Análise exploratória

Importantdo módulos e pacotes

In [1]:

```
# Imports

import os
import subprocess
import stat
import numpy as np
from numpy.random import randn
import pandas as pd
from pandas import Series, DataFrame
import seaborn as sns
#sns.set(style='white')
import matplotlib.pyplot as plt

%matplotlib inline

import datetime
from datetime import datetime
from datetime import time
from datetime import date
```

Verificando os valores nulos e substituindo por zeros e 'não possui'

In [2]:

```
#df01.dtypes
#df01.isnull().sum()
#df01copy = df01.copy()
#df02 = df01copy.fillna({
#    'dt_nasc': 0,
#    'renda': 'nao possui'
#})
#df02.head()
```

In [3]:

```
df01 = pd.read_excel('Plan.xlsx')
df01.head()
```

Out[3]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	4 / 8
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	3 / 4
2	1630	5	16	11	4	NAO	ASSINANTE	M	01.01.1900 00:00:00	pos
3	905	9	13	8	25	SIM	ASSINANTE	F	01.01.1900 00:00:00	ACI 25
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	4 / 8

In [4]:

```
df01.shape
```

Out[4]:

```
(5600, 10)
```

In [5]:

```
df01.dtypes
```

Out[5]:

```
id              int64
qt_hit          int64
diasnav         int64
notlidas        int64
visita_capa     int64
usou_app       object
perfil         object
genero         object
dt_nasc        object
renda          object
dtype: object
```

In [6]:

```
df01.isnull().sum()
```

Out[6]:

```
id             0
qt_hit         0
diasnav        0
notlidas       0
visita_capa    0
usou_app       0
perfil         0
genero         0
dt_nasc        0
renda          0
dtype: int64
```

Verificação dos dados

Agrupando os dados por gênero

In [7]:

```
df01['genero'].value_counts()
```

Out[7]:

```
M    3260
F    2204
I     136
Name: genero, dtype: int64
```

In [8]:

```
genero = df01.groupby('genero')
```

Dados do público masculino

In [9]:

```
masculino = genero.get_group('M')
masculino.head()
```

Out[9]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	ren
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	3 A 4
2	1630	5	16	11	4	NAO	ASSINANTE	M	01.01.1900 00:00:00	r pos
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	4 A 8
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	3 A 4
6	842	11	16	10	0	NAO	ASSINANTE	M	04.12.1970 00:00:00	4 A 8

In [10]:

```
masculino.dtypes
```

Out[10]:

```
id              int64
qt_hit          int64
diasnav         int64
notlidas        int64
visita_capa     int64
usou_app       object
perfil         object
genero         object
dt_nasc        object
renda          object
dtype: object
```

In [11]:

```
masculino['genero'].count()
```

Out[11]:

```
3260
```

In [12]:

```
# Média de dias navegados do público masculino = 13
# Média qt de anuncios público masculino = 8
# Média de notícias lidas público masculino = 18
## Média de visitas capa público masculino = 42
masculino.describe().round()
```

Out[12]:

	id	qt_hit	diasnav	notlidas	visita_capa
count	3260.0	3260.0	3260.0	3260.0	3260.0
mean	160665.0	8.0	13.0	18.0	42.0
std	118098.0	23.0	16.0	73.0	138.0
min	3.0	0.0	0.0	0.0	0.0
25%	48044.0	0.0	2.0	0.0	0.0
50%	154650.0	0.0	6.0	2.0	0.0
75%	262862.0	5.0	19.0	9.0	17.0
max	371187.0	345.0	60.0	1773.0	2963.0

In [13]:

```
masculino['nasc'] = pd.to_datetime(masculino['dt_nasc'], errors='coerce')
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [14]:

```
masculino['idade'] = date.today().year - masculino['nasc'].dt.year
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [15]:

```
masculino.head()
```

Out[15]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	ren
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	3 A 4
2	1630	5	16	11	4	NAO	ASSINANTE	M	01.01.1900 00:00:00	r pos
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	4 A 8
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	3 A 4
6	842	11	16	10	0	NAO	ASSINANTE	M	04.12.1970 00:00:00	4 A 8

In [17]:

```
# Dos dados que conseguimos filtrar do público masculino, 2847 estão entre 18 e 100 anos.
masculino[(masculino['idade'] > 18) & (masculino['idade'] < 100)].count()
```

Out[17]:

```
id             2847
qt_hit         2847
diasnav        2847
notlidas       2847
visita_capa    2847
usou_app       2847
perfil         2847
genero         2847
dt_nasc        2847
renda          2847
nasc           2847
idade          2847
dtype: int64
```

In [18]:

```
masculino1 = masculino[(masculino['idade'] > 18) & (masculino['idade'] < 100)]
```

In [19]:

```
masculino1.head()
```

Out[19]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	4 / 8
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	3 / 4
6	842	11	16	10	0	NAO	ASSINANTE	M	04.12.1970 00:00:00	4 / 8
19	1729	0	0	0	0	NAO	ASSINANTE	M	08.07.1989 00:00:00	3 / 4
45	9	6	51	177	0	NAO	ASSINANTE	M	16.08.1985 00:00:00	8 / 14

In [55]:

```
masculino1['genero'].count()
```

Out[55]:

```
2847
```

In [57]:

```
# Média de dias navegados do público masculino = 13
# Média qt de anuncios público masculino = 8
# Média de notícias lidas público masculino = 18
## Média de visitas capa público masculino = 42
# Média de idade = 47
masculino1.describe().round()
```

Out[57]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
count	2847.0	2847.0	2847.0	2847.0	2847.0	2847.0
mean	180900.0	8.0	13.0	18.0	42.0	47.0
std	110027.0	23.0	15.0	73.0	140.0	7.0
min	7.0	0.0	0.0	0.0	0.0	20.0
25%	86951.0	0.0	2.0	0.0	0.0	49.0
50%	179997.0	0.0	7.0	3.0	0.0	49.0
75%	274748.0	5.0	19.0	9.0	17.0	49.0
max	371187.0	345.0	60.0	1773.0	2963.0	99.0

In [20]:

```
# Média de idade do público masculino = 47
masculino1['idade'].mean()
```

Out[20]:

```
47.2781875658588
```

In [21]:

```
# média de idade masculino que usou e não usou o app
mascmediaapp = masculino1.groupby('usou_app').idade.mean()
mascmediaapp
```

Out[21]:

```
usou_app
NAO    47.192140
SIM    47.503817
Name: idade, dtype: float64
```

In [22]:

```
# média de idade masculino conforme renda
mascmediarenda = masculino1.groupby('renda').idade.mean()
mascmediarenda
```

Out[22]:

```
renda
ACIMA DE 25SM       48.257143
ATE 1SM             33.000000
DE 14SM ATE 25SM    44.600000
DE 2SM ATE 3SM      47.808511
DE 3SM ATE 4SM      45.961631
DE 4SM ATE 8SM      47.132898
DE 8SM ATE 14SM     45.789272
não possui          48.080340
Name: idade, dtype: float64
```

In [23]:

```
# Quantidade de assinantes e não assinantes dos produtos do público masculino
masculino1['perfil'].value_counts()
```

Out[23]:

```
PROSPECT     2782
ASSINANTE      65
Name: perfil, dtype: int64
```

In [24]:

```
masculino1['perfil'].count()
```

Out[24]:

```
2847
```

In [26]:

```
# Logo a porcentagem do público masculino assinante é:
# 97% do público masculino não é assinante
# 2% do público masculino é assinante
masculino1['perfil'].value_counts() / masculino1['perfil'].count() * 100
```

Out[26]:

```
PROSPECT     97.716895
ASSINANTE     2.283105
Name: perfil, dtype: float64
```

In [27]:

```
# média de idade masculino conforme perfil
mascmediaperfilidade = masculino1.groupby('perfil').idade.mean()
mascmediaperfilidade
```

Out[27]:

```
perfil
ASSINANTE    44.630769
PROSPECT     47.340043
Name: idade, dtype: float64
```

In [28]:

```
# média notlidas masculino
mascmedianotlidas = masculino1.groupby('id')['notlidas'].max().mean()
mascmedianotlidas
```

Out[28]:

```
17.703547593958554
```

In [29]:

```
# usuários masculinos com maior média de notícias lidas
mascmedianotlidas = masculino1.groupby('id').notlidas.mean().sort_values(ascending=False)
mascmedianotlidas.head()
```

Out[29]:

```
id
73396     1773
304968    1077
353950    1012
108301     850
262033     748
Name: notlidas, dtype: int64
```

In [30]:

```
# moda notlidas masculino (o valor mais frequente)
#mascmodanotlidas = masculino1.groupby('id')['notlidas'].max().mode()
#mascmodanotlidas
```

In [31]:

```
# usuários masculinos que mais visitaram a capa
mascmediacapa = masculino1.groupby('id').visita_capa.mean().sort_values(ascending=False)
mascmediacapa.head(10)
```

Out[31]:

```
id
365635    2963
145071    1544
204645    1541
353950    1390
73396     1383
3567      1366
241284    1346
354808    1299
262033    1244
78566     1243
Name: visita_capa, dtype: int64
```

In [32]:

```
# os usuários masculinos que mais receberam anúncio (qt_hit)
mascmediahit = masculino1.groupby('id').qt_hit.mean().sort_values(ascending=False)
mascmediahit.head(10)
```

Out[32]:

```
id
176232    345
72665     285
300136    249
101290    210
148988    202
268168    202
274299    192
23120     192
23935     186
24        172
Name: qt_hit, dtype: int64
```

Observação: ** desenvolver isso

In [33]:

```
###### ****** observação: posso analisar os usuários e suas características no sistema
# para verificar o perfil que mais lê notícias
```

In [34]:

```
#masculino.dtypes
```

In [35]:

```
#masculino['notlidas'] = masculino['notlidas'].astype(np.int64)
```

In [36]:

```
#masculino['qt_hit'] = masculino['notlidas'].astype(np.int64)
```

In [37]:

```
#masculino['diasnav'] = masculino['notlidas'].astype(np.int64)
```

In [38]:

```
#masculino['notlidas'] = masculino['notlidas'].astype(np.int64)
```

In [39]:

```
#masculino['visita_capa'] = masculino['notlidas'].astype(np.int64)
```

In [40]:

```
#masculino.dtypes
```

In [41]:

```
#masculino['idade'].unique()
# Com ométodo unique, observou-se que os valores nas colunas apresentam uma anomalias.
```

In [42]:

```
# valor único????
# masculino['coluna'].unique()
# alguma anomalia?
```

Dados do público feminino

In [43]:

```
feminino = genero.get_group('F')
feminino.head()
```

Out[43]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	
3	905	9	13	8	25	SIM	ASSINANTE	F	01.01.1900 00:00:00	AC 2
7	3474	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	
8	187	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	pc
11	2607	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	pc

In [44]:

```
#df01['nasc'] = pd.to_datetime(df01['dt_nasc'], errors='coerce')
```

In [45]:

```
#df01['idade'] = date.today().year - df01['nasc'].dt.year
```

In [46]:

```
feminino['nasc'] = pd.to_datetime(feminino['dt_nasc'], errors='coerce')
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [47]:

```
feminino['idade'] = date.today().year - feminino['nasc'].dt.year
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [50]:

```
feminino.head()
```

Out[50]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	
3	905	9	13	8	25	SIM	ASSINANTE	F	01.01.1900 00:00:00	AC 2
7	3474	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	
8	187	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	pc
11	2607	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	pc

In [ ]:

```
```

In [51]:

```
feminino1 = feminino[(feminino['idade'] > 18) & (feminino['idade'] < 100)]
```

In [52]:

```
feminino1.head(2)
```

Out[52]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	4 / 8
25	2621	0	0	0	0	NAO	ASSINANTE	F	25.07.1975 00:00:00	8 / 14

In [53]:

```
# Dos dados que conseguimos filtrar do público feminino, 2045 estão entre 18 e 100 anos
feminino1['perfil'].count()
```

Out[53]:

```
2045
```

In [54]:

```
# Média de dias navegados do público feminino = 9
# Média qt de anuncios público feminino = 5
# Média de notícias lidas público feminino = 16
## Média de visitas capa público feminino = 21
feminino1.describe().round()
```

Out[54]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
count	2045.0	2045.0	2045.0	2045.0	2045.0	2045.0
mean	182298.0	5.0	9.0	16.0	21.0	47.0
std	108008.0	29.0	13.0	68.0	100.0	8.0
min	5.0	0.0	0.0	0.0	0.0	19.0
25%	86322.0	0.0	2.0	1.0	0.0	49.0
50%	184872.0	0.0	4.0	3.0	0.0	49.0
75%	274666.0	1.0	10.0	8.0	1.0	49.0
max	370544.0	894.0	60.0	1102.0	2035.0	94.0

In [58]:

```
# Média de idade do público feminino = 52
feminino1['idade'].mean()
```

Out[58]:

```
47.30513447432763
```

In [59]:

```
# média de idade feminino que usou e não usou o app
femediaapp = feminino1.groupby('usou_app').idade.mean()
femediaapp
```

Out[59]:

```
usou_app
NAO    47.093732
SIM    48.671533
Name: idade, dtype: float64
```

In [60]:

```
# média de idade feminino conforme renda
femediarenda = feminino1.groupby('renda').idade.mean().round()
femediarenda
```

Out[60]:

```
renda
ACIMA DE 25SM       48.0
ATE 1SM             27.0
DE 14SM ATE 25SM    47.0
DE 2SM ATE 3SM      46.0
DE 3SM ATE 4SM      47.0
DE 4SM ATE 8SM      45.0
DE 8SM ATE 14SM     47.0
não possui          48.0
Name: idade, dtype: float64
```

In [61]:

```
# Quantidade de assinantes e não assinantes dos produtos do público feminino
feminino1['perfil'].value_counts()
```

Out[61]:

```
PROSPECT     2022
ASSINANTE      23
Name: perfil, dtype: int64
```

In [62]:

```
# Logo a porcentagem do público feminino assinante é
# 99% do público feminino não é assinante
# 1% do público feminino é assinante
feminino1['perfil'].value_counts() / feminino1['perfil'].count() * 100
```

Out[62]:

```
PROSPECT     98.875306
ASSINANTE     1.124694
Name: perfil, dtype: float64
```

In [63]:

```
# média de idade feminino conforme perfil
femediaperfilidade = feminino1.groupby('perfil').idade.mean()
femediaperfilidade
```

Out[63]:

```
perfil
ASSINANTE    40.000000
PROSPECT     47.388229
Name: idade, dtype: float64
```

In [64]:

```
# média notlidas feminino
#femedianotlidas = feminino1.groupby('id')['notlidas'].max().mean()
#femedianotlidas
```

In [65]:

```
# usuários femininos com mais noticias lidas
femedianotlidas = feminino1.groupby('id').notlidas.mean().sort_values(ascending=False)
femedianotlidas.head(10)
```

Out[65]:

```
id
227507    1102
271047     984
166174     843
66781      790
131729     775
114455     675
207654     663
32732      605
226029     580
182163     537
Name: notlidas, dtype: int64
```

In [66]:

```
# moda notlidas feminino (o valor mais frequente)
#femodanotlidas = feminino1.groupby('id')['notlidas'].max().mode()
#femodanotlidas
```

In [67]:

```
# usuários femininos com mais visitas de capa)
femediacapa = feminino1.groupby('id').visita_capa.mean().sort_values(ascending=False)
femediacapa.head(10)
```

Out[67]:

```
id
182163    2035
32732     1391
345768    1021
110672     985
281951     962
298273     953
227507     920
93603      749
226029     710
218609     681
Name: visita_capa, dtype: int64
```

In [68]:

```
# usuários femininos que mais receberam anúncio
femediahit = feminino1.groupby('id').qt_hit.mean().sort_values(ascending=False)
femediahit.head(10)
```

Out[68]:

```
id
215101    894
194429    512
305146    512
110672    222
270723    194
154353    150
188476    147
281951    141
365448    141
33084     140
Name: qt_hit, dtype: int64
```

Observação: ** desenvolver isso

In [69]:

```
###### ****** observação: posso analisar os usuários e suas características no sistema
# para verificar o perfil que mais lê notícias
```

Dados público indefinido

In [70]:

```
indef = genero.get_group('I')
indef.head()
```

Out[70]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc
605	367566	29	47	0	124	SIM	PROSPECT	I	0
648	73118	3	17	19	4	SIM	PROSPECT	I	05.04.1991 00:00:00
764	311563	6	21	8	60	SIM	PROSPECT	I	03.01.1997 00:00:00
781	240379	0	5	6	0	NAO	PROSPECT	I	16.11.1985 00:00:00
786	63535	0	6	2	0	NAO	PROSPECT	I	24.04.1967 00:00:00

In [71]:

```
indef['nasc'] = pd.to_datetime(indef['dt_nasc'], errors='coerce')
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [72]:

```
indef['idade'] = date.today().year - indef['nasc'].dt.year
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [73]:

```
indef.head(2)
```

Out[73]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc
605	367566	29	47	0	124	SIM	PROSPECT	I	0
648	73118	3	17	19	4	SIM	PROSPECT	I	05.04.1991 00:00:00

In [74]:

```
indef1 = indef[(indef['idade'] > 18) & (indef['idade'] < 100)]
```

In [75]:

```
indef1.head(2)
```

Out[75]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc
605	367566	29	47	0	124	SIM	PROSPECT	I	0
648	73118	3	17	19	4	SIM	PROSPECT	I	05.04.1991 00:00:00

In [78]:

```
# Dos dados que conseguimos filtrar do público indefinido, 136 estão entre 18 e 100 anos
indef1['genero'].count()
```

Out[78]:

```
136
```

In [82]:

```
# Média de dias navegados do público indefinido = 16
# Média qt de anuncios público indefinido = 8
# Média de notícias lidas público indefinido = 30
## Média de visitas capa público indefinido = 76
# Média idade = 48
indef1.describe().round()
```

Out[82]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
count	136.0	136.0	136.0	136.0	136.0	136.0
mean	196775.0	8.0	16.0	30.0	76.0	48.0
std	107939.0	26.0	17.0	102.0	196.0	7.0
min	5052.0	0.0	1.0	0.0	0.0	22.0
25%	105133.0	0.0	3.0	0.0	0.0	49.0
50%	183432.0	0.0	7.0	3.0	0.0	49.0
75%	300620.0	4.0	21.0	10.0	42.0	49.0
max	370463.0	184.0	60.0	922.0	1169.0	74.0

In [83]:

```
# Média de idade do público indefinino
indef1['idade'].mean()
```

Out[83]:

```
48.4264705882353
```

In [84]:

```
# média de idade indefinino que usou e não usou o app
indefmediaapp = indef1.groupby('usou_app').idade.mean()
indefmediaapp
```

Out[84]:

```
usou_app
NAO    48.206897
SIM    48.816327
Name: idade, dtype: float64
```

In [85]:

```
# média de idade indefinido conforme renda
indefmediarenda = indef1.groupby('renda').idade.mean().round()
indefmediarenda
```

Out[85]:

```
renda
ACIMA DE 25SM       52.0
DE 14SM ATE 25SM    50.0
DE 3SM ATE 4SM      50.0
DE 4SM ATE 8SM      49.0
DE 8SM ATE 14SM     50.0
não possui          48.0
Name: idade, dtype: float64
```

In [86]:

```
# Quantidade de assinantes e não assinantes dos produtos do público indefinido
indef1['perfil'].value_counts()
```

Out[86]:

```
PROSPECT     136
Name: perfil, dtype: int64
```

In [87]:

```
# Logo a porcentagem do público indefinido assinante é:
# 100% do público indefinido não é assinante
indef1['perfil'].value_counts() / indef1['perfil'].count() * 100
```

Out[87]:

```
PROSPECT     100.0
Name: perfil, dtype: float64
```

In [88]:

```
# média de idade indefinido conforme perfil
indefmediaperfilidade = indef1.groupby('perfil').idade.mean()
indefmediaperfilidade
```

Out[88]:

```
perfil
PROSPECT     48.426471
Name: idade, dtype: float64
```

In [86]:

```
# média notlidas indefinido
#indefmedianotlidas = indef1.groupby('id')['notlidas'].max().mean()
#indefmedianotlidas
```

Out[86]:

```
29.830882352941178
```

In [89]:

```
# usuários indefinidos com mais noticias
indefmedianotlidas = indef1.groupby('id').notlidas.mean().sort_values(ascending=False)
indefmedianotlidas.head(10)
```

Out[89]:

```
id
217835    922
112047    386
365791    377
24144     322
23519     278
246879    259
229442    224
93715     160
341781    111
321939     97
Name: notlidas, dtype: int64
```

In [88]:

```
# moda notlidas indefinido (o valor mais frequente)
#indefmodanotlidas = indef1.groupby('id')['notlidas'].max().mode()
#indefmodanotlidas
```

Out[88]:

```
0    0
dtype: int64
```

In [90]:

```
# usuários indefinidos com mais visita de capa
indefmediacapa = indef1.groupby('id').visita_capa.mean().sort_values(ascending=False)
indefmediacapa.head(10)
```

Out[90]:

```
id
217835    1169
7636      1100
197198     828
64814      816
24144      659
81972      623
302321     566
246879     339
23519      313
229442     297
Name: visita_capa, dtype: int64
```

In [91]:

```
# usuários indefinido com mais qt_hit
indefmediahit = indef1.groupby('id').qt_hit.mean().sort_values(ascending=False)
indefmediahit.head(10)
```

Out[91]:

```
id
7636      184
305122    180
330755    141
302321     43
197198     38
173990     37
282337     35
258461     34
81972      33
159880     31
Name: qt_hit, dtype: int64
```

Agrupando os dados por quem usou o app

In [92]:

```
app = df01.groupby('usou_app')
```

Público que usou o app

In [93]:

```
sim = app.get_group('SIM')
sim.head()
```

Out[93]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	r
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	
3	905	9	13	8	25	SIM	ASSINANTE	F	01.01.1900 00:00:00	A(2
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	
13	3081	4	4	0	20	SIM	ASSINANTE	M	01.01.1900 00:00:00	
14	3561	24	49	0	295	SIM	ASSINANTE	M	01.01.1900 00:00:00	1

In [94]:

```
sim['nasc'] = pd.to_datetime(sim['dt_nasc'], errors='coerce')
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [95]:

```
sim['idade'] = date.today().year - sim['nasc'].dt.year
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [96]:

```
sim1 = sim[(sim['idade'] > 18) & (sim['idade'] < 100)]
```

In [97]:

```
sim1.head(2)
```

Out[97]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	4 / 8
58	1605	20	12	0	16	SIM	ASSINANTE	M	14.01.1990 00:00:00	4 / 8

In [101]:

```
# Dos dados que conseguimos filtrar do público indefinido, 1109 estão entre 18 e 100 anos.
sim1['usou_app'].count()
```

Out[101]:

```
1109
```

In [102]:

```
# Média de dias navegados do público indefinido = 24
# Média qt de anuncios público indefinido = 15
# Média de notícias lidas público indefinido = 42
## Média de visitas capa público indefinido = 138
# Média idade = 48
sim1.describe().round()
```

Out[102]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
count	1109.0	1109.0	1109.0	1109.0	1109.0	1109.0
mean	180645.0	15.0	24.0	42.0	138.0	48.0
std	109948.0	28.0	19.0	136.0	239.0	8.0
min	92.0	0.0	1.0	0.0	0.0	19.0
25%	82295.0	1.0	7.0	0.0	14.0	49.0
50%	179523.0	5.0	19.0	0.0	48.0	49.0
75%	274299.0	16.0	39.0	10.0	157.0	49.0
max	370970.0	249.0	60.0	1773.0	2963.0	99.0

In [104]:

```
# Média de idade do público que usou o app
sim1['idade'].mean()
```

Out[104]:

```
47.850315599639316
```

In [106]:

```
# média de idade quem usou o app que usou e não usou o app
simmediaapp = sim1.groupby('usou_app').idade.mean()
simmediaapp
```

Out[106]:

```
usou_app
SIM    47.850316
Name: idade, dtype: float64
```

In [107]:

```
# média de idade quem usou o app conforme renda
simmediarenda = sim1.groupby('renda').idade.mean().round()
simmediarenda
```

Out[107]:

```
renda
ACIMA DE 25SM       49.0
DE 14SM ATE 25SM    44.0
DE 2SM ATE 3SM      46.0
DE 3SM ATE 4SM      46.0
DE 4SM ATE 8SM      48.0
DE 8SM ATE 14SM     48.0
não possui          48.0
Name: idade, dtype: float64
```

In [108]:

```
# Quantidade de assinantes e não assinantes dos produtos do público que usou o app
sim1['perfil'].value_counts()
```

Out[108]:

```
PROSPECT     1094
ASSINANTE      15
Name: perfil, dtype: int64
```

In [109]:

```
sim1['perfil'].count()
```

Out[109]:

```
1109
```

In [111]:

```
# Logo a porcentagem do público indefinido assinante é:
# 99% do público que usou o app não é assinante
# 1% do público que usou o app não é assinante
sim1['perfil'].value_counts() / sim1['perfil'].count() * 100
```

Out[111]:

```
PROSPECT     98.64743
ASSINANTE     1.35257
Name: perfil, dtype: float64
```

In [112]:

```
# média de idade quem usou o app conforme perfil
simmediaperfilidade = sim1.groupby('perfil').idade.mean()
simmediaperfilidade
```

Out[112]:

```
perfil
ASSINANTE    40.800000
PROSPECT     47.946984
Name: idade, dtype: float64
```

In [113]:

```
# média notlidas quem usou o app
simmedianotlidas = sim1.groupby('id')['notlidas'].max().mean()
simmedianotlidas
```

Out[113]:

```
42.396753832281334
```

In [115]:

```
# usuários que usaram app com mais noticias lidas
simmedianotlidas = sim1.groupby('id').notlidas.mean().sort_values(ascending=False)
simmedianotlidas.head(10)
```

Out[115]:

```
id
73396     1773
227507    1102
304968    1077
353950    1012
271047     984
217835     922
166174     843
66781      790
131729     775
262033     748
Name: notlidas, dtype: int64
```

In [179]:

```
# moda notlidas quem usou o app (o valor mais frequente)
#simmodanotlidas = sim.groupby('id')['notlidas'].max().mode()
#simmodanotlidas
```

Out[179]:

```
0    0
dtype: int64
```

In [116]:

```
# usuários que usaram app com mais visitas de capa
simmediacapa = sim1.groupby('id').visita_capa.mean().sort_values(ascending=False)
simmediacapa.head(10)
```

Out[116]:

```
id
365635    2963
182163    2035
145071    1544
204645    1541
32732     1391
353950    1390
73396     1383
3567      1366
241284    1346
354808    1299
Name: visita_capa, dtype: int64
```

In [117]:

```
# usuários que usaram app com mais anúncio
simmediahit = sim1.groupby('id').qt_hit.mean().sort_values(ascending=False)
simmediahit.head(10)
```

Out[117]:

```
id
300136    249
110672    222
268168    202
23120     192
274299    192
23935     186
7636      184
365635    167
154353    150
355777    147
Name: qt_hit, dtype: int64
```

Público que não usou o app

In [119]:

```
nao = app.get_group('NAO')
nao.head()
```

Out[119]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	ren
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	4 A 8
2	1630	5	16	11	4	NAO	ASSINANTE	M	01.01.1900 00:00:00	r pos
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	3 A 4
6	842	11	16	10	0	NAO	ASSINANTE	M	04.12.1970 00:00:00	4 A 8
7	3474	0	0	0	0	NAO	ASSINANTE	F	01.01.1900 00:00:00	4 A 8

In [120]:

```
nao['nasc'] = pd.to_datetime(nao['dt_nasc'], errors='coerce')
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [121]:

```
nao['idade'] = date.today().year - nao['nasc'].dt.year
```

```
C:\Users\bruno.r\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykern
el_launcher.py:1: 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: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [122]:

```
nao1 = nao[(nao['idade'] > 18) & (nao['idade'] < 100)]
```

In [123]:

```
nao1.head(2)
```

Out[123]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	ren
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	[4S A 8S
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	[3S A 4S

In [128]:

```
# Dos dados que conseguimos filtrar do público indefinido, 3919 estão entre 18 e 100 anos.
nao1['usou_app'].count()
```

Out[128]:

```
3919
```

In [129]:

```
# Média de dias navegados do público indefinido = 8
# Média qt de anuncios público indefinido = 4
# Média de notícias lidas público indefinido = 10
## Média de visitas capa público = 5
# Média idade = 47
nao1.describe().round()
```

Out[129]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
count	3919.0	3919.0	3919.0	3919.0	3919.0	3919.0
mean	182252.0	4.0	8.0	10.0	5.0	47.0
std	108951.0	24.0	11.0	35.0	29.0	7.0
min	5.0	0.0	0.0	0.0	0.0	20.0
25%	88108.0	0.0	2.0	1.0	0.0	49.0
50%	182266.0	0.0	4.0	3.0	0.0	49.0
75%	275752.0	1.0	10.0	8.0	0.0	49.0
max	371187.0	894.0	60.0	850.0	628.0	94.0

In [133]:

```
nao1['idade'].mean()
```

Out[133]:

```
47.17019647869355
```

In [131]:

```
# média de idade quem não usou o app que usou e não usou o app
naomediaapp = nao1.groupby('usou_app').idade.mean()
naomediaapp
```

Out[131]:

```
usou_app
NAO    47.170196
Name: idade, dtype: float64
```

In [134]:

```
# média de idade quem não usou o app conforme renda
naomediarenda = nao1.groupby('renda').idade.mean().round()
naomediarenda
```

Out[134]:

```
renda
ACIMA DE 25SM       48.0
ATE 1SM             31.0
DE 14SM ATE 25SM    47.0
DE 2SM ATE 3SM      48.0
DE 3SM ATE 4SM      47.0
DE 4SM ATE 8SM      46.0
DE 8SM ATE 14SM     46.0
não possui          48.0
Name: idade, dtype: float64
```

In [135]:

```
# Quantidade de assinantes e não assinantes dos produtos do público que não usou o app
nao1['perfil'].value_counts()
```

Out[135]:

```
PROSPECT     3846
ASSINANTE      73
Name: perfil, dtype: int64
```

In [137]:

```
# Logo a porcentagem do público indefinido assinante é:
# 98% do público que não usou o app não é assinante
# 2% do público que não usou o app não é assinante
nao1['perfil'].value_counts() / nao1['perfil'].count() * 100
```

Out[137]:

```
PROSPECT     98.13728
ASSINANTE     1.86272
Name: perfil, dtype: float64
```

In [139]:

```
# média de idade quem não usou o app conforme perfil
naomediaperfilidade = nao1.groupby('perfil').idade.mean()
naomediaperfilidade
```

Out[139]:

```
perfil
ASSINANTE    43.958904
PROSPECT     47.231149
Name: idade, dtype: float64
```

In [140]:

```
# usuários que não usaram app com mais noticias lidas
naomedianotlidas = nao1.groupby('id').notlidas.mean().sort_values(ascending=False)
naomedianotlidas.head(10)
```

Out[140]:

```
id
108301    850
114455    675
143723    540
122220    522
106109    490
209602    487
15101     426
209045    420
102139    339
92800     328
Name: notlidas, dtype: int64
```

In [141]:

```
# usuários que não usaram app com mais visitas de capa
naomediacapa = nao1.groupby('id').visita_capa.mean().sort_values(ascending=False)
naomediacapa.head(10)
```

Out[141]:

```
id
176232    628
143723    569
73401     527
114455    496
305146    407
194429    407
254417    348
246879    339
101290    296
113822    285
Name: visita_capa, dtype: int64
```

In [142]:

```
# usuários que não usaram app com mais anúncios
naomediahit = nao1.groupby('id').qt_hit.mean().sort_values(ascending=False)
naomediahit.head(10)
```

Out[142]:

```
id
215101    894
194429    512
305146    512
176232    345
72665     285
101290    210
148988    202
270723    194
305122    180
24        172
Name: qt_hit, dtype: int64
```

Agrupando os dados por renda

In [149]:

```
df01['nasc'] = pd.to_datetime(df01['dt_nasc'], errors='coerce')
```

In [150]:

```
df01['idade'] = date.today().year - df01['nasc'].dt.year
```

In [152]:

```
df011 = df01[(df01['idade'] > 18) & (df01['idade'] < 100)]
```

In [171]:

```
df011.shape
```

Out[171]:

```
(5028, 12)
```

In [154]:

```
renda = df011.groupby('renda')
```

In [155]:

```
nao_possui = renda.get_group('não possui')
nao_possui.head(2)
```

Out[155]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
27	1452	7	1	0	0	NAO	ASSINANTE	F	07.10.1966 00:00:00	po
65	847	4	1	0	3	SIM	ASSINANTE	M	13.05.1996 00:00:00	po

In [156]:

```
nao_possui['renda'].count()
```

Out[156]:

```
2739
```

In [157]:

```
maior25 = renda.get_group('ACIMA DE 25SM')
maior25.head(2)
```

Out[157]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc
430	1790	5	3	1	7	SIM	ASSINANTE	F	31.07.1981 00:00:00
621	363557	29	27	5	132	SIM	PROSPECT	M	0

In [158]:

```
maior25['renda'].count()
```

Out[158]:

```
55
```

In [159]:

```
de14ate25 = renda.get_group('DE 14SM ATE 25SM')
de14ate25.head(2)
```

Out[159]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
46	3153	0	0	0	0	NAO	ASSINANTE	M	0	14 / 25
68	3435	0	16	5	84	SIM	ASSINANTE	M	20.02.1982 00:00:00	14 / 25

In [160]:

```
de14ate25['renda'].count()
```

Out[160]:

```
162
```

In [161]:

```
de8ate14 = renda.get_group('DE 8SM ATE 14SM')
de8ate14.head(2)
```

Out[161]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
25	2621	0	0	0	0	NAO	ASSINANTE	F	25.07.1975 00:00:00	8 / 14
45	9	6	51	177	0	NAO	ASSINANTE	M	16.08.1985 00:00:00	8 / 14

In [162]:

```
de8ate14['renda'].count()
```

Out[162]:

```
435
```

In [163]:

```
de4ate8 = renda.get_group('DE 4SM ATE 8SM')
de4ate8.head(2)
```

Out[163]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	ren
0	3321	0	0	0	0	NAO	ASSINANTE	F	23.04.1981 00:00:00	[4S A 8S
4	1219	1	1	0	9	SIM	ASSINANTE	M	16.08.1977 00:00:00	[4S A 8S

In [164]:

```
de4ate8['renda'].count()
```

Out[164]:

```
807
```

In [165]:

```
de3ate4 = renda.get_group('DE 3SM ATE 4SM')
de3ate4.head(2)
```

Out[165]:

	id	qt_hit	diasnav	notlidas	visita_capa	usou_app	perfil	genero	dt_nasc	re
5	2645	1	6	3	7	NAO	ASSINANTE	M	28.10.1967 00:00:00	3 / 4
19	1729	0	0	0	0	NAO	ASSINANTE	M	08.07.1989 00:00:00	3 / 4

In [166]:

```
de3ate4['renda'].count()
```

Out[166]:

```
758
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: