Latent Topic Modeling on Twitter Data

# Method: CA

Correspondence Analysis (CA) is a multivariate graphical technique designed to explore relationships among categorical variables. The outcome from correspondence analysis is a graphical display of the rows and columns of a contingency table that is designed to permit visualization of the salient relationships among the variable responses in a low-dimensional space. Such a representation reveals a more global picture of the relationships among row-column pairs which would otherwise not be detected through a pairwise analysis.

**Calculate CA:**

* Step 1: Compute row and column averages
* Step 2: Compute the expected values
* Step 3: Compute the residuals
* Step 4: Plotting labels with similar residuals close together
* Step 5: Interpreting the relationship between row and column labels

**How to Interpret Correspondence Analysis Plots**

Correspondence analysis does not show us which rows have the highest numbers, nor which columns have the highest numbers. It instead shows us the relativities.

* The further things are from the origin, the more discriminating they are.
* Look at the length of the line connecting the row label to the origin. Longer lines indicate that the row label is highly associated with some of the column labels (i.e., it has at least one high residual).
* Look at the length of the label connecting the column label to the origin. Longer lines again indicate a high association between the column label and one or more row labels.
* Look at the angle formed between these two lines. Really small angles indicate association. 90 degree angles indicate no relationship. Angles near 180 degrees indicate negative associations.

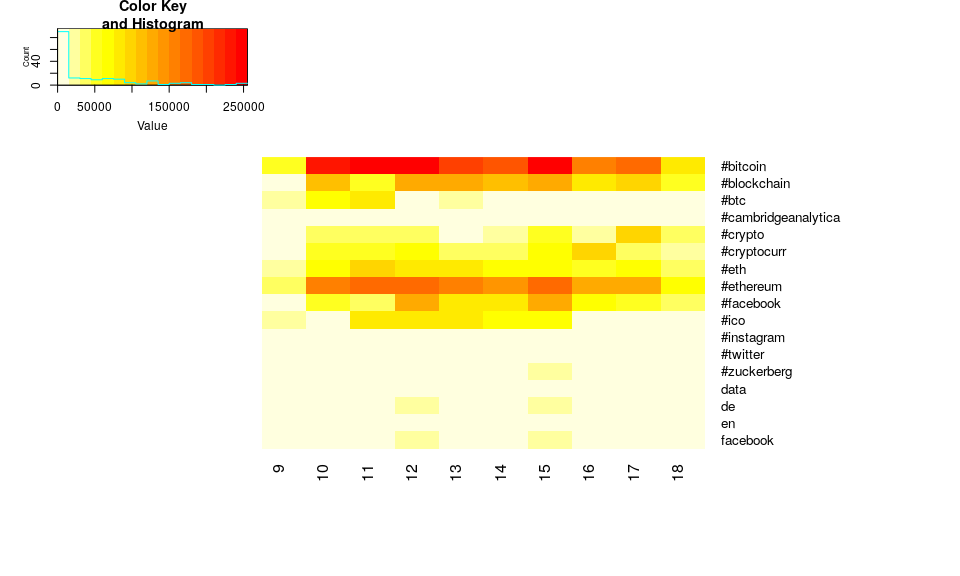
# Dataset

## 9 10 11 12 13 14 15 16  
## #bitcoin 50383 229333 245775 245612 195730 190175 255195 157049  
## #blockchain 9196 113169 54362 134326 124176 108986 133629 86186  
## #btc 21335 63269 85576 0 28262 0 0 0  
## #cambridgeanalytica 0 0 0 13947 0 0 0 0  
## #crypto 8461 41888 44723 44898 0 29718 50492 23863  
## #cryptocurr 10076 49910 57015 60767 40541 37868 65620 91314  
## #eth 24771 74637 99115 77422 87220 71144 73107 58282  
## #ethereum 37892 153520 172861 168796 158829 148538 171729 127553  
## #facebook 949 50748 41961 130670 86932 76678 128772 63718  
## #ico 17204 0 80290 75305 76068 68794 70205 0  
## #instagram 153 7778 5601 0 0 7760 0 7887  
## #twitter 160 8199 6384 0 9690 8526 0 8848  
## #zuckerberg 0 0 0 0 0 0 18855 0  
## data 0 0 0 13700 8052 0 11498 0  
## de 113 6790 5513 21954 13265 12369 19778 10902  
## en 120 5497 0 0 0 0 0 7677  
## facebook 0 0 4555 15673 10760 8853 15829 0  
## 17 18  
## #bitcoin 170079 79636  
## #blockchain 103272 50301  
## #btc 0 0  
## #cambridgeanalytica 0 0  
## #crypto 91095 43452  
## #cryptocurr 37879 17766  
## #eth 67597 32730  
## #ethereum 132309 65283  
## #facebook 57189 32773  
## #ico 0 0  
## #instagram 7987 3791  
## #twitter 8827 3895  
## #zuckerberg 0 0  
## data 0 0  
## de 8529 5478  
## en 6994 0  
## facebook 0 3817

* Research Question
  + Do we see new words appearing on a particular week

# Analysis

## Heatmap

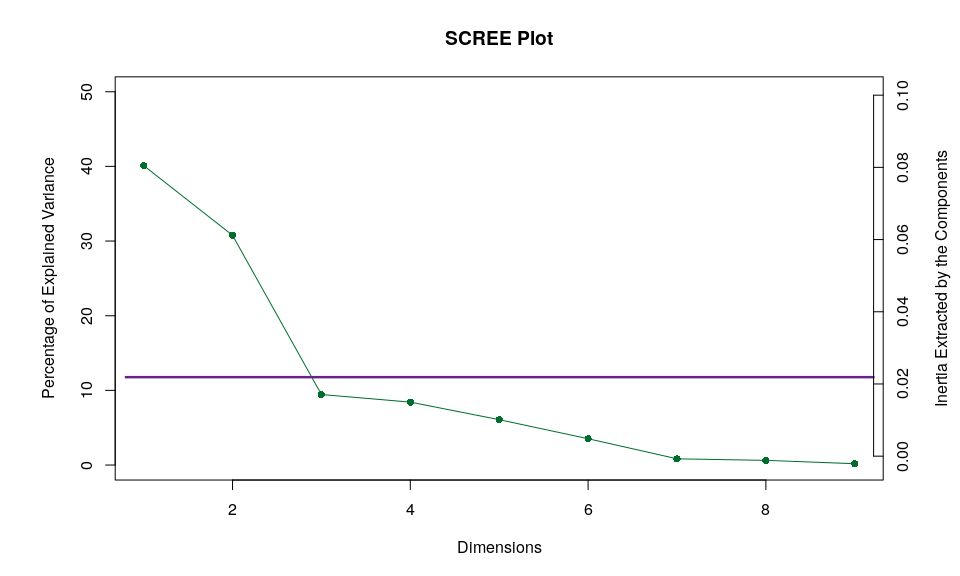


## Scree Plot

Gives amount of information explained by corresponding component. Gives an intuition to decide which components best represent data in order to answer the research question.

P.S. The most contribution component may not always be most useful for a given research question.

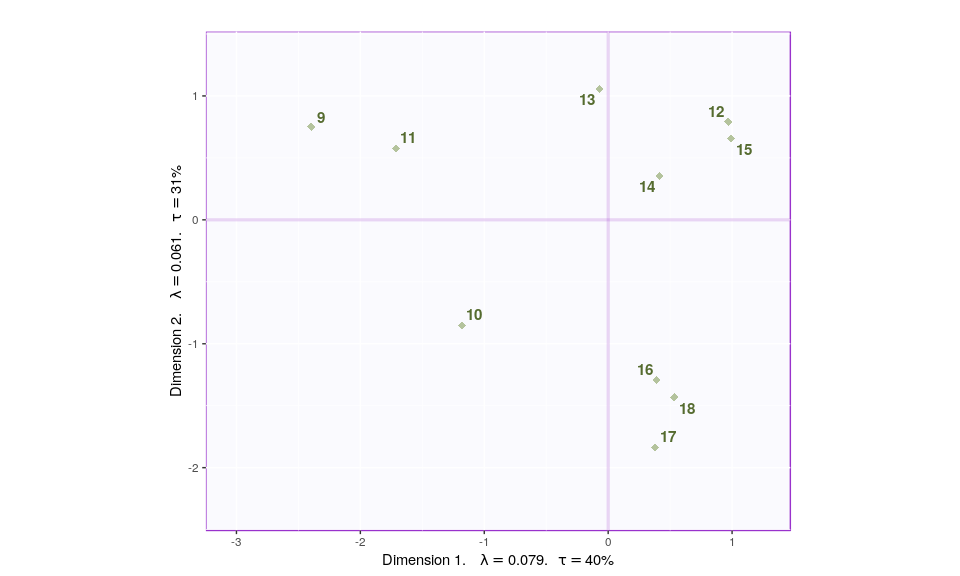
PTCA4CATA::PlotScree(ev = resCA.sym$ExPosition.Data$eigs,  
 #p.ev = we\_data\_inf$Inference.Data$components$p.vals,  
 title = 'SCREE Plot',  
 plotKaiser = TRUE  
)



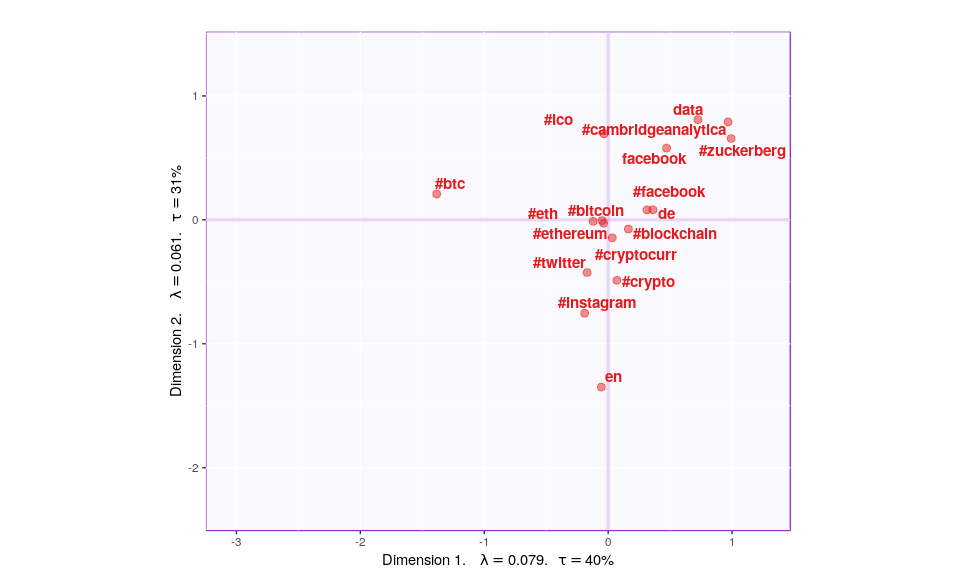
## Factor Scores

### Asymmetric Plot

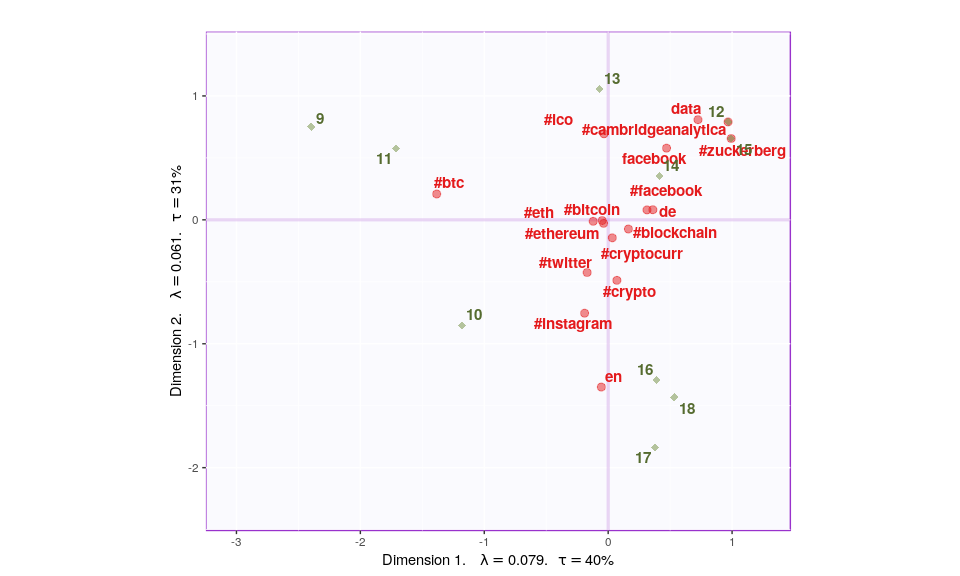
#### Asymmetric Plot  
  
map.IJ.asym <- asymMap$baseMap + asymMap$J\_labels +   
 asymMap$J\_points + labels4CA #+ legend$zeMap\_dots + legend$zeMap\_text  
print(map.IJ.asym)



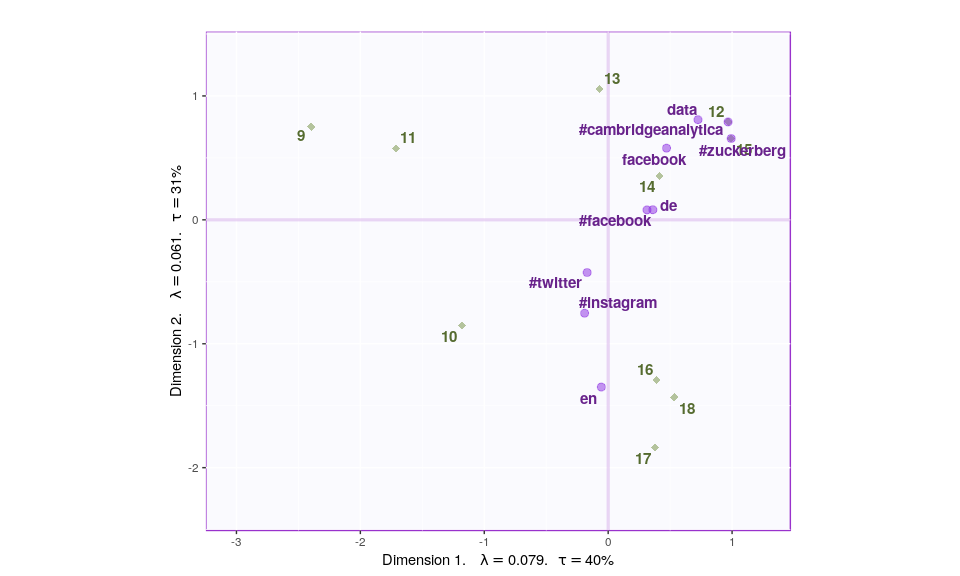
#### Asymmetric Plot  
  
map.IJ.asym <- asymMap$baseMap + asymMap$I\_labels +   
 asymMap$I\_points + labels4CA #+ legend$zeMap\_dots + legend$zeMap\_text  
print(map.IJ.asym)



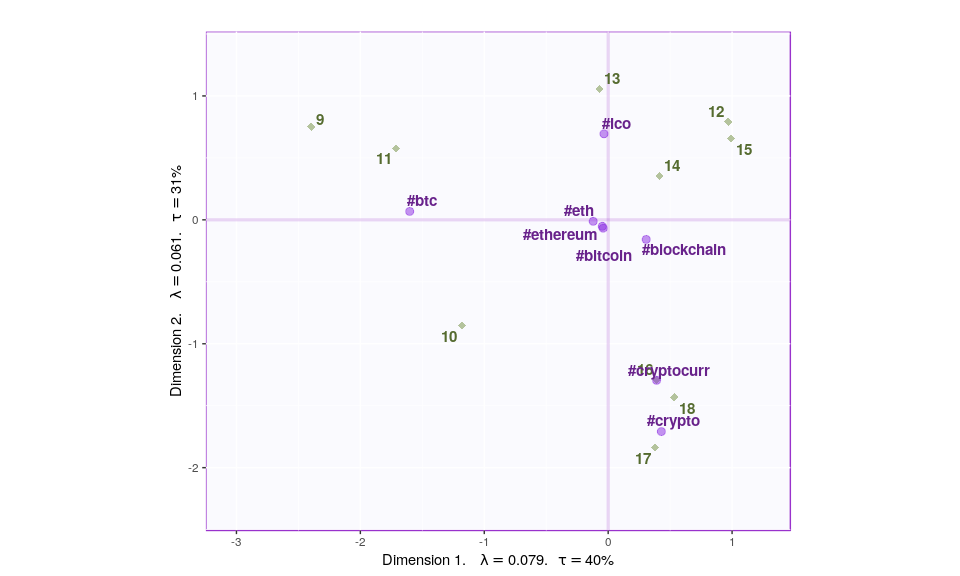
#### Asymmetric Plot  
  
map.IJ.asym <- asymMap$baseMap + asymMap$I\_labels +   
 asymMap$I\_points + asymMap$J\_labels +   
 asymMap$J\_points + labels4CA #+ legend$zeMap\_dots + legend$zeMap\_text  
print(map.IJ.asym)



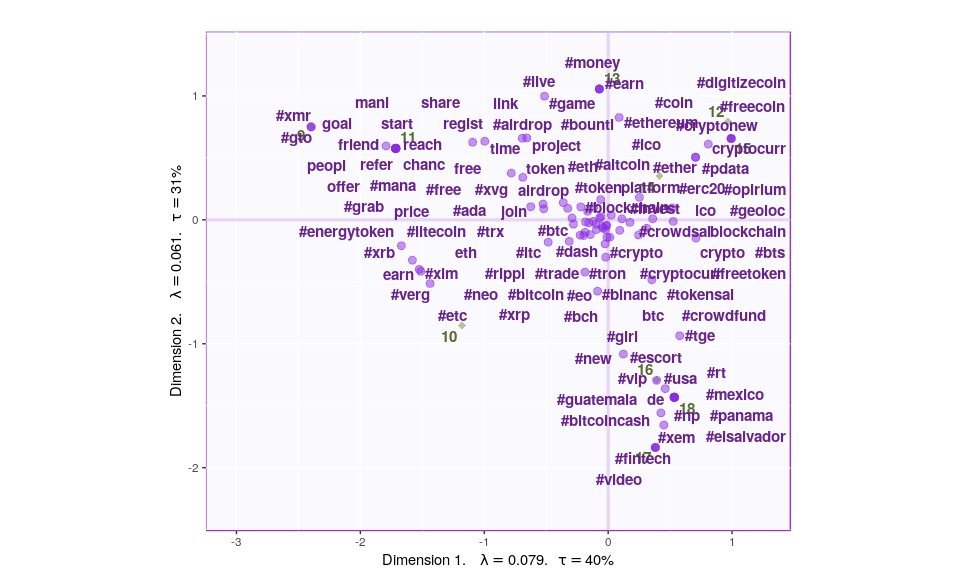
tweets\_facebook\_supp <- supplementaryRows(tweets\_facebook,resCA.asym)  
asymMap\_supp <- createFactorMapIJ(tweets\_facebook\_supp$fii,Fj,  
 #col.points.i = color4Authors,  
 #col.labels.i = color4Authors  
 )  
  
map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 #asymMap$I\_points +   
 asymMap$J\_labels +   
 asymMap$J\_points + labels4CA + #+ legend$zeMap\_dots + legend$zeMap\_text  
 asymMap\_supp$I\_points + asymMap\_supp$I\_labels  
  
print(map.IJ.asym)



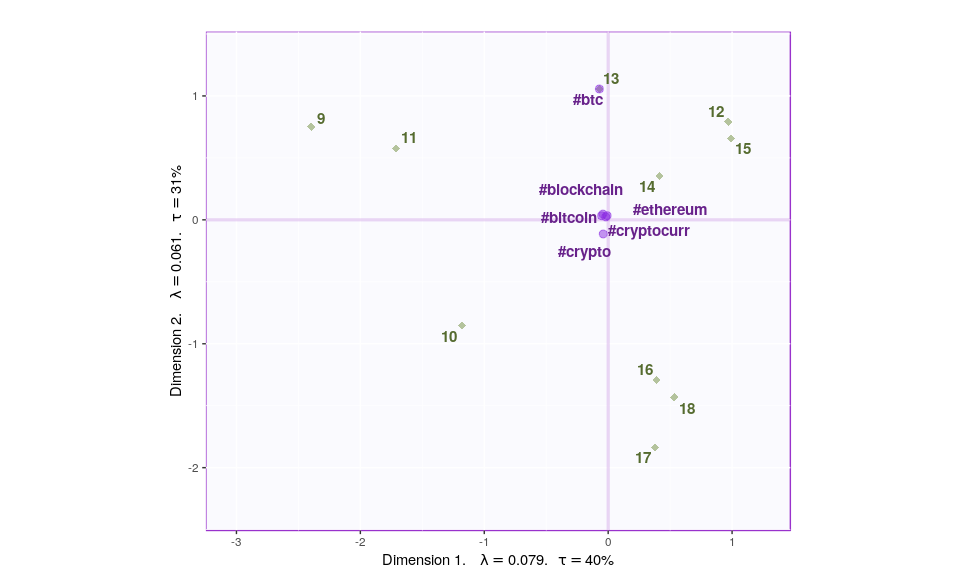
tweets\_eth\_supp <- supplementaryRows(tweets\_eth,resCA.asym)  
asymMap\_supp <- createFactorMapIJ(tweets\_eth\_supp$fii,Fj,  
 #col.points.i = color4Authors,  
 #col.labels.i = color4Authors  
 )  
  
map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 #asymMap$I\_points +   
 asymMap$J\_labels +   
 asymMap$J\_points + labels4CA + #+ legend$zeMap\_dots + legend$zeMap\_text  
 asymMap\_supp$I\_points + asymMap\_supp$I\_labels  
  
print(map.IJ.asym)



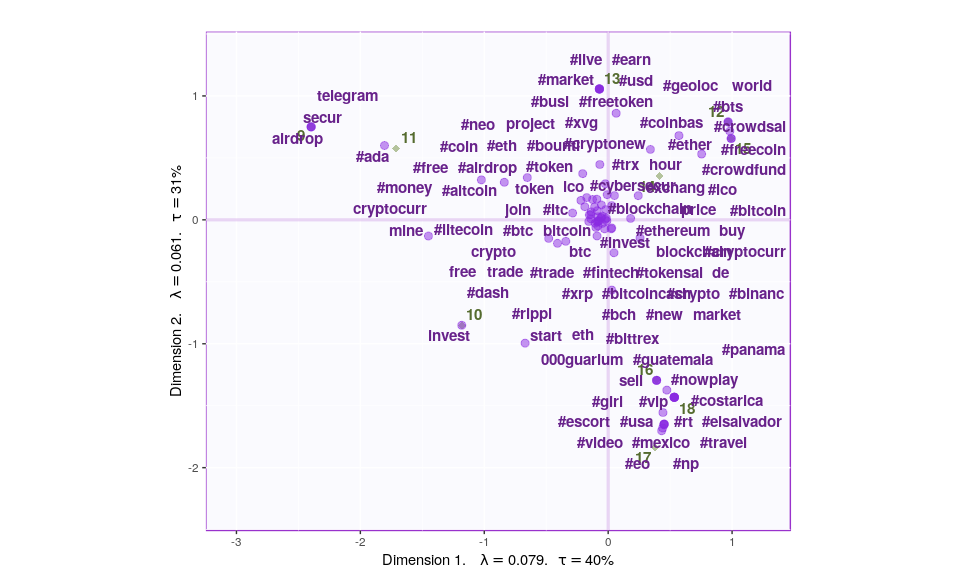
tweets\_eth\_supp <- supplementaryRows(tweets\_eth\_50,resCA.asym)  
asymMap\_supp <- createFactorMapIJ(tweets\_eth\_supp$fii,Fj,  
 #col.points.i = color4Authors,  
 #col.labels.i = color4Authors  
 )  
  
map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 #asymMap$I\_points +   
 asymMap$J\_labels +   
 asymMap$J\_points + labels4CA + #+ legend$zeMap\_dots + legend$zeMap\_text  
 asymMap\_supp$I\_points + asymMap\_supp$I\_labels  
  
print(map.IJ.asym)



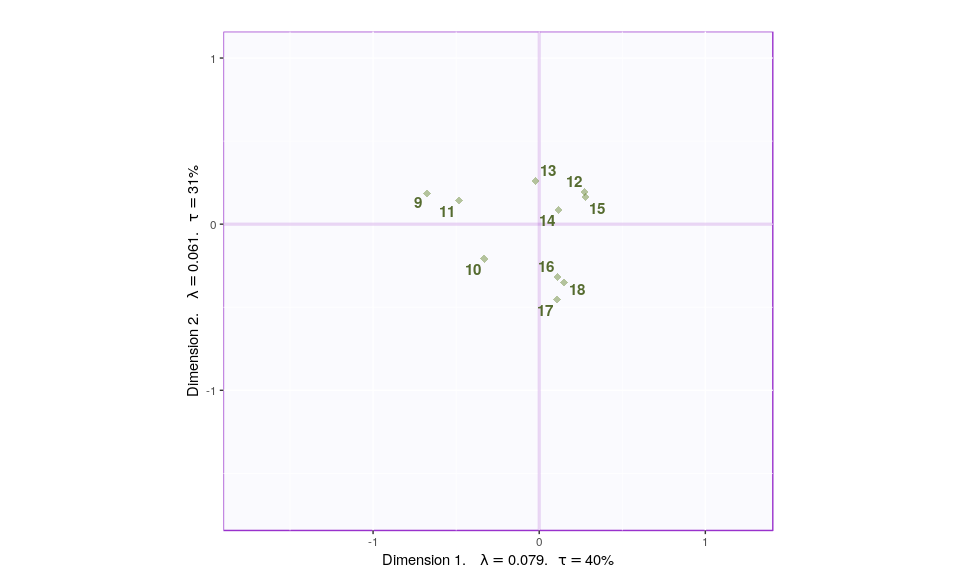
tweets\_bitcoin\_supp <- supplementaryRows(tweets\_bitcoin,resCA.asym)  
asymMap\_supp <- createFactorMapIJ(tweets\_bitcoin\_supp$fii,Fj,  
 #col.points.i = color4Authors,  
 #col.labels.i = color4Authors  
 )  
  
map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 #asymMap$I\_points +   
 asymMap$J\_labels +   
 asymMap$J\_points + labels4CA + #+ legend$zeMap\_dots + legend$zeMap\_text  
 asymMap\_supp$I\_points + asymMap\_supp$I\_labels  
  
print(map.IJ.asym)

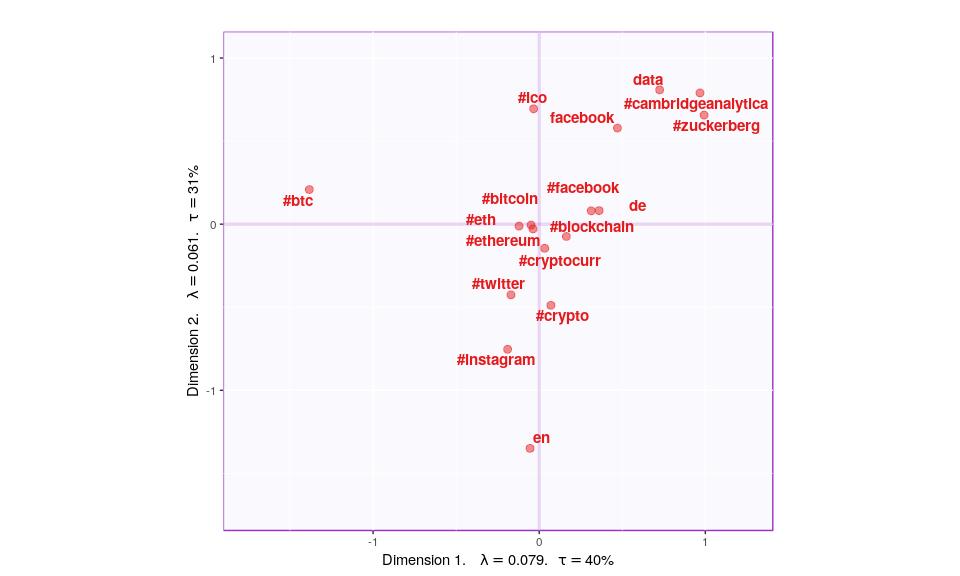


tweets\_bitcoin\_supp <- supplementaryRows(tweets\_bitcoin\_50,resCA.asym)  
asymMap\_supp <- createFactorMapIJ(tweets\_bitcoin\_supp$fii,Fj,  
 #col.points.i = color4Authors,  
 #col.labels.i = color4Authors  
 )  
  
map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 #asymMap$I\_points +   
 asymMap$J\_labels +   
 asymMap$J\_points + labels4CA + #+ legend$zeMap\_dots + legend$zeMap\_text  
 asymMap\_supp$I\_points + asymMap\_supp$I\_labels  
  
print(map.IJ.asym)



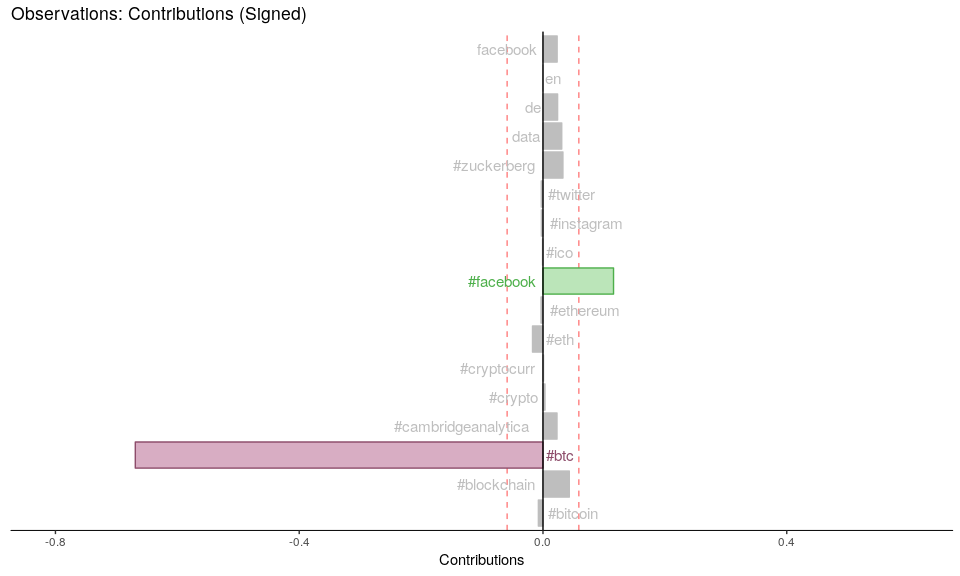
### Symmetric Plot





## Most Contributing Variables

PTCA4CATA::PrettyBarPlot2(ctr.I[,1],   
 threshold = 1 / NROW(ctr.I),   
 font.size = 4,   
 color4bar = gplots::col2hex(color4I),   
 color4ns = 'grey',   
 main = 'Observations: Contributions (Signed)',   
 ylab = 'Contributions', ylim = c(1.2\*min(ctr.I),  
 1.2\*max(ctr.I) ),   
 horizontal = FALSE )



PTCA4CATA::PrettyBarPlot2(ctr.J[,1],   
 threshold = 1 / NROW(ctr.J),   
 font.size = 4,   
 color4bar = gplots::col2hex(color4J),   
 color4ns = 'grey',   
 main = 'Observations: Contributions (Signed)',   
 ylab = 'Contributions', ylim = c(1.2\*min(ctr.J),  
 1.2\*max(ctr.J) ),   
 horizontal = FALSE )

