

Grad student life: Cost of Living @UBC

Jingxuan Huang, Devyani McLaren, Tommy Nguyen

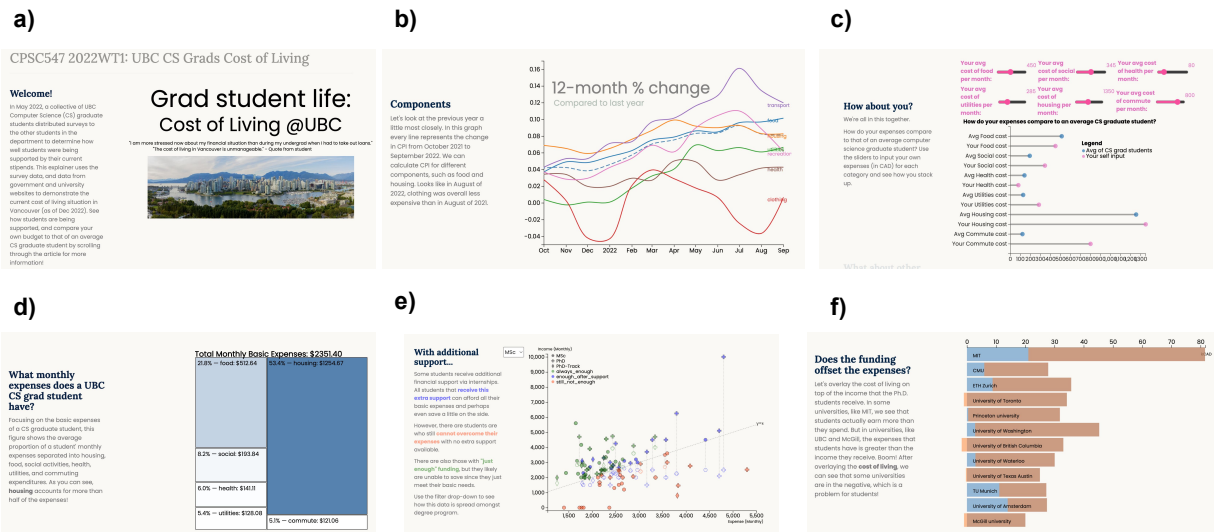


Figure 1: Images of the main visualizations used in the article. a) The top of the article, b) a multi-line graph for visualization consumer price index, c) an interactive visualization for comparing user expense input to the average expense of a CS grad student, d) a treemap showing the breakdown of expenses for a grad student, e) a scatterplot showing income vs expense for grad students, f) a barplot comparing the funding of different universities

Abstract—This paper describes an article that utilizes animated and interactive visualizations to present data about UBC computer science (CS) graduate (grad) students’ financial circumstances and the problems that these grad students face in the expensive city of Vancouver, British Columbia. The visualizations focus on 5 main tasks involving readers learning about the expenses of CS grad students, the inflation rate in Vancouver, how readers’ expenses compare to CS grad students and how the financial compensation at UBC compares to other universities. The article uses easy-to-read visualizations and applies a simple layout and format to make it easier to understand the data represented.

Index Terms—Explainer visualization, Scrollytelling, Interactive visualization

1 INTRODUCTION

As of 2022, inflation in Vancouver, British Columbia (BC), Canada, is at an overall high of 8.8% compared to 1.2% in 2021 [9]. The average rent for a 1-bedroom apartment is CA \$2,500 [7]. University students in BC are facing costly financial decisions to pay for basic expenses alongside tuition [14].

In January 2022, a small collective of Computer Science (CS) Graduate (grad) students at the University of British Columbia (UBC) came together to compile a cost of living report based on a survey they had sent to fellow CS graduate students to determine how students have

- Jingxuan is with the DMM Lab, UBC Dept of Computer Science.
- Devyani McLaren is with the SPIN Lab, UBC Dept of Computer Science.
- Tommy Nguyen is with the SPIN Lab, UBC Dept of Computer Science.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

been affected by the rising cost of living in Vancouver [19]. The results were an anonymized rich dataset that provided quantitative and qualitative insight into these students’ finances and life contexts. In May 2022, these students released a cost of living report [19] to the UBC CS department with preliminary analysis and basic data visualizations. However, careful thought is still required to further understand the data represented in the survey. Our aim is to provide more robust and interactive visualization solutions beyond the simple charts and graphs outlined in the report.

Explainer-type articles, sometimes referred to as “explainers” or “interactive explainer”, are created to promote active reading and a greater understanding of data presented in an article [15, 22]. These types of articles contain written contextual information but also allow readers to understand what is written with interactive visualized examples. Further, some explainer implement **scrollytelling** [18] —as the reader scrolls through the article, animations and visualizations help the reader walk through the article and understand its contents [17]. Scrollytelling is a powerful method to ensure visualizations are understood. However, careful considerations must be made to implement

scrollytelling, such as allowing readers to scroll back and forth without error [13].

By utilizing scrollytelling, our aim in this project is to

1. characterize the expenses and income of CS grad students using data from the cost of living survey
2. understand how inflation has affected the cost of living in Vancouver
3. discover the difference in income/expenses between the different CS grad programs
4. allow users to compare their current financial situation to UBC CS grad students
5. compare UBC’s financial support to that of other universities to understand if the problem that students face at UBC is specific to UBC or prevalent at other institutions

To meet these goals, our contribution is an explainer article that utilizes scrollytelling to walk readers through complex data with animated, interactive and comparative visualizations, using data from the cost of living survey, British Columbia (BC) inflation rates and university websites.

POSTIONALITY: Our team consists of CS graduate students sharing experiences in research, development, and design. Although we are not experts in this topic, we are actively affected by it. Given the rising prices of commodities in Vancouver and the current state of our graduate student stipends, we are constantly making decisions in our life that stem from our finances. Our team takes a lot of interest in this topic and is passionate about delivering a strong set of visualizations of graduates’ cost of living to help inform current students, prospective students, and administrators to make informed decisions for the betterment of CS grads at UBC. We are not immune to bias and acknowledge that this is a sensitive and ever-evolving situation. We hope this project respectfully provides important insights to students and the UBC administration team.

2 RELATED WORK

To the best of the authors’ knowledge, no specific visualization-focused work has been done on student finances. However, there are many examples of explainer articles and previous literature on the benefits of understanding student costs and expenses.

2.1 Explainers

“A visual introduction to machine learning” [23] is one example of a scrollytelling explainer article that allows readers to understand complex machine learning information (see Figure 2). The article has minimal text and numerous visualizations that are animated. Figure 2 provides an example of how scrollytelling is effectively showcased in this article. McKenna et al. conducted a series of studies and proved that visualizations and animated transitions have positive effect on reader-perceived engagement [17].

2.2 Current approaches to understanding financial situation

Abdullah et al. [11] investigated creating a new visualization for the Malaysian government in 2017. They found that a treemap diagram was useful as an interactive form of a budget since popup boxes are practical for displaying overlaying information, and the amount of a budget is easily encoded in the area of boxes in the diagram. This supports the findings of the US President Barack Obama administration’s attempt to create an easy-to-understand fiscal budget [8], see Figure 5, which also utilized a treemap diagram. The use of treemaps in previous work provides the foundation for supporting our first aim—characterizing the expenses and income of CS grad students.

The Cost of Living Estimator, created by Raj Tiwari [20], is a comparator-type interactive visualization that takes user-inputted income/expenses data and shows users how their income compares to

others living in large cities in the United States. This interface uses colour hue identity channels, length, and position on common scale-type channels. It is unclear why a bar graph and scatterplot idioms were chosen to represent the data. However, even without explanation, the estimator is easy to understand and demonstrates why creating simple visualizations that use effective channels is necessary for showcasing financial data. Figure 4 shows the visualization used in this interface. The estimator provides insights for us to accomplish the third aim.

The UBC Undergraduate programs and admissions page contains a “first-year cost calculator,” where prospective undergraduate students can input basic information such as “amount of support from parents” or “savings at the start of the term” [10]. Based on this input, the calculator provides a report of the costs associated with living as an undergraduate student at UBC. Notably, this calculator does not say what information it is based on, and it is unclear why this calculator is only for first-year undergraduate students when the majority of UBC is made up of other types of students. Therefore, it also cannot fulfill the aims we listed earlier.

3 ABSTRACTION

3.1 Data Abstraction

Data in this project comes from 3 main sources: 1) the cost of living survey, 2) the BC provincial government website, and 3) university websites that provide information about the funding of graduate students and articles from the news or governments that provide information about the cost of living in a city.

3.1.1 The Cost of Living Survey

The cost of living survey consisted of questions about student income and expenses [19]. An anonymized version of the survey data was released in June 2022 [6]. In total, there were 94 responses from a mix of Ph.D., MSc and Ph.D.-track students. The dataset is tabular and it has 139 items and 26 attributes. Most of the attributes are ordered, quantitative, and sequential. These attributes are where students listed their income and expenses for different aspects of their life, such as food, housing, and travel. The other attributes consist of categorical data, such as “are they the primary care provider?” The items of this dataset are individual students that responded to the survey. Table 1 contains the data abstraction for this data source. We only included the attributes that will be used in our visualizations.

Table 1: Data Abstraction for the Cost of Living Survey

Cost of Living Survey Data Abstraction		
Data Abstraction	Domain Specific (Monthly)	Range/Values
Ordered Quantitative Sequential	Food Expense	\$50-\$1500
Ordered Quantitative Sequential	Social Expense	\$0-\$600
Ordered Quantitative Sequential	Health Expense	\$0-\$800
Ordered Quantitative Sequential	Utilities Expense	\$0-\$300
Ordered Quantitative Sequential	Rent/Mortgage Expense	\$700-\$3170
Ordered Quantitative Sequential	Commute Expense	\$0-\$800
Ordered Quantitative Sequential	Research Assistant Income	\$0-\$2826
Ordered Quantitative Sequential	Teaching Assistant Income	\$0-\$2800
Ordered Quantitative Sequential	Awards	\$0-\$3333
Ordered Quantitative Sequential	Other Sources of Income	\$0-\$7500
Categorical	Student Type	M.Sc, Ph.D.-track, Ph.D.

3.1.2 Consumer Price Index (CPI) in British Columbia

To visualize the growth of the inflation rate, we collected the monthly Consumer Price Index (CPI) in British Columbia from Statistics Canada [5]. The dataset contains data from October 2015 to September 2022 and can be divided into seven consecutive years. The “All-Item” CPI is the overall indicator and it consists of 8 different components. We filtered out the “Alcoholic beverages, tobacco products and recreational cannabis” so that each remaining component could find a corresponding type of expense in our survey. Table 2 shows the data abstraction of this dataset.

3.1.3 University comparison

The same group of CS graduate students that conducted the survey also created an additional dataset to compare grad student funding



Figure 2: An example of scrollytelling from the “A visual introduction to machine learning” article. a) the reader first sees a scatterplot with text, b) as they scroll, an animation of green and blue areas being to appear c) once they finish scrolling, the animation completes and the reader sees the full green and blue area. Images reproduced without the author’s permission

Table 2: Data abstraction for the CPI dataset.

CPI Data Abstraction		
Data Abstraction	Domain Specific	Range/Values
Ordered Quantitative Sequential	Time	[15-Oct, 22-Sep]
	All-Items	[120.4, 147.8]
	Food	[134.1, 168.4]
	Shelter	[114.4, 144.1]
	Household operations, furnishings and equipment	[113.7, 129.9]
	Clothing and footwear	[101.4, 112.5]
	Transportation (including gasoline)	[125, 169.4]
	Health and personal care	[113.9, 134.4]
	Recreation, education and reading	[114.6, 142.2]

Table 3: Data abstraction for university comparison

University Comparison Data Abstraction		
Data Abstraction	Domain Specific	Range/Values
Categorical	University	names of 12 universities
Ordered Quantitative Sequential	Yearly_funding_kCAD	[20.00, 81.52]
Ordered Quantitative Sequential	Yearly_col_kCAD	[13.43, 60.49]

and cost of living at different universities. Currently, data is listed for 12 universities and only compares Ph.D. level funding from those universities. The dataset includes data including or excluding tuition; we will use the ones with tuition included. All currencies are converted into Canadian dollars (CAD) according to the average exchange rates in the last year grounded to 2 floating points, where 1 USD is 1.32 CAD, 1 CHF is 1.36 CAD, and 1 euro is 1.34 CAD. Table 3 shows the data abstraction for this dataset.

3.2 Task Abstraction

The main goal of our visualization is to provide a holistic overview of the cost of living situation for CS grad students. Since our final result will be accessible to many people, an approach that can explain data to people with different backgrounds is necessary.

3.2.1 Task 1: Identify the expenses of a UBC CS graduate student

This task allows readers to first become familiar with the type of data we will be using from the cost of living survey. Readers are expected to identify what the largest expenses are for a CS grad student and then retain this information in the back of their minds as they scroll through the rest of the article.

3.2.2 Task 2: Understand inflation rate by analyzing the consumer price index (CPI)

If we only present data about student finances, it would be difficult for readers to put such data into context. Therefore we need readers to understand the cost of living in Vancouver so that when presented with data about UBC students, they can properly contextualize the information.

3.2.3 Task 3: Discover the difference in income and expenses between masters (M.Sc.), Ph.D.-Track, and Ph.D. students

Masters and Ph.D. students receive different levels of funding. Therefore, each category of students is affected differently when living in Vancouver. University websites often only provide generalized information about student funding therefore we, want readers to obtain a more detailed understanding of graduate students by analyzing them based on the degree program they are in.

3.2.4 Task 4: Compare your expenses to that of a CS graduate student

Simply reading and viewing information is not the most effective way of making sure readers understand our message. We want readers to have a personalized experience and therefore want them to provide their own data (should they wish) and compare themselves to the graduate students they are reading about.

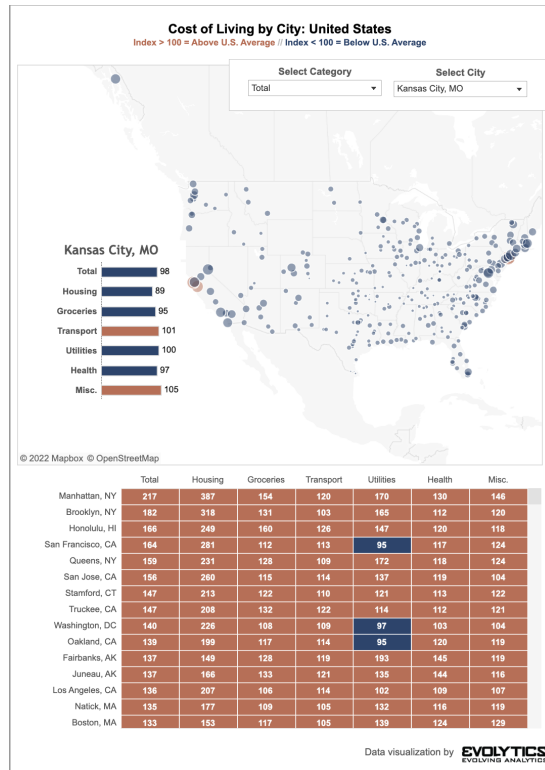


Figure 3: Example of cost of living data spatial and tabular data.



Figure 4: Idioms used in a cost of living comparator interface.

3.2.5 Task 5: Compare funding vs. cost of living of similarly ranked universities

Completing this task provides more context into the financial problems UBC CS grad students are facing. By characterizing the funding versus cost of living at similarly ranked institutions, we can highlight issues at UBC and allow readers to also understand these.

Based on these tasks our primary users are intended to be current UBC CS graduate students. Secondary users includes both prospective graduate students and current UBC and CS department administration. Our visualization communicates important financial information that is relevant to all of these users.

4 SOLUTION

The final artifact of our work will is a scrollytelling article titled "Grad student life: Cost of Living @UBC." The article and our code can be accessed on GitHub ¹.

4.1 Average Monthly Basic Expenses Treemap Diagram

We utilize the treemap diagram (Figure 6) to allow users to quickly see how individual UBC CS Grad students spend their money monthly at first glance.

¹<https://github.com/UBCCSGradCoLExplainer/scrollable-explainer>

4.1.1 What

This visualization focuses on displaying the average monthly basic expenses of the surveyed UBC CS graduate students. We present the mean data from the original dataset [6] for the respective basic expenses. This provides 6 quantitative attributes for the basic expense category: food, social, health, utilities, housing, and commute. The treemap works to provide a hierarchical topological visualization of the average monthly basic expenses of the surveyed UBC CS graduate students. Point marks are used for the dollar amount of each of the basic expenses. The proportion of the expenditure relative to the total average monthly basic expenses is doubly encoded with the area and saturation channels. The housing expense, the largest area, represents 53.4% of the entire average monthly basic expenses with the most saturated steelblue colour. Smaller percentages are presented with less saturation—closer to white. The position further encodes the relative proportion, with the largest appearing more positive on the y and then the x coordinates.

4.1.2 Why

The treemap visualization is able to directly support tasks 1 and 4. Appearing at the start of the scrollytelling explainer, the visual immediately contextualizes the expenditure of an average UBC CS graduate student. This visualization best provides an intuitive demonstration of the expenses using the area. Without reading the text, readers can immediately identify the extremities—the largest and smallest areas of

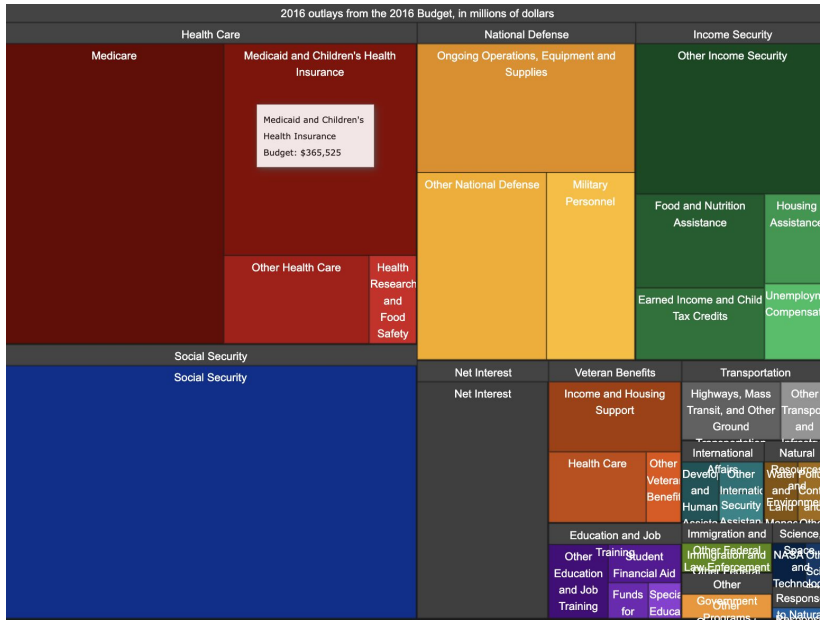


Figure 5: Treemap visualization used by the United States President Obama Administration to showcase their fiscal budget

the average basic expenses; in this visualization, only the most saturated housing expense stands out.

4.1.3 How

The aim is to allow readers to quickly process how the average basic expenses are broken down and in what proportion at first glance. Reading more closely, the text contextualizes the actual dollar value to allow users to compare the expenses while also providing insight into how much current students spend as well as readers' potential future budget if they are prospective students. The colours help bring attention to the extremities with their contrast between the saturation.

4.2 CPI Line Charts

We used a set of line charts to represent the growing trend of the Customer Price Index (CPI). These charts are connected with a series of transitions of 5 phases (Figure 7). Note that we omit one phase in Figure 7 because it is similar to phase (d) but has a different line highlighted.

4.2.1 What

The original dataset only contains monthly CPI. To highlight the gradient, we derive a new attribute: 12-month percentage change. This is the most commonly reported CPI figure [1] and is especially useful to compare the degree of changes in different periods.

The charts originally use the point mark to encode each monthly CPI (Figure 7(a)) or 12-month percentage change (Figure 7(b-d)). In order to show the trend, we also use line marks to connect the points and smooth the lines.

The major channels are horizontal and vertical positions, represented as x and y scales. We encode time using horizontal positions, is accurate to the month. Figure 7(a) and (b) use a range of the last 7 years while (c) and (d) have a range of the last year (Oct 2021 to Sep 2022). We also use the colour hue channel to encode 7 different components of the overall CPI, as shown in Figure 7(c) and (d).

4.2.2 Why

The CPI line charts visualize the inflation rate of British Columbia, which directly fulfills our task 2. Specifically, by visualizing the 12-month percentage change, we highlight the rapid growth of CPI over the past few months. The breakdown of the all-item CPI also gives the readers a chance to browse different components and compare their trends (Figure 7(c)). They can further identify the components that

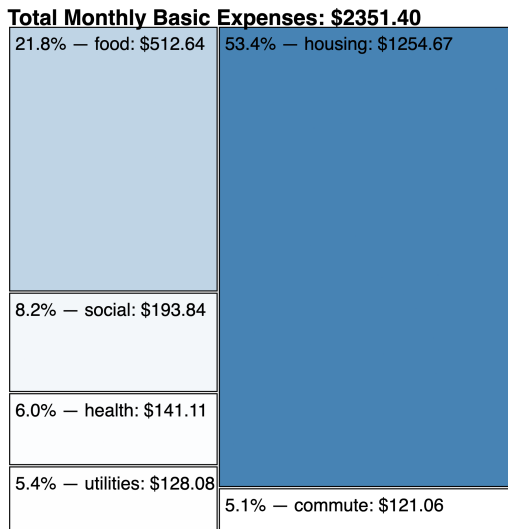


Figure 6: Treemap Diagram of Total Average Monthly Basic Expenses. This chart provides a visual hierarchical overview of the basic expenses where the respective expenses' area represents the proportion of the total average basic expenses.

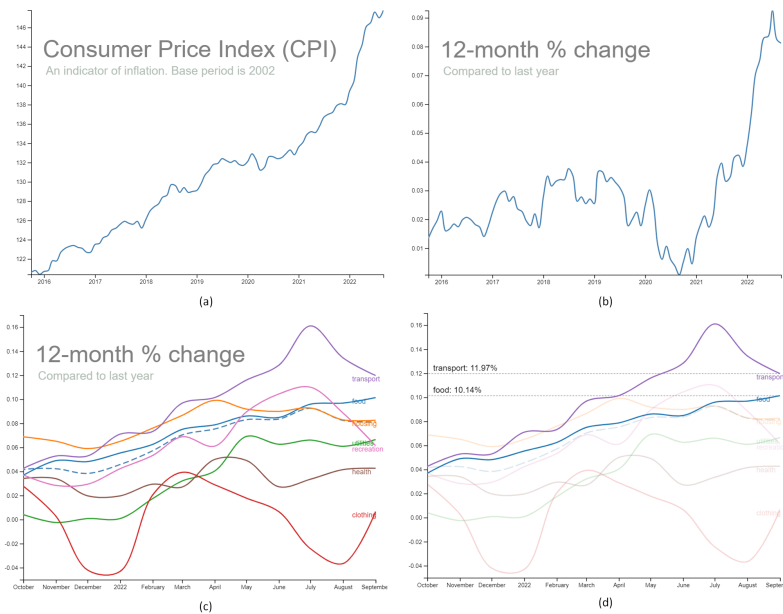


Figure 7: Line charts with different phases show the trend of CPI in British Columbia. (a) The trend of CPI over the last 7 years. (b) The trend of the 12-month percentage change of CPI over the last 7 years. (c) The trend of the 12-month percentage change of CPI in the previous year is divided into 7 components. (d) Some components are highlighted to emphasize the points.

have the most severe impact on graduate students through highlighting (Figure 7(d)).

There are many alternatives to the line chart, but they cannot resolve the task we described above in the context of a scrollable explainer. The first stream of alternatives is for the single line chart (Figure 7(a) and (b)). We first consider using the "you-draw-it" visualization, where we would invite the readers to draw the trend before we presented the real line. This idiom is first proposed in *New York Times* [12] and later used to show the trend of CO2 emission [15]. It is very effective for "teaching" a topic compared to passive reading and presents the CPI trend. However, this idiom does not fit into the context of scrollable explainers, as it may disrupt the linear narration by requiring users to switch from scrolling to clicking and drawing a line before seeing the result [13]. Another alternative is the stacked area chart or the stream graph, where 7 lines representing each year are layered together. This graph will be very helpful in identifying the difference between any two consecutive years, but not a good choice to present the overall trend.

The second stream of alternatives is for the multiple lines chart (Figure 7(c) and (d)). A common choice is also the stacked area chart or the stream graph. However, CPI is an index, so the sum of the CPI of different components is not meaningful.

4.2.3 How

We used animated transitions to connect all 5 phases of the line chart. It will help readers to understand the relationship between different phases. For example, when transforming from Figure 7(b) to (c), we used zooming-in to let users know that (c) comes from a part of (b). We also added transitions to both axes so that users will note that they have changed.

We constrained the x-axis of Figure 7(b) and derived a new view (7(c)) so that we can focus on the change in the last 12 months without overwhelming readers with too much information. In addition, we superimposed the all-item CPI layer and multi-component CPI layer in Figure 7(c) and (d) but turned down the opacity of the former one. By doing so, readers can browse the additional lines without losing the information from previous phases. Finally, we used highlighting to allow specific components pop out using the opacity channel (Figure 7(d)). This will help our readers identify the components that have the largest impact on graduate students.

4.3 Income vs Expense Scatterplot

We used a scatterplot with 3 phases to present the income and basic expenses of the respondents in the survey.

4.3.1 What

To reduce the complexity and show the overall distribution, we aggregate different types of income and expenses and derive three sequential attributes:

- **Basic income:** RA, TA, Awards.
- **Basic expenses:** Food, Social, Health, Utilities, Housing, Commute.
- **Extra support:** Other sources, Other support.

Based on these three attributes, we further derive a new categorical attribute, "financial status," that divides students into three groups:

- **Always enough.** The student can afford basic expenses with a basic income.
- **Enough after support.** The student cannot afford basic expenses with a basic income but is able to live after receiving extra financial support.
- **Still not enough.** The student cannot afford basic expenses even after receiving extra financial support.

These new attributes transform the original dataset into a more compact version that is ready for use in the scatterplot.

We mainly use the point marks to indicate students. In Figure 8(a) and (b), each point indicates one student, but in 8(c), every pair of filled and unfilled points connected with a dotted line represents one student. Some students are still represented as one filled point because they do not have extra financial support. We also used line marks to assist readers with comprehension. One is the $y = x$ auxiliary line, which separates the students who can or cannot afford basic expenses using funding. Another is the dotted lines that connect the same students. We added them to help readers track the change after a student has received extra financial support.

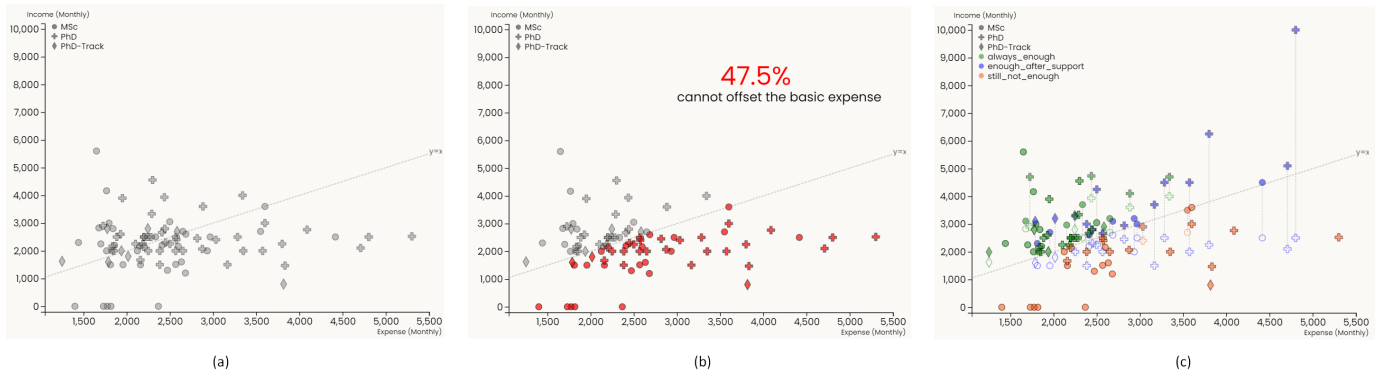


Figure 8: A scatterplot with three phases presents the income and expenses of respondents in the survey. (a) Grey points show the basic expense and funding of students. The shape encodes the program that a student is in. The auxiliary line separates those who can afford the basic expenses with funding. (b) Students who cannot afford the basic expenses with funding are highlighted in red. (c) Additional points show the position of students after receiving extra financial support. A pair of filled and unfilled points connected with a dotted line denotes the same student. Colour hue encodes the financial status of a student.

We use horizontal and vertical positions to encode monthly basic expenses and monthly income separately. Here, income denotes basic income (i.e., funding) in Figure 8(a) and (b), and still denotes basic income for unfilled points but the sum of basic and additional income for filled points in 8(c). Besides, we encode the program that a student is in with the shape channel. We also used the colour hue to indicate a student's financial status after receiving extra support. The chosen colours have some semantic meaning: green means safe, blue means almost safe, and red means unsafe. The filled and unfilled points encode the income before and after extra support.

4.3.2 Why

This visualization aligns with task 3 and shows the financial situation of current graduate students within different programs at the UBC CS department. Specifically, it presents the distribution and correlation between students' income and basic expenses. Readers can browse the plot and understand the financial struggle of current students. They can also compare the status of students from different programs. We also annotated the financial status of students based on their positions before and after support. This grouping will help readers identify students who urgently need extra funding.

When deciding on channels to encode categorical attributes, we chose the colour hue for financial status and the shape for the programs. The rationale is that we also provided a filter to highlight a specific program as we will describe below. Therefore, we used colour hue, the most effective categorical channel, to encode financial status, and used a less effective channel to encode programs.

4.3.3 How

Similar to the other visualizations in our explainer, we use animated transitions to connect different phases of the scatterplot. Typically, in order to help readers understand that a pair of points denotes the same student in Figure 8(c), we add the animation of drawing dotted lines and later showing the filled dots at the top of lines. We also superimpose two layers in Figure 8(c), where the first layer contains unfilled points and the second layer contains filled points. It can help readers compare the position of a student before and after support.

Considering the complexity in 8(c), we further added a drop-down list in our explainer to filter items according to the program. The content of the explainer will also change (Figure 9).

4.4 Safe Ball

We also design a simple visualization to show the percentage of respondents that feel safe or unsafe about current stipend amounts.

4.4.1 What

The plot uses point marks to represent different groups of students. Only two channels are used in this visualization. The area channel encodes the sequential attribute: the percentage of students in that group. The colour channel encodes the categorical attribute: if a student feels safe or unsafe about the current stipend.

4.4.2 Why

This visualization shows current graduate students' stress due to their financial status. Though it does not directly correspond to any tasks we listed above, we created it mainly to make the storyline of the explainer more consistent. Since it will appear after the scatterplot, it proves that the struggle is real and students are truly worried about it. And it will be followed by the lollipop chart, so it is a good way to evoke empathy before we invite readers to compare their own situations with other graduate students.

4.4.3 How

This visualization is relatively simple. Two points are juxtaposed for easy comparison. Readers can immediately identify that the percentage of students that feel unsafe is much larger than those who feel safe about current stipend.

4.5 Compare & Contrast: Your budget vs CS students

This visualization is presented to readers after the Income vs Expense scatterplot so that readers are first provided background information about grad student expenses and income and then interact with a subset of that information in this compare and contrast idiom. The visualization utilizes sliders to provide interactivity.

4.5.1 What

The data in this lollipop chart is from the cost of living survey, and it uses 6 of the quantitative attributes corresponding to the 6 monthly expense categories for CS grad students. Readers interact with the visualization by providing their own expenses for the same categories by using the sliders to change the corresponding attribute in the graph.

A lollipop is comprised of a line and a point mark. Where the point mark is placed on the end of the line to indicate the quantitative value that the lollipop represents.

Accordingly, the fixed round lippolops (represented in purple) are the averages (mean) of all 94 respondents.

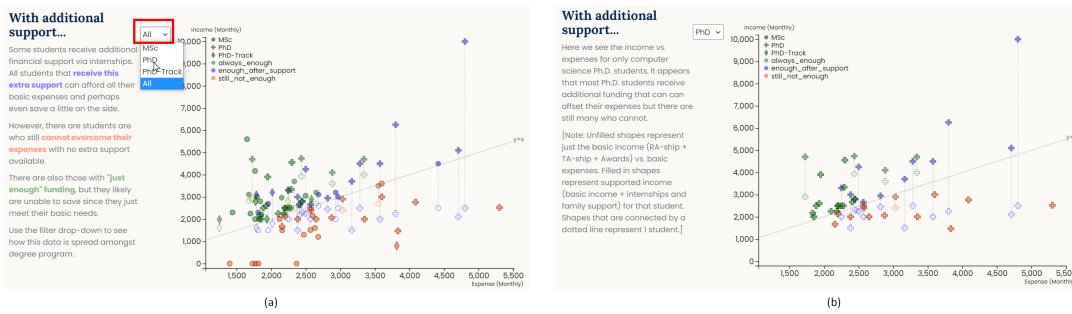


Figure 9: A drop-down list that filters items based on the program. (a) The scatterplot before filtering. (b) The scatterplot after the reader chooses "Ph.D." Only points corresponding to the Ph.D. remain in the plot. The content on the left also changes accordingly.

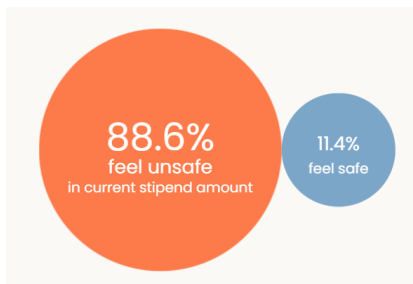


Figure 10: The "safe ball" visualization. The red ball represents the students that feel unsafe in the current stipend amount, while the blue ball represents those who feel safe.

4.5.2 Why

To support task 4 - compare your expenses to that of a CS grad student - we need access to the reader's personal expense information and a way to visually show the reader the expenses of a CS grad student, so the reader can perform a comparison task.

We, therefore, use interactive sliders for readers to input their own expenses. Readers can click and drag on the point mark on the slider to their desired position. As the slider moves, the text to the right of the slider changes to tell them the exact value of the slider, and the corresponding lollipop mark on the graph also changes. We chose this approach so readers can easily *identify* their represented data on the idiom in real-time. Further, we chose this form of interactivity so that we do not limit readers to only inputting their exact expenses. Instead, users can input different values within the range of the slider so they can see if their own monthly budget needs to change.

We chose to *present* the data in this idiom with lollipop marks because these marks use the easy-to-understand position on a common scale channel and therefore reduce the cognitive load to perform the *comparison*.

4.5.3 How

The colour hue channel *encodes* the type of data represented. Where purple is an average expense of a CS grad and pink is the reader's self-inputted value.

Sliders are used for readers to *manipulate* the idiom, where changing the position of the slider changes the position of the corresponding lollipop. The area of the colour that is filled in the slider *encodes* the quantitative value that the slider represents.

The expense categories are grouped on the y-axis so that the average expense of the grad student is presented above the reader's inputted value for that expense. This way, readers can simply note the difference in the position of 2 the lollipop marks and compare.

4.6 Overlapped Bar Chart

We used a series of bar charts to compare UBC with similar-ranked universities in terms of funding that Ph.D. students receive and the cost of living in the city where the university is located in.

4.6.1 What

The bar charts mainly use line marks to indicate each university's funding or cost of living. We used the length of line marks to encode two sequential attributes: funding and cost of living. Since the cost of living is counted as a negative value while the funding is a positive value, the bars will grow in different directions. The vertical positions encode the categorical attribute "university". We also utilized the colour hue channel to differentiate funding and cost of living.

4.6.2 Why

This visualization aligns with our task 5, where we aimed to compare UBC with similar-ranked universities regarding students' financial status.

Our design of the overlapped bar chart is quite innovative (Figure 12 (b, c)). It is based on the diverging bar chart. However, the diverging bar chart cannot show how much is remained after offsetting the cost of living with funding. An intuitive way is to draw a grouped bar chart where each university has two juxtaposed bars: the funding bar and the cost-of-living bar. This design can show the offset, but it will be hard for the readers to compare the remaining money because those remaining parts are not aligned. We finally designed the current version of the overlapped bar chart, where the cost-of-living bar will start growing from the funding bar's end so that each university's remaining parts will be aligned.

However, the initial feedback from readers shows that it is hard to understand the overlapped bar chart at first glance because its design is uncommon. To address the issue, we first showed the diverging bar chart in our explainer (Figure 12 (a)) and then transitioned to this overlapped bar chart (Figure 12 (b)), so that the readers can understand why bars can be overlapped in this way and what is the meaning of the remaining parts.

4.6.3 How

We used animation transitions to connect each bar chart. In Figure 12(a) and (c), we used flashing animation to highlight UBC. It is also a way to reduce the complexity of the visualization so that readers will not be overwhelmed by the universities they need to compare. We superimposed the funding bars and the cost-of-living bars in Figure 12(b) and (c) to see how much funding is left or how much expenses cannot be offset. We also changed the order of the bar in Figure 12(c) to ease the comparison.

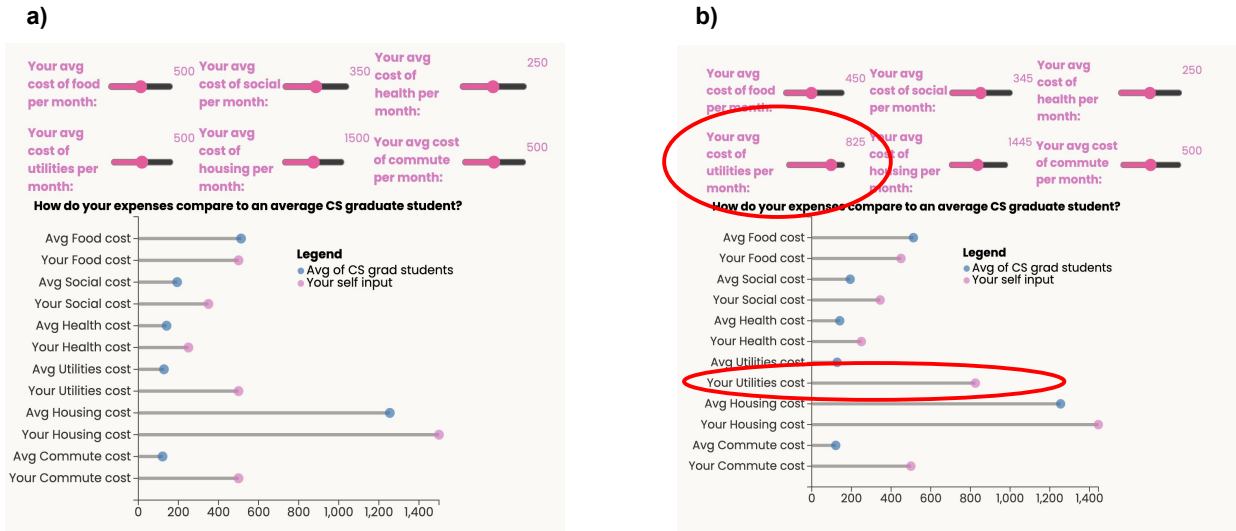


Figure 11: The interactive lollipop chart used in the article. The chart displays the average expense of a CS student (purple) with the interactive self-input (pink). By changing the slider, the user can see the real-time change of the lollipop bars (pink marks only). The difference between (a) and (b) depicts if a reader changes their utilities cost from \$500 (a) to \$825 (b). Note the change in the length of the corresponding lollipop bar. The lollipop chart is built using D3.js and the sliders are built using the observable plot library.

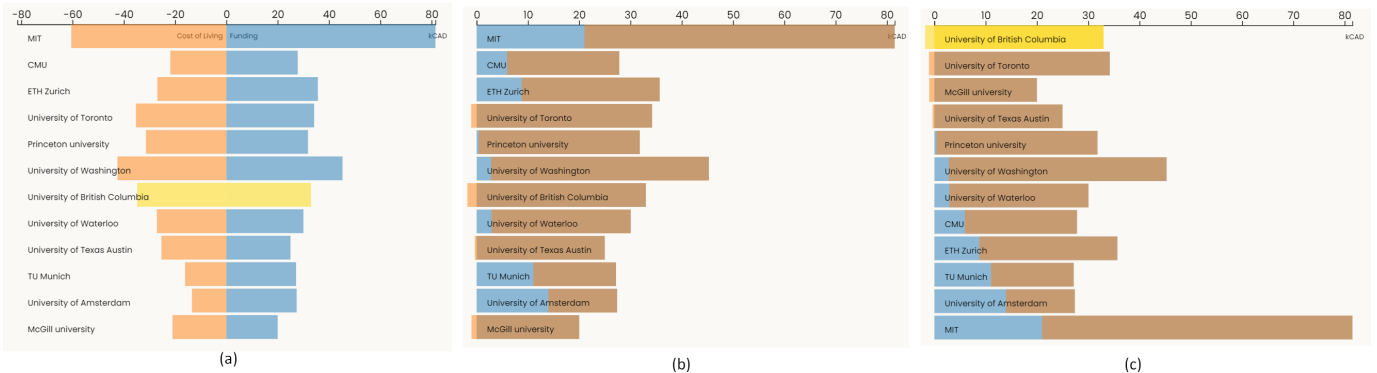


Figure 12: The bar charts depict how the funding for CS Ph.D. students at different universities compares to the cost of living in the city that the university is located in. (a) The diverging bar chart. Red bars represent the cost of living, and blue bars represent funding. Each row stands for a university. The flashing animation highlights the bars of UBC. (b) Bars in the same row are overlapped to show if students earn more than they spend. (c) The overlapped bars are reordered based on how much money is left. The flashing animation is used again to highlight UBC.

Welcome!

In May 2022, a collective of UBC Computer Science (CS) graduate students distributed surveys to the other students in the department to determine how well students were being supported by their current stipends. This explainer uses the survey data, and data from government and university websites to demonstrate the current cost of living situation in Vancouver (as of Dec 2022). See how students are being supported, and compare your own budget to that of an average CS graduate student by scrolling through the article for more information!

Grad student life: Cost of Living @UBC

"I am more stressed now about my financial situation than during my undergrad when I had to take out loans."
"The cost of living in Vancouver is unmanageable." - Quote from student



Figure 13: The welcome page. The widget within the red rectangle is moving up and down to prompt readers to scroll down and read more.

4.7 Scroller

4.7.1 Design Rationale

Finally, we built a scroll-based interactive article to include all the visualizations we designed above. The scroll-based article, or "scrollytelling", is one of the most popular ways of interactive storytelling [13]. It engages readers with interactive components while making the interaction almost effortless.

A well-known alternative to scrollers is steppers, where readers click through instead of scrolling through sections. The advocates of steppers point out the issue of "scrolljacking", where the scroller is not well designed and thus sacrifices the user experience [16]. However, we argued that this issue can be avoided by following certain design patterns, such as the five rules proposed by Mike Bostock [13]. In addition, the practitioners from the New York Times reported that readers are much less willing to interact with steppers [21]. The click-based interface tends to overload the reader with choices [13] and potentially hinder them from the interaction. Therefore, we still chose to implement the scroller and designed it in a careful way.

4.7.2 Design Details

We adopted many approaches to making sure that the five design rules are observed. On the welcome page, we used an animated widget to prompt readers to scroll down in case they don't know they can scroll (Figure). This is especially a concern because in some devices, scrolling is not the top choice of interaction [13].

In addition, readers can scroll back and forth and scroll at any speed. The transitions are designed to be able to playback so the visualization will automatically change to the phase where readers stop. Most transitions between screens are time-based with an instantaneous trigger. Only the safe ball visualization entrance is the position-based transition, so there will be a transitional region between screens. By doing so we ensured that this relatively small plot only fully shows when the content on the left is also on the current viewport. To avoid unwanted disruptions, our animated transitions are all very short and will finish within a second. Readers will not feel interrupted when reading the content because, at that time, the animation is ended. They can always scroll back and forth to retrigger the animation if they want to watch it again. Besides, we always provide instantaneous and consistent feedback when readers scroll: The content will be scrolled down just like a normal article. The position of the plot is fixed, but what is shown will change when readers scroll up and down.

Aside from the five basic rules, we also designed some parts to improve user experience. For example, we added a section to introduce the formula of CPI so that the readers can better understand the meaning of CPI line charts. The colour schema across most of the plots is consistent and colorblind-friendly. Finally, we also inserted a clickable union logo with a link to the card sign-up page to advocate for real changes.

5 IMPLEMENTATION

To execute our solution, we initially looked at several tools. However, given the limited time, we scaled down to ensure the project gets completed within our allotted time. Minimizing the amount of new complicated tools to learn, we decided to primarily focus our attention on D3. D3 is known to have a steep learning curve. However, our team has chosen to move ahead with it since it demands us to understand our data on a deeper level to visualize our data appropriately and adequately. D3's complexity comes from having to design the world of one's data; designers must have high attention to detail and make clear choices concerning the marks, channels, axis, scales, and labels they will utilize. This means our team had to do our due diligence to study the data and make meaningful informed decisions to visualize our data best properly when using D3.

The scrollytelling is implemented as an interactive scrollable website using HTML, CSS, vanilla JavaScript with D3 version 4 (v4), and Observable. The entire implementation was conducted in a CodeSandbox, a web-accessible live collaborative IDE. All codes, assets, and notes were pushed to a public GitHub repository.

We scoped out what tools are out there for the various levels of expertise when it comes to designing your own explainer and interactive scroller. Exploreables [2] is a hub for learning through interactivity. Their tools page pointed us towards two highlighted resources to build our own explainer: Idyll [3] and Observable [4]. Observable is significantly more popular than Idyll, with over 2,000 weekly downloads compared to Idyll's 85 weekly downloads (data retrieved from npmjs package pages). This gave us confidence that there will be more support and resources for Observable. When it comes to interacting with D3, Observable also has a comprehensive, robust set of tutorials that interconnects these two, especially since Observable was created by the same developer as D3. Instead of using Observable to deliver our explainer, we utilized it to facilitate some chart making to be then embedded afterwards.

To further aid the creation of this explainer, the project began as a fork of an exemplary scroller repository². We preserved the core logic of the scroller with a complete overhaul of the visualizations and stylesheets. Additionally, we added the functionality of reading and storing multiple datasets to the original scroller. Most of the visualizations were implemented using D3 v4 with the exception of the lollipop chart. The lollipop chart was created using native Observable and later embedded into the website. See Table 4 for the split of work.

Table 4: Split of work.

Work Description	Jingxuan	Devyani	Tommy	
Expenses Treemap	20%	0%	80%	
CPI Line Charts	100%	0%	0%	
Income vs Expense Scatterplot	100%	0%	0%	
Safe Ball	100%	0%	0%	
Compare & Contrast	0%	100%	0%	
Overlapped Bar Chart	100%	0%	0%	
Scroller	implementation	100%	0%	
	content	0%	100%	0%
	polish	0%	0%	100%

Initially, our team had to overcome the hurdle of learning new tools and techniques. At first, we wanted to use Vega-Lite for its simplicity and easier learning curve; however, we realized the grammar of Vega-Lite was too high-level—limiting its expressiveness and cannot meet our requirements. Therefore, we made the switch to D3 but the learning curve proved to be very steep, especially because most of our team members had no front-end development skills. After several weeks of research and practice, we began to make tangible progress with our tools to create our scrollytelling explainer.

In our first version of the scroller, we focused our attention on the visualizations while we continued to refine our understanding of D3. After consulting with the instructor, we determined some issues with the scroller, such as readers being unable to scroll back and a mismatch

²https://github.com/vlandham/scroll_demo

between the content in the viewport and the start of the animation. To address the issues, we decided to search for and build on a better exemplary scroller to build on rather than from scratch; this was our best course of action to present our project in a timely manner. When implementing the visualizations, we added the playback transitions to make sure readers could scroll back and re-interact and engage with the visualizations—deepening their learning. We also tweaked the timing of transitions to ensure a smooth reading experience.

Another large hurdle to tackle was the integration of the codes. Since the treemap and lollipop charts were implemented via Observable in contrast to everything else via JavaScript within the CodeSandbox IDE, there were some incompatibilities when it came to embedding these Observable charts. Although Observable supports embeddable exports via iframe or dynamic JavaScript, its environment was still different fairly different—preventing us from adequately controlling some elements and transitions of the embedded charts. We were able to successfully embed the lollipop chart, however, the treemap chart had to be reimplemented and simplified as D3 JavaScript within CodeSandbox.

The final outcome presents a combination of Observable and D3 working to deliver interactive visualizations on top of an interactive webpage powered by the logic of vlandham’s repository³.

6 RESULTS (SCENARIO OF USE)

Here we present 4 use cases for our readers. We present these cases as questions readers may have when they encounter this article.

6.1 Use case 1: What is a CS grad student’s largest expense?

Jenny is completing her undergraduate degree at the University of Saskatchewan and is thinking of applying to UBC for a master’s in computer science. She has heard that it is expensive to live in Vancouver and is curious to see how current CS grad students at UBC spend their money.

While looking at the first visualization presented in the article - the tree map diagram Figure 6, Jenny sees that the largest rectangle corresponds to housing. Further, she reads that housing makes up 53.4% of an average CS student’s budget.

6.2 Use case 2: Can my fellow M.Sc. students afford to live in Vancouver?

Wren is in their second year of their master’s degree at UBC and has been struggling to afford food in Vancouver while also paying their rent each month. Wren is curious if this is also the case for their fellow master’s students. Wren scrolls through the article and comes across the scatterplot section. They notice that the scatterplot is being explained using animations until they reach the last section “with additional support,” Figure 14, which shows the entire scatterplot. The scatterplot is fairly complex so they click on the drop-down menu and click “MSc” only to see the masters students in the plot. Now that some of the complexity of the plot has been removed, Wren notices that many of the M.Sc. students represented on the plot are highlighted in red and, like themselves, may also be in debt because they cannot offset their expenses.

6.3 Use case 3: As a CS grad student, do I pay too much rent?

Inspired by the scatterplot, Wren scrolls through the article and comes across the lollipop chart Figure 11. Noticing that the average monthly expenses are represented on this chart. Wren inputs their monthly rent, \$1350 using the pink slider. They see that their pink lollipop for housing expenses is longer than an average CS student.

6.4 Use case 4: Do other universities have a similar stipend problem as UBC?

Dr. Laks Lakshmanan is the associate head of grad affairs and a professor of the CS department. He has been working with CS grad students to better understand their financial needs and problems and

³https://github.com/vlandham/scroll_demo

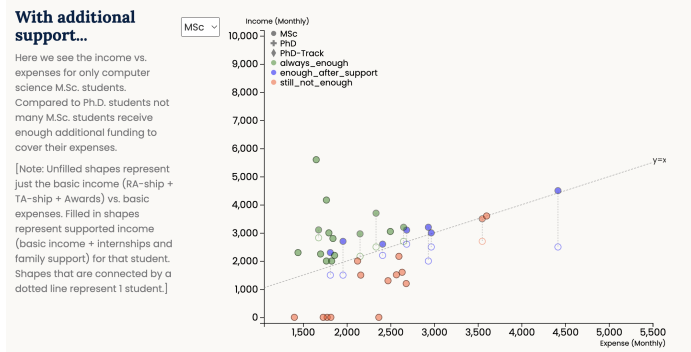


Figure 14: The welcome page. The widget within the red rectangle is moving up and down to prompt readers to scroll down and read more.

is curious whether other students and other universities face the same issues as UBC students. He scrolls through the article and comes to the beginning of the last section “What about other universities?,” in Figure 12. Laks first sees the data laid out in a diverging bar chart and where UBC is represented on this chart when it is highlighted in yellow. He continues to scroll and the animation shows him how the expenses are then overlaid on top of the funding. He immediately notices that some universities have an orange section that sticks out since the expenses are more than the funding provided. However, it’s difficult to compare these universities because they are in seemingly random order. He scrolls again, and the universities are then ordered for him so he can see which universities have the greatest deficit. He identifies that UBC has the largest deficit, although it is still fairly close to that of the University of Toronto and McGill. He finds this an interesting observation but wonders where this data came from. He scrolls to the end of the article and clicks on the GitHub link. In the readme, he finds a link to a google spreadsheet of the university comparison data along with links to university websites and news articles detailing the funding of Ph.D. students at each university and the approximate cost of living in each city.

7 DISCUSSION, FUTURE WORK AND CONCLUSIONS

7.1 Strengths, weaknesses, limitations

Strengths. The scrollytelling explainer makes data more accessible to the average reader. One can easily put themselves in the position of the surveyed CS graduate students and follow along the narrative with greater empathy before calling our readers to action. The scroller offers interactive visualizations to help readers better process the data and engage with the data for more insightful comparisons; this overall strengthens our readers understanding of the situation. The design of the scroller further adheres to the five basic rules as prescribed by Bostock and thus heightens the user experience. The scroller additionally has very low latency which is attributed to us drawing each plot in advance upon loading and then keeping them hidden until readers scroll to respective sections before calling its activation function for viewing.

Weaknesses. Although we tried our best to ensure the visualizations will be in a valid state regardless of the scrolling speed of the reader, the logic of the scroller is still not robust enough to thoroughly accommodate. Additionally, the colour schema of the lollipop chart is not consistent with the other charts because the lollipop chart’s Observable embedding changes colour out of our control. Some of the visualizations may suffer performance issues as our dataset grows—particularly important if we want to support readers contributing their own data. Also, the current version of the scatterplot visualization would not be very useful if we have hundreds or thousands of students.

- data...

Limitations. Given the limited time frame, our team unfortunately was not able to conduct a thorough evaluation of our results via a user

study. Despite that, a set of informal interviews with some current CS graduate students was able to garner some initial impressions and feedback. We already gained some constructive suggestions and made improvements, such as positioning the diverging bar chart before the overlapped bar chart. However, a more carefully designed user study is still necessary to evaluate the effectiveness of our visualizations and explainer as a whole.

We were also limited to a dataset that we did not collect. We found some interesting struggles with the data. Namely, more detail about the attributes would have been helpful. For example, one column in the dataset is labelled “commute,” however, many students may commute by car, bus or simply walk (and therefore not spend money on commuting). Thus, it’s unclear how this “commute” attribute is broken down amongst the survey respondents. For our purpose, we must assume that it includes all types of commuters but is not a precise attribute. When further considering the cost of living data beyond the ones provided for UBC CS graduates, misalignment may skew the accuracy of the comparisons.

7.2 Lessons learned

The most valuable lesson we learned is the importance of design prior to implementation. Initially, our team had worked on separate visualization with no consistent storyline in mind because we were not confident with D3 and related technologies. This incurred many problems: First, we drafted several visualizations that did not serve our narrative and could not fit into the explainer. Second, it was difficult to appropriately divide the work when little is known about the end goals. Third, the integration of code was very complicated and time-consuming as we were all working on separate environments.

We also learned that before serious implementations, we need to first prototype a demo to see if our idioms fit the data we have. For instance, initially, we tried to use the “you-draw-it” idiom for the CPI data to provide our readers with that fine-tuned interaction and compare their own data. However, after plotting it out with the regular line chart, we realized the gradient did not change dramatically, and the “you-draw-it” idiom might not be able to provide as much insightful information as we would have thought. Therefore, we switched back to the normal line chart and highlighted the gradient with the 12-month percentage change instead.

Given our endeavour and newfound confidence in our tools, we believe we can be much more tactical with a comprehensive design plan to properly alleviate these hurdles and execute our deliverables.

7.3 Future work

In the future, we plan to strengthen the robustness of the scroller to allow users to freely navigate the webpage with little to no concerns about loading issues or other hiccups. A floating table of content to allow users to quickly jump to notable sections should be implemented to further aid readers when they are reinforcing their learnings while also identifying their location in the explainer. We also hope to collect more data from the UBC CS graduate students and potentially more dimensions of the data. Future work may also consider soliciting data from readers to streamline data collection and provide readers with better insight into other prospective, current, and graduated students. Of course, work has to be done to properly vet this kind of information so that it is not abused.

There are also a lot of interesting results within the survey that we did not have time to visualize, such as students’ savings and travelling expenditures (independent of regular commutes). We plan to include them in the following version of our explainer. After we resolve the limitations mentioned above, we will release the explainer to our target readers so that it will call for real action. We wish to continue iterating on the idioms presented in this update, as well as incorporate new idioms based on the feedback we receive from course staff, peer review, and user studies.

8 CONCLUSION

In this project, we present the “Grad student life: Cost of Living UBC” explainer article that uses scrollytelling to animate visualizations and

present complex data to readers. We primarily use data from the cost of living survey sent out to UBC computer science graduate students in January 2022. Our visualizations highlight the expenses of these CS grad students, whether or not the guaranteed provided support is enough to offset their costs, the impact of high inflation on the cost of living in Vancouver, and how the expenses of readers compare to those of an average grad student and at which institutions, including UBC. To ensure that this article is accessible and easy to read, we focused on using area, position on a scale, and hue as our primary channels to encode information. To make sure complex data was presented in digestible formats we animated our visualizations to walk readers through understanding the data and provide interactions, like filter options, to help process complex data. We hope that this project can be used by current CS graduate students, prospective students, and the UBC administration to better understand the financial situation of our UBC computer science graduate students.

REFERENCES

- [1] Consumer price index: Frequently asked questions. https://www.statcan.gc.ca/en/subjects-start/prices_and_price_indexes/consumer_price_indexes/faq.
- [2] Explorables. <https://explorabl.es/>.
- [3] Idyll. <https://idyll-lang.org/docs>.
- [4] Observable. <https://observablehq.com/explore>.
- [5] Table 18-10-0004-01 consumer price index, monthly, not seasonally adjusted. <https://www150.statcan.gc.ca/t1/tb11/en/tv.action?pid=1810000401&pickMembers>.
- [6] Ubc cs graduate cost of living survey data. https://docs.google.com/spreadsheets/d/1_CkDCwjZ09Bbg0TVTyuggbz1INQOuQA2k-ygF1A32K0/edit#gid=32073719.
- [7] Vancouver, bc rent prices. <https://www.zumper.com/rent-research/vancouver-bc>.
- [8] Interactive budget - office of management and budget. <https://obamawhitehouse.archives.gov/interactive-budget>, 2016.
- [9] Annual inflation index for dcls, cac targets, and density bonus contributions. <https://vancouver.ca/home-property-development/annual-inflation-index.aspx>, 2022.
- [10] How to plan your first-year costs, 2022. <https://you.ubc.ca/financial-planning/cost/>.
- [11] N. A. S. Abdullah, N. Zulkeply, and Z. Idrus. Malaysian Budget Visualization Using Circle Packing. In M. W. Berry, B. W. Yap, A. Mohamed, and M. Köppen, eds., *Soft Computing in Data Science*, pp. 81–90. Springer Singapore, Singapore, 2019.
- [12] G. Aisch, A. Cox, and K. Quealy. You draw it: How family income predicts children’s college chances. 2015.
- [13] M. Bostock. How to scroll. <https://bost.ocks.org/mike/scroll/>, Nov. 2014.
- [14] L. Culbert. B.c.’s university, college students scrounging to pay for it all. <https://vancouver.sun.com/news/local-news/how-university-college-students-are-scrounging-to-pay-for-food-gas-2022>.
- [15] F. Hohman, M. Conlen, J. Heer, and D. H. P. Chau. Communicating with interactive articles. *Distill*, 2020. <https://distill.pub/2020/communicating-with-interactive-articles>. doi: 10.23915/distill.00028
- [16] R. Kosara. The scrollytelling scourge. <https://eagereyes.org/blog/2016/the-scrollytelling-scurge>.
- [17] S. McKenna, N. Henry Riche, B. Lee, J. Boy, and M. Meyer. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. In *Computer Graphics Forum*, vol. 36, pp. 377–387. Wiley Online Library, 2017.
- [18] D. Seyser and M. Zeiller. Scrollytelling – an analysis of visual storytelling in online journalism. In *2018 22nd International Conference Information Visualisation (IV)*, pp. 401–406, 2018. doi: 10.1109/IV.2018.00075
- [19] C. U. G. Students. Ubc cs graduate students cost of living survey. https://drive.google.com/file/d/1rGkpVYaoZpvXK061cJxG0zZjoqbXz7_E/view?usp=sharings, 2022.

- [20] R. Tiwari. Data visualization on cost of living.
<https://nycdatascience.com/blog/student-works/data-visualization-on-cost-of-living/>, 2018.
- [21] A. Tse. Why we are doing fewer interactives.
<https://github.com/archietse/malofiej-2016/blob/master/tse-malofiej-2016-slides.pdf>.
- [22] B. Victor. Explorable explanations.
<http://worrydream.com/ExplorableExplanations/>, 2011.
- [23] S. Yee and T. Chu. A visual introduction to machine learning.

9 APPENDICES

See our milestone in Table 5.

Table 5: The milestone and overall hours for each group member for this project.

Milestone	Person	Task Breakdown	Estimated Hours	Actual Hours	Estimated Date of Completion	Actual Date of Completion
Pitch	Everyone	Cost of Living Video Pitch	3	3	Sep 28	Sep 28
Pre-proposal meeting	Everyone	Group weekly meeting	3	3	Oct 11	Oct 11
	Everyone	Meet with Tamara	0.5	0.5	Oct 12	Oct 12
Proposal	Everyone	Group weekly meeting (made work plan, discussed over final results)	4	4	Oct 18	Oct 18
	TN	Report Writing	5	5	Oct 21	Oct 21
	DM		5	5	Oct 21	Oct 21
	JH		6	8	Oct 21	Oct 21
Initial Implementation	TN	Learn selected tools (D3, observable, Vega-Lite)	4	8	Nov 1	Nov 10
	DM		4	8	Nov 1	Nov 10
	JH		5	7	Nov 1	Nov 7
	Everyone	Group weekly meeting (made work plan, reviewed each other's design)	4	4	Nov 14	Nov 14
	Everyone	Data exploration and data cleaning	2	2	Oct 30	Oct 30
	Everyone	First draft sketches of 3 visualization (1 per team member)	1	1	Nov 6	Nov 5
	TN	First draft implementation of the 3 visualizations	4	4	Nov 10	Nov 10
	DM		6	7	Nov 10	Nov 10
	JH		4	4	Nov 10	Nov 10
	Update Report	Everyone	Write first draft of report	6	6	Nov 13
Everyone		Review and submit	2	2	Nov 14	Nov 14
Peer Project Review	Everyone	Prepare demo and discuss in class	4	4	Nov 15	Nov 15
	Everyone	Group weekly meeting (consolidate feedback from peer review, made work plan, reviewed each other's work)	6	6	Dec 13	Dec 13
Refine Implementation	TN	Refine initial visualizations based on the feedback	1	4	Nov 19	Nov 25
	DM		2	5	Nov 19	Nov 25
	JH		2	2	Nov 19	Nov 18
	JH	Learn D3 and js	20	25	Nov 28	Dec 1
	JH	Implement additional visualization	15	18	Dec 3	Dec 9
	JH	Implement the scroller	8	10	Dec 4	Dec 5
	JH & DM	Ask for feedback from the instructor	0.5	0.5	Dec 7	Dec 7
	JH	Refine the scroller or switch to the stepper	8	6	Dec 9	Dec 9
	DM	Write the content of the explainer	2	4	Dec 12	Dec 13
	TN	Polish the explainer	2	6	Dec 13	Dec 13
Final Presentation	DM	Prepare slides	2	3	Dec 12	Dec 13
	TN		1	2	Dec 12	Dec 13
	DM	Prepare scripts	2	3	Dec 12	Dec 13
	DM	record the video and demo	2	3	Dec 13	Dec 13
	JH & DM	present in class	0.25	0.25	Dec 14	Dec 14
Final Report	TN	First draft of report	10	10	Dec 14	Dec 16
	DM		10	12	Dec 14	Dec 16
	JH		10	10	Dec 14	Dec 16
	Everyone	Review, proofread and submit	3	3	Dec 16	Dec 16
Total	TN		65.5			
	DM		92			
	JH		129.25			