https://www.setlist.fm/search?query=scorpions (https://www.setlist.fm/search?query=scorpions)

In [2]:

```
from robobrowser import RoboBrowser
import geocoder
import folium
import pandas as pd
import dateparser
from joblib import Memory
import difflib
import pycountry
from iso3166 import countries
memory = Memory(cachedir='tmp', verbose=0)
```

In [4]:
```
1 url = 'https://www.setlist.fm/search?query=scorpions'
2 browser = RoboBrowser(history=True, parser='html5lib')
3 browser.open(url)
4 concert = browser.select('.setlistPreview')[0]
5 concert
```

Out[4]:
```
<div class="col-xs-12 setlistPreview">
<div>
<div class="condensed dateBlock">
<span class="month">Nov</span>
<span class="day">1</span>
<span class="year">2017</span>

</div>
</div>
<div>
<h2><a href="setlist/scorpions/2017/olympijskiy-stadium-moscow-russia-23e36c7b.html" title="View this
 Scorpions setlist">Scorpions at Olympijskiy Stadium, Moscow, Russia</a></h2>
<div class="details">
<span>Artist: <strong><a href="setlists/scorpions-3d63ddf.html" title="More Scorpions setlists"><span>
Scorpions</span></a></strong></span>,

<span>
Tour:
<strong><a href="search?artist=3d63ddf&amp;query=tour:%28Crazy+World+2017+Tour%29" rel="nofollow" titl
```

In [5]:
```
1 day = concert.select('.day')[0].text
2 month = concert.select('.month')[0].text
3 year = concert.select('.year')[0].text
4 desc = concert.select('h2 a')[0].text
5 print(day,month,year,desc)
```

```
1 Nov 2017 Scorpions at Olympijskiy Stadium, Moscow, Russia
```

In [6]:
```
1 idx = desc.find(' at ')+4
2 loc = desc[idx:]
3 loc_pieces = loc.split(',')
4 if len(loc_pieces)>=3:
5     loc = ','.join(loc_pieces[-3:])
6 print(loc)
```

```
Olympijskiy Stadium, Moscow, Russia
```

In [7]:
```
g=geocoder.google(loc)
print(g.geojson)

```


```
{'type': 'FeatureCollection', 'features': [{'type': 'Feature', 'properties': {'accuracy': 'ROOFTOP',
 'address': 'Olimpiyskiy pr., 16с1, Moskva, Russia, 129090', 'bbox': [37.6251080197085, 55.77970101970
85, 37.6278059802915, 55.7823989802915], 'city': 'Moskva', 'confidence': 9, 'country': 'RU', 'county':
'Moskva', 'housenumber': '16с1', 'lat': 55.78105, 'lng': 37.62645699999999, 'ok': True, 'place': 'ChIJ
QcqpSw1KtUYRrBjnAynHt5E', 'postal': '129090', 'quality': 'establishment', 'raw': {'address_component
s': [{'long_name': '16 строение 1', 'short_name': '16с1', 'types': ['street_number']}, {'long_name': 'O
limpiyskiy prospekt', 'short_name': 'Olimpiyskiy pr.', 'types': ['route']}, {'long_name': 'Tsentralnyy
administrativnyy okrug', 'short_name': 'Tsentralnyy administrativnyy okrug', 'types': ['political', 's
ublocality', 'sublocality_level_1']}, {'long_name': 'Moskva', 'short_name': 'Moskva', 'types': ['local
ity', 'political']}, {'long_name': 'Moskva', 'short_name': 'Moskva', 'types': ['administrative_area_le
vel_2', 'political']}, {'long_name': 'Russia', 'short_name': 'RU', 'types': ['country', 'political']},
{'long_name': '129090', 'short_name': '129090', 'types': ['postal_code']}], 'formatted_address': 'Olim
piyskiy pr., 16с1, Moskva, Russia, 129090', 'geometry': {'location': {'lat': 55.78105, 'lng': 37.62645
699999999}, 'location_type': 'ROOFTOP', 'viewport': {'northeast': {'lat': 55.7823989802915, 'lng': 37.
6278059802915}, 'southwest': {'lat': 55.7797010197085, 'lng': 37.6251080197085}}}, 'partial_match': Tr
ue, 'place_id': 'ChIJQcqpSw1KtUYRrBjnAynHt5E', 'types': ['establishment', 'point_of_interest'], 'locat
ion': {'lat': 55.78105, 'lng': 37.62645699999999}, 'location_type': 'ROOFTOP', 'bounds': {}, 'northeas
t': {'lat': 55.7823989802915, 'lng': 37.6278059802915}, 'southwest': {'lat': 55.7797010197085, 'lng':
 37.6251080197085}, 'street_number': {'long_name': '16 строение 1', 'short_name': '16с1'}, 'route': {'l
ong_name': 'Olimpiyskiy prospekt', 'short_name': 'Olimpiyskiy pr.'}, 'political': {'long_name': 'Russi
a', 'short_name': 'RU'}, 'sublocality': {'long_name': 'Tsentralnyy administrativnyy okrug', 'short_nam
e': 'Tsentralnyy administrativnyy okrug'}, 'sublocality_level_1': {'long_name': 'Tsentralnyy administr
ativnyy okrug', 'short_name': 'Tsentralnyy administrativnyy okrug'}, 'locality': {'long_name': 'Moskv
a', 'short_name': 'Moskva'}, 'administrative_area_level_2': {'long_name': 'Moskva', 'short_name': 'Mos
kva'}, 'country': {'long_name': 'Russia', 'short_name': 'RU'}, 'postal_code': {'long_name': '129090',
 'short_name': '129090'}}, 'status': 'OK', 'street': 'Olimpiyskiy pr.', 'sublocality': 'Tsentralnyy ad
ministrativnyy okrug'}, 'bbox': [37.6251080197085, 55.7797010197085, 37.6278059802915, 55.782398980291
5], 'geometry': {'type': 'Point', 'coordinates': [37.62645699999999, 55.78105]}}]}
```


In [8]:
```
code=g.geojson['features'][0]['properties']['country']
print(code)
```

```
RU
```

In [9]:

```
from iso3166 import countries
code=countries.get(code).alpha3
print(code)
```

```
RUS
```

In [10]:

```
@memory.cache
def get_latlng(query):
    try:
        g=geocoder.google(query)
        if g:
            country = g.geojson['features'][0]['properties']['country']
            code=countries.get(country).alpha3
            print(query, g.latlng, code)
            return g.latlng+[code]
        else:
            return None, None, None
    except:
        return None, None, None


@memory.cache
def get_data_paged(query, page):
    url = 'https://www.setlist.fm/search?page={}&query={}'.format(page, query)
    browser.open(url.format(page))
    data = []
    for concert in browser.select('.setlistPreview'):
        month = concert.select('.month')[0].text
        day = concert.select('.day')[0].text
        year = concert.select('.year')[0].text
        datetext = "{}, {} {}".format(year, month, day)
        date = dateparser.parse(datetext)
        desc = concert.select('h2 a')[0].text
        idx = desc.find(' at ')+4
        loc = desc[idx:]
        loc_pieces = loc.split(',')
        #print(loc_pieces[-1],country)
        if loc == None:
            continue
        if len(loc_pieces)>=3:
            loc = ','.join(loc_pieces[-3:])
        lat, lng, code = get_latlng(loc)
        if lat and lng:
            data.append([loc, lat, lng, code, date, desc])
    return data

columns = ['loc', 'lat', 'lon', 'code', 'date', 'desc']
concerts = pd.DataFrame(columns = columns)
```

```
for page in range(1,15):
    data = get_data_paged('Scorpions', page)
    df = pd.DataFrame(data, columns = columns)
    concerts = concerts.append(df)

concerts.head(20)
```

Out[10]:

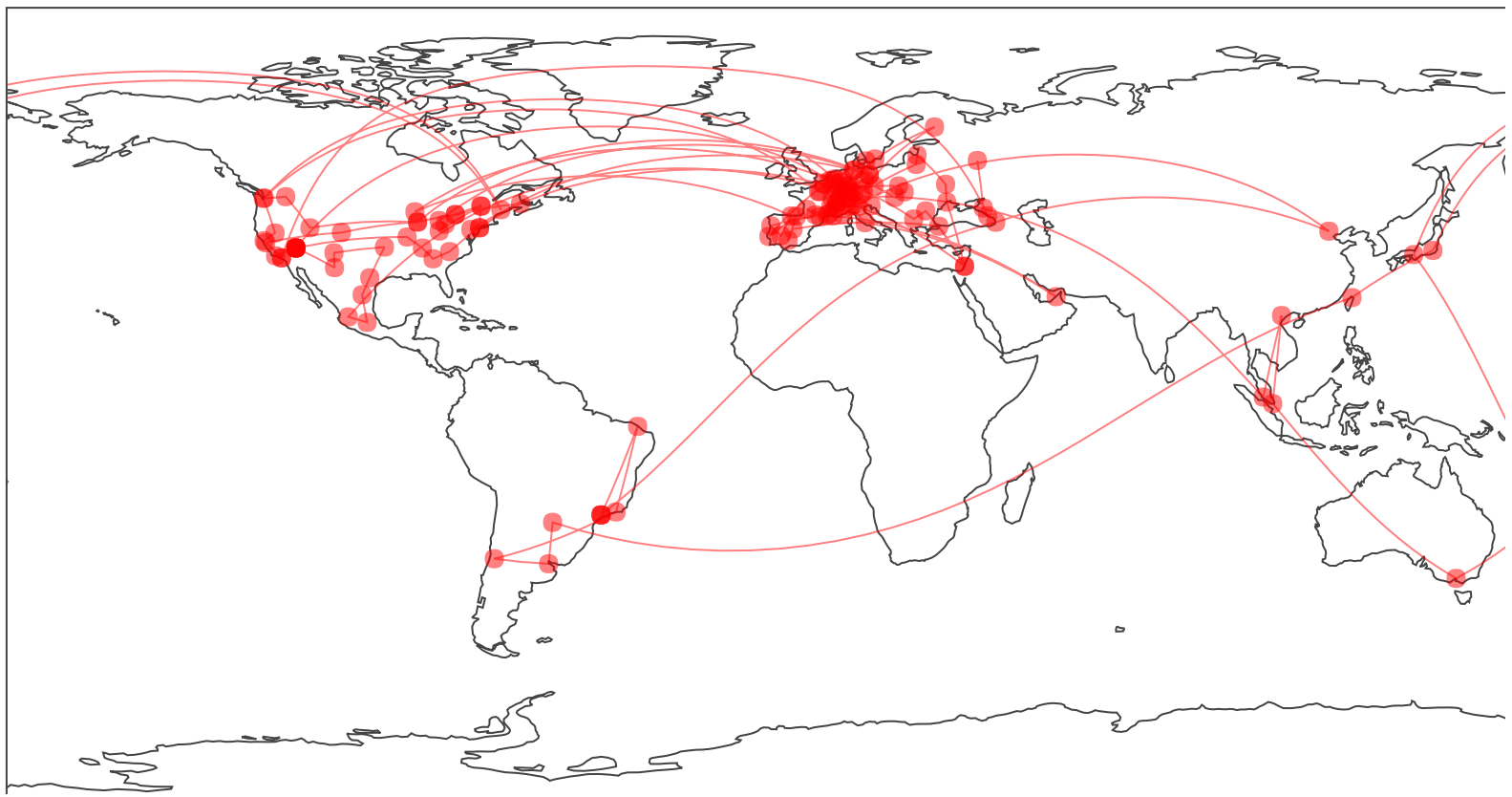
	loc	lat	lon	code	date	desc
0	Olympijskiy Stadium, Moscow, Russia	55.781050	37.626457	RUS	2017-11-01	Scorpions at Olympijskiy Stadium, Moscow, Russia
1	SK "Basket Hall", Krasnodar, Russia	45.117544	38.981297	RUS	2017-10-30	Scorpions at SK "Basket Hall", Krasnodar, Russia
2	Ledovaya Arena 'Shayba', Sochi, Russia	43.402260	39.951905	RUS	2017-10-28	Scorpions at Ledovaya Arena 'Shayba', Sochi, R...
3	Inglewood, CA, USA	33.961680	-118.353131	USA	2017-10-07	Scorpions at The Forum, Inglewood, CA, USA
4	Oakland, CA, USA	37.804364	-122.271114	USA	2017-10-04	Scorpions at Oracle Arena, Oakland, CA, USA
5	Reno, NV, USA	39.529633	-119.813803	USA	2017-10-03	Scorpions at Grand Sierra Theatre, Reno, NV, USA
6	Tacoma, WA, USA	47.252877	-122.444291	USA	2017-09-30	Scorpions at Tacoma Dome, Tacoma, WA, USA
7	Spokane, WA, USA	47.658780	-117.426046	USA	2017-09-29	Scorpions at Spokane Arena, Spokane, WA, USA
8	West Valley, UT, USA	40.691613	-112.001050	USA	2017-09-26	Scorpions at USANA Amphitheatre, West Valley, ...
0	Rosemont, IL, USA	41.986751	-87.872160	USA	2017-09-23	Scorpions at Allstate Arena, Rosemont, IL, USA
1	Toronto, ON, Canada	43.653226	-79.383184	CAN	2017-09-22	Scorpions at Budweiser Stage, Toronto, ON, Canada
2	Laval, QC, Canada	45.606649	-73.712409	CAN	2017-09-19	Scorpions at Place Bell, Laval, QC, Canada
3	New York, NY, USA	40.712775	-74.005973	USA	2017-09-16	Scorpions at Madison Square Garden, New York, ...
4	Reading, PA, USA	40.335648	-75.926875	USA	2017-09-14	Scorpions at Santander Arena, Reading, PA, USA
5	Münsterplatz, Ulm, Germany	48.399008	9.991753	DEU	2017-07-23	Scorpions at Münsterplatz, Ulm, Germany
6	Guitare en scène Festival 2017	46.139233	6.071985	FRA	2017-07-19	Scorpions at Guitare en scène Festival 2017
7	Festival de Nîmes 2017	43.834904	4.359615	FRA	2017-07-17	Scorpions at Festival de Nîmes 2017
8	Festival Marés Vivas 2017	41.139685	-8.653662	PRT	2017-07-15	Scorpions at Festival Marés Vivas 2017
9	Albergue Municipal Juvenil El Prado, Mérida, S...	38.930452	-6.401968	ESP	2017-07-14	Scorpions at Albergue Municipal Juvenil El Pra...
0	Campos del Malecón, Torrelavega, Spain	43.350213	-4.062679	ESP	2017-07-12	Scorpions at Campos del Malecón, Torrelavega, ...

In [11]:

```
import plotly.plotly as py

great_lines = [
        dict(
            type = 'scattergeo',
            lon = concerts['lon'],
            lat = concerts['lat'],
            mode = 'lines',
            line = dict(
                    width = 1,
                    color = 'rgba(255,0,0,0.5)',
                    ),
        )
        ]
venue_markers = [ dict(
        type = 'scattergeo',
        lon = concerts['lon'],
        lat = concerts['lat'],
        hoverinfo = 'loc',
        text = concerts['loc'],
        mode = 'markers',
        marker = dict(
            size=10,
            color='rgba(255,0,0,0.5)',
        ))]

layout = dict(
        title = 'recitales',
        width = 1000,
        height = 800,
            showlegend = False,

            showland = True,
            showcountries = True,
            showocean = True,
            countrywidth = 0.5,
            landcolor = '#fff',
            oceancolor = '#eee',

        geo = dict(
            projection = dict(
                type = 'Mercator',
```

```
            ),

        )
    )

fig = dict( data=great_lines+venue_markers, layout=layout )
py.iplot( fig, validate=False, filename='d3-globe' )
```

Out[11]:



In [12]:
```
1 df = concerts.groupby('code').count()
2 df.head(20)
```

Out[12]:

code	loc	lat	lon	date	desc
ARE	1	1	1	1	1
ARG	1	1	1	1	1
AUS	1	1	1	1	1
BEL	3	3	3	3	3
BGR	1	1	1	1	1
BRA	5	5	5	5	5
CAN	6	6	6	6	6
CHE	3	3	3	3	3
CHL	1	1	1	1	1
CHN	1	1	1	1	1
CZE	1	1	1	1	1
DEU	14	14	14	14	14
DNK	1	1	1	1	1
ESP	5	5	5	5	5
FIN	1	1	1	1	1
FRA	16	16	16	16	16
GEO	1	1	1	1	1
HUN	1	1	1	1	1
ISR	2	2	2	2	2
ITA	5	5	5	5	5

In [13]:
```
df=df.reset_index()
df.head(20)
```

Out[13]:

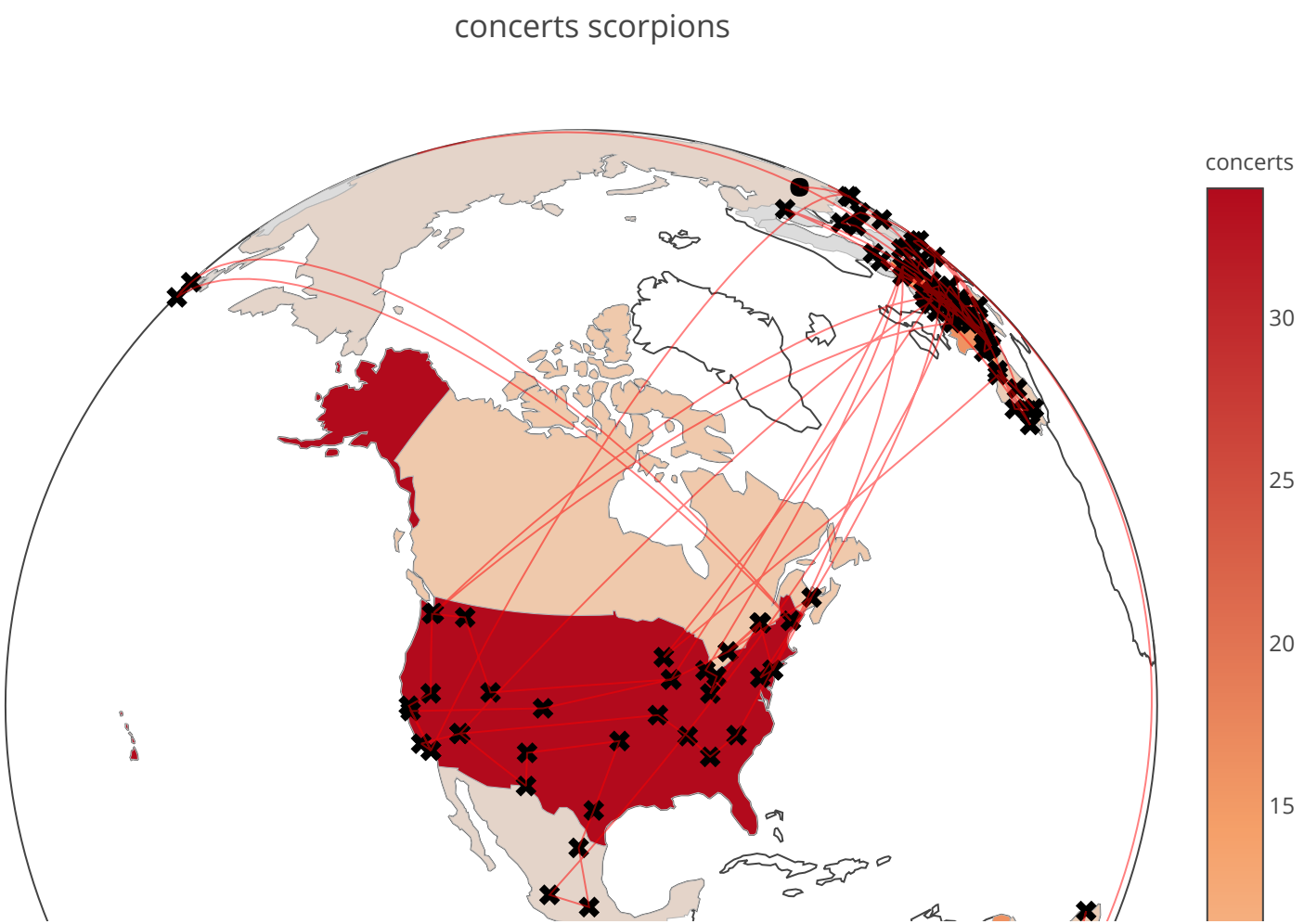
	code	loc	lat	lon	date	desc
0	ARE	1	1	1	1	1
1	ARG	1	1	1	1	1
2	AUS	1	1	1	1	1
3	BEL	3	3	3	3	3
4	BGR	1	1	1	1	1
5	BRA	5	5	5	5	5
6	CAN	6	6	6	6	6
7	CHE	3	3	3	3	3
8	CHL	1	1	1	1	1
9	CHN	1	1	1	1	1
10	CZE	1	1	1	1	1

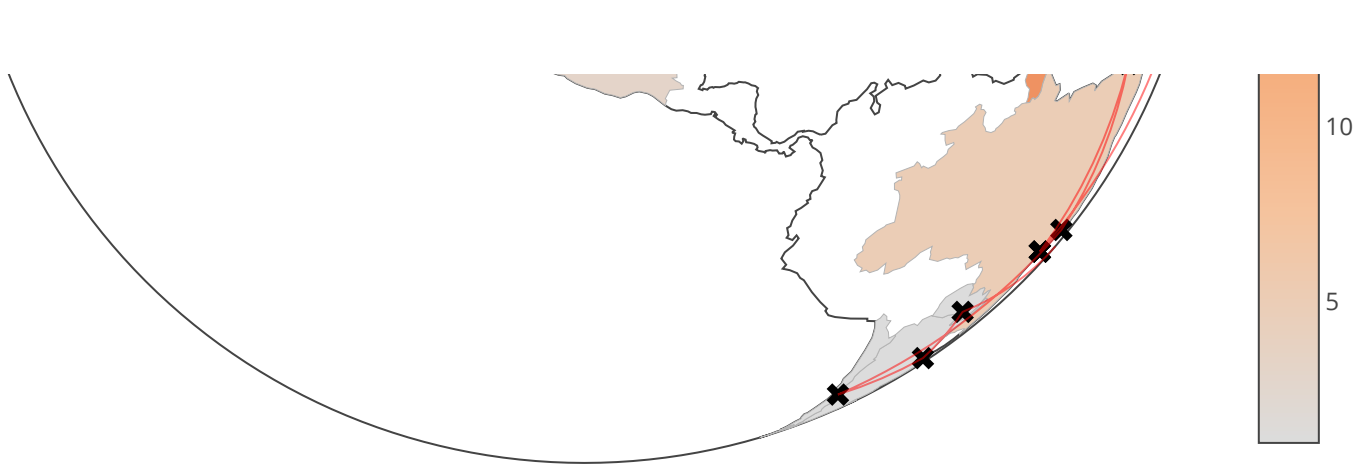
In [14]:

```
data = [ dict(
        type = 'choropleth',
        locations = df['code'],
        z = df['loc'],
        autocolorscale = True,
        marker = dict(
            line = dict (
                color = 'rgb(180,180,180)',
                width = 0.5
            ) ),
        colorbar = dict(
            autotick = False,
            tickprefix = '',
            title = 'concerts'),
      ) ]

markers = [ dict(
        type = 'scattergeo',
        lon = concerts['lon'],
        lat = concerts['lat'],
        hovertext  = concerts['loc'],
        text = concerts['loc'] ,
        textposition = 'top center',
        mode = 'markers',
        string = concerts['loc'],
        marker = dict(
            symbol = ['4'],
            size=10,
            color='black',
            linecolor = 'white'

        ))]

layout = dict(
    width = 800,
    height = 800,
    showlegend = False,
    title = 'concerts scorpions',
    geo = dict(
        showframe = True,
        showcoastlines = True,
        projection = dict(
```

```
                type = 'orthographic',
                rotation = dict(
                    lon = -100,
                    lat = 40,
                    roll = 0
                )
            ),
    )
)

fig = dict( data=data+markers+great_lines, layout=layout )
py.iplot( fig, validate=False, filename='d3-world-map' )
```

Out[14]:





EDIT CHART

In [ ]:
```
1
```