Структурно-тематическое моделирование

library('stm')

Ольга Силютина,
Высшая Школа Экономики, ВКонтакте

Что сегодня узнаем?

- В чем особенности модели
- Как подготовить данные
- Как определиться с количеством тем
- Как можно визуализировать и интерпретировать результаты
- Дополнительные кейсы

Особенности структурно-тематического моделирования

STM - генеративная модель частотности слов

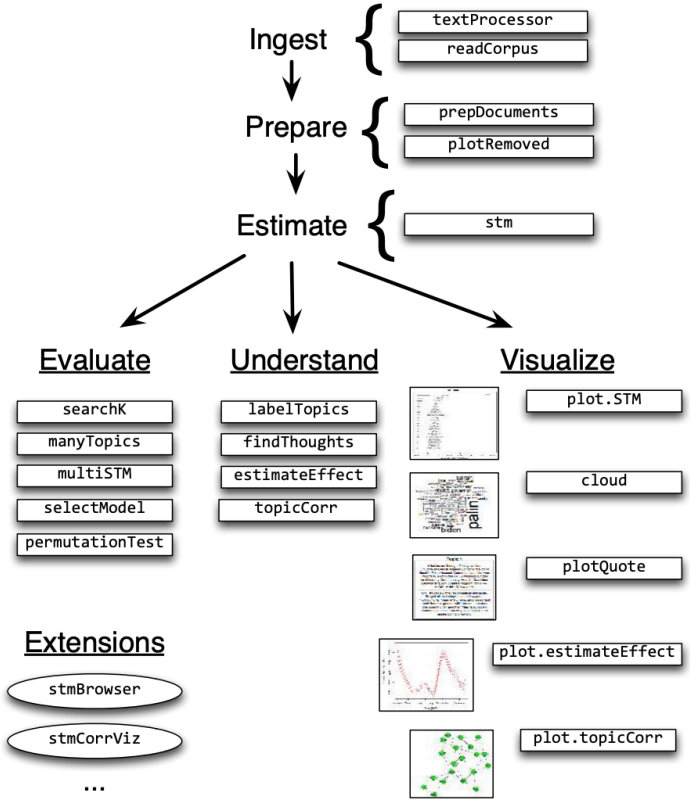
преимущества:

- корреляция между темами (~~распределение Дирихле~~ -> логнормальное распределение)
- статистическая проверка гипотез об эффекте ковариатов (GLM)
- метрики для выбора количества тем

недостатки:

- снижение качества результатов при использовании большого количества ковариатов
- все еще нужна указывать количество желаемых тем

Основные функции пакета stm



Подгрузим данные

```
library(readr)
library(dplyr)
library(ggplot2)
library(stm)
library(tm)
library(stringr)
library(igraph)
library(ggraph)
```

```
df_s <- read_csv('/Users/o.silutina/Downloads/intreg_data.csv', locale = locale(encoding = 'utf-8'))
df_s$years <- as.numeric(df_s$years)

# расширяем словарь стоп-слов
stopslova <- enc2utf8(c(stopwords("ru"), "который", "это", "этот",
                        "что", "быть", "для", "весь", "как", "при",
                        "свой", "только", "год"))
Encoding(stopslova) <- "UTF-8"
stopslova <- str_pad(stopslova, 40, "both")
stopslova <- str_replace_all(stopslova, "\\s+", " ")

for (slovo in stopslova) {
  df_s[[1]] = str_replace_all(df_s[[1]], slovo, "")
}
```

Подготавливаем данные для модели

```
set.seed(1)
df_s = as.data.frame(df_s)
processed <- textProcessor(df_s$lem_tex, metadata = df_s, stem=F, removestopwords = TRUE, language = "ru")
```

```
## Building corpus...
## Converting to Lower Case...
## Removing punctuation...
## Removing stopwords...
## Removing numbers...
## Creating Output...
```

```
out <- prepDocuments(processed$documents, processed$vocab, processed$meta)
```

```
## Removing 11831 of 17908 terms (11831 of 55095 tokens) due to frequency
## Removing 4 Documents with No Words
## Your corpus now has 2577 documents, 6077 terms and 43264 tokens.
```

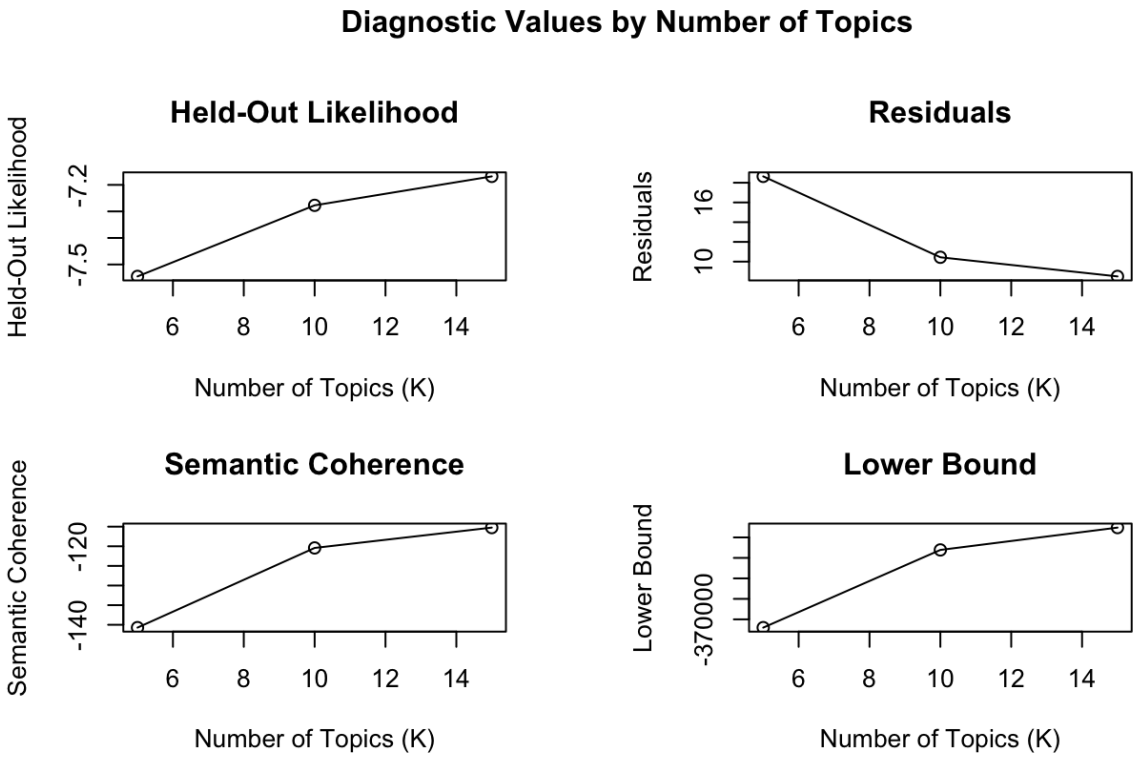
```
docs <- out$documents
vocab <- out$vocab
meta  <-out$meta
```

Находим подходящее количество тем

```
# прогоняем несколько моделей для выбора количества тем
set.seed(1)
storage <- searchK(out$documents,
                   out$vocab,
                   K = c(5, 10, 15),
                   prevalence =~ Polit_news_related + s(years),
                   data = out$meta)


storage$results
```

```
##    K   exclus    semcoh   heldout  residual     bound    lbound em.its
## 1  5 8.928040 -140.7214 -7.544977 18.641999 -374042.7 -374037.9      5
## 2 10 9.397799 -120.4211 -7.277118 10.440585 -336125.2 -336110.1    496
## 3 15 9.548202 -115.2389 -7.167970  8.499551 -325230.8 -325202.9    500
```

```
plot(storage)
```



Запускаем модель

```
set.seed(1)
poliblogPrevFit15 <- stm(documents = out$documents, vocab = out$vocab,
                         K = 15, prevalence =~ Polit_news_related + s(years),
                         max.em.its = 100, data = out$meta,
                         init.type = "Spectral")
```

Содержание тем

```
plot(poliblogPrevFit15)
```

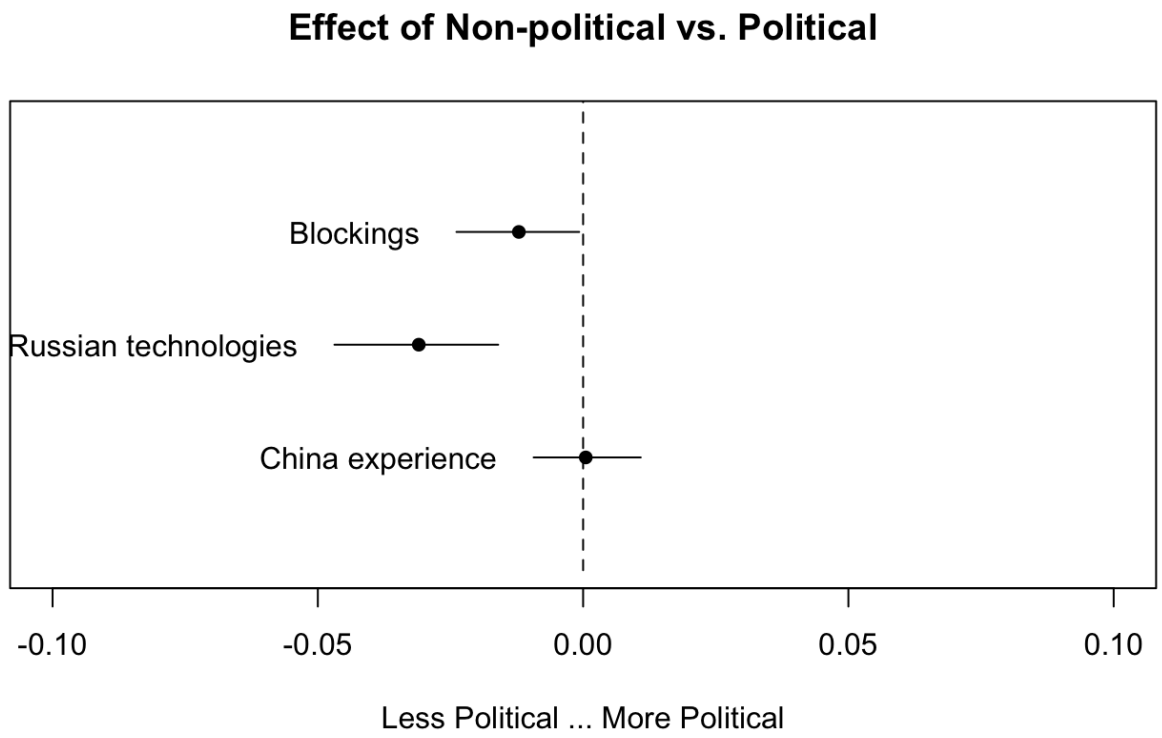
Называем темы

```
predict_topics <- estimateEffect(1:15 ~ Polit_news_related + s(years),
                                 poliblogPrevFit15,
                                 meta = out$meta,
                                 uncertainty = "Global")

topicNames <- c('Blockings', 'International experience', 'Responsible organization', 'Legislation',
                'Russian technologies', 'Trading', 'Black list', 'New opinions', 'Censorship',
                'Roskomnadzor', 'Federal net', 'President', 'China experience',
                'Ministry of Communications', 'Ecology')

top_vec <- c(1:15)
names(top_vec) <- topicNames
fift <- as.data.frame(cbind(top_vec, names(top_vec)))
colnames(fift)[2] <- "names"
colnames(fift)[1] <- "topic"
```

Стандартная визуализация эффекта ковариата

```
plot(predict_topics, covariate = "Polit_news_related", topics = c(1, 5, 13),
     model = poliblogPrevFit15, method = "difference",
     cov.value1 = "0", cov.value2 = "1",
     xlab = "Less Political ... More Political",
     main = "Effect of Non-political vs. Political",
     labeltype = "custom", xlim = c(-.1, .1),
     custom.labels = c('Blockings', 'Russian technologies', 'China experience'))
```

Стандартная визуализация эффекта ковариата



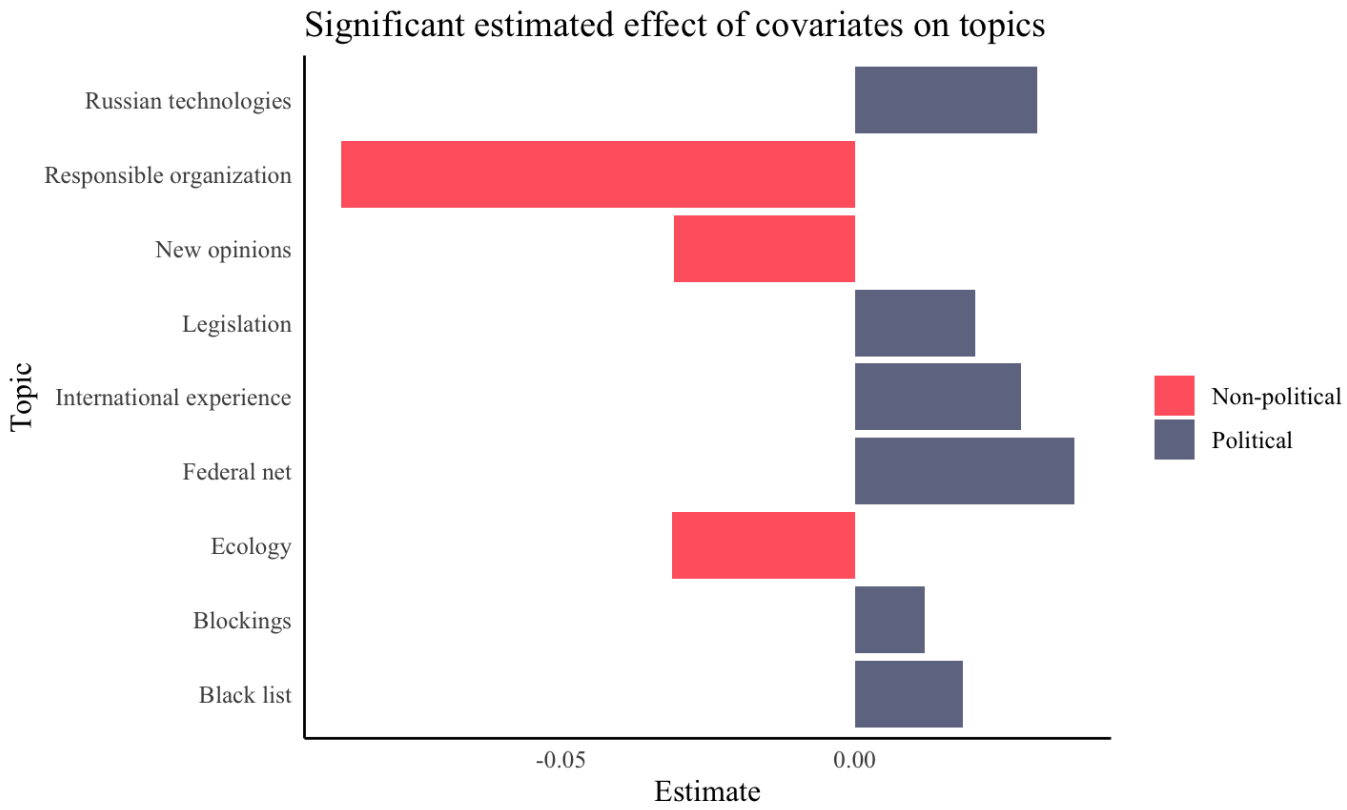
Визуализация эффекта ковариата в ggplot2

```
sumdf <- summary(predict_topics)
#getting data from the model
tblsum <- sumdf$tables

sign_topics <- as.data.frame(cbind(topic = topicNames,
                                   polit_true_sign = rep(NA, 15),
                                   estimate = rep(NA, 15)))
sign_topics$polit_true_sign = as.numeric(sign_topics$polit_true_sign)
sign_topics$estimate = as.numeric(sign_topics$estimate)

for (i in c(1:15)){
  sign_topics$polit_true_sign[[i]] = tblsum[[i]][2,4]
}

for (i in c(1:15)){
  sign_topics$estimate[[i]] = tblsum[[i]][2,1]
}

sign_topics$Group<-ifelse(sign_topics$polit_true_sign<=0.05,"Significant","Insignificant")
sign_topics$polit <- ifelse(sign_topics$estimate<0, "Non-political", "Political")
sign_topics <- sign_topics %>% filter(Group != "Insignificant")
cols <- c("Non-political"="#FF1B1C", "Political"="#202C59")
```

Визуализация эффекта ковариата в ggplot2

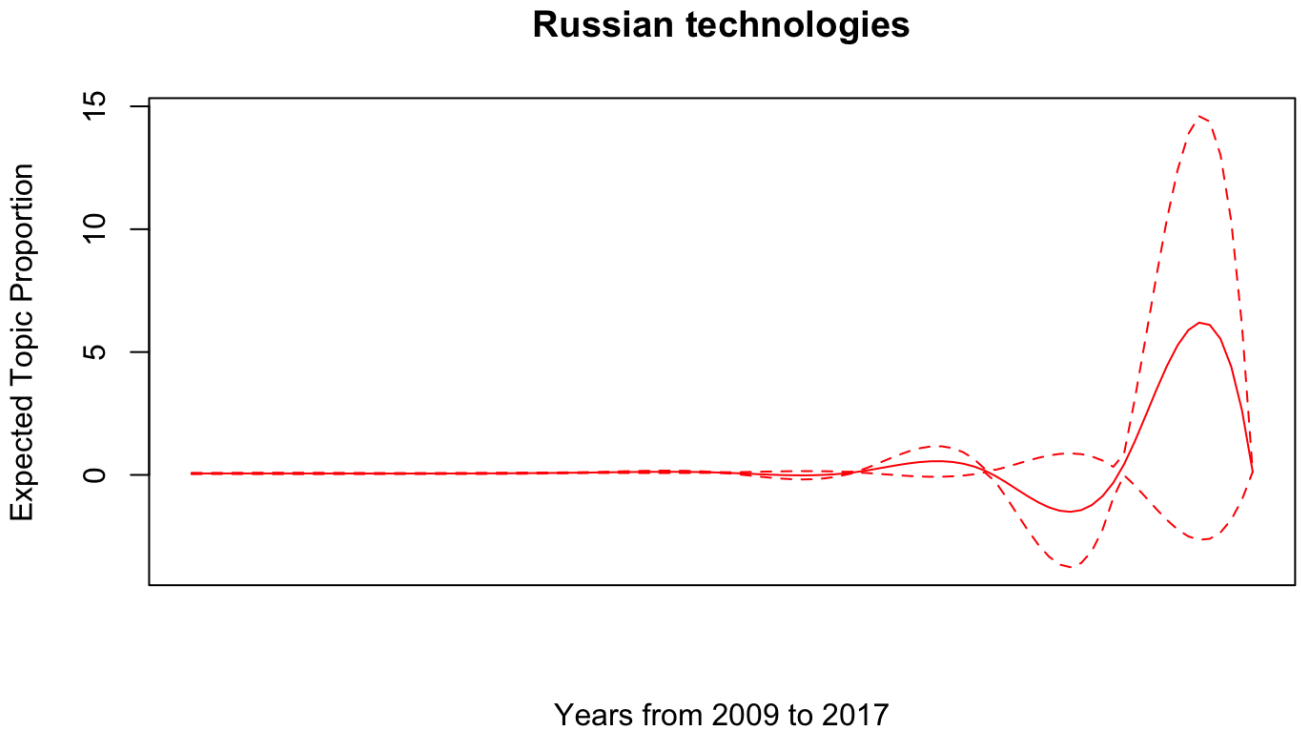
```
ggplot(sign_topics, aes(x=topic,y=estimate,fill=polit)) +
  geom_bar(stat="identity", alpha=0.7) +
  ylab("Estimate") +
  xlab("Topic") +
  ggtitle("Significant estimated effect of covariates on topics") +
  coord_flip() +
  scale_fill_manual(name="", values=cols, breaks=c("Non-political", "Political")) +
  theme_minimal() +
  theme(text=element_text(family="Times New Roman", size=12), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),panel.border = element_blank(),
        axis.line = element_line(colour = "black"))
```

Визуализация эффекта ковариата в ggplot2



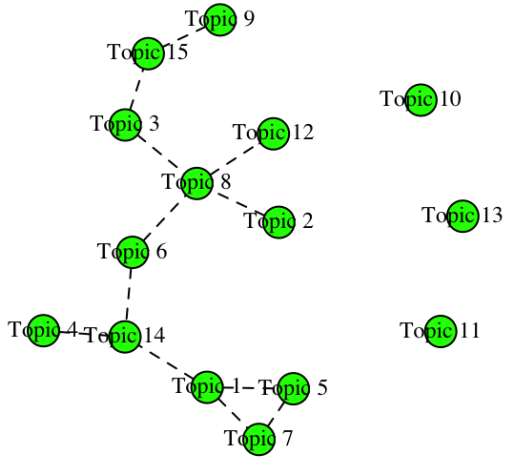
Эффект ковариата года выпуска статьи

```
plot(predict_topics, "years", method = "continuous", topics = 5,
model = z, printlegend = FALSE, xaxt = "n", xlab = "Years from 2009 to 2017")
title('Russian technologies')
```



Стандартная визуализация корреляции между темами

```
# получаем позитивные корреляции топиков
mod.out.corr <- topicCorr(poliblogPrevFit15)
plot(mod.out.corr)
```



Создание объекта igraph

```
adj_cor_topic <- mod.out.corr$posadj
adj_cor_topic_cor <- mod.out.corr$cor

# присваиваем имена топиков
adj_cor_topic <- as.matrix.data.frame(adj_cor_topic)
colnames(adj_cor_topic) <- topicNames
rownames(adj_cor_topic) <- topicNames

# получаем igraph объект из сопряженной матрицы
cor_topics <- graph.adjacency(adjmatrix = adj_cor_topic, mode = "undirected", diag = F)

# оставляем только связанные вершины
cor_topics <- delete.vertices(cor_topics,
                              V(cor_topics)[ degree(cor_topics) < 1] )

fastgreedy_topic <- fastgreedy.community(cor_topics)
```

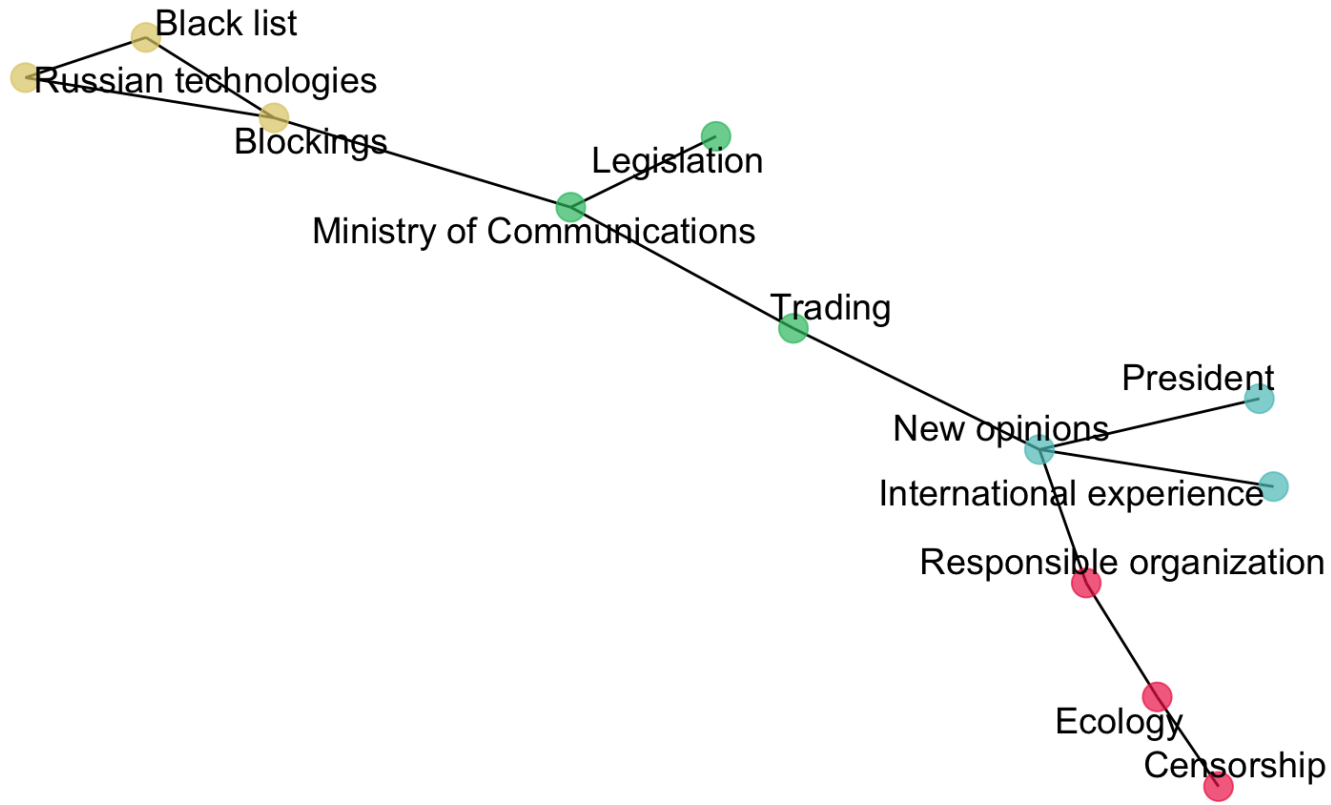
Кластеризация сети

```
# кластеризуем сеть
table_fastgreedy_topic <- cbind(fastgreedy_topic$membership, fastgreedy_topic$names)
table_fastgreedy_topic = as.data.frame(table_fastgreedy_topic)

table_fastgreedy_topic$V1 = as.character(table_fastgreedy_topic$V1)
table_fastgreedy_topic$V2 = as.character(table_fastgreedy_topic$V2)

V(cor_topics)$Clusters = as.character(table_fastgreedy_topic$V1[match(V(cor_topics)$name, table_fastgreedy_topic$V2)
```

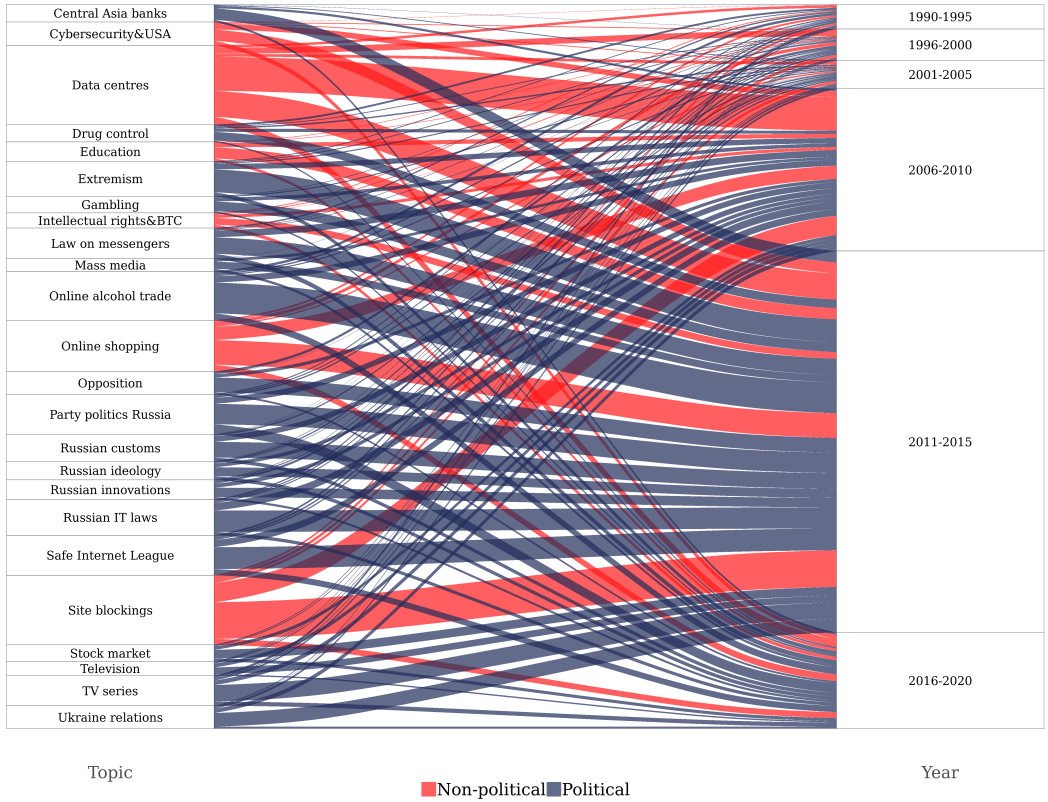
Визуализация сети корреляций тем в ggraph

```
col_topics = c("1"="#2FBF71", "2"="#EF2D56", "3"="#59C3C3",  "4"="#DFCC74")
ggraph(cor_topics, layout = "fr") +
  geom_edge_link(show.legend = FALSE) +
  geom_node_point(aes(color = Clusters), alpha = 0.7, size = 5, palette = "Set2") +
  geom_node_text(aes(label = name), repel = TRUE, size=5) +
  theme_void() +
  scale_color_manual(values=col_topics) +
  theme(legend.position="none", text=element_text(family="Times New Roman"))
```

Визуализация сети корреляций тем в ggraph

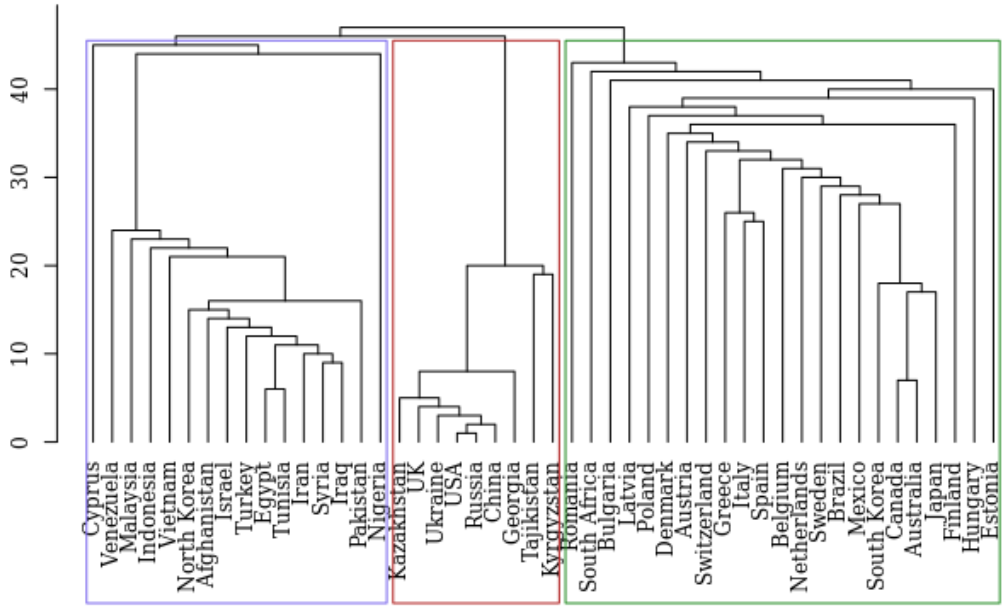


Дополнительные возможности визуализации



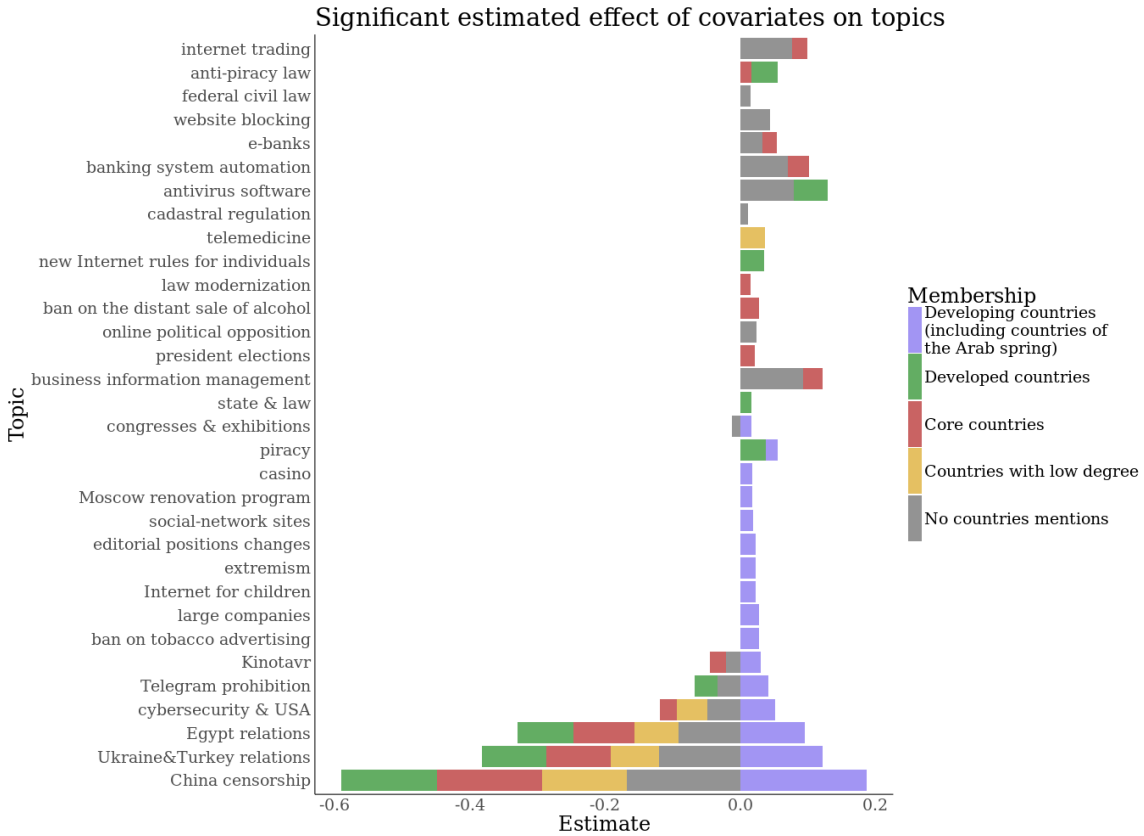
Shirokanova and Silyutina. "Internet Regulation Media Coverage in Russia: Topics and Countries" WebSci '18 Proceedings of the 10th ACM Conference on Web Science 2019. GitHub code

Дополнительные возможности визуализации



Shirokanova and Silyutina. "Internet Regulation: A Text-Based Approach to Media Coverage" International Conference on Digital Transformation and Global Society 2019. GitHub code

Дополнительные возможности визуализации



Shirokanova and Silyutina. "Internet Regulation: A Text-Based Approach to Media Coverage" International Conference on Digital Transformation and Global Society 2019. GitHub code

Ресурсы

1. stmVignette
2. Package ‘stm’
3. Chris Bail, Topic Modeling tutorial
4. structuraltopicmodel.com
5. Olga Silyutina, code for papers with stm applications
6. Wesslen, Ryan. (2018). Computer-Assisted Text Analysis for Social Science: Topic Models and Beyond.