hw4

Will Tirone

Q1)

model 1

We just want τ here, so we have the following. Note that I've abbreviated the expectation and didn't write out every detail, but many of the terms cancel and $E(\epsilon_{ij}) = 0$

$$\begin{aligned}
\tau &= E(Y_{ij}|Z_i = 1) - E(Y_{ij}|Z_i = 0) \\
&= E[\beta_0 + \beta_z Z_i + \Sigma_t \beta_t 1(j = t) + \Sigma_t \beta_{z,t} Z_i 1(j = t) + \epsilon_{ij}] - E[\beta_0 + \Sigma_t \beta_t 1(j = t) + \epsilon_{ij}] \\
&= \beta_z Z_i + \Sigma_t \beta_{z,t} Z_i 1(j = t)
\end{aligned}$$

model 2

$$Y_{ij} = \beta_0 + \beta_z Z_i + \Sigma_t \beta_t 1(j = t) + \Sigma_t \beta_{z,t} Z_i 1(j = t) + \epsilon^*;$$
$$\epsilon* = (\epsilon^*_{i1}, \dots, \epsilon^*_{i4}) \sim N(0, V_i), \quad \text{where V is a J by J PSD matrix}$$

However, nothing changes about τ since the mean structure is still the same:

$$\begin{aligned}
\tau &= E(Y_{ij}|Z_i = 1) - E(Y_{ij}|Z_i = 0) \\
&= E[\beta_0 + \beta_z Z_i + \Sigma_t \beta_t 1(j = t) + \Sigma_t \beta_{z,t} Z_i 1(j = t) + \epsilon_{ij}] - E[\beta_0 + \Sigma_t \beta_t 1(j = t) + \epsilon_{ij}] \\
&= \beta_z Z_i + \Sigma_t \beta_{z,t} Z_i 1(j = t)
\end{aligned}$$

model 3

Here we just add a random intercept for the individual-specific R.E.

$$\begin{aligned}
Y_{ij} &= \beta_0 + b_{0i} + \beta_z Z_i + \Sigma_t \beta_t 1(j=t) + \Sigma_t \beta_{z,t} Z_i 1(j=t) + \epsilon_{ij} \\
\epsilon &\sim N(0, \sigma^2) \\
b_{0i} &\sim N(0, \sigma_b^2)
\end{aligned}$$

We could either apply iterated expectation to remove the conditioning on b_{0i}, but we can also notice that they'll just cancel out as well.

$$\begin{aligned}
\tau &= E(Y_{ij}|Z_i = 1, b_{0i}) - E(Y_{ij}|Z_i = 0, b_{0i}) \\
&= E[\beta_0 + b_{0i} + \beta_z Z_i + \Sigma_t \beta_t 1(j=t) + \Sigma_t \beta_{z,t} Z_i 1(j=t) + \epsilon_{ij}] - E[\beta_0 + b_{0i} + \Sigma_t \beta_t 1(j=t) + \epsilon_{ij}] \\
&= \beta_z Z_i + \Sigma_t \beta_{z,t} Z_i 1(j=t)
\end{aligned}$$

Q2)

```
lead = read.csv('lead.csv') |>
  pivot_longer(
    -c(ID, treat_group),
    values_to = "lead_level",
    names_to = "week"
  ) |>
  mutate(y_star = if_else(lead_level > 30, 1, 0))
```

model 1)

```
m2_1 = lm(lead_level ~ treat_group*week, data = lead)
```

model 2)

Independent variance:

```
m2_2_ind = geese(lead_level ~ treat_group*week,
                 id = ID,
                 data = lead,
                 scale.fix = TRUE,
                 corstr = 'independence')
```

Exchangeable Variance

```
m2_2_exch = geese(lead_level ~ treat_group*week,
                 id = ID,
                 data = lead,
                 scale.fix = TRUE,
                 corstr = 'exchangeable')
```

AR Covariance

```
m2_2_AR = geese(lead_level ~ treat_group*week,
                 id = ID,
                 data = lead,
                 scale.fix = TRUE,
                 corstr = 'ar1')
```

model 3)

```
m_2_3 = lmer(lead_level ~  (1 | ID) + treat_group*week,
             data = lead)
```

Comparison and Conclusion:

The fitted models are below. The linear model is wrong because there's clustering that might be important for an individual. Of course, repeated measurements on an individual will not be independent, so that choice is wrong for the GEE model. Exchangeable variance, though, where "all measurements on the same unit are equally correlated" is definitely plausible. We don't have any prior knowledge to reject that, so I'll say that model is appropriate. Last, auto-regressive variance is plausible as well since we could expect lead measurement levels to decline over time based on treatment (or not). I think this is probably a scientific distinction based on the specific scenario, so either exchangeability or auto-regressive variances are appropriate here. As we've discussed in class, the clustering is more of a nuisance here since we want to generalize to a whole population, so (as long as the scientific guidance agreed) we would probably not use the mixed effects model.

It looks all the point estimates are the same, though the linear model and mixed effects models have the same S.E., while all of the GEE models have the same S.E.

```
data.frame(
  model = c("lm", "independent", 'exchangeable', 'AR', "mixed effects"),
  treatment_coeff = c(coef(summary(m2_1))[2], m2_2_ind$beta[2],
                    m2_2_exch$beta[2], m2_2_AR$beta[2],
                    coef(summary(m_2_3))[2]),
  SE = c(coef(summary(m2_1))[10], summary(m2_2_ind)$mean$san.se[2],
         summary(m2_2_exch)$mean$san.se[2],
         summary(m2_2_AR)$mean$san.se[2],
         coef(summary(m_2_3))[10])) |>
  kable()
```

model	treatment_coeff	SE
lm	-0.268	1.3251428
independent	-0.268	0.9944085
exchangeable	-0.268	0.9944085
AR	-0.268	0.9944085
mixed effects	-0.268	1.3251428

Q3)

model 2)

Since we don't have an explicit error term, we can't include how to specify the working covariance matrix. As Dr. Li noted in class, this would be specified at some step in the data augmentation process. So here the model looks the same as in Model 1.

$$\text{logit}\{\Pr(Y_{ij}^* = 1)\} = \beta_0 + \beta_z Z_i$$

Now finding τ:

$$\begin{aligned}\tau &= E(Y_{ij}|Z_i = 1) - E(Y_{ij}|Z_i = 0)\\ &= P(Y_{ij} = 1|Z_i = 1) - P(Y_{ij} = 1|Z_i = 0)\\ &= \text{expit}(\beta_0 + \beta_z) - \text{expit}(\beta_0)\end{aligned}$$

And to solve this, in a hypothetical scenario, we would use GEE to find the estimated coefficients. Then, we would plug those in and have τ:

$$\hat{\tau} = \text{expit}(\hat{\beta}_0 + \hat{\beta}_z) - \text{expit}(\hat{\beta}_0)$$

model3)

$$\text{logit}\{\Pr(Y^*_{ij} = 1)\} = \beta_0 + b_{0i} + \beta_z Z_i b_{0i} \sim N(0, \sigma_b^2)$$

Now finding τ. Importantly, GLMM's are conditional models, so we condition on b_{0i} to maintain cluster-specific interpretation.

$$\begin{aligned}\tau &= E(Y_{ij}|Z_i = 1, b_{0i}) - E(Y_{ij}|Z_i = 0, b_{0i})\\ &= P(Y_{ij} = 1|Z_i = 1, b_{0i}) - P(Y_{ij} = 1|Z_i = 0, b_{0i})\\ &= \text{expit}(\beta_0 + b_{0i} + \beta_z) - \text{expit}(\beta_0 + b_{0i})\end{aligned}$$

Again the problem is computing this quantity. In practice, we would fit a GLMM model and just plug in the estimates into the above:

$$\hat{\tau} = \text{expit}(\hat{\beta}_0 + \widehat{b_{0i}} + \hat{\beta}_z) - \text{expit}(\hat{\beta}_0 + \widehat{b_{0i}})$$

Q4)

Model Fitting

```
m4_ind = geeglm(y_star ~ treat_group,
                data =lead,
                id = ID,
                family = binomial,
                scale.fix = TRUE,
                corstr='independence')

m4_exch = geeglm(y_star ~ treat_group,
                data =lead,
                id = ID,
                family = binomial,
                scale.fix = TRUE,
                corstr='exchangeable')

m4_ar = geeglm(y_star ~ treat_group,
                data =lead,
                id = ID,
                family = binomial,
                scale.fix = TRUE,
                corstr='ar1')

mixed_glm = glmer(y_star ~ (1 | ID) +  treat_group,
                  family = binomial(link = "logit"),
                  data = lead)
```

GLM Coefficient Output

Again our results are somewhat similar with both exchangeable and independent having identical point estimates and standard errors. However, the mixed effect model has the highest point estimate along with S.E., while the AR covariance model has the lowest estimates of all 4.

I would trust the AR model since it seems to make intuitive sense that lead levels in the blood could decline over time based on a treatment or placebo, but we would have to back that up with scientific evidence. Additionally, I think we don't want the mixed effects model because we don't necessarily care about an individuals measurements, we want to generalize to an entire population of people. The clustering is a nuisance here.

```
data.frame(
  model = c("independent",
            'exchangeable',
            'AR',
            "mixed effects"),
  treatment_coeff = c(
    summary(m4_ind)$coeff[2, 'Estimate'],
    summary(m4_exch)$coeff[2, 'Estimate'],
    summary(m4_ar)$coeff[2, 'Estimate'],
    summary(mixed_glm)$coeff[2]),
  SE = c(
    summary(m4_ind)$coeff[2, "Std.err"],
    summary(m4_exch)$coeff[2, "Std.err"],
    summary(m4_ar)$coeff[2, "Std.err"],
    summary(mixed_glm)$coeff[4])) |>
  kable()
```

model	treatment_coeff	SE
independent	0.7034606	0.4253371
exchangeable	0.7034606	0.4253371
AR	0.5035720	0.4221801
mixed effects	0.7489982	0.9252927