CSA Scatterplots — Standardized & Reproducible

Shreya Shankar ([sshankar@mitre.org](mailto:sshankar@mitre.org))

2025-08-05

**Overview & Policy Context**

**What this produces:** three standardized scatterplots that relate Medicare Advantage (MA) enrollment (size of the market) and MA penetration (share of eligibles enrolled) at different geographic resolutions. The quadrants are split at sample medians so audiences can quickly see High/Low Enrollment × High/Low Penetration patterns without remembering absolute cutoffs.

**What you’ll see:**

Scatterplot A (CSA+1): Treats each CSA×State as a separate market—useful for surfacing within-CSA, cross-state differences.

Scatterplot B (CSA merged + Non-CSA by state): Collapses split-state CSAs to a single CSA unit and keeps non-CSA rows aggregated by state; color/shape distinguish CSA vs. non-CSA.

Scatterplot C (CSA only): Focuses only on merged CSAs to emphasize market-scale relationships.

*Interpretation note*: The log10 x-axis (enrollment) compresses the right tail so both small and very large markets remain visible.

**Inputs, Normalization & Reproducibility**

The setup block installs/loads packages, reads DATA\_LOCATION.txt, verifies Processed/penrate\_r.xlsx, and sets Results/Maps for exports. The helper *parse\_pen\_rate\_frac()* ensures penetration is a fraction in [0, 1] regardless of whether the sheet stored it as 0.62 or 62%. Rows labeled like “All Other Areas – XX” are flagged as non-CSA; everything else includes a parsed CSA code and state.

*Note on “eligible” back-calculation:* When penetration is non-zero, the script computes a rough eligible = enrolled / penetration for proper aggregation in B/C. This keeps penetration weighted correctly when you roll up rows.

**Quadrants & Labels: How to Read All Three Plots**

Each plot computes its own medians for enrollment and penetration from the data actually plotted on that page. Vertical and horizontal dashed lines mark those medians; colors/legends summarize the n in each quadrant. To keep labels legible, one representative point per quadrant is auto-selected (closest to the quadrant’s mean position) and annotated with *ggrepel*.

**Note**: Because the medians are plot-specific, a CSA can change quadrants between A, B, and C as the comparison set changes (CSA×State vs merged CSAs vs CSAs-only).

required <- c(  
 "tidyverse", "readxl", "ggExtra", "ggrepel", "scales", "stringr", "grid"  
)  
new\_pkgs <- setdiff(required, rownames(installed.packages()))  
if (length(new\_pkgs)) install.packages(new\_pkgs, quiet = TRUE)  
lapply(required, library, character.only = TRUE)

## [[1]]  
## [1] "lubridate" "forcats" "stringr" "dplyr" "purrr" "readr"   
## [7] "tidyr" "tibble" "ggplot2" "tidyverse" "stats" "graphics"   
## [13] "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[2]]  
## [1] "readxl" "lubridate" "forcats" "stringr" "dplyr" "purrr"   
## [7] "readr" "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [13] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[3]]  
## [1] "ggExtra" "readxl" "lubridate" "forcats" "stringr" "dplyr"   
## [7] "purrr" "readr" "tidyr" "tibble" "ggplot2" "tidyverse"  
## [13] "stats" "graphics" "grDevices" "utils" "datasets" "methods"   
## [19] "base"   
##   
## [[4]]  
## [1] "ggrepel" "ggExtra" "readxl" "lubridate" "forcats" "stringr"   
## [7] "dplyr" "purrr" "readr" "tidyr" "tibble" "ggplot2"   
## [13] "tidyverse" "stats" "graphics" "grDevices" "utils" "datasets"   
## [19] "methods" "base"   
##   
## [[5]]  
## [1] "scales" "ggrepel" "ggExtra" "readxl" "lubridate" "forcats"   
## [7] "stringr" "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [13] "ggplot2" "tidyverse" "stats" "graphics" "grDevices" "utils"   
## [19] "datasets" "methods" "base"   
##   
## [[6]]  
## [1] "scales" "ggrepel" "ggExtra" "readxl" "lubridate" "forcats"   
## [7] "stringr" "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [13] "ggplot2" "tidyverse" "stats" "graphics" "grDevices" "utils"   
## [19] "datasets" "methods" "base"   
##   
## [[7]]  
## [1] "grid" "scales" "ggrepel" "ggExtra" "readxl" "lubridate"  
## [7] "forcats" "stringr" "dplyr" "purrr" "readr" "tidyr"   
## [13] "tibble" "ggplot2" "tidyverse" "stats" "graphics" "grDevices"  
## [19] "utils" "datasets" "methods" "base"

#Results path helper  
results\_path <- function(filename) file.path(results\_dir, filename)  
  
  
stopifnot(file.exists("DATA\_LOCATION.txt"))  
data\_location <- readr::read\_file("DATA\_LOCATION.txt") |> stringr::str\_trim()  
  
penrate\_path <- file.path(data\_location, "Processed", "penrate\_r.xlsx")  
if (!file.exists(penrate\_path)) {  
 stop("Expected file not found: ", penrate\_path,  
 "\nEnsure DATA\_LOCATION.txt points to the Teams Data folder and that",  
 "\n'Processed/penrate\_r.xlsx' exists.")  
}  
  
#Save to Results/Maps  
results\_dir <- file.path(data\_location, "Results", "Maps")  
if (!dir.exists(results\_dir)) dir.create(results\_dir, recursive = TRUE)  
  
  
#Shared helpers (kept minimal to avoid changing results)  
  
#Parse Penetration Rate  
parse\_pen\_rate\_frac <- function(x) {  
 pr <- readr::parse\_number(as.character(x))  
 dplyr::if\_else(!is.na(pr) & pr > 1, pr / 100, pr)  
}  
  
#Build a minimal frame of the raw rows with normalized columns  
read\_penrate\_rows <- function(path, sheet = "Sheet1") {  
 readxl::read\_excel(path, sheet = sheet) |>  
 dplyr::select(  
 csa\_area = `CSA Area`,  
 enrolled = `Sum of Enrolled`,  
 pen\_raw = `Penetration Rate`  
 ) |>  
 dplyr::filter(  
 !is.na(csa\_area),  
 !stringr::str\_detect(stringr::str\_to\_lower(csa\_area), "grand total")  
 ) |>  
 dplyr::mutate(  
 enrolled = as.numeric(enrolled),  
 penetration\_rate = parse\_pen\_rate\_frac(pen\_raw),  
 #Flags/parts used by scatterplots B & C  
 is\_non\_csa = stringr::str\_detect(  
 csa\_area,  
 stringr::regex("^\\s\*All\\s+Other\\s+Areas\\s\*-", ignore\_case = TRUE)  
 ),  
 state = stringr::str\_trim(stringr::str\_extract(csa\_area, "[A-Z]{2}\\s\*$")),  
 csa\_code = dplyr::if\_else(  
 is\_non\_csa,  
 NA\_character\_,  
 stringr::str\_match(  
 csa\_area,  
 stringr::regex("CSA\\s\*-\\s\*(\\d{3})\\s\*-", ignore\_case = TRUE)  
 )[, 2]  
 ),  
 eligible = dplyr::if\_else(  
 dplyr::coalesce(penetration\_rate, 0) > 0,  
 enrolled / penetration\_rate,  
 NA\_real\_  
 )  
 )  
}  
  
#Add quadrant labels based on medians  
add\_quadrants <- function(df, x = enrolled, y = penetration\_rate) {  
 x <- rlang::enquo(x); y <- rlang::enquo(y)  
 med\_x <- stats::median(dplyr::pull(df, !!x), na.rm = TRUE)  
 med\_y <- stats::median(dplyr::pull(df, !!y), na.rm = TRUE)  
  
 df |>  
 dplyr::mutate(  
 quadrant = dplyr::case\_when(  
 (!!x) <= med\_x & (!!y) <= med\_y ~ "Low Enroll / Low Penetration",  
 (!!x) > med\_x & (!!y) <= med\_y ~ "High Enroll / Low Penetration",  
 (!!x) <= med\_x & (!!y) > med\_y ~ "Low Enroll / High Penetration",  
 TRUE ~ "High Enroll / High Penetration"  
 ),  
 .after = !!y  
 ) |>  
 #return medians for subtitle  
 dplyr::mutate(`.med\_x` = med\_x, `.med\_y` = med\_y)  
}  
  
#Choose one representative label per quadrant  
pick\_representatives\_by\_quadrant <- function(df, lbl\_col = "name",  
 x = enrolled, y = penetration\_rate) {  
 x <- rlang::enquo(x); y <- rlang::enquo(y)  
 df |>  
 dplyr::group\_by(quadrant) |>  
 dplyr::mutate(  
 mean\_x = mean(!!x, na.rm = TRUE),  
 mean\_y = mean(!!y, na.rm = TRUE),  
 dist\_to\_avg = sqrt(( (!!x) - mean\_x )^2 + ( (!!y) - mean\_y )^2)  
 ) |>  
 dplyr::slice\_min(dist\_to\_avg, n = 1, with\_ties = FALSE) |>  
 dplyr::ungroup() |>  
 dplyr::rename(label = dplyr::all\_of(lbl\_col))  
}  
  
#colors!  
quadrant\_colors <- c(  
 "Low Enroll / Low Penetration" = "#87DEFF",  
 "High Enroll / Low Penetration" = "#005B94",  
 "Low Enroll / High Penetration" = "#063F66",  
 "High Enroll / High Penetration" = "#0B2338"  
)  
type\_colors <- c("CSA" = "#0B2338", "Non-CSA" = "#87DEFF") #for scatterplot B  
  
pen\_rows <- read\_penrate\_rows(penrate\_path, sheet = "Sheet1")

**A) CSA+1 — Enrollment vs. Penetration (State Splits Preserved)**

**Purpose.** Keep CSA×State observations separate to reveal cross-state variation within the same CSA. This is helpful when delivery systems, plan participation, or broker networks differ across state lines.

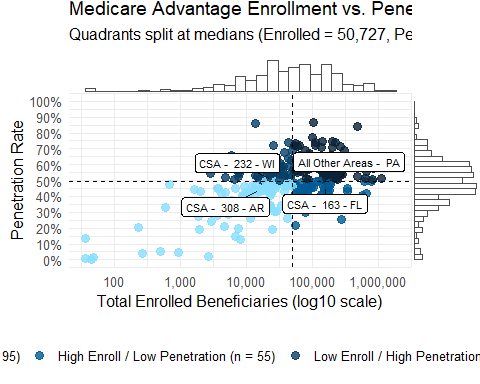
**How to read:**

Rightward means larger enrollment (log10).

Upward means higher penetration (0–100%).

Quadrant counts in the legend tell you where the mass of markets sits.

csa\_df\_A <- pen\_rows |>  
 dplyr::transmute(  
 csa = csa\_area,  
 enrolled = enrolled,  
 penetration\_rate = penetration\_rate  
 )  
  
csa\_df\_A <- add\_quadrants(csa\_df\_A, enrolled, penetration\_rate)  
med\_enrolled\_A <- unique(csa\_df\_A$.med\_x)  
med\_penrate\_A <- unique(csa\_df\_A$.med\_y)  
  
#Legend labels include counts  
tab\_q\_A <- table(csa\_df\_A$quadrant)  
legend\_labels\_A <- paste0(names(tab\_q\_A), " (n = ", as.integer(tab\_q\_A), ")") |>  
 stats::setNames(names(tab\_q\_A))  
  
label\_df\_A <- csa\_df\_A |>  
 dplyr::transmute(name = csa, dplyr::across(c(enrolled, penetration\_rate, quadrant))) |>  
 pick\_representatives\_by\_quadrant(lbl\_col = "name", x = enrolled, y = penetration\_rate)  
  
scatter\_plot\_A <- ggplot(  
 csa\_df\_A, aes(enrolled, penetration\_rate, color = quadrant)  
) +  
 geom\_point(size = 3, alpha = 0.8) +  
 geom\_vline(xintercept = med\_enrolled\_A, linetype = "dashed") +  
 geom\_hline(yintercept = med\_penrate\_A, linetype = "dashed") +  
 scale\_x\_log10(  
 breaks = scales::log\_breaks(n = 6, base = 10),  
 labels = scales::comma\_format()  
 ) +  
 scale\_y\_continuous(  
 limits = c(0, 1),  
 breaks = seq(0, 1, 0.1),  
 labels = scales::percent\_format(accuracy = 1)  
 ) +  
 scale\_color\_manual(  
 values = quadrant\_colors,  
 labels = legend\_labels\_A,  
 name = "Quadrant"  
 ) +  
 ggrepel::geom\_label\_repel(  
 data = label\_df\_A,  
 aes(label = label),  
 fill = "white",  
 colour = "black",  
 size = 3,  
 box.padding = 0.35,  
 point.padding = 0.3,  
 segment.alpha = 0.8,  
 min.segment.length = 0,  
 force = 1.2  
 ) +  
 labs(  
 title = "Medicare Advantage Enrollment vs. Penetration Rate by CSA",  
 subtitle = paste0(  
 "Quadrants split at medians (Enrolled = ",  
 scales::comma(med\_enrolled\_A), ", Penetration = ",  
 scales::percent(med\_penrate\_A, accuracy = 1), ")"  
 ),  
 x = "Total Enrolled Beneficiaries (log10 scale)",  
 y = "Penetration Rate"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 theme(  
 legend.position = "bottom",  
 legend.box.margin = margin(t = 8),  
 plot.margin = margin(t = 25, r = 10, b = 40, l = 10)  
 )  
  
scatter\_with\_marginals\_A <- ggExtra::ggMarginal(  
 scatter\_plot\_A,  
 type = "histogram",  
 margins = "both",  
 bins = 30,  
 fill = "white",  
 colour = "grey30"  
)  
  
print(scatter\_with\_marginals\_A)



ggsave(  
 filename = results\_path("MA\_enrollment\_penetration.png"),  
 plot = scatter\_with\_marginals\_A,  
 width = 15, height = 8, units = "in", dpi = 300  
)

**B) CSA (Merged) + Non-CSA by State — Enrollment vs. Penetration**

**Purpose**. Align with market-level thinking by merging split-state CSAs and by rolling non-CSA counties to a state “all-other” unit. This is often the cleanest view for policy scoping and stakeholder communication.

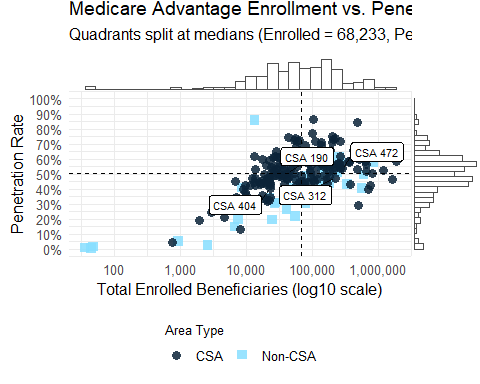
**How to read:**

Color/shape distinguish CSA vs Non-CSA.

Medians are recalculated on this merged set; expect shifts relative to A.

Marginal histograms show the density of enrollment and penetration.

raw\_B <- pen\_rows  
  
csa\_df\_B <- raw\_B |>  
 dplyr::mutate(  
 group\_id = dplyr::if\_else(  
 is\_non\_csa, paste0("All Other Areas - ", state), paste0("CSA ", csa\_code)  
 )  
 ) |>  
 dplyr::group\_by(group\_id, is\_non\_csa) |>  
 dplyr::summarise(  
 enrolled\_sum = sum(enrolled, na.rm = TRUE),  
 eligible\_sum = sum(eligible, na.rm = TRUE),  
 pen\_agg = dplyr::if\_else(eligible\_sum > 0, enrolled\_sum / eligible\_sum, NA\_real\_),  
 .groups = "drop"  
 ) |>  
 dplyr::transmute(  
 name = group\_id,  
 type = dplyr::if\_else(is\_non\_csa, "Non-CSA", "CSA"),  
 enrolled = enrolled\_sum,  
 penetration\_rate = pen\_agg  
 ) |>  
 dplyr::filter(!is.na(enrolled), !is.na(penetration\_rate), enrolled > 0, penetration\_rate >= 0)  
  
csa\_df\_B <- add\_quadrants(csa\_df\_B, enrolled, penetration\_rate)  
med\_enrolled\_B <- unique(csa\_df\_B$.med\_x)  
med\_penrate\_B <- unique(csa\_df\_B$.med\_y)  
  
label\_df\_B <- csa\_df\_B |>  
 dplyr::transmute(name, type, dplyr::across(c(enrolled, penetration\_rate, quadrant))) |>  
 pick\_representatives\_by\_quadrant(lbl\_col = "name", x = enrolled, y = penetration\_rate)  
  
scatter\_plot\_B <- ggplot(  
 csa\_df\_B, aes(x = enrolled, y = penetration\_rate, color = type, shape = type)  
) +  
 geom\_point(size = 3, alpha = 0.85) +  
 geom\_vline(xintercept = med\_enrolled\_B, linetype = "dashed") +  
 geom\_hline(yintercept = med\_penrate\_B, linetype = "dashed") +  
 scale\_x\_log10(  
 breaks = scales::log\_breaks(n = 6, base = 10),  
 labels = scales::comma\_format()  
 ) +  
 scale\_y\_continuous(  
 limits = c(0, 1),  
 breaks = seq(0, 1, 0.1),  
 labels = scales::percent\_format(accuracy = 1)  
 ) +  
 scale\_color\_manual(values = type\_colors, name = "Area Type") +  
 scale\_shape\_manual(values = c("CSA" = 16, "Non-CSA" = 15), name = "Area Type") +  
 ggrepel::geom\_label\_repel(  
 data = label\_df\_B,  
 aes(label = label), # <-- use 'label'  
 fill = "white",  
 colour = "black",  
 size = 3,  
 box.padding = 0.35,  
 point.padding = 0.3,  
 segment.alpha = 0.8,  
 min.segment.length = 0,  
 force = 1.2,  
 show.legend = FALSE  
 ) +  
 labs(  
 title = "Medicare Advantage Enrollment vs. Penetration Rate",  
 subtitle = paste0(  
 "Quadrants split at medians (Enrolled = ",  
 scales::comma(med\_enrolled\_B), ", Penetration = ",  
 scales::percent(med\_penrate\_B, accuracy = 1), ")."  
 ),  
 x = "Total Enrolled Beneficiaries (log10 scale)",  
 y = "Penetration Rate"  
 ) +  
 guides(  
 color = guide\_legend(order = 1, title.position = "top"),  
 shape = guide\_legend(order = 1, title.position = "top")  
 ) +  
 theme\_minimal(base\_size = 12) +  
 theme(  
 legend.position = "bottom",  
 legend.box = "horizontal",  
 legend.text = element\_text(size = 10),  
 legend.title = element\_text(size = 10),  
 legend.spacing.y = unit(2, "pt"),  
 legend.key.width = unit(1.2, "lines"),  
 plot.margin = margin(t = 25, r = 10, b = 40, l = 10)  
 )  
  
scatter\_with\_marginals\_B <- ggExtra::ggMarginal(  
 scatter\_plot\_B,  
 type = "histogram",  
 margins = "both",  
 bins = 30,  
 fill = "white",  
 colour = "grey30"  
)  
  
print(scatter\_with\_marginals\_B)



ggsave(  
 filename = results\_path("MA\_enrollment\_penetration\_CSA\_agg.png"),  
 plot = scatter\_with\_marginals\_B,  
 width = 15, height = 8.5, units = "in", dpi = 300  
)

**C) CSA-Only — Enrollment vs. Penetration (Non-CSA Removed)**

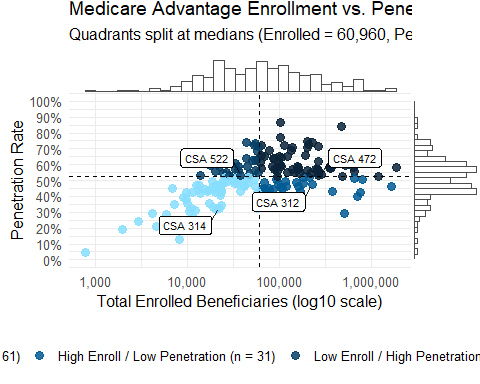
**Purpose**. Focus attention on true CSA markets only, removing the visual variability introduced by non-CSA areas. Use this when preparing market-focused design memos or when comparing CSAs to each other without rural/state “all-other” context.

**How to read:**

Same axes and median logic as B, but only CSAs are plotted.

Legend labels include n per quadrant for quick sizing.

raw\_C <- pen\_rows |>  
 dplyr::filter(is\_non\_csa == FALSE, !is.na(csa\_code))  
  
csa\_df\_C <- raw\_C |>  
 dplyr::group\_by(csa\_code) |>  
 dplyr::summarise(  
 enrolled\_sum = sum(enrolled, na.rm = TRUE),  
 eligible\_sum = sum(eligible, na.rm = TRUE),  
 pen\_agg = dplyr::if\_else(eligible\_sum > 0, enrolled\_sum / eligible\_sum, NA\_real\_),  
 .groups = "drop"  
 ) |>  
 dplyr::transmute(  
 name = paste0("CSA ", csa\_code),  
 enrolled = enrolled\_sum,  
 penetration\_rate = pen\_agg  
 ) |>  
 dplyr::filter(!is.na(enrolled), !is.na(penetration\_rate), enrolled > 0, penetration\_rate >= 0)  
  
csa\_df\_C <- add\_quadrants(csa\_df\_C, enrolled, penetration\_rate)  
med\_enrolled\_C <- unique(csa\_df\_C$.med\_x)  
med\_penrate\_C <- unique(csa\_df\_C$.med\_y)  
  
tab\_q\_C <- table(csa\_df\_C$quadrant)  
legend\_labels\_C <- paste0(names(tab\_q\_C), " (n = ", as.integer(tab\_q\_C), ")") |>  
 stats::setNames(names(tab\_q\_C))  
  
label\_df\_C <- csa\_df\_C |>  
 dplyr::transmute(name, dplyr::across(c(enrolled, penetration\_rate, quadrant))) |>  
 pick\_representatives\_by\_quadrant(lbl\_col = "name", x = enrolled, y = penetration\_rate)  
  
scatter\_plot\_C <- ggplot(  
 csa\_df\_C, aes(x = enrolled, y = penetration\_rate, color = quadrant)  
) +  
 geom\_point(size = 3, alpha = 0.85) +  
 geom\_vline(xintercept = med\_enrolled\_C, linetype = "dashed") +  
 geom\_hline(yintercept = med\_penrate\_C, linetype = "dashed") +  
 scale\_x\_log10(  
 breaks = scales::log\_breaks(n = 6, base = 10),  
 labels = scales::comma\_format()  
 ) +  
 scale\_y\_continuous(  
 limits = c(0, 1),  
 breaks = seq(0, 1, 0.1),  
 labels = scales::percent\_format(accuracy = 1)  
 ) +  
 scale\_color\_manual(  
 values = quadrant\_colors,  
 labels = legend\_labels\_C,  
 name = "Quadrant"  
 ) +  
 ggrepel::geom\_label\_repel(  
 data = label\_df\_C,  
 aes(label = label),  
 fill = "white",  
 colour = "black",  
 size = 3,  
 box.padding = 0.35,  
 point.padding = 0.3,  
 segment.alpha = 0.8,  
 min.segment.length = 0,  
 force = 1.2,  
 show.legend = FALSE  
 ) +  
 labs(  
 title = "Medicare Advantage Enrollment vs. Penetration Rate",  
 subtitle = paste0(  
 "Quadrants split at medians (Enrolled = ",  
 scales::comma(med\_enrolled\_C), ", Penetration = ",  
 scales::percent(med\_penrate\_C, accuracy = 1), ")."  
 ),  
 x = "Total Enrolled Beneficiaries (log10 scale)",  
 y = "Penetration Rate"  
 ) +  
 guides(color = guide\_legend(order = 1)) +  
 theme\_minimal(base\_size = 12) +  
 theme(  
 legend.position = "bottom",  
 legend.box.margin = margin(t = 8),  
 plot.margin = margin(t = 25, r = 10, b = 40, l = 10)  
 )  
  
scatter\_with\_marginals\_C <- ggExtra::ggMarginal(  
 scatter\_plot\_C,  
 type = "histogram",  
 margins = "both",  
 bins = 30,  
 fill = "white",  
 colour = "grey30"  
)  
  
print(scatter\_with\_marginals\_C)



ggsave(  
 filename = results\_path("MA\_enrollment\_penetration\_CSA\_only.png"),  
 plot = scatter\_with\_marginals\_C,  
 width = 15, height = 8, units = "in", dpi = 300  
)