## Plotly 소개 및 실습

d3.js를 이용하여 interactive하게 그래프를 보여준다.

## 학습내용

- Plotly 를 실습을 통해 알아본다

## 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

## cuffilinks 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 [23]:

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

```
['C:WWUsersWWUSER100WWDocumentsWWGitHubWWAI_enovationWWpart02_libraryWWCL02_05_02_pl
otly', 'C:WWProgramDataWWAnaconda3WWpython38.zip', 'C:WWProgramDataWWAnaconda3WWDLL
s', 'C:WWProgramDataWWAnaconda3WWlib', 'C:WWProgramDataWWAnaconda3', '', 'C:WWProgra
mDataWWAnaconda3WWlibWWsite-packages', 'C:WWProgramDataWWAnaconda3WWlibWWsite-packag
esWWwin32', 'C:WWProgramDataWWAnaconda3WWlibWWsite-packagesWWwin32WWlib', 'C:WWProgr
amDataWWAnaconda3WWlibWWsite-packagesWWPythonwin', 'C:WWProgramDataWWAnaconda3WWlib
WWsite-packagesWWIPythonWWextensions', 'C:WWUsersWWUSER100WW.ipython']
```

In [24]:

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

## 버전 확인

In [25]:

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

```
4.13.0
0.17.3
1.0.5
1.18.5
```

In [26]:

```
# 오프라인 모드에서도 인터렉티브한 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 [27]:

```
# 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[27]:

| | A | B | C | D |
|---|---|---|---|---|
| **0** | 0.326187 | 0.276691 | 0.576640 | 0.749804 |
| **1** | 0.633625 | -0.315355 | 0.544542 | 0.691714 |
| **2** | -0.365359 | -1.478201 | 1.122758 | -0.622982 |
| **3** | -0.979001 | -0.921833 | 1.124664 | 0.412570 |
| **4** | -0.944191 | 1.924357 | 0.731014 | 0.131426 |

In [28]:

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

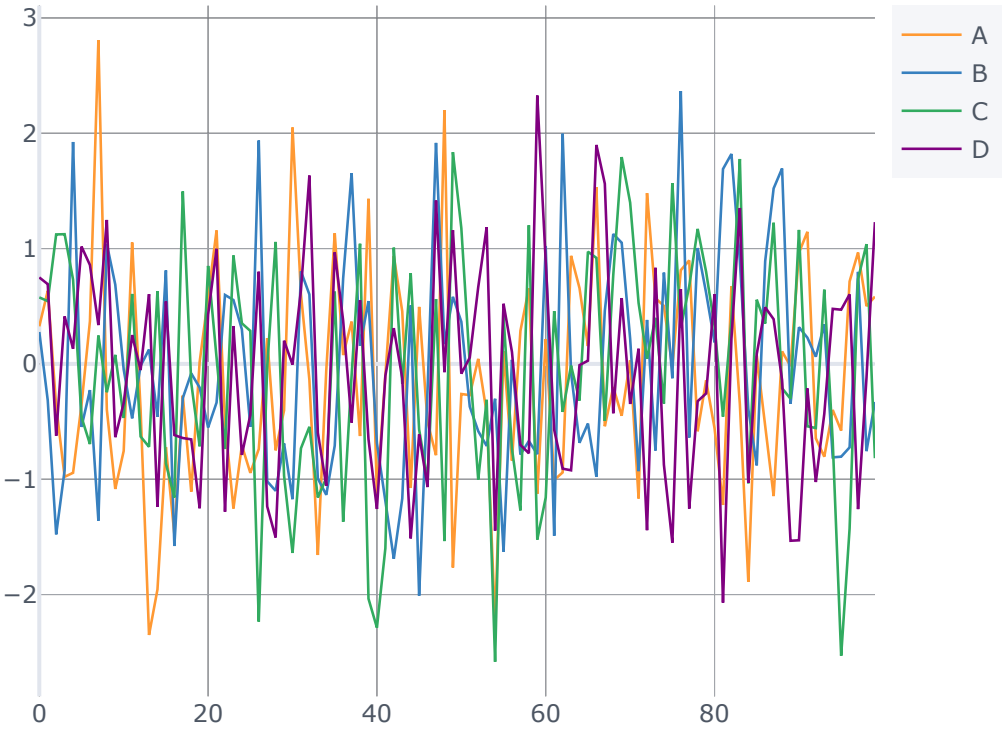
Out[28]:

| | items | Values |
|---|---|---|
| **0** | bag | 32 |
| **1** | apple | 43 |
| **2** | cap | 50 |

## Line Plot

In [29]:

```
df.iplot()
```

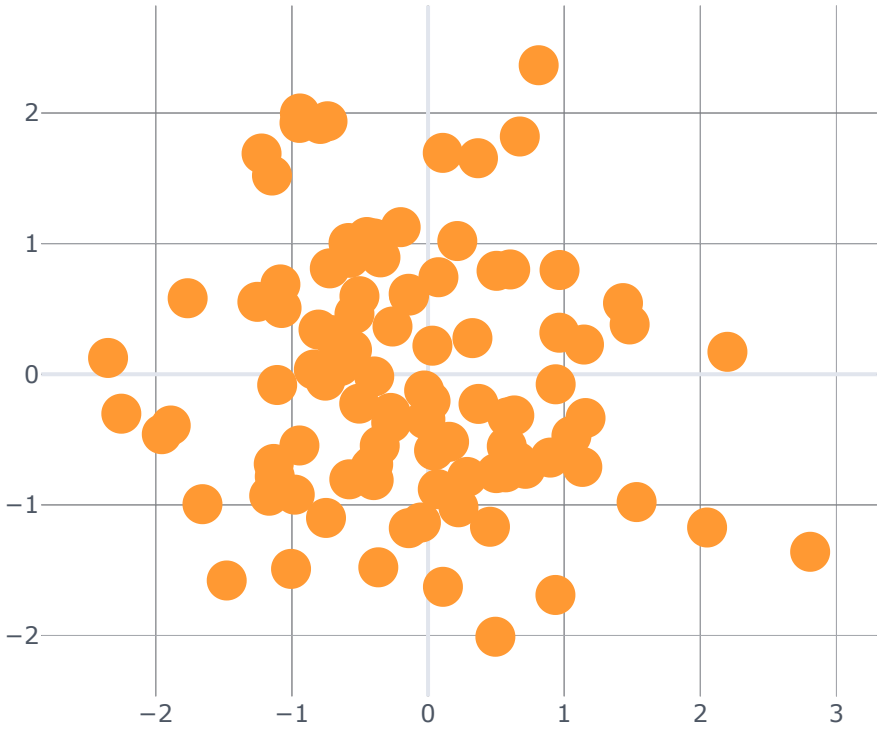



Export to plot.ly »

## Scatter Plot

In [30]:

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

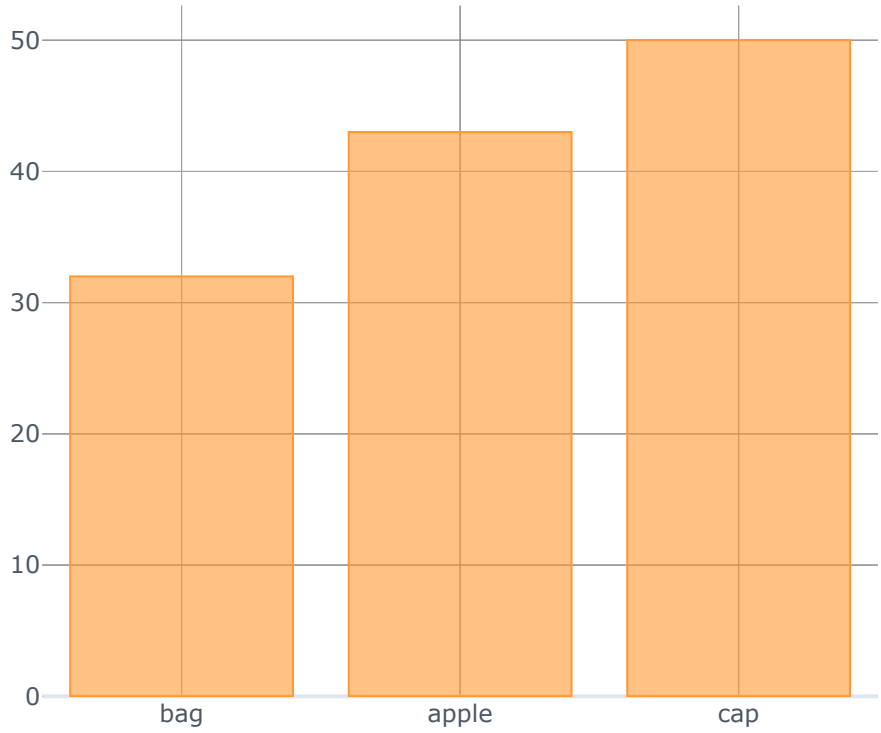



Export to plot.ly »

## Bar Plot

In [31]:

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




Export to plot.ly »

In [32]:

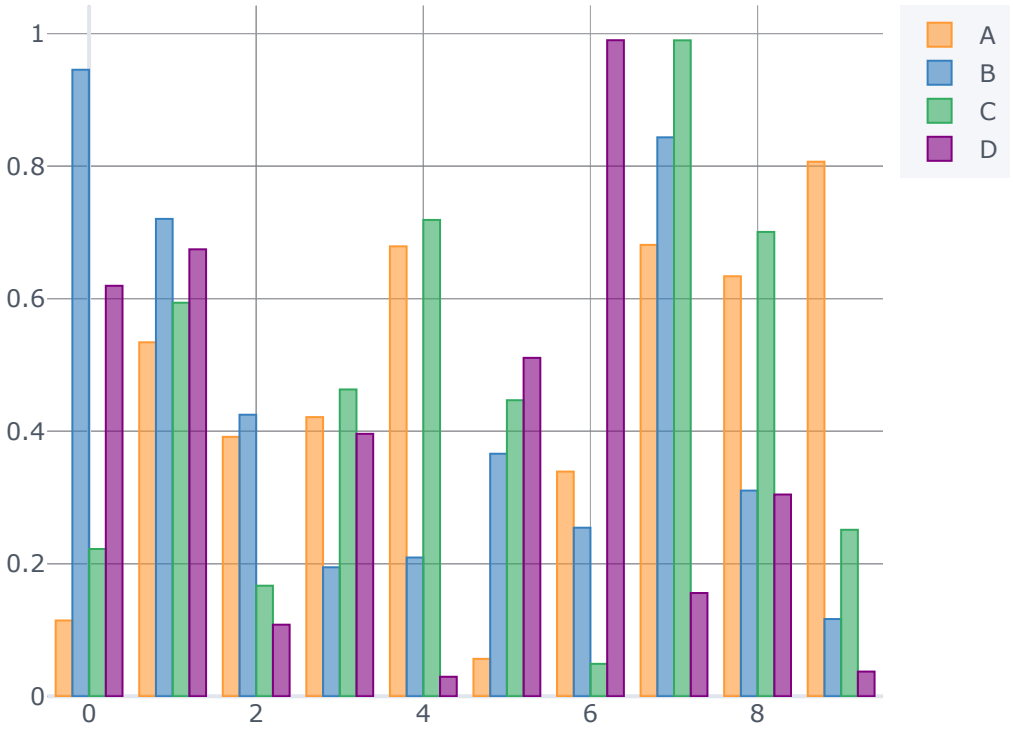
```
df = pd.DataFrame(np.random.rand(10,4),
                  columns=['A', 'B', 'C', 'D'])
df.head()
```

Out[32]:

| | A | B | C | D |
|---|---|---|---|---|
| **0** | 0.114402 | 0.945480 | 0.222283 | 0.619454 |
| **1** | 0.534173 | 0.720421 | 0.593897 | 0.674504 |
| **2** | 0.391551 | 0.424891 | 0.166907 | 0.108045 |
| **3** | 0.421303 | 0.194691 | 0.463038 | 0.396068 |
| **4** | 0.679001 | 0.209430 | 0.718828 | 0.029480 |

In [33]:

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

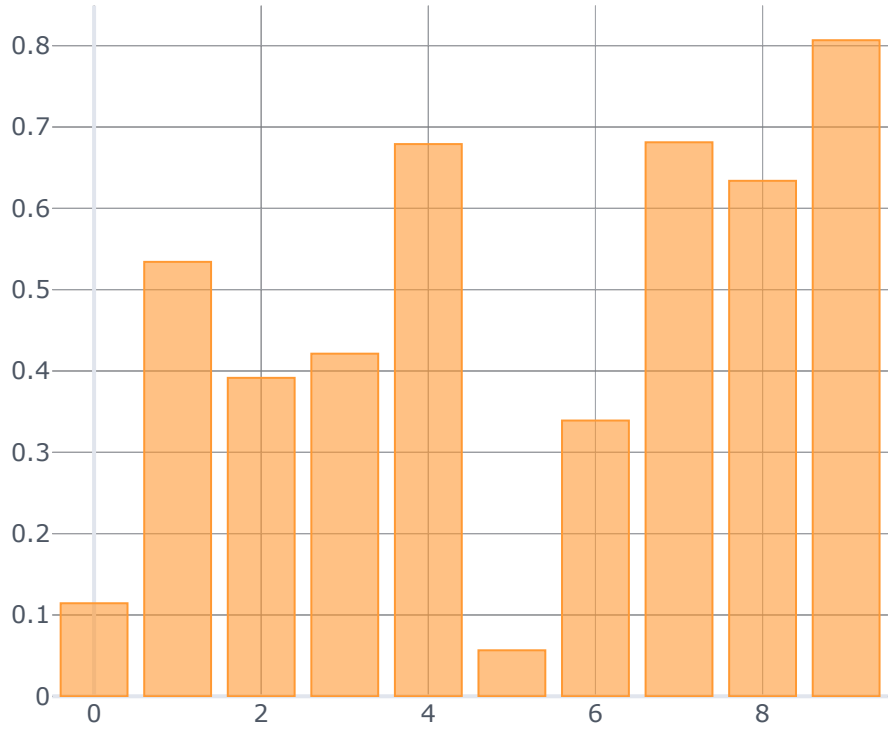



Export to plot.ly »

## A컬럼만 보기

In [34]:

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

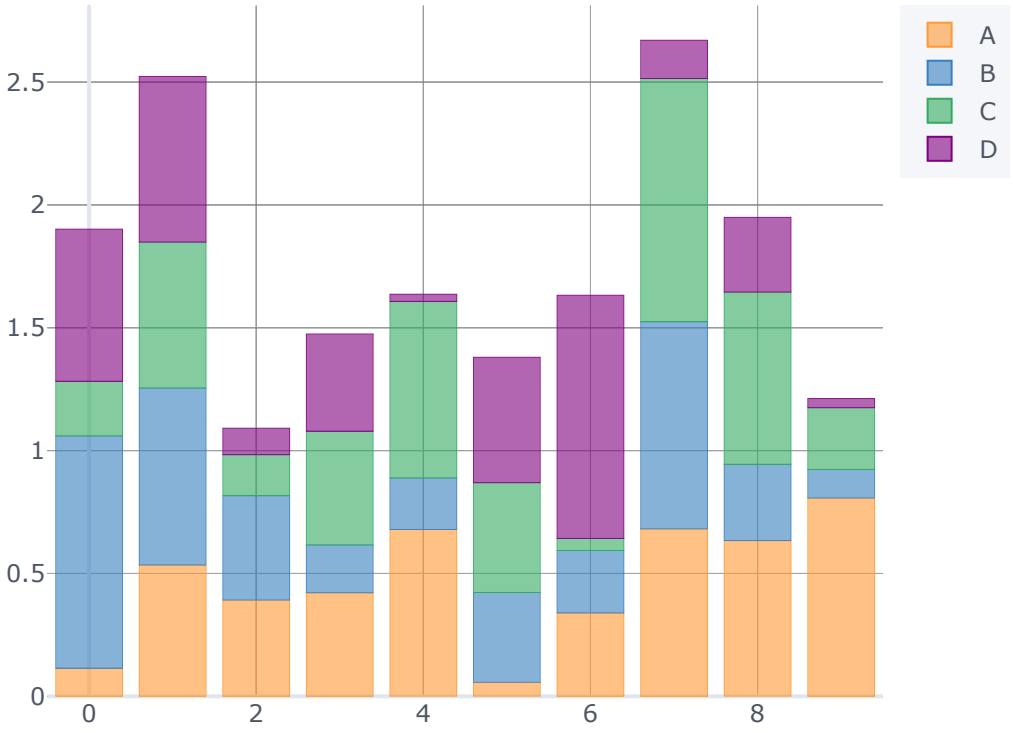



Export to plot.ly »

## Stack plot

In [35]:

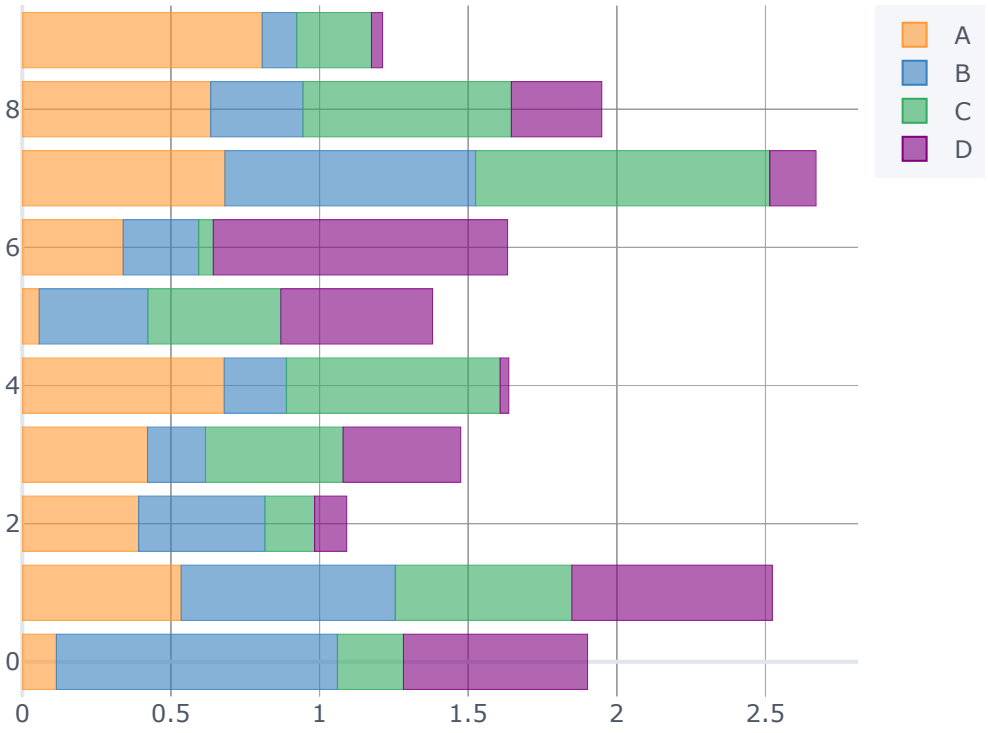
```
df.iplot(kind='bar', barmode='stack')
```




Export to plot.ly »

In [36]:

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

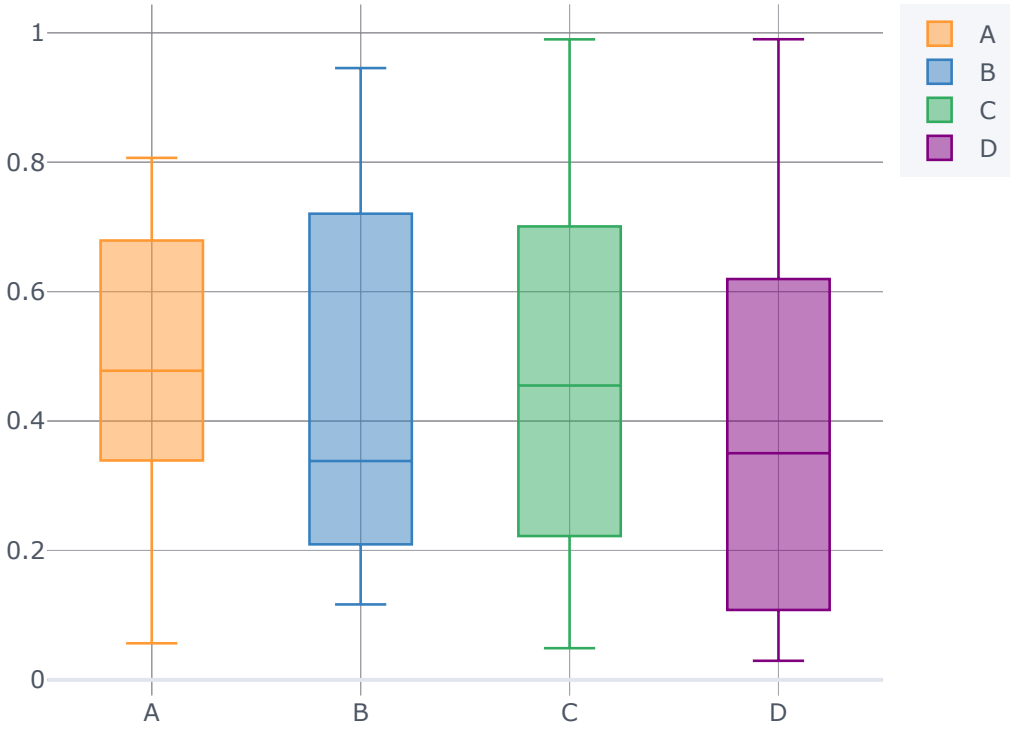



Export to plot.ly »

## Boxplot

In [37]:

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




Export to plot.ly »

## 3D Surface Plot

In [38]:

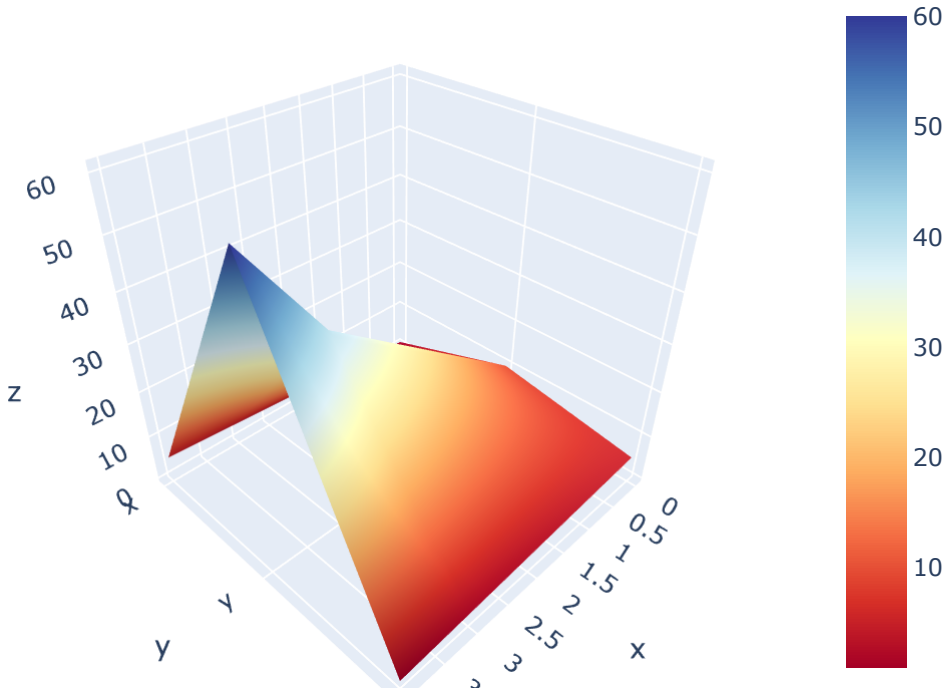
```
df3 = pd.DataFrame({'x':[1,2,3,4,5],
 'y':[10,20,30,40,60],
 'z':[5,4,3,2,1]})
df3
```

Out[38]:

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

In [39]:

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




Export to plot.ly »

## Line Charts

In [40]:

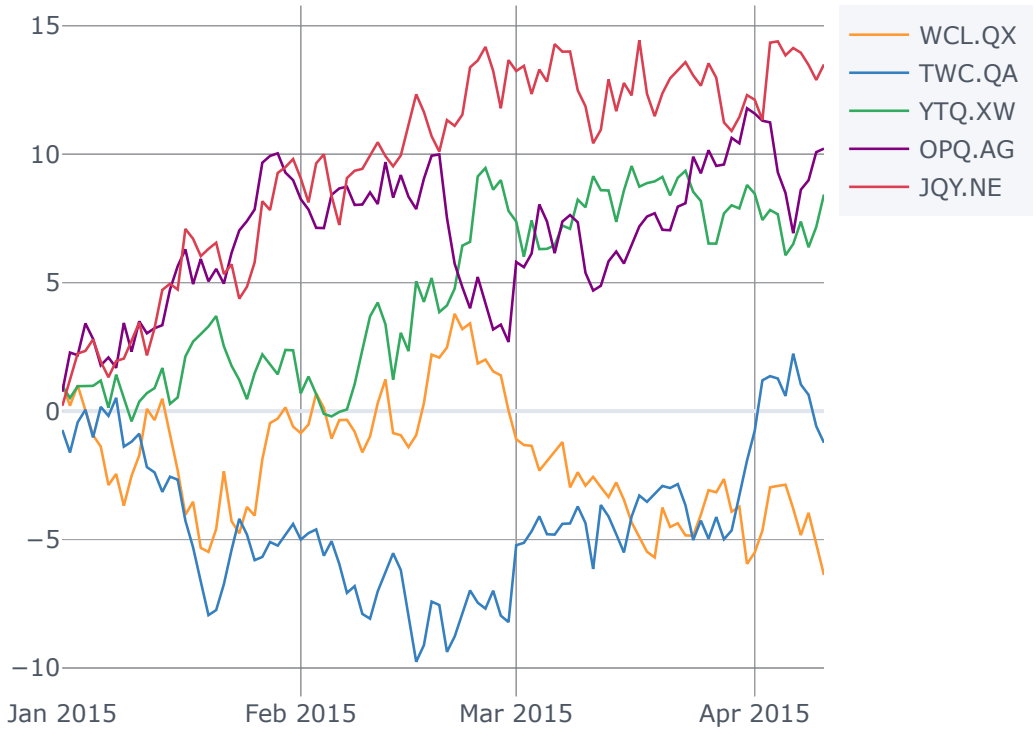
```
df = cf.datagen.lines()
df.head()
```

Out[40]:

| | WCL.QX | TWC.QA | YTQ.XW | OPQ.AG | JQY.NE |
|---|---|---|---|---|---|
| **2015-01-01** | 1.045689 | -0.733903 | 0.973287 | 0.751970 | 0.206542 |
| **2015-01-02** | 0.209828 | -1.610302 | 0.511602 | 2.278077 | 1.132667 |
| **2015-01-03** | 1.003587 | -0.435849 | 0.968907 | 2.175889 | 2.243116 |
| **2015-01-04** | 0.059543 | 0.052410 | 0.969702 | 3.412766 | 2.343262 |
| **2015-01-05** | -0.934334 | -1.024240 | 0.982690 | 2.811208 | 2.790090 |

In [41]:

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




Export to plot.ly »

In [42]:

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

```
(100, 5)
```

Out[42]:

| | WCL.QX | TWC.QA | YTQ.XW | OPQ.AG | JQY.NE |
|---|---|---|---|---|---|
| **2015-01-01** | 1.045689 | -0.733903 | 0.973287 | 0.751970 | 0.206542 |
| **2015-01-02** | 0.209828 | -1.610302 | 0.511602 | 2.278077 | 1.132667 |
| **2015-01-03** | 1.003587 | -0.435849 | 0.968907 | 2.175889 | 2.243116 |
| **2015-01-04** | 0.059543 | 0.052410 | 0.969702 | 3.412766 | 2.343262 |
| **2015-01-05** | -0.934334 | -1.024240 | 0.982690 | 2.811208 | 2.790090 |
| **2015-01-06** | -1.382608 | 0.164858 | 1.189331 | 1.780859 | 1.928705 |
| **2015-01-07** | -2.879521 | -0.186790 | 0.133336 | 2.089829 | 1.317749 |
| **2015-01-08** | -2.448178 | 0.516156 | 1.419264 | 1.683909 | 1.948982 |
| **2015-01-09** | -3.681210 | -1.383030 | 0.576473 | 3.430753 | 2.044961 |
| **2015-01-10** | -2.520215 | -1.192299 | -0.401448 | 2.313352 | 2.738623 |

## 테마설정

In [43]:

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

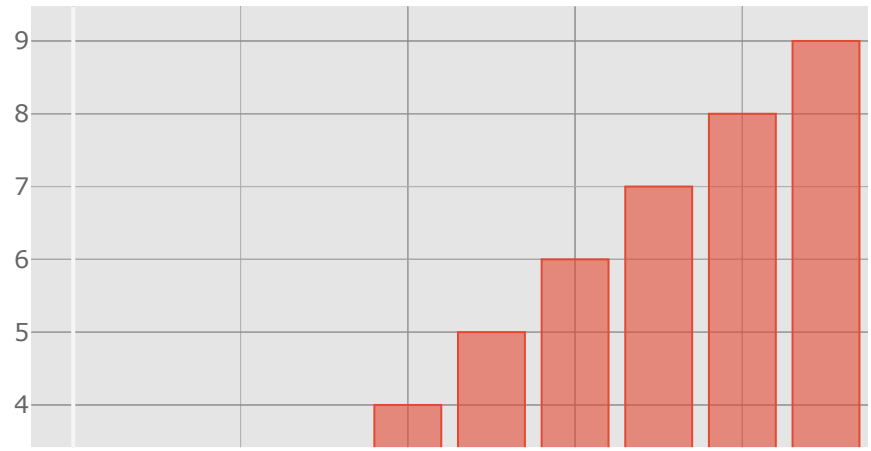
Out[43]:

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

In [44]:

```
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))