Homework 2

Lan Huong Nguyen

October 25, 2017

Contents

Please complete this assignment by 11/08 at 11:59pm. Submit your answers including your code and write-up (R or Rmd and pdf or html) on canvas.

Exercise 1: `ggplot`[20pt]

The following url: "http://cdiac.ess-dive.lbl.gov/ftp/ndp030/CSV-FILES/nation.1751_2014.csv" contains data on fossil fuel emissions.

a. Read data to R. Note that rows 1-3 contain information on the dataset itself. Delete these rows as they do not contain relevant information.

b. Compute the total yearly CO_2 emissions (column "Total.CO2.emissions.from.fossil.fuels.and.cement.production..thousand. summed over all countries (the world total CO_2 emission). You can use a forloop over "years" or `dplyr` functions.

c. Plot the world (summed over all countries) CO_2 emission over time in billion tonnes (Gt) per year, i.e. divide the quantity computed in (b) by 10^6. You can use a line or a scatter plot.

d. Now read the dataset located at "https://raw.githubusercontent.com/lukes/ISO-3166-Countries-with-Regional-Codes/master/all/all.csv" which contains an assignment of countries to regions. Merge emissions dataset to the countries dataset. Note that in emissions dataset countries are given with all caps, but not in countries dataset. You need to change that before merging the two tables. Hint: use the function `toupper()`. Add a column 'co2_emission' equal to CO_2 emission in Gt, i.e. 'Total.CO2.emissions.from.fossil.fuels.and.cement.production..thousand.metric.tons.of.C.'/10^6

e. Use `dplyr` to compute total annual CO_2 ('co2_emission') emission per 'sub.region'.

f. Use `ggplot` to generate a stacked density plot the annual CO_2 (in giga tonnes) by world regions ("sub.region"). Your plot should be similarto the one in Fig. 1 (but with other regional categories, and slightly different values). Hint: use `geom_area()` function with suitable parameters. Which region seems to produce most CO_2? You might like to modify the color scheme to better distinguish regions.

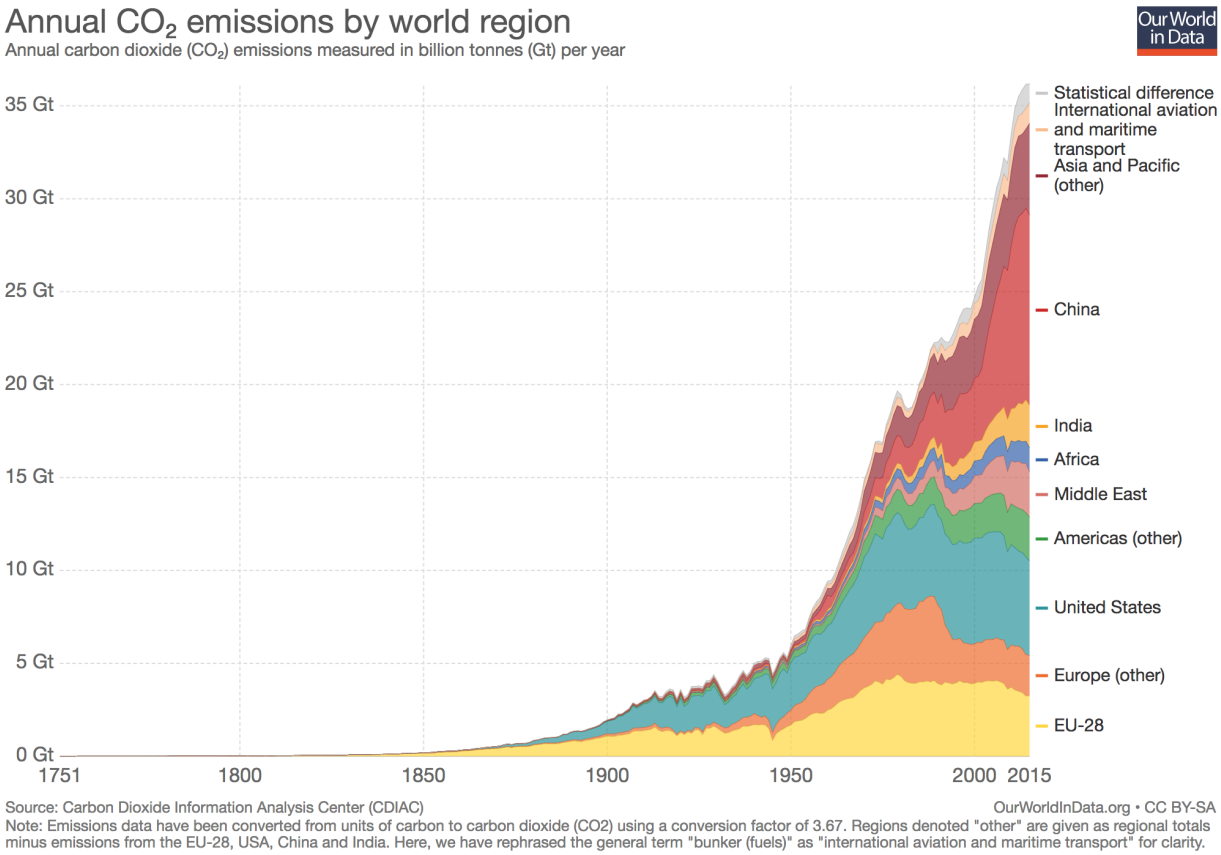

Figure 1: Source: https://ourworldindata.org/grapher/annual-co-emissions-by-region

Exercise 2: Gene expression data [20pt]

In this exercise we will use the DNA microarray gene expression data. You can read more about it on page 5 of "The Elements of Statistical Learning".

```
microarray <- read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.data")
info <- read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.info.txt",
                   skip = 12)
colnames(microarray) <- info$V1
```

In the 'microarray' matrix columns correspond to samples, and rows to genes.

a. Subset microarray to 500 most variable genes, i.e. the ones with the highest standard deviation across samples (you shoud use `order()` to find the indices). Then, plot a heatmap without clustering/dendrograms. You can use 'asp = 0.2' argument to change the aspect ratio of the heatmap.

b. Plot the previous heatmap with red/green color scheme. For your convenience here is the color vector you might like to use:

```
redgreen <- c("#FF0000", "#DB0000", "#B60000", "#920000", "#6D0000",
              "#490000", "#240000", "#000000", "#002400", "#004900",
              "#006D00", "#009200", "#00B600", "#00DB00", "#00FF00")
```

Then, plot the same graph but with dendrogram for rows (rows clustering).

d. Now instead of base heatmap, use interactive `heatmaply()` to generate the previous plot. You might want to add a command similar to the following `%>% layout(margin = list(l = 150, b = 350), autosize = F, width = 600, height = 800)` to the plot to set margins and to resize it.

e. What interesting patterns do you observe? Are there some differences between conditions? Are some genes up down regulated for certain groups? No need for long answers just look at the heatmap state what you see.

Exercise 3: Hypothesis testing [20pt]

Recall the movies data-frame we used in for lecture 3 exercises. It contains information on movies from the last three decates, which was scrapped from the IMDB database.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
url <- "https://raw.githubusercontent.com/Juanets/movie-stats/master/movies.csv"
movies <- tbl_df(read.csv(url))
```

a. Generate a boxplot of runtimes for action movies and commedies with jittered points overlaid on top. You might consider setting collor, fill and alpha arguments to modify clarity and transparency of the plot.

b. Test a hypothesis that the action movies have higher mean runtime (length) than the comedies. Is the difference statistically greater than zero at significance level $\alpha = 0.05$?

Exercise 4: linear model [20pt]

a. Read the data from "http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv" containing information on sales of a product and the amount spent on advertising using different media channels.

b. Generate a scatterplot of sales against the amount of TV advertising and add a linear fit line.

c. Now make a 3D scatterplot with axes corresponding to 'sales', 'TV' and 'radio'.

d. The dataset has 200 rows. Divide it into a train set with 150 observations and a test set with 50 observations, i.e. use `sample(1:200, n = 150)` to randomly choose row indices of the advertising dataset to include in the train set. The remaining indices should be used for the test set. Remember to choose and set the seed for randomization!

e. Fit a linear model to the training set, where the sales values are predicted by the amount of TV advertising. Print the summary of the fitted model. Then, predict the sales values for the test set and evaluate the test model accuracy in terms of root mean squared error (MSE), which measures the average level of error between the prediction and the true response.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

f. Fit a multiple linerar regression model including all the variables 'TV', 'radio', 'newspaper' to model the 'sales' in the training set. Then, compute the predicted sales for the test set with the new model and evalued the RMSE.
Did the error decrease from the one correspodning to the previous model?

g. Look at the summary output for the multiple regression model and note which of the coefficient in the model is significant. Are all of them significant? If not refit the model including only the features found significant. Which of the models should you choose?

Exercise 5: classification [20pt]

We load the following datsets including characteristics of emails and spams:

- 48 continuous real [0,100] attributes of type

'word_freq_WORD = percentage of words in the e-mail that match WORD.

- 6 continuous real [0,100] attributes of type

`char_freq_CHAR` = percentage of characters in the e-mail that match CHAR,

- 1 continuous real [1,. . .] attribute of type `capital_run_length_average` = average length of uninterrupted sequences of capital letters

- 1 continuous integer [1,. . .] attribute of type

`capital_run_length_longest` = length of longest uninterrupted sequence of capital letters

- 1 continuous integer [1,. . .] attribute of type

`capital_run_length_total` = sum of length of uninterrupted sequences of capital letters = = total number of capital letters in the e-mail

- 1 nominal {0,1} class attribute of type

spam = denotes whether the e-mail was considered spam (1) or not (0),

```
url.info <- "https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names"
spam.info <- read.table(url.info, comment.char = "|", skip = 32, stringsAsFactors = FALSE)
attributes <- gsub(":", "", spam.info[[1]])
symbols <- c("semicolon", "left.parenthesis", "left.sq.bracket", "exclamation",
             "dollar", "hashtag")
attributes[49: 54] <- paste0("char_freq_", symbols)
```

```
url.data <- "https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data"
spam <- read.csv(url.data, header = FALSE, stringsAsFactors = FALSE)
colnames(spam) <- c(attributes, "spam")
spam <- spam %>%
  mutate(spam  = factor(spam, levels = c(0, 1), labels = c("email", "spam")))
```

a. Check if the dataset contains balance classes spam vs email. To do this count the cases of spam and email, and make a barplot for the frequencies.

b. Divide the data into train and test set with a 60%-40% split. Remember to record the seed you used for randomization.

c. Use logistic regression involving all the predictors to train a model for classifying emails. Which features seem significant? Evaluate and report your model's accuracy on the test set.

d. Use random forest to train a model on the same train set. Report which variables have high importance scores. Then, evaluate the RF model's accurasy on the test set.

Exercise 6: ggmap [10pt]

a. Consider the two following locations: `from <-  c(lon = -122.169719, lat = 37.4274745)` and `to <- c(lon = -122.16242, loc = 37.44457)`. Create a vector for the bounding box of the two locations `bbox <- c(left = longitude.from, bottom = latitude.from,  right = longitude.to, top = latitude.to)` with appropriate values filled in. Then, use the `get_map()` and `ggmap` to generate a map containing the two locations. Use source: "google", maptype = "satellite" and a city-level-zoom, zoom = 15.

b. Use the function `route()` from the `ggmap` to generate a data-frame corresponding to the route from one location to the other. Then, use `geom_map()` to add the path corresponding to the route generated with the `route()` function. You can use a function `revgeocode()` to look up the addresses of the `from` and `to` locations.

c. In this exercise we will generate a map of San Francisco and include the information on the housing prices. The dataset with housing prices can be downloaded from github as follows:

```
url.housing <- "https://raw.githubusercontent.com/simonkassel/Visualizing_SF_home_prices_R/master/Data/S
sf.housing <- read.csv(url.housing, row.names = 1)
head(sf.housing)
```

```
##   X.1 X        UID ParcelID      long      lat SaleYr SaleDate SalePrice
## 1   1 1  1069035.9  1069035 -122.4498 37.78437   2009    90218 1395001.6
## 2   2 2 2448001V.9 2448001V -122.5057 37.73821   2009    90121  609998.5
## 3   3 3  2919050.9  2919050 -122.4657 37.74282   2009    90122  959996.7
## 4   4 4  2987006.9  2987006 -122.4640 37.74022   2009    90121  867999.2
## 5   5 5  5297024.9  5297024 -122.3926 37.73806   2009    90130  395003.0
## 6   6 6  6244022.9  6244022 -122.4133 37.71439   2009    90326  420003.0
```

```
##                   Neighborhood                     District PropArea Rooms
## 1 Jordan Park / Laurel Heights       District 1 - Northwest     2386     7
## 2               Outer Parkside    District 2 - Central West      878     4
## 3                  West Portal District 4 - Twin Peaks West     2783     8
## 4                  West Portal District 4 - Twin Peaks West     1650     8
## 5                      Bayview      District 10 - Southeast     1062     5
## 6            Visitacion Valley      District 10 - Southeast      875     6
##                           dist_year
## 1       District 1 - Northwest_2009
## 2    District 2 - Central West_2009
## 3 District 4 - Twin Peaks West_2009
## 4 District 4 - Twin Peaks West_2009
## 5      District 10 - Southeast_2009
## 6      District 10 - Southeast_2009
```

Use `get_map()` and `ggmap()` to plot a map of San Francisco using source = "google", maptype = "toner-lite" and zoom = 13. Then, use the data on housing prices above to add a layer of points colored by the SalePrice (use a good color scheme).