User_Data_Analysis

September 19, 2022

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

# read json file
def read():
    with open('users.json', 'r', encoding='utf-8') as f:
        data = [json.loads(line) for line in f]
        # Normalize semi-structured JSON data into a flat table.
        df = pd.json_normalize(data)
        # Convert unix timestamp into date format
        df['createdDate.$date'] = pd.to_datetime(df['createdDate.$date'],␣
 ↪unit='ms')
        df['lastLogin.$date'] = pd.to_datetime(df['lastLogin.$date'],␣
 ↪unit='ms')
    return df
# create dataframe
users = read()
```

[ ]:
```
# Print the first 5 records of the dataframe
users.head(5)
```

[ ]:

	active	role	signUpSource	state	_id.$oid	createdDate.$date	lastLogin.$date
0	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872
1	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872
2	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872
3	True	consumer	Email	WI	5ff1e1eacfcf6c399c274ae6	2021-01-03 15:25:30.554	2021-01-03 15:25:30.596999936
4	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	2021-01-03 15:25:37.857999872

[ ]:
```
# Looking at the 5 records you can see that the _id.$oid has duplicates
# Let print the dataframe shape
```

[ ]:
```
users.shape
```

[ ]: (495, 7)

[ ]:
```
# Check the index
users.index.values
```

[ ]:
```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,
       195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
       208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
       221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233,
       234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
       247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259,
       260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272,
       273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285,
       286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,
       299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,
       312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324,
       325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337,
       338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350,
       351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363,
       364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376,
       377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389,
       390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402,
       403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415,
       416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428,
       429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441,
       442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454,
       455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467,
       468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480,
       481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493,
       494], dtype=int64)
```

[ ]:
```
# Print information, shape and data type for the data frame
users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495 entries, 0 to 494
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   active             495 non-null    bool
 1   role               495 non-null    object
 2   signUpSource       447 non-null    object
 3   state              439 non-null    object
 4   _id.$oid           495 non-null    object
 5   createdDate.$date  495 non-null    datetime64[ns]
 6   lastLogin.$date    433 non-null    datetime64[ns]
dtypes: bool(1), datetime64[ns](2), object(4)
memory usage: 23.8+ KB
```

[ ]:
```
# Determine count of unique values for each column in the dataframe
users.nunique()
```

[ ]:
```
active                2
role                  2
signUpSource          2
state                 8
_id.$oid            212
createdDate.$date   212
lastLogin.$date     172
dtype: int64
```

[ ]:
```
# This tells me that only 212 of the 495 records are unique
# and this users data contains duplicates.
```