DESIGN OF A NEW INTERACTIVE DATA ANALYSIS TOOL

Group 10 | Data Visualization

González Ruiz Alberto

Retes Corada Adrián

Torrelles Rodríguez Diego Rafael

DATA SCIENCE MASTER | ETS INGENIEROS INFORMÁTICOS (UPM)

Date: January 8th, 2023

Index

Figure Index

Table Index

1. Introduction

Data visualization has become an essential tool for analyzing and communicating information in a wide range of fields. In this project, we tackle the task of visualizing data related to gas stations in Spain, leveraging a detailed and continually updated dataset obtained from the official portal of the Government of Spain. The methodology followed (Figure 1) in this project is based on a structured approach that spans from dataset selection to the implementation of an interactive application using Shiny, a powerful data analysis tool in R.

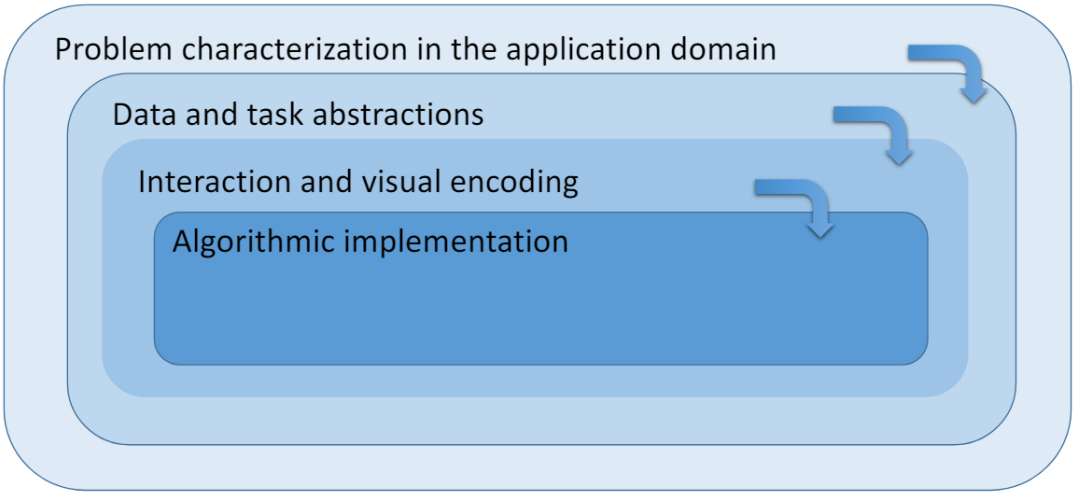

Figure 1. Design visualization nested levels.

In the initial step of our methodology, a careful selection of the dataset was made. From this dataset, a set of key questions were formulated, which will be addressed in this work. To achieve effective answers to these questions, a data and task abstraction approach will be applied, allowing for the proper structuring of data and the definition of specific visualization tasks. Subsequently, the design of a visualization tool will be undertaken, incorporating appropriate visual and interactive elements to effectively address these tasks.

Finally, the implementation of this tool will be conducted in Shiny, enabling analysts and users to interactively explore information about gas stations in Spain, providing a rich and effective data analysis experience.

Throughout this work, we will rigorously follow this methodology to achieve precise, informative data visualization tailored for decision-making, contributing to the understanding and optimization of gas station prices and locations in the Spanish context.

2. Problem characterization in the application domain

In this abstraction level we describe specific issues of the application domain and end users involved, such as the problem to solve, user demands and datasets.

2.1. Selection of Data Set

For this data visualization task, a dataset containing comprehensive information about all the gas stations prices in Spain (Figure 2) has been chosen. This dataset was obtained from the official website of the Government of Spain, specifically at the link "https://geoportalgasolineras.es/geoportal-instalaciones/DescargarFicheros". The notable advantage of this file lies in its frequent price updates, with records being refreshed every 30 minutes.

Figure 2. Fuel prices at a popular station in Spain [1].

2.2. Formulated Questions

The following questions have been formulated with the aim of exploring and analyzing the information contained in the dataset:

1. **What is the geographic variation in fuel prices in Spain for each type of fuel?**

Figure 3. Spain Provinces Map [2].

2. **What is the variation in fuel prices over time?**

Figure 4. Time and money relationship [3].

3. **What is the relationship between the prices of gas stations and their location in cities or on highways?**

Figure 5. Gas station on Spanish highway signal [4].

Question 1 is important because it seeks to uncover differences in fuel prices across different regions of Spain, enabling an understanding of geographic trends in gasoline and diesel prices, among others. As for question 2, this question is the most open; it aims to calculate the statistics to determine the bulk of the price distribution of several types of fuel throughout the country, providing an overall view of average costs for analytics. Finally, the objective of question 3 is to analyze the connection between the road location of gas stations, whether in urban settings or on highways, and the prices they offer. This could shed light on how road location, not geographical location as in the first question, influences price setting.

These questions will establish a solid framework for the development of data visualizations in the Shiny application, allowing analysts to find and answer the problems while exploring and understanding the dynamics of fuel prices in Spain.

3. Data and task abstractions

The primary objective is to translate domain-specific language related to data into generic terms. This involves a structured approach:

- Identification of dataset types.
- Recognition of attribute types within the dataset.
- Determination of cardinality, involving considerations such as the number of items, levels of categorical attributes, and the range of quantitative attributes.
- Evaluation of whether data transformations are necessary or beneficial, including processes like derivation and discretization. This systematic process ensures a comprehensive understanding of the data and paves the way for effective analysis.

3.1. Data abstractions

3.1.1. Dataset type:

The selected dataset is structured, tabular data in spreadsheet format, often referred to as a "CSV" (Comma-Separated Values) and Excel format. This type of dataset is commonly used for tabular data storage, with rows and columns, making it suitable for structured data analysis. It is composed of 11896 instances, each one described by 31 attributes.

3.1.2. Attribute types:

Table 1. Attribute types and description.

Data group	Field Name	Data Type	Description
Geographic Information	Province	Categorical	Represents the province where the gas station is located.
Geographic Information	Municipality	Categorical	Identifies the municipality of the gas station.
Geographic Information	Locality	Categorical	Describes the exact locality of the gas station.
Geographic Information	Postal Code	Ordinal	Represents the postal code of the gas station's location, which can be considered as an ordinal attribute if postal codes reflect a hierarchy or implicit order.
Geographic Information	Address	Categorical	The physical address of the gas station.
Geographic Information	Margin	Categorical	Identifies the exact side of the road.
Geographic Information	Longitude	Quantitative	The precise geographic

			coordinate representing the east-west position of the gas station.
Geographic Information	Latitude	Quantitative	The precise geographic coordinate representing the north-south position of the gas station.
Data gathering	Timestamp	Categorical	Date (day/month/year) and hour (hour:minutes) of data gathering.
Fuel Prices and components	Precio gasolina 95 E5	Quantitative	Represents the cost of gasoline with 95 octane and 5% ethanol at each gas station.
Fuel Prices and components	Precio gasolina 95 E10	Quantitative	Represents the cost of gasoline with 95 octane and 10% ethanol at each gas station.
Fuel Prices and components	Precio gasolina 95 E5 Premium	Quantitative	Represents the cost of premium gasoline with 95 octane and 5% ethanol at each gas station.
Fuel Prices and components	Precio gasolina 98 E5	Quantitative	Represents the cost of gasoline with 98 octane and 5% ethanol at each gas station.
Fuel Prices and components	Precio gasolina 98 E10	Quantitative	Represents the cost of gasoline with 98 octane and 10% ethanol at each gas station.
Fuel Prices and components	Precio gasóleo A	Quantitative	Represents the cost of standard diesel at each gas station.
Fuel Prices and components	Precio gasóleo Premium	Quantitative	Represents the cost of premium diesel at each gas station.

Fuel Prices and components	Precio gasóleo B	Quantitative	Represents the cost of biodiesel at each gas station.
Fuel Prices and components	Precio gasóleo C	Quantitative	Represents the cost of a different type of diesel at each gas station.
Fuel Prices and components	Precio bioethanol	Quantitative	Represents the cost of bioethanol at each gas station.
Fuel Prices and components	Precio biodiésel	Quantitative	Represents the cost of biodiesel at each gas station.
Fuel Prices and components	% bioalcohol	Quantitative	Percentage of bioalcohol in the fuel.
Fuel Prices and components	% methyl ester	Quantitative	Percentage of methyl ester in the fuel.
Fuel Prices and components	Prices of liquefied gases	Quantitative	Prices of liquefied gases at each gas station.
Fuel Prices and components	Prices of compressed natural gas	Quantitative	Prices of compressed natural gas at each gas station.
Fuel Prices and components	Prices of liquefied natural gas	Quantitative	Prices of liquefied natural gas at each gas station.
Fuel Prices and components	Prices of hydrogen	Quantitative	Prices of hydrogen at each gas station.
Gas Station Information	Sign	Categorical	The name or sign of the gas station.
Fuel Prices and components	Sale Type	Categorical	Describes the type of sale at the gas station.
Fuel Prices and components	Remarks	Categorical	Contains additional observations or notes about the gas station.
Fuel Prices and components	Schedule	Categorical	The operating hours of the gas station.
Fuel Prices and components	Service Type	Categorical	Describes the type of service offered by the gas station.

3.1.3. Attributes cardinality

- **Province**: Cardinality equal to 52.

- **Municipality**: Cardinality equal to 3432.
- **Locality**: Cardinality equal to 4244.
- **Postal Code**: Cardinality equal to 4544.
- **Address**: High cardinality (11811). Each physical address is unique.
- **Margin**: Low cardinality (3). Different margins are represented by letters such as "D," "I," "N."
- **Longitude and Latitude**: High cardinality (11811). They have unique values.
- **Data Collection**: Cardinality equal to 2480, multiple data collections conducted simultaneously.
- **Fuel Prices**: Cardinality depends on the fuel; there are 437 different prices for gasoline and 481 for diesel.
- **Sign**: Cardinality equal to 4072. Multiple gas station names are identical.
- **Sale Type**: Cardinality equal to 2. Distinct types of sales, such as "P" (public) or "R" (restricted).
- **Remarks**: Cardinality equal to 2. Different observations or additional notes.
- **Schedule**: Cardinality equal to 1334. Different operating hours.
- **Service Type**: Cardinality equal to 1712. Several types of services offered.

The data was enhanced by incorporating shape data that includes the boundaries of the various provinces in Spain, along with additional information such as their identifier and the corresponding borough. In the subsequent sections, we will elaborate on the distinct transformations applied to the data to implement various idioms.

3.2. Task abstractions

The methodology followed to perform task abstraction encompasses a total of 3 questions (Figure 6):

1. What?: This is related to search and query, involving the identification and definition of key elements.
2. Why?: This is related to questions and targets, aiming to understand the purpose and objectives.
3. How?: This is related to design choices, focusing on the methods and approaches used in the abstraction process

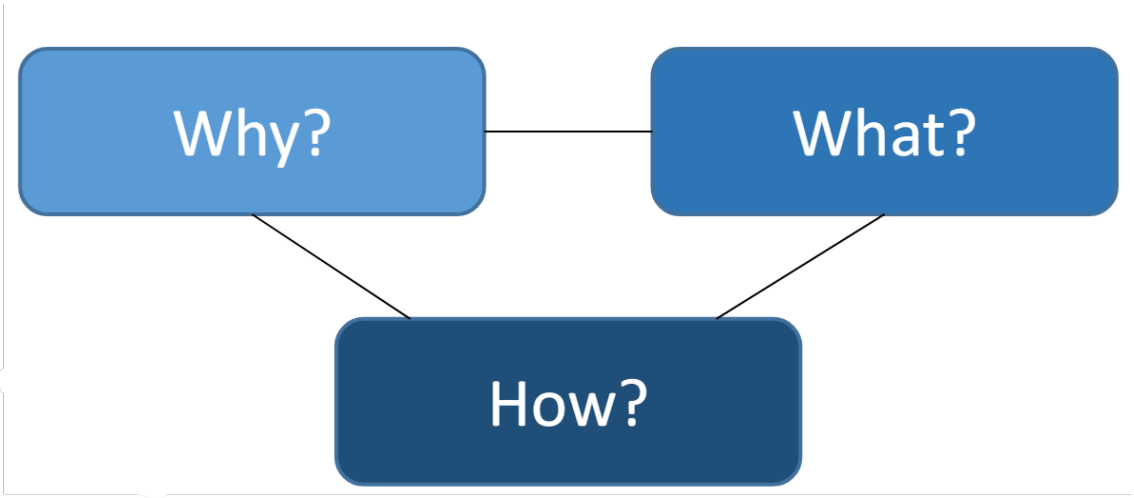

Figure 6. Abstract task, abstract data and views and methods schema.

The idea is to identify *why?* it is needed, *what?* can be achieved, and *how?* it can be implemented with a more general visualization before moving on to a specific case.

In this section, a series of questions related to task abstractions will be presented. For each question, a brief description is provided. Below these descriptions, the corresponding tables summarizing the abstract tasks are presented.

3.2.1. Visualization 1 (Price per regions)

Why is visualization being used?

The visualization is used to present clear and effective information about fuel prices in different provinces of Spain, targeting drivers and fuel distributors. Users can find new knowledge about geographical patterns and mean values in prices that were not presented before in the original dataset. That makes "discover" the main action.

What kind of search is performed based on whether the target and the location are known or not?

Users initiate a search action to locate specific information within the visualization. The patterns are predetermined, they are aware that they are seeking a relation with the price level (target), such as finding an expensive, medium, or cheap regions group. However, users are uncertain about the exact location and where to look to identify the pattern (where it will be discovered in the visualization). In this case, the search will involve "locating".

What kind of query is made based on the results of the previous question?

The three types of queries are made:

- Summarize: Since users have a full view of the map with all provinces and their respective fuel prices, the ‘summarize’ action fits well. Users can use this action to obtain a general overview of fuel prices in all provinces in a concise and understandable manner. It allows them to grasp the data overview without the need to search for specific details or compare multiple targets.
- Identify: The users can Look for a specific province to know the price.
- Compare: The users can check differences in the prices of the regions.

The primary emphasis will center around comparing the outcomes across various regions. This comparative analysis aims to highlight distinctions in fuel prices, providing users with a valuable understanding of the regional variations. By focusing on the comparison of results, users can discern trends, identify disparities, and draw insights into the factors influencing pricing differences among different provinces. In this case, the query will involve "comparing".

What are the different task targets?

The known target that the user is trying to find within an unknown location is the price level relation between provinces (or regions), such as identifying a high price shared by two provinces.

How is going to be performed?

The user is required to navigate within the visualization, selecting a particular fuel type, and choosing provinces for the purpose of comparing prices.

Summarize

Task abstraction for this question would be "discover, locate, and compare prices mean between province zones" (Table 2).

Table 2. Summarize of task abstraction for visualization 1.

What?	Why?	How?
→ Locate → Compare	• Actions: → Discover • Targets: → Price level relationship between provinces.	→ Navigate → Select (Fuel type) . → Select (Province or Provinces).

Beneficial transformation

To effectively implement the concept associated with this visualization, it would be advisable to reduce the dimensionality from the 32 variables to the four primary fuels (Gasolina 95, Gasolina 98, Diesel A, and Diesel A Plus), along with the fuel station coordinates. Subsequently, performing a join with a mapping library containing the coordinates per province would allow the determination of the province in which each fuel station is located. This step is crucial before calculating the average price per province.

3.2.2. Visualization 2 (Price over the time)

Why is visualization being used?

This visualization now focuses on depicting the temporal distribution of fuel prices in Spain, shifting from the previously planned statistical overview. The primary objective is to display how fuel prices fluctuate over time, providing a dynamic perspective on their distribution. Instead of emphasizing general statistical metrics like mean, median, quartiles, and outliers, this second approach aims to offer a visual narrative that captures the evolving trends in fuel prices. The transition to a time-centric visualization, graphically represented, enables a more nuanced exploration of how prices vary across different fuel categories throughout distinct time periods. By unfolding this temporal evolution, the audience gains valuable insights into patterns, seasonality, and potential factors influencing price changes over time. It serves as a powerful tool for discerning trends in both increments and decrements over time. This makes the main action "discover".

What kind of search is performed based on whether the target and the location are known or not?

In this case, determining the pattern involves considering two main factors, as mentioned in the preceding question time and price. Therefore, employing a comprehensive visualization displaying all relevant points at separate times would be beneficial to accommodate the unknown location of the target. This approach allows the user to explore the date and fuel and identify the price point for their analysis within the visualization. The specific target (a price point for a fuel on a date) is known, and the user needs to identify the values in the axis but knows where to look. This makes the search type "lookup".

What kind of query is made based on the results of the previous question?

The two following types of queries are made:

- Identify: The users can Look for a specific time to know the price.
- Compare: The users can look for different dates and fuel types to check the price contrast.

In this case, it will be more useful to choose between specific dates and compare it than to make comparisons between fuel types at the same date, which could be performed by another visualization in a better way. Therefore, the main query will be "compare".

What are the different task targets?

The targets focus on price values, primarily emphasizing factors such as the mean for some fuel type at a specific date. Understanding these measures provides valuable insights into the pricing dynamics and time distribution, offering a comprehensive perspective for analytical purposes.

How is going to be performed?

To conduct the process, it is necessary for the user analyst to identify the appropriate fuel category, the date or dates and retrieve the mean value following the visualization legend and understanding its functioning.

Summarize

Task abstraction for this question would be "discover, lookup, and compare price over time between fuel types" (Table 3).

Table 3. Summarize of task abstraction for visualization 2.

What?	Why?	How?
→ Lookup → Compare	• Actions → Discover • Targets: → Price points between dates and fuels.	→ Select (time range). → Find fuel and dates. → Retrieve point value.

Beneficial transformation

To implement the intended analysis for this visualization, it would be advantageous to reduce dimensionality from 32 variables to the four primary fuels (Gasoline 95, Gasoline 98, Diesel A, and Diesel A Plus) across various documents and dates. Subsequently, calculating the mean for each fuel type and plotting it in the visualization with the corresponding fuel type and date would be essential.

3.2.3. Visualization 3 (Price per road)

Why is visualization being used?

The visualization is used to explore and understand the relationship between gas station prices and the type of road, whether in urban environments or on highways. It allows for the analysis of patterns and trends that may emerge when examining how fuel prices vary in distinct locations, but with a focus on two types of roads, unlike the first visualization. By graphically representing this data, the visualization facilitates the identification of previously unknown correlations between the location of gas stations and price levels, providing valuable insights

into the price dynamics based on geographic location. Therefore, the main action will be "discover."

What kind of search is performed based on whether the target and the location are known or not?

Because the visualization is intended to be simple, with two main types (urban or highway) and four subtypes (fuel types), the user needs to be familiar with the pattern's location, as it will quickly stand out with a value clearly distinguishable from the rest. There are significantly fewer cases than in the first and second visualization, and the type of visualization used will be simpler. Additionally, as in the first and second visualization, the target is also known (the level of the price mean). Therefore, location and target known, the suitable search type will be "lookup."

What kind of query is made based on the results of the previous question?

The primary aim of this visualization is to be as simple as possible, incorporating different road types, fuel categories, and their prices to easily identify outlier values. This means that users can compare prices both within and outside the city for every fuel type they look for. Consequently, the most suitable query type for this purpose is "compare".

What are the different task targets?

The relationship between the means of all fuel prices within a designated road group. This target aims to offer a comprehensive insight into the general cost trends associated with fuel in that specific road type, with a particular emphasis on understanding how fuel prices correlate with the surrounding road infrastructure.

How is going to be performed?

- To perform the process is necessary to determine the encode, manipulations, facets, and reductions in the process.
- The primary steps in creating a visualization include data derivation, encoding, and annotation.

Summarize

Task abstraction for this question would be "discover, lookup, and compare price values between fuel and road types" (Table 4).

Table 4. Summarize of task abstraction for visualization 3.

What?	Why?	How?
→ Lookup → Compare	• Actions → Discover • Targets: → Price level relationship between fuel and roads.	→ Navigate → Select (Price range). → Find fuel and road types. → Retrieve mean values.

Beneficial transformation

To perform the task corresponding to this visualization, it would be beneficial to create a column named "LocationType" that will take the value "AUTOVIA" if the gas station has the word "autovía" or "autopista" in its address. Subsequently, reduce the dimensionality from the 32 variables to the four main fuels (Gasoline 95, Gasoline 98, Diesel A, and Diesel A Plus), and the "LocationType " variable. Finally, calculate the average prices based on the type of fuel and the presence on the highway or urban location.

4. Interaction and visual encoding

We are going to explore proposed solutions for three different challenges related to interaction and visual encoding. This exploration is intended to highlight innovative approaches suggested for addressing each specific problem.

4.1. Choropleth map (Price per regions)

At first, we were certain about using a choropleth map to represent the direct relationship between fuel prices and regions. One of the initial challenges that arose was related to the vast number of prices resulting from a large number of gas stations to contend with. It would be impossible to represent each one adequately without suffering from visual clutter. Therefore, the decision was made to divide the map into regions instead of individual gas stations. This way, the provinces of Spain (including the Canary Islands, Balearic Islands, and the autonomous cities of Ceuta and Melilla) are represented, reducing the number of visual markers from 11896 to 52. Consequently, the province map became a good visualization option.

The second issue lies in the direct representation by province. It is necessary to decide which statistic for the price is most suitable for representation in each province to observe the geographical differences. Similar to the previous case, directly representing the prices of gas stations is a bad idea due to the mentioned visual clutter. Instead, the decision was made to calculate the average prices by grouping gas stations by province and representing that average using a discretization scheme.

The third problem involves this scheme (Figure 7). Typical levels of low, medium, and high have been established, which have been transformed into low and high with subcategories of extreme, intermediate, and medium. This way, we transition from high-medium to low-medium, with the extremes being high-extreme and low-extreme. Finally, a scale from red to green has been implemented, signifying expensive to cheap, respectively, something the average user is already familiar with. To achieve this, guidance from the colorbrewer2.org [5] website was followed to select a divergent color scheme for 6 classes with the codes (red-#d7191c, orange-#fdae61, yellow-#ffffbf, greenA-#d9ef8b, greenB-#91cf60, greenC-#1a9850). Additionally, a thick black border was established to make the boundaries more distinctive. The Canary Islands were also zoomed in to reduce movement on the map.

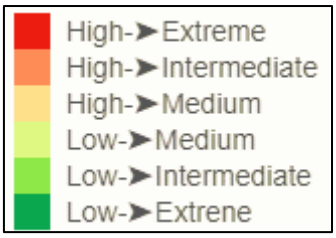

Figure 7. Color discretization schema legend for choropleth map.

The fourth issue arises when the user, in addition to identifying the discrete value through the color on the map, wants to know the continuous value. To address this, a dropdown menu has been added for each province, displaying the province's name and the average price for that province.

The fifth issue is related to the type of fuel being analyzed, given the impossibility of displaying all types simultaneously, which would result in color blending and visual clutter. Therefore, an option of a dropdown menu (Figure 8) must be provided to allow the selection of the fuel type.

To achieve this, the fuel types need to be translated into English since, in the original document, they are presented in Spanish.

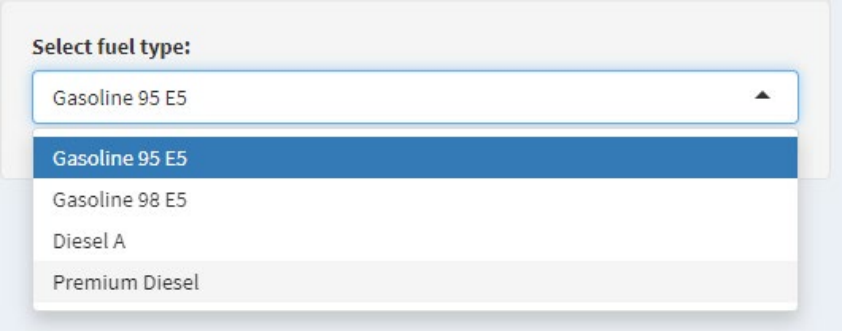

Figure 8. Fuel type selector for choropleth map.

In conclusion, the interface displays a map of provinces featuring a political map of Spain divided into the aforementioned regions with the various color shades discussed earlier. The complementary dropdown menu provides additional information.

Table 5. Solution summary for visualization 1.

	Solution
Visualization	1. Choropleth Map for Spain provinces. 2. Dropdown mean price for province. 3. Fuel selector. 4. Color legend.
Interaction	1. Select the fuel. 2. Navigate on the map (zoom in and out). 3. Locate the province. 4. Visualize the province color. 5. Check the color and dropdown info.

4.2. Line chart (Price over the time)

The first challenge we faced was managing a large volume of fuel prices from numerous service stations. In this case, the direct representation of each station or other values in a line chart would generate visual clutter. To address this challenge, we decided to focus on the temporal evolution of prices. Despite having a large amount of data, we chose to directly represent the values in the line chart, allowing a clear view of price variations over time. This way, the average can be calculated over a period, and price points can be created on the line chart.

The second concern was ensuring that the visual representation was clear and easily comprehensible for trends and variations in prices. In this context, the ability to select the month range becomes essential for locating more specific trends. We generated four distinctive lines on the line chart, each representing a type of fuel (Gasoline 95, Gasoline 98, Diesel A, Diesel A Premium). Each line was differentiated by color, and a descriptive legend (Figure 9) was included for better understanding.

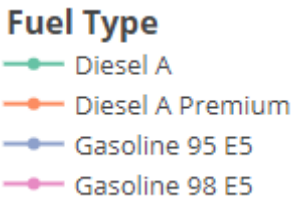

Figure 9. Fuel type legend for line chart.

The third challenge was the limitation of window size and loss of details. Given the window size constraint in the graphical representation, there was a concern that the approximation of values in the line chart might be insufficient, making it difficult to precisely identify the data. To overcome this limitation, we decided to implement interactive annotations (Figure 10) that would appear when the user hovered over a specific point on the chart. These annotations would provide exact details of the fuel price at that point, significantly enhancing accuracy and user interpretability.

Figure 10. Annotations for line chart.

In conclusion, the line chart effectively addresses challenges related to managing a large volume of fuel prices from numerous gas stations. By focusing on the temporal evolution of prices and representing the data on the line chart, we achieve a clear visualization of price variations over time. The use of distinctive lines for each fuel type, accompanied by a descriptive legend, enhances clarity and understanding. To overcome potential limitations due to window size, interactive annotations have been implemented, providing precise details when the user hovers over specific points on the chart. This approach significantly improves accuracy and user interpretability.

Table 6. Solution summary for visualization 2.

	Solution
Visualization	1. Line chart for prices and time. 2. Dropdown mean price for point. 3. Month range selector. 4. Color legend.
Interaction	1. Select the months. 2. Navigate over the time. 3. Locate the time. 4. Visualize the fuel color. 5. Check the point value and dropdown info.

4.3. Bar chart (Price per road)

The first challenge we faced was managing a large volume of fuel prices from numerous service stations. In this case, the direct representation of each station or other values in a bar chart would generate visual clutter. To address this challenge, we decided to focus on the price differences by road type. Despite having a large amount of data, we chose to directly represent the values in the bar chart, allowing a clear view of price variations by road type. This way, the average can be calculated for each road-fuel type, and price bars can be created on the bar chart.

The second concern was ensuring that the visual representation was clear and easily comprehensible for trends and variations in prices. In this context, the ability to select the price range becomes essential for locating more specific trends. We generated four distinctive bars on the bar chart, each representing a type of fuel (Gasoline 95, Gasoline 98, Diesel A, Diesel A Premium). Each bar was differentiated by color, and a descriptive legend (Figure 11) was included for better understanding.

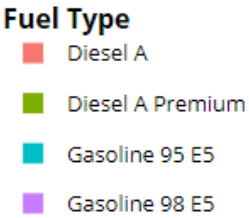

Figure 11. Fuel type legend for bar chart.

The third challenge was the limitation of window size and loss of details. Given the window size constraint in the graphical representation, there was a concern that the approximation of values in the bar chart might be insufficient, making it difficult to precisely identify the data. To overcome this limitation, we decided to implement interactive annotations (Figure 12) that would appear when the user hovered over a specific bar on the chart. These annotations would provide exact details of the fuel price for that bar, significantly enhancing accuracy and user interpretability.

Figure 12. Annotations for bar chart.

In conclusion, the bar chart effectively addresses challenges related to managing a large volume of fuel prices from numerous gas stations. By focusing on the price differences by road type and representing the data on the bar chart, we achieve a clear visualization of price variations by road type. The use of distinctive bars for each fuel type, accompanied by a descriptive legend, enhances clarity and understanding. To overcome potential limitations due to window size, interactive annotations have been implemented, providing precise details when the user hovers over specific bars on the chart. This approach significantly improves accuracy and user interpretability.

Table 7. Solution summary for visualization 3.

	Solution
Visualization	1. Grouped bar chart for prices and road type. 2. Dropdown mean price for bar. 3. Price range selector. 4. Color legend. 5. Interaction
Interaction	1. Select the price range. 2. Navigate over the road types. 3. Locate the road type. 4. Visualize the fuel color. 5. Check the bar value and dropdown info.

5. Algorithmic implementation

We have employed R [6], RStudio [7], and Shiny [8] in developing this section of the project.

5.1. Choropleth map (Price per regions)

The algorithmic implementation of the first visualization involves a systematic process with key steps to ensure effective data representation. This structured approach includes vital stages such as data loading and preprocessing, environment setup, translation function definition, user interface design, server-side logic, and final execution. Each step plays a crucial role in the successful deployment of a Shiny app for visualizing average gas station prices in Spain by province. From loading and cleaning data to dynamically rendering a leaflet map based on user selections, these key steps collectively contribute to a comprehensive and interactive data visualization experience.

1. Data Loading and Preprocessing (GasStation_dataset_load.R):
   - → Loads necessary libraries, ensuring they are installed.
   - → Reads an Excel file ("Excel_Log/preciosEESS_es.xls") containing fuel price data, skipping the first three rows.
   - → Converts comma-delimited values to numeric format.
   - → Writes the cleaned data to a CSV file ("CSV_Log/preciosEESS_es.csv").
2. Environment Setup:
   - → Installs and loads required packages for Shiny, leaflet, mapSpain, stringdist, and dplyr.
   - → Sources an external script ("GasStation_dataset_load.R") to load the dataset.
3. Translation Function:
   - → Defines a translation function translateFuelType to convert fuel type names from Spanish to English.
4. User Interface (UI):
   - → Defines a Shiny app UI with a fuel type selector and a leaflet map for visualization.
5. Server Logic:
   - → Defines server logic to render the leaflet map based on the selected fuel type.
   - → Reads the dataset and calculates the average fuel prices by province.
   - → Groups fuel station information by fuel type.
   - → Homogenizes province names, matches them with spatial data, and categorizes the prices.
   - → Defines color palette and adds polygons to the map with specified attributes (color, border…).
   - → Displays legends and labels on the map based on the selected fuel type.
   - → Incorporates interactive features like zooming and highlighting, and data dropdown.
6. Execution:
   - → Runs the Shiny app, launching the user interface and connecting it to the server logic.

5.2. Line chart (Price over the time)

The algorithmic implementation of the second visualization involves a systematic process with key steps to ensure effective data representation. This structured approach includes vital stages such as data loading and preprocessing, environment setup, translation function

definition, user interface design, server-side logic, and final execution. Each step plays a crucial role in the successful deployment of a Shiny app for visualizing average gas station prices in Spain by province over time. From loading and cleaning data to dynamically rendering a line chart based on user selections, these key steps collectively contribute to a comprehensive and interactive data visualization experience.

1. Data Loading and Preprocessing:
   - → Necessary libraries are loaded, ensuring they are installed.
   - → Several CSV files ("./CSV_Log/preciosEESS_es_19_Sept.csv", "./CSV_Log/preciosEESS_es_19_Oct.csv", "./CSV_Log/preciosEESS_es_19_Nov.csv", "./CSV_Log/preciosEESS_es_19_Dic.csv") containing fuel price data for different months are read.
   - → The read.csv function is used to load the data into the R environment.
2. Environment Setup:
   - → Required packages for Shiny, readr, and plotly are installed and loaded.
   - → An external script ("GasStation_dataset_load.R") is loaded to load the dataset.
3. User Interface (UI):
   - → A Shiny app UI is defined with a start month and end month selector, and a line chart for visualization.
4. Server Logic:
   - → Server logic is defined to render the line chart based on the selected months.
   - → The datasets are read and average fuel prices by type and month are calculated.
   - → Fuel station information is grouped by fuel type.
   - → The color palette is defined, and polygons are added to the map with specified attributes (color, border…).
   - → Legends and labels are displayed on the map based on the selected fuel type.
   - → Interactive features such as zooming and highlighting, and data dropdown are incorporated.
5. Execution:
   - → The Shiny app is run, launching the user interface, and connecting it to the server logic.

5.3. Bar chart (Price per road)

The algorithmic implementation of the third visualization involves a systematic process with key steps to ensure effective data representation. This structured approach includes vital stages such as data loading and preprocessing, environment setup, user interface design, server-side logic, and final execution. Each step plays a crucial role in the successful deployment of a Shiny app for visualizing average gas station prices in Spain by location type. From loading and cleaning data to dynamically rendering a bar chart based on user selections, these key steps collectively contribute to a comprehensive and interactive data visualization experience.

1. Data Loading and Preprocessing:
   - → Necessary libraries are loaded, ensuring they are installed.
   - → The dataset is loaded using an external script ("GasStation_dataset_load.R").

- → The location type of each gas station is determined based on the address. If the address contains “AUTOPISTA” or “AUTOVIA”, the location type is set to “Highway”. Otherwise, it is set to “Urban”.
- → The data is then split into two subsets based on the location type.
- → The average price for each fuel type is calculated for both location types.

2. Environment Setup:
   - → Required packages for Shiny, and plotly are installed and loaded.
3. User Interface (UI):
   - → A Shiny app UI is defined with a price range slider and a bar chart for visualization.
4. Server Logic:
   - → Server logic is defined to render the bar chart based on the selected price range.
   - → A summary dataframe is created with the calculated averages.
   - → A grouped bar chart is created using ggplot, with the location type on the x-axis, the mean price on the y-axis, and the fuel type as the fill color.
5. Execution:
   - → The Shiny app is run, launching the user interface and connecting it to the server logic.

6. Validation

To validate the application, it is necessary to check 2 factors per idiom: is the visualization effective in conveying information? Is the execution time reasonable?

6.1. Choropleth map (Price per regions)

After addressing the issues discussed in section 4.1 the visualization is effective, providing clear discrimination of prices by region. The boundary between regions is well-defined, and the legend is easily accessible. Additionally, the dropdown for price selection is automated, streamlining the process for users unfamiliar with the application's functionality.

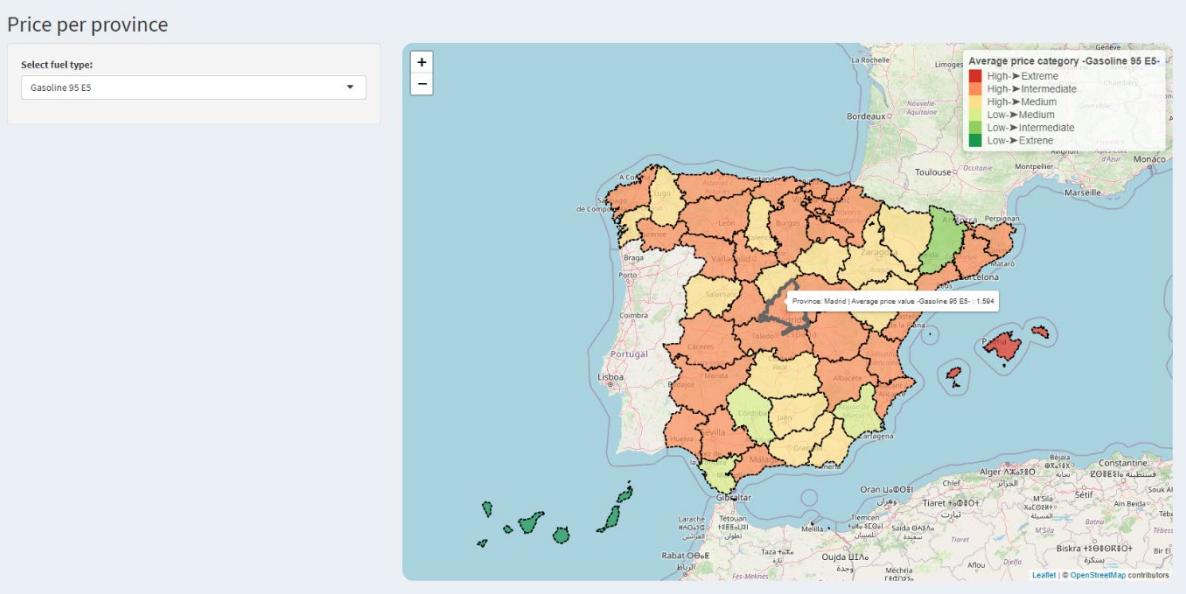

Figure 13. Validated version of choropleth map.

The loading time is 2.75 seconds without preloading in memory (reload time of 1.38 seconds) at this stage of development (local).

6.2. Line chart (Price over the time)

After addressing the issues discussed in section 4.2, the visualization is effective, providing a clear discrimination of prices by fuel type and month. The time axis is well defined, and the legend is easily accessible. In addition, the price range selector is automated, simplifying the process for users unfamiliar with the application's functionality.

The line chart offers a clear visual representation of how the average prices of different fuel types have changed over time. Users can easily select the month range they wish to visualize, allowing for flexible customization of the visualization. This facilitates understanding of fuel price trends over time.

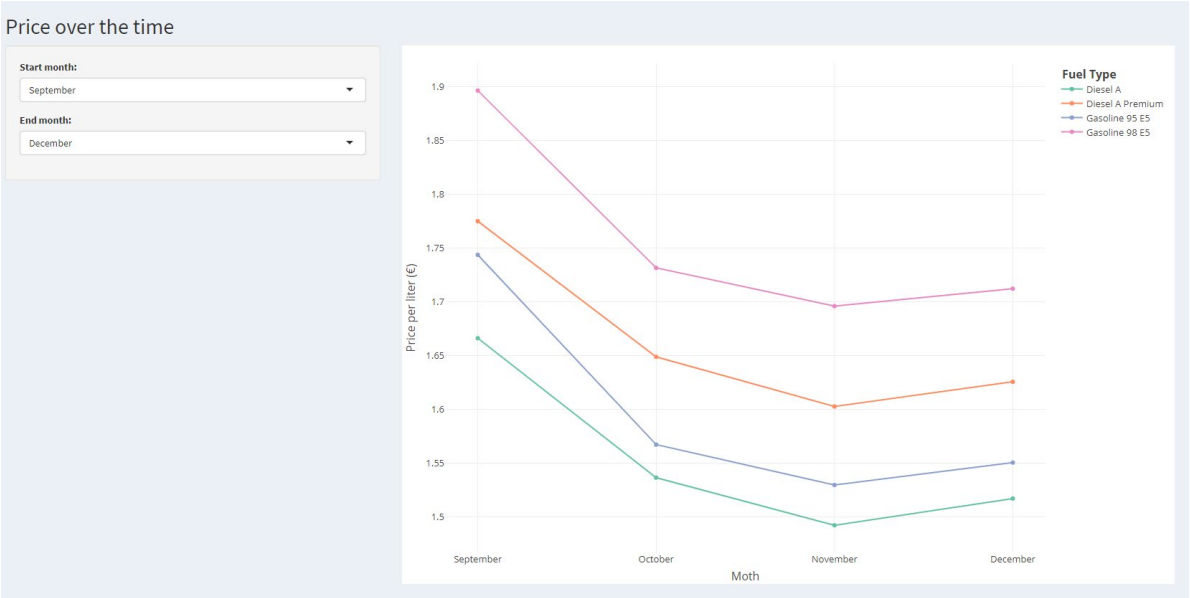

Figure 14. Validated version of line chart.

The approximate loading time for the application at this stage of development (local) is 3.42 seconds without preloading in memory. The reload time is approximately 1.17 seconds. Please note that these times may vary depending on the specific hardware and network conditions.

6.3. Bar chart (Price per road)

After addressing the issues discussed in section 4.3, the visualization is effective, providing a clear discrimination of prices by road and fuel types. The time axis is well defined, and the legend is easily accessible. In addition, the price range selector is automated, simplifying the process for users unfamiliar with the application's functionality.

The line chart offers a clear visual representation of how the average prices of different fuel types have changed based on the roads. Users can easily select the price range they wish to visualize, allowing for flexible customization of the visualization. This indeed facilitates understanding of fuel price trends on different roads.

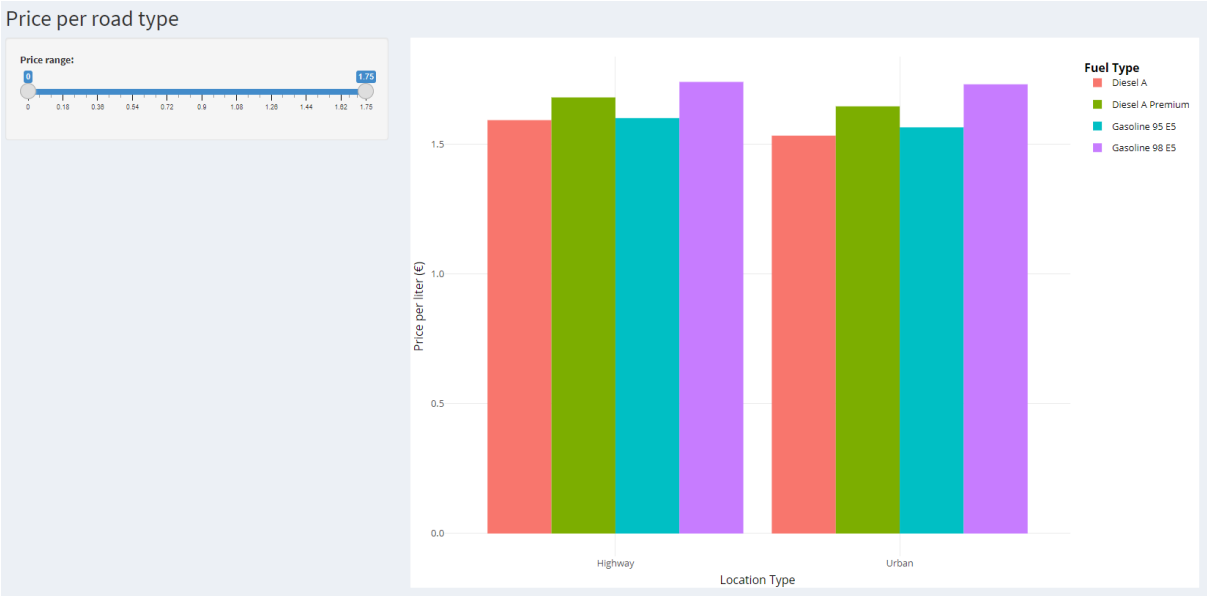

Figure 15. Validated version of bar chart.

The approximate loading time for the application at this stage of development (local) is 2.37 seconds without preloading in memory. The reload time is approximately 1.10 seconds. Please note that these times may vary depending on the specific hardware and network conditions.

7. Shiny app

The ultimate application has been crafted with a minimalist layout beyond the visualizations (idioms). A sleek, dark sidebar on the left (Figure 16) provides the option to choose a tab from the three primary topics related to the analyst's inquiries: Price per region, Price over time, and Price per road. Upon selecting a tab, the app dynamically displays the corresponding idiom within the main page body, deployed individually inside the designated window. This approach ensures a clean and focused user experience tailored to each selected topic.

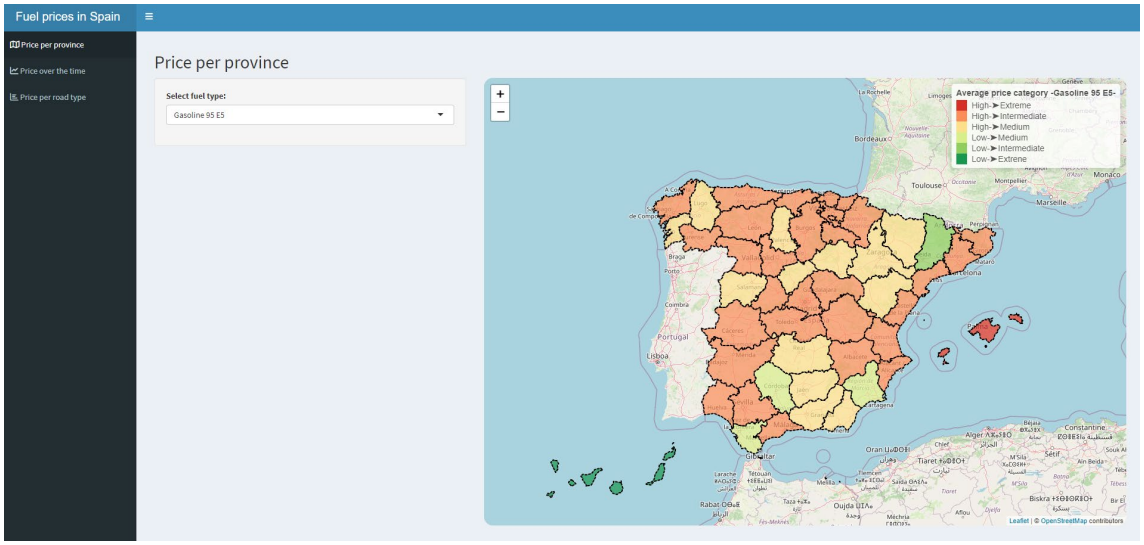

Figure 16. Full shiny app.

To finish, we have used https://shinyapps.io. The links are here:

- To our web page: https://albertodiegoadrian.shinyapps.io/secondexercise_designofanewinteractivedataanalysistool/
- The link to the source code in GitHub: https://github.com/AlbertoGRuiz/SecondExercise_DesignOfANewInteractiveDataAnalysisTool.git

8. Conclusion

The questions from the section 2.2 will be answered in this document section.

8.1. Choropleth map (Price per regions)

What is the geographic variation in fuel prices in Spain for each type of fuel?

Gasoline 95:

- Highest price: Balearic Islands (€1.664).
- High intermediate price: 30 provinces.
- Medium high price: 13 provinces.
- Medium low price: 3 provinces.
- Low intermediate price: Lleida.
- Lowest price: Canary Islands (Las Palmas and Santa Cruz de Tenerife), Ceuta and Melilla (€1.30).

Gasoline 98:

- Highest price: Balearic Islands (€1.82).
- High intermediate price: 29 provinces (€1.82).
- Medium high price: 9 provinces.
- Medium low price: 5 provinces.
- Low intermediate price: 4 provinces.
- Lowest price: Canary Islands (Las Palmas and Santa Cruz de Tenerife), Ceuta and Melilla (€1.40).

Diesel A:

- Highest price: Balearic Islands (€1.665).
- High intermediate price: 28 provinces.
- Medium high price: 11 provinces.
- Medium low price: 5 provinces.
- Low intermediate price: 3 provinces.
- Lowest price: Canary Islands (Las Palmas and Santa Cruz de Tenerife), Ceuta and Melilla (€1.29).

Diesel Premium:

- Highest price: Balearic Islands (€1.736).
- High intermediate price: 31 provinces.
- Medium high price: 11 provinces.
- Medium low price: 3 provinces.
- Low intermediate price: 2 provinces.
- Lowest price: Canary Islands (Las Palmas and Santa Cruz de Tenerife), Ceuta and Melilla (€1.42).

As a general rule, the most expensive fuel is located in the Balearic Islands, and the cheapest in the Canary Islands, Ceuta, and Melilla cities.

8.2. Line chart (Price over the time)

What is the variation in fuel prices over time?

Gasoline 98 is more expensive than Diesel Premium, the latter fuel mentioned before is more expensive than Gasoline 95 and finally, Diesel is the cheapest throughout. Based on time, the most expensive month is September, where there is a decrease until November, in December it continues to increase slightly.

8.3. Bar chart (Price per road)

What is the relationship between the prices of gas stations and their location in cities or on highways?

The price of any fuel is cheaper in urban areas than on the highway.

8.4. General conclusions

The Shiny application project has proven to be a success in visualizing the average gas station prices in Spain. Through three visualization idioms - a choropleth map, a line chart, and a bar chart - we have been able to effectively represent fuel price data by province, road type, and fuel type.

1. Choropleth Map: This idiom allowed for an intuitive geographical visualization of fuel prices by province, effectively highlighting regional differences.
2. Line Chart: This idiom provided a temporal representation of fuel prices, allowing users to see how prices fluctuate over time.
3. Bar Chart: This idiom offered a direct comparison of fuel prices in urban areas and highways, highlighting price differences based on location.

The Shiny application has proven to be a powerful tool for data visualization, allowing user interactivity and dynamic updates based on user selection. Despite initial challenges in data loading and preprocessing, as well as in environment setup, the project has achieved its goal of providing a complete and interactive data visualization experience.

This project has demonstrated the power of data visualization to transform large datasets into understandable and actionable information. With the Shiny application, we have been able to bring fuel price data to life, providing a platform for data exploration and insight discovery.

References

[1] Moveo, "Estas son las gasolineras más baratas en España, según la OCU", La Vanguardia, 02-oct-2020. [Online]. Available in: https://www.lavanguardia.com/motor/rankings/20201002/33621/son-gasolineras-mas-baratas-espana-ocu.html.

[2] " File:Provinces of Spain.Svg", Wikimedia.org. [Online]. Available in: https://commons.wikimedia.org/wiki/File:Provinces_of_Spain.svg.

[3] Blogspot.com. [Online]. Available in: https://2.bp.blogspot.com/-n_PbE_DdqNI/Ue2RGwy_vkI/AAAAAAAAANA/BHiAsSpZ5wQ/s1600/tiempo+y+dinero.jpg.

[4] N332.es. [Online]. Available in: https://n332.es/wp-content/uploads/2022/03/Finding-Fuel-on-the-Motorway.jpeg.

[5] "ColorBrewer: Color advice for maps", Colorbrewer2.org. [Online]. Available in: https://colorbrewer2.org/.

[6] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2023. [Online]. Available: https://www.Rproject.org/.

[7] RStudio Team, RStudio: Integrated Development Environment for R, RStudio, PBC, Boston, MA, 2023. [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, 2023. [Online]. Available: https://CRAN.R-project.org/package=shiny.