Pandas

Python Data Analysis Library

Developed in 2008 Open sourced 2009

AQR Capital Management a global investment firm to handle high performance quantitative analysis of financial data.

http://pandas.pydata.org/

It’s also included in Anaconda

https://www.continuum.io/downloads

Why Pandas?

It's fast, handles all the common data file formats, built in statistical tools are great but are being shifted to the stats library

Automatic data alignment, add in missing dates, data

Easy to merge and combine datasets

Easy to saved cleaned datasets to disk

Saved datasets can be imported into your favorite tools for data processing

3 main data structures

Series -> 1D series, usually time, can use other indexes

DataFrame -> 2D rows as records, columns as features

Panel -> 3D

IO Tools

Pandas can read and write:

CSV, HTML, Excel, HDF5, Feature, Msgpack, Stata, SAS, Pickle, SQL, Google Big Query

```
# read file into dataframe or series

data = pandas.read_csv("fileName.csv")
```

Get basic description of data

```
s = pd.Series([1, 2, 3])
s.describe()
count    3.0
mean     2.0
std      1.0
min      1.0
25%      1.5
50%      2.0
75%      2.5
max      3.0
```

Handle missing data

Easily count, delete rows with NaN

data = data.dropna()

Fill in NaN with interpolation, mean, backward and forward values, custom calculations using .apply

data['bonds'] = data['bonds'].interpolate()

Time Series Missing Dates

Can reindex time series that are missing dates:

```
# backward fill - missing business days

data['gdp'] = data['gdp'].resample("B").bfill()
```

Data can be grouped for calculations

```
# create new dataframe with data mean by month

month_totals = data.resample('M').mean()

# create array with count of males and females

total_by_sex = data.groupby('Sex').count()

# can use multiple columns (features)

total_by_age_sex = data.groupby(['Age', 'Sex']).count()
```

Add new features

```
data['newFeature'] = data['otherFeature'] * 3
```

Add new features using functions

```
def get_title(name):
    title = re.search(' ([A-Za-z]+)\.', name)
    if title: return title.group(1))
    else: return 'None'
data['Title'] = data['Name'].apply(get_title)
```

Combine several data files

gold = pandas.read_csv('gold.csv')

silver = pandas.read_csv('silver.csv')

metals = pandas.concat([gold, silver], keys='date')

Can also use basic sql commands for joins, merges etc

Demos

https://github.com/timestocome/Pandas-demos

DateMatching: Download several stock market indicators and create a dataframe matching up the dates for each and filling in missing values

Titanic: Read in data for Titanic problem, clean it up, break categories into one hot vectors, parse names into titles only for use as categorical data

https://github.com/timestocome/StockMarketData

Cleanup downloaded stock indexes