Pandas_example

August 20, 2019

0.0.1 Example - Pandas dataframes

[1]:
```
import pandas as pd
```

[2]:
```
dataframe1 = pd.read_csv("buy-clicks.csv")
```

[3]:
```
dataframe1.describe()
```

[3]:
```
                txId  userSessionId         team       userId        buyId  \
count   2947.000000    2947.000000  2947.000000  2947.000000  2947.000000
mean   25443.010859   22884.752290    70.318968  1187.459111     2.530709
std     9343.543793    8669.353627    40.274525   685.703809     1.779987
min     6004.000000    5652.000000     2.000000     1.000000     0.000000
25%    17991.500000   16001.000000    35.000000   590.000000     1.000000
50%    26138.000000   23429.000000    69.000000  1190.000000     2.000000
75%    33294.500000   29165.000000    99.000000  1759.000000     4.000000
max    39842.000000   39275.000000   178.000000  2387.000000     5.000000

             price
count  2947.000000
mean      7.263997
std       7.076313
min       1.000000
25%       2.000000
50%       3.000000
75%      10.000000
max      20.000000
```

[4]:
```
dataframe1.head(5)
```

[4]:

	timestamp	txId	userSessionId	team	userId	buyId	price
0	2016-05-26 15:36:54	6004	5820	9	1300	2	3.0
1	2016-05-26 15:36:54	6005	5775	35	868	4	10.0
2	2016-05-26 15:36:54	6006	5679	97	819	5	20.0
3	2016-05-26 16:36:54	6067	5665	18	121	2	3.0
4	2016-05-26 17:06:54	6093	5709	11	2222	5	20.0

[5]:
```
dataframe1.shape
```

[5]:
```
(2947, 7)
```

Filter rows and columns of a DataFrame1 to just show the txId and price

[[ ]] creates a copy of the DataFrame with only the specified columns

[6]:
```
dataframe1[['txId','price']].head(5)
```

[6]:

	txId	price
0	6004	3.0
1	6005	10.0
2	6006	20.0
3	6067	3.0
4	6093	20.0

Filter rows based on a criteria

[7]:
```
dataframe1[dataframe1['price']< 3].head(10)
```

[7]:

	timestamp	txId	userSessionId	team	userId	buyId	price
9	2016-05-26 18:36:54	6184	5697	35	2199	1	2.0
14	2016-05-26 20:06:54	6271	5706	9	1652	0	1.0
15	2016-05-26 20:36:54	6292	5921	2	518	0	1.0
18	2016-05-26 22:06:54	6395	5880	35	2146	1	2.0
19	2016-05-26 22:36:54	6411	6230	77	1457	0	1.0
21	2016-05-26 23:36:54	6458	5698	9	1143	1	2.0
22	2016-05-27 00:06:54	6473	5910	69	1162	1	2.0
31	2016-05-27 04:36:54	6684	5652	11	937	0	1.0
34	2016-05-27 06:06:54	6752	5811	97	649	0	1.0
39	2016-05-27 08:06:54	6866	5698	9	1143	1	2.0

Calculate sum and average of a column

[8]:
```
sum_price = dataframe1['price'].sum()
```

[9]:
```
sum_price
```

[9]:
```
21407.0
```

[10]:
```
mean_price = dataframe1['price'].mean()
```

[11]:
```
mean_price
```

[11]:
```
7.263997285374957
```

Combine two DataFrames

[12]:
```
dataframe2 = pd.read_csv('ad-clicks.csv')
```

[13]:
```
dataframe2.describe()
```

[13]:

	txId	userSessionId	teamId	userId	adId
count	16323.000000	16323.000000	16323.000000	16323.000000	16323.000000
mean	24613.829259	22090.773449	70.294921	1187.464192	14.654046
std	9513.244787	8780.273065	39.631995	691.561945	8.623599
min	5972.000000	5649.000000	2.000000	1.000000	0.000000
25%	16994.500000	15880.000000	35.000000	564.000000	7.000000
50%	25111.000000	21017.000000	69.000000	1161.000000	15.000000
75%	32597.500000	27912.000000	99.000000	1771.000000	22.000000
max	39833.000000	39623.000000	179.000000	2387.000000	29.000000

[14]:
```
dataframe2.head(10)
```

[14]:

	timestamp	txId	userSessionId	teamId	userId	adId	adCategory
0	2016-05-26 15:13:22	5974	5809	27	611	2	electronics
1	2016-05-26 15:17:24	5976	5705	18	1874	21	movies
2	2016-05-26 15:22:52	5978	5791	53	2139	25	computers
3	2016-05-26 15:22:57	5973	5756	63	212	10	fashion
4	2016-05-26 15:22:58	5980	5920	9	1027	20	clothing
5	2016-05-26 15:27:19	5977	5954	77	595	4	games
6	2016-05-26 15:28:51	5981	5674	54	770	6	movies
7	2016-05-26 15:35:25	5975	5919	59	2133	3	electronics
8	2016-05-26 15:35:37	5979	5945	75	253	3	electronics
9	2016-05-26 15:36:38	5972	5914	78	1821	12	computers

[15]:
```
dataframe2.shape
```

[15]:
```
(16323, 7)
```

[16]:
```
merged_df = dataframe2.merge(dataframe1, on='userId')
```

[17]:
```
merged_df.describe()
```

[17]:

	txId_x	userSessionId_x	teamId	userId	\
count	103156.000000	103156.000000	103156.000000	103156.000000	
mean	23289.894955	20768.070010	63.473526	1162.090465	
std	9489.288872	8659.817304	36.619498	693.554005	
min	5972.000000	5649.000000	2.000000	1.000000	
25%	15174.750000	12689.500000	28.000000	508.000000	
50%	23355.500000	20755.000000	64.000000	1146.000000	
75%	31266.250000	26778.000000	90.000000	1740.000000	
max	39833.000000	39623.000000	178.000000	2387.000000	

	adId	txId_y	userSessionId_y	team	\
count	103156.000000	103156.000000	103156.000000	103156.000000	
mean	14.697148	23962.979449	21410.977772	63.464714	
std	8.619775	9421.440331	8631.817943	36.608725	
min	0.000000	6004.000000	5652.000000	2.000000	
25%	7.000000	16542.000000	15799.000000	28.000000	
50%	15.000000	24176.000000	20810.000000	64.000000	
75%	22.000000	31610.000000	26874.000000	90.000000	
max	29.000000	39842.000000	39275.000000	178.000000	

	buyId	price
count	103156.000000	103156.000000
mean	2.513882	7.253868
std	1.790255	7.121550
min	0.000000	1.000000
25%	1.000000	2.000000
50%	2.000000	3.000000
75%	4.000000	10.000000

```
max          5.000000       20.000000
```

[18]:
```
merged_df.shape
```

[18]:
```
(103156, 13)
```