An Introduction to Matrix Visualization & corrplot Package

Taiyun Wei

The 2nd Chinese R Conference

2009-12

CENTRAL SOUTH UNIVERSITY

中南大學

Content

1. About corrplot
   - Graph Gallery in corrplot Package
   - Details and Tips
   - Summary
2. Seriation
   - Why need
   - Criterion
   - Method
3. Application Examples
4. GAP
   - screenshots

Outline

1. About corrplot
   - Graph Gallery in corrplot Package
   - Details and Tips
   - Summary
2. Seriation
   - Why need
   - Criterion
   - Method
3. Application Examples
4. GAP
   - screenshots

Matrix Visualization

Matrix visualization is to convert a digital matrix to a graph.

- Presentation
  1. Glyph
  2. Color
  3. Other details
- **Model**
  1. Seriation (reordering) model
  2. Optimization algorithms
  3. Partition algorithms
- Goal
  1. Display data vividly
  2. Find the hidden pattern in data (clustering?)

Function in corrplot Package

Function:

- corrplot()
- corrplot.circle()
- corrplot.ellipse()
- corrplot.number()
- corrplot.pie()
- corrplot.shade()
- corrplot.square()
- corrplot.shade()
- corrplot.mtest()

Rforge: `http://r-forge.r-project.org/projects/corrplot/`

Blog: `http://taiyun.cos.name/wp-content/uploads/2009/10/corrplot.zip`

R Graph Gallery: `http://addictedtor.free.fr/graphiques/graphcode.php?graph=152`

PCA Order

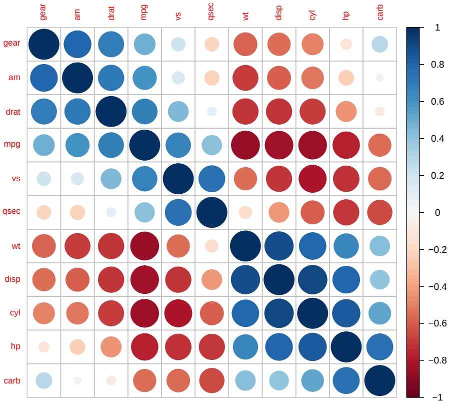


Figure: circle graph

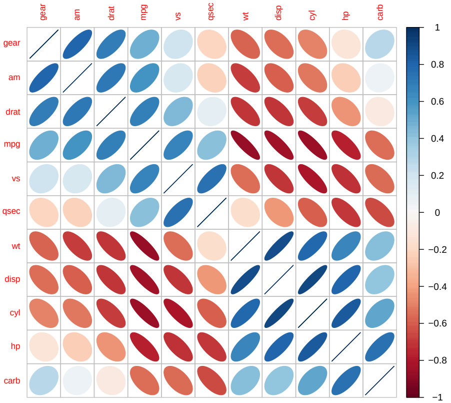


Figure: ellipse graph

HC Order (complete)

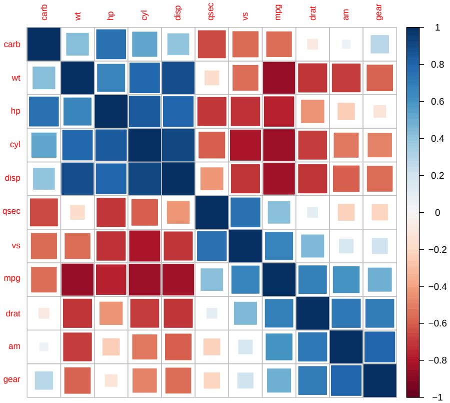


Figure: square graph

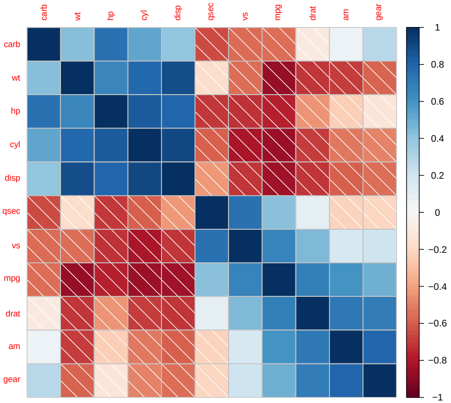


Figure: shade graph

Original Order

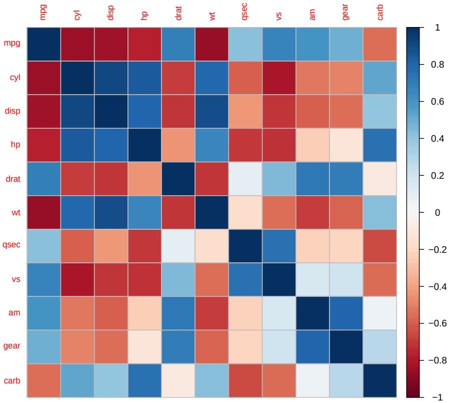


Figure: image graph

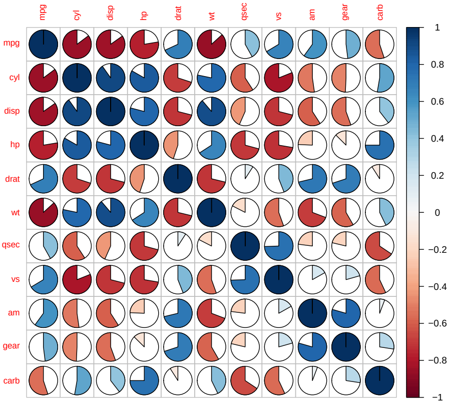
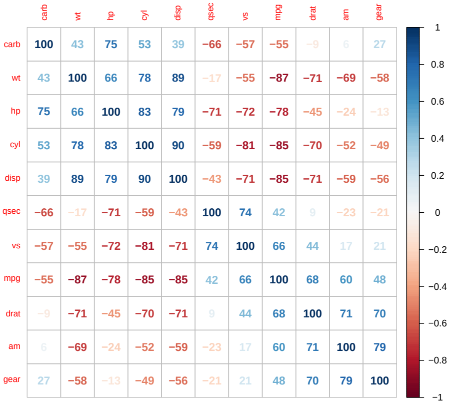


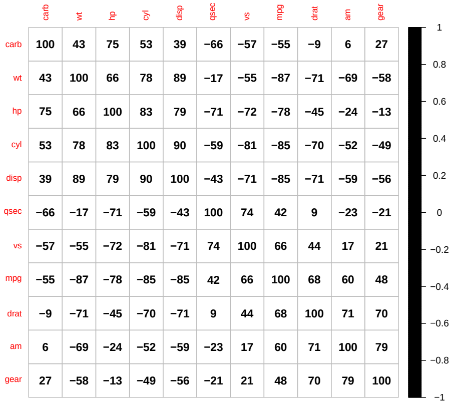
Figure: pie graph

Digital Matrix



	carb	wt	hp	cyl	disp	qsec	vs	mpg	drat	am	gear
carb	100	43	75	53	39	−66	−57	−55	−9	6	27
wt	43	100	66	78	89	−17	−55	−87	−71	−69	−58
hp	75	66	100	83	79	−71	−72	−78	−45	−24	−13
cyl	53	78	83	100	90	−59	−81	−85	−70	−52	−49
disp	39	89	79	90	100	−43	−71	−85	−71	−59	−56
qsec	−66	−17	−71	−59	−43	100	74	42	9	−23	−21
vs	−57	−55	−72	−81	−71	74	100	66	44	17	21
mpg	−55	−87	−78	−85	−85	42	66	100	68	60	48
drat	−9	−71	−45	−70	−71	9	44	68	100	71	70
am	6	−69	−24	−52	−59	−23	17	60	71	100	79
gear	27	−58	−13	−49	−56	−21	21	48	70	79	100

Figure: colored-digits graph



	carb	wt	hp	cyl	disp	qsec	vs	mpg	drat	am	gear
carb	100	43	75	53	39	−66	−57	−55	−9	6	27
wt	43	100	66	78	89	−17	−55	−87	−71	−69	−58
hp	75	66	100	83	79	−71	−72	−78	−45	−24	−13
cyl	53	78	83	100	90	−59	−81	−85	−70	−52	−49
disp	39	89	79	90	100	−43	−71	−85	−71	−59	−56
qsec	−66	−17	−71	−59	−43	100	74	42	9	−23	−21
vs	−57	−55	−72	−81	−71	74	100	66	44	17	21
mpg	−55	−87	−78	−85	−85	42	66	100	68	60	48
drat	−9	−71	−45	−70	−71	9	44	68	100	71	70
am	6	−69	−24	−52	−59	−23	17	60	71	100	79
gear	27	−58	−13	−49	−56	−21	21	48	70	79	100

Figure: black-digits graph

Print in Black and White

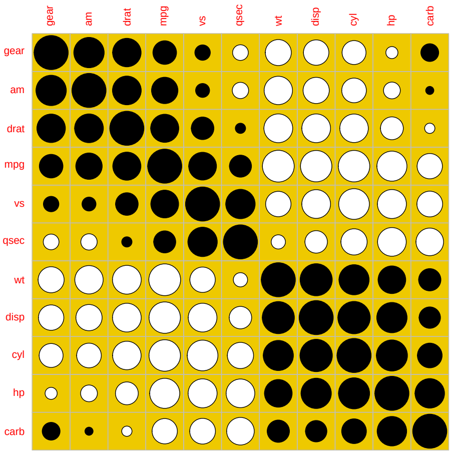


Figure: weiqi graph

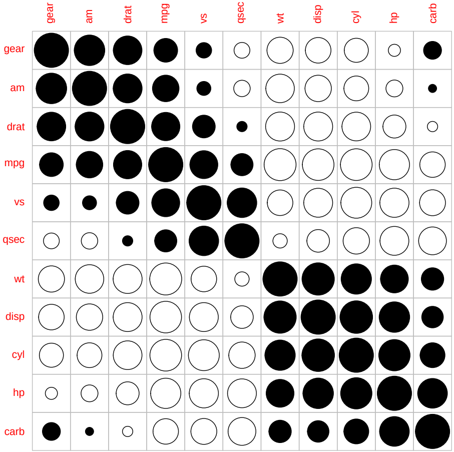


Figure: black-white graph

Test for Association/Correlation(α=0.05)

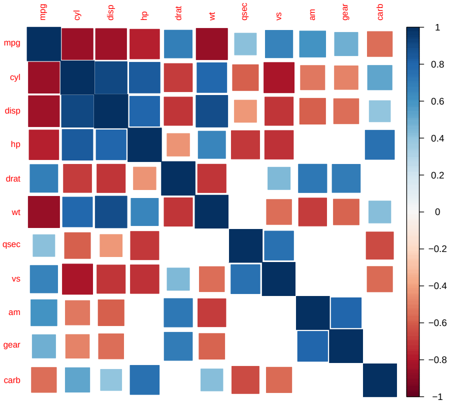


Figure: multi-correlation test (blank method)

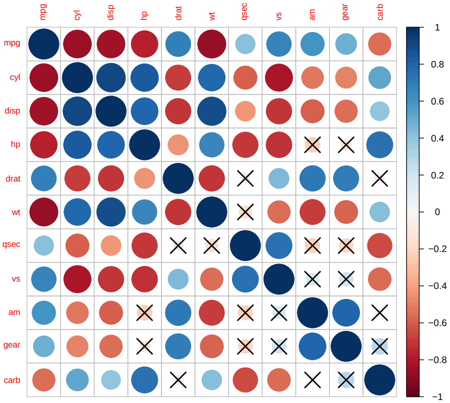


Figure: multi-correlation test (cross method)

Confidence Interval(95%)

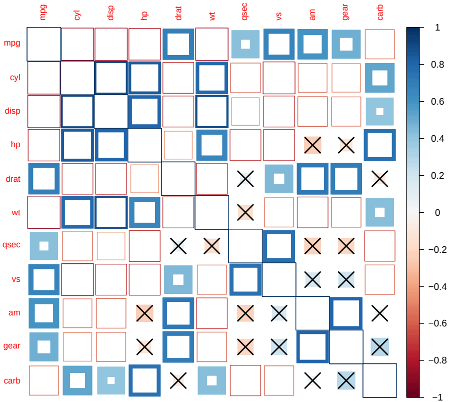


Figure: duo-square graph

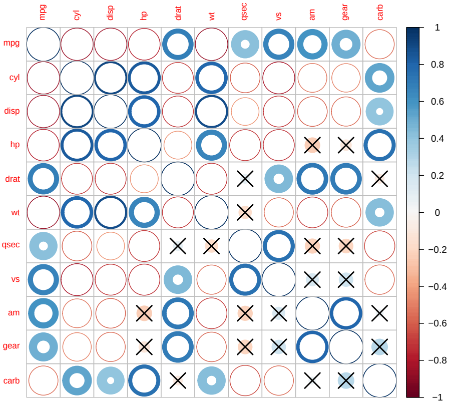


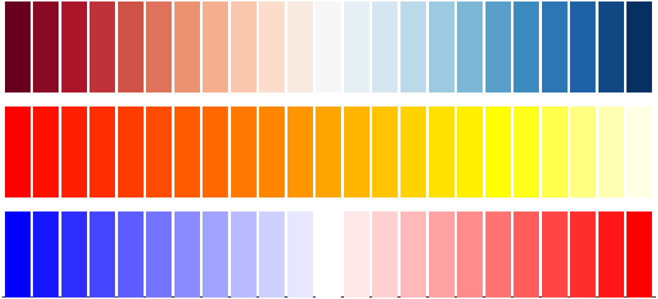
Figure: duo-circle graph

Choose Proper Color

- interpolate a set of given colors to create new color palettes

```
colorRamp(colors, bias = 1, space = c("rgb", "Lab"), ...)
colorRampPalette(colors, ...)
```

- Examples



Upper or Lower

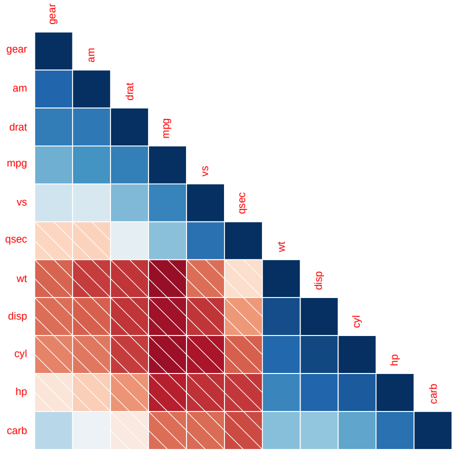


Figure: lower

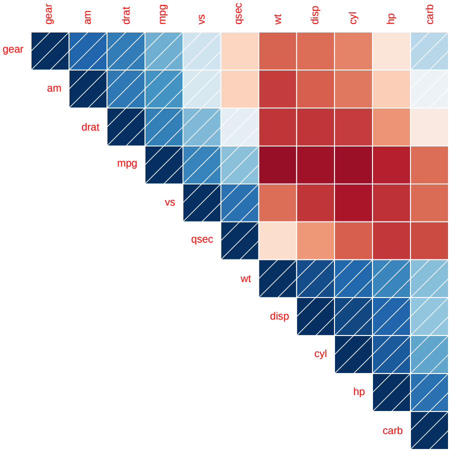


Figure: upper

Outline, colorkey, grid, text label, etc

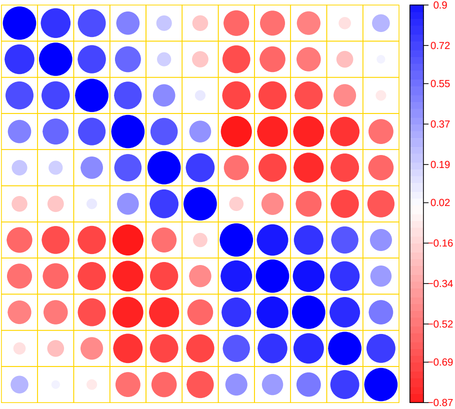


Figure: outline-0

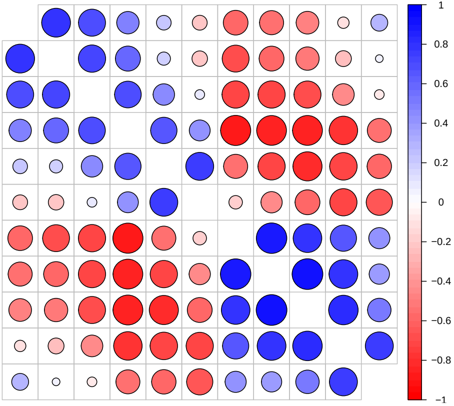


Figure: outline-1

Who cares *corrplot*?

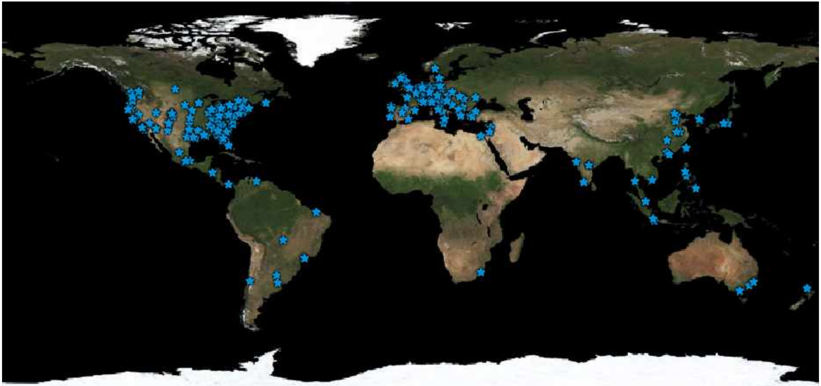


Figure: Visitor Map

Summary

- What can *corrplot* do?
  1. Basic seriation: HC, PCA, alphabet
  2. Display methods: circle, ellipse, square, etc
  3. Details: color, grid, colorkey, text-label, etc
- Advantages
  1. Creates nice and helpful pictures
  2. Flexible and good at details
  3. Easy and convenience: merely one function (about 400 lines)
- Disadvantages
  1. Lack seriation method
  2. Slow and sucks when handle large matrix
- How to get *corrplot*:
  1. From R-forge
  2. Ask me to send

Outline

1. About corrplot
   - Graph Gallery in corrplot Package
   - Details and Tips
   - Summary
2. Seriation
   - Why need
   - Criterion
   - Method
3. Application Examples
4. GAP
   - screenshots

Why need?

Get the hidden Structure and Pattern:

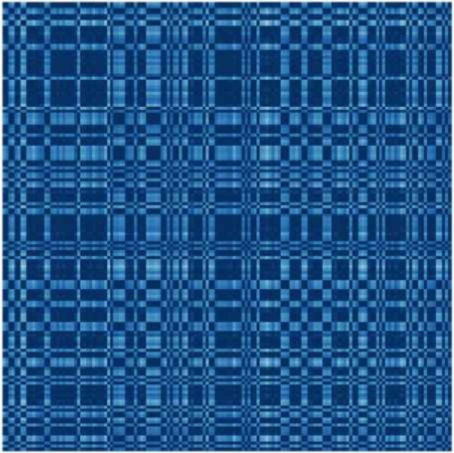


Figure: random

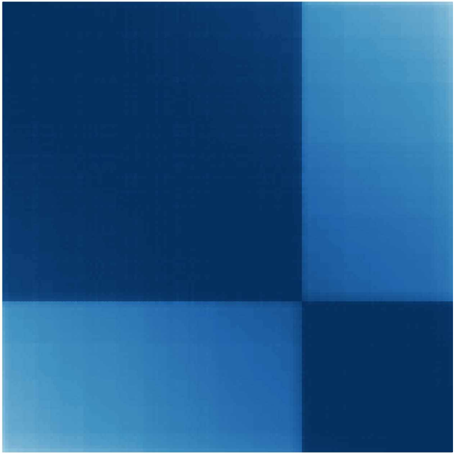


Figure: ordered

How to measure ?

Robinson Matrix and Anti-Robinson Matrix

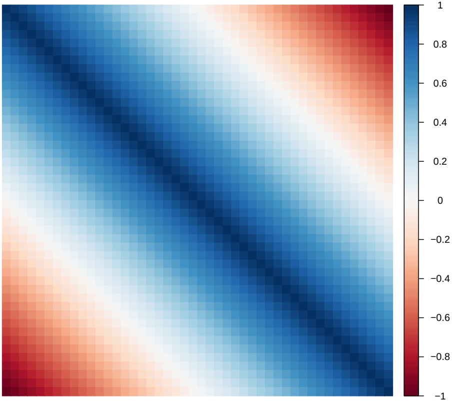


Figure: Robinson Matrix

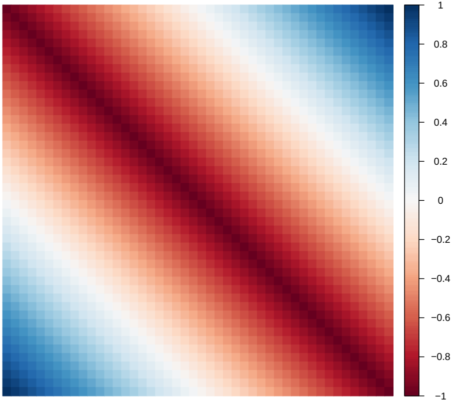


Figure: Anti Robinson Matrix

How to measure ?

Robinson Matrix and Pre-Robinson Matrix

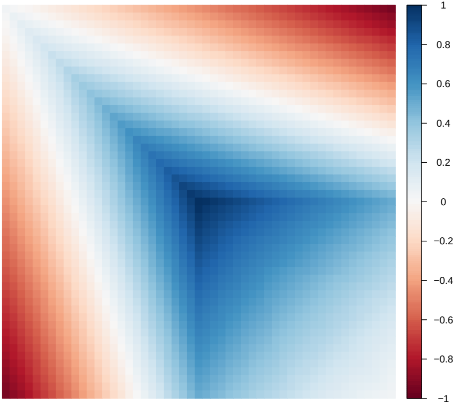


Figure: Robinson Matrix

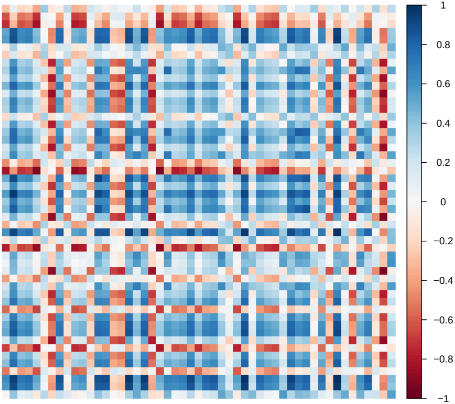


Figure: Pre Robinson Matrix

Combinatorial Optimization Model

- Anti-Robinson

$$L(\mathbf{D}) = \sum_{j<k<i} I(d_{ij} < d_{ik}) + \sum_{i<j<k} I(d_{ij} > d_{ik}) \tag{2.1}$$

- Hamiltonian path length

$$L(\mathbf{D}) = \sum_{i=1}^{n-1} d_{i,i+1} \tag{2.2}$$

- Inertia criterion

$$M(\mathbf{D}) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}|i-j|^2 \tag{2.3}$$

- Least squares criterion

$$L(\mathbf{D}) = \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij} - |i-j|)^2 \tag{2.4}$$

- Measure of effectives

$$M(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}[x_{i,j+1} + x_{i,j-1} + x_{i+1,j} + x_{i-1,j}] \quad (2.5)$$

- Stress:

$$L(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \sigma_{ij} \quad (2.6)$$

  - The Moore neighborhood:

$$\sigma_{ij} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} \sum_{l=\max(1,j-1)}^{\min(m,j+1)} (x_{ij} - x_{kl})^2 \quad (2.7)$$

  - The Neumann neighborhood :

$$\sigma_{ij} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} (x_{ij} - x_{kj})^2 + \sum_{l=\max(1,j-1)}^{\min(m,j+1)} (x_{ij} - x_{il})^2 \quad (2.8)$$

Reorder a matrix

- Five families of methods:
  1. Robinsonian： Ellipse seriation
  2. Dimension reduction： PCA, MDS
  3. Block modeling： Kmeans, Hierarchical clustering, etc
  4. Heuristics： SA, GA, PSO
  5. Graph methods： TSP
- Useful packages in R
  1. **seriation**
  2. **blockmodeling**
  3. **TSP**
  4. **Cairo**

seriation package

Table: Currently implemented methods in seriation package

Algorithm	method	Optimizes	Input data
Simulated annealing	"ARSA"	Gradient measure	dist
Branch-and-bound	"BBURCG"	Gradient measure	dist
Branch-and-bound	"BBWRCG"	Gradient measure (weighted)	dist
TSP solver	"TSP"	Hamiltonian path length	dist
Optimal leaf ordering	"OLO"	Hamiltonian path length	dist
Bond Energy Algorithm	"BEA"	Measure of effectiveness	matrix
TSP to optimize ME	"BEA_TSP"	Measure of effectiveness	matrix
Hierarchical clustering	"HC"	Other	dist
Gruvaeus and Wainer	"GW"	Other	dist
Rank-two ellipse seriation	"Chen"	Other	dist
MDS – first dimension	"MDS"	Other	dist
First principal component	"PCA"	Other	matrix

seriation package

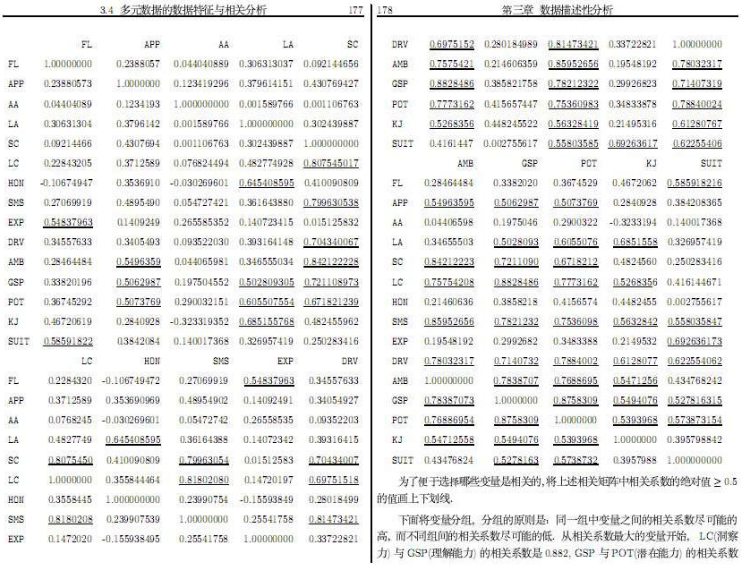
Table: Implemented loss/merit functions in function `criterion`.

Name	method	merit/loss	Input data
Anti-Robinson events	`"AR_events"`	loss	`dist`
Anti-Robinson deviations	`"AR_deviations"`	loss	`dist`
Gradient measure	`"Gradient_raw"`	merit	`dist`
Gradient measure (weighted)	`"Gradient_weighted"`	merit	`dist`
Hamiltonian path length	`"Path_length"`	loss	`dist`
Inertia criterion	`"Inertia"`	merit	`dist`
Least squares criterion	`"Least_squares"`	loss	`dist`
Measure of effectiveness	`"ME"`	merit	`matrix`
Stress (Moore neighborhood)	`"Moore_stress"`	loss	`matrix`
Stress (Neumann neighborhood)	`"Neumann_stress"`	loss	`matrix`

Outline

1. About corrplot
   - Graph Gallery in corrplot Package
   - Details and Tips
   - Summary
2. Seriation
   - Why need
   - Criterion
   - Method
3. Application Examples
4. GAP
   - screenshots

《统计建模与R软件》Section 3.4



	FL	APP	AA	LA	SC
FL	1.00000000	0.2388057	0.044040889	0.306313037	0.092144656
APP	0.23880573	1.0000000	0.123419296	0.379614151	0.430769427
AA	0.04404089	0.1234193	1.000000000	0.001589766	0.001106763
LA	0.30631304	0.3796142	0.001589766	1.000000000	0.302439887
SC	0.09214466	0.4307694	0.001106763	0.302439887	1.000000000
LC	0.22843205	0.3712589	0.076824494	0.482774928	0.807545017
HON	-0.10674947	0.3536910	-0.030269601	0.645408595	0.410090809
SMS	0.27069919	0.4895490	0.054727421	0.361643880	0.799630538
EXP	0.54837963	01409249	0.265585352	0.140723415	0.015125832
DRV	0.34557633	0.3405493	0.093522030	0.393164148	0.704340067
AMB	0.28464484	0.5496359	0.044065981	0.346555034	0.842122228
GSP	0.33820196	0.5062987	0.197504552	0.502809305	0.721108973
POT	0.36745292	0.5073769	0.290032151	0.605507554	0.671821239
KJ	0.46720619	0.2840928	-0.323319352	0.685155768	0.482455962
SUIT	0.58591822	03842084	0.140017368	0.326957419	0.250283416

	LC	HON	SMS	EXP	DRV
FL	0.2284320	-0.106749472	0.27069919	0.54837963	0.34557633
APP	0.3712589	0.353690969	0.48954902	0.14092491	0.34054927
AA	0.0768245	-0.030269601	0.05472742	0.26558535	0.09352203
LA	0.4827749	0.645408595	0.36164388	0.14072342	0.39316415
SC	0.8075450	0.410090809	0.79963054	0.01512583	0.70434007
LC	1.0000000	0.355844464	0.81802080	0.14720197	0.69751518
HON	0.3558445	1.000000000	0.23990754	-0.15593849	0.28018499
SMS	0.8180208	0.239907539	1.00000000	0.25541758	0.81473421
EXP	0.1472020	-0.155938495	0.25541758	1.00000000	0.33722821

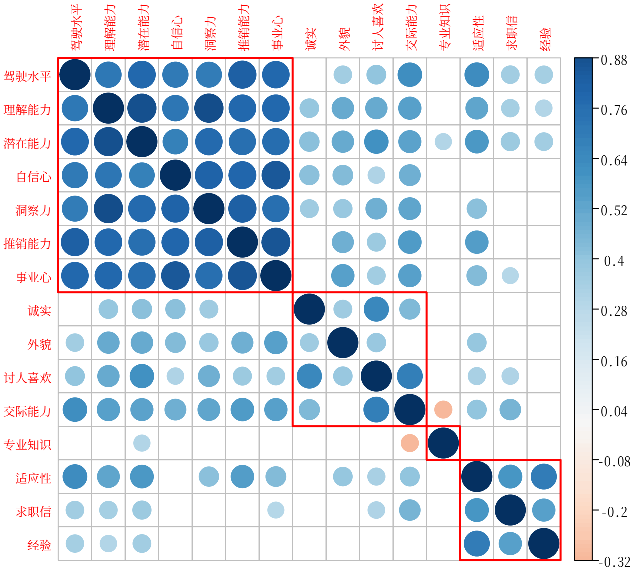
	LC	HON	SMS	EXP	DRV
DRV	0.6975152	0.280184989	0.81473421	0.33722821	1.00000000
AMB	0.7575421	0.214606359	0.85952656	0.19548192	0.78032317
GSP	0.8828486	0.385821758	0.78212322	0.29926823	0.71407319
POT	0.7773162	0.415657447	0.75360983	0.34833878	0.78840024
KJ	0.5268356	0.448245522	0.56328419	0.21495316	0.61280767
SUIT	0.4161447	0.002755617	0.55803585	0.69263617	0.62255406

	AMB	GSP	POT	KJ	SUIT
FL	0.28464484	0.3382020	0.3674529	0.4672062	0.585918216
APP	0.54963595	0.5062987	0.5073769	0.2840928	0.384208365
AA	0.04406598	0.1975046	0.2900322	-0.3233194	0.140017368
LA	0.34655503	0.5028093	0.6055076	0.6851558	0.326957419
SC	0.84212223	0.7211090	0.6718212	0.4824560	0.250283416
LC	0.75754208	0.8828486	0.7773162	0.5268356	0.416144671
HON	0.21460636	0.3858218	0.4156574	0.4482455	0.002755617
SMS	0.85952656	0.7821232	0.7536098	0.5632842	0.558035847
EXP	0.19548192	0.2992682	0.3483388	0.2149532	0.692636173
DRV	0.78032317	0.7140732	0.7884002	0.6128077	0.622554062
AMB	1.00000000	0.7838707	0.7688695	0.5471256	0.434768242
GSP	0.78387073	1.0000000	0.8758309	0.5494076	0.527816315
POT	0.76886954	0.8758309	1.0000000	0.5393968	0.573873154
KJ	0.54712558	0.5494076	0.5393968	1.0000000	0.395798842
SUIT	0.43476824	0.5278163	0.5738732	0.3957988	1.000000000

为了便于选择哪些变量是相关的,将上述相关矩阵中相关系数的绝对值 ≥ 0.5 的值画上下划线.

下面将变量分组，分组的原则是：同一组中变量之间的相关系数尽可能的高，而不同组间的相关系数尽可能的低．从相关系数最大的变量开始，LC(洞察力) 与 GSP(理解能力) 的相关系数是 0.882，GSP 与 POT(潜在能力) 的相关系数

A picture is worth a thousand words!



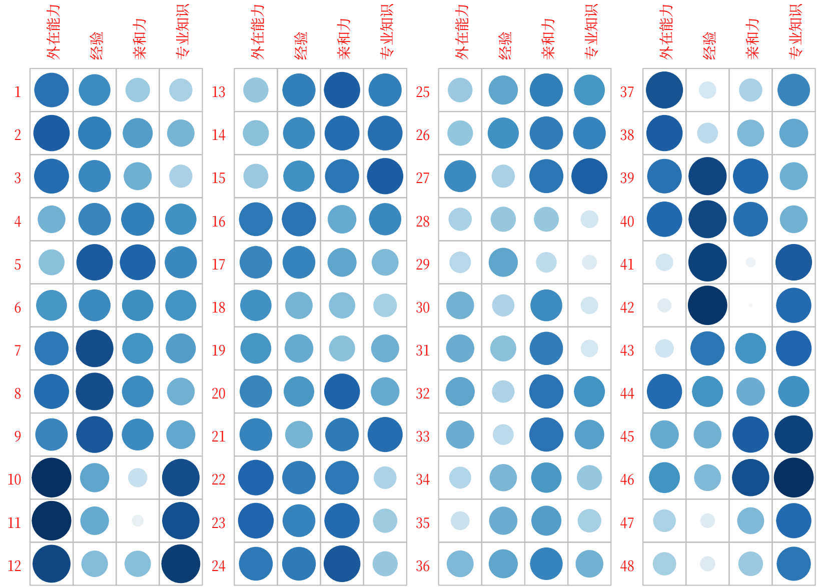
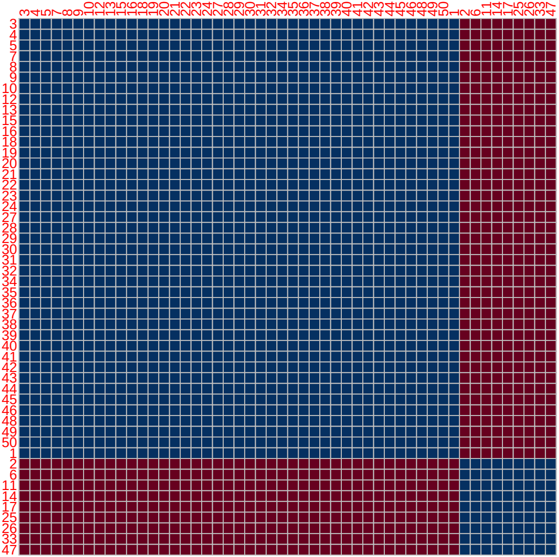
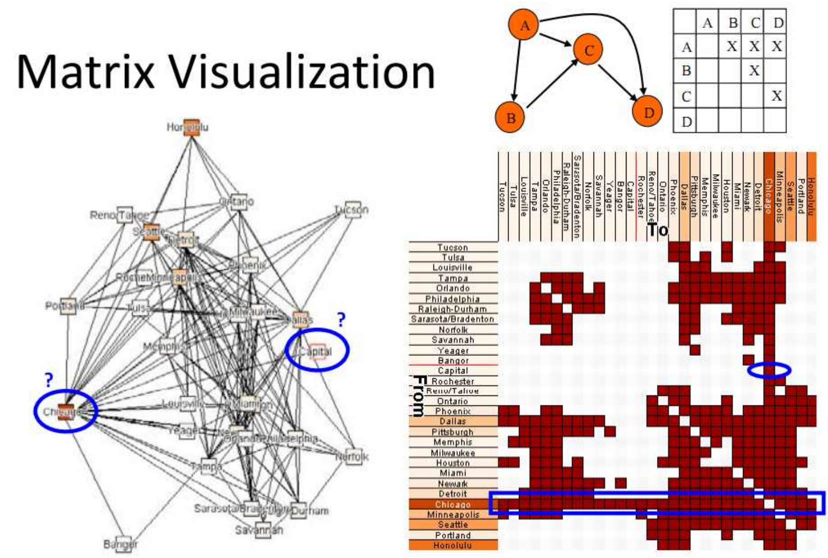


Figure: Factor Scores

Outlier Detection



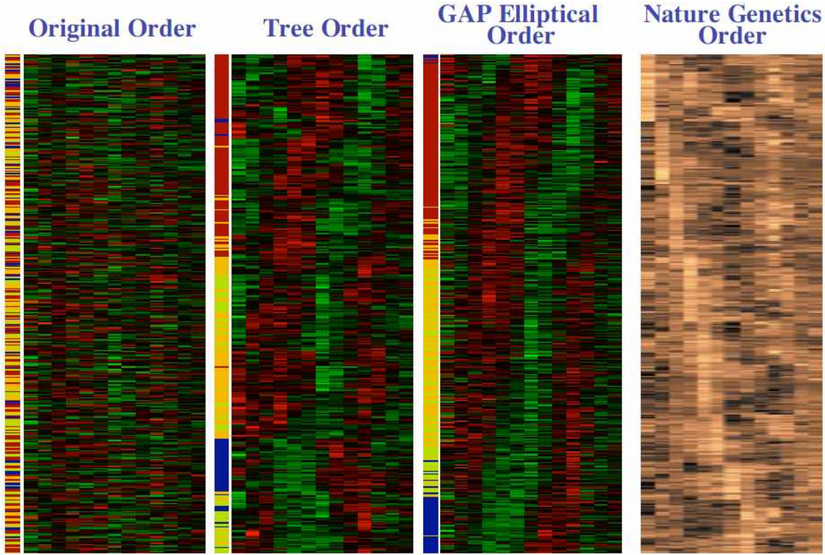
Social Networks Analysis



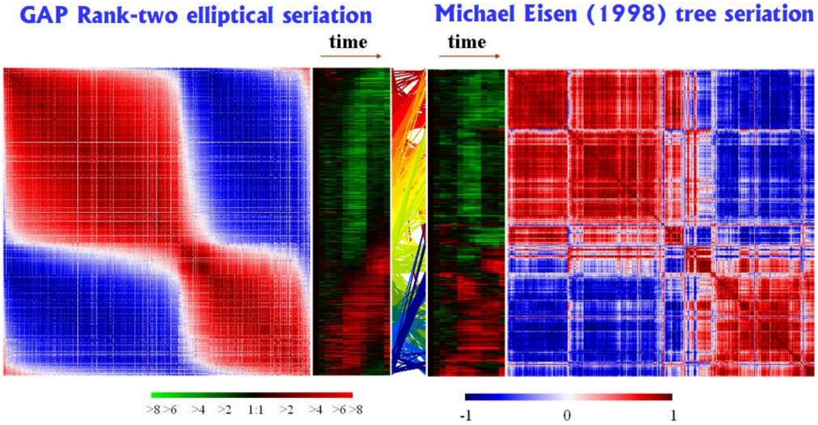
Matrix Visualization

	A	B	C	D
A		X	X	X
B			X	
C				X
D				

cDNA Microarray Analysis



cDNA Microarray Analysis



GAP Rank-two elliptical seriation

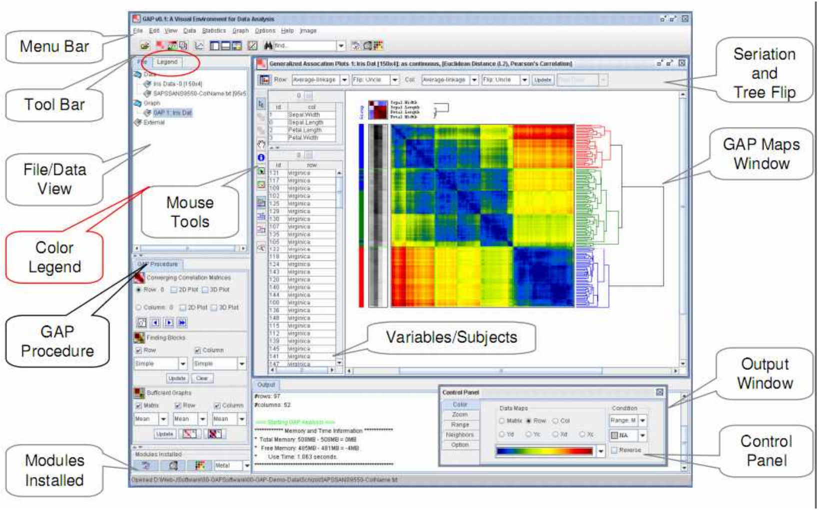
Michael Eisen (1998) tree seriation

Image source: Dr. Chen Chun-houh's slide

Outline

1. About corrplot
    - Graph Gallery in corrplot Package
    - Details and Tips
    - Summary
2. Seriation
    - Why need
    - Criterion
    - Method
3. Application Examples
4. GAP
    - screenshots

Main Window of Generalized Association Plots


GAP v0.1: A Visual Environment for Data Analysis
File Edit View Data Statistics Graph Options Help Image
File Legend
Data
Iris Data -0 [150x4]
SAPSSANS9550-ColName.txt [95x5
Graph
GAP 1: Iris Dat
External
Generalized Assocation Plots 1: Iris Dat [150x4]: as continuous, [Euclidean Distance (L2), Pearson's Correlation]
Row: Average-linkage Flip: Uncle Col: Average-linkage Flip: Uncle Update
Sepal Width
Sepal Length
Petal Length
Petal Width
Virginica
GAP Procedure
Converging Correlation Matrices
Feeding Blocks
Sufficient Graphs
Modules Installed
Output
Control Panel
Color
Zoom
Range
Neighbors
Option
Data Maps
Condition
Reverse
Opened D:\Web-J\Software\00-GAPSoftware\00-GAP-Demo-Data\Schizo\SAPSSANS9550-ColName.txt

```
#rows: 97
#columns: 52

=== Starting GAP Analysis ===
*********** Memory and Time Information ************
* Total Memory: 508MB - 508MB = 0MB
*  Free Memory: 485MB - 481MB = -4MB
*     Use Time: 1.063 seconds.
******************************************************
```

Menu Bar

Tool Bar

File/Data View

Mouse Tools

Color Legend

GAP Procedure

Modules Installed

Seriation and Tree Flip

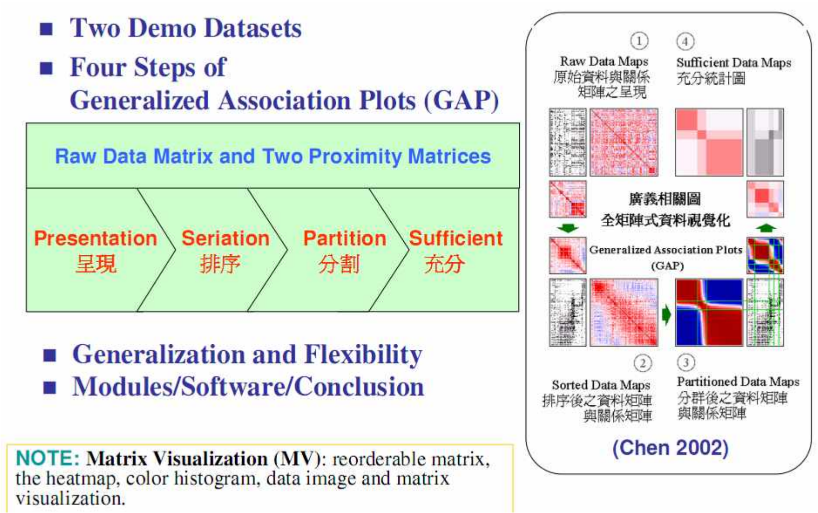
GAP Maps Window

Variables/Subjects

Output Window

Control Panel

Four Step of GAP



- Two Demo Datasets
- Four Steps of Generalized Association Plots (GAP)

Raw Data Matrix and Two Proximity Matrices

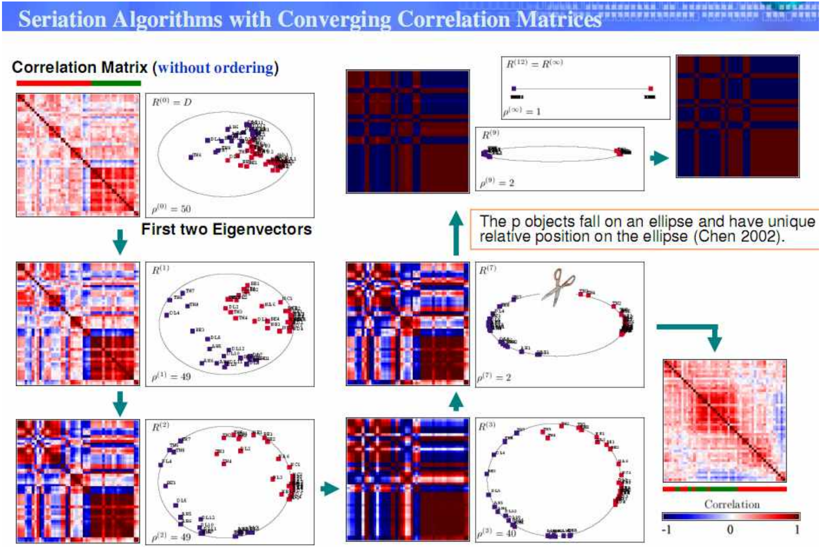
Presentation 呈現 → Seriation 排序 → Partition 分割 → Sufficient 充分

- Generalization and Flexibility
- Modules/Software/Conclusion

NOTE: **Matrix Visualization (MV)**: reorderable matrix, the heatmap, color histogram, data image and matrix visualization.

(Chen 2002)

Elliptical Seriation



Seriation Algorithms with Converging Correlation Matrices

Correlation Matrix (without ordering)

First two Eigenvectors

The p objects fall on an ellipse and have unique relative position on the ellipse (Chen 2002).

Reference:

[1] Chun-Houh Chen, **GENERALIZED ASSOCIATION PLOTS: INFORMATION VISUALIZATION VIA ITERATIVELY GENERATED CORRELATION MATRICES**, Statistica Sinica 12(2002), 7-29

[2] Han-Ming Wu, **Introduction to Generalized Association Plots for Dimension-Free Data Visualization** (slide), 2006

[3] Michael Hahsler and Christian Buchta and Kurt Hornik, **seriation: Infrastructure for seriation**, R package version 1.0-1, 2009

[4] Jean Daniel Fekete, **Visualizing Social Networks using Hybrid Matrix/NodeRepresentations**, Beijing Summer School on Visualization, 2009

[5] Han-Ming Wu and Chun-houh Chen, **GAP Software Tutorial**, 2006

[6] V. Batagelj, A. Ferligoj, P. Doreian: **Generalized blockmodeling**,2004

[7] Michael Friendly, **Corrgrams: Exploratory displays for correlation matrices**, The American Statistician, 2002

[8] 陳君厚，**全矩陣式資料視覺化與諮詢探索**，自然科學簡訊第十五卷第三期，2003

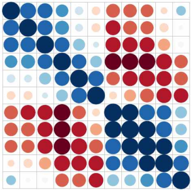
[9] 薛毅，陈丽萍. **统计建模与R软件**. 清华大学出版社, 2007.04.

Acknowledgements

- I am grateful to Yihui, linkinbird, wind, paladin1651, zwdbordeaux, miniwhale, lovelyday, Ihavenothing, Saul, pengchy, myli, soweimei, sunfeng06, 蓝枫, sbdwgu, luansheng, bjt, dingpeng, etc, for their nice comments and great suggestions in COS Home and Forum.
- I am also grateful to Shuai Huang, Roimain Francois, David Smith, Andrew Gelman, Tian Zheng, Bob, Sandip, Fangqin, Rory, Xiaoru, Michelle Zhou, Shixia, Jean Daniel, Kwanliu, Guohui, Zhanwu, Jian Huang, Hanwei, Alex Pang, etc, for their warm encouragements and relevant criticisms while we talked face-to-face and exchanged ideas via email, blog.
- Special thanks should go to Yixuan, Lanfeng, Anhua, Hao Li, Chen Zuo, Jiebiao, Ying Fang , Jian Fan, Yanping, Peng Ding, Linlin, Sizhe, Yihui, Liyun, Junwei, Tang Li, Yifeng, Chi Zhang, Xing Wang, Bo Zhang, etc, for their sweet consideration and invaluable help when I was in Beijing.
- Thank RUC, ECNU, Mango Solutions and everyone here :)

Best Wishes For You!

Thank You



Tel: 135-08489467
Email: weitaiyun@gmail.com
Blog: http://taiyun.cos.name