

✓ Importando as bibliotecas

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import random
import matplotlib.pyplot as plt

# Código para quem não tem endereço de email do Inteli, só colocar o link e id do arquivo

#import gdown
#arquivo_destino_colab = "dataset.csv"

# https://drive.google.com/drive/folders/11gFZUmhu6Arzk_Iqq0D9lRX9Re6bIzVL?usp=sharing
#doc_id = "11gFZUmhu6Arzk_Iqq0D9lRX9Re6bIzVL"

#URL = f"https://drive.google.com/uc?id={doc_id}"

#gdown.download(URL, arquivo_destino_colab, quiet=False)

#df = pd.read_csv(f"/content/{arquivo_destino_colab}", sep=";")

#df.head()

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

✓ Importação das Tabelas de 2023 e 2024

```
ga_universal_2019a2022 = pd.read_csv('/content/drive/Shareddrives/M10_SI/ga_universal_2019a2022.csv')
acesso_eventos_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/acessos_eventos_2023.csv')
acessos_fontes_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/acessos_fontes_2024.csv')
age_sessions_big_numbers_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/age_sessions_big_numbers_2024.csv')
hour_users_big_numbers_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/hour_users_big_numbers_2024.csv')
week_users_big_numbers_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/week_users_big_numbers_2024.csv')
google_ads_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/google_ads_2023.csv')
country_city_views_big_numbers_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/country_city_views_big_numbers_2023.csv')
country_city_views_big_numbers_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/country_city_views_big_numbers_2024.csv')
ga_universal_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/ga_universal_2023.csv')
acessos_fontes_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/acessos_fontes_2023.csv')
age_sessions_big_numbers_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/age_sessions_big_numbers_2023.csv')
hour_users_big_numbers_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/hour_users_big_numbers_2023.csv')
week_users_big_numbers_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/week_users_big_numbers_2023.csv')
acessos_geral_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/acessos_geral_2023.csv')
# acessos_geral_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/acessos_geral_2024.csv')
users_country_big_numbers_geo_2023 = pd.read_csv('/content/drive/Shareddrives/M10_SI/users_country_big_numbers_geo_2023.csv')
users_country_big_numbers_geo_2024 = pd.read_csv('/content/drive/Shareddrives/M10_SI/users_country_big_numbers_geo_2024.csv')
```

✓ Análise Estatística

A estatística é a base que nos permite interpretar os dados de amostras coletadas e realizar pesquisas. Para produzirmos uma análise estatística, devemos inicialmente descrever a natureza dos dados a serem analisados.

Neste contexto, aqui esta um exemplo da inferência estatística que estamos fazendo nas bases de dados. Importante citar que depois complementaremos com informações mais direcionadas

```
# Country City Views Big Numbers 2024
print("\nEstatísticas Básicas - Country City Views Big Numbers 2024")
print(country_city_views_big_numbers_2024.describe())
print("\nContagem de Valores Únicos - Country City Views Big Numbers 2024")
print(country_city_views_big_numbers_2024.nunique())
print("\nAnálise de Valores Nulos - Country City Views Big Numbers 2024")
print(country_city_views_big_numbers_2024.isnull().sum())
```

```
Estatísticas Básicas - Country City Views Big Numbers 2024
Views
count    1998.000000
mean      418.041041
std       3148.220261
```

```
min      0.000000
25%     1.000000
50%     54.000000
75%     264.000000
max     102347.000000

Contagem de Valores Únicos – Country City Views Big Numbers 2024
Country    107
City      1925
Views     636
dtype: int64

Análise de Valores Nulos – Country City Views Big Numbers 2024
Country    0
City       0
Views     0
dtype: int64

print("Estatísticas Básicas – GA Universal 2019–2022")
print(ga_universal_2019a2022.describe())
print("\nContagem de Valores Únicos – GA Universal 2019–2022")
print(ga_universal_2019a2022.nunique())
print("\nAnálise de Valores Nulos – GA Universal 2019–2022")
print(ga_universal_2019a2022.isnull().sum())
```

Estatísticas Básicas – GA Universal 2019–2022

	Users	Sessions
count	218.000000	218.000000
mean	3806.990826	4742.027523
std	31658.105844	44842.930134
min	50.000000	0.000000
25%	50.000000	0.000000
50%	50.000000	0.000000
75%	198.000000	50.000000
max	451382.000000	640895.000000

Contagem de Valores Únicos – GA Universal 2019–2022

Page Title	218
Users	43
Sessions	23

dtype: int64

Análise de Valores Nulos – GA Universal 2019–2022

Page Title	0
Users	0
Sessions	0

dtype: int64

```
ga_universal_2019a2022.describe()
```

	Users	Sessions
count	218.000000	218.000000
mean	3806.990826	4742.027523
std	31658.105844	44842.930134
min	50.000000	0.000000
25%	50.000000	0.000000
50%	50.000000	0.000000
75%	198.000000	50.000000
max	451382.000000	640895.000000

```
print("Estatísticas Básicas – GA Universal 2019–2022")
print(ga_universal_2019a2022.describe())
print("\nContagem de Valores Únicos – GA Universal 2019–2022")
print(ga_universal_2019a2022.nunique())
print("\nAnálise de Valores Nulos – GA Universal 2019–2022")
print(ga_universal_2019a2022.isnull().sum())
```

Estatísticas Básicas – GA Universal 2019–2022

	Users	Sessions
count	218.000000	218.000000
mean	3806.990826	4742.027523
std	31658.105844	44842.930134
min	50.000000	0.000000
25%	50.000000	0.000000
50%	50.000000	0.000000
75%	198.000000	50.000000
max	451382.000000	640895.000000

Contagem de Valores Únicos – GA Universal 2019–2022

```
Page Title    218
Users         43
Sessions      23
dtype: int64
```

```
Análise de Valores Nulos - GA Universal 2019-2022
Page Title    0
Users         0
Sessions      0
dtype: int64
```

✓ Tratando Outliers

Tratamento de Outliers Esta seção aborda métodos para identificar e tratar outliers em conjuntos de dados, usando como exemplo a distribuição de usuários por país. Um ponto focal é a agregação de países com baixas contagens de usuários para simplificar análises futuras.

Insight: O tratamento de outliers é crucial para evitar distorções em análises estatísticas, especialmente em dados demográficos onde grandes disparidades podem existir entre diferentes grupos.

```
def soma_por_pais(df):
    resultados = {}
    for pais in df["Country"].unique():
        resultados[pais] = df.loc[df['Country'] == pais, 'Views'].sum()

    return resultados

x = soma_por_pais(country_city_views_big_numbers_2023)
```

```
def agregarOutliers(paisesValores, paisTarguet):
    limite = paisesValores[paisTarguet]
    resultado = {"others": 0}
    for pais, valor in paisesValores.items():
        if valor < limite:
            resultado["others"] += valor
        else:
            resultado[pais] = valor

    return resultado
```

```
outliersAgregados = agregarOutliers(x, "Hungary")
```

```
outliersAgregados

{'others': 9384,
 'Brazil': 2278578,
 'Indonesia': 3422,
 'United States': 6768,
 'Hungary': 1533}
```

✓ Resto do mundo vs Brasil

```
agregarOutliers(x, "Brazil")

{'others': 21107, 'Brazil': 2278578}
```

Conclui-se que os outros países contêm menos de 1% dos usuários, logo eles podem não ser tão relevantes para as análise em relação ao público do Brasil.

✓ Idade Unknown

Analisa a distribuição de idade dos usuários, lidando especificamente com entradas desconhecidas. Métodos são aplicados para redistribuir proporções de sessões desconhecidas baseadas nas conhecidas, proporcionando uma visão mais clara do perfil de idade do usuário.

Insight: A distribuição correta de dados desconhecidos pode fornecer insights mais precisos e é fundamental em análises demográficas para uma segmentação eficaz do mercado.

```
def calcular_total_idades(x):
    soma = 0
    for idade in x.keys():
        if idade != "unknown":
            soma += x[idade]

    return soma

def distribuir_unknown_idade(df):
    res = {}
    df_dict = df.set_index('Age').to_dict()['Sessions']
    total = calcular_total_idades(df_dict)

    for idade in df_dict.keys():
        if idade != "unknown":
            prob_idade = df_dict[idade]/total
            res[idade] = df_dict[idade] + (prob_idade * df_dict["unknown"])

    return res

print(f'2023: {distribuir_unknown_idade(age_sessions_big_numbers_2023)}')
print(f'2024: {distribuir_unknown_idade(age_sessions_big_numbers_2024)}')

2023: {'18-24': 694519.5867764077, '25-34': 531026.134164721, '35-44': 349471.99115251563, '45-54': 218172.82429192794,
2024: {'18-24': 345295.46033568715, '25-34': 196935.88616900262, '35-44': 105974.73213710802, '45-54': 62171.45001180486}
```

Aqui, nos dados de idade da sessão, a partir da ponderação dos dados conhecidos, foi feita a distribuição dos dados desconhecidos.

✓ Correlação tempo e semana, preparação para Monte Carlo

Aqui há uma pequena simulação, que deve ser usada como base para a próxima sprint e está distribuindo as horas mais acessadas ao longo dos dias mais acessados.

```
def calcular_prob(df, index, value):
    res = {}
    df_dict = df.set_index(index).to_dict()[value]
    total = sum(df_dict.values())
    for chave, valor in df_dict.items():
        res[chave] = valor/total
    return res

probs_dia = calcular_prob(week_users_big_numbers_2023, "Day of week", "Active users")
probs_hora = calcular_prob(hour_users_big_numbers_2023, "Hour", "Active users")

def calcular_prob_conjuntas(primeiraProb: dict, segundaProb: dict):
    res = {}
    for dia in primeiraProb.keys():
        dia_dict = {}
        for hora in segundaProb.keys():
            dia_dict[hora] = primeiraProb[dia] * segundaProb[hora]
        res[dia] = dia_dict

    return res

probConjunta = calcular_prob_conjuntas(probs_dia, probs_hora)

def plotlyMapping(probConjunta):
    dataERes = {"data": [], "res": []}
    for dia, valores in probConjunta.items():
        dictDoDia = {"horas": [], "probs": []}
        for hora, prob in valores.items():
            dictDoDia["horas"].append(hora)
            dictDoDia["probs"].append(prob)

        dictDoDia["horas"], dictDoDia["probs"] = zip(*sorted(zip(dictDoDia["horas"], dictDoDia["probs"])))

        dataERes["res"].append(dictDoDia)
        dataERes["data"].append(dia)

    return dataERes
```

```

teste = plotlyMapping(probConjunta)

# Validar Probabilidades
def validarProbs(probConjunta):
    resultado = 0
    for chave in probConjunta.keys():
        for valor in probConjunta[chave].values():
            resultado += valor
    return resultado

validarProbs(probConjunta)

0.9999999999999992

def obter_dias_horas_mais_quentes(probConjunta: dict, quantidade):
    valores = []
    for chaveExterna in probConjunta.keys():
        for chaveInternas, valor in probConjunta[chaveExterna].items():
            valores.append([chaveExterna, chaveInternas, valor])

    maiores_valores = sorted(valores, key= lambda x: x[2], reverse = True)[:quantidade]

    maiores_valores_dict = {}

    for valor in maiores_valores:
        maiores_valores_dict[f'dia da semana: {valor[0]}, hora do dia {valor[1]}h'] = valor[2]

    return maiores_valores_dict

obter_dias_horas_mais_quentes(probConjunta, 10)

{'dia da semana: 3, hora do dia 20h': 0.011244429025660852,
'dia da semana: 4, hora do dia 20h': 0.011192161181339227,
'dia da semana: 5, hora do dia 20h': 0.011178432043177046,
'dia da semana: 3, hora do dia 19h': 0.010770693815857955,
'dia da semana: 4, hora do dia 19h': 0.010720628050284762,
'dia da semana: 2, hora do dia 20h': 0.010713672022046929,
'dia da semana: 5, hora do dia 19h': 0.010707477329766804,
'dia da semana: 3, hora do dia 21h': 0.010543940716428965,
'dia da semana: 4, hora do dia 21h': 0.010494928974645984,
'dia da semana: 5, hora do dia 21h': 0.010482055113417516}

def obter_dias_horas_mais_frias(probConjunta: dict, quantidade):
    valores = []
    for chaveExterna in probConjunta.keys():
        for chaveInternas, valor in probConjunta[chaveExterna].items():
            valores.append([chaveExterna, chaveInternas, valor])

    maiores_valores = sorted(valores, key= lambda x: x[2])[:quantidade]

    maiores_valores_dict = {}

    for valor in maiores_valores:
        maiores_valores_dict[f'dia da semana: {valor[0]}, hora do dia {valor[1]}h'] = valor[2]

    return maiores_valores_dict

obter_dias_horas_mais_frias(probConjunta, 10)

{'dia da semana: 6, hora do dia 4h': 0.0006723304212184054,
'dia da semana: 1, hora do dia 4h': 0.0006778005410950067,
'dia da semana: 7, hora do dia 4h': 0.0006965226775774223,
'dia da semana: 2, hora do dia 4h': 0.0007183735568680454,
'dia da semana: 5, hora do dia 4h': 0.0007495366640438352,
'dia da semana: 4, hora do dia 4h': 0.0007504572307546702,
'dia da semana: 3, hora do dia 4h': 0.0007539618963033152,
'dia da semana: 6, hora do dia 3h': 0.0008082507975545721,
'dia da semana: 1, hora do dia 3h': 0.0008148267736125496,
'dia da semana: 7, hora do dia 3h': 0.0008373338345252705}

```

✓ Análise Gráfica

✓ Horário e dia da semana

Análise de Horários e Dias da Semana Explora os padrões de uso baseados em horários e dias da semana, identificando picos de atividade que podem informar estratégias de marketing e operacionais.

```
probs_dia = calcular_prob(week_users_big_numbers_2023, "Day of week", "Active users")
probs_hora = calcular_prob(hour_users_big_numbers_2023, "Hour", "Active users")

def calcular_prob_conjuntas(primeiraProb: dict, segundaProb: dict):
    res = {}
    for dia in primeiraProb.keys():
        dia_dict = {}
        for hora in segundaProb.keys():
            dia_dict[hora] = primeiraProb[dia] * segundaProb[hora]
        res[dia] = dia_dict

    return res

probConjunta = calcular_prob_conjuntas(probs_dia, probs_hora)

def plotlyMapping(probConjunta):
    dataERes = {"data": [], "res": []}
    for dia, valores in probConjunta.items():
        dictDoDia = {"horas": [], "probs": []}
        for hora, prob in valores.items():
            dictDoDia["horas"].append(hora)
            dictDoDia["probs"].append(prob)

        dictDoDia["horas"], dictDoDia["probs"] = zip(*sorted(zip(dictDoDia["horas"], dictDoDia["probs"])))

        dataERes["res"].append(dictDoDia)
        dataERes["data"].append(dia)

    return dataERes

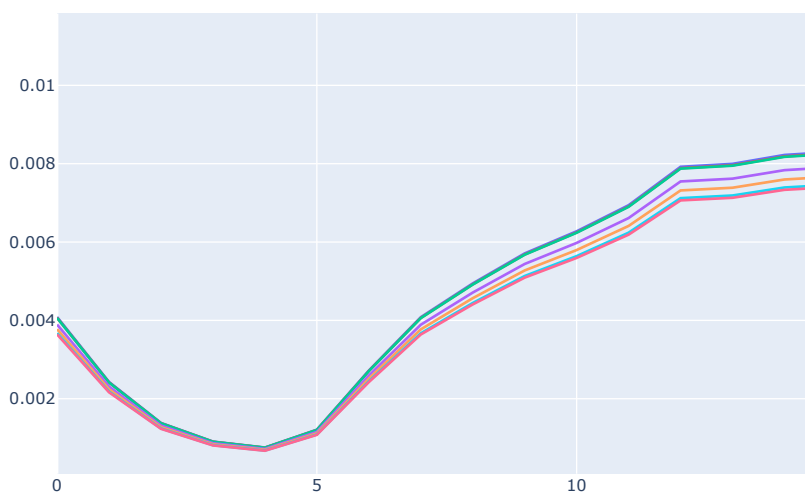
teste = plotlyMapping(probConjunta)

import plotly.graph_objects as go

fig = go.Figure()

for i in range(len(teste["data"])):
    valor = teste["res"][i]
    fig.add_trace(go.Scatter(name= f'dia: {teste["data"][i]}', x=valor['horas'], y=valor["probs"]))

fig.show()
```



Neste gráfico podemos visualizar a probabilidade de alguém iniciar uma sessão no site em certo horário e certo dia da semana. Isso pode nos dar uma visão macro dos horários e datas de maior captação de leads.

Idade

Idade 2023

```
sem_unknown = 'unknown'
age_sessions_big_numbers_2023_filtrado = age_sessions_big_numbers_2023[age_sessions_big_numbers_2023['Age'] != sem_unknown]
```

age_sessions_big_numbers_2023_filtrado

	Age	Sessions	
1	18-24	175969	
2	25-34	134545	
3	35-44	88545	
4	45-54	55278	
5	55-64	35077	
6	65+	22822	

Next steps: [View recommended plots](#)

```
print("\nEstatísticas Básicas - Age Sessions Big Numbers 2023")
print(age_sessions_big_numbers_2023.describe())
print("\nContagem de Valores Únicos - Age Sessions Big Numbers 2023")
print(age_sessions_big_numbers_2023.nunique())
print("\nAnálise de Valores Nulos - Age Sessions Big Numbers 2023")
print(age_sessions_big_numbers_2023.isnull().sum())
```

```
Estatísticas Básicas - Age Sessions Big Numbers 2023
Sessions
count    7.000000e+00
mean     2.888154e+05
std      5.410360e+05
min      2.282200e+04
25%      4.517750e+04
50%      8.854500e+04
75%      1.552570e+05
max      1.509472e+06

Contagem de Valores Únicos - Age Sessions Big Numbers 2023
Age      7
Sessions 7
dtype: int64

Análise de Valores Nulos - Age Sessions Big Numbers 2023
Age      0
Sessions 0
dtype: int64
```

```
age_sessions_big_numbers_2023_filtrado.plot(x='Age', y='Sessions', kind='barh')
plt.show()
```

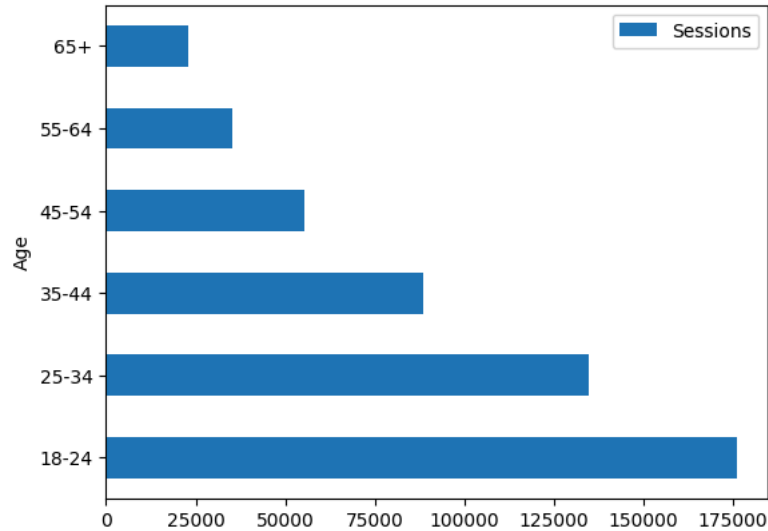


Gráfico da distribuição dos acesso de 2023. Útil para compreensão de público-alvo.

Idade 2024

age_sessions_big_numbers_2024

	Age	Sessions	
0	unknown	598046	
1	18-24	82037	
2	25-34	46789	
3	35-44	25178	
4	45-54	14771	
5	55-64	9394	
6	65+	8195	

Next steps: [View recommended plots](#)

```
print("\nEstatísticas Básicas - Age Sessions Big Numbers 2024")
print(age_sessions_big_numbers_2024.describe())
print("\nContagem de Valores Únicos - Age Sessions Big Numbers 2024")
print(age_sessions_big_numbers_2024.nunique())
print("\nAnálise de Valores Nulos - Age Sessions Big Numbers 2024")
print(age_sessions_big_numbers_2024.isnull().sum())

Estatísticas Básicas - Age Sessions Big Numbers 2024
Sessions
count      7.000000
mean    112058.571429
std     215903.509844
min       8195.000000
25%     12082.500000
50%     25178.000000
75%     64413.000000
max     598046.000000

Contagem de Valores Únicos - Age Sessions Big Numbers 2024
Age          7
Sessions     7
dtype: int64

Análise de Valores Nulos - Age Sessions Big Numbers 2024
Age          0
Sessions     0
dtype: int64

sem_unknown2 = 'unknown'
age_sessions_big_numbers_2024_filtrado = age_sessions_big_numbers_2024[age_sessions_big_numbers_2024['Age'] != sem_unknown2]
```

age_sessions_big_numbers_2024_filtrado

	Age	Sessions	
1	18-24	82037	
2	25-34	46789	
3	35-44	25178	
4	45-54	14771	
5	55-64	9394	
6	65+	8195	

Next steps: [View recommended plots](#)

```
age_sessions_big_numbers_2024_filtrado.plot(x='Age', y='Sessions', kind='barh')
plt.show()
```

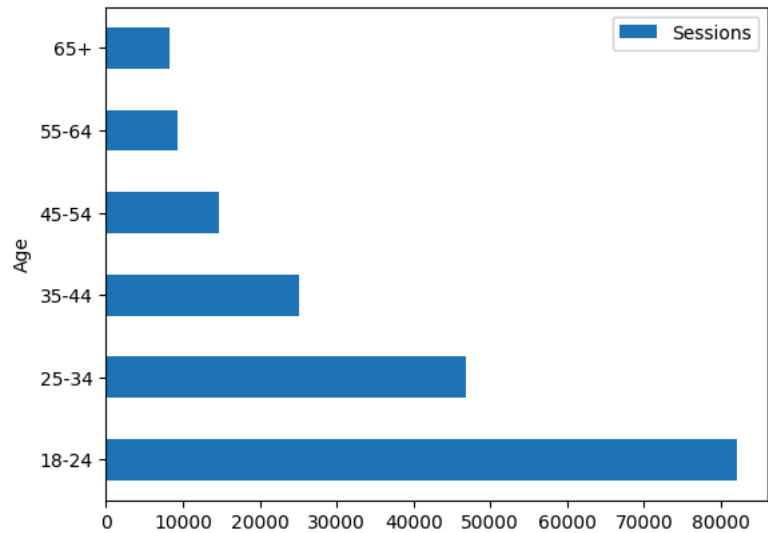



Gráfico da distribuição dos acesso de 2024. Útil para compreensão de público-alvo.

Horários

Horários 2023

hour_users_big_numbers_2023

	Hour	Active users
0	20	143411
1	19	137369
2	21	134477
3	18	126677
4	22	122786
5	17	110903
6	15	105937
7	16	105622
8	14	104875
9	13	101974
10	12	101016
11	23	92486
12	11	88507
13	10	80013
14	9	72816
15	8	63039
16	0	52117
17	7	52050
18	6	34673
19	1	30959
20	2	17621
21	5	15422
22	3	11560
23	4	9616

Next steps: [View recommended plots](#)

```
print("\nEstatísticas Básicas – Hour Users Big Numbers 2023")
print(hour_users_big_numbers_2023.describe())
print("\nContagem de Valores Únicos – Hour Users Big Numbers 2023")
print(hour_users_big_numbers_2023.unique())
print("\nAnálise de Valores Nulos – Hour Users Big Numbers 2023")
print(hour_users_big_numbers_2023.isnull().sum())
```

Estatísticas Básicas – Hour Users Big Numbers 2023

	Hour	Active users
count	24.000000	24.000000
mean	11.500000	79830.25000
std	7.071068	42936.38994
min	0.000000	9616.00000
25%	5.750000	47705.75000
50%	11.500000	90496.50000
75%	17.250000	107178.50000
max	23.000000	143411.00000

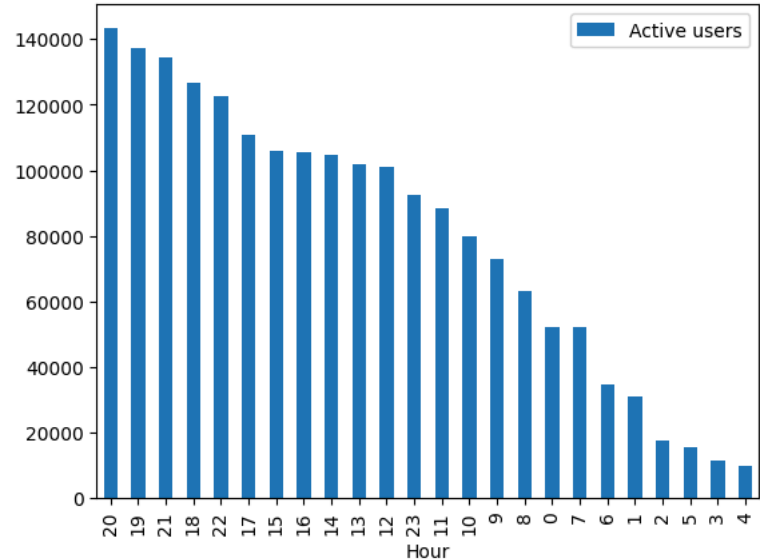
Contagem de Valores Únicos – Hour Users Big Numbers 2023

Hour 24
Active users 24
dtype: int64

Análise de Valores Nulos – Hour Users Big Numbers 2023

Hour 0
Active users 0
dtype: int64

```
hour_users_big_numbers_2023.plot(x='Hour', y='Active users', kind='bar')
plt.show()
```



Distribuição dos horários dos acessos dos usuários em 2023. Útil para entender em qual momento do dia o público-alvo tem maior chance de ser atingido.

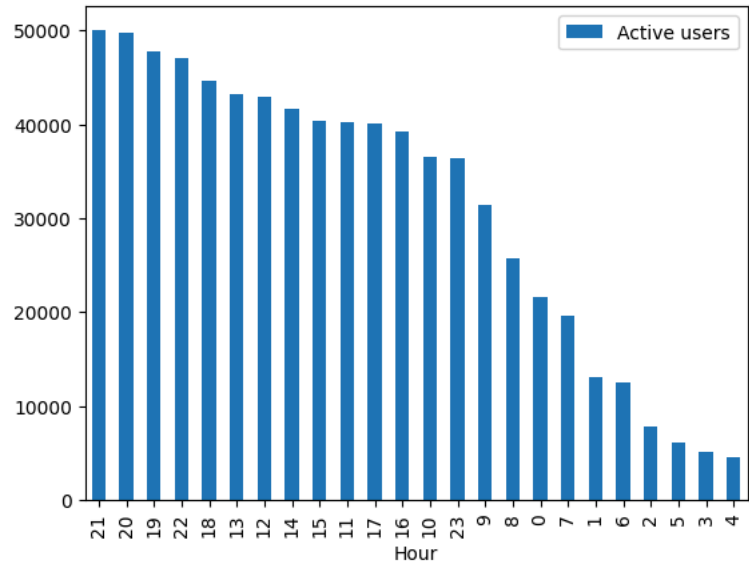
✓ Horários 2024

```
hour_users_big_numbers_2024
```

Hour	Active users	
0	21	50054
1	20	49811
2	19	47832
3	22	47037
4	18	44640
5	13	43217
6	12	42878
7	14	41606
8	15	40411
9	11	40211
10	17	40057
11	16	39276
12	10	36505
13	23	36397
14	9	31439
15	8	25717
16	0	21669
17	7	19561
18	1	13086
19	6	12463
20	2	7835
21	5	6066
22	3	5081
23	4	4493

Next steps: [View recommended plots](#)

```
hour_users_big_numbers_2024.plot(x='Hour', y='Active users', kind='bar')
plt.show()
```



Distribuição dos horários dos acessos dos usuários em 2024. Útil para entender em qual momento do dia o público-alvo tem maior chance de ser atingido.

✓ Dias da semana

week_users_big_numbers_2023

	Day of week	Active users	
0	3	254715	
1	4	253531	
2	5	253220	
3	2	242692	
4	7	235310	
5	1	228985	
6	6	227137	

Next steps: [View recommended plots](#)

```
print("\nEstatísticas Básicas - Week Users Big Numbers 2024")
print(week_users_big_numbers_2024.describe())
print("\nContagem de Valores Únicos - Week Users Big Numbers 2024")
print(week_users_big_numbers_2024.nunique())
print("\nAnálise de Valores Nulos - Week Users Big Numbers 2024")
print(week_users_big_numbers_2024.isnull().sum())
```

Estatísticas Básicas - Week Users Big Numbers 2024

	Day of week	Active users
count	7.000000	7.000000
mean	4.000000	97675.142857
std	2.160247	5926.957101
min	1.000000	90566.000000
25%	2.500000	92256.000000
50%	4.000000	98644.000000
75%	5.500000	102686.000000
max	7.000000	104632.000000

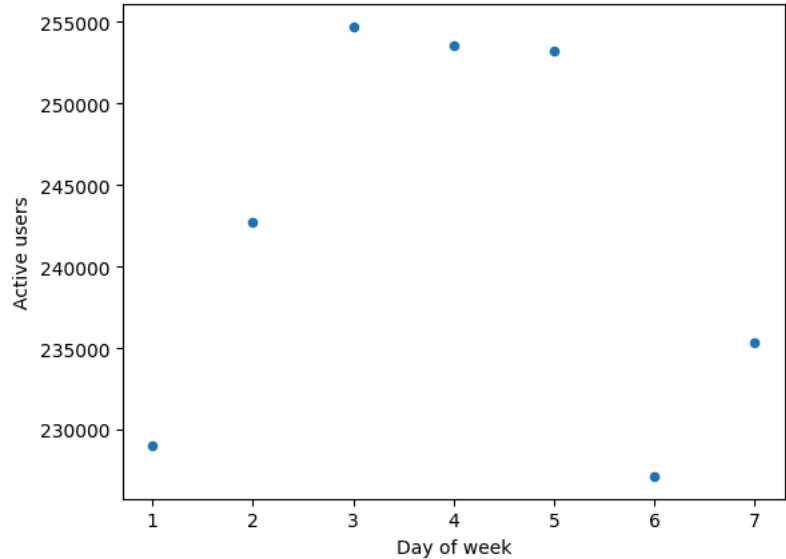
Contagem de Valores Únicos - Week Users Big Numbers 2024

Day of week 7
Active users 7
dtype: int64

Análise de Valores Nulos - Week Users Big Numbers 2024

Day of week 0
Active users 0
dtype: int64

```
week_users_big_numbers_2023.plot(x='Day of week', y='Active users', kind='scatter')
plt.show()
```



Distribuição dos dias da semana mais acessados pelos usuários em 2023. Útil para entender em qual dia da semana e, junto com os horários, o momento exato o público-alvo tem maior chance de ser atingido. De terça a quinta, das 18h as 22h é esse o "horário de pico".

✓ Lugares

Foco na distribuição geográfica dos usuários, destacando a predominância do Brasil no conjunto de dados. A seção utiliza visualizações para argumentar a relevância de focar em mercados específicos.

Seguindo esta premissa, entendemos que a análise geográfica é essencial para estratégias de localização e pode direcionar decisões de investimento em marketing e desenvolvimento de produto.

users_country_big_numbers_geo_2023

	Country	Active users	
0	Brazil	1434957	
1	United States	5879	
2	Indonesia	2247	
3	Hungary	968	
4	Germany	909	
...	
120	Sri Lanka	1	
121	São Tomé & Príncipe	1	
122	Togo	1	
123	Trinidad & Tobago	1	
124	Barbados	1	

125 rows x 2 columns

Next steps:



```
print("\nEstatísticas Básicas – Users Country Big Numbers Geo 2023")
print(users_country_big_numbers_geo_2023.describe())
print("\nContagem de Valores Únicos – Users Country Big Numbers Geo 2023")
print(users_country_big_numbers_geo_2023.nunique())
print("\nAnálise de Valores Nulos – Users Country Big Numbers Geo 2023")
print(users_country_big_numbers_geo_2023.isnull().sum())
```

Estatísticas Básicas – Users Country Big Numbers Geo 2023

	Active users
count	1.250000e+02
mean	1.161508e+04
std	1.283356e+05
min	1.000000e+00
25%	2.000000e+00
50%	6.000000e+00
75%	4.600000e+01
max	1.434957e+06

Contagem de Valores Únicos – Users Country Big Numbers Geo 2023

Country	125
Active users	53

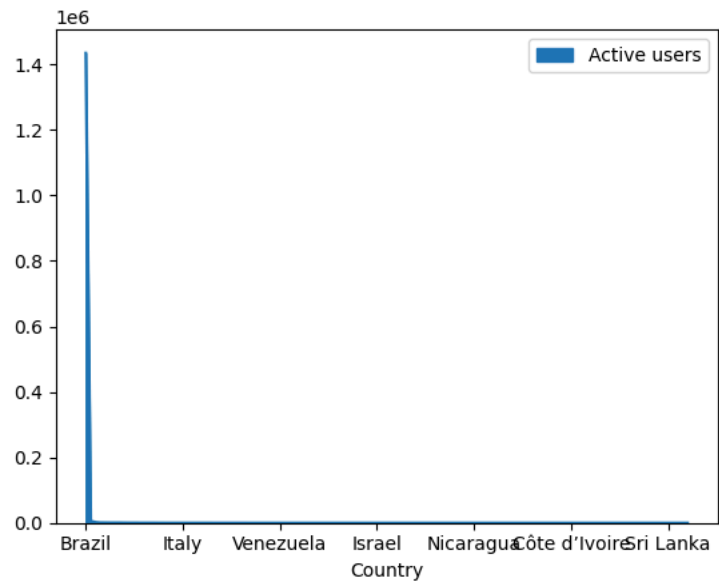
dtype: int64

Análise de Valores Nulos – Users Country Big Numbers Geo 2023

Country	0
Active users	0

dtype: int64

```
users_country_big_numbers_geo_2023.plot(x='Country', y='Active users', kind='area')
plt.show()
```



Conforme análise anterior, mais de 99% acessos são no Brasil, tornando o resto do mundo pouco relevante para as análises.

Google Ads

Entendimento do desempenho de campanhas publicitárias, medindo taxas de conversão e interações, o que é crucial para ajustar abordagens publicitárias e maximizar o retorno sobre o investimento (ROI) das campanhas, com base na conversão.

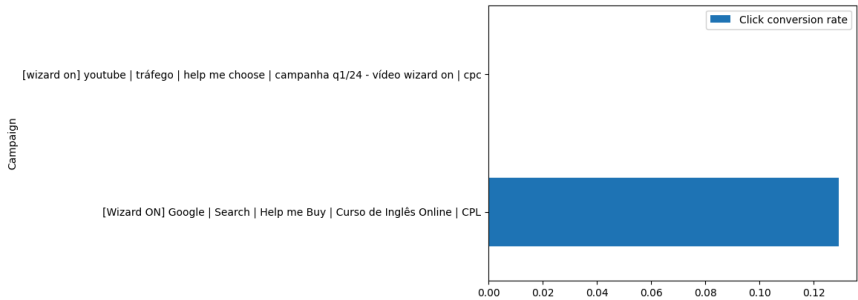
Neste ponto, isso nos ajuda a entender quais estratégias são mais eficazes e que estratégia teve mais importância.

google_ads_2023

Campaign	Campaign status	Interactions	Conversions	Clicks	Click conversion rate	Avg. CPC
[Wizard ON] Google Search Help me	Eligible	136185	17611.474371	136185	0.12932	7.490369

Next steps: ☒ View recommended plots

```
google_ads_2023.plot(x='Campaign', y='Click conversion rate', kind='barh')
plt.show()
```





Nos mostra que não há anúncios no Youtube e que a convesão no Google Ads é de 12 cliques para cada 100 anúncios.


Eventos acessados

Nesta base, temos os eventos mais acessados nos sistemas, oferecendo uma visão sobre quais funcionalidades ou conteúdos atraem mais interesse dos usuários. A intenção aqui é monitorar este eventos para entender o engajamento do usuário e otimizar funcionalidades ou conteúdos baseados em suas preferências.

acesso_eventos_2023

	Event name	Event count	
0	carregamento_pagina	22538287	
1	page_view	2299685	
2	session_start	2012375	
3	first_visit	1402714	
4	user_engagement	412624	
5	scroll	234792	
6	interaction	161115	
7	lead_enviado	74449	
8	form_start	70077	
9	play_video	65675	
10	lead_landing_pages	45046	
11	video_progress	44233	
12	lead_todos_forms	39497	
13	click	36745	
14	video_start	13278	
15	form_contato	9227	
16	file_download	8467	
17	video_complete	8467	
18	lead_enviado_bolsa_estudo	187	
19	form_submit	43	
20	erro_formulario	11	

Next steps:

 [View recommended plots](#)

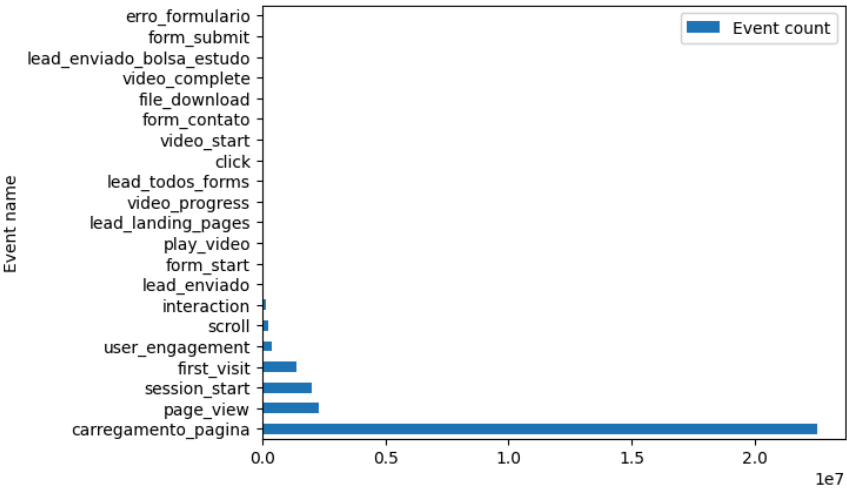
```
print("\nEstatísticas Básicas – Acessos Eventos 2023")
print(acesso_eventos_2023.describe())
print("\nContagem de Valores Únicos – Acessos Eventos 2023")
print(acesso_eventos_2023.nunique())
print("\nAnálise de Valores Nulos – Acessos Eventos 2023")
print(acesso_eventos_2023.isnull().sum())
```

```
Estatísticas Básicas – Acessos Eventos 2023
      Event count
count  2.100000e+01
mean   1.403666e+06
std    4.889671e+06
min    1.100000e+01
25%    9.227000e+03
50%    4.504600e+04
75%    2.347920e+05
max    2.253829e+07

Contagem de Valores Únicos – Acessos Eventos 2023
Event name      21
Event count     20
dtype: int64

Análise de Valores Nulos – Acessos Eventos 2023
Event name      0
Event count     0
dtype: int64

acesso_eventos_2023.plot(x='Event name', y='Event count', kind='barh')
plt.show()
```



Nos dá a consciência de que a página de carregamento é claramente o principal acesso.

Fontes de acesso

Nos dá o entendimento dos sistemas operacionais mais usadas pelos usuários para acessar os serviços, o que pode influenciar decisões de desenvolvimento e suporte técnico para diferentes dispositivos e sistemas operacionais.

Essa análise dentro das fontes de acesso é fundamental para assegurar que o serviço seja otimizado para as plataformas mais usadas, melhorando a experiência do usuário e a eficácia técnica.

acessos_fontes_2023

	Operating system	Event count	
0	Android	24814051	
1	iOS	3466762	
2	Windows	990519	
3	Linux	128601	
4	Chrome OS	36199	
5	Macintosh	35764	
6	(not set)	4747	
7	Tizen	333	
8	BlackBerry	9	
9	Playstation 4	9	

Next steps:

View recommended plots

```
print("\nEstatísticas Básicas – Acessos Fontes 2023")
print(acessos_fontes_2023.describe())
print("\nContagem de Valores Únicos – Acessos Fontes 2023")
print(acessos_fontes_2023.nunique())
print("\nAnálise de Valores Nulos – Acessos Fontes 2023")
print(acessos_fontes_2023.isnull().sum())
```

Estatísticas Básicas – Acessos Fontes 2023

Event count

count 1.000000e+01

mean 2.947699e+06

std 7.759342e+06

min 9.000000e+00

25% 1.436500e+03

50% 3.598150e+04

75% 7.750395e+05

max 2.481405e+07

Contagem de Valores Únicos – Acessos Fontes 2023

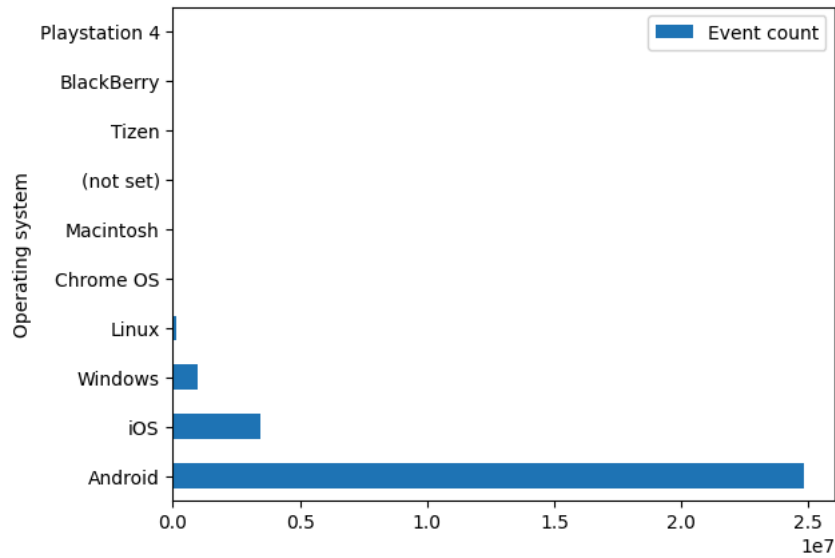
Operating system 10

Event count 9


```
dtype: int64

Análise de Valores Nulos - Acessos Fontes 2023
Operating system    0
Event count         0
dtype: int64
```

```
acessos_fontes_2023.plot(x='Operating system', y='Event count', kind='barh')
plt.show()
```



Nos mostra que o público acessa principalmente via aparelhos android, seguido de longe por aparelhos iOS e Windows.

▼ Dados universais

```
acessos_geral_2023
```

	Hostname	Page path and screen class	Active users	Sessi
0	promocoes.wizard.com.br	/wizard-on/	628757	102
1	promocoes.wizard.com.br	/compromisso-de-aprendizado/	416303	48
2	promocoes.wizard.com.br	/curso-express/	306254	39
3	promocoes.wizard.com.br	/bolsa-de-estudo/	46683	5
4	promocoes.wizard.com.br	/curso-express/agradecimento	32591	3
5	promocoes.wizard.com.br	/wizard-on/agradecimento	26599	2
6	promocoes.wizard.com.br	/matricula-zero/	22720	2
7	promocoes.wizard.com.br	/wizard-on-promo/	19052	2
8	promocoes.wizard.com.br	/bolsa-de-estudo/agradecimento	15592	1
9	promocoes.wizard.com.br	/blue-friday/	4942	1
10	promocoes.wizard.com.br	/grupo-cavalheiro/	4831	1
11	promocoes.wizard.com.br	/matricula-zero/agradecimento	4314	1
12	promocoes.wizard.com.br	/turmas-abertas/	2969	1
13	promocoes.wizard.com.br	/compromisso-de-aprendizado/agradecimento	1778	1
14	promocoes.wizard.com.br	/wizard-on-promo/agradecimento	977	1
15	promocoes.wizard.com.br	/wizard-on-afiliados/	920	1
16	promocoes.wizard.com.br	/turmas-abertas/agradecimento	886	1
17	promocoes.wizard.com.br	/itabira/	789	1
18	promocoes.wizard.com.br	/blue-friday/agradecimento	472	1
19	promocoes.wizard.com.br	/wizard-on-afiliados/agradecimento	437	1
20	promocoes.wizard.com.br	/blue-friday/obrigado	355	1
21	promocoes.wizard.com.br	/litoral/	262	1

Next steps:

[View recommended plots](#)

ga_universal_2019a2022

	Page Title	Users	Sessions	
0	Wizard ONLIFE	451382	640895	
1	Blue Wizard	72653	114332	
2	Blue Friday Wizard	81376	99762	
3	Cursos de Inglês, Espanhol e mais, matrículas ...	42968	62940	
4	Wizard Bye Bye Tradutor!	35633	40589	
...	
213	e-Book - Tudo que você precisa saber sobre a C...	50	0	
214	Promoções – Wizard Jundiaí	50	0	