

# Design: Slice card Text area - Styling and Auto Text

JIRA ticket link

<https://juiceanalytics.atlassian.net/browse/JB-2983>

## Design brief

### Summary

This design effort addresses a few things specifically around the Text ux in a Slice card:

- Currently markdown is used to apply limited styling. We need to decide the different styling requirements we really want to have and how to implement them in a UI, as well as potentially a markdown format option.
- Dynamic text improvements - we want to improve the ease of use around dynamic text, and decide what options should be possible there.
- There is some [past work](#) and design around Auto Titles but a few interaction details need to be reconsidered there. I think there can be some simplification with it.

This ticket should have some kind of prototype to do some design testing out of it.

### Key Questions

- How much should be possible via **typing** vs. UI?
- What text formatting options are needed to achieve **styling possibility goals** of a story?
- What sort of dynamic title things should we prioritize for ease of use? What, if any, advanced **dunderscore** things still be possible?
- How in sync can the **Auto text** be with the vis?
- How could a Narrative Science plugin play into this? (🚀 future)
- Do we want to allow **html** / embedding in here?
- How to keep the editor UI clean and easy to read, while still allowing markdown.

**Brand new or an improvement to related existing features?**

- We have some markdown ability in the Slice text area
- It also accepts any html pretty much.

**Specific functionality we are confident we want to support**

- Basic text styles
- typing→formatting options
- Some sort of text alignment control

**Tactics to complete this design effort**

- sketching / brainstorming
- mockup up desired story design options. Already started a bit the in the [1 Experimental App](#).

**Markdown / UI formatting to support**

- Heading 1, Heading 2, Heading 3
- Bold
- Italic
- Link
  - I love how in Paper you can highlight text and then paste links directly into it if a url was copied to the clipboard.
- Bullet list
- Underline
- Note text
- Blockquote
- Horizontal Rule
- respect line breaks

**Other**

- author `//comments` ?
- Link to anchor
  - Make every heading an anchor like Paper?
  - Or every slice
  - [Heading IDs](#) are a markdown thing too

In some cases **colored text** can be especially useful for legend purposes

- perhaps this can just be done with Auto Titles.

# Progress Notes

This is space is for showing thoughts and process / iterations along the way.

## April 29, 2020

- Want to get started on this especially since it relates partly to the other [+Design: JBO Typography Styles improvements](#) design ticket.

### Prioritizing dynamic text

1. @DataCardA chooser pill reference
  - a. "Let's look at **Countries** by **Measure 1**."
2. @Bars1 pill reference
  - a. "For those **12 Countries**, lets compare performance leaders..."
  - b. Works for current slice or others.
3. Reference the current slice
  - a. "You are looking at **12 Countries** by **Measure 1**."
4. Auto Titles

### Other dynamic text

- Specific values that aren't a slice selection reference
  - Measure values
  - Counts of dimensions

### Ideas

- What if any ingredient used previously in config before the current slice, could be referenced for certain values?
  - `@MeasureA.value` — adds to the response the calculation for that measure within the automatic filters of the current slice.
  - `@Regions.count` — gets a unique count of Regions
- What if they could be chained??
  - `@Regions.@MeasureA.value` — get the calculation for Regions within the current slice context.
- Thinking brackets may be a better approach regardless
  - `{{@DataCardsA}}`
  - `{{@DataCardsA.MeasureA}}`

→ Thinking about if the section config for horizontal layouts is really worth it right now. Seems to have more problems than uses at this point.

April 30, 2020

## Dynamic Text

### Phase 1

Repurpose/Support much of the [existing dunderscore](#) syntax in slightly simpler form.

- “Selection” is the default thing to reference, so we assume that.
- **Emphasis** is the default, so we enforce that where needed.

Reference another slice to retrieve its info in specific ways.

Given a Slice card @BarsA, by Regions and Population, *where no Regions are selected*:

(guessed at right dunderscore approach is existing)

- `{{@BarsA}}` — count and label as bold text.
  - `selectionDisplay()` → **34 Regions**
- `{{@BarsA|pill}}` — slice pill
  - `embedSlicePill()` → **34 Regions** (pill)
- `{{@BarsA|noemphasis}}` — count and label, unbolded.
  - `selectionDisplay()` → 34 Regions
- `{{@BarsA.label|noemphasis}}` — label, unbolded
  - ? → Regions
- `{{@BarsA.label|noemphasis|singular}}` — label, singular, unbolded
  - ? → Region
- `{{@BarsA.number}}` — label, count, bold
  - `selectionCount()|emphasis` → **34**

Can still reference nonselected info. Given **12 Regions** are selected,

- `{{@BarsA.dataCount}}` `dataCount()|emphasis` → **34**
- `{{@BarsA.dataCount}}` `dataCount()` → 34

Make it possible to reference a value from the selected Measure

- `{{@DataCardsA.number}}` — in a Measure data card, get the value of the selected measure, bolded
  - ?? → **38.2k**

Actually, just using `@BarsA` may be sufficient? Only reason not to might be for social @ mentioning purposes, but don't expect plans for that in this context.

## Phase 2

- Expand to allow for my filter context sensitive dynamic text
  - Measure calculations
  - Easier more accurate counts on Dimensions

## Styling x Markdown and Keyboard shortcuts

Options Under consideration

## Formatting

<b>Bold</b>	⌘+b
<i>Italic</i>	⌘+i
<u>Underline</u>	⌘+u
Link	⌘+k / Also, if link on clipboard ⌘+v to quick add

## Insert

Refer to data (@Data Card...)	@
Emoji 😊	:

## Markdown formatting

<b>Large header</b>	# Space	<h1>
---------------------	---------	------

Medium header	## Space	<h2>
Small header	### Space	<b>
1. Numbered list	1. Space	<ol>
• Bulleted list	* Space	<ul>
• <i>Blockquote</i>	> Space	<blockquote>
<i>Italic</i>	<u>italic</u>	<i>
<b>Bold</b>	<b>bold</b>	<b>
Code block	```	
code	`code`	
Divider	---	<hr>

## Other

Undo	⌘+z
Redo	⌘+Shift+z

Collapse Editor	ESC
Present	⌘+Shift+p
Keyboard shortcuts	⌘+ /

## Typing

### How should typing work for dynamic text?

- Entering “@” opens searchable typeahead for Slice names. Similar to Paper here, if you don’t select an appearing option that nothing gets linked.

- After selecting a Slice, can continue typing for more settings, or space to continue on.
- Maybe style these bits to look more like code, similar to `//comments`
- Come up with way to keep dynamic text short if possible. Maybe when collapsed it is simplified, and when focused, you see all the detail. ★

### How should typing work for markdown?

- Similar to other apps, can enter the markdown while typing, ie **bold text...**
- Can also use keyboard shortcuts, like `⌘-b` for **bold**

---

### All Styling Options

- Bold
- Italic
- Underline
- H1
- H2
- Link
- Bullet
- Blockquote
- Center-align?

### How to style

- Some styling options are made super easy via a UI popup.
  - Bold
  - Italic
  - H1
  - H2
  - Link
  - Bullet
- Markdown gives access to all UI options in addition to others
  - Blockquote
  - Underline
- Some can be done via keyboard shortcuts
  - Bold
  - Italic
  - Underline
  - Link

## More things

- emoji
- anchors

Style	UI	Keyboard command	Markdown
Header 1	h1		#
Header 2	h2		##
Body bold	b		###
Body			####
Note			#####
Bold	<b>b</b>	⌘-b	<b>*bold*</b>
Italic	<i>i</i>	⌘-i	<i>_italic_</i>
Underline		⌘-u	
Link	icon	⌘-k	[I'm an inline-style link] ( <a href="https://www.google.com">https://www.google.com</a> )
Bullet list	icon		*
Numbered list			1.
Blockquote			>
Horizontal rule			---
Left align text / Right align(?)	icon toggle		
Emoji		:	



## UI Popup via highlighted text

- Highlight some text to show a popup with format options
- Style text in the card to match the style

## Typing Markdown

- 

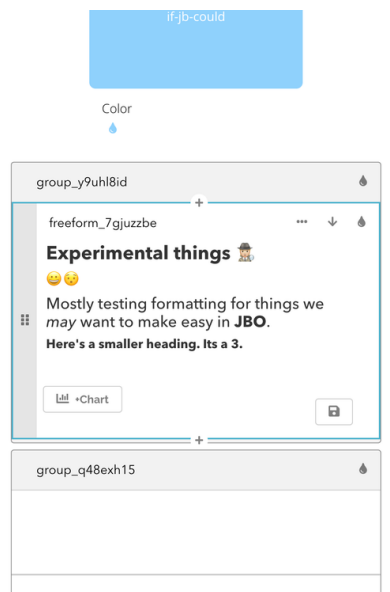
## Keyboard Shortcuts

- 

Html in the actual slice-title template turns more into something like this...

```
1 <div class="tl">
2 <h1>Experimental things 🧑🏻🔧</h1>
3
4 <h2> 😊🤔</h2>
5
6 <h2 class="fw2">Mostly testing formatting for things we <i>may</i>
  > want to make easy in <b>JB0</b>.</h2>
7
8 <h3>Here's a smaller heading. Its a 3.</h3>
9 </div>
```

While the Editor actually looks more like this...



## Experimental things 🧑

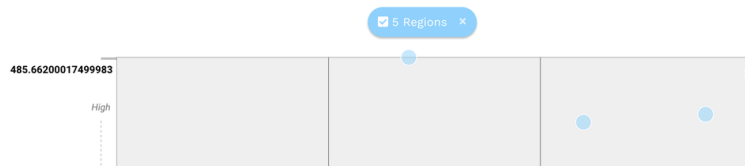


Mostly testing formatting for things we *may* want to make easy in **JBO**.

**Here's a smaller heading. Its a 3.**

### Regions by Happiness and Health

Curious if we just want to make titles left aligned by default. Definitely gives it more of a story/document feel automatically.



html for 👉 slice card

```

1 <div class="f3 b mb2">Experimental things 🧑</div>
2 <div class="f4 mb2">😊😬</div>
3 <div class="f4 mb2">Mostly testing formatting for things we <i>ma
y</i> want to make easy in <b>JBO</b>.</div>
4 <div class="f5 b">Here's a smaller heading. Its a 3.</div>

```

## Auto Title

Auto Text is text automatically added to the slice title based on the current visualization configured in that slice. It helps the author with good starting points for a description story.

- {{AutoTitle}}?
- {{Here are 🧑 **Players** as bars ranked by # **Measure A**, from greatest to least.}}

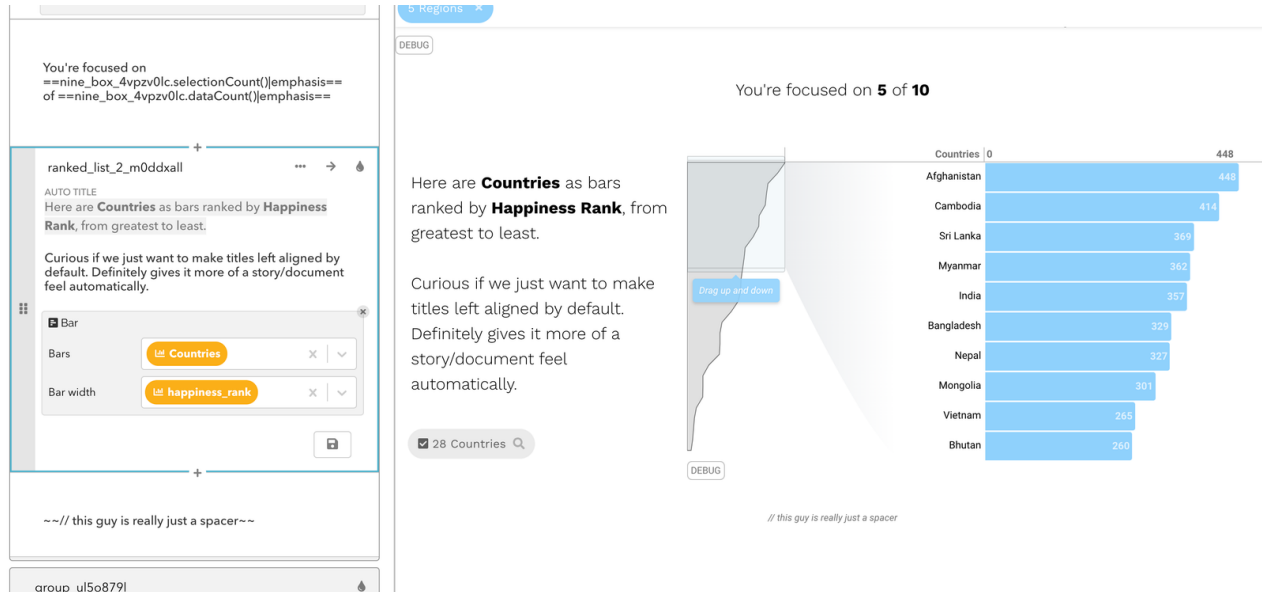
Auto Titles will need specific reference to vis config things that we otherwise don't have dunderstore templates for.

- Someday it would be cool to have more specific reference ability, like @BarsA.bar to retrieve "Players".
- If we had that we could also show the Auto Title with references when the Slice card is open.

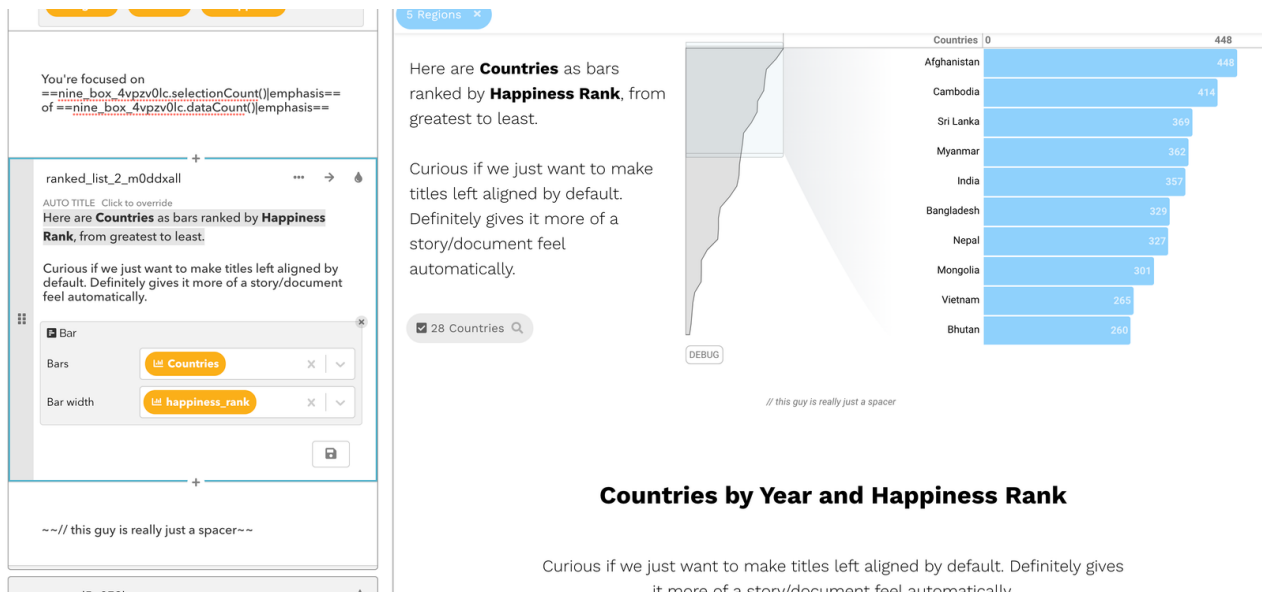
- {{Here are @BarsA.bar as bars ranked by @BarsA.measure, from greatest to least.}}
- Actually, it seems *possible* if we wanted to do this on the frontend, passing “Players” under the hood to the template, but not sure that’d good practice.

### Slice card focused

- Inserted text is styled differently and gets “AUTO TITLE” header.



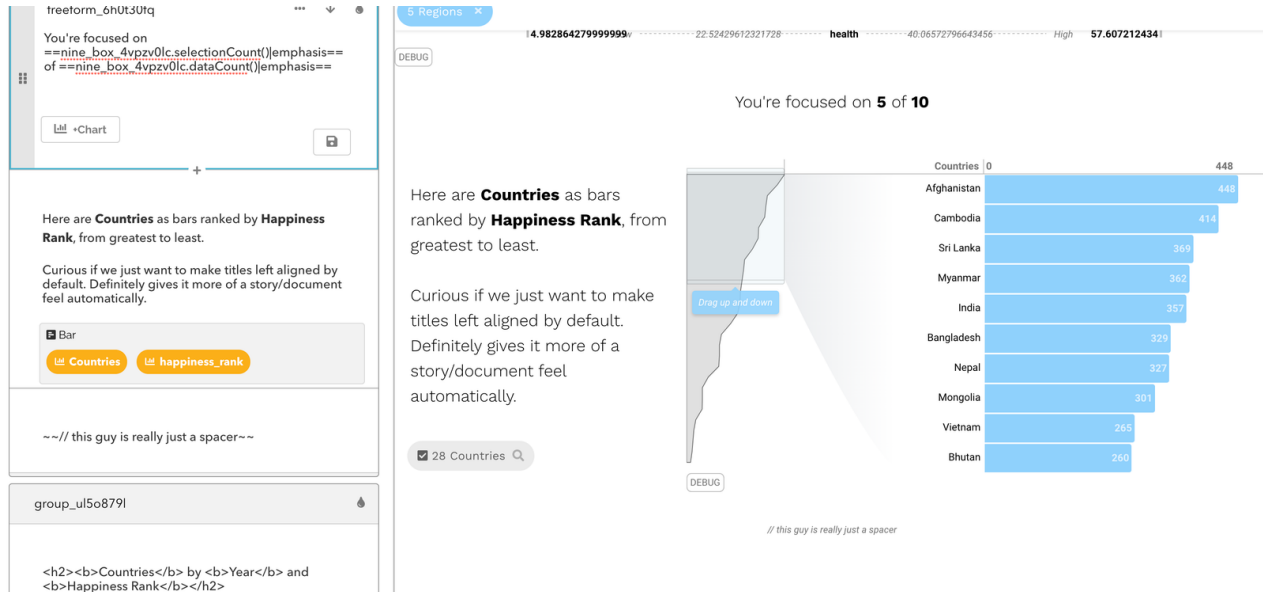
### And attempting to override text, (text cursor would be present)



- If overridden, a button appears on the vis config to add a new one (inserted at the top).

Slice card unfocused,

- Text looks more normal like other text.



## Other

- Stripping out HTML → translated to plain text or code formatting?
- No direct embedding of iframes... (Support in a future improvement)

1 Yes, embedding iframe **is** bad because any javascript that runs **in** an iframe has access to the parent **document**

2 3:48

3 I guess I should clarify, that having the user directly put **in** the html **for** the iframe like `<iframe src="" blah blah />` **is** bad

4 3:49

5 we really shouldn't let the user put any html tags **in** there. We can provide markdown syntax **for** them to format text but **no** html directly (edited)

1 jameslytle 4:16 PM

2 ok cool

3 4:17

4 re: embedding, I've noticed tools going **in** the direction **o**  
**f** just paste a link **and then** they will automatically handle the e  
mbed **if** it **is** supported. Is that what you are thinking?

1 **meghna** 4:21 PM

2 yea, providing some kind **of** mechanism **for** embedding so the  
user **is not** typing **in** html

🔗 ► **May 11, 2020**

## Design Solution

### Feature Summary

Juicebox is for data storytelling, and writing (and styling) text easily is important to make the telling an enjoyable creation experience. These design proposes a series of feature updates revolving around out text area of the Slice card.

### Design Prototype

[paste figma url](#)

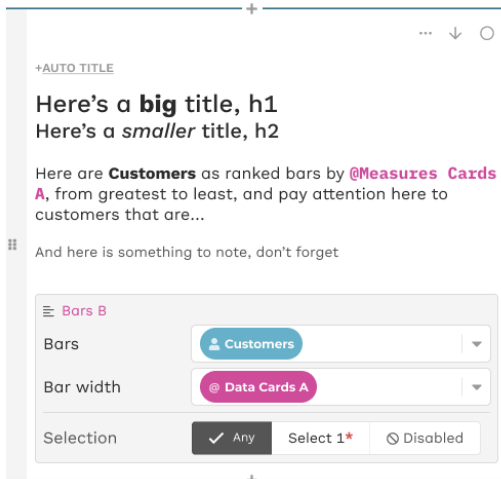
### Assets

- Maybe this [wysiwig editor](#) could be useful.

### Design File

<https://www.figma.com/file/bzpUuCMQOXytBK9qILsffN/Slice-Text-Styling?node-id=0%3A1>

*Where this is headed...*



# Here's a **big** title, h1

## Here's a *smaller* title, h2

Here are **Customers** as ranked bars by **Sales**, from greatest to least, and pay attention here to customers that are...

And here is something to note, don't forget

*...but we can make our way there...*

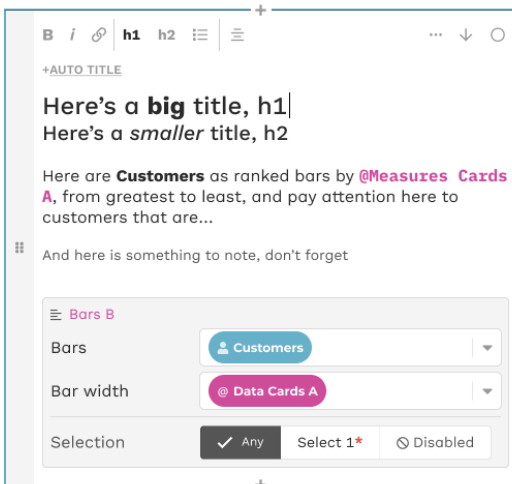
## Design Improvements / Suggested order of development

1. Apply updated Text formatting to markdown, strip out things we don't support
2. UI & Keyboard shortcuts to inject/apply markdown styles
3. Auto Titles based on vis config
4. Dynamic Text simplifications and new features
5. Typeahead popup for "@" to reference slices
6. Revamp text editor to be visual, applying styles in place

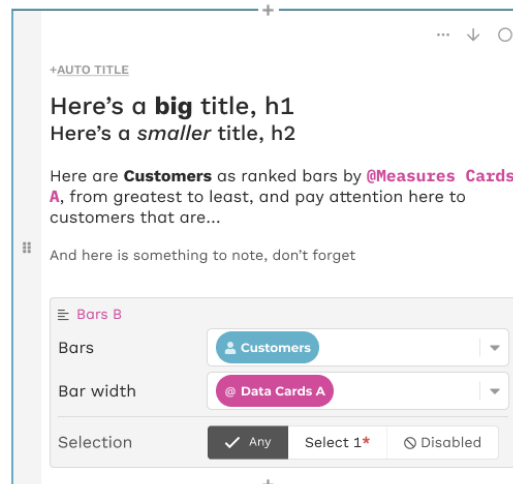
## Text Formatting

[Figma Frame](#)

Card focused, and text area in focus



Card focused, and text area NOT in focus



## All Options

- Bold
- Italic
- H1 → Large
- H2 → Medium
- H3 → Default bolded
- H4 → Default
- H5 → Small
- Link
- Image (temporary support still via markdown)
- Unordered List
- Ordered List
- Blockquote
- HR
- Line breaks are respected
- Align left, right, center → Center is still default

→ Our text elements will still get colored automatically black or white, depending on background color.

## Markdown

- Most styles are possible in markdown
- Only text alignment isn't
- Markdown will always be possible, but in the future it should render the text itself and disappear similar to other tools like Paper and Slack

## Stripping away

- All HTML
- Markdown we don't support matching styles for

## Text Formatting UI, Keyboard Shortcuts

We can start off by applying markdown to blocks and spans? Ultimately we want to style the editor text itself.

### Options of how to apply Styles

Style	UI	Keyboard command	Markdown
Header 1	h1		#
Header 2	h2		##
Body bold	b		###
Body			####
Note			#####
Bold	<b>b</b>	⌘-b	*bold*
Italic	<i>i</i>	⌘-i	<u>_italic_</u>
Link	icon	⌘-k	[I'm an inline-style link] (https://www.google.com)
Bullet list	icon		*



Numbered list			1.
Blockquote			>
Horizontal rule			---
Left align text / Right align(?)	icon toggle		

## UI for styles

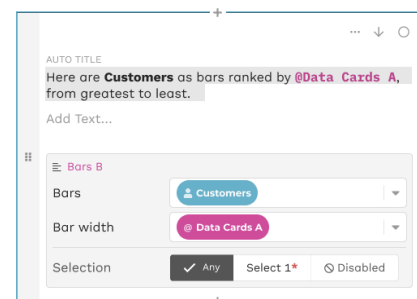
- While the text area is focused, New editor appears where the slug name used to
- UI icons both apply styling and indicate what styling has been applied
- Text alignment cannot be applied through markdown, and it applies to the entire text block. This icon would open a dropdown to select from:
  - Left Align `tl`
  - Center Align `tc` (default)
  - Right Align `tr`

## Auto Titles

### Figma frame

Auto Title is text automatically added to the slice title based on the current visualization configured in that slice. It helps the author with good starting points for a description story.

- Auto Title is present only if vis config requirements are satisfied.
- Title to be applied may be found based on the table below.
- The Auto Title updates automatically if vis config changes.
- Custom text may still be added in the editable “Add text...” textarea.



## Overriding Auto Titles

- Click directly on text, “Click to override” guidance appears
- When clicking, the auto title text is added to the editable textarea at the top. That text now no longer stays in sync with the vis config

- If the Auto Title on that Slice card has been overridden AND a valid vis config is present, then a "+ Auto Title" text appears, allow the Author to click it to add back in a fresh Auto Title based on the current vis config.

Slice Card	Auto Title	Rules	Variations
Data Card - Measures	{Here is} an overview of performance.		none
Data Card - Dimensions	Choose how you would like to break out performance:		none
Filters	Set any filters of interest:		none
Bars	{Here are} <b>Departments</b> as bars ranked by <b>Measure A</b> , from greatest to least.		none
Table	{Here is} a list of <b>Departments</b> alongside key details.	Departments = first non-measure ingredient	none
Trend	{Here is} how <b>Measure A</b> changed from <b>Jan 1, 2018</b> to <b>Jan 1, 2019</b> .		none
Scatterplot	{Here are} <b>Departments</b> sized by <b>Measure C</b> , compared by <b>Measure A</b> and <b>Measure B</b> .		If no size measure: <ul style="list-style-type: none"> <li>• {Here are} <b>Departments</b> compared by <b>Measure A</b> and <b>Measure B</b>.</li> </ul>
Map	{Here is} a map of <b>Departments</b> , sized by <b>Measure A</b> and colored by <b>Measure B</b> .		If no color measure: <ul style="list-style-type: none"> <li>• {Here is} a map of <b>Departments</b>, sized by <b>Measure A</b>.</li> </ul>

			and also, if no size measure: <ul style="list-style-type: none"> <li>• {Here is} a map of <b>Departments.</b></li> </ul>
Leaderboard	Compare which <b>Departments</b> are leaders across performance.		none
<i>Future slice cards</i>			
Cards			
Bubble	Here are (Departments) as <b>bubbles</b> scaled by (Measure A) and colored by (Measure B).		
Distribution	Here are (Departments) by (Measure A) <b>performance buckets.</b>		

### Concepts towards making it more “talk like a person”

Where {Here is/Here are} is used, we could randomly choose from similar plural / singular phrases so every titles isn’t so redundant.

Plural / Singular | Some phrases work for both.

- Here are / is
- Review
- Look at
- Take a look at

You could also randomly, insert conversational opening phrases like,

- Now,
- Next,

When used together, can randomly result in things, like,

- Now, take a look at **Departments** sized by Measure C, compared by **Measure A** and **Measure B.**

- Review **Departments** sized by Measure C, compared by **Measure A** and **Measure B**.
- Next, here are **Departments** sized by Measure C, compared by **Measure A** and **Measure B**.

### How do these work? idea for technical approach...

- We could make use of the `subtitle` config option in our slice header, to keep things separate.
- We always want the Auto Title to come first...

## Dynamic Text

### Figma Design Frame

#### Phase 1

Repurpose/Support much of the [existing dunderscore](#) syntax in slightly simpler form.

- “Selection” is the default thing to reference, so we assume that.
- **Emphasis** is the default, so we enforce that where needed.

#### Referencing slices

Want to be able to easily reference another slice to retrieve its info in specific ways.

```
==barslugwhatever.selectionDisplay()==
```

#### Example

Given a Slice card @Bars A, by Stores and Population, *where no Stories are selected: (guessed at right dunderscore approach)*

- `@Bars A` — count and label as bold text.
  - `selectionDisplay()` → **34 Regions**
- ``@BarsA|pill` — slice pill
  - `embedSlicePill()` → **34 Regions** (pill)
- `@BarsA|noemphasis` — count and label, unbolded.
  - `selectionDisplay()` → 34 Regions
- `@BarsA.label|noemphasis` — label, unbolded
  - `?` → Regions
- `@BarsA.label|noemphasis|singular` — label, singular, unbolded

- ? → Region
- @BarsA.number — label, count, bold
  - selectionCount()|emphasis → **34**

Can still reference nonselected info. Given **12 Regions** are selected,

- @BarsA.dataCount dataCount()|emphasis → **34**

NEW | Make it possible to reference a number value from the selected Measure

- `@DataCardsA.number — in a Data Card, get the value of the selected data item, bolded
  - Measure card — ?? → **38.2k**

Actually, just using @BarsA may be sufficient? Only reason not to might be for social @ mentioning purposes, but don't expect plans for that in this context.

Update Auto Titles to use new dynamic text

## Phase 2

- Expand to allow for dynamic measure calculations that respect the filter context of that slice.
  - This would allow you to reference any ingredient in text, not just ones selected in other slices.
  - Measure calculations
  - Easier more accurate counts on Dimensions

Example: @BarsA have a @TotalSales.label of @TotalSales.number

## Questions

- Can styling customizations also be applied to Auto Text?
- What if dynamic text didn't use the |noemphasis property, but just bolded/unbolded like everything else?

## Typeahead for @ slice referencing/ingredients

- This one is blocked by the slug / UI updates for referencing slices by things like "BarA, Barb, TableA..." instead of a random slug
- Ability to type "@" to get a searchable/clickable popup showing all available slices slugs (current slice and ones before it that have a vis config)

## ▶ Type & Style in place

### Adding Emoji support

- Would like to support use of emojis in text areas.
- 🤔🏠

### Other Questions

- Should any improvements be done by a specific time vs. one that would be nice in the future?
- Is anything a placeholder for now?
- Are there *someday* improvement ideas?

## To Dos

- ☐ dev ticket for finishing story style updates from markdown
- ☐ Dynamic text syntax refinement, testing
- ☐ Typeahead design
- ☐ Auto Title testing

## Tools

Doing things with typed text...(Paper based their editor on this)



react-ace

A react component for Ace Editor

[www.npmjs.com](http://www.npmjs.com)

 react-ace • [www.npmjs.com](http://www.npmjs.com)

- <https://ace.c9.io/>

---

## Development notes

### JB-3108

- want line breaks to actually add `</br>` to the template
  - Then I can reduce the vertical margin being applied from the `<p>` element
- Make links always `target="_blank"`

**juiceanalytics**

