7/5/2019 bailey_code.R

```
2
   ############ EJUF Code to Evaluate Landlord Data for Bailey ###########
   3
 4
   ## Trying to figure out who are landlords by who owes multiple properties, and which
   properties and landlords have most reported code violations
 6
 7
   setwd("/Users/joanmeiners/Dropbox/Fall 2017/Environmental
   Journalism/Bailey Landlords EJUF/")
8
9
   library(dplyr)
   library(plyr)
10
11
12 # load initial dataset from bailey of addresses and owners, just for zip 32641 (see end of
   script for code sorting all addresses by number of owners and violations)
   bailey = read.csv("Energy-Poverty 32641 homes.csv")
13
   levels(bailey$OWNERNME1) # how many different property owners are there
15 | dim(bailev)
16 landlords = dplyr::count(bailey, OWNERNME1, sort = TRUE) # count properties per owner and
   sort owners by how many properties they own
17 landlords = subset(landlords, n>1) # only keep owners that have more than one property =
   likely landlords
   View(landlords)
18
19
20 # calculate the total cost of utilities per owner
21 by owner = group by(bailey, OWNERNME1)
22 utilities = dplyr::summarise(by owner, cost = sum(Unit.Utilities.Cost))
23 View(utilities)
24
   # combine datasets on who the likely landlords are with how many properties they own and th
25
   combined utility cost at those properties (only for zip code 32641)
ownercost = plyr::join(landlords, utilities, by = 'OWNERNME1')
27 View(ownercost)
   colnames(ownercost)[colnames(ownercost)=="n"] <- "num properties" # rename column</pre>
28
   ownercost$cost per property = ownercost$cost / ownercost$num properties # add column of
   average utility cost per property for each owner
30
31 # save dataset to file
32 write.csv(ownercost, file = "owner_cost.csv", row.names=FALSE)
33
34
35 | ## Now Looking at Landlord data to find out which addresess have had the most complainst
   against them
   # load data on reported code violations
36
37 violations = read.csv("Bailey_landlord.csv", header= TRUE)
38
   dim(violations)
39 View(violations)
40
41 # code to group the reported code violations by address, commented out because saved result
   is loaded from repository in next step
42 # addresses = violations %>%
       dplyr::group by(PrimaryParty, Address) %>%
       dplyr::summarise(viol_per_address = n())
44
45 # addresses = addresses[order(-addresses$viol per address),] # sort in order of decreasing
   number of code violations
46 # View(addresses)
```

7/5/2019 bailey_code.R

```
47
   # write.csv(addresses, "worst addresses.csv", row.names = FALSE) # save to file
48
   # load file created in commented out code above for addresses with the most code violations
49
    and who ownes them
   addresses = read.csv("worst addresses.csv", header = TRUE)
50
51
52
   # reformat addresses and pull in zip code information from another dataset
   adds = tidyr::separate(addresses, Address, into = c("Number", "Street"), sep = "\\ ", extra
    = "merge") # number coded as a five digit with leading zeros, separate out and classify as
    numeric to remove differing numbers of leading zeros from address number
   adds$Number = as.numeric(adds$Number)
54
   adds$ADDRESS = paste(adds$Number, adds$Street, sep=" ") # paste address number and street
   fields back together
   adds$viols = adds$viol_per_address # rename column
56
   adds = subset(adds, select = c("ADDRESS", "viols"))
   adds$ADDRESS = trimws(adds$ADDRESS) # remove extra whitespace from address field
58
59
   dim(adds)
60
61 # pull in cleaned dataset on property values from Hal Knowles
   value = read.csv("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/value.csv",
    header = TRUE)
   zipviol = plyr::join(adds, value, by = "ADDRESS") # join property value to code violations
    dataset by address
   zipviol = subset(zipviol, POSTAL != "NA" & CNTASSDVALUE > 20000, select = c("ADDRESS",
64
    "POSTAL", "viols", "CNTASSDVALUE")) # filter out any addresses without a zip code and those
    valued at below $20,000 as likely not a residence
   zipviol$viols = as.numeric(zipviol$viols)
   zipviol$POSTAL = as.factor(zipviol$viols)
66
67
   # look for trends in violations per zip code
68
   hist(zipviol$viols) # need to transform
69
70 hist(log10(zipviol$viols)) # zero-inflated, probably passable for this simple analysis --
    checked and still significant when add 1 to values or restrict to addresses with multiple
    code violations, but this allows us to still look at those addresses with only one code
    violation for comparison along property value gradient
71 hist(log10(zipviol$CNTASSDVALUE)) # normal
72 violzip = glm(log10(viols) ~ log10(CNTASSDVALUE), data = zipviol)
73
   summary(violzip)
74
   violzip
75
76 | # plot number of code violations per address against the property value of address
   quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on line
    below instead of this one
   # tiff(filename = "Violations value.tiff", units = "in", compression = "lzw", res = 300,
    width = 12, height = 6)
79
   ggplot(aes(y = viols, x = CNTASSDVALUE), data = zipviol) +
      scale_x_log10(breaks = c(2000000 ,200000, 20000), labels = function(x) paste0("$",
80
    scales::comma(x))) +
      geom_point(color = "grey") +
81
82
      xlab("County-assessed Property Value (USD)") + ylab("Number of code violations per
    address") +
      theme(axis.title = element_text(family = "Trebuchet MS", color="#666666", face="bold",
83
    size=15)) +
     theme(axis.text = element text(family = "Trebuchet MS", color="#666666", face="bold",
84
    size=12)) +
     geom_smooth(method = "lm", se=FALSE, color="darkgreen")
85
   # dev.off()
```