第三层议程设置

Chris Chao Su, Lei Guo, Hsuan-Ting Chen

1/25/2021

## 第三层议程设置的数据建立和假设检验。

此处的实例操作以Chen, Guo & Su (2020)发表于 Journal of Communication的文章为例，来说明如何使用媒体和个人问卷数据，利用R来建立议程之间的关系矩阵，然后使用二次指派回归程序（MRQAP）来分析矩阵之间的关系，从而回答第三层议程设置的研究假设。

此处为了展示如何构建网络目的，我们随机生成了五组样本数据，包括三种政治倾向的媒体数据和两种政治倾向的公众问卷数据，五组样本数据都可以在Github上下载： \* “泛民派倾向”新闻媒体数据 (N = 350): Media\_democratic.csv \* “中立”新闻媒体数据 (N = 350): Media\_neutral.csv \* “建制派倾向”新闻媒体数据 (N = 350): Media\_proestablishment.csv \* “泛民派支持者”公众问卷数据 (N = 500): Survey\_democratic.csv \* “建制派支持者”公众问卷数据 (N = 500): Survey\_proestablishment.csv

完整的代码及样本数据详情请参见： <https://github.com/chrischaosu/JOC_2020_Chen-Guo-Su>

在此过程中，我们使用了多个R包，用于数据处理、建立矩阵、计算回归程序的统计显著性等。

### 步骤一：获得媒体数据和公众想法数据

接下来我们要看看研究中主要使用的两种不同类型的数据。

第一种数据是媒体数据，涵盖了香港的主要新闻媒体机构，根据媒体的政治倾向不一，可以大致划分为三类：建制派新闻媒体、泛民派新闻媒体、以及中立的新闻媒体。在获取了每一条新闻数据之后，我们接着对每一条新闻内容进行编码，如果该条新闻所报道的内容有涉及到某一个具体的属性，那么我们就以数字“1”记录，没有涉及到的属性则不做记录。

对每一条新闻数据，我们都有十六个变量： \* issue1 ~ issue16：十六个属性的编码结果（1 = “涉及该属性”， 0 = “不涉及该属性”）。

# 读取媒体数据  
Media\_democ <- read\_csv("Media\_democratic.csv") # 泛民派新闻媒体  
Media\_neutr <- read\_csv("Media\_neutral.csv") # 中立新闻媒体  
Media\_proes <- read\_csv("Media\_proestablishment.csv") # 建制派新闻媒体  
  
head(Media\_democ) #展示部分数据

## # A tibble: 6 x 16  
## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10 issue11  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 0 1 0 1 1 0 1 1 0 1  
## 2 0 0 1 0 0 0 0 0 0 1 0  
## 3 1 0 1 1 1 1 0 1 1 1 0  
## 4 0 1 0 0 1 1 0 0 0 0 1  
## 5 1 1 1 0 1 0 1 1 0 1 1  
## 6 0 1 1 0 1 1 1 1 0 1 0  
## # … with 5 more variables: issue12 <dbl>, issue13 <dbl>, issue14 <dbl>,  
## # issue15 <dbl>, issue16 <dbl>

第二类数据是代表公众想法的关联网络矩阵（认知属性网络）。一般而言，以问卷调查方式进行数据收集。 这一数据包含有一下信息： \* respid： 每一位有效受访者的ID； \* issue1 ~ issue5: 受访者关注的五个属性，每一个属性在数据中由1～16的代号标记（共16个属性）； \* association1 ~ association10: 受访者认为每对属性之间的关系强度（“0” = 没有关系，“10” = 极其相关）。

同样地，我们可以将数据读入到R中：

# 读取公众问卷数据  
Democratic <- read\_csv("Survey\_democratic.csv") # 泛民派支持者  
Establish <- read\_csv("Survey\_proestablishment.csv") # 建制派支持者  
  
head(Democratic) #展示部分数据

## # A tibble: 6 x 16  
## respid issue1 issue2 issue3 issue4 issue5 association1 association2  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 411 12 8 11 7 16 2 6  
## 2 53 4 14 16 15 15 5 0  
## 3 489 7 12 1 5 4 7 4  
## 4 207 1 9 12 3 16 2 2  
## 5 178 12 14 2 7 15 9 9  
## 6 422 12 8 5 1 6 7 3  
## # … with 8 more variables: association3 <dbl>, association4 <dbl>,  
## # association5 <dbl>, association6 <dbl>, association7 <dbl>,  
## # association8 <dbl>, association9 <dbl>, association10 <dbl>

### 步骤二：媒体关联网络矩阵

下来我们就可以针对每一种政治类型的媒体数据来运算其属性网络矩阵。

首先是泛民派新闻媒体，选取其16个属性的编码内容，转换成矩阵格式，计算倒置矩阵，再通过倒置矩阵与原矩阵相乘的方式得到16\*16的属性网络矩阵。

Media\_democ <- Media\_democ %>% select(issue1:issue16) #选取属性的编码内容  
Media\_democ <- as.matrix(Media\_democ) # 转换为矩阵格式  
Media\_democ\_T <- t(Media\_democ) # 计算倒置矩阵  
Media\_democ\_matrix <- Media\_democ\_T %\*% Media\_democ # 计算属性网络矩阵

然后为了方便之后的MRQAP程序，我们对矩阵的数据类型进行修正，并且将所以对角线数据（即属性与自己之间的关系）全部设置为0。

Media\_democ\_matrix <- Media\_democ\_matrix %>% as.matrix  
diag(Media\_democ\_matrix) <- 0

最后，可以看到属性网络矩阵的最后数据形式和内容：

Media\_democ\_matrix # 展示数据

## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10  
## issue1 0 100 106 95 96 86 99 96 93 89  
## issue2 100 0 94 93 96 91 95 85 99 89  
## issue3 106 94 0 86 105 89 94 83 88 92  
## issue4 95 93 86 0 86 94 85 81 91 86  
## issue5 96 96 105 86 0 81 90 83 85 84  
## issue6 86 91 89 94 81 0 87 82 88 84  
## issue7 99 95 94 85 90 87 0 85 97 92  
## issue8 96 85 83 81 83 82 85 0 76 77  
## issue9 93 99 88 91 85 88 97 76 0 89  
## issue10 89 89 92 86 84 84 92 77 89 0  
## issue11 102 109 99 91 93 85 105 95 103 83  
## issue12 101 99 95 91 85 74 85 74 86 78  
## issue13 93 86 81 76 84 78 81 77 83 72  
## issue14 98 97 91 88 87 84 89 82 92 89  
## issue15 85 86 92 80 81 77 78 74 82 78  
## issue16 101 97 89 89 97 84 74 78 88 88  
## issue11 issue12 issue13 issue14 issue15 issue16  
## issue1 102 101 93 98 85 101  
## issue2 109 99 86 97 86 97  
## issue3 99 95 81 91 92 89  
## issue4 91 91 76 88 80 89  
## issue5 93 85 84 87 81 97  
## issue6 85 74 78 84 77 84  
## issue7 105 85 81 89 78 74  
## issue8 95 74 77 82 74 78  
## issue9 103 86 83 92 82 88  
## issue10 83 78 72 89 78 88  
## issue11 0 92 89 101 96 87  
## issue12 92 0 84 80 82 84  
## issue13 89 84 0 85 74 81  
## issue14 101 80 85 0 76 85  
## issue15 96 82 74 76 0 83  
## issue16 87 84 81 85 83 0

接着我们可以以同样的方式计算中立媒体以及建制派新闻媒体16\*16的属性网络矩阵。

# 中立媒体的属性关系矩阵  
  
Media\_neutr <- Media\_neutr %>% select(issue1:issue16)  
  
Media\_neutr <- as.matrix(Media\_neutr)  
Media\_neutr\_T <- t(Media\_neutr)  
Media\_neutr\_matrix <- Media\_neutr\_T %\*% Media\_neutr  
  
Media\_neutr\_matrix <- Media\_neutr\_matrix %>% as.matrix  
diag(Media\_neutr\_matrix) <- 0  
  
Media\_neutr\_matrix

## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10  
## issue1 0 94 97 90 87 87 82 93 79 90  
## issue2 94 0 96 89 92 86 89 93 84 94  
## issue3 97 96 0 87 99 81 87 83 88 92  
## issue4 90 89 87 0 99 89 92 100 89 93  
## issue5 87 92 99 99 0 86 82 84 89 96  
## issue6 87 86 81 89 86 0 82 86 87 85  
## issue7 82 89 87 92 82 82 0 77 87 89  
## issue8 93 93 83 100 84 86 77 0 87 89  
## issue9 79 84 88 89 89 87 87 87 0 94  
## issue10 90 94 92 93 96 85 89 89 94 0  
## issue11 93 85 92 105 83 87 80 80 85 88  
## issue12 83 82 76 83 78 74 70 72 74 82  
## issue13 82 84 69 89 82 87 74 76 82 91  
## issue14 86 93 105 100 96 87 84 89 97 92  
## issue15 99 86 93 103 92 98 94 82 90 96  
## issue16 84 91 91 92 85 84 78 81 94 89  
## issue11 issue12 issue13 issue14 issue15 issue16  
## issue1 93 83 82 86 99 84  
## issue2 85 82 84 93 86 91  
## issue3 92 76 69 105 93 91  
## issue4 105 83 89 100 103 92  
## issue5 83 78 82 96 92 85  
## issue6 87 74 87 87 98 84  
## issue7 80 70 74 84 94 78  
## issue8 80 72 76 89 82 81  
## issue9 85 74 82 97 90 94  
## issue10 88 82 91 92 96 89  
## issue11 0 80 89 98 90 91  
## issue12 80 0 75 73 83 69  
## issue13 89 75 0 78 80 80  
## issue14 98 73 78 0 91 87  
## issue15 90 83 80 91 0 90  
## issue16 91 69 80 87 90 0

# 建制派媒体的属性关系矩阵  
  
Media\_proes <- Media\_proes %>% select(issue1:issue16)  
  
Media\_proes <- as.matrix(Media\_proes)  
Media\_proes\_T <- t(Media\_proes)  
Media\_proes\_matrix <- Media\_proes\_T %\*% Media\_proes  
  
Media\_proes\_matrix <- Media\_proes\_matrix %>% as.matrix  
diag(Media\_proes\_matrix) <- 0  
  
Media\_proes\_matrix

## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10  
## issue1 0 91 95 83 96 88 91 91 98 90  
## issue2 91 0 77 90 85 87 79 65 82 92  
## issue3 95 77 0 83 86 84 85 82 79 89  
## issue4 83 90 83 0 90 81 87 77 93 91  
## issue5 96 85 86 90 0 90 82 70 88 89  
## issue6 88 87 84 81 90 0 87 74 81 93  
## issue7 91 79 85 87 82 87 0 79 83 94  
## issue8 91 65 82 77 70 74 79 0 82 87  
## issue9 98 82 79 93 88 81 83 82 0 90  
## issue10 90 92 89 91 89 93 94 87 90 0  
## issue11 103 80 96 84 93 87 89 89 87 99  
## issue12 101 97 93 94 81 92 90 74 90 96  
## issue13 82 72 84 77 79 82 82 79 79 88  
## issue14 96 87 88 96 93 88 96 88 96 99  
## issue15 99 79 88 83 85 85 92 83 89 90  
## issue16 100 83 81 86 91 84 86 88 79 102  
## issue11 issue12 issue13 issue14 issue15 issue16  
## issue1 103 101 82 96 99 100  
## issue2 80 97 72 87 79 83  
## issue3 96 93 84 88 88 81  
## issue4 84 94 77 96 83 86  
## issue5 93 81 79 93 85 91  
## issue6 87 92 82 88 85 84  
## issue7 89 90 82 96 92 86  
## issue8 89 74 79 88 83 88  
## issue9 87 90 79 96 89 79  
## issue10 99 96 88 99 90 102  
## issue11 0 94 82 97 88 88  
## issue12 94 0 84 92 84 90  
## issue13 82 84 0 87 80 85  
## issue14 97 92 87 0 88 96  
## issue15 88 84 80 88 0 90  
## issue16 88 90 85 96 90 0

### 步骤三：公众关联网络矩阵（认知属性网络）

公众关联网络矩阵的建立则与媒体矩阵稍有不同，主要差别在于，我们需要考虑到每个受访者所回答的属性关联强度。因此，我们需要首先对所有可能的出现的属性建立对应关系（五个属性最大可能出现10对关系），然后与问卷中与每一对属性的关联强度进行一一映射，最后还要将每一对属性之间（pairs of every two given issues) 的总关系强度进行整合。

以泛民支持者的认知属性网络为例。我们首先需要选取五个属性变量（issue1 ～ issue5）以及十个属性间关系变量（association1 ～ association10）。

# 选取属性变量  
D\_self\_issues <- Democratic %>% select(respid, issue1:issue5)  
  
head(D\_self\_issues)

## # A tibble: 6 x 6  
## respid issue1 issue2 issue3 issue4 issue5  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 411 12 8 11 7 16  
## 2 53 4 14 16 15 15  
## 3 489 7 12 1 5 4  
## 4 207 1 9 12 3 16  
## 5 178 12 14 2 7 15  
## 6 422 12 8 5 1 6

# 选取属性关系变量  
D\_self\_weights <- Democratic %>% select(respid, association1:association10)  
  
head(D\_self\_weights)

## # A tibble: 6 x 11  
## respid association1 association2 association3 association4 association5  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 411 2 6 7 1 0  
## 2 53 5 0 6 2 1  
## 3 489 7 4 7 6 4  
## 4 207 2 2 1 9 3  
## 5 178 9 9 6 0 8  
## 6 422 7 3 9 6 3  
## # … with 5 more variables: association6 <dbl>, association7 <dbl>,  
## # association8 <dbl>, association9 <dbl>, association10 <dbl>

然后对数据中的每一对属性（a pair of issues）建立一种关系结构，即边关系列表（edgelist）。对每一条数据而言，五种属性一共有10种关系（issue1和issue2、issue1和issue3、issue和issue4…距此类推）。因此500条数据则一共有100\*10 = 5,000条关系。

# 建立每一对属性之间的边关系结构  
D\_self = data.frame()  
for (i in 1: length(dimnames(D\_self\_issues)[[1]])) {  
 v = unlist(D\_self\_issues[i, 2:6])   
 v = t(combn(v,2))  
 D\_self = rbind(D\_self, v)  
}   
D\_self <- t(apply(D\_self,1,sort)) %>% as.data.frame() # 在这里，属性之间的关系是无方向的  
  
head(D\_self)

## V1 V2  
## 1 8 12  
## 2 11 12  
## 3 7 12  
## 4 12 16  
## 5 8 11  
## 6 7 8

如上表所示的边关系数据，V1和V2分别是两种属性，由于公众提供的具体话题有可能属于同一种属性，因此V1和V2也可能属于同一种属性。

接下来，我们将属性和属性之间的强度加到边关系数据中：

# 增加属性之间的关系强度值（weights）  
  
# 提取属性之间的强度值  
D\_self\_w = c()  
for (i in 1: length(dimnames(D\_self\_weights)[[1]])) {  
 w = unlist(D\_self\_weights[i, 2:11])  
 w = t(combn(w,1))  
 D\_self\_w = append(D\_self\_w,w)  
}  
D\_self$weight <- D\_self\_w # 将属性之间的强度值赋予边关系列表  
D\_self <- aggregate(weight~V1+V2, data = D\_self, FUN = sum) # 整合同属性对之间的关系强度，只保留每一对独特的边关系   
  
head(D\_self)

## V1 V2 weight  
## 1 1 1 67  
## 2 1 2 112  
## 3 2 2 94  
## 4 1 3 250  
## 5 2 3 231  
## 6 3 3 101

将边关系列表转换为矩阵形式：

D\_self\_matrix <- AdjacencyFromEdgelist(D\_self, check.full = TRUE) %>% as.data.frame() # 边关系列表转换为矩阵  
D\_self\_matrix <- D\_self\_matrix[1:16, 1:16] #去除最后一列nodelist  
  
D\_self\_matrix # 展示数据

## adjacency.1 adjacency.2 adjacency.3 adjacency.4 adjacency.5 adjacency.6  
## 1 67 112 250 183 144 180  
## 2 0 94 231 191 236 227  
## 3 0 0 101 254 228 263  
## 4 0 0 0 71 205 156  
## 5 0 0 0 0 104 241  
## 6 0 0 0 0 0 62  
## 7 0 0 0 0 0 0  
## 8 0 0 0 0 0 0  
## 9 0 0 0 0 0 0  
## 10 0 0 0 0 0 0  
## 11 0 0 0 0 0 0  
## 12 0 0 0 0 0 0  
## 13 0 0 0 0 0 0  
## 14 0 0 0 0 0 0  
## 15 0 0 0 0 0 0  
## 16 0 0 0 0 0 0  
## adjacency.7 adjacency.8 adjacency.9 adjacency.10 adjacency.11 adjacency.12  
## 1 183 153 165 177 161 240  
## 2 170 153 147 167 181 236  
## 3 240 198 269 241 209 287  
## 4 164 225 225 280 214 200  
## 5 202 207 245 207 152 242  
## 6 194 228 255 204 163 235  
## 7 116 175 213 205 174 203  
## 8 0 81 179 183 203 142  
## 9 0 0 126 211 155 253  
## 10 0 0 0 93 194 212  
## 11 0 0 0 0 82 175  
## 12 0 0 0 0 0 95  
## 13 0 0 0 0 0 0  
## 14 0 0 0 0 0 0  
## 15 0 0 0 0 0 0  
## 16 0 0 0 0 0 0  
## adjacency.13 adjacency.14 adjacency.15 adjacency.16  
## 1 102 141 152 180  
## 2 175 187 193 92  
## 3 135 211 153 207  
## 4 163 252 171 154  
## 5 169 326 144 202  
## 6 215 185 152 203  
## 7 183 187 183 190  
## 8 182 179 159 215  
## 9 245 207 159 213  
## 10 153 256 184 178  
## 11 231 169 146 195  
## 12 146 196 144 193  
## 13 99 177 143 133  
## 14 0 86 214 179  
## 15 0 0 105 149  
## 16 0 0 0 70

对认知属性矩阵进行重命名和调整：

colnames(D\_self\_matrix) <- paste0("issue",seq(1:16),"")  
rownames(D\_self\_matrix) <- paste0("issue",seq(1:16),"")  
  
D\_self\_matrix <- D\_self\_matrix %>% as.matrix  
diag(D\_self\_matrix) <- 0 # 去除属性自身之间的强度值  
lowerTriangle(D\_self\_matrix) <- upperTriangle(D\_self\_matrix, byrow = TRUE) #生成整体矩阵  
  
D\_self\_matrix # 展示数据

## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10  
## issue1 0 112 250 183 144 180 183 153 165 177  
## issue2 112 0 231 191 236 227 170 153 147 167  
## issue3 250 231 0 254 228 263 240 198 269 241  
## issue4 183 191 254 0 205 156 164 225 225 280  
## issue5 144 236 228 205 0 241 202 207 245 207  
## issue6 180 227 263 156 241 0 194 228 255 204  
## issue7 183 170 240 164 202 194 0 175 213 205  
## issue8 153 153 198 225 207 228 175 0 179 183  
## issue9 165 147 269 225 245 255 213 179 0 211  
## issue10 177 167 241 280 207 204 205 183 211 0  
## issue11 161 181 209 214 152 163 174 203 155 194  
## issue12 240 236 287 200 242 235 203 142 253 212  
## issue13 102 175 135 163 169 215 183 182 245 153  
## issue14 141 187 211 252 326 185 187 179 207 256  
## issue15 152 193 153 171 144 152 183 159 159 184  
## issue16 180 92 207 154 202 203 190 215 213 178  
## issue11 issue12 issue13 issue14 issue15 issue16  
## issue1 161 240 102 141 152 180  
## issue2 181 236 175 187 193 92  
## issue3 209 287 135 211 153 207  
## issue4 214 200 163 252 171 154  
## issue5 152 242 169 326 144 202  
## issue6 163 235 215 185 152 203  
## issue7 174 203 183 187 183 190  
## issue8 203 142 182 179 159 215  
## issue9 155 253 245 207 159 213  
## issue10 194 212 153 256 184 178  
## issue11 0 175 231 169 146 195  
## issue12 175 0 146 196 144 193  
## issue13 231 146 0 177 143 133  
## issue14 169 196 177 0 214 179  
## issue15 146 144 143 214 0 149  
## issue16 195 193 133 179 149 0

以相同的方式，我们可以建立建制派支持者的认知属性网络：

# 选取属性变量  
E\_self\_issues <- Establish %>% select(respid, issue1:issue5)  
  
# 选取属性关系变量  
E\_self\_weights <- Establish %>% select(respid, association1:association10)  
  
# 建立每一对属性之间的边关系结构  
E\_self = data.frame()  
for (i in 1: length(dimnames(E\_self\_issues)[[1]])) {  
 v = unlist(E\_self\_issues[i, 2:6])   
 v = t(combn(v,2))  
 E\_self = rbind(E\_self, v)  
}   
E\_self <- t(apply(E\_self,1,sort)) %>% as.data.frame()   
  
# 属性之间的关系强度值（weights）  
E\_self\_w = c()  
for (i in 1: length(dimnames(E\_self\_weights)[[1]])) {  
 w = unlist(E\_self\_weights[i, 2:11])  
 w = t(combn(w,1))  
 E\_self\_w = append(E\_self\_w,w)  
}  
E\_self$weight <- E\_self\_w  
  
E\_self <- aggregate(weight~V1+V2, data = E\_self, FUN = sum)  
  
# 建立认知属性矩阵  
E\_self\_matrix <- AdjacencyFromEdgelist(E\_self, check.full = TRUE) %>% as.data.frame()  
E\_self\_matrix <- E\_self\_matrix[1:16, 1:16]  
  
# 对认知属性矩阵进行重命名和调整  
colnames(E\_self\_matrix) <- paste0("issue",seq(1:16),"")  
rownames(E\_self\_matrix) <- paste0("issue",seq(1:16),"")  
  
E\_self\_matrix <- E\_self\_matrix %>% as.matrix  
diag(E\_self\_matrix) <- 0 # 去除属性自身之间的强度值  
lowerTriangle(E\_self\_matrix) <- upperTriangle(E\_self\_matrix, byrow = TRUE) #生成整体矩阵  
  
E\_self\_matrix

## issue1 issue2 issue3 issue4 issue5 issue6 issue7 issue8 issue9 issue10  
## issue1 0 212 164 89 249 188 198 227 120 135  
## issue2 212 0 201 212 166 182 167 198 178 176  
## issue3 164 201 0 116 255 273 195 191 197 217  
## issue4 89 212 116 0 164 136 183 163 177 206  
## issue5 249 166 255 164 0 238 235 237 215 175  
## issue6 188 182 273 136 238 0 197 227 183 201  
## issue7 198 167 195 183 235 197 0 216 182 132  
## issue8 227 198 191 163 237 227 216 0 159 231  
## issue9 120 178 197 177 215 183 182 159 0 97  
## issue10 135 176 217 206 175 201 132 231 97 0  
## issue11 242 212 227 152 215 279 199 241 222 169  
## issue12 202 272 213 119 158 174 145 210 165 154  
## issue13 102 208 202 167 147 244 147 201 128 137  
## issue14 171 204 247 178 265 175 281 242 178 226  
## issue15 157 118 201 164 257 219 238 207 176 200  
## issue16 176 259 210 191 222 292 185 226 222 214  
## issue11 issue12 issue13 issue14 issue15 issue16  
## issue1 242 202 102 171 157 176  
## issue2 212 272 208 204 118 259  
## issue3 227 213 202 247 201 210  
## issue4 152 119 167 178 164 191  
## issue5 215 158 147 265 257 222  
## issue6 279 174 244 175 219 292  
## issue7 199 145 147 281 238 185  
## issue8 241 210 201 242 207 226  
## issue9 222 165 128 178 176 222  
## issue10 169 154 137 226 200 214  
## issue11 0 205 178 209 167 214  
## issue12 205 0 236 182 226 231  
## issue13 178 236 0 175 142 215  
## issue14 209 182 175 0 202 209  
## issue15 167 226 142 202 0 187  
## issue16 214 231 215 209 187 0

### 步骤四：研究假设验证（MRQAP）

在此，我们使用MRQAP对新闻媒体的属性网络和公众的认知属性网络之关联性进行分析，并验证以下两个研究假设： \* 假设一：香港建制派新闻媒体的属性网络和建制派支持者的认知属性网络存在显著关联。 \* 假设二：香港泛民派新闻媒体的属性网络和泛民派支持者的认知属性网络存在显著关联。

在这里我们使用的R包是“asnipe”，实际上也可以使用sna包中的netlm函数

验证假设一：

proesfit <- mrqap.dsp(E\_self\_matrix ~ Media\_proes\_matrix + Media\_democ\_matrix + Media\_neutr\_matrix, intercept = TRUE, directed = "undirected", diagonal = FALSE, test.statistic = "t-value", tol = 1e-07, randomisations = 1000)

验证假设二：

democfit <- mrqap.dsp(D\_self\_matrix ~ Media\_proes\_matrix + Media\_democ\_matrix + Media\_neutr\_matrix, intercept = TRUE, directed = "undirected", diagonal = FALSE, test.statistic = "t-value", tol = 1e-07, randomisations = 1000)