# 2. Descriptive Statistics

## Outline

- Frequency
- Measures of central tendency
- Measures of dispersion
- Normalization and Standardization
- Coefficients of correlation

## Frequency

資料常常需要計算出現的頻率， `.value_counts()` 可以統計某個欄位中每個值出現的次數。

In [ ]:

```
import pandas as pd
from pathlib import Path
data_folder = Path("../data/")

news = pd.read_csv(data_folder / "news.csv")
news.head()
```

In [ ]:

```
news['provider'].value_counts()
```

In [ ]:

```
word = '柯文哲'
news[word] = [word in text for text in news.content]
news[word].value_counts()
```

In [ ]:

```
word = '姚文智'
news[word] = [word in text for text in news.content]
pd.crosstab(news["柯文哲"], news["姚文智"])
```

In [ ]:

```
word = '民進黨'
news[word] = [text.count(word) for text in news.content]
news[word].value_counts()
```

## Measures of central tendency

可以使用 `.mode()` 得到眾數、 `.median()` 得到中位數、 `.mean()` 得到平均數。

In [ ]:

```
# mode
news['provider'].mode()
```

In [ ]:

```python
# count the news length
news['length'] = news['content'].apply(len)
```

In [ ]:

```python
# median
news['length'].median()
```

In [ ]:

```python
# mean
news['length'].mean()
```

## Measures of dispersion

可以用 `.max()` 得到最大值、 `.min()` 得到最小值、相減即為全距。
可以用 `.quantile()` 得到百分位數、 `.std()` 得到標準差、 `.var()` 得到變異數。
`.describe()` 則是數據表格的統計，包含平均數、標準差、最大最小值、中位數和四分位數。

In [ ]:

```python
# range
news.length.max() - news.length.min()
```

In [ ]:

```python
# Quantiles and quartiles
news.length.quantile(0.25)
```

In [ ]:

```python
# Standard deviation
news.length.std()
```

In [ ]:

```python
# Variance
news.length.var()
```

In [ ]:

```python
news.length.std() ** 2
```

In [ ]:

```python
news.describe()
```

## Normalization and Standardization

在建立模型前，通常會成資料標準化，常見的方法有下面兩種。
Normalization:
$x_{\text{norm}} = (x - x_{\text{min}})/(x_{\text{max}} - x_{\text{min}})$
$x_{\text{norm}}$'s are between 0 and 1.

Standardization:
$x_{\text{std}} = (x - \mu)/\sigma$
$x_{\text{std}}$'s have mean 0 and standard deviation 1.

In [ ]:

```python
news['length_norm'] = (news.length - news.length.min())/(news.length.max() - news.length.min())
news['length_std'] = (news.length - news.length.mean())/news.length.std()
```

In [ ]:

```
%matplotlib inline

news['length_norm'].hist()
```

In [ ]:

```
news['length_std'].hist()
```

In [ ]:

```
news['length'].hist()
```

## Coefficients of correlation

可以使用 `.corr()` 來看兩個欄位之間的相關係數（預設是 Pearson ，也可以用 Kendall 或Spearman 的方法）。

In [ ]:

```
news.loc[:,['柯文哲','姚文智','民進黨']].corr()
```