Assessing sampling methods for generalization from RCTs: Modeling recruitment and participation

Gleb Furman[1] & James E. Pustejovsky[1]

[1] University of Texas at Austin

Assessing sampling methods for generalization from RCTs: Modeling recruitment and participation

```
# Seed for random number generation
# set.seed(42)
```

```
source("ParGenSource.R")
load("Data/base data.rdata")
# Read Data
```

```
to_z <- function(x) (x - mean(x))/sd(x)


dist_plots <- df[, c("DSID",subs_f_vars)] %>%
  mutate(n_log = log(n),
         MEDINC_log = log(MEDINC/1000),
         MEDINC = MEDINC / 1000)


dist_plots %>%
  gather(key = var, value = value, n, n_log, MEDINC, MEDINC_log) %>%
  ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~ var, scales = "free") +
  theme_apa()
```

```
dist_plots %>%
  gather(key = var, value = value, pED:pMin) %>%
  group_by(var) %>%
  # mutate(value = value - mean(value)) %>%
  ggplot(aes(x = value)) +
```

```
  geom_histogram() +
  facet_wrap(~ var, scales = "free") +
  theme_apa()
```

```
load("Paper Data/clusters-full-logs.rdata")
ch_f <- chPlot
clusters_f <- clusters


ch_f$subs <- "F"


# load("Paper Data/clusters-OV-logs.rdata")
# ch_ov <- chPlot
# clusters_ov <- clusters
#
# ch_ov$subs <- "OV"
#
# chPlot <- rbind(ch_f, ch_ov)
```

```
chPlot %>%
  gather(key = method, value = value, ch2) %>%
  ggplot(aes(x = k, y = value)) +
  geom_point() +
  geom_line() +
  theme_apa() +
  scale_x_discrete(limits = c(1:K))
```

```
# cls_ov <- bind_cols(lapply(clusters_ov, function(x) data.frame(x$cluster)))
cls_f <- bind_cols(lapply(clusters, function(x) data.frame(x$cluster)))
```

```r
# ratio_data <- rbind(data.frame(k = 1:K, subs = "F", vrat = unlist(lapply(clusters_f,
#                       data.frame(k = 1:K, subs = "OV", vrat = unlist(lapply(clusters_o

ratio_data <- data.frame(k = 1:K, subs = "F", vrat = unlist(lapply(clusters, function(x

# levels(ratio_data$subs) <- c("Full", "Omitted Variable")

ratio_data %>%
  # group_by(subs) %>%
  mutate(min80 = sum(vrat < .8) + .5) %>%
  # ungroup() %>%
  ggplot(aes(x = k, y = vrat)) +
  geom_point() +
  labs(y = "Between Cluster Variance",
       x = "Number of Strata (k)") +
  geom_line() +
  geom_vline(aes(xintercept = min80), linetype = "dashed") +
  theme_apa(box = F) +
  scale_x_discrete(limits = c(1:K)) +
  scale_y_continuous(breaks = seq(0, 1, .1)) +
  # facet_grid(subs ~ ., , scales = "free") +
  # facet_wrap( ~ subs, , scales = "free", ncol = 1) +
  theme(legend.position = "none",
        panel.spacing = unit(2, "lines"),
        text = element_text(size=20),
        legend.title = element_text(size=15))
```

```r
ggsave("Figs/Elbow.jpg", dpi = 1000, width = 10, height = 8.5)
```

```r
# names(cls_ov) <- names(cls_f) <- 1:K
names(cls_f) <- 1:K
# cls_ov$subs <- "OV"
cls_f$subs <- "F"



# rbind(cls_f, cls_ov) %>%
cls_f %>%
  gather(key = k, value = cluster, -subs) %>%
  filter(k > 1) %>%
  mutate(k = as.numeric(k)) %>%
  group_by(k, cluster, subs) %>%
  summarise(n = n()) %>%
  group_by(k, subs) %>%
  mutate(sample = (n / sum(n)) * 60) %>%
  ggplot(aes(x = k, y = sample)) +
  geom_point() +
  theme_apa() +
  labs(y = "Allocated Sample Size",
       x = "Number of Strata (k)") +
  scale_x_discrete(limits = c(1:K)) +
  scale_y_continuous(breaks = seq(0, 30, 5)) +
  stat_function(fun = function(x) 60/x, geom = "line", linetype = "dashed") +
  # facet_grid(subs ~ .) +
  geom_hline(yintercept = 5, linetype = "dotted")
```

```
# multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
#   library(grid)
#
#   # Make a list from the ... arguments and plotlist
#   plots <- c(list(...), plotlist)
#
#   numPlots = length(plots)
#
#   # If layout is NULL, then use 'cols' to determine layout
#   if (is.null(layout)) {
#     # Make the panel
#     # ncol: Number of columns of plots
#     # nrow: Number of rows needed, calculated from # of cols
#     layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
#                      ncol = cols, nrow = ceiling(numPlots/cols))
#   }
#
#  if (numPlots==1) {
#     print(plots[[1]])
#
#   } else {
#     # Set up the page
#     grid.newpage()
#     pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
#
#     # Make each plot, in the correct location
#     for (i in 1:numPlots) {
```

```
#       # Get the i,j matrix positions of the regions that contain this subplot
#       matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
#
#       print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
#                                       layout.pos.col = matchidx$col))
#     }
#   }
# }
#
# multiplot(ch_full, ratio_full, k_size_full, cols = 1)
```

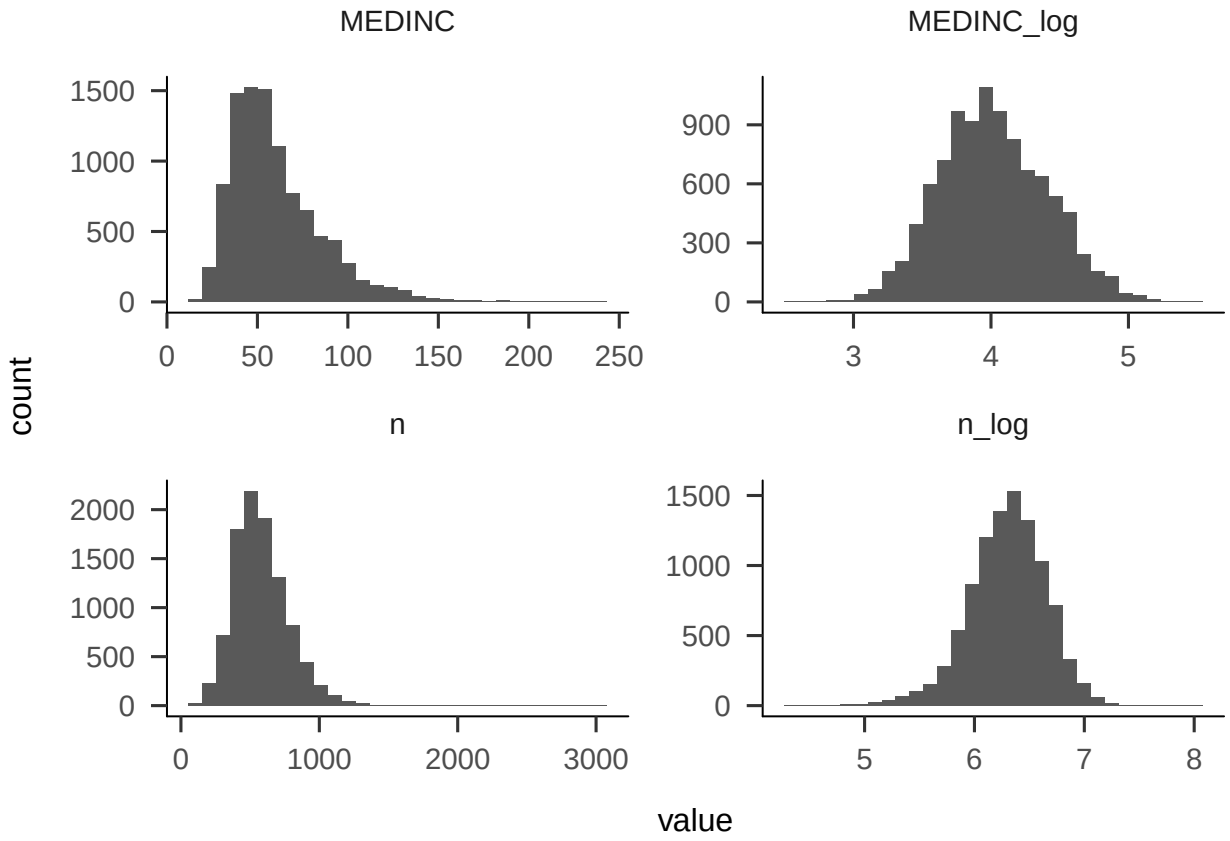

Figure 1. Comparison of covariate distributions and their log transformations.

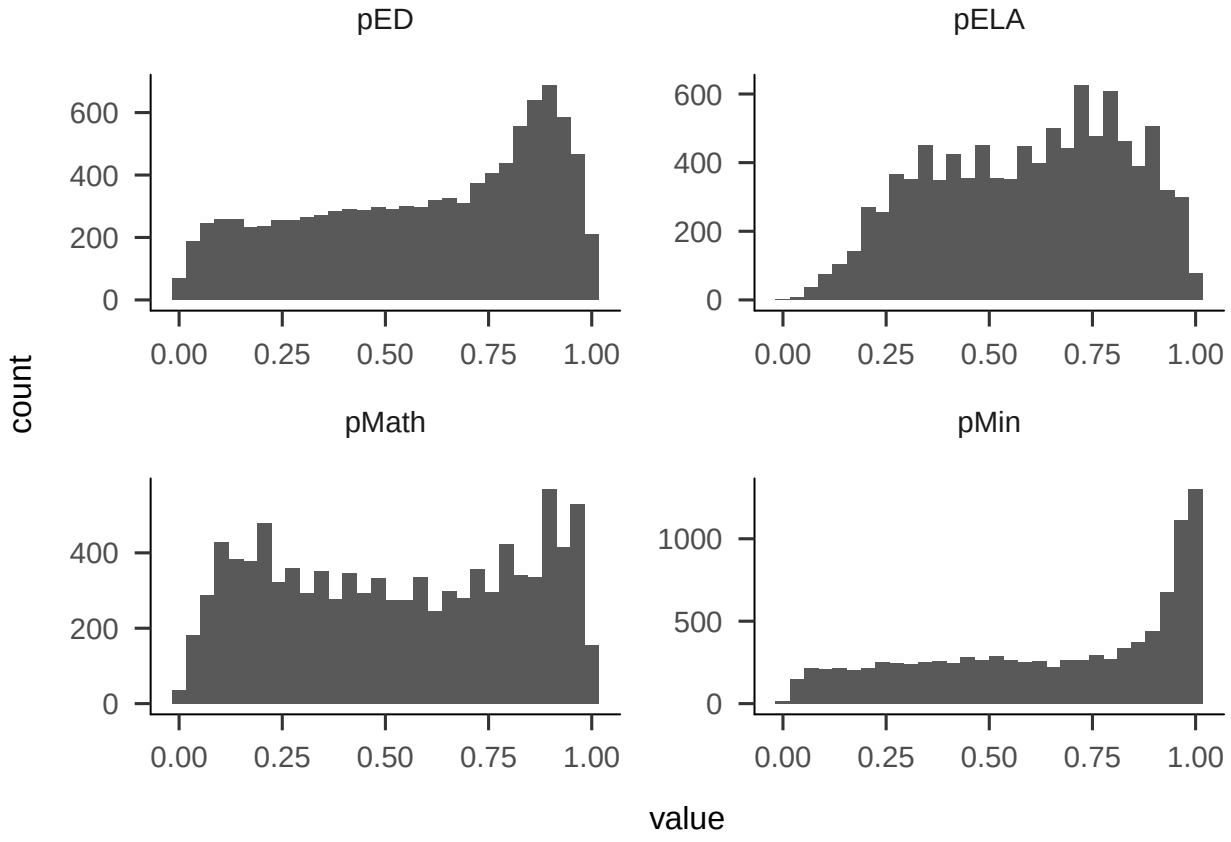

Figure 2. Distributions of the remaining continuous covariates.

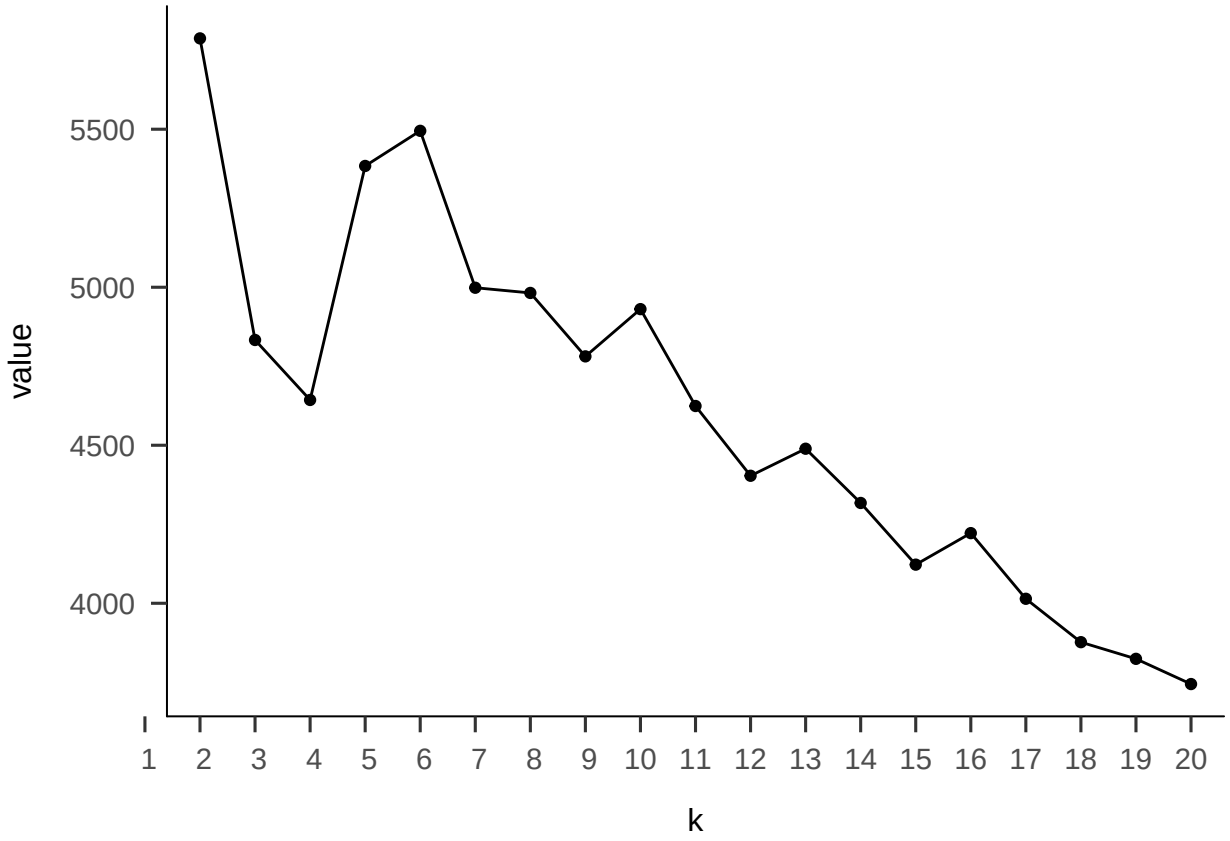

Figure 3. Generalizd Calinski-Harabasz index

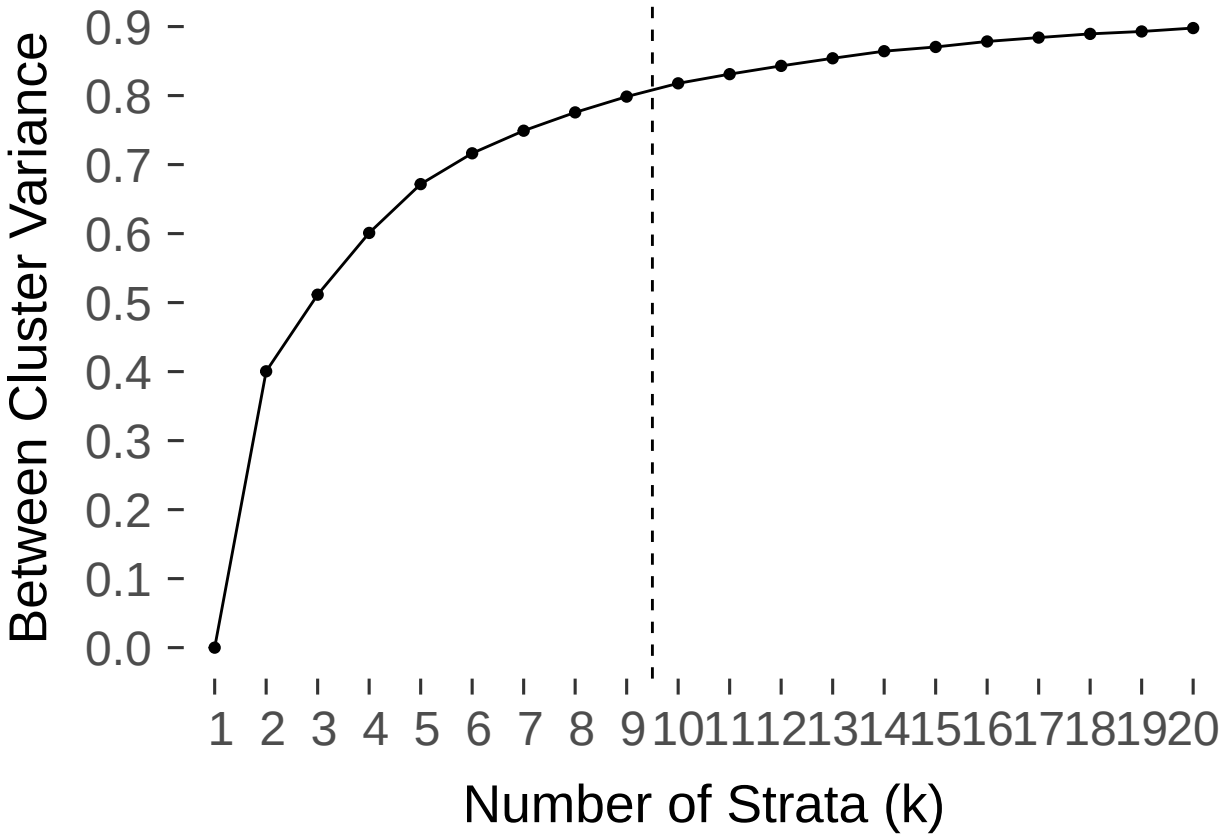

Figure 4. Ratio of between cluster sum of squares to total cluster sum of squares

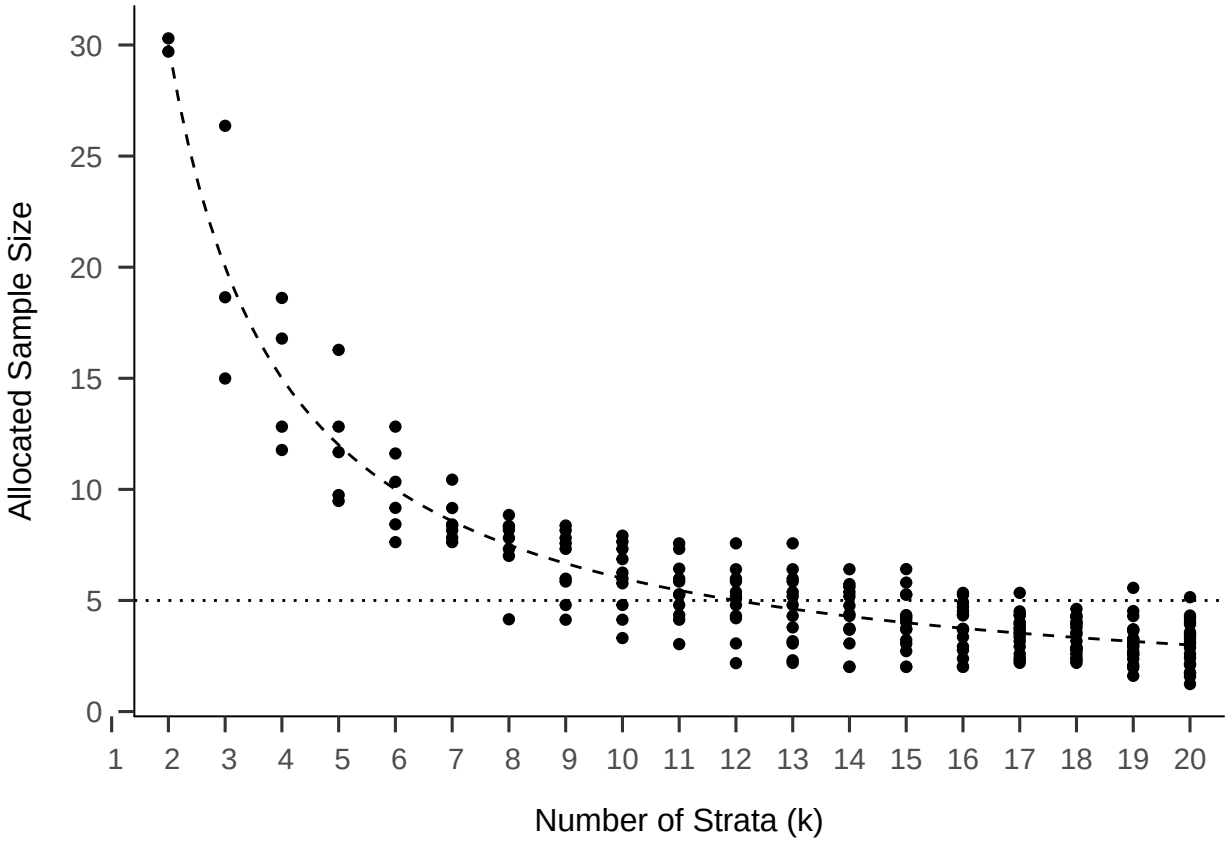

Figure 5. Sampling requirements for each cluster