Príklady sietí, základné pojmy, základy práce so sieťami v R-ku

Beáta Stehlíková

2-EFM-155 Analýza sociálnych sietí

Fakulta matematiky, fyziky a informatiky, UK v Bratislave

Sylabus

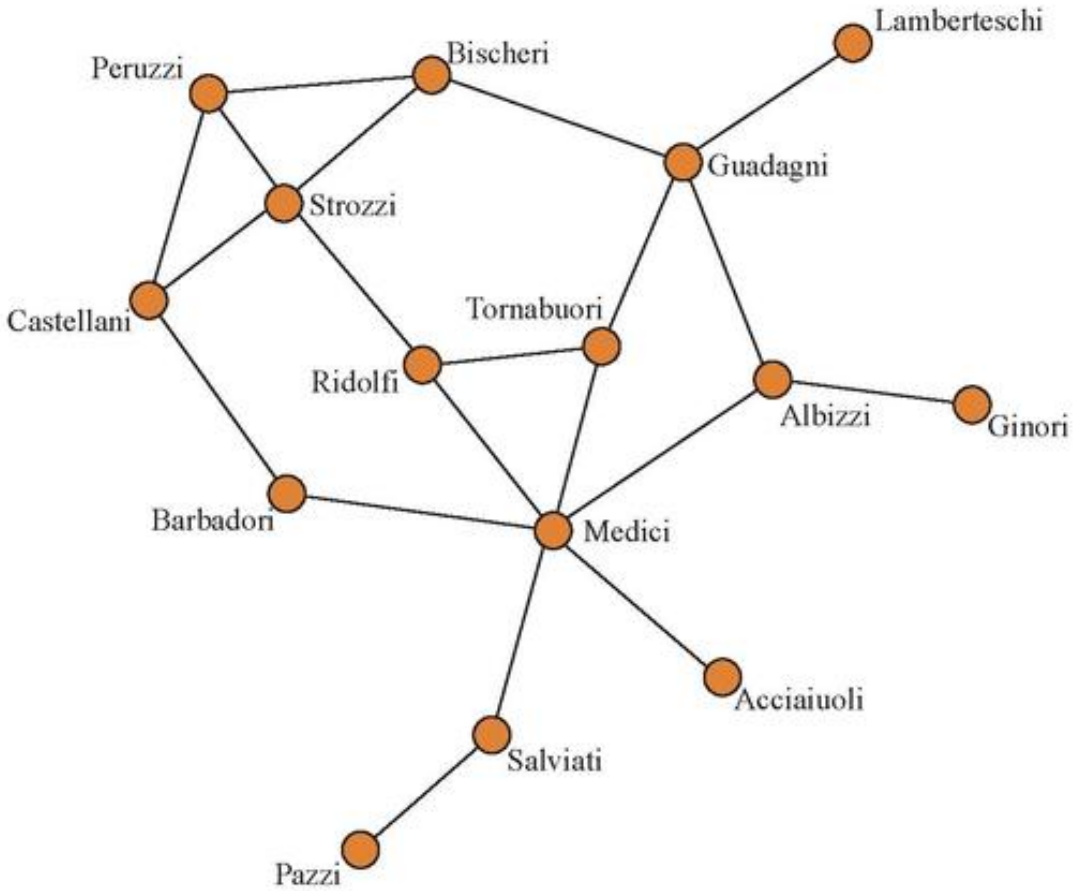
Sylabus z informačného listu

- Základné pojmy z teórie grafov, príklady grafov/sietí, ich vizualizácia
- Miery centrality vrcholov
- Hľadanie komunít v sieti
- Siete založené na koreláciách
- Náhodné grafy a ich vlastnosti
- Základy štatistických modelov

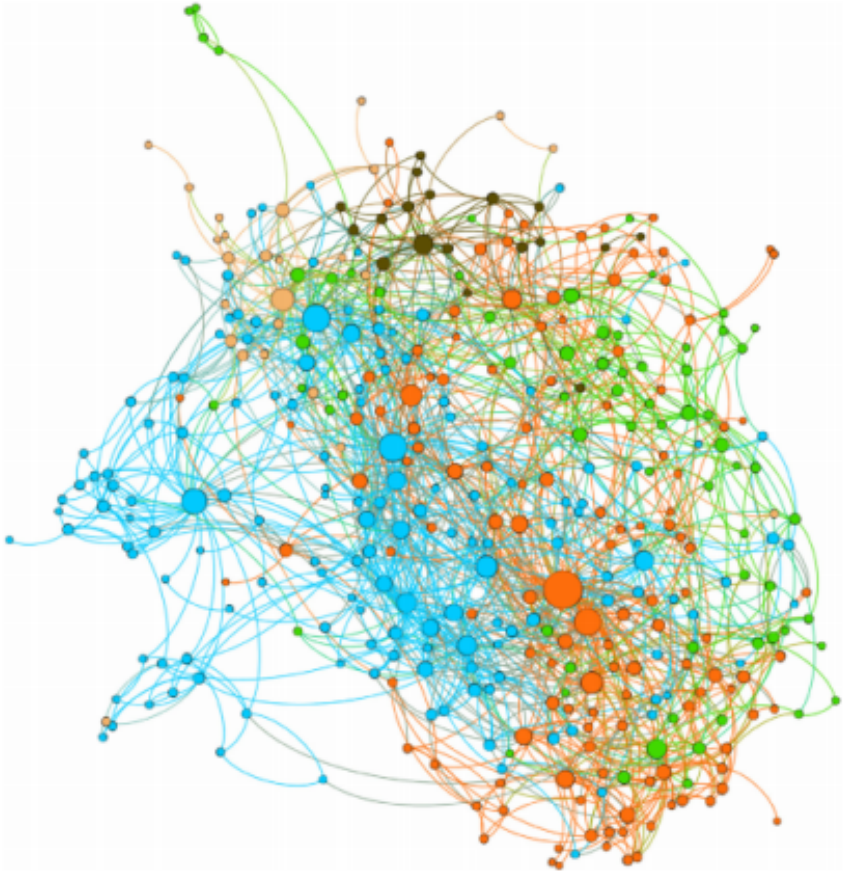
Príklady sietí

Príklad 1: Manželstvá v renesančnej Florencii

Manželstvá medzi významnými rodinami v renesančnej Florencii (15. storočie)



Príklad 2: Zločinecké organizácie

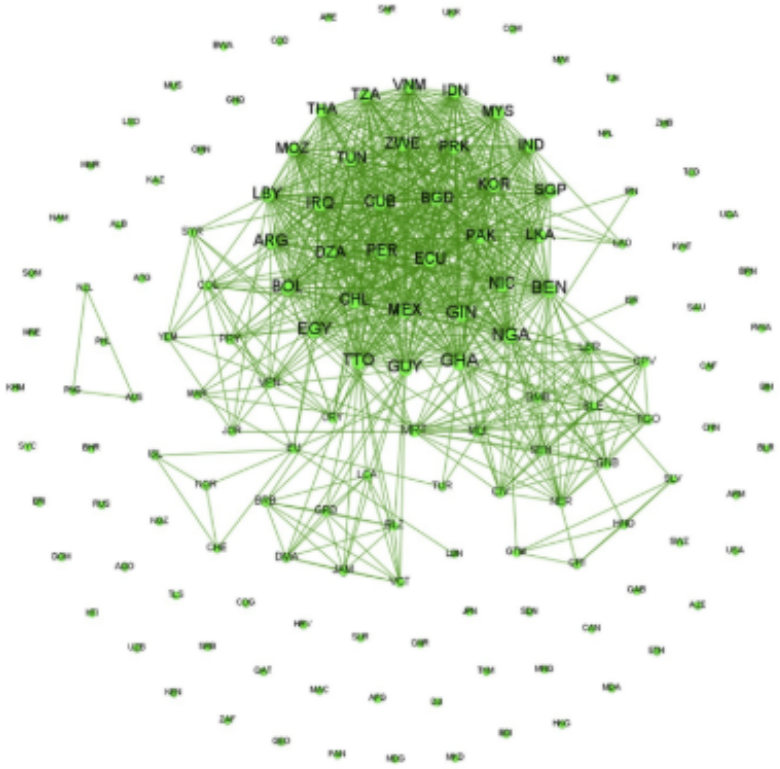


The Five Families of New York City. Note: Nodes are colored according to family membership and sized according to degree.

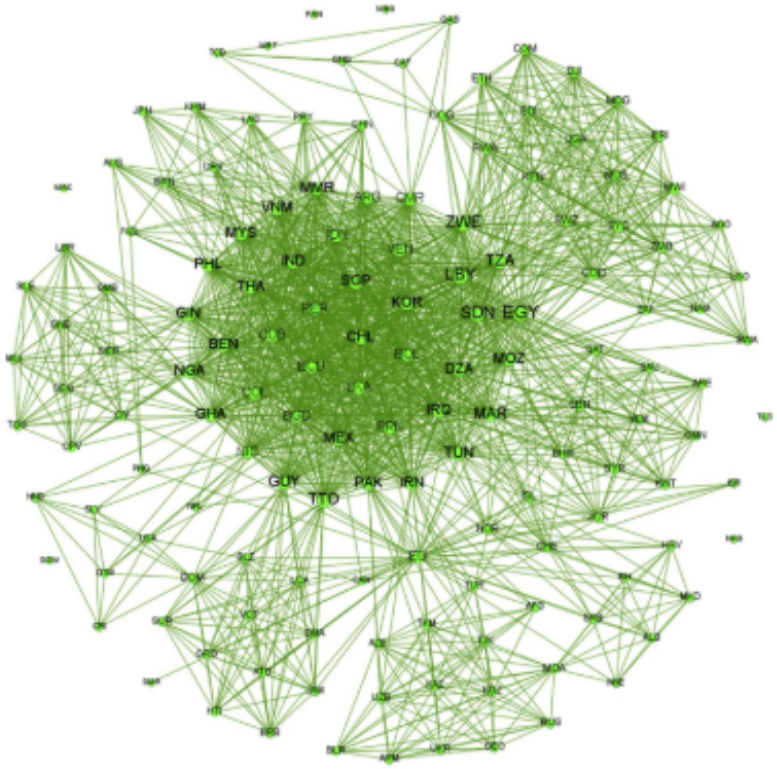
DellaPosta, D. (2017). Network closure and integration in the mid-20th century American mafia. Social Networks, 51, 148-157.

Príklad 3: Zahraničný obchod

Regionálne obchodné dohody



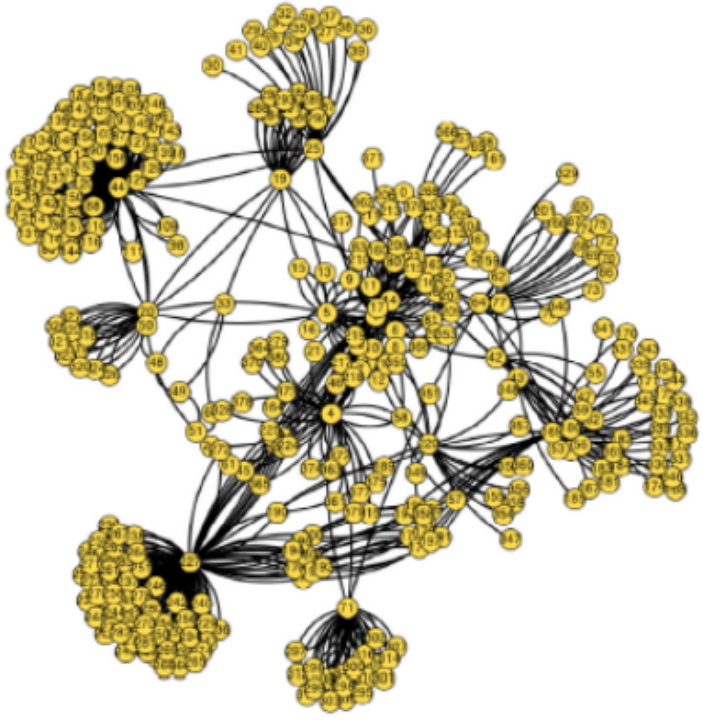
Year 1990



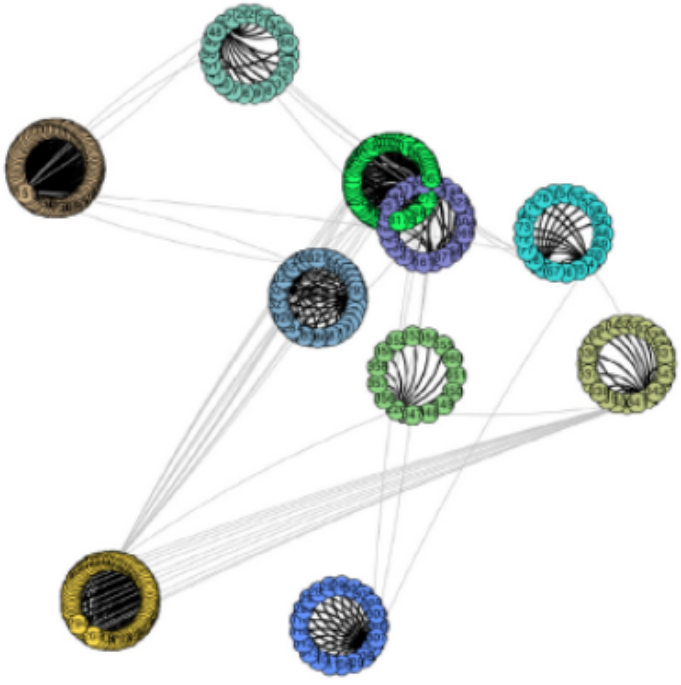
Year 2010

Htwe, N. N., Lim, S., & Kakinaka, M. (2019). The coevolution of trade agreements and investment treaties: Some evidence from network analysis. Social Networks.

Príklad 4: Komunikácia



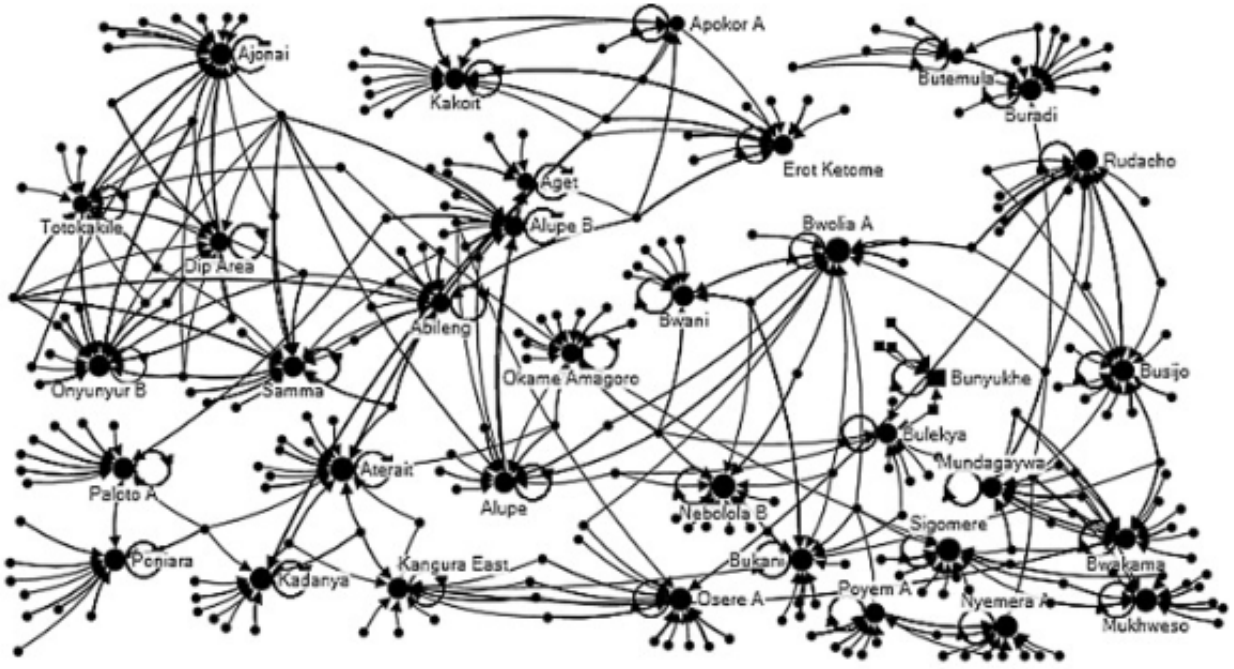
(a) *Phone call network of 148 nodes and 210 edges.*



(b) *Clusters detected after 46 edges deleted.*

Ferrara, E., De Meo, P., Catanese, S., & Fiumara, G. (2014). Detecting criminal organizations in mobile phone networks. Expert Systems with Applications, 41(13), 5733-5750.

Príklad 5: Výskyt chorôb



Network of pig movements between sampled villages ($n = 38$) and other villages through purchase ($n = 1486$ pigs).

Lichoti, J. K., Davies, J., et al. (2016). Social network analysis provides insights into African swine fever epidemiology. Preventive veterinary medicine, 126, 1-10.

Základné pojmy

- Graf, sieť (*graph*, *network*)
- Vrchol (*vertex*, *node*)
- Hrana (*edge*, *tie*)
- Hrany môžu byť
  - orientované/neorientované (*oriented/unoriented*)
  - vážené/nevážené (*weighted/unweigted*)
  - ...
- Vrcholy a hrany môžu mať atribúty (*attributes*)

Práca so sieťami v R-ku: balíky

- Budeme používať balík `igraph`
- Nainštalujte si túto knižnicu

Práca so sieťami v R-ku: príklady sietí, ukážky analýz

Príklad 1: Náhodné grafy Erdösa a Rényiho

Definícia a generovanie náhodného grafu v R

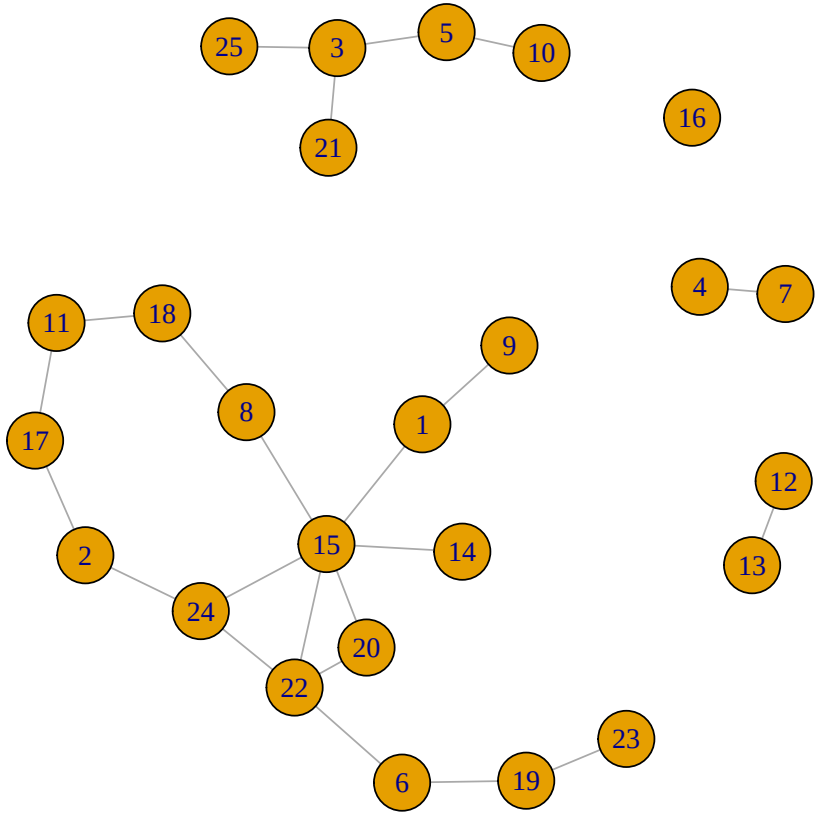
Pametere:

- n = počet vrcholov
- $p \in (0, 1)$ = pravdepodobnosť vzniku hrany
- hrany vznikajú nezávisle na sebe
- v R-ku:
  - knižnica `igraph`
  - funkcia `sample_gnp` - generovaný graf sa označuje ako $G(n, p)$
  - staršia funkcia `erdos.renyi.game` má v názve autorov modelu

```
set.seed(12345) # kvoli reprodukovatelnosti
g <- sample_gnp(n = 25, p = 0.08)
plot(g)
```

Definícia a generovanie náhodného grafu v R

Úprava 1: Chceli by sme mať hrany nakreslené hrubšou čiarou a výraznejšou farbou.



Parametre kreslenia grafu

Základný princíp:

- parametre týkajúce sa hrán majú tvar `edge. ...`, napr. `edge.color =`
- parametre týkajúce sa hrán majú tvar `vertex. ...`, napr. `vertex.size =`

Prehľad:

- https://kateto.net/network-visualization, *Plotting parameters*
- priamo v R-ku pomocou `igraph.plotting`

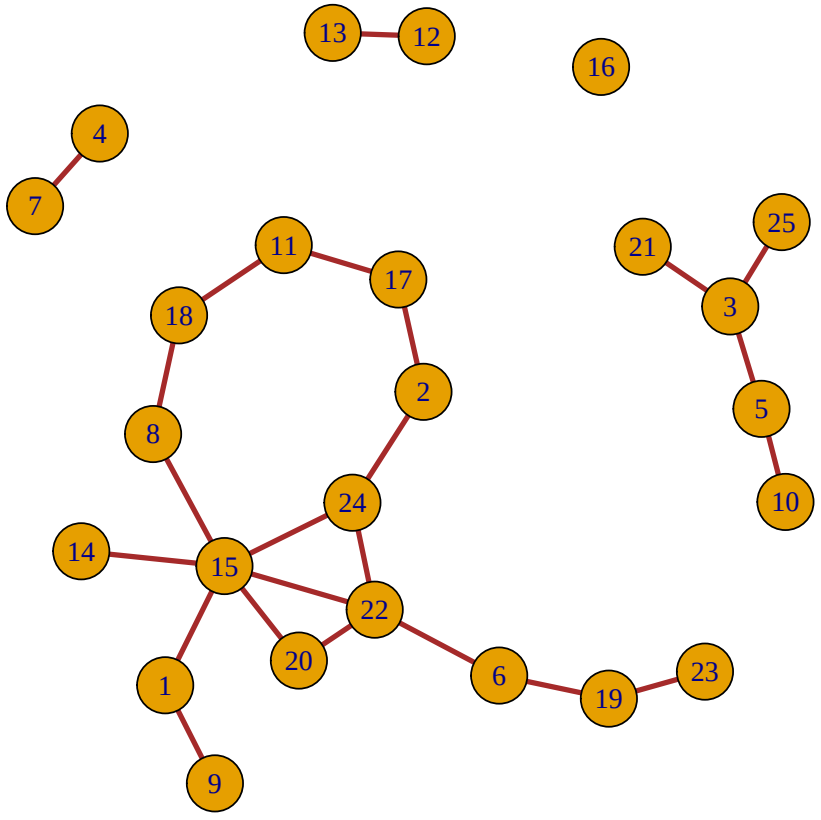
V našom prípade:

- chceme hrubšiu čiaru znázorňujúcu hranu: nastavíme parameter `edge.width` (prednastavená hodnota je 1, vyskúšame vyššie)
- chceme hnedú čiaru: nastavíme `edge.color` na `"brown"`

```
plot(g, edge.width = ..., edge.color = "brown")
```

Parametre kreslenia grafu

Výstup môže byť napríklad takýto:



Parametre kreslenia grafu

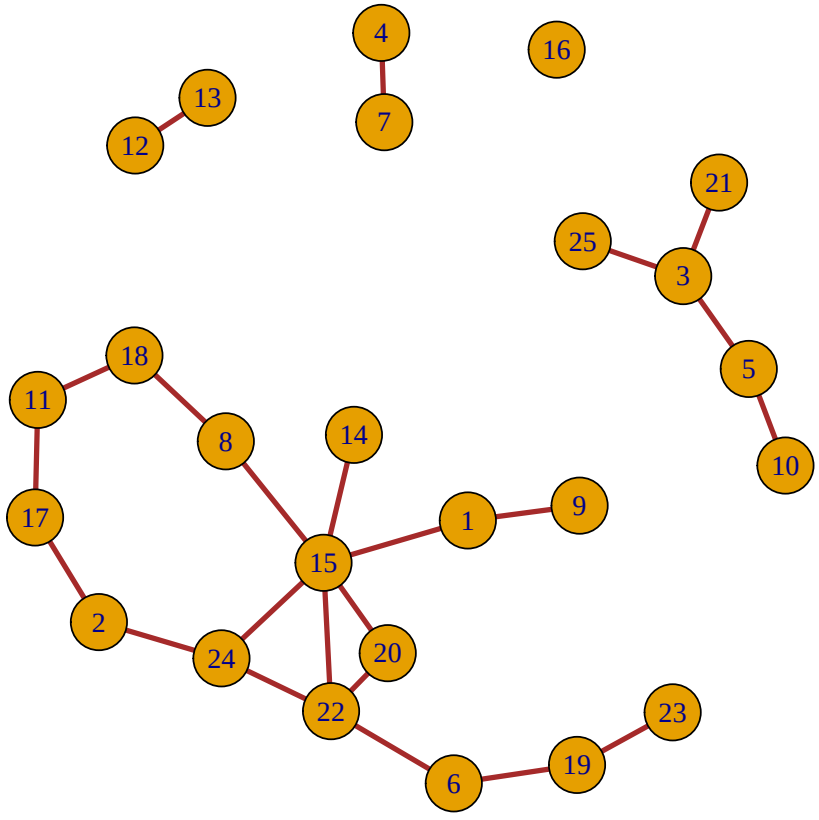
Úpravy 2: Všimnime si, že pri opätovnom kreslení nemusia byť vrcholy rozmiestnené rovnako. Stabilizujme preto rozmiestnenie vrcholov.

- Tento fakt je dôsledkom náhodnosti algoritmu, ktorý počíta polohu vrcholov.
- Zvolíme konkrétny algoritmus a voľbu náhodných čísel pomocou `set.seed`
- Budeme potrebovať parameter `layout`, zvolíme metódu (`layout_with_graphopt`, `...` - oplatí sa vyskúšať ich niekoľko a vybrať tú, pri ktorej sa nám výstup najviac páči) alebo to necháme na R-ko voľbou `layout_nicely`

```
set.seed(123) # kvoli nahodnosti algoritmu
plot(g, edge.width = ..., edge.color = "brown",
     layout = ...
```

Parametre kreslenia grafu

Napríklad:



Parametre kreslenia grafu

Aby sa výpočet polohy vrcholov nemusel opakovať, keď budeme graf kresliť viackrát, rozloženie vrcholov si uložíme:

```
set.seed(1234) # kvoli nahodnosti algoritmu
layout1 = layout_nicely(g)
```

Teraz môžeme spraviť:

```
plot(g,
     ...,  # ostatne parametre kreslenia
     layout = layout1)
```

Parametre kreslenia grafu

Úpravy 3: Vidíme, že graf sa rozpadá na niekoľko súvislých podgrafov, tzv. komponentov súvislosti. Chceli by sme ich farebne odlíšiť.

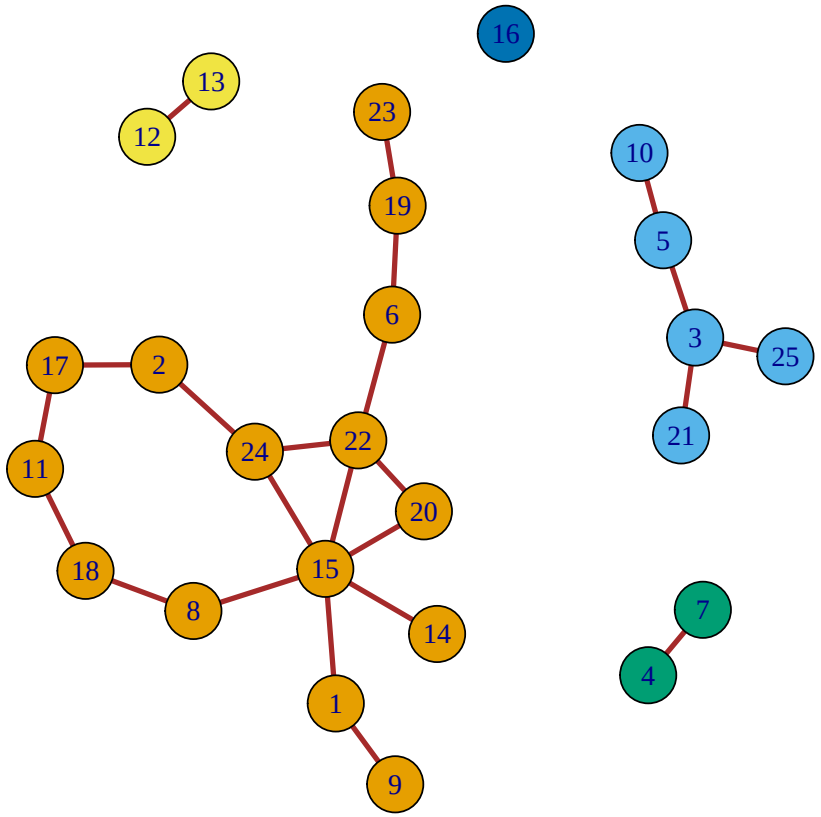
- potrebujeme zmeniť hodnotu `vertex.color`
- užitočná informácia je, že sa pripúšťa aj číselná hodnota, skúste napríklad `vertex.color = 5`
- na určenie komponentov použijeme funkciu `componenets`, ktorá má ako vstupný parameter študovanú sieť

```
components(g)
```

```
## $membership
##  [1] 1 1 2 3 2 1 3 1 1 2 1 4 4 1 1 5 1 1 1 1 2 1 1 1 2
##
## $csize
## [1] 15  5  2  2  1
##
```

Parametre kreslenia grafu

Ako hodnotu `vertex.color` teda môžeme zobrať `components(g)$membership`



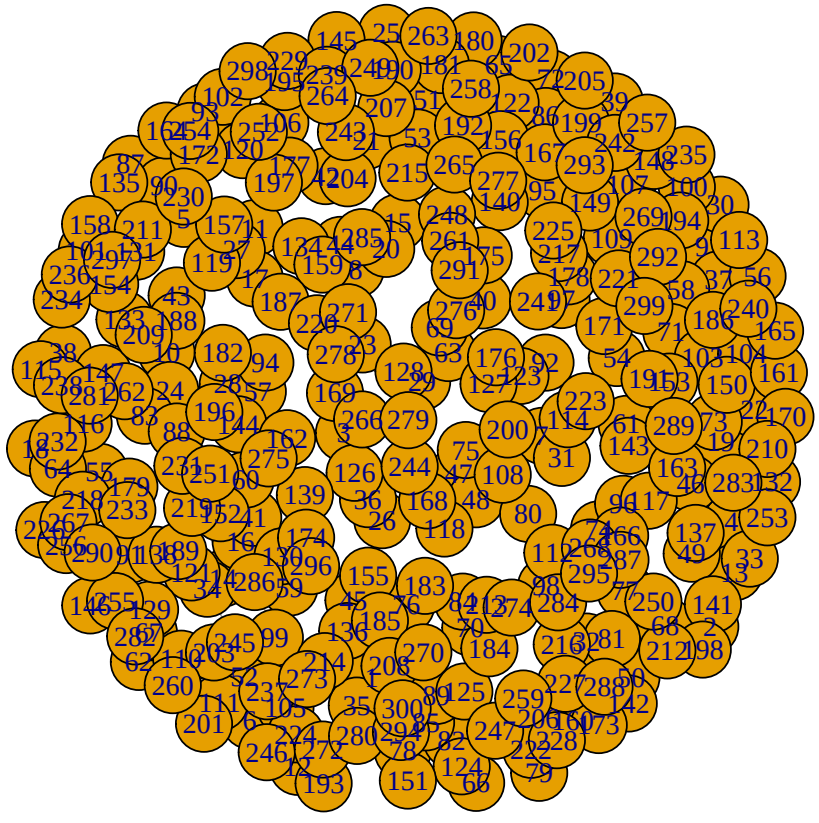
Cvičenie

Pri veľkom počte vrcholov nie je zobrazenie grafu s očíslovanými vrcholmi prehľadné. Zrušte preto pomocou `vertex.label = NA` označenie vrcholov a spravte podľa vlastného uváženia ďalšie úpravy v zobrazení nasledujúcej siete:

```
set.seed(123)
g <- sample_gnp(n = 300, p = 0.005)
plot(g)
```

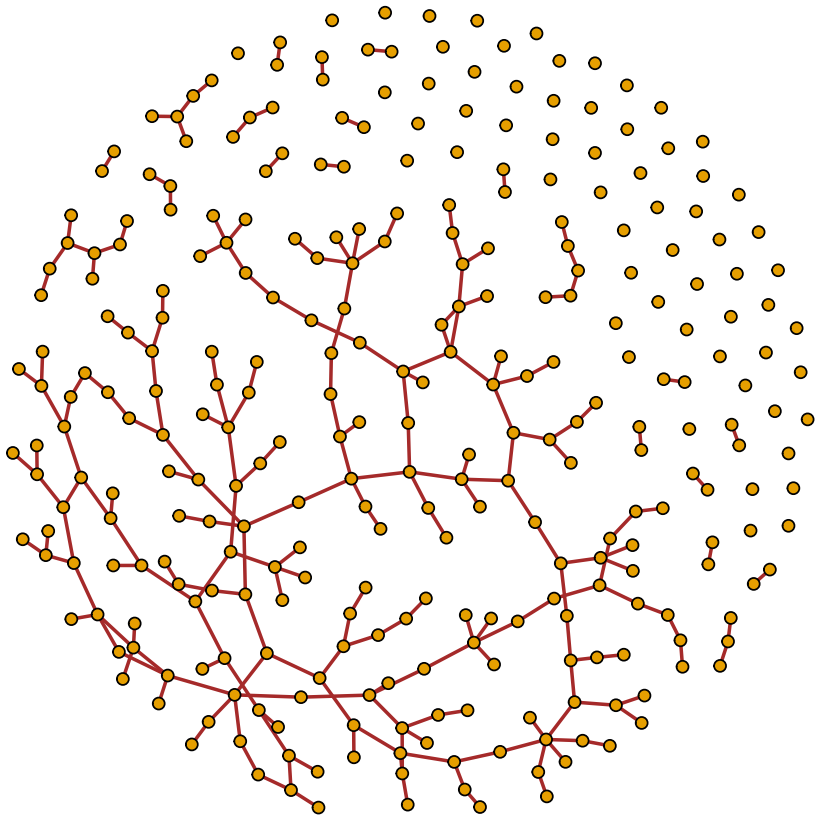
Cvičenie

Pôvodné zobrazenie:



Cvičenie

Ukážka možnej úpravy obrázku (vaša úprava môže byť iná):



Príklad 2: Kradnuté autá

Dáta

Budeme pracovať s dátami zo stránky https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/togo

Základná informácia zo stránky: *Project Togo began in February 1998 when a Toronto-based ringing operation was dismantled and one of its participants informed the police that he was previously employed by a Montreal businessman who was also active in the resale of stolen vehicles. This initial tip was corroborated soon after by a thief who had been arrested while driving a stolen vehicle. By December 1998, the Togo investigation was under way. It spanned into February 1999 and 20 cars that were destined for France, Ghana, and local buyers in southern Quebec were retrieved.*

Dáta

Popis dát zo stránky:

- *1-mode matrix 33 x 33 person by person. Undirected ties.*
- *Ties are communication exchanges between criminals.*
- *Data comes from police wiretapping.*

Jeden z dostupných formátor je CSV, ten vieme načítať do R-ka:

```
togo <- read.csv("TOGO.csv",
                 header = TRUE, # prvy riadok je hlavicka
                 check.names = FALSE, # nazvy stlpcov
                                      # zostanu 1, 2, ...
                                      # inak X1, X2, ...
                 row.names = 1) # nazvy riadkov
                                # su v prvom stlpci
```

Náš cieľ: **spraviť z týchto dát sieť**

Matica susednosti

Matica susednosti (*adjacency matrix*) pre nevážený neorientovaný graf - má v i-tom riadku a j-tom stĺpci

- hodnotu 1, ak sú vrcholy i, j spojené hranou
- inak má hodnotu 0

Pre iné grafy:

- Ak je graf vážený, namiesto hodnoty 1 je v matici váha príslušnej hrany.
- Ak je graf orientovaný, $A_{ij} = 1$, ak existuje hrana z vrcholu i do vrcholu j; analogicky orientované vážené grafy

Vytvorenie siete z matice susednosti

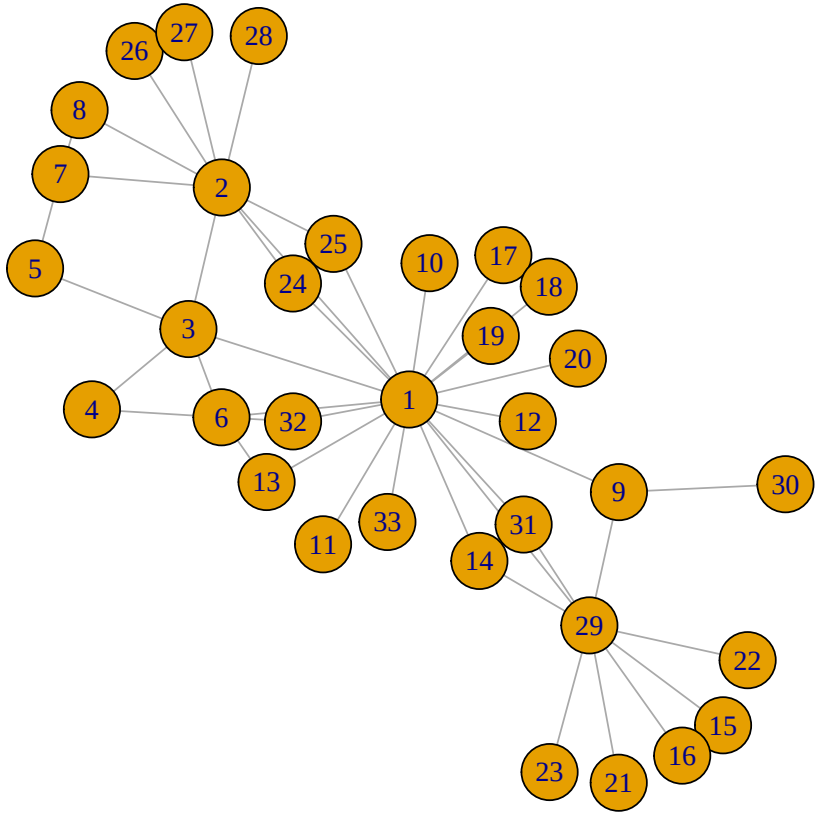
V našom prípade:

- R-ko má funkciu `graph_from_adjacency_matrix`
- z dát uložených v premennej data spravíme maticu
- špecifikujeme, že má vzniknúť neorientovaný nevážený graf
- Mená vrcholov sa automaticky zoberú z mien stĺpcov matice `A`

```
A <- as.matrix(togo)
g_togo <- graph_from_adjacency_matrix(A,
                                      # neorientovany:
                                      mode = "undirected",
                                      # nevazeny:
                                      weighted = NULL
                                      )

plot(g_togo)
```

Vytvorenie siete z matice susednosti



Vytvorenie siete z matice susednosti

```
g_togo
```

```
## IGRAPH 7042eb8 UN-- 33 47 --
## + attr: name (v/c)
## + edges from 7042eb8 (vertex names):
##  [1] 1 --2  1 --3  1 --6  1 --9  1 --10 1 --11 1 --12 1
## [11] 1 --18 1 --19 1 --20 1 --24 1 --25 1 --29 1 --31 1
## [21] 2 --7  2 --8  2 --24 2 --25 2 --26 2 --27 2 --28 3
## [31] 4 --6  5 --7  6 --13 6 --32 7 --8  9 --29 9 --30 14
## [41] 16--29 17--18 21--29 22--29 23--29 26--27 29--31
```

Centralita vrcholov

Aká je centralita (dôležitosť) vrcholov siete? Teda: **Aká je centralita (dôležitosť) ľudí, ktorých predstavujú?**

Rôzne pohľady na to, čo znamená centralita:

- S koľkými vrcholmi je daný vrchol spojený?
- Ako rýchlo sa informácia od neho dostane k ostatným vrcholom (resp. naopak - od ostatných k nemu)?
- Ako často sa vyskytuje v najkratších cestách, ktoré spájajú dva vrcholy?

Teraz len základné myšlienky pre neorientované nevážené grafy, podrobnosti a ďalšie miery centrality neskôr

Centralita stupňa

Stupeň vrchola (*degree*) - počet hrán, ktoré vychádzajú z vrchola (pri orientovaných sa rozlišuje počet hrán, ktoré vchádzajú a ktoré vychádzajú)

Funkcia `degree`:

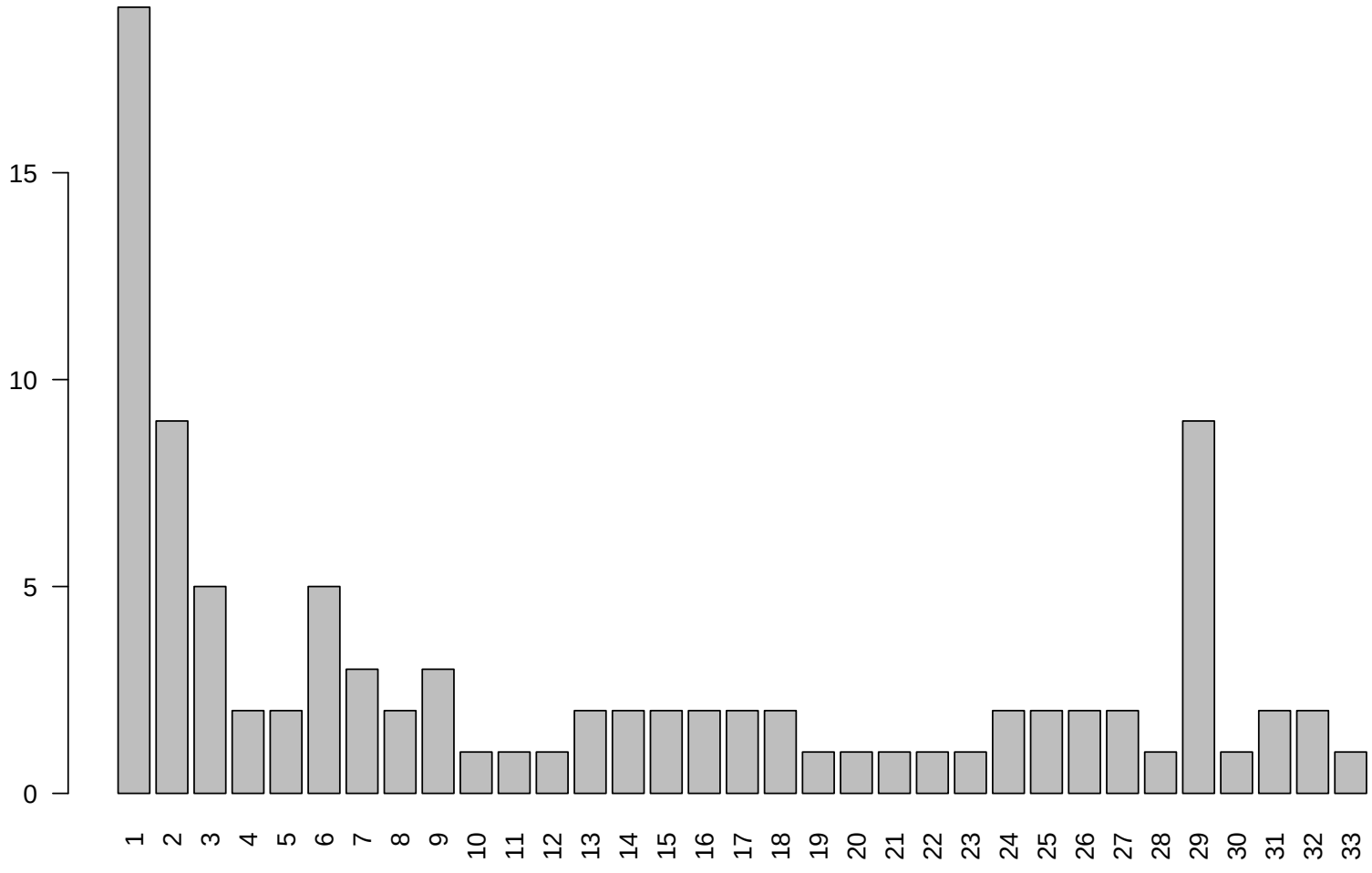
- ako vstup dostane graf
- výstupom je vektor s hodnotami stupňov jednotlivých vrcholov

```
degree(g_togo)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
## 19  9  5  2  2  5  3  2  3  1  1  1  2  2  2  2  2  2  1
## 26 27 28 29 30 31 32 33
##  2  2  1  9  1  2  2  1
```

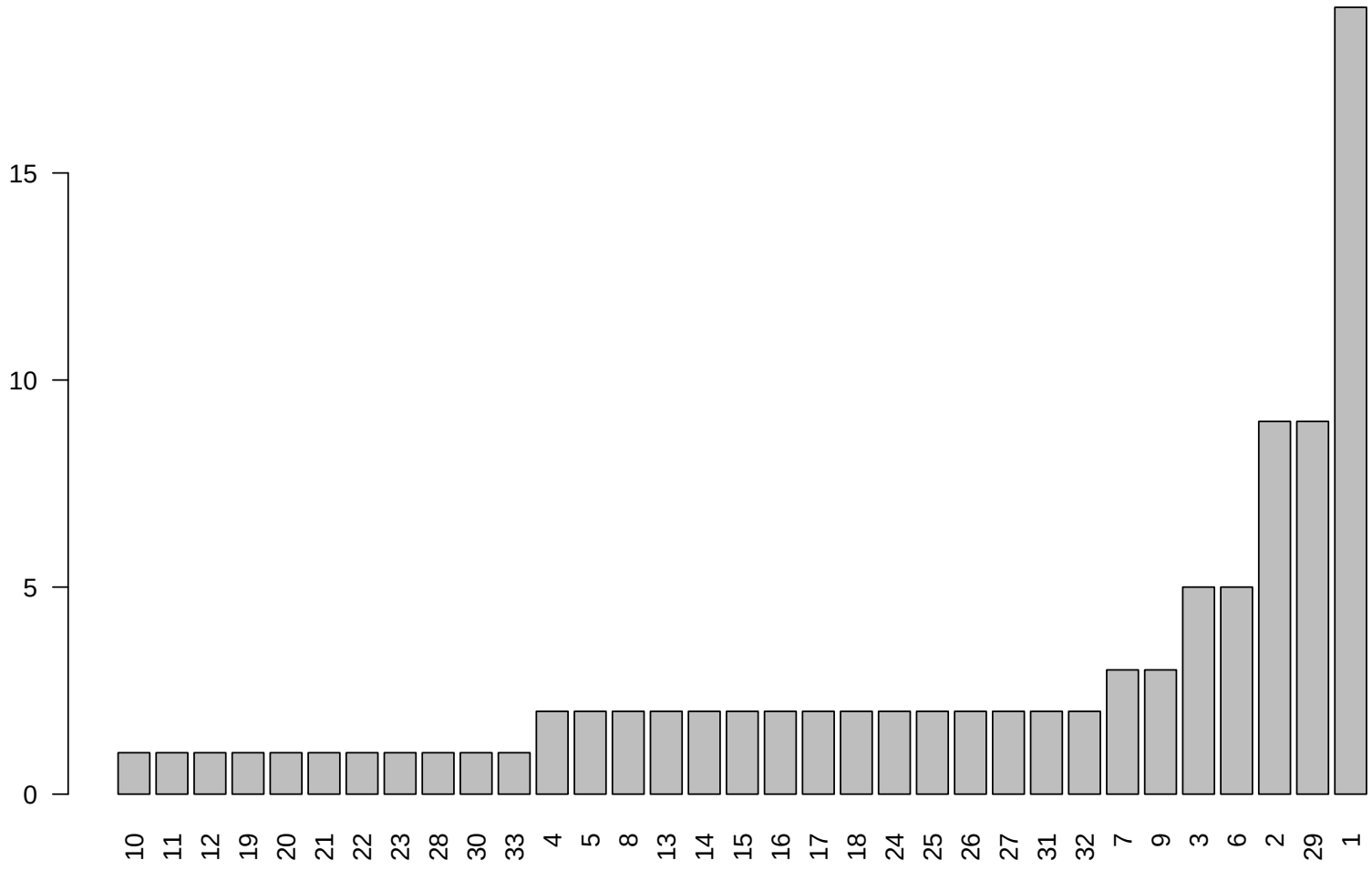
Centralita stupňa

```
barplot(degree(g_togo), las = 2)
```



Centralita stupňa

Usporiadajte vrcholy tak, aby sme ľahko videli, ktoré majú najvyššiu centralitu:



Centralita blízkosti a medzipolohy

Centralita blízkosti (*closeness*)

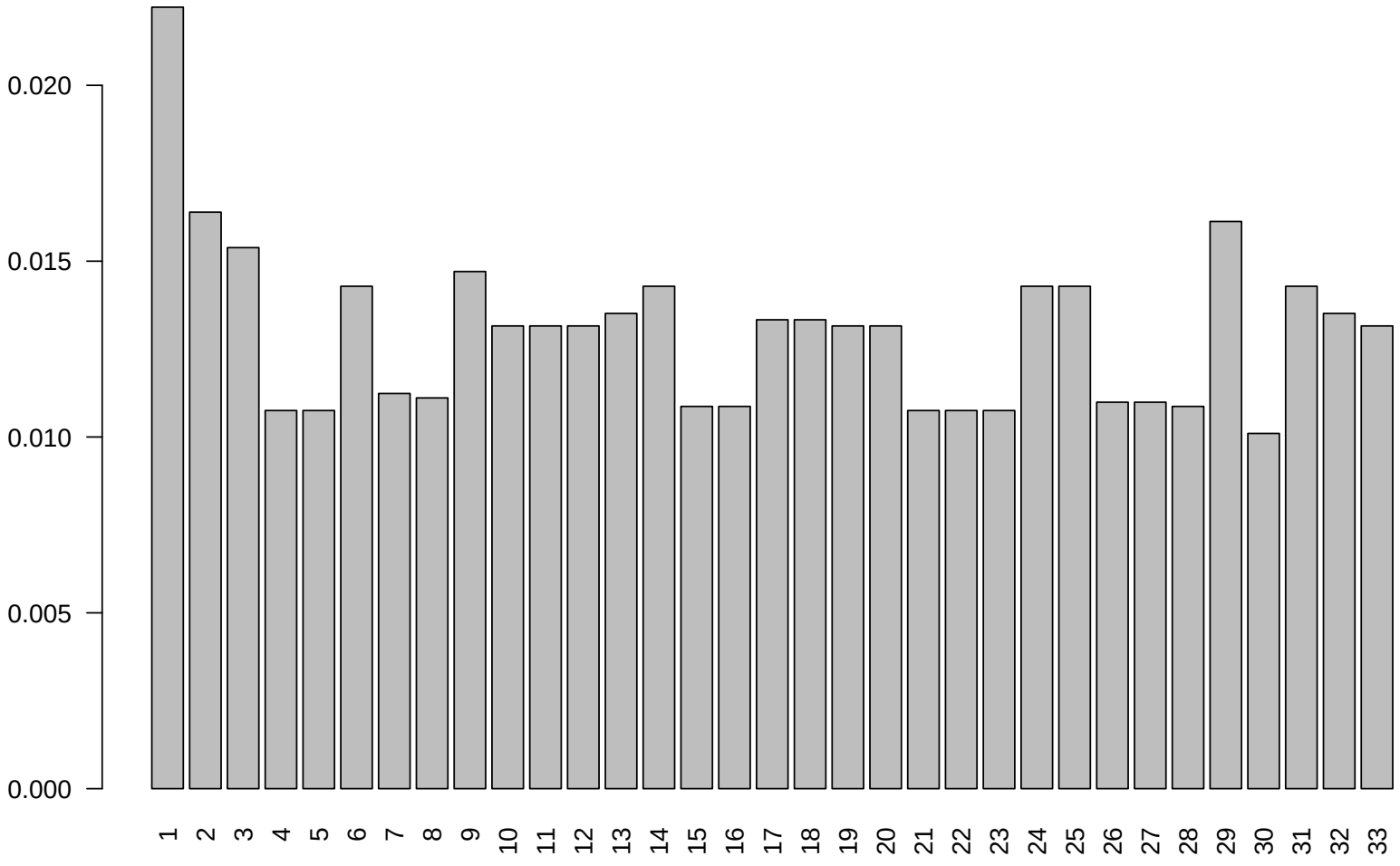
- vzdialenosť vrcholov i a j definujeme ako dĺžku najkratšej cesty (počet hrán v ceste), ktorá ich spája, ozn. $d(i, j)$
- centralita blízkosti vrchola i je nepriamo úmerná $\sum_{j \neq i} d(i, j)$
- v R-ku funkcia `closeness`

Centralita medzipolohy (*betweenness*)

- $P(i, j) =$ počet najkratších ciest medzi i a j
- $P_k(i, j) =$ počet najkratších ciest medzi i a j, ktoré obsahujú vrchol k
- centralita medzipolohy vrchola k je priamo úmerná $\sum_{j \neq i} \frac{P_k(i,j)}{P(i,j)}$
- v R-ku funkcia `betweenness`

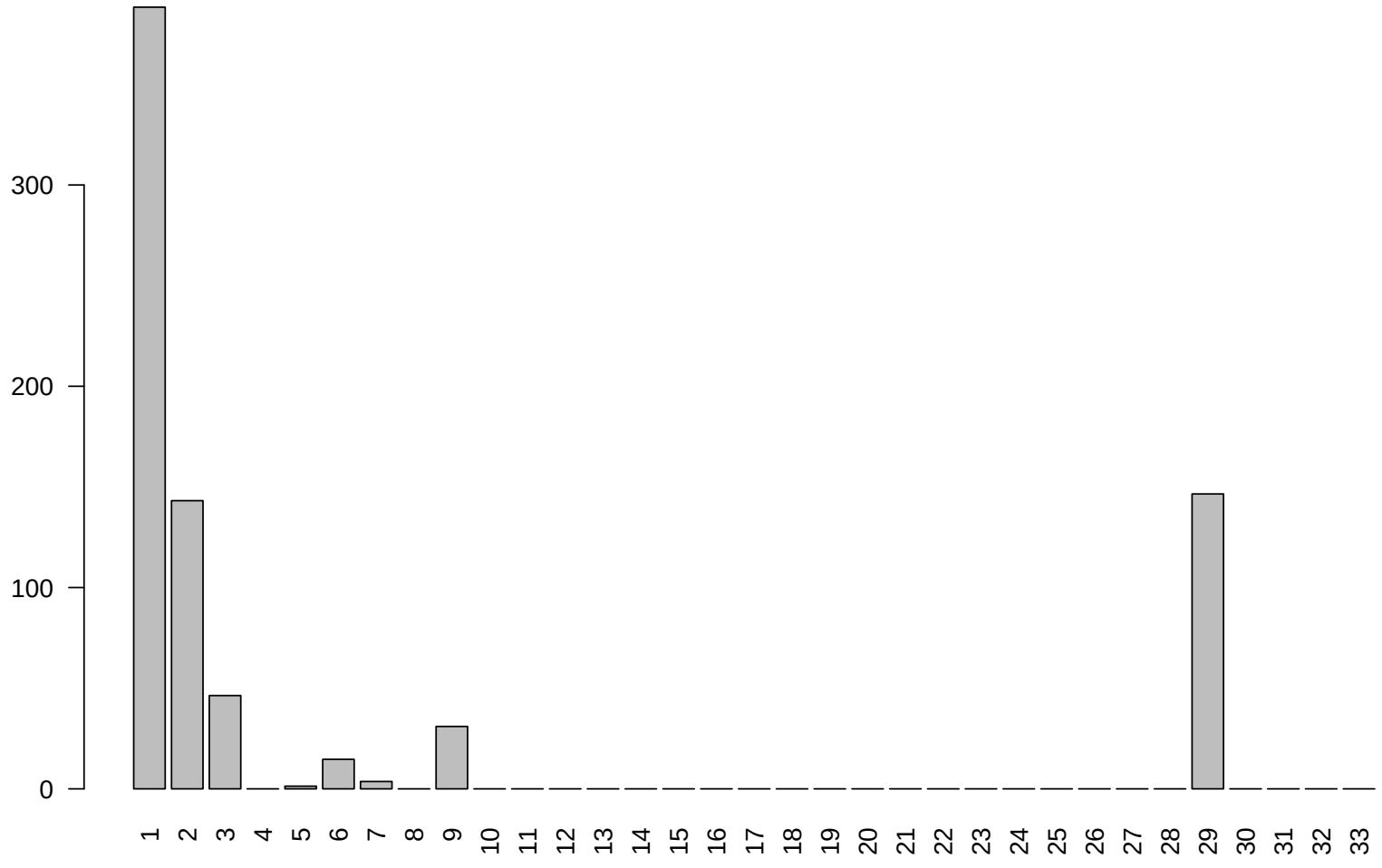
Centralita blízkosti

```
barplot(closeness(g_togo), las = 2)
```



Centralita medzipolohy

```
barplot(betweenness(g_togo), las = 2)
```



Príklad 3: Zacharyho karate klub

Dáta

Pozrieme sa na sieť Zacharyho karate klubu pomocou balíka `igraphdata`. Nainštalujte si ho, potom:

```
data(karate)  # nacitanie dat, t.j. siete
g <- karate   # siet vlozime do premennej `g
```

Dáta

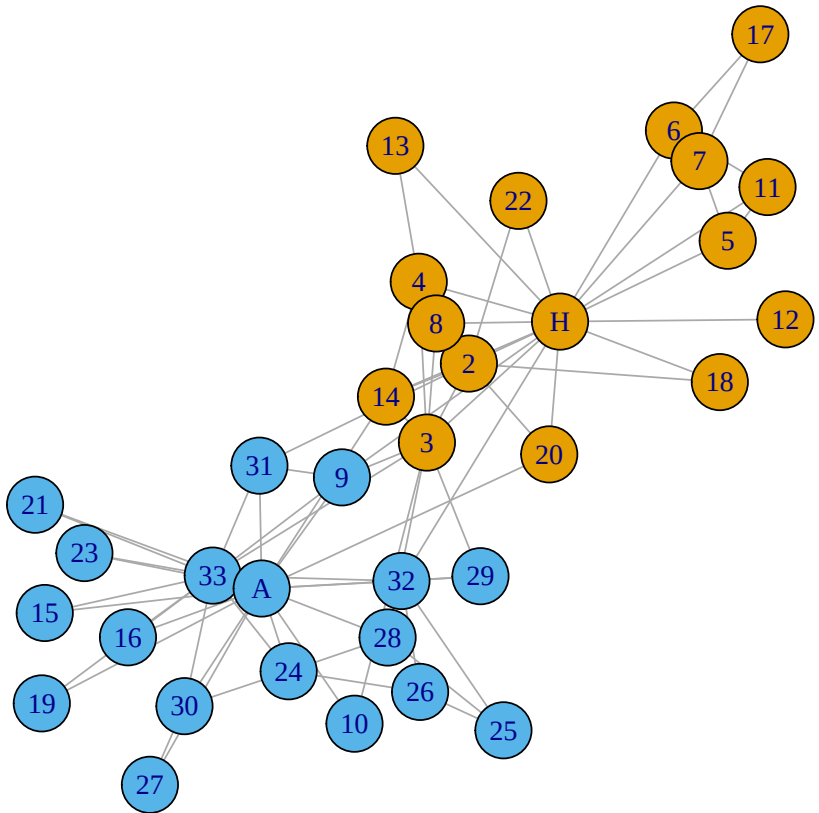
Informáciu o dátach zobrazíme pomocou `?karate`

- *Social network between members of a university karate club, led by president John A. and karate instructor Mr. Hi (pseudonyms).*
- *The edge weights are the number of common activities the club members took part of.*
- *Zachary studied conflict and fission in this network, as the karate club was split into two separate clubs, after long disputes between two factions of the club, one led by John A., the other by Mr. Hi.*
- *The `Faction` vertex attribute gives the faction memberships of the actors*

Grafické zobrazenie

Nakreslíme graf (bez špecifikovania parametrov, použijú sa defaultne alebo už definované v grafe):

```
plot(g)
```



Vrcholy a hrany

Pozrieme sa na vrcholy (*vertices*, preto `V`) a hrany (*edges*, preto `E`) nášho grafu:

```
V(g)
```

```
## + 34/34 vertices, named, from 4b458a1:
##  [1] Mr Hi    Actor 2  Actor 3  Actor 4  Actor 5  Actor
##  [8] Actor 8  Actor 9  Actor 10 Actor 11 Actor 12 Actor
## [15] Actor 15 Actor 16 Actor 17 Actor 18 Actor 19 Actor
## [22] Actor 22 Actor 23 Actor 24 Actor 25 Actor 26 Actor
## [29] Actor 29 Actor 30 Actor 31 Actor 32 Actor 33 John A
```

```
E(g)
```

```
## + 78/78 edges from 4b458a1 (vertex names):
##  [1] Mr Hi  --Actor 2  Mr Hi  --Actor 3  Mr Hi  --Actor
##  [4] Mr Hi  --Actor 5  Mr Hi  --Actor 6  Mr Hi  --Actor
##  [7] Mr Hi  --Actor 8  Mr Hi  --Actor 9  Mr Hi  --Actor
```

Informácia o sieti

```
summary(g)
```

```
## IGRAPH 4b458a1 UNW- 34 78 -- Zachary's karate club netwo
## + attr: name (g/c), Citation (g/c), Author (g/c), Factio
## | name (v/c), label (v/c), color (v/n), weight (e/n)
```

4 znaky charakterizujú graf - v našom prípade **UNW1-*

- **D** - *directed*, **U** - *undirected*
- **N** - *named*, ak majú vrcholy definovaný atribút `name`
- **W** - *weighted*, ak majú hrany definovaný atribút `weight`
- **B** - *bipartite*, vrcholy majú definovaný atribút `type`, ide o tzv. bipartitný graf

Nasleduje počet vrcholov a hrán, názov grafu (ak ho graf má) a informácia o atribútoch

Atribúty

Atribúty - čoho sa týkajú:

- grafu (**g** - *graph*)
- vrcholov (**v** - *vertex*)
- hrán (**e** - *edge*)

a akého sú typu:

- **c** - *character*
- **n** - *numeric*
- **l** - *logical*
- **x** - iné

Napríklad `weight` je atribút hrany (**e**) a je to číslo (**n**).

Pozrite si konkrétne hodnoty atribútov:

```
graph.attributes(g)
```

```
## $name
```

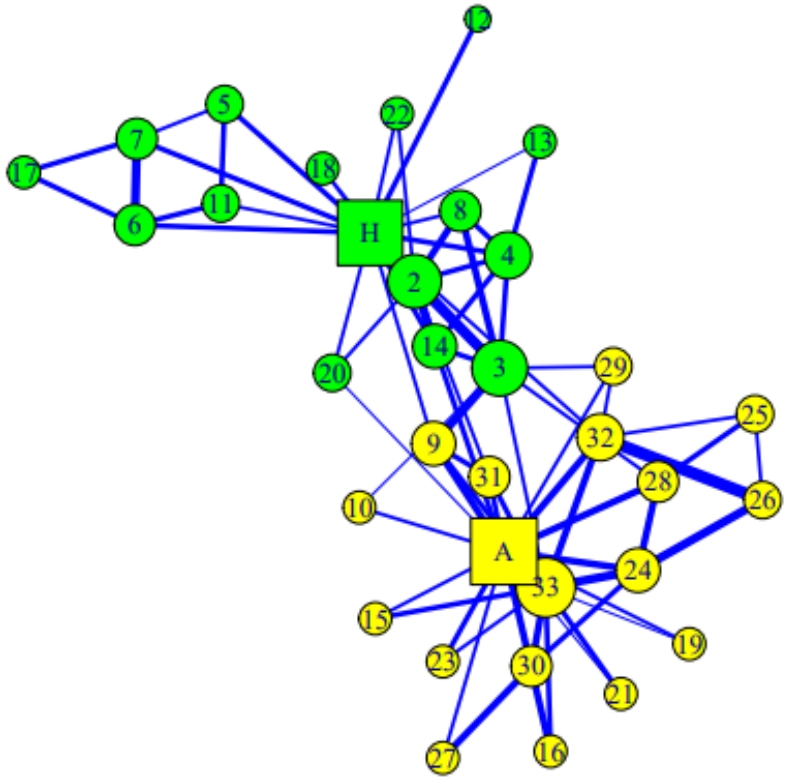
Cvičenie

Upravme obrázok so sieťou nasledovne:

- Hrany budú mať modrú farbu a hrúbka hrán bude úmerná váhe
- Zmeňme farbu vrcholov na zelenú a žltú
- Vrcholy *Mr. Hi* a *John A* budú mať tvar štvorca
- Veľkosť vrchola bude závisieť od počtu hrán, ktoré z neho vychádzajú (viac hrán - väčší vrchol grafu)

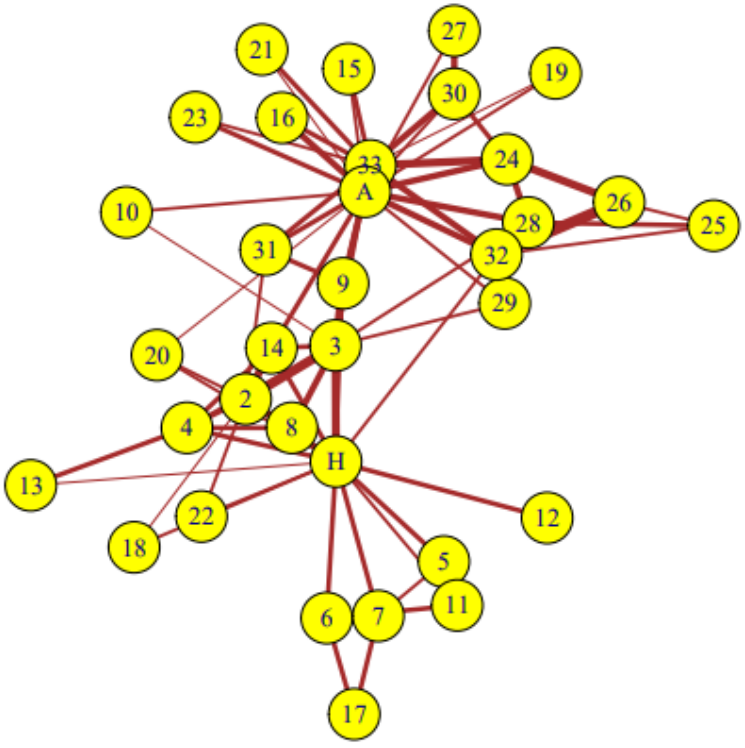
Cvičenie

Ukážka možného výstupu:



Hľadanie komunít (zhlukovanie) v sieťach

- Zobrazme si sieť vzťahov v klube bez informácie atribúte `Faction`, pričom zobrazíme silu kontaktov
- Dalo by sa rozdelenie klubu predpovedať?

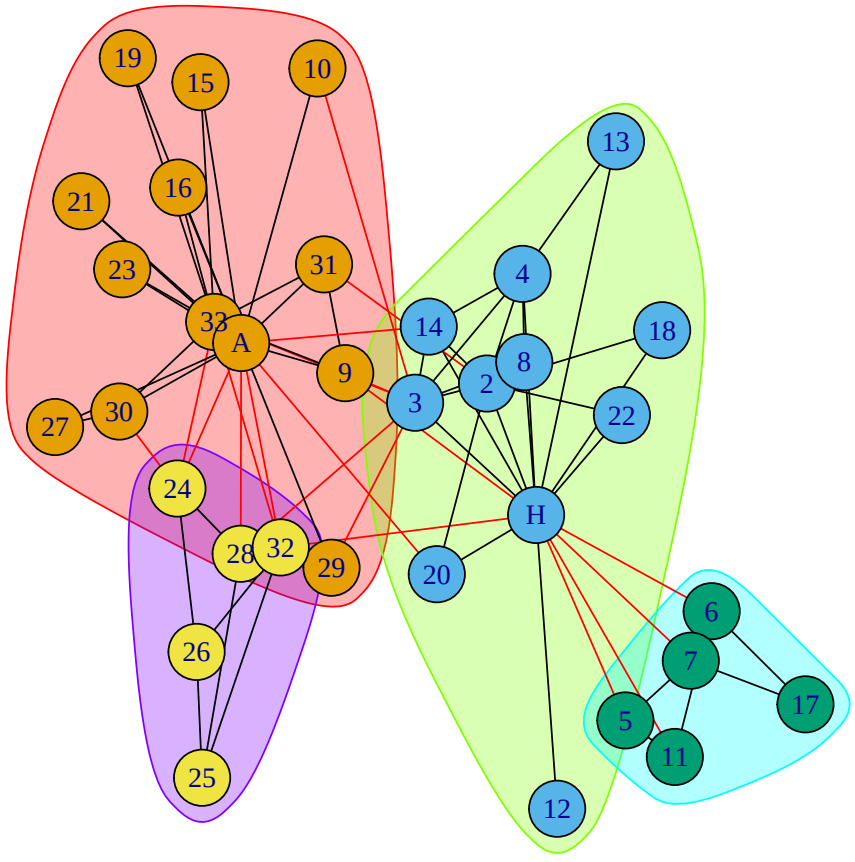


“Walktrap” algoritmus

- Existuje veľa algoritmov na hľadanie komunít, resp. zhlukov v sieťach - budeme sa nimi zaoberať
- Na ukážku: funkcia `cluster_walktrap`
- Základná myšlienka algoritmu: pri krátkej náhodnej prechádzke po hranách grafu sa dá očakávať, že zostaneme v tej istej komunite (v tom istom zhluku)

```
zhlukovanie <- cluster_walktrap(g)
plot(zhlukovanie, g)
```

“Walktrap” algoritmus



"Walktrap" algoritmus: porovnanie s realitou

Porovnajme výsledky zhlukovania s rozpadom klubu.

Budeme potrebovať informáciu o tom, do ktorého zhluku patria jednotlivé vrcholy siete:

```
zhlukovanie$membership
```

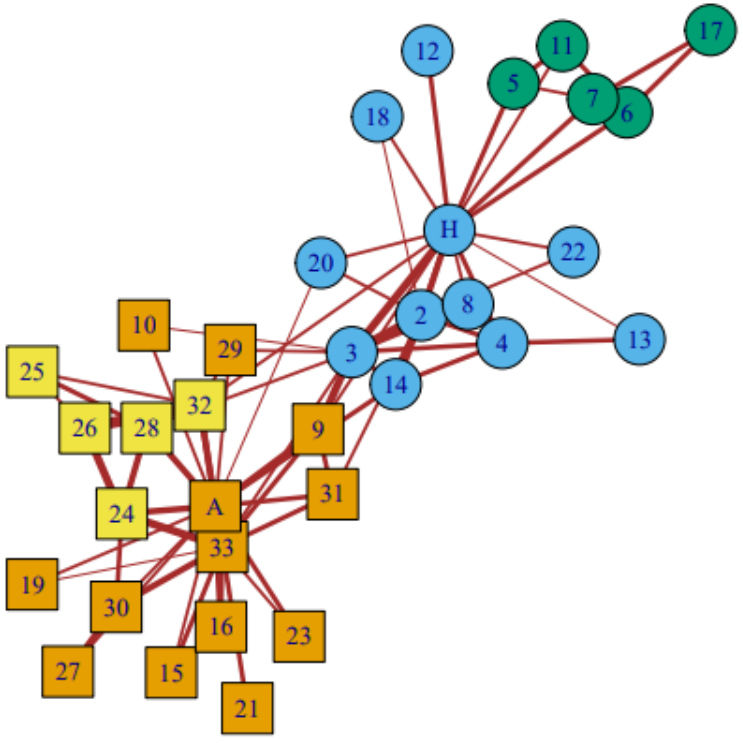
```
##  [1] 2 2 2 2 3 3 3 2 1 1 3 2 2 2 1 1 3 2 1 2 1 2 1 4 4 4
```

Teraz spravíme grafické porovnanie:

- Farbami vrcholov odlíšime jednotlivé zhluky
- Tvarom odlíšime skutočné rozdelenie klubu

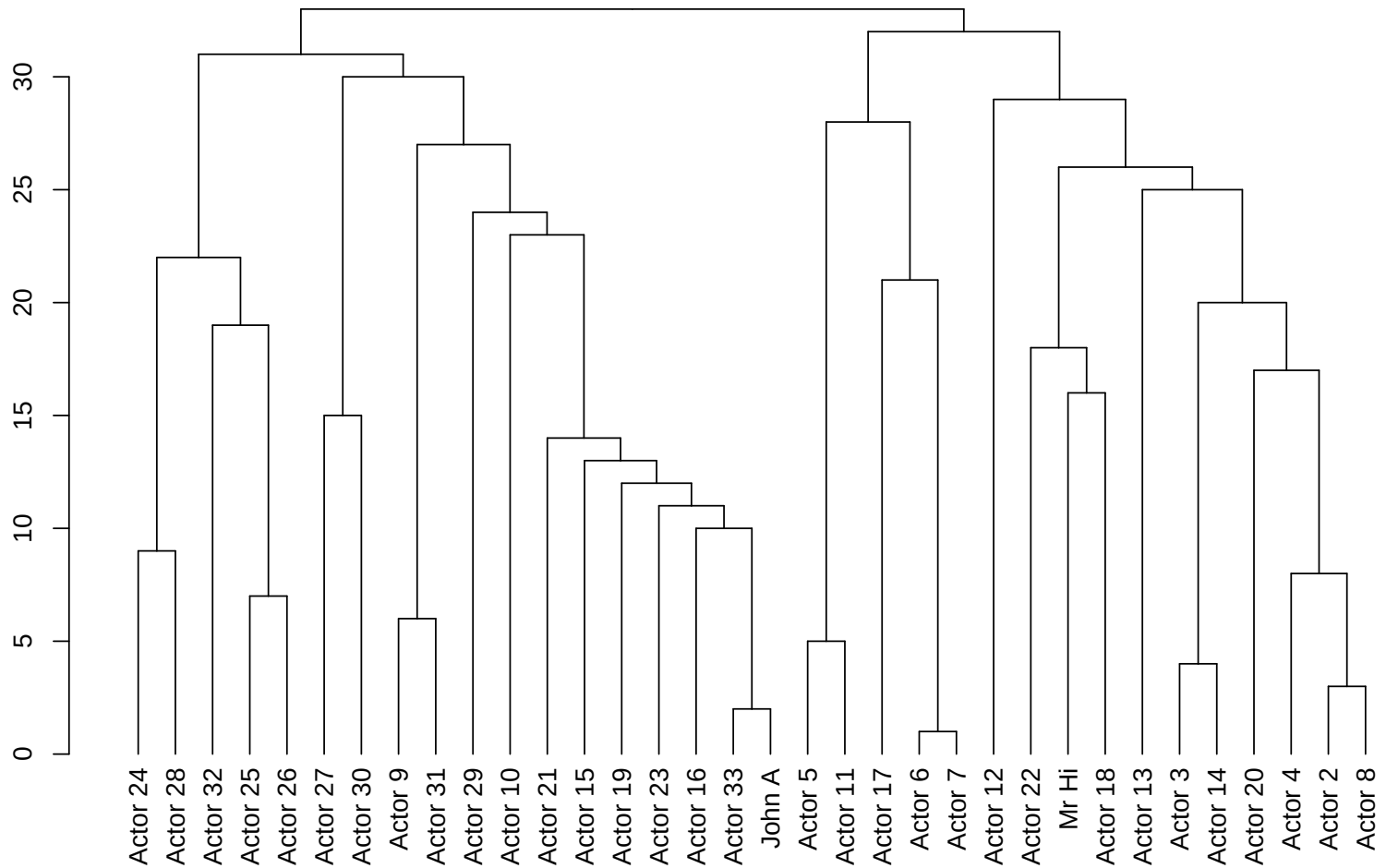
```
tvary <- c("circle", "square")
plot(g,
     vertex.color = ...,
     vertex.shape = ...)
```

“Walktrap” algoritmus: porovnanie s realitou



“Walktrap” algoritmus: vlastný počet zhlukov

```
plot(as.dendrogram(zhlukovanie))
```



"Walktrap" algoritmus: vlastný počet zhlukov

- Algoritmus určil počet zhlukov na zákade určitého kritéria.
- My ale môžeme niektorým algoritmom zadať vlastný počet zhlukov
- Ide o to, kde odrežeme dendrogram
- Funkcia v R-ku: `cut_at`

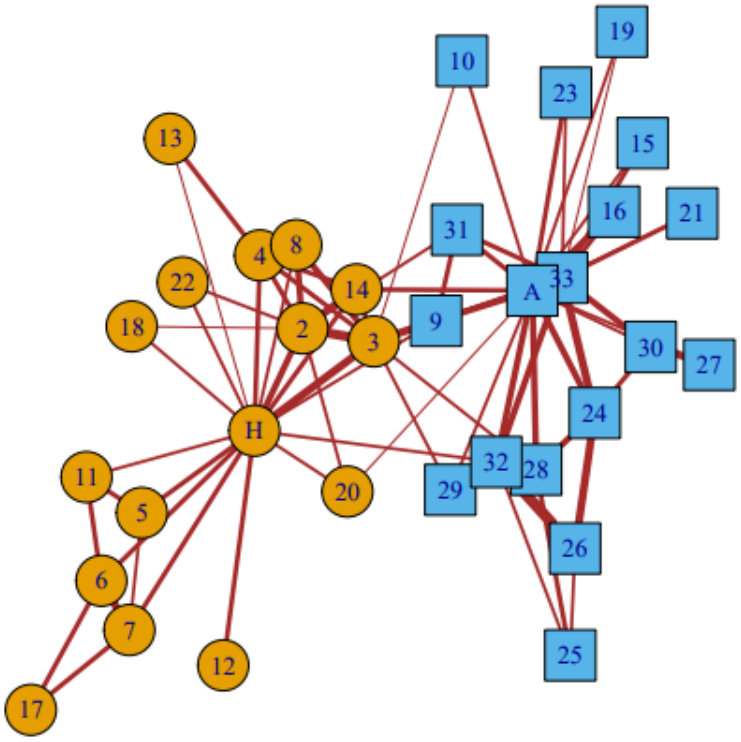
Vytvorme dva zhluky a porovnajme ich s rozdelením klubu:

```
zhlukovanie2 <- cut_at(zhlukovanie, n = 2)
zhlukovanie2
```

```
##  [1] 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1 2 1 2 1 2 2 2 2
```

Spravte teraz grafické porovnanie ako v predchádzajúcom prípade

“Walktrap” algoritmus: vlastný počet zhlukov



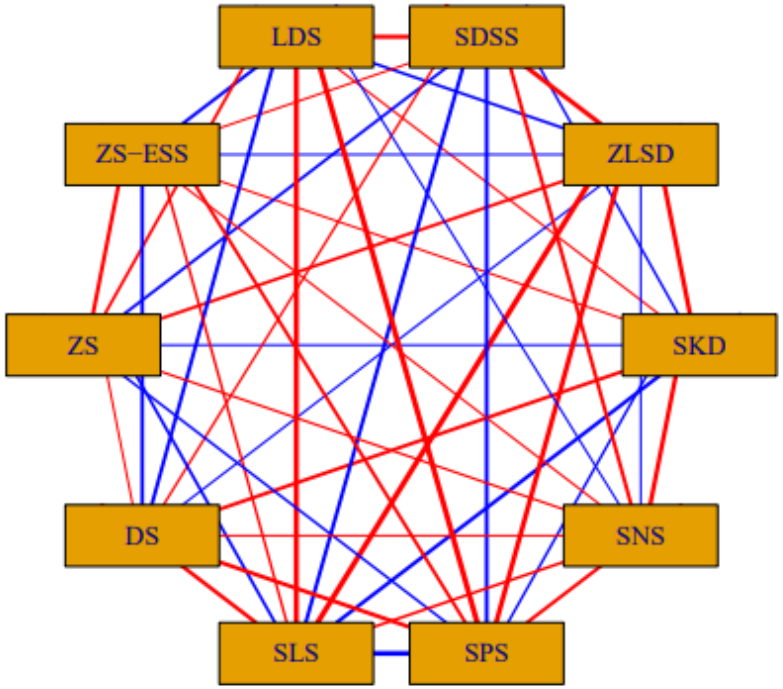
Príklad 4: Politické strany v Slovinsku

Dáta

- Dáta a ich popis na stránke http://vlado.fmf.uni-lj.si/pub/networks/data/soc/Samo/Stranke94.htm
- Vyjadrujú podobnosť politických strán, hodnoty sú priradené na základe dotazníkov
- Váha hrany v sieti je mierou podobnosti strán
- Samostatne zostrojte obrázky na nasledujúcich stranách, resp. spravte vlastnú vizualizáciu tejto siete

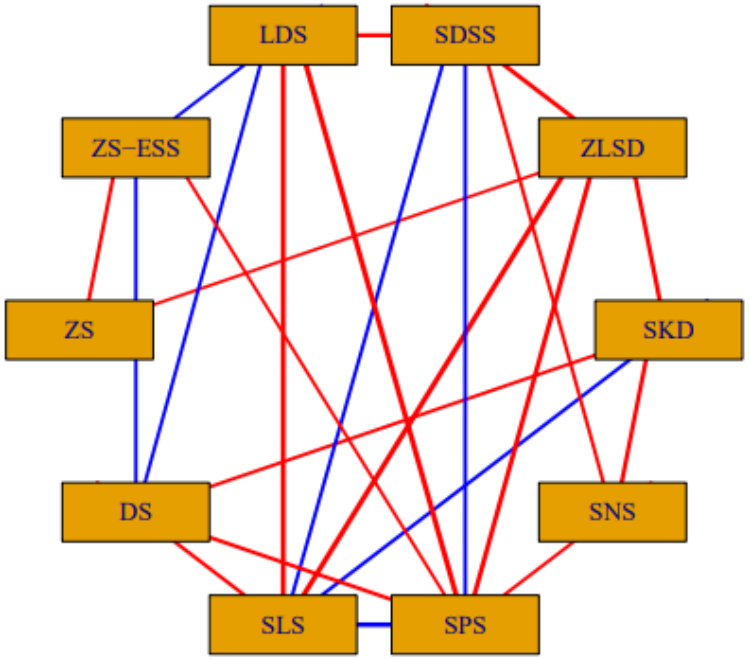
Príklad vizualizácie

Červenou farbou záporné váhy, modrou kladné, hrúbka čiary je úmerná absolútnej hodnote váhy



Príklad vizualizácie

Pre lepšiu prehľadnosť vynecháme v predchádzajúcom grafe hrany s absolútnou hodnotou menšou ako 150



Príklad 5: Futbal

Dáta

Vhodným spôsobom zobrazte sieť (je orientovaná a vážená) danú nasledovnou tabuľkou:

Table 1. Passing pattern of Arsenal against Aston Villa; Saturday, August 19, 2006, Emirates Stadium.

	Fabregas	Silva	Hleb	Toure	Djourou	Henry	Eboue	Hoyte
Fabregas	–	9	24	5	2	12	10	3
Silva	17	–	15	11	5	8	3	11
Hleb	17	8	–	3	1	15	7	–
Toure	8	9	14	–	13	4	10	1
Djourou	5	13	2	17	–	1	–	6
Henry	4	5	10	3	2	–	3	1
Eboue	12	9	7	12	2	2	–	1
Hoyte	12	12	2	–	9	2	3	–

Note: Values indicate the number of passes from row to column player. Only information for the 8 most active players are shown. Ljungberg, Adebayor and Hoyte were substituted. Lehman was the goalkeeper.

Grund, T. U. (2012). Network structure and team performance: The case of English Premier League soccer teams. Social Networks, 34(4), 682-690.

Vizualizácia

Čo treba určite spraviť:

- Pri prvom pohľade na orientovanú sieť vidieť, že treba zmenšiť šípky, ktoré ukazujú orientáciu hrán
- Hrany musia byť oblé, aby sa dali rozlíšiť hrany typu A -> B a B -> A

Ostatné je na vás, chceme, aby bol obrázok pekný, prehľadný a výstižný :)