

Plotly 소개

d3.js를 이용하여 interactive하게 그래프를 보여준다.

사전 파이썬 버전 확인

- (base) C:\Users\toto>python --version
- Python 3.7.7

plotly를 pandas와 함께 사용하는 법

- cufflinks 설정과 .iplot()을 활용. pandas.plot()와 같이 판다스 데이터 시각화
- plotly.express 라이브러리 활용

cufflinks 는 무엇

- 판다스 데이터 프레임과 Plotly를 연결하여 사용자가 판다스로부터 직접 시각화를 할 수 있는 라이브러리

01 시작하기 - 설치(Plotly and Cufflinks)

- pip install plotly or conda install plotly
- pip install cufflinks
- 에러 발생시
 - 방법1
 - python -m pip install --upgrade pip
 - python -m pip install plotly
 - 방법2
 - C:\ProgramData\Anaconda3\python -m pip install plotly
 - C:\ProgramData\Anaconda3\python -m pip install cufflinks

In [8]:

```
import sys
print(sys.path)
```

```
['C:\\Users\\user\\Documents\\GitHub\\AI_enovation\\part02_library\\20201016_class_datavis_pandas\\part02_03_04_plotly', 'C:\\ProgramData\\Anaconda3\\python38.zip', 'C:\\ProgramData\\Anaconda3\\DLLs', 'C:\\ProgramData\\Anaconda3\\lib', 'C:\\ProgramData\\Anaconda3', '', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\user\\.ipython']
```

In [9]:



```
import plotly
import cufflinks as cf
import pandas as pd
import numpy as np
```

버전 확인

In [10]:



```
print(plotly.__version__)
print(cf.__version__)
print(pd.__version__)
print(np.__version__)
```

```
4.11.0
0.17.3
1.0.5
1.18.5
```

In [11]:



```
# 오프라인 모드에서도 인터랙티브한 그래픽을 가능하도록 하기
# Enabling the offline mode for interactive plotting locally
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
cf.go_offline()
```

데이터 생성 및 plot

In [12]:



```
# 데이터 생성
df = pd.DataFrame(np.random.randn(100,4), # 100개 4개 컬럼
                  columns='A B C D'.split())
print(df.shape)
df.head()
```

(100, 4)

Out[12]:

	A	B	C	D
0	1.435795	-0.482263	-0.909916	-0.520652
1	-1.878866	-0.777998	0.508750	1.435092
2	0.663726	0.434423	1.594917	0.154812
3	0.119395	0.980587	-0.914278	-0.386150
4	0.868688	-0.363241	-0.359944	-0.742103

In [13]:

```
df2 = pd.DataFrame({'items':['bag','apple','cap'],'Values':[32,43,50]})  
df2
```

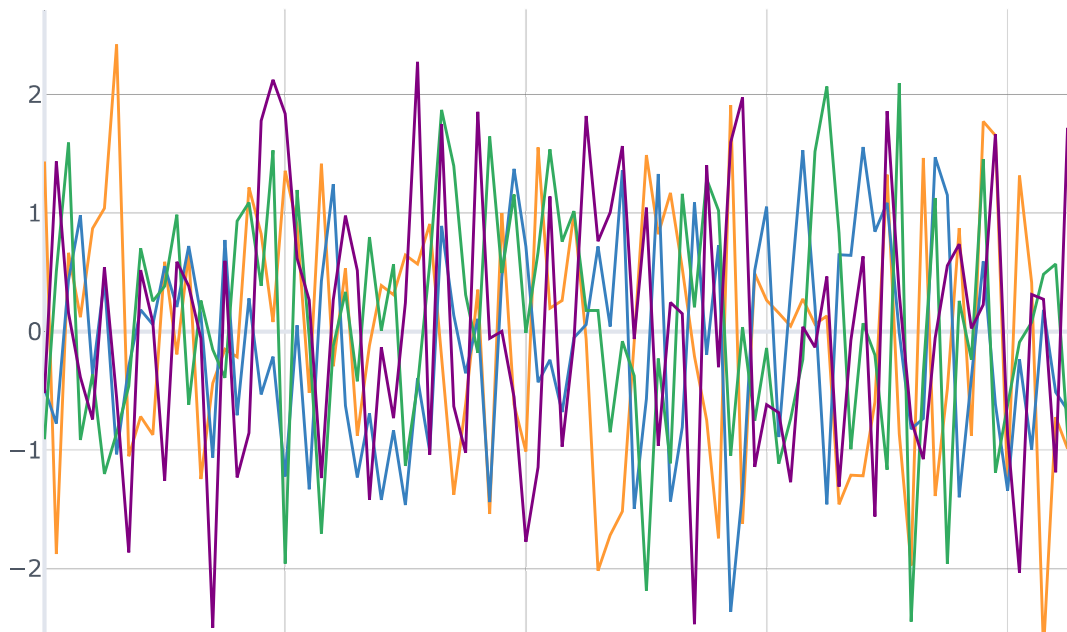
Out[13]:

	items	Values
0	bag	32
1	apple	43
2	cap	50

Line Plot

In [14]:

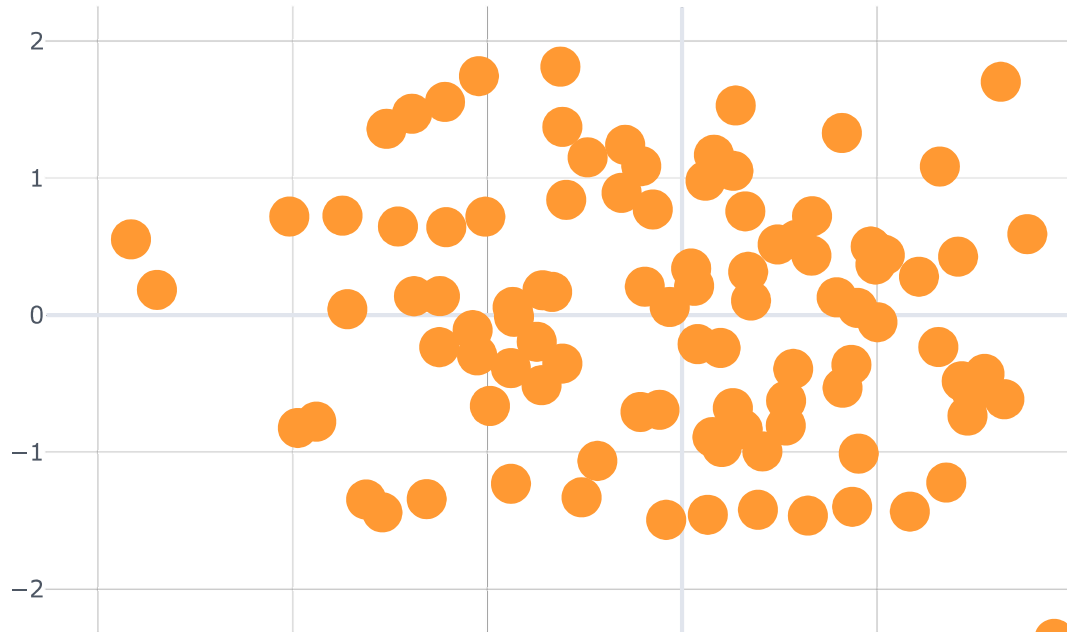
```
df.iplot()
```



Scatter Plot

In [15]:

```
df.iplot(kind='scatter', x='A',y='B',mode='markers',size=20)
```

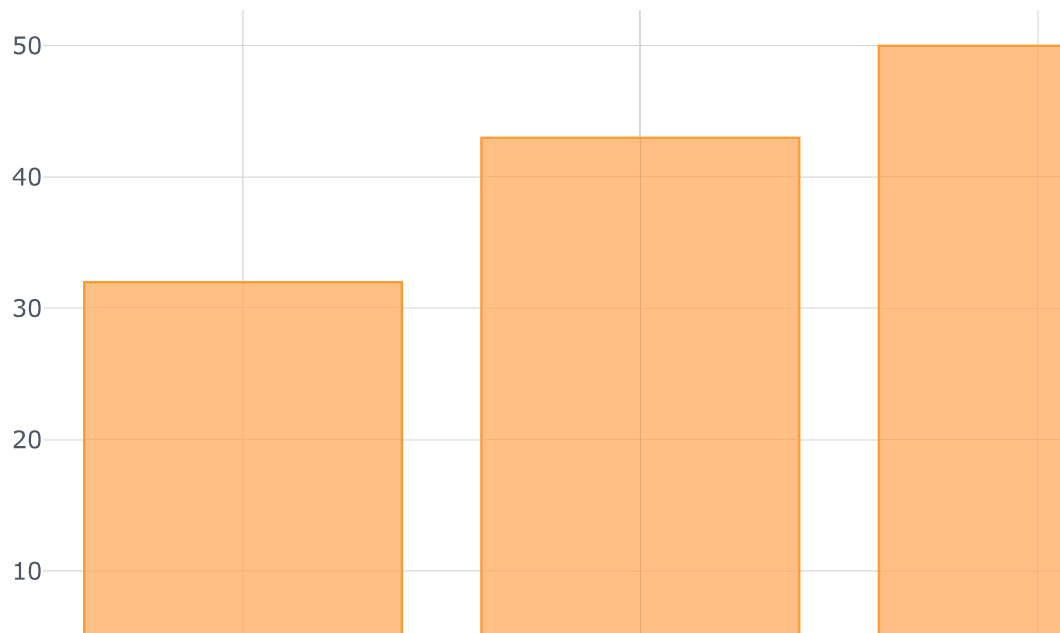


Bar Plot

In [16]:



```
df2.ipplot(kind='bar',x='items',y='Values')
```



In [17]:



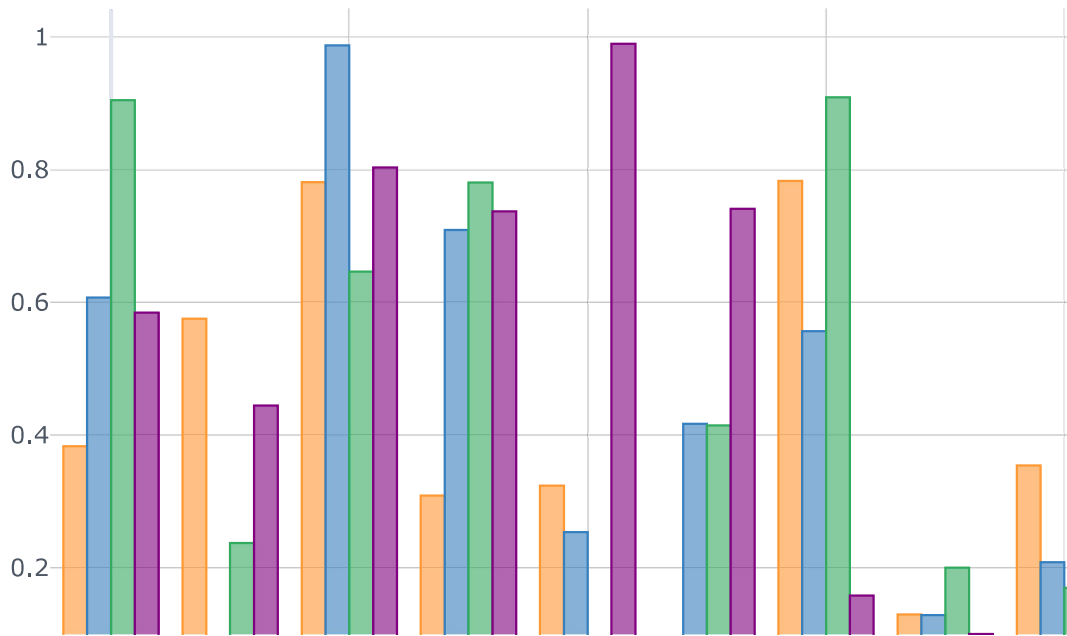
```
df = pd.DataFrame(np.random.rand(10,4),  
                  columns=['A', 'B', 'C', 'D'])  
df.head()
```

Out[17]:

	A	B	C	D
0	0.383038	0.607206	0.904592	0.584291
1	0.575204	0.038478	0.237307	0.444344
2	0.781147	0.986921	0.646131	0.802950
3	0.308719	0.709180	0.780318	0.736823
4	0.323907	0.253448	0.038868	0.989773

In [18]:

```
df.iplot(kind='bar')
```

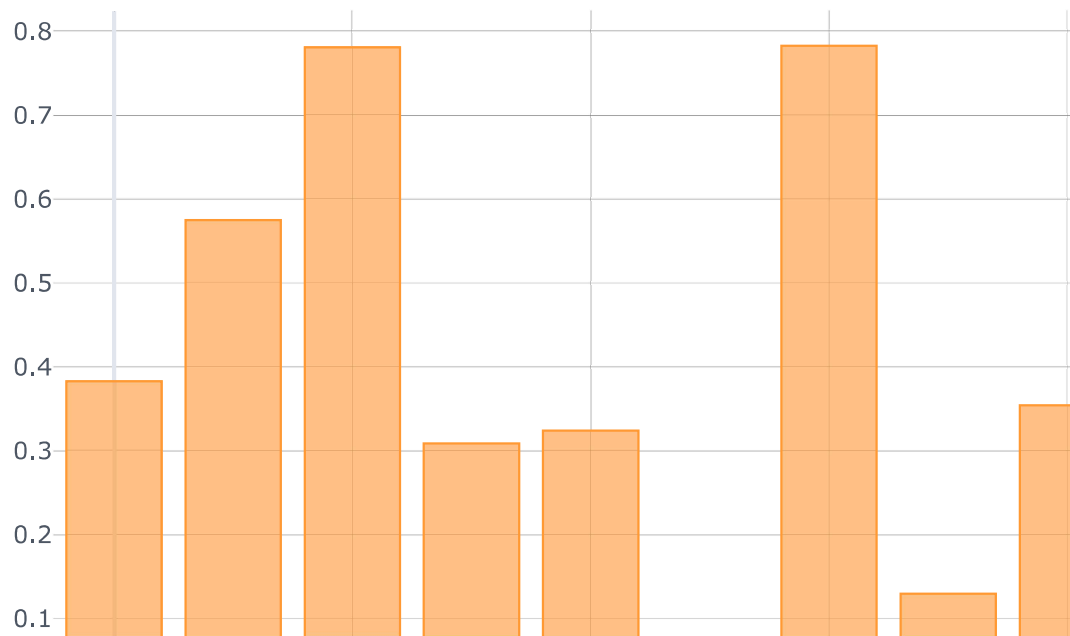


A컬럼만 보기

In [19]:



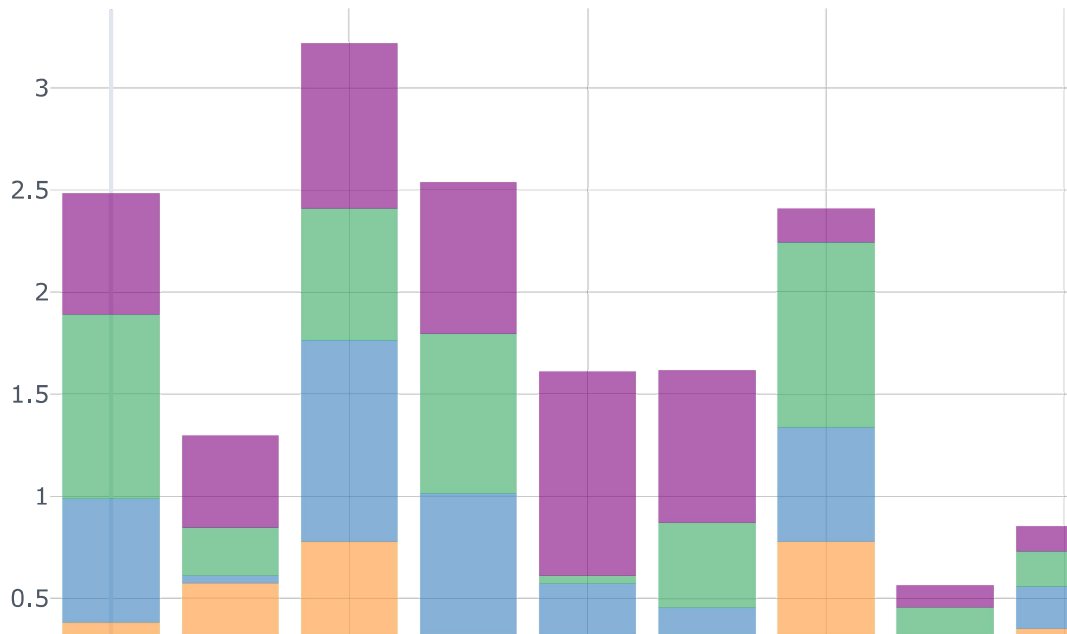
```
df['A'].iplot(kind='bar')
```



Stack plot

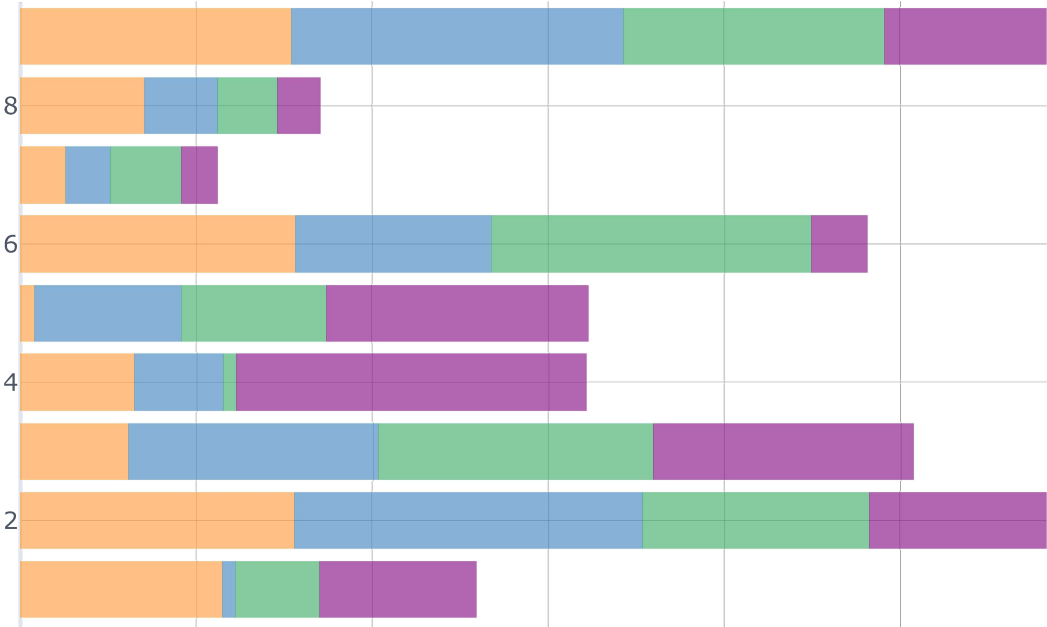
In [20]:

```
df.iplot(kind='bar', barmode='stack')
```



In [21]:

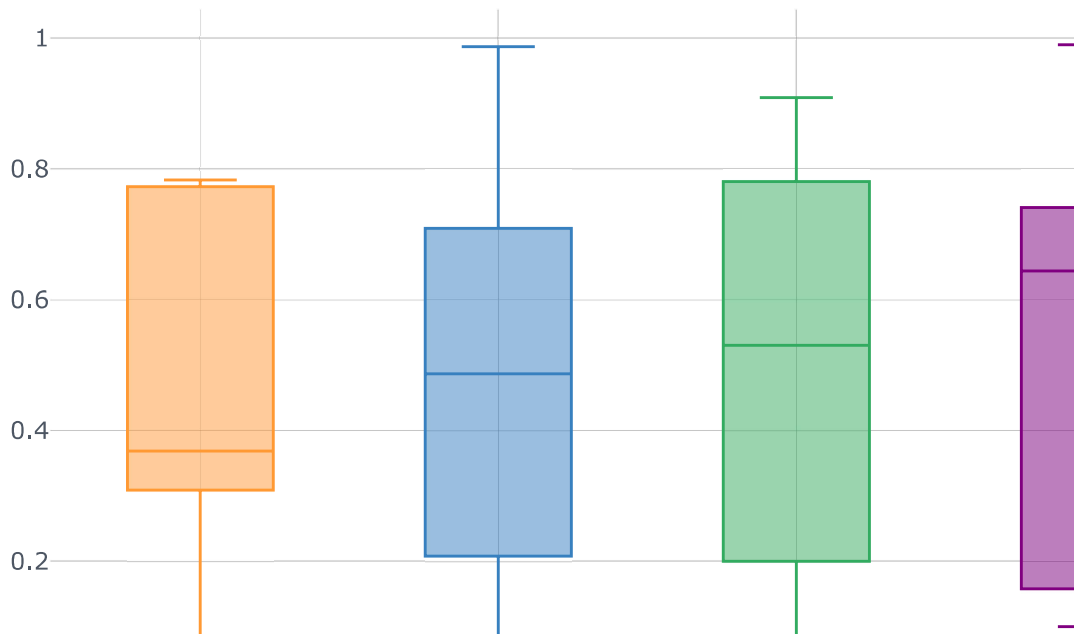
```
df.iplot(kind='barh', barmode='stack')
```



Boxplot

In [22]:

```
df.iplot(kind='box')
```



3D Surface Plot

In [23]:

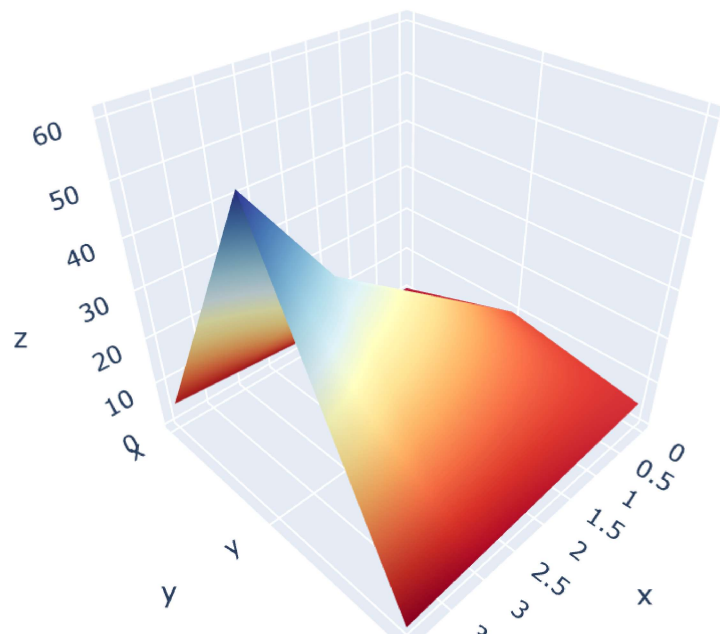
```
df3 = pd.DataFrame({'x':[1,2,3,4,5],  
                    'y':[10,20,30,40,60],  
                    'z':[5,4,3,2,1]})  
df3
```

Out[23]:

	x	y	z
0	1	10	5
1	2	20	4
2	3	30	3
3	4	40	2
4	5	60	1

In [24]:

```
df3.iplot(kind='surface',colorscale='rdylbu')
```



Line Charts

In [25]:

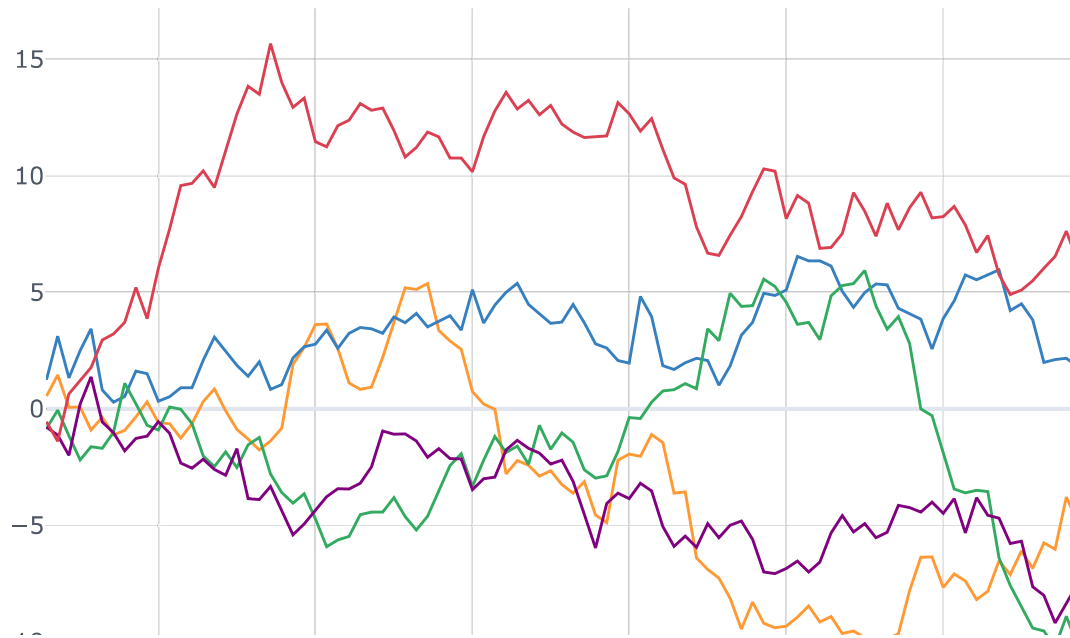
```
df = cf.datagen.lines()  
df.head()
```

Out[25]:

	OMM.HA	JEF.NY	MXE.CN	XVE.ZA	PJC.ON
2015-01-01	0.543922	1.253403	-0.833687	-0.778902	-0.547880
2015-01-02	1.459067	3.120556	-0.058942	-1.131742	-1.387056
2015-01-03	0.065940	1.329625	-1.153095	-1.995149	0.642025
2015-01-04	0.076844	2.498670	-2.193319	0.208328	1.194015
2015-01-05	-0.907959	3.441480	-1.628579	1.375014	1.775915

In [26]:

```
df.iplot(kind='line')
```



In [27]:



```
print(df.shape)
df.head(10)
```

(100, 5)

Out[27]:

	OMM.HA	JEF.NY	MXE.CN	XVE.ZA	PJC.ON
2015-01-01	0.543922	1.253403	-0.833687	-0.778902	-0.547880
2015-01-02	1.459067	3.120556	-0.058942	-1.131742	-1.387056
2015-01-03	0.065940	1.329625	-1.153095	-1.995149	0.642025
2015-01-04	0.076844	2.498670	-2.193319	0.208328	1.194015
2015-01-05	-0.907959	3.441480	-1.628579	1.375014	1.775915
2015-01-06	-0.356532	0.808727	-1.687369	-0.569445	2.952790
2015-01-07	-1.120472	0.284618	-0.998933	-1.024235	3.210097
2015-01-08	-0.925890	0.539543	1.095569	-1.805842	3.713102
2015-01-09	-0.352089	1.623625	0.212873	-1.266423	5.209424
2015-01-10	0.303970	1.509805	-0.704936	-1.167446	3.868988

테마설정

In [28]:



```
themes = cf.getThemes()
themes
```

Out[28]:

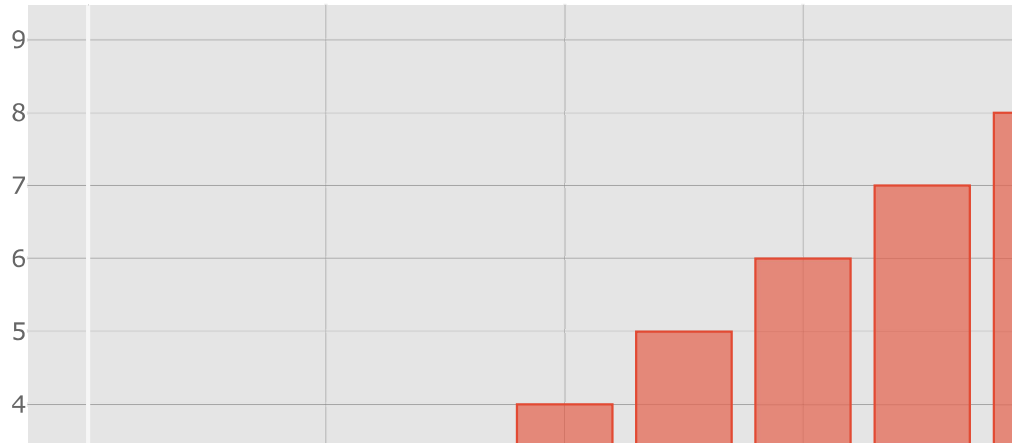
```
['ggplot', 'pearl', 'solar', 'space', 'white', 'polar', 'henanigans']
```

In [29]:

```
data = pd.Series(range(10))  
for theme in themes:  
    data.iplot(kind='bar', theme=theme, title=theme)
```



ggplot



REF

- cufflinks.datagen module
- <https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html>
(<https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html>)
(<https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html>)
(<https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html>)
- Plotly Express in Python
- <https://plot.ly/python/plotly-express/#plotly-express> (<https://plot.ly/python/plotly-express/#plotly-express>)
(<https://plot.ly/python/plotly-express/#plotly-express>) (<https://plot.ly/python/plotly-express/#plotly-express>)