
인문학 텍스트 마이닝

D3.js based data visualization

☐ ☐ ☐ ☐

- D3.js

[Overview](#) [Examples](#) [Documentation](#) [Source](#)



<http://d3js.org/>

D3.js based data visualization

● 패키지 설치

```
> install.packages("devtools")
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0     0     0      0      0  --:--:--  --:--:--  --:--:--    0  90  335k   90  304k
  0     0  1602k    0  --:--:--  --:--:--  --:--:--  1600k100  335k  100  335k    0     0  1725k    0
--:--:--  --:--:--  --:--:--  1720k
```

The downloaded binary packages are in
/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpRELQRF/downloaded_packages

```
> library(devtools)
> install.packages("RCurl")
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0     0     0      0      0  --:--:--  --:--:--  --:--:--    0  44  849k   44  375k
  0     0   496k    0  0:00:01  --:--:--  0:00:01  496k100  849k  100  849k    0     0   975k    0
--:--:--  --:--:--  --:--:--   975k
```

The downloaded binary packages are in
/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpRELQRF/downloaded_packages

```
> library(RCurl)
필요한 패키지를 로딩중입니다: bitops
> install_github("christophergandrud/d3Network")
Downloading GitHub repo christophergandrud/d3Network@master
Installing d3Network
'/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file --no-environ --no-save \
--no-restore CMD INSTALL \

'/private/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpRELQRF/devtools3dd324c44e3c/christophe
rgandrud-d3Network-dc76a26' \
--library='/Library/Frameworks/R.framework/Versions/3.2/Resources/library' --install-tests
```

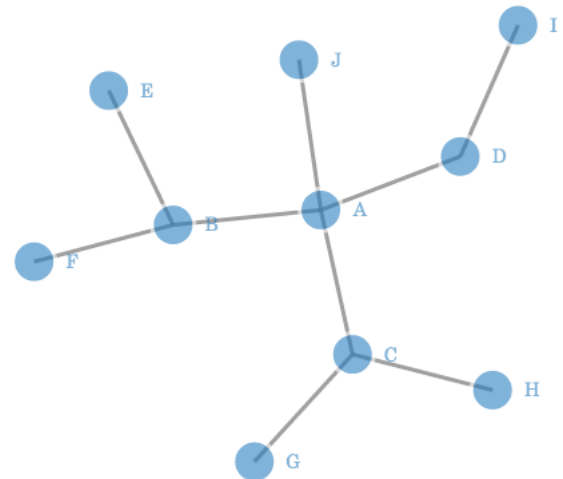
```
* installing *source* package 'd3Network' ...
** R
** data
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (d3Network)
> library(d3Network)
```

D3.js based data visualization

● 데이터 방향 지정 및 생성

```
> Source<-c("A","A","A","A","B","B","C","C","D")
> Target<-c("B","C","D","J","E","F","G","H","I")
> NetworkData<-data.frame(Source,Target)
> head(NetworkData)
```

	Source	Target
1	A	B
2	A	C
3	A	D
4	A	J
5	B	E
6	B	F

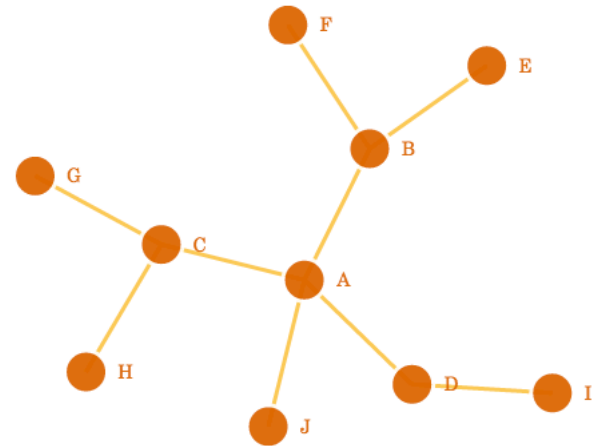


● 네트워크 시각화 구현

```
> ericOpenHtml<-function(filename){if(Sys.info()["sysname"]=="windows"){shell.exec(filename)}else{system(paste("open",filename))}}
> d3SimpleNetwork(NetworkData,width=400,height=250,file="test1.html")
> ericOpenHtml("test1.html")
```

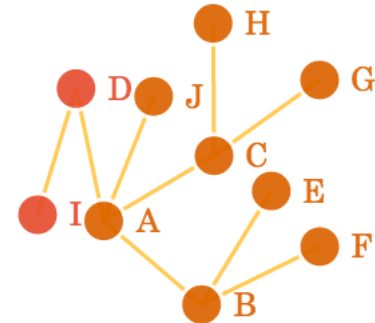
D3.js based data visualization

— — — —



● 노드와 링크에 대한 색상 변경

```
> d3SimpleNetwork(NetworkData, width = 400, height = 250, textColour = "#D95F0E", linkColour = "#FEC44F", nodeColour = "#D95F0E", opacity = 0.9, file="test2.html")  
> ericOpenHtml("test2.html")
```



● Charge(노드간의 인력과 척력)값 변경

```
> d3SimpleNetwork(NetworkData, width = 400, height = 250, textColour = "#D95F0E", linkColour = "#FEC44F", nodeColour = "#D95F0E", opacity = 0.9, charge = -50, fontsize = 12, file="test3.html")  
> ericOpenHtml("test3.html")
```

D3.js based data visualization

● 패키지 설치

```
> install.packages("RCurl")
```

```
% Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
                               Dload  Upload    Total   Spent    Left   Speed

  0      0    0     0    0     0      0     0 --:--:-- --:--:-- --:--:--    0 60  849k   60  511k
  0      0 1039k    0 --:--:-- --:--:-- --:--:-- 1039k100  849k   100  849k    0    0 1430k    0
--:--:-- --:--:-- --:--:-- 1429k
```

The downloaded binary packages are in

/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpfGSRj8/downloaded_packages

```
> install.packages("d3Network")
```

```
% Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
                               Dload  Upload    Total   Spent    Left   Speed

  0      0    0     0    0     0      0     0 --:--:-- --:--:-- --:--:--    0  0 65600    0    0
  0      0    0     0 --:--:-- 0:00:01 --:--:--    0 24 65600   24 16384    0    0  8813    0
0:00:07 0:00:01 0:00:06 8813100 65600  100 65600    0    0 28892    0 0:00:02 0:00:02 --:--
:-- 28898
```

The downloaded binary packages are in

/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpfGSRj8/downloaded_packages

```
> library(d3Network)
```

```
> library(RCurl)
```

필요한 패키지를 로딩중입니다: bitops

D3.js based data visualization

● 데이터 링크 확인

```
> URL="https://raw.githubusercontent.com/christophergandrud/d3Network/master/JSONdata/miserables.js  
n"
```

```
> MisJson<-getURL(URL,ssl.verifypeer=FALSE)
```

```
{  
  "nodes": [  
    {"name": "Myriel", "group": 1},  
    {"name": "Napoleon", "group": 1},  
    {"name": "Mlle.Baptistine", "group": 1},  
    {"name": "Mme.Magloire", "group": 1},  
    {"name": "CountessdeLo", "group": 1},  
    {"name": "Geborand", "group": 1},  
    {"name": "Champtercier", "group": 1},  
    {"name": "Cravatte", "group": 1},  
    {"name": "Count", "group": 1},  
    {"name": "OldMan", "group": 1},  
    {"name": "Labarre", "group": 2},  
    {"name": "Valjean", "group": 2},  
    {"name": "Marguerite", "group": 3},  
    {"name": "Mme.deR", "group": 2},  
    {"name": "Isabeau", "group": 2},  
    {"name": "Gervais", "group": 2},  
    {"name": "Tholomyes", "group": 3},  
    {"name": "Listolier", "group": 3},  
    {"name": "Fameuil", "group": 3},  
    {"name": "Blacheville", "group": 3},  
    {"name": "Favourite", "group": 3},  
    {"name": "Dahlia", "group": 3},  
    {"name": "Zephine", "group": 3},  
    {"name": "Fantine", "group": 3},  
    {"name": "Mme.Thenardier", "group": 4},  
    {"name": "Thenardier", "group": 4},  
    {"name": "Cosette", "group": 5},  
    {"name": "Fauchelevent", "group": 0},  
    {"name": "Bamatabc  
etue", "group": 3},  
    {"name": "Simplice", "group": 2},  
    {"name": "r
```

```
[1] "{\n  \"nodes\": [\n    {\n      \"name\": \"Myriel\", \"group\": 1,\n      \"name\": \"Mlle.Baptistine\", \"group\": 1,\n      \"name\": \"Mme.Magloire\", \"group\": 1,\n      \"name\": \"CountessdeLo\", \"group\": 1,\n      \"name\": \"Geborand\", \"group\": 1,\n      \"name\": \"Champtercier\", \"group\": 1,\n      \"name\": \"Cravatte\", \"group\": 1,\n      \"name\": \"Count\", \"group\": 1,\n      \"name\": \"OldMan\", \"group\": 1,\n      \"name\": \"Labarre\", \"group\": 2,\n      \"name\": \"Valjean\", \"group\": 2,\n      \"name\": \"Marguerite\", \"group\": 3,\n      \"name\": \"Mme.deR\", \"group\": 2,\n      \"name\": \"Isabeau\", \"group\": 2,\n      \"name\": \"Gervais\", \"group\": 2,\n      \"name\": \"Tholomyes\", \"group\": 3,\n      \"name\": \"Listolier\", \"group\": 3,\n      \"name\": \"Fameuil\", \"group\": 3,\n      \"name\": \"Blacheville\", \"group\": 3,\n      \"name\": \"Favourite\", \"group\": 3,\n      \"name\": \"Dahlia\", \"group\": 3,\n      \"name\": \"Zephine\", \"group\": 3,\n      \"name\": \"Fantine\", \"group\": 3,\n      \"name\": \"Mme.Thenardier\", \"group\": 4,\n      \"name\": \"Thenardier\", \"group\": 4,\n      \"name\": \"Cosette\", \"group\": 5,\n      \"name\": \"Fauchelevent\", \"group\": 0,\n      \"name\": \"Bamatabc  
etue\", \"group\": 3,\n      \"name\": \"Simplice\", \"group\": 2,\n      \"name\": \"r
```


D3.js based data visualization

● 데이터 확인

```
> MisLinks<-JSONtoDF(jsonStr=MisJson,array="links")
```

```
> head(MisLinks)
```

	source	target	value
1	1	0	1
2	2	0	8
3	3	0	10
4	3	2	6
5	4	0	1
6	5	0	1

```
> MisNodes<-JSONtoDF(jsonStr=MisJson,array="nodes")
```

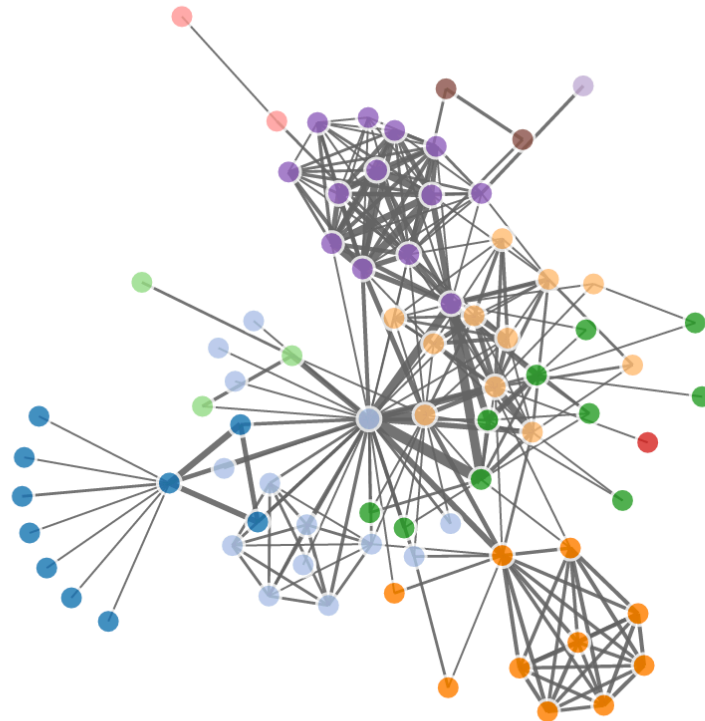
```
> head(MisNodes)
```

	name	group
1	Myriel	1
2	Napoleon	1
3	Mlle.Baptistine	1
4	Mme.Magloire	1
5	CountessdeLo	1
6	Geborand	1

D3.js based data visualization

● Forced-Direct Network 생성

```
> ericOpenHtml<-function(filename){if(Sys.info()["sysname"]=="windows"){shell.exec(filename)}else{system(paste("open",filename))}}  
> d3ForceNetwork(Links=MisLinks,Nodes=MisNodes,Source="source",Target="target",Value="value",NodeID="name",Group="group",width=1200,height=800,opacity=0.9,zoom=TRUE,file="test4.html")  
> ericOpenHtml("test4.html")
```



☐ ☐ ☐ ☐

```
> library(d3Network)
```

D3.js based data visualization

● Forced-Direct Network 직접 만들기

• 데이터 생성

```
> source_c=c(1,2,3,3,4,5,6,7,0)
> target_c=c(2,0,0,2,0,0,0,0,3)
> value_c=c(1,8,10,6,1,1,1,1,20)
> ML<-data.frame(source_c=source_c,target_c=target_c,value_c=value_c)
> head(ML)
  source_c target_c value_c
1         1         2         1
2         2         0         8
3         3         0        10
4         3         2         6
5         4         0         1
6         5         0         1
>
> node_c=c("A","B","C","D","E","F","G","H")
> group_c=c(1,1,1,1,2,2,2,2)
> MN=data.frame(node_c=node_c,group_c=group_c)
> head(MN)
  node_c group_c
1      A         1
2      B         1
3      C         1
4      D         1
5      E         2
6      F         2
```

D3.js based data visualization

● Forced-Direct Network 직접 만들기

• 시각화 생성

```
> ericOpenHtml<-function(filename){if(Sys.info()["sysname"]=="windows"){shell.exec(filename)}else{system(paste("open",filename))}}  
> d3ForceNetwork(Links = ML, Nodes = MN,Source = "source_c", Target = "target_c",Value = "value_c",  
NodeID = "node_c",Group = "group_c", width = 550, height = 400,opacity = 0.9, zoom = TRUE,file="test  
5.html")  
> ericOpenHtml("test5.html")
```



D3.js based data visualization

● D3tree만들기

• 패키지 설치

```
> install.packages("d3Network")
```

```
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
                               Dload  Upload    Total   Spent    Left   Speed
  0      0    0       0      0      0      0      0 --:--:-- --:--:-- --:--:--    0 100 65600  100 65600
0      0  431k      0 --:--:-- --:--:-- --:--:--  432k
```

The downloaded binary packages are in

/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//Rtmpy7rNC7/downloaded_packages

```
> library(d3Network)
```

```
> install.packages("RCurl")
```

```
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
                               Dload  Upload    Total   Spent    Left   Speed
  0      0    0       0      0      0      0      0 --:--:-- --:--:-- --:--:--    0 11  849k  11 97431
0      0  561k      0 0:00:01 --:--:-- 0:00:01 559k100  849k  100  849k    0    0 2995k    0
--:--:-- --:--:-- --:--:-- 2990k
```

The downloaded binary packages are in

/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//Rtmpy7rNC7/downloaded_packages

```
> library(RCurl)
```

필요한 패키지를 로딩중입니다: bitops

D3.js based data visualization

● D3tree만들기

- 데이터 확인하기

```
> URL <- "https://raw.githubusercontent.com/christophergandrud/d3Network/master/JSONdata/flare.json"
```

```
{
  "name": "flare",
  "children": [
    {
      "name": "analytics",
      "children": [
        {
          "name": "cluster",
          "children": [
            {"name": "AgglomerativeCluster", "size": 3938},
            {"name": "CommunityStructure", "size": 3812},
            {"name": "HierarchicalCluster", "size": 6714},
            {"name": "MergeEdge", "size": 743}
          ]
        },
      ],
    },
    {
      "name": "graph",
      "children": [
        {"name": "BetweennessCentrality", "size": 3534},
        {"name": "LinkDistance", "size": 5731},
        {"name": "MaxFlowMinCut", "size": 7840},
        {"name": "ShortestPaths", "size": 5914},
        {"name": "SpanningTree", "size": 3416}
      ]
    },
    {
      "name": "optimization",
      "children": [
        {"name": "AspectRatioBanker", "size": 7074}
      ]
    }
  ],
  "animate": true
}
```

D3.js based data visualization

● D3tree만들기

- 데이터 크롤링하기

```
> Flare <- getURL(URL)
> head(Flare)
```

```
[1] "{\n  \"name\": \"flare\", \n  \"children\": [\n    {\n      \"name\": \"analytics\n    {\n      \"name\": \"cluster\", \n      \"children\": [\n        {\n          \"name\": \"CommunityStructure\", \"size\": 3812\n        },\n        {\n          \"name\": \"HierarchicalCluster\", \"size\": 6714\n        },\n        {\n          \"name\": \"MergeEdge\", \"size\": 3534\n        },\n        {\n          \"name\": \"LinkDistance\", \"size\": 5731\n        },\n        {\n          \"name\": \"ShortestPaths\", \"size\": 591\n        },\n        {\n          \"name\": \"SpanningTree\", \"size\": 3416\n        }\n      ],\n      \"name\": \"opt\n    {\n      \"name\": \"AspectRatioBanker\", \"size\": 7074\n    },\n    {\n      \"name\": \"animate\", \n      \"children\": [\n        {\n          \"name\": \"Easing\", \n          \"name\": \"FunctionSequence\", \"size\": 5842\n        },\n        {\n          \"name\": \"int\n        {\n          \"name\": \"ArrayInterpolator\", \"size\": 1983\n        },\n        {\n          \"name\": \"DateInterpolator\", \"size\": 8746\n        },\n        {\n          \"name\": \"MatrixInterpolator\", \n          \"name\": \"NumberInterpolator\", \"size\": 1382\n        },\n        {\n          \"name\": \"Object\n        {\n          \"name\": \"PointInterpolator\", \"size\": 1675\n        }\n      ],\n    }\n  ]\n}"
```


D3.js based data visualization

● D3tree만들기

- List형태로 변환

```
> Flare <- rjson::fromJSON(Flare)
> head(Flare)
```

```
$children[[10]]
$children[[10]]$name
[1] "vis"
```

```
$children[[10]]$children
$children[[10]]$children[[1]]
$children[[10]]$children[[1]]$name
[1] "axis"
```

```
$children[[10]]$children[[1]]$children
$children[[10]]$children[[1]]$children[[1]]
$children[[10]]$children[[1]]$children[[1]]$name
[1] "Axes"
```

```
$children[[10]]$children[[1]]$children[[1]]$size
[1] 1302
```

☐ ☐ ☐ ☐

- 시각화 생성



```
> ericOpenHtml<-function(filename){if(Sys.info()["sysname"]=="windows"){shell.exec(filename)}else{system(paste("open",filename))}}
> d3Tree(List = Flare, width = 1000, height = 1000, fontsize = 8, diameter = 900, file="test6.html")
> ericOpenHtml("test6.html")
```

D3.js based data visualization

● D3tree만들기

• 시각화 생성

```
> ericOpenHtml<-function(filename){if(Sys.info()["sysname"]=="windows"){shell.exec(filename)}else{system(paste("open",filename))}}  
> d3ClusterDendro(List = Flare, fontsize = 12, zoom = TRUE, widthCollapse = 0.8, file="test7.html")  
> ericOpenHtml("test7.html")
```

