Liya Li

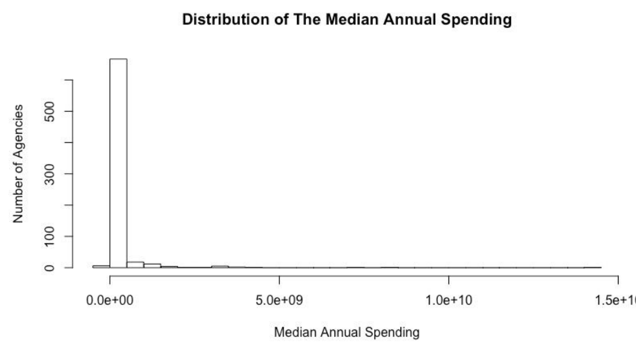
914978037

STA 141C HW1 Report

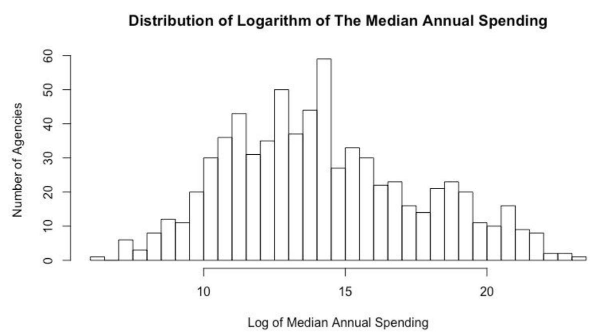
I. Computation

1. Agency with funding ID 1219 has the highest median annual spending.

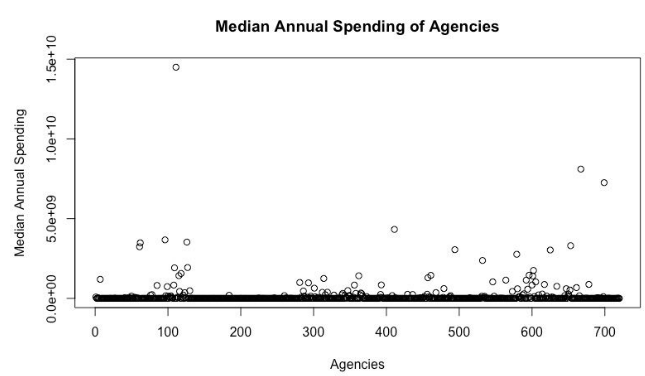
2. According to the histogram on the right, the distribution of median annual spending is very right-skewed. The obvious peak on the left hand side indicates that most agencies have similar low median annual spending while few agencies have high median annual spending which making the tail very long and almost invisible.



3. According to the histogram on the right, the distribution of logarithm of the median annual spending is approximately normal, even though it is a little bit right-skewed. The peaks occur on the left hand side of the center, and bars have approximately decreasing heights from the center to the tails.



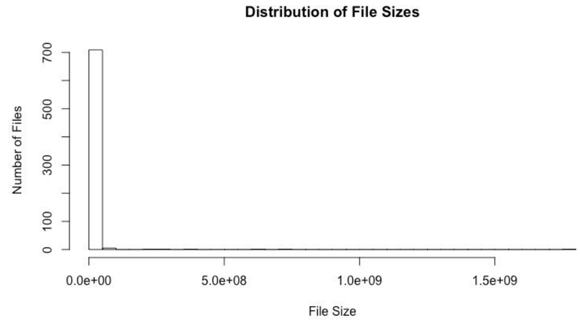
4. There is a clear separation between agencies that spend a large amount of money, and those which spend less money. According to the point plot below, each data point represents an agency and approximately most points are located at low median annual spending areas. We could consider 2.5e+09 median annual spending as the separation judge for high and low spending agencies. Agencies having median annual spending value above it are considered spending a large amount of money.



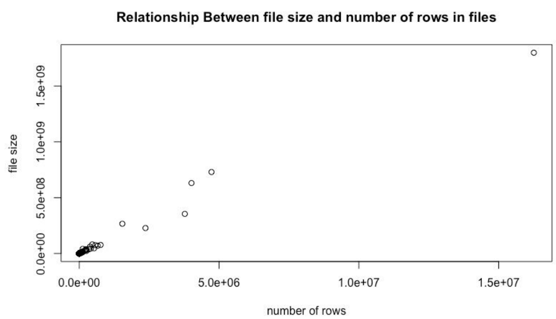
Liya Li
914978037

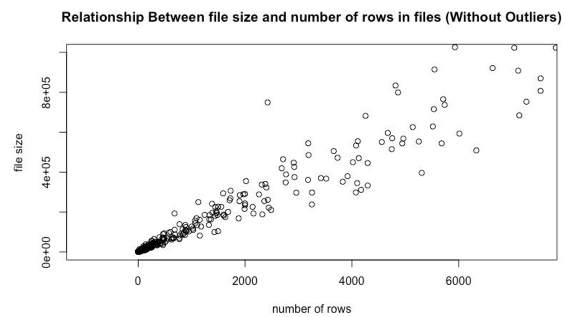
II. Reflecting

1. The histogram below shows file sizes distribution is right-skewed. File sizes are mostly very small, few file sizes are very large, and that's why the tail is very long and slim.



2. Without outliers that having extremely large row numbers, point plots below show that mostly file with more rows has a larger file size. The relationship between file sizes and row numbers is approximately linear.





3. Processing the whole R script with all the data takes 40.71274 mins.

4. I think this same approach I took also work for 10 times as many files because it won't affect the system running time and memory that much. However, if each file is 10 times larger, this same approach won't work on my computer which has low system memory.

5. I could make it faster by using fread() in data.table instead of using read.csv(), and removing testing codes in R script, such as those lines to check if 0.csv is removed, those lines to view data pieces, etc. The most expensive way to improve is to upgrade my computer's inner memory.

III. Appendix: R Script

```
#### STA141C HW1 - LIYA LI
##############
############## Timer Setting
############## before everything, set timer to calculate execution time of the whole script
start_time = Sys.time()
```

Liya Li
914978037

```
###############
############### Load in the dataset
###############
# unzip the original data zip file
zip_path = "~/Desktop/Winter Quarter 2019/STA 141C/hw1/awards.zip"
all_files = unzip(zip_path, list = TRUE)
head(all_files)

# extract only needed files
files = all_files[-1,]                # delete 0.csv which is in the first row
# same: files = all_files[2:dim(all_files[1])[1],]
head(files)

# double check if 0.csv is removed by checking dimensions
dim(all_files)
dim(files)                            # 0.csv is removed

###############
############### This function inputs fname and returns the annual median dataframe for this agency
############### file size of the file and number of rows in the file
get_med = function(fname){
  unzip(zip_path, files = fname)
  file_size = file.size(fname)
  sub_csv = read.csv(fname)
  row_n = dim(sub_csv)[1]

  # extract useful cols and modify them
  data_csv = sub_csv[c("total_obligation", "period_of_performance_start_date")]
  colnames(data_csv) = c("spending", "date") # change column names
  data_csv$date = as.Date(data_csv$date)     # convert to Date datatype

  # extract the year from date and drop the date column
  data_csv$year = format(data_csv$date, "%Y")
  data_csv$date = NULL

  # compute the sum annual spending for this agency
  sum_df = tapply(data_csv$spending, data_csv$year, sum, na.rm = TRUE)
  sum_df = as.data.frame(sum_df)

  # find the median annual spending for this agency
  med_result = as.data.frame(median(sum_df$sum_df))

  # add file names to the result dataframes
  # note: strsplit(targetstr, whatasseparator)[[1]][1]
  med_result$file = as.numeric(strsplit(fname, "[.]")[[1]][1])

  # change col names of med_result
  colnames(med_result) = c("med_annual_spending", "file_code")

  # delete the file as I go
  unlink(fname)

  # add file_size and row_n to med_result
  med_result$file_size = file_size
  med_result$row_n = row_n

  # return the results
  med_result
}
```

```
###############
############### Get the dataframe we need
###############
# get the file names
names = as.array(files$Name)

# apply get_med function to get median annual spending for each agency
result_list = lapply(names, get_med)

# convert the list of dataframe into df by column names
library(dplyr)
df = bind_rows(result_list, .id = NULL)

###############
############### Q1:Computation
###############
# 1. Which agencies have the highest median annual spending?
max_spending = max(df$med_annual_spending)
max_index = which(df$med_annual_spending == max_spending)
target_agency = df[max_index,]$file_code
target_agency
# agency with id 1219 has the highest median annual spending

# 2. Qualitatively describe the distribution of median annual spending.
hist(df$med_annual_spending, breaks = 40, main = "Distribution of The Median Annual Spending", ylab = "Number of Agencies" , xlab = "Median Annual Spending")

# 3.Qualitatively describe the distribution of the logarithm of the median annual spending.
hist(log(df$med_annual_spending), breaks = 40, main = "Distribution of Logarithm of The Median Annual Spending", ylab = "Number of Agencies" , xlab = "Log of Median Annual Spending")

# 4.Is there a clear separation between agencies that spend a large amount of money, and those which spend less money?
plot(df$med_annual_spending, main = "Median Annual Spending of Agencies", xlab = "Agencies", ylab = "Median Annual Spending")


###############
############### Q2:Reflecting
###############
# 1. Qualitatively describe the distribution of the file sizes.
hist(df$file_size, breaks = 30, main = "Distribution of File Sizes", ylab = "Number of Files", xlab = "File Size")

# 2. How does the size of the file relate to the number of rows in that file?
plot(df$file_size~df$row_n, main = "Relationship Between file size and number of rows in files", xlab = "number of rows", ylab = "file size")
plot(df$file_size~df$row_n, main = "Relationship Between file size and number of rows in files (Without Outliers)", xlab = "number of rows", ylab = "file size", xlim = c(-1000,7500), ylim = c(0,1000000))

# 3. How long does it take to process all the data?
# continue time calculation
end_time = Sys.time()

running_time = end_time - start_time
running_time

# 4. 5. See Report.
```