GROUP 10 - Florence SAVARYDEBEAUREGARD, Yongsheng LOH, Chenpei TAN

*Note: After Group10_R_Code is opened, click File > Reopen with Encoding > select UTF-8 before running it

Q1. Create a summary data.frame with one row per loft

a. Info datasets

- Latitude & Longitude are in degrees, minutes and seconds. Replaced accent with alphabets. We have used the function dg2dec to change the coordinates to decimal degrees.

b. Results datasets

- **Data cleansing**
  - **Replaced all accent with alphabets**
  - **Cleaned column "distance"**
    As the distance between liberation point to loft is only available for the first pigeon that arrived for the race and the rest of the pigeons are indicated by the position they have arrived. We have replaced distance with the max distance for each race per loft_id and created another column that states the arriving position of the pigeons for each race per loft_id.
  - **Removed observations with NA values in "position".**
    As only top 25% of the pigeons are awarded a place on the results dataset, some observations were not awarded a position and hence, we have decided to keep only those that are within top 25% which leaves us with 4300 unique loft_id from the initial 4312 loft_id. Reason being, the 12 loft_id (eg. 305628-78 and 240789-35) only have races with NA values in position, information such as the distance and basket are missing and since 11 out of these 12 loft_id have participated only in 1 race, it is also not possible to calculate the loft's coordinates. Since there is not much meaningful information we can gather from these loft_id with NA values in position, we have removed them.
  - **Cleaned column "town"**
    For towns that have additional information enclosed in parenthesis, we have break it down into town and province and replaced short form into long form. Values have been converted into lower case.

c. Merge liberation point of 24 races from info dataset to the results dataset

In order to merge info and results datasets of 24 races together that have no common key, we have created a unique key (UKey_Race) based on race indicated on their filenames for merging.

d. Created final data.frame for 4300 unique loft_id containing the following information

- loft_id, loft, town, province from results dataset
- Postalcode from zipcode dataset
- Loft_latitude and loft_longitude that are calculated using calccoord function with coordinates from liberation point and distance
- 20 new variables that summarize the performance of the loft

Q2. Link postal codes to town

Result: Out of 4300 unique loft_id, we have managed to match 4235 (98%) of the loft_id with a Postalcode, only 65 loft_id could not be matched with a Postalcode (NA).

Scenarios encountered:

a. **Some towns are in short form or with accent**
Replaced all accent with alphabets and the province or town that have short form into long form (eg. “Antw.” to “Antwerpen” & “St.-Truiden' to “Sint-Truiden”)

b. **Some towns have different ways of capitalizing the letters**
We have converted the values in the relevant columns containing town information into lower case.

c. **Some towns are sub-towns (without own postal code) of larger towns**
For towns that have additional information enclosed in parenthesis, we have break it down into town and province to merge with zipcode dataset sub_town and town catered in dataset “full_table_merge1”.

d. **Some towns have the same name (in different province)**
We have split the town in results dataset into town and province catered in dataset “full_table_merge2”.

e. **Some towns are sub_town or Pure_town (empty sub_town in zipcode)**
Catered in datasets “full_table_merge3” and “full_table_merge4”

f. **Zipcode has the same information for two languages (Eg. Ukkel – Dutch and Uccle - French)**
We have kept only one observation with Dutch version in “Town” and French in “Translation”. To cater to loft with province in French language, we have merge province with translation in dataset “full_table_merge5”

g. **Additional information enclosed in parenthesis in town is actually a town**
Split town in results dataset into town and province and match province to town. Catered in dataset “missing_postal_merge1”.

h. **Towns are joined with “-“ or “/” but only first word is sufficient to get a match**
We have split the town by these 2 symbols and matched to either Pure_town or sub_town in datasets “missing_postal_merge2” and “missing_postal_merge3” respectively.

Q3: Calculating coordinates (latitude and longitude) for each loft

Result: We obtained coordinates for 3241 (3139+102) loft_id which is 75% of all 4300 loft_id.

There is no way to calculate the coordinates of the loft in the below 2 scenarios.

1. For the 986 loft_id that have only participated in **1 race** (eg. Loft_id 106617-14 only participated in Tulle), there is only 1 circle, hence there is no intersection point.
2. For the 50 loft_id that have only participated in **1 race location** (eg. Loft_id 108998-67 participated in only Argenton.I and Argenton.III), the distance, latitude and longitude of the 2 liberation points would be the same hence there will be infinite intersection points.

For the rest of the 3264 loft_id that have participated in 2 or more different race locations, we can apply the x and y coordinates of liberation points along with the distance into the function to find intersection points. We have prepared the following 2 datasets to apply the function.

1. For loft_id that have participated **only in 2 different race locations**, the dataset will contain x and y information of the 2 different race locations. We have obtained results that were either a list of 4 or NULL. With the 2 intersecting points (third & fourth elements) in the list of 4, we will pick the coordinates of the intersecting point that fall within the Belgium's coordinates. If both are within Belgium, we will select the first solution since we are unable to determine which is the right one.
2. For loft_id that have participated in **more than 2 different race locations**, we have picked 3 race locations. With 3 different coordinates, we have created 3 different combinations of x and y (1st combination: Race 1 & 2, 2nd combination: Race 1 & 3 and 3rd combination: Race 2 & 3). We have applied the function to these 3 combinations of x and y to derive intersection points. The results were either a list of 4 or NULL. With the 2 intersecting points derived from the list of 4 for each combination, we have picked the first set of coordinates that fall within the Belgium's coordinates. These leave us with 1 solution for each of the combinations for each loft_id hence we need to approximate the actual coordinate of the loft by looking at the closeness of these solutions. We then calculate the absolute difference between each solution and its mean for each loft. The actual location of the loft is most likely to fall within the range of coordinate solutions that are close to each other (ie, small absolute difference from its mean). We then picked the coordinates with the least difference as the loft's coordinates. Illustration using loft_id (100042-35) can be found in the code.

As there is only 3139 loft_id with solution, we have filter out the 125 loft_id that have a NULL output. It is likely that the circles have no intersecting points due to rounding errors. Hence, we have explored adding an additional 5KM to the distance of the smaller circle (shorter distance of x or y) and apply the function again which only 30 loft_id falls within Belgium while many are in United Kingdom. With trial and error, we have decided to add 400M instead and have obtained a solution for 102 loft_id with a list of 4 that has 2 solutions and we have picked the coordinates that fall within Belgium. We are unable to derive an intersection point for the last 23 loft_id.

Q4: New variables that summarize performance of the loft

- **v1_avg_velocity** - Average velocity of all the pigeons that are in the top 25% in each loft.
- **v2_max_velocity** - Maximum velocity that has been recorded per loft among all races that it has participated.
- **v3_number_pigeon** - Number of pigeons per loft that have arrived in top 25%
- **v4_max_race** - Among all the pigeons each loft has, find the maximum number of races a pigeon has participated in.
- **v5_best_position** - Best position the pigeon has achieved for each loft among all races participated.
- **v6_avg_position** - Average position for each loft among all races participated
- **v7_avg_basket** - Average basket size for each loft among all races participated
- **v8_sum_basket** - Total number of pigeons participated in all races per loft even if they did not arrive in top 25%
- **v9_race_count** - Total number of races each loft has participated
- **v10_avg_distance** - Average distance of all races each loft has participated in
- **v11_accuracy_percent** – Accuracy divided by number of pigeons in top 25%.
- Accuracy: Assigned value of 1 if basket_nr (owner's prediction) = position_within_loft_race (actual arriving position) and 0 if differs. Sum the values for each loft_id.
- **v12_number_top_25** – Percentage of pigeons arrived in top 25% compared to number of pigeons participated per loft
- **v13_best_pigeon** - Best pigeon in each loft that has lowest average position
- **v14_avg_best_pigeon_position** - Average position of the best pigeon in each loft
- **v15_std_error_pred** - Measures standard error between owner's prediction (basket_nr) and their actual arriving position (position_within_loft_race).
- **v16_most_consistent_pigeon** - It has the least standard deviation in term of its velocity across all races participated in each loft.
- **v17_most_inconsistent_pigeon** - It has the highest standard deviation in term of its velocity across all races participated in each loft.

  *Note: For **v16_most_consistent_pigeon** & **v17_most_inconsistent_pigeon**, we did not include those pigeons that have only raced once as we cannot evaluate if it is consistent/inconsistent based on only one race. Limitation for these 2 metrics is that for the loft_id that has only 1 pigeon that raced more than once, it will appear as both most inconsistent and most consistent pigeon.
- **v18_sd_velocity** - Variability of all pigeon performance (velocity) for each loft across all races
- **v19_daysbetweenraces** - Average number of days between races for each loft

  *Note: There are 987 loft_id with NaN as they have only participated in one race hence we are unable to determine the days between races.
- **v20_avg_raceinterval_pigeon** - Average race interval (number of days) for each pigeon within each loft

Q5: Create a shiny app

a. Lofts' location

- The map shows the location of lofts in Belgium which helps user get a view of how lofts are distributed and validate the accuracy of the output from calccoord function. For instance, the map clearly shows that the lofts are mostly concentrated in northern Belgium. It also allows user to select all or specific lofts and maps' magnification slicer to view their location. At least one loft must be selected for the map to show, else there will be an aesthetics error.
- Limitation – When map is magnified, it doesn't allow user to move the map or choose a focal point. Sometimes, there will be an error loading the map. Since there is no submit button to update view, user can either maximize or minimize the window or adjust the map's magnification slider to refresh the view again. As the map is constantly re-downloaded from google each time a user changes the input, there is a lag time of about 1 minute when refreshing the map. Note that there must be at least one loft_id selected for map to appear.

b. Lofts' performance

- User can compare a loft's performance to overall average as well as other selected lofts measured by variables. Eg. loft_id 100042-35 having 94th in v5_best_position has performed better than overall average of 688th but worse than loft_id 100168-64 with 86th best position.

c. Graphs

- User is able to select variables to plot a histogram that shows distribution of the lofts. The scatter plot shows the relationship between average velocity and distance. It illustrates how distance affects velocity of pigeons. The point of maximum average velocity occurs at average distance of approximately 600km. This could indicate that on average, pigeons tend to perform worse if the race distance is more than 600km. User is able to move or zoom in/out of scatter plot to have a clearer view and obtain lofts information by hovering on the point.

d. Loft information by race

- User is able to select all or specific races to view the lofts' information. Search boxes can be used to search in general or specific columns. It allows user to select the number of entries to view as well as sort the columns by ascending or descending order.

e. Race Performance

- Bar plot compares the performance of a user-selected race metric across the 24 races. It is interesting to note that the Barcelona race has significantly lower average velocity than other race, however the average distance covered is not significantly higher than the rest.

f. Race information

- User is able to select a range of dates to get information about the race that have happened within this period such as the number of lofts and pigeons that have participated in each race.