Analise Exploratoria

Mikael M Coletto

2023-06-30

Índice

# 1. Análise dos dados da Steam

## 1.1 Introdução

A ideia desta análise é ilustrar os conhecimentos adquiridos em R, começando com uma análise exploratória dos dados e tentando trazer alguns insights relacionados à jogos de sucesso no mercado. Para isso vou passar pelas etapas de elaboração do objetivo, trabalho no tratamento e limpeza dos dados, fazer análises de variáveis de interesse individualmente e depois cruzando algumas delas para tentar traçar algumas conclusões, que, por fim, podem sugerir alguns caminhos para empresas que pretendem lançar seus jogos terem sucesso na Steam.

## 1.2 O objetivo

Somos um estúdio de jogos e queremos analisar:

1. Qual gênero/categoria de jogo tem mais sucesso na steam, definindo sucesso como melhores avaliações;
2. Também queremos analisar qual faixa de preço é mais interessante para o lançamento desse novo jogo, como a faixa de preço se relaciona com o número de vendas e com a categoria.

## 1.3 Análise exploratória das bases de dados

Inicialmente irei fazer uma análise exploratória de algumas bases de dados e suas variáveis que podem ser interessantes para determinar sucesso de um jogo, a faixa de preço, a quantidade de vendas e relacionar essas questões aos gêneros e categorias. Foram feitas análises de três bases de dados, a primeira mais completa, e as duas seguintes mais focadas em reviews de usuários. São bases de dadas extraídas do Kaggle que serão utilizadas para o estudo.

### 1.3.1 Base de dados - 1

### 1.3.2 Base de dados - 2

### 1.3.3 Base de dados - 3

### 1.3.4 Diferenças entre base 1 e 3

## 1.4 Unindo as bases

## 1.5 Analisando dados

Agora que já uni as duas bases, e tenho uma quantidade mais razoável de jogos a serem analisados, precisarei definir qual será a forma de avaliar os estilos/gêneros/categorias dos jogos, por ser uma coluna presente em ambas as bases, a variável tags é uma ótima candidata, porém ela apresenta um número de possibilidades bastante grande e distinta, o que poderia ser prejudicial à análise. Já as outras variáveis categories e genres que são mais específicas, não estão presentes em boa parte dos dados. A análise será feita inicialmente nas três para posterior decisão de qual ou quais usar.

### 1.5.1 Categorias

if(params$run\_all\_code == "F"){  
 df\_all\_games <- data.table::fread(here::here("data-raw/created-tables/db-all-games-tags-all.csv"))  
}  
rows\_categ <- df\_all\_games |>   
 dplyr::filter(!(is.na(categories)) & categories != "") |>   
 nrow()

Ou seja, na base de dados aproximadamente 60% das linhas não tem informação de categorias, a variável poderia ser utilizada, porém com algumas ressalvas.

### 1.5.2 Gêneros

rows\_genres <- df\_all\_games |>   
 dplyr::filter(!(is.na(genres)) & genres != "") |>   
 nrow()

Novamente, cerca de 60% da base não possui informações de gênero.

### 1.5.3 Categorias e Gênero

rows\_gen\_cat <- df\_all\_games |>   
 dplyr::filter(!(is.na(categories)) & categories != "") |>   
 dplyr::filter(!(is.na(genres)) & genres != "") |>   
 nrow()

E, para efeito de comparação, podemos ver que os valores sem categorias e gênero tem o mesmo percentual (60%), o que faz bastante sentido já essas informações faltantes vem da primeira base que não possuía essas colunas, os valores que temos, os 40%, são dos jogos adicionados pela base 1.

### 1.5.4 Tags

if(params$run\_all\_code == "F"){  
 rows\_tags <- df\_all\_games |>   
 dplyr::filter(!(is.na(tags\_all)) & tags\_all != "") |>   
 nrow()  
}else{  
 rows\_tags <- df\_all\_games |>   
 dplyr::filter(!(is.na(tags)) & tags != "" ||  
 (is.na(tags\_extra)) & tags\_extra != "") |>   
 nrow()  
}

Já no caso das tags, 100% das observações possuem alguma informação nas variáveis tags ou tags\_extra (o que também faz sentido, já que unimos duas bases diferentes e uma delas possuia a coluna tags e a outra tags\_extra).

Portanto, farei uma análise mais aprofundada desta variável para gerar uma única coluna que possa ser analisada.

Como uni duas bases de dados, precisamos tratar e unificar a nossa coluna que será definida como classificadora dos jogos. A base 1 possuía dados nas colunas genres, categories e tags, já a base 3 apenas em tags.

A coluna que se adequava melhor aos objetivos no caso da base 1, era a variável tags, já que possuía mais classificações para os jogos que a variável genres, normalmente possuindo tudo que estava presente nela e mais algumas outras características. E a variável Categories estava mais associada ao modo de jogo e informações da Steam, como se possuía *Achievements*, conquistas nos jogos, se tinha suporte a controle pela Steam, etc.

Como alguns jogos estavam em ambas as bases de dados, por fim acabei ficando com duas colunas de classificação, e para não perder nenhuma classificação, achei interessante comparar as características e adicionar as faltantes, fazendo uma união de tudo que aparecia em ambas as variáveis.

##### 1.5.4.0.1 Separação de tags

teste\_interno <- T  
if(teste\_interno){  
df\_all\_games <- data.table::fread(here::here("data-raw/created-tables/db-reviews-join.csv")) |>   
 dplyr::mutate(tags = ifelse(tags == "", NA, tags),  
 categories = ifelse(categories == "", NA, categories),  
 genres = ifelse(genres == "", NA, genres),  
 tags\_extra = ifelse(tags\_extra == "", NA, tags\_extra))  
}  
  
## Verificando as diferenças entre tags de um df e do outro (tags e tags\_extra)  
df\_tags <- dplyr::select(df\_all\_games,   
 app\_id, title, tags, tags\_extra) |>   
 dplyr::filter(!is.na(tags) | !is.na(tags\_extra))  
  
df\_tags\_equal <- df\_tags[tags == tags\_extra,]  
df\_tags\_not\_equal <- df\_tags[tags != tags\_extra,]  
  
## Primeiro check será em jogos que possuem números de tags diferentes  
tags\_dif\_ids <- df\_tags\_not\_equal |>   
 dplyr::select(app\_id, tags, tags\_extra) |>   
 dplyr::mutate(check\_lenght = ifelse(stringr::str\_length(tags) == stringr::str\_length(tags\_extra), T, F))

##### 1.5.4.0.2 Analisando tags diferentes

Então, será criado uma nova coluna para as tags diferentes. É feita uma comparação, entre as duas colunas, e é mantido em um vetor o que temos de elementos iguais (equal), depois em outro vetor o que temos a mais em uma coluna (diff1) e no terceiro o que temos a mais na outra coluna (diff2).

Depois transformei novamente estes vetores em colunas para poder inspecionar como ficaram as novas colunas. E nesse momento verifiquei que por serem bases diferentes, em alguns casos existiam pequenas variações no nome da classificação, que foram ajustados para que não influênciassem na contagem final. Um exemplo destes casos era a categorização “Rogue like”, que em uma base estava definida como “Rogue like” e na outra “Rogue-like”, isso e mais algumas outras categorizações foram normalizadas.

### XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX ####  
### ALTERAR TAGS ALL, usar tidyr e dplyr (consultar fct\_ajustando\_tags)  
### não é possível de forma tão simples, talvez usando trible? (lista dentro do df)  
### Gerar uma lista com os elementos e fazer um "union" entre elas e um "intersect"  
### XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX ####  
  
vector\_tags <- df\_tags\_not\_equal$tags |>   
 purrr::map(\(x) stringr::str\_sort(stringr::str\_split\_1(x, pattern = ",")))  
  
vector\_tags\_ex <- df\_tags\_not\_equal$tags\_extra |>   
 purrr::map(\(x) stringr::str\_sort(stringr::str\_split\_1(x, pattern = ",")))  
  
# vector\_tags\_aux <- vector\_tags[1:5]  
# vector\_tags\_ex\_aux <- vector\_tags\_ex[1:5]  
  
## Criando uma lista com todas as tags q estão presentes em uma coluna e não na outra   
## (estão em tags e não estão em tags\_extra, estão em tags\_extra e não estão em tags)  
diff\_elem <- NULL  
for(i in 1:length(vector\_tags)){  
 elem\_1 <- pluck(vector\_tags, i)  
 elem\_2 <- pluck(vector\_tags\_ex, i)  
   
 diff1 <- setdiff(elem\_1, elem\_2)  
 diff2 <- setdiff(elem\_2, elem\_1)  
 equal <- intersect(elem\_1, elem\_2)  
   
 pluck(diff\_elem, i, 1) <- paste0(equal, collapse = ",")  
 pluck(diff\_elem, i, 2) <- paste0("1 - ", paste0(diff1, collapse = ","), ". 2 - ", paste0(diff2, collapse = ","))  
 pluck(diff\_elem, i, 3) <- pluck(df\_tags\_not\_equal, 1, i)  
}  
  
## Alterando para DF, para melhor visualização  
df\_diff <- data.frame(t(sapply(diff\_elem, c)))  
colnames(df\_diff) <- c("equal", "diff", "id")  
  
df\_diff <- df\_diff |>   
 tidyr::separate\_wider\_delim(diff, ". 2", names = c("first", "seccond")) |>   
 dplyr::mutate(seccond = paste0("2", seccond)) |>   
 dplyr::mutate(equal = as.character(equal),  
 id = as.integer(id))  
  
## Investigando tags com mesmo tamanho, buscando pequenas alterações na escrita  
df\_diff\_ <- df\_diff |>   
 dplyr::filter(!(first == "1 - ") & !(seccond == "2 - ")) |>   
 dplyr::filter(stringr::str\_length(first) == stringr::str\_length(seccond))  
  
## Palavras com grafias levemente diferentes serão alteradas  
## Roguelike e Roguelite para Rogue-like e Rogue-lite,  
## Base Building para Base-Building  
## e-sports para eSports  
## Football (Soccer) para Soccer  
## Football (American) para Football  
## Puzzle Platformer para Puzzle-Platformer  
  
df\_diff <- func\_ajustando\_tags(df\_diff, debug)  
  
  
df\_diff <- df\_diff |>   
 dplyr::select(id, tags)

##### 1.5.4.0.3 Tags NA

Após a separação e tratamento de ambas os conjuntos de dados, observei um número grande de dados que não havia sido incluído em nenhuma base, nem no que possuía as tags iguais nem nos que possuíam as tags diferentes. Descobri que isso foi causado pela forma como o R trata NA, as colunas que possuem NA não são comparadas diretamente, então elas ficaram excluídas da junção anterior. Portanto, tratei de forma que também adicionasse estes casos de forma simples, copiando os dados de uma coluna à outra, para tratar de forma similar à que seria tratado o conjunto de dados que possuía as colunas iguais.

## Como ainda possuíamos um número grande de NAs em uma das duas colunas, a solução foi copiar  
## a coluna preenchida e tratar como se fosse uma junção que não precisaria de tratamento  
## Tags e Tags\_extra NA  
df\_tags\_na\_1 <- df\_tags[is.na(tags)]  
## Clonando a coluna de tags para que seja tratada da mesma forma para o df com tags iguais  
df\_tags\_na\_1$tags <- df\_tags\_na\_1$tags\_extra  
  
df\_tags\_na\_3 <- df\_tags[is.na(tags\_extra)]  
## Clonando a coluna de tags para que seja tratada da mesma forma para o df com tags iguais  
df\_tags\_na\_3$tags\_extra <- df\_tags\_na\_3$tags  
  
df\_tags\_equal <- rbind(df\_tags\_equal, df\_tags\_na\_1, df\_tags\_na\_3)

##### 1.5.4.0.4 Visualzando tags mais frequentes

Após a organização e limpeza das tags, vou começar com uma visualização simples para identificar as tags mais utilizadas (e as menos utilizadas). Outro detalhe é que farei uma junção, agora da tabela com as tags diferentes e com as que defini como tags iguais.

df\_tags\_all <- df\_tags\_equal |>   
 dplyr::select(app\_id, tags) |>   
 rbind(df\_diff |>   
 dplyr::rename(app\_id = id))  
  
all\_ordered <- df\_tags\_all$tags |>   
 purrr::map(\(x) stringr::str\_sort(stringr::str\_split\_1(x, pattern = ",")))  
  
## Colocando tudo numa única lista para contagem  
all\_string\_listwise <- tolower(unlist(lapply(all\_ordered, unique)))  
  
## Fazendo a contagem e transformando em um dataframe, podemos verifiicar diretamente na tabela  
df\_tags\_count <- as.data.frame(table(all\_string\_listwise), stringsAsFactors = F)  
## Renomeando colunas  
names(df\_tags\_count) <- c("tag", "count")  
  
## Escrevendo base de dados  
if(params$overwrite\_tables == "T"){  
 data.table::fwrite(df\_tags\_count, here::here("data-raw/created-tables/df-tags-count.csv"))  
}  
  
## Utilllizamos as 50 tags que mais aparecem e criamos uma nova tabela para visualização  
df\_tags\_count |>   
 dplyr::arrange(desc(count)) |>   
 dplyr::slice\_head(n = 10) |>   
 gt::gt()

| tag | count |
| --- | --- |
| singleplayer | 9632 |
| indie | 9323 |
| action | 8605 |
| adventure | 7752 |
| casual | 5281 |
| strategy | 4669 |
| simulation | 4649 |
| rpg | 4623 |
| multiplayer | 4554 |
| 2d | 4035 |

##### 1.5.4.0.5 Organizando nova tabela

E então, para finalizar a organização das tags, nós vamos excluir as características dos jogos que vieram de ambas as bases, e vamos adicionar essa nova coluna tratada.

df\_all\_games\_new\_tags <- df\_all\_games |>   
 dplyr::select(-tags, -tags\_extra) |>   
 dplyr::inner\_join(df\_tags\_all, by = "app\_id") |>   
 dplyr::rename(tags\_all = tags)  
  
  
## Escrevendo base de dados  
if(params$overwrite\_tables){  
 data.table::fwrite(df\_all\_games\_new\_tags, here::here("data-raw/created-tables/db-all-games-tags-all.csv"))  
}

##### 1.5.4.0.6 Criando base no formato long

Para futuras comparações, decidi criar dois tipos de bases de dados, uma no formato long, com cada linha possuindo uma única tag, e outra com todas as tags agrupadas em uma única coluna, unida por vírgula.

## dataframe filtrando generos e tags que não são considerados jogos  
df\_all\_games\_split <- df\_all\_games\_new\_tags |>   
 dplyr::filter(title != "Houdini Indie") |>   
 tidyr::separate\_longer\_delim(tags\_all, delim = ",") |>   
 dplyr::rowwise()  
  
## Escrevendo base de dados  
if(params$overwrite\_tables){  
 data.table::fwrite(df\_all\_games\_split, here::here("data-raw/created-tables/db-all-games-tags-all-split.csv"))  
}  
  
df\_all\_games\_filt <- df\_all\_games\_split |>   
 dplyr::group\_by(app\_id) |>   
 dplyr::mutate(tags\_all\_ = paste0(tags\_all, collapse = ",")) |>   
 dplyr::distinct(app\_id, .keep\_all = T) |>   
 dplyr::select(-tags\_all) |>   
 dplyr::rename(tags\_all = tags\_all\_) |>   
 dplyr::ungroup()  
  
all.equal(df\_all\_games\_new\_tags, df\_all\_games\_filt)

[1] "target is data.table, current is tbl\_df"

compare::compare(df\_all\_games\_new\_tags, df\_all\_games\_filt, allowAll = T, equal = T)

TRUE  
 dropped attributes

### 1.5.5 Investigando preços

Vou agora me debruçar sobre as faixas de preços e em seguida farei um cruzamento dos dados de preços e tags.

Utilizarei a base criada anteriormente, com a coluna de característica dos jogos ajustada (tags). E farei uma tabela com as informações descritivas relacionadas ao preço, para essa análise irei excluir os jogos gratuitos, para que não influêncie tanto nas métricas.

## Usado para escrever o qmd, porém não preciso utilizar ao rodar a análise completa (outro qmd já fará a leitura desse arquivo)  
df\_all\_games <- data.table::fread(here::here("data-raw/created-tables/db-all-games-tags-all.csv"))  
# ## Base 1  
# df\_base\_1 <- data.table::fread(here::here("data-raw/created-tables/db-games-1.csv"))  
# ## Base 3  
# df\_base\_3 <- data.table::fread(here::here("data-raw/created-tables/db-games-3.csv"))  
  
## Selecionando apenas colunas utilizadas  
df\_payed\_games <- df\_all\_games |>   
 dplyr::filter(price\_final > 0 & price\_original > 0)  
## Verificando quartis  
quant\_games <- quantile(df\_payed\_games$price\_original, probs = c(0, 0.25, 0.5, 0.75, 1))  
  
df\_med\_games <- data.frame(`Métrica` = character(),  
 `Valores` = character())  
  
# summary(df\_payed\_games$price\_original)  
  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mínimo", round(quant\_games[1], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("1° Quartil", round(quant\_games[2], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mediana", round(quant\_games[3], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("3° Quartil", round(quant\_games[4], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Máximo", round(quant\_games[5], 2))  
  
df\_med\_games |>   
 gt::gt()

| Métrica | Valores |
| --- | --- |
| Mínimo | 0.5 |
| 1° Quartil | 5.99 |
| Mediana | 10.99 |
| 3° Quartil | 19.99 |
| Máximo | 109.99 |

## Verificando decis  
# Preço original  
dec\_orig <- quantile(df\_payed\_games$price\_original, probs = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))  
# dec\_orig  
# Preço final  
dec\_fin <- quantile(df\_payed\_games$price\_final, probs = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))  
# dec\_fin

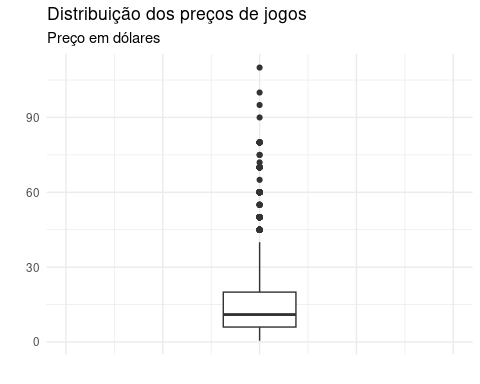
Aqui já posso tirar algumas conclusões:

* As medidas de **tendência central** apontam para **faixas de preço** dos jogos pagos relativamente **baixa**, apesar de termos jogos mais caros (com o máximo sendo de $110). Vamos analisar com mais cautela as categorias de preços dos jogos, já que será uma das bases de decisão na escolha do tipo de jogo a ser lançado.
* Também nos decis comparei a faixa de **preço do jogo lançado**, que será utilizada para as visualizações e análises, com a **faixa de preço atual**. Em geral vi uma **variação para baixo** nos decis quando comparamos o preço de lançamento e o preço atual, indicando que os **jogos tendem a cair de preço** com o tempo, o que é visto com frequência para quem acompanha os preços da Steam, **quando** os jogos começam a ficar **mais velhos**, é comum existirem **cortes de preços e promoções para** tentar **aumentar o número de vendas**.

#### 1.5.5.1 Visualizações das faixas de preço

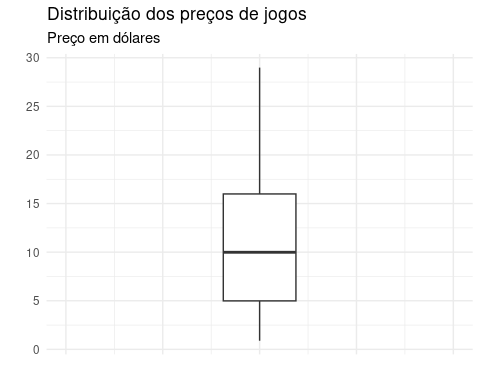
Vou utilizar um gráfico para ilustrar as diferentes faixas de preços dos jogos. Para isto, escolhi um boxplot que apresenta algumas informações interessantes.

ggplot(df\_payed\_games, aes(y = price\_original)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 xlim(-2, 2) +  
 labs(title = "Distribuição dos preços de jogos",  
 subtitle = "Preço em dólares",  
 alt = "Gráfico de distribuição do preço dos jogos em dólares",  
 x = "",  
 y = "") +  
 theme(axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())



E aqui podemos ver que ele classificou muitos jogos como outliers, em faixas superiores à $45, portanto resolvi tirar os jogos do primeiro e último decis, de acordo com seu preço. Depois de remover os 10% inferiores e superiores:

ggplot(df\_payed\_games[(df\_payed\_games$price\_original > dec\_orig[1] & df\_payed\_games$price\_original < dec\_orig[10]),],   
 aes(y = price\_original)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 xlim(-2, 2) +  
 labs(title = "Distribuição dos preços de jogos",  
 subtitle = "Preço em dólares",  
 alt = "Gráfico de distribuição do preço dos jogos em dólares",  
 x = "",  
 y = "") +  
 theme(axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank()) +  
 scale\_y\_continuous(breaks=seq(0, 30, 5))



Observei então uma distribuição com menos valores discrepantes, tendo sua **mediana** em torno dos **$12**, **mínimo próximo de zero** e **máximo próximo de $30**, com **1° quartil em $5** e **3° quartil em $15**.

Investigando a hipótese da possibilidade de descartar estes jogos para a análise, inicialmente eu defendo a exclusão dos mais caros, pois costumeiramente são os jogos mais famosos e conhecidos, que já tem seu público definido e podem se dar ao luxo de cobrar mais caro. Já os jogos mais baratos, imagino que sejam jogos menores e que não se encaixam na nossa proposta de um jogo que fosse lucrativo, muitas vezes também são jogos mais antigos que foram tendo cortes de preços para continuarem vendendo. De qualquer forma, investigaremos nos dados ambas as hipóteses.

##### 1.5.5.1.1 Investigando os 10% mais caros

# browser()  
df\_all\_games\_pricier <- df\_payed\_games |>   
 dplyr::filter(price\_original >= dec\_orig[10]) |>  
 dplyr::select(app\_id, title, date\_release, rating, positive\_ratio, user\_reviews, price\_original, price\_final, tags\_all)

Verifiquei muitos jogos considerados “AAA”, jogos de grandes editoras e publicadoras, como The Witcher 3, Cyberpunk 2077, Red Dead Redemption 2, e alguns um pouco menores mas que fizeram bastante sucesso como Rust, DayZ, Deep Rock Galactica. Com preços variando de $29.99 até $109.99. Citando como exemplo de jogos mais caro em seu preço original o TEKKEN 7, lançado em 2017 e com preço de lançamento de $109.99, interessante de se notar que o preço atual (em maio/2023) do jogo é de $16.49.

##### 1.5.5.1.2 Investigando os 10% mais baratos

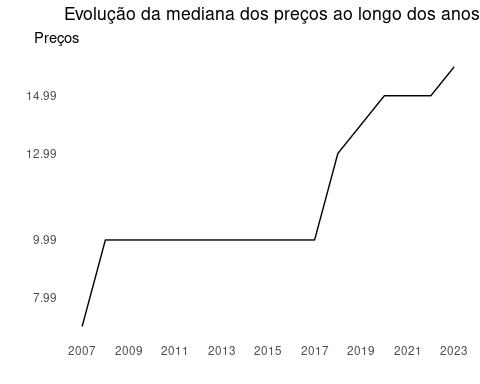
df\_all\_games\_cheaper <- df\_all\_games |>   
 dplyr::filter(price\_original <= dec\_orig[2]) |>  
 dplyr::select(app\_id, title, date\_release, rating, positive\_ratio, user\_reviews, price\_original, price\_final, tags\_all)

Já na análise dos preços mais baratos, verifiquei que os jogos que possuem mais reviews na categoria dos mais baratos, são os jogos gratuitos, porém são jogos que possuem uma estratégia diferente, e que tem crescido bastante em popularidade, os jogos gratuitos com microtransações. Jogos que costumam fazer receita através da venda de itens cosméticos dentro do próprio jogo, ou algum sistema de assinatura que garanta benefícios (cosméticos ou facilitação para o jogador).

##### 1.5.5.1.3 Evolução de preços

Outra coisa interessante que notei, é que os **jogos mais antigos eram lançados por preços mais baixos**, parece ter havido um **aumento de preço médio dos jogos**, o que pode explicar em partes a quantidade de jogos nas faixas mais baratas. Para isso, vamos tentar observar a variação de preço ao longo dos anos, com o intuito de entender as faixas de preços atuais e definir a melhor faixa de preço para o lançamento de um jogo.

## Excluindo jogos gratuitos  
df\_all\_games\_notFree <- df\_all\_games |>   
 dplyr::filter(price\_original > 0 & price\_final > 0)  
  
df\_all\_games\_year <- df\_all\_games\_notFree |>   
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>  
 dplyr::filter(year\_release > 2006) |>   
 dplyr::group\_by(year\_release) |>   
 dplyr::summarise(mean\_price = mean(price\_original), median\_price = median(price\_original),   
 .groups = "drop")  
  
ggplot(df\_all\_games\_year, aes(x = year\_release, y = median\_price)) +  
 geom\_line() +  
 theme\_minimal() +  
 labs(title = "Evolução da mediana dos preços ao longo dos anos",  
 subtitle = "Preços",  
 x = "",  
 y = "") +  
 scale\_y\_continuous(breaks = c(7.99, 9.99, 12.99, 14.99)) +  
 scale\_x\_continuous(breaks = c(seq(2007, 2023, 2))) +  
 theme(panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.subtitle = element\_text(hjust = -0.08))



Apesar de já ter verificado um **aumento na mediana e na média de preços**, por ter acompanhado esse mercado por alguns anos tive a impressão de que o resultado não refletia exatamente os aumentos observados por mim, então decidi me debruçar mais sobre alguns exemplos de jogos considerados “triple A”.

Seguindo alguns exemplos de **jogos que são grandes franquias e são lançados quase que anualmente**, pude observar bem esta tendência de aumento no preço. Vamos usar como exemplo os jogos Assassin’s Creed, Battlefield, Call Of Duty, Far Cry, Final Fantasy, NieR que são franquias já muito conhecidas e que estiveram em alguns períodos no catálogo da Steam (alguns dos jogos não estão presentes de forma integral por figurarem apenas em lojas próprias das empresas). Removi os Remasters nos jogos que possuíam, que são jogos que são relançados geralmente com pequenas alterações gráficas e acabam tendo preço mais baixo, por ser um jogo mais antigo e no caso do Final Fantasy, como existem muitos jogos menores sendo relançados, mantive apenas o preço mais alto do ano. As franquias Battlefield e Call Of Duty também tiveram que ser corrigida em seus anos de lançamento, pois muitos jogos foram adicionados com sua data de lançamento incorreta.

## Selecionando jogos  
df\_ac <- df\_all\_games |>   
 dplyr::filter(stringr::str\_detect(title, "Assassin's Creed")) |>   
 dplyr::filter(!stringr::str\_detect(title, "Remastered|Pack"),  
 categories != "") |>  
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::mutate(title = "Assassin's Creed") |>  
 dplyr::select(title, year\_release, price\_original)  
  
df\_bf <- df\_all\_games |>   
 dplyr::filter(stringr::str\_detect(title, "Battlefield")) |>   
 dplyr::filter(!stringr::str\_detect(title, "WWII|War Dust|Pack|Eternal"),  
 categories != "") |>   
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::select(title, year\_release, price\_original)  
  
df\_bf$year\_release[df\_bf$title == 'Battlefield 3™'] <- 2011  
df\_bf$year\_release[df\_bf$title == 'Battlefield 4™'] <- 2013  
df\_bf$year\_release[df\_bf$title == 'Battlefield™ Hardline'] <- 2015  
df\_bf$year\_release[df\_bf$title == 'Battlefield™ 1'] <- 2016  
df\_bf$year\_release[df\_bf$title == 'Battlefield™ V'] <- 2018  
  
df\_bf <- df\_bf |>   
 dplyr::mutate(title = "Battlefield")  
  
df\_cod <- df\_all\_games |>   
 dplyr::filter(stringr::str\_detect(title, "Call of Duty")) |>   
 dplyr::filter(!stringr::str\_detect(title, "Pack|Commandos|Remastered"),  
 categories != "",  
 price\_original > 0) |>  
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::select(title, year\_release, price\_original)  
  
  
df\_cod$year\_release[df\_cod$title == 'Call of Duty®'] <- 2003  
df\_cod$year\_release[df\_cod$title == 'Call of Duty: United Offensive'] <- 2004  
df\_cod$year\_release[df\_cod$title == 'Call of Duty® 2'] <- 2005  
df\_cod$year\_release[df\_cod$title == 'Call of Duty®: Ghosts'] <- 2013  
df\_cod$year\_release[df\_cod$title == 'Call of Duty®: Modern Warfare®'] <- 2019  
df\_cod$year\_release[df\_cod$title == 'Call of Duty®: Black Ops Cold War'] <- 2020  
  
df\_cod <- df\_cod |>   
 dplyr::mutate(title = "Call Of Duty")  
  
df\_fc <- df\_all\_games |>   
 dplyr::filter(stringr::str\_detect(title, "Far Cry")) |>   
 dplyr::filter(!stringr::str\_detect(title, "Pack"),  
 categories != "") |>  
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::mutate(title = "Far Cry") |>  
 dplyr::select(title, year\_release, price\_original)  
  
df\_ff <- df\_all\_games |>   
 dplyr::filter(stringr::str\_detect(title, "FINAL FANTASY")) |>   
 dplyr::filter(!stringr::str\_detect(title, "Pack|REMASTERED|Remaster|DISSIDIA"),  
 categories != "") |>  
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::mutate(title = "Final Fantasy") |>  
 dplyr::select(title, year\_release, price\_original) |>   
 dplyr::arrange(desc(price\_original)) |>   
 dplyr::distinct(year\_release, .keep\_all = T)  
  
df\_nr <- df\_all\_games |>   
dplyr::filter(stringr::str\_detect(title, "NieR")) |>   
 dplyr::filter(!stringr::str\_detect(title, "Pack"),  
 categories != "") |>  
 dplyr::mutate(year\_release = lubridate::year(date\_release)) |>   
 dplyr::mutate(title = "NieR") |>  
 dplyr::select(title, year\_release, price\_original)  
  
df\_games\_sel <- rbind(df\_ac, df\_bf, df\_cod, df\_fc, df\_ff, df\_nr)  
  
ggplot(df\_games\_sel, aes(x = year\_release, y = price\_original, color = title)) +  
 geom\_line() +  
 theme\_minimal() +  
 labs(title = "Evolução dos preços ao longo dos anos",  
 subtitle = "Preços",  
 x = "",  
 y = "") +  
 facet\_wrap(~title) +  
 scale\_y\_continuous(breaks = c(19.99, 39.99, 59.99, 69.99)) +  
 scale\_x\_continuous(breaks = c(seq(2010, 2022, 6))) +  
 scale\_color\_manual(values = rep(c("lightblue"), 6)) +  
 theme(panel.grid.major = element\_blank(),  
 # panel.grid.minor = element\_blank(),  
 plot.subtitle = element\_text(hjust = -0.09)) +  
 guides(color = "none")



**Confirmei** então **essa tendência de aumento**, mesmo que em alguns casos não seja uma tendência clara, em geral **os preços dos jogos** **considerados “AAA”** **subiram** desde os $19.99 nos anos de 2010 até os $59.99 e até $69.99 em alguns casos até o ano de 2022.

### 1.5.6 Investigando jogos bem avaliados

#### 1.5.6.1 Mais avaliações positivas

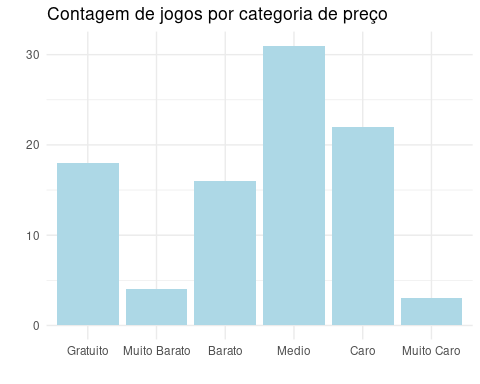
Primeiramente vou analisar os jogos que tem mais avaliações positivas, decidi utilizar os 100 jogos mais bem avaliados da base para tentar trazer algum insight. Além disso, faremos um filtro de ano, para que tenhamos jogos mais atuais.

Para isso, defini alguns filtros para os jogos:

1. Ter sido lançados de 2019 até os dias de hoje, para que tenhamos um mercado observado mais próximo dos dias atuais.
2. Possuírem pelo menos 50000 avaliações.

## Usado para escrever o qmd, porém não preciso utilizar ao rodar a análise completa (outro qmd já fará a leitura desse arquivo)  
df\_all\_games <- data.table::fread(here::here("data-raw/created-tables/db-all-games-tags-all.csv"))  
  
df\_more\_reviewed <- func\_add\_price\_categ(df\_all\_games, F) |>   
 dplyr::filter(user\_reviews > 50000 & date\_release > '2019-01-01') |>  
 dplyr::arrange(desc(user\_reviews)) |>   
 dplyr::slice\_head(n = 100)  
  
## Menor valor de preço  
min\_best <- min(df\_more\_reviewed$price\_final, df\_more\_reviewed$price\_original)  
## Maior valor de preço  
max\_best <- max(df\_more\_reviewed$price\_final, df\_more\_reviewed$price\_original)  
## Mediana de valores de preço  
median\_best <- median(df\_more\_reviewed$price\_original)  
  
## Verificando quartis  
quant\_best <- quantile(df\_more\_reviewed$price\_original, probs = c(0, 0.25, 0.5, 0.75, 1))  
  
  
df\_med\_games <- data.frame(metrica = character(),  
 valores = character())  
  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mínimo", min\_best)  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("1° Quartil", round(quant\_best[2], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mediana", median\_best)  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("3° Quartil", round(quant\_best[4], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Máximo", max\_best)

df\_more\_reviewed$price\_categ <- as.factor(df\_more\_reviewed$price\_categ)  
  
df\_more\_reviewed$price\_categ <- factor(df\_more\_reviewed$price\_categ,  
 levels = c("Gratuito", "Muito Barato", "Barato",  
 "Medio", "Caro", "Muito Caro"))  
  
ggplot(df\_more\_reviewed, aes(x = price\_categ, )) +  
 geom\_bar(fill = "lightblue") +  
 theme\_minimal() +  
 labs(title = "Contagem de jogos por categoria de preço",  
 x = "",  
 y = "")



O primeiro gráfico não trouxe nenhuma conclusão, existem jogos baratos que foram lançados e fizeram sucesso, como Phasmophobia, Valheim e Raft, existem jogos que são gratuitos e lucram através de itens cosméticos como o Apex Legends, Lost Ark e Destiny 2 (que foi pago e se tornou gratuito), e existem jogos caros que foram lançados e fizeram sucesso, como CyberPunk 2077, Elden Ring, Red Dead Redemption 2, Forza e Sekiro. Em resumo, os **jogos com maior número de avaliações positivas não** **parecem ter um padrão relacionado à faixa de preço**.

#### 1.5.6.2 Mais avaliações positivas

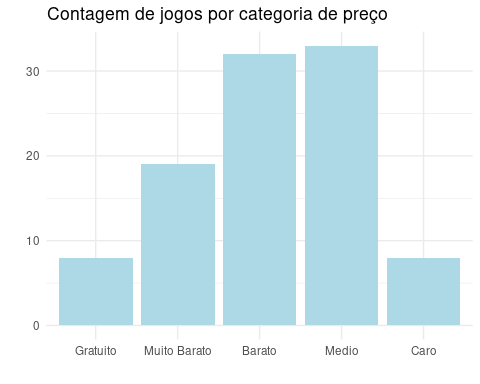
Decidi então observar os jogos que possuiam mais avaliações positivas. E para isso tambem definimos filtros similares.

1 - Também terão que ser lançados recentemente, de 2019 até os dias de hoje.

2 - Possuírem pelo menos 10000 avaliações.

df\_best\_ratio <- df\_all\_games |>   
 dplyr::filter(user\_reviews > 10000 & date\_release > '2019-01-01') |>  
 dplyr::arrange(desc(positive\_ratio)) |>  
 dplyr::slice\_head(n = 100)  
  
## Menor valor de preço  
min\_best <- min(df\_best\_ratio$price\_final, df\_best\_ratio$price\_original)  
## Maior valor de preço  
max\_best <- max(df\_best\_ratio$price\_final, df\_best\_ratio$price\_original)  
## Mediana de valores de preço  
median\_best <- median(df\_best\_ratio$price\_original)  
  
## Verificando quartis  
quant\_best <- quantile(df\_best\_ratio$price\_original, probs = c(0, 0.25, 0.5, 0.75, 1))  
  
  
df\_med\_games <- data.frame(metrica = character(),  
 valores = character())  
  
# summary(df\_payed\_games$price\_original)  
  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mínimo", min\_best)  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("1° Quartil", round(quant\_best[2], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Mediana", median\_best)  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("3° Quartil", round(quant\_best[4], 2))  
df\_med\_games[nrow(df\_med\_games)+1, ] = c("Máximo", max\_best)

df\_best\_ratio <- func\_add\_price\_categ(df\_best\_ratio, F)  
  
df\_best\_ratio$price\_categ <- as.factor(df\_best\_ratio$price\_categ)  
  
df\_best\_ratio$price\_categ <- factor(df\_best\_ratio$price\_categ,  
 levels = c("Gratuito", "Muito Barato", "Barato",  
 "Medio", "Caro", "Muito Caro"))  
  
ggplot(df\_best\_ratio, aes(x = price\_categ)) +  
 geom\_bar(fill = "lightblue") +  
 theme\_minimal() +  
 labs(title = "Contagem de jogos por categoria de preço",  
 x = "",  
 y = "")



Dessa forma, agora ranqueando pelos jogos mais bem avaliados, observei uma **concentração maior** nas **faixas de preço mais baixas**, dos **muito baratos até os médios**, ainda com alguma representação de jogos caros, mas sem jogos muito caros como era visto anteriormente. Comparando com o gráfico anterior, vemos **uma diminuição grande** de **jogos gratuitos e jogos caros**, praticamente metade em ambos os casos, e um **crescimento grande dos jogos muito baratos e baratos**, os **jogos médios se mantiveram** mais ou menos com a mesma proporção, e os jogos muito caros não apareceram no segundo gráfico.

O que **pode indicar** **uma faixa de preço** interessante para o lançamento de um jogo seja tanto esse aumento de **jogos muito baratos e baratos**, quanto essa constância dos jogos de **preço médio**. Porém, vamos tentar analisar outros aspectos usando análises mais complexas.

## 1.6 Analise de variáveis combinadas

### 1.6.1 Investigando variações das tags em categorias de preços

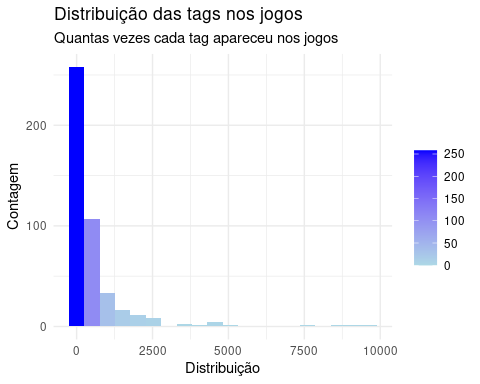
#### 1.6.1.1 Tags e preços

Agora vou cruzar as informações relacionadas às tags e preços.

Vou fazer um agrupamento entre as tags dos jogos e uma recategorização para a variável de preços, visando facilitar a comparação entre os jogos. Essa recategorização irá replicar cada jogo separando as tags, portanto teremos ao final um número de linhas muito maior que o número de jogos da nossa base inicial, porque cada jogo pode possuir um número de 1 à 35 tags.

O número de **tags distintas** na base de dados é de **445**, e para cada tag, a **média** de vezes em que ela aparece nos jogos é de **542** e a mediana é de **185**, e pela diferença entre média e mediana percebemos que a **distribuição** das tags nos jogos é bastante **irregular**, analisaremos então esta diferença de outra forma.

ggplot(df\_categs, aes(x = games, fill = after\_stat(count))) +  
 geom\_histogram(bins = 20) +  
 scale\_fill\_gradient(low = "lightblue", high = "blue") +  
 theme\_minimal() +  
 labs(title = "Distribuição das tags nos jogos",  
 subtitle = "Quantas vezes cada tag apareceu nos jogos",  
 alt = "Histograma de distribuição da contagem de tags nos jogos",  
 x = "Distribuição",  
 y = "Contagem",  
 fill = "")



Utilizando o histograma é possível verificar quantas vezes cada tag apareceu, e temos uma grande parte distribuída entre as primeiras barras do histograma, que foi separado por faixas, porém podemos observar algumas barras muito menores que aparecem mais vezes, ou seja, tags que aparecem em muitos jogos. Algumas dessas mais frequentes inclusive estão presentes em mais de metade dos **17139 jogos**.

#### 1.6.1.2 Análise por faixa de preço

A análise será feita nas faixas de preço definidas, os jogos foram separados em ***Gratuitos*** (preço = 0), jogos **Muito baratos** (preço entre $0.01 e $10.00), jogos **Baratos** (preço entre $10.01 e $20.00), jogos de preço **Médio** (preço entre $20.01 e $40.00), jogos **Caros** (preço entre $40.01 e $60.00) e jogos **Muito caros** (preço de $60.01 ou mais).

Para avaliar os segmentos, defini um número máximo de tags à serem analisadas por faixa de preço, utilizaremos as 20 tags que tem mais aparecem nos jogos por intervalo de preço.

##### 1.6.1.2.1 Avaliando os jogos muito caros (Muito Caro)

## Verificando média e mediana por categoria "Muito Caro"  
mean\_verExp <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Caro"]), 2)  
median\_verExp <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Caro"]), 2)  
sd\_verExp <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Caro"]), 2)

Defini essa faixa de preço dos muito caros como jogos **acima de 60 dólares**, e vemos valores de média (3.19) e mediana (2) próximos, aliados ao desvio padão que também não é alto (3.58), ou seja, temos **pouca variação** do número de **jogos por tag** nessa faixa de preço, indicando uma **distribuição mais uniforme** das categorias.

df\_price\_categ\_verExp <- df\_price\_categ |>   
 dplyr::filter(price\_categ == "Muito Caro") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_verExp |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>  
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Muito Caros")

| Categorias | Contagem Muito Caros |
| --- | --- |
| Action | 21 |
| Simulation | 21 |
| Singleplayer | 18 |
| Multiplayer | 15 |
| Adventure | 11 |
| Co-op | 10 |
| Military | 10 |
| Realistic | 10 |
| Open World | 9 |
| Third Person | 9 |

Observei que a maioria das tags possui poucas amostras, ou seja, na nossa base de dados e em geral na Steam, **poucos jogos são lançados acima dos $60 independente da tag**, o que reflete na pouca quantidade de tags dentre os jogos.

Em comparação com a população de jogos, o tamanho da amostra para essa categoria de preço é pequena comparada à nossa base de dados, do total de **17139** apenas **0.22%** ou 38 jogos (um total de 437 tags) estão nesta faixa de valores.

##### 1.6.1.2.2 Avaliando os jogos caros (Caro)

## Verificando média e mediana por categoria "Caro"  
mean\_exp <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Caro"]), 2)  
median\_exp <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Caro"]), 2)  
sd\_exp <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Caro"]), 2)

Já nos jogos caros, entre **40 e 60 dólares**, os valores de média (18) e mediana (6) possuem uma diferença maior, indicando uma **distribuição mais irregular**. O desvio padrão aumentou (33.27), junto com a diferença entre média e mediana **indica uma variação maior entre tags**.

df\_price\_categ\_exp <- df\_price\_categ |>   
 dplyr::filter(price\_categ == "Caro") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_exp |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Caros")

| Categorias | Contagem Caros |
| --- | --- |
| Singleplayer | 287 |
| Action | 227 |
| Adventure | 173 |
| Multiplayer | 172 |
| RPG | 158 |
| Story Rich | 158 |
| Anime | 143 |
| Open World | 118 |
| Simulation | 109 |
| Great Soundtrack | 108 |

O número de contagem por tag **já é consideravelmente maior** que a faixa anterior, ou seja temos uma quantidade de jogos maior nessa faixa de preços.

Nesta faixa temos **2.12%** dos jogos, ou 363 jogos (um total de 5940 tags) nesse segmento. Indicando um aumento significativo da quantidade de jogos, mas ainda uma proporção pequena na Steam.

##### 1.6.1.2.3 Avaliando os jogos medianos (Medio)

## Verificando média e mediana por categoria "Medio"  
mean\_med <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Medio"]), 2)  
median\_med <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Medio"]), 2)  
sd\_med <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Medio"]), 2)

Os jogos de preço médio, **entre 20 e 40 dólares**, vemos valores de média (72.75), mediana (26) e desvio padrão (134.74) novamente **aumentando sua variação**.

df\_price\_categ\_med <- df\_price\_categ |>   
 dplyr::filter(price\_categ == "Medio") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_med |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Médios")

| Categorias | Contagem Médios |
| --- | --- |
| Singleplayer | 1329 |
| Action | 969 |
| Adventure | 917 |
| Multiplayer | 642 |
| Simulation | 625 |
| Indie | 620 |
| RPG | 568 |
| Story Rich | 568 |
| Strategy | 566 |
| Atmospheric | 534 |

Os números na faixa de preço continuam crescendo, com **10.62%** dos jogos, ou 1821 jogos (um total de 30262 tags) nesse segmento, chegando à um valor mais expressivo de 10% da base de dados.

##### 1.6.1.2.4 Avaliando os jogos “baratos” (Barato)

## Verificando média e mediana por categoria "Barato"  
mean\_low <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Barato"]), 2)  
median\_low <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Barato"]), 2)  
sd\_low <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Barato"]), 2)

Nos jogos de preço baixo, **entre 10 e 20 dólares**, a variação aumenta temos de média 166.79, mediana 62 e desvio padrão 318.74. O que pode ser explicado pelo aumento do número de jogos e concentração de jogos em algumas tags.

df\_price\_categ\_low <- df\_price\_categ |>   
 dplyr::filter(price\_categ == "Barato") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_low |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Baratos")

| Categorias | Contagem Baratos |
| --- | --- |
| Singleplayer | 2976 |
| Indie | 2513 |
| Adventure | 2280 |
| Action | 2263 |
| RPG | 1375 |
| Simulation | 1355 |
| Strategy | 1272 |
| 2D | 1238 |
| Atmospheric | 1201 |
| Story Rich | 1169 |

Os números de jogos para esse intervalo de preço são **26.07%** dos jogos, ou 4468 jogos (um total de 71551 tags) nesse segmento, mais um crescimento grande, agora com um quarto dos dados da nossa base.

##### 1.6.1.2.5 Avaliando os jogos “muito baratos” (Muito Barato)

## Verificando média e mediana por categoria "Muito Barato"  
mean\_verLow <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Barato"]), 2)  
median\_verLow <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Barato"]), 2)  
sd\_verLow <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Muito Barato"]), 2)

Para os jogos de preços **até 10 dólares**, classificados como “Muito Baratos” temos variações ainda crescentes mas mais próximos do segmento anterior, a média é de 189.48, mediana 61 e o desvio padrão 413.06.

df\_price\_categ\_verLow <- df\_price\_categ |>   
 dplyr::filter(price\_categ == "Muito Barato") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_verLow |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Muito Baratos")

| Categorias | Contagem Muito Baratos |
| --- | --- |
| Indie | 4022 |
| Singleplayer | 3324 |
| Action | 3152 |
| Adventure | 2799 |
| Casual | 2424 |
| 2D | 1745 |
| Strategy | 1689 |
| Simulation | 1685 |
| RPG | 1506 |
| Atmospheric | 1298 |

Os números na faixa de preço crescem um pouco mais, com **38.16%** dos jogos, ou 6540 jogos (um total de 80717 tags) nesse segmento, ou seja, mais de um terço da nossa base. Se pensarmos nos dois intervalos mais baratas, temos mais da metade dos jogos cadastrados, sem dúvida onde os jogos estão mais concentrados.

##### 1.6.1.2.6 Avaliando os jogos gratuitos (Gratuitos)

## Verificando média e mediana por categoria "Free"  
mean\_free <- round(mean(df\_price\_categ$count[df\_price\_categ$price\_categ == "Gratuito"]), 2)  
median\_free <- round(median(df\_price\_categ$count[df\_price\_categ$price\_categ == "Gratuito"]), 2)  
sd\_free <- round(sd(df\_price\_categ$count[df\_price\_categ$price\_categ == "Gratuito"]), 2)

Na última faixa de preços, os **jogos gratuitos**, as variações diminuem. A média é de 121, mediana 38 e o desvio padrão 267.31.

df\_price\_categ\_free <- df\_price\_categ |>   
dplyr::filter(price\_categ == "Gratuito") |>   
 dplyr::arrange(desc(count))  
  
df\_price\_categ\_free |>   
 dplyr::slice\_head(n = n\_show) |>   
 dplyr::select(-price\_categ) |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem Gratuitos")

| Categorias | Contagem Gratuitos |
| --- | --- |
| Free to Play | 2402 |
| Indie | 2153 |
| Action | 1973 |
| Singleplayer | 1698 |
| Adventure | 1572 |
| Multiplayer | 1495 |
| Casual | 1493 |
| Strategy | 1054 |
| RPG | 1008 |
| Simulation | 854 |

Agora temos uma pequena diminução do número de jogos em relação ao segmento anterior, podendo ser causada tanto pelas filtragens, sendo uma delas a quantidade de avaliações e usuários por jogo, quanto a própria baixa popularidade nessa faixa de valor, com **22.81%** dos jogos, ou 3909 jogos (um total de 52273 tags) nesse segmento, ainda uma quantidade considerável.

#### 1.6.1.3 Resumo das faixas de preço

Um ponto interessante e que foi observado foi de praticamente **dois terços**, que se encaixaram nos nossos filtros de avaliações positivas e números de usuários, estão nas faixas de preço até $20 (de $0-10 como “Muito Baratos” ou de $10-20 como “Baratos”), e mais 25% de jogos gratuitos, são faixas de preços que exibem uma **grande diversidade de tags e classificações**. É um comportamento interessante, porque por serem **jogos baratos ou gratuitos**, contam com o **preço** como forma de **atrair usuários**, e podem pensar em **estratégias diferentes de acertar um público alvo**.

### 1.6.2 Visualizando top tags por faixa de preço

## Visualizar no formato de tabelas  
df\_price\_categ\_all <- cbind(df\_price\_categ\_verExp |>   
 dplyr::slice\_head(n = n\_show),  
 df\_price\_categ\_exp |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_med |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_low |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_verLow |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_free |>   
 dplyr::slice\_head(n = n\_show)) |>   
 dplyr::rename(tags\_verExp = 1, price\_verExp = 2, count\_verExp = 3,  
 tags\_exp = 4, price\_exp = 5, count\_exp = 6,  
 tags\_med = 7, price\_med = 8, count\_med = 9,  
 tags\_low = 10, price\_low = 11, count\_low = 12,  
 tags\_verLow = 13, price\_verLow = 14, count\_verLow = 15,  
 tags\_free = 16, price\_free = 17, count\_free = 18) |>   
 dplyr::select(-2, -5, -8, -11, -14, -17) |>  
 dplyr::mutate(prop\_free = round(count\_free/sum(count\_free)\*100, 1),   
 prop\_verLow = round(count\_verLow/sum(count\_verLow)\*100, 1),  
 prop\_low = round(count\_low/sum(count\_low)\*100, 1),   
 prop\_med = round(count\_med/sum(count\_med)\*100, 1),   
 prop\_exp = round(count\_exp/sum(count\_exp)\*100, 1),   
 prop\_verExp = round(count\_verExp/sum(count\_verExp)\*100, 1)) |>   
 dplyr::select(tags\_free, count\_free, prop\_free,  
 tags\_verLow, count\_verLow,prop\_verLow,  
 tags\_low, count\_low, prop\_low,  
 tags\_med, count\_med, prop\_med,  
 tags\_exp, count\_exp, prop\_exp,  
 tags\_verExp, count\_verExp, prop\_verExp)  
  
tabela\_price\_categ\_all <- df\_price\_categ\_all |>   
 dplyr::mutate(prop\_free = paste0(prop\_free, "%"),  
 prop\_verLow = paste0(prop\_verLow, "%"),  
 prop\_low = paste0(prop\_low, "%"),  
 prop\_med = paste0(prop\_med, "%"),  
 prop\_exp = paste0(prop\_exp, "%"),  
 prop\_verExp = paste0(prop\_verExp, "%")) |>   
 gt::gt() |>   
 gt::tab\_spanner(  
 label = "Gratuitos",  
 columns = c("tags\_free", "count\_free", "prop\_free")  
 ) |>   
 gt::tab\_spanner(  
 label = "Muito baratos",  
 columns = c("tags\_verLow", "count\_verLow", "prop\_verLow")  
 ) |>   
 gt::tab\_spanner(  
 label = "Baratos",  
 columns = c("tags\_low", "count\_low", "prop\_low")  
 ) |>   
 gt::tab\_spanner(  
 label = "Médios",  
 columns = c("tags\_med", "count\_med", "prop\_med")  
 ) |>   
 gt::tab\_spanner(  
 label = "Caros",  
 columns = c("tags\_exp", "count\_exp", "prop\_exp")  
 ) |>   
 gt::tab\_spanner(  
 label = "Muito Caros",  
 columns = c("tags\_verExp", "count\_verExp", "prop\_verExp")  
 ) |>   
 gt::cols\_label("tags\_free" = "Categoria", "count\_free" = "N", "prop\_free" = "Perc",  
 "tags\_verLow" = "Categoria", "count\_verLow" = "N", "prop\_verLow" = "Perc",  
 "tags\_low" = "Categoria", "count\_low" = "N", "prop\_low" = "Perc",  
 "tags\_med" = "Categoria", "count\_med" = "N", "prop\_med" = "Perc",  
 "tags\_exp" = "Categoria", "count\_exp" = "N", "prop\_exp" = "Perc",  
 "tags\_verExp" = "Categoria", "count\_verExp" = "N", "prop\_verExp" = "Perc")  
  
tabela\_price\_categ\_all

| Gratuitos | | | Muito baratos | | | Baratos | | | Médios | | | Caros | | | Muito Caros | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc |
| Free to Play | 2402 | 15.3% | Indie | 4022 | 17% | Singleplayer | 2976 | 16.9% | Singleplayer | 1329 | 18.1% | Singleplayer | 287 | 17.4% | Action | 21 | 15.7% |
| Indie | 2153 | 13.7% | Singleplayer | 3324 | 14.1% | Indie | 2513 | 14.2% | Action | 969 | 13.2% | Action | 227 | 13.7% | Simulation | 21 | 15.7% |
| Action | 1973 | 12.6% | Action | 3152 | 13.3% | Adventure | 2280 | 12.9% | Adventure | 917 | 12.5% | Adventure | 173 | 10.5% | Singleplayer | 18 | 13.4% |
| Singleplayer | 1698 | 10.8% | Adventure | 2799 | 11.8% | Action | 2263 | 12.8% | Multiplayer | 642 | 8.7% | Multiplayer | 172 | 10.4% | Multiplayer | 15 | 11.2% |
| Adventure | 1572 | 10% | Casual | 2424 | 10.3% | RPG | 1375 | 7.8% | Simulation | 625 | 8.5% | RPG | 158 | 9.6% | Adventure | 11 | 8.2% |
| Multiplayer | 1495 | 9.5% | 2D | 1745 | 7.4% | Simulation | 1355 | 7.7% | Indie | 620 | 8.4% | Story Rich | 158 | 9.6% | Co-op | 10 | 7.5% |
| Casual | 1493 | 9.5% | Strategy | 1689 | 7.1% | Strategy | 1272 | 7.2% | RPG | 568 | 7.7% | Anime | 143 | 8.7% | Military | 10 | 7.5% |
| Strategy | 1054 | 6.7% | Simulation | 1685 | 7.1% | 2D | 1238 | 7% | Story Rich | 568 | 7.7% | Open World | 118 | 7.1% | Realistic | 10 | 7.5% |
| RPG | 1008 | 6.4% | RPG | 1506 | 6.4% | Atmospheric | 1201 | 6.8% | Strategy | 566 | 7.7% | Simulation | 109 | 6.6% | Open World | 9 | 6.7% |
| Simulation | 854 | 5.4% | Atmospheric | 1298 | 5.5% | Story Rich | 1169 | 6.6% | Atmospheric | 534 | 7.3% | Great Soundtrack | 108 | 6.5% | Third Person | 9 | 6.7% |

Para efeito de comparação, por se tratar do modo de jogo e não de uma categoria propriamente dita, eu vou remover as tags “Singleplayer” e Multiplayer” para as futuras análises, mas eles também serão tratados na conclusão.

## Visualizar no formato de tabelas  
df\_price\_categ\_all <- cbind(df\_price\_categ\_verExp |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show),  
 df\_price\_categ\_exp |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_med |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_low |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_verLow |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show),   
 df\_price\_categ\_free |>   
 dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
 dplyr::slice\_head(n = n\_show)) |>   
 dplyr::rename(tags\_verExp = 1, price\_verExp = 2, count\_verExp = 3,  
 tags\_exp = 4, price\_exp = 5, count\_exp = 6,  
 tags\_med = 7, price\_med = 8, count\_med = 9,  
 tags\_low = 10, price\_low = 11, count\_low = 12,  
 tags\_verLow = 13, price\_verLow = 14, count\_verLow = 15,  
 tags\_free = 16, price\_free = 17, count\_free = 18) |>   
 dplyr::select(-2, -5, -8, -11, -14, -17) |>  
 dplyr::mutate(prop\_free = round(count\_free/sum(count\_free)\*100, 1),   
 prop\_verLow = round(count\_verLow/sum(count\_verLow)\*100, 1),  
 prop\_low = round(count\_low/sum(count\_low)\*100, 1),   
 prop\_med = round(count\_med/sum(count\_med)\*100, 1),   
 prop\_exp = round(count\_exp/sum(count\_exp)\*100, 1),   
 prop\_verExp = round(count\_verExp/sum(count\_verExp)\*100, 1)) |>   
 dplyr::select(tags\_free, count\_free, prop\_free,  
 tags\_verLow, count\_verLow,prop\_verLow,  
 tags\_low, count\_low, prop\_low,  
 tags\_med, count\_med, prop\_med,  
 tags\_exp, count\_exp, prop\_exp,  
 tags\_verExp, count\_verExp, prop\_verExp)  
  
tabela\_price\_categ\_all <- df\_price\_categ\_all |>   
 dplyr::mutate(prop\_free = paste0(prop\_free, "%"),  
 prop\_verLow = paste0(prop\_verLow, "%"),  
 prop\_low = paste0(prop\_low, "%"),  
 prop\_med = paste0(prop\_med, "%"),  
 prop\_exp = paste0(prop\_exp, "%"),  
 prop\_verExp = paste0(prop\_verExp, "%")) |>   
 gt::gt() |>   
 gt::tab\_spanner(  
 label = "Gratuitos",  
 columns = c("tags\_free", "count\_free", "prop\_free")  
 ) |>   
 gt::tab\_spanner(  
 label = "Muito baratos",  
 columns = c("tags\_verLow", "count\_verLow", "prop\_verLow")  
 ) |>   
 gt::tab\_spanner(  
 label = "Baratos",  
 columns = c("tags\_low", "count\_low", "prop\_low")  
 ) |>   
 gt::tab\_spanner(  
 label = "Médios",  
 columns = c("tags\_med", "count\_med", "prop\_med")  
 ) |>   
 gt::tab\_spanner(  
 label = "Caros",  
 columns = c("tags\_exp", "count\_exp", "prop\_exp")  
 ) |>   
 gt::tab\_spanner(  
 label = "Muito Caros",  
 columns = c("tags\_verExp", "count\_verExp", "prop\_verExp")  
 ) |>   
 gt::cols\_label("tags\_free" = "Categoria", "count\_free" = "N", "prop\_free" = "Perc",  
 "tags\_verLow" = "Categoria", "count\_verLow" = "N", "prop\_verLow" = "Perc",  
 "tags\_low" = "Categoria", "count\_low" = "N", "prop\_low" = "Perc",  
 "tags\_med" = "Categoria", "count\_med" = "N", "prop\_med" = "Perc",  
 "tags\_exp" = "Categoria", "count\_exp" = "N", "prop\_exp" = "Perc",  
 "tags\_verExp" = "Categoria", "count\_verExp" = "N", "prop\_verExp" = "Perc")  
  
tabela\_price\_categ\_all

| Gratuitos | | | Muito baratos | | | Baratos | | | Médios | | | Caros | | | Muito Caros | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc | Categoria | N | Perc |
| Free to Play | 2402 | 17.2% | Indie | 4022 | 18.6% | Indie | 2513 | 16% | Action | 969 | 15.6% | Action | 227 | 16.2% | Action | 21 | 17.9% |
| Indie | 2153 | 15.4% | Action | 3152 | 14.6% | Adventure | 2280 | 14.5% | Adventure | 917 | 14.8% | Adventure | 173 | 12.4% | Simulation | 21 | 17.9% |
| Action | 1973 | 14.1% | Adventure | 2799 | 13% | Action | 2263 | 14.4% | Simulation | 625 | 10.1% | RPG | 158 | 11.3% | Adventure | 11 | 9.4% |
| Adventure | 1572 | 11.2% | Casual | 2424 | 11.2% | RPG | 1375 | 8.7% | Indie | 620 | 10% | Story Rich | 158 | 11.3% | Co-op | 10 | 8.5% |
| Casual | 1493 | 10.7% | 2D | 1745 | 8.1% | Simulation | 1355 | 8.6% | RPG | 568 | 9.2% | Anime | 143 | 10.2% | Military | 10 | 8.5% |
| Strategy | 1054 | 7.5% | Strategy | 1689 | 7.8% | Strategy | 1272 | 8.1% | Story Rich | 568 | 9.2% | Open World | 118 | 8.4% | Realistic | 10 | 8.5% |
| RPG | 1008 | 7.2% | Simulation | 1685 | 7.8% | 2D | 1238 | 7.9% | Strategy | 566 | 9.1% | Simulation | 109 | 7.8% | Open World | 9 | 7.7% |
| Simulation | 854 | 6.1% | RPG | 1506 | 7% | Atmospheric | 1201 | 7.6% | Atmospheric | 534 | 8.6% | Great Soundtrack | 108 | 7.7% | Third Person | 9 | 7.7% |
| 2D | 802 | 5.7% | Atmospheric | 1298 | 6% | Story Rich | 1169 | 7.4% | Open World | 473 | 7.6% | Third Person | 106 | 7.6% | Flight | 8 | 6.8% |
| Massively Multiplayer | 671 | 4.8% | Puzzle | 1273 | 5.9% | Casual | 1050 | 6.7% | Anime | 362 | 5.8% | Co-op | 97 | 6.9% | RPG | 8 | 6.8% |

Coloquei as informações separadas por categoria de preços, mostrando a proporção de cada tag nas faixas de preço, mas, pelo tamanho da tabela, decidi apresentar de forma diferente.

## Removendo coluna de preço para tabela  
df\_tab <- df\_price\_categ\_all  
#   
# newline <- data.frame(tags\_free = "Total", count\_free = sum(df\_tab$count\_free), prop\_free = 100,  
# tags\_verLow = "Total", count\_verLow = sum(df\_tab$count\_verLow), prop\_verLow = 100,  
# tags\_low = "Total", count\_low = sum(df\_tab$count\_low), prop\_low = 100,  
# tags\_med = "Total", count\_med = sum(df\_tab$count\_med), prop\_med = 100,  
# tags\_exp = "Total", count\_exp = sum(df\_tab$count\_exp), prop\_exp = 100,  
# tags\_verExp = "Total", count\_verExp = sum(df\_tab$count\_verExp), prop\_verExp = 100)  
#   
# # df\_tab <- df\_tab |>   
# # dplyr::select(-dplyr::starts\_with("count")) |>  
# # dplyr::slice\_head(n = 20)  
#   
# df\_tab <- rbind(df\_tab, newline)  
#   
# tabela\_categ\_all <- df\_tab |>   
# DT::datatable()  
#  
# ## Usar na construção do gráfico  
# df\_price\_categ\_graph <- rbind(df\_price\_categ\_verExp |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer")))|>   
# dplyr::slice\_head(n = 10),  
# df\_price\_categ\_exp |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
# dplyr::slice\_head(n = 10),  
# df\_price\_categ\_med |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
# dplyr::slice\_head(n = 10),  
# df\_price\_categ\_low |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
# dplyr::slice\_head(n = 10),  
# df\_price\_categ\_verLow |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
# dplyr::slice\_head(n = 10),  
# df\_price\_categ\_free |>  
# dplyr::filter(!(tags\_all %in% c("Singleplayer", "Multiplayer"))) |>   
# dplyr::slice\_head(n = 10)) |>  
# dplyr::group\_by(price\_categ) |>  
# dplyr::mutate(prop = round(count/sum(count)\*100, 2),  
# price\_categ = as.factor(price\_categ),  
# price\_categ = forcats::fct\_relevel(price\_categ, "Free", "Muito Barato", "Barato", "Medio", "Caro", "Muito Caro")) |>  
# dplyr::ungroup()  
#  
# ggplot(df\_price\_categ\_graph, aes(x = price\_categ, y = prop, group = tags\_all, colour = tags\_all)) +  
# geom\_line() +  
# theme\_minimal() +  
# labs()  
  
## Trabalhar com Highlight Table ou Spark Lines ou Small Multiples

Utilizando a forma gráfica por linhas, por termos muitas tags, o gráfico ficou confuso, então achei melhor voltar ao formato de tabela, mas usando uma técnica diferente.

##### 1.6.2.0.1 Trabalhando com tabela de destaque

library(gt)  
  
df\_tab[1:n\_show,] |>   
 mutate(`Gratuito` = prop\_free, `Muito Barato` = prop\_verLow, `Barato` = prop\_low,  
 `Medio` = prop\_med, `Caro` = prop\_exp, `Muito Caro` = prop\_verExp) |>   
 gt() |>   
 data\_color(  
 columns = c(`Gratuito`, `Muito Barato`, `Barato`,  
 `Medio`, `Caro`, `Muito Caro`),  
 fn = scales::col\_numeric(  
 palette = c("#DAE9E8", "#2A7DC0"),  
 domain = NULL  
 ),  
 apply\_to = "fill"  
 ) |>   
cols\_merge(  
 columns = c(`Gratuito`, tags\_free),  
 pattern = "{2}") |>   
 cols\_merge(  
 columns = c(`Muito Barato`, tags\_verLow),  
 pattern = "{2}") |>   
 cols\_merge(  
 columns = c(`Barato`, tags\_low),  
 pattern = "{2}") |>   
 cols\_merge(  
 columns = c(`Medio`, tags\_med),  
 pattern = "{2}") |>   
 cols\_merge(  
 columns = c(`Caro`, tags\_exp),  
 pattern = "{2}") |>   
 cols\_merge(  
 columns = c(`Muito Caro`, tags\_verExp),  
 pattern = "{2}") |>   
 cols\_move\_to\_start(c(`Gratuito`, `Muito Barato`, `Barato`,  
 `Medio`, `Caro`, `Muito Caro`)) |>   
 gt::cols\_hide(columns = contains(c("tags", "prop", "count")))

| Gratuito | Muito Barato | Barato | Medio | Caro | Muito Caro |
| --- | --- | --- | --- | --- | --- |
| Free to Play | Indie | Indie | Action | Action | Action |
| Indie | Action | Adventure | Adventure | Adventure | Simulation |
| Action | Adventure | Action | Simulation | RPG | Adventure |
| Adventure | Casual | RPG | Indie | Story Rich | Co-op |
| Casual | 2D | Simulation | RPG | Anime | Military |
| Strategy | Strategy | Strategy | Story Rich | Open World | Realistic |
| RPG | Simulation | 2D | Strategy | Simulation | Open World |
| Simulation | RPG | Atmospheric | Atmospheric | Great Soundtrack | Third Person |
| 2D | Atmospheric | Story Rich | Open World | Third Person | Flight |
| Massively Multiplayer | Puzzle | Casual | Anime | Co-op | RPG |

|  |
| --- |
| Cores e percentuais  Paleta de cores |

#### 1.6.2.1 Análises preço e tags

Podemos ver algumas tags de jogos que parecem fazer sucesso, relembrando que as tags não são excludentes, um jogo pode ter mais de uma tag, e em geral todos os jogos possuem muitas delas. Em relação ao modo de jogo, um jogo pode ser definido como sendo de único jogador (Singleplayer) e/ou multijogador (Multiplayer). E existem proporcionalmente mais jogos com modo singleplayer do que multiplayer, ainda que esse filtro esteja presente , provavelmente por conta do custo e da complexidade de se desenvolver um jogo multiplayer.

##### 1.6.2.1.1 Singleplayer vs Multiplayer

Mesmo sendo **mais custosos**, muitos dos **jogos mais avaliados são multiplayer**, dos 20 mais avaliados, apenas não possuem modo multiplayer (The Witcher 3 e Cyberpunk 2077), são jogos que possuem entre 450 mil e 7 milhões de avaliações e em geral, são jogos que fazem bastante sucesso, ou seja, são muito vendidos (no caso dos pagos) e jogados. Alem disso existem **diferentes modelos de negócio** dentro destes jogos mais avaliados, existem muitos **jogos gratuitos** **com vendas** de itens **dentro do jogo**, como o CS:GO, Dota 2, Apex, Destiny 2 e Warframe, que são gratuitos mas vendem benefícios dentro do jogo para quem quiser comprar. Existem **jogos com preços variados**, citando Terraria e Among Us como jogos **muito baratos**, Tom Clancy’s Rainbow Six® Siege e ARK como jogos **baratos**, Rust e The Witcher 3 como jogos **médios** e Elden Ring e Cyberpunk 2077 como jogos **mais novos e caros**, a **única** categoria que **não** apareceu foi a de **muito caros**.

Porém, se **alterarmos a métrica** **para** os jogos mais **bem avaliados**, o cenário muda. Ao ordenarmos por taxa de avaliação positiva, os 20 **jogos mais bem avaliados** com nota máxima **são Singleplayer**, **mas** estes jogos **possuem um número** significativamente **menor de avaliações**, entre 300 e 11 mil **e consequentemente** um **número menor de vendas**, além disso são **jogos com preço mais baixos e jogos menos conhecidos**, e são, em geral, jogos ou baratos ou muito baratos.

##### 1.6.2.1.2 Tags mais populares nos 100 jogos com mais reviews

Uma possível verificação que eu acho interessante de ser feita, são as avaliações das tags dos 100 jogos mais populares, definido pelo número de avaliações positivas no jogo, para verificar dentre esses que tem mais “sucesso”, o que se destaca.

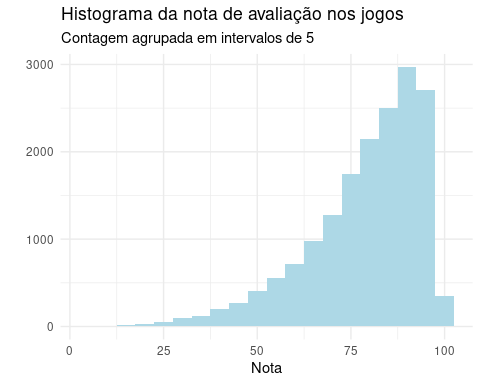
top100\_id <- df\_all\_games |>   
 dplyr::arrange(desc(user\_reviews)) |>   
 dplyr::slice\_head(n = 100) |>   
 dplyr::pull(app\_id)  
  
df\_top100\_tags <- df\_price\_categ\_split |>   
 dplyr::filter(app\_id %in% top100\_id) |>   
dplyr::group\_by(tags\_all) |>   
 dplyr::summarise(count = n(), .groups = NULL) |>   
 dplyr::arrange(desc(count)) |>  
 dplyr::slice\_head(n = 10)  
  
## Jogos do top100  
df\_top100\_games <- df\_all\_games |>  
 dplyr::arrange(desc(user\_reviews))  
  
  
df\_top100\_tags |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem")

| Categorias | Contagem |
| --- | --- |
| Action | 91 |
| Multiplayer | 75 |
| Singleplayer | 69 |
| Adventure | 57 |
| First-Person | 56 |
| Co-op | 55 |
| Open World | 54 |
| Atmospheric | 45 |
| FPS | 41 |
| Shooter | 41 |

Primeiramente vale lembrar que cada jogo pode possuir um número ilimitado de tags, por isso vemos uma contagem tão alta de aparecimento delas. Também é interessante observar como a **tag Ação** passou à frente das demais, estando **presente em quase todos os jogos** (91%), e como no caso dos jogos do top 100, existem **mais jogos que se definem como Multiplayer do que Singleplayer** (Respectivamente, 74% e 69%), o que nos mostra que **boa parte dos jogos** desse top **possui ambos os modos**, tentando agradar diferentes públicos. Outras tags que valem a pena serem citadas como bastante presentes são a de **Cooperativo e Open World** ambos com 54%, que não são tão fortes na nossa tabela geral, e que **ganham mais destaque em jogos mais caros**, indicando possivelmente que são jogos mais complexos e mais custosos de serem feitos. A tag FPS aparece bastante também com 41% e Indie também tem seu destaque com 36%.

Testei novamente a base com os top 200 jogos ao invés de 100, e o resultado foi muito próximo, mostrando o que parece ser uma tendência real dos jogos mais populares.

quant\_top100\_games <- quantile(df\_top100\_games$positive\_ratio, probs = c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))  
  
ggplot(df\_top100\_games, aes(x = positive\_ratio)) +  
 geom\_histogram(binwidth = 5, fill = "lightblue") +  
 theme\_minimal() +  
 labs(title = "Histograma da nota de avaliação nos jogos",  
 subtitle = "Contagem agrupada em intervalos de 5",  
 alt = "Histograma de distribuição da nota de avaliação nos jogos",  
 x = "Nota",  
 y = "",  
 fill = "")



Observando a taxa de avaliações positivas, que vai de 0 a 99 e é calculado com base na relação de opiniões positivas e negativas dos usuários, temos um índice bem alto, na maioria dos jogos. Com praticamente metade desses jogos tendo avaliações acima de 90, e concentração das avaliações entre 70-98. Alguns poucos jogos ficaram abaixo disso, e possivelmente são jogos que podem ser investigados para entender o que motiva essas avaliações tão negativas. O maior exemplo é o Battlefield 2042 com o pior índice de avaliações (35/100), Tale of Immortal (51/100) e PUBG: Battlegrounds (57/100). Em minha experiência, são jogos que realmente foram lançados com problemas e/ou tiveram problemas pós lançamento, portanto é justificável esse resultado baixo nas avaliações.

##### 1.6.2.1.3 Tags em jogos com melhores avaliações

Outra análise interessante é a dos jogos mais bem avaliados, definido pela taxa de avaliações positivas, que pode não ser tão precisa pois acaba tendo jogos com um número de avaliações pequeno em comparação ao tópico anterior, mas pode trazer alguns insights em relação a tags de jogos que são bem avaliadas.

top100\_id <- df\_all\_games |>   
 dplyr::arrange(desc(positive\_ratio)) |>   
 dplyr::slice\_head(n = 100) |>   
 dplyr::pull(app\_id)  
  
df\_top100\_tags <- df\_price\_categ\_split |>   
 dplyr::filter(app\_id %in% top100\_id) |>   
 dplyr::group\_by(tags\_all) |>   
 dplyr::summarise(count = n(), .groups = NULL) |>   
 dplyr::arrange(desc(count)) |>  
 dplyr::slice\_head(n = 10)  
  
df\_top100\_tags |>   
 gt::gt() |>   
 gt::cols\_label("tags\_all" = "Categorias", "count" = "Contagem")

| Categorias | Contagem |
| --- | --- |
| Singleplayer | 66 |
| Indie | 61 |
| 2D | 49 |
| Casual | 49 |
| Adventure | 46 |
| Cute | 37 |
| Action | 34 |
| Story Rich | 32 |
| Puzzle | 31 |
| Pixel Graphics | 28 |

Vemos uma diferença considerável nas tags comparado aos jogos com mais avaliações, agora mais em linha com os jogos gratuitos e baratos. A **tag Singleplayer** está presente em **67% dos jogos**, e a **tag Multiplayer** aparece em **apenas um jogo**, mostrando uma diferença enorme da classificação anterior. Outra tag que **ganha mais destaque** é a de **jogos indie**, com **quase o dobro de jogos** que a avaliação anterior. As **tags Action, Open World e Co-Op caem drásticamente**, com respectivos 32% (contra 91% da anterior), 2% (Contra 54% da anterior) e 2% (Contra 54%). Mostrando que as diferenças nessas duas formas de ranquear os jogos refletem em mudanças bem grandes dos jogos.

##### 1.6.2.1.4 Jogos competitivos

Outra análise adicional que achei interessante de fazer, é sobre jogos com modos competitivos, são jogos que existe toda uma comunidade em volta, de pessoas que transmitem seus jogos através do youtube ou interagindo mais com seu público em lives na twitch ou youtube, gerando assim mais engajamento e aumentando a vida útil do jogo. Os jogadores também costumam se manter no jogo por mais tempo, por criar objetivos dentro do jogo de se melhorar e atingir níveis mais altos de ranking, mostrando melhores habilidades. Também são jogos que geralmente são jogados com amigos e grupos, o que tende a aumentar o tempo que um jogador permanece jogando. Porém, são jogos que também trazem seus problemas, principalmente ligados à balanceamento, por serem jogos que dependem de um público grande, é preciso que existam muitos jogadores para que todos possam jogar contra jogadores de seu nível. Além disso, o modelo de negócio, por depender de uma base de jogadores grande, tende a ser gratuita para que muitos possam jogar, com a venda de itens dentro do jogo, normalmente itens cosméticos, que não influênciam nas mecânicas. A questão da distância dos jogadores com os servidores também pode impactar negativamente a experiência do jogo. Além, é claro, dos jogadores que usam programas ilegais para obterem benefícios como mira automática, mais recursos no jogo, etc.

Vou retornar à lista dos 100 jogos mais populares, e analisar quantos deles tem modos competitivos.

df\_top100\_games\_comp <- df\_top100\_games |>   
 dplyr::filter(stringr::str\_detect(stringr::str\_to\_lower(tags\_all), "comp") |  
 stringr::str\_detect(stringr::str\_to\_lower(categories), "comp") |  
 stringr::str\_detect(stringr::str\_to\_lower(genres), "comp"))

Em análise através das tags, verificamos um total de 10 jogos competitivos, como a minha experiência de jogador achou um número baixo, decidi investigar os 100 jogos um à um, e cheguei num total de 16 jogos (+6 além dos que já haviam sido encontrados). O jogo mais bem avaliado deste top 100 que possui modo competitivo é o Counter Strike (sua primeira versão), que é um fenômeno único e ainda é um jogo muito jogado em suas versões mais novas, tanto que o CS GO também aparece neste top 100. Alguns jogos como Dota 2, Apex, Battlefields e Call of dutys, também apareceram e são jogos que tem seu modo competitivo bastante falado (atualmente).

Concluindo, mesmo totalizando 16 jogos, **não é um número tão relevante** se considerarmos a população de 100 jogos **e dado às complicações** que esta tag pode trazer, **a recomendação** seria de manter **apenas** com modo **Multiplayer casual**, ou seja, sem um ranqueamento, parecem ser mais bem aceitos atualmente.

## 1.7 Conclusão

Ao finalizar essa análise descritiva dos dados de jogos da Steam, pude chegar à algumas conclusões, principalmente após as análises referentes a preços e categorias.

A primeira delas é que **se existe orçamento** capaz de suportar o **desenvolvimento de jogos multiplayers**, a recomendação é que o faça, já que **são os jogos com maior número de jogadores e com mais avaliações**, a interação entre jogadores, quando bem feita, é algo que costuma prender os jogadores e mantê-los jogando por longos períodos de tempo. Nesse caso é preciso estudar o mercado, buscando jogos similares, e **identificar qual modelo de negócio é mais adequado**. Para um **novo estúdio ou nova franquia** os jogos que tem feito mais sucesso são os **jogos gratuitos** com formas de **transações dentro do jogo**, seja algum tipo de facilitação dentro do jogo ou itens cosméticos. A **vantagem dos jogos gratuitos** é que atingem um **grande público**, o que é vital para o desenvolvimento de uma comunidade de jogadores, e **dentro desta comunidade** **uma parcela** **poderá** **gastar dentro do jogo** . Lembrando que, em geral, são jogos que **vendem cosméticos ou acelerações de desempenho**, portanto uma questão **muito importante** é trabalhar de forma que **torne os itens** cosméticos e as acelerações de desempenho itens **desejáveis**, **porém não desbalanceados** a ponto de incomodar os jogadores que não gastam no jogo.

Já caso o **orçamento** seja **limitado**, jogos com **preços baixos** e que **foquem** na **experiência Singleplayer** se saem melhor, são jogos que podem ter um **orçamento menor ou estúdos independentes**. Os jogos Indie, que surgem por estúdios de desenvolvimento de jogos independentes costumam ter um gasto menor com desenvolvimento, frequentemente com equipes reduzidas e **tentam trazer algum diferencial**, podendo ser a **história** do jogo, o **design** **ou** até mesmo alguma **mecânica de jogabilidade**. Jogos como o FTL, Minecraft, Super Meat Boy e Torchlight são exemplos de jogos que começaram como indie e fizeram muito sucesso no mercado de jogos. Inclusive o Minecraft e Torchlight são jogos que perduram até hoje, o primeiro tendo sido comprado por uma gigante no mercado de tecnologias, a Microsoft, e o segundo com o crescimento do estúdio e podendo ser considerado já um estúdio de médio porte.

Ainda, **caso o estúdio possa investir**, porém **não quer se preocupar** com tanto **suporte à longo prazo** de servidores, na questão de hackers ou mesmo tenha foco em modos competivivos, existem **jogos Singleplayers** que tem se destacado no mercado, como o recente Hogwarts Legacy que teve ótima crítica e tem sido bastante falado, e os jogos The Witcher 3 ou Cyberpunk que estão na lista dos 20 jogos mais bem avaliados, contudo, fica a ressalva de que esses jogos **não são jogos comuns**, The Witcher já é o terceiro jogo da série e foi crescendo em público com o tempo, e Cyberpunk é um jogo da mesma desenvolvedora e conseguiu construir uma campanha de marketing para se tornar um jogo muito aguardado. E o Hogwarts Legacy é um jogo que se utiliza de uma franquia muito famosa no mundo do cinema para atrair o público.