Chartability

This workbook is intended to serve as a lightweight, high-level audit for catching data experience\* design failures related to accessibility. Note that **CRITICAL** tests in this workbook are not the “most important” tests (as every test is important in Chartability) but rather tests that are either: very common or expensive to remediate. For more on Chartability, see: <https://chartability.fizz.studio/>

\*A data experience (DX) could refer to a data visualization such as a chart, graph, or plot, a “bespoke” (highly customized) graphic based on data, a model, or an algorithm, or a data driven interface or system.

# Method

This workbook organizes tests into seven categories, four of which are based on broad accessibility principles: Perceivable, Operable, Understandable, and Robust (POUR). The last three extend Robust: Compromising, Assistive, and Flexible (CAF).

An auditor should proceed through each test and determine whether the content fails or succeeds using techniques provided or their own methodology. The auditor should assume failure when tests are inconclusive and (if early in the design process) a proven pass to a future consideration has not yet been demonstrated.

# Failures

Unlike a compliance audit (which determines adherence to regulatory guidelines), Chartability is meant to identify design failures. Generally, Chartability approaches accessibility as a scale rather than a state: how accessible a DX is is determined by how few failures it contains. It should be assumed that even the absolute best DX may contain several failures, even after remediation. Note that Chartability should never be used in place of a compliance audit but always in tandem with it.

Chartability’s insistence on a scale (instead of a state) of accessibility requires that designers and creators consider their choices carefully: they must be willing to argue that lack of scope, time, or research or perhaps a unique consideration led to a given failure. No failure should be left unconsidered.

# Outcomes

Chartability is an auditing tool that not only highlights failures for remediation, but also encourages healthy design critique and critical design discourse related to accessibility. Because excellence in accessibility is never a finished state, the outcome of a Chartability audit is less about *arriving* at accessibility and more about working *towards* an accessible experience.

# Perceivable Failures

User must be able to easily identify content using their senses: sight, sound, and touch.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: Low contrast on DX elements and text. | Geometries and large text must have >3:1 contrast against background, Regular text must have >4.5:1. Use [WebAIM Contrast Tool](https://webaim.org/resources/contrastchecker/) or dropper contrast tool.  |  |
| CRITICAL: Text size too small. | Text (any) must not be smaller than 12px in size. |  |
| CRITICAL: Information only available visually (no screen reader/braille support). | At a minimum test using [JAWS](https://www.freedomscientific.com/products/software/jaws/) + Internet Explorer 11+ and [NVDA](https://www.nvaccess.org/download/) + Chrome on Windows, VoiceOver + Safari on Mac. Must be able to access DX information. All annotations, “visually apparent” trends or features, and all major narrative elements must be exposed to screen readers. |  |
| Color choice is not “colorblind safe” (inaccessible to people with color vision deficiencies). | Use [Viz Palette](https://projects.susielu.com/viz-palette) or [Chroma](https://gka.github.io/palettes) to test palette. Must not have major warnings on either. |  |
| Color is the only channel for meaningful information. | Categorical only: Textures, shapes, or size (for filled elements) or dash patterns (for lines and paths) are required.  |  |
| Meaningful elements cannot be discerned. | Primary DX elements must not be obscured by other elements (only a failure if [discriminability](https://homepage.divms.uiowa.edu/~luke/classes/STAT4580/percep.html#discriminability) or [separability](https://homepage.divms.uiowa.edu/~luke/classes/STAT4580/percep.html#separability) is required to understand the DX). Elements must have at least 1px white space between on “stacked” DXs (like stacked bars or pie DXs where elements “touch”). Text (any) must not be obscured or overlapped by any other elements. |  |
| Inappropriate use of spacing. | Too much or too little white space on DXs with intervals (like a bar DX with thin bars and large gaps or vice a versa). |  |

Notes:

# Operable Failures

All controls must be error-tolerant, discoverable, and multi-modal (not just mouse operable, but using keyboard, etc).

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: DX interactivity is not available through multiple means. | If DX is interactive it must also be made interactive to keyboard only and screen reader users. Test navigating to DX with keyboard, using tab and arrow keys. If DX elements are interactive with a mouse, they must also be interactive with a keyboard. Focusing should mirror hovering, selecting (enter or spacebar) should mirror clicks. |  |
| CRITICAL: Instructions not provided for interaction | If DX has any interactive capabilities at all, it must be explained somewhere for users to understand. All keyboard controls must also be explained as well. |  |
| Contrast is low on interactive elements. | All interactive elements must have at least 3:1 contrast against background. Use [WebAIM Contrast Tool](https://webaim.org/resources/contrastchecker/) or dropper tool. Change in hover/focus and click/select states must not use color alone (must change stroke thickness, dash pattern, or size.) |  |
| Keyboard focus indication is hard to see. | Focus indicator must have 4.5:1 contrast against background, must not be fully obscured, and must have at least a 2px border. Use [WebAIM Contrast Tool](https://webaim.org/resources/contrastchecker/) or dropper tool. |  |
| Inappropriate use of TAB stops. | Every DX element must NOT have its own tab stop (DX cannot be skipped). At least one tab stop should be provided if a data table succeeds the DX. |  |
| Custom keyboard controls override screen reader settings. | Any custom key controls must only apply when the DX or elements have focus (no page or app overrides). |  |
| Special actions (brushing/zooming/filtering) that use custom DX controls have no UI alternative. | All custom DX controls (brushing, zooming, filtering, selecting, etc) must also be exposed for access by alternative UI controls. These controls must be clear and easy to use with a keyboard and screen reader. |  |

Notes:

# Understandable Failures

Any information or data are presented without ambiguity, with clarity, and in a way that minimizes cognitive load.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: If DX is novel or unconventional and does not have an explanation. | DX should have an explanation for how to read, use, and interpret it. |  |
| CRITICAL: No title, summary, context, or caption is provided. | A title, summary, context, or caption must be provided. |  |
| Axis labels are not present or unclear. Axis is truncated without a clear label. | In rare cases axes may be removed (if adequate text explanation or annotation is provided). Otherwise, axis should be present and clearly labeled. Axis labels may be abbreviated but with a clear convention. Axis may only be truncated if truncation is called out with a label. |  |
| Inappropriate controls are provided for the task. | Controls must not be irrelevant to the message, question, or task of the DX. DX scope and functionality must not be too broad. Fail if the DX can be subdivided or have irrelevant functionality removed. |  |
| Metrics or variables are misleading or undefined. | [DX must not lie](https://www.amazon.com/How-Charts-Lie-Getting-Information/dp/1324001569/) and data (and source) must be defined. Metadata, metrics, calculations, and variables must be defined. |  |
| Statistical confidence/uncertainty is not clearly and unambiguously communicated. | If any statistical confidence interval exists, it must use clear conventions and provide textual explanation. |  |
| Information complexity inappropriate to the task or goal of the visual. | DX must not have more than one Y or X axis without first presenting two DX separately. DX must not encode along a third spatial dimension (z axis) unless the data itself is 3D (sensory, modeling, etc). DX should not contain more than 5 data categories.  |  |
| If DX interacts with surroundings, this purpose and function is not clearly communicated. | If DX can affect the logic or layout of the page or if receives data or parameters from other UI controls or logic, this must be clearly communicated in text. |  |
| DX changes are not easy to follow. | If DX’s data change is meaningful, it must employ animations for object constancy (no faster than 200ms or longer than 2s). Changes to state must be announced to screen reader users. |  |

Notes:

# Robust Failures

The design is compliant with existing standards and works with the user’s compliant, assistive technologies of choice.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| Does not follow compliance standards. | The DX must pass all relevant WCAG 2.1, Section 508, or equivalent requirements. (This is intended as an automatic failure until the DX can be fully evaluated.) |  |
| Semantically invalid use of document elements (if it functions like a button, but it is semantically other than a <button>, etc). | DX must be semantically valid according to modern standards (if delivered on the web). Initial testing may be automated using any combination of: [Axe-core](https://github.com/dequelabs/axe-core), [HTML Codesniffer](https://squizlabs.github.io/HTML_CodeSniffer/), [Accessibility Insights](https://accessibilityinsights.io/), or [W3C Markup Validation](https://validator.w3.org/) but may only pass once a screen reader test has also verified the experience (see: *Perceivable Failures* for screen reader info). |  |
| DX “works best” on only one browser, device, or operating system. | DX access must not be isolated to one browser, device, software, or operating system. There must be a diversity of technological means to access the DX and its information. |  |

Notes:

# Compromising Failures

Information can be approached at both a high and low level. Layout, flow, actions, and tasks all have alternative routes.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: A table is not provided that the DX is based on. | A table must be provided that contains a human-readable version of the data the DX is based on. This may be excluded if the DX title, summary, context, or annotations are sufficient at conveying all relevant information contained in the DX. |  |
| Provided table is not downloadable, filterable, or sortable. | If a table is provided, it must be easily downloadable or it must contain robust filter and sorting capabilities. |  |
| There is not more than one process available to reach the same information. | If DX is contained within or participates in complex user interface flows, such as transitions between views or states, there must be alternative paths to reach that same state (such as with search features, parallel UI controls, etc).  |  |
| DX state is not easy to share and reproduce. | If an analysis or complex interaction can produce a customized view, this view must be easy to share without instructions (such as with a single link, file, or saved state). |  |
| Current view path is not easy to understand or return to. | Similar to “more than one process” and “easy to share and reproduce,” current view and state of visualization (in a complex environment like a dashboard or app) must provide the user with bread-crumbs to guide their path as well as the ability to save, reload, and navigate history. |  |
| DX data cannot be navigated according to its data structure. | DX data that contains sub-grouping (like a stacked bar DX) or nesting (like a treemap or hierarchy) must provide keyboard navigation that can navigate between levels and/or laterally across levels (in a non-linear fashion). Keyboard navigation must be comparable to the data structure (including cases where the data structure is novel) as well as provide linear or tabular navigation (like in a table or list). |  |
| Interactions and operations are not forgivable. | When the visualization is interactive or has the ability to perform a task, the user must be able to undo or redo their actions. |  |

Notes:

# Assistive Failures

“Heavy lifting” (difficult/manual statistical or data-related) tasks are automatically handled or assisted whenever possible.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: Data is not presented at an appropriate density. | If more than 20 elements are competing for the same space (approximate limit is based on cognitive load): clustering or patterns (or lack of) must be explained, DX must be aggregated to a higher level with less elements, or DX must be divided into smaller DXs with less data. Most dense DXs should be summarized and aggregated or enable splitting into different views. |  |
| “Visually apparent” features are not described through text summaries or via sonification. | Trends, clusters, patterns, outliers, or significant statistical findings that are considered “visually apparent” must be described through text. Optionally, these features may also be exposed using sonification. |  |
| Use of space inappropriately handles extreme difference or similarity in the data. | Both extreme statistical differences and similarities can produce unreadable DXs. If DX elements are squished into margins due to outliers or together by too much similarity, this fails. DX must automatically handle these issues or else it must be made clear to the user through annotations what is happening. If data is dynamic or producing automatic annotations is not possible, then DX must provide a way for the user to sort, divide, or filter the DX space on their own.  |  |
| If DX must be built by the user, no default, opinionated state is given as a starting example. | If the user is required to craft their own DX (say by combining variables in an analytic environment), a default, opinionated view of the data must be provided as a starting point. |  |
| Data is not formatted to be human-readable. | All textual information displayed (in data labels, annotations, axes, tables, legends, etc) must be formatted to an understandable level of content (ie “human readable”). These formats must also be made into versions that can be read and parsed comfortably by screen readers. (For example: 6500000000 should be formatted to 6.5b visually and to “six point five billion” when used in screen reader labels and alt text.) |  |

Notes:

# Flexible Failures

Styling and presentation can be altered and adjusted according to user preference. User changes are respected.

|  |  |  |
| --- | --- | --- |
| Description | Test for failure | Failure? |
| CRITICAL: Styling changed by the user is not respected. | DX must not interfere with or override styling changes made by the user (such as importing a custom style sheet for use in an HTML application or web site). |  |
| Longer, video-style or explanatory animations cannot be paused, stopped, and started over. | Animations lasting more than 2 seconds or any looping animations must be able to be paused or stopped. Animations used to communicate transitions in the data that last more than 2 seconds must be provide a way for the user to start over. |  |
| Scrolling experiences cannot be adjusted or opted out of. | Infinite scrolling, parallax scrolling, and “scrollytelling” experiences must come with the ability to be turned off or used optionally, with an option like “load more” or “next” in its place for keyboard only users. |  |
| DX space cannot be zoomed. | It must be possible to zoom in and out of the DX space. Text, geometries, and all elements must change size appropriate to the type of zoom used. |  |
| Contrast or textures cannot be adjusted as-needed | User must be able to change contrast as needed. DX must not interfere with or override user’s independent contrast adjustments and DX must adjust accordingly to new settings. DX textures (such as those used on fills) must be able to be turned on or off according to user preference. |  |
| Text spacing and font-size changed by the user is not respected. | DX must not interfere with programmatic changes to font sizes or text spacing, such as importing a custom style sheet or using a browser’s build in zoom function. Font size and spacing must adjust accordingly. |  |
| Design is not consistent and familiar by default. | DXs must be made consistent with one another across an application or environment, including sharing default styling and settings as well as those set by the user. |  |

Notes: