Relatório - A2

Angel Machado Elainne Gutiérrez

Junho de 2025

Conteúdo

1 Introdução

Este relatório, em LaTeX, explica com maior abrangência as soluções dadas para as questões do trabalho de Introdução a Computação com Pandas, detalhando os tipos de dados trabalhados no processo. Depois, dedica-se a avaliar as respostas e os gráficos gerados a partir destas, contribuindo para uma análise aprofundada das dinâmicas apresentadas pela base de dados original.

2 Análise estatística descritiva dos dados

A seguir apresentaremos a base de dados sobre filmes, sessões de cinema, entre outros assuntos relacionados ao cinema que utilizaremos para responder as questões e realizar uma Análise exploratória dos dados.

Temos as sete seguintes tabelas:

Tabela	Numero de Linhas	Numero de colunas
distribuidora	72	3
filme	515	6
grupo_exibidor	64	1
exibidor	180	2
complexo	683	4
sala	3.231	3
sessao	1.748.363	5

Tabela 1: Tabelas presentes na base de dados

A análise estatística dos dados de cada tabela, será realizada da seguinte forma:

1. A partir de uma Análise prévia fazemos uma breve descrição sobre o conteúdo da tabela, se necessário, e definimos os tipos de dados de cada coluna de acordo com a Tabela 2
2. Exibimos as primeiras 5 linhas da tabela, onde a primeira linha que exibimos (que não faz parte da tabela) será o tipo de dado de cada coluna
3. Calculamos métricas simples das colunas que contenham dados discretos e datas, tais como: Media, Mediana, Quartis, Valor Máximo, Valor Mínimo, Desvio Padrão
4. Calculamos a Moda das coluna com dados categóricos

cada métrica terá uma breve explicação sobre o que ela significa.

Tipo	Coluna que contem
`object`	texto, rótulos
`category`	valores categóricos
`int64`	inteiros, dados discretos
`datetime64`	datas

Tabela 2: Tipo de dado de cada coluna de acordo com seu conteúdo

2.1 Distribuidora

Contém as distribuidoras de filmes, são empresas responsáveis por levar um filme produzido ao público.

int64	object	object
id	nome	cnpj
1	O2 PRODUÇÕES ARTÍSTICAS E CINEMATOGRÁFICAS LTDA.	67.431.718/0001-03
2	WARNER BROS. (SOUTH) INC.	33.015.827/0001-28
3	THE WALT DISNEY COMPANY (BRASIL) LTDA.	73.042.962/0001-87
4	FREESPIRIT DISTRIBUIDORA DE FILMES LTDA.	07.616.202/0001-01
5	A2 DISTRIBUIDORA DE FILMES LTDA EPP	18.338.912/0001-33

Tabela 3: Primeiras linhas da tabela distribuidora

2.2 Filme

contém os filmes que passaram nos cinemas no ano de 2023, junto com informações relevantes a cada um deles, como título, país de origem e a sua distribuidora.

int64	object	object	object	category	category
id	titulo_original	titulo_br	cpb_roe	pais_origem	from_distribuidora
1	DURVAL DISCOS	None	B0200001000000	BRASIL	24
2	NOSSO LAR	None	B1001259400000	BRASIL	3
3	REMOÇÃO	None	B1301918400000	BRASIL	51
4	HOJE EU QUERO VOLTAR SOZINHO	None	B1402094700000	BRASIL	24
5	O HOMEM DE LAGOA SANTA	None	B1500150500000	BRASIL	40

Tabela 4: Primeiras linhas da tabela filme

- A moda da coluna `pais_origem` é ESTADOS UNIDOS, são 182 filmes com esse país de origem.
- A moda da coluna `from_distribuidora` é a distribuidora com id "2"que corresponde a WARNER BROS. (SOUTH) INC. com 71 filmes que foram distribuídos por ela.

2.3 grupo_exibidor

Esta tabela apenas contém o id de cada grupo exibidor, um grupo exibidor é um conjunto de duas ou mais empresas exibidoras (cinemas) que atuam em conjunto, que nesta base de dados ao total são 64.

int64
id
5002765
6000000
6000002
6000003
6000006

Tabela 5: Primeiras linhas da tabela grupo_exibidor

2.4 exibidor

esta tabela contém o id de cada exibidor junto com o seu respectivo grupo_exibidor

int64	category
id	from_grupo
260	6000049
437	6000013
592	6000030
749	6000058
1660	6000046

Tabela 6: primeiras linhas da tabela exibidor

A moda da coluna `from_grupo` é o id 6000000, com 16 exibidores, isso quer dizer que esse grupo contém 16 exibidores distintos pertencentes a um mesmo grupo empresarial responsável pela operação de salas de cinema.

2.5 complexo

Contém os complexos de salas de cinema (pense em um complexo como um prédio com várias salas de cinema), com a cidade e estado a que pertencem e o exibidor dono desse complexo.

int64	category	category	category
id	municipio	UF	from_exibidor
438	BELO HORIZONTE	MG	437
895	SÃO PAULO	SP	1843
896	SANTO ANDRÉ	SP	1843
897	SÃO PAULO	SP	1843
898	SÃO PAULO	SP	1843

Tabela 7: Primeiras linhas da tabela complexo

A moda da coluna `município` é São Paulo, com 65 complexos de cinema na cidade. Isso quer dizer, em princípio, que São Paulo em 2023 foi a cidade com mais salas de cinema do Brasil.

A moda da coluna `UF` é o estado de São Paulo (SP), com 198 complexos no estado. Interpretando da mesma forma que a moda anterior, isso quer dizer que o estado de São Paulo foi o estado com mais salas de cinema em comparação a outros estados do Brasil.

A moda da coluna `from_exibidor` é o exibidor de id "1843"com 85 complexos de cinema pertencentes a esse exibidor. Dando uma pesquisada rápida, poderíamos suspeitar que esse exibidor é o Cinemark.

2.6 sala

int64	object	category
id	nome	from_complexo
5000056	KINOPLEX IGUAÇU TOP SALA 01 - KINOEVOLUTION	2387
5000057	KINOPLEX IGUAÇU TOP SALA 02	2387
5000058	KINOPLEX IGUAÇU TOP SALA 03	2387
5000061	SALA KINOPLEX MADUREIRA 01	2389
5000062	SALA KINOPLEX MADUREIRA 02	2389

Tabela 8: Primeiras linhas da tabela sala

A moda da coluna `from_complexo` é o complexo de id 2620 com 18 salas de cinema, pesquisando na tabela complexo vemos que ele está na Cidade Rio de Janeiro do Estado de Rio de Janeiro. Podemos tirar uma pequena conclusão de que São Paulo é a cidade com mais salas de cinema, mas Rio de Janeiro possui o complexo com mais salas de cinema.

2.7 sessão

A tabela sessão é a maior tabela presente na base de dados, contendo informações sobre cada sessão registradas nas salas de cinema presentes na tabela `sala` junto com o filme e a data da sessão.

int64	category	category	int64	datetime64[ns]
id	filme_id	sala_id	publico	data_exibicao
1	253	5004897	8	2023-01-01
2	253	5001565	15	2023-01-01
3	253	5001566	14	2023-01-01
4	253	5003464	23	2023-01-01
5	253	5001137	37	2023-01-01

Tabela 9: primeiras linhas da tabela sessão

A seguir algumas métricas simples das colunas, `publico` e `data_exibicao`

Metrica	publico	data_exibicao
Media	63	2023-07-01
Minimo	1	2023-01-01
1° Quartil	11	2023-04-03
Mediana	30	2023-07-04
3° Quartil	75	2023-10-02
Maximo	3.501	2023-12-31
Desvio Padrão	95,62	None

Tabela 10: métricas simples da tabela sessão

A moda da coluna `filme_id` é o filme com id 256 que na tabela `filme` é GATO DE BOTAS 2: O ÚLTIMO PEDIDO com 70676 sessões

A moda da coluna `sala_id` é a sala com id 5001791 que na tabela `sala` é SALADEARTE - CINEMA DO MUSEU da cidade de SALVADOR do estado da BAHIA , com 1176 sessões no ano de 2023

A moda da coluna `data_exibicao` é 08/07/2023 com 6552 sessões, isso quer dizer que o dia com mais sessões foi esse.

3 Questionário comentado

3.1 Questão 1

Qual o total de bilheteria de todos os filmes, ou seja, o público que foi aos filmes listados?

Fazendo o uso de `sum()`, foram somados os valores de cada linha da coluna "`publico`", pertencente à tabela "`sessao`":

```
total_bilheteria_dataframe = sessao.publico
total_bilheteria = sum(total_bilheteria_dataframe)
```

A função `questao_1` retorna um dado do tipo `'int'`, `total_bilheteria`, que representa tal soma:

```
return total_bilheteria
```

3.2 Questão 2

Qual o filme de maior bilheteira em 2023, por país de origem?

Acessamos a duas tabelas da nossa base de dados (`type(database)='str'`), a tabela "`filme`" e a tabela "`sessao`", da primeira extraímos e guardamos na variável `id_titulo_pais` as colunas "`id`", "`titulo_original`" e "`pais_origem`" e da segunda guardamos na variável `filme_id_publico`, "`filme_id`" e "`publico`"

A abordagem para esta questão foi a partir de dicionários e o módulo do Python `defaultdict`; inicialmente criamos as seguintes variáveis com ele:

```
filme_de_maior_bilheteria_por_pais = defaultdict(str)
bilheteria_de_cada_filme_por_id = defaultdict(int)
id_de_cada_filme_por_pais = defaultdict(list)
nome_de_cada_filme_por_id = defaultdict(str)
```

Essas variáveis seguem o padrão `A_por_B` onde B é a chave do valor A, isso nos ajudou a segmentar as partes que queríamos das nossas tabelas, feito da maneira a seguir:

```
for index in id_titulo_pais.itertuples(index= False):
        filme_id = index.id
        titulo = index.titulo_original
        pais = index.pais_origem
        id_de_cada_filme_por_pais[pais].append(filme_id)
        nome_de_cada_filme_por_id[filme_id] = titulo
```

O método `itertuples` nos permite iterar sobre cada linha da tabela `id_titulo_pais` e acessar seus valores nessas linhas, assim a variável `id_de_cada_filme_por_pais` receberá cada país como uma chave, e como o valor de qualquer chave é uma lista por `defaultdict` usamos os métodos `append()` para adicionar o filme no final da lista, assim teremos um dicionário com todos os países existentes na base de dados como chave e cada chave possui uma lista dos filmes pertencentes a esse país, de maneira análoga `nome_de_cada_filme_por_id[filme_id]` recebe o id como chave e o nome de cada filme como valor.

De uma maneira parecida, `bilheteria_de_cada_filme_por_id` terá cada id de filme como chave e sua bilheteria total como valor

```
    for index in filme_id_publico.itertuples(index=False):
        filme_id = index.filme_id
        publico = index.publico
        bilheteria_de_cada_filme_por_id[filme_id] += publico
```

Com nossos dicionários criados iteramos em cada país que esteja como chave no dicionário `id_de_cada_filme_por_pais`, a cada iteração instanciamos as variáveis ‘‘`bilheteria`’’ e ‘‘`filme`’’ com campos vazios, depois com cada filme que tenha esse país como chave no dicionário, comparamos se a bilheteira desse filme é maior do que a variável ‘‘`bilheteria`’’, se for maior a variável é atualizada para esse valor e a variável ‘‘`filme`’’ é atualizada para o nome desse filme, então o dicionário
`filme_de_maior_bilheteria_por_pais` recebe o país da iteração atual como chave e a variável ‘`filme`’ como valor, que vai conter o nome do filme com maior bilheteira nesse país, a seguir o código:

```
    for cada_pais in id_de_cada_filme_por_pais.keys():
        bilheteria = 0
        filme = ""

        for cada_filme_id in id_de_cada_filme_por_pais[cada_pais]:
            if bilheteria_de_cada_filme_por_id[cada_filme_id] > bilheteria:
                bilheteria = bilheteria_de_cada_filme_por_id[cada_filme_id]
                filme = nome_de_cada_filme_por_id[cada_filme_id]

        filme_de_maior_bilheteria_por_pais[cada_pais] = filme
```

com isso a função retorna um DataFrame (‘`pandas.core.frame.DataFrame`’), proveniente da transformação de um ‘`collections.defaultdict`’ em uma lista (‘`list`’), que por sua vez em dataframe.

3.3 Questão 3

Quais são as 100 cidades com maior bilheteria em 2023, ordenadas de forma decrescente de bilheteria?

Da tabela ‘`sessao`’ pegamos as colunas `sala_id` e público, com a função `groupby()`, agrupamos objetos iguais na coluna `sala_id` ou seja, linhas com mesma id serão juntadas numa só linha, mas agora temos todos os públicos de uma só id. Com todos os públicos de uma só id usamos a função `sum()`, assim todos os públicos serão adicionados em um só, teremos como resultado: `sala_id`, `publico_total`, reiniciamos o `index` pois a função `groupby()` coloca `sala_id` automaticamente como `index` e não queremos isso, queremos que `sala_id` seja também uma coluna.

```
sala_id_publico_total = (
        sessao[["sala_id", "publico"]]
        .groupby("sala_id").sum()
        .reset_index()
        )
```

Da tabela ‘`sessao`’ pegamos as colunas, `id` e `complexo`, que contêm o id de cada sala e o complexo ao qual elas pertencem. Renomeamos a coluna id para `sala_id` para conseguir utilizar a função `merge()` sem muita dificuldade, com a tabela anterior.

```
sala_id_compexo = (
    sala[["id","from_complexo"]]
    .rename(columns={"id":"sala_id"})
    )
```

Da tabela `'complexo'` selecionamos as colunas `id` e `município`, então realizamos um processo análogo ao da tabela anterior.

```
complexo_e_cidade = (
    complexo[["id","municipio"]]
    .rename(columns={"id":"from_complexo"})
    )
```

Usamos a função `merge()` para juntar em uma só tabela cada tabela que temos, a partir das colunas com mesmo nome, assim conseguimos uma tabela maior com a que conseguiremos usar a função `groupby()` com cada cidade, agrupando as cidades semelhantes e somando o público total de cada uma

```
sala_id_publico_e_complexo = (
    sala_id_publico_total.merge(
        sala_id_compexo, how="left",on="sala_id"
        )
    )
sala_id_publico_complexo_e_cidade = (
    sala_id_publico_e_complexo.merge(
        complexo_e_cidade, how="left",on="from_complexo"
        )
    )
cidades_bilheteria_total = (
    sala_id_publico_complexo_e_cidade[["municipio","publico"]]
    .groupby("municipio")
    .sum(numeric_only=True)
    )
```

Então ordenamos em ordem decrescente, pegamos as primeiras 100 linhas, reiniciamos o `index` e renomeamos as colunas.

```
cem100_cidades_com_mais_bilheteria = (
    cidades_bilheteria_total.sort_values(
        "publico", ascending=False
        )
    .head(100)
    .reset_index()
    .rename(
        columns={"publico": "bilheteria_total","municipio":"cidade"}
        )
    )
```

com isso a função retorna um DataFrame (`'pandas.core.frame.DataFrame'`) de 100 linhas e 2 colunas (`cidade` e `bilheteria_total`).

3.4 Questão 4

Qual o filme com maior bilheteria em cada cidade?

Acessamos a quatro tabelas da nossa base de dados: "filme", "sessao", "complexo"e "sala".

Renomeamos as colunas selecionadas das tabelas "sala", "complexo" e "filme" para que tenham o mesmo nome entre elas e as escolhidas da tabela "sessao", a fim de poder realizar o método merge entre elas. A partir da tabela "sala", selecionamos as colunas "id" e "from_complexo", renomeando-as para, respectivamente, "sala_id" e "cidade_id", atribuindo esse DataFrame à variável de nome composto pelo nome das colunas; "sala_id__cidade_id":

```
sala_id__cidade__id = (
    sala[["id", "from_complexo"]]
    .rename(columns={"id":"sala_id","from_complexo":"cidade_id"})
    )
```

Partindo da tabela "complexo", selecionamos as colunas "id"e "municipio", renomeando-as para, respectivamente, "cidade_id"e "cidade":

```
cidade_id__cidade = (
    complexo[["id", "municipio"]]
    .rename(columns={"id":"cidade_id","municipio":"cidade"})
    )
```

A partir da tabela filme, selecionamos as colunas id e titulo_original, renomeando a primeira para filme_id.

```
filme_id__titulo_original = (
    filme[["id", "titulo_original"]]
    .rename(columns={"id":"filme_id"})
    )
```

Da tabela "sessao", selecionamos as colunas "sala_id", "filme_id" e "publico". Agrupamos por "sala_id" e "filme_id" semelhantes, somando o público total de cada sala para cada filme.

```
sala_id__filme_id__publico_total = (
    sessao[["filme_id","sala_id","publico"]]
    .groupby(["sala_id","filme_id"])
    .sum()
    .reset_index()
    )
```

Usamos a função merge() para juntar cada tabela que temos em uma só, a partir das colunas com mesmo nome. Assim conseguimos uma tabela maior de 171492 linhas e 4 colunas, sendo estas sala_id, titulo_original, publico e cidade.

```
sala_id__filme_id__publico_total__titulo = (
    sala_id__filme_id__publico_total
    .merge(filme_id__titulo_original, how="left", on="filme_id")
    )

sala_id__cidade_id__cidade = (
    sala_id__cidade__id
```

```
    .merge(cidade_id__cidade, how="left", on="cidade_id")
    )

sala_id__titulo_original__publico_total__cidade = (
    sala_id__filme_id__publico_total__titulo[["sala_id",
    "titulo_original", "publico"]]
    .merge(sala_id__cidade_id__cidade[["sala_id","cidade"]],
        how="left", on="sala_id")
    )
```

Assim, escolhemos todas colunas, exceto `sala_id`, ordenamos a coluna `cidade` em ordem alfabética e os valores da coluna `publico` em ordem decrescente. Com o método `drop_duplicates`, mantivemos apenas as linhas de primeira ocorrência de cada cidade, removendo o resto.

```
cidade__filme__bilheteria = (
    sala_id__titulo_original__publico_total__cidade[["cidade",
    "titulo_original","publico"]]
    .sort_values(by=["cidade","publico"], ascending=[True,False])
    .drop_duplicates(subset="cidade")
    .reset_index(drop=True)
    .rename(columns={"cidade":"CIDADE","titulo_original":"FILME",
    "publico":"BILHETERIA"})
    )
```

com isso a função retorna um DataFrame (`'pandas.core.frame.DataFrame'`) de 345 linhas e 3 colunas (CIDADE, FILME e BILHETERIA).

3.5 Questão 5

Quais as cidades com as maiores bilheterias para filmes brasileiros?

Analogamente à questão anterior, acessamos às quatro tabelas da nossa base de dados

```
sessao = carrega_tabela(database, "sessao")
filme = carrega_tabela(database, "filme")
complexo = carrega_tabela(database, "complexo")
sala = carrega_tabela(database, "sala")
```

e renomeamos as colunas selecionadas das tabelas "`sala`", "`complexo`" e "`filme`" para que tenham o mesmo nome entre elas e as escolhidas da tabela "`sessao`", a fim de poder realizar o método `merge` entre elas. No entanto, como dessa vez escolhemos também a coluna "`pais_origem`"de `sala_id__filme_id__publico_total__titulo`, o DataFrame resultante do `merge` estará acrescido dessa coluna:

```
sala_id__titulo_original__publico_total__pais_origem__cidade = (
    sala_id__filme_id__publico_total__titulo[["sala_id",
    "titulo_original", "publico","pais_origem"]]
    .merge(sala_id__cidade_id__cidade[["sala_id","cidade"]],
    how="left", on="sala_id")
    )
```

Agora, agrupamos por `cidade` e `pais_origem`, somando o público total de cada cidade por país que lançou os filmes citados no database.

```
cidade__pais_origem__publico = (
    sala_id__titulo_original__publico_total__pais_origem__cidade[["cidade",
    "pais_origem","publico"]]
    .groupby(["cidade","pais_origem"])
    .sum("publico")
    .reset_index()
    )
```

Da tabela criada imediatamente acima, criamos esta aqui, escolhendo apenas as linhas cujos públicos por cidade sejam referentes a filmes brasileiros e renomeamos as colunas para os nomes pedidos pela questão

```
cidade__bilheteria_br = (
    cidade__pais_origem__publico
    .loc[cidade__pais_origem__publico["pais_origem"] == "BRASIL"]
    [["cidade","publico"]]
    .reset_index(drop=True)
    .rename(columns={"cidade":"CIDADE","publico":"BILHETERIA_BR"})

    )
```

Pelo mesmo processo, criamos uma tabela escolhendo as linhas cujos públicos por cidade sejam referentes a filmes estrangeiros

```
cidade__bilheteria_estrangeira = (
    cidade__pais_origem__publico
    .loc[cidade__pais_origem__publico["pais_origem"] != "BRASIL"]
    [["cidade","publico"]]
    .groupby("cidade")
    .sum("publico")
    .reset_index()
    .rename(columns={"cidade":"CIDADE","publico":"BILHETERIA_ESTRANGEIRA"})
)
```

Fazemos o `merge` das tabelas sobre bilheterias brasileira e estrangeira, preenchemos seus valores nulos (NaN) com zeros e convertemos os valores de bilheterias do tipo int para o long-int, que permite realizar cálculos com valores inteiros de maior tamanho.

```
cidade__bilheteria_br__bilheteria_estrangeira = (
    cidade__bilheteria_br.merge(cidade__bilheteria_estrangeira,
    how="right", on="CIDADE")
)

cidade__bilheteria_br__bilheteria_estrangeira["BILHETERIA_BR"] = (
    cidade__bilheteria_br__bilheteria_estrangeira["BILHETERIA_BR"]
    .fillna(0)
    .astype("Int64")
)
```

Com isso, a função retorna um DataFrame (‘`pandas.core.frame.DataFrame`’) de 342 linhas e 3 colunas com a bilheteria brasileira em ordem decrescente e com o índice reiniciado.

```
return cidade__bilheteria_br__bilheteria_estrangeira
.sort_values(ascending=False, by="BILHETERIA_BR").reset_index(drop=True)
```

4 Análise exploratória dos dados

O objetivo desta análise exploratória é entender melhor as tendências e os padrões das sessões de cinema do ano de 2023. Queremos conseguir responder às seguintes perguntas:

- Quais são os 10 Filmes mais assistidos e menos assistidos nos cinemas em 2023?
- Qual é a media de o tempo de vida de um filme nos cinemas ?
- como evoluiu a compra de bilheteria e a quantidade de sessões ao longo do ano ?
- Qual o dia da semana em que as pessoas mais assistem filmes nos cinemas ?

para cada pergunta iremos analisar os dados e responder com base neles, criando visualizações que permitam tirar outras conclusões além das perguntas para cumprir com o objetivo desta análise

4.1 Quais são os 10 Filmes mais assistidos e menos assistidos nos cinemas em 2023?

Esta pergunta inicial pode ser respondida através da tabela `filme`, primeiramente agrupamos por filme e somamos a bilheteria de cada filme, obtendo a bilheteria total, depois contamos a quantidade total de sessões de cada filme e finalmente juntamos isso tudo em uma só tabela, que quando organizada em ordem decrescente. tera como primeiras 10 linhas os filmes com maior bilheteria enquanto as ultimas 10 os filmes com menor bilheteria, o resultado é o seguinte:

Id	Bilheteria	Sessões	Titulo Original
399	10.238.598	56.364	BARBIE
324	6.235.953	64.425	THE SUPER MARIO BROS. MOVIE
329	6.132.354	50.614	FAST X
269	5.631.389	65.808	AVATAR: THE WAY OF WATER
256	5.358.525	70.676	PUSS IN BOOTS: THE LAST WISH
363	4.211.764	40.509	GUARDIANS OF THE GALAXY VOL. 3
360	3.987.722	41.205	THE LITTLE MERMAID
389	3.985.017	49.623	ELEMENTAL
344	3.328.797	36.391	SPIDER-MAN: ACROSS THE SPIDER-VERSE
455	2.839.181	36.641	THE NUN 2

Tabela 11: 10 filmes com maior bilheteria

é interessante perceber que dos nenhum dos 10 filmes com maior bilheteria tem como país de origem o Brasil (pois nenhum título original é em português) enquanto nos 10 filmes com menor bilheteria podemos encontrar 5 filmes com Brasil como país de origem

Id	Bilheteria	Sessões	Titulo Original
258	1	1	UNE FEMME DU MONDE
34	3	2	PEQUENOS GUERREIROS
47	4	2	BALA SEM NOME
223	5	1	THE ROUNDUP
244	5	2	WHITE NOISE
10	5	4	MIRANTE
200	7	1	DEMON SLAYER: MUGEN TRAIN
33	8	1	MIÚDA E O GUARDA-CHUVA
240	8	2	FOURMI
26	9	1	ALÉM DA LENDA - FILME

Tabela 12: 10 filmes com menor bilheteria

4.2 Qual é a media de o tempo de vida de um filme nos cinemas ?

Analisaremos isto a partir do público e do número de sessões de cada filme, de forma separada.

primeiro calcularemos o tempo de vida com base na bilheteria que o filme teve em todo 2023,então dado um filme, filtramos a tabela sessão apenas com as sessões que passaram esse filme, depois agrupamos por data e somamos o público, chamaremos essa tabela `bilheteria_por_dia`, calculamos o 1% do máximo da coluna que contém a bilheteria total de cada dia e agora filtramos a tabela `bilheteria_por_dia` para manter apenas as colunas que tiverem público maior do que 1% do máximo que calculamos, depois somamos a quantidade total de dias que restaram e essa soma é o tempo de vida do filme.

Fazendo isso com cada filme, podemos calcular a Média, Moda e Mediana, junto com sua distribuição, do tempo de vida de um filme em relação ao público.

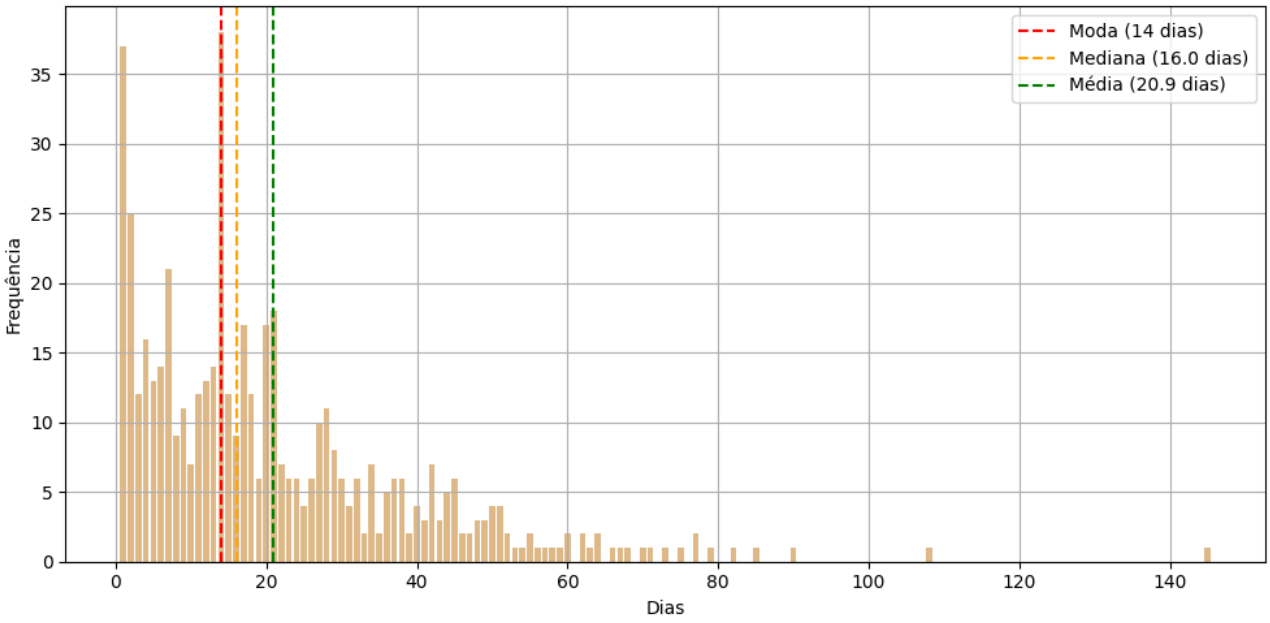

Figura 1: Distribuição do tempo de vida em relação a bilheteria

Respondendo a pergunta inicial então podemos dizer que a média é de 21 dias aproximadamente,vendo a distribuição percebemos alguns outliers, com poucos filmes com tempos de vida superiores a 80 dias, e 37 filmes com apenas um dia de vida que puxam nossa media para baixo, então neste caso poderíamos considerar de bom grado a media de 21 dias.

Podemos então realizar o mesmo procedimento em relação ao numero de total de sessões de um filme, permanecendo apenas com os dias em que o numero de sessões foi maior ao 1% do numero máximo de sessões que um filme já teve, assim obtendo o seguinte gráfico:

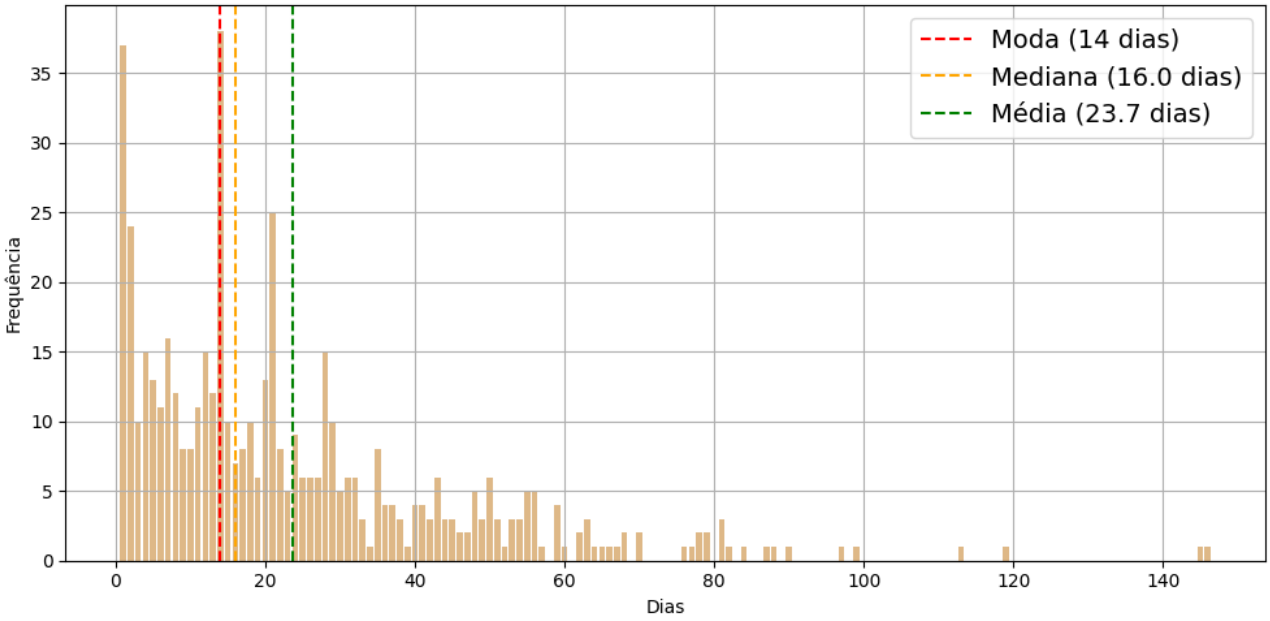

Figura 2: Distribuição do tempo de vida em relação a sessões

os dois gráficos são muito parecidos, demostrando a relação entre bilheteria e o numero de sessões, é interessante perceber que a média de dias de vida em relação a de sessões é 23.7 dias e em relação a de bilheteria é 20.9 dias, com isso podemos concluir que em média um filme sai de cartela 3 dias depois da queda do seu publico.

a seguir apresentaremos dois gráficos, que mostram a evolução do publico e do numero de sessões onde se encontra refletido o tempo de vida, dos 10 filmes com maior bilheteria, a fim de curiosidade.

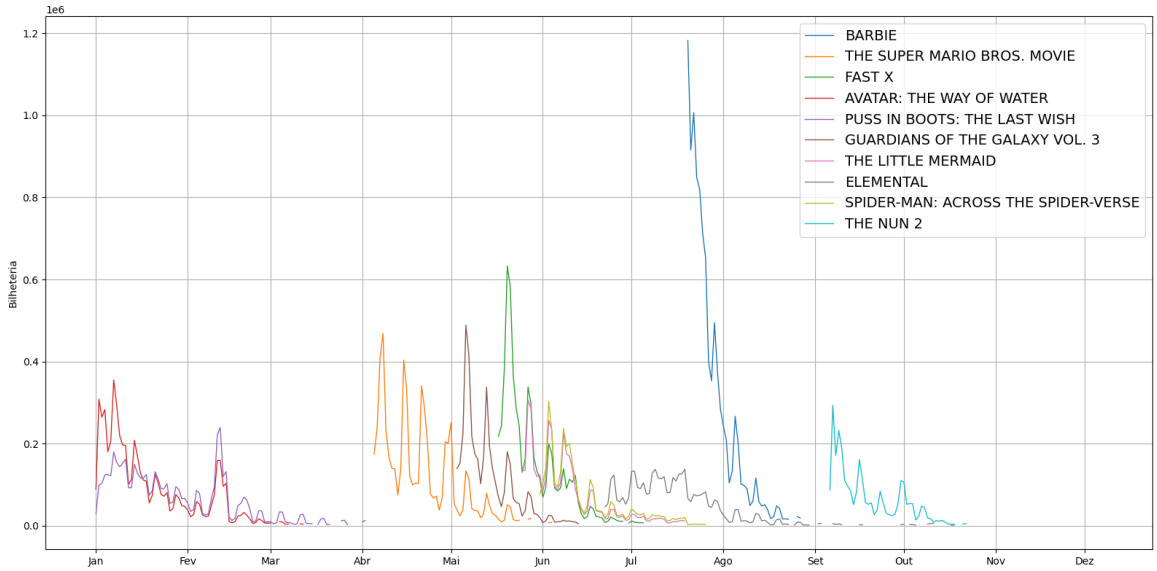

Figura 3: Evolução da bilheteria em relação a 1% do seu máximo, dos 10 filmes com maior bilheteria

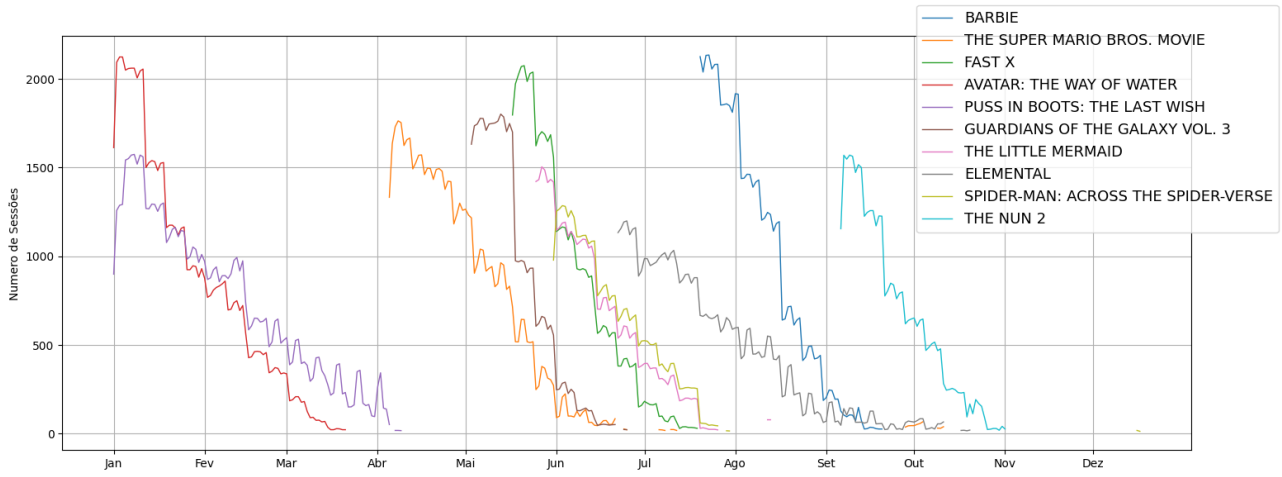

Figura 4: Evolução das sessões em relação a 1% do seu máximo, dos 10 filmes com maior bilheteria

4.3 como evoluiu a compra de bilheteria e a quantidade de sessões ao longo do ano ?

Para responder esta pergunta, agrupamos a tabela `sessao` por data de exibição, e somamos todos os valores da coluna `publico` assim conseguimos a serie temporal, uma maneira análoga é feita para obter a serie temporal das sessões, a seguir os graficos:

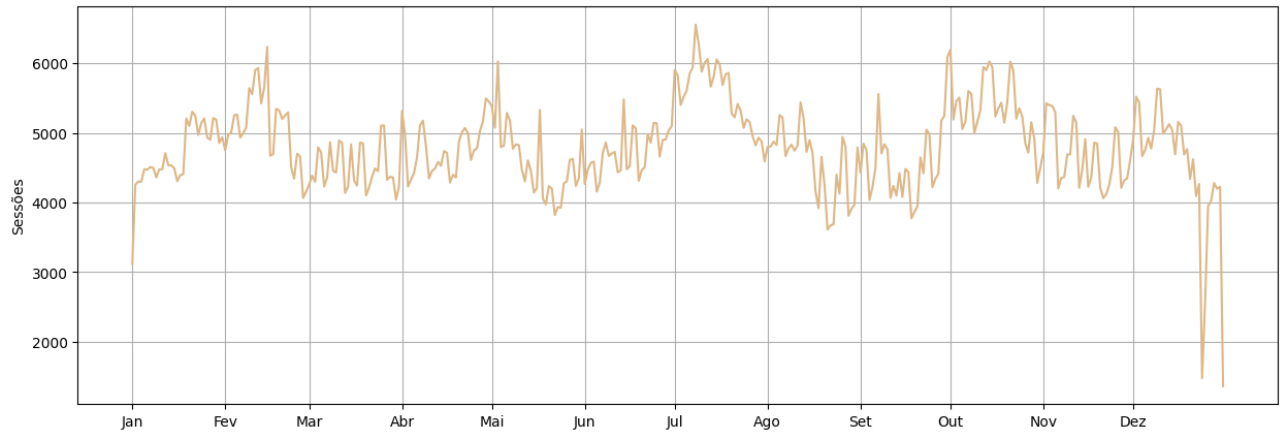

Figura 5: serie temporal por dia, das sessões do ano de 2023

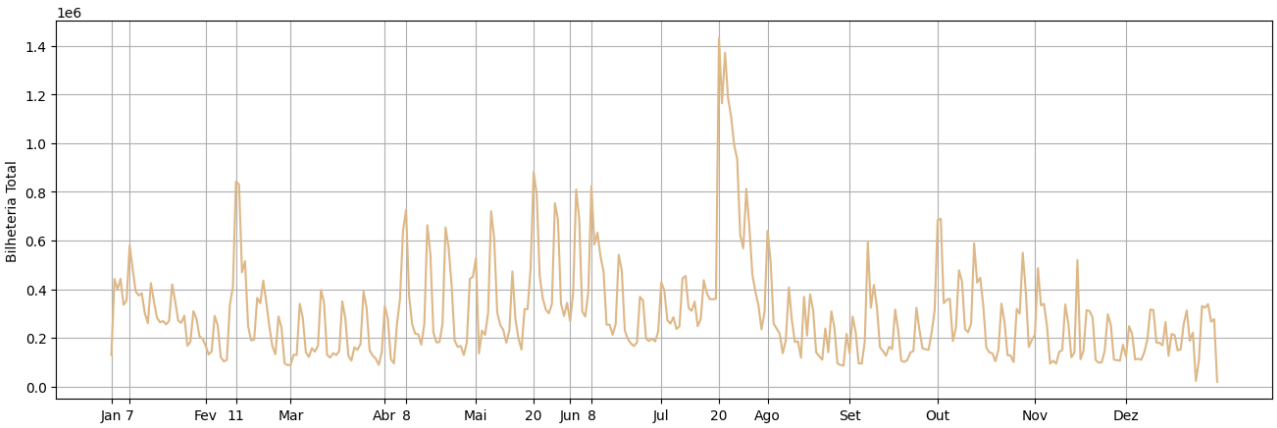

Figura 6: serie temporal por dia, da bilheteria do ano de 2023

É interessante perceber os dias de pico de cada mês da serie temporal por bilheteria. A partir de uma analise rápida podemos concluir o seguinte dos dias de pico:

- dia 10 e 11 de fevereiro que foram os dias pico, os filmes mais assistidos nesses dias foram Gato de Botas 2, Megan e Avatar que ja tinham sido estreados faz um mês mais ou menos, mas por algum motivo que é incerto, estavam tendo seu pico agora, o que sim sabemos é 11 de fevereiro foi sábado.
- Guardiões da Galáxia Vol. 3 estreou em 4 de maio que foi o primeiro pico de maio mas Velozes e Furiosos 10 estreou dia 18 de maio, 18 de maio foi em uma quinta feira por isso maio só teve seu segundo pico no sábado dia 20 de maio.
- 05 de Abril tinha estreado Super Mario Brós o Filme, somente que era quarta feira, logo teve seu pico no sábado dia 8 de abril.
- 01 de junho se estreou Spider-man: Acros the Spider-verse, mas 1 de junho foi domingo, não foi ate o sábado 8 de junho que teve seu pico.
- 20 de julho se estreou Barbie, 20 de julho foi sábado.

4.4 Qual o dia da semana em que as pessoas mais assistem filmes nos cinemas ?

Como vimos na sessão **4.3** todos os dias de pico foram no sábado, podemos ter a hipótese que sábado é o dia em que mais se assistem, vamos confirmar isso, agrupando a tabela `sessao` por data de exibição e separando de 7 em 7 dias, para obter cada dia da semana de segunda a domingo, depois somamos as bilheterias com dias de semana iguais.

Com isso obtemos o seguinte resultado:

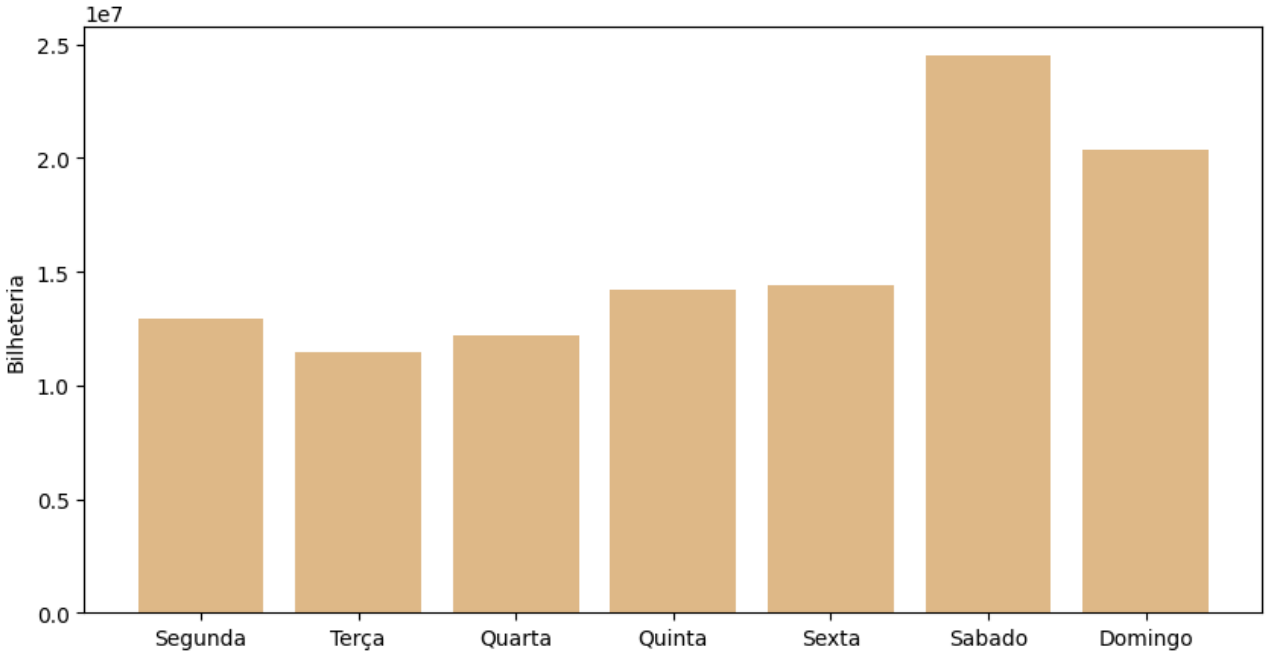

Figura 7: Quantidade total de bilheteria por dia da semana

Assim confirmamos que é no sábado o dia da semana que mais se assistem filmes, junto com o domingo, já os outros dias da semana são similares.

5 Conclusão

Os dados analisados, possuem as bases suficientes para responder questões relevantes ao assunto, mas pode-se dizer que para se aprofundar um pouco mais nas questões precisaríamos de outros dados, estão bem organizados e prontos para uso o que facilitou nossa analise exploratória onde entendemos um pouco mais as sessões de filmes ao longo do ano, o tempo de vida médio dos filmes em cartela, as tendencias em filmes no ano de 2023, e que no final de semana é quando as pessoas mais assistem filmes.