Aunt Lily Can Say Her Visualizations: Directing Analysis, Design, and Storytelling in Natural Language (A Vision)

Zening Qu*
zqu@uw.edu
University of Washington

Fan Du
fdu@adobe.com
Adobe Research

Ryan A. Rossi
ryrossi@adobe.com
Adobe Research

Bill Howe
billhowe@uw.edu
University of Washington

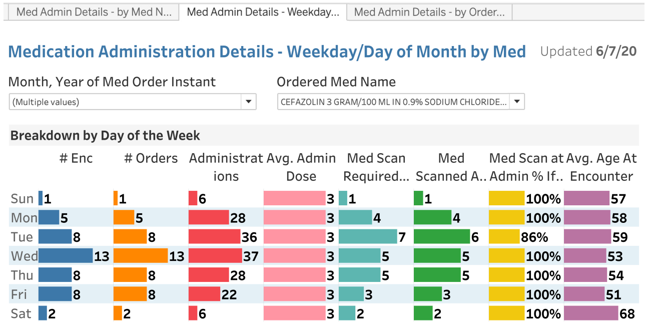
Abstract

We envision Lily, a flexible, lightweight, expressive, mixed-initiative authoring tool for anyone to direct data stories end-to-end – from insight-finding to storytelling – by saying words. We assume our vocabulary of visualizations is represented as an open repository of templates shared by authors. Our approach associates user queries with templates using learned embeddings, derives a set of variable bindings, and ranks the bindings by similarity with the user's query. By supporting data binding and template customization via natural language, Lily inherits the expressiveness of the union of popular authoring tools like D3 and Data Illustrator, while relying more on the scale and quality of NLP models. We discuss how Lily can accelerate high- and low-level authoring workflows, facilitate coarse- and fine-grain customization, and accommodate different design styles and expression habits. We intend Lily to provide a natural means of data storytelling, for novices and experts alike.

Index Terms: Human-centered computing—Visualization—Visualization systems and tools

1 Introduction

"Could you help me learn Tableau?" requested Aunt Lily on a Sunday afternoon. She is a medication safety officer working with patient data. After attempting to create a dashboard similar to the ones left off by her former colleague (Fig. 1), and after much tutorial-cramming, she lamented, *"The Tableau videos are helpful. However, it takes time and experience..."*


Med Admin Details - by Med N...
Med Admin Details - Weekday...
Med Admin Details - by Order...
Month, Year of Med Order Instant
(Multiple values)
Ordered Med Name
CEFAZOLIN 3 GRAM/100 ML IN 0.9% SODIUM CHLORIDE...

Medication Administration Details - Weekday/Day of Month by Med Updated **6/7/20**

Breakdown by Day of the Week

	# Enc	# Orders	Administrations	Avg. Admin Dose	Med Scan Required...	Med Scanned A..	Med Scan at Admin % If..	Avg. Age At Encounter
Sun	1	1	6	3	1	1	100%	57
Mon	5	5	28	3	4	4	100%	58
Tue	8	8	36	3	7	6	86%	59
Wed	13	13	37	3	5	5	100%	53
Thu	8	8	28	3	5	5	100%	54
Fri	8	8	22	3	3	3	100%	51
Sat	2	2	6	3	2	2	100%	68

Figure 1: An interactive dashboard like this requires not only design skills, but also technical know-how: how to build small multiples, how to add drop-down filters, how to show axis titles at the top of bar charts instead of at the bottom (the default formatting), how to publish dashboards as separate tabs to Tableau Server, and so on. Who knows all these execution steps by heart?

Menus and icons – our concrete, constrained, and oftentimes complicated way of communication with our computers – are powerful but necessarily become more complex as tasks become more sophisticated. Into countless clicks and drags we translate our intents, no matter what they are. The more control we want, the more widgets we add. A plethora of tools surround visualization authors to help them analyze data and tell stories – data cleaning [17, 18], analysis [42], insight finding [9], visualization design [22, 34, 36], interaction design [56], animation [43], and narration [35, 41] – each offering its own expressiveness, constraints, and user interface. Much learning is needed.

Theoretically, Aunt Lily's lack of design, analysis, and storytelling experience as well as technical know-hows could be compensated by natural language processing (NLP) and mixed-initiative recommendations. NLP could parse Aunt Lily's intents, allowing her to direct where the project is going from where she is. And the unknown or unsaid design choices could be filled in or enumerated by machine recommendations. Compound NLP and recommendation in a mixed-initiative system, the author can gain more control by saying more, or give more control to the system by saying less.

Several systems have endeavored to use natural language for visualization authoring, either as the main channel of creation, or as a complementary to menus and icons [12, 30, 38, 40, 54]. However, the authors of these systems report it is not easy to create a good NLP-driven authoring experience. Why is the problem so hard?

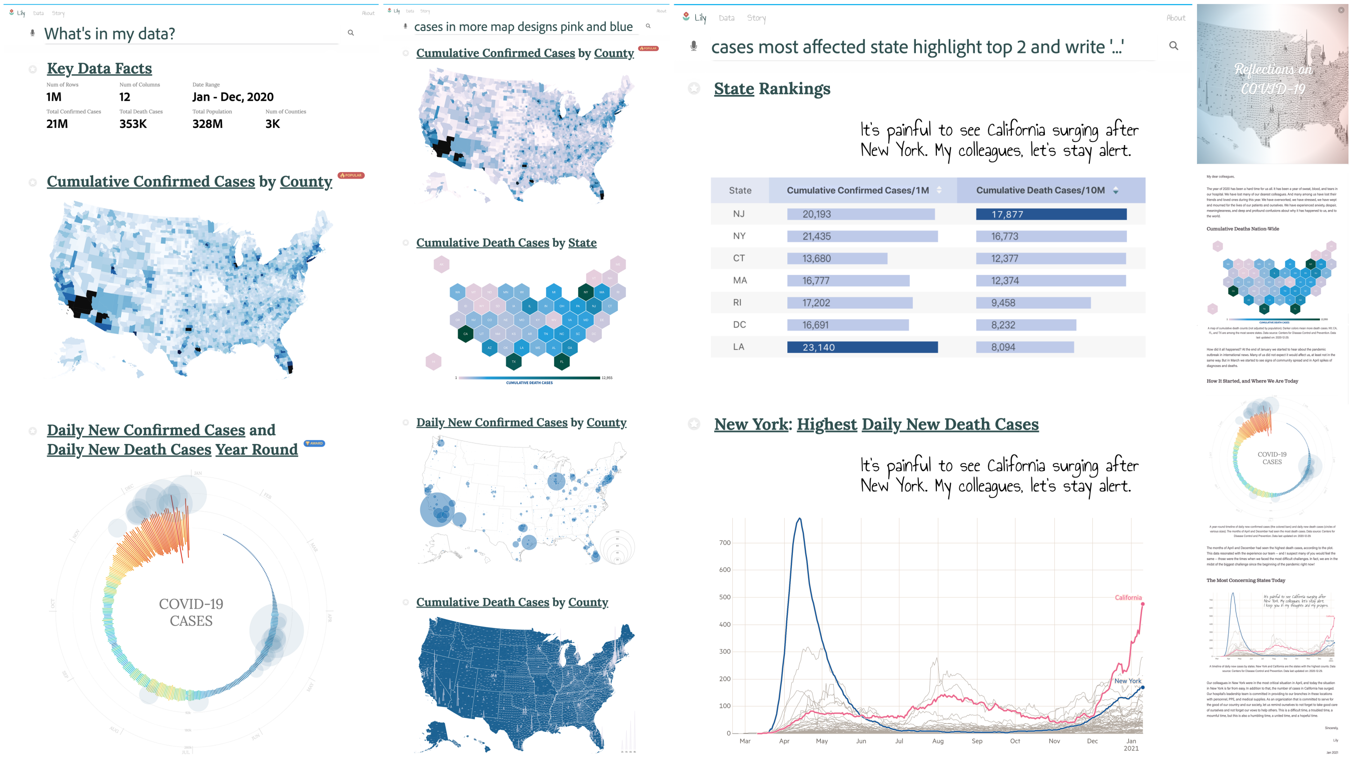
NL-driven visualization authoring has to be both **expressive** and **precise**. It is hard to achieve both goals at the same time. Expressive means being able to render what the author has (vaguely envisioned) on her mind, including the macro and micro design choices, the styles and so on. Being precise means being able to respond to the subtle changes in the author's expressions – parsing the author's intents accurately. Inevitably, as expressiveness increases, the space of possible mappings from natural to some visual language grows exponentially, making it much harder for NLP to have an accurate understanding of the author's intents.

Moreover, for a NL-driven authoring tool to be practically useful, **the entire workflow** (data analysis, visualization design, storytelling) should be readily accessible by natural language inputs. There should be no roadblocks. A single hurdle along the way can scare off masses of inexperienced potential authors. If Aunt Lily could create a visualization with a few words in a matter of seconds, but still needed to explore her data in Tableau or post-edit her data story in Adobe Illustrator, that would defeat the purpose of a fast and agile NL-driven authoring tool.

We need a better understanding of the natural language steering wheels along the authoring workflow. Therefore, we surveyed the literature and conducted interviews to crystalize the workflow – understand common steps taken by authors. Using this knowledge we designed lily, a data analysis - storytelling workflow directed by natural language. **We contribute:**

1. A framework for characterizing and automatically recommending visualization templates, including their discoverability, customizability, constraints, and compatibility.

*This was Zening's internship project at Adobe Research.



What's in my data?

Key Data Facts

Num of Rows	Num of Columns	Date Range	
1M	12	Jan - Dec, 2020	
Total Confirmed Cases	Total Death Cases	Total Population	Num of Counties
21M	353K	328M	3K

Cumulative Confirmed Cases by County

Daily New Confirmed Cases and Daily New Death Cases Year Round

cases in more map designs pink and blue

Cumulative Confirmed Cases by County

Cumulative Death Cases by State

Daily New Confirmed Cases by County

Cumulative Death Cases by County

cases most affected state highlight top 2 and write '...'

State Rankings

It's painful to see California surging after New York. My colleagues, let's stay alert.

State	Cumulative Confirmed Cases/1M	Cumulative Death Cases/10M
NJ	20,193	17,877
NY	21,435	16,773
CT	13,680	12,377
MA	16,777	12,374
RI	17,202	9,458
DC	16,691	8,232
LA	23,140	8,094

New York: Highest Daily New Death Cases

It's painful to see California surging after New York. My colleagues, let's stay alert.

Reflections on COVID-19

My dear colleagues,
Cumulative Deaths Nation-Wide
How It Started, and Where We Are Today
COVID-19
CASES
The Most Concerning States Today

Figure 2: Four snapshots from a data-to-story[E] workflow in LILY. (a) The author asked a macro-level question[F] "What's in my data?" LILY returned the fundamental *Key Data Facts*, and multiple data distributions across *space* and *time* using popular and award-winning designs.[X] (b) The author specified data and design preferences ("cases", "map", "pink and blue"). And the results were all cases maps in pink and blue colors. (c) The author wanted to *find extremes* ("most affected state", "top 2") and call out an insight ("highlight", "write ..."). LILY dimmed, highlighted, and annotated accordingly. (d) The author composed a letter to her colleagues. She specifically picked two charts and asked LILY to select a few more.[L]

2. A design for LILY, an NL-driven authoring tool that accommodates the entire workflow from data wrangling to storytelling.

By improving both expressiveness and accuracy along the entire authoring workflow, LILY brings us one step closer to the vision of custom visualizations created for and by the masses.

2 DESIGN GOALS

We designed LILY to achieve the FLEX goals:

FLEXIBLE[F] Recognize natural language instructions of various types, succinct or elaborate, at macro and micro levels, expressed with different word choices and styles, and tolerate errors and ambiguities.

LIGHTWEIGHT[L] Provide a simple user interface that does not demand much learning about the interface itself. Avoid overdemanding the author with the system input, or overwhelming them with the output.

END-TO-END[E] Support the entire data-to-story authoring workflow

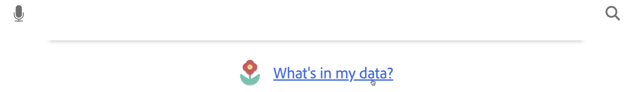
eXPRESSIVE[X] Support diverse visualization and data story designs. Allow authors to have control over the design details.

3 USER EXPERIENCE (UX) DESIGN

We show an end-to-end workflow[E] in which Aunt Lily overviewed **data**, customized **design** details, searched for an **insight**, and commanded a **story** being told (Fig. 2a-d). These macro and micro instructions[F] all go through a single NLI (Fig. 3).[L]

It had been a year since COVID-19 broke out in the US. Aunt Lily wanted to pull the latest data, reflect, and write a letter to strengthen her colleagues, who had been fighting unprecedentedly hard and long. She selected a dataset from CDC.

At first, she didn't know what to say. The mixed-initiative agent suggested an ice-breaking query: "What's in my data?"[F] She clicked on it and saw a variety of **data overviews** (Fig. 2a)[X], each covering a distinct subset of **data attributes**, exposed as underlined links in the automatically generated chart titles.[L] If clicked, an attribute will become a new query.[F]



What's in my data?

Figure 3: The user interface of LILY is minimal[L]– a text (and voice) input[F] and a mixed-initiative agent that suggests and disambiguates queries.[F]

But Aunt Lily didn't want to fixate on any specific attributes yet. Instead, she loosely expressed her interests[F] as "cases in more map designs". Now the only attributes visualized are those matching "cases", plus the auxiliary geographical fields required by "maps". Meanwhile, more diverse **map designs** are shown: besides the choropleth map, there is also a hexbin map, a bubble map, and a spike map.[X] The design names show up as tags beside the chart titles.[L] They can also form future queries.

The **colors** were also diversified: *cumulative confirmed cases* in sequential shades of blue, *cumulative confirmed cases* in diverging shades of pink and teal, and *daily new confirmed cases* in orange. Such diversification follows Qu and Hullman's consistency principles [32], which says *different data should wear different colors*.[L] It also reveals many color scheme options to the author.[X] For instance, Aunt Lily picked "pink and blue" (Fig. 2b).

The spike map made Aunt Lily quite concerned about New York.

She revised her query to "cases most affected state". Because the query conveyed an intent to *find extremes*, the result visualization contained **highlights** – the mark representing the highest value was emphasized whereas all others were dimmed. Because the query said "state", all charts were aggregated at the *state* level. The word "most" in the query was retold as rankings and highest in the chart titles. The synonyms provide a flexible way to explore data by a statistical **insight**.[F]

In the timelines, Aunt Lily saw that California is surpassing New York in recent death counts. She held the microphone 🎙 and said: *"Highlight California and New York and write 'It's painful to see California surging after New York. My colleagues, let's stay alert."'* And the system updated the highlights and **annotated** all charts (Fig. 2c).

Then Aunt Lily said: "Write a letter using the line chart, and some other charts we've seen." Now the annotated timelines became a part of a scrolly, which was automatically pre-populated[L] by the year round radial, the hexbin map, and the spike map (serving as a background image for the title). The year round radial was in the letter because Aunt Lily had clicked its storytelling icon ✨ during "What's in my data?" The hexbin map and the spike map were added by the system because Aunt Lily had said "use ... some other charts" but did not specify which ones (Fig. 2d).[L] Aunt Lily went on to edit the titles, the captions, and the body text of her letter. She could *add*, *remove*, *replace*, and *reorder* the visualizations in her letter by saying queries like "replace the hexbin map with the one that had blue bubbles" – if she could only recall certain chart names, data attributes, or colors. Alternatively, she could collect story pieces ✨ from her query history, and cut ✂ and move ✥ pieces in her letter.[F] Finally, Aunt Lily signed her letter and sent it.[E]

4 Envisioning the System

Lily consists of a user interface, a natural language processing module, a recommender, a renderer, and a collaborative platform that serves repositories of data and insights, and templates and components (Fig. 4). A *template* is a data story or visualization with customizable data and design *components* (e.g., rebind data attributes, hide axes, replace a drop down menu with a slider). Any data story or visualization that can be represented as a string (e.g., HTML and D3 [3], Jupyter Notebook and Altair [46], R Markdown and ggplot2 [49]) may become a template in Lily. When all components are specified, a template becomes a renderable *specification*. Each specification is associated with a string signature, which can be any appropriate linearized representation. Each signature is encoded using a deep language model (e.g., GPT-3 [4], CPM-2 [55], BERT [10], ELMo [31]), possibly fine-tuned on available examples (NL2VIS specifications, revisions, constraints, and comments, scrapped or synthesized [24]), building on recent approaches [5].

Query To begin, the user speaks or types a natural language query. The query may refer to the resources on the collaborative platform: data tables, attributes, values, insights, and transformations; design templates, components, and customizations. The same language model encodes both queries and specification signatures, making the distance between them measurable.

Recommend The recommender searches and ranks the top k specifications that best match a query given its context (Fig. 4). First, it filters templates with the data constraints (e.g., "cases") and design constraints (e.g., "red map") in the query. For each template, the recommender enumerates all possible customized *specifications* based on the query. We bind attributes to template variables according to typical visualization principles using automated constraint satisfaction tools [29]. Each attribute is encoded, and the distance from the encoded attribute and the encoded query is used as a soft constraint to guide the solver [29]. The result is all possible variable configurations that satisfy the template and preferring attribute similarity.

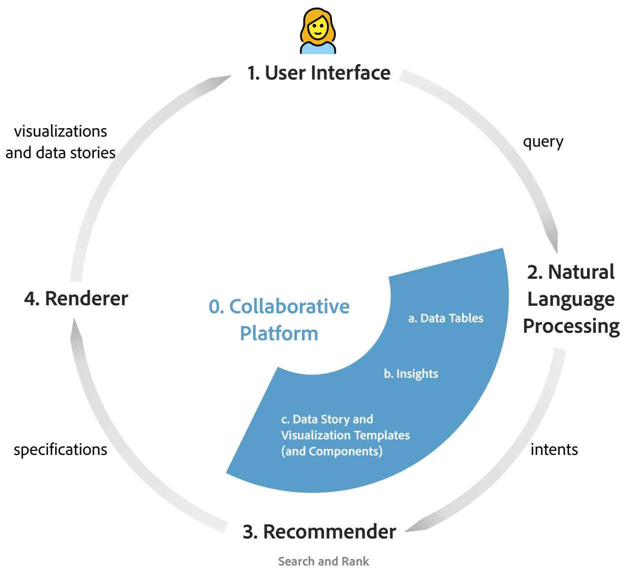

1. User Interface
query
2. Natural Language Processing
intents
3. Recommender
Search and Rank
specifications
4. Renderer
visualizations and data stories
0. Collaborative Platform
a. Data Tables
b. Insights
c. Data Story and Visualization Templates (and Components)

Figure 4: System architecture. Modules 1-4 form one iteration; module 0 supplies modules 2-3 with the following resources: data tables shared by authors and organizations (a), insights precomputed or as function calls (b), and customizable templates and components (c). The user directs a data story by typing or speaking a query (1). A query has a *context*, including the resources on the collaborative platform (0) and the interaction history (1-4). Within such context, Lily detects query intents (2) and recommends specifications – by searching under the constraints of attributes, insights, templates, components, and customization options, and then ranking the results (3).

Each recommendation is considered a set of terms r. Each recommendation set r is scored and ranked using the formula:

$$\text{score} = \frac{1}{|r|}\sum_{x\in r} w(x)\cdot x \tag{1}$$

where $w_q(x)$ is a weighting function

$$w(x) = \max\{\text{distance to query} \in [0,1],\\ \text{popularity on collaborative platform} \in [0,\lambda]\}$$

and $|r|$ is the size of the recommendation set.

Render Our text-oriented approach is compatible with popular visualization authoring tools and libraries such as d3.js [3], Vega [37], plotly [16], Data Illustrator [22], and so on. The renderer can use typical libraries to output (interactive) visualizations in svg or canvas and provide an HTML snippet for developers to embed the visualizations in their own web pages.

Upon rendering, each specifications becomes a visualization in the user interface. The user can (1) keep a visualization as a story piece, (2) revise her original query to start a new iteration.

Regardless of the choice, the selected visualization is appended to the story, and is used as part of the context in future iterations. On each iteration, the sequence of previous visualizations and the query issued by the author are embedded together as a sequence and used for matching during search and recommendation.

4.1 System Discussion

We intend our design to enable study of the limits of NLP methods as well as the capabilities. To that end, we emphasize breadth in our requirements.

Queries and Intents In all NLP Vis systems, the goal is to infer an author's *intent* (what they are trying to achieve) given their *query* (the actual sequence of words they typed or spoke) [45].

Rather than try to derive a vocabulary of intents from a small set of actions supported by a system, we treat every stemmed n-gram as a potential intent, and every possible instantiation of a template as a potential recommendation. While we avoid needing to design a universe of discourse ourselves, we are now more dependent on the amount and quality of the data. However, Lily can use pre-trained models at first while data is collected, and then apply joint learning techniques [14, 53] to learn embeddings from associated pairs of templates and queries, validated by author selections.

To develop the initial vocabulary, we anticipate scraping visualization textbooks, papers, tutorials, and forums for the top-*d*-frequently-used words and phrases (we will call them *expressions* from now on), we obtain a feature space of *d* dimensions. Since Shakespeare's vocabulary was estimated to be between 15,000 and 25,000 words [6, 11], we hypothesize that with a *d* of hundreds of thousands, most queries and contexts (data, current visualizations, recent history) can be expressed in this feature space.

Although we do not explicitly categorize utterances during interaction, we can characterize the types of queries we anticipate. Queries need not be complete ("mean" of what, "aggregation" of what, "correlation" of what and what) nor non-divisible: "normal" and "distribution" are valid expressions, even if "normal distribution" is much more likely and specific.

We anticipate a variety of potential utterances, all of which can *potentially* be interpreted in our general approach of matching utterances to templates.

- a (partial) specification ("write a letter using ..."),
- a direct revision ("change colors to pink and blue")
- a constraint (e.g., "axis titles on top"),
- a comment (e.g., "it's too colorful"),
- a concept (e.g., "overview", "surprise me!", "correlation", "normal distribution")
- a reference (e.g., "cases", "the one that had blue bubbles") or a question (e.g., "what's in my data?")

Improving Data Quality with Computed Labels To improve the matching of variables to data columns, we can compute insights [20] on all available data. Data facts as implemented in Cui et al. can also be wrapped as functions [7].

To improve the matching of templates to queries, we can apply weak supervision by writing multiple simple yet unreliable data labelers [33]. For example, templates from a particular website with a name that includes *scatter* can be tagged as a scatter plot. In this way, data quality can be improved incrementally and continuously by adding new labels.

Analysts and Contributors We intend Lily for three communities of users: data analysts (our primary focus), data contributors who want to enable more use of their data, and template contributors who want to popularize their novel chart invention, distinct styles, animated transitions, or new interaction techniques. We intend to support analysts of any skill level, looking for opportunities to automate decisions that complicate typical tools.

Template Registration When a new template is contributed to the platform, we ask the contributor four questions:

1. Discoverability: Please describe this template in your own words. What data story does it tell? This will make it easier for other users to find your template.
2. Constraints: What kind of insights, data distributions, cardinalities, themes etc. is this template best for? Are there any occasions or situations where using this template would be inappropriate?
3. Customizability: What parts of the template can be changed by the user (e.g., variables, constants, APIs)?
4. Compatibility: Can this template be repeated or linked or combined with other templates or interactive widgets?

5 Related Work

Manual Visualization Authoring Tools Most of the current business intelligence [23, 28, 42] and web analytics [1, 13] offerings focus on a manual workflow for data exploration and visualization authoring, where analysts need to select which variables to explore, decide what kind of visualization charts to use, inspect if useful insights exist, and repeat. These tools are powerful yet too sophisticated for non-experts who have limited data science knowledge or graphic design skills.

Automated Analytics Tools More recently, automated data analytics tools have gained great traction [21]. Data science models have been proposed to automatically select interesting data variables [8, 9, 47, 50, 51] or generate appropriate visual representations [15, 25, 29]. Among commercial products, Tableau offers a "Show Me" panel, which automatically recommends chart types based on users' variable selections [26]. These automated analytics tools can provide helpful guidance, but they lack an input channel for users to communicate their intents so as to get desired results quickly.

Search Analytics Tools During the last decade, data analytical workflows driven by natural language queries have emerged [52,54]. Adobe Analytics pioneered a technology prototype called DataTone that allows people to retrieve statistical data facts from a dataset in natural language [2, 12]. Tableau and ThoughtSpot also released similar "Search Analytics" products [27,38,44]. While these tools provide an innovative natural language interface for users to specify their analytical intents, they often require the users to be familiar with the data and specify concrete data queries, such as "Show me a barchart of sum of order total value by geo cities". However, when authoring a data story, non-expert users often only have vague ideasor questions, such as "How is the revenue". It is difficult for them to map their ideas to the concrete queries. To address this challenge, our system uses intent recognition techniques to infer users' analytical interests from ambiguous or partial queries and uses recommendation models to generate complete analytics specifications.

6 Two Case Studies

We show how our proposed system design works in two extreme cases: using Lily as a "one-button machine" that generates data stories with minimal human intervention, and using Lily for elaborate, precise control. The point is, Lily works in both scenarios, which shows it flexibility.[F] And being able to recreate the expert dashboard in Fig. 1 further demonstrates Lily's expressiveness.[X]

Case 1: One-Button Machine Uncle Nelson is a **busy** user. He has the same data as Aunt Lily, but has no time for exploration, customization, or composition. He wants a data story and he wants it now! He says: "data story." And Lily returns a poster, a storyline, a data video, and so on. The poster compares COVID-19 case counts in various cities in a small multiples design. Data attributes were binded and data insights were annotated by Lily. The original poster template had won an Information is Beautiful award [19]. The storyline used a Calliope [39] template, showing insights captured by DataShot [48]. The data video template was contributed by editors at Johns Hopkins University. Uncle Nelson prints out the poster and heads to his meeting.

In this case, Lily detects the intents "data", "story", and "data story", which is more specific and likely to be accurate than "data" and "story" alone. Since the most likely intent is at "data story" level, Lily's recommender decided to diversify at the next-level specificity – the **genre** of the story. Awards and the popularity affected the ranking of the binded and customized templates. **Data attributes**, being top-level and unspecified in the query, is also diversified. If Uncle Nelson were to implement these data stories on his own, he would need to start with researching data story templates, and handle the data binding and customization on his own, which he would not have time for.

Case 2: Precise Control Uncle Rodney is an **expert** user. He and Aunt Lily sat down together to replicate the dashboard by their ex-colleague (Fig. 1). To achieve this goal, they need precise control over the visual encodings, the formatting of axes and fonts, the layout, the colors of the alternating rows, and so on. Our supplementary material records 19 queries they said in order to replicate the dashboard. When we compared their queries with the equivalent 19 groups of concrete UI interactions in Tableau, we became more hopeful – systems like Lily, if implemented well, could alleviate the pain of technical know-hows for many people.

Acknowledgments

We thank Aunt Lily, Uncle Rodney, and Uncle Nelson for letting us using their names in our paper. We thank Flaticon.com for the icons 🌷📖✂️✥, and OpenMoji.org for 👩.

References

[1] Adobe. Adobe Analytics, 2020. https://www.adobe.com/analytics/adobe-analytics.html.

[2] A. Blog. Analytics as simple as asking a question, March 2015. https://theblog.adobe.com/datatone/.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization & Computer Graphics*, (12):2301–2309, 2011.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. 2020.

[5] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. Ponde, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[6] H. Craig. Shakespeare's vocabulary: Myth and reality. *Shakespeare Quarterly*, 62(1):53–74, 2011.

[7] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-Viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE transactions on visualization and computer graphics*, 26(1):906–916, 2019.

[8] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Rapid data exploration through guideposts. In *Workshop on Data Systems for Interactive Analysis (DSIA) at IEEE VIS 2017*, 2017.

[9] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, vol. 10, 2017.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1423

[11] A. Ellegård. Estimating vocabulary size. *Word*, 16(2):219–244, 1960.

[12] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. DataTone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 489–500, 2015.

[13] Google. Google Analytics, 2020. https://analytics.google.com/analytics/web/.

[14] M. Grechkin, H. Poon, and B. Howe. Ezlearn: Exploiting organic supervision in large-scale data annotation. *arXiv preprint arXiv:1709.08600*, 2017.

[15] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. VizML: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.

[16] P. T. Inc. Collaborative data science, 2015. https://plot.ly.

[17] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372. ACM, 2011.

[18] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 547–554, 2012.

[19] T. Kekeritz. Weather radials: An infographic on heat waves and snow storms in 35 cities around the globe. *Raureif*, 2013. http://www.weather-radials.com.

[20] P.-M. Law, A. Endert, and J. Stasko. Characterizing automated data insights. In *2020 IEEE Visualization Conference (VIS)*, pp. 171–175. IEEE, 2020.

[21] D. J.-L. Lee. Insight machines: The past, present, and future of visualization recommendation, February 2020. https://medium.com/multiple-views-visualization-research-explained/insight-machines-the-past-present-and-future-of-visualization-recommendation-2185c33a09aa.

[22] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[23] Looker Data Sciences, Inc. Looker, 2020. https://looker.com/.

[24] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin. Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1235–1247, 2021.

[25] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions On Graphics (Tog)*, 5(2):110–141, 1986.

[26] J. Mackinlay, P. Hanrahan, and C. Stolte. Show Me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics*, 13(6):1137–1144, 2007.

[27] R. Markas. Ask Data: Simplifying analytics with natural language, November 2018. https://www.tableau.com/about/blog/2018/11/ask-data-simplifying-analytics-natural-language-98655.

[28] Microsoft. Power BI, 2020. https://powerbi.microsoft.com.

[29] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. *IEEE Transactions on Visualization and Computer Graphics*, 2018. .

[30] A. Narechania, A. Srinivasan, and J. Stasko. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE transactions on visualization and computer graphics*, 2020.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[32] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 2017. .

[33] A. Ratner, S. Bach, P. Varma, and C. Ré. Weak supervision: the new programming paradigm for machine learning. *Hazy Research. Available via https://dawn. cs. stanford. edu//2017/07/16/weak-supervision/. Accessed*, pp. 05–09, 2019.

[34] D. Ren, B. Lee, and M. Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE transactions on visualization and computer graphics*, 25(1):789–799, 2018.

[35] A. Satyanarayan and J. Heer. Authoring narrative visualizations with Ellipsis. In *Computer Graphics Forum*, vol. 33, pp. 361–370. Wiley Online Library, 2014.

[36] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design

environment. In *Computer Graphics Forum*, vol. 33, pp. 351–360. Wiley Online Library, 2014.

[37] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2016. `http://idl.cs.washington.edu/papers/reactive-vega-architecture`.

[38] V. Setlur, M. Tory, and A. Djalali. Inferencing underspecified natural language utterances in visual analysis. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 40–51, 2019.

[39] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):453–463, 2020.

[40] A. Srinivasan and J. Stasko. Nl4dv: Toolkit for natural language driven data visualization. 2016.

[41] N. Sultanum, F. Chevalier, Z. Bylinskii, and Z. Liu. Leveraging text-chart links to support authoring of data-driven articles with vizflow. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2021.

[42] Tableau Software. Tableau, 2020. `https://www.tableau.com`.

[43] J. Thompson, Z. Liu, and J. Stasko. Data animator: Authoring expressive animated data graphics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, vol. 21, 2021.

[44] ThoughtSpot Inc. ThoughtSpot, 2020. `https://www.thoughtspot.com/`.

[45] M. Tory and V. Setlur. Do what I mean, not what I say! Design considerations for supporting intent and context in analytical conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 93–103. IEEE, 2019.

[46] J. VanderPlas, B. Granger, J. Heer, D. Moritz, K. Wongsuphasawat, A. Satyanarayan, E. Lees, I. Timofeev, B. Welsh, and S. Sievert. Altair: Interactive statistical visualizations for python. *Journal of open source software*, 3(32):1057, 2018.

[47] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. SEEDB: Efficient data-driven visualization recommendations to support visual analytics. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, vol. 8, p. 2182. NIH Public Access, 2015.

[48] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. DataShot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics*, 26(1):895–905, 2019.

[49] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. `http://ggplot2.org`.

[50] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016.

[51] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2648–2659. ACM, 2017.

[52] T. Wu, K. Wongsuphasawat, D. Ren, K. Patel, and C. DuBois. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2020.

[53] S. T. Yang, K.-H. Huang, and B. Howe. Jecl: Joint embedding and cluster learning for image-text pairs. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8344–8351. IEEE, 2021.

[54] B. Yu and C. T. Silva. FlowSense: A natural language interface for visual data exploration within a dataflow system. *IEEE transactions on visualization and computer graphics*, 26(1):1–11, 2019.

[55] Z. Zhang, Y. Gu, X. Han, S. Chen, C. Xiao, Z. Sun, Y. Yao, F. Qi, J. Guan, P. Ke, et al. Cpm-2: Large-scale cost-effective pre-trained language models. *arXiv preprint arXiv:2106.10715*, 2021.

[56] J. Zong, D. Barnwal, R. Neogy, and A. Satyanarayan. Lyra 2: Designing interactive visualizations by demonstration. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

Supplementary Material

Aunt Lily Can Say Her Visualizations

A Expert Dashboard in Lily and Tableau

Tableau: Us re-creating the expert dashboard in Fig. 1 in Tableau. We take the easiest route we know of. We omit the googling about how to do certain things in the UI in these notes.
Novice: How a novice author such as Aunt Lily could achieve the same result using Lily.
Expert: How an expert author could do the same.

A.1 Specify X Y Encoding

Tableau: Find each of the eight measures – '# Enc', '# Orders', 'Administrations', 'Admin Dose', 'Medication Scan Required...', 'Med Scanned At Admin', 'Med Scan at Admin % ...', and 'Age At Encounter' – in the 'Data' panel by eyeballing or searching. Drag each measure to the end of the Columns shelf. Find 'Day of the Week' and drag it to the 'Rows' shelf.
Novice: "Create a bar table that look like those in Tableau, using Day of the Week and # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter."
Expert: "Tableau-style concatenated bar charts. Day of the Week by # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter."

A.2 Add Colors

Tableau: Make sure 'Marks' is set to 'All'. Drag 'Measure Names' to 'Color'.
Novice: "Add colors." "Color each bar chart differently." "Color the distinct fields."
Expert: "Color by measure names." "Color the quantitative fields."

A.3 Take Averages

Tableau: For two measures ('Admin Dose', 'Age At Encounter'), click on their drop-down arrow, 'Measure (Sum)', 'Average'.

A.4 Add Labels

Tableau: Make sure 'Marks' is set to 'All', click 'Label', check the box 'Show mark labels' and make the font bold 'B'. Check the 'Allow labels to overlap other marks' checkbox. Move mouse to 'Med Scan at Admin %', right click and select 'Format', click on the 'Pane' tab, click on the 'Default Numbers' dropdown menu, select 'Percentage', set 'Decimal Places' to 0. Move mouse to 'Avg. Admin Dose', right click and select 'Format', click on the 'Pane' tab, click on the 'Default Numbers' dropdown menu, select 'Numbers (Custom)', set 'Decimal Places' to 0. Move mouse to 'Avg. Age At Encounter', right click and select 'Format', click on the 'Pane' tab, click on the 'Default Numbers' dropdown menu, select 'Numbers (Custom)', set 'Decimal Places' to 0.
Novice: "Show the numbers in bold. Don't show the .00s."
Expert: "Label the values in bold. Don't show decimals."

A.5 Hide Axis Numbers and Ticks

Tableau: For each of the eight measures, right click on the axis title, select 'Edit Axis...'. In the popup window, click on the 'Tick Marks' tab, under 'Major Tick Marks' select 'None'. Click on the top-right 'X' to close the popup window.
Novice: "Don't show the ruler."
Expert: "Hide axis ticks and numbers."

A.6 Sort from Sunday to Saturday

Tableau: Click on 'Day of the Week' dropdown arrow, click on 'Sort...'. In the popup window, set 'Sort By' to 'Manual', and sort the days from Sunday to Saturday.
Novice: "Start the week on Sunday."
Expert: "Sort Sunday to Saturday."

A.7 Bold All Axis Titles

Tableau: Right-click on the white space in the worksheet, select 'Format...'. Click on the 'Format Font' icon. Under 'Default', click on the 'Header' dropdown menu and toggle the 'B' icon to bold the header text.
Novice: "Make # Enc, # Orders, Administrations, etc. bold."
Expert: "Bold all axis titles."

A.8 Alternate Row Background Colors

Tableau: Click on the 'Format Shading' icon. Select 'Rows' tab. Under 'Row Banding', select a light blue color for 'Pane', select the same color for 'Header', and drag 'Band Size' to 1.
Novice: "Blue and white rows."
Expert: "Alternate blue and white bands between rows."

A.9 Reduce Lines

Tableau: Click on the 'Format Borders' icon. Select 'Rows' tab. Under 'Row Divider', click on the 'Pane' drop down menu and select 'None'. Select 'Columns' tab. Under 'Column Divider', click on the 'Header' drop down menu and select 'None'. Click on the 'Format Lines' icon. Select 'Columns' tab. Under 'Lines', set 'Grid Lines' to 'None'.
Novice: "Keep grid lines etc. to the minimal." "Hide all non-essential/unnecessary/auxiliary light gray lines."
Expert: "Reduce grid lines." "Minimalistic flat design." "Hide the row and column dividers and the grid lines."

A.10 Move Axis Titles to the Top

Tableau: Drag another '#Enc' measure and put it after '#Enc' in 'Columns'. Right-click the second copy of '#Enc' and select 'Dual Axis'. Double-click the top axis title. In the popup window, select 'Tick Marks' tab. Under 'Major Tick Marks', select 'None'. Click on the top-right 'X' to close the popup window. Double-click the bottom axis title. In the popup window, select 'General' tab. Under 'Axis Titles', erase 'Title'. Click on the top-right 'X' to close the popup window. Repeat for '# Orders', 'Administrations', 'Avg. Admin Dose', 'Medication Scan Required...', 'Med Scanned At Ad...', 'Med Scan at Admin % If R...', and 'Avg. Age At Encounter'. In 'Marks' panel, make sure 'All' fields are selected, select 'Bar' marks. Click 'Color', 'Edit Colors...', select the 'Tableau 10' color palette. Assign the same colors for the duplicated measures.
Novice: "Move # Enc, # Orders, Administrations, etc. to the top."
Expert: "Show axis titles on top." "Top axis titles."

A.11 Add Dropdown for Time

Tableau: Drag 'Month, Year of Med Order Instant' to 'Filters' card. In the popup window, select 'Month/Year'. Click 'Next >'. Select multiple months by checking the boxes. Click 'OK'. In the 'Filters' card, click on the dropdown arrow of 'MY(Month, Year of Med Order Instant)' and select 'Show Filter'. In the filter card, click on the dropdown arrow and select 'Edit Title...'. In the popup window, set the title as 'Month, Year of Med Order Instant'. In the filter card, click on the dropdown arrow and select 'Multiple Values (dropdown)'. Now the filter card appears as a dropdown menu.
Novice: "Allow people to select multiple Month, Year of Med Order Instant."
Expert: "Add a multi-value dropdown menu for Month, Year of Med Order Instant."

A.12 Add Dropdown for Antibiotics

Tableau: Drag ‘Ordered Med Name’ to ‘Filters’ card. In the popup window, click the ‘Use all’ radio button. Click ‘OK’. In the ‘Filters’ card, click on the dropdown arrow of ‘Ordered Med Name’ and select ‘Show Filter’. In the filter card, click on the dropdown arrow and select ‘Single Value (dropdown)’.
Novice: "Let people change Antibiotics."
Expert: "A single-selection dropdown menu for Antibiotics."

A.13 Hide Color Legend

Tableau: Delete the color legend for ‘Measure Names’ by clicking the ‘X’ icon on the upper-right.
Novice: "Don't show which color is which field."
Expert: "Hide color legend." "Omit the color legend."

A.14 Edit Dashboard Title

Tableau: Create a new dashboard. Drag the worksheet to the new empty dashboard. Check ‘Show dashboard title’. Double click on the default title ‘Dashboard 1’ and name the dashboard ‘Medication Administration Details - Weekday/Day of Month by Med’. Select Tableau Bold 45 pt font, blue color, align center.
Novice: "Dashboard." Then directly edit the dashboard title to ‘Medication Administration Details - Weekday/Day of Month by Med’. Select 18 pt bold font, blue color, align center.
Expert: "Make it a dashboard titled `Medication Administration Details - Weekday/Day of Month by Med', 45 pt bold, blue color, align center."

A.15 Edit Visualization Title

Tableau: Double click on the default worksheet title ‘Sheet 1’ and name the sheet ‘Breakdown by Day of the Week’. Select Tableau Bold 36 pt font, black color, align left. Right click on the sheet title, select ‘Format Title’, under ‘Title Shading’ select a light gray color.
Novice: Directly edit the visualization title ‘Breakdown by Day of the Week’. Select 36 pt bold font, black color, lightgray background, align left.
Expert: "Title the bar charts `Breakdown by Day of the Week', 36 pt bold, lightgray background, align left."

A.16 Hide the Y Axis Title

Tableau: Right click on the rows axis title ‘Day of the...’, select ‘Hide Field Labels for Rows’.
Novice: "Erase the text Day of the Week."
Expert: "Hide the axis title Day of the Week."

A.17 Edit Dashboard Layout

Tableau: Click on the dashboard, click on the ‘Layout’ tab on the left. Set ‘w’ and ‘h’ to 1600 and 500. Select the dashboard title, in the ‘Layout’ tab, check the ‘floating’ checkbox. Repeat this for the two filters and the small multiples bar chart. Select the dashboard title, in the ‘Layout’ tab, set its ‘Position’ to ‘x = 0’, ‘y = 10’, and set its size to ‘w = 1600’, ‘h = 50’. Select the ‘Month, Year of Med Order Instant’ filter. In the ‘Layout’ tab, set its ‘Position’ to ‘x = 495’, ‘y = 60’, and set its size to ‘w = 300’, ‘h = 50’. Select the ‘Ordered Med Name’ filter. In the ‘Layout’ tab, set its ‘Position’ to ‘x = 805’, ‘y = 60’, and set its size to ‘w = 300’, ‘h = 50’. Select the small multiples bar chart. In the ‘Layout’ tab, set its ‘Position’ to ‘x = 10’, ‘y = 120’, and set its size to ‘w = 1580’, ‘h = 350’.
Novice: Drag and drop the elements in the dashboard and the X Y W H layout boxes should appear like in Adobe Photoshop.
Expert: "Bring out the layout panel." or "Edit dashboard layout." or simply "Layout." Or do this in queries "Make the dashboard 1600px wide and 500px tall." "The dashboard title should be 1600px wide and 50px tall, position it at 0,10." "Position the two dropdown menues under the dashboard title. Each should be 300px wide and 50px tall, align left." "Put the bar chart at X=10, Y=120. It's size should be 1580x350." or "Put the bar chart underneath the dropdown menues. It should take up the entire width of the dashboard".

A.18 Add Timestamp

Tableau: Make sure the dashboard is selected. In the ‘Objects’ panel, click the ‘Floating’ toggle, and drag ‘Text’ to the dashboard. In the popup window, write ‘Updated 6/7/20’, and bold ‘6/7/20’. Select the text and click on the ‘Layout’ tab. Set its ‘Position’ to ‘x = 1490’, ‘y = 10’, and set its size to ‘w = 100’, ‘h = 50’.
Novice: Click the textbox icon and add the timestamp via direct manipulation.
Expert: "Write `Updated 6/7/20' and bold `6/7/20', 36 pt, right aligned, and center aligned to the dashboard title."

A.19 Publish as Tabs

Tableau: Double-click ‘Dashboard 1’ at the bottom of the window and rename the dashboard to ‘Med Admin Details - Weekday/Day of Month by Med’.
Click on the ‘Server’ menu and select ‘Publish Workbook...’. In the popup window, select this dashboard along with three others and check the ‘Show sheets as tabs’ checkbox. Click ’Publish’.
(Recreated with Tableau Desktop Professional Edition, 2020.3.10)
Novice: "This is a tab. The other tabs should be Med Admin Details - by Med Name and Med Admin Details - by Order. Publish these three tabs together."
Expert: "Add in Med Admin Details - by Med Name and Med Admin Details - by Order as separate tabs in this dashboard. Publish."