_0} \bar{y}_1 + \frac{n}{n_0} \bar{y} = \bar{y}_0$.

**Model 2** $y_i | x_i \sim \text{Ber}(\mu_i)$, with $\mu_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$

$$\log \prod_{i=1}^n f(y_i|x_i;\mu_i) = \sum_{i=1}^n \left(y_i \log \frac{\mu_i}{1-\mu_i} - \log \frac{1}{1-\mu_i}\right)$$

$$= \sum_{i=1}^n \left(y_i \log e^{x_i^T\beta} - \log\left(1+e^{x_i^T\beta}\right)\right)$$

$$= \sum_{i=1}^n \left(y_i(\beta_0+\beta_1 z_i) - \log\left(1+e^{\beta_0+\beta_1 z_i}\right)\right)$$

$$S(\beta_0) = \sum_{i=1}^n \left(y_i - \frac{e^{\beta_0+\beta_1 z_i}}{1+e^{\beta_0+\beta_1 z_i}}\right) = 0 \quad \text{——— (1)}$$

Check $\frac{\partial S(\beta_0)}{\partial \beta_0} = \sum_{i=1}^n - \frac{e^{2(\beta_0+\beta_1 z_i)}}{(1+e^{\beta_0+\beta_1 z_i})^2} < 0$

$$S(\beta_1) = \sum_{i=1}^n \left(y_i - \frac{e^{\beta_0+\beta_1 z_i}}{1+e^{\beta_0+\beta_1 z_i}}\right) \cdot z_i = 0 \quad \text{——— (2)}$$

Check $\frac{\partial S(\beta_1)}{\partial \beta_1} = \sum_{i=1}^n - z_i^2 \cdot \frac{e^{2(\beta_0+\beta_1 z_i)}}{(1+e^{\beta_0+\beta_1 z_i})^2} < 0$

Hence Combining (1) & (2)

$$\begin{cases} \left(\bar{y}_0 - \frac{e^{\hat\beta_0}}{1+e^{\hat\beta_0}}\right) \cdot n_0 + \left(\bar{y}_1 - \frac{e^{\hat\beta_0+\hat\beta_1}}{1+e^{\hat\beta_0+\hat\beta_1}}\right) n_1 = 0 \\ \bar{y}_1 - \frac{e^{\hat\beta_0+\hat\beta_1}}{1+e^{\hat\beta_0+\hat\beta_1}} = 0 \end{cases}$$

$$\Rightarrow \hat\beta_0^{MLE} = \log \frac{\bar{y}_0}{1-\bar{y}_0} \qquad \hat\beta_1^{MLE} = \log \frac{\bar{y}_1}{1-\bar{y}_1} - \log \frac{\bar{y}_0}{1-\bar{y}_0}$$

Model 3 $\quad y_i \mid x_i \sim \text{Poi}(\mu_i)$ with $\mu_i = e^{x_i^T \beta}$

$$\log f(y_i \mid x_i; \mu_i) = y_i \log \mu_i - \mu_i - \log y_i!$$

$$= y_i(\beta_0 + \beta_1 z_i) - e^{\beta_0 + \beta_1 z_i} - \log y_i!$$

$$S(\beta_0) = \sum_{i=1}^n (y_i - e^{\beta_0 + \beta_1 z_i}) = 0 \quad \text{——— (1)}$$

Check $\frac{\partial S(\beta_0)}{\partial \beta_0} = \sum_{i=1}^n -e^{\beta_0 + \beta_1 z_i} < 0$.

$$S(\beta_1) = \sum_{i=1}^n (y_i - e^{\beta_0 + \beta_1 z_i}) z_i = 0 \quad \text{——— (2)}$$

Check $\frac{\partial S(\beta_1)}{\partial \beta_1} = \sum_{i=1}^n z_i^2 (-e^{\beta_0 + \beta_1 z_i}) < 0$

Hence combining (1) & (2).

$$\begin{cases} (\bar{y}_0 - e^{\beta_0}) \cdot n_0 + (\bar{y}_1 - e^{\beta_1 + \beta_1}) \cdot n_1 = 0 \\ \bar{y}_1 - e^{\beta_0 + \beta_1} = 0 \end{cases}$$

$$\Rightarrow \hat{\beta}_0^{MLE} = \log \bar{y}_0 \qquad \hat{\beta}_1^{MLE} = \log \bar{y}_1 - \log \bar{y}_0$$

**Model 4** $y_i|x_i \sim NB(\mu_i, \delta)$ with $\mu_i = e^{x_i^T\beta}$

$$\log f(y_i|x_i;\mu_i) = y_i \log\frac{\mu_i}{\mu_i+\delta} - \delta\log(\mu_i+\delta) + c(y_i,\delta)$$

$$= y_i\left[(x_i^T\beta - \log(e^{x_i^T\beta}+\delta)\right] - \delta\log(e^{x_i^T\beta}+\delta)) + c(y_i,\delta)$$

$$= y_i\left[(\beta_0+\beta_1 z_i - \log(e^{\beta_0+\beta_1 z_i}+\delta)\right] - \delta\log(e^{\beta_0+\beta_1 z_i}+\delta) + c(y_i,\delta)$$

$$S(\beta_0) = \sum_{i=1}^n \left(y_i - \frac{y_i e^{\beta_0+\beta_1 z_i}}{e^{\beta_0+\beta_1 z_i}+\delta} - \frac{\delta e^{\beta_0+\beta_1 z_i}}{e^{\beta_0+\beta_1 z_i}+\delta}\right) = 0. \quad \text{—— (1)}$$

Check $\frac{\partial S(\beta_0)}{\partial \beta_0} < 0$

$$S(\beta_1) = \sum_{i=1}^n \left(y_i - \frac{y_i e^{\beta_0+\beta_1 z_i}}{e^{\beta_0+\beta_1 z_i}+\delta} - \frac{\delta e^{\beta_0+\beta_1 z_i}}{e^{\beta_0+\beta_1 z_i}+\delta}\right) z_i = 0 \quad \text{—— (2)}$$

Check $\frac{\partial S(\beta_1)}{\partial \beta_1} < 0$

Hence Combing (1) & (2).

$$\begin{cases} \bar{y}_0 - \dfrac{\bar{y}_0 e^{\beta_0}}{e^{\beta_0}+\delta} - \dfrac{\delta e^{\beta_0}}{e^{\beta_0}+\delta} = 0 \\ \bar{y}_1 - \dfrac{\bar{y}_1 e^{\beta_0+\beta_1}}{e^{\beta_0+\beta_1}+\delta} - \dfrac{\delta\cdot e^{\beta_0+\beta_1}}{e^{\beta_0+\beta_1}+\delta} = 0 \end{cases}$$

$$\Rightarrow \hat{\beta}_0^{MLE} = \log \bar{y}_0 \quad , \quad \hat{\beta}_1^{MLE} = \log \bar{y}_1 - \log \bar{y}_0$$

T. Lecture 21. Q2.

**2 Sandwich asymptotic covariance matrix for GEE** Verify the formulas of $B$ and $M$ in Section 3.2.



By Taylor Expension: $\mu(X_i,\beta) \approx \mu(X_i,\hat{\beta}) + \frac{\partial \mu(X_i,\hat{\beta})}{\beta}(\beta-\hat{\beta})$

FOC: $\sum_{i=1}^n \frac{\partial \mu^T(X_i,\hat{\beta})}{\partial \hat{\beta}} \tilde{V}^{-1}(X_i,\hat{\beta})(Y-\mu(X_i,\hat{\beta}))=0$

$\Rightarrow \sum_{i=1}^n \frac{\partial \mu^T(X_i,\hat{\beta})}{\partial \hat{\beta}} \tilde{V}^{-1}(X_i,\hat{\beta})\left(Y-\mu(X_i,\beta)-\frac{\partial \mu(X_i,\hat{\beta})}{\partial \hat{\beta}^T}(\hat{\beta}-\beta)\right)\approx 0$

$\Rightarrow \sum_{i=1}^n D_i^T(\hat{\beta})\tilde{V}^{-1}(X_i,\hat{\beta})(Y-\mu(X_i,\beta)-D_i(\hat{\beta})(\hat{\beta}-\beta))\approx 0$

$\Rightarrow \sqrt{n}(\hat{\beta}-\beta) \approx -(\sum_{i=1}^n D_i^T(\hat{\beta})\tilde{V}^{-1}(X_i,\hat{\beta})D_i(\hat{\beta}))^{-1}\cdot \frac{1}{\sqrt{n}}\sum_{i=1}^n D_i^T(\hat{\beta})\tilde{V}^{-1}(X_i,\hat{\beta})\cdot(Y_i-\mu(X_i,\beta))$

Because $\frac{1}{\sqrt{n}}\sum_{i=1}^n D_i^T(\hat{\beta})\cdot\tilde{V}^{-1}(X_i,\hat{\beta})(Y-\mu(X_i,\hat{\beta})) \xrightarrow{d} N(0,M)$

where $M = E\left(\frac{1}{n}\sum_{i=1}^n D_i^T(\beta)\tilde{V}^{-1}(X_i,\beta)V^{-1}(X_i)\tilde{V}^{-1}(X_i,\beta)D_i(\beta)\right)$

And Let $B = E\left(\frac{1}{n}\sum_{i=1}^n D_i^T(\beta)\tilde{V}^{-1}(X_i,\beta)D_i^T(\beta)\right)$

Hence $\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{d} N(0, B^{-1}MB^{-1})$

8. Lecture 21. Q3.

**3 Cluster-robust standard error in OLS with a constant binary regressor** Consider a special case with $x_{it} = (1, x_i)$ and $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$, and view "1" as treatment and "0" as control. Show ① that the coefficient of $x_i$ in the pooled OLS fit of $y_{it}$ on $x_{it} = (1, x_i)$ equals $\hat{\tau} = \bar{y}_1 - \bar{y}_0$ where

$$\bar{y}_1 = \sum_{i=1}^{n}\sum_{t=1}^{n_i} x_i y_{it}/N_1, \quad \bar{y}_0 = \sum_{i=1}^{n}\sum_{t=1}^{n_i}(1-x_i)y_{it}/N_0,$$

with $N_1 = \sum_{i=1}^{n} n_i x_i$ and $N_0 = \sum_{i=1}^{n} n_i(1-x_i)$ denoting the total number of observations under treatment and control, respectively. ② Show further that the cluster-robust standard error of $\hat{\tau}$ equals the square root of

$$\sum_{i=1}^{n} x_i R_i^2/N_1^2 + \sum_{i=1}^{n}(1-x_i)R_i^2/N_0^2,$$

where

$$R_i = \begin{cases} \sum_{t=1}^{n_i}(y_{it} - \bar{y}_1), & \text{if } x_i = 1, \\ \sum_{t=1}^{n_i}(y_{it} - \bar{y}_0), & \text{if } x_i = 0. \end{cases}$$

① $y_{it} = \alpha + \tau \cdot x_i + \varepsilon_{it}$ $\quad i = 1, \cdots n$ $\quad t = 1, \cdots, n_i$

By OLS, $\hat{\tau} = \arg\min_{\tau} \sum_{i=1}^{n}\sum_{t=1}^{n_i} (y_{it} - (\alpha + \tau x_i))^2$

FOC: $\sum_{i=1}^{n}\sum_{t=1}^{n_i} (y_{it} - \alpha - \tau x_i) = 0$

$\sum_{i=1}^{n}\sum_{t=1}^{n_i} (y_{it} - \alpha - \tau x_i) x_i = 0$

$\Rightarrow$ For all $i$ s.t $x_i = 1$. Suppose $i_1, i_2, \cdots i_{N_1}$

$\sum_{i=i_1}^{i_{N_1}}\sum_{t=1}^{n_i} (y_{it} \cdot x_i - \alpha - \tau) = 0$

For all $j$ s.t $x_j = 0$. Suppose $j_1, j_2, \cdots j_{N_0}$

$\sum_{j=j_1}^{j_{N_0}}\sum_{t=1}^{n_j} (y_{jt}(1 - x_i) - \alpha) = 0$

$\Rightarrow$ $N_1 \cdot \bar{y}_1 - N_1 \cdot (\alpha + \tau) = 0$, $N_0 \cdot \bar{y}_0 - N_0 \alpha = 0$

$\Rightarrow$ $\hat{\alpha} = \bar{y}_0$ $\quad \hat{\tau} = \bar{y}_1 - \bar{y}_0$

② $\mathrm{Var}(\hat{\tau}) = \mathrm{Var}(\bar{y}_1) + \mathrm{Var}(\bar{y}_0) - 2\,\mathrm{Cov}(\bar{y}_1, \bar{y}_0)$

$\mathrm{Cov}(\bar{y}_1, \bar{y}_0) = 0$ because there is no same $X_i$ in $\bar{y}_1$ and $\bar{y}_0$

in $\bar{y}_1$, only contains i s.t. $X_i = 1$.

$$\mathrm{Var}(\bar{y}_1) = \sum_{i=i_1}^{iN_1} \left( \frac{\left(\sum_{t=1}^{n_i} (y_{it} - \bar{y}_1)\right)^2}{N_1^2} \right) = \sum_{i=i_1}^{iN_1} X_i \left( \frac{\left(\sum_{t=1}^{n_i} (y_{it} - \bar{y}_1)\right)^2}{N_1^2} \right)$$

$$= \sum_{i=1}^{n} X_i \cdot \left( \frac{\left(\sum_{t=1}^{n_i} (y_{it} - \bar{y}_1)\right)^2}{N_1^2} \right)$$

in $\bar{y}_0$, only contains j s.t $X_j = 0$

$$\mathrm{Var}(\bar{y}_0) = \sum_{j=j_1}^{jN_0} \left( \frac{\left(\sum_{t=1}^{n_j} (y_{jt} - \bar{y}_0)\right)^2}{N_0^2} \right) = \sum_{j=j_1}^{jN_0} (1 - X_j) \left( \frac{\left(\sum_{t=1}^{n_j} (y_{jt} - \bar{y}_0)\right)^2}{N_0^2} \right)$$

$$= \sum_{j=1}^{n} (1 - X_j) \left( \frac{\left(\sum_{t=1}^{n_j} (y_{jt} - \bar{y}_0)\right)^2}{N_0^2} \right)$$

Suppose $R_i = \begin{cases} \sum_{i=1}^{n_i} (y_{it} - \bar{y}_1) & \text{if } X_i = 1 \\ \sum_{i=1}^{n_i} (y_{it} - \bar{y}_0) & \text{if } X_i = 0 \end{cases}$

Hence $\mathrm{Var}(\tau^2) = \sum_{i=1}^{n} X_i R_i^2 / N_1^2 + \sum_{i=1}^{n} (1 - X_i) R_i^2 / N_0^2$

# HW7-4

Yifan Zheng

4/23/2020

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(sandwich)
library(MASS)
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

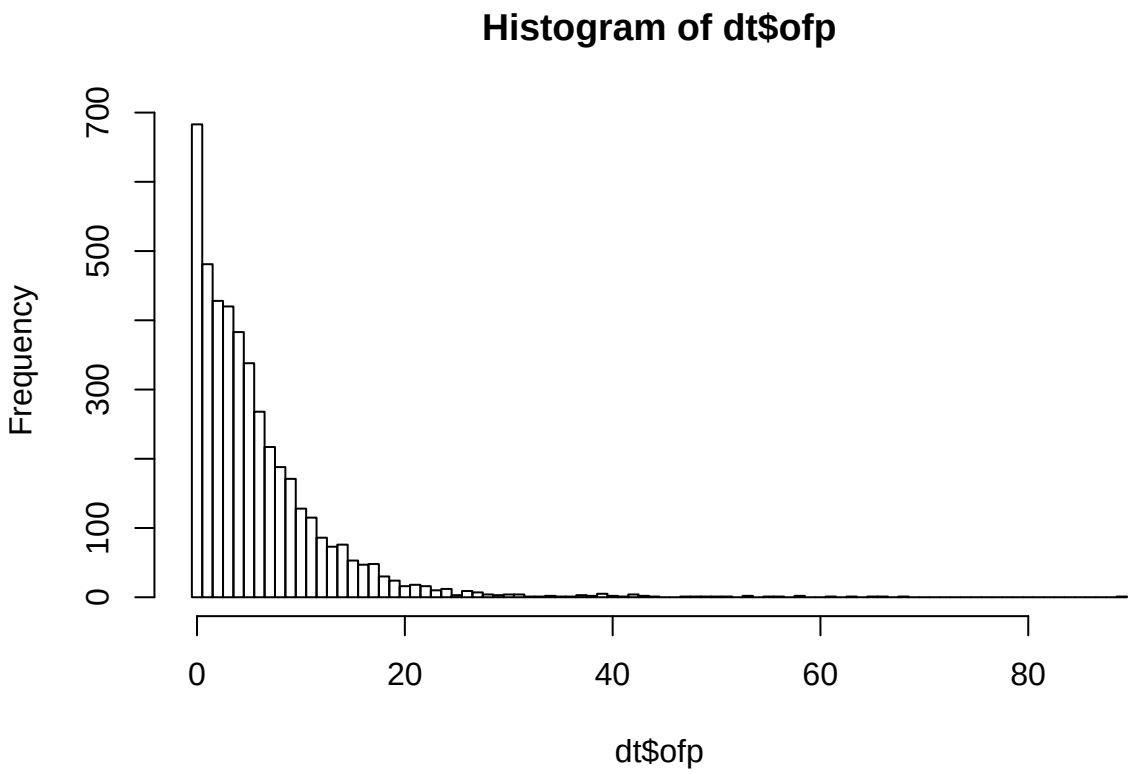
```
library(nnet)
```

## Lecture 19, Problem 10, Data Analysis

Zeileis et al. (2008) give a tutorial on count outcome regressions using the dataset from Deb and Trivedi (1997). Replicate and extend their analysis based on the discussion in this chapter.

```
## get data
load("DebTrivedi.rda")
dt <- DebTrivedi[,c(1,6:8,13,15,18)]
```

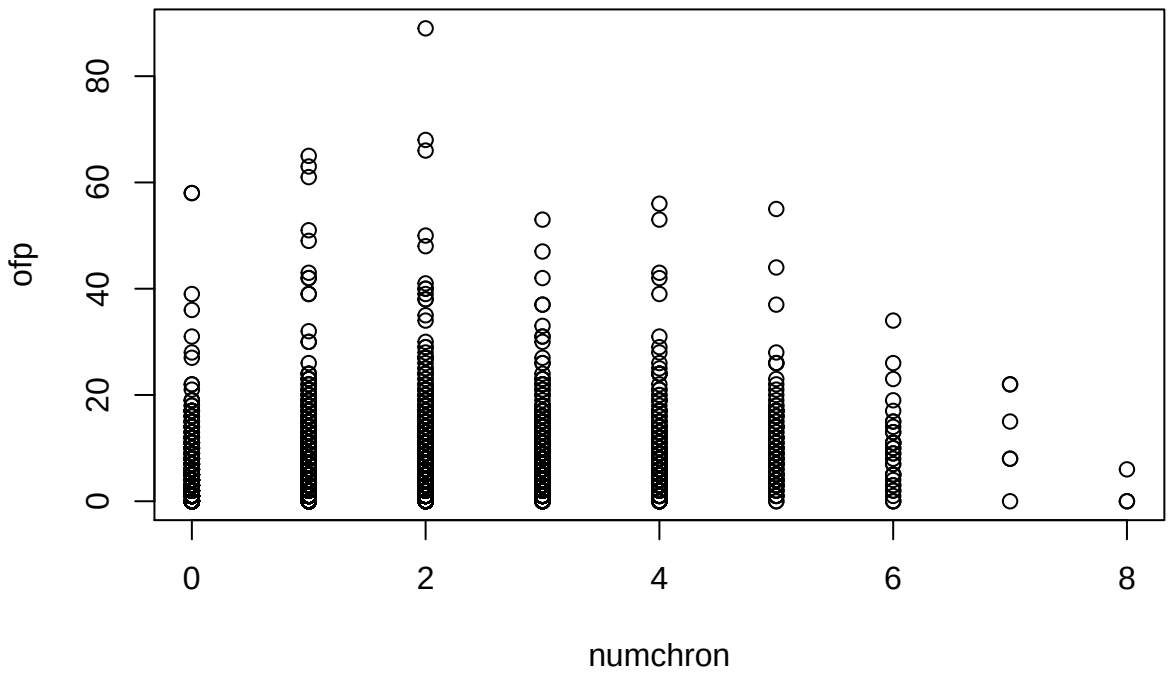
```
## plot overview
hist(dt$ofp, breaks = 0:90 - 0.5)
```




The histogram illustrates that the marginal distribution exhibits both substantial variation and a rather larger number of zeros

```
## look at pairwise bivariate
## displays of the dependent variable against each of the regressors
## bringing out the partial relationships.

plot(ofp~numchron, data = dt)
```




This is not useful as both variables are count variables producing numerous ties in the bivariate distribution and thus obscruring a large number of points in the display.

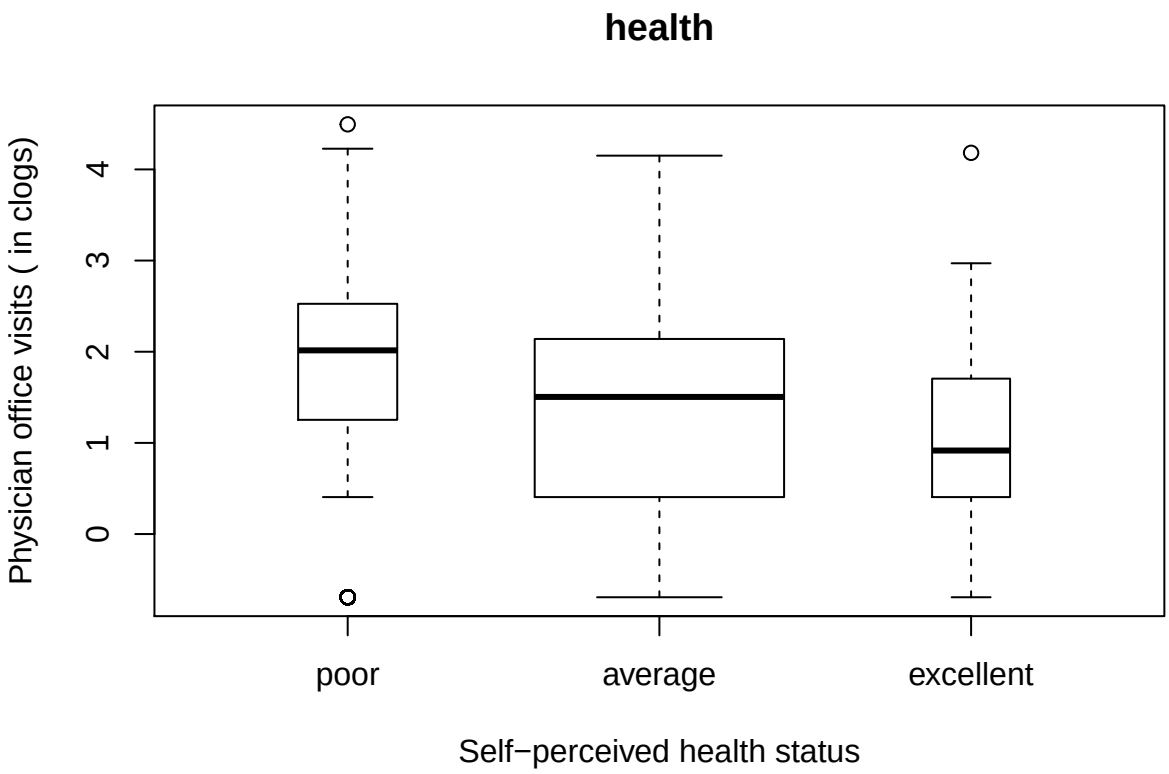
To overcome the problem, it is useful to group the numer of chronic conditions into a factor with levels 0,1,2, 3 or more and produce a boxplot instead of a scatter plot.

Furthermore, the picture is much clearer if the dependent variable is log-transformed.

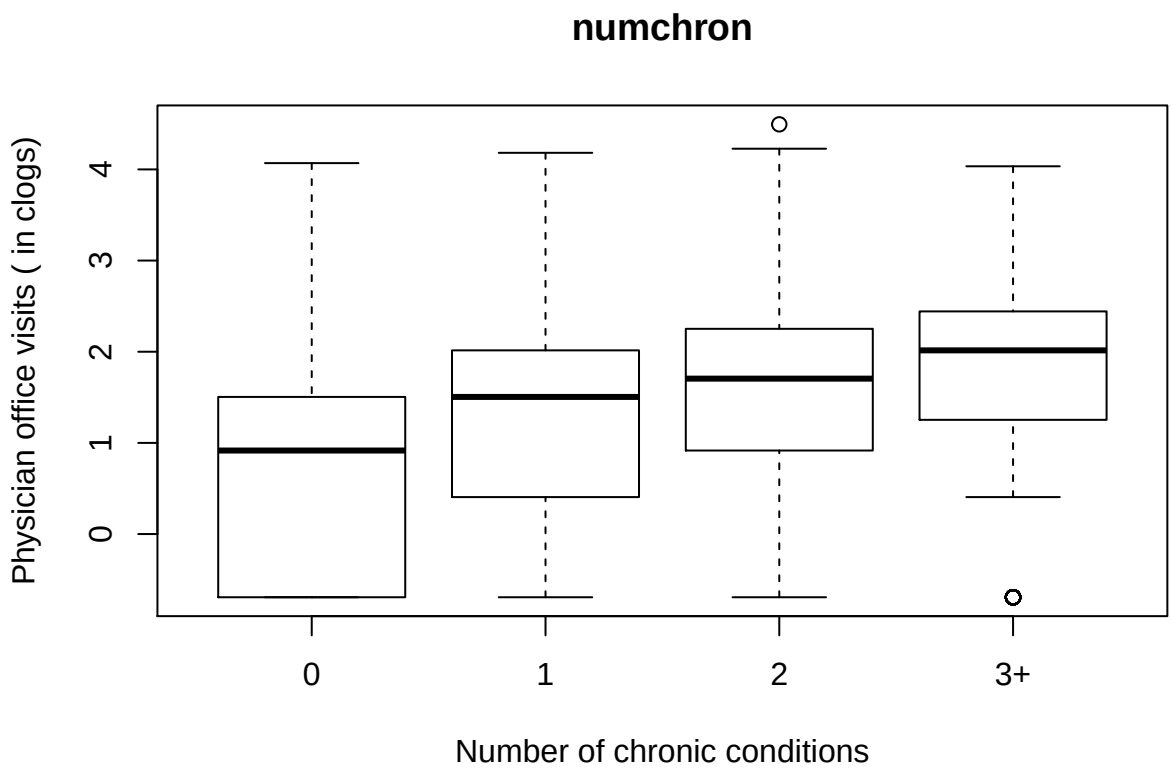
```
## function for log-transform
clog <- function(x){log(x+0.5)}

## function change categories as 0,1,2,3+
cfac <- function(x, breaks = NULL){
  if (is.null(breaks)) {
    breaks <- unique(quantile(x, 0:10/10))}
  x <- cut(x, breaks, include.lowest = TRUE, right = FALSE)
  levels(x) <- paste(breaks[-length((breaks))], ifelse(diff(breaks)>1, c(paste("-", breaks[-c(1, length
  return(x)
}
```

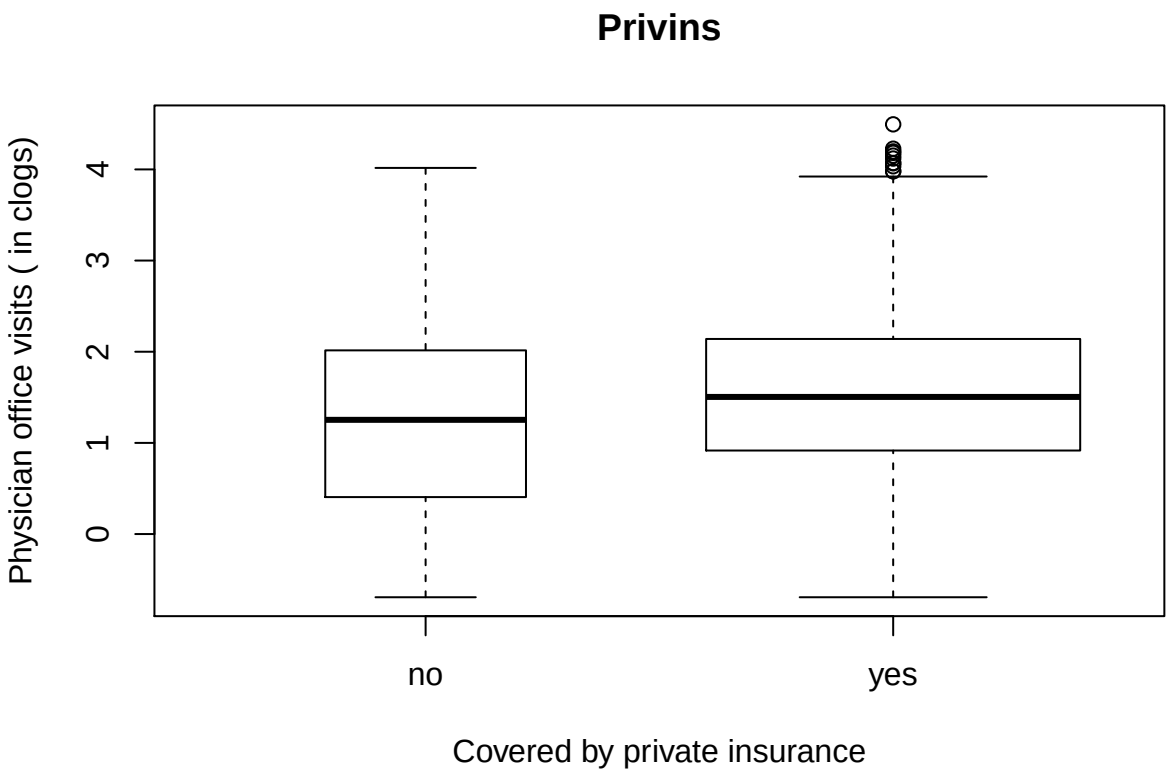
```
## displays for the number of physician office visits against health status
plot(clog(ofp) ~ factor(health, levels = c("poor", "average","excellent")), data = dt, varwidth = TRUE,
     main = "health",xlab = "Self-perceived health status",
     ylab = "Physician office visits ( in clogs)")
```




```
## displays for the number of physician office visits against number of chronic conditions
plot(clog(ofp) ~ cfac(numchron), data = dt,
     main = "numchron", xlab = "Number of chronic conditions",
     ylab = "Physician office visits ( in clogs)")
```

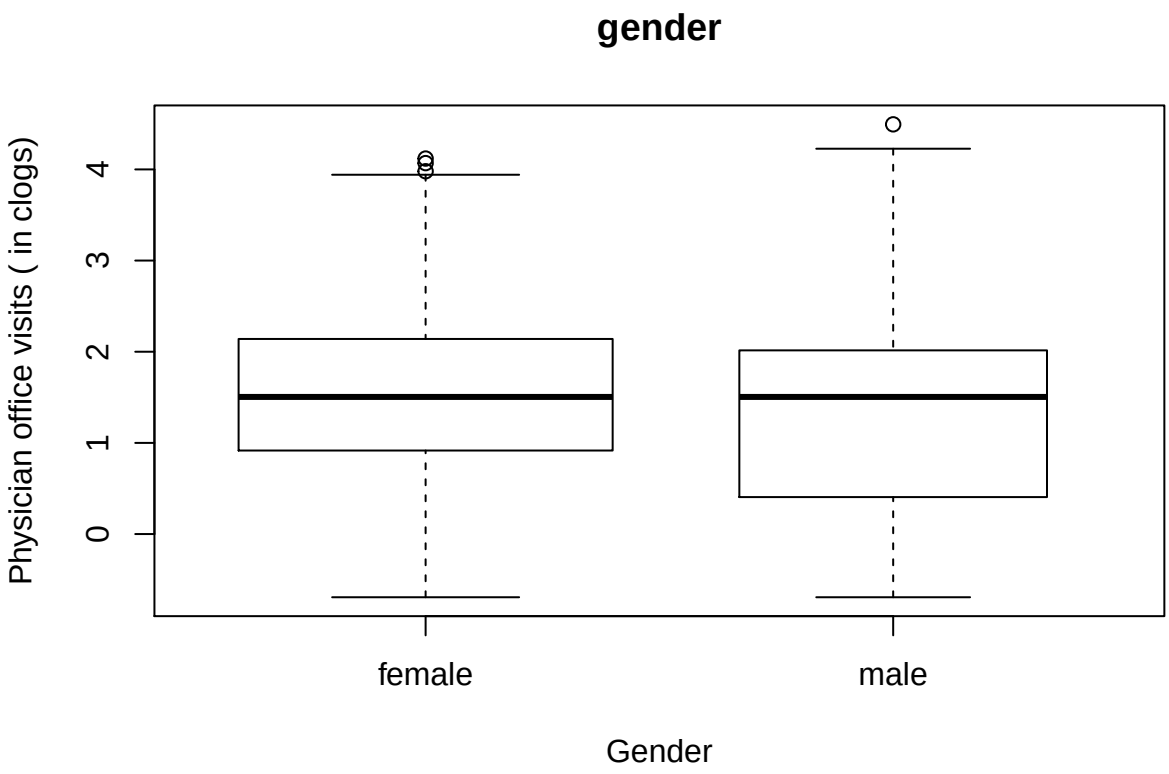



```
## displays for the number of physician office visits against whether covered by private insurance
plot(clog(ofp) ~ privins, data = dt, varwidth = TRUE,
     main = "Privins", xlab = "Covered by private insurance",
     ylab = "Physician office visits ( in clogs)")
```

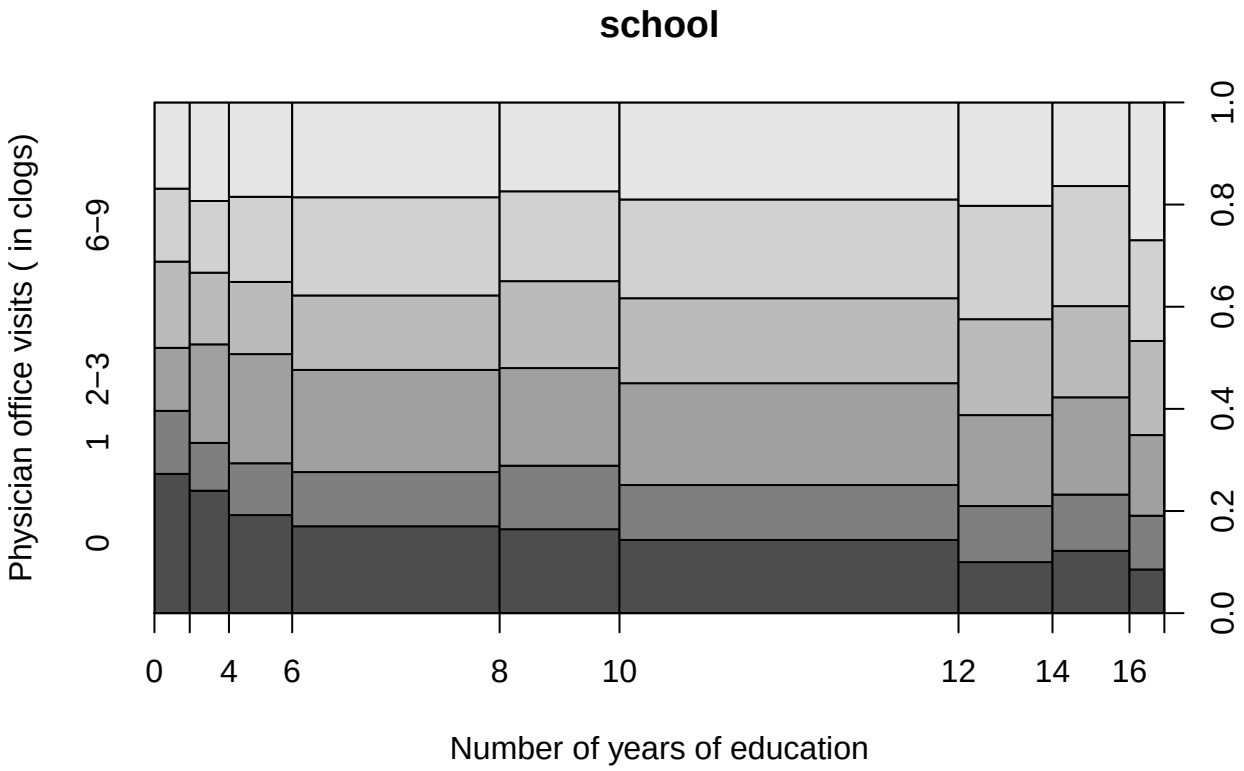



```
## displays for the number of physician office visits against number of hospital stays
plot(clog(ofp) ~ cfac(hosp, c(0:2, 8)), data = dt,
     main = "hosp", xlab = "Number of hospital stays",
     ylab = "Physician office visits ( in clogs)")
```


```
## displays for the number of physician office visits against gender
plot(clog(ofp) ~ gender, data = dt, varwidth = TRUE,
     main = "gender", xlab = "Gender",
     ylab = "Physician office visits ( in clogs)")
```




```
## displays for the number of physician office visits against number of years of education
plot(cfac(ofp, c(0:2, 4, 6, 10, 100)) ~ school, data = dt, breaks = 9,
      main = "school", xlab = "Number of years of education",
     ylab = "Physician office visits ( in clogs)")
```




All displays show that the number of doctor visits increases or decreases with the regressors as expected:

ofp decreases with the general health status but increases with the number of chronic conditions or hospital stays.

The median number of visits is also slightly higher for patients with a private insurance and higher level of education. It is slightly lower for male compared to female patients. The overall impression from all displays is that the changes in the mean can only explain a modest amount of variation in the data.

**Poisson regression**

```
## fit the basic Poisson regression model
fm_pois <- glm(ofp~., data=dt, family = poisson)
## obtain the coefficient estimates along with associated partial Wald tests
summary(fm_pois)
```

```
##
## Call:
## glm(formula = ofp ~ ., family = poisson, data = dt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4055  -1.9962  -0.6737   0.7049  16.3620
##
## Coefficients:
```

```
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.028874   0.023785  43.258   <2e-16 ***
## hosp             0.164797   0.005997  27.478   <2e-16 ***
## healthexcellent -0.361993   0.030304 -11.945   <2e-16 ***
## healthpoor       0.248307   0.017845  13.915   <2e-16 ***
## numchron         0.146639   0.004580  32.020   <2e-16 ***
## gendermale      -0.112320   0.012945  -8.677   <2e-16 ***
## school           0.026143   0.001843  14.182   <2e-16 ***
## privinsyes       0.201687   0.016860  11.963   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 26943  on 4405  degrees of freedom
## Residual deviance: 23168  on 4398  degrees of freedom
## AIC: 35959
##
## Number of Fisher Scoring iterations: 5
```

All coecients are highly signicant with the health variables leading to somewhat larger Wald statistics compared to the socio-economic variables.

However, the Wald test results might be too optimistic due to a misspecication of the likelihood. As the exploratory analysis suggested that over-dispersion is present in this data set, we re-compute the Wald tests using sandwich standard errors.

```
coeftest(fm_pois, vcov = sandwich)
```

```
##
## z test of coefficients:
##
##                  Estimate Std. Error z value  Pr(>|z|)
## (Intercept)      1.028874   0.064530 15.9442 < 2.2e-16 ***
## hosp             0.164797   0.021945  7.5095 5.935e-14 ***
## healthexcellent -0.361993   0.077449 -4.6740 2.954e-06 ***
## healthpoor       0.248307   0.054022  4.5964 4.298e-06 ***
## numchron         0.146639   0.012908 11.3605 < 2.2e-16 ***
## gendermale      -0.112320   0.035343 -3.1780  0.001483 **
## school           0.026143   0.005084  5.1422 2.715e-07 ***
## privinsyes       0.201687   0.043128  4.6765 2.919e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All regressors are still signicant but the standard errors seem to be more appropriate.

### Quasi-Possion regression

```
## quasi-possion model
fm_qpois <- glm(ofp~., data=dt, family = quasipoisson)
summary(fm_qpois)
```

```
##
## Call:
## glm(formula = ofp ~ ., family = quasipoisson, data = dt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4055  -1.9962  -0.6737   0.7049  16.3620
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.028874   0.061594  16.704  < 2e-16 ***
## hosp             0.164797   0.015531  10.611  < 2e-16 ***
## healthexcellent -0.361993   0.078476  -4.613 4.09e-06 ***
## healthpoor       0.248307   0.046211   5.373 8.13e-08 ***
## numchron         0.146639   0.011860  12.364  < 2e-16 ***
## gendermale      -0.112320   0.033523  -3.351 0.000813 ***
## school           0.026143   0.004774   5.477 4.58e-08 ***
## privinsyes       0.201687   0.043661   4.619 3.96e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 6.706254)
##
##     Null deviance: 26943  on 4405  degrees of freedom
## Residual deviance: 23168  on 4398  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Dispersion parameter for quasipoisson family taken to be 6.706254. It is larger than 1 confirming that over-dispersion is present in the data

### Negative binomial regression

```
## negative binomial model
fm_nbin <- glm.nb(ofp~.,data=dt)
summary(fm_nbin)
```

```
##
## Call:
## glm.nb(formula = ofp ~ ., data = dt, init.theta = 1.206603534,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0469  -0.9955  -0.2948   0.2961   5.8185
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.929257   0.054591  17.022  < 2e-16 ***
## hosp             0.217772   0.020176  10.793  < 2e-16 ***
## healthexcellent -0.341807   0.060924  -5.610 2.02e-08 ***
```

```
## healthpoor       0.305013   0.048511   6.288 3.23e-10 ***
## numchron         0.174916   0.012092  14.466  < 2e-16 ***
## gendermale      -0.126488   0.031216  -4.052 5.08e-05 ***
## school           0.026815   0.004394   6.103 1.04e-09 ***
## privinsyes       0.224402   0.039464   5.686 1.30e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.2066) family taken to be 1)
##
##     Null deviance: 5743.7  on 4405  degrees of freedom
## Residual deviance: 5044.5  on 4398  degrees of freedom
## AIC: 24359
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.2066
##           Std. Err.:  0.0336
##
##  2 x log-likelihood:  -24341.1070
```

Both regression coefficients and standard errors are rather similar to the quasi-Poisson and the sandwich-adjusted Poisson results above. Thus, in terms of predicted means all three models give very similar results; the associated partial Wald tests also lead to the same conclusions.

### Hurdle regression

```
## a negative binomial herdle model
fm_hurdle0 <- hurdle(ofp ~ ., data = dt, dist = "negbin")
summary(fm_hurdle0)
```

```
##
## Call:
## hurdle(formula = ofp ~ ., data = dt, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1718 -0.7080 -0.2737  0.3196 18.0092
##
## Count model coefficients (truncated negbin with log link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.197699   0.058973  20.309  < 2e-16 ***
## hosp              0.211898   0.021396   9.904  < 2e-16 ***
## healthexcellent -0.331861   0.066093  -5.021 5.14e-07 ***
## healthpoor        0.315958   0.048056   6.575 4.87e-11 ***
## numchron          0.126421   0.012452  10.152  < 2e-16 ***
## gendermale       -0.068317   0.032416  -2.108   0.0351 *
## school            0.020693   0.004535   4.563 5.04e-06 ***
## privinsyes        0.100172   0.042619   2.350   0.0188 *
## Log(theta)        0.333255   0.042754   7.795 6.46e-15 ***
## Zero hurdle model coefficients (binomial with logit link):
```

```
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.043147   0.139852   0.309 0.757688
## hosp             0.312449   0.091437   3.417 0.000633 ***
## healthexcellent -0.289570   0.142682  -2.029 0.042409 *
## healthpoor      -0.008716   0.161024  -0.054 0.956833
## numchron         0.535213   0.045378  11.794  < 2e-16 ***
## gendermale      -0.415658   0.087608  -4.745 2.09e-06 ***
## school           0.058541   0.011989   4.883 1.05e-06 ***
## privinsyes       0.747120   0.100880   7.406 1.30e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 1.3955
## Number of iterations in BFGS optimization: 16
## Log-likelihood: -1.209e+04 on 17 Df
```

The coefficients in the count component resemble those from the previous models, but the increase in the log-likelihood conveys that the model has improved by including the hurdle component. However, it might be possible to omit the health variable from the hurdle model. To test this hypothesis, the reduced model is fitted via

```
fm_hurdle <- hurdle(ofp ~ . | hosp + numchron + privins + school + gender,
                    data = dt, dist = "negbin")
waldtest(fm_hurdle0, fm_hurdle)
```

```
## Wald test
##
## Model 1: ofp ~ .
## Model 2: ofp ~ . | hosp + numchron + privins + school + gender
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   4389
## 2   4391 -2 4.1213     0.1274
```

**Zero-inflated regression**

```
## zero-inflated negative binomial regression, with addtional probability weight for zero counts

# model 1
fm_zinb0 <- zeroinfl(ofp ~ ., data = dt, dist = "negbin")

# model 2
fm_zinb <- zeroinfl(ofp ~ . | hosp + numchron + privins + school + gender,
                    data = dt, dist = "negbin")
```

**Note here I got different outcome with the wald test in paper. The author said it improves the ZINB fit significantly, while I found basically there is no difference between two models and both of them are insignificant:**

```
waldtest(fm_zinb0, fm_zinb)
```

```
## Wald test
##
## Model 1: ofp ~ .
## Model 2: ofp ~ . | hosp + numchron + privins + school + gender
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   4389
## 2   4391 -2 0.1584     0.9239
```

```
summary(fm_zinb)
```

```
##
## Call:
## zeroinfl(formula = ofp ~ . | hosp + numchron + privins + school + gender,
##     data = dt, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1963 -0.7105 -0.2779  0.3253 17.8451
##
## Count model coefficients (negbin with log link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.193716   0.056661  21.068  < 2e-16 ***
## hosp              0.201477   0.020360   9.896  < 2e-16 ***
## healthexcellent -0.319336   0.060405  -5.287 1.25e-07 ***
## healthpoor        0.285133   0.045093   6.323 2.56e-10 ***
## numchron          0.128999   0.011930  10.813  < 2e-16 ***
## gendermale       -0.080276   0.031024  -2.588  0.00967 **
## school            0.021424   0.004358   4.916 8.81e-07 ***
## privinsyes        0.125861   0.041587   3.026  0.00247 **
## Log(theta)        0.394148   0.035035  11.250  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.04692    0.26855  -0.175  0.86131
## hosp        -0.80048    0.42081  -1.902  0.05714 .
## numchron    -1.24790    0.17830  -6.999 2.58e-12 ***
## privinsyes  -1.17562    0.22012  -5.341 9.25e-08 ***
## school      -0.08377    0.02625  -3.191  0.00142 **
## gendermale   0.64769    0.20011   3.237  0.00121 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 1.4831
## Number of iterations in BFGS optimization: 26
## Log-likelihood: -1.209e+04 on 15 Df
```

### Comparison

1. Inspect the estimated regression coefficients

```
fm <- list("ML-Pois" = fm_pois, "Quasi-Pois" = fm_qpois, "NB" = fm_nbin,
           "Hurdle-NB" = fm_hurdle, "ZINB" = fm_zinb)
sapply(fm, function(x) coef(x)[1:8])
```

```
##                     ML-Pois  Quasi-Pois          NB   Hurdle-NB        ZINB
## (Intercept)      1.02887420  1.02887420  0.92925658  1.19769892  1.19371596
## hosp             0.16479739  0.16479739  0.21777223  0.21189820  0.20147665
## healthexcellent -0.36199320 -0.36199320 -0.34180660 -0.33186113 -0.31933605
## healthpoor       0.24830697  0.24830697  0.30501303  0.31595757  0.28513296
## numchron         0.14663928  0.14663928  0.17491552  0.12642059  0.12899905
## gendermale      -0.11231992 -0.11231992 -0.12648813 -0.06831702 -0.08027596
## school           0.02614299  0.02614299  0.02681508  0.02069321  0.02142363
## privinsyes       0.20168688  0.20168688  0.22440187  0.10017164  0.12586120
```

There are some small differences, especially between the GLMs and the zero-augmented models.

However, the zero-augmented models are interpreted differently, because tthe difference between mean functions.

Moreover, the associated estimated standard errors are very similar as well

```
cbind("ML-Pois" = sqrt(diag(vcov(fm_pois))),
      "Adj-Pois" = sqrt(diag(sandwich(fm_pois))),
      sapply(fm[-1], function(x) sqrt(diag(vcov(x)))[1:8]))
```

```
##                     ML-Pois    Adj-Pois  Quasi-Pois          NB  Hurdle-NB
## (Intercept)     0.023784601 0.064529808 0.061593641 0.054591271 0.05897349
## hosp            0.005997367 0.021945186 0.015531043 0.020176492 0.02139606
## healthexcellent 0.030303905 0.077448586 0.078476316 0.060923623 0.06609306
## healthpoor      0.017844531 0.054021990 0.046210977 0.048510797 0.04805566
## numchron        0.004579677 0.012907865 0.011859732 0.012091749 0.01245231
## gendermale      0.012945146 0.035343487 0.033523316 0.031215523 0.03241561
## school          0.001843329 0.005084002 0.004773565 0.004393971 0.00453483
## privinsyes      0.016859826 0.043128006 0.043660942 0.039463744 0.04261858
##                       ZINB
## (Intercept)     0.056660810
## hosp            0.020359692
## healthexcellent 0.060404914
## healthpoor      0.045092607
## numchron        0.011930497
## gendermale      0.031023982
## school          0.004357567
## privinsyes      0.041587497
```

The only exception are the model-based standard errors for the Poisson model.

In summary, the models are not too different with respect to their fitted mean functions. The differences become obvious if not only the mean but the full likelihood is considered:

```
rbind(logLik = sapply(fm, function(x) round(logLik(x), digits = 0)),
      Df = sapply(fm, function(x) attr(logLik(x), "df")))
```

```
##        ML-Pois Quasi-Pois     NB Hurdle-NB   ZINB
## logLik  -17972         NA -12171    -12090 -12091
## Df          8          8      9        15     15
```

The ML Poisson model is inferior to all other fits.

The quasi-Poisson model and the sandwich-adjusted Poisson model are not associated with a fitted likelihood.

The negative binomial already improves the fit dramatically but can in turn be improved by the hurdle and zero-inflated models which give almost identical fits.

This also reflects that the over-dispersion in the data is captured better by the negative-binomial-based models than the plain Poisson model. Additionally, it is of interest how the zero counts are captured by the various models.

Therefore, the observed zero counts are compared to the expected number of zero counts for the likelihood-based models:

```
round(c("Obs" = sum(dt$ofp < 1),
        "ML-Pois" = sum(dpois(0, fitted(fm_pois))),
        "NB" = sum(dnbinom(0, mu = fitted(fm_nbin), size = fm_nbin$theta)),
        "NB-Hurdle" = sum(predict(fm_hurdle, type = "prob")[,1]),
        "ZINB" = sum(predict(fm_zinb, type = "prob")[,1])))
```

```
##       Obs   ML-Pois        NB NB-Hurdle      ZINB
##       683        47       608       683       709
```

The expected number of zero counts in the hurdle model matches the observed number.

In summary, the hurdle and zero-inflation models lead to the best results (in terms of likelihood) on this data set. Above, their mean function for the count component was already shown to be very similar, below we take a look at the fitted zero components:

```
t(sapply(fm[4:5], function(x) round(x$coefficients$zero, digits = 3)))
```

```
##           (Intercept)   hosp numchron privinsyes school gendermale
## Hurdle-NB       0.016  0.318    0.548      0.746  0.057     -0.419
## ZINB           -0.047 -0.800   -1.248     -1.176 -0.084      0.648
```

The signs of the coefficients match, i.e., are just inversed.

For the hurdle model, the zero hurdle component describes the probability of observing a positive count whereas, for the ZINB model, the zero-in ation component predicts the probability of observing a zero count from the point mass component.

Overall, both models lead to the same qualitative results and very similar model fits. Perhaps the hurdle model is slightly preferable because it has the nicer interpretation: there is one process that controls whether a patient sees a physician or not, and a second process that determines how many office visits are made.

## Lecture 20, Problem 4, Robust standard errors in the Karolinska data

Report the robust standard errors in the case study of Section 4 in Lecture 18.

```
## read data
karolinska = read.table ("karolinska.txt" , header = TRUE)
karolinska = karolinska [, c("HighVolDiagHosp", "HighVolTreatHosp", "AgeAtDiagnosis", "FromRuralArea",
```

```r
## binary logistic for the treatment

## (1) HighVolDiagHosp
diagglm = glm(HighVolDiagHosp ~ AgeAtDiagnosis + FromRuralArea + Male, data = karolinska, family = bino

summary(diagglm)
```

```
##
## Call:
## glm(formula = HighVolDiagHosp ~ AgeAtDiagnosis + FromRuralArea +
##     Male, family = binomial(link = "logit"), data = karolinska)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.06147  -0.98645  -0.05759   1.01391   1.75696
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.46604    1.14545   3.026 0.002479 **
## AgeAtDiagnosis -0.03124    0.01481  -2.110 0.034854 *
## FromRuralArea  -1.26322    0.34530  -3.658 0.000254 ***
## Male           -0.97524    0.41303  -2.361 0.018216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 219.03  on 157  degrees of freedom
## Residual deviance: 195.69  on 154  degrees of freedom
## AIC: 203.69
##
## Number of Fisher Scoring iterations: 4
```

```r
## report the robust standard errors
coeftest(diagglm, vcov = sandwich)
```

```
##
## z test of coefficients:
##
##                 Estimate Std. Error z value  Pr(>|z|)
## (Intercept)     3.466039   1.222072  2.8362 0.0045654 **
## AgeAtDiagnosis -0.031243   0.015347 -2.0357 0.0417773 *
## FromRuralArea  -1.263219   0.347742 -3.6326 0.0002805 ***
## Male           -0.975236   0.426254 -2.2879 0.0221421 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## (2) HighVolTreatHosp
treatglm = glm(HighVolTreatHosp ~ AgeAtDiagnosis + FromRuralArea + Male, data = karolinska, family = bi

summary(treatglm)
```

```
##
## Call:
## glm(formula = HighVolTreatHosp ~ AgeAtDiagnosis + FromRuralArea +
##     Male, family = binomial(link = "logit"), data = karolinska)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2912  -0.9978   0.5387   0.8408   1.4810
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     6.44683    1.49544   4.311 1.63e-05 ***
## AgeAtDiagnosis -0.06297    0.01890  -3.332 0.000862 ***
## FromRuralArea  -1.28777    0.39572  -3.254 0.001137 **
## Male           -0.74856    0.45285  -1.653 0.098329 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 194.04  on 157  degrees of freedom
## Residual deviance: 167.21  on 154  degrees of freedom
## AIC: 175.21
##
## Number of Fisher Scoring iterations: 4
```

```
## Report the robust standard errors.
coeftest(treatglm, vcov = sandwich)
```

```
##
## z test of coefficients:
##
##                 Estimate Std. Error z value  Pr(>|z|)
## (Intercept)     6.446834   1.383229  4.6607 3.151e-06 ***
## AgeAtDiagnosis -0.062973   0.017348 -3.6300 0.0002834 ***
## FromRuralArea  -1.287768   0.403546 -3.1911 0.0014172 **
## Male           -0.748559   0.435244 -1.7199 0.0854577 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Binary logistic for the outcome

## (1) use HighVolDiagHosp
karolinska$loneyear = (karolinska$YearsSurvivingAfterDiagnosis != "1")

loneyearglm = glm(loneyear ~ HighVolDiagHosp + AgeAtDiagnosis + FromRuralArea + Male, data = karolinska

summary(loneyearglm)
```

```
##
## Call:
## glm(formula = loneyear ~ HighVolDiagHosp + AgeAtDiagnosis + FromRuralArea +
##     Male, family = binomial(link = "logit"), data = karolinska)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1755  -0.9936  -0.7739   1.3024   1.8557
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.22919    1.15545  -1.064   0.2874
## HighVolDiagHosp  0.13684    0.36586   0.374   0.7084
## AgeAtDiagnosis  -0.00389    0.01411  -0.276   0.7829
## FromRuralArea    0.33360    0.35798   0.932   0.3514
## Male             0.86706    0.44034   1.969   0.0489 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 206.62  on 157  degrees of freedom
## Residual deviance: 201.16  on 153  degrees of freedom
## AIC: 211.16
##
## Number of Fisher Scoring iterations: 4
```

```
## Report the robust standard errors.
coeftest(loneyearglm, vcov = sandwich)
```

```
##
## z test of coefficients:
##
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.2291912  1.1026603 -1.1148   0.2650
## HighVolDiagHosp  0.1368369  0.3703163  0.3695   0.7117
## AgeAtDiagnosis  -0.0038898  0.0136228 -0.2855   0.7752
## FromRuralArea    0.3335974  0.3580537  0.9317   0.3515
## Male             0.8670612  0.4404522  1.9686   0.0490 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## (2) use HighVolTreatHosp
loneyearglm = glm(loneyear ~ HighVolTreatHosp + AgeAtDiagnosis + FromRuralArea + Male, data = karolinska

summary(loneyearglm)
```

```
##
## Call:
## glm(formula = loneyear ~ HighVolTreatHosp + AgeAtDiagnosis +
##     FromRuralArea + Male, family = binomial(link = "logit"),
##     data = karolinska)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3767  -0.9683  -0.6784   1.0813   2.0833
##
```

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.353977   1.317942  -2.545  0.01093 *
## HighVolTreatHosp  1.417458   0.455603   3.111  0.00186 **
## AgeAtDiagnosis    0.008725   0.014840   0.588  0.55655
## FromRuralArea     0.633278   0.368525   1.718  0.08572 .
## Male              1.079973   0.452191   2.388  0.01693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 206.62  on 157  degrees of freedom
## Residual deviance: 190.36  on 153  degrees of freedom
## AIC: 200.36
##
## Number of Fisher Scoring iterations: 3
```

```
## Report the robust standard errors.
coeftest(loneyearglm, vcov = sandwich)
```

```
##
## z test of coefficients:
##
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.3539771  1.3195207 -2.5418 0.011028 *
## HighVolTreatHosp  1.4174575  0.4701220  3.0151 0.002569 **
## AgeAtDiagnosis    0.0087254  0.0148046  0.5894 0.555613
## FromRuralArea     0.6332775  0.3689165  1.7166 0.086055 .
## Male              1.0799735  0.4436277  2.4344 0.014916 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Multinomial logistic for the outcome

## (1) Use HighVolDiagHosp
yearmultinom = multinom(YearsSurvivingAfterDiagnosis ~ HighVolDiagHosp + AgeAtDiagnosis + FromRuralArea

result = summary(yearmultinom)

result
```

```
## Call:
## multinom(formula = YearsSurvivingAfterDiagnosis ~ HighVolDiagHosp +
##     AgeAtDiagnosis + FromRuralArea + Male, data = karolinska,
##     trace = FALSE)
##
## Coefficients:
##     (Intercept) HighVolDiagHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4   -1.075818     -0.06973187   -0.004624030     0.1744256 0.5028786
## 5+    -4.180416      0.64036289   -0.001846453     0.7365111 2.1628717
##
## Std. Errors:
```

```
##     (Intercept) HighVolDiagHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4    1.286987       0.4113006     0.01596377     0.4014718 0.4716831
## 5+     2.003581       0.5816365     0.02148936     0.5741017 1.0741239
##
## Residual Deviance: 268.2616
## AIC: 288.2616
```

```
## For reporting the robust standard errors, here use nonparametric bootstrap
B = 5000
n = nrow(karolinska)
boot.coef = list()

for (i in 1:B){
  id = sample(c(1:n), n, replace = TRUE)
  boot.dat = karolinska[id,]
  boot.yearmultinon = multinom(YearsSurvivingAfterDiagnosis ~ HighVolDiagHosp + AgeAtDiagnosis + FromRu
  boot.result = summary(boot.yearmultinon)
  boot.coef [[i]] = boot.result$coefficients
}

robust.se = matrix(0, 2, 5)
for(i in 1:2){
  for(j in 1:5){
    tmp = c()
    for(k in 1:B){
      tmp[k] = boot.coef[[k]][i, j]
      }
    robust.se[i, j] = sqrt(var(tmp))
  }
}

coef = result$coefficients
robust.z = coef/robust.se
colnames(robust.se) = colnames(robust.z)
rownames(robust.se) = rownames(robust.z)

## report the robust standard errors
robust.se
```

```
##     (Intercept) HighVolDiagHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4    1.338658       0.4489893     0.01675556     0.4290269 0.5400089
## 5+     3.905956       0.6734710     0.02245013     0.7023679 3.7457735
```

```
## report z values
robust.z
```

```
##     (Intercept) HighVolDiagHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4  -0.8036546      -0.1553085    -0.27596986     0.4065611 0.9312413
## 5+   -1.0702672       0.9508396    -0.08224687     1.0486115 0.5774166
```

```
## (2) Use HighVolTreatHosp
yearmultinom = multinom(YearsSurvivingAfterDiagnosis ~ HighVolTreatHosp + AgeAtDiagnosis + FromRuralArea
```

```r
result = summary(yearmultinom)

result
```

```
## Call:
## multinom(formula = YearsSurvivingAfterDiagnosis ~ HighVolTreatHosp +
##     AgeAtDiagnosis + FromRuralArea + Male, data = karolinska,
##     trace = FALSE)
##
## Coefficients:
##     (Intercept) HighVolTreatHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4   -3.312433         1.326354    0.008527561     0.5186654 0.7514451
## 5+    -5.935172         1.627711    0.008978103     0.9063831 2.2780877
##
## Std. Errors:
##     (Intercept) HighVolTreatHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4    1.463258        0.5141127     0.01660648     0.4085976 0.4806953
## 5+     2.190305        0.7320788     0.02244867     0.5645595 1.0739669
##
## Residual Deviance: 258.5675
## AIC: 278.5675
```

```r
## For reporting the robust standard errors, here use nonparametric bootstrap
B = 5000
n = nrow(karolinska)
boot.coef = list()

for (i in 1:B){
  id = sample(c(1:n), n, replace = TRUE)
  boot.dat = karolinska[id,]
  boot.yearmultinon = multinom(YearsSurvivingAfterDiagnosis ~ HighVolTreatHosp + AgeAtDiagnosis + FromRu
  boot.result = summary(boot.yearmultinon)
  boot.coef [[i]] = boot.result$coefficients
}

robust.se = matrix(0, 2, 5)
for(i in 1:2){
  for(j in 1:5){
    tmp = c()
    for(k in 1:B){
      tmp[k] = boot.coef[[k]][i, j]
      }
    robust.se[i, j] = sqrt(var(tmp))
  }
}

coef = result$coefficients
robust.z = coef/robust.se
colnames(robust.se) = colnames(robust.z)
rownames(robust.se) = rownames(robust.z)

## report the robust standard errors
robust.se
```

```
##     (Intercept) HighVolTreatHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4    1.668460        0.6472103     0.01792938     0.4315407 0.5464329
## 5+     4.474504        1.9308743     0.02317520     0.7088812 3.8127659
```

```
## report z values
robust.z
```

```
##     (Intercept) HighVolTreatHosp AgeAtDiagnosis FromRuralArea      Male
## 2-4   -1.985324        2.0493392      0.4756195      1.201892 1.3751826
## 5+    -1.326443        0.8429916      0.3874014      1.278611 0.5974895
```

```
## Obtain the fitted probabilities of each category
predict(yearmultinom , type = "probs")[1:5 , ]
```

```
##           1       2-4         5+
## 1 0.7046514 0.1952602 0.10008835
## 2 0.7064547 0.1940977 0.09944761
## 3 0.7589625 0.2187152 0.02232230
## 4 0.7046514 0.1952602 0.10008835
## 5 0.5312053 0.3190527 0.14974200
```

```
## proportional odds regression on outcome with orders

## (1) HighVolDiagHosp
yearpo = polr(YearsSurvivingAfterDiagnosis ~ HighVolDiagHosp + AgeAtDiagnosis +
                FromRuralArea + Male, Hess = TRUE, data = karolinska)
summary(yearpo)
```

```
## Call:
## polr(formula = YearsSurvivingAfterDiagnosis ~ HighVolDiagHosp +
##     AgeAtDiagnosis + FromRuralArea + Male, data = karolinska,
##     Hess = TRUE)
##
## Coefficients:
##                     Value Std. Error t value
## HighVolDiagHosp  0.216755    0.35892  0.6039
## AgeAtDiagnosis  -0.002881    0.01378 -0.2091
## FromRuralArea    0.371898    0.35313  1.0532
## Male             0.943955    0.43588  2.1656
##
## Intercepts:
##        Value   Std. Error t value
## 1|2-4   1.4079  1.1309     1.2450
## 2-4|5+  2.9284  1.1514     2.5434
##
## Residual Deviance: 271.0778
## AIC: 283.0778
```

```
## Report the robust standard errors.
coeftest(yearpo, vcov = sandwich)
```

```
##
## t test of coefficients:
##
##                  Estimate Std. Error t value Pr(>|t|)
## HighVolDiagHosp  0.216755   0.369167  0.5871  0.55798
## AgeAtDiagnosis  -0.002881   0.013263 -0.2172  0.82833
## FromRuralArea    0.371898   0.361450  1.0289  0.30516
## Male             0.943955   0.422094  2.2364  0.02678 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## (2) HighVolTreatHosp
yearpo = polr(YearsSurvivingAfterDiagnosis ~ HighVolTreatHosp + AgeAtDiagnosis +
FromRuralArea + Male, Hess = TRUE, data = karolinska)
summary(yearpo)
```

```
## Call:
## polr(formula = YearsSurvivingAfterDiagnosis ~ HighVolTreatHosp +
##     AgeAtDiagnosis + FromRuralArea + Male, data = karolinska,
##     Hess = TRUE)
##
## Coefficients:
##                     Value Std. Error t value
## HighVolTreatHosp 1.399538    0.44518  3.1438
## AgeAtDiagnosis   0.008032    0.01438  0.5584
## FromRuralArea    0.638862    0.35450  1.8022
## Male             1.122698    0.44377  2.5299
##
## Intercepts:
##        Value  Std. Error t value
## 1|2-4  3.3273 1.2752     2.6092
## 2-4|5+ 4.9258 1.3106     3.7583
##
## Residual Deviance: 260.2831
## AIC: 272.2831
```

```
## Report the robust standard errors.
coeftest(yearpo, vcov = sandwich)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## HighVolTreatHosp 1.3995378  0.4603514  3.0402 0.002785 **
## AgeAtDiagnosis   0.0080323  0.0142499  0.5637 0.573806
## FromRuralArea    0.6388620  0.3571815  1.7886 0.075668 .
## Male             1.1226975  0.4185509  2.6823 0.008120 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```