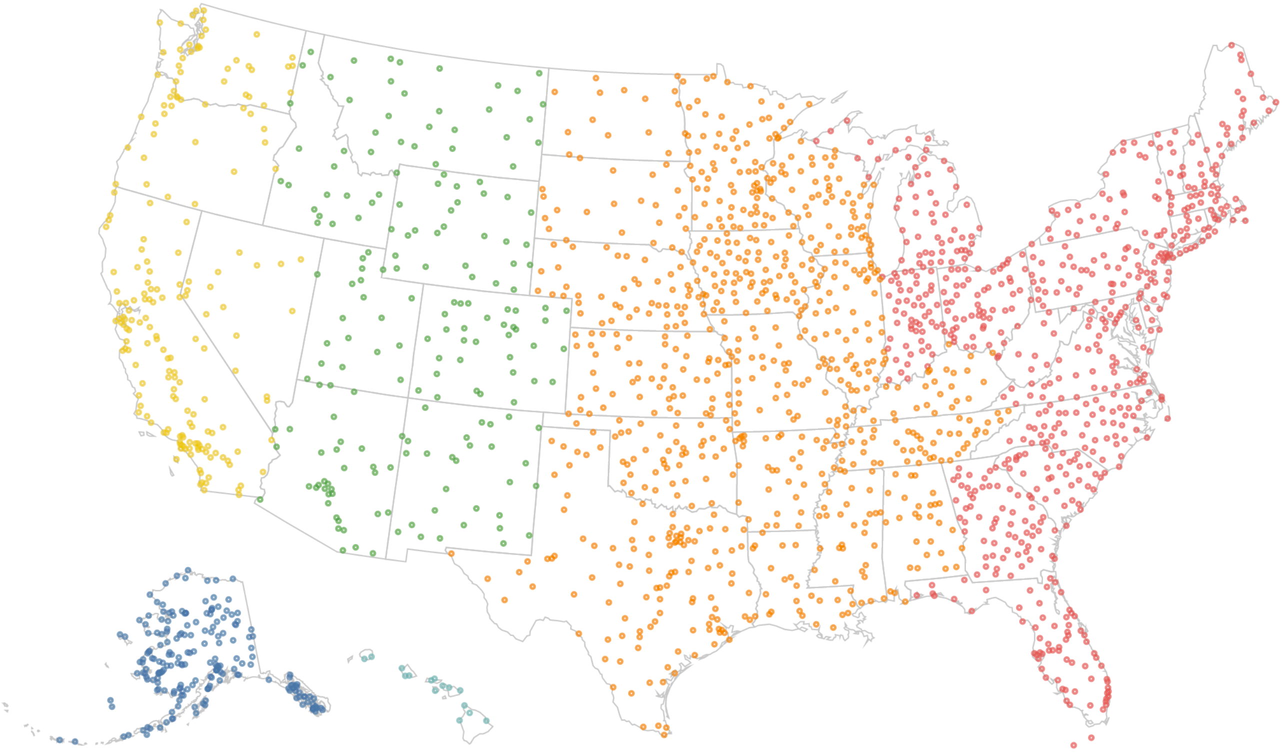
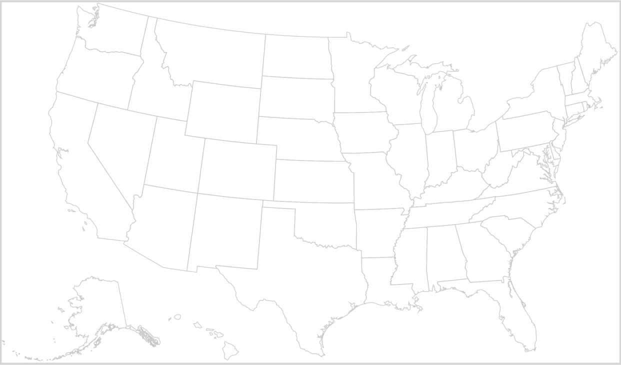
M2ID (2020-2021) - TD s05

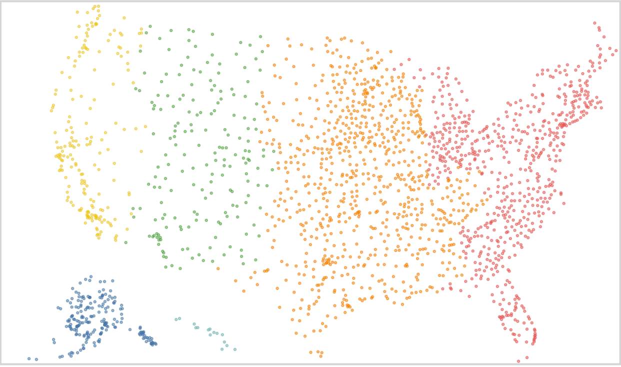
Plotting Airports in the USA

Goal: chart airports in the USA, color-coded by time zone, using Vega-Lite.



The visualization will be composed of two superimposed layers (we already did this for the line plot in TD s#02). https://vega.github.io/vega-lite/docs/layer.html





1. Base Map (bottom layer)

Take inspiration from https://vega.github.io/vega-lite/examples/geo_layer.html to draw the states' borders from GeoJSON file `us-10m.json` using Albers projection.

Adapt the shapes' `fill` and `stroke` colors to match the above illustration.

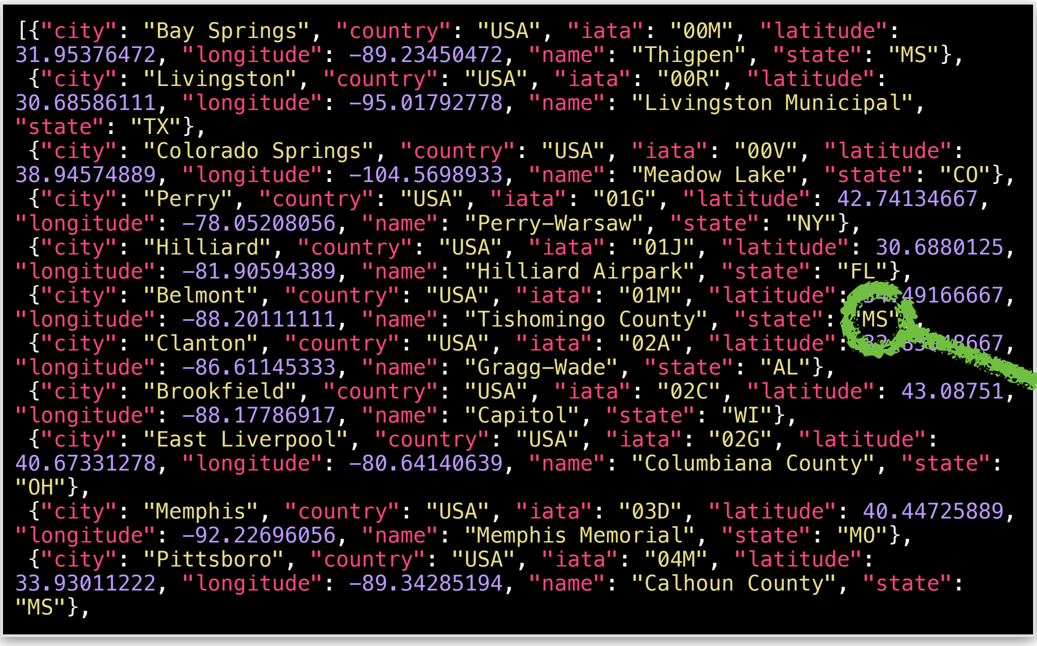
2. Airports (top layer)

The same above-referenced example shows how to plot a second layer. Use `point` marks instead of `circle` marks.

Color code airports based on the time zone of the parent state:

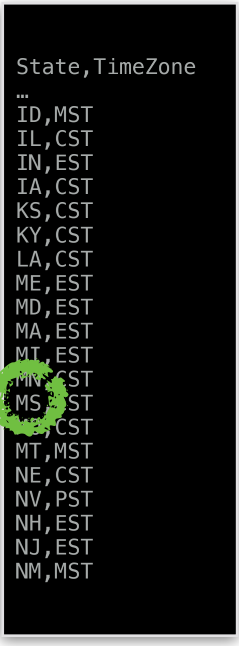
- for each airport, lookup the time zone in `states_tz.csv` and add it as a new attribute of that airport, using a Vega-Lite transform. Take inspiration from example at https://vega.github.io/vega-lite/docs/lookup.html

airports.json



```
[{"city": "Bay Springs", "country": "USA", "iata": "00M", "latitude":
31.95376472, "longitude": -89.23450472, "name": "Thigpen", "state": "MS"},
 {"city": "Livingston", "country": "USA", "iata": "00R", "latitude":
30.68586111, "longitude": -95.01792778, "name": "Livingston Municipal",
"state": "TX"},
 {"city": "Colorado Springs", "country": "USA", "iata": "00V", "latitude":
38.94574889, "longitude": -104.5698933, "name": "Meadow Lake", "state": "CO"},
 {"city": "Perry", "country": "USA", "iata": "01G", "latitude": 42.74134667,
"longitude": -78.05208056, "name": "Perry-Warsaw", "state": "NY"},
 {"city": "Hilliard", "country": "USA", "iata": "01J", "latitude": 30.6880125,
"longitude": -81.90594389, "name": "Hilliard Airpark", "state": "FL"},
 {"city": "Belmont", "country": "USA", "iata": "01M", "latitude": 34.49166667,
"longitude": -88.20111111, "name": "Tishomingo County", "state": "MS"},
 {"city": "Clanton", "country": "USA", "iata": "02A", "latitude": 32.85048667,
"longitude": -86.61145333, "name": "Gragg-Wade", "state": "AL"},
 {"city": "Brookfield", "country": "USA", "iata": "02C", "latitude": 43.08751,
"longitude": -88.17786917, "name": "Capitol", "state": "WI"},
 {"city": "East Liverpool", "country": "USA", "iata": "02G", "latitude":
40.67331278, "longitude": -80.64140639, "name": "Columbiana County", "state":
"OH"},
 {"city": "Memphis", "country": "USA", "iata": "03D", "latitude": 40.44725889,
"longitude": -92.22696056, "name": "Memphis Memorial", "state": "MO"},
 {"city": "Pittsboro", "country": "USA", "iata": "04M", "latitude":
33.93011222, "longitude": -89.34285194, "name": "Calhoun County", "state":
"MS"},
```

states_tz.csv



```
State,TimeZone
…
ID,MST
IL,CST
IN,EST
IA,CST
KS,CST
KY,CST
LA,CST
ME,EST
MD,EST
MA,EST
MI,EST
MN,CST
MS,CST
MO,CST
MT,MST
NE,CST
NV,PST
NH,EST
NJ,EST
NM,MST
```

- then encode that `nominal` attribute using `color` as the encoding channel;

- finally, filter out airports with numbers in their 3-letter IATA code.

Tip: regular expressions `/[0-9]/` or `/\\d/` will return `true` if any of the 3 chars is a number. Create a `filter` which uses the `test(…)` regexp function, accessing the data value with keyword `datum`.
https://vega.github.io/vega-lite/docs/filter.html
https://vega.github.io/vega/docs/expressions/#regexp-functions