Assignment, R (Big Data Programming, IT 715A) Demonstration 1a

Version 2016-11-04:

PART I: Learning basic R (for demonstration 1a)

Task: “Data aggregation in R”

In this assignment task, you work with car fuel efficiency data, for a simple analysis of vehicles. Note that this task should be done as an individual task and should also be presented individually during a demonstration (“Demonstration 1a”). You will show the instructor your code once it is ready, and more importantly, you should explain how you have solved the task. Book a demonstration slot for this (one slot per student), see course web page.

Preparing R for the assignment

In this assignment you use a few R packages that are very common for data analysis: *plyr*, *ggplot2* and *reshape2*.

Install the packages in R studio:

- install.packages("plyr")
- install.packages("ggplot2")
- install.packages("reshape2")

Load these packages in R studio, type:

- library(plyr)
- library(ggplot2)
- library(reshape2)

- Learn about these packages. This helps you develop the code to use them.
    - Refer to the *plyr* reference manual at http://cran.r-project.org/web/packages/plyr/plyr.pdf
    - Refer to the *ggplot2* reference manual at http://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
- Learning plyr
    - This link is the project page by the author http://plyr.had.co.nz/
    - Article as a good starting point http://www.jstatsoft.org/article/view/v040i01
    - "r-bloggers" http://www.r-bloggers.com/data-manipulation-with-dplyr/

Getting data into R and some basic manipulation

1. Download the data from http://www.fueleconomy.gov/feg/epadata/vehicles.csv.zip

2. Unzip the data file to manually inspect the data format in an editor. You'll need to understand the data structure.
3. Go to the description of the data, found at http://www.fueleconomy.gov/feg/ws/index.shtml#vehicle
4. Copy the data descriptions under the **vehicle** heading (not **emissions**). Save the file as "**varlabels.txt**". End the file with a newline.
5. Set the working directory in R studio: "**setwd**(your-path)". your-path is where you have your working files.
6. Now you can load the data directly from the zip file pointing out the data file within it: type **vehicles <- read.csv(unz("vehicles.csv.zip", "vehicles.csv"),stringsAsFactors = F)**.
    1. Make sure you understand what's going on here.
    2. Inspect some of the loaded data by typing **head(vehicles)**
7. Now we want to read the labels in **varlabels.txt** using the "-" character as a delimiter.
    1. Important note: there are such characters (-) in **varlabels.txt** (inside text of descriptors), replace them in an editor (see line 11 in **varlabels.txt**).
    2. An alternative to editing the file would be to use this command instead **labels <- do.call(rbind, strsplit(readLines("varlabels.txt")," - "))**.
8. Now you can read the labels: **labels <- read.table("varlabels.txt", sep = "-", header = FALSE)**.
    1. See some of the data by typing **head(labels)**
9. Check some more things:
    1. **nrow(vehicles)** will give the number of observations.
    2. **ncol(vehicles)** shows the number of data columns for each of them.
    3. **names(vehicles)** displays the data names (from varlabels.txt).
10. Now, get your first descriptive numbers for this data:
    1. Find out "How many unique years of data do we have?"
    2. Use **length(unique(vehicles[,"year"]))**
    3. Make sure you understand what's going on here

DEMONSTRATION 1 of "Demonstration 1a": "Explore and describe fuel efficiency"

- You will be asked to demonstrate this:
    - Find out a command for the first and last years of the data in a similar way, using the *min* and *max* functions.
    - Also, you will get another test question presented to you during the demonstration (but not before).

More aggregation

1. We want to count the number of data points (car models) with automatic and manual gear box
2. There's data with empty info about this, so we need to take care of that by assigning NA to it
3. **vehicles$trany** is the data column used for this, but it stores different texts for the gear box type
4. So, do this:
    1. First, set missing data to NA : **vehicles$trany[vehicles$trany == ""] <- NA**

    2. Then, create a new column to store only Auto or Manual based on the first four letters from vehicles$trany: **vehicles$trany2 <- ifelse(substr(vehicles$trany, 1, 4) == "Auto", "Auto", "Manual")**
    3. Convert this new variable to a factor: **vehicles$trany2 <- as.factor(vehicles$trany2)**
    4. Now: use **table()** to summarize: **table(vehicles$trany2)**

5. What happened here? Make sure you understand this code, so that you can explain it

Analyze fuel efficiency over time and visualize it

1. Now we will use ddply, check it out by typing **?ddply.** There seems to be many options.
2. We want to see the overall trend over the years on fuel efficiency (*Miles Per Gallon*/MPG). For this we aggregate rows by year, and for each group we compute the mean *highway*, *city* and *combine* fuel efficiency.
3. We store in a new data frame **mpgByYr**.
4. This is our first split-apply-combine: We split in groups by year, we apply the mean function to specific variables, then we combine it into a new data frame:

```
mpgByYr <- ddply(
  vehicles,
  ~year,
  summarise,
  avgMPG = mean(comb08),
  avgHghy = mean(highway08),
  avgCity = mean(city08)
)
```

5. Make sure you understand what's going on ☺
6. Now, let's plot this using **ggplot(),** plotting against the **year** variable. We also the graph nice, with axis labels, a title, a smoothed conditional mean(geom_smooth()) as a shaded region:

```
ggplot(mpgByYr, aes(year,  avgMPG))
  + geom_point()
  + geom_smooth()
  + xlab("Year")
  + ylab("Average MPG")
  + ggtitle("All cars")
```

- Think about what this graph means:
    - Does it really mean that there has been a dramatic change in fuel efficiency in cars sold during the last years?
    - How about the mix of cars sold? Can you find out if that has changed? How?

DEMONSTRATION PART 2 of “Demonstration 1a”:
"Gasoline cars, fuel efficiency over time visualization"

- You will be asked to demonstrate the following:
    - Create a new data frame based on **vehicles** with gasoline powered cars, using the **subset()** function. Find out how to use **subset()**.
        - Gas cars include “Regular Gasoline", "Premium Gasoline", "Midgrade Gasoline” in **fuelType1**
        - To filter that out, you can use the following code:
          fuelType1
          %in%
          c("Regular Gasoline", "Premium Gasoline", "Midgrade Gasoline")
        - We need a filter condition: **fuelType2** == ""
        - We also need a filter condition: **atvType** != "Hybrid"
        - Combine **fuelType1**, **fuelType2** and **atvType** in such a way when using **subset()**
    - Save the gas car subset in the new object: **gasCars**
    - Use **ddply()** to aggregate in the same way as above, over years, but now for **gasCars** instead of all cars, use **avgMPG = mean(comb08)**
    - Plot the graph for gas cars in the same way as above.