STAT/BIOST 571: Homework 2

Philip Pham

February 1, 2019

Problem 1: Two-stage least squares (10 points)

Consider a two-stage least squares estimation methodology similar to that described in the lecture notes on slides 1.93 and 1.94. Make the simplifying assumption that the $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are each comprised of an intercept and one other covariate; that is, consider one between-subject covariate and one within-subject covariate. Thus, in the first stage of the two-stage procedure we obtain $(\hat{\alpha}_{0i}, \hat{\alpha}_{1i})$ by applying OLS to each subject's data, as if the linear model

$$E(Y_{ij} | z_{ij}) = \alpha_{0i} + \alpha_{1i} z_{ij} \tag{1}$$

holds, and in the second stage we obtain $\hat{\beta}_0$ and $\hat{\beta}_1$ by applying OLS to $\hat{\alpha}_{1i}$ as if the linear model

$$E(\hat{\alpha}_{1i} | x_i) = \beta_0 + \beta_1 x_i \tag{2}$$

holds. Note that neither equation (1) nor (2) is meant to specify what actually happens in the data-generating mechanism; they are shorthand to describe the two-stage OLS procedure that gives rise to $\hat{\beta}_0$ and $\hat{\beta}_1$.

Rather, assume that

$$Y_{ij} = \gamma_0 + \gamma_1 x_i + \gamma_2 z_{ij} + \gamma_3 x_i z_{ij} + \epsilon_{ij} \tag{3}$$

for fixed but unknown regression coefficients $\gamma_0, \ldots, \gamma_3$, where the vectors $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{im_i})$ are multivariate normal with mean zero and a common variance for all observations of all subjects, and are independent of each other. The elements of $\boldsymbol{\epsilon}_i$ may be correlated, but we assume that $m_i = m_{i'}$ for all i, i' and that the covariance matrices are the same for all subjects.

(a) *Identify which of the regression coefficients in (3) is consistently estimated by $\hat{\beta}_1$ obtained from two-stage least squares. Justify your answer.*

Solution: $\hat{\beta}_1$ consistently estimates γ_3 in Equation 3.

To see this, we can rewrite Equation 3 as

$$Y_{ij} = (\gamma_0 + \gamma_1 x_i) + (\gamma_2 + \gamma_3 x_i) z_{ij} + \epsilon_{ij}.$$

Using the least squares estimator, we have that

$$\hat{\alpha}_i = \begin{pmatrix} \hat{\alpha}_{0i} \\ \hat{\alpha}_{1i} \end{pmatrix} = \left(Z_i^\intercal Z_i\right)^{-1} Z_i^\intercal Y_i, = \alpha_i + \left(Z_i^\intercal Z_i\right)^{-1} Z_i^\intercal \boldsymbol{\epsilon}_i, = \begin{pmatrix} \gamma_0 + \gamma_1 x_i \\ \gamma_2 + \gamma_3 x_i \end{pmatrix} + \left(Z_i^\intercal Z_i\right)^{-1} Z_i^\intercal \boldsymbol{\epsilon}_i,$$

where Z_i is an $m_i \times 2$ matrix, where $Z_{ij1} = 1$ and $Z_{ij2} = z_{ij}$. Taking the expectation, we obtain

$$\mathbb{E}\left[\hat{\alpha}_i\right] = \alpha_i. \tag{4}$$

Let $\hat{A} = \begin{pmatrix} \hat{\alpha}_1 & \cdots & \hat{\alpha}_n \end{pmatrix}^\intercal$, that is, the ith row is $\begin{pmatrix} \hat{\alpha}_{0i} & \hat{\alpha}_{1i} \end{pmatrix}$. Let $\hat{A}_2$ be second column of $\hat{A}$, so $\hat{A}_{2i} = \hat{\alpha}_{1i}$. Let X be an $n \times 2$ matrix, where $X_{i1} = 1$ and $X_{i2} = x_i$. In the next stage, we estimate β with

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \left(X^\intercal X\right)^{-1} X^\intercal \hat{A}_2.$$

Thus, we'll have that

$$\begin{aligned} \mathbb{E}\left[\hat{\beta}\right] &= \mathbb{E}\left[\mathbb{E}\left[\hat{\beta} \mid \hat{A}\right]\right] = \left(X^\intercal X\right)^{-1} X^\intercal \mathbb{E}\left[\hat{A}_2\right] \\ &= \left(X^\intercal X\right)^{-1} X^\intercal \begin{pmatrix} \gamma_2 + \gamma_3 x_i \\ \vdots \\ \gamma_2 + \gamma_3 x_n \end{pmatrix} \\ &= \left(X^\intercal X\right)^{-1} X^\intercal X \begin{pmatrix} \gamma_2 \\ \gamma_3 \end{pmatrix} = \begin{pmatrix} \gamma_2 \\ \gamma_3 \end{pmatrix}. \end{aligned}$$

By law of large numbers $\hat{\beta}_1 \to \mathbb{E}\left[\hat{\beta}_1\right] = \gamma_3$ as desired.

(b) *Consider four approaches to deriving Wald type 95% confidence intervals for the parameter identified in part (a)*

  i. *One-stage OLS based on (3), with maximum likelihood standard errors*
  ii. *One-stage OLS based on (3), with sandwich standard errors*
  iii. *Two-stage OLS based on (1) and (2), with maximum likelihood standard errors from fitting (2)*
  iv. *Two-stage OLS based on (1) and (2), with sandwich standard errors from fitting (2)*

Explain which of these approaches will lead to asymptotically valid confidence intervals.

Solution: The answer depends on if the elements of $\boldsymbol{\epsilon}_i$ are correlated. Let $\boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \Sigma_i\right)$, where $\Sigma_i = \Sigma_j$ for all i and j. Let

$$\Sigma = \begin{pmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_n \end{pmatrix}. \tag{5}$$

i. To do one-stage OLS. Let $X^{\star}$ be a $\sum_{i=1}^{n} m_i \times 4$ matrix where the columns are 1s, the x_is, the z_{ij}s, and $x_i z_{ij}$, respectively. We would estimate γ with $\hat{\gamma} = \left(X^{\star\intercal}X^{\star}\right)^{-1} X^{\star\intercal}Y$, which has distribution $\hat{\gamma} \sim \mathcal{N}\left(\gamma, \left(X^{\star\intercal}X^{\star}\right)^{-1} X^{\star\intercal}\Sigma X^{\star}\left(X^{\star\intercal}X^{\star}\right)^{-1}\right)$.

   In general, we would not get valid asymptotic standard errors if we were to use the usual OLS estimate for variance, $\hat{\sigma}^2\left(X^{\star\intercal}X^{\star}\right)^{-1}$. In the special case that $\Sigma_i = \sigma^2 I$ for all i, we would get valid asymptotic standard errors.

ii. The sandwich estimator produces valid standard errors only if we choose $\hat{\Sigma}$ such that $\hat{\Sigma} \to \Sigma$. One way to do this is to use the assumption that each cluster has the same variance, and let $\hat{\Sigma}_j = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - X_i^{\star}\hat{\gamma}\right)^{\intercal}\left(Y_i - X_i^{\star}\hat{\gamma}\right)$, where $X_i^{\star}$ and Y_i are the covariates and response corresponding to each cluster.

   If we use the usual covariance matrix estimate that assumes independence, we would not get valid standard errors, however, unless the data were actually independent.

iii. Following the first stage, we have that our estimator for α_i is

$$\hat{\alpha}_i \sim \mathcal{N}\left(\alpha_i, \left(Z_i^{\intercal}Z_i\right)^{-1} Z_i^{\intercal}\Sigma_i Z_i\left(Z_i^{\intercal}Z_i\right)^{-1}\right). \tag{6}$$

   In this way, we can write $\hat{\alpha}_{1i} = \gamma_2 + \gamma_3 x_i + \delta_i$, where

$$\delta_i \sim \mathcal{N}\left(0, \left(\left(Z_i^{\intercal}Z_i\right)^{-1} Z_i^{\intercal}\Sigma_i Z_i\left(Z_i^{\intercal}Z_i\right)^{-1}\right)_{22}\right),$$

   and are independent since each cluster is independent.

   In the second stage, we regress on $\hat{\alpha}_{1i}$. While each response observation is independent, the errors are not identically distributed, so in general, our standard errors would not be valid.

iv. Sandwich estimation deals with heteroscedasticity by taking the sample covariance. Using $X^{\intercal}\,\mathrm{diag}\left(\left(\hat{A}_2 - X\hat{\beta}\right)^{\intercal}\left(\hat{A}_2 - X\hat{\beta}\right)\right)X$ as the meat of the sandwich should give valid standard errors.

(c) *Does your answer to part (b) change if you assume $\boldsymbol{z}_i = \boldsymbol{z}_{i'}$ for all i, i'?*

Solution: Yes for part (iii). Having $\boldsymbol{z}_i = \boldsymbol{z}_{i'}$ for all i, i' along with the fact tht $\Sigma_i = \Sigma_{i'}$ for all i, i' ensures that the δ_i are idependent and identically distributed. In this case, standard errors are accurately estimated by $\hat{\delta}^2\left(X^{\intercal}X\right)^{-1}$.

Problem 2: Overdispersion and correlation in clustered binary data (10 points)

Download the Indonesian Children's Health Study (ICHS) dataset from the course website. The file contains data for 275 *preschool children examined for respiratory infection at up to six quarterly visits. The respiratory infection status, current age, and the baseline age at the first visit were recorded for each child at each visit. Later in the course, we will use this dataset to estimate the association between vitamin A deficiency (assessed through an occular measurement of xerophthalmia) in preschool children and occurrence of respiratory infections. For now, we will focus on estimating the correlation and/or* *overdispersion that result from repeated measurements of the same subjects. The variables you will need to pay attention to are:*

$$\begin{aligned} \texttt{id} &= \textit{Subject ID} \\ \texttt{infect} &= \textit{Infection status at current visit} \\ \texttt{baseage} &= \textit{Age at baseline (months-36)} \\ \texttt{xero} &= \textit{Xerophthalmia status at current visit} \end{aligned}$$

(a) *Fit an overdispersed logistic regression model with only an intercept to the Bernoulli version of these data (i.e., one binary observation for each child, at each visit). What is the estimated dispersion factor? What can you say theoretically about the dispersion factor? If there is a difference between the theoretical and estimated values, how can you explain it?*

Solution: Let there be $n = 275$ subjects. Suppose each subject i has m_i observations. Let $Y_{ij} \in \{0, 1\}$ where $j \in \{1, 2, \ldots, m_i\}$.

In this case, we are just fitting the model

$$\begin{aligned} Y_{ij} &\sim \text{Bernoulli}\,(p) \\ p &= \frac{1}{1 + \exp\left(-\beta_0\right)} \Rightarrow \log \frac{p}{1-p} = \beta_0. \end{aligned}$$

We'd estimate that $\hat{p} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_i \Big/ \sum_{i=1}^{n} m_i \approx \boxed{0.0892}$ and that $\hat{\beta}_0 \approx -2.324$.

We'd have that $\mathbb{E}\left[\hat{p}\right] = p$ and $\operatorname{var}\left(\hat{p}\right) = \frac{p(1-p)}{\sum_{i=1}^{n} m_i}$. It's as if we're just drawing one sample from a binomial distribution, so the dispersion factor is just 1.

Indeed, the estimated dispersion factor is

$$\hat{\alpha} = \sum_{i=1} \sum_{j=1}^{m_i} \frac{\left(y_{ij} - \hat{p}\right)^2}{\hat{p}\left(1 - \hat{p}\right)} \Bigg/ \left(\sum_{i=1}^{n} m_i - 1\right) \approx 1.000834,$$

which agrees with the theoretical dispersion factor and will approach 1 with increasing sample size.

(b) *Now convert the data into binomial observations, by aggregating over all visits to obtain a single binomial outcome for each child. Again fit an overdispersed logistic regression model with only an intercept, but this time do it using both quasi-likelihood and beta-binomial models. Fit the beta-binomial model two different ways, using the `vglm()` function in the `VGAM` package in `R` and by direct optimization of the log-likelihood. For each of these three model fits, report the estimated overdispersion or correlation parameter (as appropriate) and calculate a corresponding 95% confidence interval if this is possible without much additional work (i.e., if standard error estimates are reported or can be computed from a Hessian matrix). Discuss how the point estimates for overdispersion/correlation from these three model fits relate to each other.*

Solution: In this case, the binomial model is

$$\begin{aligned} Y_i &\sim \text{Binomial}\,(m_i, p) \\ p &= \frac{1}{1 + \exp\left(-\beta_0\right)} \Rightarrow \log \frac{p}{1-p} = \beta_0. \end{aligned}$$

The log-likelihood function is

$$l\left(\beta_0\right)=\sum_{i=1}^{n}\left(\log\binom{m_i}{y_i}+y_i\log p+\left(m_i-y_i\right)\log\left(1-p\right)\right),$$

so the score function is

$$S\left(\beta_0\right)=\sum_{i=1}^{n}\left(\frac{y_i}{p}-\frac{m_i-y_i}{1-p}\right)\frac{\mathrm{d}p}{\mathrm{d}\beta_0}=\frac{\mathrm{d}p}{\mathrm{d}\beta_0}\sum_{i=1}^{n}\frac{y_i-pm_i}{p\left(1-p\right)}.$$

The expected information matrix can be computed

$$I_n\left(\beta_0\right)=\operatorname{var}\left(S\left(p\right)\right)=\left(\frac{\mathrm{d}p}{\mathrm{d}\beta_0}\right)^2\sum_{i=1}^{n}\frac{m_i}{p\left(1-p\right)}=p^4\exp\left(-2\beta_0\right)\sum_{i=1}^{n}\frac{m_i}{p\left(1-p\right)}.$$

Solving $S\left(\hat{p}\right)=0$ gives that $\hat{p}=\sum_{i=1}^{n}\sum_{j=1}^{m_i}y_i\Big/\sum_{i=1}^{n}m_i\approx\boxed{0.0892}$ and that $\hat{\beta}_0\approx-2.324$ just as before. In this case, the dispersion factor differs:

$$\hat{\alpha}=\sum_{i=1}\frac{\left(y_i-m_i\hat{p}\right)^2}{m_i\hat{p}\left(1-\hat{p}\right)}\Bigg/\left(n-1\right)\approx 1.3462872,$$

so we have evidence of overdispersion. The variance of $\hat{\beta}_0$ is then $\hat{\alpha}I_n^{-1}\left(\hat{p}\right)\approx 0.013814$, so we have the 95% confidence interval for β_0, $[-2.554212,-2.09349]$.

The beta-binomial model makes different assumptions:

$$\begin{aligned}Y_i&\sim\operatorname{Binomial}\left(m_i,p_i\right)\\p_i&\sim\operatorname{Beta}\left(\alpha,\beta\right),\end{aligned}$$

where we can parameterize α and β in terms of $\mathbb{E}\left[p_i\right]=\mu$ and within-cluster correlation ρ as

$$\begin{aligned}\mu&=\frac{\alpha}{\alpha+\beta}\Rightarrow\alpha=\mu\frac{1-\rho}{\rho}\\\rho&=\frac{1}{1+\alpha+\beta}\Rightarrow\beta=\left(1-\mu\right)\frac{1-\rho}{\rho}.\end{aligned}$$

Let $\operatorname{logit}\left(\mu\right)=\gamma_{\mu,0}$ and $\operatorname{logit}\left(\rho\right)=\gamma_{\rho,0}$. Let B be the beta function. The log-likelihood function is now

$$l\left(\gamma_{\mu,0},\gamma_{\rho,0}\right)=\sum_{i=1}^{n}\left(\log\binom{m_i}{y_i}+\log B\left(\alpha+y_i,\beta+m_i-y_i\right)-\log B\left(\alpha,\beta\right)\right).$$

Let ψ be the digamma function. The score function is then

$$S\left(\gamma_{\mu,0},\gamma_{\rho,0}\right)=D_\gamma^\intercal\left(\begin{pmatrix}\mu\\\rho\end{pmatrix}\right)\Bigg|_{\gamma_{\mu,0},\gamma_{\rho,0}}\begin{pmatrix}\frac{\partial\alpha}{\partial\mu}&\frac{\partial\beta}{\partial\mu}\\\frac{\partial\alpha}{\partial\rho}&\frac{\partial\beta}{\partial\rho}\end{pmatrix}\sum_{i=1}^{n}\begin{pmatrix}\psi\left(\alpha+y_i\right)-\psi\left(\alpha+\beta+m_i\right)-\left(\psi\left(\alpha\right)-\psi\left(\alpha+\beta\right)\right)\\\psi\left(\beta+m_i-y_i\right)-\psi\left(\alpha+\beta+m_i\right)-\left(\psi\left(\beta\right)-\psi\left(\alpha+\beta\right)\right)\end{pmatrix}.$$

Both vglm and numerical optimization find that

$$\hat{\mu} = 0.0901559 \Rightarrow \hat{\gamma}_{\mu,0} = -2.31173$$
$$\hat{\rho} = 0.082218 \Rightarrow \hat{\gamma}_{\rho,0} = -2.4126.$$

The 95% confidence interval using the standard errors reported by vglm are $[-2.53886, -2.08454]$ for $\gamma_{\mu,0}$ and $[-3.23343, -1.5918]$ for $\gamma_{\rho,0}$. The dispersion factor for each i is then $\hat{\alpha}_i = 1 + \hat{\rho}\,(m_i - 1)$.

(c) Repeat all of the steps in part (b), but this time include a linear term for baseline age in the mean component of the logistic regression model.

Solution: The steps are largely the same. However, in the binomial model, we now have that $\text{logit}\,(p_i) = \beta_0 + \beta_1 x_i$, where x_i is the baseage for subject i.

For β_0, we obtain the estimate $\hat{\beta}_0 = -2.549413$ and 95% confidence interval $[-2.8156, -2.2832]$. For β_1, we obtain the estimate $\hat{\beta}_1 = -0.0275$ and 95% confidence interval $[-0.04, -0.015]$. The dispersion factor has estimate $\hat{\alpha} = 1.196207$.

In the beta-binomial model, we now let μ vary among subects, that is, $\text{logit}\,(\mu_i) = \gamma_{\mu,0} + \gamma_{\mu,0} x_i$. $\hat{\gamma}_{\mu,0} = -2.521$ with 95% confidence interval $[-2.7857, -2.2561]$, $\hat{\gamma}_{\mu,1} = -0.026456$ with 95% confidence interval $[-0.03881, -0.0141]$. $\hat{\gamma}_{\rho,0} = -2.707920$ with 95% confidence interval $[-3.7018, -1.714]$, so $\hat{\rho} = 0.06251$.

(d) Discuss the main differences between your findings in parts (b) and (c).

Solution: All coefficient estimates were statistically significant at level 0.05 since none of the confidence intervals include 0.

In part (b) there is more dispersion than part (c), whether estimated through the dispersion factor $\hat{\alpha}$ in the binomial model or within-cluster correlation $\hat{\rho}$. By including baseage as a covariate, we're accounting for more of the difference between clusters. Without the covariate, the differences between subjects is either accounted by the dispersion factor or explained away by within-cluster correlation.

It's also somewhat notable that the standard errors for the intercept terms increase in part (c) since estimating more parameters introduces variance.

Appendix

The models were fit in R. The parameter estimates for the beta-binomial model were verified in Python by numerically maximizing the log-likelihood function.

Fitting Various Models in R

In [1]:

```
library(data.table)
library(MASS)
library(VGAM)

ichs.data <- data.table(read.table('ichs.txt'))
head(ichs.data)
```

```
Loading required package: stats4
Loading required package: splines
```

id	gender	height	cosv	sinv	xero	baseage	age	infect
121013	0	-3	-1	0	0	31	31	0
121013	0	-3	0	-1	0	31	34	0
121013	0	-2	1	0	0	31	37	0
121013	0	-2	0	1	0	31	40	0
121013	0	-2	-1	0	0	31	43	1
121013	0	-3	0	-1	0	31	46	0

In [2]:

```
bernoulli.model <- glm(infect ~ 1, data=ichs.data, family = quasibinomial)
summary(bernoulli.model)
```

```
Call:
glm(formula = infect ~ 1, family = quasibinomial, data = ichs.data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.4322  -0.4322  -0.4322  -0.4322   2.1987  

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -2.3239     0.1013  -22.93   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.000834)

    Null deviance: 721.45  on 1199  degrees of freedom
Residual deviance: 721.45  on 1199  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

In [3]:

```
binomial.model <- glm(
    cbind(infect, n - infect) ~ 1, family = quasibinomial,
    data=ichs.data[,list(n=length(infect), infect=sum(infect)),by=.(id)])
summary(binomial.model)
```

```
Call:
glm(formula = cbind(infect, n - infect) ~ 1, family = quasibinomial,
    data = ichs.data[, list(n = length(infect), infect = sum(infect)),
        by = .(id)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0587  -1.0587  -0.7486   0.6014   4.3975

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.3239     0.1175  -19.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.346287)

    Null deviance: 312.47  on 274  degrees of freedom
Residual deviance: 312.47  on 274  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

In [4]:

```
beta.binomial.model <- vglm(
    cbind(infect, n - infect) ~ 1, family = betabinomial,
    data=ichs.data[,list(n=length(infect), infect=sum(infect)),by=.(id)])
summary(beta.binomial.model)
```

```
Warning message in checkwz(wz, M, trace = trace, wzepsilon = control$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"

Call:
vglm(formula = cbind(infect, n - infect) ~ 1, family = betabinomial,
    data = ichs.data[, list(n = length(infect), infect = sum(infect)),
        by = .(id)])


Pearson residuals:
               Min         1Q  Median     3Q    Max
logit(mu)  -0.7131 -7.131e-01 -0.5279 0.8009  4.098
logit(rho) -1.2916  2.847e-11  0.1347 0.2801 11.956

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  -2.3117     0.1159  -19.95  < 2e-16 ***
(Intercept):2  -2.4126     0.4188   -5.76  8.4e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  2

Names of linear predictors: logit(mu), logit(rho)

Log-likelihood: -221.2778 on 548 degrees of freedom

Number of iterations: 3

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):2'
```

In [5]:

```
binomial.model <- glm(
    cbind(infect, n - infect) ~ 1 + baseage, family = quasibinomial,
    data=ichs.data[,list(n=length(infect), infect=sum(infect), baseage=mean(baseag
e)),by=.(id)])
summary(binomial.model)
```

```
Call:
glm(formula = cbind(infect, n - infect) ~ 1 + baseage, family = quasibinomial,
    data = ichs.data[, list(n = length(infect), infect = sum(infect),
        baseage = mean(baseage)), by = .(id)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4390  -0.8773  -0.5940   0.3292   4.1245

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.549413   0.135822 -18.770  < 2e-16 ***
baseage     -0.027499   0.006375  -4.314 2.25e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.196207)

    Null deviance: 312.47  on 274  degrees of freedom
Residual deviance: 287.74  on 273  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

In [6]:

```
beta.binomial.model <- vglm(
    cbind(infect, n - infect) ~ 1 + baseage, family = betabinomial,
    data=ichs.data[,list(n=length(infect), infect=sum(infect), baseage=mean(baseag
e)),by=.(id)])
summary(beta.binomial.model)
```

```
Warning message in checkwz(wz, M, trace = trace, wzepsilon = control$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in checkwz(wz, M = M, trace = trace, wzepsilon = contro
l$wzepsilon):
"22 diagonal elements of the working weights variable 'wz' have been replaced by
1.819e-12"Warning message in matrix.power(wz, M = M, power = 0.5, fast = TRUE):
"Some weight matrices have negative eigenvalues. They will be assigned NAs"Warnin
g message in matrix.power(wz, M = M, power = 0.5, fast = TRUE):
"Some weight matrices have negative eigenvalues. They will be assigned NAs"

Call:
vglm(formula = cbind(infect, n - infect) ~ 1 + baseage, family = betabinomial,
    data = ichs.data[, list(n = length(infect), infect = sum(infect),
        baseage = mean(baseage)), by = .(id)])


Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept):1 -2.520914   0.135098 -18.660  < 2e-16 ***
(Intercept):2 -2.707920   0.507098  -5.340 9.29e-08 ***
baseage       -0.026456   0.006302  -4.198 2.69e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  2

Names of linear predictors: logit(mu), logit(rho)

Log-likelihood: -211.5263 on 547 degrees of freedom

Number of iterations: 6

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):1', '(Intercept):2'
```

Beta-Binomial Model

In [1]:

```
import pandas as pd
import numpy as np
from scipy import special
from scipy import optimize

ichs_data = pd.read_csv('ICHS.txt', sep=' ')
ichs_data.head()
```

Out[1]:

	id	gender	height	cosv	sinv	xero	baseage	age	infect
1	121013	0	-3	-1	0	0	31	31	0
2	121013	0	-3	0	-1	0	31	34	0
3	121013	0	-2	1	0	0	31	37	0
4	121013	0	-2	0	1	0	31	40	0
5	121013	0	-2	-1	0	0	31	43	1

In [2]:

```
ichs_data_grouped = pd.DataFrame({
    'baseage': ichs_data[['id', 'baseage']].groupby('id').mean()['baseage'],
    'infect': ichs_data[['id', 'infect']].groupby('id').sum()['infect'],
    'n': ichs_data.groupby('id').size()
})

ichs_data_grouped.head()
```

Out[2]:

	baseage	infect	n
id			
121013	31	1	6
121113	-9	1	6
121114	-26	2	6
121140	-19	0	2
121215	0	2	6

In [3]:

```
def score_beta_binomial(X, y, n, gamma):
    X = np.column_stack((np.ones_like(y), X))

    mu = np.matmul(X, gamma[:-1])
    rho = gamma[-1]

    alpha = mu*(1-rho)/rho
    beta = (1 - mu)*(1 - rho)/rho

    return np.sum(np.array([
        special.digamma(alpha + y) - special.digamma(alpha + beta + n) - special.d
igamma(alpha) + special.digamma(alpha + beta),
        special.digamma(beta + n - y) - special.digamma(alpha + beta + n) - specia
l.digamma(beta) + special.digamma(alpha + beta),
    ]), axis=-1)
```

In [4]:

```
mu, rho = optimize.root(
    lambda x: score_beta_binomial(ichs_data_grouped[[]].values,
                                  ichs_data_grouped['infect'].values,
                                  ichs_data_grouped['n'].values,
                                  x),
    [0.1, 0.1])['x']
mu, rho
```

Out[4]:

```
(0.09015590102588235, 0.08221790596403289)
```

In [5]:

```
special.logit(mu), special.logit(rho)
```

Out[5]:

```
(-2.311732859660446, -2.4125868796604526)
```