```
################################################################################
# source functions for corruption_eda_w203.Rmd
# authors: Andre Fernandes, Keenan Szulik, and Erik Hou
#
#   this script is separated into two sections:
#       1. packages
#       2. functions
################################################################################

################################################################################
# Packages
################################################################################

if (!require("pacman")) install.packages("pacman")
pacman::p_load(readr, ggplot2, dplyr, reshape, knitr, kableExtra, RColorBrewer,corrplot,
grid, gridExtra)

################################################################################
# Functions
################################################################################

#-------------------------------------------------------------------------------
# set_directories:
#
#     description:
#         stores working directory information for each user and avoids merge conflicts
#     parameters:
#         @author: string
#     returns
#         @work_dir: directory folder location
#-------------------------------------------------------------------------------

set_directories <- function(author=author){

  if(author=="Andre"){
    work_dir <- "C:/Users/afern/Desktop/Git/Berkeley/w203/corruption_eda_w203"
  }else if(author=="Keenan"){
    work_dir <- "/Users/keenanszulik/Documents/corruption_eda_w203"
  }else if(author=="Erik"){
    work_dir <- "c:/other/mids/w203/homework/lab_1/corruption_eda_w203"
  }else{
    stop(paste0(author,": Please add yourself to set_directories in functions.R in
~/utils/"))
  }

  return(work_dir)
}

#-------------------------------------------------------------------------------
# load_rda:
#
#     description:
#         loads rda and assigns it to user-defined variable
#     parameters:
#         @loc: string
#     returns
#         @df: dataframe
#-------------------------------------------------------------------------------

load_rda <- function(loc="data/Corrupt.Rdata"){

  df <- as.data.frame(eval(as.name(load(loc))))
```

```
  return(df)
}

#-------------------------------------------------------------------------------
# multiplot:
#
#     description:
#         returns multiple plots as one
#
#-------------------------------------------------------------------------------

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}

#-------------------------------------------------------------------------------
# plot_correlation:
#
#     description:
#         returns correlation plot for continuous variables
#
#-------------------------------------------------------------------------------

plot_correlation <- function(df_clean, min_unique){
  df_clean_dropped <- df_clean %>% select(-distUNplz)
  df_clean_cor <- cor(df_clean_dropped[, unlist(lapply(df_clean_dropped, function(x)
is.numeric(x) & length(unique(x))>min_unique))], use='pairwise.complete.obs')
  df_clean_cor[is.na(df_clean_cor)] <- 0
  plot_p <- corrplot(df_clean_cor, method="color",
                     type="upper", addCoef.col = "black",tl.col="black", tl.srt=45,
sig.level = 0.01, insig = "blank",
                     diag=FALSE, number.cex=.65, tl.cex=.7)
}
```

```
#-------------------------------------------------------------------------------
# lm_eqn:
#
#     description:
#         returns equation for the linear model
#
#-------------------------------------------------------------------------------

lm_eqn <- function(df, x, y, log=FALSE){
  if (log==TRUE){
    m <- lm(eval(as.name(y)) ~ eval(as.name(x)), df);
    eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r)^2~"="~r2,
                     list(a = format(coef(m)[1], digits = 2),
                          b = format(coef(m)[2], digits = 2),
                          r2 = format(summary(m)$r.squared, digits = 3)))
    as.character(as.expression(eq));
  } else{
    m <- lm(eval(as.name(y)) ~ eval(as.name(x)), df);
    eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r)^2~"="~r2,
                     list(a = format(coef(m)[1], digits = 2),
                          b = format(coef(m)[2], digits = 2),
                          r2 = format(summary(m)$r.squared, digits = 3)))
    as.character(as.expression(eq));
  }
}

#-------------------------------------------------------------------------------
# plot_vars:
#
#     description:
#         returns four view multiple with boxplot/scatter and histogram/density
#
#-------------------------------------------------------------------------------

plot_vars <- function(df=df, target, variable, var_type, facet_var){
  if (var_type == 'cat'){
    df_data <- as.data.frame(df)
    df_data$target <- log(df_data[, target]+1)
    df_data$variable <- as.factor(as.character(df_data[, variable]))
    df_data$facet_var <- as.factor(as.character(df_data[, facet_var]))
    df_data$facet_var <- factor(df_data$facet_var, levels=c("pre","pos"),
labels=c("Before 2002","After 2002"))


    p <- ggplot(df_data, aes(x=variable, y=target, color=variable)) +
      geom_boxplot() + guides(fill=FALSE)+ facet_grid(. ~ facet_var)+
      labs(title = paste(target, "by", variable),x=variable, y=paste0("log(",target,
"+1)"))  + theme(legend.position="none")
    p_hist <- ggplot(df_data, aes(x=variable)) + geom_bar(fill='darkgoldenrod1',
alpha=.5) +
      facet_grid(. ~ facet_var)+labs(title = paste("Histogram for", variable),
x=variable)
    multiplot(p, p_hist, cols=1)

  }

  if (var_type == 'numeric'){
    df_data <- as.data.frame(df)
    df_data$target <- log(df_data[, target]+1)
    df_data$facet_var <- as.factor(as.character(df_data[, facet_var]))
    df_data$facet_var <- factor(df_data$facet_var, levels=c("pre","pos"),
labels=c("Before 2002","After 2002"))
```

```
    p <- ggplot(df_data, aes(y=target, x=eval(as.name(variable)))) +
      geom_point()+ facet_grid(. ~ facet_var)+ geom_smooth(method=lm,  linetype="dashed",
                                                           color="darkred",
fill="blue")+labs(title = paste0("log(",target, "+1) ", "by ", variable), x=variable,
y=paste0("log(",target, "+1)"))
    den <- ggplot(df_data, aes(x=eval(as.name(variable)))) +
geom_density(fill='darkgoldenrod1', alpha=.5) +
      facet_grid(. ~ facet_var)+labs(title = paste("Density Plot for", variable),
x=variable)
    multiplot(p, den, cols=1)
  }

}
```