

A2 - Análise Exploratória de Dados

João Vitor Sousa e Matheus de Moncada Assis

30 de Junho de 2020

Analisando estatísticas do jogo
Counter-Strike: Global Offensive

1 Introdução

Counter-Strike, mais conhecido como CS, é uma franquia desenvolvida pela Valve Corporation. No seu modo de jogo mais jogado atualmente, entrega ao consumidor um ambiente centrado no combate com armas de fogo, no qual se controla o personagem do jogo em uma perspectiva de primeira pessoa.

Com o passar dos anos, o cenário competitivo de CS cresceu e se tornou o que é hoje, um dos maiores jogos de e-sport do planeta, conquistando mais e mais interessados no assunto para assistir, além de movimentar mais e mais milhões a cada ano.

Pensando nisso, estaremos analisando o comportamento de algumas franquias e jogadores desse cenário com o passar dos anos em relação a algumas características do jogo: performance dos times, escolhas de mapas por time e seu rendimento neles, performance dos jogadores.

Utilizando os datasets encontrados nesse Kaggle:

<https://www.kaggle.com/mateusdmachado/csgo-professional-matches>

Com ele, fomos capazes de gerar as análises desejadas, juntamente com gráficos esclarecedores, fortalecendo alguns argumentos que tínhamos em mente ao pensarmos no problema.

Finalmente, a nossa apresentação pode ser encontrada nesse link:

[https://drive.google.com/drive/folders/
1iBLss1hT1tQ4vSW5NAO9akl3qxzEheEB?usp=sharing](https://drive.google.com/drive/folders/1iBLss1hT1tQ4vSW5NAO9akl3qxzEheEB?usp=sharing)

Basta fazer o download do arquivo e assistir nossa apresentação.

2 Times

2.1 Análise dos Times

Dentro do cenário de Counter-Strike, podemos reconhecer diversas franquias com nomes consagrados, tais como: FaZe, Astralis, G2, etc. Além disso, também temos algumas organizações brasileiras que se destacam nesse meio, sendo duas delas importantes no cenário: FURIA e MIBR. Tais times, como em outros esportes, tem seus próprios jogadores contratados e conquistam legiões de fãs baseados nisso, fazendo com que a competitividade do cenário floresça.

Conforme os anos vão se passando, com o acúmulo de dados, podemos ver como esses times se comportam e se há alguma ascensão (ou queda) notável em um intervalo de tempo. Também temos que considerar as mudanças de organizações por parte dos players, fazendo com que um time inteiro possa se transferir para outra organização e carregar a alma do time junto. É importante, portanto, considerar esses dois fatores para gerarmos nossas primeiras análises.

Escolhendo uma parcela de times, vamos analisar ano a ano, desde 2017, como foi sua taxa de vitórias em partidas oficiais do cenário. Consideramos as partidas melhores de 1 e melhores de 3, sendo dois dos modos mais comuns de enfrentamento nesses torneios.

2.1.1 2017

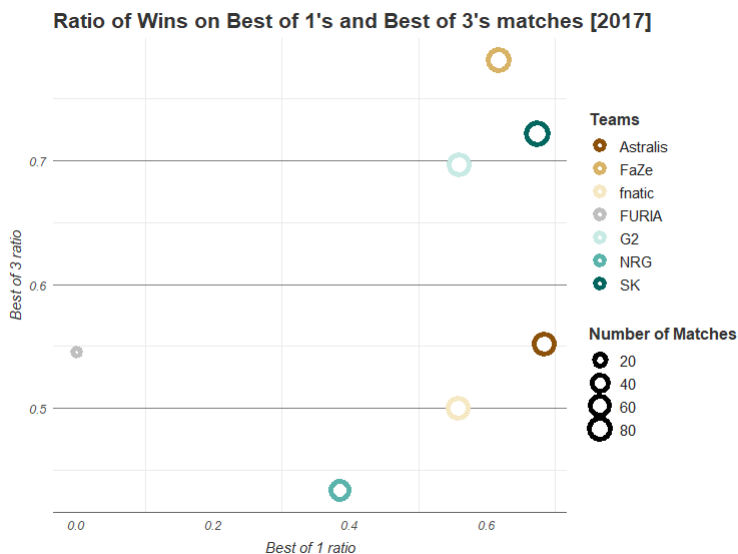


Figure 1: Match Performance in 2017

Escolhendo os times da Astralis (Dinamarca), FaZe (Europa), fnatic (Suécia), FURIA (Brasil), G2 (França), NRG (EUA) e SK (Brasil), podemos perceber a dominância da SK e da FaZe, apresentando mais de 70% de vitórias em Melhores de 1 e quase 70% em Melhores de 3. Ademais, o número de partidas jogadas por ambas as equipes é próximo de 80, o que torna essa estatística bem relevante.

Analisando o outro time brasileiro, a FURIA, podemos perceber que ele não obteve um resultado muito bom em partidas Melhores de 1, mesmo que tenha jogado por volta de 10 partidas no total. Outro time que apresenta um resultado ruim é o da NRG, que se aproximava de 40% em ambos os tipos de confronto.

2.1.2 2018

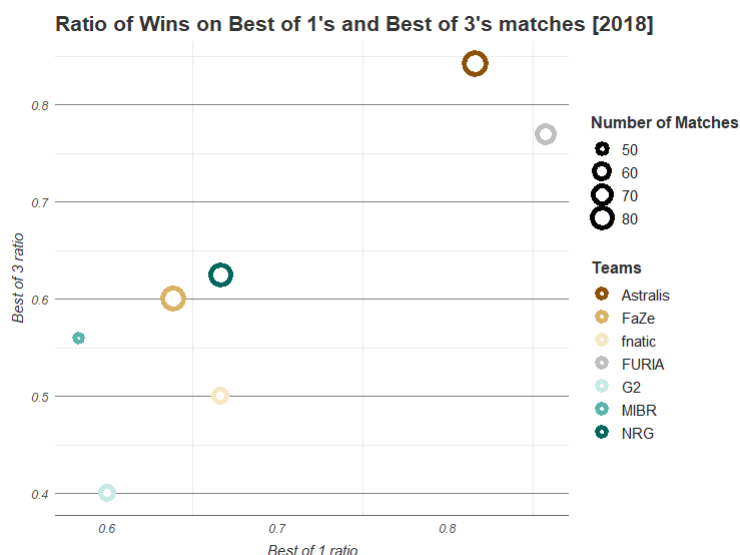


Figure 2: Match Performance in 2018

Em 2018, já podemos notar uma mudança em um dos times brasileiros. Com alguns problemas internos, os jogadores da antiga SK decidiram mudar para outra organização, a MIBR, e portanto, estaremos analisando seus resultados como MIBR a partir de 2018.

Ademais, podemos ver como a Astralis e a FURIA começaram a apresentar resultados fenomenais, se aproximando de 90% em partidas Melhores de 3 e Melhores de 1, respectivamente. Com um número de partidas alto, ambas as equipes conseguiram resultados fenomenais de um ano para o outro, conseguindo deixar as outras melhores equipes para trás.

Finalmente, vale ressaltar também a queda de rendimento da G2, que não conseguiu manter o nível do último ano, e a ascensão da NRG, que conseguiu melhorar consideravelmente suas estatísticas em pouco tempo.

2.1.3 2019

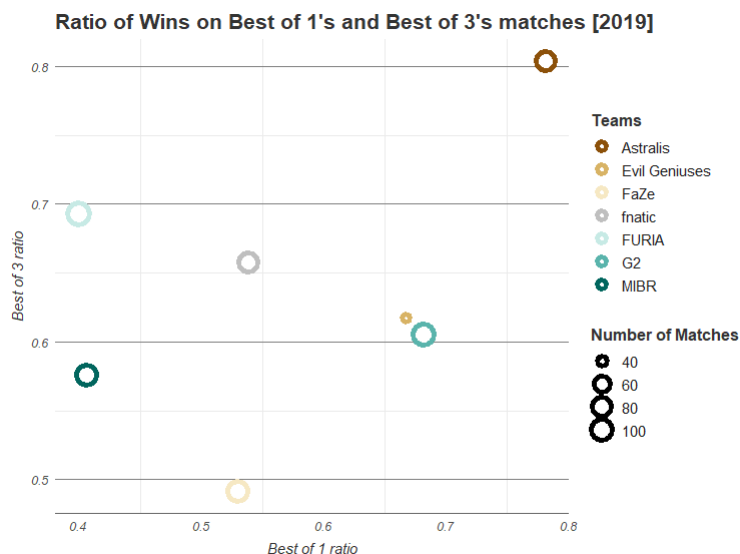


Figure 3: Match Performance in 2019

Em 2019, a última mudança das franquias que será mostrada nesse gráfico aconteceu. A antiga NRG acabou e foi contratada pela Evil Geniuses. Esse acaba se tornando um ponto muito importante para a análise de dados, tendo em vista a possibilidade de desconsiderar períodos de atuação das equipes dada uma mudança no meio do ano.

Quanto ao gráfico de performance, podemos notar como a Astralis foi dominante sobre qualquer outro time nesse tempo. Não à toa, nesse ano eles se mantiveram no TOP 1 do Ranking Mundial durante 9 meses, vencendo praticamente todos os torneios disputados. Além disso, com uma performance tão boa, podemos ver que a maioria dos outros times não conseguiu se manter no nível deles, tendo uma queda de rendimento considerável.

Finalmente, podemos ver uma queda ano a ano da performance deles, enquanto a FURIA mantém um bom rendimento em partidas de Melhor de 3. Além disso, também podemos ver que o time com rendimento mais fraco nesse ano foi a FaZe, apresentando 50% de vitórias em ambos os tipos de partidas.

2.1.4 Ao longo de 4.5 anos

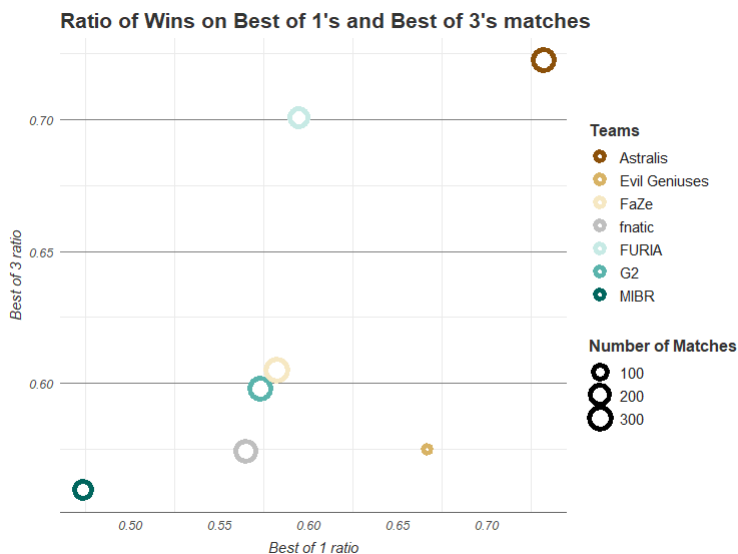


Figure 4: Match Performance All Time

Analizando o último gráfico, podemos perceber a dominância da Astralis, principalmente pelos resultados conquistados ao longo dos últimos anos. Dessa maneira, conseguindo provar sua competência a nível mundial, conseguiram deixar vários outros times que oscilaram durante os anos para trás.

Além disso, vale ressaltar também a performance da FURIA, conseguindo representar bem o Brasil nos últimos anos. A MIBR, no entanto, teve uma queda de performance ao longo dos últimos anos e não conseguiu se encontrar, fazendo mudanças de elenco durante diversas partes da temporada e perdendo torneios regularmente. É importante lembrar que o elenco da MIBR, na época da SK, já ganhou o Mundial de CS:GO, sendo considerada a melhor equipe do mundo durante alguns meses.

Finalmente, podemos ver a regularidade das outras equipes, alcançando resultados próximos de 60% em ambos os modos de jogo ao longo dos 4.5 anos.

2.2 Código em R para o plot

```
library(ggplot2)
library(operators)
library(hash)
library(roperators)

players <- read.csv("C:/Users/Matheus/Desktop/Trabalho AED/players.csv")
picks <- read.csv("C:/Users/Matheus/Desktop/Trabalho AED/picks.csv")
economy <- read.csv("C:/Users/Matheus/Desktop/Trabalho AED/economy.csv")
results <- read.csv("C:/Users/Matheus/Desktop/Trabalho AED/results.csv")

#####
# Creating Dataframe
#####

# creating specific dataframe to get wins
df_wins <- results[c('date', 'match_id', 'event_id', 'team_1', 'team_2',
                    'match_winner')]
new_df_wins <- unique(df_wins, by="match_id")
df_wins_best_of <- merge(new_df_wins, picks[c('match_id', 'best_of')],
                        by='match_id')
df_wins_best_of <- df_wins_best_of[order(as.Date(df_wins_best_of$date,
                                                format="%Y-%m-%d")),]

# getting specific year
YEAR = "2017"
initial_i = 0
max_i = length(df_wins_best_of$match_id)
for (i in c(1:(length(df_wins_best_of$match_id)))) {
  date = toString(df_wins_best_of$date[i])
  if (substring(date, 1, 4) == YEAR) {
    if (initial_i == 0) {
      initial_i = i
    }
    max_i = i
  }
}
df_wins_best_of <- df_wins_best_of[c(initial_i:max_i),]

# adding winner and loser names columns
df_wins_best_of$winner_name = "teste"
df_wins_best_of$loser_name = "teste"
for (i in c(1:(length(df_wins_best_of$match_id)))) {
  if (df_wins_best_of$match_winner[i] == 1) {
    df_wins_best_of$winner_name[i] = toString(df_wins_best_of$team_1[i])
    df_wins_best_of$loser_name[i] = toString(df_wins_best_of$team_2[i])
  }
}
```

```

    } else if (df_wins_best_of$match_winner[i] == 2) {
      df_wins_best_of$winner_name[i] = toString(df_wins_best_of$team_2[i])
      df_wins_best_of$loser_name[i] = toString(df_wins_best_of$team_1[i])
    }
  }

# getting all teams
team_list = list()
for (i in c(1:length(df_wins_best_of$match_id))) {
  if (df_wins_best_of$team_1[i] %!in% team_list)
    team_list <- c(team_list, toString(df_wins_best_of$team_1[i]))
  if (df_wins_best_of$team_2[i] %!in% team_list)
    team_list <- c(team_list, toString(df_wins_best_of$team_2[i]))
}

# creating a hash table to keep track of wins and total matches for each
  team
team_dic = hash()
for (team in team_list) {
  team_dic[team] <- hash(c(
    "B01_WINS"=0,
    "B01_PLAYED"=0,
    "B03_WINS"=0,
    "B03_PLAYED"=0,
    "B05_WINS"=0,
    "B05_PLAYED"=0
  ))
}

# populating our hash table
for (i in c(1:length(df_wins_best_of$match_id))) {
  best_of = toString(df_wins_best_of$best_of[i])
  winner = df_wins_best_of$winner_name[i]
  loser = df_wins_best_of$loser_name[i]

  if (best_of == '1') {
    team_dic[[winner]][["B01_WINS"]] %+=% 1
    team_dic[[winner]][["B01_PLAYED"]] %+=% 1
    team_dic[[loser]][["B01_PLAYED"]] %+=% 1
  }
  else if (best_of == '3') {
    team_dic[[winner]][["B03_WINS"]] %+=% 1
    team_dic[[winner]][["B03_PLAYED"]] %+=% 1
    team_dic[[loser]][["B03_PLAYED"]] %+=% 1
  }
  else if (best_of == '5') {
    team_dic[[winner]][["B05_WINS"]] %+=% 1

```



```

        team_dic[[winner]][["B05_PLAYED"]] %+=% 1
        team_dic[[loser]][["B05_PLAYED"]] %+=% 1
    }
}

# getting ratio values
astralis_ratio_bo1 =
    team_dic[["Astralis"]][["B01_WINS"]]/team_dic[["Astralis"]][["B01_PLAYED"]]
astralis_ratio_bo3 =
    team_dic[["Astralis"]][["B03_WINS"]]/team_dic[["Astralis"]][["B03_PLAYED"]]
astralis_matches =
    team_dic[["Astralis"]][["B01_PLAYED"]]+team_dic[["Astralis"]][["B03_PLAYED"]]

if (YEAR %in% c("2016", "2017")) {
    mibr_name = "SK"
} else if (YEAR %in% c("2018", "2019", "2020")) {
    mibr_name = "MIBR"
}

mibr_ratio_bo1 =
    team_dic[[mibr_name]][["B01_WINS"]]/team_dic[[mibr_name]][["B01_PLAYED"]]
mibr_ratio_bo3 =
    team_dic[[mibr_name]][["B03_WINS"]]/team_dic[[mibr_name]][["B03_PLAYED"]]
mibr_matches =
    team_dic[[mibr_name]][["B01_PLAYED"]]+team_dic[[mibr_name]][["B03_PLAYED"]]

furia_ratio_bo1 =
    team_dic[["FURIA"]][["B01_WINS"]]/team_dic[["FURIA"]][["B01_PLAYED"]]
furia_ratio_bo3 =
    team_dic[["FURIA"]][["B03_WINS"]]/team_dic[["FURIA"]][["B03_PLAYED"]]
furia_matches =
    team_dic[["FURIA"]][["B01_PLAYED"]]+team_dic[["FURIA"]][["B03_PLAYED"]]

g2_ratio_bo1 =
    team_dic[["G2"]][["B01_WINS"]]/team_dic[["G2"]][["B01_PLAYED"]]
g2_ratio_bo3 =
    team_dic[["G2"]][["B03_WINS"]]/team_dic[["G2"]][["B03_PLAYED"]]
g2_matches =
    team_dic[["G2"]][["B01_PLAYED"]]+team_dic[["G2"]][["B03_PLAYED"]]

faze_ratio_bo1 =
    team_dic[["FaZe"]][["B01_WINS"]]/team_dic[["FaZe"]][["B01_PLAYED"]]
faze_ratio_bo3 =
    team_dic[["FaZe"]][["B03_WINS"]]/team_dic[["FaZe"]][["B03_PLAYED"]]
faze_matches =
    team_dic[["FaZe"]][["B01_PLAYED"]]+team_dic[["FaZe"]][["B03_PLAYED"]]

fnatic_ratio_bo1 =
    team_dic[["fnatic"]][["B01_WINS"]]/team_dic[["fnatic"]][["B01_PLAYED"]]

```

```

fnatic_ratio_bo3 =
  team_dic[["fnatic"]][["B03_WINS"]]/team_dic[["fnatic"]][["B03_PLAYED"]]
fnatic_matches =
  team_dic[["fnatic"]][["B01_PLAYED"]]+team_dic[["fnatic"]][["B03_PLAYED"]]

if (YEAR %in% c("2016", "2017", "2018")) {
  evil_geniuses_name = "NRG"
} else if (YEAR %in% c("2019", "2020")) {
  evil_geniuses_name = "Evil Geniuses"
}

evilgeniuses_ratio_bo1 =
  team_dic[evil_geniuses_name][["B01_WINS"]]/team_dic[evil_geniuses_name][["B01_PLAYED"]]
evilgeniuses_ratio_bo3 =
  team_dic[evil_geniuses_name][["B03_WINS"]]/team_dic[evil_geniuses_name][["B03_PLAYED"]]
evilgeniuses_matches =
  team_dic[evil_geniuses_name][["B01_PLAYED"]]+team_dic[evil_geniuses_name][["B03_PLAYED"]]

final_df = data.frame(
  Teams <- c("Astralis", mibr_name, "FURIA", "G2", "FaZe", "fnatic",
    evil_geniuses_name),
  B01 <- c(astralis_ratio_bo1, mibr_ratio_bo1, furia_ratio_bo1,
    g2_ratio_bo1, faze_ratio_bo1, fnatic_ratio_bo1,
    evilgeniuses_ratio_bo1),
  B03 <- c(astralis_ratio_bo3, mibr_ratio_bo3, furia_ratio_bo3,
    g2_ratio_bo3, faze_ratio_bo3, fnatic_ratio_bo3,
    evilgeniuses_ratio_bo3),
  Num_matches <- c(astralis_matches, mibr_matches, furia_matches,
    g2_matches, faze_matches, fnatic_matches, evilgeniuses_matches)
)
colnames(final_df) <- c("Teams", "B01", "B03", "Matches")

write.csv(final_df, "C:/Users/Matheus/Desktop/Trabalho
  AED/best_of_dataframe.csv")

#####
# Plotting
#####

library(ggplot2)

ideal_df <- read.csv("C:/Users/Matheus/Desktop/Trabalho
  AED/best_of_dataframe.csv", row.names = "X")

pc1 <- ggplot(ideal_df, aes(x = B01, y = B03)) +
  geom_point(
    shape = 21,
    stroke = 3,

```

```

    fill = "white",
    aes(color = factor(Teams), size=Num_matches)
  )
pc1

pc2 <- pc1 +
  ggtitle(
    "Ratio of Wins on Best of 1's and Best of 3's matches [2017]"
  ) +
  labs(
    color = "Teams",
    size = "Number of Matches",
    x = "Best of 1 ratio",
    y = "Best of 3 ratio"
  ) +
  scale_color_manual(name = "Teams",
    values = c("#8c510a",
               "#d8b365",
               "#f6e8c3",
               "#BFBFBF",
               "#c7eae5",
               "#5ab4ac",
               "#01665e"))

pc2

pc3 <- pc2 +
  theme_minimal() +
  theme(
    text = element_text(color = "gray20"),
    legend.title = element_text(size=12, face="bold"),
    legend.position = "right",
    legend.direction = "vertical",
    legend.justification = 0.4,
    legend.text = element_text(size = 11, color = "gray10"),
    axis.text = element_text(face = "italic"),
    axis.title.x = element_text(vjust = -1, face="italic"),
    axis.title.y = element_text(vjust = 2, face="italic"),
    axis.ticks.y = element_blank(),
    axis.line = element_line(color = "gray40", size = 0.5),
    axis.line.y = element_blank(),
    panel.grid.major = element_line(color = "gray50", size = 0.5),
    panel.grid.major.x = element_blank(),
    plot.caption = element_text(hjust = 0),
    plot.title = element_text(size = 16, face="bold")
  )

pc3

```

3 Escolha de Mapas

Se observa no cenário competitivo de Counter-Strike vários campeonatos que adotam confrontos em série, tanto em melhores de 3 quanto em melhores de 5. Parte fundamental do preparo dos times para essas séries é escolher quais dos 7 mapas eles estão dispostos a jogar, quais eles querem vetar, quais são suas preferências. Visto que em séries melhores de 3 cada time, além de vetar 2 mapas, pode escolher um mapa, é interessante analisar o quão efetivo são as escolhas feitas pelos times.

3.1 Análise das Escolhas de Mapas

3.1.1 2017

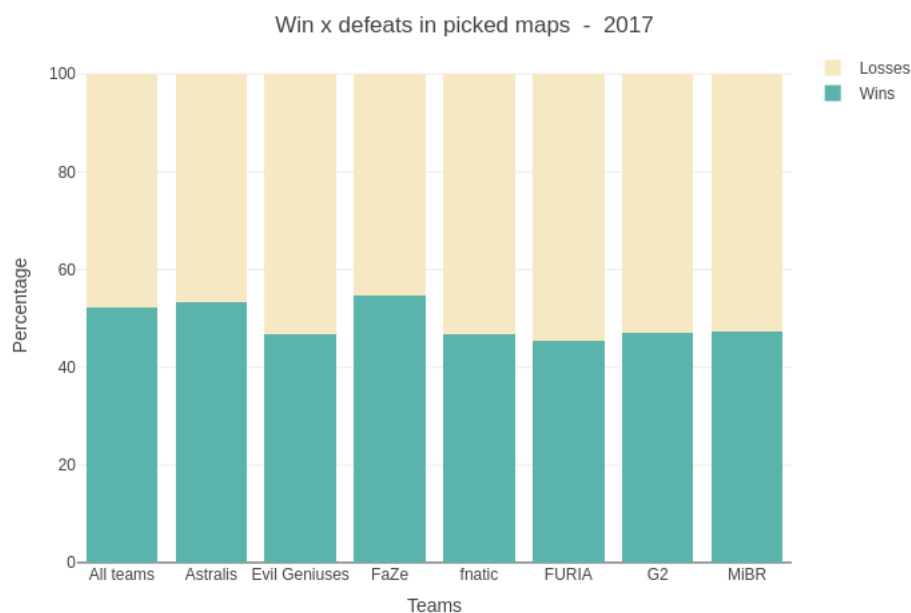


Figure 5: Win rate on picked maps

No ano de 2017 a equipe com a maior taxa de vitórias em mapas escolhidos foi a FaZe, isso reflete a boa fase vivida por eles com também o maior índice de vitórias em séries. Um dado conflitante é a taxa de vitória do time da MIBR/SK. Apesar de ter conseguido o título no Major de Cologne e ter a segunda maior taxa de vitória em séries, o time teve baixo aproveitamento em mapas que por eles foram escolhidos, cerca de 47%.

3.1.2 2018

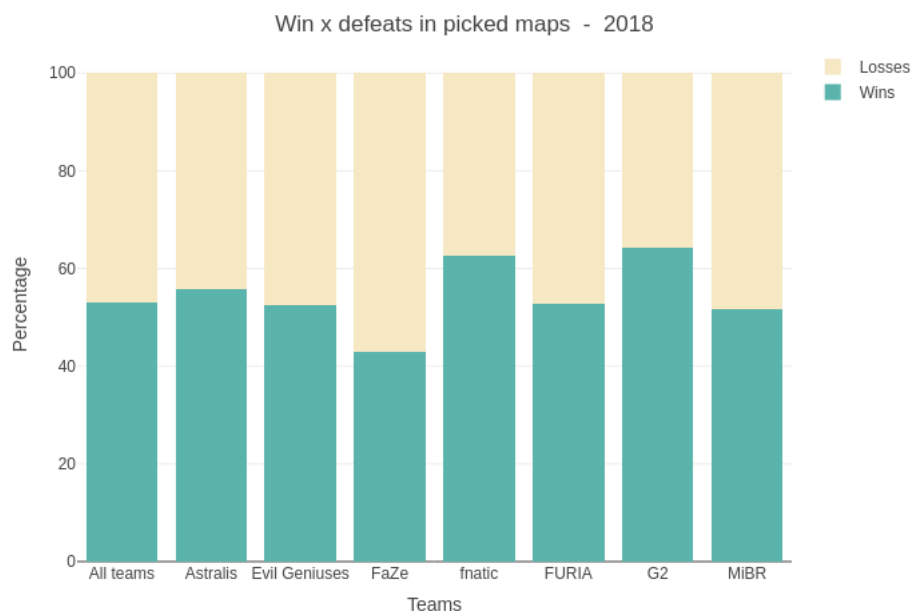


Figure 6: Win rate on picked maps

Em 2018 começa a "Era Astralis", dita por especialistas e amantes do jogo. A partir daí o time dinamarquês conquistaria 4 títulos mundiais consecutivos. Apesar do expressivo número de vitórias em séries, se aproximando dos 85%, seu aproveitamento em mapas escolhidos chegava apenas a 56%, bem próximo a média geral, 53%. Temos duas surpresas:

- O time da G2 que apesar de uma taxa de vitória próxima a 40% em séries, conseguia sair vencedora em 64% dos mapas escolhidos por eles.
- O time da Furia, assim como a Astralis mantinha uma alta taxa de vitórias em séries apesar do relativo baixo aproveitamento em mapas de sua escolha.

3.1.3 2019

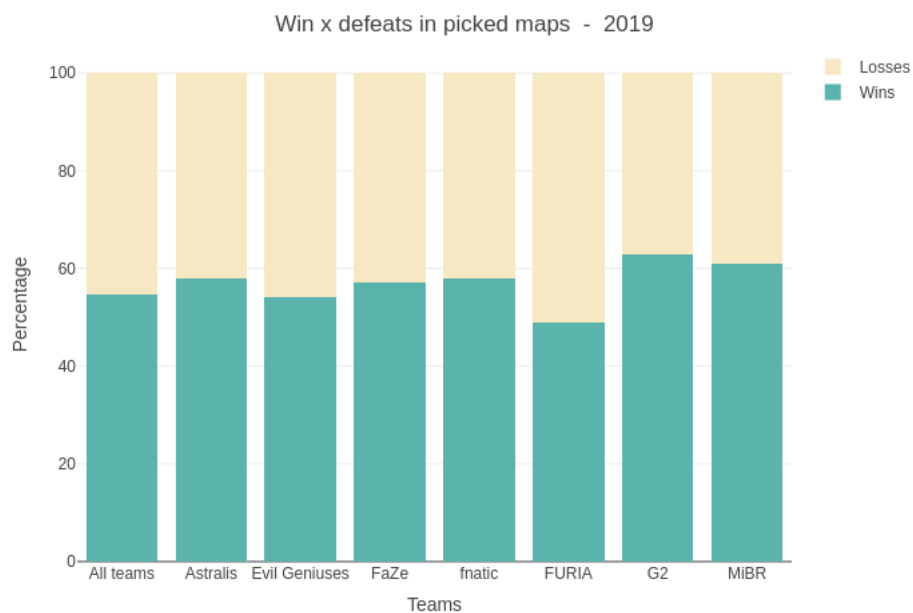


Figure 7: Win rate on picked maps

Em 2019, segundo ano da "Era Astralis", observamos uma leve queda no aproveitamento do time tanto em séries quanto em mapas escolhidos. Temos como destaque em 2019 a G2, que se reergueu da péssima fase que vivera no ano anterior e manteve 63% de taxa de vitória em mapas de sua escolha. Esse crescimento da equipe nos leva diretamente ao cenário atual, onde a equipe da G2 ocupa o primeiro lugar no ranking da HLTV. A MIBR, time brasileiro, também teve nesse ano uma taxa altíssima de vitórias em mapa de sua escolha, apesar da má fase em séries.

3.2 Código em R para o plot

```
library(readr)
library(dplyr)
library(plotly)

picks <- read_csv("datasets/picks.csv")
results <- read_csv("datasets/results.csv")

#merging tables and filtering for only bof3 matches
bof3 <- picks[picks$best_of == 3, ]
merged <- merge(bof3, results, by = "match_id")

bof3_picks <- merged %>% select(match_id, inverted_teams, t1 =
  team_1.y, t2 = team_2.y, event_id = event_id.x, t1pick =
  t1_picked_1, t2pick = t2_picked_1, pick_help = t1_picked_1,
  left_over, date = date.x, map = '_map', map_winner, rank_1,
  rank_2)

#handling inverted picks
bof3_picks <- bof3_picks[bof3_picks$map != bof3_picks$left_over,]

picks <- transmute(bof3_picks, t1pick = if_else(inverted_teams
  == 1, t2pick, t1pick), t2pick = if_else(inverted_teams == 1,
  pick_help, t2pick))

bof3_picks$t1pick <- picks[1]
bof3_picks$t2pick <- picks[2]

colnames(bof3_picks)[6] <- "t1pick"
colnames(bof3_picks)[7] <- "t2pick"

bof3_picks <- bof3_picks %>% select(match_id, t1, t2, event_id,
  t1pick, t2pick, date, map, map_winner, rank_1, rank_2)

#Help column for plot
bof3_picks$picker_won <- NA
has_picker_won <- function(x){
  has_won <- 0
  t1pick <- x[5]
  t2pick <- x[6]
  map <- x[8]
  map_winner <- x[9]
  if(map == t1pick && map_winner == 1){
    has_won <- 1
  } else {
    if(map == t2pick && map_winner == 2){
      has_won <- 1
    }
  }
}
```

```

    }
    has_won
  }
  picker_won <- apply(bof3_picks, 1, has_picker_won)
  bof3_picks$picker_won <- picker_won

#Generates the plots
make_plot <- function(data, year) {
  if(year != -1){
    data <- data[substring(data$date, 1, 4) == year,]
  }

  MIBR <- "MIBR"

  if(year < 2019){
    MIBR <- "SK"
  }

  if(year == -1){
    year <- " - All time"
  }else{
    year <- paste(" - ", year)
  }

  picks_astralis <- data[data$t1 == "Astralis" | data$t2 ==
    "Astralis",]
  picks_mibr <- data[data$t1 == MIBR | data$t2 == MIBR,]
  picks_furia <- data[data$t1 == "FURIA" | data$t2 == "FURIA",]
  picks_g2 <- data[data$t1 == "G2" | data$t2 == "G2",]
  picks_evilg <- data[data$t1 == "NRG" | data$t2 == "NGR",]
  picks_faze <- data[data$t1 == "FaZe" | data$t2 == "FaZe",]
  picks_fnatic <- data[data$t1 == "fnatic" | data$t2 == "fnatic",]

  percentage_table <- function(x){
    percentage <- c(100*x[1]/sum(x), 100*x[2]/sum(x))
    percentage
  }

  table_astralis <-
    percentage_table(table(picks_astralis$picker_won))
  table_mibr <- percentage_table(table(picks_mibr$picker_won))
  table_furia <- percentage_table(table(picks_furia$picker_won))
  table_g2 <- percentage_table(table(picks_g2$picker_won))
  table_evilg <- percentage_table(table(picks_evilg$picker_won))
  table_faze <- percentage_table(table(picks_faze$picker_won))
  table_fnatic <- percentage_table(table(picks_fnatic$picker_won))
  table_general <- percentage_table(table(data$picker_won))

  teams <- c("Astralis", "MiBR", "FURIA", "G2", "Evil Geniuses",
    "FaZe", "fnatic", "All teams")

```



```

lost <- c(table_astralis[1], table_mibr[1], table_furia[1],
          table_g2[1], table_evilg[1], table_faze[1],
          table_fnatic[1], table_general[1])

won <- c(table_astralis[2], table_mibr[2], table_furia[2],
         table_g2[2], table_evilg[2], table_faze[2],
         table_fnatic[2], table_general[2])

df <- data.frame(teams, lost, won)

fig <- plot_ly(x = df$teams, y = df$won, type = "bar", name =
               "Wins", marker = list(color = "#5ab4ac"))
fig <- fig %>% add_trace(y = ~lost, name = "Losses", marker =
               list(color = "#f6e8c3"))
#fig <- fig %>% add_trace(x=~teams,y=table_general[2],
               type='scatter',mode='lines',
               line=list(color='black'),showlegend=F)
fig <- fig %>% layout(yaxis = list(title = 'Percentage'), xaxis
                    = list(title = "Teams"), barmode = 'stack', categoryorder
                    = TRUE, title = paste("Win x defeats in picked maps",
                    year))
fig
}

make_plot(bof3_picks, "2017")
make_plot(bof3_picks, "2018")
make_plot(bof3_picks, "2019")

```

4 Jogadores

4.1 Análise dos Jogadores

Em se tratando dos jogadores, podemos observar que diversos deles começaram a jogar desde os primórdios do falecido Counter-Strike 1.6, jogo antecedente ao Counter-Strike: Global Offensive. Ademais, muitos deles se arriscaram numa carreira incerta e com um cenário inexistente, fazendo parte da criação e do estabelecimento do CS como nome principal do meio competitivo de jogos.

É importante ressaltar a longevidade de muitos desses nomes no cenário e, mesmo que mais velhos comparados a média do cenário, ainda apresentam resultados formidáveis. Contudo, também podemos perceber como novos nomes no cenário vão conquistando o meio e elevando o nível do jogo.

Finalmente, também podemos analisar o desempenho dos brasileiros boltz, coldzera, fer e FalleN, que são referências para todo o cenário e que conseguem manter o nível de jogo alto há mais de 5 anos seguidos.

Vale dizer que a variável rating é utilizada para definir os melhores jogadores do cenário atualmente, já que ela é calculada a partir de diversas outras variáveis em relação a cada player.

4.1.1 Top 32

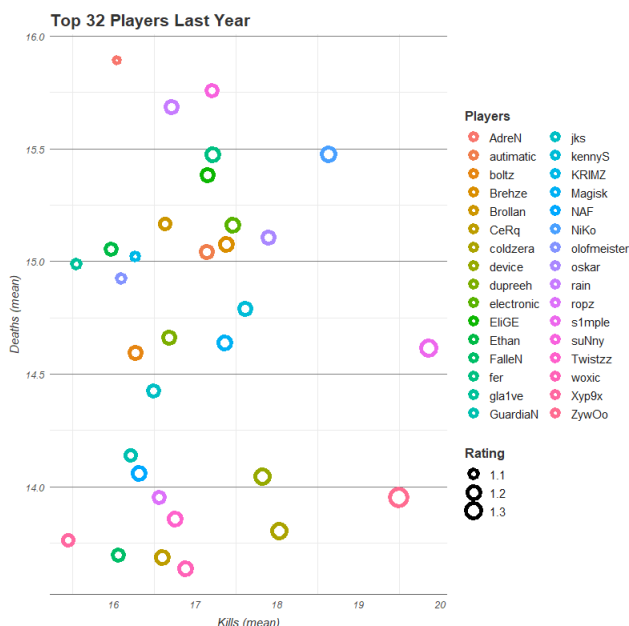


Figure 8: Top 32 Players

Analisando o gráfico gerado, podemos perceber a predominância de um rating alto em jogadores no canto inferior direito (mais kills e menos mortes). Com isso, podemos concluir que realmente há, em cada time, um melhor jogador que ficará responsável por matar mais jogadores do time inimigo, sendo sempre o segundo a aparecer junto de seu companheiro para realizar a dita "troca de kills", em que o time inimigo acerta seu companheiro que entrou na frente, e ele finaliza os jogadores que estão concentrados em seu colega.

Ademais, é importante ressaltar que mais mortes não necessariamente significam rating menor, tanto que vemos diversos jogadores na parte de cima da tabela mantendo ratings bons, não desmerecendo assim a posição de "entry fragger", o cara que entra primeiro em algum local junto com quem vai finalizar os inimigos.

Portanto, vale verificar a presença de poucos jogadores com uma média muito grande de kills, e no caso podemos ressaltar o Oleksandr "S1mple" Kostyliw e o Mathieu "ZywOo" Herbaut, considerados atualmente dois dos melhores jogadores do cenário.

4.2 Código em R para o plot

```
library(ggplot2)
library(deploy)

#####
# Creating Dataframe
#####

players <- read.csv("C:/Users/Matheus/Desktop/Trabalho AED/players.csv")

sum_maps <- function(player, df){
  qtt_map1 <- sum(table(df[df$player_name == player,]$map_1))
  qtt_map2 <- sum(table(df[df$player_name == player & df$best_of ==
    3,]$map_2))
  qtt_map3 <- sum(table(df[df$player_name == player & df$best_of ==
    3,]$map_3))
  qtt_map1 + qtt_map2 + qtt_map3
}

summarise_player <- function(player, df){
  maps_played <- sum_maps(player, df)
  kills <- sum(df[df$player_name == player,]$kills)
  deaths <- sum(df[df$player_name == player,]$deaths)
  hs <- sum(df[df$player_name == player,]$hs)
  mean_rating <- mean(df[df$player_name == player &
    !is.na(df$rating),]$rating)
  mean_kast <- mean(df[df$player_name == player & !is.na(df$kast),]$kast)
  mean_kills <- kills / maps_played
  mean_deaths <- deaths / maps_played
}
```

```

    mean_hs <- hs / maps_played
    return(c(maps_played, kills, deaths, hs, mean_rating, mean_kast,
             mean_kills, mean_deaths, mean_hs))
}

# defining players to our analysis
selected_players <- c('EliGE', 'NAF', 'Magisk', 'Xyp9x', 'electronic',
                     'autimatic', 'dupreeh', 'suNny', 'rain', 'fer', 'Twistzz',
                     'coldzera', 'Ethan', 'Brehze', 'kennyS', 'CeRq', 'GuardiaN',
                     'NiKo', 'FalleN', 'olofmeister', 'Brollan', 'oskar', 'woxic',
                     'glaive', 'AdreN', 'jks', 'Zyw0o', 'ropz', 'boltz', 'device',
                     'KRIMZ', 'simple')

# separating dataframe with the desired columns
df <- players[players$player_name %in% selected_players &
              players$best_of != 5,]
df <- df[c('player_name', 'player_id', 'best_of', 'kast', 'rating',
           'deaths', 'kills', 'map_1', 'map_2', 'map_3', 'hs')]

summarized_players <- c()
for(player in selected_players){
  summarized_players <- c(summarized_players, summarise_player(player,
                                                                df))
}

# creating and populating dataframe with desired info
summary <- matrix(summarized_players, nrow=32, byrow=TRUE)
players_summary <- cbind(matrix(selected_players, ncol = 1), summary)
colnames(players_summary) <- c("Player", "maps_played", "kills",
                              "deaths", "hs", "mean_rating", "mean_kast", "mean_kills",
                              "mean_deaths", "mean_hs")

players_summary <- data.frame(players_summary)

#####
# Plotting
#####

# function to format y axis
fmt_dcimals <- function(decimals=0){
  function(x) format(x, nsmall = decimals, scientific = FALSE)
}

# changing values to numeric
players_summary$mean_kills=as.numeric(levels(players_summary$mean_kills))[players_summary$mean_kills]
players_summary$mean_deaths=as.numeric(levels(players_summary$mean_deaths))[players_summary$mean_deaths]
players_summary$mean_rating=as.numeric(levels(players_summary$mean_rating))[players_summary$mean_rating]

```

```

pc1 <- ggplot(players_summary, aes(x = mean_kills, y = mean_deaths)) +
  geom_point(
    shape = 21,
    stroke = 3,
    fill = "white",
    aes(color = factor(Player), size = mean_rating)
  )
pc1

pc2 <- pc1 +
  ggtitle(
    "Top 32 Players Last Year"
  ) +
  labs(
    color = "Players",
    size = "Rating",
    x = "Kills (mean)",
    y = "Deaths (mean)"
  ) +
  scale_y_continuous(
    labels = fmt_dcmals(1)
  )
pc2

pc3 <- pc2 +
  theme_minimal() +
  theme(
    text = element_text(color = "gray20"),
    legend.title = element_text(size=12, face="bold"),
    legend.position = "right",
    legend.direction = "vertical",
    legend.justification = 0.4,
    legend.text = element_text(size = 11, color = "gray10"),
    axis.text = element_text(face = "italic"),
    axis.title.x = element_text(vjust = -1, face="italic"),
    axis.title.y = element_text(vjust = 2, face="italic"),
    axis.ticks.y = element_blank(),
    axis.line = element_line(color = "gray40", size = 0.5),
    axis.line.y = element_blank(),
    panel.grid.major = element_line(color = "gray50", size = 0.5),
    panel.grid.major.x = element_blank(),
    plot.caption = element_text(hjust = 0),
    plot.title = element_text(size = 16, face="bold")
  )
pc3

```
