Project 6 Overview

For this project, you will create a data visualization from a data set that tells a story or highlights trends or patterns in the data. You will need to use either dimple.js or d3.js to create the visualization. Your work should be a reflection of the theory and practice of data visualization, such as visual encodings, design principles, and effective communication.

Prepare for this project with: Data Visualization.

Note

If you have successfully completed the project for the Data Visualization course in the past (which entails having graduated from the course and having access to your course certificate), simply email us at dataanalyst-project@udacity.com with your passing evaluation and we'll give you credit for this project.

What do I need to install?

To work on your data visualization, you will need to start a local server on your computer. To start a local web server, you will need to have Python 2.7.8 or higher installed on your machine.

If you do not have Python installed on your machine, please watch the instructions for Downloading Python. These instructions come from the Programming Foundations with Python course.

Once you have Python installed, you can start a local web server and view your data visualization. Refer to the following video to see how to do so.

Remember, you must start your web server in the top level directory to serve all code and data files. If you do not use this folder as the root directory for the web server, be aware that you will need to change the file paths.

There are other ways to start a local web server. To learn more about why you need to start a local web server and other ways of setting up a local web server, please read Setting Up A Local Web Server from Scott Murray's book, Interactive Data Visualization for the Web.

Why this Project?

This project will touch on the overarching attitudes and beliefs important to effective data visualization, such as:

- visualization is a dialog
- showcasing and sharing visualization with others
- visualization is a fluid process that typically requires multiple iterations of improvements

You will have an opportunity to experience the end-to-end process of creating effective data visualizations and highlighting important information from data that may otherwise be hidden or hard to uncover.

What will I learn?

After completing the project, you will be able to:

- Demonstrate the ability to choose optimal visual elements to encode data and critically assess the effectiveness of the visualization
- Communicate a story or finding to the appropriate audience using interactive visualizations
- Undergo the iterative process of creating a visualization, and build interactive visualizations with dimple.js or d3.js.

Why is this Important to my Career?

Data analyst are storytellers that can translate data findings that other people can easily understand. They view data visualization as an important form of communication.

If you, as a data analyst, can create visualizations to explore data, articulate clear findings to drive business decisions, or use data to elicit consensus from diverse perspectives, then you will be a deeply invaluable member on your team.

How do I Complete this Project?

This project is connected to the **Data Visualization** course, but depending on your background knowledge of data visualization, **dimple.js**, and **d3.js** you may not need to take the whole course to complete this project.

After completing Lesson 2 and Problem Set 2 of the course, you will be able to complete this project since you will have learned about dimple.js.

If you want to become more technical and expand your skill set, you can continue to Lesson 3 and Lesson 4, in which you will learn more about narrative structures and how to create graphics using d3.js. The d3.js library has a steeper learning curve, and we encourage you to take on the challenge if you desire.

The process for evaluating your project is not affected by your choice of using **dimple.js** or **d3.js**.

Introduction

For the final project, you will create an **explanatory** data visualization from a data set that communicates a clear finding or that highlights relationships or patterns in a data set. Your work should be a reflection of the theory and practice of data visualization, and you must use either **dimple.js** or **d3.js**.

We will provide some options of data sets to explore; however, you may choose to explore an entirely different data set. You should be aware that finding your own data set and cleaning it using Python, R, or some other language can take considerable time and effort. This can add as much as a day, a week, or even months to your project so embark on the adventure to find and clean a data set if you are truly prepared with programming and data wrangling skills.

You have three options for this project. You should pick an option based on your prior experience with data munging and exploratory data analysis. The option you choose will not affect the evaluation of the project.

- **Option 1**
  Select one of the beginner data sets, which already has a summary of findings, from the **Data Set Options**document. Then, create a visualization that communicates the findings.

- **Option 2**
  Select one of the intermediate data sets from the **Data Set Options** document. You will investigate the data set to share a story or message about the data and then create a suitable visualization.

- **Option 3**
  Find a data set, investigate it, and share your findings in a visualization. Your final graphic should primarily be explanatory, but it may also contain exploratory components. You can find a list of recommended websites to find data sets in the **Data Set Options** document. You should be aware that finding your own data set, cleaning the data set, and analyzing it (using R, iPython Notebook, or another tool) can take considerable time and effort. This can lengthen the time you spend on your project by days, weeks, or even months. Choose the option only if you feel prepared for a challenge!

Now, on to the details!

Step One - Choose a Data Set

First, you will choose a data set from the **Data Set Options** document or find a data set to explore and visualize. You should choose a data set based on your prior experiences in programming and working with data. The data set you choose will not increase or decrease your chances of passing this project.

Step Two - Get Organized

Eventually you'll want to submit your project and share it. If you are familiar with **GitHub**, we encourage you to create a public repository or a public **Gist** for your project to track changes. Otherwise, you need to create the following files.

- an **index.html** file containing the code to create your visualization (you may include the JavaScript and CSS in this file or separate them in other files)
- a **README.md** file that includes four sections...
    - **Summary** - in no more than 4 sentences, briefly introduce your data visualization and add any context that can help readers understand it
    - **Design** - explain any design choices you made including changes to the visualization after collecting feedback
    - **Feedback** - include all feedback you received from others on your visualization from the first sketch to the final visualization
    - **Resources** - list any sources you consulted to create your visualization
- **data files**
    - the final data set used to create the visualization (usually .csv, .tsv, or .json file)
    - a codebook or other files related to the data set (description, readme, license)
- OPTIONAL FOLDERS IF YOU USE **GITHUB**
    - **data** folder to include all the data related files
    - **js** folder to include .js files (not needed if javascript is in the index.html file)
    - **css** folder to include .css files (not needed if CSS is in the index.html file)

Step Three - Find a Data Story

Explore your data set and craft a message or story around your data! Think about the overall message you want to convey and think about the comparison(s) or relationship(s) you want your readers to see.

Step Four - Create Your Visualization

First, sketch ideas for your visualization. Once you settle on a sketch, explain any design choices in that sketch, such as chart type, visual encodings, and layout, in the **Design** section of the **README.md** file. Then, write code to create your visualization using either **dimple.js** or **d3.js**. The visualization must include animation, interaction, or both. See the **Project Rubric** for more information.

Step Five - Get Feedback

Share your visualization with **at least 3 other people** and document their feedback. There are many ways to get feedback, and more feedback is generally better! Here are some options. - Share your visualization with others in person and have them think aloud as they read and explore the graphic so you can document what stands out to them and how they interpret the graphic. - Share a link to your repository in the discussions and ask others to share constructive criticisms. Be sure to offer advice to others who are seeking feedback too!

- Create and share a **Gist**, which contains an **index.html** file, data file, and any .js or .css files). Directions for creating and sharing a Gist can be found at **http://bl.ocks.org/**.
    - **Box Plots Gist EXAMPLE**:
        - https://gist.github.com/mbostock/4061502
        - http://bl.ocks.org/mbostock/4061502

You might need to ask specific questions to prompt the reader. Here are some questions to help you. You can, of course, ask others.

- What do you notice in the visualization?
- What questions do you have about the data?
- What relationships do you notice?
- What do you think is the main takeaway from this visualization?
- Is there something you don’t understand in the graphic?

Step Six - Document Feedback and Improve the Visualization

For each person that gives you feedback, add the person’s feedback to your **README.md** file in the *Feedback*section. As you improve and iterate on your visualization, update your code **AND** describe any changes in the*Design* section of the **README.md file**.

You should save multiple versions of your data visualization after you make changes to it. You can do this using GitHub or a Gist by making commits to your project, or you can simply save multiple version of you data visualization such as index1.html, index2.html, … , index_final.html. Remember to save related files with similar numbers…

- main1.js, main2.js, … , main_final.js (if you separate your Javascript from the HTML file)
- style1.css, style2.css, … , style_final.css (if you separate your styling from the HTML file)

When should you save your files? You should save your files whenever you have a working version of your data visualization. If you get feedback and make changes, then wait to save the file until you have the data visualization working as you want it. Your goal is to build evidence that you have shared your visualization, received feedback, and responded to that feedback. You will need to submit the different versions of your visualization. **At a minimum, you need to submit an initial version of your data visualization (either as a sketch or as code) and the final index.html file and related files.**

Step Seven - Review

Use the **Project Rubric** to review your project. If you are happy with your submission, then you're ready to submit your project. If you see room for improvement, keep working to improve your project!

Evaluation

Use the **Project Rubric** to review your project. If you are happy with your submission, then you are ready to submit! If you see room for improvement in **any** category in which you do not meet specifications, keep working!

Your project will be evaluated by a Udacity reviewer according to the same **Project Rubric**. Your project must "meet specifications" or "exceed specifications" in each category in order for your submission to pass.

Submission

Ready to submit your project? Go back to your Udacity Home, click on the project, and follow the instructions to submit!

- You can either send us a GitHub link of the files or upload a compressed directory (zip file).
- Inside the zip folder include a text file with a list of Web sites, books, forums, blog posts, GitHub repositories etc that you referred to or used in this submission (Add N/A if you did not use such resources).

It can take us up to a week to grade this project, but in most cases it is much faster. You will receive an email when your submission has been reviewed.

If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com.

What to include in your submission?

1. the original index.html file for the first version of your graphic
2. the final index.html file for the final version of your graphic
3. the README.md file with the sections Summary, Design, Feedback, and Resources
4. the final data set file used for the graphic (usually .csv, .tsv, or .json)
5. A list of Web sites, books, forums, blog posts, github repositories, etc. that you referred to or used in creating your submission (add N/A if you did not use any such resources).
6. OPTIONAL: additional versions of your index.html as you iterated on your visualization based on feedback (index1.html, index2.html, index3.html, ... , index_final.html)

Data Visualization Project Rubric

Overview

This rubric is here to help you understand the specifications for the data visualization that you create. It is the same rubric that the person evaluating your project will use. We will refer to this person as the "project evaluator" in this document. You should look at the rubric **before you begin working** on your analysis **and before you submit it**.

How to Use: before you begin

1. Look at the bold headings under the criteria column to understand what the project evaluator will be looking for in your project.
2. Go through each criteria item in more detail.
3. Familiarize yourself with what is required for your project to "meet specifications" or to be "completely Udacious." In order to gain a certificate, you need to "meet specifications", however, to gain the most benefit and learn most from the experience, we encourage you to aim for "completely Udacious".

How to Use: before you submit

1. Once your project is complete, go through each criteria item and do your best to honestly evaluate where you think your project falls.
2. If you think your project does not meet specifications for **any** criteria item, then you should make some changes to your analysis.
3. Once you’re confident that your project "meets specifications" or is "completely Udacious," go ahead and follow the Project Submission Instructions to submit!

How Grading Works

1. Your project evaluator will use this rubric to evaluate your analysis.
2. Your grade will simply be "pass" or "doesn't pass."
    1. You earn a "pass" by having **all** criteria items in the "meet specifications" column.
    2. If any criteria item does not meet specifications, you will not pass. You will be able to make changes and re-submit the project.

The Rubric

Criteria	Meets specifications	Exceeds specifications (Completely Udacious)
Code Structure and Functionality		
Does the code work?	The visualization renders and any interactions or animations work as the reader interacts with the visualization.	***Not Applicable***
Is the code in the `index.html` commented in a way that is useful and not excessive?	Large code chunks are commented, and all complex code is adequately explained with comments. Comments are not overused to explain obvious code.	***Not Applicable***
Does student's code use formatting techniques (indents, spaces, line breaks, etc...) to improve readability? (Refer to Google's Style Guide for Javascript)	The code uses formatting techniques in a consistent and effective manner to improve code readability.	***Not Applicable***
Visualization is Explanatory		

Does the visualization have a clear finding?	The visualization centers on a specific, clear finding in the data.	***Not Applicable***
Does the visualization focus on its finding?	The selected finding is clearly communicated. Design choices foster communication between the reader and the visualization.	
Design		
Does the written summary reflect what a reader would interpret from the graphic?	A reader's summary of the graphic would closely match the written summary in the README.md file, or a reader would identify at least 1 main point or relationship that the graphic attempts to convey.	***Not Applicable***
Does the data visualization incorporate interaction or animation?	The visualization includes interaction or animation. The interaction or animation may be simple, such as a hover, tooltip, or transition. Interaction or animation enhances understanding of the data.	The data visualization incorporates more advanced techniques beyond the scope of the class. The techniques enhance the reader's ability to understand the data and interact with the graphic.
Are initial design decisions documented?	The student explains initial design decisions such as chart type, visual encodings, layout, legends, or hierarchy. These are included at the beginning of the Design section in the README.md file.	***Not Applicable***
Feedback and Iteration		
Does the student collect feedback after sharing the initial visualization? *We encourage you to collect feedback from the first sketch to the final visualization.*	The student collects feedback from at least three people throughout the process of creating the data visualization. The feedback is documented in the Feedback section of the README.md file.	The student collects feedback from many people throughout the process of creating the data visualization. The student provides other evidence of feedback, such as screenshots with annotations, audio files,

		videos, discussion forum links, or images of sketches with handwritten comments.
Does the student iterate on the visualization? **Does the student incorporate feedback to improve the visualization? If not, does the student explain why the design of the visualization did not change?**	The student presents evidence that the visualization has been improved since the first sketch or the first coded version of the visualization. The student has listed all of the feedback in the Feedback section of the README.md file. Most design choices and changes are accounted for in the Design section of the README.md file. If no changes were made to the visualization after gathering feedback, this decision is explained.	The student presents overwhelming evidence that the visualization has been improved. The student has listed all of the feedback in the Feedback section of the README.md file. All design choices and changes are accounted for in the Design section of the README.md file.