Report for Ashley

František Kalvas

2022-03-07

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)


# 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, Francesco, 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, these data are extended by extra simulations with low size of small-world network neighborhood; and longData.RData, which is additional data file from experiments running 3650 steps – thanks to it we might test the effect of simulation length.

Now we load these data:

```
load("shortData.RData")
load("longData.RData")
```

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                     ESBG_3650
##                                   (1)                           (2)
## -------------------------------------------------------------------------------
## id_threshold                   0.557***                      0.706***
##                                (0.003)                       (0.006)
##
## `use_identity?`                0.102***                      0.100***
##                                (0.0005)                      (0.001)
##
## boundary                      -0.137***                     -0.089***
##                                (0.004)                       (0.005)
##
## modevaguely-speak             -0.133***                     -0.068***
##                                (0.0004)                      (0.0005)
##
## `tolerance-level`             -0.016***                     -0.011***
##                                (0.001)                       (0.001)
##
## `p-speaking-level`            -0.013***                     -0.009***
##                                (0.002)                       (0.003)
##
## `conformity-level`            -0.045***                     -0.013***
##                                (0.003)                       (0.003)
##
## `p-random`                      -0.001                        0.002
##                                (0.006)                       (0.007)
##
## `n-neis`                      -0.001***                     -0.0001**
##                               (0.00001)                     (0.00004)
##
## Constant                      -0.072***                     -0.242***
##                                (0.003)                       (0.005)
##
## -------------------------------------------------------------------------------
## Observations                   227,073                        64,885
## R2                              0.491                         0.505
## Adjusted R2                     0.491                         0.505
## Residual Std. Error      0.095 (df = 227063)           0.058 (df = 64875)
## F Statistic         24,298.600*** (df = 9; 227063) 7,350.408*** (df = 9; 64875)
## ===============================================================================
## Note:                                               *p<0.1; **p<0.05; ***p<0.01
```

```
##
## ================================================================================
##                                          Dependent variable:
##                     ------------------------------------------------------------
##                             normalized_365                normalized_3650
##                                  (1)                            (2)
## --------------------------------------------------------------------------------
## id_threshold                  0.711***                       0.882***
##                                (0.003)                        (0.007)
##
## ‘use_identity?‘                0.092***                      0.100***
##                                (0.0005)                       (0.001)
##
## boundary                      -0.200***                      -0.172***
##                                (0.004)                        (0.005)
##
## modevaguely-speak             -0.166***                      -0.123***
##                                (0.0004)                      (0.0005)
##
## ‘tolerance-level‘             -0.023***                      -0.024***
##                                (0.001)                        (0.001)
##
## ‘p-speaking-level‘            -0.018***                      -0.010***
##                                (0.002)                        (0.003)
##
## ‘conformity-level‘            -0.059***                        0.005
##                                (0.003)                       (0.003)
##
## ‘p-random‘                      -0.001                        -0.003
##                                (0.006)                        (0.007)
##
## ‘n-neis‘                      -0.001***                       -0.0001
##                               (0.00001)                      (0.00004)
##
## Constant                      -0.057***                      -0.244***
##                                (0.003)                        (0.005)
##
## --------------------------------------------------------------------------------
## Observations                   227,073                         64,885
## R2                              0.556                          0.625
## Adjusted R2                     0.556                          0.625
## Residual Std. Error      0.097 (df = 227063)            0.063 (df = 64875)
## F Statistic         31,544.360*** (df = 9; 227063) 11,993.320*** (df = 9; 64875)
## ================================================================================
## Note:                                                *p<0.1; **p<0.05; ***p<0.01
```

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.

2. I check the problem of `use_identity?` – I estimated same regression model on sub-sample of simulation with `use_identity?==TRUE`, naturally, effect of mere `use_identity?` is not estimable, but good news is that effect of `id_threshold` is completely same (OK, up to 5th decimal place).

3. Just for curiosity I estimated the model for subsample `use_identity?==FALSE`, I was surprised that all effects were roughly by one order lower (10^{-1}).

Graphs

Sampling

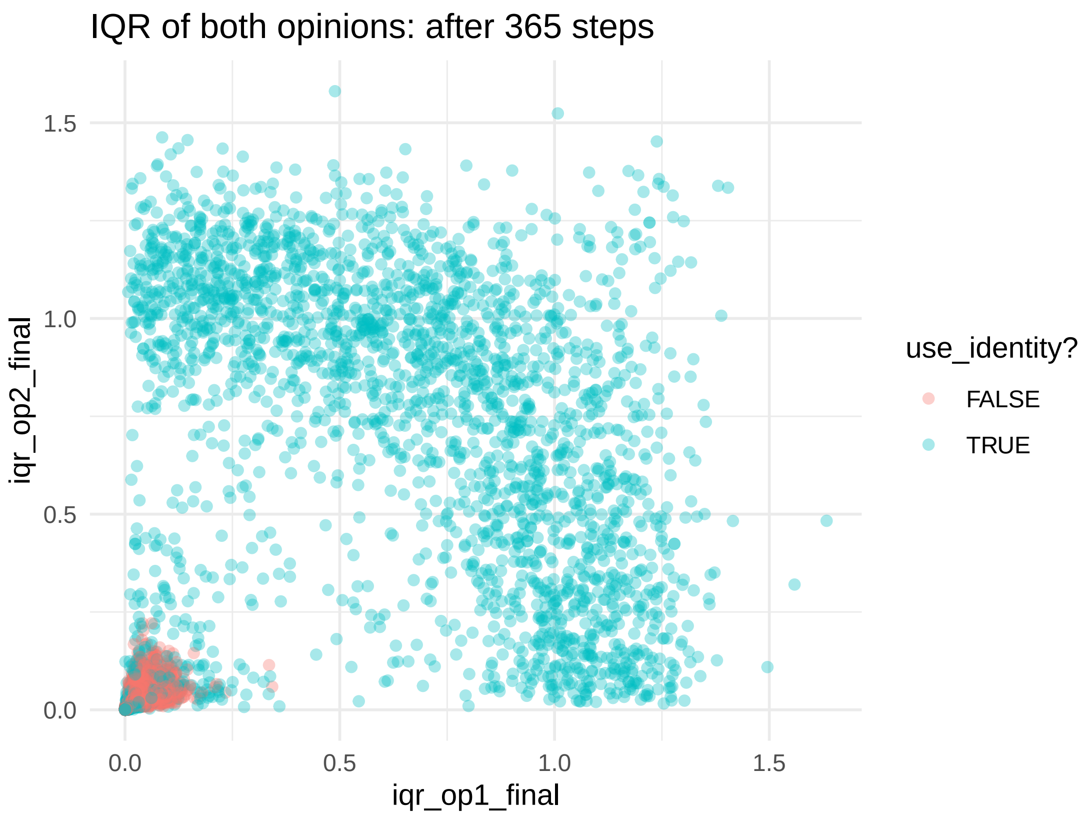
I produced graphs after some random sampling. Both files standard (365 steps) and long (3650 steps) are huge with many thousands of observations. So I created two samples, each of 10,000 observations – 5,000 simulations using identity, 5,000 not using identity.

```
res_sample = sample_n(res[res$`use_identity?`,], 5000) %>%
  add_row(sample_n(res[!res$`use_identity?`,], 5000)) %>%
  sample_n(10000)

long_sample = sample_n(long[long$`use_identity?`,], 5000) %>%
  add_row(sample_n(long[!long$`use_identity?`,], 5000)) %>%
  sample_n(10000)
```

Inter-quartile range

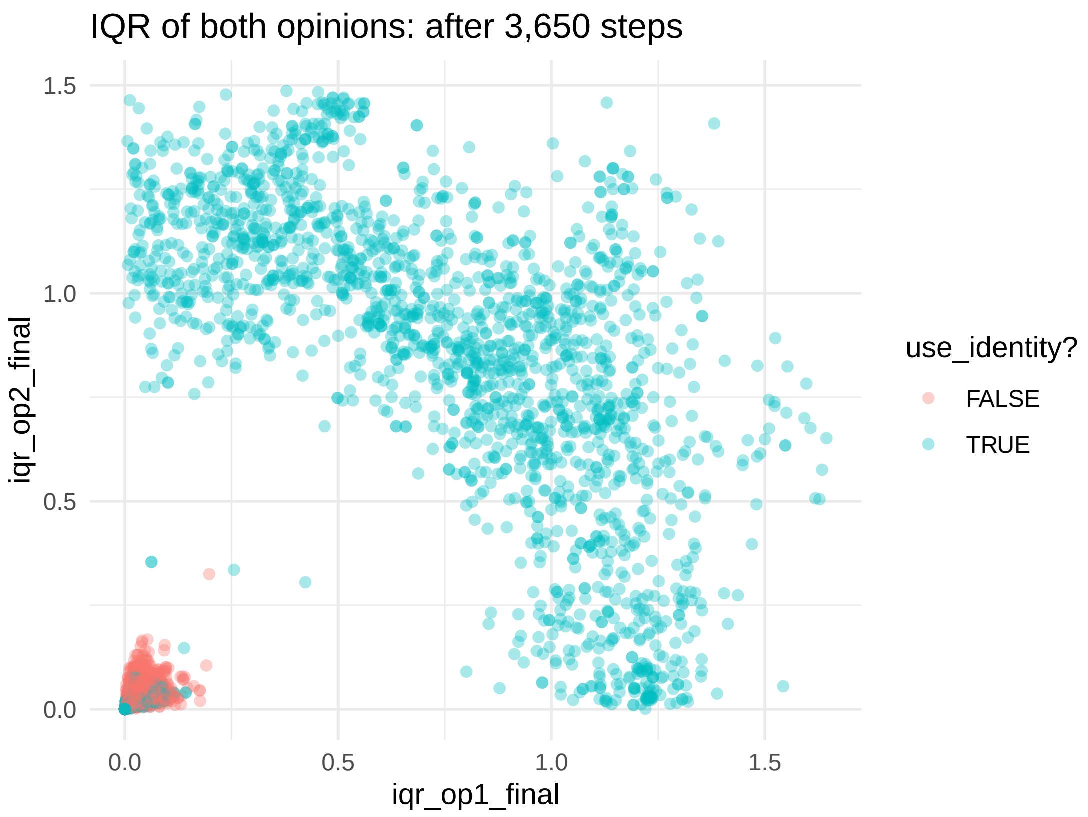
Here we look at depiction of distribution of interquartile range of both opinions. The first graph is made from standard (365 steps) data, the second from long (3,650 steps) data.

```
res_sample %>%
  ggplot(aes(x = iqr_op1_final, y = iqr_op2_final, col = `use_identity?`)) +
  geom_point(alpha = 0.35) +
  labs(title = "IQR of both opinions: after 365 steps",
       caption = "Sample of 5,000 simulations using udentity and 5,000 simulations not using identity.")
  theme_minimal()
```



ample of 5,000 simulations using udentity and 5,000 simulations not using identity.

```
long_sample %>%
  ggplot(aes(x = iqr_op1_final, y = iqr_op2_final, col = `use_identity?`)) +
  geom_point(alpha = 0.35) +
  labs(title = "IQR of both opinions: after 3,650 steps",
       caption = "Sample of 5,000 simulations using udentity and 5,000 simulations not using identity."
  theme_minimal()
```



ample of 5,000 simulations using udentity and 5,000 simulations not using identity.

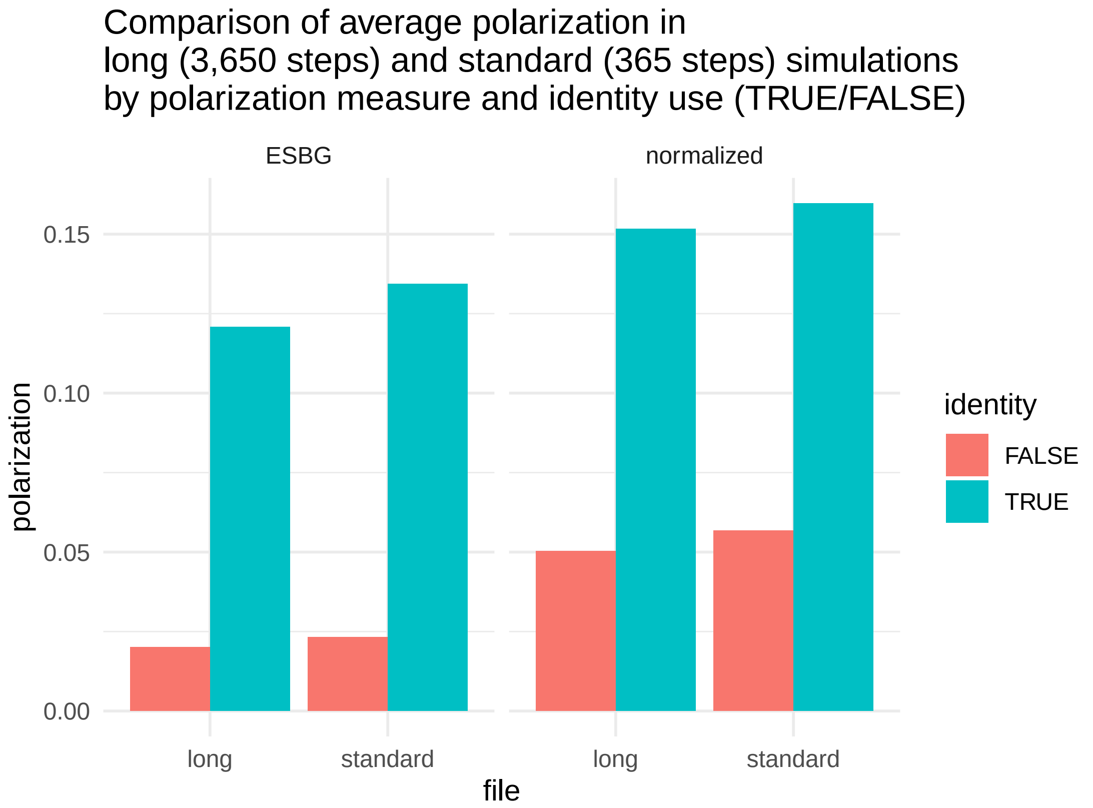
For me the basic logic is same in both graphs: some part of simulations ends up with consensus, mainly its simulations not using identity (red dots). Simulation using identity (turquoise dots) sometimes ends up with consensus as well, but also frequently ends up polarized, which is reflected by turquoise 'perimeter'. It seems to me that this basic logic – identity use = perimeter of discord – is same regardless the length of simulation.

But different is cleanness of this pattern. In long data (3,650 steps) it is very clear and there are almost no observations between 'red consensus dot' in left down corner and 'turquoise discord perimeter'. In standard data (365 steps) there are some observations and the perimeter seems fatter. The result is obvious: some standard simulations (365 steps) ended too early, because their 'longer twins' moved from 'discord perimeter' or space in between to 'concensus dot'. So, let's check the differences in polarization between standard and long data:

```
df = res %>% mutate(file = "standard") %>%
  rename(ESBG = ESBG_365, normalized = normalized_365, identity = `use_identity?`) %>%
  add_row(long %>% mutate(file = "long") %>%
            rename(ESBG = ESBG_3650, normalized = normalized_3650, identity = `use_identity?`)) %>%
  group_by(file, identity) %>%
  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 ’file’. You can override using the ‘.groups‘ argument.
```

```
ggplot(df, aes(x = file, y = polarization, fill = identity)) +
  facet_wrap(vars(polarization_measure)) +
  geom_col(position = position_dodge()) +
  labs(title = "Comparison of average polarization in \nlong (3,650 steps) and standard (365 steps) simu
  theme_minimal()
```



Full aggregated sample.

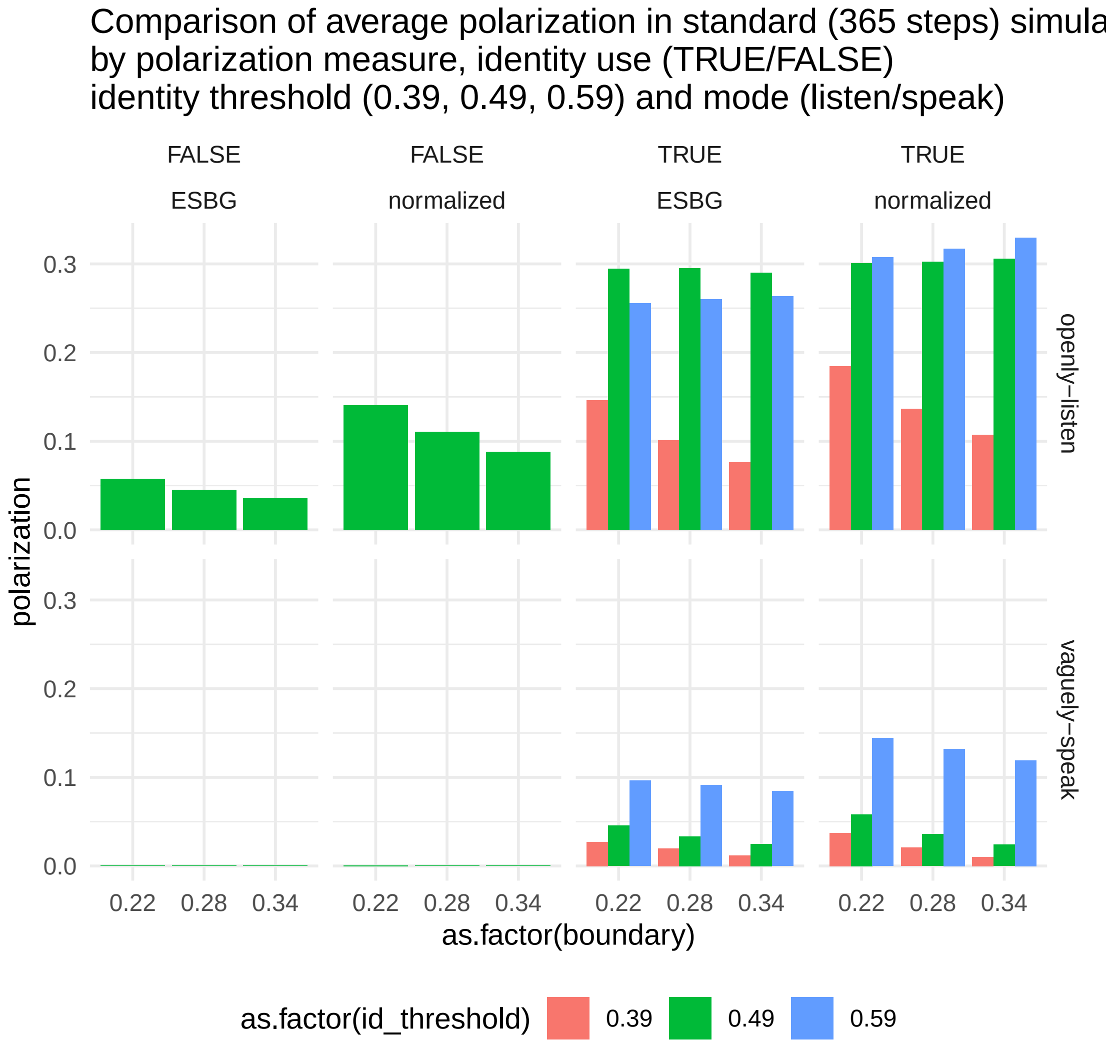
We see that long (3,650 steps) simulation are tiny slightly less polarized than short (365 steps) ones, i.e. on the average, the polarization in further more than 3,000 steps slightly decreases from initial value. We also see that `normalized` measure shows slightly higher polarization than `ESBG`. So, we might be quite confident that the length of simulation doesn't spoil the results that much – since there is some tiny differences in aggregate results, it makes sense to do further analyses on individual level, i.e. level of individual simulation, and compute and plot how many times polarization increases from 365th to 3,650th step and how much, but for now we see that after 365 steps we received almost same picture as after 3,650 steps.

But the main difference is obviously whether we use identity process or not – regardless the level of identity threshold (but note that we simulate it only for values 0.39, 0.49, 0.59, since it is so important parameter, we now could look at it in more detail). So, let's look now graphically in same way on data, as we did in regression tables:

```
df = res %>% mutate(file = "standard") %>%
  rename(ESBG = ESBG_365, normalized = normalized_365, identity = `use_identity?`) %>%
  add_row(long %>% mutate(file = "long") %>%
            rename(ESBG = ESBG_3650, normalized = normalized_3650, identity = `use_identity?`)) %>%
  group_by(file, id_threshold, identity, 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 ’file’, ’id_threshold’, ’identity’, ’boundary’. You can override
```

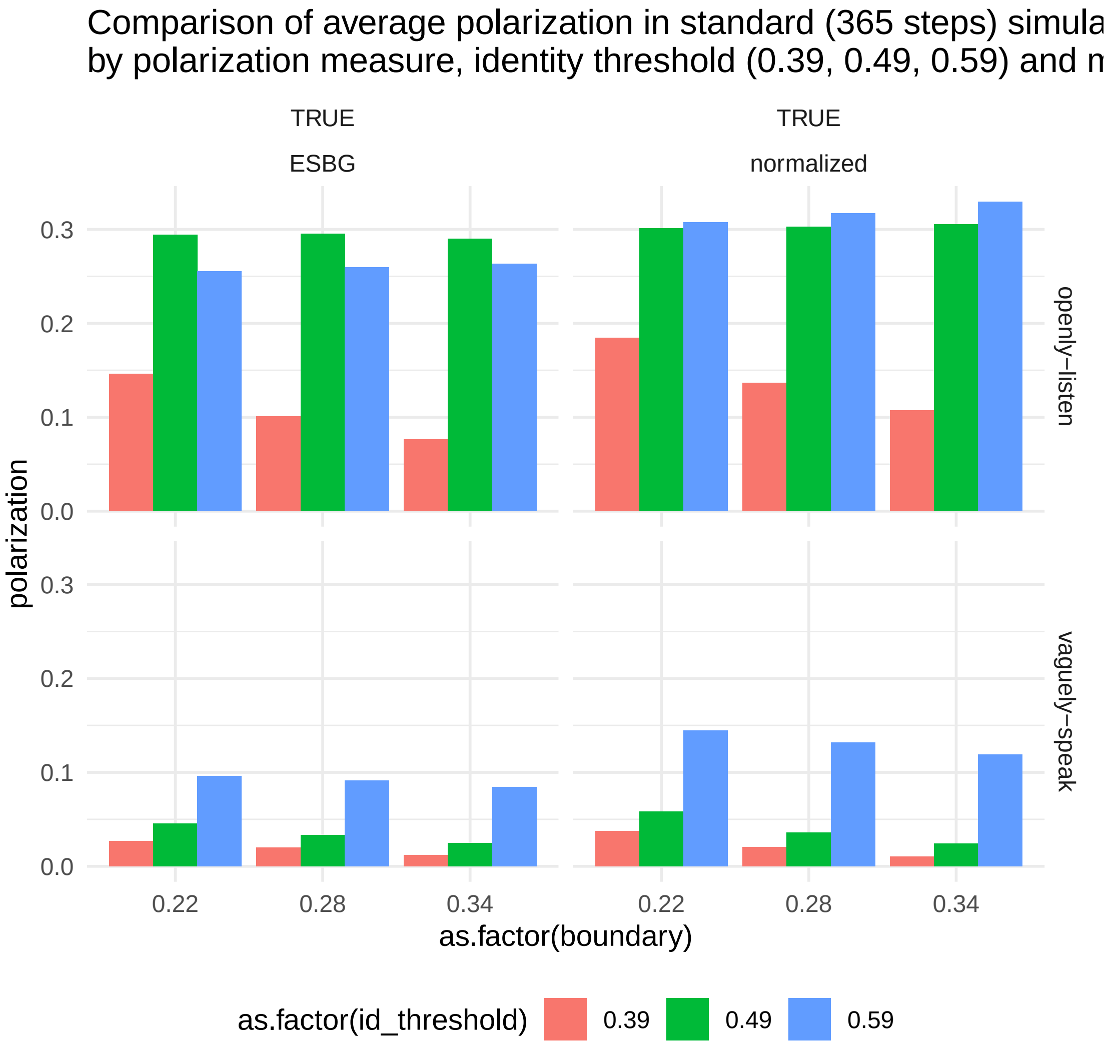
```
# `tolerance-level`             -0.024***                     -0.025***
#                                (0.001)                        (0.001)
#
# `p-speaking-level`            -0.018***                     -0.011***
#                                (0.002)                        (0.003)
#
# `conformity-level`            -0.056***                     0.011***
                               # (0.003)                        (0.003)

df %>% filter(file == "standard") %>%
  ggplot(aes(fill = as.factor(id_threshold), y = polarization, x = as.factor(boundary))) +
  facet_grid(cols = vars(identity, polarization_measure), rows = vars(mode)) +
  geom_col(position = position_dodge()) +
  labs(title = "Comparison of average polarization in standard (365 steps) simulations\nby polarization
  theme_minimal()  +
  theme(legend.position = "bottom")
```



Full aggregated sample.

Again same graph, just for better view only simulations using identity.

```
df %>% filter(file == "standard", identity) %>%
  ggplot(aes(fill = as.factor(id_threshold), y = polarization, x = as.factor(boundary))) +
  facet_grid(cols = vars(identity, polarization_measure), rows = vars(mode)) +
  geom_col(position = position_dodge()) +
  labs(title = "Comparison of average polarization in standard (365 steps) simulations\nby polarization
  theme_minimal() +
  theme(legend.position = "bottom")
```



Full aggregated sample.

In previous graph we saw that while some polarisation might happen even without using identity (especially with narrower boundaries), more polarized simulations on average are that using identity. Effect of identity threshold is non-linear: in simulations with ‘openly listen’ mode the main polarization increase is between 0.39 and 0.49 values, in mode ‘vaguely speak’ between values 0.49 and 0.59 (but generally, the later mode is less polarized). It is also interesting, that in mode ‘vaguely speak’ with boundary widening the polarization always decreases, but in mode ‘openly listen’ this happens only for the lowest identity threshold value (0.39), for other threshold values (0.49, 0.59) the polarization stays same with widening of boundary or even very slightly increases!

The last result is very surprising – Hegselmann-Krause model usually finds overall concensus and avoids polarization with wider boundary, it’s one of basic results. But when we introduce identity, then this old true changes or is contingent on simulation mode (speaking/listening) and identity threshold. The classical HK findings still hold true, but only for ‘vaguely speak’ mode and low identity threshold values.