Final\_dissertation\_script

Below is the programming script for the paper *Towards a theory of Russian information manipulation: What do tweets from the ‘Internet Research Agency’ tell us about Russia’s misinformation strategy*, using the programming language ‘R’. The code roughly follows the order of analysis in the paper, and annotations have been added to assist the reader in understanding the code.

All of the packages except {tidystm} are available on CRAN. For {tidystm} it is necessary to install directly from Github, the code for doing so is provided in the chunk below. The data can be accessed at https://transparency.twitter.com/en/information-operations.html

and the stopwords can be found at: <https://github.com/stopwords-iso/stopwords-ru/blob/master/stopwords-ru.txt>

Initially, I set the parameters for the paper and add the libraries.

options(digits = 2, scipen = 999)

set.seed(1234)

library(gt) #for making tables  
library(tidyverse)

library(devtools) #installing non-cran packages

library(lubridate) #dates

library(tidytext) #text analysis

# devtools::install\_github("mikajoh/tidystm", dependencies = TRUE)  
library(stm) #stm

library(ggthemes) #for ggplot2 themes  
library(patchwork) #for patching together plots   
library(tidystm) #STM eval but tidy  
library(igraph) #network analysis

library(ggraph) #network analysis   
library(tidygraph) #more network analysis but tidy so can wrangle graphs

library(SnowballC) #stemming words  
library(furrr) #in order to implement tidy parallel processing

plan(multiprocess) #engage multiple cores

Data import and clean

data <- read\_csv("ira\_tweets\_csv\_hashed.csv")  
  
tweets <- data %>%   
 filter(tweet\_language == "ru" | account\_language == "ru") #filter for only Russian tweets  
  
rm(data)  
  
tweets <- tweets %>%   
 select(-user\_display\_name, -user\_profile\_description, -user\_profile\_url, -user\_reported\_location, -account\_creation\_date, -longitude, -latitude, -poll\_choices) #remove variables not used  
  
  
user\_data\_info <- read\_csv("/Users/user/Desktop/R\_Studio\_Work/Dissertation/Dissertation\_R/ira\_users\_csv\_hashed.csv") #data about accounts, not tweets  
  
user\_data\_info <- user\_data\_info %>%   
 filter(account\_language == "ru")  
  
  
  
### import stopwords lexicon  
  
english\_stopwords <- get\_stopwords("en")  
stopwords\_ru <- read\_delim("/Users/user/Desktop/R\_Studio\_Work/Dissertation/Dissertation\_R/stopwords-ru.txt", delim = ",", col\_names = F)  
names(stopwords\_ru) <- c('word')  
stopwords\_ru <- as\_tibble(stopwords\_ru)  
  
custom\_stop\_words <- bind\_rows(tibble(word = c("https", "t.co", "rt", "amp", "все", "это", 'мной', 'почему', 'спасибо', 'просто', 'очень', "привет", "всё", "тебе", "вообще", "день", "думаю", "пока", "такое", "такие", "таких", "изза", "ещё", "знаю")),   
 stopwords\_ru)  
extra\_stop\_words <- c('сам', 'хочет', 'наш')

#### Competency analysis

These are the competency dictionaries provided by Guriev and Treisman (2019b)

violence\_dictionary<-c('мертвый','смерть','смертельный','случайный','умереть','умер','умирает','умирать','уничтожить','уничтожить','смертельный','похороны','холокост','убить','бойня','скорбеть','убить','война','войны','воюя','разбить','разгромить','маршруты','маршрутизация','забастовка','ударил','беспокоящий','конфликт','враждебное','оружие','пистолет','оружие','застрелен','битва','сражения','вооруженные','больно','ранит','вред','вредугнетать','уничтожать','разрушать','тюрьма','наказывать','порабощать','раб','добыча','кровь','кровоточить','кровоточить','мученик','мученики','мученики','армии','армия','боль','болезненные','боли','вторжение','вторжение','насилие','насильственный','взорваться','взрывается','бомба','раздавить','ранить','ранить','сражаться','преследовать','тиранизировать','уничтожить','перестрелка','солдат','завоевать','пушка','террор','терроризм','террорист','злодеяние','зверства','жестокие','жестокие','мучения','штык','старв','осада','сдача','разбить','вооружение','танки','артиллерия','миномет','броня','завоевание','военный','крестовыйпоход','преступник','преступление','арест','просекут','флот','враг','враги','вражда','пленник','бич','мутилат','гибель','опустошение','варвар','полиция','побеждение','жертва','заложник','пуля','оружие','мясник','гибель','войска','грабеж','ненависть','страдания','бригада','задержание','ликвидация','жестокоеобращение','тюремноезаключение','заключениеподстражу','заложники')

economic\_dictionary<-c('доступный','аудитор','аудиторы','одолжить','купил','бюджет','купить','дешево','дешевле','валюта','клиент','долг','депозит','скидка','доллар','доллары','заработок','эконом','рецессия','аренда','розничнаяторговля','выручка','богаче','богатство','богатейший','салар','продажа','продажа','экономия','продажа','продажа','магазин','продажа','магазин','торговля','торговля','заработнаяплата','заработнаяплата','богатство','богаче','богатейших','богатых','обмен','расходы','дорого','финансы','фонд','доход','страхование','инвестирование','инвестиции','инвестирование','инвестирование','аренда','кредитование','кредитование','кредит','рынок','купец','деньги','монополия','ипотека','пенсия','песеты','бедность','цена','цены','прибыль','покупки','зарплата','акции','коммерция','рост','работа','работа','продукция','промышленность','отрасли','промышленность','индустриализация','индустриализация','производство','труд','труд','труд','труд','труд','работа','продукция','потребитель','фабрика','фабрики','remunerat','товары','занятые','безработица','инфляция','сельскоехозяйство','аграрныйсектор','тариф','рацион','нормирование','экспорт','импорт','импорт','импорт','выпуск','предприниматель','эффективность','проспэр','дефицит','сельскоехозяйство','выращивание')

social\_dictionary<-c('расходы','медицинские','медицина','образование','жилье','школа','школы','университеты','университет','класснаякомната','уходзадетьми','больница','больницы','доктор','материнство','инфраструктура','грамотность','администрация','транспорт','выходнапенсию','финансирование','инвалид','доход','бюджет','сборы','фонд','страхование','пенсия')

competency\_dictionary <- c(violence\_dictionary, economic\_dictionary, social\_dictionary)  
competency\_dictionary\_or <- paste(competency\_dictionary, collapse = "|") #add logical operators for analysis  
  
violence\_dictionary\_or <- paste(violence\_dictionary, collapse = "|")  
economic\_dictionary\_or <- paste(economic\_dictionary, collapse = "|")  
social\_dictionary\_or <- paste(social\_dictionary, collapse = "|")

Competency tokens

competency\_tokens <- tweets %>%   
 select(tweet\_text, tweet\_time) %>%   
 mutate(tweet\_text = tolower(tweet\_text),  
 tweet\_time = as.Date(tweet\_time),  
 tweet\_time = floor\_date(tweet\_time, "2 weeks")) %>% #tweets rounded to two week periods  
 unnest\_tokens(tokens, tweet\_text, token = "words") #unnested so one row per word

## Warning in floor\_date(tweet\_time, "2 weeks"): Multi-unit not supported for  
## weeks. Ignoring.

competency\_tokens\_filtered <- competency\_tokens %>%   
 add\_count(tweet\_time, name = 'obs') %>%   
 filter(tokens %in% c(economic\_dictionary, violence\_dictionary, social\_dictionary)) #keep only competency words  
   
competency\_tokens\_filtered$theme[competency\_tokens\_filtered$tokens %in% economic\_dictionary] <- 'Economic'

## Warning: Unknown or uninitialised column: `theme`.

competency\_tokens\_filtered$theme[competency\_tokens\_filtered$tokens %in% violence\_dictionary] <- 'Violent'  
competency\_tokens\_filtered$theme[competency\_tokens\_filtered$tokens %in% social\_dictionary] <- 'Social'  
  
  
competency\_tokens\_processed <- competency\_tokens\_filtered %>%   
 mutate(theme = as.factor(theme)) %>%   
 add\_count(tweet\_time, theme, name = 'n') %>%   
 transmute(tweet\_time, theme, percent = (n / obs \* 100)) %>%   
 filter(tweet\_time > as.Date("2012-01-01") & tweet\_time < as.Date("2017-06-01")) %>%   
 distinct() #calculate frequency of tokens  
  
  
competency\_tokens\_graph <- competency\_tokens\_processed %>%   
 ggplot(aes(tweet\_time, percent, col = theme)) +  
 geom\_line(alpha = 0.5) +  
 theme\_solarized() +  
 scale\_color\_solarized() +  
 scale\_x\_date(breaks = scales::date\_breaks(width = "1 year"),  
 labels = scales::date\_format(format = "%Y"),  
 minor\_breaks = "6 months") +  
 expand\_limits(y = c(0, 1.5)) +  
 labs(title = "Frequency of 'competency' words as a % of all words", x = "", y = 'Frequency of terms (%) in IRA tweets', color = "Competency") +  
 theme(legend.position = "none")  
  
competency\_tokens\_graph <- competency\_tokens\_graph +   
 theme(axis.title.y = element\_text(size = 20),  
 plot.title = element\_text(size = 25),  
 axis.text.x = element\_text(size = 16),  
 axis.text.y = element\_text(size = 16)) #graph formatting

The next code chunk calculates not the percentage frequency of tokens but the frequency of tweets mentioning one competency word

competency\_tweets <- tweets %>%   
 select(tweet\_text, tweet\_time) %>%   
 mutate(tweet\_text = tolower(tweet\_text),  
 tweet\_time = as.Date(tweet\_time),  
 tweet\_time = floor\_date(tweet\_time, "2 weeks"))

## Warning in floor\_date(tweet\_time, "2 weeks"): Multi-unit not supported for  
## weeks. Ignoring.

violence\_tweets\_filtered <- competency\_tweets %>%   
 add\_count(tweet\_time, name = 'obs') %>%   
 filter(str\_detect(tweet\_text, violence\_dictionary\_or)) %>%   
 mutate(theme = "Violent")  
   
economic\_tweets\_filtered <- competency\_tweets %>%   
 add\_count(tweet\_time, name = 'obs') %>%   
 filter(str\_detect(tweet\_text, economic\_dictionary\_or)) %>%   
 mutate(theme = "Economic")  
  
social\_tweets\_filtered <- competency\_tweets %>%   
 add\_count(tweet\_time, name = 'obs') %>%   
 filter(str\_detect(tweet\_text, social\_dictionary\_or)) %>%   
 mutate(theme = "Social")  
  
competency\_tweets\_filtered <- rbind(violence\_tweets\_filtered, economic\_tweets\_filtered, social\_tweets\_filtered)  
  
competency\_tweets\_processed <- competency\_tweets\_filtered %>%   
 filter(tweet\_time > as.Date("2012-01-01") & tweet\_time < as.Date("2017-06-01")) %>%   
 mutate(theme = as.factor(theme)) %>%   
 add\_count(tweet\_time, theme, name = 'n') %>%   
 transmute(tweet\_time, theme, percent = (n / obs \* 100)) %>%   
 distinct()  
  
competency\_tweets\_graph <- competency\_tweets\_processed %>%   
 ggplot(aes(tweet\_time, percent, col = theme)) +  
 geom\_line(alpha = 0.5) +  
 #scale\_color\_brewer(palette = "Paired", direction = -2) +  
 theme\_solarized() +  
 scale\_color\_solarized() +  
 scale\_x\_date(breaks = scales::date\_breaks(width = "1 year"),  
 labels = scales::date\_format(format = "%Y")) +  
 #minor\_breaks = "6 months") +  
 #expand\_limits(y = c(0, 2)) +  
 labs(title = "IRA tweets mentioning at least one 'Competency' key words", x = "", y = "Frequency of terms (%)", color = "Competency")  
  
  
competency\_tweets\_graph <- competency\_tweets\_graph +   
 theme(axis.title.y = element\_text(size = 22),  
 axis.text.x = element\_text(size = 16),  
 axis.text.y = element\_text(size = 16),  
 plot.title = element\_text(size = 25))

Graph objects

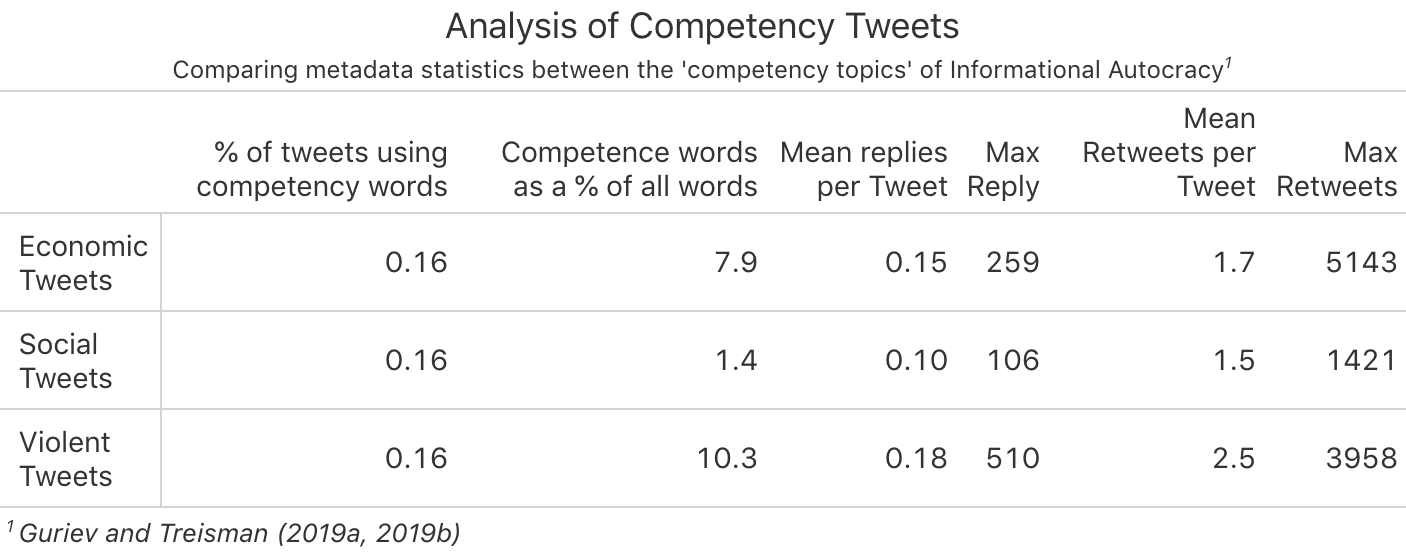
patchwork <- competency\_tokens\_graph / competency\_tweets\_graph + plot\_layout(guides = "collect")  
patchwork +  
 plot\_annotation(  
 title = "Competency Topic Frequencies",  
 theme = theme\_solarized(base\_size = 30)  
 ) +  
 theme(legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 26),  
 legend.key = element\_rect(ggthemes\_data$solarized$Base$name['base3']),  
 legend.key.size = unit(4, "lines"),  
 axis.title.y = element\_text(size = 20)) #formatting

A close up of a map

Description automatically generated

This code chunk analyses the metadata for tweets mentioning competency topics

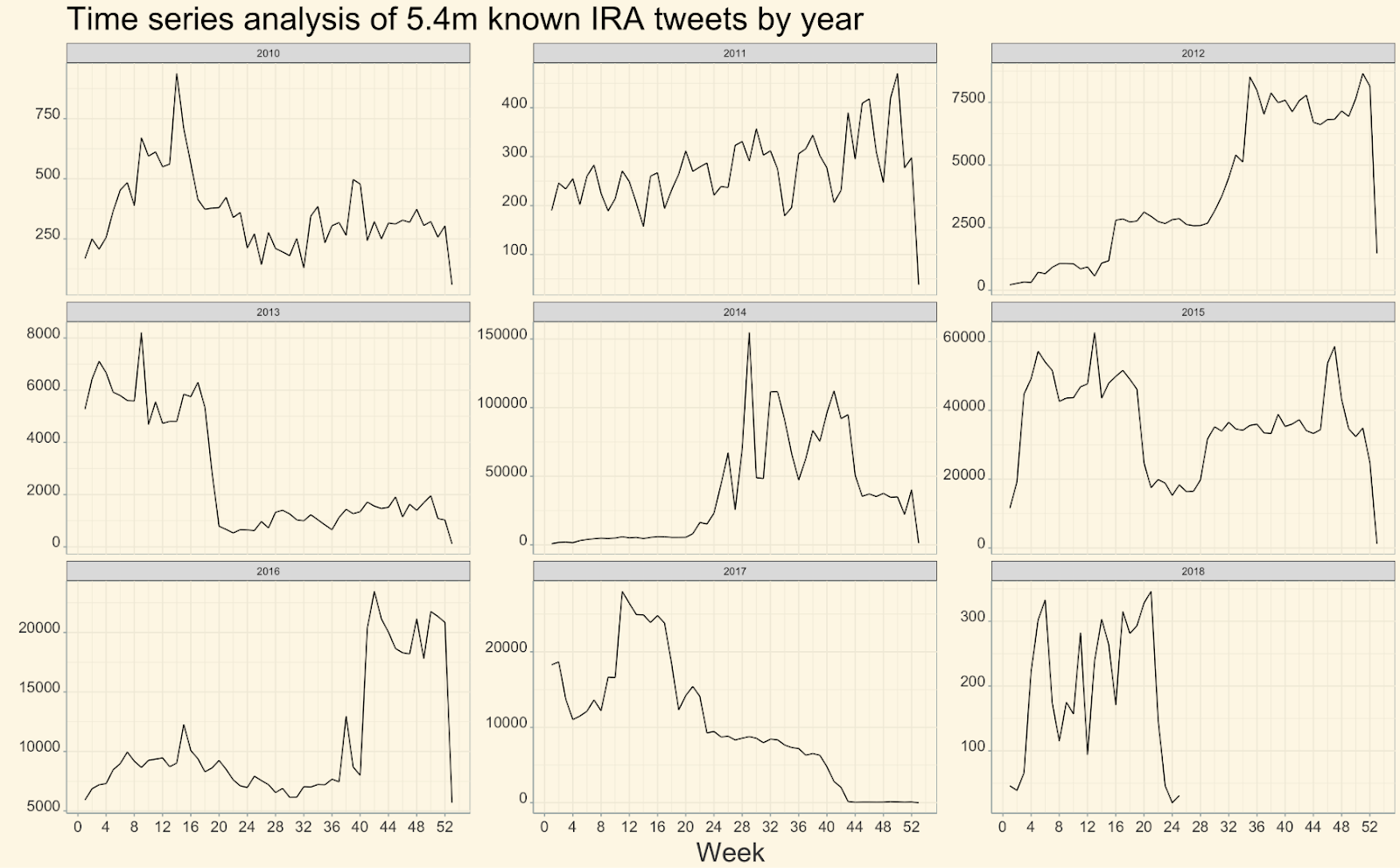
economic\_token\_freq <- competency\_tokens\_filtered %>%   
 filter(theme == 'Economic') %>%   
 summarise(token\_pcnt = nrow(.) / nrow(competency\_tokens) \* 100) #shoudnt this be competency\_tweets?  
  
social\_token\_freq <- competency\_tokens\_filtered %>%   
 filter(theme == 'Social') %>%   
 summarise(token\_pcnt = nrow(.) / nrow(competency\_tokens) \* 100)  
  
violent\_token\_freq <- competency\_tokens\_filtered %>%   
 filter(theme == 'Violent') %>%   
 summarise(token\_pcnt = nrow(.) / nrow(competency\_tokens) \* 100)  
  
#violent tweets  
  
violent\_tweets\_info <- tweets %>%  
 mutate(rows = nrow(.)) %>%   
 mutate(tweet\_text = tolower(tweet\_text)) %>%   
 filter(str\_detect(tweet\_text, paste(violence\_dictionary, collapse = '|'))) %>%  
 mutate(pcnt\_total = nrow(.) / rows \* 100)  
  
violent\_tweets\_summary <- violent\_tweets\_info %>%   
 summarise(Topic = 'Violent Tweets',  
 `% of tweets using competency words` = economic\_token\_freq$token\_pcnt,  
 `Competence words as a % of all words` = mean(pcnt\_total),  
 `Mean replies per Tweet` = mean(reply\_count, na.rm = T),  
 `Max Reply` = max(reply\_count, na.rm = T),  
 `Mean Retweets per Tweet` = mean(retweet\_count, na.rm = T),  
 `Max Retweets` = max(retweet\_count, na.rm = T))  
  
  
#social tweets  
  
social\_tweets\_info <- tweets %>%  
 mutate(rows = nrow(.)) %>%   
 mutate(tweet\_text = tolower(tweet\_text)) %>%   
 filter(str\_detect(tweet\_text, paste(social\_dictionary, collapse = '|'))) %>%  
 mutate(pcnt\_total = nrow(.) / rows \* 100)   
  
social\_tweets\_summary <- social\_tweets\_info %>%   
 summarise(Topic = 'Social Tweets',  
 `% of tweets using competency words` = economic\_token\_freq$token\_pcnt,  
 `Competence words as a % of all words` = mean(pcnt\_total),  
 `Mean replies per Tweet` = mean(reply\_count, na.rm = T),  
 `Max Reply` = max(reply\_count, na.rm = T),  
 `Mean Retweets per Tweet` = mean(retweet\_count, na.rm = T),  
 `Max Retweets` = max(retweet\_count, na.rm = T))  
  
  
#economic tweets  
  
economic\_tweets\_info <- tweets%>%  
 mutate(rows = nrow(.)) %>%   
 mutate(tweet\_text = tolower(tweet\_text)) %>%   
 filter(str\_detect(tweet\_text, paste(economic\_dictionary, collapse = '|'))) %>%  
 mutate(pcnt\_total = nrow(.) / rows \* 100)  
  
economic\_tweets\_summary <- economic\_tweets\_info %>%   
 summarise(Topic = 'Economic Tweets',   
 `% of tweets using competency words` = economic\_token\_freq$token\_pcnt,  
 `Competence words as a % of all words` = mean(pcnt\_total),  
 `Mean replies per Tweet` = mean(reply\_count, na.rm = T),  
 `Max Reply` = max(reply\_count, na.rm = T),  
 `Mean Retweets per Tweet` = mean(retweet\_count, na.rm = T),  
 `Max Retweets` = max(retweet\_count, na.rm = T))  
  
  
gt\_competency\_summaries <- rbind(economic\_tweets\_summary, social\_tweets\_summary, violent\_tweets\_summary) %>%   
 gt(rowname\_col = "Topic") %>%   
 tab\_header(  
 title = "Analysis of Competency Tweets",  
 subtitle = "Comparing metadata statistics between the 'competency topics' of Informational Autocracy"  
 ) %>%   
 tab\_footnote(  
 footnote = md('\_Guriev and Treisman (2019a, 2019b)\_'),  
 locations = cells\_title(groups = c("subtitle"))  
 )



#### Meta data analysis

tweet\_dates <- tweets %>%  
 mutate(year = lubridate::year(tweet\_time)) %>%  
 filter(year != 2009) %>%   
 mutate(tweet\_time = lubridate::week(tweet\_time)) %>%   
 select(year, tweet\_time)

tweet\_dates\_graph <- tweet\_dates %>%   
 add\_count(year, tweet\_time) %>%   
 ggplot(aes(tweet\_time, n)) +  
 geom\_line() +  
 facet\_wrap(~year, scales = "free\_y", ncol =3) +  
 theme\_solarized() +  
 scale\_x\_continuous(breaks = seq(0, 52, 4),  
 labels = c(seq(0, 52, 4))) +  
 labs(title = "Time series analysis of 5.4m known IRA tweets by year", x = "Week", y = "") +  
 theme(plot.title = element\_text(hjust = 0, size = 25, color = "black")) +  
 theme\_solarized() +  
 theme(axis.text.x = element\_text(color = "grey20", size = 14, angle = 0, hjust = .5, vjust = .5, face = "plain"),  
 axis.text.y = element\_text(color = "grey20", size = 14, angle = 0, hjust = 1, vjust = 0, face = "plain"),   
 axis.title.x = element\_text(color = "grey20", size = 25, angle = 0, hjust = .5, vjust = 0, face = "plain"),  
 axis.title.y = element\_text(color = "grey20", size = 12, angle = 90, hjust = .5, vjust = .5, face = "plain")) +  
 theme(plot.title = element\_text(size=30, face = "plain", colour = "black"))  
  
tweet\_dates\_graph

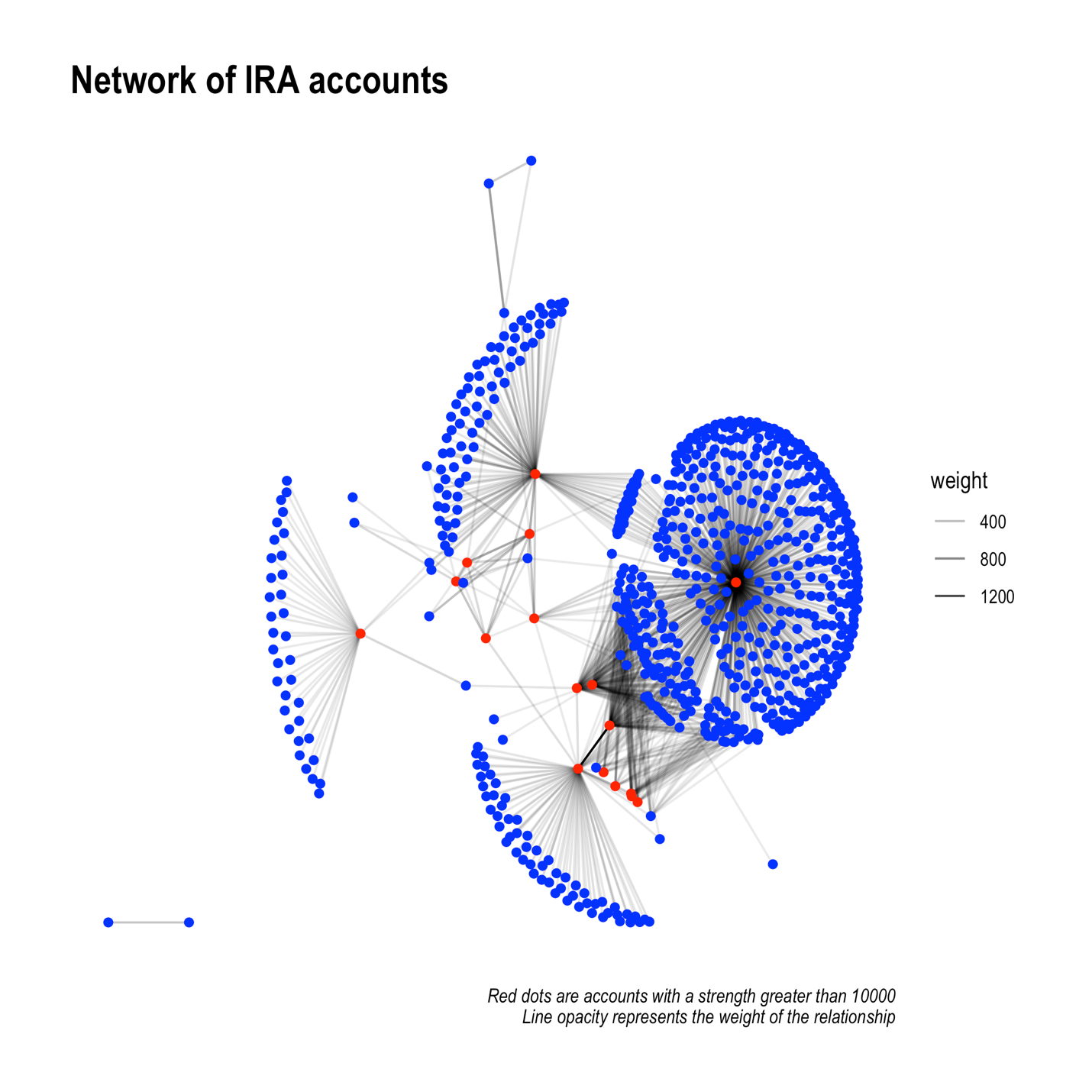


urls <- tweets %>%   
 select(urls) %>%   
 filter(!is.na(urls)) %>%   
 filter(urls != '[]')   
  
  
unique\_links <- urls %>%   
 count(urls)  
   
  
  
  
  
shortened\_urls <- urls %>%   
 filter(!str\_detect(urls, c('bit.ly'))) %>% #remove the non-intelligble links  
 filter(!str\_detect(urls, c('goo.gl'))) %>%   
 filter(!str\_detect(urls, c('j.mp'))) %>%   
 filter(!str\_detect(urls, c('dlvr.it'))) %>%   
 filter(!str\_detect(urls, c('tinyurl')))  
  
  
news\_sources <- c('news', 'new', 'ria', 'bbc', 'nyt', 'cnn', 'tass', 'vesti', 'gezeta', 'rbc', 'rt', 'meduzaproject', 'LIFENEWS\_RU', 'harkovnews', 'Pravdiva\_pravda')  
news\_sources <- paste(news\_sources, collapse = "|")  
  
  
live\_journal\_urls <- urls %>%   
 filter(str\_detect(urls, 'livejournal')) %>% #some regex to isolate livejournal urls  
 mutate(urls = str\_replace\_all(urls, 'http://', '')) %>%   
 mutate(urls = str\_replace\_all(urls, '[0-9]+\\.html', '')) %>%   
 mutate(urls = str\_replace\_all(urls, 'livejournal', '')) %>%   
 mutate(urls = str\_replace\_all(urls, '.[a-z]{2,5}(:[0-9]{1,5})?(\\/.\*)?$', '')) %>%   
 mutate(urls = str\_replace\_all(urls, 'co', '')) %>%   
 mutate(urls = str\_replace\_all(urls, 'com', ''))   
  
unique\_live\_journal\_urls <- live\_journal\_urls %>%   
 count(urls) %>%   
 nrow()  
  
news\_tweets <- urls %>%   
 filter(str\_detect(urls, news\_sources)) %>%   
 nrow()  
  
cat('Number of urls =', nrow(urls), '\nNumber of unique links =', nrow(unique\_links), '\nNumber of links to LiveJournal posts =', nrow(live\_journal\_urls), '\nNumber of unique LiveJournal links =', unique\_live\_journal\_urls, '\nNumber of tweets mentioning news organisations (conservative estimate) =', news\_tweets) #maybe look into using {glue} for this

## Number of urls = 3197415   
## Number of unique links = 1788930   
## Number of links to LiveJournal posts = 443582   
## Number of unique LiveJournal links = 10089   
## Number of tweets mentioning news organisations (conservative estimate) = 1050014

#### Network analysis

retweet\_network\_data <- tweets %>%   
 filter(retweet\_userid %in% tweets$userid) %>% #filter for internal retweets  
 select(userid, retweet\_userid) %>%   
 na.omit() %>%   
 add\_count(retweet\_userid, userid, sort = T) %>% #count the number of times accounts were retweeted  
 distinct(.) #remove duplicates made from the add\_count() function  
   
retweet\_network\_data <- retweet\_network\_data %>%   
 rename(weight = n)  
  
   
retweet\_network\_graph <- retweet\_network\_data %>%   
 filter(weight>50) %>% #remove from visualisation links that have a weight less than 50  
 graph\_from\_data\_frame() #this is a bit of a hack to turn the wrangled data frame back into a graph object  
  
retweet\_network\_graph\_df <- igraph::as\_data\_frame(retweet\_network\_graph, what = "both")  
retweet\_nodes <- retweet\_network\_graph\_df$vertices %>%   
 mutate(degree = igraph::degree(retweet\_network\_graph),  
 strength = igraph::strength(retweet\_network\_graph)) #make a separate object that contains details about nodes so I can include   
  
  
tidy\_network <- as\_tbl\_graph(retweet\_network\_graph) #now tidy the graph object into a tbl\_graph   
  
tidy\_network %>%   
 activate(nodes) %>% #isolate df of nodes within the tibble  
 mutate(strength = strength(tidy\_network)) %>% #add strength   
 activate(edges) %>%   
 mutate(weight > 20) %>%   
 ggraph(layout = "stress") +  
 geom\_edge\_link(aes(alpha = weight)) +  
 geom\_node\_point(color = ifelse(strength(retweet\_network\_graph) > 1000, 'red', 'blue')) + #highlight colour of nodes  
 theme\_graph() +  
 labs(title = "Network of IRA accounts", caption = "Red dots are accounts with a strength greater than 10000\nLine opacity represents the weight of the relationship")

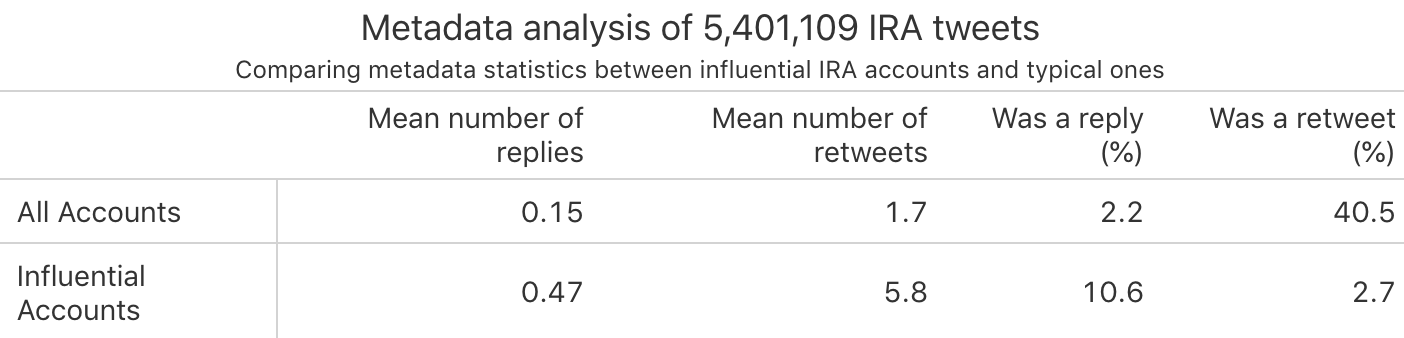


Was the relationship between the small number of influential accounts and the supporting accounts reciprocal?

top\_17\_rt\_strength <- retweet\_nodes %>%   
 arrange(desc(strength)) %>%   
 top\_n(strength, n = 17) #extract the nodes with >10,000 strength  
  
  
non\_influential\_accounts <- user\_data\_info %>%   
 filter(!userid %in% top\_17\_rt\_strength$userid)   
  
tweets %>%   
 filter(userid %in% top\_17\_rt\_strength$name) %>% #keep only influential accounts  
 select(userid, in\_reply\_to\_userid) %>%   
 na.omit() %>%   
 filter(in\_reply\_to\_userid %in% non\_influential\_accounts$userid) %>% #at this stage there are only six accounts that engage internally  
 add\_count(userid) %>%   
 dplyr::distinct(n, .keep\_all = T) %>%   
 select(-in\_reply\_to\_userid) %>%   
 rename(`Times tweeting non-influential accounts` = n)

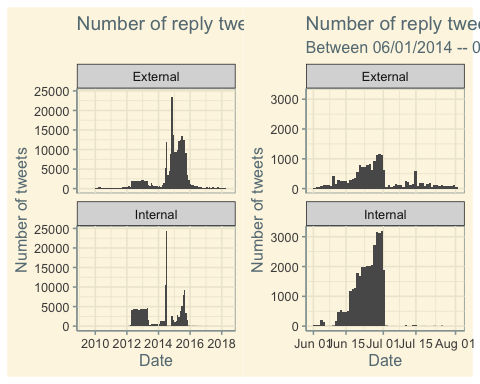
## # A tibble: 9 x 2  
## userid `Times tweeting non-influential accounts`  
## <chr> <int>  
## 1 1240007161 336  
## 2 462272724 4124  
## 3 471868887 3230  
## 4 2665564544 91  
## 5 2808833544 241  
## 6 2589513234 179  
## 7 2572896396 9  
## 8 2481527106 1  
## 9 1930747698 3

number\_of\_replies <- tweets %>%   
 filter(!in\_reply\_to\_userid %in% user\_data\_info$userid) %>%   
 select(in\_reply\_to\_tweetid) %>%   
 na.omit() %>%   
 nrow()  
  
  
number\_of\_retweets <- tweets %>%   
 #filter(!retweet\_userid %in% user\_data\_info$userid) %>%   
 select(retweet\_userid) %>%   
 na.omit() %>%   
 nrow()  
  
  
tweets\_meta\_data <- tweets %>%   
 summarise(   
 Account = 'All Accounts',  
 `Mean number of replies` = mean(reply\_count, na.rm =T),  
 `Mean number of retweets` = mean(retweet\_count, na.rm = T),  
 `Was a reply (%)` = number\_of\_replies / nrow(tweets) \* 100,  
 `Was a retweet (%)` = number\_of\_retweets / nrow(tweets) \* 100  
 )  
  
  
  
  
  
### Lets do the same process but for influential accoun  
  
  
number\_of\_influential\_replies <- tweets %>%   
 filter(userid %in% top\_17\_rt\_strength$name) %>%   
 select(in\_reply\_to\_tweetid) %>%   
 na.omit() %>%   
 nrow()  
  
  
number\_of\_influential\_retweets <- tweets %>%   
 filter(userid %in% top\_17\_rt\_strength$name) %>%   
 select(retweet\_userid) %>%   
 na.omit() %>%   
 nrow()  
  
tweets\_influential\_accounts <- tweets %>%   
 filter(userid %in% top\_17\_rt\_strength$name)  
  
  
tweets\_influential\_accounts\_summary <- tweets\_influential\_accounts %>%   
 summarise(   
 Account = 'Influential Accounts',  
 `Mean number of replies` = mean(reply\_count, na.rm =T),  
 `Mean number of retweets` = mean(retweet\_count, na.rm = T),  
 `Was a reply (%)` = number\_of\_influential\_replies / nrow(tweets\_influential\_accounts) \* 100,  
 `Was a retweet (%)` = number\_of\_influential\_retweets / nrow(tweets\_influential\_accounts) \* 100  
 )  
  
  
gt\_meta\_data <- rbind(tweets\_meta\_data, tweets\_influential\_accounts\_summary) %>%   
 gt(rowname\_col = 'Account') %>%   
 tab\_header(  
 title = "Metadata analysis of 5,401,109 IRA tweets",  
 subtitle = "Comparing metadata statistics between influential IRA accounts and typical ones"  
 )



Before the topic modelling, here is the that suggests there may have been some change in strategy in the way the IRA engaged with external accounts after July 2014

reply\_tweets <- tweets %>%   
 filter(str\_detect(tweet\_text, '^@')) #n = 350k  
  
reply\_tweets\_filtered <- reply\_tweets %>%   
 mutate(rownumber = row\_number()) %>%   
 mutate(reply = ifelse(in\_reply\_to\_userid %in% userid, 'Internal', 'External'))  
  
  
number\_of\_reply\_tweets <- reply\_tweets\_filtered %>%   
 ggplot(aes(tweet\_time)) +  
 geom\_histogram(bins = 104, show.legend = F) +  
 labs(x = "Date", y = "Number of tweets", title = "Number of reply tweets") +  
 facet\_wrap(~ reply, nrow=2) +  
 theme\_solarized()  
  
  
reply\_tweets\_filtered\_2014 <- reply\_tweets\_filtered %>%   
 filter(tweet\_time > as.Date("2014-06-01") & tweet\_time < as.Date("2014-08-01")) %>%   
 ggplot(aes(tweet\_time)) +  
 geom\_histogram(bins = 60, show.legend = F) +  
 labs(x = "Date", y = "Number of tweets", title = "Number of reply tweets", subtitle = "Between 06/01/2014 -- 01/08/2014") +  
 facet\_wrap(~ reply, nrow=2) +  
 theme\_solarized()  
  
number\_of\_reply\_tweets + reply\_tweets\_filtered\_2014



#### Structural topic modelling

The data cleaning / manipulation steps here are, at times, a bit convoluted.

pre\_mh17\_network\_data <- tweets %>%   
 filter(tweet\_time < as.Date("2014-07-17")) %>%  
 filter(retweet\_userid %in% tweets$userid) %>%   
 select(userid, retweet\_userid) %>%   
 na.omit() %>%   
 add\_count(retweet\_userid, userid, sort = T) %>%   
 distinct(.) %>%   
 rename(weight = n) %>%   
 graph\_from\_data\_frame()  
  
pre\_mh17\_network\_graph\_df <- igraph::as\_data\_frame(pre\_mh17\_network\_data, what = "both")   
  
mh17\_accounts\_nodes <- pre\_mh17\_network\_graph\_df$vertices %>%   
 mutate(degree = igraph::degree(pre\_mh17\_network\_data),  
 strength = igraph::strength(pre\_mh17\_network\_data))  
  
mh\_17\_top\_100 <- mh17\_accounts\_nodes %>%   
 arrange(desc(strength)) %>%   
 top\_n(strength, n = 50)  
  
  
influential\_MH17\_tweets <- tweets %>%   
 filter(tweet\_time > as.Date("2014-07-17") & tweet\_time < as.Date("2014-08-01")) %>% #filter for tweets in two weeks following crash  
 filter(userid %in% mh\_17\_top\_100$name) %>% #filter for influential accounts  
 select(tweet\_text, userid, tweet\_time) %>%  
 mutate(rows = row\_number())  
  
influential\_MH17\_tokens <- influential\_MH17\_tweets %>%   
 unnest\_tokens(word, tweet\_text, token = "tweets") %>% #unnest tweets into single word format  
 filter(!word %in% stopwords\_ru$word) %>% #remove stopwords  
 filter(!word %in% english\_stopwords$word) %>%   
 filter(!str\_detect(word, "^@")) %>%   
 filter(!str\_detect(word, "^http")) %>%   
 filter(!word == "rt") %>%   
 mutate(word = SnowballC::wordStem(word, language = "russian")) %>% #stem words  
 filter(!word %in% extra\_stop\_words) #now remove some common stopword stems

## Using `to\_lower = TRUE` with `token = 'tweets'` may not preserve URLs.

influential\_MH17\_tweets\_rebuilt <- influential\_MH17\_tokens %>%   
 group\_by(rows) %>%   
 summarise(tweet\_text = paste(word, collapse = " ")) %>% #now I re-construct the tweets based on the row\_number variable  
 left\_join(influential\_MH17\_tokens,  
 by = 'rows') %>%   
 select(-word) %>%   
 distinct(rows, .keep\_all = TRUE) %>%   
 left\_join(mh\_17\_top\_100,   
 by = c('userid' = 'name')) #now I have added account details, such as strength  
  
  
## our dataframe now looks like this:  
  
  
influential\_MH17\_tweets\_rebuilt %>% head()

## # A tibble: 6 x 6  
## rows tweet\_text userid tweet\_time degree strength  
## <int> <chr> <chr> <dttm> <dbl> <dbl>  
## 1 1 новост росс прекр… 7978eae32b842d0b… 2014-07-17 23:51:00 146 243  
## 2 2 этокрут украин жд… f22e2d20ba79d712… 2014-07-18 13:56:00 224 635  
## 3 3 документ ukraine … 1b8d9c46fd475a91… 2014-07-25 11:27:00 149 248  
## 4 4 украин неймет про… 7978eae32b842d0b… 2014-07-18 18:54:00 146 243  
## 5 5 любопытн стат пов… 78483e43d73c356f… 2014-07-18 19:26:00 155 252  
## 6 6 санктпетербург ра… 3f7b3d0bcb5d86a1… 2014-07-29 17:57:00 84 1800

OK now it’s time for some machine learning! The aim of this section is to better understand what the IRA accounts were talking about. Literature suggests there could be a number of possible outcomes. Firstly, the accounts are not directly talking about political events but ‘cheerleading’ for the regime. Secondly, the accounts are aiming to distract away from events through creating ‘meaninglessness’. Finally, that the accounts are specifically addressing issues in order to frame them a particular way.

processed\_tweets <- textProcessor(influential\_MH17\_tweets\_rebuilt$tweet\_text, metadata = influential\_MH17\_tweets\_rebuilt)

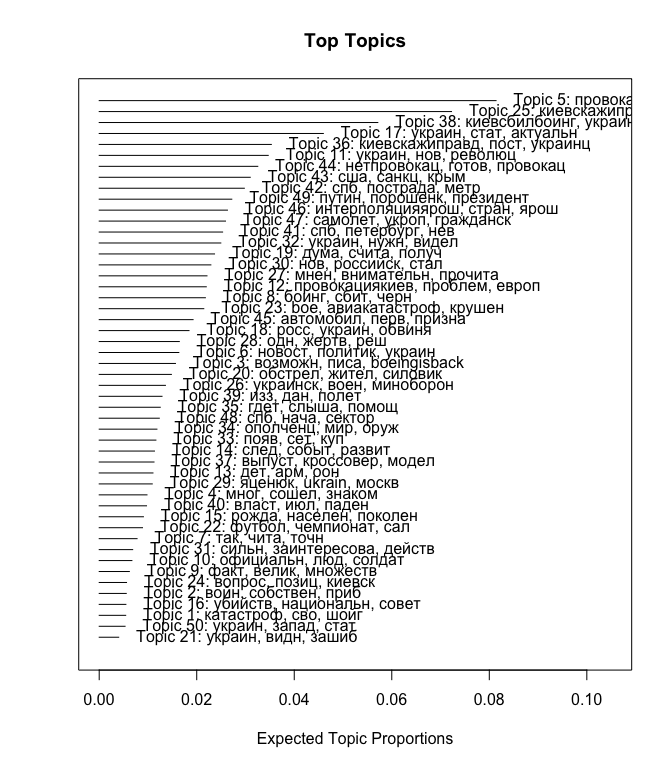
## Building corpus...   
## Converting to Lower Case...   
## Removing punctuation...   
## Removing stopwords...   
## Removing numbers...   
## Stemming...   
## Creating Output...

#plotRemoved(processed\_tweets$documents, lower.thresh = seq(1, 100, by = 10)) #this gives an idea of the threshold for removing infrequent numbers   
  
out <- prepDocuments(processed\_tweets$documents, processed\_tweets$vocab, processed\_tweets$meta) #run out #no lower threshold

## Removing 2977 of 7116 terms (2977 of 79492 tokens) due to frequency   
## Removing 72 Documents with No Words   
## Your corpus now has 12894 documents, 4139 terms and 76515 tokens.

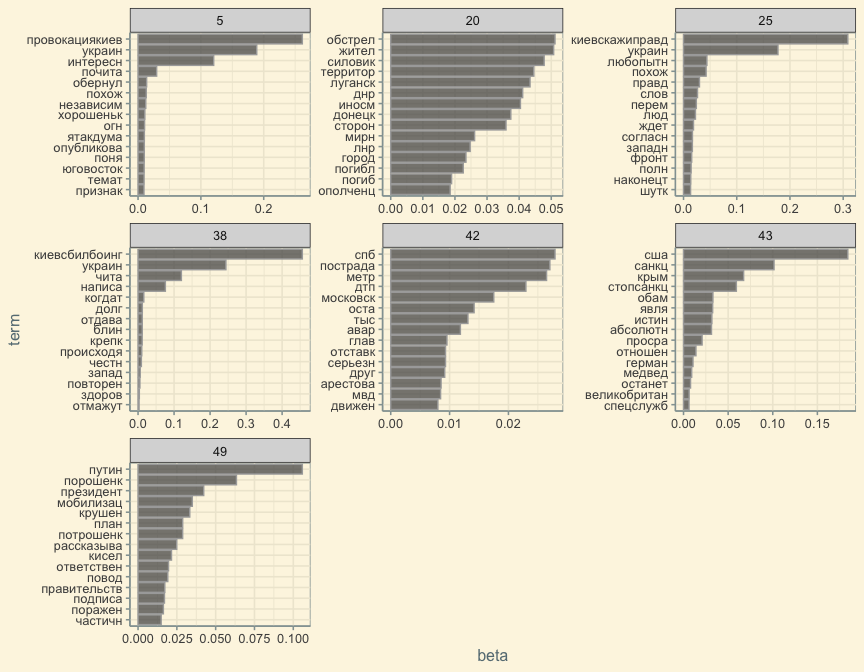
docs <- out$documents  
vocab <- out$vocab  
meta <- out$meta  
out$meta$tweet\_time <- as.numeric(out$meta$tweet\_time - min(out$meta$tweet\_time)) #need to convert dates to numeric

stm\_mh17\_accounts <- stm(documents = out$documents, vocab= out$vocab, K = 50, prevalence = ~s(tweet\_time) + strength, data = out$meta, init.type = "Spectral", seed = 1234, verbose = F, emtol = 1e-4)   
  
plot(stm\_mh17\_accounts, type = "summary", xlim = c(0, 0.105), cex = 2)



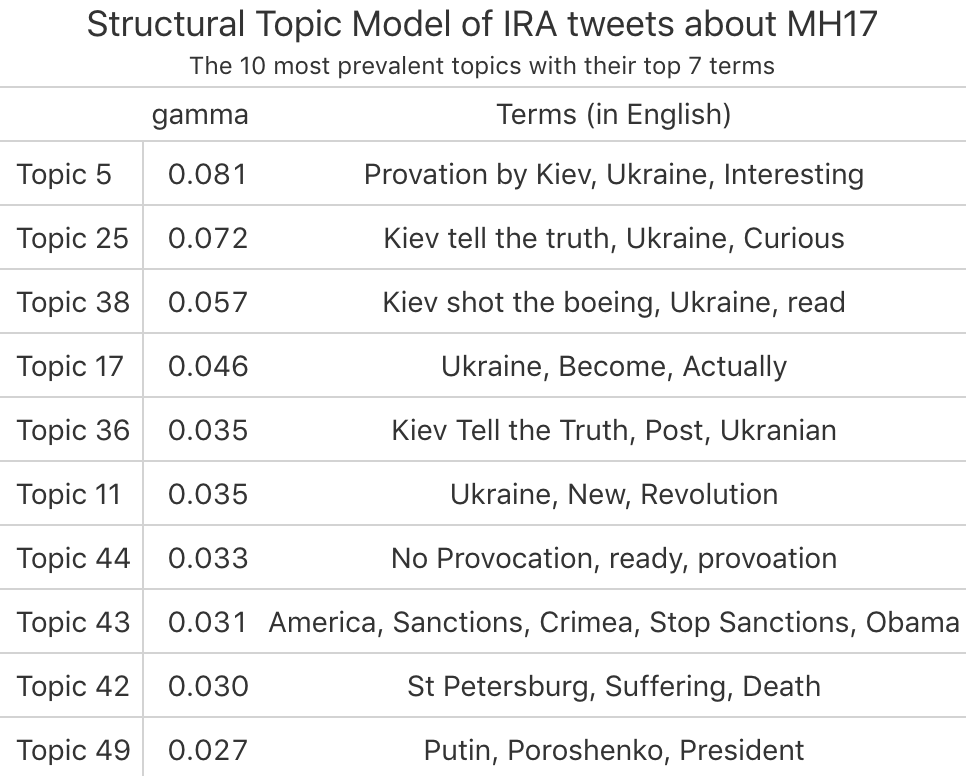
Key words for interesting topics can be seen a bit more clearly here:

tidy\_stm\_beta <- tidy(stm\_mh17\_accounts, matrix = "beta")  
  
tidy\_stm\_beta\_topics <- tidy\_stm\_beta %>%   
 filter(topic %in% c(5, 25, 38, 43, 49, 20, 42)) %>%   
 group\_by(topic) %>%   
 top\_n(15, beta) %>%   
 ungroup() %>%   
 arrange(topic, -beta)  
  
beta\_12\_graph <- tidy\_stm\_beta\_topics %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(term, beta, color = as.factor(topic))) +  
 geom\_col(show.legend = FALSE, alpha = 0.75, colour = "dark grey") +  
 coord\_flip() +  
 tidytext::scale\_x\_reordered() +  
 facet\_wrap(~topic, scales = "free", ncol = 3) +  
 theme\_solarized()   
  
beta\_12\_graph



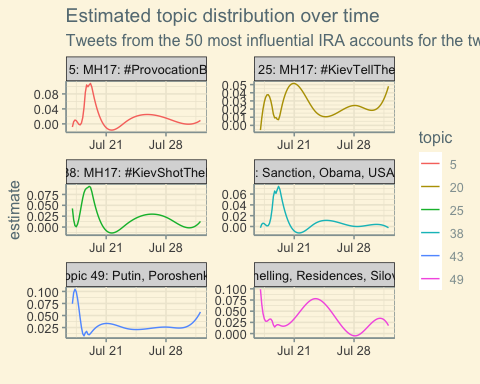
model viz

MH17\_sparse <- influential\_MH17\_tokens %>%   
 count(rows, word) %>%   
 cast\_sparse(rows, word, n)  
  
  
beta\_top\_terms <- tidy\_stm\_beta %>%  
 arrange(beta) %>%  
 group\_by(topic) %>%  
 top\_n(7, beta) %>%  
 arrange(-beta) %>%  
 select(topic, term) %>%  
 summarise(terms = list(term)) %>%  
 mutate(terms = map(terms, paste, collapse = ", ")) %>%   
 unnest(cols = c(terms))  
  
tidy\_stm\_gamma <- tidy(stm\_mh17\_accounts, matrix = "gamma", document\_names = rownames(MH17\_sparse))  
  
gamma\_terms <- tidy\_stm\_gamma %>%  
 group\_by(topic) %>%  
 summarise(gamma = mean(gamma)) %>%  
 arrange(desc(gamma)) %>%  
 left\_join(beta\_top\_terms, by = "topic") %>%   
 filter(topic %in% c('5', '25', '38', '17', '36', '11', '44', '43', '49', '42')) %>% #remove if not needed - filters out top topics  
 mutate(topic = paste0("Topic ", topic),  
 topic = reorder(topic, gamma))  
  
  
English\_terms <- c('Provation by Kiev, Ukraine, Interesting',   
 'Kiev tell the truth, Ukraine, Curious',  
 'Kiev shot the boeing, Ukraine, read',  
 'Ukraine, Become, Actually',  
 'Kiev Tell the Truth, Post, Ukranian',  
 'Ukraine, New, Revolution',  
 'No Provocation, ready, provoation',  
 'America, Sanctions, Crimea, Stop Sanctions, Obama',  
 'St Petersburg, Suffering, Death',  
 'Putin, Poroshenko, President')  
   
  
gt\_stm\_terms <- data.frame(gamma\_terms, English\_terms) %>%   
 select(-terms) %>%   
 rename('Terms (in English)' = English\_terms) %>%   
 gt(rowname\_col = "topic") %>%   
 tab\_header(  
 title = "Structural Topic Model of IRA tweets about MH17",  
 subtitle = "The 10 most prevalent topics with their top 7 terms"  
 )



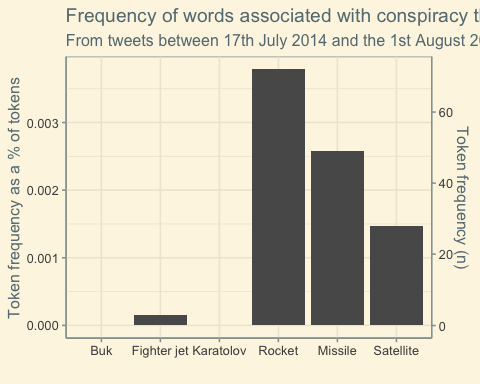
Estimating topic effect over time

prep <- estimateEffect( ~s(tweet\_time) + strength, stm\_mh17\_accounts, meta = out$meta, uncertainty = "Global")  
  
  
effect <- tidystm::extract.estimateEffect(prep, "tweet\_time", method = "continuous") #tidy  
  
#relabel our facet labels  
variable\_names <- list(   
 "5" = "Topic 5: MH17: #ProvocationByKiev" ,  
 "25" = "Topic 25: MH17: #KievTellTheTruth" ,  
 "38" = "Topic 38: MH17: #KievShotTheBoeing",  
 '43' = 'Topic 43: Sanction, Obama, USA, Crimea',  
 '49' = 'Topic 49: Putin, Poroshenko',  
 '20' = 'Topic 20: Shelling, Residences, Silovik, Donetsk'  
)  
  
  
#make a function that acts inside the call to labeller in the facet\_wrap() function  
variable\_labeller <- function(variable,value){  
 return(variable\_names[value])  
}  
  
  
topic\_dist\_over\_time\_graph <- effect %>%   
 mutate(covariate.value = as.POSIXct(covariate.value, origin = "2014-07-17 ")) %>% #convert back from numeric into date  
 filter(topic %in% c(5, 25, 38, 43, 49, 20)) %>% #select desired topics  
 mutate(topic = as.factor(topic)) %>%   
 ggplot(aes(x = covariate.value, y = estimate, color = topic)) +  
 geom\_line() +  
 facet\_wrap(~topic, labeller=variable\_labeller, ncol = 2, scales = "free") + #facet and label   
 theme\_solarized() +  
 labs(x = "", title = "Estimated topic distribution over time", subtitle = "Tweets from the 50 most influential IRA accounts for the two week period after 14th July 2014")  
  
   
  
topic\_dist\_over\_time\_graph



Lets see the extent to which the MH17 tweets also picked up on the conspiracies being promoted within the media. The topic model did not suggest this was the case, but I want to manually explore whether this is the case.

conspiracy\_words <- c('Каратолов','ракета', 'снаряд', 'истребитель', 'спутник')  
conspiracy\_words <- paste(conspiracy\_words, collapse= "|")  
tokens <- c('Каратолов', 'бук')  
n <- as.numeric(0, 0)  
pcnt <- as.numeric(0, 0)  
  
  
conspiracy\_tokens <- tweets %>%   
 select(tweet\_time, tweet\_text) %>%   
 filter(tweet\_time > as.Date("2014-07-17") & tweet\_time < as.Date("2014-08-01")) %>%   
 unnest\_tokens(tokens, tweet\_text, token = "words")  
  
conspiracy\_tokens <- conspiracy\_tokens %>%   
 mutate(obs = nrow(.))  
  
labels = data.frame(tokens = c("бук",   
 'истребител',   
 'Картаполов',  
 "ракет",  
 "снаряд"),  
 len = c('0', '3','0', '72', '49'))  
  
  
  
conspiracy\_tokens\_graph <- conspiracy\_tokens %>%   
 filter(str\_detect(tokens, conspiracy\_words)) %>%   
 mutate(tokens = SnowballC::wordStem(tokens, language = "russian")) %>%   
 dplyr::mutate(tokens = dplyr::recode(tokens,   
 'спутников' = 'спутник',   
 'биоспутник' = 'спутник')) %>%   
 add\_count(tokens) %>%   
 dplyr::distinct(n, .keep\_all =T) %>%   
 mutate(pcnt = n / obs \* 100) %>%   
 select(-tweet\_time, -obs) %>%   
 rbind(data.frame(tokens, n, pcnt)) %>%   
 ggplot(aes(tokens, pcnt)) +  
 geom\_col() +  
 labs(y = "Token frequency as a % of tokens", x = "", title = "Frequency of words associated with conspiracy theories", subtitle = "From tweets between 17th July 2014 and the 1st August 2014") +  
 theme\_solarized() +  
 scale\_x\_discrete(labels=c("бук" = "Buk",   
 'истребител' = 'Fighter jet',   
 'Каратолов'= 'Karatolov',  
 "ракет" = "Rocket",  
 "снаряд" = "Missile",  
 'спутник' = 'Satellite')) +  
 scale\_y\_continuous(sec.axis = sec\_axis(~ . \*19013.2, name = "Token frequency (n)"))  
conspiracy\_tokens\_graph

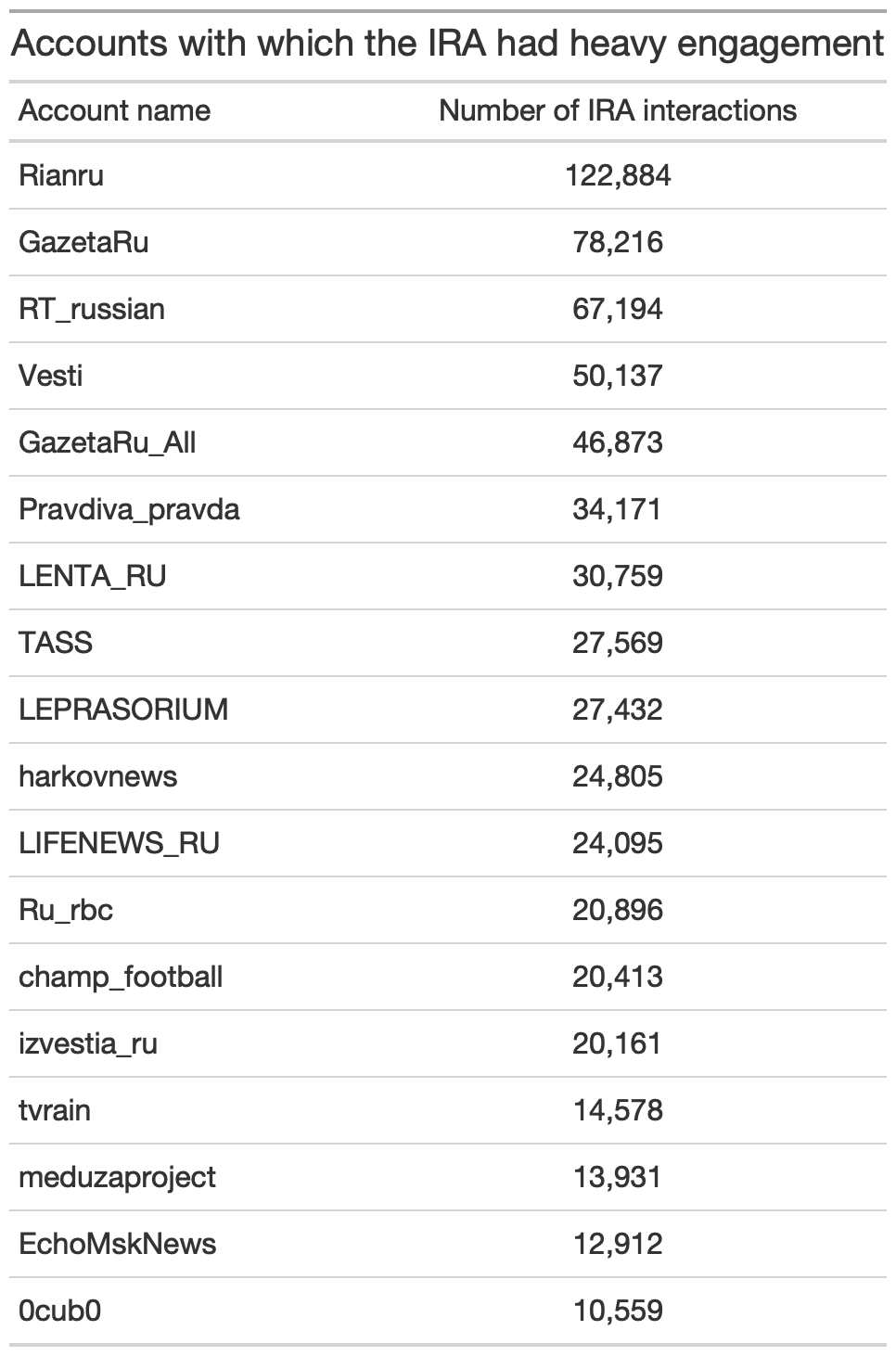


Are there any influential accounts that aren’t included in the IRA list?

non\_IRA\_influential\_accounts <- tweets %>%   
 select(userid, retweet\_userid) %>%   
 na.omit() %>%   
 count(retweet\_userid, sort = T) %>%   
 dplyr::top\_n(100) %>%   
 filter(!retweet\_userid %in% tweets$userid) %>%   
 filter(n > 10000)

## Selecting by n

non\_IRA\_influential\_accounts$account\_name <- c('Rianru', 'GazetaRu', 'RT\_russian', 'Vesti', 'GazetaRu\_All', 'Pravdiva\_pravda', 'LENTA\_RU', 'TASS', 'LEPRASORIUM', 'harkovnews', 'LIFENEWS\_RU', 'Ru\_rbc', 'champ\_football', 'izvestia\_ru', 'tvrain', 'meduzaproject', 'EchoMskNews', '0cub0')  
  
non\_IRA\_influential\_accounts\_table <- non\_IRA\_influential\_accounts %>%   
 select(account\_name, n) %>%   
 dplyr::rename('Account name' = account\_name,  
 'Number of IRA interactions' = n) %>%   
 gt() %>%   
 fmt\_number(  
 columns = 'Number of IRA interactions',  
 sep\_mark = ",",  
 decimals = F  
 ) %>%   
 tab\_header(  
 title = "Accounts with which the IRA had heavy engagement"  
 )  
  
#gtsave(non\_IRA\_influential\_accounts\_table, 'IRA\_accounts.png')



As all of these accounts are identifiable as not being run by the IRA

This provides some diagnostic analysis of our model to inform our selection of the number of clusters. As these processes can be quite memory intensive (due to parallel processing creating more than one instance of R) I am going to remove some objects from my environment first.

rm(tweet\_dates\_graph)   
rm(competency\_tweets)  
rm(competency\_tokens)  
rm(path)

## Warning in rm(path): object 'path' not found

rm(retweet\_network\_graph)  
rm(tweet\_dates\_graph)

## Warning in rm(tweet\_dates\_graph): object 'tweet\_dates\_graph' not found

rm(tweets)  
  
### ok that should make sure that you / I don't run out of memory.

MH17\_sparse <- influential\_MH17\_tokens %>%   
 count(rows, word) %>%   
 cast\_sparse(rows, word, n)

many\_models <- tibble(K = c(20, 40, 50, 60, 75, 100)) %>%  
 mutate(topic\_model = future\_map(K, ~stm::stm(MH17\_sparse, K = ., verbose = FALSE))) #future\_map here is the multicore processing version of {purrr's} map(). Given there is quite a lot of data to be analysed this should speed it up a bit.

heldout <- make.heldout(MH17\_sparse)  
  
k\_result <- many\_models %>%  
 mutate(exclusivity = map(topic\_model, exclusivity),  
 semantic\_coherence = map(topic\_model, semanticCoherence, MH17\_sparse),  
 eval\_heldout = map(topic\_model, eval.heldout, heldout$missing),  
 residual = map(topic\_model, checkResiduals, MH17\_sparse),  
 bound = map\_dbl(topic\_model, function(x) max(x$convergence$bound)),  
 lfact = map\_dbl(topic\_model, function(x) lfactorial(x$settings$dim$K)),  
 lbound = bound + lfact,  
 iterations = map\_dbl(topic\_model, function(x) length(x$convergence$bound)))

k\_result\_image <- k\_result %>%  
 transmute(K,  
 Exclusivity = map\_dbl(exclusivity, mean),  
 `Lower bound` = lbound,  
 Residuals = map\_dbl(residual, "dispersion"),  
 `Semantic coherence` = map\_dbl(semantic\_coherence, mean),  
 `Held-out likelihood` = map\_dbl(eval\_heldout, "expected.heldout")) %>%  
 gather(Metric, Value, -K) %>%  
 filter(Metric %in% c('Residuals', 'Semantic coherence', 'Held-out likelihood', 'Exclusivity')) %>%   
 ggplot(aes(K, Value, color = Metric)) +  
 geom\_line(size = 1.5, alpha = 0.7, show.legend = FALSE) +  
 facet\_wrap(~Metric, scales = "free\_y") +  
 labs(x = "K (number of topics)",  
 y = NULL,  
 title = "Model diagnostics by number of topics",  
 subtitle = "These diagnostics indicate that a good number of topics would be around 50")  
  
k\_result\_image

A close up of a map

Description automatically generated