Ahmed Al-dayel – some data visualizations using by python.

exploratory analyses:

descriptive enough for you to gain insights into the data.

explanatory analyses:

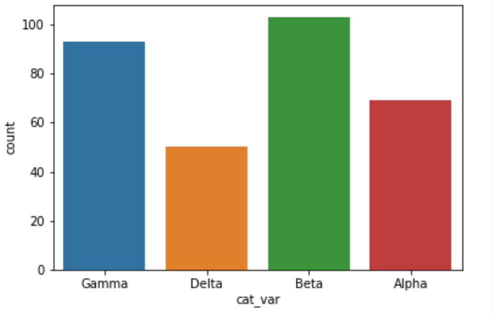
your goal will be to convey your findings to other people who don't have the level of hands-on experience with the data as you. Visualizations under this banner should be focused on telling a specific story that you want to convey to that particular audience.

univariate exploration

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

%matplotlib inline
```

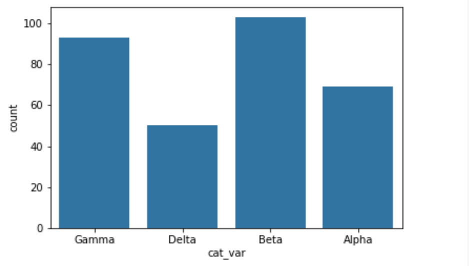
محور x

```
sb.countplot(data = df, x = 'cat_var')
```



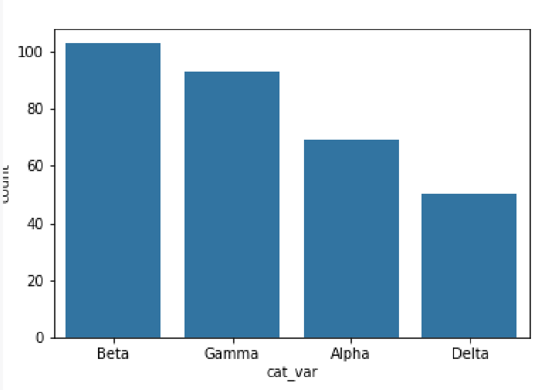
color

```
base_color = sb.color_palette()[0]
sb.countplot(data = df, x = 'cat_var', color = base_color)
```

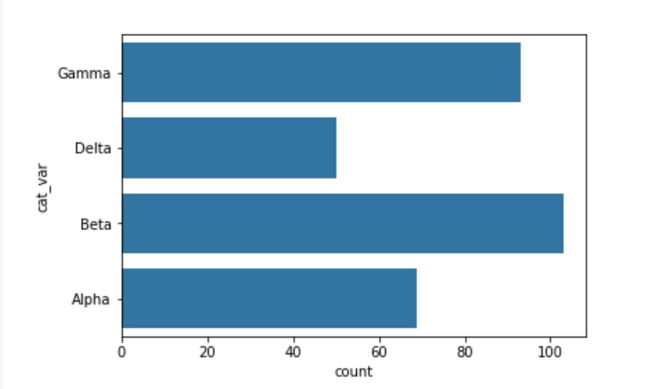


order

```
base_color = sb.color_palette()[0]
cat_order = df['cat_var'].value_counts().index
sb.countplot(data = df, x = 'cat_var', color =
base_color, order = cat_order)
```



```
 محـور y
lpase_color = sb.color_palette()[0]
sb.countplot(data = df, y = 'cat_var', color =
base_color)
```

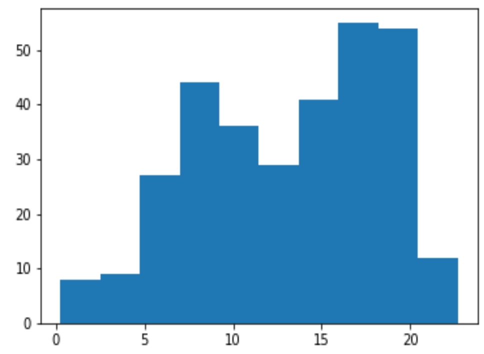


Counting Missing Data

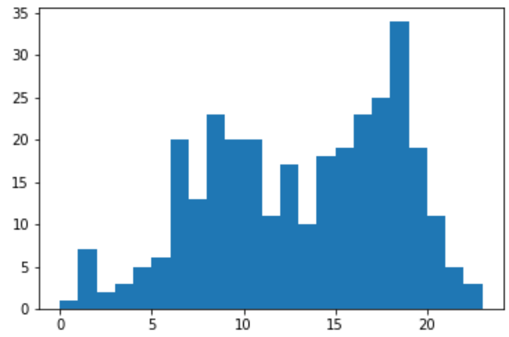
```
df.isna().sum()
```

Histograms

```
plt.hist(data = df, x = 'num_var')
```



```
bin_edges = np.arange(0, df['num_var'].max()+1, 1)
plt.hist(data = df, x = 'num_var', bins = bin_edges)
```



يحط لك خط

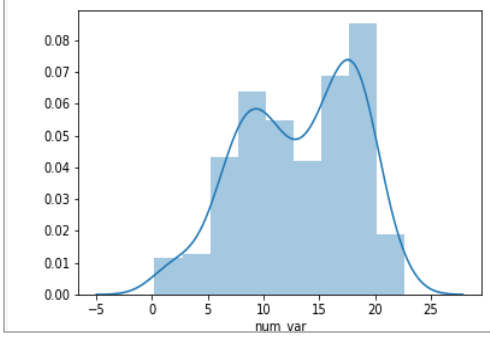
```
sb.distplot(df['num_var'])
```

تحط لك لميت

```
plt.xlim(0, 6)
```

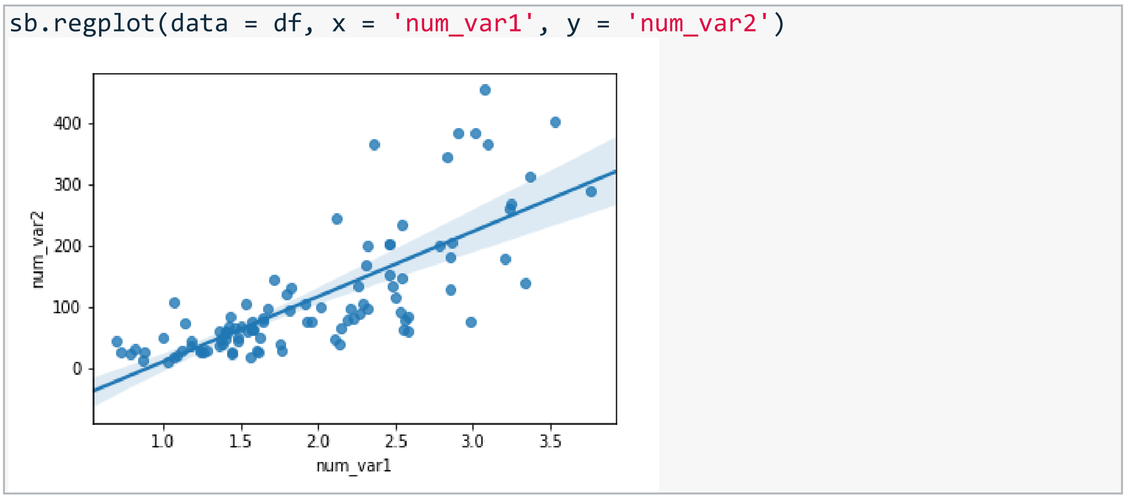
يحط لك نقاط

```
plt.scatter(data = df, x = 'num_var1', y = 'num_var2')
```



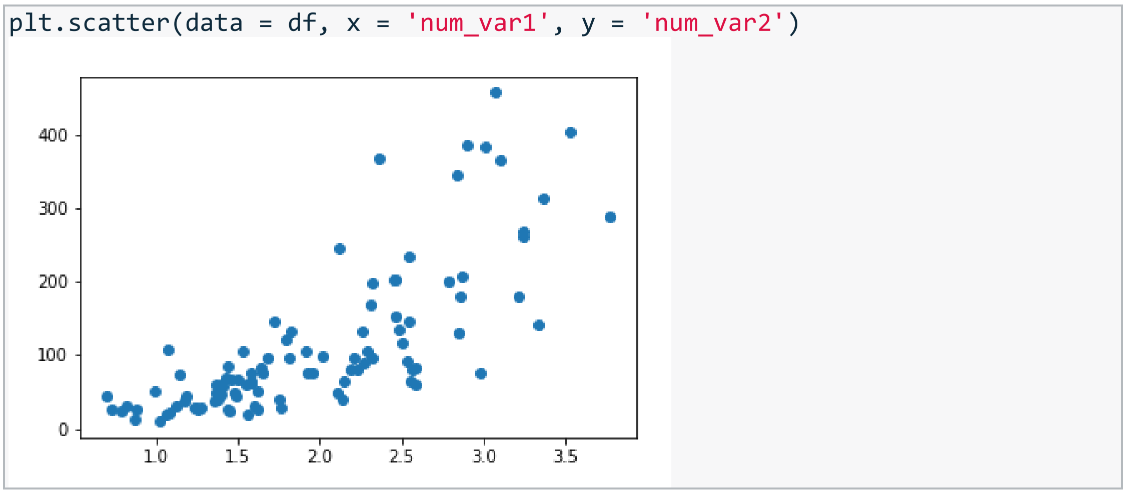
يحط لك خط على النقاط :

Bivariate exploration:



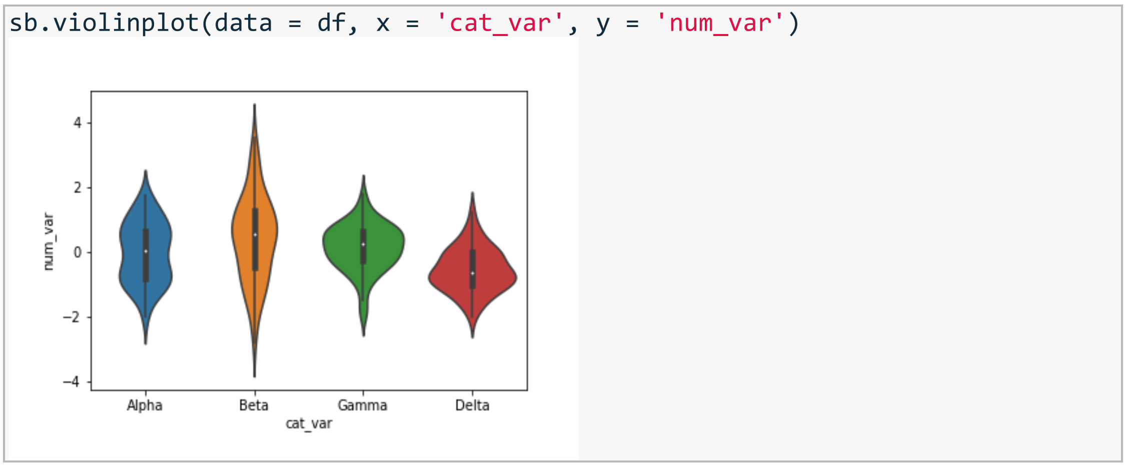
```
sb.regplot(data = df, x = 'num_var1', y = 'num_var2')
```

Scatterplots



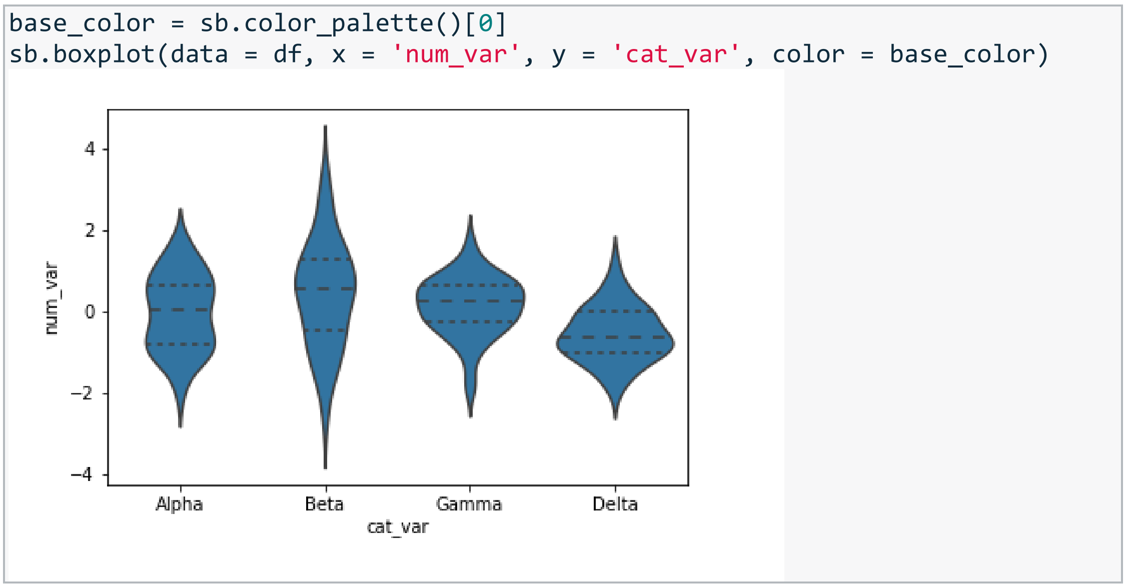
```
plt.scatter(data = df, x = 'num_var1', y = 'num_var2')
```

Violin Plots



```
sb.violinplot(data = df, x = 'cat_var', y = 'num_var')
```

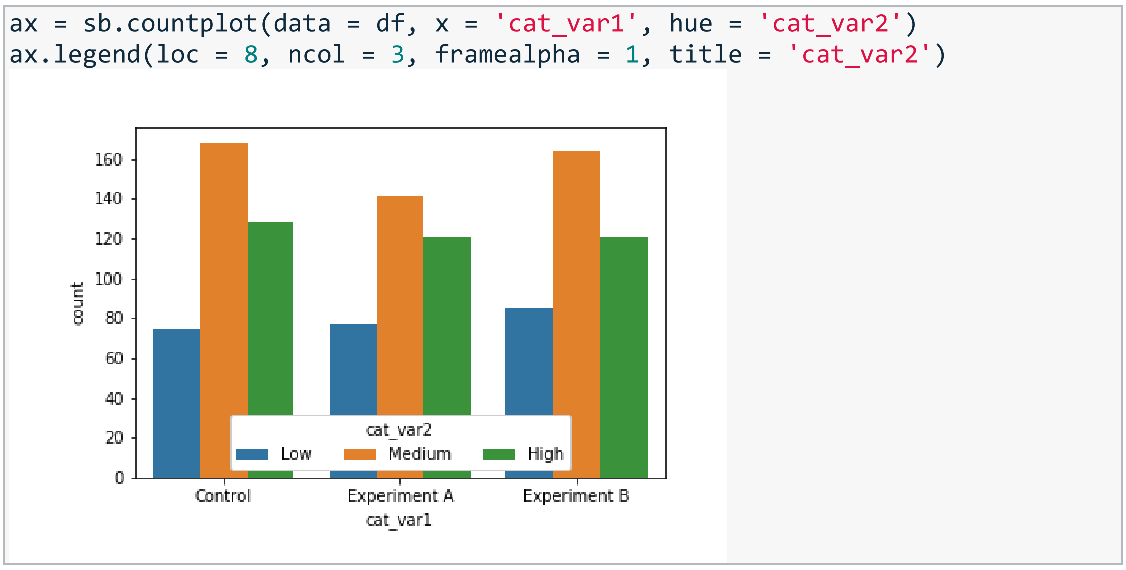
box plot



```
base_color = sb.color_palette()[0]
sb.boxplot(data = df, x = 'num_var', y = 'cat_var', color = base_color)
```

countplot

رهيبة لازم تستخدمها في المشروع

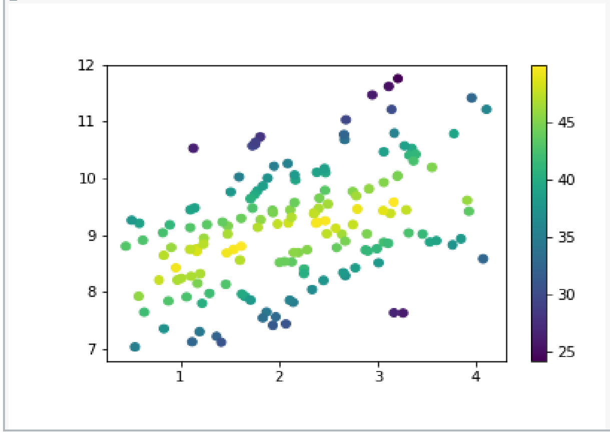


```
ax = sb.countplot(data = df, x = 'cat_var1', hue = 'cat_var2')
ax.legend(loc = 8, ncol = 3, framealpha = 1, title = 'cat_var2')
```

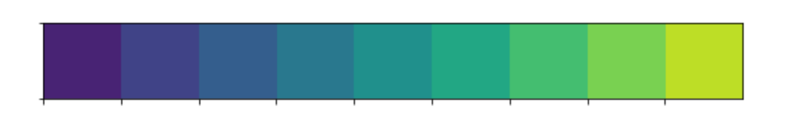
Multivariate exploration:

colorbar

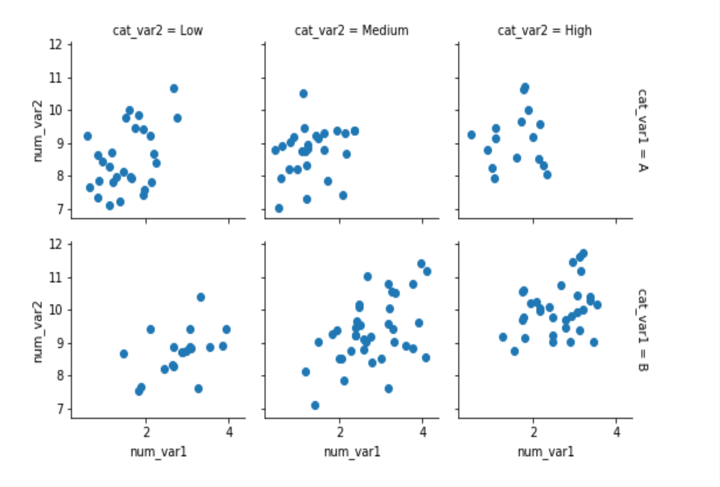
```
plt.scatter(data = df, x = 'num_var1', y = 'num_var2', c
= 'num_var3',
                  cmap = 'mako_r')
plt.colorbar()
```

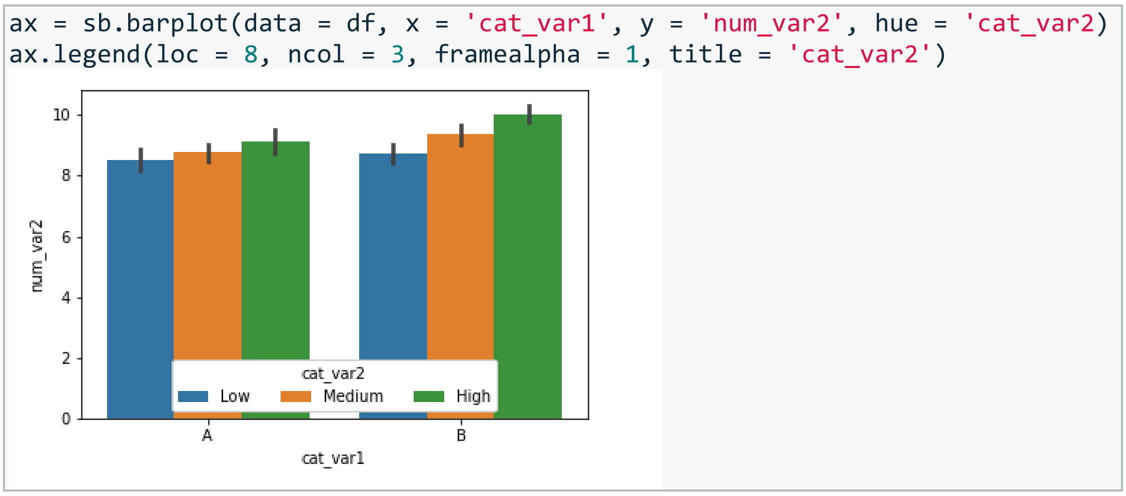


```
sb.palplot(sb.color_palette('viridis', 9))
```

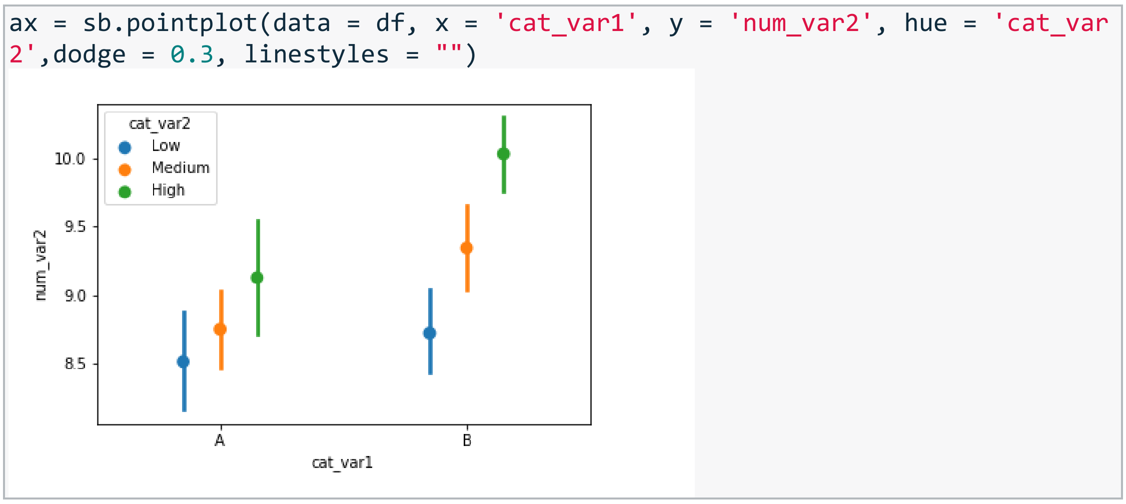


```
g = sb.FacetGrid(data = df, col = 'cat_var2', row = 'cat_var1', size = 2.5,
                 margin_titles = True)
g.map(plt.scatter, 'num_var1', 'num_var2')
```

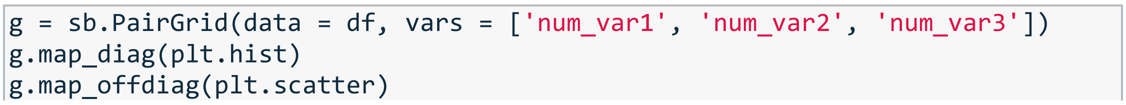




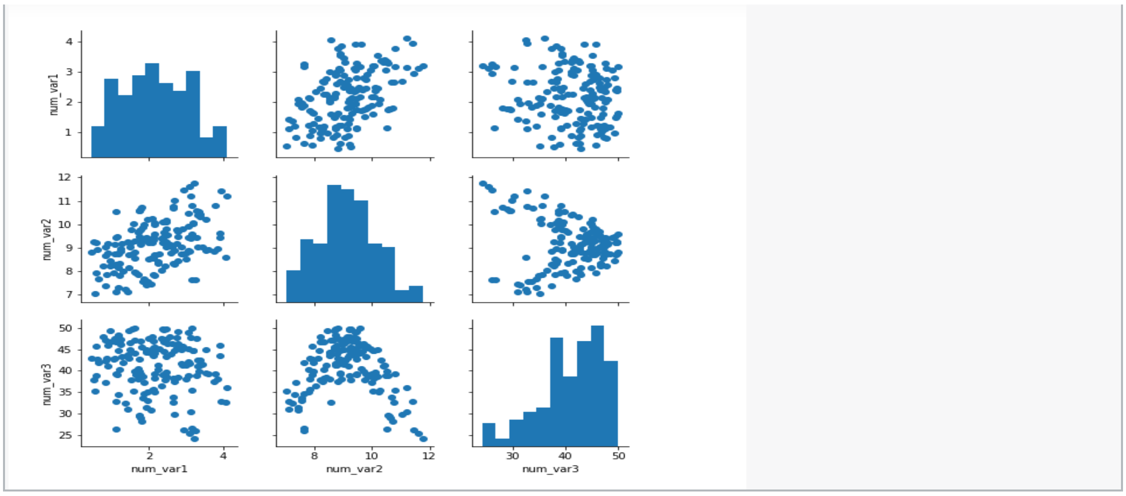
```
ax = sb.barplot(data = df, x = 'cat_var1', y = 'num_var2', hue = 'cat_var2)
ax.legend(loc = 8, ncol = 3, framealpha = 1, title = 'cat_var2')
```

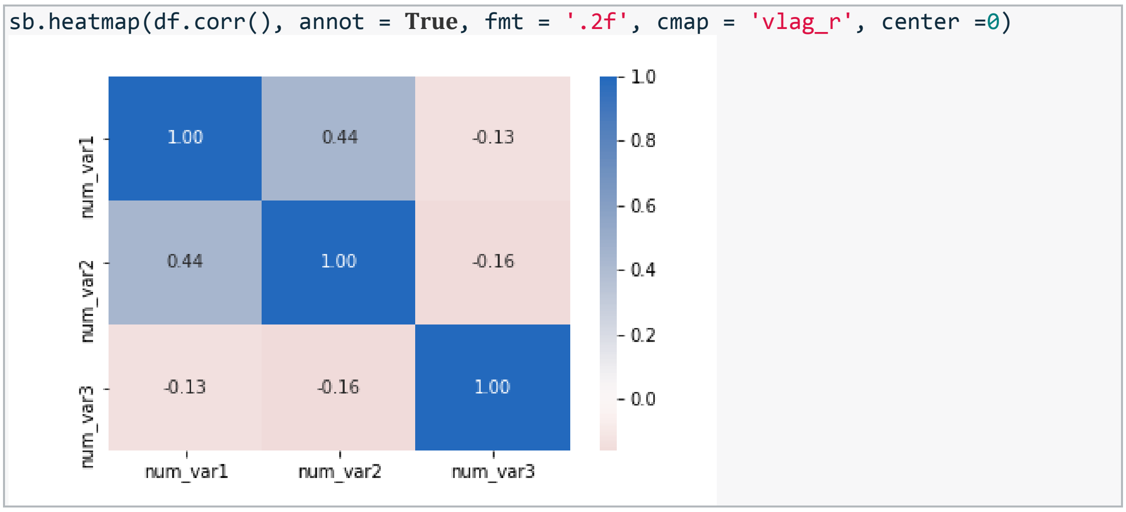


```
ax = sb.pointplot(data = df, x = 'cat_var1', y = 'num_var2', hue = 'cat_var
2',dodge = 0.3, linestyles = "")
```



```
g = sb.PairGrid(data = df, vars = ['num_var1', 'num_var2', 'num_var3'])
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter)
```





```
sb.heatmap(df.corr(), annot = True, fmt = '.2f', cmap = 'vlag_r', center =0)
```