

```
---
title: "Treatment Mapping"
format: pdf
editor: visual
---
```

### ### Load Libraries

```
`r`{
library(tidyverse)
library(here)
library(arrow)
library(sf)
library(urbanmapr)
library(naniar)
library(janitor)
library(ggiraph)
options(scipen = 99)
}
```

### ### Today's Data

The data we will analyze today is SAMHSA's TEDS-D Dataset. The metadata can be found [here](<https://www.samhsa.gov/data/system/files/media-puf-file/TEDS-D-2021-DS0001-info-codebook.pdf>)

Reading in feather files with `arrow`

```
`r`{
teds_d <- read_parquet(here("data/teds_d_lecture.parquet"))
}
```

### ### Clean names

```
`r`{
teds_d <- teds_d %>%
  clean_names()
}
```

### #### Selecting for relevant columns for today's class

- State
- Frequency of use at discharge
- Treatment Service
- Length of Stay
- Reason for Discharge

```
`r`{
teds_d_select <- teds_d %>%
  select(freq1_d, stfips, services_d, los, reason)
}
```

```
`r`{
write_parquet(teds_d_select, here("data/teds_d_lecture.parquet"))
}
```

```
`r`{
teds_d_select <- read_parquet(here("data/teds_d_lecture.parquet"))
}
```

#### #### NA Analysis

How does the documentation label missing data?

```
` `{r}
teds_d_select[teds_d_select == "-9"] <- NA
` `
```

```
` `{r}
miss_var_summary(teds_d_select)
` `
```

#### #### Variable Re-coding

##### ##### Frequency of Use at Discharge

```
` `{r}
teds_d_select$freq1_d <- as.character(teds_d_select$freq1_d)

teds_d_select$freq1_d[teds_d_select$freq1_d == "1"] <- "no use"
teds_d_select$freq1_d[teds_d_select$freq1_d == "2"] <- "some use"
teds_d_select$freq1_d[teds_d_select$freq1_d == "3"] <- "daily use"
teds_d_select$freq1_d[is.na(teds_d_select$freq1_d)] <- "unknown"
` `
```

##### ##### Services

```
` `{r}
teds_d_select$services_d <- as.character(teds_d_select$services_d)

teds_d_select$services_d[teds_d_select$services_d == "1"] <- "Detox, 24-hour, hospital
inpatient"

teds_d_select$services_d[teds_d_select$services_d == "2"] <- "Detox, 24-hour, free-
standing residential"

teds_d_select$services_d[teds_d_select$services_d == "3"] <- "Rehab/residential, hospital
(non-detox)"

teds_d_select$services_d[teds_d_select$services_d == "4"] <- "Rehab/residential, short
term (30 days or fewer)"

teds_d_select$services_d[teds_d_select$services_d == "5"] <- "Rehab/residential, long term
(more than 30 days)"

teds_d_select$services_d[teds_d_select$services_d == "6"] <- "Ambulatory, intensive
outpatient"

teds_d_select$services_d[teds_d_select$services_d == "7"] <- "Ambulatory, non-intensive
outpatient"

teds_d_select$services_d[teds_d_select$services_d == "8"] <- "Ambulatory, detoxification"
teds_d_select$services_d[is.na(teds_d_select$services_d)] <- "unknown"
` `
```

##### ##### Reason

```
` `{r}
teds_d_select$reason <- as.character(teds_d_select$reason)
```

```

teds_d_select$reason[teds_d_select$reason == "1"] <- "completed"
teds_d_select$reason[teds_d_select$reason == "2"] <- "dropped out"
teds_d_select$reason[teds_d_select$reason == "3"] <- "terminated by facility"
teds_d_select$reason[teds_d_select$reason == "4"] <- "transferred"
teds_d_select$reason[teds_d_select$reason == "5"] <- "incarcerated"
teds_d_select$reason[teds_d_select$reason == "6"] <- "death"
teds_d_select$reason[teds_d_select$reason == "7"] <- "other"
...

```

### ### Mapping

We want to map the percentage of complete treatments by state

First, let's calculate the percentage of completed treatments by state

```

```{r}
percent_completed_by_state <- teds_d_select %>%
  group_by(stfips) %>%
  summarize(
    total_cases = n(),
    completed_cases = sum(reason == "completed", na.rm = TRUE)
  ) %>%
  mutate(percentage_completed = (completed_cases / total_cases) * 100)
...

```

Next, let's bring in some mapping data

```

```{r}
states_map <- get_urban_map(map = "states", sf = TRUE)
```

```

What do we notice that's different between the teds-d stfips column and the states\_map stfips column?

```

```{r}
percent_completed_by_state$stfips_recode <- sprintf('%02d',
percent_completed_by_state$stfips)
```

```{r}
colnames(percent_completed_by_state)[colnames(percent_completed_by_state) ==
"stfips_recode"] <- "state_fips"
```

```

Joining data

```

```{r}
percent_completed_by_state_map <- full_join(percent_completed_by_state,
states_map,
by = "state_fips")
...

```

Plotting Map

```

```{r}
ggplot(percent_completed_by_state_map) +
  geom_sf(percent_completed_by_state_map,

```

```

      mapping = aes(geometry = geometry, fill = percentage_completed),
      color = "#ffffff", size = 0.25) +
labs(fill = "% of Completed Treatment Episodes") +
  coord_sf(datum = NA)+
  theme_minimal()
...

```

Making interactive with `ggiprah`

```

```{r}

interactive_completed_treatment_map <- ggplot(percent_completed_by_state_map) +
  geom_sf_interactive(
    mapping = aes(
      geometry = geometry,
      fill = percentage_completed,
      tooltip = paste("State FIPS:", stfips, "State Name:", state_name, "<br>Completed:",
percentage_completed, "%")
    ),
    color = "#ffffff",
    size = 0.25
  ) +
  labs(fill = "% of Completed Treatment Episodes") +
  coord_sf(datum = NA) +
  theme_minimal()

# Use `girafe` to render the interactive plot
girafe(ggobj = interactive_completed_treatment_map)
...

```

Round & Add state name to tooltip

Adding color bins

```

```{r}
percent_completed_by_state_map <- percent_completed_by_state_map %>%
  mutate(percentage_bin = cut(percentage_completed, breaks=c(0, 10,20,30,40,50, 60, 70,
80)))

ggplot(percent_completed_by_state_map) +
  geom_sf(mapping = aes(geometry = geometry, fill = percentage_bin),
    color = "#ffffff", size = 0.25) +
  labs(fill = "% of CompletedTreatment Episodes",
    title = "Completed Treatment Episodes by State",
    subtitle = "TEDS-D Dataset (SAMHSA)") +
  scale_fill_viridis_d(option = "D") +
  coord_sf(datum = NA) +
  theme_minimal() +
  theme(
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    legend.text = element_text(size = 4),
    legend.title = element_text(size = 5),
    strip.text = element_text(size = 4)
  )
...

```

### Assignment

1. Make an interactive map with `ggiraph` showing the percentage of completed treatments that end with no use at discharge (freq1\_d)

```

--\> no use at discharge / completed treatments

```{r}
percent_completed_by_states_map <- full_join(percent_completed_by_state,
                                             states_map,
                                             by = "state_fips")

ggplot(percent_completed_by_state_map) +
  geom_sf(percent_completed_by_state_map,
          mapping = aes(geometry = geometry, fill = percentage_completed),
          color = "#ffffff", size = 0.25) +
  labs(fill = "% of Completed Treatment Episodes") +
  coord_sf(datum = NA) +
  theme_minimal()
...

```{r}
percent_no_use_discharge <- teds_d_select %>%
  group_by(stfips) %>%
  summarize(
    total_cases = n(),
    completed_cases = sum(reason == "completed", na.rm = TRUE),
    completed_no_use_discharge = sum(freq1_d == "no use" & reason == "completed", na.rm =
TRUE)
  ) %>%
  mutate(percentage_no_use = (completed_no_use_discharge / completed_cases) * 100)
...

```{r}
percent_no_use_discharge$stfips_recode <- sprintf('%02d', percent_no_use_discharge$stfips)
...

```{r}
colnames(percent_no_use_discharge)[colnames(percent_no_use_discharge) == "stfips_recode"]
<- "state_fips"
...

```{r}
percent_no_use_discharge_map <- full_join(percent_no_use_discharge,
                                           states_map,
                                           by = "state_fips")
...

```{r}
interactive_no_use_discharge_map <- ggplot(percent_no_use_discharge_map) +
  geom_sf_interactive(
    mapping = aes(
      geometry = geometry,
      fill = percentage_no_use,
      tooltip = paste("State FIPS:", stfips, "<br>Completed with no use:",
round(percent_no_use, 2), "%", "<br>State:", state_name)
    ),
    color = "#ffffff",
    size = 0.1
  ) +
  labs(fill = "% of Completed Treatment Resulting in No Use",
       title = "Completed Treatments that end with No Use at Discharge") +
  coord_sf(datum = NA) +
  theme_minimal() +
  theme(
    panel.background = element_blank(),
    legend.text = element_text(size = 4),
    legend.title = element_text(size = 5)
  )

```

```

interactive_no_use_discharge_map
```

```{r}
interactive_no_use_discharge_map <- ggplot(percent_completed_by_state_map) +
  geom_sf_interactive(
    mapping = aes(
      geometry = geometry,
      fill = percentage_completed,
      tooltip = paste("State FIPS:", stfips, "State Name:", state_name, "<br>Completed:",
percentage_completed, "%")
    ),
    color = "#ffffff",
    size = 0.25
  ) +
  labs(fill = "% of Completed Treatment Resulting in No Use",
    title = "Completed Treatments that end with No Use at Discharge") +
  coord_sf(datum = NA) +
  theme_minimal()

# Use `girafe` to render the interactive plot
girafe(ggobj = interactive_no_use_discharge_map)
```

```

1. How does the percentage of treatments being [completed] & percentage of treatments ending with no use [vary by the service] (completed\_cases) and [\*length of stay\*]. (services\_d) Create at least 3 visualizations to try and answer this question

1. group\_by service or LOS, etc.

```

```{r}
summary(as.factor(teds_d_select$services_d))
```

```{r}
summary(as.factor(teds_d_select$los))
```

```{r}
percent_completed_w_services <- teds_d_select %>%
  group_by(stfips, services_d) %>%
  summarize(
    total_cases = n(),
    completed_cases = sum(reason == "completed", na.rm = TRUE)
  ) %>%
  mutate(percent_complete_by_service = (completed_cases / total_cases) * 100)

nlevels(as.factor(percent_completed_w_services$stfips)) # checking to see that all stfips
are represented

percent_completed_w_services$state_fips <- sprintf('%02d',
percent_completed_w_services$stfips)

percent_completed_w_services_map <- full_join(percent_completed_w_services,
  states_map,
  by = "state_fips")

ggplot(percent_completed_w_services_map, aes(x = state_abbv, y = completed_cases, fill =
services_d)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```{r}

```

```

ggplot(percent_completed_w_services_map, aes(x = state_abbv, y = completed_cases)) +
  geom_point(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
...

```{r}
ggplot(percent_completed_w_services_map, aes(x = state_abbv, y = completed_cases)) +
  geom_histogram(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
...

```{r}
percent_completed_w_services <- teds_d_select %>%
  group_by(services_d) %>%
  summarize(
    total_cases = n(),
    completed_cases = sum(reason == "completed", na.rm = TRUE)
  ) %>%
  mutate(percent_complete_by_service = (completed_cases / total_cases) * 100)

ggplot(percent_completed_w_services, aes(x = percent_complete_by_service, y = services_d,
fill = services_d)) +
  geom_bar(stat = "identity")
...

```