



A Night at the Museum: Exploring the MoMA Data

ABSTRACT

The Museum of Modern Art (MoMA) makes their artist and artwork databases publicly available on GitHub. This paper details the work of Team Arts & Entertainment in their development of a browser-based application that allows users to explore the data using interactive visualizations.

Team Arts & Entertainment:

Conor Lynch, Sean Shepherd, Carrie Wu

NORTHWESTERN UNIVERSITY SCHOOL OF PROFESSIONAL STUDIES

IDS_455 DATA AND TEXT VISUALIZATION

August 27, 2017

Introduction

The Museum of Modern Art (MoMA) opened its doors in 1929, the vision of three patrons who wished to shake up the staid and conservative view of traditional museums. The goal, according to its founding director, Alfred H. Barr, was to help people enjoy the visual arts of their time, and not just masterpieces of centuries before.¹

Though the museum began in the 20th century, the definition of modern art finds its origins in the Industrial Revolution of the 18th and 19th centuries, as factories took over from farms and cities grew at a rapid pace.² New ideas blossomed in economic, social, and cultural environments, including the art world. The age of commissioned artwork, religious and mythological imagery, and exaggerated romanticism gave way to experimentation in colors, materials, and techniques. With few exceptions, the MoMA has collected art that was produced after 1880, the museum's definition of the beginning of the modern age in art.³ The collection now holds over 200,000 paintings, sculptures, drawings, photographs, prints, and other works of art.⁴

In 2015, the museum decided to share its dataset of 125,000 pieces of art on GitHub using a Creative Commons Zero (CC0) license.⁵ The terms of this license allow owners of protected content to “waive those interests in their works and thereby place them as completely as possible in the public domain.”⁶ MoMA applied the CC0 license to the dataset only, and not the works within museum. However, this arrangement allows us to use the data for our exploration into the collection.

Team Approach

We began our project with the intention of creating an interactive presentation that explored the meaning of modern art through the data. We created a list of static visualizations that we planned to present in a carousel display, telling a story about the MoMA's collection.

However, we stepped back from this course for two reasons. First, we did not think such a presentation would be interactive enough, and that it would not satisfy our understanding of an application. Second, we did not want to tell a story about modern art as a whole discipline.

Our focus was on the data from the MoMA, and we wanted to explore what made this collection unique.

Our assignments introduced us to `crossfilter.js`, a JavaScript library that connects multi-variate databases within a browser display.⁷ `Crossfilter.js` allows for the creation of dimensions and groups within data, giving the user the ability (through the browser) to interact with the data. However, `crossfilter.js` does not render the graphs. A second JavaScript library, `dc.js`, helps combine the graph-drawing functionality of `d3.js` with the data manipulation in `crossfilter.js`.⁸

The combination of `d3.js`, `crossfilter.js`, and `dc.js` gave us the opportunity to create an interactive application within a browser that would allow a user to manipulate and view the MoMA data in real time. Rather than tell a story with our data presentations, we would build a program that would give users the chance to explore the data on their own and form their own hypotheses.

Analysis of the Data

The datasets from MoMA are available on GitHub.⁹ The data are published in two files, one listing the Artists in the museum (15,226 records) and the other detailing the Artworks present in the museum (131,266 records). To focus our research, we decided to focus solely on the entries classified as “Paintings” in the Artworks data, which includes some of the most famous works in the museum. Cleaning the database left us with more than 2,200 records, which we believed would be a more manageable number

Our data normalization began by cleaning up the column heads, turning them into single strings. We then deleted the records with a Classification other than “Painting,” which included Films, Videos, Photographs, Prints, Drawings, and more. We also normalized the Nationality field; when multiple artists were listed as creators of a painting, we chose the nationality of the first artist credited. We applied this same logic to the Gender field, choosing the gender of the first artist for the whole record. The Date of Production field also contained multiple variations, including single dates, date ranges, “circa” dates. For the sake of consistency, we chose the latest date listed. We also rounded the measurement fields (`length_cm`, `width_cm`, `depth_cm`) to num-

bers with up to 2 decimal places.

As a group, we had some concerns about our ability to manipulate data formats and calculations within JavaScript. Therefore, we added two fields to the dataset, in which we calculated the age of the artwork at the time of acquisition (`AgeAtAcquisition`) and the area of the artwork in square meters (`Area_m2`). We also decided to build smaller groups around the artists' nationalities, creating a `Continent` field that we could use in our visualizations.

Finally, we deleted the columns from the dataset that would not be used in our visualizations.

Production

We modeled our work after a visualization available on bl.ocks.org.¹⁰ This structure utilizes four different JavaScript libraries: `d3.js`, `crossfilter.js`, `dc.js`, and `bootstrap.js`. We also used two stylesheets, `bootstrap.css` and `dc.css`.

As a group, we chose six visualizations for our application according to principles established by Kirk¹¹:

- **Count of the paintings in each year they were acquired:** The bar chart displayed the quantities of paintings over the time variable.
- **Region of the artist's nationality:** The row chart could show quantity of artwork over a smaller variable set (continents) with longer names that might skew the look of a typical bar chart.
- **Gender of the artist:** The donut chart worked well because we had only two constituent categories (male/female) that made up the whole.
- **Count of the paintings by their size:** The bar chart showed the totals over the square area of the artwork.
- **Count of the paintings by the age they were when acquired:** Again, the bar chart effectively showed the quantities of artwork over the age of the work at the acquisition date.
- Table listing the artworks selected by the filter

Each member of the group used the same dataset and basic outline of the code to construct two of the visualizations. We discussed the dimensions and groups we thought we would need for the project. Then we separated to work on the code separately, establishing the variables

to be filled by the dataset, creating the dimensions and groups that would be necessary for manipulating the data, and building the individual visualizations with the functions available in dc.js. We then came back as group to review the dimensions and groups to find commonality within the code. One group member made the code and styling more efficient, another worked on the summary report, and the third tested the application for consistency and accuracy.

Result

The application is a streamlined, single-page dashboard that allows the user to explore the data of the MoMA's collection of paintings. This tool enables users to generate and test hypotheses about the collection itself. For instance, a user could use the tool's filters to answer the following questions:

1. When did the museum begin to acquire female artists in larger numbers?
2. Who painted the largest paintings in the collection?
3. Which regions of the world are underrepresented in the collection, and has that trend begun to change?
4. What makes art modern, the year it is acquired or the age at which it is acquired?

Exploration of the collection's data is a satisfactory experience when built on the libraries of d3.js, crossfilter.js, dc.js, and bootstrap.js. However, we did not wish to ignore the reason why all of this information exists--namely, the art itself. Therefore, we added a record in the table that will allow the user to click a link and view the artwork's image directly from the MoMA's website. And if the image is not available on the site (and a hundreds of them are not), the link takes users to the MoMA's home page, where they can begin their own virtual tour of the museum.

References

1. “Museum of Modern Art | MoMA.” The Museum of Modern Art, www.moma.org/about/who-we-are/moma-history. Accessed August 21, 2017.
2. “MoMA Learning: What is Modern Art?” The Museum of Modern Art, www.moma.org/learn/moma_learning/themes/what-is-modern-art. Accessed August 21, 2017.
3. Ibid.
4. “Museum of Modern Art | MoMA.” The Museum of Modern Art, www.moma.org/about/who-we-are/moma-history. Accessed August 21, 2017.
5. Romeo, Fiona. “Thousands of Exhausted Things, or Why We Dedicated MoMA’s Collection Data to the Public Domain.” *Medium*, MoMA, 27 July 2015, medium.com/digital-moma/thousands-of-exhausted-things-or-why-we-dedicated-moma-s-collection-data-to-the-public-domain-7e0a7165e999. Accessed August 21, 2017.
6. “CC0.” Creative Commons, creativecommons.org/share-your-work/public-domain/cc0. Accessed August 24, 2017.
7. “Crossfilter.” Github, Square, 2012, square.github.io/crossfilter/. Accessed August 18, 2017.
8. “Introduction to Dc.js.” D3.js Tips and Tricks, 30 July 2013, www.d3noob.org/2013/07/introduction-to-dcjs.html. Accessed August 22, 2017.
9. Museum of Modern Art. “MuseumofModernArt/Collection.” Github, 1 Aug. 2017, github.com/MuseumofModernArt/collection. Accessed July 24, 2017.
10. “Earthquake Data Discovery Using Dc.js, Crossfilter, d3.js and Bootstrap.” Popular Blocks, bl.ocks.org/d3noob/6077996. Accessed August 16, 2017.
11. Kirk, Andy. *Data Visualisation: a Handbook for Data Driven Design*. Sage, 2016.