# Homework 0: Exploratory Data Analysis on the El Salavordan Conflict

Olivier Binette, STA 490/690

***General instructions for homeworks***: Please follow the uploading file instructions according to the syllabus. Your code must be completely reproducible and must compile.

***Advice***: Start early on the homeworks and it is advised that you not wait until the day of as these homeworks are meant to be longer and treated as case studies.

***Commenting code*** Code should be commented. See the Google style guide for questions regarding commenting or how to write code https://google.github.io/styleguide/Rguide.xml. No late homework's will be accepted.

***R Markdown Test***

0. Open a new R Markdown file; set the output to HTML mode and "Knit". This should produce a web page with the knitting procedure executing your code blocks. You can edit this new file to produce your homework submission.

**Total points on assignment: 5 (reproducibility) + 10 points for the assignment.**

# Introduction

A Durham pantry provides food and basic necessities to students in need of assistance. Since the beginning of the COVID-19 pandemic, pantry usage has mainly been restricted to the weekly bag program: students fill bag orders through a Qualtrics survey, and the bags are then delivered to them or ready for pick-up on the next Saturday.

As a volunteer statistician at the Pantry, you are tasked to:

1. describe the composition of pantry users through bag order records; and
2. provide recommendations regarding how the pantry could improve its data collection efforts to better evaluate the need of students.

Your analysis will assist the Pantry in its negotiation with University stakeholders and will help them prepare for the upcoming Campus Food Insecurity Symposium.

# Data

The raw Pantry records consists of weekly bag orders filled between June 24 and November 4 2020 using a Qualtrics survey. This is an online survey made available to all of Duke's graduate and professional students. The survey is mainly advertised over email in periodical newsletters and it records all bag orders for the given time period. Its content is summarized in Table 1.

| | Question no. | Question | Answer form |
|---|---|---|---|
| Identifiers | 2 | First name and last initial | Free form |
| | 3 | Duke email | Free form |
| | 4 | Phone number | Free form |
| Order | 5 | Delivery or Pickup? | Multiple choices |
| | $6-7$ | Adress and delivery instructions | Free form |
| | 8 | Food allergies | Free form |
| | 9 | Number of members in household | 1-2 or 3+ |
| | 10 | Want baby bag? | Yes or no |
| | $11-29$ | Order items | Multiple choices |
| Survey | 30 | Degree | Multiple choices or Other |
| | 31 | School | Multiple choices or Other |
| | 32 | Year in graduate school | Multiple choices |
| | 33 | Number of adults in household | Multiple choices |
| | 34 | Number of children in household | Multiple choices |
| | 35 | Main challenges accessing food | Multiple choices or Other |
| | 36 | Feedback | Free form |

Table 1: Summary of the Qualtrics "Weekly Grocery Bag Request Survey" questions. Note that number of sub-questions which are not relevant to this analysis have been omitted.

In order to protect the privacy of pantry users, the raw records have been anonymized and only a subset of this data is made available to you. The personal identifiers `name`, `phone` and `email` have been encrypted using an MD5 hash function. Furthermore, only the non free-form survey answers (questions 30-34) have been provided to you for your analysis. This data is contained in the `order_data.rds` file and the anonymization pre-processing script can be found in `encrypt_responses.R`.

# Methodology

Many pantry users have filed more than one order in the considered time period. Your first task is therefore to resolve individual pantry users using this data. Next, you will consolidate survey answers for each individual. Finally, you will provide summary statistics and provide recommendations regarding the Pantry's data collection efforts.

## Task 1: record linkage

The fields `name`, `phone` and `email` are all expected to contain variations. For example, I could have entered my name as "Olivier", "Olivier B." or "Olivier Binette" in different days, or pantry volunteers may have recorded my name as "Oliver". I have two duke email addresses: olivier.binette@duke.edu as well as ob37@duke.edu, and on some days I could have used my gmail address. My phone number changed a few months after I moved to Durham.

In order to identify unique individuals, perform deterministic record linkage using the three fields `name`, `phone` and `email` in combination: define two records to be a match if they agree on at least one of these fields. Given this record linkage, assign a unique entity identifier to each pantry user.

You may use parts (a)-(g) below as a guide, or you may submit a different solution of your own.

**Part a)** Read-in the encrypted data `order_data.rds` and view it using the function `View()`. How many non-empty orders were placed in the considered time period?

**Solution.**

```
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
```

```
data = readRDS("order_data.rds")
View(data)
sum(!is.na(data$name))
```

```
## [1] 549
```

**Part b)** Construct a function `linkage_rule(A, B)` which returns `TRUE` if the record number `A` matches record number `B` on any of the fields `name`, `phone` or `email`.

**Solution.**

```
linkage_rule <- function(A, B) {
  recordA = data[A, c("name", "phone", "email"), drop=TRUE]
  recordB = data[B, c("name", "phone", "email"), drop=TRUE]
  return(any(recordA == recordB, na.rm=TRUE))
}
```

**Part c)** Enumerate all possible record pairs using `pairs = t(combn(1:nrow(data), 2))`. Apply the function `linkage_rule` to each pair in order to determine if they match and store the result in a vector named `matches`. This might take a few minutes to run.

**Solution.**

```
pairs = t(combn(1:nrow(data), 2))
matches = sapply(1:nrow(pairs), function(i) linkage_rule(pairs[i,1], pairs[i,2]))
```

**Part d)** Construct a matrix with two columns, where each row is a pair of two matching record numbers. Make sure that each record is matching itself and that you remove NA values. This is called an edge list.

**Solution.**

```
matching_pairs = pairs[matches, ]
edge_list = rbind(matching_pairs,
                  cbind(1:nrow(data), 1:nrow(data)))
```

**Part e)** Use the function `igraph::graph_from_edgelist()` to construct a linkage graph based on the edge list. Then use `igraph::components()` to find connected components and assign unique entity identifiers. Hint: use `igraph::components(g)$membership`.

**Solution.**

```
g = igraph::graph_from_edgelist(edge_list)
IDs = igraph::components(g)$membership
```

**Part f)** Add a column to the data with the unique entity identifiers. How many distinct individuals visited the pantry in the considered time period?

```
data = data %>% mutate(uniqueID = IDs)
length(unique(IDs))
```

```
## [1] 178
```

**Part g)** (Optional challenge) Can you perform the record linkage much more efficiently, without enumerating all record pairs? Hint: use sorting.

**Solution.**

```
links <- function(field) {
  o = order(data[[field]])
  sorted = (1:nrow(data))[o]
  edges = t(sapply(1:(length(sorted)-1), function(i) {
    if (linkage_rule(sorted[i], sorted[i+1])) {
      return(c(sorted[i], sorted[i+1]))
    }
    return(c(NA, NA))
  }))

  edges[complete.cases(edges), ]
}

edge_list = rbind(links("name"),
                  links("phone"),
                  links("email"))

g = igraph::graph_from_edgelist(edge_list)
IDs = igraph::components(g)$membership
```

## Task 2

Construct a data frame where each row represents a unique pantry user and each column is a representative answer to the survey question. This does not have to be complicated. For example, you can choose the representative answer to be the first non NA answer for this individual.

You may use parts (a)-(b) below as a guide or provide your own complete solution.

**Part a)** Write a function `first_non_na()` which returns the first non-NA entry of a vector or otherwise returns "NA".

```
first_non_na <- function(x) {
  if (all(is.na(x))) return("NA")
  x[!is.na(x)][[1]]
}
```

**Part b)** Construct a data frame where each row represent a unique pantry user and each column is the first non-NA data entry for this person. Hint: group `data` by unique ID and use the function `dplyr::summarize_all()` to summarize each column using the `first_non_na()` function.

```
representers = data %>%
  group_by(uniqueID) %>%
  summarize_all(first_non_na)
```

## Task 3

Given the above, answer the following questions:

1. How many individuals have only used the pantry only one time in this time period?

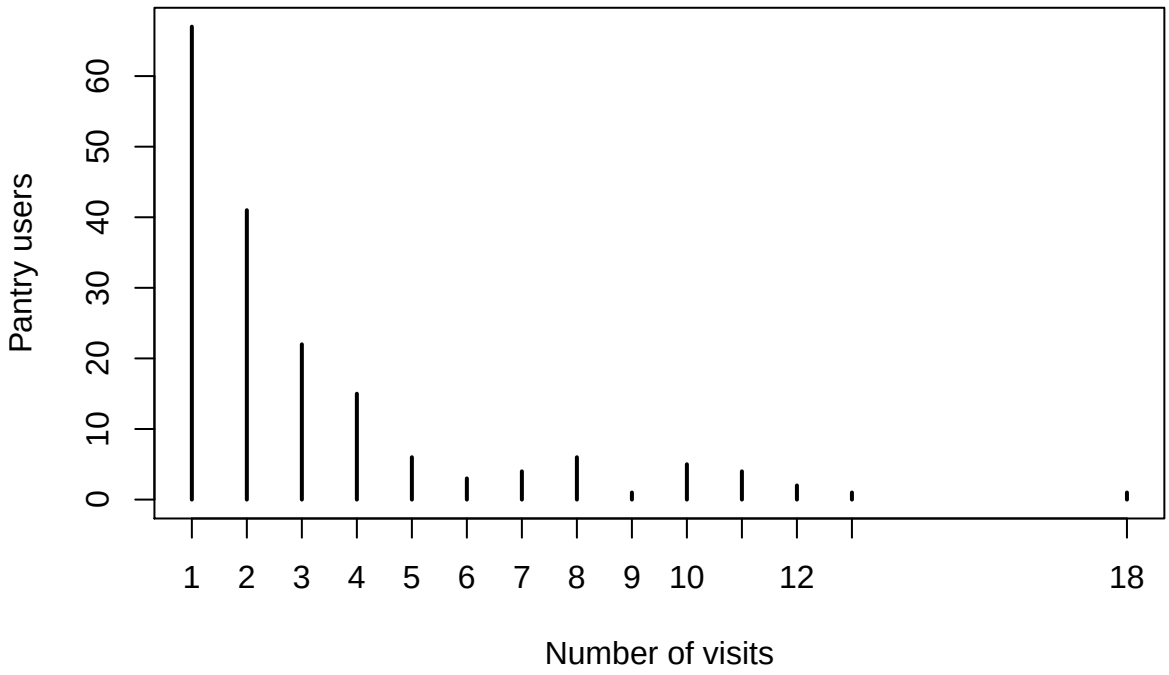
**Solution.**

```
table(table(data$uniqueID))[1]
```

```
##  1
## 67
```

2. Plot the distribution of the number of visits per pantry user.

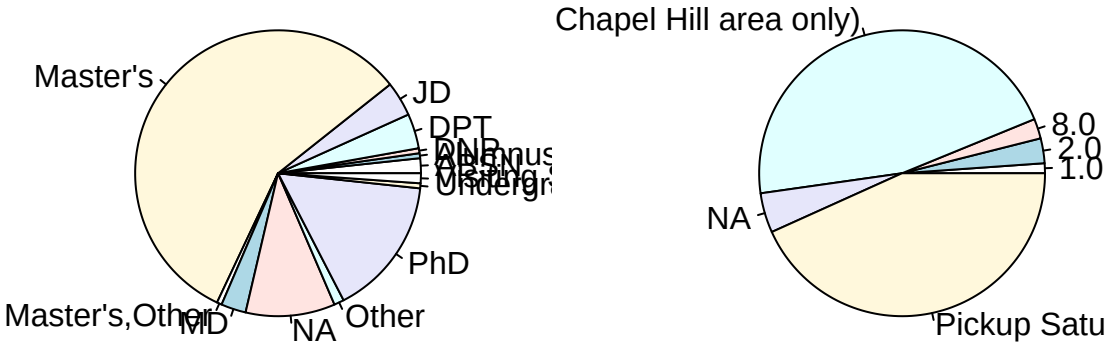
```
plot(table(table(data$uniqueID)),
     ylab="Pantry users",
     xlab="Number of visits")
```




3. What is the composition of degree type and order type among pantry users? Hint: you can use pie charts for simplicity.

**Solution.**

```
# Note: this is not cleaned up.
par(mfrow=c(1,2))
pie(table(representers$degree))
pie(table(representers$how))
```




Furthermore, discuss the following: How could the pantry improve its data collection efforts in order to gain more meaningful insights into its users and their needs? Keep in mind the need to protect the privacy of the pantry users.