Washington State Cannabis Datasets Nov 2021

Dirty Data Discussion: Data types

Defining Washington State Leaf Data Systems dataset fields. (pandas.DataFrame.dtypes)

Author: Candy O’Sullivan (Sutherland) 4/14/2022
https://github.com/candy-o

Resources:

Links:

“Python Pandas DataFrame dtypes page”
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dtypes.html

“How pandas infers data types when parsing CSV files”
https://rushter.com/blog/pandas-data-type-inference/

How to Change Data Type for One or More Columns in Pandas DataFrame?
https://cmdlinetips.com/2018/09/how-to-change-data-type-for-one-or-more-columns-in-pandas-dataframe/

Data sources:

- WA State Traceability Data January 2018 - November 2021 Link(s)

  https://lcb.app.box.com/s/e89t59s0yb558tjoncjsid710oirqbgd?page=1

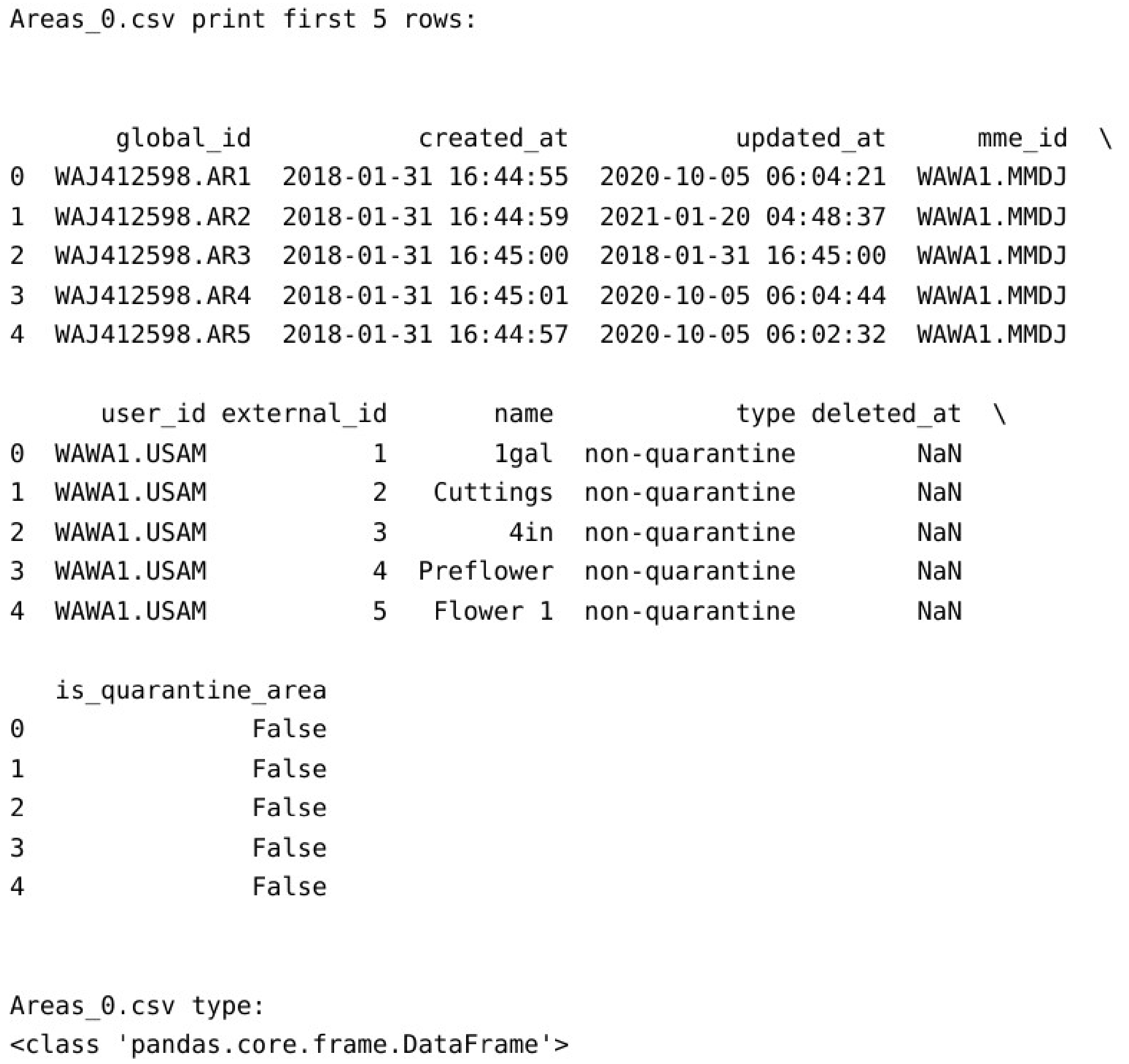
  https://lcb.app.box.com/s/e89t59s0yb558tjoncjsid710oirqbgd?page=2

Data Guide:

- Washington State Leaf Data Systems Guide Link

https://lcb.wa.gov/sites/default/files/publications/Marijuana/traceability/WALeafDataSystems_UserManual_v1.37.5_AddendumC_LicenseeUser.pdf

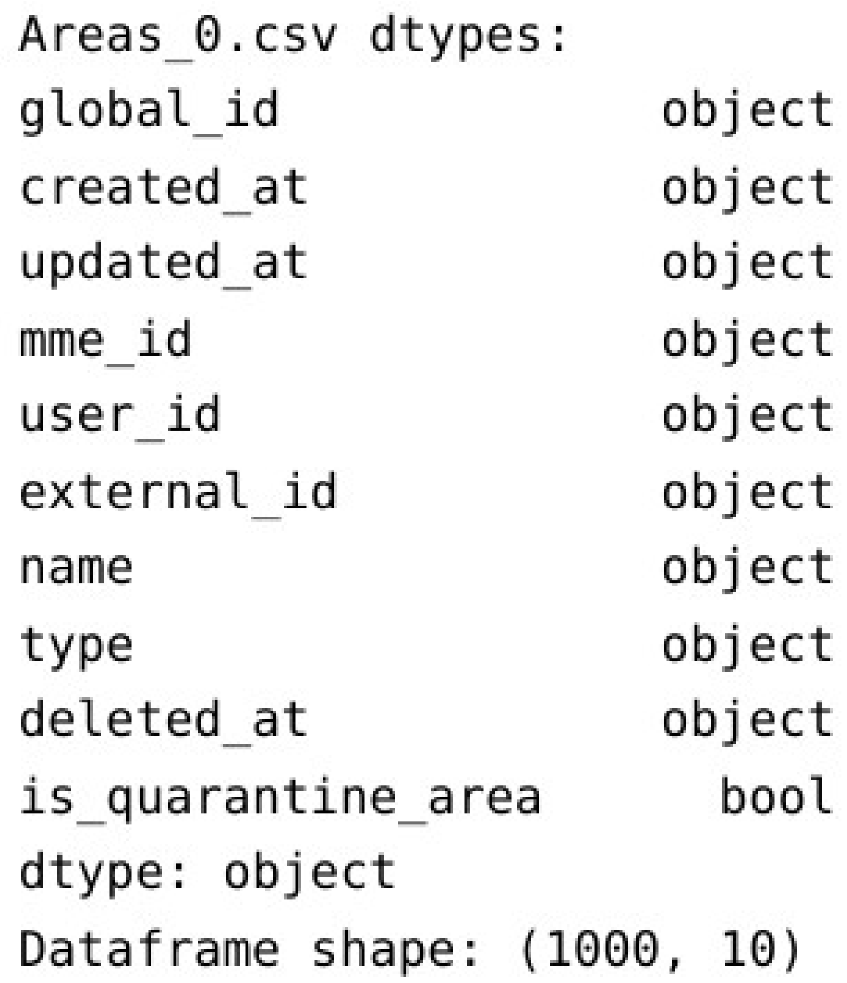
Areas_0.csv DataFrame fields



```
Areas_0.csv print first 5 rows:


      global_id           created_at           updated_at      mme_id  \
0  WAJ412598.AR1  2018-01-31 16:44:55  2020-10-05 06:04:21  WAWA1.MMDJ
1  WAJ412598.AR2  2018-01-31 16:44:59  2021-01-20 04:48:37  WAWA1.MMDJ
2  WAJ412598.AR3  2018-01-31 16:45:00  2018-01-31 16:45:00  WAWA1.MMDJ
3  WAJ412598.AR4  2018-01-31 16:45:01  2020-10-05 06:04:44  WAWA1.MMDJ
4  WAJ412598.AR5  2018-01-31 16:44:57  2020-10-05 06:02:32  WAWA1.MMDJ

      user_id external_id       name          type deleted_at  \
0  WAWA1.USAM           1       1gal  non-quarantine        NaN
1  WAWA1.USAM           2   Cuttings  non-quarantine        NaN
2  WAWA1.USAM           3        4in  non-quarantine        NaN
3  WAWA1.USAM           4  Preflower  non-quarantine        NaN
4  WAWA1.USAM           5   Flower 1  non-quarantine        NaN

   is_quarantine_area
0               False
1               False
2               False
3               False
4               False


Areas_0.csv type:
<class 'pandas.core.frame.DataFrame'>
```

Areas_0.csv DataFrame default dtypes

Pandas .read_csv tries to guess the type for each element of a column.



```
Areas_0.csv dtypes:
global_id             object
created_at            object
updated_at            object
mme_id                object
user_id               object
external_id           object
name                  object
type                  object
deleted_at            object
is_quarantine_area      bool
dtype: object
Dataframe shape: (1000, 10)
```

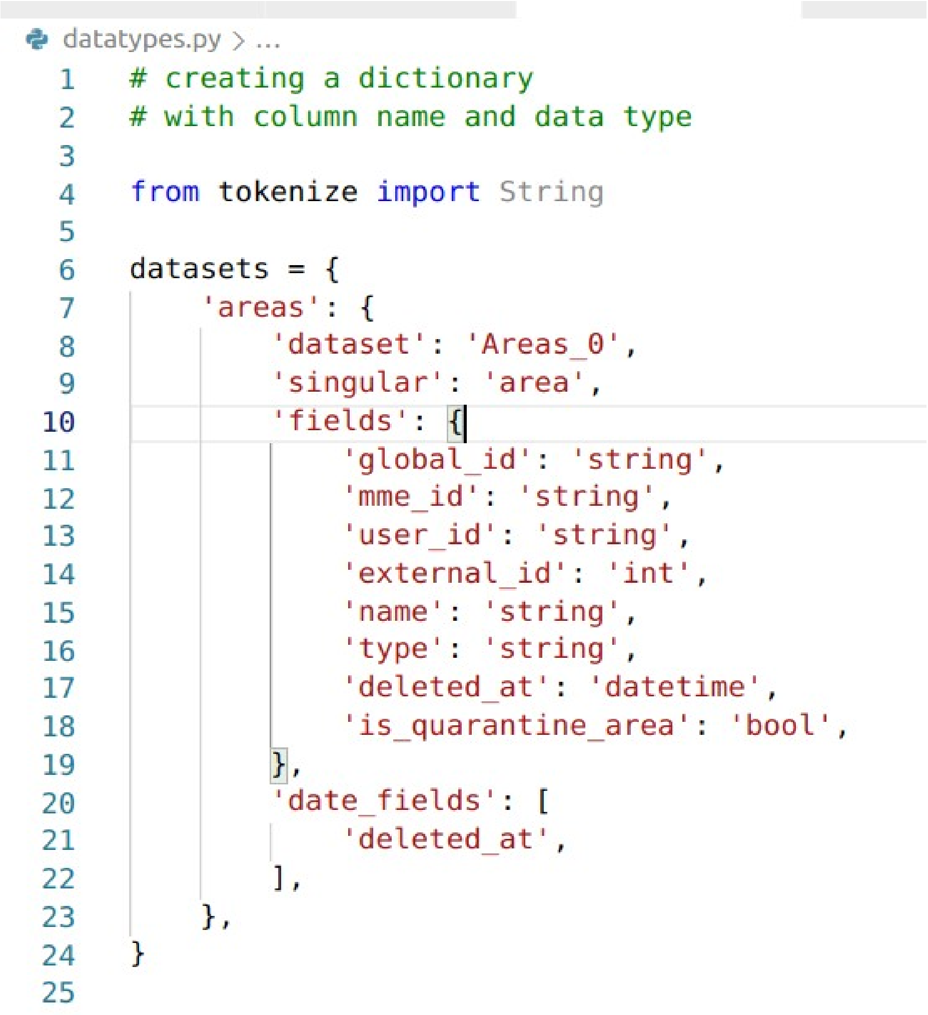
Pandas DataFrame with mixed types

Common error when creating a DataFrame with columns of mixed types, where Pandas .read_csv guesses the datatypes.

DtypeWarning: Columns () have mixed types. Specify dtype option on import or set low_memory=False.

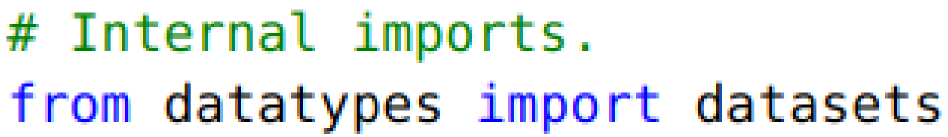
Create dictionary of datatypes

Eliminate Pandas guesswork overhead via a mapping dictionary with variable/column names as keys and data type you want as values.



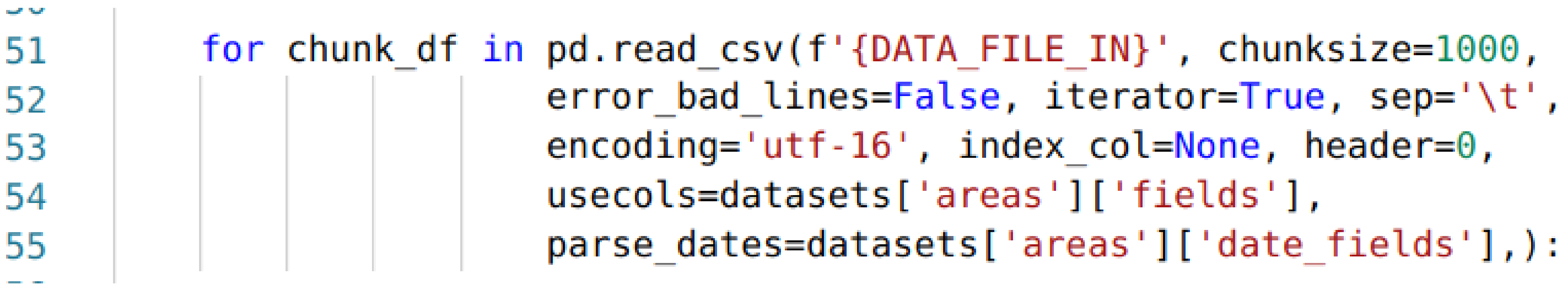
```python
# creating a dictionary
# with column name and data type

from tokenize import String

datasets = {
    'areas': {
        'dataset': 'Areas_0',
        'singular': 'area',
        'fields': {
            'global_id': 'string',
            'mme_id': 'string',
            'user_id': 'string',
            'external_id': 'int',
            'name': 'string',
            'type': 'string',
            'deleted_at': 'datetime',
            'is_quarantine_area': 'bool',
        },
        'date_fields': [
            'deleted_at',
        ],
    },
}

```

Import datatypes.py datasets into your code



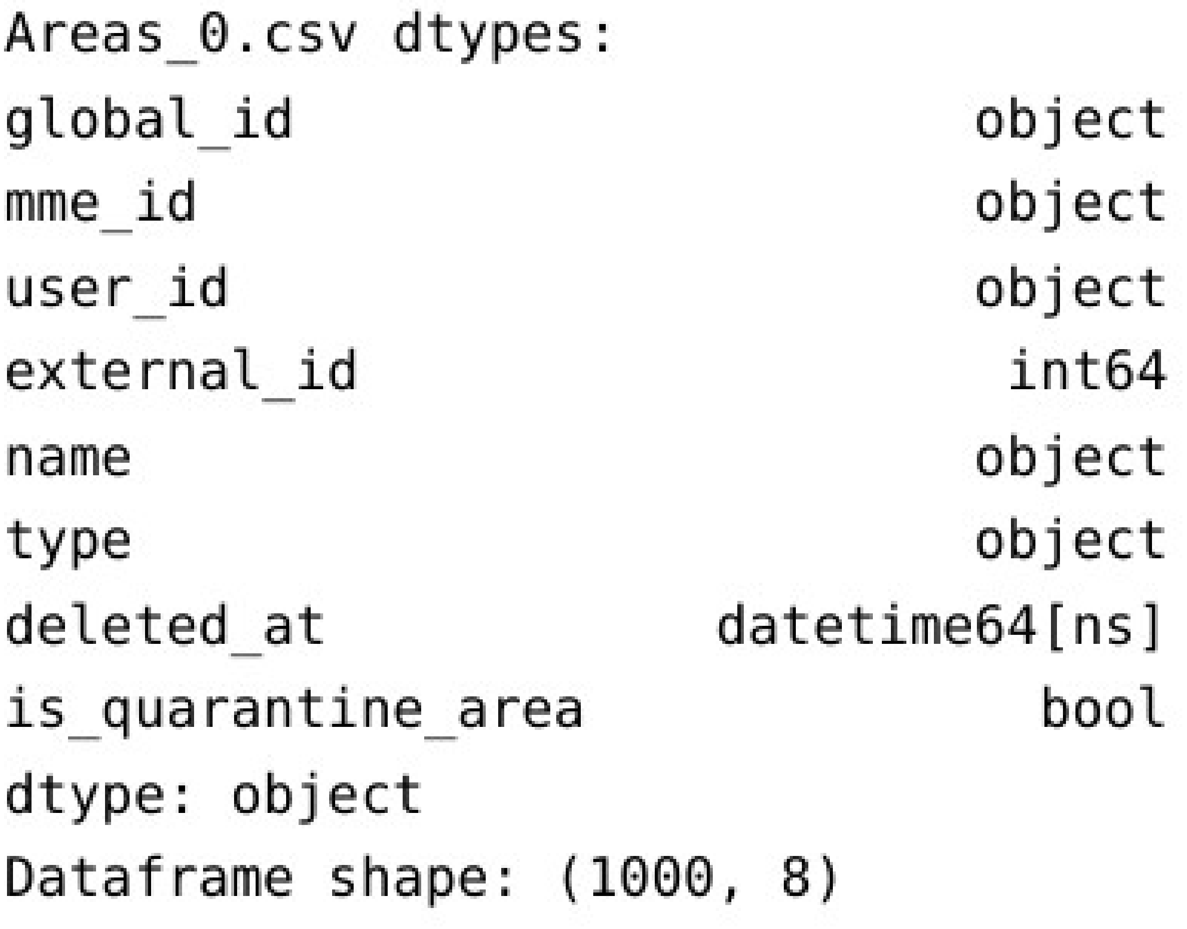
```python
# Internal imports.
from datatypes import datasets
```

pd.read_csv using imported“datasets”



```
    for chunk_df in pd.read_csv(f'{DATA_FILE_IN}', chunksize=1000,
                    error_bad_lines=False, iterator=True, sep='\t',
                    encoding='utf-16', index_col=None, header=0,
                    usecols=datasets['areas']['fields'],
                    parse_dates=datasets['areas']['date_fields'],):
```

Areas_0.csv DataFrame new dtypes



```
Areas_0.csv dtypes:
global_id                     object
mme_id                        object
user_id                       object
external_id                    int64
name                          object
type                          object
deleted_at            datetime64[ns]
is_quarantine_area              bool
dtype: object
Dataframe shape: (1000, 8)
```

Pandas stores strings as objects