In the Dark: Exploring Racial Disparities in Traffic Stops Before and After Sunset

Brandon Tran '21, Pomona College, Claremont, CA
Preet Khowaja '20, Pomona College, Claremont, CA

December 2019

1 | Introduction & Motivation (Preet)

2 | The Data (Brandon)

In order to address our question in the context of California, it was necessary to acquire data that contained police records of traffic stops conducted throughout the past decade that included information on the time and location of stop, as well as the race of the driver who was stopped. Fortunately, The Stanford Open Policing Project hosts an public amalgamation of such records for forty-one US states, and since the published records were collected from government-hosted sources that are open to the public, we faced no ethical issues in terms of utilizing the data.

As expected, some manipulation was necessary in order to merge the data from cities across California and determine (1) whether or not a stop occurred during daylight or not and (2) how many minutes prior to or after sunset the stop occurred. The overarching goal was to write a series of functions that, given merely URLs of data hosted on Stanford's site and city codes for the corresponding cities, would return an amalgamated dataset with the following variables: `date`, `time`, `lat` (latitude), `lng` (longitude), `subject_race`, `daytime` (a boolean that returns `TRUE` if the stop occurs during the daytime), `minsto.set` (minutes to or after sunset), and `city` (the city code). A walkthrough of these functions will be included here, and the reproducible code itself can be found in **Appendix A**.

Via the R packages `StreamMetabolism` and `hms`, we were able to write **get_time**, which returns a vector containing sunrise and sunset times when given a latitude, longitude, and date. The function corrects for daylight savings when applicable and calculates the exact sunrise and sunset times for the given day. **classify** then returns whether or not a given time has occurred before sunset and after sunrise via a boolean that evaluates to `TRUE`. Both of these functions inform **mutateClass**, which calculates the aforementioned boolean for each row in a dataframe and appends a column, `daytime`, with these classifications. It also determines the number of minutes before or after sunset that the stop occurred, where values before are positive and values after are negative.

Now, in order to circumvent the issue of some datasets missing the exact latitude and longitude of each stop, we wrote **getCityCoords**, which takes in a link to a city's data and returns the city center's coordinates. Unfortunately, this is the one non-automated step of our data collection in that the city center's coordinates must be manually inputted into the function. We hope to automate this in the future to allow for easy amalgamation of all data that lacks specific coordinates.

This function is then used in **getData**, which takes in only a link to data and returns a sample (1:100) of the data with all of the modifications of **mutateClass** applied. At this point, the function is ready to be utilized on individual datasets specific to each city.

Our final step was to write a master function that would perform **getData** on any amount of datasets, then amalgamate all of the resulting samples into one large dataframe. This function is **parseData**, which takes in a vector of links to data from The Stanford Open Policing Project site and a vector of corresponding city codes and returns the desired dataframe.

The following code will generate a copy of that data and load it into your Global Environment as `ca.df`, though note that it will be necessary to evaluate the R chunks in Appendix A **prior** to running this code (at this time, all chunks have been set to `eval=FALSE`). With four cores, it will take approximately five to ten minutes to generate the data.

```
links <- c('https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_san_francisco_2019_08_13.r
           'https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_oakland_2019_08_13.rds',
           'https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_san_jose_2019_08_13.rds',
           'https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_bakersfield_2019_08_13.rds
           'https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_los_angeles_2019_08_13.rds
           'https://stacks.stanford.edu/file/druid:hp256wp2687/hp256wp2687_ca_san_diego_2019_08_13.rds')
city.codes <- c('sf', 'ok', 'sj', 'bf', 'la', 'sd')

ca.df <- parseData(links, city.codes)
```

As an alternative, a saved version of the dataset has been loaded into the **Data** folder of this repository and can be loaded and stored into your Global Environment with the following code:

```
ca.df <- readRDS(file='Data/cadata.Rda')
```

This completes the preprocessing necessary to access the data that was used in our analyses.

3 | Statistical Findings (Brandon)

We performed a χ^2 test for Independence in order to evaluate the null hypothesis: that `race` and `time of arrest` are independent of each other, where `time of arrest` refers to whether or not a stop is before or after sunset. With 5 degrees of freedom, the test yielded a p-value of $p = 2.2 \times 10^{-16}$, which is less than $\alpha = .05$, thus allowing us to reject the null and conlcude that the two variables are, in fact, dependent.

In more colloquial terms, we disproved the assumption that knowing what time of day it is has no effect on one's guess of the race of a random driver. Rather, a relationship does, in fact, exist here: If given the time of day, we should guess a certain race moreso than the others when considering a random driver.

Which race, specifically, should we guess? Unforunately, our test does not tell us this, but some local analysis allowed us to test some of our predispositions. Specifically, we tested `white` against `Asian/Pacific Islander`, `Black`, and `Hispanic`. For the sake of technicality: Under Bonferroni's Inequality, we needed the sum of our p-values to be less than $\alpha = .05$ for significance. When comparing each of the above races against `white`, we gain small p-values that sum to less than $\alpha = .05$. Specifically, these p-values are 4.03×10^{-12}, 2.2×10^{-16}, and 2.2×10^{-16}, for `Asian/Pacific Islander`, `Black`, and `Hispanic`, respectively, suggesting that the distribution of `white` traffic stops differs from the rest.

The code for the requisite χ^2 tests can be found in **Appendix B** and can be run after generating the dataset specified in Section 2 and Appendix A.

Visualizations will aid in the interpretation of these results.

4 | Visualizations (Preet)

5 | Limitations (Brandon & Preet)

From a statistical standpoint, we cannot generally say any more than the fact that a driver's race is **not** independent of whether or not a traffic stop is conducted during the daytime or nighttime. However, local analysis reveals that this independence is individually seen between each non-white race and white stops;

that is, there is some indication that the disparity between daytime and nighttime stops for white drivers is different from that of every other race.

Visualizations suggest that the difference detected by the χ^2 tests is that white drivers are stopped more often during the day than at night, whereas for non-white drivers, an approximately equal number of stops are conducted during the day and night. While this does not allow us to statistically conclude that police bias exists, or even that white drivers are stopped more often during the day than at night, it suggests that these assertions are true.

6 | Conclusion (Preet)

Include something about what we learned outside of what was covered in class.

Works Cited

Kirill Müller (2019). hms: Pretty Time of Day. R package version 0.5.2. https://CRAN.R-project.org/package=hms.

Pierson, Emma, et al. "A large-scale analysis of racial disparities in police stops across the United States." arXiv preprint arXiv:1706.05678 (2017). https://5harad.com/papers/100M-stops.pdf.

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Stephen Sefick Jr. (2016). Stream Metabolism-A package for calculating single station metabolism from diurnal Oxygen curves R package version 1.1.2. https://cran.r-project.org/package=StreamMetabolism.

Appendix A

The following is code that generated the dataset that was used for our statistical analyses and visualizations. Explanations for each function can be found in Section 2. All code has been set to `eval=FALSE` in order to allow you to execute code chunk by chunk.

```r
library(StreamMetabolism)
library(hms)
```

```r
get_time <- function(lat, long, date) {
  #Takes lat, long, time, and date as input and classifies
  #TRUE if the time is at day and FALSE if at night.

  sunrise <- sunrise.set(lat, long, date)$sunrise
  attributes(sunrise)$tzone <- 'America/Los_Angeles'
  sunrise <- as_hms(sunrise)

  sunset <- sunrise.set(lat, long, date)$sunset
  attributes(sunset)$tzone <- 'America/Los_Angeles'
  sunset <- as_hms(sunset)

  return(c(sunrise, sunset))
}

classify <- function(rise.set, time) {
  #Takes in a vector with sunrise and sunset times,
  #returning True if the time is during daylight.

  if (time < rise.set[2] && time >= rise.set[1])
    return(TRUE)

  else
    return(FALSE)
}

mutateClass <- function(df) {
  #Takes a dataframe and adds the boolean day/night
  #classification found via `classify` as a new column.

  classifications <- c()
  timetoset <- c()

  for (i in 1:nrow(df)) {
    time <- get_time(df$lat[i], df$lng[i], df$date[i])
    timetoset[i] <- (time[2] - df$time[i])/60
    classifications[i] <- classify(time, df$time[i])
  }

  df <- df %>%
    mutate(daytime = classifications) %>%
    mutate(minsto.set = timetoset)
}

getCityCoords <- function(link) {
  #Takes a link to data and assigns appropriate lat
  #and lng coordinates.

  if (grepl('los_angeles', link))
    return(c(34.052235, -118.243683))

  if (grepl('san_diego', link))
    return(c(32.715736, -117.161087))

  if (grepl('oakland', link))
    return(c(37.804363, -122.271111))

  if (grepl('san_jose', link))
    return(c(37.335480, -121.893028))
}

getData <- function(link) {
  #Takes a link to data and configures it as necessary.
  #Samples n/10000 traffic stops.  Returns a mutated
  #`daytime` variable.

  set.seed(47)

  raw <- readRDS(file = url(link))

  if (! 'lat' %in% colnames(raw)) {
    raw <- raw %>%
      mutate(lat = getCityCoords(link)[1]) %>%
      mutate(lng = getCityCoords(link)[2])
  }

  clean <- raw %>%
    select(c(date, time, lat, lng, subject_race)) %>%
    drop_na() %>%
    sample_n(nrow(raw)/100)

  return(mutateClass(clean))
}

parseData <- function(links, city.codes) {
  #Takes in a list of URLs and city codes and
  #merges the datasets that are scraped from the sources.

  master.frame <- getData(links[1])
  master.frame <- master.frame %>% mutate(city = city.codes[1])

  for (i in 2:length(links)) {
    data <- getData(links[i])
    data <- data %>% mutate(city = city.codes[i])
    master.frame <- full_join(master.frame, data)
  }

  return(master.frame)
```

```
}
```

Appendix B

The following is code that will perform the χ^2 tests for independence that were described in Section 3. This must be run after you have either loaded the `ca.df` dataset into your Global Environment or generated a local copy of the data (see Section 2 and Appendix A).

```
#In General
chisq.test(ca.df$subject_race, ca.df$daytime)

#Local Inference for API
ca.api <- ca.df %>%
  filter(subject_race == "asian/pacific islander" | subject_race == "white")

chisq.test(ca.api$subject_race, ca.api$daytime)

#Local Inference for Black
ca.black <- ca.df %>%
  filter(subject_race == "black" | subject_race == "white")

chisq.test(ca.black$subject_race, ca.black$daytime)

#Local Inference for Hispanic
ca.hisp <- ca.df %>%
  filter(subject_race == "hispanic" | subject_race == "white")

chisq.test(ca.hisp$subject_race, ca.hisp$daytime)
```