11. Python Pandas

3ikakke

Monday 28th March 2022

Outline

- Objectives
- First Steps With Data
- Unique Identifiers
- Data Manipulation
- Indexing and Slicing in Pandas
- Selecting columns
- Filtering rows
- Arranging DataFrame
- Combining DataFrames
- Long vs Wide Data
- Review of objectives
- Q&A
- Gist of the day

Objectives

- Understand data manipulation with pandas
  - Selecting columns
  - Filtering rows
  - Arranging data
  - Creating new variables (analytical variables)
  - Comining datasets
- Understanding Unique Identifiers
- Understanding Long vs Wide datasets

First Steps With Data

- When you obtain a new dataset you want:
  - Context
  - Documentation - Data Dictionary or Code Book
  - High level summary of the data - Observations and Variables
- What am I supposed to do with the data?
- Answering this informs:
  - Data Cleaning
  - Data preparation/pre-processing
- All of these have to do with some data maipulation

Unique Identifiers

- Unique identifiers are as the name implies some identifier that is unique to each 'individual' in a dataset
- Examples may include things like Serial number, NIN, BVN, email, phone number etc
- Names are not unique!!!
- In datasets where each record has a unique identifier that identiifer is referred to as a primary key
- I datasets where unique identifiers appear in more than one record they are referred to as foreign keys

Data Manipulation

- Selecting variables
- Filtering rows
- Arranging data by one or more variables
- Renaming variables
- Combining datasets
  - Joining
  - Merging
- Aggreggating**
- Converting from long to wide data**
- Converting from wide to long data**
- Filtering with time**

Indexing in Pandas DataFrames

- Pandas dataframes are based off NumPy matrices (2 dimentional arrays) and as such can be indexed using 2 numbers
- Pandas DataFrames can be sliced
  - Directly
  - Using the iloc (index location)
  - Using loc (named location) and using Boolean conditions
- NumPy introduces a new pair of operators for comparison (`&`, and `|`)
- NumPy comparison operators also impose a requirement that items to be compared must be wraped in brackets

Indexing and Slicing in Pandas

```
#remember to start by importing pandas
import pandas as pd
dataset = pd.read_csv("dataset.csv")
print(dataset.info())
print(dataset.shape)
print(dataset.columns)

#Direct indexing
print(dataset[['var', 'var2', 'var3']])
print(dataset[['var']])
print(dataset['var']) #if a single variable
print(dataset.var) #if a single variable that has no space or special characters in its name

print(dataset[['var', 'var2', 'var3']]) #this is a mini dataframe
print(dataset[['var']])[0]
print(dataset['var'])[1]
print(dataset.var)[2]
```

Indexing and Slicing in Pandas using iloc

```
print(dataset.iloc[2, 3])
print(dataset.iloc[1:3, 4:12])
```

Indexing and Slicing in Pandas using loc

```
print(dataset.loc[['var', 'var2', 'var3']])
print(dataset.loc[['var']])
print(dataset.loc['var'])
```

Selecting columns

- SQL syntax:

```
SELECT * FROM dataset
```

-Pandas

```
print(dataset)
```

- SQL syntax:

```
SELECT var, var2 FROM dataset
```

-Pandas

```
print(dataset[['var', 'var2']])
```

- SQL syntax:

```
SELECT var, var2 AS col2 FROM dataset
```

-Pandas

```
print(dataset[['var', 'var2']].rename(columns={'var2': 'col2'}))
```

Filtering rows

- SQL syntax:

```
SELECT *
FROM dataset
WHERE var = 'value'
```

-Pandas

```
print(dataset.loc[dataset['var'] == 'value'])
```

- SQL syntax:

```
SELECT var, var2, var3
FROM dataset
WHERE var = 'value'
```

-Pandas

```
print(dataset.loc[dataset['var'] == 'value', ['var', 'var2', 'var3']])
```

- SQL syntax:

```
SELECT *
FROM dataset
WHERE var IN ('value', 'value2', 'value3')
```

-Pandas

```
print(dataset.loc[dataset['var'].isin('value', 'value2', 'value3')])
```

- SQL syntax:

```sql
SELECT *
FROM dataset
WHERE var LIKE '%value%'
```

-Pandas

```
print(dataset.loc[dataset['var'].str.contains('value')])
```

- SQL syntax:

```sql
SELECT *
FROM dataset
WHERE var = 'value' AND var2 = 'value2'
```

-Pandas

```
print(dataset.loc[(dataset['var']=='value') & (dataset['var2']=='value2')])
```

Arranging DataFrame

- SQL

```sql
SELECT *
FROM dataset
ORDER BY var
```

- Pandas

```
print(dataset.sort_values('var'))
```

- SQL

```sql
SELECT *
FROM dataset
ORDER BY var DESC
```

- Pandas

```
print(dataset.sort_values('var', ascending=False))
```

- SQL

```sql
SELECT *
FROM dataset
ORDER BY var, var2
```

- Pandas

```
print(dataset.sort_values(['var', 'var2']))
print(dataset.sort_values(['var', 'var2'], ascending=[True, False])
```

Creating new variables

```sql
SELECT var, value AS new_var
FROM dataset
```

```
dataset['new_var'] = 'value'
```

```sql
SELECT var, var2+var3 AS new_var
FROM dataset
```

```
dataset['new_var'] = dataset.loc['var2'] + dataset.loc['var3']
```

```sql
SELECT
    var
    ,CASE var
        WHEN 'value' THEN 'new_value'
        WHEN 'value2' THEN 'new_value2'
        ELSE 'default_val'
    END AS new_var
FROM dataset
```

```
dataset['new_var'] = 'default_val'
dataset.loc[dataset['var'] == 'value', 'var'] = 'new_value'
dataset.loc[dataset['var'] == 'value2', 'var'] = 'new_value2'
```

Combining DataFrames

- Sometimes you want to combine 2 data sets side by side to complete details of individual records (horizontal)
- Othertimes you want to combine 2 or more data sets to add new records (vertical)
- We will see both approaches

Combining (merging/joining)

- SQL

```sql
SELECT *
FROM dataset AS a
    LEFT JOIN dataset2 AS b ON a.id=b.id
```

- Pandas

```
dataset1.merge(dataset2, left_on=id, right_on=id)
```

Combining (concatenating/union)

- SQL

```sql
SELECT *
FROM dataset
UNION
SELECT *
FROM dataset2
```

- Pandas

```
pd.concat([dataset1, dataset2])
```

Long vs Wide Data

- Wide data has records for unique 'individuals'
- Long data has multiple values for 'individuals'
  - Example the weight of people measured over time
  - Some variables are usually fixed incuding a unique identifier
  - Repeated measures usually change over time
  - There is usually a variable to indicate the time of the measure
- In future we will explore converting long data to wide and vice versa

Review of objectives

Q&A

Gist of the day

- Get the pdf version of todays conversation
- Get the gist
- The Jupyter Notebook will be uploaded on the Slack channel

We are done!