1) a) i) $AA^T = I$

$$A^T = A^{-1}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$a^2 + b^2 = 1$$

$$c^2 + d^2 = 1$$

$$ac + bd = 0$$

$$a = 1/\sqrt{2}$$

$$b = -1/\sqrt{2}$$

one has to be -ve

$$c = 1/\sqrt{2}$$

$$d = 1/\sqrt{2}$$

$$A = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

EIGENDECOMPOSITION:

$$Av = \lambda v$$

$$|A - \lambda I| = 0$$

$$\begin{vmatrix} 1/\sqrt{2} - \lambda & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} - \lambda \end{vmatrix} = 0$$

$$(1/\sqrt{2} - \lambda)^2 + 1/2 = 0$$

$$1/2 + \lambda^2 - 2\lambda/\sqrt{2} + 1/2 = 0$$

$$\lambda^2 - \sqrt{2}\lambda + 1 = 0$$

$$\lambda = \frac{\sqrt{2} \pm \sqrt{2 - 4}}{2} = \frac{\sqrt{2} \pm \sqrt{2}i}{2}$$

$$= 1/\sqrt{2} \pm i/\sqrt{2}$$

EIGENVALUES:

$$\lambda_1 = 1/\sqrt{2} + i/\sqrt{2} \qquad \lambda_2 = 1/\sqrt{2} - i/\sqrt{2}$$

$$A - \lambda_1 I = \begin{bmatrix} -i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & -i/\sqrt{2} \end{bmatrix} x = 0$$

$$-i/\sqrt{2}\, x_1 - 1/\sqrt{2}\, x_2 = 0$$

$$1/\sqrt{2}\, x_1 - i/\sqrt{2}\, x_2 = 0$$

$$\left.\begin{aligned} -ix_1 &= x_2 \\ x_1 &= ix_2 \end{aligned}\right]$$ SAME EQUATION

SO A PLANE OF POSSIBLE EIGENVECTORS.

ASSUME $x_2 = 1$

$\Rightarrow x_1 = i$

$$X_1 = \begin{bmatrix} i \\ 1 \end{bmatrix}$$

$$A - \lambda_2 I = \begin{bmatrix} i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & i/\sqrt{2} \end{bmatrix}$$

$$(A - \lambda_2 I)x = 0$$

$$ix_1 - x_2 = 0$$

$$x_1 + ix_2 = 0$$

$$\left.\begin{aligned} x_2 &= ix_1 \\ x_1 &= -ix_2 \end{aligned}\right]$$ SAME EQUATION

SO, A PLANE OF POSSIBLE EIGENVECTORS.

ASSUME $x_2 = 1$

$x_1 = -i$

$$X_2 = \begin{bmatrix} -i \\ 1 \end{bmatrix}$$

EIGENVECTORS:

$$x_1 = \begin{bmatrix} i \\ 1 \end{bmatrix} \qquad x_2 = \begin{bmatrix} -i \\ 1 \end{bmatrix}$$

OBSERVATIONS:

→ EIGENVALUES:

* Can be complex. When complex, they exist in complex conjugates.
* They have norm = 1.

→ EIGENVECTORS:

* They are orthogonal to each other if the eigenvalues are distinct.

$$x_1^H x_2 = \begin{bmatrix} -i & 1 \end{bmatrix} \begin{bmatrix} -i \\ 1 \end{bmatrix} = i^2 + 1 = 0 \,//$$

ii) $AA^T = I$ A is real

$$Ax = \lambda x \quad -①$$

$$(Ax)^H = (\lambda x)^H \longrightarrow \text{TAKE CONJUGATE TRANSPOSE}$$

$$x^H A^T = \bar{\lambda} x^H \quad -②$$

[$A^H = A^T$ as A is real]

[$\lambda^H = \bar{\lambda}$ as it is scalar]

$$AA^T = I \Rightarrow A^T = A^{-1} \Rightarrow A^T A = I$$

(left and right inverse of square matrices are same)

② × ①

$$x^H \underbrace{A^T A}_{I} x = \bar{\lambda} x^H \lambda x$$

$$x^H x = (\bar{\lambda}\lambda)\, x^H x$$

$$x^H x(1 - \bar{\lambda}\lambda) = 0$$

$$\|x\|^2 (1 - |\lambda|^2) = 0$$

WE ASSUME $\|x\|$ cannot be 0.

(non-zero eigenvectors).

Then $|\lambda|^2 = 1$

$|\lambda| = 1$ as it is always positive.

THEREFORE,

EIGENVALUES HAVE NORM = 1.

WE CAN ALSO ASSUME ALL ARE REAL:

$$Ax = \lambda x \quad - (1)$$

$$(Ax)^T = (\lambda x)^T$$

$$x^T A^T = \lambda x^T \quad - (2)$$

$$(2) \times (1)$$

$$x^T \underline{A^T} A x = |\lambda|^2 x^T x$$

$$\|x\|^2 = |\lambda|^2 \|x\|^2$$

↳ non-zero eigenvectors

$$|\lambda|^2 = 1$$

iii) $Ax_1 = \lambda_1 x_1$

$Ax_2 = \lambda_2 x_2$

$$x_1^H x_2 = x_1^H \cdot I \cdot x_2 \qquad [A^T A = A A^T = I]$$

$$= x_1^H \cdot A^T A \cdot x_2 \qquad [A^H = A^T]$$

$$= (Ax_1)^H (Ax_2)$$

$$= (\lambda_1 x_1)^H (\lambda_2 x_2)$$

$$= \bar{\lambda}_1 x_1^H \cdot \lambda_2 x_2$$

$$= \bar{\lambda}_1 \lambda_2 (x_1^H x_2)$$

$$(x_1^H x_2)(\bar{\lambda}_1 \lambda_2 - 1) = 0$$

EITHER $x_1^H x_2 = 0$

OR

$\bar{\lambda}_1 \lambda_2 = 1$

$$\left.\begin{aligned} \lambda_1 &= e^{i\theta} \\ \lambda_2 &= e^{i\phi} \end{aligned}\right\} |\lambda_1| = |\lambda_2| = 1 \text{, from (1 ii).}$$

$$\bar{\lambda}_1 \lambda_2 = (e^{-i\theta} \cdot e^{i\phi}) = 1$$

$$e^{i(\phi - \theta)} = 1$$

$$= e^0$$

$$\phi - \theta = 0$$

$$\phi = \theta$$

$$\Rightarrow \lambda_1 = e^{i\theta}$$

$$\lambda_2 = e^{i\theta}$$

$$\Rightarrow \lambda_1 = \lambda_2$$

BUT QUESTION SAYS DISTINCT EIGENVALUES

CONTRADICTION

$$\therefore \bar{\lambda}_1 \lambda_2 \neq 1$$

HENCE, $x_1^H x_2 = 0$ $\Rightarrow$ EIGENVECTORS CORRESPONDING TO DISTINCT EIGENVALUES ARE ORTHOGONAL //

WE CAN ALSO ASSUME ALL ARE REAL:

$A x_1 = \lambda_1 x_1$

$A x_2 = \lambda_2 x_2$

$$x_1^T x_2 = x_1^T \cdot I \cdot x_2 = x_1^T A^T A x_2 = (A x_1)^T (A x_2)$$

$$= (\lambda_1 x_1)^T (\lambda_2 x_2)$$

$$= \lambda_1 x_1^T \lambda_2 x_2$$

$$= \lambda_1 \lambda_2 x_1^T x_2$$

$$(\lambda_1 \lambda_2 - 1) x_1^T x_2 = 0$$

Since $\lambda_1 \neq \lambda_2$ and we know from (iii) that $|\lambda_1| = |\lambda_2| = 1$, $x_1^T x_2 = 0$ //

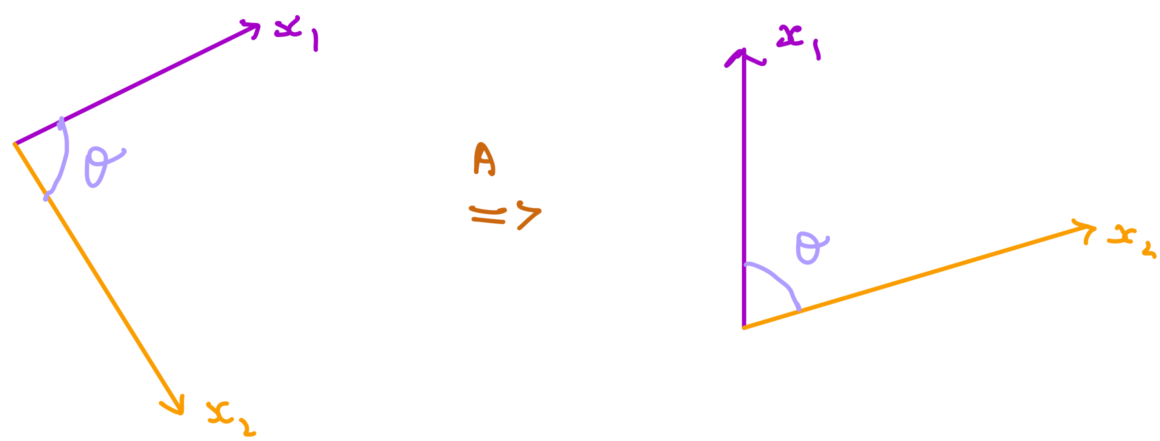
iv) Ax

THE ORTHOGONAL MATRIX PRESERVES THE INNER PRODUCT OF VECTORS AND HENCE IT DOES A ROTATION/ REFLECTION OF ANY VECTOR, WHEN APPLIED.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Any rotation transformation by angle θ, can be shown to be an orthogonal matrix.

It preserves length and angles between vectors.
↳ magnitude of x is maintained.



So for vector x,
→ magnitude / length is maintained
→ rotated / reflected by an angle //

b) i) $A = U \Sigma V^T$

$$AA^T = U \Sigma V^T (U \Sigma V^T)^T$$

$$= U \Sigma \underbrace{V^T V}_{\text{ORTHONORMAL } [V^T V = I]} \Sigma^T U^T$$

$$= U (\Sigma \Sigma^T) U^T$$

COMPARE THIS WITH EIGENDECOMPOSITION OF AA^T

$$AA^T = S \Lambda S^{-1}$$

So S here is U and $\underbrace{S^{-1} = U^T}$

As X here is AA^T, which is always a symmetric positive definite matrix, $S^{-1} = S^T$

Symmetric matrices have orthogonal eigenvectors. So if we normalise the eigenvectors, $S^{-1} = S^T$ and $X = S \Lambda S^T$.

Therefore U are the eigenvectors of AA^T.

$\Rightarrow$ Left singular vectors of A are the eigenvectors of AA^T.

$A = U \Sigma V^T$

$$A^T A = V \Sigma^T \underbrace{U^T U}_{\text{ORTHONORMAL}} \Sigma V^T$$

$$= V (\Sigma^T \Sigma) V^T$$

EIGENDECOMPOSITION :

$$A^T A = S \Lambda S^{-1} = S \Lambda S^T$$

$A^T A \rightarrow$ Symmetric positive definite.

$$S = V$$

$$\Lambda = \Sigma^T \Sigma$$

Therefore V are the eigenvectors of $A^T A$.

$\Rightarrow$ Right Singular vectors of A are the eigenvectors of $A^T A$.

ii)

$$A = U \Sigma V^T$$

$$AA^T = U \Sigma V^T (U \Sigma V^T)^T$$

$$= U \Sigma \underbrace{V^T V}_{} \Sigma^T U^T$$

ORTHONORMAL $[V^T V = I]$

$$= U (\Sigma \Sigma^T) U^T$$

COMPARE THIS WITH EIGENDECOMPOSITION OF AA^T

$$AA^T = S \Lambda S^{-1} = S \Lambda S^T$$ (SYMMETRIC POSITIVE DEFINITE)

$$\Lambda = \Sigma \Sigma^T = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix}$$

Therefore the square of singular values of A are the eigenvalues of AA^T.

$$A = U \Sigma V^T$$

$$A^T A = V \Sigma^T \underbrace{U^T U}_{\text{ORTHONORMAL}} \Sigma V^T$$

$$= V (\Sigma^T \Sigma) V^T$$

EIGENDECOMPOSITION :

$$A^TA = S\Lambda S^{-1} = S\Lambda S^T$$

$A^TA \rightarrow$ Symmetric positive definite.

$$\Lambda = \Sigma^T \Sigma = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix}$$

Therefore the square of singular values of A are the eigenvalues of A^TA as well. So A^TA and AA^T have the same eigenvalues //

or

The singular values of A are the square roots of the eigenvalues of AA^T and A^TA.

c) i) FALSE

Every linear operator in an n-dimensional vector space has at most n distinct eigenvalues.

eg: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is a linear operator in 2-D vector space.

$n = 2$

EIGENVALUES ARE (1, 1)

DISTINCT EIGENVALUES = 1 < 2 //

ii) FALSE

$Ax_1 = \lambda_1 x_1 \quad -①$

$Ax_2 = \lambda_2 x_2 \quad -②$

if $\lambda_1 = \lambda_2$

$A(x_1 + x_2) = \lambda(x_1 + x_2)$

$x_1 + x_2$ is an eigenvector

BUT IF $\lambda_1 \neq \lambda_2$ AND ASSUME $x_1 + x_2$ IS AN EIGENVECTOR, WITH EIGENVALUE, μ.

$A(x_1 + x_2) = \mu(x_1 + x_2)$

$Ax_1 + Ax_2 = \mu x_1 + \mu x_2$

$\lambda_1 x_1 + \lambda_2 x_2 = \mu x_1 + \mu x_2$ [FROM ①, ②]

$$x_1(\lambda_1 - \mu) + x_2(\lambda_2 - \mu) = 0 \quad - (3)$$

$$A \times (3) \Rightarrow Ax_1(\lambda_1 - \mu) + Ax_2(\lambda_2 - \mu) = 0$$

$$\lambda_1 x_1(\lambda_1 - \mu) + \lambda_2 x_2(\lambda_2 - \mu) = 0 \quad - (4)$$

$$\lambda_1 \times (3) \Rightarrow \lambda_1 x_1(\lambda_1 - \mu) + \lambda_1 x_2(\lambda_2 - \mu) = 0 \quad - (5)$$

$$(4) - (5)$$

$$(\lambda_2 - \lambda_1) x_2 (\lambda_2 - \mu) = 0.$$

$\lambda_1 \neq \lambda_2$ (ASSUMPTION)

$x_2 \neq 0$ (NON ZERO EIGENVECTORS)

NON-TRIVIAL.

So $\lambda_2 - \mu = 0$

$$\lambda_2 = \mu$$

$$\lambda_2 \times (3) \Rightarrow \lambda_2 x_1(\lambda_1 - \mu) + \lambda_2 x_2(\lambda_2 - \mu) = 0 \quad - (6)$$

$$(4) - (6)$$

$$\underbrace{(\lambda_1 - \lambda_2)}_{\neq 0} \overset{\neq 0}{x_1} (\lambda_1 - \mu) = 0$$

So $\lambda_1 = \mu$

So $\lambda_1 = \lambda_2 = \mu$ [CONTRADICTION]

THEREFORE, A NON ZERO SUM OF 2 EIGENVECTORS OF A MATRIX A IS NOT ALWAYS AN EIGENVECTOR.

eg: $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

$|A - \lambda I| = 0$

$$\begin{vmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{vmatrix} = 0$$

$(1-\lambda)^2 = 4$

$1-\lambda = 2$ $\qquad$ $1-\lambda = 2$

$\lambda_1 = -1$ $\qquad$ $\lambda_2 = 3$

$(A - \lambda_1 I)x = 0$ $\qquad$ $(A - \lambda_2)x = 0$

$$\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$x_1 + x_2 = 0$ $\qquad$ $2x_1 = 2x_2$

$x_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ $\qquad$ $x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$x_3 = x_1 + x_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$Ax_3 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

If $Ax_3 = \lambda_3 x_3$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2\lambda_3 \end{bmatrix}$$

$$4 \neq 0$$

HENCE PROVED

$x_1 + x_2 = x_3$ IS NOT AN EIGENVECTOR //

iii) TRUE

$x^T A x \geq 0$ $\forall x$, including eigenvectors

Let x_i be an eigenvector

$$x_i^T A x_i \geq 0$$

$$A x_i = \lambda_i x_i$$

$$x_i^T \lambda_i x_i \geq 0$$

$$\lambda_i x_i^T x_i \geq 0$$

$$\lambda_i \|x_i\|^2 \geq 0$$

$\hookrightarrow$ NORM ≥ 0

$$\Rightarrow \lambda_i \geq 0 \; //$$

iv) TRUE

The rank can exceed the number of distinct non-zero eigenvalues. eg:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

RANK = 2

EIGENVALUES = (1, 1)

NUMBER OF DISTINCT EIGENVALUES, $n = 1$.

RANK $> n$ //

v) TRUE

$A x_1 = \lambda x_1$

$A x_2 = \lambda x_2$

PROVE $x_1 + x_2$ is an eigenvector

$$A(x_1 + x_2) = A x_1 + A x_2 = \lambda x_1 + \lambda x_2 = \lambda(x_1 + x_2) \text{ //}$$

2) a) i) $P(H \mid M50) = 0.5$ $\quad P(T \mid M50) = 0.5$

$P(H \mid M60) = 0.6$ $\quad P(T \mid M60) = 0.4$

$P(M50) = P(M60) = 1/2$

$$P(M50 \mid T) = \frac{P(T \mid M50) \times P(M50)}{P(T)}$$

$$= \frac{P(T \mid M50) \times P(M50)}{P(T \mid M50) \times P(M50) + P(T \mid M60) \times P(M60)}$$

$$= \frac{0.5 \times 0.5}{0.5 \times 0.5 + 0.4 \times 0.5}$$

$$= 5/9$$

$$= 0.55 //$$

ii) $P(H50) = P(H60) = 1/2$

$$P(H50 \mid THHH) = \frac{P(THHH \mid H50) \cdot P(H50)}{P(THHH \mid H50) \cdot P(H50) + P(THHH \mid H60) \cdot P(H60)}$$

$$= \frac{(1/2)^4}{(1/2)^4 + (0.4)(0.6)^3}$$

$$= \frac{0.0625}{0.0625 + 0.0864}$$

$$= \frac{0.0625}{0.1489}$$

$= 0.4197$

iii) $$P(H50 \mid 9HIT) = \frac{P(9HIT \mid H50) \cdot P(H50)}{P(9HIT \mid H50) \cdot P(H50) + P(9HIT \mid H55) \cdot P(H55) + P(9HIT \mid H60) \cdot P(H60)}$$

$$= \frac{(0.5)^{10}}{(0.5)^{10} + (0.55)^9 (0.45) + (0.6)^9 (0.4)}$$

$$\approx 0.137931 //$$

$$P(H55 \mid 9HIT) = \frac{P(9HIT \mid H55) \cdot P(H55)}{P(9HIT)}$$

$$= \frac{(0.55)^9 (0.45)}{(0.5)^{10} + (0.55)^9 (0.45) + (0.6)^9 (0.4)}$$

$$= \frac{0.00207}{0.00708005862}$$

$$= 0.292711$$

$$P(H60 \mid 9HIT) = \frac{P(9HIT \mid H60) \cdot P(H60)}{P(9HIT)}$$

$$= \frac{(0.6)^9 (0.4)}{0.00708005862}$$

$$= 0.569356 //$$

b) $P(+ \mid Pg) = 99/100$

$P(- \mid Pg) = 1/100$

$P(+ / NP) = 10/100$

$P(- / NP) = 9/10$

$P(NP) = 99/100$

$P(Pg) = 1/100$

$$P(Pg \mid +) = \frac{P(+ \mid Pg) \cdot P(Pg)}{P(+ \mid Pg) \cdot P(Pg) + P(+ \mid NP) \cdot P(NP)}$$

$$= \frac{99/100 \times 1/100}{99/100 \times 1/100 + 10/100 \times 99/100}$$

$$= \frac{99}{99 + 990}$$

$$= \frac{99}{1089}$$

$$= 0.0909$$

The test is not that good with 10% failure when not pregnant. Moreover, the popululation has quite a huge majority (99%) of not pregnant women. Therefore the positive test results from non-pregnant women, though just 10% is still 10% of the 99% population and hence far outnumbers the total number of pregnant women.

Moreover the test has high false positive rate of 10%.

So assume 100,000 people. 99000 are not pregnant and 1000 are pregnant. 10% of 99,000 → 9900 are tested positive.

99% of 1000 => 990 are tested positive.

We can see that false positives far outnumber the true positive cases.

c) $E(\underbrace{A}_{m\times n}\underbrace{x}_{n\times 1}+\underbrace{b}_{m\times 1})$

$$E(x) = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_n) \end{bmatrix}$$

$$c = Ax + b = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \vdots \\ \vdots & & & \\ a_{m1} & & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + b_m \end{bmatrix}$$

$$c_i = \sum_{j=1}^{n} (a_{ij} x_j) + b_i$$

$$E[c_i] = E(a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + b_i)$$

$$= a_{i1}E(x_1) + a_{i2}E[x_2] + \cdots + a_{in}E(x_n) + b_i$$

$$E[c] = \begin{bmatrix} a_{11}E(x_1) + a_{12}E(x_2) \cdots + a_{1n}E(x_n) + b_1 \\ \vdots \\ a_{m1}E(x_1) + a_{m2}E(x_2) \cdots + a_{mn}E(x_n) + b_m \end{bmatrix}$$

$$E[c] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \vdots \\ \vdots & & & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix} \begin{bmatrix} E(x_1) \\ E(x_2) \\ \vdots \\ E(x_n) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$E[c] = A\,E(x) + b$$

$$E(Ax+b) = AE(x) + b \,//$$

OR

Without expanding the matrix, we can solve it

$$E(Ax_i + b_i) = E(Ax_i) + E(b_i)$$

$$= E\left(\sum_{j=1}^{n} A_{ij}\, x_j\right) + b_i$$

$$= \sum_{j=1}^{n} A_{ij}\, E(x_j) + b_i$$

$$= \sum_{j=1}^{n} A_{ij}\, E(x)_j + b_i$$

$$= [A\,E(x)]_i + b_i$$

$$E(Ax+b) = A\,E(x) + b \,//$$

d) $\text{cov}(x) = E[(x - Ex)(x - Ex)^T]$ — ①

$$\text{cov}(Ax+b) = E[(Ax+b - E(Ax+b))(Ax+b - E(Ax+b))^T]$$

$$= E[(Ax+b - AE(x) - b)(Ax+b - AE(x) - b)^T]$$

$$= E[A\underbrace{(x - E(x))(x - E(x))^T}A^T]$$

As A and A^T are deterministic, they can be taken out of the expectation like in question (c)

$$\text{cov}(Ax+b) = A \cdot E(x - E(x))(x - E(x))^T \cdot A^T$$

$$\text{cov}(Ax+b) = A \cdot \text{cov}(x) \cdot A^T \;//$$

3) a) $x \in \mathbb{R}^n, y \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$

$$\nabla_x x^T A y$$

LET $z = x^T A y$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \\ \vdots & & & \\ a_{n1} & \cdots & & a_{nm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11}y_1 + a_{12}y_2 + \cdots + a_{1m}y_m \\ \vdots \\ a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m \end{bmatrix}$$

$$= (a_{11}y_1 + a_{12}y_2 + \cdots + a_{1m}y_m)x_1 +$$
$$(a_{21}y_1 + a_{22}y_2 + \cdots + a_{2m}y_m)x_2 +$$
$$\vdots$$
$$(a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m)x_n$$

$$z = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} y_j) x_i$$

$$\frac{\partial z}{\partial x_i} = \sum_{j=1}^{m} a_{ij} y_j$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \sum_{j=1}^{m} a_{1j} y_j \\ \sum_{j=1}^{m} a_{2j} y_j \\ \vdots \\ \sum_{j=1}^{m} a_{nj} y_j \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \vdots \\ \vdots & & & \\ a_{n1} & \cdots & & a_{nm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= Ay$$

$$\nabla_x x^T A y = Ay$$

b) $x \in \mathbb{R}^n, y \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$

$\nabla_y x^T A y$

Let $z = x^T A y$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \\ \vdots & & & \\ a_{n1} & & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11}y_1 + a_{12}y_2 + \cdots + a_{1m}y_m \\ \vdots \\ a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m \end{bmatrix}$$

$$= (a_{11}y_1 + a_{12}y_2 + \dots + a_{1m}y_m)x_1 +$$
$$(a_{21}y_1 + a_{22}y_2 + \dots + a_{2m}y_m)x_2 +$$
$$\vdots$$
$$(a_{n1}y_1 + a_{n2}y_2 + \dots + a_{nm}y_m)x_n$$

$$z = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} y_j) x_i$$

$$\frac{\partial z}{\partial y_j} = \sum_{i=1}^{n} (a_{ij} x_i)$$

$$\frac{\partial z}{\partial y} = \begin{bmatrix} a_{11}x_1 + a_{21}x_2 + \cdots + a_{n1}x_n \\ a_{12}x_1 + a_{22}x_2 + \cdots + a_{n2}x_2 \\ \vdots \\ a_{1m}x_1 + a_{2m}x_2 + \cdots + a_{nm}x_n \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & & & \vdots \\ \vdots & & & \\ a_{1m} & \cdots & & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= A^T x$$

$$\nabla_y x^T A y = A^T x$$

c) $x \in \mathbb{R}^n, y \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$

$$\nabla_A \; x^T A y$$

LET $\quad z = x^T A y$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \\ \vdots & & & \\ a_{n1} & & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11}y_1 + a_{12}y_2 + \cdots + a_{1m}y_m \\ \vdots \\ a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m \end{bmatrix}$$

$$= (a_{11}y_1 + a_{12}y_2 + \ldots + a_{1m}y_m)x_1 +$$
$$(a_{21}y_1 + a_{22}y_2 + \ldots + a_{2m}y_m)x_2 +$$
$$\vdots$$
$$(a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m)x_n$$

$$z = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij} y_j) x_i$$

$$\frac{\partial z}{\partial a_{ij}} = x_i y_j$$

$$\frac{\partial y}{\partial A} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_m \\ x_2 y_1 & & & \vdots \\ \vdots & & & x_n y_m \\ x_n y_1 & \cdots & & \end{bmatrix}$$

$$= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}$$

$$= x y^T$$

$$\nabla_A x^T A y = x y^T \quad //$$

d) $f = x^T A x + b^T x$

Let $g = x^T A x$ and $h = b^T x$

$$g = [x_1 \quad x_2 \quad \cdots \quad x_n] \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \\ \vdots & & & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

if $i = j$

$$\frac{\partial g}{\partial x_i} = 2 a_{ii} x_i$$

if $i \neq j$

$$\frac{\partial g}{\partial x_i} = \sum_{\substack{j=1, \\ i \neq j}}^{n} a_{ij} x_j + \sum_{\substack{j=1 \\ i \neq j}}^{n} a_{ji} x_j$$

$$\frac{\partial g}{\partial x_i} = 2 a_{ii} x_i + \sum_{\substack{j=1, \\ i \neq j}}^{n} (a_{ij} + a_{ji}) x_j$$

$$= \sum_{j=1}^{n} (a_{ij} + a_{ji}) x_j$$

$$= \sum_{j=1}^{n} a_{ij} x_j + \sum_{j=1}^{n} a_{ji} x_j$$

$$\frac{\partial g}{\partial x} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{n1} & & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$+ \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ \vdots & & & \vdots \\ a_{1n} & & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= Ax + A^T x$$

$$h = b^T x$$

$$h = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= b_1 x_1 + \dots + b_n x_n$$

$$= \sum_{i=1}^{n} b_i x_i$$

$$\frac{\partial h}{\partial x_i} = b_i$$

$$\frac{\partial h}{\partial x} = b$$

$$\nabla_x f = \frac{\partial f}{\partial x} = \frac{\partial g}{\partial x} + \frac{\partial h}{\partial x} = (A + A^T)x + b \,//$$

e) $f = tr(AB)$

$$AB = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & & & \\ a_{n1} & & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \\ b_{m1} & \cdots & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1m}b_{m1} & \cdots & a_{11}b_{1n} + a_{12}b_{2n} + \dots + a_{1m}b_{mn} \\ & a_{21}b_{12} + a_{22}b_{21} + \dots + a_{2m}b_{m2} & \\ & \ddots & \\ & & a_{n1}b_{1n} + a_{n2}b_{2n} + \dots + a_{nm}b_{mn} \end{bmatrix}$$

$$tr(AB) = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} b_{ji}$$

$$\frac{\partial f}{\partial a_{ij}} = b_{ji}$$

$$\frac{\partial f}{\partial A} = B^T \quad //$$

4) $\hat{y} = Wx$

$$L = \frac{1}{2} \sum_{i=1}^{n} \| y^{(i)} - Wx^{(i)} \|^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} (y_i - Wx_i)^T (y_i - Wx_i)$$

Let us take

$$y - Wx = \begin{bmatrix} \vdots & \vdots & & \vdots \\ y_1 & y_2 & \cdots & y_n \\ \vdots & \vdots & & \vdots \end{bmatrix} - \begin{bmatrix} & W & \end{bmatrix} \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

$$(y - Wx)^T (y - Wx) = \begin{bmatrix} (y_1 - Wx_1)^T (y_1 - Wx_1) & (y_1 - Wx_1)^T (y_2 - Wx_2) & \cdots \\ & (y_2 - Wx_2)^T (y_2 - Wx_2) & \\ & & \ddots \\ & & & (y_n - Wx_n)^T (y_n - Wx_n) \end{bmatrix}$$

All the elements of L are along the diagonals of $(y - Wx)^T (y - Wx)$

$$L = \frac{1}{2} \operatorname{tr}\left[(y - Wx)^T (y - Wx) \right]$$

$$= \frac{1}{2} \operatorname{tr} \left[(y^T - x^T W^T)(y - Wx) \right]$$

$$= \frac{1}{2} \operatorname{tr} \left(y^T y - x^T W^T y - y^T Wx + x^T W^T Wx \right)$$

Find W to minimize L, $\frac{\partial L}{\partial W} = 0$

$$\frac{\partial L}{\partial W} = \frac{\partial}{\partial W} tr(y^T y) + \frac{\partial}{\partial W} tr(x^T W^T W x) - \frac{\partial}{\partial W} tr(x^T W^T y)$$
$$- \frac{\partial}{\partial W} tr(y^T W x)$$

WE KNOW THAT

$$tr(A^T) = tr(A)$$

$$tr(ABC) = tr(CAB) = tr(BCA)$$

$$\frac{\partial}{\partial W} tr(y^T y) = 0 \quad [\text{independent of } W]$$

$$\frac{\partial}{\partial W} tr(x^T W^T W x) = \frac{\partial}{\partial W} tr(x x^T W^T W) = \frac{\partial}{\partial W} tr(W x x^T W^T)$$

We know $\frac{\partial}{\partial W} tr(W A W^T) = W A^T + W A$

So $$\frac{\partial}{\partial W} tr(x^T W^T W x) = \frac{\partial}{\partial W} tr(W \underline{x x^T} W^T) = W (x x^T)^T + W x x^T$$

$$= W x x^T + W x x^T$$

$$= 2 W x x^T$$

$$\frac{\partial}{\partial W} tr(x^T W^T y) = \frac{\partial}{\partial W} tr(x^T W^T y)^T = \frac{\partial}{\partial W} tr(y^T W x)$$

$$= \frac{\partial}{\partial W} tr(x y^T W)$$

$$= \frac{\partial}{\partial W} tr(W x y^T)$$

$$= y x^T$$

$$\left[\text{FROM } \frac{\partial}{\partial W} tr(WA) = A^T \right]$$

$$\frac{\partial L}{\partial W} = 0 + 2Wxx^T - yx^T - yx^T$$

$$= 2Wxx^T - 2yx^T$$

$$\frac{\partial L}{\partial W} = 0$$

$$Wxx^T = yx^T$$

$$W = yx^T (xx^T)^{-1} \text{ //}$$

Linear regression workbook

This workbook will walk you through a linear regression example. It will provide familiarity with Jupyter Notebook and Python. Please print (to pdf) a completed version of this workbook for submission with HW #1.

ECE C147/C247 Winter Quarter 2022, Prof. J.C. Kao, TAs Y. Li, P. Lu, T. Monsoor, T. wang

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

#allows matlab plots to be generated in line
%matplotlib inline
```

Data generation

For any example, we first have to generate some appropriate data to use. The following cell generates data according to the model: $y = x - 2x^2 + x^3 + \epsilon$

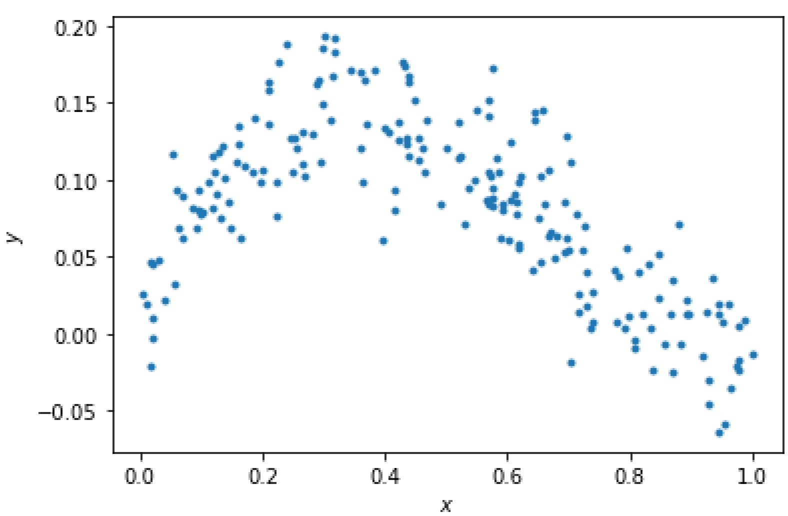
In [2]:

```
np.random.seed(0)  # Sets the random seed.
num_train = 200     # Number of training data points

# Generate the training data
x = np.random.uniform(low=0, high=1, size=(num_train,))
y = x - 2*x**2 + x**3 + np.random.normal(loc=0, scale=0.03, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

Out[2]:

```
Text(0, 0.5, '$y$')
```



QUESTIONS:

Write your answers in the markdown cell below this one:

(1) What is the generating distribution of x?

(2) What is the distribution of the additive noise ϵ?

ANSWERS:

(1) The generating distribution of x is Uniform Distribution. The parameters are a=0 and b=1.

(2) The distribution of the additive noise is Normal/Gaussian Distribution with mean 0 and standard deviation 0.03.

Fitting data to the model (5 points)

Here, we'll do linear regression to fit the parameters of a model $y = ax + b$.

In [3]:

```
# xhat = (x, 1)
xhat = np.vstack((x, np.ones_like(x)))

# ==================== #
# START YOUR CODE HERE #
# ==================== #
# GOAL: create a variable theta; theta is a numpy array whose elements are [a, b]
theta = np.linalg.inv(xhat @ xhat.T) @ (xhat @ y) # please modify this line

# ================== #
# END YOUR CODE HERE #
# ================== #
```

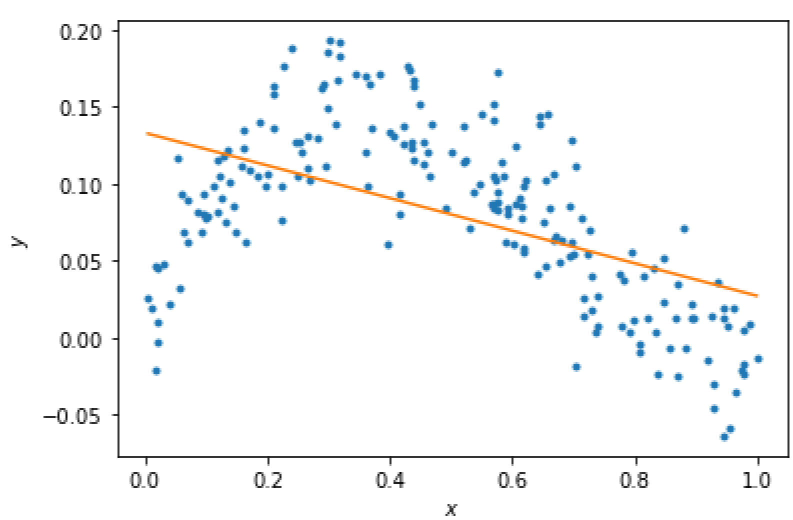
In [4]:

```
# Plot the data and your model fit.
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression line
xs = np.linspace(min(x), max(x),50)
xs = np.vstack((xs, np.ones_like(xs)))
plt.plot(xs[0,:], theta.dot(xs))
```

Out[4]:

```
[<matplotlib.lines.Line2D at 0x7faecaaa6040>]
```



QUESTIONS

(1) Does the linear model under- or overfit the data?

(2) How to change the model to improve the fitting?

ANSWERS

(1) The linear model underfits the data.

(2) Model is too simple. We need higher order polynomials to improve the fitting like a quadratic or a cubic model.

Fitting data to the model (10 points)

Here, we'll now do regression to polynomial models of orders 1 to 5. Note, the order 1 model is the linear model you prior fit.

In [5]:

```
N = 5
xhats = []
thetas = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable thetas.
# thetas is a list, where theta[i] are the model parameters for the polynomial fit of orde
#   i.e., thetas[0] is equivalent to theta above.
#   i.e., thetas[1] should be a length 3 np.array with the coefficients of the x^2, x, and
#   ... etc.

xhat = np.vstack((x, np.ones_like(x)))

for i in np.arange(1, N + 1):
    theta = np.linalg.inv(xhat @ xhat.T) @ (xhat @ y)
    xhats.append(xhat)
    thetas.append(theta)

    xhat = np.vstack((x ** (i+1), xhat))

# ================== #
# END YOUR CODE HERE #
# ================== #
```

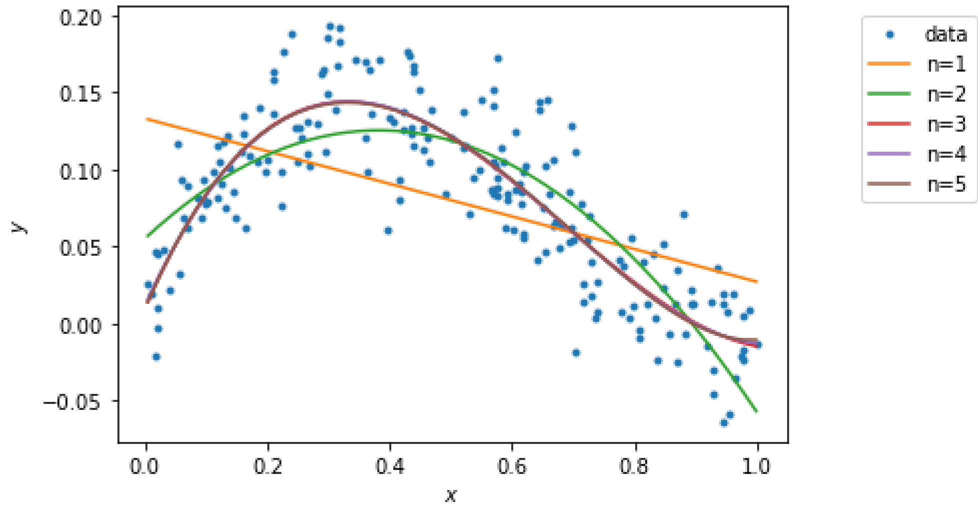
In [6]:

```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
    plot_xs.append(plot_x)

for i in np.arange(N):
    ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))
```

```
labels = ['data']
[labels.append('n={}'.format(i+1)) for i in np.arange(N)]
bbox_to_anchor=(1.3, 1)
lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
```



Calculating the training error (10 points)

Here, we'll now calculate the training error of polynomial models of orders 1 to 5:

$L(\theta) = \frac{1}{2}\sum_j(\hat{y}_j - y_j)^2$

In [7]:

```
training_errors = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable training_errors, a list of 5 elements,
# where training_errors[i] are the training loss for the polynomial fit of order i+1.
for i in np.arange(N):
    yi = thetas[i].dot(xhats[i])
    training_errors.append(np.sum(np.square(yi - y)) / 2)

# ================== #
# END YOUR CODE HERE #
# ================== #

print ('Training errors are: \n', training_errors)
```

```
Training errors are: 
 [0.2379961088362701, 0.10924922209268531, 0.08169603801105374, 0.0816535373529698, 0.0816
1479195525297]
```

QUESTIONS

(1) Which polynomial model has the best training error?

(2) Why is this expected?

ANSWERS

(1) Polynomial model of order 5 has the best training error.

(2) The model fits to the training data. So higher the order, the more the model dimensions and more the model fits to the training data and lesser the training error. This is because they have more parameters and hence better fit the data. But this could be a case of overfitting the training data and might give more error on the testing/validation data.

Generating new samples and validation error (5 points)

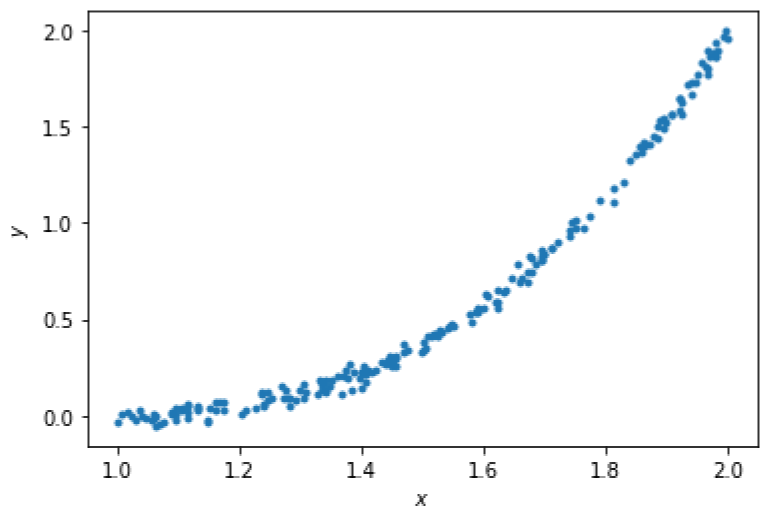
Here, we'll now generate new samples and calculate the validation error of polynomial models of orders 1 to 5.

In [8]:

```
x = np.random.uniform(low=1, high=2, size=(num_train,))
y = x - 2*x**2 + x**3 + np.random.normal(loc=0, scale=0.03, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

Out[8]:

```
Text(0, 0.5, '$y$')
```



In [9]:

```
xhats = []
for i in np.arange(N):
    if i == 0:
        xhat = np.vstack((x, np.ones_like(x)))
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        xhat = np.vstack((x**(i+1), xhat))
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))

    xhats.append(xhat)
```

In [10]:

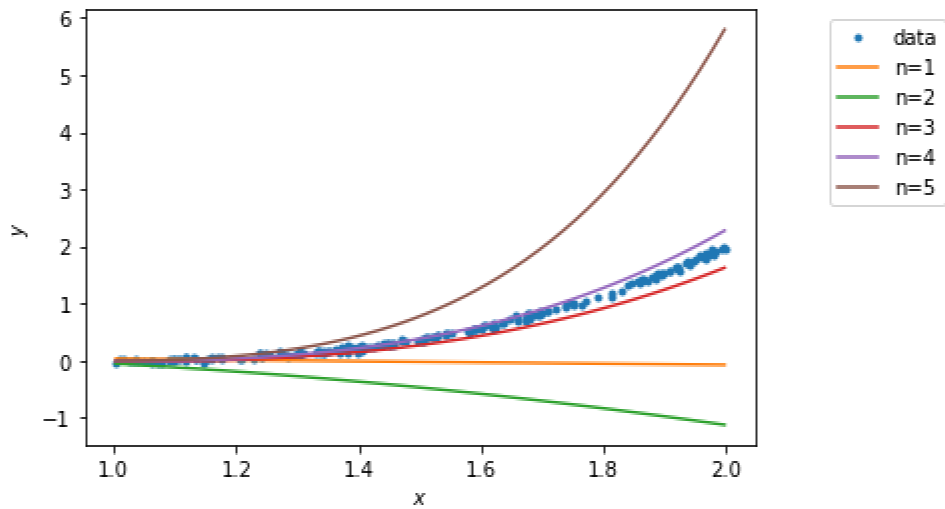
```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
```

```python
        else:
            plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
        plot_xs.append(plot_x)

    for i in np.arange(N):
        ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))

    labels = ['data']
    [labels.append('n={}'.format(i+1)) for i in np.arange(N)]
    bbox_to_anchor=(1.3, 1)
    lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
```



In [11]:

```python
validation_errors = []

# ==================== #
# START YOUR CODE HERE #
# ==================== #

# GOAL: create a variable validation_errors, a list of 5 elements,
# where validation_errors[i] are the validation loss for the polynomial fit of order i+1.
for i in np.arange(N):
    yi = thetas[i].dot(xhats[i])
    validation_errors.append(0.5 * np.sum((yi - y)**2))

# ================== #
# END YOUR CODE HERE #
# ================== #

print ('Validation errors are: \n', validation_errors)
```

```
Validation errors are: 
 [80.86165184550586, 213.19192445057962, 3.1256971082784704, 1.1870765196576452, 214.91021
81440583]
```

QUESTIONS

(1) Which polynomial model has the best validation error?

(2) Why does the order-5 polynomial model not generalize well?

ANSWERS

(1) Polynomial with order 4 has the best validation error.

(2) Polynomial with order-5 overfits to the given training data. The model not only fits to the given input data but to the noise as well. So when a different input is provided, it is not able to generalize well. Therefore, it doesn't generalize well to the validation data.

In [ ]:

```

```