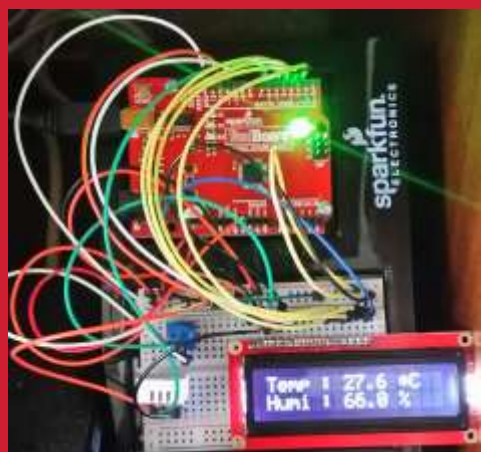




Arduino-IOT

[wk09]

Arduino + node.js Data visualization I



Visualization of Signals using Arduino,
Node.js & Storing Signals in MongoDB
& Mining Data using Python



Comsi, INJE University

2nd semester, 2019

Email : chaos21c@gmail.com



My ID

ID	성명
AA01	김관용
AA02	백동진
AA03	김도훈
AA04	김희찬
AA05	류재현
AA06	문민규
AA07	박진석
AA08	이승협
AA09	표혜성
AA10	김다영
AA11	성소진
AA12	김해인
AA13	신송주
AA14	윤지훈



[Review]

◆ [wk06]

- **Arduino sensors + Node.js**
- **Complete your project**
- **Upload folder: AAnn_Rpt06**

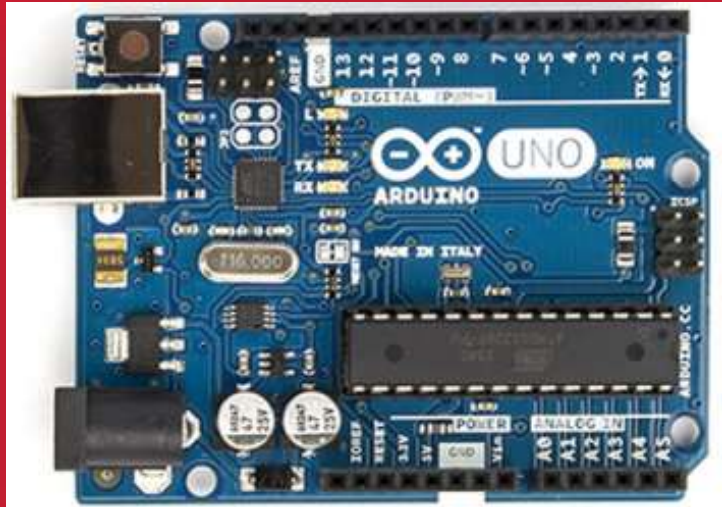
◆ [Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

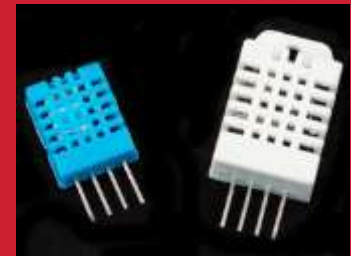
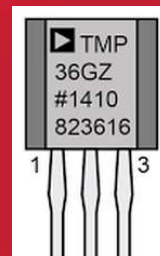
제출폴더명 : **AAnn_Rpt06**

- 압축할 파일들

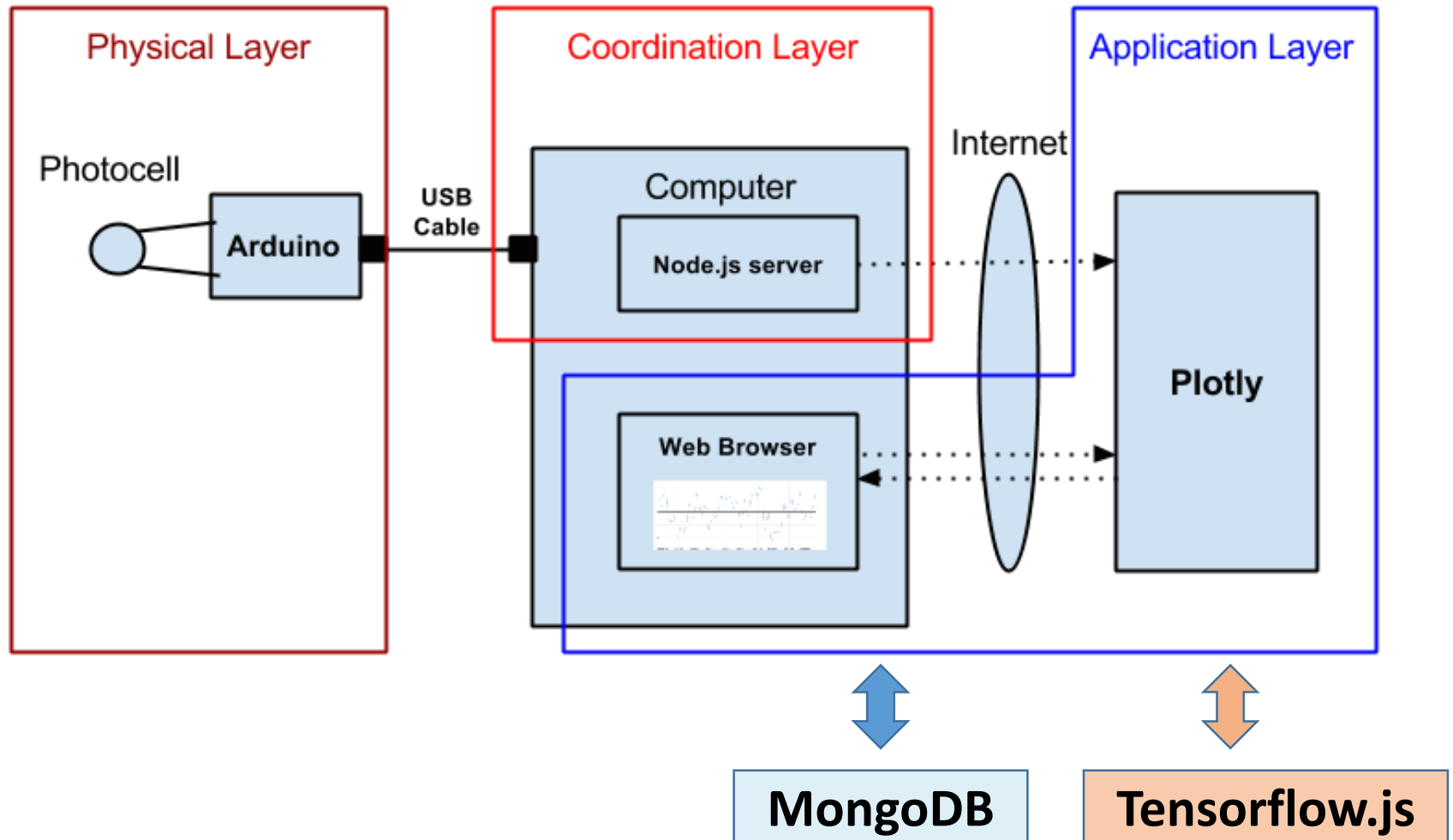
- ① **AAnn_cds_IOT_data.png**
- ② **AAnn_cds_tmp36_lcd.png**
- ③ **AAnn_cds_tmp36_IOT.png**
- ④ **AAnn_multi_signals_node.png**
- ⑤ **All *.ino**
- ⑥ **All *.js**

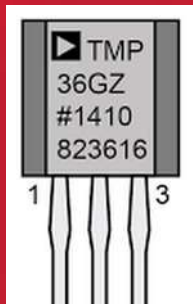
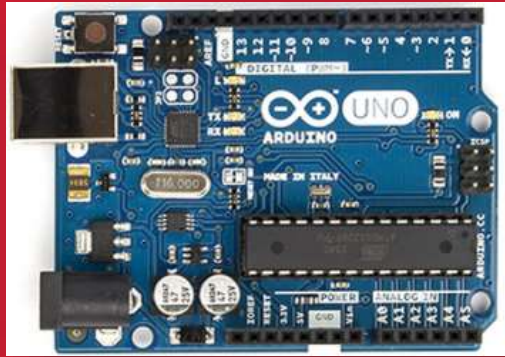


Arduino & Node.js



Layout [H S C]

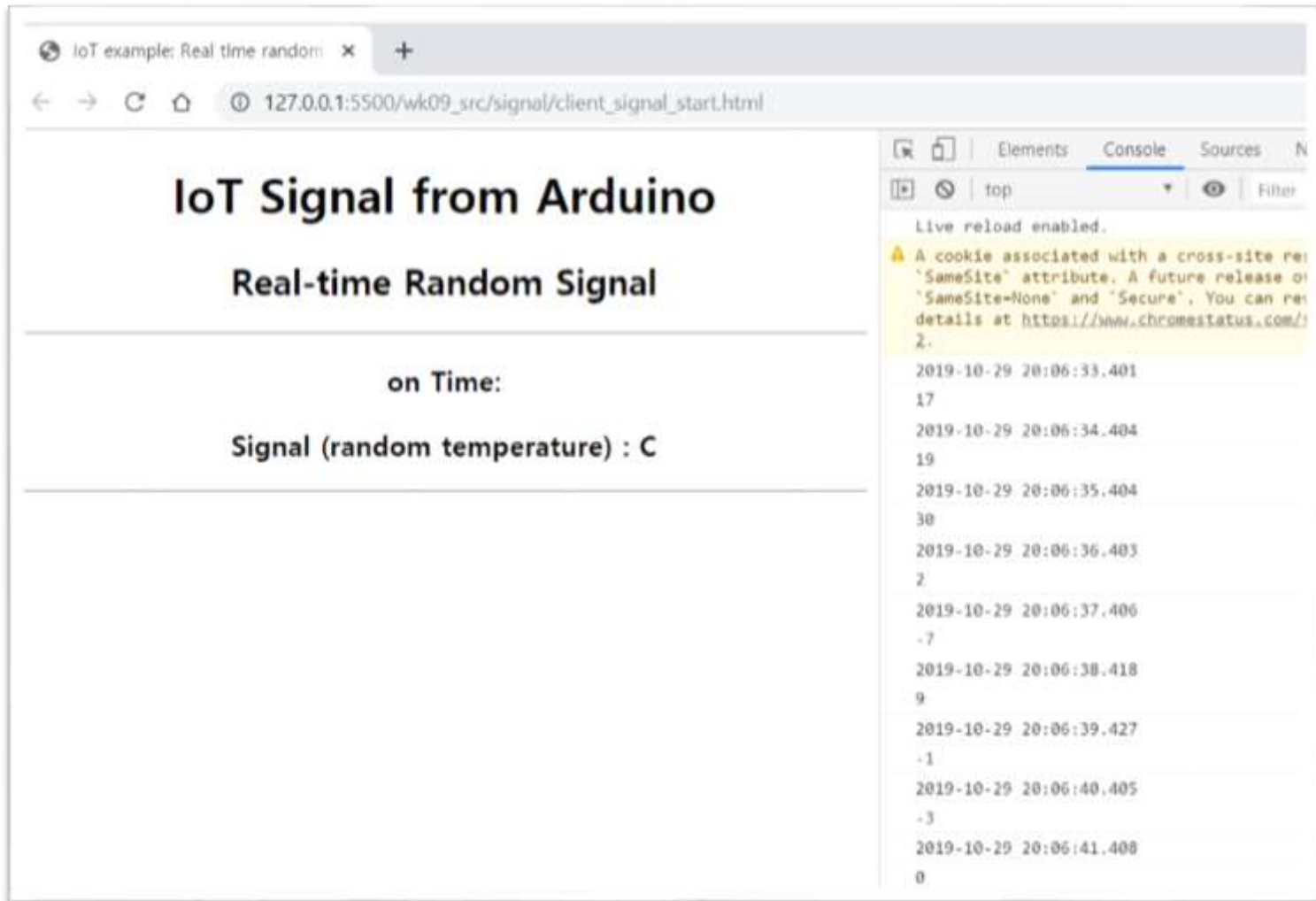




Data visualization using **play.ly**



Arduino data on network socket



The screenshot shows a web browser window with the title 'IoT example: Real time random...' and the address bar displaying '127.0.0.1:5500/wk09_src/signal/client_signal_start.html'. The page content includes the heading 'IoT Signal from Arduino' and 'Real-time Random Signal'. Below this, it says 'on Time:' and 'Signal (random temperature) : C'. The right-hand side of the browser shows the 'Console' tab with the following messages:

- Live reload enabled.
- A cookie associated with a cross-site request is missing the 'SameSite' attribute. A future release of Chrome will require the 'SameSite' attribute to be present on cookies. You can read more about this at <https://www.chromestatus.com/feature/5718547748771840>.
- 2019-10-29 20:06:33.401 17
- 2019-10-29 20:06:34.404 19
- 2019-10-29 20:06:35.404 30
- 2019-10-29 20:06:36.403 2
- 2019-10-29 20:06:37.406 -7
- 2019-10-29 20:06:38.418 9
- 2019-10-29 20:06:39.427 -1
- 2019-10-29 20:06:40.405 -3
- 2019-10-29 20:06:41.408 0

**Real-time console showing a signal from Arduino
in Chrome browser**

Arduino data on network socket



The screenshot shows a web browser window with the title "IoT example: Real time random". The address bar displays the URL "127.0.0.1:5500/wk09_src/signal/...". The page content includes the heading "IoT Signal from Arduino" and "Real-time Random Signal". Below this, it shows the timestamp "on Time: 2019-10-29 19:53:10.127" and the signal value "Signal (random temperature) : 1 C". A yellow banner at the bottom of the browser window contains the text "Real-time monitoring of a signal from Arduino".

IoT Signal from Arduino

Real-time Random Signal

on Time: 2019-10-29 19:53:10.127

Signal (random temperature) : 1 C

Real-time monitoring of a signal from Arduino

Arduino data + plotly

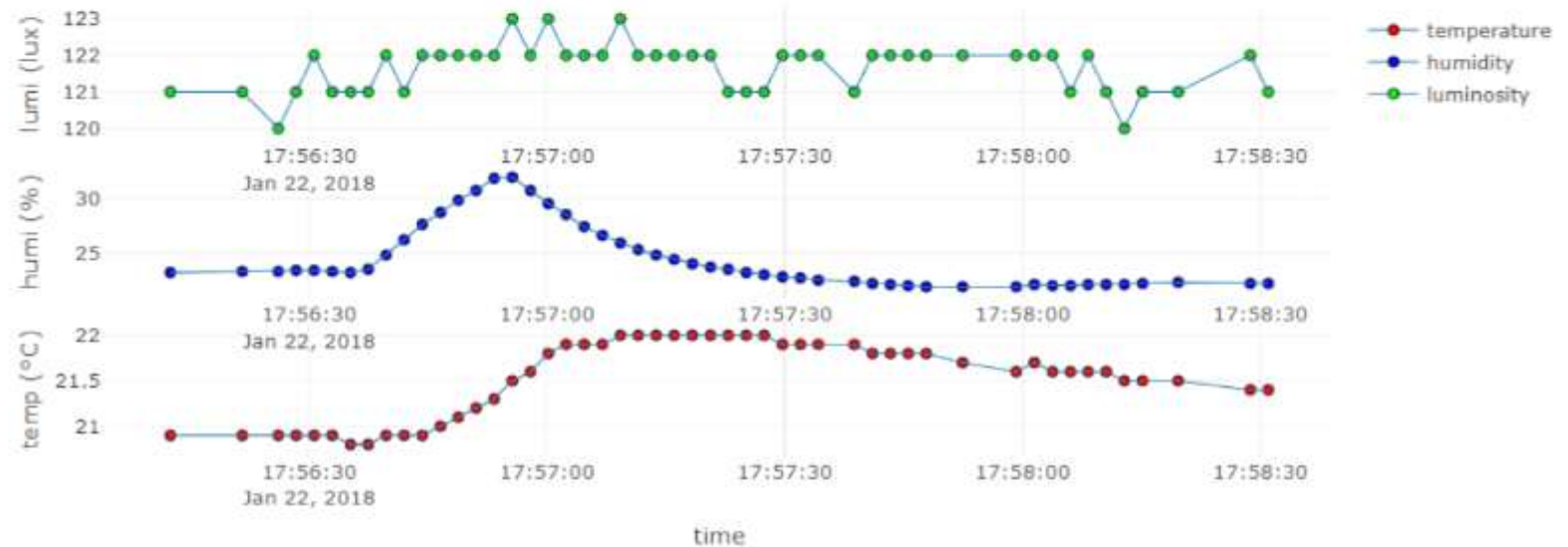
Time series by AA00



Real-time Weather Station from sensors



on Time: 2018-01-22 17:58:31.012





A5. Introduction to visualization

System (Arduino, sDevice, ...)



Data (signal, image, sns, ...)



Visualization & monitoring



Data storing & mining



Service





A5.1 Introduction to data visualization

아두이노 센서 회로

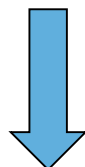


직렬모니터/플로터 모니터링



LCD 모니터링

Node.js



Plotly.js




웹 모니터링




A5.1.1 D3.js

[←](#) [→](#) [C](#) [H](#) [d3js.org](#) [5](#) [☆](#) [↓](#) [☰](#)

[Overview](#) [Examples](#) [Documentation](#) [Source](#)



Data-Driven Documents



D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

[See more examples.](#)

<http://d3js.org/>



A5.1.2 plot.ly

The screenshot shows the plot.ly website homepage. At the top, there is a navigation bar with links for PRODUCTS, INDUSTRIES, RESOURCES, SIGN UP, and a DEMO DASH button. Below the navigation bar, a blue banner promotes a weekly live group demo for Dash Enterprise, with a REGISTER HERE link. The main content area features the headline "Bring data science out of the lab and into the business" and a sub-headline stating "Dash is the trusted solution for operationalizing Python and R models." Two buttons, "DASH FOR ENTERPRISE" and "DASH OPEN SOURCE", are positioned below the text. To the right, an isometric illustration depicts a modern office environment with people working at desks, surrounded by stylized trees and clouds. At the bottom, a row of logos for partner companies (Shell, Amazon, Cisco, S&P Global, Invesco) is displayed. A chatbot interface is visible in the bottom right corner, with a speech bubble saying "Hi! How can I help?" and a button with a red notification badge.

plot.ly

PRODUCTS ▾ INDUSTRIES ▾ RESOURCES ▾ SIGN UP ▾ DEMO DASH

Interested in Dash Enterprise? Join us for our weekly live group demo. [REGISTER HERE!](#)

Bring data science out of the lab and into the business

Dash is the trusted solution for operationalizing Python and R models.

DASH FOR ENTERPRISE DASH OPEN SOURCE

Shell amazon CISCO S&P Global Invesco

Hi! How can I help?



A5.1.3 plotly.js




plotly.js is Plotly's **client-side**,
interactive JavaScript graphing
library, built on top of **D3.js**,
stack.gl.

<https://plot.ly/javascript/>




A5.1.4 Introduction to plotly.js

 **plotly** | Graphing Libraries

[Help](#) | [Open Source Graphing Libraries](#) | [JavaScript](#) [Edit this page on GitHub](#)

Quick Start
Getting Started
Cheat Sheet
CDN
Download
Full Reference
Event Reference
Function Reference
Configuration Options

Examples
Plotly Fundamentals
Basic
Statistical
Scientific
Financial
Maps
3D
Subplots
Chart Events
Transforms
Custom Controls



Plotly JavaScript Open Source Graphing Library


Built on top of [d3.js](#) and [stack.gl](#), plotly.js is a high-level, declarative charting library. plotly.js ships with 20 chart types, including 3D charts, statistical graphs, and SVG maps.

[Learn about why we open sourced plotly.js](#) or [view the source on GitHub](#).


► Read more about plotly.js features

Search

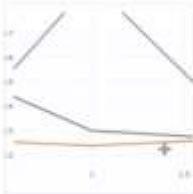
Plotly Fundamentals




Static Image Export




Responsive / Fluid Layout



UI revision in Plotly.react



React Plotly.js



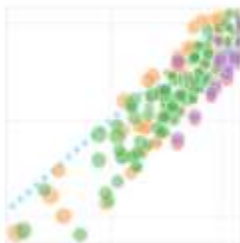
More Fundamentals

<https://plot.ly/javascript/>

17

A5.1.5 Introduction to plotly.js charts

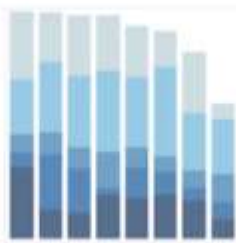
Basic Charts [🔗](#)



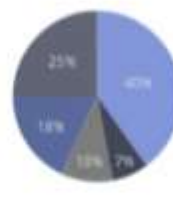
Scatter Plots



Line Charts



Bar Charts

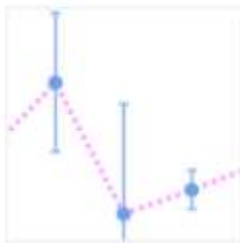


Pie Charts

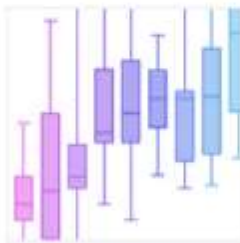


More Basic Charts

Statistical Charts [🔗](#)



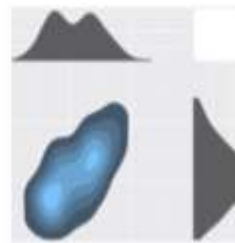
Error Bars



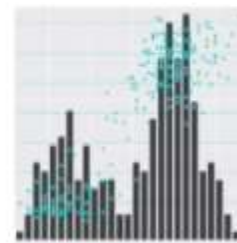
Box Plots



Histograms



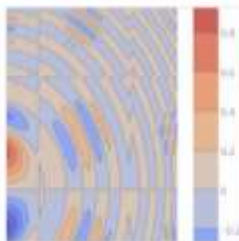
2d Density Plots



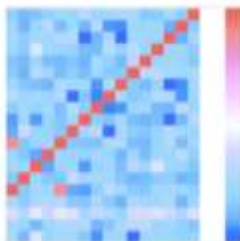
More Statistical

A5.1.6 Introduction to plotly.js charts

Scientific Charts [🔗](#)



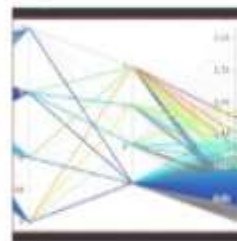
Contour
Plots



Heatmaps



Ternary Plots

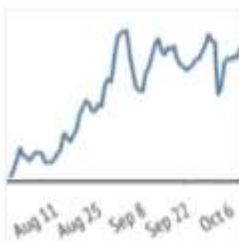


Parallel
Coordinates
Plot

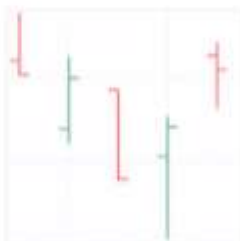


More
Scientific
Charts

Financial Charts [🔗](#)



Time Series



OHLC Charts



Candlestick
Charts

A5.1.7 Introduction to plotly.js charts

Maps [🔗](#)



Choropleth
Maps



Scatter Plots
on Maps



Bubble Maps



Lines on
Maps



Scatter Plots
on Mapbox

3D Charts [🔗](#)



3D Scatter
Plots



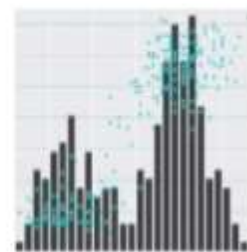
Ribbon Plots



3D Surface
Plots



3D Mesh
Plots



More 3D
Charts



A5.1.8 plotly.js: time series & streaming



<https://plot.ly/javascript/time-series/>



<https://plot.ly/javascript/streaming/>



A5.1.9 Getting started: plotly.js



Getting Started with plotly.js

Getting Started with plotly for JavaScript.



Scala



ggplot2



R



plotly.js



Python



Pandas



node.js



matplotlib



MATLAB

<https://plot.ly/javascript/getting-started/>

<https://plot.ly/>



A5.1.10 Getting started: plotly.js

plotly.js CDN [🔗](#)

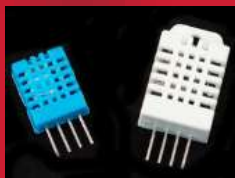
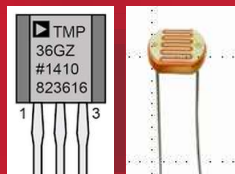
You can also use the ultrafast plotly.js CDN link. This CDN is graciously provided by the incredible team at [Fastly](#).

```
<head>  
  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>  
</head>
```

Else, if you want to get a specific version of plotly.js, say 1.2.0:

```
<head>  
  <script src="https://cdn.plot.ly/plotly-1.2.0.min.js"></script>  
</head>
```

```
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
```

Navigation

Basic Line Plot

Line and Scatter Plot

Adding Names to Line and Scatter Plot

Line and Scatter Styling

Styling Line Plot

Colored and Styled Scatter Plot

Line Shape Options for Interpolation

Graph and Axes Titles

Line Dash

Connect Gaps Between Data

Labelling Lines with Annotations

← Back To Plotly.js



Line Charts in plotly.js

How to make D3.js-based line charts in JavaScript.

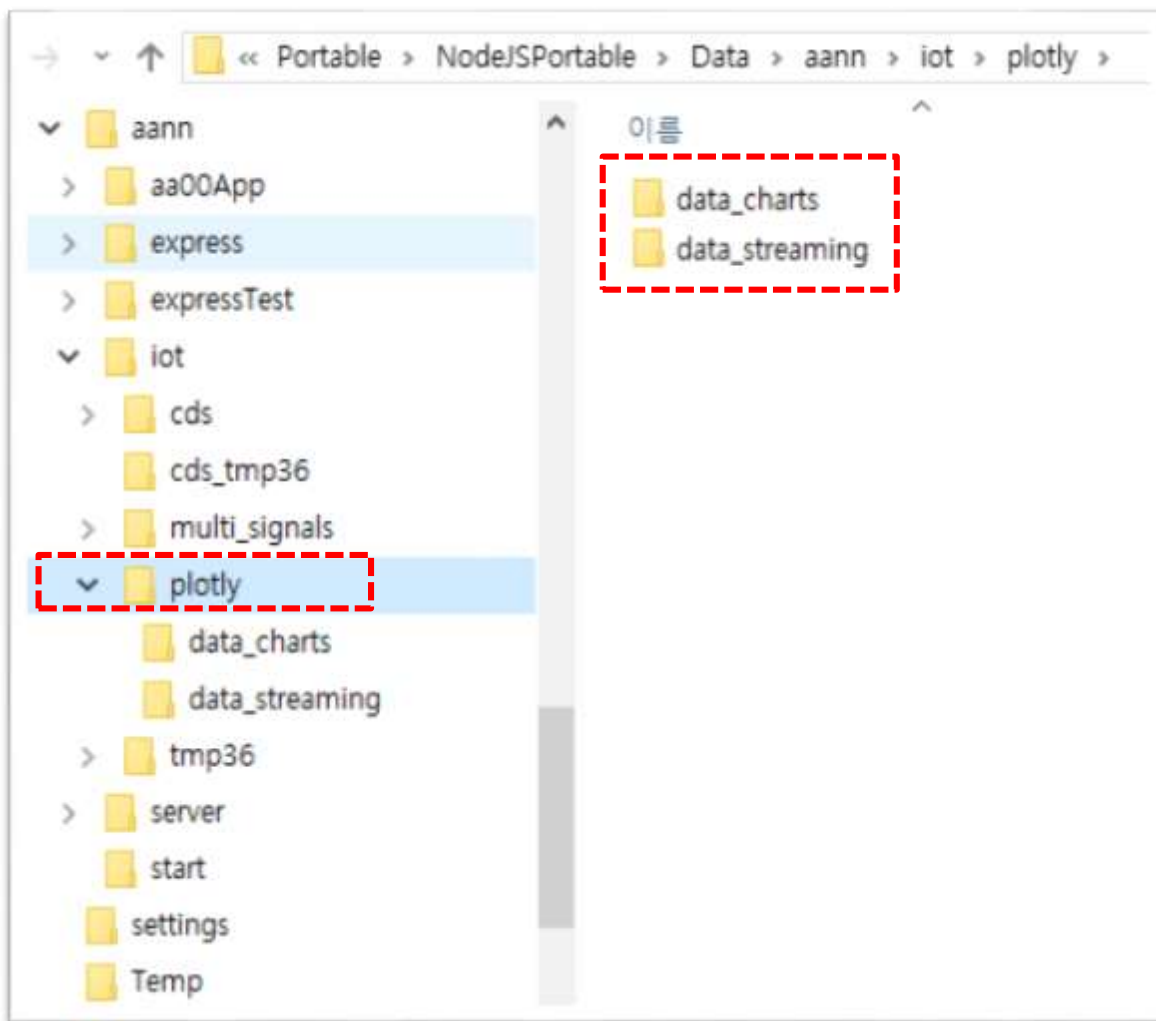


Basic Line Plot [↗](#)

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  type: 'scatter'
};

var trace2 = {
  x: [1, 2, 3, 4],
  y: [16, 5, 11, 9],
  type: 'scatter'
};
```

A5.2.1 Working folders





A5.2.2.1 Starting plotly basic chart

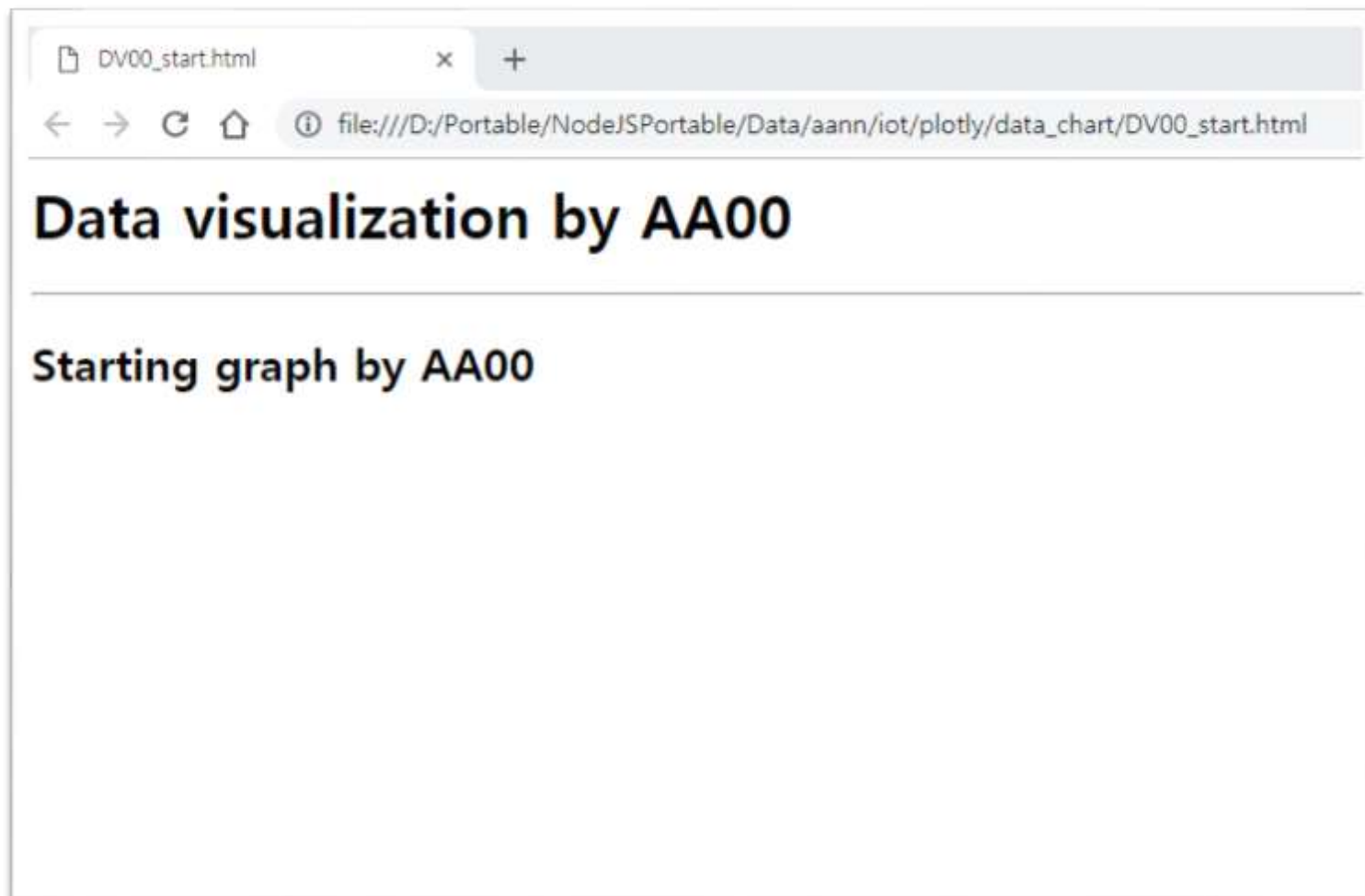
Starting chart!

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <!-- Plotly.js -->
5   <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
6 </head>
7 <body>
8   <h1>Data visualization by AA00</h1>
9   <hr>
10  <h2>Starting graph by AA00</h2>
11
12  <!-- Plotly chart will be drawn inside this DIV -->
13  <div id="myDiv" style="width: 500px;height: 300px"></div>
14
15  <script>
16    <!-- JAVASCRIPT CODE GOES HERE -->
17
18
19  </script>
20 </body>
21 </html>
22
```



A5.2.2.2 Starting plotly basic chart

SB3, ^B





[Tip] Using WEB browser in SB text3

[Tool] Sublime Text - 현재 작업 중인 파일을 웹브라우저로 열기

1. **Tool > Developer > New Plugin**을 실행 한 후 아래 내용으로 덮어 씌운 후 '**open_browser**'으로 저장한다.

```
import sublime, sublime_plugin
import webbrowser

class OpenBrowserCommand(sublime_plugin.TextCommand):
    def run(self,edit):
        url = self.view.file_name()
        webbrowser.open_new(url)
```

2. **Preferences -> Key Bindings - User**로 이동한 후 단축키를 할당한다.

```
{ "keys": ["f10"], "command": "open_browser" }
```



A5.2.3.1 Hello plotly basic chart

Hello plotly data chart!

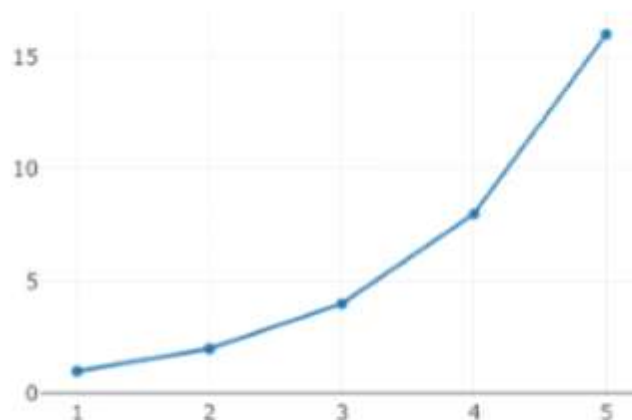
```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <!-- Plotly.js -->
5   <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
6 </head>
7 <body>
8   <h1>Data visualization by AA00</h1>
9   <hr>
10  <h2>Hello plotly!</h2>
11  <!-- Plotly chart will be drawn inside this DIV -->
12  <div id="myDiv" style="width: 500px; height: 400px"></div>
13  <hr>
14  <script>
15    <!-- JAVASCRIPT CODE GOES HERE -->
16    var data = [
17      {
18        x: [1, 2, 3, 4, 5],
19        y: [1, 2, 4, 8, 16],
20        type: 'scatter'
21      }
22    ];
23    Plotly.newPlot('myDiv', data);
24  </script>
25 </body>
26 </html>
```

data는 무엇?
그래프 객체들의 배열

Graph : Hello plotly chart!

Data visualization by AA00

Hello plotly!





A5.2.4 plotly.js: Line Charts

[1] Basic multi-line charts

```
<script>
  <!-- JAVASCRIPT CODE GOES HERE -->

  var trace1 = {
    x: [1, 2, 3, 4],
    y: [10, 15, 13, 17],
    type: 'scatter'
  };

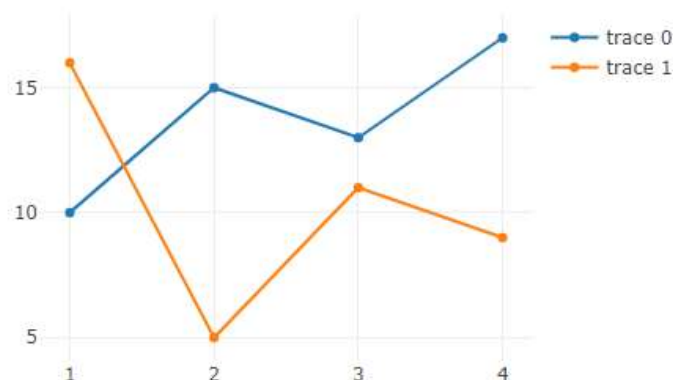
  var trace2 = {
    x: [1, 2, 3, 4],
    y: [16, 5, 11, 9],
    type: 'scatter'
  };

  var data = [trace1, trace2];

  Plotly.newPlot('myDiv', data);

</script>
```

Line charts by AA00



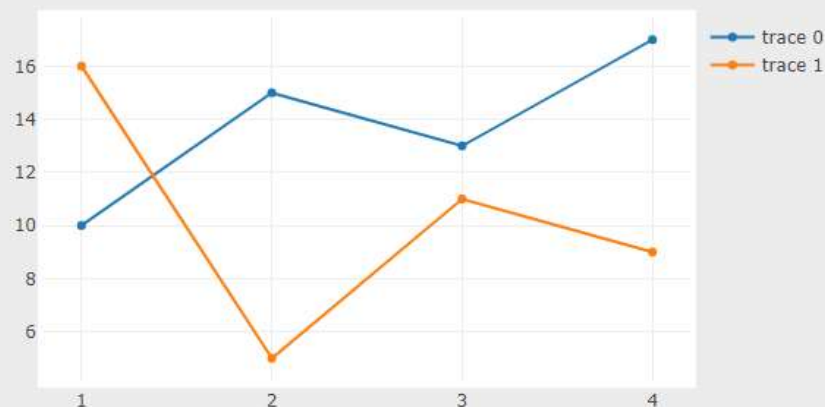
[2] Basic line charts with `layout`

```
var layout = {
  autosize: false,
  width: 600,
  height: 450,
  margin: {
    l: 50, // left
    r: 50, // right
    b: 100, // bottom
    t: 100, // top
    pad: 4 // padding
  },
  paper_bgcolor: '#ecec',
  plot_bgcolor: '#ffffff' // '#rrggbb'
};

Plotly.newPlot('myDiv', data, layout);
```

Test: pad → 40

Line charts with layout by AA00



AAnn_Chart_Layout.png

[3] Line & scatter plot

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'markers'
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines'
};

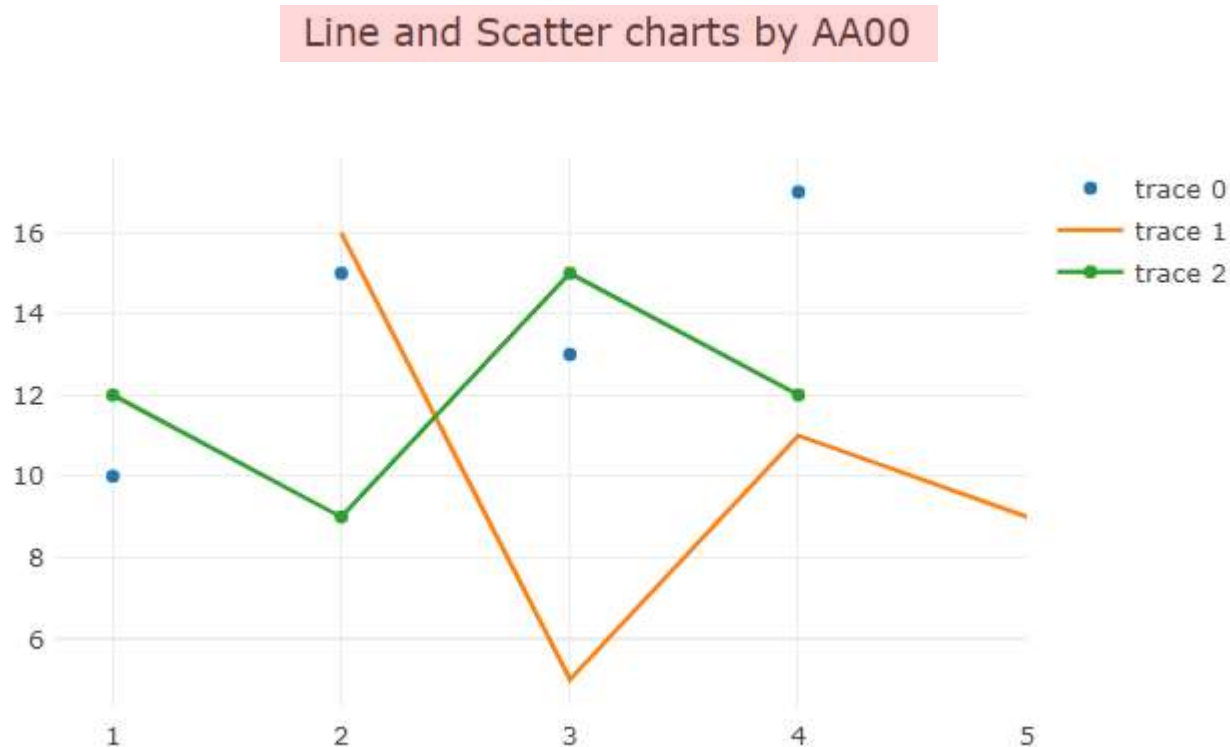
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines+markers'
};
```

```
var data = [ trace1, trace2, trace3 ];

var layout = {
  title: 'Line and Scatter charts by AA00',
  width: 600,
  height: 450,
  margin: {
    l: 50,
    r: 50,
    b: 100,
    t: 100,
    pad: 4
  },
};

Plotly.newPlot('myDiv', data, layout);
```

[3.1] Line & scatter plot **with title**

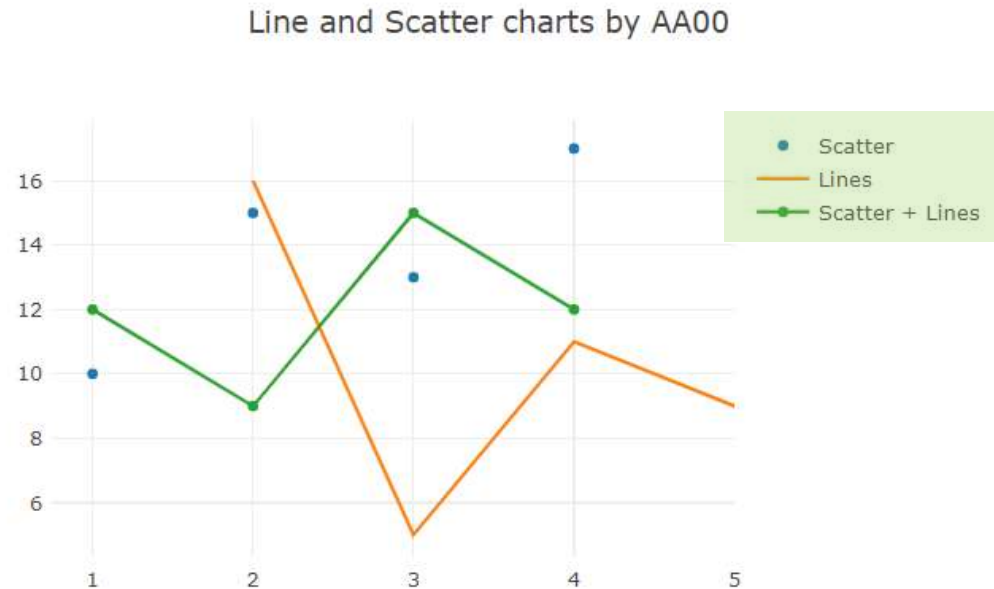


[3.2] Line & scatter plot with axis name

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'markers',
  name: 'Scatter'
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'Lines'
};

var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines+markers',
  name: 'Scatter + Lines'
};
```



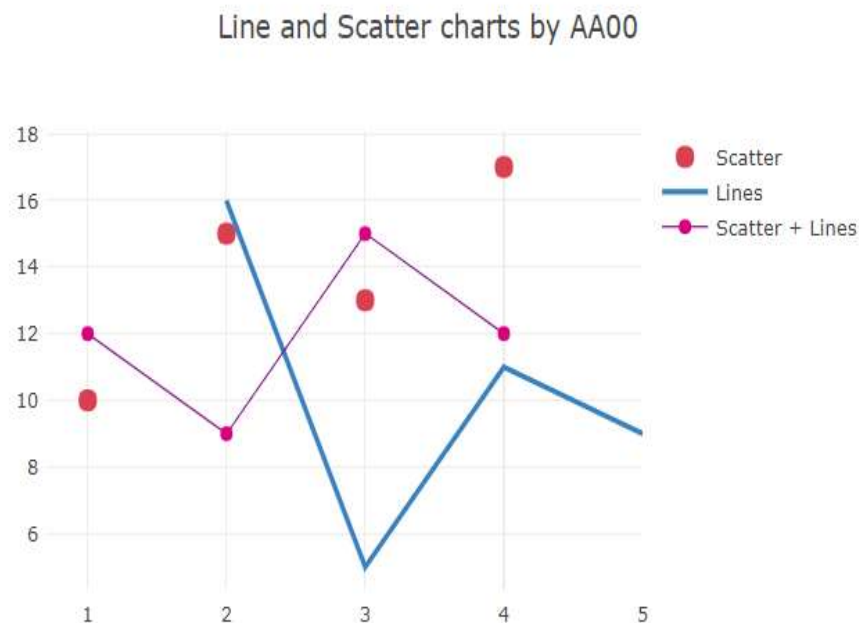


A5.2.6.4 plotly.js: Line & Scatter plot

[3.3] Line & scatter plot with style

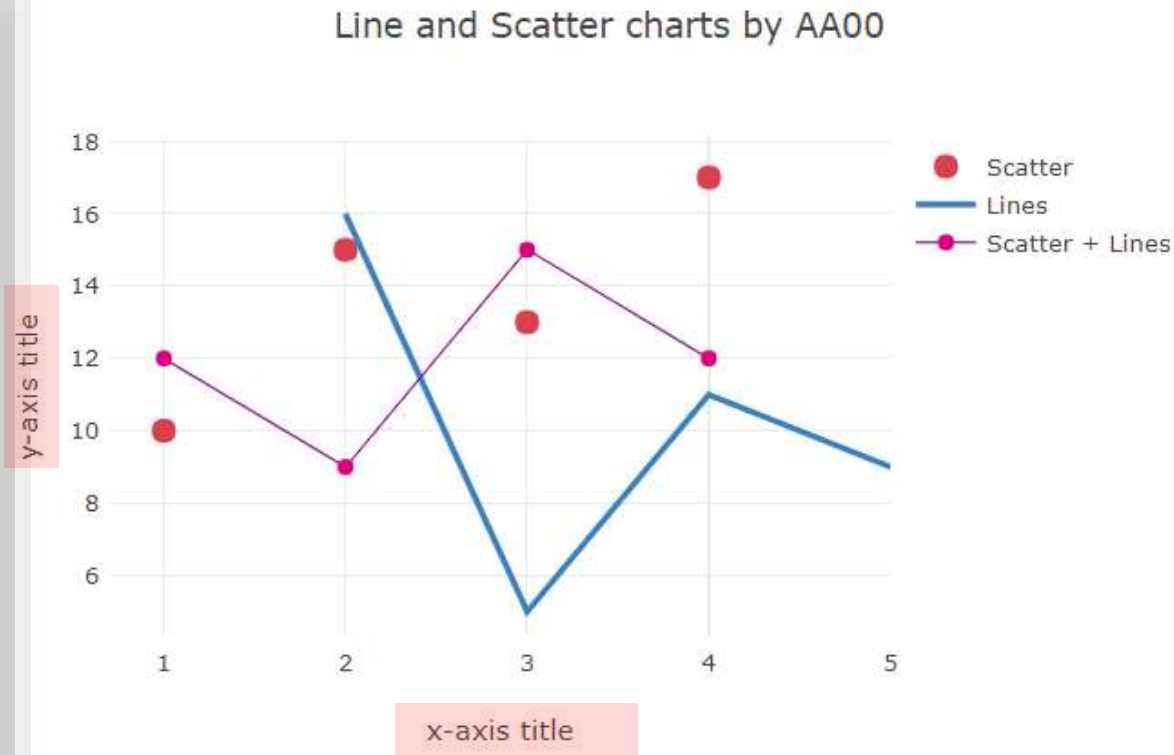
```
var trace1 = {  
  x: [1, 2, 3, 4],  
  y: [10, 15, 13, 17],  
  mode: 'markers',  
  name: 'Scatter',  
  marker: {  
    color: 'rgb(219, 64, 82)',  
    size: 12  
  }  
};  
  
var trace2 = {  
  x: [2, 3, 4, 5],  
  y: [16, 5, 11, 9],  
  mode: 'lines',  
  name: 'Lines',  
  line: {  
    color: 'rgb(55, 128, 191)',  
    width: 3  
  }  
};
```

```
var trace3 = {  
  x: [1, 2, 3, 4],  
  y: [12, 9, 15, 12],  
  mode: 'lines+markers',  
  name: 'Scatter + Lines',  
  marker: {  
    color: 'rgb(128, 0, 128)',  
    size: 8  
  },  
  line: {  
    color: 'rgb(128, 0, 128)',  
    width: 1  
  }  
};
```



[3.4] Line & scatter plot with axis titles

```
var layout = {
  title: 'Line and Scatter Plot',
  width: 600, height: 450,
  margin: {
    l: 50,
    r: 50,
    b: 100,
    t: 100,
    pad: 4
  },
  xaxis: {
    title: 'x-axis title'
  },
  yaxis: {
    title: 'y-axis title'
  }
};
```



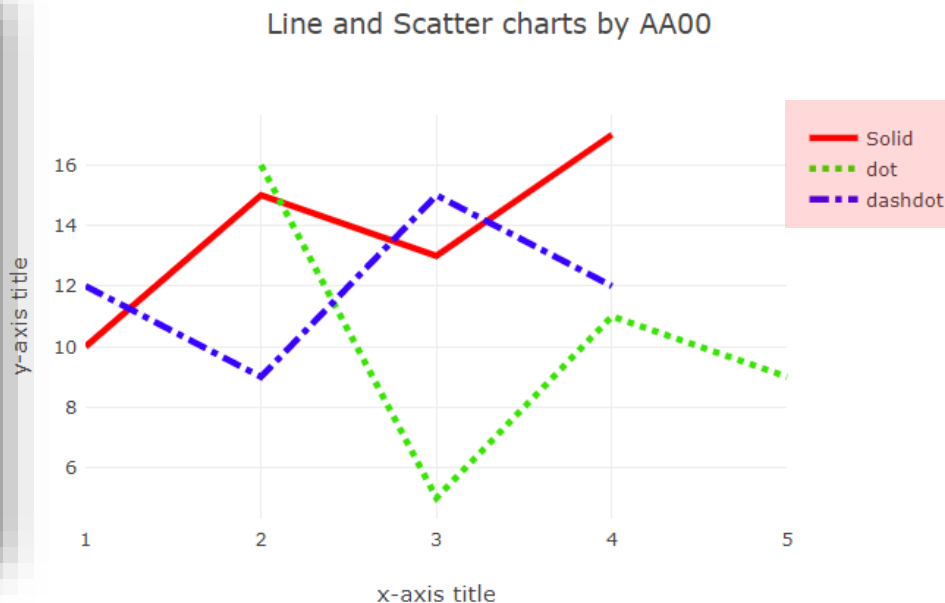
AAnn_Axis_Title.png

[3.5] Line & scatter plot with dash and dot

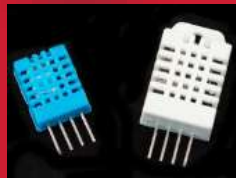
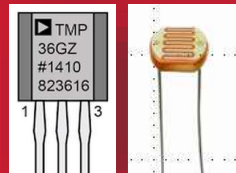
```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'lines',
  name: 'Solid',
  line: {
    color: 'rgb(255, 0, 0)',
    dash: 'solid',
    width: 4
  }
};
```

```
var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'dot',
  line: {
    color: 'rgb(55, 228, 0)',
    dash: 'dot',
    width: 4
  }
};
```

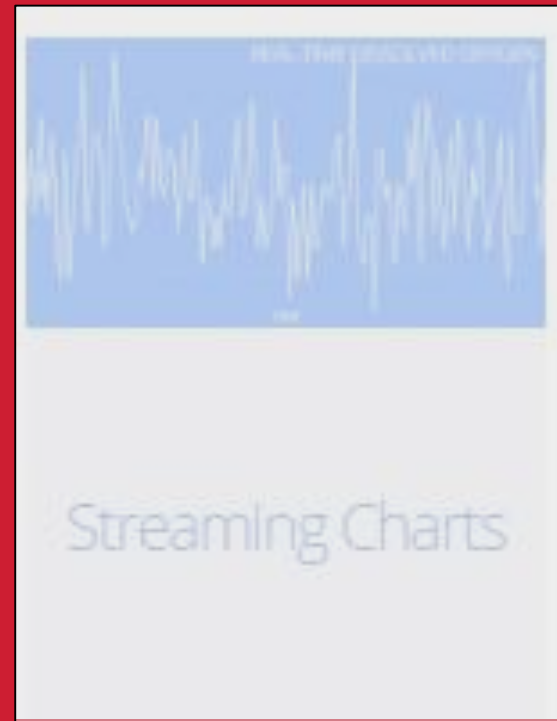
```
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines',
  name: 'dashdot',
  line: {
    color: 'rgb(55, 0, 255)',
    dash: 'dashdot',
    width: 4
  }
};
```



AAnn_Line_Dash_Dot.png



Data visualization using **plotly.js**



Navigation

Date Strings

[Basic Time Series](#)

Manually Set Range

Time Series with Rangeslider

[← Back To Plotly.js](#)

Time Series in plotly.js



How to plot D3.js-based date and time in Plotly.js. An example of a time-series plot.



R



Python



matplotlib



plotly.js



Pandas



node.js



MATLAB

Date Strings [↗](#)

```
var data = [
  {
    x: ['2013-10-04 22:23:00', '2013-11-04 22:23:00', '2013-12-04 22:23:00'],
    y: [1, 3, 6],
    type: 'scatter'
  }
];
```

```
Plotly.newPlot('myDiv', data);
```



A5.3.1 plotly.js: Time series

[1] Time series : date strings

```
<!-- Plotly chart will be drawn inside this DIV -->
<div id="myDiv" style="width: 500px;height: 400px"></div>

<script>
  <!-- JAVASCRIPT CODE GOES HERE -->

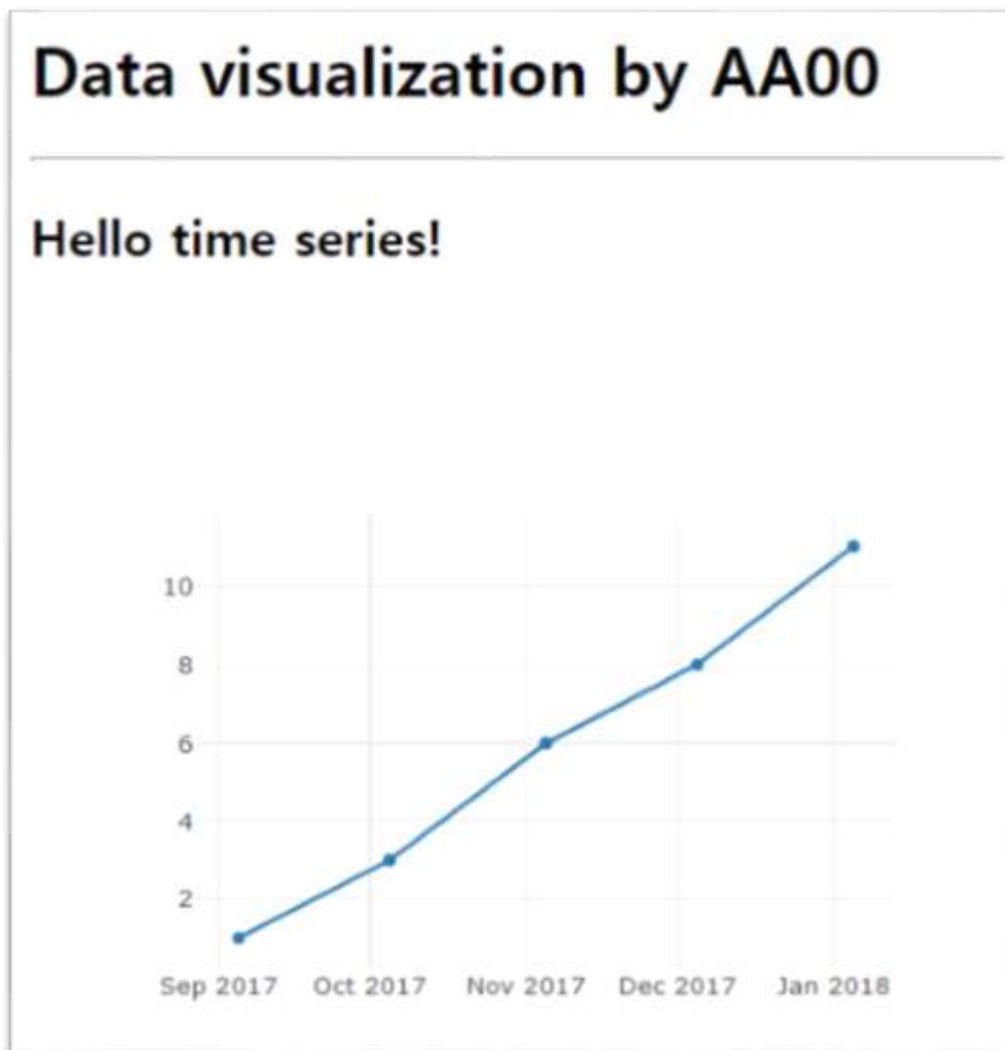
  var data = [
    {
      x: ['2017-9-04 22:23:00',
        '2017-10-04 22:23:00',
        '2017-11-04 22:23:00',
        '2017-12-04 22:23:00'],
      y: [1, 3, 6, 8],
      type: 'scatter'
    }
  ];

  Plotly.newPlot('myDiv', data);
</script>
```



A5.3.2 plotly.js: Time series

Time series : date strings – result



오늘 날짜와
데이터를 추가



A5.3.3.1 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

← → ↺ ⌂ 🔒 안전함 | <https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-a...> 🔍 ☆

```
Date,AAPL.Open,AAPL.High,AAPL.Low,AAPL.Close,AAPL.Volume,AAPL.Adjusted,dh,mavg,up,direction
2015-02-17,127.489998,128.880005,126.919998,127.830002,63152400,122.905254,106.7410523,117.9276669,129.1142814,Increasing
2015-02-18,127.629997,128.779999,127.449997,128.720001,44891700,123.760965,107.842423,118.9403335,130.0382439,Increasing
2015-02-19,128.479996,129.029999,128.330002,128.449997,37362400,123.501363,108.8942449,119.8891668,130.8840887,Decreasing
2015-02-20,128.619995,129.5,128.050003,129.5,48948400,124.510914,109.7854494,120.7635001,131.7415509,Increasing
2015-02-23,130.020004,133.129.660004,133.70974100,127.876074,110.3725162,121.7201668,133.0678174,Increasing
2015-02-24,132.940002,133.600006,131.169998,132.169998,69228100,127.078049,111.0948689,122.6648335,134.2347981,Decreasing
2015-02-25,131.559998,131.600006,128.149994,128.789993,74711700,123.828261,113.2119183,123.6296667,134.0474151,Decreasing
2015-02-26,128.789993,130.869995,126.610001,130.419998,91287500,125.395469,114.1652991,124.2823333,134.3993674,Increasing
2015-02-27,130.130.570007,128.240005,128.460007,62014800,123.510987,114.9668484,124.8426669,134.7184854,Decreasing
2015-03-02,129.25,130.279999,128.300003,129.089996,48096700,124.116706,115.8770904,125.4036668,134.9302432,Decreasing
2015-03-03,128.960007,129.520004,128.089996,129.360001,37816300,124.376308,116.9535132,125.9551669,134.9568205,Increasing
2015-03-04,129.100006,129.559998,128.320007,128.539993,31666300,123.587892,118.0874253,126.4730002,134.8585751,Decreasing
2015-03-05,128.580002,128.75,125.760002,126.410004,56517100,121.539962,119.1048311,126.848667,134.5925029,Decreasing
2015-03-06,128.399994,129.369995,126.260002,126.599998,72842100,121.722637,120.190797,127.2288335,134.26687,Decreasing
2015-03-09,127.959999,129.570007,125.059998,127.139999,88528500,122.241834,121.6289771,127.631167,133.6333568,Decreasing
2015-03-10,126.410004,127.220001,123.800003,124.510002,68856600,119.71316,123.1164763,127.9235004,132.7305246,Decreasing
2015-03-11,124.75,124.769997,122.110001,122.239998,68939000,117.530609,123.592756,128.0093337,132.4139113,Decreasing
2015-03-12,122.309998,124.900002,121.629997,124.449997,48362700,119.655466,123.4894559,127.9813337,132.4732114,Increasing
2015-03-13,124.400002,125.400002,122.580002,123.589996,51827300,118.828598,123.045606,127.8490003,132.6523946,Decreasing
2015-03-16,123.879997,124.949997,122.870003,124.949997,35874300,120.136203,122.6967016,127.7283335,132.7599655,Increasing
2015-03-17,125.900002,127.32,125.650002,127.040001,51023100,122.145688,122.616033,127.6680002,132.7199674,Increasing
2015-03-18,127.129.160004,126.370003,128.470001,65270900,123.520597,122.6064498,127.652167,132.6978842,Increasing
2015-03-19,128.75,129.25,127.400002,127.5,45809500,122.587966,122.5939029,127.6245004,132.6550879,Decreasing
2015-03-20,128.25,128.399994,125.160004,125.900002,68695100,121.049608,122.4865925,127.4980004,132.5094083,Decreasing
2015-03-23,127.120003,127.849998,126.519997,127.209999,37709700,122.309137,122.6741703,127.2633335,131.8524968,Increasing
2015-03-24,127.230003,128.039993,126.559998,126.690002,32842300,121.809174,123.0410183,127.0025001,130.9639818,Decreasing
2015-03-25,126.540001,126.82,123.379997,123.379997,51655200,118.626689,122.8276392,126.7531667,130.6786943,Decreasing
2015-03-26,122.760002,124.879997,122.599998,124.239998,47572900,119.453558,122.5538523,126.4835001,130.4131478,Increasing
2015-03-27,124.57,124.699997,122.910004,123.25,39546200,118.5017,122.2826504,126.2099998,130.1373491,Decreasing
2015-03-30,124.050003,126.400002,124.126.370003,47099700,121.501502,122.346906,126.0283332,129.7097604,Increasing
2015-03-31,126.089996,126.489998,124.360001,124.43,42090600,119.63624,122.395242,125.8334998,129.2717577,Decreasing
2015-04-01,124.82,125.120003,123.099998,124.25,40621400,119.463174,122.3761274,125.6009999,128.8258723,Decreasing
```



A5.3.3.2 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

```
Plotly.d3.csv("https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv", function(err, rows){

    function unpack(rows, key) {
        return rows.map(function(row) { return row[key]; });
    }

    var trace1 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL High',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.High'),
        line: {color: '#17BECF'}
    }

    var trace2 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL Low',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.Low'),
        line: {color: '#7F7F7F'}
    }

    var data = [trace1,trace2];
```



A5.3.3.3 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

```
var data = [trace1,trace2];  
  
var layout = {  
  title: 'AAPL Price Time Series',  
};  
  
Plotly.newPlot('myDiv', data, layout);
```

Time series by AA00



[2] Time series : financial data strings – set range

```
var data = [trace1, trace2];

var layout = {
  title: 'AAPL Price Time Series with range',
  xaxis: {
    range: ['2016-07-01', '2016-12-31'],
    type: 'date'
  },
  yaxis: {
    autorange: true,
    range: [86.8700008333, 138.870004167],
    type: 'linear'
  }
};

Plotly.newPlot('myDiv', data, layout);
```

Time series by AA00



날짜와 주가의 범위를 지정

[2] Time series : financial data strings – Range slider

```
var layout = {
  title: 'AAPL Price Time Series with rangeslider',
  xaxis: {
    autorange: true,
    range: ['2015-02-17', '2017-02-16'],
    rangeselector: {buttons: [
      {
        count: 1,
        label: '1m',
        step: 'month',
        stepmode: 'backward'
      },
      {
        count: 6,
        label: '6m',
        step: 'month',
        stepmode: 'backward'
      },
      {step: 'all'}
    ]},
    rangeslider: {range: ['2015-02-17', '2017-02-16']},
    type: 'date'
  },
  yaxis: {
    autorange: true,
    range: [86.8700008333, 138.870004167],
    type: 'linear'
  }
};
```


[2] Time series : financial data strings – Range slider

Time series by AA00

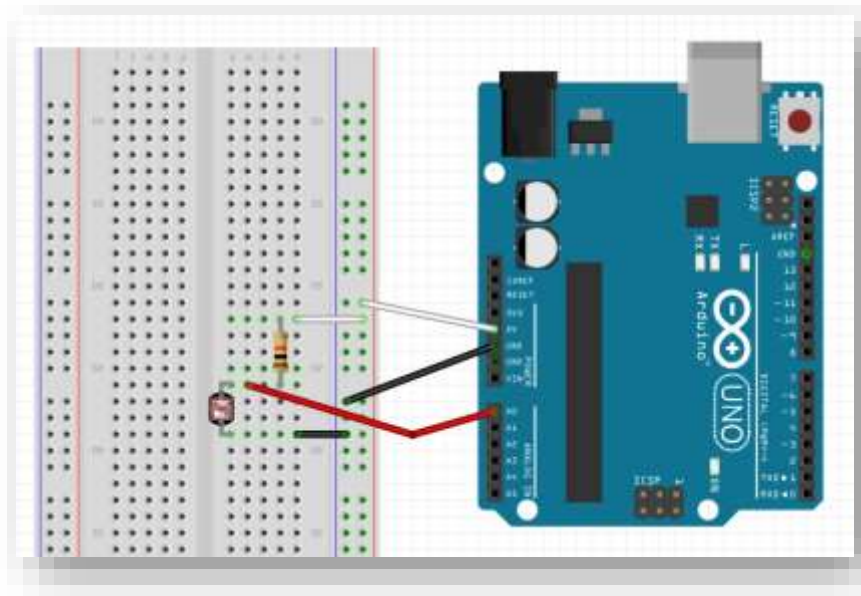
AAPL Price Time Series with rangeslider



[3] Time series : my lux data

```
'2015-11-05 12:09:41.382',
'2015-11-05 12:09:42.380',
'2015-11-05 12:09:43.378',
'2015-11-05 12:09:44.377',
'2015-11-05 12:09:45.375',
'2015-11-05 12:09:46.389',
'2015-11-05 12:09:47.388',
'2015-11-05 12:09:48.386',
'2015-11-05 12:09:49.384',
'2015-11-05 12:09:50.383',
'2015-11-05 12:09:51.381',
'2015-11-05 12:09:52.380',
'2015-11-05 12:09:53.394',
'2015-11-05 12:09:54.392',
'2015-11-05 12:09:55.391',
'2015-11-05 12:09:56.389',
'2015-11-05 12:09:57.387',
'2015-11-05 12:09:58.386',
'2015-11-05 12:09:59.384',
'2015-11-05 12:10:00.398',
'2015-11-05 12:10:01.397',
```

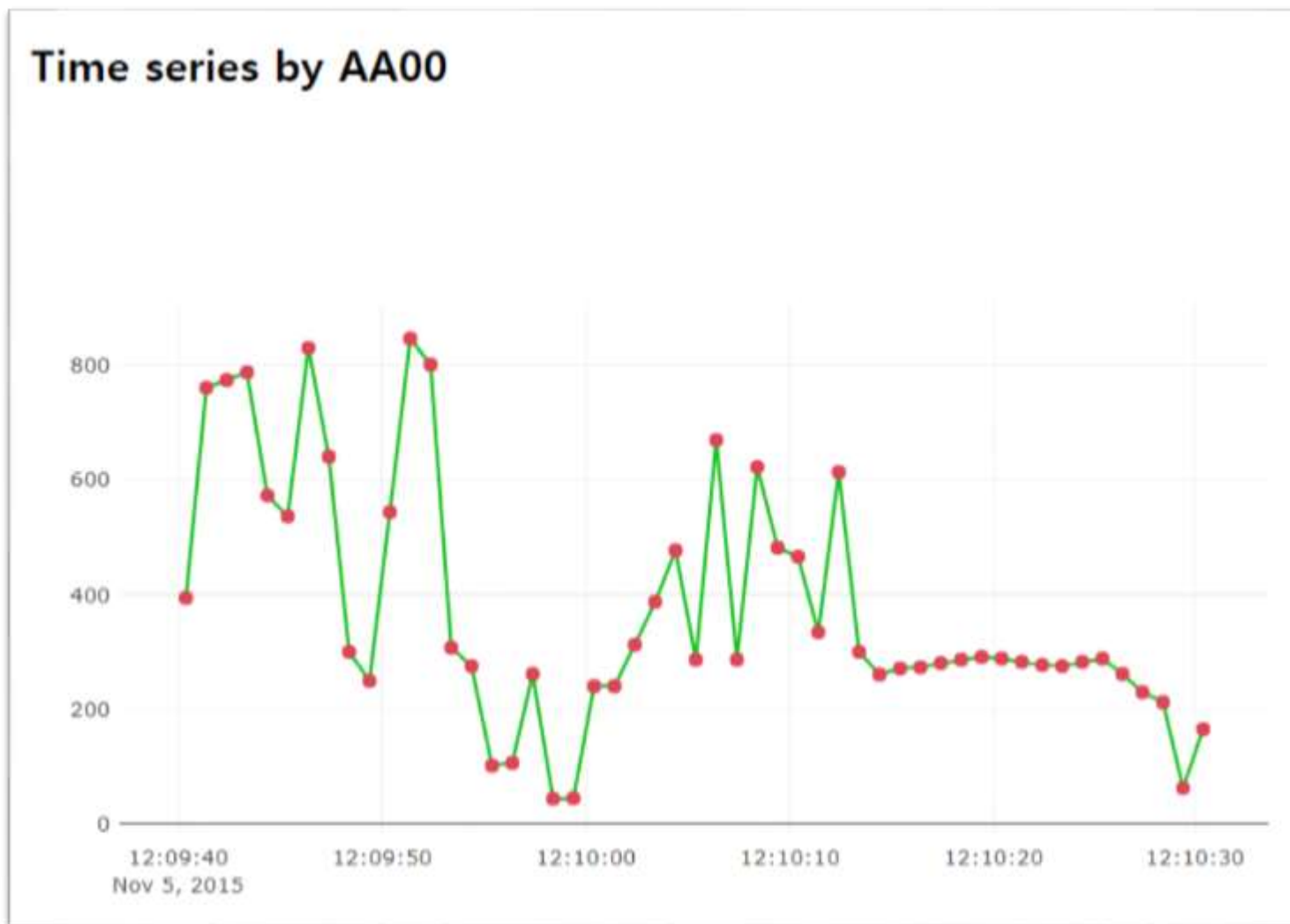
Data :
date,value





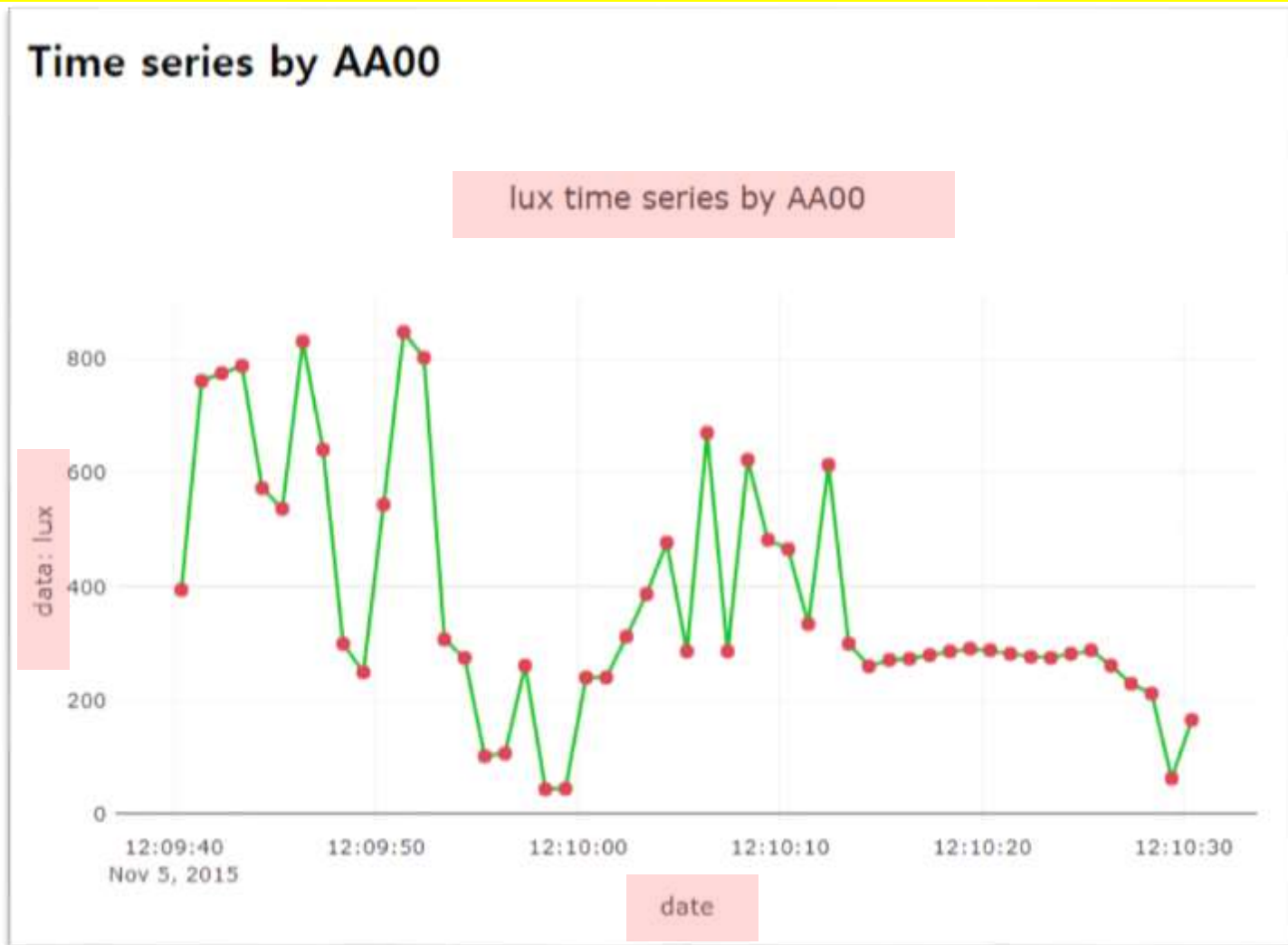
A5.3.4.2 plotly.js: Time series

[3] Time series : my lux data → DV_ts03_sensor_chart.html



A5.3.4.3 plotly.js: Time series

[3] Time series : my lux data – [DIY] → Set title and axis title



AAnn_lux_Time_Series.png

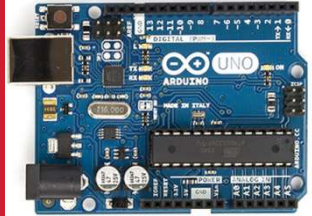


Project: Time series with Rangelslider

[Project-DIY] AAnn_lux_Rangelslider.html



AAnn_lux_Rangelslider.png



[Practice]

◆ [wk09]

- Charts by plotly
- Complete your plotly chart project
- Upload folder: AAnn_Rpt07

◆ [Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

제출폴더명 : **AAnn_Rpt07**

- 압축할 파일들

- ① **AAnn_Chart_Layout.png**
- ② **AAnn_Axis_Title.png**
- ③ **AAnn_Line_Dash_Dot.png**
- ④ **AAnn_lux_Time_Series.png**
- ⑤ **AAnn_lux_Rangeslider.png**
- ⑥ **All *.ino**
- ⑦ **All *.js**
- ⑧ **All *.html**

[Upload to github]

◆ [wk09]

- upload all work of this week
- Use repo “aann” in github
- upload folder “aann_rpt07” in your github.

● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <https://plot.ly/> plotly
- ✓ <https://www.mongodb.com/> MongoDB
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub



주교재 및 참고도서

아두이노와 Node.js에 기반한 IOT 신호 시각화 | 저자 이 상 훈 | 인제대학교 출판부

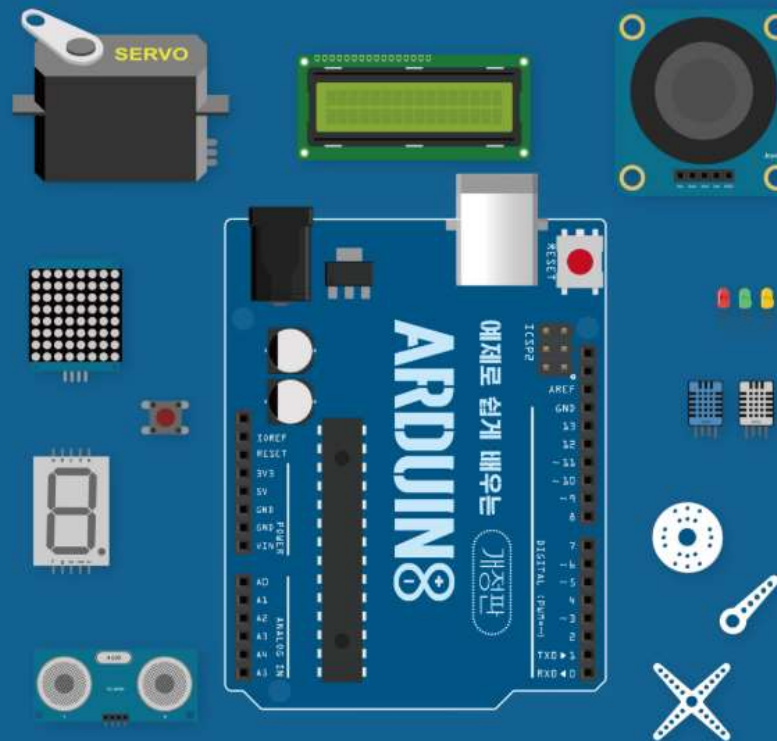
아두이노와 Node.js에 기반한

IOT 신호 시각화

| 저자 이 상 훈 |



인제대학교 출판부



예제로 쉽게 배우는

아두이노

개정판

장성용 · 김진환 지음



Target of this class

Real-time Weather Station from sensors



on Time: 2018-01-22 17:58:31.012



Another target of this class

PPG with rangeslider

