Exploratory Data Analysis

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
vote_prec_clean <- read_csv("data/vote_prec_clean.csv")
```

```
Rows: 48337 Columns: 76
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (49): fips, svprec, svprec_key, election, geo_type, assaip01,
assdem01, ...
dbl (27): county, addist, cddist, sddist, bedist, totreg, demreg, repreg,
ai...

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.
```

Question 1

What is the range and distribution of total votes cast per precinct?

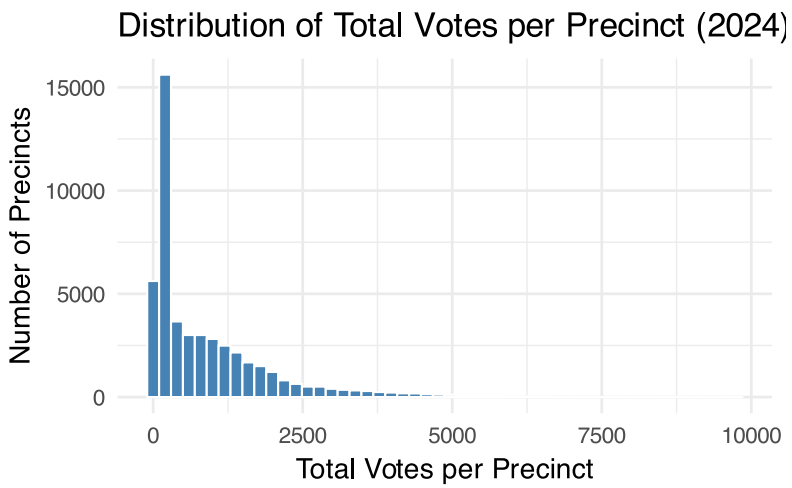
Answer 1

```
dont_sum <-
c("fips","county_fips","year","district","geoid","precinct_code","total_votes")
vote_cols <- vote_prec_clean |>
```

```
  select(where(is.numeric)) |>
  select(-any_of(dont_sum)) |>
  names()

vote_prec_clean <- vote_prec_clean |>
  mutate(total_votes = rowSums(across(all_of(vote_cols)), na.rm = TRUE))

vote_prec_clean_2 <- vote_prec_clean |>
    select(svprec, total_votes)

summary(vote_prec_clean_2$total_votes)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1     147     454    2724    1296 9530802 
```

```
ggplot(vote_prec_clean_2 |>
     filter(total_votes < 10000), aes(x = total_votes)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white") +
  labs(
    title = "Distribution of Total Votes per Precinct (2024)",
    x = "Total Votes per Precinct",
    y = "Number of Precincts"
  ) +
  theme_minimal(base_size = 14)
```



Question 2

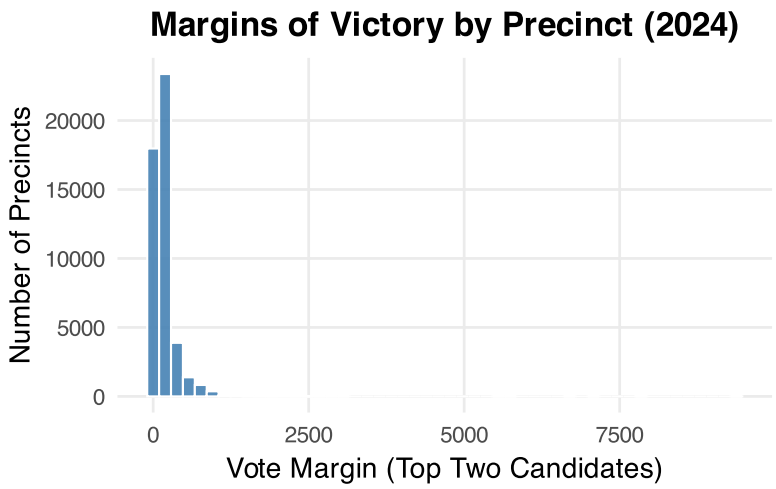
Which precincts had the closest races between the top two candidates?

Answer 2

```
closest_races <- vote_prec_clean |>
    pivot_longer(cols = where(is.numeric), names_to = "candidate", values_to =
"votes") |>
    group_by(svprec) |>
    arrange(desc(votes), .by_group = TRUE) |>
    slice_head(n = 2) |>
    summarise(margin = diff(votes), total = sum(votes)) |>
    arrange(margin)

closest_races_clean <- closest_races |>
    mutate(margin = abs(margin)) |>
    filter(!is.na(margin), margin > 0, margin < 10000)

head(closest_races, 10)
```

```
# A tibble: 10 × 3
   svprec     margin    total
   <chr>       <dbl>    <dbl>
 1 BE03_TOT -3793999 15267605
 2 BE02_TOT  -683070  2604368
 3 CNTYTOT   -683070  2604368
 4 SOVTOT    -683070  2605678
 5 SD24_TOT  -514503  1846575
 6 SD37_TOT  -483566  1697518
 7 SD40_TOT  -461538  1637152
 8 SD39_TOT  -453014  1661066
 9 SD25_TOT  -415750  1587264
10 SD11_TOT  -412269  1456799
```

```
ggplot(closest_races_clean, aes(x = margin)) +
  geom_histogram(bins = 50, fill = "steelblue", color = "white", alpha = 0.9)
+
  labs(
    title = "Margins of Victory by Precinct (2024)",
    x = "Vote Margin (Top Two Candidates)",
    y = "Number of Precincts"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    panel.grid.minor = element_blank()
  )
```



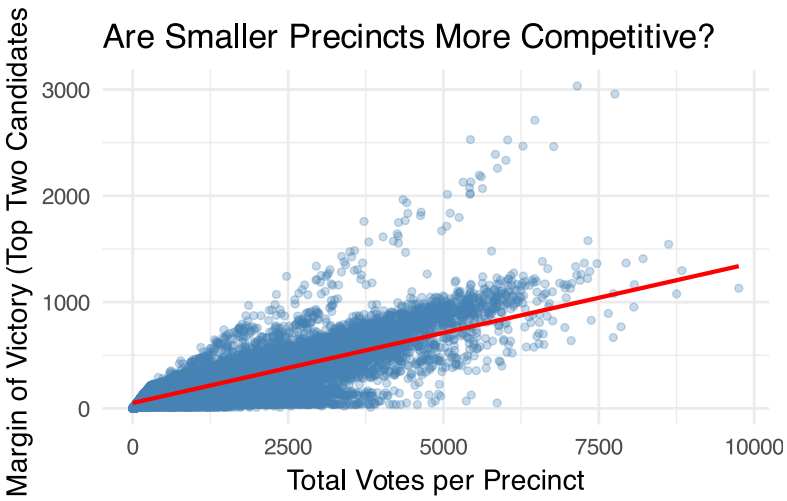
Question 3

Are smaller precincts more competitive?

Answer 3

```
competition <- vote_prec_clean |>
  select(svprec, total_votes) |>
  left_join(closest_races, by = "svprec")

competition <- competition |>
  mutate(margin = abs(margin)) |>
  filter(total_votes > 0, total_votes < 10000, margin < 10000)

ggplot(competition, aes(x = total_votes, y = margin)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(
    title = "Are Smaller Precincts More Competitive?",
    x = "Total Votes per Precinct",
    y = "Margin of Victory (Top Two Candidates)"
  ) +
  theme_minimal(base_size = 14)
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
cor_result <- cor(competition$total_votes, competition$margin, use =
"complete.obs")
cor_result
```

```
[1] 0.811316
```