# Analyzing the 2016 US Presidential Election

## Introduction

We analyze returns from the 2012 and 2016 elections in order to understand the social and demographic trends that may have contributed to Donald Trump's victory in 2016.

We will first examine how Republican vote share at the county level has changed from 2012 to 2016. Then, we will look at four variables that were prominent in the discourse around the election – race, education, unemployment, and immigration – to see how well they predict GOP electoral gains at the county level.

We will be working with the data set `uselection.csv` which has one observation per county and contains the following variables (note that some counties including those of Alaska are missing from the data):

| Name | Description |
|---|---|
| `FIPS` | FIPS code (unique county identifier) |
| `state` | State abbreviation |
| `county` | County name |
| `votes_dem_12` | Number of votes cast for Democratic candidate, 2012 election |
| `votes_gop_12` | Number of votes cast for Republican candidate, 2012 election |
| `votes_total_12` | Total number of votes cast in 2012 election |
| `votes_dem_16` | Number of votes cast for Democratic candidate, 2016 election |
| `votes_gop_16` | Number of votes cast for Republican candidate, 2016 election |
| `votes_total_16` | Total number of votes cast in 2016 election |
| `pct_for_born15` | Percent of county's population that is "foreign born" according to the U.S. Census, meaning anyone who is not a U.S. citizen at birth (measured over 2011-2015) |
| `pct_bach_deg15` | Percent of county population holding a Bachelor's degree or above (2011-2015) |
| `pct_non_white15` | Percent of county population that is not white (2011-2015) |
| `pct_unemp12` | Percent of county population that is unemployed, BLS estimates (average, Jan-Oct 2012) |
| `pct_unemp16` | Percent of county population that is unemployed, BLS estimates (average, Jan-Oct 2016) |

### Question 1: Reading data into R

We first need to load the data into R and make it as `tibble` object. Load the `tidyverse` package, read the data using the `read_csv()` function and save it as `elec` (using `read_csv()` will automatically make `elec` a tibble).

How many counties are there included in `elec`?

### Answer 1

```
# load tidyverse
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# read data
elec <- read_csv("data/uselection.csv")
```

```
## Rows: 3112 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): state, county
## dbl (12): FIPS, votes_dem_12, votes_gop_12, votes_total_12, votes_dem_16, vo...
##
## 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.
```

```
# check the data
elec
```

```
## # A tibble: 3,112 x 14
##     FIPS state county   votes_dem_12 votes_gop_12 votes_total_12 votes_dem_16
##    <dbl> <chr> <chr>           <dbl>        <dbl>          <dbl>        <dbl>
##  1  1001 AL    Autauga          6354        17366          23909         5908
##  2  1003 AL    Baldwin         18329        65772          84988        18409
##  3  1005 AL    Barbour          5873         5539          11459         4848
##  4  1007 AL    Bibb             2200         6131           8391         1874
##  5  1009 AL    Blount           2961        20741          23980         2150
##  6  1011 AL    Bullock          4058         1250           5318         3530
##  7  1013 AL    Butler           4367         5081           9483         3716
##  8  1015 AL    Calhoun         15500        30272          46240        13197
##  9  1017 AL    Chambers         6853         7596          14562         5763
## 10  1019 AL    Cherokee         2126         7494           9761         1524
## # i 3,102 more rows
## # i 7 more variables: votes_gop_16 <dbl>, votes_total_16 <dbl>,
## #   pct_for_born15 <dbl>, pct_bach_deg15 <dbl>, pct_non_white15 <dbl>,
## #   pct_unemp12 <dbl>, pct_unemp16 <dbl>
```

There are 3112 counties in the data.

## Question 2: Preprocessing the data

Before we investigate the data, let's create some new variables: called `gop_vs_12`, `gop_vs_16`, and `gop_vs_diff`. Compute the following and add each to `elec` as a new column:

- `gop_vs_12`: compute the Republican vote share as a proportion of total votes in 2012 (Number of votes for the Republican party in the 2012 election/ Total number of votes in the 2012 election).
- `gop_vs_16`: compute the Republican vote share as a proportion of total votes in 2016 (Number of votes for the Republican party in the 2016 election/ Total number of votes in the 2016 election).

- `gop_vs_diff`: compute the *percent difference* in this Republican vote share variable from the 2012 to 2016 election (i.e., `(gop_vs_16 - gop_vs_12)/gop_vs_12 * 100`).

*Hint*: Use the `mutate()` function and the pipe operator (`|>`). Check the lecture slides for some details.

## Answer 2

```
# Create new columns, and save those to original object for future use
elec <- elec |>
  mutate(gop_vs_12 = votes_gop_12/votes_total_12,
         gop_vs_16 = votes_gop_16/votes_total_16) |>
  mutate(gop_vs_diff = (gop_vs_16 - gop_vs_12)/gop_vs_12 * 100)
```

## Question 3

Once you created the columns, print the `head` of the `elec` dataframe for *only* those three new columns (`gop_vs_12`, `gop_vs_16`, and `gop_vs_diff`). Use `knitr::kable()` to produce a nicely formatted table.

*Hint*: Use the `select()` function and the pipe operator (`|>`).

## Answer 3

```
# Print three new columns
elec |>
  head() |>
  select(gop_vs_12, gop_vs_16, gop_vs_diff) |>
  knitr::kable()
```

| gop_vs_12 | gop_vs_16 | gop_vs_diff |
|---|---|---|
| 0.7263374 | 0.7343579 | 1.1042431 |
| 0.7738975 | 0.7735147 | -0.0494602 |
| 0.4833755 | 0.5227141 | 8.1383178 |
| 0.7306638 | 0.7696616 | 5.3373152 |
| 0.8649291 | 0.8985188 | 3.8835142 |
| 0.2350508 | 0.2422889 | 3.0793789 |

## Question 4: Subsetting the data

Subset your `elec` data to just the "battleground" states: Florida (FL), North Carolina (NC), Ohio (OH), Pennsylvania (PA), New Hampshire (NH), Michigan (MI), Wisconsin (WI), Iowa (IA), Nevada (NV), Colorado (CO), and Virginia (VA). Save this subset as a new `tibble` object called `elec_battle`.

*Hint*: You may create a new vector that contains all the 2-letter abbreviations of battleground states `battlestates_abb <- c(...)`. Then, use `filter()` and `%in%` to subset the data to the battleground states with `state` column.

### Answer 4

```
# 2-letter abbreviations of battleground states
battlestates_abb <- c("FL", "NC", "OH", "PA", "NH", "MI", "WI", "IA", "NV", "CO", "VA")

# subset of data
elec_battle <- elec |>
  filter(state %in% battlestates_abb)

elec_battle
```

```
## # A tibble: 800 x 17
##     FIPS state county     votes_dem_12 votes_gop_12 votes_total_12 votes_dem_16
##    <dbl> <chr> <chr>             <dbl>        <dbl>          <dbl>        <dbl>
##  1  8001 CO    Adams             90843        66531         161495        86471
##  2  8003 CO    Alamosa            3782         2693           6671         3168
##  3  8005 CO    Arapahoe         135433       114232         254746       148365
##  4  8007 CO    Archuleta          2637         3831           6646         2489
##  5  8009 CO    Baca                462         1554           2096          278
##  6  8011 CO    Bent                778         1053           1881          581
##  7  8013 CO    Boulder          120485        48526         173207       124715
##  8  8014 CO    Broomfield        16653        14765          32191        19530
##  9  8015 CO    Chaffee            4967         4949          10217         4773
## 10  8017 CO    Cheyenne            162          858           1045          127
## # i 790 more rows
## # i 10 more variables: votes_gop_16 <dbl>, votes_total_16 <dbl>,
## #   pct_for_born15 <dbl>, pct_bach_deg15 <dbl>, pct_non_white15 <dbl>,
## #   pct_unemp12 <dbl>, pct_unemp16 <dbl>, gop_vs_12 <dbl>, gop_vs_16 <dbl>,
## #   gop_vs_diff <dbl>
```

### Question 5: State-level summarize

Now let's create a state-level summary of this subset, `elec_battle` with `summarize()`. Create a state-level average of socio-demographic variables (`pct_for_born15`, `pct_bach_deg15`, `pct_non_white15`, `pct_non_white15`, `pct_unemp12`, `pct_unemp16`) and vote share variables (`gop_vs_12`, `gop_vs_16`, `gop_vs_diff`).

*Hint*: Review `group_by()`, `select()` and `summarize()` functions.

### Answer 5

```
elec_battle |>
  group_by(state) |>
  select(pct_for_born15:gop_vs_diff) |>
  summarize(across(where(is.numeric), mean, na.rm = TRUE))
```

```
## Adding missing grouping variables: `state`
```

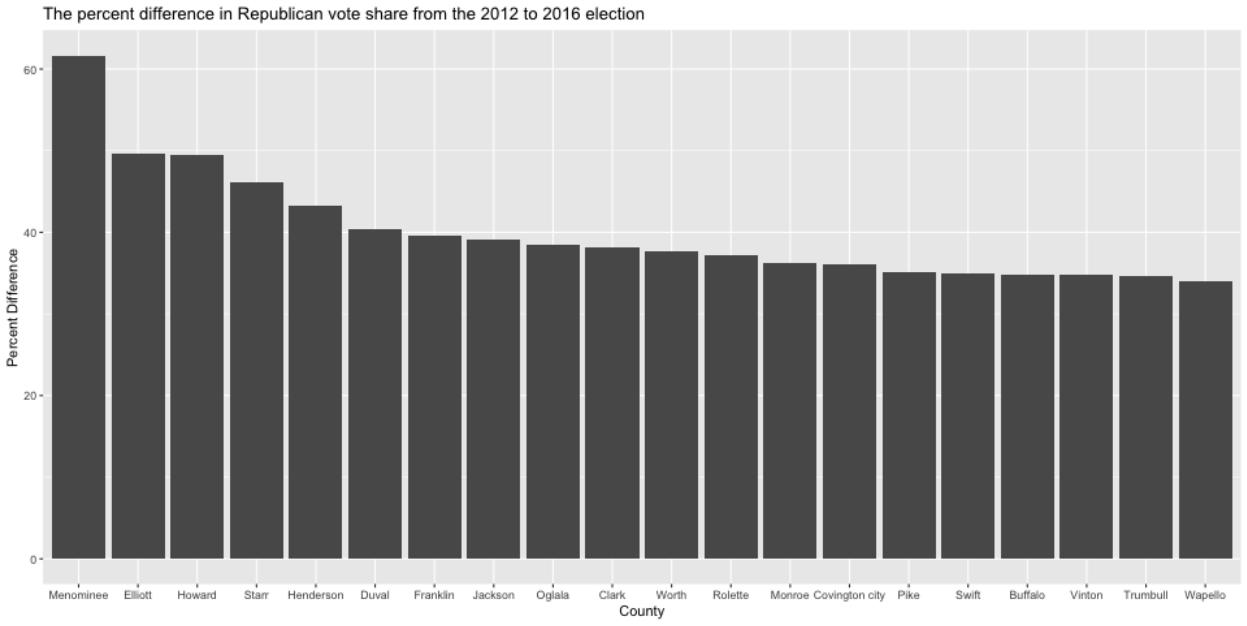
```
## Warning: There was 1 warning in `summarize()`.
## i In argument: `across(where(is.numeric), mean, na.rm = TRUE)`.
## i In group 1: `state = "CO"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
```

```
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
## # A tibble: 11 x 9
##    state pct_for_born15 pct_bach_deg15 pct_non_white15 pct_unemp12 pct_unemp16
##    <chr>          <dbl>          <dbl>           <dbl>       <dbl>       <dbl>
##  1 CO              6.43           30.0            9.65        7.20        3.11
##  2 FL              9.54           20.5           20.8         8.36        5.20
##  3 IA              2.93           20.3            4.99        4.42        3.42
##  4 MI              2.62           20.4            9.30        8.56        4.83
##  5 NC              4.93           20.3           27.7         9.65        5.36
##  6 NH              4.42           31.9            5.10        5.02        2.3
##  7 NV              8.68           17.6           15.2         9.31        5.29
##  8 OH              1.93           18.8            7.82        7.36        4.91
##  9 PA              3.30           21.6            8.94        7.57        5.68
## 10 VA              5.25           25.1           24.7         6.58        4.63
## 11 WI              2.62           21.7            8.03        6.26        3.64
## # i 3 more variables: gop_vs_12 <dbl>, gop_vs_16 <dbl>, gop_vs_diff <dbl>
```

## Question 6: Barplot

Create a barplot for the top 20 counties in terms of the difference in GOP vote share between the 2012 and 2016 elections (`gop_vs_diff`), using `elec` data. Order the bars based on the values of vote share difference. The result looks like the following:



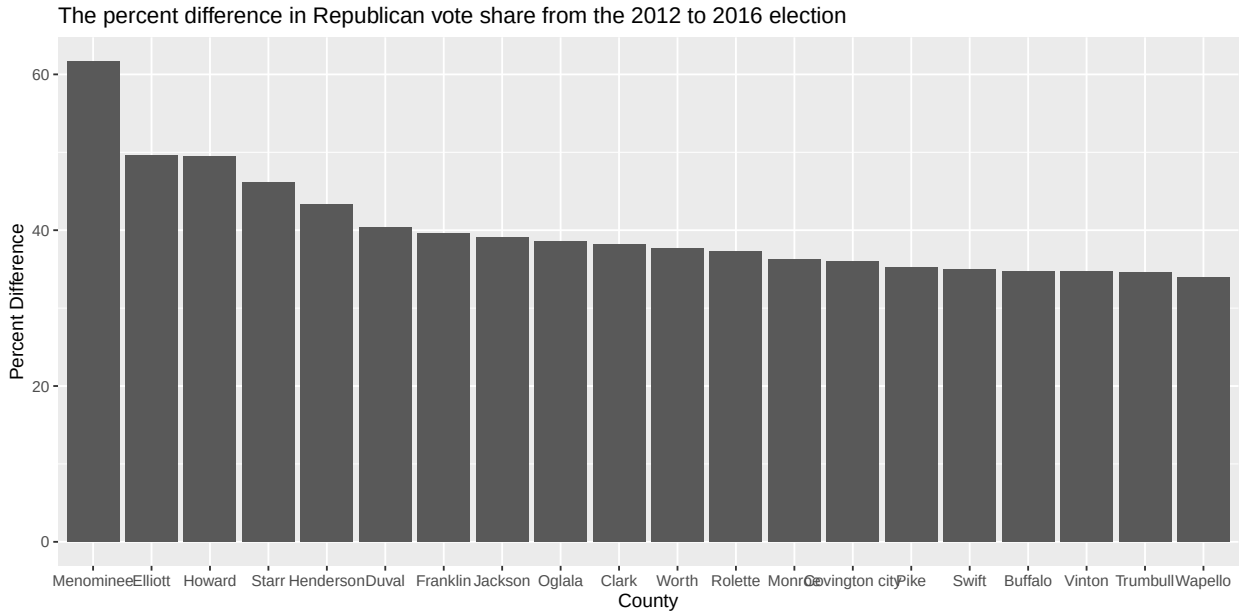

*Hint*: Sample codes using `geom_bar()`

```
# TODO: Choose either option 1 or option 2, and replace <...>
```

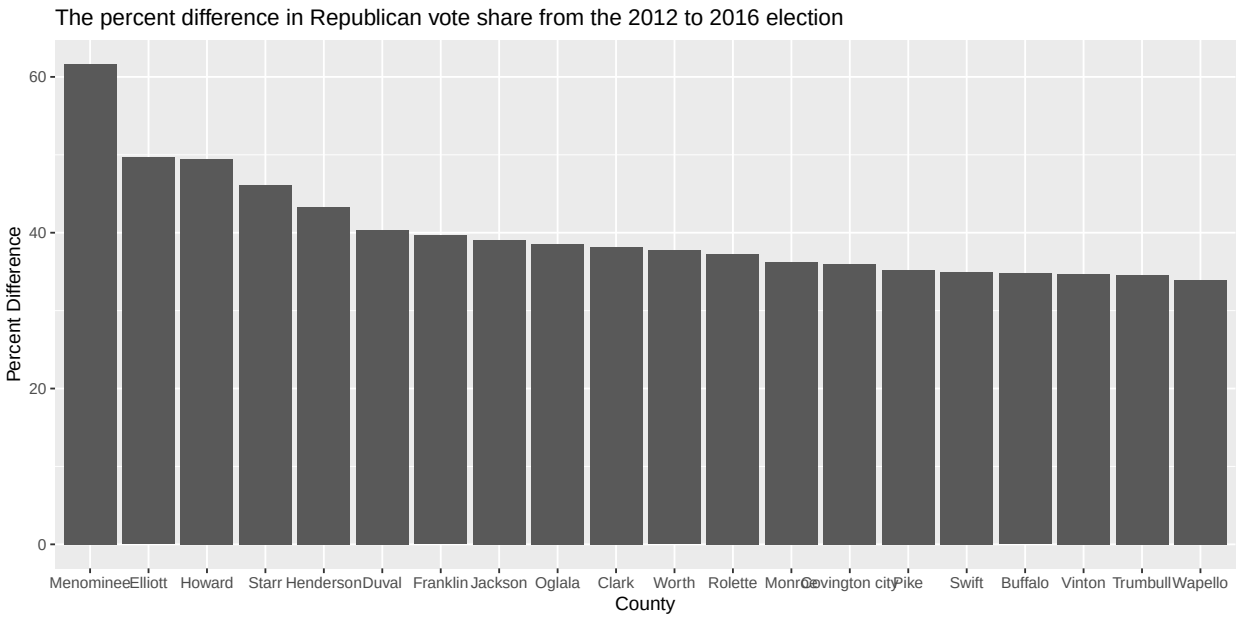
```
# Option 1 (geom_bar)
## geom_bar() uses stat_count() by default: it counts the number of cases at each x position.
## for the purpose of this question, we need to change stat argument (see below).
<data> |>
  slice_max(<variable1>, n = 20) |>
  ggplot(aes(x = fct_reorder(<variable2>, desc(<variable1>)), y = <variable1>)) +
  geom_bar(stat = "identity") +
  labs(title = <title>,
       x = <xlab>, y = <ylab>)

# Option 2 (geom_col)
## geom_col() uses stat_identity(): it leaves the data as is.
<data> |>
  slice_max(<variable1>, n = 20) |>
  ggplot(aes(x = fct_reorder(<variable2>, desc(<variable1>)), y = <variable1>)) +
  geom_col() +
  labs(title = <title>,
       x = <xlab>, y = <ylab>)
```

## Answer 6

```
# Option 1 (geom_bar)
## geom_bar() uses stat_count() by default: it counts the number of cases at each x position.
## for the purpose of this question, we need to change stat argument (see below).
elec |>
  slice_max(gop_vs_diff, n = 20) |>
  ggplot(aes(x = fct_reorder(county, desc(gop_vs_diff)), y = gop_vs_diff)) +
  geom_bar(stat = "identity") +
  labs(title = "The percent difference in Republican vote share from the 2012 to 2016 election",
       x = "County", y = "Percent Difference")
```



```
# Option 2 (geom_col)
## geom_col() uses stat_identity(): it leaves the data as is.
elec |>
  slice_max(gop_vs_diff, n = 20) |>
  ggplot(aes(x = fct_reorder(county, desc(gop_vs_diff)), y = gop_vs_diff)) +
  geom_col() +
  labs(title = "The percent difference in Republican vote share from the 2012 to 2016 election",
       x = "County", y = "Percent Difference")
```




## Question 7: Republican gains in Democrat counties

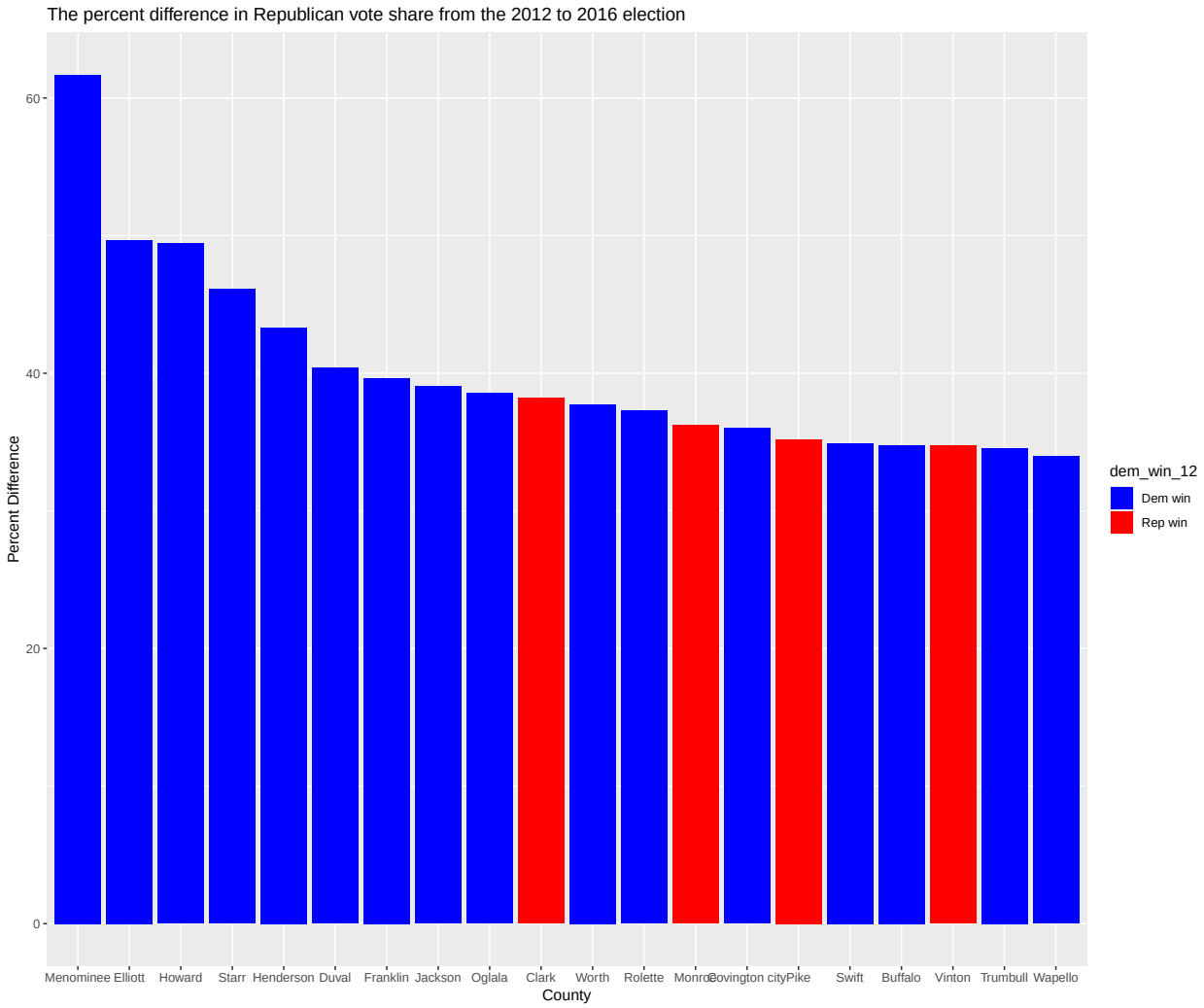
Some of the counties where the Republican party saw greater gains were counties where the Democratic party had the most votes in 2012. Run the following code to create a binary variable that takes the value of 1 whenever the Democrats had the most votes in 2012, and 0 otherwise.

```
elec <- elec |>
  mutate(dem_win_12 = dplyr::if_else(votes_dem_12 > votes_gop_12, "Dem win", "Rep win"))
```

Now repeat the plot in Question 5, adding `mapping = aes(fill = dem_win_12)` to the `geom_bar` function. What is your interpretation of this figure?

## Answer 7

```
elec |>
  slice_max(gop_vs_diff, n = 20) |>
  ggplot(aes(x = fct_reorder(county, desc(gop_vs_diff)), y = gop_vs_diff)) +
  geom_bar(mapping = aes(fill = dem_win_12), stat = "identity") +
  labs(title = "The percent difference in Republican vote share from the 2012 to 2016 election",
       x = "County", y = "Percent Difference") +
  scale_fill_manual(values = c("blue", "red"))
```




## Question 8: Table

Create a table for the top 20 counties in terms of the difference in GOP vote share between the 2012 and 2016 elections (`gop_vs_diff`), using `elec` data. Include `state`, `county`, socio-demographic variables (`pct_for_born15`, `pct_bach_deg15`, `pct_non_white15`, `pct_non_white15`, `pct_unemp12`, `pct_unemp16`) and vote share variables (`gop_vs_12`, `gop_vs_16`, `gop_vs_diff`) as columns. Order the rows based on the values of vote share difference.

*Hint*: Use `knitr::kable()` to produce a nicely formatted table. [Optional] To make the table neater, round off numbers to two decimal places and change the column names. See R documentation (`?kable`) for the arguments.

## Answer 8

```
elec |>
  slice_max(gop_vs_diff, n = 20) |>
  select(state, county, pct_for_born15:gop_vs_diff) |>
  arrange(desc(gop_vs_diff)) |>
```

```
knitr::kable(col.names = c("State", "County",
                           "Foreign born", "Degree", "Non-white", "Unemp. 2012", "Unemp. 2016",
                           "Rep. 2012", "Rep. 2016", "Rep. difference"),
             digits = 2)
```

| State | County | Foreign born | Degree | Non-white | Unemp. 2012 | Unemp. 2016 | Rep. 2012 | Rep. 2016 | Rep. difference |
|---|---|---|---|---|---|---|---|---|---|
| WI | Menominee | 2.85 | 16.11 | 88.99 | 14.2 | 6.4 | 0.13 | 0.21 | 61.68 |
| KY | Elliott | 0.21 | 7.48 | 2.91 | 12.5 | 10.2 | 0.47 | 0.70 | 49.67 |
| IA | Howard | 0.68 | 12.81 | 1.75 | 3.6 | 3.0 | 0.39 | 0.58 | 49.43 |
| TX | Starr | 33.11 | 9.10 | 5.07 | 13.1 | 11.7 | 0.13 | 0.19 | 46.10 |
| IL | Henderson | 0.99 | 13.91 | 2.37 | 7.5 | 5.0 | 0.43 | 0.62 | 43.33 |
| TX | Duval | 4.11 | 8.08 | 14.29 | 6.4 | 10.7 | 0.23 | 0.32 | 40.37 |
| NY | Franklin | 3.78 | 17.68 | 17.02 | 8.6 | 5.1 | 0.36 | 0.50 | 39.63 |
| IA | Jackson | 0.82 | 15.29 | 2.70 | 4.7 | 3.6 | 0.41 | 0.57 | 39.04 |
| SD | Oglala | 0.19 | 11.43 | 94.99 | 13.7 | 10.0 | 0.06 | 0.08 | 38.55 |
| MO | Clark | 0.16 | 12.80 | 2.30 | 7.2 | 6.6 | 0.54 | 0.74 | 38.18 |
| IA | Worth | 0.74 | 15.38 | 2.80 | 4.6 | 3.3 | 0.42 | 0.58 | 37.73 |
| ND | Rolette | 0.59 | 20.86 | 79.94 | 7.4 | 6.6 | 0.24 | 0.33 | 37.26 |
| OH | Monroe | 0.14 | 9.86 | 2.00 | 8.1 | 9.1 | 0.52 | 0.72 | 36.21 |
| VA | Covington city | 1.48 | 9.28 | 18.31 | 7.8 | 5.3 | 0.42 | 0.57 | 36.02 |
| OH | Pike | 0.58 | 11.83 | 3.71 | 11.9 | 6.9 | 0.49 | 0.67 | 35.18 |
| MN | Swift | 1.72 | 16.18 | 4.08 | 4.6 | 3.6 | 0.44 | 0.60 | 34.92 |
| SD | Buffalo | 0.00 | 9.51 | 81.50 | 10.3 | 8.0 | 0.26 | 0.35 | 34.76 |
| OH | Vinton | 0.18 | 9.15 | 2.84 | 10.3 | 6.2 | 0.52 | 0.70 | 34.74 |
| OH | Trumbull | 1.58 | 17.32 | 11.23 | 8.8 | 6.0 | 0.38 | 0.51 | 34.56 |
| IA | Wapello | 7.78 | 16.76 | 7.99 | 6.5 | 6.8 | 0.43 | 0.58 | 33.96 |