

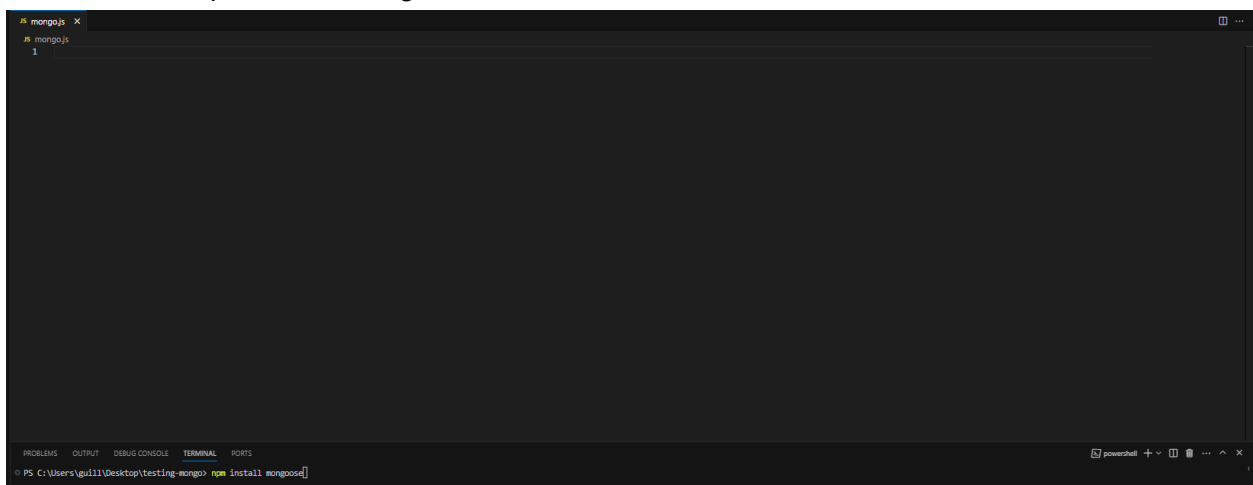
# MongoDB Documentation (Initial Configuration & Storing)

## Ensemble Generation

### Initial Configuration

The full file mongo.js can be found [here](#).

1. Download and install [MongoDB Compass](#).
2. Start with an empty .js file. (In this example we write to a file called **mongo.js** in a folder called **testing-mongodb**)
3. In order to start setup you need to install the mongoose dependency (open terminal and run)
  - npm install mongoose



- Note: Dependency is used to reduce the number of code and lines in order to setup the mongo debugging.
4. Declare the exact dependency needed: **Steps 4 - 8 are written on the js file **mongo.js****

```
const mongoose=require('mongoose')
```

5. Connect to local mongodb:

```
mongoose.connect("mongodb://localhost:27017/testing-mongodb")
.then(()=> {
  console.log('mongodb connected');
})
.catch(()=> {
  console.log('error')
})
```

- **[localhost:27017]**: default identifier of local for mongodb
  - Note [localhost:27017] might not work for everyone. In such a case try using **[0.0.0.0:27017]** instead.
- **[testing-mongodb]**: the name of your database

```
.then(() => {
  console.log('mongodb connected');
})
```

- What is fired off when a connection is successful.

```
.catch(() => {
  console.log('error')
})
```

- 
- What is fired off when encountering an error leading to a failed connection.

6. After connecting to mongodb you must define a [schema](#) which declares how your document will be presented. **In this example we are going to store the following:**  
**Take a moment to review this data.**

```
let data = {
  "eg_histogram": [
    [
      1,
      0,
      0,
      0,
      0,
      0,
      1,
      0,
      1,
      13
    ],
    [
      0.03653414175356576,
      0.044787688764618416,
      0.053041235775671075,
      0.06129478278672374,
```

```
        0.06954832979777639,  
        0.07780187680882905,  
        0.08605542381988172,  
        0.09430897083093438,  
        0.10256251784198704,  
        0.1108160648530397,  
        0.11906961186409236  
    ]  
],  
"d_districts_histogram": [  
    [  
        0,  
        0,  
        0,  
        0,  
        0,  
        16,  
        0,  
        0,  
        0,  
        0  
    ],  
    [  
        0.5,  
        0.6,  
        0.7,  
        0.8,  
        0.9,  
        1.0,  
        1.1,  
        1.2000000000000002,  
        1.3,  
        1.4,  
        1.5  
    ]  
],  
"r_districts_histogram": [  
    [  
        0,  
        0,
```

```
        0,  
        0,  
        0,  
        16,  
        0,  
        0,  
        0,  
        0  
    ],  
    [  
        5.5,  
        5.6,  
        5.7,  
        5.8,  
        5.9,  
        6.0,  
        6.1,  
        6.2,  
        6.3,  
        6.4,  
        6.5  
    ]  
],  
"compactness_histogram": [  
    [  
        1,  
        1,  
        0,  
        2,  
        4,  
        5,  
        2,  
        0,  
        0,  
        1  
    ],  
    [  
        0.11790689066657764,  
        0.11984717405653585,  
        0.12178745744649405,
```

```
        0.12372774083645224,  
        0.12566802422641046,  
        0.12760830761636865,  
        0.12954859100632685,  
        0.13148887439628507,  
        0.13342915778624326,  
        0.13536944117620148,  
        0.13730972456615967  
    ]  
],  
  "split_county_histogram": [  
    [  
      5,  
      4,  
      2,  
      2,  
      0,  
      1,  
      1,  
      0,  
      0,  
      1  
    ],  
    [  
      11.010502948771734,  
      25.572159728680134,  
      40.13381650858853,  
      54.69547328849693,  
      69.25713006840533,  
      83.81878684831372,  
      98.38044362822212,  
      112.94210040813051,  
      127.50375718803892,  
      142.06541396794734,  
      156.62707074785573  
    ]  
  ],  
  "vote_split_histogram": [  
    [  
      0,  

```

```
        0,  
        0,  
        0,  
        0,  
        16,  
        0,  
        0,  
        0,  
        0  
    ],  
    [  
        -5.5,  
        -5.4,  
        -5.3,  
        -5.2,  
        -5.1,  
        -5.0,  
        -4.9,  
        -4.8,  
        -4.7,  
        -4.6,  
        -4.5  
    ]  
],  
"eq_pop_histogram": [  
    [  
        11,  
        4,  
        0,  
        0,  
        0,  
        0,  
        0,  
        0,  
        0,  
        1  
    ],  
    [  
        0.03429550872384382,  
        0.04387046134315768,
```

```
        0.053445413962471544,  
        0.0630203665817854,  
        0.07259531920109927,  
        0.08217027182041313,  
        0.09174522443972699,  
        0.10132017705904085,  
        0.11089512967835471,  
        0.12047008229766858,  
        0.13004503491698244  
    ]  
],  
  "dem_responsiveness_histogram": [  
    [  
      1,  
      3,  
      10,  
      0,  
      1,  
      0,  
      0,  
      0,  
      0,  
      1  
    ],  
    [  
      0.1428571428571429,  
      0.2485714285714287,  
      0.3542857142857145,  
      0.4600000000000003,  
      0.5657142857142861,  
      0.6714285714285718,  
      0.7771428571428577,  
      0.8828571428571435,  
      0.9885714285714292,  
      1.094285714285715,  
      1.2000000000000006  
    ]  
  ],  
  "rep_responsiveness_histogram": [  
    [  

```

```
1,
3,
10,
0,
1,
0,
0,
0,
0,
0,
1
],
[
0.14285714285714346,
0.248571428571429,
0.35428571428571454,
0.46000000000000001,
0.5657142857142856,
0.6714285714285712,
0.7771428571428567,
0.8828571428571422,
0.9885714285714278,
1.0942857142857134,
1.1999999999999998
]
],
"dem_bias_histogram": [
[
14,
1,
0,
0,
0,
0,
0,
0,
0,
0,
1
],
[
-0.3514285714285714,
```







```
        0.0,  
        0.1,  
        0.2,  
        0.30000000000000004,  
        0.4,  
        0.5,  
        0.6000000000000001,  
        0.7000000000000001,  
        0.8,  
        0.9,  
        1.0  
    ]  
],  
  "latino_min_majority_histogram": [  
    [  
      0,  
      0,  
      0,  
      0,  
      0,  
      16,  
      0,  
      0,  
      0,  
      0  
    ],  
    [  
      -0.5,  
      -0.4,  
      -0.3,  
      -0.19999999999999996,  
      -0.09999999999999998,  
      0.0,  
      0.10000000000000009,  
      0.20000000000000007,  
      0.30000000000000004,  
      0.4,  
      0.5  
    ]  
  ],  
]
```

```
"asian_min_majority_histogram": [  
  [  
    0,  
    0,  
    0,  
    0,  
    0,  
    16,  
    0,  
    0,  
    0,  
    0  
  ],  
  [  
    -0.5,  
    -0.4,  
    -0.3,  
    -0.19999999999999996,  
    -0.09999999999999998,  
    0.0,  
    0.100000000000000009,  
    0.200000000000000007,  
    0.300000000000000004,  
    0.4,  
    0.5  
  ]  
],  
"democrat_min_majority_histogram": [  
  [  
    0,  
    0,  
    0,  
    0,  
    0,  
    16,  
    0,  
    0,  
    0,  
    0  
  ],  
  ]
```

```
[
  -0.5,
  -0.4,
  -0.3,
  -0.19999999999999996,
  -0.09999999999999998,
  0.0,
  0.10000000000000009,
  0.20000000000000007,
  0.30000000000000004,
  0.4,
  0.5
],
"republican_min_majority_histogram": [
  0,
  0,
  0,
  0,
  0,
  0,
  16,
  0,
  0,
  0,
  0
],
[
  -0.5,
  -0.4,
  -0.3,
  -0.19999999999999996,
  -0.09999999999999998,
  0.0,
  0.10000000000000009,
  0.20000000000000007,
  0.30000000000000004,
  0.4,
  0.5
]
```

```
],
"boxWhiskers": {
  "black": {
    "group": "black",
    "minimum": [
      0.07473518158247104,
      0.15092319208217242,
      0.17340723335242386,
      0.2421859248544354,
      0.26777921372515967,
      0.28304794773092634,
      0.37702654909143024
    ],
    "firstQ": [
      0.07473518158247104,
      0.17340723335242386,
      0.17760547557356504,
      0.2593860062690989,
      0.26777921372515967,
      0.3117577124554551,
      0.5905034442985214
    ],
    "median": [
      0.07473518158247104,
      0.17340723335242386,
      0.17760547557356504,
      0.25939036719450337,
      0.26777921372515967,
      0.3117577124554551,
      0.5981348015369794
    ],
    "thirdQ": [
      0.07473518158247104,
      0.17340723335242386,
      0.17955400068181515,
      0.25939036719450337,
      0.26777921372515967,
      0.3122432357505054,
      0.5981348015369794
    ]
  },
}
```

```
      "maximum": [  
        0.11771840343726214,  
        0.17760547557356504,  
        0.25939036719450337,  
        0.26777921372515967,  
        0.3117577124554551,  
        0.3547661160294695,  
        0.5986788783921014  
      ],  
    },  
    "latino": {  
      "group": "latino",  
      "minimum": [  
        0.03716550037939465,  
        0.038066082550406416,  
        0.044869054224922905,  
        0.04531490414604732,  
        0.054003751035397354,  
        0.060660003566279916,  
        0.0650197783327598  
      ],  
      "firstQ": [  
        0.03739819956036172,  
        0.03851088752753028,  
        0.04510418767840879,  
        0.045895119090509424,  
        0.05878052218543678,  
        0.06494450018285998,  
        0.06520802445016566  
      ],  
      "median": [  
        0.03739819956036172,  
        0.03851088752753028,  
        0.04510418767840879,  
        0.046022500693672724,  
        0.060270272025351804,  
        0.0650197783327598,  
        0.06520802445016566  
      ],  
      "thirdQ": [  
        0.03739819956036172,  
        0.03851088752753028,  
        0.04510418767840879,  
        0.046022500693672724,  
        0.060270272025351804,  
        0.0650197783327598,  
        0.06520802445016566  
      ],  
      "maximum": [  
        0.11771840343726214,  
        0.17760547557356504,  
        0.25939036719450337,  
        0.26777921372515967,  
        0.3117577124554551,  
        0.3547661160294695,  
        0.5986788783921014  
      ]  
    }  
  ]  
}
```

```
        0.03739819956036172,  
        0.03851135630623474,  
        0.04525839379387276,  
        0.04678877023029504,  
        0.060270272025351804,  
        0.0650197783327598,  
        0.06520802445016566  
    ],  
    "maximum": [  
        0.03851088752753028,  
        0.04510418767840879,  
        0.045895119090509424,  
        0.051693877852012446,  
        0.060270272025351804,  
        0.0650197783327598,  
        0.06674317459564909  
    ]  
},  
"asian": {  
    "group": "asian",  
    "minimum": [  
        0.005230229696180741,  
        0.010744889035610498,  
        0.012980005801119911,  
        0.015494284871448988,  
        0.016417984125017254,  
        0.01848080330141466,  
        0.022653697100989124  
    ],  
    "firstQ": [  
        0.006328725771116479,  
        0.010744889035610498,  
        0.013556011956063301,  
        0.015494284871448988,  
        0.018471364145154514,  
        0.019503022115377126,  
        0.026019026019026018  
    ],  
    "median": [  
        0.006328725771116479,
```



```
        0.010744889035610498,  
        0.013556011956063301,  
        0.015494284871448988,  
        0.01848080330141466,  
        0.019504782684927238,  
        0.026019026019026018  
    ],  
    "thirdQ": [  
        0.006328725771116479,  
        0.010894862258368016,  
        0.013556011956063301,  
        0.015494473477613504,  
        0.01848080330141466,  
        0.019504782684927238,  
        0.026019026019026018  
    ],  
    "maximum": [  
        0.006328725771116479,  
        0.012249761901410763,  
        0.01456086827217713,  
        0.015612589121524616,  
        0.018529080027567896,  
        0.022012655095013354,  
        0.026019026019026018  
    ]  
},  
    "democrat": {  
        "group": "democrat",  
        "minimum": [  
            0.08420118511563059,  
            0.12567316089325073,  
            0.13863589539265214,  
            0.15248874948861313,  
            0.1598588622179842,  
            0.16170518054050953,  
            0.22374968938627987  
        ],  
        "firstQ": [  
            0.08832107337914774,  
            0.13863589539265214,
```

```
        0.14911460152824124,  
        0.1598588622179842,  
        0.1602347601282456,  
        0.1632871368795484,  
        0.3061428585776602  
    ],  
    "median": [  
        0.08832107337914774,  
        0.13863589539265214,  
        0.15248874948861313,  
        0.1598588622179842,  
        0.1602347601282456,  
        0.1632871368795484,  
        0.30732287201202413  
    ],  
    "thirdQ": [  
        0.08832107337914774,  
        0.13863589539265214,  
        0.15248874948861313,  
        0.16004598456076782,  
        0.16129547925945115,  
        0.16366145605413127,  
        0.30732287201202413  
    ],  
    "maximum": [  
        0.08977256185056046,  
        0.14356659005042088,  
        0.1598588622179842,  
        0.1602347601282456,  
        0.1632871368795484,  
        0.22029084778736427,  
        0.3078783568630089  
    ]  
},  
    "republican": {  
        "group": "republican",  
        "minimum": [  
            0.1362110334797331,  
            0.264491954691676,  
            0.2671645374348077,
```

```
        0.2769685113124836,  
        0.28435381634362067,  
        0.2897975432536853,  
        0.35743044737048024  
    ],  
    "firstQ": [  
        0.14143733920360851,  
        0.2671645374348077,  
        0.2769685113124836,  
        0.28306448813374685,  
        0.28970677051034943,  
        0.31087516965056855,  
        0.3701018388227718  
    ],  
    "median": [  
        0.14143733920360851,  
        0.2671645374348077,  
        0.2769685113124836,  
        0.28435381634362067,  
        0.2897975432536853,  
        0.31087516965056855,  
        0.3701018388227718  
    ],  
    "thirdQ": [  
        0.14303446160752423,  
        0.2671645374348077,  
        0.2771837996882832,  
        0.28435381634362067,  
        0.2897975432536853,  
        0.31094770985909026,  
        0.3701018388227718  
    ],  
    "maximum": [  
        0.2095094472587764,  
        0.27144760138268276,  
        0.28435381634362067,  
        0.2897975432536853,  
        0.29436644498398956,  
        0.3301731548569323,  
        0.3701018388227718
```

```

    ]
  }
}
}

```

- Our data stored ***must*** match the schema by layout and types. From this we see our schema is the following:

```

const ensemble_schema=new mongoose.Schema({
  eg_histogram: {
    type: [[Number]],    // the type of this variable
    required: true       // set as a required field
  },
  d_districts_histogram: {
    type: [[Number]],
    required: true
  },
  r_districts_histogram: {
    type: [[Number]],
    required: true
  },
  compactness_histogram: {
    type: [[Number]],
    required: true
  },
  split_county_histogram: {
    type: [[Number]],
    required: true
  },
  vote_split_histogram: {
    type: [[Number]],
    required: true
  },
  eq_pop_histogram: {
    type: [[Number]],
    required: true
  }
})

```

```
    },
    dem_responsiveness_histogram: {
      type: [[Number]],
      required: true
    },
    rep_responsiveness_histogram: {
      type: [[Number]],
      required: true
    },
    dem_bias_histogram: {
      type: [[Number]],
      required: true
    },
    rep_bias_histogram: {
      type: [[Number]],
      required: true
    },
    symmetry_histogram: {
      type: [[Number]],
      required: true
    },
    black_min_majority_histogram: {
      type: [[Number]],
      required: true
    },
    latino_min_majority_histogram: {
      type: [[Number]],
      required: true
    },
    asian_min_majority_histogram: {
      type: [[Number]],
      required: true
    },
    democrat_min_majority_histogram: {
      type: [[Number]],
```

```

        required: true
      },
      republican_min_majority_histogram: {
        type: [[Number]],
        required: true
      },
      boxWhiskers: {
        type: Object,
        required: true
      }
    })
  })
}

```

-

7. A collection must now be defined before the final step with the name of the collection and the schema to be followed specified:

```
const collection=new mongoose.model('ensemble_collection',ensemble_schema)
```

- **[ensemble\_collection]**: the name of the collection
- **[ensemble\_schema]**: the schema the documents in the collection shall follow

8. Add 1 more dependency to run the server quicker.

- npm install nodemon

Then run nodemon with the specified path

- nodemon mongo.js
- **[mongo.js]**: the specified path

Successful connection will print out your defined output in step 5

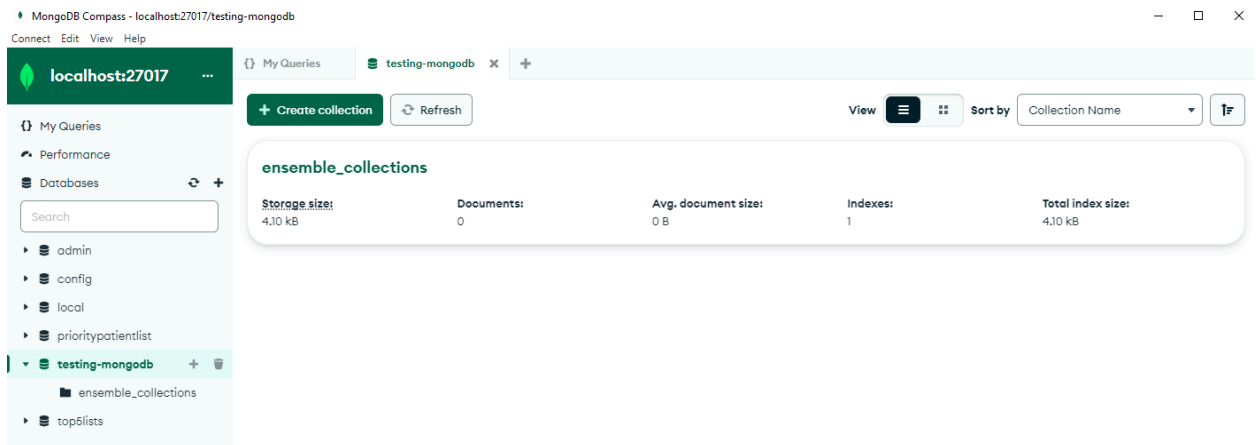
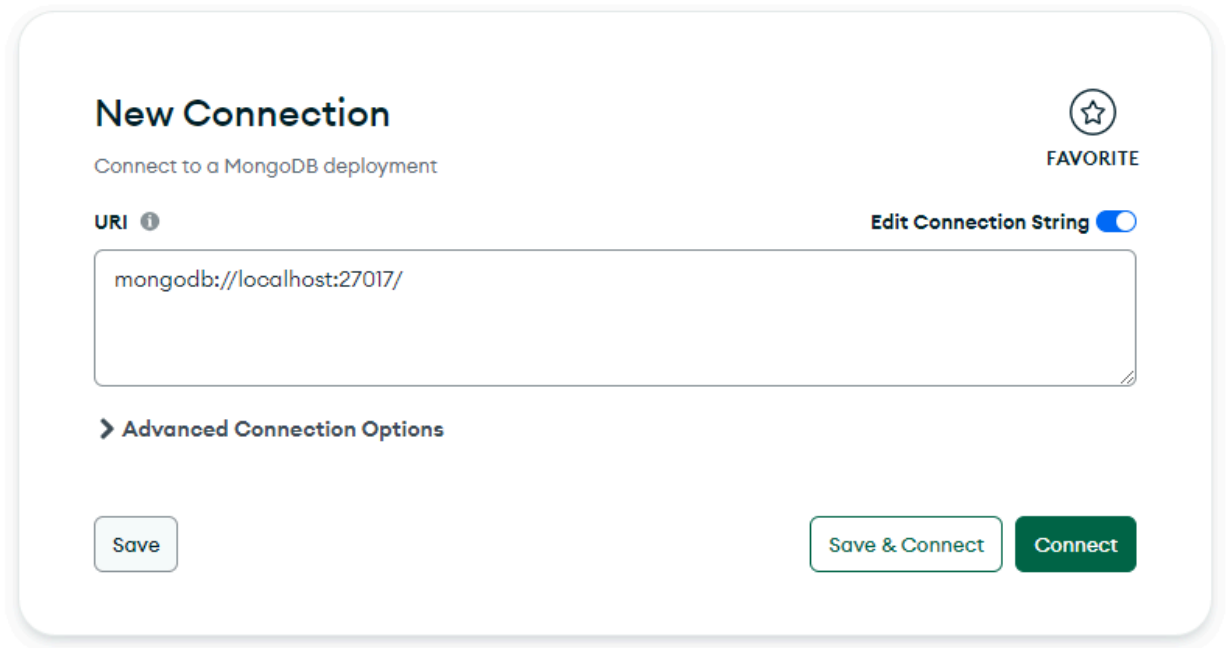
```

PS C:\Users\guill\Desktop\testing-mongo> nodemon mongo.js
[nodemon] 2.0.13
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node mongo.js`
mongodb connected

```

## Adding to the database

1. Open MongoDB Compass and Connect to it.



- MongoDB should now have your testing\_mongodb database defined in step 5 with the collection you named in step 7.
2. The adding precondition is that a piece of data should exist to store inside the insert method. Inserting will be done with the insertMany() method. Multiple insertions possible.

```
collection.insertMany([data]);
```

- **[data]**: the data defined in step 6.

Saving the file restarts MongoDB and adds our file into our collection:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node mongo.js`
mongodb connected
[nodemon] restarting due to changes...
[nodemon] starting `node mongo.js`
mongodb connected
█
```

After refreshing we see 1 new Document added to our ensemble\_collections:

My Queries

testing-mongodb x +

+ Create collection

Refresh

View

Sort by Collection Name

ensemble\_collections

Storage size: 20.48 kB	Documents: 1	Avg. document size: 6.48 kB	Indexes: 1	Total index size: 20.48 kB
---------------------------	-----------------	--------------------------------	---------------	-------------------------------

My Queries

testing-mongodb

ensemble\_collections x +

testing-mongodb.ensemble\_collections

1DOCUMENTS1INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Filter

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

1 - 1 of 1

```
_id: ObjectId('660471f33a16c3e6d36e6981')
eg_histogram: Array (2)
d_districts_histogram: Array (2)
r_districts_histogram: Array (2)
compactness_histogram: Array (2)
split_county_histogram: Array (2)
vote_split_histogram: Array (2)
eq_pop_histogram: Array (2)
dem_responsiveness_histogram: Array (2)
rep_responsiveness_histogram: Array (2)
dem_bias_histogram: Array (2)
rep_bias_histogram: Array (2)
symmetry_histogram: Array (2)
black_min_majority_histogram: Array (2)
latino_min_majority_histogram: Array (2)
asian_min_majority_histogram: Array (2)
democrat_min_majority_histogram: Array (2)
republican_min_majority_histogram: Array (2)
boxWhiskers: Object
__v: 0
```