```
knitr::opts_chunk$set(warning=FALSE, message=FALSE)
```

```
library(readxl)
library(cooccur)
library(visNetwork)
library(igraph)
library(ggraph)
library(ggplot2)
library(factoextra)
library(ggbiplot)
library(FactoMineR)
library(cooccur)
library(networkD3)
library(repr)
library(ndtv)

data <- read_excel("DocumentTermMatrix2ndOrderFinal.xlsx")
```

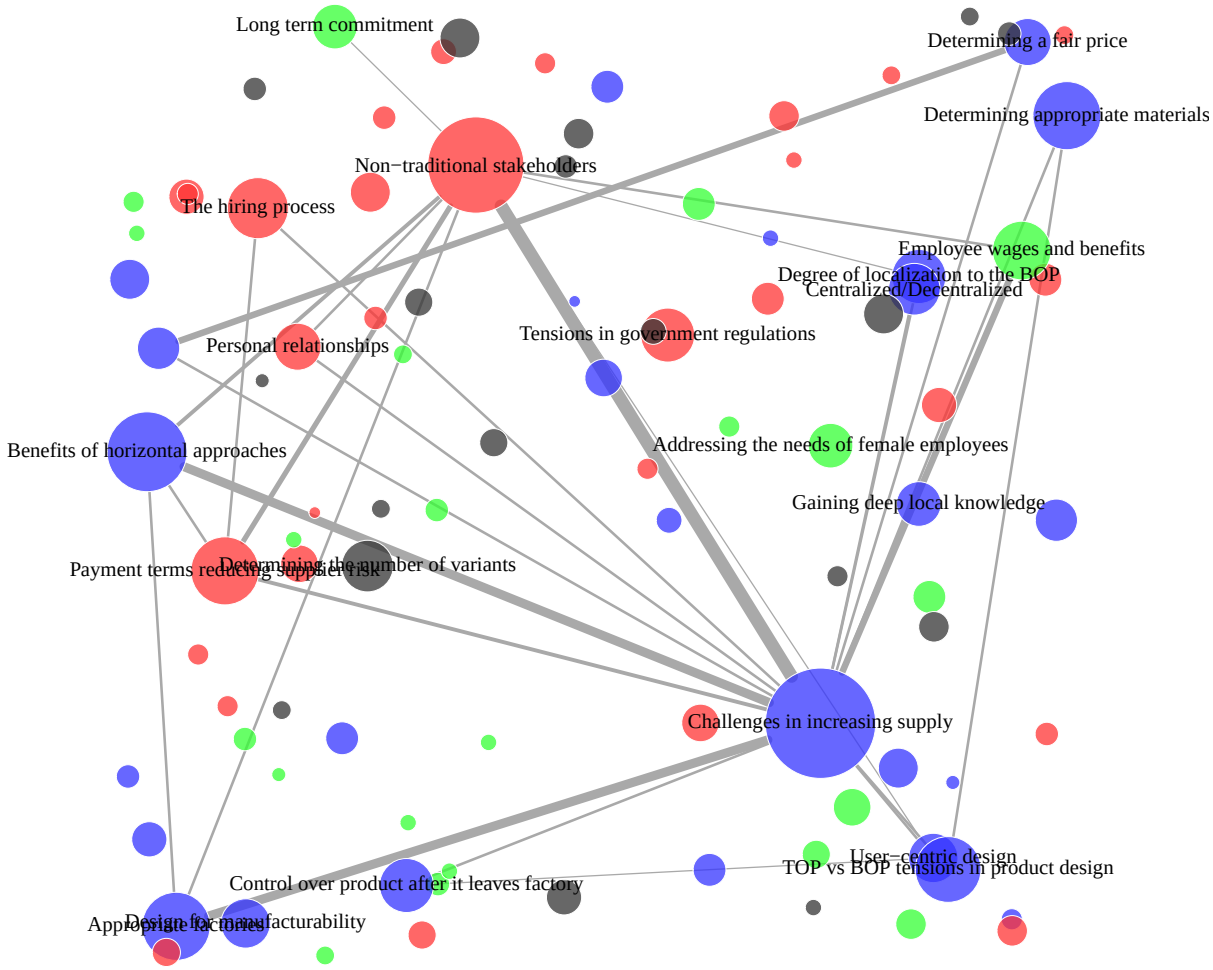
## Co-occurence Analysis

```
binary_data <- data[, -c(1:6)]

freq <- c()
for (i in 1:length(binary_data)){
  freq <- c(freq, sum(binary_data[i]))
}

binary_data <- data.matrix(binary_data)
a <- matrix(0, nrow=95, ncol=95)
for (i in 1:95){
  for (j in 1:95){
    x<-c()
    for (k in 1:95){
      x <- append(x,sum(binary_data[k,i]==1 && binary_data[k,j]==1))
    }
    a[i,j] <- sum(x)
  }
}

network <- graph_from_adjacency_matrix(a , mode='undirected', weighted = T, diag=F)
E(network)$width <- ifelse(E(network)$weight>=10,(E(network)$weight-9),0)
fc <- fastgreedy.community(as.undirected(network))
V(network)$color <- ifelse(membership(fc)==1,rgb(1.0,0.2,0.2,0.75),
                           ifelse(membership(fc)==2,rgb(0.2,1.0,0.2,0.75),
                                  ifelse(membership(fc)==3,rgb(0.2,0.2,1.0,0.75),
                                                           rgb(0.2,0.2,0.2,0.75))))
network <- delete_edges(network, E(network)[E(network)$weight < 10])
```

Weighted Network

```
set.seed(500)
plot(network,
     vertex.label.cex=2,
```

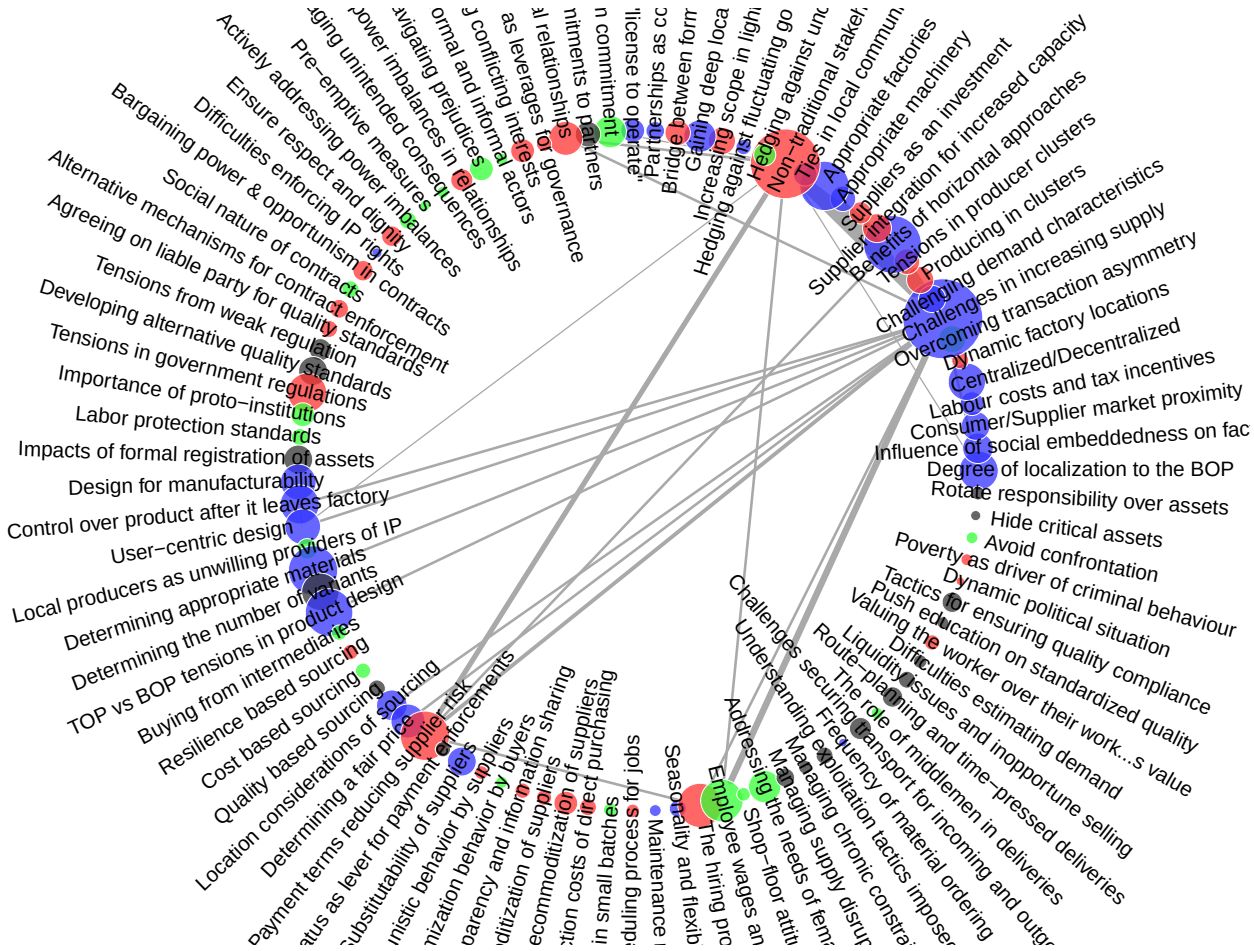
```
     layout=layout.random,
     vertex.label=ifelse(freq>=15,colnames(binary_data),""),
     vertex.size=2+(freq/2),
     vertex.frame.color = "white",
     vertex.label.color="black",
     edge.color="darkgray",
     edge.width=E(network)$width*2)
```




Chord Diagram

```
plot(network,
     layout=layout.circle,
     vertex.size=2+(freq/2),
     vertex.frame.color = "white",
```

```
     vertex.label.cex=0.1,
     edge.color="darkgray",
     edge.width=E(network)$width*2)
la<- layout.circle(network)
x <- la[,1]*1.3
y <- la[,2]*1.3
angle = ifelse(atan(-(la[,1]/la[,2]))*(180/pi) < 0, 90 + atan(- (la[,1]/la[,2]))*(180/pi),
               270 + atan(-la[,1]/la[,2])*(180/pi))
for (i in 1:length(x)) {
  text(x=x[i], y=y[i], labels=colnames(binary_data)[i], adj=NULL, pos=NULL, cex=2,
       col="black", srt=angle[i], xpd=T)
}
```

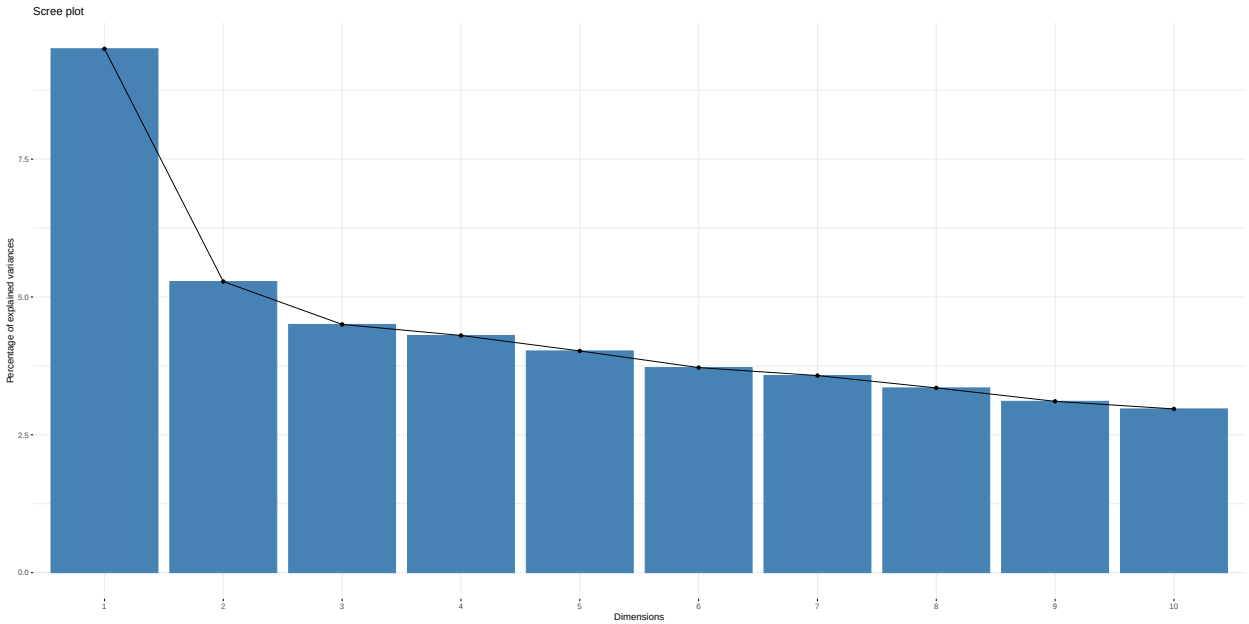



# Multiple Correspondence Analysis

### Based on Terms

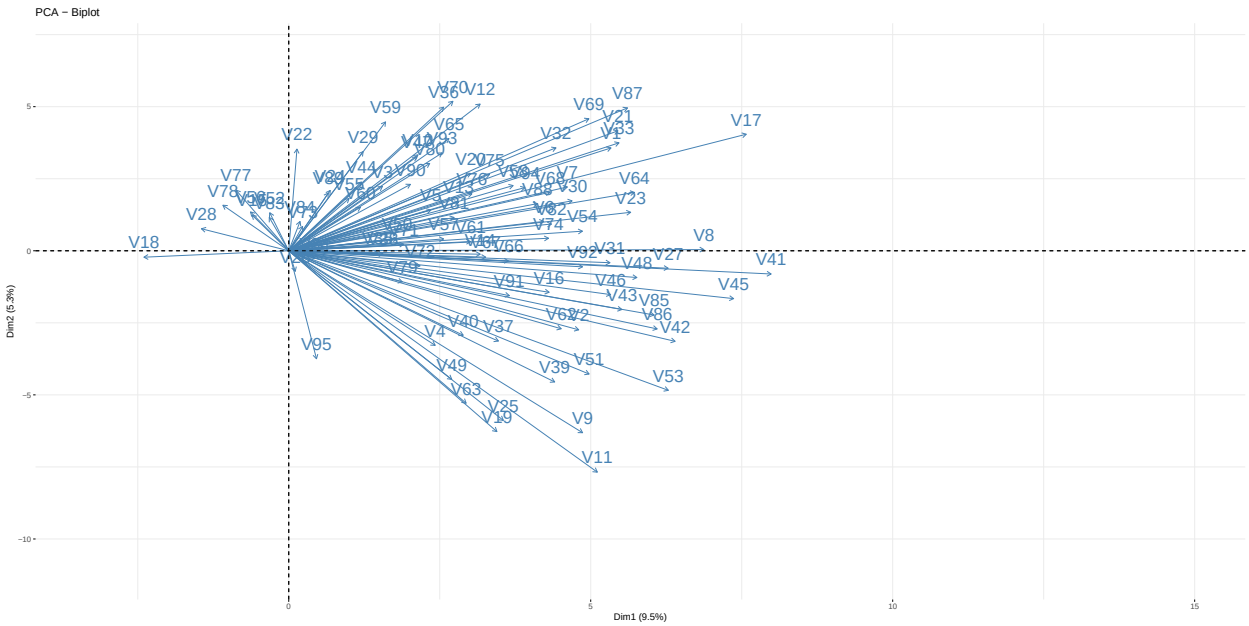
Eigenvalues

```
fviz_eig(mca_model)
```




Biplot

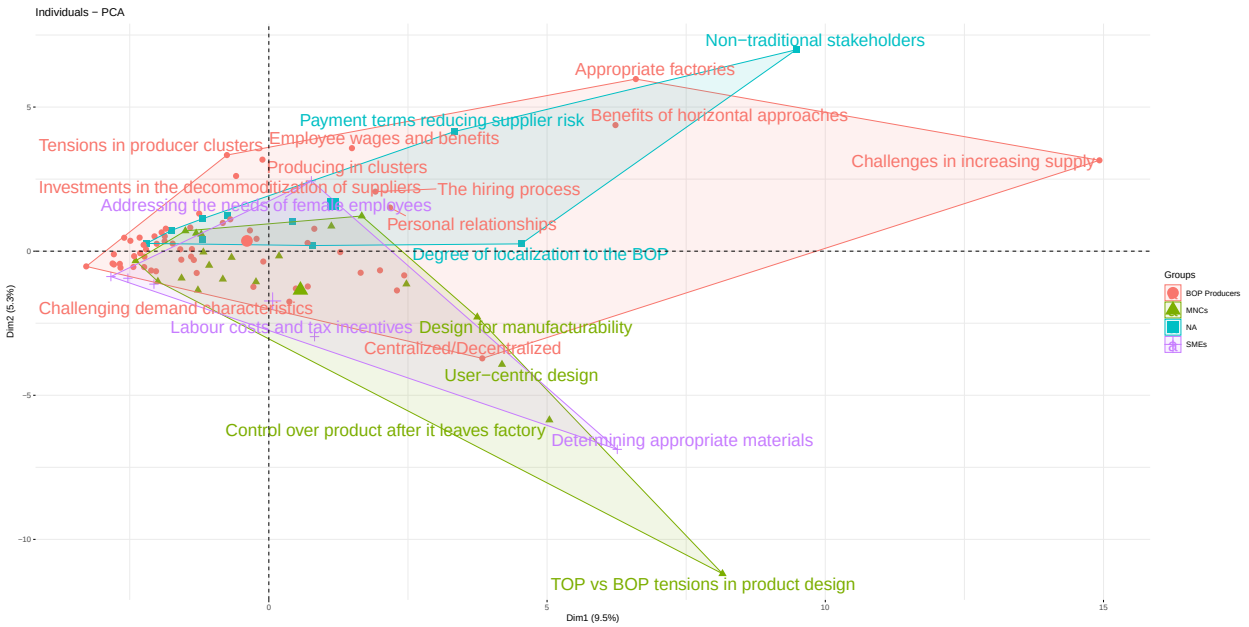
```
fviz_pca_biplot(mca_model, invisible = "ind", labelsize = 7)
```




Clustered MCA based on “Focal Organization Type”

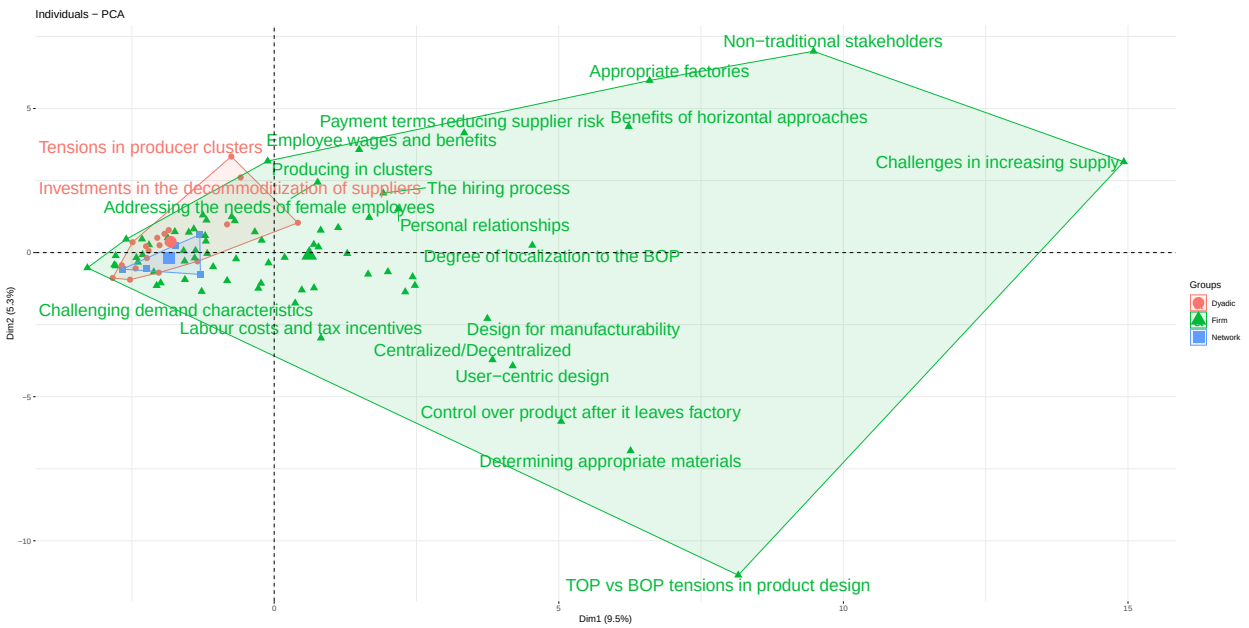
```
groups<-c()
for (i in 7:101){
  x <- data[data[i]==1,]
  x <- count(x$`Focal Organzation Type`)
  x <- as.character(x$x[x$freq==max(x$freq)])
  groups <- append(groups, x[1])
```

```
}
fviz_pca_ind(mca_model,
             col.ind = groups,
             labelsize =7,
             pointsize=3,
             repel = TRUE,
             addEllipses = TRUE,
             ellipse.type = "convex",
             legend.title = "Groups",)
```



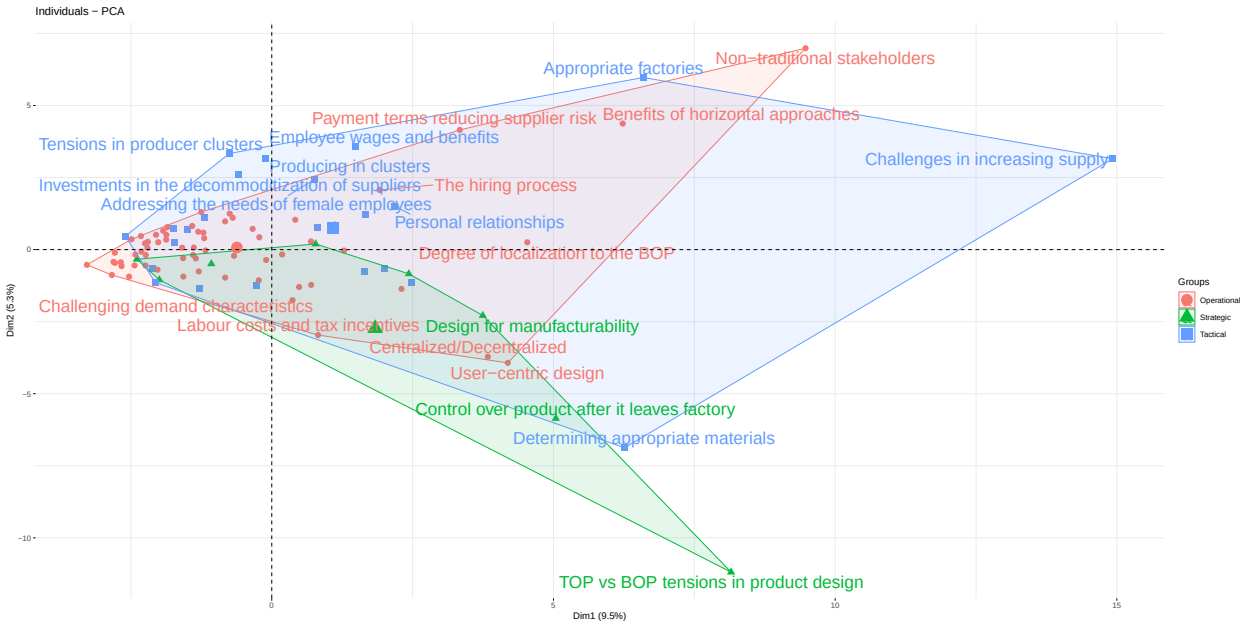

Clustered MCA based on "Unit of Analysis (Scope)"

```
groups<-c()
for (i in 7:101){
  x <- data[data[i]==1,]
  x <- count(x$`Unit of Analysis  (Scope)`)
  x <- as.character(x$x[x$freq==max(x$freq)])
  groups <- append(groups, x[1])
}
fviz_pca_ind(mca_model,
             col.ind = groups,
             labelsize = 7,
             pointsize=3,
             repel = TRUE,
             addEllipses = TRUE,
             ellipse.type = "convex",
             legend.title = "Groups",)
```




Clustered MCA based on "Level of Analysis (Depth)"

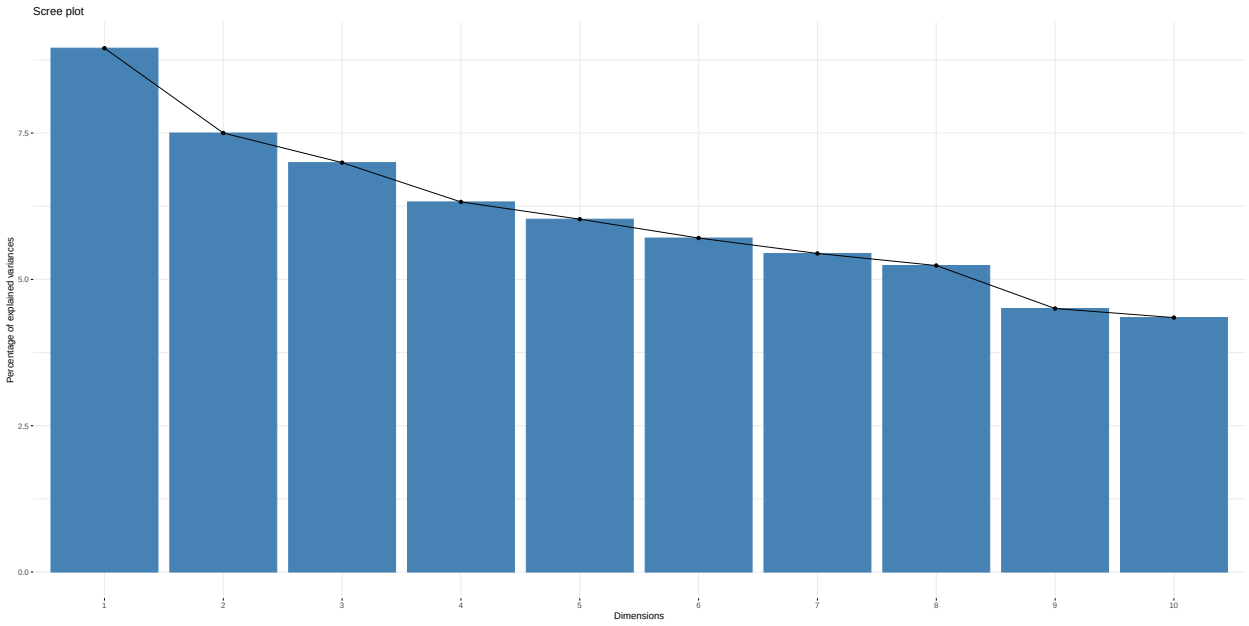
```
groups<-c()
for (i in 7:101){
  x <- data[data[i]==1,]
  x <- count(x$`Level of Analysis (Depth)`)
  x <- as.character(x$x[x$freq==max(x$freq)])
  groups <- append(groups, x[1])
}
fviz_pca_ind(mca_model,
             col.ind = groups,
             labelsize =7,
             pointsize=3,
             repel = TRUE,
             addEllipses = TRUE,
             ellipse.type = "convex",
             legend.title = "Groups",)
```




### Based on Categorical data

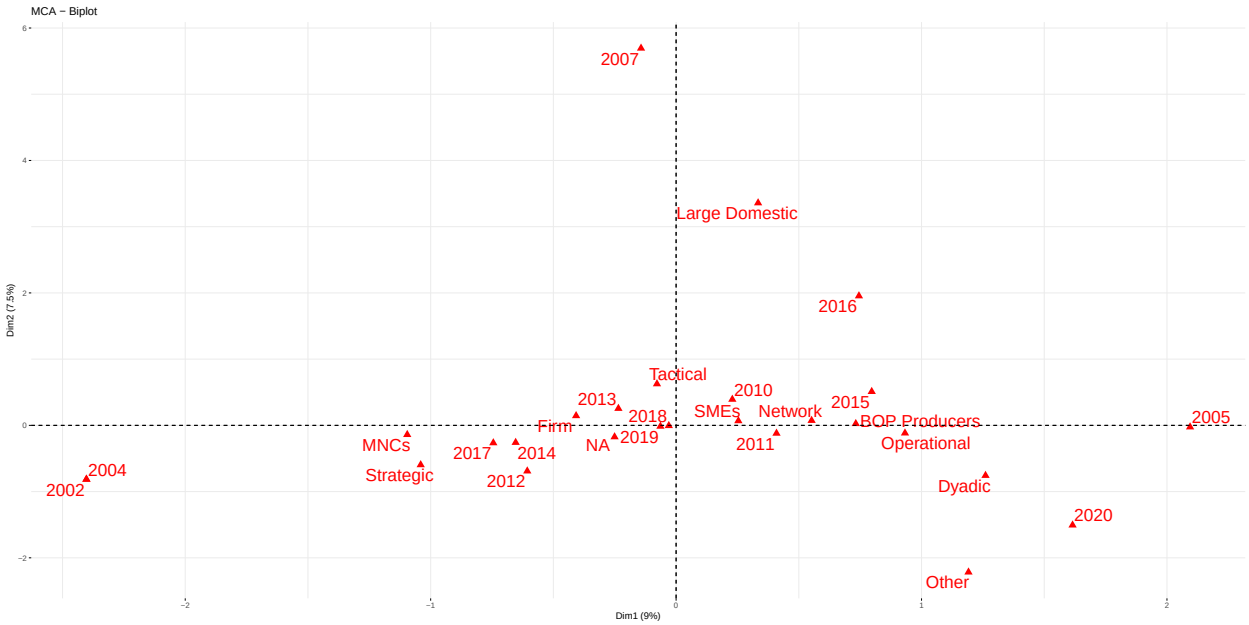
Eigenvalues

```
fviz_eig(mca_model)
```




Biplot

```
fviz_mca_biplot(mca_model, invisible = "ind", labelsize =7, repel = TRUE, pointsize=3)
```



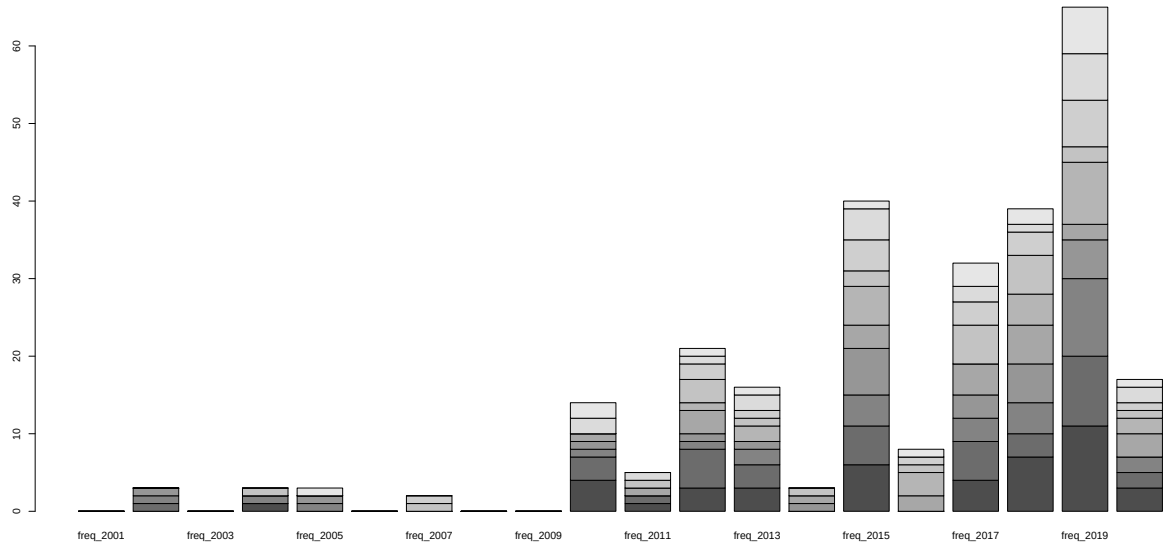

# Occurrences over time

```
Year <- categorical_data$Year
mixed_data <- cbind(binary_data, Year)
mixed_data <- mixed_data[order(Year),]
mixed_data <- data.frame(mixed_data)

year_2001 <- mixed_data[mixed_data[96] == 2001, ]
year_2002 <- mixed_data[mixed_data[96] == 2002, ]
year_2003 <- mixed_data[mixed_data[96] == 2003, ]
year_2004 <- mixed_data[mixed_data[96] == 2004, ]
year_2005 <- mixed_data[mixed_data[96] == 2005, ]
year_2006 <- mixed_data[mixed_data[96] == 2006, ]
year_2007 <- mixed_data[mixed_data[96] == 2007, ]
year_2008 <- mixed_data[mixed_data[96] == 2008, ]
year_2009 <- mixed_data[mixed_data[96] == 2009, ]
year_2010 <- mixed_data[mixed_data[96] == 2010, ]
year_2011 <- mixed_data[mixed_data[96] == 2011, ]
year_2012 <- mixed_data[mixed_data[96] == 2012, ]
year_2013 <- mixed_data[mixed_data[96] == 2013, ]
year_2014 <- mixed_data[mixed_data[96] == 2014, ]
year_2015 <- mixed_data[mixed_data[96] == 2015, ]
year_2016 <- mixed_data[mixed_data[96] == 2016, ]
year_2017 <- mixed_data[mixed_data[96] == 2017, ]
year_2018 <- mixed_data[mixed_data[96] == 2018, ]
year_2019 <- mixed_data[mixed_data[96] == 2019, ]
year_2020 <- mixed_data[mixed_data[96] == 2020, ]

freq_2001 <- as.double(rep.int(0, 95))
freq_2003 <- as.double(rep.int(0, 95))
freq_2006 <- as.double(rep.int(0, 95))
freq_2008 <- as.double(rep.int(0, 95))
```

```
freq_2009 <- as.double(rep.int(0, 95))

freq_2002 <- c()
freq_2004 <- c()
freq_2005 <- c()
freq_2007 <- c()
freq_2010 <- c()
freq_2011 <- c()
freq_2012 <- c()
freq_2013 <- c()
freq_2014 <- c()
freq_2015 <- c()
freq_2016 <- c()
freq_2017 <- c()
freq_2018 <- c()
freq_2019 <- c()
freq_2020 <- c()

for (i in 1:95){
  freq_2002 <- c(freq_2002, sum(year_2002[i]))
  freq_2004 <- c(freq_2004, sum(year_2004[i]))
  freq_2005 <- c(freq_2005, sum(year_2005[i]))
  freq_2007 <- c(freq_2007, sum(year_2007[i]))
  freq_2010 <- c(freq_2010, sum(year_2010[i]))
  freq_2011 <- c(freq_2011, sum(year_2011[i]))
  freq_2012 <- c(freq_2012, sum(year_2012[i]))
  freq_2013 <- c(freq_2013, sum(year_2013[i]))
  freq_2014 <- c(freq_2014, sum(year_2014[i]))
  freq_2015 <- c(freq_2015, sum(year_2015[i]))
  freq_2016 <- c(freq_2016, sum(year_2016[i]))
  freq_2017 <- c(freq_2017, sum(year_2017[i]))
  freq_2018 <- c(freq_2018, sum(year_2018[i]))
  freq_2019 <- c(freq_2019, sum(year_2019[i]))
  freq_2020 <- c(freq_2020, sum(year_2020[i]))
}

new_data <- data.frame(cbind(freq_2001,freq_2002,freq_2003,freq_2004,freq_2005,freq_2006,
                             freq_2007,freq_2008,freq_2009,freq_2010,freq_2011,freq_2012,
                             freq_2013,freq_2014,freq_2015,freq_2016,freq_2017,freq_2018,
                             freq_2019,freq_2020), row.names = colnames(binary_data))
new_data <- cbind(new_data, freq)
new_data <- new_data[order(new_data$freq, decreasing = TRUE),]
new_data <- new_data[,-21]
new_data <- head(new_data,10)
new_data <- data.frame(t(new_data))
```

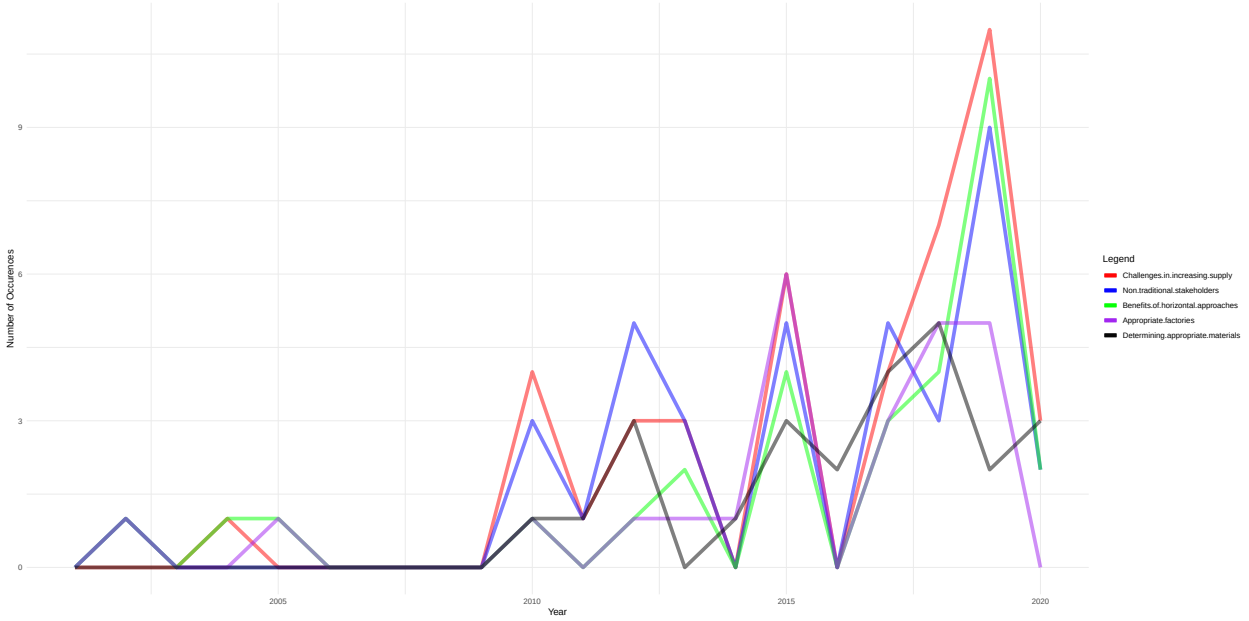
Stacked Barplot of top 10 most occurring terms

```
barplot(height = t(new_data),beside = FALSE)
```




Line chart of top 5 most occurring terms

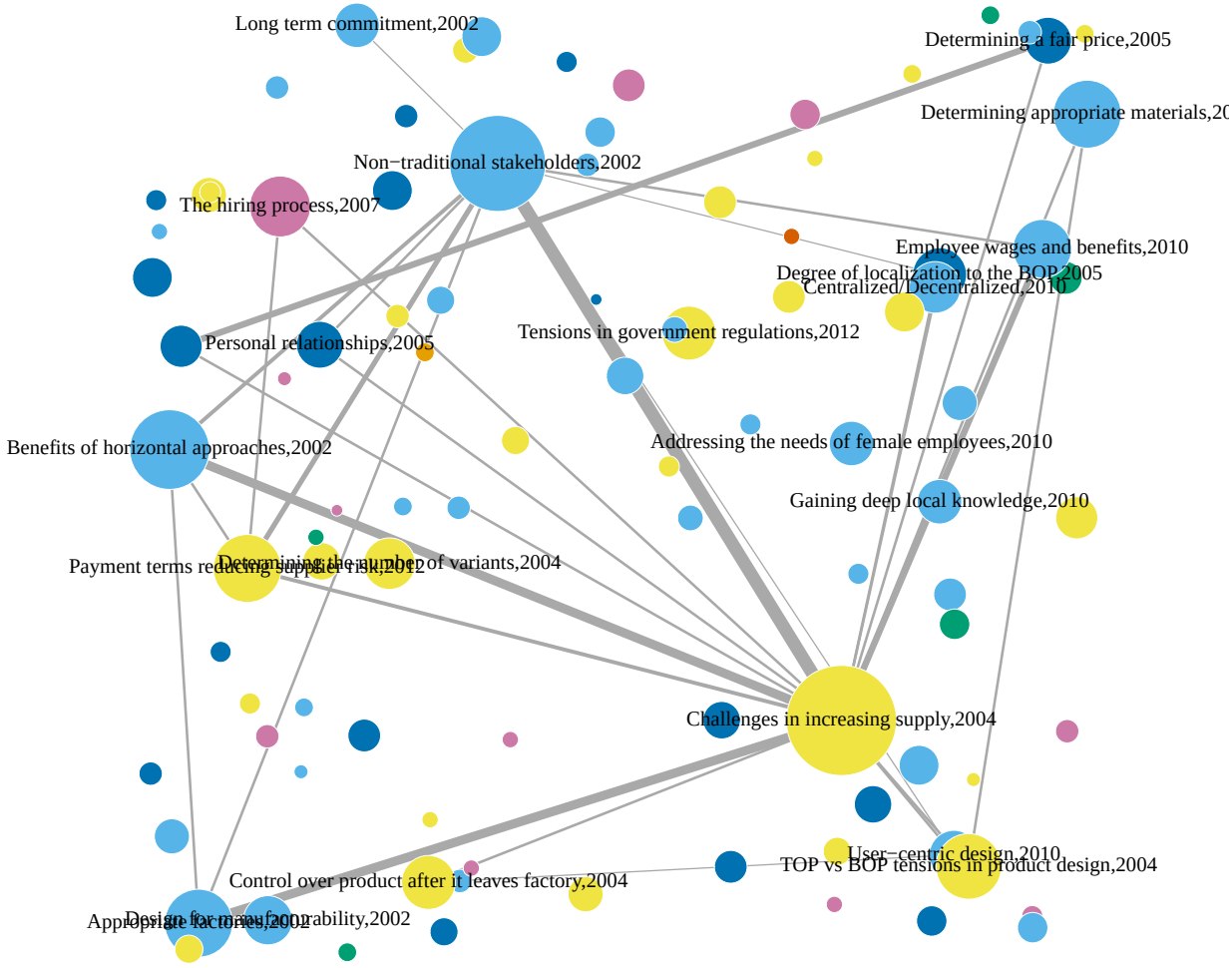
```
colors <- c("Challenges.in.increasing.supply" = "red",
            "Non.traditional.stakeholders" = "blue",
            "Benefits.of.horizontal.approaches" = "green",
            "Appropriate.factories" = "purple",
            "Determining.appropriate.materials" = "black")
ggplot(new_data) +
  geom_line(aes(x = 2001:2020, y = new_data[,1], color="Challenges.in.increasing.supply"),
            stat = "identity", alpha=0.5, size=2)+
  geom_line(aes(x = 2001:2020, y = new_data[,2], color="Non.traditional.stakeholders"),
            stat = "identity", alpha=0.5, size=2)+
  geom_line(aes(x = 2001:2020, y = new_data[,3], color="Benefits.of.horizontal.approaches"),
            stat = "identity", alpha=0.5, size=2)+
  geom_line(aes(x = 2001:2020, y = new_data[,4], color="Appropriate.factories"),
            stat = "identity", alpha=0.5, size=2)+
  geom_line(aes(x = 2001:2020, y = new_data[,5], color="Determining.appropriate.materials"),
            stat = "identity", alpha=0.5, size=2)+
  labs(x = "Year", y = "Number of Occurences", color = "Legend") +
  scale_color_manual(values = colors)+
  theme_minimal()
```




```
first_year <- c()
for (i in 1:95){
  for (j in 1:95){
    if (binary_data[j,i]==1){
      t <- data$Year[j]
      break
    }
  }
  first_year <- append(first_year, t)
}

network <- graph_from_adjacency_matrix(a , mode='undirected', weighted = T, diag=F)
E(network)$width <- ifelse(E(network)$weight>=10,(E(network)$weight-9),0)
network <- delete_edges(network, E(network)[E(network)$weight < 10])
```

Co-occurrence network of terms grouped by the first year of appearance

```
set.seed(500)
plot(network,
     vertex.label.cex=2,
     layout=layout.random,
     vertex.label= ifelse(freq>=15,paste(colnames(binary_data),first_year, sep = ","),""),
     vertex.size=2+(freq/2),
     vertex.frame.color = "white",
     vertex.label.color="black",
     vertex.color=first_year,
     edge.color="darkgray",
     edge.width=E(network)$width*2)
```




```
binary_data <- data[, -c(1:6)]
binary_data <- t(t(binary_data)[order(freq, decreasing = TRUE),])
#binary_data <- binary_data[,1:20]
a <- matrix(0, nrow=95, ncol=95)
for (i in 1:95){
  for (j in 1:95){
    x<-c()
    for (k in 1:95){
      x <- append(x,sum(binary_data[k,i]==1 && binary_data[k,j]==1))
    }
    a[i,j] <- sum(x)
  }
}
network <- graph.tree(a, n = 20, mode = "undirected")
```

```
el <- as_edgelist(network, names = FALSE)
infectionTimes <- first_year
infTree<-network(el,vertex.attr = list(infectionTimes),
                 vertex.attrnames = list('infectionTimes'))
transmissionTimeline(infTree,
                     time.attr='infectionTimes',
                     label = colnames(data[7:101])[order(freq, decreasing = TRUE)][1:20],
                     displaylabels = TRUE,# !missing(label),
                     label.cex = 2,
                     label.col = 1,
                     vertex.col = 2,
                     vertex.border = 1,
                     vertex.lwd = 1,
                     vertex.sides = 50,
                     vertex.cex = ((sort(freq, decreasing = TRUE)[1:20])-10)/2,
                     jitter=TRUE,
                     edge.col = "gray",
                     edge.lty = 1,
                     edge.lwd = 1,
                     xlab = "Year of first occurence")
```