plotly를 활용한 타이타닉 데이터 시각화

In [1]:

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

In [2]:

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

In [3]:

```
train = pd.read_csv("./titanic/train.csv")
test = pd.read_csv("./titanic/train.csv")

#train = pd.read_csv("train.csv")
#test = pd.read_csv("train.csv")
```

In [4]:

```
train.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	

In [ ]:

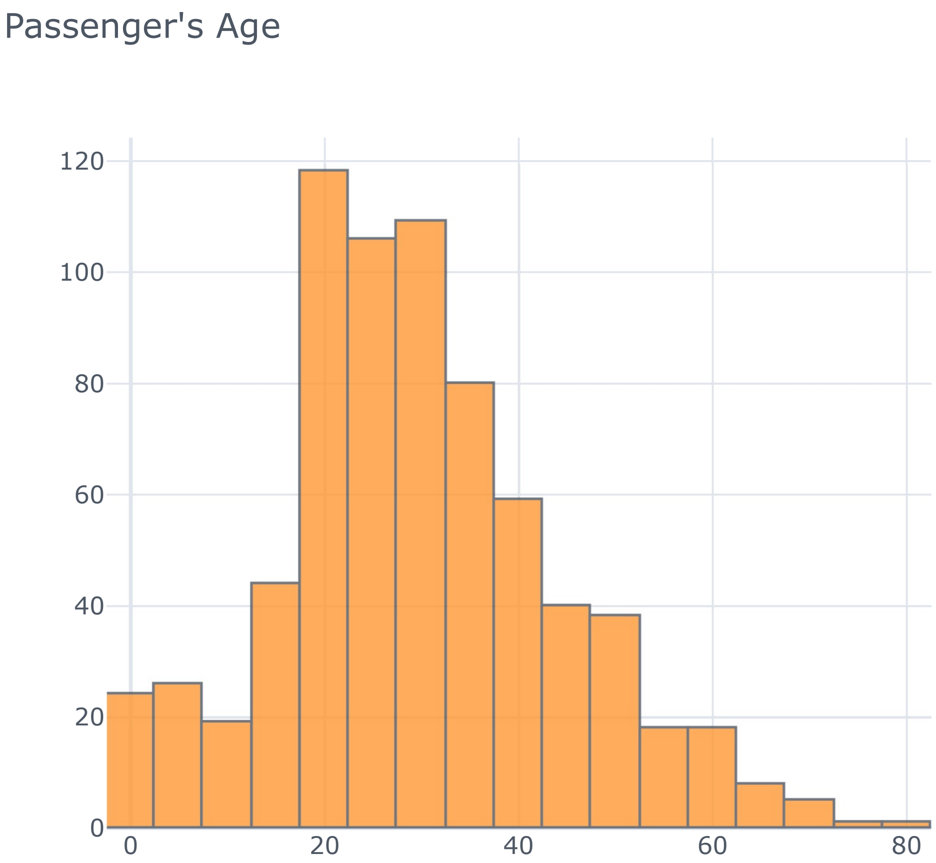
In [8]:

```
train.iplot(kind='bar', x='Pclass', y='Survived')
```

- Pclass는 1,2,3의 값이 있다.

In [9]:

```
train['Age'].iplot(kind='histogram', bins=20,
                   title="Passenger's Age")
```



In [10]:

```
train['Fare'].iplot(kind='histogram', bins=20,
                   title="Passenger's Age")
```

클래스별 요금 히스토그램

In [11]:

```
train['Fare_1'] = train[ train['Pclass']==1 ]['Fare']
train['Fare_2'] = train[ train['Pclass']==2 ]['Fare']
train['Fare_3'] = train[ train['Pclass']==3 ]['Fare']
```

In [12]:

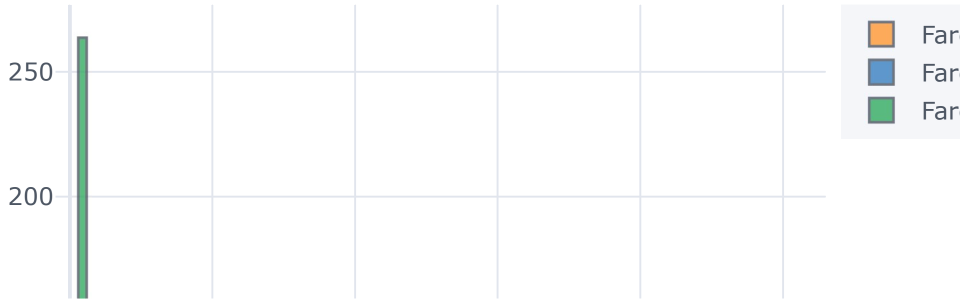
```
train.columns
```

Out[12]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp
',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked', 'Fare_1', 'Fare_2
',
       'Fare_3'],
      dtype='object')
```

In [14]:

```
sel = ['Fare_1', 'Fare_2', 'Fare_3']
train[ sel ].iplot(kind='histogram',
                   bins=30,
                   xTitle='Fare',
                   yTitle='Count')
```



Heatmap

In [15]:

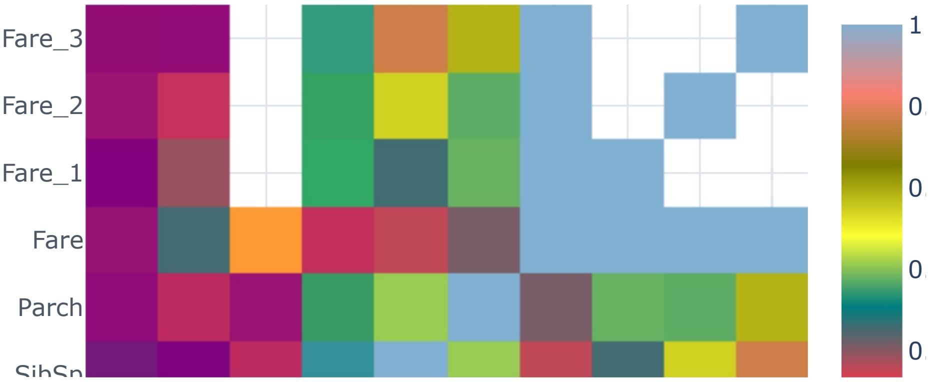
```
train.corr()
```

Out[15]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	F
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000
Fare_1	-0.026348	0.190966	NaN	-0.218611	0.259492	0.380963	1.000
Fare_2	0.021281	0.098628	NaN	-0.197038	0.546093	0.373920	1.000
Fare_3	0.002053	0.000930	NaN	-0.260315	0.766003	0.585265	1.000

In [16]:

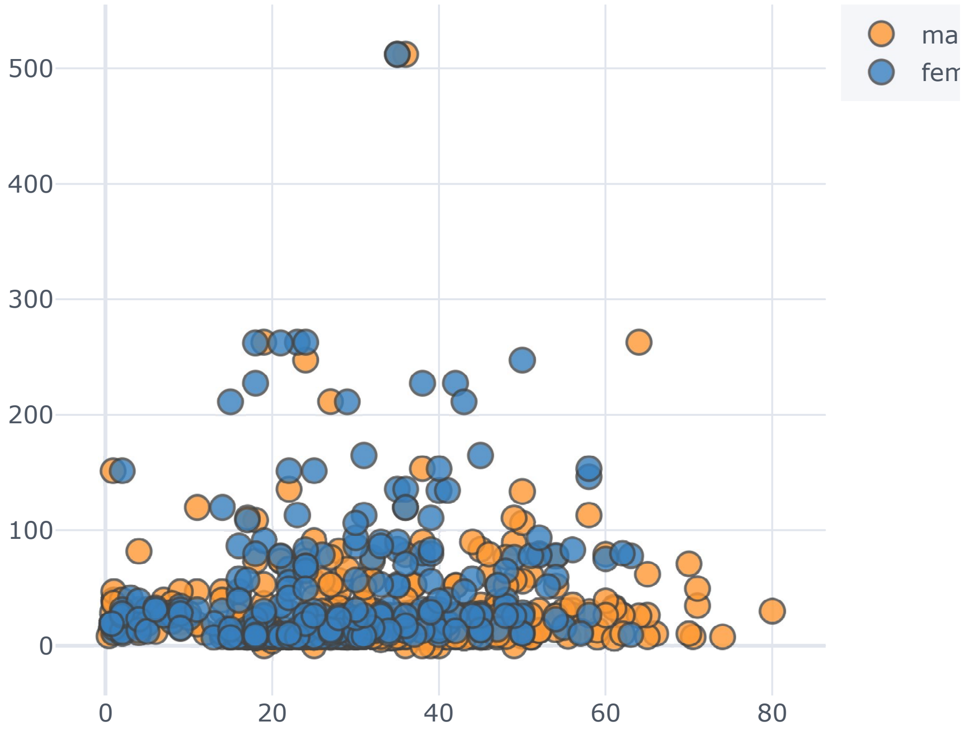
```
train.corr().iplot(kind='heatmap')
```



요금과 나이의 관계 보기

In [21]:

```
train.iplot(kind='scatter',
            x = 'Age',
            y = 'Fare',
            categories='Sex')
```



In [ ]: