# Chapter 05 Getting Started with pandas

* **Pandas is designed for working with tabular or heterogeneous data.**
* **NumPy, by contrast, is best suited for working with homogeneous numerical array data.**

**import** **pandas** **as** **pd**

* 1. **Pandas Data Structures: Series and DataFrame**

**Series**

* **One-dimensional array-like object, containing a sequence of values**

**obj = pd.Series([4, 7, -5, 3])**

**obj2 = pd.Series([4, 7, -5, 3], index=['d', 'b', 'a', 'c'])**

**obj.values**

**obj.index**

* **Get single value**

**obj2['a']**

**DataFrame**

* **A DataFrame represents a rectangular table of data.**
* **Contains an ordered collection of columns, each of which can be a different value type (numeric, string, boolean, etc.).**

data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada', 'Nevada'],

'year': [2000, 2001, 2002, 2001, 2002, 2003],

'pop': [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]}

* **Create a DataFrame from dictionary**

**df = pd.DataFrame(data)**

* **Create a DataFrame by reading data from a csv file**

**df = pd.read\_csv('data/e05GameSalary.csv')**

* **Display number of rows and columns**

**df.shape**

* **Display the first n rows(by default n=5).**

**df.head(n)**

* **Display the last n rows (by default n=5).**

**df.tail()**

* **Return DataFrame column names, which is a Series**

**df.columns**

* 1. **DataFrame Essential Functionality**
* **Select rows/columns**

**Use loc to selects single row or subset of rows/columns**

Example:

Df.loc[5]

df.loc[0:5]

df.loc[10:15]

#Selects single column or subset of columns by label

df.loc[:, 'Last Name']

df.loc[:, ['Last Name','Title']]

* **Selection one column, return one column of DataFrame, which is a Series**

**# Example**

**df['Last Name']**

**df['state'] vs. df.state**

* **Select more columns, use double [[]]**

**data[['Last Name', 'First Name', 'Title']]**

* **Filtering**

**# Example**

**data[data.Title=='Programmer']**

**# vs.**

**data[data['Title']=='Programmer']**

**# Use & for AND logic**

**# Use | for OR logic**

**res = df[ (df.Title=='Senior Programmer') & (df.Gender=='F')]**

**# Show how many people satisfied the condition**

**res.ID.count()**

**# or**

**len(data[data.Performance=='Excellent'])**

**Example:**

**# How many transaction in Living Room Department and Trans\_Type is Promotion**

**Step 1: Filtering**

**res = s[(s.Department=='Living Room') &**

**(s.Trans\_Type =='Promotion')]**

**Step 2: Select Salary column**

**len(res)**

**res.ID.count()**

**Include two steps:**

**s = sales**

**res = s[(s.Department=='Living Room') & (s.Trans\_Type =='Promotion')]**

* **Sorting**

Sort from smallest to largest: **ascending=True**

Sort from largest to smallest: **ascending=False**

**#Sort Index**

**df.sort\_index(ascending=True)**

**#Sort Values**

#Sort one column

**Example:**

**df.sort\_values(by='Salary', ascending=False)**

**df.sort\_values(by='Salary', ascending=True)**

# Show top three highest salary

**df.sort\_values(by='Salary', ascending=False).head（3）**

#Sort more columns

**#Example:**

**df.sort\_values(by=['Gender','Salary'],ascending = False)**

e05GameSalary.csv

