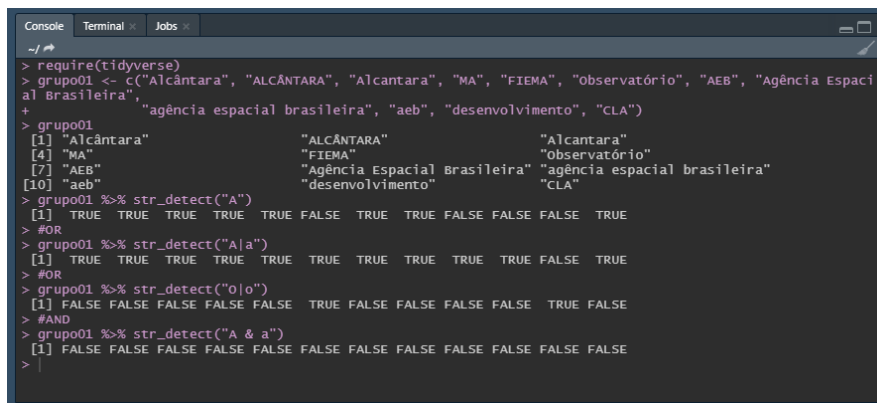


1 Resumo

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).

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.

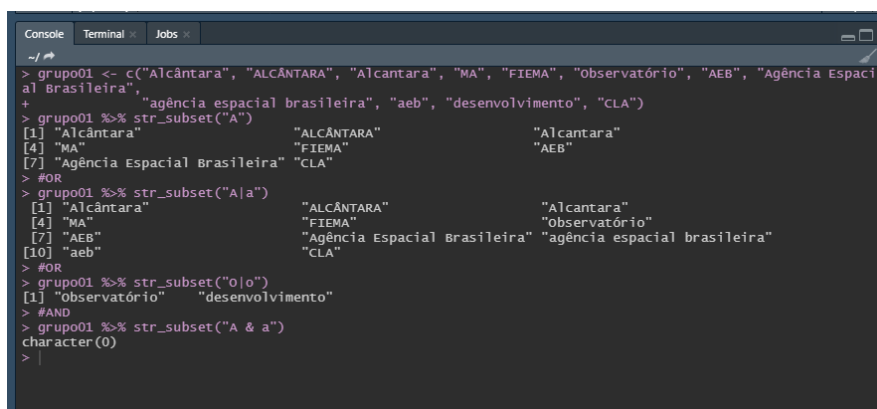


```
Console Terminal Jobs x
~/
> require(tidyverse)
> grupo01 <- c("Alcantara", "ALCANTARA", "Alcantara", "MA", "FIEMA", "Observatório", "AEB", "Agência Espaci
al Brasileira", "agência espacial brasileira", "aeb", "desenvolvimento", "CLA")
+
> grupo01
[1] "Alcantara"          "ALCANTARA"          "Alcantara"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

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.

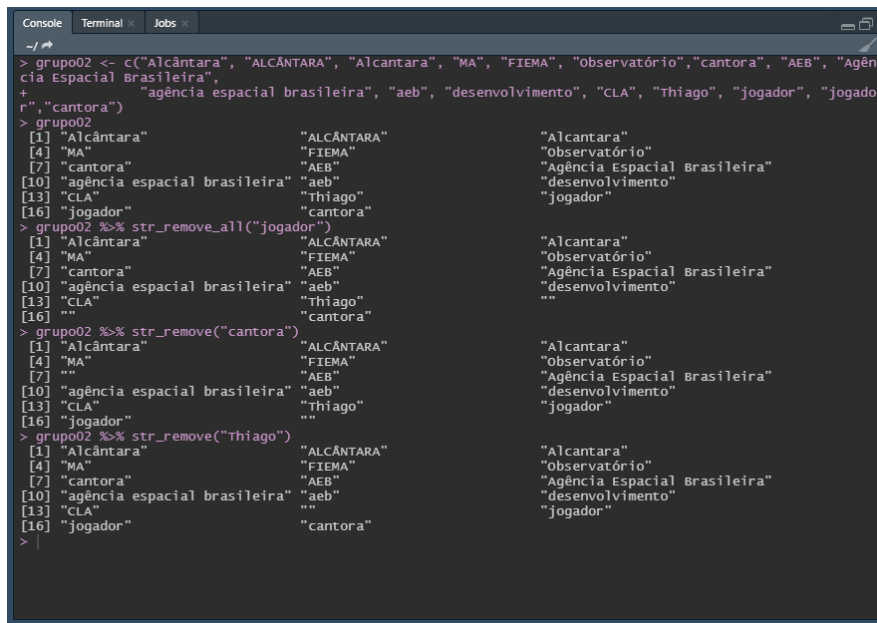


```
Console Terminal Jobs x
~/
> grupo01 <- c("Alcantara", "ALCANTARA", "Alcantara", "MA", "FIEMA", "Observatório", "AEB", "Agência Espaci
al Brasileira", "agência espacial brasileira", "aeb", "desenvolvimento", "CLA")
+
> grupo01 %>% str_subset("A")
[1] "Alcantara"          "ALCANTARA"          "Alcantara"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

Abbildung 2: Indexação

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.

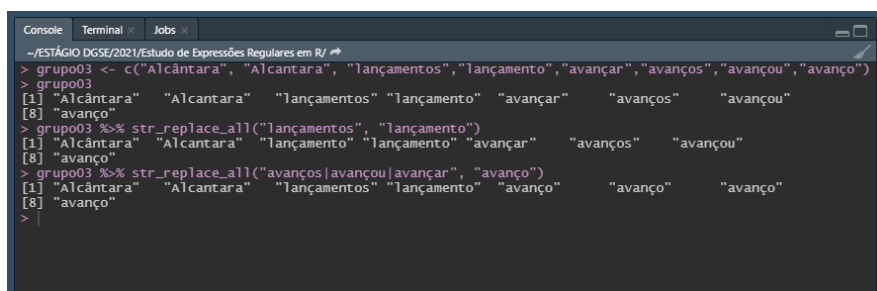


```
Console Terminal Jobs x
~/
> grupo02 <- c("Alcântara", "ALCÂNTARA", "Alcantara", "MA", "FIEMA", "observatório", "cantora", "AEB", "Agên
cia Espacial Brasileira",
+ "agência espacial brasileira", "aeb", "desenvolvimento", "CLA", "Thiago", "jogador", "jogado
r", "cantora")
> grupo02
[1] "Alcântara"          "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] ""
> 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"             "jogador"
[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 x
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/
> 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

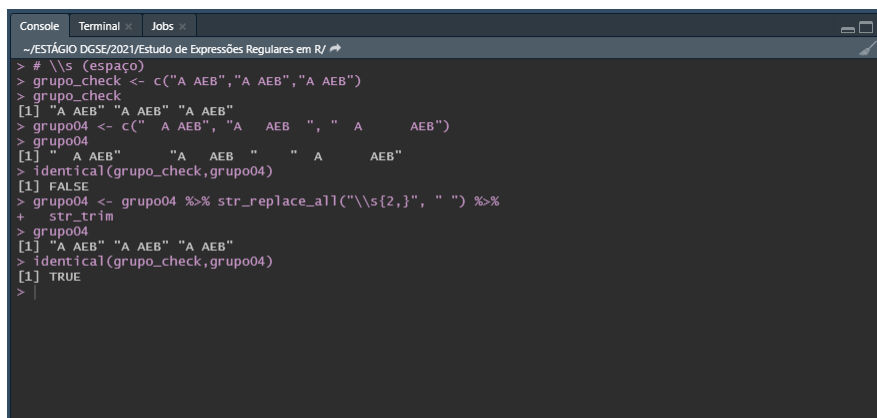
2 Expressões Regulares

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 do final palavra, para esse tratamento é utilizada a função 'trim'. Esse tratamento se torna útil no momento do tratamento de dados que são coletados.

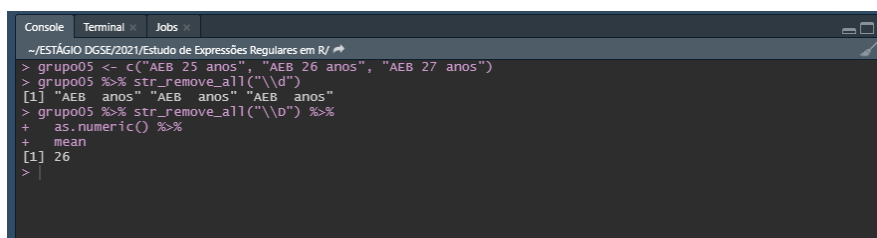


```
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> # \s (espaço)
> grupo_check <- c("A AEB", "A AEB", "A AEB")
> grupo_check
[1] "A AEB" "A AEB" "A AEB"
> grupo04 <- c(" A AEB", "A AEB ", " A AEB ")
> grupo04
[1] " A AEB" "A AEB " " A AEB "
> identical(grupo_check, grupo04)
[1] FALSE
> 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.

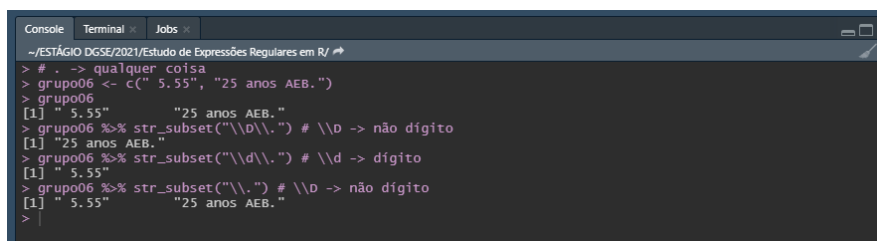


```
~/ESTÁGIO DGSE/2021/Estudo de Expressões Regulares em R/ ➤
> grupo05 <- c("AEB 25 anos", "AEB 26 anos", "AEB 27 anos")
> grupo05 %>% str_remove_all("\\d")
[1] "AEB anos" "AEB anos" "AEB anos"
> grupo05 %>% str_remove_all("\\D") %>%
+   as.numeric() %>%
+   mean
[1] 26
> |
```

Abbildung 6: Dígitos e não dígitos

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.



```
~/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 7: Ponto

3 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 ou na detecção de expressões que são de interesse do projeto e

manter apenas ela no script ou da retirada de outliers.