**######特徵萃取與資料視覺化 (降維)**

PCA 的計算方式並未考慮誤差( error )，相關矩陣的對角線數值皆為 1 ，而 FA 則考慮誤差，導致相關矩陣的對角線數值會小於 1 。於是有些統計學家稱 PCA 為「 Variance Oriented 」方法，因為它可以得到最大的解釋變異，而 FA 則稱為「 Correlation Oriented 」方法，因為考慮誤差之後，它可以更清楚看出抽取出因素與問項之間的真實相關性。從數學解的角度來說， PCA 也可看成是 FA 的特例，因FA不一定要採用 PCA 作為因素萃取的計算方法。

**\*\*\*對應分析 (CA- correspondence analysis)**

install.packages("MASS")

library(MASS)

male= c(Bush=315, Perot=152, Clinton=337)  
female= c(Bush=346, Perot=126, Clinton=571)

president1<- rbind(male, female)

usa.ca1 <- corresp(president1, nf=2)

options(digits=3)

biplot(usa.ca1)

president2= data.frame(high\_school=c(439,159,279), college=c(26,48,58), university=c(146,102,178), graduate=c(63,19,111))

rownames(president2)= c("Bush","Perot","Clinton")

usa.ca2 <- corresp(president2, nf=2)

options(digits=3)

biplot(usa.ca2)

install.packages("ca")

library(ca)

brand=data.frame(low=c(2,39,4,4,15), medium=c(7,7,5,45,2), high=c(16,3,23,5,6))

rownames(brand)=c("A","B","C","D","E")

options(digits=3)

brand.ca=ca(brand)

brand.ca

names(brand.ca)

plot(brand.ca)

**\*\*\*正典相關 (canonical correlation)**

# x1:Weight, x2-Waist, x3-Pulse, y1-Chins, y2-Situps, y3-Jumps

body.data <-read.csv("BodyIndex.csv", header = T, sep = ",")

cor(body.data)

body.scale<-scale(body.data)

body.cann<- cancor(body.scale[, 1:3], body.scale[, 4:6])

names(body.cann)

body.cann$cor

body.cann$xcoef

body.cann$ycoef

cancor.test(body.scale[, 1:3], body.scale[, 4:6], plot=T)

#计算数据在典型变量下的得分 U=AX,  V=BY  
U<-as.matrix(body.scale [, 1:3]) %\*% body.cann$xcoef   
V<-as.matrix(body.scale [, 4:6]) %\*% body.cann $ycoef

#繪出U1、V1和U3、V3為组表的資料散佈圖

plot(U[,1], V[,1], xlab="U1", ylab="V1")

plot(U[,2], V[,2], xlab="U2", ylab="V2")  
plot(U[,3], V[,3], xlab="U3", ylab="V3")

**\*\*\*\*\*多維尺度(MDS- multi-dimensional scaling)**

#x1: 農業、x2: 林業、x3: 畜牧、x4: 工業、x5: 服務///20個農村城市

city.data <-read.csv("city.csv", header = T, sep = ",")

dis=dist([city.data[,-1]](http://inside-r.org/r-doc/utils/data), method='euclidean')

city.mds<- [cmdscale](http://inside-r.org/r-doc/stats/cmdscale)(dis, k=3, eig=T)

sum((city.mds$eig[1:2])^2)/sum((city.mds$eig)^2) #計算前兩個特徵值的%

cityx = city.mds$points[,1]

cityy = city.mds$points[,2]

plot(cityx, cityy, pch=0:19, col=1:20, xlab = "1st dim", ylab = "2nd dim")

text(cityx, cityy, labels = city.data$city)

#iris資料

dist1=[dist](http://inside-r.org/r-doc/stats/dist)([iris[,-5]](http://inside-r.org/r-doc/utils/data), method='euclidean')

[heatmap](http://inside-r.org/r-doc/stats/heatmap)([as.matrix](http://inside-r.org/r-doc/base/as.matrix)(dist1), labRow = F, labCol = F)

dis.cls=[hclust](http://inside-r.org/r-doc/stats/hclust)(dist1, method='ward.D')

result1=[cutree](http://inside-r.org/r-doc/stats/cutree)(dis.cls, k=3)

iris.mds<- [cmdscale](http://inside-r.org/r-doc/stats/cmdscale)(dist1, k=2, eig=T)

mdsx = iris.mds$points[,1]

mdsy = iris.mds$points[,2]

result.mds<- data.frame(cbind(mdsx, mdsy, iris[, 5]))

attach(result.mds)

plot(mdsx, mdsy, pch=V3, col=V3+1)

dist2=dist(result.mds[,-3], method='euclidean')

model2=[hclust](http://inside-r.org/r-doc/stats/hclust)(dist2, method='ward.D')

result2=[cutree](http://inside-r.org/r-doc/stats/cutree)(model2, k=3)

[library](http://inside-r.org/r-doc/base/library)([ggplot2](http://inside-r.org/packages/cran/ggplot2))

p=[ggplot](http://inside-r.org/packages/cran/ggplot)([data.frame](http://inside-r.org/r-doc/base/data.frame)(mdsx, mdsy), aes(mdsx, mdsy))

p+geom\_point(size=3, aes(colour=[factor](http://inside-r.org/r-doc/base/factor)(result1), [shape](http://inside-r.org/packages/cran/shape)=[iris](http://inside-r.org/r-doc/datasets/iris)$Species))

p+geom\_point(size=3, aes(colour=[factor](http://inside-r.org/r-doc/base/factor)(result2), [shape](http://inside-r.org/packages/cran/shape)=[iris](http://inside-r.org/r-doc/datasets/iris)$Species))

**\*\*\*\*\*奇異值分解(SVD- singular value decomposition)**

attach(iris)

iris.svd <- svd(iris[, c(1:4)])

iris.svd$d^2/sum(iris.svd$d^2)

plot(iris.svd$d^2/sum(iris.svd$d^2), pch=19, xlab="singluar vector", ylab= "variance explained")

species<-as.numeric(iris$Species)

plot(iris.svd$u[,1], iris.svd$u[,2], pch= species, col= species+1, xlab="1st left singular vector", ylab="2nd left singular vector")

regene.iris <- iris.svd$u[,1:4] %\*% diag(iris.svd$d[1:4]) %\*% t(iris.svd$v[,1:4])

head(regene.iris)

head(iris)

**\*\*\*\*\*因素分析(FA- factor analysis)**

install.packages("psych")

library(psych)

install.packages("GPArotation")

library(GPArotation)

bsc.data <-read.csv("BSC.csv", header = T)

bsc.cov<- cov(bsc.data)

bsc.cor <- cor(bsc.data)

bsc.fa<- fa(bsc.cor, nfactors=4, rotate="varimax", fm="pa", score=T)

bsc.fa

names(bsc.fa)

bsc.fa$loadings

factor.plot(bsc.fa,labels=rownames(bsc.fa$loadings))

fa.parallel(bsc.cor,fa="both",n.obs=395, main="Scree plots with parallel analysis")

factanal(bsc, factors=5) #??? Error

**\*\*\*主成分分析(PCA- principal component analysis)**

attach(iris)

iris.pca<-prcomp(~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width, center=TRUE, scale=TRUE)

summary(iris.pca)

plot(iris.pca, type="line", main="Screen Plot for iris flower")   #繪製陡坡圖

iris.pca$sdev^2       #因子的標準差平方就是特徵值

fload<-iris.pca$rotation #factor loading

biplot(iris.pca, choices=1:2)  #因素負荷量圖

install.packages("psych")

library(psych)

pca2=principal(iris[,-5], nfactors=2, score=T)

print(pca2)

result.pca<- data.frame(cbind(pca2$scores, iris[, 5]))

attach(result.pca)

plot(PC1, PC2, pch=V3, col=V3+1)

install.packages("class") #以KNN測試PCA的分類績效

library(class)

pca\_knn=knn(result.pca[,-3], result.pca[,-3], cl= result.pca[,3], k = 3)

table(result.pca[,3], pca\_knn)

**\*\*\*\*\*大聯盟資料**

mlb<-read.csv("2012MLB.csv", header=T, sep=",")  
head(mlb)

attach(mlb)

mlb.pca<-prcomp(~OBP+HR+RBI+SB+H, data=mlb, center=TRUE, scale=TRUE)

summary(mlb.pca)

load=mlb.pca$rotation  #萃取轉軸後的因素負荷量

plot(mlb.pca, type="line", main="Screen Plot for 2012MLB")   #繪製陡坡圖

mlb.pca$sdev^2       #因子的標準差平方就是特徵值

biplot(mlb.pca, choices=1:2)  #因素負荷量圖

Team    # 呼叫Team變數內容

sorted.loadings=load[order(load[,1]), 1]        #將因素負荷量排序

dotchart(sorted.loadings, main="Loading Plot for PC1", xlab="Variable Loadings", cex=1.5, col="red")   #繪製因素負荷量點圖

sorted.loadings=load[order(load[,2]), 2]

dotchart(sorted.loadings, main="Loading Plot for PC2", xlab="Variable Loadings", cex=1.5, col="blue")      
sorted.loadings=load[order(load[,3]),3]   
dotchart(sorted.loadings, main="Loading Plot for PC3", xlab="Variable Loadings", cex=1.5, col="green")

sorted.loadings=load[order(load[,4]),4]   
dotchart(sorted.loadings, main="Loading Plot for PC4", xlab="Variable Loadings", cex=1.5, col="black")

model<-lm(RBI~AVG+OBP+SB+BB+HR, data=mlb, x=T)

summary(model)

summary(step(model), k=2, method="both")

**\*\*\*\*\*\*\*\*\*\*\*以IC設計資料實作**

install.packages("clusterSim")

library(clusterSim)

IC\_data <- read.csv("IC\_design.csv") # IC\_資料集

IC\_norm<-data.Normalization( IC\_data[,1:24], type="n4", normalization="column") IC.cor=cor(IC\_norm) #查看資料間之相關係數

old<- as.matrix(IC\_data[, 1:24]) #將原始資料轉為矩陣格式

**\*\*\*\*\*\*\*\*\*\*\*\*\*主成分分析**

IC.pca <- prcomp(IC\_norm, center = TRUE, scale=TRUE) #主成份建模

summary(IC.pca)

eig <- (IC.pca$sdev)^2 #計算特徵值

variance <- eig\*100/sum(eig) #計算變異量百分比

cumvar <- cumsum(variance) #計算累積變異量

##每個主成分的變異量

pca.eig <- data.frame(eig = eig, variance = variance, cumvariance = cumvar)

plot\_cul<-barplot(pca.eig [,2], names.arg=1:nrow(pca.eig), main = "Variances", xlab = "Principal Components", ylab = "Percentage of variances", col ="steelblue")

names(IC.pca)

IC.pca$rotation #查看factor loading

plot(IC.pca, type="line", main="Screen Plot for IC-design")

\*\*\*\*\*\*\*\*\*由於factor loading不明顯，故使用最大變異法進行轉軸

install.packages("psych")

library(psych)

pc<- principal(IC.cor, nfactors=8, scores=TRUE, rotate= "varimax")

pc

names(pc)

round(pc$weights, 3) ##獲取主成分得分的係數

load <- pc$weights ##計算新的座標值

pca.weight=load[,1:6] ##只取6個主成分

data1= old %\*% pca.weight ##兩矩陣相乘

data1=data.frame(data1)

colnames(data1)=c("pc1","pc2","pc3","pc4","pc5","pc6")

**\*\*\*\*\*\*\*\*\*\*\*\*\*因素分析**

KMO(IC.cor) #KMO取樣適切性量數檢定

bartlett.test (IC\_data[,1:24]) #球形檢定

IC.fa <- fa(IC.cor, nfactors=8, rotate="varimax", fm="pa") #因素分析  
print(IC.fa, digits = 3) #輸出結果，並顯示小數後5位(預設2位)

names(IC.fa)

#因素計分

fa.score <- factor.scores(IC.cor, f= IC.fa$loadings, method="Harman")

fa.score

names(fa.score)

fa.weight<- fa.score$weights

data2<-old %\*% fa.weight #兩矩陣相乘

data2=data.frame(data2)

colnames(data2)=c("fa1","fa2","fa3","fa4","fa5","fa6","fa7","fa8")

**\*\*\*\*\*\*\*\*\*\*\*\*\*奇異值分解**

install.packages("dplyr")

library(dplyr)

##fit svd model #ICnew <-select(IC,1:24)

IC.svd <- svd(IC\_norm)

names(IC.svd)

plot(IC.svd$d, xlab = "column", ylab = "eigen value", pch=19)

plot(IC.svd$d^2/sum(IC.svd$d^2), xlab = "column",ylab = "Percentage of variance explained",pch=19)

##full reconstruction

IC.new1<- IC.svd$u %\*% diag(IC.svd$d) %\*% t(IC.svd$v)

error1<- IC\_norm- IC.new1

##show singular value

IC.svd$d

sum(IC.svd$d[c(1:4)]^2) / sum(IC.svd$d^2)

sum(IC.svd$d[c(1:3)]^2) / sum(IC.svd$d^2)

##partial reconstruction

u <- IC.svd$u[,1:3]

d <- IC.svd$d[c(1:3)]

v <- t(IC.svd$v[,1:3])

IC.new2 <- u %\*% diag(d)%\*%v

##dimension reduction

data3 <- u %\*% diag(d)

round(data3, 3)

plot(data3[,1], data3[,2], xlab = "dim 1", ylab = "dim 2", pch=19)

plot(data3[,1], data3[,3], xlab = "dim 1", ylab = "dim 3", pch=19)

plot(data3[,2], data3[,3], xlab = "dim 2", ylab = "dim 3" ,pch=19)

colnames(data3) <- c("sv1", "sv2", "sv3","sv4","sv5","sv6","sv7")

##rename column names

colnames(IC.new2) <- c("A1", "A2", "A3","A4","A5","A6","A7","A8","A9","A10", "B1","B2","B3","B4","B5", "C1","C2","C3","C4","D1","D2","D3","D4","D5")

error2 <- IC\_norm - IC.new2

heatmap(IC.new1, Colv=NA, Rowv=NA, labRow="")

heatmap(IC.new2, Colv=NA, Rowv=NA, labRow="")