

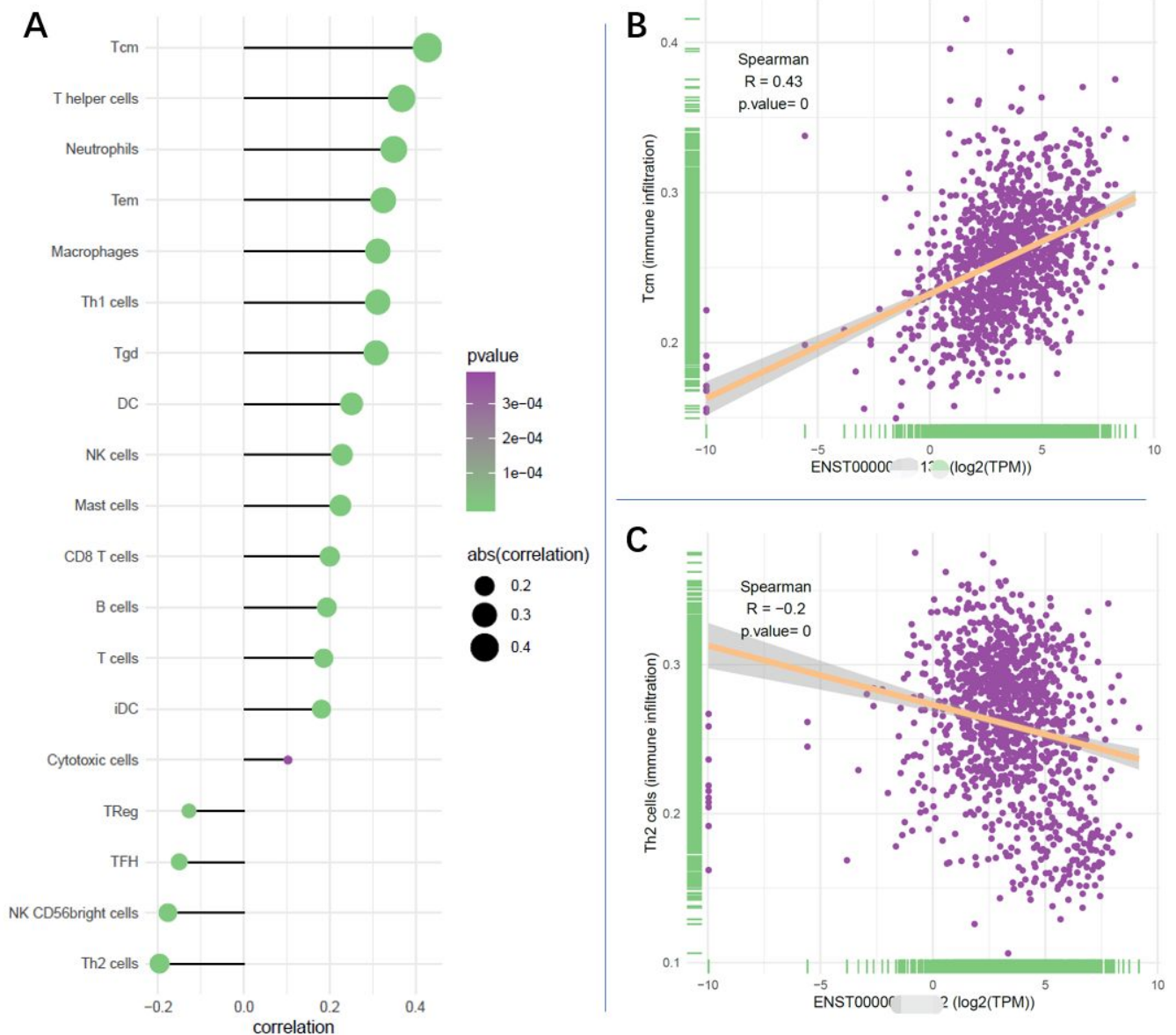
R语言：多个基因的相关性分析与展示

作者：果子 来源：果子学生信

本文原地址：<https://www.iikx.com/news/statistics/6104.html>

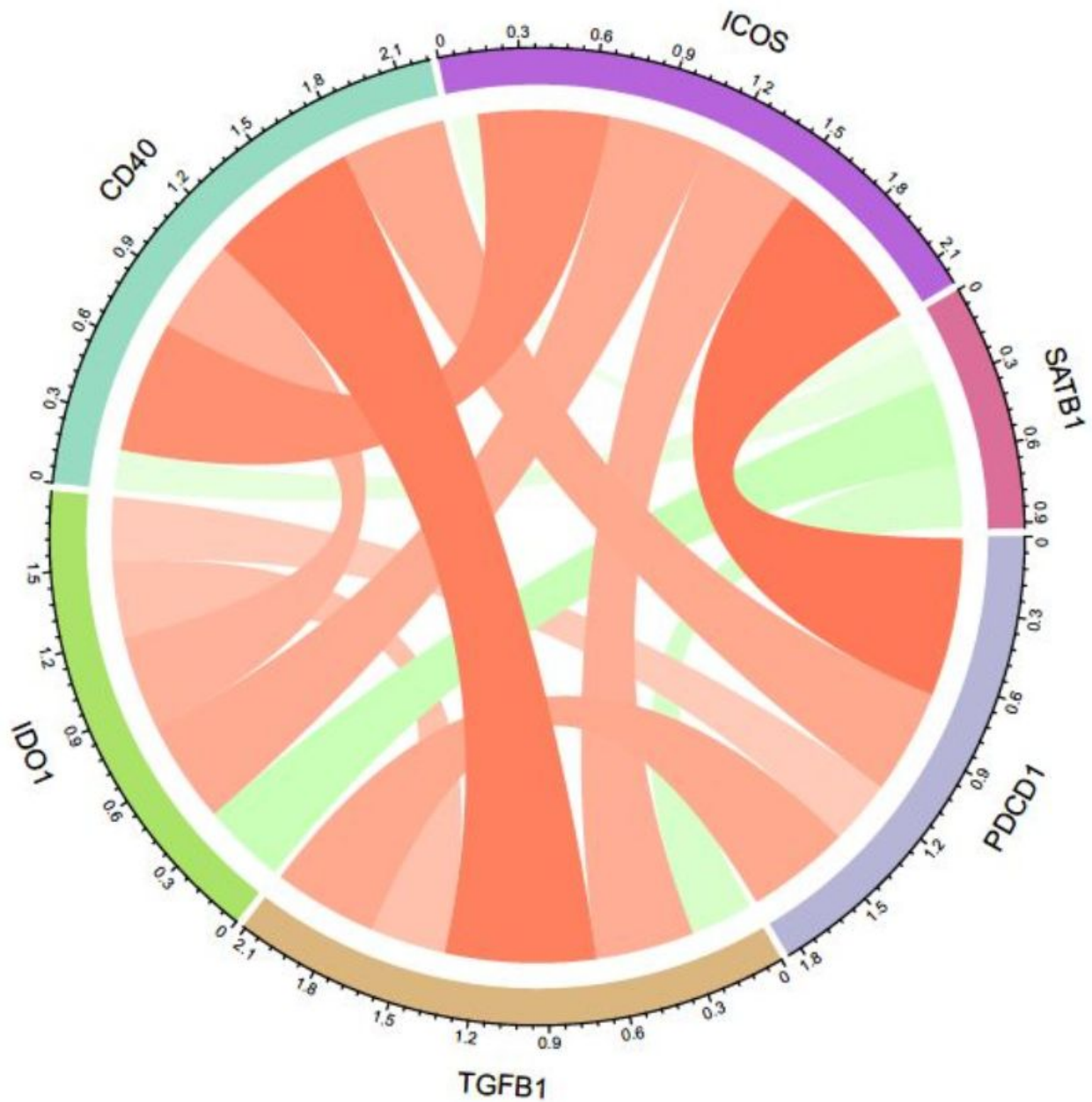
本文仅供学习交流之用，版权归原作者所有，请勿用于商业用途！

R语言：多个基因的相关性分析与展示。关于批量相关性分析，我们发过两个帖子。单基因批量相关性分析的妙用，又是神器！基于单基因批量相关性分析的GSEA。两两分析的肯定也是没有问题：



现在的问题是，如果是多个基因分相关性分析，如何快速，方便地分析，然后高效地呈现呢？

我用圈图实现过这个操作：



不过比较麻烦，今天介绍一个R包 `corr`，可以方便地做这个事情，而且我认为做的更好。

他有一个主函数correlate可以迅速地分析，实现从A到B的转换。

更重要的是他还配了两个可视化的函数，一个是rplot画热图，一个是network_plot画网络图。

出了主函数外，总共有7个函数

最后一个框里有两个函数已经介绍，还有一个fashion可以简洁化展示数据，去掉NA。

第一个框内的shave函数是把剃刀，可以去掉相关性结果上三角或者下三角并设置为NA
rearrange函数可以按照相关性系数聚类排序。

第二个框内的focus函数，类似于select函数，可以用来筛选想要查看的某行某列数据。

stretch函数可以实现数据从宽边长，如图所示。

下面我们就来实战一下：

首先设置镜像以及按照R包

```
options("repos"=c(CRAN="https://mirrors.tuna.tsinghua.edu.cn/CRAN/"))  
install.packages("corr")
```

运行主函数看看结果

```
library(corr)  
library(dplyr)  
x <- datasets::mtcars %>%  
  correlate()
```

rowname	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	
1	mpg	NA	-0.8521620	-0.8475514	-0.7761684	0.68117191	-0.8676594	0.41868403	0.6640389	0.59983243	0.4802848	-0.55092507
2	cyl	-0.8521620	NA	0.9020329	0.8324475	-0.69993811	0.7824958	-0.59124207	-0.8108118	-0.52260705	-0.4926866	0.52698829
3	disp	-0.8475514	0.9020329	NA	0.7909486	-0.71021393	0.8879799	-0.43369788	-0.7104159	-0.59122704	-0.5555692	0.39497686
4	hp	-0.7761684	0.8324475	0.7909486	NA	-0.44875912	0.6587479	-0.70822339	-0.7230967	-0.24320426	-0.1257043	0.74981247
5	drat	0.6811719	-0.6999381	-0.7102139	-0.4487591	NA	-0.7124406	0.09120476	0.4402785	0.71271113	0.6996101	-0.09078980
6	wt	-0.8676594	0.7824958	0.8879799	0.6587479	-0.71244065	NA	-0.17471588	-0.5549157	-0.69249526	-0.5832870	0.42760594
7	qsec	0.4186840	-0.5912421	-0.4336979	-0.7082234	0.09120476	-0.1747159	NA	0.7445354	-0.22986086	-0.2126822	-0.65624923
8	vs	0.6640389	-0.8108118	-0.7104159	-0.7230967	0.44027846	-0.5549157	0.74453544	NA	0.16834512	0.2060233	-0.56960714
9	am	0.5998324	-0.5226070	-0.5912270	-0.2432043	0.71271113	-0.6924953	-0.22986086	0.1683451	NA	0.7940588	0.05753435
10	gear	0.4802848	-0.4926866	-0.5555692	-0.1257043	0.69961013	-0.5832870	-0.21268223	0.2060233	0.79405876	NA	0.27407284
11	carb	-0.5509251	0.5269883	0.3949769	0.7498125	-0.09078980	0.4276059	-0.65624923	-0.5696071	0.05753435	0.2740728	NA

接着往下处理，focus选择要观察的数据，rearrange按照相关性系数排序，shave设置上三角的数据为NA

```
x <- datasets::mtcars %>%  
  correlate() %>%  
  focus(-cyl, -vs, mirror = TRUE) %>%  
  rearrange() %>%  
  shave()
```

	rowname	mpg	drat	am	gear	qsec	carb	hp	wt	disp
1	mpg	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	drat	0.6811719	NA	NA	NA	NA	NA	NA	NA	NA
3	am	0.5998324	0.71271113	NA	NA	NA	NA	NA	NA	NA
4	gear	0.4802848	0.69961013	0.79405876	NA	NA	NA	NA	NA	NA
5	qsec	0.4186840	0.09120476	-0.22986086	-0.2126822	NA	NA	NA	NA	NA
6	carb	-0.5509251	-0.09078980	0.05753435	0.2740728	-0.6562492	NA	NA	NA	NA
7	hp	-0.7761684	-0.44875912	-0.24320426	-0.1257043	-0.7082234	0.7498125	NA	NA	NA
8	wt	-0.8676594	-0.71244065	-0.69249526	-0.5832870	-0.1747159	0.4276059	0.6587479	NA	NA
9	disp	-0.8475514	-0.71021393	-0.59122704	-0.5555692	-0.4336979	0.3949769	0.7909486	0.8879799	NA

可以用fashion简化数据

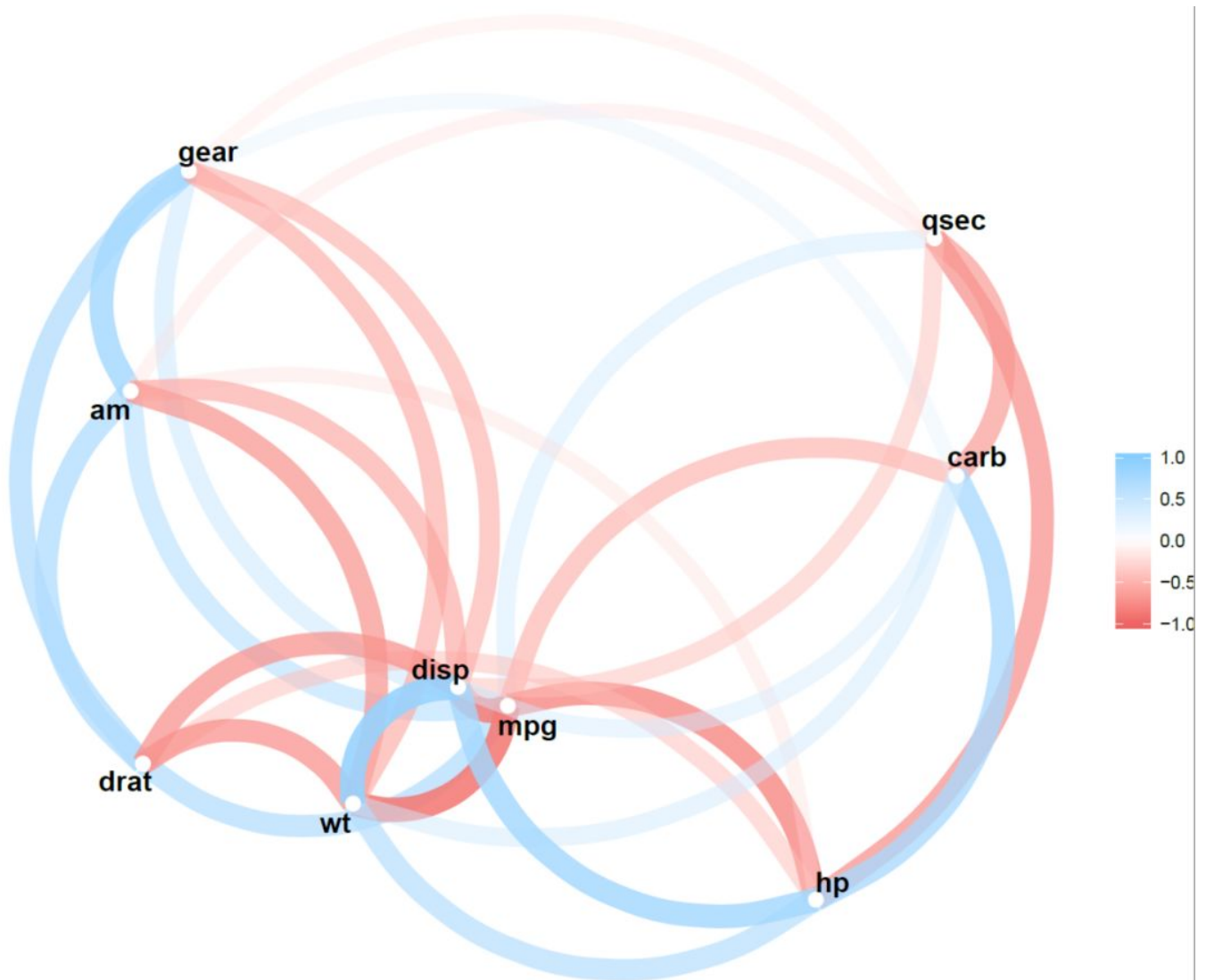
	rowname	mpg	drat	am	gear	qsec	carb	hp	wt	disp
1	mpg									
2	drat	.68								
3	am	.60	.71							
4	gear	.48	.70	.79						
5	qsec	.42	.09	-.23	-.21					
6	carb	-.55	-.09	.06	.27	-.66				
7	hp	-.78	-.45	-.24	-.13	-.71	.75			
8	wt	-.87	-.71	-.69	-.58	-.17	.43	.66		
9	disp	-.85	-.71	-.59	-.56	-.43	.39	.79	.89	

fashion(x)

可以用rplot来画图展示数据

也可以用网络图来展示

```
datasets::mtcars %>%  
  correlate() %>%  
  focus(-cyl, -vs, mirror = TRUE) %>%  
  rearrange() %>%  
  network_plot(min_cor = .2)
```



stretch可以实现数据变换

```
x <- datasets::mtcars %>%  
  correlate() %>% # Create correlation data frame (cor_df)  
  focus(-cyl, -vs, mirror = TRUE) %>% # Focus on cor_df without 'cyl'  
  and 'vs'  
  rearrange() %>%  
  stretch()
```

	x	y	r
1	mpg	mpg	NA
2	mpg	drat	0.68117191
3	mpg	am	0.59983243
4	mpg	gear	0.48028476
5	mpg	qsec	0.41868403
6	mpg	carb	-0.55092507
7	mpg	hp	-0.77616837
8	mpg	wt	-0.86765938
9	mpg	disp	-0.84755138
10	drat	mpg	0.68117191
11	drat	drat	NA
12	drat	am	0.71271113
13	drat	gear	0.69961013

学习一个R包是第一步，第二步就应该想着如何用来展示自己的数据。先来看看我们拥有的数据

```
load(file = "TCGA_steal_data.Rdata")
```

	subgroup	sample	ESR1	GATA3	ERBB2	FOXA1	NAT1	BRCA1	BRCA2	BAG1	BIRC5	MKI67
TCGA-AC-A30D-01B-06R-A220-07	LumA	Tumor	17.114199	14.643833	13.51263	14.193508	13.156971	11.001648	10.675147	12.011507	9.323800	14.036013
TCGA-AR-A251-01A-12R-A169-07	Basal	Tumor	10.269148	11.465225	12.88438	8.055327	7.814799	11.617031	10.194470	10.968246	12.661417	14.382202
TCGA-BH-A0B5-11A-23R-A12P-07	Normal	Normal	11.145986	10.013736	10.72469	6.556766	8.707859	9.177904	7.104063	12.332490	6.602116	7.676837
TCGA-E2-A1IK-01A-11R-A144-07	LumA	Tumor	16.578416	15.887390	12.31860	15.445781	10.071785	9.625762	8.348021	12.448767	10.424083	10.896239
TCGA-E2-A15C-01A-31R-A12D-07	LumA	Tumor	15.862655	15.311567	15.00679	14.561364	13.758525	9.898625	9.482084	12.523272	9.049277	11.058604
TCGA-B6-A2IU-01A-32R-A18M-07	LumA	Tumor	14.918471	16.558116	15.35859	14.572697	14.939840	10.477670	9.173315	12.878968	10.969791	11.890226
TCGA-AR-A1A5-01A-11R-A12P-07	LumB	Tumor	15.282775	15.303211	14.34886	14.609227	11.613343	10.896709	9.577009	11.894230	10.791917	12.003516
TCGA-BH-A0BD-01A-11R-A034-07	Basal	Tumor	13.886896	15.479474	13.75643	13.537705	11.542180	11.008007	10.501126	12.094719	11.519836	13.704851
TCGA-BH-A209-11A-42R-A157-07	Normal	Normal	12.821612	13.364987	13.13093	12.205101	9.186887	9.498667	8.968440	11.647156	8.105495	9.721665
TCGA-AQ-A04J-01A-02R-A034-07	Basal	Tumor	9.487192	11.350316	13.24661	9.569889	8.718935	8.829380	10.809382	11.608352	12.146518	13.822314
TCGA-A7-A13D-01A-13R-A12P-07	Basal	Tumor	8.332443	11.923866	15.02004	9.375892	8.357641	10.208300	9.866419	11.001496	12.563890	14.004373
TCGA-E2-A158-11A-22R-A12D-07	Normal	Normal	12.033338	11.024592	12.80200	9.960504	8.512144	8.933581	8.260284	12.105891	8.132622	9.364193
TCGA-E9-A22D-01A-11R-A157-07	LumB	Tumor	14.027911	15.607285	16.74466	14.671518	11.211293	10.569418	9.624489	12.572274	12.051591	12.958580
TCGA-C8-A138-01A-11R-A115-07	Her2	Tumor	11.634667	14.919210	15.13878	14.275422	12.174608	9.303917	10.302666	12.471778	10.658132	11.591685
TCGA-AO-A12H-01A-11R-A115-07	LumA	Tumor	17.179502	16.468731	13.72460	16.195024	11.490957	9.100716	8.842238	14.412712	10.195152	11.046726
TCGA-BH-A0E0-01A-11R-A056-07	Basal	Tumor	10.441638	14.232268	13.72584	9.705740	7.768527	10.256238	8.668756	12.270381	11.420427	14.191153

这个数据我们已经很熟悉了，前面两列可以不要

```
data <- TCGA_steal_data[,-c(1:2)]
```

这样我们就跟这个R包对接上了。

直接画图试试

```
data %>%  
  correlate() %>%  
  rearrange() %>%  
  network_plot()
```

更多 统计方法 请访问 <https://www.iikx.com/news/statistics/>

本文版权归原作者所有，请勿用于商业用途，[爱科学iikx.com](http://www.iikx.com)转发