TEXT AS DATA: WEEK 10

MATTHIAS HABER

17 NOVEMBER 2021

GOALS FOR TODAY

GOALS

- Organizational stuff
- Topic models

TOPIC MODELS

TOPIC MODELS: BASIC IDEA

We often have collections of documents that we'd like to divide into natural groups so that we can understand them separately. Topic modeling is a method for unsupervised classification of such documents, which finds natural groups of items even when we're not sure what we're looking for.

TOPIC MODELS: BASIC IDEA

- Topic models are exploratory probability models that
  - weaken the contraints required in dictionary based content analysis
  - have been intensively studied in the computer science literature
- Topic models work best with large amounts of text with a thematic structure

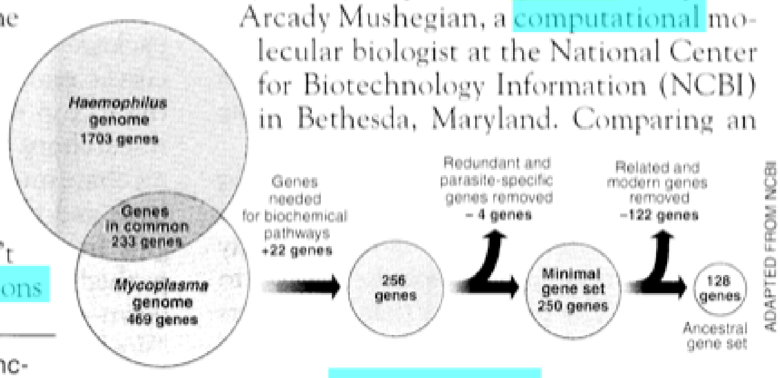
TOPIC MODELS: LDA

Latent Dirichlet allocation (LDA) is a popular method for fitting a topic model. It treats each document as a mixture of topics, and each topic as a mixture of words.LDA is a method for estimating both of these at the same time: the bag of words associated with each topic, and the bag of topics that describe each document.

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

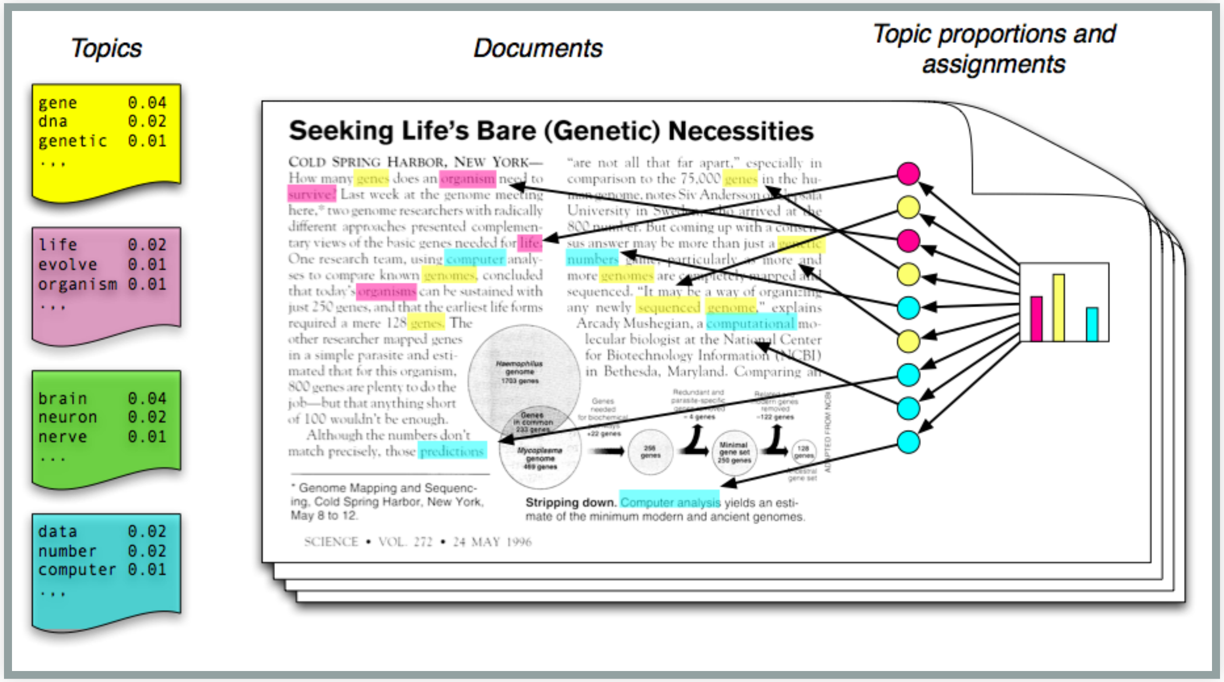


Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

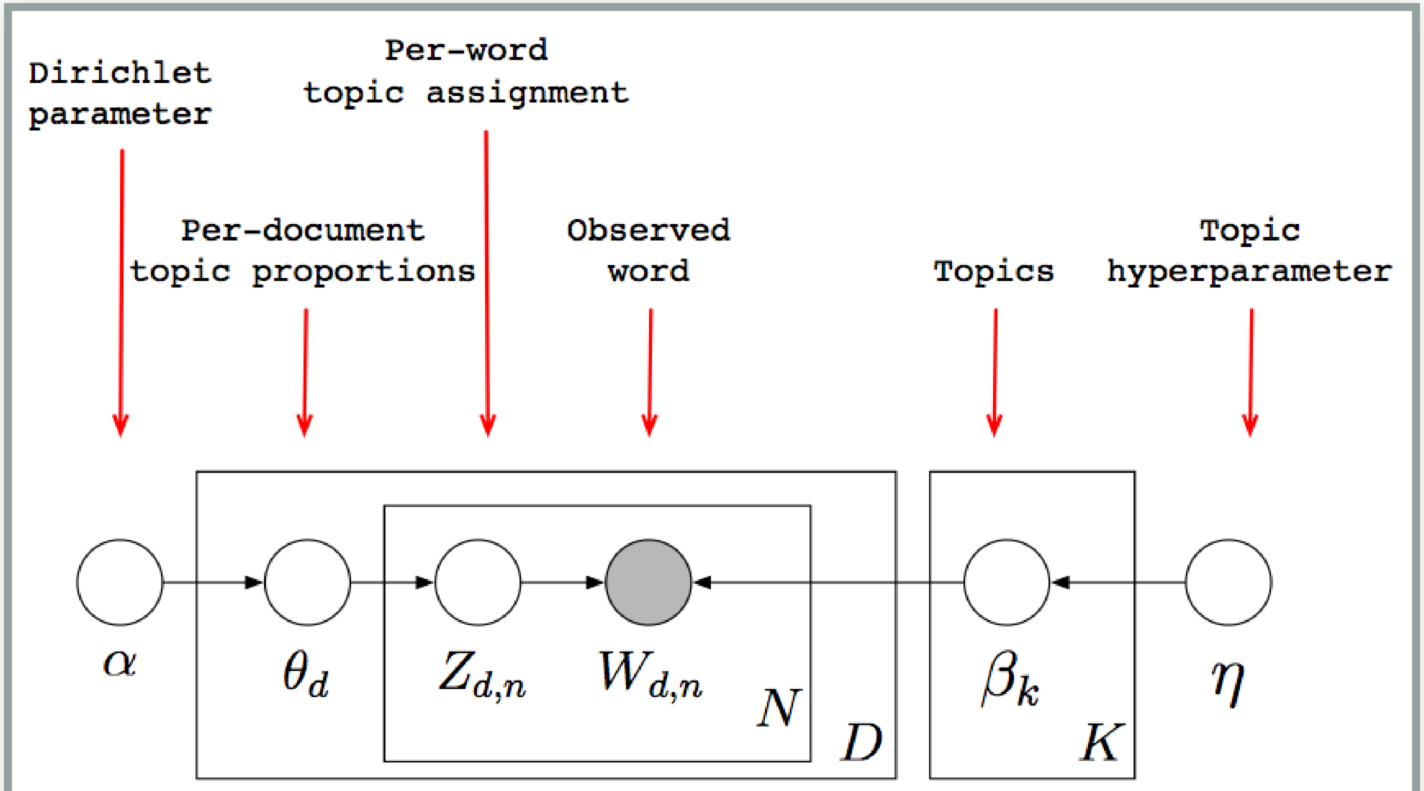
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

TOPIC MODELS: LDA (II)

We assume that some number of topics exists for the whole collection of documents. Each document is generated by first choosing a distribution over the topics, then, for each word, choosing a topic assignment and choosing the word from the corresponding topic



TOPIC MODELS: LDA (III)



Each piece of the structure is a random variable.

TOPIC MODEL: LDA (IV)

- Topic models giveth:
  - a probabilistic view of the relationship between W, Z and θ
  - a full statistical framework for learning most aspects of the relationship
- and taketh away:
  - substantive control: You do not get to assert what the topics mean (inevitable when the Z and θ are both unobserved)

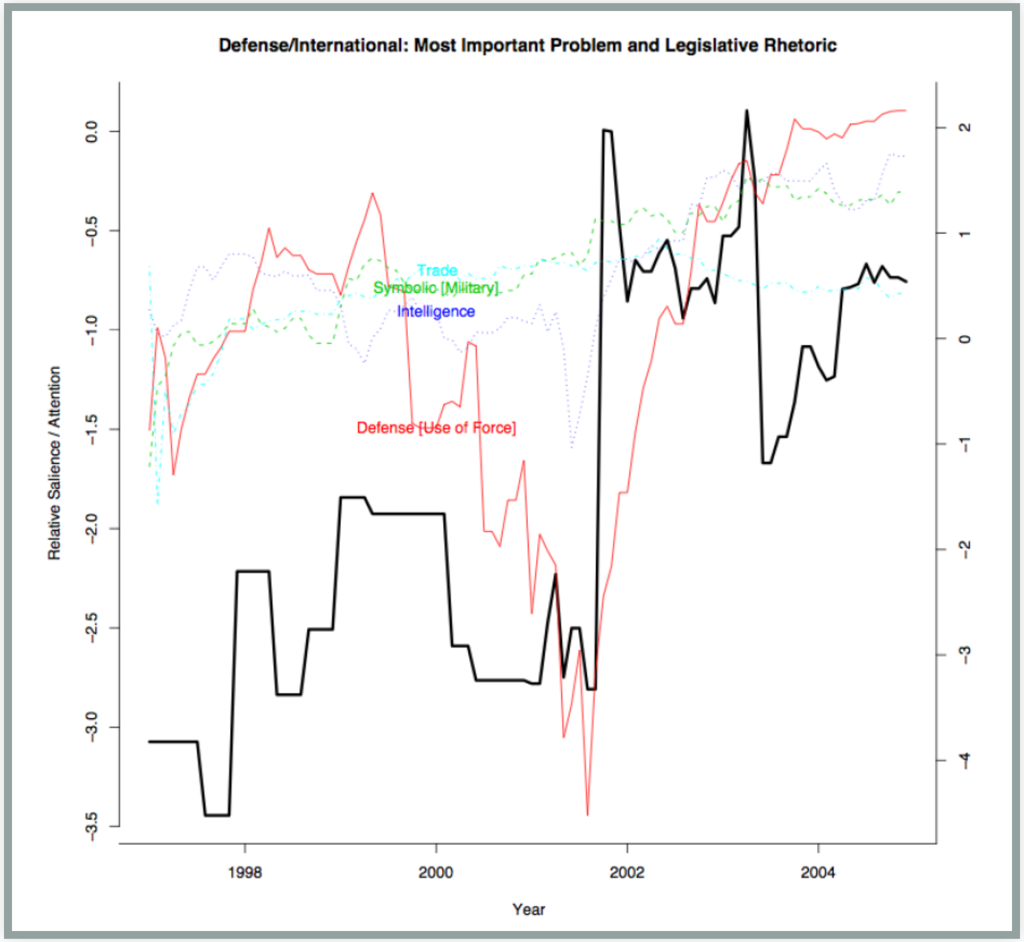
TOPIC MODEL: GIBBS SAMPLING

- Topic models need to estimate lots of unknowns simultaneously and thus can be quite time consuming to estimate. To estimate the correct weights LDA uses Gibbs sampling, an algorithm for successively sampling conditional distributions of variables.

$$\boxed{p(z_{d,n}=k|\vec{z}_{-d,n},\vec{w},\alpha,\lambda)=\frac{n_{d,k}+\alpha_k}{\sum_i^K n_{d,i}+\alpha_i}\frac{v_{k,w_{d,n}}+\lambda_{w_{d,n}}}{\sum_i v_{k,i}+\lambda_i}}$$

APPLICATION: POLICY AGENDA

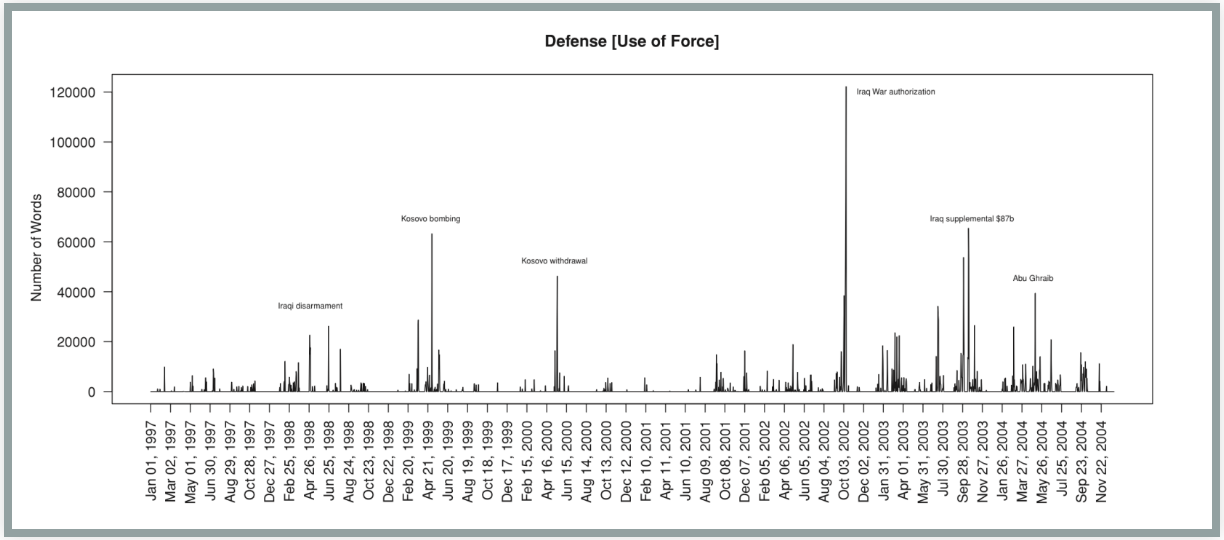
- Quinn et al. analyze 118,065 congressional speeches from 1997-2004.



OUTBUT β

Topic (Short Label)	Keys
1. Judicial Nominations	*nomine, confirm, nomin, circuit, hear, court, judg, judici, case, vacanc*
2. Constitutional	*case, court, attornei, supreme, justic, nomin, judg, m, decis, constitut*
3. Campaign Finance	*campaign, candid, elect, monei, contribut, polit, soft, ad, parti, limit*
4. Abortion	*procedur, abort, babi, thi, life, doctor, human, ban, decis, or*
5. Crime 1 [Violent]	*enforc, act, crime, gun, law, victim, violenc, abus, prevent, juvenil*
6. Child Protection	*gun, tobacco, smoke, kid, show, firearm, crime, kill, law, school*
7. Health 1 [Medical]	*diseas, cancer, research, health, prevent, patient, treatment, devic, food*
8. Social Welfare	*care, health, act, home, hospit, support, children, educ, student, nurs*
9. Education	*school, teacher, educ, student, children, test, local, learn, district, class*
10. Military 1 [Manpower]	*veteran, va, forc, militari, care, reserv, serv, men, guard, member*
11. Military 2 [Infrastructure]	*appropri, defens, forc, report, request, confer, guard, depart, fund, project*
12. Intelligence	*intellig, homeland, commiss, depart, agenc, director, secur, base, defens*
13. Crime 2 [Federal]	*act, inform, enforc, record, law, court, section, crimin, internet, investig*
14. Environment 1 [Public Lands]	*land, water, park, act, river, natur, wildlif, area, conserv, forest*
15. Commercial Infrastructure	*small, busi, act, highwai, transport, internet, loan, credit, local , capit*
16. Banking / Finance	*bankruptci, bank, credit, case, ir, compani, file, card, financi, lawyer*
17. Labor 1 [Workers]	*worker, social, retir, benefit, plan, act, employ, pension, small, employe*

OUTPUT θ

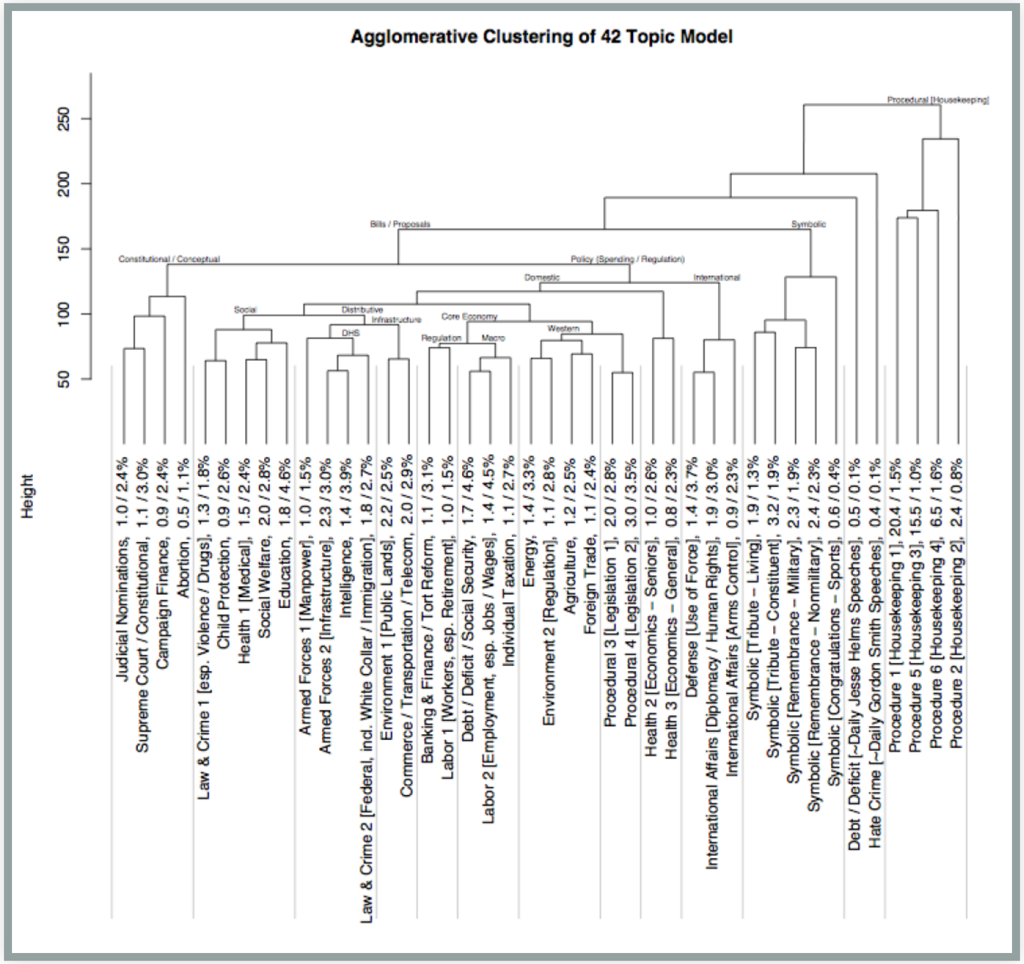


TOPIC MODEL EVALUATION

- There are two main modes of evaluation:
  - Statistical
  - Human
- and two natural levels
  - The model as a whole: model fit, K, and topic relationships
  - Topic structure: word precision, topic coherence

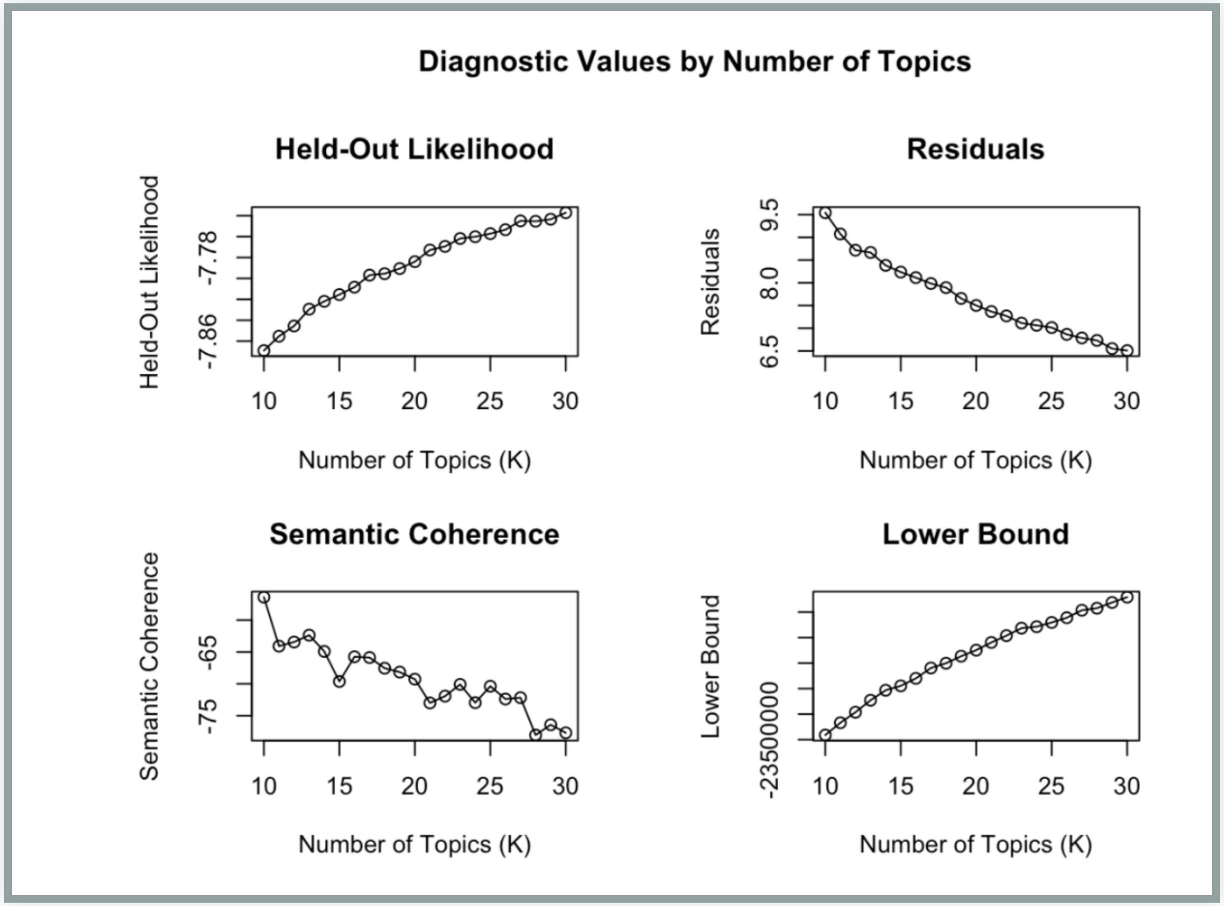
CONSTRUCT VALIDITY

Procedure: 1. Choose number of topics K 2. Fit Model 3. Label Topics 4. Cluster the β^k



CHOOSING K

The number of topics assumed a priori has a large effect on the results.

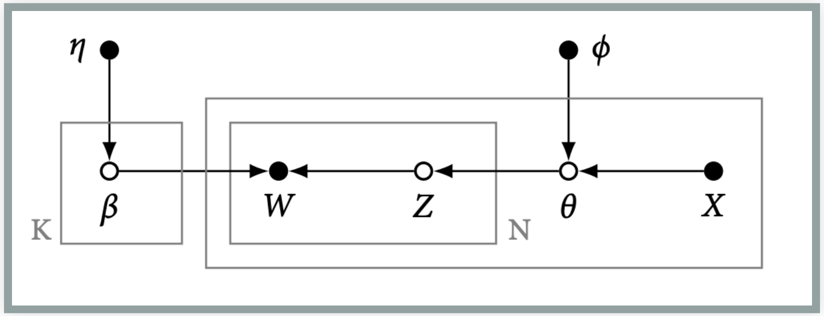


VARIATIONS: SEEDED LDA

- Seeded LDA is a semi-supervised automated content analysis model and a variant of the standard LDA approach. While standard LDA does not assume the topics to be found a priori, seeded LDA uses “seed words” to weigh the prior distribution of topics before fitting the model.
- R: `install_packages("seededlda")` (also comes with great diagnostic functions)

VARIATIONS: STRUCTURAL TOPIC MODEL

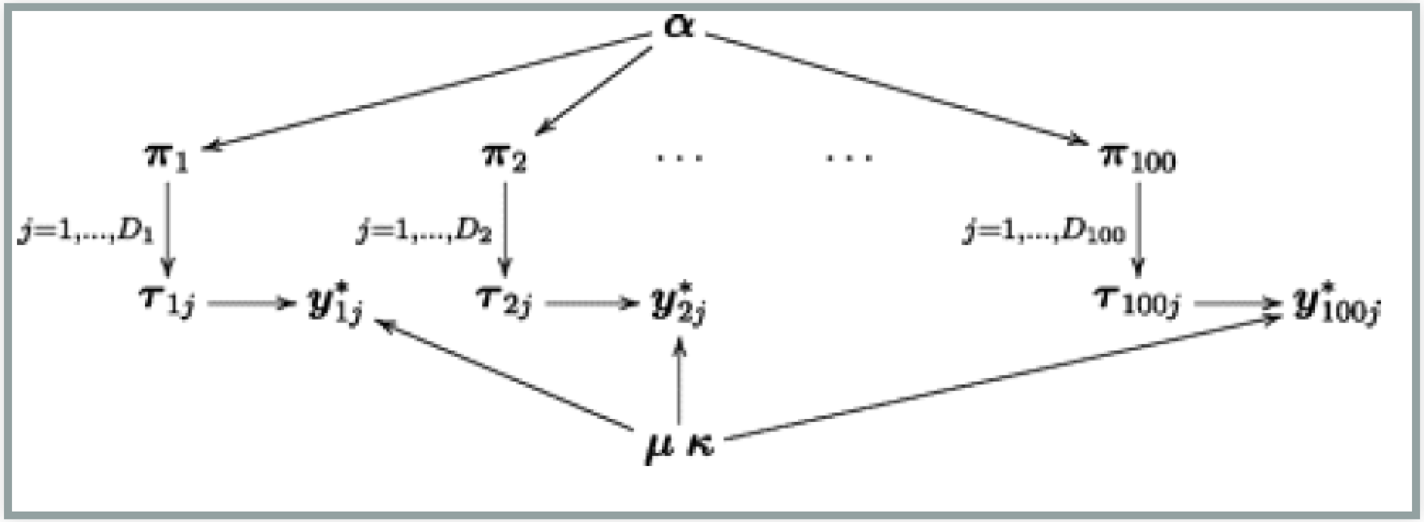
Structural topic models (STM) are similar to LDA models but allow to include metadata (the information about each document) into the topicmodel.



- R: `install_packages("stm")` (also comes with great diagnostic functions)

VARIATIONS: EXPRESSED AGENDA MODEL

In a simpler variation on LDA, Grimmer (2009) defines an expressed agenda model as



- Here there are not multiple topics per press release, but there are observed authors drawn from a population
- R:

```
install_github("christophergandrud/ExpAgenda
```

VARIATIONS: CORRELATED TOPIC MODELS

- The Dirichlet multinomial assumptions hide a constraint about topic covariation
  - LDA cannot represent free covariation of topic proportions
  - The correlated topic model can
- Replace the Dirichlet with a Logistic Normal structure (Aitchison, 1986) with arbitrary covariance matrix
- R: `topicmodels`

GROUP EXCERCISE

TOPIC MODEL EXERCISE: LOAD DATA

We will take another look at the US Senate debate on partial birth abortion.

```
load("data/corpus_us_debate_speaker.rda")
summary(corpus_us_debate_speaker, n = 5)
```

```
## Corpus consisting of 23 documents, showing 5 documents:
##
##        Text Types Tokens Sentences party   speaker
##      ALLARD   400   1165        53     R    ALLARD
##        BOND   129    232         9     R      BOND
##       BOXER  2231  18527       886     D     BOXER
##   BROWNBACK   646   2884       168     R BROWNBACK
##     BUNNING   281    593        32     R   BUNNING
```

TOPIC MODEL EXERCISE: RESHAPE TO PARAGRAPHS

The 23 speeches are probably too big to cover only one topic. So we'll reshape them to paragraphs treating each paragraph as a separate document instead. We can use the `corpus_reshape()` function for that purpose.

```
speeches_para <- corpus_reshape(corpus_us_debate_speaker, to =
        "paragraphs")
head(summary(speeches_para))
```

```
##        Text Types Tokens Sentences party speaker
## 1 ALLARD.1     32     48         3     R  ALLARD
## 2 ALLARD.2     41     64         4     R  ALLARD
## 3 ALLARD.3     25     29         1     R  ALLARD
## 4 ALLARD.4     22     24         1     R  ALLARD
## 5 ALLARD.5     68    144         3     R  ALLARD
## 6 ALLARD.6     55     89         3     R  ALLARD
```

TOPIC MODEL EXERCISE: RESHAPE TO PARAGRAPHS?

The paragraph splitter does not always produce very good results.

```
table(ntoken(speeches_para))
```

```
## 
##   0   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  15  53  24  57  54  43  62 129 210 211 214 138  77  35  14  12  11
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
##   6   3   5   5   4   3   4   9   5   8   8   6  10   4   2   7   6
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57
##   8   8   5   3   5  11   1   3   9   4   4   5   4   6   8   3   9
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77
##   5   9   3   8   6   3  10   7   4   6  13  11   5   3   4   3   9
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97
##   4   2   5   2   3   6   6   3  11   4   1   5   4   3   4   4   5
## 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
##   3   3   2   3   3  10   3   2   4   2   3   2   5   1   2   3   3
## 122 123 124 125 126 127 128 129 131 132 133 134 135 137 138 139 140
##   4   1   3   4   1   2   2   5   2   2   1   2   1   4   4   3   1
## 144 145 146 148 151 155 157 158 160 161 165 166 168 169 174 175 182
##   2   1   1   2   1   1   1   2   1   1   1   1   1   1   1   1   1
## 200 206 209 211 235 259 276
```

TOPIC MODEL EXERCISE: CREATE SUBSET AND DFM

We'll only consider those paragraphs that contain at least 8 words, remove punctuation, numbers, stop words, and tokens with less than 2 characters.

```
speeches_para <- corpus_subset(speeches_para, ntoken(speeches_para) >
        7)

para_tokens <- tokens(speeches_para,
               remove_punct = TRUE,
               remove_numbers = TRUE) %>%
  tokens_remove(stopwords()) %>%
  tokens_select(min_nchar = 2)

para_dfm <- dfm(para_tokens)
```

TOPIC MODEL EXERCISE: LDA TOPIC MODEL

Quanteda does not have any built-in topic models but we can load the required functions from the `topicmodels`, the `seedlda`, the `stm`, or similar packages. The packages each support different types of topic models and come with different functions for further analysis. We will run an LDA model with 10 topic categories using the `seededlda` package.

```
library(seededlda)
para_lda <- textmodel_lda(para_dfm, k = 10)
```

TOPIC MODEL EXERCISE: INVESTIGATE TOPIC MODEL OUTPUT

Let’s look at the most important term for each topic

```
terms(para_lda, 10)
```

```
##       topic1     topic2          topic3    topic4      topic5
##  [1,] "people"   "physicians"    "women"   "baby"      "abortion"
##  [2,] "just"     "medical"       "health"  "child"     "partial-birth"
##  [3,] "child"    "physician"     "want"    "procedure" "ban"
##  [4,] "like"     "induction"     "say"     "can"       "abortions"
##  [5,] "think"    "patient"       "woman"   "fetus"     "birth"
##  [6,] "us"       "best"          "going"   "mother"    "procedure"
##  [7,] "children" "used"          "side"    "living"    "procedures"
##  [8,] "know"     "medicine"      "think"   "baby's"    "term"
##  [9,] "see"      "appropriate"   "believe" "pregnancy" "bill"
## [10,] "life"     "procedures"    "doctors" "born"      "partial"
##       topic6      topic7     topic8          topic9        topic10
##  [1,] "procedure" "roe"      "women"         "senator"     "court"
##  [2,] "life"      "wade"     "health"        "president"   "bill"
##  [3,] "health"    "senate"   "right"         "time"        "supreme"
##  [4,] "medical"   "abortion" "women's"       "mr"          "health"
##  [5,] "necessary" "law"      "choose"        "debate"      "exception
##  [6,] "mother"    "states"   "rights"        "senate"      "legislati
```

TOPIC MODEL EXERCISE: PLOT MOST IMPORTANT TERMS

We can extract the beta coefficients for each word from the model output into a data frame and tidy them up a bit to plot the key words for each topic.

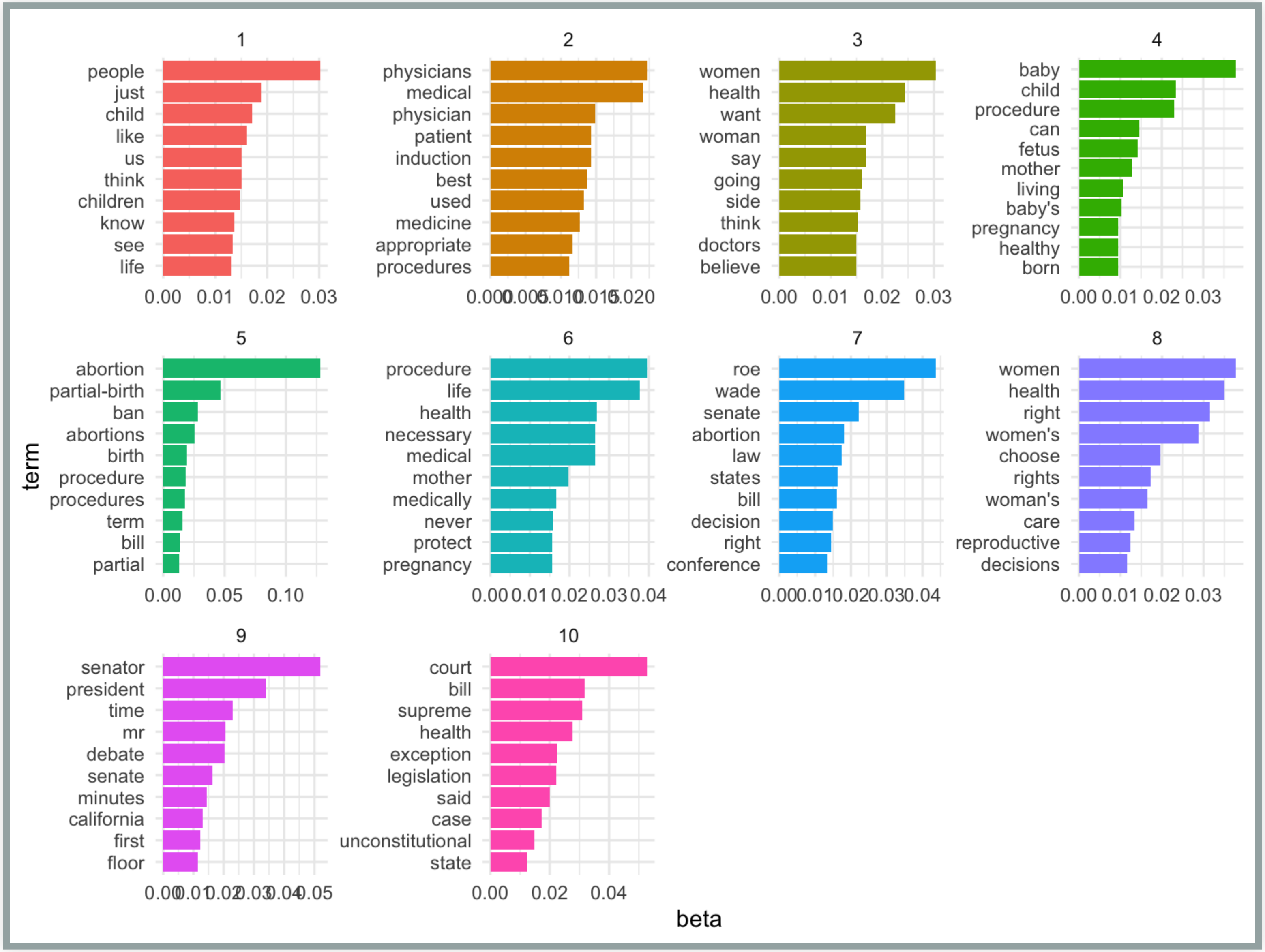
```
terms_df <- as_tibble(para_lda$phi) %>%
  mutate(topic = 1:10) %>%
  gather(term, beta, -topic) %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

TOPIC MODEL EXERCISE: PLOT MOST IMPORTANT TERMS

Then we can plot them using our familiar ggplot syntax.

```
terms_df %>%
    mutate(term = reorder_within(term, beta, topic)) %>%
    ggplot(aes(beta, term, fill = factor(topic))) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~ topic, scales = "free") +
    scale_y_reordered()
```

TOPIC MODEL EXERCISE: PLOT MOST IMPORTANT TERMS



TOPIC MODEL EXERCISE: ASSIGN TOPICS TO DOCUMENTS

We can use the `topics()` function from the `seedlda` package to obtain the most likely topic for each document and assign them as a new document-level variable.

```
para_dfm$topic <- topics(para_lda)
```

TOPIC MODEL EXERCISE: CREATE TOPIC LABELS

We can use the most important terms to create a label for each topic that helps us to differentiate between them.

```
top_terms <- terms(para_lda, 4)
topic_names <- apply(top_terms, 2, paste, collapse="_")
```

TOPIC MODEL EXERCISE: CREATE TOPIC LABELS

```
##                                      topic1
##                    "people_just_child_like"
##                                      topic2
## "physicians_medical_physician_induction"
##                                      topic3
##                     "women_health_want_say"
##                                      topic4
##                  "baby_child_procedure_can"
##                                      topic5
##   "abortion_partial-birth_ban_abortions"
##                                      topic6
##           "procedure_life_health_medical"
##                                      topic7
##                  "roe_wade_senate_abortion"
##                                      topic8
##                "women_health_right_women's"
##                                      topic9
##                 "senator_president_time_mr"
```

TOPIC MODEL EXERCISE: PLOT TOPIC DISTRIBUTION

Similar to how we plotted the most important words per topic we can also extract the gamma coefficients of the model to plot the prevalence of topics across all documents.

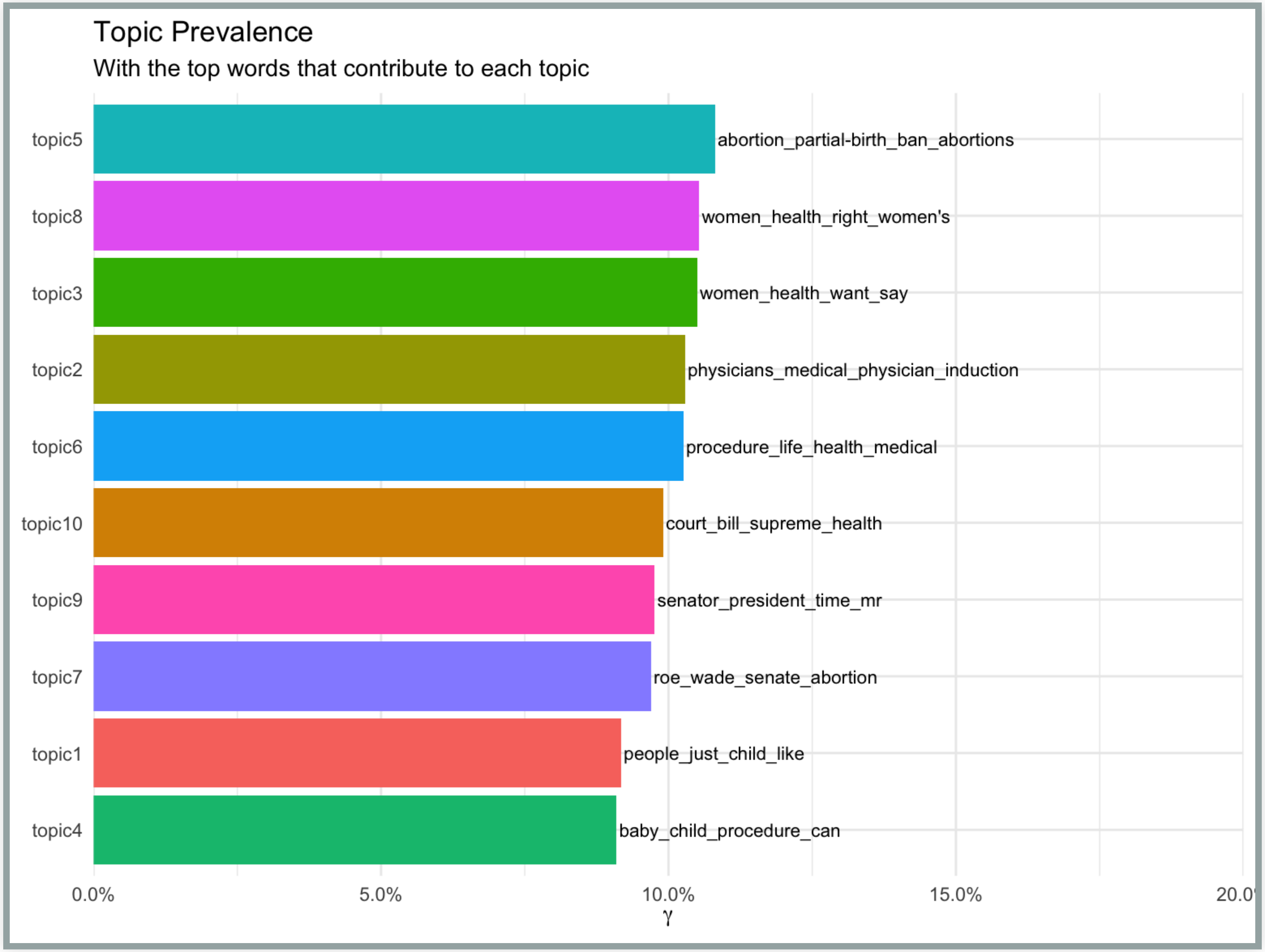
```
topic_names_df <- dplyr::bind_rows(topic_names) %>%
  gather(topic, names)

topics_df <- as_tibble(para_lda$theta) %>%
  mutate(document = rownames(.)) %>%
  gather(topic, gamma, -document) %>%
  group_by(topic) %>%
  summarise(gamma = mean(gamma)) %>%
  arrange(desc(gamma)) %>%
  left_join(topic_names_df, by = "topic")
```

TOPIC MODEL EXERCISE: PLOT MOST IMPORTANT TERMS

Then we can plot them using our familiar ggplot syntax.

```
topics_df %>%
  ggplot(aes(reorder(topic, gamma), gamma, label = names, fill =
        topic)) +
  geom_col(show.legend = FALSE) +
  geom_text(hjust = 0, nudge_y = 0.0005, size = 3) +
  coord_flip() +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 0.2),
                     labels = scales::percent_format()) +
  labs(x = NULL, y = expression(gamma),
       title = "Topic Prevalence",
       subtitle = "With the top words that contribute to each topic")
```

TOPIC MODEL EXERCISE: PLOT MOST IMPORTANT TERMS



ASSIGNMENT 2

ASSIGNMENT 2

So far, we have looked only at one variant of the topic model. For the 2nd assignment you will explore the structural topic model from the `stm` package along with various diagnostic functions. You find the instructions for the assignment on GitHub and Moodle.

Due date: 30 November 2021 Submission form: RMarkdown document

WRAPPING UP

QUESTIONS?

OUTLOOK FOR OUR NEXT SESSION

Next week we will look at the very powerful `spacyr` package.

THAT'S IT FOR TODAY

Thanks for your attention!