@martgnz

22/03/2017

TopoJSON

The new API & more

d3.js Madrid @ CARTO

@martgnz

22/03/2017

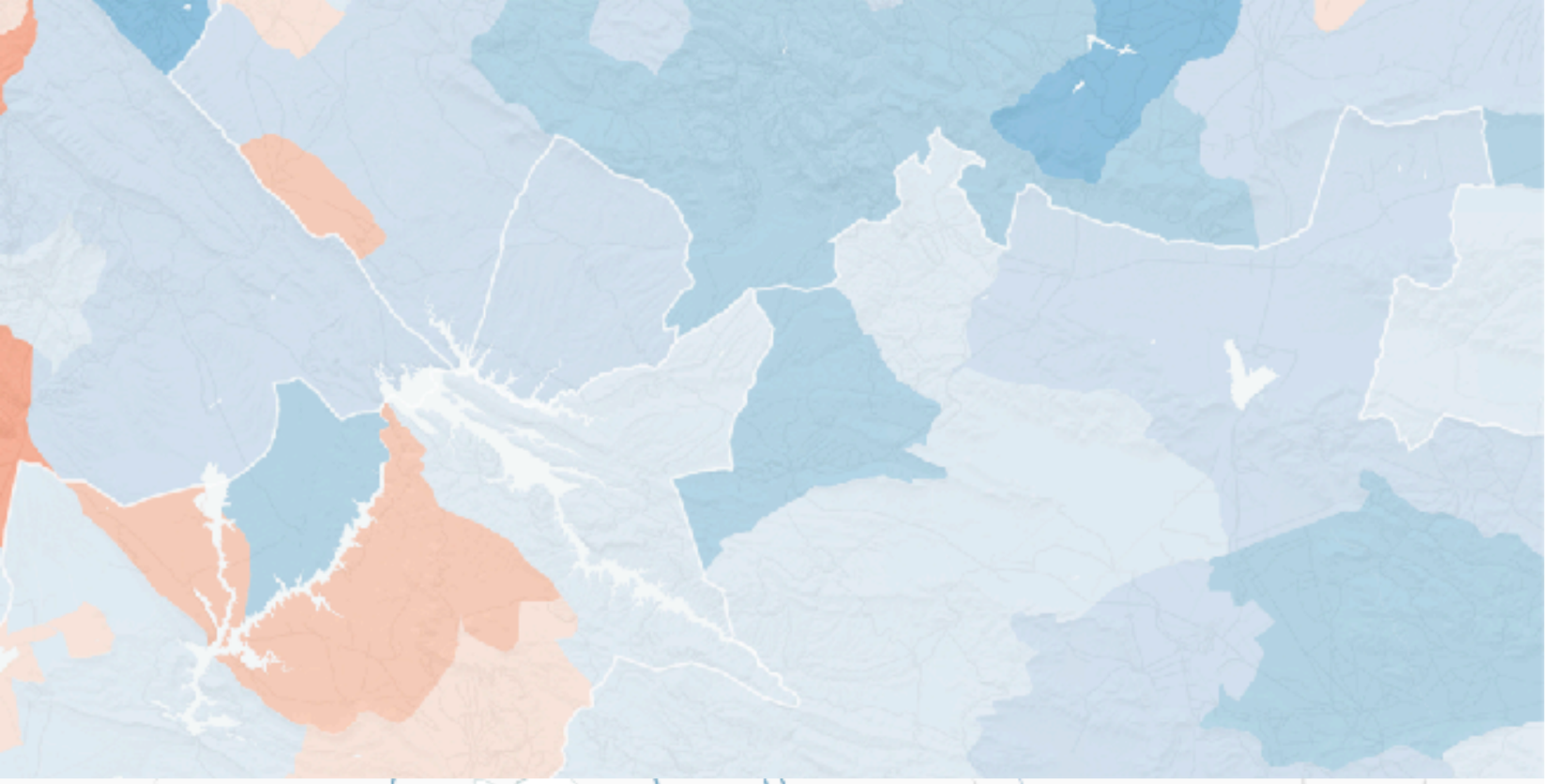
TopoJSON

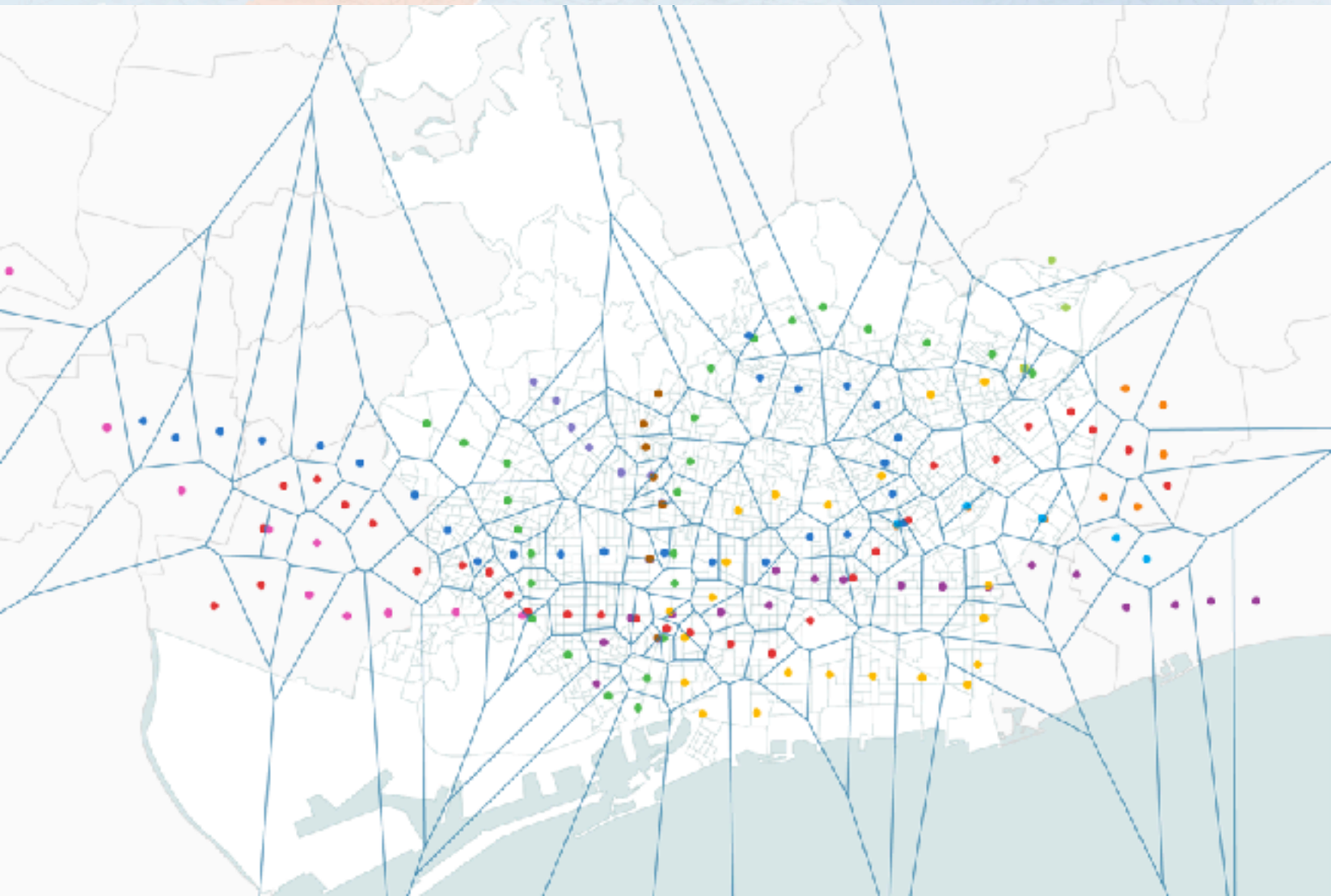
The new API & more

d3.js Madrid @ CARTO

Populate.tools
Politibot

Freelance
El Español





“TopoJSON is an extension of GeoJSON that encodes topology”

– Mike Bostock

before

```
topojson input.shp -o output.json
```

shapefile

topojson

before

```
topojson input.shp -o output.json
```

2.0. #291

Merged **mbostock** merged 7 commits into `master` from `2` on Oct 31, 2016

Conversation 1 · Commits 7 · Files changed 96

mbostock commented on Oct 26, 2016 • edited — Member

Not ready for merge.

- [x] shp2json
- [x] geostitch (d3/d3-geo-projection#72)
- [x] geoproject (d3/d3-geo-projection#72)
- [x] take newline-delimited GeoJSON streams as input with `-n`
- [x] topojson
- [x] topojson-simplify (replaces --simplify, --simplify-proportion)
- [x] post-simplification pruning
- [x] post-simplification filtering
- [x] planar simplification
- [ ] spherical simplification
- [ ] clockwise-forcing? (--force-clockwise)
- [ ] topojson-quantize (replaces --post-quantization)
- [ ] topojson-merge
- [ ] topojson-svg
- [ ] topojson-geojson
- [ ] topojson-transform (replaces --width, --height, --margin, --invert)
- [ ] topojson-group
- [ ] demonstrate how to stitch GeoJSON using geostitch (replaces --stitch-poles)?
- [ ] demonstrate how to project GeoJSON using geoproject (replaces --projection)?
- [ ] demonstrate how to read CSV point geometries (replaces -x, -y)
- [ ] demonstrate how to read CSV properties (replaces --external-properties)
- [ ] demonstrate how to transform properties (replaces --properties, --id-property)

TopoJSON

topojson-client: topo2geo, topomerge, topoquantize
topojson-server: geo2topo
topojson-simplify

Geo utils

d3-geo
geo2svg
geograticule
geoproject
geostitch
ndjson-cli

now

```
geo2topo -n map=< ( shp2json -n input.shp) > output.json
```

GeoJSON to
TopoJSON

now

```
geo2topo -n map=<( shp2json -n input.shp) > output.json
```

GeoJSON to
TopoJSON

now

```
geo2topo -n map=<( shp2json -n input.shp) > output.json
```

accept a newline-
delimited json

GeoJSON to TopoJSON

our final object

now

```
geo2topo -n map=<( shp2json -n input.shp) > output.json
```

accept a newline-delimited json

GeoJSON to TopoJSON

our final object

now

```
geo2topo -n map=<( shp2json -n input.shp) > output.json
```

accept a newline-delimited json

shapefile to GeoJSON

GeoJSON to TopoJSON

our final object

write the TopoJSON

now

```
geo2topo -n map=<( shp2json -n input.shp) > output.json
```

accept a newline-delimited json

shapefile to GeoJSON

mix two shapefiles without merging

```
geo2topo -n provinces=<( \
  shp2json --encoding utf8 -n build/recintos_provinciales_inspire_peninbal_etrs89.shp \
  | ndjson-map '(d.id = d.properties.NATCODE.slice(4, 6), d)'
  shp2json --encoding utf8 -n build/recintos_provinciales_inspire_canarias_wgs84.shp \
  | ndjson-map '(d.id = d.properties.NATCODE.slice(4, 6), d)') \
> es/provinces.json
```

mix two shapefiles without merging

```
geo2topo -n provinces=<( \
  shp2json --encoding utf8 -n build/recintos_provinciales_inspire_peninbal_etrs89.shp \
  | ndjson-map '(d.id = d.properties.NATCODE.slice(4, 6), d)'
  shp2json --encoding utf8 -n build/recintos_provinciales_inspire_canarias_wgs84.shp \
  | ndjson-map '(d.id = d.properties.NATCODE.slice(4, 6), d)') \
> es/provinces.json
```

yep, we can use JavaScript inside!

seamless merging

```
geo2topo -n provinces=<( ... ) \
  | topomerge autonomous_regions=provinces -k 'd.properties.NATCODE.slice(2, 4)' \
| topomerge nation=autonomous_regions \
> es/provinces.json
```

seamless merging

```
geo2topo -n provinces=<( ... ) \
  | topomerge autonomous_regions=provinces -k 'd.properties.NATCODE.slice(2, 4)' \
| topomerge nation=autonomous_regions \
> es/provinces.json
```

the first two numbers
in each province ID equal
the autonomous regions

preprojection

just paint!

preprojection

just paint!

preprojection with custom projections

composite-projections

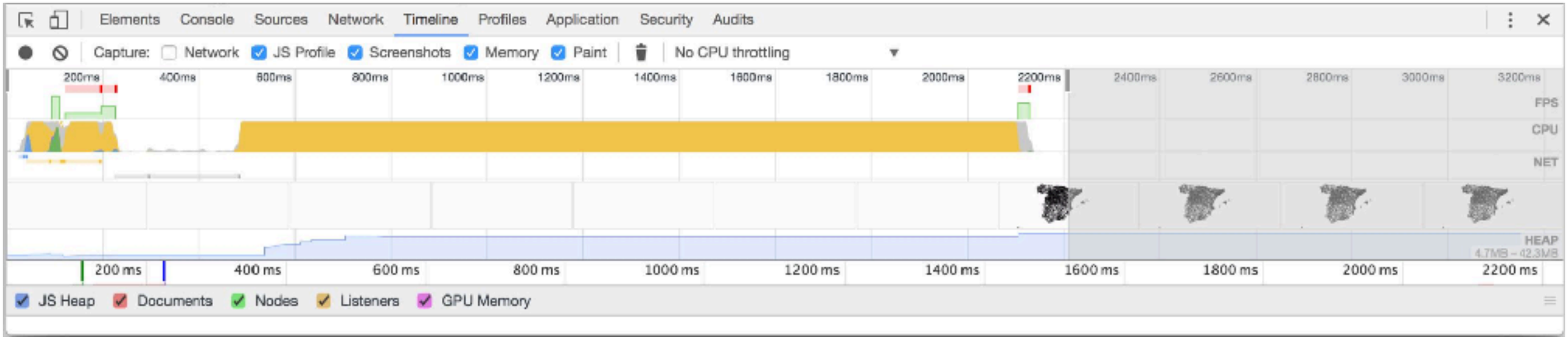
d3-geo-projection fork

custom preprojection

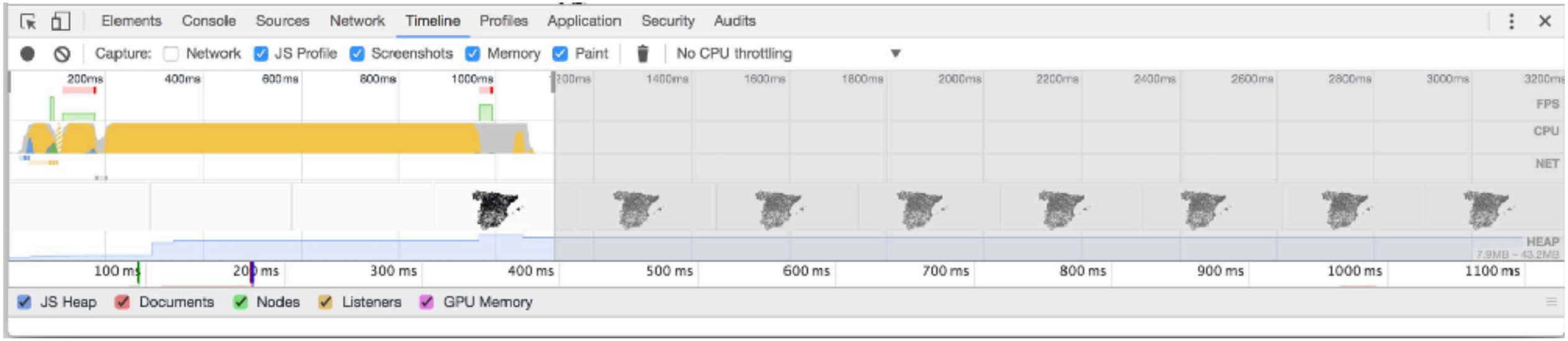
```
geo2topo -n \
  municipalities=<( \
      cat es/_municipalities.json \
        | geoproject –n 'd3.geoConicConformalSpain().translate([480, 300])' \
 > es/municipalities.json
```

that’s it

Without preprojection: first paint is around 2200ms



Awww yeah, cut in a half!



us-atlas

world-atlas

mad-atlas

bcn-atlas

es-atlas

reference

Command line cartography (I, II, III, & IV)
California Population Density II

thanks!