Assignment 5 – Dashboard design and Tableau stories

Professor John Sokol | Due 3/8

Background Information:

This week we put the puzzle pieces together. All the concepts learned thus far with Tableau data types, bar graphs, line graphs and calculated fields are combined to create a dashboard. This is a buzzword that you most likely heard at some point. The business world is *enamored* by dashboards. Why? Few static reporting tools emulate the interactivity and drill down capabilities of a dashboard, making dashboards an incredibly powerful decision-making tool.

- Canvas
- Formatting
- Filters
- Tooltips
- Geospatial Data
- Dashboard performance

For the dashboard demonstration, we'll be using the Long Beach Island dataset.

Open an empty dashboard tab:

Open a dashboard tab similar to opening a new worksheet. Click the **New Dashboard** icon at the bottom of the workbook, indicated by the box divided into four sections:

Canvas Selection:

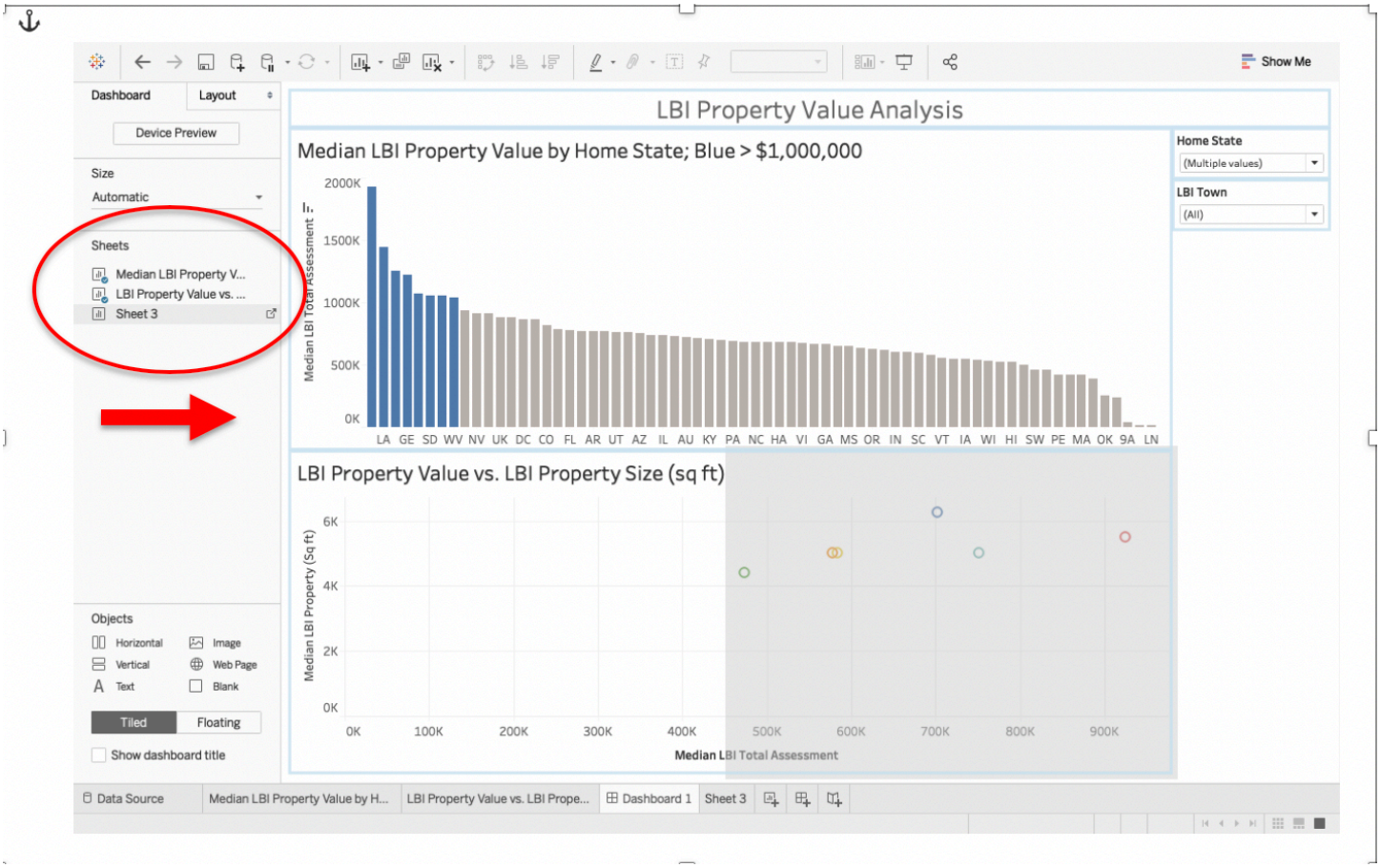
Go to the 'Size' tab on the left side of the dashboard screen. Change the canvas from 'fixed size' to automatic. With this option, the dashboard will resize to fit any screen it is displayed on.

Although Davis, the nice fellow in the other Tableau videos, suggests not to use the automatic canvas option due to apparent unreliability, I have not experienced this with dashboards. If no more than 3-4 worksheets are placed in a single dashboard tab at a time, then the automatic

option won't be an issue. There should NEVER be more than 4 worksheets views in a single dashboard tab, as this is when the automatic canvas option becomes problematic.

Adding Worksheet Visualizations:

Once a dashboard sheet is created, click the worksheet views you built (listed under **Sheets** to the left) and drag them to your dashboard sheet on the right. A gray, shaded area indicates where you can drop your visualization:



Tiled vs. Floating Objects:

Tiled:

- Tiled visualizations become part of a single-layer grid that resizes based on the dashboard size. Pairs well with the automatic canvas selection. Tiled visualizations do not overlap.

Floating:

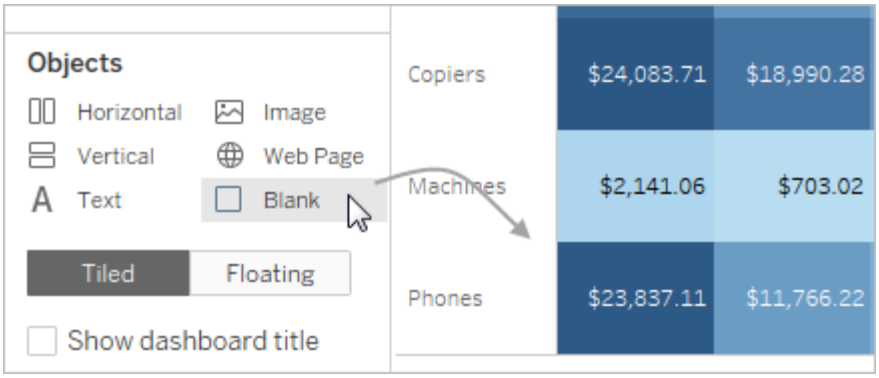
- Free-floating visualizations that can be layered over other objects. Equivalent to Microsoft Word Wrap Text: In front of text. In the example below, a map of the United States floats over tiled visualizations:

I believe using tiled object organization is the option for dashboard design, so this is how I organized my dashboard in my tutorial video.

Adding Objects:

In addition to adding worksheet visualizations to your dashboard, you can add web pages, text boxes, blank space, layout containers, and images.

To add an object, select an item under **Objects** on the bottom left, and drag it to the dashboard sheet on the right:



Fonts to Guide Analysis:

Use font to guide the Tableau viewers through a visualization, specifically font selection, size, and formatting.

Font selection

- Limit the number of fonts to one or two, and a second font only if the font selected for the words on the view does not look good when applied to the numbers on the view. Occasionally, use a secondary or even tertiary font if there is a special section on the dashboard where you want to call attention.

Font size

- Use a hierarchy with larger or smaller sizes to help denote where new sections begin and/or communicate the relative importance of sections. For example, the title font may be 18 point; section-headers 14 point; and annotations or tooltips 10 point.

Font format

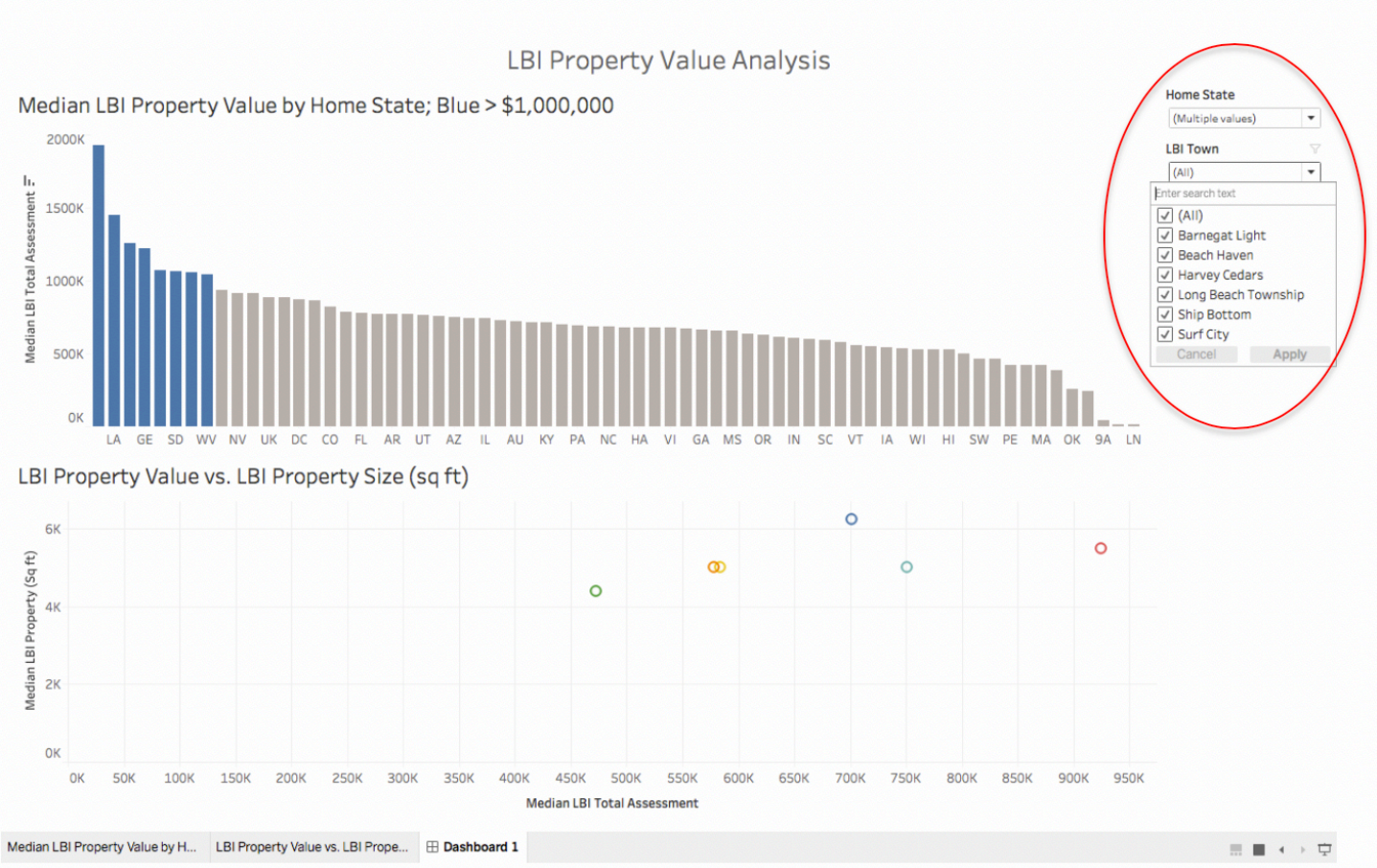
- Another way to explain how sections are broken up and can also be used to provide instructions. For example, I may bold titles and section headers, but use italic lettering to communicate that the user can use a filter or dashboard action. When you consistently

use the same formats as a subtle way to explain the user experience of your dashboards, your users will become conditioned to know how to use them.

-

Filters:

The power of a dashboard is the ability to set up filters and interactive components to change the data in the visualization to ultimately enhance your users' analysis. In a dashboard, a good rule to follow is to organize the filters as a list on the right side of the dashboard:



Formatting filters:

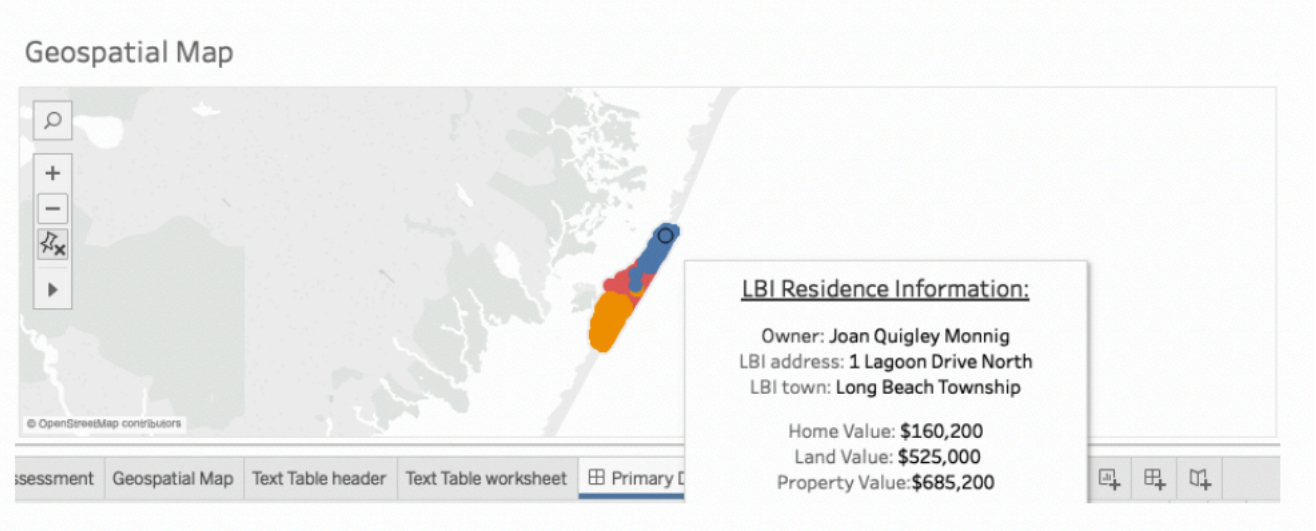
These are the recommended filter formatting options:

- Right click on filter tab > Apply to worksheets > **All using related data sources**
    - Allows the filter to affect all visualizations in the dashboard that use the data source

- Right click on filter tab > **Multiple Values (dropdown)**
    - User can select more than one filter condition
- Right click on filter tab > customize > **show apply button**
    - User can select several filter conditions before selecting to apply them.
    - Only available with more than 2 filter options

Tooltips:

- Details that appear when you rest the pointer over one or more marks in your visualization. Tooltips can include both dynamic (changing based upon the filtering conditions) and static (not changing) information.
- Additional tooltip information



Geospatial Data:

- Visualize location-based data, in this case the address of each home in the dataset.
- Need longitude and latitude values, the x and y axis components of the geographic coordinate system
    - Longitude: Specifies the east-west position of a point on the Earth's surface.
    - Latitude: Specifies the north-south position of a point on the Earth's surface.

- Columns: Longitude
- Rows: Latitude
- Drag the attribute you want to visualize to the detail selection in the Marks card. In the example below, bring the home addresses field to 'detail'.

Dashboard performance:

Performance is the speed with which you can interact and work in Tableau. The volume of the datasets I provide are small compared to big data warehouses companies and organizations use to store petabytes of proprietary data and hundreds of millions of records. Loading datasets of this size into Tableau can cause the dashboard to lag significantly. When working with big data, dashboard performance should be optimized by adhering to several performance practices.

Work with extracts instead of live data

- You can choose between a live or extract connection on the data source page.
  A *live* connection is a direct connection to your data. A Tableau data *extract* is a compressed snapshot of data stored locally into memory as required to render a Tableau visualization. Extracts are designed to use all parts of your computer's memory optimally.
- There are several reasons to use an extract, but the main performance-related reason is if your query execution is slow. The extract data format is designed to provide a fast response to analytic queries. In this case, you can think of the extract as a query acceleration cache.
- For text files (CSV, Microsoft Excel spreadsheets), it is best practice to use extracts because the dataset is not continuously updating with new data in the background like a Tableau SQL server connection.

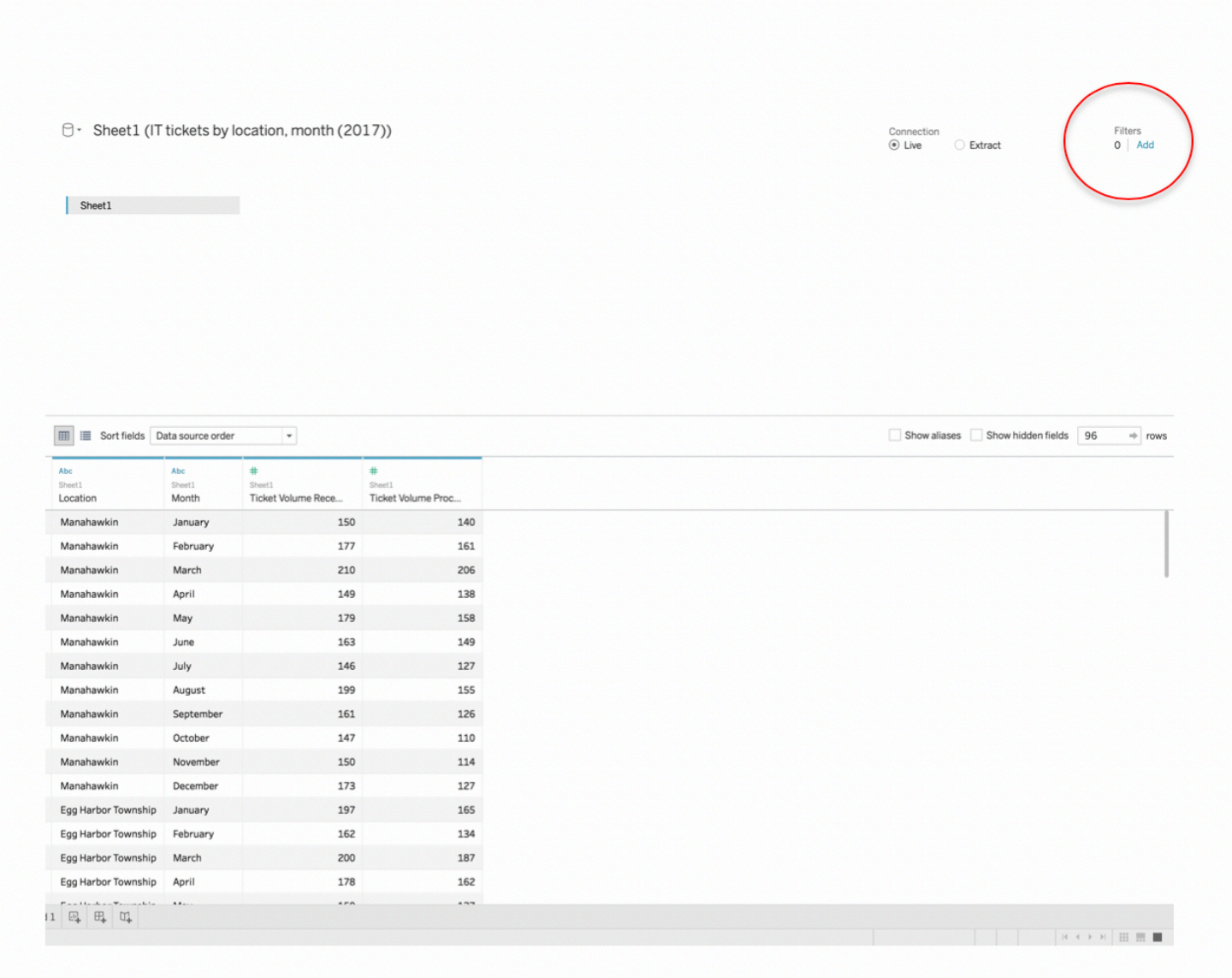
Hide unused fields

- Hidden fields are not included when you create an extract. Use the **Hide All Unused Fields** option to hide unnecessary fields before you create an extract. This makes the extract smaller, which improves performance.

Filter your data carefully: Filters provide enormous flexibility, but you should use them carefully. They can be computationally expensive, so reducing the number of filters can improve workbook performance.

Filter your data in the data source

- When you create a filter from within the data source tab, you reduce the amount of data in the data source, which may yield tremendous control over the performance of queries issued by Tableau.



Location	Month	Ticket Volume Rece...	Ticket Volume Proc...
Manahawkin	January	150	140
Manahawkin	February	177	161
Manahawkin	March	210	206
Manahawkin	April	149	138
Manahawkin	May	179	158
Manahawkin	June	163	149
Manahawkin	July	146	127
Manahawkin	August	199	155
Manahawkin	September	161	126
Manahawkin	October	147	110
Manahawkin	November	150	114
Manahawkin	December	173	127
Egg Harbor Township	January	197	165
Egg Harbor Township	February	162	134
Egg Harbor Township	March	200	187
Egg Harbor Township	April	178	162

Keep the workbook to a reasonable size

- The fewer worksheets and data sources in a workbook, the faster the workbook will perform. If there is a big topic to explore and you are tempted to put everything into a single monolithic workbook, reconsider. If performance starts to suffer as you enlarge the scope of your workbook, think about breaking your workbook into separate files. Be strategic when designing your visualization—the fewer sheets and data sources, the faster your visualization will perform.

Limit the number of filters you show in the view

- Filters that you show in your view are called quick filters. When you add an interactive filter to a view, each filter requires a query in order to populate the options. If you add a lot of interactive filters to your dashboard, it can cause the dashboard to take a long time to render.
- When you use "Only relevant values" on a filter, it requires a query to update the shown values each time other filters are changed. Use this feature sparingly.

Create efficient calculations

- Use basic and aggregate calculations, as these scale very well.
- Booleans and integers perform faster than strings. If your calculation produces a binary result (for example, yes/no, pass/fail, over/under), be sure to return a Boolean result rather than a string.
- Convert date fields. Users often have date data that is not stored in native date formats—for example, a date might be a string or a numeric timestamp. You can use the DATEPARSE function if your data supports it. If the originating data is a numeric field, converting it first to a string and then to a date is very inefficient. It is much better to keep the data as numeric and use DATEADD and date literal values to perform the calculation. The performance gains can be significant with large data sets.
- Miscellaneous calculation tips:

- Distinct counting values is one of the slowest aggregation types in almost all data sources. Use the COUNTD aggregation sparingly.
- Remember that all basic calculations are passed through to the underlying data—even literal calculations like label strings. If you need to create labels (for example, for column headers) and your data is very large, create a simple text/Excel file data source with just one record to hold them so they don't add overhead on the big data source.

Tableau Stories:

A **story** is a sequence of visualizations that work together to convey information. You can create stories to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or simply make a compelling case with your data. A story is a sheet, so the methods to create, name, and manage worksheets and dashboards also apply to stories. Each individual sheet in a story is called a **story point**. The rationale for using Tableau stories is they are used to highlight the important parts of your data that you want to show to an audience. These emphasized points should answer your research question.

Deliverables:

In a **single packaged Tableau workbook**

Tableau Dashboards:

Two dashboard tabs:

- A single dashboard tab of the 2 LBI worksheets completed in class and also a Blackboard instructional video
    - 2 worksheet visualization view
        - Bar graph of average LBI property assessment by home state
        - Geospatial view of LBI addresses
    - 3 Filters on the right side of the dashboard
        - LBI Town
        - Home state
        - LBI last day of sale
- A single dashboard tab that visualizes the Walmart dataset:
    - 3 worksheet visualization views
        - 1 line graph (top left of the dashboard)
        - 1 bar graph (top right of the dashboard)
        - 1 geospatial data map (bottom of the dashboard)
    - Minimum 3 regular filters on the right side of the dashboard

Tableau parameters

- For the Walmart bar graph created above:
    - Add color to the bar graph based on an arbitrary threshold value
        - Create a parameter target reference line
        - Insert this parameter into the color calculated field
    - Place the bar graph in a dashboard view
    - Show the parameter controls in the dashboard view