Data Manipulation with Pandas

Pandas 라이브러리에서 제공하는데이터 구조를 알아본다. Pandas 는 NumPy 를 기반으로 만들어진 새로운 패키지로서 DataFrame 이라는 효율적인 자료구조를 제공한다. DataFrame 은 근본적으로 행과 열 레이블이 부착된 다차원 배열로서, 여러가지 타입의 데이터를 가질 수 있으며 데이터 누락도 허용된다. Pandas 는 레이블이 붙은 데이터를 위한 편리한 스토리지 인터페이스를 제공할 뿐만 아니라 데이터 베이스프레임워크와 스프레드시트 프로그램 사용자에게 익숙한 강력한 데이터 연산을 구현한다.

NumPy 의 ndarray 데이터 구조는 잘 정리된데이터 타입의 핵심적 기능을 제공하지만 유연성이 더 필요하고(데이터에 레이블을 붙이거나 누락된 데이터로 작업하는 등)요소 단위의 브로드캐스팅에 잘 매핑되지 않는 연산(그룹화, 피벗 등)을 하고자 하는 경우에는 한계가 있다

Pandasdptjsms 특히 series 와 DataFrame 객체는 NumPy 배열 구조를 기반으로 하며데이터 과학자의 시간을 대부분 잡아먹는 "데이터 먼징(datamunging)" 작업을 효율적으로 수행할 수 있게 된다. .

Pandas 설치 및 사용

시스템에 Pandas 를 설치하려면 먼저 NumPy 가 설치돼 있어야 하고, 소스에서 라이브러리를 빌드하려면 Pandas 가 구축된 C 와 Cython 소스를 컴파일하기 위한 적절한 도구가 필요하다. 이 설치와 관련한 자세한 내용은 Pandas 공식 문서(pandas.pydata.org)를 참고하자. 서문에서 설명한 권고사항에 따라 아나콘다(Anaconda) 스택을 사용한 독자라면 이미 Pandas 가 설치돼 있을 것이다.

Pandas 를 설치하고 나면 그것을 임포트해서 버전을 확인할 수 있다

```
In [1]: import pandas
        pandas.__version__
Out[1]: '0.18.1'
```

일반적으로 NumPy 를 np 라는 별칭으로 임포트하는 것처럼 Pandas 도 Pd 라는 별칭으로 임포트할 것이다:

```
In [2]: import pandas as pd
```

일반적으로 NumPy 를 np 라는 별칭으로 임포트하는 것처럼 Pandas 도 Pd 라는 별칭으로 임포트할 것이다

내장 문서를 기억하자

이번장을 읽으면서 Python 은 다양한 함수에 대한 문서뿐만 아니라 패키지의 내용을 신속하게 살펴볼 수 있는 기능을 제공한다는 사실을 잊지 말자

예를 들어 Pandas 네임스페이스의 모든 내용을 표시하려면 다음과 같이 입력한다.

```
In [3]: pd.<TAB>
```

그리고 내장된 Pandas 문서를 표시하려면 다음 명령어를 사용하면 된다.:

```
In [4]: pd?
```