

CS 171 - Visualization

CrashCaster

Staying Safe in Cambridge

Process book

Kyle Maguire

Rick Farmer

Kartik Trasi

Adam DiChiara

Contents

1. Project Proposal
2. Project Plan
3. Final Project Plan
4. Project Redesign
5. Project Prototype V1
6. Project Prototype V2
7. Final Project V2

Chapter 1: Project Proposal

Week 1 (3/21-3/27)

Overview and Motivation

// Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.

Public safety is an important concern for residents and commuters in the Boston metro area, but the information they rely on to avoid safety risks may be lacking. In particular, traffic safety issues are common and frequent, yet with the exception of severe accidents the information the general public has regarding traffic safety is mostly based on personal judgements from anecdotal or personal experiences. In general, any information provided fails to inform the public in a way that represents systemic problems or complex inter-related traffic dynamics at play. Our project aims to develop insightful visualizations representing public traffic safety data in a way which reveals probable "cause and effect" relationships between influencing factors and incident occurrences. Our goal is to employ good data visualization design principles and effective storytelling to create an end product that is both informative and visually appealing.

We plan on exploring datasets on city traffic data provided by the cities of Cambridge and/or Boston, MA. Our identified datasets include geographic location, incident type, date, time of day, and other related details which allow filtering, faceting, and sorting with various methods. This data can also be merged with other available datasets on traffic density and potentially weather. Additionally, we plan to classify and represent geo-areas into safe and potentially unsafe zones based on risk factors, which would give residents a guide to reroute their travel via safer paths. We were attracted to these datasets due to their completeness and readiness for analysis, which will allow us to focus our efforts on developing effective visualizations. Also, the cities of Cambridge and Boston provide a significant number of other related datasets with information that could potentially enrich our visualizations in meaningful ways. We think there's a lot to be explored in the data.

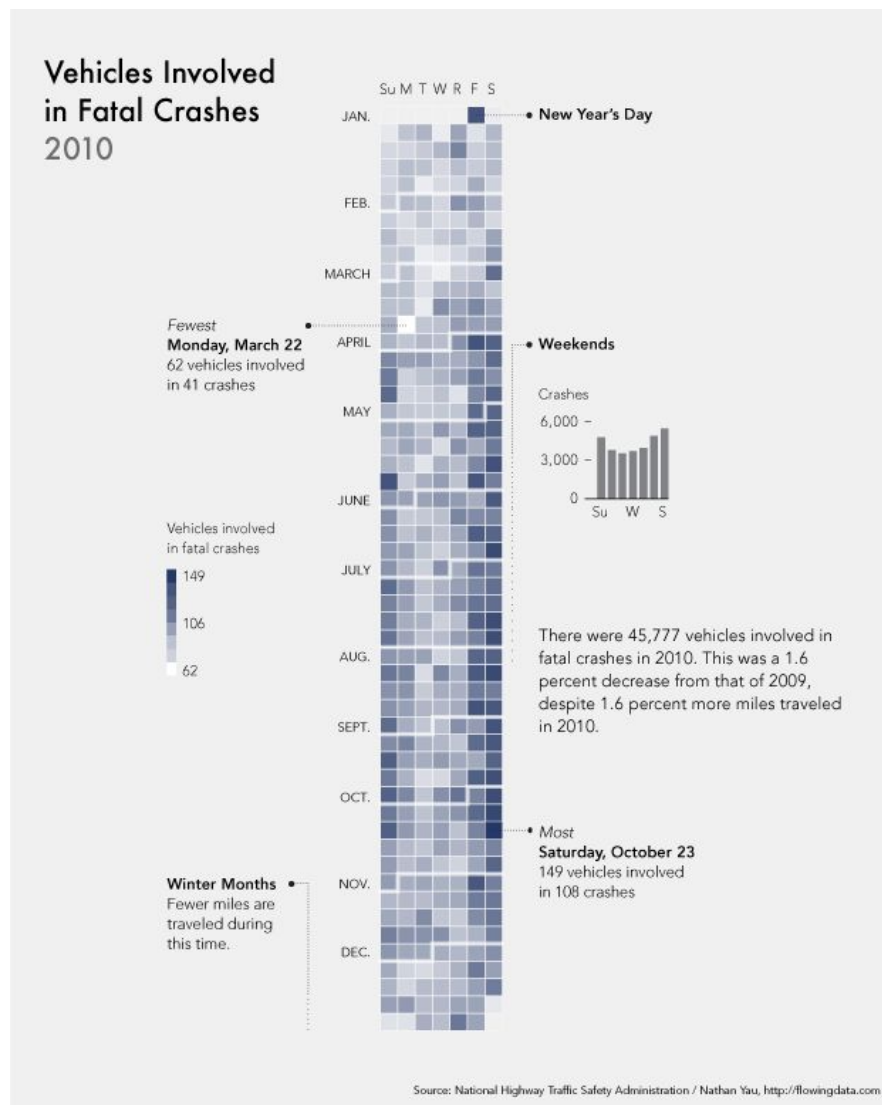
Related Work

// Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.

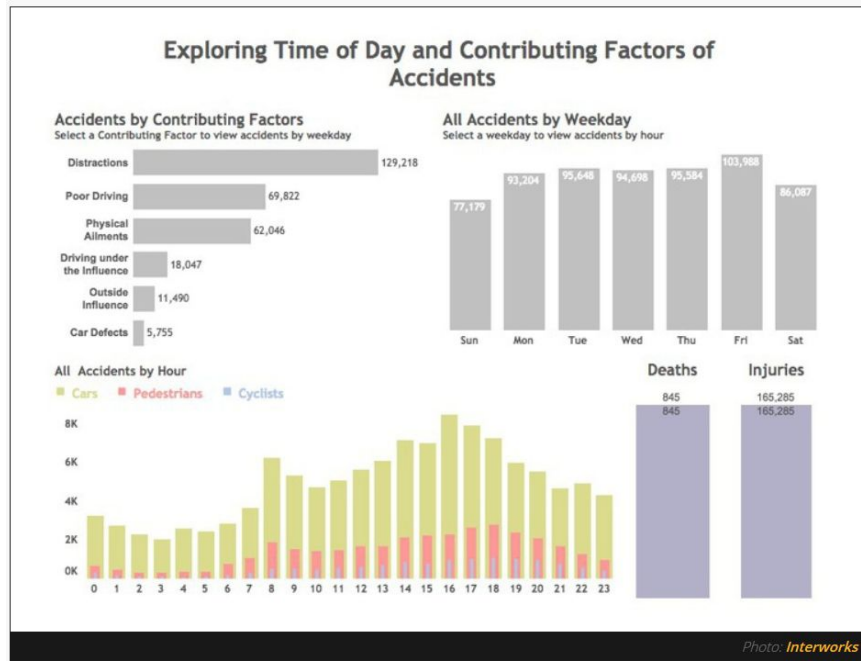
Our team identified several sources of inspiration based on other crash/traffic studies, as well as various other topics.

This static visualization inspired the calendar heatmap we used.

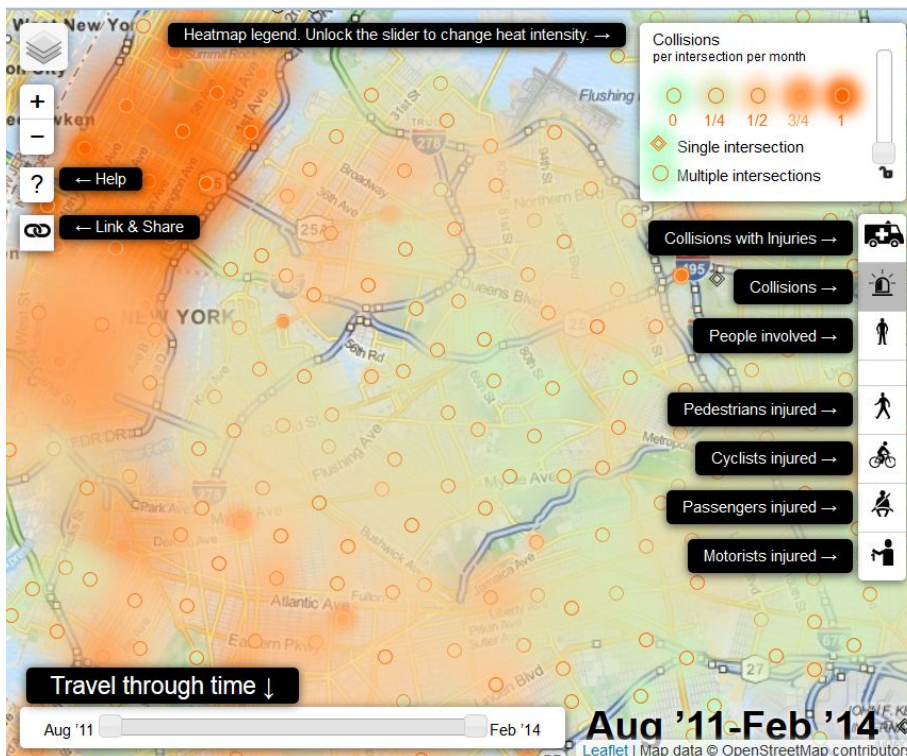
<http://www.coolinfographics.com/blog/2012/1/11/calendar-visualization-of-fatal-car-crashes.html>



<https://viewing.nyc/interworks-exploration-and-analysis-of-new-york-city-vehicle-crash-data/>



<http://nyc.crashmapper.com>

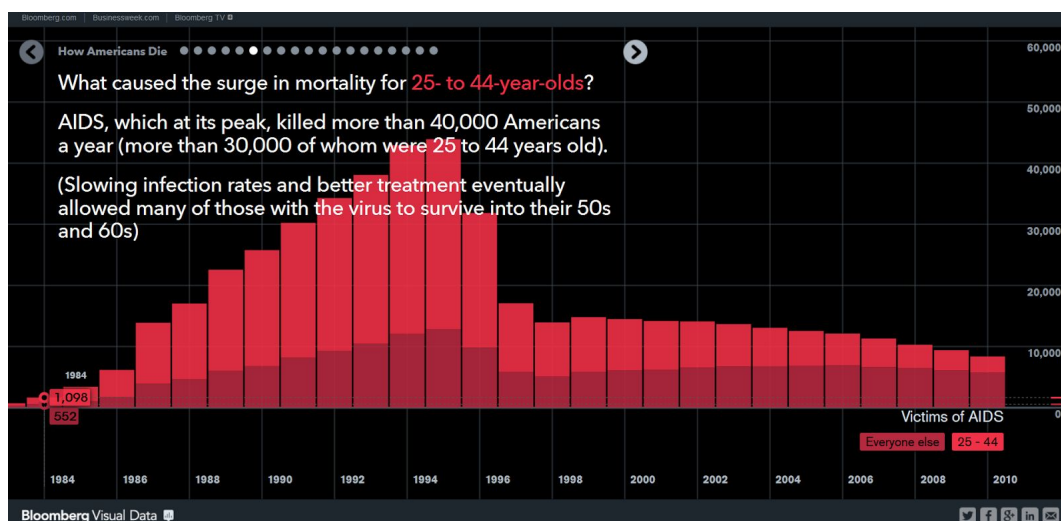


Some youtube videos of biking in cambridge showed us just how dangerous the city can be for cyclists, as well as how much bike riders can be at fault for accidents due to their aggressive riding.



This example from bloomberg visual data did not have much to do with our project topic or specific visualizations, but we loved the effectiveness of it for drawing insights about the data. In particular, how outside research was used for explaining results in the data. We used this approach in our storytelling, by researching the interesting/significant findings in our data and bringing in outside information to add context and background.

<http://www.bloomberg.com/graphics/dataview/how-americans-die/>



Boston crime statistics have been evaluated by *likeustic* which is a based on a similar dataset which we plan to evaluate. However, we plan to focus on the traffic statistics.

https://github.com/likekeustc/Boston_Crime_Data/blob/master/crime_markdown.md

likekeustc / Boston_Crime_Data

Watch 1

Star 0

Fork 1

Code

Issues 0

Pull requests 0

Pulse

Graphs

Branch: master

Boston_Crime_Data / crime_markdown.md

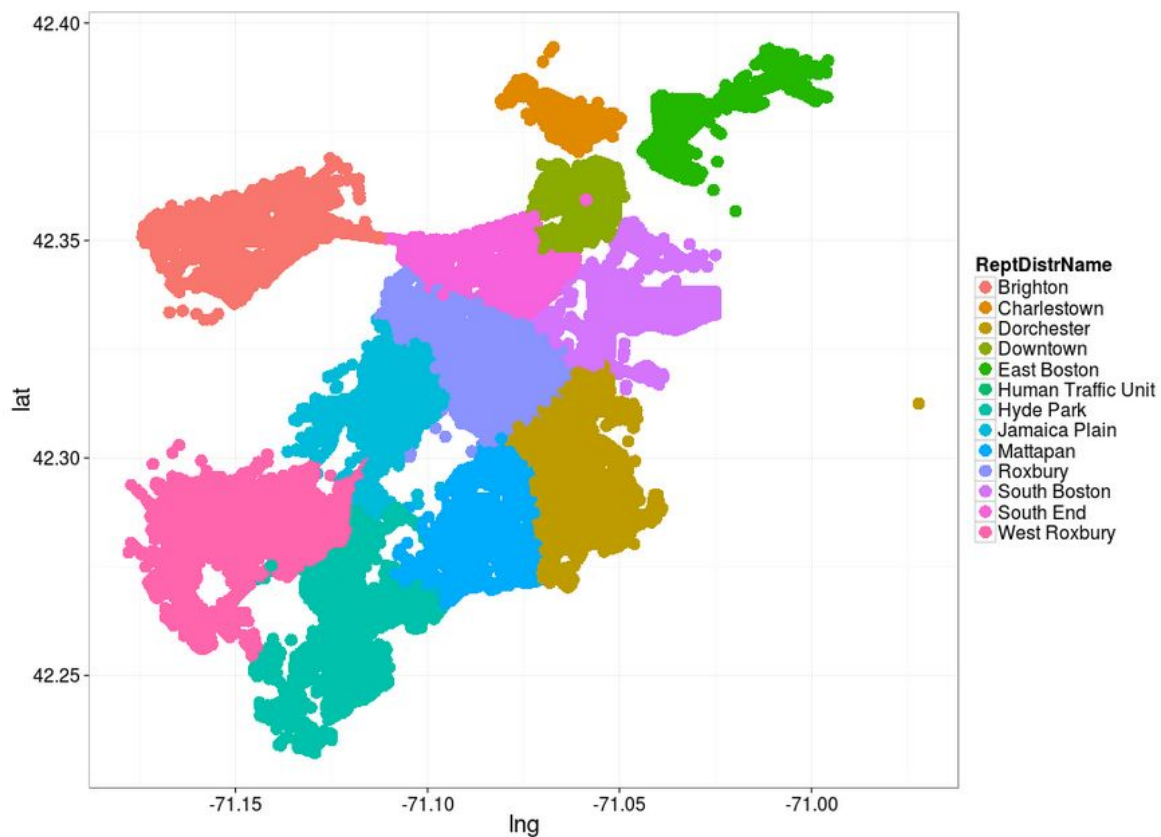
Find file

Copy path

likekeustc cache

31c240e on Feb 17, 2015

1 contributor



Questions

// What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?

1. Are there systemic problems in Boston that can be addressed?
2. Are there complex inter-related traffic dynamics at play?
3. Do we see any "cause and effect" patterns between influencing factors and incident occurrences?
4. Are there safe and unsafe areas that the residents should be aware of?

Data Variable / Source Outline

// Source, scraping method, cleanup, etc.

Complete Dataset

We are trying to answer the following questions at a high level with our vizualization project:

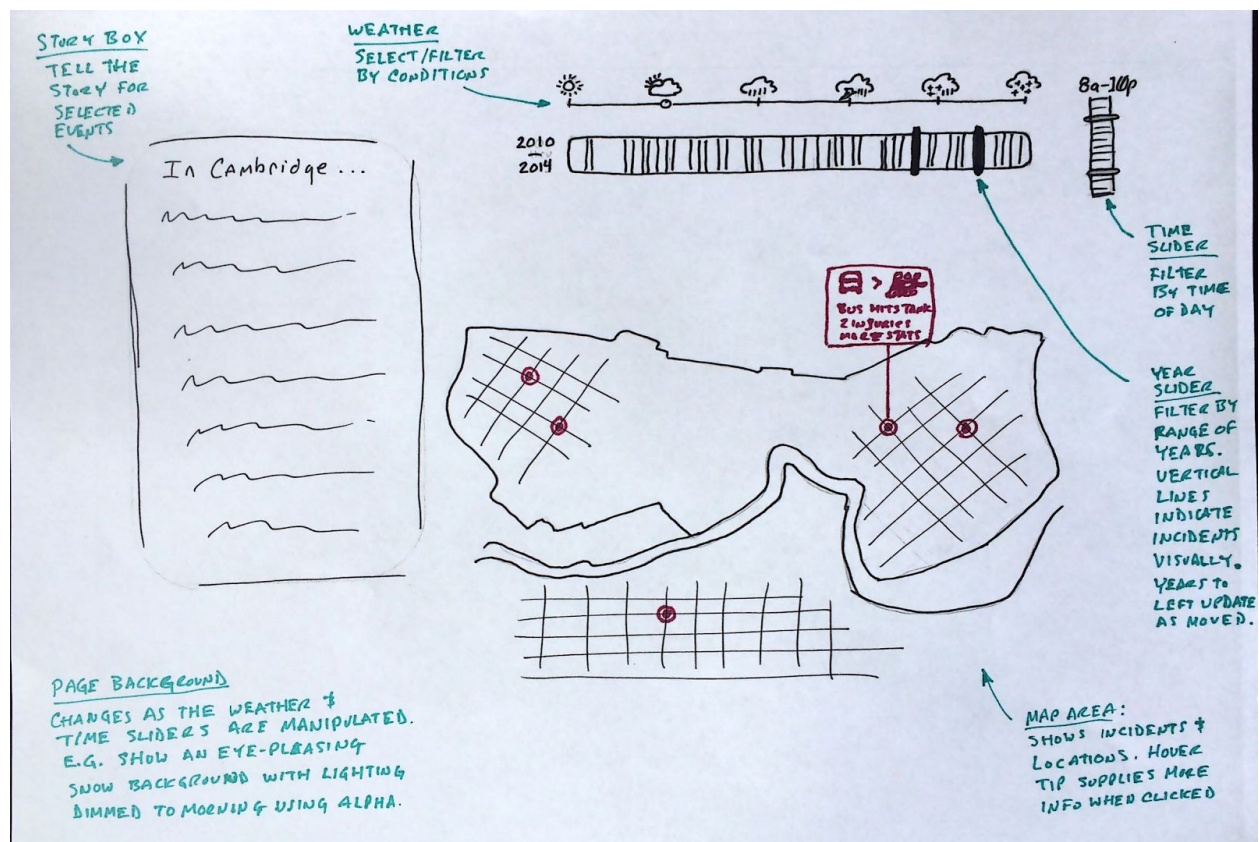
- Where are the most dangerous intersections/road locations?
- Where are most police citations for dangerous driving issued?
- How are traffic accidents/citations affected by various factors (time/weather/traffic-volume/road safety-features)?
- Measurable Outcomes – “Effects” (Dependent Variables)
 - Traffic Accidents
 - Location:
 - Coordinates (for mapping)
 - Street / Cross-street info (for referencing)
 - Filterable by accident type (auto/auto, auto/pedestrian, etc)
 - Sources
 - 2010-2013:
<https://data.cambridgema.gov/Public-Safety/ACCIDENT-2010-2013/ybny-g9cv>

- 2014:
 - <https://data.cambridgema.gov/Public-Safety/ACCIDENT-2014/7fai-h9wk>
 - Visualization ideas:
 - Plot locations of accidents on map.
 - Color code by type?
 - Filterable by type and cross-linked factors?
- Police Citations
 - Location:
 - Coordinates (for mapping)
 - Street / Cross-street info (for referencing)
 - Filterable by citation type (Speeding, run light, drunk driving, etc.) (some non-relevant types (e.g., revoked license, no insurance) can be filtered out of dataset.
 - Sources: 2010-2014
 - <https://data.cambridgema.gov/Public-Safety/Police-Citations-2010-2014/gmq6-8ver>
 - Visualization ideas:
 - Plot locations of citations?
 - Heat-map of citations (Overlaid with accident data as points)?
- Measurable Factors – “Causes” (Independent Variables)
 - Date/Time of Accident/Citation
 - Year (2010 – 2014)
 - Potential factor: Increase in accidents due to population growth, commuter rise.
 - Time of Year (Date/time string, parseable)
 - Potential factor: More incidents during school year, etc. More during winter.
 - Day of Week (Day value present)
 - Potential factor: More commuter accidents during work week. More nighttime accidents on Friday/Saturday.
 - Time of Day ((Date/time string, parseable)
 - Potential factor: More accidents during rush hour.
 - Visualization ideas:

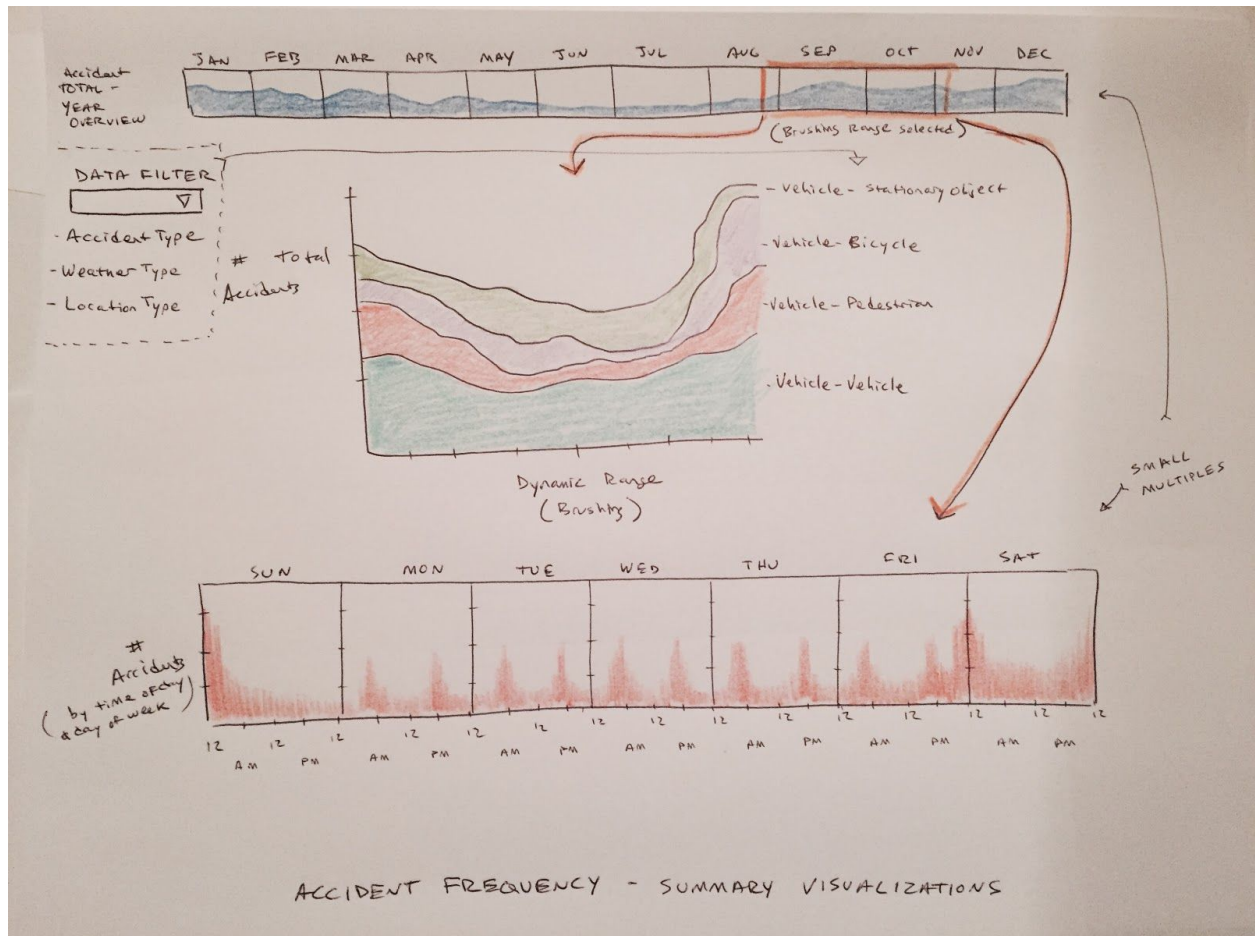
- Filter mapped data by timeline brushing?
 - Summary charts showing distribution of total accidents/citations by date/time
- Weather
 - *Weather condition*
 - Potential factor: More accidents during adverse weather conditions
 - *Daily Temperature?*
 - Potential factor: More road accidents during freezing temperatures. More pedestrian/bike accidents during warmer temperatures.
 - Visualization ideas:
 - Filter mapped data by selectable weather condition icons / filter options?
 - Filter mapped data by brush-able temperature range?
- Traffic Volume
 - Data for certain intersections in Cambridge showing average & peak daily traffic counts.
 - Potential factor: Higher traffic volume = higher accident rate
 - Mapped by location (Same as above)
 - *Source: 1972-2014 (can be filtered to our applicable range)*
 - <https://data.cambridgema.gov/Traffic-Parking-and-Transportation/Average-Daily-Traffic-Counts-1972-to-2014/v43b-kqeq>
 - Visualization ideas:
 - Summary chart(s) showing distribution of volume levels by date/time?
- Presence/Non-Presence of Traffic Signal
 - GeoJSON locations of traffic lights.
 - Potential factor: Safer intersections, less accidents.
 - *Source:*
https://github.com/codeforboston/open_data_cambridge/blob/master/Traffic/TrafficLights.geojson
 - Visualization ideas:
 - Enable/Disable viewable layer showing locations of traffic signals?
- Presence/Non-Presence of Walk Signal
 - GeoJSON locations of pedestrian crosswalk signals.

- Potential factor: Safer road crossings for pedestrians, less pedestrian accidents.
 - *Source:*
https://github.com/codeforboston/open_data_cambridge/blob/master/Traffic/WalkSignals.geojson
 - Visualization ideas:
 - Enable/Disable viewable layer showing locations of walk signals?
- Presence/Non-Presence of Crosswalk
 - GeoJSON layer of marked crosswalks on roads.
 - Potential factor: Safer intersections for pedestrians
 - Source:*
https://raw.githubusercontent.com/codeforboston/open_data_cambridge/master/Traffic/PavementMarkings.geojson
 - Visualization ideas:
 - Enable/Disable viewable layer showing locations of bike routes on roads?
- Presence/Non-Presence of Bike Route
 - GeoJSON layer of marked bike routes on roads.
 - Potential factor: Safer roads for bike riders
 - *Source:*
<https://data.cambridgema.gov/Geographic-Information-GIS-/Bike-Facilities/tdg4-6twm>
 - https://github.com/codeforboston/open_data_cambridge/blob/master/Recreation/BikeFacilities.geojson
 - Visualization ideas:
 - Enable/Disable viewable layer showing locations of bike routes on roads?
- Supplemental Data
 - GIS data layers (Basemap, Roads, Sidewalks, Buildings, place names, locations of interest)

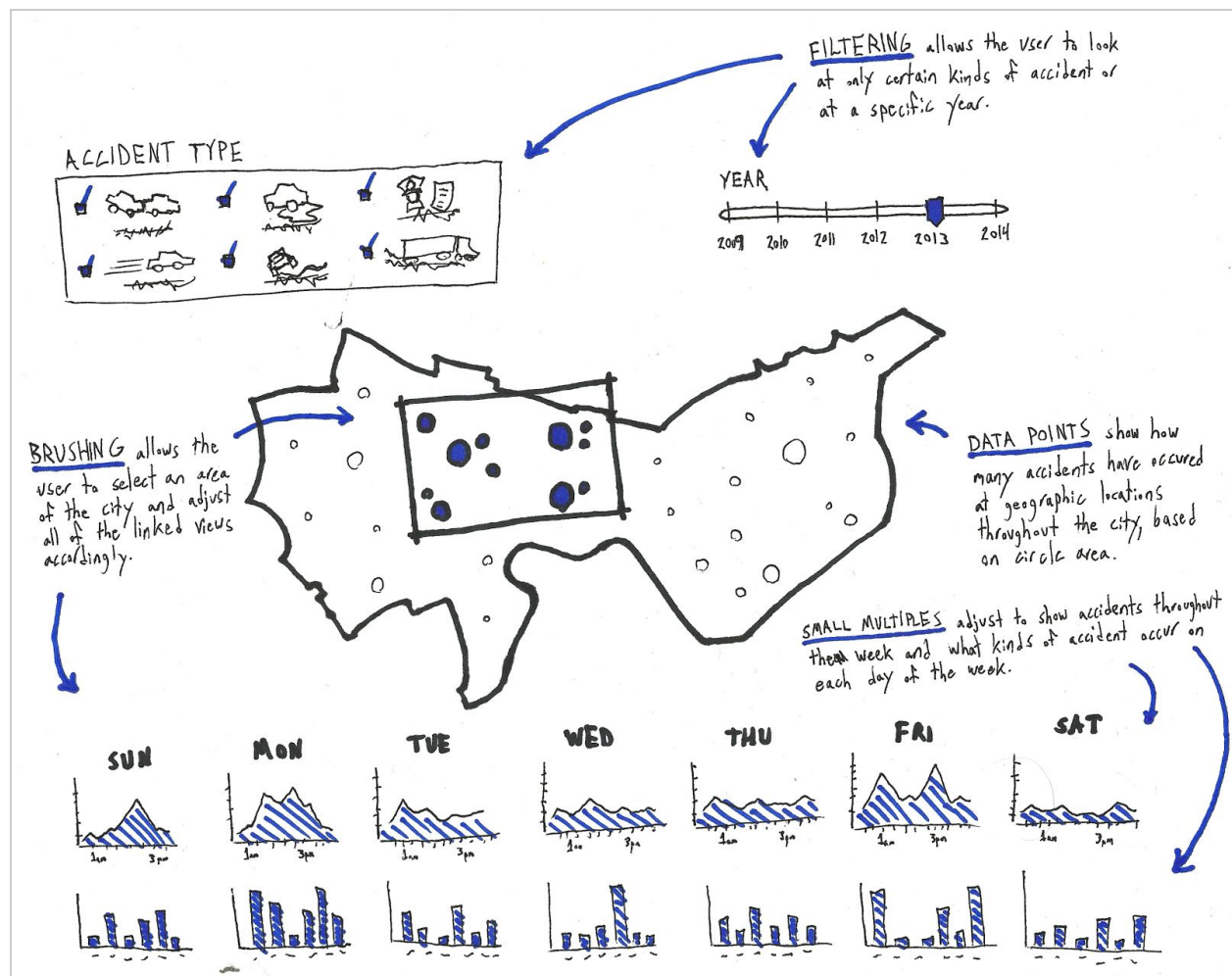
// What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?



See full-sized version at <http://harvardviz.github.io/traffic/docs/traffic-sketch-1-rf.png>



See full-sized version at <http://harvardviz.github.io/traffic/docs/sketch2-ad.JPG>.



See full-sized version at <http://harvardviz.github.io/traffic/docs/traffic-sketch-3-km.png>.

Implementation

// Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

Evaluation

// What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?

Chapter 2: Project Plan

Project Timeline and Milestones:

Milestone	Submit Project Proposal	Final Project Plan	Project re-design	Project prototype V1	Project prototype V2	Final project V2	Demo
Progress							
Deadline	3/27	4/3	4/10	4/17	4/24	5/1	5/8

Goals and Tasks

Our preliminary goals and tasks for this project are based on visualization process steps outlined in class. These goals and tasks will be refined as we make progress on the project.

Screencast

A screencast presentation of the Initial Project Plan can be found at <https://youtu.be/L33ZCXyVEVo>.

1. Target

a) Choose domain. We've generally decided upon studying traffic data from Cambridge, MA. We must decide what aspect(s) we will specifically examine. Some initial ideas are traffic density, traffic crimes, and street layouts.

b) Define question. We need to decide what questions we're trying to answer using the datasets we have access to. There are the obvious high level questions like "Where and when is traffic the worst / best?" and "Where are you most likely to get a ticket?". There are also the more interesting questions that involve correlation of multiple datasets like "How is traffic density related to traffic crime?" or even more interestingly "How do traffic crime and safety relate to poverty levels?". These are the deeper questions we need to better define moving forward.

c) Explore existing solutions. We will look at existing visualizations and data analysis that relate to our subject. Luckily, traffic data is a common subject of analysis. Additionally, any effective display of location-based data could be relevant to our project. We will compile our findings to be used as reference when we decide how we want to present our data.

d) Formulate data analysis tasks. Using the questions we've decided we want to answer, we must define how we will go about analyzing the data. This essentially means taking those questions and making specific decisions about what data will be involved and how to actually go about finding an answer.

2. Data Wrangling

a) Find and clean data. Once we know what questions we want to answer, we will have a better idea of what format we need the data in and what parts of the data are important to our project. We need to define the high-level concept of how we want to manipulate and format the data, then actually implement code to export the data in this format.

b) EDA (exploratory data analysis). This is the first step in drawing conclusions from our datasets. We will need to do some more data manipulation and try to find very basic answers to some of our questions. This might mean computing some simple averages or counts for data points in a given time period or physical area. This will be our first look into what the data can tell us and will set the stage for all of our analysis and presentation of the data moving forward.

c) Transform and summarize data. Having explored what the data can tell us, we will transform the data into a format that gets even closer to the final data we will use for creating our visualizations. After making a decision on the ideal format(s), we will once again implement code to export the data in this format.

3. Design

a) Design visual encoding. Having decided upon interesting questions and formatted the data, we will decide the best way to present that data in an intuitive and understandable way. We will likely reference resources like Bertin's Taxonomy and the existing solutions that we explored for guidance on what will best serve our needs.

b) Design interactions. Going a step further, we will expand the visual encoding design with useful interactions that can provide even more insights into the data. Our data will probably lend itself well to interactions as it generally spans both time and geographic location, thus it has many levels of depth (high-dimensional data). We will drive our visual interactions using the “Modalities for Interaction”, User Intent, and the Heer and Shneiderman taxonomy for interaction.

c) Design layout and storytelling. We now must take all of the analytical work and frame it in a design layout to drive a compelling story. We will use Gestalt principles not only in our visualizations but in our design layout which will adhere to CRAP and Tufte’s principles.

4. Implement

a) Rapid prototypes. At this point we’ll have a great idea of what our final product might look like and we can begin implementing some basic visualizations that reflect all of our planning work. During this step we can further explore what is effective and what is possible / feasible.

b) Define data structure. We have worked with the datasets quite a bit at this point and during this step we will potentially perform the final transformations of the data. This involves implementing code and exporting to a suitable format. The final format of the data will be informed by all of our exploratory work up to this point, and what we have decided on for our final implementation targets.

c) Explore efficient algorithms. With our implementation goals in mind, we must consider what may be computationally demanding and how we can best approach the problem. We will avoid premature optimization, though, and simply try to approach problems in a logical and methodical way so as not to create unnecessary problems for ourselves in the future.

d) Design system architecture. We will implement the final program structure, having explored data formats and efficient algorithms. This implementation will require a full knowledge of exactly what we want the final product to be.

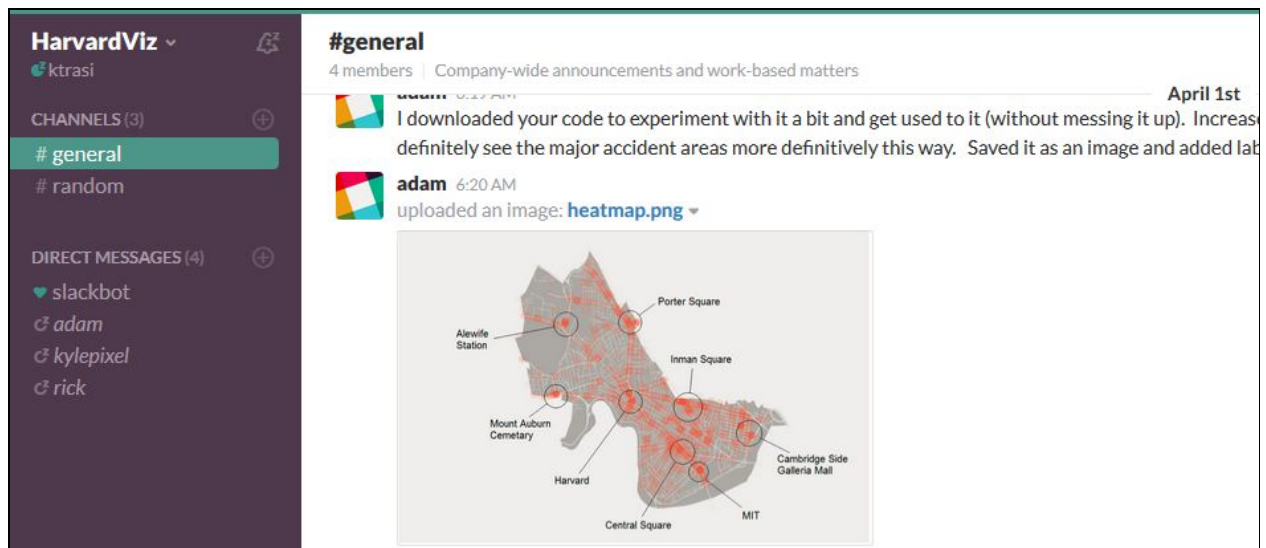
5. Evaluate

a) Is the abstract right? We will evaluate whether we accomplished what we set out to do. Does our implementation properly explore the questions we set out to answer? Can any small adjustments be made to get us closer to our original goal?

b) Do encoding and interaction support the task and/or provide (new) insights? We must look back at all of the planning we have done, even as far back as the existing solutions we explored, and ask ourselves if we chose the best possible encoding for the data. Does the visualization convey the information we wanted to convey? Do the interactions support this, as well? Can the user explore the data in a way that is useful for answering the questions that interest us? Can the interaction lead to interesting insights that are not necessarily present in the static visualization?

Communication Method(s) / Expectations:

Slack will be be our primary method of communication. The expectation is that the team members will check slack throughout the project duration and respond as needed to notifications for @everyone or @username.



On Slack, to notify all members of the team please use the @everyone or @channel keywords. To notify individual members of the team please use the form @jamesbond

Meeting Schedule / Method(s):

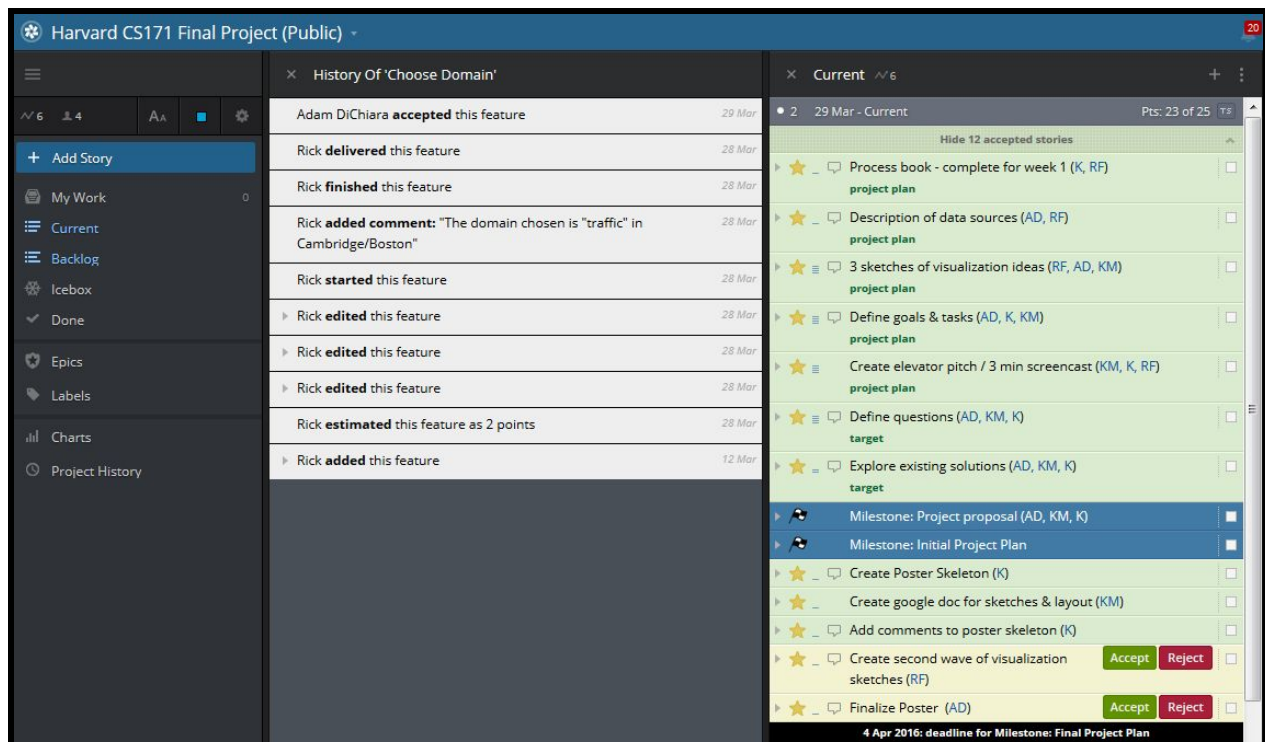
A dial-in conference call is used to ensure that the team has assigned current work to individuals, is on track for milestones, and to demo individual work to the other team members as we progress, as well as discuss planning and resolve outstanding questions.

We have a weekly meeting at 8pm EST each Wednesday. Our dial-in number is 1 (888) 643-3084,,,96848588#. Additional ad hoc meetings may be scheduled (most likely on the weekend) for required collaboration on tasks and mid-week updates. We will use Screenhero for screen sharing (<https://screenhero.com>).

Implementation / Collaboration Plan:

Pivotal Tracker will contain the living implementation and collaboration plan

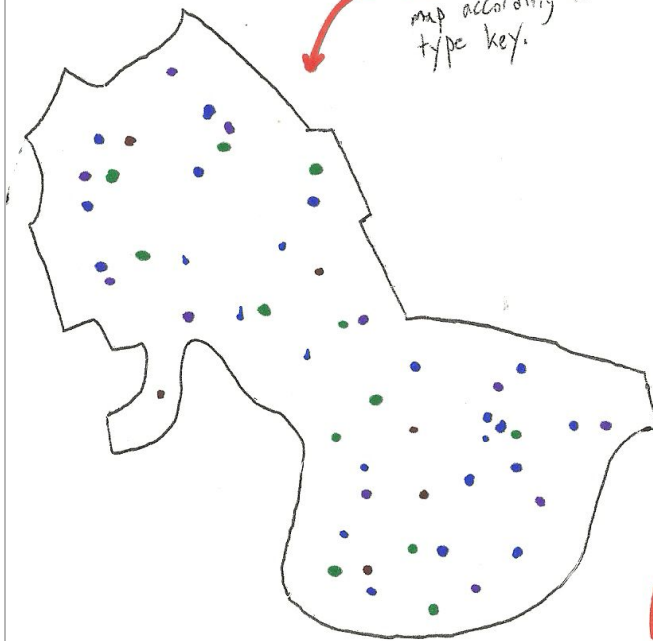
(<https://www.pivotaltracker.com/n/projects/1554113>). As the project evolves our tracker will be updated with the details of the assignments for each member of our team and associated delivery horizon.



As we begin to implement our project, when possible the team will pair on items in tracker. Pair-programming (aka XP) helps to make complex items less difficult, promote efficiency, and share knowledge among our team members while allowing us to deliver quality work more quickly. Screenhero is a tool that enables remote pairing. **Second iteration of Sketches**

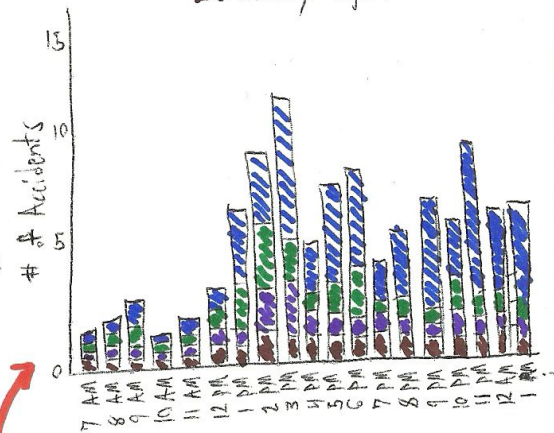
Here are the second iteration of the sketches. There is a common theme of using geo-mapping and small multiples or trends with crossfiltering.

Data Points are colored on the map according to the accident type key.



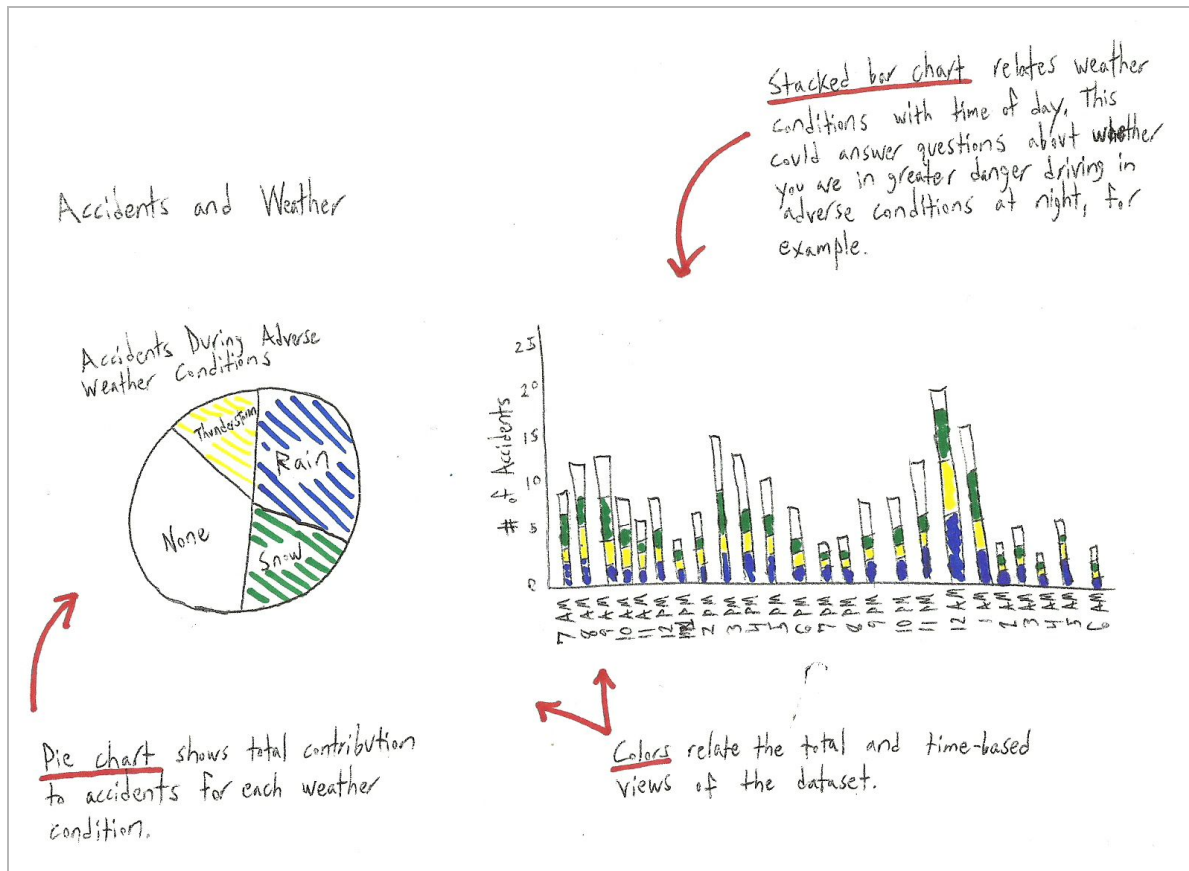
Accident Types by Hour

■ Car ■ Bike ■ Pedestrian ■ stationary object

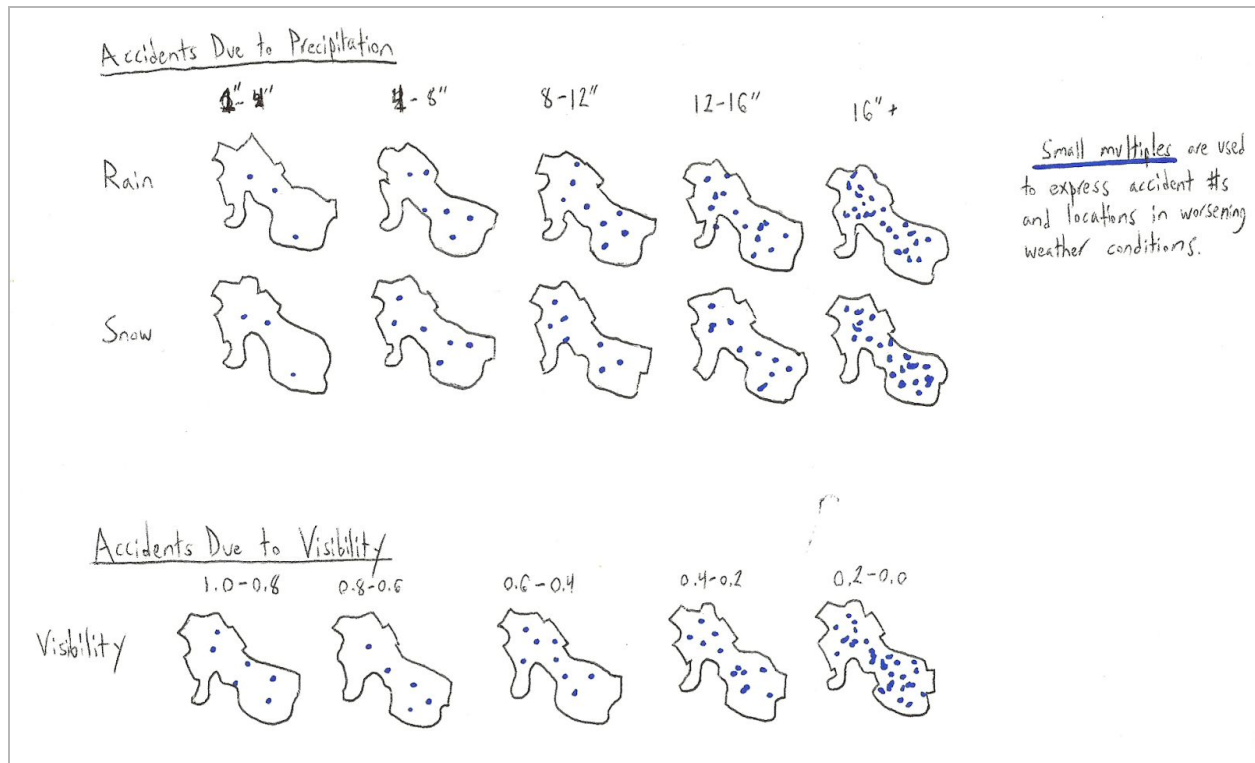


stacked Bar chart shows # of accident types based on time of day.

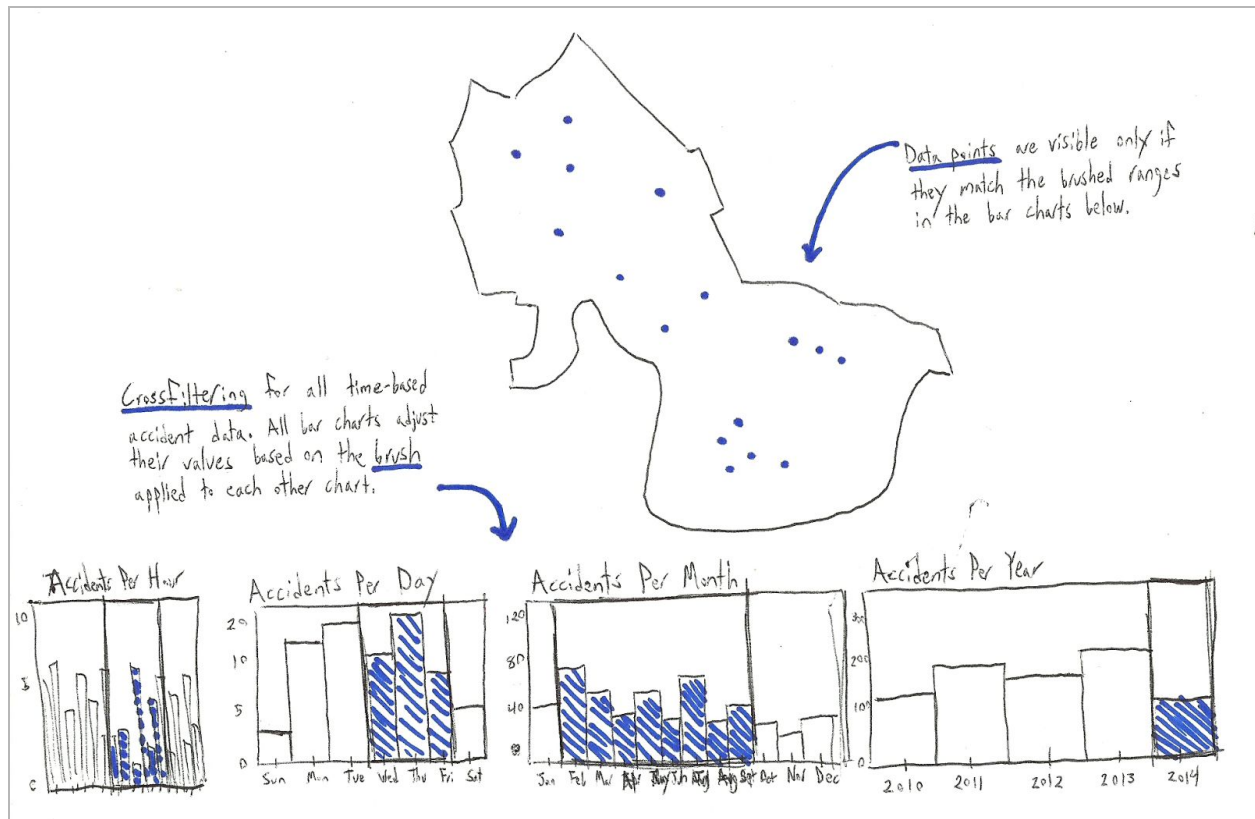
Sketch 1: Accidents expressed by accident type, and by time of day.



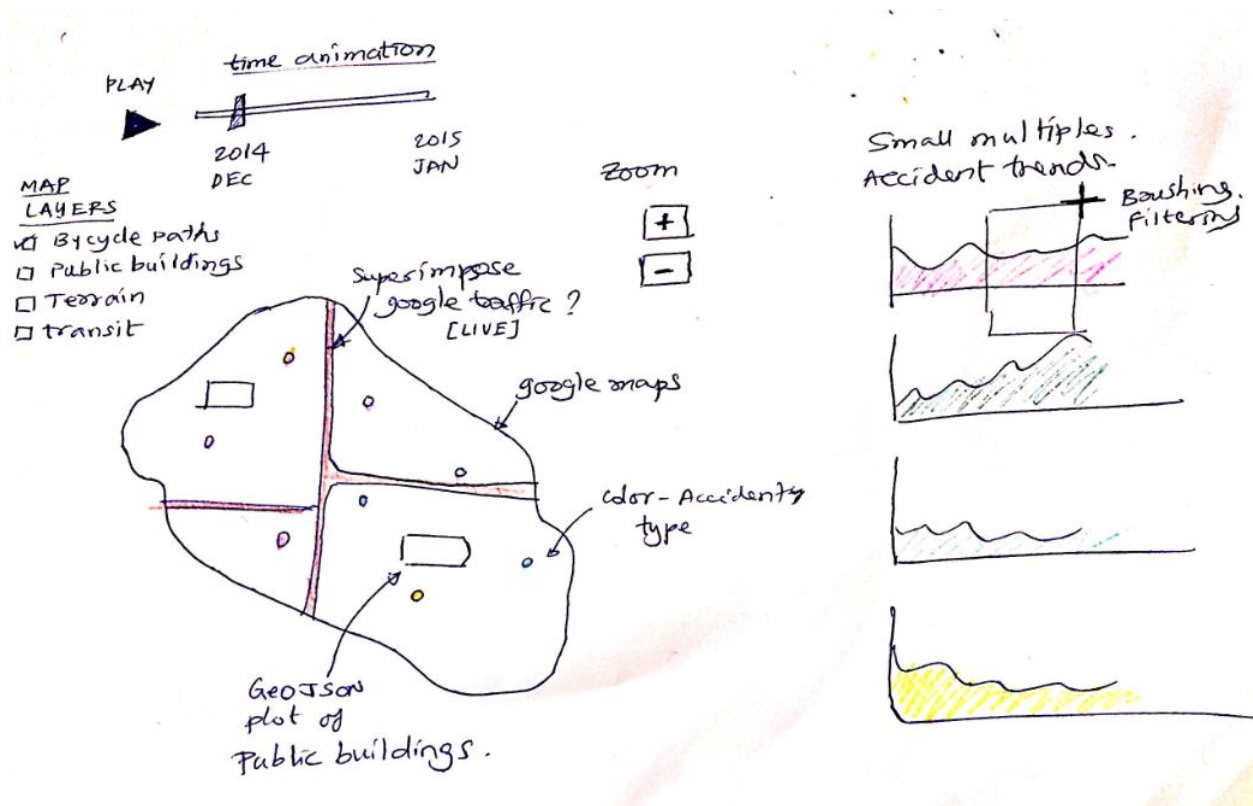
Sketch 2: Weather contribution to accidents, both overall and based on time of day.



Sketch 3. Small multiples to express adverse weather conditions and their contribution to accidents at increasing levels of severity.

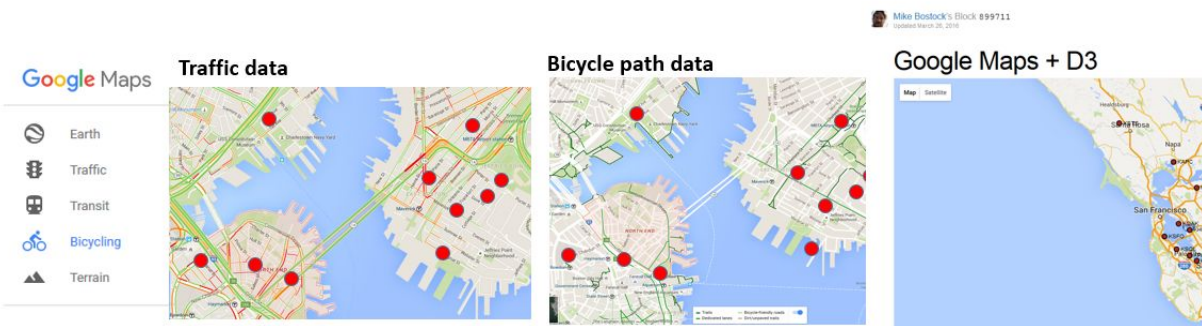


Sketch 4. Extensive time-based cross-filtering to display accidents in Cambridge.



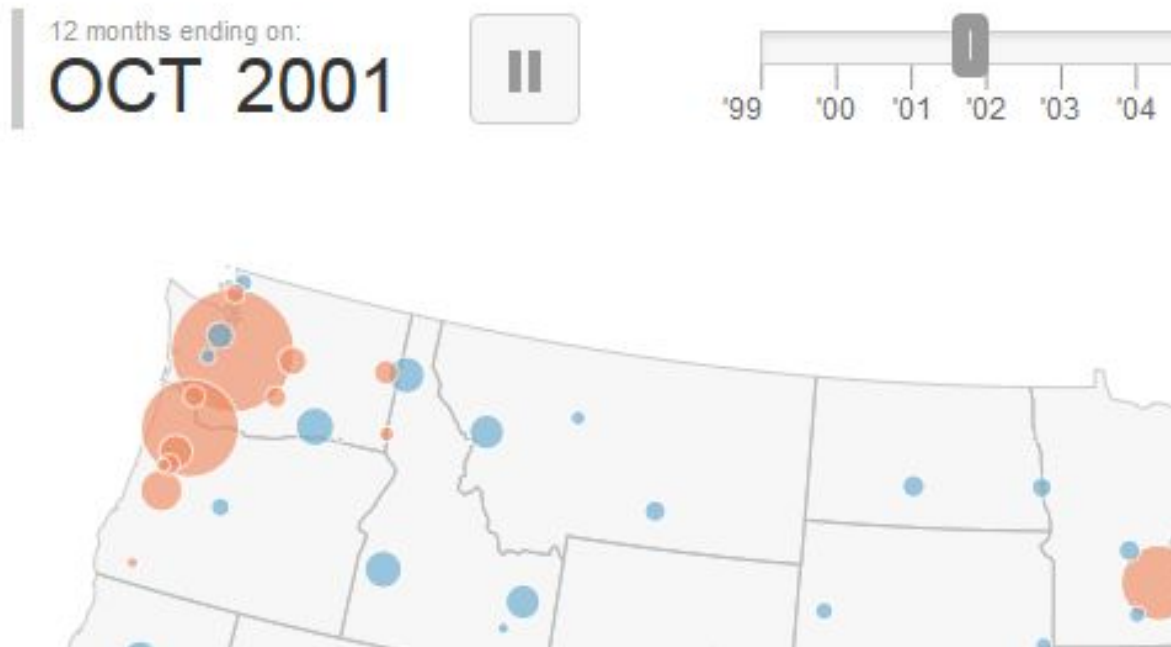
Sketch 5 - Geomaps, cross filtering, brushing, google maps / or equivalent (layers: traffic, terrain, transit, bicycle paths), public buildings

1. Google map integration will allow flexibility to use additional google map features
2. Live traffic data superimposed with our data will be interesting viz
3. Adding bicycle paths, terrain, transit... could be interesting as well
4. <http://nyc.crashmapper.com/11/8/14/2/standard/collisions/2/11/40.711/-73.772>



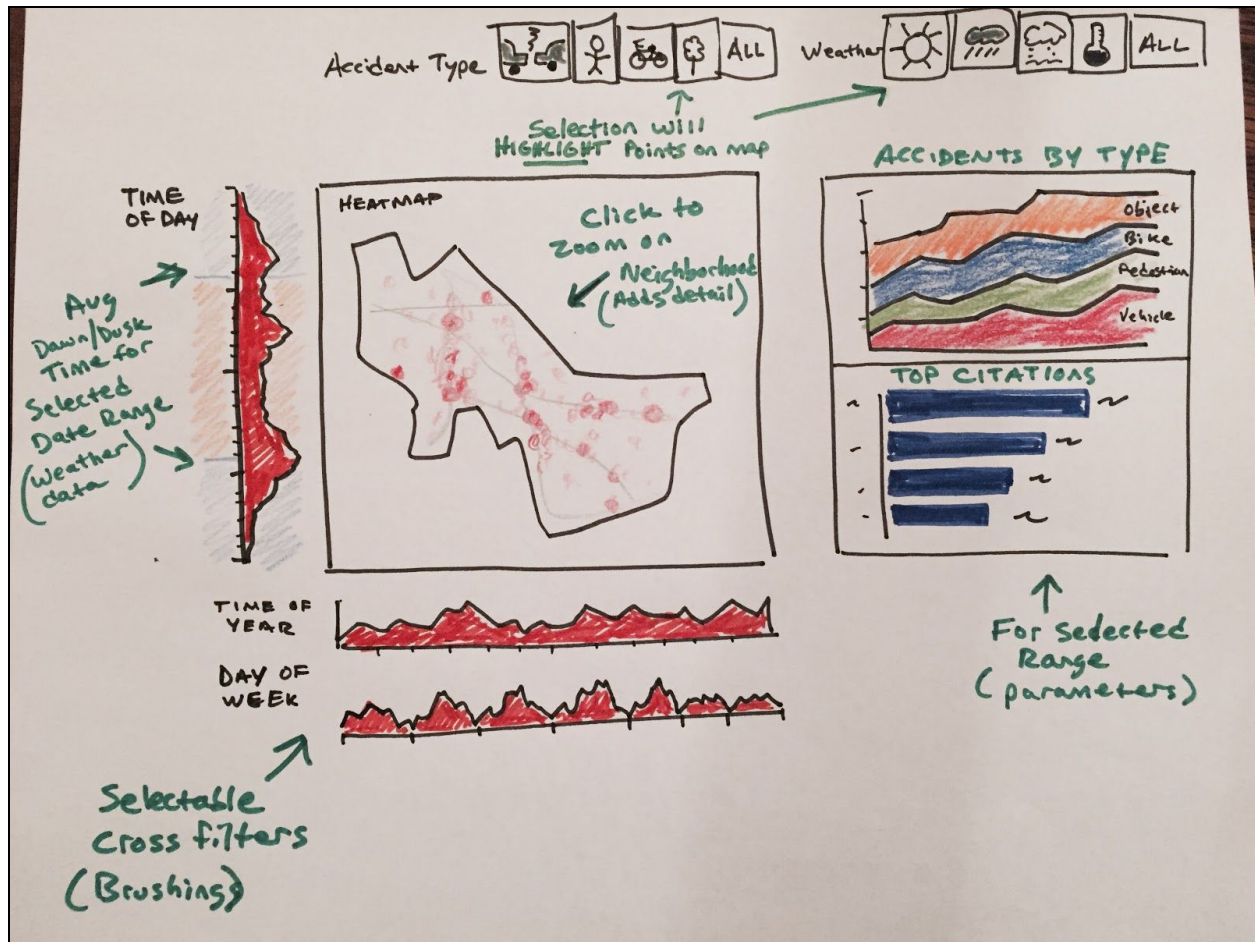
Sketch 5A: - Layers showing traffic, terrain, transit, bicycle paths...

Link to D3 sample code: <http://bl.ocks.org/mbostock/899711>

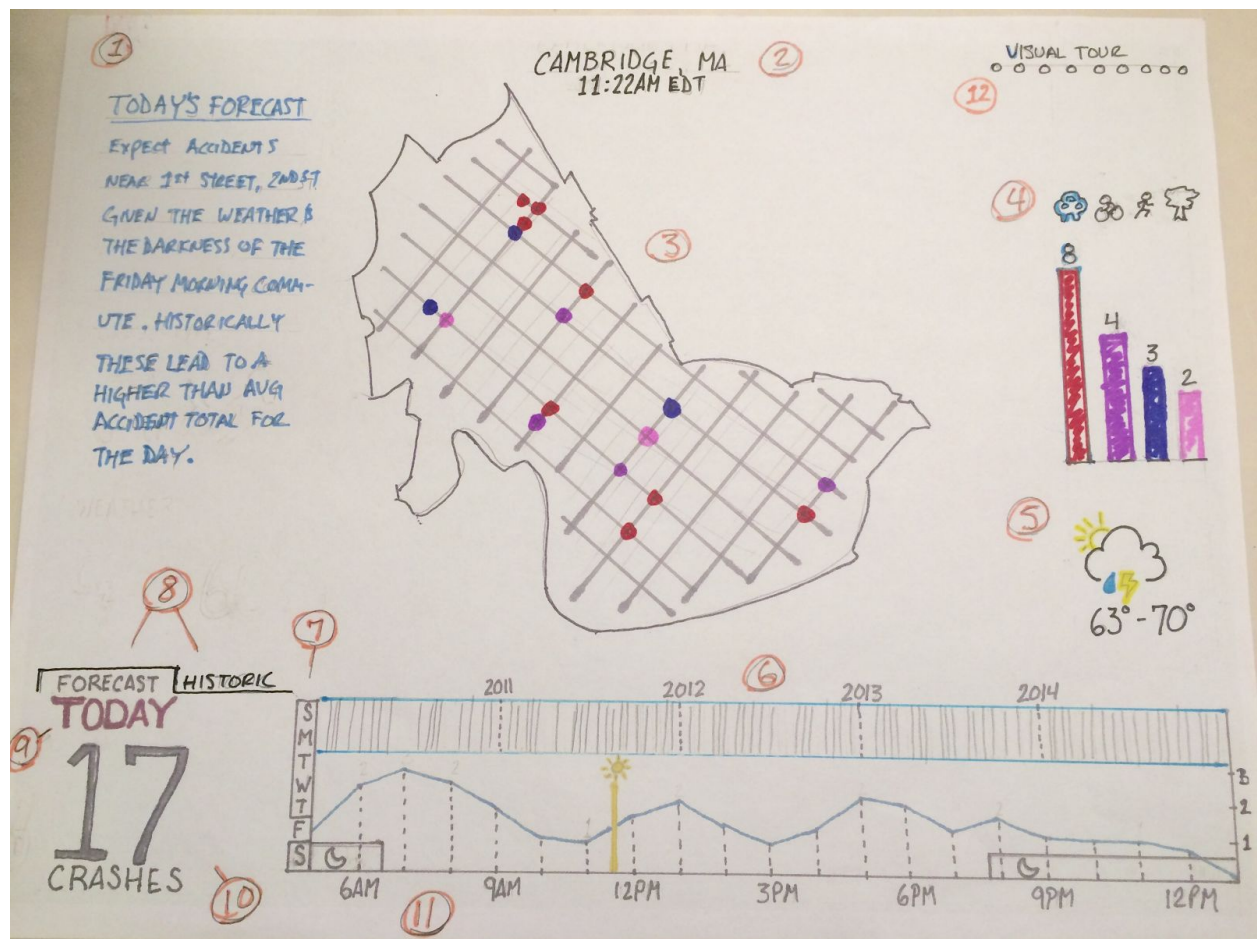


Sketch 5B: Animation through time will give the appearance of the data being dynamic

Link to example: <http://tipstrategies.com/geography-of-jobs/>



Sketch 6: main display multiple linked visualizations with multiple crossfilters.

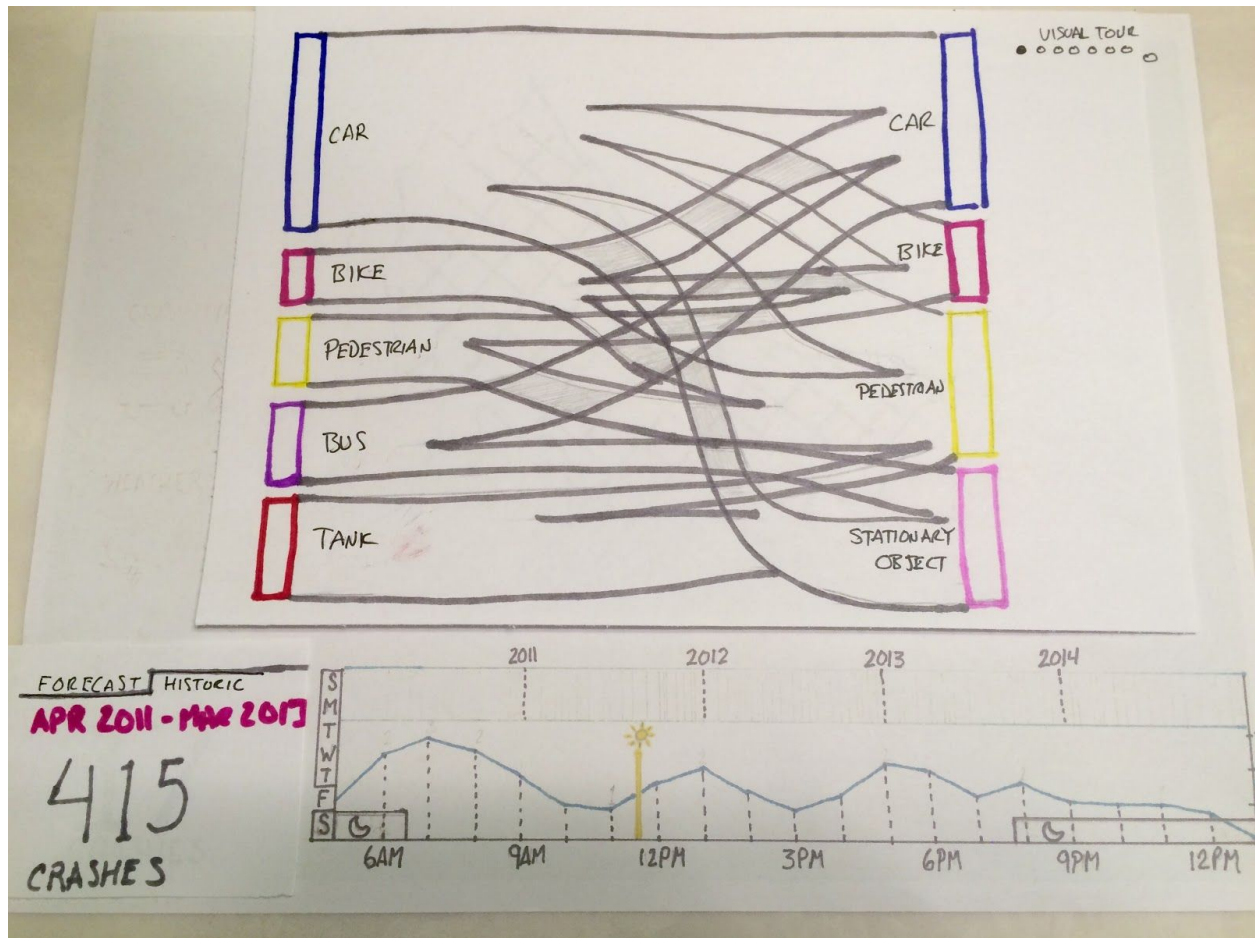


Sketch 6A: Crash Forecast (Crashcast?) landing page. This would make the project useful everyday for a commuter in Cambridge. Additionally, the user may explore historic crash data and could take a visual tour of the story-based data about crashes in Cambridge using the excellent visual design the rest of the team has suggested above.

Using the circled numbers above:

1. Today's crash forecast. Auto-generated forecast text that takes into account the various dimensions of Cambridge data (week of year, day of week, weather conditions, etc.) to make some generalized prediction about intersections to avoid based on the average expected total of accidents for a similar day.
2. City, State, current time

3. *Map that shows the various types of accidents at the intersections that have previously had them. Clicking on a colored dot will give a hover tip that describes the previous accidents that the forecast is based upon.*
4. *The bars and icons are clickable filters that are linked to the map. Clicking either adds or removes the dots from the map based on crash types, for example cars, bicycles, pedestrians, stationary objects, etc.*
5. *Today's weather forecast that is used to help make the forecast. Clicking the icon, expands the weather condition icons out to the left and allows the selection of different weather conditions for historic filtering. The temperature below is clickable and will allow the selection of a range of temperatures to filter by for analyzing historic data on the map.*
6. *A brush filter that allows the selection of a range of years to display in the visualization above (the map may not be present, if the user selected visual tour, there may be some other viz present that can be manipulated as well). The vertical lines show the days with crashes on them.*
7. *A filter to allow the selection of "day of week" that works with items (6) and (11). For instance the day of week shown is Friday.*
8. *Selection of "Forecast" or "Historic" mode. Forecast mode is shown which predicts the number of crashes today or for the week, e.g. 17 as shown in item (10). In "Historic" mode "Today's Forecast" (item 1) is hidden.*
9. *The selected date. In "Forecast" mode this shows "Today". In "Historic" mode it shows the date range the brush filter (item 6) is set to, it changes as the brush ends are moved.*
10. *The count of either "Forecast" crashes or in "Historic" mode the total count of crashes within the year brush range (item 6). Updates as the brush ends are moved.*
11. *Time of day brush. In "Forecast" mode this shows the count of accident per hour throughout the day, along with the current time (see the sun icon and line at 11:22am), darkness based on the time of sunrise and sunset (see the moon icon and dark band at the bottom). In "Historic" mode the brush end allow the user to select a specific time of day range, the sun and moon icons are hidden.*
12. *"Visual Tour" allows the selection of additional visualizations that the team has chosen for storytelling about Cambridge via the dots or a left swipe.*



Sketch 6B: Crash Forecast (Crashcast?) “Visual Tour” page. This shows what happens once a user has selected a dot under the “Visual Tour” or swiped to the left to move to a curated selection of additional visualizations that the team has chosen for storytelling about Cambridge. Everything above the the bottom filters (year range, day of week, time of day range, historic/forecast) may be changed based on the various visualization the team has determined help tell the story of traffic safety in Cambridge.

Here I’ve used a Sankey diagram to map the size of the relationships between crasher and crashee. For example code that look much closer to the final viz, please see the experimental code at

<http://tributary.io/inlet/1217283ecbd2093073d4>

Description of team roles

Our project roles are determined by the work we are currently doing. Each of us contributes to assigned tasks that we agree to take on throughout the project. As a result, we all get to learn a variety of skills and engage in the process. We pair up to tackle complex tasks as needed to ensure that one person does not get stuck or overloaded. We each take responsibility to contribute to the success of our project in any role needed.

Administrative Roles are specifically assigned but Functional Roles and Domain Anchors will change from week to week, depending on what is required for current tasks.

Administrative Roles:

- Communications Coordinator (Rick Farmer): Responsible for managing communications and sharing platforms (Slack, GitHub, Pivotal Tracker, conference calls).
- Agenda Coordinator (Adam DiChiara): Proposes order of weekly tasks and initiates calls for group action. Responsible for keeping forward progress towards weekly goals.
- Documents Coordinator (Kyle Maguire): Responsible for coordinating assignment documents, ensuring completeness, finalizing, and submitting.
- External Liaison (Kartik Trasi): Responsible for coordinating with TF and/or studio on any team issues, questions, or feedback. (via Piazza)

Functional Roles:

- Presenter: Presents team assignments via Piazza, screencasts, YouTube, etc.
- Writer: Consolidates team input into initial drafts for assignments.
- Sketch artist: Consolidates team sketch designs / ideas into final presentable sketches.

- Researcher: Conducts online research on assigned topics to inform team decisions / alternatives.

Domain Anchors:

- Targeting: Choose domain, define question(s), explore existing solutions, formulate data analysis tasks.

- Data Wrangling: Find and clean data, EDA (exploratory data analysis), transform and summarize data.
- Design: Design visual encoding, design interactions, design layout and storytelling.
- Implementation: Rapid prototypes, define data structure, explore efficient algorithms, design system architecture.
- Evaluation: Is the abstract right? Do encoding and interaction support the task? And, provide (new) insights?

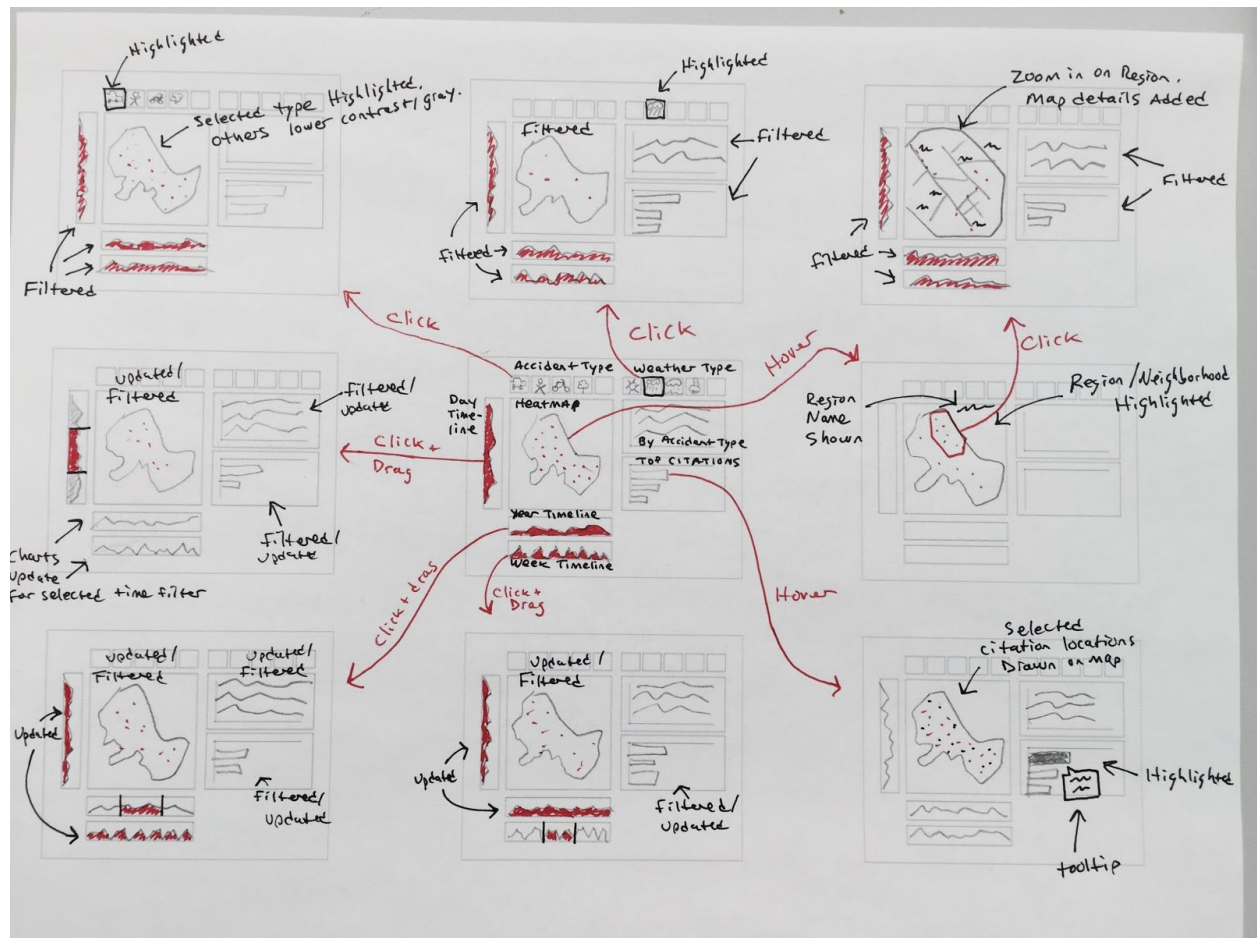
Chapter 3: Final Project Plan

Feature list

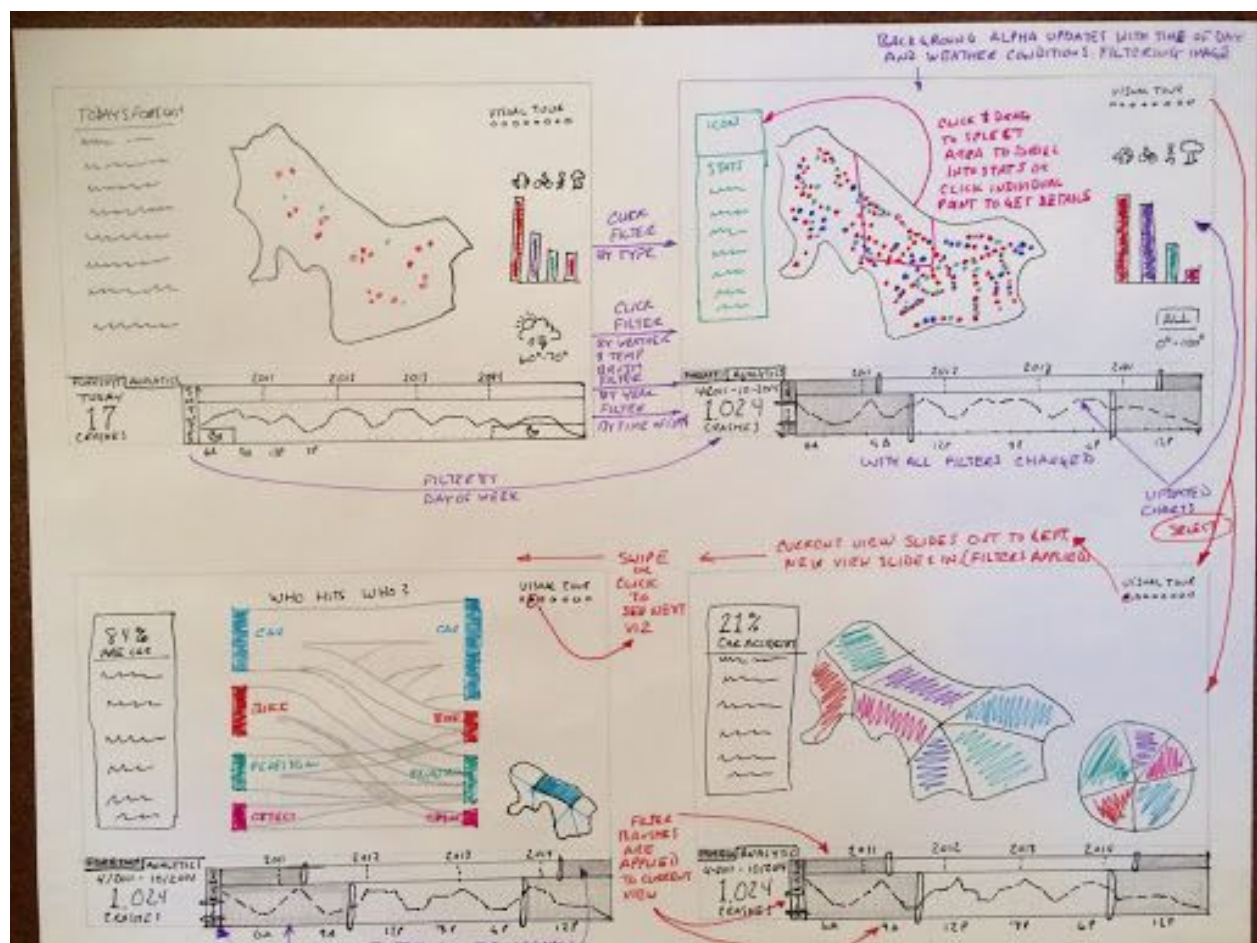
We plan to implement the following features:

1. Geo-mapping of data points
2. Forecast data plots
3. Trends by accident types
4. Cross-filtering
5. Plot of citations
6. Correlation with weather data
7. Additional map layers (transit, bicycle maps, public buildings)
8. Sankey diagram to map the size of the relationships between crasher and crashee

Interaction Storyboard

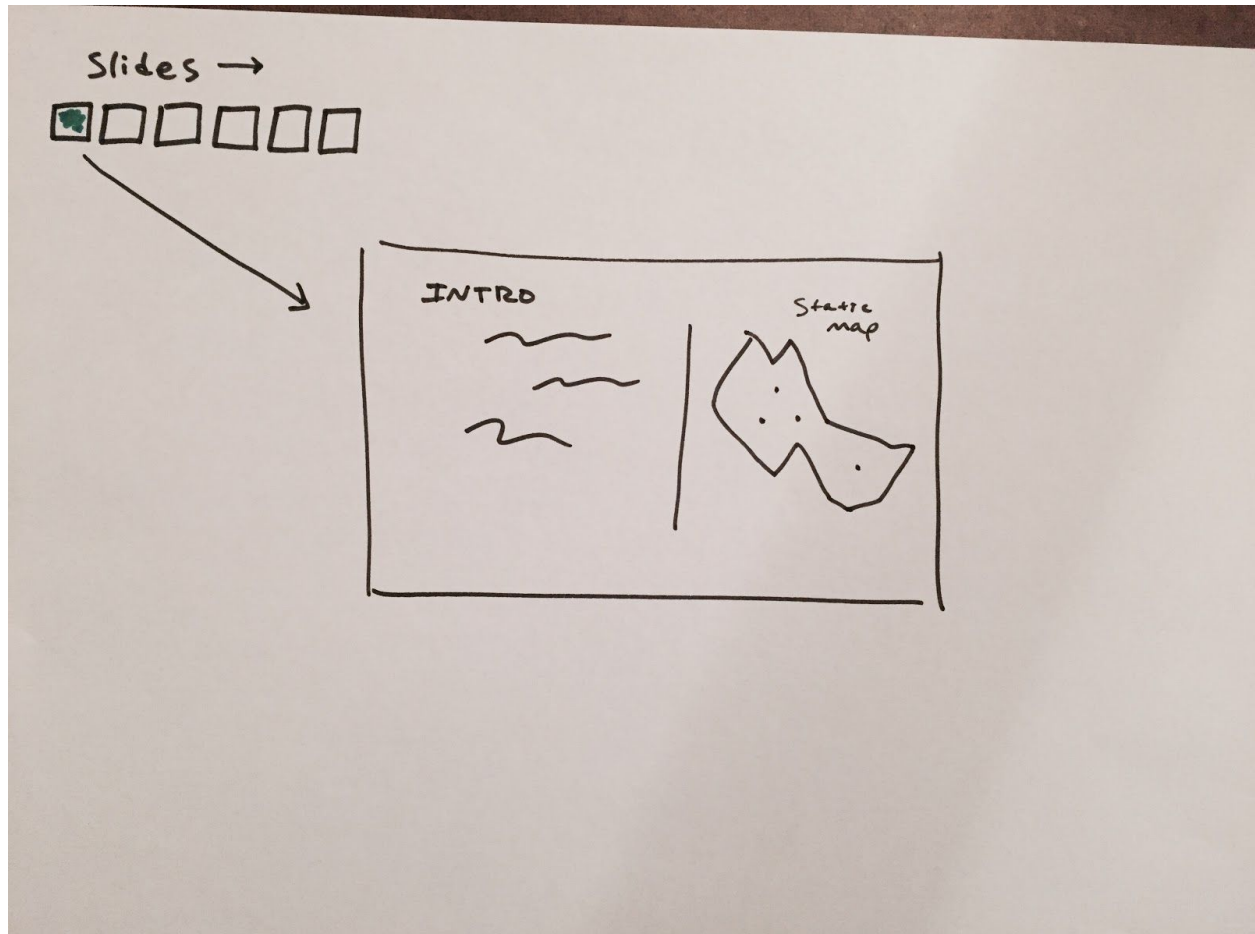


Single page, multiple views - Initial page is a forecast of today's accident hotspots to avoid. The user may switch from "forecast mode" into "analytics mode" by changing any of the filters (day of week, brush range over dd/mm/yyyy, brush range over time of day, type of weather, temperature, type of accident).



Webpage Layout & Storytelling

Our story telling approach will use a “martini glass approach” where we progress through highlight slides telling story and then lets user explore data in free form



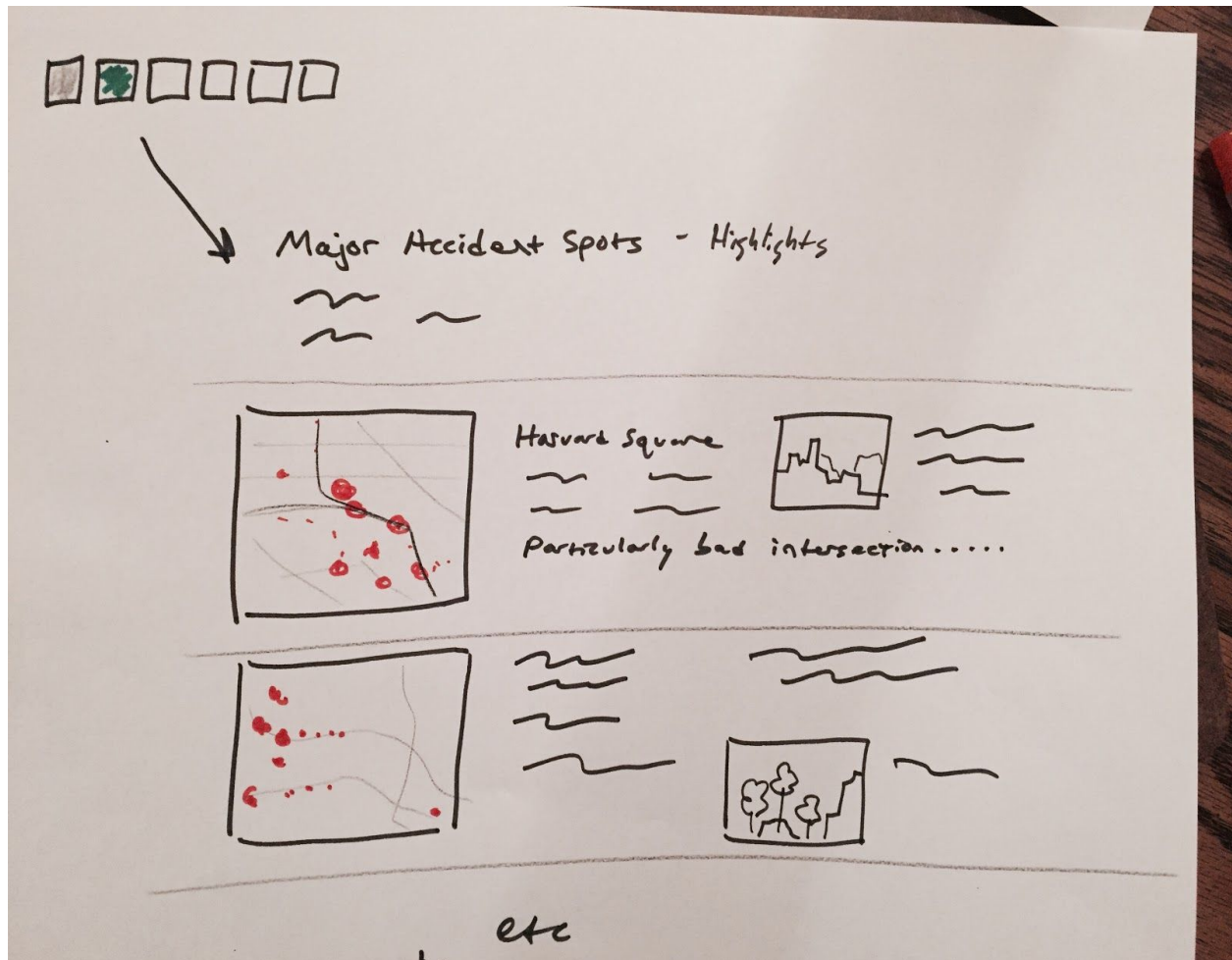


Figure: The first visualization will show our first highlighted story, navigation bar allows user to navigate through multiple stories

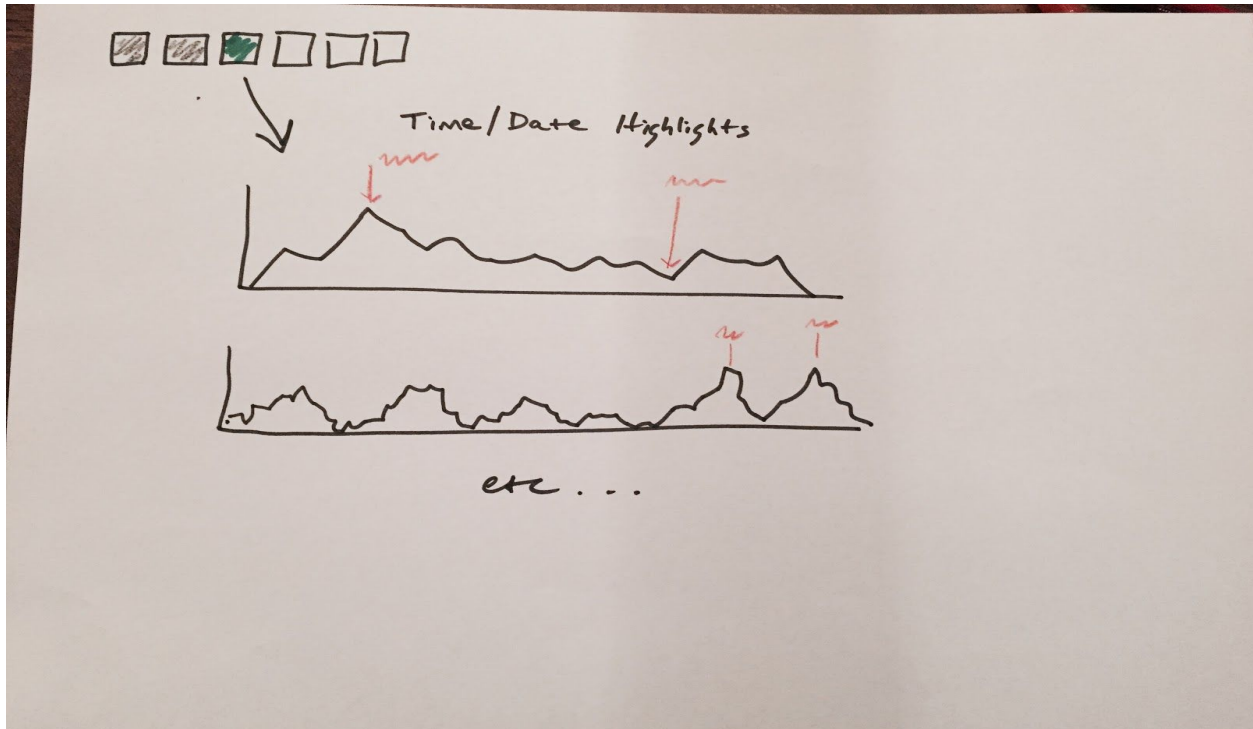
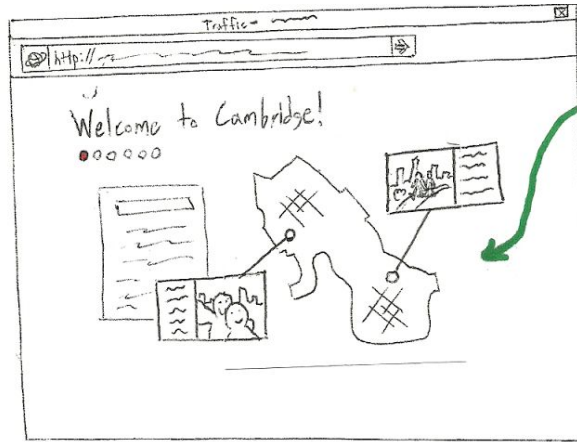


Figure: The first visualization will show our second highlighted story, navigation bar allows user to navigate through multiple stories

- ① Set the stage with some human elements. Engage with the viewer emotionally.

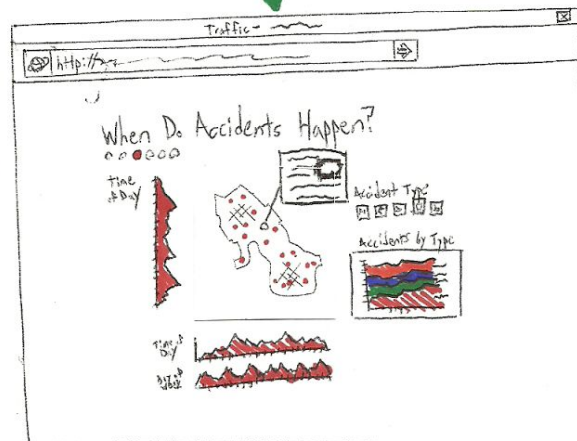


A common thread. The map of Cambridge acts as a constant touchstone for the storytelling and presentation.

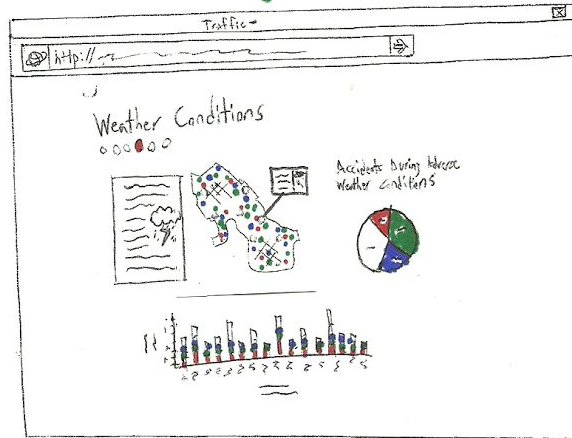
- ② Introduce the danger and show some basic, introductory data points without overwhelming the viewer. Highlight areas of interest.



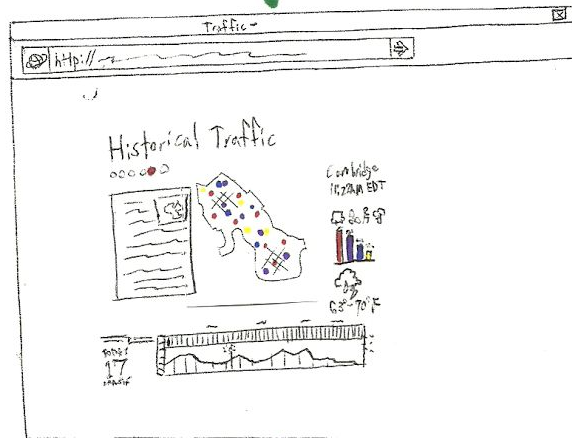
- ③ Discuss more specifics about when accidents happen and the contributing factors. Allow for some exploration on the part of the viewer.



- ④ Continue to elaborate on the various factors that may contribute to the accidents. Bring up all parts of the contributing sources.



- ⑤ Present an extremely flexible and dynamic view, allowing for unhindered exploration of the data by the viewer.



- ⑥ Summarize, and connect the story to the present. Present other sources of information and an infographic.



Chapter 4: Project Redesign

Heuristic Evaluation

Thank you for participating in a heuristic evaluation of our paper prototypes.

Our project aims to develop insightful visualizations representing public traffic safety data in a way which reveals probable "cause and effect" relationships between influencing factors and incident occurrences. Our goal is to employ good data visualization design principles and effective storytelling to create an end product that is both informative and visually appealing.

How to perform a Heuristic Evaluation

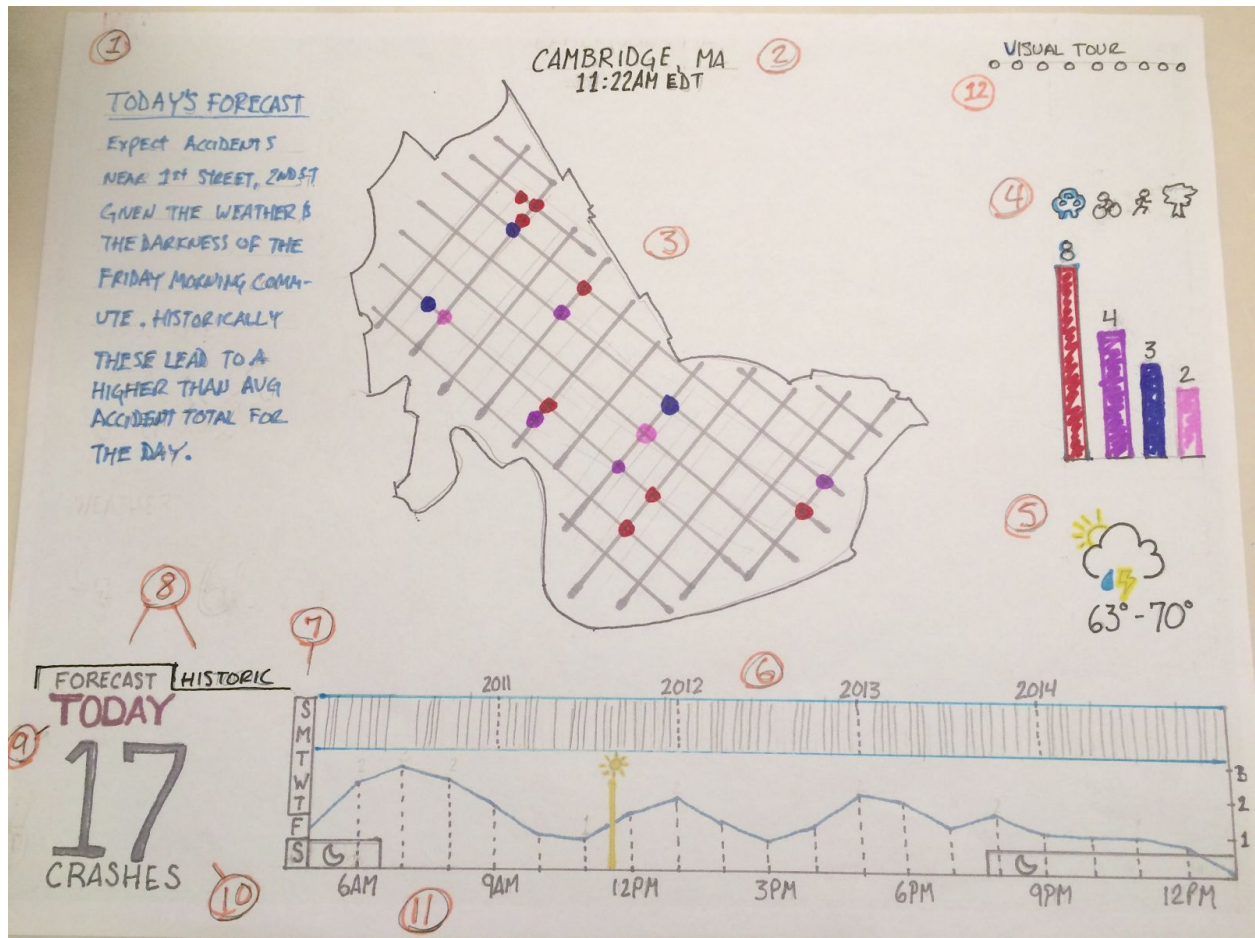
Please refer to the following post <https://piazza.com/class/ihv3pvmbyc86g7?cid=676>

Heuristic Evaluation of Project Crashed

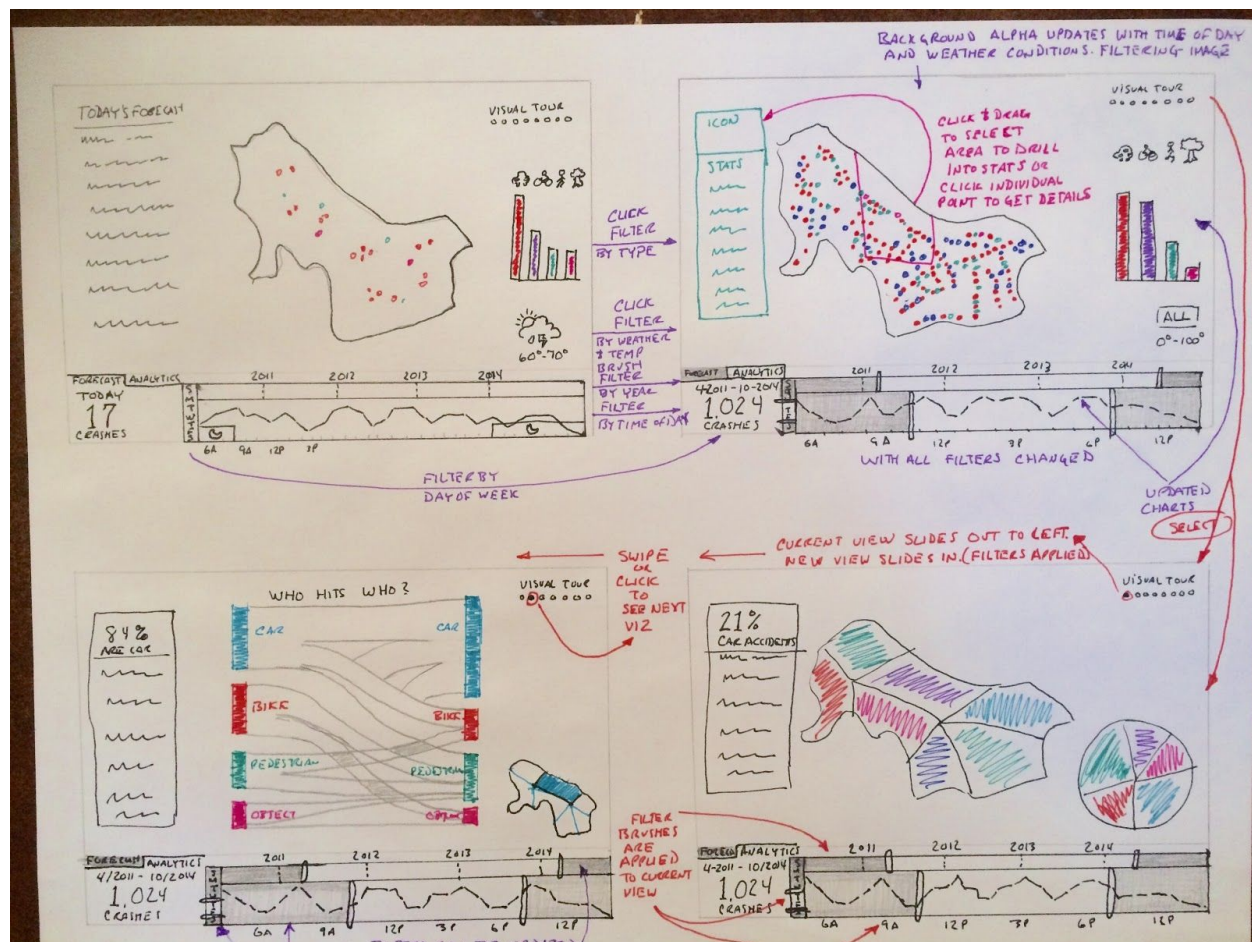
Question 1:

Given the following landing page detail and flow, please tell us in what ways you might explore historic/analytic crash data?

Landing Page (detail)



Landing Page (interactive flow)



Please provide your Heuristic Evaluation for question #1

Andrew:

First, I was confused by the large number under "today" -- I thought it was today's date, and only later realized it was the number of crashes predicted for today. The fact that the top of the screen has the current time but not the current date reinforces this misimpression.

I find the weather forecast confusingly placed, because it's presumably an "input" to your model, as opposed to everything else which is an "output".

In general, when I first saw this, I thought I was looking at "actual"/historical data; it took me a while to realize that everything I'm looking at is a forecast for today.

As to how I would explore:

I'd probably start my exploration with the dots labeled "Visual tour" in the upper right.

I'd also mouse over the dots on the map expecting to find details. Since these are forecasts, are you intending to give some indication of probabilities? (Perhaps by the alpha channel?)

I'd also try clicking on the icons above the bar chart, expecting them to filter the rest of the page.

The time span on the bottom looks like it should be brushable, so I'd play with that as well.

In looking at your second page, I see that you have a lot of ways to filter and slice the historical data. What's not clear to me now is how to link the historical filters to today's conditions.

Bertrand:

At the bottom left of the page (marked as 8) I am assuming that the label "Historic" is in fact a tab, that if you were to click on it you would see a time series view of actual accidents. Although not explicit, I am assuming that the dates include the ability to select a range of dates.

1. Visual Tour [Flexibility and efficiency of use] [Severity 3]

It is unclear to me what "Visual Tour" is on this landing page, it is somewhat disembodied from the rest of the presentation. The purpose of it should be more explicit, or possibly even remove it.

2. Temperature Visual [Aesthetic and minimalist design] [Severity 2]

I would suggest moving the Temperature Visual (Item 5) next to the Daily Forecast (Item 10)

Qutab:

At the bottom (marked as 7) I am assuming assuming that the days name include the ability to select a range of days. It is not clear how this will go together with dates and time.

I am assuming Line chart in the bottom showing the number of accident occur in different hours of the day but again it is not cleared that how will compare the accidents between different hours and days. [Severity 2]

Instead of today forecast (marked as 7) , if you can use today highlights or Today news, will be really enabled the team to share more information such as number of accidents happened so far or any other news regarding accidents. [Severity 3]

“Visual Tour” (marked as 12) on this landing page is not cleared.

Senthil:

Overall, I think it is a very impressive prototype. The level of effort went into the design and sketches are commendable. But there is always room for improvement. Following are my observations,

1. Visibility of system status [Severity 3]

Today's forecast for crashes is good but the placement of it is not ideal. The users tend to browse the pages in a Z pattern going from top left->top right->bottom left->bottom right. But since it is enlarged, it helps the user to focus.

Today's weather is not part of crash data, so it is better placed outside of the crash charts. It will be helpful to visually distinguish between forecast and actual accidents on the map.

2. Match between system and the real world [Severity 3]

While the different visualizations are intuitive, it took me a while to realize what each are.

Perhaps adding a heading to each chart will provide more context.

How do I interpret 1.024 crashes in real world?!

3. User control and freedom [Severity 3]

I would presume that clicking on an empty area in the map, clears the map and the filters below. But is not very straight forward how one can go back to the default state.

4. Consistency and standards [Severity 1]

Overall, I find the design consistent. The coloring scheme on the Parallel coordinates is a bit off though.

5. Error prevention [Severity 1]

There is not much room for the users to commit any errors. Not sure if the filtering status will be maintained when you switch between forecast and historic timeline chart.

6. Recognition rather than recall [Severity 2]

Visual tour is a helpful tool to navigate but one has to browse through them to find a specific layout. If you have few that won't be a problem though.

7. Flexibility and efficiency of use [Severity 3]

Some of the interactions are not very intuitive, like selecting an area on the map. Expert users will try that but novice users will need some hint.

8. Aesthetic and minimalist design [Severity 3]

In each page other than the landing page, there is the primary visualization and then there is a small viz on the side. What purpose does it serve? It is a distractor that takes the focus away from main viz on that page.

9. Help users recognize, diagnose, and recover from errors [Severity 0]

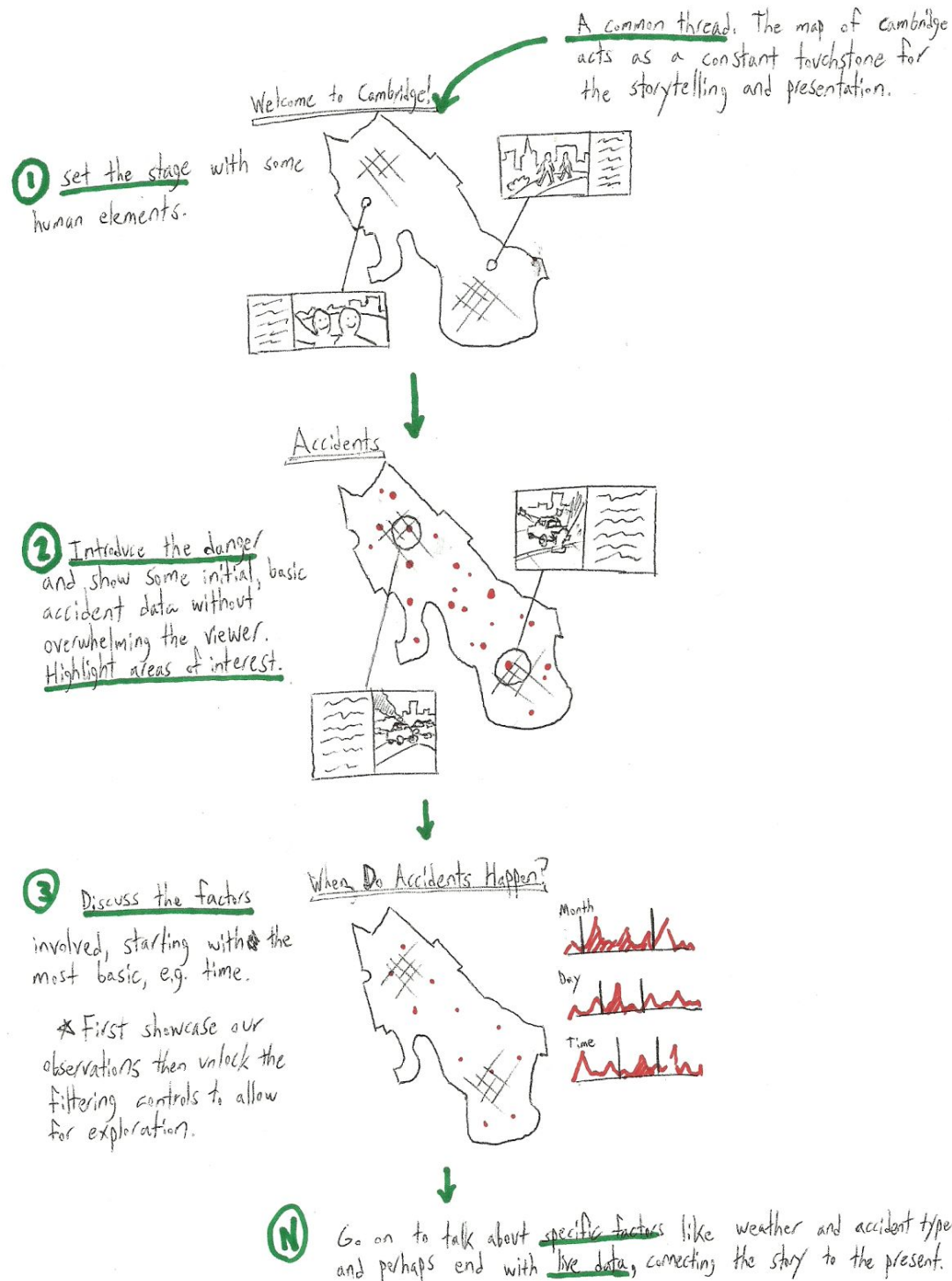
Not much room for error here.

10. Help and documentation [Severity 2]

The story helps set the context but some more hints on the landing page will help the user interactivity, as the user will forget what they saw in the story.

Question 2:

Given the following Storytelling Layout and flow, please tell us the story we intend to tell and provide insights into our storytelling flow? Storytelling Layout (flow)



Please provide your Heuristic Evaluation for question #2

Andrew:

First, I'm confused about how this relates to the landing page. It seems like *this* is what the user should see first, before dropping them into the full visualization.

I think your story is "There are accidents in Cambridge; by analyzing past patterns we can identify which are the most significant contributing factors." It is a small concern to me that you don't introduce your main topic until the second page.

You are also introducing your visual vocabulary here, which will help the user navigate the full view.

Bertrand:

I am assuming that each of the large visualizations will be a separate page.

The flow is one of increasing focus, starting off with filtering on day of week, or type, you flow to a page that shows a view of what was filtered upon. If I wanted to know what accidents occur on a Friday, the next page should show me that information. Then I can select further to see a breakdown of accident types, then on the following page who hits who.

1. Visual Tour [Recognition rather than recall] [Severity 3]

With "Visual Tour" as a user clicks through the sequence of pages, it can be confusing because as a user you need to remember the context under which you first selected, but then clicked on the visual tour to take you somewhere else. The user can lose context of where they are. Then you run the risk of them thinking that this is too difficult or complex to use.

2. Map with Pie chart [Aesthetic and minimalist design] [Severity 3]

The page in the lower right is not additive. You already have two other map oriented pages. You may want to consider dropping this page and going to the "Who Hits Who" page. Your story or narrative will not suffer as a consequence.

3. Page and Visual Complexity [Aesthetic and minimalist design] [Severity 3]

Overall there is a lot going on in each page, as in Map, Sliders (multiple), Bar Graphs, narrative. Simplifying would be visually easier to digest. For example: On "Who Hits

Who” page, there is a little Map, is this map really necessary? Does each visualization add to the story or flow, if not remove it?

4. “Home” page Navigation [Flexibility and efficiency of use] [Severity 3]

It is unclear from the flow how you get back to the 1st landing page.

Qutab:

Navigation in pages are not cleared. I think it will be better to use proper and attractive sub story title, icon or element instead of small dots, which will make the visualization user friendly and will provide a good experience to users. [Severity 3]

Senthil:

This is a nice feature to set the context for the story.

1. Visibility of system status [Severity 3]

The user does not know how long the story will be, so providing a status to how far they are in the storyline or offer a way to skip the story will be helpful.

2. Match between system and the real world [Severity 1]

While we know about Cambridge, if this is intended for a global audience, they may need some background on Cambridge. But it depends on your intended audience.

3. User control and freedom [Severity 2]

Provide a way to navigate or skip through the story.

4. Consistency and standards [Severity 2]

Using a same coloring scheme and controls are the landing page in the story slides will help.

5. Error prevention [Severity 2]

What is the user accidentally skips the story but wants to get back to it. Provide a link in the landing page to go back to the story.

6. Recognition rather than recall [Severity 2]

See comments under Consistency and standards.

7. Flexibility and efficiency of use [Severity 1]

Navigation buttons will go a long way

8. Aesthetic and minimalist design [Severity 1]

Less text and more hints on top of the actual viz will be helpful.

9. Help users recognize, diagnose, and recover from errors [Severity 2]

see comments under Error prevention

10. Help and documentation [Severity 0]

Since this is the documentation, it is a welcome addition to set the context for the visualization.

Innovative Solution

Weakness:

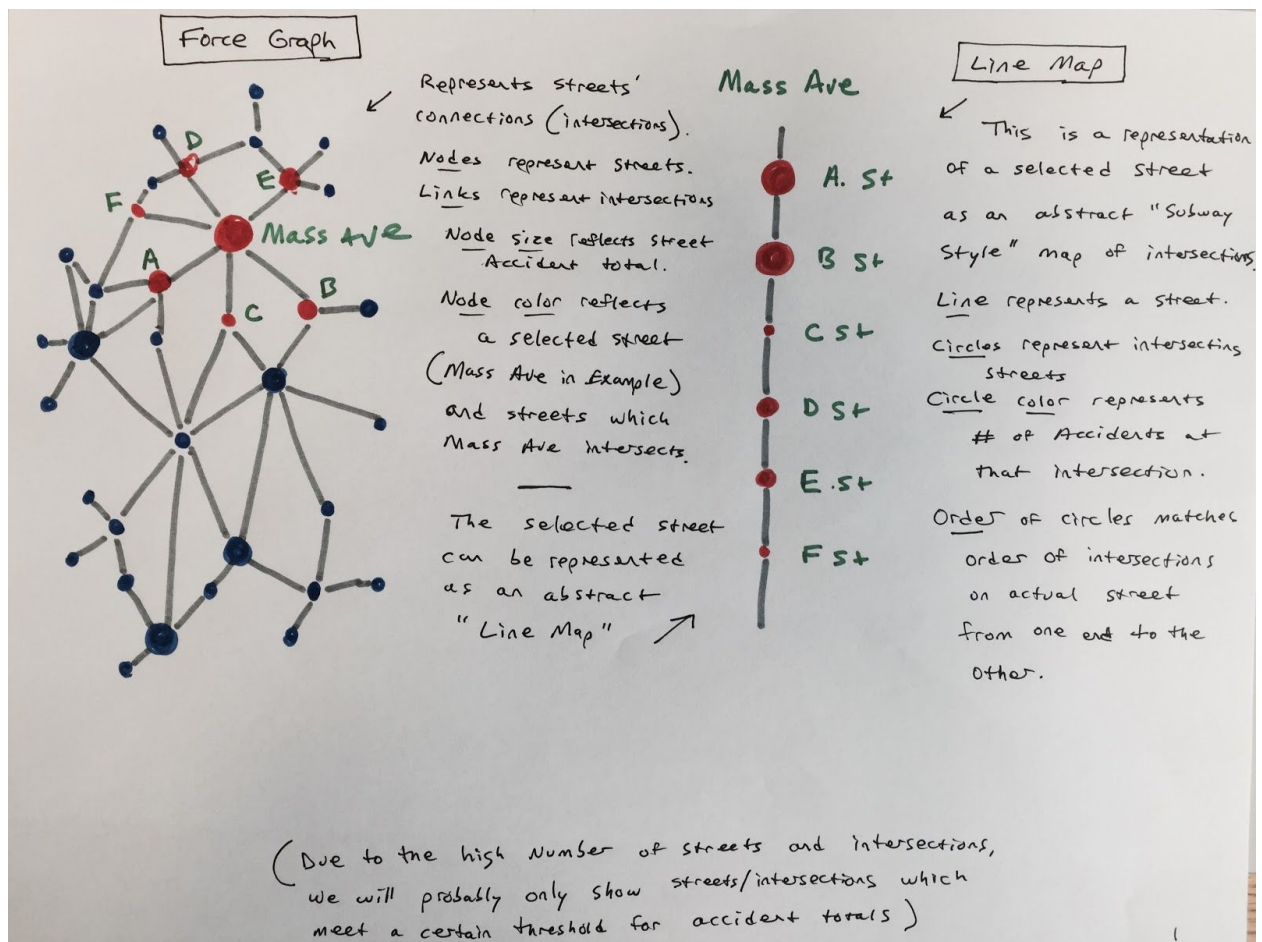
A weakness we identified in our designs is an *overreliance on a map* for representing our key data. While we still feel a map visualization is critical to our project, we also felt the need to create visualizations that do not rely as heavily on the geographic nature of our dataset. We also recognize that the amount of data shown on the map can become overwhelming, and although it is great for seeing overall trends it may not be useful for identifying issues on particular streets.

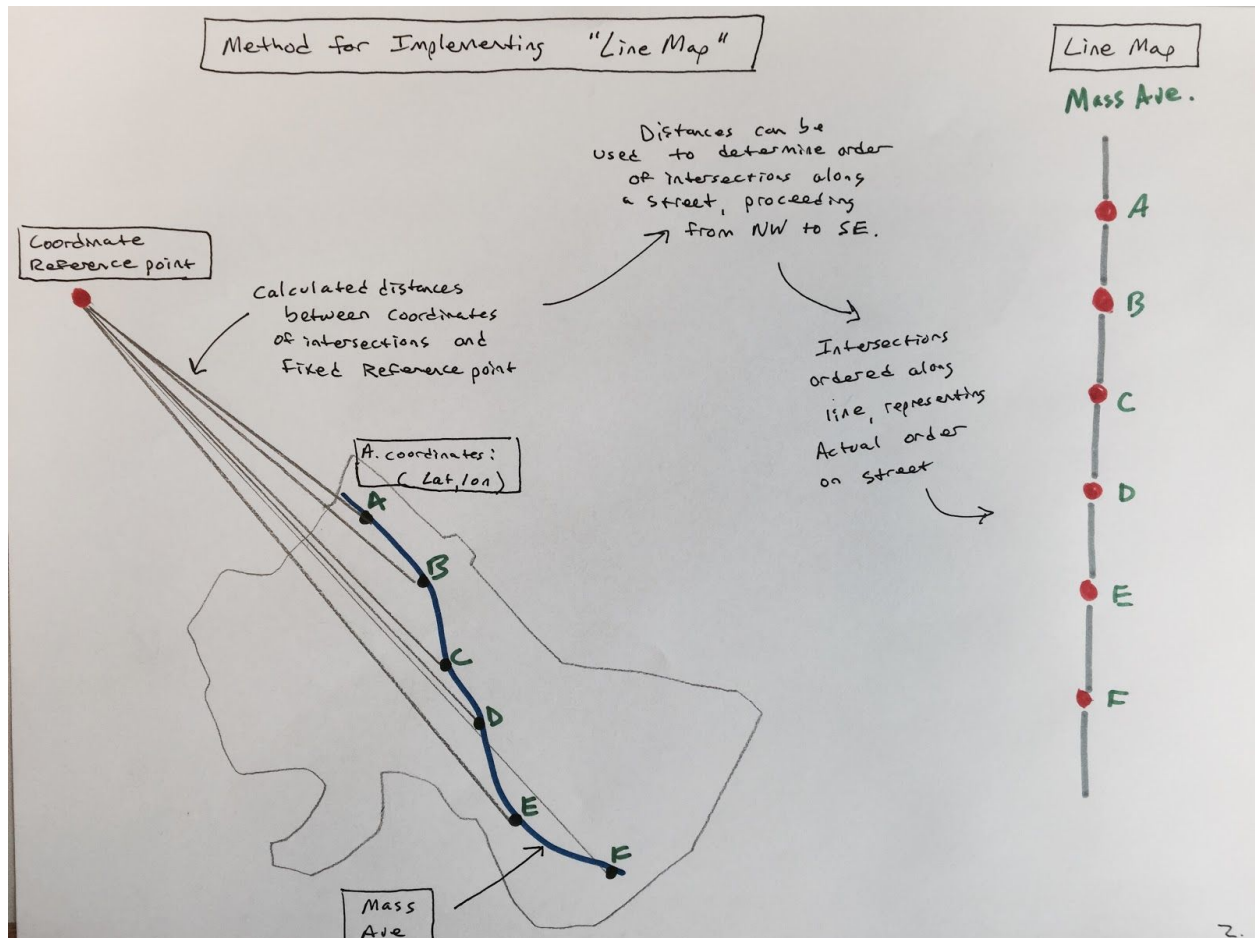
Solution:

Thinking about the question “how do you represent geographic data without a map”, we designed a set of visualizations to support the map and other visualizations which focusses on particular streets and their intersections, without representing their exact geographic location. For representing a street we will use what we’re calling a “Line Map” which is an abstract, subway style diagram, of a particular street and it’s major intersections. The street itself will be

represented by a line, while the intersections will be represented by circles along this line, positioned in order of their actual ordering on the street. (This ordering will be determined dynamically by calculating each intersection's distance to a fixed reference point outside cambridge, using lat/lon coordinates). The circle sizes would represent accident numbers for those intersections. This visualization would answer the question "If I'm driving along a particular street, which intersections are the most dangerous?". By removing the exact geographic data but retaining order we achieve a visualization that simply and effectively answers this question. The second related visualization is a force graph in which the nodes represent streets and the links represent intersections between streets. Node size would represent total accidents for that street. This visualization would be linked to the Line Map so that a selected node(street) on the Force Graph would create/update an adjacent Line Map.

Sketches:





Feedback Integration

Weakness:

As part of our peer feedback we received we found that there was some weakness in the *transition points* of our design. Particularly the lead-in to the "Visual Tour" and helping lower cognitive load as the user traverses the key visual points we are driving them through.

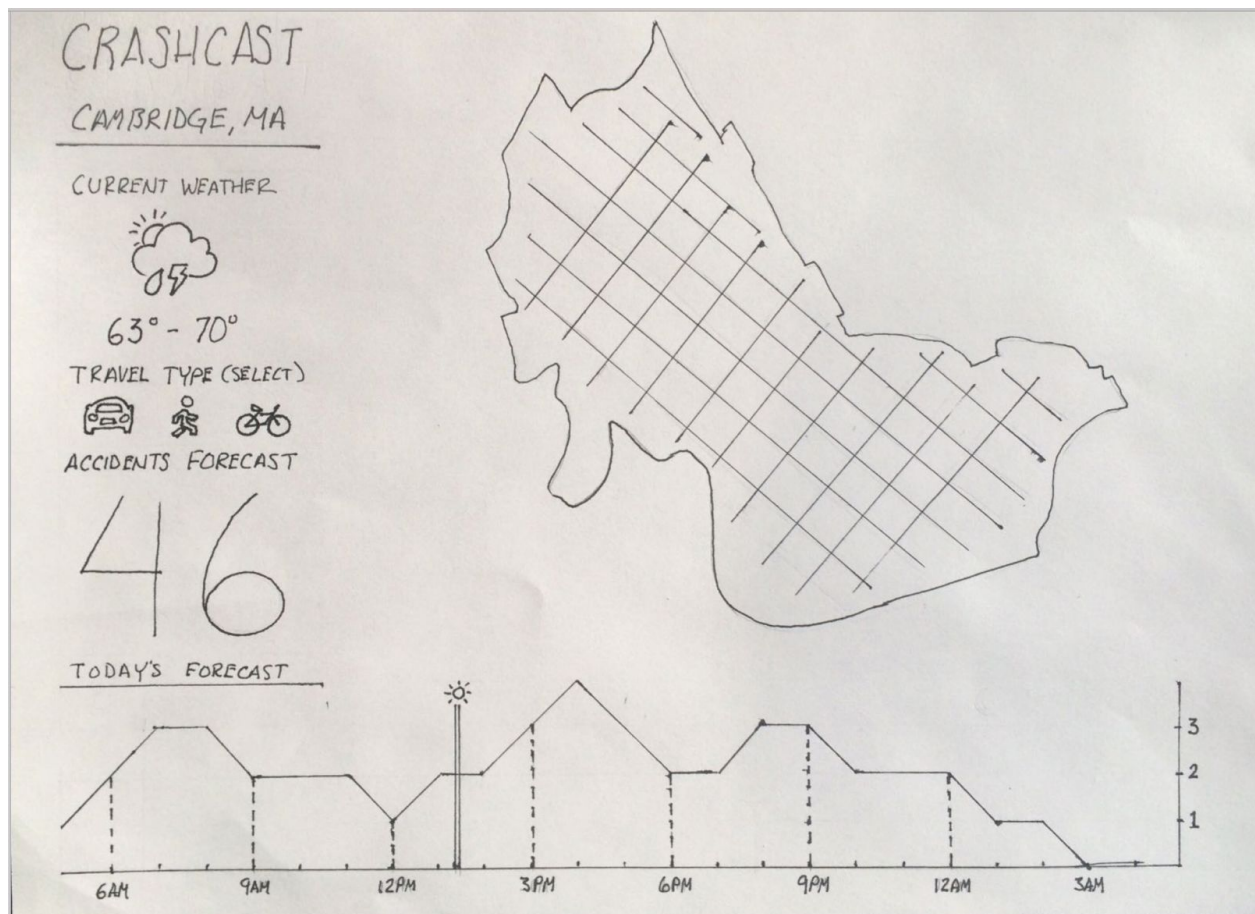
Solution:

As a result, we have modified our flow and simplified the initial design points to enable the user to acclimate to the capability and story we are trying to drive.

- We have simplified the “Crashcast” forecast landing page to be less interactive; but, more informative on the key points the forecast is trying to assist the user with.
- We then transition the user to the introductory storytelling elements, then build those in stages to the conclusion of the story we want to tell.
- We then transition the user to the “Crashboard” which gives them the full interactive experience as shown in our “Crashcast” Analytics view previously.

Sketches:

Simplified “Crashcast” landing page



Redesigned page flow from “Crashcast” landing page through “Visual Tour” through “Crashboard” followed by the “Process Book”.

FLOW

CRASHCAST

①

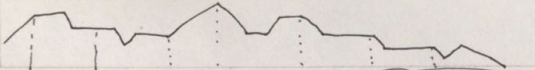
CRASHCAST
CAMBODIA, MA



46

46

FORECAST



SCROLL DOWN

SCROLL DOWN

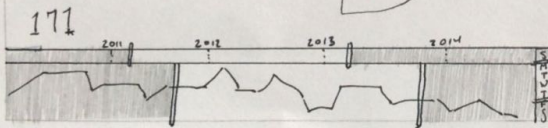
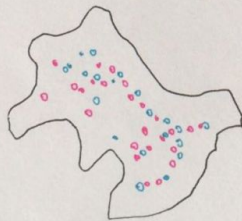
CRASHBOARD

③

CRASHBOARD



171



②

VISUAL TOOL

VIZ Ø

VIZ N

CLICK TO
SLIDE LEFT OR
RIGHT

④

PROCESS BOOK

MAAT

MAAT
MAAT
MAAT

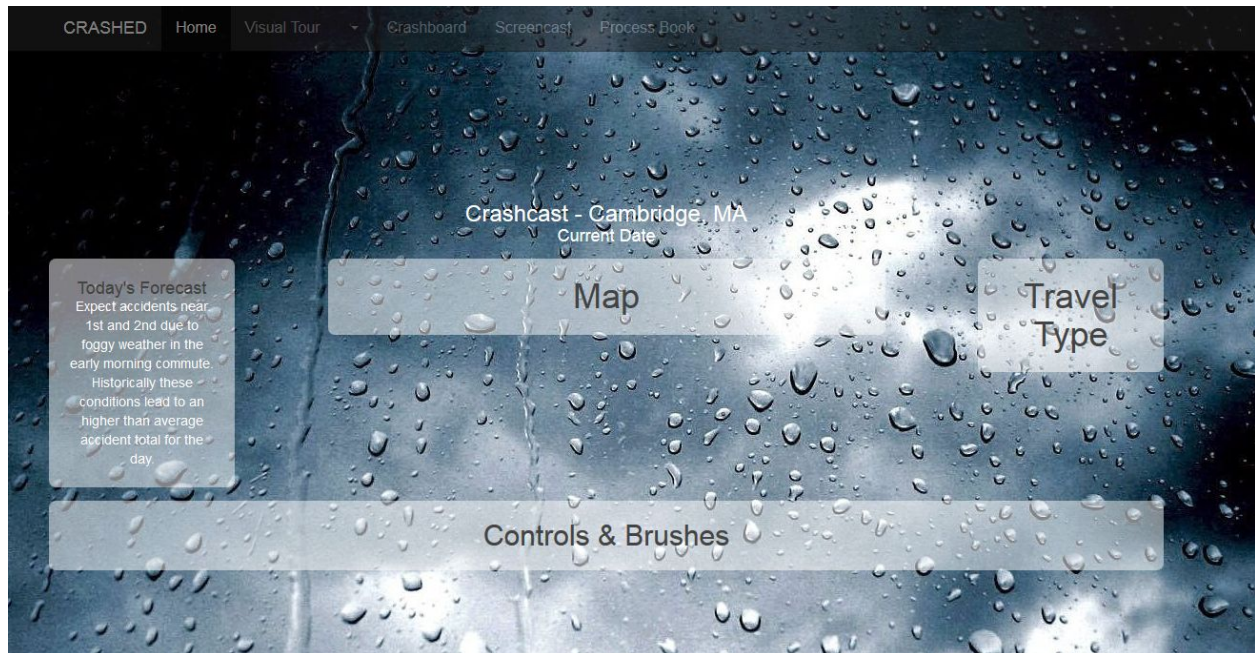
MAAT

MAAT
MAAT
MAAT

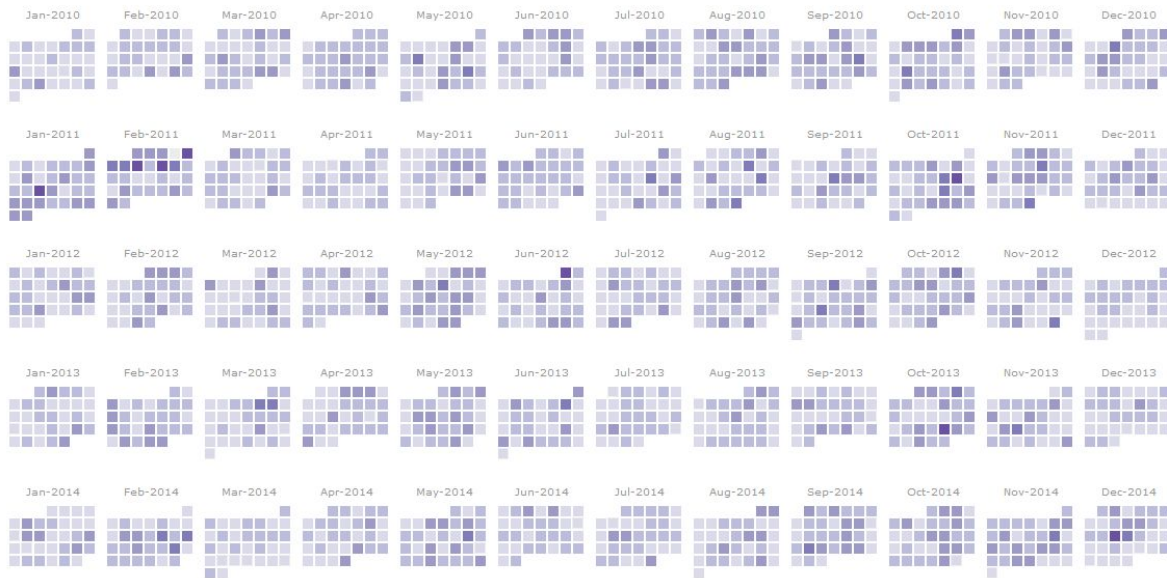
Chapter 5: Project prototype V1

No.	Visualization	URL
1	Linemap, forced layout of crossroads and accidents	http://harvardviz.github.io/crashed/vis-development/linemap.html
2	Heatmap of accidents over time	http://harvardviz.github.io/crashed/vis-development/heatmap.html
3	Geo Plot, Choropleth and Cross Filtering	http://harvardviz.github.io/crashed/vis-development/index.html
4	Sankey Diagram	http://harvardviz.github.io/crashed/index-sankey.html
5	Final Webpage (template)	http://harvardviz.github.io/crashed/index.html

Webpage Template - [link](#)

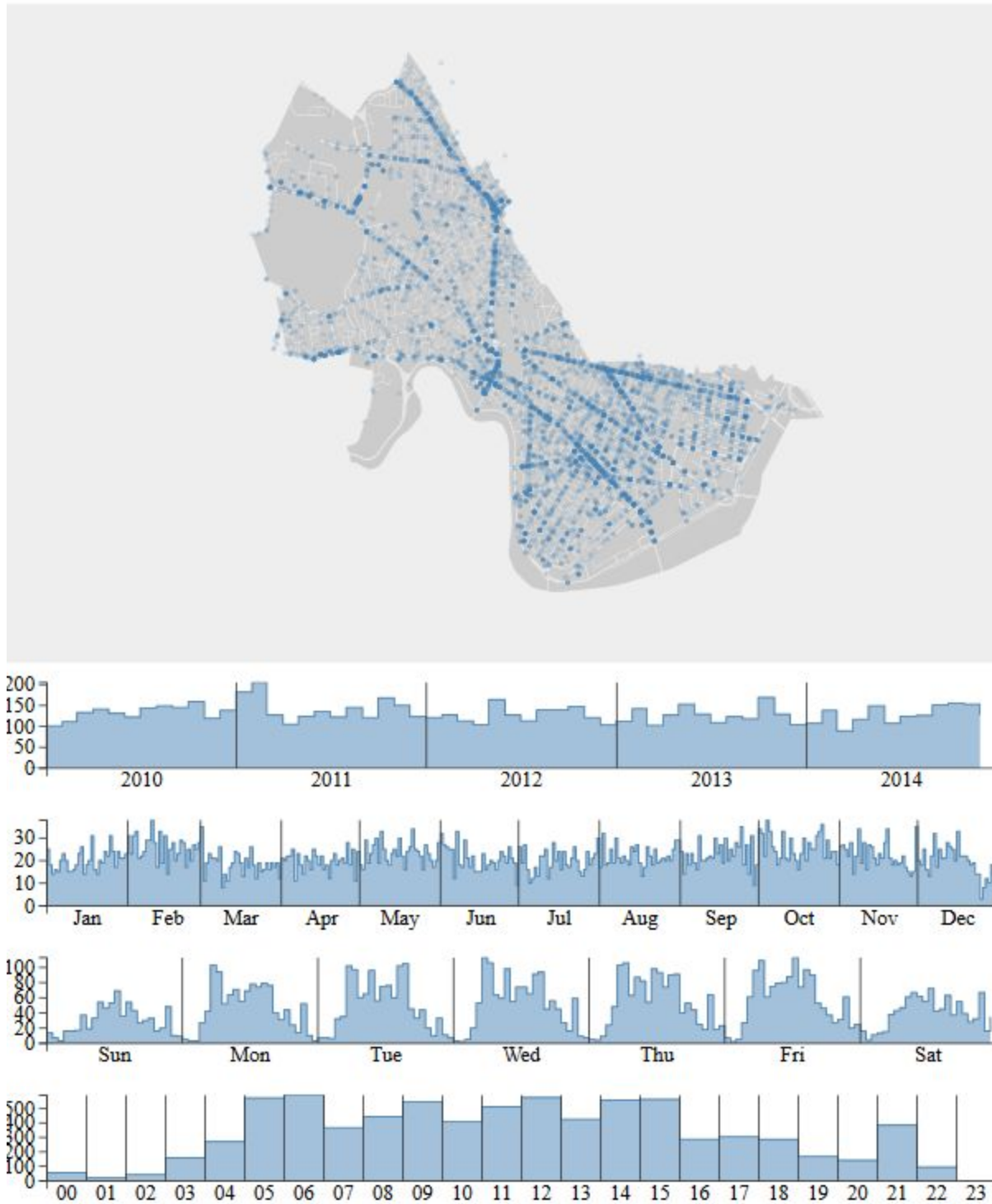


Heatmap of number of accidents - [link](#)



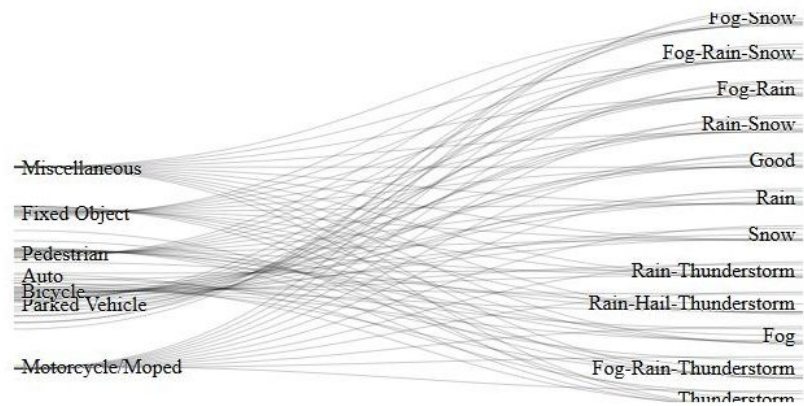
Used cal-heatmap.js library. Will add stories to specific highlighted dates. Currently has tooltip on rollover... could add more interactivity.

Geo plot of Accidents and Time Crossfilter - [link](#)

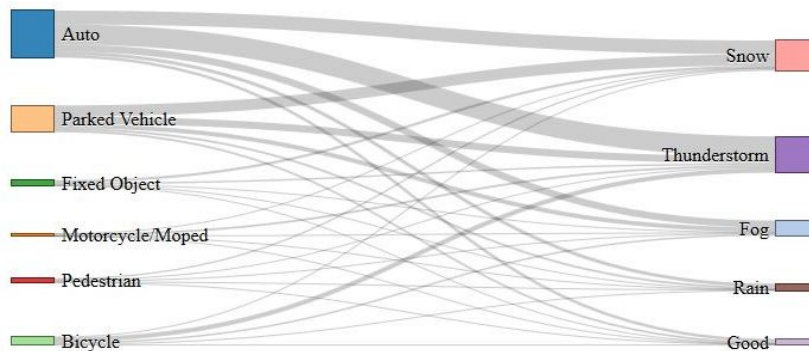


Sankey Diagram - [link](#)

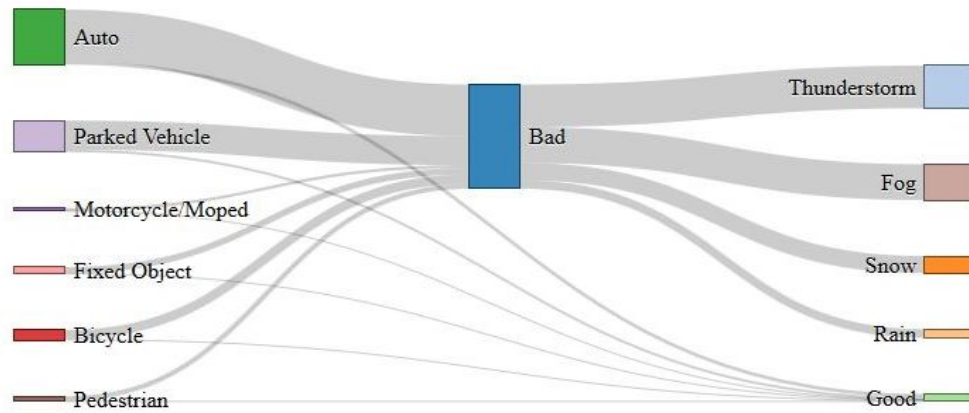
Iteration #1 - Accidents mapped to weather sankey diagram



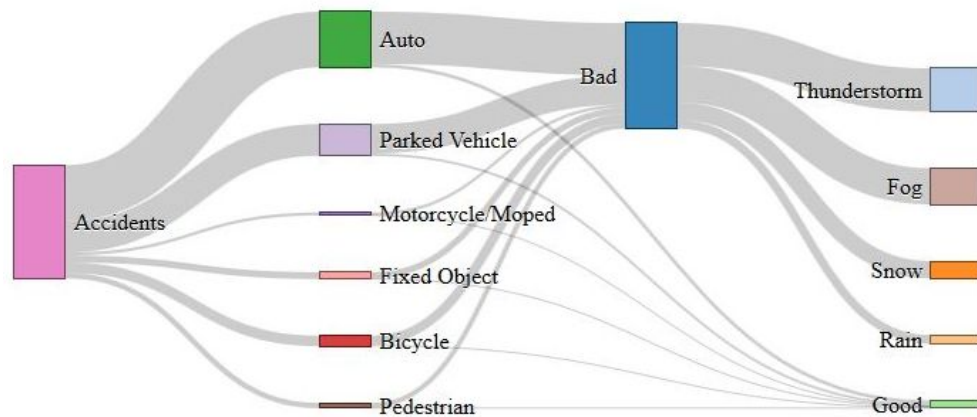
Iteration #2 - Consolidated categories



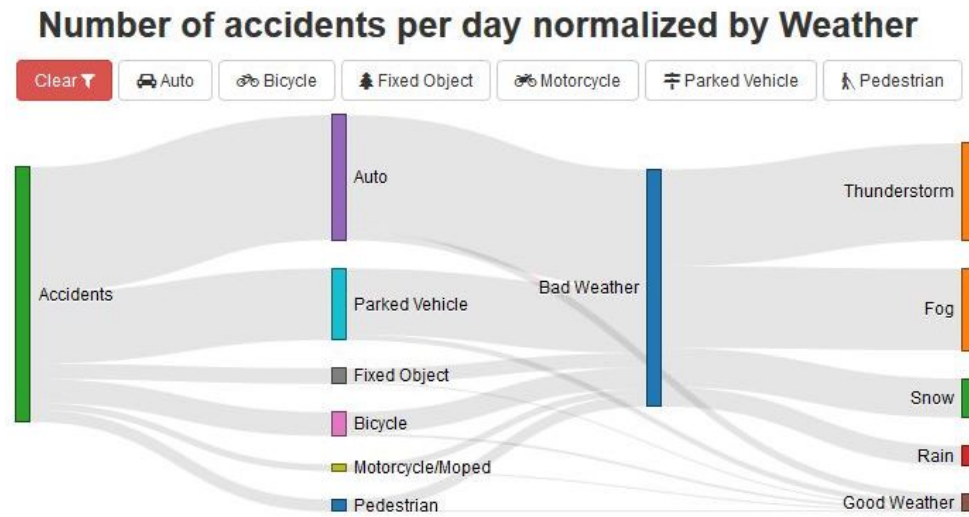
Iteration #3 - Aggregated bad weather category



Iteration #4 - Added accidents category




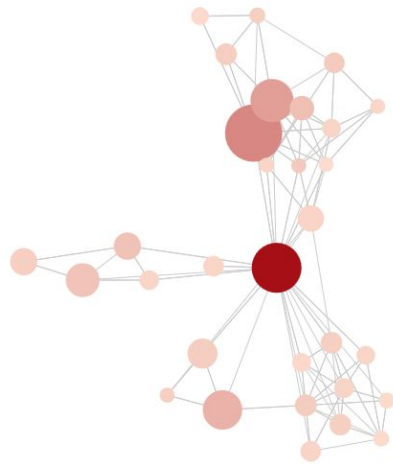
Iteration #5 - Final version with filter (**Note: filters are incomplete**)



Linemap and Forced Layout - [link](#)

Massachusetts Ave ▾

Westbound  Eastbound
Massachusetts Ave Intersections (Rollover circles for more detail. Click for street views)



Concept is that the “linemap” represents the order of intersections along a particular road. Force graph represents network of major roads and their intersecting connections. Clicking on a node updates the linemap.

Storytelling Mock up

Story title

The most dangerous intersection for pedestrians: 14 killed over 23 years!



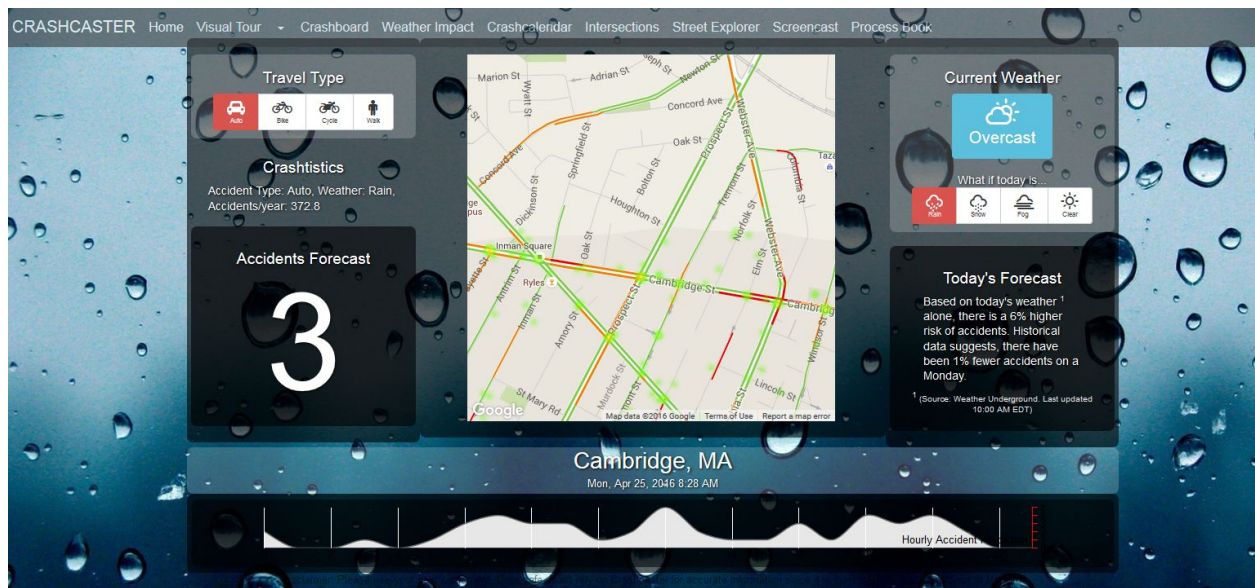
Map showing live traffic
and worst intersections –
corresponding to the
storyline

Here is some random text One person was killed and two more were injured when two cars collided in a Scott County, Iowa intersection. The collision happened in the intersection of Utica Ridge Road and 210th Street just after 6 p.m. on Wednesday, March 16, 2016, according to a statement from the Scott County Sheriff's Department. Thirty-four-year-old Lucas Johnson of rural Scott County was driving a Mercury Grand Marquis headed northbound on Utica Ridge Road when he was struck by a Pontiac Grand Prix headed eastbound on 210th Street. Police say the Grand Marquis ran the posted stop sign at the intersection. The two cars collided and ended up in a ditch in the northeast corner of the intersection, according to the statement. The Grand Marquis hit a utility pole. Johnson, the driver of the Grand Marquis, was pronounced dead at the scene, the statement said. Both the driver and the passenger of the Grand Prix, two 18-year-olds from Davenport, were taken to the hospital; the driver was in non-life-threatening condition and the passenger was in serious condition.

Chapter 6: Project prototype V2

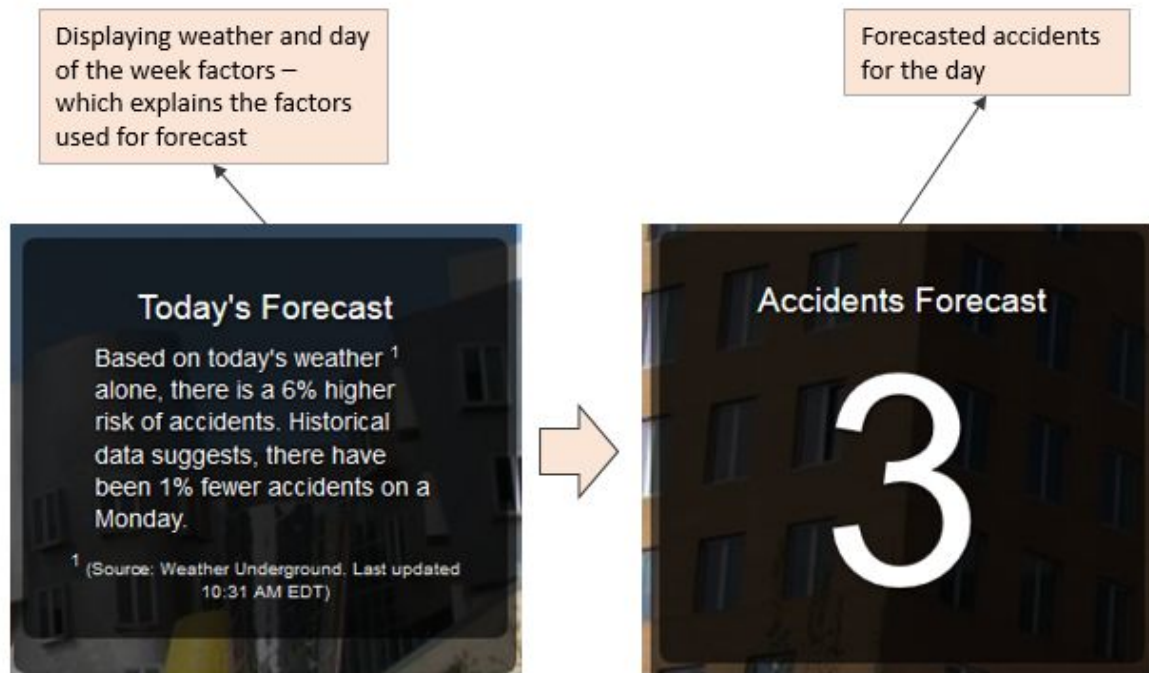
No.	Visualization	URL
1	Webpage - landing page	http://crashcaster.com/
2	Crashboard (former: Geo Plot, Choropleth and Cross Filtering)	http://crashcaster.com/#crashboard
3	Weather Impact (former: Sankey Diagram)	http://crashcaster.com/#sankey
4	Crashcalendar	http://crashcaster.com/#crashcalendar
5	Intersections (Former: Linemap)	http://crashcaster.com/#linemap
6	Street Explorer (former: Force Graph)	http://crashcaster.com/#forcegraph
7	Screencast (placeholder)	http://crashcaster.com/#screencast
8	Process Book (placeholder)	http://crashcaster.com/#processbook

Webpage Landing Page - [link](#)



Forecasting Model

A linear forecasting model is used to predict the forecasted accidents for the day based on historical data and weather information.



Forecast Formula

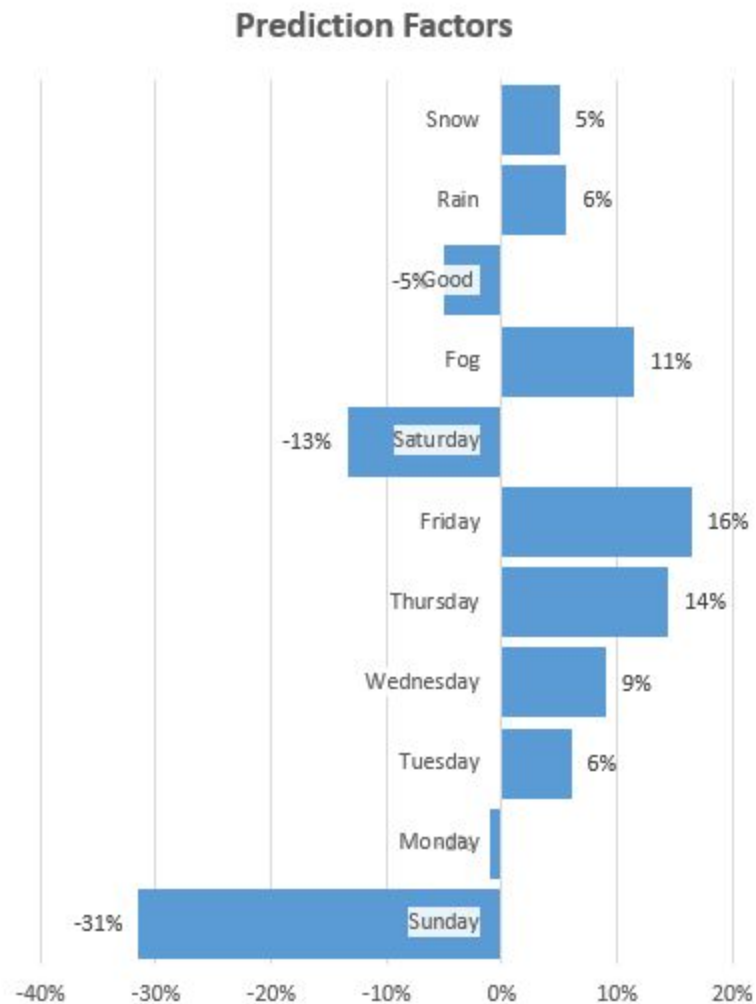
Forecasted Accidents = Roundup [Avg Accidents x (1+ Weather Factor) x (1 + Day Factor)]

Factors calculated based on correlation with weather data

Day/Weather	Fog	Good	Rain	Snow
Sunday	-24%	-35%	-28%	-28%
Monday	10%	-6%	5%	4%
Tuesday	18%	1%	12%	11%
Wednesday	22%	4%	15%	14%
Thursday	28%	9%	21%	20%
Friday	30%	11%	23%	22%
Saturday	-3%	-18%	-9%	-9%

Individual Prediction Factors for Weather and Day of Week

The calculated factors are now used to calculate the forecasted accidents.



These factors indicate that Friday is the riskiest day of the week and Fog is the worst weather condition.

Calculation of “Weather Factor”:

In the table below “Days” refers to the number of days of (Fog/Snow/Good/Rain) weather in the period under consideration.

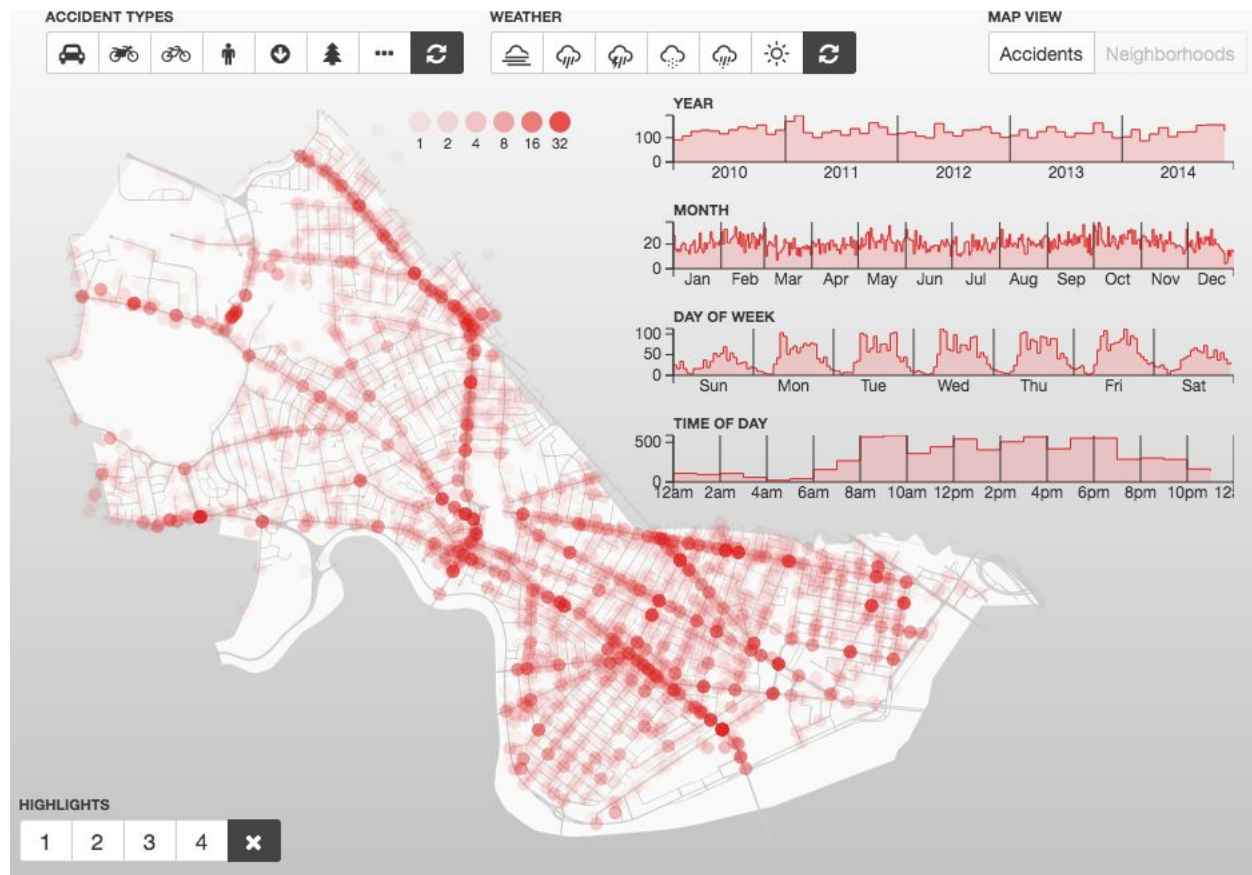
Weather	Days	# of Accidents	Average	Compared to average
Fog	144	692	4.81	11%
Good	1035	4241	4.10	-5%
Rain	508	2313	4.55	6%
Snow	137	620	4.53	5%

The weather correlated data reveals that the accidents were 11% higher than average during fog and 5% lower than average during clear weather.

Calculation of the “Day Factor”:

Row Labels	Fog	Rain	Snow	Good	Grand Total	Average Acciden	Day Factor
Sunday	53	213	57	387	710	2.95	-31%
Monday	75	272	95	610	1052	4.27	-1%
Tuesday	95	388	91	557	1131	4.57	6%
Wednesday	90	432	94	549	1165	4.70	9%
Thursday	125	341	95	665	1226	4.93	14%
Friday	133	322	105	688	1248	5.02	16%
Saturday	80	222	58	553	913	3.73	-13%
Unknown				1	1	distributed evenly for each day	
(blank)	41	123	25	231	420	distributed evenly for each day	
Grand Total	692	2313	620	4241	7866		

Crashboard - [link](#)



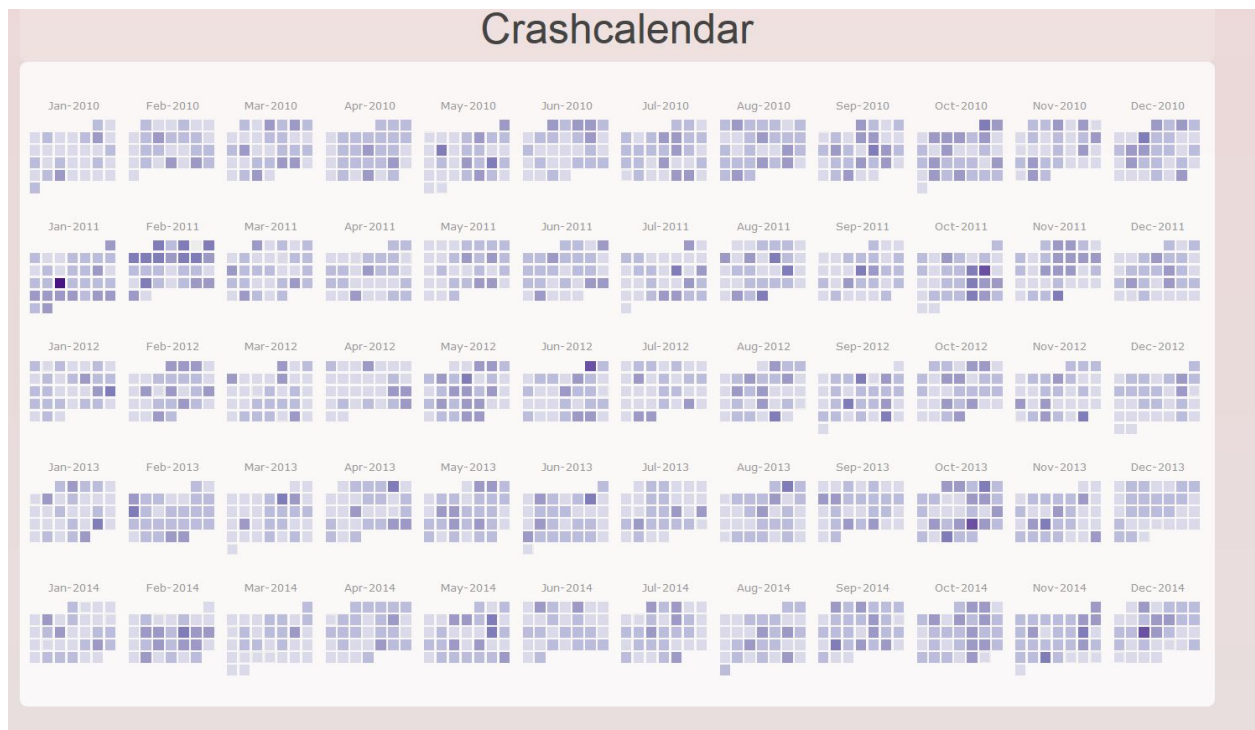
Concept:

- A high-level view of the entire dataset, exposing as many details as possible.
- Originated as a way for us, as a development team, to explore the data for interesting observations and turned into a tool to allow users of the site to do the same.

Data wrangling & implementation process:

- Data was pre-processed using Node.js for this visualization. A lot of this was just data cleaning and type conversion. Some things were for pre-processing, like neighborhood association for the choropleth (Neighborhoods) map, using D3's map features to check which geo feature accident coordinates fell within.
- [Crossfilter.js](#) was used to facilitate the enormous amount of crossfiltering that occurs in this visualization.
- A class-based approach for each chart and visualization was employed, communicating crossfilter updates using jQuery event emitters.

Crash Calendar:



Concept:

- Visually encode accident data by date for the entire dataset. Allow user to quickly identify dates with notably high/low, and provide contextual background as much as possible.

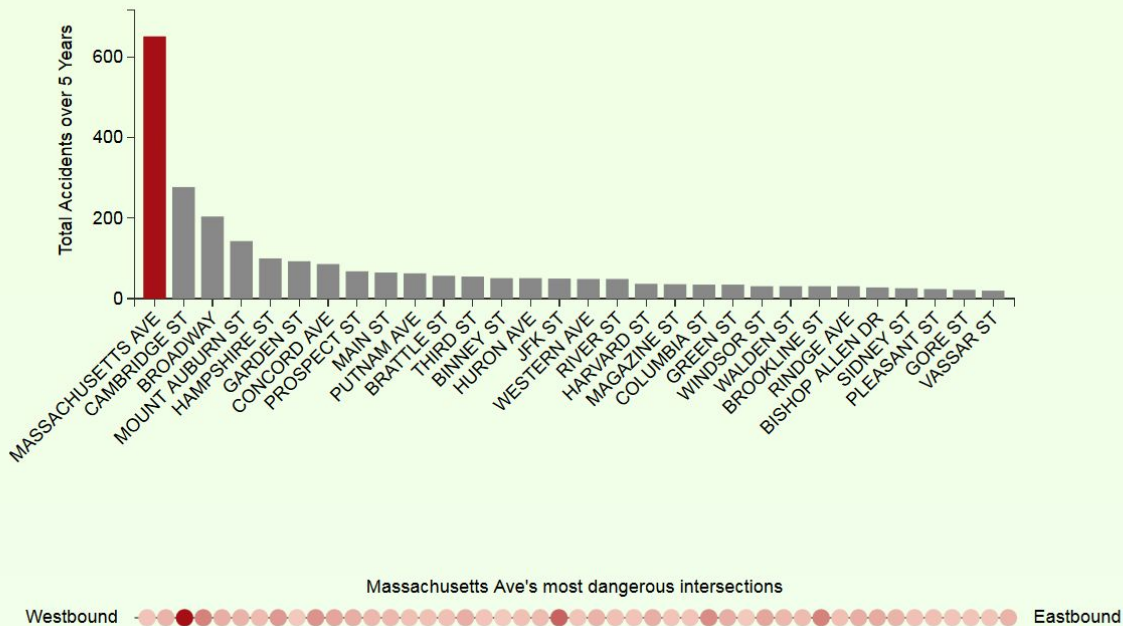
Data wrangling & implementation process:

- Cal-heatmap.js library was used to create this visualization. <http://cal-heatmap.com/>
- Aggregate accident data by date (D3.nest). First convert date object to string for property keys.
- Convert date to timestamp and convert from milliseconds to seconds format (required format for cal-heatmap.js library)
- Set cal-heatmap variable properties for layout, format, and range.
- Loop over each year of dataset to create 5 year calendar range.
- Add tooltips
- To be added: storytelling elements for noteworthy dates.

Intersections:

Accident Totals by Street and Intersection

Click on the bars in the chart below to see where along each road most accidents occur.



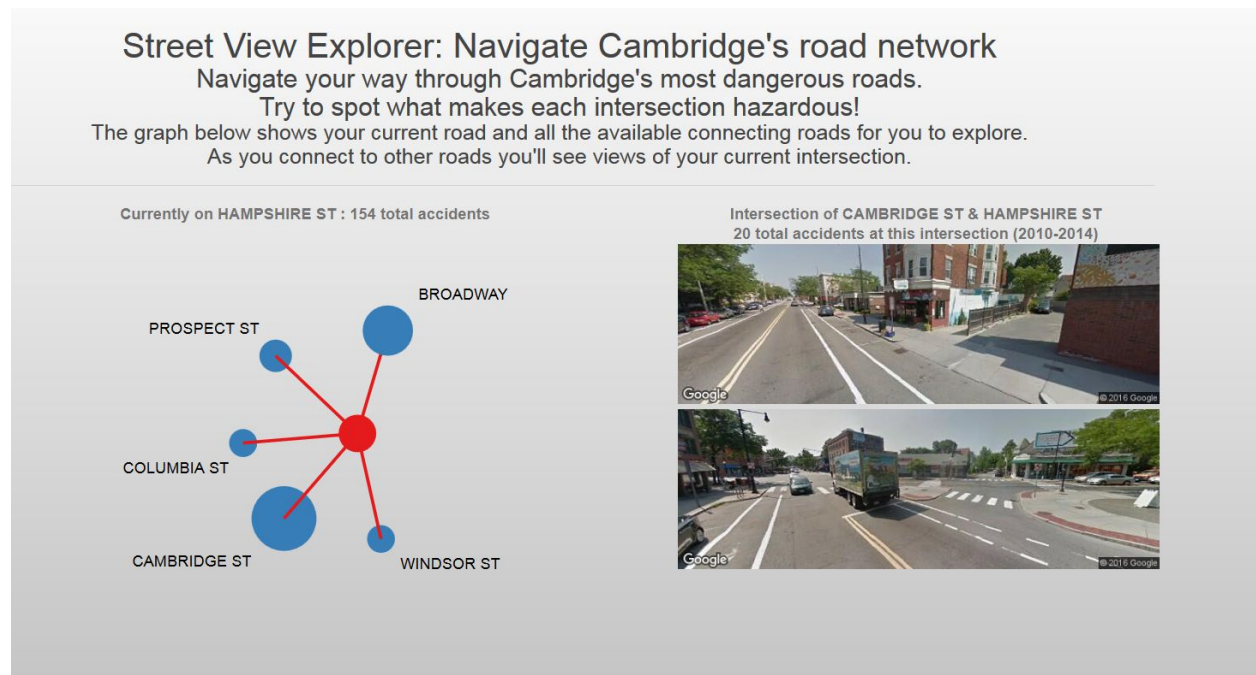
Concept:

- Provide a straightforward representation of which streets are the most dangerous, and where along each street accidents are most likely to occur.
- Bar chart added... force graph removed to increase effectiveness and decrease confusion.

Data wrangling and implementation process:

- aggregates accident data by each road, and then each intersecting road on level under that. (D3.nest)
- Calculates summary stats for each road, save as object property for each road.
- Calculates order of intersections along each road using coordinates function library (latlon-spherical.js & dms.js)
- Filters out accidents that did not occur at intersections
- Filters down to 30 highest accident roads
- Plots accident totals by road on bar chart
- Rollover bars highlights and shows tooltip
- Clicking bars updates linemap to show major intersections along that street (ordered west-east)

Street View Explorer:



Concept:

- Provide an innovative interactive visualization with high engagement factor.
- Represent the network of roads visually and non-geographically but without creating a hairball (by using a first person perspective for navigating the network).
- Intention is not to provide an encompassing overview of all the roads but to make the accident locations more personal, both by representing each location visually and by making the user feel like they are moving throughout the streets.
- Leave users with a “fun” last stop on our visualization website.

Data wrangling and implementation process:

- Utilizes similar data wrangling process as intersection visualizations.
- aggregates accident data by each road, and then each intersecting road on level under that. (D3.nest)
- Calculates summary stats for each road, save as object property for each road.
- Filters out accidents that did not occur at intersections
- Filters down to 30 highest accident roads
- Create force graph layout “links” array containing all “source-target” connections between node elements.
- Filter force graph array to a single road and it’s connecting intersections.
- Start force simulation, use enter/update/exit process for showing/removing nodes & links.

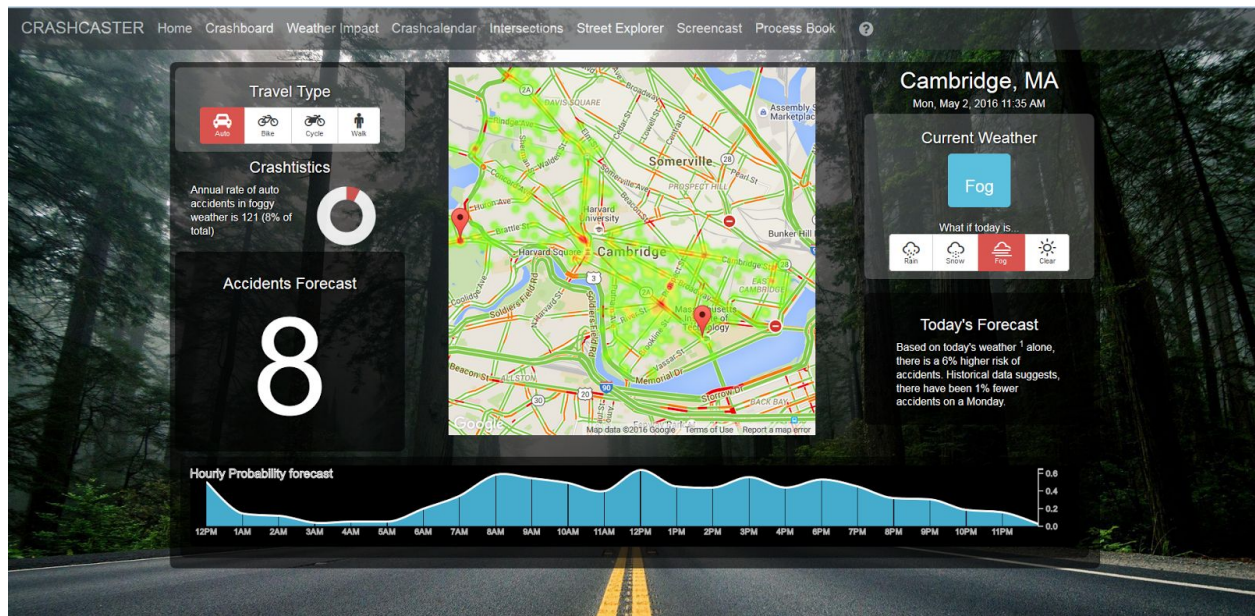
- When intersection node is clicked on, call google api requesting street view images at specified coordinates. (request images for 180 & 360 degree viewing directions. Use 180 field of view at -10 pitch downward to best show intersections)
- Update force graph input data to new selected road.

Chapter 7: Final Project V2

Overall Site updates:

- Site load/update improvements to address lag. Callback chaining,
- Dropped visual tour section for storytelling. Incorporated storytelling into each visualization page instead.
- Consistent color scheme for visualization accident encoding

Visualization updates:



Home screen forecast model:

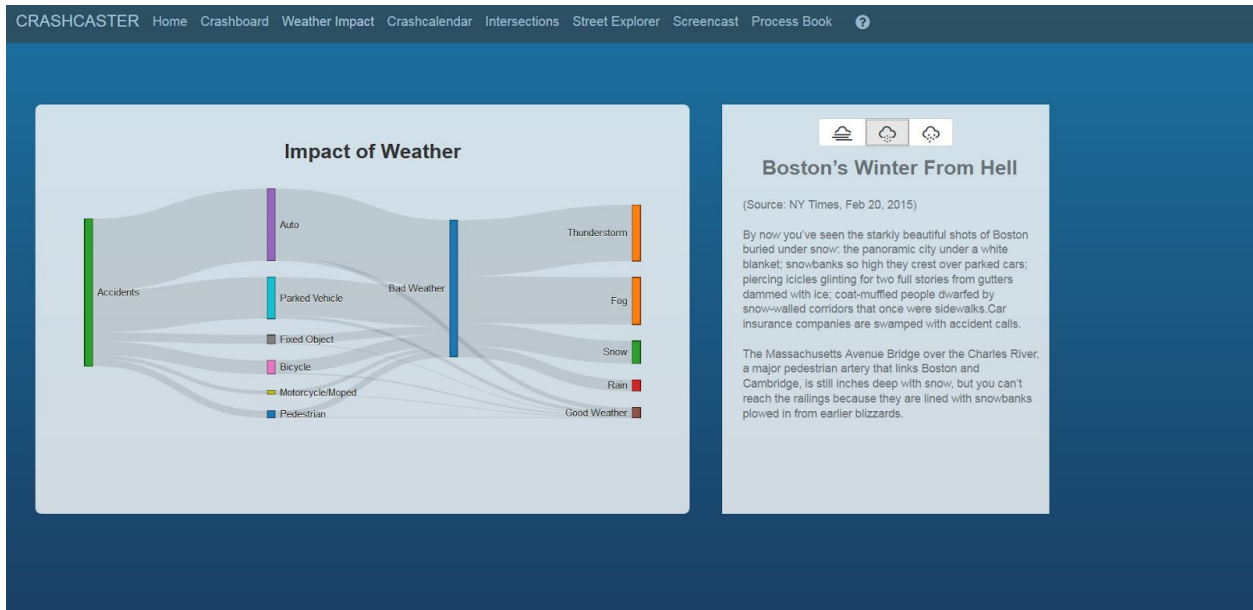
- Added Crashtistics donut visualization for representing percentages of total accidents recorded by accident-type/weather.
- Updated hourly forecast chart for improved visual appearance.

- Improved overall effective linking between visualizations in the forecast model.



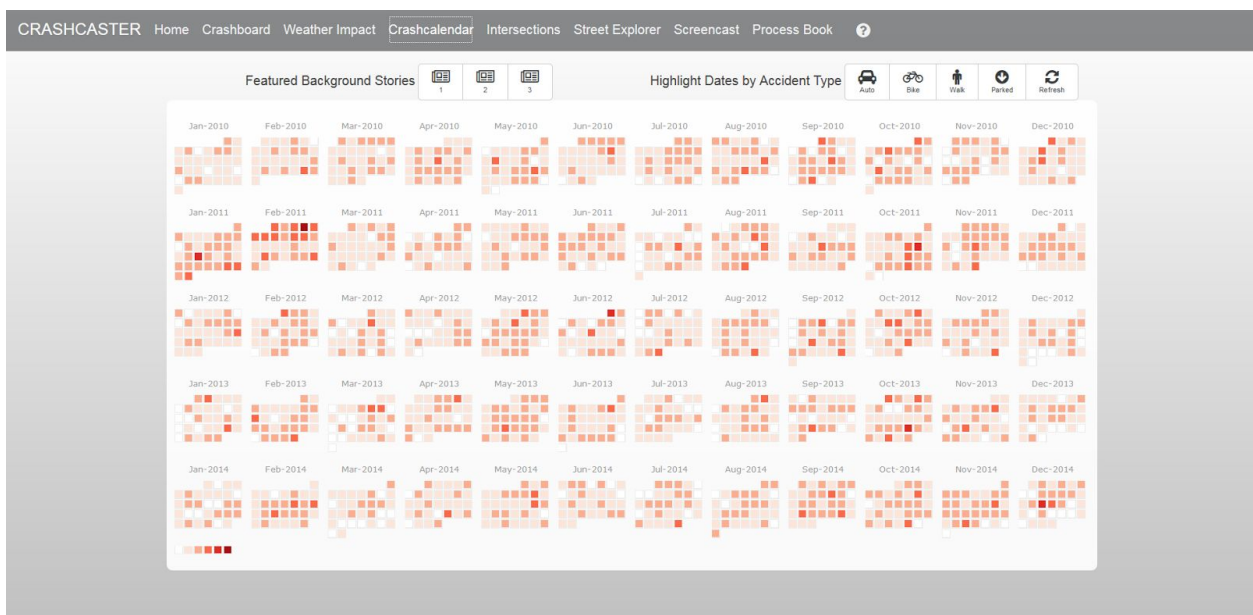
Crashboard:

- Changed color scheme to red with white map background for better pop-out effect.
- Improved legend accuracy
- Corrected time formatting
- Added description popup



Sankey - Weather Impact:

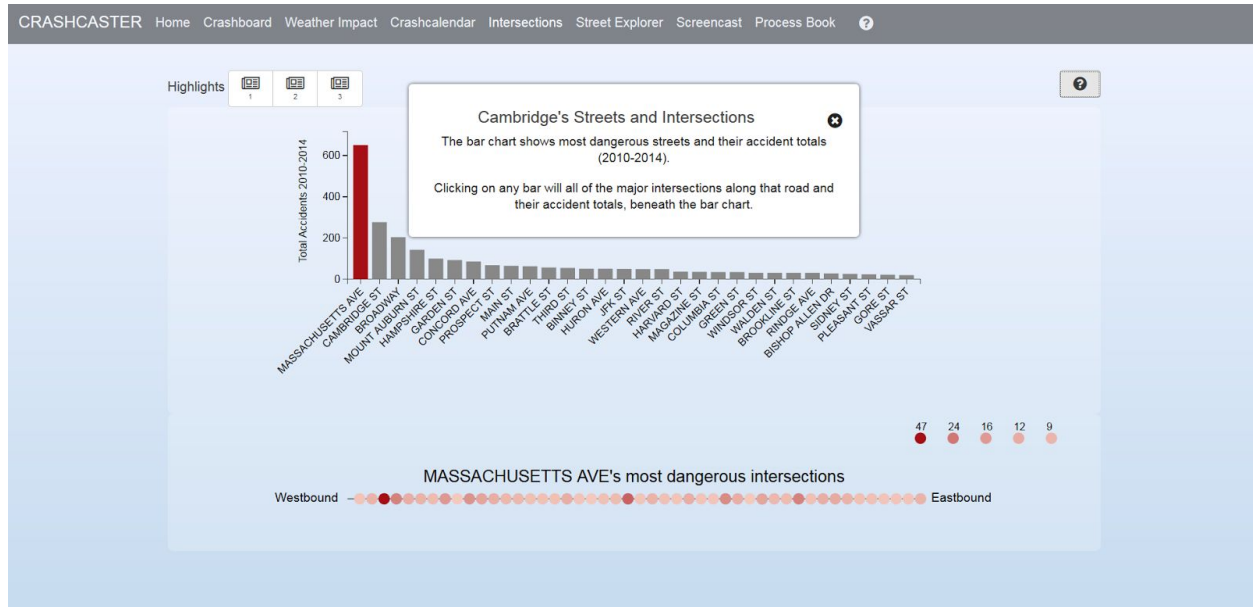
- Added storytelling feature with weather related content.
- Added visualization description



CrashCalendar:

- Added buttons for highlighting dates by accident types (that occurred on those dates)

- Added storytelling feature with content relevant to specific dates/date-ranges.
- Added legend
- Changed color scheme for site consistency



Intersections:

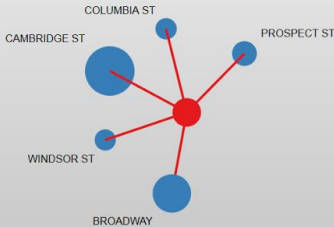
- Added storytelling with features highlighting specific intersections
- Added explanation of visualization popup info button
- Added legend
- Updated intersection visualization appearance
- Improved labeling

CRASHCASTER Home Crashboard Weather Impact Crashcalendar Intersections Street Explorer Screencast Process Book ?


Street View Explorer: Navigate Cambridge's road network

Navigate your way through Cambridge's most dangerous roads.
Can you spot something about these intersections that makes them hazardous?
The graph below shows your current road and all the available connecting roads for you to explore.
As you connect to other roads you'll see views of your current intersection.

Currently on HAMPSHIRE ST: 154 total accidents



Intersection of CAMBRIDGE ST & HAMPSHIRE ST
20 total accidents at this intersection (2010-2014)



Street View Explorer:

- Improved missing data handling

CRASHCASTER Home Crashboard Weather Impact Crashcalendar Intersections Street Explorer Screencast Process Book ?

What can we do to save more lives in Cambridge?

That was the motivating question for our CS171 Visualization project, Crashcaster. We found ourselves deeply interested in using innovation, particularly in visualization, to improve society beyond the bounds of our final project, well into the future.

The truth is that traffic accidents are one of the leading killers of Americans -- right up there with heart disease and cancer. Crashcaster is about the collisions between strangers' lives, the impacts upon them, where they happen most often, and how to avoid finding yourself at a serious, life-altering intersection on the streets of Cambridge.

Using visualizations to create engagement and understanding we thought could help with our goal. We start our story in the future with our forecast, "Crashcaster". We then invite you to explore the historic data we used to create our predictive model via our dashboard, "Crashboard". Finally, we unravel the stories in the data that we found interesting while enabling you to explore for yourself in what we hope are novel ways.

Stay safe on the streets of Cambridge!
Kyle Maguire, Rick Farmer, Kartik Trasi and Adam DiChiara

¹ (Source: Weather Underground. Last updated 10:26 AM EDT)

✕ Close

Added "About" section link on menu bar, with description of our project and overview of our visualizations.

Final Thoughts

Overview and Motivation:

“What can we do to save more lives in Cambridge?”

That was the motivating question for our CS171 Visualization project, **Crashcaster**.

We found ourselves deeply interested in using innovation, particularly in visualization, to improve society beyond the bounds of our final project, well into the future.

The truth is that traffic accidents are one of the leading killers of Americans -- right up there with heart disease and cancer. Crashcaster is about the collisions between strangers' lives, the impacts upon them, where they happen most often, and how to avoid finding yourself at a serious, life-altering intersection on the streets of Cambridge.

Using visualizations to create engagement and understanding we thought could help with our goal. We start our story in the future with our forecast, "Crashcaster". We then invite you to explore the historic data we used to create our predictive model via our dashboard, "Crashboard". Finally, we unravel the stories in the data that we found interesting while enabling you to explore for yourself in what we hope are novel ways.

Questions:

With Crashcaster, we tried to answer questions of what information would a resident or visitor to Cambridge need to know to increase their safety on the streets, and what is the best way to communicate that information to them visually. We envisioned the newcomer who is faced with all of Cambridge's quirks and nuances (that in some ways add to its character, but also can be a source of confusion and danger for those not familiar with local city knowledge) and thought about what information about accidents would be most important for them to know.

We found that location, timing, and weather related causes were the most critical factors that affected accident likelihood. We built our visualizations to explore these topics from multiple perspectives.

Then, taking things a step further we decided that in addition to presenting this information on historical accidents, we could use our knowledge to predict accidents in the presence based on a model that takes into account these factors.

Evaluation:

As with any project there is always room for improvement. Given more time our team might have further researched and refined our storytelling to include additional pieces of interest. A

feature we were not able to include due to lack of time included a zoom-and-pan capability on the CrashBoard map. (we considered Leaflet but instead opted for D3 for quicker deployment and greater aesthetic control). And our predictive forecast model, while broadly representative of the relationship between historic conditions and various present-day factors, is still a somewhat primitive model which could benefit from further refinement.

Overall we are happy with the result of the Crashcaster project and believe we have developed interesting and useful visualization tools for better understanding the accident situation in Cambridge. We were able to gain a greater understanding about the dynamics between Cambridge's busy roads and densely packed population centers, the differing activity patterns between the commuter and student populations, and the impact of weather in the region. We are satisfied with the belief that our site would be beneficial to both Cambridge newcomers and residents by alerting them to some of the dangers they face daily, but may not fully understand.