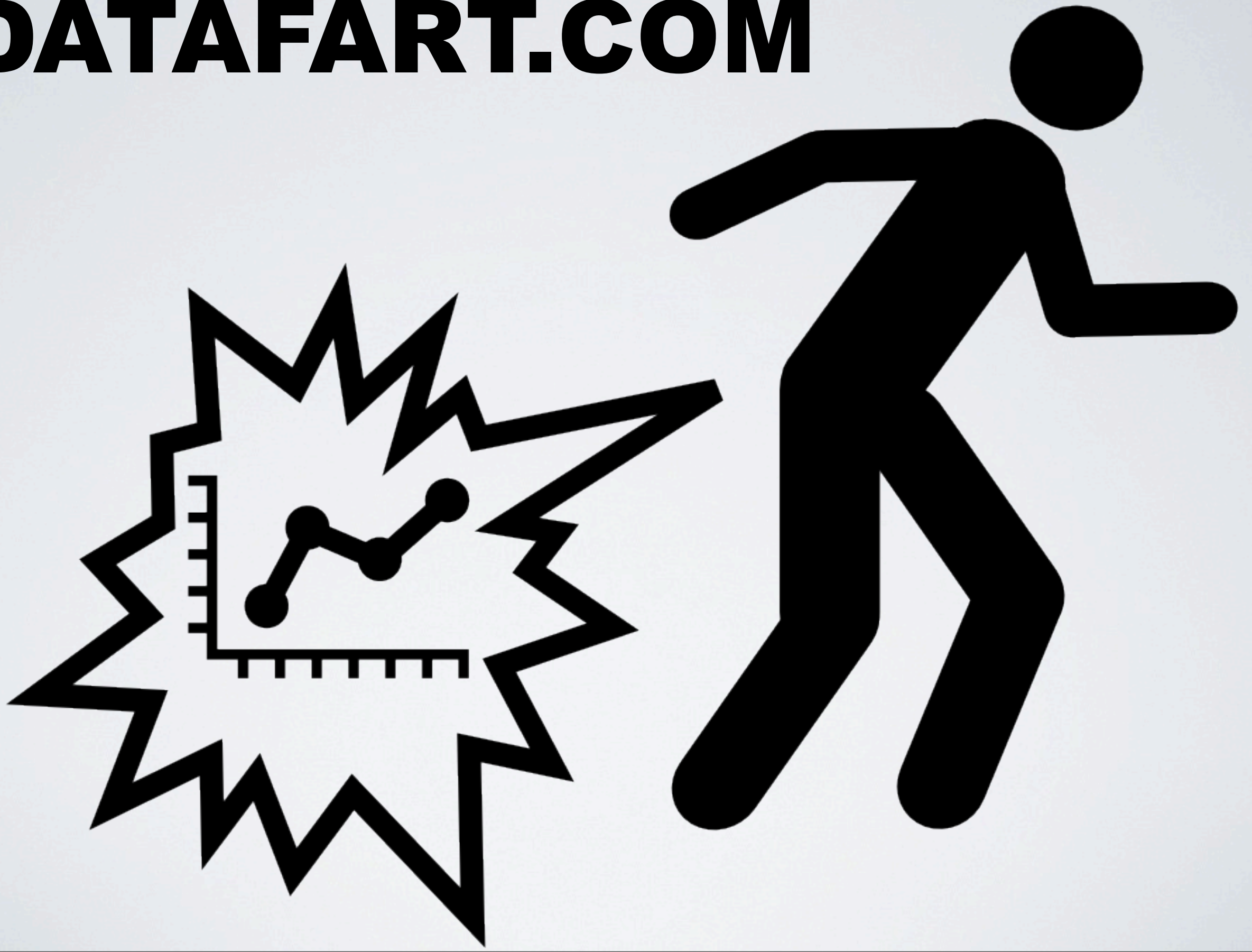
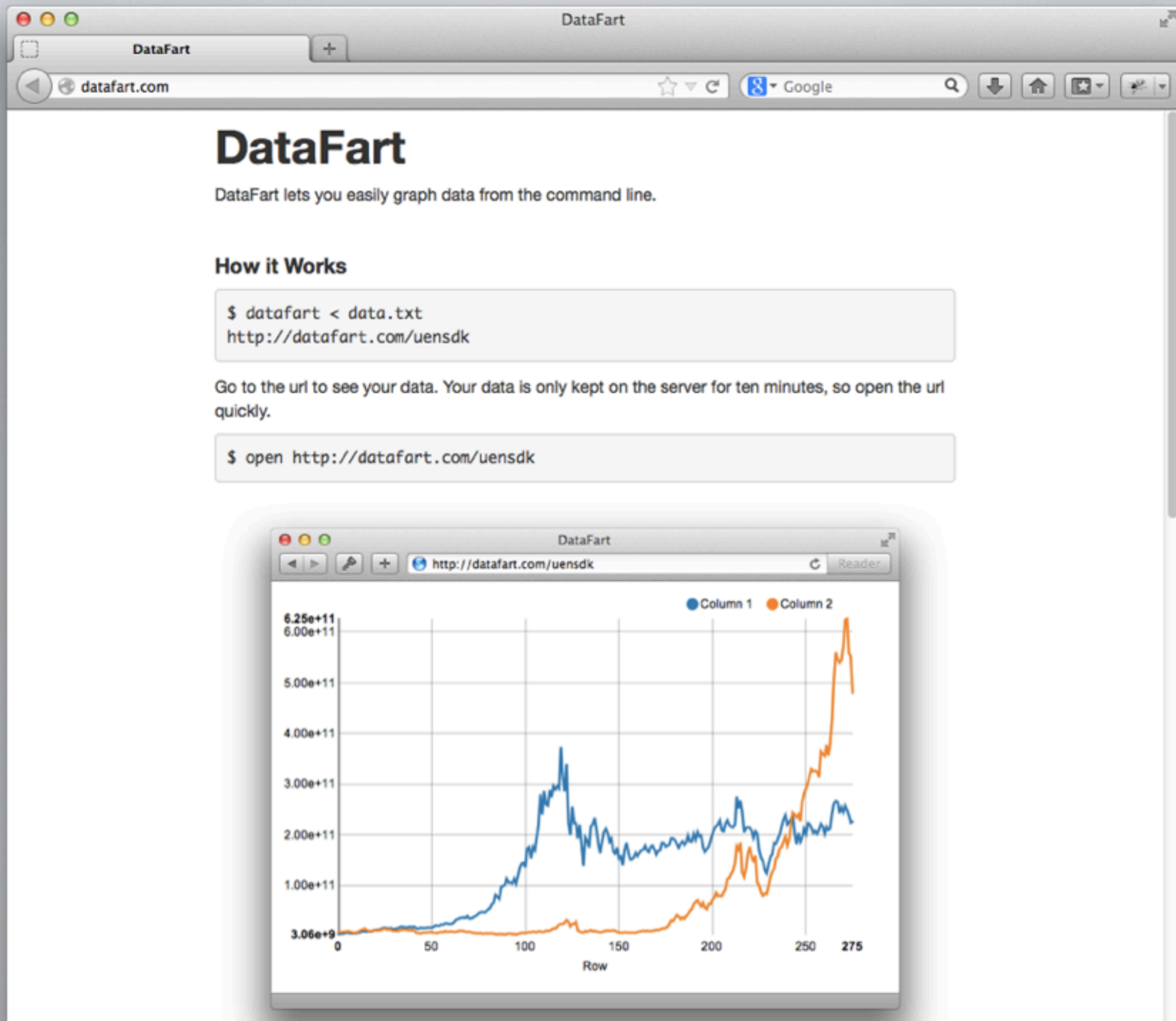


MAPFART.COM

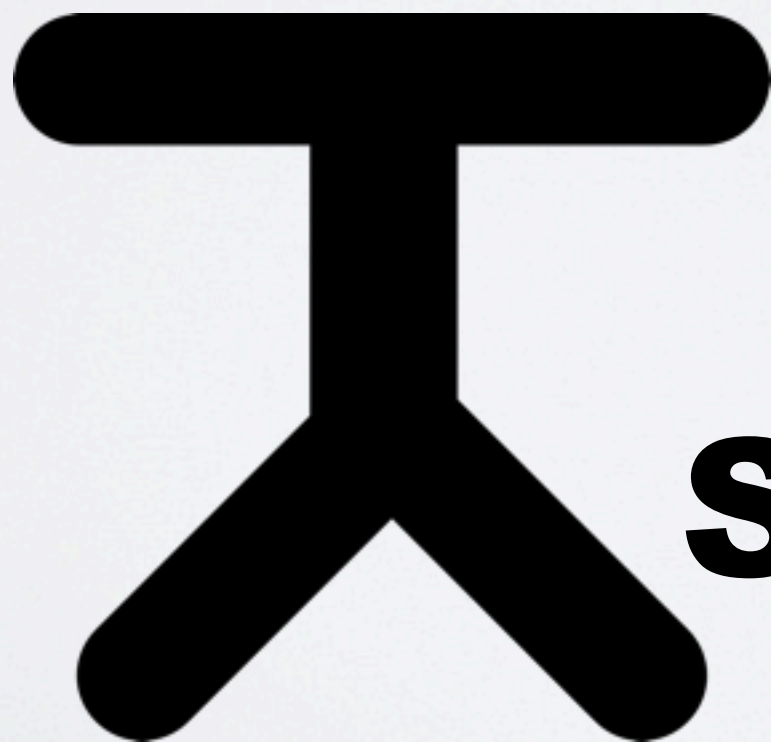
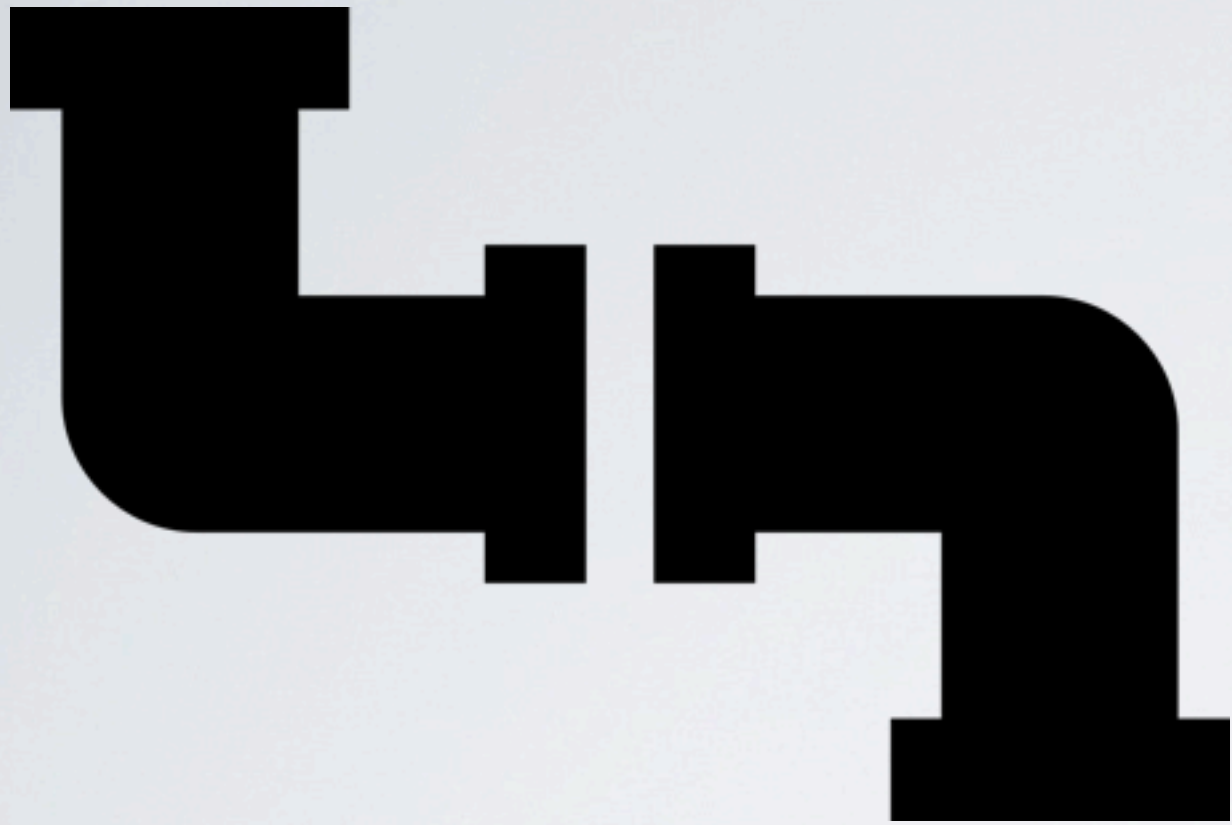


DATAFART.COM





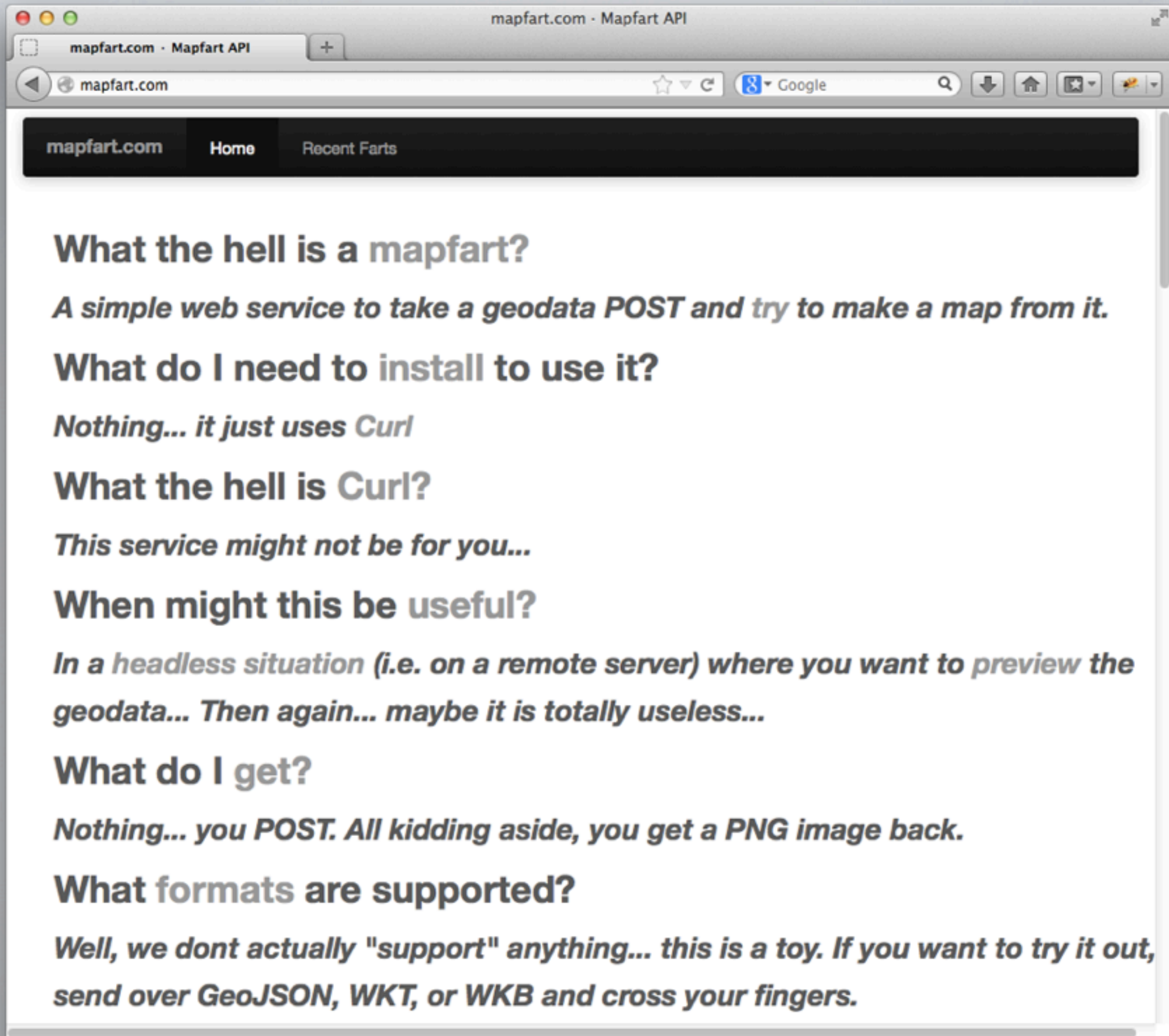
PIPES ARE COOL



**HEADLESS
SITUATIONS**

MAPFART.COM





What the hell is a mapfart?

A simple web service to take a geodata POST and try to make a map from it.

What do I need to install to use it?

Nothing... it just uses Curl

What the hell is Curl?

This service might not be for you...

When might this be useful?

In a headless situation (i.e. on a remote server) where you want to preview the geodata... Then again... maybe it is totally useless...

What do I get?

Nothing... you POST. All kidding aside, you get a PNG image back.

What formats are supported?

Well, we dont actually "support" anything... this is a toy. If you want to try it out, send over GeoJSON, WKT, or WKB and cross your fingers.

Simple

```
alias mapfart='curl -H "Content-Type: application/octet-stream" --  
data-binary @- mapfart.com/api/fart'
```

Crunchy

(change output projection)

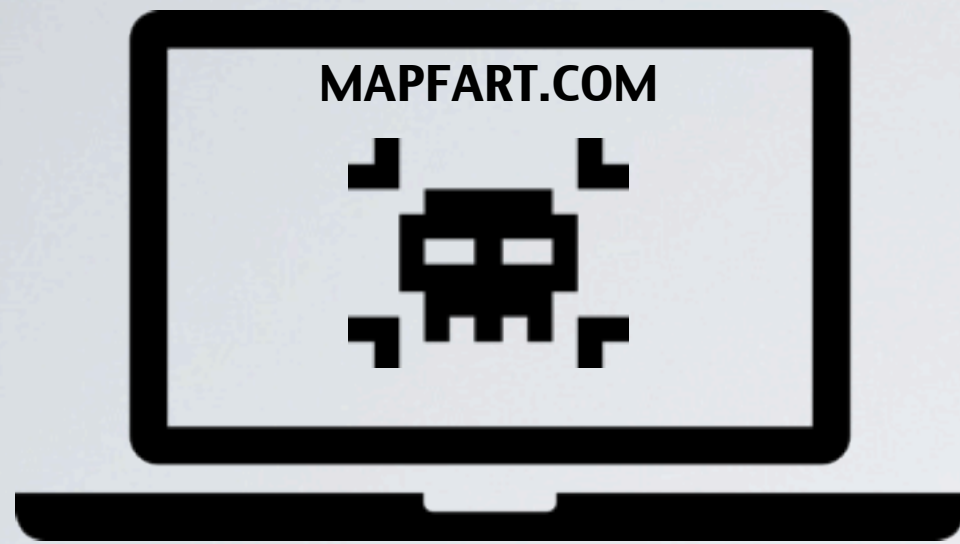
```
alias mapfart='curl -H "Content-Type: application/octet-stream" --  
data-binary @- mapfart.com/api/3857/fart'
```

Hipster

(change output projection AND mapsize)

```
alias mapfart='curl -H "Content-Type: application/octet-stream" --  
data-binary @- mapfart.com/api/3857/2000/2000/fart'
```

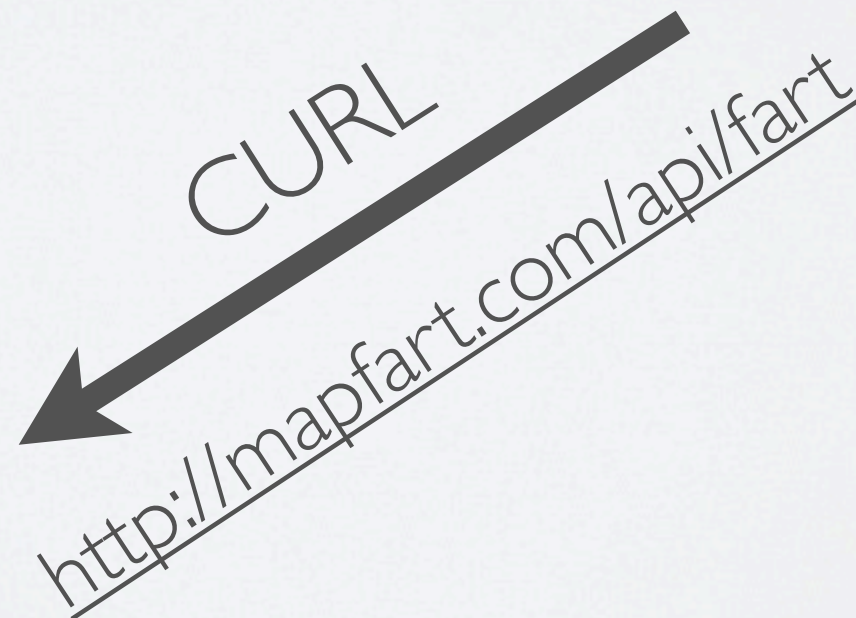
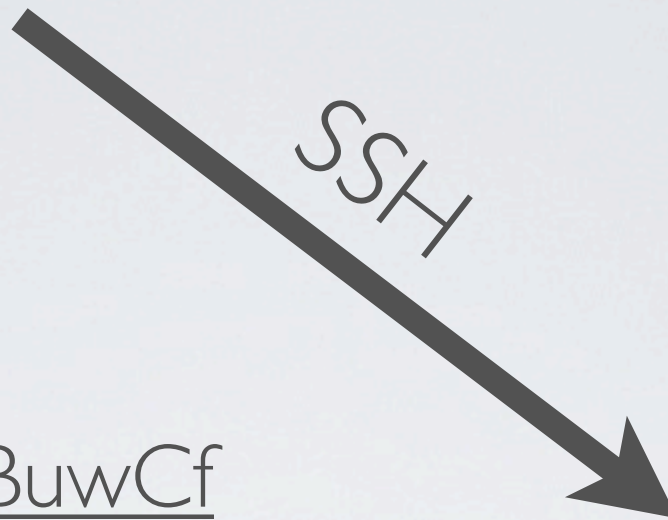
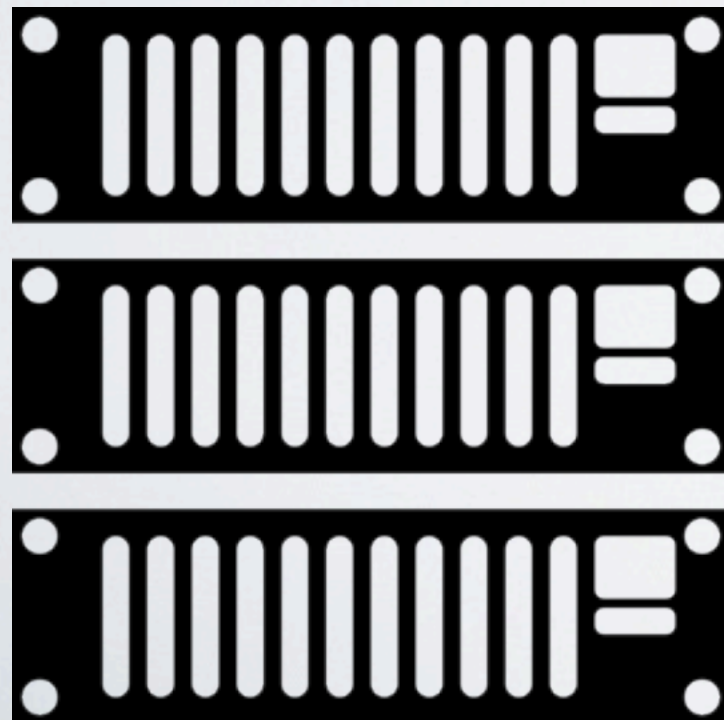

THE SHIZZLE



http://mapfart.com/fart_4BuwCf



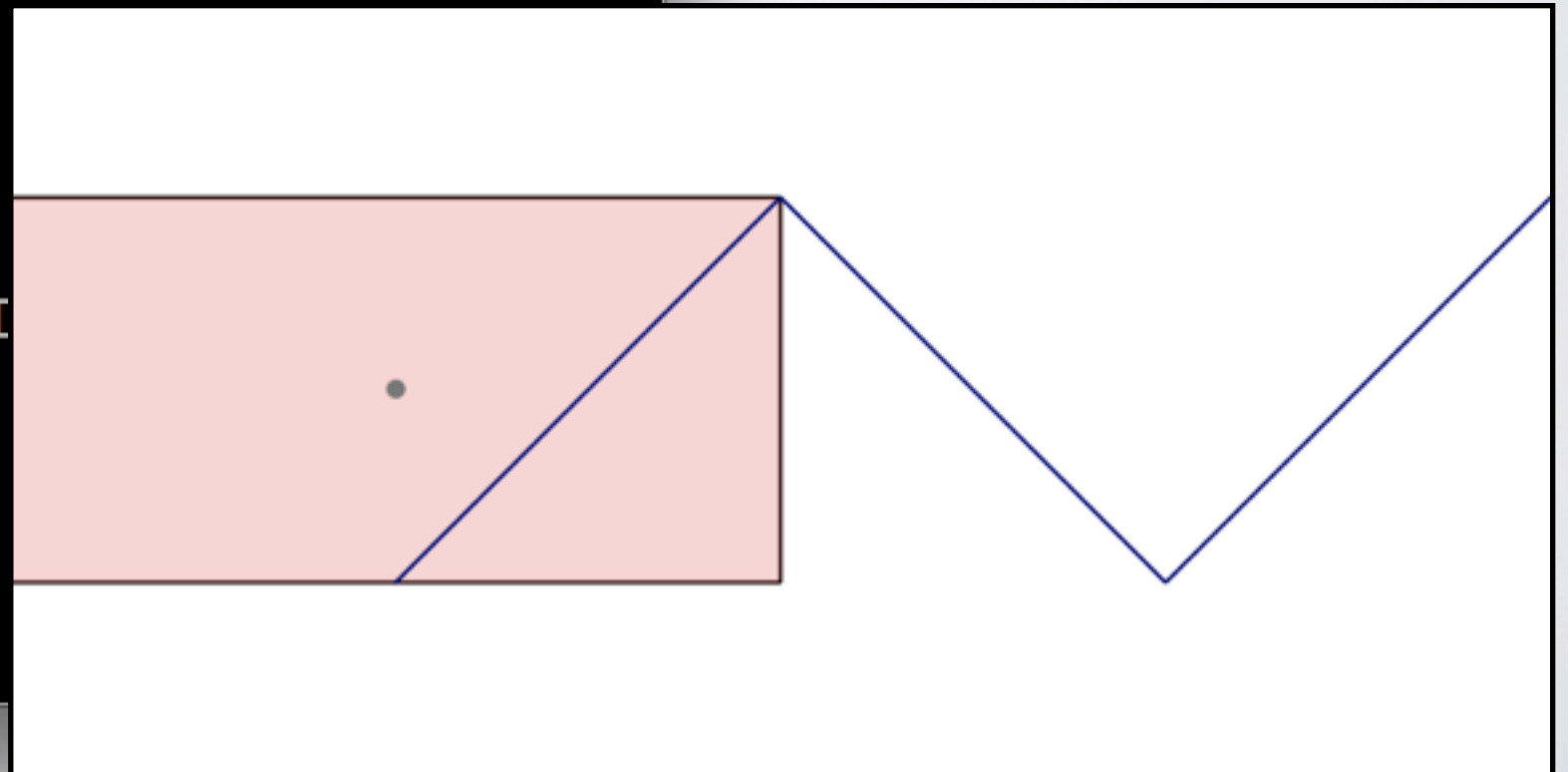
MAPFART.COM



Simple GeoJSON

z-air: aaronr\$ **cat** simple.json | **mapfart**

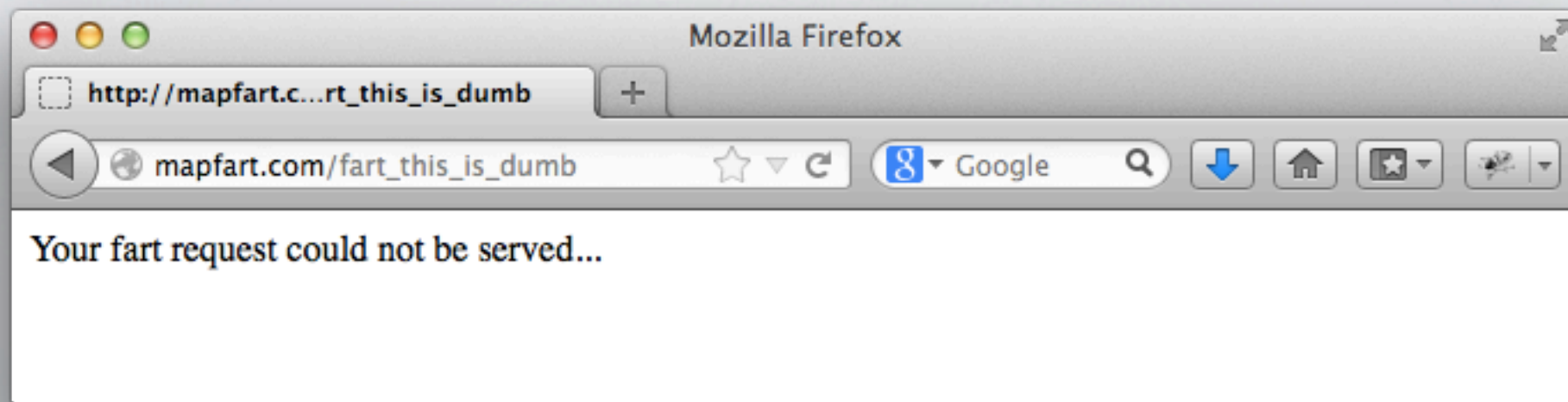
```
5. aaronr@linode: ~ (bash)
z-air:~ aaronr$ cat simple.json
{ "type": "FeatureCollection",
  "features": [
    { "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [102.0, 0.5] }
    },
    { "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
        ]
      }
    },
    { "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [ [101.0, 0.0], [103.0, 0.0], [
            [101.0, 1.0], [101.0, 0.0] ]
          ]
        ]
      }
    }
  ]
}
```



http://mapfart.com/fart_g69fAT

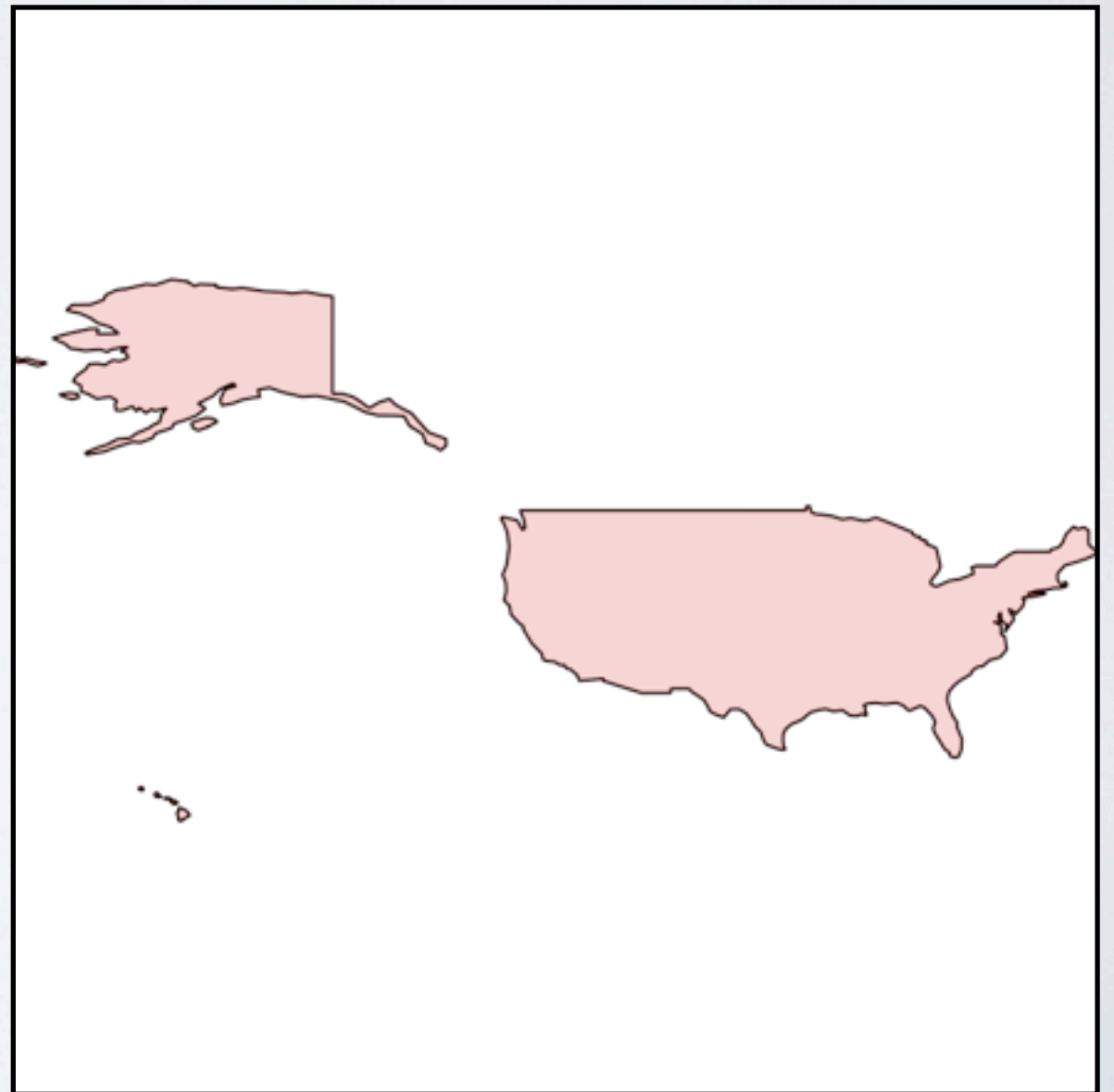
Robust Error Handling

```
z-air:~ aaronr$ echo 'foss4g-na' | mapfart  
your farts stink... try again!
```



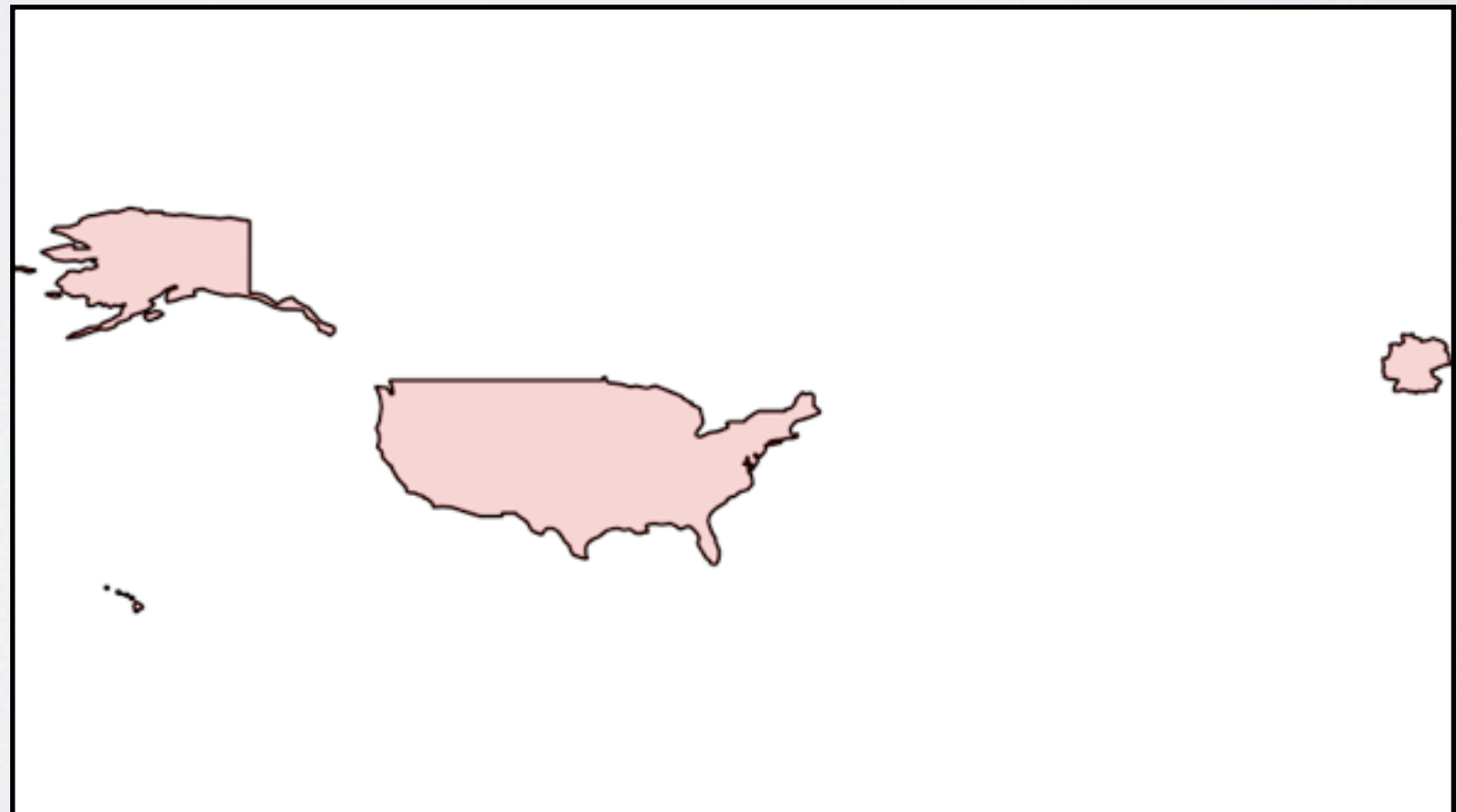
Local Shapefile via ogr2ogr

```
z-air:~ aaronr$ ogr2ogr -f GeoJSON /vsistdout/ \  
> ne_110m_admin_0_countries.shp \  
> -sql "select * from ne_110m_admin_0_countries \  
> where name='United States'" \  
> | mapfart
```



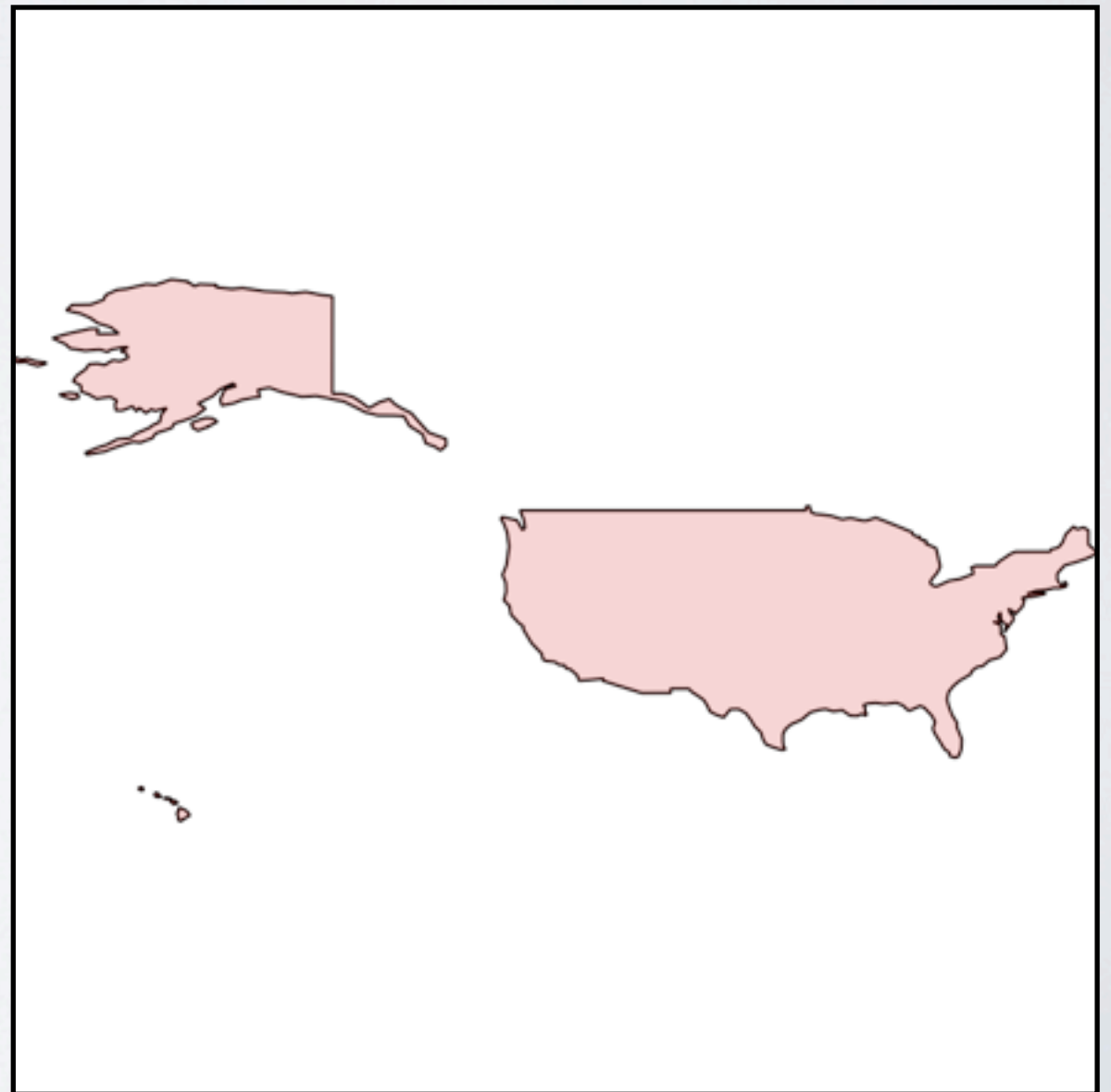
Remote Shapefile via ogr2ogr

```
z-air:~ aaronr$ ogr2ogr -f GeoJSON /vsistdout/ \  
>/vsizip/vsicurl/http://www.naturalearthdata.com/  
http://www.naturalearthdata.com/download/110m/  
cultural/ne_110m_admin_0_countries.zip/  
ne_110m_admin_0_countries.shp \  
>-sql "select * from ne_110m_admin_0_countries  
where name='United States' or name='Germany'" \  
>| mapfart
```



PostGIS via ogr2ogr

```
z-air:~ aaronr$ ogr2ogr -f "GeoJSON" /vsistdout/ \  
>PG:dbname=test ne_admin_bounds -where "name = \  
'United States'" \  
>| mapfart
```

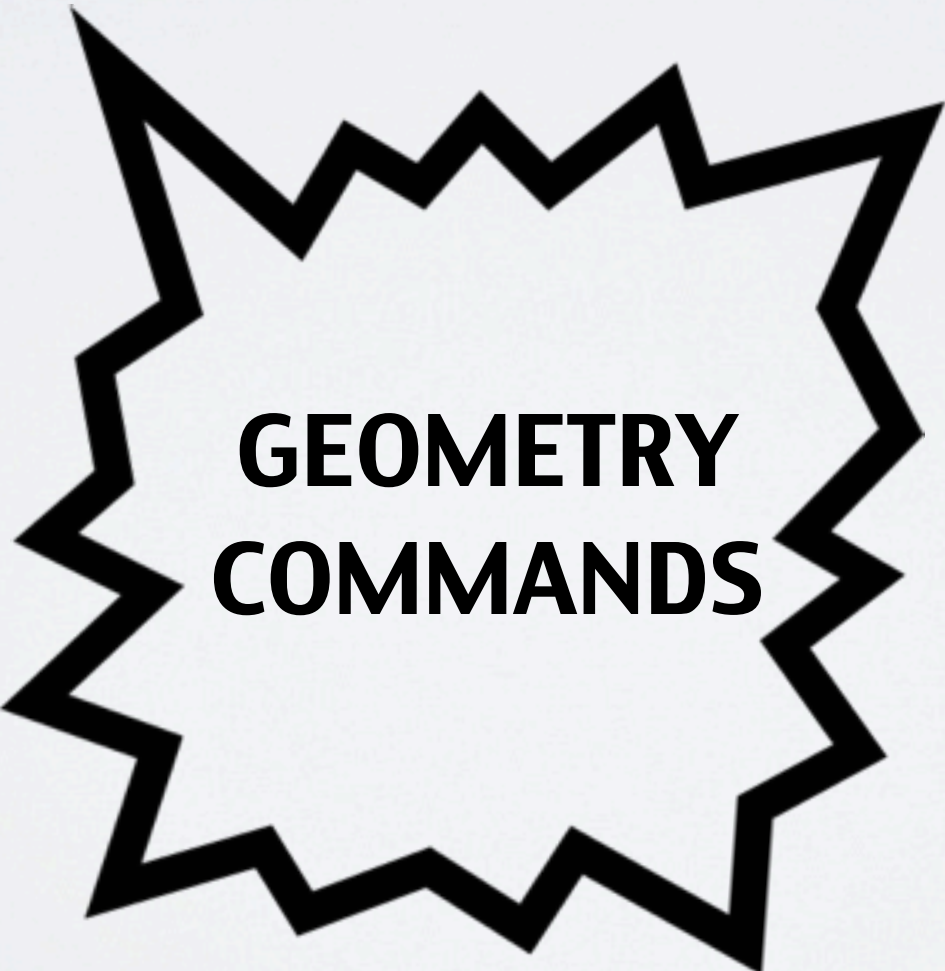


GEOJSON



MAPSERVER

WKT/WKB

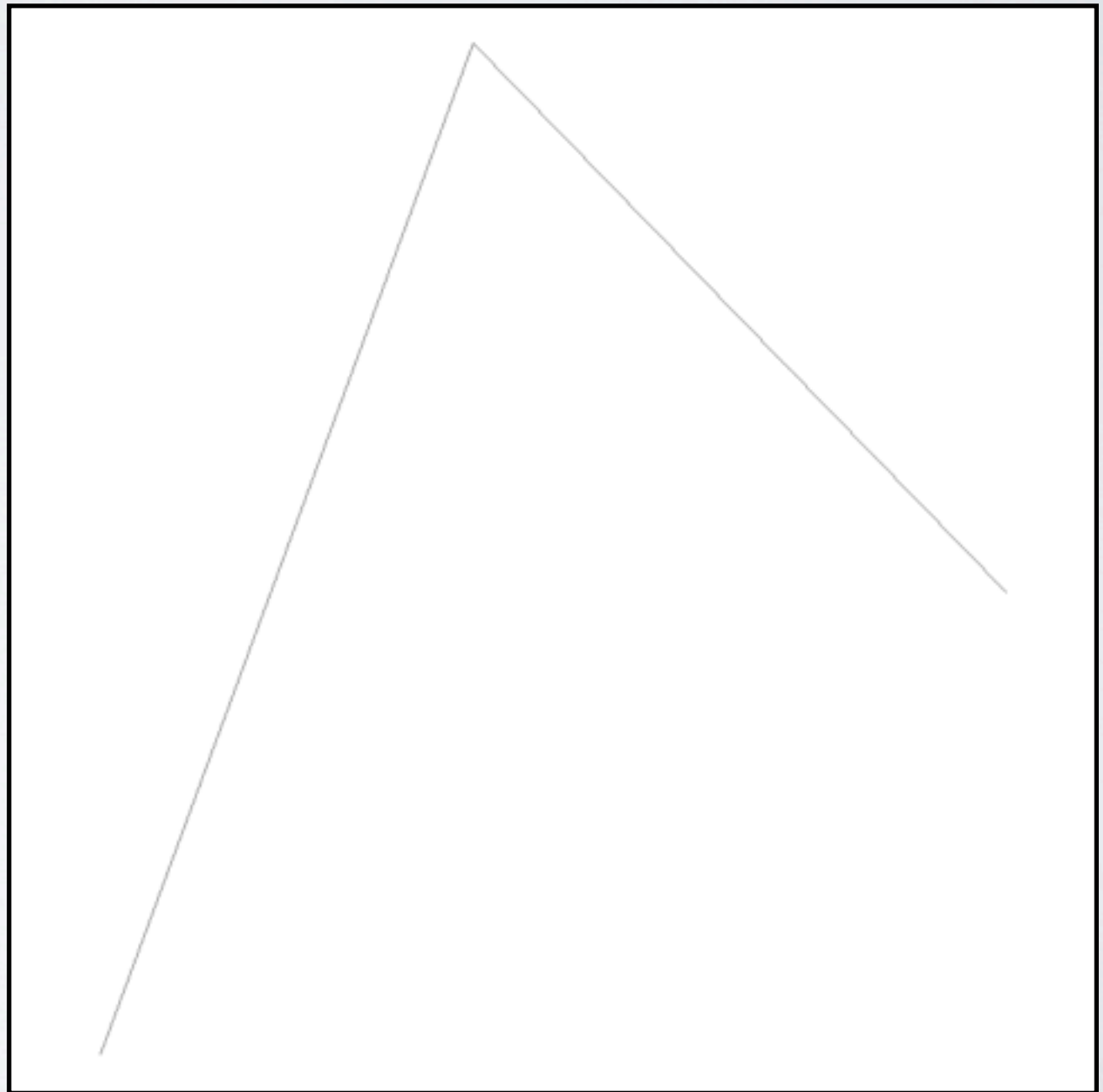


GEOMETRY
COMMANDS

<http://jericks.github.io/geometrycommands/index.html>

WKT

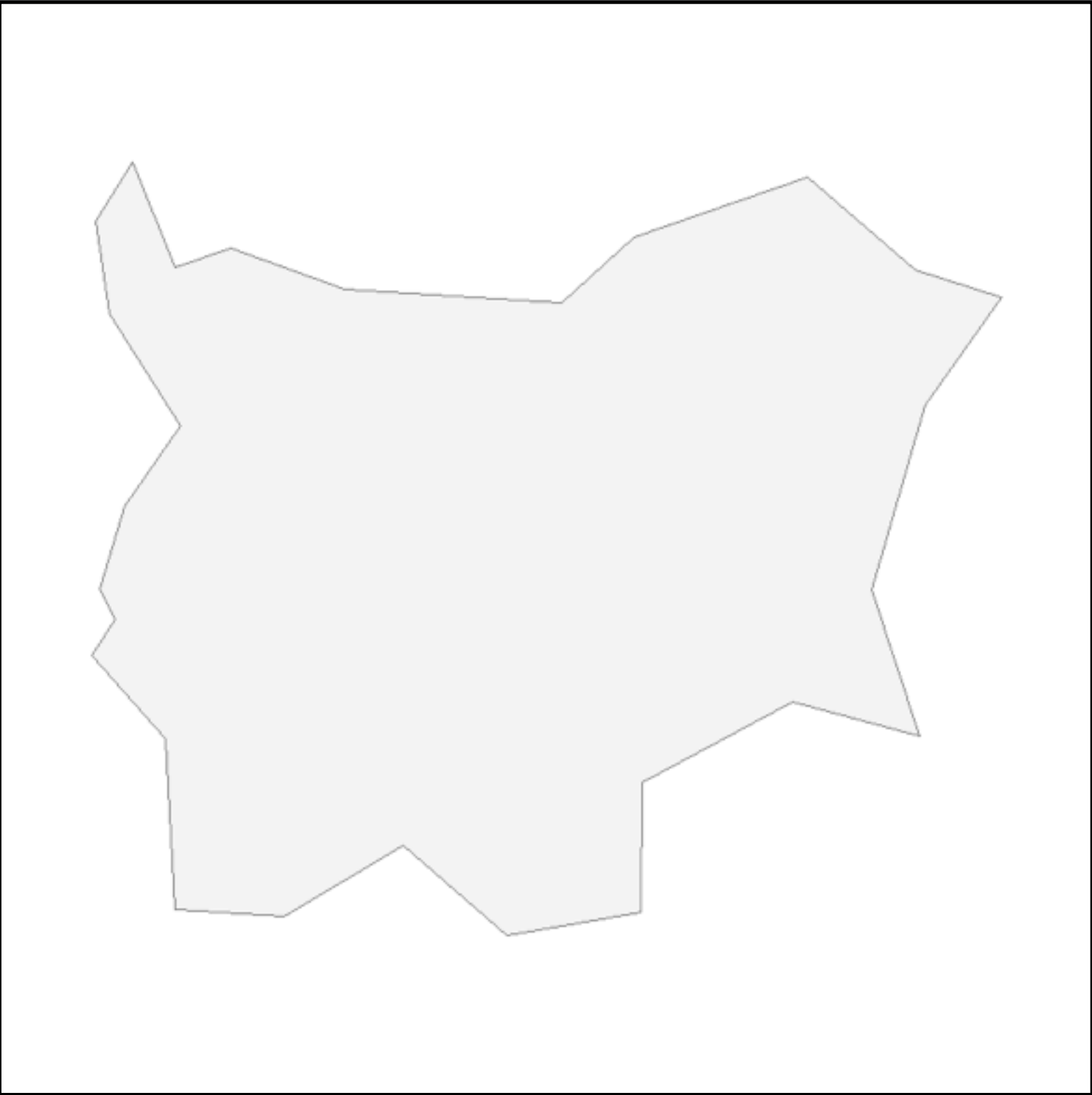
```
z-air:~ aaronr$ echo 'LINESTRING(3 4,10 50,20 25)' \  
>| mapfart
```



WKB

z-air:~ aaronr\$ **echo**

'0106000020E61000000010000000010300000000100000001C000000078CA55F63AA8364
08E75F6F4111E4640E8C41B89E0F1364016B200CC71E94540409A22C311553740805
90E40D1F2454028C3E11BC61938407C112DC5DADE45408044F6C9BB9139405A74F6F
41ED84540F0ED03
64640694182EF58
0E874B522020A3C
3A41129FF3B406E
4F61D3A401CB19D
23940D6E90AA303
03B57ABF68FA744
060F8E8FF444007
90D3B4540D0E43D
34540E0577BB56E
0400AFD03136936
>| **mapfart**

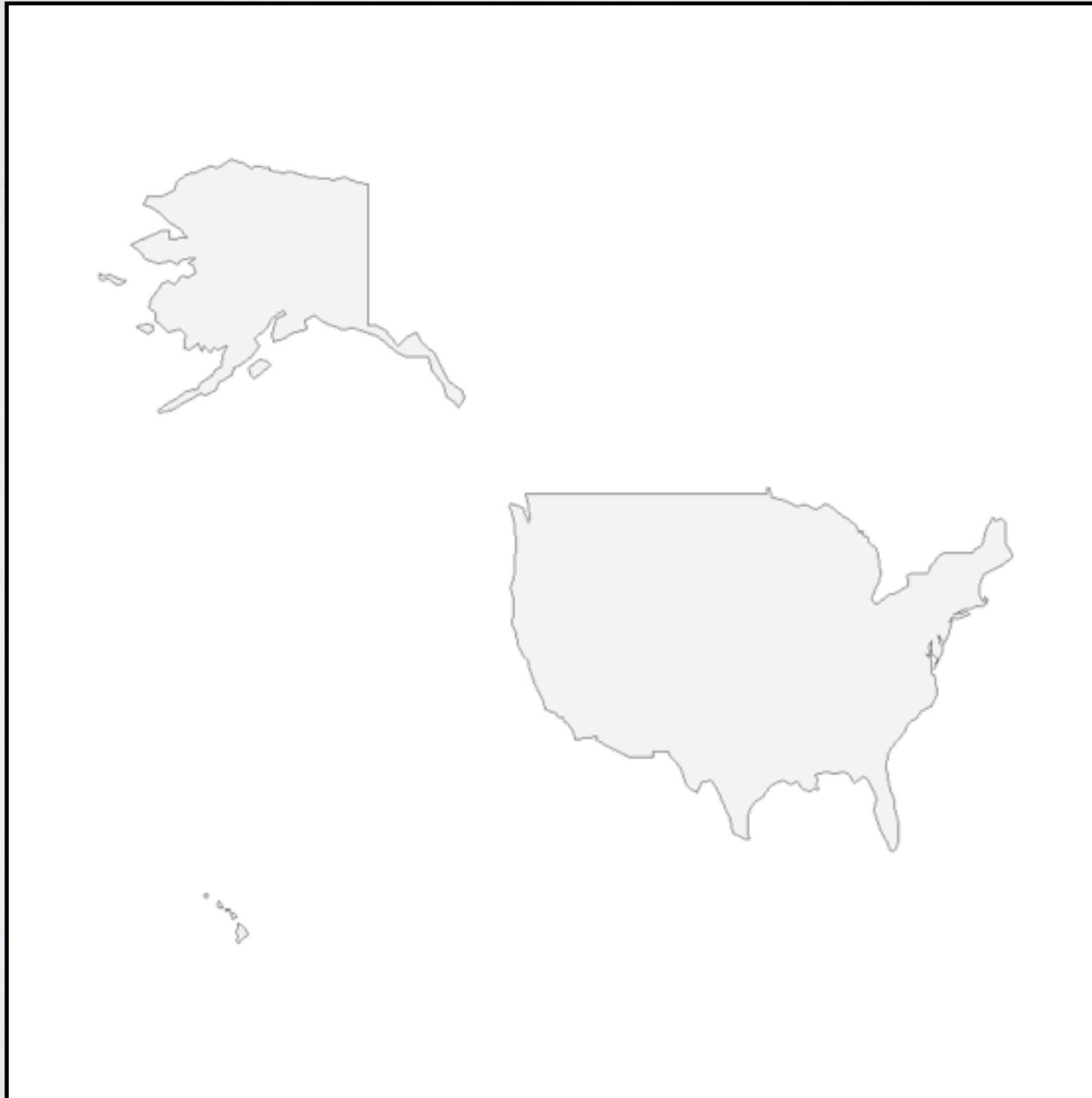


407BE9D0B5861
66821A8EDA454
60F8494540A96
12454098F1A07
40E042F6C97B3
FB4EBC2BB1374
B5A1E13640DA4
8B3640E81804E
40EC8DDEA9027
7059BE47D2454
F4111E4640' \

Bulgaria

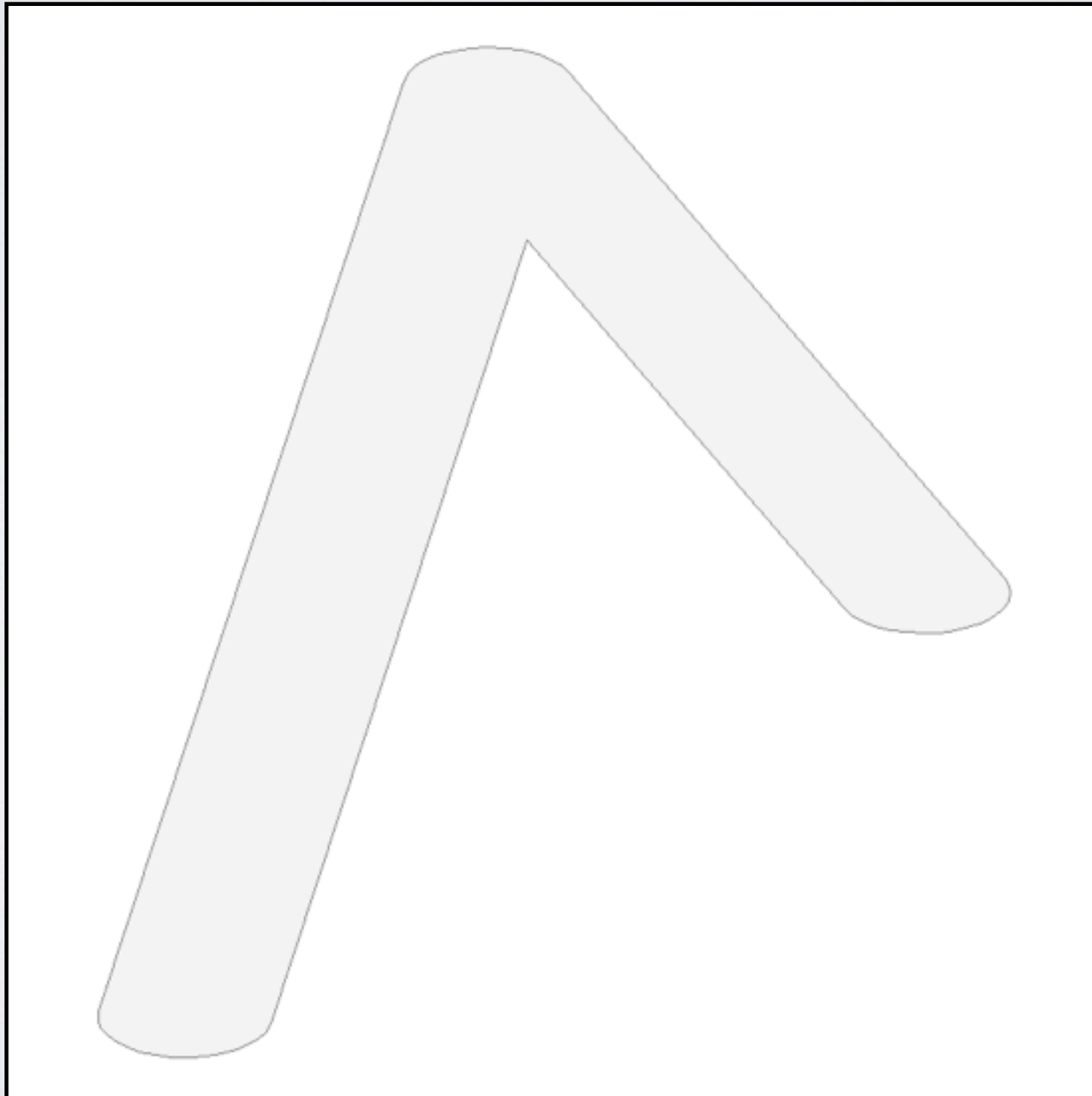
WKB (single shape) via psql

```
z-air:~ aaronr$ psql -A -t -c "select geom from ne_admin_bounds  
where name = 'United States'" testdb | mapfart
```

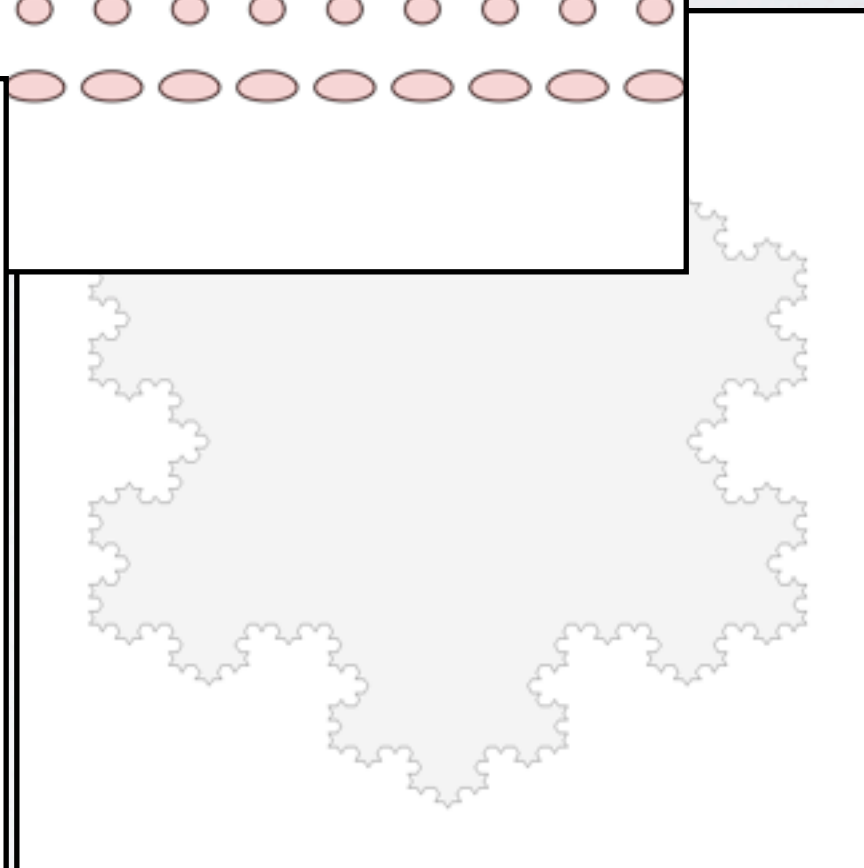
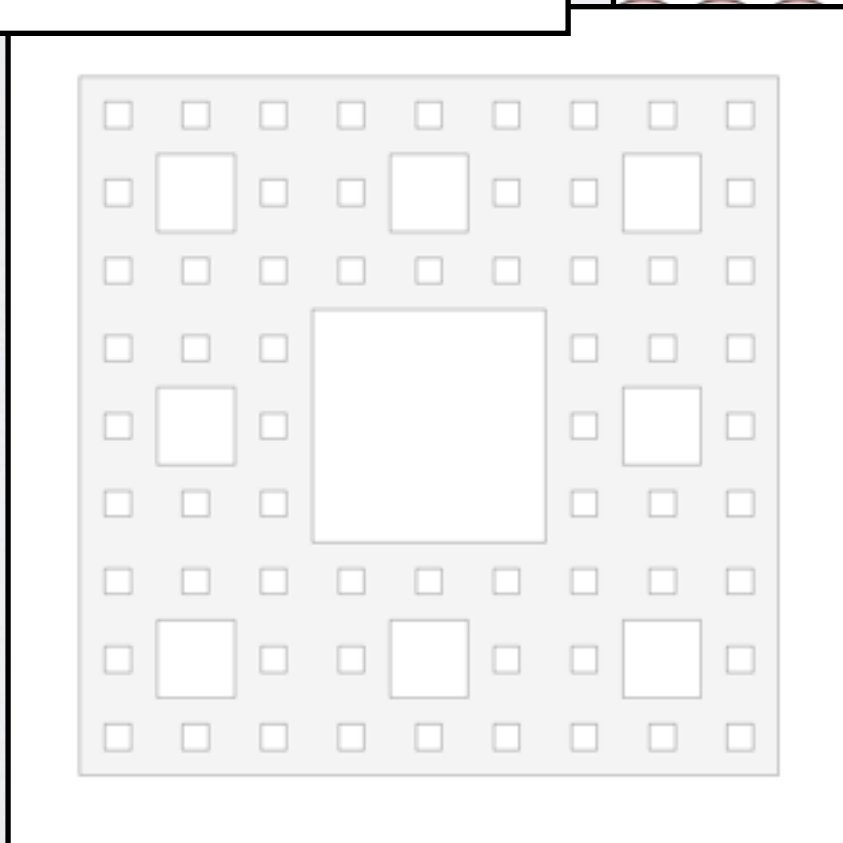
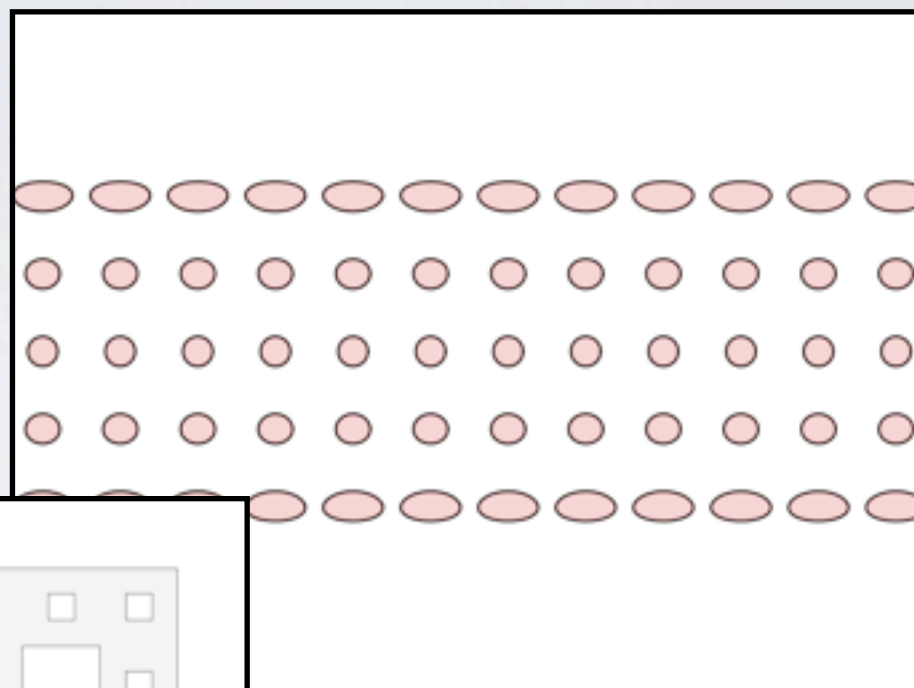


WKB via GeometryCommands

```
z-air:~ aaronr$ geom buffer -g 'LINESTRING(3 4,10 50,20 25)' -d 2 \  
>| mapfart
```



Cool



@REPROJECTED



Z-PULLEY



CUGOS