Exploratory data analysis

Autumn 2020

```
################################################
##                                            ##
##  Machine Learning Fundamentals with Python  ##
##                                            ##
################################################

§1 Supervised Learning with scikit-learn

§1.1 Classification

§1.1.2 Exploratory data analysis
```

1. What are the features and the target variable of the Iris dataset?

- Features: petal length, petal width, sepal length, sepal width
- Target variable: species (*versicolor*, *virginica*, *setosa*)

2. Code of the Iris dataset in scikit-learn:

[1]:
```
from sklearn import datasets

iris = datasets.load_iris()
type(iris)
```

[1]:
```
sklearn.utils.Bunch
```

[2]:
```
print(iris.keys())
```

```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names',
'filename'])
```

[3]:
```
type(iris.data), type(iris.target)
```

[3]:
```
(numpy.ndarray, numpy.ndarray)
```

[4]:
```
iris.data.shape
```

[4]:
```
(150, 4)
```

[5]:
```
iris.target_names
```

[5]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

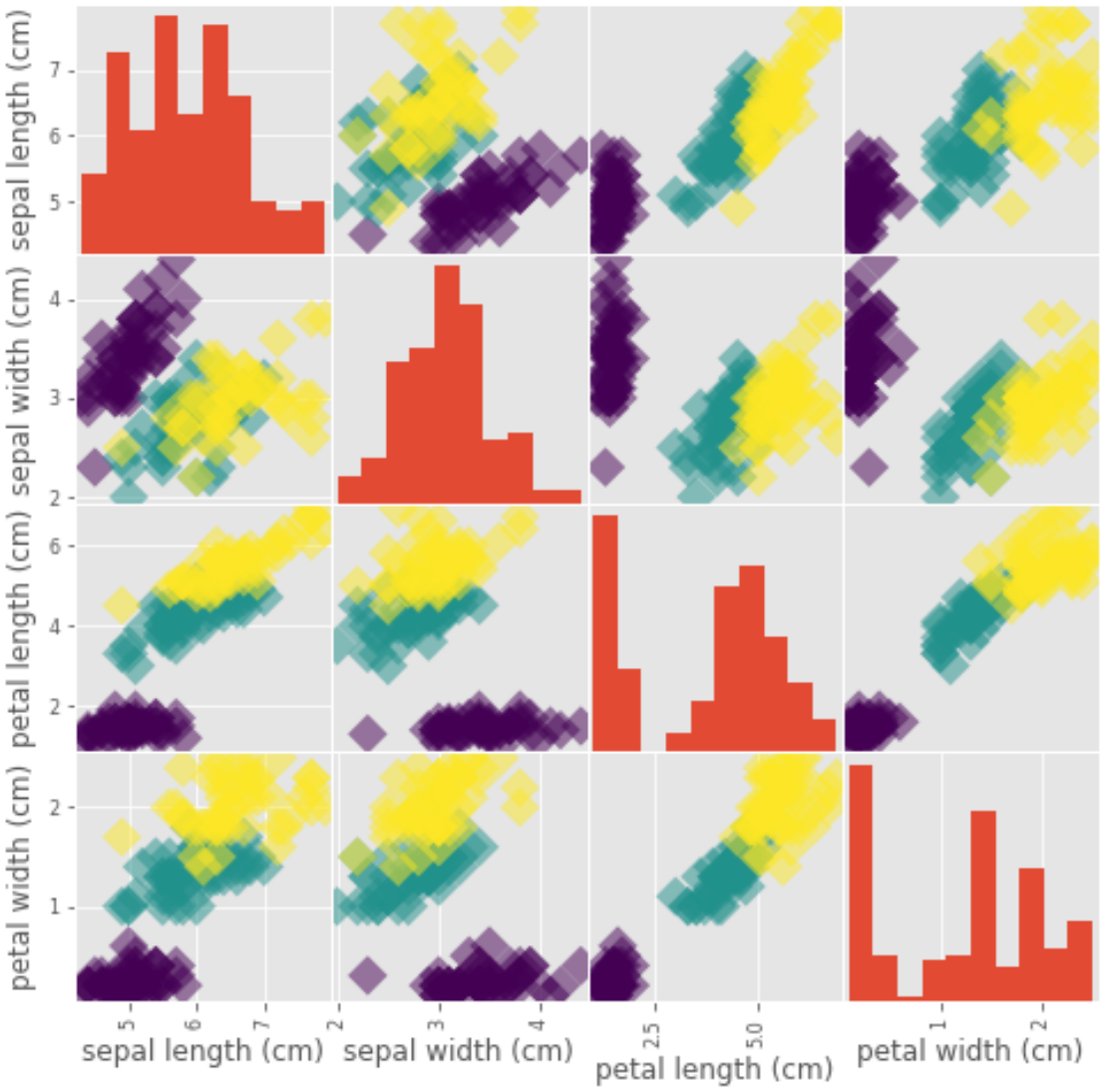
[6]:
```
import pandas as pd

X = iris.data
y = iris.target
df = pd.DataFrame(X, columns=iris.feature_names)
print(df.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2
```

[7]:
```
import matplotlib.pyplot as plt

plt.style.use('ggplot')
_ = pd.plotting.scatter_matrix(df, c=y, figsize=[8, 8], s=150,
                               marker='D')  # 'D' means diamond.
```



3. Practice exercises for exploratory data analysis (EDA):

▶ Data pre-loading:

[8]:
```
import pandas as pd

df = pd.read_csv(
    "https://archive.ics.uci.edu/ml/machine-learning-databases/voting-records/
 ↪house-votes-84.data",
    header=None)
df.columns = [
    'party', 'infants', 'water', 'budget', 'physician', 'salvador',
    'religious', 'satellite', 'aid', 'missile', 'immigration', 'synfuels',
    'education', 'superfund', 'crime', 'duty_free_exports', 'eaa_rsa'
```

```
]
df.replace(['y', 'n', '?'], [1, 0, 0.5], inplace=True)
```

▶ **Numerical EDA practice:**

[9]:
```
df.shape
```

[9]:
```
(435, 17)
```

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435 entries, 0 to 434
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   party               435 non-null    object 
 1   infants             435 non-null    float64
 2   water               435 non-null    float64
 3   budget              435 non-null    float64
 4   physician           435 non-null    float64
 5   salvador            435 non-null    float64
 6   religious           435 non-null    float64
 7   satellite           435 non-null    float64
 8   aid                 435 non-null    float64
 9   missile             435 non-null    float64
 10  immigration         435 non-null    float64
 11  synfuels            435 non-null    float64
 12  education           435 non-null    float64
 13  superfund           435 non-null    float64
 14  crime               435 non-null    float64
 15  duty_free_exports   435 non-null    float64
 16  eaa_rsa             435 non-null    float64
dtypes: float64(16), object(1)
memory usage: 57.9+ KB
```

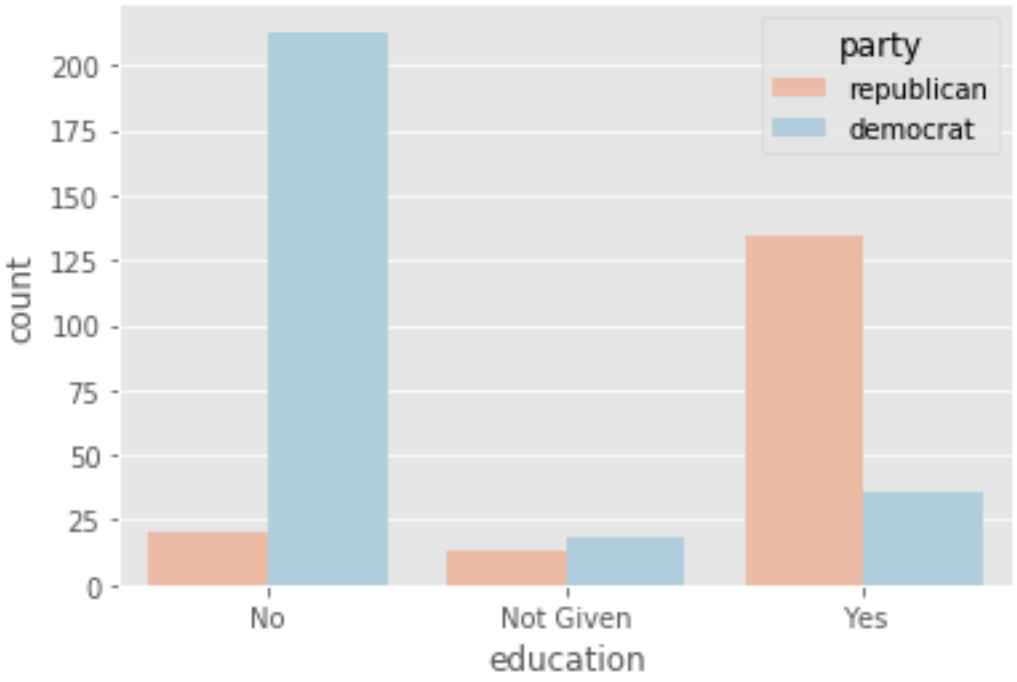
[11]:
```
df['party'].head()
```

[11]:
```
0    republican
1    republican
2      democrat
3      democrat
4      democrat
Name: party, dtype: object
```

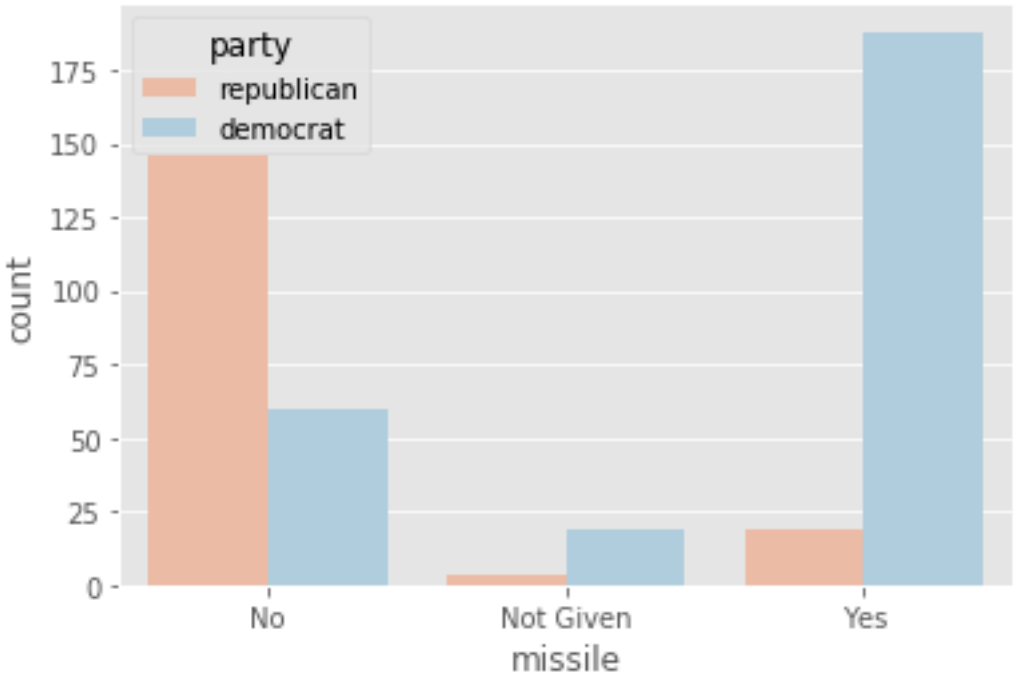
▶ **Visual EDA practice:**

[12]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure()
sns.countplot(x='education', hue='party', data=df, palette='RdBu')
plt.xticks([0, 1, 2], ['No', 'Not Given', 'Yes'])
plt.show()
```



[13]:
```
plt.figure()
sns.countplot(x='missile', hue='party', data=df, palette='RdBu')
plt.xticks([0, 1, 2], ['No', 'Not Given', 'Yes'])
plt.show()
```



[14]:
```
plt.figure()
sns.countplot(x='satellite', hue='party', data=df, palette='RdBu')
plt.xticks([0, 1, 2], ['No', 'Not Given', 'Yes'])
plt.show()
```