Exploring Data with Python

Getting a first look at your datasets…

Exploring Data

- Spend time with your dataset:
    - Understand where it came from - can you live with the assumptions the data collectors made?
    - Look at the data
    - Plot the data
    - Where are there missing values? Inconsistencies? Anomalies?
- Clean your data, find better datasets, get more data

The Pandas Library

- Python has an R-like library for data analysis: Pandas
- http://pandas.pydata.org/
- "import pandas as pd"

Reading Files

- read_csv
- read_excel
- read_sql <- reads database files
- read_json
- read_html <- reads tables from HTML pages
- read_stata <- reads .dat files
- read_clipboard <- reads from your PC's clipboard

For example…

- import pandas as pd
- df = pd.read_stata('LSMS Data/AG_SEC12A.dta')

- Reads World Bank LSMS survey data into a “dataframe”

AG_SEC12A.dta

```
1   7102 0100 0e00 1536 0000 0073 b175 ca13
2   0134 0000 0000 0000 0000 dd02 0000 0902
3   0000 4f13 016e 0000 0000 0000 0000 2000
4   cc00 ffff ffff 36b3 7302 80b1 7302 0000
5   0000 6aaa 7302 44c1 7d00 e4a6 0f02 60aa
6   7302 0000 0000 433a 5c55 7332 3720 4a75
7   6e20 3230 3132 2030 353a 3037 0010 fb19
8   fbfb fbfb fbfb fbfb fbfd fb79 325f 6868
9   6964 0038 0000 0000 0000 0000 0000 0000
10  0000 0000 0000 0000 0000 0000 736f 7572
11  6365 6964 0000 6300 0000 0000 0000 0000
```

What's in the DataFrame?

- len(df)
- df.dtypes
- df.describe()

```
>>> df.dtypes
y2_hhid          object
sourceid       category
ag12a_0b         object
ag12a_01       category
ag12a_02_1     category
ag12a_02_2     category
ag12a_02_3     category
ag12a_02_4     category
ag12a_02_5     category
ag12a_02_6     category
ag12a_03       category
ag12a_04       category
ag12a_05        float64
ag12a_06        float64
dtype: object
>>> df.describe()
             ag12a_05    ag12a_06
count       37.000000  398.000000
mean     11559.459459    2.113065
std      32714.883289    6.167664
min        300.000000    0.000000
25%       2000.000000    0.000000
50%       3500.000000    1.000000
75%      10000.000000    2.000000
max     200000.000000   99.000000
```

View the DataFrame's head/tail rows

- df.head()
- df.tail()
- df.head(3)
- df[10:20]

```
>>> df.head(3)
             y2_hhid                                sourceid  \
0  0101014002017101                 GOVERNMENT EXTENSION
1  0101014002017101                                     NGO
2  0101014002017101  COOPERATIVE/FARMER'S ASSOCIATION

                  ag12a_0b ag12a_01 ag12a_02_1 ag12a_02_2 ag12a_02_3  \
0                  SERIKALI      YES         YES          NO         YES
1                       NGO       NO         NaN         NaN         NaN
2  USHIRIKA / CHAMA CHA WAKU       NO         NaN         NaN         NaN

  ag12a_02_4 ag12a_02_5 ag12a_02_6 ag12a_03 ag12a_04  ag12a_05  ag12a_06
0         NO        YES        YES  AVERAGE        NO       NaN         1
1        NaN        NaN        NaN      NaN       NaN       NaN       NaN
2        NaN        NaN        NaN      NaN       NaN       NaN       NaN
```

Access individual columns

df[‘sourceid’]

df[[‘sourceid’,’ag12a_01','ag12a_02_2']]

df[df.ag12a_01 == ‘YES’]

df[(df.ag12a_01 == 'YES') & (df.ag12a_02_1 == 'NO')]

Check Unique Column Values

```
>>> for k in df.keys():
...   print("{}: {}".format(k, df[k].unique()))
...
y2_hhid: ['0101014002017101' '0101014002028401' '0101014002029701' ...,
 '5502018021005902' '5502018021005905' '5502018021006801']
sourceid: ['GOVERNMENT EXTENSION' 'NGO' "COOPERATIVE/FARMER'S ASSOCIATION"
 'LARGE SCALE FARMER' 'OTHER']
ag12a_0b: ['SERIKALI' 'NGO' 'USHIRIKA / CHAMA CHA WAKU' 'MKULIMA/MFUGAJI MKUBWA'
 'NYINGINE, TAJA' 'GOVERNMENT EXTENSION' "COOPERATIVE/FARMER'S ASSO"
 'LARGE SCALE FARMER' 'OTHER']
ag12a_01: ['YES' 'NO']
ag12a_02_1: ['YES' 'NO' nan]
ag12a_02_2: ['YES' 'NO' nan]
ag12a_02_3: ['YES' 'NO' nan]
ag12a_02_4: ['YES' 'NO' nan]
ag12a_02_5: ['YES' 'NO' nan]
ag12a_02_6: ['YES' 'NO' nan]
ag12a_03: ['GOOD' 'AVERAGE' 'BAD' nan]
ag12a_04: ['YES' 'NO' nan]
ag12a_05: [     nan    1000.    3500.    7500.   11200.   16000.  200000.    1500.
    5000.    1200.   20000.    3000.   10000.    2500.    8000.   12000.
    2000.   40000.   17000.    6000.     300.    4500.     500.]
ag12a_06: [  1.  nan   2.   0.  24.   4.  12.   3.   6.  40.  99.   5.  10.  30.  25.]
```