

# Code Appendix

Zirui Zhang

```
data0 = data.frame(read_excel("./MRI.xlsx", sheet = "Sheet2")) |>
  janitor::clean_names() |>
  rename("id" = "subject_id",
         "sex" = "m_f") |>
  select(-mri_id, -group, -mr_delay, -hand, -cdr, -asf, -e_tiv, -age) |>
  filter(visit != 4 & visit != 5) |>
  group_by(id) |>
  mutate(base_mmse = ifelse(visit == 1, mmse, 0),
         sex = ifelse(sex == "M", 0, 1)) |>
  ungroup()
data0$base_mmse = ifelse(data0$base_mmse == 0, NA, data0$base_mmse)
data0 = data0 |> fill(base_mmse)

# filter out patients with incoherent data
is_coherent = function(visit) {
  all(diff(visit) == 1)
}
data = data0 |> group_by(id) |> filter(is_coherent(visit)) |> ungroup() |> as.data.frame()
# define complete subjects
ids = length(unique(data$id))
length(data$visit[data$visit == 1]) # 144

## [1] 144

length(data$visit[data$visit == 2]) # 144

## [1] 144

length(data$visit[data$visit == 3]) # 52

## [1] 52

idsC = unique(data$id[data$visit == 3])
data$completer = as.numeric(is.element(data$id, idsC))
data = data |>
  select(id, visit, completer, sex, educ, ses, base_mmse, n_wbv) |>
  mutate(id = as.factor(id),
         sex = as.factor(sex))
# overall drop-out rate
num_drop = length(unique(data[data$completer == 0, ]$id))
p_drop = num_drop/ids # 63.89%
```

```

# missing value for predictors
null_counts = colSums(is.na(data)) # missing mmse is not the baseline mmese, never mind
median_ses = median(data$ses, na.rm = TRUE) # impute missing as median in ses
data$ses = ifelse(is.na(data$ses), median_ses, data$ses)

```

```

only2 = data |> group_by(id) |>
  filter(n() == 2)
visit3 = only2 |>
  mutate(visit = 3, n_wbv = NA) |>
  distinct()
data_miss = bind_rows(data, visit3) |> arrange(id, visit)
# missing data imputation for the outcome variable
## multiple imputation
imp = mice(data_miss, m = 20, seed=200427)

```

```

##
## iter imp variable
## 1 1 n_wbv*
## 1 2 n_wbv*
## 1 3 n_wbv*
## 1 4 n_wbv*
## 1 5 n_wbv*
## 1 6 n_wbv*
## 1 7 n_wbv*
## 1 8 n_wbv*
## 1 9 n_wbv*
## 1 10 n_wbv*
## 1 11 n_wbv*
## 1 12 n_wbv*
## 1 13 n_wbv*
## 1 14 n_wbv*
## 1 15 n_wbv*
## 1 16 n_wbv*
## 1 17 n_wbv*
## 1 18 n_wbv*
## 1 19 n_wbv*
## 1 20 n_wbv*
## 2 1 n_wbv*
## 2 2 n_wbv*
## 2 3 n_wbv*
## 2 4 n_wbv*
## 2 5 n_wbv*
## 2 6 n_wbv*
## 2 7 n_wbv*
## 2 8 n_wbv*
## 2 9 n_wbv*
## 2 10 n_wbv*
## 2 11 n_wbv*
## 2 12 n_wbv*
## 2 13 n_wbv*
## 2 14 n_wbv*
## 2 15 n_wbv*
## 2 16 n_wbv*

```

```

## 2 17 n_wbv*
## 2 18 n_wbv*
## 2 19 n_wbv*
## 2 20 n_wbv*
## 3 1 n_wbv*
## 3 2 n_wbv*
## 3 3 n_wbv*
## 3 4 n_wbv*
## 3 5 n_wbv*
## 3 6 n_wbv*
## 3 7 n_wbv*
## 3 8 n_wbv*
## 3 9 n_wbv*
## 3 10 n_wbv*
## 3 11 n_wbv*
## 3 12 n_wbv*
## 3 13 n_wbv*
## 3 14 n_wbv*
## 3 15 n_wbv*
## 3 16 n_wbv*
## 3 17 n_wbv*
## 3 18 n_wbv*
## 3 19 n_wbv*
## 3 20 n_wbv*
## 4 1 n_wbv*
## 4 2 n_wbv*
## 4 3 n_wbv*
## 4 4 n_wbv*
## 4 5 n_wbv*
## 4 6 n_wbv*
## 4 7 n_wbv*
## 4 8 n_wbv*
## 4 9 n_wbv*
## 4 10 n_wbv*
## 4 11 n_wbv*
## 4 12 n_wbv*
## 4 13 n_wbv*
## 4 14 n_wbv*
## 4 15 n_wbv*
## 4 16 n_wbv*
## 4 17 n_wbv*
## 4 18 n_wbv*
## 4 19 n_wbv*
## 4 20 n_wbv*
## 5 1 n_wbv*
## 5 2 n_wbv*
## 5 3 n_wbv*
## 5 4 n_wbv*
## 5 5 n_wbv*
## 5 6 n_wbv*
## 5 7 n_wbv*
## 5 8 n_wbv*
## 5 9 n_wbv*
## 5 10 n_wbv*

```

```
## 5 11 n_wbv*
## 5 12 n_wbv*
## 5 13 n_wbv*
## 5 14 n_wbv*
## 5 15 n_wbv*
## 5 16 n_wbv*
## 5 17 n_wbv*
## 5 18 n_wbv*
## 5 19 n_wbv*
## 5 20 n_wbv*
```

```
completed_datasets = list()
# Loop over each imputation
for (i in 1:20) {
  # completed dataset for imputation i
  di = complete(imp, i)
  # add weights based on ipw
  di$nonmissY = !is.na(data_miss$n_wbv)
  ps_fit = glm(completer ~ visit + sex + educ + ses + base_mmse, data = di, family = binomial)
  di$weights = ifelse(di$nonmissY, 1 / fitted(ps_fit)[di$nonmissY], 1)
  completed_datasets[[i]] = di
}
imputed_dataset = do.call(rbind, completed_datasets)
data_imp = imputed_dataset |>
  group_by(id, visit, completer, sex, educ, ses, base_mmse) |>
  summarize(n_wbv = round(min(n_wbv, na.rm = TRUE), 3))
```

## 'summarise()' has grouped output by 'id', 'visit', 'completer', 'sex', 'educ',  
## 'ses'. You can override using the '.groups' argument.

```
az0 = data.frame(data_imp)
# baseline character
length(unique(az0$id)) #144
```

```
## [1] 144
```

```
summary(az0$base_mmse)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 17.00   26.00   29.00   27.49   30.00   30.00
```

```
p.h1 = az0 |> ggplot(aes(x = base_mmse)) + geom_histogram()
p.h2 = az0 |> ggplot(aes(x = n_wbv)) + geom_histogram()
length(unique((az0 |> filter(sex == 0))$id)) # 59 male; 85 female
```

```
## [1] 59
```

```
az = az0|>
  select(-completer) |>
  mutate(id = as.character(id),
```

```

educ = scale(educ),
ses = scale(ses),
base_mmse = scale(base_mmse))
# spaghetti plot
p.s = ggplot(az, aes(x = visit, y = n_wbv, group = id)) + geom_line() # random intercept

```

```

# 2-stage analysis
K = 144
az.2 = az |>
  group_by(id) |>
  mutate(idd = group_indices()) |>
  ungroup() |>
  as.data.frame()
mmse = az.2$base_mmse
# Stage 1
betaMat = data.frame(beta0=rep(NA, K), beta.time=rep(NA, K))
for(k in 1:K) {
  temp.k = az.2[az.2$idd == k,]
  fit.k = lm(n_wbv ~ visit, data = temp.k)
  betaMat[k, 1:2] = c(fit.k$coef)
}

# Stage 2
data_2 = cbind(mmse, betaMat)
model_visit = lm(beta.time ~ mmse, data = data_2)
sum_2_stage = summary(model_visit)$coefficients

```

```

# covariance structure
fitf = lm(n_wbv ~ visit, data = az)
resMat = matrix(residuals(fitf), ncol=3, byrow=TRUE)
sd = round(sqrt(diag(cov(resMat))), 2)
sd = c(0.04, 0.04, 0.04)
comat = round(cov(resMat), 2)
diag(comat) = sd # ar1 seems most suitable

```

```

fit.I = geeglm(n_wbv ~ visit+base_mmse+visit*base_mmse+sex+educ+ses, id=id, az, family=binomial(link="logit"), scale.fix=TRUE)
fit.E = geeglm(n_wbv ~ visit+base_mmse+sex+educ+ses, id=id, az, family=binomial(link="logit"), scale.fix=TRUE)
fit.AR = geeglm(n_wbv ~ visit+base_mmse+sex+educ+ses, id=id, az, family=binomial(link="logit"), scale.fix=TRUE)
summary(fit.I)

```

```

##
## Call:
## geeglm(formula = n_wbv ~ visit + base_mmse + visit * base_mmse +
## sex + educ + ses, family = binomial(link = "logit"), data = az,
## id = id, corstr = "independence", scale.fix = TRUE)
##
## Coefficients:
##              Estimate      Std. err      Wald Pr(>|W|)
## (Intercept)   1.197e+00  3.889e-16  9.483e+30  <2e-16 ***
## visit        -1.546e-01  1.099e-16  1.976e+30  <2e-16 ***
## base_mmse     7.401e-02  7.518e-17  9.690e+29  <2e-16 ***
## sex1         3.374e-02  3.067e-17  1.211e+30  <2e-16 ***
## educ         4.515e-03  2.307e-17  3.828e+28  <2e-16 ***

```

```
## ses                1.230e-02  4.626e-17  7.075e+28   <2e-16 ***
## visit:base_mmse -8.927e-03  3.169e-17  7.933e+28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Scale is fixed.
##
## Number of clusters: 1 Maximum cluster size: 432
```

```
fit.ri = lme(fixed = n_wbv ~ visit+base_mmse+sex+educ+ses, random=reStruct(~ 1 | id), data=az, method="REML")
summary(fit.ri)
```

```
## Linear mixed-effects model fit by maximum likelihood
## Data: az
## AIC BIC logLik
## -1630 -1597 822.9
##
## Random effects:
## Formula: ~1 | id
## (Intercept) Residual
## StdDev: 0.023 0.03051
##
## Fixed effects: n_wbv ~ visit + base_mmse + sex + educ + ses
## Value Std.Error DF t-value p-value
## (Intercept) 0.7708 0.005305 287 145.30 0.0000
## visit -0.0316 0.001810 287 -17.43 0.0000
## base_mmse 0.0116 0.002583 139 4.49 0.0000
## sex1 0.0069 0.005117 139 1.35 0.1803
## educ 0.0009 0.003477 139 0.26 0.7965
## ses 0.0025 0.003411 139 0.73 0.4640
## Correlation:
## (Intr) visit bs_mms sex1 educ
## visit -0.683
## base_mmse 0.140 0.000
## sex1 -0.569 0.000 -0.246
## educ -0.065 0.000 -0.182 0.114
## ses -0.015 0.000 0.006 0.026 0.687
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -2.32830 -0.62896 0.01546 0.63605 2.50218
##
## Number of Observations: 432
## Number of Groups: 144
```

```
# assume slope for visit depends on the value of baseline mmse
```

```
knitr::kable(sum_2_stage, format = "markdown")
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.0316	0.0011	-27.514	0.0000
mmse	0.0021	0.0011	1.794	0.0735