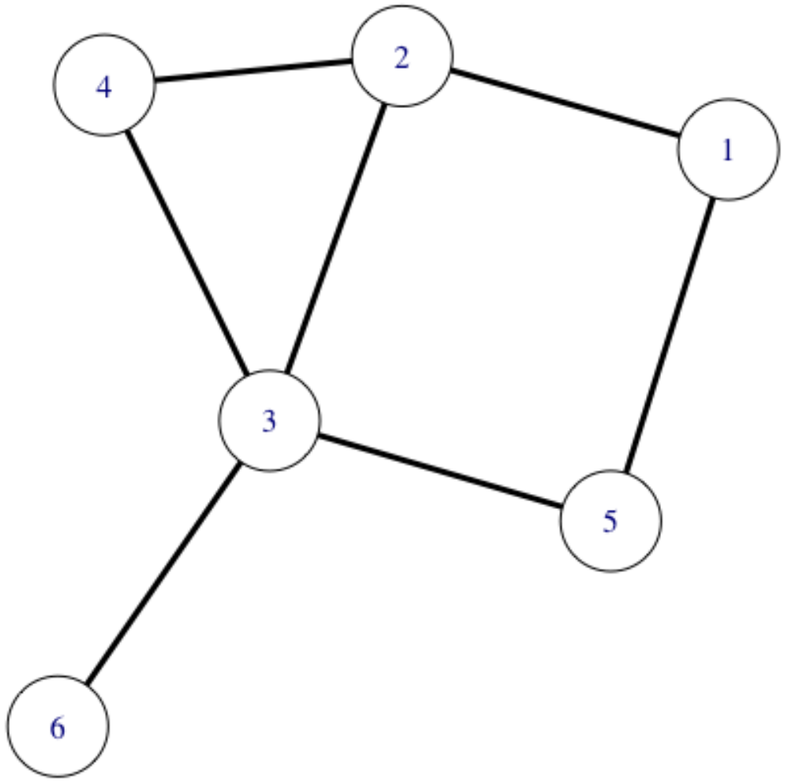
Connection patterns

NETWORK ANALYSIS IN THE TIDYVERSE

Massimo Franceschet

Prof. of Data Science, University of Udine (Italy)

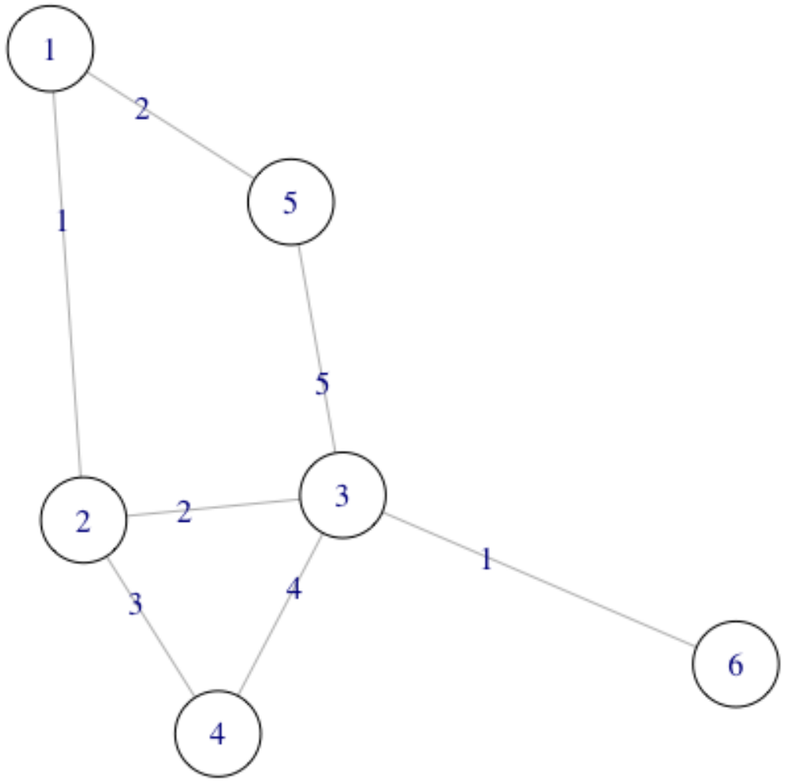
The adjacency matrix (part 1)




```
as_adjacency_matrix(g)
```

```
   1 2 3 4 5 6
1  0 1 0 0 1 0
2  1 0 1 1 0 0
3  0 1 0 1 1 1
4  0 1 1 0 0 0
5  1 0 1 0 0 0
6  0 0 1 0 0 0
```

The adjacency matrix (part 2)




```
as_adjacency_matrix(g,
                    attr="weight")
```

	1	2	3	4	5	6
1	0	1	0	0	2	0
2	1	0	2	3	0	0
3	0	2	0	4	5	1
4	0	3	4	0	0	0
5	2	0	5	0	0	0
6	0	0	1	0	0	0

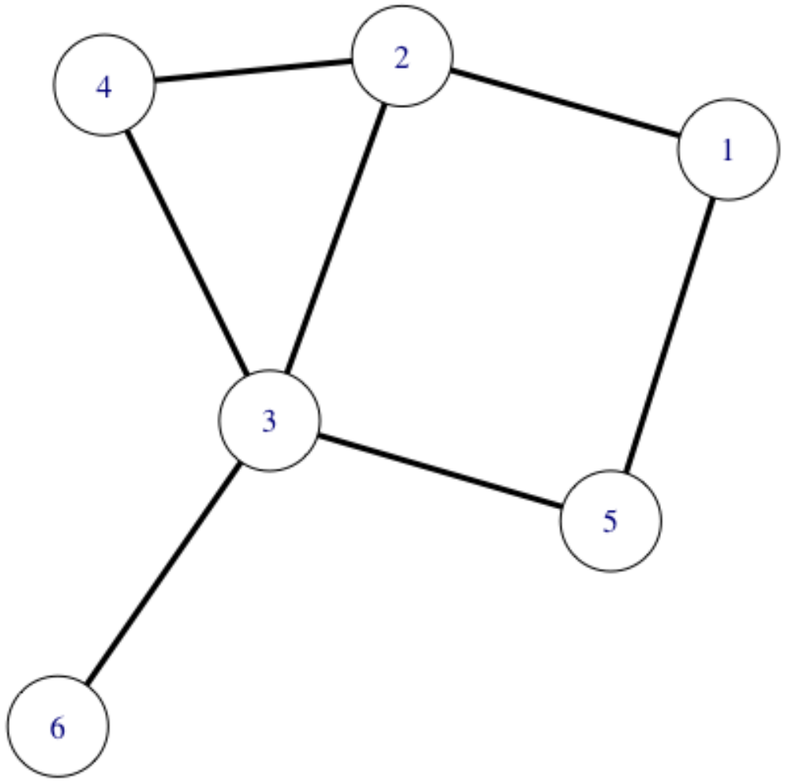
Working with adjacency matrices

```
# get the adjacency matrix of network g
A = as_adjacency_matrix(g)
# get the weighted adjacency matrix of weighted network g
A = as_adjacency_matrix(g, attr = "weight")
```

```
# first row of matrix A
A[1, ]

# first column of matrix A
A[, 1]
# diagonal of matrix A
diag(A)
```

Pearson similarity




```
as_adjacency_matrix(g)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1    0    0    1    0
[2,]    1    0    1    1    0    0
[3,]    0    1    0    1    1    1
[4,]    0    1    1    0    0    0
[5,]    1    0    1    0    0    0
[6,]    0    0    1    0    0    0
```

Let's try some examples!

NETWORK ANALYSIS IN THE TIDYVERSE

Pearson similarity

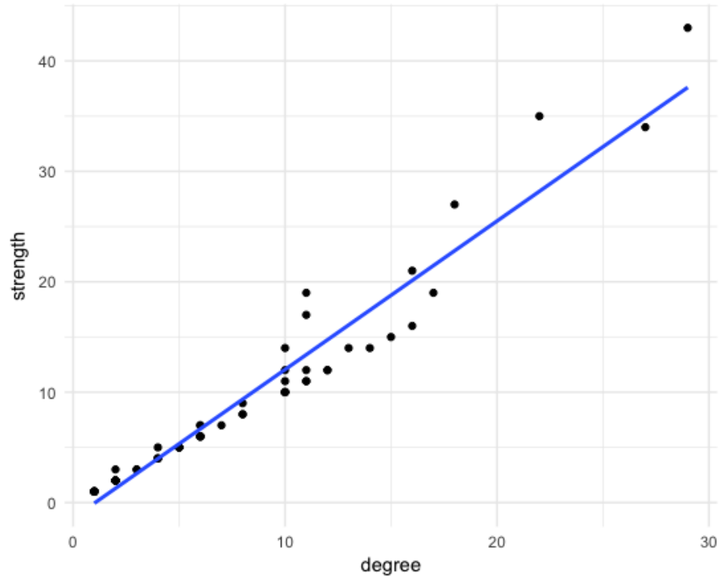
NETWORK ANALYSIS IN THE TIDYVERSE

Massimo Franceschet

Prof. of Data Science, University of Udine (Italy)

Visualizing correlation

```
# scatterplot of degree and strength
ggplot(data = nodes, mapping = aes(x = degree, y = strength)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)
```



Computing correlation

- Positive values indicate positive correlation
- Negative values indicate negative correlation
- Null values indicate no correlation

```
# Pearson correlation coefficient
cor(nodes$degree, nodes$strength)
```

```
0.9708946
```

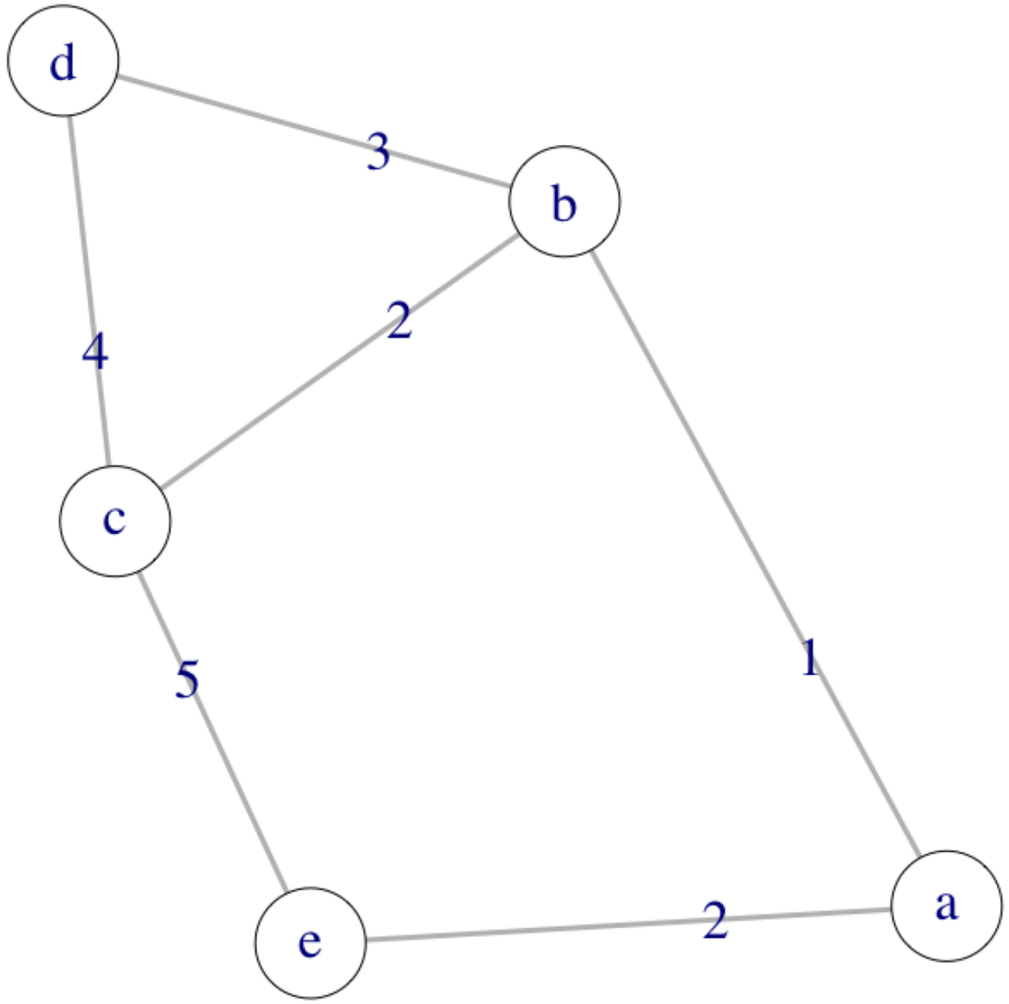
Let's practice!

NETWORK ANALYSIS IN THE TIDYVERSE

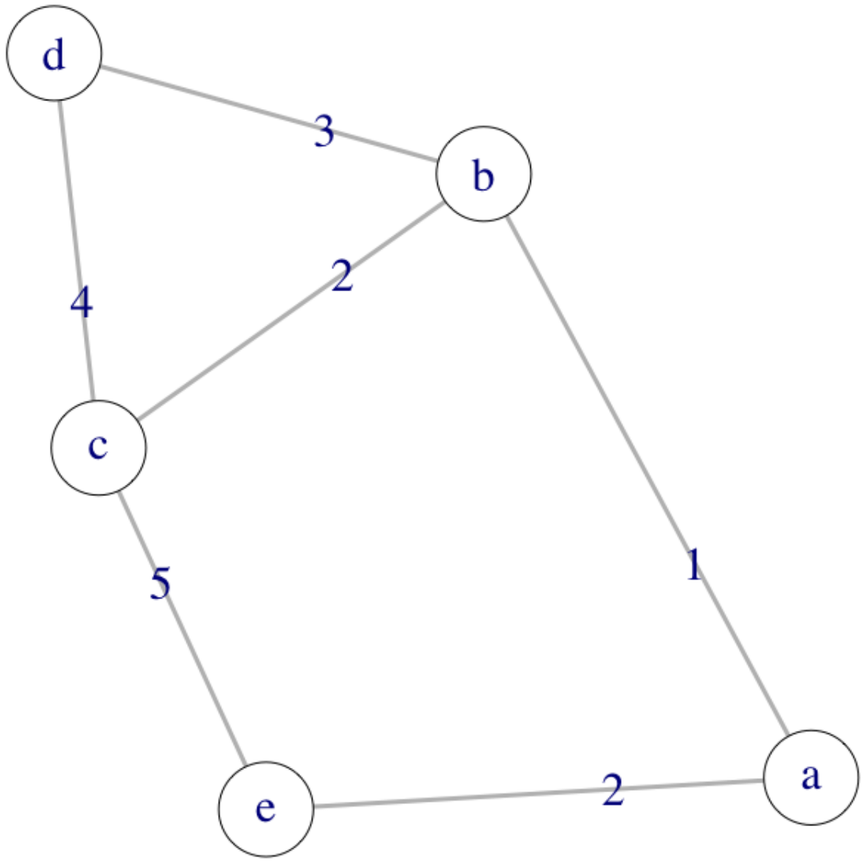
Most similar and most dissimilar nodes

NETWORK ANALYSIS IN THE TIDYVERSE

Massimo Franceschet
Prof. of Data Science, University of Udine (Italy)


d
3
b
2
4
c
5
1
e
2
a

A network as a matrix




```
as_adjacency_matrix(g,
                    attr = "weight")
```

```
  a b c d e
a 0 1 0 0 2
b 1 0 2 3 0
c 0 2 0 4 5
d 0 3 4 0 0
e 2 0 5 0 0
```

A network as a data frame

```
as_data_frame(g, what = "both")
```

```
$nodes
  name
a    a
b    b
c    c
d    d
e    e
```

```
$ties
  from to weight
1    a  b      1
2    a  e      2
3    b  c      2
4    b  d      3
5    c  d      4
6    c  e      5
```

Mapping representations

```
# graph to matrix
A <- as_adjacency_matrix(g)

# matrix to graph
g <- graph_from_adjacency_matrix(A)
# graph to data frame
df = as_data_frame(g, what = "both")

# data frame to graph
g <- graph_from_data_frame(df$ties, vertices = df$nodes)
# matrix to data frame
df = as_data_frame(graph_from_adjacency_matrix(A), what = "both")

# data frame to matrix
A <- as_adjacency_matrix(graph_from_data_frame(df$ties,
     vertices = df$nodes))
```

Let's try more examples!

NETWORK ANALYSIS IN THE TIDYVERSE