Descriptive Analysis

Reference script: 1_descriptive.R

1. Source Files

Load source files. We will use `primary_results.csv` and `county_facts.csv` extensively. `county_facts_dictionary.csv` will be a reference to demographic codes.

```
srcPrimary <- read.csv('primary_results.csv', stringsAsFactors = FALSE, encoding = 'UTF-8')
srcDemogr <- read.csv('county_facts.csv', stringsAsFactors = FALSE, encoding = 'UTF-8')
srcDict <- read.csv('county_facts_dictionary.csv', stringsAsFactors = FALSE, encoding = 'UTF-8')
```

Load county real gross domestic product (RGDP) and population data. We'll use these data sets in **Section 6**.

```
srcRgdp <- read.csv('county_rgdp.csv', stringsAsFactors = FALSE, encoding = 'UTF-8')
srcPop <- read.csv('county_pop.csv', stringsAsFactors = FALSE, encoding = 'UTF-8')
```

---

2. Cleanup Primary Results Data

It's important to change county names to lower case (for all data sets) to prevent future merging duplicates or errors; e.g. `St Louis City` vs `St Louis city`

```
primary <- select(srcPrimary, fips, state = state_abbreviation, county,
                  candidate, party, votes, fraction_votes) %>% mutate(county = tolower(county))
```

---

3. Extract Demographic Data

Extract some useful demographic data to work with. Refer to `county_facts_dictionary.csv` to match demographic codes. For now, we'll use:

Code	Description
INC110213	Median household income, 2009-2013
EDU685213	Bachelor's degree or higher, percent of persons age 25+, 2009-2013
POP060210	Population per square mile, 2010
RHI825214	White alone, not Hispanic or Latino, percent, 2014
RHI725214	Hispanic or Latino, percent, 2014
HSD310213	Persons per household, 2009-2013

We might be interested in certain states/region and not the whole nation at once. The `demogrSomeF()` function returns the county demographic data we selected above in states specified by the user. Function usage: Input state abbreviations (not full state names) as function arguments. Quotations or capitalizations are not necessary.

```
demogrSomeF <- function(...) {
  states <- gsub('\"', '', toupper(sapply(substitute(list(...)), deparse)[-1]))

  demogrSome <- filter(srcDemogr, state_abbreviation %in% states) %>%
    select(fips, state = state_abbreviation, county = area_name,
           income = INC110213, education = EDU685213, density = POP060210,
           white = RHI825214, hispanic = RHI725214, household = HSD310213) %>%
    mutate(county = tolower(gsub(' County', '', county)))

  assign('demogrSome', demogrSome, envir = globalenv())
}
```

Select one of the four US regions: We'll focus on the Midwest:

```
# Northeast:
demogrSomeF(CT, ME, MA, NH, NJ, NY, PA, RI, VT)
# South:
demogrSomeF(AL, AR, DE, FL, GA, KY, LA, MD, MS, NC, OK, SC, TN, TX, VA, WV)
# West:
demogrSomeF(AK, AZ, CA, CO, HI, ID, MT, NV, NM, OR, UT, WA, WY)
# Midwest:
demogrSomeF(IL, IN, IA, KS, MI, MN, MO, NE, ND, OH, SD, WI)
```

Important: The rest of the script were tested on the 'Midwest' region. As such, the model and result interpretations may be different if the user uses a different region. For the first run-through, using the 'Midwest' region is recommended.

Let's narrow down the list of candidates to some 'big names':

```
candidates <- c('Donald Trump', 'Ted Cruz', 'Hillary Clinton', 'Bernie Sanders')
```

Merge primary results with filtered county demographic data for the region.

```
main <- merge(primary, demogrSome, by = c('fips', 'state', 'county')) %>%
  filter(candidate %in% candidates)
```

For each party, find the primary winners of each county in the region.

```
winners <- rbind(
  group_by(primary, fips, state, county, party) %>% filter(party == 'Democrat') %>%
    summarize(winner = candidate[which.max(fraction_votes)],
              votes = max(votes),
              fraction_votes = max(fraction_votes)),
  group_by(primary, fips, state, county, party) %>% filter(party == 'Republican') %>%
    summarize(winner = candidate[which.max(fraction_votes)],
              votes = max(votes),
              fraction_votes = max(fraction_votes))
  ) %>% filter(winner %in% candidates)
```

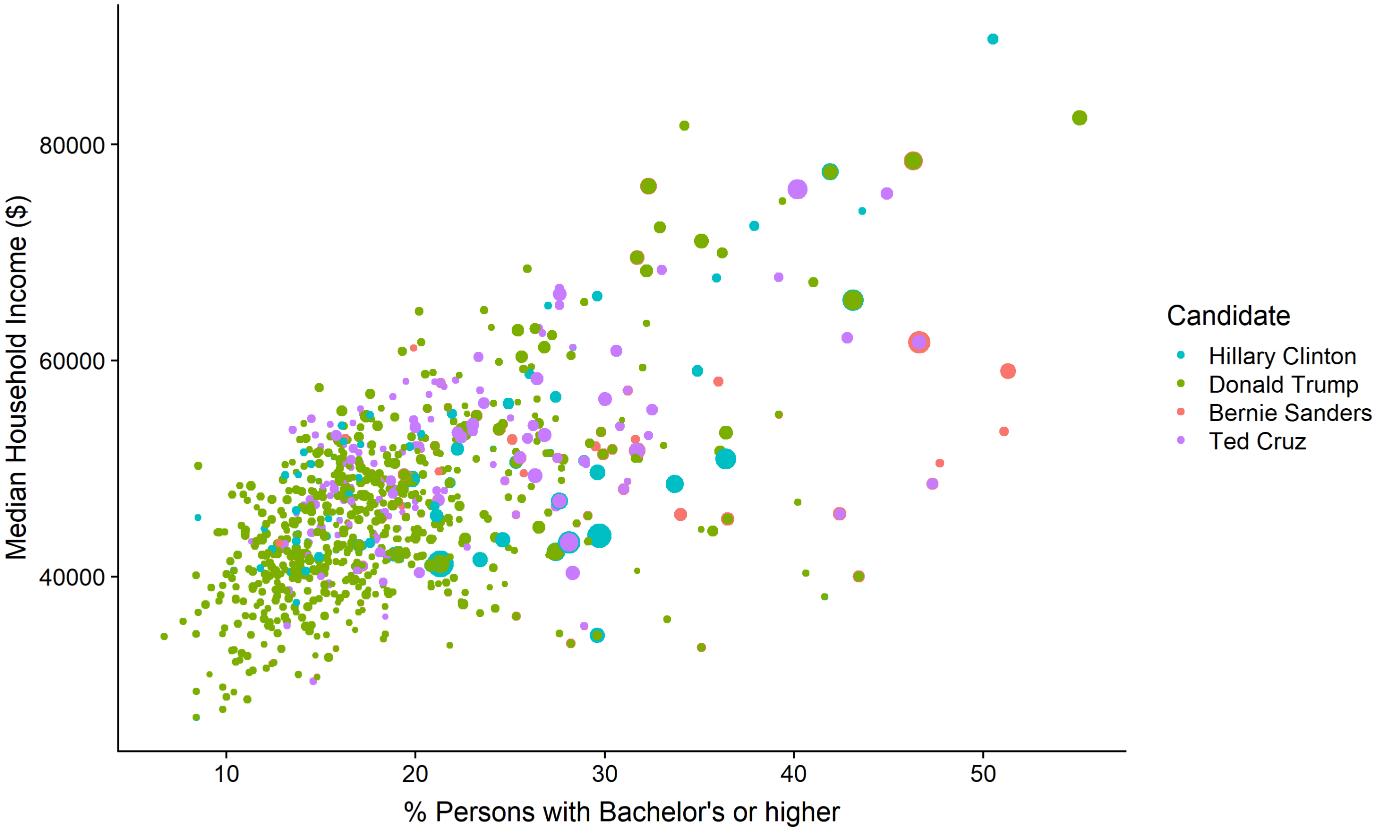
Merge primary winners with filtered demographic data for the region.

```
winners <- merge(winners, demogrSome, by = c('fips', 'state', 'county'))
```

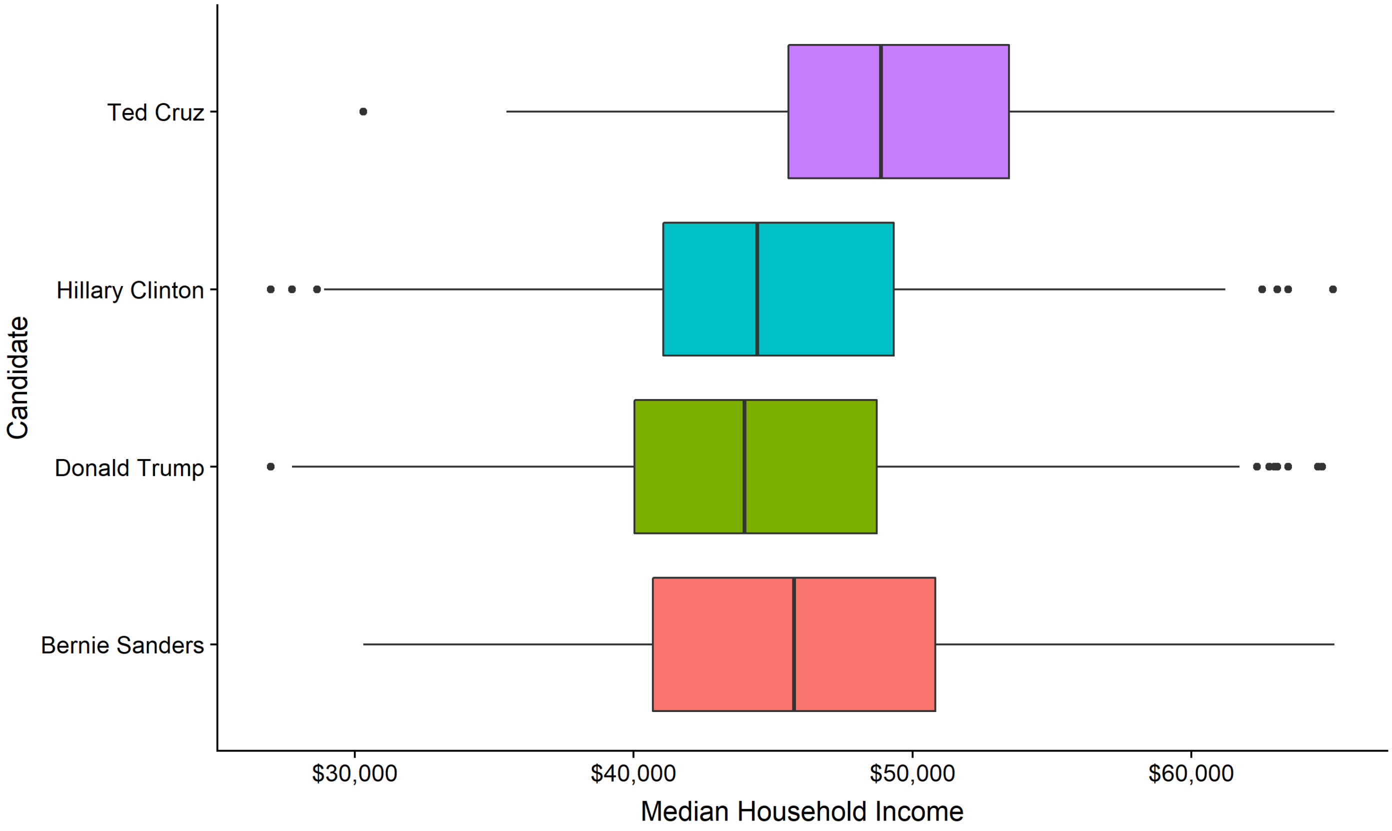
Summarize primary winners and the average demographic they attract:

```
group_by(winners, winner) %>%
  summarize(income = round(mean(income)), education = round(mean(education)),
            density = round(mean(density)), white = round(mean(white)),
            hispanic = round(mean(hispanic)), household = round(mean(household))) %>%
  dplyr::rename(candidate = winner)

# Output:
candidate          income   education   density   white   hispanic  household
1 Bernie Sanders   46981    20          106       89      4         2
2 Donald Trump     45151    18          103       89      4         2
3 Hillary Clinton  45795    18          177       90      4         2
4 Ted Cruz         50239    21          149       90      4         2
```

Simple scatterplot of primary winners based on `income` and `education`:



Simple box plot of primary winners and the income demographic they attract:



4. Extract Candidate Data by County and Demographics

Keep in mind that that we're still focusing on only 4 big candidates in the region (Midwest in this example). Instead of winners, we'll now focus on each candidate and their performance `fraction_votes` in all Midwestern counties.

The loop below populates the list `cddList` with data frames. Each of the 4 candidates has a data frame containing his/her performance in all counties within the region, merged with county demographic data.

```
cddList <- list()

for (i in candidates) {
  cddList[[match(i, candidates)]] <- filter(main, candidate == i)

  # Name each item in the list (each data frame) with the candidate's last name
  names(cddList)[match(i, candidates)] <- strsplit(i, ' ') %>%
    sapply('[[', length(unlist(strsplit(i, ' '))))

  rm (i)
}
```

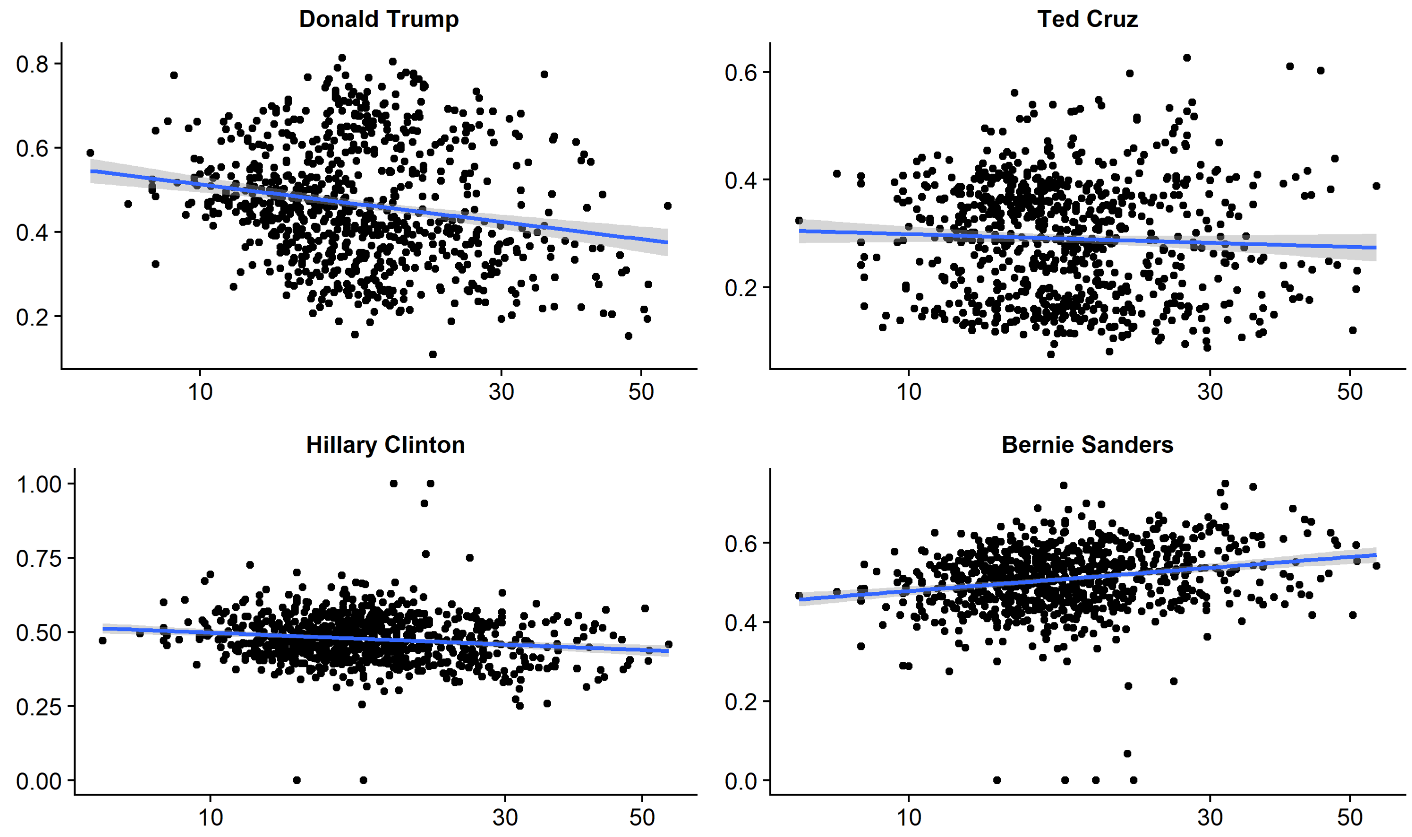
---

5. Build Plot Functions

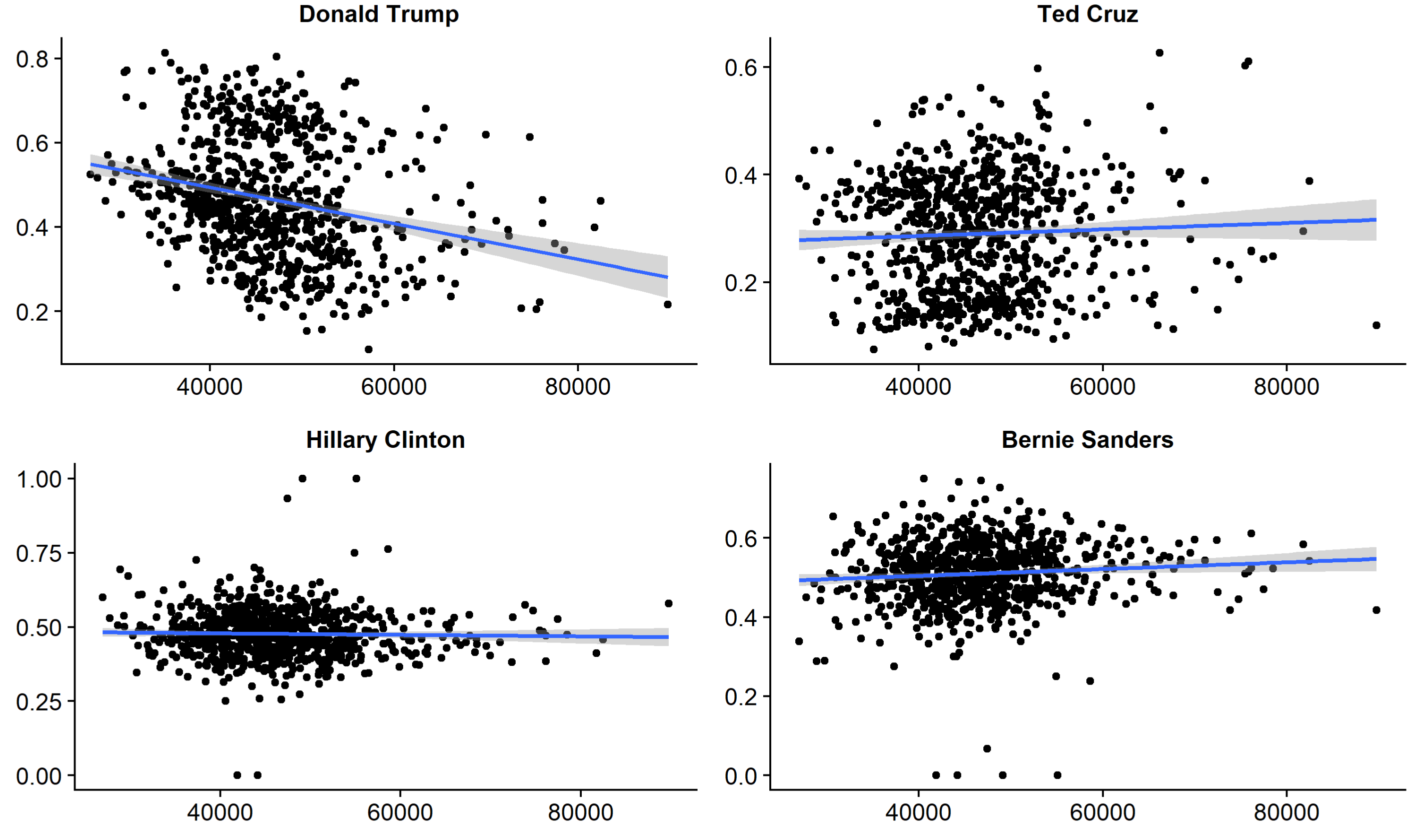
We now have a populated list of candidates and their respective vote statistics (merged with demographic data) in `cddList`. The next step is to plot the fraction of votes (a performance metric) against various demographic data. We'll build functions `cddPlot()` and `cddPlotLog()` to maintain consistency, reduce clutter, and reduce the potential for mistakes. It's not exactly clear when to use a log scale, but it's probably a good idea when small values are compressed down to the bottom of the graph. For normal plots, use `cddPlot()` and for log transformations on the explanatory variable (x-axis), use `cddPlotLog()`.

```
cddPlot <- function(metric) {
  plot_grid(plotlist = lapply(cddList, function(df)
    ggplot(df, aes(x = eval(parse(text = metric)), y = fraction_votes)) +
      geom_point() +
      geom_smooth(method = 'lm', formula = y ~ x) +
      ggtitle(label = df[1, 'candidate']) +
      theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
            plot.title = element_text(size = 12))), align = 'h')
  }

cddPlotLog <- function(metric) {
  plot_grid(plotlist = lapply(cddList, function(df)
    ggplot(df, aes(x = eval(parse(text = metric)), y = fraction_votes)) +
      geom_point() +
      scale_x_log10() +
      geom_smooth(method = 'lm', formula = y ~ x) +
      ggtitle(label = df[1, 'candidate']) +
      theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
            plot.title = element_text(size = 12))), align = 'h')
  }
```

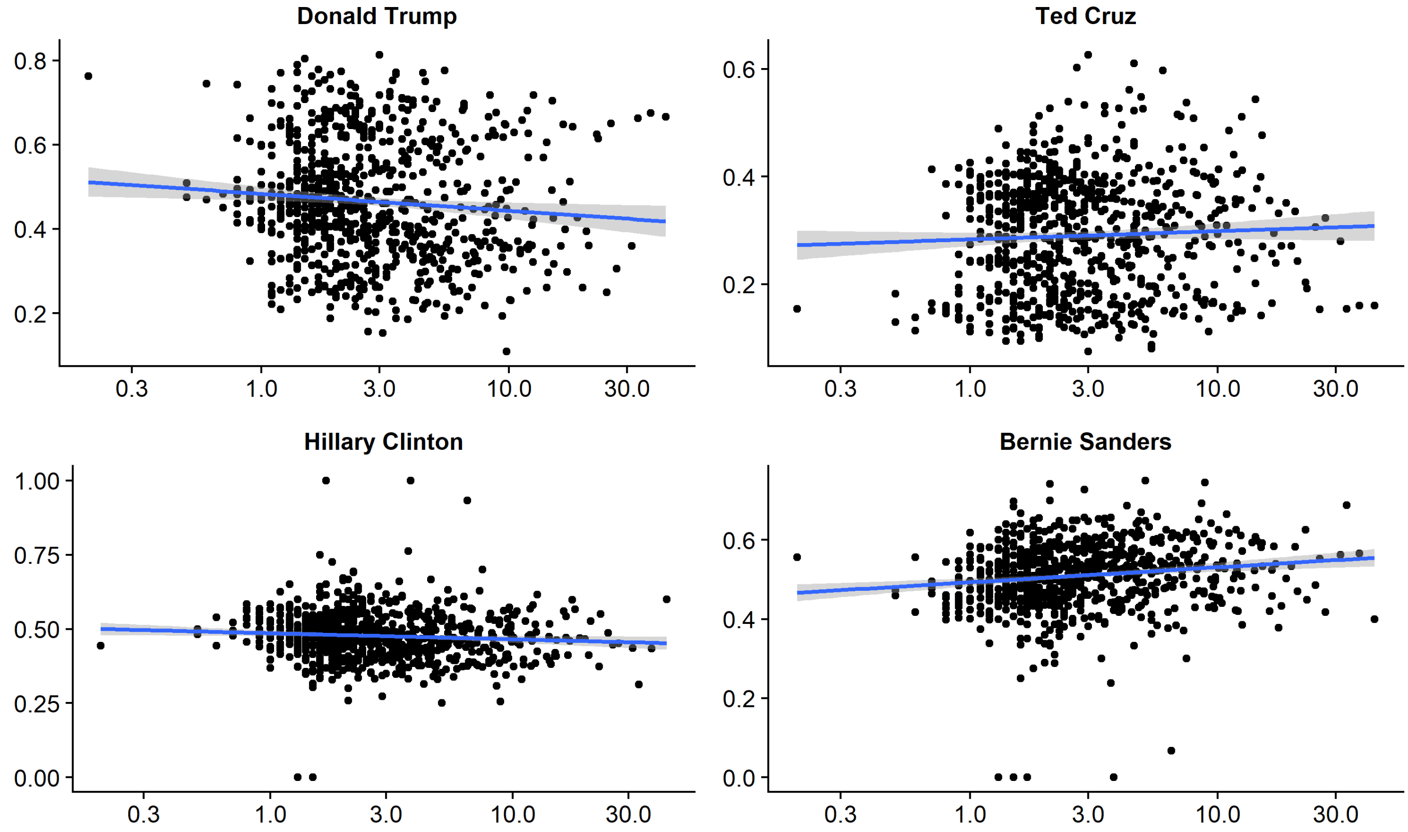
Plot `fraction_votes` as a function of log `education`. A negative slope of the regression line means that the fraction of votes in a county decreases as the log of percentage of population with a bachelor's degree or higher increases. In the case of Donald Trump, the fraction of votes for him tends to decrease in areas of high log `education`. On the other hand, Bernie Sanders enjoy high popularity in areas of high log `education`.



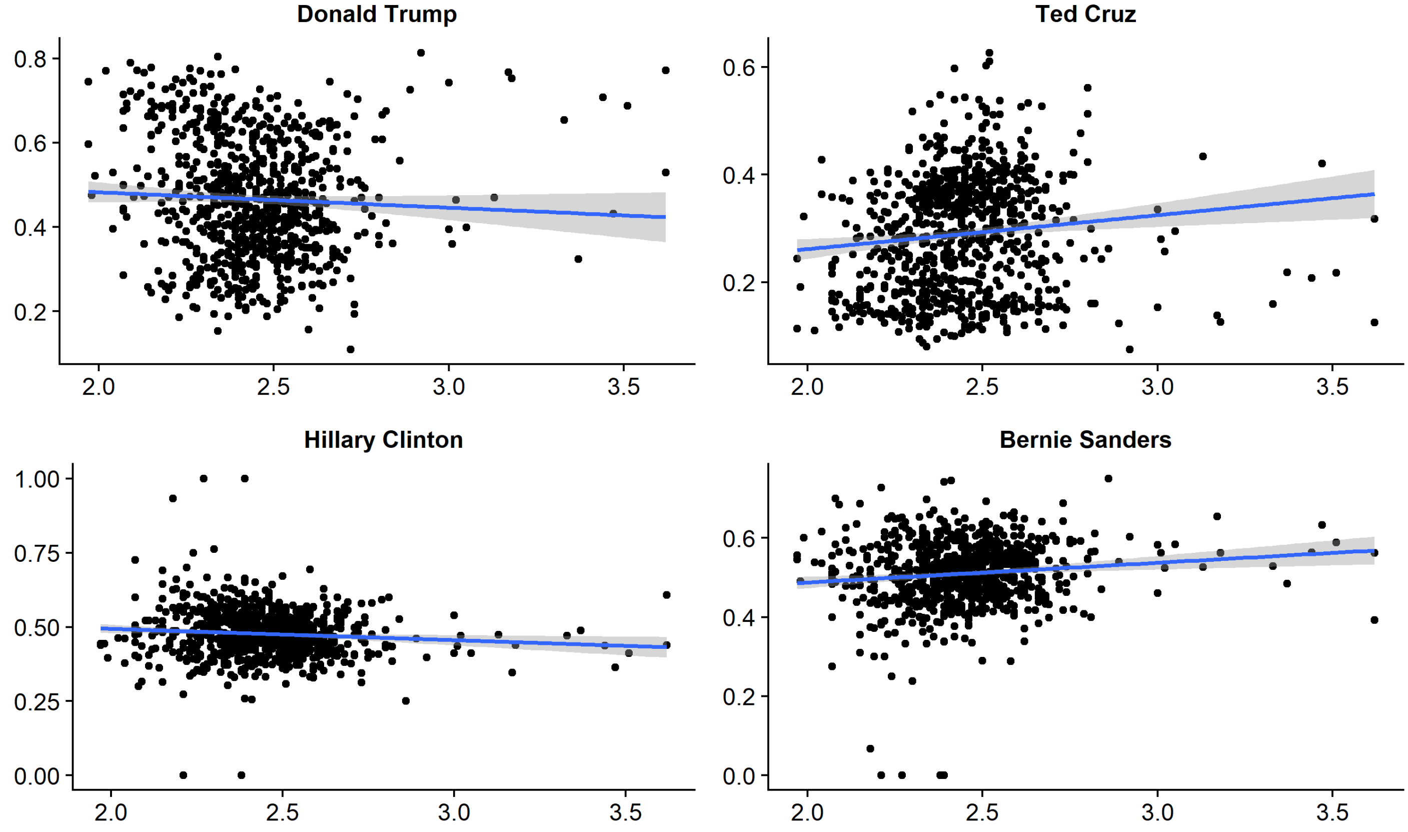
Plot `fraction_votes` as a function of `income`. Trump's popularity drastically decrease as median income increase.



Plot fraction_votes as a function of log hispanic. Compared to other candidates, Trump seems to be slightly unpopular in areas of high log hispanic.



Plot fraction_votes as a function of household. Ted Cruz seems to be largely popular with large households.



6. Real Gross Domestic Product and Population

Join the region demographic data with the RGDP and population data sets.

```
main <- inner_join(inner_join(main, mutate(srcRgdp, county = tolower(county)),
                              by = c('state', 'county')),
                   mutate(srcPop, county = tolower(county)), by = c('fips', 'county'))
```

Compute 2012 and 2015 RGDP per capita:

```
main$rgdppc12 <- main$rgdp2012 / main$pop2012
main$rgdppc15 <- main$rgdp2015 / main$pop2015
```

We can derive trends from annual data. A simple trend would be to calculate RGDP per capita change from 2012 to 2015 (decimals).

```
main$rgdppcDelta <- (main$rgdppc15 - main$rgdppc12) / main$rgdppc12
```

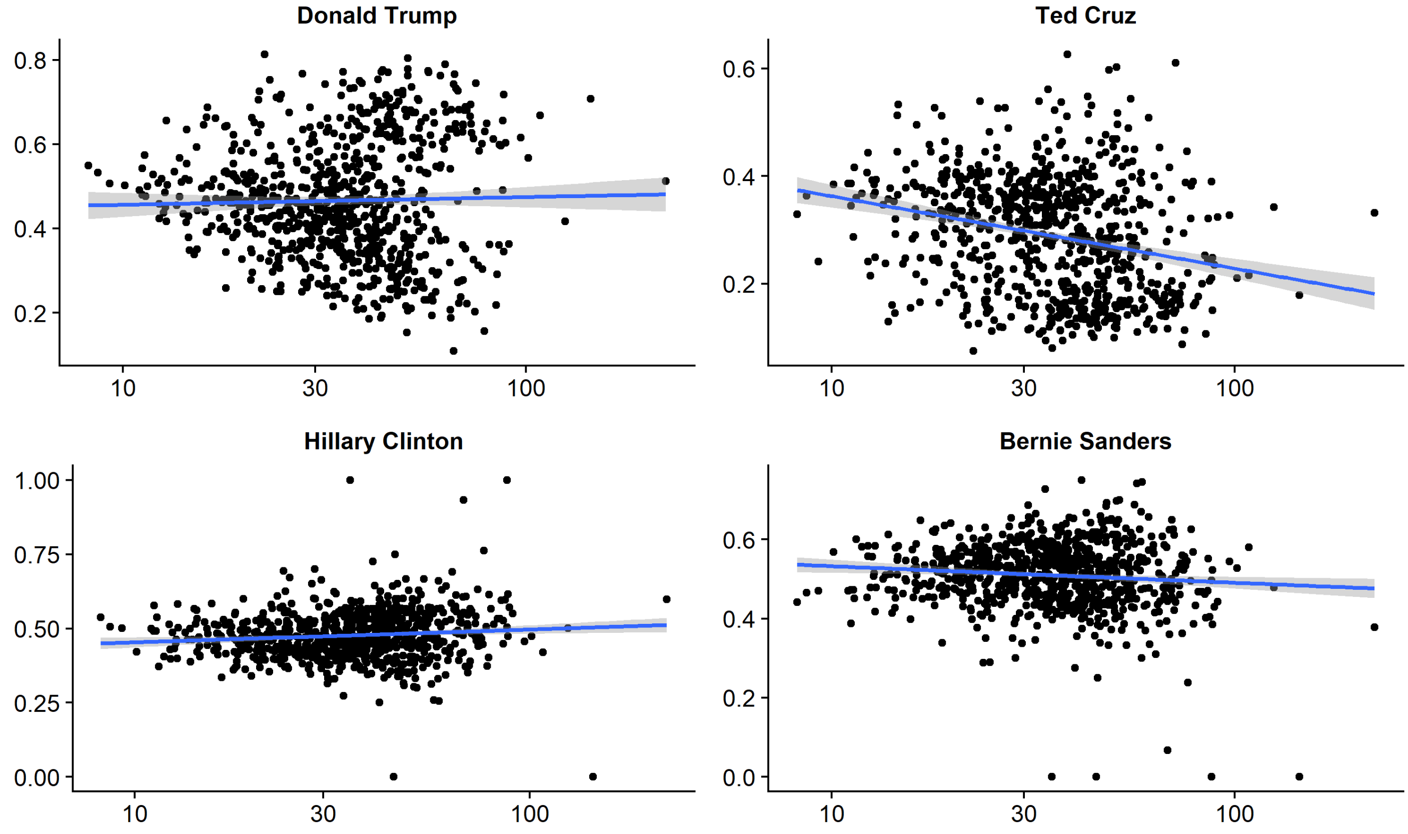
We may need to perform log transformations, but negative values (decreasing trend) will be ignored (undefined)! We'll normalize `rgdppcDelta` by rescaling it from 0 to 1.

```
main$rgdppcDeltaNorm <- (main$rgdppcDelta - min(main$rgdppcDelta)) /
  (max(main$rgdppcDelta) - min(main$rgdppcDelta))
```

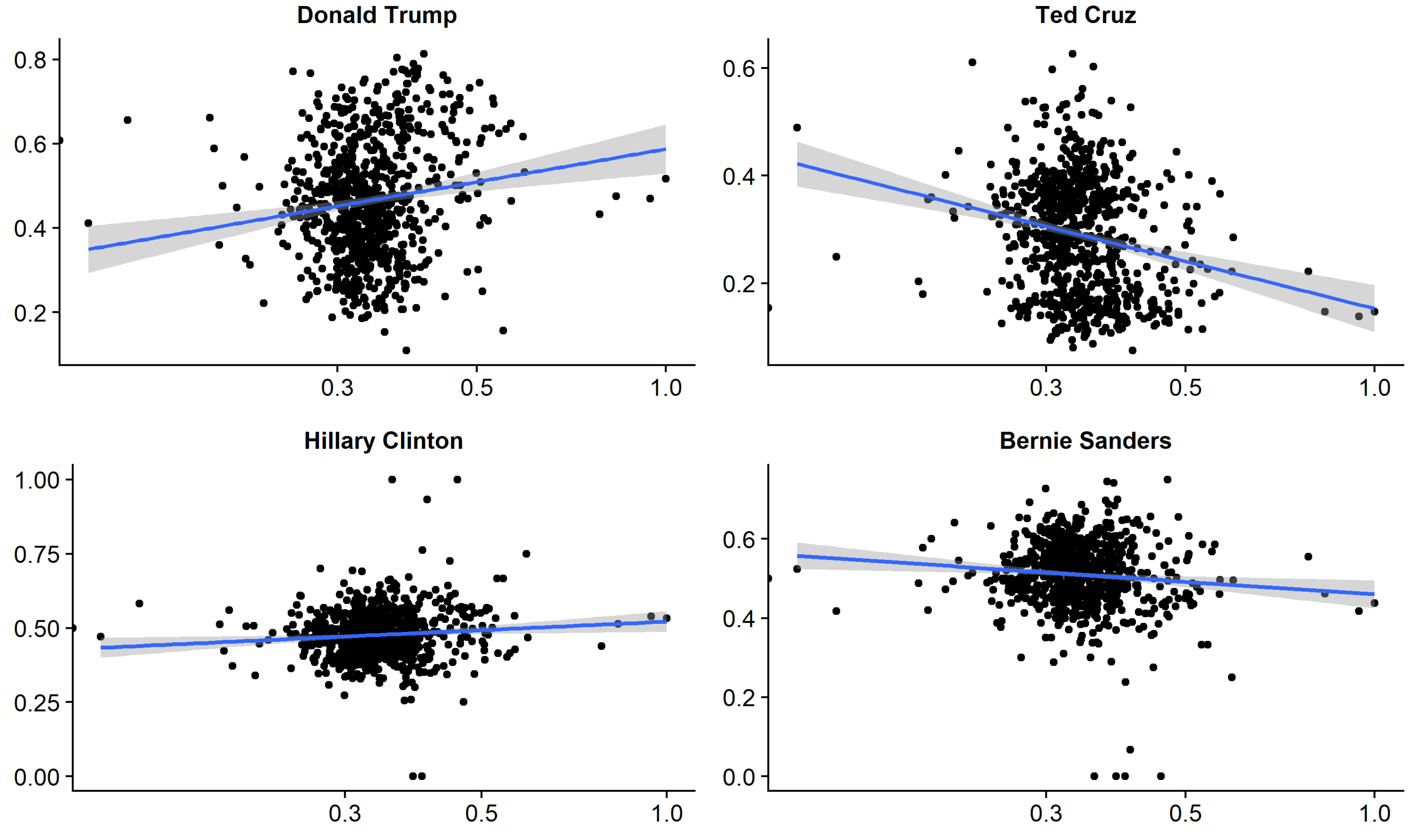
Join (update) the list of candidates `cddList` we created earlier with new data:

```
for (i in seq(length(cddList))) {
  cddList[[i]] <- merge(cddList[[i]], main)
  rm(i)
}
```

Notice Ted Cruz's popularity decline in areas of high log RGDP per capita.



Notice Donald Trump's popularity increase in areas of increasing log RGDP per capita from 2012 to 2015. Earlier, we noticed that Donald Trump's probability of winning increases as median household income decreases. We can infer that Trump's chance of winning is high in areas of low income but increasing RGDP per capita.



To be safe, we'll perform a correlation test to make sure that the explanatory variables `income` and `rgdppcDelta` are not significantly correlated to each other. The null hypothesis is that the true correlation is equal to 0. Since the P-value is `0.24`, we cannot reject the null hypothesis and must conclude that there is no significant correlation. This is a simple test for multicollinearity.

```
cor.test(main$income, main$rgdppcDelta, method = 'pearson')

# Output:
t = -1.1752, df = 3218, p-value = 0.24
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.05521418  0.01384058
sample estimates:
       cor
-0.0207115
```

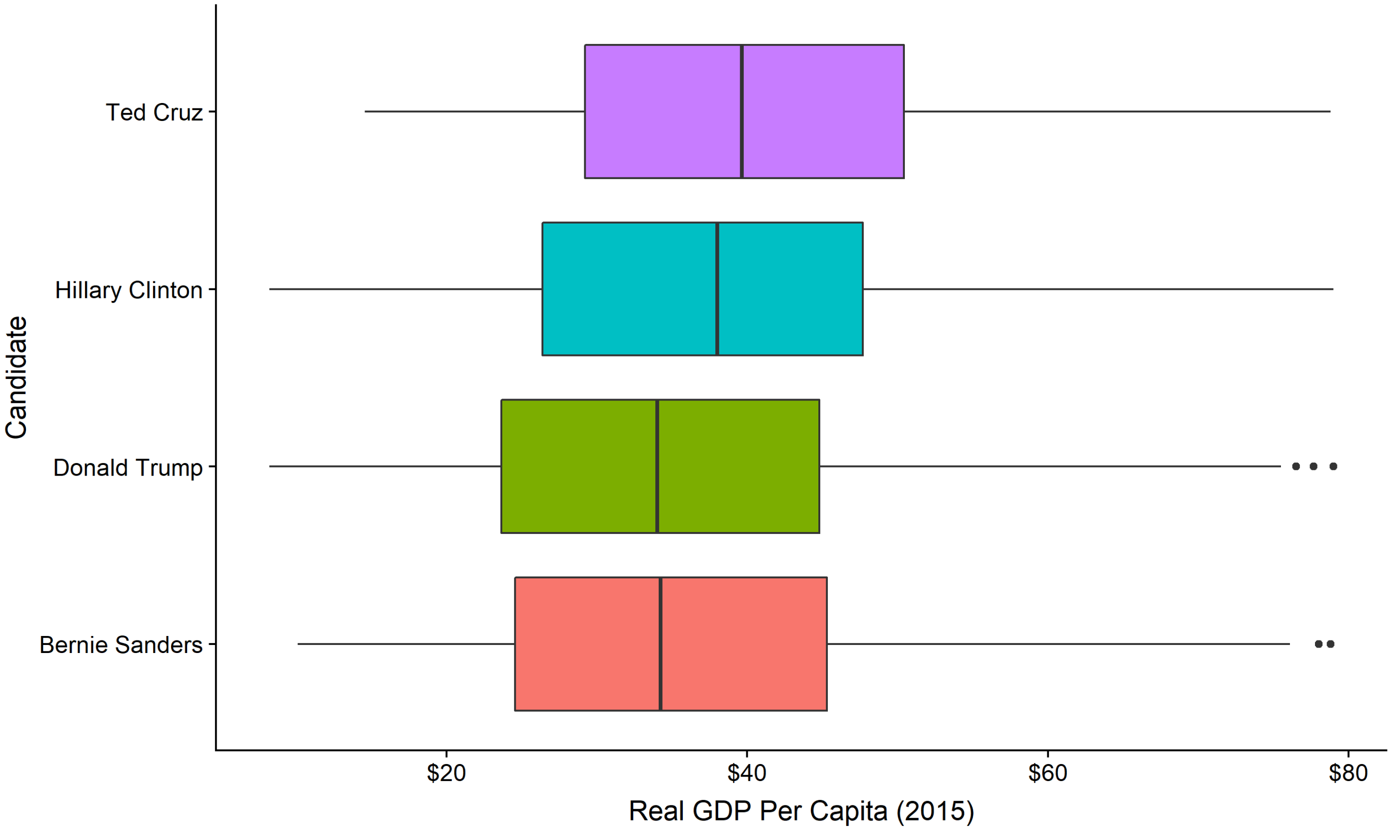
With no significant correlation between predictor variables `income` and `rgdppcDelta`, we'll proceed with multiple regression and examine the output:

```
summary(lm(fraction_votes ~ income + rgdppcDelta, data = cddList[['Trump']]))

# Output:
Residuals:
     Min       1Q   Median       3Q      Max
-0.34112 -0.09722 -0.01863  0.09665  0.33332

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.527e-01  2.696e-02  24.214  < 2e-16 ***
income      -4.223e-06  5.670e-07  -7.448 2.45e-13 ***
rgdppcDelta  1.380e-01  3.434e-02   4.019 6.40e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1342 on 802 degrees of freedom
Multiple R-squared:  0.08332,   Adjusted R-squared:  0.08103
F-statistic: 36.45 on 2 and 802 DF,  p-value: 7.081e-16
```

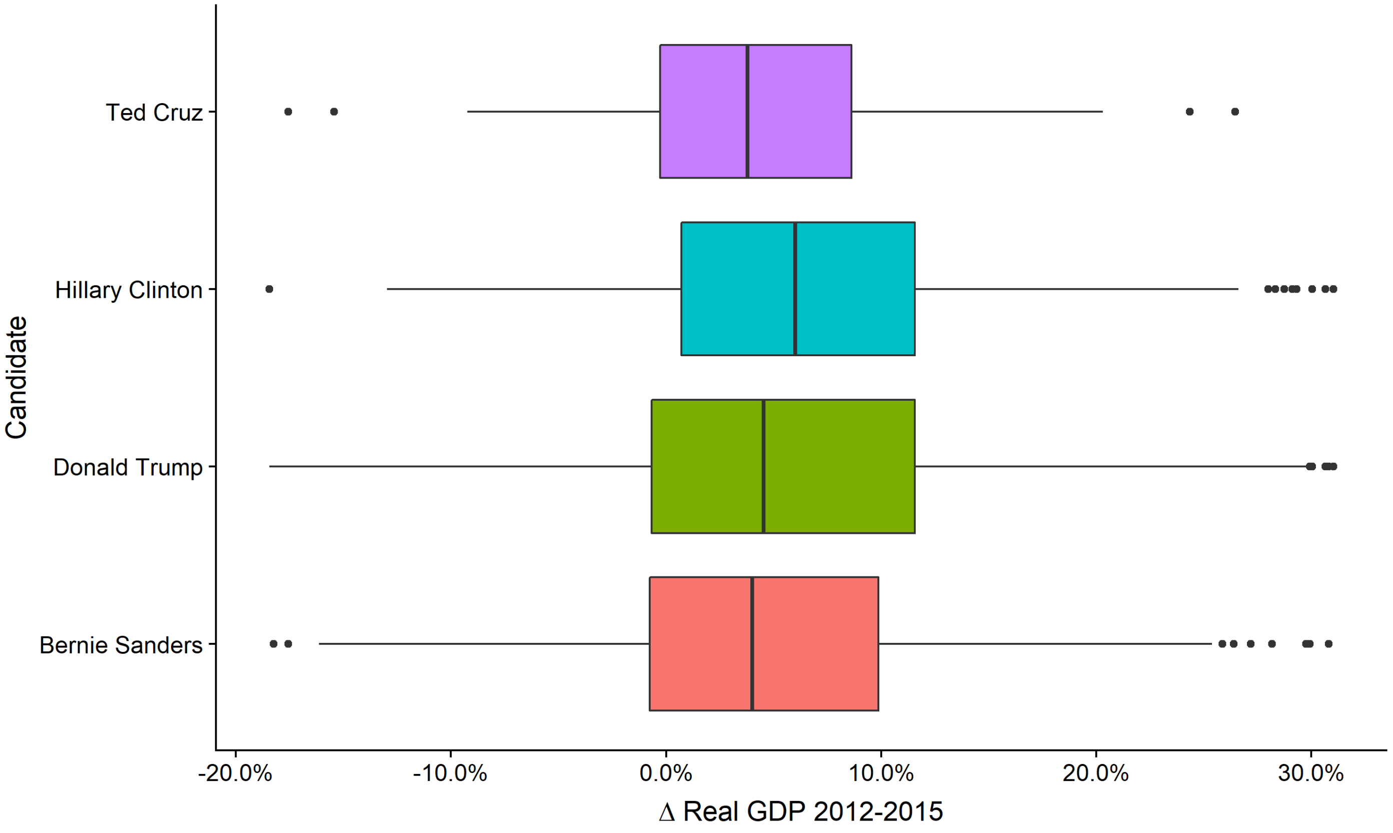
Join each party with our updated `main` data frame.

```
winners <- merge(winners, main)
```

Simple box plot of Republican winners as a function of 2015 RGDP per capita. We can remove some outliers using `boxplot.stats(df$y)$stats[c(1, 5)]`



Box plot of Republican winners as a function of RGDP change from 2012-2015.


Candidate
Ted Cruz
Hillary Clinton
Donald Trump
Bernie Sanders
Δ Real GDP 2012-2015