# data exploration

There isnt really a process here, mostly aimless exploration. Often the written comments/text will refer to whatever is above it

```
dbListTables(con)
```

```
 [1] "ACGR"          "CensusDay"     "Counties"      "Districts"    
 [5] "MeetingDates"  "PriorityNine"  "PriorityOne"   "PrioritySeven"
 [9] "PrioritySix"   "PriorityTen"   "PriorityThree" "PriorityTwo"  
[13] "PublicSchools" "Schools"      
```

just some data exploration

```
dbGetQuery(con, "
  SELECT
  CAST(CountyCode AS TEXT) || CAST(DistrictCode AS TEXT) || CAST(SchoolCode AS TEXT) AS cdsCode, 
FROM CensusDay where LENGTH(SchoolCode) = 7 limit 5;
")
```

```
Warning: Column `GR_TK`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_KN`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_01`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_02`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_03`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_04`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_05`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_06`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_07`: mixed type, first seen values of type integer,
coercing other values of type string
```

```
Warning: Column `GR_08`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_09`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_10`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_11`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `GR_12`: mixed type, first seen values of type integer,
coercing other values of type string

        cdsCode AcademicYear AggregateLevel CountyCode DistrictCode SchoolCode
1 1100176001788      2023-24              S          1        10017    6001788
2 1100176001788      2023-24              S          1        10017    6001788
3 1100176001788      2023-24              S          1        10017    6001788
4 1100176001788      2023-24              S          1        10017    6001788
5 1100176001788      2023-24              S          1        10017    6001788
  Charter ReportingCategory TOTAL_ENR GR_TK GR_KN GR_01 GR_02 GR_03 GR_04 GR_05
1       Y           AR_0418       464    24    75    60    69    81    68    87
2       Y           ELAS_EL       253    23    44    33    33    41    43    36
3       Y           ELAS_EO       172     0    30    24    26    34    21    36
4       Y         ELAS_IFEP        14     0     0     0     0     0     0     0
5       Y         ELAS_RFEP        24     0     0     0     0     0     0     0
  GR_06 GR_07 GR_08 GR_09 GR_10 GR_11 GR_12
1     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0
3     0     0     0     0     0     0     0
4     0     0     0     0     0     0     0
5     0     0     0     0     0     0     0
```

```
dbGetQuery(con, "
  SELECT * FROM ACGR GROUP BY Year LIMIT 5;
")
```

```
  AdultEd AdultEdRate AggregateLevel Biliteracy BiliteracyRate CPP CPPRate
1     161         0.1              T      40348           18.5 485     0.2
  CharterSchool CohortStudents CountyCode DASS DistrictCode Dropout DropoutRate
1           All         246697          0  All           NA   19145         7.8
  Exemption ExemptionRate GED GEDRate Merit MeritRate Other OtherRate
1      3491           1.6 141     0.1 79802      36.6  1396       0.6
  RegHSDiploma RegHSDiplomaRaet ReportingCategory SPED SPEDRate SchoolCode
1       217760             88.3                GF 1553      0.6         NA
  StillEnrolled StillEnrolledRate UniReqs UniReqsPercent    Year cdsCode
1          6056               2.5  124388           57.1 2023-24       0
```

```
dbListFields(con, "PublicSchools")
```

```
 [1] "CDSCode"              "NCESDist"             "NCESSchool"
 [4] "StatusType"           "Street"               "City"
 [7] "Zip"                  "MailStreet"           "MailCity"
[10] "MailZip"              "Phone"                "PhoneExt"
[13] "FaxNumber"            "Website"              "OpenDate"
[16] "ClosedDate"           "Charter"              "CharterNum"
[19] "FundingType"          "DOC"                  "DOCType"
[22] "SOC"                  "SOCType"              "EdOpsCode"
[25] "EdOpsName"            "EILCode"              "EILName"
[28] "GSoffered"            "GSserved"             "Virtual"
[31] "Magnet"               "YearRound"            "FederalDFCDistrictID"
[34] "Latitude"             "Longitude"            "AdmFName"
[37] "AdmLName"             "LastUpDate"           "Multilingual"
[40] "CountyCode"           "DistrictCode"         "SchoolCode"
```

Above I explored the CensusDay table at aggregate level S by combining the codes into cdsCode since that would be necessary for potential joining. I also investigated the ACGR dataset where I learned we are only using one year, and it is formatted in “2023-2024” whereas other tables will only list one year. Potential issue in future joining, can be easily addressed.

```
df1<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PriorityOne WHERE year = 2024;
")
df2<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PriorityTwo WHERE year = 2024;
")
df3<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PriorityThree WHERE year = 2024;
")

df6<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PrioritySix WHERE year = 2024;
")
df7<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PrioritySeven WHERE year = 2024;
")
df9<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PriorityNine WHERE year = 2024;
")
df10<- dbGetQuery(con, "
  SELECT cdsCode, countyPerformance FROM PriorityTen WHERE year = 2024;
")

df1 <- df1 %>% rename(perf_1 = countyPerformance)
df2 <- df2 %>% rename(perf_2 = countyPerformance)
df3 <- df3 %>% rename(perf_3 = countyPerformance)
df6 <- df6 %>% rename(perf_6 = countyPerformance)
df7 <- df7 %>% rename(perf_7 = countyPerformance)
```

```
df9 <- df9 %>% rename(perf_9 = countyPerformance)
df10 <- df10 %>% rename(perf_10 = countyPerformance)
outcome<- dbGetQuery(con, "
  SELECT cdsCode, RegHSDiplomaRaet FROM ACGR WHERE cdsCode != 0 AND RegHSDiplomaRaet NOT IN ('*',
")
outcome <- outcome %>%
  mutate(RegHSDiplomaRaet = as.numeric(RegHSDiplomaRaet))
averaged_out <- outcome %>%
  group_by(cdsCode) %>%
  summarise(averageHSDiplomaRate = mean(RegHSDiplomaRaet, na.rm = TRUE), .groups = 'drop')
head(averaged_out)
```

```
# A tibble: 6 × 2
        cdsCode averageHSDiplomaRate
        <int64>                <dbl>
1 1000000000000                78.2
2 1100170000000                50.8
3 1100170112607                89.6
4 1100170130401                 7.85
5 1100170130419                22.7
6 1100170130625                65.6
```

```
dfs <- list(df1, df2, df3, df6, df7,averaged_out) #exlucding 9 and 10

final_df <- reduce(dfs, function(x, y) full_join(x, y, by = "cdsCode"))
head(final_df)
```

```
        cdsCode perf_1 perf_2 perf_3 perf_6 perf_7 averageHSDiplomaRate
1 1100170000000    Met    Met    Met    Met    Met             50.80897
2 1100170112607    Met    Met    Met    Met    Met             89.63333
3 1100170123968    Met    Met    Met    Met    Met                   NA
4 1100170124172    Met    Met    Met    Met    Met                   NA
5 1100170125567    Met    Met    Met    Met    Met                   NA
6 1100170130625    Met    Met    Met    Met    Met             65.56667
```

```
colSums(is.na(final_df))
```

```
             cdsCode               perf_1               perf_2 
                   0                 1851                 1851 
              perf_3               perf_6               perf_7 
                1851                 1851                 1851 
averageHSDiplomaRate 
                1199 
```

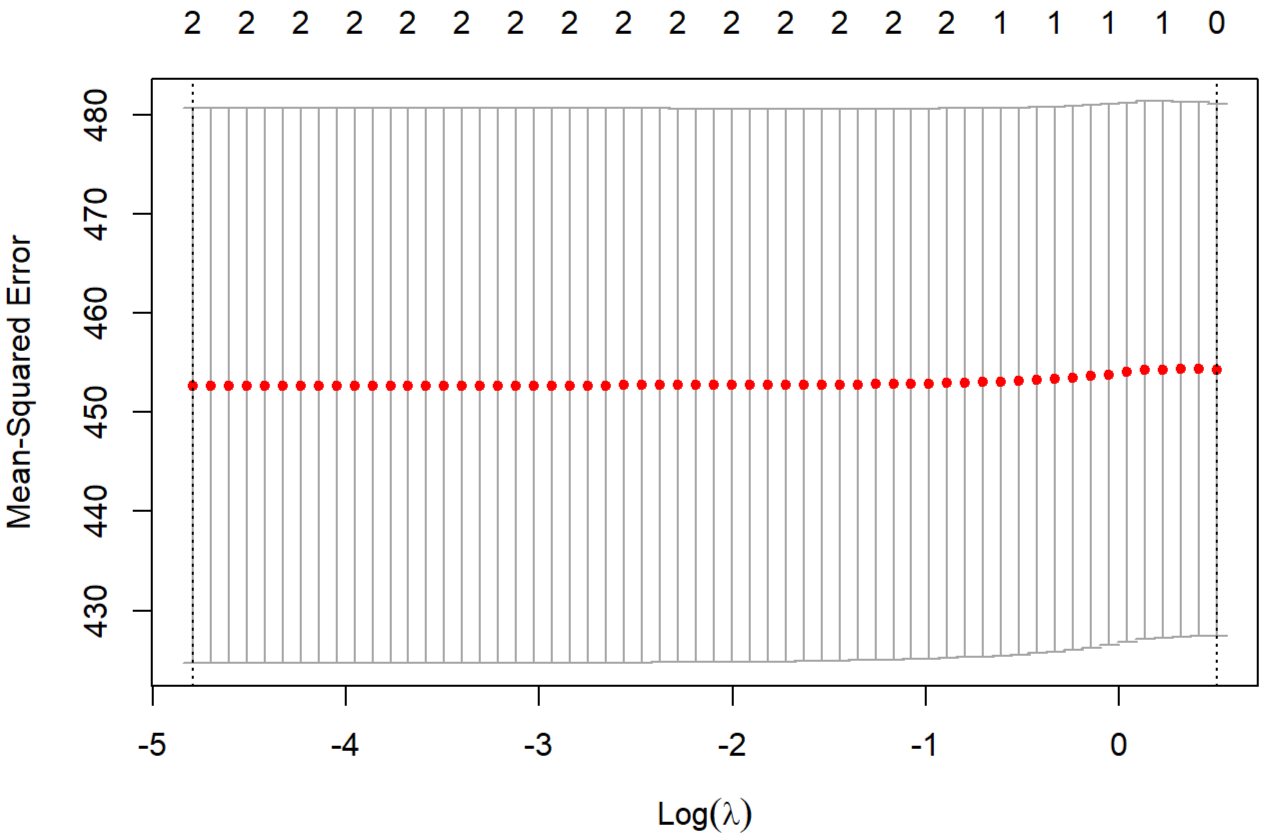
```
nrow(final_df)
```

```
[1] 4113
```

```
# for priorities 9 and 10, 4055/4113 is NA
```

Here I did some joining of the priorities so it could be easier to look at them. As stated above, for priorities 9 and 10, 4055/4113 is NA. So for my model building, i just ommitted those for now. also, there are a good amount of N/A for the other columns, even the outcome. Speaking of the outcome, I also chose to average the HSDiplomaRate, which was originally reported by ethnic category. Since the priorities are not grouped this way, I felt this was necessary for analysis.

```
# general elastic code
filtered_df <- final_df %>%
  select(where(~ n_distinct(na.omit(.)) > 1)) %>%   # Remove constant columns
  mutate(across(starts_with("perf_"), as.factor)) %>%
  drop_na()                                         # Drop rows with NA in *any* column
#nrow(filtered_df)
#colSums(is.na(filtered_df))
x <- model.matrix(averageHSDiplomaRate ~ . - cdsCode, data = filtered_df)[, -1]  # drop intercept
y <- filtered_df$averageHSDiplomaRate
#nrow(x) - length(y)
#length(y)
# Fit using cross-validation to find best alpha/lambda
cv_fit <- cv.glmnet(x, y, alpha = 1)  # alpha = 0.5 is Elastic Net 1 is lasso

# Plot CV error
plot(cv_fit)
```



```
# Get coefficients
coef(cv_fit, s = "lambda.min")
```

```
11 x 1 sparse Matrix of class "dgCMatrix"
                                        s1
(Intercept)                       80.35120
perf_1Not Met                    -17.12096
perf_1Not Met For Two or More Years  16.63915
perf_2Not Met                        .
perf_2Not Met For Two or More Years   .
perf_3Not Met                        .
perf_3Not Met For Two or More Years   .
perf_6Not Met                        .
perf_6Not Met For Two or More Years   .
perf_7Not Met                        .
perf_7Not Met For Two or More Years   .
```

```
head(filtered_df)
```

```
        cdsCode perf_1 perf_2 perf_3 perf_6 perf_7 averageHSDiplomaRate
1 1100170000000    Met    Met    Met    Met    Met             50.80897
2 1100170112607    Met    Met    Met    Met    Met             89.63333
```

```
3 1100170130625    Met     Met     Met     Met     Met               65.56667
4 1100170136101    Met     Met     Met     Met     Met               92.02500
5 1100170136226    Met     Met     Met     Met     Met               50.17778
6 1611190000000    Met     Met     Met     Met     Met               83.01358
```

```r
filtered_df$perf_1 <- as.factor(filtered_df$perf_1)
filtered_df$perf_2 <- as.factor(filtered_df$perf_2)
filtered_df$perf_3 <- as.factor(filtered_df$perf_3)
filtered_df$perf_6 <- as.factor(filtered_df$perf_6)
filtered_df$perf_7 <- as.factor(filtered_df$perf_7)

# Run a multivariate linear regression with all five predictors
model <- lm(averageHSDiplomaRate ~ perf_1 + perf_2 + perf_3 + perf_6 + perf_7, data = filtered_df

# View the summary of the regression
summary(model)
```

```

Call:
lm(formula = averageHSDiplomaRate ~ perf_1 + perf_2 + perf_3 + 
    perf_6 + perf_7, data = filtered_df)

Residuals:
    Min      1Q  Median      3Q     Max 
-79.009  -3.884   7.360  13.933  24.232 

Coefficients: (8 not defined because of singularities)
                                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)                           80.352      0.655 122.679   <2e-16 ***
perf_1Not Met                        -17.206      6.750  -2.549   0.0109 *  
perf_1Not Met For Two or More Years   16.908     21.254   0.796   0.4265    
perf_2Not Met                             NA         NA      NA       NA    
perf_2Not Met For Two or More Years       NA         NA      NA       NA    
perf_3Not Met                             NA         NA      NA       NA    
perf_3Not Met For Two or More Years       NA         NA      NA       NA    
perf_6Not Met                             NA         NA      NA       NA    
perf_6Not Met For Two or More Years       NA         NA      NA       NA    
perf_7Not Met                             NA         NA      NA       NA    
perf_7Not Met For Two or More Years       NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.24 on 1060 degrees of freedom
Multiple R-squared:  0.006694,  Adjusted R-squared:  0.00482 
F-statistic: 3.572 on 2 and 1060 DF,  p-value: 0.02845
```

Okay this section is kind of a mess, but i decided to leave it all in. Basically I was trying the Elastic Net Regression, which worked, but was not applicable in this context. The first plot is supposed to have a U shape with some minima which represents the best lambda to use for the regularization term, but its kind of just a straight line. I then tried a multivariate linear regression with the priorities, and found that the correlation between priority one is significantly influencing AverageHSDiplomaRate. However, I also found that the priorities are all perfectly correlated. A little fishy, but I haven't really thought about it too hard. Anyway, these arent the visualizations I had hoped to create but I hope this exploration will help us in the future.

# Visualizations of the Public Schools Dataset

Here we will take a look at various visualizations to get a better feel for what the data looks like in the Public Schools Dataset.

```
library(ggplot2)
library(sf)
```

```
Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(tigris)
```

```
To enable caching of data, set `options(tigris_use_cache = TRUE)`
in your R script or .Rprofile.
```

```
# Load California state boundary (set cb = TRUE for simplified version)
california <- st_read("C:/Users/Deek/Downloads/cb_2021_us_state_20m/cb_2021_us_state_20m.shp") %>
  filter(STUSPS == "CA") %>%
  st_transform(crs = 4326)  # Ensure same CRS as your point data
```

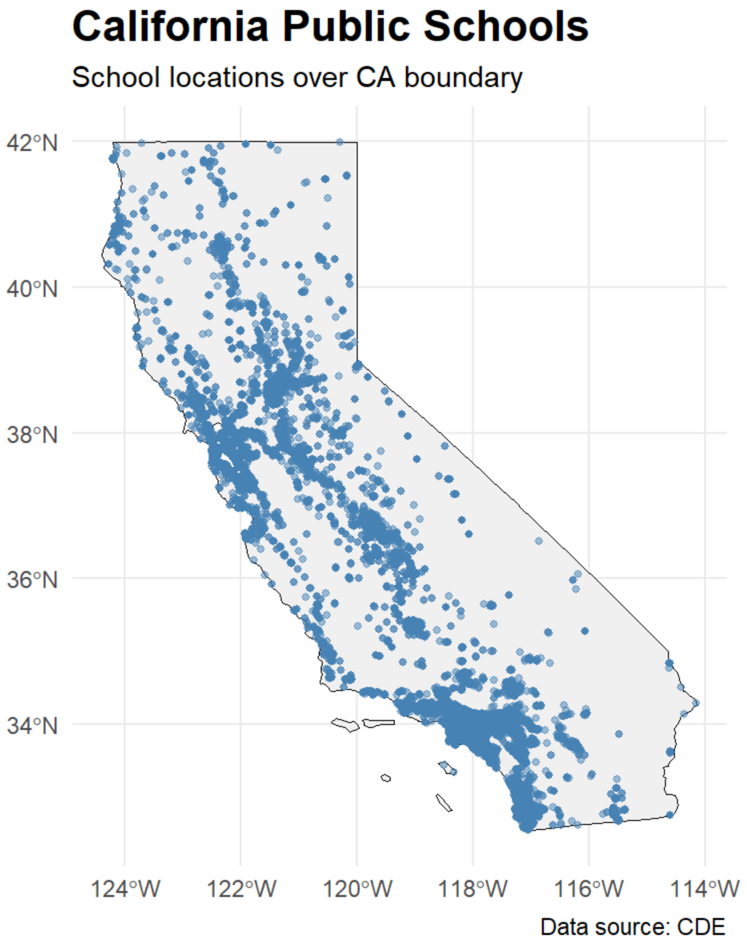
```
Reading layer `cb_2021_us_state_20m' from data source
  `C:\Users\Deek\Downloads\cb_2021_us_state_20m\cb_2021_us_state_20m.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 52 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -179.1743 ymin: 17.91377 xmax: 179.7739 ymax: 71.35256
Geodetic CRS:  NAD83
```

```
publicschools<- dbGetQuery(con, "
  SELECT * FROM PublicSchools;
")
```

```
Warning: Column `NCESDist`: mixed type, first seen values of type integer,
coercing other values of type string

Warning: Column `NCESSchool`: mixed type, first seen values of type string,
coercing other values of type integer

Warning: Column `CharterNum`: mixed type, first seen values of type string,
coercing other values of type integer
```

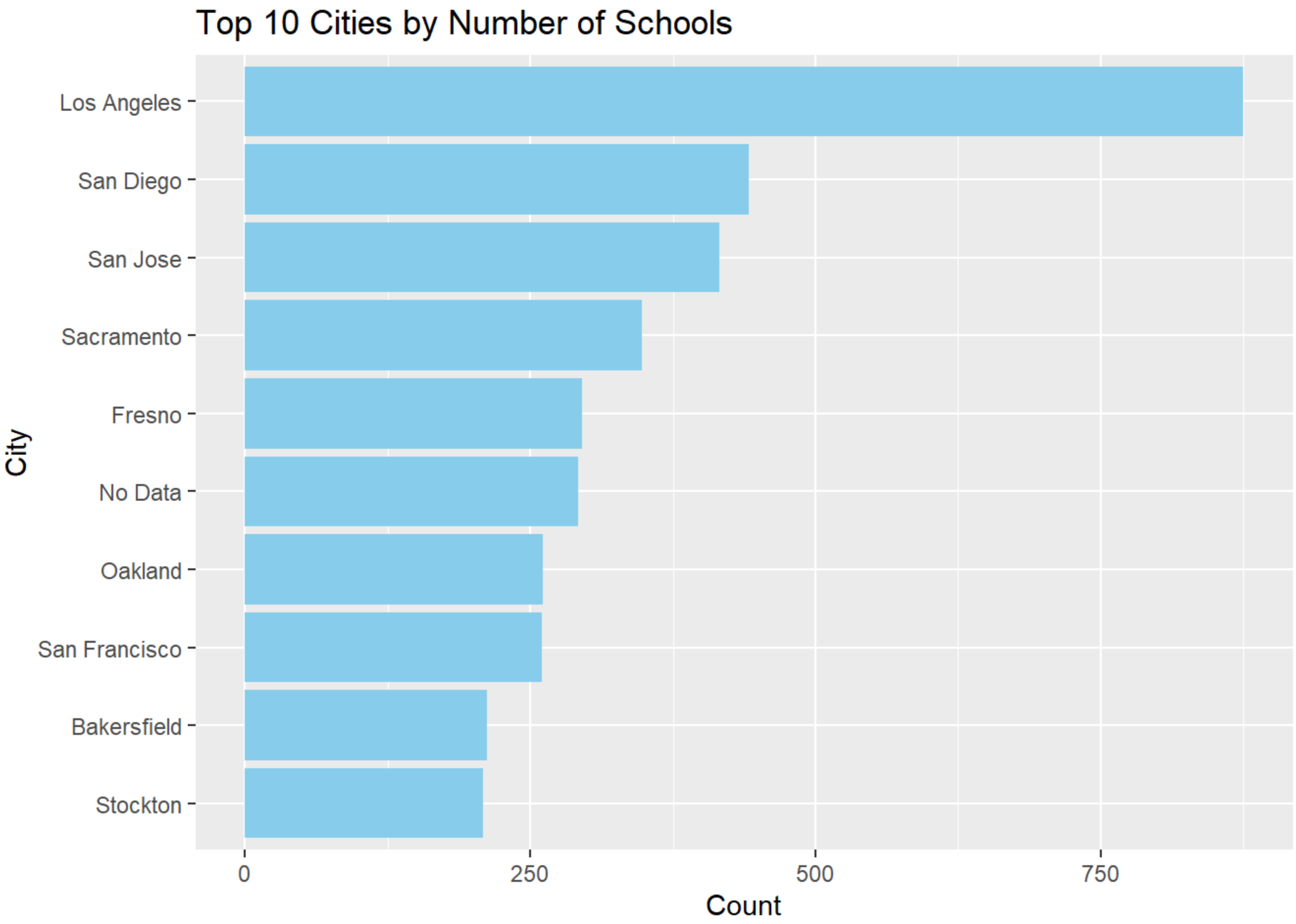
```
Warning: Column `SOC`: mixed type, first seen values of type string, coercing
other values of type integer

Warning: Column `FederalDFCDistrictID`: mixed type, first seen values of type
string, coercing other values of type integer

Warning: Column `Latitude`: mixed type, first seen values of type real,
coercing other values of type string

Warning: Column `Longitude`: mixed type, first seen values of type real,
coercing other values of type string
```

```r
# Convert school coordinates to an sf object
schools_sf <- publicschools %>%
  filter(!is.na(Longitude), !is.na(Latitude),
         Longitude != 0, Latitude != 0) %>%
  st_as_sf(coords = c("Longitude", "Latitude"), crs = 4326)

ggplot() +
  geom_sf(data = california, fill = "gray95", color = "black") +
  geom_sf(data = schools_sf, color = "steelblue", alpha = 0.5, size = 1) +
  theme_minimal() +
  labs(title = "California Public Schools",
       subtitle = "School locations over CA boundary",
       caption = "Data source: CDE") +
  theme(plot.title = element_text(size = 16, face = "bold"))
```

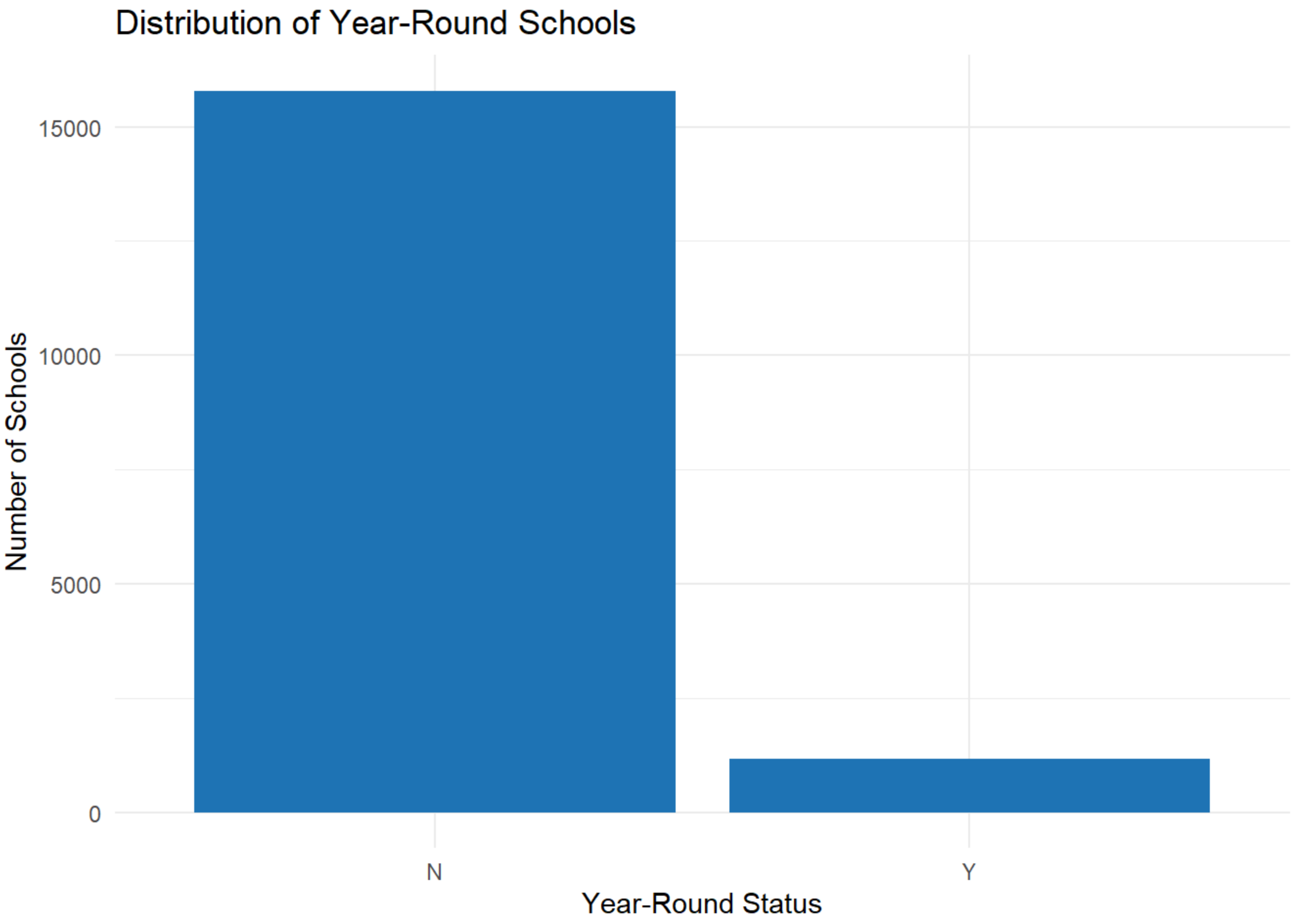


```
publicschools %>%
  count(City, sort = TRUE) %>%
  top_n(10) %>%
  ggplot(aes(x = reorder(City, n), y = n)) +
  geom_col(fill = "skyblue") +
  coord_flip() +
  labs(title = "Top 10 Cities by Number of Schools", x = "City", y = "Count")
```

```
Selecting by n
```

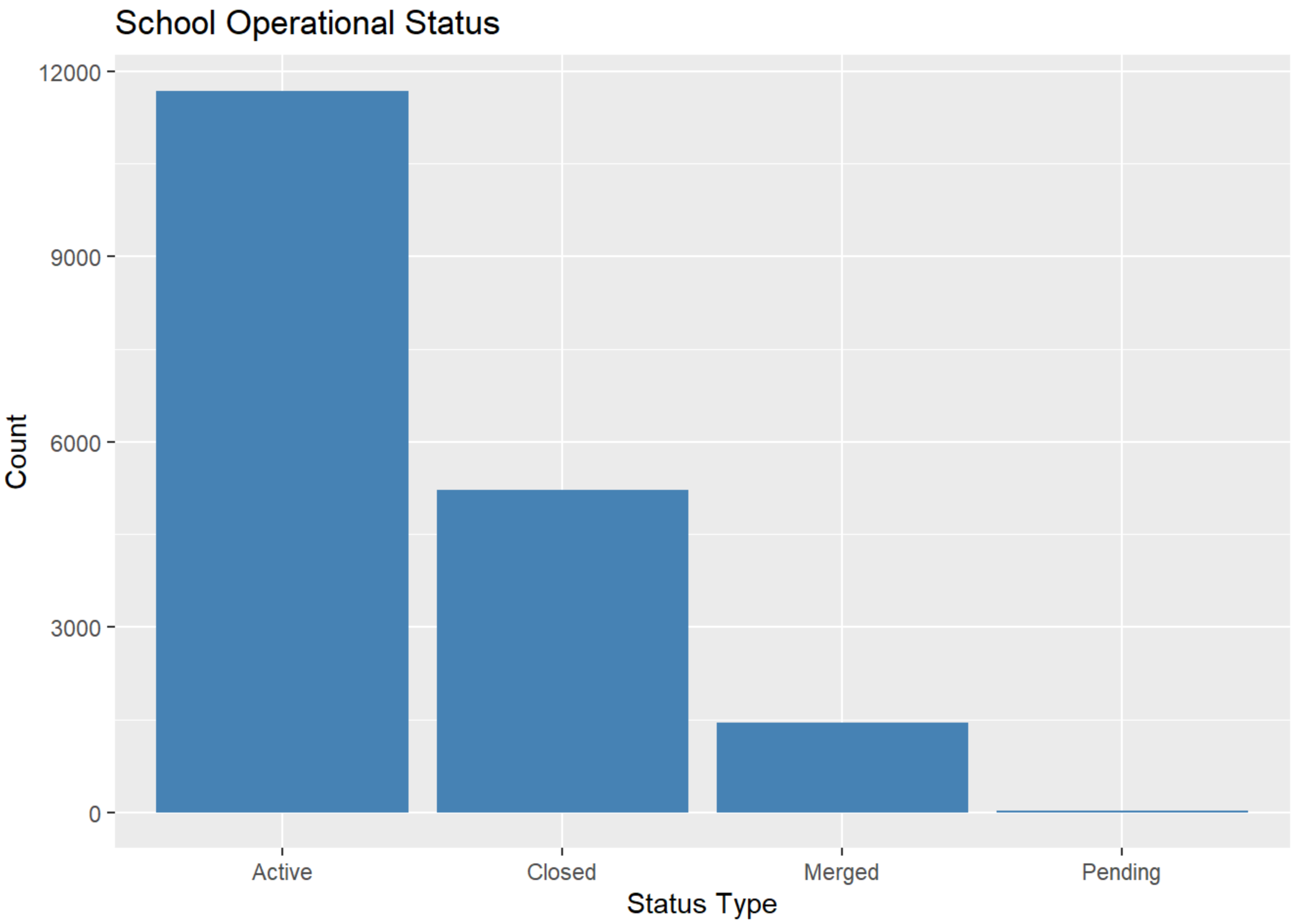


```r
df_yearround <- publicschools %>%
  filter(!is.na(YearRound), YearRound != "No Data")

# Plot
ggplot(df_yearround, aes(x = YearRound)) +
  geom_bar(fill = "#1f77b4") +
  labs(title = "Distribution of Year-Round Schools",
       x = "Year-Round Status",
       y = "Number of Schools") +
  theme_minimal()
```

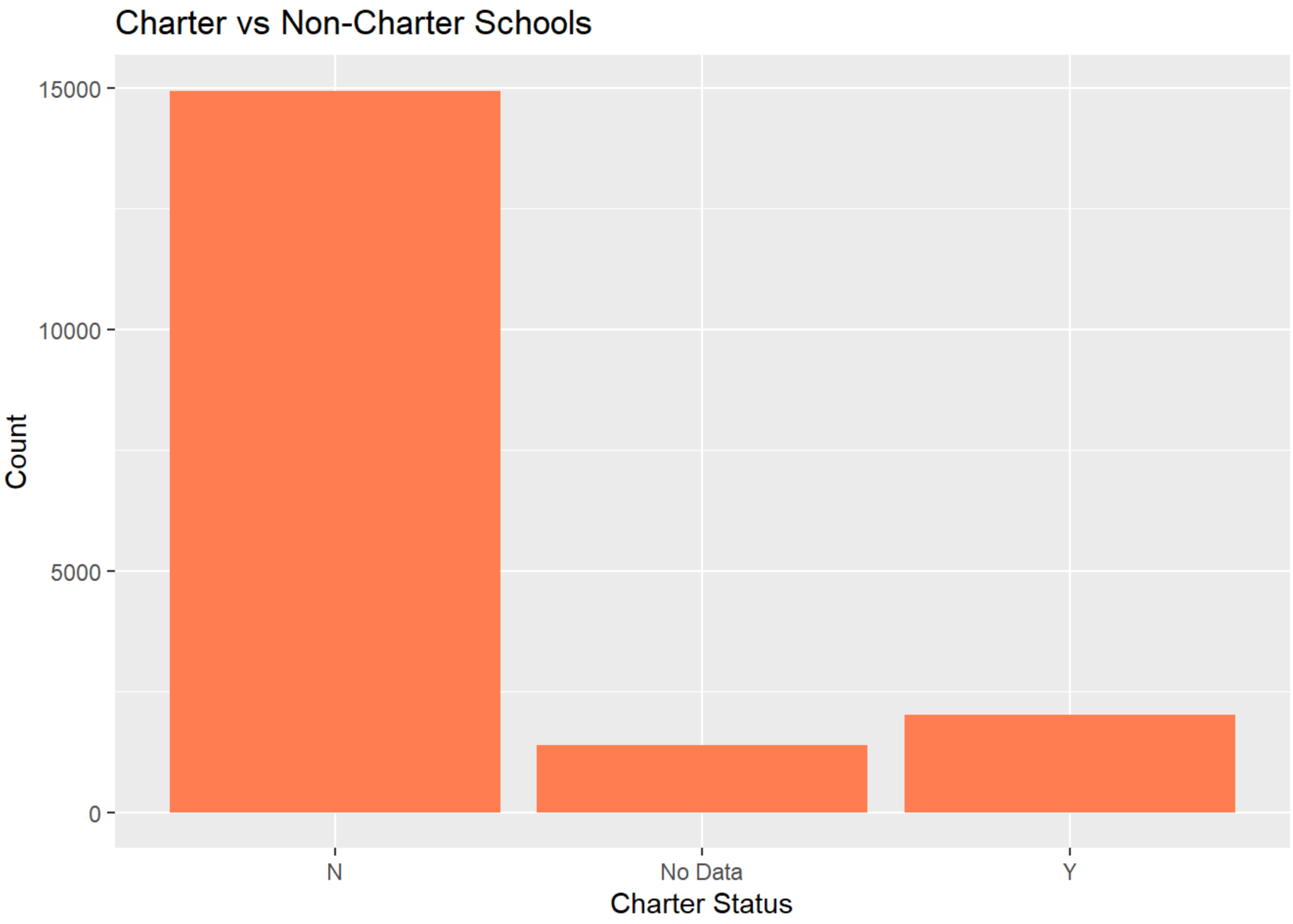


Distribution of Year-Round Schools

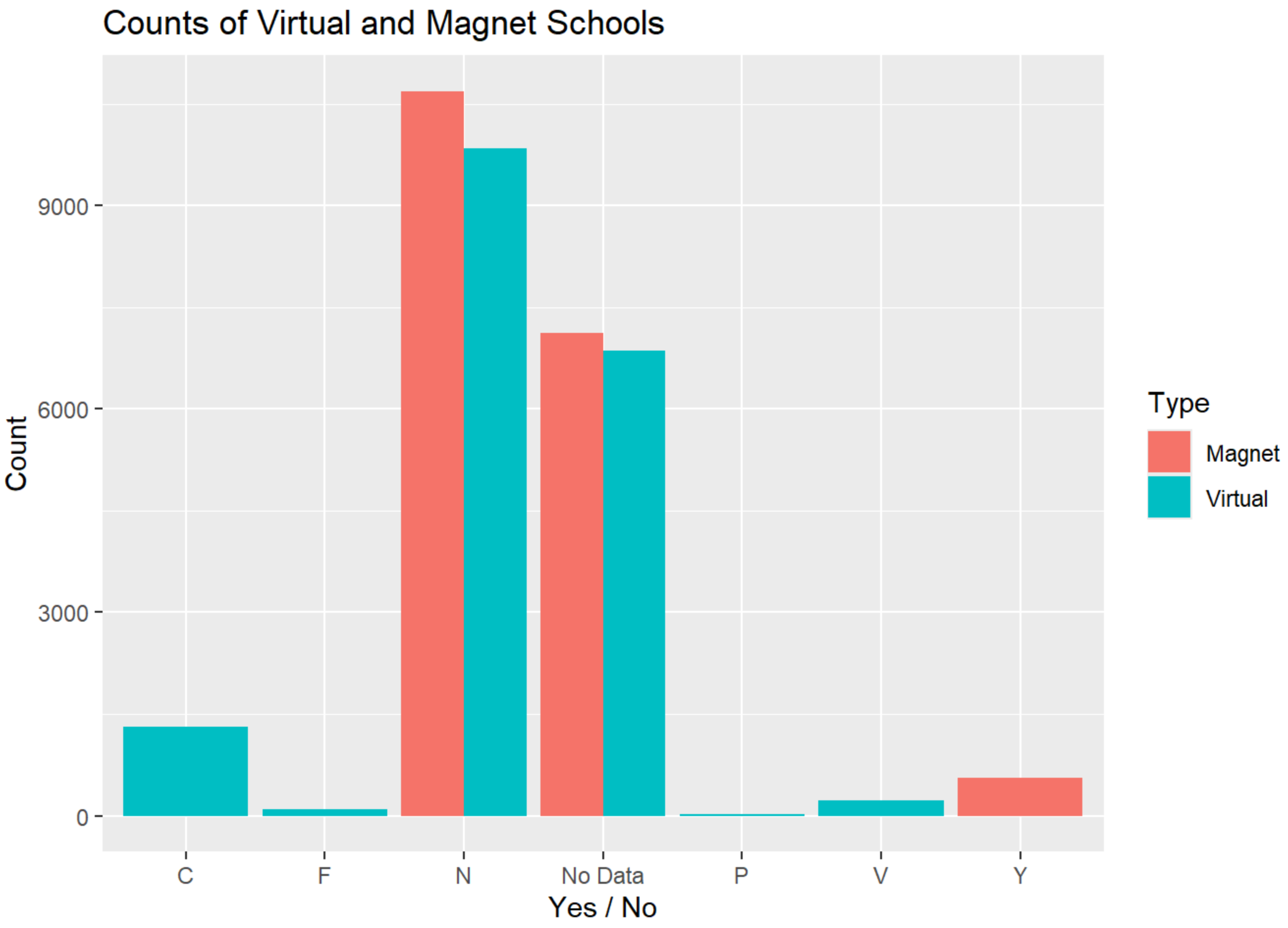
```
ggplot(publicschools, aes(x = StatusType)) +
  geom_bar(fill = "steelblue") +
  labs(title = "School Operational Status", x = "Status Type", y = "Count")
```



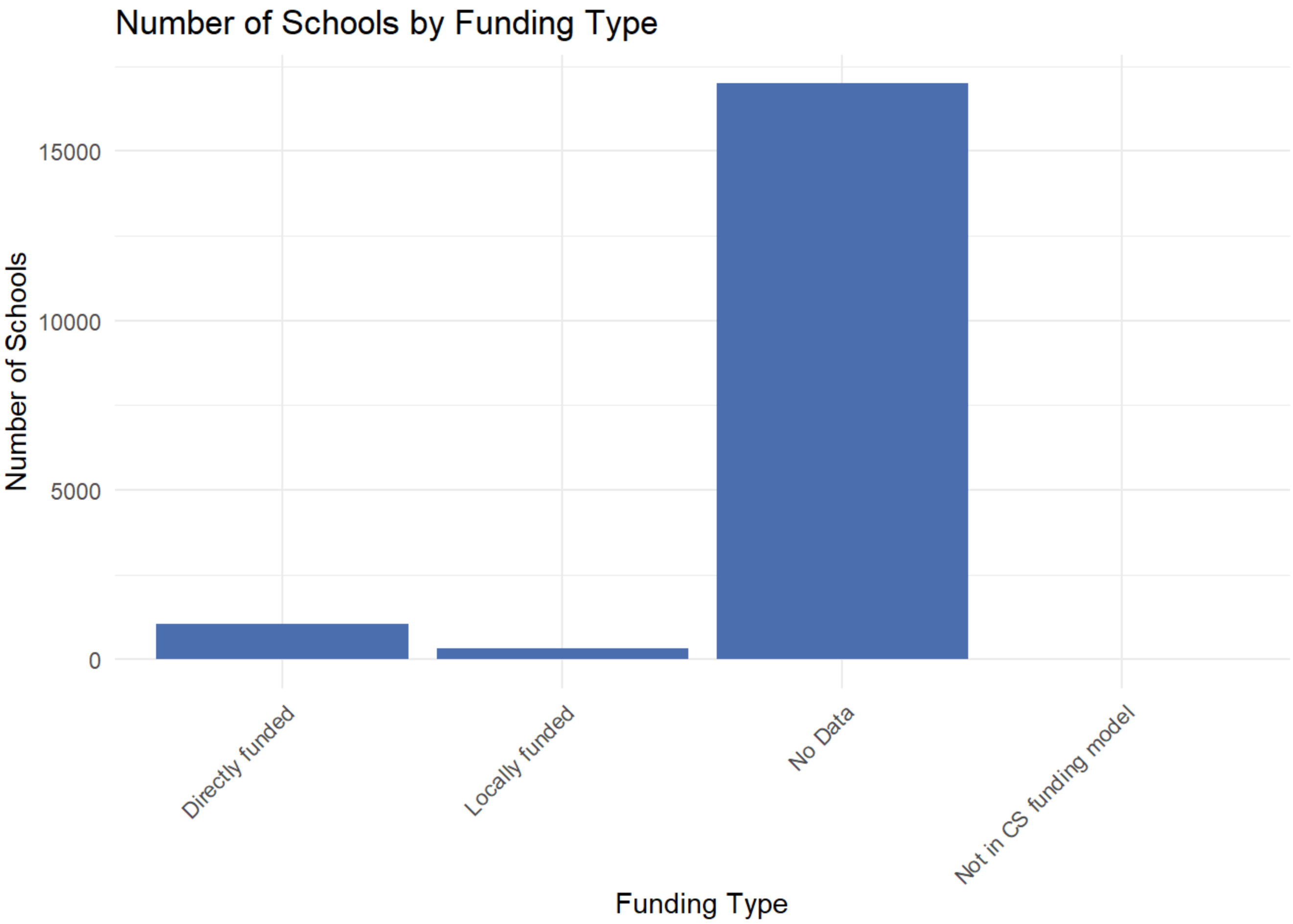
```
ggplot(publicschools, aes(x = Charter)) +
  geom_bar(fill = "coral") +
  labs(title = "Charter vs Non-Charter Schools", x = "Charter Status", y = "Count")
```



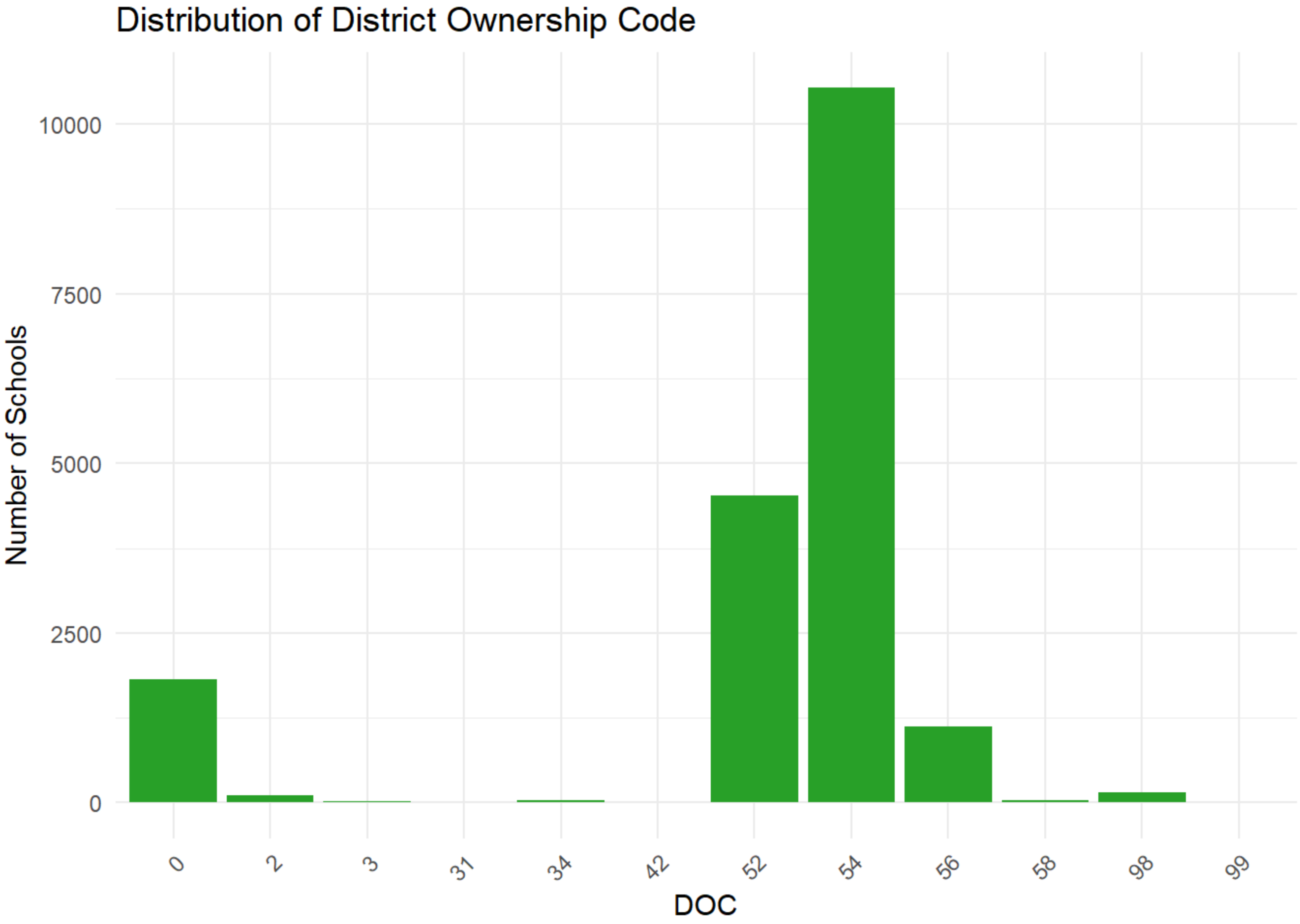
```
publicschools %>%
  pivot_longer(cols = c(Virtual, Magnet), names_to = "Type", values_to = "Value") %>%
  ggplot(aes(x = Value, fill = Type)) +
  geom_bar(position = "dodge") +
  labs(title = "Counts of Virtual and Magnet Schools", x = "Yes / No", y = "Count")
```



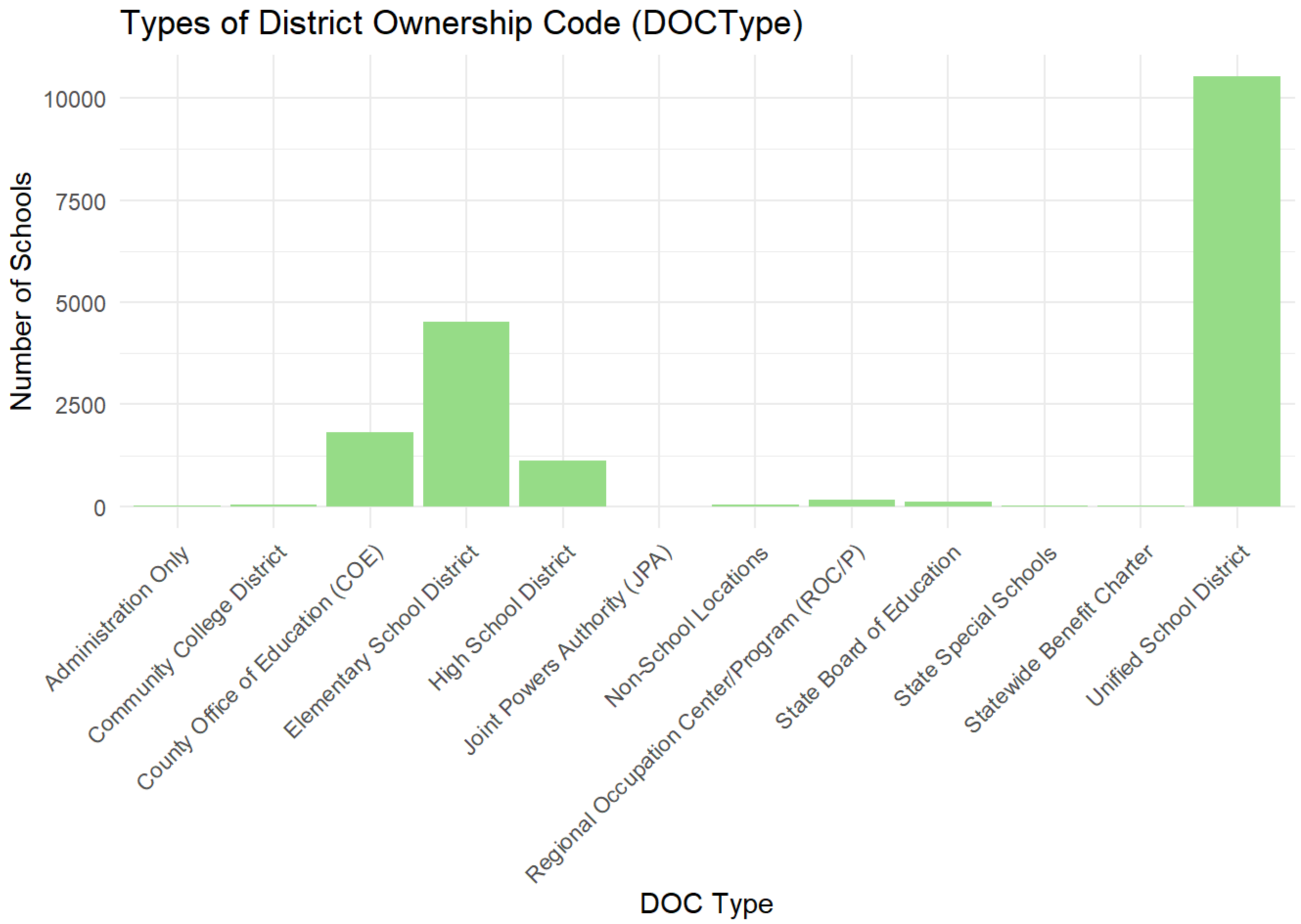
```
df_clean <- publicschools %>%
  filter(!is.na(FundingType))

# Basic bar plot
ggplot(df_clean, aes(x = FundingType)) +
  geom_bar(fill = "#4C72B0") +
  labs(
    title = "Number of Schools by Funding Type",
    x = "Funding Type",
    y = "Number of Schools"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
publicschools %>%
  filter(!is.na(DOC), DOC != "No Data") %>%
  ggplot(aes(x = factor(DOC))) +  # Convert DOC to factor
  geom_bar(fill = "#2ca02c") +
  labs(title = "Distribution of District Ownership Code",
       x = "DOC",
       y = "Number of Schools") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate labels if needed
```

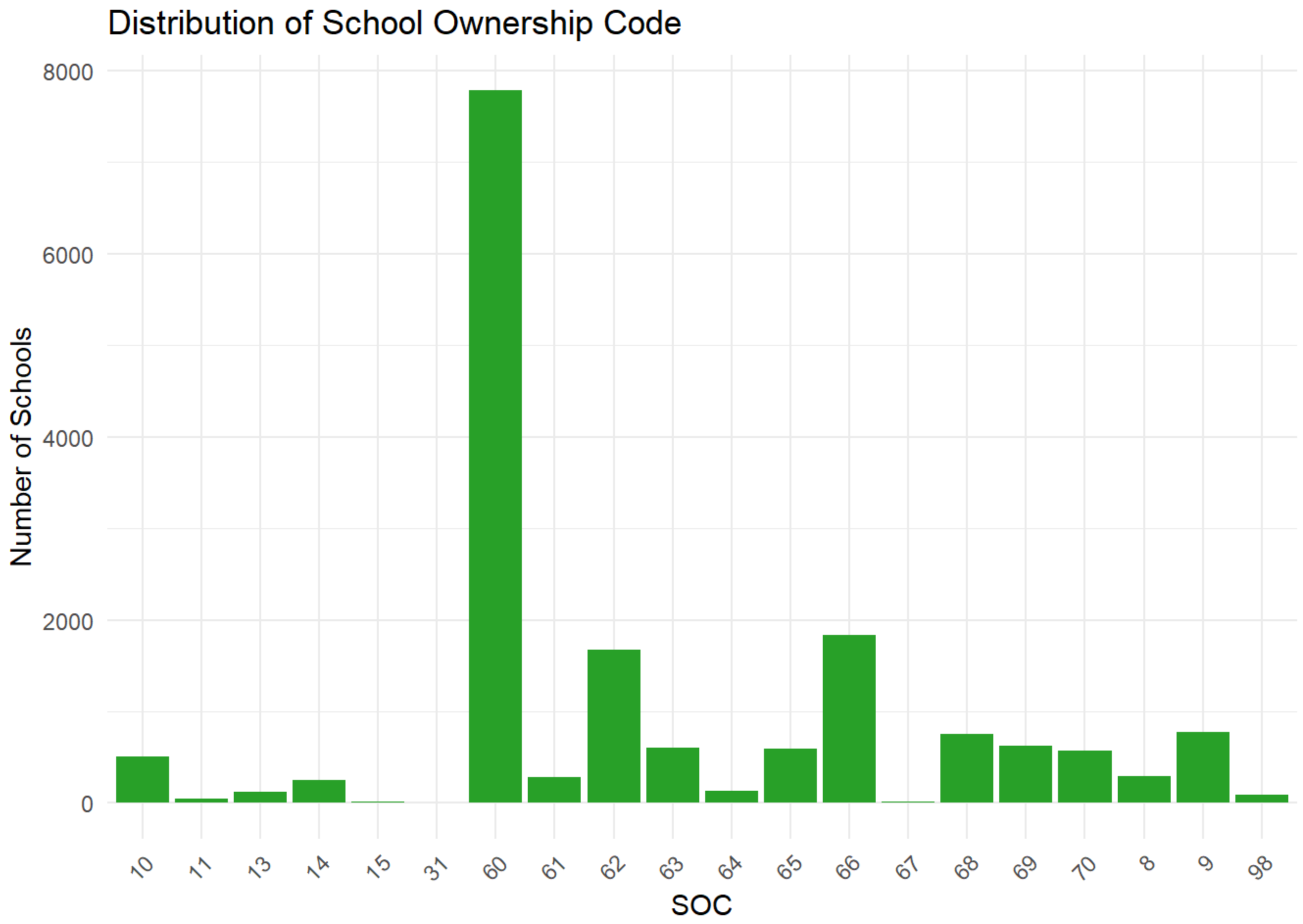


```
publicschools %>%
  filter(!is.na(DOCType), DOCType != "No Data") %>%
  ggplot(aes(x = factor(DOCType))) +  # Use factor for sorting
  geom_bar(fill = "#98df8a") +
  labs(title = "Types of District Ownership Code (DOCType)",
       x = "DOC Type",
       y = "Number of Schools") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate labels if needed
```

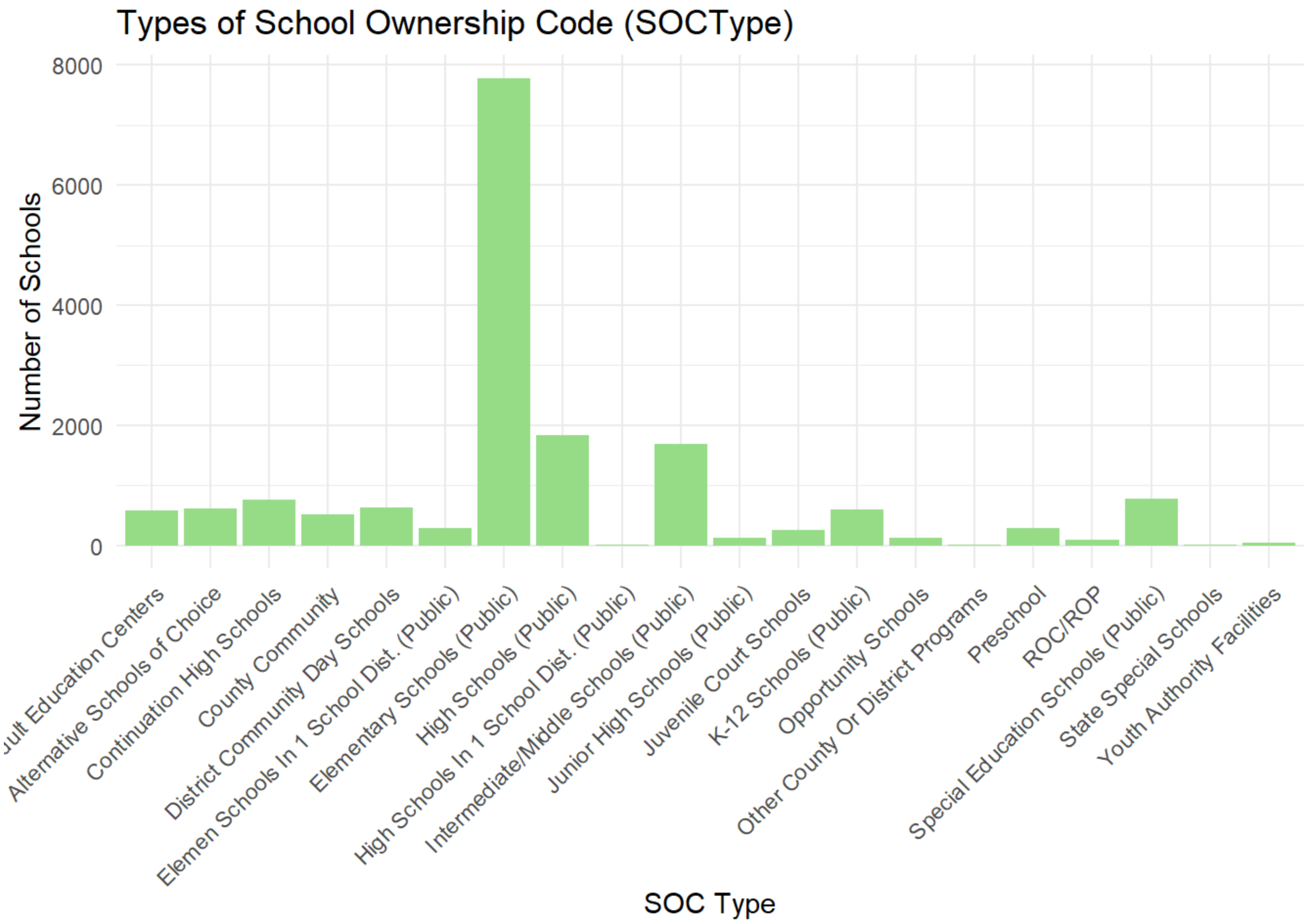


Interesting Note, these graphs should be the same, but it appears some of the codes entered are Typos, seeing as 58 and 99 are invalid codes yet they appear in the graph, as does 42 possibly.

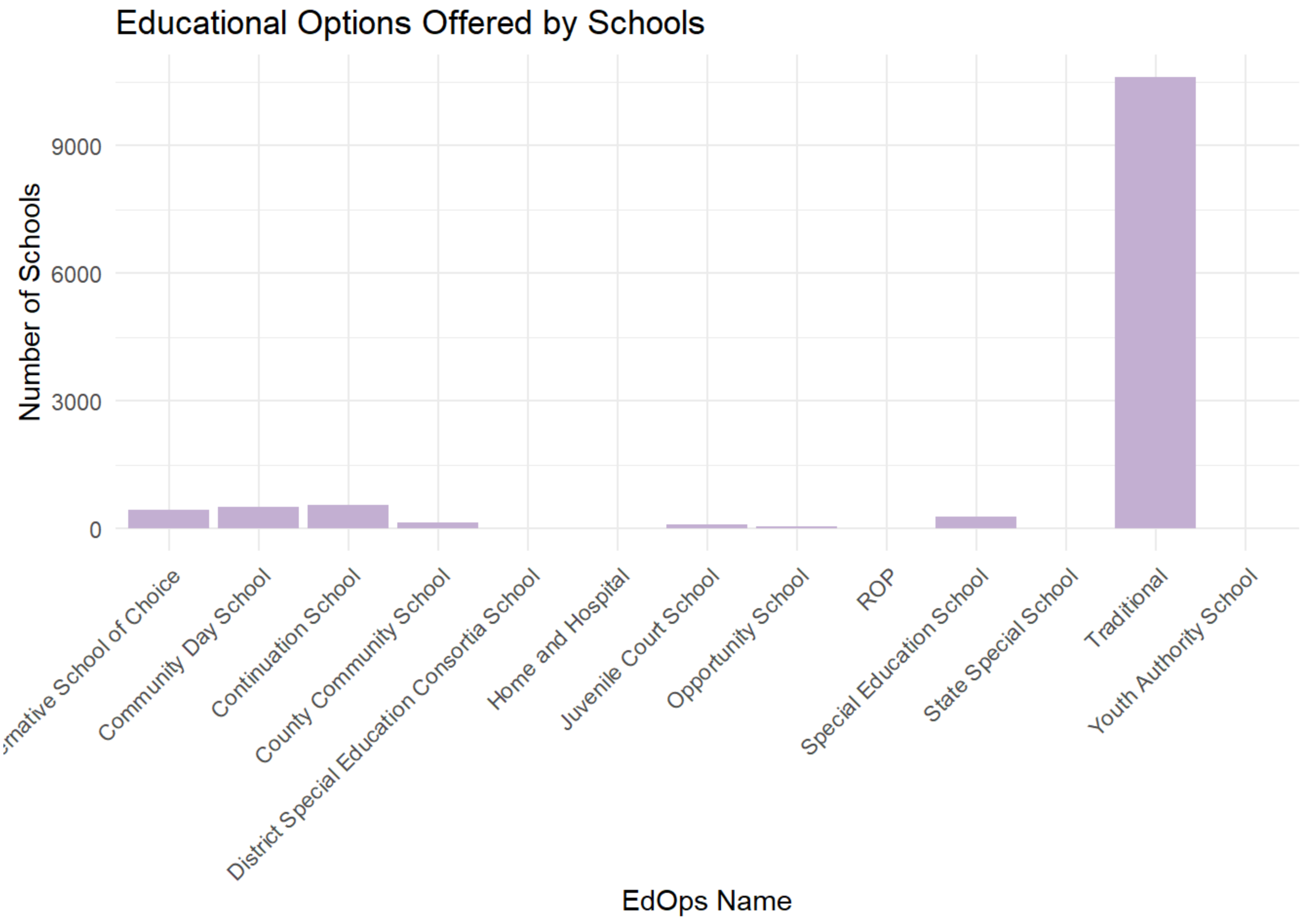
```r
publicschools %>%
  filter(!is.na(SOC), SOC != "No Data") %>%
  ggplot(aes(x = factor(SOC))) +  # Convert DOC to factor
  geom_bar(fill = "#2ca02c") +
  labs(title = "Distribution of School Ownership Code",
       x = "SOC",
       y = "Number of Schools") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate labels if needed
```



```r
publicschools %>%
  filter(!is.na(SOCType), SOCType != "No Data") %>%
  ggplot(aes(x = factor(SOCType))) +  # Use factor for sorting
  geom_bar(fill = "#98df8a") +
  labs(title = "Types of School Ownership Code (SOCType)",
       x = "SOC Type",
       y = "Number of Schools") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate labels if needed
```



```
publicschools %>%
  filter(!is.na(EdOpsName), EdOpsName != "No Data") %>%
  ggplot(aes(x = factor(EdOpsName))) +
  geom_bar(fill = "#c5b0d5") +
  labs(title = "Educational Options Offered by Schools",
       x = "EdOps Name",
       y = "Number of Schools") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Educational Options Offered by Schools
Number of Schools
9000
6000
3000
0
rnative School of Choice
Community Day School
Continuation School
County Community School
District Special Education Consortia School
Home and Hospital
Juvenile Court School
Opportunity School
ROP
Special Education School
State Special School
Traditional
Youth Authority School
EdOps Name