COVID-19 Vaccination Rates and Google Search Data

Arianna Schmid and Stacey Frank

2021-12-13

Contents

Introduction

Vaccines to control the coronavirus disease 2019 (COVID-19) became available to the public in the first half of 2021. Rejection and indecision towards being vaccinated is evident across the United States.The motivation for this study is to provide a better understanding of reasons for COVID-19 vaccine refusal in the United States. This can help public health messaging campaigns be more targeted and effective when promoting vaccination.

Google data is useful for exploring this topic because there is previous research that people feel freer to Google socially stigmatized topics than they would be to admit such opinions in a survey or other form of data collection. As a result, our primary research question is what is the relationship, if any, between state-level COVID-19 vaccine rates and the types of Google searches that are made about vaccines? In particular, are vaccine myths more commonly searched for in states that also have low vaccination rates? A secondary question we investigate is does the relationship between COVID vaccine rates and Google searches change between June and September 2021?

Data Collection

1. Google Trends and Keywords

The CDC provides lists of the most common questions about the COVID-19 vaccine (**CDC_2021?**). Similarly, the Mayo Clinic provides information on the most common myths surrounding the vaccine (**COVID-19?** vaccine myths debunked_2021). Using these two data sources, a list of 12 keyword search terms was constructed. We call this list "k" to signify "keywords." It consists of the two general searches "covid vaccine" and "covid vaccine near me," five meainstream searches such as "covid vaccine side effects," and 5 myth-related searches such as "covid vaccine microchip."

```r
# Gtrends keyword searches
# Info about keyword searches: https://github.com/PMassicotte/gtrendsR/issues/268
k <- c( "covid vaccine",
        "covid vaccine near me",
        "covid vaccine safe",
        "covid vaccine ingredients",
        "covid vaccine pregnant",
        "covid vaccine protect",
        "covid vaccine side effects",
        "covid vaccine microchip",
        "covid vaccine dna",
        "covid vaccine fetal",
        "covid vaccine infertility",
        "covid vaccine magnet")
```

The gtrendsR package was used to work with Google Trends Queries. This allowed us to look at the trends, or number of hits, for each of the 12 keyword searches. In addition, we studied the hit results in each of the 50 states and the District of Columbia. Trends data was pulled for three time periods: 1/1/21-9/20/21, 4/1/21-6/20/21, and 7/1/21-9/20/21 since vaccine availability varied by state. Furthermore, each element in "k" was renamed based on its index (hits.1, hits.2, . . . hits.12) for code efficiency.

```r
get.hits.results <-  function(date){
    for (i in 1:length(k)){
          new_frame <- paste("Keyword",i,sep = "")
          assign(new_frame, gtrends(k[i], geo = "US",
                   time = date, low_search_volume = T)
                 )
    }

    hits_results <- Keyword1$interest_by_region %>%
      left_join(Keyword2$interest_by_region, by = "location") %>%
      left_join(Keyword3$interest_by_region, by = "location") %>%
      left_join(Keyword4$interest_by_region, by = "location") %>%
      left_join(Keyword5$interest_by_region, by = "location") %>%
      left_join(Keyword6$interest_by_region, by = "location") %>%
      left_join(Keyword7$interest_by_region, by = "location") %>%
      left_join(Keyword8$interest_by_region, by = "location") %>%
      left_join(Keyword9$interest_by_region, by = "location") %>%
      left_join(Keyword10$interest_by_region, by = "location") %>%
      left_join(Keyword11$interest_by_region, by = "location") %>%
      left_join(Keyword12$interest_by_region, by = "location") %>%
      as_tibble() %>%
      select(c(1,2,6,10,14,18,22,26,30,34,38,42,46))

    hits_results %<>% rename( hits.1 = hits.x,
                          hits.2 = hits.y,
                          hits.3 = hits.x.x,
                          hits.4 = hits.y.y,
                          hits.5 = hits.x.x.x,
                          hits.6 = hits.y.y.y,
                          hits.7 = hits.x.x.x.x,
                          hits.8 = hits.y.y.y.y,
                          hits.9 = hits.x.x.x.x.x,
                          hits.10 = hits.y.y.y.y.y,
                          hits.11 = hits.x.x.x.x.x.x,
                          hits.12 = hits.y.y.y.y.y.y)
    print(hits_results)

}

hits.results.jan <- get.hits.results("2021-01-1 2021-09-20")
hits.results.june <- get.hits.results("2021-04-1 2021-06-20")
hits.results.sept <- get.hits.results("2021-07-1 2021-09-20")
```

```
print(hits.results.jan)
```

```
## # A tibble: 51 x 13
##    location      hits.1 hits.2 hits.3 hits.4 hits.5 hits.6 hits.7 hits.8 hits.9
##    <chr>          <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
##  1 New Jersey       100     87     73     54     77     38     83     47     55
##  2 Massachusetts     93     76     62     58     70     33     80     40     64
##  3 Connecticut       93     73     76     49     87     29     91     78     56
##  4 Rhode Island      88     70    100     68    100     22     89      0     32
##  5 Pennsylvania      86     98     68     64     65     26     93     12     60
##  6 Maryland          83     81     54     62     71     29     80     68     32
##  7 Delaware          82     93     75     38     77     36     87      0     71
##  8 New York          82     78     57     57     64     27     65     42     42
##  9 Vermont           79     53     54     26     59      0     67      0      0
## 10 Maine             79     63     69    100     80     28     99      0     55
## # ... with 41 more rows, and 3 more variables: hits.10 <int>, hits.11 <int>,
## #   hits.12 <int>
```

```
print(hits.results.june)
```

```
## # A tibble: 51 x 13
##    location      hits.1 hits.2 hits.3 hits.4 hits.5 hits.6 hits.7 hits.8 hits.9
##    <chr>          <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
##  1 Vermont          100     61     18      0     39      0     78      0      0
##  2 Maine             98     80     48     63     52      0     87      0     43
##  3 Massachusetts     95     68     46     32     74     23     77     77     47
##  4 Rhode Island      95     54    100     89     20    100     81      0      0
##  5 Connecticut       94     78     41     57     53     53     76     50     52
##  6 Oregon            87     86     28     32     62     17     83      0     51
##  7 New Jersey        84     80     53     36     57     21     64     40     42
##   8 Washington        82    100     41     54     65      9     71     26     60
##   9 Maryland          80     72     34     58     55     10     72      0     38
##  10 Delaware          75     92     50    100      0     38     86      0     55
## # ... with 41 more rows, and 3 more variables: hits.10 <int>, hits.11 <int>,
## #   hits.12 <int>
```

```
print(hits.results.sept)
```

```
## # A tibble: 51 x 13
##    location       hits.1 hits.2 hits.3 hits.4 hits.5 hits.6 hits.7 hits.8 hits.9
##    <chr>           <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
##  1 Hawaii            100     80    100     58     50     26     50      0     34
##  2 Arkansas           96     83     49     71    100     28     86      0     38
##  3 Alaska             93     80     81     35     41      0     68      0      0
##  4 Alabama            90     86     86     91     65     38     84      0     51
##  5 Louisiana          89     81     83     62     65     64     78      0      0
##  6 Oregon             86     77     64     74     69     16     49      0     33
##  7 Idaho              84     82     63     27     78     48    100      0     32
##  8 Tennessee          84     86     60     50     45     20     65      0     46
##  9 South Carolina     83     82     57     51     50     56     69      0     47
## 10 Washington         83     81     39     68     87     45     47     50     30
## # ... with 41 more rows, and 3 more variables: hits.10 <int>, hits.11 <int>,
## #   hits.12 <int>
```

Next, a data frame was created to list the count of states that have a gtrends ranking present for the specified search term.

```
#create a data frame that lists the count of states that have a gtrends ranking for the specified searc
get.search.terms <- function(hits_results){
  j <- c("hits.1", "hits.2", "hits.3", "hits.4",
        "hits.5", "hits.6", "hits.7", "hits.8",
        "hits.9", "hits.10","hits.11","hits.12")

  search_terms <- apply(!is.na(hits_results), 2, sum) %>%
    as_tibble() %>%
    slice_tail(n=12) %>%
    cbind(j,k) %>%
    relocate(value,.after = k)

  search_terms %<>% rename(var_name = j, search = k, num_states = value)

  print(search_terms)
}

search.terms.jan <- get.search.terms(hits.results.jan)
search.terms.june <- get.search.terms(hits.results.june)
search.terms.sept <- get.search.terms(hits.results.sept)
```

```
print(search.terms.jan)
```

```
##    var_name                     search num_states
## 1    hits.1              covid vaccine         51
## 2    hits.2      covid vaccine near me         51
## 3    hits.3         covid vaccine safe         51
## 4    hits.4  covid vaccine ingredients         51
## 5    hits.5     covid vaccine pregnant         50
## 6    hits.6      covid vaccine protect         44
## 7    hits.7 covid vaccine side effects         51
## 8    hits.8    covid vaccine microchip         23
## 9    hits.9          covid vaccine dna         48
## 10  hits.10        covid vaccine fetal         31
## 11  hits.11  covid vaccine infertility         48
## 12  hits.12       covid vaccine magnet         43
```

```
print(search.terms.june)
```

```
##    var_name                     search num_states
## 1    hits.1              covid vaccine         51
## 2    hits.2      covid vaccine near me         51
## 3    hits.3         covid vaccine safe         50
## 4    hits.4  covid vaccine ingredients         47
## 5    hits.5     covid vaccine pregnant         45
## 6    hits.6      covid vaccine protect         40
## 7    hits.7 covid vaccine side effects         51
## 8    hits.8    covid vaccine microchip         17
## 9    hits.9          covid vaccine dna         43
## 10  hits.10        covid vaccine fetal         26
## 11  hits.11  covid vaccine infertility         44
## 12  hits.12       covid vaccine magnet         40
```

```
print(search.terms.sept)
```

```
##    var_name                     search num_states
## 1    hits.1              covid vaccine         51
## 2    hits.2      covid vaccine near me         51
## 3    hits.3         covid vaccine safe         49
## 4    hits.4  covid vaccine ingredients         49
## 5    hits.5     covid vaccine pregnant         49
## 6    hits.6      covid vaccine protect         42
## 7    hits.7 covid vaccine side effects         51
## 8    hits.8    covid vaccine microchip         10
## 9    hits.9          covid vaccine dna         41
## 10  hits.10        covid vaccine fetal         28
## 11  hits.11  covid vaccine infertility         42
## 12  hits.12       covid vaccine magnet         23
```

2. Vaccine Rates

```
# visualize doses administered over time for entire US

cdc.df.50 %>%
  group_by(date) %>%
  summarize(administered = sum(administered)) %>%
  ggplot() +
  geom_line(mapping = aes(x = date, y = administered))
```

Data for the vaccine and rates is acquired by using RSocrata to pull CDC COVID vaccine data through their API. After cleaning, two datasets are created for our vaccination dates of interest,vax.June21 and vax.Sept21.

```
## Create two datasets for our vaccination dates of interest

vax.June21 <- cdc.df.50 %>%
  filter(date == "2021-06-21")
vax.Sept21 <- cdc.df.50 %>%
  filter(date == "2021-09-21")
```

3. State-level Demographics

More data is needed to control for state-level demographic factors. Voter information was pulled to get the share of republican votes in 2020. In addition, median household income, percent of state population by age group, and race data was pulled and joined with the two CDC vaccine rate data. Finally, the three gtrends datasets hits.results.jan, hits.results.june, hits.results.sept are joined with either vax.June21 or vax.Sept21, depending on the dates the Trends are covering.

location	series_complete_pop_pct
West Virginia	40.2
Wyoming	40.8
Idaho	40.9
Alabama	41.6
Mississippi	42.5
North Dakota	43.4
Georgia	44.1
Louisiana	44.5
Tennessee	44.5
Arkansas	44.7
South Carolina	46.2
Oklahoma	46.6
Missouri	47.1
Indiana	47.8
Montana	47.9
North Carolina	48.8
Alaska	49.3
Ohio	49.7
Utah	49.8
Nevada	50.0
Texas	50.3
Kansas	50.4
Arizona	50.4
South Dakota	51.0
Kentucky	51.2
Michigan	51.8
Illinois	52.8

location	series_complete_pop_pct
Iowa	53.4
Nebraska	54.0
Wisconsin	55.7
Florida	56.3
Delaware	56.8
Pennsylvania	57.0
Hawaii	57.1
Minnesota	57.6
California	58.1
Colorado	58.7
District of Columbia	59.3
Virginia	59.8
Oregon	60.0
New Hampshire	61.1
New Mexico	62.3
Washington	62.6
New York	62.7
Maryland	63.4
New Jersey	63.6
Rhode Island	67.1
Massachusetts	67.4
Maine	67.8
Connecticut	68.0
Vermont	69.0

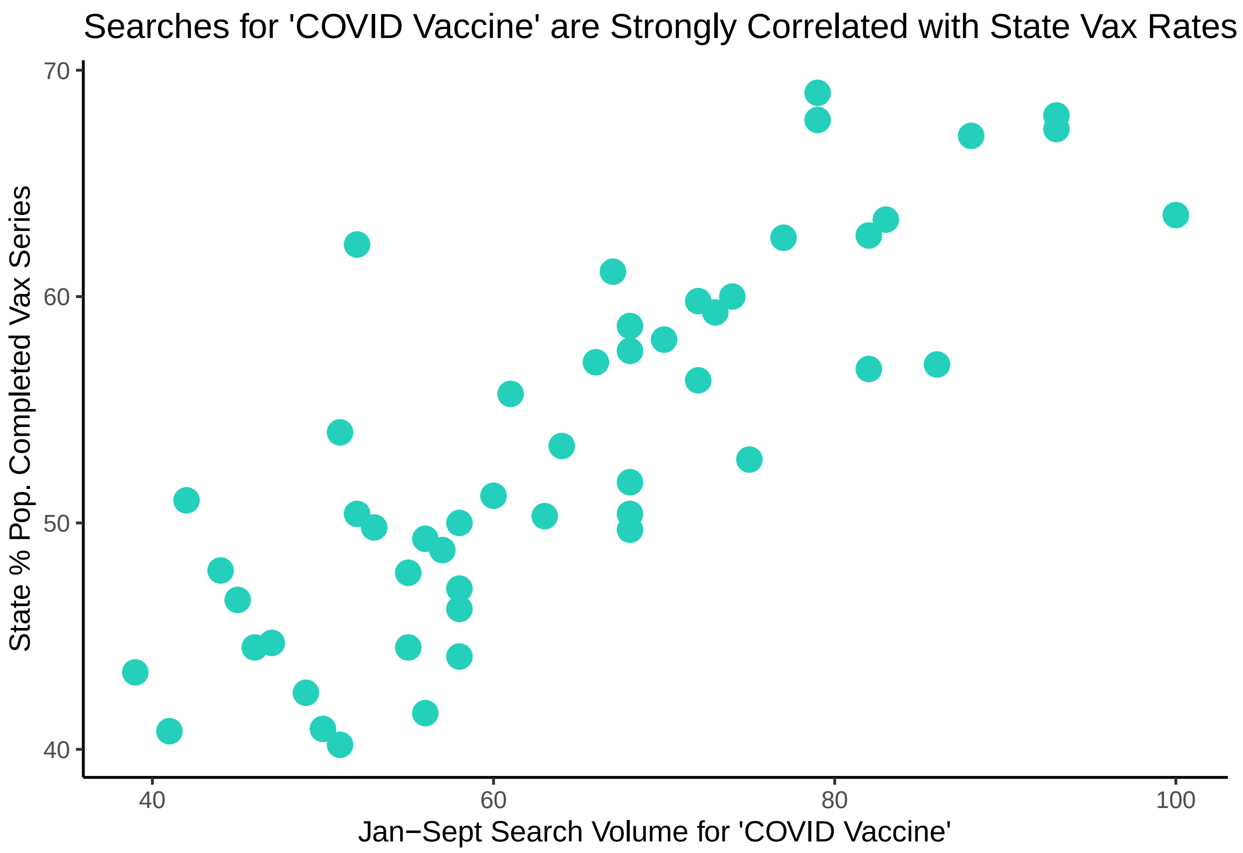
```
## This function joins the gtrends dataset with vaccine info dataset
join.gtrends.vaccine <- function (hits.results.month,vax.month){

  month.analysis <- vax.month %>%
      select(location,date, admin_per_100k, series_complete_pop_pct,
             pct.vote.rep, med.income, pct.18.to.24, pct.25.to.64, pct.65.over,
             pct.white, pct.black, pct.hispanic, pct.asian, pct.other.multiple) %>%
        full_join(hits.results.month, by = "location") %>%
        arrange(location)

  print(month.analysis)

}

Jan01.analysis <- join.gtrends.vaccine(hits.results.jan,vax.Sept21)
Sept21.analysis <- join.gtrends.vaccine(hits.results.sept,vax.Sept21)
June21.analysis <- join.gtrends.vaccine(hits.results.june,vax.June21)
```

Analysis

1. Correlation Analysis

```
## This function pulls the correlations for all 3 data sets
get.correlations <- function(month.analysis){
      #Loop for correlations for each search term
      j <- c("hits.1", "hits.2",
      "hits.3", "hits.4",
      "hits.5", "hits.6",
      "hits.7", "hits.8",
      "hits.9", "hits.10",
      "hits.11","hits.12")

      correlations <- data.frame(estimate=numeric(26), p.value=numeric(26))

      for(i in 15:ncol(month.analysis)){
        test <- cor.test(month.analysis[, i], month.analysis$series_complete_pop_pct)
        correlations$estimate[i] = test$estimate
        correlations$p.value[i] = test$p.value
      }

      correlations %<>%
        slice_tail(n=12) %>%
        cbind(j,k) %>%
        relocate(estimate, p.value,.after = k)

      correlations %<>% rename(var_name = j, search = k)

      print(correlations)

}

Jan01.correlations <- get.correlations(Jan01.analysis)
Sept21.correlations <- get.correlations(Sept21.analysis)
June21.correlations <- get.correlations(June21.analysis)
```

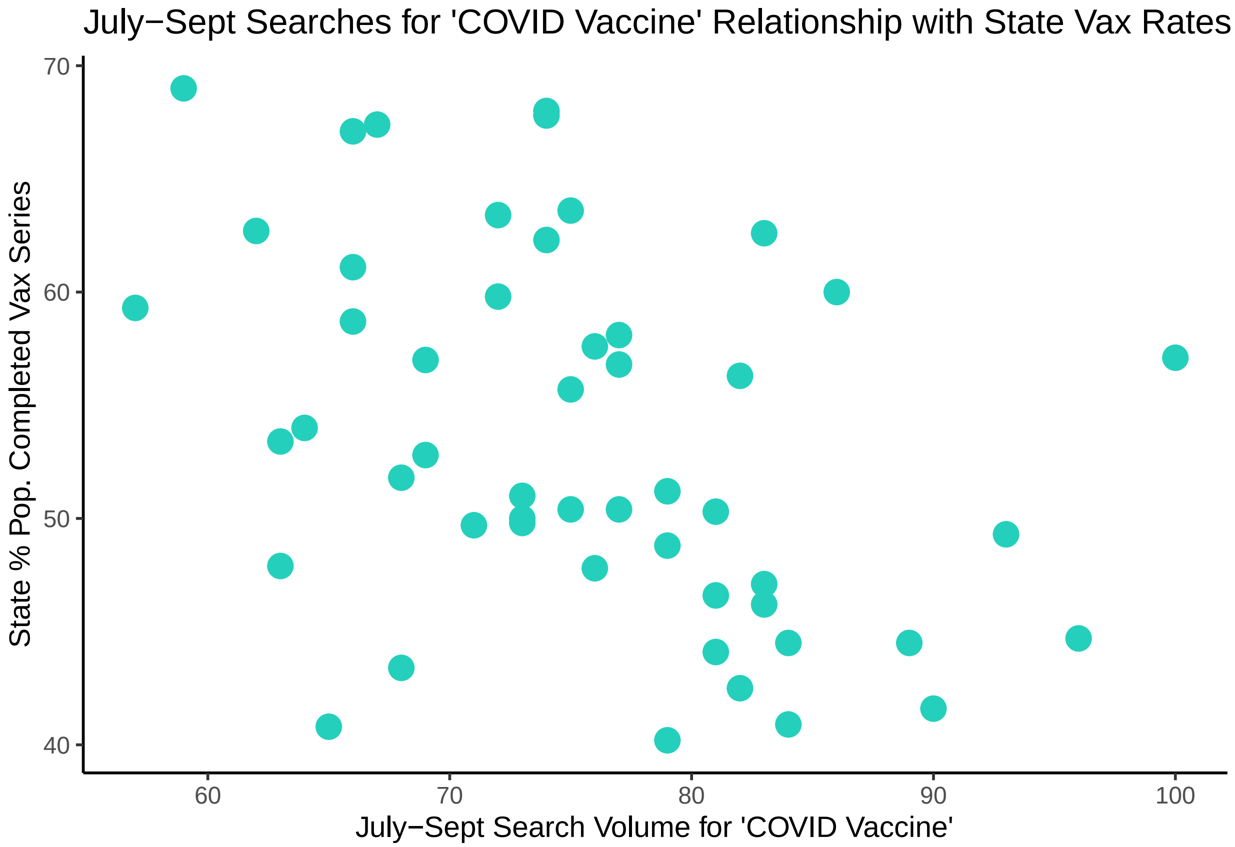
```
##Plotting of correlations

#Jan-Sept Searches
# using series_complete_pop_pct as measure for state vaccination rate
ggplot(Jan01.analysis) + geom_point(aes(hits.1, series_complete_pop_pct), color = '#24d0bc', size = 4) +
      labs(y = "State % Pop. Completed Vax Series", x = "Jan-Sept Search Volume for 'COVID Vaccine'") +
      ggtitle("Searches for 'COVID Vaccine' are Strongly Correlated with State Vax Rates") +
      theme_classic()
```



```
ggsave("covid.correlation.Jan.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
##Plotting of correlations continued

#July-September Searches
# using series_complete_pop_pct as measure for state vaccination rate
ggplot(Sept21.analysis) + geom_point(aes(hits.1, series_complete_pop_pct), color = '#24d0bc', size = 4)
      labs(y = "State % Pop. Completed Vax Series", x = "July-Sept Search Volume for 'COVID Vaccine'") +
      ggtitle("July-Sept Searches for 'COVID Vaccine' Relationship with State Vax Rates") +
      theme_classic()
```

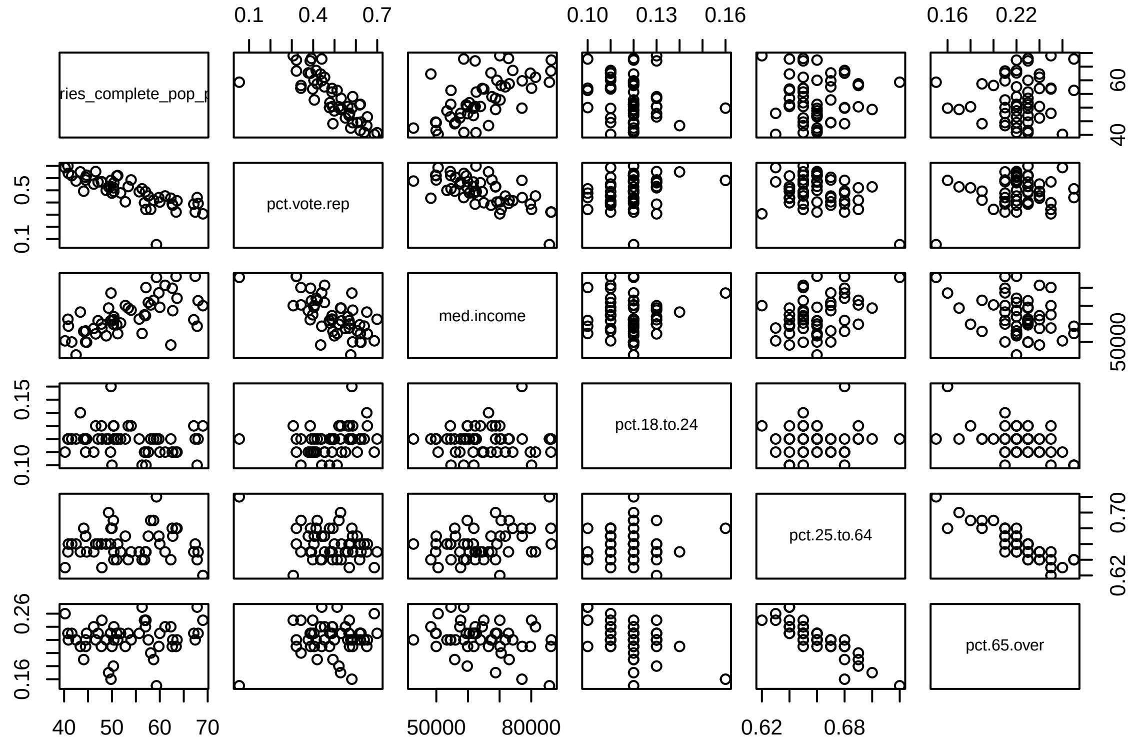


```r
ggsave("covid.correlation.Sept.png")
```

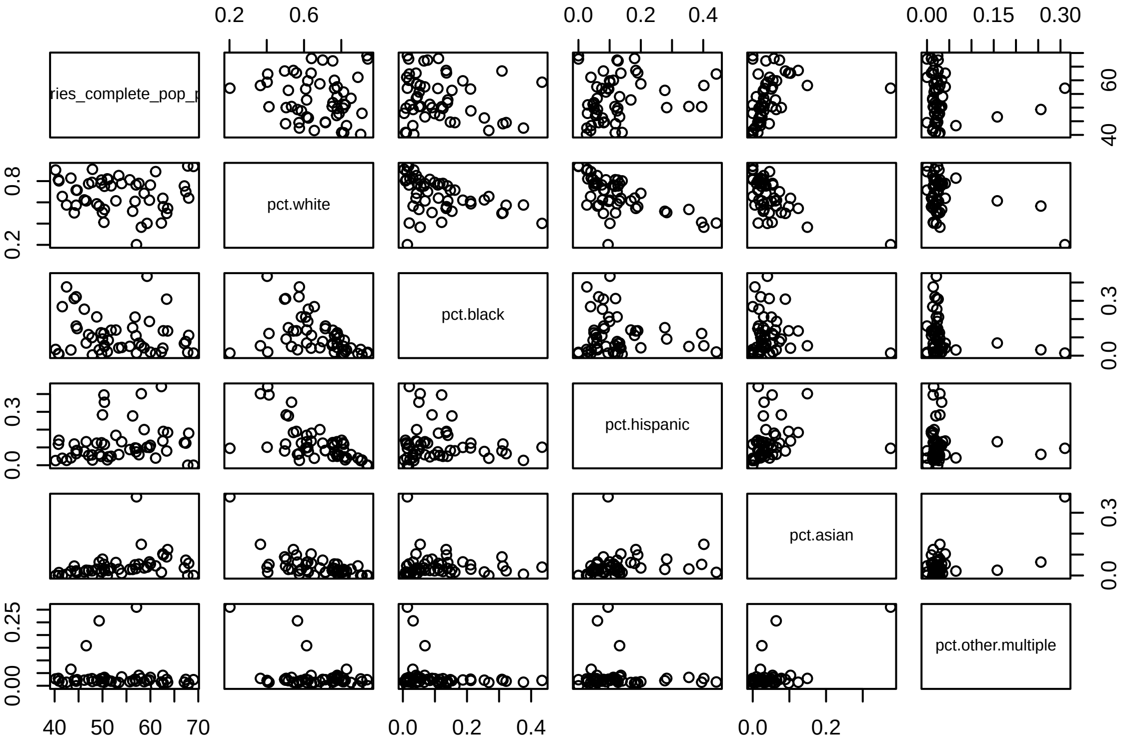
```
## Saving 6.5 x 4.5 in image
```

2. Regression Analysis

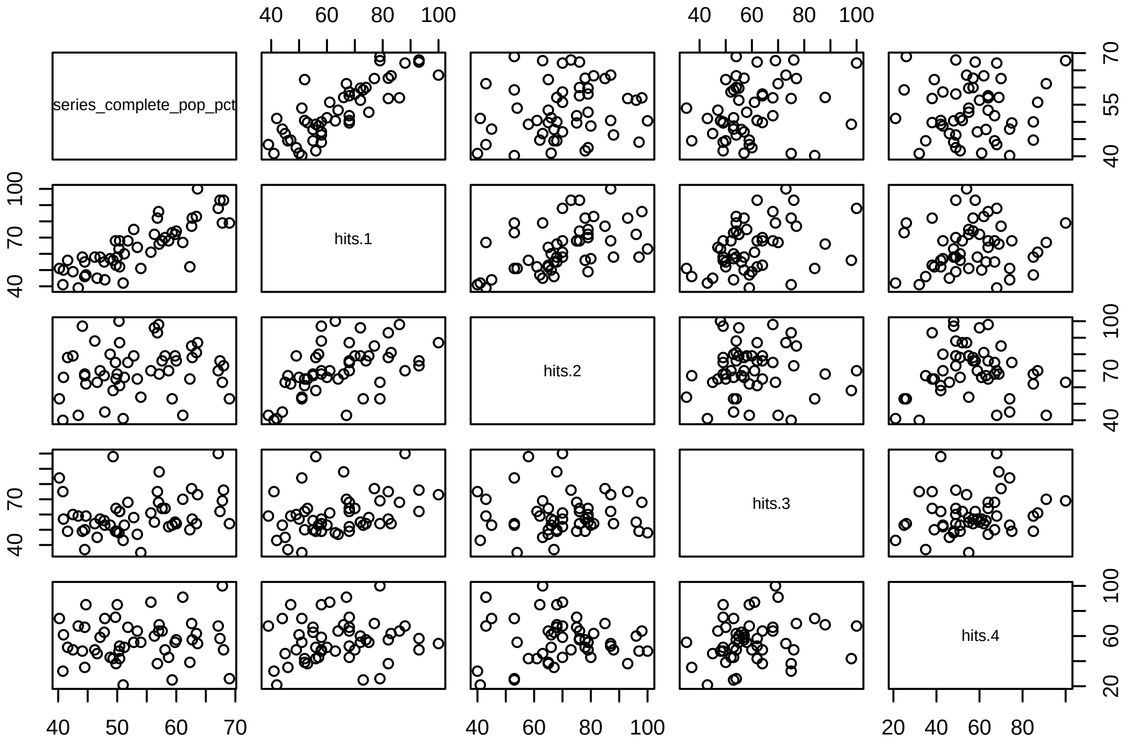
```r
# Plotting and regression analysis

Jan01.analysis %>% select(series_complete_pop_pct, pct.vote.rep,med.income,
                          pct.18.to.24,pct.25.to.64,pct.65.over) %>%
                          plot()
```



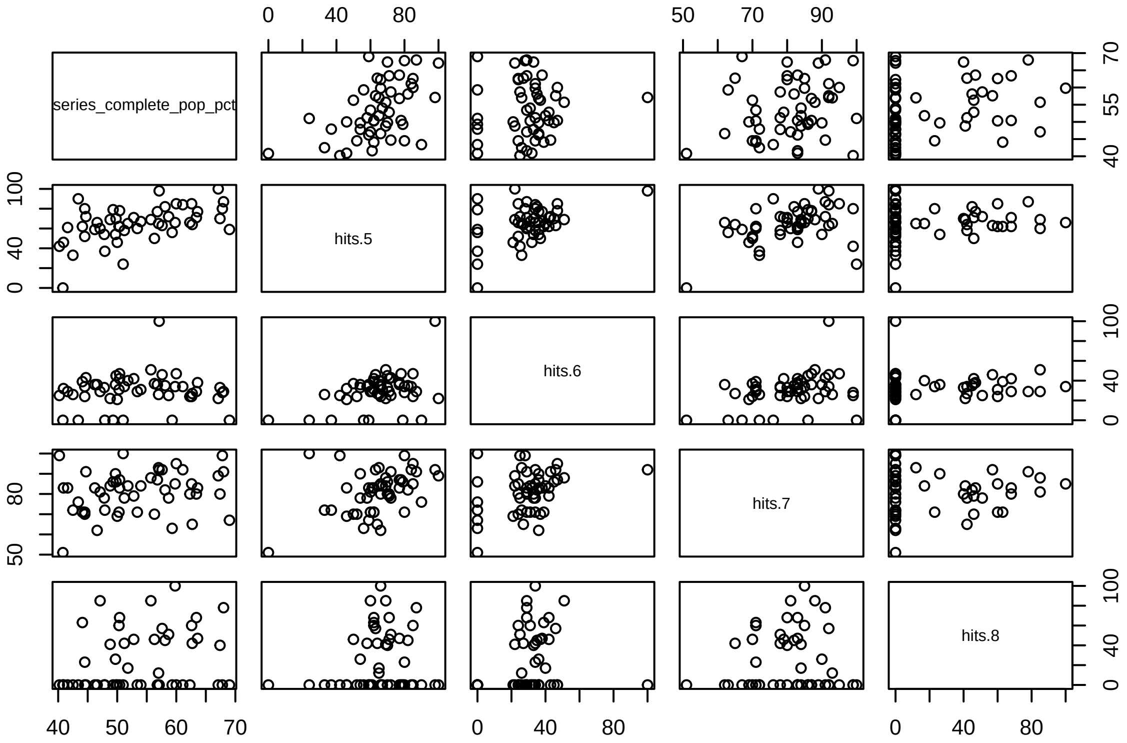
```
##percent republican and median income seem to have a linear relationship with series_complete_pop_pct;

Jan01.analysis %>% select(series_complete_pop_pct, pct.white,pct.black,pct.hispanic,pct.asian,pct.other
                         plot()
```



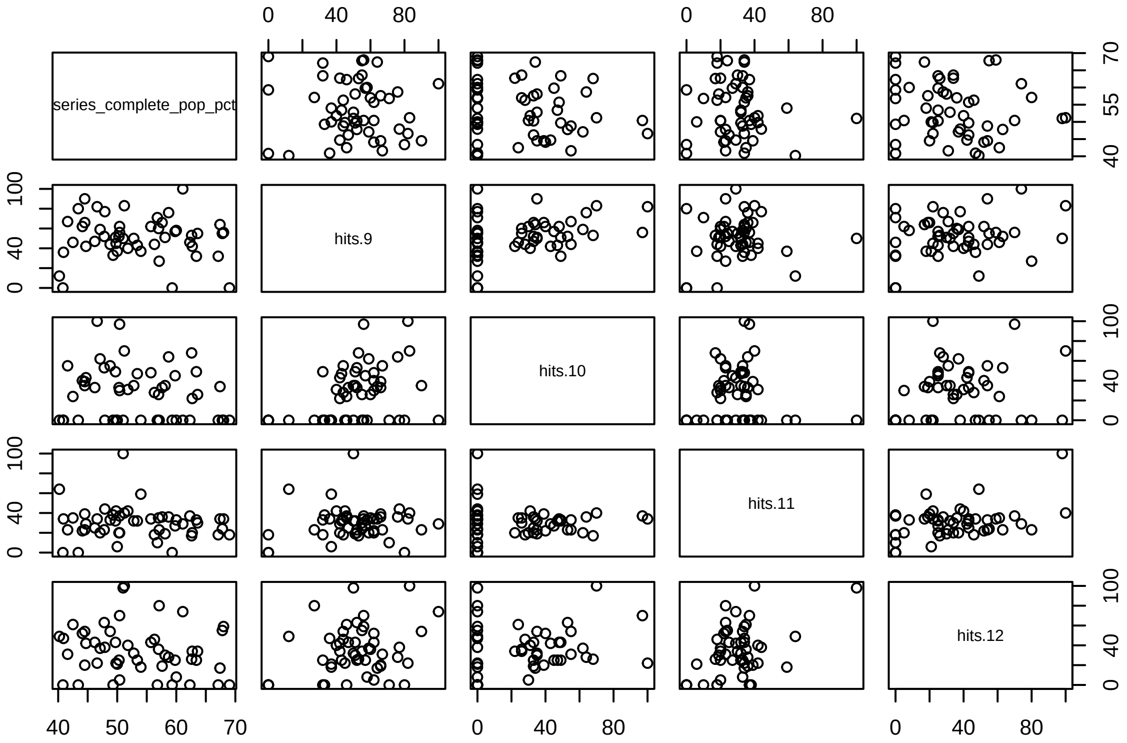
```
##None of the race variables seem to be related to vax rates


Jan01.analysis %>% select(series_complete_pop_pct, hits.1,hits.2,hits.3,hits.4) %>%
                    plot()
```



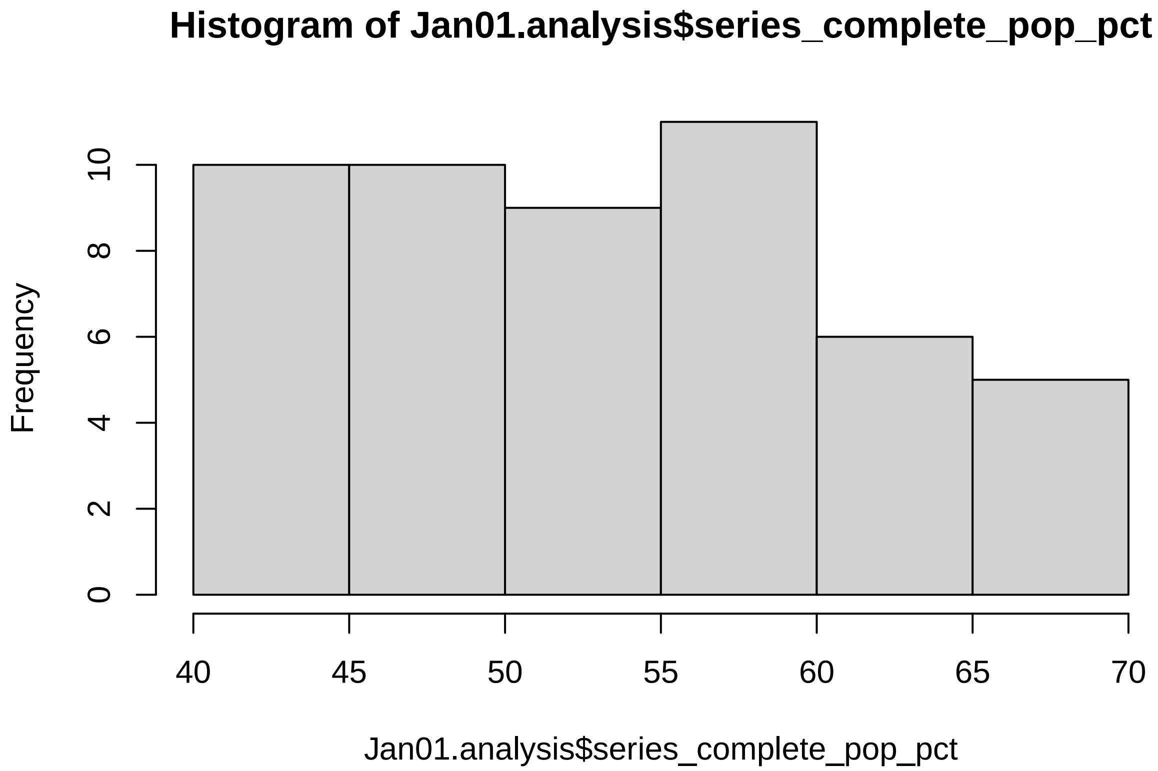
```
#Hits1 is related; other plots are widely scattered

Jan01.analysis %>% select(series_complete_pop_pct, hits.5,hits.6,hits.7,hits.8) %>%
                      plot()
```



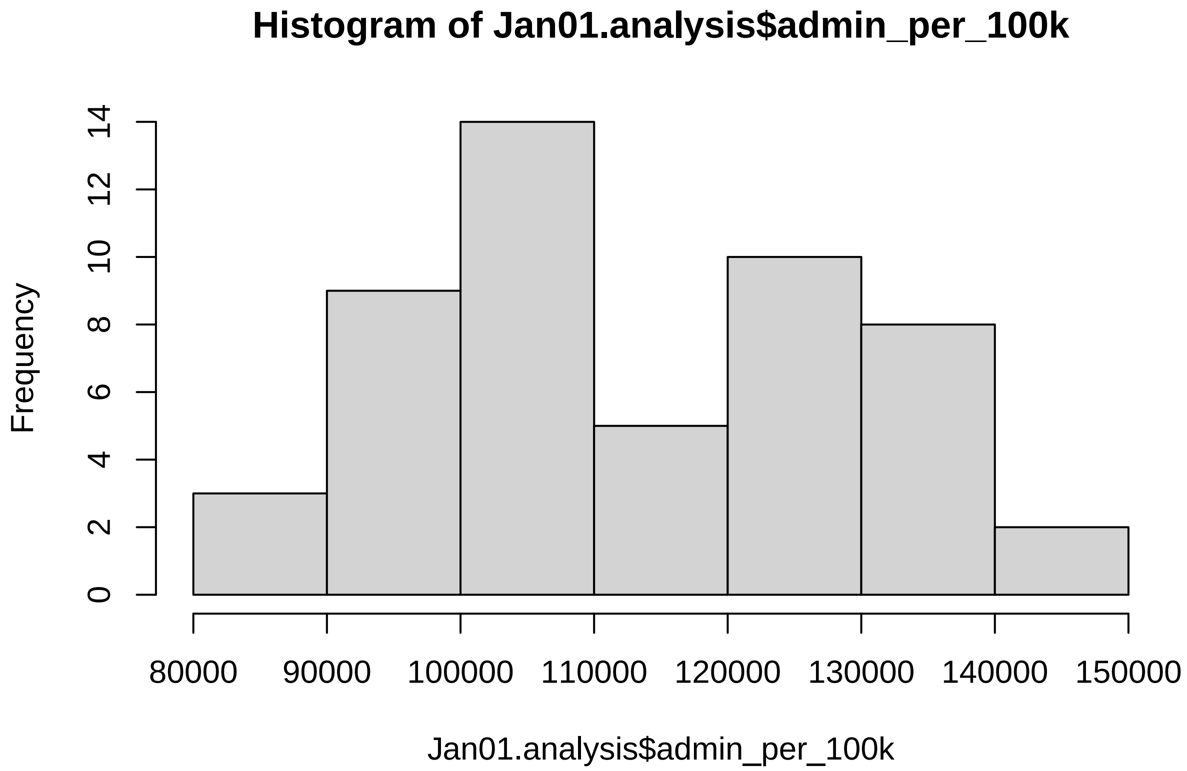
```
#Hits5 has some relationship; others not so much

Jan01.analysis %>% select(series_complete_pop_pct, hits.9,hits.10,hits.11,hits.12) %>%
                          plot()
```



```
#No strong relationships here


##histogram of outcome variable

hist(Jan01.analysis$series_complete_pop_pct)
```



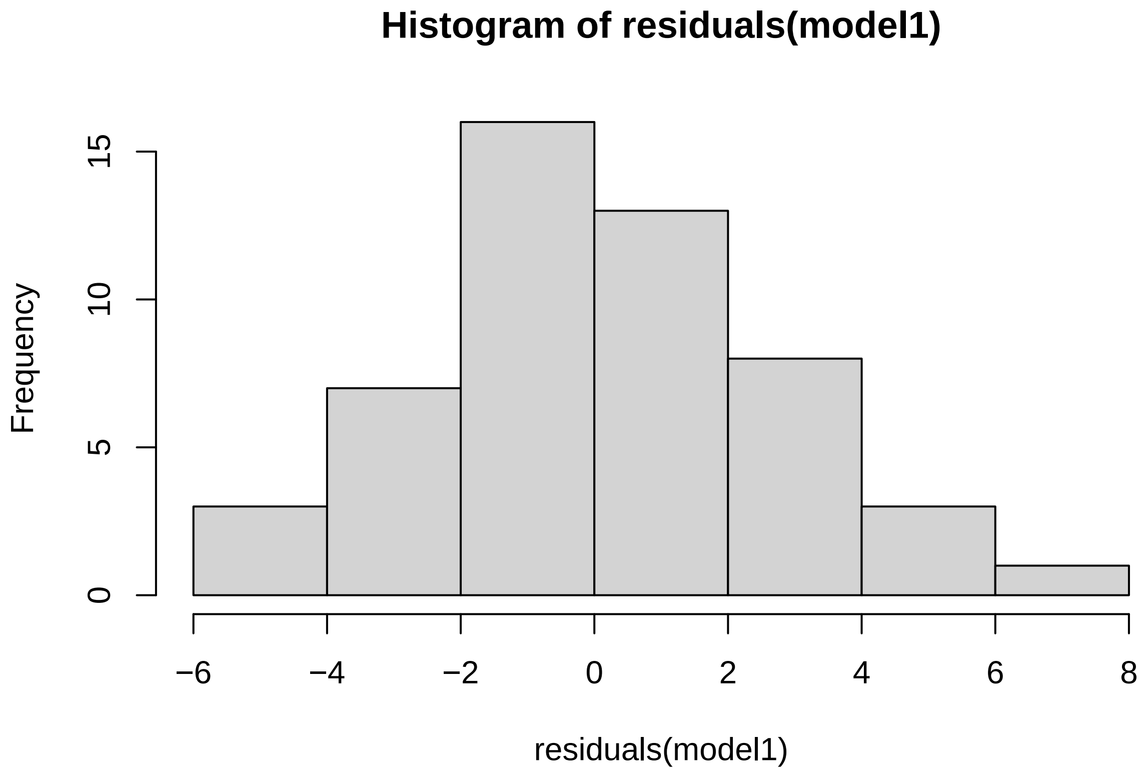
```
hist(Jan01.analysis$admin_per_100k)
```



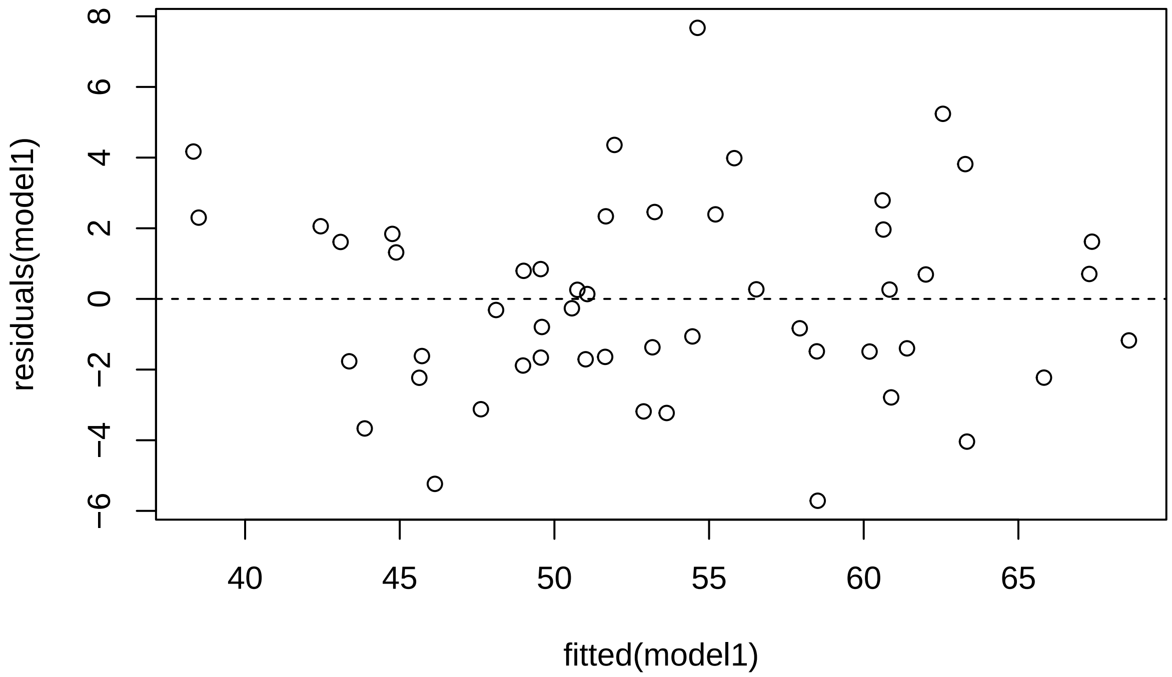
```
##Linear model

model1 <- lm(series_complete_pop_pct ~ pct.vote.rep + pct.white + pct.black + hits.1 + hits.2 + hits.3 +

summary(model1)
```

```
##
## Call:
## lm(formula = series_complete_pop_pct ~ pct.vote.rep + pct.white +
##     pct.black + hits.1 + hits.2 + hits.3 + hits.4 + hits.5 +
##     hits.6 + hits.7 + hits.8 + hits.9 + hits.10 + hits.11 + hits.12,
##     data = Jan01.analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7127 -1.6852 -0.2653  1.9016  7.6727
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.577e+01  6.716e+00   8.303 8.67e-10 ***
## pct.vote.rep -2.851e+01  9.199e+00  -3.099 0.003814 **
## pct.white    -1.523e+00  5.263e+00  -0.289 0.774004
## pct.black    -2.223e+01  6.187e+00  -3.593 0.000995 ***
## hits.1        3.474e-01  8.097e-02   4.290 0.000134 ***
## hits.2       -8.337e-02  6.730e-02  -1.239 0.223675
## hits.3       -8.722e-02  5.140e-02  -1.697 0.098609 .
## hits.4       -7.262e-03  3.543e-02  -0.205 0.838762
## hits.5        5.730e-02  4.619e-02   1.241 0.222954
## hits.6       -4.797e-02  4.233e-02  -1.133 0.264723
## hits.7        4.173e-04  8.533e-02   0.005 0.996126
## hits.8        5.368e-03  2.026e-02   0.265 0.792645
## hits.9        1.086e-02  3.036e-02   0.358 0.722698
## hits.10      -4.966e-03  2.417e-02  -0.205 0.838416
## hits.11       3.135e-02  4.718e-02   0.664 0.510760
## hits.12       2.003e-02  2.701e-02   0.742 0.463288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.261 on 35 degrees of freedom
## Multiple R-squared:  0.8867, Adjusted R-squared:  0.8381
## F-statistic: 18.26 on 15 and 35 DF,  p-value: 2.981e-12
```

```
#Check that residuals are normally distributed

hist(residuals(model1))
```



```
#Check for homoskedasticity in residual variances (looks ok)

plot(fitted(model1), residuals(model1))
abline(h = 0, lty = 2)
```



```
#Linear model with interaction
#When adding interaction between hits.1 and % who voted republican, the main effects and the interactio

model2 <- lm(series_complete_pop_pct ~ pct.vote.rep + pct.black + hits.1  + hits.1*pct.vote.rep, data =

summary(model2)
```

```
##
## Call:
## lm(formula = series_complete_pop_pct ~ pct.vote.rep + pct.black +
##     hits.1 + hits.1 * pct.vote.rep, data = Jan01.analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5878 -2.1744 -0.2679  2.5307  7.9335
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           52.3502    12.8896   4.061 0.000188 ***
## pct.vote.rep         -25.7923    23.4115  -1.102 0.276326
## pct.black            -25.3249     4.7632  -5.317 3.01e-06 ***
## hits.1                 0.3237     0.1778   1.821 0.075133 .
## pct.vote.rep:hits.1  -0.1366     0.3496  -0.391 0.697849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.381 on 46 degrees of freedom
## Multiple R-squared:   0.84,  Adjusted R-squared:  0.826
## F-statistic: 60.36 on 4 and 46 DF,  p-value: < 2.2e-16
```

```
save.image(file = "shared_work_space.RData")
```

References