Graded exercise:

The Pigeon Racing Datasets

Teacher: prof. dr. Dries F. Benoit
Course: Open-source programming tools: R

Practical arrangements:

- This is a group work, not an individual assignment.
- Deadline (strict): Sunday December 31, 9am
(email to: Dries.Benoit@UGent.be).
- Deliver your project as three files.
  - Describe how you've tackled the questions in a pdf document (max 1 page per question).
  - Put all your R code in a separate R script with single flexible path definition.
  - A separate R script that contains your ShinyApp.
- Make sure the names of every group member are written on the first page.
- Make sure the final document clearly explains how you've solved the questions.
- Grading is based on the following aspects:
  - working code
  - clarity/readability of code (comment your code!)
  - consistency of code
  - efficiency of code
  - originality of the solution

Context:

Pigeon racing originated about 200 years ago in Belgium. The goal of the sport is to breed and train the fastest flying pigeon that, due to their unique homing instinct, is able to fly back to her pigeon loft in the shortest possible time. Since a couple of decades, there has been an steady decline in the

number of pigeon fanciers, however about 20,000 of them are still competing week after week on several races in Belgium. The sport is however gaining popularity in the (middle)east and other countries. In 2017, a Belgian champion pigeon was sold for €400,000 to a Chinese fancier. Another bird was sold for €360,000 to a South-African pigeon racer.
The datasets for this exercise contain the results of the national races in Belgium. The race distances vary from about 500km (Bourges, France) to more than 1,000km (Barcelone, Spain). Pigeon fanciers bring their birds to their club and the pigeons are then registered (chipped) and transported to location of liberation. Because the distances to the loft of every fancier is different, the winner is not the first arriving pigeon, but the bird with the highest meters-per-minute. Only the first 25% of the pigeons are awarded a place on the results list. The information about each national race is saved in two RData sets.

The info_* dataset has the following variables:

- "race" : name of the race
- "lat" : latitude of liberation point
- "long" : longitude of liberation point
- "date" : date of liberation
- "time" : liberation time
- "result": name of organizer
- "nbr_lofts" : number of participating lofts/fanciers
- "nbr_birds" : number of participating birds
- "age" : age category of birds

The results_* dataset has the following variables:

- "position" : the position of the pigeon in the end result
- "loft" : name of the loft/fancier
- "town" : town of loft/fancier
- "loft_id" : id of the loft/fancier
- "basket" : number of birds in this race by loft/fancier
- "basket_nr" : fanciers have to order their birds from best (1) to worst (basket)

- "distance" : distance in meters to the loft
- "ring" : ring number, unique identifier of the pigeon
- "arr_time" : time of arrival
- "diff" : not relevant for exercise
- "velocity" : flying speed in meter-per-minute
- "coeficient" : not relevant for exercise
- "next_pos" : next position of loft/fancier in result table

Questions:

1. Create a summary data.frame with one row per loft.
   The overall goal of this assignment is to create a single data.frame with one row per loft where the columns contain relevant information about the loft.

2. Link postal codes to towns.
   For this exercise you have to link the postal code of every town. You can use the *zipcodes_BE3.csv* file for creating this link. Some difficulties you might encounter:

   - Some towns are sub-towns (without own postal code) of larger towns
   - Some towns have the same name (in different provinces)
   - Some towns are misspelled

   The file *zipcodes_BE3.csv* contains the following variables:

   - "Postalcode" : postal code
   - "Town" : name of the (main) town
   - "Sub_town" : name of the sub-town
   - "Province" : province of the town
   - "N_inhab" : number of inhabitants
   - "translation" : translate town from Dutch to French (or from French to Dutch)

   Note: due to the above difficulties, it is quasi impossible to correctly link a postal code to every loft. Try to make the link as perfect as possible. The overall goal of this assignment is to create a single data.frame with one row per loft where the columns contain relevant information about the loft.

3. For every loft, calculate the latitude, longitude.
   For every loft that appears in two (or more) races, we can calculate the coordinates (latitude, longitude) of the loft. This is because:

   - we know the coordinates of the location of liberation
   - we know the distance of the location of liberation to every loft

   So, the intersection of the sphere of the earth, with the spheres defined by location and distance of the liberation points give us 1 or 2 possible coordinates for the loft. You can use the R-function *calc_coord.r* for this task. The function takes as inputs the coordinates (in decimal degrees!) and distances of the liberation locations. The output is:

   - NULL if there are no intersecting points
   - a list of 2, if there is 1 intersection point (first element of the list is location in (x,y,z) notation, second in lat/long)
   - a list of 4, if there are 2 intersecting point (first/second element of the list is location in (x,y,z) notation, third/fourth in lat/long)

   Note: Because the earth is not a perfect sphere (as is assumed in the *calc_coord.r* function), the output is imperfect and only precise up to some error (< 5km). Motivate how you've dealt with this.

4. Create new variables.
   Create minimum 15 relevant and new (loft_id, town, postal code, lat, long, etc. not included) variables that summarize the performance of the loft. Make sure you have some variation in the type of variable.

5. Create a Shiny app.
   Visualise relevant/interesting information in a good looking Shiny app. Take into account the good practices of data visualisation as discussed in class.