```r
######################################################################
####### Sort, join & analyze GRU data on Power Outages after Irma #######
######################### Joan Meiners 2017 #############################

# load libraries
library(dplyr)
library(plyr)
library(tidyr)
library(ggplot2)
library(lubridate)
library(MASS)

# load linear model function for ggplot plotting
lm_eqn = function(m) {

  l <- list(a = format(coef(m)[1], digits = 2),
            b = format(abs(coef(m)[2]), digits = 2),
            r2 = format(summary(m)$r.squared, digits = 3));

  if (coef(m)[2] >= 0)  {
    eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r)^2~"="~r2,l)
  } else {
    eq <- substitute(italic(y) == a - b %.% italic(x)*","~~italic(r)^2~"="~r2,l)
  }

  as.character(as.expression(eq));
}

################################
setwd("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/Energy Burden Project Files")

# load power data from GRU -- addresses that lost power and duration of outage
power = read.csv("GRU_power.csv", header = TRUE)
power$ADDRESS = trimws(power$ADDRESS) # remove extra whitespaces in address field
power = tidyr::separate(power, ADDRESS, into = c("ADDRESS", "extraADD"), sep = "\\,") # separate out extraneous address fields that won't join with other datasets
power$POSTAL = as.character(strtrim(power$POSTAL, width = 5)) # limit POSTAL field to 5 characters
unique(power$POSTAL) # find out which POSTAL codes are included in data
power$POWER.DURATION = as.numeric(as.duration(hm(power$POWER.DURATION))) # convert power duration to a numeric field rather than hr:min format given

# calculate correct power outage time difference (GRU calculation did not add in mutliple days of power outage)
power = tidyr::separate(power, POWER.OUT.TIME, into = c("DAY.OUT", "HOUR.OUT"), sep = "\\ ") # restructure GRU data to calculate days out of power
power = tidyr::separate(power, POWER.RESTORE.TIME, into = c("DAY.RESTORE", "HOUR.RESTORE"), sep = "\\ ")
power$DAY.OUT = as.Date(power$DAY.OUT, "%m/%d/%y") # change date format
power$DAY.RESTORE = as.Date(power$DAY.RESTORE, "%m/%d/%y")

power$DURATION.DAYS = as.numeric(difftime(power$DAY.RESTORE, power$DAY.OUT), units="days") # calculate number of days out of power
power$POWER.DURATION = power$POWER.DURATION / (60 * 60 * 24) # convert minutes to days
```

```r
power$CORRECT.DAYS = power$POWER.DURATION + power$DURATION.DAYS # create new column with calculated days out of power added to GRUs calculated hour:min out of power
power$CORRECT.DAYS = as.numeric(power$CORRECT.DAYS) # make sure field is numeric
power = subset(power, select = c("ADDRESS", "CORRECT.DAYS", "POSTAL"))
power = power[!duplicated(power$ADDRESS),] # eliminate duplicated addresses
dim(power)

# load water data from GRU -- addresses hooked up to residential city water lines
water = read.csv("GRU_water.csv", header = TRUE)
water$ADDRESS = trimws(water$ADDRESS) # delete extra whitespaces in address field
water = tidyr::separate(water, ADDRESS, into = c("ADDRESS", "extraADD"), sep = "\\,") # remove extra address text that won't join to other datasets
water$POSTAL = as.character(strtrim(water$POSTAL, width = 5)) # restrict POSTAL field to first 5 characters
water = subset(water, WATER == "CITY", select = c("ADDRESS", "POSTAL", "WATER"))
dim(water)

## Clean community parcels data from Hal Knowles -- commented out because cleaned dataset loaded below
# load, subset, write, reload property value data from Hal Knowles
# value = read.csv("CommunityParcels.csv", header = TRUE)
# value$ADDRESS = trimws(value$ADDRESS)
# value$POSTAL = as.character(strtrim(value$POSTAL, width = 5))
# 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")
# write.csv(value, "value.csv", row.names = FALSE)

# load cleaned dataset on property values from Hal Knowles
value = read.csv("value.csv", header = TRUE)
value$ADDRESS = trimws(value$ADDRESS) # remove extra white space from address field
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
value = subset(value, select = c("ADDRESS", "CNTASSDVALUE", "POSTAL"))
dim(value)

# combine GRU power data and GRU water data frames by address
GRU = plyr::join(power, water, by = "ADDRESS")

# combine GRU data to Hal Knowles' property value data by address
combined = plyr::join(GRU, value, by = "ADDRESS")
combined = subset(combined, CNTASSDVALUE != "NA" & CORRECT.DAYS > 1 & POSTAL != "32614" & POSTAL != "32615" & POSTAL != "32612" & POSTAL != "32603") # exclude strictly campus zipcodes and error zipcodes
#combined$POSTAL = as.factor(combined$POSTAL)

# limit dataset to properties valued at above $20,000 and below $2 million to restrict list to likely residences
combined = subset(combined, CNTASSDVALUE > 20000 & CNTASSDVALUE < 2000000, select = c("ADDRESS", "CORRECT.DAYS", "POSTAL", "WATER", "CNTASSDVALUE"))
combined$WATER <- as.character(combined$WATER)
combined$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
combined = combined[!duplicated(combined),] # remove duplicated addresses

```

```r
# test for property value patterns with power outage duration
#combined <- within(combined, POSTAL <- relevel(POSTAL, ref = "32641"))
hist(log10(combined$CORRECT.DAYS)) # looks normalish
hist(log10(combined$CNTASSDVALUE)) # looks very normal
powerdiff = glm(log10(combined$CORRECT.DAYS) ~ log10(combined$CNTASSDVALUE))
summary(powerdiff) # significant relationship ***
powerdiff # m = -0.2162, b = 1.5775

# test relationship between property value and water category (city/well)
unique(combined$WATER) # check that only two levels here
water_lm = glm(log10(combined$CNTASSDVALUE) ~ combined$WATER)
summary(water_lm) # significant relationship ***
water_lm # m -0.02925, b = 5.06512

# load special library and function for plotting on a log scale
library("scales")
reverselog_trans <- function(base = exp(1)) {
  trans <- function(x) -log(x, base)
  inv <- function(x) base^(-x)
  trans_new(paste0("reverselog-", format(base)), trans, inv,
            log_breaks(base = base),
            domain = c(1e-100, Inf))
}

# plot power outage duration against property value on log scale
quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on line below instead of this one
#tiff(filename = "Irma_power_poverty.tiff", units = "in", compression = "lzw", res = 300, width = 12, height = 6)
ggplot(aes(x = CNTASSDVALUE, y= CORRECT.DAYS), data = combined) +
  scale_x_log10(breaks = c(2000000 ,200000, 20000), labels = function(x) paste0("$", scales::comma(x))) +
  #scale_y_continuous(trans = "reverse") +
  geom_point(color = "grey") +
  geom_quantile(quantiles = c(0.25, 0.75)) +
  xlab("County-assessed Property Value (USD)") + ylab("Irma Power Outage Duration (days)") +
  theme(axis.title = element_text(family = "Trebuchet MS", color="#666666", face="bold", size=15)) +
  theme(axis.text = element_text(family = "Trebuchet MS", color="#666666", face="bold", size=12)) +
  geom_smooth(method = "lm", se=FALSE, color="darkgreen")
# dev.off() # run this line after figure code to finish saving out figure to file

# Testing power outage duration and property value differences in postal zones
combined$POSTAL = as.factor(combined$POSTAL) # make sure POSTAL field not numeric
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)
overall = glm(log10(combined$CORRECT.DAYS) ~ combined$POSTAL)
summary(overall) # significant differences in duration power outage between 32606 and ALL other zip codes
overall

# test whether there is significant difference in property value between zip codes
postalproperty = glm(combined$CNTASSDVALUE ~ combined$POSTAL)
summary(postalproperty) # yes, significant property value diffs between zip codes
```

```r
postalproperty

# plot some boxplots to look at differences between POSTAL codes
quartz(width = 10, height = 6)
boxplot(log10(combined$CNTASSDVALUE) ~ combined$POSTAL)

quartz(width = 10, height = 6)
ggplot(combined, aes(x=POSTAL, y=CNTASSDVALUE)) +
  geom_violin() +
  scale_y_log10() +
  geom_boxplot(width = 0.1)

quartz(width = 10, height = 6)
boxplot(combined$CORRECT.DAYS ~ combined$POSTAL)

quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on line below instead of this one
#tiff(filename = "Irma_power_poverty_POSTAL.tiff", units = "in", compression = "lzw", res = 300, width = 12, height = 6)
ggplot(combined, aes(x=POSTAL, y=CORRECT.DAYS)) +
  geom_violin() +
  geom_boxplot(width = 0.1) +
  xlab("GRU service area zip codes, ordered left to right by increasing average income") +
  ylab("Irma Power Outage Duration (days)")
# dev.off() # run this line after figure code to finish saving out figure to file

# load demographic and power outage data by zip code
postal = read.csv("Postal_map.csv", header = TRUE)
names(postal)

# plot zip code power outage duration against average property value in zip code
quartz(width = 12, height = 6) # this is view window, to save figure to file, turn on line below instead of this one
#tiff(filename = "Irma_power_poverty_demographics.tiff", units = "in", compression = "lzw", res = 300, width = 12, height = 6)
ggplot(postal, aes(x= AVGVALUE, y = DAYSPOWEROUTLONGERTHAN32606), label = POSTAL) +
  scale_x_log10(breaks = c(100000 ,125000, 150000, 200000), labels = function(x) paste0("$", scales::comma(x))) +
  geom_point(color = "grey") +
  geom_text(aes(label=POSTAL), vjust= c(-1, -1, 1.5, 2, -1, -1, -1, -1), hjust= 0.5) +
  geom_quantile(quantiles = c(0.25, 0.75)) +
  xlab("Average Property Value (USD)") + ylab("Irma Power Outage Duration longer than zip 32606 (days)") +
  theme(axis.title = element_text(family = "Trebuchet MS", color="#666666", face="bold", size=11)) +
  theme(axis.text = element_text(family = "Trebuchet MS", color="#666666", face="bold", size=10)) +
  geom_smooth(method = "lm", se=FALSE, color="black")
# dev.off() # run this line after figure code to finish saving out figure to file

## extra figure code for experimental postal density plots
#quartz(width = 10, height = 6)
# ggplot(aes(x= CNTASSDVALUE, y = CORRECT.DAYS, colour = POSTAL), data = combined) +
#   scale_x_log10() +
#   facet_wrap(~POSTAL) +
#   #geom_jitter(aes(colour = POSTAL, shape = WATER)) +
#   geom_density2d() +
#   scale_shape_manual(values = c(1, 17)) +
#   xlab("County-assessed Property Value (USD)") + ylab("Irma Power Outage Duration (days)") +
#   geom_smooth(method ='lm', se=FALSE, color="black")
```