VISUALIZATION PRACTICAL WORK
DATA VISUALIZATION

GROUP: 2
MEMBERS: PABLO CRUCERA BARRERO, JAVIER GALLEGO GUTIÉRREZ AND JÚLIA SÁNCHEZ MARTÍNEZ
DEGREE: MASTER'S PROGRAMME IN DATA SCIENCE

DATE: JANUARY 24, 2022

Contents

1 Introduction

The visual analytics process aims to extract knowledge from graphical representations of data. That is why data visualization tools are a trend in business and research, specifically related to medicine. They allow users to gain a better understanding of the structure and patterns of their data, which can help them in their process of developing strategies to tackle a particular problem.

Along this practical work, we developed an interactive tool as an attempt to solve the particular problems of an end user. The analysis and design of that tool was done following the four levels of visualization design shown in Figure 1 [1].

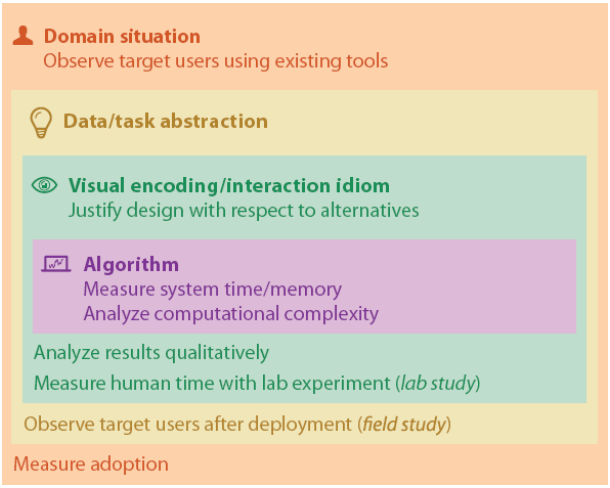

Figure 1. The four nested levels of visualization design.

2 Problem characterization

The first step of the project had to do with finding a data set. We decided to use the "TLC Trip Record Data", a data set from the New York City Taxi and Limousine Commission [2] that tracks the information about all the taxi trips done in the city of New York. This data set is quite big and includes data about different types of vehicles, so we decided to focus on the yellow taxi trips in 2019.

Figure 2. Classical yellow taxis of New York City.

Once we had the data, we proposed the particular "end user" or "analyst" whose problems we would try to solve with our tool. This "analyst" is a licensed taxi driver of New York City and the questions he or she would like to answer are the following:

1. How is the taxi trip flow between the different areas of the city according to the different days and times?
2. How could different taxi zones relate to the analyst's economy (tips, trips' length and trips that end up in dispute)?
3. Can I group taxi trips into sufficiently differentiated groups? What are those groups like and what useful information can be extracted from them?

Question 1 is important for users because they should know where the busiest areas are. Given that information, they could decide the best place for working depending on the week day and time. Otherwise, Question 2 represents useful information for analysts because their main goal is to earn as much money as they can. For that purpose, they may be interested in earning the largest possible tips and covering the longest distances while avoiding trouble as far as possible. Finally, Question 3 is the most open of all three. Taxi drivers could be interested in discovering hidden patterns in taxi trips, like different types of costumers or trips. This knowledge may help them make decisions related to their daily work.

3 Data and task abstractions

3.1 Data abstractions

The selected data set is composed of 7667792 instances, each one described by 18 attributes. Table 1 contains each of the attributes, its data type and also a brief description of what they are [2].

Field Name	Data Type	Description
VendorID	Integer	A code indicating the TPEP provider that provided the record: 1 = Creative Mobile Technologies LLC 2 = VeriFone Inc.
tpep_pickup_datetime	Factor	The date and time when the meter was engaged.
tpep_dropoff_datetime	Factor	The date and time when the meter was disengaged.
passenger_count	Integer	The number of passengers in the vehicle. This is a driver-entered value.
trip_distance	Double	The elapsed trip distance in miles reported by the taximeter.
RatecodeID	Integer	The final rate code in effect at the end of the trip: 1 = Standard rate 2 = JFK 3 = Newark 4 = Nassau or Westchester 5 = Negotiated fare 6 = Group ride
store_and_fwd_flag	Character factor	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward", because the vehicle did not have a connection to the server: Y = store and forward trip N = not a store and forward trip
PULocationID	Integer	TLC Taxi Zone in which the taximeter was engaged.
DOLocationID	Integer	TLC Taxi Zone in which the taximeter was disengaged.
payment_type	Integer factor	A numeric code signifying how the passenger paid for the trip: 1 = Credit card 2 = Cash 3 = No charge 4 = Dispute 5 = Unknown 6 = Voided trip
fare_amount	Double	The time-and-distance fare calculated by the meter.
extra	Double	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
mta_tax	Double	$0.50 MTA tax that is automatically triggered based on the metered rate in use.
tip_amount	Double	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
tolls_amount	Double	Total amount of all tolls paid in trip.
improvement_surcharge	Double	$0.30 improvement surcharge assessed on hailed trips at the flag drop. The improvement surcharge began being levied in 2015.
total_amount	Double	The total amount charged to passengers. Does not include cash tips.
congestion_surcharge	Integer	Surcharge in case of congestion.

Table 1. Data set variables.

This data was enriched with shape data containing the boundaries for the different taxi zones along with other information relating to them such as their identifier and the Borough they belong to. [2]

In the following sections, we will explain the separate types of transformations performed to the data of Table 1 made to implement the different types of idioms.

3.1.1 Trips flow

According to the data abstractions framework, in Question 1 we have a network: taxi zones are nodes and edges are represented by trips to other zones. It could be seen as a weighted directed graph where weights are the number of trips from origin node to destination node in a period of time. We also manage geometry data, as we have the shapes of the 263 taxi zones in which New York City is divided.

3.1.2 Economic stats

In this case, we have geometry data with the shapes of the different taxi zones and a table with quantitative (ordered sequential) data, with one attribute per statistic that the analyst would like to be computed.

3.1.3 Clustering

We used a table dataset structure with derived attributes from the original ones to perform clustering. The original attributes were a mix of categorical and quantitative and the final table had only quantitative features.

3.2 Task abstractions

3.2.1 Trips flow

Question 1 is actually an open question. Users will have to find new knowledge about taxi trips distribution along different week days and hours. That reason makes “discover” the high-level action. The target of this search is a trend, because users want to identify which pattern taxi trips follow depending on the time and week day. Peak taxi hours represent important information for our analyst. Otherwise, we also need to determine mid-level and low-level actions choices. As users do not know exactly what they are looking for (patterns are unanticipated) and the location to find the target is also unknown (peak taxi zone is not known), the type search is explore. Finally, for the query types we chose “compare” because the analyst is interested in the differences between trends along taxi zones and time. Summarizing, task abstraction for this question would be “discover, explore, and compare trends between taxi zones”.

3.2.2 Economic stats

What the analyst expects to obtain from this visualization is to acquire information that is too hard to extract and read from the, which may help them make decisions in the future. Therefore, what we are trying to do is to discover new knowledge (economic stats related to taxi zones) that is not presented in the raw data file, and from there, to compare data between taxi zones.

3.2.3 Clustering

The answer to the third analyst's questions is largely open. The goal of clustering the data of taxi trips was to visualize the different groups in which these trips could be separated and analyze their values. The task was to discover which attributes distinguished each cluster and explore the results for useful unknown information. For that, we compared different feature values and extracted some conclusions.

The output of this task was also intended to be the generation of new hypotheses, so that the analyst could extract more knowledge from further representations.

4 Interaction and visual encoding

For each of the three different problems to solve, we discuss the proposed solutions:

4.1 Trips flow

In the first moment, we were sure we wanted to use a flow map and a chord diagram to represent trips flow. One of the first problems we faced had to do with the large amount of taxi zones we had to deal with. It was impossible to represent them all in the chord diagram. For that reason, chord diagram changed taxi zones to boroughs. As only seven boroughs had to be represented (Brooklyn, Queens, Manhattan, The Bronx, Staten Island, EWR and another borough for Unknown locations), the chord diagram was now a good visualization option. This visualization present areas of a circle as marks for the amount of links and hue (color) for representing categorical variables (boroughs).

Regarding the map, the most important decision was how to represent trips flow. We chose animated lines with variable width depending on the number of trips between zones. These lines are the marks that represent links (connection mark) and areas are the ones for taxi zones. The width, direction and speed of the arc are the channels. A problem with this visual encoding was that it was not possible to represent the flow within the same taxi zone. However, we decided to keep going and not represent this particular case.

After all that, interactions had to be decided. There were multiple options to discriminate between days and hours, but we had to keep in mind the impact this decision would have in the algorithmic implementation in terms of time and memory performance. The final decision was to divide the data set in week days (from Monday to Sunday) and each day in 24 periods of one hour. This way, the analyst would get reliable information in order to distinguish between work days, weekends and different hours within them. A negative aspect of this decision had to do with the inclusion of some holidays in work days, but it would be a lot of work and there are not that many holidays.

A summary of the visualization and interactions included in the final application (some of the derived from validation part detailed in Section 6.1 is depicted in Table 2.

	Solution
Visualization	1. Flow map for taxi zones. 2. Chord diagram for boroughs and taxi trips. 3. Histogram for pickup taxi zones.
Interaction	1. Select a week day. 2. Select a period of one hour (from XX:00 to XX:59). 3. Zoom in and out. 4. Change "pitch". 5. Change "bearing". 6. Change flow width. 7. Change arcs length. 8. Change animation speed. 9. Change map background (with an associated change in zones shapes color). 10. Show only trips where a particular zone is involved.

Table 2. Trips flow interaction and visual encoding.

4.2 Economic stats

Our objective was to relate each of the taxi zones mentioned above to numerical values which are the statistics. The visual encoding chosen for tackling this problem was to create a choropleth map for each attribute that wanted to visualized, which easily allowed to relate a geographical region with their corresponding value, thus allowing to observe one quantitative attribute each time.

The interactions with the map that the end-user could do are organized following a certain hierarchy schema:

Variable to be visualized (tips/trip length/disputes) in form of a list selection:

- Tips. The user can choose among two different elements:
  - For which taxi zone the stat has been calculated (Origin/Destination) as radio buttons.
  - What statistic has been computed (Mean/Median) as radio buttons.
- Trip length. The user can choose among two different elements:
  - For which taxi zone the stat has been calculated (Origin/Destination) as radio buttons.
  - What statistic has been computed (Mean/Median) as radio buttons.
- Disputes. The user can only choose over one option:
  - For which taxi zone the stat has been calculated (Origin/Destination) as radio buttons.

The color palette for the visualization varies depending on the variable to visualize.

Yellow-green sequential palette has been chosen for visualizing tips because of the implicit association that humans do between money and green. More saturated colors (more pale-like) imply lower tips, whereas pure green colors are bound to the other end.

Similarly, yellow-orange-red sequential palette has been the one that we picked for visualizing dispute percentages. More intense red represents a higher level of alert, and therefore, is associated with danger (i.e. higher percentage of disputes).

Finally, a diverging blue-purple palette has been our choice for visualizing trip distances, with blue representing shorter distances and purple greater ones. The choice for this palette was to use different hues from the other two. In all cases, color gray has been the one selected for depicting NA values.

The color bins have been designed based on empirical observations over data, with shorter intervals for values with higher frequencies and larger containers for less frequent ones, so it would be easier to discriminate (pop-out) extreme values and there was a representative number of instances for all groups. The color bins appear in as a color legend in the upper left corner, where it initially does not overlap any taxi zone.

Each zone is bound to a label showing the taxi zone identifier, its borough, what statistic is being visualized, the value of the statistic for that zone, and the corresponding units.

	Solution
Visualization	Choropleth map depicting the requested statistic per taxi zones, both in origin and destination.
Interaction	1. Select the variable that wants to be visualized. 2. Select whether the statistic wants to be related to the Pick-up taxi zone or to the Drop-off zone. 3. Choose if the statistic to visualize is the mean or the median (only for tips and trip length). 4. Display a box with information about a taxi zone when placing the mouse over that zone. 5. Zoom in and out. 6. Drag map.

Table 3. Economic stats interaction and visual encoding.

4.3 Clustering

The first challenge we faced was finding which representations would help us better understand the insights that can be extracted from the clusters, as clustering alone does not answer any analytical questions about the data.

First, we represented the k-means algorithm results with the two first Principal Components. PCA or Principal Component Analysis aims to represent multidimensional data by means of two new variables which contain the maximum information possible. This first plot enabled us to draw some initial conclusions about how different the instances clustered in different groups were taking the inter-cluster distance into account. The colors representing each cluster (green, blue and red) are the ones that come by default and change in every computation. The importance of hue relies on the ability of visually differentiating the instances belonging to each cluster.

Next, in order to determine the attributes that most influence each cluster and the particular value they take we plotted a heat map. In heat maps each variable is in a cell represented by a two-dimensional mark that has been assigned a color. We have chosen a diverging color palette that goes from green to read, representing low and high values respectively. This type of visual encoding really helps to understand high information density data.

Finally, we thought of representing a pair plot of the most relevant quantitative variables according to the heat map to see pairwise relationships in the data. Each color represents a different cluster to which the instances belong. Again the colors (blue,green,red) are randomly assigned to a cluster as in the PC plot. This plot visualizes each numerical variable in the X and Y axes, so that off-diagonal cells contain a scatter plot (below) and the correlation statistics (above) between two variables. The diagonal shows the marginal distribution of the variables differentiating between clusters.

Additionally, we included two different kinds of interactions to the above visual encodings. We used a slider to choose the number of clusters we want the kmeans algorithm to compute and a select box to select which one of the three plots we want to show.

The final visualization and interactions are summed up in Table 4.

	Solution
Visualization	1. k-means cluster plot [3]: 2D plot of the data points according to the first two principal components. 2. Heat map [4]: Heat map of the variables and the cluster to which the instances belong to. 3. Pair plot [5]: Pair plot of the most relevant variables to see pairwise relationships in a data.
Interaction	1. Slider: Select a number of clusters (from 1 to 5). 2. Select Box: Select the type of plot you want (cluster, heat map or pair plot).

Table 4. Clustering interaction and visual encoding.

5 Algorithmic implementation

We used *R* [6], *RStudio* [7] and *Shiny* [8] for the development of this part of the project.

5.1 Trips flow

The algorithmic implementation of the first visualization can be divided in two different parts: preprocessing data and displaying clean data.

5.1.1 Preprocessing

The transformation of the data for this part of the project had to do with the use of aggregation functions. Splitting the data set by week day and hour ($7 \times 24 = 168$ options), grouping trips by zone origin and destination and counting the number of trips for each case was the process followed to get the data needed for the flow map. In the case of the chord diagram we followed the same process, but trips were grouped by borough origin and destination instead of by taxi zone origin and destination.

5.1.2 Display

To get the visualization described in Section 4.1 we mainly used the three following *R* packages: *Mapdeck* [9], *Chorddiag* [10] and *rgdal* [11]. The first one let us show the animated flow map,

the second was was useful to get the shapes of all the taxi zones and the third one allowed us to show a chord diagram. With just a bit of tweaking with this packages and *Shiny*, it was not hard to get the desired visualization.

5.2 Economic stats

5.2.1 Preprocessing

Each of the stats were not explicitly shown in the dataset, so they had to be computed. Each of them followed a different computation process:

- Tips: the variable *tip_amount* always equals 0 unless the client paid with credit card. We first filter the trips in which the payment method was credit card (variable *payment_type* equals 1). After that, the stats can be easily computed thanks to the `aggregate` function from *dplyr* package [12]. This function takes three input arguments:
  1. `x`: the column from which to extract the value, in this case *tip_amount*.
  2. `by`: the common element to aggregate instances. Set to $PULocationID$ for computing tips in the origin zone or *DOLocation* in destination.
  3. `FUN`: the function to apply in the aggregation. It could be either `mean` or `median`.
- Trip length: calculated in a very similar way to tips. The only two differences are that no initial filtering is needed and the argument `x` takes the reference of *trip_distance*.
- Disputes: the percentage of trips that end up in dispute in a given taxi zone can be calculated as $\frac{\#disputes}{\#trips}$. This could be easily implemented with *R*'s base function `tabulate`, which counts the number of occurrences of a value for all the natural numbers comprised between 1 and the highest integer in the sequence: the sequence passed to the function corresponds to the values of *PULocation* in case of wanting to visualize disputes in the origin or *DOLocation* otherwise. For obtaining the denominator, the entire dataset is used, while for the numerator we have to filter so that only trips that ended up in dispute remain (*payment_type* equals 4). In case that $\#trips$ ($\geq \#disputes$) is 0, then the value for that zone is set to `NA`.

  All these operations have an algorithmic asymptotic cost of $O(n)$.

5.2.2 Display

We used *rgdal* [11] for reading the geospatial data and convert it to shape data and *Leaflet* [13] to plot the choropleth map.

The economic variable to visualize, its numerical values and the radio buttons' values are defined as reactive values. For generating the choropleth map, first a map is generated over the city of New York. After that, by observing the input, the color palette is chosen and its bins are defined. Then, polygons are added to the map with shapes defined by the shape data (geographical data), filled with the color that matches the stat value for that zone in the color palette bins and their label is created by combining predefined strings with the the data boroughs, the numerical values of the stats and strings that depend on the selected input.

5.3 Clustering

5.3.1 Preprocessing

The original dataset was a table composed of quantitative and numerical attributes (see Table 1). In order to implement the kmeans algorithm we needed all variables to be numerical. The

first preprocessing step was remove the attributes that were clearly unbalanced in favor of one single value (small variance) and transformed the categorical attributes into numerical. We changed the date and time variables into the number of seconds since January 1st, 1970 at 00:00:00 UTC to have them as numerical values and for the rest we used one hot encoding. Moreover, since the original dataset was very large, we used a random sample of 10000 instances to do the computations.

5.3.2 Kmeans Algorithm

The only machine learning algorithm used in the project was the k-means clustering algorithm. We have chosen this because it is wildly used in R, it is easy to implement and allows to select the number of clusters. Nevertheless, it is very inefficient when use in large data applications. The k-means algorithm has a quadratic time complexity of $O(n^2)$, where n is the input data size [14].
For very large data sets a single run of k-means need a lot of memory and also takes more than a few seconds to conclude. That is why we have used just 10000 random instances, so it would be more efficient in terms of system response and used memory.

6 Validation

Figure 3 shows what are the elements to keep in mind for validating a proposed solution in terms of the interaction and visual encoding and the algorithmic implementation.

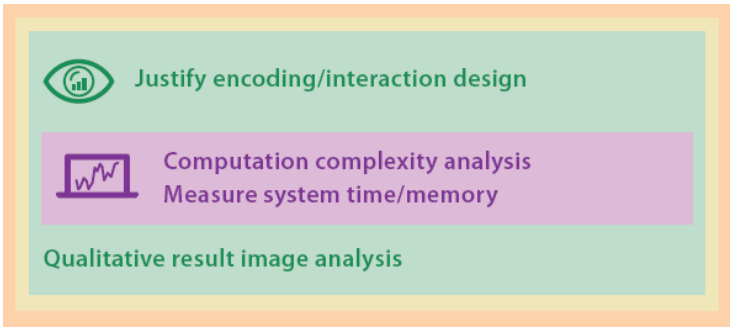

Figure 3. Validation method.

6.1 Trips flow

The first idea of this visualization seemed great, but we faced some problems during the process. The biggest one had to do with discriminability. The width was not enough to distinguish between different lines. Moreover, width is not a good channel if you have more than four different attributes values [1]. However, as we spent a lot of time in this visualization, we tried to find some partial solutions applied: taxi zone selection and 3D navigation. The first one let the user select a particular taxi zone and see the flow this particular area participates in. The second one may have been risky. 3D has to be properly justified, because the perception humans have about depth is not the best. Despite that, we believe in this case it was helpful for the user in order to see trips flow better. These two partial solutions are shown in Figures 4 and 5 and Figure 6 shows the 2D default visualization.

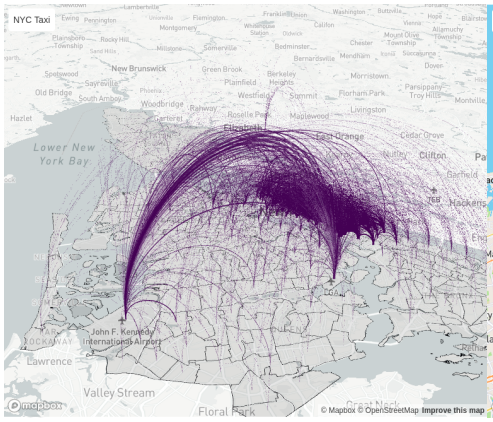

Figure 4. 3D flow map.

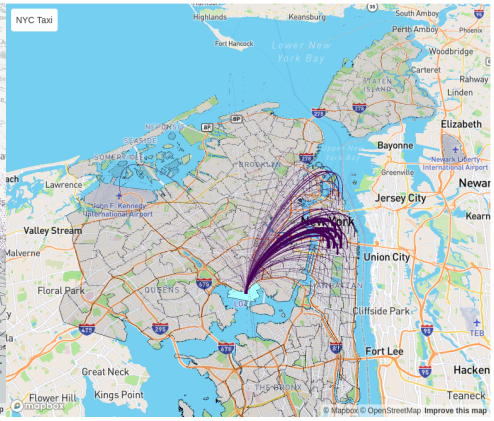

Figure 5. Zone selection.

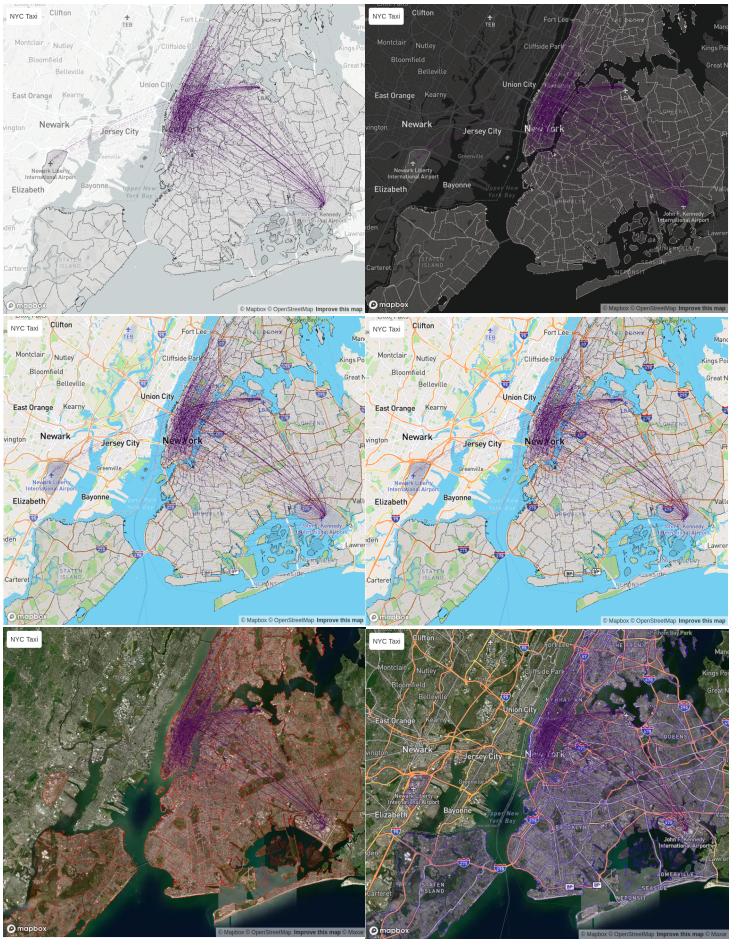

Figure 6. Different 2D visualizations available.

6.2 Economic stats

At first, no color bins in color palette were set. This configuration led to a linear representation of saturation with values, which was not suited to our data distribution. This made most of the values look alike, and only let the user differentiate very high values from the rest.

Solution: creation of color bins.

The program changes its contents in a matter of a few seconds, a time that is completely affordable. Different color tones are distinguishable and easily relatable to the values that they

represent. Labels are readable and correctly display information.

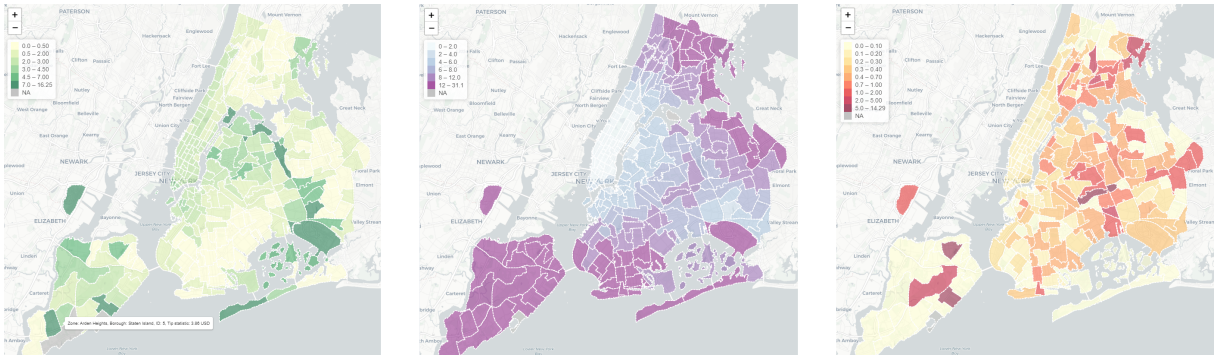

Figure 7. Choropleth maps for mean tip by origin (left), median trip length by destination (center) and percentage of trips that end up in dispute by origin (right)

6.3 Clustering

The three different visualizations obtained for the clustering are shown in Figures 8, 9 and 10. All these images are for just 3 clusters, but the web application enables to chose the number of clusters as well as the type of visual representation the user wants.

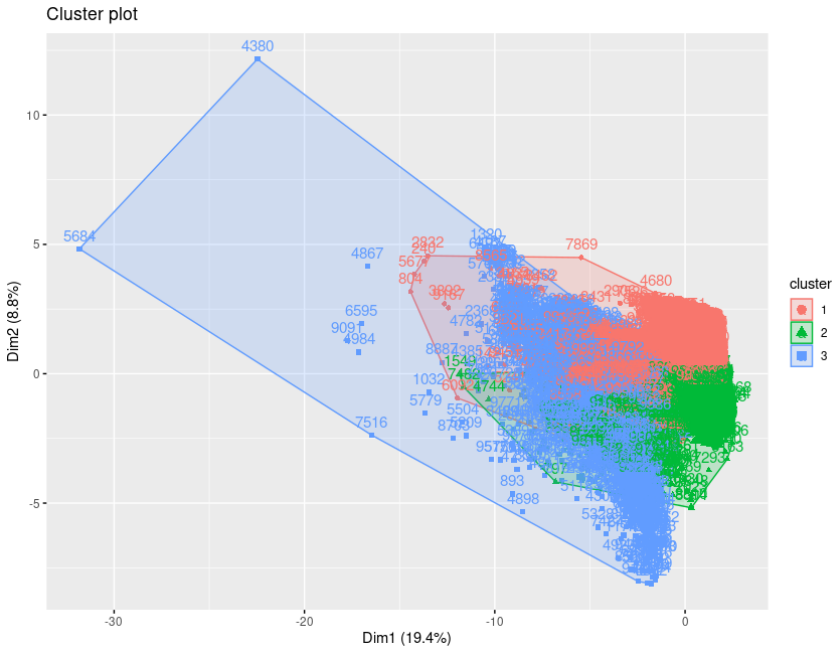

Figure 8. PC Cluster Plot

As we can see from the plots above, for the PC plot and the pair plot the clustered instances are easily differentiated by colors. In the case of the heatmap, the colors also allow us to easily distinguish between values. The pair plot is very huge and for a large number of attributes it is difficult to interpret. That is why we had to just select specific variables for which we wanted to see pairwise correlations.

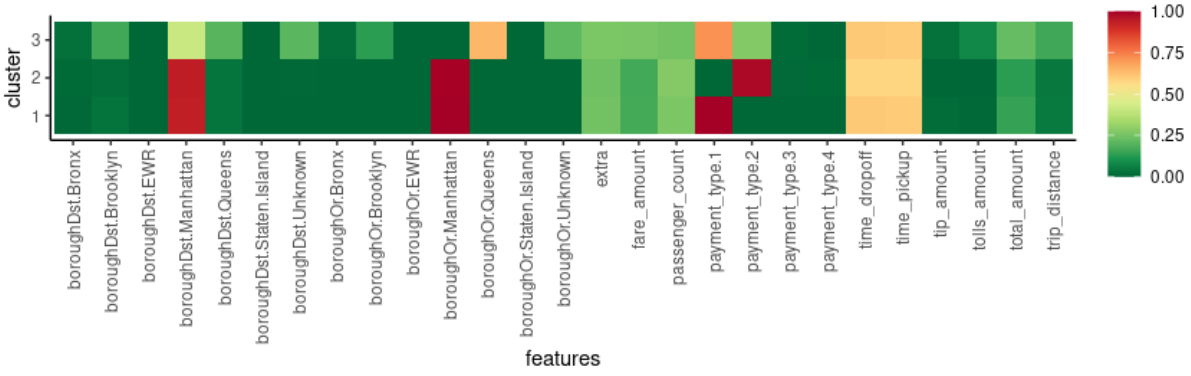

Figure 9. Heat map

Figure 10. Pair plot

7 Shiny App

The final application has been designed to preserve a minimalist layout outside the idioms. A dark sidebar on the left allows to select a tab among the three main topics of the analyst's questions: Trips flow, Economic stats or Clustering. Upon selecting one of them, the app displays one idiom or another depending on the clicked choice on the main page body.

Input options for trips flow (day of week and hour), for economic statistics (what variable, what statistic and which zone) and for clustering (type of plot and number of clusters) are presented in a gray panel between the tab selector, lying on its left, and the chart(s), at its right.

Finally, in the trips flow map interactive elements are displayed below the graph.

To finish, we have used shinyapps.io to share the application as a web page. The link to our web page is here and the access to the source code is here.

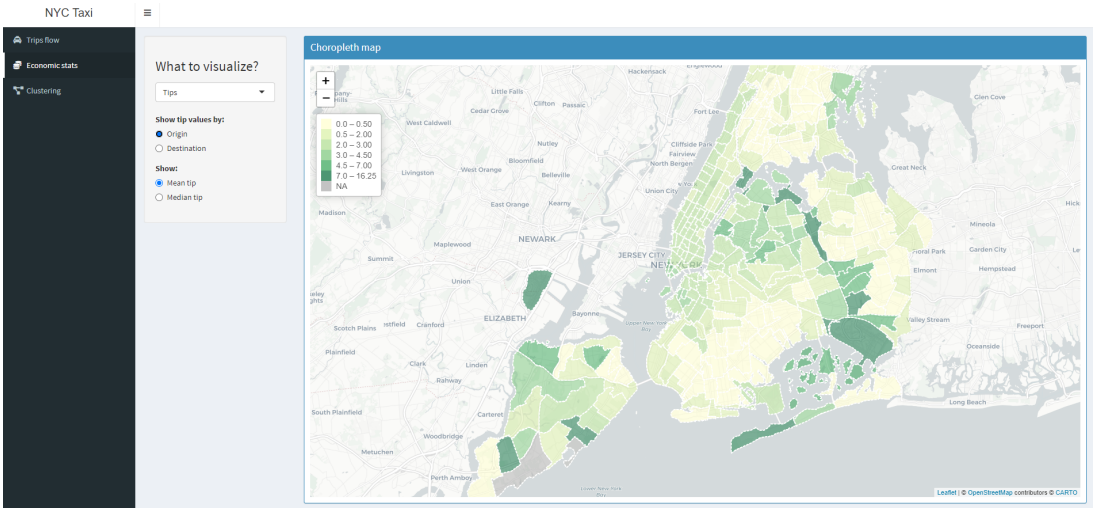

Figure 11. Image of the final Application. We show the Economic stats (mean tip by origin).

8 Conclusions

8.1 Answers to the analyst's questions

1. **How is the taxi trip flow between the different areas of the city according to the different days and times?**

   The visualization derived from this question shows Manhattan as the main taxi area. Inside this borough, Lower Manhattan is the busiest part. The other busy places are the airports. It is also remarkable the flow difference some areas present depending on the week day or time. Some airports have a lot of trips at particular periods of the day and too little at others.

   Another piece of knowledge extracted from this visualization is the best time of the day a taxi driver can work. For example, labour days in the morning (from 8:00 to 8:59) a lot of people take a taxi. We have the same pattern around 14:00 and 20:00. It may seem obvious, because these are entrance to work and out of work moments, but it gets much more clear now.

   Finally, we believe New York City natives could exploit this visualization much more than us. They know better the city, important locations or events and they would take advantage of that. For that reason, we have the opinion that much more knowledge could be extracted from it.

2. **How could different taxi zones relate to the analyst's economy (tips, trips' length and trips that end up in dispute)?**

   (a) Tips are less likely to be given by users that pick taxis outside Manhattan, the closest parts of Brooklyn and Queens to Manhattan and the right half of Staten Island. The latter is a good place to go for generous tips, but no one ensures that you will get some.

   (b) Since nearly every single trips starts or finishes (or both) in Manhattan, longest trip distances by origin are expected to be found as far as possible from the city center.

   (c) The most conflictive travelers are going from/to The Bronx and some parts of Queens. People taking taxis in some parts of Brooklyn, especially around Ocean Parkway, also have more chances to dispute. The data is so scarce for Staten Island that values are

quite extreme. Apparently, some particular zones show a high rate of disputes both in origin and destination. Would it be by chance?

3. **Can I group taxi trips into sufficiently differentiated groups? What are those groups like and what useful information can be extracted from them?**

   Taking all the number of possible clusters into consideration, we have seen that 2 or 3 clusters are the best choices for differentiating groups in the data. In the case of three clusters (Figure 9), we see how cluster one and cluster two are characterized by having Manhattan as the majority of origins and destinations. These two clusters are differentiated for having different payment type, credit card and cash. Furthermore, when looking at the pick up and drop off times, we can see how the for cash payment types the pick up and drop off times of the passengers are by mean sooner in the day than for credit card payments.

   On the contrary, cluster three represents larger trip distances, since it includes more variety of trips involving district changes. We see how for this cluster the main pick up origin is Queens and the payment type is credit card. We can then infer that larger distances are more often paid by credit card and also that they are directly correlated with larger tolls and total amounts.

   Finally, Figure 10 shows the pairwise relationships between total_amount, trip_distance, tolls_amount, and passenger_count. From this plot we can see how total amount and trip distance are linearly correlated. We can more or less associate one of the clusters with larger distances and larger total amount as also seen in the heat map plot. Moreover, we can conclude that the number of passengers is not really related to the trip being more expensive as one could initially think.

   All this information helps the analyst to have a general overview of the type of trips and clients. Also, we can state that having a payment terminal in the taxi is very important, since we have seen how larger trips are usually paid with credit card.

8.2 General conclusions

In this work, we have developed a visualization solution to a data analytics problem regarding recorded taxi trips in New York City. Besides data collection, we have carried out all the steps in the project: from raising questions that might be useful for an end user, to designing an entire application to give answers to these questions by bearing in mind the levels in the visualization design and trying to implement them.

We believe that the application designed helps the analyst answer the questions thrown in the initial part of the process. Although deciding whether or not the results allow drawing firm conclusions is subject to each individual's opinion, what is undeniable is that the proposed solutions permit the end user get proper and interesting insights on the data and find some patterns and solid facts.

By now, the app only displays data for yellow taxis in January 2019. This is because of the large volume of each of the data files with those records: preprocessing all available data files would have been highly time-consuming, while allowing the user to choose an input file would have made the app much slower. This is only a fact that could lead to further improvements for the app, but with a single file, the app is still totally functional and provides reliable results.

All in all, visual analytics has allowed to provide results which are rich in information that, presented in other formats, would have not been as simple, intuitive and effective (and entertaining!).

References

[1] T. Munzner, *Visualization Analysis and Design*, 1st edition. New York: A K Peters/CRC Press, 2014. DOI: https://doi.org/10.1201/b17511.

[2] NYC Taxi and Limousine Commission. "TLC Trip Record Data." (2022), [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page (visited on 01/15/2022).

[3] B. Boehmke. "K-means Cluster Analysis · UC Business Analytics R Programming Guide." (2022), [Online]. Available: https://uc-r.github.io/kmeans_clustering (visited on 01/16/2022).

[4] A. Kassambara. "Heatmap in R: Static and Interactive Visualization - Datanovia." (2022), [Online]. Available: https://www.datanovia.com/en/lessons/heatmap-in-r-static-and-interactive-visualization/ (visited on 01/16/2022).

[5] A. Kassambara. "Scatter Plot Matrices - R Base Graphs - Easy Guides - Wiki - STHDA." (2022), [Online]. Available: http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs (visited on 01/16/2022).

[6] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: https://www.R-project.org/.

[7] RStudio Team, *RStudio: Integrated Development Environment for R*, RStudio, PBC, Boston, MA, 2021. [Online]. Available: http://www.rstudio.com/.

[8] W. Chang, J. Cheng, J. Allaire, *et al.*, *shiny: Web Application Framework for R*, R package version 1.7.1, 2021. [Online]. Available: https://CRAN.R-project.org/package=shiny.

[9] D. Cooley, *mapdeck: Interactive Maps Using 'Mapbox GL JS' and 'Deck.gl'*, R package version 0.3.4, 2020. [Online]. Available: https://CRAN.R-project.org/package=mapdeck.

[10] M. Flor, *chorddiag: Interactive Chord Diagrams*, R package version 0.1.3, 2022. [Online]. Available: https://github.com/mattflor/chorddiag/.

[11] R. Bivand, T. Keitt, and B. Rowlingson, *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*, R package version 1.5-28, 2021. [Online]. Available: https://CRAN.R-project.org/package=rgdal.

[12] H. Wickham, R. François, L. Henry, and K. Müller, *Dplyr: A grammar of data manipulation*, R package version 1.0.7, 2018. [Online]. Available: https://CRAN.R-project.org/package=dplyr.

[13] J. Cheng, B. Karambelkar, and Y. Xie, *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*, R package version 2.0.4.1, 2021. [Online]. Available: https://CRAN.R-project.org/package=leaflet.

[14] M. K. Pakhira, "A linear time-complexity k-means algorithm using cluster shifting," in *2014 International Conference on Computational Intelligence and Communication Networks*, 2014, pp. 1047–1051. DOI: 10.1109/CICN.2014.220.