

DATA VISUALIZATION WITH OBSERVABLE

Andrew Mok, 2020

Andrew Mok, 2020

TABLE OF CONTENTS

01



Why
Visualization?

02



Data Sources

03



Tools & Demo

04

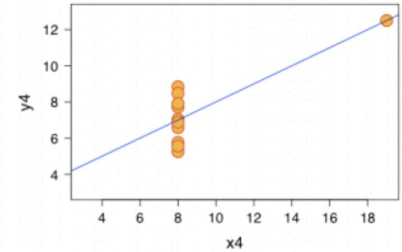
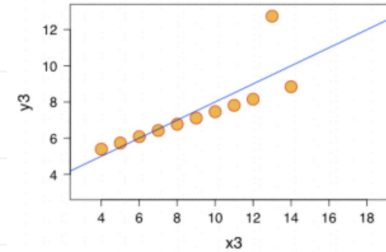
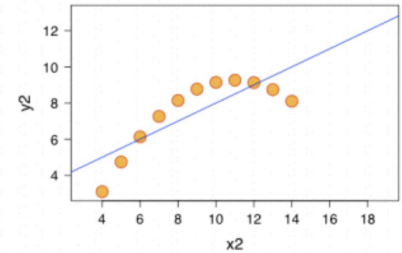
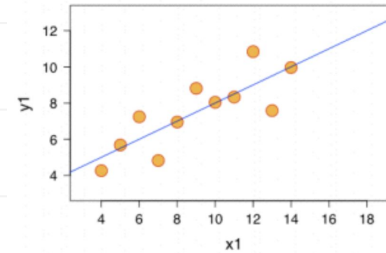


Others

01 WHY VISUALIZATION?

Anscombe's Quartet: Raw Data

	1		2		3		4	
	X	Y	X	Y	X	Y	X	Y
	10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
	8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
	13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
	9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
	11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
	14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
	6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
	4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
	12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
	7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
	5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89
Mean	9.0	7.5	9.0	7.5	9.0	7.5	9.0	7.5
Variance	10.0	3.75	10.0	3.75	10.0	3.75	10.0	3.75
Correlation	0.816		0.816		0.816		0.816	

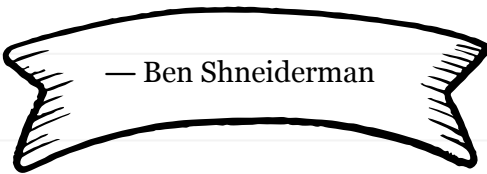


Anscombe's Quartet

Anscombe, 1973



**"THE PURPOSE OF VISUALIZATION IS
INSIGHT, NOT PICTURES."**



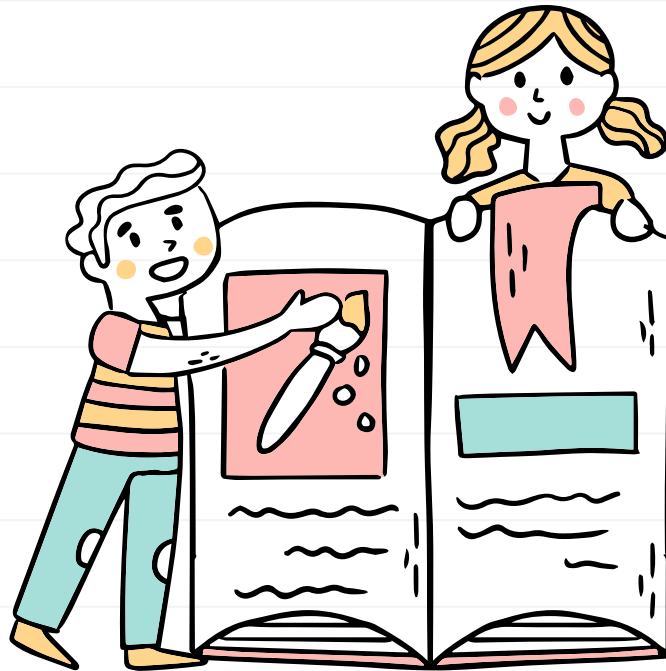
— Ben Shneiderman



02 WHERE CAN I FIND DATA?

Hong Kong

- [Data.gov.hk](https://data.gov.hk)
 - Traffic snapshots
 - Weather forecast
 - Coronavirus disease
- [Hong Kong Geodata Store](https://data.gov.hk/geodata)
- [HKMA Open API](https://hkma.gov.hk/open-api)
- [APIX \(JETCO\)](https://api.jetc.co)



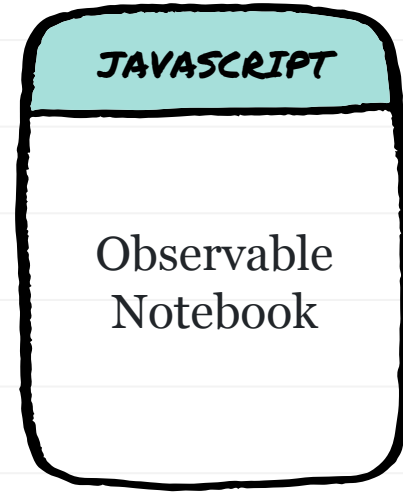
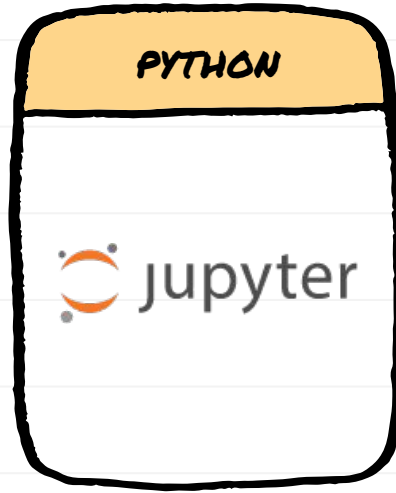
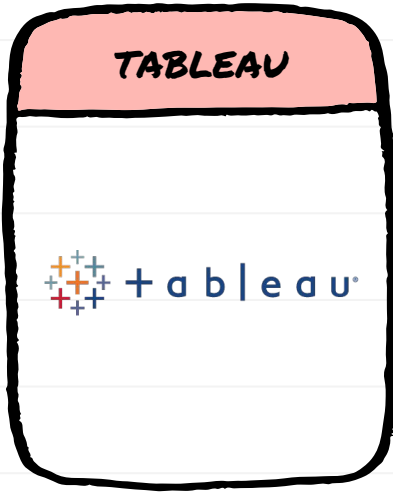
02 WHERE CAN I FIND DATA?

Worldwide

- [National Data by Chinese Government](#)
- [U.S. Government's open data](#)
- [World Bank Open Data](#)
- [WHO Data repository](#)
- [@awesomedata/awesome-public-datasets](#)
- [Open Data Inception](#)



03 TOOLS FOR VISUALIZATION

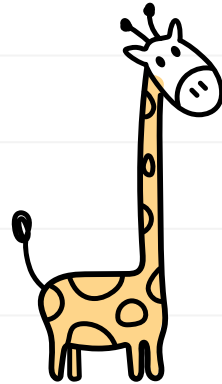


- Plotly
- Matplotlib

- d3.js
- Vega-lite
- Plotly

DEMO

[Observablehq.com](https://observablehq.com)



DEMO

Introduction

- [Observable Notebook 101](#)

Demo (Observable + Plotly + Open data by HK Gov.)

- [Latest situation of reported cases of COVID-19 in Hong Kong](#)
- [How COVID-19 spreaded to the World \(Outside Mainland China\)](#)
- [Buildings in which confirmed cases have resided in past 14 days](#)

04

OTHER DATA PORTALS OF COVID-19

Tableau

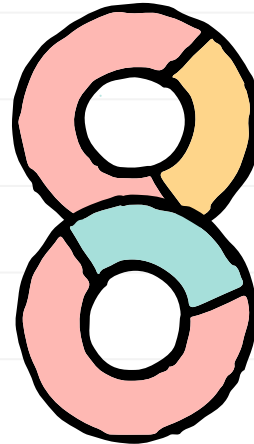
- [COVID-19 Dashboard](#)

Johns Hopkins University

- [Visual Dashboard by JHU](#)

Hong Kong Government

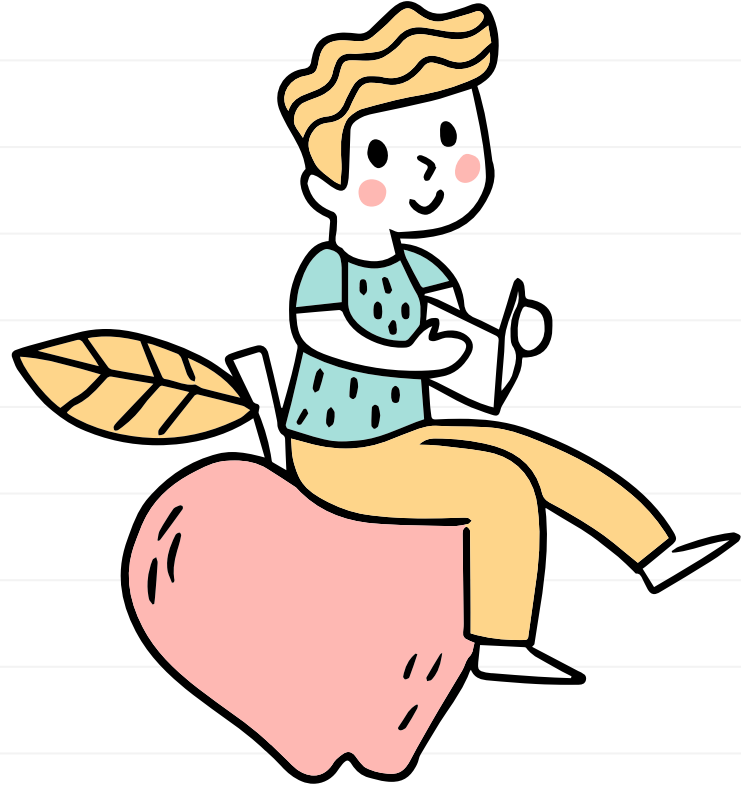
- [Latest Situation of Coronavirus Disease \(COVID-19\) in Hong Kong](#)



THANKS!

STAY CALM AND DON'T PANIC.

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.



REFERENCES

Stewart, M. (2019, May 16). The Power of Visualization in Data Science. Retrieved from

<https://towardsdatascience.com/the-power-of-visualization-in-data-science-1995d56e4208>

Tsang, B. (2020, February 19). Data Visualisation. Retrieved from

<https://bobtsang.github.io/2020/02/19/Data-Visualisation.html>

Tufte, E. R. (2016). *Visual and statistical thinking: displays of evidence for making decisions*.

Cheshire, CT: Graphics Press.



Andrew Mok • andrewmmc.com

I am a web developer from Hong Kong. I write and I code.



Published Mar 20

Observable Notebook 101

md`# Observable Notebook 101`

A notebook is made up of a series of **cells**, and each cell is defined by its JavaScript source code.

5

```
// basic calculation
2 + 3
```

5050

```
// return value from code block
{
  let sum = 0;
  for (let i = 0; i <= 100; ++i) {
    sum += i;
  }
  return sum;
}
```

```
color = "red"
```

```
// declare variables
color = "red"
```

```
"My favorite color is red."
```

```
// re-evaluated automatically when reference values changed
`My favorite color is ${color}.`
```

Hello *world*.

```
// support HTML & markdown
// can generate DOM by standard DOM API (document.createElement)
// or using build-in template literal
html`<span style="background:yellow;">
  Hello <i>world</i>.
</span>`
```

```
status = ▶ Object {resolved: 2020-03-20T03:44:58.307}
```

```
// support Promise for async function
```

```
status = new Promise(resolve => {  
  setTimeout(() => {  
    resolve({resolved: new Date});  
  }, 2000);  
})
```

```
_ = f(value)
```

```
// support npm library
```

```
_ = require('lodash')
```

```
// _.map()
```



```
// support dynamic input from DOM element
```

```
viewof value = html`<input type=range>`
```

```
50
```

```
value
```

```
rawForecast = ▶ Object {@context: Array(2), type: "Feature", geometry: Object, property:
```

```
// support Fetch API
```

```
rawForecast = (await  
fetch("https://api.weather.gov/gridpoints/OKX/33,37/forecast")).json()
```





Andrew Mok • andrewmmc.com

I am a web developer from Hong Kong. I write and I code.



Published Mar 20

Latest situation of reported cases of COVID-19 in Hong Kong

md`# Latest situation of reported cases of COVID-19 in Hong Kong`

```
plotly = ▶Object {version: "1.52.3", register: f(_modules), plot: f(gd, data, layout,
// https://plot.ly/javascript/
plotly = require('plotly.js-dist')}

data = ▶Object {header: Array(9), rows: Array(72)}

confirmedCases = ▶Object {x: Array(72), y: Array(72), type: "line", name: "Confirmed c
confirmedCases = ({ x: data.rows.map(row => row[0]), y: data.rows.map(row =>
row[2]), type: 'line', name: 'Confirmed cases' })

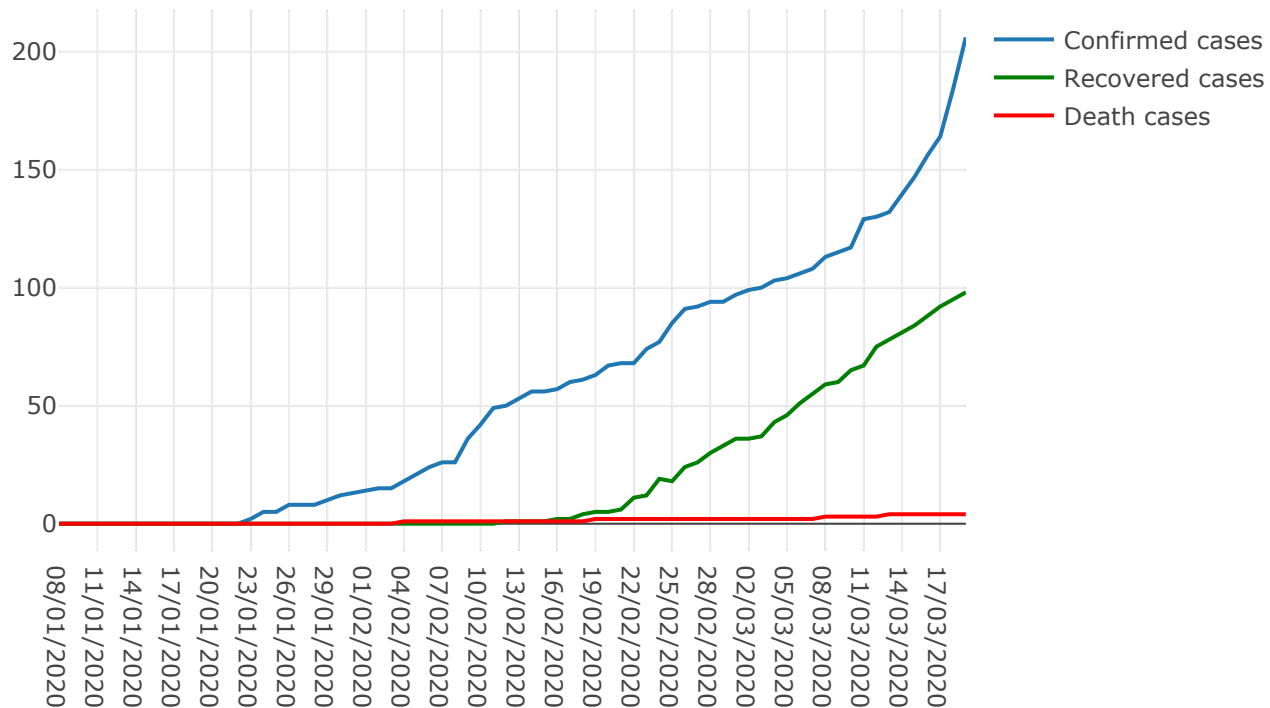
recoveredCases = ▶Object {x: Array(72), y: Array(72), type: "line", name: "Recovered c
recoveredCases = ({ x: data.rows.map(row => row[0]), y: data.rows.map(row =>
row[7]), type: 'line', name: 'Recovered cases', line: { color: 'green' } })

deathCases = ▶Object {x: Array(72), y: Array(72), type: "line", name: "Death cases", 1
deathCases = ({ x: data.rows.map(row => row[0]), y: data.rows.map(row => row[6]),
type: 'line', name: 'Death cases', line: { color: 'red' } })

parsedData = ▶Array(3) [Object, Object, Object]
parsedData = [confirmedCases, recoveredCases, deathCases];

options = ▶Object {title: "Latest situation of reported cases of COVID-19 in Hong Kong
options = ({
  title: 'Latest situation of reported cases of COVID-19 in Hong Kong'
})
```

Latest situation of reported cases of COVID-19 in Hong Kong



```
// create DOM element here for printing graph
{ const p = document.createElement("p"); plotly.newPlot(p, parsedData, options);
return p; }
```



[Publish](#)

Andrew Mok • andrewmmc.com

I am a web developer from Hong Kong. I write and I code.



Published Mar 20

How COVID-19 spreaded to the World (Outside Mainland China)

```
plotly = > Object {version: "1.52.3", register: f(_modules), plot: f(gd, data, layout,
```

```
lodash = f(value)
```

```
lodash = require('lodash')
```

```
convertDate = f(...)
```

```
convertDate = (dateString = "") => new Date(dateString.split("/").reverse().join("-").getTime()
```

```
data = > Object {header: Array(6), rows: Array(6196)}
```

```
// Countries/areas outside Mainland China have reported cases of COVID-19
```

```
// https://data.gov.hk/en-data/dataset/hk-dh-chpsebcddr-novel-infectious-
```

```
agent/resource/aeaf9663-f60c-48dd-97d0-13cace6f604b
```

```
data = (await fetch('https://api.data.gov.hk/v1/filter?
q=%7B%22resource%22%3A%22http%3A%2F%2Fwww.chp.gov.hk%2Ffiles%2Fmisc%2Fcountries_area
s_outside_mainland_china_have_reported_cases_eng.csv%22%2C%22section%22%3A1%2C%22for
mat%22%3A%22json%22%7D')).json()
```

```
// do some data massage here to convert date string to timestamp
```

```
dataWithTimestamp = data.rows.map((row, i) => { const [date, ...others] = row;
return [convertDate(date), ...others] })
```

```
// create date slider
```

```
viewof date = html`<input type=range min=${convertDate('15/02/2020')}
max=${convertDate('19/03/2020')} step=${60*60*24}>`
```

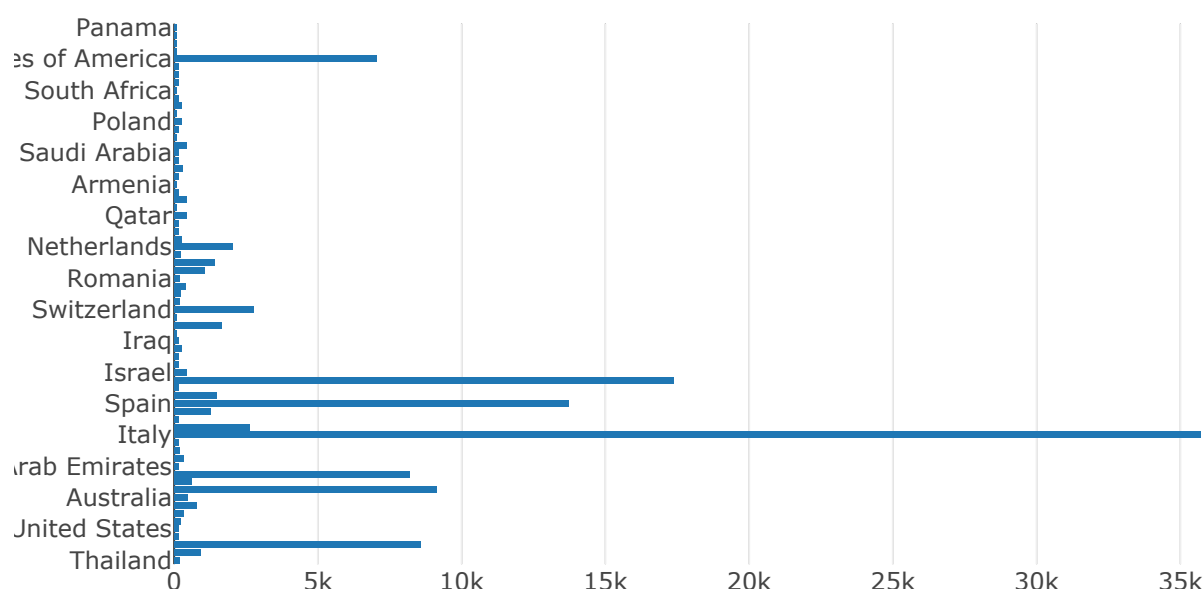
```
confirmedCases = > Object {Thailand: 177, Japan: 914, Korea: 8565, Taiwan: 108, United
```

```
// filter confirmed cases before date selected
confirmedCases = dataWithTimestamp.reduce((arr, row) => { if (row[0] > date) return
arr; arr[row[2]] = row[3]; return arr }, {})
```

```
confirmedCasesMoreThan50 = lodash.pickBy(confirmedCases, cases => cases >= 50)
```

```
parsedData = [{ x: Object.values(confirmedCasesMoreThan50), y:
Object.keys(confirmedCasesMoreThan50), type: 'bar', orientation: 'h' }]
```

Thu Mar 19 2020 08:00:00 GMT+0800 (Hong Kong Standard Time)



```
// create DOM element here for printing graph
{
  const p = document.createElement("p");
  plotly.newPlot(p, parsedData, { title: new Date(date).toString() });
  return p;
}
```

[Publish](#)

Andrew Mok • andrewmmc.com

I am a web developer from Hong Kong. I write and I code.



Published Mar 20

Fork of Buildings in which confirmed cases have resided in past 14 days

Buildings in which confirmed cases have resided in past 14 days

```
md`# Buildings in which confirmed cases have resided in past 14 days`
```

```
plotly = ▶ Object {version: "1.52.3", register: f(_modules), plot: f(gd, data, layout,
plotly = require('plotly.js-dist')}
```

```
_ = f(value)
```

```
_ = require('lodash')
```

```
mapData = ▶ Object {type: "FeatureCollection", features: Array(18)}
```

```
// GeoJSON data of Hong Kong
```

```
// https://data.gov.hk/en-data/dataset/hk-had-json1-hong-kong-administrative-
boundaries
```

```
mapData = (await fetch('https://cors-
anywhere.herokuapp.com/https://had.gov.hk/psi/hong-kong-administrative-
boundaries/hksar_18_district_boundary.json')).json()
```

```
// Residential buildings in which probable/confirmed cases have resided in the past
14 days or non-residential building with 2 or more probable/confirmed cases in the
past 14 days
```

```
// https://data.gov.hk/en-data/dataset/hk-dh-chpsebcdrr-novel-infectious-
agent/resource/3b5a7266-9d35-4d9c-a31c-b5208be4fdc9
```

```
data = (await fetch('https://api.data.gov.hk/v2/filter?
q=%7B%22resource%22%3A%22http%3A%2F%2Fwww.chp.gov.hk%2Ffiles%2Fmisc%2Fbuilding_list_
eng.csv%22%2C%22section%22%3A1%2C%22format%22%3A%22json%22%7D')).json())
```

```
// do some data massage here to fix typo
```

```
fixedData = data.reduce((acc, curr) => {
```

```
  let confirmedCase = curr;
```

```
  if (confirmedCase.District === 'Central and Western') confirmedCase.District =
  'Central & Western';
```

```

    'Central & Western',
    if (confirmedCase.District === 'Yau Tsim Mon') confirmedCase.District = 'Yau Tsim
Mong';
    if (confirmedCase.District === 'Yuen Long District') confirmedCase.District = 'Yuen
Long';
    return [...acc, confirmedCase]
}, []);

```

```

parsedData = _.countBy(fixedData, 'District')

```

```

// create DOM element here
{
    const data = [{
        type: 'choroplethmapbox',
        geojson: mapData,
        featureidkey: 'properties.District',
        locations: Object.keys(parsedData),
        z: Object.values(parsedData), // count of confirmed cases by locations
    }];

```

```
const layout = {
  mapbox: {
    center: { lon: 114.157, lat: 22.285 },
    zoom: 9
  },
  width: 900,
  height: 600
};
const p = document.createElement("p");
plotly.newPlot(p, data, layout, { mapboxAccessToken: mapboxAccessToken });
return p;
}
```

