

```

1 #####
2 ##### Sort, join & analyze GRU data on Power Outages after Irma #####
3 ##### Joan Meiners 2017 #####
4
5 # Load libraries
6 library(dplyr)
7 library(plyr)
8 library(tidyr)
9 library(ggplot2)
10 library(lubridate)
11 library(MASS)
12
13 # Load linear model function for ggplot plotting
14 lm_eqn = function(m) {
15
16   l <- list(a = format(coef(m)[1], digits = 2),
17             b = format(abs(coef(m)[2]), digits = 2),
18             r2 = format(summary(m)$r.squared, digits = 3));
19
20   if (coef(m)[2] >= 0) {
21     eq <- substitute(italic(y) == a + b %.% italic(x)*", "~italic(r)^2~"="~r2,l)
22   } else {
23     eq <- substitute(italic(y) == a - b %.% italic(x)*", "~italic(r)^2~"="~r2,l)
24   }
25
26   as.character(as.expression(eq));
27 }
28
29 #####
30 setwd("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/Energy Burden
Project Files")
31
32 # Load power data from GRU -- addresses that lost power and duration of outage
33 power = read.csv("GRU_power.csv", header = TRUE)
34 power$ADDRESS = trimws(power$ADDRESS) # remove extra whitespaces in address field
35 power = tidyr::separate(power, ADDRESS, into = c("ADDRESS", "extraADD"), sep = "\\,") #
separate out extraneous address fields that won't join with other datasets
36 power$POSTAL = as.character(strtrim(power$POSTAL, width = 5)) # limit POSTAL field to 5
characters
37 unique(power$POSTAL) # find out which POSTAL codes are included in data
38 power$POWER.DURATION = as.numeric(as.duration(hm(power$POWER.DURATION))) # convert
power duration to a numeric field rather than hr:min format given
39
40 # calculate correct power outage time difference (GRU calculation did not add in
multiple days of power outage)
41 power = tidyr::separate(power, POWER.OUT.TIME, into = c("DAY.OUT", "HOUR.OUT"), sep =
"\\ ") # restructure GRU data to calculate days out of power
42 power = tidyr::separate(power, POWER.RESTORE.TIME, into = c("DAY.RESTORE",
"HOUR.RESTORE"), sep = "\\ ")
43 power$DAY.OUT = as.Date(power$DAY.OUT, "%m/%d/%y") # change date format
44 power$DAY.RESTORE = as.Date(power$DAY.RESTORE, "%m/%d/%y")
45
46 power$DURATION.DAYS = as.numeric(difftime(power$DAY.RESTORE, power$DAY.OUT),
units="days") # calculate number of days out of power
47 power$POWER.DURATION = power$POWER.DURATION / (60 * 60 * 24) # convert minutes to days

```

```

48 power$CORRECT.DAYS = power$POWER.DURATION + power$DURATION.DAYS # create new column
   with calculated days out of power added to GRUs calculated hour:min out of power
49 power$CORRECT.DAYS = as.numeric(power$CORRECT.DAYS) # make sure field is numeric
50 power = subset(power, select = c("ADDRESS", "CORRECT.DAYS", "POSTAL"))
51 power = power[!duplicated(power$ADDRESS),] # eliminate duplicated addresses
52 dim(power)
53
54 # Load water data from GRU -- addresses hooked up to residential city water lines
55 water = read.csv("GRU_water.csv", header = TRUE)
56 water$ADDRESS = trimws(water$ADDRESS) # delete extra whitespaces in address field
57 water = tidyr::separate(water, ADDRESS, into = c("ADDRESS", "extraADD"), sep = "\\,",) #
   remove extra address text that won't join to other datasets
58 water$POSTAL = as.character(strtrim(water$POSTAL, width = 5)) # restrict POSTAL field
   to first 5 characters
59 water = subset(water, WATER == "CITY", select = c("ADDRESS", "POSTAL", "WATER"))
60 dim(water)
61
62 ## Clean community parcels data from Hal Knowles -- commented out because cleaned
   dataset loaded below
63 # Load, subset, write, reload property value data from Hal Knowles
64 # value = read.csv("CommunityParcels.csv", header = TRUE)
65 # value$ADDRESS = trimws(value$ADDRESS)
66 # value$POSTAL = as.character(strtrim(value$POSTAL, width = 5))
67 # value = subset(value, POSTAL == "32612" | POSTAL == "32607" | POSTAL == "32641" |
   POSTAL == "32653" | POSTAL == "32606" | POSTAL == "32608" | POSTAL == "32605" | POSTAL
   == "32601" | POSTAL == "32669" | POSTAL == "32603" | POSTAL == "32609")
68 # write.csv(value, "value.csv", row.names = FALSE)
69
70 # Load cleaned dataset on property values from Hal Knowles
71 value = read.csv("value.csv", header = TRUE)
72 value$ADDRESS = trimws(value$ADDRESS) # remove extra white space from address field
73 value = tidyr::separate(value, ADDRESS, into = c("ADDRESS", "extraADD"), sep = "\\,",) #
   remove extra address details that are formatted differently in each dataset and won't
   join well
74 value = subset(value, select = c("ADDRESS", "CNTASSDVALUE", "POSTAL"))
75 dim(value)
76
77 # combine GRU power data and GRU water data frames by address
78 GRU = plyr::join(power, water, by = "ADDRESS")
79
80 # combine GRU data to Hal Knowles' property value data by address
81 combined = plyr::join(GRU, value, by = "ADDRESS")
82 combined = subset(combined, CNTASSDVALUE != "NA" & CORRECT.DAYS > 1 & POSTAL != "32614"
   & POSTAL != "32615" & POSTAL != "32612" & POSTAL != "32603") # exclude strictly campus
   zipcodes and error zipcodes
83 #combined$POSTAL = as.factor(combined$POSTAL)
84
85 # Limit dataset to properties valued at above $20,000 and below $2 million to restrict
   list to likely residences
86 combined = subset(combined, CNTASSDVALUE > 20000 & CNTASSDVALUE < 2000000, select =
   c("ADDRESS", "CORRECT.DAYS", "POSTAL", "WATER", "CNTASSDVALUE"))
87 combined$WATER <- as.character(combined$WATER)
88 combined$WATER <- ifelse(is.na(combined$WATER), 'WELL', combined$WATER) # assumption
   (deemed ok by Jenn McElroy at GRU) that those addresses not hooked up to city water are
   likely on well water
89 combined = combined[!duplicated(combined),] # remove duplicated addresses
90

```

```

91 # test for property value patterns with power outage duration
92 #combined <- within(combined, POSTAL <- relevel(POSTAL, ref = "32641"))
93 hist(log10(combined$CORRECT.DAYS)) # Looks normalish
94 hist(log10(combined$CNTASSDVALUE)) # Looks very normal
95 powerdiff = glm(log10(combined$CORRECT.DAYS) ~ log10(combined$CNTASSDVALUE))
96 summary(powerdiff) # significant relationship ***
97 powerdiff # m = -0.2162, b = 1.5775
98
99 # test relationship between property value and water category (city/well)
100 unique(combined$WATER) # check that only two levels here
101 water_lm = glm(log10(combined$CNTASSDVALUE) ~ combined$WATER)
102 summary(water_lm) # significant relationship ***
103 water_lm # m -0.02925, b = 5.06512
104
105 # Load special library and function for plotting on a log scale
106 library("scales")
107 reverselog_trans <- function(base = exp(1)) {
108   trans <- function(x) -log(x, base)
109   inv <- function(x) base^(-x)
110   trans_new(paste0("reverselog-", format(base)), trans, inv,
111             log_breaks(base = base),
112             domain = c(1e-100, Inf))
113 }
114
115 # plot power outage duration against property value on log scale
116 quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on
  line below instead of this one
117 #tiff(filename = "Irma_power_poverty.tiff", units = "in", compression = "lzw", res =
  300, width = 12, height = 6)
118 ggplot(aes(x = CNTASSDVALUE, y = CORRECT.DAYS), data = combined) +
119   scale_x_log10(breaks = c(2000000, 200000, 20000), labels = function(x) paste0("$",
  scales::comma(x))) +
120   #scale_y_continuous(trans = "reverse") +
121   geom_point(color = "grey") +
122   geom_quantile(quantiles = c(0.25, 0.75)) +
123   xlab("County-assessed Property Value (USD)") + ylab("Irma Power Outage Duration
  (days)") +
124   theme(axis.title = element_text(family = "Trebuchet MS", color="#666666",
  face="bold", size=15)) +
125   theme(axis.text = element_text(family = "Trebuchet MS", color="#666666", face="bold",
  size=12)) +
126   geom_smooth(method = "lm", se=FALSE, color="darkgreen")
127 # dev.off() # run this line after figure code to finish saving out figure to file
128
129 # Testing power outage duration and property value differences in postal zones
130 combined$POSTAL = as.factor(combined$POSTAL) # make sure POSTAL field not numeric
131 combined <- within(combined, POSTAL <- relevel(POSTAL, ref = "32606")) # ref category
  of zip code with lowest percent residents below poverty level (also one of highest
  average incomes)
132 overall = glm(log10(combined$CORRECT.DAYS) ~ combined$POSTAL)
133 summary(overall) # significant differences in duration power outage between 32606 and
  ALL other zip codes
134 overall
135
136 # test whether there is significant difference in property value between zip codes
137 postalproperty = glm(combined$CNTASSDVALUE ~ combined$POSTAL)
138 summary(postalproperty) # yes, significant property value diffs between zip codes

```

```

139 postalproperty
140
141 # plot some boxplots to look at differences between POSTAL codes
142 quartz(width = 10, height = 6)
143 boxplot(log10(combined$CNTASSDVALUE) ~ combined$POSTAL)
144
145 quartz(width = 10, height = 6)
146 ggplot(combined, aes(x=POSTAL, y=CNTASSDVALUE)) +
147   geom_violin() +
148   scale_y_log10() +
149   geom_boxplot(width = 0.1)
150
151 quartz(width = 10, height = 6)
152 boxplot(combined$CORRECT.DAYS ~ combined$POSTAL)
153
154 quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on
  line below instead of this one
155 #tiff(filename = "Irma_power_poverty_POSTAL.tiff", units = "in", compression = "lzw",
  res = 300, width = 12, height = 6)
156 ggplot(combined, aes(x=POSTAL, y=CORRECT.DAYS)) +
157   geom_violin() +
158   geom_boxplot(width = 0.1) +
159   xlab("GRU service area zip codes, ordered left to right by increasing average
  income") +
160   ylab("Irma Power Outage Duration (days)")
161 # dev.off() # run this line after figure code to finish saving out figure to file
162
163 # Load demographic and power outage data by zip code
164 postal = read.csv("Postal_map.csv", header = TRUE)
165 names(postal)
166
167 # plot zip code power outage duration against average property value in zip code
168 quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on
  line below instead of this one
169 #tiff(filename = "Irma_power_poverty_demographics.tiff", units = "in", compression =
  "lzw", res = 300, width = 12, height = 6)
170 ggplot(postal, aes(x= AVGVALUE, y = DAYSPOWEROUTLONGERTHAN32606), label = POSTAL) +
171   scale_x_log10(breaks = c(100000, 125000, 150000, 200000), labels = function(x)
  paste0("$", scales::comma(x))) +
172   geom_point(color = "grey") +
173   geom_text(aes(label=POSTAL), vjust= c(-1, -1, 1.5, 2, -1, -1, -1, -1), hjust= 0.5) +
174   geom_quantile(quantiles = c(0.25, 0.75)) +
175   xlab("Average Property Value (USD)") + ylab("Irma Power Outage Duration longer than
  zip 32606 (days)") +
176   theme(axis.title = element_text(family = "Trebuchet MS", color="#666666",
  face="bold", size=11)) +
177   theme(axis.text = element_text(family = "Trebuchet MS", color="#666666", face="bold",
  size=10)) +
178   geom_smooth(method = "lm", se=FALSE, color="black")
179 # dev.off() # run this line after figure code to finish saving out figure to file
180
181 ## extra figure code for experimental postal density plots
182 #quartz(width = 10, height = 6)
183 # ggplot(aes(x= CNTASSDVALUE, y = CORRECT.DAYS, colour = POSTAL), data = combined) +
184 #   scale_x_log10() +
185 #   facet_wrap(~POSTAL) +
186 #   #geom_jitter(aes(colour = POSTAL, shape = WATER)) +

```

```
187 # geom_density2d() +  
188 # scale_shape_manual(values = c(1, 17)) +  
189 # xlab("County-assessed Property Value (USD)") + ylab("Irma Power Outage Duration  
    (days)") +  
190 # geom_smooth(method = 'lm', se=FALSE, color="black")
```