HW1 Code

Jared Yu

January 11, 2019

```r
# 1 - Computation
# Compute the median annual spending for each agency.

# Gather file names from awards.zip
file_names <- as.character(unzip('awards.zip', list = TRUE)$Name)
file_names <- file_names[-1] # remove 0.csv

# Specify which columns to read
which_columns = c("NULL", rep("vector", 2), rep("NULL", 2), "vector", "character")

# Create list of all files
all_files <- lapply(file_names, function(x) read.csv(unz('awards.zip', x),
                                                   header = TRUE, colClasses = which_columns))

# Create large data frame of the files and format the date
all_files_comb <- do.call("rbind", all_files)
all_files_comb[,2] <- format(as.Date(all_files_comb[,2]), '%Y')
all_files_comb[,1] <- as.numeric(all_files_comb[,1])

# Counting NA's
na_per_col <- sapply(all_files_comb, function(x) sum(is.na(x)))

spending_na <- (is.na(all_files_comb[1])) # Count cumulative NA's for spending/date
date_na <- (is.na(all_files_comb[2]))
table(spending_na, date_na) # 488241

# NA the dates which are too far out
table(all_files_comb[,2])
sum(all_files_comb$period_of_performance_start_date > 2018, na.rm = TRUE) # 9141
all_files_comb$total_obligation[all_files_comb$period_of_performance_start_date > 2018] = NA

# Group files by agency ID (unique)
agencies <- split(all_files_comb, all_files_comb$funding_agency_id)

# Calculate the median annual spending, and query the name/ID
group_func <- function(agency_list) {
  annual_sum <- tapply(agency_list$total_obligation,
                       agency_list$period_of_performance_start_date,
                       function(x) sum(x, na.rm = TRUE))
  median_ann <- median(annual_sum, na.rm = TRUE)
  agency_name <- agency_list[1,4]
  agency_id <- agency_list[1,3]

  query <- data.frame(median_annual_spending = median_ann,
                      agency_name,
                      agency_id)
```

```r
  return(query)
}

# 'Query' the median annual spending data using group_func()
annual_median_data <- lapply(agencies, group_func)
annual_median_data <- do.call('rbind', annual_median_data)

# 1. Which agencies have the highest median annual spending?
head(median_annual_data[order(median_annual_data$median_annual_spending, # Check top 5 median
                              decreasing = TRUE),])

# 2. Qualitatively describe the distribution of median annual spending.
par(mfrow = c(2,1))
hist(median_annual_data$median_annual_spending, main = 'Histogram of Median Annual Spending',
     xlab = 'Median Annual Spending') # Histogram of median annual spending
boxplot(median_annual_data$median_annual_spending, main = 'Boxplot of Median Annual Spending',
        ylab = 'Median Annual Spending')
dev.off()

# 3. Qualitatively describe the distribution of the logarithm of the median annual spending.
# Plot the histogram.
hist(log(median_annual_data$median_annual_spending), # Histogram of log median annual spending
     main = 'Histogram of the Logarithm of Median Annual Spending',
     xlab = 'Logarithm 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?

# 2 - Reflecting
# 1. Qualitatively describe the distribution of the file sizes.
# Find out the file size in terms of Length for each file
zip_file_path <- "C:/Users/qizhe/Desktop/STA 141C/hw1/awards.zip"
zip_info <- unzip(zip_file_path, list = TRUE)

# Plot the boxplot of size, along with the log histogram
par(mfrow = c(2,1))
boxplot(zip_info$Length, main = 'Boxplot of File Sizes', ylab = 'File Sizes')
hist(log(zip_info$Length), main = 'Histogram of Logarithm of File Sizes',
     xlab = 'Logarithm of Numeric Size of Files')
dev.off()

# 2. How does the size of the file relate to the number of rows in that file?
# Determine the number of rows
zip_filenames = zip_info$Name
numb_rows = sapply(zip_filenames[-1], function(x)
  nrow(read.csv(unz("awards.zip", x), colClasses = c("NULL", "vector", rep("NULL", 5)))))

par(mfrow = c(1,2)) # Scatterplot of files vs rows
plot(numb_rows, zip_info$Length[-1], main = 'Size of Files vs Number of Rows',
     xlab = 'Number of Rows', ylab = 'Size of Files')

# Create a new dataframe to remove the outliers
size_rows <- cbind(numb_rows, zip_info$Length[-1])
```

```r
colnames(size_rows) <- c('Rows', 'Size')
size_rows <- data.frame(size_rows)
size_rows2 <- size_rows[order(size_rows$Rows,
                              decreasing = TRUE),]
size_rows2 <- size_rows2[size_rows2$Rows < 800000,]

plot(size_rows2$Rows, size_rows2$Size, # Scatterplot of subset with regression line
     main = 'Size of Files vs Number of Rows (removed outliers)',
     xlab = 'Number of Rows', ylab = 'Size of Files')
abline(lm(size_rows2$Size~size_rows2$Rows))

# 3. How long does it take to process all the data?
# Time the various major sections of code
all_files_time <- system.time(lapply(file_names, function(x) read.csv(unz('awards.zip', x),
                                header = TRUE, colClasses = which_columns)))

data_frame_time <- system.time(do.call("rbind", all_files))

group_func_time <- system.time(lapply(agencies, group_func))

num_rows_time <- system.time(sapply(zip_filenames[-1], function(x)
  nrow(read.csv(unz("awards.zip", x), colClasses = c("NULL", "vector", rep("NULL", 5))))))

# 4. Do you think this same approach you took work for 10 times as many files? What if
# each file was 10 times larger?
sum(numb_rows) # Compare awards.zip with max dataframe size
2^31

# See if the max file length x10 would exceed the size of RAM
max_length <- zip_info[order(zip_info$Length, decreasing = TRUE),][2,2]
filex10 <- (max_length * 10) / (10^9)

# 5. How do you imagine you could make it faster?
```