Responsive Data Visualization

Information Visualization, SS 2022
2022-05-11

Group G2:
Korduba, Schintler, Steinkellner

Responsive Web Design

- Predominant web design philosophy
- Visualization responsiveness can affect [Andrews, 2018]:
  - Layout
  - Display density
  - Interaction
- Content adapts to the characteristics of the display device

Responsive Web for Mobile

- **Smartphones**
- **Tablets**
- Wearables
- Head-mounted displays
- ...

Picture is taken from https://pixabay.com free image stock, no attribution required

Native vs. Responsive Web

	Technology	Advantages
Native	Android: Kotlin / Java iOS: Swift / Objective-C	• Platform specific features • Performance
Responsive Web & PWA	pure JavaScript React Angular	• Code sharing to all platforms • Reduced time to market • Large 3rd-party ecosystem (e.g. d3)

Mobile Advantages

- Touch interaction support
- High resolution (pixels per inch)
- Access to sensor data:
  - Location
  - Orientation
  - Accelerometer
  - …
- Widespread availability

Image: "Data@Hand: Fostering Visual Exploration of Personal Data on Smartphones Leveraging Speech and Touch Interaction" [Kim et al. 2021]

Mobile Challenges

- Low performance
- No hover for tooltips
- Fat finger problem, lack of precise pointing
- Small screen
- Different intended viewing timespan (glanceability is often key)

Responsive Visualization Patterns

Established Design Patterns

- “Techniques for Flexible Responsive Visualization Design” [Hoffswell, Li, and Liu 2020]
  - Survey of 231 responsive visualizations
  - Identifies four design guidelines
- “Design Patterns and Trade-Offs in Responsive Visualization for Communication” [Kim, Moritz, and Hullman 2021]
  - Survey of 378 visualization pairs
  - Identifies 76 total strategies
- “Responsive Data Visualisation” [Adler, Jahaj, Petritz and Yeli 2021]
  - Identifies 9 visualization patterns

Terminology

- 76 strategies by [Kim, Moritz, and Hullman 2021]
- Grouped into target and action categories

- Action categories:
  - Recompose
  - Rescale
  - Transpose
  - Reposition
  - Compensate

- Target categories:
  - Data
  - Encoding
  - Interaction
  - Narrative
  - References / Layout

Proposed Patterns

P1: Scaling the Chart Down

P2: Using Viewport-Specific Images

P3: Data Generalization

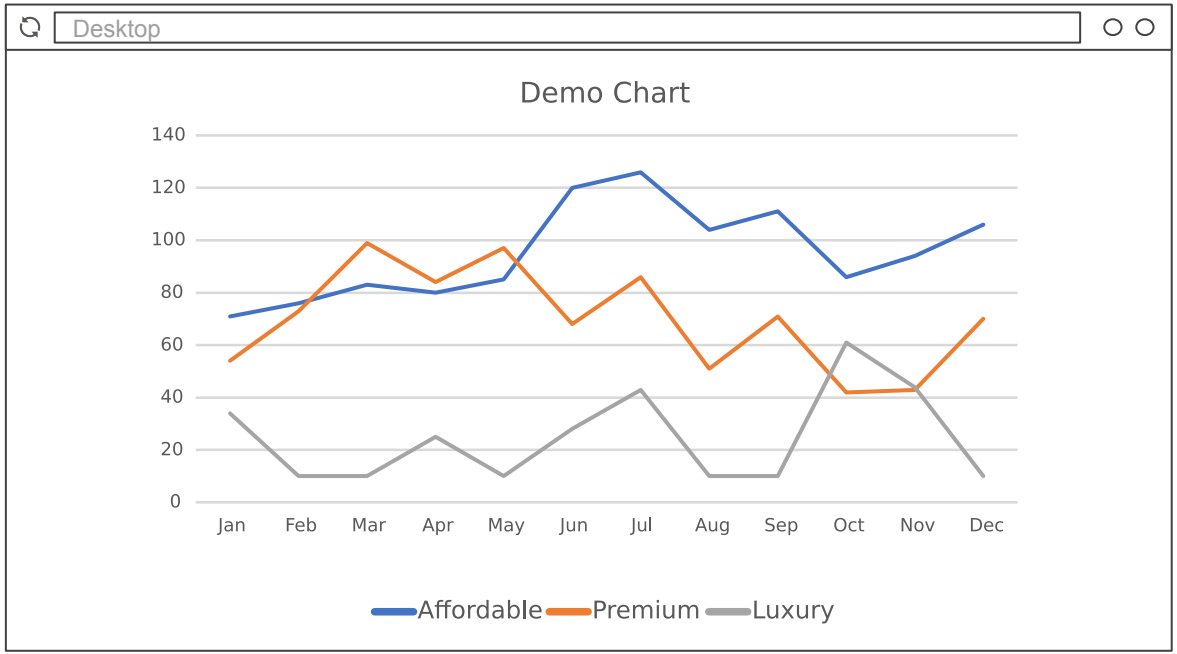
P4: Semantic Zooming

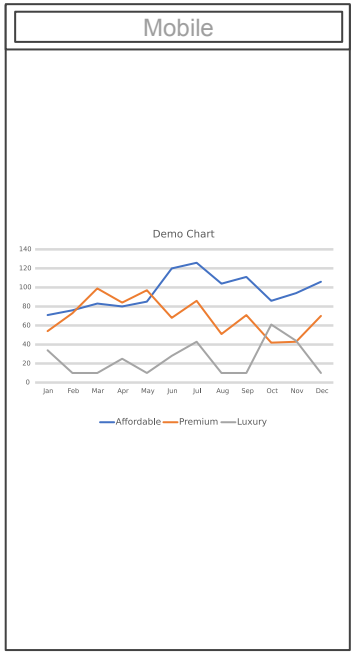
P5: Viewport-Specific Layouting
- P5a: Automatic
- P5b: Manual

P6: Changing Components
- P6a: Axes, tick marks, grid lines
- P6b: Labels

P7: Using different charts

P1: Scaling the Chart Down





P1: Scaling the Chart Down

- Resize chart to device or parent dimensions
- Disregards concerns about readability, explorability, etc.

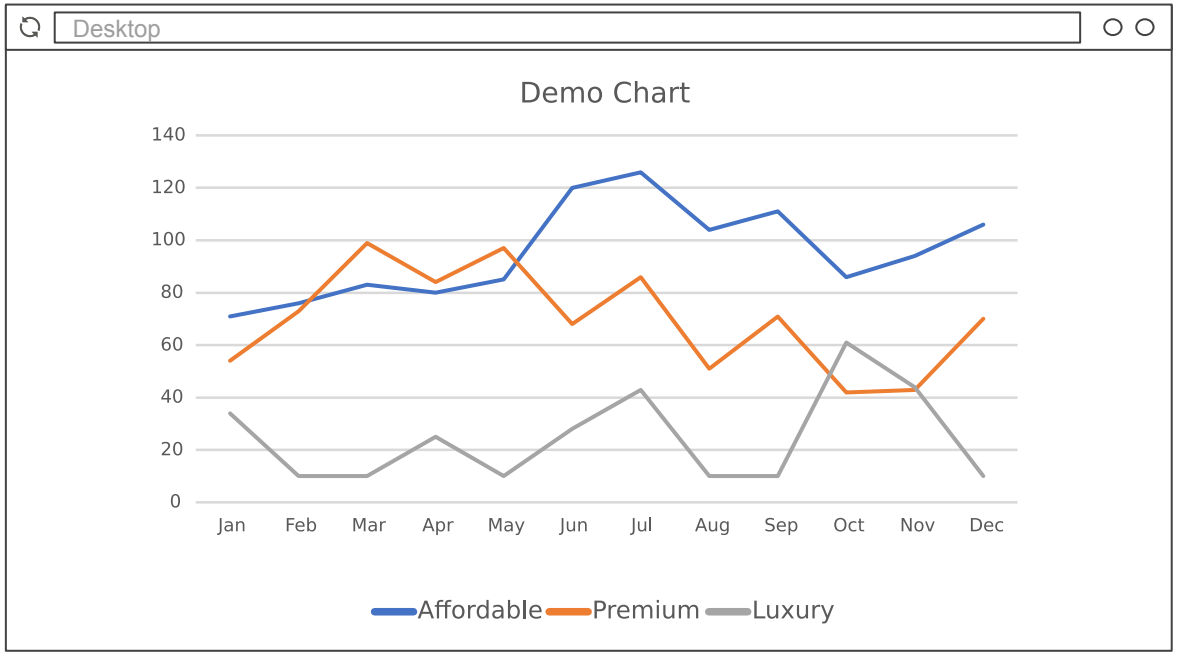
Advantages

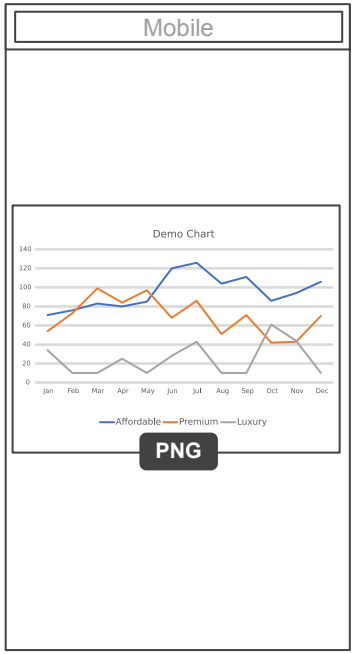
- Simple
- Quick to implement
- Works on any device

Drawbacks

- Text unreadable
- Features hard to see
- No improved interaction

P2: Using Viewport-Specific Images





P2: Using Viewport-Specific Images

- Display an image based on media queries
- Disregards concerns about readability, explorability, etc.

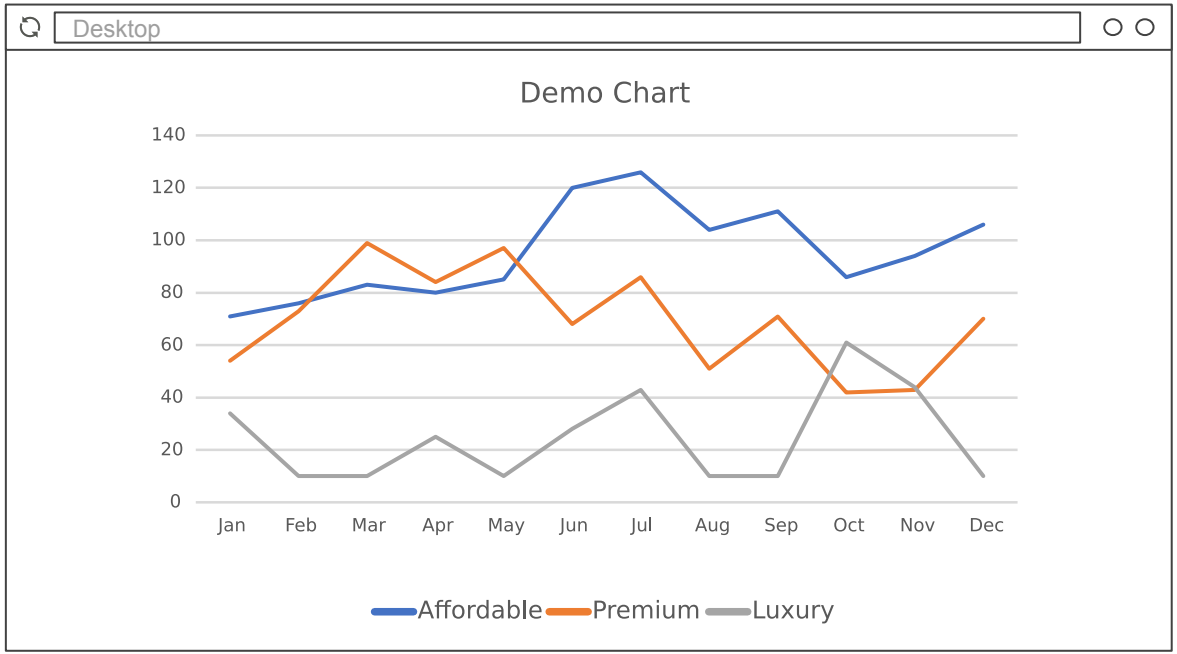
Advantages

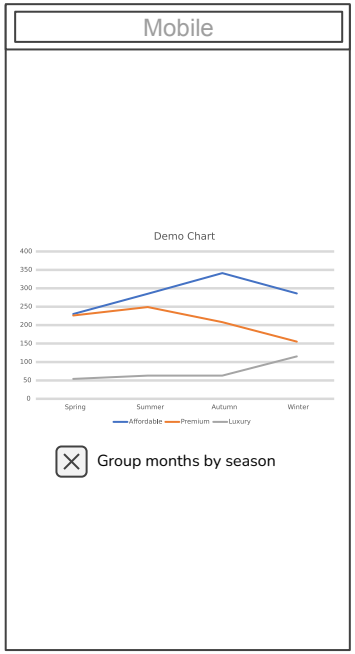
- Reduces time to “first contentful paint”
- Reduces computational complexity

Drawbacks

- No interaction
- Visualization has to be manually created for each supported viewport

P3: Data Generalization





P3: Data Generalization

- Eliminate or generalize data on purpose
- Aim for a reduction in granularity
- User input can be used to toggle the reduction

Advantages

- Easy reduction in visual complexity
- Moderately simple to implement

Drawbacks

- Generalization can obscure outliers and influence insights
- Manual process for some visualization
- Additional interactive features may be required

P4: Semantic Zooming

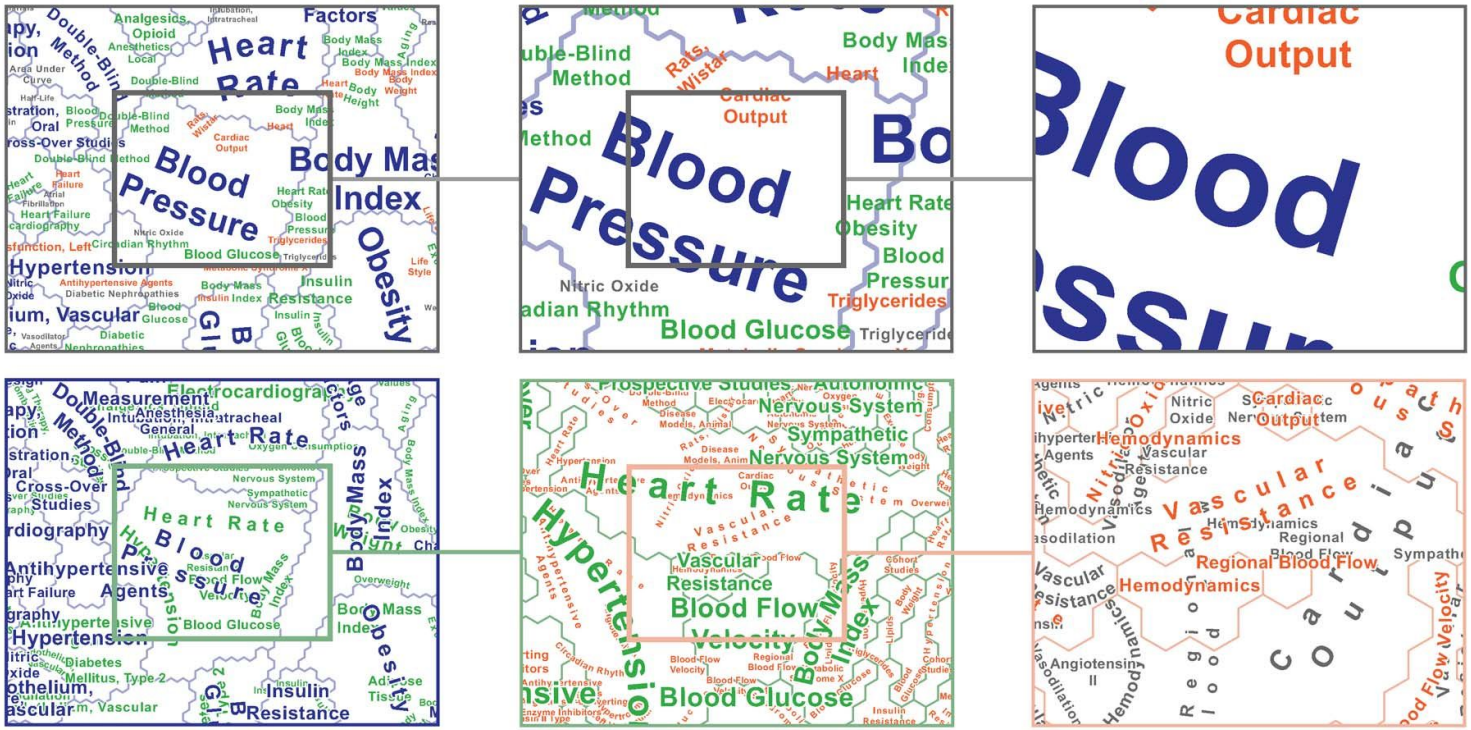


Image: Geometric zooming versus semantic zooming [Skupin, Biberstine, and Börner 2013]

P4: Semantic Zooming

- Provide different levels of detail according to zoom level
- Not exclusive to mobile visualizations
- Regularly used for desktop visualizations

Advantages

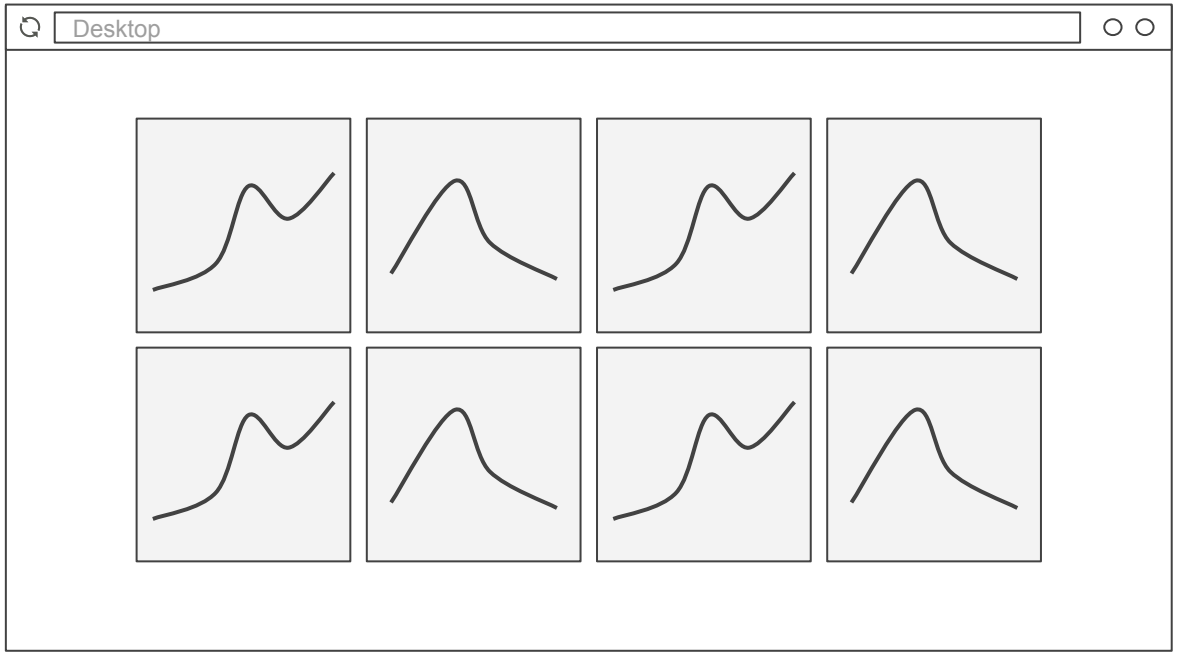
- Combines elimination and generalization with zooming
- Allows full data granularity
- Easy exploration on mobile (touch interaction widely accepted)

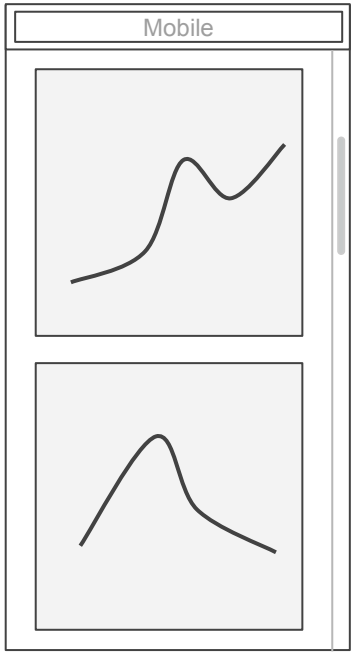
Drawbacks

- Works well for maps or graphs, tedious to replicate for other visualizations
- Time consuming implementation
- Loss of overview when zoomed in, requires additional techniques (focus + context, overview + detail, ...)

P5a: Viewport-Specific Layouting

Automatic





P5a: Viewport-Specific Layouting

Automatic

- Use vertical space instead of horizontal side-by-side visualizations
- Usually combined with other patterns
- Scrolling as an intuitive interaction

Advantages

- Simple to execute
- Entire page layout is updated, not just the visualization

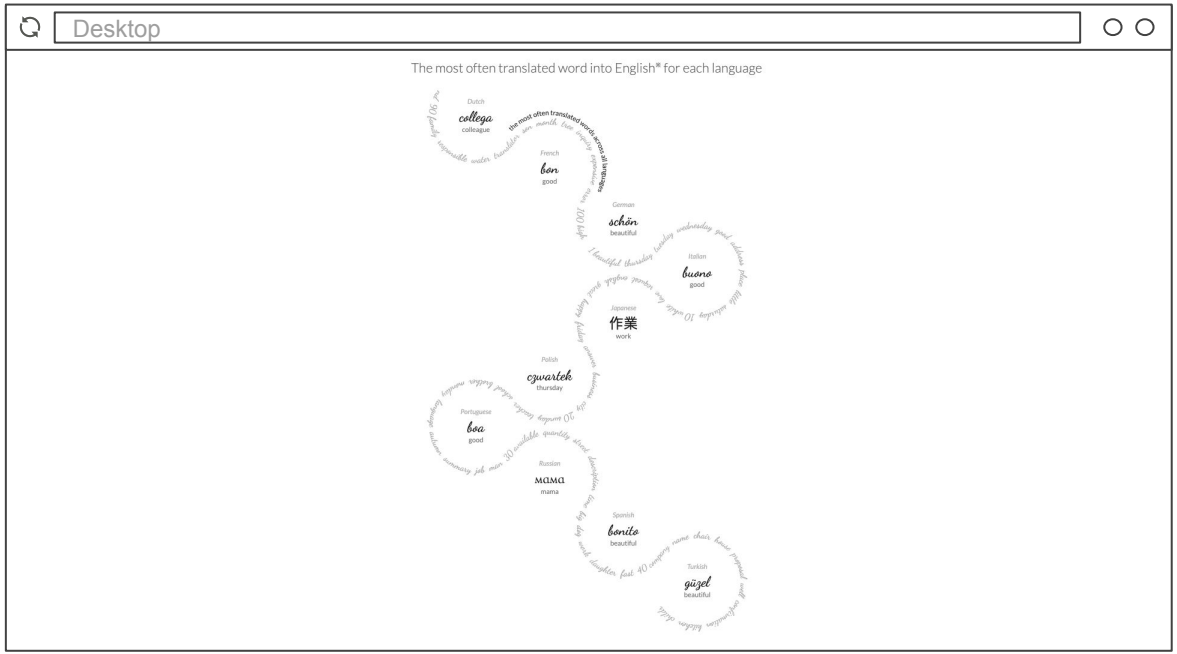
Drawbacks

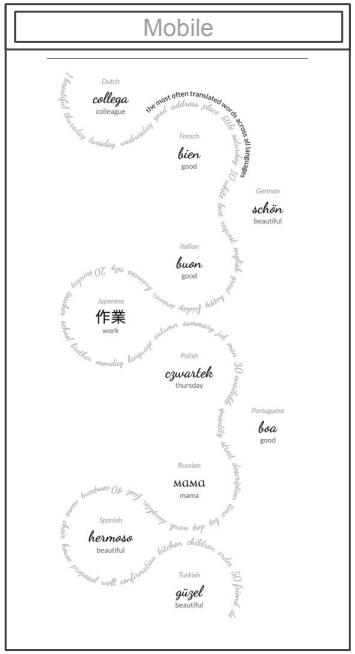
- Still needs other patterns for the visualizations
- Horizontal relationships between data are lost / must be reconstructed

P5b: Viewport-Specific Layouting

Manual

Example taken from: http://www.beautifulinenglish.com/





P5b: Viewport-Specific Layouting

Manual

- Use more of the available height through clever repositioning
- Display identical data in a different layout

Advantages

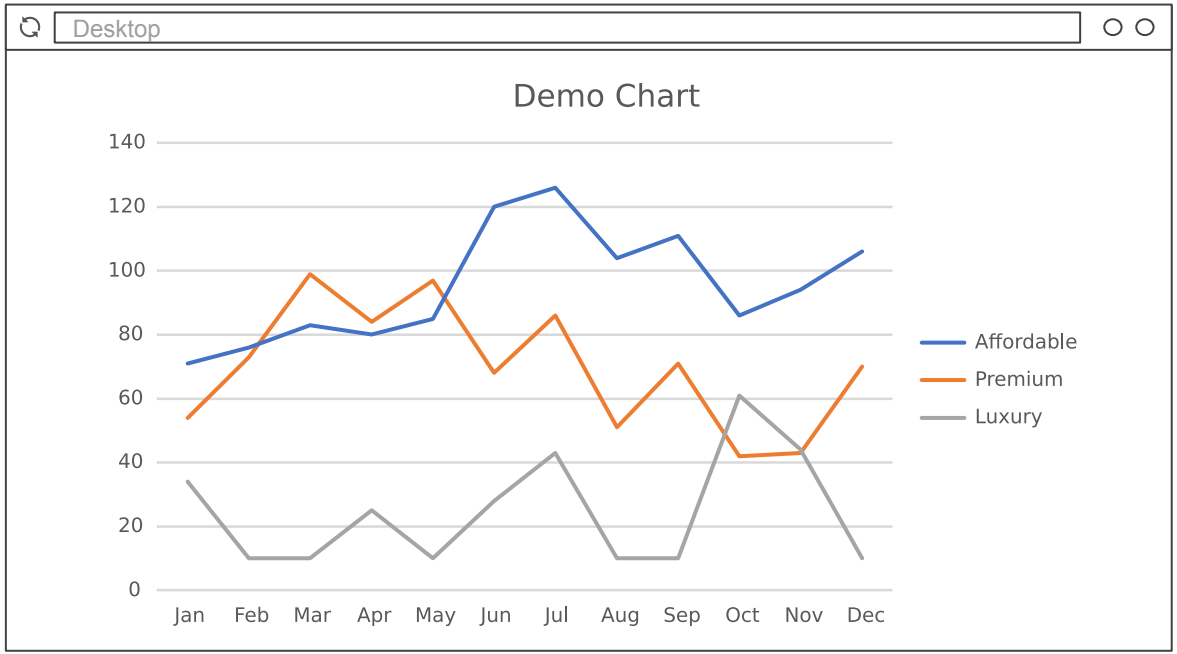
- Fully detailed visualization
- Identical data
- Provides additional insight compared to the desktop visualization

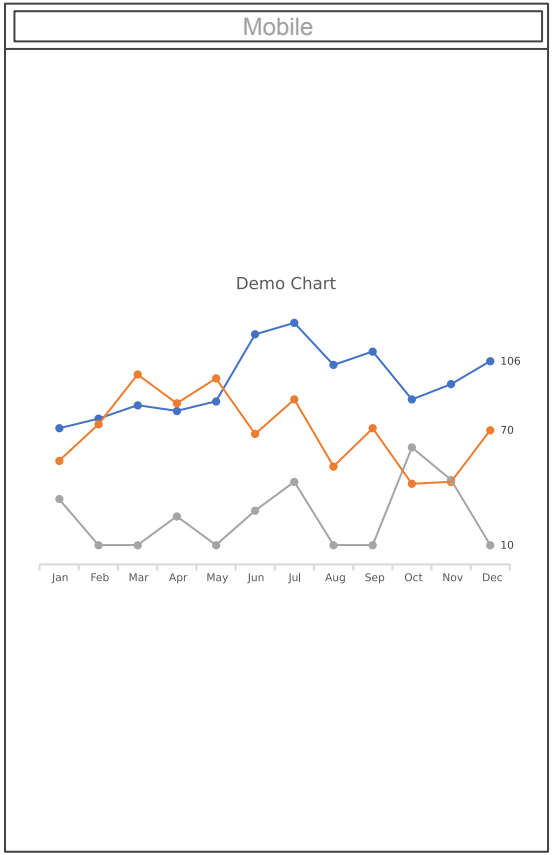
Drawbacks

- Manual process
- Hard and time consuming
- Must be tailored towards a few generic screen sizes

P6a: Changing Components

Axes, Tick Marks and Grid Lines





P6a: Changing Components

Axes, Tick Marks and Grid Lines

- Adapt chart components (axes, ticks, highlights, etc.)

Advantages

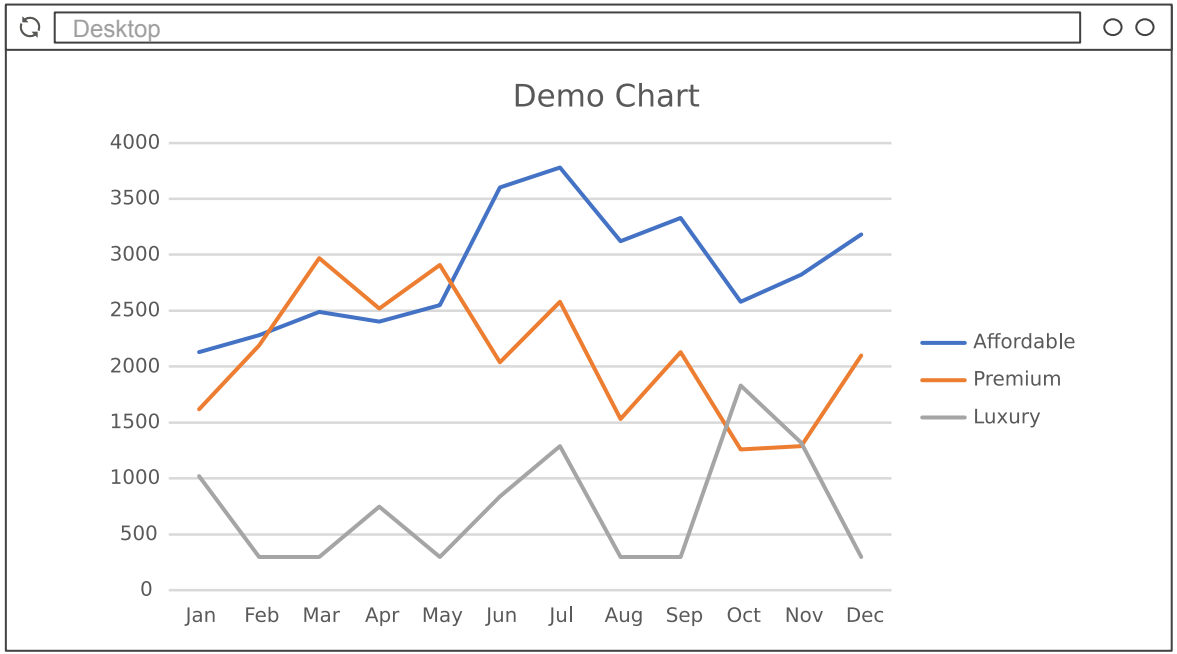
- Less clutter on the visualization
- Improved readability

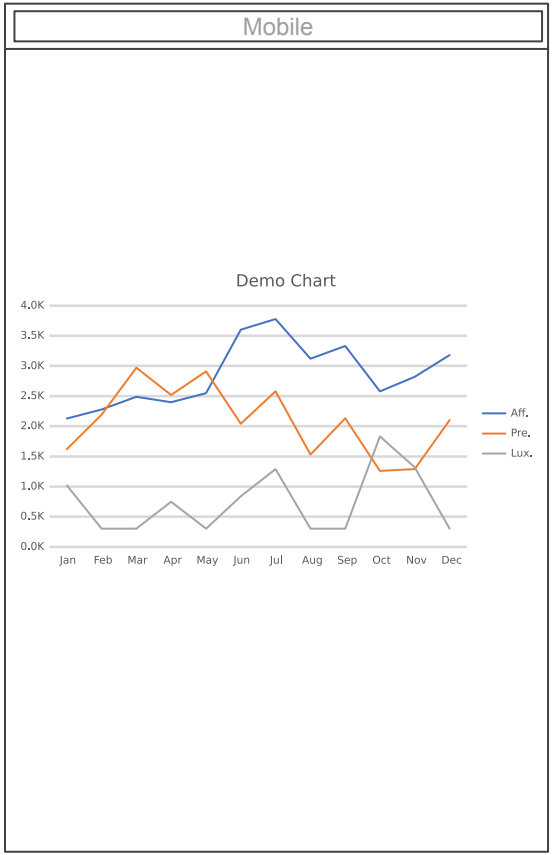
Drawbacks

- Users have to spend more time analysing the data structure
- Manual decision process for designers

P6b: Changing Components

Labels





P6b: Changing Components

Labels

- Abbreviate labels on axes, legends or tooltips
- Strategically change or remove less important annotations

Advantages

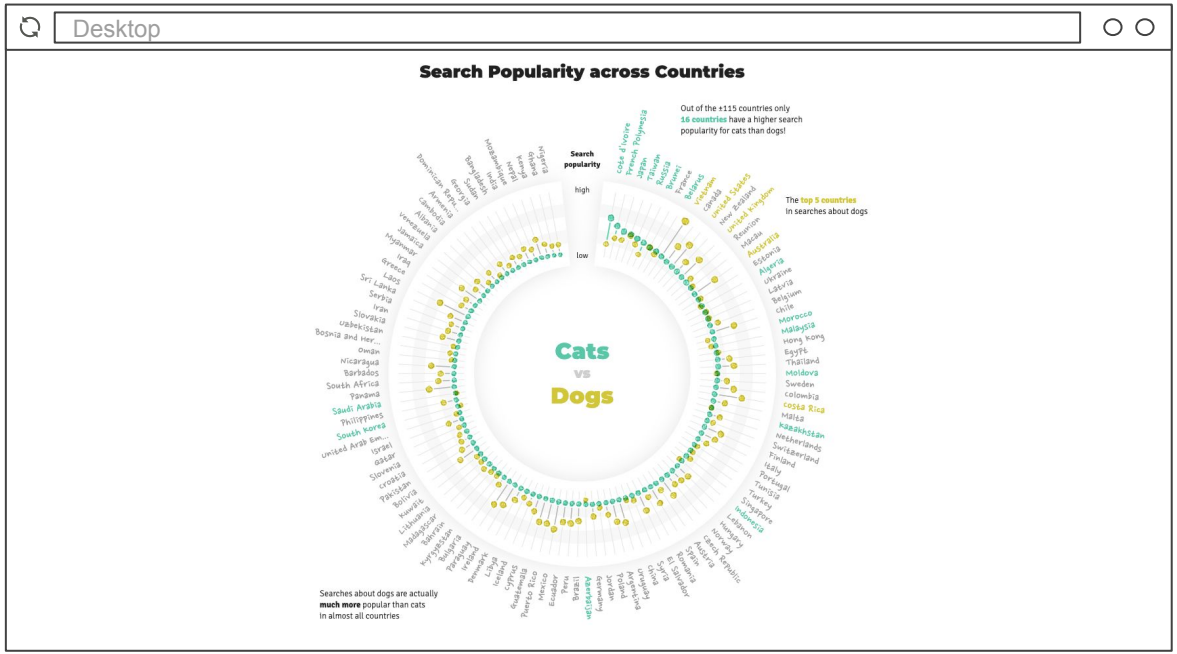
- Fully detailed visualization
- Less clutter on the visualization
- Easy implementation

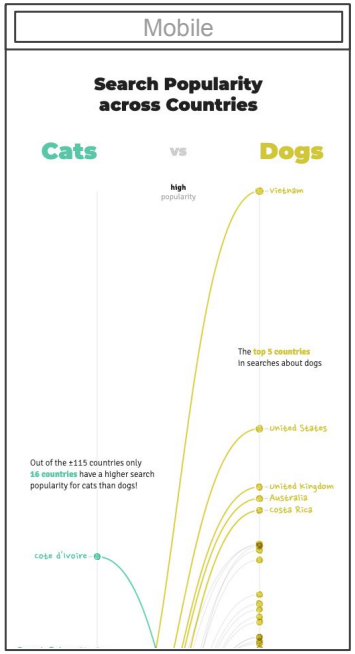
Drawbacks

- Some abbreviations should be listed and explained somewhere

P7: Using Different Charts

Example taken from: https://whydocatsanddogs.com/cats#chart-vs





P7: Using Different Charts

- Visualization with a different chart type
- Simple repositioning not enough

Advantages

- Fully detailed visualization
- Identical data
- Provides additional insight compared to the desktop visualization

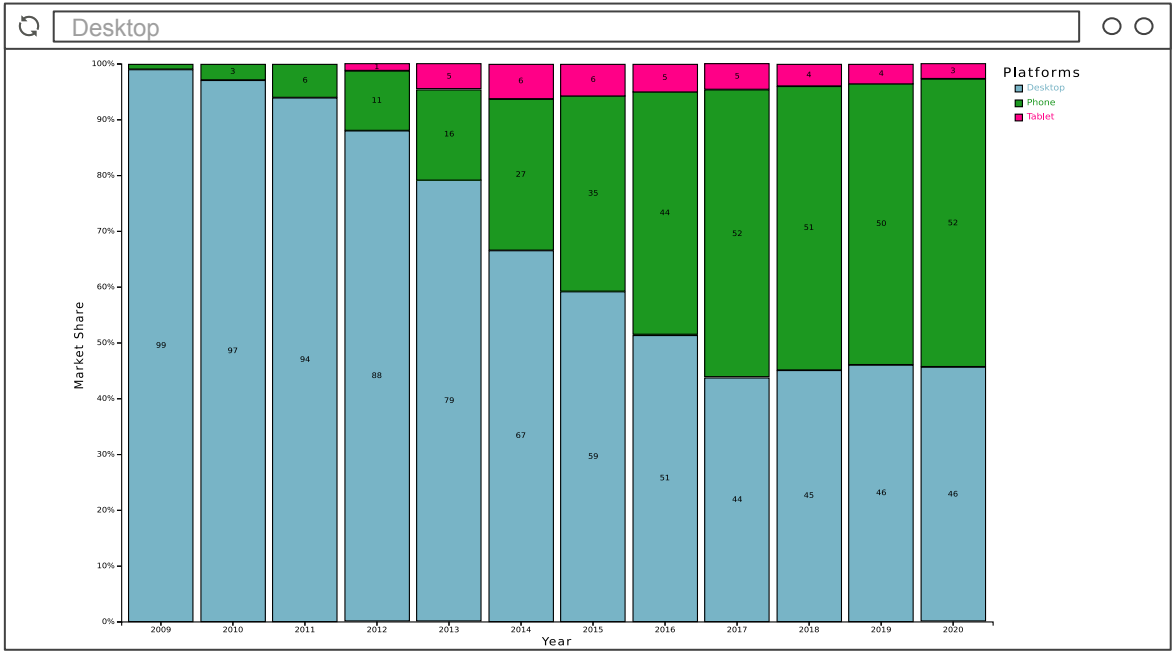
Drawbacks

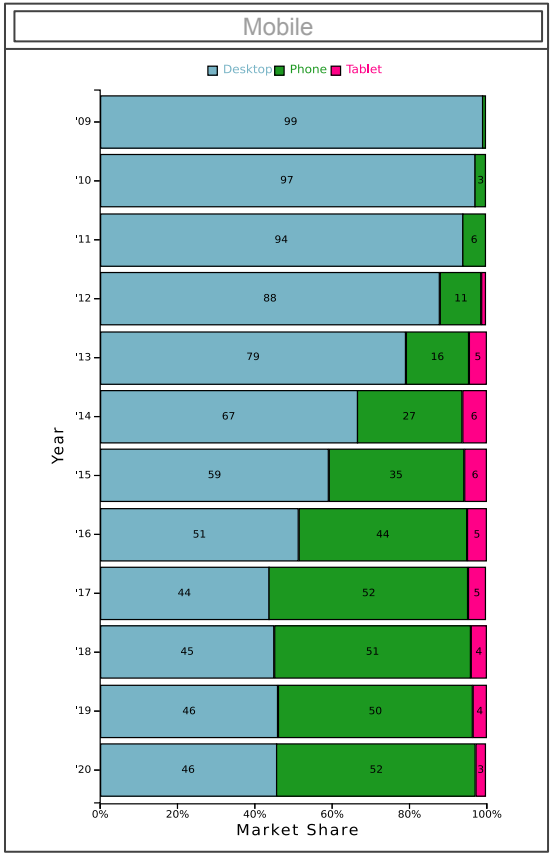
- Manual process
- Hard and time consuming
- Must be tailored towards a few generic screen sizes

Examples

Stacked Bar Chart: Device Market Share

Image taken from:
https://respvis.netlify.app/examples/stacked-bar.html

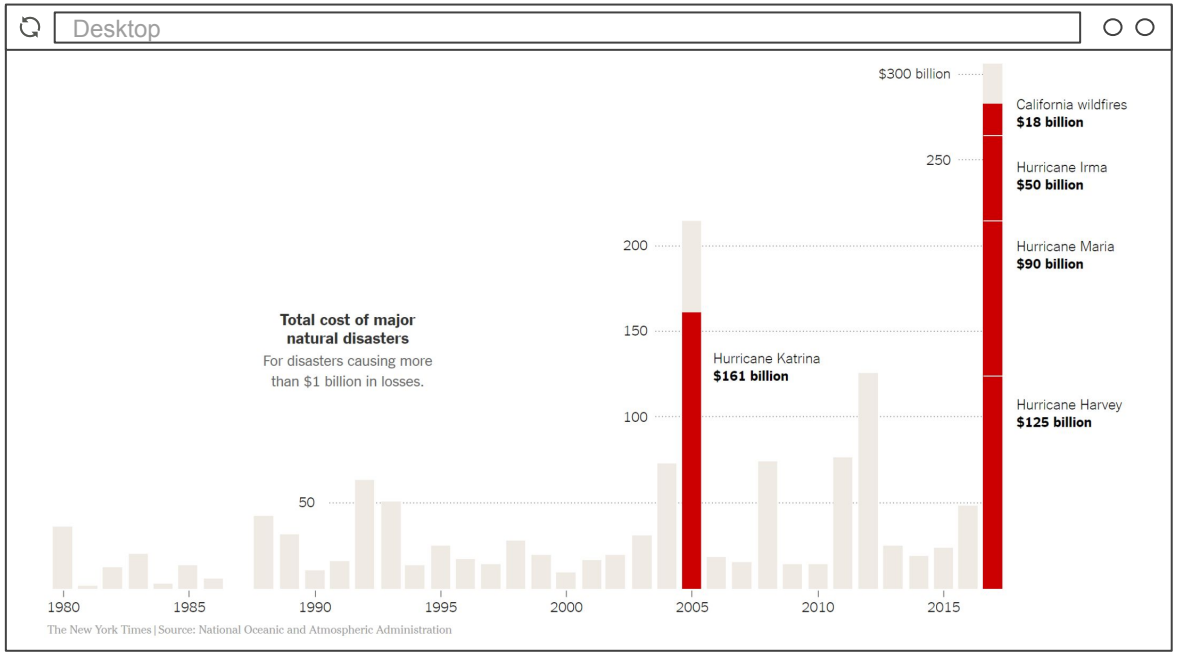


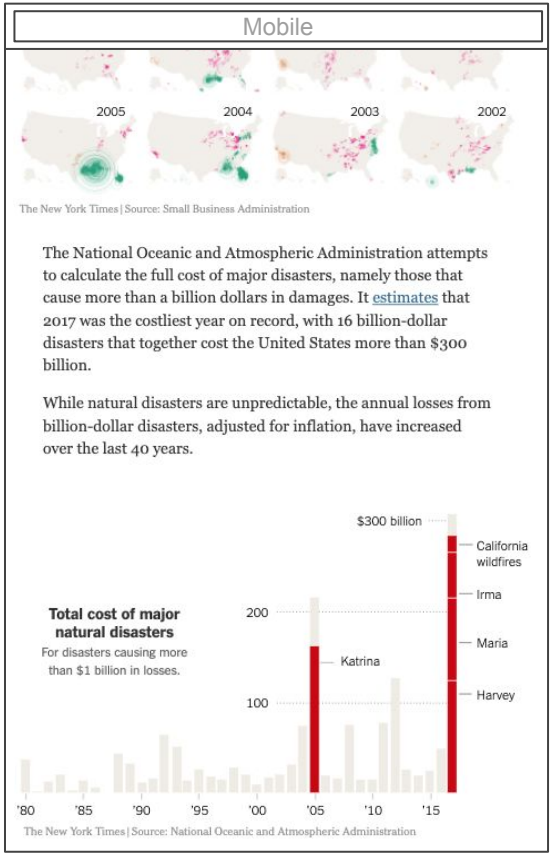


Demonstration video:
https://youtu.be/ybTWEkxkL98

Bar Chart: Cost of Disasters

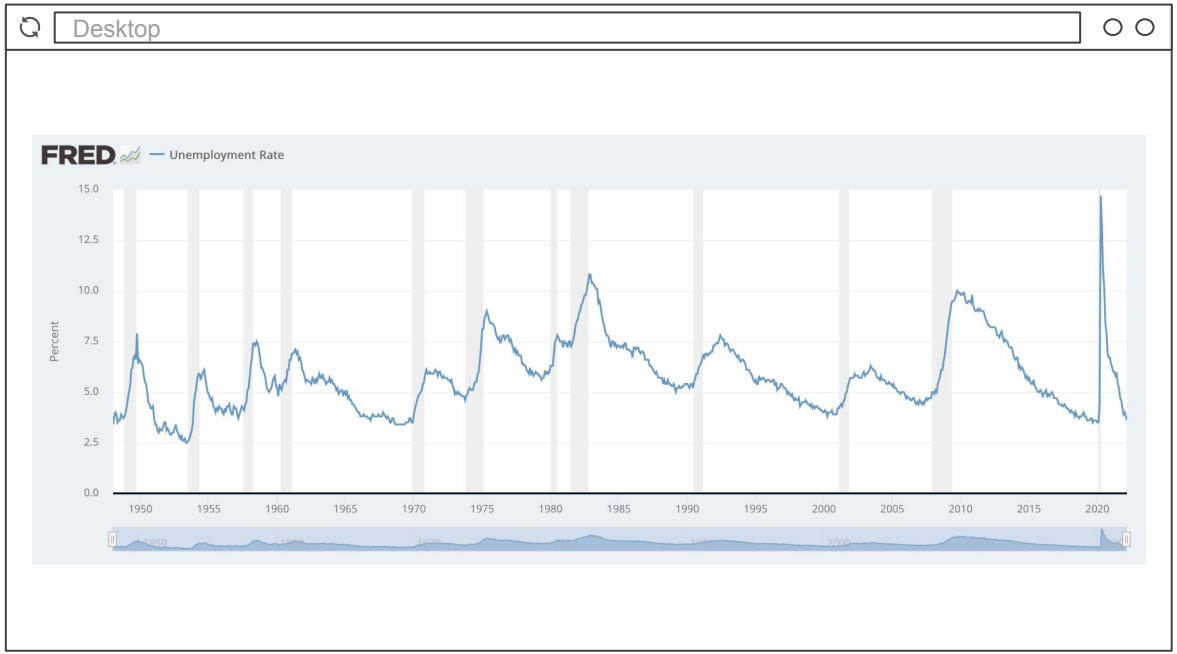
Image taken from: https://nyti.ms/2GJjoe4





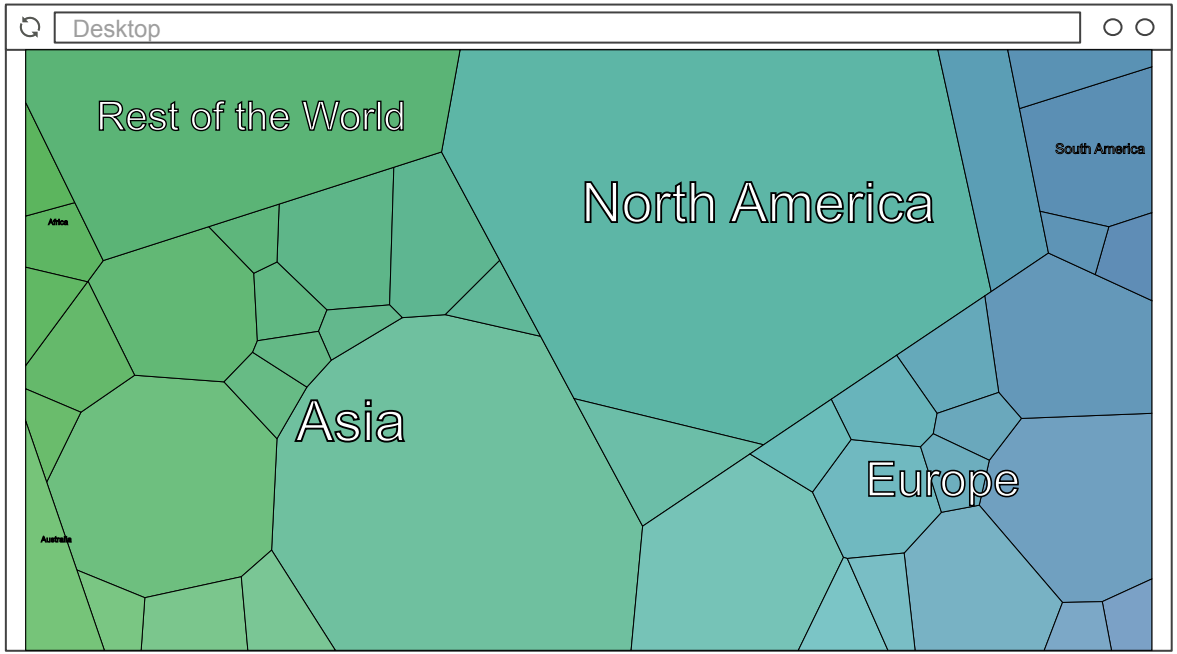
Line Chart: US Unemployment Rate

Image taken from: https://fred.stlouisfed.org/series/UNRATE



Voronoi Treemap: World GDP

Images taken from:
https://somestudentcoder.github.io/vorotree/



Demonstration video:
https://youtu.be/AR928NcAaeo

Tools

Visualization Design Tool by Hoffswell et al.

- Published within the 2020 CHI conference
- A visualization design tool
- Supports designing for multiple different target screen sizes
- Not available to the public

Demonstration video:
https://youtu.be/EdbpR29MQyA?t=18

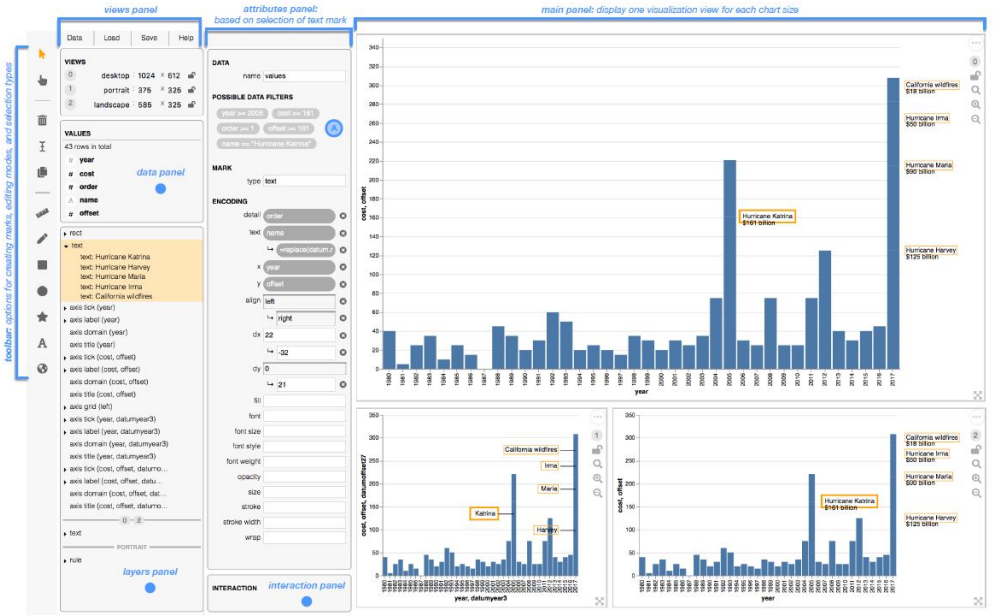


Image: Techniques for Flexible Responsive Visualization Design [Hoffswell et al. 2020]

MobileVisFixer

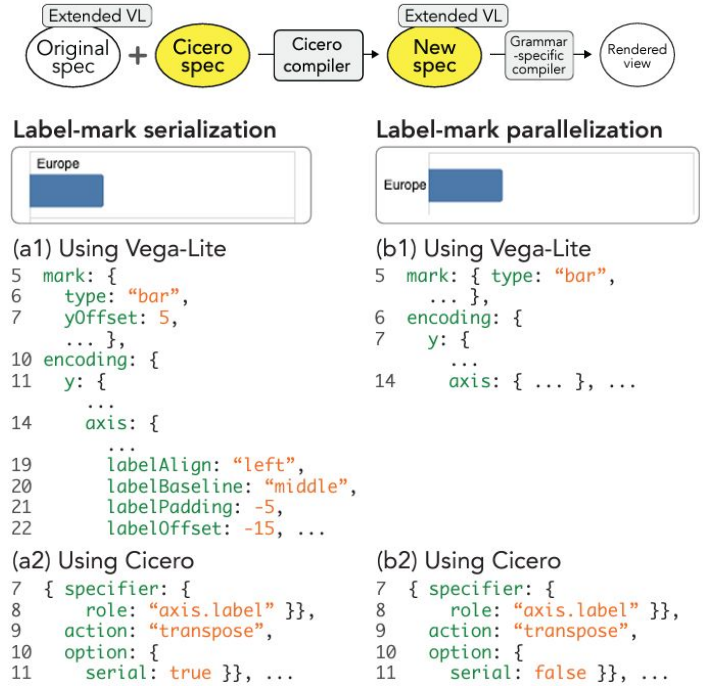
- Published in IEEE VIS 2020
- Automated method to transform svg-based graphs to a mobile friendly version
- Reinforcement Learning
- Not available to the public

Image: MobileVisFixer [Wu et al. 2021]

Demonstration video:
https://youtu.be/7vJkxLGdRBU?t=47

Cicero Compiler by Kim et al.

- Accepted to the 2022 CHI Conference
- Parses custom Cicero design specifications together with the original spec and turns them into an extended Vega-Lite specification
- Not available to the public



Images from: Cicero: A Declarative Grammar for Responsive Visualization [Kim et al. 2022]

Conference video:
https://youtu.be/2sawlX9PsBQ

Conclusion

Thanks for your attention!

References

Adler, Jahaj, Petritz and Yeli (2021). Responsive Data Visualisation. https://courses.isds.tugraz.at/ivis/surveys/ss2021/ivis-ss2021-g2-survey-resp-datavis.pdf

Andrews, K. (2018). Responsive Visualisation. *Data Visualization on Mobile Devices Workshop at CHI 2018 (MobileVis)*. https://mobilevis.github.io/assets/mobilevis2018_paper_4.pdf

Hoffswell, J., Li, W., & Liu, Z. (2020). Techniques for Flexible Responsive Visualization Design. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3313831.3376777

Kim, H., Rossi, R., Du, F., Koh, E., Guo, S., Hullman, J., & Hoffswell, J. (2022). Cicero: A Declarative Grammar for Responsive Visualization. *ArXiv:2203.08314 [Cs]*. https://doi.org/10.1145/3491102.3517455

Kim, H., Moritz, D., & Hullman, J. (2021). Design Patterns and Trade-Offs in Responsive Visualization for Communication. *Computer Graphics Forum*, *40*(3), 459–470. https://doi.org/10.1111/cgf.14321

Kim, Y.-H., Lee, B., Srinivasan, A., & Choe, E. K. (2021). Data@Hand: Fostering Visual Exploration of Personal Data on Smartphones Leveraging Speech and Touch Interaction. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–17. https://doi.org/10.1145/3411764.3445421

Skupin, A., Biberstine, J. R., & Börner, K. (2013). Visualizing the Topical Structure of the Medical Sciences: A Self-Organizing Map Approach. *PLoS ONE*, *8*(3), e58779. https://doi.org/10.1371/journal.pone.0058779

Wu, A., Tong, W., Dwyer, T., Lee, B., Isenberg, P., & Qu, H. (2021). MobileVisFixer: Tailoring Web Visualizations for Mobile Phones Leveraging an Explainable Reinforcement Learning Framework. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 464–474. https://doi.org/10.1109/TVCG.2020.3030423