Fall 2019, DS 5220
Homework 1
Mounica Subramani

Problem 1 :

(a) Expectation, $E(x) = \sum x \cdot p(x)$

$\Rightarrow 1 \times \frac{19}{100} + 2 \times \frac{12}{100} + 3 \times \frac{42}{100} + 4 \times \frac{16}{100} + 5 \times \frac{11}{100}$

$\Rightarrow \frac{288}{100} = \underline{\underline{2.88}}$ (we would expect them to rate their mood

(b) standard deviation $= \sqrt{\text{Variance}}$

$Var(x) = E(x^2) - [E(x)]^2$

$\Rightarrow 1^2 \times \frac{19}{100} + 2^2 \times \frac{12}{100} + 3^2 \times \frac{42}{100} + 4^2 \times \frac{16}{100} + 5^2 \times \frac{11}{100}$

$\Rightarrow 9.76$

$var(x) \Rightarrow 9.76 - (2.88)^2$

$\Rightarrow 1.4656.$

$\sigma = \sqrt{var(x)}$

$\underline{\underline{\sigma = 1.2106}}$. (expect the mood to deviate by 1)

(c) $P(\text{Happy} \;\&\; \text{Miserable}) = P(\text{happy}) \cdot P(\text{Miserable})$

$\Rightarrow \left(\frac{16}{100} + \frac{11}{100}\right) \cdot \frac{19}{100}$

$\Rightarrow \frac{27 \times 19}{100 \times 100}$

$\Rightarrow \cancel{0.0297}\ \underline{\underline{0.0513}}$

(d) $E(x) = \mu \qquad var(x) = \sigma$

$E[x(x-1)] = E[x^2 - x]$

$\Rightarrow \sum (x^2 - x) P(x)$

$\Rightarrow \sum x^2 P(x) - \sum x P(x)$

$\Rightarrow E(x^2) - E(x)$

$var(x) = \sigma$

$\Rightarrow E(x^2) - [E(x)]^2 = \sigma$

$\Rightarrow E(x^2) - \mu^2 = \sigma$

$\Rightarrow E(x^2) = \sigma + \mu^2$

$\Rightarrow \sigma + \mu^2 - \mu$

$\Rightarrow \sigma + \mu(\mu - 1)$

Problem 2 :

(a) $P(m|w) = \frac{P(w|m) \cdot P(m)}{P(w)}$ [Baye's]

w \ m	1	2	3	4	5	
G	7	4	20	12	7	50
B	12	8	22	4	4	50
	19	12	42	16	11	100

$E(m|w_G) = \sum x P(m|w_G)$

$\Rightarrow 1 \times 0.07 + 2 \times 0.04 + 3 \times 0.2 + 4 \times 0.12 + 5 \times 0.7$

$\Rightarrow 0.07 + 0.08 + 0.6 + 0.48 + 0.35$

$\Rightarrow \underline{\underline{1.58}}$

(b) $P(w_B|m_{1/2}) = \frac{P(m_{1 \text{ or } 2}|w_B)\, P(w_B)}{P(m_{1 \text{ or } 2})}$

$P(m_{1 \text{ or } 2}|w_B) = \frac{12}{19} + \frac{8}{12} \Rightarrow \frac{8}{19}$

$P(w_B) = \frac{50}{100} \Rightarrow \frac{1}{2}$

$P(m_{1 \text{ or } 2}) = \left[P(m_{1 \text{ or } 2}|w_G) * P(w_G) + P(m_{1 \text{ or } 2}) P(w_B)\right] P(w_B)$

$\Rightarrow \left(\frac{7}{19} * \frac{4}{12}\right) * \frac{1}{2} + \left(\frac{12}{19} * \frac{8}{12}\right) * \frac{1}{2}$

$\Rightarrow \frac{31}{100}$

$$P(w_B|m_{1\cup 2}) = \frac{\frac{8}{19} * \frac{1}{12}}{\frac{31}{100}} \Rightarrow \frac{20}{31} = \underline{0.645}$$

(c) $P(m=5 | w=B)$

$$P(m_5|w_B) = \frac{P(w_B|m_5)\,P(m_5)}{P(w_B)}$$

$$P(w_B|m_5) = \frac{4}{11}$$

$$P(m_5) = \frac{11}{100}$$

$$P(w_B) = P(w_B|m_1)P(m_1) + P(w_B|m_2)P(m_2) + P(w_B|m_3)P(m_3) + P(w_B|m_4)P(m_4) + P(w_B|m_5).P(m_5)$$

$$\Rightarrow \frac{7}{19} * \frac{19}{100} + \frac{4}{12} * \frac{12}{100} + \frac{20}{42} * \frac{42}{100} + \frac{12}{16} * \frac{16}{100} + \frac{7}{11} * \frac{11}{100}$$

$$\Rightarrow \frac{50}{100}$$

$$P(m_5|w_B) = \frac{\frac{4}{\cancel{11}} * \frac{\cancel{11}}{\cancel{100}}}{\frac{50}{\cancel{100}}} \Rightarrow \frac{4}{50} = \underline{0.08}$$

(d) $P(w=B | m=5)$

$$P(w_B|m_5) = \frac{P(m_5|w_B)\,P(w_B)}{P(m_5)}$$

$= 4/11$ (from part c).

(e) Total rating for mood $1 = 19$.

weather:	good	bad	total
mood 1:	7	12	19

$$\therefore P(w) = \frac{7}{19} \quad \frac{12}{19}$$

Naive Bayes can also be used.

In other words, the weather can be predicted just by dividing ~~by~~ the total rating of each mood.

Eg: predict weather when mood is 3.

The chance that weather is good when mood is 3 is $20/42$.

chance that weather is bad is $22/42$.

Problem 3

$$D_1 = \begin{bmatrix} 2 & 4 & 1 & 4 \\ 4 & 10 & 3 & 8 \\ 1 & 7 & 3 & 12 \\ 5 & 21 & 8 & 11 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & -2 \\ 1 & 0 & -3 & 1 \\ 1 & 0 & 3 & 0 \end{bmatrix}$$

(a) $D_1 = \begin{bmatrix} 2 & 4 & 1 & 4 \\ 4 & 10 & 3 & 8 \\ 1 & 7 & 3 & 12 \\ 5 & 21 & 8 & 11 \end{bmatrix}$

$R_2 \to R_2 - 2R_1$
$R_3 \to 2R_3 - R_1$ $\begin{bmatrix} 2 & 4 & 1 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 10 & 5 & 20 \\ 5 & 21 & 8 & 11 \end{bmatrix}$

$$R_3 \to R_3 - 5R_2 \begin{bmatrix} 2 & 4 & 1 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 20 \\ 5 & 21 & 8 & 11 \end{bmatrix} \xrightarrow[R_4 \to R_4 - 3R_1]{R_1 \to R_1 - R_2} \begin{bmatrix} 2 & 2 & 0 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 20 \\ -1 & 5 & 8 & -1 \end{bmatrix} \xrightarrow{R_4 \to R_4 - 7R_2} \begin{bmatrix} 2 & 2 & 0 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 20 \\ -1 & 1 & 1 & -1 \end{bmatrix}$$

$$R_4 \to 2R_4 + R_1 \begin{bmatrix} 2 & 2 & 0 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 20 \\ 0 & 4 & 2 & 2 \end{bmatrix} \xrightarrow{R_4 \to R_4 - 2R_2} \begin{bmatrix} 2 & 2 & 0 & 4 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 20 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Taking 2 out as common.

$$\Rightarrow \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & 1 & 1/2 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_4 \to 10R_4 - R_3} \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & 1 & 1/2 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The columns are linearly dependent for D1.

- $D_2 = \begin{bmatrix} 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & -2 \\ 1 & 0 & -3 & 1 \\ 1 & 0 & 3 & 0 \end{bmatrix}$

$$R_4 \to R_4 + R_3 \begin{bmatrix} 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & -2 \\ 1 & 0 & -3 & 1 \\ 2 & 0 & 0 & 1 \end{bmatrix} \xrightarrow[R_2 \to R_2 - 2R_4]{R_1 \to R_1 - R_4} \begin{bmatrix} 0 & 3 & 0 & 1 \\ 0 & 1 & 0 & -4 \\ 1 & 0 & -3 & 1 \\ 2 & 0 & 0 & 1 \end{bmatrix}$$

Row operation will not reduce the matrix any more.
The column vectors are linearly independent.

(b) Rank (D_1) = no. of non-zero rows

$= 3.$

Rank (D_2) = no. of non zero rows.

$= 4.$

(c) $\theta = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.8 \\ 1.2 \end{bmatrix}$

θD_2 cannot be computed as the dimensions of matrix do not compliment matrix multiplication.

Similarly $D_2\theta^T$ can also be not computed.

$$D_2\theta = \begin{bmatrix} 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & -2 \\ 1 & 0 & -3 & 1 \\ 1 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.2 \\ 0.8 \\ 1.2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1+0.6+0+1.2 \\ 2+0.2+0-2.4 \\ 0.5+0-2.4+1.2 \\ 0.5+0+2.4+0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 2.8 \\ -0.2 \\ -0.7 \\ 2.9 \end{bmatrix}$$

$$\theta^T D_2 = \begin{bmatrix} 0.5 & 0.2 & 0.8 & 1.2 \end{bmatrix} \begin{bmatrix} 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & -2 \\ 1 & 0 & -3 & 1 \\ 1 & 0 & 3 & 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1+0.8+0.8+1.2 & 1.5+0.2 & -2.4+3.6 & 0.5-1+0.8 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 3.8 & 1.7 & 1.2 & 0.3 \end{bmatrix}$$

(4) Given:

If. $A^T = A$, ~~and~~ A is symmetric.

To Prove:

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

(1) $A^{-1}A = AA^{-1} = I$ (hypothesis)

Proof:

$I = I^T$

Since, $AA^{-1} = I$.

$AA^{-1} = (AA^{-1})^T$

Since $(AB)^T = B^T A^T$

$AA^{-1} = (A^{-1})^T A^T$

Since $AA^{-1} = A^{-1}A$

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

given $A = A^T$, sub in above eq.

$A^{-1}\cancel{A} = (A^{-1})^T \cancel{A}$

$\therefore A^{-1} = (A^{-1})^T$

$\therefore$ The inverse of a symmetric matrix is also a symmetric matrix.

(b) $A = \begin{array}{c|c} & \begin{array}{ccccc} F & G & E & I & S \end{array} \\ \begin{array}{c} A \\ G \\ J \\ L \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}_{4\times 5}$

$A^T = \begin{array}{c|c} & \begin{array}{cccc} A & G & J & L \end{array} \\ \begin{array}{c} F \\ G \\ E \\ I \\ S \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}_{5\times 4}$.

$$AA^T = \begin{bmatrix} 1+1 & 1+0 & 0 & 1 \\ 1 & 1+1+1 & 1+1 & 1+1 \\ 0 & 1+1 & 1+1+1 & 1+1+1 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 2 & 2 \\ 0 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

AA^T signifies total number of languages spoken by Anton, Geraldine, James and Lauren.

$$A^TA = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 3 & 3 & 2 \\ 1 & 1 & 3 & 3 & 2 \\ 0 & 1 & 2 & 2 & 2 \end{bmatrix}$$

A^TA matrix signifies total number of people speaking each language (French, German, English, Italian and Spanish respectively.

(c)

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

A^{-1} exists if and only if $|A| \neq 0$.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

(5) Given:

$\text{cov}(x,y) = E[x - E(x)]E[y - E(y)]$ [Ref: clark University]

(a) If x and y are independent, $\text{cov}(x,y) = 0$.

Proof:

$\text{cov}(x,y) = E[x - \mu_x]E[y - \mu_y]$

$\text{var}(x)$ is just the $\text{cov}(x)$ with itself.

$\text{var}(x) = E[x - \mu_x]^2 = \text{cov}(x,x)$

$\text{var}(x) = E(x^2) - \mu_x^2$ (Identity for variance)

$\boxed{\text{cov}(x) = E(xy) - \mu_x\mu_y}$

Proof:

$\text{cov}(x,y) \Rightarrow E(x - \mu_x)E(y - \mu_y)$

$\Rightarrow E(xy - \mu_x y + \mu_x\mu_y - x\mu_y)$

$\Rightarrow E(xy) - \mu_x E(y) - E(x)\mu_y + \mu_x\mu_y$

$\mathrm{cov}(X,Y) = E(X,Y) - \mu_X\mu_Y$

Covariance can be positive, zero (or) negative.

* Positive indicates that there's overall tendency that when one variable increases so do the other.

* while negative indicates an overall tendency that when one increases, the other one decreases.

* If X and Y are independent variables, then their covariance is zero.

$$\Rightarrow E(X,Y) - \mu_X\mu_Y = 0.$$
$$E(X,Y) = \mu_X\mu_Y$$

(b) Converse is not true.

$\mathrm{cov}(X,Y) = 0$ for variables X and Y that are not independent.

Example: covariance is 0 but X and Y aren't independent, let there be three outcomes $(-1,1)$ $(0,-2)$ and $(1,1)$ all with same probability of $\frac{1}{3}$. They are clearly not independent since value of X determines value of Y.

Note that, $\mu_X = 0$ and $\mu_Y = 0$.

$$\mathrm{cov}(X,Y) = E[(X-\mu_X)(Y-\mu_Y)]$$
$$= E(XY)$$
$$= \frac{1}{3}(-1) + \frac{1}{3}(0) + \frac{1}{3}(1) = 0.$$

(c) $\mathrm{cov}(x_i - \bar{x}, \bar{x}) = 0$ (to prove)

$$\mathrm{cov}(x,y) = E(xy) - E(x)E(y)$$
$$\mathrm{cov}(x_i - \bar{x}, \bar{x}) = E[(x_i-\bar{x})(\bar{x})] - E[x_i - \bar{x}]E[\bar{x}]$$
$$\Rightarrow E[x_i\bar{x}] - E[\bar{x}^2] - E[x_i]E(\bar{x}) + E[\bar{x}]E[\bar{x}]$$

$$\left.\begin{array}{l} E(x_i) = \mu \\ \mathrm{var}(x_i) = \sigma^2 \end{array}\right\} \text{for all } i = 1,2\ldots n$$

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} \Rightarrow \frac{1}{n}(x_1 + x_2 + \ldots + x_n)$$

$$E(\bar{x}) = \frac{1}{n}E(x_1 + x_2 + \ldots + x_n) = \frac{\cancel{n}\mu}{\cancel{n}}$$

$$\mathrm{var}(\bar{x}) = \left(\frac{1}{n}\right)^2 \mathrm{var}(x_1 + x_2 + \ldots + x_n)$$

$$\Rightarrow \frac{1}{n\cancel{2}}\cancel{n}\sigma^2 = \frac{\sigma^2}{n}$$

continued

$$\text{cov}(ax+by, cz+dw) = ab\,\text{cov}(x,z) + bc\,\text{cov}(y,z) + ad\,\text{cov}(x,w) + bd\,\text{cov}(y,w)$$

$$\text{cov}(x_i - \bar{x}, \bar{x}) = \text{cov}(x_i, \bar{x}) - \text{cov}(\bar{x}, \bar{x})$$

$$\text{cov}(\bar{x}, \bar{x}) = \text{Var}(\bar{x}) = \frac{\sigma^2}{n}$$

$$\text{cov}(x_i, \bar{x}) = E(x_i, \bar{x}) - E(x_i)E(\bar{x})$$

$$\Rightarrow \text{cov}\left(x_i, \frac{x_1 + x_2 + \dots + x_n}{n}\right)$$

[$\text{cov}(x_i, x_1), \text{cov}(x_i, x_2), \text{cov}(x_i, x_3)$ .... are similar]

$$\Rightarrow \text{cov}\left(x_i, \frac{x_i}{n}\right)$$

$$\boxed{x_1 + x_2 + \dots + x_n = x_i}$$

$$\Rightarrow \frac{1}{n}\,\text{cov}(x_i, x_i)$$

$$\Rightarrow \frac{1}{n}\,\text{var}(x_i) \Rightarrow \frac{\sigma^2}{n}$$

$$\therefore \frac{\sigma^2}{n} - \frac{\sigma^2}{n} = 0.$$

$$\text{cov}(x_i - \bar{x}, \bar{x}) = 0$$

(6) $$f_{x,y}(x,y) = \frac{\exp\left[\frac{-1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]}{2\pi\sqrt{|\Sigma|}}$$

μ - mean vector $\begin{bmatrix}\mu_x \\ \mu_y\end{bmatrix}$

Σ - covariance matrix $= \begin{bmatrix}\sigma_x^2 & \sigma_x\sigma_y\rho \\ \rho\sigma_x\sigma_y & \sigma_y^2\end{bmatrix}$

$$\Sigma^{-1} = \frac{1}{\sigma_x^2\sigma_y^2 - \rho^2\sigma_x^2\sigma_y^2}\begin{bmatrix}\sigma_y^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_y^2\end{bmatrix}$$

$$|\Sigma| = \sigma_x^2\sigma_y^2 - \rho^2\sigma_x^2\sigma_y^2 \Rightarrow \sigma_x^2\sigma_y^2(1-\rho^2)$$

$$(x-\mu)^T = \begin{bmatrix}x-\mu_x \\ y-\mu_y\end{bmatrix}^T$$

$$\Sigma^{-1}(x-\mu) = \frac{1}{\sigma_x^2\sigma_y^2(1-\rho^2)}\underset{2\times 2}{\begin{bmatrix}\sigma_y^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_x^2\end{bmatrix}}\underset{2\times 1}{\begin{bmatrix}x-\mu_x \\ y-\mu_y\end{bmatrix}}$$

$$\Rightarrow \frac{1}{\sigma_x^2\sigma_y^2(1-\rho^2)}\underset{2\times 1}{\begin{bmatrix}\sigma_y^2(x-\mu_x) - \rho\sigma_x\sigma_y(y-\mu_y) \\ -\rho\sigma_x\sigma_y(x-\mu_x) + \sigma_x^2(y-\mu_y)\end{bmatrix}}$$

$$f_{x,y}(x,y) = \frac{\exp\left[-\frac{1}{2}\begin{bmatrix}x-\mu_x \\ x-\mu_y\end{bmatrix}^T \cdot \frac{1}{\sigma_x^2\sigma_y^2(1-\rho^2)}\begin{bmatrix}\sigma_y^2(x-\mu_x) - \rho\sigma_x\sigma_y(y-\mu_y) \\ \sigma_x^2(y-\mu_y) - \rho\sigma_x\sigma_y(x-\mu_x)\end{bmatrix}\right.}{2\pi\sqrt{\sigma_x^2\sigma_y^2(1-\rho^2)}}$$

$$\Rightarrow \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}}\exp\left[\frac{-1}{2\sigma_x^2\sigma_y^2(1-\rho^2)}\underset{1\times 2}{\begin{bmatrix}x-\mu_x & x-\mu_y\end{bmatrix}}\begin{bmatrix}\sigma_y^2(x-\mu_x) - \rho\sigma_x\sigma_y(y-\mu_y) \\ \sigma_x^2(y-\mu_y) - \rho\sigma_x\sigma_y(x-\mu_x)\end{bmatrix}\right.$$

$$\Rightarrow \frac{1}{2\pi.\sigma_x\sigma_y\sqrt{1-\rho^2}}\exp\left[\frac{-1}{2\sigma_x^2\sigma_y^2(1-\rho^2)}\left(\sigma_y^2(x-\mu_x)^2 - \rho\sigma_x\sigma_y(y-\mu_y)(x-\mu_x) + \sigma_x^2(y-\mu_y)^2 - \rho\sigma_x\sigma_y(x-\mu_x)(y-\mu_y)\right)\right.$$

$$\Rightarrow \frac{1}{2\pi.\sigma_x\sigma_y\sqrt{1-\rho^2}}\exp\left[\frac{-1}{2(1-\rho^2)}\left(\frac{\sigma_y^2(x-\mu_x)^2}{\sigma_x^2\sigma_y^2} - \frac{2\rho\sigma_x\sigma_y(y-\mu_y)(x-\mu_x)}{\sigma_x^2\sigma_y^2} + \frac{\sigma_x^2(y-\mu_y)^2}{\sigma_x^2\sigma_y^2}\right)\right.$$

~~$-\rho\sigma_x\sigma_y(x-\mu$~~

$$\therefore f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - 2\rho\frac{(x-\mu_x)}{\sigma_x}\frac{(y-\mu_y)}{\sigma_y}\right)\right]$$

(b) $Y_1 = 2X_1 + X_2$

$Y_2 = X_1 - X_2$.

Find $E(Y_1)$, $E(Y_2)$

$\text{cov}(Y_1, Y_2)$

f_{Y_1,Y_2} (Joint PDF)

Given:

X_1 and X_2 independent standard normal random variables.

with $N(0,1)$

$\mu = 0$.

$\sqrt{\text{var}(X_1)} = \sqrt{\text{var}(X_2)} = 1$.

$E(Y_1) = E(2X_1 + X_2)$

$= E(2(0)+0) = 0$.

$E(Y_2) = E(X_1 - X_2)$

$\Rightarrow E(0-0) = 0$.

$\text{var}(X_1) = E(X_1^2) - E(X_1)^2$

$1 = E(X_1^2) - 0$.

$\boxed{\therefore E(X_1^2) = 1}$

Similarly $E(X_2^2) = 1$.

$\text{cov}(Y_1, Y_2) = E[(Y_1 - E(Y_1))(Y_2 - E(Y_2))]$

$\Rightarrow E(Y_1 Y_2)$

$\Rightarrow E[(2X_1 + X_2)(X_1 - X_2)]$

$\Rightarrow E[2X_1^2 - 2X_1X_2 + X_1X_2 - X_2^2]$

$\Rightarrow E(2X_1^2) - E(X_1X_2) - E(X_2^2)$

$\Rightarrow 2 - 0 - 1 = 1$.

[$E(X_1X_2) = 0$ as X_1 and X_2 are independent R.V]

$\text{cov}(Y_1, Y_2) = 1$.

$\text{var}(Y_1) = E(Y_1^2) - E(Y_1)^2$

$\Rightarrow E(2X_1 + X_2)^2 - 0$.

$\Rightarrow E(4X_1^2 + 4X_1X_2 + X_2^2)$

$\Rightarrow E(4X_1^2) + E(4X_1X_2) + E(X_2^2)$

$\Rightarrow 4 + 0 + 1$

$\boxed{\text{Var}(Y_1) = 5}$

$\text{var}(Y_2) = E(Y_2^2) - E(Y_2)^2$

$= E(X_1 - X_2)^2 - 0$

$\Rightarrow E(X_1^2 - 2X_1X_2 + X_2^2)$

$\Rightarrow E(X_1^2) - E(2X_1X_2) + E(X_2^2)$

$\Rightarrow 1 - 0 + 1$

$\boxed{\text{var}(Y_2) = 2.}$

Joint PDF.

required

$$\rho = \frac{cov(y, Y_2)}{\sigma_{Y_1}\sigma_{Y_2}}$$

$$\Sigma = \begin{bmatrix} \sigma_{Y_1}^2 & \rho\sigma_{Y_1}\sigma_{Y_2} \\ \rho\sigma_{Y_1}\sigma_{Y_2} & \sigma_{Y_2}^2 \end{bmatrix}$$

$\mu_{Y_1} = E(Y_1) = 0.$

$\mu_{Y_2} = E(Y_2) = 0.$

$\sigma_{Y_1} = \sqrt{Var(Y_1)} = \sqrt{5}.$

$\sigma_{Y_2} = \sqrt{Var(Y_2)} = \sqrt{2}.$

$$\rho = \frac{1}{\sqrt{10}}$$

$$f_{Y_1,Y_2}(y_1,y_2) = \frac{1}{2\pi\sqrt{10}\left(\sqrt{1-\frac{1}{10}}\right)} \exp\left[-\frac{1}{2(1-1/10)}\frac{(Y_1-0)^2}{5} + \frac{(Y_2-0)^2}{2} - \frac{2}{10}\frac{(Y_1-0)}{\sqrt{5}}\frac{(Y_2-0)}{\sqrt{2}}\right]$$

$$\Rightarrow \frac{1}{2\pi\cancel{\sqrt{10}}\cdot\sqrt{\frac{9}{\cancel{10}}}} \exp\left[-\frac{1}{\cancel{2}\cdot\frac{9}{\cancel{10}_{5}}}\frac{y_1^2}{5} + \frac{Y_2^2}{2} - \frac{1}{5}\frac{Y_1}{\sqrt{5}}\frac{Y_2}{\sqrt{2}}\right]$$

$$\Rightarrow \frac{1}{6\pi}\exp\left[-\frac{5}{9}\left(\frac{Y_1^2}{5}+\frac{Y_2^2}{2}\right) - \frac{1}{5}\left(\frac{Y_1 Y_2}{\sqrt{10}}\right)\right]$$

SML_HW1_MounicaSubramani

September 23, 2019

[78]:
```
import pandas as pd
import statistics
import numpy as np
import matplotlib.pyplot as mp
```

[84]:
```
KChouse = pd.read_csv("C:/Users/mouni/Downloads/kc_house_data.csv")
KChouse = KChouse.drop(columns = ['id','zipcode','date'])
print(KChouse.head())
```

```
      price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  waterfront  \
0  221900.0         3       1.00         1180      5650     1.0           0
1  538000.0         3       2.25         2570      7242     2.0           0
2  180000.0         2       1.00          770     10000     1.0           0
3  604000.0         4       3.00         1960      5000     1.0           0
4  510000.0         3       2.00         1680      8080     1.0           0

   view  condition  grade  sqft_above  sqft_basement  yr_built  yr_renovated  \
0     0          3      7        1180              0      1955             0
1     0          3      7        2170            400      1951          1991
2     0          3      6         770              0      1933             0
3     0          5      7        1050            910      1965             0
4     0          3      8        1680              0      1987             0

       lat     long  sqft_living15  sqft_lot15
0  47.5112 -122.257           1340        5650
1  47.7210 -122.319           1690        7639
2  47.7379 -122.233           2720        8062
3  47.5208 -122.393           1360        5000
4  47.6168 -122.045           1800        7503
```

For each feature, write code to compute the average value, the min and max values, as well as its variance. Which features have the lowest and highest variance?

[85]:
```
print("Average Value of each feature")
KChouse.mean()
```

```
Average Value of each feature
```

[85]:
```
price            540182.158793
bedrooms              3.370842
bathrooms             2.114757
sqft_living        2079.899736
sqft_lot          15106.967566
floors                1.494309
waterfront            0.007542
view                  0.234303
condition             3.409430
grade                 7.656873
sqft_above         1788.390691
sqft_basement       291.509045
yr_built           1971.005136
yr_renovated         84.402258
lat                  47.560053
long               -122.213896
sqft_living15      1986.552492
sqft_lot15        12768.455652
dtype: float64
```

[86]:
```python
print("Minimum Value of each feature")
KChouse.min()
```

```
Minimum Value of each feature
```

[86]:
```
price            75000.0000
bedrooms             0.0000
bathrooms            0.0000
sqft_living        290.0000
sqft_lot           520.0000
floors               1.0000
waterfront           0.0000
view                 0.0000
condition            1.0000
grade                1.0000
sqft_above         290.0000
sqft_basement        0.0000
yr_built          1900.0000
yr_renovated         0.0000
lat                 47.1559
long              -122.5190
sqft_living15      399.0000
sqft_lot15         651.0000
dtype: float64
```

[87]:
```python
print("Maximum Value of each feature")
KChouse.max()
```

```
Maximum Value of each feature
```

[87]:
```
price            7.700000e+06
bedrooms         3.300000e+01
bathrooms        8.000000e+00
sqft_living      1.354000e+04
sqft_lot         1.651359e+06
floors           3.500000e+00
waterfront       1.000000e+00
view             4.000000e+00
condition        5.000000e+00
grade            1.300000e+01
sqft_above       9.410000e+03
sqft_basement    4.820000e+03
yr_built         2.015000e+03
yr_renovated     2.015000e+03
lat              4.777760e+01
long            -1.213150e+02
sqft_living15    6.210000e+03
sqft_lot15       8.712000e+05
dtype: float64
```

[88]:
```
print("Variance of each feature")
var = KChouse.var()
var
```

```
Variance of each feature
```

[88]:
```
price            1.349550e+11
bedrooms         8.650150e-01
bathrooms        5.931513e-01
sqft_living      8.435337e+05
sqft_lot         1.715659e+09
floors           2.915880e-01
waterfront       7.485226e-03
view             5.872426e-01
condition        4.234665e-01
grade            1.381703e+00
sqft_above       6.857347e+05
sqft_basement    1.958727e+05
yr_built         8.627973e+02
yr_renovated     1.613462e+05
lat              1.919990e-02
long             1.983262e-02
sqft_living15    4.697612e+05
sqft_lot15       7.455182e+08
dtype: float64
```

[89]:
```
print("Feature with highest variance")
var.idxmax(axis =1)
```

```
Feature with highest variance
```

[89]:
```
'price'
```

[90]:
```python
print("Feature with lowest variance")
var.idxmin(axis =1)
```

```
Feature with lowest variance
```

[90]:
```
'waterfront'
```

Compute the correlation coefficient of each feature with the response. Which feature are positively correlated (i.e., have positive correlation coefficient) and which ones are negatively correlated with the response? Which feature has highest correlation with the response (either positive or negative)?

[92]:
```python
print("Correlation matrix for the KC housing dataset")
cor_mat = KChouse.corr()
print(cor_mat.head())
```

```
Correlation matrix for the KC housing dataset
                price  bedrooms  bathrooms  sqft_living  sqft_lot    floors  \
price        1.000000  0.308338   0.525134     0.702044  0.089655  0.256786
bedrooms     0.308338  1.000000   0.515884     0.576671  0.031703  0.175429
bathrooms    0.525134  0.515884   1.000000     0.754665  0.087740  0.500653
sqft_living  0.702044  0.576671   0.754665     1.000000  0.172826  0.353949
sqft_lot     0.089655  0.031703   0.087740     0.172826  1.000000 -0.005201

             waterfront      view  condition     grade  sqft_above  \
price          0.266331  0.397346   0.036392  0.667463    0.605566
bedrooms      -0.006582  0.079532   0.028472  0.356967    0.477600
bathrooms      0.063744  0.187737  -0.124982  0.664983    0.685342
sqft_living    0.103818  0.284611  -0.058753  0.762704    0.876597
sqft_lot       0.021604  0.074710  -0.008958  0.113621    0.183512

             sqft_basement  yr_built  yr_renovated       lat      long  \
price             0.323837  0.053982      0.126442  0.306919  0.021571
bedrooms          0.303093  0.154178      0.018841 -0.008931  0.129473
bathrooms         0.283770  0.506019      0.050739  0.024573  0.223042
sqft_living       0.435043  0.318049      0.055363  0.052529  0.240223
sqft_lot          0.015286  0.053080      0.007644 -0.085683  0.229521

             sqft_living15  sqft_lot15
price             0.585374    0.082456
bedrooms          0.391638    0.029244
bathrooms         0.568634    0.087175
sqft_living       0.756420    0.183286
sqft_lot          0.144608    0.718557
```

[96]:
```
corr_with_price = cor_mat[:1]

print("drop price as it has to be predicted\n")
KC_price_dropped = corr_with_price.drop(columns = 'price')

print(KC_price_dropped)
```

```
drop price as it has to be predicted

       bedrooms  bathrooms  sqft_living  sqft_lot    floors  waterfront  \
price  0.308338   0.525134     0.702044  0.089655  0.256786    0.266331

           view  condition     grade  sqft_above  sqft_basement  yr_built  \
price  0.397346   0.036392  0.667463    0.605566       0.323837  0.053982

       yr_renovated       lat      long  sqft_living15  sqft_lot15
price      0.126442  0.306919  0.021571       0.585374    0.082456
```

Every feature in the data set is positively correlated and there is no negative correlation.

[97]:
```
print("Feature that has highest correlation with the response")
KC_price_dropped.idxmax(axis = 1, skipna = True)
```

```
Feature that has highest correlation with the response
```

[97]:
```
price    sqft_living
dtype: object
```

Let X be a random variable with normal distribution N(0, 2). Sample at random 50 points from the same normal distribution. Define Y = 3 + 2X + error, where error is normal noise drawn from N(0,). Plot Y as a function of X for several values of {0, 1, 4} and compute the correlation coefficient for each case.

[60]:
```
x = np.random.normal(0,2,50)
noise1 = np.random.normal(0,0,50)
noise2 = np.random.normal(0,1,50)
noise3 = np.random.normal(0,4,50)
```

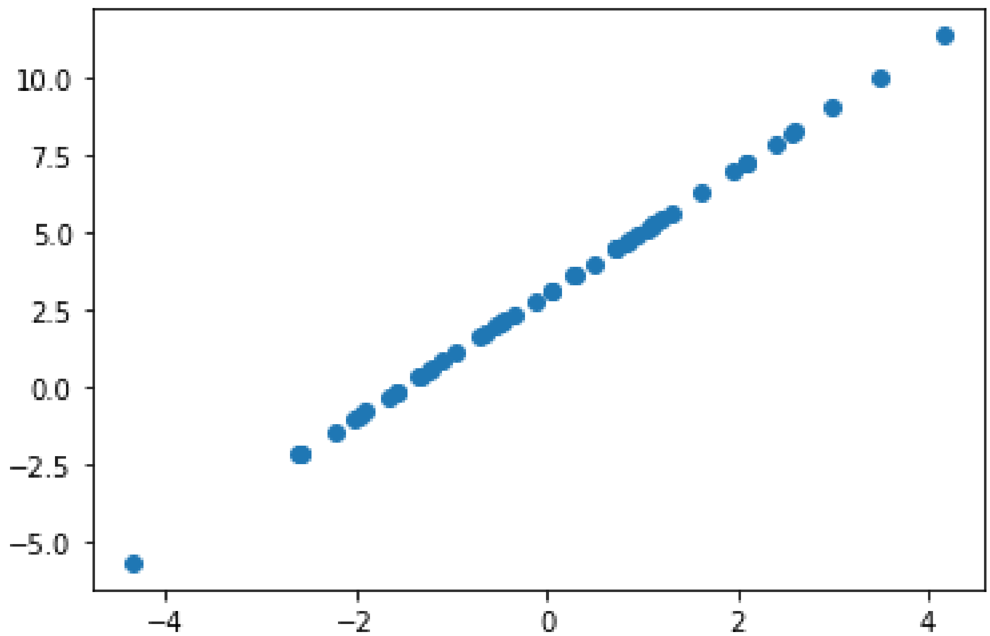
[58]:
```
x
```

[58]:
```
array([-3.11750605,  1.57799316, -0.75875675,  1.01889526,  0.315232  ,
       -2.20606866,  1.71517155, -0.98826098, -0.7299293 , -0.8524771 ,
       -1.10693562, -0.15371256, -0.06583069,  1.8901162 ,  0.53431469,
        3.78891958, -1.55956892,  3.98209523,  2.19238732,  2.07425095,
        1.04666605, -1.77791311,  1.16934276, -2.41750786,  0.17325692,
       -2.54949285,  1.83500079,  1.8584326 , -1.03183115, -0.04868496,
       -2.98507745, -1.14446211, -0.29210313, -0.73728669, -0.47872947,
        3.34917929, -0.3304718 , -0.55897856, -4.9794456 , -0.31600176,
        1.11453659,  0.00631809, -2.16213946, -0.13416733, -1.31088693,
        0.7420693 , -1.4762786 ,  1.98950409, -1.22014924,  0.16765788])
```

[62]:
```
y1 = 3 + 2*x + noise1
y2 = 3 + 2*x + noise2
y3 = 3 + 2*x + noise3
```

[80]:
```
xy1_df = {'X':x, 'Y':y1}
to_df1 = pd.DataFrame(xy1_df)
to_df1.corr()
mp.scatter(x,y1)
```

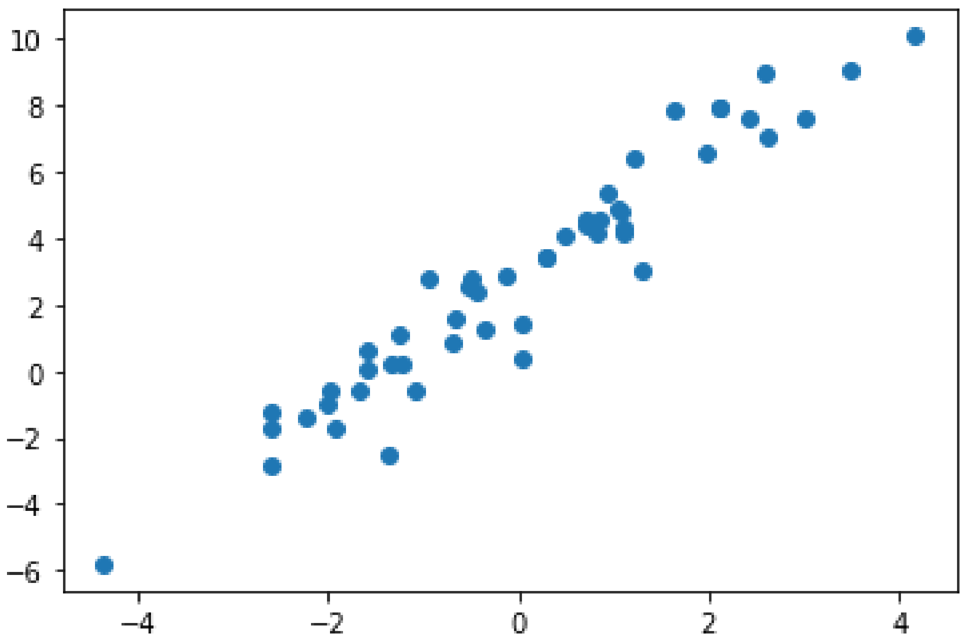
[80]: <matplotlib.collections.PathCollection at 0x2255b312d30>



[81]:
```
xy2_df = {'X':x, 'Y':y1}
to_df2 = pd.DataFrame(xy2_df)

to_df2.corr()
mp.scatter(x,y2)
```

[81]: <matplotlib.collections.PathCollection at 0x2255b4f4518>



[82]:
```
xy3_df = {'X':x, 'Y':y3}
to_df3 = pd.DataFrame(xy3_df)
to_df3.corr()
mp.scatter(x,y3)
```

[82]: <matplotlib.collections.PathCollection at 0x2255b59b198>