**DATA VISUALIZATION** with *Seaborn* & *Matplotlib*:

**Objective:** To showcase how data can be visualized in various ways, using *Python*.

```
# Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and inter
```

Importing Libraries:

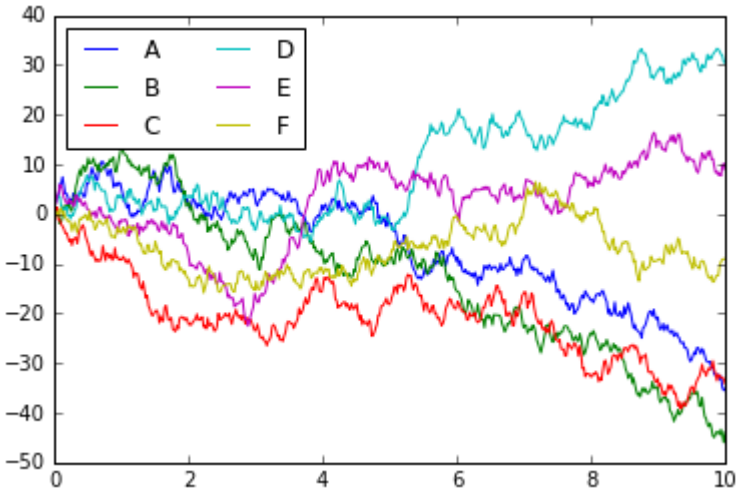
```
import matplotlib.pyplot as plt
plt.style.use('classic')
%matplotlib inline
import numpy as np
import pandas as pd
```

```
#Now we create some random walk data:
```

```
# Create some data
rng = np.random.RandomState(0)
x = np.linspace(0, 10, 500)
y = np.cumsum(rng.randn(500, 6), 0)
```

```
#And do a simple plot:
```

```
# Plot the data with Matplotlib defaults
plt.plot(x, y)
plt.legend('ABCDEF', ncol=2, loc='upper left');
```




```
# Seaborn:  -  Seaborn is an open-source Python library built on top of matplotlib. It is use
#           -  Seaborn works easily with dataframes and the Pandas library. The graphs created
```

## Importing Seaborn:

```
import seaborn as sns
sns.set()
```

```
# same plotting code as above!
plt.plot(x, y)
plt.legend('ABCDEF', ncol=2, loc='upper left');
```

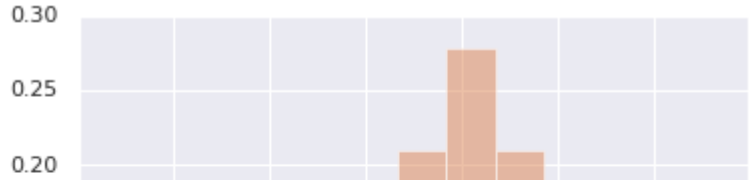

```
#Exploring Seaborn Plots:
```

```
#Histograms, KDE, and densities
```

## Plotting univariate or bivariate histogram to show distributions of datasets:

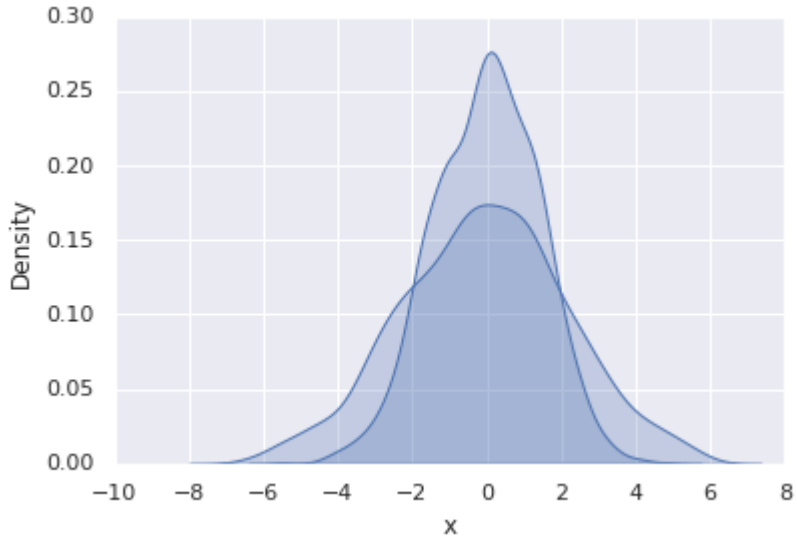
```
data = np.random.multivariate_normal([0, 0], [[5, 2], [2, 2]], size=2000)
data = pd.DataFrame(data, columns=['x', 'y'])

for col in 'xy':
    plt.hist(data[col], density=True, alpha=0.5)
```




Rather than a histogram, we can get a smooth estimate of the distribution using a kernel density estimation

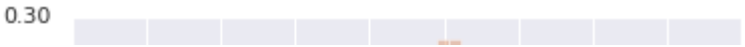
```
for col in 'xy':
    sns.kdeplot(data[col], shade=True)
```




Histograms and KDE can be combined using distplot:

```
sns.distplot(data['x'])
sns.distplot(data['y']);
```

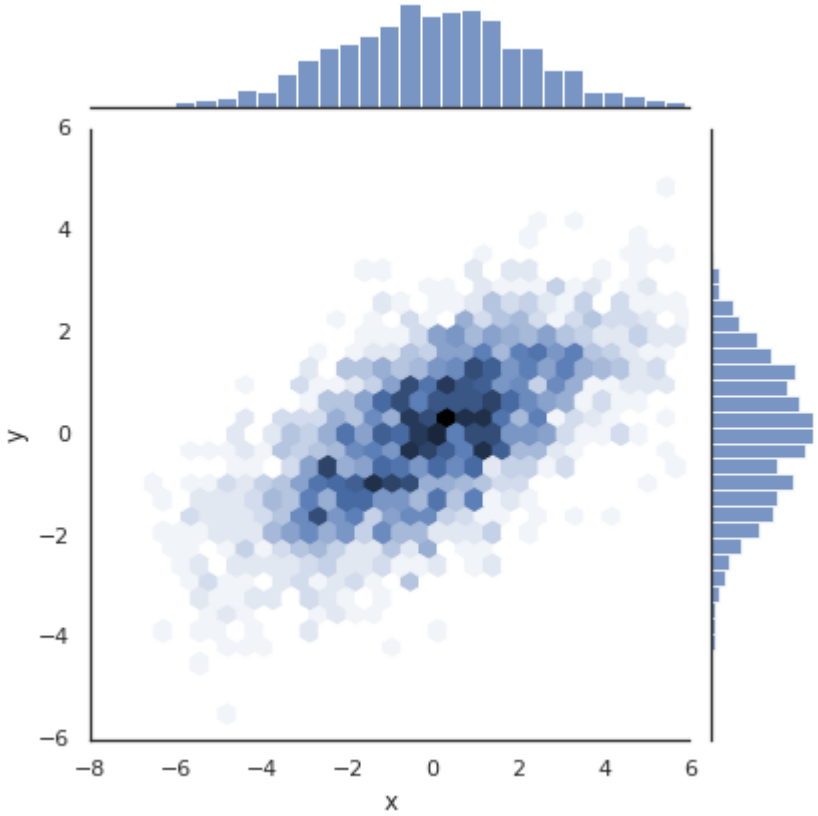
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
```




Jointplot() in Seaborn library creates a scatter plot with two histograms at the top and right margins of the graph by default.

```
with sns.axes_style('white'):
    sns.jointplot("x", "y", data, kind='hex')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: P
  FutureWarning
```




Working with Wine quality Dataset:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
%matplotlib inline
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files winequalityN.csv

- **winequalityN.csv**(application/vnd.ms-excel) - 390376 bytes, last modified: 8/27/2021 - 100% done

```
Saving winequalityN.csv to winequalityN.csv
```

```
import io
df = pd.read_csv(io.BytesIO(uploaded['winequalityN.csv']))


# Dataset is now stored in a Pandas Dataframe
```

Viewing the Data:

```
df.head(20)
```

| | type | fixed | volatile | citric | residual | chlorides | free sulfur | total sulfur | densit |
|---|---|---|---|---|---|---|---|---|---|
| **0** | white | 7.0 | 0.27 | 0.36 | 20.70 | 0.045 | 45.0 | 170.0 | 1.001 |

Cleaning Null values:

```
df=df.fillna(df.mean())
```

Statistical Description of the dataset:

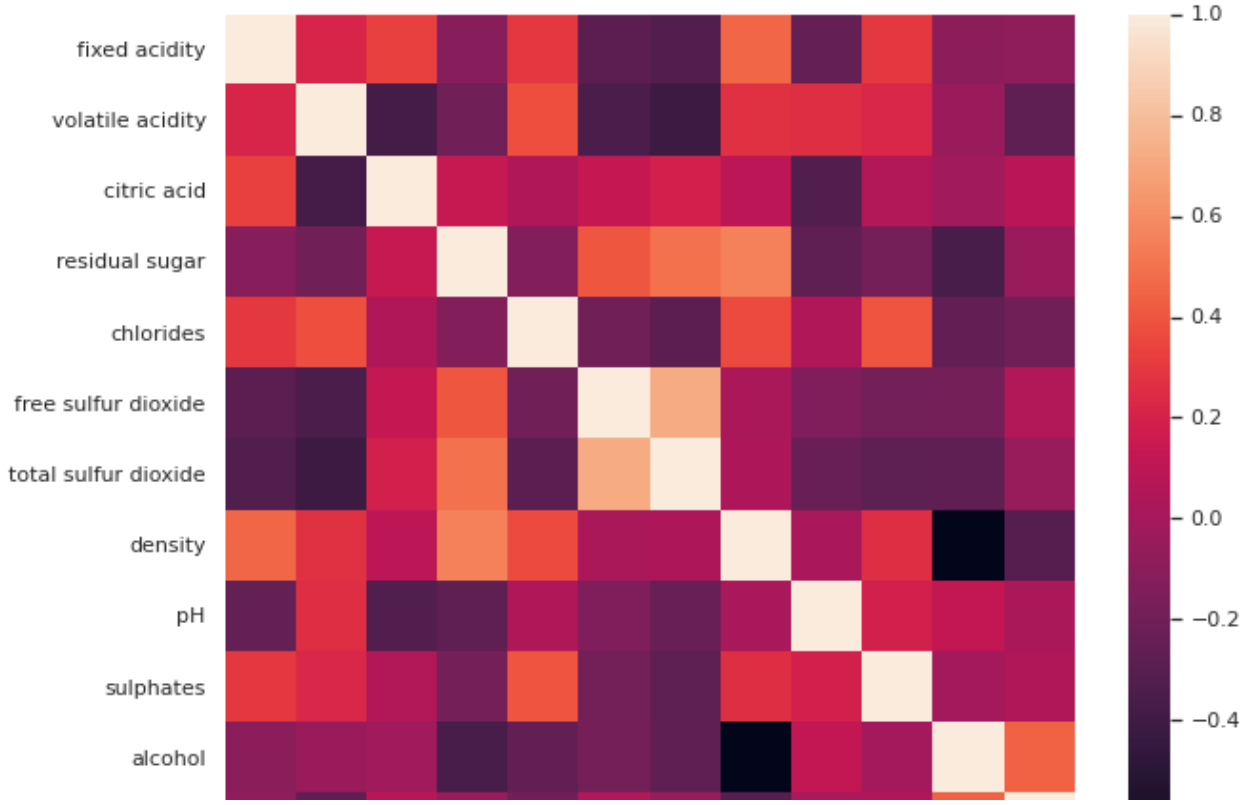
```
df.describe()
```

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide |
|---|---|---|---|---|---|---|
| **count** | 6497.000000 | 6497.000000 | 6497.000000 | 6497.000000 | 6497.000000 | 6497.000000 |
| **mean** | 7.216579 | 0.339691 | 0.318722 | 5.444326 | 0.056042 | 30.525319 |
| **std** | 1.295751 | 0.164548 | 0.145231 | 4.757392 | 0.035031 | 17.749400 |
| **min** | 3.800000 | 0.080000 | 0.000000 | 0.600000 | 0.009000 | 1.000000 |
| **25%** | 6.400000 | 0.230000 | 0.250000 | 1.800000 | 0.038000 | 17.000000 |
| **50%** | 7.000000 | 0.290000 | 0.310000 | 3.000000 | 0.047000 | 29.000000 |
| **75%** | 7.700000 | 0.400000 | 0.390000 | 8.100000 | 0.065000 | 41.000000 |
| **max** | 15.900000 | 1.580000 | 1.660000 | 65.800000 | 0.611000 | 289.000000 |

**Heatmap** is defined as a graphical representation of data using colors to visualize the values.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **17** | white | NaN | 0.66 | 0.48 | 1.20 | 0.029 | 29.0 | 75.0 | 0.98 |

```
import seaborn as sns
sns.set(rc={'figure.figsize':(10,8)})
corr = df.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
plt.show()
```



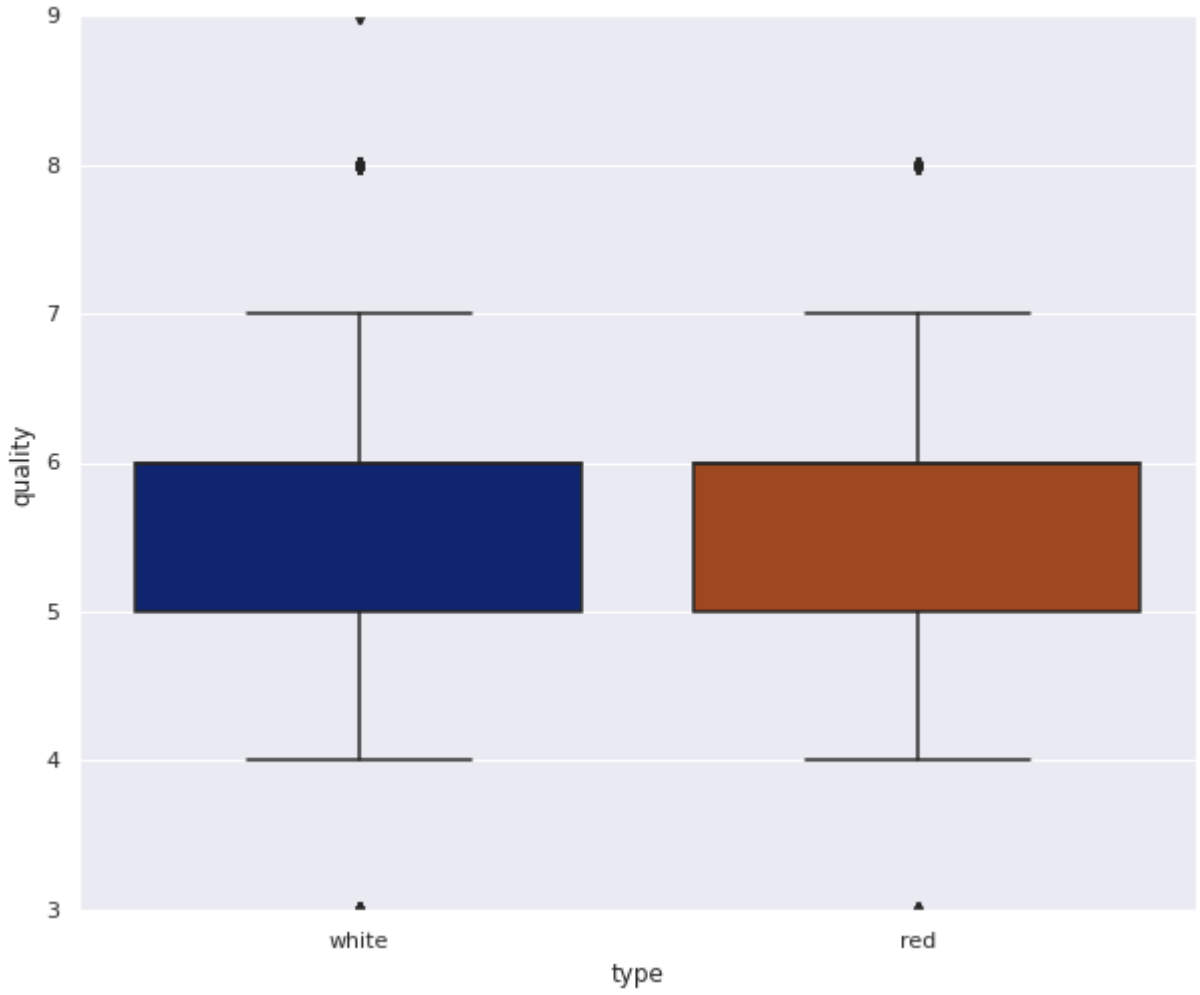

The **scatter plot** is a mainstay of statistical visualization.

```
plt.scatter("alcohol","quality",data=df)
plt.show()
```


**Box Plot** is the visual representation of the depicting groups of numerical data through their quartiles.



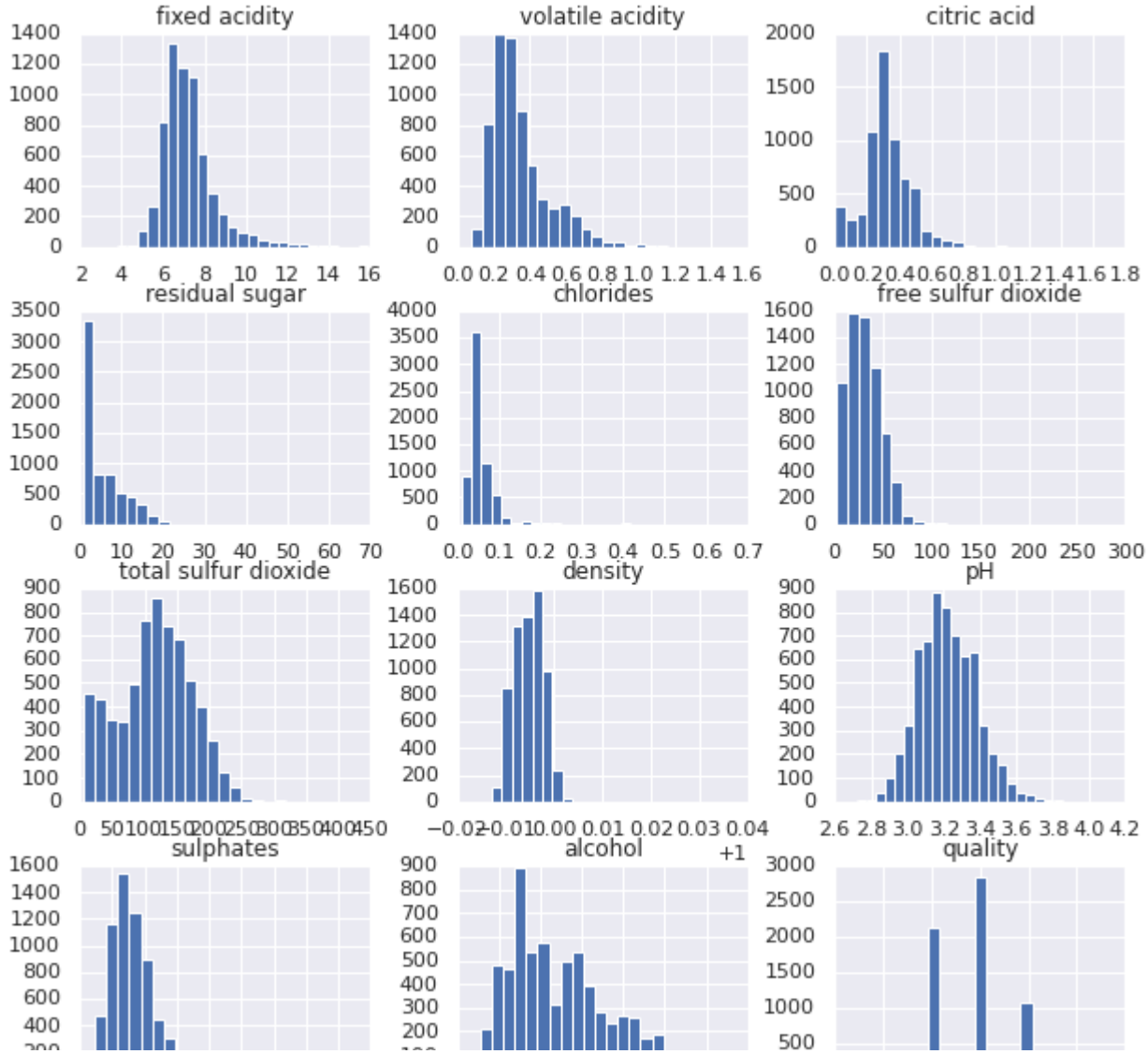

```
sns.boxplot(x="type",y="quality",data=df, palette="dark")
plt.show()
```




```
df=df[df.columns.drop('type')]
```

**Histograms** are used to display the distribution of one or several numerical variables.

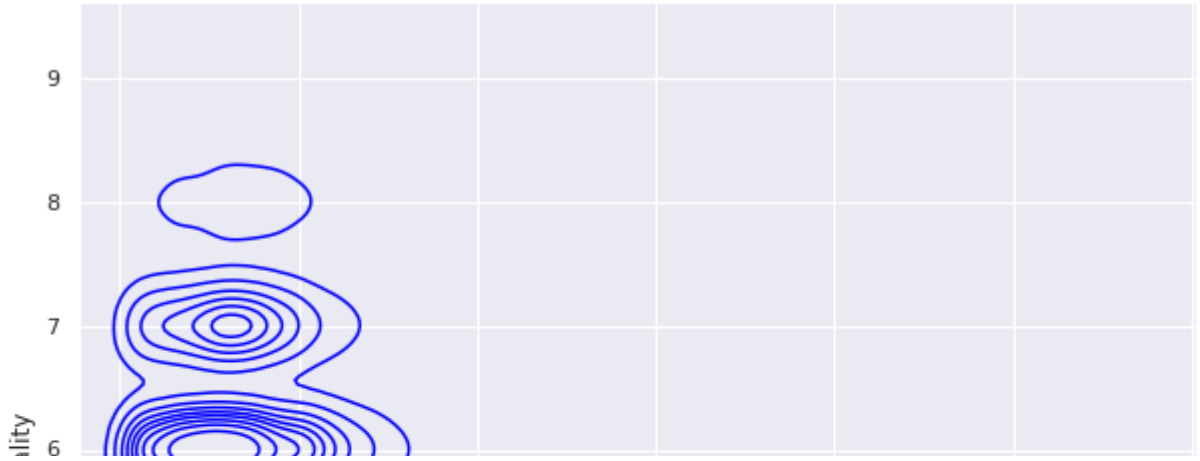
```
df.hist(bins=25,figsize=(10,10))
# display histogram
plt.show()
```




**Kdeplot** is a *Kernel Distribution Estimation Plot* which depicts the probability density function of the data variables

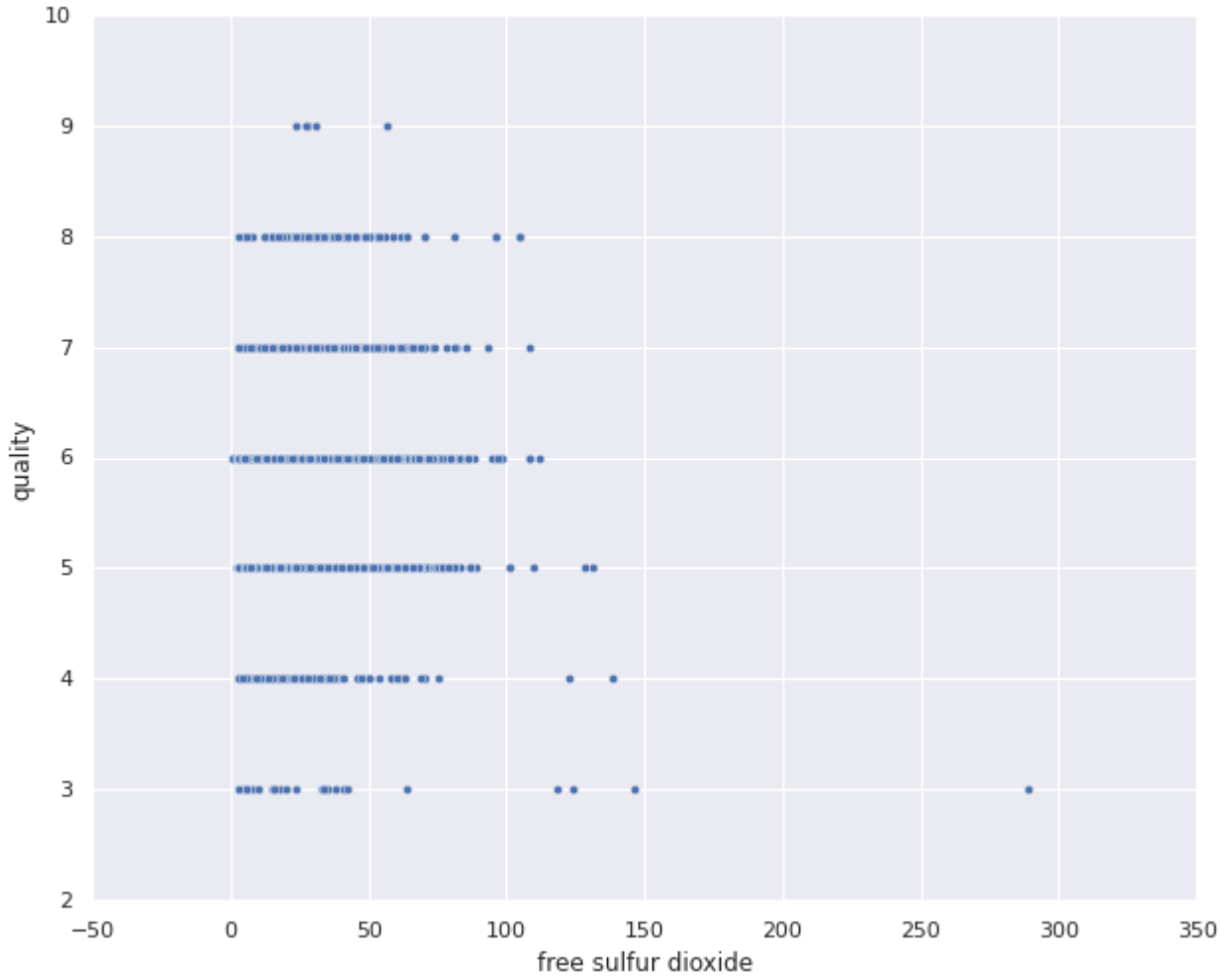
```
sns.kdeplot(x = 'free sulfur dioxide', y='quality' , data = df , color = 'blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7f8ec5fb90>
```




```
sns.scatterplot(x='free sulfur dioxide', y ='quality' ,
data = df )
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7f8e9c75d0>
```




Interactive 3D plot to visualize 3 data dimensions :

```
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
```

```
sns.set(style = "darkgrid")

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')

x = df['fixed acidity']
y = df['total sulfur dioxide']
z = df['quality']

ax.set_xlabel("Fixed Acidity")
ax.set_ylabel("Total SO2")
ax.set_zlabel("quality")

ax.scatter(x, y, z)

plt.show()
```