CSAIS Community Types

Jeremy Price

10/4/2021

```r
tags$style("@import url(https://use.fontawesome.com/releases/v5.8.2/css/all.css);")
```

```r
usa.map <<- governors %>%
  mutate(tile_map = generate_map(geometry, square = FALSE, flat_topped = TRUE))

communities.list <- c("Big Cities",
                      "Centers of Socioeconomic Inequality",
                      "Diverse Small Cities",
                      "Diverse Suburban Jewish Centers",
                      "Mid-Sized City Republicans",
                      "Small Town Democrats",
                      "Suburban Jewish Centers",
                      "Wealthy Enclaves",
                      "White Republicans",
                      "White Suburbs")

colors.diverge <<- c("#006298", "#FAF7F2", "#056E41")
colors.iu.light <- c("#FF636A", "#FFA690", "#FFE694", "#A7D094", "#94D2E7", "#C09EBB", "#FFD6DB", "#C6E
colors.iu <- c("#990000", "#DF3603", "#FFAA00", "#056E41", "#006298", "#59264D", "#5A0C0C",
               "#00385F", "#004421", "#330D2B")

overall.frame <- read_csv("cluster_designations-10.csv")
```

```
## Rows: 583 Columns: 15

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): cityID, community_type, color
## dbl (12): diversity_index, pop_category, median_income, political_scale, sta...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
sd.frame <- read_csv("sundown-cluster.csv")
```

```
## Rows: 28 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): cityID
## dbl (12): kmed_13, diversity_index, pop_category, median_income, political_s...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
just.cities <- overall.frame %>% select("cityID", "community_type") %>% na.omit

coded.frame <- read_csv("coded-incidents-full.csv")
```

```
## Rows: 1380 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): city, state, Type, Nature, Perpetrator, Target, Location, cityID
## dbl  (1): adl_id
## date (1): date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
coded.frame <- coded.frame %>% full_join(just.cities) %>% na.omit()
```

```
## Joining, by = "cityID"
```

```
overall.frame$jewish_infrastructure <- rescale(overall.frame$jewish_infrastructure,
                                               to = c(-1, 1))
overall.frame$community_interaction <- rescale(overall.frame$community_interaction, to = c(0, 1))
overall.frame$diversity_index <- rescale(overall.frame$diversity_index, to = c(0, 1))
overall.frame$pop_category <- rescale(overall.frame$pop_category, to = c(0, 1))
overall.frame$political_scale <- rescale(overall.frame$political_scale, to = c(0, 1))
overall.frame$median_income <- rescale(overall.frame$median_income, to = c(0,1))

median.diversity <<- median(overall.frame$diversity_index)
income.median <<- median(overall.frame$median_income)
population.median <<- median(overall.frame$pop_category)
political.median <<- median(overall.frame$political_scale)
jewish.median <<- median(overall.frame$jewish_infrastructure)
community.median <<- median(overall.frame$community_interaction)

overall.frame <<- overall.frame %>% group_by(community_type) %>%
  mutate(diversity_median = median(diversity_index),
         income_median = median(median_income),
         population_median = median(pop_category),
         political_median = median(political_scale),
         jewish_median = median(jewish_infrastructure),
         community_median = median(community_interaction)) %>%
  na.omit() %>% ungroup()

sundown.frame <- overall.frame %>% filter(cityID %in% sd.frame$cityID) %>%
  select(cityID, community_type, community_interaction, diversity_index,
         median_income, jewish_infrastructure, political_median,
         population_median)

just.cities <- just.cities[!(just.cities$cityID == "ia-bettendorf" | just.cities$cityID == "ia-des moin
variables.key <- c(community_median = "Community Interaction",
                   diversity_median = "Diversity Level",
                   income_median = "Median Income",
                   jewish_median = "Jewish Infrastructure",
                   political_median = "Politics",
                   population_median = "Population")
months.key <- c(Jan = "01", Feb = "02", Mar = "03", Apr = "04", May = "05", Jun = "06",
                Jul = "07", Aug = "08", Sep = "09", Oct = "10", Nov = "11", Dec = "12")
months.list <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
                 "Sep", "Oct", "Nov", "Dec")
seasons.list <- c("Winter", "Spring", "Summer", "Fall")
cities.frame <- read_csv("list-incidents-cities.csv", col_names = TRUE)
```

```
## Rows: 1001 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): city, county, state, fips_county
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
cities.frame$cityID <- paste(cities.frame$state, cities.frame$city, sep = "-")
cities.frame$cityID <- tolower(cities.frame$cityID)
months.frame <- cities.frame %>% right_join(overall.frame) %>%
  select(cityID, community_type, date)
```

```
## Joining, by = "cityID"
```

```
#months.frame$date <- months.frame$date %>% as.yearmon("2016-01-01") #as.yearmon(seq(ISOdate(2016,1,1),
months.frame <- months.frame %>% separate(date, into = c("year", "month"))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 994 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
#months.frame$month <- recode(months.frame$month, !!!months.key)
#months.frame$date <- paste(months.frame$year, months.frame$month, sep = "-")
months.frame$date <- paste("01", months.frame$month, months.frame$year, sep = "-")
months.frame <- months.frame %>% group_by(community_type) %>% count(date) %>% ungroup()
months.overall <- months.frame %>% select(date, n)
#months.overall <- months.overall %>% separate(date, into = c("month", "year"))
avg.frame <- months.frame %>% group_by(date) %>% summarize(n = mean(n))
#avg.frame <- avg.frame %>% rename("avg" = "n")
avg.frame$community_type <- "avg"
total.community.count <- length(unique(overall.frame$cityID))
average.incidence.rate <- length(coded.frame$adl_id) / total.community.count / 47

#months.frame$date <- as.Date(months.frame$date, "%d-%m-%Y")
#months.frame <- months.frame %>%
#  pivot_wider(names_from = community_type, values_from = n)
#months.frame[is.na(months.frame)] <- 0
months.overall <- as.data.frame(months.overall) %>% group_by(date) %>% tally(n) %>% ungroup()
months.overall$date <- as.Date(months.overall$date, "%d-%m-%Y")
months.xts <- as.xts(months.overall, order.by = months.overall$date)
#ct.xts <- as.xts(months.frame, order.by = months.frame$date)

usa_states <- map_data("state")
usa <- ggplot(usa_states, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group), color = "#A7A9AB",
               fill = "#EDEBEB", size = 0.2) +
```

```r
  coord_quickmap()

nature.frame <- read_csv("nature-count-type.csv", col_names = TRUE) %>%
  pivot_longer(cols = !community_type)
```

```
## Rows: 10 Columns: 8

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): community_type
## dbl (7): general_antisemitism, nazi, oppression_connection, israel, money, c...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
type.wide.frame <- read_csv("type-count-type.csv", col_names = TRUE)
```

```
## Rows: 10 Columns: 8

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): community_type
## dbl (7): type_bomb_threat, type_vandalism, type_harassment, type_literature_...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
type.frame <- type.wide.frame %>% pivot_longer(cols = !community_type)
type.levels <- c("Vandalism", "Harassment", "Cyberbullying", "Performance", "Literature Dump", "Bomb Thi
nature.levels <- c("Nazi", "General Antisemitism", "Oppression Connection", "Money", "Holocaust Denial"
nature.key <- c(nazi = "Nazi", general_antisemitism = "General Antisemitism",
                oppression_connection = "Oppression Connection", israel = "Israel",
                money = "Money", conspiracy = "Conspiracy",
                holocaust_denial = "Holocaust Denial")
type.key <- c(type_vandalism = "Vandalism", type_harassment = "Harassment",
              type_cyberbullying = "Cyberbullying", type_performance = "Performance",
              type_literature_dump = "Literature Dump", type_bomb_threat = "Bomb Threat",
              type_not_specified = "Not Specified")
nature.frame$name <- recode(nature.frame$name, !!!nature.key)
nature.frame <- nature.frame %>% mutate(name = factor(name, levels = nature.levels))
type.frame$name <- recode(type.frame$name, !!!type.key)
type.frame <- type.frame %>% mutate(name = factor(name, levels = type.levels))

nt.frame <<- coded.frame %>% select(community_type, Nature, Type) %>%
  group_by(community_type, Nature, Type) %>% add_tally() %>% ungroup() %>% distinct()
nt.frame <- nt.frame %>% mutate(Nature = factor(Nature, levels = nature.levels))
nt.frame <- nt.frame %>% mutate(Type = factor(Type, levels = type.levels))

cluster.frame <- overall.frame %>% select(cityID, diversity_index, pop_category,
                                          median_income, political_scale,
                                          jewish_infrastructure, community_interaction) %>%
  column_to_rownames(var = "cityID")
cluster.frame$diversity_index <- scale(cluster.frame$diversity_index)
cluster.frame$pop_category <- scale(cluster.frame$pop_category)
cluster.frame$median_income <- scale(cluster.frame$median_income)
cluster.frame$political_scale <- scale(cluster.frame$political_scale)
cluster.frame$jewish_infrastructure <- scale(cluster.frame$jewish_infrastructure)
cluster.frame$community_interaction <- scale(cluster.frame$community_interaction)
mds <- cluster.frame %>%
  dist() %>%
  cmdscale() %>%
  as_tibble()
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
mds <- mds %>%
  mutate(groups = as.factor(overall.frame$community_type))
colnames(mds) <- c("dim.1", "dim.2", "community_type")
mds$cityID <- just.cities$cityID
#mds$dim.1 <- -mds$dim.1
#mds$dim.2 <- -mds$dim.2
# Plot and color by groups
#scatter <- ggscatter(mds, x = "dim.1", y = "dim.2",
#                     label = overall.frame$cityID,
#                     color = "Community Type",
#                     palette = colors.iu,
#                     size = 0.75,
#                     ellipse = TRUE,
#                     ellipse.type = "convex",
#                     repel = TRUE,
#                     font.family = "Open Sans") +
#  bgcolor("#F7F7F8") +
#  guides(color = guide_legend(nrow = 3)) +
#  scale_x_continuous(name = "Urbanization", breaks = c(-2, 2),
#                     labels = c("Less", "More")) +
#  scale_y_continuous(name = "Jewish Centrality", breaks = c(-6, 2),
#                     labels = c("Less", "More")) +
#  theme(axis.text = element_text(color = "#243142"),
#        axis.ticks = element_line(color = "#243142"),
#        axis.line = element_line(color = "#243142"),
        #plot.background = element_rect(fill = "#FAF7F2"),
#        legend.background = element_rect(fill = NA),
#        legend.position = "bottom", legend.title = element_blank())

measures.plot <- function(var.community) {
  graph.frame <- overall.frame %>% filter(community_type == var.community) %>%
    select(community_type, income_median, diversity_median, population_median, political_median,
           jewish_median, community_median)
  graph.frame$diversity_median <- graph.frame$diversity_median - median.diversity
  graph.frame$jewish_median <- graph.frame$jewish_median - jewish.median
  graph.frame$population_median <- graph.frame$population_median - population.median
  graph.frame$income_median <- graph.frame$income_median - income.median
  graph.frame$political_median <- graph.frame$political_median - political.median
  graph.frame$community_median <- graph.frame$community_median - community.median
  graph.frame <- graph.frame %>% pivot_longer(cols = !community_type, names_to = "measure") %>%
    distinct()
  graph.frame$value_group <- factor(case_when(graph.frame$value < -0.1 ~ "low",
                                              graph.frame$value > 0.1 ~ "high",
                                              TRUE ~ "middle"),
                                    levels = c("low", "middle", "high"))
  variable.key <- c("community_median" = "Community Type",
                    "diversity_median" = "Diversity Level",
                    "income_median" = "Income Level",
                    "jewish_median" = "Jewish Infrastructure",
                    "political_median" = "Political Scale",
                    "population_median" = "Population")
  graph.frame$measure <- recode(graph.frame$measure, !!!variable.key)
  the.plot <- ggbarplot(graph.frame, x = "measure", y = "value",
                        color = "#b5c8b8", fill = "value_group",
                        palette = colors.diverge, #c("#006298", "#ffffff", "#990000"),
                        order = c("Population", "Political Scale",
                                  "Jewish Infrastructure", "Income Level",
                                  "Diversity Level", "Community Type")) +
    #title = plot.title,
    #subtitle = sub.title,
    #caption = cap.title) +
    scale_y_continuous(limits = c(-0.85, 0.85)) +
    geom_hline(yintercept = 0, linetype = 2, color = "#79a7ac") +
    theme(axis.text.x = element_blank(),
          axis.text.y = element_text(color = "#243142", family = "Monoid"),
          axis.title = element_blank(),
          axis.ticks = element_blank(),
          axis.line = element_line(color = "#b5c8b8"),
          legend.position = "none",
          panel.grid.major = element_line(color = "#edeac2", linetype = 8)) +
    rotate()
  return(the.plot)
}

community.counts <- c(length(which(overall.frame$community_type == "Big Cities")),
                      length(which(overall.frame$community_type == "Centers of Socioeconomic Inequality
                      length(which(overall.frame$community_type == "Diverse Small Cities")),
                      length(which(overall.frame$community_type == "Diverse Suburban Jewish Centers")),
                      length(which(overall.frame$community_type == "Mid-Sized City Republicans")),
                      length(which(overall.frame$community_type == "Small Town Democrats")),
                      length(which(overall.frame$community_type == "Suburban Jewish Centers")),
                      length(which(overall.frame$community_type == "Wealthy Enclaves")),
                      length(which(overall.frame$community_type == "White Republicans")),
                      length(which(overall.frame$community_type == "White Suburbs")))

community.icons <<- data.frame(community_type = communities.list,
                              icons = c("fas fa-city",
                                        "fa-not-equal",
                                        "fas fa-palette",
                                        "fas fa-hamsa",
                                        "fas fa-industry",
                                        "fas fa-mountain",
                                        "fas fa-torah",
                                        "fas fa-funnel-dollar",
                                        "fas fa-compress-arrows-alt",
                                        "fa-archway"),
                              counts = community.counts,
                              colors = colors.iu,
                              light_colors = colors.iu.light)

map.frame <- overall.frame %>%
  select(community_type, cityID, lat, lon) %>%
  rename("longitude" = "lon", "latitude" = "lat") %>%
  separate(cityID, into = c("state","city"), sep = "-", remove = FALSE)
map.frame$city <- toTitleCase(map.frame$city)
map.frame$state <- toupper(map.frame$state)
map.frame$label <- paste(map.frame$city, map.frame$state, sep = ", ")

icon.frame <- data.frame(community_type = c("Big Cities", "Centers of Socioeconomic Inequality",
                                            "Diverse Small Cities", "Diverse Suburban Jewish Centers",
                                            "Mid-Sized City Republicans", "Small Town Democrats",
                                            "Suburban Jewish Centers", "Wealthy Enclaves",
                                            "White Republicans", "White Suburbs"),
                         icon = c("fa-city", "fa-not-equal", "fa-palette", "fa-hamsa",
                                  "fa-industry", "fa-mountain", "fa-torah", "fa-funnel-dollar",
                                  "fa-compress-arrows-alt", "fa-archway"),
                         extraClasses = c("fas", "fa", "fas", "fas", "fas", "fas",
                                          "fas", "fas", "fas", "fas"),
                         markerColor = c("#990000", "#DF3603", "#FFAA00", "#056E41",
                                         "#006298", "#59264D", "#800000", "#004F80",
                                         "#005C31", "#48183D"))

#map.frame <- map.frame %>% right_join(icon.frame)
IconSet <- awesomeIconList(
  "Big Cities" = makeAwesomeIcon(icon= "city", extraClasses = "fas", library = "fa",
                                 markerColor = "red", iconColor = "black"),
  "Centers of Socioeconomic Inequality" = makeAwesomeIcon(markerColor = "orange", icon= "not-equal",
                                                          library = "fa", iconColor = "black"),
  "Diverse Small Cities" = makeAwesomeIcon(icon= "palette", extraClasses = "fas", library = "fa", marke
  "Diverse Suburban Jewish Centers" = makeAwesomeIcon(icon= "hamsa", extraClasses = "fas", library = "f
  "Mid-Sized City Republicans" = makeAwesomeIcon(icon= "industry", extraClasses = "fas", library = "fa"
  "Small Town Democrats" = makeAwesomeIcon(icon= "fa-mountain", extraClasses = "fas", library = "fa", ma
  "Suburban Jewish Centers" = makeAwesomeIcon(icon= "fa-torah", extraClasses = "fas", library = "fa", ma
  "Wealthy Enclaves" = makeAwesomeIcon(icon= "fa-funnel-dollar", extraClasses = "fas", library = "fa", m
  "White Republicans" = makeAwesomeIcon(icon= "fa-compress-arrows-alt", extraClasses = "fas", library =
  "White Suburbs" = makeAwesomeIcon(icon= "fa-archway", extraClasses = "fas", library = "fa", markerCol
)

markerLegendHTML <- function(IconSet) {
    # container div:
    legendHtml <- "<div style='padding: 10px; padding-bottom: 10px;'><h4 style='padding-top:0; padding-

    n <- 1
    # add each icon for font-awesome icons icons:
    for (Icon in IconSet) {
        if (Icon[["library"]] == "fa") {
        legendHtml<- paste0(legendHtml, "<div style='width: auto; height: 36px'>",
                             "<div style='position: relative; display: inline-block; width: 36px; heigh
                               "<i style='margin-left: 6px; margin-top: 9px; 'class= 'fa fa-",Icon[["ic
                             "</div>",
                             "<p style='position: relative; top: 10px; display: inline-block; ' >", name
                           "</div>")
        }
        n <- n + 1
    }
    paste0(legendHtml, "</div>")
}

create.map <- function(df) {
  the.map <- df %>%
    leaflet() %>%
    addTiles() %>%  # Add default OpenStreetMap map tiles
    addAwesomeMarkers(lng = df$latitude, lat = df$longitude,
                      popup = df$label, icon = ~IconSet[community_type]) %>%
    addControl(html = markerLegendHTML(IconSet = IconSet), position = "bottomright")
  return(the.map)
}

overall.map <- create.map(map.frame)

create.title <- function(ct) {
  the.icon <- community.icons$icons[community.icons$community_type == ct]
  the.color <- community.icons$colors[community.icons$community_type == ct]
  the.count <- community.icons$counts[community.icons$community_type == ct]
  the.caption <- paste0("There are ", the.count, " communities identified as ", ct)
  valueBox(ct, icon = the.icon, caption = the.caption, ct)
}

create.cloud <- function(ct) {
    cloud.frame <- coded.frame %>% filter(community_type == ct) %>%
    select(Type, Nature)
  nature.list <- cloud.frame$Nature
  type.list <- cloud.frame$Type
  cloud.list <- c(nature.list, type.list)
  use.frame <- data.frame(code = cloud.list)
  use.frame <- use.frame %>% group_by(code) %>% tally() %>% drop_na()
  use.frame$code <- str_replace(use.frame$code, "General Antisemitism", "General")
  incidents.cloud <- ggplot(use.frame, aes(label = code, size = n, color = n)) +
    geom_text_wordcloud_area(family = "Public Sans") +
    scale_size_area(max_size = 24) +
    scale_color_steps(low = community.icons$light_colors[community.icons$community_type == ct], high = 
    theme(plot.background = element_rect(fill = "#F7F7F8"),
          panel.background = element_blank())
 return(incidents.cloud)
}

create.tilemap <- function(ct) {
    plot.frame <- coded.frame %>% filter(community_type == ct) %>%
      select(community_type, Type, Nature)
  incidents.count <- plot.frame %>% group_by(community_type, Type, Nature) %>% add_tally() %>% drop_na(
  type.levels <- c("Vandalism", "Harassment", "Cyberbullying", "Performance", "Literature Dump", "Bomb T
  nature.levels <- c("Nazi", "General Antisemitism", "Oppression Connection", "Money", "Holocaust Denial
  incidents.count <- incidents.count %>% mutate(Type = factor(Type, levels = type.levels)) %>%
    mutate(Nature = factor(Nature, levels = nature.levels))

  incidents.plot <- ggplot(incidents.count, aes(Type, Nature)) +
    geom_tile(aes(fill = n), color = "#243142") +
    geom_text(aes(label = n), size = 3)  +
    scale_fill_steps(low = "#EDEBEB", high = "#990000", name = "Number", na.value = "#F7F7F8") +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          title = element_blank(),
          axis.title = element_blank(),
          axis.text.x = element_text(angle = 90),
          legend.position = "bottom",
          text = element_text(family = "Open Sans"))
  return(incidents.plot)
}

create.calendar <- function(ct) {
  full.cal <- data.frame(community_type = ct,
                         year = c("2016", "2016", "2016", "2016", "2016",
                                  "2016", "2016", "2016", "2016", "2016",
                                  "2016", "2016", "2017", "2017", "2017",
                                  "2017", "2017", "2017", "2017", "2017",
                                  "2017", "2017", "2017", "2017", "2018",
                                  "2018", "2018", "2018", "2018", "2018",
                                  "2018", "2018", "2018", "2018", "2018",
                                  "2018", "2019", "2019", "2019", "2019",
                                  "2019", "2019", "2019", "2019", "2019",
                                  "2019", "2019", "2019"),
                         month = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec",
                                   "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec",
                                   "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec",
                                   "Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                   "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))
  the.time <- months.frame %>% filter(community_type == ct) %>%
    separate(date, into = c("year", "month", "day")) %>%
    select(-day)
  the.time$month <- case_when(the.time$month == "01" ~ "Jan",
                              the.time$month == "02" ~ "Feb",
                              the.time$month == "03" ~ "Mar",
                              the.time$month == "04" ~ "Apr",
                              the.time$month == "05" ~ "May",
                              the.time$month == "06" ~ "Jun",
                              the.time$month == "07" ~ "Jul",
                              the.time$month == "08" ~ "Aug",
                              the.time$month == "09" ~ "Sep",
                              the.time$month == "10" ~ "Oct",
                              the.time$month == "11" ~ "Nov",
                              the.time$month == "12" ~ "Dec")
  the.time <- the.time %>%
    full_join(full.cal)# %>%
  the.time$n <- the.time$n %>% replace_na(0)
  season.colors <- c("#96B3DF", "#65AF58", "#E1C6B3", "#750C1A")
  the.avg <- the.time %>%
    group_by(month) %>%
    summarize(avg = mean(n))
  the.avg <- the.avg %>%
    mutate(month = factor(month, levels = months.list))
  the.max <- max(the.avg$avg) + 1
  time.plot <- ggplot(the.avg, aes(x = month, y = avg)) +
    geom_bar(stat = "identity", position = "dodge", aes(fill = avg)) +
    scale_fill_steps(low = "#EDEBEB", high = community.icons$colors[community.icons$community_type == c
    coord_polar(start = 0) +
    ylim(-2, the.max) +
    geom_bar(stat = "identity", aes(x = month, y = -2), fill = "white") +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          title = element_blank(),
          axis.title = element_blank(),
          axis.ticks = element_blank(),
          axis.text.y = element_blank(),
          text = element_text(family = "Open Sans"),
          legend.position = "bottom",
          panel.background = element_rect(fill = "white"))
  #time.plot <- ggplot(the.time, aes(x = date, y = n)) +
  #  stat_occurrence(color = community.icons$colors[community.icons$community_type == input$community_t
  #  theme(panel.grid.major = element_blank(),
  #        panel.grid.minor = element_blank(),
  #        title = element_blank(),
  #        axis.title = element_blank(),
  #        text = element_text(family = "Monoid"),
  #        panel.background = element_rect(fill = "#F7F7F8"))
  return(time.plot)
}

create.hex <- function(ct) {
    m.frame <- overall.frame %>% filter(community_type == ct) %>%
    select(cityID, color, lat, lon) %>%
    separate(cityID, into = c("abbreviation","city"), sep = "-", remove = FALSE)
  m.frame$city <- toTitleCase(m.frame$city)
  m.frame$abbreviation <- toupper(m.frame$abbreviation)
  count.frame <- m.frame %>% count(abbreviation)
  the.map <- usa.map %>% full_join(count.frame)
  hex.map <- ggplot(the.map) +
    geom_sf(aes(geometry = tile_map, fill = n), color = "#A7A9AB") +
    geom_sf_text(aes(geometry = tile_map, label = abbreviation),
                 fun.geometry = function(x) st_centroid(x), family = "Public Sans") +
    #scale_fill_gradient(low = "#EDEBEB", high = "#990000", name = "Number", na.value = "#F7F7F8") +
    scale_fill_steps(low = "#EDEBEB", high = community.icons$colors[community.icons$community_type == c
    theme_void() +
    theme(legend.position = "bottom",
          text = element_text(family = "Open Sans"))
  return(hex.map)
  hex.map
}

create.tile <- function(ct) {
  nt.frame <- nt.frame %>% filter(community_type == ct)
  incidents.plot <- ggplot(nt.frame, aes(Type, Nature)) +
    geom_tile(aes(fill = n), color = "#243142") +
    geom_text(aes(label = n), size = 3) +
    #scale_fill_gradient(low = "#EDEBEB", high = "#990000", name = "Number", na.value = "#F7F7F8") +
    scale_fill_steps(low = "#EDEBEB", high = community.icons$colors[community.icons$community_type == c
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          title = element_blank(),
          axis.title = element_blank(),
          axis.text.x = element_text(angle = 45),
          legend.position = "bottom",
          text = element_text(family = "Open Sans"))
  incidents.plot <- ggplotly(incidents.plot)
  return(incidents.plot)
}

create.tile_agents <- function(ct) {
  plot.frame <- coded.frame %>% filter(community_type == ct) %>%
    select(community_type, Perpetrator, Target)
  incidents.count <- plot.frame %>% group_by(Perpetrator, Target) %>% add_tally() %>% drop_na() %>% dis
  target.levels <- c("Not Specified", "Student", "School", "Teacher", "Outsider", "Parent", "School Pers
  perpetrator.levels <- c("Not Specified", "Student", "Outsider", "Teacher", "Parent", "Crowd", "School
  incidents.plot <- ggplot(incidents.count, aes(Perpetrator, Target)) +
    geom_tile(aes(fill = n), color = "#243142") +
    geom_text(aes(label = n), size = 3) +
    #scale_fill_gradient(low = "#EDEBEB", high = "#990000", name = "Number", na.value = "#F7F7F8") +
    scale_fill_steps(low = "#EDEBEB", high = community.icons$colors[community.icons$community_type == c
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          title = element_blank(),
          axis.title = element_blank(),
          axis.text.x = element_text(angle = 45),
          legend.position = "bottom",
          text = element_text(family = "Open Sans"))
  incidents.plot <- ggplotly(incidents.plot)
  return(incidents.plot)
}

create.table <- function(ct) {
  table.frame <- overall.frame %>% filter(community_type == ct) %>%
    select(community_type, community_median, diversity_median, income_median,
           jewish_median, political_median, population_median) %>%
  distinct() %>% pivot_longer(!community_type) %>% select(-community_type)
  table.frame$adj <- NA
  table.frame[1, 3] <- table.frame[1, 2] - community.median
  table.frame[2, 3] <- table.frame[2, 2] - median.diversity
  table.frame[3, 3] <- table.frame[3, 2] - income.median
  table.frame[4, 3] <- table.frame[4, 2] - jewish.median
  table.frame[5, 3] <- table.frame[5, 2] - political.median
  table.frame[6, 3] <- table.frame[6, 2] - population.median
  table.frame$desc <- NA
  table.frame[1, 4] <- case_when(table.frame[1, 3] < -0.1 ~ '<i class="fa-building"></i> More Urban',
                                 table.frame[1, 3] > 0.1 ~  '<i class="fa-home"></i> More Rural',
                                 TRUE ~ '<i class="fas fa-ban"></i> N/A')
#                   '<i class="fab fa-hornbill"></i> High Diversity',
#                   '<i class="fas fa-money-bill-wave-alt"></i> Low Median Income',
#                   '<i class="fas fa-ban"></i> N/A',
#                   '<i class="fas fa-ban"></i> N/A',
#                   '<i class="fa fa-users"></i> High Population')
  table.frame <- table.frame %>% rename("Measure" = "name", "Median Score" = "value",
                                "Adjusted Score" = "adj", "Description" = "desc")
  table.table <- knitr::kable(table.frame)
  return(table.table)
}
```

Overview

Antisemitic incidents in schools are a widespread and pervasive phenomenon. The Anti-Defamation League has reported a three-fold increase in school-based antisemitic incidences between 2015 and 2018, and a nearly two-fold increase in violent and non-violent incidences of antisemitism in general nationwide.

This project utilizes data from the ADL H.E.A.T. Map between 2016 and 2019 to identify incidents of antisemitism that specifically took place in schools. These incidents in schools are influenced by demographic, historical, social, and political factors. This project brings this data together to construct a community typology at the national level. This typology will provide insight into the ways that school-based incidents of hate are enacted and reported in context.

Developing a community typology will allow providers to better target specific demographic, historical, and political attributes of the communities in which these incidents occur through curriculum and learning experiences.