# Traffic stops case study

*Andrew Ba Tran*

*2018-05-27T21:13:14-05:00*

## Contents

This is from the fifth chapter of learn.r-journalism.com.

We're going to take what we've learned so far and do some spatial analysis of traffic stops.

**Goal:** We'll figure out which town and census tract each stop occurred in and then pull in demographic data from the Census to determine what types of neighborhoods police tend to pull people over more often.

You could conduct this analysis using software like ArcGIS or QGIS, but we're going to be doing it all in R.

It is better to stay in a single environment from data importing, to analysis, to exporting visualizations because the produced scripts make it easier for others (including your future self) to replicate and verify your work in the future.

Start with the data. It's raw traffic stops between 2013 and 2014. It includes race, reasons for the stop, and many other factors. The state of Connecticut collects this information from all police departments but only a handful of them included location-specific information. Researchers at Central Connecticut State University's Center for Municipal and Regional Policy geolocated as many as possible, focusing on eight departments that showed signs of racial profiling.

About 34,000 stops were geolocated.

We're looking specifically at Hamden for this case study— so about 5,500 stops. We'll revisit the other towns later.

Let's load the packages and data we'll need.

```
# if you don't have any of these packages installed yet, uncomment and run the lines below
#install.packages("tidycensus", "ggplot2", "dplyr", "sf", "readr")

library(tidycensus)
library(ggplot2)
library(dplyr)
library(sf)
library(readr)

stops <- read_csv("data/hamden_stops.csv")
```

```
View(stops)
```

Let's get rid of some of the bad data– where there was no latitude or longitude data (Don't worry, that's only about two percent of the data).

```
stops <- filter(stops, InterventionLocationLatitude!=0)
```

Let's map what we've got. Let's download the Census tract shapefiles in Hamden's county– New Haven County– with **tigris**.

```
# If you don't have tigris installed yet, uncomment the line below and run
#install.packages("tigris")
```

Filter

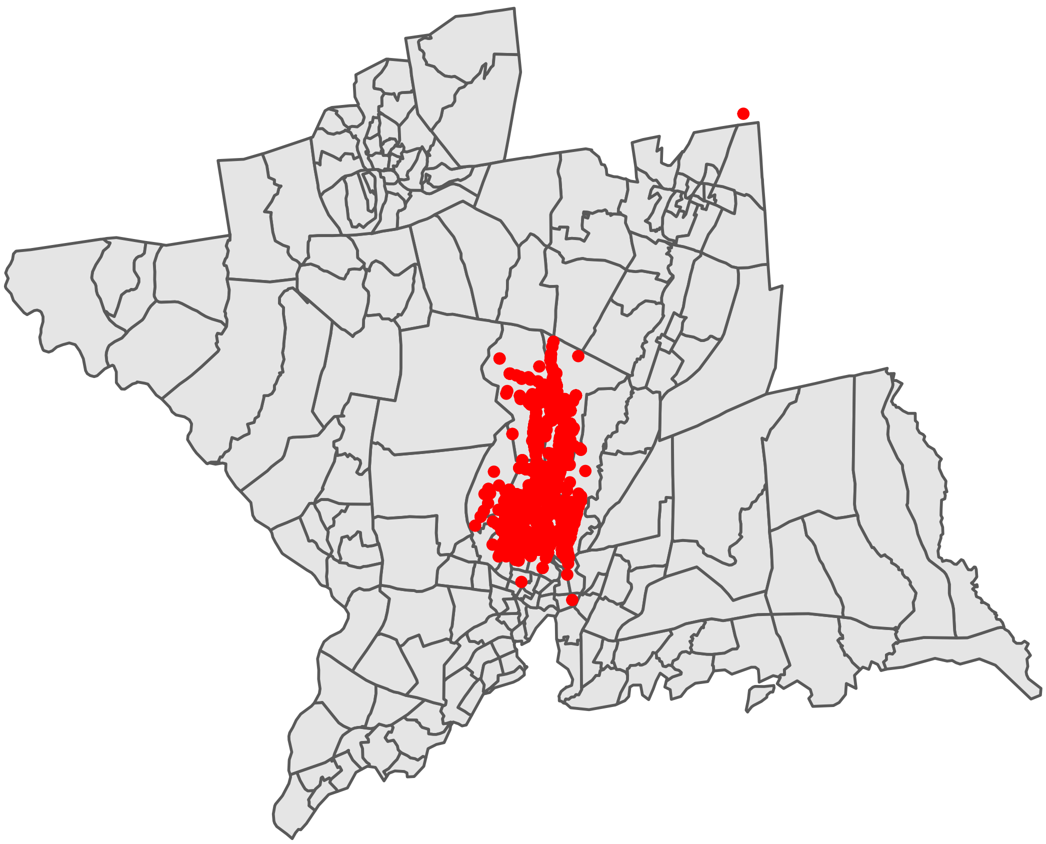
| InterventionDate | Month | SubjectRaceCode | SubjectEthnicityCode | SubjectSexCode | SubjectAge | ResidentIndicator | TownRecidentIndicator | Inte |
|---|---|---|---|---|---|---|---|---|
| Tuesday, October 01, 2013 | NA | W | N | F | 28 | TRUE | FALSE | H |
| Tuesday, October 01, 2013 | NA | B | N | M | 33 | TRUE | TRUE | H |
| Tuesday, October 01, 2013 | NA | B | N | M | 25 | TRUE | FALSE | H |
| Tuesday, October 01, 2013 | NA | W | N | F | 33 | TRUE | FALSE | N |
| Tuesday, October 01, 2013 | NA | W | N | M | 45 | TRUE | TRUE | H |
| Tuesday, October 01, 2013 | NA | W | H | F | 19 | TRUE | TRUE | H |
| Tuesday, October 01, 2013 | NA | W | N | M | 20 | FALSE | FALSE | H |
| Tuesday, October 01, 2013 | NA | B | N | M | 23 | TRUE | TRUE | H |
| Tuesday, October 01, 2013 | NA | B | N | M | 25 | TRUE | FALSE | H |
| Tuesday, October 01, 2013 | NA | W | N | F | 26 | TRUE | FALSE | H |
| Tuesday, October 01, 2013 | NA | B | N | M | 25 | TRUE | FALSE | H |
| Wednesday, October 02, 2013 | NA | W | N | F | 50 | TRUE | TRUE | H |
| Wednesday, October 02, 2013 | NA | W | N | M | 27 | TRUE | FALSE | H |
| Wednesday, October 02, 2013 | NA | B | N | M | 24 | TRUE | TRUE | H |
| Wednesday, October 02, 2013 | NA | W | H | M | 35 | TRUE | FALSE | H |
| Wednesday, October 02, 2013 | NA | W | N | F | 50 | TRUE | FALSE | H |
| Wednesday, October 02, 2013 | NA | W | H | M | 25 | TRUE | TRUE | H |

Figure 1:

```r
library(tigris)

# set sf option

options(tigris_class = "sf")

new_haven <- tracts(state="CT", county="New Haven", cb=T)

# If cb is set to TRUE, download a generalized (1:500k) counties file. Defaults to FALSE
# (the most detailed TIGER file).
```

```r
ggplot(new_haven) +
  geom_sf() +
  geom_point(data=stops,
             aes(x=InterventionLocationLongitude,
                 y=InterventionLocationLatitude), color="red") +
  #geom_point(data=stops_spatial, aes(x=lon, y=lat), color="blue") +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent')) +
  labs(title="Traffic stops in Hamden")
```

Traffic stops in Hamden



Alright, it's a start.

### Deeper analysis

If you have a data set with latitude and longitude information, it's easy to just throw it on a map with a dot for every instance.

But what would that tell you? You see the intensity of the cluster of dots over the area but that's it.

If there's no context or explanation it's a flashy visualization and that's it.

### Heat map

One way is to visualize the distribution of stops.

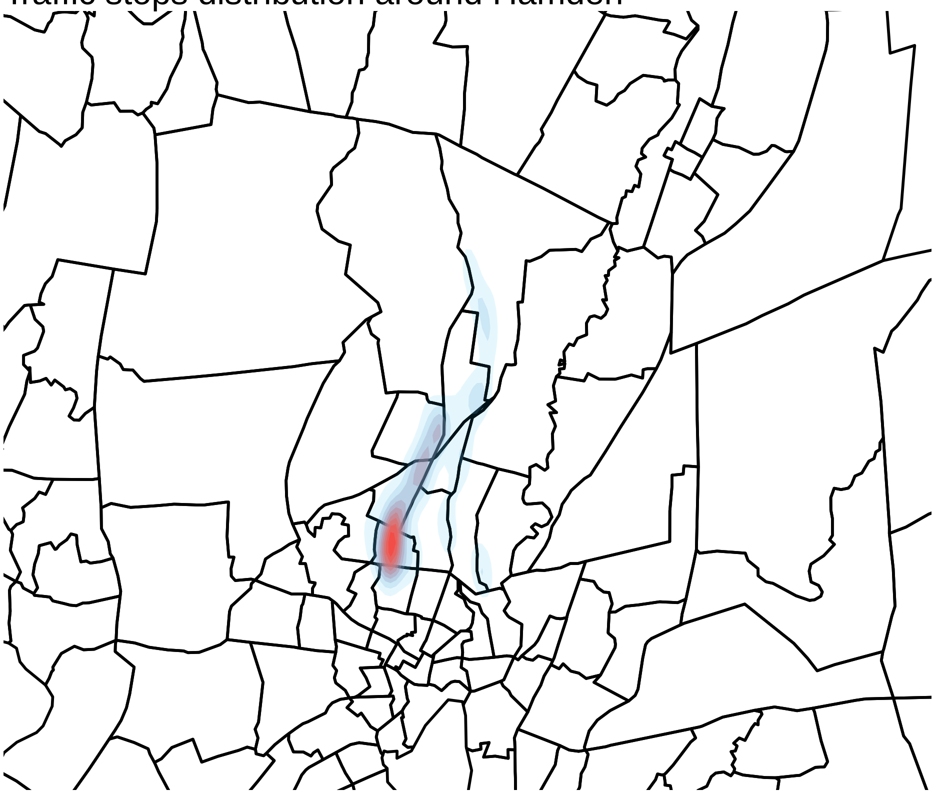
We'll use the `stat_density2d()` function within ggplot2 and use `coord_map()` and `xlim` and `ylim` to set the boundaries on the map so it's zoomed in more.

**TIP:** When you're stacking ggplots or dplyr commands, the line can't end with a + or a %>% normally, right? Well, if you stick a NULL at the last line, you can have the + or %>% precede it. This is kind of an advanced tip for folks who play around with scripts and are tired of commenting out before the + or %>%.

```
ggplot() +
  geom_sf(data=new_haven, color = "black", fill=NA, size=0.5) +
  stat_density2d(data=stops, show.legend=F,
          aes(x=InterventionLocationLongitude, y=InterventionLocationLatitude,
```

```
        fill=..level.., alpha=..level..), geom="polygon", size=4, bins=10) +
  scale_fill_gradient(low="deepskyblue2", high="firebrick1", name="Distribution") +
  coord_sf(xlim=c(-73.067649, -72.743739), ylim=c(41.280972, 41.485011)) +
  labs(x=NULL, y=NULL,
                title="Traffic stops distribution around Hamden",
                subtitle=NULL,
                caption="Source: data.ct.gov") +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent'))
```

Traffic stops distribution around Hamden



Source: data.ct.gov

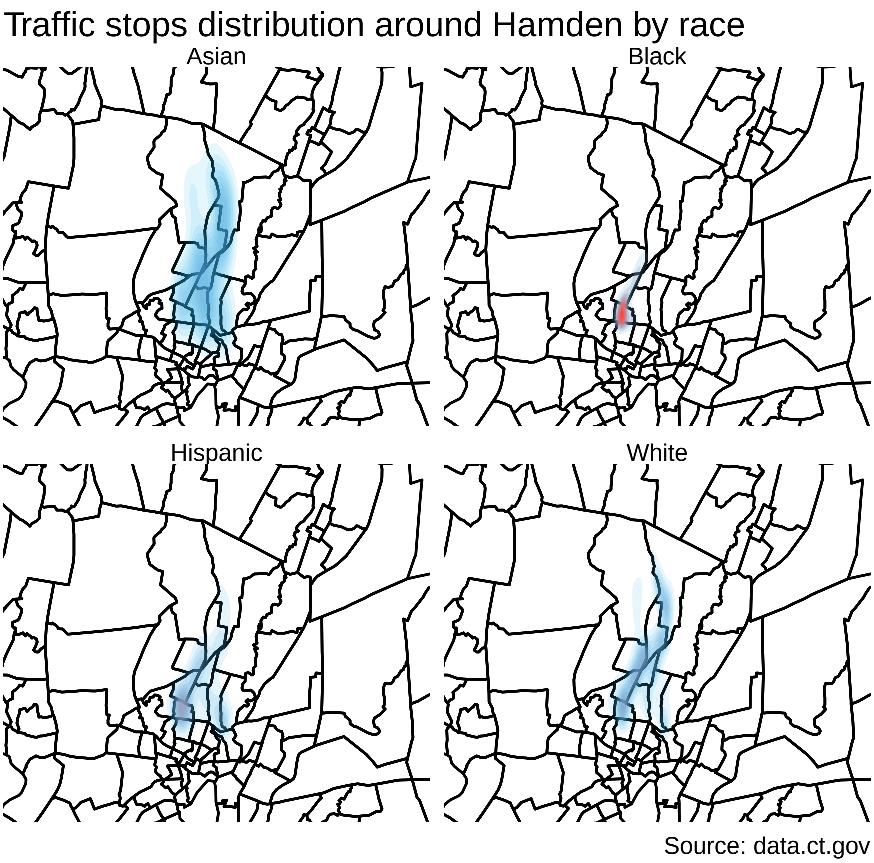
That's interesting.

What's nice about ggplot2, is the functionality called `facets`, which allows the construction of small multiples based on factors.

Let's try this again but faceted by race.

```
# Creating a race column
stops$race <- ifelse(stops$SubjectEthnicityCode=="H", "H", stops$SubjectRaceCode)
stops <- stops %>%
  mutate(race=case_when(
    race=="A" ~ "Asian",
    race=="B" ~ "Black",
    race=="H" ~ "Hispanic",
    race=="W" ~ "White"
  ))
```

```
ggplot() +
  geom_sf(data=new_haven, color = "black", fill=NA, size=0.5) +
  stat_density2d(data=stops, show.legend=F,
         aes(x=InterventionLocationLongitude, y=InterventionLocationLatitude,
         fill=..level.., alpha=..level..), geom="polygon", size=4, bins=10) +
  scale_fill_gradient(low="deepskyblue2", high="firebrick1", name="Distribution") +
  coord_sf(xlim=c(-73.067649, -72.743739), ylim=c(41.280972, 41.485011)) +
  #This is the new line being added
  facet_wrap(~race) +
  labs(x=NULL, y=NULL,
                  title="Traffic stops distribution around Hamden by race",
                  subtitle=NULL,
                  caption="Source: data.ct.gov") +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent'))
```




Interesting.

But it still doesn't tell the full story because it's still a bit misleading.

Here's what I mean.

```
stops %>%
  group_by(race) %>%
  count()
```

```
## # A tibble: 4 x 2
## # Groups:   race [4]
```

Filter

| | GEOID | NAME | summary_value | X1 | X | ProfileNo | OrganizationIdentificationID | DepartmentName | O |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 09009154600 | Census Tract 1546 | 4029 | NA | NA | NA | NA | NA | N |
| 2 | 09009154700 | Census Tract 1547 | 6384 | NA | NA | NA | NA | NA | N |
| 3 | 09009157200 | Census Tract 1572 | 3837 | NA | NA | NA | NA | NA | N |
| 4 | 09009165100 | Census Tract 1651 | 3928 | 15046 | 15046 | 1918577 | CT0006200 | Hamden | Ra |
| 4.1 | 09009165100 | Census Tract 1651 | 3928 | 15071 | 15071 | 1918630 | CT0006200 | Hamden | Ra |
| 4.2 | 09009165100 | Census Tract 1651 | 3928 | 15091 | 15091 | 1918493 | CT0006200 | Hamden | Ra |
| 4.3 | 09009165100 | Census Tract 1651 | 3928 | 15125 | 15125 | 1918413 | CT0006200 | Hamden | Ra |
| 4.4 | 09009165100 | Census Tract 1651 | 3928 | 15127 | 15127 | 1918415 | CT0006200 | Hamden | Ra |
| 4.5 | 09009165100 | Census Tract 1651 | 3928 | 15130 | 15130 | 1918414 | CT0006200 | Hamden | Ra |

Figure 2:

```
##   race         n
##   <chr>    <int>
## 1 Asian       71
## 2 Black     2035
## 3 Hispanic   444
## 4 White     2808
```

The distribution is comparitive to its own group and not as a whole.

Gotta go deeper.

Let's look at which neighborhoods police tend to pull people over more often and compare it to demographic data from the Census.

So we need to count up the instances with the `st_join()` function from the **sf** package.

### Points in a Polygon

We already have the shape file of Census tracts in Hamden.

We just need to count up how many times a traffic stop occurred in each tract.

First, let's make sure it will match the correct coordinate reference system (CRS) as the shapefile we've just downloaded. We'll use the `st_as_sf()` function to create a new geometry with the latitude and longitude data from the **stops** data frame. And we'll transform the CRS so it matches the CRS from the **new_haven** shapefile we downloaded.

```
stops_spatial <- stops %>%
  st_as_sf(coords=c("InterventionLocationLongitude", "InterventionLocationLatitude"),
           crs = "+proj=longlat") %>%
  st_transform(crs=st_crs(new_haven))
```

Now we use the `spatial_join()` function that sees where the geometries we've set in **stops_spatial** fall into which polygon in **new_haven**.

```
points_in <- st_join(new_haven, stops_spatial, left=T)
```

```
View(points_in)
```

This is great.

What just happened: Every point in the original **stops** data frame now has a corresponding census tract and has been saved in the **points_in** data frame.

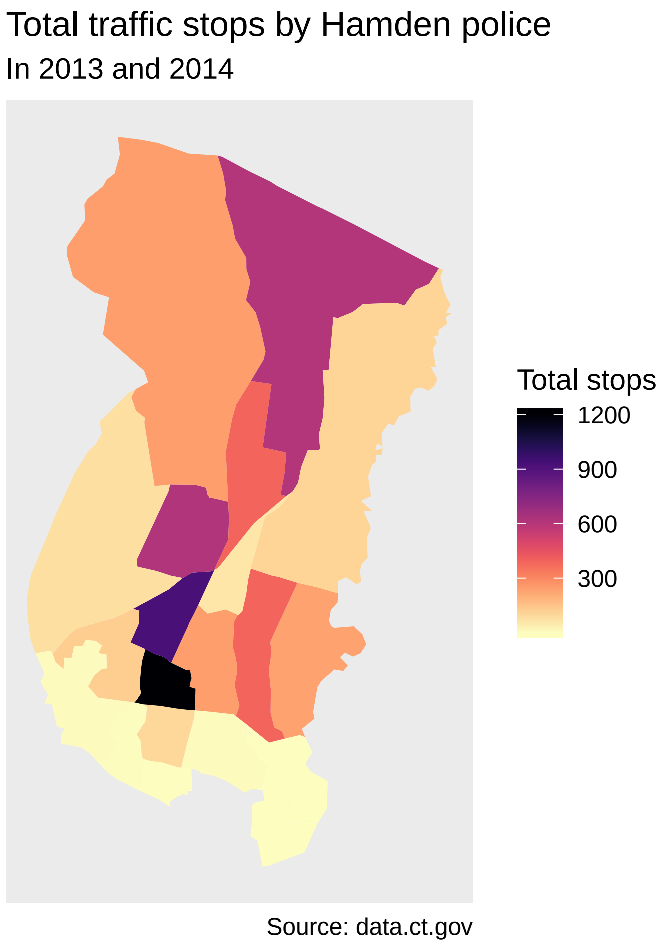
Now, we can summarize the data by count and merge it back to the shape file and visualize it.

```
by_tract <- points_in %>%
  filter(!is.na(X)) %>%
  group_by(GEOID) %>%
  summarise(total=n())

head(by_tract)
```

```
## Simple feature collection with 6 features and 2 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -72.97291 ymin: 41.31282 xmax: -72.9033 ymax: 41.35039
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
## # A tibble: 6 x 3
##   GEOID       total                                               geometry
##   <chr>       <int>                                         <POLYGON [°]>
## 1 09009141300     6 ((-72.97291 41.3474, -72.96833 41.34797, -72.96767 41~
## 2 09009141400     3 ((-72.95252 41.3228, -72.95084 41.32394, -72.94852 41~
## 3 09009141500    97 ((-72.94236 41.32899, -72.9397 41.33203, -72.93923 41~
## 4 09009141600     1 ((-72.94043 41.31991, -72.94131 41.32058, -72.94048 4~
## 5 09009141800     6 ((-72.92738 41.32651, -72.92708 41.32729, -72.92654 4~
## 6 09009141900     9 ((-72.91891 41.31964, -72.91649 41.32468, -72.91579 4~
```

We have enough here to visualize it.

```
# If you don't have viridis installed yet, uncomment and run the line below
#install.packages("viridis")

library(viridis)

ggplot(by_tract) +
  geom_sf(aes(fill = total), color=NA) +
  coord_sf(datum=NA) +
  labs(title = "Total traffic stops by Hamden police",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov",
       fill = "Total stops") +
  scale_fill_viridis(option="magma", direction=-1)
```



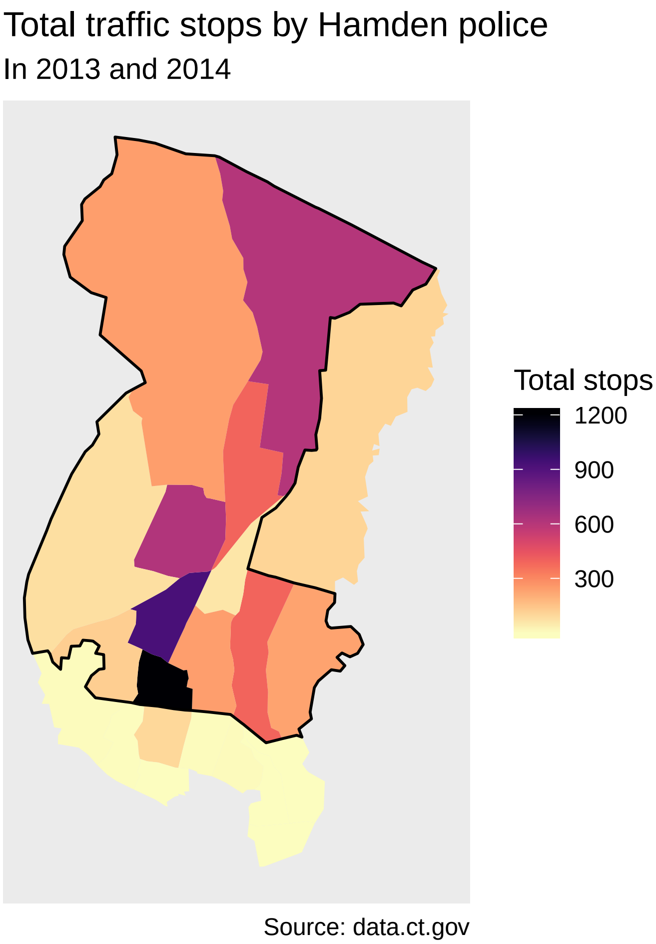

Pretty, but we're unclear which part is Hamden and which are parts of other towns.

That's fine because we can layer in a tract of Hamden only with **tigris**.

```
new_haven_towns <- county_subdivisions(state="CT", county="New Haven", cb=T)
hamden_town <- filter(new_haven_towns, NAME=="Hamden")
```

We've got a single polygon for Hamden now. Let's place it on top of our other map layers with a second `geom_sf()`.

```
ggplot() +
  geom_sf(data=by_tract, aes(fill = total), color=NA) +
  geom_sf(data=hamden_town, fill=NA, color="black") +
  coord_sf(datum=NA) +
  labs(title = "Total traffic stops by Hamden police",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov",
       fill = "Total stops") +
  scale_fill_viridis(option="magma", direction=-1) +
NULL
```




Alright, excellent.

It's much clearer now that the bulk of the traffic stops occur at the southern border of Hamden.

We can go deeper by going to our joined data frame and summarize by race and adding more variables to `group_by()`

```
by_tract_race <- points_in %>%
  filter(!is.na(X)) %>%
  group_by(GEOID, race) %>%
  summarise(total=n())

head(by_tract_race)
```

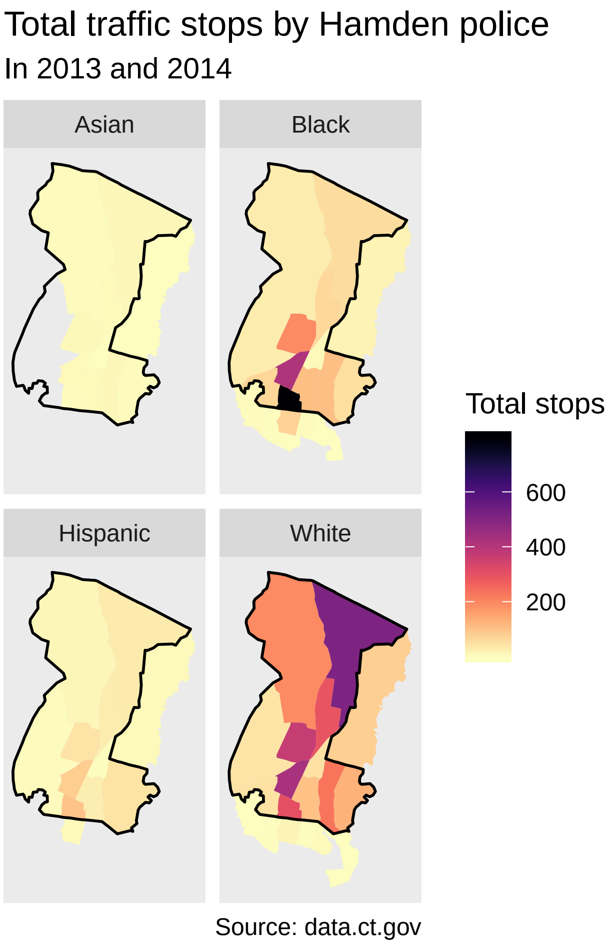
```
## Simple feature collection with 6 features and 3 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -72.97291 ymin: 41.31666 xmax: -72.9249 ymax: 41.35039
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
## # A tibble: 6 x 4
## # Groups:   GEOID [3]
##   GEOID       race     total                                       geometry
##   <chr>       <chr>    <int>                                  <POLYGON [°]>
## 1 09009141300 Black        5 ((-72.97291 41.3474, -72.96833 41.34797, -72~
## 2 09009141300 White        1 ((-72.97291 41.3474, -72.96833 41.34797, -72~
```

```
## 3 09009141400 Black        2 ((-72.95252 41.3228, -72.95084 41.32394, -72~
## 4 09009141400 White        1 ((-72.95252 41.3228, -72.95084 41.32394, -72~
## 5 09009141500 Black       73 ((-72.94236 41.32899, -72.9397 41.33203, -72~
## 6 09009141500 Hispanic     7 ((-72.94236 41.32899, -72.9397 41.33203, -72~
```

Very tidy data frame!

We can repurpose the map code above and add a single line of code to facet it.

```
ggplot() +
  geom_sf(data=by_tract_race, aes(fill = total), color=NA) +
  geom_sf(data=hamden_town, fill=NA, color="black") +
  coord_sf(datum=NA) +
  labs(title = "Total traffic stops by Hamden police",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov",
       fill = "Total stops") +
  scale_fill_viridis(option="magma", direction=-1) +
  facet_wrap(~race)
```




Well, that's pretty revealing.

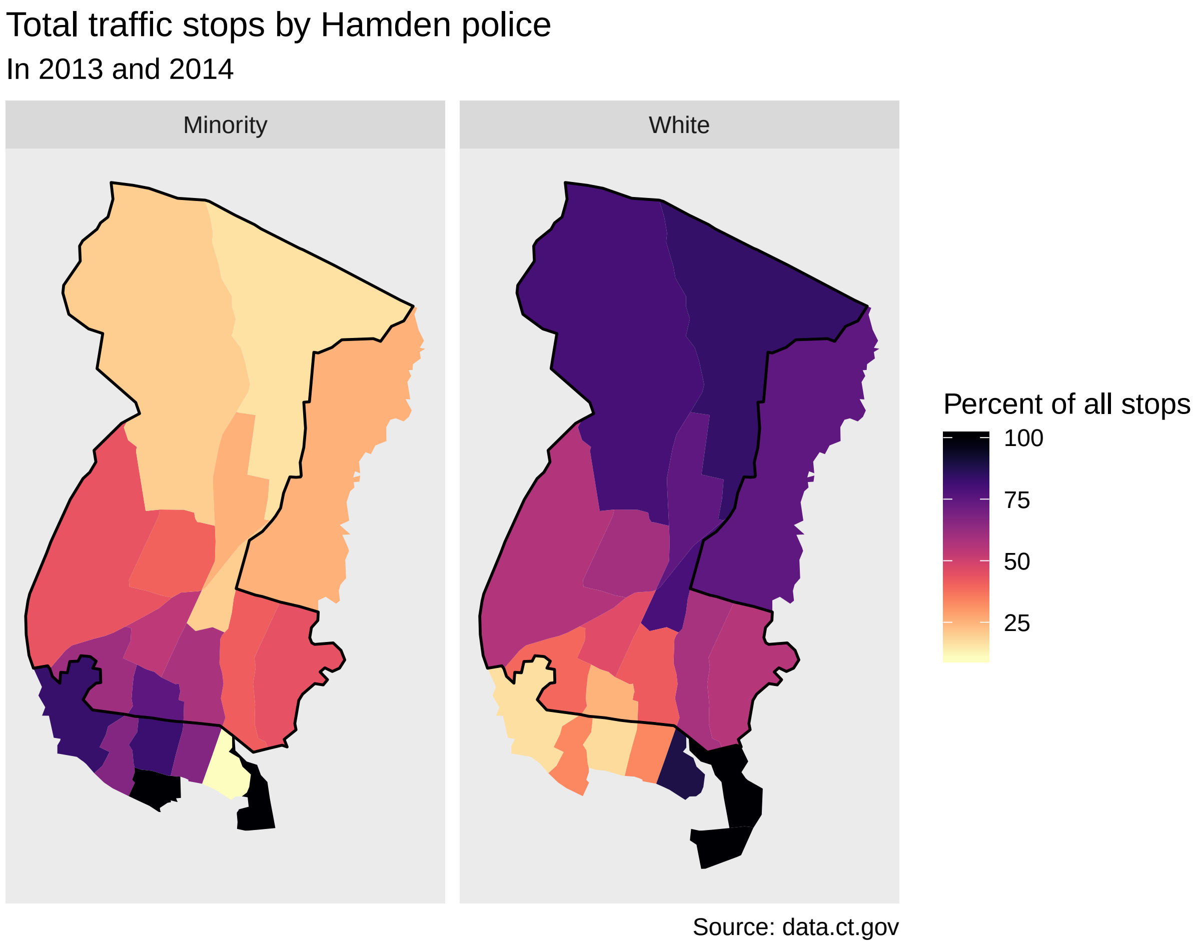
So these are raw numbers. Let's try to figure out the percent breakdown of drivers who are White versus those who aren't per Census tract. We just have to wrangle *by_tract_race* data frame a little bit. We've done this before in previous sections.

```r
by_tract_race_percent <- by_tract_race %>%
  mutate(type=case_when(
    race=="White" ~ "White",
    TRUE ~ "Minority")) %>%
  group_by(GEOID, type) %>%
  summarize(total=sum(total)) %>%
  mutate(percent=round(total/sum(total, na.rm=T)*100,2))

head(by_tract_race_percent)
```

```
## Simple feature collection with 6 features and 4 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: -72.97291 ymin: 41.31666 xmax: -72.9249 ymax: 41.35039
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
## # A tibble: 6 x 5
## # Groups:   GEOID [3]
##   GEOID       type     total percent                              geometry
##   <chr>       <chr>    <int>   <dbl>                         <POLYGON [°]>
## 1 09009141300 Minority     5    83.3 ((-72.97291 41.3474, -72.96833 41.34~
## 2 09009141300 White        1    16.7 ((-72.97291 41.3474, -72.96833 41.34~
## 3 09009141400 Minority     2    66.7 ((-72.95252 41.3228, -72.95084 41.32~
## 4 09009141400 White        1    33.3 ((-72.95252 41.3228, -72.95084 41.32~
## 5 09009141500 Minority    80    82.5 ((-72.94236 41.32899, -72.9397 41.33~
## 6 09009141500 White       17    17.5 ((-72.94236 41.32899, -72.9397 41.33~
```

We can easily this.

```r
ggplot() +
  geom_sf(data=by_tract_race_percent, aes(fill = percent), color=NA) +
  geom_sf(data=hamden_town, fill=NA, color="black") +
  coord_sf(datum=NA) +
  labs(title = "Total traffic stops by Hamden police",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov",
       fill = "Percent of all stops") +
  scale_fill_viridis(option="magma", direction=-1) +
  facet_wrap(~type)
```




So that's even more stark difference.

What's it tell us? Most of the stops up north are White drivers.

Most of the stops in the southern part of the town, particularly by the town border, are Minority drivers.

What's one argument that could explain this?

"Well, maybe that's where minorities live."

Perhaps. But we can measure that thanks to the Census.

We know the percent make up of traffic steps in Hamden.

Let's calculate the percent make up of residents in those neighborhoods and compare them.

Ideally, the rate of traffic stops should match the rate of residents, right?

We'll use the **tidycensus** package.

Don't forget to load your Census API key.

```
census_api_key("YOUR API KEY GOES HERE")
```

```
## To install your API key for use in future sessions, run this function with `install = TRUE`.
```

```
racevars <- c(Total_Pop = "B02001_001E",
              White_Pop = "B02001_002E")

hamden_pop <- get_acs(geography = "tract", variables = racevars,
```

```r
                  state = "CT", county = "New Haven County")
head(hamden_pop)
```

Great, we have total population and white population per tract.

Let's summarize this data by *GEOID*.

```r
library(tidyr)

hamden_pop_perc <- hamden_pop %>%
  mutate(variable=case_when(
    variable=="B02001_001" ~ "Total",
    variable=="B02001_002" ~ "White")) %>%
  # dropping Margin of Error-- I know this is not ideal but for this purpose,
  # we'll get rid of it for now
  select(-moe) %>%
  spread(variable, estimate) %>%
  mutate(white_residents=round(White/Total*100,2),
         minority_residents=100-white_residents)

head(hamden_pop_perc)
```

```
## # A tibble: 6 x 6
##   GEOID       NAME             Total White white_residents minority_reside~
##   <chr>       <chr>            <dbl> <dbl>           <dbl>            <dbl>
## 1 09009120100 Census Tract 1~  6189  4762            76.9             23.1
## 2 09009120200 Census Tract 1~  6566  5112            77.9             22.1
## 3 09009125100 Census Tract 1~  4557  3880            85.1             14.9
## 4 09009125200 Census Tract 1~  6002  4933            82.2             17.8
## 5 09009125300 Census Tract 1~  4932  3103            62.9             37.1
## 6 09009125400 Census Tract 1~  3459  2612            75.5             24.5
```

Nice. Let's join it back to the *by_tract_race_percent* dataframe so we can calculate the gap.

```r
by_tract_race_percent_spread <- by_tract_race_percent %>%
  select(-total) %>%
  spread(type, percent) %>%
  rename(minority_stopped=Minority, white_stopped=White) %>%
  filter(!is.na(minority_stopped) & !is.na(white_stopped))

stops_population <- left_join(by_tract_race_percent_spread, hamden_pop_perc)
```

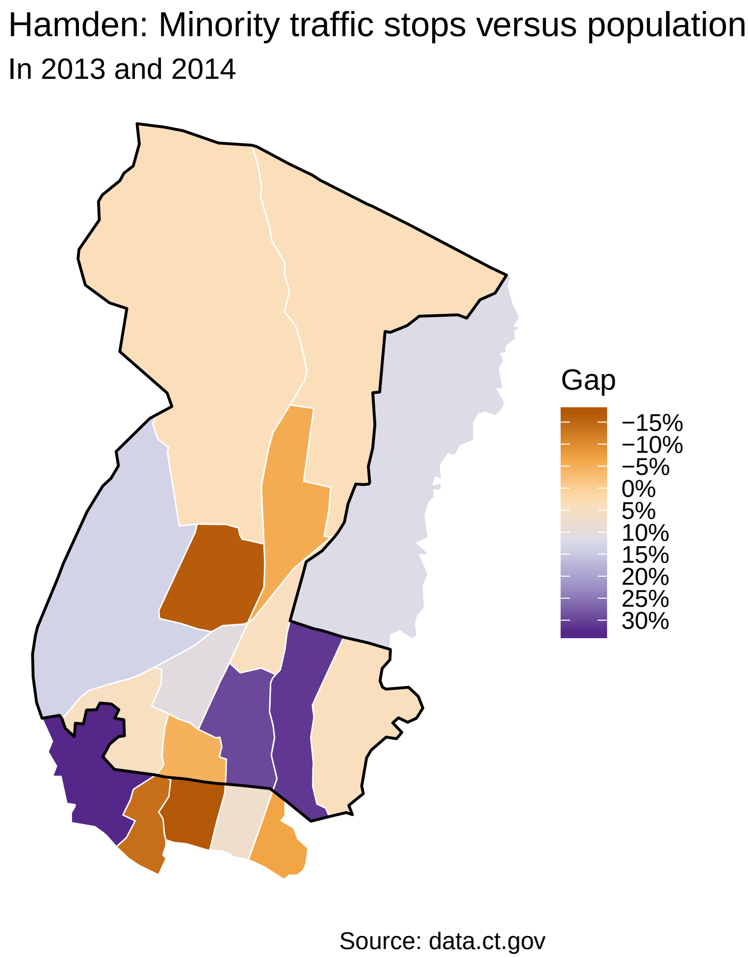
Great. Let's do some math real quick and visualize what we have with a diverging color palette from the **scales** package.

```r
# If you don't have scales installed yet, uncomment and run the line below
#install.packages("scales")

library(scales)

stops_population$gap <- (stops_population$minority_stopped -
                           stops_population$minority_residents)/100

ggplot() +
  geom_sf(data=stops_population, aes(fill = gap), color="white", size=.25) +
  geom_sf(data=hamden_town, fill=NA, color="black") +
  coord_sf(datum=NA) +
```

| | GEOID | minority_stopped | white_stopped | NAME | Total | White | white_residents | minority_residents | geometry |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 09009141300 | 83.33 | 16.67 | Census Tract 1413, New Haven County, Connecticut | 6415 | 3170 | 49.42 | 50.58 | list(c(−72.9508 |
| 2 | 09009141400 | 66.67 | 33.33 | Census Tract 1414, New Haven County, Connecticut | 4561 | 885 | 19.40 | 80.60 | list(c(−72.9423 |
| 3 | 09009141500 | 82.29 | 17.71 | Census Tract 1415, New Haven County, Connecticut | 5639 | 33 | 0.59 | 99.41 | list(c(−72.9273 |
| 4 | 09009141800 | 66.67 | 33.33 | Census Tract 1418, New Haven County, Connecticut | 4226 | 1705 | 40.35 | 59.65 | list(c(−72.9277 |
| 5 | 09009141900 | 11.11 | 88.89 | Census Tract 1419, New Haven County, Connecticut | 5305 | 4365 | 82.28 | 17.72 | list(c(−72.9157 |
| 6 | 09009165100 | 45.00 | 55.00 | Census Tract 1651, New Haven County, Connecticut | 4226 | 2533 | 59.94 | 40.06 | list(c(−72.9025 |
| 7 | 09009165200 | 41.09 | 58.91 | Census Tract 1652, New Haven County, Connecticut | 2309 | 2073 | 89.78 | 10.22 | list(c(−72.9130 |
| 8 | 09009165300 | 18.97 | 81.03 | Census Tract 1653, New Haven County, Connecticut | 2407 | 2028 | 84.25 | 15.75 | list(c(−72.9088 |
| 9 | 09009165400 | 58.20 | 41.80 | Census Tract 1654, New Haven County, Connecticut | 4849 | 3464 | 71.44 | 28.56 | list(c(−72.9130 |
| 0 | 09009165500 | 74.73 | 25.27 | Census Tract 1655, New Haven County, Connecticut | 4602 | 947 | 20.58 | 79.42 | list(c(−72.9246 |
| 1 | 09009165600 | 54.07 | 45.93 | Census Tract 1656, New Haven County, Connecticut | 5915 | 3352 | 56.67 | 43.33 | list(c(−72.9397 |
| 2 | 09009165700 | 60.90 | 39.10 | Census Tract 1657, New Haven County, Connecticut | 4432 | 1961 | 44.25 | 55.75 | list(c(−72.9405 |
| 3 | 09009165801 | 39.64 | 60.36 | Census Tract 1658.01, New Haven County, Connecticut | 6124 | 2644 | 43.17 | 56.83 | list(c(−72.9423 |
| 4 | 09009165802 | 48.91 | 51.09 | Census Tract 1658.02, New Haven County, Connecticut | 4501 | 3161 | 70.23 | 29.77 | list(c(−72.9423 |
| 5 | 09009165900 | 20.00 | 80.00 | Census Tract 1659, New Haven County, Connecticut | 8183 | 6860 | 83.83 | 16.17 | list(c(−72.9136 |
| 6 | 09009166001 | 25.45 | 74.55 | Census Tract 1660.01, New Haven County, Connecticut | 6293 | 4348 | 69.09 | 30.91 | list(c(−72.9154 |
| 7 | 09009166002 | 16.28 | 83.72 | Census Tract 1660.02, New Haven County, Connecticut | 7635 | 6679 | 87.48 | 12.52 | list(c(−72.8888 |
| 8 | 09009167100 | 25.47 | 74.53 | Census Tract 1671, New Haven County, Connecticut | 8222 | 7112 | 86.50 | 13.50 | list(c(−72.8746 |

Figure 3:

```
  labs(title = "Hamden: Minority traffic stops versus population",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov") +
  scale_fill_distiller(type="seq", trans="reverse", palette = "PuOr",
                       label=percent, breaks=pretty_breaks(n=10), name="Gap") +
  #continuous_scale(limits=c(-30, 30)) +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent')) +
NULL
```




#### Add annotations to graphic

We'll use the annotate() function with a combination of segments, points, and text.

```
stops_population$gap <- (stops_population$minority_stopped -
                           stops_population$minority_residents)/100

ggplot() +
  geom_sf(data=stops_population, aes(fill = gap), color="white", size=.25) +
  geom_sf(data=hamden_town, fill=NA, color="black") +
  coord_sf(datum=NA) +
  labs(title = "Hamden: Minority traffic stops versus population",
       subtitle = "In 2013 and 2014",
       caption = "Source: data.ct.gov") +
  scale_fill_distiller(type="seq", trans="reverse", palette = "PuOr", label=percent,
                       breaks=pretty_breaks(n=10), name="Gap") +
  #continuous_scale(limits=c(-30, 30)) +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent')) +
  # NEW CODE HERE
  annotate("segment", x = -72.93, xend = -72.87, y = 41.325, yend = 41.325,
           colour = "lightblue", size=.5) +
  annotate("point", x = -72.93, y = 41.325, colour = "lightblue", size = 2) +
```

```
annotate("text", x = -72.85, y = 41.325, label = "New Haven", size=5,
         colour="gray30") +
annotate("segment", x = -72.89, xend = -72.86, y = 41.375, yend = 41.375,
         colour = "lightblue", size=.5) +
annotate("point", x = -72.89, y = 41.375, colour = "lightblue", size = 2) +
annotate("text", x = -72.845, y = 41.375, label = "Hamden", size=5,
         colour="gray30") +
annotate("point", x = -72.83, y = 41.375, colour="white", size=.2) +
NULL
```