PANDAS

A POWERFUL DATA MANIPULATION TOOL

Jason A Myers / @jasonamyers / Emma

WHAT'S SO SPECIAL ABOUT PANDAS?

1. Tabular/Matrix
2. Data Flexibility
3. Data Manipulation
4. Time Series

INSTALLATION

```
pip install pandas

pip install pandas as pd
```

PANDAS DATA STRUCTURES

- Series - basically an ordered dict that can be named
- Dataframe - A labeled two dimensional datatype

SERIES

```python
import pandas as pd

cookies = pd.Series(
    [
        'Chocolate Chip,'
        'Peanut Butter,'
        'Ginger Molasses,'
        'Oatmeal Raisin,'
        'Sugar',
        'Oreo',
    ]
)
```

WHAT DOES IT LOOK LIKE?

```
0     Chocolate Chip
1      Peanut Butter
2    Ginger Molasses
3     Oatmeal Raisin
4              Sugar
5               Oreo
dtype: object
```

PROPERTIES

```
>>> cookies.values

    array(['Chocolate Chip', 'Peanut Butter', 'Ginger Molasses',
            'Oatmeal Raisin', 'Sugar', 'Oreo'], dtype=object)

>>> cookies.index

    Int64Index([0, 1, 2, 3, 4, 5], dtype='int64')
```

SPECIFYING THE INDEX

```
cookies = pd.Series([12, 10, 8, 6, 4, 2], index=['Chocolate Chip',
    'Peanut Butter',
    'Ginger Molasses',
    'Oatmeal Raisin',
    'Sugar',
    'Powder Sugar'
])
```

INDEXED SERIES

```
Chocolate Chip     12
Peanut Butter      10
Ginger Molasses     8
Oatmeal Raisin      6
Sugar               4
Powder Sugar        2
dtype: int64
```

NAMING THE VALUES AND INDEXES

```
>>> cookies.name = 'counts'
>>> cookies.index.name = 'type'

    type
    Chocolate Chip     12
    Peanut Butter      10
    Ginger Molasses     8
    Oatmeal Raisin      6
    Sugar               4
    Powder Sugar        2
    Name: counts, dtype: int64
```

ACCESSING ELEMENTS

```
>>> cookies[[name.endswith('Sugar') for name in cookies.index]]

    Sugar                 4
    Powder Sugar          2
    dtype: int64

>>> cookies[cookies > 10]

    Chocolate Chip       12
    Name: counts, dtype: int64
```

DATAFRAMES

```
df = pd.DataFrame({
    'count': [12, 10, 8, 6, 2, 2, 2],
    'type': ['Chocolate Chip', 'Peanut Butter', 'Ginger Molasses', 'Oatmeal R
    'owner': ['Jason', 'Jason', 'Jason', 'Jason', 'Jason', 'Jason', 'Marvin']
})
```

```
   count   owner              type
0     12   Jason    Chocolate Chip
1     10   Jason     Peanut Butter
2      8   Jason  Ginger Molasses
3      6   Jason    Oatmeal Raisin
4      2   Jason             Sugar
5      2   Jason      Powder Sugar
6      2  Marvin             Sugar
```

ACCESSING COLUMNS

```
>>> df['type']

0     Chocolate Chip
1      Peanut Butter
2    Ginger Molasses
3     Oatmeal Raisin
4              Sugar
5       Powder Sugar
6              Sugar
Name: type, dtype: object
```

ACCESSING ROWS

```
>>> df.loc[2]

    count                     8
    owner                 Jason
    type       Ginger Molasses
    Name: 2, dtype: object
```

SLICING ROWS

```
>>> df.loc[2:5]
       count  owner             type
    2      8  Jason  Ginger Molasses
    3      6  Jason   Oatmeal Raisin
    4      2  Jason            Sugar
    5      2  Jason     Powder Sugar
```

PIVOTING

```
>>> df.loc[3:4].T
                         3      4
    count                6      2
    owner            Jason  Jason
    type   Oatmeal Raisin  Sugar
```

GROUPING

```
>>> df.groupby('owner').sum()

    count
    owner
    Jason      40
    Marvin      2
```

```
>>> df.groupby(['type','owner']).sum()

                               count
    type              owner
    Chocolate Chip    Jason       12
    Ginger Molasses   Jason        8
    Oatmeal Raisin    Jason        6
    Peanut Butter     Jason       10
    Powder Sugar      Jason        2
    Sugar             Jason        2
                      Marvin       2
```

RENAMING COLUMNS

```
>>> g_sum = df.groupby(['type']).sum()
>>> g_sum.columns = ['Total']

                 Total
    sum
    Chocolate Chip      12
    Ginger Molasses      8
    Oatmeal Raisin       6
    Peanut Butter       10
    Powder Sugar         2
    Sugar                4
```

PIVOT TABLES

```
>>> pd.pivot_table(df, values='count', index=['type'], columns=['owner'])

      Owner             Jason  Marvin
      type
      Chocolate Chip       12     NaN
      Ginger Molasses       8     NaN
      Oatmeal Raisin        6     NaN
      Peanut Butter        10     NaN
      Powder Sugar          2     NaN
      Sugar                 2       2
```

JOINING

```
>>> df = pivot_t.join(g_sum)
>>> df.fillna(0, inplace=True)

                      Jason  Marvin  Total
    type
    Chocolate Chip       12       0     12
    Ginger Molasses       8       0      8
    Oatmeal Raisin        6       0      6
    Peanut Butter        10       0     10
    Powder Sugar          2       0      2
    Sugar                 2       2      4
```