1. (Gram–Schmidt process) 有一向量空間基底$S_1 = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n\}$, 從建構一組單範正交基底 $S_2 = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_n\}$, 其中

$$\boldsymbol{e}_i^T \boldsymbol{e}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

- 選取$\boldsymbol{u}_1 = \boldsymbol{v}_1$, 將其正規化, 得到第一個單位向量$\boldsymbol{e}_1 = \boldsymbol{u}_1/\|\boldsymbol{u}_1\|$
- 選取$\boldsymbol{v}_2$投影至子空間span$\{\boldsymbol{e}_1\}$, 得到殘差向量$\boldsymbol{u}_2$, 再正規化得$\boldsymbol{e}_2$.

$$\boldsymbol{u}_2 = \boldsymbol{v}_2 - (\boldsymbol{v}_2^T \boldsymbol{e}_1)\boldsymbol{e}_1 \xRightarrow[\text{正規化}]{} \boldsymbol{e}_2 = \boldsymbol{u}_2/\|\boldsymbol{u}_2\|$$

- 選取$\boldsymbol{v}_3$投影至子空間span$\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$, 得到殘差向量$\boldsymbol{u}_3$, 再正規化得$\boldsymbol{e}_3$.

$$\boldsymbol{u}_3 = \boldsymbol{v}_3 - (\boldsymbol{v}_3^T \boldsymbol{e}_1)\boldsymbol{e}_1 - (\boldsymbol{v}_3^T \boldsymbol{e}_2)\boldsymbol{e}_2 \xRightarrow[\text{正規化}]{} \boldsymbol{e}_3 = \boldsymbol{u}_3/\|\boldsymbol{u}_3\|$$

- 選取$\boldsymbol{v}_4$投影至子空間span$\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$, 得到殘差向量$\boldsymbol{u}_4$, 再正規化得$\boldsymbol{e}_4$.

$$\boldsymbol{u}_4 = \boldsymbol{v}_4 - (\boldsymbol{v}_4^T \boldsymbol{e}_1)\boldsymbol{e}_1 - (\boldsymbol{v}_4^T \boldsymbol{e}_2)\boldsymbol{e}_2 - (\boldsymbol{v}_4^T \boldsymbol{e}_3)\boldsymbol{e}_3 \xRightarrow[\text{正規化}]{} \boldsymbol{e}_4 = \boldsymbol{u}_4/\|\boldsymbol{u}_4\|$$

- …
- 選取$\boldsymbol{v}_r$投影至子空間span$\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_{r-1}\}$, 得到殘差向量$\boldsymbol{u}_r$, 再正規化得$\boldsymbol{e}_r$.

$$\boldsymbol{u}_r = \boldsymbol{v}_r - \sum_{i=1}^{r-1} (\boldsymbol{v}_r^T \boldsymbol{e}_i)\boldsymbol{e}_i \xRightarrow[\text{正規化}]{} \boldsymbol{e}_r = \boldsymbol{u}_r/\|\boldsymbol{u}_r\|$$

表格 1 Gram–Schmidt 算法

| | 殘差向量 | 正交向量 |
|---|---|---|
| $r = 1$ | $\boldsymbol{u}_1 = v_1$ | $e_1 = \dfrac{\boldsymbol{u}_1}{\|\boldsymbol{u}_1\|}$ |
| $2 \leq r \leq n$ | $\boldsymbol{u}_r = \boldsymbol{v}_r - \sum_{i=1}^{r-1} (\boldsymbol{v}_r^T \boldsymbol{e}_i)\boldsymbol{e}_i$ | $e_r = \dfrac{\boldsymbol{u}_r}{\|\boldsymbol{u}_r\|}$ |

更多細節可以參考這裡和這裡, 閱讀時要注意, 不同參考資料中, 變數名稱定義會有不同.
撰寫一 python 程式(hw1.py)實作 Gram–Schmidt process, 函數宣告如下:

```
def gram_schmidt(S1: np.ndarray):
    """
    Parameters
    ----------
    S1 : np.ndarray
        A m x n matrix with columns that need to be orthogonalized using Gram-
Schmidt process.
        It is assumed that vectors in S = [v1 v2 ... vn] are linear independent.

    Returns
    -------
    S2 : np.ndarray
        S2 = [e1 e2 ... en] is a mxn orthogonal matrix such that span(S1)=span(S2)

    """
```

```
    S2 = np.zeros (S1.shape)
    # write your code here
    return S2
```