

Pathfinder Networks for Learning Objectives

Robin Gandhi

12/04/2017

Load required packages

Pathfinder requires the *igraph* and *comato* packages. We also load the *dplyr* package to work with dataframes.

```
library("igraph")
library("comato")
library("dplyr")

# Pretty print the tables
library(lemon)
knit_print.data.frame <- lemon_print
```

Create graphs

Load the graphs of all panelists. For each panelist, the graphs connect the first Learning objective required to master for each responsibility area. The responsibility areas are ordered based on rank ordered consensus among the panelists.

Read the graphs for each panelist

```
# set working directory
setwd("~/Box/NSA Core Curriculum")
# Read all files from a directory and import as conceptmaps object.
# Strip = True removes nodes without adjacent edges
cml = read.folder.tgf("LO maps/1stposition", strip = TRUE)

# Extract the individual conceptmaps from cml and create a conceptmap object.
i <- 1
j <- length(cml[[1]][[3]][[i]]$concepts)
l <- list()
while(i<=10) {
  l[[i]] <- cml[[1]][[3]][[i]]
  if (j > length(cml[[1]][[3]][[i]]$concepts))
    j <- length(cml[[1]][[3]][[i]]$concepts)
  i <- i + 1
}
cms = conceptmaps(l)
```

Pathfinder Analysis

Now we take conceptmaps developed by panelist and run pathfinder analysis on them.

Pathfinder networks as described by Schvaneveldt, Durso & Dearholt (1989) are graph based representation of the similarity (or dissimilarity) of entities. Originally, the data that is represented consists of pairwise similarity ratings given by persons, usually using a numeric scale. The similarity ratings can be modeled as a weighted, complete graph with each entity becoming a node and the weight of each edge is the similarity value of the pair of incident entities of that edge. Such a representation is called a “network”. Schvaneveldt et al. (1989, p. 252) note that “[a]s psychological models, networks entail the assumption that concepts and their relations can be represented by a structure consisting of nodes (concepts) and links (relations). Strength of relations are reflected by link weights and the intentional meaning of a concept is determined by its connections to other concepts”. Algorithmic methods can then be used to analyze such a network, or extract prominent features. The Pathfinder algorithm is one such method and an alternative is, for example, multi-dimensional scaling (MDS) developed by Kruskal (cf. Bartholomew, Steele, Moustaki & Galbraith 2008, p. 55ff.).

```
# Create a weighted igraph object using the pathfinder algorithm with q = # of nodes -1 and r = Inf
pathfindernetwork = pathfinder(cms, q=j-1, r=Inf, return.cm=FALSE)
plot(pathfindernetwork)
```

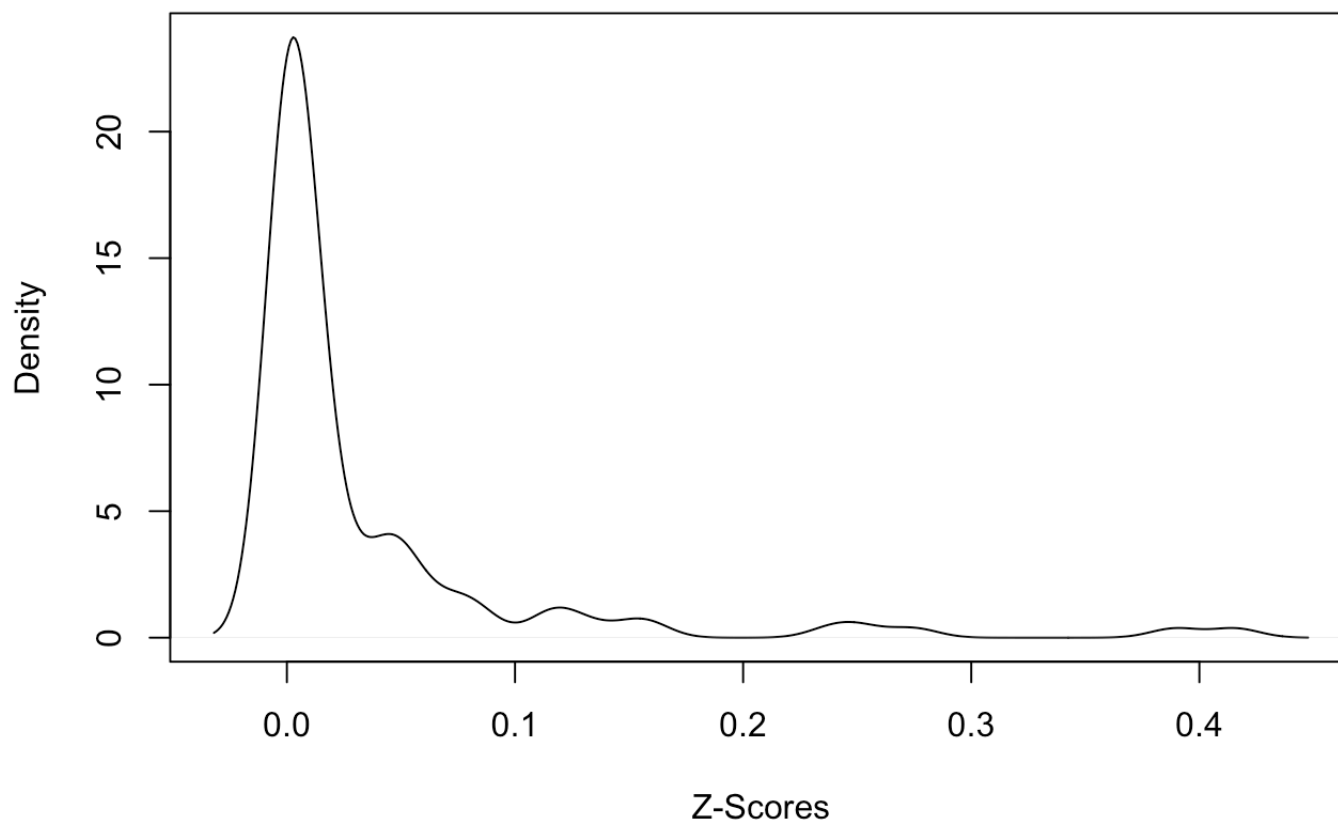


```
## [1] 16 10 16 8 12 16 8 18 18 20 8 8 8 18 4 16 6 4 18 18 20 20 20
## [24] 20 8 20 8 16 18 20 20 20 14 20 20 14 18 20 20 16 18 4 20 16 18 -7
## [47] 10 2 16 12 16 14 20 16 20 18 18 20 20 20 18 20 20 16 20 18 16 18 18
## [70] 20 20 14 14 18 16 16 18 20 12 18 20 20 3 18 16 20 20 18 20 20 18 20
## [93] 20 18 16 20 10 20 20 20 18 20 14 20 20 18 18 20 20 14 20 20 14 20 14
## [116] 18 18 18 18 20 20 18 16 18 18 18 18 20 16 16 18 20 16 16 16 20 20 16
## [139] 18 20 20 20 20 20 20 18 20 20 20 20 9 20 20 20 20 20 20
```

```
# Make sure all edges weights are positive
for (i in 1:length(E(pathfindernetwork)$weight)){
  if (E(pathfindernetwork)$weight[i] < 0)
    E(pathfindernetwork)$weight[i] <- 1
}

# Compute normalized (to make scores compatible, not to be confused with normal distribution) betweenness scores
bvalues <- betweenness(pathfindernetwork, normalized = TRUE)
plot(density(bvalues),type="l",xlab="Z-Scores")
```

density.default(x = bvalues)



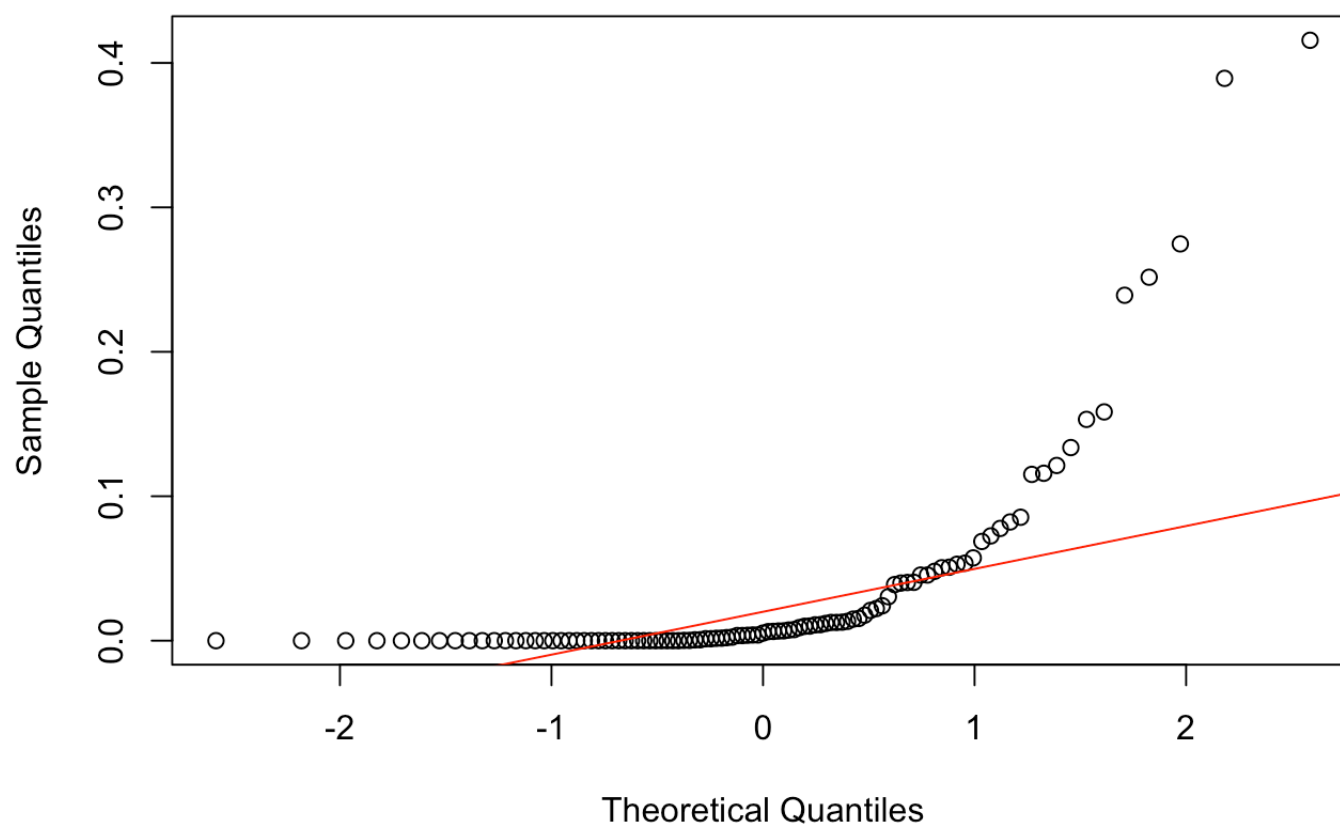
```
## Perform the normality test
shapiro.test(bvalues)
```

```
##
## Shapiro-Wilk normality test
##
## data:  bvalues
## W = 0.53277, p-value < 2.2e-16
```

The null-hypothesis of the Shapiro-Wilk test is that the population is normally distributed. Thus, if the p-value is less than the chosen alpha level, then the null hypothesis is rejected and there is evidence that the data tested are not from a normally distributed population; in other words, the data are not normal.

```
## Plot using a qqplot
qqnorm(bvalues)
# By default qqline draws a line through the first and third quartiles.
qqline(bvalues, col = "red")
```

Normal Q-Q Plot



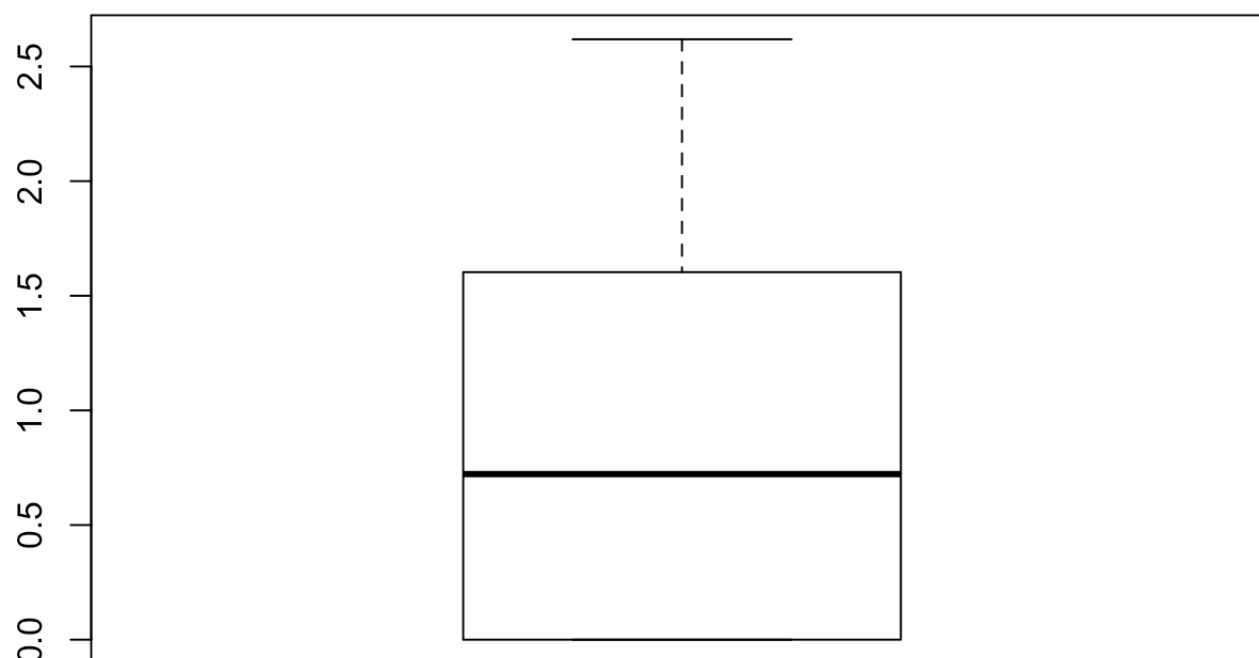
```
write.tgf(conceptmap(pathfindernetwork), "LO maps/1stposition/output/pfnv2.tgf")
```

Normally distributing the betweenness scores for Thresholding

The above graph has a log-normal distribution. We need to obtain a normally distributed graph for significance testing. So we apply a log transformation.

A variable X is lognormally distributed if $Y=\ln(X)$ is normally distributed with “LN” denoting the natural logarithm. Source (<http://www.itl.nist.gov/div898/handbook/eda/section3/eda3669.htm>)

```
# Create a log-normal distribution of the betweenness scores
signedlog10 = function(x) {
  ifelse(abs(x*1000) <= 1, 0, sign(x)*log10(abs(x*1000)))
}
normalbvalues <- signedlog10(bvalues)
boxplot(normalbvalues)
```



```
summary(normalbvalues)
```

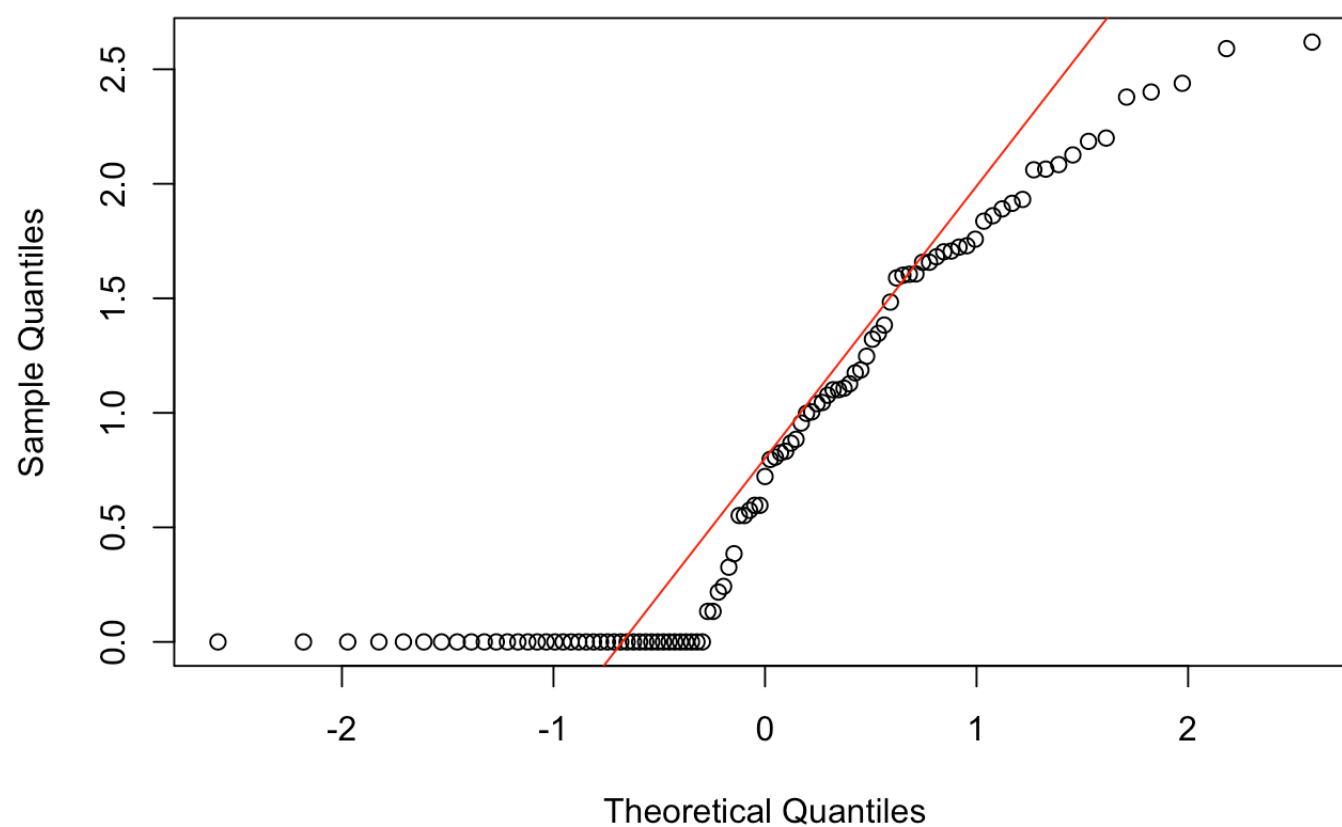
```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000  0.7221  0.8189  1.6030  2.6188
```

```
## Perform the normality test
shapiro.test(normalbvalues)
```

```
##
## Shapiro-Wilk normality test
##
## data:  normalbvalues
## W = 0.85548, p-value = 1.321e-08
```

```
## Plot using a qqplot
qqnorm(normalbvalues)
qqline(normalbvalues, col = "red")
```

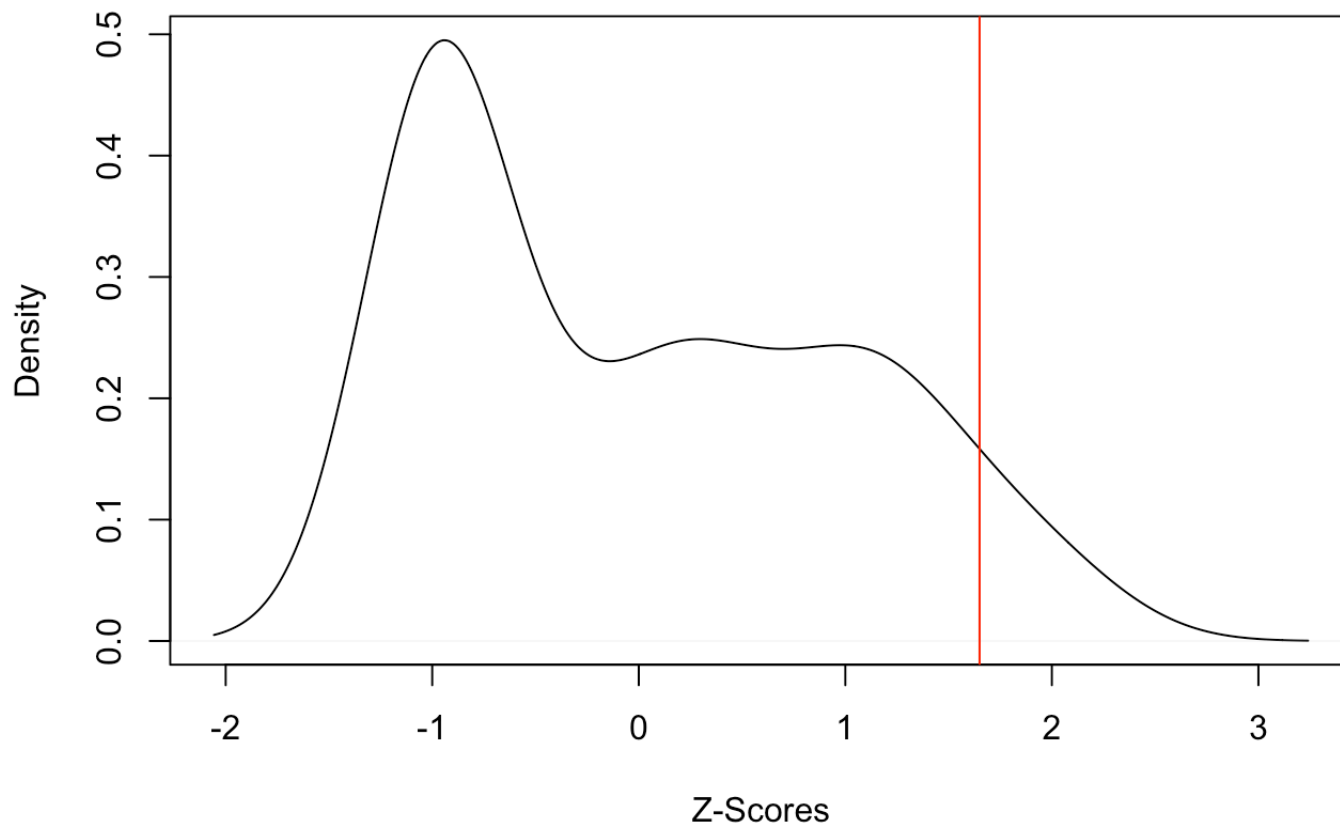
Normal Q-Q Plot



```
# Center and scale the values for Z-scores
normalbvalues <- scale(normalbvalues, center = TRUE, scale = TRUE)

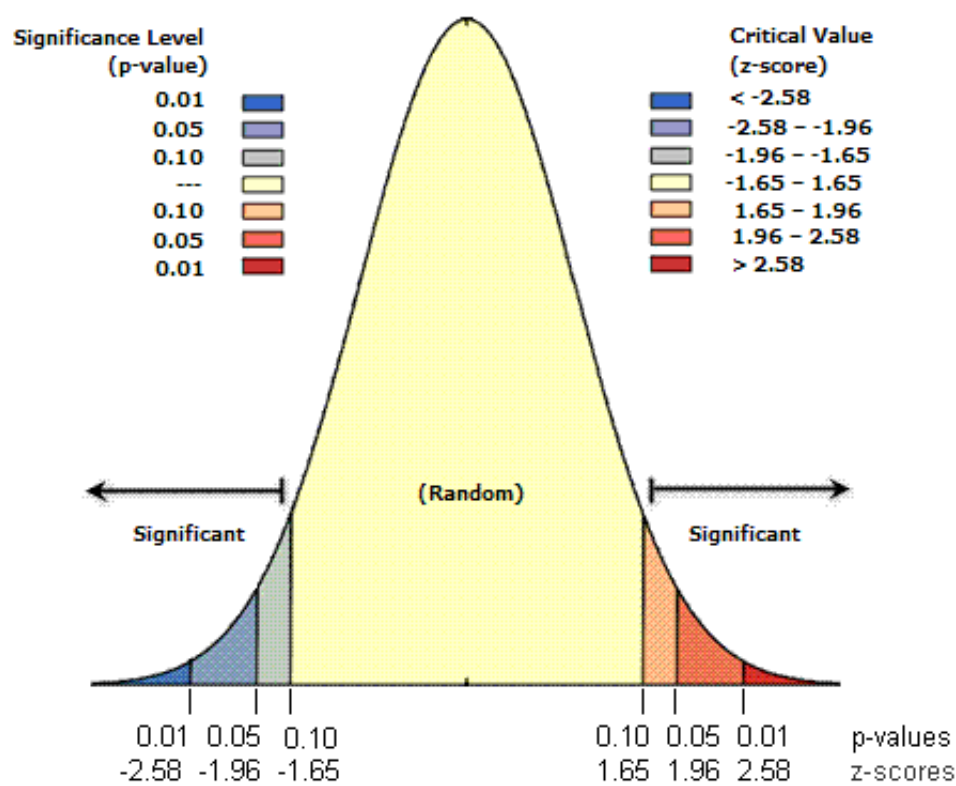
# Plot the distribution
plot(density(normalbvalues), type="l", xlab="Z-Scores")
abline(v = 1.65, col = "red")
```

density.default(x = normalbvalues)



Threshold

If the data is normally distributed then by selecting nodes with a high Z-score, we can separate the nodes that are **significantly** different than the other nodes. Based on the image below, let us select the Z-score cutoff of **1.65** to pick the most significant nodes in the network.



Z-scores and p-values

Concepts sorted based on Betweenness Centrality Scores

```
nbDM = as.data.frame(normalbvalues)
# Add the row names as a column
nbDM["LO"] <- rownames(nbDM)
# Reorder the columns
nbDM <- nbDM[,c(2,1)]
nbDM[order(-nbDM[,2]), ]
```

	LO	V1
LO 7682	LO 7682	2.1718730
LO 1741	LO 1741	2.1375259
LO 879	LO 879	1.9548216
LO 1351	LO 1351	1.9088274
LO 1748	LO 1748	1.8821489

LO 1468	LO 1468	1.6662114
LO 1738	LO 1738	1.6487956
LO 1597	LO 1597	1.5774083
LO 1618	LO 1618	1.5264319
LO 1655	LO 1655	1.5026102
LO 1739	LO 1739	1.4990286
LO 1750	LO 1750	1.3428378
LO 908	LO 908	1.3224029
LO 1533	LO 1533	1.2930624
LO 912	LO 912	1.2570027
LO 1609	LO 1609	1.2284215
LO 1591	LO 1591	1.1334172
LO 930	LO 930	1.0988965
LO 1381	LO 1381	1.0915442
LO 1592	LO 1592	1.0709518
LO 884	LO 884	1.0671375
LO 1548	LO 1548	1.0407621
LO 1466	LO 1466	1.0120941
LO 1363	LO 1363	1.0117209
LO 904	LO 904	0.9504575
LO 1581	LO 1581	0.9487345
LO 878	LO 878	0.9436589
LO 1628	LO 1628	0.9294419
LO 1541	LO 1541	0.8020257
LO 872	LO 872	0.6810322
LO 1626	LO 1626	0.6371523
LO 882	LO 882	0.6065245
LO 1710	LO 1710	0.5167682
LO 1586	LO 1586	0.4448668
LO 1354	LO 1354	0.4292221
LO 1360	LO 1360	0.3717335
LO 1580	LO 1580	0.3484382
LO 1553	LO 1553	0.3404371
LO 1248	LO 1248	0.3404371
LO 1531	LO 1531	0.3114305
LO 7693	LO 7693	0.2731395
LO 1709	LO 1709	0.2669922
LO 885	LO 885	0.2234969
LO 1740	LO 1740	0.2158834
LO 1623	LO 1623	0.1649118
LO 887	LO 887	0.0794108
LO 1629	LO 1629	0.0591222
LO 1082	LO 1082	0.0160246
LO 1453	LO 1453	0.0084841
LO 1458	LO 1458	-0.0148112
LO 1579	LO 1579	-0.0268591

LO 1720	LO 1720	-0.1167494
LO 1575	LO 1575	-0.2685843
LO 1752	LO 1752	-0.2685843
LO 1601	LO 1601	-0.2950130
LO 1452	LO 1452	-0.3223694
LO 1546	LO 1546	-0.3223694
LO 875	LO 875	-0.5235558
LO 1645	LO 1645	-0.5945332
LO 7700	LO 7700	-0.6957111
LO 1582	LO 1582	-0.7256654
LO 1598	LO 1598	-0.8274144
LO 907	LO 907	-0.8274144
LO 1471	LO 1471	-0.9881532
LO 923	LO 923	-0.9881532
LO 1301	LO 1301	-0.9881532
LO 1257	LO 1257	-0.9881532
LO 1585	LO 1585	-0.9881532
LO 1357	LO 1357	-0.9881532
LO 1526	LO 1526	-0.9881532
LO 1358	LO 1358	-0.9881532
LO 1269	LO 1269	-0.9881532
LO 1587	LO 1587	-0.9881532
LO 928	LO 928	-0.9881532
LO 1255	LO 1255	-0.9881532
LO 1454	LO 1454	-0.9881532
LO 1578	LO 1578	-0.9881532
LO 880	LO 880	-0.9881532
LO 1576	LO 1576	-0.9881532
LO 1560	LO 1560	-0.9881532
LO 1352	LO 1352	-0.9881532
LO 1256	LO 1256	-0.9881532
LO 1589	LO 1589	-0.9881532
LO 1646	LO 1646	-0.9881532
LO 1619	LO 1619	-0.9881532
LO 1235	LO 1235	-0.9881532
LO 871	LO 871	-0.9881532
LO 1510	LO 1510	-0.9881532
LO 1757	LO 1757	-0.9881532
LO 918	LO 918	-0.9881532
LO 1620	LO 1620	-0.9881532
LO 1263	LO 1263	-0.9881532
LO 874	LO 874	-0.9881532
LO 1605	LO 1605	-0.9881532
LO 1448	LO 1448	-0.9881532
LO 1238	LO 1238	-0.9881532

LO 1264	LO 1264	-0.9881532
LO 915	LO 915	-0.9881532
LO 1528	LO 1528	-0.9881532
LO 7701	LO 7701	-0.9881532
LO 1111	LO 1111	-0.9881532
LO 1584	LO 1584	-0.9881532
LO 1588	LO 1588	-0.9881532

List of Threshold Concepts based on Betweenness Centrality

We filter the list of concepts based on a cutoff z-score of 2.58 for p-value of 0.01

```
#Filter the dataframe based on a cutoff z-score of 2.58 for p-value of 0.01
nbDMFiltered <- filter(nbDM, round(normalbvalues, 2) >= 1.65)
nbDMFiltered[order(-nbDMFiltered[,2]), ]
```

	LO	V1
4	LO 7682	2.171873
6	LO 1741	2.137526
7	LO 879	1.954822
3	LO 1351	1.908827
2	LO 1748	1.882149
1	LO 1468	1.666211
5	LO 1738	1.648796

Closeness Centrality

The farness/peripherality of a node v is defined as the sum of its distances to all other nodes. The closeness is defined as the inverse of the farness. Source (http://www2.unb.ca/~ddu/6634/Lecture_notes/Lecture_4_centrality_measure.pdf)

For comparison purpose, we can standardize the closeness by dividing by the maximum possible value $1/(n - 1)$

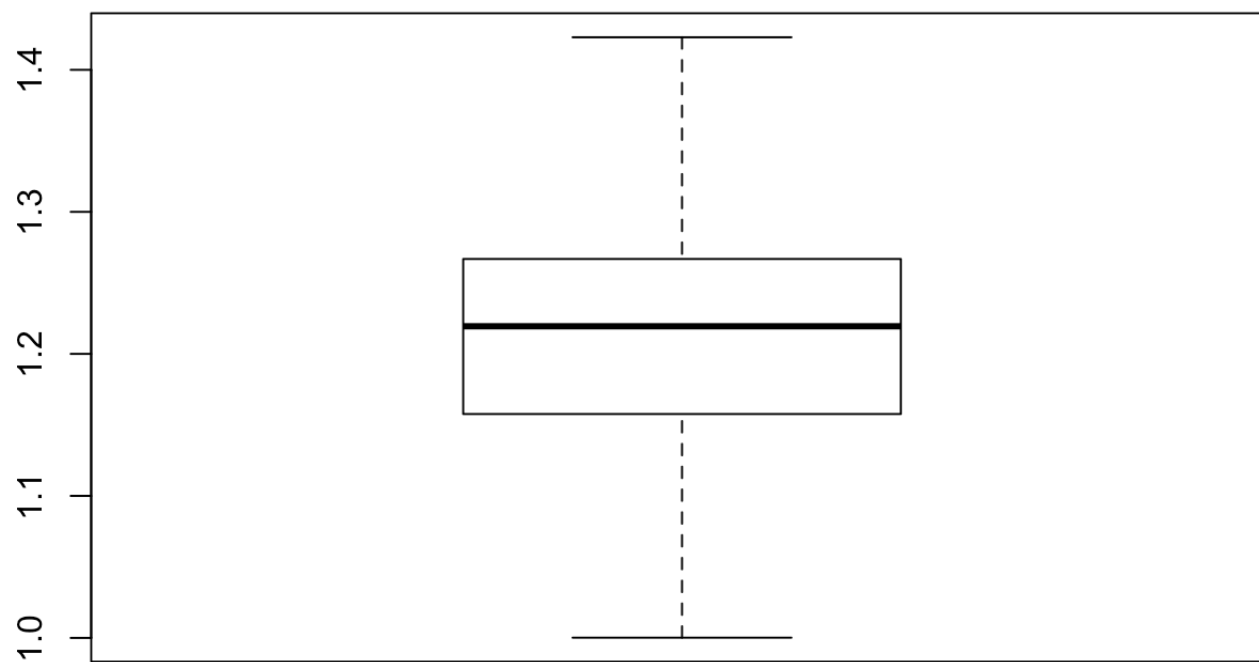
The more central a node is, the lower its total distance to all other nodes. Closeness can be regarded as a measure of how long it will take to spread information from v to all other nodes sequentially.

Let's repeat the above analysis using closeness centrality instead of betweenness centrality.

```
# Compute the standard closeness centrality. Standard closeness allows us to compare the score across nodes.
closestd <- closeness(pathfindernetwork, normalized = TRUE)

# Make the distribution normal by applying a log transformation
closestd <- signedlog10(closestd)

boxplot(closestd)
```



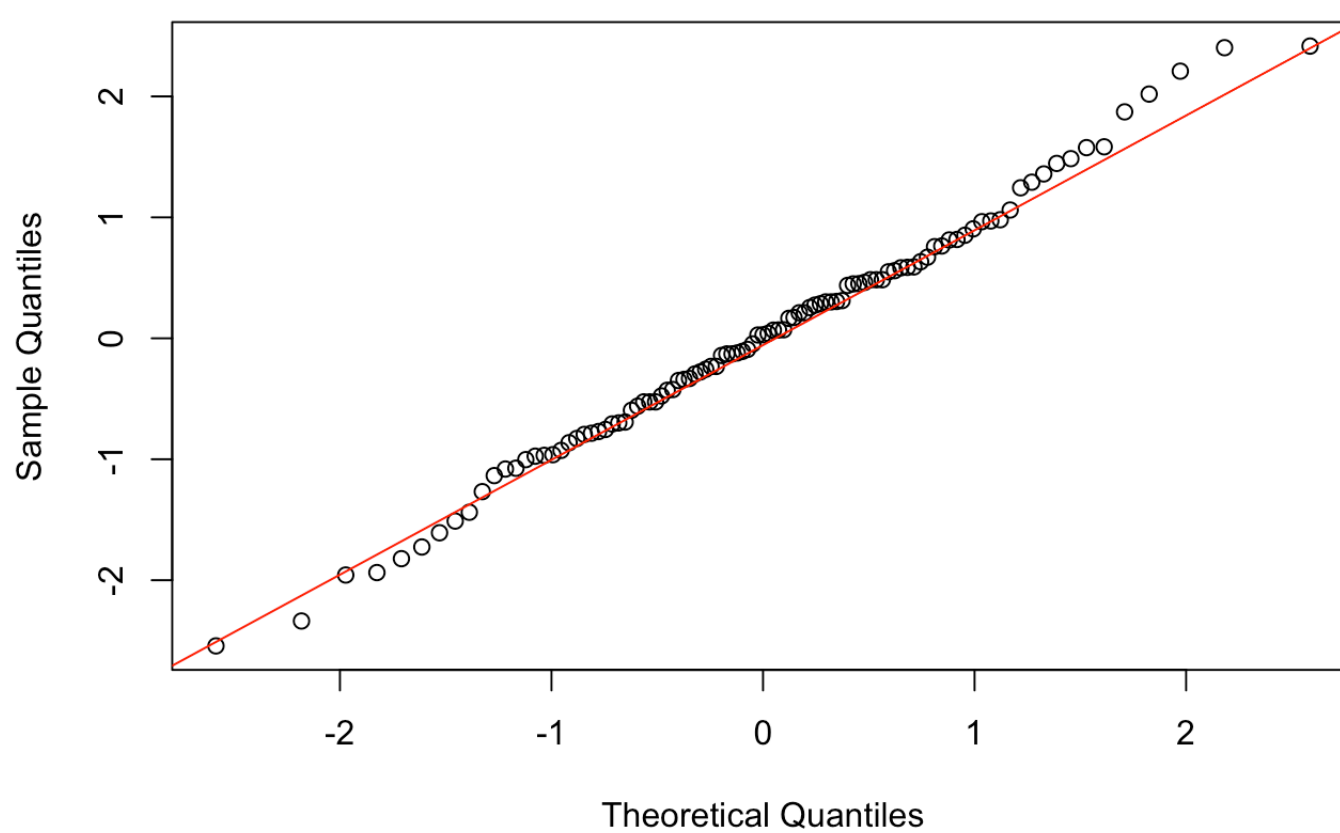
```
# Center and scale the variable
closestd <- scale(closestd, center = TRUE, scale = TRUE)
```

```
## Perform the normality test
shapiro.test(closestd)
```

```
##
## Shapiro-Wilk normality test
##
## data:  closestd
## W = 0.99328, p-value = 0.8953
```

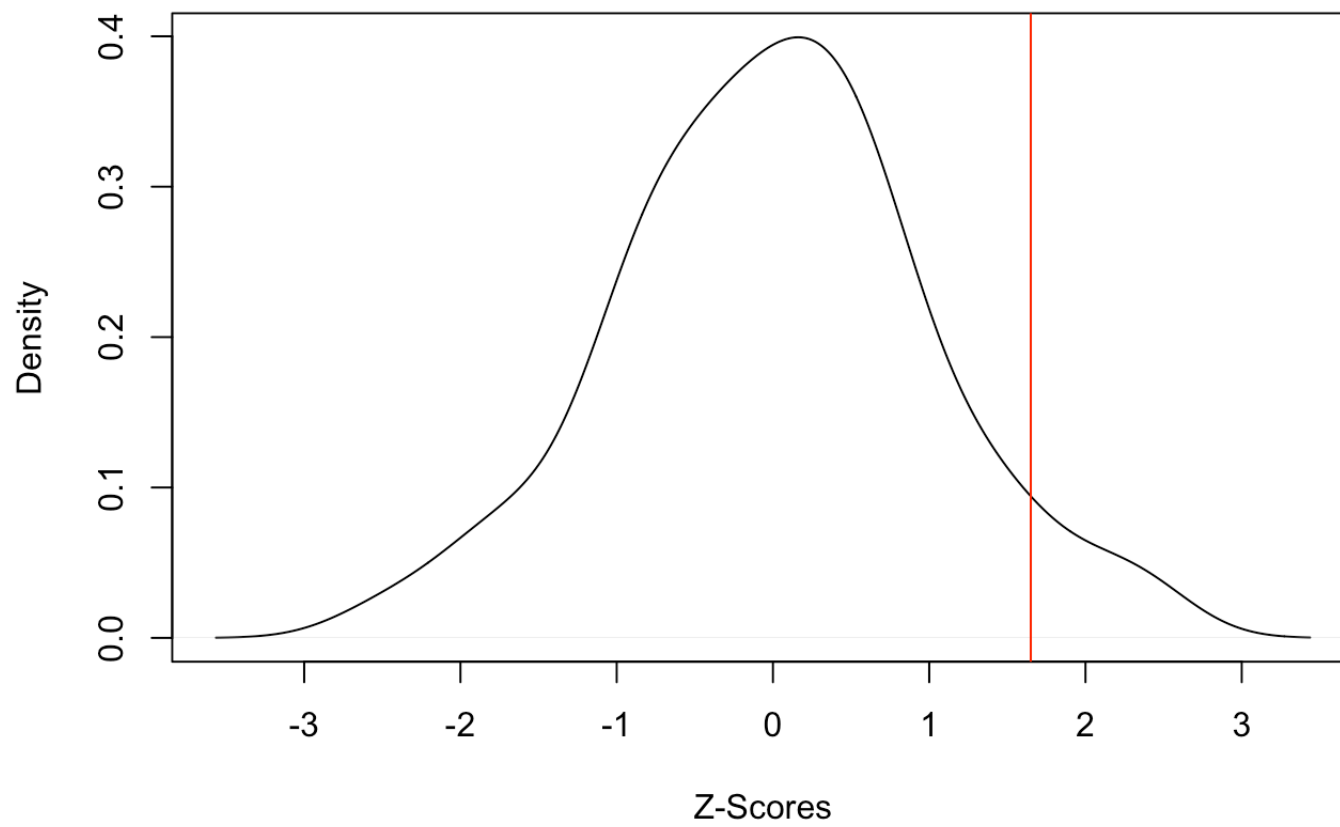
```
## Plot using a qqplot
qqnorm(closestd)
qqline(closestd, col = "red")
```

Normal Q-Q Plot



```
# Plot the distribution
plot(density(closestd), type="l", xlab="Z-Scores")
abline(v = 1.65, col = "red")
```

density.default(x = closestd)



Concepts sorted based on Closeness Centrality

```
ncDM = as.data.frame(closestd)
ncDM["LO"] <- rownames(ncDM)
ncDM <- ncDM[,c(2,1)]
ncDM[order(-ncDM[,2]), ]
```

	LO	V1
LO 1741	LO 1741	2.4160525
LO 7682	LO 7682	2.4028432
LO 1351	LO 1351	2.2087068
LO 1748	LO 1748	2.0192496
LO 879	LO 879	1.8721227
LO 1381	LO 1381	1.5832891
LO 1750	LO 1750	1.5765544
LO 1618	LO 1618	1.4853944
LO 1739	LO 1739	1.4458825
LO 1468	LO 1468	1.3591601
LO 1738	LO 1738	1.2908296
LO 1553	LO 1553	1.2443784
LO 872	LO 872	1.0606686
LO 930	LO 930	0.9802119
LO 1710	LO 1710	0.9702435
LO 1609	LO 1609	0.9642719
LO 1597	LO 1597	0.9049372
LO 1354	LO 1354	0.8521204
LO 882	LO 882	0.8172110
LO 1358	LO 1358	0.8133469
LO 1623	LO 1623	0.7633784
LO 1363	LO 1363	0.7576443
LO 1357	LO 1357	0.6705189

LO 908	LO 908	0.6330981
LO 1257	LO 1257	0.5904013
LO 1628	LO 1628	0.5867054
LO 874	LO 874	0.5830121
LO 1533	LO 1533	0.5590711
LO 1589	LO 1589	0.5498929
LO 1301	LO 1301	0.4842935
LO 1255	LO 1255	0.4842935
LO 1352	LO 1352	0.4842935
LO 1256	LO 1256	0.4626134
LO 1584	LO 1584	0.4518079
LO 1655	LO 1655	0.4500092
LO 7701	LO 7701	0.4374360
LO 1263	LO 1263	0.3116317
LO 1269	LO 1269	0.3046357
LO 912	LO 912	0.3011414
LO 1248	LO 1248	0.3011414
LO 1264	LO 1264	0.2854462
LO 7693	LO 7693	0.2750096
LO 1082	LO 1082	0.2542002
LO 1452	LO 1452	0.2111185
LO 1546	LO 1546	0.2111185
LO 1548	LO 1548	0.1718025
LO 1360	LO 1360	0.1649958
LO 880	LO 880	0.0706446
LO 1629	LO 1629	0.0689755
LO 1619	LO 1619	0.0673070
LO 1510	LO 1510	0.0406848
LO 1238	LO 1238	0.0290812
LO 1235	LO 1235	0.0274257
LO 887	LO 887	-0.0465207
LO 1541	LO 1541	-0.0919959
LO 878	LO 878	-0.1081392
LO 923	LO 923	-0.1210171
LO 1592	LO 1592	-0.1290493
LO 1620	LO 1620	-0.1290493
LO 915	LO 915	-0.1418745
LO 1578	LO 1578	-0.2323315
LO 1453	LO 1453	-0.2323315
LO 1471	LO 1471	-0.2558714
LO 1531	LO 1531	-0.2777443
LO 1454	LO 1454	-0.2948646
LO 1598	LO 1598	-0.3320193
LO 1740	LO 1740	-0.3397258
LO 907	LO 907	-0.3489584
LO 1591	LO 1591	-0.4237376

LO 1579	LO 1579	-0.4297939
LO 1586	LO 1586	-0.4764873
LO 1575	LO 1575	-0.5257273
LO 1601	LO 1601	-0.5257273
LO 1752	LO 1752	-0.5257273
LO 885	LO 885	-0.5612417
LO 1560	LO 1560	-0.5950456
LO 1466	LO 1466	-0.6922585
LO 884	LO 884	-0.6994394
LO 1528	LO 1528	-0.7080431
LO 7700	LO 7700	-0.7536856
LO 1626	LO 1626	-0.7706967
LO 1582	LO 1582	-0.7848294
LO 1581	LO 1581	-0.7946989
LO 1526	LO 1526	-0.8283931
LO 1580	LO 1580	-0.8632558
LO 1709	LO 1709	-0.9267864
LO 1111	LO 1111	-0.9623477
LO 871	LO 871	-0.9677967
LO 1757	LO 1757	-0.9745998
LO 918	LO 918	-1.0030739
LO 1720	LO 1720	-1.0742373
LO 1587	LO 1587	-1.0808998
LO 1458	LO 1458	-1.1352062
LO 1605	LO 1605	-1.2678506
LO 904	LO 904	-1.4382616
LO 875	LO 875	-1.5109843
LO 1588	LO 1588	-1.6079621
LO 1645	LO 1645	-1.7254715
LO 1585	LO 1585	-1.8219429
LO 1576	LO 1576	-1.9369241
LO 1646	LO 1646	-1.9571461
LO 928	LO 928	-2.3372748
LO 1448	LO 1448	-2.5432570

List of Threshold Concepts based on Closeness Centrality

We filter the list of concepts based on a cutoff z-score of 2.58 for p-value of 0.01

```
#Filter the dataframe based on a cutoff z-score of 2.58 for p-value of 0.01
ncDMFiltered <- filter(ncDM, round(closestd, 2) >= 1.65)
ncDMFiltered[order(-ncDMFiltered[,2]), ]
```

	LO	V1
4	LO 1741	2.416053
3	LO 7682	2.402843
2	LO 1351	2.208707
1	LO 1748	2.019250

Final Combined list of Threshold Concepts based on Betweenness and Closeness Centrality

```
merged <- rbind(nbDMFiltered[order(-nbDMFiltered[,2]), ], ncDMFiltered[order(-ncDMFiltered[,2]), ])  
merged <- unique( merged[ , 1] )  
merged
```

```
## [1] "LO 7682" "LO 1741" "LO 879" "LO 1351" "LO 1748" "LO 1468" "LO 1738"
```