BIOSTAT 571 Homework 3

Wenxiao Gu

January 30, 2014

Contents

1 Problem 1

1.1 Part(a)

- The true mean model is as following,

$$Y_{ij} = f(x_{i1}) + \beta_L(x_{ij} - x_{i1}) + \epsilon_{ij}$$

- The model we use is as following,

$$EY_{ij} = \beta_0 + \beta_C x_{i1} + \beta_L(x_{ij} - x_{i1})$$

- A true mean model will produce unbiased parameter estimates. If if we want the estimated coefficient β_L to be unbiased, then we have to check if the following holds,

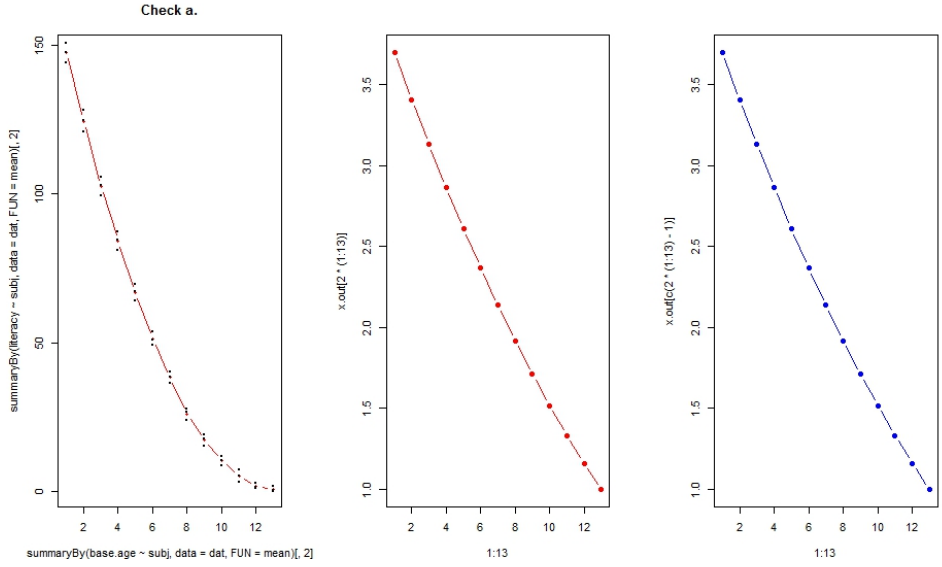
$$f(x_{i1}) = \beta_0 + \beta_C x_{i1}.$$

1.2 Part(b)

- We hope to make x_{i1} to be precision variable. Thus the coefficient of it would not be changed. (And also the variance of its coefficient would be smaller.)
- So we need to check if x_{i1} and the other covariate $x_{ij} - x_{i1}$ are independent.
- If space of $x_{ij} - x_{i1}$ for all subjects is equal, then the columns of the design matrix are orthogonal, and produce the unbiased coefficient β_L.

1.3 Part(c)

- To check (a), we plot literacy scores against baseline age
- We take the delta age difference between $j = 0$ and $j = 1$ and the difference between $j = 1$ and $j = 2$. for each i. I draw the two plots separately. There are 13 points for each plot.
- The following plots shows the results for (a) and (b) with our data. The data are not linear in baseline age, so the sucient condition of part (a) is not satisfied. We also observe that the delta age is not equally spaced for each subject.



- In part(b), the two covariates are orthognal, and thus the response projected onto the column space of the design matrix would give the estimated \beta_L exactly the same as the true \beta_L.

1.4 Part(d)

- The model 2 is below,

$$EY_{ij} \quad = \quad \alpha_i + \beta_L(x_{ij} - x_{i1}).$$

- With the model above, we do not have to specify $f(x_{i1})$.
- The results are in the table below. The confidence interval of Fixed effect model is narrower, because of a smaller confidence interval. But two models have similar estimates.

Model	Estimate	95% CI	SD of $\hat{\beta}_L$
Partitioned exposure model	$\hat{\beta}_L$= 1.25	(-0.84, 3.33)	1.06
Fixed effect model	$\hat{\beta}_L$= 1.01	(0.94 1.07)	0.034

1.5 Part(e) Bias

In all simulations, we assume the β_L = 1.5 , and σ = 1. The partitioned exposure models fail to produce unbiased estimates of L, if neither the mean model correctly specied nor the fixed linear design. There are 4 settings for x_{i1}. $f(x_{i1}) = log(x_{i1}),\ x_{i1},\ x_{i1}^2,\ or\ x_{i1}^3$.

In time interval $(x_{ij} - x_{i1}) = \begin{cases} 2 + 0.5 \times g(x_{i1}), & j = 2 \\ 4 + 0.5 \times g(x_{i1}), & j = 3 \end{cases}$, $g(x_{i1}) = 0,\ x_{i1},\ or\ x_{ii}^2$. $g(x_{i1})$ depends on how design the mean model. In other words,

$$g(x_{i1}) = \begin{cases} 0 & equally\ spaced \\ x_{i1} & linear \\ x_{i1}^2 & quardrtic. \end{cases}$$

The table of bias (*0.001) is following,

$f(x_{i1})$	x_{i1} **part(a)**	x_{i1} **part(a)**	x_{i1}^2	x_{i1}^2	x_{i1}^3	x_{i1}^3	$log(x_{i1})$	$log(x_{i1})$
$g(x_{i1})$	Exposure partition	**Fixed effect**	partition	**Fixed effect**	partition	**Fixed effect**	partition	**Fixed effect**
0 part(b)	0.68	.68	0.632	0.632	3.69	3.69	2.12	2.12
x_{i1}	2.47	2.47	1.67	1.67	-1.19	-1.19	1.05	1.05
x_{i1}^2	0.092	0.132	*464*	-0.25	*1152*	-0.064	942	-0.66

- Exposure partitioncan fail to give unbiased estimates of beta_L without at least one of the additional conditions from parts (a) and (b).
- On the other hand, if either works, they produce the unbiased estimates, in the first column and in the first row.
- The fixed effect model will work regardless.
- Hence, we can see in the table, all biases with at least column or row bolded are small.

1.6 Part(f) Coverage and Standard Error

$f(x_{i1})$	x_{i1} **part(a)**	x_{i1} **part(a)**	x_{i1}^2	x_{i1}^2	x_{i1}^3	x_{i1}^3	$log(x_{i1})$	$log(x_{i1})$
$g(x_{i1})$	Exposure partition	**Fixed effect**	partition	**Fixed effect**	partition	**Fixed effect**	partition	**Fixed effect**
0 part(b)	0.943	0.940	0.986	0.944	0.99	0.942	1	0.9422
x_{i1}	0.932	0.946	0.99	0.941	0.97	0.944	1	0.943
x_{i1}^2	0.939	0.942	0.99	0.941	0.99	0.944	0.99	0.942

- If both (a) and (b) are met, the coverage would be accurate.

- If we use fixed effect model, the coverage would be accurate.
- If only part (b) is met, we can see that the coverage in exposure partition model which is misspecified is overestimated.

Using sandwich estimate from vcov() function would not change the coverage.

1.7 Part(g)

In Biost 570, we learnt that sandwich standard error will overestimate when model is wrong and x is fixed. Since the value of delta age is fixed when the follow-up time is equally spaced, when the exposure partition model misspecifies the truth, we will have a too large coverage.

2 Problem 2

2.1 Part(a)

The point estimate (se) from by-hand optimization is 1.006 (0.027); the one from **lm()**, as in (d), is 1.006 (0.034). We can see that the point estimates are close, while the standard error from optimization is smaller than the one from **lm()**.

2.2 Part(b)

The standard error estimate from optimization is smaller than the one from **lm()**. Since in the simulation in problem 1(f), the OLS gives close to 95% CI coverage, the smaller estimate from direct optimization is not valid. I believe OLS one.

2.3 Part(c)

The MLE of standard error for $\hat{\beta_L}$ is $\hat{\sigma^2_{MLE}} = \frac{1}{N}\sum_{i=1}^{N}[\frac{1}{n_i}\sum_{j=1}^{n_i} residual^2]$. The OLS gives unbiased estimate $\hat{\sigma_{OLS}} = \frac{1}{N}\sum_{i=1}^{N}[\frac{1}{n_i-1}\sum_{j=1}^{n_i} residual^2]$ with degrees-of-freedom correction. When $n_i = 3$, $\hat{\sigma^2_{MLE}} = \frac{3-1}{3}\hat{\sigma^2_{OLS}}$. In fact we can check that $\frac{0.0335}{0.0274} \approx \frac{2}{3}$.

2.4 Part(d)

The problem in (b) is the Neymann-Scott problem, where the degrees of freedom is not adjusted. Since the estimator from MLE (optimization) and OLS (lm()) are off by $\frac{n_i-1}{n_i}$, if we increase number of observations per subject (n_i), the difference goes to 1.

3 Problem 3

$$\begin{aligned} E(\tilde{\beta}_W) &= E[(x^T W^{-1} x)^{-1} x^T W^{-1} Y] \\ &= (x^T W^{-1} x)^{-1} x^T W^{-1} E(Y) \\ &= (x^T W^{-1} x)^{-1} x^T W^{-1} (x^T \beta) \\ &= \beta \end{aligned}$$

. Thus $E(\tilde{\beta}_W) = \beta$, and $\tilde{\beta}_W$ is unbiased.

In homework 4 of Biostat 570, we have the proofs. If we have the following,

$$Var(\hat{\beta}_W) = (x^T W^{-1} x) x^T W^{-1} V W^{-1} x (x^T W^{-1} x)^{-1}$$

$$Var(\hat{\beta}_V) \quad = \quad (x^T V^{-1} x)^{-1}$$

.

Then let $G = G(W) = (x^T W^{-1})(Y - \mu)$ be the estimating equation with weight matrix W^{-1}, $\mu = x^T \beta$. Also let $U = G(V) = (x^T V^{-1})(Y - \mu)$.

Then $E(G^T G) = (x^T W^{-1}) V W^{-1} x$, $E(G^T U) = x^T W^{-1} x$ and $E(U^T U) = x^T V^{-1} x$.

Then let $M = \begin{bmatrix} E(G^T G) & E(G^T U) \\ E(U^T G) & E(U^T U) \end{bmatrix}$.

We have two conditions, as below

- (1) $E(U^T U) = x^T V^{-1} x$ where $V = Var(Y)$, is positive definite;
- (2) $E(G^T G) = (x^T W^{-1}) V W^{-1} x$ is symmetric and square.

Thus by Schur Complement Lemma, $E(G^T G) - E(G^T U) E(U^T U) E(U^T G)$ is positive semidefinite. In other words,$\forall a \in R^m$, $a^T [E(G^T G) - E(G^T U) E(U^T U) E(U^T G)] a \geqslant 0$.

If $a = (x^T W^{-1} x) a'$, then

$$\begin{aligned} a^T [E(G^T G) - E(G^T U) E(U^T U) E(U^T G)] (x^T W^{-1} x) a &= a'^T [(x^T W^{-1} x) x^T W^{-1} V W^{-1} x (x^T W^{-1} x) - (x^T V^{-1} x)^{-1}] a' \\ &= a'^T [Var(\hat{\beta}_W) - Var(\hat{\beta}_V)] a' \\ &\geqslant 0 \end{aligned}$$

.

Therefore, $Var(\hat{\beta}_W) \geqslant Var(\hat{\beta}_V)$. The proof is done.

Appendix

```
## pr1
dat = read.csv("computer-data-hw3.csv")
head(dat)
library(doBy)
par(mfrow = c(1, 3), mar = c(4, 4, 4, 4))
plot(summaryBy(base.age ~ subj, data = dat, FUN = mean)[, 2], summaryBy(literacy ~
    subj, data = dat, FUN = mean)[, 2], type = "b", col = "red", pch = 16, cex = 0.7,
    main = "Check a.")
points(dat$base.age, dat$literacy, pch = 19, cex = 0.5)

## follow-up times = total age

summaryBy(delta.age ~ subj + base.age, data = dat, FUN = mean)

x.s = c(dat$delta.age[-1], 0) - dat$delta.age
x.out = x.s[-3 * (1:13)]
length(x.out)
x.out
plot(1:13, x.out[2 * (1:13)], pch = 19, type = "b", col = "red")
plot(1:13, x.out[c(2 * (1:13) - 1)], pch = 19, type = "b", col = "blue")

# d)

# 1st proposal
model1 = lm(literacy ~ base.age + delta.age, data = dat)
summary(model1)$coef[3, 1] + c(0, -1, 1) * qnorm(0.975) * summary(model1)$coef[3,
    2]

# 2nd regreesion
model2 = lm(literacy ~ 0 + delta.age + factor(subj), data = dat)
summary(model2)$coef[1, 1] + c(0, -1, 1) * qnorm(0.975) * summary(model2)$coef[1,
    2]

## alternative way
model2a = lm(literacy ~ delta.age + factor(subj), data = dat)
summary(model2a)$coef[2, 1] + c(0, -1, 1) * qnorm(0.975) * summary(model2a)$coef[2,
    2]


# e)
fit.0  # without condition
fit.a  # with (a)
fit.b  # with (b)
fit.c  # the fixed effect model

do.one = function(beta.L, n_subject, m_time) {
    # n_subject = 5 m_time = 3
    subj = rep(1:n_subject, each = m_time)
    subj
    base.age = subj

    follow.per = rep(1:m_time, n_subject)
```

```
follow.per

for (i in 1:(m_time)) {
    # print(i)
    delta[c(m_time * (0:(n_subject - 1)) + i)] = (i - 1) * 2
}

delta
delta.2 = delta
delta.3 = delta

for (i in 2:(m_time)) {
    print(i)
    tmr1 = delta[c(m_time * (0:(n_subject - 1)) + i)]
    delta.2[c(m_time * (0:(n_subject - 1)) + i)] = tmr1 + 0.5 * base.age[c(m_time *
        (0:(n_subject - 1)) + i)]
    delta.3[c(m_time * (0:(n_subject - 1)) + i)] = tmr1 + 0.5 * (base.age[c(m_time *
        (0:(n_subject - 1)) + i)])^2
}

# cbind(subj,base.age,delta,delta.lin,delta.nonlin, yL,yQ,yC,ylog,
# yL2,yQ2,yC2,ylog2, yL3,yQ3,yC3,ylog3, follow.per)
n.row = n_subject * m_time
# beta.L = 1.5
fL = 0.5 + base.age
fQ = 0.5 + base.age^2
fC = 0.5 + base.age^3
flog = log(base.age)

yL = rnorm(n.row, 0, 1) + beta.L * (delta) + fL
yQ = rnorm(n.row, 0, 1) + beta.L * (delta) + fQ
yC = rnorm(n.row, 0, 1) + beta.L * (delta) + fC
ylog = rnorm(n.row, 0, 1) + beta.L * (delta) + flog

yL2 = rnorm(n.row, 0, 1) + beta.L * (delta.2) + fL
yQ2 = rnorm(n.row, 0, 1) + beta.L * (delta.2) + fQ
yC2 = rnorm(n.row, 0, 1) + beta.L * (delta.2) + fC
ylog2 = rnorm(n.row, 0, 1) + beta.L * (delta.2) + flog

yL3 = rnorm(n.row, 0, 1) + beta.L * (delta.3) + fL
yQ3 = rnorm(n.row, 0, 1) + beta.L * (delta.3) + fQ
yC3 = rnorm(n.row, 0, 1) + beta.L * (delta.3) + fC
ylog3 = rnorm(n.row, 0, 1) + beta.L * (delta.3) + flog



# beta.L Linear########################################################## 1st
# proposal
model.L1P = lm(yL ~ base.age + delta)
model.L2P = lm(yL ~ base.age + delta.2)
model.L3P = lm(yL ~ base.age + delta.3)

# summary(model1)$coef[3, 1] + c(0, -1,
# 1)*qnorm(0.975)*summary(model1)$coef[3, 2]
```

```
# 2nd regreesion
model.L1F = lm(yL ~ 0 + factor(base.age) + delta)
model.L2F = lm(yL ~ 0 + factor(base.age) + delta.2)
model.L3F = lm(yL ~ 0 + factor(base.age) + delta.3)

################################### Quar########################################################## 1st propo
model.Q1P = lm(yQ ~ base.age + delta)
model.Q2P = lm(yQ ~ base.age + delta.2)
model.Q3P = lm(yQ ~ base.age + delta.3)



# 2nd regreesion
model.Q1F = lm(yQ ~ 0 + factor(base.age) + delta)
model.Q2F = lm(yQ ~ 0 + factor(base.age) + delta.2)
model.Q3F = lm(yQ ~ 0 + factor(base.age) + delta.3)

###################################

################## Cubic########################################################## 1st proposal
model.C1P = lm(yC ~ base.age + delta)
model.C2P = lm(yC ~ base.age + delta.2)
model.C3P = lm(yC ~ base.age + delta.3)

# summary(model1)$coef[3, 1] + c(0, -1,
# 1)*qnorm(0.975)*summary(model1)$coef[3, 2]

# 2nd regreesion
model.C1F = lm(yC ~ 0 + factor(base.age) + delta)
model.C2F = lm(yC ~ 0 + factor(base.age) + delta.2)
model.C3F = lm(yC ~ 0 + factor(base.age) + delta.3)

###################################

################## Cubic########################################################## 1st proposal
model.log1P = lm(ylog ~ base.age + delta)
model.log2P = lm(ylog ~ base.age + delta.2)
model.log3P = lm(ylog ~ base.age + delta.3)

# summary(model1)$coef[3, 1] + c(0, -1,
# 1)*qnorm(0.975)*summary(model1)$coef[3, 2]

# 2nd regreesion
model.log1F = lm(ylog ~ 0 + factor(base.age) + delta)
model.log2F = lm(ylog ~ 0 + factor(base.age) + delta.2)
model.log3F = lm(ylog ~ 0 + factor(base.age) + delta.3)

###################################


mo = c("F", "P")
na = c("L", "Q", "C", "log")
num = 1:3
```

```
    mm.co = matrix(0, nrow = 24, ncol = 2)
    l = 0
    for (k in 1:2) {
        for (i in 1:4) {
            for (j in 1:3) {
                tmr = eval(parse(text = paste("model.", na[i], num[j], mo[k],
                  sep = "")))
                l = l + 1
                mm.co[l, ] = c(summary(tmr)$coef[length(tmr$coef), 1:2])



            }
        }
    }
    cov = bias = c()
    for (i in 1:24) {
        tmr = mm.co[i, 1] + mm.co[i, 2] * c(-1, 1) * 1.96
        cov[i] = as.numeric(tmr[1] < beta.L & beta.L < tmr[2])
    }
    bias = rep(beta.L, 24) - mm.co[, 1]
    out = cbind(cov, bias)
    out
    return(out)
}
# h)
do.one(1.5, 6, 4)

## simulation
sims = function(n.sims = 50, beta.L, n_subject, m_time) {
    out.tmr = t(replicate(n.sims, do.one(beta.L, n_subject, m_time)[, 2]))
    return(output)
}

n.sims = 500
head(sims(n.sims, x, sigma, rho, b0, b1))
apply(out, 1, mean)
out.bias = replicate(n.sims, do.one(beta.L, n_subject, m_time)[, 2])
apply(out.bias, 1, mean)[13:24]


## pr3
dat = read.csv("computer-data-hw3.csv")
attach(dat)
# Design Matrix
X = model.matrix(~factor(dat$base.age))
X = cbind(dat$delta.age, dat$literacy, X)

# log likkelihood function
l = function(par, X) {
    X = as.matrix(X)
    l = dnorm(X[, 2], mean = par[1] * X[, 1] + X[, 3:ncol(X)] %*% as.vector(par[2:(12 +
        2)]), sd = par[15], log = T)
```

```
    return(-sum(l))
}
optim = nlm(l, c(1, 140, seq(-20, -140, length.out = 12), 0.5), X = X, hessian = T)
# point estimate
optim$est[1]
# se estimate
sqrt(diag(solve(optim$hessian)))[14]
```