```r
#######################################################################
############# EJUF Code to Evaluate Solar Data for Liana ##############
########################### Joan Meiners 2017 ##########################

setwd("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/Liana_Solar_EJUF/")

library(dplyr)
library(ggplot2)

# load data
solar = read.csv("Liana_Solar.csv")
solar$kWDC = as.numeric(solar$kWDC) # make sure kilowatt-hours reading in as numeric
solar$ADDRESS = trimws(solar$ADDRESS) # delete extra whitespace around addresses
dim(solar)
View(solar)

# join to value dataset and pull in zip code information from community parcels dataset
value = read.csv("/Users/joanmeiners/Dropbox/Fall 2017/Environmental Journalism/value.csv", header = TRUE)
solarval = plyr::join(solar, value, by = "ADDRESS")
solarval = subset(solarval, POSTAL != "NA" & CNTASSDVALUE < 2000000, select = c("ADDRESS", "POSTAL", "kWDC", "CNTASSDVALUE")) # filter out all addresses without an associated zipcode and all properties valued at over $2 million since they are likely not personal residences
solarval$POSTAL = as.factor(solarval$POSTAL)
solarval$ADDRESS = as.character(solarval$ADDRESS)

# transform variables to reasonable degree of normalcy
hist(solarval$kWDC) # needs to be transformed
hist(log10(solarval$kWDC)) # roughly normally distributed, not perfect (right skewed)
hist(log10(solarval$CNTASSDVALUE)) # pretty normal, slightly left skewed

# look for trends in solar usage by property value
solar1 = glm(log10(kWDC) ~ log10(CNTASSDVALUE), data = solarval)
summary(solar1)
solar1

# graph the results
quartz(width = 12, height = 6)
ggplot(aes(y = kWDC, x = CNTASSDVALUE), data = solarval) +
  scale_x_log10(breaks = c(2000000 ,200000, 20000), limits = c(19000, 3000000), labels = function(x) paste0("$", scales::comma(x))) +
  scale_y_log10(breaks = c(0, 25, 50, 100, 200, 400)) +
  geom_point(color = "grey") +
  xlab("County-assessed Property Value (USD)") + ylab("Solar energy usage per address (kWDC)") +
  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")

# save figure to file
```

```
tiff(filename = "Solar_value.tiff", units = "in", compression = "lzw", res = 300,
width = 12, height = 6)
# run figure code from above section here (w/o quartz line), then next step
dev.off()
```