# Tableau Accessibility Testing Plan

# Introduction

It's important to consider accessibility early and throughout the design and development process. Including accessibility from the start of a project increases the positive impact of designing for everyone while decreasing development costs when accessibility is addressed much later. This testing guide will help you understand how to test Tableau dashboards using multiple methods.

## Test early, test often

Testing at the end of the development process has two risks:

1. Projects tend to run over time and budget. Testing is often rushed, omitted, or ignored thanks to such pressures.
2. It is more work to fix problems discovered late in a process than to do things right from the start.

## Before you start

Tableau is a business intelligence (BI) tool that allows for analytics and visualization of data. First, ask yourself, is this the right tool for the project?

In addition, other considerations include:

- How will you be sharing this information?
- Will you be publishing to Tableau server, or Tableau public?
- Will the dashboard be embedded on a webpage or, in an application?
- Will viewing the dashboard require Tableau Reader?

Keeping in mind the business requirements and how you share the information, it's important to consider accessibility early and throughout the design and development process for seamless integration into the project. The Tableau Accessibility Design team has published a helpful and informative guide to demonstrate for developers how they can incorporate accessibility in their design during the development of Tableau content. https://education.mn.gov/mdeprod/idcplg?IdcService=GET_FILE&dDocName=PROD046623&RevisionSelectionMethod=latestReleased&Rendition=primary

## Accessibility and usability definitions

**Accessibility** addresses discriminatory aspects related to equivalent user experience for people with disabilities. Web accessibility means that people with disabilities can equally perceive, understand, navigate, and interact with websites and tools. It also means that they can contribute equally without barriers.

**Usability** is about designing products to be effective, efficient, and satisfying. Usability includes *user experience design*. This may include general aspects that impact everyone and do not disproportionally impact people with disabilities. Usability practice and research often does not sufficiently address the needs of people with disabilities.

## Legal requirements

The goal of the state of Minnesota's Accessibility Standard is to improve the accessibility and usability of information technology products and services for all government end users in the state of Minnesota. The standard incorporates the Web Content Accessibility Guidelines 2.0 and Section 508 of the Rehabilitation Act of 1973.

Statutes on public records (363A.42) and continuing education (363A.43) effective January 1, 2013, supplementing the standard to make agencies financially liable for inaccessible records and other documents and course material modified or produced after that date.

The Office of Accessibility website provides these and related links for reference on its Policies and Standards page.

## Testing tools

Automated testing tools will help you test for about 20% of the accessibility guidelines. A good testing process starts with an automated testing tool and is followed by manual testing.

To test a dashboard, you need to use more than one of the testing practices outlined in this guide. For example, if WAVE is your choice for automated testing, you will also need to do visual, keyboard-only, and browser zoom testing.

There are other testing tools available, however we have chosen to focus on the tools more widely used at the state.

### Semi-Automated testing tools

- WAVE – free plugin (Chrome/Firefox)
- Colour Contrast Analyzer – free download

### Manual testing

- Visual testing
- Keyboard only
- NVDA screen reader – free download
- Browser zoom
- ANDI (https://www.ssa.gov/accessibility/andi/help/howtouse.html)

**Note**: you may need to submit a MN Service Hub request for some of the above tools, and request training from your agency's accessibility coordinator.

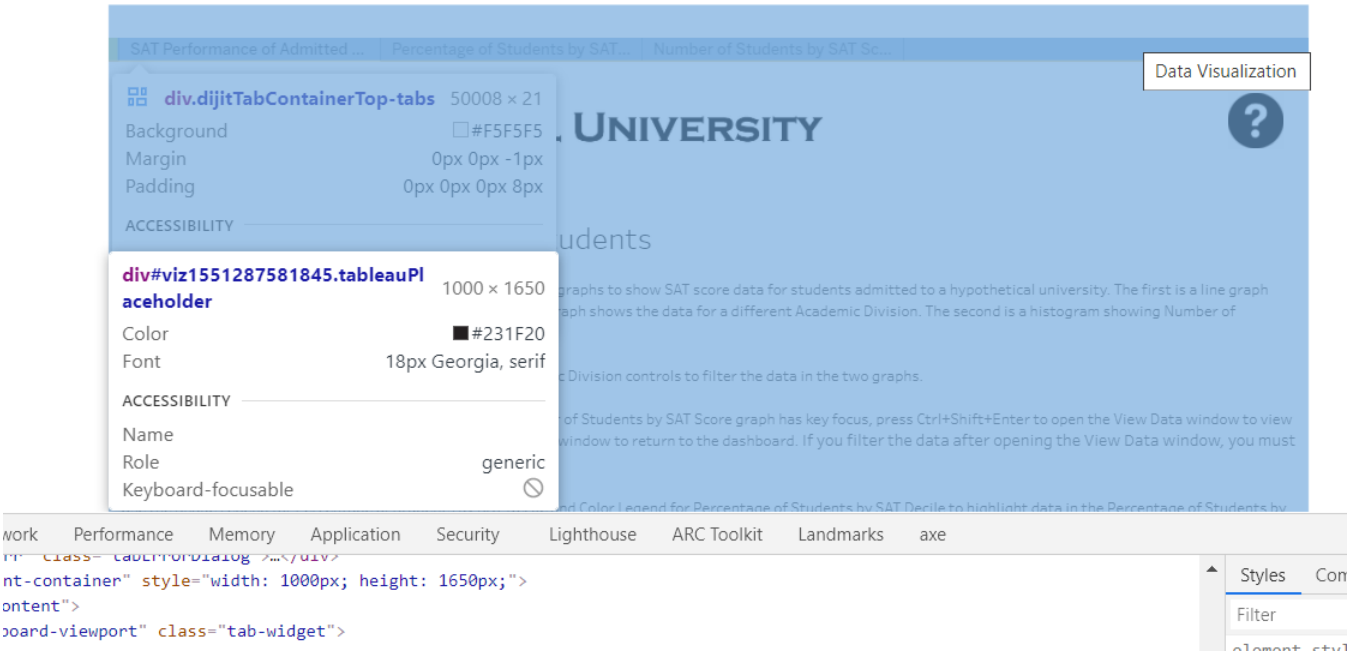
## Tableau dashboard publishing considerations for testing

It's important to make sure you are testing the dashboard within the browser. To properly test the dashboard, it may be necessary to isolate the dashboard in a separate browser window. This will allow some of the testing tools to work properly and test just the dashboard elements without the surrounding webpage, for example, if using WAVE to test a dashboard embedded on a web page with an iFrame.

The following is one method for isolating the dashboard embedded on a page using an iFrame:

1. Open the webpage and use F12 to bring up the developer toolbar.
2. Use the select element icon to focus on the dashboard embedded with the iFrame. This will help you focus on where it is located in the source code.

Tableau has to say about working with Equal Entry 🔗.




a. You may need to expand elements to get to the right section of the source code. Once you find the iFrame, you can copy the URL.
    - Sample code:

      <iframe frameboarder=”0” marginheight=”0” marginwidth=”0” title=”Data Visualization” allowtransparency=”true” allowfullscreen=”true” <class=”tableauViz” style=”display: block; width: 100px; height: 1650px; margin: 0px; padding: 0px; border: none;” src=”https//public.tableau.com/views/AccessibilityTestDashboardforTableau2_splay_spinner=no&:display_overlay=yes&:display_count=yes&:loadOrderID=0”>

3. Paste the URL into a new browser window and you’re ready to test.

## Dashboards

What areas of a Tableau dashboard can be tested?

- Page language (published to Tableau server or embedded)
- Page Title
- Images (other than visualizations)
- Page structure
    - Headings
    - Lists
    - Tables

    - Tabs/Sections/Worksheets
- Filtering components and other form elements (including Legend (highlight selected items))
- Data Visualization (Charts and Graphs) (including testing keyboard shortcut works to bring up data HTML table)
- Page text
    - Visualization Summary
    - Other text elements
- Links

# Tests

## Visual Test

### *How to test*

1. Scan the page to determine if there are sensory characteristics of components such as shape, size, visual location, or orientation used when providing instructions or operating procedures.
2. Determine whether a color is the only method used to convey information, for example color used in dashboard visualizations includes other means of identification for the data elements. Remember to check items such as error messages and verify color alone is not used.

### *Success Criteria tested*

- 1.3.3 Sensory Characteristics - Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, color, size, visual location, orientation, or sound.
- 1.4.1 Use of Color - Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

## WAVE

The Web Accessibility Evaluation (WAVE) tool has many checks for compliance issues found in Section 508 and WCAG 2.0 guidelines.

You can install the WAVE extension for the Firefox or Google Chrome browser from the Chrome Web Store or Firefox Add-ons page. To do so in Chrome, search for “WAVE Evaluation Tool.” Select the “Add to Chrome” button to install.

Detailed instructions on using the WAVE tool can be found on the WebAIM website (https://wave.webaim.org).

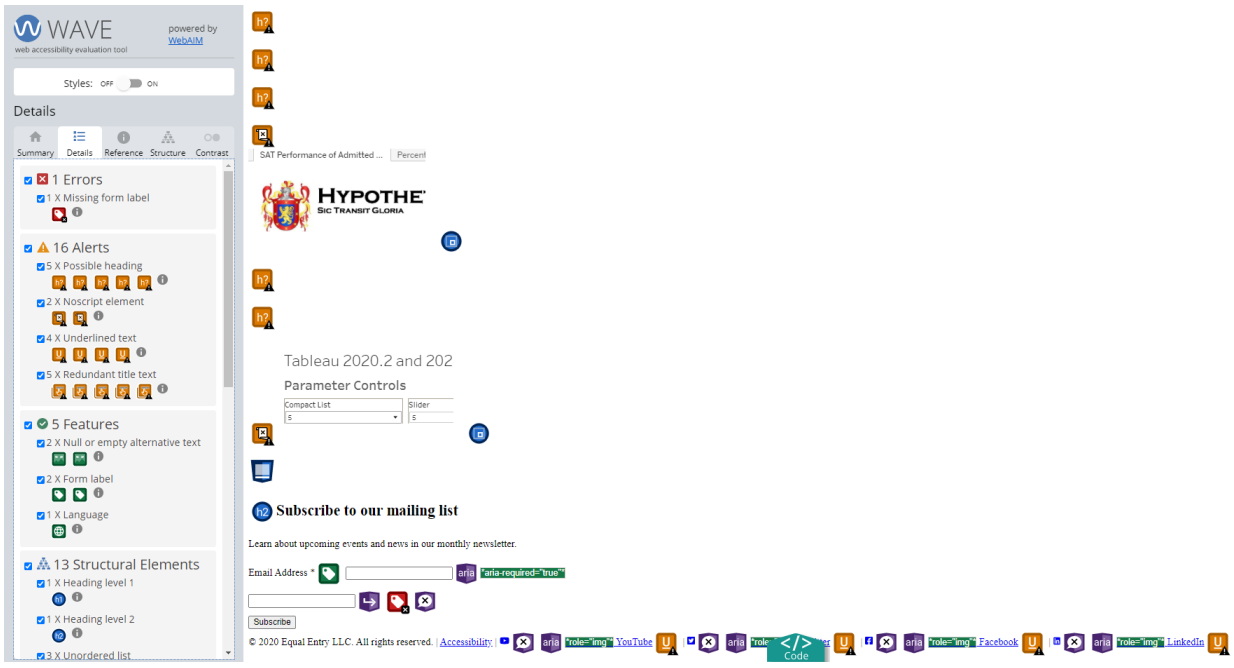
Important:  If you used an iFrame to embed the dashboard on a webpage, use the ANDI tool for testing. WAVE can only be used if you can first isolate the dashboard in a separate browser window.

You can view the mapping of WAVE items to WCAG success criteria on the WebAIM help webpage.

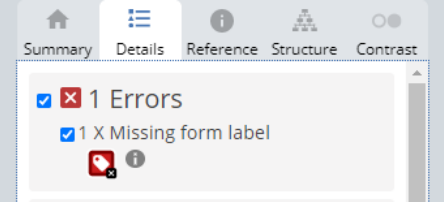
### *How to test*

To test the dashboard, do the following steps:

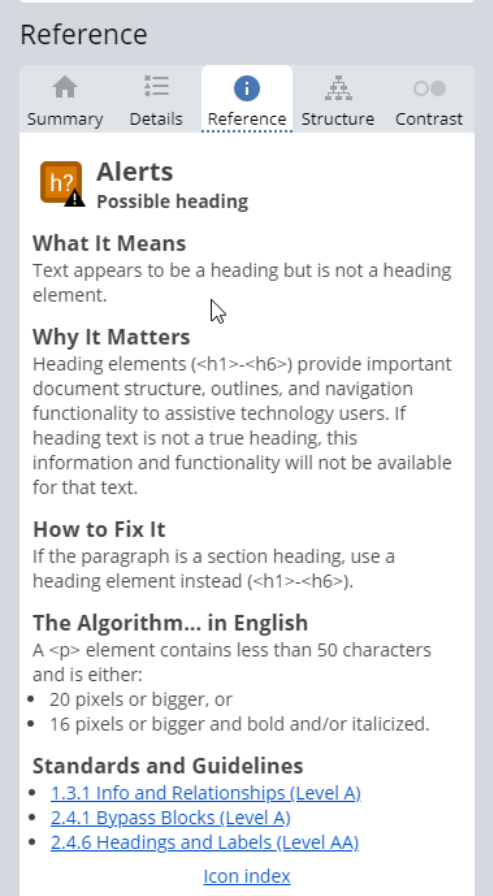
1. Launch the WAVE tool.
2. Once the WAVE extension is enabled, a panel will open on the left side of the browser window with these features:
   a. At the top of the panel there is an option to toggle the styles on and off:
      - Styles on (activated on initial load): The page displays as it normally does with CSS/styles intact.
      - Styles off: The page displays without CSS/styles, which can be very useful when going through an interactive web mapping application.
   b. In addition, the following options are present in the left-hand panel:
      - Summary (activated on initial load): A snapshot of items on the page, grouped by: Errors, Alerts, Features, Structural Elements, HTML5 and ARIA, and Contrast errors. For example Errors 1, and Alerts 1.
      - Details: Includes the full detailed list of the summary items. For example Errors 1: 1 x Missing alternative text.
      - Reference: An explanation of the WAVE icons and how to make your page more accessible.
      - Structure: The page's heading structure (h1, h2, h3, etc.).
      - Contrast: Any color contrast issues on the page will be displayed once selected.
      - Select the Details tab.
      - Review Errors (issues), manual check Alerts, Features (alt text, form label, Language), and Structural Elements (Headings, inline frame, Navigation).




3. Select the embedded icon, the focus will jump to the page with the element highlight.




4. Select the Information icon, Reference tab will display detailed information regarding that element.




5. When verifying and fixing the errors, alerts, and color contrast elements, consult with the recommendations provided in the WAVE extension and the Web Content Accessibility Guidelines (WCAG) 2.0. Another check via the WAVE extension tool can determine if you have met the criteria set by WAVE and/or WCAG 2.0.

***Success Criteria tested:***

- 1.1.1 Non-text Content - All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for these situations: controls/input, time-based media, test, sensory, CAPTCHA, decoration/formatting/invisible.

- 1.3.1 Info and Relationships - Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text (will need additional testing for an empty label).
- 2.1.1 Keyboard - All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- 2.4.1 Bypass Blocks - A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.
- 2.4.2 Page Titled - Web pages have titles that describe the topic or purpose.
- 2.4.4 Link Purpose (In Context) - The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.
- 2.4.6 Headings and Labels - Headings and labels describe the topic or purpose.
- 3.1.1 Language of Page - The default human language of each Web page can be programmatically determined.
- 3.1.2 Language of Parts - The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.
- 3.3.2 Labels or Instructions - Labels or instructions are provided when content requires user input.
- 4.1.2 Name, Role, Value – For all user interface components, the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

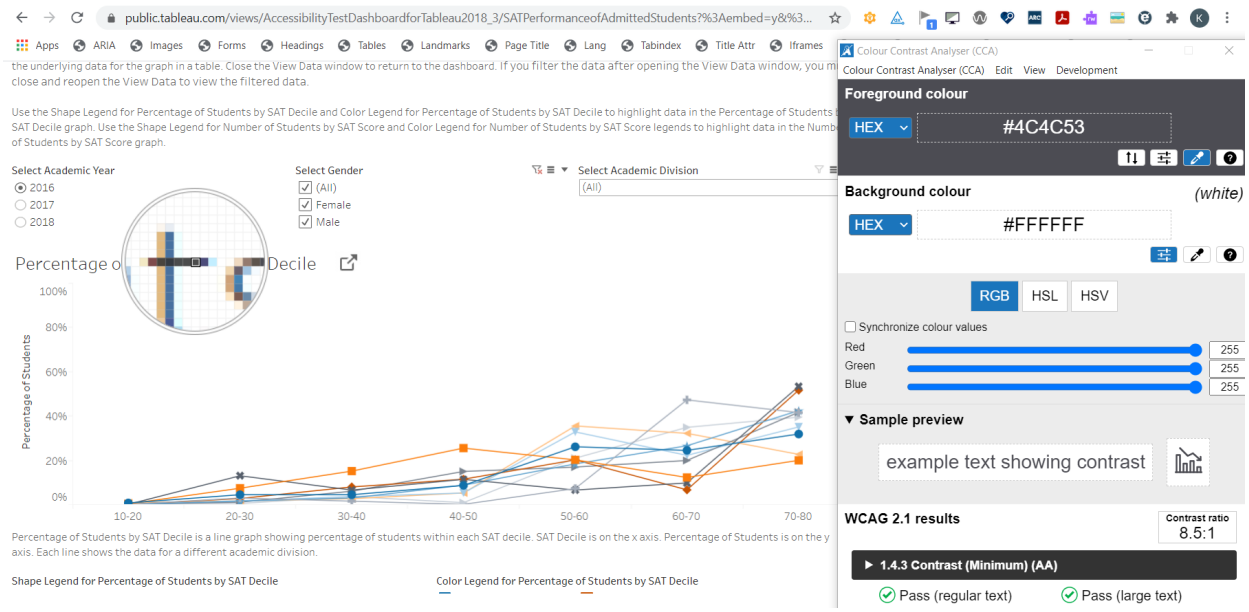
## Colour Contrast Analyzer

The Colour Contrast Analyser can help determine if the foreground and background of the text and visual elements, such as charts and graphs, logos, etc. meet the guidelines for color contrast.

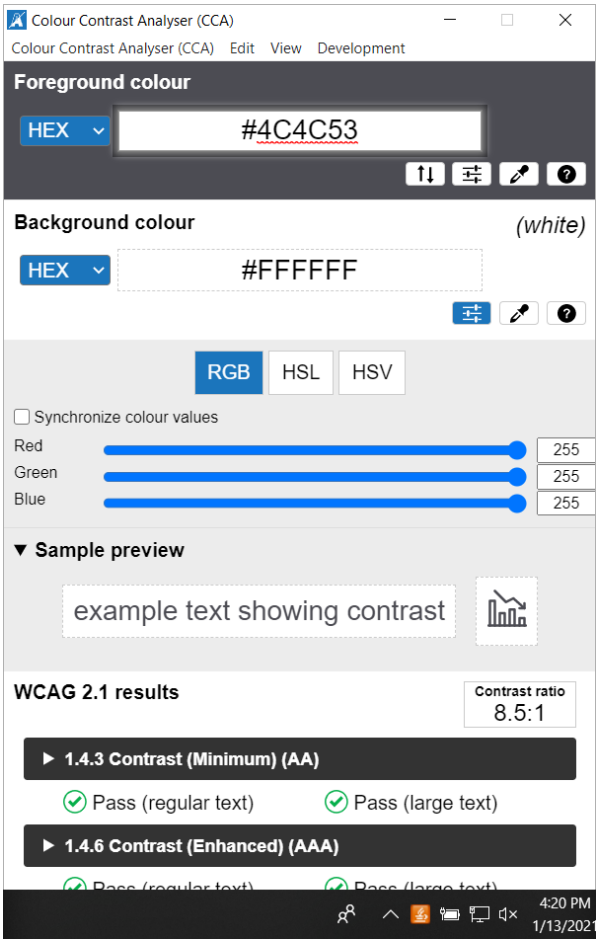
You can download the software from the Colour Contrast Analyser website https://developer.paciellogroup.com/resources/contrastanalyser/).

### *How to test*

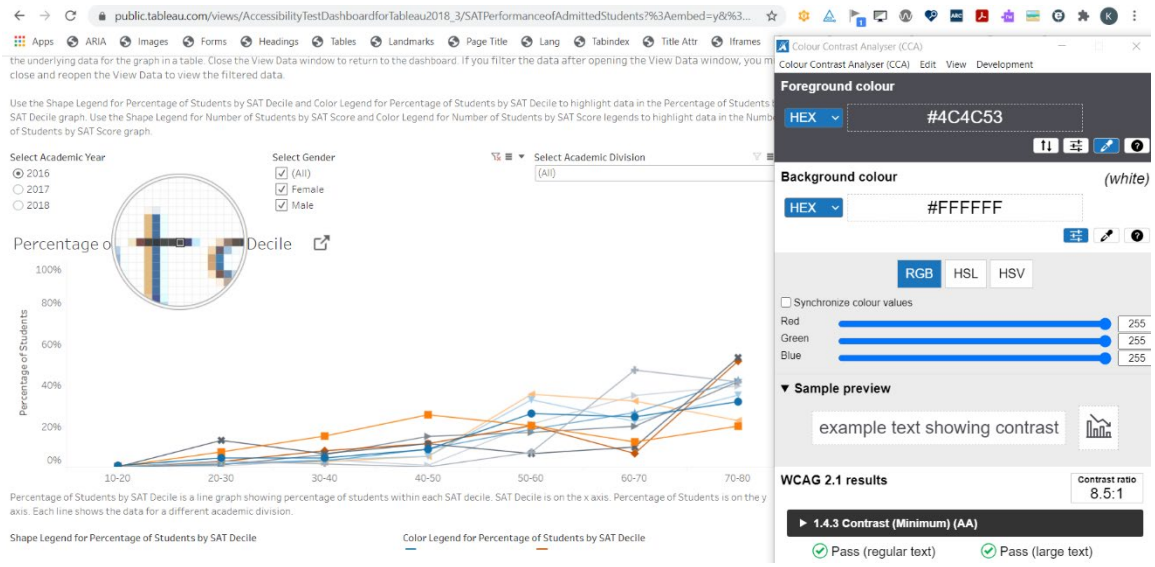
1. Open the Colour Contrast Analyser.



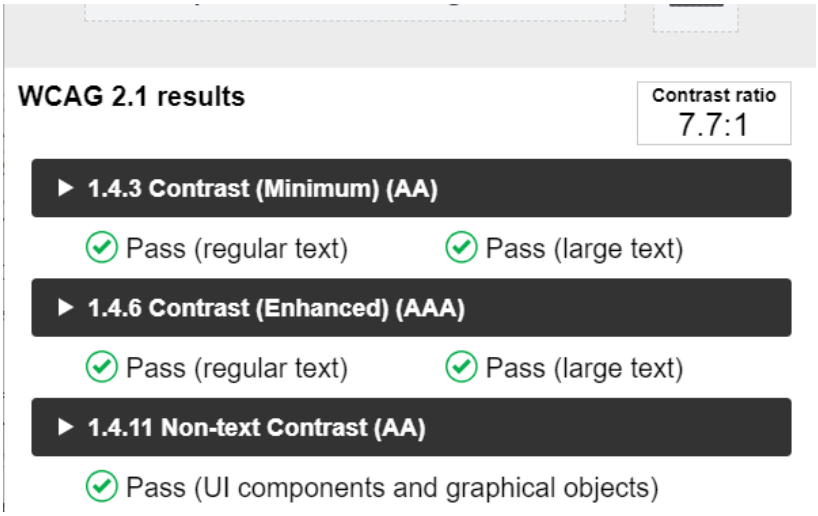

2. Select the eyedropper in the Foreground colour section. Move the eyedropper to the section of the application you want to test and select using the left-mouse button. The Hex code will now display in the Foreground colour section of the tool.



3. Select the eyedropper in the Background colour section. Move the eyedropper to the section of the application you want to test and select, using the left-mouse button. The Hex code will now display in the Background colour section of the tool.




4. View the WCAG 2.1 results section. Verify you have passed the contrast ratio requirements.




***Success Criteria tested:***

- 1.4.3 Contrast (Minimum) - The visual presentation of text and images of text has a contrast ratio of at least 4.5:1

## Keyboard Only

Keyboard testing is one of the most efficient, impactful, and easy tasks that any team member can do to find accessibility issues. When keyboard testing is performed multiple times by multiple people at every possible stage of a project, you will be a step closer to the inclusion we all want.

### *How to test*

#### *Keyboard navigation*

1. Use the keyboard to navigate through all the interactive interface components in the dashboard.
    a. Use **Tab** and **Shift+Tab** to navigate between widgets both forwards and backwards.
    b. Use the arrow keys to navigate between the focusable elements within a composite widget.
2. If a “keyboard trap" prevents the focus from leaving a widget:
    a. Use your mouse to move the focus to the next widget.
    b. Resume testing.
3. If you encounter a trigger component that reveals hidden content:
    a. Activate the trigger.
    b. Navigate through the revealed content.
    c. Close the revealed content.
    d. Resume navigating in the dashboard.
4. Verify that you can navigate to all interactive components using the keyboard.
5. Record your results.
    a. If you find any failures, select **Fail**, then add them as failure instances.
    b. Otherwise, select **Pass**.

#### *No Keyboard Trap*

1. Use standard keyboard commands (unmodified Tab and arrow keys) to navigate through all the interactive interface components in the dashboard.
2. If you can't navigate away from a component using standard keyboard commands:
    a. Examine the component's accessible name and accessible description to determine whether they describe an alternative keyboard command.
    b. If an alternative keyboard command is documented, test whether it works.
3. Verify that you can navigate away from all components using either*:*
    a. Standard keyboard commands, or
    b. An alternative keyboard command that is described to users.
4. Record your results:
    a. If you find any failures, select **Fail**, then add them as failure instances.
    b. Otherwise, select **Pass**.

#### *On Focus*

1. Verify that the keyboard focus indicator is visible on all the interactive interface components.
    a. Press the tab keyboard to navigate through all components in the dashboard.
2. Verify that moving focus to a component does not trigger any unexpected change of context, such as:
    a. Submitting a form automatically
    b. Launching a new window
    c. Shifting focus automatically to another component
3. Record your results.

a. If you find any failures, select **Fail**, then add them as failure instances.
b. Otherwise, select **Pass**.

***On Input***

1. Use standard keyboard commands to navigate through all the interactive interface components in the dashboard.
    a. Enter data in each text field and then **Tab** away from the field.
    b. Change selections for selectable components such as toggle buttons, radio buttons, checkboxes, and list boxes.
2. Verify that changing the component's settings does not trigger any unexpected change in context, such as:
    a. Submitting a form automatically
    b. Launching a new window
3. Record your results.
    a. If you find any failures, select **Fail**, then add them as failure instances.
    b. Otherwise, select **Pass**.

***No Keystroke timing***

1. Use the keyboard to perform all functions that are available using a mouse.
2. Verify that individual keystrokes do not require specific timings, such as requiring users to:
    a. Repeatedly press a key within a short amount of time
    b. Enter a series of keystrokes within a short amount of time
    c. Press and hold a key for an extended amount of time
3. Record your results.
    a. If you find any failures, select **Fail**, then add them as failure instances.
    b. Otherwise, select **Pass**.

***Data Visualization (Charts and Graphs)***

In Tableau, the data visualizations are not available for navigation and control using the keyboard. There is, however, another way for the screen reader user, and all users, to bring up an accessible HTML data table containing the attributes used to create the visualization.

Once you have navigated to and focus is on the data visualization, use the keyboard shortcut Ctrl+Shift+Enter to open the HTML data table. If you have made a change to the data, using one of the filtering options, navigate back to the data visualization, choose Ctrl+Shift+Enter to open the HTML data table, and verify the filter has been applied to the data.

***Success Criteria tested***

- 2.4.3 Focus Order - If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability

- 2.1.1 Keyboard - All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- 2.1.2 No Keyboard Trap - If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving the focus away.
- 2.4.7 Focus Visible - Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.
- 3.2.1 On Focus - When any user interface component receives focus, it does not initiate a change of context.
- 3.2.2 On Input - Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component.

## NVDA

NVDA is a free screen reader available for anyone to download and install. In addition, you can find a great list of NVDA keyboard shortcuts on the WebAIM website. Before beginning your test, you will need to download, install, and configure NVDA. If you haven't already done so, the WebAIM website has some basic instructions for configuration.

Below, we have included some basic instructions on testing a dashboard using the NVDA screen reader. The instructions are not intended to be a full description of how to use NVDA. You can find a complete user guide on the NV Access Help webpage. In addition, you can find a great list of NVDA keyboard shortcuts on the WebAIM website.

Not all WCAG success criterion can be tested using a screen reader, however, using NVDA to read and navigate the dashboard can be a good tool in comparing the visual output, and coded structure and elements.

Before you begin testing with NVDA, review the list of success criteria for this section. Keep the success criteria in mind while reading and navigating the dashboard. This will help you with your accessibility testing.

Please note, when using the NVDA+F7 keyboard shortcut to bring up a list of elements, you will first need to isolate the dashboard. In addition, if you are not able to isolate the dashboard without the "embed=y" (because of a redirect) you will not be able to use the list of elements function.

### *How to test*

#### *Reading the dashboard*

1. Using the keyboard, try navigating the page using the following (keep in mind, the NVDA key is set during the configuration.):
    a. Read from the current position - NVDA+down arrow
    b. Reread current line – NVDA+up arrow
    c. Read previous or next word – Ctrl+left/right arrow

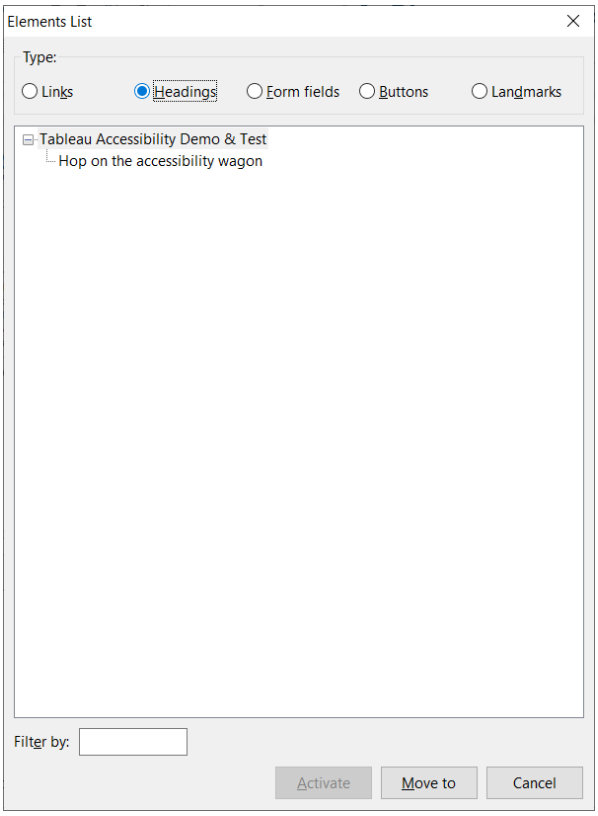
d. Read previous/next word – up/down arrow
e. Read previous/next character – left/right arrow
f. Stop reading – Ctrl

2. Images, such as logos, should contain alternative text that will be read by NVDA. While reading the dashboard, identify images and verify the alternative text is available and is descriptive.

***Page structure***

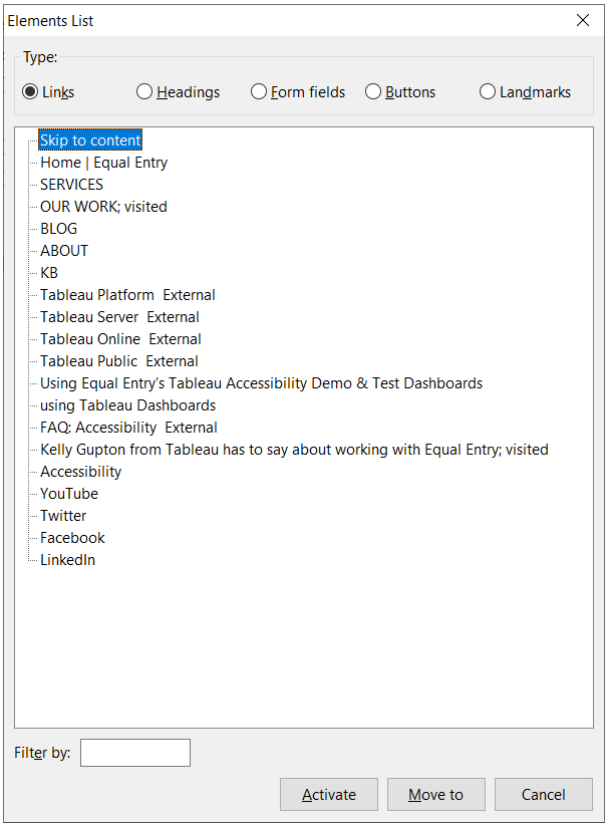
NVDA provides navigational shortcuts that allow the user to view a list of different elements and navigate to them directly.

1. Start by navigating the dashboard using the Tab key. This form of navigation will take the focus to each of the links and form controls on the dashboard.
2. Using the following keyboard shortcuts, verify page structure use of headings:
    a. Open the elements list – NVDA+F7
    b. Navigate to the headings radio button. Confirm headings are used in the dashboard, are correctly structured by heading level, and are descriptive.




    c. Additionally, test the use of headings and heading structure by navigating the dashboard with the single-key shortcut “H”. If the heading is properly coded in the dashboard, selecting the H on your keyboard should take focus to the heading element. Using single-key shortcuts 1-6 will allow the user to jump between heading levels.
3. Using the following keyboard shortcuts, verify page structure use of links:

a. Open the elements list – NVDA+F7
b. Navigate to the links radio button. Confirm the link text is descriptive and matches the visual links used in the dashboard. Note, an element visually identified as a button should not be listed in this list.
c. Additionally, test the use of headings and heading structure by navigating the dashboard with the "K" key.



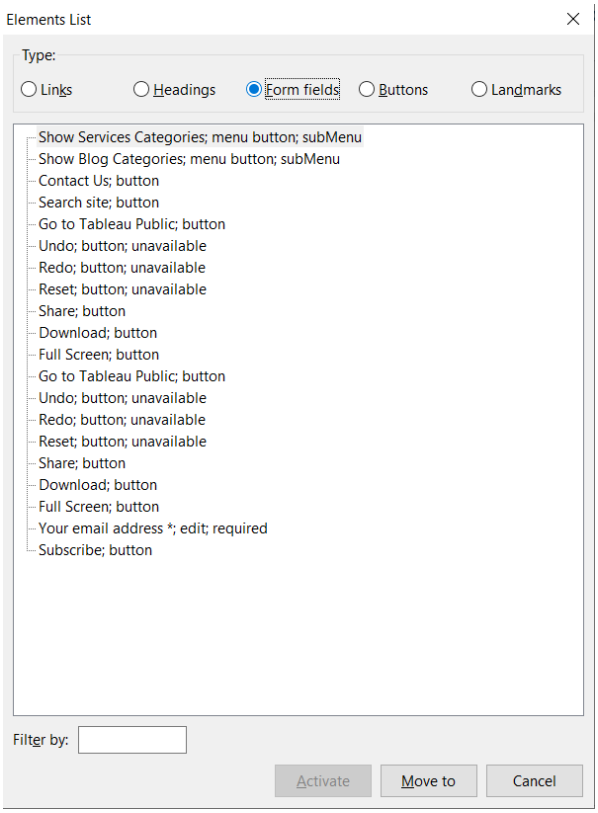

4. To test the dashboard for lists, use the single-key shortcut "L" to navigate. If the list is properly coded in the dashboard, selecting the L on your keyboard should take the focus to the list element. Use the single-key shortcut "I" to jump to individual items in a list.
5. While reading through the dashboard, you may have identified the use of tables. To quickly navigate to a table element, use the single-key shortcut "T". You can navigate within the data table by holding down Ctrl+Alt and using the up arrow, down arrow, left arrow, and right arrow to move from cell to cell. While navigating, if the table is properly structured with row and column headers, they will be read automatically.

***Form Elements***

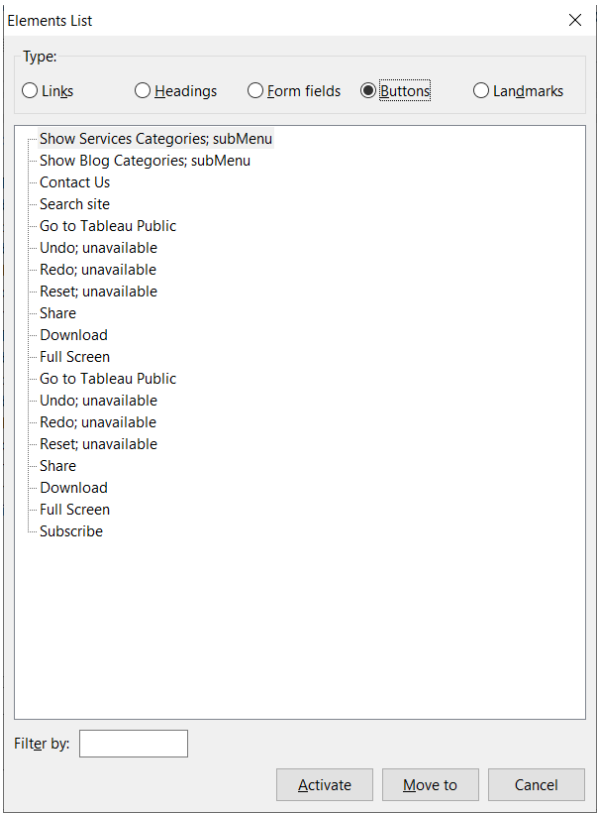
Some common form controls include inputs, checkboxes, radio buttons, combo boxes, and buttons. When a form control receives focus, the label is read by NVDA along with the type of form control.

There are a few ways you can navigate through the form controls in the dashboard:

1. Open the elements list – NVDA+F7
2. Navigate to the form fields radio button. Confirm the list of form controls includes a label that is descriptive and matches the expected form control behavior.




3. Open the elements list – NVDA+F7
4. Navigate to the buttons radio button. Confirm the list of buttons is descriptive and matches the expected behavior (a visual button is included in this list).
    a. Buttons used to submit a form should use the Enter key




5. Use the Tab and Shift+Tab keys to navigate through the form controls in the dashboard.
6. Checkboxes and radio buttons should be contained in a fieldset with a legend. A fieldset groups the controls together and the legend is read when you navigate any of the items within the group. Verify this with the following keyboard commands:
    a. Select and deselect checkboxes – Space
    b. Select within a group of radio buttons – up/down arrow
    c. Select an option in a combo box – up/down arrow

***Data Visualization (Charts and Graphs)***

In Tableau, the data visualizations are not available for navigation and control. There is, however, another way for the screen reader user, and all users, to bring up an accessible HTML data table containing the attributes used to create the visualization.

Once you have navigated to, and focus is on, the data visualization, use the keyboard shortcut Ctrl+Shift+Enter to open the HTML data table. If you have made a change to the data, using one of the filtering options, navigate back to the data visualization, choose Ctrl+Shift+Enter to open the HTML data table, and verify the filter has been applied to the data.

***Success Criteria tested***

- 1.1.1 Non-text Content - All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below

- 1.3.1 Info and Relationships - Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.
- 1.3.2 Meaningful Sequence - When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.
- 1.3.3 Sensory Characteristics - Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, color, size, visual location, orientation, or sound.
- 2.1.1 Keyboard - All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- 2.1.2 No Keyboard Trap - If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving the focus away.
- 2.4.1 Bypass Blocks - A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.
- 2.4.3 Focus Order - If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability
- 2.4.4 Link Purpose (In Context) - The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.
- 2.4.6 Headings and Labels - Headings and labels describe the topic or purpose.
- 3.2.2 On Input - Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component.
- 3.2.3 Consistent Navigation - Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated unless a change is initiated by the user.
- 3.2.4 Consistent Identification - Components that have the same functionality within a set of Web pages are identified consistently.
- 3.3.2 Labels or Instructions - Labels or instructions are provided when content requires user input.

## Browser Zoom

### *How to test*

1. Go to browser setting and change the Zoom level to 200%.
2. Review page text, images, and layout regions to ensure all content or functionality is not lost.
3. Review to ensure all the content scales uniformly, and the user agent provides scroll bars, if necessary.

#### *Success Criteria tested*

- 1.4.4 Resize text - Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.

## ANDI

The DHS ANDI tool can help test Tableau dashboards embedded on a page using an iFrame by allowing you to easily bring up just the dashboard in another browser window and use the ANDI modules to test.

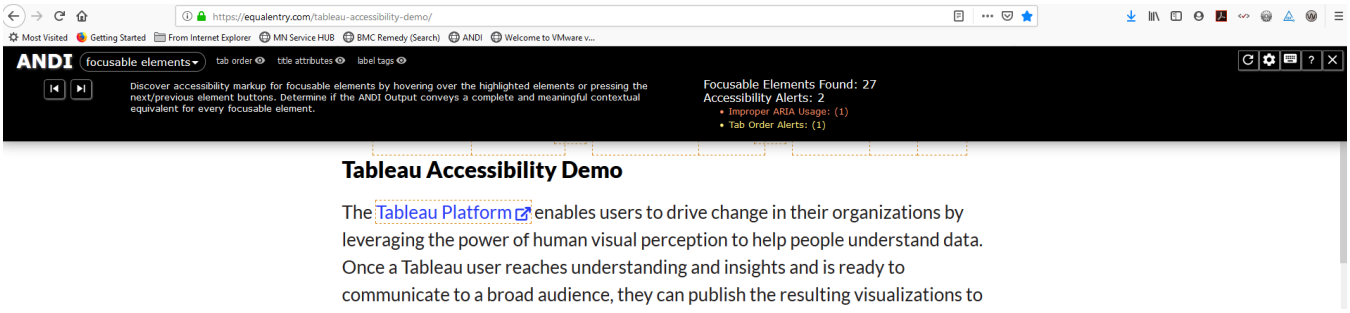
Installation instructions can be found on the Social Security Administration website install page (https://www.ssa.gov/accessibility/andi/help/install.html).

### *How to test*

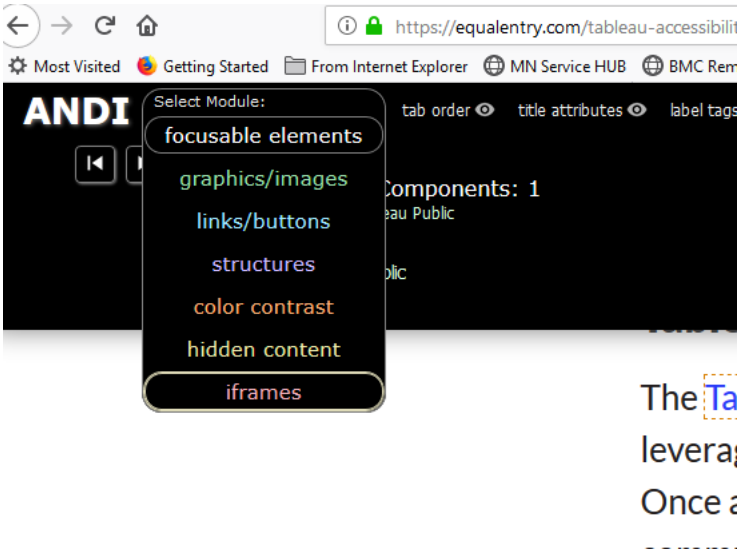
Detailed instructions on using the ANDI tool can be found on the Social Security Administration website (https://www.ssa.gov/accessibility/andi/help/modules.html).

To test an iFrame embedded dashboard, you will first do the following:

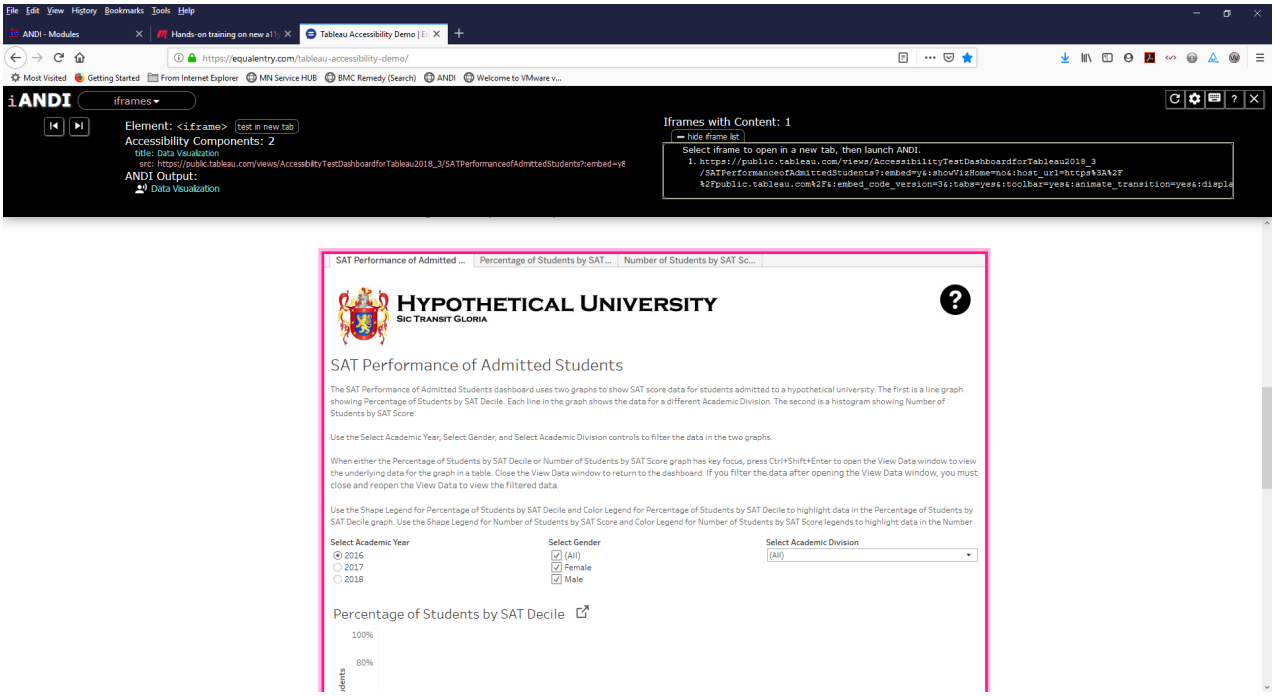
1. Open the webpage
2. Launch ANDI




3. Select iframes from the dropdown.



4. The iframe will be highlighted on the page. You can choose “test in new tab” or open the iframe list to show the link to open in a new tab.



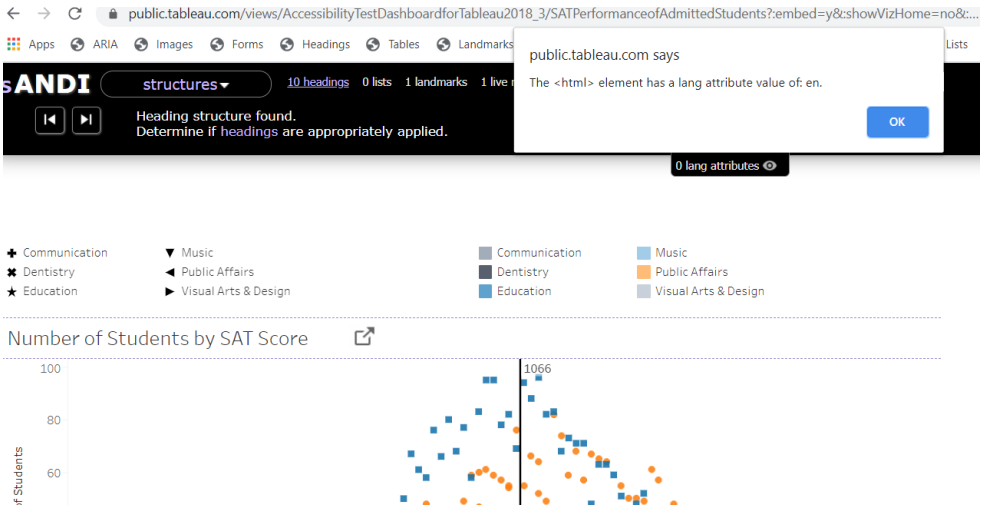

5. Once you have the iframe content open in a new tab, launch ANDI to begin testing.

### *Page language*

Verify the language of the page and any other page information, with a language different than the default page language, is identified and correct.

#### ***Language of Page***

1. ANDI: structures > more details > page language.
2. You can find a listing of the two or three letter language codes used on web pages on the Internet Assigned Numbers Authority's (IANA) Language subtag registry page. Once you have identified the page language, validate the codes used.




***Language of Parts***

If there are areas of the dashboard that differ from the default page language. Do the following…

1. ANDI: structures > more details > page language.
2. Internet Assigned Numbers Authority's (IANA) Language subtag registry to validate the two or three letter language codes used on the page.

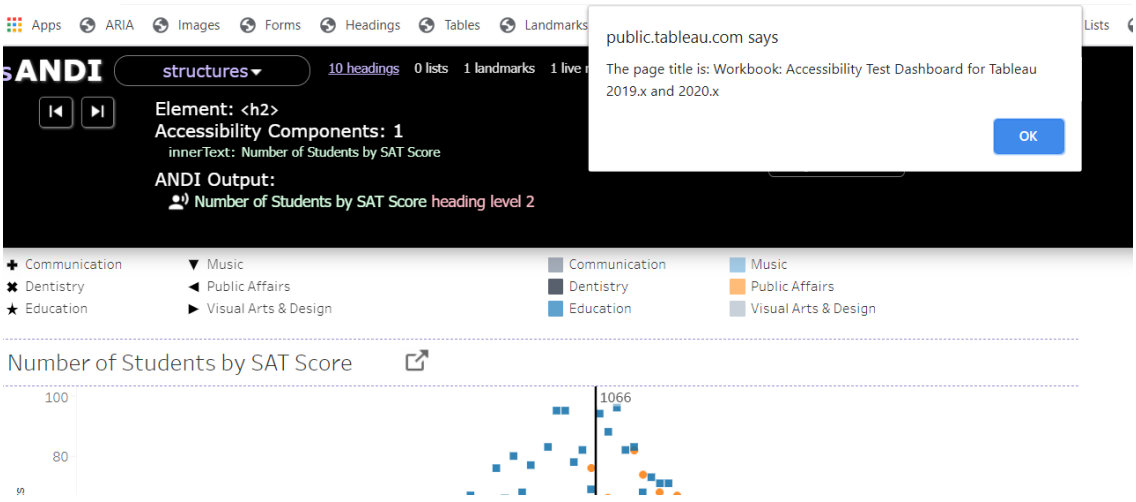
***Success Criteria tested:***

- 3.1.1 Language of Page - The default human language of each Web page can be programmatically determined.
- 3.1.2 Language of Parts - The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.

*Page Title*

Verify the page title is defined and is descriptive.

1. ANDI: structures > more details > page title.
2. Is there a page title? Is it descriptive?




***Success Criteria tested:***

1. 2.4.2 Page Titled - Web pages have titles that describe the topic or purpose.

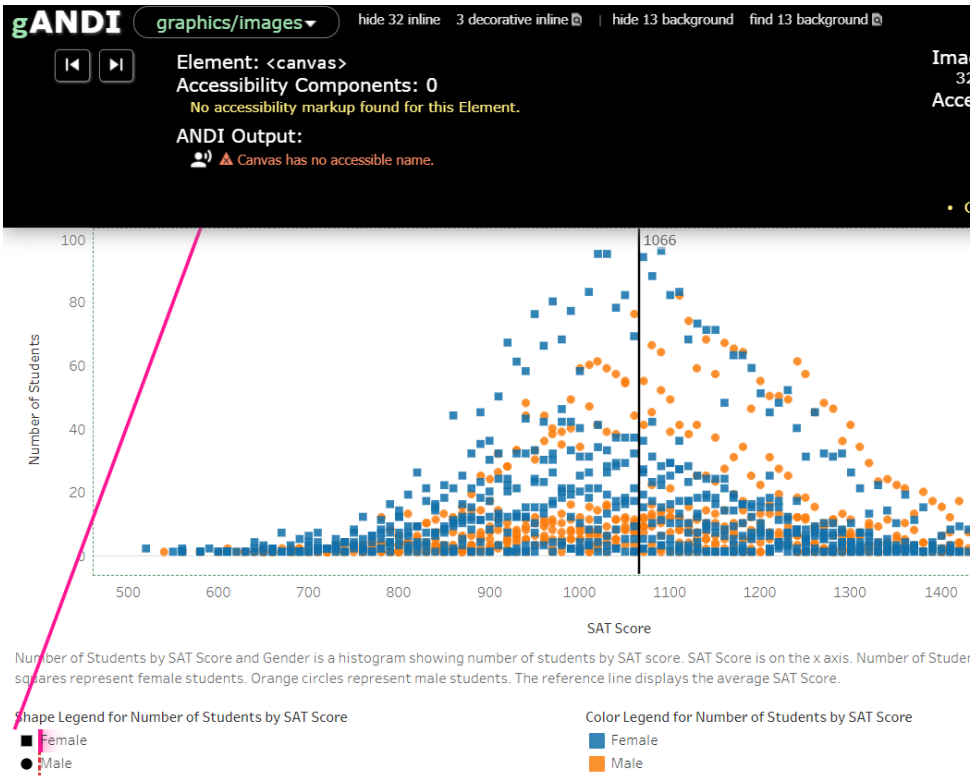
*Images*

***Non-text content***

All non-text content has a text alternative that serves the equivalent description.

1. Select the ANDI: graphics/images module to identify all images. Initially, a dotted line appears around each image.

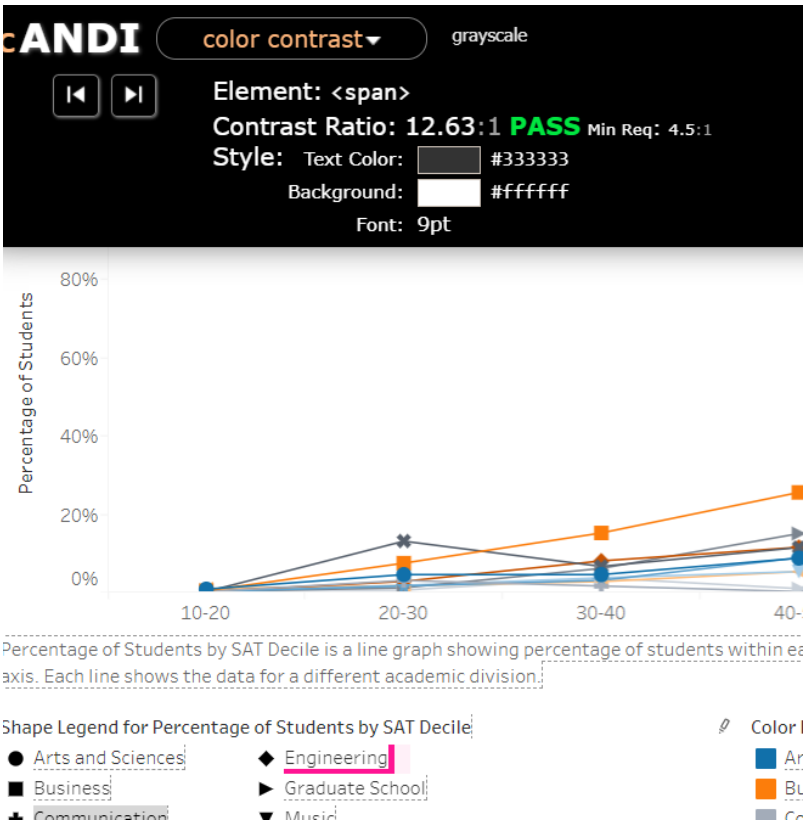

2. Verify the image is meaningful to the page and if so, is the accessibility output descriptive.




***Contrast***

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1.

1. Select the ANDI: graphics/images module to identify all images. Initially, a dotted line appears around each image.
2. Verify the contrast ratio passes with at least 4.5:1.




***Success Criteria tested:***

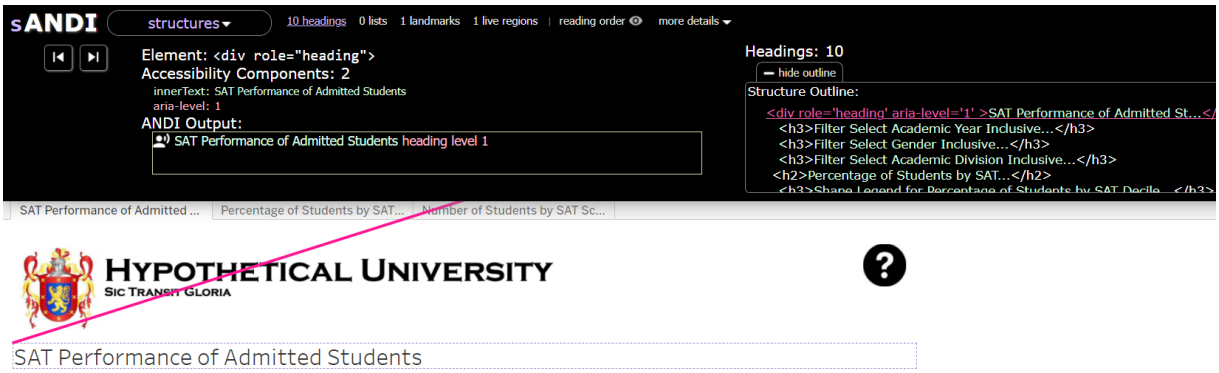
- 1.1.1 Non-text Content - All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.
- 1.4.3 Contrast (Minimum) - The visual presentation of text and images of text has a contrast ratio of at least 4.5:1.

*Page Structure*

***Headings***

Verify the visual content structure of headings and lists are programmatically determinable and coded properly.

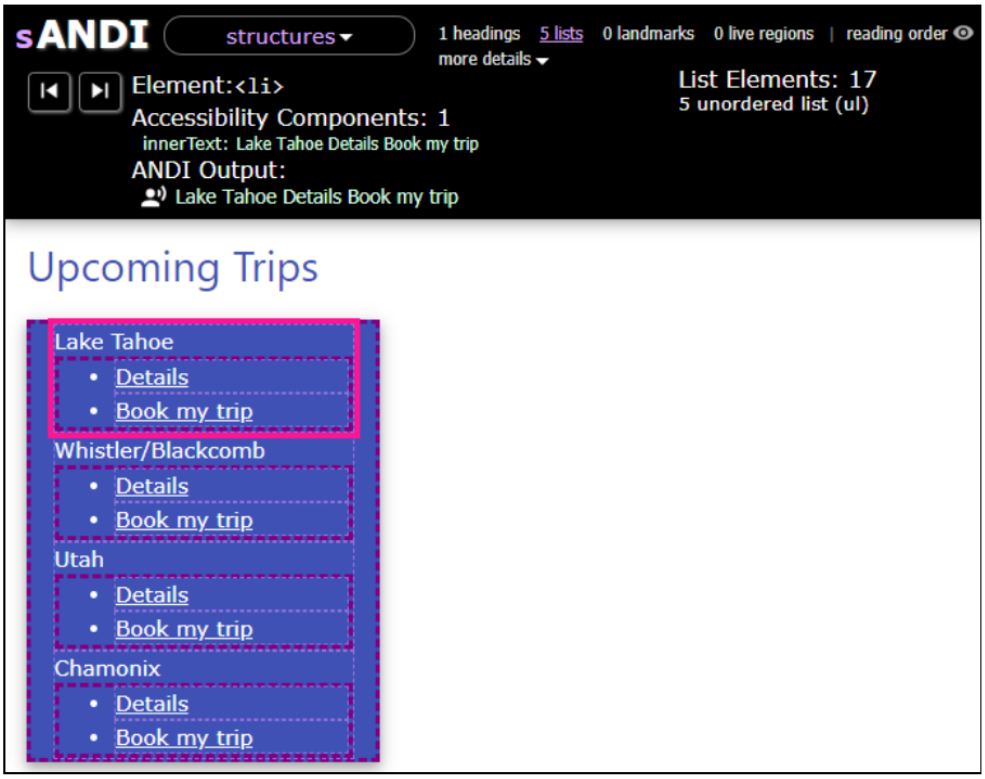
1. Identify all visually apparent headings, which denote sections of content on the page. Headings are often in larger, bolded font and separated from paragraphs of text or other page elements with extra spacing.
2. Select ANDI: structures and then the headings button.
3. Look for defined headings <h1> to <h6> or ARIA role=heading.




***Lists***

Ensure that all visual lists are programmatically identified as a list.

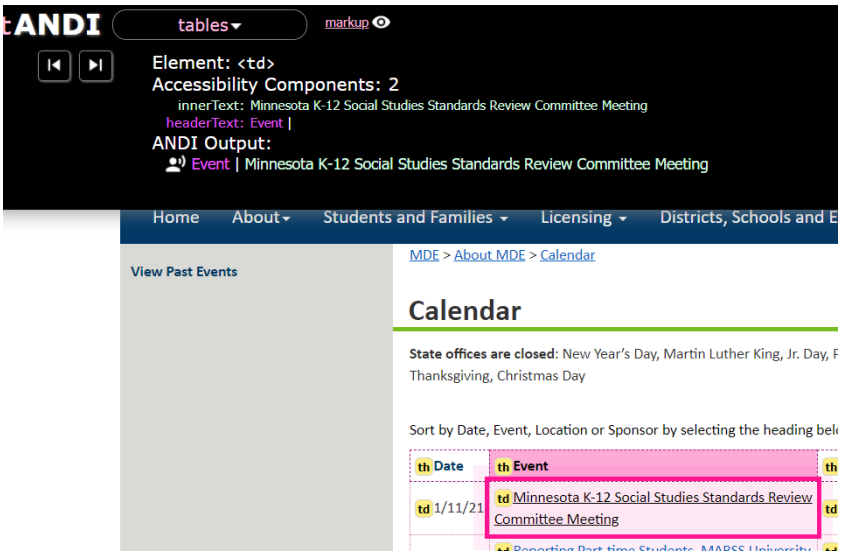
1. Identify all visually apparent lists. A list that can appear in any order is often, but not always, visually formatted with some form of a bullet symbol. A list that needs to be in a specific order is preceded by a number.
2. Select ANDI: structures and then the lists button.
3. Look for grouped items, like menus and submenus. Vertically presented items.




***Tables***

For this test, you are going to ensure that when a data table is used, it has programmatic table markup.

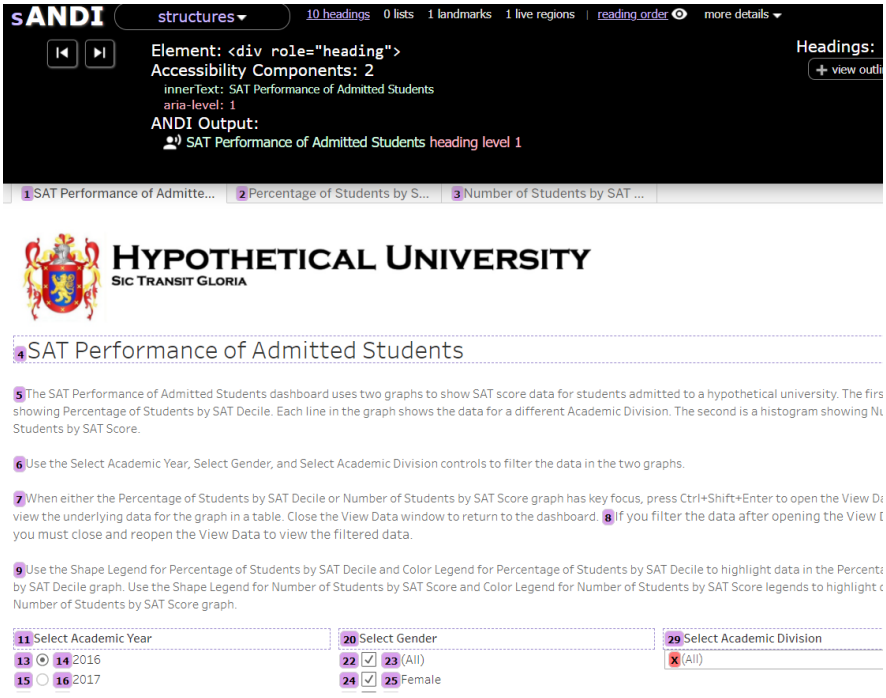
1. Select ANDI: tables. Note, if the “tables” module does not display in the dropdown menu, ANDI has not detected any data tables.
2. If the “tables” module does display, choose “Analyze Next Table” to go through the tables on the page.
3. Verify data tables include table headers and proper cell markup.




*Tabs/Sections/Worksheets*

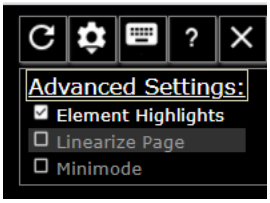
***Meaningful Sequence – reading order***

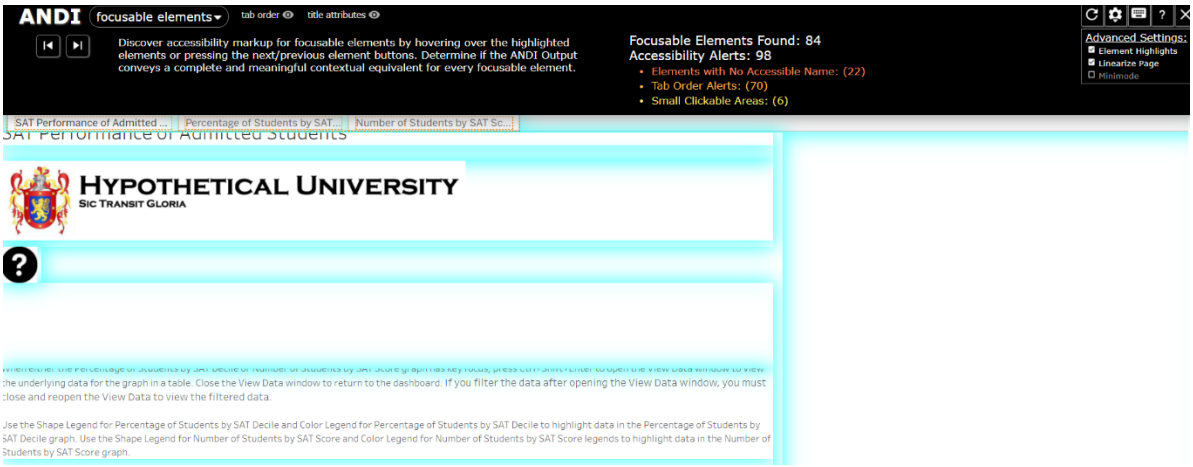
1. Select ANDI: structures and then the reading order button.
2. Review the numbers for the reading order of elements on the page and verify the order makes sense.



***Meaningful Sequence – without CSS positioning***

1. Select ANDI: Advanced Settings icon: Linearize Page.
2. Review all highlighted linearized content and determine whether the reading order of content is still logical and understandable.
3. Verify data tables include table headers and proper cell markup.

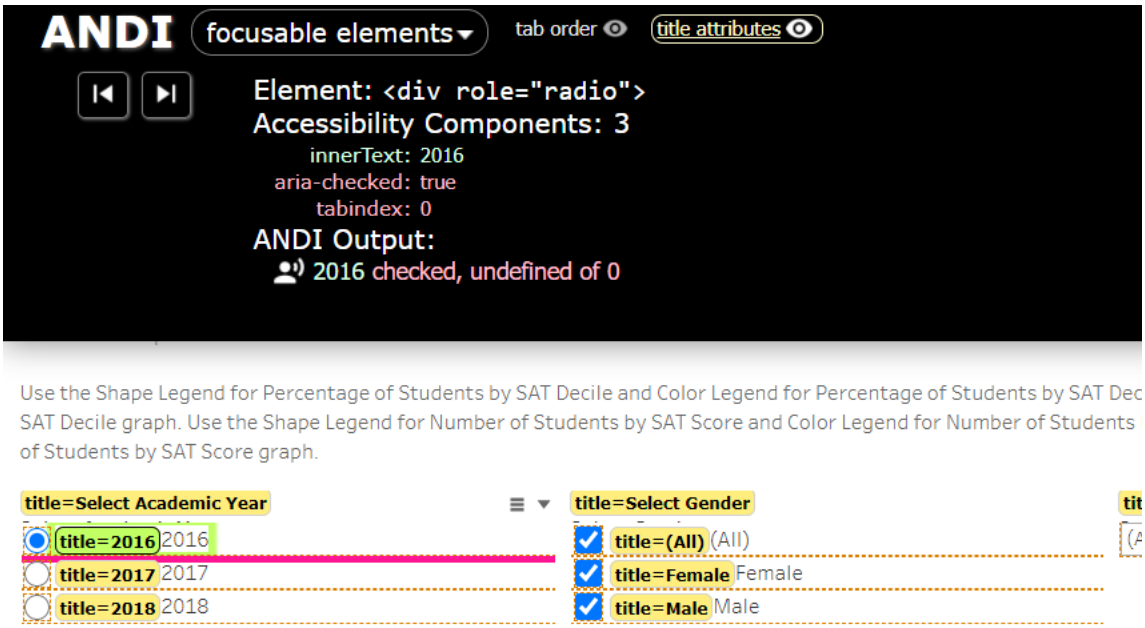



***Keyboard***

All functionality and all information that is essential or required to complete an activity can be accessed using only the keyboard.

1. Review the page and use a mouse to determine what interactions and available functions are provided.
    a. You can use the ANDI: Focusable elements: title attributes button to help you identify. Identify where the title attribute provides information that is essential to understanding or operating the page content.
2. Identify any information that is essential or provided with only a mouse.
    a. You can use the ANDI: Focusable elements: title attributes button to help you identify. Identify where the title attribute provides information that is essential to understanding or operating the page content.




3. Using only the keyboard, verify all the items identified above are also available via keyboard and are functional with only the keyboard.

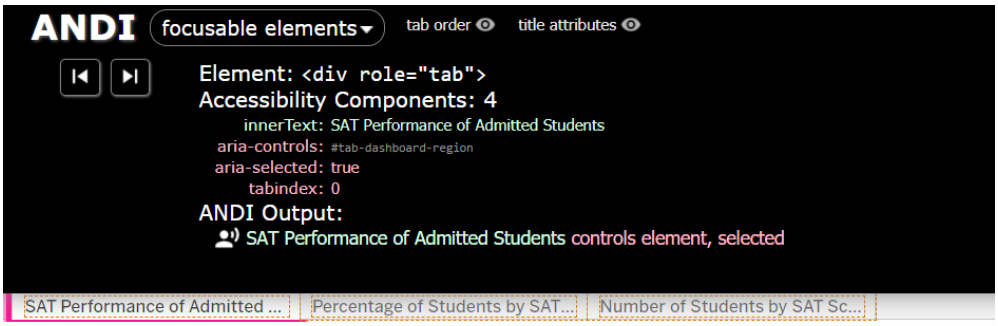
***No Keyboard Trap***

1. Tab through the entire page using only the keyboard. Determine if users become trapped by an element, revealed content in a section of the page.
2. Determine whether there are any instances where keyboard navigation becomes trapped:
    a. Keyboard users are unable to move away from an element, for example pressing the TAB or arrow keys.
    b. Keyboard access is restricted to a small section of the page with no way to navigate out of the "loop" to the rest of the page.
3. If a keyboard trap is found:
    a. Inspect any help (contextual help, or application help) and documentation for notification of available alternate keyboard commands (e.g., non-standard keyboard controls, access keys, hotkeys) to escape/avoid the keyboard trap.
4. Determine whether the alternate command(s) work.

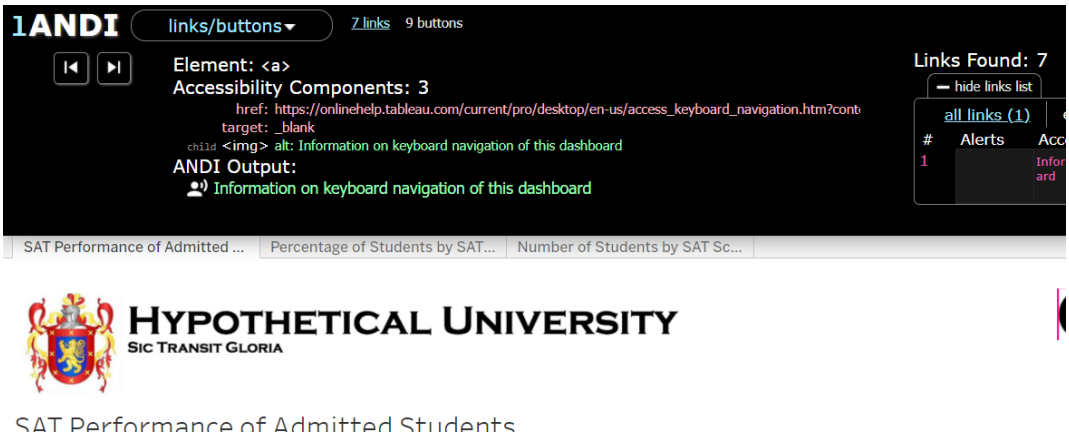
***Bypass Blocks***

A keyboard-accessible way is provided to bypass repetitive content. Identify blocks of content that are repeated. This may include navigation links, page headers, tabs, and banners.

1. Tab through the entire page using only the keyboard. Remember, some bypass blocks may not be visible until they receive focus.
2. Use ANDI: focusable elements to check for skip links, hide options collapse menu, and other elements with similar bypass functionality.

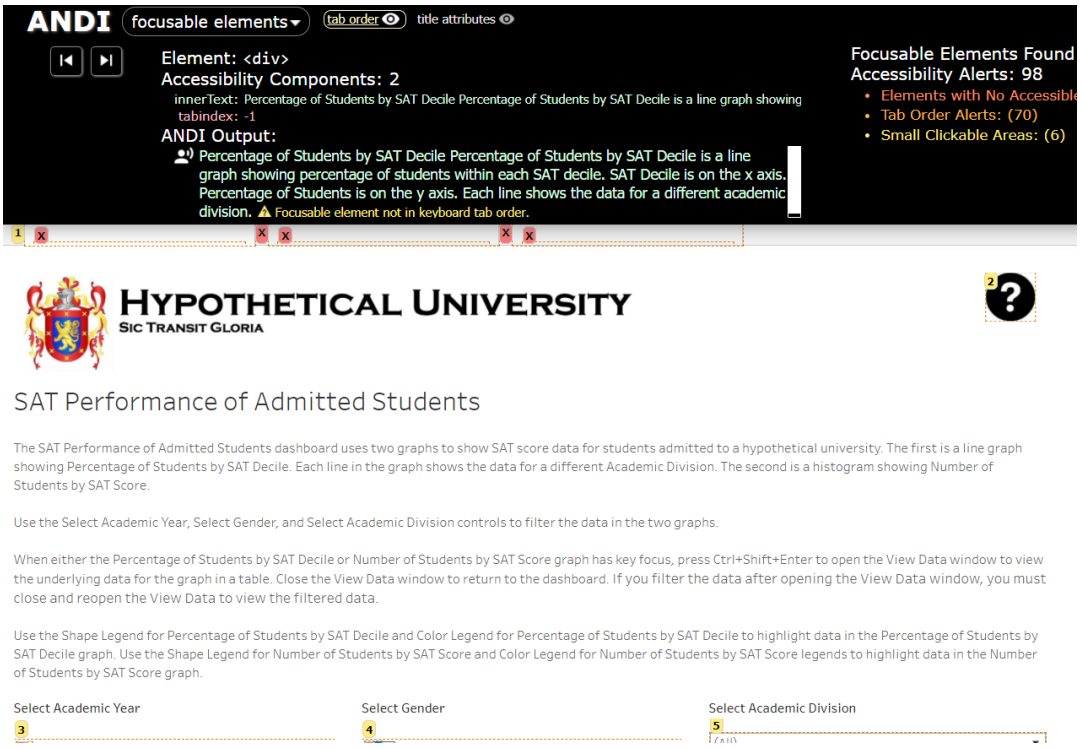



3. You can also use ANDI: links/buttons and the "View links list" button to help identify if there are skip links or other elements not easily identified visually or through tab order



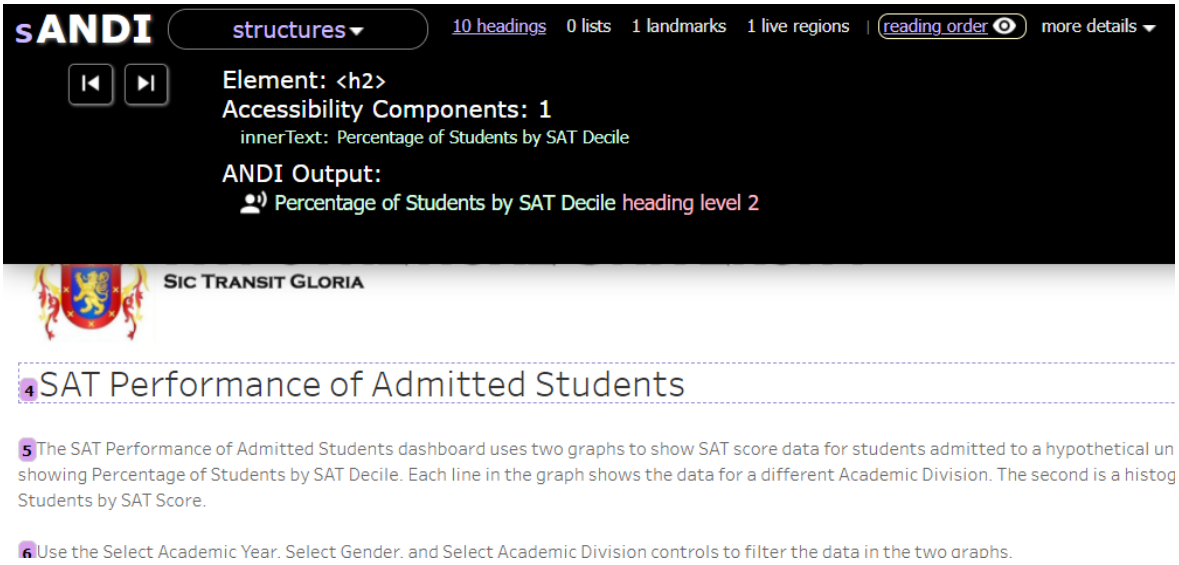

***Focus Order***

1. Use the tab key to move focus through the page. Determine if the focus order matches the logical order of page elements and preserves the operability of the page.
2. It may be helpful to use ANDI: focusable elements then tab order button to identify the order of elements on the page.




***Consistent Navigation***

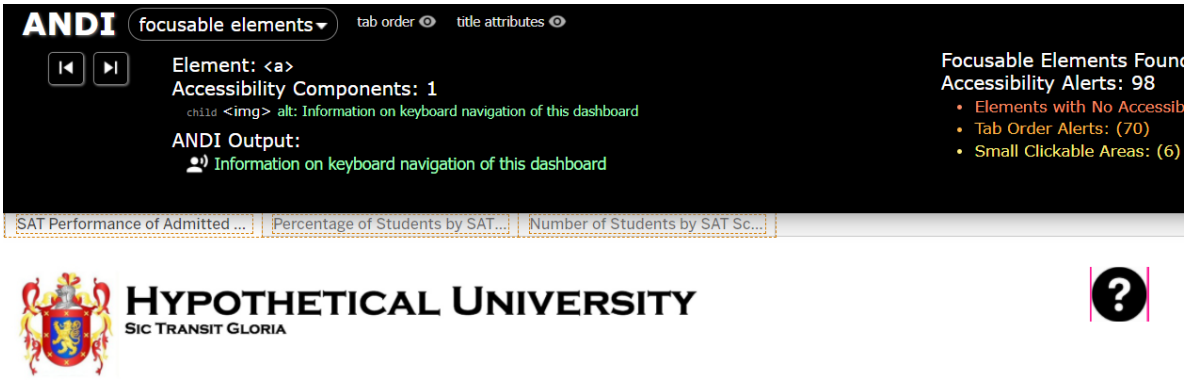
1. When multiple dashboards are included (tabs), review the dashboards to identify navigational components that are repeated on multiple pages.
2. Review the order of the navigational elements and compare it to the order on the other pages where they appear.
3. Use ANDI: focusable elements then choose the “tab order” button. This may help evaluate the focus order of interactive interface components.
4. Use ANDI: structures then choose the “reading order” button to help evaluate the content order of both focusable and non-focusable components.




***Consistent Identification***

When multiple dashboards are included (tabs), review the dashboards to identify components that have the same functionality across the dashboards. The same functionality means that the components produce the same set of results.

1. Open ANDI: focusable elements.
2. Examine the ANDI Output for each of the identified elements and verify the ANDI Output is consistent across all dashboards.




***Success Criteria tested***

- 1.3.1 Info and Relationships - Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.
- 1.3.2 Meaningful Sequence - When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.
- 1.4.1 Use of Color - Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

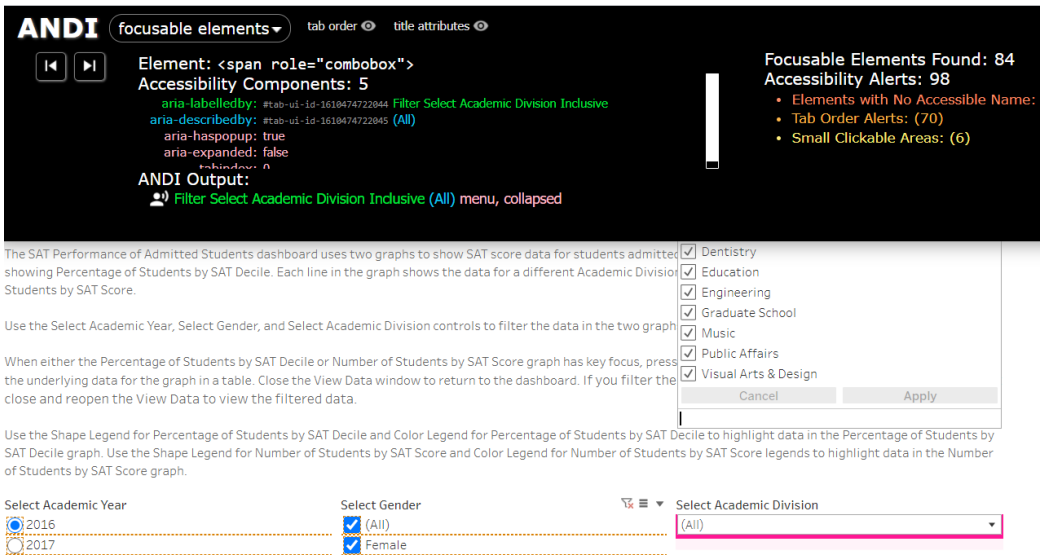
- 1.4.3 Contrast (Minimum) - The visual presentation of text and images of text has a contrast ratio of at least 4.5:1.
- 2.1.1 Keyboard - All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- 2.1.2 No Keyboard Trap - If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving the focus away.
- 2.4.1 Bypass Blocks - A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.
- 2.4.3 Focus Order - If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.
- 2.4.6 Headings and Labels - Headings and labels describe the topic or purpose.
- 3.2.3 Consistent Navigation - Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.
- 3.2.4 Consistent Identification - Components that have the same functionality within a set of Web pages are identified consistently.

### *Filtering components and other form elements*

#### ***Label Provided***

Verify that form fields have on-screen labels or instructions. When an interactive form field is included on the dashboard, it must have a label or instructions.

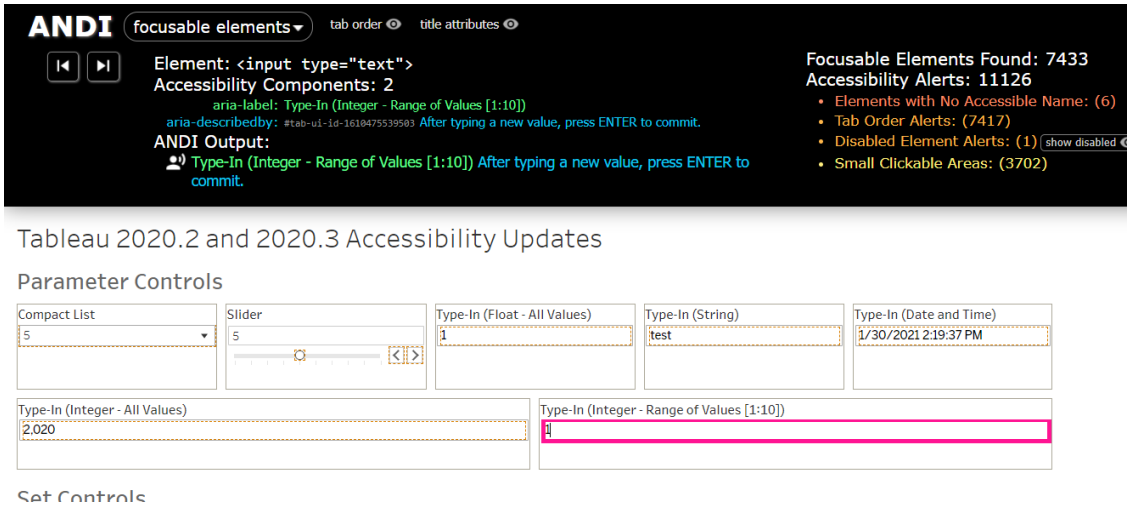
1. Open ANDI: focusable elements. Identify any form elements on the page, for example text fields, radio buttons, checkboxes, and multi-select lists.
2. Look for all instructions and cues that are related to form components/controls.
3. Verify the labels and/or instructions are provided for each form element.




***Descriptive Label***

Verify that if a form element has an on-screen label and/or instructions, the on-screen label and/or instructions are sufficiently clear and descriptive.

1. Open ANDI: focusable elements. Identify any form elements on the page, for example text fields, radio buttons, checkboxes, and multi-select lists.
2. Look for all instructions and cues that are related to form components/controls. For example, groupings, order of completion, special conditions, or format instructions.
3. Verify the labels and/or instructions sufficiently describe the purpose and applicable data requirements, i.e., data formats, required fields, etc.




***Programmatic Label***

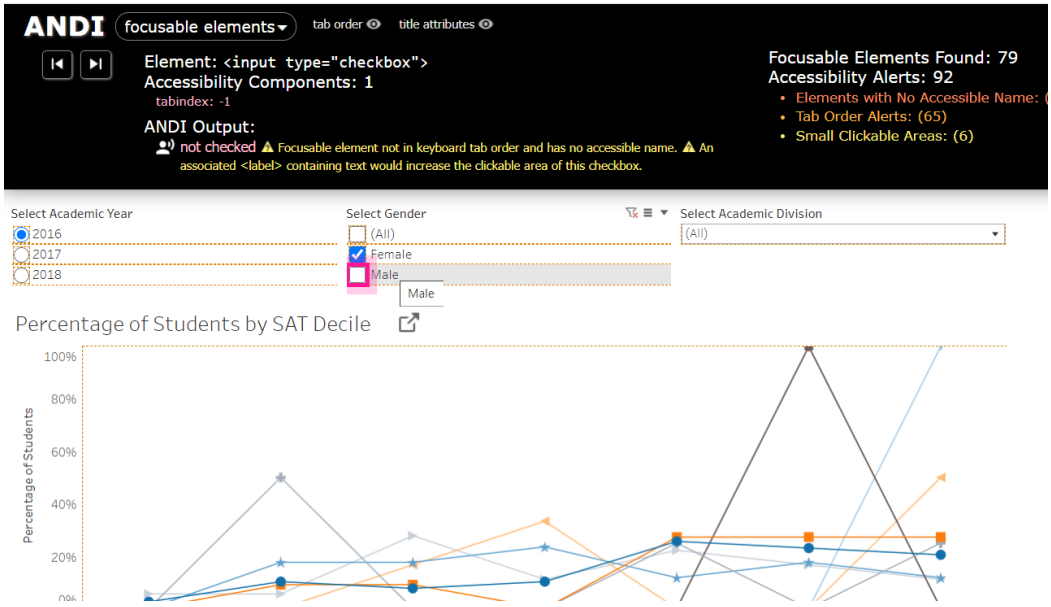
Ensure that each form input element has programmatic associations, that also accurately describe the form element and include all relevant instructions and cues.

1. Open ANDI: focusable elements.
2. Review the ANDI Output for all focusable form fields and verify all labels and visible cues and instructions are identified in the ANDI Output. Note: if any changes to form elements occur automatically or as a result of an interaction, the information should be included in the output.

***Change Notification***

Determine whether the dashboard provides sufficient notification and description of any changes to the content on the dashboard that occur as a result of interaction with a form element.

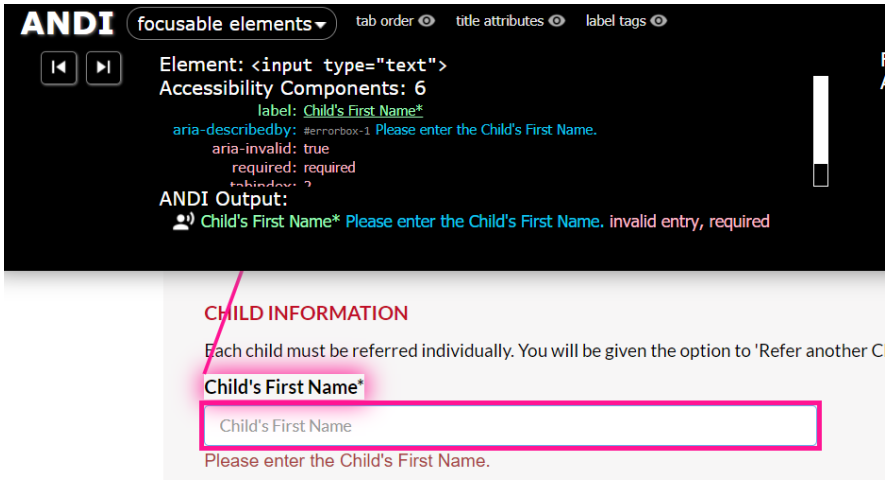
1. Open ANDI: focusable elements. Identify any form elements on the page, for example text fields, radio buttons, checkboxes, and multi-select lists.
2. Look for all instructions and cues that are related to form components/controls. For example, groupings, order of completion, special conditions, or format instructions.
3. Identify any content changes that occur on the page as a result of interaction with form elements. For example, changes to other content or controls, error messages, textual information, new content, or content removed.
4. If the content changes are the direct result of a user's interaction with a form element, and this is expected behavior and located within the same control, typically, this is ok. However, if there is a dialog that appears or a change to the page and page focus is not taken to the element, this is a failure. For example, a chart on the page that changes after selecting items within a filter component.




***Error Identification***

Ensure that users receive the cues and instructions needed to understand what form input field is in error whenever automatic error detection is provided on form elements.

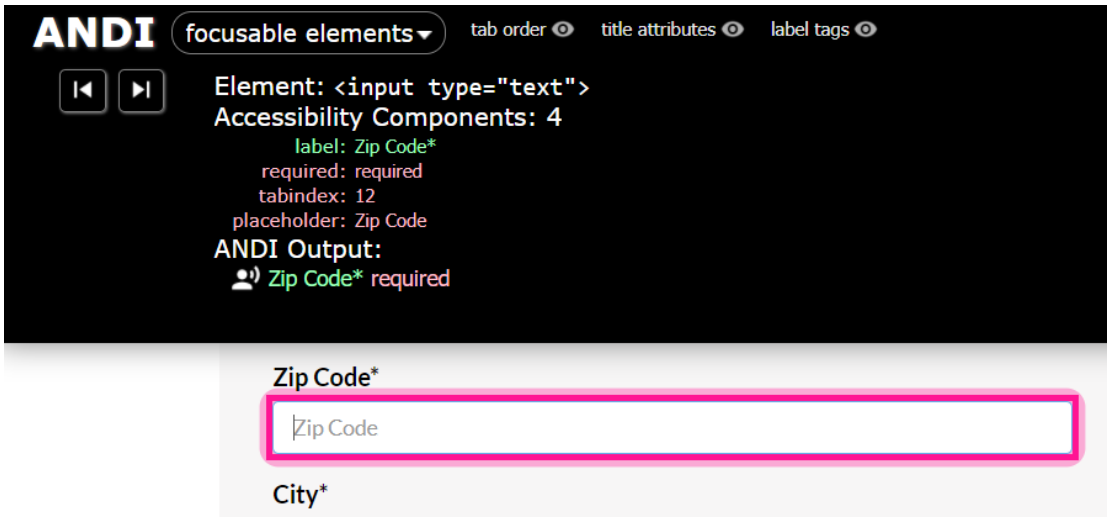
1. Look for form components/controls on the page. Intentionally violate formatting and other form instructions on the page. For example, enter a different date format, do not fill in required fields, and attempt to submit the form or move to the next page to determine the error.
2. Open ANDI: focusable elements. Verify the error is identified and described in the text and included in the ANDI Output.




***Error Suggestion***

Validate that when automatic error detections occur, users receive guidance for correction on an input error (when possible).

1. Look for form components/controls on the page. Intentionally violate formatting and other form instructions on the page. For example, enter a different date format, do not fill in required fields, and attempt to submit the form or move to the next page to determine the error.
2. Open ANDI: focusable elements. Determine if the guidance provides enough detail for how to correct the error and/or gives suggestions for corrected input.




***Success Criteria tested***

- 1.3.1 Info and Relationships - Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.
- 2.4.6 Headings and Labels - Headings and labels describe the topic or purpose.
- 3.2.1 On Focus - When any user interface component receives focus, it does not initiate a change of context.
- 3.3.1 – Error identification – If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.
- 3.3.2 Labels or Instructions - Labels or instructions are provided when content requires user input.
- 3.3.3 Error Suggestion – If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.
- 4.1.2 Name, Role, Value – For all user interface components, the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

### *Visualization (Charts and Graphs)*

***Including testing keyboard shortcut works to bring up data HTML table.*** At this time, Tableau visualization labels and data points are not accessible for assistive technology users. Ensure you are following the testing outlined for visual tests and keyboard tests.

- Follow the guidance in the Non-text content and Contrast sections, above, to test for text alternative and color contrast.
- When background data tables are available, the keyboard shortcut Ctrl + Shift + Enter will show that data with the filtered results. This shortcut should be identified in every dashboard. Follow the guidance in the section above to test this data table keyboard shortcut is available.
- In addition, follow the guidance in the sections above to test for keyboard navigation, keyboard traps, focus visible, and on-focus.
- You can check the responsiveness of the visualizations on the dashboard using the guidance in the section.

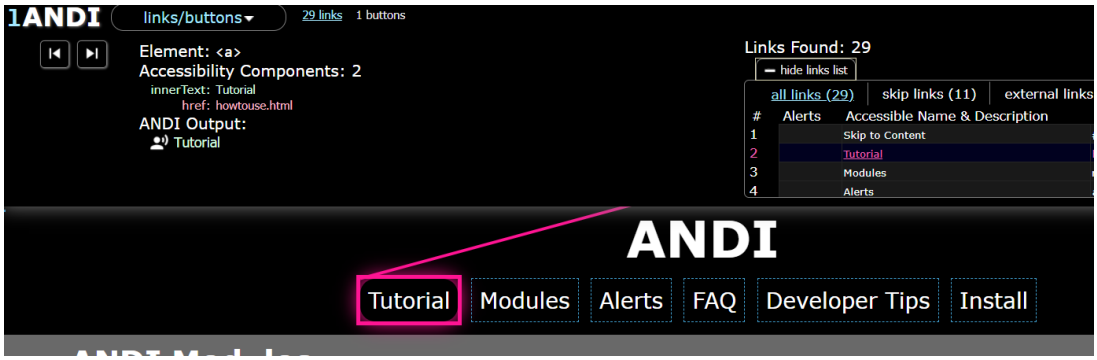
### *Page Text*

The ANDI tool can't be used to test text added to the dashboard, however, there are other tests you should do.

- Follow the guidance for visual testing to check for the use of sensory characteristics and use of color.
- Use the Colour Contrast testing outline for verification the foreground and background color meet the contrast ratio of 4.5:1.
- You can check the responsiveness of the page text on the dashboard using the Browser Zoom guidance

### *Links*

1. Open ANDI: Links/buttons then select the links button to identify all the links.




2. Determine whether the ANDI Output, along with the programmatically determined link text, describes the link purpose or function.

**Note**: follow the Colour Contrast testing outline for verification the link text color (in all states) meets the contrast ratio of 4.5:1.

***Success Criteria tested***

- 2.4.4 Link Purpose (In Context) - The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.