# Jogadores Tóxicos em Partidas de Jogos *Online*

*Joaquim Neto e Kazuki Yokoyama*

*07 de outubro de 2016*

## Introdução

Juntamente com a popularização das redes sociais na internet, veio a popularização dos jogos online. Nestes jogos, o comportamento abusivo de uma parcela dos jogadores é um problema constante. Dentro da comunidade destes jogos, esse tipo de comportamento é denominado tóxico, e pode destruir completamente o esforço de equipe entre os jogadores, arruinando a partida dos jogadores afetados, o que pode levar a uma gama de problemas, desde jogadores desistiram de consumir estes jogos, até a problemas psicológicos resultantes do bullying, nos casos mais sérios.

Consideramos um jogador tóxico, um jogador que incorre em comportamento abusivo durante o jogo. Neste trabalho, nós consideramos que jogadores tóxicos contaminam o ambiente ao seu redor, tornando ambiente de jogo pior para todo. Dizemos que ocorre uma contaminação tóxica, quando as ações de um jogador tóxico alteram negativamente o desempenho e/ou o comportamento de outros jogadores com quem ele interage.

## Objetivos

Pretendemos aqui primeiramente avaliar, através da análise dos dados das partidas e dos jogadores, qual é a extensão da contaminação tóxica aos outros jogadores. Para isso, analisaremos os dados com a finalidade de gerar um modelo para uma métrica de toxicidade. Depois tentaremos descobrir, entre os diferentes tipos de ofensas catalogadas pelo jogo, quais delas geram maior contaminação tóxica. Assim, podemos resumir os objetivos primários do trabalho como sendo

- Avaliar a extensão da contaminação tóxica a outros jogadores a partir dos dados da partida;
- Descobrir que tipos de ofensas geram mais contaminação.

### Objetivos da mineração

- Definir três categorias de jogadores distintas: tóxicos, contaminados e limpos;
- Definir uma métrica do desempenho dos jogadores;
- Mostrar que o jogador tóxico tem uma influência negativa no desempenho dos outros jogadores;
- Criar métrica para quantificar a contaminação dos jogadores considerados contaminados.

A fim de conseguirmos tais objetivos, precisamos primeiramente definir o que caracteriza um jogador tóxico, um jogador contaminado, e um jogador não contaminado (limpo). Após isso, teremos que definir uma métrica que quantifique o desempenho dos jogadores para podermos mostrar que o jogador tóxico exerce uma influência negativa no desempenho de outros jogadores. Finalmente, com essas informações, poderemos criar uma métrica que quantifique o grau da contaminação que o jogador tóxico exerce sobre os contaminados.

## Dados

Nosso trabalho tem como base os dados de partidas do jogo *online League of Legends* - LoL. O *dataset* é formado por denúncias de comportamento tóxico (abusivo) por jogadores do jogo. O *dataset* completo contém

1.46 milhões de casos de denúncias, que podem conter uma ou mais partidas com o jogador denunciado, e no total possui mais de 10 milhões de partidas, tornando-o grande demais para o escopo do nosso trabalho.

Devido a grande extensão dos dados, obtemos uma amostra, com aproximadamente 37 mil partidas. Todas as informações estavam originalmente no formato JSON. Para podermos trabalhar mais facilmente com os dados, escolhemos os atributos que consideramos relevantes ao nosso problema, e os dividimos em dois arquivos CSV: um contendo os dados inerentes à própria partida (`matches.csv`), outro contendo os dados dos jogadores de cada partida (`players.csv`), cujos atributos estão descritos abaixo:

### matches.csv

- *case* - caso dentro do dataset a partida pertence;
- *match* - numeração da partida dentro do caso;
- *premade* - se foi uma partida combinada anteriormente com os outros jogadores do time (1), ou se foi uma partida aonde o jogo combinou os jogadores (0);
- *most.common.offense* - qual foi o tipo de ofensa mais comum do jogador denunciado que foi reportada pelos outros jogadores da partida;
- *reports.allies* - quantidade de denúncias feitas pelo time aliado do jogador denunciado;
- *reports.enemies* - quantidade de denúncias feita pelo time inimigo ao jogador denunciado;
- *reports.case* - quantidade total de denuncias em todas as partidas do caso;
- *time.played* - tempo de jogo decorrido nessa partida.

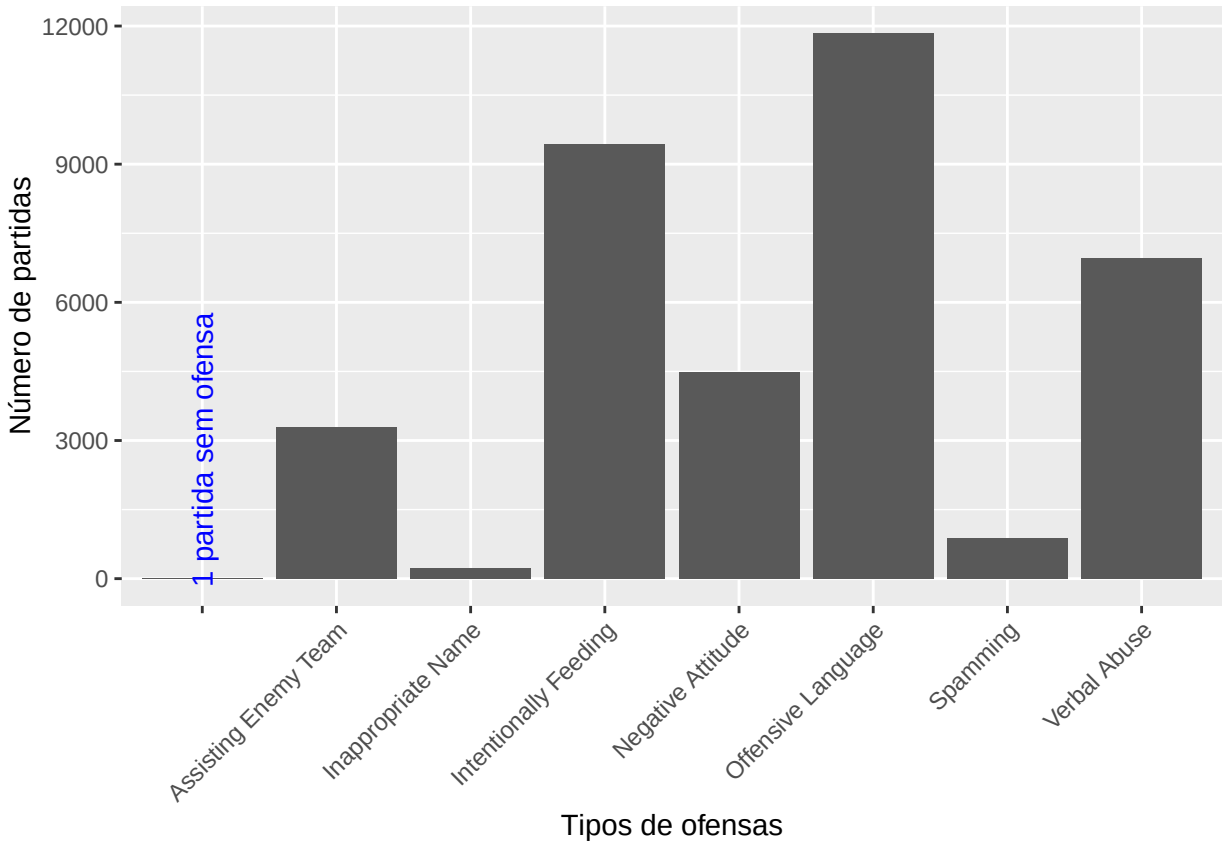
### players.csv

- *case* - caso dentro do dataset a partida pertence;
- *match* - numeração da partida dentro do caso;
- *relation.offender* - se este jogador está no mesmo time do denunciado (*ally*), em time diferente (*enemy*) ou se é o próprio denunciado (*offender*);
- *champion* - com qual personagem do jogo este jogador está jogando.Note que dois jogadores no mesmo time não podem ter o mesmo personagem;
- *kills* - quantidade de vezes que este jogador matou outros jogadores durante a partida;
- *deaths* - quantidade de vezes que este jogador morreu na partida;
- *assists* - quantidade de vezes que este jogador proveu algum tipo de assistência para o abate de um jogador inimigo, mas não realizou o abate ele mesmo;
- *gold* - quantidade de ouro acumulada durante a partida. Normalmente é usado para medir o desempenho de um jogador;
- *outcome* - Se este jogador ganhou a partida (*Win*), se ele a perdeu (*Loss*) ou se ele saiu do jogo antes dele acabar (*Leave*).

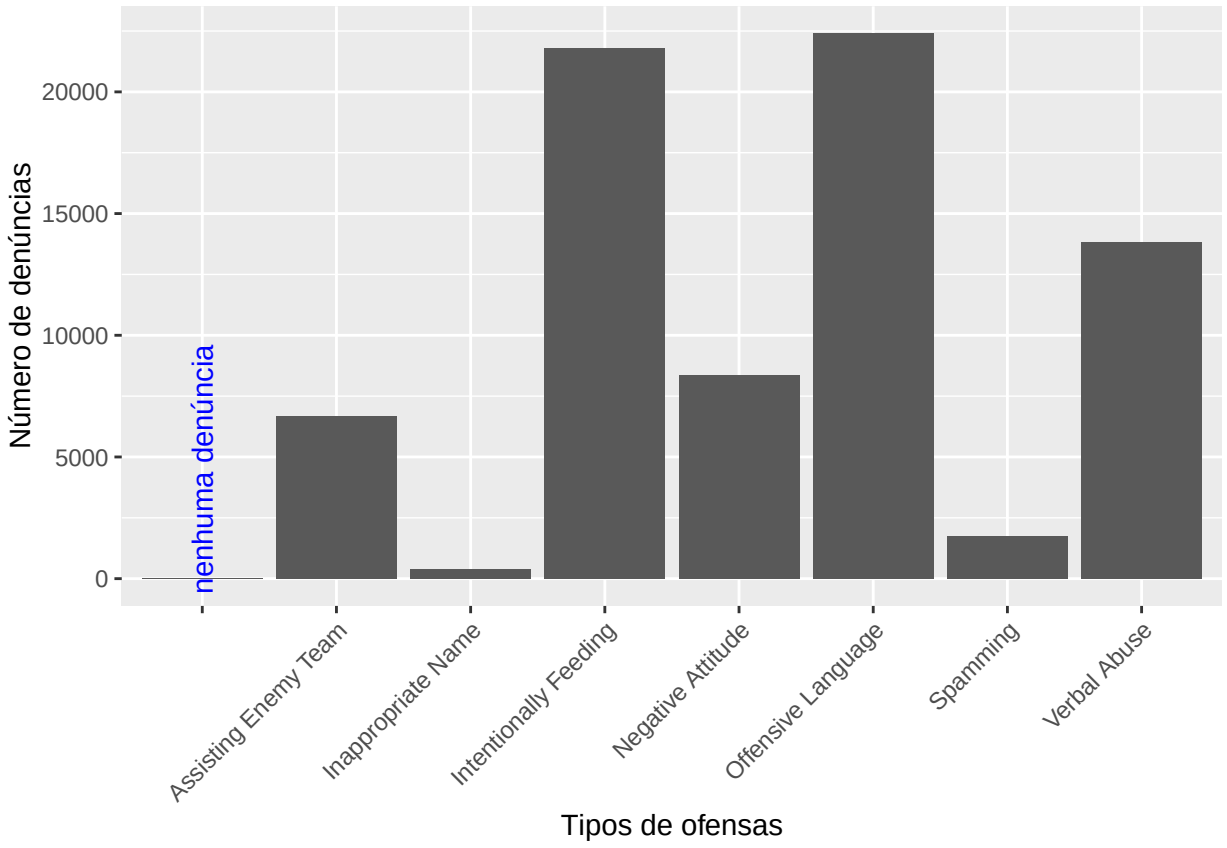
# EDA

```
setwd("..")

players <- read.csv("data/csv/players.csv", header = FALSE)
matchs <- read.csv("data/csv/matchs.csv", header = FALSE)

names(matchs) <- c("case", "match", "premade", "most.common.offense",
                   "reports.allies", "reports.enemies", "reports.case", "time.played")

names(players) <- c("case", "match", "relation.offender", "champion", "kills", "deaths",
```

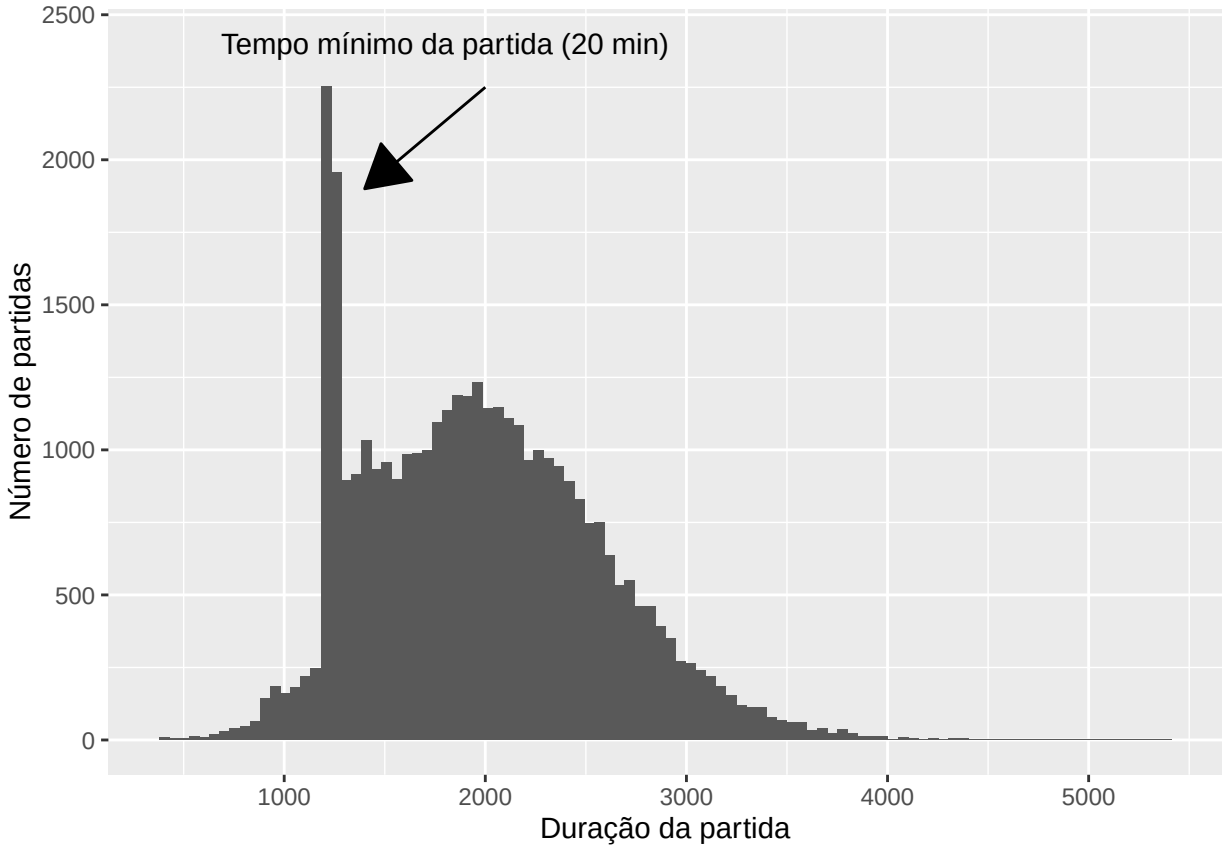
```
                    "assists", "gold", "outcome")

matchs.players <- matchs %>% left_join(players, by = c("case", "match"))
matchs.players <- matchs.players %>% mutate(kda = (kills + assists)/(deaths + 1))

allies <- matchs.players %>% filter(relation.offender == "ally")
enemies <- matchs.players %>% filter(relation.offender == "enemy")
offenders <- matchs.players %>% filter(relation.offender == "offender")

reason.by.team <- unique(matchs.players[, 1:8]) %>%
                  select(most.common.offense, reports.allies, reports.enemies) %>%
                  mutate(total.reports = reports.allies + reports.enemies)

reports.by.reason <- aggregate(total.reports ~ most.common.offense,
                               data = reason.by.team, FUN = sum)

case.match <- unique(matchs.players %>%
              select(case, match, relation.offender, outcome))

allies.win <- allies %>% filter(outcome == "Win")
enemies.win <- enemies %>% filter(outcome == "Win")
offenders.win <- offenders %>% filter(outcome == "Win")

common.offense.barplot <- ggplot(data = matchs, aes(x = most.common.offense)) +
                          geom_bar() +
                          xlab("Tipos de ofensas") + ylab("Número de partidas") +
                          annotate("text", x = 1, y = 2800, label = "1 partida sem ofensa",
                                   angle = 90, color = "blue") +
                          theme(axis.text.x=element_text(angle = 45, hjust = 1))

reports.by.reason.barplot <- ggplot(data = reports.by.reason) +
                             geom_bar(aes(x = most.common.offense, y = total.reports),
                                      stat = "identity") +
                             xlab("Tipos de ofensas") + ylab("Número de denúncias") +
                             annotate("text", x = 1, y = 4500, label = "nenhuma denúncia",
                                      angle = 90, color = "blue") +
                             theme(axis.text.x=element_text(angle = 45, hjust = 1))

time.played.hist <- ggplot(data = matchs, aes(x = time.played)) +
                    geom_histogram(bins = 100) +
                    xlab("Duração da partida") + ylab("Número de partidas") +
                    annotate("text", x = 1800, y = 2400,
                             label = "Tempo mínimo da partida (20 min)") +
                    geom_line(arrow = arrow(ends = "first", type = "closed"),
                              data = data.frame(x = c(2000, 1400), y = c(2250, 1900)),
                              aes(x = x, y = y))

outcome.relation.barplot <- ggplot(data = players) +
                            geom_bar(aes(x = relation.offender, fill = outcome)) +
                            xlab("Associação com ofensor") + ylab("Número de partidas")

print(common.offense.barplot)
```



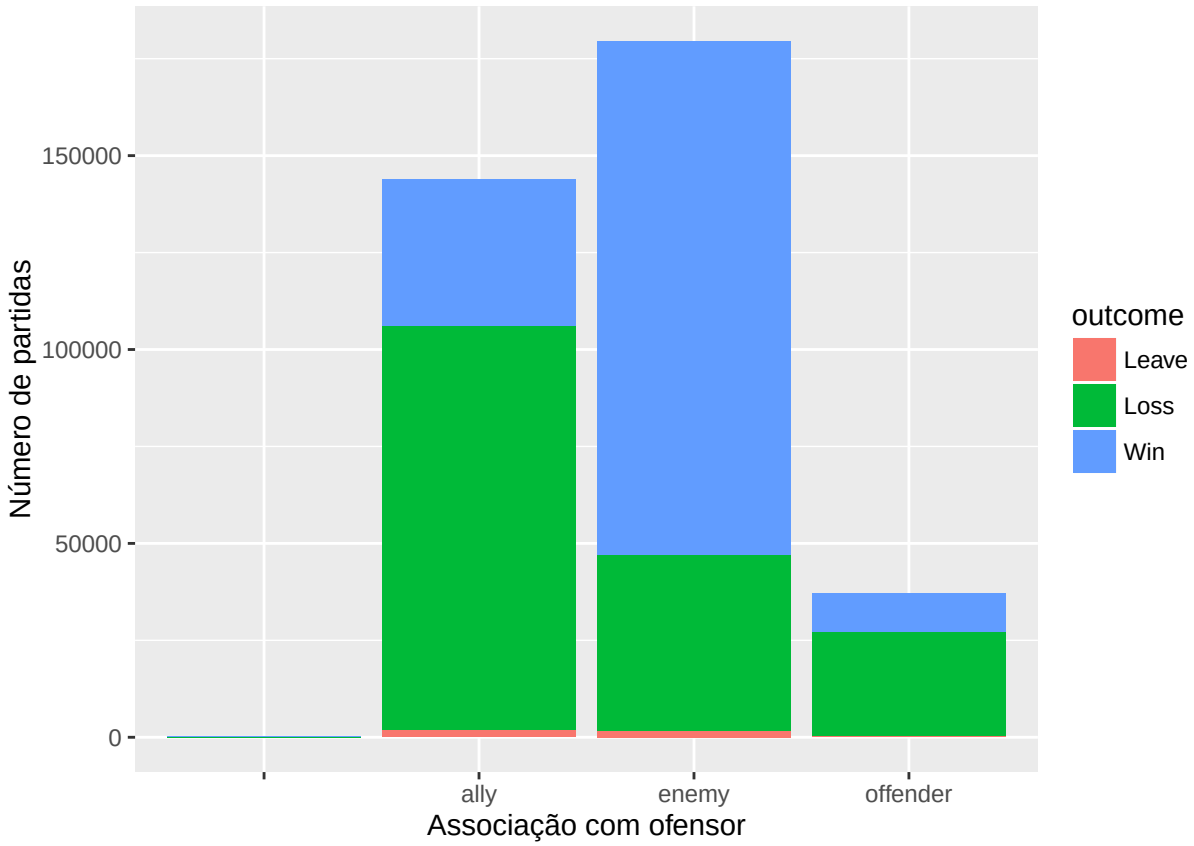

```
print(reports.by.reason.barplot)
```




```
print(time.played.hist)
```




```
print(outcome.relation.barplot)
```


150000
100000
50000
0
Número de partidas
ally
enemy
offender
Associação com ofensor
outcome
Leave
Loss
Win