

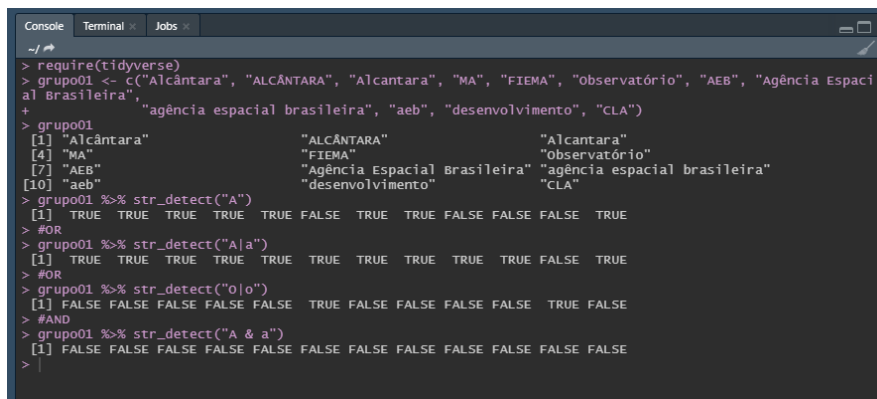
## 1 Resumo

Para o estudo de caso foi considerado que ao analisar um grupo de strings deve ser encontrado algum padrão da linguagem que está sendo utilizado. Dessa forma no momento da extração de dados é possível encontrar esses padrões e retirá-los do conjunto de dados que está sendo analisado. Para isso, foi utilizado a pacote tidyverse do R e com ele foram feitos alguns testes para verificar se o script é capaz de identificar esses padrões. Inicialmente foram feitos testes utilizando as funções do pacote tidyverse e posteriormente foram feitos testes utilizando as expressões regulares (REGEX). Com o uso do REGEX o que foi notado é que ele permite que sejam feitas análises em um texto, sendo que esse texto pode possuir expressões de diferentes formas, seja texto, data ou hora. Com o REGEX é possível buscá-las diretamente no texto. Dessa forma pode-se observar que o REGEX permite realizar:

1. Buscas, verificar se o padrão desejado se encontra no texto.
2. Validação, verificar se uma determinada sequência de caracteres segue um padrão definido.
3. Substituições, realizar as substituições por através dos padrões necessários.

### 1.1 Detecção de Padrão

Para o primeiro caso foi criado um vetor de strings e sequencialmente foi verificado se é possível obter a detecção do padrão de texto, como teste foi considerado o padrão 'A'. No grupo01 foram inseridos strings que têm relação direta com o tema da AEB. Também foi levado em consideração que a linguagem R é case sensitive. Foi feito o teste utilizando também operador lógico para verificação.



```
Console Terminal Jobs
~/R
> require(tidyverse)
> grupo01 <- c("Alcantara", "ALCANTARA", "Alcantara", "MA", "FIEMA", "Observatório", "AEB", "Agência Espaci
al Brasileira",
+ "agência espacial brasileira", "aeb", "desenvolvimento", "CLA")
> grupo01
[1] "Alcantara"          "ALCANTARA"          "Alcantara"
[4] "MA"                 "FIEMA"              "Observatório"
[7] "AEB"                "Agência Espacial Brasileira" "agência espacial brasileira"
[10] "aeb"                "desenvolvimento"    "CLA"
> grupo01 %>% str_detect("A")
[1] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE TRUE
> #OR
> grupo01 %>% str_detect("A|a")
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
> #OR
> grupo01 %>% str_detect("o|O")
[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE
> #AND
> grupo01 %>% str_detect("A & a")
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Abbildung 1: Detecção de padrão

### 1.2 Indexação

Para que seja possível realizar a localização a partir do índice é possível utilizar diretamente a função de subset que existe no pacote tidyverse, dessa forma é possível realizar a identificação do padrão através do índice.

### 1.3 Remoção de padrões

Nesse caso, o tratamento seria feito para realizar a retirada de outliers que se encontram no conjunto de dados que foi extraído. Para simulação foi criado um segundo vetor de strings que possui os três outliers que são mais encontrados na coleta do twitter do projeto para o PDI-CEA. Sendo eles 'cantora', 'jogador' e 'Thiago'. O que foi observado é que a função remove consegue detectar e retirar a palavra que foge do padrão do grupo de dados, no entanto ao realizar uma retirada de um segundo padrão em seguida ele retorna o padrão que havia sido removido anteriormente, causando um problema no processo de remoção do padrão.

```

Console Terminal Jobs
~/ #
> grupo01 <- c("Alcântara", "ALCÂNTARA", "Alcantara", "MA", "FIEMA", "Observatório", "AEB", "Agência Espacial Brasileira",
+ "agência espacial brasileira", "aeb", "desenvolvimento", "CLA")
> grupo01 %>% str_subset("A")
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "AEB"
[7] "Agência Espacial Brasileira" "CLA"
> #OR
> grupo01 %>% str_subset("A|a")
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "Observatório"
[7] "AEB"            "Agência Espacial Brasileira" "agência espacial brasileira"
[10] "aeb"            "CLA"
> #OR
> grupo01 %>% str_subset("O|o")
[1] "Observatório" "desenvolvimento"
> #AND
> grupo01 %>% str_subset("A & a")
character(0)
>

```

Abbildung 2: Indexação

```

Console Terminal Jobs
~/ #
> grupo02 <- c("Alcântara", "ALCÂNTARA", "Alcantara", "MA", "FIEMA", "Observatório", "cantora", "AEB", "Agência Espacial Brasileira",
+ "agência espacial brasileira", "aeb", "desenvolvimento", "CLA", "Thiago", "jogador", "jogador", "cantora")
> grupo02
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "Observatório"
[7] "cantora"        "AEB"            "Agência Espacial Brasileira"
[10] "agência espacial brasileira" "aeb"            "desenvolvimento"
[13] "CLA"            "Thiago"         "jogador"
[16] "jogador"        "cantora"
> grupo02 %>% str_remove_all("jogador")
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "Observatório"
[7] "cantora"        "AEB"            "Agência Espacial Brasileira"
[10] "agência espacial brasileira" "aeb"            "desenvolvimento"
[13] "CLA"            "Thiago"         ""
[16] ""              "cantora"
> grupo02 %>% str_remove("cantora")
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "Observatório"
[7] ""              "AEB"            "Agência Espacial Brasileira"
[10] "agência espacial brasileira" "aeb"            "desenvolvimento"
[13] "CLA"            "Thiago"         ""
[16] "jogador"        ""
> grupo02 %>% str_remove("Thiago")
[1] "Alcântara"      "ALCÂNTARA"      "Alcantara"
[4] "MA"             "FIEMA"          "Observatório"
[7] "cantora"        "AEB"            "Agência Espacial Brasileira"
[10] "agência espacial brasileira" "aeb"            "desenvolvimento"
[13] "CLA"            ""              "jogador"
[16] "jogador"        "cantora"
>

```

Abbildung 3: Remoção de padrões

## 1.4 Substituição de padrões

A função `replace all` permite que sejam trocados os padrões que aparecem por outro termo. Essa função pode ser utilizada mais para o tratamento de dados da wordcloud quando feito em R. Visto que nele aparecem termos como 'lançamentos' e 'lançamento' para a wordcloud é necessário apenas um dos termos, sendo assim pode ser feita a substituição.

```

Console Terminal Jobs
~/ #
> grupo03 <- c("Alcântara", "Alcantara", "lançamentos", "lançamento", "avançar", "avanços", "avançou", "avanço")
> grupo03
[1] "Alcântara"      "Alcantara"      "lançamentos"    "lançamento"     "avançar"        "avanços"        "avançou"
[8] "avanço"
> grupo03 %>% str_replace_all("lançamentos", "lançamento")
[1] "Alcântara"      "Alcantara"      "lançamento"     "lançamento"     "avançar"        "avanços"        "avançou"
[8] "avanço"
> grupo03 %>% str_replace_all("avanços|avançou|avançar", "avanço")
[1] "Alcântara"      "Alcantara"      "lançamentos"    "lançamento"     "avanço"         "avanço"         "avanço"
[8] "avanço"
>

```

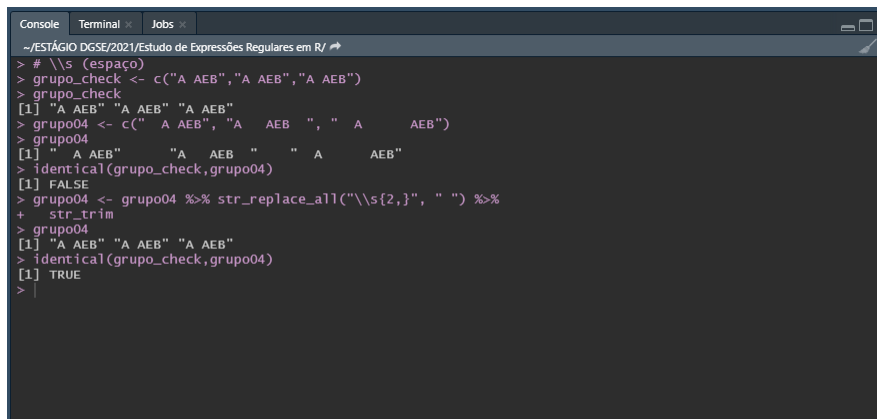
Abbildung 4: Substituição de padrões

## 2 Expressões Regulares

Para a demonstração de casos foram feitos casos no script R utilizando apenas as expressões regulares.

### 2.1 Espaço

Quando em um conjunto de dados existem espaços que fogem ao padrão de espaçamento entre palavras pode ser determinado no uso da expressão regular  $s$ , referente ao espaço. Como teste foi considerado inicialmente que existe um espaçamento maior entre as palavras, sendo o limite máximo permitido dois espaçamentos. Quando existe mais que isso o espaço deve ser substituído por um espaçamento. Além disso no grupo a ser testado foi inserido o tratamento de dados quando existe espaçamento antes do início da palavra e depois da final palavra, para esse tratamento é utilizada a função 'trim'. Esse tratamento se torna útil no momento do tratamento de dados que são coletados.

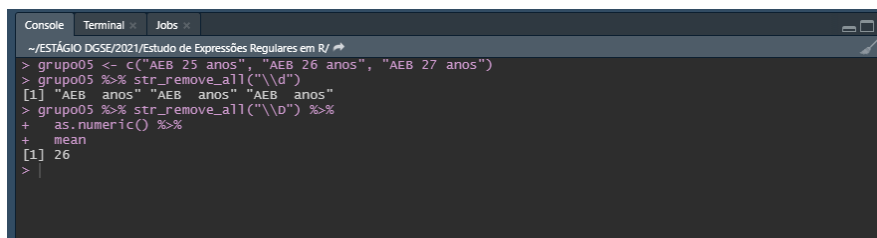


```
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> # \\s (espaço)
> grupo_check <- c("A AEB", "A AEB", "A AEB")
> grupo_check
[1] "A AEB" "A AEB" "A AEB"
> grupo04 <- c(" A AEB", "A AEB ", " A AEB")
> grupo04
[1] " A AEB" "A AEB " " A AEB"
> identical(grupo_check, grupo04)
[1] FALSE
> grupo04 <- grupo04 %>% str_replace_all("\\s{2,}", " ") %>%
+ str_trim
> grupo04
[1] "A AEB" "A AEB" "A AEB"
> identical(grupo_check, grupo04)
[1] TRUE
> |
```

Abbildung 5: Espaçamento

### 2.2 Dígitos e Não dígitos

Quando em um conjunto de dados existem dígitos e todos os termos que não são dígitos pode ser feito o isolamento de todos os dígitos através da expressão  $d$  e de todos os não dígitos através da expressão  $D$ . Quando utilizando o isolamento de todos os dígitos caso não tenha nenhuma string ou espaçamento a mais no conjunto de dados é possível transformar os termos em termos numéricos podendo assim extrair alguma informação útil do conjunto que está sendo analisado. Para realizar pesquisas que tenham dígitos é possível utilizar a



```
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> grupo05 <- c("AEB 25 anos", "AEB 26 anos", "AEB 27 anos")
> grupo05 %>% str_remove_all("\\d")
[1] "AEB anos" "AEB anos" "AEB anos"
> grupo05 %>% str_remove_all("\\D") %>%
+ as.numeric() %>%
+ mean
[1] 26
> |
```

Abbildung 6: Dígitos e não dígitos

expressão  $[0 - 9]$ , para realizar buscas por texto,  $[a - z]$ . Por meio de expressões como  $^ [a - z]$  é possível obter os textos que começam com letras ou caso seja de interesse ter textos que terminem com letras basta utilizar símbolo do dólar ao final da expressão  $[a - z] \$$ . O mesmo vale para encontrar expressões que sejam compostas por números.

### 2.3 Ponto

O ponto,  $.$ , serve para isolar qualquer termo. Nesse caso foi feito exemplo realizando uma combinação de expressões regulares a fim de demonstrar a sua funcionalidade.

```

Console Terminal Jobs x
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> grupo03
[1] "Alcantara" "Alcantara" "lançamentos" "lançamento" "avançar" "avanços" "avançou"
[8] "avanço"
> grep("[a-z]", grupo03)
[1] 1 2 3 4 5 6 7 8
> grep("[0-9]", grupo03)
integer(0)
> grupo06
[1] "5.55" "25 anos AEB."
> grep("[a-z]", grupo06)
[1] 2
> grep("[0-9]$", grupo06)
[1] 1
>

```

Abbildung 7: Agrupamento

```

Console Terminal Jobs x
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> # . -> qualquer coisa
> grupo06 <- c("5.55", "25 anos AEB.")
> grupo06
[1] "5.55" "25 anos AEB."
> grupo06 %>% str_subset("\\D\\.") # \\D -> não dígito
[1] "25 anos AEB."
> grupo06 %>% str_subset("\\d\\.") # \\d -> dígito
[1] "5.55"
> grupo06 %>% str_subset("\\.") # \\D -> não dígito
[1] "5.55" "25 anos AEB."
>

```

Abbildung 8: Ponto

## 2.4 Agrupamento

É possível realizar o agrupamento dos padrões que devem ser encontrados no grupo de texto que está sendo tratado e para isso as expressões regulares se tornam úteis. Dessa forma é possível escolher quais palavras padrões você quer visualizar. Esse agrupamento pode ser feito a partir da função `grep()`. Na opção de

```

Console Terminal Jobs x
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> # capturando por grupo de captura
> grupo03
[1] "Alcantara" "Alcantara" "lançamentos" "lançamento" "avançar" "avanços" "avançou"
[8] "avanço"
> grep("lançament[o|os]", grupo03)
[1] 3 4
> grep("Alc[â|a]ntara", grupo03)
[1] 1 2
> grep("avanç[ar|os|ou|o]", grupo03)
[1] 5 6 7 8
>

```

Abbildung 9: Agrupamento

agrupamento ele sinaliza pelo índice aonde se encontra o padrão procurado, para obter a visualização desse padrão tem-se a opção de `value = True` adicionada ao script, dessa forma os valores que correspondentes serão apresentados.

```

Console Terminal Jobs x
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> grupo03
[1] "Alcantara" "Alcantara" "lançamentos" "lançamento" "avançar" "avanços" "avançou"
[8] "avanço"
> grep("[a-z]", grupo03, value = TRUE)
[1] "Alcantara" "Alcantara" "lançamentos" "lançamento" "avançar" "avanços" "avançou"
[8] "avanço"
> grep("[0-9]", grupo03, value = TRUE)
character(0)
> grupo06
[1] "5.55" "25 anos AEB."
> grep("[a-z]", grupo06)
[1] 2
> grep("[a-z]", grupo06, value = TRUE)
[1] "25 anos AEB."
> grep("[0-9]$", grupo06)
[1] 1
> grep("[0-9]$", grupo06, value = TRUE)
[1] "5.55"
>

```

Abbildung 10: Agrupamento com value

### 3 Script de teste

O script realiza a busca por termos como alcântara e alcantara e um segundo teste foi adicionado utilizando a busca por base de alcântara. Essa escolha foi feita pois aparecem menos outliers possibilitando identificar mais rápido se o dado foi tratado ou não. Foi adicionado ao script busca alcântara o pacote tidyverse e posteriormente no momento de realizar a busca com a função search tweets foi mantida a busca por Alcântara. Tendo essa busca é feito um tratamento de forma que sejam realizadas as extrações apenas dos padrões que são necessários para o estudo. Foi utilizado inicialmente um filtro que utiliza apenas as colunas desejadas para análise, esse método foi adicionado pois ele coleta apenas as colunas de interesse para o banco de dados do aebsocialdata. No teste, foi notado que ao realizar a busca utilizando o str detect é possível identificar com true ou false se o elemento que foi colocado para busca foi encontrado ou não. Mas também foi notado que ele não permite que seja exibido a tabela com os valores utilizando apenas essa forma do str detect. As imagens adicionadas abaixo são relacionadas a busca por base de alcântara no twitter utilizando a função search tweets.

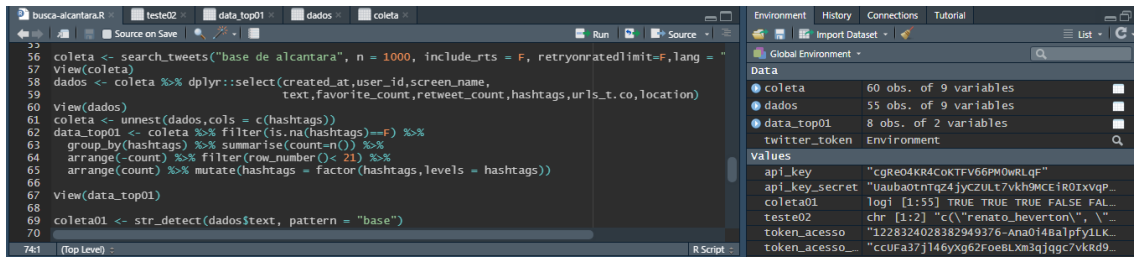


Abbildung 11: script

É possível ver que é realizada a redução de colunas no momento da coleta, sendo essa adaptação útil para o script de coleta atual.

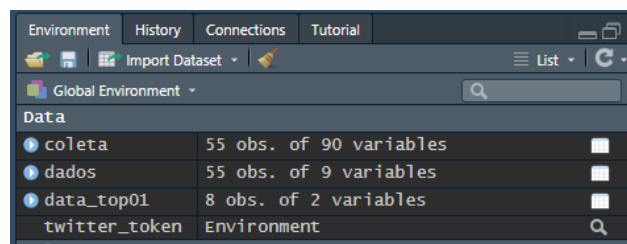


Abbildung 12: Tratamento das colunas

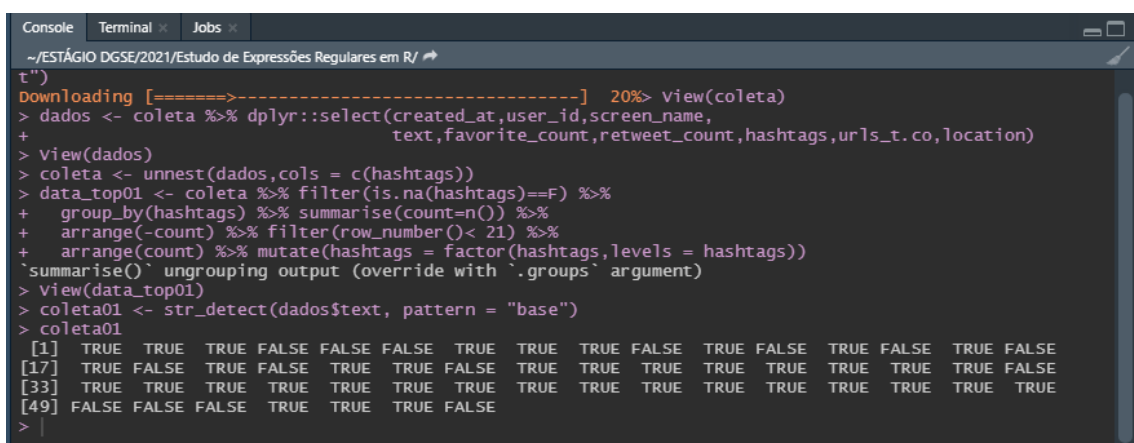


Abbildung 13: str detect

Foi feito um ranking das hashtags que foram citadas na coleta direto no script.

	hashtags	count
1	alcantara	1
2	Brasil	1
3	LulaPresidente	1
4	maranhão	1
5	projetoVLM1	1
6	SantaMaria	1
7	satelite	1
8	TáNoGMC	1

Abbildung 14: hashtags

## 4 Conclusão

Diante do que foi apresentado, verifica-se a possibilidade de inserir algumas expressões regulares no script de coleta do projeto sobre o PDI-CEA. Visando na detecção de expressões que são de interesse do projeto e manter apenas ela no script ou da retirada de outliers. As expressões seriam utilizadas de maneira preferencial no momento da coleta e ao invés de ser salvo o conjunto de textos e realizar o tratamento depois, no momento da coleta realizar esse tratamento com expressões regulares e salvar o arquivo já tratado no script principal. São necessárias adaptações no script para que ele funcione de maneira ideal para o projeto.