```r
 1  # ---------------------------------Documentation --------------------------------
    ----------------
 2  #
 3  # By:        Hannah Fresques, ProPublica
 4  # Date:      March 25, 2019
 5  # Project:   Gutting the IRS
 6  # Purpose:   Translate excel data to JSON.
 7  #            Used for an interactive map (https://projects.propublica.org/graphics/eitc-
    audit)
 8  #_____
    _____

 9
10
11  # setup -------------------------------------------------------------
12
13  library(readxl)
14  library(readr)
15  library(dplyr)
16  library(janitor)
17  library(purrr)
18  library(jsonlite)
19  library(stringr)
20
21  # read in data ----------------------------------------------------
22
23  # estimated exams
24  counties <- read_xlsx(
25    path="data/raw/Bloomquist - Regional Bias in IRS Audit  Selection Data.xlsx",
26    sheet="estimatedExams",
27    col_types="text"
28  )
29
30
31  # filings
32  years <- 2012:2015
33
34  read_filings <- function(year){
35    df <- read_xlsx(
36      path=paste0("data/raw/County-",year,".xlsx"),
37      skip=6,
38      col_names=FALSE,
39      col_types="text"
40    )
41    df <- df[,c(1:5)]
42    colnames(df) <-
    c("State_FIPS_code","State","County_FIPS_code","County_name","Number_of_returns")
43    df <- df %>%
44      mutate(
45        year=year
46      )
47  }
48
49  filings <- years %>% map(read_filings) %>% bind_rows()
50
51
52
```

```r
53   # clean up filings data ----------------------------------------------------
54
55   filings %>% filter(is.na(County_FIPS_code)) %>% print(n=Inf)
56   # these are all notes from the bottom of files.
57   # drop them.
58
59   filings %>% filter(County_FIPS_code=="0") %>% count(County_name) %>% print(n=Inf)
60   # Except for DC, these are all state and country-wide totals.
61   # drop them.
62
63   filings2 <- filings %>%
64     mutate(
65       County_FIPS_code=case_when(
66         State=="DC"~"001",
67         TRUE~County_FIPS_code
68       )
69     ) %>%
70     filter(!is.na(County_FIPS_code) & County_FIPS_code!="0") %>%
71     mutate(
72       fips = paste0(
73         str_pad(State_FIPS_code , width=2, pad="0", side="left"),
74         str_pad(County_FIPS_code, width=3, pad="0", side="left")
75       )
76     )
77
78   # wade became kusilvak
79   # shannon became oglala
80
81   filings3 <- filings2 %>%
82     mutate(
83       County_name=case_when(
84         # named LaSalle Parish some years, La Salle Parish others. Just standardizing.
85         fips=="22059"~"La Salle Parish",
86         # use new names.
87         fips=="02270"~"Kusilvak Census Area",
88         fips=="46113"~"Oglala County",
89         TRUE~County_name
90       ),
91       fips=case_when(
92         # needs a real fips
93         State=="DC"~"11001",
94         # use old fips codes (because that's what the javascript mapping library expects)
95         fips=="02158"~"02270",
96         fips=="46102"~"46113",
97         TRUE~fips
98       )
99     ) %>%
100    group_by(fips,State,County_name) %>%
101    summarize(
102      years=n(),
103      Number_of_returns = sum(as.numeric(Number_of_returns))
104    )
105
106
107  # clean up counties data -----------------------------------------------------
108
109  counties <- counties %>%
```

```r
110    clean_names() %>%
111    mutate(
112      # fips codes were missing leading zeros on the excel file
113      fips = str_pad(fips, width=5, pad="0", side="left")
114    )
115
116  counties %>% filter(fips %in% c("02270","46113","02158","46102"))
117  # this file uses the old fips codes and old county names for the SD and AK counties.
118
119  counties <- counties %>%
120    mutate(
121      county=case_when(
122        # use new names.
123        fips=="02270"~"Kusilvak Census Area",
124        fips=="46113"~"Oglala County",
125        TRUE~county
126      )
127      # keep old fips codes (because that's what the javascript mapping library expects)
128    )
129
130
131  # put data together ----------------------------------------------------
132
133  counties2 <- counties %>%
134    left_join(
135      filings3,
136      by="fips"
137    )
138
139  counties2 %>% count(state,State) %>% print(n=Inf)
140
141
142  # check and clean data -------------------------------------------------
143
144
145  counties3 <- counties2 %>%
146    mutate(
147      name = paste0(County_name,", ",state),
148      estimated_exams = as.numeric(estimated_exams),
149      Number_of_returns = as.numeric(Number_of_returns),
150      audit_rate = (estimated_exams / Number_of_returns)*1000
151    )
152
153
154  # national -------------------------------------------------------------
155
156  national <- counties3 %>%
157    summarize(
158      estimated_exams=sum(estimated_exams),
159      Number_of_returns=sum(Number_of_returns)
160    )
161
162  # estimated_exams Number_of_returns
163  #         4506034          586148520
164
165  national_average <- (national$estimated_exams / national$Number_of_returns) * 1000
166  # national rate is 7.687529 per 1,000 filings
```

```
167
168  # is anyone right on the average?
169  counties3 %>% filter(audit_rate==national_average) # none
170  counties3 %>% filter(audit_rate>national_average) %>% nrow() # 1514 above average
171  counties3 %>% filter(audit_rate<national_average) %>% nrow() # 1627 below average
172
173
174
175  # some checks ------------------------------------------------------------
176
177  library(ggplot2)
178  counties3 %>%
179    ggplot(aes(x=audit_rate)) +
180    geom_histogram()
181
182  # this distribution will not map well on a linear color scale of 0-12 per 1,000
     filings.
183  # make a new value of audit_rate that has a floor and ceiling
184  counties3 <- counties3 %>%
185    mutate(audit_rate_trunk=case_when(
186      audit_rate<= 6 ~ 6,
187      audit_rate>= 11 ~ 11,
188      TRUE ~ audit_rate
189    ))
190
191  counties3 %>%
192    ggplot(aes(x=audit_rate_trunk)) +
193    geom_histogram() +
194    geom_vline(xintercept=national_average)
195
196
197
198
199
200  # save data -------------------------------------------------------------
201
202  counties4 <- counties3 %>%
203    select(fips,name,state,Number_of_returns,estimated_exams,audit_rate,audit_rate_trunk)
204
205  # save to csv
206  # write_csv(counties4, "data/cleaned/auditsData_2019.04.03.csv")
207
208  # save to json
209  myJSON <- counties4 %>%
210    transpose() %>%
211    set_names(counties3$fips) %>%
212    toJSON(auto_unbox = TRUE)
213  head(myJSON)
214  # save the text string as a json file
215  # fileConn<-file("data/cleaned/auditsData_2019.04.03.json")
216  # writeLines(myJSON, fileConn)
217  # close(fileConn)
218
```