Análise baseada no perfil

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
```

In [2]:

```
df02 = pd.read_excel('Plancopy.xlsx')
df02.head()
```

Out[2]:

	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 [3]:

```
df02['nasc'] = pd.to_datetime(df02['dt_nasc'], errors='coerce')
```

In [4]:

```
df02['idade'] = date.today().year - df02['nasc'].dt.year
```

In [5]:

```
df02.head(2)
```

Out[5]:

	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
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	[3S A 4S

In [6]:

```
df02.isnull().values.any()
```

Out[6]:

```
True
```

In [7]:

```
df02.isnull().sum()
```

Out[7]:

```
id              0
qt_hit          0
diasnav         0
notlidas        0
visita_capa     0
usou_app        0
perfil          0
genero          0
dt_nasc         0
renda           0
nasc           10
idade          10
dtype: int64
```

In [8]:

```
df03 = df02.fillna(0)
```

In [9]:

```
df03.head(2)
```

Out[9]:

	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
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	[3S A 4S

In [10]:

```
df03.isnull().values.any()
```

Out[10]:

```
False
```

In [11]:

```
df04 = df03[(df03['idade'] > 18) & (df03['idade'] < 100)]
```

In [12]:

```
df04.head(2)
```

Out[12]:

	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 [13]:

```
df04.isnull().sum()
```

Out[13]:

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

In [29]:

```
df04.corr()
```

Out[29]:

	id	qt_hit	diasnav	notlidas	visita_capa	idade
id	1.000000	0.004051	0.001443	-0.014344	0.008686	0.021035
qt_hit	0.004051	1.000000	0.388475	0.030716	0.304910	0.024269
diasnav	0.001443	0.388475	1.000000	0.498112	0.617039	0.016166
notlidas	-0.014344	0.030716	0.498112	1.000000	0.498644	-0.009484
visita_capa	0.008686	0.304910	0.617039	0.498644	1.000000	0.030587
idade	0.021035	0.024269	0.016166	-0.009484	0.030587	1.000000

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
```

In [14]:

```
df04.groupby(['perfil', 'genero'])['notlidas'].aggregate('mean').unstack()
# média de notícias lidas por perfil conforme o gênero
```

Out[14]:

genero	F	I	M
perfil			
ASSINANTE	22.652174	NaN	30.323077
PROSPECT	16.380811	29.830882	17.408699

In [ ]:

```
```

In [20]:

```
df04.groupby(['perfil', 'genero'])['idade'].aggregate('mean').unstack()
# média de idade por perfil conforme o gênero
```

Out[20]:

genero	F	I	M
perfil			
ASSINANTE	40.000000	NaN	44.630769
PROSPECT	47.388229	48.426471	47.340043

In [ ]:

```
```

In [ ]:

```
```

In [21]:

```
df04.groupby(['perfil', 'renda'])['notlidas'].aggregate('mean').unstack()
# média de idade por perfil conforme renda
```

Out[21]:

renda	ACIMA DE 25SM	ATE 1SM	DE 14SM ATE 25SM	DE 2SM ATE 3SM	DE 3SM ATE 4SM	DE 4SM ATE 8SM	DE 8SM ATE 14SM	não possui
perfil								
ASSINANTE	1.00000	NaN	51.000000	3.000000	35.500000	25.368421	40.071429	13.760000
PROSPECT	52.62963	8.0	36.352941	10.313433	26.527027	22.729695	24.966746	10.478629

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [25]:

```
#df04.groupby(['perfil', 'usou_app'])['idade'].aggregate('mean').unstack()
# média de idade quem usou ou não o app por perfil conforme o gênero
```

In [26]:

```
#df04.groupby(['perfil', 'usou_app'])['qt_hit'].aggregate('mean').unstack()
# média de anuncios quem usou ou não o app por perfil
```

In [27]:

```
df04.groupby(['perfil', 'usou_app', ])\
['qt_hit', 'diasnav', 'notlidas', 'visita_capa'].aggregate('mean').unstack()
# média dos atributos quem usou ou não o app por perfil
```

Out[27]:

	qt_hit		diasnav		notlidas		visita_capa	
usou_app	NAO	SIM	NAO	SIM	NAO	SIM	NAO	SIM
perfil								
ASSINANTE	7.095890	18.333333	8.904110	20.266667	19.534247	71.066667	9.630137	160.466
PROSPECT	4.181487	14.935101	8.201248	24.095978	10.311492	42.003656	4.752210	137.450

In [28]:

```
df04.groupby(['genero', 'perfil', ])\
['qt_hit', 'diasnav', 'notlidas', 'visita_capa'].aggregate('mean').unstack()
# média dos atributos conforme perfil por gênero
```

Out[28]:

	qt_hit		diasnav		notlidas		visita_cap
perfil	ASSINANTE	PROSPECT	ASSINANTE	PROSPECT	ASSINANTE	PROSPECT	ASSINAN
genero							
F	3.217391	4.802176	6.434783	9.029674	22.652174	16.380811	2.5217
I	NaN	8.463235	NaN	15.529412	NaN	29.830882	N
M	11.061538	7.749820	12.400000	13.491373	30.323077	17.408699	46.9538

In [19]:

```
#df04.groupby(['renda', 'perfil', ])\
#['qt_hit', 'diasnav', 'notlidas', 'visita_capa'].aggregate('mean').unstack()
# média dos atributos conforme perfil por renda
```

Aplicando crosstab

In [33]:

```
pd.crosstab(df04['genero'],df04['perfil'])
```

Out[33]:

perfil	ASSINANTE	PROSPECT
genero		
F	23	2022
I	0	136
M	65	2782

In [34]:

```
pd.crosstab(df04['genero'],df04['usou_app'])
```

Out[34]:

usou_app	NAO	SIM
genero		
F	1771	274
I	87	49
M	2061	786

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

Agrupando por perfil

In [8]:

```
perfil = df02.groupby('perfil')
```

Análise do assinante

In [10]:

```
assinante = perfil.get_group('ASSINANTE')
assinante.head()
```

Out[10]:

	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 [37]:

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

Out[37]:

```
600
```

In [38]:

```
assinante.dtypes
```

Out[38]:

```
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 [39]:

```
# 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
assinante.describe().round()
```

Out[39]:

	id	qt_hit	diasnav	notlidas	visita_capa
count	600.0	600.0	600.0	600.0	600.0
mean	1891.0	8.0	10.0	21.0	36.0
std	1096.0	23.0	16.0	75.0	121.0
min	3.0	0.0	0.0	0.0	0.0
25%	910.0	0.0	0.0	0.0	0.0
50%	1936.0	0.0	0.0	0.0	0.0
75%	2906.0	6.0	14.0	6.0	7.0
max	3781.0	296.0	60.0	896.0	1366.0

In [40]:

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

```
C:\Users\Resende\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: Settin
gWithCopyWarning:
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 [41]:

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

```
C:\Users\Resende\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: Settin
gWithCopyWarning:
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 [42]:

```
assinante.dtypes
```

Out[42]:

```
id                      int64
qt_hit                  int64
diasnav                 int64
notlidas                int64
visita_capa             int64
usou_app               object
perfil                 object
genero                 object
dt_nasc                object
renda                  object
nasc           datetime64[ns]
idade                   int64
dtype: object
```

In [43]:

```
assinante.head(2)
```

Out[43]:

	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
1	1459	1	23	0	362	SIM	ASSINANTE	M	01.01.1900 00:00:00	[3S A 4S

In [44]:

```
assinante[(assinante['idade'] > 18) & (assinante['idade'] < 100)].count()
```

Out[44]:

```
id             88
qt_hit         88
diasnav        88
notlidas       88
visita_capa    88
usou_app       88
perfil         88
genero         88
dt_nasc        88
renda          88
nasc           88
idade          88
dtype: int64
```

In [45]:

```
# Dos dados que conseguimos filtrar do público indefinido, 88 estão entre 18 e 100 anos
```

In [46]:

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

Out[46]:

```
107.73666666666666
```

In [48]:

```
# média de idade assinante que usou e não usou o app
assmediaapp = assinante.groupby('usou_app').idade.mean()
assmediaapp
```

Out[48]:

```
usou_app
NAO    107.484536
SIM    108.800000
Name: idade, dtype: float64
```

In [49]:

```
# média de idade assinante conforme renda
assmediarenda = assinante.groupby('renda').idade.mean().round()
assmediarenda
```

Out[49]:

```
renda
ACIMA DE 25SM       113.0
DE 14SM ATE 25SM     97.0
DE 2SM ATE 3SM       92.0
DE 3SM ATE 4SM      108.0
DE 4SM ATE 8SM      107.0
DE 8SM ATE 14SM     107.0
não possui          110.0
Name: idade, dtype: float64
```

In [53]:

```
# os usuários que leram mais noticias (notlidas) assinante
assmedianotlidas = assinante.groupby('id').notlidas.mean().sort_values(ascending=False)
assmedianotlidas.head(10)
```

Out[53]:

```
id
2490    896
3749    579
1261    560
1020    524
556     520
3567    410
485     387
1999    382
858     324
3064    318
Name: notlidas, dtype: int64
```

In [54]:

```
# moda notlidas assinante (o valor mais frequente)
assmodanotlidas = assinante.groupby('id')['notlidas'].max().mode()
assmodanotlidas
```

Out[54]:

```
0    0
dtype: int64
```

In [55]:

```
# os usuários que mais visitaram capa (visita_capa) assinante
assmediacapa = assinante.groupby('id').visita_capa.mean().sort_values(ascending=False)
assmediacapa.head(10)
```

Out[55]:

```
id
3567    1366
1999     970
1020     908
2490     873
2963     725
2295     708
145      543
3064     533
550      499
1906     495
Name: visita_capa, dtype: int64
```

In [56]:

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

Out[56]:

```
id
145     296
1999    232
24      172
485     126
2963    120
3144    102
1232     99
3360     96
3370     92
3135     88
Name: qt_hit, dtype: int64
```

Agrupando por não assinantes

In [ ]:

```
#### Não consigo filtrar pelo valor 'PROSPECT'
```

In [57]:

```
prospect = perfil.get_group('PROSPECT')
prospect.head()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-57-65c9f7d9033b> in <module>()
----> 1 prospect = perfil.get_group('PROSPECT')
      2 prospect.head()

~\Anaconda3\lib\site-packages\pandas\core\groupby\groupby.py in get_group(se
lf, name, obj)
    875         inds = self._get_index(name)
    876         if not len(inds):
--> 877             raise KeyError(name)
    878
    879         return obj._take(inds, axis=self.axis)

KeyError: 'PROSPECT'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```