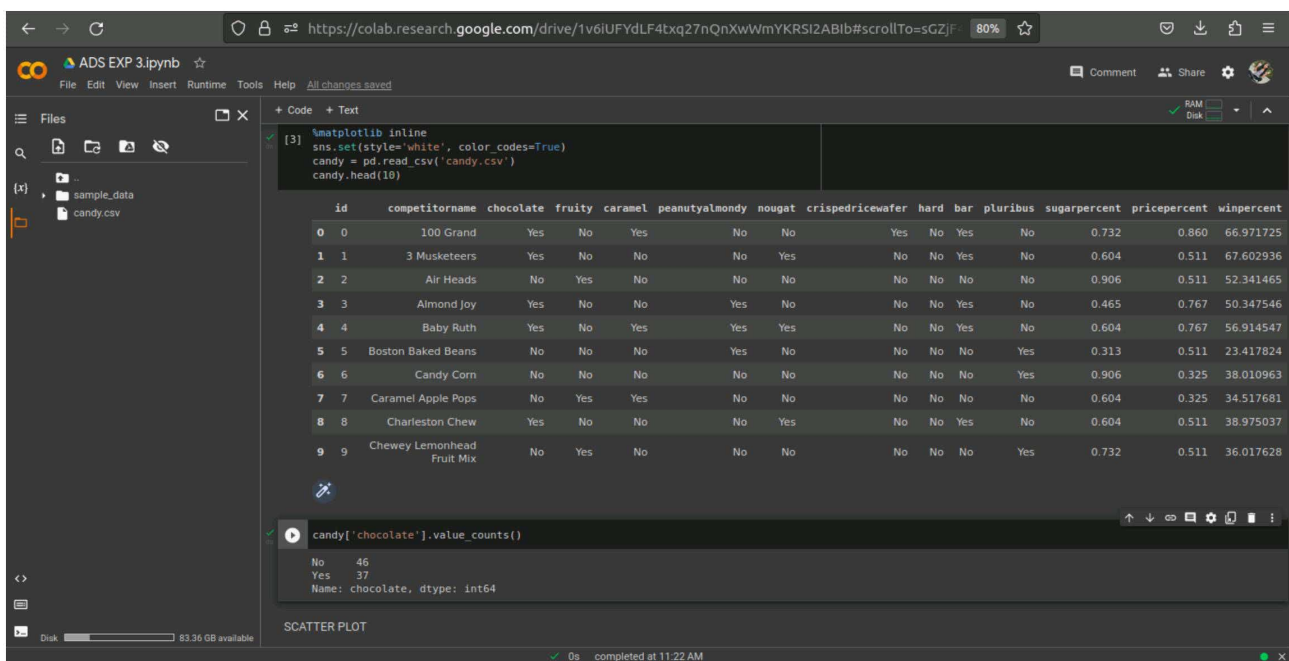


Experiment no: 3

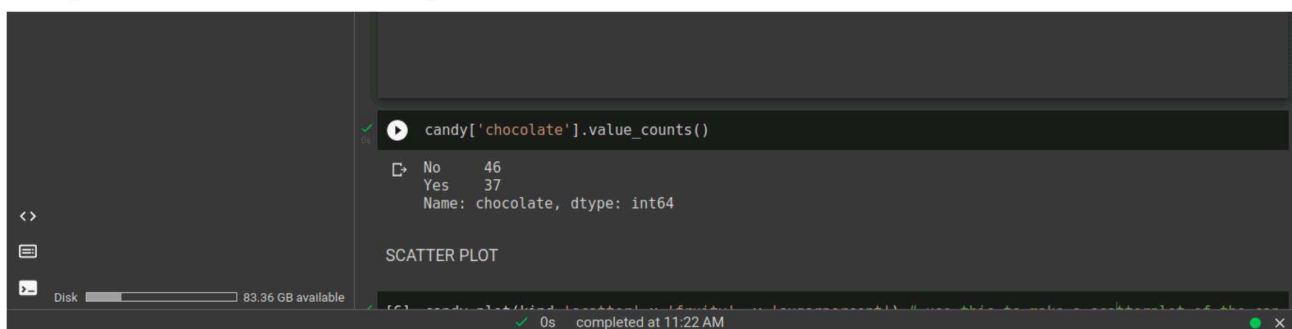
Aim: Explore data visualization techniques.

Program with outputs:

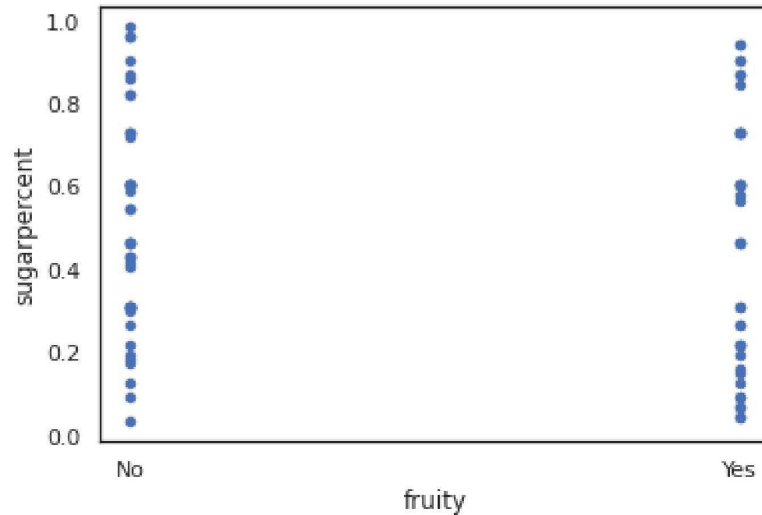
```
import numpy as np #linear algebra
import pandas as pd # a data processing and CSV I/O library
import warnings # current version of seaborn generates a bunch of warnings that will be ignore
warnings.filterwarnings('ignore')
# Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style='white', color_codes=True)
candy = pd.read_csv('candy.csv')
candy.head(10)
```



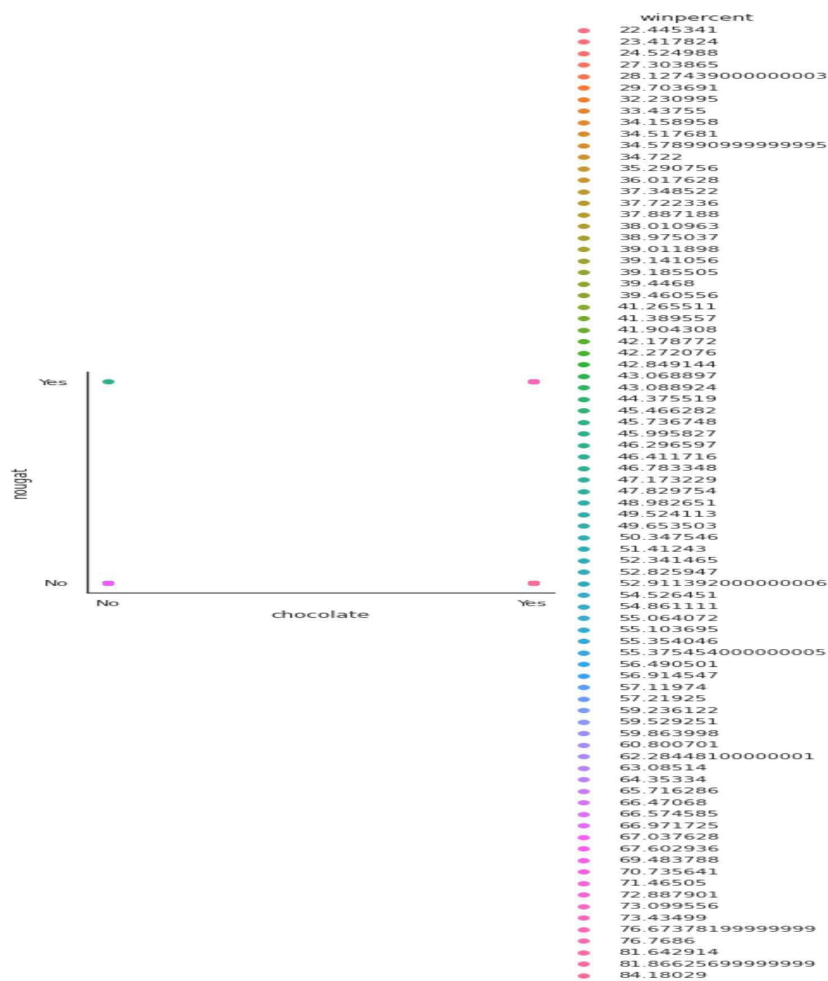
`candy['chocolate'].value_counts()`



```
candy.plot(kind='scatter',x='fruity', y='sugarpercent') # use this to make a scatterplot of the candy features.
```

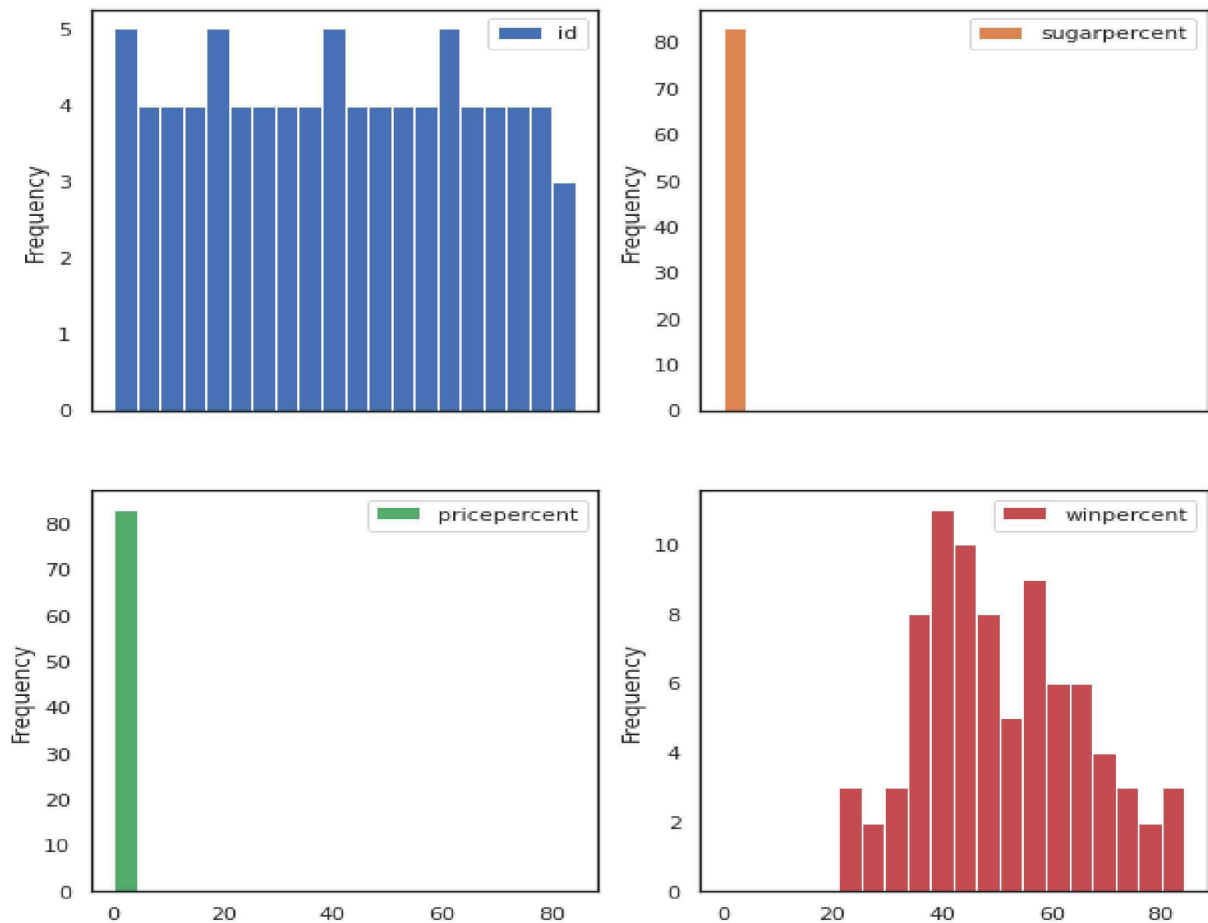


```
sns.FacetGrid(candy, hue = 'winpercent', size=5) \
.map(plt.scatter, 'chocolate','nougat') \
.add_legend()
```

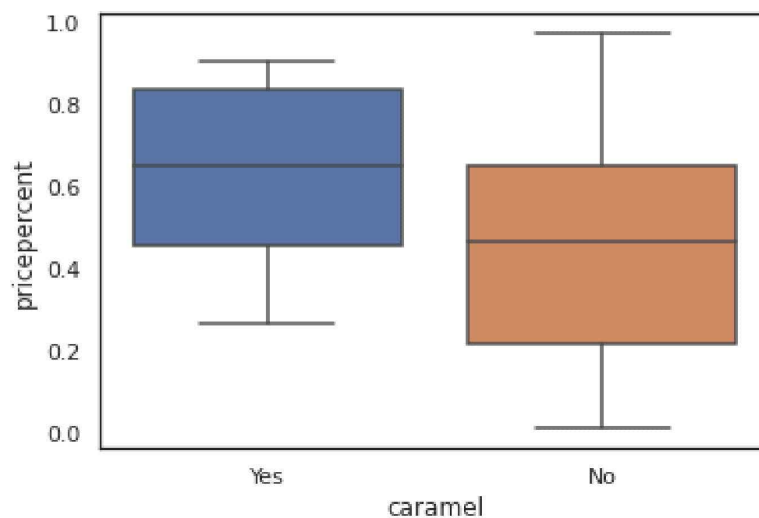


```
candy.plot.hist(subplots=True, layout=(2,2), figsize=(10, 10), bins=20)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f45022b46d0>,  
      <matplotlib.axes._subplots.AxesSubplot object at 0x7f45022bc670>],  
      [<matplotlib.axes._subplots.AxesSubplot object at 0x7f45007e1880>,  
      <matplotlib.axes._subplots.AxesSubplot object at 0x7f450078f9a0>]],  
      dtype=object)
```



```
# We can look at an individual feature in Seaborn through a boxplot  
sns.boxplot(x='caramel', y='pricepercent', data=candy)  
<matplotlib.axes._subplots.AxesSubplot at 0x7f450042d460>
```



```

# get correlation matrix
corr = candy.corr()
fig, ax = plt.subplots()
# create heatmap
im = ax.imshow(corr.values)

# set labels
ax.set_xticks(np.arange(len(corr.columns)))
ax.set_yticks(np.arange(len(corr.columns)))
ax.set_xticklabels(corr.columns)
ax.set_yticklabels(corr.columns)

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(len(corr.columns)):
    for j in range(len(corr.columns)):
        text = ax.text(j, i, np.around(corr.iloc[i, j], decimals=2),
            ha="center", va="center", color="black")

```

