

# Trabalho AV1

Erick Brito e Vinícius Mitsuo

## Projeto

Os Esportes Eletrônicos ou apenas eSports, como é mundialmente conhecido, vêm dominando o mundo gamer e tecnológico. Numa definição simples, os eSports giram em torno de competições de jogos eletrônicos em que atletas profissionais disputam partidas, seja presencial ou on-line. Na maioria delas, os espectadores estão presentes no local do evento ou em plataformas de streaming.

Dado o crescimento dos Esportes Eletrônicos, assim como o crescimento desse mercado, decidimos então realizar uma análise exploratória de dados e visualizações em uma base de dados de partidas oficiais de um jogo específico, chamado Counter Strike: Global Offensive, ou CS:GO para os mais íntimos.

## Motivação

A sensação dos Esportes Eletrônicos fez com que cada vez mais empresas/organizações se interessassem e investissem nesse mercado. No caso do CS:GO não é diferente.

Hoje em dia existem organizações muito grandes participando do cenário competitivo de CS:GO, como por exemplo, Team Liquid, FaZe Clan, TSM, entre outras mais, que são muito valiosas e possuem times profissionais excelentes. Essas organizações estão constantemente alterando o elenco de seus times e procurando um meio de se sobressair no cenário competitivo.

Alterar um time, buscar uma melhora técnica em jogo, criar táticas, tudo isso depende de informações, depende de estar seguindo o meta – Meta pode ser usado como um acrônimo para “most effective tactics available”, em português “táticas mais eficazes disponíveis”, e chamar algo de “meta” significa que é uma maneira eficaz de atingir o objetivo do jogo, seja vencer outros jogadores ou vencer o próprio jogo.

Portanto, uma análise de dados trazendo informações sobre o jogo (parte tática e mecânica) e sobre os jogadores se faz necessária e é de interesse de organizações/empresas que estão nesse meio. Daí, nossa motivação para começar esse projeto.

## Base de Dados

A base de dados que escolhemos está disponível em CS:GO Professional Matches.

Nessa base temos as seguintes tabelas:

- **results.csv**: com as pontuações do mapa e classificações da equipe
- **picks.csv**: com a ordem de escolha de mapas e vetos no processo de seleção de mapas.
- **economy.csv**: com o valor do equipamento inicial da rodada para todas as rodadas jogadas em um partida
- **players.csv**: com as performances individuais dos jogadores em cada mapa.

## Lendo as Bases

Usando a função `read.csv2()` do R Base, conseguimos ler e salvar em variáveis as nossas tabelas.

```
### Usando uma função do R Base para ler os csvs e assim os armazenamos nas seguintes variáveis
players <- read.csv2("players.csv", sep=",")
results <- read.csv2("results.csv", sep=",")
picks <- read.csv2("picks.csv", sep=",")
economy <- read.csv2("economy.csv", sep=",")
```

```
### Transformando em tibble
players <- as_tibble(players)
results <- as_tibble(results)
picks <- as_tibble(picks)
economy <- as_tibble(economy)
```

Dando uma olhada

```
head(players[1:4],1)
```

```
## # A tibble: 1 x 4
##   date      player_name team      opponent
##   <chr>    <chr>      <chr>    <chr>
## 1 2020-02-26 Brehze      Evil Geniuses Liquid
```

## Limpeza dos dados

Analisando os dados, percebemos que haviam alguns vazios, apenas como “. Então, decidimos verificar quantas linhas com vazio tem cada coluna:

- **players\_empty\_strings**: tibble que contém a soma das linhas com “ ” na tabela **players.csv** por coluna.
- **results\_empty\_strings**: tibble que contém a soma das linhas com “ ” na tabela **results.csv** por coluna.
- **picks\_empty\_strings**: tibble que contém a soma das linhas com “ ” na tabela **picks.csv** por coluna.
- **economy\_empty\_strings**: tibble que contém a soma das linhas com “ ” na tabela **economy.csv** por coluna.

```
### Verificamos a soma de dados que estão da forma "" para cada coluna

### (Obs: a função funs() foi descontinuada, mas usando as funções similares
### recomendadas não conseguimos o mesmo resultado, algumas até geraram erros.)
```

```
players_empty_strings <- players %>%
  summarise_all(funs(sum(. == "")))

results_empty_strings <- results %>%
  summarise_all(funs(sum(. == "")))

picks_empty_strings <- picks %>%
  summarise_all(funs(sum(. == "")))

economy_empty_strings <- economy %>%
  summarise_all(funs(sum(. == "")))
```

Com isso, conseguimos ter uma noção de quantas observações(linhas) podemos remover para que assim alcancemos a base limpa.

Observando **results\_empty\_strings** e **picks\_empty\_strings** percebemos que elas já estão limpas

e prontas para uso. Mas ao observar as `players_empty_strings`(o mesmo ocorre para `economy_empty_strings`) temos, em quantidades grandes até, dados faltando em algumas colunas:

```
players_empty_strings[10:15]
```

```
## # A tibble: 1 x 6
##   best_of map_1 map_2 map_3 kills assists
##   <int> <int> <int> <int> <int> <int>
## 1      0      0 186794 314639      0      0
```

Como por exemplo as colunas `map_2` e `map_3`, que mostram quais foram os segundos e terceiros mapas jogados em uma partida. Isso só mostra que existe uma grande quantidade de partidas que não possui segundo e terceiro mapa na nossa base de dados, mas com isso já temos um direcionamento para alcançar a base limpa.

Com essas informações e usando também do nosso conhecimento sobre algumas coisas do jogo, uma limpeza que pode ser feita de já, é remover algumas colunas que já sabemos que não vão ser necessárias para a análise,

```
### Aqui estamos removendo colunas que estão com grande quantidade de dados faltando
### e que não viriam a ser tão utilizadas.
```

```
players <- players %>%
  select(-flash_assists, -kast, -adr, -c(m1_kills:m3_rating_t)) %>%
  subset(player_name!="")
```

```
### Removendo colunas que não importaram para nossa análise, decidimos analisar
### só certos rounds, que acontecem de certo em uma partida. No CS:GO uma partida
### tem até 30 rounds no "tempo regulamentar", podendo ter muito mais caso essa
### partida vá pro overtime. Mas de certo, 16 rounds ocorrem em qualquer partida
### então decidimos analisar apenas esses.
```

```
economy_mod <- economy %>%
  select(-c(X17_t1:X30_t1), -c(X17_t2:X30_t2), -c(X17_winner:X30_winner))
```

## Referências

- [1] Mercado de eSports: faturamento, audiência e o cenário no Brasil
- [2] Top 10 organizações mais valiosas no eSports - 13/04/2021