## WordCloud

Decidi criar uma **Nuvem de Palavras** utilizando a própria linguagem R para utilizar como Capa do **LinkedIn**. Selecionei à dedo termos relacionado à Ciência de dados que fazem parte do meu dia a dia.

Utilizei como base a documentação oficial neste link.

900

500

500

500

## 3 SQL

## 6 Spark

## 4 Statistics

## 5 Big Data

Para começar temos que instalar os pacotes e importar as bibliotecas.

Vamos iniciar importando o csv que eu criei contendo as palavras que escolhi e o peso que atribuí à elas. O peso influencia o tamanho das palavras na imagem final. Ordenei pelo peso de forma decrescente, pois a função que monta a nuvem de palavras vai iniciar pelas palavras de maior peso e a organização vai ficar mais atrativa.

```
df <- read_delim("wordcloud.csv", delim=";") %>% arrange(desc(freq))
## Rows: 147 Columns: 2
## -- Column specification ------
## Delimiter: ";"
## chr (1): words
## dbl (1): freq
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
df %>% head()
## # A tibble: 6 x 2
##
  words
               freq
  <chr>
##
               <dbl>
## 1 Data Science 1200
## 2 R
                 900
```

- Na função informei a base com as palavras e pesos, através do parâmetro size adequei o tamanho das palavras.
- Para as cores, gostei mais do fundo preto com cores claras para as palavras para melhorar o contraste.
- Para o formato da nuvem, me agradou o formato de "estrela" na distribuição das palavras (existem outros formatos: 'circle', 'cardioid', 'diamond', 'triangleforward', 'triangle', 'pentagon')
- E por fim, rotacionei as palavras todas na mesma direção ao invés de uma ordem mais irregular como é por padrão.

```
my_graph <-
wordcloud2(df
    ,size=0.6
    ,color='random-light'
    ,backgroundColor="black"
    ,shape = "star"
    ,minRotation = -pi/6
    ,maxRotation = -pi/6
    ,rotateRatio = 1)
# my_graph</pre>
```

Por padrão a função gera um resultado em html, então temos uma maneira de salvar esse arquivo html e tirar um "screeshot" para gerar um arquivo do tipo imagem (png). Utilizei o tamanho de imagem mais utilizado no LinkedIn. Vale lembrar que a cada execução muda um pouco a ordem e cores de forma aleatória.

```
saveWidget(my_graph,"tmp.html",selfcontained = F)
webshot("tmp.html","result.png", delay = 5, vwidth = 1584, vheight=396)
```

