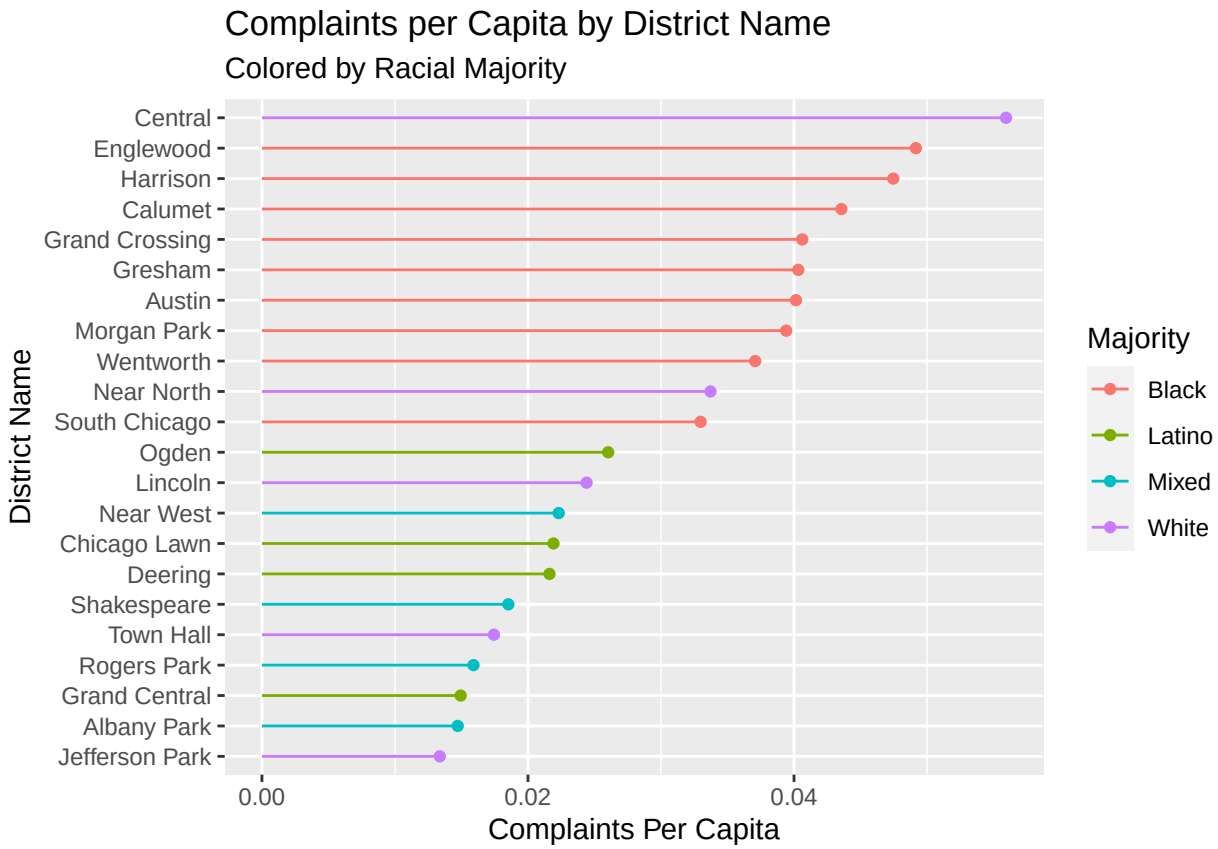
Visualizations

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v dplyr   1.0.8
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'kableExtra'
## The following object is masked from 'package:dplyr':
##
##     group_rows
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
## Skipping install of 'NHSRtheme' from a github remote, the SHA1 (6d2c3c82) has not changed since last
##   Use `force = TRUE` to force installation
## Rows: 22 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): District_Name, Latino%, White%, Black%, Native_American%, Asian%, O...
## dbl (2): District_No, Native_American
##
## 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.
## Rows: 13530 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): LAST_NME, FIRST_NME, EMPLOYEE_POSITION, CPD_UNIT_ASSIGNED_NO, UNITA...
## dbl (2): AGE, STAR_NO
##
## 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.
## Warning: One or more parsing issues, see `problems()` for details
## Rows: 125581 Columns: 22
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): gender, race, current_rank, complaint_category, recommended_findi...
## dbl   (7): row_id, cr_id, birth_year, current_unit, current_star, recommende...
## lgl   (2): middle_initial, middle_initial2
## date  (1): appointed_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.
## Rows: 48214 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): gender, race
## dbl (2): cr_id, age
##
## 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.
## Rows: 131142 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): beat, location_code, address_number, street, apartment_number, cit...
## dbl  (2): row_id, cr_id
## date (3): incident_date, complaint_date, closed_date
## time (1): incident_time
##
## 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
district_complaints <- complaints_accused %>%
  filter(current_unit %in% 1:25) %>%
  group_by(current_unit) %>%
  summarise(n = n()) %>%
  arrange(desc(n))

total_district_complaints <- full_join(district_complaints,
                                       district_demographics,
                                       by = c("current_unit" = "District_No")) %>%
  mutate(complaints_per_capita = n/Population)

total_district_complaints %>%
  filter(is.na(District_Name) == FALSE) %>%
         ggplot(mapping = aes(
                  x = fct_reorder(District_Name, complaints_per_capita),
                  y = complaints_per_capita,
                color = Majority)) +
  geom_point() +
  geom_segment(aes(x = fct_reorder(District_Name, complaints_per_capita),
                   xend = fct_reorder(District_Name, complaints_per_capita),
                   y = 0, yend = complaints_per_capita)) +
  coord_flip() +
  labs(title = "Complaints per Capita by District Name",
       subtitle = "Colored by Racial Majority",
       x = "District Name",
       y = "Complaints Per Capita")
```




```
district_complaints_1 <- complaints_accused %>%
  filter(current_unit %in% 1:25) %>%
  group_by(current_unit)

total_district_complaints_findings <- full_join(district_complaints_1,
                                        district_demographics,
                                        by = c("current_unit" = "District_No"))

#stat = "identity"

data1 <- total_district_complaints_findings %>%
  mutate(final_decision =
           as.factor(case_when(final_finding %in% "SU" ~ "Sustained",
                               final_finding %in% "DIS" ~ "Sustained",
                               is.na(final_finding) == TRUE ~ "Missing",
                               final_finding %in% "NAF" ~ "No Affidavit or Cooperation",
                               final_finding %in% "NC" ~ "No Affidavit or Cooperation",
                               final_finding %in% "NS" ~ "Not Sustained",
                               final_finding %in% "EX" ~ "Not Sustained",
                               final_finding %in% "UN" ~ "Not Sustained",
                               ))) %>%
  mutate(District_Name = fct_relevel(District_Name,
                                     "Central",
                                     "Englewood",
                                     "Harrison",
                                     "Calumet",
                                     "Grand Crossing",
```

```
                                    "Gresham",
                                    "Austin",
                                    "Morgan Park",
                                    "Wentworth",
                                    "Near North",
                                    "South Chicago",
                                    "Ogden",
                                    "Lincoln",
                                    "Near West",
                                    "Chicago Lawn",
                                    "Deering",
                                    "Shakespeare",
                                    "Town Hall",
                                    "Rogers Park",
                                    "Grand Central",
                                    "Albany Park",
                                    "Jefferson Park"
                                    )) %>%
  group_by(final_decision, District_Name)  %>%
  summarize(n = n())
```

```
## Warning: Unknown levels in `f`: Englewood, Harrison, Calumet, Grand Crossing,
## Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Gresham,
## Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden, Lincoln, Near
## West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park, Grand Central,
## Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, Ogden, Lincoln,
## Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park, Grand
## Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Grand Crossing,
## Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Harrison, Calumet, Grand Crossing,
## Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park
```

```
## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
```

```
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Deering, Shakespeare, Town Hall, Rogers Park, Grand
## Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Calumet, Grand Crossing,
## Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Town Hall, Rogers Park, Grand
## Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Morgan Park, Wentworth, Near North, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Grand Central, Albany Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Grand Central, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, South Chicago, Ogden,
## Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
```

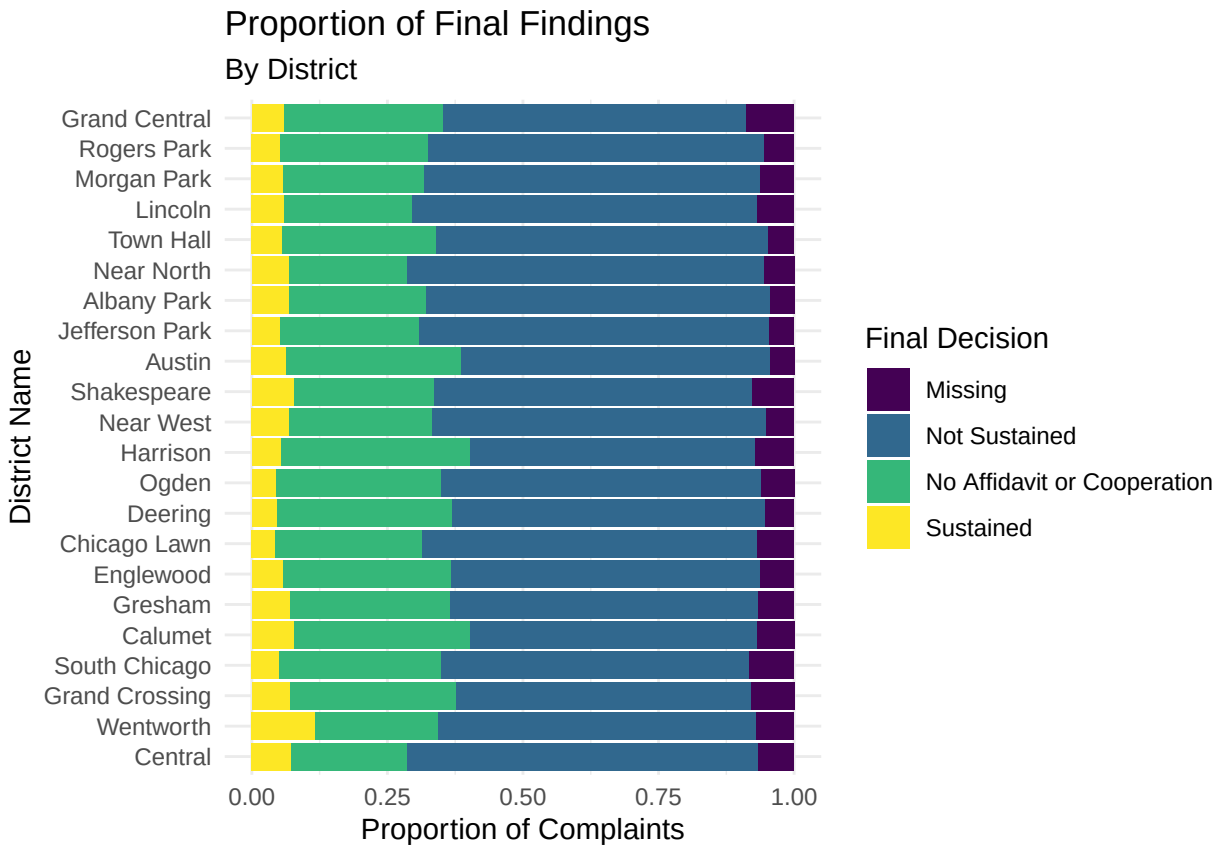
```
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park,
## Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Wentworth, Near North, South Chicago, Ogden, Lincoln,
## Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers Park, Grand
## Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Grand Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Grand
## Central, Albany Park, Jefferson Park

## Warning: Unknown levels in `f`: Central, Englewood, Harrison, Calumet, Grand
## Crossing, Gresham, Austin, Morgan Park, Wentworth, Near North, South Chicago,
## Ogden, Lincoln, Near West, Chicago Lawn, Deering, Shakespeare, Town Hall, Rogers
## Park, Albany Park, Jefferson Park

## `summarise()` has grouped output by 'final_decision'. You can override using
## the `.groups` argument.
```

```
data1 %>%
  group_by(final_decision) %>%
  summarize(n = n())
```

```
## # A tibble: 4 x 2
##   final_decision                 n
##   <fct>                      <int>
## 1 Missing                       23
## 2 No Affidavit or Cooperation   23
## 3 Not Sustained                 23
## 4 Sustained                     23
```

```
#reorder so that missing is at the end and change colors, take out the NA
  data1 %>%
  filter(is.na(District_Name) == FALSE) %>%
  ggplot(aes(fill = factor(final_decision, levels = c("Missing", "Not Sustained", "No Affidavit or Coop
                           x = fct_relevel(District_Name), district_levels,
                           y = n)) +
  geom_bar(position = "fill", stat = "identity") +
  theme_minimal() +
  scale_fill_viridis_d() +
  coord_flip() +
   labs(title = "Proportion of Final Findings",
        subtitle = "By District",
        x = "District Name",
        y = "Proportion of Complaints",
```

```
        fill = "Final Decision"
        )
```




```
total_district_complaints_findings %>%
  group_by(final_finding) %>%
  summarise(n = n(), perc_of_data = (n/71960)*100)
```

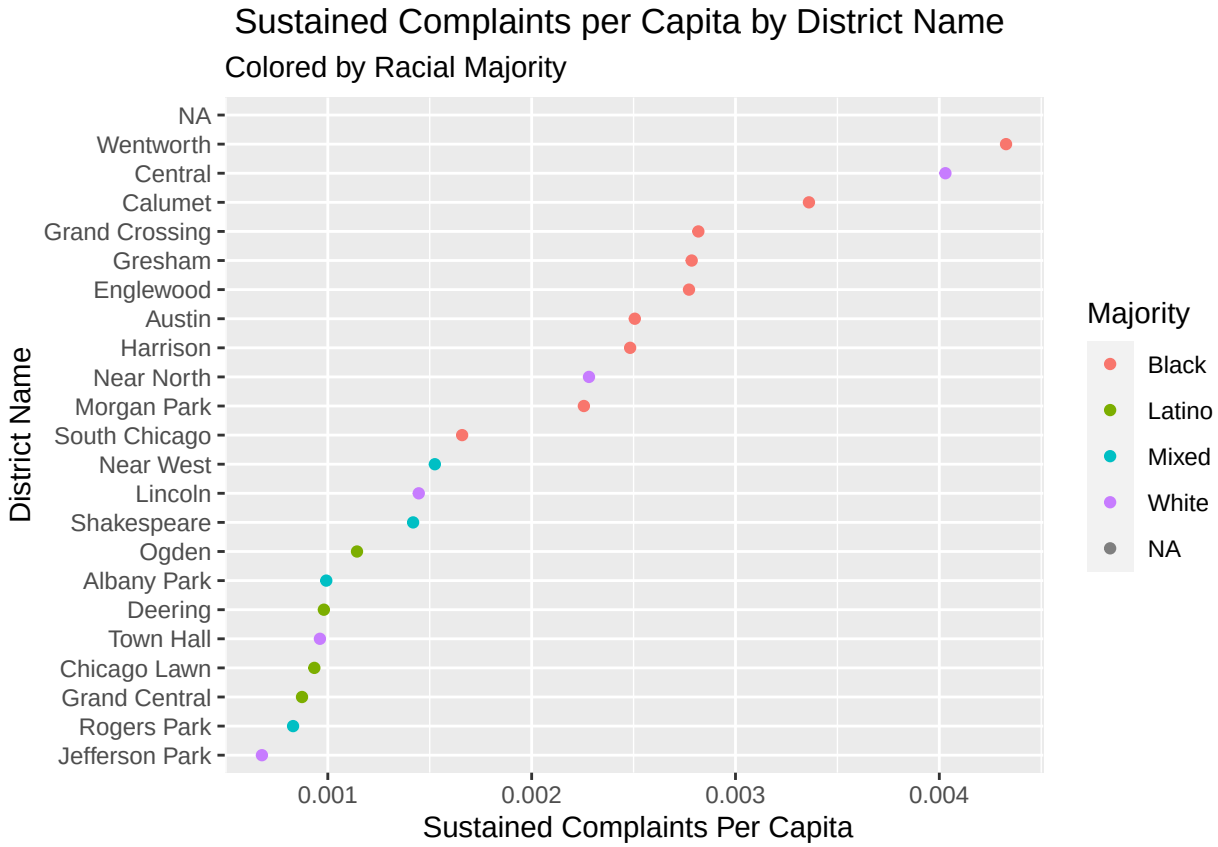
```
## # A tibble: 8 x 3
##   final_finding     n perc_of_data
##   <chr>         <int>        <dbl>
## 1 DIS               6      0.00834
## 2 EX             4714      6.55
## 3 NAF           19995     27.8
## 4 NC               29      0.0403
## 5 NS            21864     30.4
## 6 SU             4504      6.26
## 7 UN            16241     22.6
## 8 <NA>           4607      6.40
```

```
sustained_data <- total_district_complaints_findings %>%
  group_by(final_finding, District_Name, Majority, Population) %>%
  filter(final_finding == "SU") %>%
  summarize(n = n()) %>%
  mutate(complaints_per_capita = n/Population)
```

```
## `summarise()` has grouped output by 'final_finding', 'District_Name',
## 'Majority'. You can override using the `.groups` argument.
```

```
  ggplot(data = sustained_data,
                mapping = aes(
                  x = fct_reorder(District_Name, complaints_per_capita),
                  y = complaints_per_capita,
                color = Majority)) +
  geom_point() +
  coord_flip() +
  labs(title = " Sustained Complaints per Capita by District Name",
       subtitle = "Colored by Racial Majority",
       x = "District Name",
       y = " Sustained Complaints Per Capita")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



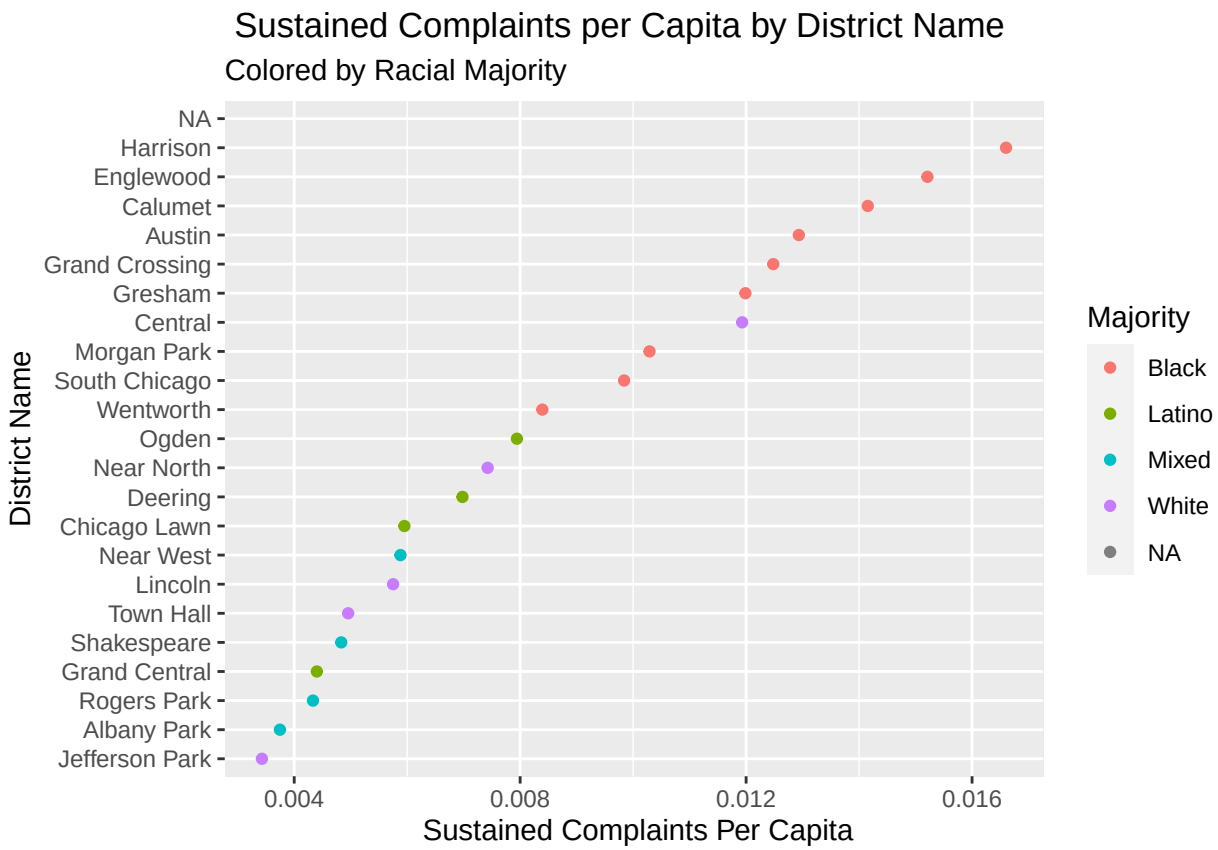

```
no_ca_data <- total_district_complaints_findings %>%
  group_by(final_finding, District_Name, Majority, Population) %>%
  filter(final_finding == "NAF") %>%
  summarize(n = n()) %>%
  mutate(complaints_per_capita = n/Population)
```

```
## `summarise()` has grouped output by 'final_finding', 'District_Name',
## 'Majority'. You can override using the `.groups` argument.
```

```
  ggplot(data = no_ca_data,
                mapping = aes(
                  x = fct_reorder(District_Name, complaints_per_capita),
                  y = complaints_per_capita,
                color = Majority)) +
```

```
  geom_point() +
  coord_flip() +
  labs(title = " Sustained Complaints per Capita by District Name",
       subtitle = "Colored by Racial Majority",
       x = "District Name",
       y = " Sustained Complaints Per Capita")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
#The districts of which demogrpahic majoirty compromise most of the data?

total_district_complaints_findings %>%
  group_by(Majority) %>%
  summarise(n = n(), prop_of_data = n/71960)
```

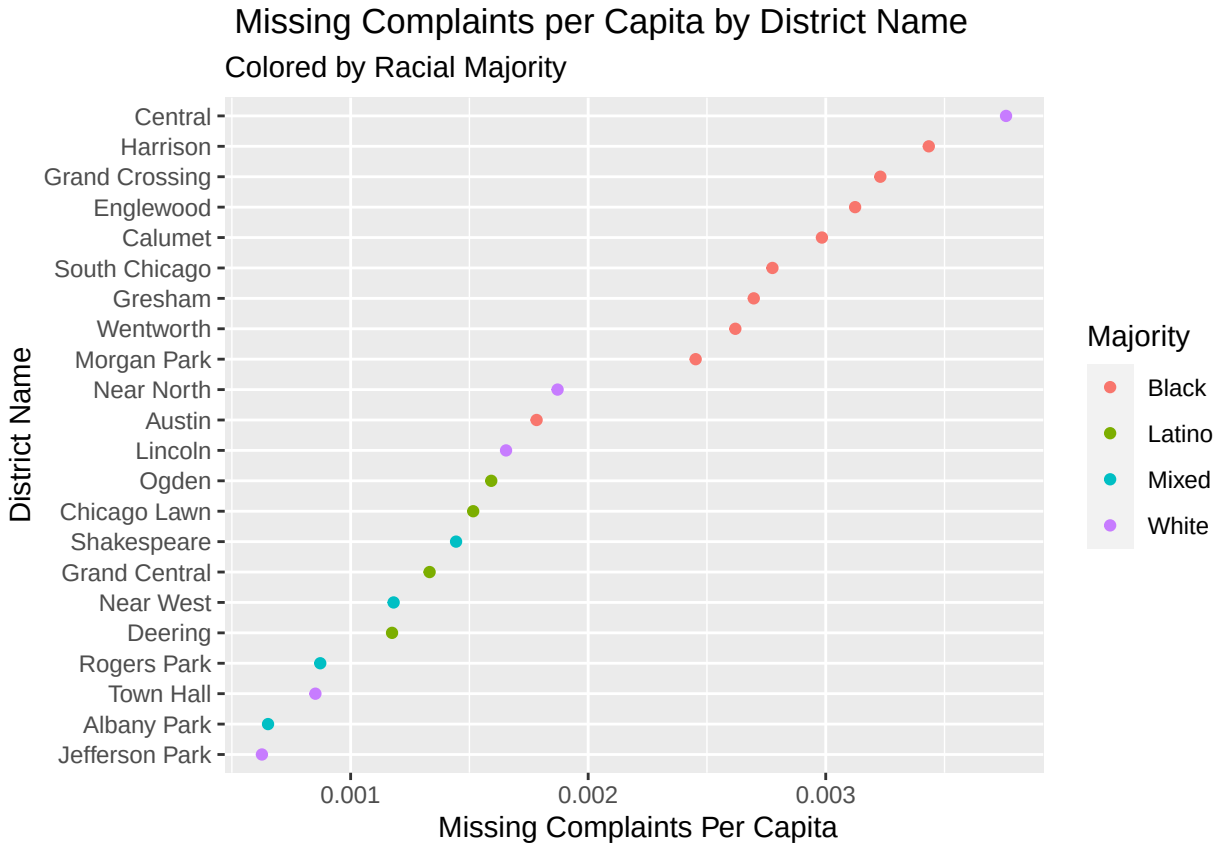
```
## # A tibble: 5 x 3
##   Majority     n prop_of_data
##   <chr>    <int>        <dbl>
## 1 Black    30818       0.428
## 2 Latino   15061       0.209
## 3 Mixed     9398       0.131
## 4 White    15856       0.220
## 5 <NA>       827       0.0115
```

```
missing_data <- total_district_complaints_findings %>%
  group_by(final_finding, District_Name, Majority, Population) %>%
  filter(is.na(final_finding)) %>%
  filter(!is.na(District_Name)) %>%
```

```
  summarize(n = n()) %>%
  mutate(complaints_per_capita = n/Population)
```

```
## `summarise()` has grouped output by 'final_finding', 'District_Name',
## 'Majority'. You can override using the `.groups` argument.
```

```
 ggplot(data = missing_data,
                mapping = aes(
                  x = fct_reorder(District_Name, complaints_per_capita),
                  y = complaints_per_capita,
                color = Majority)) +
  geom_point() +
  coord_flip() +
  labs(title = " Missing Complaints per Capita by District Name",
       subtitle = "Colored by Racial Majority",
       x = "District Name",
       y = " Missing Complaints Per Capita")
```




```
#no 21 or 23 district but 31st district included?

chicago_police_district_spatial <- st_read(dsn = "/cloud/project/data/geo_export_2efb16ec-aa66-49b0-92a
```

```
## Reading layer `geo_export_2efb16ec-aa66-49b0-92a0-2d6f5e0f81d9' from data source `/cloud/project/data
##   using driver `ESRI Shapefile'
## Simple feature collection with 25 features and 2 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -87.94011 ymin: 41.64455 xmax: -87.52414 ymax: 42.02303
```

```
## Geodetic CRS:  WGS84(DD)
```

```
total_district_complaints_spatial <- total_district_complaints %>%
  mutate(`Latino%` = str_remove(`Latino%`, "%"),
         `White%` = str_remove(`White%`, "%"),
         `Black%` = str_remove(`Black%`, "%"),
         `Asian%` = str_remove(`Asian%`, "%"),
         `Native_American%` = str_remove(`Native_American%`, "%"),
         `Other%` = str_remove(`Other%`, "%"),
         `Latino%` = as.numeric(`Latino%`),
         `White%` = as.numeric(`White%`),
         `Black%` = as.numeric(`Black%`),
         `Asian%` = as.numeric(`Asian%`),
         `Native_American%` = as.numeric(`Native_American%`),
         `Other%` = as.numeric(`Other%`)) %>%
  mutate(current_unit = as.character(current_unit))%>%
  left_join(chicago_police_district_spatial,
                                    by = c("current_unit" = "dist_num")) %>%
  st_as_sf() %>%
  st_transform("+init=epsg:4326")
```

```
## Warning in CPL_crs_from_input(x): GDAL Message 1: +init=epsg:XXXX syntax is
## deprecated. It might return a CRS with a non-EPSG compliant axis order.
```

```
bins <- seq(from = 0, to = 100, by = 12.5)
pal_perc <- colorBin("OrRd", domain = total_district_complaints_spatial , bins = bins)
#https://laurielbaker.github.io/DSCA_leaflet_mapping_in_r/slides/leaflet_slides3.html#58

m <- leaflet(total_district_complaints_spatial) %>%
# Now add tiles to it
  addTiles() %>%
# Setting the middle of where the map should be and the zoom level
  setView(-87.633506, 41.876067, zoom = 9.5) %>%
  addProviderTiles(providers$CartoDB.Positron)
```

```
Black_perc_m <- m %>%
  addPolygons(
    fillOpacity = 1,
    color = "black",
    opacity = 0.7,
    weight = 1,
    fillColor = ~pal_perc(total_district_complaints_spatial$`Black%`),
    highlightOptions = highlightOptions(
    weight = 5,
    color = "#666",
    fillOpacity = 0.7,
    bringToFront = FALSE))

Black_perc_m <- Black_perc_m %>%
  addLegend(
    position = "topright",
    pal = pal_perc,
    values = ~total_district_complaints_spatial$`Black%`,
    title = "Percent Black residents",
    opacity = 1)
```

```
Black_perc_m
```

```
## QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rstudio-user'
## TypeError: Attempting to change the setter of an unconfigurable property.
## TypeError: Attempting to change the setter of an unconfigurable property.
```

```
# White_perc_m <- m %>%
#   addPolygons(
#     fillOpacity = 1,
#     color = "black",
#     opacity = 0.7,
#     weight = 1,
#     fillColor = ~pal_perc(total_district_complaints_spatial$`White%`))
#
# White_perc_m <- White_perc_m %>%
#   addLegend(
#     position = "topright",
#     pal = pal_perc,
#     values = ~total_district_complaints_spatial$`White%`,
#     title = "Percent White residents",
#     opacity = 1)
#
# White_perc_m
```

```
# Latino_perc_m <- m %>%
#   addPolygons(
```

```
#     fillOpacity = 1,
#     color = "black",
#     opacity = 0.7,
#     weight = 1,
#     fillColor = ~pal_perc(total_district_complaints_spatial$`Latino%`))
#
# Latino_perc_m <- Latino_perc_m %>%
#   addLegend(
#     position = "topright",
#     pal = pal_perc,
#     values = ~total_district_complaints_spatial$`Latino%`,
#     title = "Percent Latino residents",
#     opacity = 1)
#
# Latino_perc_m
```

```
# creating map showing neighborhoods with most missing data ie when the final finding is either NA (mis
#baseline complaints per capita

# bins_2 <- seq(from = 0, to = 0.06, by = 0.01)
# pal_per_cap <- colorBin("OrRd", domain = total_district_complaints_spatial, bins = bins_2)
#
#  complaints_perc_m <- m %>%
#   addPolygons(
#     fillOpacity = 1,
#     color = "black",
#     opacity = 0.7,
#     weight = 1,
#     fillColor = ~pal_per_cap(total_district_complaints_spatial$`complaints_per_capita`))
#
# complaints_perc_m <- complaints_perc_m %>%
#   addLegend(
#     position = "topright",
#     pal = pal_per_cap,
#     values = ~total_district_complaints_spatial$`complaints_per_capita`,
#     title = "Complaints per capita",
#     opacity = 1)
#
# complaints_perc_m
```