Re-Chart: A Platform for Redesigning and Improving Existing Bar and Line Charts

Erin Singer
Stanford University
esinger2@stanford.edu

John Newcomb
Stanford University
jnewcs@stanford.edu

Pat Briggs
Stanford University
pkbriggs@stanford.edu

ABSTRACT
Poor aesthetics and image quality plague many of today's visualizations in reports and throughout the Web. In response, we present Re-Chart: a system for extracting data from an existing chart, designing and customizing a new chart, and exporting it as a high-quality image. Re-Chart focuses on the customization of the new chart - it provides the ability to apply a preset theme, extract colors from an image and apply them to the chart, and minutely customize nearly every detail of the chart. In user tests, Re-Chart was received well - special attention was given to the image color extraction system for its novel approach in chart design.

Author Keywords
Visualization; redesign; color extraction; interaction; charts; direct manipulation.

ACM Classification Keywords
H5.2 [Information Interfaces and Presentation]: User Interfaces – Graphical User Interfaces.

INTRODUCTION
Graphics in papers, presentations, or online posts can make understanding the underlying data difficult due to low image quality, bad design principles, and poor aesthetics. Few tools currently exist to solve this problem and the easiest solution is often re-creating the chart or graphic in a familiar coding language (such as R) or application (such as Tableau). This process is difficult and tedious.

Consider the two charts in Figure 1 (right), which depict survey data on the favorite sport of kids at a fictitious elementary school. The top chart has been saved, extracted, and re-saved multiple times to exhibit the effect of the re-use of a low-resolution graphic. In addition, it contains many poor aesthetic choices - dark grid lines, a font that is difficult to read, and poor contrast. The bottom chart is a re-creation of the chart using Re-Chart. It uses one of the default themes, and it exemplifies better aesthetic principles: light grid lines, high contrast colors, and a legible font.

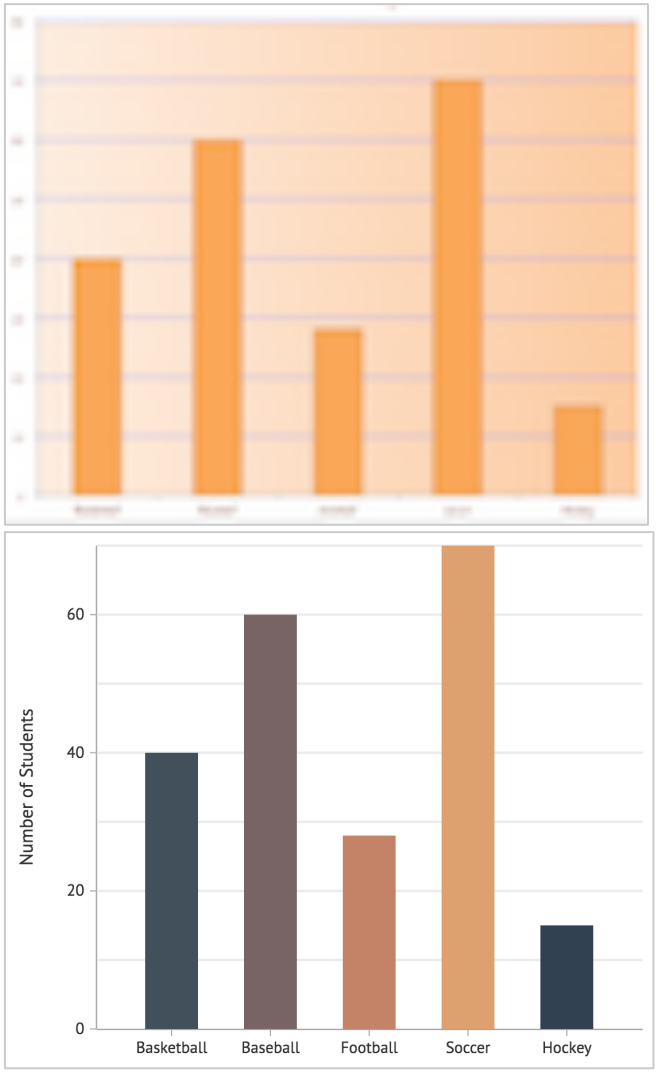

Figure 1: Chart redesign. Top: A bar chart with poor aesthetic principles at a low resolution. Bottom: The same chart re-created using one of Re-Chart's preset themes.

We present Re-Chart, a web application that allows the user to upload an existing bar chart, automatically extract its underlying data, then create a new, customizable chart using their data. The user can either select from a preset list of themes or upload an image whose primary colors are analyzed and used in the new chart. If these options are not enough, our platform also allows the user to fully customize

a countless number of attributes of the chart (color, font, grid style, etc.). Finally, the user is free to export their new chart as a high-resolution image.

RELATED WORK

The Re-Chart platform builds on three different pieces of related research and one consumer-facing application.

Restyling D3 Visualizations

In their paper "Deconstructing and Restyling D3 Visualizations" [3], Harper and Agrawala discuss the creation of a web extension to reconstruct and restyle charts. The extension is solely intended for D3-based visualizations, and works by extracting data values from the D3 code. Once this has been done, the user can restyle various parts of the chart without having to write any code themselves. This project is admirable in that it can extract data with 100% accuracy and it uses a highly accessible medium. However, these strengths come with the limitations of being restricted to D3-based charts and the fact that it does not transfer interactions or animations from the original to the revised chart.

Automated Classification, Analysis, and Redesign of Charts

In "ReVision: Automated Classification, Analysis and Redesign of Chart Images" [7], Savva et al. describe their system for extracting data from charts and creating a new, customized chart with the data. Their research boasts great performance in the classification of chart type and in the extraction of data from charts. However, it was not created for an accessible medium and does not provide an extremely high level of customization for the chart.

With these limitations in mind, ReVision serves as the main inspiration for our project. We, however, focus on different key areas. Most notably, we focus less on the data extraction portion and put more emphasis on a high level of customization and on novel interactions. We go into more detail about the key differences and our main contributions in the *Discussion* section.

Graphical Overlays

Kong and Agrawala discuss their method of adding overlays to existing charts in "Graphical Overlays: Using Layered Elements to Aid Chart Reading" [5]. The problem they address is that reading a visualization often requires the reader to extract, compare, or aggregate numerical values, and that most visualizations do not help the viewer accomplish all these tasks. Their proposed solution is a system that allows the user to add overlays to an existing chart to make it more compelling or useful.

Their solution allows for five different types of overlays: reference structures (grid lines), highlights, redundant encodings (line to illustrate trend), summary statistics (mean line), and annotations (user notes to give context). They draw their inspiration partially from Tufte's chapter on layering and separation in *Envisioning Information*. The system is limited in that it does not have an understanding of the raw data, thus it does not always position the overlays correctly. This can lead to parts of the chart being obstructed and the overlays not being as effective as intended. However, this work provides great inspiration and insight into how overlays can make reading and understanding a visualization much easier.

Web Application for Chart Creation

The web application Plot.ly [6] describes itself as the "Github for data scientists" - it allows people to create, share, and edit visualizations together. It is a consumer-facing website which offers an extensive set of tools for customizing every part of a chart and it exhibits a very polished, accessible interface. The main functional limitation is that, in order to create a chart, the user must input raw data. In addition, it is a paid tool, making it inherently less accessible to the community.

METHODS

We began our project with a simple goal in mind - being able to improve the quality of a bar chart. We had seen too many bar charts with impractical 3D effects, unreadable fonts, and clashing fonts at such low resolution we could barely read them. We started by brainstorming every part of a bar chart one could realistically customize - bar width and height, bar color, label font, label size, grid color, grid line width, etc. After coming up with an exhaustive list, we were able to group these attributes into four main categories: bar, grid, axis, and chart.

We then set out to create a basic bar chart with some of the interactive controls from the four categories we found. Once we had this working, we sketched out a few different possible layouts, considering trade-offs for the right balance of visibility, minimalism, and size. Our team then split up the next milestone: finish all the controls, implement a revised design, and add a novel interaction technique for chart customization. After some more brainstorming and speaking to our professor about possible ideas we could explore, we decided to implement a system for extracting primary colors from images and applying them to a chart.

Finally, we finished creating all of the customization controls, implementing the color extraction system, adding a preset themes list, and adding the ability to export as an image. With this functionality in place, we developed a Wizard of Oz system for the existing chart upload and data extraction systems, and tied all of the remaining pieces together. We will now go into an overview of our system as a whole.

Re-Chart is composed of a three stage pipeline: (1) uploading an existing chart and extracting the underlying data, (2) redesign and customization of a new chart, (3) exporting. We explicitly chose to include all steps of the pipeline, from start to finish, in order to emphasize that our platform is intended to be a solution to the task of redesigning a chart.

Chart Upload & Extraction

A fair amount of recent research has focused on the problem of data extraction [3, 7]. For instance, Harper and Agrawala were able to extract data from existing D3 visualizations with 100% accuracy. Also, Savva et al. were able to extract data from any image with 71% accuracy for bar charts and 64% for pie charts. Given the primary focus of our project was on the end-to-end chart revision process and a high level of customization, the specifics of the exact chart upload and extraction algorithm were not a high priority for our group. In addition, since existing systems from research achieve a high level of accuracy, we chose to not to create an upload and extraction system.

We instead chose to apply the Wizard of Oz method to the chart upload and extraction step. This allows the user to select an aesthetically poor, low quality chart from a curated list. The platform then shows the user the extracted data from the chart, giving the impression that it was extracted dynamically. The user is then free to continue with redesigning and customizing a new chart with this data.

Redesign & Customization

The core of our efforts went into the design and customization of charts in our platform. The core of our charting system is built on the popular library D3.js, which gave us a great deal of control over creating a completely customizable chart [2]. Specifically, Re-Chart provides three primary methods for customizing one's chart: (1) setting the color scheme from a source image, (2) setting a preset theme, and (3) manual customization. Through our informal research, we found that none of these methods alone adequately let the user accomplish their task of creating a bar chart that fit their needs, thus we implemented a combination of the three.

Image Color Scheme Extraction

One novel interaction we created for Re-Chart was the ability to extract the primary colors from an image and apply this color scheme to the user's chart. Figure 2 illustrates this system. This allows the user to: borrow a color scheme from another chart they like, style the chart to fit the color scheme of their website, customize their chart to match the theme in a PowerPoint presentation, or adopt the color scheme from any other image.

The color extraction from bitmap images is done using Lokesh Dhakar's Color Thief library [1], which uses a median cut algorithm to cluster similar colors together. The top N colors (where N is the number of bars or lines in the chart) are then computed by picking the most prominent colors while maximizing distance between clusters. The user is then shown the generated list of colors and allowed to select any subset of the colors for their chart.

Figure 2: The image color extraction system. After the user uploads an image, the primary colors are extracted and the user can select which colors to use in their chart.

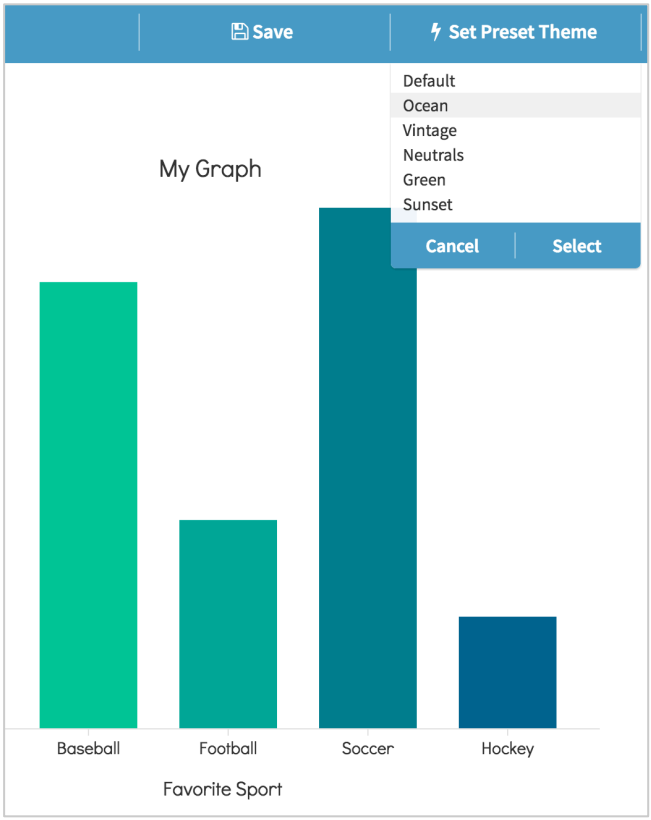

Figure 3: Preset theme list. The user can select any premade theme and further customize it or use it as-is.

Preset Themes

Preset themes are not new in the world of graphics, but they play an important role in providing aesthetic guidance to the user. Since not all people using charts are well versed in design principles, presets provide a low-risk and minimal effort way of creating an aesthetically pleasing chart. We created all of our preset themes using Re-Chart itself, and attempted to create a large variety in order to maximize the chance that the user is able to quickly find a theme that fits their needs. See Figure 3.

Manual Customization

While preset themes and extracted color schemes from images may give users a solid aesthetic foundation, they also inherently restrict customizability and restrict power-users or designers from accomplishing their tasks. Thus, we invested significant effort into making every appropriate piece of the chart customizable - from fonts and size to the color and width of grid lines. These customization tools follow the principles of direct manipulation in order to provide immediate feedback to the user and to make values easily editable. Figure 4 shows some of these controls.

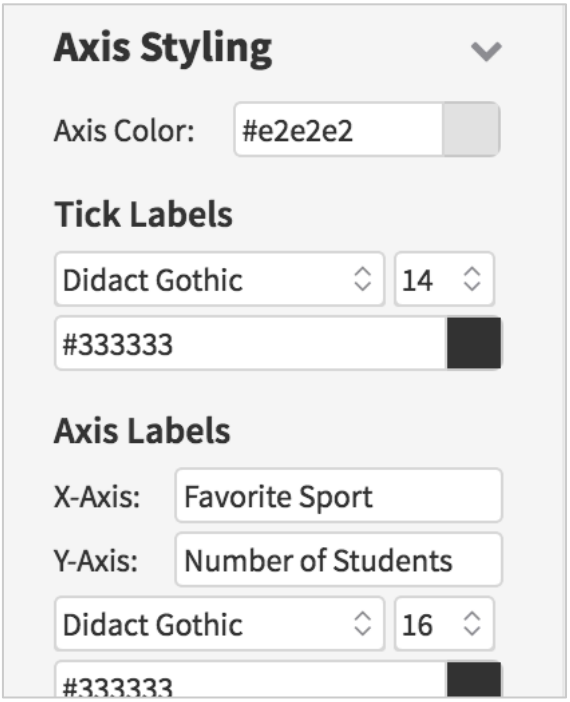

Figure 4: Customization controls. Using these controls, the user can modify nearly every detail of their chart.

Exporting

The final step that a user performs when they are done customizing their chart is to export it. When the user clicks on the "Save" button, a snapshot is taken of their current chart and it is saved to their computer as a high resolution image. While we originally had plans to also support exporting interactive charts (i.e. as embeddable code), we chose to focus our efforts on the redesign and customization step of the process. This is discussed in greater detail in our *Future Work* section.

SCENARIOS

Re-Chart is useful for the process of taking an existing bar or line chart, extracting its data, then creating a new, customized chart from the data. We demonstrate its capabilities by considering the following scenarios.

Scenario #1

A researcher is studying the famous barley growing experiment run in Minnesota in the 1930s [4]. The researcher is working on building upon and analyzing this data set, and wishes to use a chart from the original paper in her own work to provide context to her readers. Unfortunately, the graphic is a small, low resolution, black-and-white chart that makes the data difficult to read. She considers recreating this chart herself using the original published data, but the data from the experiment contains 120 observations, thus doing so would be a slow, tedious process. She instead decides to use Re-Chart: she uploads the original chart, chooses a nice preset theme, tweaks a few of the colors and values, and within a few minutes she has a high quality chart to use in her research paper.

Scenario #2

A writer for the New York Times is writing a story on the increased frequency of 3-point shots in basketball over the past century. She is able to find some historical line charts that depict the averages of a collection of teams over a few seasons. However, all of the illustrations have different styles and also do not include data from the most recent seasons. Unfortunately, her editor won't allow her to publish all of the historical charts, as they do not come close to remotely match the design of their site. She decides to use Re-Chart to give all of the old charts a new, standardized design and create new charts to fill in the gaps. To ensure the charts match the New York Times site, she takes a screenshot of the homepage and uploads it to the color extractor, and chooses a few of the nicest colors to include in her charts. After a few minutes of tweaking and showing her editor her updated work, she is ready to publish her story.

RESULTS

After developing our platform, we showed the system to a variety of people and received a generous amount of feedback. The feedback was overall very positive and it contained many suggestions for future work.

The component that received the most amount of feedback was our color scheme extractor. People expressed great interest in the ability to make their graphs match their websites and presentations. While using the tool, many people played with it and used it to extract colors from slides and websites (as expected), but a large number also used the tool to extract colors from images that they found and liked. We found this usage interesting and it confirmed that the color extraction system indeed provides utility, and that there may be many other use cases that we simply have not thought of.

Preset themes also garnered a large amount of attention in our tests. While some appreciated the ability to customize nearly every piece of their chart, many preferred the ability to select a theme as a starting point and to customize one or two pieces afterwards. Most people we spoke to did not claim to have a design background, and we believe this led to them not being extremely particular about font or color choices and instead being mostly satisfied by the meticulously crafted preset themes.

One user testing our system appeared to be particularly detail oriented and customized countless parts of their chart. They repeatedly asked if they could customize particular pieces and, when told yes, were interested in giving it a unique look. While they did not use Re-Chart like the majority of users, this user definitely appreciated the wide variety of customization tools.

In addition to giving us feedback on parts of the platform we showed them, many people expressed interest in additional functionality or features. A few people suggested the possibility of also being able to enter a website URL instead of an image to extract colors. Other people also expressed interest in the ability to extract fonts from the images they uploaded. We agreed with both of these suggestions and think they would be great enhancements to the tool. A few people also inquired about adding support for more types of graphs, such as pie charts and scatter plots, and we think this would be an important issue to tackle for future work.

DISCUSSION

In this section, we focus on two points of discussion. First, we discuss how our work compares to that of ReVision and Ploy.ly [7, 6], and second, the importance of color adapting UI.

Work Differentiation

With this project, our goal was to do two things. First, we wanted to provide a platform for the end-to-end experience of uploading an existing chart, extracting its data, and creating a new, customized chart with this data. Second, we wanted to add a novel interaction to the chart customization process - the color adapting UI. Given the timeframe of this project was limited to a few weeks rather than months, we acknowledged early on that it was simply infeasible to implement every single part of our vision.

We still, however, wanted to showcase a working prototype of the experience we imagined. Thus, we focused our efforts on the areas where we wanted to make contributions and spent less time on the other areas. As a result, we chose to 'Wizard of Oz' the existing chart uploader and data extracting process by providing presets. Thus, our work is different from ReVision in that we do not perform any chart classification or automatic data extraction, and that we have a much larger focus on the customization of the new chart.

Re-Chart differs from Plot.ly in two significant ways. First, we provide the complete experience of chart recreation (uploading and extraction, customization, then exporting). While the first of these steps is not fully implemented, it results in satisfying a different set of needs and providing a different user experience. Second, we add the novel interaction of color-adapting charts, which is discussed further in the next subsection. Thus, while somewhat similar in interface and tools provided, Re-Chart has a different focus than Plot.ly.

Color Adapting UI

One of the largest pieces of differentiating work in Re-Chart is the ability to create color-adapting charts. This interaction is unique in that it not only provides a solution to the task of redesigning charts, but also considers the motivation of the task. One primary motivation for re-creating charts is to match a chart to an existing site or presentation's design. Since this is currently done by manually creating a new chart and guessing at the best way to match a design, we envisioned a better way - color extraction directly from the source to be mirrored.

We believe this simple interaction could in fact lead to a fundamental shift in how we think of charts: no longer are they floating, mismatching additions to content, they are perfectly matching complements to stories, slides, and other content sources. Thus, we find this interaction to be one of the major highlights of our project.

FUTURE WORK

In Re-Chart's current implementation, we focused on the last two stages of the pipeline discussed above: redesigning/customizing a new chart and exporting it. We believe that there are several directions our project can go from this starting point.

Implement Chart Upload & Data Extraction

The logical next step for our project would be to fully implement the chart upload and data extraction feature that forms the first step in our pipeline. In order to be a true consumer facing application, Re-Chart needs to allow users to import any bar/line chart, customize that chart, and export it for easy reuse. Given that the ReVision system [7] already does this step, we would be adapting a lot of that codebase and fitting it to our system.

Add More Visualization Types

Currently, Re-Chart only works with bar and line charts. While these types of visualizations are incredibly popular, they represent only a small subset of the entire visualization space. Expanding Re-Chart's focus to maps and networks could provide value to a lot of consumers.

Expand Export Options

We envision many different types of people using Re-Chart: researchers, students, and coders among many others. The ability to export high-quality images certainly fits a lot of use cases, but certainly not all of them. Allowing users to export usable D3 code of their visualization could be hugely valuable and allow users to plug the code into their own sites, while retaining the interactive features that aren't possible in an image (like hover interactions or dynamic changing of the underlying data).

Conduct Additional User Research

For the current implementation of Re-Chart, we did not have a lot of time to focus on user research and feedback. To make our product better, we need an even better idea of how people use our application in the real world and what they would want to see in next versions.

Add Capabilities to Color Extraction System

During our user tests, we received great feedback regarding our image color extraction system. While users appreciated the current system, they expressed desire for additional functionality. Many of them requested the ability to insert the URL of a website or a PDF in order to directly extract colors from a webpage. Others recommended the ability to also extract fonts from images. We believe implementing both of these suggestions would provide a large benefit to potential users and help streamline the tool even further.

REFERENCES

1. Color Thief. https://github.com/lokesh/color-thief/. Retrieved June 2016.
2. D3. https://d3js.org/. Retrieved May 2016.
3. J. Harper and M. Agrawala. Deconstructing and Restyling D3 Visualizations. In UIST 2014 (October 2014), pages 253-262.
4. F. R. Immer, H. K. Hayes, and Le Roy Powers. Statistical Determination of Barley Varietal Adaption. Journal of the American Society of Agronomy, 26: 403-419, 1934.
5. N. Kong and M. Agrawala. Graphical Overlays: Using Layered Elements to Aid Chart Reading. In InfoVis 2012 (October 2012), pages 2631-2638.
6. Plotly. https://plot.ly/. Retrieved May 2016.
7. M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. ReVision: Automated Classification, Analysis and Redesign of Chart Images. In UIST 2011, pages 393-402.