データの読み込み ¶

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
#  以上3つのライブラリが「3種の神器」
```

In [2]:
```
pd.read_csv("D:2022_数理統計学/StatData/StatData02_1.csv").head()  # データを読み込み、のぞき見
```

Out[2]:

	番号	身長	体重
0	1	46.0	2700
1	2	49.5	3220
2	3	50.0	3360
3	4	50.0	3500
4	5	49.0	3120

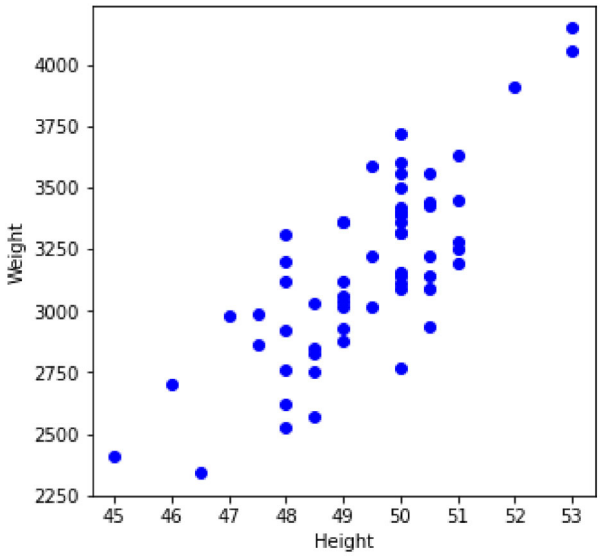
In [3]:
```
# csv ファイルの中身を見たうえで、前もって整形して読み込み
Data=pd.read_csv(
        "D:2022_数理統計学/StatData/StatData02_1.csv",
        skiprows=1,                          # 1行目を飛ばす
        names=['No', 'Height', 'Weight']   # カラム名は英語で
        )
del Data['No']   # No カラム不要なので削除
Data.head()
```

Out[3]:

	Height	Weight
0	46.0	2700
1	49.5	3220
2	50.0	3360
3	50.0	3500
4	49.0	3120

散布図

In [4]:
```
# 散布図
plt.figure(figsize=(5,5))                                    # 図の大きさ (6,4) がデフォルト
plt.scatter(Data['Height'],Data['Weight'],c='blue')          # 散布図
plt.xlabel('Height')                                         # x軸に変量を明記
plt.ylabel('Weight')                                         # y軸に変量を明記
# 画像ファイルを保存 (形式として, png jpeg eps など)
plt.savefig('StatData02_1_HW-SP.jpeg')                               # 画像ファイル(png)保存 (作業フォルダ
plt.savefig('D:2022_数理統計学/PNG/StatData02_1_HW-SP.png')  # 画像ファイル(png)保存 (パス指定)
```



統計量の計算

In [5]:
```
# 定義に戻って統計量の計算をする
Data['H^2']=Data['Height']**2
Data['W^2']=Data['Weight']**2
Data['HW']=Data['Height']*Data['Weight']
Data.head()
```

Out[5]:

	Height	Weight	H^2	W^2	HW
0	46.0	2700	2116.00	7290000	124200.0
1	49.5	3220	2450.25	10368400	159390.0
2	50.0	3360	2500.00	11289600	168000.0
3	50.0	3500	2500.00	12250000	175000.0
4	49.0	3120	2401.00	9734400	152880.0

In [6]:
```
# 平均値の計算
H_mean=np.mean(Data['Height'])  # 変量 Height の平均値
W_mean=np.mean(Data['Weight'])  # 変量 Weight の平均値
HH_mean=np.mean(Data['H^2'])    # 変量 H^2 の平均値
WW_mean=np.mean(Data['W^2'])    # 変量 W^2 の平均値
HW_mean=np.mean(Data['HW'])     # 変量 HW の平均値
print(H_mean, W_mean, HH_mean, WW_mean, HW_mean)
```

```
49.416666666666664 3172.8333333333335 2444.1916666666666 10200905.0 157215.08333333334
```

In [7]:
```
# 分散・共分散の計算
H_var=HH_mean-H_mean**2          # 変量 Height の分散
W_var=WW_mean-W_mean**2          # 変量 Weight の分散
HW_cov=HW_mean-H_mean*W_mean     # 変量 Height と Weight の共分散
print(H_var,W_var,HW_cov)
```

```
2.1847222222222626 134033.63888888806 424.23611111112405
```

In [8]:
```
# 標準偏差・相関係数の計算
H_std=np.sqrt(H_var)          # 変量 Height の標準偏差
W_std=np.sqrt(W_var)          # 変量 Weight の標準偏差
HW_corr=HW_cov/(H_std*W_std) # 変量 Height と Weight の相関係数
print(H_std, W_std, HW_corr)
```

```
1.478080587188081 366.1060486920259 0.7839757255555182
```

In [9]:
```
# 分散を与えるコマンドを使ってもよい.
np.var(Data['Height'])  # Python の数値計算は桁の先の方は正しくないので注意
```

Out[9]:
```
2.184722222222224
```

In [10]:
```
# 標準偏差を与えるコマンドを使ってもよい.
np.std(Data['Height'])  # Python の数値計算は桁の先の方は正しくないので注意
```

Out[10]:
```
1.478080587188068
```

In [11]:
```
# 共分散を与えるコマンドを使ってもよい
Data['Height'].cov(Data['Weight'],ddof=0)    # ddof=0 は必要
                                             # 省略すると ddof=1 となり、不偏分散に対応するものを計
```

Out[11]:
```
424.2361111111112
```

In [12]:
```
# 相関係数を与えるコマンドを使ってもよい
Data['Height'].corr(Data['Weight'])
```

Out[12]:
```
0.7839757255554989
```

In [13]:
```
# 基本統計量のまとめ
print(
    ['平均値', H_mean, W_mean],
    ['分散', H_var, W_var],
    ['標準偏差', H_std, W_std],
    ['共分散', HW_cov],
    ['相関係数', HW_corr],
    ['サイズ', len(Data['Weight'])]
    )
```

```
['平均値', 49.416666666666664, 3172.8333333333335] ['分散', 2.1847222222222626, 134033.638888888
06] ['標準偏差', 1.478080587188081, 366.1060486920259] ['共分散', 424.23611111112405] ['相関係
数', 0.7839757255555182] ['サイズ', 60]
```

In [14]:
```
# 無駄に長い小数表示を整理すること
StatSummary=pd.DataFrame([
    ['平均値', np.round(H_mean,2), np.round(W_mean,2)],
    ['分散', np.round(H_var,2), np.round(W_var,2)],
    ['標準偏差', np.round(H_std,2), np.round(W_std,2)],
    ['共分散', np.round(HW_cov,2)],
    ['相関係数', np.round(HW_corr,2)],
    ['サイズ', len(Data['Weight'])] ])
StatSummary=StatSummary.rename(columns={0:'統計量'})
StatSummary=StatSummary.rename(columns={1:'Height'})
StatSummary=StatSummary.rename(columns={2:'Weight'})
StatSummary
```

Out[14]:

	統計量	Height	Weight
0	平均値	49.42	3172.83
1	分散	2.18	134033.64
2	標準偏差	1.48	366.11
3	共分散	424.24	NaN
4	相関係数	0.78	NaN
5	サイズ	60.00	NaN

標準化されたデータの散布図

In [15]:
```
# 標準化して散布図を描く
NH=(Data['Height']-H_mean)/H_std   # 変量 Height の標準化を NH とした
NW=(Data['Weight']-W_mean)/W_std   # 変量 Weight の標準化を NW とした
```

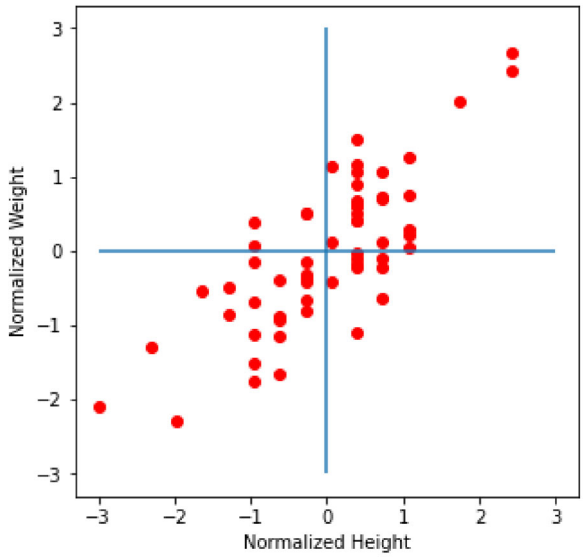
In [16]:
```
# DataFrame に追記
Data['H_normalized']=NH
Data['W_normalized']=NW
Data.head()
```

Out[16]:

	Height	Weight	H^2	W^2	HW	H_normalized	W_normalized
0	46.0	2700	2116.00	7290000	124200.0	-2.311556	-1.291520
1	49.5	3220	2450.25	10368400	159390.0	0.056379	0.128833
2	50.0	3360	2500.00	11289600	168000.0	0.394656	0.511236
3	50.0	3500	2500.00	12250000	175000.0	0.394656	0.893639
4	49.0	3120	2401.00	9734400	152880.0	-0.281897	-0.144312

In [17]:
```
# 標準化された変数に対して散布図を書く
plt.figure(figsize=(5,5))  # 図の大きさ (6,4) がデフォルト
plt.scatter(Data['H_normalized'],Data['W_normalized'],c='red')     #散布図を表示
plt.xlabel('Normalized Height')
plt.ylabel('Normalized Weight')
plt.hlines(0,-3,3)     # 直線 y=0 を -3<x<3 で図示
plt.vlines(0,-3,3)     # 直線 x=0 を -3<y<3 で図示
```

Out[17]: <matplotlib.collections.LineCollection at 0x27e887a2070>



In [ ]: