Web based graphics chart library

"Designed for data visualizing of NHN web accessibility reporting system."

Developed by

Insook Choe
Hansol Kim

Index

Introduce.

Development of 'wagon', a chart library was initiated as (we were) seeking a good visualization mechanism for presenting accessibility state reports within N-WARS (NHN Web Accessibility Reporting System). Given the business requirements of the report, polygon shape of chart was seemingly the best choice.

Although free chart libraries are available, we were unable to find one that meets the business requirements and it was extremely inefficient to purchase a license in order to add or modify the functionalities as needed. Therefore we've decided to build a simple and easy-to-use chart library.

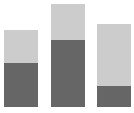
The name 'wagon' is composed of the acronym of Web Accessibility, and 'gon' as in polygon since the development was initiated while attempting to create a polygon shaped chart for web accessibility state. But more importantly the name 'wagon' is to imply our intention and effort to make web services that can be used by people with disabilities as comfortably as they are on a wagon ride.

Source repository: http://dev.ui.nts-corp.com/svn/a11y/lab/chart/

KEY POINTS

Easy to use

· easy data preparation(takes JSON object as function parameter)
· visual chart is one function call away
· easily changeable color theme

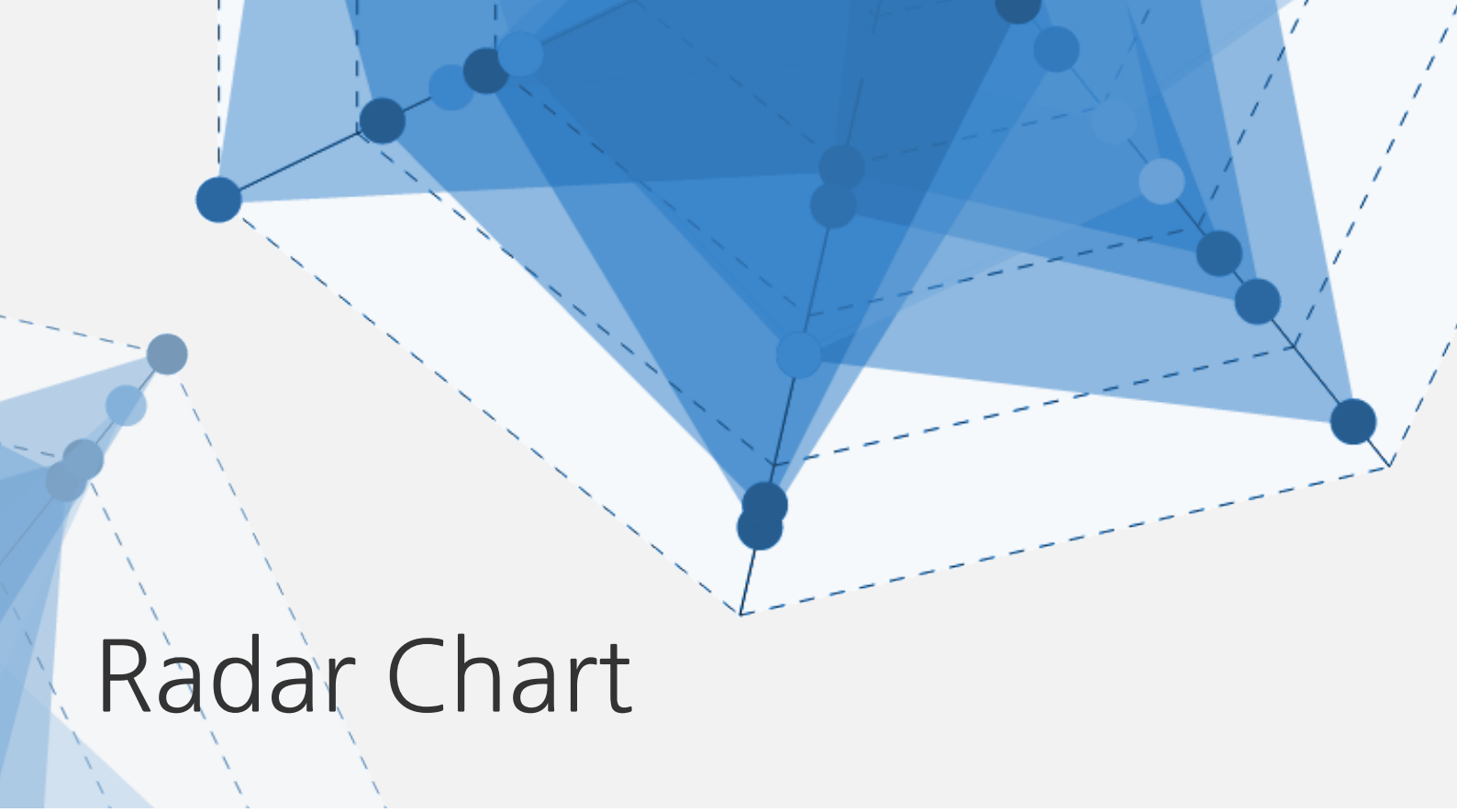


Simple yet beautiful modern design

Fast rendering with SVG (Scalable Vector Graphics)

Support all modern browsers (Chrome, FireFox, Safari, IE since v.9, etc)

Radar Chart



Radar Chart

Overview

A radar chart a.k.a spider chart or web chart or many other names serves a great visual presentation for comparing 3 or more data with the same measurement metric and displaying current states of data as compared to what they need to attain. Wagon's radar chart becomes extremely helpful when comparing multiple datasets within one chart area with the areas formed by vertexes on the radar.

JSON syntax

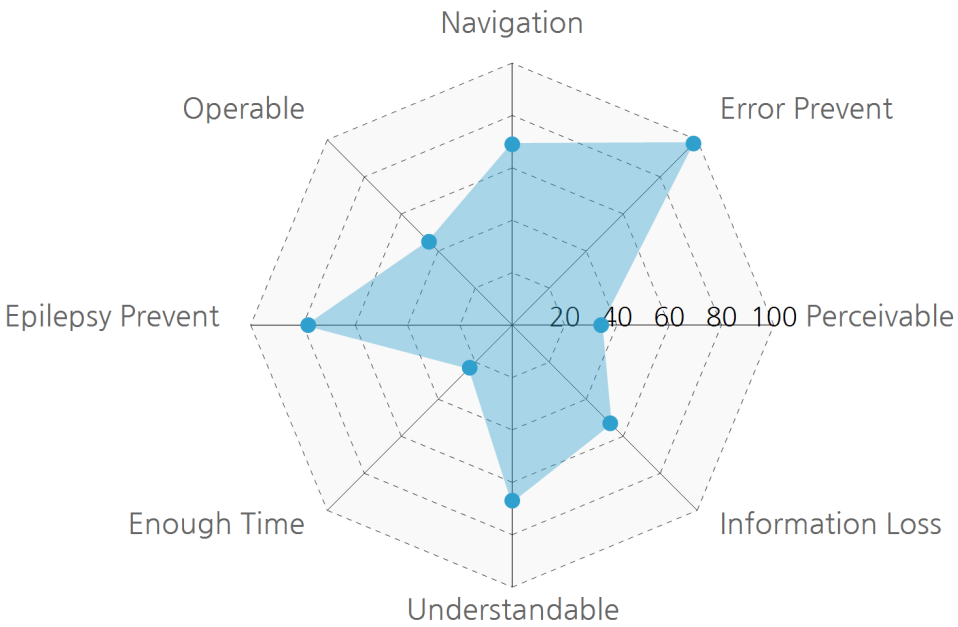
Radar chart takes a JSON object as its only parameter and the keys and values are as following.

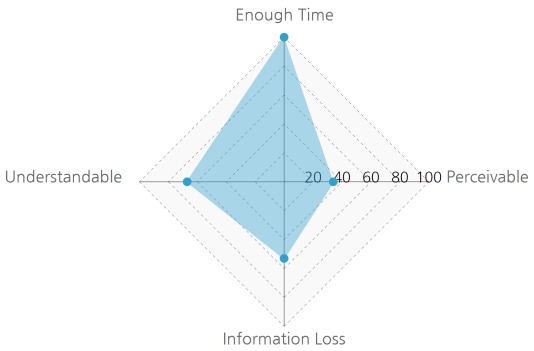
- `'legend'`: JSON object with 2 `'key: value'` pairs
    - `'names'`: set of string values of discrete dataset.
    - `'hrefs'`: set of string values of links that each name value refers to. (optional- can be left as a blank array)
- `'dataset'`: JSON object with 4 `'key: value'` pairs
    - `'title'`: title of the chart
    - `'values'`: set of numbers for each vertex.
    - `'bgColor'`: string value in HEX (i.e `'#ccc'`) for background
    - `'fgColor'`: string value in HEX for foreground polygon
- `'chartDiv'`: id of `<div>` in which the chart should be drawn
- `'chartType'`: string value of chart type
- `'chartSize'`: JSON object with the width and the height of a viewport.

Code Example

```
<div id="nWagon"></div>
<script>
    var options = {
        'legend':{
            names: ['Perceivable', 'Information Loss', 'Understandable', 'Enough Time', 'Epilepsy
Prevent', 'Operable', 'Navigation', 'Error Prevent'],
            hrefs: ['http://nuli.nhncorp.com/accessibility#k1',
                    'http://nuli.nhncorp.com/accessibility#k2',
                    'http://nuli.nhncorp.com/accessibility#k3',
                    'http://nuli.nhncorp.com/accessibility#k4',
                    'http://nuli.nhncorp.com/accessibility#k5',
                    'http://nuli.nhncorp.com/accessibility#k6',
                    'http://nuli.nhncorp.com/accessibility#k7',
                    'http://nuli.nhncorp.com/accessibility#k8']
        },
        'dataset': {
            title: 'Web accessibility status',
            values: [[34,53,67,23,78,45,69,98]],
            bgColor: '#f9f9f9',
            fgColor: '#30a1ce',
        },
        'chartDiv': 'nWagon',
        'chartType': 'radar',
        'chartSize': {width:500, height:300}
    };
    nWagon.chart(options);
</script>
```

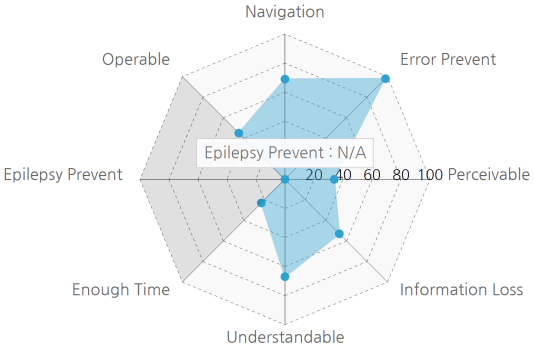
View Example





Easy data preparation

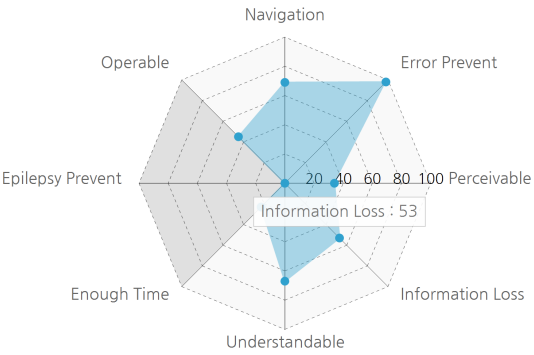
As shown above, data preparation is very simple and options such as number of radars, color of background or foreground can be dynamically applied to the chart.



Dimmed background for 'N/A' items

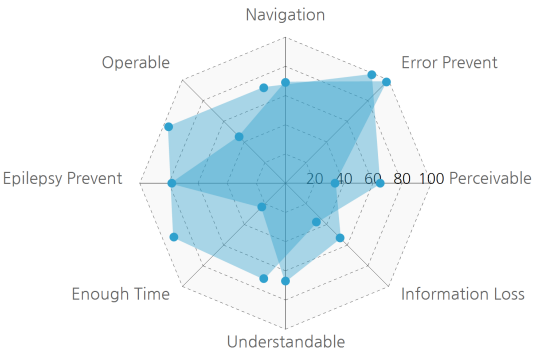
As shown in the image left, the chart is designed to display 'dimmed' background for non-applicable items without tweaking the data, which are typically handled as an error. Wagon's radar chart detects values that are not in a number format (i.e 'N/A', 'Non-Applicable') and apply alpha value to the area associated with it.

```
'dataset': {
    title: 'Web accessibility status',
    values: [[34,53,67,23,'N/A',45,69,98]],
    bgColor: '#f9f9f9',
    fgColor: '#30a1ce',
}
```



Tooltip for presenting the exact values

The library automatically creates mouse events on each vertex in order to display the exact value in a tooltip format that are taken from the dataset.

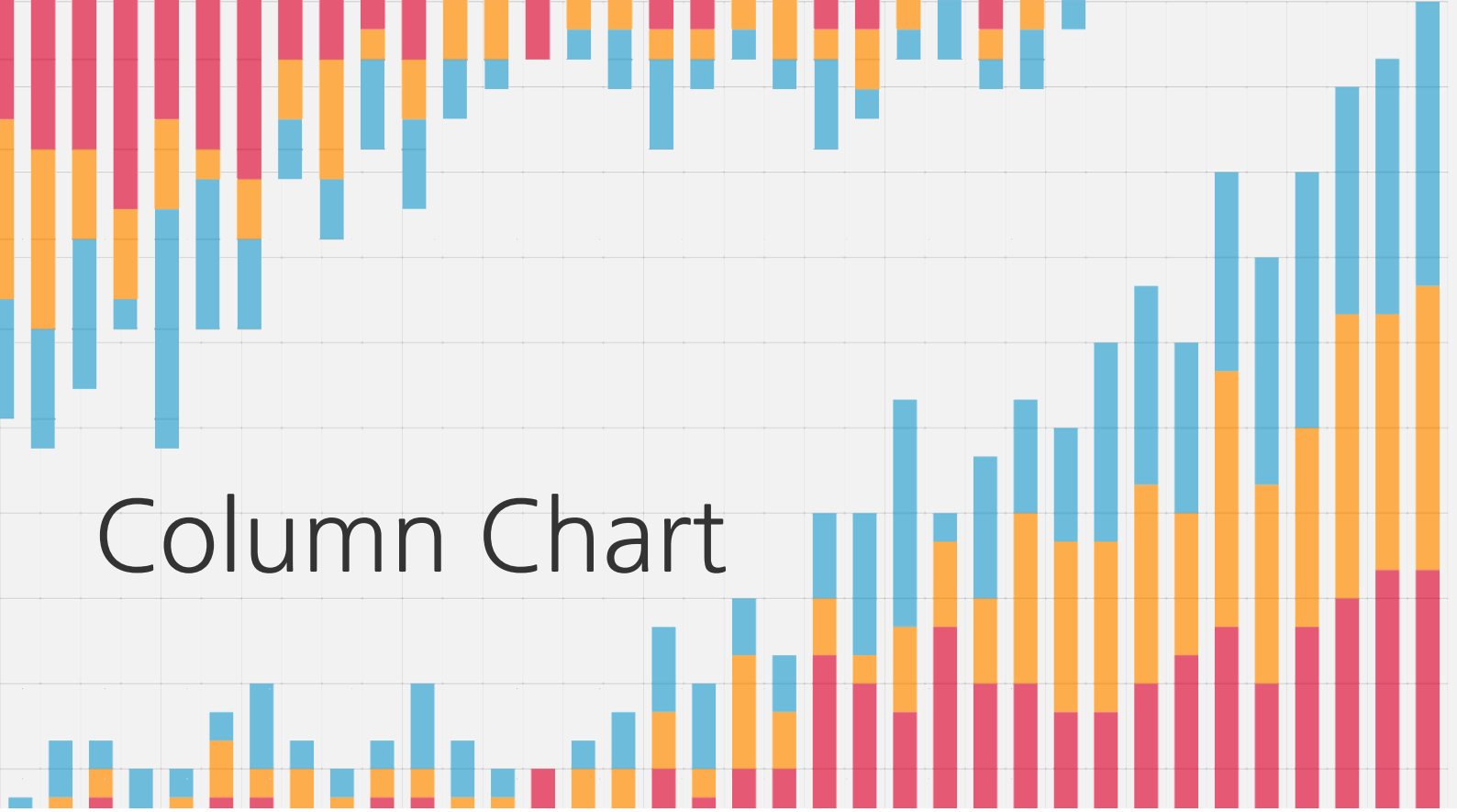


Multiple charts within one area

It is extremely easy and helpful when comparing two or more charts. By just adding datasets, the chart library creates and stacks multiple polygons on a same background as shown in the image left. In this case, tooltip creation will be omitted to eliminating any confusion.

```
'dataset': {
    title: 'Web accessibility status',
    values: [[34,53,67,23,78,45,69,98],
             [65,34,67,85,89,67,95]],
    bgColor: '#f9f9f9',
    fgColor: '#30a1ce',
}
```

Column Chart



Column Chart

Overview

Single column chart is the simplest version of column chart and the data presentation is very straightforward. As the name indicates, single column chart is used to compare distinctive data set within a single category.

Types

Single column:

Single column chart is the simplest version of column chart and the data presentation is very straightforward. As the name indicates, single column chart is used to compare distinctive data set within a single category.

Stacked column:

As one of variations of the column chart, stacked column chart is also used to compare total values or ratios within a same magnitude. But in a stacked column chart, each column is composed of two or more discrete portions.

Multi column:

A multi column chart, another variation of column chart serves a similar objective as a stacked column chart. One distinctive difference is that the data is displayed as a group of individual columns rather than components of a column.

Single column

JSON syntax

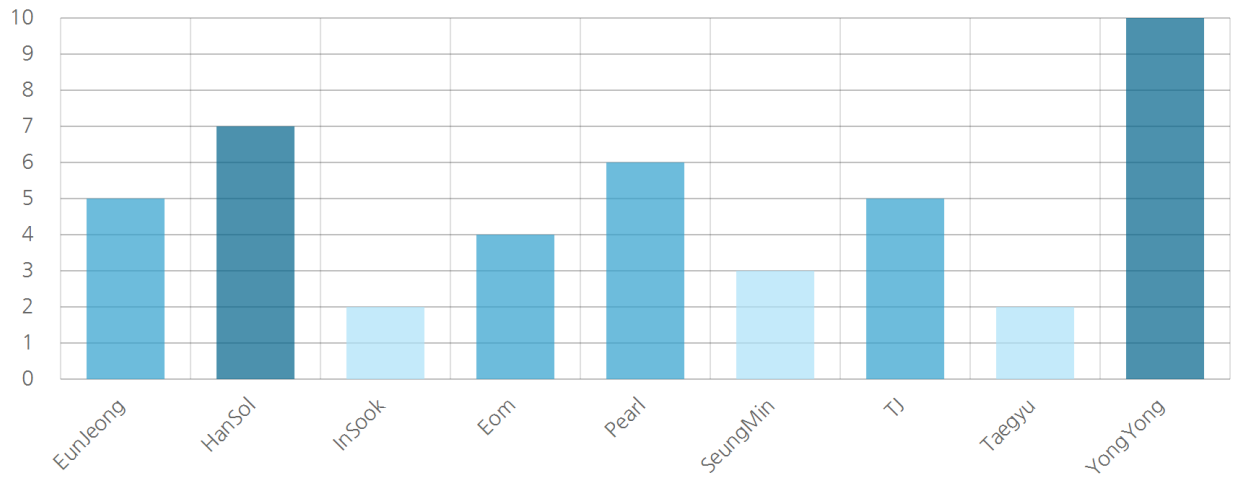
Similar to the radar chart, column chart also takes a JSON object as its only parameter and the keys and values are as following.

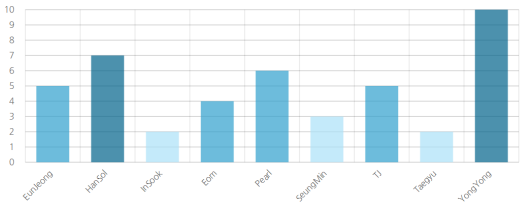
- `'legend'`: JSON object with 2 `'key: value'` pairs
  - `'names'`: set of string values of discrete dataset.
  - `'hrefs'`: set of string values of links that each name value refers to. (optional- can be left as a blank array)
- `'dataset'`: JSON object with 4 `'key: value'` pairs
  - `'title'`: title of the chart
  - `'values'`: set of numbers for each vertex.
  - `'colorset'`: array of string values in HEX with which categorical groups can be expressed. Refer to the features section for more detail.
- `'chartDiv'`: id of `<div>` in which the chart should be drawn
- `'chartType'`: string value of chart type
- `'chartSize'`: JSON object with the width and the height of a viewport.
- `'maxValue'`: number that represents the highest possible value in a dataset. If omitted, the maximum value from the dataset will be used.
- `'increment'`: number that is used to draw horizontal background lines and incremental values on the y-coordinate.

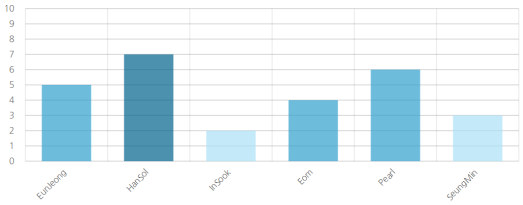
Code Example

```
<div id="nWagon"></div>

<script>
    var options = {
        'legend': {
            names: ['EunJeong', 'HanSol', 'InSook', 'Eom', 'Pearl', 'SeungMin', 'TJ', 'Taegyu',
'YongYong'],
            hrefs: ['http://nuli.nhncorp.com/blog/1132444',
                    'http://nuli.nhncorp.com/blog/1132442',
                    'http://nuli.nhncorp.com/blog/1132439',
                    'http://nuli.nhncorp.com/blog/1132426',
                    'http://nuli.nhncorp.com/blog/1115205',
                    'http://nuli.nhncorp.com/blog/1111811',
                    'http://nuli.nhncorp.com/blog/1111181',
                    'http://nuli.nhncorp.com/blog/1096163',
                    'http://nuli.nhncorp.com/blog/1079940']
            },
            'dataset': {
                title: 'Playing time per day',
                values: [5,7,2,4,6,3,5,2,10],
                colorset: ['#DC143C', '#FF8C00', "#30a1ce"]
            },
        'chartDiv': 'nWagon',
        'chartType': 'column',
        'chartSize': {width:700, height:300},
        'maxValue': 10,
        'increment': 1
    };
    nWagon.chart(options);
</script>
```

View Example

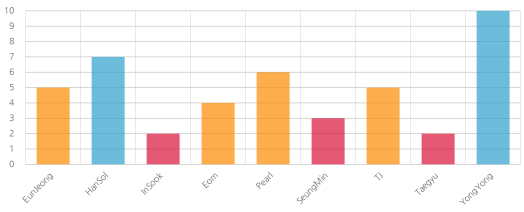






Easy data preparation

As explained above, data preparation is easy. Depends on the number of data values, columns will be drawn within the assigned viewport and the width of columns will be calculated with the width of the viewport and the number of values.



#DC143C 〈 #FF8C00 〈 #30A1CE

Easily customizable color theme

As briefly explained in the previous section, range of values can be expressed with a user-defined color theme. Depends on the number of colors, the categorical groups can be defined and appropriate colors will be used to fill the bars within the groups. For example, if the maximum value is set to 100 and 4 different colors are present in the 'colorset', the first value in the 'colorset' is used for the columns whose values are 0~25, the second value is for columns whose values are 26~50 and so on.

```
'dataset': {
    title: 'Playing time per day',
    values: [5,7,2,4,6,3],
    colorset: ['#DC143C', '#FF8C00', "#30A1CE"]
}
```

Stacked column

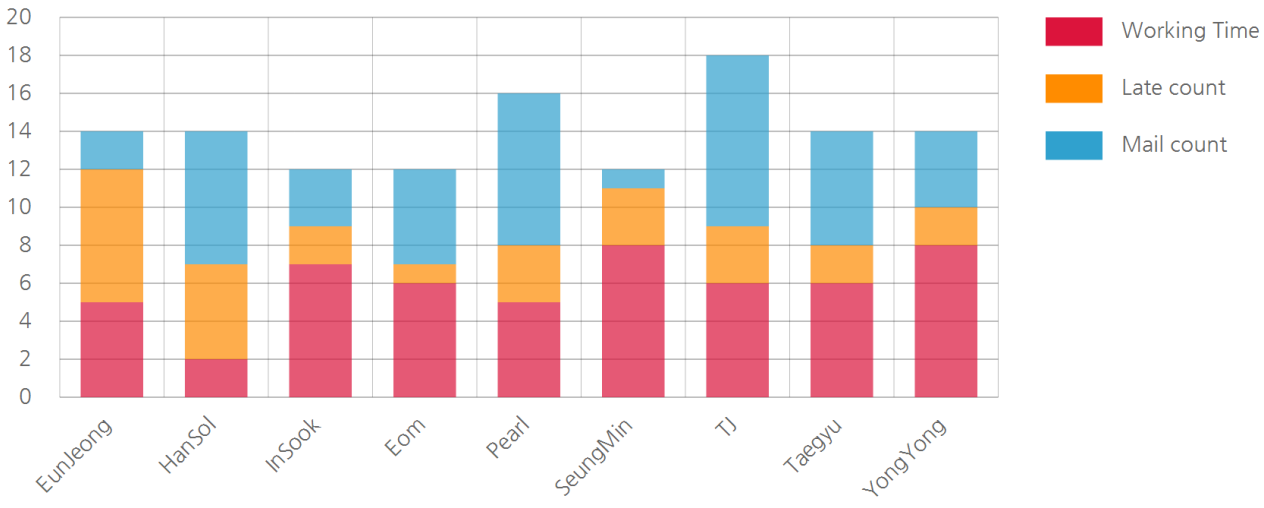
JSON syntax

As expected, syntax for creating a stacked column chart is very similar to the one for a regular column chart with minor differences. And the differences are as shown below.

- `'dataset'`: JSON object with 4 `'key: value'` pairs
  - `'title'`: title of the chart
  - `'values'`: arrays of number set for each column. Each array includes values for discrete potions.
  - `'colorset'`: array of string values in HEX that are used to define each discrete portion.
  - `'names'`: array of string where each value represents each discrete portion.
- `'chartType'`: string value of chart type

Code Example

```
// --- Code omitted ---

'dataset': {
    title: 'Playing time per day',
    values: [[5,7,2], [2,5,7], [7,2,3], [6,1,5], [5,3,8], [8,3,1], [6,3,9], [6,2,6], [8,2,4]],
    colorset: ['#DC143C', '#FF8C00', "#30a1ce"],
    fields: ['Working Time', 'Late count', 'Mail count']
},
'chartDiv': 'nWagon',
'chartType': 'stacked_column',

// --- Code omitted ---
```



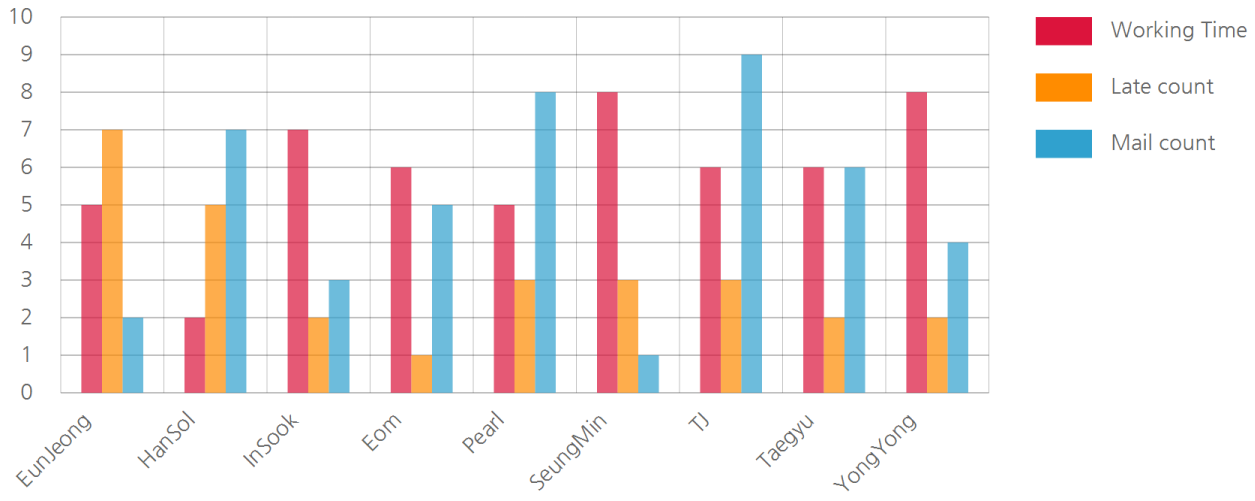
Multi column

JSON syntax

One difference in preparing data for a multi column chart is the value of chart type

Code Example

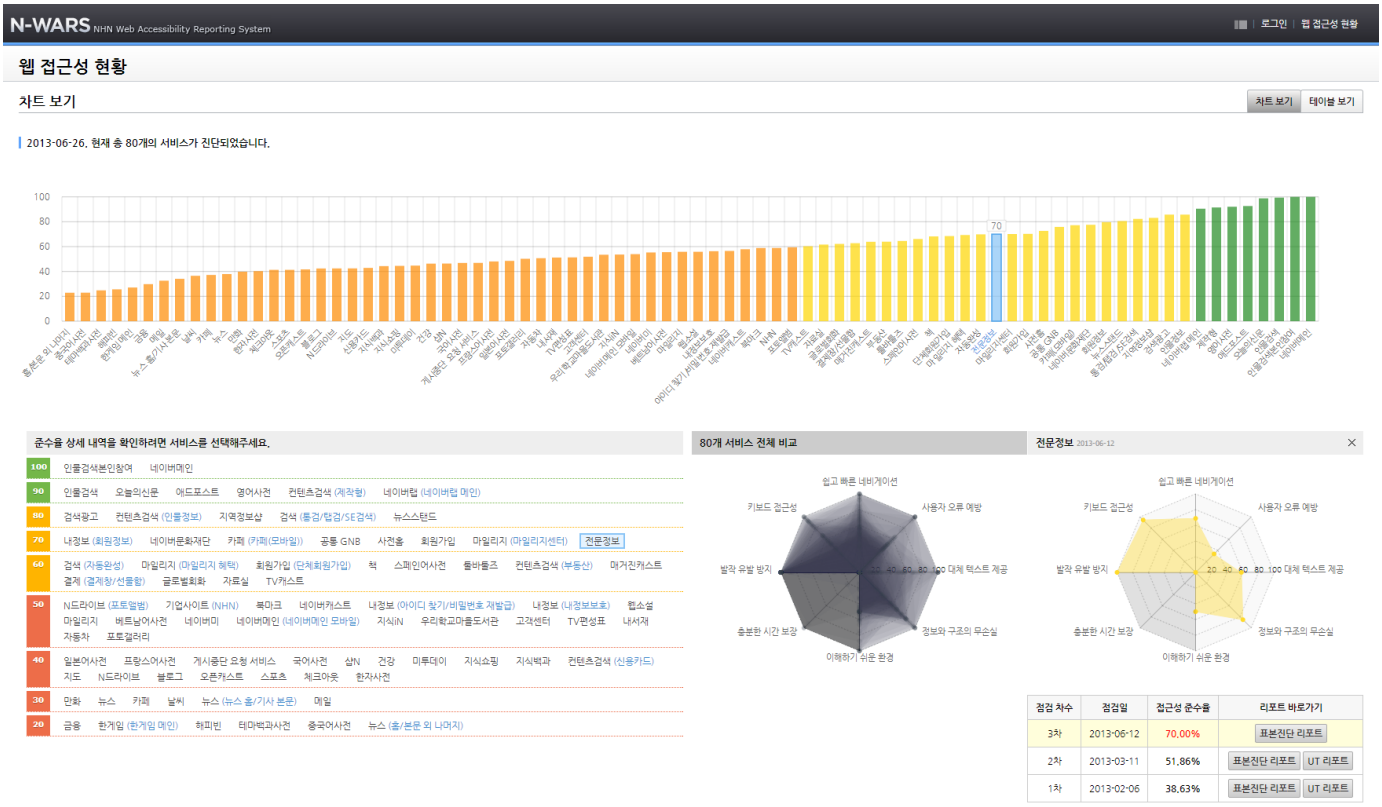
```
// --- Code omitted ---

'dataset': {
    title: 'Playing time per day',
    values: [[5,7,2], [2,5,7], [7,2,3], [6,1,5], [5,3,8], [8,3,1], [6,3,9], [6,2,6], [8,2,4]],
    colorset: ['#DC143C', '#FF8C00', "#30a1ce"],
    fields: ['Working Time', 'Late count', 'Mail count']
},
'chartDiv': 'nWagon',
'chartType': 'multi_column',

// --- Code omitted ---
```



Use case

N-WARS (NHN Web Accessibility Reporting System)

Dashboard



Report summary