Professores:
Eduardo de Rezende Francisco
Gustavo Corrêa Mirapalheta

Nome do Aluno:	***RAFAEL NONATO OLIVEIRA SILVA***

TRABALHO EM GRUPO
(CONSTRUI SOZINHO POIS NÃO HAVIA GRUPO DISPONIVEL)

Questões

Questão 1)

a. Elaborar uma análise em uma base de textos escrita em linguagem natural, a partir dos protocolos de M.T. estudados na disciplina.

Análise:

Olá Professor, por uma questão de organização ali dos grupos não tive oportunidade de faze-lo com outros colegas, então segue meu trabalho "individual". Basicamente professor, usei toda informação e scripts de sala de aula.

GIT: https://github.com/rafanonato/text-mining

Considerei o que foi demonstrado em sala de aula e abaixo segue uma análise seguindo a Lei de Zipf, foi um protocolo que achei interessante pela forma visual e analítica de visualizar os Tf (t*erm frequency*), assim escolhi para minha análise.

1. Considerações sobre Mineração de Texto

Nas aulas percebi que nos concentramos em identificar a frequência de termos individuais em um documento, juntamente com os sentimentos que essas palavras fornecem. Também percebi a importância compreender que as palavras fornecem dentro dos documentos. Nas aulas com os textos "baixados" da Austen e organizados para encontrar as frequências de termo (tf) identifica a frequência com que uma palavra ocorreu dentro dele. Descobrimos que muitas palavras comuns, como "o", "é", "para" etc., geralmente encabeçam as listas de frequência de termo. Uma abordagem para corrigir essas palavras comuns, mas de baixo contexto, é remover essas palavras usando uma lista chamada *stopwords*.

Uma outra abordagem que usamos foi a frequência inversa de documento de um termo (idf), o que diminui o peso para palavras comumente usadas e aumenta o peso para palavras que não são muito usadas em uma coleção de documentos. Onde o idf de um determinado termo (t) em um conjunto de documentos (D) é uma função do número total de documentos sendo avaliados (N) e o número de documentos onde o termo t aparece (nt) - com isso podemos combinar as estatísticas tf e idf em uma única estatística tf-idf, que calcula a frequência de um termo ajustado, confesso que foi difícil entender mas entendi no final a pra que serve.

Dado a esse contexto mais geral de preparação, consegui organizar-me para a Lei de Zipf.

2. Frequência de Termos e Lei de Zipf

A lei de Zipf estabelece que, dentro de um documento de texto em linguagem natural, a frequência de qualquer palavra é inversamente proporcional à sua classificação em uma tabela de frequência. A palavra mais frequente ocorrerá aproximadamente duas vezes mais que a segunda palavra mais frequente, três vezes mais que a terceira palavra mais frequente, etc. A lei é mais facilmente observada traçando os dados em um gráfico log-log, com os eixos sendo log (ordem de classificação) e log (frequência de termo).

Para começar a análise, encontrei uma biblioteca dos livros do Harry Potter aqui (https://github.com/bradleyboehmke/harrypotter) e decidi utilizar a exemplo das aulas com os livros clássicos da Austen e Broten que utilizou.

Após carregado a biblioteca usei alguns capítulos:

```
> philosophers_stone[1:2]
[1] "THE BOY WHO LIVED    Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank yo
u very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with
 such nonsense.    Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any ne
ck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which ca
me in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son cal
led Dudley and in their opinion there was no finer boy anywhere.    The Dursleys had everything they wanted, but they also had a secret,
 and their greatest fear was that somebody would discover it. They didn't think they could bear it if anyone found out about the Potter
```

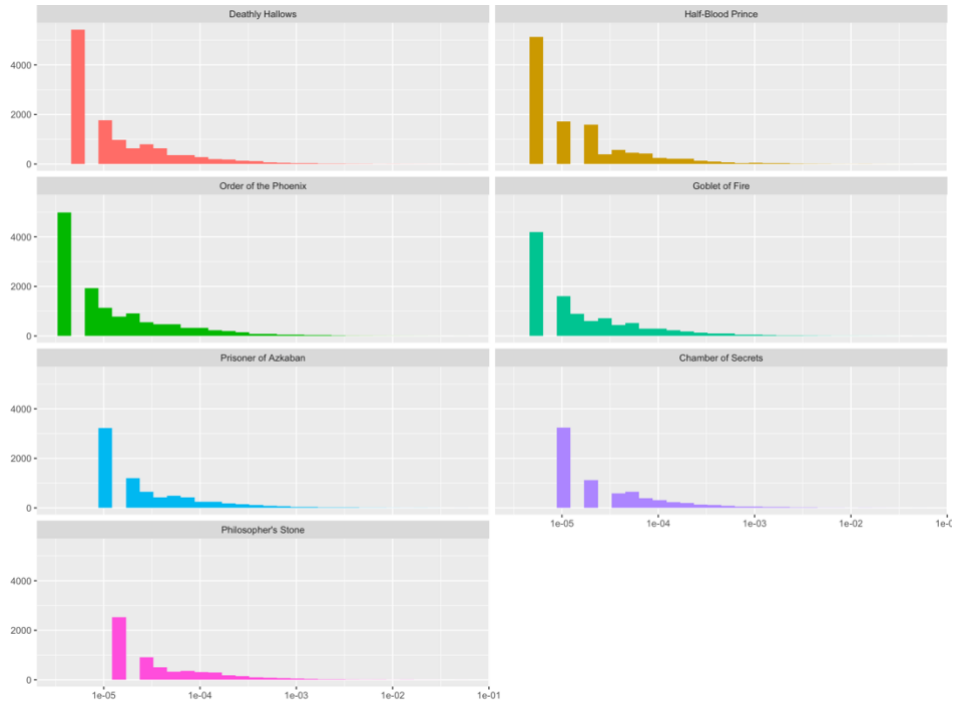
Após isso para calcular as frequências de termos, as (Term Frequency), precisamos aqui ter nossos dados em um formato organizado. Na sequência organizei alguns dos sete romances de Harry Potter em um *tibble* que tem cada palavra por capítulo por livro, assim temos as frequências, como na saída abaixo:

```
> series
# A tibble: 1,089,386 x 3
   book               chapter word
   <fct>                <int> <chr>
 1 Philosopher's Stone      1 the
 2 Philosopher's Stone      1 boy
 3 Philosopher's Stone      1 who
 4 Philosopher's Stone      1 lived
 5 Philosopher's Stone      1 mr
 6 Philosopher's Stone      1 and
 7 Philosopher's Stone      1 mrs
 8 Philosopher's Stone      1 dursley
 9 Philosopher's Stone      1 of
10 Philosopher's Stone      1 number
# … with 1,089,376 more rows
```

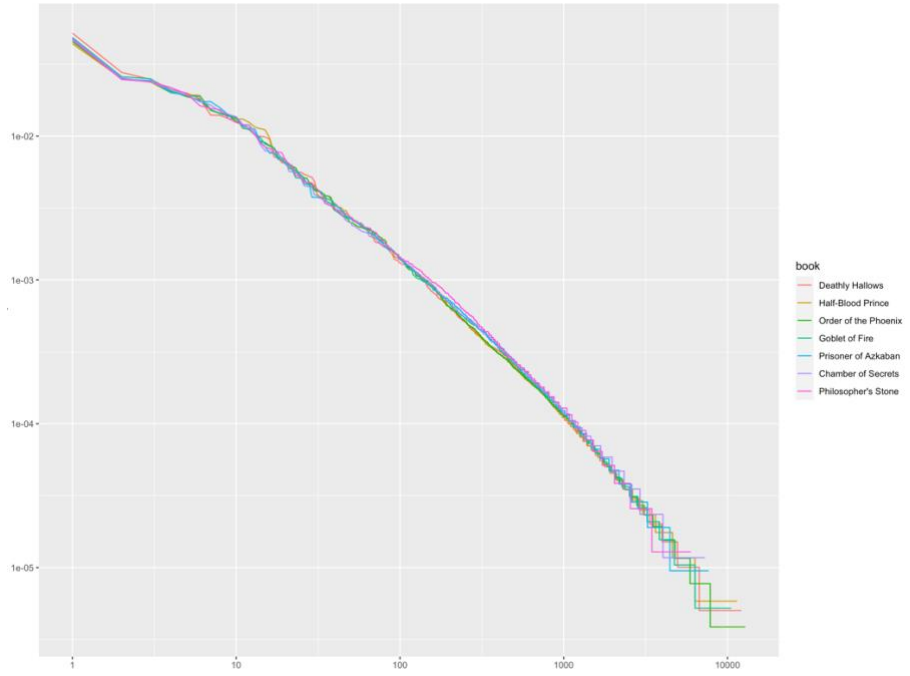
Com o texto mais organizado, podemos calcular a frequência do termo para cada palavra.

```
> book_words
# A tibble: 67,881 x 4
   book                word      n  total
   <fct>               <chr> <int>  <int>
 1 Order of the Phoenix the   11740 258763
 2 Deathly Hallows     the   10335 198906
 3 Goblet of Fire      the    9305 191882
 4 Half-Blood Prince   the    7508 171284
 5 Order of the Phoenix to     6518 258763
 6 Order of the Phoenix and    6189 258763
 7 Deathly Hallows     and    5510 198906
 8 Order of the Phoenix of     5332 258763
 9 Prisoner of Azkaban the    4990 105275
10 Goblet of Fire      and    4959 191882
# … with 67,871 more rows
```

Neste momento notamos que as palavras comuns e não contextuais (“o”, “para”, “e”, “de”, etc.) tomam completamente a lista de frequência. Podemos visualizar a distribuição da frequência para cada romance. Aqui, veremos a distribuição de *n/total*. Pude ver que existem longas barras para certas palavras extremamente comuns.



Pude ver, dentro do cenário professor, a distribuição abaixo é semelhante nos sete livros. Além disso, podemos pude comparar a distribuição a uma linha de regressão simples, o que pra mim, como disse inicialmente, é muito interessante e fácil de notar a alta correlação. Vemos que as caudas da distribuição se desviam, sugerindo que nossa distribuição não segue a lei de Zipf perfeitamente; no entanto, é próximo o suficiente para afirmar de maneira geral que a lei se aplica aproximadamente aos textos de Harry Potter assim como os de Austen.



Para encontrar o *tf-idf* onde as palavras importantes para o conteúdo de cada documento diminuindo o peso das palavras comumente usadas e aumentando o peso das palavras que não são muito utilizadas em uma coleção do texto, neste caso, a série Harry Potter. Podemos calcular facilmente o *idf* e o *tf-idf* usando a *bind_tf_idf* função fornecida pelo t*idytext* pacote.

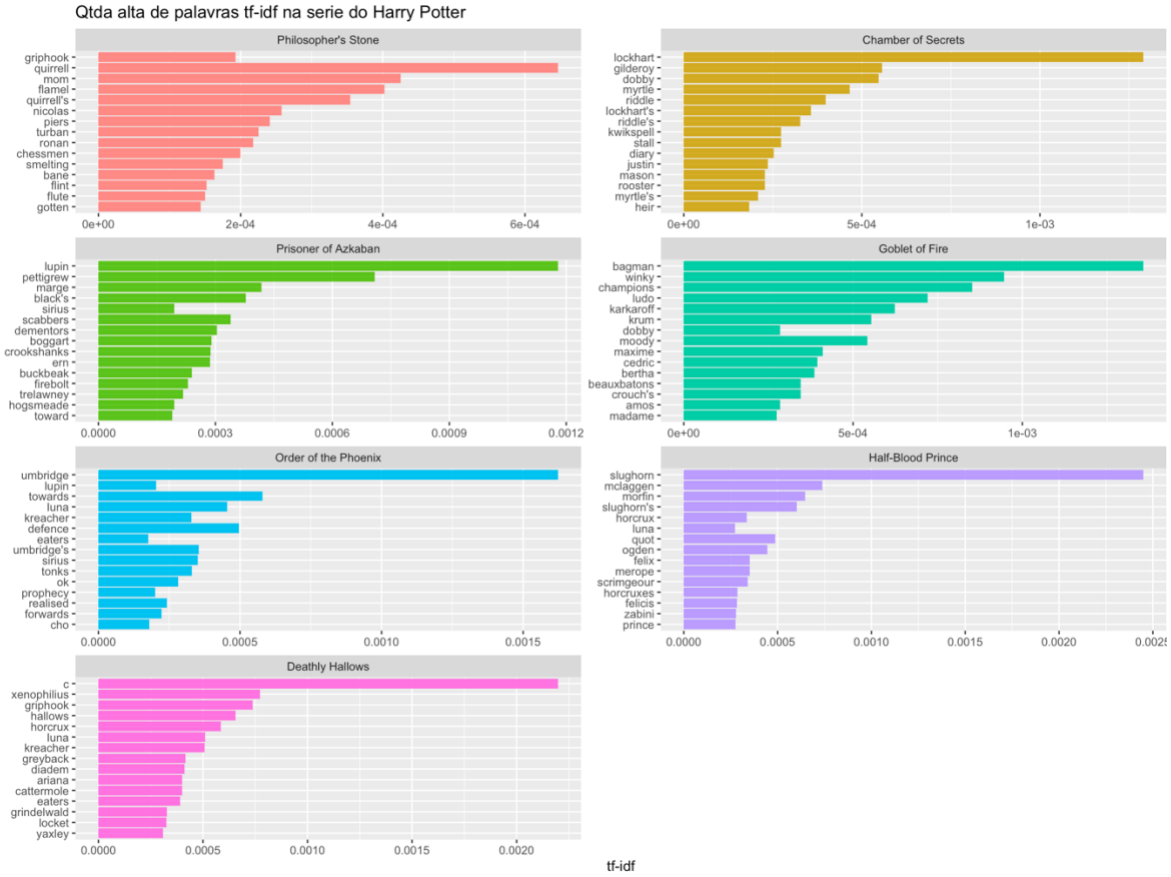
```
> book_words <- book_words %>%
+   bind_tf_idf(word, book, n)
> book_words
# A tibble: 67,881 x 7
   book                word      n  total     tf   idf tf_idf
   <fct>               <chr> <int>  <int>  <dbl> <dbl>  <dbl>
 1 Order of the Phoenix the   11740 258763 0.0454     0      0
 2 Deathly Hallows      the   10335 198906 0.0520     0      0
 3 Goblet of Fire       the    9305 191882 0.0485     0      0
 4 Half-Blood Prince    the    7508 171284 0.0438     0      0
 5 Order of the Phoenix to     6518 258763 0.0252     0      0
 6 Order of the Phoenix and    6189 258763 0.0239     0      0
 7 Deathly Hallows      and    5510 198906 0.0277     0      0
 8 Order of the Phoenix of     5332 258763 0.0206     0      0
 9 Prisoner of Azkaban  the    4990 105275 0.0474     0      0
10 Goblet of Fire       and    4959 191882 0.0258     0      0
# … with 67,871 more rows
```

Percebi que as palavras não contextuais comuns têm altos valores tf, mas seus valores idf e tf-idf são 0.

Abaixo vi que as palavras com os maiores valores tf-idf. Agora, principalmente nomes de personagens em cada livro que são exclusivos daquele livro e, portanto, usados com frequência, mas estão ausentes ou quase ausentes nos outros livros.

```
> book_words %>%
+   arrange(desc(tf_idf))
# A tibble: 67,881 x 7
   book                word          n  total      tf   idf   tf_idf
   <fct>               <chr>     <int>  <int>   <dbl> <dbl>    <dbl>
 1 Half-Blood Prince   slughorn    335 171284 0.00196  1.25  0.00245
 2 Deathly Hallows     c          1300 198906 0.00654  0.336 0.00220
 3 Order of the Phoenix umbridge   496 258763 0.00192  0.847 0.00162
 4 Goblet of Fire      bagman      208 191882 0.00108  1.25  0.00136
 5 Chamber of Secrets  lockhart    197  85401 0.00231  0.560 0.00129
 6 Prisoner of Azkaban lupin       369 105275 0.00351  0.336 0.00118
 7 Goblet of Fire      winky       145 191882 0.000756 1.25  0.000947
 8 Goblet of Fire      champions    84 191882 0.000438 1.95  0.000852
 9 Deathly Hallows     xenophilius  79 198906 0.000397 1.95  0.000773
10 Half-Blood Prince   mclaggen     65 171284 0.000379 1.95  0.000738
# … with 67,871 more rows
```

Por fim, para entender as palavras contextuais mais comuns em cada livro, dei uma olhada nas 15 principais palavras com *tf-idf* mais alta.


Qtda alta de palavras tf-idf na serie do Harry Potter
Philosopher's Stone
griphook
quirrell
mom
flamel
quirrell's
nicolas
piers
turban
ronan
chessmen
smelting
bane
flint
flute
gotten
0e+00
2e-04
4e-04
6e-04
Chamber of Secrets
lockhart
gilderoy
dobby
myrtle
riddle
lockhart's
riddle's
kwikspell
stall
diary
justin
mason
rooster
myrtle's
heir
0e+00
5e-04
1e-03
Prisoner of Azkaban
lupin
pettigrew
marge
black's
sirius
scabbers
dementors
boggart
crookshanks
ern
buckbeak
firebolt
trelawney
hogsmeade
toward
0.0000
0.0003
0.0006
0.0009
0.0012
Goblet of Fire
bagman
winky
champions
ludo
karkaroff
krum
dobby
moody
maxime
cedric
bertha
beauxbatons
crouch's
amos
madame
0e+00
5e-04
1e-03
Order of the Phoenix
umbridge
lupin
towards
luna
kreacher
defence
eaters
umbridge's
sirius
tonks
ok
prophecy
realised
forwards
cho
0.0000
0.0005
0.0010
0.0015
Half-Blood Prince
slughorn
mclaggen
morfin
slughorn's
horcrux
luna
quot
ogden
felix
merope
scrimgeour
horcruxes
felicis
zabini
prince
0.0000
0.0005
0.0010
0.0015
0.0020
Deathly Hallows
c
xenophilius
griphook
hallows
horcrux
luna
kreacher
greyback
diadem
ariana
cattermole
eaters
grindelwald
locket
yaxley
0.0000
0.0005
0.0010
0.0015
0.0020
tf-idf