Lecture 11 Handout: Models for Clustered Data

Elizabeth Colantuoni

2/28/2021

I. Objectives

Upon completion of this session, you will be able to do the following:

- Implement a weighted least squares regression analysis to account for correlated data or heteroskedasticity
- Obtain robust variance estimates following implementation of an ordinal least squares or weighted least squares analysis
- Describe the generalized estimating equations approach and know that weighted least squares is a special case of this estimation procedure

II. Weighted Least Squares (Review)

Consider a longitudinal design where you have $i = 1, ..., m$ subjects and $j = 1, ..., n_i$ observations for each subject. The data are (Y_{ij}, X_{ij}). The goal of the analysis is to describe the mean of Y_ij using a linear model, $E(Y_{ij}|X_{ij}) = X_{ij}^{'}\beta$.

In Lecture 10, we discussed the issues around estimation of β and making inference for β using ordinary least squares; namely, if we ignore the correlation structure,

1. Estimation of β is unbiased
2. Standard error estimates for $\hat{\beta}$ are biased.

NOTE: In Lab 6, you will explore these findings via a simulation study.

We discussed two solutions:

1. Weighted least squares, where we specify a model for both $X\beta$ and $Var(Y) = \Sigma$, where Σ is a block diagonal matrix containing $Var(Y_i) = V_i(\theta)$ on the diagonal elements and 0 otherwise (i.e. people are independent). θ are the parameters that define the structure of V_i. Estimates of β and θ are obtained via an iterative procedure (get initial values of β, estimate θ, update β, update θ, etc.)
2. Robust variance estimation, where we fit the model assuming independence (i.e. we get unbiased estimates of β) and then adjust the standard errors using a robust variance estimate.

Note that robust variance estimation is a good idea even if we specify a working model for $V_i(\theta)$ because we may be wrong!

In this lecture, we will demonstrate how to fit weighted least squares using the *gls* function in R and then describe how to fit models and obtain robust variance estimates in R using the *gee* package.

III. Implementing Weighted Least Squares

We will consider a hypothetical longitudinal study, where we enrolled 60 children from Nepal ages 1 to 5 months. For each child, we will measure weight at enrollment and then every 4 months for 4 follow-up assessments.

The goal of the analysis is to describe / estimate the monthly increase in weight as a function of age.

A. Exploratory analysis

We will explore three main features of the data:

- the mean model
- the correlation structure
- the variance structure

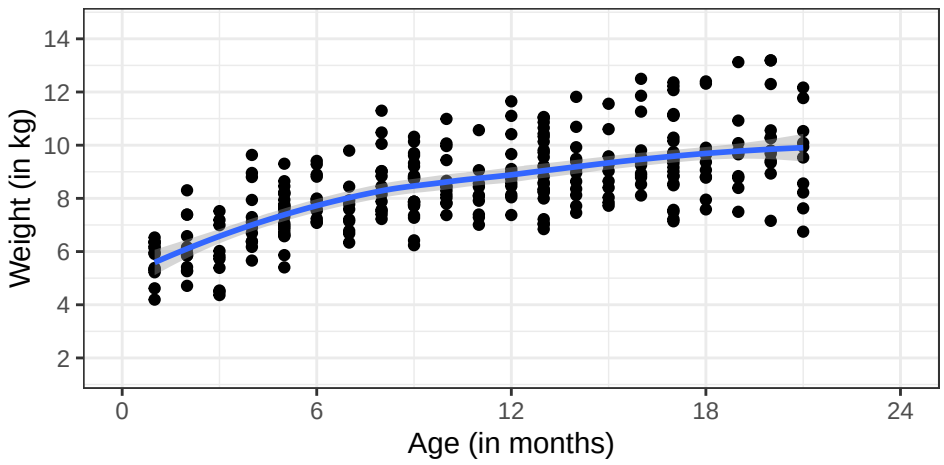
1. Exploration of the mean model

```
load("nepal_simulated.rda")
```

To assess the general trends in the data, we will make both a scatter plot and spaghetti plot. The scatterplot allows us to assess how the population mean weight changes as a function of age. Whereas the spaghetti plot allows us to look at variation between/across children and also get a sense of fluctuations or trends in weight within a child over time.

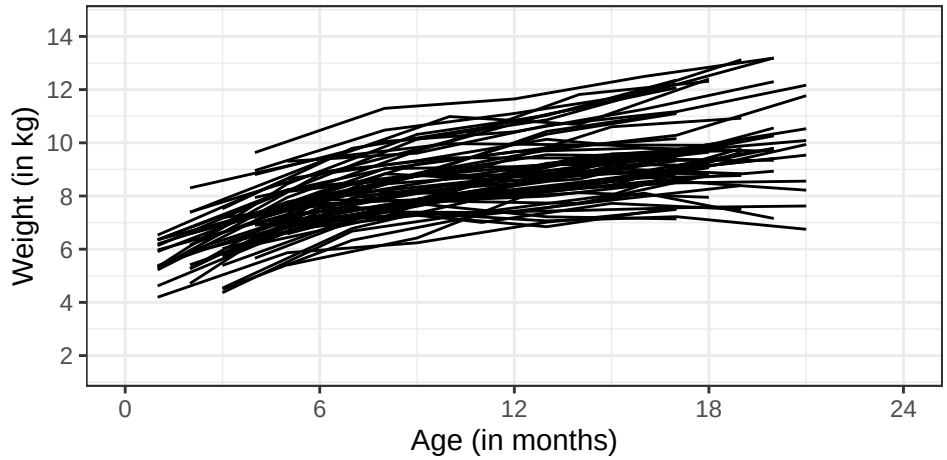
```
ggplot(data = nepal1, aes(x = age, y = wt)) +
  geom_point() + theme_bw() +
  geom_smooth() +
  labs(y="Weight (in kg)",x="Age (in months)") +
  scale_y_continuous(breaks=seq(2,14,2),limits=c(1.5,14.5)) +
  scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data = nepal1, aes(x = age, y = wt, group = factor(id))) +
  geom_line() + theme_bw() +
  labs(y="Weight (in kg)",x="Age (in months)") +
```

```
  scale_y_continuous(breaks=seq(2,14,2),limits=c(1.5,14.5)) +
  scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

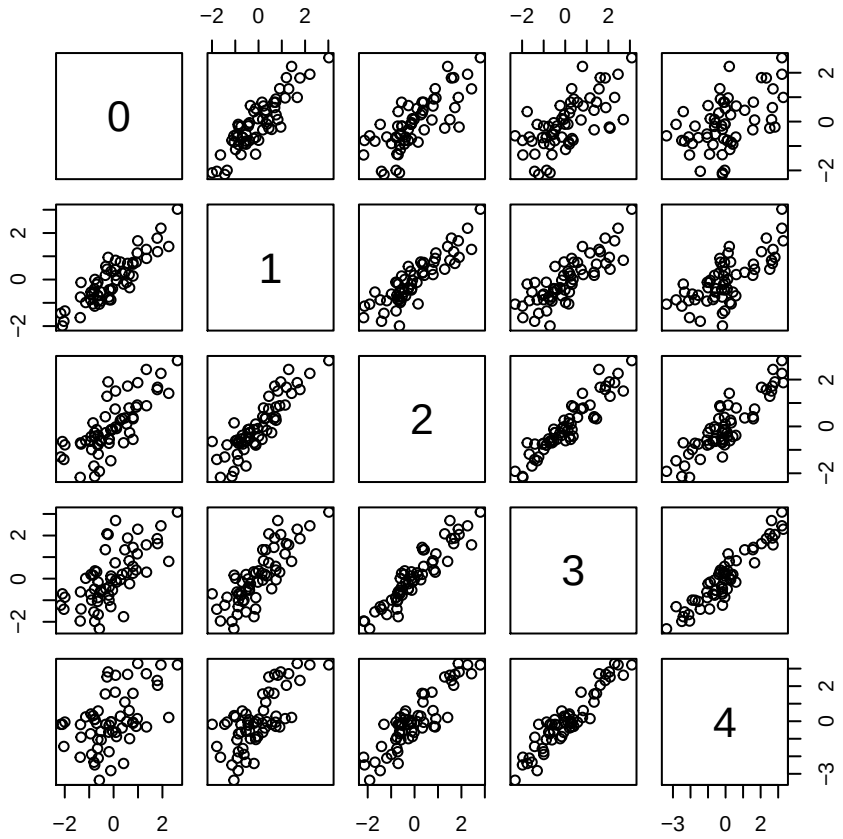


What do you notice about the data? Can you describe some patterns you observe?

2. Explore the correlation structure

Next, we will explore the correlation structure in the data by computing $Corr(r_{ij}, r_{ik})$ where r are residuals from a linear spline model assuming a single knot at 6 months of age.

```
## Here you need to get the set of residuals and then look at the correlation between residuals at the
nepal1$residuals = residuals(lm(wt~age+age_sp6,data=nepal1))
nepal1_wide = nepal1 %>% select(id, fuvisit, residuals) %>% spread(fuvisit,residuals)
pairs(nepal1_wide[,-1])
```




```
cor(nepal1_wide[,-1])
```

```
##           0         1         2         3         4
## 0 1.0000000 0.8785910 0.7595446 0.6685220 0.4958610
## 1 0.8785910 1.0000000 0.8814021 0.8246947 0.7181845
## 2 0.7595446 0.8814021 1.0000000 0.9350597 0.8890514
## 3 0.6685220 0.8246947 0.9350597 1.0000000 0.9389962
## 4 0.4958610 0.7181845 0.8890514 0.9389962 1.0000000
```

Can you look at the table of correlation estimates and provide a rough estimate for the autocorrelation function?

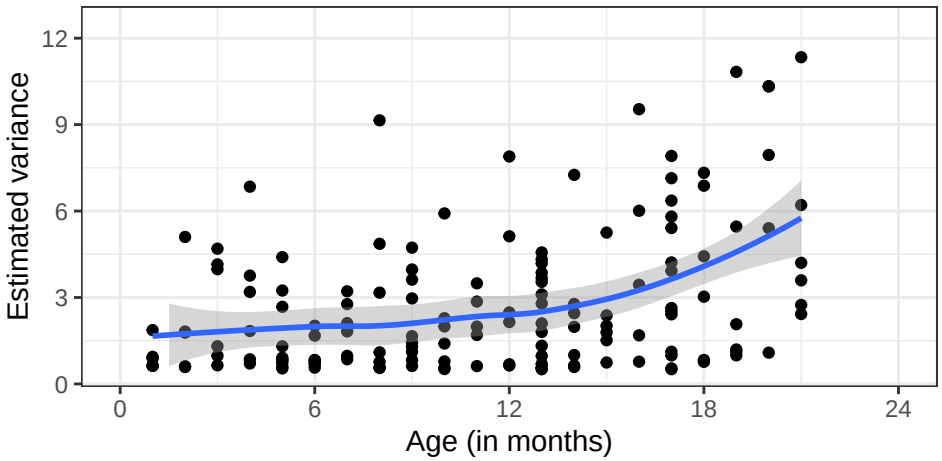
3. Explore the variance structure

In addition to understanding the correlation structure, we need understand if the variance in the residuals is the same at all ages or the variance of the residuals changes over age.

```
ggplot(nepal1,aes(x=age,y=residuals^2)) +
    geom_point() + geom_smooth() + theme_bw() +
    labs(y="Estimated variance",x="Age (in months)") +
    scale_y_continuous(breaks=seq(0,12,3),limits=c(0.5,12.5)) +
  scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 146 rows containing non-finite values (stat_smooth).
## Warning: Removed 146 rows containing missing values (geom_point).
```



4. Summary of the exploratory analysis

From the exploratory analysis, comment on the trends you observed in the data:

- What is the general trend in the mean weight as a function of age? Propose a model for the mean.
- Are the observations independent? Propose a model for the correlation.
- Are the observations homoscedastic? Propose a model for the variance.

B. Generalized least squares (GLS)

In this section, we will fit several weighted least squares models. Weighted least squares are also known as "generalized least squares".

1. Assume constant variance but correlated residuals

First we will consider two correlation models:

- `corCompSymm` gives the exchangeable correlation structure
- `corSymm` gives the unstructured correlation

```
library(nlme)
# Exchangeable correlation structure
mod.gls.exch = gls(wt ~ age + age_sp6, data = nepal1,correlation=corCompSymm(form=~1|id))
summary(mod.gls.exch)
```

```
## Generalized least squares fit by REML
##   Model: wt ~ age + age_sp6
##   Data: nepal1
##        AIC     BIC    logLik
##   729.3473 747.816 -359.6737
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
```

```
##  Parameter estimate(s):
##       Rho
## 0.7706506
##
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  4.915432 0.19908885  24.68964       0
## age          0.511532 0.02921332  17.51022       0
## age_sp6     -0.365864 0.03414815 -10.71402       0
##
##  Correlation:
##         (Intr) age
## age     -0.665
## age_sp6  0.610 -0.979
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.77916143 -0.60321814 -0.09831765  0.58609803  2.63663993
##
## Residual standard error: 1.230864
## Degrees of freedom: 300 total; 297 residual
```

```
summary(mod.gls.exch)$tTable
```

```
##                  Value  Std.Error   t-value      p-value
## (Intercept)  4.9154323 0.19908885  24.68964 6.029971e-74
## age          0.5115317 0.02921332  17.51022 1.183383e-47
## age_sp6     -0.3658639 0.03414815 -10.71402 7.308832e-23
```

```
# Unstructured
mod.gls.uns = gls(wt ~ age + age_sp6, data = nepal1,correlation=corSymm(form=~1|id))
summary(mod.gls.uns)
```

```
## Generalized least squares fit by REML
##   Model: wt ~ age + age_sp6
##   Data: nepal1
##        AIC      BIC    logLik
##   563.7659 615.4781 -267.8829
##
## Correlation Structure: General
##  Formula: ~1 | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3     4
## 2 0.944
## 3 0.891 0.938
## 4 0.829 0.893 0.951
## 5 0.637 0.746 0.863 0.917
##
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  4.870561 0.17956594  27.12408       0
## age          0.501658 0.01717792  29.20363       0
## age_sp6     -0.349676 0.01944934 -17.97882       0
##
```

```
##  Correlation:
##         (Intr) age
## age     -0.270
## age_sp6  0.037 -0.853
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.38070896 -0.46445346 -0.04842459  0.55233727  2.28023407
##
## Residual standard error: 1.432917
## Degrees of freedom: 300 total; 297 residual
```

```
summary(mod.gls.uns)$tTable
```

```
##                 Value  Std.Error   t-value      p-value
## (Intercept)  4.8705612 0.17956594  27.12408 2.325036e-82
## age          0.5016575 0.01717792  29.20363 2.682774e-89
## age_sp6     -0.3496761 0.01944934 -17.97881 2.064560e-49
```

2. Assume non-constant variance and correlated residuals

What about heteroscedasticity? Use the `weights` argument in `gls` to account fo heteroscedasticity.

```
mod.gls.exch.het = gls(wt ~ age + age_sp6,
    data = nepal1,correlation = corCompSymm(form = ~1|id),
    weights = varFunc(~age))
summary(mod.gls.exch.het)$tTable
```

```
##                 Value  Std.Error   t-value       p-value
## (Intercept)  5.1161159 0.08188405  62.48001 6.626584e-173
## age          0.4684656 0.02484724  18.85382  1.092163e-52
## age_sp6     -0.3137424 0.02959251 -10.60209  1.745166e-22
```

```
mod.gls.uns.het = gls(wt ~ age + age_sp6,
    data = nepal1,correlation=corSymm(form=~1|id),
    weight=varFunc(~age))
summary(mod.gls.uns.het)
```

```
## Generalized least squares fit by REML
##   Model: wt ~ age + age_sp6
##   Data: nepal1
##        AIC      BIC    logLik
##   604.5766 656.2889 -288.2883
##
## Correlation Structure: General
##  Formula: ~1 | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3     4
## 2 0.751
## 3 0.674 0.953
## 4 0.625 0.934 0.975
## 5 0.552 0.896 0.959 0.975
## Variance function:
##  Structure: fixed weights
```

```
##  Formula: ~age
##
## Coefficients:
##                 Value  Std.Error   t-value p-value
## (Intercept)  5.099711 0.09917412  51.42179       0
## age          0.460847 0.03011970  15.30053       0
## age_sp6     -0.317750 0.02528837 -12.56506       0
##
##  Correlation:
##         (Intr) age
## age     -0.349
## age_sp6  0.325 -0.910
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.48944292 -0.39144633 -0.01614359  0.55262959  2.93829176
##
## Residual standard error: 0.5492805
## Degrees of freedom: 300 total; 297 residual
```

```
summary(mod.gls.uns.het)$tTable
```

```
##                  Value  Std.Error   t-value       p-value
## (Intercept)  5.0997111 0.09917412  51.42179 6.554920e-150
## age          0.4608473 0.03011970  15.30053  2.270749e-39
## age_sp6     -0.3177498 0.02528837 -12.56506  2.505260e-29
```

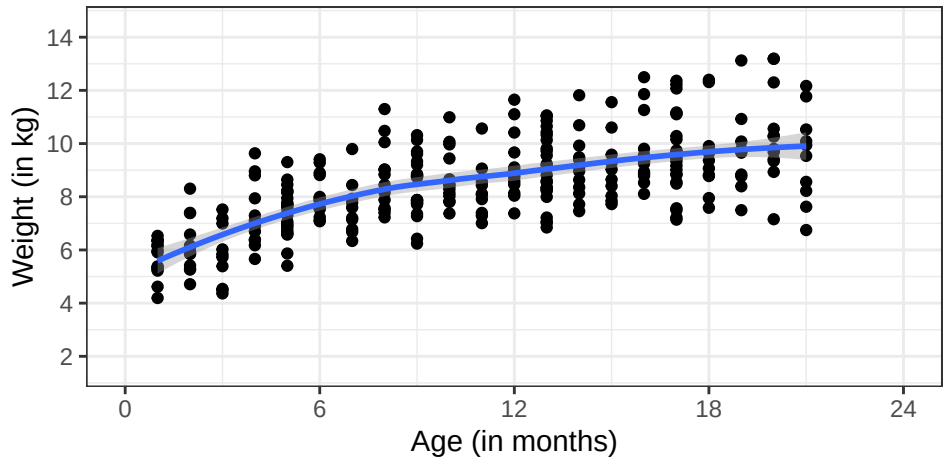
3. Exercise

Fit appropriate models to analyze the relationship between age and weight in dataset `nepal2`.

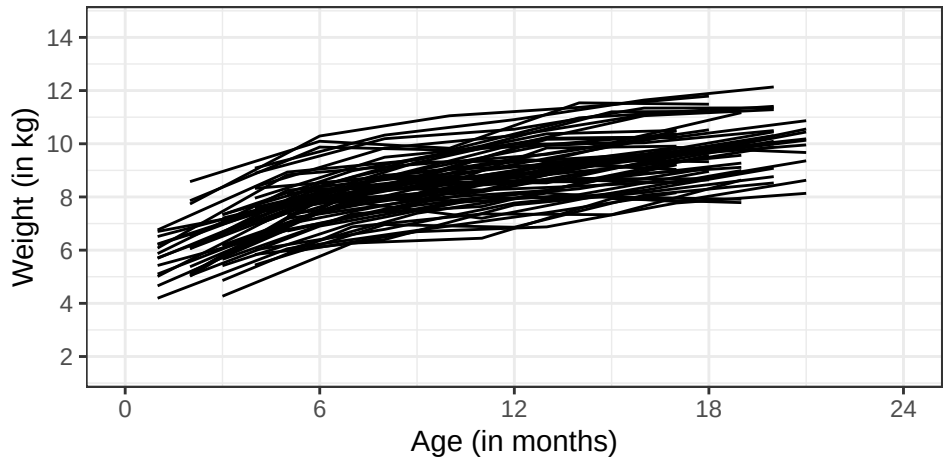
Exploratory analyses are given below. What's the main difference between the two datasets?

```
ggplot(data = nepal1, aes(x = age, y = wt)) +
  geom_point() + theme_bw() +
  geom_smooth() +
  labs(y="Weight (in kg)",x="Age (in months)") +
  scale_y_continuous(breaks=seq(2,14,2),limits=c(1.5,14.5)) +
  scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data = nepal2, aes(x = age, y = wt, group = factor(id))) +
  geom_line() + theme_bw() +
  labs(y="Weight (in kg)",x="Age (in months)") +
  scale_y_continuous(breaks=seq(2,14,2),limits=c(1.5,14.5)) +
  scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```



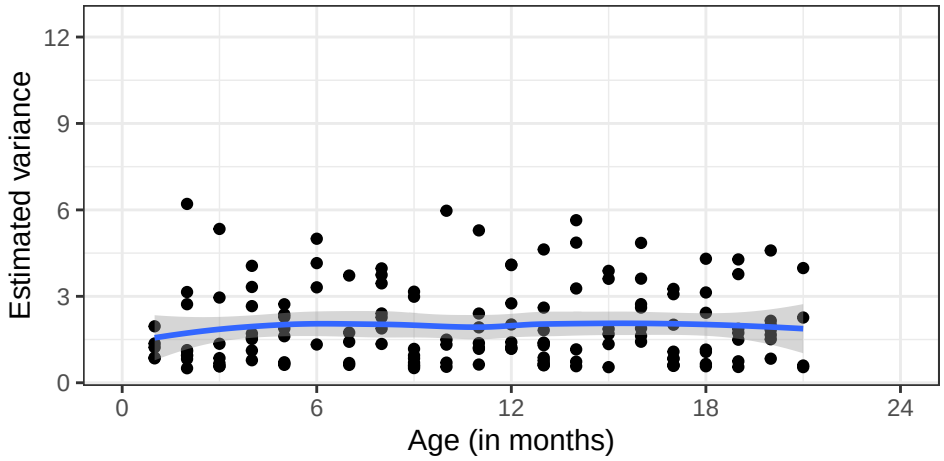

```
nepal2$residuals = residuals(lm(wt~age+age_sp6,data=nepal2))
nepal2_wide = nepal2 %>% select(id, fuvisit, residuals) %>% spread(fuvisit,residuals)
#pairs(nepal2_wide[,-1])
cor(nepal2_wide[,-1])
```

```
##           0         1         2         3         4
## 0 1.0000000 0.9020224 0.8690523 0.9210077 0.9273866
## 1 0.9020224 1.0000000 0.8928467 0.9026381 0.9190152
## 2 0.8690523 0.8928467 1.0000000 0.8884843 0.9119631
## 3 0.9210077 0.9026381 0.8884843 1.0000000 0.9356709
## 4 0.9273866 0.9190152 0.9119631 0.9356709 1.0000000
```

```
ggplot(nepal2,aes(x=age,y=residuals^2)) +
    geom_point() + theme_bw() +
    geom_smooth() +
    labs(y="Estimated variance",x="Age (in months)") +
    scale_y_continuous(breaks=seq(0,12,3),limits=c(0.5,12.5)) +
```

```
    scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 154 rows containing non-finite values (stat_smooth).
## Warning: Removed 154 rows containing missing values (geom_point).
```



IV. Generalized Estimating Equations Approach

Weighted least squares / generalized least squares is a special case of a general method called Generalized Estimating Equations (GEE).

In the case of $Y_i \sim MVN(X_i\beta, V_i)$, the WLS/GEE method finds the values of β that equates the score equations (i.e. estimating equations) to 0. In the case of independent Y_i, for $i = 1, ..., m$, the $\hat{\beta}_{wls}$ solves:

$$S(\beta, \theta) = \sum_{i=1}^{m} \frac{\partial X_i\beta}{\partial \beta} V_i(\theta)^{-1}(Y_i - X_i\beta) = 0$$

The estimation procedure is iterative, same as WLS.

A. Why do we care about GEE?

1. We can use the *R* implementation of GEE to obtain robust variance estimates.

- NOTE: the *gls* function in R does not have this capability. There is a "clubSandwich" package that computes robust variance estimates for model fits from *gls*, but I can't get this package to produce estimates consistently (i.e. I get error messages often and haven't been able to figure out how to solve the errors)
- NOTE: one additional comment on the implementation of GEE in *R*. The GEE packages in *R* assume $V_i(\theta) = 1/\sigma^2 R_i(\theta)$, where R_i is the correlation matrix and $Var(Y_{ij}) = \sigma^2$, i.e. constant variance is assumed. Therefore, you should ignore the "naive s.e." (model based) standard error estimates that are produced by this function; since the constant variance assumption may be violated.

2. We will revisit GEE in the 4th term since this method is general and can fit longitudinal models to outcomes that are continuous/normal (i.e. WLS) or binary or count!

B. Example: NEPAL1

Go back to the *nepal*1 dataset and look at the results of implementing WLS with and without robust variance estimates given several working models for Σ.

```
## OLS for reference
mod.ols = summary(lm(wt~age+age_sp6,data=nepal1))$coeff
mod.ols.out = paste0(round(mod.ols[,1],3)," (",round(mod.ols[,2],3),")")

## Weighted least squares:  constant variance, exchangeable correlation
mod.gls.exch.fit = gls(wt ~ age + age_sp6,
      data = nepal1,correlation=corCompSymm(form=~1|id))
mod.gls.exch = summary(mod.gls.exch.fit)$tTable
mod.gls.exch.out = paste0(round(mod.gls.exch[,1],3)," (",round(mod.gls.exch[,2],3),")")
## Weighted least squares:  non-constant variance, exchangeable correlation
mod.gls.exch.het.fit = gls(wt ~ age + age_sp6,
      data = nepal1,correlation=corCompSymm(form=~1|id),
      weights=varFunc(~age))
mod.gls.exch.het = summary(mod.gls.exch.het.fit)$tTable
mod.gls.exch.het.out = paste0(round(mod.gls.exch.het[,1],3)," (",round(mod.gls.exch.het[,2],3),")")
## GEE:  constant variance, exchangeable correlation, robust variance estimate
mod.gee.exch = summary(gee(wt ~ age + age_sp6, id = id,
    data = nepal1,family = gaussian,
    corstr = "exchangeable"))$coefficients
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

## running glm to get initial regression estimate

## (Intercept)         age     age_sp6
##   5.0739520   0.4857227  -0.3438533
```

```
mod.gee.exch.out = paste0(round(mod.gee.exch[,1],3)," (",round(mod.gee.exch[,4],3),")")
## GEE:  constant variance, independence model, robust variance estimate
mod.gee.ind = summary(gee(wt ~ age + age_sp6, id = id,
    data = nepal1,family = gaussian,
    corstr = "independence"))$coefficients
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate

## (Intercept)         age     age_sp6
##   5.0739520   0.4857227  -0.3438533
```

```
mod.gee.ind.out = paste0(round(mod.gee.ind[,1],3)," (",round(mod.gee.ind[,4],3),")")

## Weighted least squares:  constant variance, AR1
mod.gls.ar1.fit = gls(wt ~ age + age_sp6,
    data = nepal1,correlation=corAR1(form=~num|id))
mod.gls.ar1 = summary(mod.gls.ar1.fit)$tTable
mod.gls.ar1.out = paste0(round(mod.gls.ar1[,1],3)," (",round(mod.gls.ar1[,2],3),")")
## Weighted least squares:  non-constant variance, AR1
mod.gls.ar1.het.fit = gls(wt ~ age + age_sp6,
    data = nepal1,correlation=corAR1(form=~num|id),
    weights=varFunc(~age))
mod.gls.ar1.het = summary(mod.gls.ar1.het.fit)$tTable
mod.gls.ar1.het.out = paste0(round(mod.gls.ar1.het[,1],3)," (",round(mod.gls.ar1.het[,2],3),")")
```

```
## GEE:  constant variance, AR1, robust variance estimate
mod.gee.ar1 = summary(gee(wt ~ age + age_sp6, id = id,
    data = nepal1,family = gaussian,
    corstr = "AR-M",Mv=1))$coefficients
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate
```

```
## (Intercept)         age     age_sp6
##   5.0739520   0.4857227  -0.3438533
```

```
mod.gee.ar1.out = paste0(round(mod.gee.ar1[,1],3)," (",round(mod.gee.ar1[,4],3),")")


## Create an output table
out1 = as.data.frame(cbind(mod.ols.out,mod.gee.ind.out,
    mod.gls.exch.out,mod.gls.exch.het.out,
    mod.gee.exch.out))
out2 = as.data.frame(cbind(mod.ols.out,mod.gee.ind.out,
    mod.gls.ar1.out,mod.gls.ar1.het.out,
    mod.gee.ar1.out))
names(out1) = c("OLS","OLS-RV","Exch","Exch-Het","Exch-RV")
names(out2) = c("OLS","OLS-RV","AR1","AR1-Het","AR1-RV")
row.names(out1) = row.names(out2) = c("Intercept","Age","Age_SP1")
```

```
print(xtable(out1,align="|c|cc|ccc|"))
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Sun Feb 28 21:54:39 2021

	OLS	OLS-RV	Exch	Exch-Het	Exch-RV
Intercept	5.074 (0.289)	5.074 (0.157)	4.915 (0.199)	5.116 (0.082)	4.916 (0.192)
Age	0.486 (0.06)	0.486 (0.026)	0.512 (0.029)	0.468 (0.025)	0.511 (0.032)
Age_SP1	-0.344 (0.071)	-0.344 (0.028)	-0.366 (0.034)	-0.314 (0.03)	-0.366 (0.034)

```
print(xtable(out2,align="|c|cc|ccc|"))
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Sun Feb 28 21:54:39 2021

	OLS	OLS-RV	AR1	AR1-Het	AR1-RV
Intercept	5.074 (0.289)	5.074 (0.157)	4.98 (0.19)	5.106 (0.073)	4.99 (0.165)
Age	0.486 (0.06)	0.486 (0.026)	0.495 (0.022)	0.467 (0.024)	0.494 (0.023)
Age_SP1	-0.344 (0.071)	-0.344 (0.028)	-0.348 (0.025)	-0.313 (0.028)	-0.347 (0.023)

How would we decide which model to pick?

We could evaluate an information criteria statistic like the Akaike information criteria (AIC): $-2 \times ll + 2 \times p$, where ll is the log-likeihood and p is the number of parameters estimated in the model. Smaller values of -2 x ll indicate smaller residuals (better fit) and the AIC adds a penalty for how complicated the model is.

```
AIC(mod.gls.exch.fit,mod.gls.exch.het.fit,mod.gls.ar1.fit,mod.gls.ar1.het.fit)
```

```
##                       df      AIC
## mod.gls.exch.fit       5 729.3473
## mod.gls.exch.het.fit   5 827.3266
## mod.gls.ar1.fit        5 589.6508
## mod.gls.ar1.het.fit    5 731.3010
```

B. Example: NEPAL2

Using the NEPAL2 dataset, run the same code as above and notice that the model with the exchangeable correlation structure with constant variance assumption is the winner according to the AIC.Does this jive with your exploratory analysis that you conducted?

V. Appendix: Code for simulating the datasets

Here I have provided the code I used to generate the two example datasets for today's lecture. We will discuss the models used to generate the data in Lecture 12.

```
library(MASS)
library(dplyr)
library(ggplot2)
load("nepal.anthro.rdata")
nepal = nepal.anthro[nepal.anthro$id %in% unique(nepal.anthro$id)[1:60],]
nepal = nepal[order(nepal$id,nepal$fuvisit),]

nepal = nepal %>% select(id, age, num, fuvisit,agemin)
# Rescale age so that all children are
# recruited between ages 1 to 6 months
nepal$agemin = (nepal$agemin %% 5)+1
nepal$age = nepal$agemin+nepal$fuvisit*4
## Create the spline term
nepal$agec = nepal$age - 6
nepal$age_sp6 = ifelse(nepal$age > 6, nepal$age - 6, 0)

# Model parameters
b0 = 5
b1 = 0.5
b2 = -0.35
c01 = -0.15*sqrt(0.85)*sqrt(0.005)
V = matrix(c(0.85,c01,c01,0.005), nrow=2)
sigmae = 0.3

set.seed(435534)
u = mvrnorm(length(unique(nepal$id)), mu=c(0,0), Sigma=V)

u0 = rep(u[,1], each=5)
u1 = rep(u[,2], each=5)

Y = b0 + u0 + (b1 + u1)*nepal$age + b2*nepal$age_sp6 + rnorm(nrow(nepal), mean = 0, sd = sigmae)
nepal1 = nepal
nepal1$wt = Y
summary(Y)
ggplot(data = nepal1, aes(x=age, y=wt)) +
    geom_point()

Y = b0 + u0 + b1*nepal$age + b2*nepal$age_sp6 +
    rnorm(nrow(nepal), mean = 0, sd = sigmae)
nepal2 = nepal
nepal2$wt = Y
ggplot(data = nepal2, aes(x=age, y=wt)) +
```

```
    geom_point()

save(nepal1, nepal2, file = "nepal_simulated.rda")
```