**Overview over Graph Characteristics**

**http://pablobarbera.com/big-data-upf/html/02b-networks-descriptive-analysis.html**

**NETWORK PROPERTIES**

**Dameter**

INTERPRETATION: diameter ist die Länge des längsten Weges (in Anzahl der Kanten) zwischen zwei Knoten und steht für die Größe eines Netzes

**Edge Density**

INTERPRETATION: die Anzahl der Verbindungen, die ein Agent hat, geteilt durch die Gesamtzahl der möglichen Verbindungen, die ein Agent haben könnte - ein Wert für das Netzwerk

**Reciprocity**

INTERPRETATION: Wahrscheinlichkeit, dass die Knoten eines gerichteten Netzes miteinander verbunden sind.

**Transitivity**

INTERPRETATION: transitivity, auch bekannt als Clustering-Koeffizient, misst die Wahrscheinlichkeit, dass benachbarte Knoten eines Netzes miteinander verbunden sind. Mit anderen Worten: Wenn i mit j und j mit k verbunden ist, wie hoch ist dann die Wahrscheinlichkeit, dass i auch mit k verbunden ist?

**triad\_census (gehört zu transitivity)**

INTERPRETATION: Der Dreiklangszensus von Davis und Leinhardt besteht aus einer Klassifizierung aller gerichteten Dreiklänge in eine von 16 verschiedenen Kategorien; https://www.rdocumentation.org/packages/sna/versions/2.6/topics/triad.census

**NODE PROPERTIES**

**Strength – missing (sort(strength(igraph))**

INTERPRETATION: Die Strength ist ein gewichtetes Maß für den Grad, die die Anzahl der Kanten berücksichtigt, die von einem Knoten zu einem anderen führen. In diesem Netzwerk ist es die Gesamtzahl der Interaktionen jedes Charakters mit jedem anderen.

**Degree**

INTERPRETATION: Das grundlegendste Maß ist der Degree, die Anzahl der benachbarten Kanten zu jedem Knoten. Er wird oft als Maß für den direkten Einfluss angesehen. Im Star Wars-Netzwerk ist es die Anzahl der Charaktere, mit denen jeder Charakter interagiert.

**degree\_distribution**

INTERPRETATION: Die Verteilung der Knotengrade ist eines der wichtigsten Merkmale zur Charakterisierung von Netzwerken. Sie zeigt, wie unterschiedlich die Kanten auf die einzelnen Knoten im Netzwerk verteilt sein können

**Closeness**

INTERPRETATION: Die Closenessmisst, wie viele Schritte erforderlich sind, um von einem bestimmten Knoten aus jeden anderen Knoten zu erreichen. Sie ist ein Maß dafür, wie lange es dauert, bis Informationen ankommen (wer erfährt Neuigkeiten zuerst?). Höhere Werte bedeuten weniger Zentralität.

**Eigenvector centrality – ? nicht sinnvoll (nur für ungerichtete Netzwerke)**

INTERPRETATION: Die Eigenvektor-Zentralität ist ein Maß für die gute Verbindung zu den gut Verbundenen. Erster Eigenvektor der Adjazenzmatrix des Graphen. Funktioniert nur bei ungerichteten Netzen.

**Betweenness**

INTERPRETATION: Betweenness misst das Vermittlungs- oder Gatekeeping-Potenzial. Sie ist (ungefähr) die Anzahl der kürzesten Wege zwischen Knoten, die durch einen bestimmten Knoten führen (wichtigste Figur?)

**HubScore**

#HubScore: The hub scores of the vertices are defined as the principal eigenvector of A\*t(A), where A is the adjacency matrix of the graph.

**Authority score**

INTERPRETATION**: ist ein weiteres Maß für die Zentralität, das ursprünglich auf das Web angewandt wurde. Ein Knoten hat eine hohe Autorität, wenn er von vielen anderen Knoten verlinkt wird, die wiederum viele andere Knoten verlinken.**

**Page rank - missing (sort(page\_rank(igraph)$vector)**

INTERPRETATION**: Page rank** ist ein Näherungswert für die Wahrscheinlichkeit, dass eine Nachricht an einem bestimmten Knotenpunkt ankommt. Dieser Algorithmus wurde von den Google-Gründern entwickelt und ursprünglich auf Website-Links angewandt.

**neighbors**

**INTERPRETATION**: Nicht wirklich ein Maß für die Zentralität, aber wir können mehr darüber erfahren, mit wem jeder Knoten verbunden ist, indem wir die folgenden Funktionen verwenden: neighbors (für direkte Nachbarn) und ego (für Nachbarn, die bis zu n Nachbarn entfernt sind)

neighbors(g, v=which(V(g)$name=="DARTH VADER"))

**NETWORK COMMUNITIES communities (give it a try)**

components(igraph)

**giant component:**

giant <- decompose(igraph)[[1]]

cluster\_infomap(giant)

comm <- cluster\_infomap(giant)

modularity(comm) # modularity score

par(mar=c(0,0,0,0)); plot(comm, giant)

V(giant)$color <- membership(comm)

par(mar=c(0,0,0,0)); plot(giant)