Report II.

František Kalvas

2022-03-18

## Introduction

Main focus of the previous report was on weight of individual variables and their blocks. Here I will focus more on some addenda – sensitivity of our model to extreme values of boundary (low, average of 0.1, instead of 0.22–0.34) and probability of speaking (high, average of 0.95, instead of 0.39–0.59) and comparison of 1D vs. 2D opinion. I will neglect here effect of simulation length: from previous report we know that there is some small effect – after thousands of steps sometimes groups merge into one or become more concentrated if agents were scattered after 365 steps, in some cases scattered agents become more concentrated in two opposing groups, which increases polarization, but on the average the processes decreasing polarization are more frequent than process increasing polarization, that’s why on the average polarization slightly decreases in the long running simulations.

## Packages etc.

library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(ggplot2)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(forcats)  
library(sjmisc)

##   
## Attaching package: 'sjmisc'

## The following object is masked from 'package:tidyr':  
##   
## replace\_na

# My own functon for renaming in Tidyverse  
prejmenuj = function(data, positions, new.names) {  
 names(data)[positions] = new.names  
 data  
}

## Loading data

Data are at <http://github.com/frantisek901/Spirals/Experiment>. Experiment is still running and I, FranČesko, from time to time actualize the \*.csv files at GitHub, then I run script experiment.R which loads the data. Later version probably finds better names for variables, but now, I use default names from NetLogo experiment.

Who is not interested in working with megabytes of \*.csv files, might use compiled \*.RData, there are two files: shortData.RData, which is main data file from experiments running only 365 steps in 2D opinion space, these data are extended by extra simulations with low size of small-world network neighborhood, very narrow boundary and high probability of speaking; and shortData1D.RData, which is same data (365 steps, extra narrow network neighborhood, extra narrow boundary, extra high probability of speaking), but in 1D opinion space.

For avoiding statistical artifact we sampling data – 1,000 observations from 2D data and 1,000 observations from 1D data.

Now we load and join these data:

load("shortData.RData")  
load("shortData1D.RData")  
load("shortData4D.RData")  
  
# We control the ratios of boundary and identity use  
df = sample\_n(res4D[res4D$`use\_identity?` & res4D$boundary > 0.2,], 900) %>%  
 add\_row(sample\_n(res4D[!res4D$`use\_identity?` & res4D$boundary > 0.2,], 900)) %>%   
 add\_row(sample\_n(res4D[res4D$`use\_identity?` & res4D$boundary < 0.2,], 300)) %>%  
 add\_row(sample\_n(res4D[!res4D$`use\_identity?` & res4D$boundary < 0.2,], 300)) %>%   
 add\_row(sample\_n(res[res$`use\_identity?` & res$boundary > 0.2,], 900)) %>%  
 add\_row(sample\_n(res[!res$`use\_identity?` & res$boundary > 0.2,], 900)) %>%  
 add\_row(sample\_n(res[res$`use\_identity?` & res$boundary < 0.2,], 300)) %>%  
 add\_row(sample\_n(res[!res$`use\_identity?` & res$boundary < 0.2,], 300)) %>%  
 add\_row(sample\_n(res1D[res1D$`use\_identity?` & res1D$boundary > 0.2,], 900)) %>%  
 add\_row(sample\_n(res1D[!res1D$`use\_identity?` & res1D$boundary > 0.2,], 900)) %>%   
 add\_row(sample\_n(res1D[res1D$`use\_identity?` & res1D$boundary < 0.2,], 300)) %>%  
 add\_row(sample\_n(res1D[!res1D$`use\_identity?` & res1D$boundary < 0.2,], 300))

## Regressions

On the two following pages, there are 4 regressions in 2 tables (I’m starting with stargazer, later I will produce better output, but for now…). The first table uses ESBG polarization measure, after 365 and 3650 steps, the second uses my normalized polarization measure after same number of steps.

##   
## =============================================================================================================================  
## Dependent variable:   
## ---------------------------------------------------------------------------------------------------------  
## ESBG\_365 normalized\_365   
## (1) (2) (3) (4)   
## -----------------------------------------------------------------------------------------------------------------------------  
## factor(opinions)2 -0.135\*\*\* -0.096\*\*\*   
## (0.004) (0.004)   
##   
## factor(opinions)4 -0.184\*\*\* -0.142\*\*\*   
## (0.005) (0.005)   
##   
## id\_threshold 0.475\*\*\* 0.471\*\*\* 0.557\*\*\* 0.555\*\*\*   
## (0.028) (0.024) (0.027) (0.025)   
##   
## `use\_identity?` 0.126\*\*\* 0.127\*\*\* 0.117\*\*\* 0.118\*\*\*   
## (0.003) (0.003) (0.003) (0.003)   
##   
## boundary -0.524\*\*\* -0.531\*\*\* -1.062\*\*\* -1.068\*\*\*   
## (0.018) (0.016) (0.018) (0.017)   
##   
## modevaguely-speak -0.061\*\*\* -0.065\*\*\* -0.086\*\*\* -0.088\*\*\*   
## (0.003) (0.003) (0.003) (0.003)   
##   
## `conformity-level` 0.020 0.004 0.018 0.006   
## (0.027) (0.023) (0.026) (0.024)   
##   
## `p-speaking-level` -0.047\*\*\* -0.014\* -0.039\*\*\* -0.012   
## (0.008) (0.007) (0.008) (0.008)   
##   
## `tolerance-level` 0.004 -0.013\*\* -0.001 -0.011\*   
## (0.007) (0.006) (0.007) (0.006)   
##   
## `p-random` 0.357\*\*\* -0.028 0.292\*\*\* -0.017   
## (0.017) (0.020) (0.017) (0.021)   
##   
## `n-neis` -0.0002\*\* -0.0001 0.00001 0.00002   
## (0.0001) (0.0001) (0.0001) (0.0001)   
##   
## Constant -0.076\*\*\* 0.130\*\*\* 0.088\*\*\* 0.246\*\*\*   
## (0.022) (0.020) (0.021) (0.020)   
##   
## -----------------------------------------------------------------------------------------------------------------------------  
## Observations 7,200 7,200 7,200 7,200   
## R2 0.332 0.493 0.479 0.553   
## Adjusted R2 0.331 0.493 0.479 0.552   
## Residual Std. Error 0.137 (df = 7190) 0.119 (df = 7188) 0.133 (df = 7190) 0.124 (df = 7188)   
## F Statistic 397.417\*\*\* (df = 9; 7190) 636.310\*\*\* (df = 11; 7188) 735.432\*\*\* (df = 9; 7190) 807.703\*\*\* (df = 11; 7188)  
## =============================================================================================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

WHOOOOOOOOOOOOOOAA! One more opinion dimension supresses polarization hugely! note that in 2D world the average polarization was around 0.15 (only in scenarios using identity!), so the effects of opinion dimension here means that in 1D world is the polarization doubled! We will construct exact graph for it later so we will se precise figures, but since now we know that opinion dimensions have effect! The logical step is to simulate same thing in 4D opinion space and see the change of polarization with increase of opinion dimensions.

But at least for now we have another interesting story – adding dimensions probably suppresses polarization. It seems that opinion dimensions are bridges, not walls, now it seems that introducing more dimensions is equivalent of building bridges. We don’t exactly simulate this, we now just simulate 1D and 2D worlds and we compare polarization in them. But bulding bridges hypothesis (adding new dimension/opinion/issue to the polarized debate) is testable and simulable – we prepare simulation for X dimensions, but then we let it start and run in X-1 dimensions and after 365 steps we increase dimensions back to X and let it run another 365 steps, and we store results/states at the start, 365 steps and at the end (730 steps). But firstly it is needed to explore effect of number of opinion dimensions – is it linear? still decreasing? or some strange kind of non-linear effect?

## Analysis of Variance Table  
##   
## Model 1: ESBG\_365 ~ id\_threshold + `use\_identity?` + boundary + mode +   
## `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 7190 135.04   
## 2 7188 102.45 2 32.587 1143.2 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Likelihood ratio test  
##   
## Model 1: ESBG\_365 ~ id\_threshold + `use\_identity?` + boundary + mode +   
## `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 11 4098.2   
## 2 13 5092.4 2 1988.4 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] ""

## [1] "BIC comparison: -1970.7"

## [1] ""

## Analysis of Variance Table  
##   
## Model 1: normalized\_365 ~ id\_threshold + `use\_identity?` + boundary +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 7190 127.89   
## 2 7188 109.85 2 18.044 590.37 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Likelihood ratio test  
##   
## Model 1: normalized\_365 ~ id\_threshold + `use\_identity?` + boundary +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 11 4294.0   
## 2 13 4841.5 2 1095 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] ""

## [1] "BIC comparison: -1077.3"

## [1] ""

### Note:

1. Variables mode:vaguely-speak and use\_identity? are binary, n-neis is measured on scale 1–64, and all other variables (id\_threshold, boundary etc.) are measured on scale 0–1.

### Note on polarization measures:

Please notice, that both polarization measures differ in effect of variable boundary on them. It happened when I introduced just for sure very low level of boundary variable (0.1 besides standard margin 0.22–0.34). From previous analyses I know that low value of this variable has polarization effect, so I was wandering, why Genetic algorithms (GAs) find best fitness (i.e., highest values of polarization for sum of both measures) for boundary == 0.28 and I was a bit unsure whether this relatively wider value is not mistake. So I introduced also value 0.1 for variable boundary and I have been running simulations since Friday.

Narrow boundary (0.1) very probably produces not two, but several groups, probably all relatively dense. Normalized polarization is sensitive to density (i.e. tight aggregation around group’s mean) and distance of groups, so if the groups are evenly distributed in opinion space, it produces relatively high normalized polarization. On the other hand, ESBG is very sensitive to distributions in two groups of equal sizes, their density (i.e. tight aggregation around goup’s mean) and distance. When we receive 4 or 5 dense groups, evenly distributed over opinion space, ESBG assesses it as low polarization.

That is why both measures differ regarding estimation of effect of narrow boundaries and why GAs find wider boundaries as part of most polarizing set of parameters. During GAs I used sum of both polarization measures as fitness parameter. With value of boundary 0.28 and right combination of other parameters we receive high polarization in both measures, with narrow boundary we receive higher polarization through normalized measure, but much weaker through ESBG measure, so when we sum both measures, we receive higher combined polarization with wider boundaries than with narrow ones.

I also introduced high value of probability of speaking, 0.95, just to be sure this variable makes really no difference – but here the previous results are confirmed, now even with such a high values, probability of speaking still makes no difference, so if this assurance will be confirmed with the rest of extra simulations (now are done 40 sets out of 150, i.e. now we are on 25%, but we might be quite sure…), then in further explorations we might set probability of speaking to some constant value – may be to 0.49 which was found by GA, or to 1, but the value 0.49 seems more realistic to me, since not everyone talks always.

## Checking the meaning of ‘boundary’ and ‘probability of speaking’

### Boundary

## Analysis of Variance Table  
##   
## Model 1: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 7189 118.17   
## 2 7188 102.45 1 15.721 1103 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Likelihood ratio test  
##   
## Model 1: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 12 4578.5   
## 2 13 5092.4 1 1027.8 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] ""

## [1] "BIC comparison: -1018.9"

## [1] ""

## Analysis of Variance Table  
##   
## Model 1: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 7189 173.43   
## 2 7188 109.85 1 63.584 4160.7 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Likelihood ratio test  
##   
## Model 1: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## mode + `conformity-level` + `p-speaking-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 12 3197.4   
## 2 13 4841.5 1 3288.2 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] ""

## [1] "BIC comparison: -3279.3"

## [1] ""

YES!! So, Boundary is significant, BICs -12 or -60 tell us that boundary improves model really significantly! I also made model comparison on data without average boundary = 0.1, and it shows that meaning of this variable, i.e. its effect, decreases significantly. It means that including simulations with low value of boundary is important.

### Probability of speaking

## Analysis of Variance Table  
##   
## Model 1: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 7189 102.50   
## 2 7188 102.45 1 0.048695 3.4164 0.06459 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Likelihood ratio test  
##   
## Model 1: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: ESBG\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 12 5090.7   
## 2 13 5092.4 1 3.4213 0.06436 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## [1] ""

## [1] "BIC comparison: 5.5"

## [1] ""

## Analysis of Variance Table  
##   
## Model 1: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 7189 109.89   
## 2 7188 109.85 1 0.036539 2.3909 0.1221

## Likelihood ratio test  
##   
## Model 1: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `tolerance-level` +   
## `p-random` + `n-neis`  
## Model 2: normalized\_365 ~ factor(opinions) + id\_threshold + `use\_identity?` +   
## boundary + mode + `conformity-level` + `p-speaking-level` +   
## `tolerance-level` + `p-random` + `n-neis`  
## #Df LogLik Df Chisq Pr(>Chisq)  
## 1 12 4840.3   
## 2 13 4841.5 1 2.3945 0.1218

## [1] ""

## [1] "BIC comparison: 6.5"

## [1] ""

YES!!! We might be sure that probability of speaking is not important, so we might neglect it.

## Graph

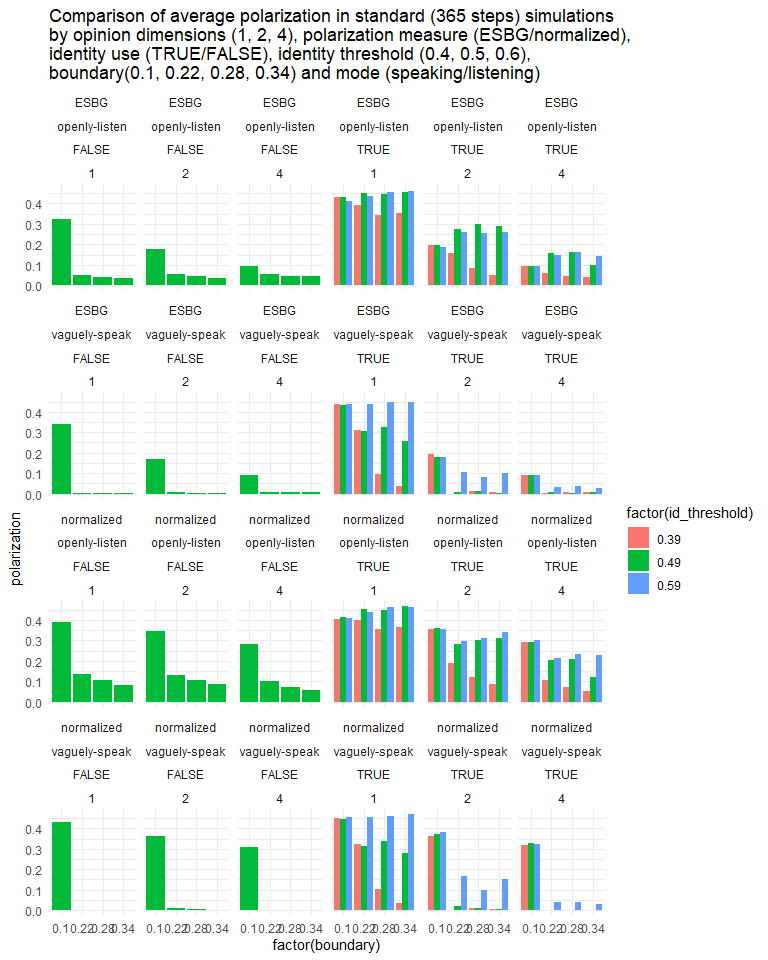
Now, let’s show our results graphically!

### Drawing graphs

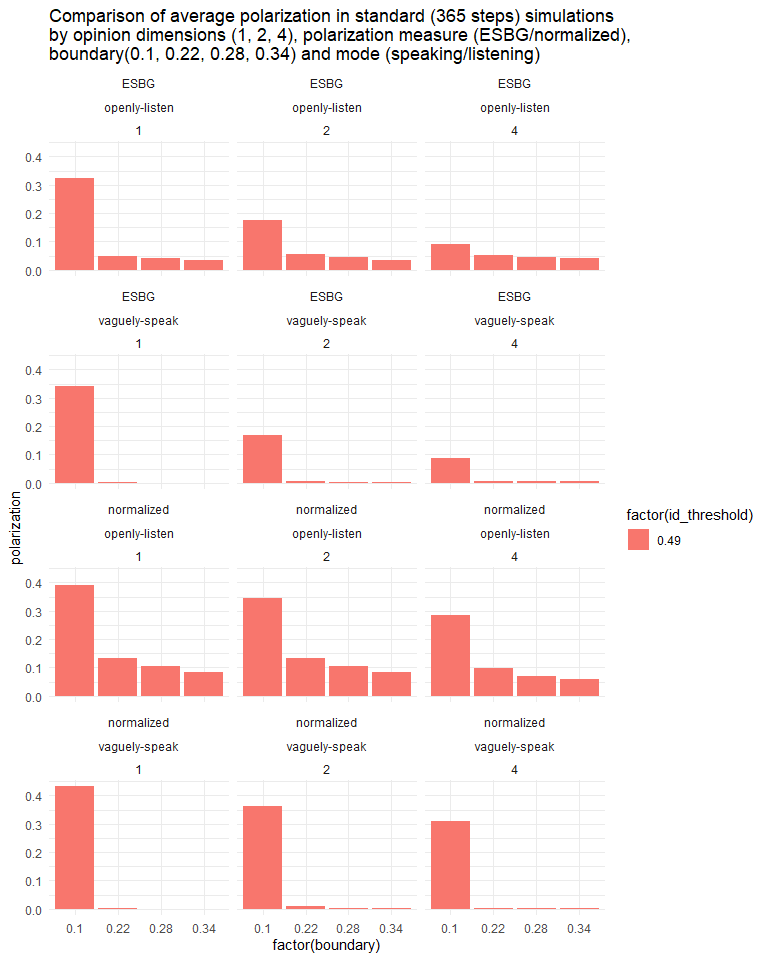
dfg = rename(df, ESBG = ESBG\_365, normalized = normalized\_365, identity = `use\_identity?`) %>%   
 group\_by(opinions, identity, id\_threshold, boundary, mode) %>%   
 summarise(ESBG = mean(ESBG), normalized = mean(normalized)) %>%   
 pivot\_longer(cols = c(ESBG, normalized), names\_to = "polarization\_measure", values\_to = "polarization")

## `summarise()` has grouped output by 'opinions', 'identity', 'id\_threshold', 'boundary'. You can override using the `.groups` argument.

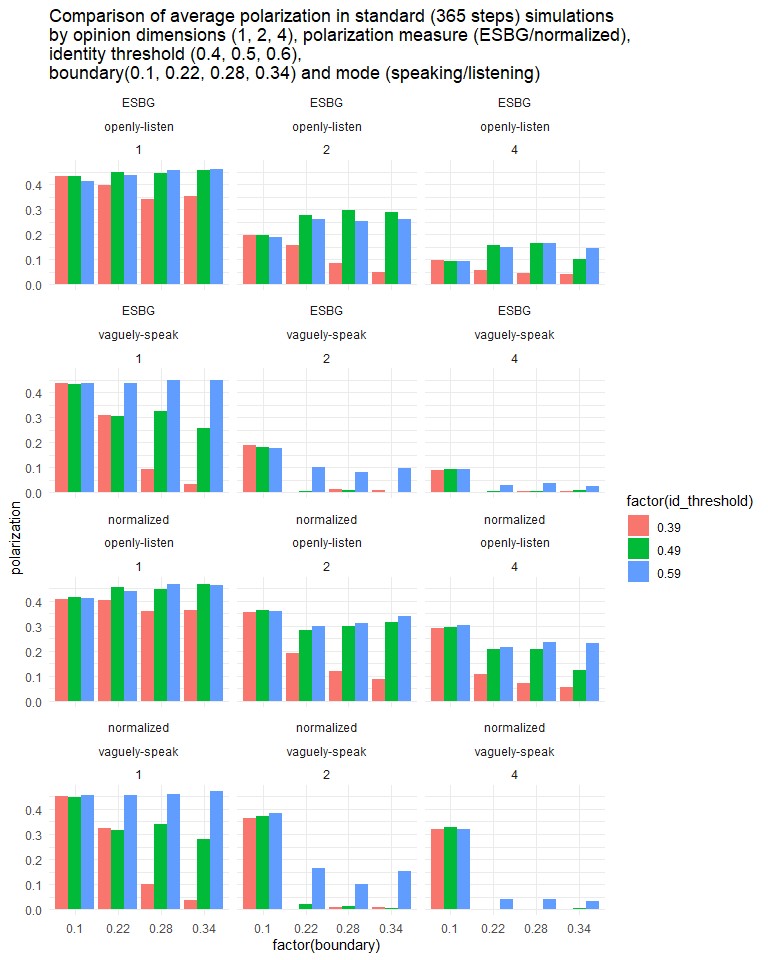
ggplot(dfg, aes(x = factor(boundary), y = polarization, fill = factor(id\_threshold))) +  
 facet\_wrap(vars(polarization\_measure, mode, identity, opinions), ncol=6) +  
 geom\_col(position = position\_dodge()) +  
 labs(title = "Comparison of average polarization in standard (365 steps) simulations\nby opinion dimensions (1, 2, 4), polarization measure (ESBG/normalized), \nidentity use (TRUE/FALSE), identity threshold (0.4, 0.5, 0.6), \nboundary(0.1, 0.22, 0.28, 0.34) and mode (speaking/listening)") +  
 theme\_minimal()



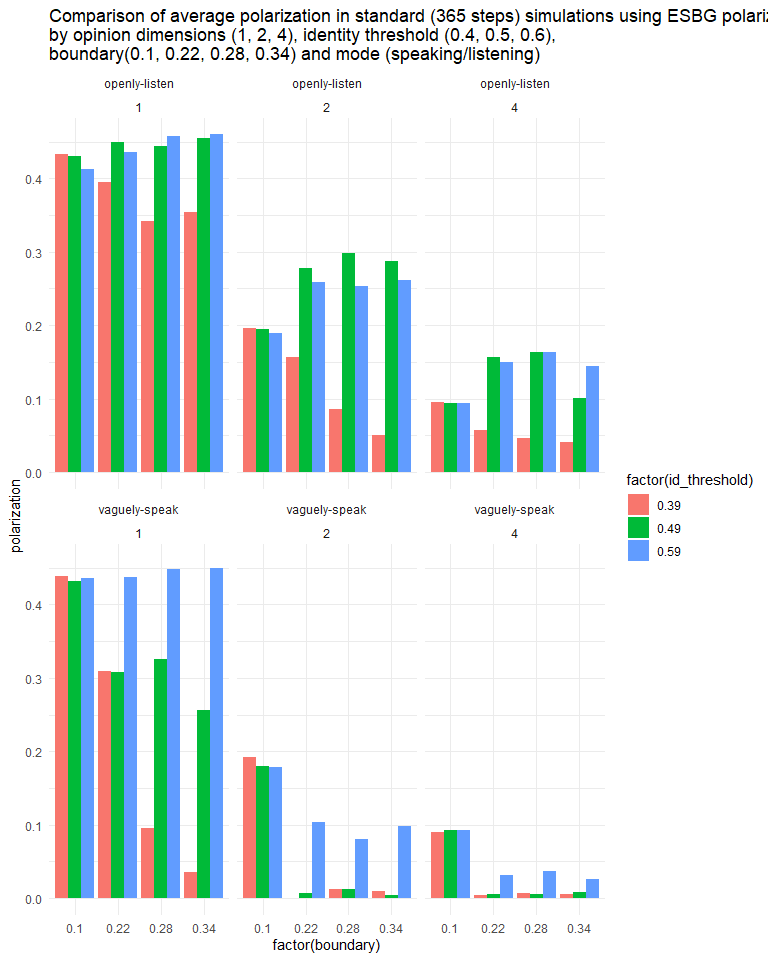
dfg %>% filter(!identity) %>%   
 ggplot(aes(x = factor(boundary), y = polarization, fill = factor(id\_threshold))) +  
 facet\_wrap(vars(polarization\_measure, mode, opinions), ncol=3) +  
 geom\_col(position = position\_dodge()) +  
 labs(title = "Comparison of average polarization in standard (365 steps) simulations\nby opinion dimensions (1, 2, 4), polarization measure (ESBG/normalized), \nboundary(0.1, 0.22, 0.28, 0.34) and mode (speaking/listening)") +  
 theme\_minimal()



dfg %>% filter(identity) %>%   
 ggplot(aes(x = factor(boundary), y = polarization, fill = factor(id\_threshold))) +  
 facet\_wrap(vars(polarization\_measure, mode, opinions), ncol=3) +  
 geom\_col(position = position\_dodge()) +  
 labs(title = "Comparison of average polarization in standard (365 steps) simulations\nby opinion dimensions (1, 2, 4), polarization measure (ESBG/normalized), \nidentity threshold (0.4, 0.5, 0.6), \nboundary(0.1, 0.22, 0.28, 0.34) and mode (speaking/listening)") +  
 theme\_minimal()



dfg %>% filter(identity, polarization\_measure == "ESBG") %>%   
 ggplot(aes(x = factor(boundary), y = polarization, fill = factor(id\_threshold))) +  
 facet\_wrap(vars(mode, opinions), ncol=3) +  
 geom\_col(position = position\_dodge()) +  
 labs(title = "Comparison of average polarization in standard (365 steps) simulations using ESBG polarization measure\nby opinion dimensions (1, 2, 4), identity threshold (0.4, 0.5, 0.6), \nboundary(0.1, 0.22, 0.28, 0.34) and mode (speaking/listening)") +  
 theme\_minimal()



OK, results for 1D and boundary = 0.1 are not simulated yet, so we have to wait a while…