

```

1 #####
2 ##### EJUF Code to Evaluate Landlord Data for Bailey #####
3 ##### Joan Meiners 2017 #####
4
5 ## 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)
10 library(plyr)
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)
13 bailey = read.csv("Energy-Poverty 32641 homes.csv")
14 levels(bailey$OWNERNME1) # how many different property owners are there
15 dim(bailey)
16 landlords = dplyr::count(bailey, OWNERNME1, sort = TRUE) # count properties per owner and
   sort owners by how many properties they own
17 landlords = subset(landlords, n>1) # only keep owners that have more than one property =
   likely landlords
18 View(landlords)
19
20 # calculate the total cost of utilities per owner
21 by_owner = group_by(bailey, OWNERNME1)
22 utilities = dplyr::summarise(by_owner, cost = sum(Unit.Utilities.Cost))
23 View(utilities)
24
25 # combine datasets on who the likely Landlords are with how many properties they own and th
   combined utility cost at those properties (only for zip code 32641)
26 ownercost = plyr::join(landlords, utilities, by = 'OWNERNME1')
27 View(ownercost)
28 colnames(ownercost)[colnames(ownercost)=="n"] <- "num_properties" # rename column
29 ownercost$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 addresseess have had the most complainst
   against them
36 # Load data on reported code violations
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 %>%
43 #   dplyr::group_by(PrimaryParty, Address) %>%
44 #   dplyr::summarise(viol_per_address = n())
45 # addresses = addresses[order(-addresses$viol_per_address),] # sort in order of decreasing
   number of code violations
46 # View(addresses)

```

```

47 # write.csv(addresses, "worst_addresses.csv", row.names = FALSE) # save to file
48
49 # Load file created in commented out code above for addresses with the most code violations
  and who owns them
50 addresses = read.csv("worst_addresses.csv", header = TRUE)
51
52 # reformat addresses and pull in zip code information from another dataset
53 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
54 adds$Number = as.numeric(adds$Number)
55 adds$ADDRESS = paste(adds$Number, adds$Street, sep=" ") # paste address number and street
  fields back together
56 adds$viols = adds$viol_per_address # rename column
57 adds = subset(adds, select = c("ADDRESS", "viols"))
58 adds$ADDRESS = trimws(adds$ADDRESS) # remove extra whitespace from address field
59 dim(adds)
60
61 # pull in cleaned dataset on property values from Hal Knowles
62 value = read.csv("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/value.csv",
  header = TRUE)
63 zipviol = plyr::join(adds, value, by = "ADDRESS") # join property value to code violations
  dataset by address
64 zipviol = subset(zipviol, POSTAL != "NA" & CNTASSDVALUE > 20000, select = c("ADDRESS",
  "POSTAL", "viols", "CNTASSDVALUE")) # filter out any addresses without a zip code and those
  valued at below $20,000 as likely not a residence
65 zipviol$viols = as.numeric(zipviol$viols)
66 zipviol$POSTAL = as.factor(zipviol$viols)
67
68 # Look for trends in violations per zip code
69 hist(zipviol$viols) # need to transform
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
77 quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on line
  below instead of this one
78 # tiff(filename = "Violations_value.tiff", units = "in", compression = "lzw", res = 300,
  width = 12, height = 6)
79 ggplot(aes(y = viols, x = CNTASSDVALUE), data = zipviol) +
80   scale_x_log10(breaks = c(2000000, 200000, 20000), labels = function(x) paste0("$",
  scales::comma(x))) +
81   geom_point(color = "grey") +
82   xlab("County-assessed Property Value (USD)") + ylab("Number of code violations per
  address") +
83   theme(axis.title = element_text(family = "Trebuchet MS", color="#666666", face="bold",
  size=15)) +
84   theme(axis.text = element_text(family = "Trebuchet MS", color="#666666", face="bold",
  size=12)) +
85   geom_smooth(method = "lm", se=FALSE, color="darkgreen")
86 # dev.off()

```

```
87
88 # arrange data by owners with most addresses
89 owners = addresses %>%
90   dplyr::group_by(PrimaryParty) %>%
91   dplyr::summarize(addresses_per_owner = n())
92 owners = owners[order(-owners$addresses_per_owner),]
93 View(owners)
94
95 # arrange data by owners with most code violations
96 viol = subset(addresses[, c("PrimaryParty", "viol_per_address")])
97 viol = viol %>%
98   dplyr::group_by(PrimaryParty) %>%
99   dplyr::summarize(violations_per_owner = sum(viol_per_address))
100 viol = viol[order(-viol$violations_per_owner),]
101 View(viol)
102
103 # combine datasets on number of properties and number of code violations by owner
104 owner_violations = plyr::join(viol, owners, by = "PrimaryParty")
105 owner_violations$avg_owner_violations_per_address = owner_violations$violations_per_owner /
106   owner_violations$addresses_per_owner
107 owner_violations = subset(owner_violations, addresses_per_owner > 1)
108 owner_violations = owner_violations[order(-
109   owner_violations$avg_owner_violations_per_address),]
110 View(owner_violations)
111
112 # write out file of most code-violating owners
113 write.csv(owner_violations, "owner_violations", row.names = FALSE)
```