Bar Charts

- Bar plots are used to visualize a continuous variable versus a categorical variable. They provide a great way to visualize the magnitudes of a quantitative variable in terms of a qualitative variable.Depending on the software we used to create a bar plot, the height of the bars can show either the maximum value or the average value of the quantitative variable.

ADVANTAGES:

1. Summarize a large amount of data in a very interpretable way.
2. Easily readable by a large amount of audience.
3. Easy to display the contribution for multiple categories

DISADVANTAGES:

1. Sometimes need some extra explanation.
2. Fails to show the assumptions behind the data.

In [*]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

In [*]:
```
employee_ages = np.array([24, 25, 27, 26, 29, 28, 31, 30, 33, 32, 35, 34])

analyst_salary = np.array([27995, 35600, 52500, 36251, 38823, 63704,
                    66500, 51815, 75428, 56817,79248, 63259])

scientist_salary = np.array([55872, 49350, 59376, 64350, 46787, 73516,
                      76498, 59503, 80500, 81996, 85870, 94140])

developers_salary = np.array([48310, 34300, 33015, 36323, 38793, 42937,
                       45873, 51875, 77174, 58245, 79246, 64083])

colors = ['#CD6155','#5499C7','#AF7AC5','#48C9B0','#52BE80','#F4D03F']
```

In [*]:
```
plt.bar(employee_ages, analyst_salary, color = colors[3], ec = 'k')
plt.xlabel('Ages')
plt.ylabel('Salary')
plt.show()
```

In [*]:
```
indices = np.arange(len(employee_ages))
width = 0.25
```

In [*]:
```
plt.bar(indices, analyst_salary, color = colors[2], ec = 'k',
        label = 'Analyst', width = width)
plt.bar(indices + width , scientist_salary, color = colors[3], ec = 'k',
        label = 'Scientist', width = width)
plt.xlabel('Ages')
plt.ylabel('Salary')
plt.legend()
plt.xticks(ticks = indices, labels = employee_ages)
plt.show()
```

In [*]:
```
plt.figure(figsize = (8,6))
plt.bar(indices - width, developers_salary, color = colors[5], ec = 'k',
       label = 'Developer', width = width)
plt.bar(indices, analyst_salary, color = colors[2], ec = 'k',
        label = 'Analyst', width = width)
plt.bar(indices + width , scientist_salary, color = colors[3], ec = 'k',
        label = 'Scientist', width = width)
plt.xlabel('Ages')
plt.ylabel('Salary')
plt.legend()
plt.xticks(ticks = indices, labels = employee_ages)
plt.show()
```

In [*]:
```
salary = pd.read_csv(r'E:\Downloads\salary_cleaned.csv')
```

In [*]:
```
salary.head()
```

In [*]:
```
plt.barh(salary['occupation'], salary['age'], color = colors[0])
plt.xlabel('Age')
plt.ylabel('Occupation')
plt.show()
```

In [*]:
```
salary[['age','occupation']].groupby('occupation').max().sort_values(by = 'age')
```

In [*]:
```
plt.bar(employee_ages, analyst_salary, label = 'Analyst', color = colors[5])
plt.bar(employee_ages, scientist_salary, label = 'Scientist', color = colors[3],
       bottom = analyst_salary)
plt.bar(employee_ages, developers_salary, label = 'Developers', color = colors[1
       bottom = analyst_salary + scientist_salary)
plt.xlabel('Ages')
plt.ylabel('Salary')
plt.legend()
plt.show()
```

In [*]:
```
import seaborn as sns
```

In [*]:
```
iris = sns.load_dataset('iris')
```

In [*]:
```
iris.head()
```

In [*]:
```
plt.bar('Sepal Length', iris['sepal_length'].mean(), yerr = iris['sepal_length']
       color = colors[1])
plt.bar('Petal Length', iris['petal_length'].mean(), yerr = iris['petal_length']
       color = colors[3])
plt.bar('Sepal Width', iris['sepal_width'].mean(), yerr = iris['sepal_width'].st
       color = colors[5])
plt.bar('Petal Width', iris['petal_width'].mean(), yerr = iris['petal_width'].st
       color = colors[4])
plt.show()
```

In [ ]:
```

```