Scatter Plots in Python

How to make scatter plots in Python with Plotly.

Scala MATLAB R Python Pandas

plotly.js node.js matplotlib

New to Plotly?

Plotly's Python library is free and open source! Get started by downloading the client and reading the primer.

You can set up Plotly to work in online or offline mode, or in jupyter notebooks.

We also have a quick-reference cheatsheet (new!) to help you get started!

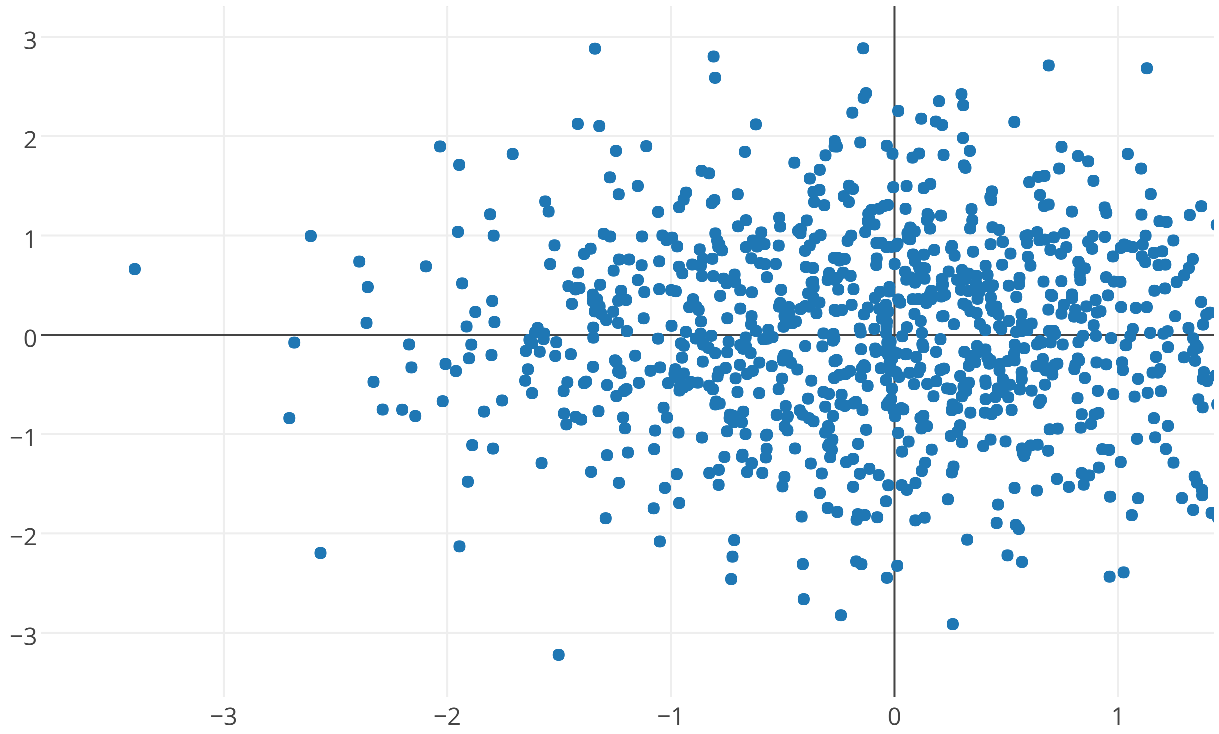
Simple Scatter Plot

In [1]:

Copy to clipboard!

```
import plotly.plotly as py
import plotly.graph_objs as go

# Create random data with numpy
import numpy as np

N = 1000
random_x = np.random.randn(N)
random_y = np.random.randn(N)

# Create a trace
trace = go.Scatter(
    x = random_x,
    y = random_y,
    mode = 'markers'
)

data = [trace]

# Plot and embed in ipython notebook!
py.iplot(data, filename='basic-scatter')

# or plot with: plot_url = py.plot(data, filename='basic-line')
```

Out[1]:



EDIT CHART

Line and Scatter Plots

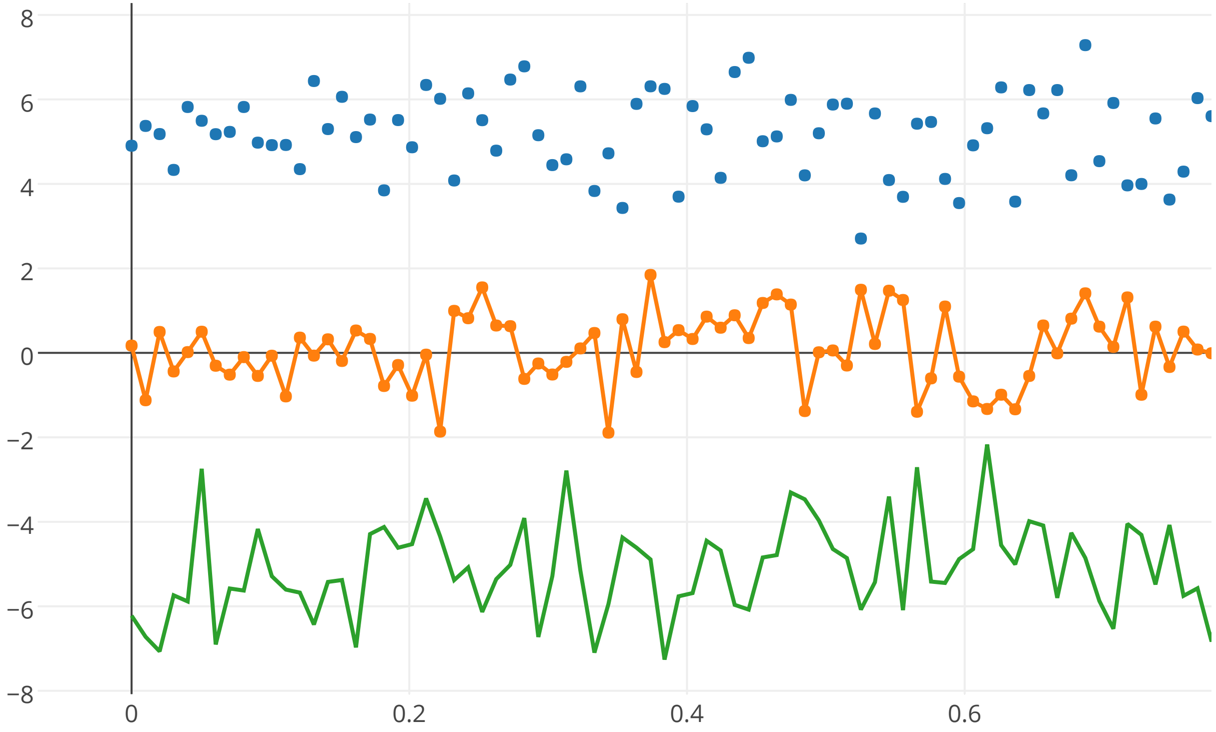
```
In [2]:
```

Copy to clipboard!

```
import plotly.plotly as py
import plotly.graph_objs as go

# Create random data with numpy
import numpy as np

N = 100
random_x = np.linspace(0, 1, N)
random_y0 = np.random.randn(N)+5
random_y1 = np.random.randn(N)
random_y2 = np.random.randn(N)-5

# Create traces
trace0 = go.Scatter(
    x = random_x,
    y = random_y0,
    mode = 'markers',
    name = 'markers'
)
trace1 = go.Scatter(
    x = random_x,
    y = random_y1,
    mode = 'lines+markers',
    name = 'lines+markers'
)
trace2 = go.Scatter(
    x = random_x,
    y = random_y2,
    mode = 'lines',
    name = 'lines'
)
```

```
data = [trace0, trace1, trace2]
py.iplot(data, filename='scatter-mode')
```

Out[2]:



EDIT CHART

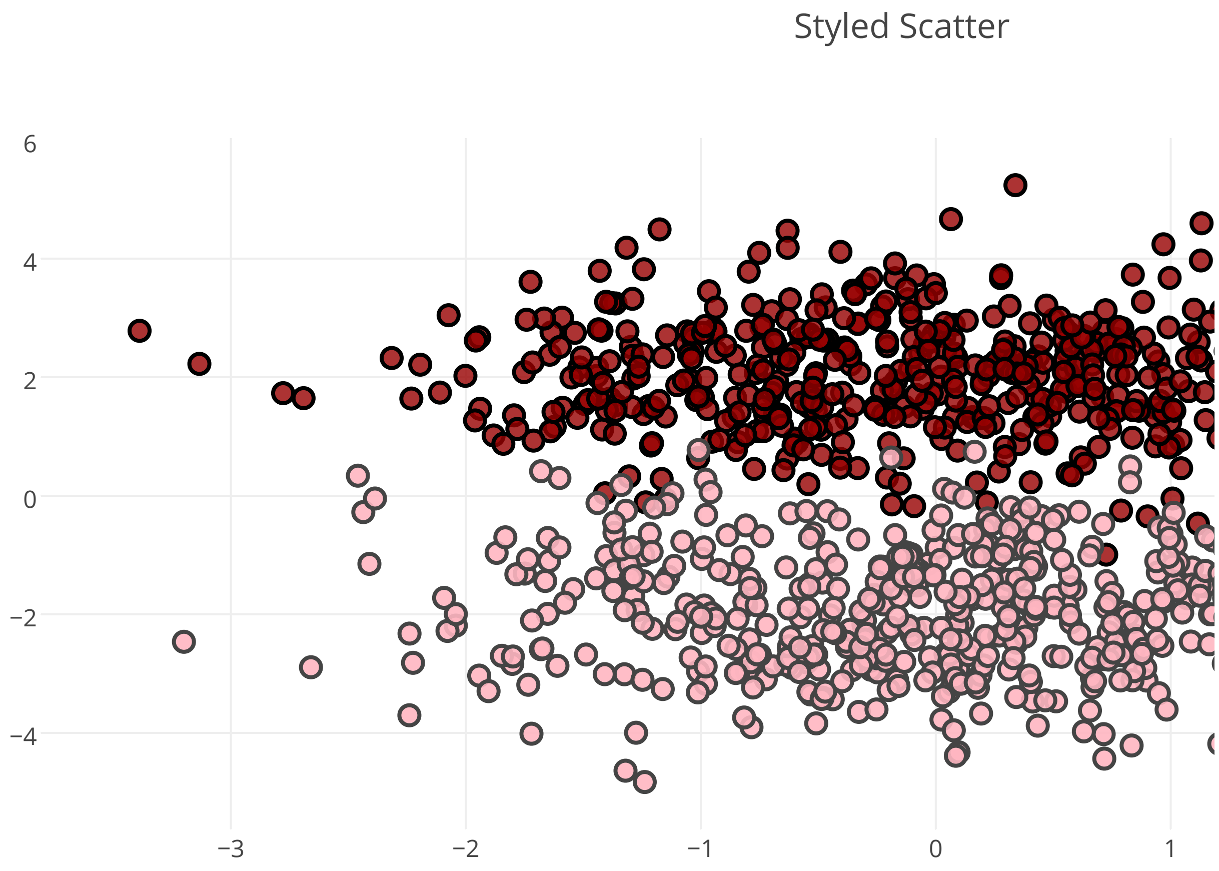
Style Scatter Plots

In [3]:

Copy to clipboard!

```
import plotly.plotly as py
import plotly.graph_objs as go

import numpy as np

N = 500

trace0 = go.Scatter(
    x = np.random.randn(N),
    y = np.random.randn(N)+2,
    name = 'Above',
    mode = 'markers',
    marker = dict(
        size = 10,
        color = 'rgba(152, 0, 0, .8)',
        line = dict(
            width = 2,
            color = 'rgb(0, 0, 0)'
        )
    )
)

trace1 = go.Scatter(
    x = np.random.randn(N),
    y = np.random.randn(N)-2,
    name = 'Below',
    mode = 'markers',
    marker = dict(
        size = 10,
        color = 'rgba(255, 182, 193, .9)',
        line = dict(
            width = 2,
        )
    )
)

data = [trace0, trace1]

layout = dict(title = 'Styled Scatter',
              yaxis = dict(zeroline = False),
              xaxis = dict(zeroline = False)
             )

fig = dict(data=data, layout=layout)
py.iplot(fig, filename='styled-scatter')
```

Out[3]:



EDIT CHART

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Navigation

- New to Plotly
- Simple Scatter Plot
- Line and Scatter Plots
- Style Scatter Plots
- Data Labels on Hover
- Scatter with a Color Dimension
- Categorical Dot Plot
- Large Data Sets
- Reference

Copy to clipboard!

```python
import plotly.plotly as py
import plotly.graph_objs as go
import random
import numpy as np
import pandas as pd

l= []
y= []
data= pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/2014_usa_states.csv")
# Setting colors for plot.
N= 53
c= ['hsl('+str(h)+',50%'+',50%)' for h in np.linspace(0, 360, N)]

for i in range(int(N)):
    y.append((2000+i))
    trace0= go.Scatter(
        x= data['rank'],
        y= data['pop']+(i*1000000),
        mode= 'markers',
        marker= dict(size= 14,
                    line= dict(width=1),
                    color= c[i],
                    opacity= 0.3
                   ),name= y[i],
        text= data['state']) # The hover text goes here...
    l.append(trace0);
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots | Fork on Github

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

```
        ticklen= 5,
        zeroline= False,
        gridwidth= 2,
    ),
    yaxis=dict(
        title= 'Rank',
        ticklen= 5,
        gridwidth= 2,
    ),
    showlegend= False
)
fig= go.Figure(data=l, layout=layout)
py.iplot(fig)
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

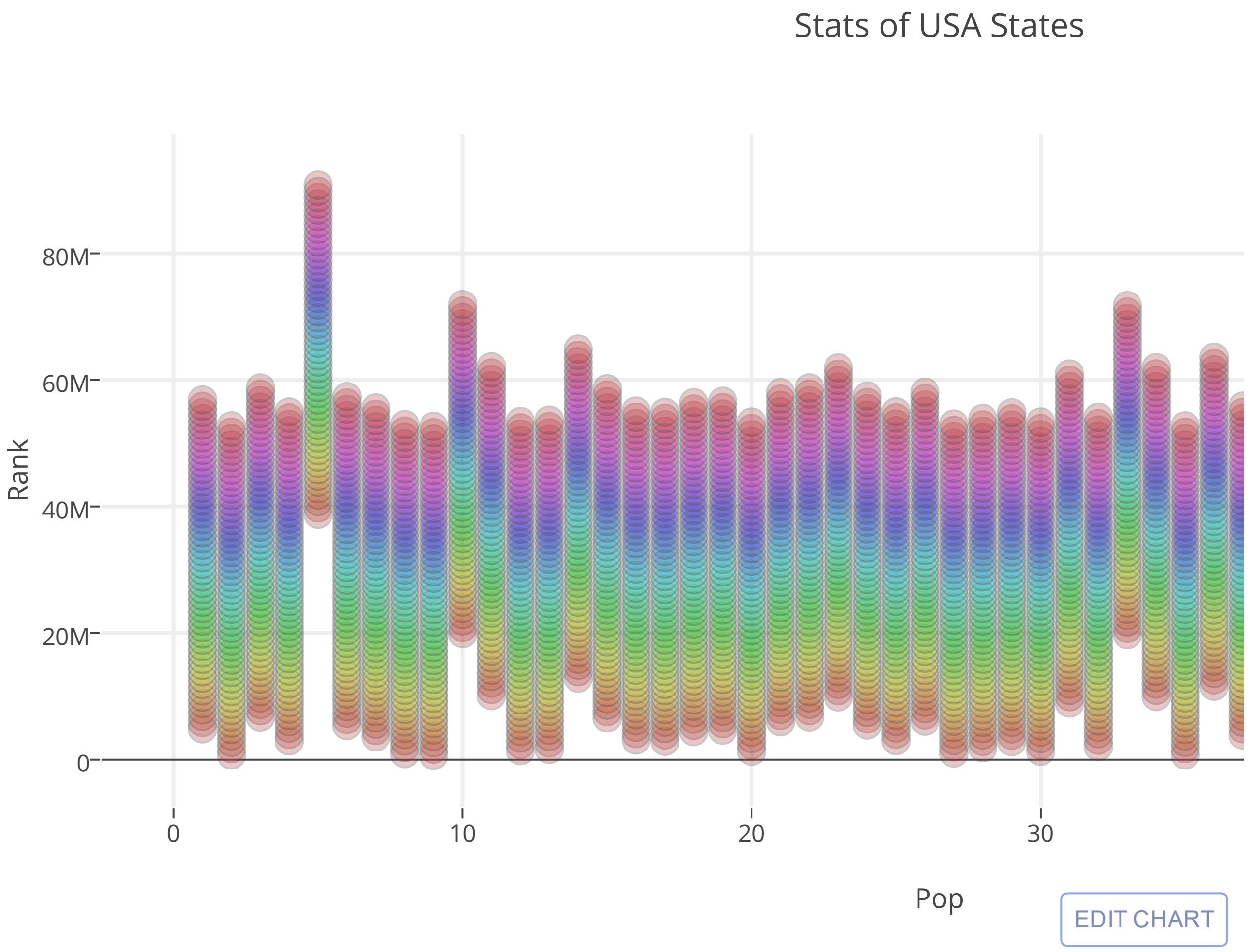
Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

```
Out[4]:
```



Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

In [5]:

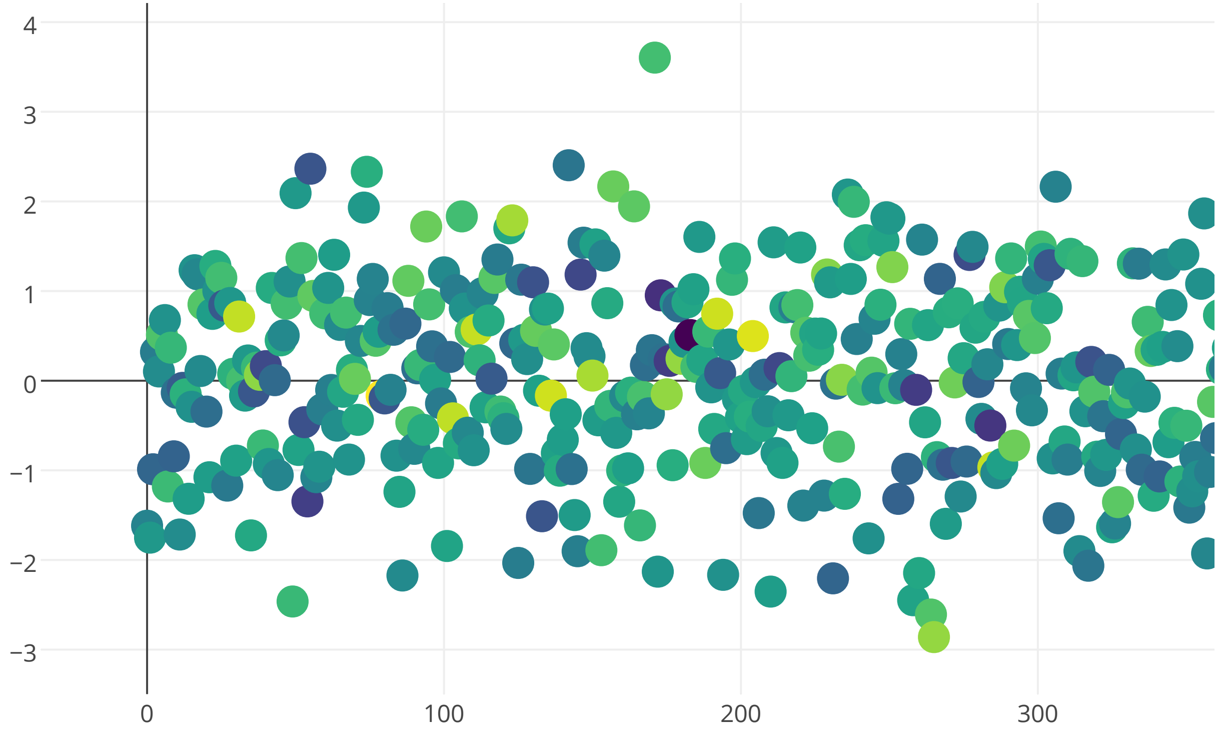
Copy to clipboard!

```
import plotly.graph_objs as go
import plotly.plotly as py

import numpy as np

trace1 = go.Scatter(
    y = np.random.randn(500),
    mode='markers',
    marker=dict(
        size='16',
        color = np.random.randn(500), #set color equal to a variable
        colorscale='Viridis',
        showscale=True
    )
)
data = [trace1]

py.iplot(data, filename='scatter-plot-with-colorscale')
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots | Fork on Github

Out[5]:



EDIT CHART

Navigation

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

Copy to clipboard!

```
import plotly.plotly as py
import plotly.graph_objs as go

country = ['Switzerland (2011)', 'Chile (2013)', 'Japan (2014)', 'United States (2012)', 'Slovenia (2014)', 'Canada (2011)', 'Poland (2010)', 'Estonia (2015)', 'Luxembourg (2013)', 'Portugal (2011)']
voting_pop = [40, 45.7, 52, 53.6, 54.1, 54.2, 54.5, 54.7, 55.1, 56.6]
reg_voters = [49.1, 42, 52.7, 84.3, 51.7, 61.1, 55.3, 64.2, 91.1, 58.9]

trace0 = go.Scatter(
    x=voting_pop,
    y=country,
    mode='markers',
    name='Percent of estimated voting age population',
    marker=dict(
        color='rgba(156, 165, 196, 0.95)',
        line=dict(
            color='rgba(156, 165, 196, 1.0)',
            width=1,
        ),
        symbol='circle',
        size=16,
    )
)
trace1 = go.Scatter(
    x=reg_voters,
    y=country,
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots | Fork on Github

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

```
            width=1,
        ),
        symbol='circle',
        size=16,
    )
)
data = [trace0, trace1]
layout = go.Layout(
    title="Votes cast for ten lowest voting age population in OECD countries",
    xaxis=dict(
        showgrid=False,
        showline=True,
        linecolor='rgb(102, 102, 102)',
        titlefont=dict(
            color='rgb(204, 204, 204)'
        ),
        tickfont=dict(
            color='rgb(102, 102, 102)',
        ),
        autotick=False,
        dtick=10,
        ticks='outside',
        tickcolor='rgb(102, 102, 102)',
    ),
    margin=dict(
        l=140,
        r=40,
        b=50,
```

plotly Pricing PLOTCON NYC API

Help API Libraries Python Scatter Plots Fork on Github

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

Style Scatter Plots

Data Labels on Hover

Scatter with a Color Dimension

Categorical Dot Plot

Large Data Sets

Reference

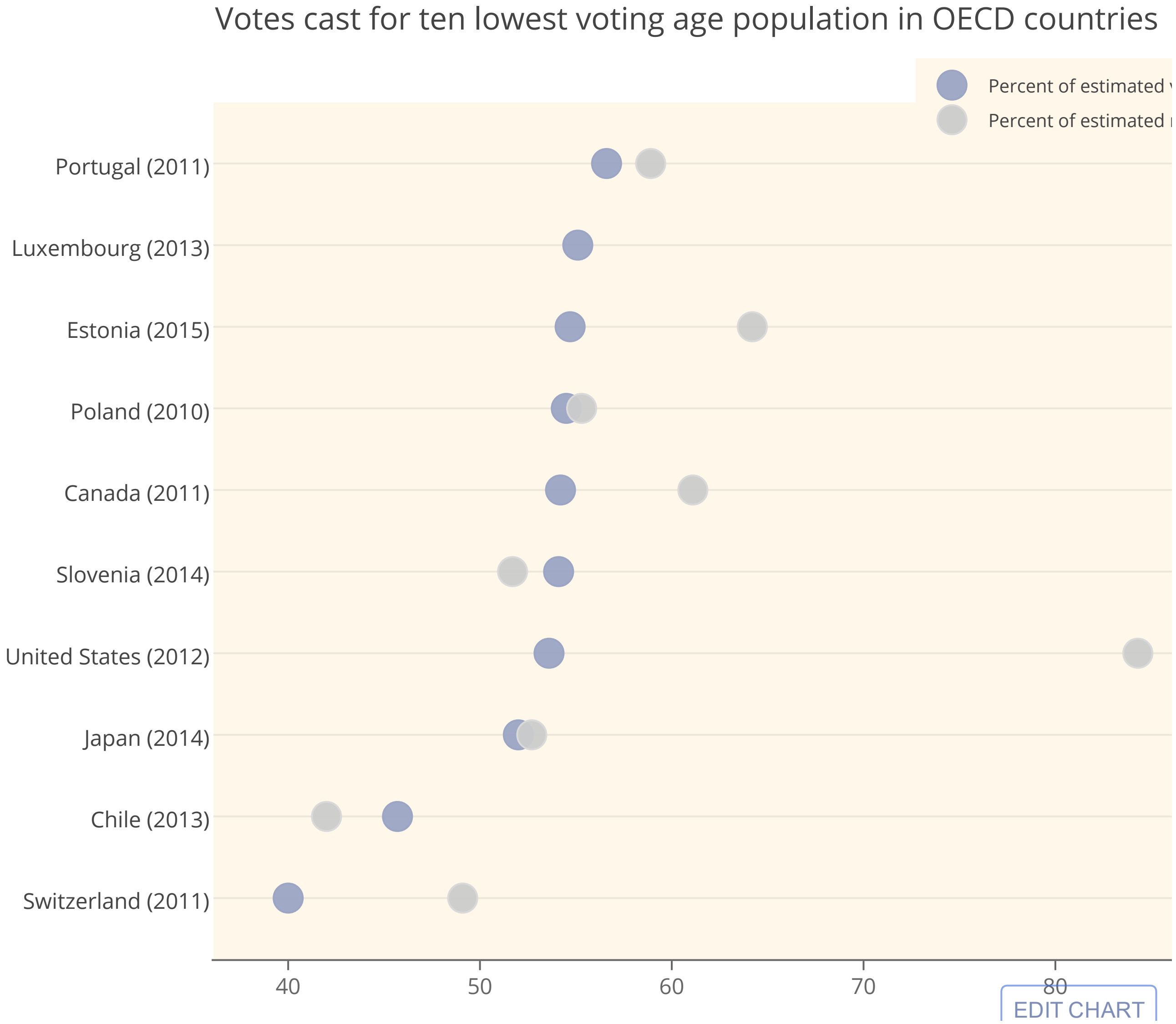
```
        yanchor='middle',
        xanchor='right',
    ),
    width=800,
    height=600,
    paper_bgcolor='rgb(254, 247, 234)',
    plot_bgcolor='rgb(254, 247, 234)',
    hovermode='closest',
)
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='lowest-oecd-votes-cast')
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Out[6]:


Votes cast for ten lowest voting age population in OECD countries
Percent of estimated v
Percent of estimated r
Portugal (2011)
Luxembourg (2013)
Estonia (2015)
Poland (2010)
Canada (2011)
Slovenia (2014)
United States (2012)
Japan (2014)
Chile (2013)
Switzerland (2011)
40
50
60
70
80
EDIT CHART

Navigation

Large Data Sets

Now in Ploty you can implement WebGL with `Scattergl()` in place of `Scatter()` for increased speed, improved interactivity, and the ability to plot even more data!

In [7]:

Copy to clipboard!

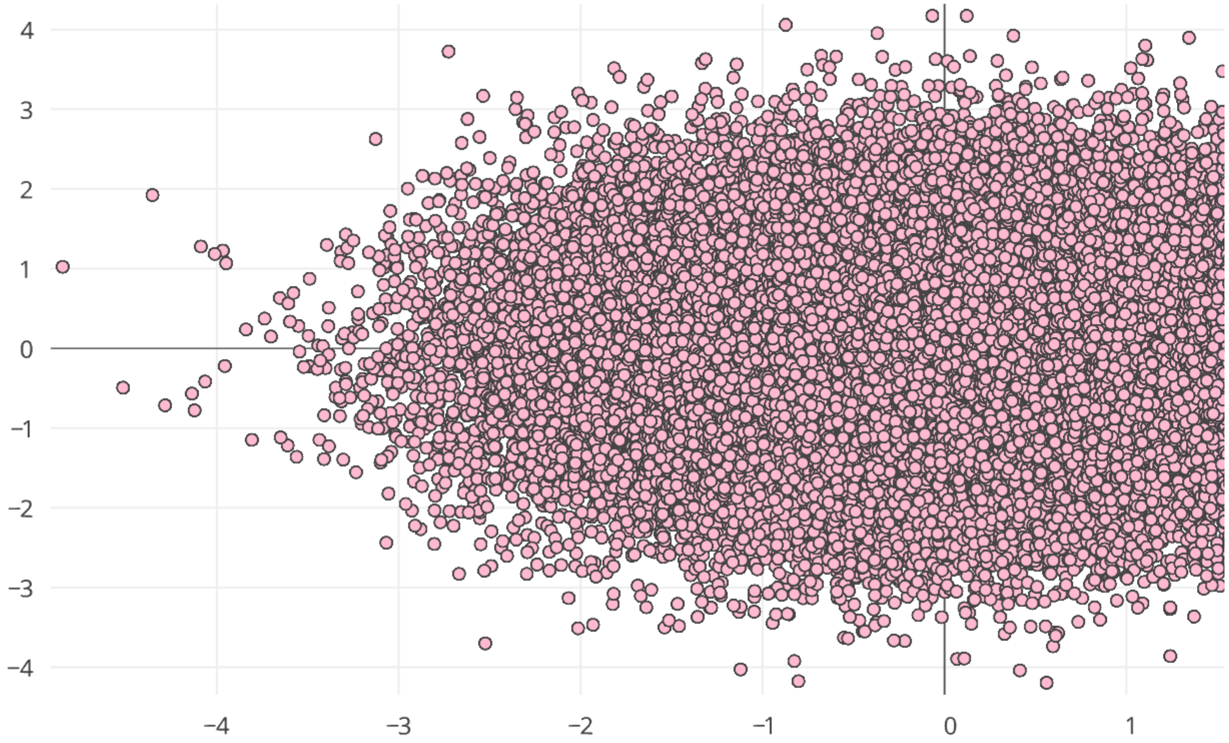
```
import plotly.plotly as py
import plotly.graph_objs as go
import numpy as np

N = 100000
trace = go.Scattergl(
    x = np.random.randn(N),
    y = np.random.randn(N),
    mode = 'markers',
    marker = dict(
        color = 'FFBAD2',
        line = dict(width = 1)
    )
)
data = [trace]
py.iplot(data, filename='compare_webgl')
```

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Out[7]:



EDIT CHART

Navigation

New to Plotly

Simple Scatter Plot

Line and Scatter Plots

See https://plot.ly/python/reference/#scatter or https://plot.ly/python/reference/#scattergl for more information and chart attribute options!

Still need help?

Contact Us

community.plot.ly | @plotlygraphs | github.com/plotly

For guaranteed 24 hour response turnarounds, upgrade to our Premium or Enterprise plans.

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | Scatter Plots

Fork on Github

Plotly.js
Hardware

Modern Data

Solutions

Plans & Pricing
Enterprise
Education
Plotly.js

Connect

Twitter Facebook Github
LinkedIn Google+

Terms of Service Privacy Policy

Categorical Dot Plot

Large Data Sets

Reference

plotly Pricing PLOTCON NYC API

Help | API Libraries | Python | **Scatter Plots**

Fork on Github