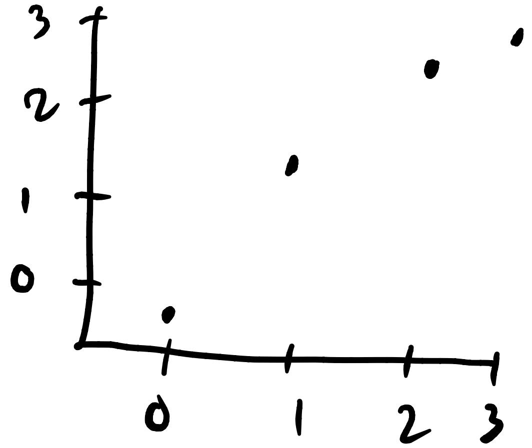
Example of PCA workflow by hand

X	Y
0	0.5
1	1.1
2	2.3
3	2.5



1) Normalize (Skip — assume data homogeneous)

2) Center data. $\mu_x = \frac{0+1+2+3}{4} = \frac{6}{4} = \frac{3}{2} = 1.5$

$\vec{\mu} = (\mu_x, \mu_y)$ $\quad \mu_y = \frac{0.5+1.1+2.3+2.5}{4} = \frac{6.4}{4} = 1.6$

X_c	Y_c
0 - 1.5	0.5 - 1.6
1 - 1.5	1.1 - 1.6
2 - 1.5	2.3 - 1.6
3 - 1.5	2.5 - 1.6

=

X_c	Y_c
-1.5	-1.1
-0.5	-0.5
0.5	0.7
1.5	0.9

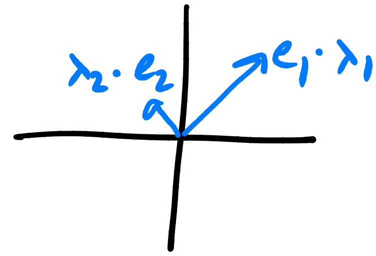
3) Compute covariance matrix : $\frac{A_c^T A}{N-1}$

$$\Sigma = \frac{1}{3}\begin{bmatrix} -1.5 & -0.5 & 0.5 & 1.5 \\ -1.1 & -0.5 & 0.7 & 0.9 \end{bmatrix}\begin{bmatrix} -1.5 & -1.1 \\ -0.5 & -0.5 \\ 0.5 & 0.7 \\ 1.5 & 0.9 \end{bmatrix} = \begin{bmatrix} 1.67 & 1.2 \\ 1.2 & 0.92 \end{bmatrix}$$

4) Compute eigenvals/vecs of Σ.

eigenvalues: $= \begin{array}{cc} \lambda_1 & \lambda_2 \\ \end{array}$ $[2.55, 0.04]$

eigenvectors: $P = \begin{bmatrix} 0.81 & -0.59 \\ 0.59 & 0.81 \end{bmatrix}$ with columns $\vec{e}_1$, $\vec{e}_2$



unit vectors (length = 1)
b/c Σ is symmetric — see spectral theorem coming soon

5) Sort eigenvalues high-to-low: $[2.55, 0.04]$

6) a) Compute proportion variance accounted for by each principal component (PC).

eigendimension

$$\boxed{\text{prop\_var}_i = \frac{\lambda_i}{\sum_{j=1}^{M} \lambda_j}} = \left[\frac{2.55}{2.55+0.04}, \frac{0.04}{2.55+0.04}\right]$$

$$= [\underbrace{0.98}, 0.02]$$

$\vec{e}_1$ accounts for 98% of variance

5 b) Compute Cumulative Variance accounted for by top K PCs

$$\text{Cum Prop Var} = \frac{\sum_{j=1}^{K} \lambda_j}{\sum_{j=1}^{M} \lambda_j} = \sum_{j=1}^{K} \text{prop-var-K}$$

1st PC accounts for 98% of Variance

always should ≈ 1.0

$$= \left[\frac{0.98}{0.98+0.02}, \frac{0.98+0.02}{0.98+0.02} \right] = \left[0.98, 1.0 \right]$$

$K=1$ $\quad$ $K=2$

top 2 PCs account for 100% of data Variance

6) Threshold out $\vec{e}_i$ corresponding to low cum-prop-var:

e.g. $\hat{P} = \begin{bmatrix} 0.81 \\ 0.59 \end{bmatrix}$ [could toss out $\vec{e}_2$].

For this example, we won't threshold (i.e. $\hat{P} = P$).

7. Project data into PCA space: $A_c = [\vec{x}_c \ \vec{y}_c]$

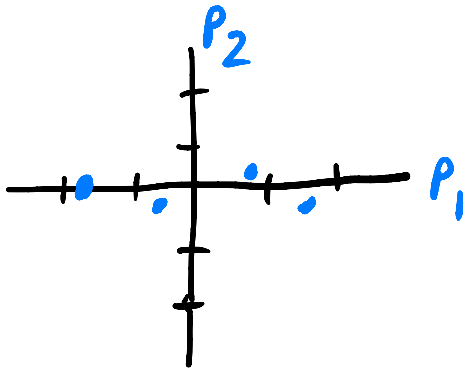
$$\hat{A}_c = \underbrace{A_c}_{(N,M)} @ \underbrace{\hat{P}}_{(M,K)}$$

K = # surviving dimensions

$K \leq M$

$$\hat{A}_c = \underbrace{\begin{bmatrix} -1.5 & -1.1 \\ -0.5 & -0.5 \\ 0.5 & 0.7 \\ 1.5 & 0.9 \end{bmatrix}}_{\substack{4 \times 2 \\ (N,M)}} \underbrace{\begin{bmatrix} 0.81 & -0.59 \\ 0.59 & 0.81 \end{bmatrix}}_{\substack{2 \times 2 \\ (M,K)}}$$

$$\hat{A}_c = \begin{bmatrix} \underbrace{\begin{matrix} -1.86 \\ -0.7 \\ 0.82 \\ 1.74 \end{matrix}}_{PC1} & \underbrace{\begin{matrix} 0 \\ -0.1 \\ 0.27 \\ -0.16 \end{matrix}}_{PC2} \end{bmatrix}$$



What would happen if we thresholded out $\vec{e}_2$?

— all data on x axis (P1 axis)

---

8. Reconstruct data from top K PCs.

$$A_R = \underbrace{\hat{A}_C}_{(N,K)} @ \underbrace{\hat{P}.T}_{\times(K,M)} + \vec{\mu}$$

$(M,)$

] want: A_R: (N, M) = shape(A)

- Note if $\hat{P} = P$ (no $\vec{e}_i$ dropped) $\Rightarrow \hat{A}_C = A_c @ P$

$$A_R = A_c @ \underbrace{P @ P.T}_{\substack{\text{Orthogonal} \\ \Rightarrow I}} + \vec{\mu}$$

$$A_R = \underbrace{A_c}_{\text{mean } \vec{0} \text{ — centered}} + \vec{\mu}$$

] recovers raw Data!

★ Need to Scale $\hat{A}_c @ \hat{P}.T$ by original data range if we normalize (step 1). Shape will look ok, but range will be incorrect.

- Show slides of reconstructed diamond data.