

# データの読み込み

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import linear_model      # 回帰分析用のライブラリ
```

```
In [2]: Data=pd.read_csv(
        "D:2022_数理統計学/StatData/StatData02_2.csv",
        skiprows=1, # 1行目を飛ばす
        names=['mid (x)', 'final (y)']) # カラム名称 (英語が好ましい)
Data
```

Out[2]:

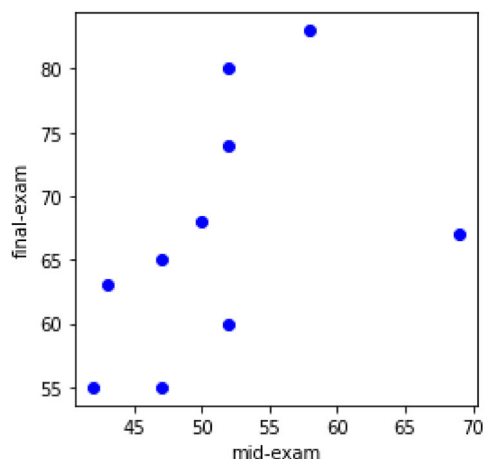
	mid (x)	final (y)
0	50	68
1	58	83
2	52	74
3	52	80
4	43	63
5	47	55
6	52	60
7	69	67
8	47	65
9	42	55

# 回帰分析

```
In [3]: x=np.c_[Data['mid (x)']]      #変量 mid (x) をコラムベクトル x として抽出
y=np.c_[Data['final (y)']]          #変量 final (y) をコラムベクトル y として抽出
```

```
In [4]: plt.figure(figsize=(4,4))    # 図の大きさ (6,4) がデフォルト
plt.scatter(x,y,c='blue')           # 散布図
plt.xlabel('mid-exam')
plt.ylabel('final-exam')
```

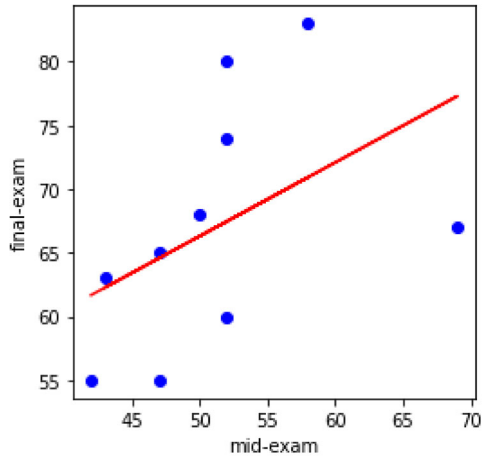
Out[4]: Text(0, 0.5, 'final-exam')



```
In [5]: model=linear_model.LinearRegression()
        model.fit(x, y)           # 回帰直線を求める
```

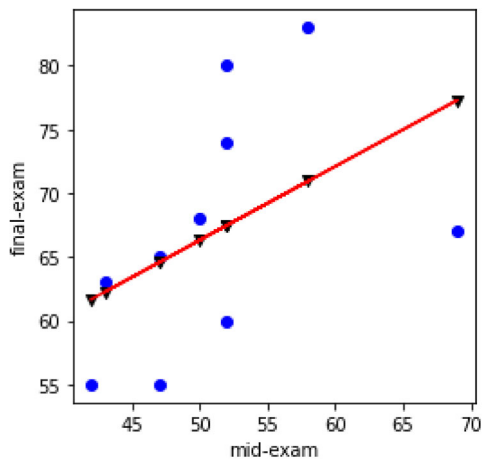
Out[5]: LinearRegression()

```
In [6]: plt.figure(figsize=(4,4)) # 図の大きさ (6,4) がデフォルト
        plt.scatter(x, y, c='blue') # 散布図を表示
        plt.xlabel('mid-exam')
        plt.ylabel('final-exam')
        plt.plot(x, model.predict(x), color = 'red') # 回帰直線
        plt.savefig('StatData02_2_MF-Regression.png') # 画像ファイル(png)として保存 (作業ディレクトリに)
```



```
In [7]: plt.figure(figsize=(4,4)) # 図の大きさ (6,4) がデフォルト
        plt.scatter(x, y, c='blue') # 散布図
        plt.scatter(x, model.predict(x), color = 'k', marker = 'v') # fitted value
        plt.plot(x, model.predict(x), color = 'red') # 回帰直線
        plt.xlabel('mid-exam')
        plt.ylabel('final-exam')
```

Out[7]: Text(0, 0.5, 'final-exam')



```
In [8]: p=model.coef_           # 回帰直線 y=px+q の傾き
        q=model.intercept_     # 回帰直線 y=px+q の切片
        print(p, q)
```

[[0.57803468]] [37.40462428]

```
In [9]: p=model.coef_          # 回帰直線 y=px+q の傾き
        q=model.intercept_    # 回帰直線 y=px+q の切片
        p=round(p[0][0], 2)
        q=round(q[0], 2)
        print('y=px+q', 'p='+str(p), 'q='+str(q))
```

y=px+q p=0.58 q=37.4

In [ ]: