236369

Winter 2019-2020

Managing Data on the World-Wide Web

Assignment 3 – Frontend

In this assignment you will build a page that analyzes and visualizes data and statistics.

Assignment is 15% Takef.

Submission date: 23.12.2019 23:55 PM

Introduction

An important skill when dealing with data analysis and research visualization and display both the data and the analysis' results. This can be easily done in web programming using different tools.

This assignment focuses on frontend programming. You will implement a page that visualizes and analyzes data using frontend tools: HTML APIs, JavaScript libraries and CSS.

Dataset

You are given a (partial) Michelin restaurant dataset taken from Kaggle. The data is separated into three **csv** files, according to the number of Michelin stars.
The structure of the data is as follows:

- Name – the restaurant's name
- Year – the awarded year
- Latitude – the restaurant's latitude
- Longitude - the restaurant's longitude
- City – the restaurant's city
- Region – the restaurant's region according to the Michelin guide
- zipCode – the restaurant's zip code, if exists (otherwise N/A)
- Cuisine – the restaurant's cuisine style
- Price – the restaurant's price range, represented by "$" signs, N/A means unknown price range

- URL – the restaurant's URL in the Michelin guide site

Displaying and Analyzing the Data

Data as Interactive Table

The first step is to display the data as an interactive table, allowing exploration and better view of the data. The table should have the following capabilities:

- Sorting – can sort the data by **any** column
- Filtering – filter restaurants by name
- Pagination – display 10 rows per page, and enable pagination
- Download data – be able to download the data as CSV

Of course, you aren't expected to implement everything yourself. We recommend using the **Tabulator** JS library which enables doing all the above easily.

You can display 3 separate tables or add the number of stars to the table programmatically, **do not** change the data itself as you do not submit it.

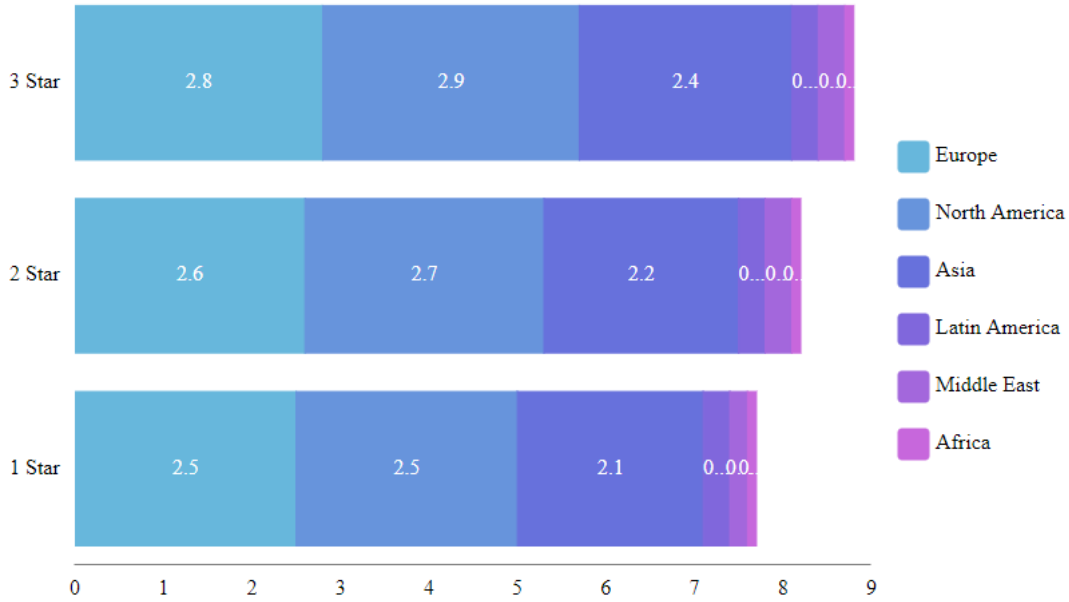
Basic Analysis

Next, we want to gain some basic (but interesting) insights about our data – we can do it by visualizing certain statistics.
The most interesting properties in our case are:

- Cuisine
- Price range
- Region

You will do this visualization using **amCharts** which is another JS library, specifically for charting and visualizing data.
You are required to implement the following (For the whole data as one):

- A **pie chart** for the prices, displaying the relative part of each price range.
- A **stacked bar chart** for the regions, displaying for each rating how many restaurants from each region has this rating.
  An **example** for how it should **look** like (the data isn't real):



- A **column chart** for the cuisines, displaying the relative part of each cuisine.

Mapping

Now, we want to use the geographic information to view the restaurants. For this part you will use **Leaflet**, a JS library for mapping and geo-information.
You are required to implement a map, where the user can decide which restaurants to display (as markers on the map) according to their number of Michelin stars, E.g. the user can choose one of the following:

- Display all – display all the restaurants markers
- Hide all – hide all the restaurants markers
- Show 1 star – show only restaurants with exactly 1 Michelin star
- Show 2 star - show only restaurants with exactly 2 Michelin stars
- Show 3 star - show only restaurants with exactly 3 Michelin stars

In addition, the map should support **regional** search – when the user inputs a region the map will show only the restaurants in that region. When the input is empty you should display all the restaurants. No need to handle non-existent regions.

Advanced Analysis and Visualization

For the final step, we want a more advanced analysis of the data such as the relationship between different features. First, you are required to visualize the relation between **cuisine** and **average price**, i.e., display each cuisine's average price, in an informative chart of your choice.

Next, you are required to create **one** additional graph\chart that explains your own insights regarding the data. ***You are free to choose how to***

display your conclusions! However, they must be non-trivial. Attach a text file containing a short explanation of the choices you've made. Spherically the file should explain the following:

- How you've reached your conclusions.
- Your choice of display.

Note that interesting work will be rewarded with a bonus of up to 3 points.

Finishing Touches

You are required to display your results in a **single page**, like a reactive application. The flow should be simple and **user friendly**, you don't need to implement anything too complicated.

Your application is required to have a CSS design in a separate (non-empty) file. The purpose of this is to have you practice CSS coding and design. Extensive and creative design will be rewarded with a 1-point bonus.

Getting Started

Installing Libraries

You can include all the libraries as follows:

- Tabulator –
  - <link href="https://unpkg.com/tabulator-tables@4.0.5/dist/css/tabulator.min.css" rel="stylesheet">
  - <script type="text/javascript" src="https://unpkg.com/tabulator-tables@4.0.5/dist/js/tabulator.min.js"></script>
- Leaflet –
  - <link rel="stylesheet" href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
  - <script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
- AmCharts –
  - <script src="https://www.amcharts.com/lib/4/core.js"></script>
  - <script src="https://www.amcharts.com/lib/4/charts.js"></script>

Using the Libraries

Although we see these libraries in class, they are simple to use, and have an extensive documentation and examples.

Tabulator – very simple to use, you can find various examples and documentation here.

amCharts – a bit more complex, with a lot of options. You can find examples (with their code) here. Note that you don't need to know and understand every detail, just stick to the relevant examples.

Leaflet – somewhere in-between the above complexity-wise. You can find here everything you need. Again, not all features are required for this assignment.

Tips and Guidelines

- Your graphs and charts should be informative and cohesive, you are also required to add a title for the charts.
- You can work on each part separately and connect them all at the end.
- Use the given links! Almost everything you need to implement becomes easy if you understand how to use the libraries.
- The single page can be very simple, don't waste your time trying to implement very complicated control-flow logics.

Submission

You should submit a zip file named ***hw3.zip*** containing the following files:

- hw3.html – the HTML file for your application.
- hw3_design.css – the CSS file for your application.
- hw3_exp.txt – the text file containing the explanation in the advanced analysis section. This file should be in **English.**
- Any other files you need for your application.

Good Luck!