<Week3 - Lecture 1>:

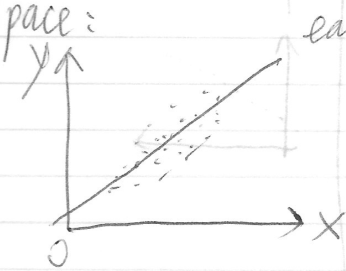
> ~~Correl~~ Covariance / Correlation :

obs	X	Y
1	X_1	Y_1
2	X_2	Y_2
⋮	⋮	⋮
n	X_n	Y_n

centralize →

	X	Y
1	$X_1-\bar{X}$	$Y_1-\bar{Y}$
2	$X_2-\bar{X}$	$Y_2-\bar{Y}$
⋮	⋮	⋮
n	$X_n-\bar{X}$	$Y_n-\bar{Y}$

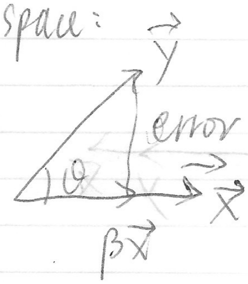
Row Space :



each point (x_i, y_i)

$$(y_i-\bar{y})=\beta(x_i-\bar{x})$$

Column Space :



$$\rho=\cos\theta=\frac{\langle\vec{x},\vec{y}\rangle}{|\vec{x}||\vec{y}|}$$

$$C_{xy}=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})$$

$$V_x=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2=C_{xx}$$

$$V_y=\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{x})^2=C_{yx}$$

$$\rho=\frac{C_{xy}}{\sqrt{V_x}\sqrt{V_y}}$$

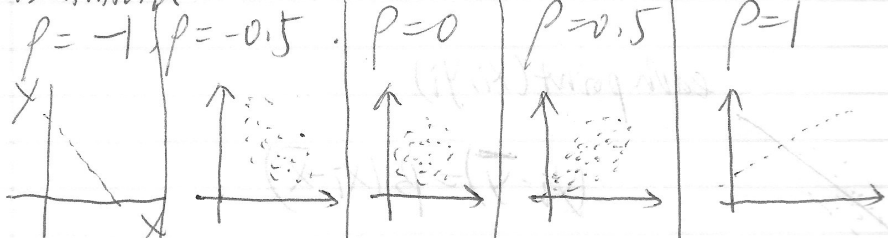
$(X,Y)\sim f(x,y)$, $Cov(X,Y)=E((X-\mu_X)(Y-\mu_Y))$

$\mu_X=E(X)$ $\mu_Y=E(Y)$

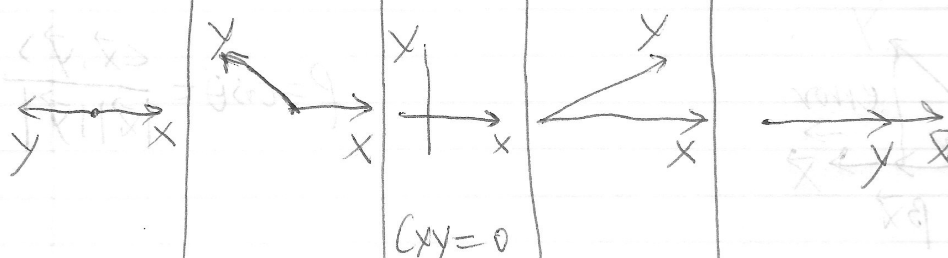
$$Var(X) = E[(X-E(X))^2] = E[(X-\mu_X)^2]$$
$$Var(Y) = E[(Y-E(Y))^2] = E[(Y-\mu_Y)^2]$$

$$\rho = \cos\theta = \frac{\langle \vec{x}, \vec{y} \rangle}{|\vec{x}||\vec{y}|} = \frac{C_{xy}}{\sqrt{V_x}\sqrt{V_y}}$$ (ρ越大，θ越小，error越小)

Row Representation:



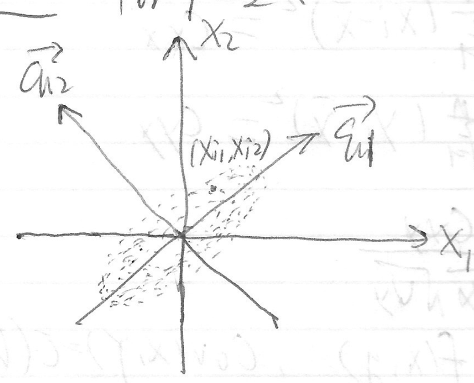
Column Representation

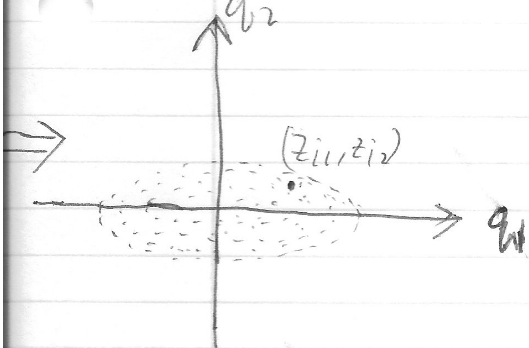


>multivariable Analysis / Principle Component Analysis:

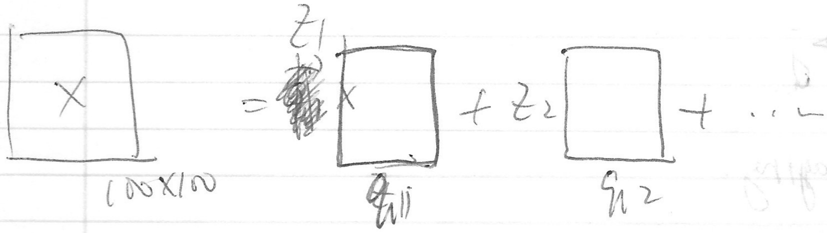
obs	$X_1 X_2 \cdots X_p$	Row Space
1	$x_{11}\ x_{12} \cdots x_{1p}$	$\vec{x}_1'$
2	$x_{21}\ x_{22} \cdots x_{2p}$	$\vec{x}_2'$
⋮	⋮	⋮
n	$x_{n1}\ x_{n2} \cdots x_{np}$	$\vec{x}_n'$
	$\bar{x}_1=0\ \bar{x}_2=0 \cdots \bar{x}_p=0$	

For $p=2$:





$(X_{i1}, X_{i2}) \to (z_{i1})$
dimensional reduction.



$$\vec{x_i} = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} \xrightarrow{Q=(q_1 q_2 \cdots q_p)} \vec{z_i} = \begin{pmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{ip} \end{pmatrix}$$

$$\vec{x_i} = z_{i1}\vec{q_1} + z_{i2}\vec{q_2} + \cdots + z_{ip}\vec{q_p} = (\vec{q_1} \cdots \vec{q_p}) \begin{pmatrix} z_{i1} \\ z_{i2} \\ \vdots \\ z_{ip} \end{pmatrix} = Q\vec{z_i}$$

packaging

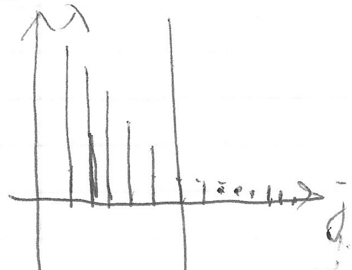
representation

Covariance: z_{ij} and z_{ik}.

	1 2 ⋯ j ⋯	k ⋯ p
1	z_{1j}	z_{1k}
⋮	z_{2j}	z_{2k}
	⋮	⋮
n	z_{nj}	z_{nk}

$$\frac{1}{n}\sum_{i=1}^{n} z_{ij} \cdot z_{ik} = 0$$

$$\boxed{\begin{gathered}\frac{1}{n}\sum_{i=1}^{n} z_{ij}^2 = \lambda_j \\ \lambda_1 \geq \lambda_2 \geq \cdots \lambda_j \geq \cdots \geq \lambda_p\end{gathered}}$$



packaging.

$$\frac{1}{n}\sum_{i=1}^{n} z_i z_i^T = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{pmatrix} = \Lambda$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T = \frac{1}{n}\sum_{i=1}^{n} Q z_i z_i^T Q^T = Q \Lambda Q^T$$

$$\Sigma Q = Q \Lambda$$

$$\Sigma (q_1 q_2 \cdots q_p) = (q_1 q_2 \cdots q_p) \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{pmatrix}$$

$$\Sigma q_j = \lambda_j q_j .$$

> Input Σ, Output Q, Λ. (Sequential way)

Power: $$\Sigma v = Q \Lambda Q^T Q u = Q \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix}$$

$$= Q \begin{pmatrix} \lambda_1 u_1 \\ \lambda_2 u_2 \\ \vdots \\ \lambda_p u_p \end{pmatrix}$$

$$U \xrightarrow{\Sigma^n} \begin{pmatrix} \lambda_1^n u_1 \\ \lambda_2^n u_2 \\ \vdots \\ \lambda_p^n u_p \end{pmatrix} \xrightarrow{n \to \infty} \begin{pmatrix} \text{Huge} \\ \text{Small} \\ \vdots \\ \text{Small} \end{pmatrix}$$

(1) $V \to \tilde{V} = \frac{V}{|V|}$

(2) $V = \Sigma \tilde{V}$

go back to (1)

$\tilde{V} \to \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ Ospace $= \vec{q_1}$

To get q_2, choose $V \perp \vec{q_1}$

power $\to \vec{q_2}$

To get q_3, choose $V \perp \vec{q_1}$, $V \perp \vec{q_2}$

power $\to \vec{q_3}$

> matrix version:

p-vectors,

$V_{p\times p} = (\vec{V_1}\,\vec{V_2} \cdots \vec{V_p}) \to \tilde{V} = (\tilde{V_1}\,\tilde{V_2} \cdots \tilde{V_p})$

orthogonal.

(1) $\tilde{V}$ = orthogonalize V

(2) $V = \Sigma \tilde{V}$

$V = QR$

$\downarrow$

$\tilde{V}$

> R code:

```
myeigen <- function(A)
{
  T=1000
  n=nrow(A)
  V= matrix(rnorm(n*n), nrow=n)
  for(i in 1:T)
    { V= myqr(V)$Q
      V = A%*%V
    }
  B =myqr(V)
  result<-list(B$Q, diag(B$R))
  names(result)<-c("vectors", "values")
  result
}

n=100
p=5
X=matrix(rnorm(n*p), nrow=n)
A = t(X)%*%X/n
myeigen(A)
eigen(A)
```

> python code:

```
Input numpy as np
from scipy import linalg
  ⋮
  ⋮ (refer to official codes)
```

<Week3 - Lecture 2>:

> Principle component Analysis (PCA):

Population

$X \sim f(x)$

$E(X) = \int x f(x) dx.$

$E[h(x)] = \int f(x) h(x) dx$

$Var(X) = E[(X-\mu)^2], \mu = E(X)$

$(X, Y) \sim f(x,y).$

$Cov(X,Y) = E(X-\mu_x)E(Y-\mu_y)$

$\mu_x \to E(X)$, $\mu_y \to E(Y).$

$Corr(X,Y) = \frac{Cov(X,Y)}{\sqrt{V_X}\sqrt{V_Y}}$

$\vec{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix} \sim f(\vec{X}) = f(x_1, x_2, \dots x_p)$

$\underset{p\times p}{Var(\vec{X})} = E[\underset{p\times 1}{(X-\mu)}\underset{1\times p}{(X-\mu)^T}] = \Sigma$

$= \left(E[(X_i-\mu_i)(X_j-\mu_j)]\right)_{p\times p} = Cov(X_i, X_j)_{p\times p}$

$= \begin{pmatrix} Var(X_i) & \\ Cov(X_i,X_j) & \end{pmatrix}$

$\underset{p\times 1}{Z} = \underset{p\times p}{A}\,\underset{p\times 1}{X}$ $\quad Var(Z) = E[[Z - E(Z)][Z - E(Z)]^T]$

$= E[(AX - A\mu_X)(AX - A\mu_X)^T]$

$= E[A(X-\mu_X)(X-\mu_X)^T A^T]$

$= A \, Var(X) A^T.$

> PCA:

$Z = Q^T X$

$X = QZ$

$Var(Z) = Q^T \Sigma Q = \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \lambda_p \end{pmatrix}$

$\Downarrow$

$\Sigma = Q \Lambda Q^T$

$\Downarrow$

$\Sigma Q = Q \lambda$

> Power method: (parallel way)

$V_{p \times p}$

(1) $\tilde{V} = normalize(V)$

(2) $V = \Sigma \tilde{V}$; go back to (1).

(1): $V = QR$

$\tilde{V} = Q$

$(v_1 v_2 \cdots v_p) = (q_1 q_2 \cdots q_p) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ & r_{22} & \cdots & r_{2p} \\ & & & \vdots \\ & & & r_{pp} \end{pmatrix}$

$v_1 = r_{11} q_1$

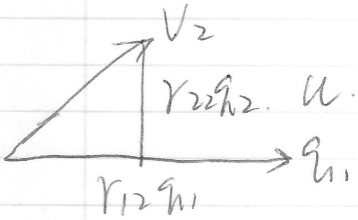
$v_2 = r_{12} q_1 + r_{22} q_2$

$v_3 = r_{13} q_1 + r_{23} q_2 + r_{33} q_3 \cdots$

... Parallel

> Gran-Schmidz Orthogonalization:

$V_1 \rightarrow q_1 = \frac{V_1}{|V_1|}$

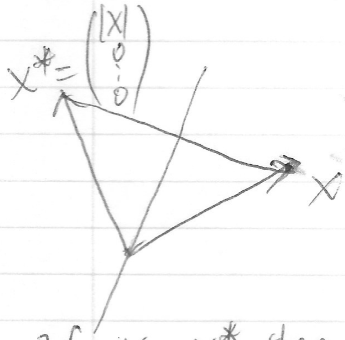


$r_{12} = \langle V_2, q_1 \rangle$

$u = V_2 - V_{12}q_1$

$q_2 = \frac{u}{|u|}$

> HouseHolder:



$\frac{x - x^*}{|x - x^*|}$

If $x = x^*$ then there is problem (divide by 0)
change sign to solve it!
(refer to the code).

> logistic regression:

obs		
1		y_1
2		y_2
$\vdots$	$x_{i1} x_{i2} \cdots x_{ip}$	y_i
$\vdots$		$\vdots$
n		y_n.

$y_i \in \{0,1\}$

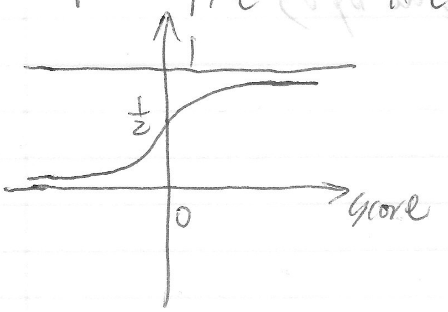
$y_i \in \{+,-\}$

$y_i \sim Bernoulli(p_i)$

$p(y_i=1) = q_i$.

$$\log \frac{p_i}{1-p_i} = x_i^T\beta$$

$$p_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} = \frac{1}{1+e^{-x_i^T\beta}}.$$



> Maximum likelihood:

$$L(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} \quad \begin{cases} y_i=1 \rightarrow p_i \\ y_i=0 \rightarrow 1-p_i \end{cases}$$

$$= \prod_{i=1}^{n} \left(\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}\right)^{y_i} \left(\frac{1}{1+e^{x_i^T\beta}}\right)^{1-y_i}$$

$$= \prod_{i=1}^{n} \frac{e^{x_i^T\beta y_i}}{1+e^{x_i^T\beta}}$$

$$l(\beta) = \log L(\beta) = \sum_{i=1}^{n} y_i x_i^T\beta - \log(1+e^{x_i^T\beta})$$

$$l'(\beta) = \sum_{i=1}^{n} y_i x_i - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} x_i$$

$$= \sum_{i=1}^{n} (y_i - p_i) x_i$$

> Gradient Descent (hill climbing):

$$\beta_0 = 0$$

$$\beta_{t+1} = \beta_t + \underbrace{\gamma}_{\text{step size}} l'(\beta_t)$$

β_t: knowledge

$$\beta_{t+1} = \beta_t + \underbrace{\gamma}_{\text{learning rate}} \sum_{i=1}^{n} (y_i - p_i) x_i$$

> stochastic gradient descent (stochastic version):

randomly pick i.

$$\beta_{t+1} = \beta_t + \gamma(y_i - p_i) x_i$$

↳ accumulated memory of x_i

$$\underset{p\times p}{l''(\beta)} = \sum_{i=1}^{n} \frac{\partial}{\partial \beta}\left(\frac{1}{1+e^{x_i^T\beta}}\right) x_i$$

$$= -\sum_{i=1}^{n} \frac{e^{x_i^T\beta}}{(1+e^{x_i^T\beta})^2} x_i x_i^T$$

$$= -\sum_{i=1}^{n} p_i(1-p_i) x_i x_i^T.$$

> Newton Raphson Algorithm:

$$\beta_{t+1} = \beta_t - l''(\beta_t)^{-1} l'(\beta_t)$$

$$= \beta_t + (\sum \underbrace{W_i}_{p_i(1-p_i)} \cancel{x_i} x_i x_i^T)^{-1} (y_i - p_i) x_i$$

$$= (\sum W_i x_i x_i^T)^{-1} \left[\sum W_i x_i x_i^T \beta_t + x_i(y_i - p_i)\right]$$

$$= \sum (W_i x_i x_i^T)^{-1} \left[\sum W_i x_i \underbrace{(\overset{\eta_i}{x_i^T\beta} + \frac{y_i - p_i}{W_i})}_{z_i}\right]$$

$$= \sum (W_i W_i x_i^T)^{-1} (\sum W_i x_i z_i)$$

$$\tilde{x}_i = \frac{x_i}{\sqrt{W_i}} \qquad \tilde{z}_i = \frac{z_i}{\sqrt{W_i}}$$

$$= (\sum \tilde{x}_i \tilde{x}_i^T)^{-1} (\sum \tilde{x}_i \tilde{z}_i)$$

Iterative Reweighted ~~Linear Regression~~ least squares :
(IRL~~R~~S)

$$\beta_t \Longrightarrow \eta_i = x_i^T\beta \longrightarrow p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}} \longrightarrow w_i = p_i(1-p_i)$$

$$z_i = \eta_i + \frac{y_i - p_i}{w_i}$$

$$\begin{cases} \tilde{x}_i = \dfrac{x_i}{\sqrt{w_i}} \\ \tilde{z}_i = \dfrac{z_i}{\sqrt{w_i}} \end{cases}$$

$$\beta_{t+1} \longleftarrow \mathrm{regr}(\tilde{z}_i, \tilde{x}_i)$$