

Hausaufgabe_1_1

Vorgehensweise

In den folgenden Aufgaben werden Sie Funktionen schreiben. Jede Funktion erhält ein Netzwerk, führt darauf Berechnungen aus und gibt eine Lösung zurück. Um Ihre Funktionen zu testen, können sie die Dateien von der Übungswebsite benutzen und diese in ein Netzwerk umwandeln. Im Abgabesystem werden ihre Funktionen mit einem *neuen Netzwerk* getestet. Das neue Netzwerk ist ähnlich aufgebaut, hat aber eine andere Anzahl an Knoten und Kanten. Fügen Sie also nur die Befehle in ihre Abgabe ein, die in den Aufgaben gefordert sind. Alle anderen Befehle können ihr Ergebnis verfälschen. Achten sie unbedingt darauf, keinen Plot-Befehl einzufügen.

Hintergrund

Lisa möchte die Freundschaftsbeziehungen von Erstklässlern analysieren. Sie hat 4 Netzwerke zur Verfügung: * Ein echtes Freundschafts-Netzwerk * Ein Erdős-Renyi-Netzwerk * Ein Small-World-Netzwerk * Ein Barabasi-Albert-Netzwerk Allerdings weiß Lisa nicht mehr, welche Datei zu welchem Netzwerk gehört. In dieser Aufgabe helfen Sie Lisa dabei, die Dateien dem richtigen Netzwerktyp zuzordnen.

Vorbereitung

Lesen sie die Netzwerke g1, g2, g3 und g4 mit folgenden Befehlen ein:

```
g1=read_graph("g1.txt", format = "edgelist", directed=F)
g2=read_graph("g2.txt", format = "edgelist", directed=F)
g3=read_graph("g3.txt", format = "edgelist", directed=F)
g4=read_graph("g4.txt", format = "edgelist", directed=F)
```

Schauen sie sich die Netzwerke genau an (z.B. mit der Plot-Funktion) und suchen sie nach Unterschieden zwischen den Netzwerken. **Bevor Sie die Abgabe hochladen, müssen Sie z.B. alle Befehle zum Einlesen der Daten und zum Plotten löschen oder auskommentieren (=ein # vor den Befehl schreiben), sonst erhalten sie die Fehlermeldung "Error: cannot open the connection"**

a)

Maximale Punktzahl: 2 Punkte Zuerst möchte Lisa die Transitivität der vier Netzwerke vergleichen.

Ihre Funktion bekommt vier Netzwerke. Berechnen sie für jedes Netzwerk den globalen Clustering-Koeffizienten (=globale Transitivität) und speichern sie das Ergebnis in den Variablen tg1 bis tg4. Geben Sie die Variablen zurück. Ihre Funktion soll folgendes Format erfüllen:

```
aufgabe_a<-function (g1, g2, g3, g4){
  #Hier kommen Ihre Befehle
  #....
  return(c(tg1, tg2, tg3, tg4))
}
```

b)

Maximale Punktzahl: 2 Punkte Als nächstes möchte Lisa überprüfen, wie die Degrees der Knoten in den einzelnen Netzwerken verteilt sind.

Ihre Funktion bekommt vier Netzwerke. Berechnen sie für jedes Netzwerk die degree distribution und speichern sie das Ergebnis in den Variablen dg1 bis dg4. Geben Sie die Variablen in einer Liste zurück. Ihre Funktion soll folgendes Format erfüllen:

```
aufgabe_b<-function (g1,g2,g3,g4){  
  #Hier kommen Ihre Befehle  
  #....  
  return(list(dg1,dg2,dg3,dg4))  
}
```

c)

Maximale Punktzahl: 6 Punkte Lisa weiß aus der Forschung, dass Kinder im Grundschulalter oft geschlechtshomogene Freundschaften bevorzugen. Daher möchte sie die Homophilie der Netzwerke in Bezug auf Geschlecht untersuchen. Sie weiß, dass im echten Netzwerk die Knoten 1-15 weiblich, die Knoten 16-30 männlich sind.

Vergleichen Sie die Homophilie der Netzwerke in Bezug auf Geschlecht. Gehen sie dabei wie folgt vor:

- 1) Schreiben sie eine Funktion homophily. Ihre Funktion bekommt *ein* Netzwerk g, das ein Knotenattribut 'geschlecht' hat. Berechnen sie den Anteil der Kanten zwischen Knoten mit dem gleichen Geschlecht an allen Kanten. Speichern Sie das Ergebnis in der Variable internalprop und geben Sie diese zurück. Ihre Funktion soll folgendes Format erfüllen:

```
homophily<-function (g){  
  #Hier kommen Ihre Befehle  
  #....  
  return (internalprop)  
}
```

- 2) Schreiben sie eine Funktion aufgabe_c. Ihre Funktion bekommt vier Netzwerke. Fügen Sie zu jedem Netzwerk ein Knotenattribut 'geschlecht' hinzu. Das Knotenattribut soll einen Zahlenvektor enthalten, mit 15 Einsen, gefolgt von 15 Zweiern. Berechnen Sie anschließend für jedes Netzwerk die Homophilie, indem sie ihre Funktion homophily verwenden, und speichern sie das Ergebnis in den Variablen hg1 bis hg4. Geben Sie die Variablen zurück. Ihre Funktion soll folgendes Format erfüllen:

```
aufgabe_c<-function (g1,g2,g3,g4){  
  #Hier kommen Ihre Befehle  
  #....  
  return(c(hg1, hg2, hg3, hg4))  
}
```

d)

Maximale Punktzahl: 4 Punkte Zuletzt möchte Lisa wissen, welches der Netzwerke g1, g2, g3 und g4 nun das echte Freundschaftsnetzwerk ist.

Nutzen sie ihre Funktionen, um Unterschiede zwischen den Netzwerken festzustellen. Eventuell ist es auch hilfreich, die Netzwerke zu plotten und die Knoten nach Geschlecht einzufärben.

Ordnen sie anschließend jedes Netzwerk seinem Netzwerktyp zu. Speichern sie hierzu den Namen des Netzwerks (z.B. g1) als character-Wert (=in Anführungszeichen) in den folgenden Variablen:

```
echtes_netzwerk='Hier kommt Ihre Lösung'  
erdos_renyi='Hier kommt Ihre Lösung'  
small_world='Hier kommt Ihre Lösung'  
barabasi_albert='Hier kommt Ihre Lösung'
```

Sie müssen hier keine Funktion schreiben, sondern nur die Variablen speichern.