Nome do desafiado: Saulo de Toledo Pereira
Email: saulodetp@gmail.com
Telefone: (11) 98700-5762

Collab do projeto: Clique aqui

Desafio 1: Defesa da 11ª temporada de Friends

O seriado Friends é uma sitcom americana de grande sucesso, apresentada pela rede de televisão NBC entre 1994 e 2004. A série girava em torno das desventuras em Nova York de seis jovens unidos por laços familiares, românticos e, principalmente, de amizade.

Você é o responsável pela análise que servirá de alicerce para a defesa de uma 11ª temporada do seriado, que será apresentada para os investidores. Considerando os dados apresentados, elabore uma análise que responda as seguintes perguntas:

- Quais os padrões, tendências e principais características podemos observar nestes dados?
- A sinopse, escritores e/ou diretor dos episódios influenciam na audiência?
- Usando uma técnica de previsão, qual seria a audiência de uma nova temporada?
- Imaginando que você poderia pedir para os produtores da série outras informações, quais dados você acredita que enriqueceriam a análise e poderiam auxiliar na previsão?

Sumário

O caminho que iremos seguir consiste basicamente de alguns passos que servirão para melhor entendimento do problema e como chegar a uma solução.

1. Entendimento do Desafio:

Lendo o enunciado é possível notar que parece não existir opção de não ter uma 11ª temporada da série, ou seja, independente das analises, se mostrarem bons resultados ou não, temos que convencer os investidores. Então talvez precise de mais do que bons dados, mas muito jogo de cintura de acordo com os resultados que serão extraídos dos dados. Em relação às perguntas, irei respondendo no decorrer da analise, mas ao final de tudo faço um compilado com as perguntas e respostas novamente.

2. Entendimento da empresa/área

A NBC é uma das maiores emissoras televisivas norte-americanas. Diante disso, a emissora obtém um alto faturamento proveniente dos seus anunciantes e zela por uma programação de qualidade para manter seu público, além de possuir alguns recordes históricos de audiência com alguns episódios de séries e jogos de football.

3. Extração/Obtenção de dados

Para esta etapa, já temos a base de dados em uma planilha de Excel. Então já podemos importar os dados para começarmos as analises inicias.

```
## primeiramente é necessário importar a biblioteca que iremos utilizar para a # manipulação e visualização dos dados, o "pandas".

import pandas as pd

## em seguida podemos abrir nossa base de dados e ver como está a 'cara' dela
main_df = pd.read_excel('/content/drive/MyDrive/Friends/DESAFIO1_friends_episodes_aud.xlsx')
main_df
```

Saída:

	Temporada	Episodio	Exibicao_orig	Titulo_orig	Duracao	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
0	1	1	22 de setembro de 1994	The One with the Sonogram at the End	22	Ross finds out his ex-wife is pregnant. Rachel...	James Burrows	David Crane e Marta Kauffman	21.50	8.1	4888
1	1	2	29 de setembro de 1994	The One with the Thumb	22	Monica becomes irritated when everyone likes h...	James Burrows	David Crane e Marta Kauffman	20.20	8.2	4605
2	1	3	6 de outubro de 1994	The One with George Stephanopoulos	22	Joey and Chandler take Ross to a hockey game t...	James Burrows	Jeffrey Astrof e Mike Sikowitz	19.50	8.1	4468
3	1	4	13 de outubro de 1994	The One with the East German Laundry Detergent	22	Eager to spend time with Rachel, Ross pretends...	Pamela Fryman	Alexa Junge	19.70	8.5	4438
4	1	5	20 de outubro de 1994	The One with the Butt	22	Monica's obsessiveness is put to the test afte...	Arlene Sanford	Jeff Greenstein e Jeff Strauss	18.60	8.1	4274
...	...	...	...	...	...	...	...	...	...	...	...
230	10	231	19 de fevereiro de 2004	The One with Princess Consuela	22	When Phoebe goes to get her name changed she r...	Gary Halvorson	Sherry Bilsing-Graham e Ellen Plummer	24.27	8.6	2989
231	10	232	26 de fevereiro de 2004	The One Where Estelle Dies	22	Ross tries to get Rachel to go back to Ralph L...	Gary Halvorson	História por: Robert Carlock | Roteiro por: ...	22.83	8.5	2771
232	10	233	22 de abril de 2004	The One with Rachel's Going Away Party	22	The gang throws Rachel a goodbye party, during...	Gary Halvorson	História por: Mark Kunerth | Roteiro por: Da...	22.64	8.9	3141
233	10	234	29 de abril de 2004	The Last One	30	Erica gives birth to the baby that Monica and ...	Kevin Bright	Andrew Reich e Ted Cohen	24.51	9.5	6221
234	10	235	6 de maio de 2004	The Last One	30	Phoebe races Ross to the airport in a bid to s...	Kevin Bright	Marta Kauffman e David Crane	52.46	9.7	10381

235 rows × 11 columns

4. Ajustes de Dados (Modelagem / limpeza e organização)

Essa etapa é a bem trabalhosa dependendo de como está a base de dados, é nela que iremos ajustar valores, manipular linhas e colunas e etc.

O comando display() mostra as 5 primeiras e 5 ultimas linhas do DataFrame (DF) selecionado além de informações de quantas linhas e colunas ele possui, nesse caso o DF a ser analisado possui 235 linhas e 11 colunas. Se observarmos bem, as linhas 233 e 234 possuem o mesmo nome para o titulo do episódio (EP) (The Last One), então para evitar problemas futuros com duplicação vou verificar se tem mais algum valor duplicado na coluna do título de EPs.

```
## a linha a seguir percorre a coluna 'Titulo_orig' do DF 'main_df'## contando valores dupli
##cados e o ao final somamos para saber quantos arquivos temos

main_df['Titulo_orig'].duplicated().sum()
```

Saída: 5

Como o script nos retornou o valor 5, quer dizer que temos 5 valores (ou nomes) iguais, então agora vamos localiza-los para alterar seus valores.

```
## agora criaremos um novo DF apenas com os valores duplicados para melhor
## visualização

eps_duplicados = main_df[main_df['Titulo_orig'].duplicated()]
display(eps_duplicados)
```

saída:

	Temporada	Episodio	Exibicao_orig	Titulo_orig	Duracao	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
95	4	96	7 de maio de 1998	The One with Ross's Wedding	30	Phoebe tries to warn the gang that Rachel is c...	Kevin Bright	Michael Borkow	31.60	9.2	4217
135	6	136	17 de fevereiro de 2000	The One That Could Have Been	30	The gang continue to think about how different...	Michael Lembeck	Gregory S. Malins e Adam Chase	20.70	8.5	3037
144	6	145	18 de maio de 2000	The One with the Proposal	30	Chandler continues to pretend to hate the idea...	Kevin Bright	Shana Goldberg-Meehan e Scott Silveri	30.70	9.3	4186
216	9	217	15 de maio de 2003	The One in Barbados	22	To the other friends' fury, it keeps raining e...	Kevin Bright	Shana Goldberg-Meehan e Scott Silveri	25.46	8.6	2844
234	10	235	6 de maio de 2004	The Last One	30	Phoebe races Ross to the airport in a bid to s...	Kevin Bright	Marta Kauffman e David Crane	52.46	9.7	10381

Um ponto importante é que o pandas vai selecionar por padrão a ultima linha com o valor duplicado, então sabemos que no index 95 temos um valor duplicado na coluna do titulo do episódio, entretanto não sabemos se o nome anterior é o valor duplicado (index 94). Então antes de tudo, vou consultar a linha com index 94 para saber se o episódio anterior é o que está com o mesmo valor. (Teoricamente deve ser, uma vez que não faria sentido termos um EP e sua continuação em outra temporada, então teoricamente o valor duplicado deve ser o antecessor).

```
## mostrar os valores para ter certeza que estão no index correto

print(main_df.loc[94,'Titulo_orig'])
print(main_df.loc[95,"Titulo_orig"])

print(main_df.loc[143,'Titulo_orig'])
print(main_df.loc[144,"Titulo_orig"])

print(main_df.loc[134,'Titulo_orig'])
print(main_df.loc[135,"Titulo_orig"])

print(main_df.loc[215,'Titulo_orig'])
print(main_df.loc[216,"Titulo_orig"])

print(main_df.loc[233,'Titulo_orig'])
print(main_df.loc[234,"Titulo_orig"])
```

saída:

```
The One with Ross's Wedding
The One with Ross's Wedding
The One with the Proposal
The One with the Proposal
The One That Could Have Been
The One That Could Have Been
The One in Barbados
The One in Barbados
The Last One
The Last One
```

Localizados os EPs com os mesmos nomes, podemos fazer a alteração nos seus valores.

```
## alteração nos valores

main_df.loc[94,'Titulo_orig'] = "The One with Ross's Wedding I"
main_df.loc[95,"Titulo_orig"] = "The One with Ross's Wedding II"

main_df.loc[143,'Titulo_orig'] = "The One with the Proposal I"
main_df.loc[144,"Titulo_orig"] = "The One with the Proposal II"

main_df.loc[134,'Titulo_orig'] = "The One That Could Have Been I"
main_df.loc[135,"Titulo_orig"] = "The One That Could Have Been II"

main_df.loc[215,'Titulo_orig'] = "The One in Barbados I"
main_df.loc[216,"Titulo_orig"] = "The One in Barbados II"

main_df.loc[233,'Titulo_orig'] = "The Last One I"
main_df.loc[234,"Titulo_orig"] = "The Last One II"
```

Depois de feita a mudança nos nomes, podemos rodar o script para verificar se temos mais dados duplicados na coluna de nomes dos EPs.

```
## contagem de valores duplicados novamente para certificar
## que não temos mais nenhum

main_df['Titulo_orig'].duplicated().sum()
```

```
saída: 0
```

Agora sem nomes duplicados, podemos passar para o próximo passo, verificar as informações que o DF pode nos trazer a principio com os comandos .info() e também .describe()

```
## o .info nos traz informações sobre as colunas, valores nulos e o tipo de
## dados que tem em cada coluna (se é int, string, etc)

main_df.info()
main_df.describe()
```

Saída:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 235 entries, 0 to 234
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Temporada       235 non-null    int64
 1   Episodio        235 non-null    int64
 2   Exibicao_orig   233 non-null    object
 3   Titulo_orig     235 non-null    object
 4   Duracao         235 non-null    int64
 5   Sinopse_orig    235 non-null    object
 6   Diretor         235 non-null    object
 7   Escrito_por     235 non-null    object
 8   Audiencia       233 non-null    float64
 9   Estrelas_IMDB   235 non-null    float64
 10  Votos_IMDB      235 non-null    int64
dtypes: float64(2), int64(4), object(5)
memory usage: 20.3+ KB
```

	Temporada	Episodio	Duracao	Audiencia	Estrelas_IMDB	Votos_IMDB
count	235.000000	235.000000	235.000000	233.000000	235.000000	235.000000
mean	5.395745	118.000000	22.340426	34.026052	8.459574	3352.285106
std	2.805821	67.982841	1.517372	134.637156	0.397029	824.214570
min	1.000000	1.000000	22.000000	15.650000	7.200000	2557.000000
25%	3.000000	59.500000	22.000000	22.300000	8.200000	2885.500000
50%	5.000000	118.000000	22.000000	24.460000	8.400000	3147.000000
75%	8.000000	176.500000	22.000000	27.700000	8.700000	3579.500000
max	10.000000	235.000000	30.000000	2079.000000	9.700000	10381.000000

É possível notar certas coisas nessas duas tabelas:

- A primeira tabela nos mostra 4 valores nulos, 2 na coluna 'Exibicao_orig' e mais 2 na coluna 'Audiencia'. Como são apenas 4 valores vou excluir as linhas com esse campos.
- Já na segunda tabela é possível notar discrepância no valor máximo para a coluna 'Audiencia' em que aparece o valor de 2079 onde o certo seria 20.79.

O próximo passo agora será excluir as linhas com valores nulos e alterar o valor errado na coluna 'Audiencia'

```
## o comando .dropna() vai excluir a linha inteira se nela conter algum campo
## com valor vazio ou nulo

main_df = main_df.dropna()
main_df.info()

## ajustando o valor alterado na coluna "Audiencia"
main_df.loc[211,'Audiencia'] = 20.79
main_df.describe()
```

Saída:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 233 entries, 0 to 234
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Temporada      233 non-null    int64
 1   Episodio       233 non-null    int64
 2   Exibicao_orig  233 non-null    object
 3   Titulo_orig    233 non-null    object
 4   Duracao        233 non-null    int64
 5   Sinopse_orig   233 non-null    object
 6   Diretor        233 non-null    object
 7   Escrito_por    233 non-null    object
 8   Audiencia      233 non-null    float64
 9   Estrelas_IMDB  233 non-null    float64
 10  Votos_IMDB     233 non-null    int64
dtypes: float64(2), int64(4), object(5)
memory usage: 21.8+ KB
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1763: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
  isetter(loc, value)
```

	Temporada	Episodio	Duracao	Audiencia	Estrelas_IMDB	Votos_IMDB
count	233.000000	233.000000	233.000000	233.000000	233.000000	233.000000
mean	5.369099	117.450644	22.343348	25.192532	8.457940	3355.549356
std	2.802598	68.004807	1.523568	4.903147	0.398314	826.994052
min	1.000000	1.000000	22.000000	15.650000	7.200000	2557.000000
25%	3.000000	59.000000	22.000000	22.290000	8.200000	2883.000000
50%	5.000000	117.000000	22.000000	24.430000	8.400000	3150.000000
75%	8.000000	176.000000	22.000000	27.520000	8.700000	3581.000000
max	10.000000	235.000000	30.000000	52.900000	9.700000	10381.000000

Tudo certo agora, podemos continuar analisando as colunas do DF.

Na coluna 'Esccrito_por' é possível observar que em alguns campos está escrito 'História por:' e o nome da pessoa além de 'roteiro por:', seria melhor termos apenas os nomes dos escritores então faremos a mudança a seguir:

```
## o código abaixo utiliza o comando .replace() onde é possível manipular o
## conteúdo dos campos trocando por exemplo uma virgula por um espaço vazio.
main_df['Escrito_por'] = main_df['Escrito_por'].replace('História por: ','', regex=True)
main_df['Escrito_por'] = main_df['Escrito_por'].replace('Roteiro por: ','', regex=True)
main_df['Escrito_por'] = main_df['Escrito_por'].replace('&','', regex=True)
main_df['Escrito_por'] = main_df['Escrito_por'].replace('|','', regex=True)
```

Por algum motivo não consigo alterar o '|' sem separar todas as letras (tentei de tudo), fora que tem algumas palavras onde precisa de algum encoding específico (tentei vários), pois teve alguns campos em que o 'história por:' não foi alterado. Então depois de muito tempo, exportei pra Excel, alterei na mão mesmo (tinha uns 30 campos + ou -) e continuamos agora com o DF lido novamente.

	Temporada	Episodio	Exibicao_orig	Titulo_orig	Duracao	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
0	1	1	22 de setembro de 1994	The One with the Sonogram at the End	22	Ross finds out his ex-wife is pregnant. Rachel...	James Burrows	David Crane e Marta Kauffman	21.50	8.1	4888
1	1	2	29 de setembro de 1994	The One with the Thumb	22	Monica becomes irritated when everyone likes h...	James Burrows	David Crane e Marta Kauffman	20.20	8.2	4605
2	1	3	6 de outubro de 1994	The One with George Stephanopoulos	22	Joey and Chandler take Ross to a hockey game t...	James Burrows	Jeffrey Astrof e Mike Sikowitz	19.50	8.1	4468
3	1	4	13 de outubro de 1994	The One with the East German Laundry Detergent	22	Eager to spend time with Rachel, Ross pretends...	Pamela Fryman	Alexa Junge	19.70	8.5	4438
4	1	5	20 de outubro de 1994	The One with the Butt	22	Monica's obsessiveness is put to the test afte...	Arlene Sanford	Jeff Greenstein e Jeff Strauss	18.60	8.1	4274
...	...	...	...	...	...	...	...	...	...	...	...
228	10	231	19 de fevereiro de 2004	The One with Princess Consuela	22	When Phoebe goes to get her name changed she r...	Gary Halvorson	Sherry Bilsing-Graham e Ellen Plummer	24.27	8.6	2989
229	10	232	26 de fevereiro de 2004	The One Where Estelle Dies	22	Ross tries to get Rachel to go back to Ralph L...	Gary Halvorson	Robert Carlock e Tracy Reilly	22.83	8.5	2771
230	10	233	22 de abril de 2004	The One with Rachel's Going Away Party	22	The gang throws Rachel a goodbye party, during...	Gary Halvorson	Mark Kunerth e David Crane e Marta Kauffman	22.64	8.9	3141
231	10	234	29 de abril de 2004	The Last One I	30	Erica gives birth to the baby that Monica and ...	Kevin Bright	Andrew Reich e Ted Cohen	24.51	9.5	6221
232	10	235	6 de maio de 2004	The Last One II	30	Phoebe races Ross to the airport in a bid to s...	Kevin Bright	Marta Kauffman e David Crane	52.46	9.7	10381

233 rows × 11 columns

Pelo visto agora só resta ajustar a coluna de datas, vou cortar os dias e o ano que acredito serem irrelevantes, e ficarão só os meses. Assim podemos ver se os números de audiência podem ter alguma relação com a época do ano em que o EP vai ao ar.

```
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace(' de ','', regex=True)

main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('1','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('2','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('3','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('4','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('5','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('6','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('7','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('8','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('9','', regex=True)
main_df['Exibicao_orig'] = main_df['Exibicao_orig'].replace('0','', regex=True)
main_df
```

	Temporada	Episodio	Exibicao_orig	Titulo_orig	Duracao	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
0	1	1	setembro	The One with the Sonogram at the End	22	Ross finds out his ex-wife is pregnant. Rachel...	James Burrows	David Crane e Marta Kauffman	21.50	8.1	4888
1	1	2	setembro	The One with the Thumb	22	Monica becomes irritated when everyone likes h...	James Burrows	David Crane e Marta Kauffman	20.20	8.2	4605
2	1	3	outubro	The One with George Stephanopoulos	22	Joey and Chandler take Ross to a hockey game t...	James Burrows	Jeffrey Astrof e Mike Sikowitz	19.50	8.1	4468
3	1	4	outubro	The One with the East German Laundry Detergent	22	Eager to spend time with Rachel, Ross pretends...	Pamela Fryman	Alexa Junge	19.70	8.5	4438
4	1	5	outubro	The One with the Butt	22	Monica's obsessiveness is put to the test afte...	Arlene Sanford	Jeff Greenstein e Jeff Strauss	18.60	8.1	4274
...	...	...	...	...	...	...	...	...	...	...	...
228	10	231	fevereiro	The One with Princess Consuela	22	When Phoebe goes to get her name changed she r...	Gary Halvorson	Sherry Bilsing-Graham e Ellen Plummer	24.27	8.6	2989
229	10	232	fevereiro	The One Where Estelle Dies	22	Ross tries to get Rachel to go back to Ralph L...	Gary Halvorson	Robert Carlock e Tracy Reilly	22.83	8.5	2771
230	10	233	abril	The One with Rachel's Going Away Party	22	The gang throws Rachel a goodbye party, during...	Gary Halvorson	Mark Kunerth e David Crane e Marta Kauffman	22.64	8.9	3141
231	10	234	abril	The Last One I	30	Erica gives birth to the baby that Monica and ...	Kevin Bright	Andrew Reich e Ted Cohen	24.51	9.5	6221
232	10	235	maio	The Last One II	30	Phoebe races Ross to the airport in a bid to s...	Kevin Bright	Marta Kauffman e David Crane	52.46	9.7	10381

233 rows × 11 columns

Agora com a base de dados já organizada podemos iniciar a analise exploratória.

5. Análise exploratória

Essa etapa consiste em manipular os dados buscando por padrões ou informações que façam algum sentido e nos ajudem a desenvolver insights sobre os desafios e as possíveis soluções. Inicialmente farei a contagem dos EPs por temporada.

```
## Contando quantos EPs tem cada temporada
main_df['Temporada'].value_counts().sort_index()
```

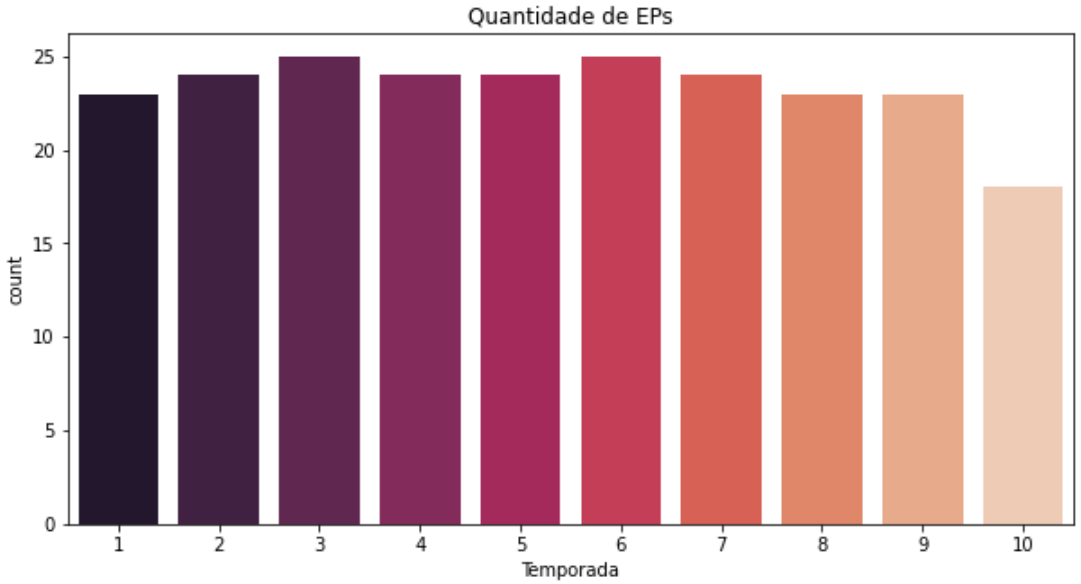
saída:

```
1     23
2     24
3     25
4     24
5     24
6     25
7     24
8     23
9     23
10    18
Name: Temporada, dtype: int64
```

Para melhor visualização irei plotar os dados em gráfico utilizando as bibliotecas matplotlib e seaborn:

```
## importando as bibliotecas
import matplotlib.pyplot as plt
import seaborn as sns

## Ajustando tamanho do gráfico e os títulos
plt.figure(figsize=(10,5))
plt.xlabel("Temporada")
plt.title("Quantidade de EPs")

## criando o gráfico
sns.countplot(x = "Temporada", data = main_df, palette='rocket')
```

Saída:



É possível observar que as temporadas 3 e 6 possuem maior número de EPs, enquanto a ultima temporada foi a que teve a menor quantidade. (Por enquanto esses dados não querem dizer nada, mas a ideia a princípio seria verificar se existe relação entre a quantidade de EPs com os índices de audiência e ver se os dados 'conversam').

Agora veremos a duração das temporadas em minutos

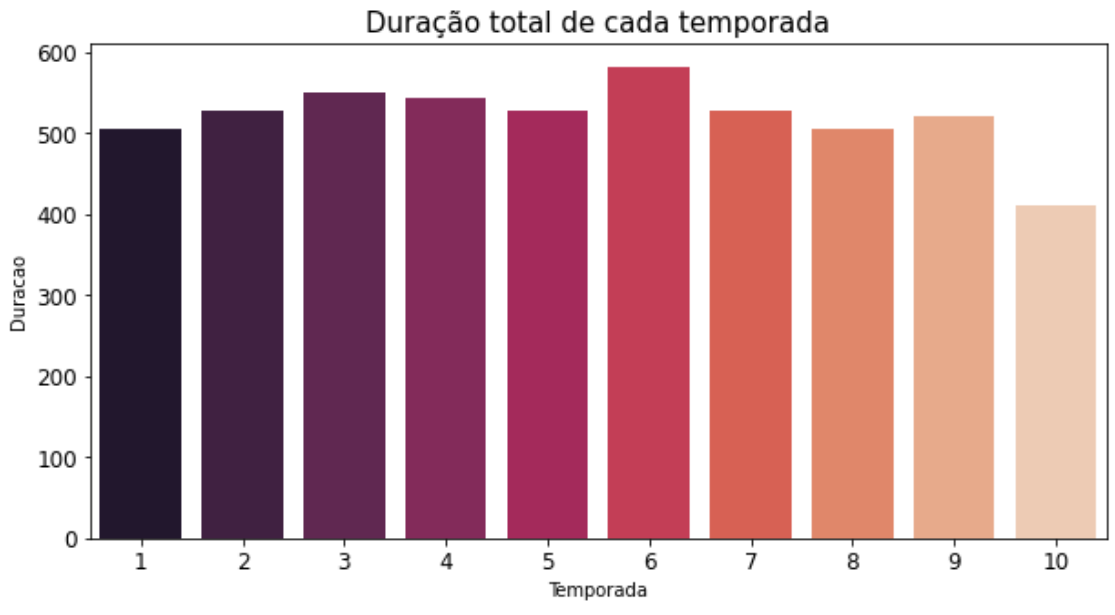
```
duracao_df = main_df.groupby('Temporada').Duracao.sum().to_frame().reset_index()
display(duracao_df)
```

Saída:

	Temporada	Duracao
0	1	506
1	2	528
2	3	550
3	4	544
4	5	528
5	6	582
6	7	528
7	8	506
8	9	522
9	10	412

```
plt.figure(figsize=(10,5))
plt.title('Duração total de cada temporada', fontsize=15)
plt.xlabel('Temporada')
plt.ylabel('Duraçao (minutos)')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
sns.barplot(x=duracao_df.Temporada, y=duracao_df.Duracao, palette='rocket')
```

Saída:



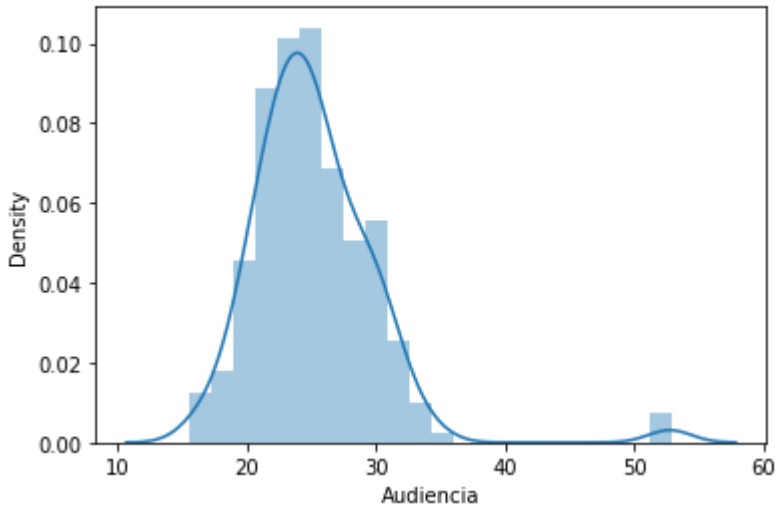
Os dados demonstram que mesmo algumas temporadas possuindo a mesma quantidade de EPs, a duração em minutos pode variar. Além disso, agora sabemos que a temporada 6 possui maior quantidade de EPs e maior duração em relação as demais temporadas, em contraste a temporada 10 possui a menor quantidade de EPs e menor duração em minutos também. Vamos verificar agora a média de audiência que cada temporada obteve.

Primeiramente vamos analisar a distribuição dos dados

```
## como estavam aparecendo muitos avisos, importei essa biblioteca abaixo para
## o código ficar um pouco mais limpo
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

## distribuição dos dados de audiencia
sns.distplot(main_df['Audiencia'])
```

Saída:

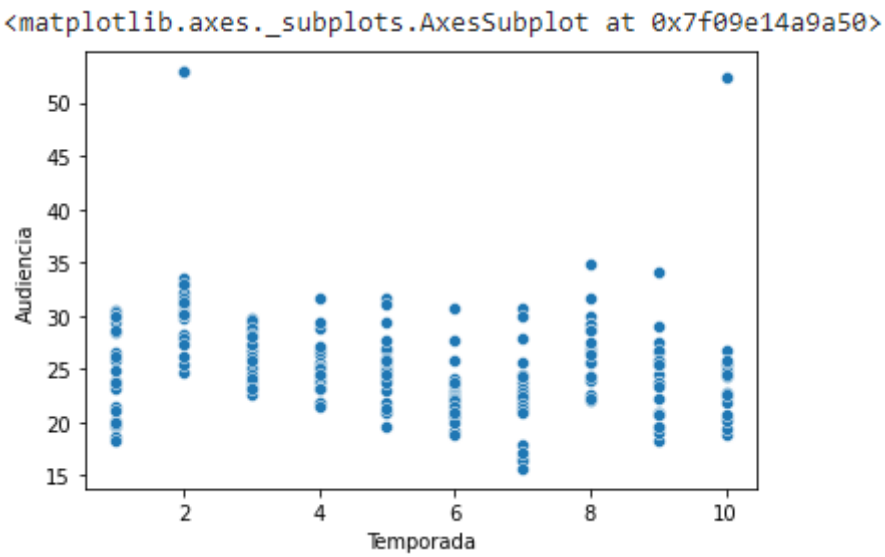
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09d2528390>
```



O gráfico nos mostra que os dados não possuem uma distribuição normal. A sua direita temos dados que talvez sejam possíveis outliers, vamos plotar o gráfico de outra forma para vermos onde exatamente estão esses dados.

```
## podemos utilizar o scatterplot para verificarmos quais os EPs estão 'fora do
## pradrão' e analisarmos se continuamos ou retiramos os dados da analise.
sns.scatterplot(x = 'Temporada', y= 'Audiencia', data=main_df)
```

saída:



O gráfico mostra que existem 2 EPs (3 na verdade) com audiências acima do normal, 2 na segunda temporada e outro na ultima.

- Uma busca rápida pela web foi suficiente para descobrir que o ultimo EP da ultima temporada foi o que teve maior audiência. Em relação aos números de audiência dos EPs da segunda temporada, um dos EPs contou com a participação de um ator que esteve no auge nos anos 90 (Jean Claude Van Dame) e talvez por isso a audiência possa ter sido elevada, já em relação ao outro EP, nada foi encontrado, o que me faz pensar que talvez o dado possa estar errado, e se caso contrário, não há nada que explique o pico de audiência, por isso acho melhor retirar esses dados da analise.

```
##localizando as linhas que iremos retirar
display(main_df.sort_values(by='Audiencia', ascending=False))
```

saída:

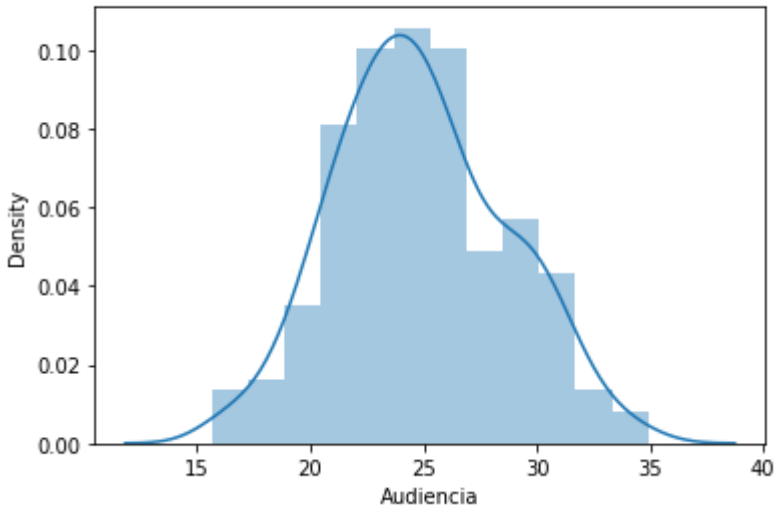
	Temporada	Episodio	Exibicao_orig	Titulo_orig	Duracao	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
36	2	37	janeiro	The One with the Prom Video	22	The gang watches a home video from the night o...	James Burrows	Michael Borkow	52.90	9.4	5736
35	2	36	janeiro	The One After the Superbowl: Part 2	22	Ross finds Marcel on the set of a new movie, w...	Michael Lembeck	Jeffrey Astrof e Mike Sikowitz	52.90	8.8	3864
232	10	235	maio	The Last One II	30	Phoebe races Ross to the airport in a bid to s...	Kevin Bright	Marta Kauffman e David Crane	52.46	9.7	10381
191	8	193	maio	The One Where Rachel Has a Baby: Part 2	22	After Rachel gives birth to her baby, she must...	Kevin Bright	Scott Silveri	34.91	8.9	3150
192	9	195	setembro	The One Where Emma Cries	22	Chandler, having trouble getting enough sleep ...	Sheldon Epps	Sherry Bilsing-Graham e Ellen Plummer	34.01	8.5	2957
...	...	...	...	...	...	...	...	...	...	...	...
167	7	168	maio	The One with Monica and Chandler's Wedding: Pa...	22	So close to the wedding, Chandler suddenly rea...	Kevin Bright	Gregory S. Malins	17.23	8.9	3178
153	7	154	novembro	The One with All the Candy	22	Monica makes candy to get to know her neighbou...	David Schwimmer	Patty Lin	16.57	8.1	2753
164	7	165	abril	The One with Rachel's Big Kiss	22	Rachel's college friend can't remember a scand...	Gary Halvorson	Andrew Reich e Ted Cohen	16.55	8.4	2894
165	7	166	abril	The One with the Vows	22	Monica and Chandler are getting married in fou...	Gary Halvorson	Scott Silveri e Shana Goldberg-Meehan	16.30	7.5	2832
166	7	167	maio	The One with Chandler's Dad	22	Ross and Rachel hit the freeway together when ...	Kevin Bright	Doty Abrams	15.65	8.4	2822

233 rows × 11 columns

```
## excluindo as linhas pelo indice
main_df = main_df.drop(36)
main_df = main_df.drop(35)
main_df = main_df.drop(232)
main_df = main_df.reset_index()

## plot dos dados após a exclusão dos outliers
sns.distplot(main_df['Audiencia'])
```

Saída:

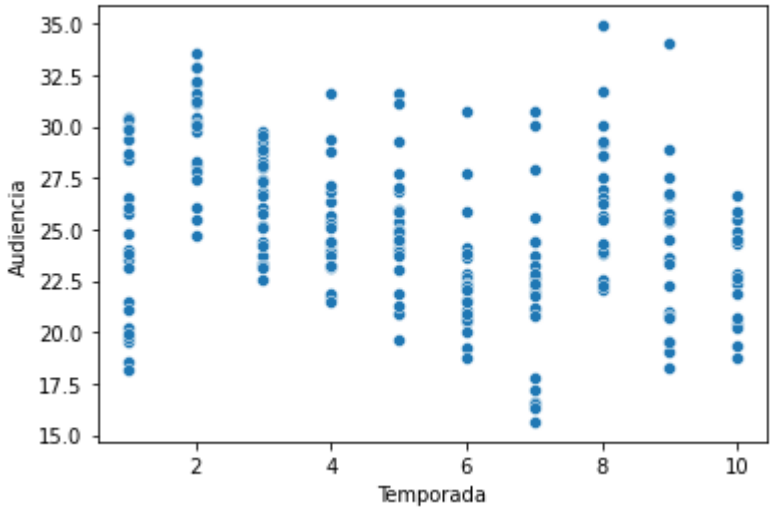
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09d041cb90>
```




```
sns.scatterplot(x = 'Temporada', y= 'Audiencia', data=main_df)
```

saída:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09d03f0dd0>
```

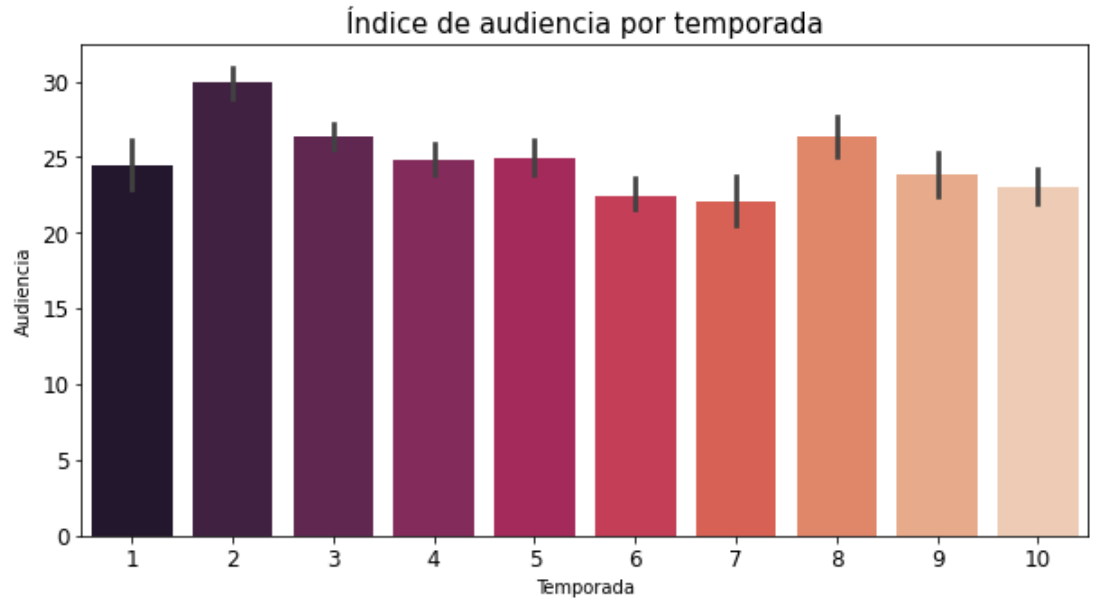


Agora temos dados sem discrepância, não ideais, mas melhores do que estavam antes.

```
## plot em barras dos dados de audiencia por temporada
plt.figure(figsize=(10,5))
plt.title('Índice de audiencia por temporada', fontsize=15)
plt.xlabel('Temporada')
plt.ylabel('Índice de audiencia')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
sns.barplot(x=main_df.Temporada, y=main_df.Audiencia, palette='rocket')
```

saída:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09d0324d50>
```



A imagem acima mostra que a série teve seu pico de audiência na segunda temporada. A partir da terceira os dados mostram uma queda gradativa até a sétima temporada. Na oitava temporada a audiência subiu novamente até o nível de início da sua queda, entretanto o índice voltou a cair nas temporadas seguintes. Talvez averiguar quem estava na direção dos EPs nas temporadas 2 e 8 nos traga alguma informação relevante.

```
## criando DFs filtrando o main_df pelas temporadas 2 e 8
filtrotemp2_df = main_df.loc[main_df['Temporada'] == 2]
filtrotemp8_df = main_df.loc[main_df['Temporada'] == 8]
## agrupando os diretores e contando quantos EPs cada um dirigiu al
ém da média
## de audiencia que tiveram
diretor8_audiencia_df = filtrotemp8_df.groupby("Diretor").Audiencia
.agg(['count','mean']).sort_values(by="mean", ascending=False)
```

```
diretor2_audiencia_df = filtrotemp2_df.groupby("Diretor").Audiencia
.agg(['count','mean']).sort_values(by="mean", ascending=False)

display(diretor2_audiencia_df)
display(diretor8_audiencia_df)
```

Saída:

Diretor	count	mean
Kevin Bright	2	31.350000
Mary Kay Place	1	30.500000
Peter Bonerz	1	30.200000
Thomas Schlamme	2	30.000000
Michael Lembeck	13	29.830769
Gail Mancuso	2	29.250000
Ellen Gittelsohn	1	28.100000

Diretor	count	mean
Sheldon Epps	1	30.040000
Kevin Bright	8	27.512500
Ben Weiss	2	27.085000
David Schwimmer	3	26.596667
Gary Halvorson	9	24.697778

As tabelas mostram que:

- Na segunda temporada o diretor Michael Lembeck foi o que dirigiu mais EPs além de ter uma média 'ok' de audiência com 29,8 pontos. Houve diretores com médias maiores, mas devido ao baixo número amostral apenas ignorei.
- Na temporada 8, dois diretores tiveram destaque, em primeiro lugar Kevin Bright com 8 EPs dirigidos e 27,5 pontos de audiência e em segundo lugar temos o diretor Gary Halvorson que apesar de ter tido a média de audiência mais baixa em relação aos outros diretores, foi o que dirigiu a maior quantidade de EPs totalizando 9 nessa temporada.

Com esses dados temos que, respectivamente os diretores:

1. Michael Lembeck
2. Kevin Bright
3. Gary Halvorson

São por enquanto os mais indicados para dirigirem uma nova temporada da série. Mais a frente faremos essa mesma análise mas em relação a todas as temporadas.

Com os produtores mexeremos depois, pois são muitos, então acho melhor não utilizar de filtros em temporadas e sim pegar o DF completo. Vamos analisar agora os dados do IMDB

```
## Criando outro DF calculando a média de estrelas a Temporada
estrelasxtemp_df = main_df[['Temporada', 'Estrelas_IMDB']].groupby('Temporada', as_index=False).mean()
estrelasxtemp_df = estrelasxtemp_df.sort_values(by='Estrelas_IMDB',ascending=False)

## Criando outro DF dessa vez calculando a somatória dos votos em relação a Temporada
Votos_IMDB = main_df[['Temporada', 'Votos_IMDB']].groupby('Temporada',as_index=False).sum()
Votos_IMDB = Votos_IMDB.sort_values(by='Votos_IMDB', ascending=False)

display(estrelasxtemp_df)
display(Votos_IMDB)
```

saída:

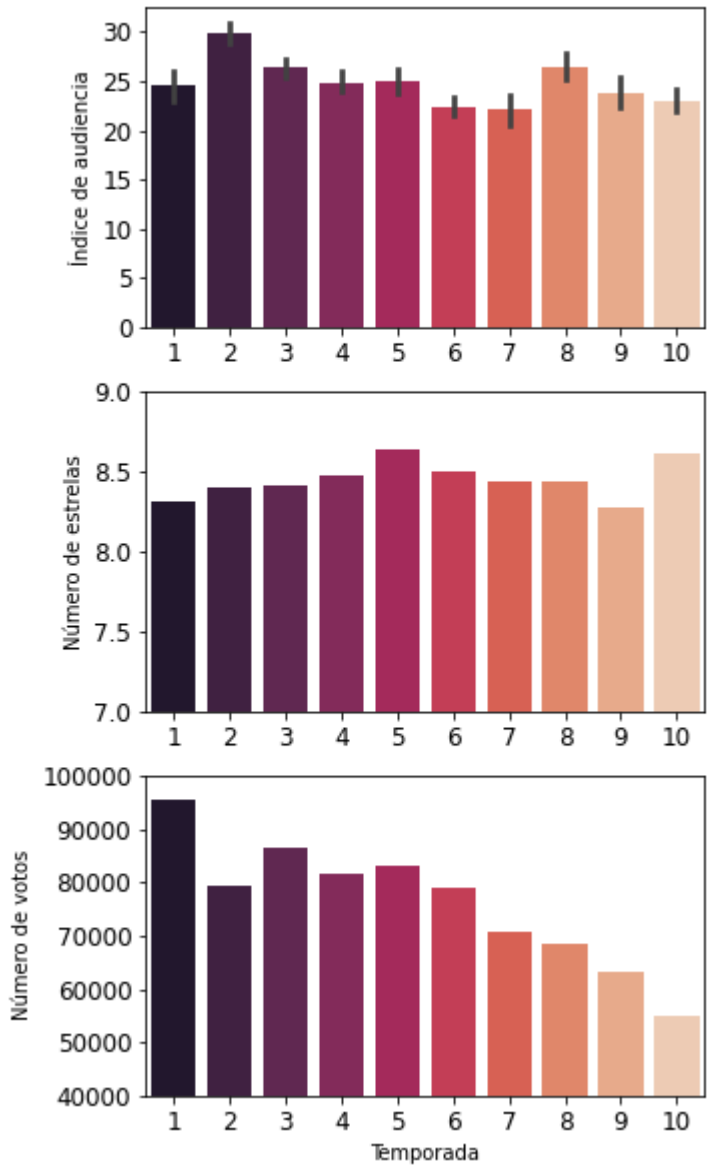
	Temporada	Estrelas_IMDB
4	5	8.637500
9	10	8.617647
5	6	8.496000
3	4	8.475000
6	7	8.437500
7	8	8.434783
2	3	8.408000
1	2	8.400000
0	1	8.317391
8	9	8.278261

	Temporada	Votos_IMDB
0	1	95397
2	3	86462
4	5	82942
3	4	81441
1	2	79414
5	6	78827
6	7	70577
7	8	68604
8	9	63315
9	10	54883

```
## Plotando os gráficos de Audiencia => média de estrelas => quantidade de votos
plt.figure(figsize=(5,10))

plt.subplot(3, 1, 1)
sns.barplot(x=main_df.Temporada, y=main_df.Audiencia, palette='rocket')
plt.ylabel('Índice de audiencia')
plt.xlabel('')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)


plt.subplot(3, 1, 2)
sns.barplot(x=estrelasxtemp_df.Temporada, y=estrelasxtemp_df.Estrelas_IMDB,
            palette='rocket')
plt.ylabel('Número de estrelas')
plt.xlabel('')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylim(7, 9)

plt.subplot(3, 1, 3)
sns.barplot(x=Votos_IMDB.Temporada, y=Votos_IMDB.Votos_IMDB, palette='rocket')
plt.xlabel('Temporada')
plt.ylabel('Número de votos')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.ylim(40000, 100000)
```

Saída:

```
(40000.0, 100000.0)
```



Plotando os gráficos em uma única imagem, facilita a visualização de uma possível correlação.

- Em relação a quantidade de votos X média de estrelas é possível observar que a temporada 1 teve quase 100K de votos entretanto sua média de estrelas no IMDB foi a segunda mais baixa ficando à frente apenas da temporada 9 demonstrando que mesmo não tendo uma audiência tão relevante, aqueles que assistiram parecem não ter gostado da série no primeiro momento.
- A temporada 2 foi a que teve maior audiência na série toda, consequentemente também teve muitos votos, porém, menos que a temporada 1. Por outro lado, mesmo tendo tanta audiência, sua média de estrelas no IMDB não foi tão alta ficando em quinto lugar entre todas as temporadas.

- A temporada 10 por sua vez teve audiência 'ok', mas o que mais chama atenção foi a baixa quantidade de votos X média de estrelas, onde é possível observar o inverso da primeira temporada onde teve muitos votos e uma nota 'baixa', no caso da ultima temporada os telespectadores parecem ter gostado muito pois foram muitas notas altas o que torna a décima temporada a primeira no ranking de estrelas mesmo com número baixo de votos se comparado as temporadas iniciais.

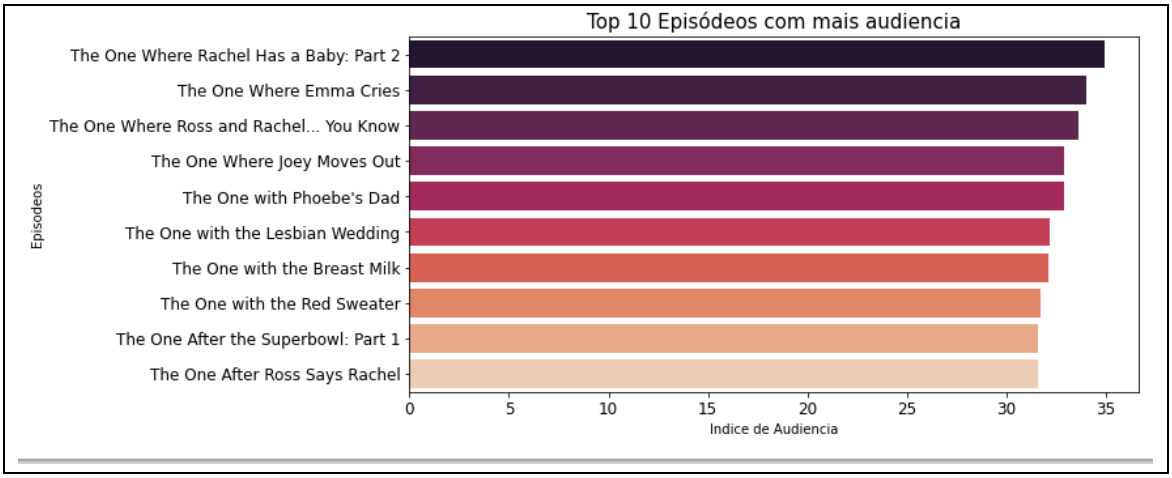
Vamos verificar agora quais os EPs atingiram maior audiência no decorrer da série:

```
## criando um novo DF filtrando pelos 10 EPs com melhores audiencia
melhoreseps_df = main_df[['Titulo_orig', 'Sinopse_orig','Diretor', 'Escrito_por','Audiencia', 'Estrelas_IMDB']].sort_values('Audiencia', ascending=False).head(10).reset_index(drop=True)
display(melhoreseps_df)

## Plotando essas informações em gráfico
plt.figure(figsize=(10,5))
sns.barplot(y=melhoreseps_df.Titulo_orig, x=melhoreseps_df.Audiencia, palette='rocket', orient='h')
plt.title('Top 10 Episódeos com mais audiencia', fontsize=15)
plt.xlabel('Indice de Audiencia')
plt.ylabel('Episodeos')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
```

Saída:

	Titulo_orig	Sinopse_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB
0	The One Where Rachel Has a Baby: Part 2	After Rachel gives birth to her baby, she must...	Kevin Bright	Scott Silveri	34.91	8.9
1	The One Where Emma Cries	Chandler, having trouble getting enough sleep ...	Sheldon Epps	Sherry Bilsing-Graham e Ellen Plummer	34.01	8.5
2	The One Where Ross and Rachel... You Know	Monica becomes infatuated with a friend of her...	Michael Lembeck	Alexa Junge	33.60	8.9
3	The One Where Joey Moves Out	Joey and Chandler's friendship is jeopardized ...	Michael Lembeck	Michael Curtis e Gregory S. Malins	32.90	8.6
4	The One with Phoebe's Dad	Phoebe tracks down her father, but isn't sure ...	Kevin Bright	David Crane e Marta Kauffman	32.90	8.0
5	The One with the Lesbian Wedding	Rachel's mom comes to visit with big news. Mon...	Thomas Schlamme	David Crane e Marta Kauffman	32.20	8.1
6	The One with the Breast Milk	Monica goes shopping with Julie and tries to k...	Michael Lembeck	Jeffrey Astrof e Mike Sikowitz	32.10	8.2
7	The One with the Red Sweater	Monica, Phoebe and Joey advise Rachel to tell ...	David Schwimmer	David Crane e Marta Kauffman	31.70	9.1
8	The One After the Superbowl: Part 1	Ross goes to visit Marcel whilst on a trip to ...	Michael Lembeck	Doty Abrams	31.60	8.6
9	The One After Ross Says Rachel	A humiliated Emily runs away after Ross says R...	Kevin Bright	Jill Condon e Amy Toomin	31.60	8.9

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 <a list of 10 Text major ticklabel objects>)
```



A tabela mostra alguns dados interessantes onde vemos mais uma vez nomes de Diretores se repetindo Michael Lembeck e Kevin Bright, no Top10 Eps com melhor audiência, Kevin aparece 3 vezes enquanto Michael aparece 4 vezes trazendo fortes indícios de que esses diretores seriam os mais indicados para a direção de uma 11ª temporada.

```
melhoreseps_df.describe()
```

Saída:

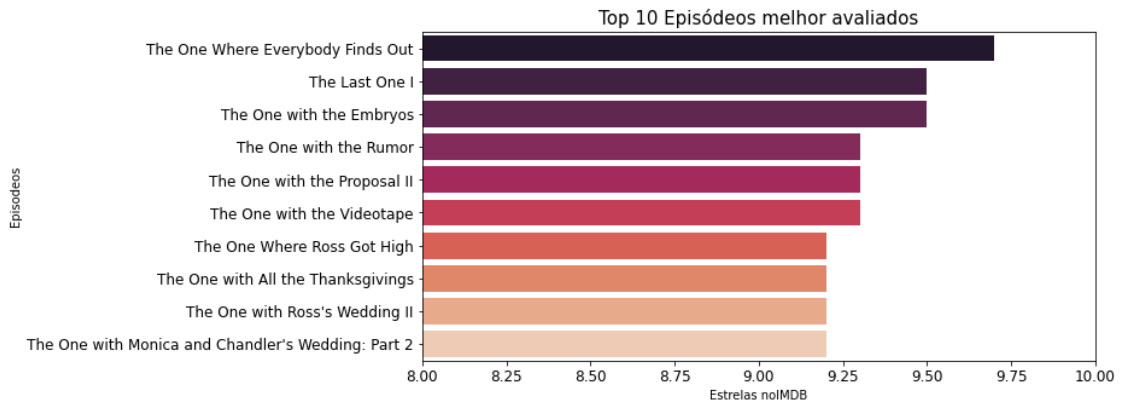
	Audiencia	Estrelas_IMDB
count	10.000000	10.000000
mean	32.752000	8.580000
std	1.130052	0.379473
min	31.600000	8.000000
25%	31.800000	8.275000
50%	32.550000	8.600000
75%	33.425000	8.900000
max	34.910000	9.100000

Sabendo que a média de audiência de toda a série é de 25.19 e a do TOP10 audiência é 32,75. Temos que conseguir um modelo que esteja sempre mais próximo de 30. Por hora é isso que pode ser observado, a seguir podemos analisar quais EPs obtiveram as melhores notas.

```
epsmelhorrankingtop10_df = main_df[['Titulo_orig', 'Sinopse_orig','Diretor', 'Escrito_por','Estrelas_IMDB']].sort_values('Estrelas_IMDB', ascending=False).head(10).reset_index(drop=True)

## plot com o TOP10 EPs com melhores notas no IMDB
plt.figure(figsize=(10,5))
sns.barplot(y=epsmelhorrankingtop10_df.Titulo_orig, x=epsmelhorrankingtop10_df.Estrelas_IMDB, palette='rocket', orient='h')
plt.title('Top 10 Episódeos melhor avaliados', fontsize=15)
plt.xlabel('Estrelas noIMDB')
plt.ylabel('Episodeos')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.xlim(8, 10)
```

Saída:

```
(8.0, 10.0)
```



Olhando os 2 gráficos (EPs com maior audiencia e Eps melhor avaliados) é possível notar que índice de audiência e notas do IMBD não possuem nenhuma correlação, uma vez que nenhum EP se repete em algum dos gráficos, são completamente distintos. Vamos tentar buscar alguma informação agora em relação à direção dos Eps.

```
## Criando outro DF para analisarmos a nota dos melhores EPs e seus respectivos Diretores e Escritores.
epsmelhorranking_df = main_df[['Titulo_orig', 'Sinopse_orig','Diretor', 'Escrito_por','Estrelas_IMDB']].sort_values('Estrelas_IMDB', ascending=False).head(20).reset_index(drop=True)
display(epsmelhorranking_df)
```

Saída:

	Titulo_orig	Sinopse_orig	Diretor	Escrito_por	Estrelas_IMDB
0	The One Where Everybody Finds Out	When Phoebe finds out about Monica and Chandle...	Michael Lembeck	Michael Curtis e Seth Kurland	9.7
1	The Last One I	Erica gives birth to the baby that Monica and ...	Kevin Bright	Andrew Reich e Ted Cohen	9.5
2	The One with the Embryos	Phoebe's uterus is examined for implantation o...	Kevin Bright	Seth Kurland	9.5
3	The One with the Rumor	Monica invites Will, an old school friend of h...	Gary Halvorson	Andrew Reich e Ted Cohen	9.3
4	The One with the Proposal II	Chandler continues to pretend to hate the idea...	Kevin Bright	Shana Goldberg-Meehan e Scott Silveri	9.3
5	The One with the Videotape	Ross and Rachel tell everyone about the night ...	Kevin Bright	Sherry Bilsing e Ellen Plummer	9.3
6	The One Where Ross Got High	Ross is forced to reveal the reason why Jack a...	Kevin Bright	Andrew Reich e Ted Cohen e Perry Rein e Gigi M...	9.2
7	The One with All the Thanksgivings	The gang remember and share with each other th...	Kevin Bright	Gigi McCreery e Perry Rein	9.2
8	The One with Ross's Wedding II	Phoebe tries to warn the gang that Rachel is c...	Kevin Bright	Michael Borkow	9.2
9	The One with Monica and Chandler's Wedding: Pa...	Ross tries to find Chandler with Phoebe's help...	Kevin Bright	Gregory S. Malins	9.2
10	The One with the Jellyfish	Ross chooses between Rachel and his bald-heade...	Shelley Jensen	Pang-Ni Landrum e Mark Kunerth	9.1
11	The One with All the Resolutions	The gang make their New Years resolutions. Cha...	Joe Regalbuto	Shana Goldberg-Meehan	9.1
12	The One with Unagi	Rachel and Phoebe take self-defense classes. C...	Gary Halvorson	David Crane e Marta Kauffman	9.1
13	The One in Vegas: Part 2	Chandler and Monica reconcile and hastily deci...	Kevin Bright	Ted Cohen e Andrew Reich	9.1
14	The One with the Red Sweater	Monica, Phoebe and Joey advise Rachel to tell ...	David Schwimmer	David Crane e Marta Kauffman	9.1
15	The One with the Morning After	Ross is guilt-ridden after sleeping with Chloe...	James Burrows	Michael Borkow	9.1
16	The One with Ross's Sandwich	Joey finds himself constantly covering for Cha...	Gary Halvorson	Gregory S. Malins	9.1
17	The One with Chandler in a Box	Chandler tries to earn Joey's forgiveness by l...	Peter Bonerz	Adam Chase	9.1
18	The One with the Flashback	The gang remember the events three years ago, ...	Peter Bonerz	Shana Goldberg-Meehan e Scott Silveri	9.1
19	The One with All the Kissing	Chandler must constantly kiss the girls to cov...	Gary Halvorson	Seth Kurland	9.0

Na tabela acima é possível notar que o nome do diretor Kevin Bright aparece diversas vezes (45%), coincidência que ele apareça tanto nos EPs com maiores notas? Talvez, mas essa é uma coincidência que o público gostou e um ponto muito importante a se guardar. Em relação aos escritores, a princípio não parecem que se repetem muito. É preciso averiguar

```
## Contagem de quantas vezes os escritores aparecem nesse top20
escritor_audiencia_df = epsmelhorranking_df.groupby("Escrito_por").Estrelas_IMDB.agg(['count','mean']).sort_values(by="mean", ascending=False)
display(escritor_audiencia_df)
```

Saída:

Escrito_por	count	mean
Michael Curtis e Seth Kurland	1	9.70
Andrew Reich e Ted Cohen	2	9.40
Sherry Bilsing e Ellen Plummer	1	9.30
Seth Kurland	2	9.25
Andrew Reich e Ted Cohen e Perry Rein e Gigi McCreery	1	9.20
Gigi McCreery e Perry Rein	1	9.20
Shana Goldberg-Meehan e Scott Silveri	2	9.20
Gregory S. Malins	2	9.15
Michael Borkow	2	9.15
Adam Chase	1	9.10
David Crane e Marta Kauffman	2	9.10
Pang-Ni Landrum e Mark Kunerth	1	9.10
Shana Goldberg-Meehan	1	9.10
Ted Cohen e Andrew Reich	1	9.10

Não acho que dê pra considerar essa quantidade de amostras dos escritores para vermos quais tiveram as melhores notas, então vou buscar no DF completo e calcular a média para termos dados mais legais.

```
## Contagem de quantas vezes os escritores aparecem durante a série completa
escritor_audiencia_df = main_df.groupby("Escrito_por").Estrelas_IMDB.agg(['count','mean']).sort_values(by="count", ascending=False)

display(escritor_audiencia_df.head(10))
```

Saída:

Escrito_por	count	mean
Andrew Reich e Ted Cohen	12	8.525000
David Crane e Marta Kauffman	11	8.463636
Alexa Junge	11	8.381818
Shana Goldberg-Meehan	10	8.490000
Seth Kurland	10	8.470000
Doty Abrams	9	8.355556
Sherry Bilsing-Graham e Ellen Plummer	8	8.462500
Scott Silveri	8	8.325000
Wil Calhoun	7	8.457143
Dana Klein Borkow	7	8.300000

De acordo com a tabela gerada o top 5 escritores com maior quantidade de EPs escritos (e que irei considerar) foram:

- Andrew Reich e Ted Cohen 8.525000
- David Crane e Marta Kauffman 8.463636
- Alexa Junge 8.381818
- Shana Goldberg-Meehan 8.490000
- Seth Kurland 8.470000

Dessa lista, os Escritores que apareceram no TOP20 anterior dos EPs mais bem avaliados foram:

- Andrew Reich e Ted Cohen 3 aparições
- Seth Kurland 3 aparições
- David Crane e Marta Kauffman 2 aparições
- Shana Goldberg-Meehan 2 aparições

Então, diria que esses escritores seriam os mais indicados para uma próxima temporada.

Vamos agora analisar o conteúdo de cada EP desses melhores avaliados e tentar extrair alguma informação sobre quais os temas o EP abordava e tentar encontrar algo que seja relevante.

```
## para tentarmos identificar os principais temas dos EPs, podemos usar uma
## wordcloud, assim, as palavras que mais forem mencionadas nas sinopses
## irão se destacar e podemos analisar sobre o que acontece nos EPs mais
## bem avaliados

## importando a biblioteca para criar a wordcloud
from wordcloud import WordCloud, STOPWORDS

stopwords = set(STOPWORDS)

## criarei uma variável para não contar os nomes dos personagens algumas palavras que não
## são relevantes e podermos focar nos temas em si.
retirar = {"Rachel","Ross","Monica","Phoebe","Joey","Chandler", 'Richard',
        'Meanwhile', 'constantly'}
stopwords.update(retirar)
wordcloud = WordCloud(background_color='black',
                    stopwords=stopwords,
                    max_words=200,
                    max_font_size=40,
                    random_state=42
                    ).generate("".join(epsmelhorranking_df['Sinopse_orig']))

print(wordcloud)
plt.figure(figsize=(10,5))
fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

Saída:

```
<wordcloud.wordcloud.WordCloud object at 0x7f09d2523e90>
```

A imagem acima mostra as palavras que mais foram citadas nas sinopses dos EPs com maiores notas, o que pode ser de grande ajuda na hora de se escrever um novo roteiro, por exemplo, casamento, tentativas, juntos, amigos, dia de ação de graças, relacionamento, 'caso'. Todas essas palavras podem indicar que histórias voltadas para um tema envolvendo romances ou relacionamentos podem agradar mais ao público do que histórias com mais ação.

Um ponto que poderia ser abordado são que estudos na área de psicologia e neurociência apontam que os seres humanos são bem susceptíveis a se interessar por assuntos que servem de base para a sua sobrevivência, Steven Pinker cita isso em seu livro 'O novo iluminismo: Em defesa da razão, da ciência e do humanismo', apontando como exemplo a sexualidade humana e as formas que fazemos muitas coisas sem perceber, por instinto de reprodução, como nos vestir bem (aumentando suas chances de atração), praticar exercícios (se mantendo saudável para atrair um parceiro), estudarmos (exercitando a mente para desenvolver pensamentos e poder adaptar a situações para atrair um parceiro), tudo isso de certa forma tem como objetivo atrair um parceiro para um possível relacionamento e a partir disso gerar um descendente mantendo a evolução da espécie. Ou seja, temas envolvendo relacionamentos são mais suscetíveis a gerar certo nível de interesse do público mesmo que de uma maneira inconsciente. Talvez um seja interessante um integrante no time com formação em psicologia ou neurociência. Mas voltando a análise. . .

Outra coluna que podemos talvez tirar alguma informação relevante seria a com as datas que os EPs foram apresentados, uma vez que o período pode influenciar na audiência, pois existem épocas que as pessoas ficam mais em casa (inverno), ou estão viajando como nas férias de verão (nos EUA).

```
## criando um novo DF com as médias de audiencia pelo mês exibido
data_df = main_df.groupby('Exibicao_orig').Audiencia.mean().to_frame().reset_index().sort_values(
by='Audiencia', ascending=False)
data_df
```

Saída:

	Exibicao_orig	Audiencia
8	setembro	27.573077
3	janeiro	26.242400
4	maio	25.576897
2	fevereiro	25.330000
5	março	25.000000
6	novembro	24.730000
7	outubro	24.303158
1	dezembro	23.382143
0	abril	22.199231

Com a tabela acima, temos que o mês de setembro é o mês que tem a maior média de audiência. Setembro é o mês em que se tem início o ano letivo nos EUA, quando muitas famílias voltam de viagens de férias, talvez seja o 'março brasileiro' que é quando se passam as festas de fim de ano e carnaval e a rotina das pessoas volta a sua normalidade. Em janeiro tivemos a audiência um pouco acima da média também, o que talvez possa ser explicado pelo recesso de fim de ano para comemoração dos feriados de natal e ano novo, com isso pode ser que as famílias se reúnam e fiquem mais em casa. Diante desses dados, temos que os meses com maior audiência seriam setembro e janeiro, e os piores, abril e dezembro.

Agora iremos analisar se existe alguma correlação entre as colunas. Para isso vou criar um novo DF apenas com os dados que mais relevantes.

```
## Criando um novo DF para verificar se existe correlação entre as colunas abaixo.
correlate_df  = main_df[['Exibicao_orig','Diretor', 'Escrito_por', 'Audiencia', 'Estrelas_IMDB', 'Votos
_IMDB']]
correlate_df
```

Saída:

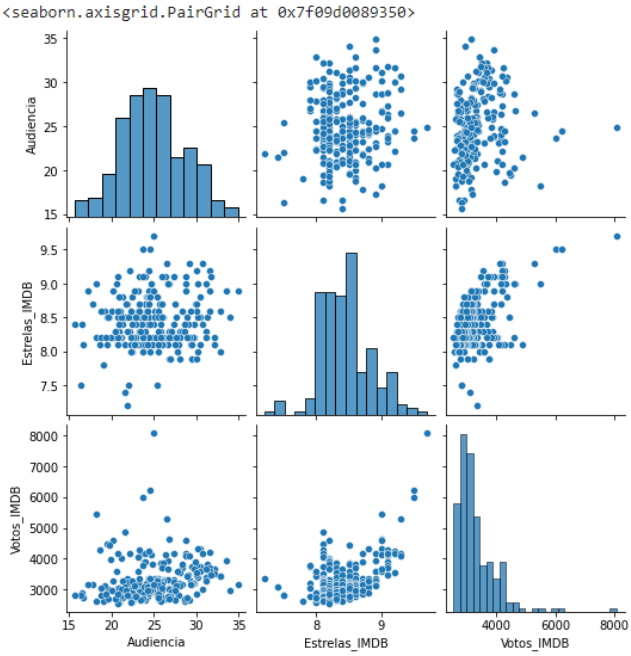
	Exibicao_orig	Diretor	Escrito_por	Audiencia	Estrelas_IMDB	Votos_IMDB
0	setembro	James Burrows	David Crane e Marta Kauffman	21.50	8.1	4888
1	setembro	James Burrows	David Crane e Marta Kauffman	20.20	8.2	4605
2	outubro	James Burrows	Jeffrey Astrof e Mike Sikowitz	19.50	8.1	4468
3	outubro	Pamela Fryman	Alexa Junge	19.70	8.5	4438
4	outubro	Arlene Sanford	Jeff Greenstein e Jeff Strauss	18.60	8.1	4274
...	...	...	...	...	...	...
225	fevereiro	Gary Halvorson	Robert Carlock e Dana Klein Borkow	25.90	8.5	3044
226	fevereiro	Gary Halvorson	Sherry Bilsing-Graham e Ellen Plummer	24.27	8.6	2989
227	fevereiro	Gary Halvorson	Robert Carlock e Tracy Reilly	22.83	8.5	2771
228	abril	Gary Halvorson	Mark Kunerth e David Crane e Marta Kauffman	22.64	8.9	3141
229	abril	Kevin Bright	Andrew Reich e Ted Cohen	24.51	9.5	6221

230 rows × 6 columns

```
## O pairplot serve exatamente para analisarmos algum padrão nos gr
áficos que
## correlacionem os dados.
sns.pairplot(correlate_df)
```

Saída:

```
<seaborn.axisgrid.PairGrid at 0x7f09d0089350>
```

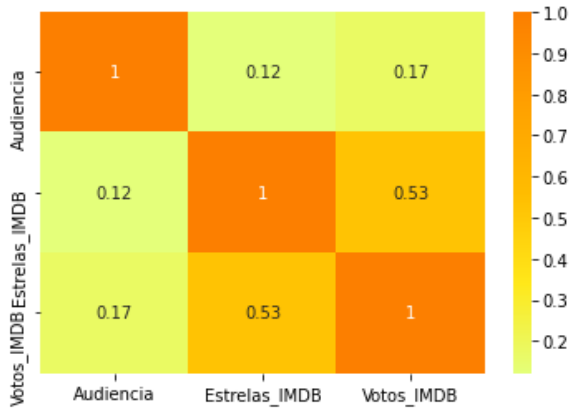


A princípio os dados não parecem ter alguma correlação. Vamos dar uma olhada em outro tipo de gráfico para confirmarmos.

```
## o Heatmap é um grafico de 'calor' que mostra em números e cores qual a
## relação que uma coluna possui com a outra.
sns.heatmap(correlate_df.corr(), cmap='Wistia', annot=True)
```

Saída:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09c53b3c90>
```



Os dados que mais se aproximam são os de votos e estrelas do IMDB, o que faz certo sentido.

6. Algorotímos de machine learning

Nessa etapa iremos selecionar os dados que iremos usar para o treinamento preditivo, após selecionados, é feita a escolha do algoritmo que mais se encaixa para o problema.

Para realizar a predição da audiência vou considerar as variáveis de tempo (mês), diretor e escritor uma vez que os votos e a nota do IMDB surgem apenas depois que o EP é lançado e visto pelo público. Temos então 3 variáveis que juntas irão prever os dados numéricos dos índices de audiência, então usaremos um modelo de arvore de decisão e regressão, nesse caso testaremos os algoritmos randomforest e extratree.

```
## Criando um novo DF apenas com os dados que serão utilizados para o
## treinamento do algoritimo
train_df = correlate_df[['Exibicao_orig','Diretor', 'Escrito_por', 'Audiencia']]
## exportarei uma cópia dessa base para podermos localizar os nomes,
## ficará mais claro o motivo a seguir
train_df.to_csv('plan-string.csv', sep=';')
train_df
```

Saída:

	Exibicao_orig	Diretor	Escrito_por	Audiencia
0	setembro	James Burrows	David Crane e Marta Kauffman	21.50
1	setembro	James Burrows	David Crane e Marta Kauffman	20.20
2	outubro	James Burrows	Jeffrey Astrof e Mike Sikowitz	19.50
3	outubro	Pamela Fryman	Alexa Junge	19.70
4	outubro	Arlene Sanford	Jeff Greenstein e Jeff Strauss	18.60
...	...	...	...	...
225	fevereiro	Gary Halvorson	Robert Carlock e Dana Klein Borkow	25.90
226	fevereiro	Gary Halvorson	Sherry Bilsing-Graham e Ellen Plummer	24.27
227	fevereiro	Gary Halvorson	Robert Carlock e Tracy Reilly	22.83
228	abril	Gary Halvorson	Mark Kunerth e David Crane e Marta Kauffman	22.64
229	abril	Kevin Bright	Andrew Reich e Ted Cohen	24.51

230 rows × 4 columns

Como temos dados em string, vamos utilizar uma nova função para converter esses dados em valores numéricos para que nosso algoritmo consiga 'entender' os valores.

```
## importando a função preprocessing da biblioteca sklearn
from sklearn import preprocessing

## convertendo os valores de string para numéricos
label_encoder = preprocessing.LabelEncoder()
train_df['Exibicao_orig'] = label_encoder.fit_transform(train_df['Exibicao_orig'])
train_df['Diretor'] = label_encoder.fit_transform(train_df['Diretor'])
train_df['Escrito_por'] = label_encoder.fit_transform(train_df['Escrito_por'])

## agora exportarei a mesma base de dados, mas agora com as strings convertidas
## assim podemos abrir as duas bases de dados e ver quem é quem para
## fazer predições
train_df.to_csv('plan_convert.csv', sep=';')
train_df
```

Saída:

	Exibicao_orig	Diretor	Escrito_por	Audiencia
0	8	10	22	21.50
1	8	10	22	20.20
2	7	10	38	19.50
3	7	15	3	19.70
4	7	2	36	18.60
...	...	...	...	...
225	2	9	59	25.90
226	2	9	74	24.27
227	2	9	60	22.83
228	0	9	42	22.64
229	0	12	8	24.51

230 rows × 4 columns

Agora iremos separar os nossos dados de treino do algoritmo e de teste.

```
## importando a função de teste
from sklearn.model_selection import train_test_split

## separando os dados
x = train_df.drop('Audiencia', axis=1)
y = train_df['Audiencia']

## neste caso o algorito irá treinar com 80% dos dados e os 20% restante
## ele usará como teste para as predições
x_treino, x_teste, y_treino, y_teste = train_test_split(x,y, test_size=0.20)
```

Após a separação, precisamos treinar o algoritmo com os dados para posteriormente fazermos alguma previsão.

```
## importando as bibliotecas dos algoritimos que serão utilizados
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
randomforest = RandomForestRegressor()
extratree = ExtraTreesRegressor()
## treinando o algoritimo
randomforest.fit(x_treino, y_treino)
extratree.fit(x_treino, y_treino)
```

Após treinar o algoritmo, precisamos medir métricas que ele possui:

```
from sklearn import metrics
teste_random = randomforest.predict(x_teste)
teste_extra = extratree.predict(x_teste)

r2_random = metrics.r2_score(y_teste, teste_random)
r2_extra = metrics.r2_score(y_teste, teste_extra)

print('R²')
print(f'randomforest = {r2_random:.2f}')
print(f'extratree    = {r2_extra:.2f}')
print('='*20)

erro_random = metrics.mean_squared_error(y_teste, teste_random)
erro_extratree = metrics.mean_squared_error(y_teste, teste_extra)

print('erro')
print(f'randomforest = {erro_random:.2f}')
print(f'extratree    = {erro_extratree:.2f}')
print('='*20)

## Esse comando serve para ver qual o peso de cada variável, nesse caso podemos
## ver que o algoritimo da um peso maior para os escritores.
print(randomforest.feature_importances_)
```

Saída:

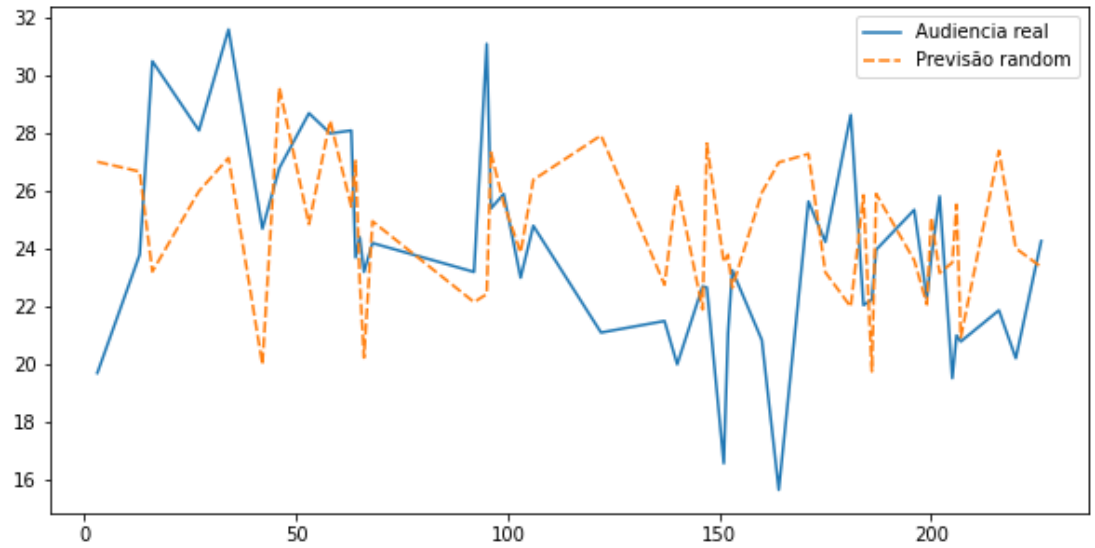
```
R²
randomforest = -0.29
extratree    = -0.48
====================
erro
randomforest = 14.94
extratree    = 17.12
====================
[0.26575667 0.28587243 0.44837089]
```

Neste caso temos um R^2 negativo, o que indica que nenhuns dos algoritimos se adequaram muito bem aos dados, por outro lado o algoritmo randomforest tem uma taxa de erro menor que o extratree, então vamos utilizar o randomforest para realizar as predições.

```
## criando outro DF para adicionar os valores testados nas previsões.
tabela_comp = pd.DataFrame()
tabela_comp['Audiencia real'] = y_teste
tabela_comp['Previsão random'] = teste_extra

## plot com as informações
plt.figure(figsize=(10,5))
sns.lineplot(data=tabela_comp)
```

Saída:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09c01d3310>
```



O gráfico acima relaciona os dados de audiência reais e os dados que o algoritmo tentou prever. Apesar de não parecer muito eficiente, cabe-se ressaltar que dependendo dos dados fica difícil realizar previsões, sendo necessário às vezes adicionar mais variáveis para uma melhor precisão. Nesse caso só não utilizamos as palavras chave dos EPs, o que talvez pudesse ajudar para uma maior precisão das previsões. Por outro lado no decorrer da análise exploratória encontramos informações relevantes acerca de audiência e notas dos EPs além de quem dirigiu e escreveu as histórias. Assim podemos testar algumas combinações e ver qual seria a melhor para obtermos uma melhor audiência para uma nova temporada.

Como já visto no decorrer da analise, os melhores diretores foram:

- Michael Lembeck (14)
- Kevin Bright (12)
- Gary Halvorson (9)

Os melhores escritores:

- Andrew Reich e Ted Cohen (8)
- David Crane e Marta Kauffman (22)
- Alexa Junge (3)
- Shana Goldberg-Meehan (69)
- Seth Kurland (67)

As melhores épocas para audiência:

- Setembro (8)
- Janeiro (3)

*Os números entre parênteses representam os nomes depois de converter as strings em valores numéricos.

Agora podemos testar todas as possibilidades entre diretores e escritores e ver qual seriam as combinações que dariam maior audiência.

```
## Podemos adicionar os números de seus respectivos diretores e escritores, e
## assim, verificar qual seria a previsão de audiencia que a combinação teria
## para a data vou deixar como padrão "setembro", pois é quando as temporadas
## tem seu início
print('Michael')
print(randomforest.predict([['8','14', '8']]))
print(randomforest.predict([['8','14', '22']]))
print(randomforest.predict([['8','14', '3']]))
print(randomforest.predict([['8','14', '69']]))
print(randomforest.predict([['8','14', '67']]))
print('====================')
print('Kevin')
print(randomforest.predict([['8','12', '8']]))
print(randomforest.predict([['8','12', '22']]))
print(randomforest.predict([['8','12', '3']]))
print(randomforest.predict([['8','12', '69']]))
print(randomforest.predict([['8','12', '67']]))
print('====================')
print('Gary')
print(randomforest.predict([['8','9', '8']]))
print(randomforest.predict([['8','9', '22']]))
print(randomforest.predict([['8','9', '3']]))
print(randomforest.predict([['8','9', '69']]))
print(randomforest.predict([['8','9', '67']]))
```

Saída:

```
Michael
[28.181625]
[30.5122]
[29.4662]
[29.41938333]
[29.06301667]
====================
Kevin
[26.58973333]
[28.2267]
[27.98694167]
[27.53403333]
[26.87376667]
====================
Gary
[26.9567]
[25.74098333]
[26.2795]
[23.1557]
[22.84755]
```

7. Interpretação dos Resultados

Esses dados indicam que o diretor Michael Lembeck seria a melhor opção junto aos escritores David Crane e Marta Kauffman, temos uma previsão de 30.50 pontos de audiência. Outro ponto importante seria o fato de apenas o diretor Gary Halvorson ter previsões que ficariam abaixo da média da série (25.19). Por outro lado, mesmo tendo as audiências mais baixas, uma nova temporada ainda ficaria entre o TOP 5 programas com maior audiência dos EUA, pois segundo a tabela gerada em 2018 pela Advertising Ageda a partir da base de dados da Nielsen, os programas com maior audiência foram os de esporte:

1. Super Bowl LII (NBC) 43.1
2. State of the Union Address (multiple networks) 26.9
3. AFC Championship Game (CBS) 24.3
4. NFC Championship Game (Fox) 21.7
5. NFC Divisional Playoff (Fox) 19.3

Com isso, podemos concluir que uma nova temporada seria muito bem vinda, pois manteria a audiência para a emissora e seguindo as métricas corretas como diretores e escritores com experiências anteriores na série, além de um roteiro com temas focados em relacionamentos, e também participações especiais com atores que estivessem na mídia poderiam aumentar ainda mais a audiência gerando mais retorno para a empresa.

8. Perguntas do desafio

- Quais os padrões, tendências e principais características podemos observar nestes dados?
  R: Os dados brutos vieram com alguns erros de digitação e algumas colunas não possuíam um padrão de escrita. Em relação as analises foi possível observar que o início das temporadas sempre tem mais audiência (mês de setembro) e que vai caindo com o passar dos meses tendo seus piores índices de audiência sempre no mês de abril, quando volta a subir ao final da temporada (mês de maio).
  Com o passar do tempo a série foi perdendo engajamento pois a partir da temporada 5, a quantidade de votos no IMDB foi diminuindo gradativamente a cada temporada.
  Alguns diretores e escritores tiveram seus nomes listados diversas vezes nos EPs com maiores notas e audiências.
  Temas envolvendo relacionamentos afetivos foram os que tiveram as maiores avaliações no IMDB.

- A sinopse, escritores e/ou diretor dos episódios influenciam na audiência?
  R: Sim, embora estatisticamente não encontrarmos nenhuma correlação entre direção X audiência ou escritor X audiência, alguns nomes de diretores assim como os escritores apareceram diversas vezes nas analises enquanto buscávamos informações sobre os índices de audiência. Já para as sinopses, parece que os episódios envolvendo temas sobre relacionamentos afetivos tiveram certa atenção do público.

- Usando uma técnica de previsão, qual seria a audiência de uma nova temporada?
  R: Utilizando a melhor combinação entre data, direção e escritores a previsão seria de 30.51 pontos.

- Imaginando que você poderia pedir para os produtores da série outras informações, quais dados você acredita que enriqueceriam a análise e poderiam auxiliar na previsão?
  R: É sempre bom conhecer o público de interesse, então acredito que informações sobre faixa etária e dados demográficos seriam bem vindos além de dados sobre orçamento e faturamento seriam interessantes também para saber se valeria a pena à produção de uma nova temporada.