CSA Enrollment and Penetration Maps

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

2025-08-05

**Executive Summary**

Why this matters: Medicare Advantage (MA) enrollment is concentrated in distinct regional markets that do not always follow state lines. To evaluate where new payment demonstrations might work best, we aggregated county-level MA data (eligibility, enrollment, penetration rates) up to Combined Statistical Areas (CSAs)— a market-like geography the Centers for Medicare & Medicaid Innovation (CMMI) is using to explore a Value Incentive Program (VIP).

This document walks through: (1) how the data are harmonized and validated, (2) how CSAs are assembled from counties, (3) three complementary map views of MA enrollment and penetration, and (4) optional overlays that highlight CSA clusters based on dual eligibility and VIP feasibility flags.

**Data Lineage & Study Scope**

**Source & preparation**. Prior to this README, a spreadsheet was created to clean and standardize county-level MA data and then re-aggregate to CSAs. For counties not assigned to a CSA, values are aggregated at the state “all other” level so the maps maintain complete geographic coverage.

**What’s in the file**. Each record carries:

enrolled – total MA enrollees (count). pen\_rate – MA penetration among eligible beneficiaries (0–100%). csa\_number – the three-digit CSA code when applicable. state\_abbr – two-letter state code (also used for “all other” areas).

**Geographic scope**. Alaska, Hawaii, and U.S. territories are excluded from mapping to keep the contiguous U.S. focus (the full logic remains in code so scope can easily be extended).

**Analytical objective**. Provide market-readable maps that (i) visualize enrollment concentration, (ii) contrast penetration rates, and (iii) surface potential market demonstration areas.

**Reproducibility & Environment**

This section installs/loads required packages, sets caching and GIS options, and defines helper utilities (e.g., a results\_path() function) used throughout.

##########  
#Packages and Global Options  
##########  
required <- c(  
 "tidyverse", "readxl", "readr", "janitor", "sf", "tigris",  
 "stringr", "RColorBrewer", "rlang", "scales"  
)  
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] "readxl" "lubridate" "forcats" "stringr" "dplyr" "purrr"   
## [7] "readr" "tidyr" "tibble" "ggplot2" "tidyverse" "stats"   
## [13] "graphics" "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[4]]  
## [1] "janitor" "readxl" "lubridate" "forcats" "stringr" "dplyr"   
## [7] "purrr" "readr" "tidyr" "tibble" "ggplot2" "tidyverse"  
## [13] "stats" "graphics" "grDevices" "utils" "datasets" "methods"   
## [19] "base"   
##   
## [[5]]  
## [1] "sf" "janitor" "readxl" "lubridate" "forcats" "stringr"   
## [7] "dplyr" "purrr" "readr" "tidyr" "tibble" "ggplot2"   
## [13] "tidyverse" "stats" "graphics" "grDevices" "utils" "datasets"   
## [19] "methods" "base"   
##   
## [[6]]  
## [1] "tigris" "sf" "janitor" "readxl" "lubridate" "forcats"   
## [7] "stringr" "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [13] "ggplot2" "tidyverse" "stats" "graphics" "grDevices" "utils"   
## [19] "datasets" "methods" "base"   
##   
## [[7]]  
## [1] "tigris" "sf" "janitor" "readxl" "lubridate" "forcats"   
## [7] "stringr" "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [13] "ggplot2" "tidyverse" "stats" "graphics" "grDevices" "utils"   
## [19] "datasets" "methods" "base"   
##   
## [[8]]  
## [1] "RColorBrewer" "tigris" "sf" "janitor" "readxl"   
## [6] "lubridate" "forcats" "stringr" "dplyr" "purrr"   
## [11] "readr" "tidyr" "tibble" "ggplot2" "tidyverse"   
## [16] "stats" "graphics" "grDevices" "utils" "datasets"   
## [21] "methods" "base"   
##   
## [[9]]  
## [1] "rlang" "RColorBrewer" "tigris" "sf" "janitor"   
## [6] "readxl" "lubridate" "forcats" "stringr" "dplyr"   
## [11] "purrr" "readr" "tidyr" "tibble" "ggplot2"   
## [16] "tidyverse" "stats" "graphics" "grDevices" "utils"   
## [21] "datasets" "methods" "base"   
##   
## [[10]]  
## [1] "scales" "rlang" "RColorBrewer" "tigris" "sf"   
## [6] "janitor" "readxl" "lubridate" "forcats" "stringr"   
## [11] "dplyr" "purrr" "readr" "tidyr" "tibble"   
## [16] "ggplot2" "tidyverse" "stats" "graphics" "grDevices"   
## [21] "utils" "datasets" "methods" "base"

options(tigris\_use\_cache = TRUE, tigris\_class = "sf")  
  
#helper to get results saved to correct folder  
results\_path <- function(filename) file.path(results\_dir, filename)

**Paths, Inputs, and Outputs**

This block reads a single DATA\_LOCATION.txt that points to a shared Teams directory, verifies that Excel file exists, and creates a Results/Maps folder for images.

##########  
#File Paths  
##########  
#DATA\_LOCATION.txt contains the file path to the Teams Data folder  
stopifnot(file.exists("DATA\_LOCATION.txt"))  
data\_location <- readr::read\_file("DATA\_LOCATION.txt") |> stringr::str\_trim()  
  
#load the file from Processed folder  
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.")  
}  
  
#make sure results can save to the right folder  
results\_dir <- file.path(data\_location, "Results", "Maps")  
if (!dir.exists(results\_dir)) dir.create(results\_dir, recursive = TRUE)

**Helper Functions (Data & Maps)**

This section defines three building blocks:

**1. parse\_pen\_rate()** Ensures the penetration column is numeric, rescales decimals to percentages when needed, and clamps values to [0, 100]. This protects the visuals from formatting inconsistencies (e.g., “12%” vs “0.12”).

**2. build\_df()** Normalizes column names, extracts the CSA code and state abbreviation from your label field, and enforces a consistent pen\_rate. This keeps data hygiene inside the pipeline so plots won’t break if column names drift slightly.

**3. load\_geographies()** Downloads counties, states, and CSA boundaries, then assigns each county to a CSA via centroid-in-polygon. This method is robust for mapping and avoids edge cases where complex county geometries straddle CSA borders.

**4. Map styling helpers.** make\_map\_cont() renders a continuous color scale (good for penetration). make\_map\_cat() renders labeled categories/bins (good for enrollment quintiles). Both functions support optional state outlines to help orient viewers who think in state boundaries.

##########  
#Shared Functions  
##########  
  
#Parse through the penetration rate column  
#(numeric vs character, decimal vs percent storage, and clamps to [0,100]).  
parse\_pen\_rate <- function(df\_raw) {  
 #rlang's %||% returns LHS unless NULL  
 pen\_col <- rlang::`%||%`(df\_raw$penetrration\_rate, df\_raw$penetration\_rate)  
  
 if (is.character(pen\_col)) {  
 pen\_rate\_num <- readr::parse\_number(pen\_col)  
 } else if (is.numeric(pen\_col)) {  
 pen\_rate\_num <- as.numeric(pen\_col)  
 } else {  
 pen\_rate\_num <- as.numeric(pen\_col)  
 }  
  
 max\_pen <- suppressWarnings(max(pen\_rate\_num, na.rm = TRUE))  
 #If stored as decimals (e.g., 0.12), scale to percent (12)  
 if (is.finite(max\_pen) && max\_pen <= 1.05) pen\_rate\_num <- pen\_rate\_num \* 100  
  
 #Clamp to [0, 100]  
 pmin(pmax(pen\_rate\_num, 0), 100)  
}  
  
#Build the working analysis frame from the Excel file  
build\_df <- function(path, sheet = "Sheet1") {  
 df\_raw <- readxl::read\_excel(path, sheet = sheet) |> janitor::clean\_names()  
 pen\_rate\_num <- parse\_pen\_rate(df\_raw)  
  
 df\_raw |>  
 transmute(  
 csa\_area = stringr::str\_trim(csa\_area),  
 enrolled = sum\_of\_enrolled,  
 pen\_rate = pen\_rate\_num,  
 csa\_number = stringr::str\_extract(csa\_area, "\\d+"),  
 state\_abbr = stringr::str\_extract(csa\_area, "[A-Z]{2}$")  
 ) |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
}  
  
#Load in geographices once (counties, states, CSA boundaries) and assign counties to CSAs via centroids  
load\_geographies <- function() {  
 drop\_fips <- c("02","15","72","78","60","66","69") #AK, HI, PR, VI, AS, GU, MP  
  
 state\_lookup <- tigris::fips\_codes |>  
 distinct(state\_code, state) |>  
 rename(STATEFP = state\_code, state\_abbr = state) #'state' is 2-letter abbr  
  
 counties\_sf <- tigris::counties(cb = TRUE, year = 2023, refresh = TRUE) |>  
 filter(!STATEFP %in% drop\_fips) |>  
 left\_join(state\_lookup, by = "STATEFP") |>  
 mutate(geoid = GEOID)  
  
 states\_sf <- tigris::states(cb = TRUE, year = 2023, refresh = TRUE) |>  
 filter(!STATEFP %in% drop\_fips)  
  
 csa\_sf <- tigris::combined\_statistical\_areas(cb = TRUE, year = 2023) |>  
 select(csa\_number = GEOID)  
  
 #Assign counties to CSAs via centroid-in-polygon  
 county\_centroids <- sf::st\_centroid(counties\_sf) |>  
 sf::st\_join(csa\_sf, left = TRUE) |>  
 sf::st\_drop\_geometry() |>  
 select(geoid, csa\_number)  
  
 counties\_sf <- counties\_sf |>  
 left\_join(county\_centroids, by = "geoid")  
  
 list(counties\_sf = counties\_sf, states\_sf = states\_sf, csa\_sf = csa\_sf)  
}  
  
#Standardized map theming with parameterized legend behavior.  
make\_map\_cont <- function(  
 sf\_obj, var, title, legend\_lab, palette,  
 pen\_display = c("percent","share"),  
 overlay\_states = FALSE, states\_layer = NULL,  
 na\_gray = FALSE  
) {  
 pen\_display <- match.arg(pen\_display)  
 var\_quo <- rlang::enquo(var)  
  
 p <- ggplot(sf\_obj) +  
 geom\_sf(aes(fill = !!var\_quo), color = "white", size = 0.15)  
  
 if (overlay\_states && !is.null(states\_layer)) {  
 p <- p + geom\_sf(data = states\_layer, fill = NA, color = "grey60", size = 0.2, inherit.aes = FALSE)  
 }  
  
 if (pen\_display == "percent") {  
 #Map A behavior: no fixed limits/breaks; legend "Penetration %"  
 p <- p +  
 scale\_fill\_gradientn(  
 colours = palette,  
 name = legend\_lab,  
 na.value = if (na\_gray) "grey80" else "grey70"  
 )  
 } else {  
 #Maps B/C behavior: limits [0, 100]; 20/40/60/80 ticks; labeled as shares  
 p <- p +  
 scale\_fill\_gradientn(  
 colours = palette,  
 name = legend\_lab,  
 limits = c(0, 100),  
 breaks = c(20, 40, 60, 80),  
 labels = function(x) scales::number(x / 100, accuracy = 0.1),  
 na.value = if (na\_gray) "grey80" else "grey70",  
 oob = scales::squish,  
 guide = guide\_colorbar(reverse = FALSE)  
 )  
 }  
  
 p +  
 coord\_sf(xlim = c(-125, -66), ylim = c(24, 50), expand = FALSE) +  
 labs(title = title, subtitle = "Geographic Distribution", caption = "Data: CMS, 2025") +  
 theme\_void(base\_size = 12) +  
 theme(  
 plot.title = element\_text(size = 15, face = "bold", hjust = .5),  
 plot.subtitle = element\_text(hjust = .5),  
 legend.title = element\_text(size = 11),  
 legend.text = element\_text(size = 9)  
 )  
}  
  
make\_map\_cat <- function(  
 sf\_obj, var, title, legend\_lab, palette,  
 overlay\_states = FALSE, states\_layer = NULL,  
 na\_translate = FALSE, na\_color = "grey80"  
) {  
 p <- ggplot(sf\_obj) +  
 geom\_sf(aes(fill = {{ var }}), color = "white", size = 0.15)  
  
 if (overlay\_states && !is.null(states\_layer)) {  
 p <- p + geom\_sf(data = states\_layer, fill = NA, color = "grey60", size = 0.2, inherit.aes = FALSE)  
 }  
  
 p +  
 scale\_fill\_manual(values = palette, name = legend\_lab,  
 drop = FALSE, na.translate = na\_translate, na.value = na\_color) +  
 coord\_sf(xlim = c(-125, -66), ylim = c(24, 50), expand = FALSE) +  
 labs(title = title, subtitle = "Geographic Distribution", caption = "Data: CMS, 2025") +  
 theme\_void(base\_size = 12) +  
 theme(  
 plot.title = element\_text(size = 15, face = "bold", hjust = .5),  
 plot.subtitle = element\_text(hjust = .5),  
 legend.title = element\_text(size = 11),  
 legend.text = element\_text(size = 9)  
 )  
}  
  
#Reusable color palettes!  
blues\_q <- RColorBrewer::brewer.pal(5, "Blues")  
blues\_pct <- colorRampPalette(RColorBrewer::brewer.pal(9, "Blues")[-1])(100)

**Load Data & Geographies**

At this point, the cleaned spreadsheet is read, geometries are fetched once, and— if available— lwgeom is used to validate shapes. This reduces the risk of topology errors when merging counties into larger polygons.

**How to read downstream logs**: Occasionally there are console outputs summarizing bin cut-offs and quintile thresholds. Those are deliberate diagnostics to keep transparency about how the bins were defined in each run.

###########  
#Load Data + Geographies  
###########  
df <- build\_df(penrate\_path, sheet = "Sheet1")  
geo <- load\_geographies()

## | | | 0% | | | 1% | |= | 1% | |= | 2% | |== | 2% | |== | 3% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 7% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======= | 11% | |======== | 11% | |======== | 12% | |========= | 12% | |========= | 13% | |========== | 14% | |========== | 15% | |=========== | 15% | |=========== | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | |============== | 19% | |============== | 20% | |============== | 21% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | |================ | 24% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 26% | |=================== | 27% | |=================== | 28% | |==================== | 28% | |==================== | 29% | |===================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================= | 34% | |======================== | 34% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | |=========================== | 39% | |============================ | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 42% | |============================== | 43% | |============================== | 44% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 49% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 52% | |===================================== | 53% | |===================================== | 54% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 56% | |======================================== | 57% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |========================================== | 61% | |=========================================== | 61% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 67% | |=============================================== | 68% | |================================================ | 68% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================= | 71% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 77% | |====================================================== | 78% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |========================================================== | 84% | |=========================================================== | 84% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 87% | |============================================================= | 88% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |=============================================================== | 91% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | |================================================================= | 93% | |================================================================= | 94% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 99% | |======================================================================| 100%  
## | | | 0% | |= | 1% | |= | 2% | |== | 3% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 7% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======== | 11% | |======== | 12% | |========= | 12% | |========= | 13% | |========= | 14% | |========== | 14% | |========== | 15% | |=========== | 15% | |=========== | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | |============== | 20% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 27% | |==================== | 28% | |==================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================== | 34% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | |=========================== | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 43% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 53% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 57% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |=========================================== | 61% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 67% | |================================================ | 68% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 77% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |=========================================================== | 84% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 87% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | |================================================================= | 93% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 100%

counties\_sf <- geo$counties\_sf  
states\_sf <- geo$states\_sf  
csa\_sf <- geo$csa\_sf  
  
#Validate geometries to avoid pathological unions  
if (requireNamespace("lwgeom", quietly = TRUE)) {  
 counties\_sf <- sf::st\_make\_valid(counties\_sf)  
}

**A) CSA+1 Maps — Enrollment Quintiles & Penetration (State-Split Preserved)**

**Goal**. Show fine-grained variation inside multi-state CSAs by keeping each CSA×State as a separate market polygon (“CSA+1”). Non-CSA counties are aggregated to a state ‘all other’ region, so the map remains wall-to-wall.

**What the code does.**

1. Dissolves counties to CSA×State units and attaches the associated enrollment and penetration for each unit.
2. Builds a companion layer for non-CSA counties aggregated to the state.
3. Computes dynamic enrollment quintiles based on the actual distribution in the loaded data.
4. Produces two complementary maps: Enrollment (categorical) with data-driven quintiles. Penetration (continuous) with a percentage legend (0–100, auto-scaled).

**How to read this graphic:** Darker blues in the enrollment map mean larger MA markets (by headcount). Darker blues in the penetration map mean higher MA share of eligible beneficiaries.

Because CSA+1 preserves state splits, you can see within-CSA, cross-state differences that would be hidden if we merged them.

**Why both maps**: Enrollment highlights market size (implementation scale, potential absolute impact). Penetration highlights market adoption (policy acceptance, competitive intensity). Together they help separate big-but-low-share markets from small-but-high-share niches.

#County --> CSA (by state)  
county\_with\_csa\_dissolved\_A <- counties\_sf |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 left\_join(  
 df |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 select(csa\_number, state\_abbr, enrolled, pen\_rate),  
 by = c("csa\_number", "state\_abbr")  
 ) |>  
 group\_by(csa\_number, state\_abbr) |>  
 summarise(  
 enrolled = dplyr::first(enrolled),  
 pen\_rate = dplyr::first(pen\_rate),  
 .groups = "drop"  
 )  
  
#Non-CSA --> "all other" by state  
state\_other\_vals\_A <- df |>  
 filter(is.na(csa\_number)) |>  
 select(state\_abbr, enrolled, pen\_rate)  
  
county\_no\_csa\_dissolved\_A <- counties\_sf |>  
 filter(is.na(csa\_number)) |>  
 left\_join(state\_other\_vals\_A, by = "state\_abbr") |>  
 group\_by(state\_abbr) |>  
 summarise(  
 enrolled = dplyr::first(enrolled),  
 pen\_rate = dplyr::first(pen\_rate),  
 .groups = "drop"  
 )  
  
#final map   
plot\_df\_A <- bind\_rows(  
 county\_with\_csa\_dissolved\_A,  
 county\_no\_csa\_dissolved\_A  
) |>  
 st\_as\_sf() |>  
 mutate(  
 enroll\_q = dplyr::ntile(enrolled, 5),  
 enroll\_q = factor(  
 enroll\_q, levels = 1:5,  
 labels = c("Q1 (Lowest)", "Q2", "Q3", "Q4", "Q5 (Highest)")  
 )  
 )  
  
#diagnostics  
cat("\n===== Enrollment Quintile Cut-points =====\n")

##   
## ===== Enrollment Quintile Cut-points =====

quintile\_breaks\_A <- quantile(plot\_df\_A$enrolled, probs = seq(0, 1, 0.20), na.rm = TRUE)  
print(quintile\_breaks\_A)

## 0% 20% 40% 60% 80% 100%   
## 37 13793 32966 77338 162618 1865030

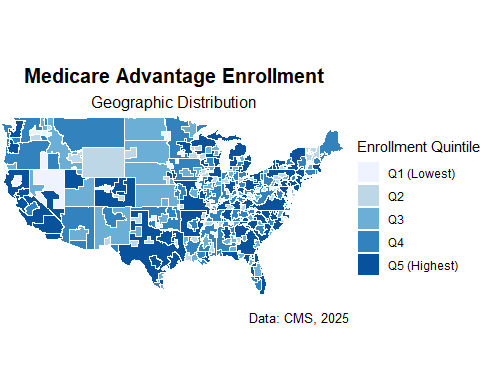
cat("\n===== Min / Max Enrollment in Each Quintile =====\n")

##   
## ===== Min / Max Enrollment in Each Quintile =====

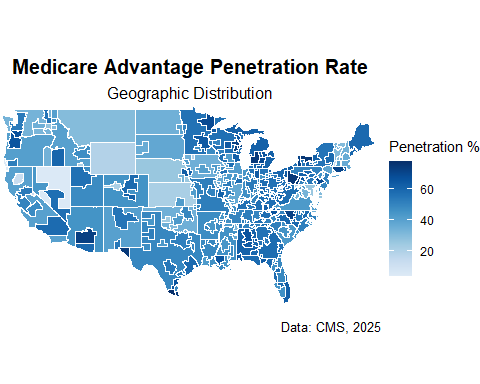
quintile\_ranges\_A <- plot\_df\_A |>  
 st\_drop\_geometry() |>  
 group\_by(enroll\_q) |>  
 summarise(  
 min\_enrolled = min(enrolled, na.rm = TRUE),  
 max\_enrolled = max(enrolled, na.rm = TRUE),  
 n\_areas = dplyr::n(),  
 .groups = "drop"  
 ) |>  
 arrange(enroll\_q)  
print(quintile\_ranges\_A)

## # A tibble: 6 × 4  
## enroll\_q min\_enrolled max\_enrolled n\_areas  
## <fct> <dbl> <dbl> <int>  
## 1 Q1 (Lowest) 37 13793 59  
## 2 Q2 14746 32966 58  
## 3 Q3 32989 77338 58  
## 4 Q4 77472 162618 58  
## 5 Q5 (Highest) 164174 1865030 58  
## 6 <NA> Inf -Inf 1

#maps  
enrollment\_map\_A <- make\_map\_cat(  
 sf\_obj = plot\_df\_A,  
 var = enroll\_q,  
 title = "Medicare Advantage Enrollment",  
 legend\_lab = "Enrollment Quintile",  
 palette = blues\_q,  
 overlay\_states = FALSE  
)  
  
penetration\_map\_A <- make\_map\_cont(  
 sf\_obj = plot\_df\_A,  
 var = pen\_rate,  
 title = "Medicare Advantage Penetration Rate",  
 legend\_lab = "Penetration %",  
 palette = blues\_pct,  
 pen\_display = "percent", # Section A: "percent" legend behavior  
 overlay\_states = FALSE  
)  
  
print(enrollment\_map\_A)



print(penetration\_map\_A)



ggsave(results\_path("enrollment\_map.png"), enrollment\_map\_A, width = 11, height = 6, dpi = 300)  
ggsave(results\_path("penetration\_map.png"), penetration\_map\_A, width = 11, height = 6, dpi = 300)

**B) CSA (Merged) — Fixed Enrollment Bins & Penetration (Shares)**

**Goal**. Present a market-level view by merging split-state CSAs into single units and summarizing penetration as an enrollment-weighted average.

**What the code does.**

Sums enrollment across a CSA (regardless of state).

Calculates penetration as a weighted mean (weights = enrolled), then enforces [0, 100].

Keeps non-CSA areas aggregated to their states for complete coverage.

Uses fixed enrollment ranges (e.g., “50k–100k”) instead of dynamic quintiles, which makes cross-run comparisons easier for stakeholders.

**How to read this graphic:**

Enrollment (fixed bins) provides a stable legend (e.g., “Q3: 50–100k”), which helps when you compare renders across months or scenarios.

Penetration (shares) fixes the legend to 0–1.00 (displayed as 0–100%), with reference ticks at 0.20/0.40/0.60/0.80, nudging viewers to think in fractions as well as percentages.

**When to use A vs B:**

Use A (CSA+1) to spotlight intra-CSA, cross-state variation.

Use B (CSA merged) when you want a single policy unit aligned to how demonstrations are scoped.

#Merge split-state CSAs on data (enrollment-weighted penetration)  
df\_csa\_merged\_B <- df |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 group\_by(csa\_number) |>  
 summarise(  
 enrolled = sum(enrolled, na.rm = TRUE),  
 pen\_rate = {  
 keep <- !is.na(pen\_rate) & !is.na(enrolled) & enrolled > 0  
 w <- sum(enrolled[keep], na.rm = TRUE)  
 if (w == 0) NA\_real\_  
 else sum(pen\_rate[keep] \* enrolled[keep], na.rm = TRUE) / w  
 },  
 .groups = "drop"  
 ) |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
  
#Remove state splits  
old\_s2 <- sf::sf\_use\_s2()  
sf::sf\_use\_s2(FALSE)  
  
county\_with\_csa\_dissolved\_B <- counties\_sf |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 group\_by(csa\_number) |>  
 summarise(.groups = "drop") |>  
 left\_join(df\_csa\_merged\_B, by = "csa\_number")  
  
sf::sf\_use\_s2(old\_s2)  
  
  
#non-CSA values kept at state level  
state\_other\_vals\_B <- df |>  
 filter(is.na(csa\_number)) |>  
 select(state\_abbr, enrolled, pen\_rate)  
  
county\_no\_csa\_dissolved\_B <- counties\_sf |>  
 filter(is.na(csa\_number)) |>  
 left\_join(state\_other\_vals\_B, by = "state\_abbr") |>  
 group\_by(state\_abbr) |>  
 summarise(  
 enrolled = dplyr::first(enrolled),  
 pen\_rate = dplyr::first(pen\_rate),  
 .groups = "drop"  
 ) |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
  
#fixed ranges for quintiles  
enroll\_breaks\_B <- c(770, 25000, 50000, 100000, 200000, 2000000)  
enroll\_labels\_B <- c(  
 "Q1: 770–25,000",  
 "Q2: 25,000–50,000",  
 "Q3: 50,000–100,000",  
 "Q4: 100,000–200,000",  
 "Q5: 200,000–2,000,000"  
)  
  
plot\_df\_B <- bind\_rows(  
 county\_with\_csa\_dissolved\_B,  
 county\_no\_csa\_dissolved\_B  
) |>  
 st\_as\_sf() |>  
 mutate(  
 pen\_rate = pmin(pmax(pen\_rate, 0), 100),  
 enroll\_q = case\_when(  
 enrolled < enroll\_breaks\_B[1] ~ enroll\_labels\_B[1],  
 enrolled >= enroll\_breaks\_B[6] ~ enroll\_labels\_B[5],  
 TRUE ~ as.character(cut(  
 enrolled,  
 breaks = enroll\_breaks\_B,  
 labels = enroll\_labels\_B,  
 right = FALSE,  
 include.lowest = TRUE  
 ))  
 ),  
 enroll\_q = factor(enroll\_q, levels = enroll\_labels\_B)  
 )  
  
cat("\n=== Enrollment fixed-bin counts (Q1–Q5) ===\n")

##   
## === Enrollment fixed-bin counts (Q1–Q5) ===

enr\_fixed\_counts\_B <- plot\_df\_B |>  
 st\_drop\_geometry() |>  
 group\_by(enroll\_q) |>  
 summarise(n\_areas = dplyr::n(), .groups = "drop")  
print(enr\_fixed\_counts\_B)

## # A tibble: 6 × 2  
## enroll\_q n\_areas  
## <fct> <int>  
## 1 Q1: 770–25,000 54  
## 2 Q2: 25,000–50,000 34  
## 3 Q3: 50,000–100,000 49  
## 4 Q4: 100,000–200,000 43  
## 5 Q5: 200,000–2,000,000 46  
## 6 <NA> 1

cat("\n=== Enrollment actual ranges by fixed bin (min–max) ===\n")

##   
## === Enrollment actual ranges by fixed bin (min–max) ===

enr\_fixed\_ranges\_B <- plot\_df\_B |>  
 st\_drop\_geometry() |>  
 group\_by(enroll\_q) |>  
 summarise(  
 bin\_min = min(enrolled, na.rm = TRUE),  
 bin\_max = max(enrolled, na.rm = TRUE),  
 .groups = "drop"  
 ) |>  
 arrange(enroll\_q) |>  
 mutate(  
 bin\_min = scales::comma(bin\_min, accuracy = 1),  
 bin\_max = scales::comma(bin\_max, accuracy = 1)  
 )  
print(enr\_fixed\_ranges\_B)

## # A tibble: 6 × 3  
## enroll\_q bin\_min bin\_max   
## <fct> <chr> <chr>   
## 1 Q1: 770–25,000 770 24,897   
## 2 Q2: 25,000–50,000 25,096 47,220   
## 3 Q3: 50,000–100,000 51,289 97,651   
## 4 Q4: 100,000–200,000 100,448 192,061   
## 5 Q5: 200,000–2,000,000 202,298 1,865,030  
## 6 <NA> Inf -Inf

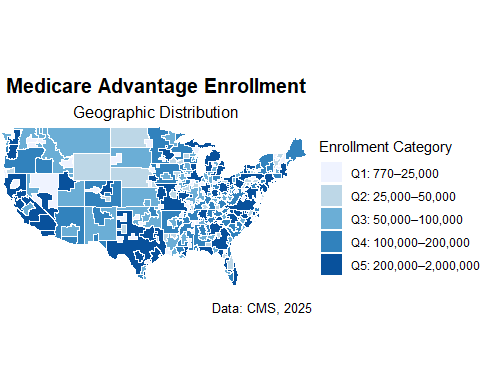
missing\_all\_other\_B <- counties\_sf |>  
 filter(is.na(csa\_number)) |>  
 distinct(state\_abbr) |>  
 anti\_join(df |> filter(is.na(csa\_number)) |> distinct(state\_abbr), by = "state\_abbr")  
cat("\nStates missing 'all other' penetration in the sheet:\n")

##   
## States missing 'all other' penetration in the sheet:

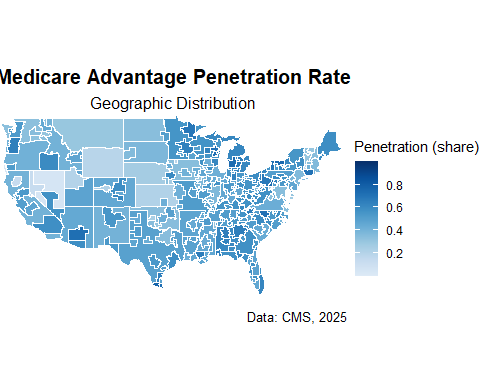
print(missing\_all\_other\_B)

## state\_abbr  
## 1 RI

#maps  
enrollment\_map\_B <- make\_map\_cat(  
 sf\_obj = plot\_df\_B,  
 var = enroll\_q,  
 title = "Medicare Advantage Enrollment",  
 legend\_lab = "Enrollment Category",  
 palette = blues\_q  
)  
  
penetration\_map\_B <- make\_map\_cont(  
 sf\_obj = plot\_df\_B,  
 var = pen\_rate,  
 title = "Medicare Advantage Penetration Rate",  
 legend\_lab = "Penetration (share)",  
 palette = blues\_pct,  
 pen\_display = "share"  
)  
  
print(enrollment\_map\_B)



print(penetration\_map\_B)



ggsave(results\_path("enrollment\_map\_csa-merged.png"), enrollment\_map\_B, width = 11, height = 6, dpi = 300)  
ggsave(results\_path("penetration\_map\_csa-merged.png"), penetration\_map\_B, width = 11, height = 6, dpi = 300)

**C) CSA-Only Maps + Centroid Overlays**

**Goal**. Focus exclusively on CSAs (non-CSA areas are shown in neutral grey) and layer centroid markers for four selections pulled from an ad-hoc workbook. These selections are driven by two flags:

dual\_lis\_flag (0/1) – indicates presence of dual-eligible/LIS populations. viable\_area\_units\_gt500\_… (0/1) – denotes feasibility thresholds for VIP (e.g., sufficient area units).

**What the code does.**

1. Merges each CSA’s counties into a single polygon and computes enrollment and weighted penetration.
2. Categorizes enrollment using the same fixed bins as section B (for comparability).
3. Creates four overlay layers by filtering the ad-hoc table and plotting CSA centroids on top of the base map:

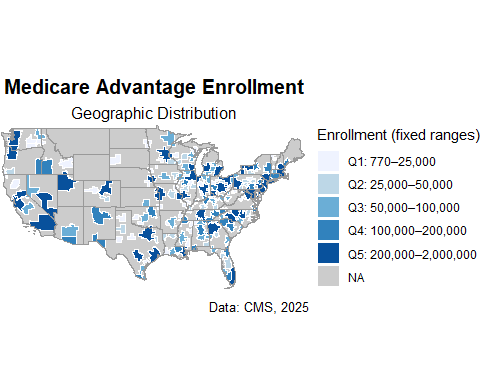
Dual 0, VIP Comp 0 Dual 0, Current VIP 0 Dual 1, VIP Comp 0 Dual 1, Current VIP 0

**How to read these graphics:**

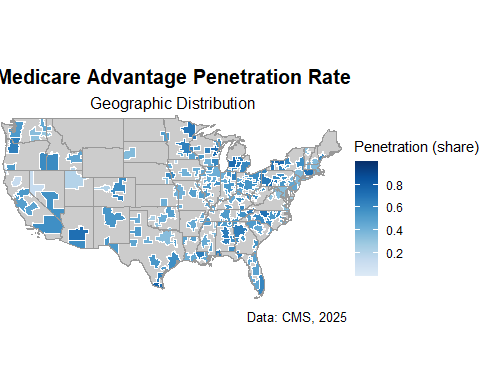
The base fill shows enrollment category across CSAs; grey indicates non-CSA areas.

Black dots mark CSAs that meet the specified selection criteria.

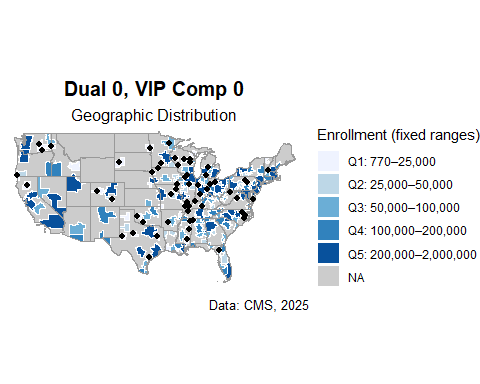
#Dissolve CSAs (merged across states) for CSA-only maps  
df\_csa\_merged\_C <- df |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 group\_by(csa\_number) |>  
 summarise(  
 enrolled = sum(enrolled, na.rm = TRUE),  
 pen\_rate = {  
 keep <- !is.na(pen\_rate) & !is.na(enrolled) & enrolled > 0  
 w <- sum(enrolled[keep], na.rm = TRUE)  
 if (w == 0) NA\_real\_  
 else sum(pen\_rate[keep] \* enrolled[keep], na.rm = TRUE) / w  
 },  
 .groups = "drop"  
 ) |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
  
old\_s2 <- sf::sf\_use\_s2()  
sf::sf\_use\_s2(FALSE)  
  
county\_with\_csa\_dissolved\_C <- counties\_sf |>  
 filter(!is.na(csa\_number)) |>  
 mutate(csa\_number = as.character(csa\_number)) |>  
 group\_by(csa\_number) |>  
 summarise(.groups = "drop") |>  
 left\_join(df\_csa\_merged\_C, by = "csa\_number")  
  
sf::sf\_use\_s2(old\_s2)  
  
state\_other\_vals\_C <- df |>  
 filter(is.na(csa\_number)) |>  
 select(state\_abbr, enrolled, pen\_rate)  
  
county\_no\_csa\_dissolved\_C <- counties\_sf |>  
 filter(is.na(csa\_number)) |>  
 left\_join(state\_other\_vals\_C, by = "state\_abbr") |>  
 group\_by(state\_abbr) |>  
 summarise(  
 enrolled = dplyr::first(enrolled),  
 pen\_rate = dplyr::first(pen\_rate),  
 .groups = "drop"  
 ) |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
  
plot\_df\_C <- bind\_rows(  
 county\_with\_csa\_dissolved\_C,  
 county\_no\_csa\_dissolved\_C  
) |>  
 st\_as\_sf() |>  
 mutate(pen\_rate = pmin(pmax(pen\_rate, 0), 100))  
  
#fixed ranges for quintiles  
enroll\_breaks\_C <- c(770, 25000, 50000, 100000, 200000, 2000000)  
enroll\_labels\_C <- c(  
 "Q1: 770–25,000",  
 "Q2: 25,000–50,000",  
 "Q3: 50,000–100,000",  
 "Q4: 100,000–200,000",  
 "Q5: 200,000–2,000,000"  
)  
  
csa\_only\_tbl <- plot\_df\_C |>  
 st\_drop\_geometry() |>  
 filter(!is.na(csa\_number))  
  
csa\_bins\_C <- csa\_only\_tbl |>  
 mutate(  
 enroll\_q = case\_when(  
 enrolled < enroll\_breaks\_C[1] ~ enroll\_labels\_C[1],  
 enrolled >= enroll\_breaks\_C[6] ~ enroll\_labels\_C[5],  
 TRUE ~ as.character(cut(  
 enrolled,  
 breaks = enroll\_breaks\_C,  
 labels = enroll\_labels\_C,  
 right = FALSE,  
 include.lowest = TRUE  
 ))  
 )  
 ) |>  
 mutate(enroll\_q = factor(enroll\_q, levels = enroll\_labels\_C)) |>  
 select(csa\_number, enroll\_q)  
  
plot\_df\_C <- plot\_df\_C |>  
 left\_join(csa\_bins\_C, by = "csa\_number") |>  
 mutate(  
 #Only CSAs get penetration color-- non-CSA areas are NA (grey)  
 pen\_rate\_csa = if\_else(is.na(csa\_number), NA\_real\_, pen\_rate)  
 )  
  
#maps  
enrollment\_map\_C <- make\_map\_cat(  
 sf\_obj = plot\_df\_C,  
 var = enroll\_q,  
 title = "Medicare Advantage Enrollment",  
 legend\_lab = "Enrollment (fixed ranges)",  
 palette = blues\_q,  
 overlay\_states = TRUE, states\_layer = states\_sf,   
 na\_translate = TRUE, na\_color = "grey80"  
)  
  
penetration\_map\_C <- make\_map\_cont(  
 sf\_obj = plot\_df\_C,  
 var = pen\_rate\_csa,  
 title = "Medicare Advantage Penetration Rate",  
 legend\_lab = "Penetration (share)",  
 palette = blues\_pct,  
 pen\_display = "share",  
 overlay\_states = TRUE, states\_layer = states\_sf,  
 na\_gray = TRUE  
)  
  
print(enrollment\_map\_C)



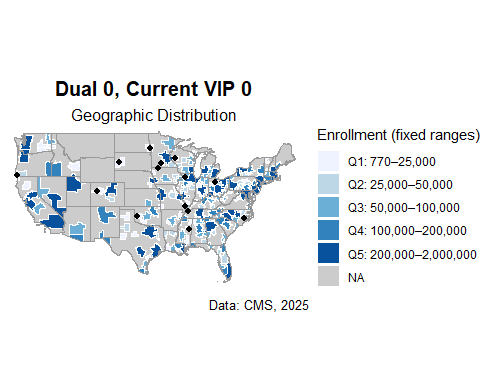
print(penetration\_map\_C)



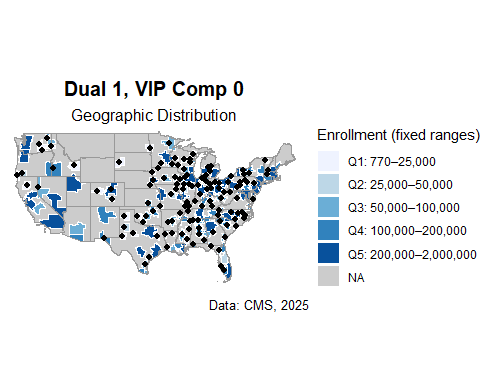
ggsave(results\_path("enrollment\_map\_csa-only.png"), enrollment\_map\_C, width = 11, height = 6, dpi = 300)  
ggsave(results\_path("penetration\_map\_csa-only.png"), penetration\_map\_C, width = 11, height = 6, dpi = 300)  
  
# =========  
# Centroids (dynamic selections from the spreadsheet)  
# =========  
  
dropoff\_path <- file.path(  
 data\_location, "Processed", "Ad Hoc", "DropoffCounts\_CSAPG\_ReportingUnits.xlsx"  
)  
if (!file.exists(dropoff\_path)) {  
 stop("File not found: ", dropoff\_path)  
}  
  
drop\_tbl <- readxl::read\_excel(dropoff\_path) |>  
 janitor::clean\_names() |>  
 #Only keep rows where the name contains "CSA"  
 dplyr::filter(stringr::str\_detect(csa\_area, stringr::regex("\\bCSA\\b", ignore\_case = TRUE))) |>  
 dplyr::mutate(  
 csa\_code = stringr::str\_extract(csa\_area, "\\b\\d{3}\\b"),  
 #Coerce filters to numeric so 0/1 tests work even if sheet typed as text  
 dual\_lis\_flag = as.numeric(dual\_lis\_flag),  
 viable\_area\_units\_gt500\_current\_vip\_comp = as.numeric(viable\_area\_units\_gt500\_current\_vip\_comp),  
 viable\_area\_units\_gt500\_current\_vip = as.numeric(viable\_area\_units\_gt500\_current\_vip)  
 )  
  
#Build the four selection vectors  
selected\_csas\_Dual0\_VIP0 <- drop\_tbl |>  
 dplyr::filter(dual\_lis\_flag == 0, viable\_area\_units\_gt500\_current\_vip\_comp == 0) |>  
 dplyr::pull(csa\_code) |> unique() |> stats::na.omit() |> as.character()  
  
selected\_csas\_Dual1\_Curr <- drop\_tbl |>  
 dplyr::filter(dual\_lis\_flag == 1, viable\_area\_units\_gt500\_current\_vip == 0) |>  
 dplyr::pull(csa\_code) |> unique() |> stats::na.omit() |> as.character()  
  
selected\_csas\_Dual1\_VIP0 <- drop\_tbl |>  
 dplyr::filter(dual\_lis\_flag == 1, viable\_area\_units\_gt500\_current\_vip\_comp == 0) |>  
 dplyr::pull(csa\_code) |> unique() |> stats::na.omit() |> as.character()  
  
selected\_csas\_Dual0\_Curr <- drop\_tbl |>  
 dplyr::filter(dual\_lis\_flag == 0, viable\_area\_units\_gt500\_current\_vip == 0) |>  
 dplyr::pull(csa\_code) |> unique() |> stats::na.omit() |> as.character()  
  
#Convert those selections to centroid layers  
# --- Dual 0, VIP Comp 0 ---  
csa\_centroids\_Dual0\_VIP0 <- county\_with\_csa\_dissolved\_C |>  
 dplyr::filter(!is.na(csa\_number), csa\_number %in% selected\_csas\_Dual0\_VIP0) |>  
 sf::st\_point\_on\_surface()  
  
dual0\_vip0\_map <- make\_map\_cat(  
 plot\_df\_C, enroll\_q,  
 title = "Dual 0, VIP Comp 0",  
 legend\_lab = "Enrollment (fixed ranges)",  
 palette = blues\_q,  
 overlay\_states = TRUE, states\_layer = states\_sf,  
 na\_translate = TRUE, na\_color = "grey80"  
) +  
 geom\_sf(data = csa\_centroids\_Dual0\_VIP0, color = "black", size = 1.8, shape = 16, inherit.aes = FALSE)  
  
print(dual0\_vip0\_map)



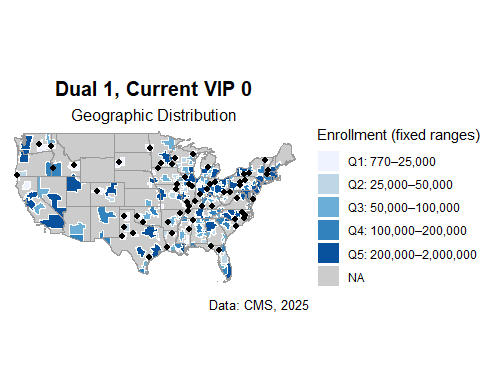
ggsave(results\_path("dual0\_vip0\_map.png"), dual0\_vip0\_map, width = 11, height = 6, dpi = 300)  
  
# --- Dual 0, Current VIP 0 ---  
csa\_centroids\_Dual0\_Curr <- county\_with\_csa\_dissolved\_C |>  
 dplyr::filter(!is.na(csa\_number), csa\_number %in% selected\_csas\_Dual0\_Curr) |>  
 sf::st\_point\_on\_surface()  
  
dual0\_curr\_vip0\_map <- make\_map\_cat(  
 plot\_df\_C, enroll\_q,  
 title = "Dual 0, Current VIP 0",  
 legend\_lab = "Enrollment (fixed ranges)",  
 palette = blues\_q,  
 overlay\_states = TRUE, states\_layer = states\_sf,  
 na\_translate = TRUE, na\_color = "grey80"  
) +  
 geom\_sf(data = csa\_centroids\_Dual0\_Curr, color = "black", size = 1.8, shape = 16, inherit.aes = FALSE)  
  
print(dual0\_curr\_vip0\_map)



ggsave(results\_path("dual0\_curr\_vip0\_map.png"), dual0\_curr\_vip0\_map, width = 11, height = 6, dpi = 300)  
  
# --- Dual 1, VIP Comp 0 ---  
csa\_centroids\_Dual1\_VIP0 <- county\_with\_csa\_dissolved\_C |>  
 dplyr::filter(!is.na(csa\_number), csa\_number %in% selected\_csas\_Dual1\_VIP0) |>  
 sf::st\_point\_on\_surface()  
  
dual1\_vip0\_map <- make\_map\_cat(  
 plot\_df\_C, enroll\_q,  
 title = "Dual 1, VIP Comp 0",  
 legend\_lab = "Enrollment (fixed ranges)",  
 palette = blues\_q,  
 overlay\_states = TRUE, states\_layer = states\_sf,  
 na\_translate = TRUE, na\_color = "grey80"  
) +  
 geom\_sf(data = csa\_centroids\_Dual1\_VIP0, color = "black", size = 1.8, shape = 16, inherit.aes = FALSE)  
  
print(dual1\_vip0\_map)



ggsave(results\_path("dual1\_vip0\_map.png"), dual1\_vip0\_map, width = 11, height = 6, dpi = 300)  
  
# --- Dual 1, Current VIP 0 ---  
csa\_centroids\_Dual1\_Curr <- county\_with\_csa\_dissolved\_C |>  
 dplyr::filter(!is.na(csa\_number), csa\_number %in% selected\_csas\_Dual1\_Curr) |>  
 sf::st\_point\_on\_surface()  
  
dual1\_curr\_vip0\_map <- make\_map\_cat(  
 plot\_df\_C, enroll\_q,  
 title = "Dual 1, Current VIP 0",  
 legend\_lab = "Enrollment (fixed ranges)",  
 palette = blues\_q,  
 overlay\_states = TRUE, states\_layer = states\_sf,  
 na\_translate = TRUE, na\_color = "grey80"  
) +  
 geom\_sf(data = csa\_centroids\_Dual1\_Curr, color = "black", size = 1.8, shape = 16, inherit.aes = FALSE)  
  
print(dual1\_curr\_vip0\_map)



ggsave(results\_path("dual1\_curr\_vip0\_map.png"), dual1\_curr\_vip0\_map, width = 11, height = 6, dpi = 300)