STAT 571 HW8

Wenxiao Gu

March 12, 2014

Contents

1 GEE (Generalized linear model) 2

1.1 Non-iterative way . . . 2
1.2 Iterative way 2.54 . . . 2
1.3 Fisher scoring 4.20 . . . 3
1.4 gee() function . . . 4
1.5 Summary . . . 4

2 GEE: marginal parameters 5

2.1 Model 1 with both age and income . . . 5
2.2 Model 1 difference in both Covariances . . . 6
2.3 Model 1 summary . . . 6
2.4 Model 2 summary . . . 6
2.5 Model 2 using Missing dataset . . . 6
2.6 Discussion on the difference in complete and missing datasets in Model 2 . . . 7

1 GEE (Generalized linear model)

1.1 Non-iterative way

Table 1 displays the results of the non-iterative way. We have the estimates of parameters and the 95% confidence intervals.

Table 1 – Point estimates obtained with robust standard errors and 95% confidence intervals

	Point est	Sandwich se	Lower CI	Upper CI
Intercept	22.7497	0.5668	21.6388	23.8606
I(age-8)	0.7695	0.1049	0.5638	0.9752
Sex	-1.5582	0.8156	-3.1569	0.0404
I(age-8)*Sex	-0.2857	0.1224	-0.5256	-0.0459
σ^2	4.9051			
ρ	0.6114			

1. The model is displayed

$$E[Y_{ij}|x_{ij}] \quad = \quad \beta_0 + \beta_1(age_{ij}\text{-}8) + \beta_0\left(1_{Female=1}\right) + \beta_0(age_{ij}\text{-}8) \times (1_{Female=1})$$

2. Estimates of β from OLS
   (a) $\tilde{\beta} = (X^T X)^{-1} XY$
   (b) The errors are assumed to follow multivariate normal with AR-1. Working covariance matrix is $V = \begin{bmatrix} 1 & \cdots & \alpha^3 \\ \alpha & & \\ \vdots & \ddots & \\ \alpha^3 & & 1 \end{bmatrix}$ as well as the weighting matrix $W = V^{-1}$ with the following estimates of the elements in the matrix,
      i. $\hat{\sigma}^2 = \frac{1}{nm}\Sigma_i^n \left(Y_i - X_i\tilde{\beta}\right)^T \left(Y_i - X_i\tilde{\beta}\right)$
      ii. $\hat{\alpha} = \frac{1}{\hat{\sigma}^2 n(m-1)}\Sigma_i^n\Sigma_j^{m-1} \left(Y_{ij} - X_{ij}\tilde{\beta}\right)^T \left(Y_{i(j+1)} - X_{i(j+1)}\tilde{\beta}\right)$
   (c) Estimates of β

$$\hat{\beta} \quad = \quad (X^T\hat{W}X)^{-1}X\hat{W}Y$$

   (d) Robust/empirical sandwich SEs (in 571 we assume correlation and hereby)

$$Cov(\hat{\beta}) \quad = \quad \left(X^T\hat{W}X\right)^{-1}\left[X^T\hat{W}\left(Y - X\hat{\beta}\right)\left(Y - X\hat{\beta}\right)^T\hat{W}X\right]\left(X^T\hat{W}X\right)^{-1}.$$

1.2 Iterative way 2.54

Table 2 displays the results of the iterative way. We have the estimates of parameters and the 95% confidence intervals.

Table 2 – Point estimates obtained with robust standard errors and 95% confidence intervals

	Point est	Sandwich se	Lower CI	Upper CI
Intercept	22.7503	0.5669	21.6391	23.8614
I(age-8)	0.7695	0.1050	0.5637	0.9752
Sex	-1.5589	0.8158	-3.1579	0.0401
I(age-8)*Sex	-0.2857	0.1224	-0.5256	-0.0458
σ^2	4.9106			
ρ	0.6135			

Repeat Part 1.1 until the convergence conditions. The convergence is achieved at 4th iteration.

1. The initial step is at *k=1*, $\hat{\beta}_1 = (X^T X)^{-1} XY$.
2. Working covariance matrix is $V_k = \begin{bmatrix} 1 & \cdots & \alpha^3 \\ \alpha & & \\ \vdots & \ddots & \\ \alpha^3 & & 1 \end{bmatrix}$ as well as the weighting matrix $W_k = V_k^{-1}$ with the following estimates of the elements in the matrix,
   (a) $\hat{\sigma}_k^2 = \frac{1}{nm} \Sigma_i^n \left(Y_i - X_i \hat{\beta}_k\right)^T \left(Y_i - X_i \hat{\beta}_k\right)$
   (b) $\hat{\alpha}_k = \frac{1}{\hat{\sigma}_k^2 n(m-1)} \Sigma_i^n \Sigma_j^{m-1} \left(Y_{ij} - X_{ij} \hat{\beta}_k\right)^T \left(Y_{i(j+1)} - X_{i(j+1)} \hat{\beta}_k\right)$
   (c) From the second step,$\hat{\beta}_{k+1} = (X^T \hat{W}_k X)^{-1} X \hat{W}_k Y$.
   (d) Robust/empirical sandwich SEs (in 571 we assume correlation and hereby)

$$Cov(\hat{\beta}_{k+1}) = \left(X^T \hat{W}_k X\right)^{-1} \left[X^T \hat{W}_k \left(Y - X\hat{\beta}_{k+1}\right) \left(Y - X\hat{\beta}_{k+1}\right)^T \hat{W}_k X\right] \left(X^T \hat{W}_k X\right)^{-1}.$$

1.3 Fisher scoring 4.20

Repeat Part 1.1 until the convergence conditions. The convergence is achieved at 3rd iteration.

1. The initial step is at *k=1*, $\hat{\beta}_1 = (X^T X)^{-1} XY$.
2. Working covariance matrix is $V_k = \begin{bmatrix} 1 & \cdots & \alpha^3 \\ \alpha & & \\ \vdots & \ddots & \\ \alpha^3 & & 1 \end{bmatrix}$ as well as the weighting matrix $W_k = V_k^{-1}$ with the following estimates of the elements in the matrix,
   (a)

$$\hat{\sigma}_k^2 = \frac{1}{nm-4} \Sigma_i^n \left(Y_i - X_i \hat{\beta}_k\right)^T \left(Y_i - X_i \hat{\beta}_k\right) \quad (1)$$

   (b)

$$\hat{\alpha}_k = \frac{1}{\hat{\sigma}_k^2 n\,(m-1)} \Sigma_i^n \Sigma_j^{m-1} \left(Y_{ij} - X_{ij} \hat{\beta}_k\right)^T \left(Y_{i(j+1)} - X_{i(j+1)} \hat{\beta}_k\right) \quad (2)$$

   (c) From the second step,$\hat{\beta}_{k+1} = \hat{\beta}_k + \left(X^T \hat{W}_k X\right)^{-1} \left[X^T \hat{W}_k \left(Y - X\hat{\beta}_k\right)\right]$.
   (d) Robust/empirical sandwich SEs (in 571 we assume correlation and hereby)

$$Cov(\hat{\beta}_{k+1}) = \left(X^T \hat{W}_k X\right)^{-1} \left[X^T \hat{W}_k \left(Y - X\hat{\beta}_{k+1}\right) \left(Y - X\hat{\beta}_{k+1}\right)^T \hat{W}_k X\right] \left(X^T \hat{W}_k X\right)^{-1}.$$

Table 3 displays the results of the Fisher scoring way. We have the estimates of parameters and the 95% confidence intervals. However, we have a problem in 1. If we use $\frac{1}{nm}$ instead of $\frac{1}{nm-4}$, we will get the numbers in Table 3a. If we keep $\frac{1}{nm-4}$, we will get the numbers in Table 3b. I believe that the numbers should be the same as the gee() function gives in part (d), but I am not sure why the numbers are different if we use $\frac{1}{nm-4}$, as given in the slides. I would guess that there is a typo in the slides, because

$\hat{\sigma}_k^2 = \frac{1}{nm-4}\Sigma_i^n \left(Y_i - X_i\hat{\beta}_k\right)^T \left(Y_i - X_i\hat{\beta}_k\right)$ should exist only in estimating σ^2 but not in the iteration, and $\hat{\sigma}_k^2 = \frac{1}{nm}\Sigma_i^n \left(Y_i - X_i\hat{\beta}_k\right)^T \left(Y_i - X_i\hat{\beta}_k\right)$ should be used to estimate coefficients in the iteration.

Table 3 – Point estimates obtained with robust standard errors and 95% confidence intervals

(a) $\frac{1}{nm}$

	Point est	Sandwich se	Lower CI	Upper CI
Intercept	22.7503	0.5669	21.6391	23.8614
I(age-8)	0.7695	0.1050	0.5637	0.9752
Sex	-1.5589	0.8158	-3.1579	0.0401
I(age-8)*Sex	-0.2857	0.1224	-0.5256	-0.0458
σ^2	4.9106			
ρ	0.6135			

(b) $\frac{1}{nm-4}$

	Point est	Sandwich se	Lower CI	Upper CI
Intercept	22.7443	0.5655	21.6359	23.8526
I(age-8)	0.7699	0.1047	0.5647	0.9751
Sex	-1.5522	0.8140	-3.1477	0.0432
I(age-8)*Sex	-0.2862	0.1221	-0.5256	-0.0469
σ^2	5.0989			
ρ	0.5906			

1.4 gee() function

Table 4 displays the results from gee. We have the estimates of parameters and the 95% confidence intervals.

Table 4 – Point estimates obtained with robust standard errors and 95% confidence intervals

	Point est	Sandwich se	Lower CI	Upper CI
Intercept	22.7503	0.5669	21.6391	23.8614
I(age-8)	0.7695	0.1050	0.5637	0.9752
Sex	-1.5589	0.8158	-3.1579	0.0401
I(age-8)*Sex	-0.2857	0.1224	-0.5256	-0.0458
σ^2	5.0995			
ρ	0.6135			

1.5 Summary

In the estimating equation, we have $D = X$. And thus, in the iteration step in part (b), we have the following

$$
\begin{aligned}
\hat{\beta}^{(k)} &= [D^T V(\hat{\alpha}^{(k-1)})^{-1} D]^{-1} \left[D^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right] \\
\hat{\beta}^{(k+1)} &= \hat{\beta}^{(k)} + [D^T V(\hat{\alpha}^{(k)})^{-1} D][D^T V(\hat{\alpha}^{(k)})^{-1}(Y - X\hat{\beta}^{(k)})] \\
&= [D^T V(\hat{\alpha}^{(k-1)})^{-1} D]^{-1} \left[D^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right] \\
&\quad + [D^T V(\hat{\alpha}^{(k)})^{-1} D]^{-1} \left\{ \left[D^T V(\hat{\alpha}^{(k)})^{-1}\right] \left[Y - X\left(\left[D^T V(\hat{\alpha}^{(k-1)})^{-1} D\right]^{-1} D^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right)\right]\right\} \\
&= [X^T V(\hat{\alpha}^{(k-1)})^{-1} X]^{-1} \left[X^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right] \\
&\quad + [X^T V(\hat{\alpha}^{(k)})^{-1} X]^{-1} \left\{ \left[X^T V(\hat{\alpha}^{(k)})^{-1}\right] \left[Y - X\left(\left[X^T V(\hat{\alpha}^{(k-1)})^{-1} X\right]^{-1} X^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right)\right]\right\} \\
&= \left[\left(X^T V(\hat{\alpha}^{(k-1)})^{-1} X\right)^{-1} \left(X^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right)\right] \\
&\quad + \left\{[X^T V(\hat{\alpha}^{(k)})^{-1} X]^{-1} [X^T V(\hat{\alpha}^{(k)})^{-1} Y]\right\} \\
&\quad - \left[XV(\hat{\alpha}^{(k)})^{-1} X\right]^{-1} \left[XV(\hat{\alpha}^{(k)})^{-1} X\right] \left[\left(X^T V(\hat{\alpha}^{(k-1)})^{-1} X\right)^{-1} \left(X^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \left[[X^T V(\hat{\alpha}^{(k-1)})^{-1} X]^{-1} [X^T V(\hat{\alpha}^{(k-1)})^{-1} Y]\right] \\
&\quad + \left\{[X^T V(\hat{\alpha}^{(k)})^{-1} X]^{-1} [X^T V(\hat{\alpha}^{(k)})^{-1} Y]\right\} \\
&\quad - \left[\left(X^T V(\hat{\alpha}^{(k-1)})^{-1} X\right)^{-1} \left(X^T V(\hat{\alpha}^{(k-1)})^{-1} Y\right)\right] \\
&= \left\{[X^T V(\hat{\alpha}^{(k)})^{-1} X]^{-1} [X^T V(\hat{\alpha}^{(k)})^{-1} Y]\right\} \\
&= \left\{[X^T W(\hat{\alpha}^{(k)}) X]^{-1} [X^T W(\hat{\alpha}^{(k)}) Y]\right\}
\end{aligned}
$$

Thus we have proved that why part (b) and part (c) are the same if we keep (nm) instead of (nm-4) in part (c). Moreover, we know *nm = 108*, and *nm-4 = 104*, and thus

$$
\begin{aligned}
\hat{\sigma}^2_{Fisher} &= \hat{\sigma}^2_{iter} \times \frac{nm}{nm-4} \\
&= 4.9106 \times \frac{108}{104} \\
&= 5.0995.
\end{aligned}
$$

Table 5 displays the outputs from the previous sub problems. We can see that the estimated coefficients except $\hat{\sigma}^2$ in part (d), (b) and (c) are the same, which is expected. Part (b) and (c) converge in a similar time. However, on the condition of $D \neq X$, the Fisher scoring method uses more (both current and previous steps) information on the derivatives and thus converges faster.

Table 5 – Point estimates obtained with robust standard errors and 95% confidence intervals

	Non-iterative way		Iterative way		Fisher		gee	
	Point est	Sandwich se	Point est	Sandwich se	Point est	Sandwich se	Point est	Sandwich se
Intercept	22.7497	0.5668	22.7503	0.5669	22.7503	0.5669	22.7503	0.5669
I(age-8)	0.7695	0.1049	0.7695	0.1050	0.7695	0.1050	0.7695	0.1050
Sex	-1.5582	0.8156	-1.5589	0.8158	-1.5589	0.8158	-1.5589	0.8158
I(age-8)*Sex	-0.2857	0.1224	-0.2857	0.1224	-0.2857	0.1224	-0.2857	0.1224
σ^2	4.9051		4.9106		4.9106	$(\frac{1}{nm})$	5.0995	$(\frac{1}{nm-4})$
ρ	0.6114		0.6135		0.6135		0.6135	

2 GEE: marginal parameters

2.1 Model 1 with both age and income

Table 6 displays the gee estimates for model (1).

	Independence Cov		Exchangeable Cov	
	Point est	Sandwich se	Point est	Sandwich se
(intercept)	0.576	0.117	0.525	0.115
age	-0.0487	0.0178	-0.0236	0.0176
income	-0.964	0.149	-0.945	0.145
age*income	0.0768	0.0225	0.062	0.0221

Table 6 – Point estimates and empirical/robust sandwich se for model (1)

2.2 Model 1 difference in both Covariances

- As GLS, the GEE should give consistent for the marginal regression parameter (fixed effects), no matter how the working covariance is defined. Moreover, the estimate of β_1 from GEE with independence working covariance is smaller than the one exchangable working covariance.
- The robust standard errors from both covariances are not far different.

2.3 Model 1 summary

- GEE with independence covariance

In the fluoride intake study, higher age was significantly associated with lower odds of fluoride intake more than 0.05 mg/kg (p-value = 0.006), and high maternal incomes (> 30 thousand dollars per year) was significantly associated with larger odds ratio comparing children with higher age to those with lower age (p-value = 0.0006).

We estimated that, on average within the study population of children with low maternal income (< 30 thousand dollars per year), the odds ratio that fluoride intake is greater than 0.05 mg/kg associated with one month older is 0.952 (95% CI: 0.920, 0.986). On average within the study population of children with high maternal income (> 30 thousand dollars per year), the odds ratio that fluoride intake is greater than 0.05 mg/kg associated with one month older is 1.03 (95% CI: 1.00, 1.06).

- GEE with exchangeable covariance

In the fluoride intake study, there was indication that higher age was associated with lower odds of fluoride intake more than 0.05 mg/kg. However, there was not enough evidence that this association was significant at $\alpha = 0.05$ (p-value = 0.181). High maternal incomes (> 30 thousand dollars per year) was significantly associated with higher odds ratio comparing children with higher age to those with lower age (p-value = 0.005).

We estimated that, on average within the study population of children with low maternal income (< 30 thousand dollars per year), the odds ratio that fluoride intake is greater than 0.05 mg/kg associated with one month older is 0.977 (95% CI: 0.944, 1.011). On average within the study population of children with high maternal income (> 30 thousand dollars per year), the odds ratio that fluoride intake is greater than 0.05 mg/kg associated with one month older is 1.04 (95% CI: 1.01, 1.07).

2.4 Model 2 summary

Table 7 displays the gee estimates for model (2).

	Independence Cov		Exchangeable Cov	
	Point est	Sandwich se	Point est	Sandwich se
(intercept)	-0.0245	0.0706	-0.0599	0.0689
age	-0.00228	0.0108	0.0154	0.0104

Table 7 – Point estimates and empirica;/robust sandwich se for model (2)

2.5 Model 2 using Missing dataset

Table 8 displays the gee estimates for model (2) with missing data.

	Independence Cov		Exchangeable Cov	
	Point est	Sandwich se	Point est	Sandwich se
(intercept)	-0.166	0.0778	-0.173	0.0751
age	0.00815	0.0120	0.0217	0.0115

Table 8 – Point estimates and empirical/robust sandwich se for model (2) with missing data

2.6 Discussion on the difference in complete and missing datasets in Model 2

The slope of age is from negative value to a positive value. We first want to find out which part of data are missing. We found that the missing data are from the low income group while the high income group is complete, as dislayed in Table 9. So the sandwich se gets larger due to smaller sample size. From part(a), we know that the age effect would be a combination of both age and income. In both covariances, we all have the following,

$$\begin{aligned} \beta_{age}^{(high\ income)} &= \beta_{age}^{(low\ income)} + \beta_{age \times income} \\ &= value(low\ income) < 0 + value > 0 \\ &= value > 0 \end{aligned}$$

But now, the lower income has missing data. So the effect from the lower income is partly missing, and thus $\beta_{age}^{(low\ income)}$<0 plays in smaller weight and $\beta_{age}^{(high\ income)}$>0 plays in larger weight, so that the value of estimated $\beta_{age}^{marginal}$ turns to be positive.

	Numtbers of rows in complete data set	Numbers of rows in missing dataset
Low income	1398	691
High income	2466	2466

Table 9 – Missing data

Appendix

```
## functions in problem 1
WORKING.AR1 = function(X, Y, b, m, Q) {
    n = length(Y)/m
    res = Y - X %*% b
    if (Q != "c") {
        sigmasq = sum((Y - X %*% b) * (Y - X %*% b))/(n * m)
    } else if (Q == "c") {
        df1 = (n * m - 4)
        df2 = (n * m)
        sigmasq = sum(res * res)/df1
    }
    sum_rho = c()
    for (i in 1:n) {
        for (j in 1:(m - 1)) {
            ind = m * (i - 1) + j
            # print(ind) slides
            y = Y[ind]
            y1 = Y[ind + 1]
            x = X[ind, ]
            x1 = X[ind + 1, ]
            sum_rho = c(sum_rho, (y - x %*% b) %*% (y1 - x1 %*% b))
        }
    }
    if (Q == "a") {
        alpha.hat = sum(sum_rho)/n/(m - 1)/sigmasq
    }
    if (Q == "b") {
        alpha.hat = sum(sum_rho)/n/(m - 1)/sigmasq
    }
    if (Q == "c") {
        ## df issue df = n
        alpha.hat = sum(sum_rho)/n/(m - 1)/sigmasq
    }

    V = replicate(n = n, matrix(0, m, m), simplify = FALSE)
    for (k in 1:n) {
        for (i in 1:m) {
            for (j in 1:m) {
                V[[k]][i, j] = alpha.hat^(abs(i - j))
            }
        }
    }
    V = as.matrix(bdiag(V)) * sigmasq
    # W = solve(V)
    return(V)
}

## function of sandwich
SANDWICH = function(X, Y, beta.lm, m, V) {
    b = beta.lm
    p = dim(X)[2]
    nm = dim(X)[1]
```

```
    n = nm/m
    W = solve(V)
    B = matrix(0, m, m)  #sigma over m
    for (i in 1:n) {
        ind = (m * (i - 1) + 1):(m * i)
        y = Y[ind]
        x = X[ind, ]
        res = y - x %*% b
        s = V[ind, ind]
        B = B + t(x) %*% solve(s) %*% res %*% t(res) %*% solve(s) %*% x
    }
    sand.var = solve(t(X) %*% W %*% X) %*% B %*% solve(t(X) %*% W %*% X)
    out = sqrt(diag(sand.var))  # robust/empirical sandwich se's
    return(out)
}

## main function
METHODS = function(data, Q) {
    data$sex = as.numeric(data$Sex) - 1
    if (Q == "d") {
        library(gee)
        fit = gee(distance ~ I(age - 8) * Sex, id = Subject, data = data, corstr = "AR-M",
            Mv = 1)
        out = summary(fit)$coef[, c(1, 4)]
        out = cbind(out, out[, 1] - 1.96 * out[, 2], out[, 1] + 1.96 * out[,
            2])
        names(fit)
        param = c(fit$scale, fit$working.correlation[1, 3]/fit$working.correlation[1,
            2])
    } else {
        Y = data$distance
        X = cbind(rep(1, nrow(data)), I(data$age - 8), data$sex, I(data$age -
            8) * data$sex)
        n = length(unique(data$Subject))
        m = nrow(data)/n
        b = matrix(coef(lm(Y ~ X))[-2], ncol = 1)
        beta.lm = b
        V = WORKING.AR1(X, Y, beta.lm, m, Q)
        W = solve(V)
        beta.estimate = solve(t(X) %*% W %*% X) %*% (t(X) %*% W %*% Y)
        sand.se = SANDWICH(X, Y, beta.estimate, m, V)
        sand.se
        para = c(V[1, 1], V[1, 2]/V[1, 1])
        if (Q == "a") {
            out = cbind(beta.estimate, sand.se)
            out = cbind(out, out[, 1] - 1.96 * out[, 2], out[, 1] + 1.96 * out[,
                2])
            param = para
        }
        if (Q == "c" | Q == "b") {
            # b,c
            beta.new = beta.estimate
            beta.iter = beta.estimate
```

```
            sand.iter = sand.se
            para.iter = para
            n.max = 10
            i = 1
            err = 1
            while (err > 1e-12 & i < n.max) {
                i = i + 1
                new.v = WORKING.AR1(X, Y, beta.new, m, Q)
                new.W = solve(new.v)
                if (Q == "b") {
                  beta.new = solve(t(X) %*% new.W %*% X) %*% (t(X) %*% new.W %*%
                    Y)
                } else if (Q == "c") {
                  beta.new = beta.new + solve(t(X) %*% new.W %*% X) %*% (t(X) %*%
                    new.W %*% (Y - X %*% beta.new))
                }
                para.new = c(new.v[1, 1], new.v[1, 2]/new.v[1, 1])
                sand.se.new = SANDWICH(X, Y, beta.new, m, new.v)
                beta.iter = cbind(beta.iter, beta.new)
                sand.iter = cbind(sand.iter, sand.se.new)
                para.iter = cbind(para.iter, para.new)
                err = sum((beta.iter[, i - 1] - beta.new)^2)
            }
            param = para.iter
            out = cbind(beta.iter[, dim(beta.iter)[2]], sand.iter[, dim(beta.iter)[2]])
            out = cbind(out, out[, 1] - 1.96 * out[, 2], out[, 1] + 1.96 * out[,
                2])
            # rownames(out) = paste(1:dim(out)[1],'iter')
        }
    }
    return(list(out = out, param = param))
}
```

```
## problem 1
library(nlme)
library(Matrix)
source("d:/Dropbox/Dropbox/571/GWXHW8/gwxhw8_FUNCS.R")
data = data.frame(Orthodont)
METHODS(data, "a")
METHODS(data, "b")
METHODS(data, "c")
METHODS(data, "d")

## problem 2
setwd("d:/Dropbox/Dropbox/571/GWXHW8/")
data = read.csv("fluoride.csv")
# e: use missing data
data = read.csv("fluoride.miss.csv")
# a
library(geepack)
library(gee)
a.independent = geeglm(fl ~ age * income, id = id, data = data, corstr = "independence",
    family = "binomial")
```

```r
a.independent2 = gee(fl ~ age * income, id = id, data = data, corstr = "independence",
    family = "binomial")
a.exchangeable = geeglm(fl ~ age * income, id = id, data = data, corstr = "exchangeable",
    family = "binomial")
a.exchangeable2 = gee(fl ~ age * income, id = id, data = data, corstr = "exchangeable",
    family = "binomial")

summary(a.independent)
summary(a.exchangeable)

coef.independent = summary(a.independent)$coefficients
# beta_age in low income grp
exp(coef.independent[2, 1])
# CI beta_age in low income grp
exp(coef.independent[2, 1] + c(-1, 1) * 1.96 * coef.independent[2, 2])
# beta_age in high income grp
exp(coef.independent[2, 1] + coef.independent[4, 1])
# CI beta_age in high income grp
exp(coef.independent[2, 1] + coef.independent[4, 1] + c(-1, 1) * 1.96 * sqrt(c(0,
    1, 0, 1) %*% a.independent2$robust.variance %*% c(0, 1, 0, 1)))
coef.exchangeable = summary(a.exchangeable)$coefficients
# beta_age in low income grp
exp(coef.exchangeable[2, 1])
# CI beta_age in low income grp
exp(coef.exchangeable[2, 1] + c(-1, 1) * 1.96 * coef.exchangeable[2, 2])
# beta_age in high income grp
exp(coef.exchangeable[2, 1] + coef.exchangeable[4, 1])
# CI beta_age in high income grp
exp(coef.exchangeable[2, 1] + coef.exchangeable[4, 1] + c(-1, 1) * 1.96 * sqrt(c(0,
    1, 0, 1) %*% a.exchangeable2$robust.variance %*% c(0, 1, 0, 1)))

# b
b.independent = geeglm(fl ~ age, id = id, data = data, corstr = "independence",
    family = "binomial")
b.exchangeable = geeglm(fl ~ age, id = id, data = data, corstr = "exchangeable",
    family = "binomial")
summary(b.independent)
summary(b.exchangeable)
coef.independent = summary(b.independent)$coefficients
# beta_age in low income grp
exp(coef.independent[2, 1])
# CI beta_age in low income grp
exp(coef.independent[2, 1] + c(-1, 1) * 1.96 * coef.independent[2, 2])
# beta_age in high income grp
exp(coef.independent[2, 1] + coef.independent[4, 1])
# CI beta_age in high income grp
exp(coef.independent[2, 1] + coef.independent[4, 1] + c(-1, 1) * 1.96 * sqrt(c(0,
    1, 0, 1) %*% b.independent2$robust.variance %*% c(0, 1, 0, 1)))
coef.exchangeable = summary(b.exchangeable)$coefficients
# beta_age in low income grp
exp(coef.exchangeable[2, 1])
# CI beta_age in low income grp
exp(coef.exchangeable[2, 1] + c(-1, 1) * 1.96 * coef.exchangeable[2, 2])
```

```r
# beta_age in high income grp
exp(coef.exchangeable[2, 1] + coef.exchangeable[4, 1])
# CI beta_age in high income grp
exp(coef.exchangeable[2, 1] + coef.exchangeable[4, 1] + c(-1, 1) * 1.96 * sqrt(c(0,
    1, 0, 1) %*% b.exchangeable2$robust.variance %*% c(0, 1, 0, 1)))

# f
data.comp = read.csv("fluoride.csv")
data = read.csv("fluoride.miss.csv")
table(data.comp$income == 1)
table(data$income == 1)
```