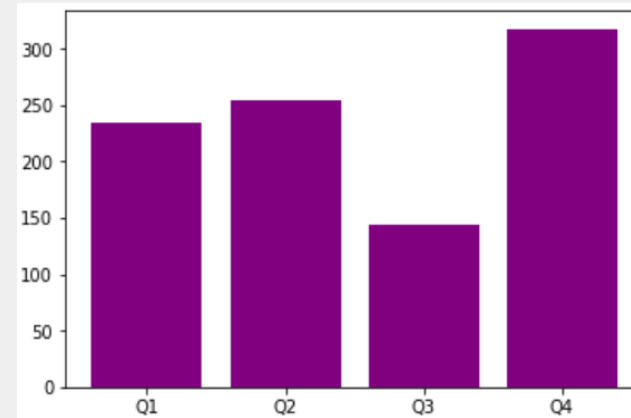


데이터 시각화

Matplotlib 패키지 : 바차트

Bar chart:

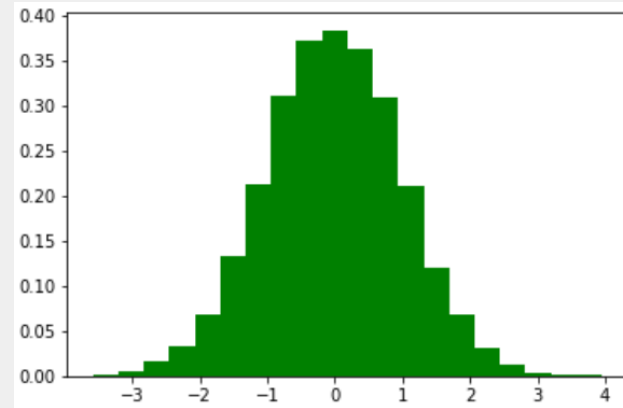
```
In[1] : import matplotlib.pyplot as plt  
import numpy as np  
%matplotlib inline  
x = np.array(['Q1', 'Q2', 'Q3','Q4'])  
y = np.array([ 234.0, 254.7, 144.6, 317.6])  
plt.bar(x,y,color = 'purple')  
plt.show()
```



Matplotlib 패키지 : 히스토그램

Histogram:

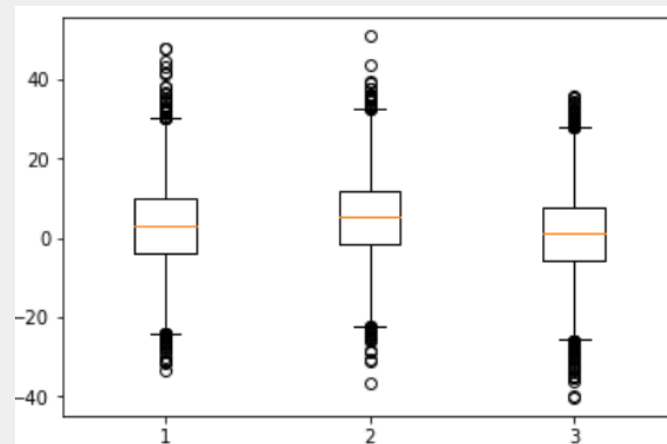
```
In[1] : x = np.random.randn(10000)
plt.hist(x,bins=20,color='green', density=True)
plt.show()
```



Matplotlib 패키지 : 박스플롯

Box plot:

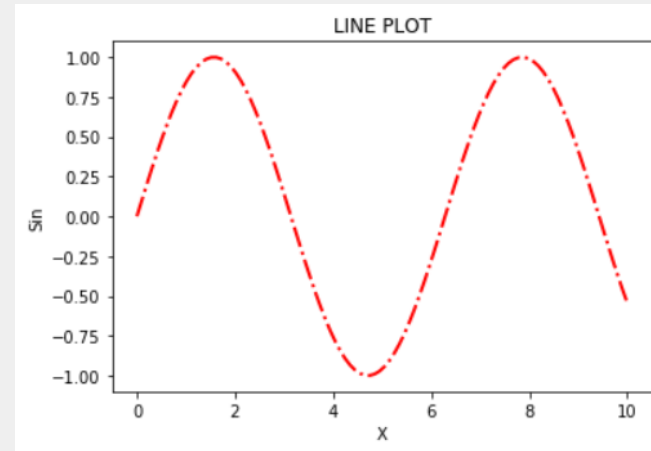
```
In[1] : x = np.random.randn(10000)*10+3  
        y = np.random.randn(10000)*10+5  
        z = np.random.randn(10000)*10+1  
        plt.boxplot([x, y, z], 0)  
        plt.show()
```



Matplotlib 패키지 : 라인

Line plot:

```
In[1] : x = np.linspace(0,10,100)
        y = np.sin(x)
        plt.plot(x,y,color='red',linestyle='-.',linewidth=2)
        plt.xlabel('X')
        plt.ylabel('Sin')
        plt.title('LINE PLOT')
        plt.show()
```



Matplotlib 패키지 : plot 메서드의 인자

plot 메서드의 인자:

인자	설명
color	컬러.
alpha	투명도.
linewidth	선 굵기.
linestyle	선 스타일.
marker	마커 타입.
markersize	마커 크기.
markerfacecolor	마커 내부 컬러.
markeredgecolor	마커 테두리선 컬러.
markeredgewidth	마커 테두리선 굵기.

Matplotlib 패키지 : 라인스타일

라인스타일 유형:

linestyle	설명
none	라인 없음.
:	점선.
--	단속 선.
-.	선과 점.
-	연속선.
steps	스텝.

Matplotlib 패키지 : 마커 타입

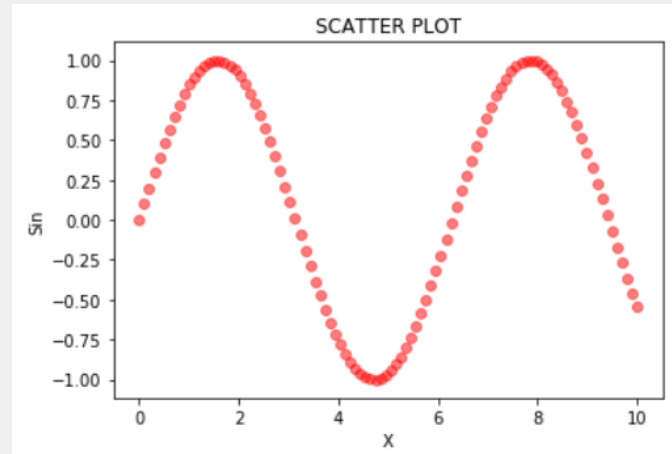
마커 타입:

marker	설명
.	점.
,	픽셀.
o	원.
^	삼각형.
v	역삼각형.
s	사각형.
*	별표.
+	플러스.
x	엑스.
D	다이아몬드 (마름모).
p	정오각형.

Matplotlib 패키지 : 산점도

Scatter plot:

```
In[1] : x = np.linspace(0,10,100)
        y = np.sin(x)
        plt.scatter(x,y,c='red',marker='o',alpha=0.5)
        plt.xlabel('X')
        plt.ylabel('Sin')
        plt.title('SCATTER PLOT')
        plt.show()
```



Matplotlib 패키지 : 객체 지향형 시각화

객체 지향형 시각화:

```
In[1] : fig=plt.figure()
```

```
axes = fig.add_axes([0,0,1,1])
```

Left, Bottom, Width, Height

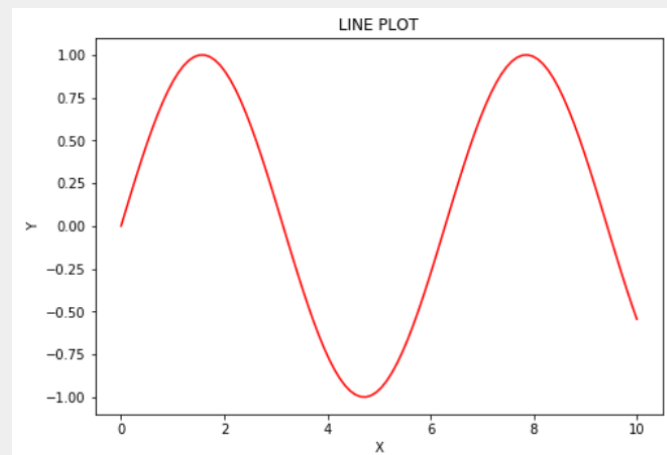
```
axes.plot(x,y,color='red',linestyle='-')
```

```
axes.set_xlabel('X')
```

```
axes.set_ylabel('Y')
```

```
axes.set_title('LINE PLOT')
```

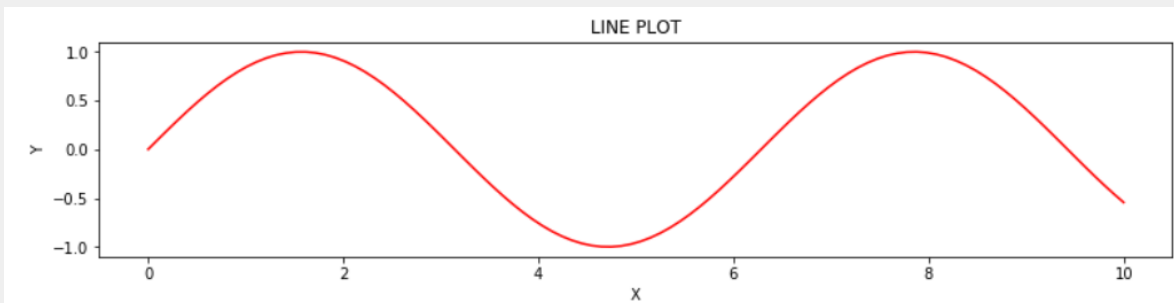
```
plt.show()
```



Matplotlib 패키지 : 객체 지향형 시각화

figsize 인자를 통한 비율 조정:

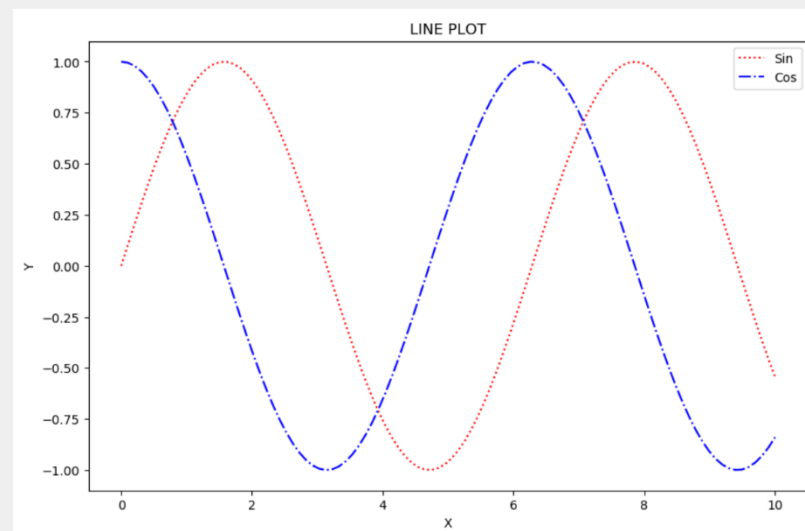
```
In[1] : fig=plt.figure(figsize=(10,2))      # 가로, 세로 크기.  
        axes = fig.add_axes([0,0,1,1])      # Left, Bottom, Width, Height  
        axes.plot(x,y,color='red',linestyle='-')  
        axes.set_xlabel('X')  
        axes.set_ylabel('Y')  
        axes.set_title('LINE PLOT')  
        plt.show()
```



Matplotlib 패키지 : 객체 지향형 시각화

하나의 패널에 다중 시각화 (범례 사용):

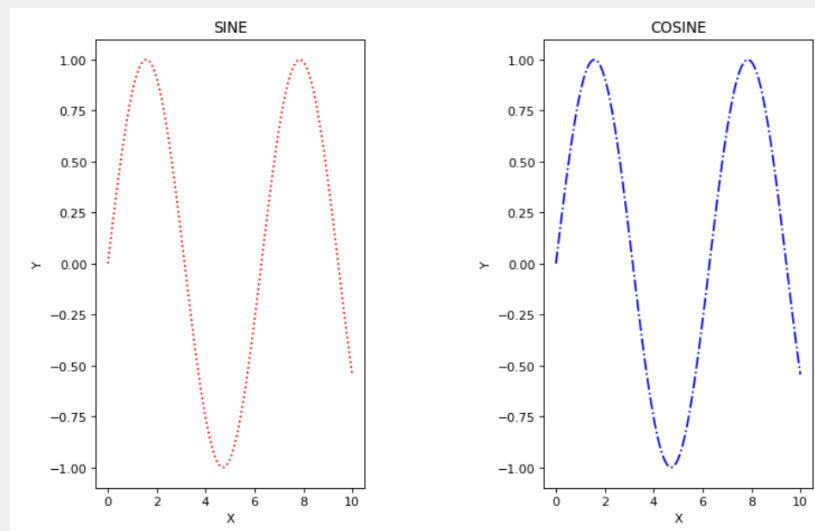
```
In[1] : fig=plt.figure(figsize=(8,5), dpi=100)           # 가로, 세로 크기. DPI 세팅.  
        axes = fig.add_axes([0,0,1,1])                  # Left, Bottom, Width, Height  
        axes.plot(x,y,color='red',linestyle=':', label='Sin')      # 범례를 위한 label 인자.  
        axes.plot(x,z,color='blue',linestyle='-.', label='Cos')    # 범례를 위한 label 인자.  
        axes.legend(loc=0)                                     # Top-Right에 범례 출력.  
        axes.set_xlabel('X')  
        axes.set_ylabel('Y')  
        axes.set_title('LINE PLOT')  
        plt.show()
```



Matplotlib 패키지 : 객체 지향형 시각화

여러개의 패널이 있는 다중 시각화 1:

```
In[1] : fig=plt.figure(figsize=(10,5), dpi=80)      # 가로, 세로 크기. DPI 세팅.  
        axes1 = fig.add_axes([0,0,0.3,1])           # Left, Bottom, Width, Height  
        axes2 = fig.add_axes([0.5,0,0.3,1])         # Left, Bottom, Width, Height  
        axes1.plot(x,y,color='red',linestyle=':')  
        axes2.plot(x,y,color='blue',linestyle='-.')  
        axes1.set_xlabel('X')  
        axes1.set_ylabel('Y')  
        axes1.set_title('SINE')  
        axes2.set_xlabel('X')  
        axes2.set_ylabel('Y')  
        axes2.set_title('COSINE')  
        plt.show()
```



Matplotlib 패키지 : 객체 지향형 시각화

여러개의 패널이 있는 다중 시각화 2:

```
In[1] : fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10,10))
```

```
# (0,0)
```

```
axes[0,0].plot(x,y,color='green',linestyle=':')
```

```
axes[0,0].set_xlabel('X')
```

```
axes[0,0].set_ylabel('Y')
```

```
axes[0,0].set_title('SINE')
```

```
# (0,1)
```

```
⋮
```

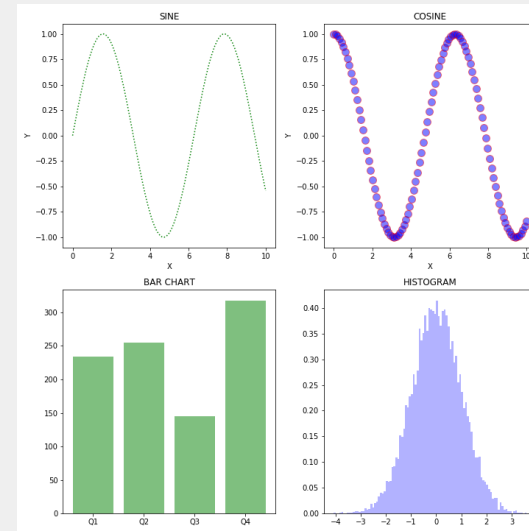
```
⋮
```

```
⋮
```

```
⋮
```

```
plt.tight_layout()
```

```
plt.show()
```



서로 겹치는 현상을 피함.

Pandas 패키지 : 시각화

Pandas 데이터 프레임: 시각화

```
In[1] : df.plot(x = 'A' , y = 'B' )
```

```
In[2] : df.plot.scatter(x = 'A' , y = 'B' )
```

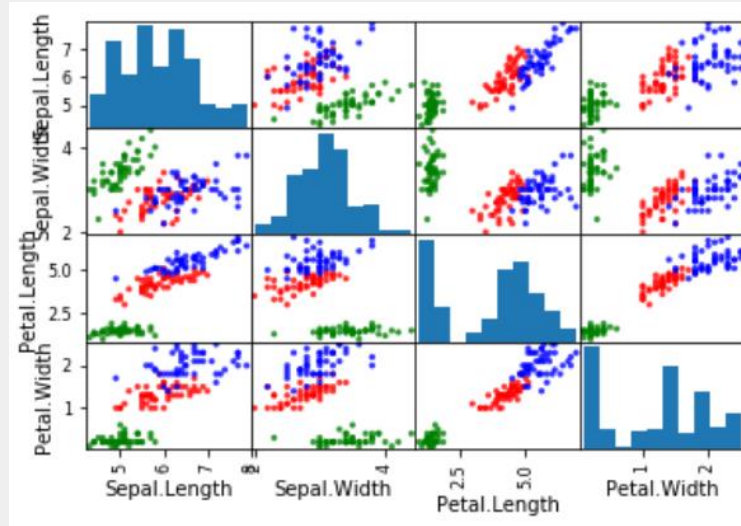
```
In[3] : my_cols_dict = {'setosa':'red', 'virginica':'green', 'versicolor':'blue'}  
my_cols = df0['Species'].apply(lambda x: my_cols_dict[x])  
pd.plotting.scatter_matrix(df, c=my_cols, marker='o', alpha=0.5)  
plt.show()
```

A 대 B 라인 플롯.

A 대 B 산점도.

Species 유형을 컬러로 **번역**.

산점도 행렬.



Seaborn 패키지

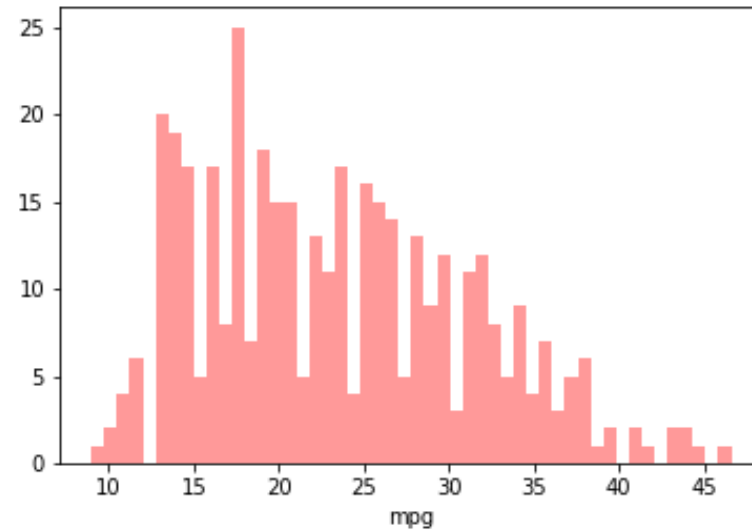
특장점:

- 내장 데이터를 제공한다: `load_dataset`.
- 기본 시각화 유형: `distplot`, `jointplot`, `kdeplot`, `rugplot`, `barplot`, `countplot`, 등.
- 다중 행렬 시각화 유형: `pairplot`, `PairGrid`, `FacetGrid`, 등.
- 회귀선 추가 기능: `lmplot`, `jointplot`, 등.
- 2D 특수 시각화: `heatmap`, `clustermap`, 등.
- 기본 시각화의 변형: `violinplot`, `swarmplot`, `stripplot`, 등.

Seaborn 패키지 : 히스토그램

히스토그램:

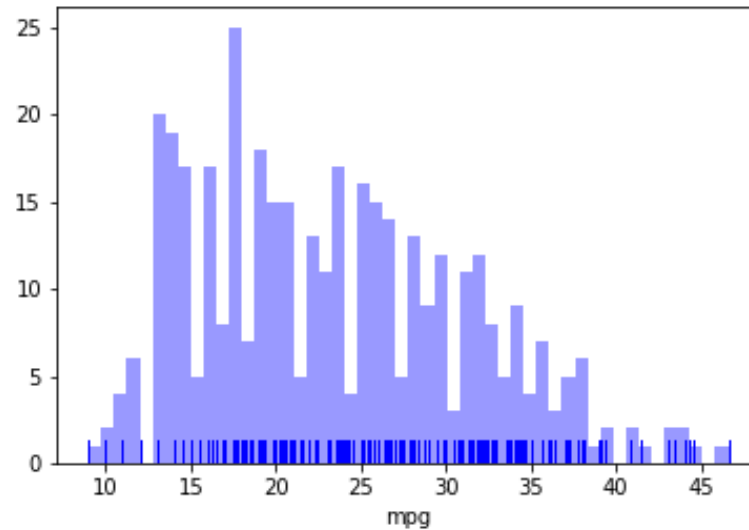
```
sns.distplot(dat.mpg, kde=False, rug=False, bins=50, color='red')  
plt.show()
```



Seaborn 패키지 : 히스토그램 + Rug

히스토그램 + Rug:

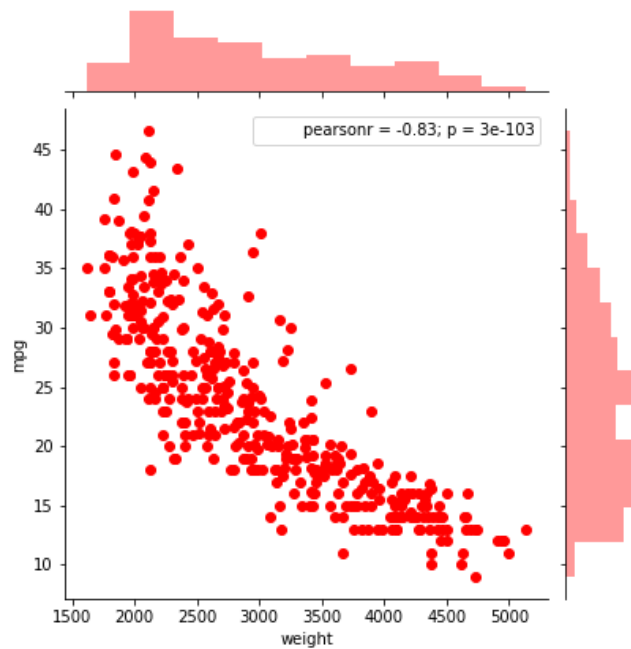
```
sns.distplot(dat.mpg, kde=False, rug=True, bins=50, color='blue')  
plt.show()
```



Seaborn 패키지 : 산점도

산점도:

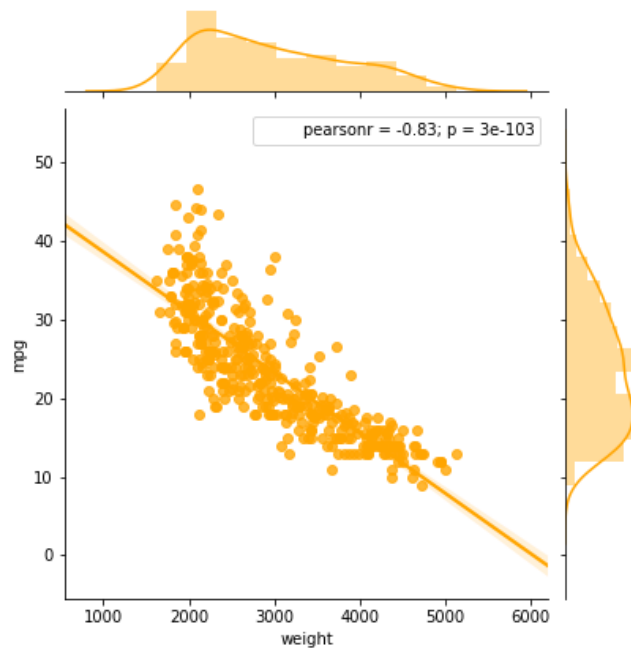
```
sns.jointplot(x='weight', y='mpg', data=dat, color='red', kind='scatter' )  
plt.show()
```



Seaborn 패키지 : 산점도 + 회귀선

산점도 + 회귀선:

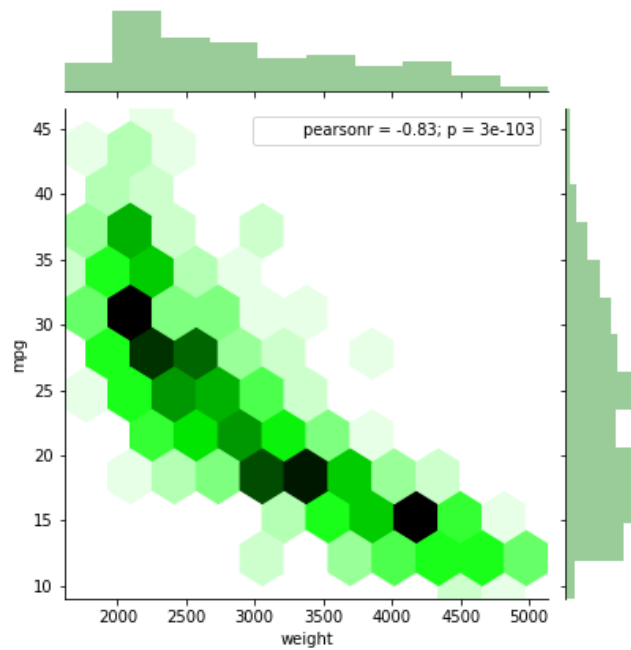
```
sns.jointplot(x='weight', y='mpg', data=dat, color='orange', kind='reg' )  
plt.show()
```



Seaborn 패키지 : Hex

Hex:

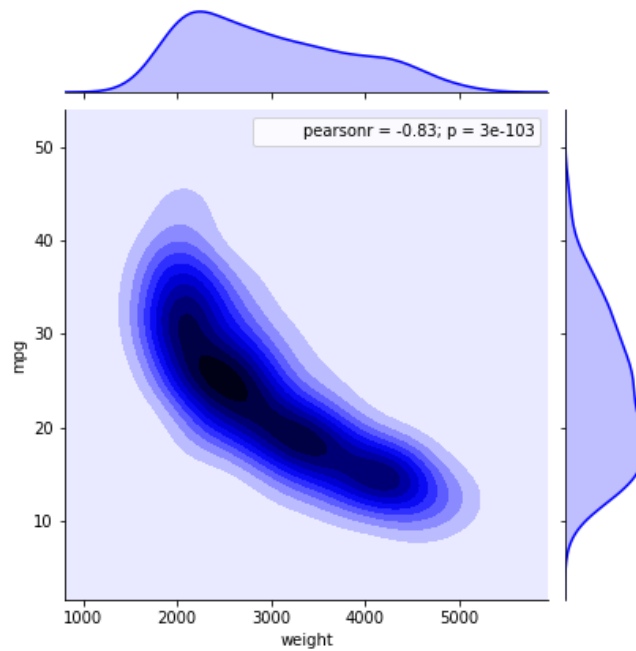
```
sns.jointplot(x='weight', y='mpg', data=dat, color='green', kind='hex' )  
plt.show()
```



Seaborn 패키지 : KDE

KDE:

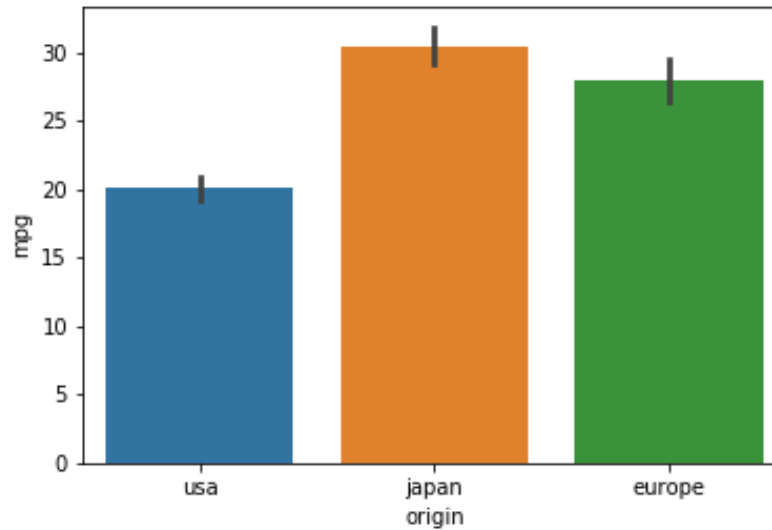
```
sns.jointplot(x='weight', y='mpg', data=dat, color='blue', kind='kde' )  
plt.show()
```



Seaborn 패키지 : 막대그림

막대그림:

```
sns.barplot(x='origin', y='mpg', data=dat)  
plt.show()
```



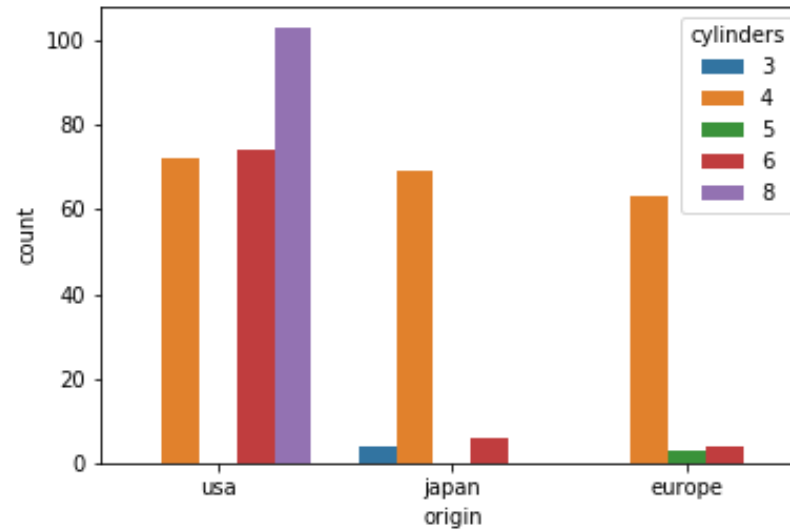
Seaborn 패키지 : 막대그림

막대그림:

또다른 변수를 인자 hue의 값으로 포함.

```
sns.countplot(x='origin', data=dat, hue='cylinders')
```

```
plt.show()
```

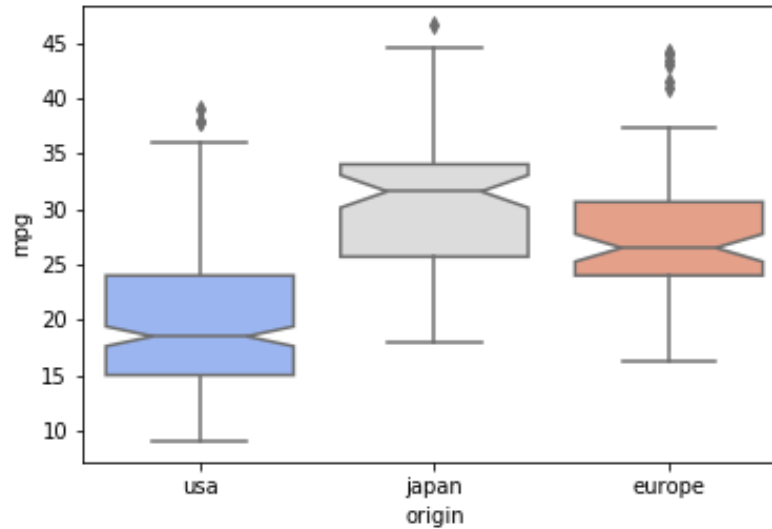


Seaborn 패키지 : 상자그림

상자그림:

다중 상자그림.

```
sns.boxplot(x='origin', y='mpg', data=dat, palette='coolwarm', notch=True)  
plt.show()
```

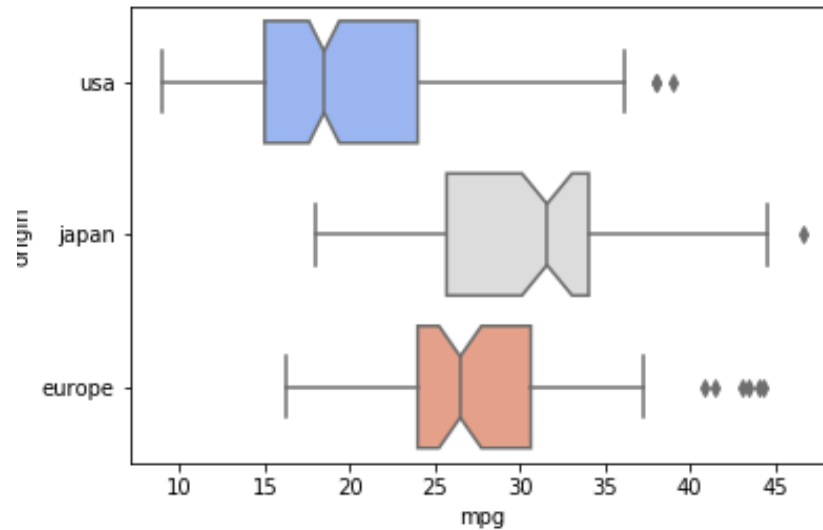


Seaborn 패키지 : 상자그림

상자그림:

다중 상자그림.

```
sns.boxplot(x='mpg', y='origin', data=dat, palette='coolwarm', notch=True)  
plt.show()
```

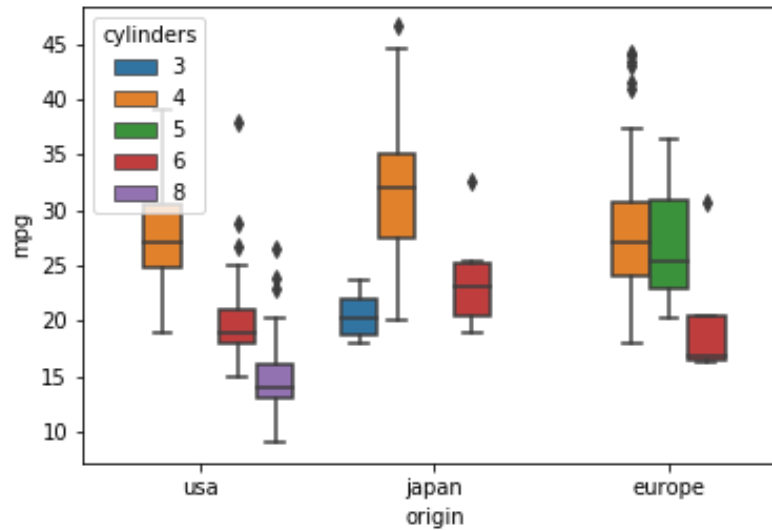


Seaborn 패키지 : 상자그림

상자그림:

인자 hue의 값으로 또 다른 변수 포함.

```
sns.boxplot(x='origin', y='mpg', data=dat, hue='cylinders' )  
plt.show()
```

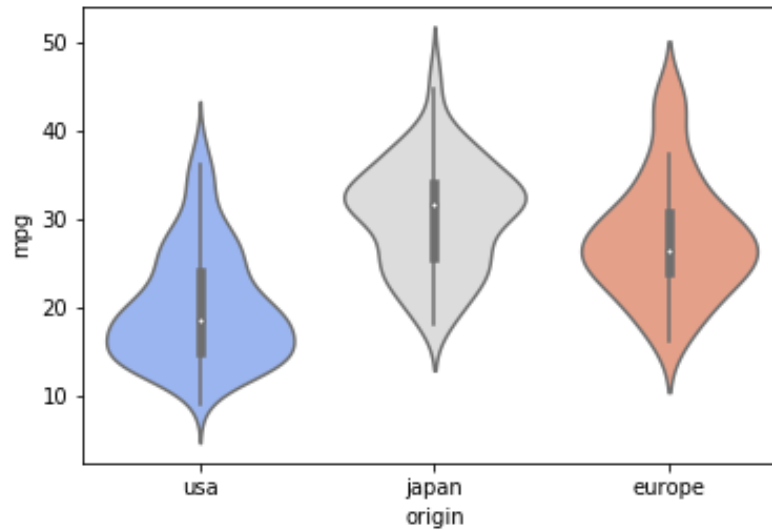


Seaborn 패키지 : 바이올린 플롯

바이올린 플롯:

boxplot과 인자가 같음.

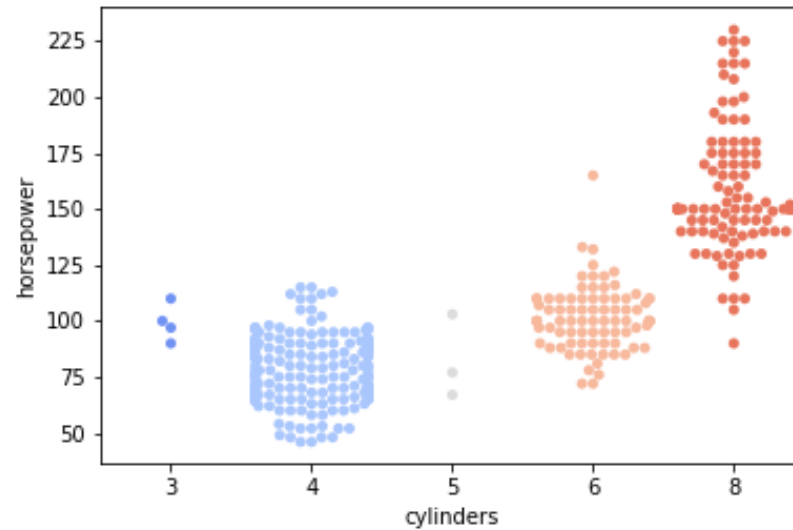
```
sns.violinplot(x='origin', y='mpg', data=dat, palette='coolwarm' )  
plt.show()
```



Seaborn 패키지 : Swarm plot

Swarm plot:

```
sns.swarmplot(x='cylinders', y='horsepower', data=dat, palette='coolwarm')  
plt.show()
```

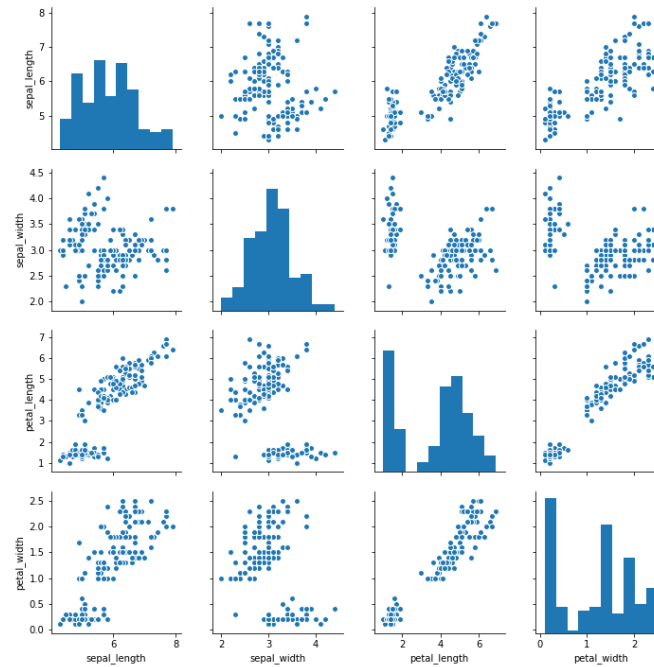


Seaborn 패키지 : 산점도 행렬

산점도 행렬:

```
sns.pairplot(dat)
```

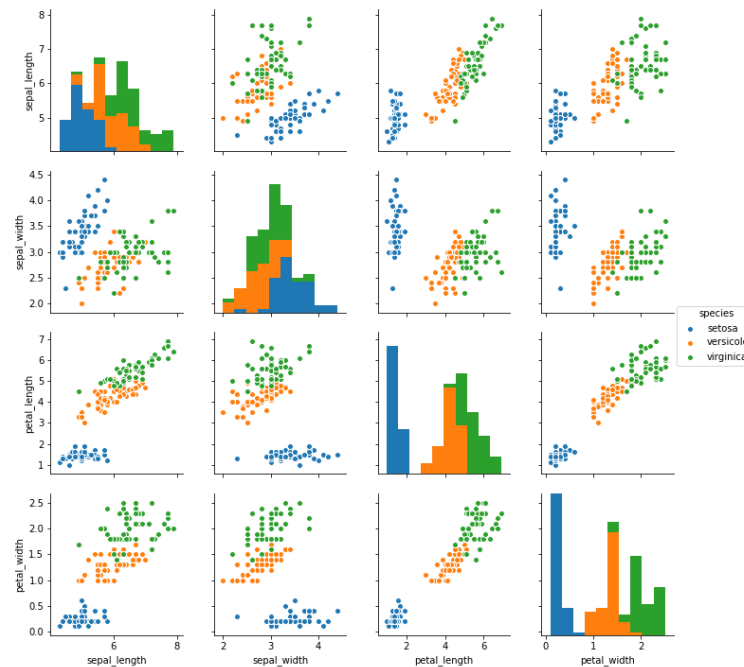
```
plt.show()
```



Seaborn 패키지 : 산점도 행렬

산점도 행렬:

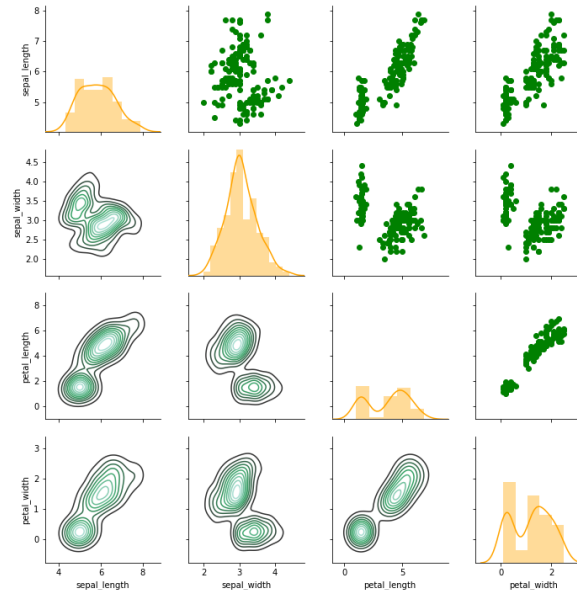
```
sns.pairplot(dat, hue='species' )  
plt.show()
```



Seaborn 패키지 : 혼합 유형 시각화 행렬

혼합 유형 시각화 행렬:

```
g=sns.PairGrid(dat)
g.map_diag(sns.distplot, color='orange')      # 대각선 = 히스토그램.
g.map_upper(plt.scatter, color='green')       # 위 삼각 = 산점도.
g.map_lower(sns.kdeplot, color='blue')        # 아래 삼각 = KDE.
plt.show()
```



Seaborn 패키지 : 다중 시각화

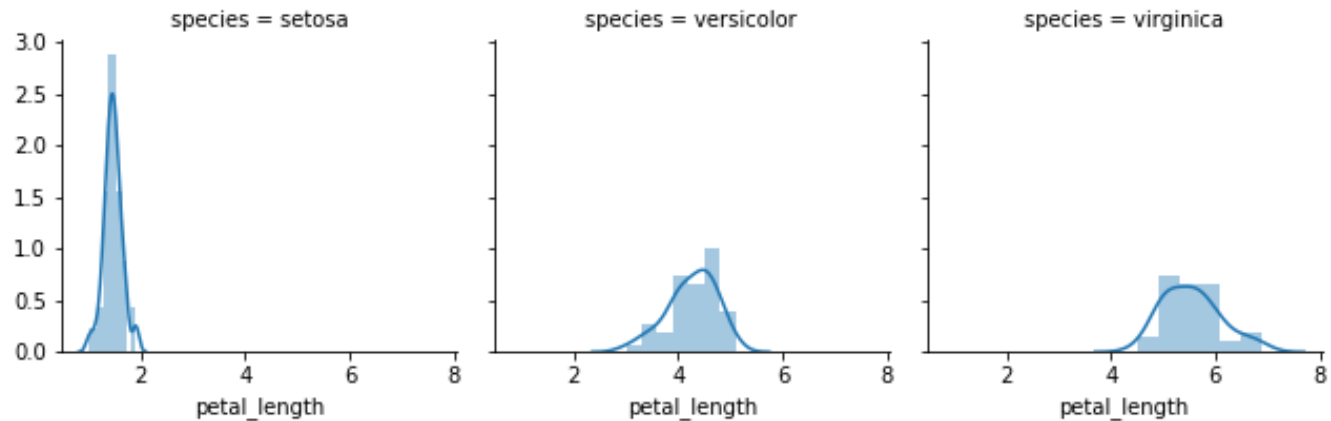
다중 시각화:

다중 히스토그램.

```
g=sns.FacetGrid(data=dat, col='species')
```

```
g.map(sns.distplot, 'petal_length')
```

```
plt.show()
```

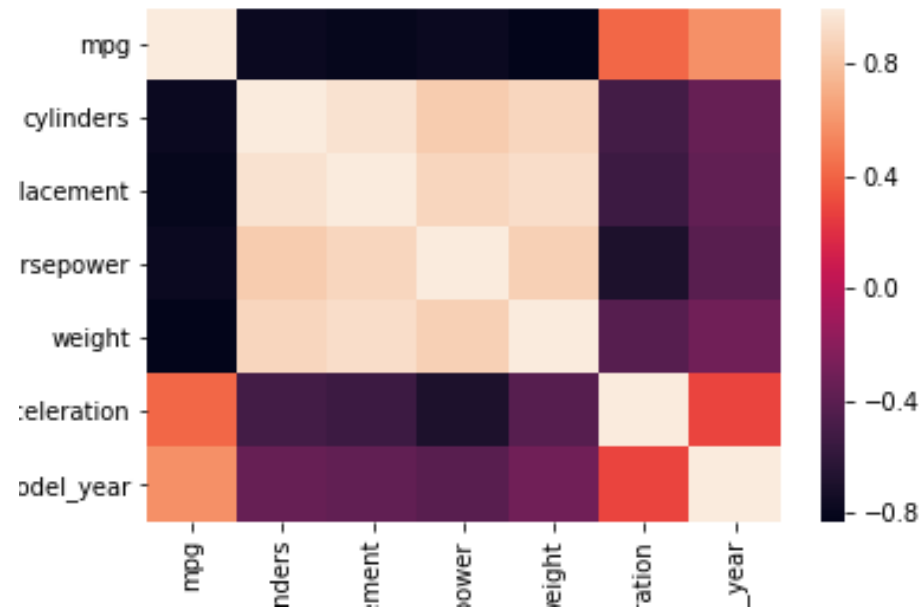


Seaborn 패키지 : Heatmap

Heatmap:

```
sns.heatmap(x)
```

```
plt.show()
```

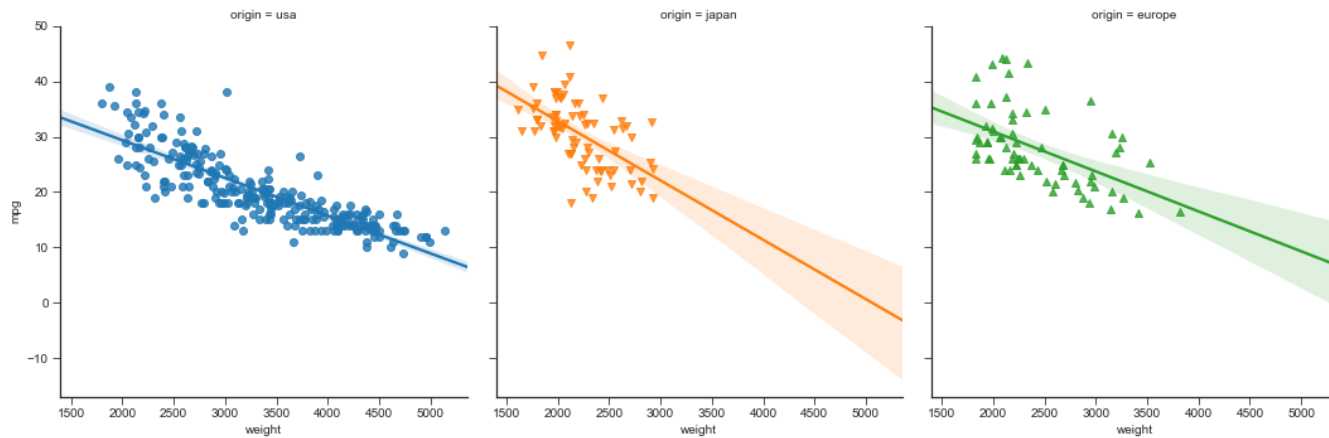


Seaborn 패키지 : 산점도 + 회귀선

산점도 + 회귀선:

다중 시각화.

```
sns.lmplot(data=dat, x='weight', y='mpg', col = 'origin', hue = 'origin')  
plt.show()
```



Seaborn 패키지 : 컬러 Palette

컬러 Palette:

```
sns.countplot(x='origin', data=dat, palette='cubehelix')  
# sns.countplot(x='origin', data=dat, palette='coolwarm')  
plt.show()
```

