00_exploring_data

September 23, 2019

1 Notebook 0. First Encounter With the Data

1.1 ### By Max Pechyonkin

1.2 Introduction

> Note: I understand that you might have limited time. So, if you would rather read only one notebook, rather than all three, please only read the last one, called `02_data_pipeline_refactored.ipynb`, it has the final result. The first two notebooks contained information about how I arrived at that result.

This is my first attempt of analyzing the data, in which I wasn't aware of any pitfalls lying in front of me. It has my first encounters with inconsistencies within the data and some of preliminary exploration, when I was desiding on approaches to take regarding each variable. Some of the information from this analysis went into the hand-written digital PDF notes accompanying these notebooks. I suggest reading the notes as they are more structured.

1.3 Exploring variables and what they mean

Data comes from this link, and data schema is here.

The data has a lot of features, so let's have a look at the columns first, and then dig into the data.

[1]:
```
import pandas as pd

DATA_PATH = '7004_1.csv'
```

[2]:
```
with open(DATA_PATH) as f:
    header = f.readline()

col_names = header.strip().split(',')
len(col_names)
```

[2]:
```
48
```

[3]:
```
col_names
```

[3]:
```
['id',
 'asins',
 'brand',
 'categories',
```

```
'colors',
'count',
'dateAdded',
'dateUpdated',
'descriptions',
'dimension',
'ean',
'features',
'flavors',
'imageURLs',
'isbn',
'keys',
'manufacturer',
'manufacturerNumber',
'merchants',
'name',
'prices.amountMin',
'prices.amountMax',
'prices.availability',
'prices.color',
'prices.condition',
'prices.count',
'prices.currency',
'prices.dateAdded',
'prices.dateSeen',
'prices.flavor',
'prices.isSale',
'prices.merchant',
'prices.offer',
'prices.returnPolicy',
'prices.shipping',
'prices.size',
'prices.source',
'prices.sourceURLs',
'prices.warranty',
'quantities',
'reviews',
'sizes',
'skus',
'sourceURLs',
'upc',
'vin',
'websiteIDs',
'weight']
```

[4]:
```
# schema = {
# 'id': 'str',
```

```
# 'asins': 'list-str',
# 'brand': 'str',
# 'categories': ,
# 'colors',
# 'count',
# 'dateAdded',
# 'dateUpdated',
# 'descriptions',
# 'dimension',
# 'ean',
# 'features',
# 'flavors',
# 'imageURLs',
# 'isbn',
# 'keys',
# 'manufacturer',
# 'manufacturerNumber',
# 'merchants',
# 'name',
# 'prices.amountMin',
# 'prices.amountMax',
# 'prices.availability',
# 'prices.color',
# 'prices.condition',
# 'prices.count',
# 'prices.currency',
# 'prices.dateAdded',
# 'prices.dateSeen',
# 'prices.flavor',
# 'prices.isSale',
# 'prices.merchant',
# 'prices.offer',
# 'prices.returnPolicy',
# 'prices.shipping',
# 'prices.size',
# 'prices.source',
# 'prices.sourceURLs',
# 'prices.warranty',
# 'quantities',
# 'reviews',
# 'sizes',
# 'skus',
# 'sourceURLs',
# 'upc',
# 'vin',
# 'websiteIDs',
# 'weight'
```

```
# }
```

It turns out the `csv` file provided for the assignment is not valid – some of the rows have an incorrect number of fields. Let's ignore those rows, but print the warnings.

[5]:
```
df = pd.read_csv(DATA_PATH, error_bad_lines=False, warn_bad_lines=True)
```

```
b'Skipping line 251: expected 48 fields, saw 49\nSkipping line 444: expected 48
fields, saw 50\nSkipping line 847: expected 48 fields, saw 49\nSkipping line
848: expected 48 fields, saw 49\nSkipping line 1018: expected 48 fields, saw
49\nSkipping line 1575: expected 48 fields, saw 51\nSkipping line 2133: expected
48 fields, saw 49\nSkipping line 2922: expected 48 fields, saw 51\nSkipping line
3777: expected 48 fields, saw 51\nSkipping line 4057: expected 48 fields, saw
49\nSkipping line 4239: expected 48 fields, saw 49\nSkipping line 4240: expected
48 fields, saw 49\nSkipping line 4384: expected 48 fields, saw 49\nSkipping line
4385: expected 48 fields, saw 49\nSkipping line 5388: expected 48 fields, saw
49\nSkipping line 5480: expected 48 fields, saw 49\nSkipping line 5481: expected
48 fields, saw 49\nSkipping line 5907: expected 48 fields, saw 50\nSkipping line
5908: expected 48 fields, saw 50\nSkipping line 6600: expected 48 fields, saw
49\nSkipping line 6601: expected 48 fields, saw 49\nSkipping line 6602: expected
48 fields, saw 49\nSkipping line 6603: expected 48 fields, saw 49\nSkipping line
7227: expected 48 fields, saw 49\nSkipping line 7228: expected 48 fields, saw
49\nSkipping line 7229: expected 48 fields, saw 49\nSkipping line 7267: expected
48 fields, saw 51\nSkipping line 9025: expected 48 fields, saw 49\nSkipping line
9026: expected 48 fields, saw 49\nSkipping line 9027: expected 48 fields, saw
49\nSkipping line 9417: expected 48 fields, saw 50\nSkipping line 10815:
expected 48 fields, saw 49\nSkipping line 11034: expected 48 fields, saw
50\nSkipping line 11059: expected 48 fields, saw 49\nSkipping line 11060:
expected 48 fields, saw 49\nSkipping line 11339: expected 48 fields, saw
50\nSkipping line 11357: expected 48 fields, saw 49\nSkipping line 11358:
expected 48 fields, saw 49\nSkipping line 11646: expected 48 fields, saw
49\nSkipping line 11647: expected 48 fields, saw 49\nSkipping line 12161:
expected 48 fields, saw 50\nSkipping line 12307: expected 48 fields, saw
49\nSkipping line 12308: expected 48 fields, saw 49\nSkipping line 12614:
expected 48 fields, saw 49\nSkipping line 12615: expected 48 fields, saw
49\nSkipping line 12616: expected 48 fields, saw 49\nSkipping line 12617:
expected 48 fields, saw 49\nSkipping line 12618: expected 48 fields, saw
49\nSkipping line 12619: expected 48 fields, saw 49\nSkipping line 12620:
expected 48 fields, saw 49\nSkipping line 12621: expected 48 fields, saw
49\nSkipping line 12622: expected 48 fields, saw 49\nSkipping line 12623:
expected 48 fields, saw 49\nSkipping line 12799: expected 48 fields, saw
49\nSkipping line 14200: expected 48 fields, saw 49\nSkipping line 14595:
expected 48 fields, saw 49\n'
b'Skipping line 16747: expected 48 fields, saw 49\nSkipping line 16748: expected
48 fields, saw 49\nSkipping line 16749: expected 48 fields, saw 49\nSkipping
line 16750: expected 48 fields, saw 49\nSkipping line 16751: expected 48 fields,
saw 49\nSkipping line 16752: expected 48 fields, saw 49\nSkipping line 16753:
expected 48 fields, saw 49\nSkipping line 17318: expected 48 fields, saw
49\nSkipping line 17319: expected 48 fields, saw 49\nSkipping line 17766:
```

```
expected 48 fields, saw 49\nSkipping line 17767: expected 48 fields, saw
49\nSkipping line 18000: expected 48 fields, saw 52\nSkipping line 18001:
expected 48 fields, saw 52\nSkipping line 18308: expected 48 fields, saw
51\nSkipping line 19223: expected 48 fields, saw 52\nSkipping line 19224:
expected 48 fields, saw 52\n'
/Users/fazzl/miniconda3/envs/fastai_dev/lib/python3.7/site-
packages/IPython/core/interactiveshell.py:3058: DtypeWarning: Columns
(20,21,25,29,30,36) have mixed types. Specify dtype option on import or set
low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Below I copied the warnings into a Python string to count the number of bad rows.

[6]:

```
output = 'Skipping line 251: expected 48 fields, saw 49\nSkipping line 444:␣
 ↪expected 48 fields, saw 50\nSkipping line 847: expected 48 fields, saw␣
 ↪49\nSkipping line 848: expected 48 fields, saw 49\nSkipping line 1018:␣
 ↪expected 48 fields, saw 49\nSkipping line 1575: expected 48 fields, saw␣
 ↪51\nSkipping line 2133: expected 48 fields, saw 49\nSkipping line 2922:␣
 ↪expected 48 fields, saw 51\nSkipping line 3777: expected 48 fields, saw␣
 ↪51\nSkipping line 4057: expected 48 fields, saw 49\nSkipping line 4239:␣
 ↪expected 48 fields, saw 49\nSkipping line 4240: expected 48 fields, saw␣
 ↪49\nSkipping line 4384: expected 48 fields, saw 49\nSkipping line 4385:␣
 ↪expected 48 fields, saw 49\nSkipping line 5388: expected 48 fields, saw␣
 ↪49\nSkipping line 5480: expected 48 fields, saw 49\nSkipping line 5481:␣
 ↪expected 48 fields, saw 49\nSkipping line 5907: expected 48 fields, saw␣
 ↪50\nSkipping line 5908: expected 48 fields, saw 50\nSkipping line 6600:␣
 ↪expected 48 fields, saw 49\nSkipping line 6601: expected 48 fields, saw␣
 ↪49\nSkipping line 6602: expected 48 fields, saw 49\nSkipping line 6603:␣
 ↪expected 48 fields, saw 49\nSkipping line 7227: expected 48 fields, saw␣
 ↪49\nSkipping line 7228: expected 48 fields, saw 49\nSkipping line 7229:␣
 ↪expected 48 fields, saw 49\nSkipping line 7267: expected 48 fields, saw␣
 ↪51\nSkipping line 9025: expected 48 fields, saw 49\nSkipping line 9026:␣
 ↪expected 48 fields, saw 49\nSkipping line 9027: expected 48 fields, saw␣
 ↪49\nSkipping line 9417: expected 48 fields, saw 50\nSkipping line 10815:␣
 ↪expected 48 fields, saw 49\nSkipping line 11034: expected 48 fields, saw␣
 ↪50\nSkipping line 11059: expected 48 fields, saw 49\nSkipping line 11060:␣
 ↪expected 48 fields, saw 49\nSkipping line 11339: expected 48 fields, saw␣
 ↪50\nSkipping line 11357: expected 48 fields, saw 49\nSkipping line 11358:␣
 ↪expected 48 fields, saw 49\nSkipping line 11646: expected 48 fields, saw␣
 ↪49\nSkipping line 11647: expected 48 fields, saw 49\nSkipping line 12161:␣
 ↪expected 48 fields, saw 50\nSkipping line 12307: expected 48 fields, saw␣
 ↪49\nSkipping line 12308: expected 48 fields, saw 49\nSkipping line 12614:␣
 ↪expected 48 fields, saw 49\nSkipping line 12615: expected 48 fields, saw␣
 ↪49\nSkipping line 12616: expected 48 fields, saw 49\nSkipping line 12617:␣
 ↪expected 48 fields, saw 49\nSkipping line 12618: expected 48 fields, saw␣
 ↪49\nSkipping line 12619: expected 48 fields, saw 49\nSkipping line 12620:␣
 ↪expected 48 fields, saw 49\nSkipping line 12621: expected 48 fields, saw␣
 ↪49\nSkipping line 12622: expected 48 fields, saw 49\nSkipping line 12623:␣
 ↪expected 48 fields, saw 49\nSkipping line 12799: expected 48 fields, saw␣
 ↪49\nSkipping line 14200: expected 48 fields, saw 49\nSkipping line 14595:␣
 ↪expected 48 fields, saw 49\n'
```

```
b'Skipping line 16747: expected 48 fields, saw 49\nSkipping line 16748:␣
 ↪expected 48 fields, saw 49\nSkipping line 16749: expected 48 fields, saw␣
 ↪49\nSkipping line 16750: expected 48 fields, saw 49\nSkipping line 16751:␣
 ↪expected 48 fields, saw 49\nSkipping line 16752: expected 48 fields, saw␣
 ↪49\nSkipping line 16753: expected 48 fields, saw 49\nSkipping line 17318:␣
 ↪expected 48 fields, saw 49\nSkipping line 17319: expected 48 fields, saw␣
 ↪49\nSkipping line 17766: expected 48 fields, saw 49\nSkipping line 17767:␣
 ↪expected 48 fields, saw 49\nSkipping line 18000: expected 48 fields, saw␣
 ↪52\nSkipping line 18001: expected 48 fields, saw 52\nSkipping line 18308:␣
 ↪expected 48 fields, saw 51\nSkipping line 19223: expected 48 fields, saw␣
 ↪52\nSkipping line 19224: expected 48 fields, saw 52\n'
```

[6]:
```
b'Skipping line 16747: expected 48 fields, saw 49\nSkipping line 16748: expected
48 fields, saw 49\nSkipping line 16749: expected 48 fields, saw 49\nSkipping
line 16750: expected 48 fields, saw 49\nSkipping line 16751: expected 48 fields,
saw 49\nSkipping line 16752: expected 48 fields, saw 49\nSkipping line 16753:
expected 48 fields, saw 49\nSkipping line 17318: expected 48 fields, saw
49\nSkipping line 17319: expected 48 fields, saw 49\nSkipping line 17766:
expected 48 fields, saw 49\nSkipping line 17767: expected 48 fields, saw
49\nSkipping line 18000: expected 48 fields, saw 52\nSkipping line 18001:
expected 48 fields, saw 52\nSkipping line 18308: expected 48 fields, saw
51\nSkipping line 19223: expected 48 fields, saw 52\nSkipping line 19224:
expected 48 fields, saw 52\n'
```

[7]:
```
output.count('Skipping')
```

[7]:
```
56
```

As we can see, in total 56 rows were invalid in the data, so we skipped them.
Now, we can have a look at the data.

[8]:
```
df.head(20)
```

[8]:

	id	asins	brand	\
0	AVpfHrJ6ilAPnD_xVXOI	NaN	Josmo	
1	AVpfHrJ6ilAPnD_xVXOI	NaN	Josmo	
2	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
3	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
4	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
5	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
6	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
7	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
8	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
9	AVpfHsWP1cnluZ0-eVZ7	NaN	SERVUS BY HONEYWELL	
10	AVpfAwehilAPnD_xS_eB	NaN	NATIVE	
11	AVpfAwehilAPnD_xS_eB	NaN	NATIVE	
12	AVpfAwehilAPnD_xS_eB	NaN	NATIVE	
13	AVpfFCHeilAPnD_xUeS0	NaN	MAUI AND SONS	
14	AVpfFCHeilAPnD_xUeS0	NaN	MAUI AND SONS	
15	AVpfFCHeilAPnD_xUeS0	NaN	MAUI AND SONS	

```
16  AVpfFCHeilAPnD_xUeS0   NaN        MAUI AND SONS
17  AVpfFCHeilAPnD_xUeS0   NaN        MAUI AND SONS
18  AVpfHssKilAPnD_xVXu5   NaN               Hao-bo
19  AVpe8bocLJeJML43y4Kx   NaN            Twisted X

                                          categories  \
0          Clothing,Shoes,Men's Shoes,All Men's Shoes
1          Clothing,Shoes,Men's Shoes,All Men's Shoes
2          All Men's Shoes,Shoes,Men's Shoes,Clothing
3          All Men's Shoes,Shoes,Men's Shoes,Clothing
4          All Men's Shoes,Shoes,Men's Shoes,Clothing
5          All Men's Shoes,Shoes,Men's Shoes,Clothing
6          All Men's Shoes,Shoes,Men's Shoes,Clothing
7          All Men's Shoes,Shoes,Men's Shoes,Clothing
8          All Men's Shoes,Shoes,Men's Shoes,Clothing
9          All Men's Shoes,Shoes,Men's Shoes,Clothing
10      Men's Casual Shoes,Shoes,Men's Shoes,Clothing
11      Men's Casual Shoes,Shoes,Men's Shoes,Clothing
12      Men's Casual Shoes,Shoes,Men's Shoes,Clothing
13         All Men's Shoes,Shoes,Men's Shoes,Clothing
14         All Men's Shoes,Shoes,Men's Shoes,Clothing
15         All Men's Shoes,Shoes,Men's Shoes,Clothing
16         All Men's Shoes,Shoes,Men's Shoes,Clothing
17         All Men's Shoes,Shoes,Men's Shoes,Clothing
18  Clothing, Shoes, Accessories,Bags, Briefcases,...
19         Clothing,Shoes,Men's Shoes,All Men's Shoes

                                                 colors  count  \
0                                                   NaN    NaN
1                                                   NaN    NaN
2                                                   NaN    NaN
3                                                   NaN    NaN
4                                                   NaN    NaN
5                                                   NaN    NaN
6                                                   NaN    NaN
7                                                   NaN    NaN
8                                                   NaN    NaN
9                                                   NaN    NaN
10                                                  NaN    NaN
11                                                  NaN    NaN
12                                                  NaN    NaN
13  Red,Black,Sketchy Slant,Yellow,BlackBlue,Black...    NaN
14  Red,Black,Sketchy Slant,Yellow,BlackBlue,Black...    NaN
15  Red,Black,Sketchy Slant,Yellow,BlackBlue,Black...    NaN
16  Red,Black,Sketchy Slant,Yellow,BlackBlue,Black...    NaN
17  Red,Black,Sketchy Slant,Yellow,BlackBlue,Black...    NaN
18                                              Brown    NaN
```

```
19                                                  NaN    NaN

              dateAdded           dateUpdated  \
0   2016-11-07T00:45:12Z  2016-11-07T00:45:12Z
1   2016-11-07T00:45:12Z  2016-11-07T00:45:12Z
2   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
3   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
4   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
5   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
6   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
7   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
8   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
9   2016-06-14T04:29:57Z  2016-07-09T20:26:48Z
10  2016-05-06T14:41:57Z  2016-05-18T00:09:36Z
11  2016-05-06T14:41:57Z  2016-05-18T00:09:36Z
12  2016-05-06T14:41:57Z  2016-05-18T00:09:36Z
13  2016-11-15T12:45:09Z  2016-11-15T12:45:09Z
14  2016-11-15T12:45:09Z  2016-11-15T12:45:09Z
15  2016-11-15T12:45:09Z  2016-11-15T12:45:09Z
16  2016-11-15T12:45:09Z  2016-11-15T12:45:09Z
17  2016-11-15T12:45:09Z  2016-11-15T12:45:09Z
18  2016-05-08T17:59:26Z  2016-05-09T13:58:45Z
19  2016-08-02T04:08:25Z  2016-08-04T04:07:24Z

                                          descriptions dimension  ...  \
0   [{"dateSeen":["2016-11-07T00:45:12Z"],"sourceU...        NaN  ...
1   [{"dateSeen":["2016-11-07T00:45:12Z"],"sourceU...        NaN  ...
2   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
3   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
4   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
5   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
6   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
7   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
8   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
9   [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...        NaN  ...
10                                                NaN        NaN  ...
11                                                NaN        NaN  ...
12                                                NaN        NaN  ...
13  [{"dateSeen":["2016-08-04T04:07:53Z"],"sourceU...        NaN  ...
14  [{"dateSeen":["2016-08-04T04:07:53Z"],"sourceU...        NaN  ...
15  [{"dateSeen":["2016-08-04T04:07:53Z"],"sourceU...        NaN  ...
16  [{"dateSeen":["2016-08-04T04:07:53Z"],"sourceU...        NaN  ...
17  [{"dateSeen":["2016-08-04T04:07:53Z"],"sourceU...        NaN  ...
18                                                NaN        NaN  ...
19  [{"dateSeen":["2016-08-04T04:07:24Z"],"sourceU...        NaN  ...

    prices.warranty quantities  reviews sizes  \
```

```
0               NaN        NaN      NaN   NaN
1               NaN        NaN      NaN   NaN
2               NaN        NaN      NaN   NaN
3               NaN        NaN      NaN   NaN
4               NaN        NaN      NaN   NaN
5               NaN        NaN      NaN   NaN
6               NaN        NaN      NaN   NaN
7               NaN        NaN      NaN   NaN
8               NaN        NaN      NaN   NaN
9               NaN        NaN      NaN   NaN
10              NaN        NaN      NaN   NaN
11              NaN        NaN      NaN   NaN
12              NaN        NaN      NaN   NaN
13              NaN        NaN      NaN   NaN
14              NaN        NaN      NaN   NaN
15              NaN        NaN      NaN   NaN
16              NaN        NaN      NaN   NaN
17              NaN        NaN      NaN   NaN
18              NaN        NaN      NaN   NaN
19              NaN        NaN      NaN   NaN

                                                  skus  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                                  NaN
5                                                  NaN
6                                                  NaN
7                                                  NaN
8                                                  NaN
9                                                  NaN
10                                                 NaN
11                                                 NaN
12                                                 NaN
13                                                 NaN
14                                                 NaN
15                                                 NaN
16                                                 NaN
17                                                 NaN
18  [{"sourceURLs":["http://www.ebay.com.au/itm/Me...
19                                                 NaN

                                            sourceURLs           upc vin  \
0   https://www.walmart.com/ip/Josmo-8190-Plain-In...  6.993020e+11 NaN
1   https://www.walmart.com/ip/Josmo-8190-Plain-In...  6.993020e+11 NaN
2   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
```

```
3   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
4   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
5   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
6   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
7   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
8   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
9   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...           NaN NaN
10  http://www.walmart.com/ip/Native-Miller-Mens-S...           NaN NaN
11  http://www.walmart.com/ip/Native-Miller-Mens-S...           NaN NaN
12  http://www.walmart.com/ip/Native-Miller-Mens-S...           NaN NaN
13  https://www.walmart.com/ip/Maui-And-Sons-David...  7.451098e+11 NaN
14  https://www.walmart.com/ip/Maui-And-Sons-David...  7.451098e+11 NaN
15  https://www.walmart.com/ip/Maui-And-Sons-David...  7.451098e+11 NaN
16  https://www.walmart.com/ip/Maui-And-Sons-David...  7.451098e+11 NaN
17  https://www.walmart.com/ip/Maui-And-Sons-David...  7.451098e+11 NaN
18  http://www.ebay.com.au/itm/Mens-Faux-Leather-B...  7.135244e+11 NaN
19  http://www.walmart.com/ip/Twisted-X-Western-Bo...  8.848822e+11 NaN

   websiteIDs weight
0         NaN    NaN
1         NaN    NaN
2         NaN    NaN
3         NaN    NaN
4         NaN    NaN
5         NaN    NaN
6         NaN    NaN
7         NaN    NaN
8         NaN    NaN
9         NaN    NaN
10        NaN    NaN
11        NaN    NaN
12        NaN    NaN
13        NaN    NaN
14        NaN    NaN
15        NaN    NaN
16        NaN    NaN
17        NaN    NaN
18        NaN    NaN
19        NaN    NaN

[20 rows x 48 columns]
```

Upon inspecting the first 20 rows of the data frame, it seems that there are duplicates in `id` field of the data, at least in the visible columns. Let's inspect this hypothesis and if true, we will need to remove the duplicates.

I will use vertical layout for better visibility.

[9]:
```
row0 = df.iloc[0]
row1 = df.iloc[1]
data1 = pd.DataFrame([row0, row1]).dropna(axis=1, how='all') # ignore labels␣
 ↪that are NaNs in both rows
data1 = pd.DataFrame([data1.iloc[0], data1.iloc[1], data1.iloc[0] == data1.
 ↪iloc[1]])
data1.transpose()
```

[9]:

	0	1
id	AVpfHrJ6ilAPnD_xVXOI	AVpfHrJ6ilAPnD_xVXOI
brand	Josmo	Josmo
categories	Clothing,Shoes,Men's Shoes,All Men's Shoes	Clothing,Shoes,Men's Shoes,All Men's Shoes
dateAdded	2016-11-07T00:45:12Z	2016-11-07T00:45:12Z
dateUpdated	2016-11-07T00:45:12Z	2016-11-07T00:45:12Z
descriptions	[{"dateSeen":["2016-11-07T00:45:12Z"],"sourceU...	[{"dateSeen":["2016-11-07T00:45:12Z"],"sourceU...
ean	6.99302e+11	6.99302e+11
features	[{"key":"Gender","value":["Men"]},{"key":"Shoe...	[{"key":"Gender","value":["Men"]},{"key":"Shoe...
imageURLs	https://i5.walmartimages.com/asr/13ac3d61-003c...	https://i5.walmartimages.com/asr/13ac3d61-003c...
keys	josmo/8190wnavy75,699302044036,0699302044036	josmo/8190wnavy75,699302044036,0699302044036
manufacturerNumber	8190-W-NAVY-7.5	8190-W-NAVY-7.5
merchants	[{"dateSeen":["2016-11-07T00:45:12Z"],"name":"...	
name	Josmo 8190 Plain Infant Walking Shoes, Navy - ...	
prices.amountMin	39.89	
prices.amountMax	39.89	
prices.condition	NaN	
prices.currency	USD	
prices.dateAdded	2016-11-07T00:45:12Z	
prices.dateSeen	2016-11-05T00:00:00Z	
prices.isSale	true	
prices.merchant	NaN	
prices.offer	REDUCED USD 12.10	
prices.sourceURLs	https://www.walmart.com/ip/Josmo-8190-Plain-In...	
sourceURLs	https://www.walmart.com/ip/Josmo-8190-Plain-In...	
upc	6.99302e+11	

```
merchants           [{"dateSeen":["2016-11-07T00:45:12Z"],"name":"...
name                Josmo 8190 Plain Infant Walking Shoes, Navy - ...
prices.amountMin                                                51.99
prices.amountMax                                                51.99
prices.condition                                                  new
prices.currency                                                   USD
prices.dateAdded                                2016-11-07T00:45:12Z
prices.dateSeen                                 2016-11-05T00:00:00Z
prices.isSale                                                  false
prices.merchant                       UnbeatableSale - Walmart.com
prices.offer                                        REDUCED USD 12.10
prices.sourceURLs   https://www.walmart.com/ip/Josmo-8190-Plain-In...
sourceURLs          https://www.walmart.com/ip/Josmo-8190-Plain-In...
upc                                                      6.99302e+11
```

```
                  Unnamed 0
id                     True
brand                  True
categories             True
dateAdded              True
dateUpdated            True
descriptions           True
ean                    True
features               True
imageURLs              True
keys                   True
manufacturerNumber     True
merchants              True
name                   True
prices.amountMin      False
prices.amountMax      False
prices.condition      False
prices.currency        True
prices.dateAdded       True
prices.dateSeen        True
prices.isSale         False
prices.merchant       False
prices.offer           True
prices.sourceURLs      True
sourceURLs             True
upc                    True
```

The differences are in columns: - `prices.isSale` - this probably determines the minimum and maximum prices to be different - `prices.amountMin` - `prices.amountMax` - `prices.condition` - NaN vs new - `prices.merchant` - NaN vs UnbeatableSale - Walmart.com

OK, it looks like the same `id` can be in different rows depending on the merchant and sale.

Let's look further into more examples, and pay attention to columns that were different in the first example.

Let's grab the second unique `id` from the dataframe and have a look at difference in rows.

[10]:
```
sample_id = 'AVpfHsWP1cnluZ0-eVZ7'
data2 = df[df.id == sample_id].dropna(axis=1, how='all') # again, drop NaNs to␣
 ↪save visual space
data2.transpose()
```

[10]:

	2	3
id	AVpfHsWP1cnluZ0-eVZ7	AVpfHsWP1cnluZ0-eVZ7
brand	SERVUS BY HONEYWELL	SERVUS BY HONEYWELL
categories	All Men's Shoes,Shoes,Men's Shoes,Clothing	All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded	2016-06-14T04:29:57Z	2016-06-14T04:29:57Z
dateUpdated	2016-07-09T20:26:48Z	2016-07-09T20:26:48Z
descriptions	[{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...	[{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features	[{"key":"Gender","value":["Men"]},{"key":"Colo...	[{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs	http://i5.walmartimages.com/dfw/dce07b8c-5844/...	http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys	servusbyhoneywell/zsr101blmlg	servusbyhoneywell/zsr101blmlg
manufacturerNumber	ZSR101BLMLG	ZSR101BLMLG
merchants	[{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...	[{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name	Servus By Honeywell Shoe Studs Zsr101blmlg	Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin	40.02	50.31
prices.amountMax	40.02	50.31
prices.condition	new	new
prices.currency	USD	
prices.dateAdded	2016-06-14T04:29:57Z	
prices.dateSeen	2016-03-08T00:00:00Z	
prices.isSale	false	
prices.merchant	SIM Supply Inc - Walmart.com	
prices.offer	NaN	
prices.sourceURLs	http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...	
sourceURLs	http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...	

```
prices.currency                                                    USD
prices.dateAdded                                    2016-06-14T04:29:57Z
prices.dateSeen                                     2015-11-30T00:00:00Z
prices.isSale                                                     false
prices.merchant                           SIM Supply Inc - Walmart.com
prices.offer                                                        NaN
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                     4  \
id                                                  AVpfHsWP1cnluZ0-eVZ7
brand                                                SERVUS BY HONEYWELL
categories                All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                           2016-06-14T04:29:57Z
dateUpdated                                         2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                    servusbyhoneywell/zsr101blmlg
manufacturerNumber                                          ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin                                                  46.26
prices.amountMax                                                  46.26
prices.condition                                                    new
prices.currency                                                     USD
prices.dateAdded                                    2016-06-14T04:29:57Z
prices.dateSeen                                     2016-04-29T00:00:00Z
prices.isSale                                                     false
prices.merchant                           SIM Supply Inc - Walmart.com
prices.offer                                                        NaN
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                     5  \
id                                                  AVpfHsWP1cnluZ0-eVZ7
brand                                                SERVUS BY HONEYWELL
categories                All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                           2016-06-14T04:29:57Z
dateUpdated                                         2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                    servusbyhoneywell/zsr101blmlg
manufacturerNumber                                          ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
```

```
prices.amountMin                                                  55.99
prices.amountMax                                                  55.99
prices.condition                                                    new
prices.currency                                                    USD
prices.dateAdded                                  2016-06-14T04:29:57Z
prices.dateSeen                                   2016-07-09T00:00:00Z
prices.isSale                                                    false
prices.merchant                        SIM Supply Inc - Walmart.com
prices.offer                                                       NaN
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                     6  \
id                                                AVpfHsWP1cnluZ0-eVZ7
brand                                              SERVUS BY HONEYWELL
categories                All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                         2016-06-14T04:29:57Z
dateUpdated                                       2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                    servusbyhoneywell/zsr101blmlg
manufacturerNumber                                         ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin                                                  41.12
prices.amountMax                                                  41.12
prices.condition                                                    new
prices.currency                                                    USD
prices.dateAdded                                  2016-06-14T04:29:57Z
prices.dateSeen                                   2016-03-10T00:00:00Z
prices.isSale                                                    false
prices.merchant                        SIM Supply Inc - Walmart.com
prices.offer                                                       NaN
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                     7  \
id                                                AVpfHsWP1cnluZ0-eVZ7
brand                                              SERVUS BY HONEYWELL
categories                All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                         2016-06-14T04:29:57Z
dateUpdated                                       2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                    servusbyhoneywell/zsr101blmlg
```

```
manufacturerNumber                                          ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin                                                46.19
prices.amountMax                                                46.19
prices.condition                                                  new
prices.currency                                                   USD
prices.dateAdded                                  2016-06-14T04:29:57Z
prices.dateSeen                                   2016-06-13T00:00:00Z
prices.isSale                                                    true
prices.merchant                        SIM Supply Inc - Walmart.com
prices.offer                                      SAVINGS_AMT USD 9.80
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                   8  \
id                                                AVpfHsWP1cnluZ0-eVZ7
brand                                              SERVUS BY HONEYWELL
categories             All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                         2016-06-14T04:29:57Z
dateUpdated                                       2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                   servusbyhoneywell/zsr101blmlg
manufacturerNumber                                          ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin                                                21.40
prices.amountMax                                                21.40
prices.condition                                                  new
prices.currency                                                   USD
prices.dateAdded                                  2016-06-14T04:29:57Z
prices.dateSeen                                   2016-07-09T00:00:00Z
prices.isSale                                                    true
prices.merchant                        SIM Supply Inc - Walmart.com
prices.offer                                     SAVINGS_AMT USD 34.59
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...

                                                                   9
id                                                AVpfHsWP1cnluZ0-eVZ7
brand                                              SERVUS BY HONEYWELL
categories             All Men's Shoes,Shoes,Men's Shoes,Clothing
dateAdded                                         2016-06-14T04:29:57Z
dateUpdated                                       2016-07-09T20:26:48Z
descriptions        [{"dateSeen":["2016-07-09T20:26:48Z"],"sourceU...
```

```
features            [{"key":"Gender","value":["Men"]},{"key":"Colo...
imageURLs           http://i5.walmartimages.com/dfw/dce07b8c-5844/...
keys                                  servusbyhoneywell/zsr101blmlg
manufacturerNumber                                        ZSR101BLMLG
merchants           [{"dateSeen":["2016-06-14T04:29:57Z"],"name":"...
name                       Servus By Honeywell Shoe Studs Zsr101blmlg
prices.amountMin                                                45.23
prices.amountMax                                                45.23
prices.condition                                                  new
prices.currency                                                   USD
prices.dateAdded                                2016-06-14T04:29:57Z
prices.dateSeen                                 2016-04-18T00:00:00Z
prices.isSale                                                   false
prices.merchant                          SIM Supply Inc - Walmart.com
prices.offer                                    SAVINGS_AMT USD 34.59
prices.sourceURLs   http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
sourceURLs          http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
```

Note 1: An interesting observation. This `id` is associated with this product. There are not shoes per se, but rather overshoes that are worn in showy and icy weather that increase grip between shoes and surface. For now I will assume this is a valid product for the task I was given.

Note 2: I wonder if there are irrelevant products in the data, that are not shoes for men. I will get to exploring this later.

My interest at this point is to see for which columns there are differences. I will retreive column numbers that have more than 1 unique values and use them to only see those columns.

[11]:
```
unique_numbers = data2.nunique(axis=0)
non_unique_cols = unique_numbers[unique_numbers > 1].index
non_unique_cols
```

[11]:
```
Index(['prices.amountMin', 'prices.amountMax', 'prices.dateSeen',
       'prices.isSale', 'prices.offer'],
      dtype='object')
```

[12]:
```
data2[non_unique_cols]
```

[12]:
```
  prices.amountMin prices.amountMax       prices.dateSeen prices.isSale  \
2            40.02            40.02  2016-03-08T00:00:00Z         false
3            50.31            50.31  2015-11-30T00:00:00Z         false
4            46.26            46.26  2016-04-29T00:00:00Z         false
5            55.99            55.99  2016-07-09T00:00:00Z         false
6            41.12            41.12  2016-03-10T00:00:00Z         false
7            46.19            46.19  2016-06-13T00:00:00Z          true
8            21.40            21.40  2016-07-09T00:00:00Z          true
9            45.23            45.23  2016-04-18T00:00:00Z         false

            prices.offer
2                    NaN
3                    NaN
4                    NaN
```

```
5                    NaN
6                    NaN
7   SAVINGS_AMT USD 9.80
8  SAVINGS_AMT USD 34.59
9  SAVINGS_AMT USD 34.59
```

For the second unique `id` that I examined, we can see the following three columns are similar to the example we saw before: - `prices.isSale` - `prices.amountMin` - `prices.amountMax` In addition, there are two more columns with differences: - `prices.dateSeen` - `prices.offer`

My conclusion at this point regarding mupltiple same `id`s in the data is as follows (it also takes into account explanation in the PDF that came with this assignment):

The same product `id` can be seen in the multiple rows in the data because there might be: - different vendors selling the product - different times the same vendor was selling this `id` at differenct prices - sales that also affect the price

1.4 Dealing with NaNs

First, let's have a look at the unqique values, sorted from largest to smallest.

[13]:
```
df.nunique(axis=0).sort_values(ascending=False)
```

[13]:
```
prices.sourceURLs      11827
id                      9963
keys                    9963
sourceURLs              9956
name                    9634
imageURLs               9205
prices.dateAdded        8008
dateUpdated             7966
dateAdded               7855
manufacturerNumber      6925
prices.amountMin        6651
features                6539
prices.amountMax        6526
merchants               6132
upc                     6082
ean                     5653
descriptions            5118
skus                    4450
colors                  2235
brand                   1953
categories              1263
prices.offer            1244
prices.dateSeen         1237
sizes                   1087
asins                    885
prices.merchant          747
manufacturer             669
reviews                  442
```

```
prices.shipping          316
dimension                277
weight                   111
prices.color              66
prices.size               64
prices.condition          11
prices.currency           10
prices.returnPolicy       10
prices.isSale             10
prices.availability        9
prices.warranty            8
prices.count               5
prices.source              5
prices.flavor              3
quantities                 2
websiteIDs                 0
flavors                    0
isbn                       0
count                      0
vin                        0
dtype: int64
```

The last 5 rows are suspicious – they must be NaNs. Let's have a closer look.

[14]:
```
all_nan_cols = df.isnull().all()
all_nan_cols[all_nan_cols]
```

[14]:
```
count         True
flavors       True
isbn          True
vin           True
websiteIDs    True
dtype: bool
```

[15]:
```
all_nan_col_names = all_nan_cols[all_nan_cols].index
all_nan_col_names
```

[15]:
```
Index(['count', 'flavors', 'isbn', 'vin', 'websiteIDs'], dtype='object')
```

Let's get rid of these.

[16]:
```
df = df.drop(all_nan_col_names, axis=1)
```

While we are at it, let's also see percentages of NaNs for each column. If some columns have too many NaNs, let's get rid of those too.

[17]:
```
nrows, ncols = df.shape
nrows, ncols
```

[17]:
```
(19315, 43)
```

[18]:
```
def get_nans_percentages(data):
    nrows, _ = data.shape
```

```
    return (data.isnull().sum(axis=0) / nrows).sort_values(ascending=False)
```

[19]:
```
nans_percentages_by_column = get_nans_percentages(df)
nans_percentages_by_column
```

[19]:
```
quantities              0.999793
prices.flavor           0.999586
prices.source           0.999327
prices.count            0.999275
prices.warranty         0.997308
prices.availability     0.993528
prices.size             0.966192
prices.color            0.961584
weight                  0.955734
prices.returnPolicy     0.949159
reviews                 0.914108
asins                   0.866891
dimension               0.845146
prices.shipping         0.703546
sizes                   0.691017
prices.offer            0.687756
manufacturer            0.655035
skus                    0.547657
descriptions            0.490500
ean                     0.484183
upc                     0.443127
colors                  0.429821
prices.condition        0.347036
prices.merchant         0.286668
features                0.279058
merchants               0.276831
manufacturerNumber      0.218897
imageURLs               0.054051
brand                   0.013357
sourceURLs              0.001087
prices.dateSeen         0.001035
prices.sourceURLs       0.000777
prices.currency         0.000570
prices.isSale           0.000518
prices.amountMin        0.000518
prices.dateAdded        0.000207
prices.amountMax        0.000000
keys                    0.000000
dateUpdated             0.000000
dateAdded               0.000000
categories              0.000000
name                    0.000000
id                      0.000000
```

```
dtype: float64
```

[20]:
```
df.nunique()
```

[20]:
```
id                      9963
asins                    885
brand                   1953
categories              1263
colors                  2235
dateAdded               7855
dateUpdated             7966
descriptions            5118
dimension                277
ean                     5653
features                6539
imageURLs               9205
keys                    9963
manufacturer             669
manufacturerNumber      6925
merchants               6132
name                    9634
prices.amountMin        6651
prices.amountMax        6526
prices.availability        9
prices.color              66
prices.condition          11
prices.count               5
prices.currency           10
prices.dateAdded        8008
prices.dateSeen         1237
prices.flavor              3
prices.isSale             10
prices.merchant          747
prices.offer            1244
prices.returnPolicy       10
prices.shipping          316
prices.size               64
prices.source              5
prices.sourceURLs      11827
prices.warranty            8
quantities                 2
reviews                  442
sizes                   1087
skus                    4450
sourceURLs              9956
upc                     6082
weight                   111
dtype: int64
```

One stragedy would be to remove some of the variables with missing data. At this point there are several possibilities to deal with this:

- manually select some threshold for proportion of missing values and remove all columns that are above the threshold. This method is very easy but has drawbacks, e.g. why this particular threshold?
- tread threshold for removing columns as a hyperparameter of the learning algorithm and select that using cross-validation
- treat NaNs as one value of a categorical variable, because the NaN itself can be a signal to the model
- use encoding that utilizes the knowledge about NaNs, such as CatBoost encoder

For now, my approach is this:

- remove all vars with more then 99% missing values
- go through each of remaining vars and decide what transformation / parsing needs to be done
- perform transformation or parsing

1.5 Get rid of columns with more than 99% missing

[21]:
```
more_than_99_missing_cols =␣
 ↪nans_percentages_by_column[nans_percentages_by_column > 0.99].index
more_than_99_missing_cols
```

[21]:
```
Index(['quantities', 'prices.flavor', 'prices.source', 'prices.count',
       'prices.warranty', 'prices.availability'],
      dtype='object')
```

[22]:
```
df = df.drop(more_than_99_missing_cols, axis=1)
```

[24]:
```
# recalculate percentages because we removed some columns
nans_percentages_by_column = get_nans_percentages(df)
```

[32]:
```
def check_non_nan_entries(colname: str, data: pd.DataFrame = df) -> pd.Series:
    return data[colname][data[colname].isnull() == False]

for cname in nans_percentages_by_column[nans_percentages_by_column < 0.01].
 ↪index:
    print(f"COLUMN {cname}")
    print(check_non_nan_entries(cname))
    print("-"*40)
```

```
COLUMN sourceURLs
0        https://www.walmart.com/ip/Josmo-8190-Plain-In...
1        https://www.walmart.com/ip/Josmo-8190-Plain-In...
2        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
3        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
4        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
                              ...
```

```
19310    http://www.ebay.com/itm/NEW-XRay-Hester-Mens-L...
19311    https://www.walmart.com/ip/Fila-Men-s-Original...
19312    http://www.sears.com/content/pdp/products/pric...
19313    http://www.sears.com/content/pdp/products/pric...
19314    http://www.sears.com/content/pdp/products/pric...
Name: sourceURLs, Length: 19294, dtype: object
----------------------------------------
COLUMN prices.dateSeen
0        2016-11-05T00:00:00Z
1        2016-11-05T00:00:00Z
2        2016-03-08T00:00:00Z
3        2015-11-30T00:00:00Z
4        2016-04-29T00:00:00Z
                ...
19310    2016-02-27T00:00:00Z
19311    2016-10-05T00:00:00Z
19312    2016-03-17T00:00:00Z
19313    2016-03-31T00:00:00Z
19314    2016-03-31T00:00:00Z
Name: prices.dateSeen, Length: 19295, dtype: object
----------------------------------------
COLUMN prices.sourceURLs
0        https://www.walmart.com/ip/Josmo-8190-Plain-In...
1        https://www.walmart.com/ip/Josmo-8190-Plain-In...
2        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
3        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
4        http://www.walmart.com/ip/Studs-Shoe-Large-Pr-...
                              ...
19310    http://www.ebay.com/itm/NEW-XRay-Hester-Mens-L...
19311    https://www.walmart.com/ip/Fila-Men-s-Original...
19312    http://www.sears.com/content/pdp/products/pric...
19313    http://www.sears.com/content/pdp/products/pric...
19314    http://www.sears.com/content/pdp/products/pric...
Name: prices.sourceURLs, Length: 19300, dtype: object
----------------------------------------
COLUMN prices.currency
0        USD
1        USD
2        USD
3        USD
4        USD
        ...
19310    USD
19311    USD
19312    USD
19313    USD
19314    USD
Name: prices.currency, Length: 19304, dtype: object
```

```
---------------------------------------
COLUMN prices.amountMin
0        39.89
1        51.99
2        40.02
3        50.31
4        46.26
         ...
19310    29.95
19311    64.95
19312    32.48
19313    31.48
19314       72
Name: prices.amountMin, Length: 19305, dtype: object
---------------------------------------
COLUMN prices.isSale
0         true
1        false
2        false
3        false
4        false
         ...
19310    False
19311    False
19312     True
19313     True
19314    False
Name: prices.isSale, Length: 19305, dtype: object
---------------------------------------
COLUMN prices.dateAdded
0        2016-11-07T00:45:12Z
1        2016-11-07T00:45:12Z
2        2016-06-14T04:29:57Z
3        2016-06-14T04:29:57Z
4        2016-06-14T04:29:57Z
                 ...
19310    2016-03-04T05:54:20Z
19311    2016-10-05T22:23:17Z
19312    2016-01-02T02:39:11Z
19313    2016-01-02T02:39:11Z
19314    2016-01-02T02:39:11Z
Name: prices.dateAdded, Length: 19311, dtype: object
---------------------------------------
COLUMN categories
0               Clothing,Shoes,Men's Shoes,All Men's Shoes
1               Clothing,Shoes,Men's Shoes,All Men's Shoes
2               All Men's Shoes,Shoes,Men's Shoes,Clothing
3               All Men's Shoes,Shoes,Men's Shoes,Clothing
```

```
4               All Men's Shoes,Shoes,Men's Shoes,Clothing
                                ...
19310      Clothing, Shoes & Accessories,Men's Shoes,Boots
19311           Clothing,Shoes,Men's Shoes,All Men's Shoes
19312    Women's Sunglasses,Sunglasses,Bags & Accessori...
19313    Women's Sunglasses,Sunglasses,Bags & Accessori...
19314    Women's Sunglasses,Sunglasses,Bags & Accessori...
Name: categories, Length: 19315, dtype: object
----------------------------------------
COLUMN prices.amountMax
0        39.89
1        51.99
2        40.02
3        50.31
4        46.26
         ...
19310    29.95
19311    64.95
19312    32.48
19313    31.48
19314       72
Name: prices.amountMax, Length: 19315, dtype: object
----------------------------------------
COLUMN dateAdded
0        2016-11-07T00:45:12Z
1        2016-11-07T00:45:12Z
2        2016-06-14T04:29:57Z
3        2016-06-14T04:29:57Z
4        2016-06-14T04:29:57Z
                 ...
19310    2016-03-04T05:54:20Z
19311    2016-10-05T22:23:17Z
19312    2016-01-02T02:39:11Z
19313    2016-01-02T02:39:11Z
19314    2016-01-02T02:39:11Z
Name: dateAdded, Length: 19315, dtype: object
----------------------------------------
COLUMN dateUpdated
0        2016-11-07T00:45:12Z
1        2016-11-07T00:45:12Z
2        2016-07-09T20:26:48Z
3        2016-07-09T20:26:48Z
4        2016-07-09T20:26:48Z
                 ...
19310    2016-03-04T05:54:20Z
19311    2016-10-05T22:23:17Z
19312    2016-04-05T23:07:58Z
19313    2016-04-05T23:07:58Z
```

```
19314    2016-04-05T23:07:58Z
Name: dateUpdated, Length: 19315, dtype: object
----------------------------------------
COLUMN keys
0             josmo/8190wnavy75,699302044036,0699302044036
1             josmo/8190wnavy75,699302044036,0699302044036
2                           servusbyhoneywell/zsr101blmlg
3                           servusbyhoneywell/zsr101blmlg
4                           servusbyhoneywell/zsr101blmlg
                              ...
19310    newxrayhestermenslaceupbootsblacksize12/281938...
19311                          789482167691,0789482167691
19312    seekoptics/0027040801,seekopticsnewseekpolariz...
19313    seekoptics/0027040801,seekopticsnewseekpolariz...
19314    seekoptics/0027040801,seekopticsnewseekpolariz...
Name: keys, Length: 19315, dtype: object
----------------------------------------
COLUMN name
0        Josmo 8190 Plain Infant Walking Shoes, Navy - ...
1        Josmo 8190 Plain Infant Walking Shoes, Navy - ...
2               Servus By Honeywell Shoe Studs Zsr101blmlg
3               Servus By Honeywell Shoe Studs Zsr101blmlg
4               Servus By Honeywell Shoe Studs Zsr101blmlg
                              ...
19310    New Xray Hester Men's Lace Up Boots - Black - ...
19311    Fila Men's Original Tennis Navy/white/gold Fas...
19312    Seek Optics New Seek Polarized Replacement Len...
19313    Seek Optics New Seek Polarized Replacement Len...
19314    Seek Optics New Seek Polarized Replacement Len...
Name: name, Length: 19315, dtype: object
----------------------------------------
COLUMN id
0        AVpfHrJ6ilAPnD_xVXOI
1        AVpfHrJ6ilAPnD_xVXOI
2        AVpfHsWP1cnluZ0-eVZ7
3        AVpfHsWP1cnluZ0-eVZ7
4        AVpfHsWP1cnluZ0-eVZ7
                 ...
19310    AVpfdSjlilAPnD_xcGPm
19311    AVpf3bFWilAPnD_xjrQ2
19312    AVpf0fJXLJeJML43EVe9
19313    AVpf0fJXLJeJML43EVe9
19314    AVpf0fJXLJeJML43EVe9
Name: id, Length: 19315, dtype: object
----------------------------------------
```

[31]:
```
df.sourceURLs.iloc[1500]
```

[31]:
```
'http://www.ebay.com/itm/Snickers-Rip-Stop-Cordura-Work-Trousers-with-Kneepad-Holster-Pockets-3213-/191851599675?var=&hash=item2cab3f173b:m:mu-iy1haPji_Omc2Q5Jr6rA'
```

[ ]:
```
df.iloc[-1]['name']
```

[33]:
```
df.categories.iloc[-1]
```

[33]:
```
"Women's Sunglasses,Sunglasses,Bags & Accessories,Men's Accessories,Men's Sunglasses,Clothing, Shoes & Jewelry,Women's Accessories"
```