STAT/BIOSTAT 571: Homework 4 Key

1. **[Diagnostics for GEE]** *For the GEE regressions you implemented in HW3 Q2, give diagnostic plots or plots addressing the influence of individual clusters, the validity of the assumed mean model, and the appropriateness of the working covariance assumptions. For each plot, briefly state what is shown, and (writing for a non-statistician) advise whether you think inference on the non-interpret parameters should be re-evaluated.*

   Answer: To address the influence of individual clusters,use a leave-one-out analysis to look at how the non-intercept estimates $\hat{\beta}_i$'s and the robust intervals change when omitting one cluster at a time. Figure 1 suggests cluster 18 and 19 are slightly more influential than others for estimating β_1. Figure 2 suggests cluster 10 and 26 are slightly more influential than others for estimating β_2. Figure 3 suggests cluster 13 is slightly more influential than others for estimating β_3.

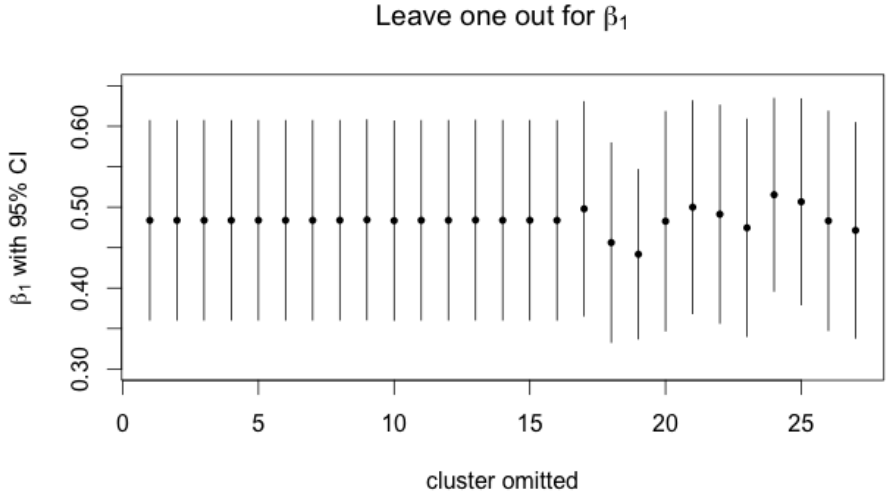


Figure 1: Leave one out for estimation of β_1 and robust 95% interval.

See Figure 4 which shows both the raw and Pearson residuals versus the fitted values from the fit. There do not seem to be any problems with the mean model here; the lowess smooth of the data do not reveal any noticeable departure from the mean model. The residuals are nicely centered around zero across the range of the fitted values.

Figure 5 is an attempt at assessing the working covariance assumption. The horizontal axis are the residuals at the earliest time point. The vertical axis shows the residuals at lags 3, 2, and 1 (moving from left to right). The gray boxes at the top of the plot show the GEE-estimated correlation between residuals at each lag. The dotted line shows the fitted relationship between the lagged residuals, while the solid line shows the relationship between the residuals as estimated by GEE. We see that at lag 1, the actual and estimated relationships are nearly identical; we get less agreement at longer lags but we also have fewer data at those lags. Overall, the AR1 assumption doesn't appear too egregious; and recall that we only check this for efficiency's sake, anyway.

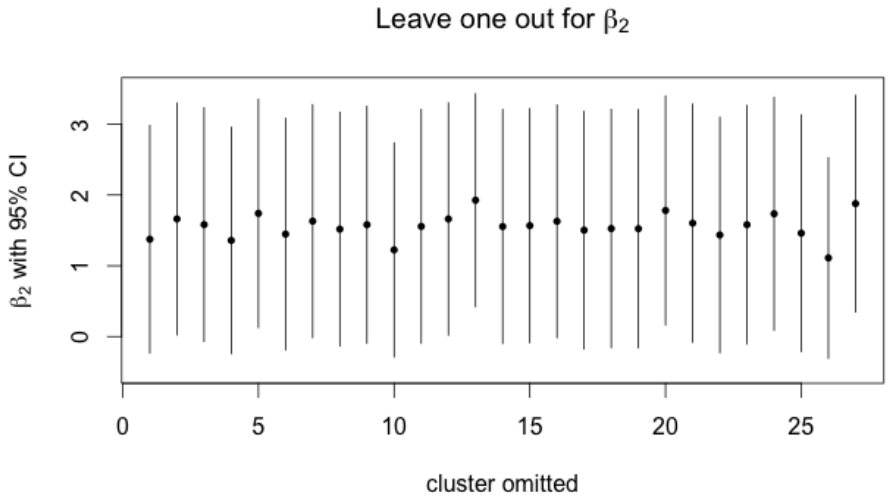


Figure 2: Leave one out for estimation of β_2 and robust 95% interval.

2. [**Coverage of intervals in GEE**] *Suppose the data generating mechanism is;*

$$\begin{aligned} n_i &\sim Pois(3)+1, \text{ for } 1 \le i \le n \\ Z_{ij} &\overset{i.i.d.}{\sim} \Gamma(3,2), \text{ for } 1 \le j \le n_i, 1 \le i \le n \\ X_{ij} &= \sum_{j'=1}^{j} Z_{ij'} \\ b_i &\overset{i.i.d.}{\sim} N(0,\tau^2) \\ \mu_{ij} &= b_i + \beta_0 + \beta_1 X_{ij} \\ Y_{ij}|X_{ij},b_i &\overset{ind}{\sim} N(\mu_{ij},\sigma^2) \end{aligned}$$

Here, the n_i are from a shifted Poisson distribution, and $\Gamma(3,2)$ indicates the Gamma distribution with `shape=3, rate=2`*; the cumulative exposure X_{ij} might be seen in e.g. a dosing experiment.*

For $n = 50, \sigma = 1, \tau = 1, \beta_0 = 0, \beta_1 = 0.8$, implement GEE for linear regression of Y on X, using independence working correlation, and estimate the true coverage of robust intervals for β_1 that have nominal 95% coverage. You should find that the coverage is not exactly 95%; what properties of the data generating mechanism (and hence the behavior of $\hat{\beta}_1$ and its estimated standard error) play the biggest role in the discrepancy? Explain your answer using your simulation results.

Answer: For this model, I had a coverage of 91% from 10,000 simulations. Simulations with larger n did show that the correct coverage of 95% was indeed being approached (by $n = 500$ the actual coverage was at the nominal level). See Figure 6 (especially the histogram of $\hat{\beta}_1$ and the QQ-plot of $\hat{\beta}_1$): note that the asymptotic assumption of normally distributed $\hat{\beta}_1$ appears to hold with $n = 50$, so that is not a problem here. However, this data generating mechanism generates covariates X_{ij} which are very skewed (a histogram of a couple realizations of the $\{X_{ij}\}$ is enough to show us this). This, along with the variability in the cluster size, creates estimates of $SE(\hat{\beta}_1)$ that are somewhat sensitive to extreme values of the covariate. The lower right-hand plot of Figure 6 shows that the variability in the estimated standard errors has not shrunk to a size that

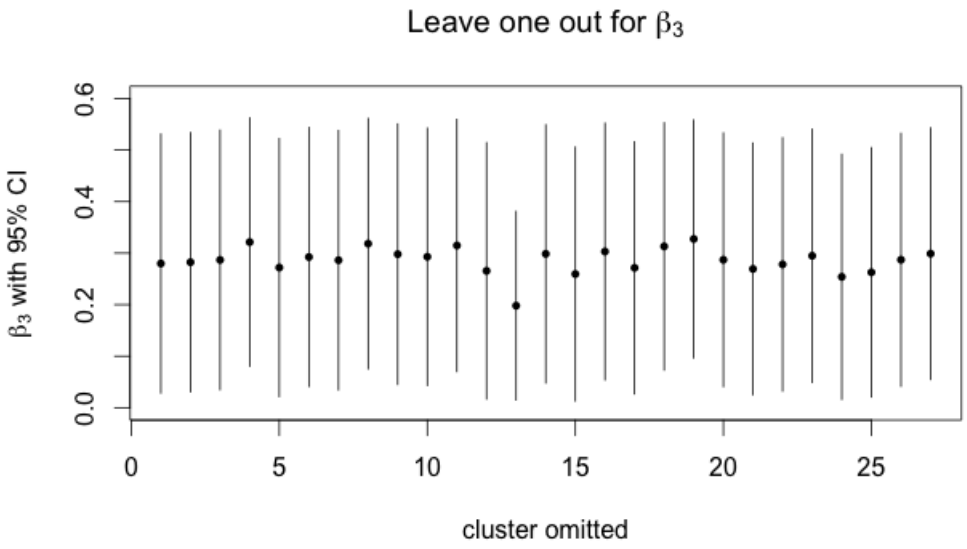


Figure 3: Leave one out for estimation of β_3 and robust 95% interval

is negligible relative to the variability of $\hat{\beta}_1$ for this n (we know from theory that this negligible variance in the estimated SE's relative to $\hat{\beta}$ is important for the asymptotics to work). Figure 7 shows this shrinkage for a much larger n. In addition, no matter what n is we are not "saved" by a correlation between $\hat{\beta}_1$ and its estimated standard error. That is, for large values of $\hat{\beta}_1$, we do not tend to get larger values of the estimated standard error. This makes the asymptotics kick in more slowly, and we've see that $n = 50$ is not enough for the asymptotics to have taken effect (though they do kick in eventually).

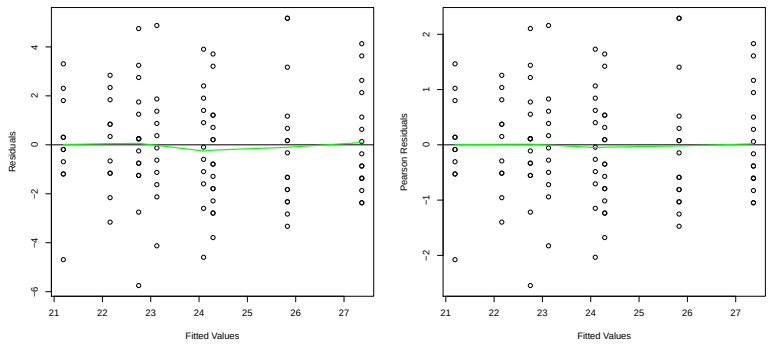


Figure 4: Raw and Pearson residuals vs. the fitted values from the GEE fit to the dental data

3. [**Mean model assumptions in GEE**] This question illustrates the impact of a mis-specified mean model, in a simple setting where GEE might be used. Suppose data are generated as follows;

$$\begin{aligned} x_{ij} &= j,\ 1 \le j \le 4 \\ \mu_{ij} &= \beta_1 x_{ij} + \beta_2 \mathbf{1}_{\{x_{ij}=2\}} - \beta_2 \mathbf{1}_{\{x_{ij}=3\}} \\ Y_{ij} &\overset{indept}{\sim} N(\mu_{ij}, 1) \end{aligned}$$

where $n_i = 4$ for all clusters, $\beta_1 = 5$, and $\mathbf{1}_A$ denotes the indicator of event A. Assuming that data are used in GEE linear regression of Y on x, i.e. with just a linear term in x as well as the intercept;

(a) For use of Independent and AR-1 working correlation matrices, plot the limiting values of the estimated 'slope' returned by GEE, for $\beta_2 \in [-5, 5]$. For different β_2, explain any discrepancies between use of Independence/AR-1; why is one limiting value bigger or smaller than the other? Why are they equal?

The limiting values are plotted below;

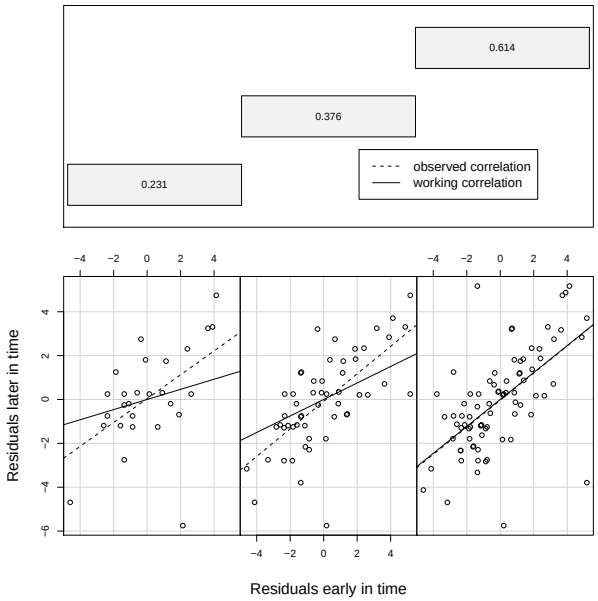
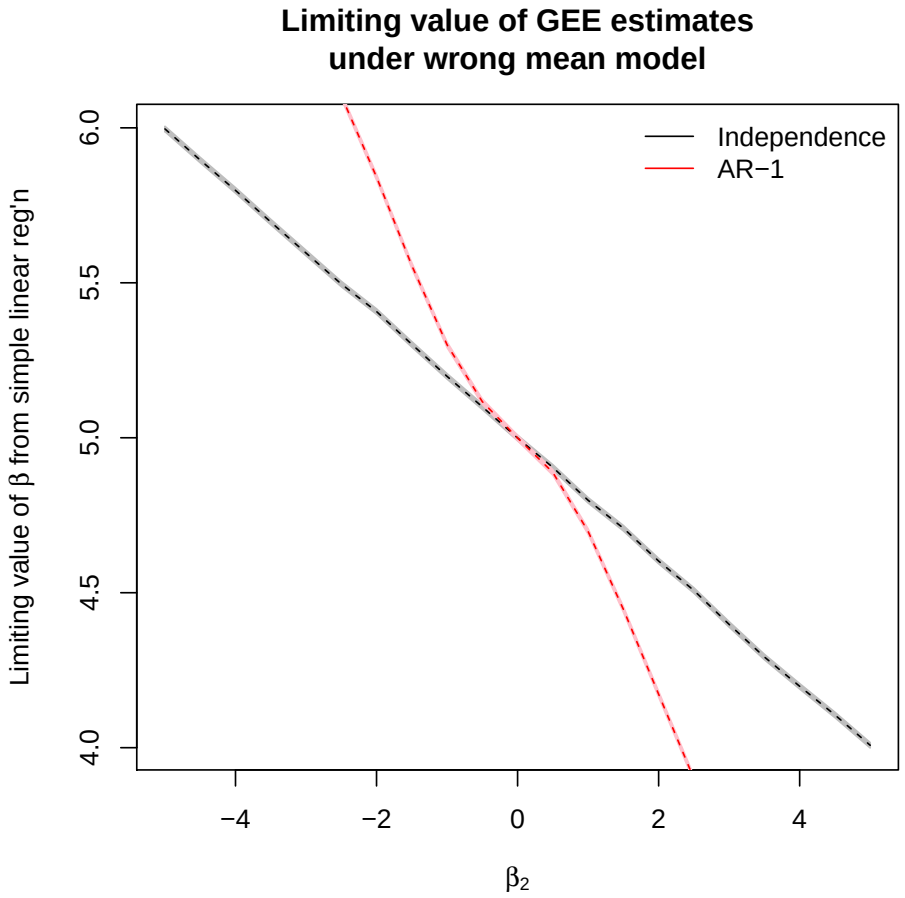


Figure 5: Actual and estimated association of residuals at different lags



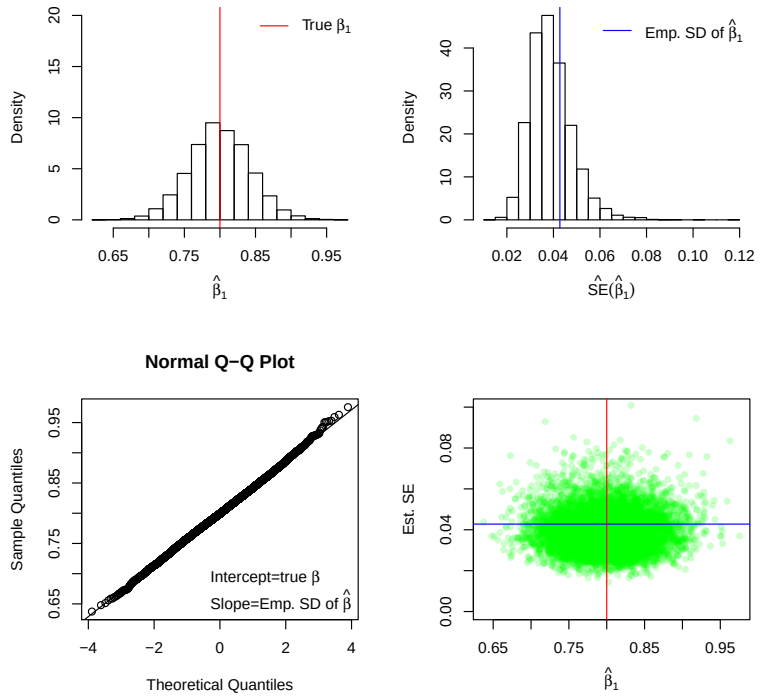


Figure 6: Behavior of $\hat{\beta}_1$ and its estimated standard error; $n = 50$

These are approximated by taking $n = 10,000$ clusters, and the (narrow) intervals around the line correspond to the nominal 95% intervals returned by GEE. (The exact values can also be established without Monte Carlo methods, but this does not affect anything we do in this question.) Of course, we see considerable bias compared to $\beta_1 = 5$, but more importantly the value for which we are consistent depends on the chosen working correlation matrix. This is important, because the primary advantages of GEE over other methods is that, when the mean model is correct, then this choice does not affect validity of inference. When the mean model is mis-specificed, then the choice of working correlation determines the parameter that is being consistently estimated, so this appealing robustness property no longer holds.

In the plot we see that the working AR-1 version's estimated slope is higher than the version using working Independence when $\beta_2 < 0$, and lower when $\beta_2 > 0$. We know that the working Independence estimator is equivalent to OLS, and thus (from 570) represents a weighted average of $\Delta(Y)/\Delta(X)$ over all pairs of observations. In large samples, the AR-1 version always fits $\hat{\alpha} < 0$, corresponding to the alternating signs of the β_2 terms at $x = 2$ and $x = 3$. How this affects the weights can be determined by looking at the inverse of the working correlation matrix;

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha & 1 & \alpha & \alpha^2 \\ \alpha^2 & \alpha & 1 & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}^{-1} \propto \begin{pmatrix} 1-2\alpha^2+\alpha^4 & -\alpha+2\alpha^3-\alpha^5 & 0 & 0 \\ -\alpha+2\alpha^3-\alpha^5 & 1-\alpha^2-\alpha^4+\alpha^6 & -\alpha+2\alpha^3-\alpha^5 & 0 \\ 0 & -\alpha+2\alpha^3-\alpha^5 & 1-\alpha^2-\alpha^4+\alpha^6 & -\alpha+2\alpha^3-\alpha^5 \\ 0 & 0 & -\alpha+2\alpha^3-\alpha^5 & 1-2\alpha^2+\alpha^4 \end{pmatrix}$$

The point here is not the exact values (which Mathematica produced) but that there are zeros

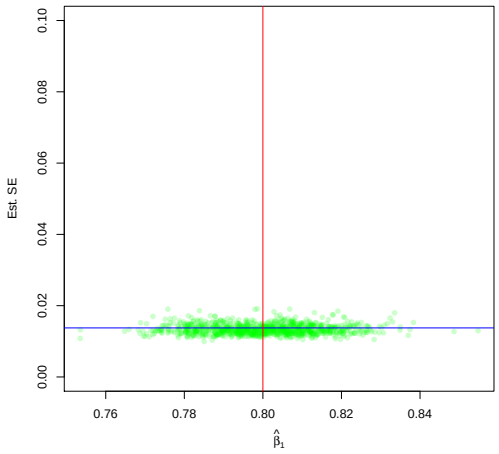


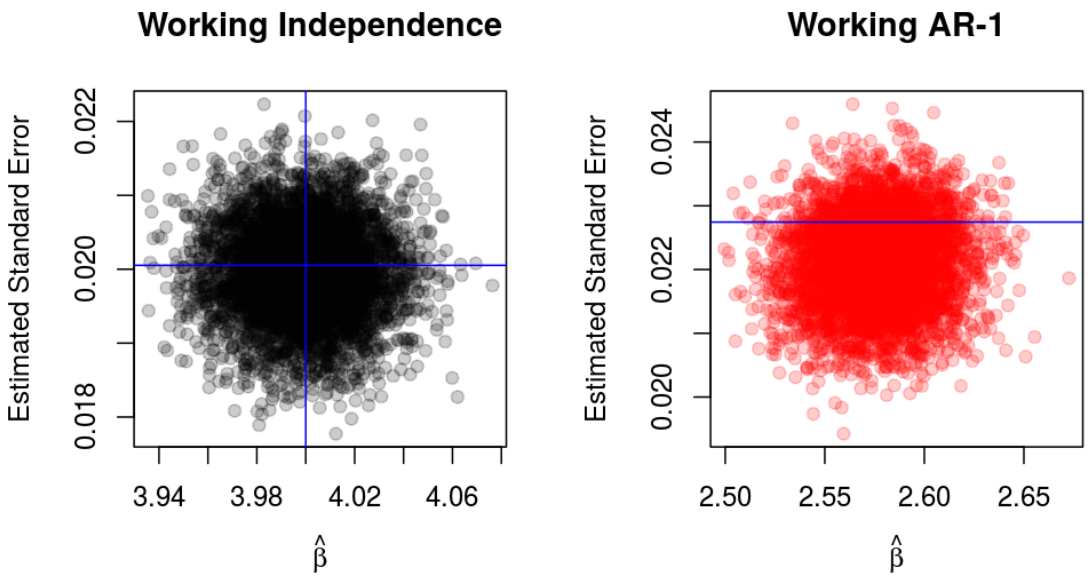
Figure 7: Behavior of $\hat{\beta}_1$ and its estimated standard error; $n = 500$

in the positions at greater than lag 1. We see that each lag 1 pair receives the same weight, observations at greater lags receive zero weight. Thus for positive β_2, with a large negative difference between observations 2 and 3, the AR-1 estimator results in a lower estimate than weighting all observations equally. For $\beta_2 < 0$ this situation is reversed, giving the pattern seen here.

(b) *In the situations in a), do confidence intervals from GEE 'work'? That is, in large samples do they give nominal coverage of the slope parameter value for which they are consistent? Justify your answer.*

In short, the AR-1 intevals do not work unless the model is correctly specified, i.e. unless $\beta_2 = 0$. The key step in showing this analytically occur on slide 3.47, where the 'bread' matrix we use in GEE assumes two terms are zero; this assumption is incorrect unless the mean model is accurate, giving inaccurate coverage. As an example, when considering $\beta_2 = 5$, we obtain (approximately) 92.2% coverage using working AR-1. Because its residuals are orthogonal to the design matrix, the working Independence does give correct coverage; a more direct way to see this is that, working through the full non-parametric sandwich estimate, we get the same answers as with GEE.

These properties are illustrated in simulations below;



Here, $n = 500, \beta_1 = 5, \beta_2 = 5$. The blue lines indicate the true value being estimated, and the true standard error of the estimate. There is no problem with bias, or with the standard error estimates being non-ignorably noisy. Instead, the AR-1 estimates of standard error are typically too small, leading to undercoverage.

R code

```
#===========
#Question 1
#===========

gee3 <- gee(distance~I(age-8)*Sex,id=Subject,data=Orthodont,corstr='AR-M')

pdf('HW5_residplot1.pdf')
with(dent.fit,plot(fitted.values,residuals,xlab='Fitted Values',ylab='Residuals'))
abline(h=0)
with(dent.fit,lines(lowess(fitted.values,residuals,iter=0),lwd=2,col='green'))
dev.off()

pdf('HW5_residplot2.pdf')
with(dent.fit,plot(fitted.values,residuals/sqrt(scale),
    xlab='Fitted Values',ylab='Pearson Residuals'))
abline(h=0)
with(dent.fit,lines(lowess(fitted.values,residuals/scale,iter=0),lwd=2,col='green'))
dev.off()


# helper function for making the coplot
panel.func <- function(x,y,col,pch, ...) {
points(x,y)
abline(lm(y~x), lty=2)
cor = round(cor(x,y),1)
if (cor == .5)
abline(0, .231)
else if (cor==.7)
abline(0, .376)
else
abline(0, .614)
}
# residuals
res = gee3$residuals
# get appropriate combinations of residuals ready for co-plot
mat.1=cbind(res[-seq(4, 108, by=4)],
res[-seq(1, 108, by=4)],
rep(.614, length(res[-seq(1, 108, by=4)])))

mat.2=cbind(res[c(seq(1, 108, by=4), seq(2,108, by=4))],
res[c(seq(3, 108, by=4), seq(4,108, by=4))],
rep(.376, length(res[c(seq(3, 108, by=4), seq(4,108, by=4))])))

mat.3=cbind(res[seq(1, 108, by=4)],
```

```
res[seq(4, 108, by=4)],
rep(.231, length(res[seq(4, 108, by=4)])))

pdf('HW5_coplot.pdf')
cor.mat = rbind(mat.1, mat.2, mat.3)
cor.dt = data.frame(cor.mat)
colnames(cor.dt) = c("y", "x", "alpha")
coplot(y~x|as.factor(alpha), data=cor.dt, rows=1, panel=panel.func,
xlab=c("Residuals early in time", ""), ylab=c("Residuals later in time"))
legend(2.2, 2.5, legend=c("observed correlation", "working correlation"), lty=c(2,1),
cex=.8)
dev.off()
#===========
#Question 2
#===========


do.one <- function(n) {
 ni <- rpois(n,3)+1
 ind <- rep(1:n,ni)
 Z <- rgamma(sum(ni),3,2)
 X <- unlist(by(Z,ind,cumsum))
 b <- rep(rnorm(n),ni)
 mu <- b + 0 + 0.8*X
 Y <- rnorm(sum(ni),mu,1)
 gee.fit <- gee(Y~X,corstr='independence',id=ind)
 ci.cov <-  (gee.fit$coef[2]-0.8)^2/(gee.fit$robust.variance[2,2])<=3.84
 list('coverage'=ci.cov,'bhat'=gee.fit$coef[2],'SE'=sqrt(gee.fit$robust.variance[2,2]))
}

many <- replicate(10000,do.one(50))
many2 <- replicate(1000,do.one(500))

mean(unlist(many[1,]))

pdf('HW5_q3.pdf')
par(mfrow=c(2,2))
hist(unlist(many[2,]),main='',xlab=expression(hat(beta)[1]),freq=FALSE,ylim=c(0,20))
abline(v=0.8,col='red')
legend('topright',expression(paste('True ',beta[1])),lty=1,col='red',bty='n')
hist(unlist(many[3,]),main='',xlab=expression(hat(SE)(hat(beta)[1])),freq=FALSE)
abline(v=sd(unlist(many[2,])),col='blue')
legend('topright',expression(paste('Emp. SD of ',hat(beta)[1])),col='blue',lty=1,bty='n')
qqnorm(unlist(many[2,]))
abline(0.8,sd(unlist(many[2,])))
legend('bottomright',c(expression(paste('Intercept=true ',beta)),
         expression(paste('Slope=Emp. SD of ',hat(beta)))),bty='n')
```

```
plot(many[2,],many[3,],ylim=c(0,0.1),pch=16,col=rgb(0,1,0,alpha=0.2),
     ylab='Est. SE',xlab=expression(hat(beta)[1]))
abline(h=sd(unlist(many[2,])),col='blue')
abline(v=0.8,col='red')
dev.off()

pdf('HW5_q3ii.pdf')
plot(many2[2,],many2[3,],ylim=c(0,0.1),pch=16,col=rgb(0,1,0,alpha=0.2),
     ylab='Est. SE',xlab=expression(hat(beta)[1]))
abline(h=sd(unlist(many2[2,])),col='blue')
abline(v=0.8,col='red')
dev.off()


###
### Q3
### Behavior of GEE with wrong mean model
###

library("gee")

do.one <- function(beta, n=1000){
xvals <- 1:4
d <- data.frame(x=rep(xvals, n))
d$y <- with(d, rnorm(length(xvals)*n, 0 + 5*x + beta*(x==2) - beta*(x==3), 1))
d$id <- rep(1:n, each=length(xvals))
gee1 <- gee(y~x, data=d, id=id, corstr="independence")
gee3 <- gee(y~x, data=d, id=id, corstr="AR-M", Mv=1)
gee1
c(
gee1$coeff[2], sqrt(diag(gee1$robust.variance))[2],
gee3$coeff[2], sqrt(diag(gee3$robust.variance))[2],
gee3$working.correlation[1,2])
}

set.seed(4)
t1 <- sapply(seq(-10,10,l=21), function(beta){c(beta, do.one(beta, n=10000))} )

pdf("hw6fig8.pdf", w=6, h=6)
plot(t(t1[1:2,]), type="n", xlim=c(-5,5),
xlab=expression(beta[2]), ylab=expression("Limiting value of "*hat(beta)*" from simple linear reg'

polygon(x=c(t1[1,], rev(t1[1,])),
y=c(t1[2,]-1.96*t1[3,], rev(t1[2,]+1.96*t1[3,])),
density=NA, col="gray"
)
polygon(x=c(t1[1,], rev(t1[1,])),
```

```
y=c(t1[4,]-1.96*t1[5,], rev(t1[4,]+1.96*t1[5,])),
density=NA, col="pink"
)
lines(t(t1[1:2,]), type="l", lwd=1, lty=2) # Ind betahat vs beta2
lines(t(t1[c(1,4),]), col=2, lwd=1, lty=2) # AR1 betahat vs beta2
legend("topright", lty=1, col=1:2, c("Independence","AR-1"), bty="n")
title("Limiting value of GEE estimates\nunder wrong mean model")
dev.off()

# plot of the fitted alphas, for AR-1 (not shown)
plot(t(t1[c(1,6),]), type="l")

# some example inverted correlation matrices -- note the zeros

round(solve(outer(1:4, 1:4, function(i,j){(-0.1)^abs(i-j)})), 3)
round(solve(outer(1:4, 1:4, function(i,j){(-0.3)^abs(i-j)})), 3)
round(solve(outer(1:4, 1:4, function(i,j){(-0.6)^abs(i-j)})), 3)
round(solve(outer(1:4, 1:4, function(i,j){(-0.99)^abs(i-j)})), 3)

# simulations for coverage, using beta2=-5,0,5

# "truth" from earlier simulations;
t1[,c(6,11,16)]

set.seed(16)
simm5 <- replicate(1000, do.one(-5, n=500) )
sim0 <- replicate(1000, do.one(0, n=500) )
sim5 <- replicate(5000, do.one(5, n=500) )

mean( ((simm5[1,]-t1[2,6])/simm5[2,])^2 < qchisq(0.95, df=1) )
mean( ((simm5[3,]-t1[4,6])/simm5[4,])^2 < qchisq(0.95, df=1) )

mean( ((sim0[1,]-t1[2,11])/sim0[2,])^2 < qchisq(0.95, df=1) )
mean( ((sim0[3,]-t1[4,11])/sim0[4,])^2 < qchisq(0.95, df=1) )

mean( ((sim5[1,]-t1[2,16])/sim5[2,])^2 < qchisq(0.95, df=1) )
mean( ((sim5[3,]-t1[4,16])/sim5[4,])^2 < qchisq(0.95, df=1) )

# truth using Independence;

tru <- function(beta1, beta2){
X <- cbind(rep(1, 4), 1:4)
EY <- beta1*(1:4) + c(0,beta2, -beta2, 0)
solve(t(X) %*% X) %*% t(X) %*% EY
}
tru(5, 5)
tru(5, -5)
```

```
sapply(seq(-10,10,l=21), function(b2){ tru(5, b2)[2,1]} )
t1[2,]

mean( ((sim5[1,]-tru(5,5)[2,1])/sim5[2,])^2 < qchisq(0.95, df=1) )
mean( ((sim5[3,]-t1[4,16])/sim5[4,])^2 < qchisq(0.95, df=1) )

png("hw6fig9.png", w=7*144, h=4*144, res=72*144/72)
par(mfrow=c(1,2))
plot(sim5[1,], sim5[2,], pch=19, col="#00000033",
xlab=expression(hat(beta)), ylab="Estimated Standard Error",
main="Working Independence")
abline(h=sd(sim5[1,]), lty=1, col="blue")
abline(v=tru(5,5)[2,1], lty=1, col="blue")
plot(sim5[3,], sim5[4,], pch=19, col="#FF000033",
xlab=expression(hat(beta)), ylab="Estimated Standard Error",
main="Working AR-1")
abline(h=sd(sim5[3,]), lty=1, col="blue")
abline(v=t1[4,16], lty=1, col="blue")
dev.off()
```