KossLab
KOREA OPEN SOURCE SOFTWARE DEVELOPERS LAB
공개SW개발자 Lab

빠르고 유연한 데이터 처리를 위한 파이썬 라이브러리

Pandas

공개 SW 개발자 Lab 오픈소스프론티어 3 기 김영근

KossLab

pandas는 데이터를 쉽고 빠르게 다양한 형식으로 가공할 수 있는 풍부한 자료 구조와 함수를 제공하는 파이썬 라이브러리로, 파이썬을 활용한 데이터 분석에서 빠질 수 없는 필수 스택 중 하나다.

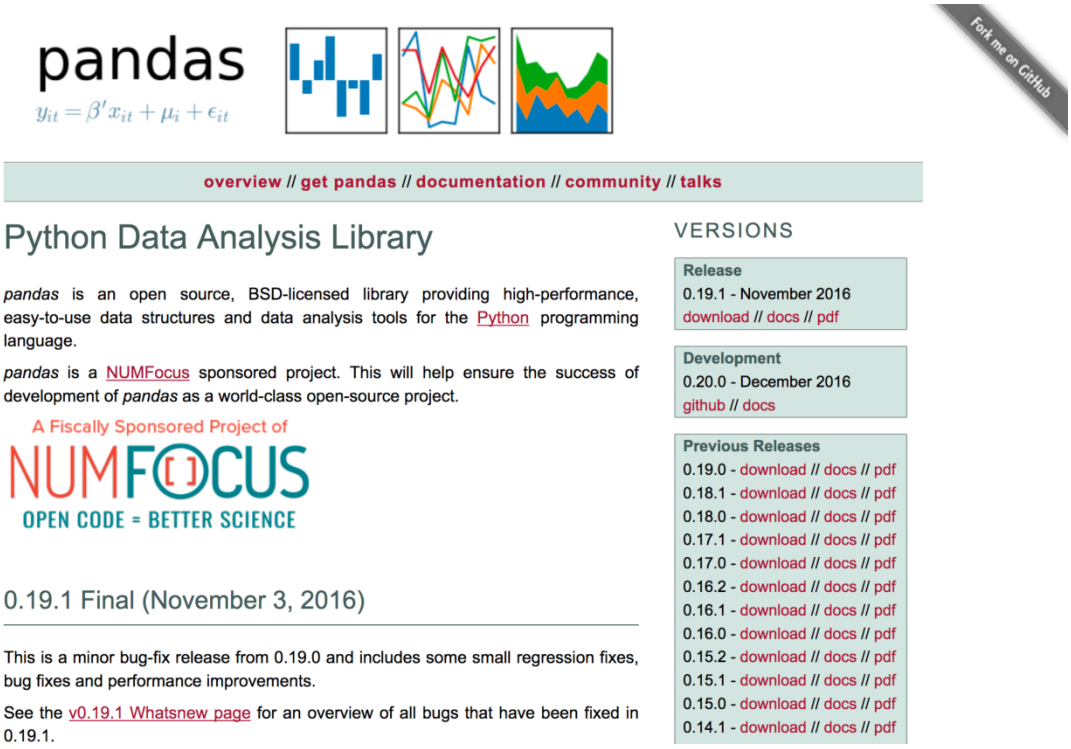
최근 불고 있는 코딩 열풍과 함께 데이터로부터 인사이트를 얻으려는 니즈가 결합되면서 파이썬은 주목받고 있으며 덩달아 pandas의 인기도 치솟고 있다. PyData, SciPy 같은 과학계산 커뮤니티는 물론이고 일반인들까지도 파이썬과 pandas를 접해보려는 시도가 크게 늘어났다.

이 글에서는 pandas 소개 및 프로젝트의 현재 상황과 앞으로의 계획을 다룬다.

프로젝트명	Pandas
개요	Flexible and powerful data analysis / manipulation library for Python
특징	직관적인 자료구조, BSD License
목표	빠르고 유연한 데이터 처리
기대효과	-
리퍼지토리	https://github.com/pandas-dev/pandas

[목차]

1 pandas 소개



[그림 1] http://pandas.pydata.org

pandas는 헤지 펀드에서 퀀트로 근무하고 있던 Wes McKinney가 업무에 바로 사용할 수 있는 적합한 도구가 없음에 답답함을 느끼고 직접 파이썬을 배우면서 만들기 시작한 라이브러리다. 본인의 전공은 수학이었고 프로그래밍에 익숙하지 않았기 때문에 초창기에는 CS를 전공한 같은 학교 친구인 Chang She와 함께 파이썬을 공부하면서 코드의 많은 부분을 작성하기 시작했으며 2008년부터 본격적으로 개발이 진행되었다.

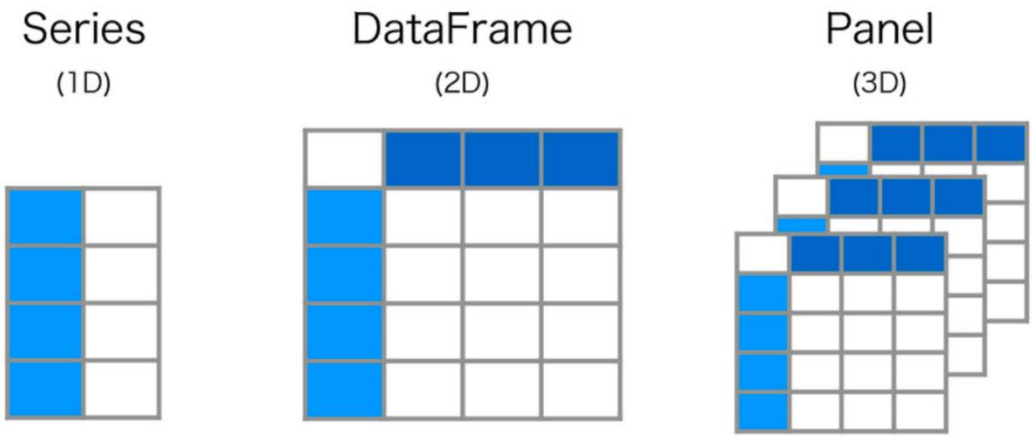
pandas는 직관적인 자료구조를 제공하여 실제 데이터 분석 업무 과정에서 80%의 시간을 차지하는 데이터 정제를 손쉽게 할 수 있도록 도와주며 간단한 통계 및 시각화를 위한 기능도 내장하고 있다. 필요한 경우 외부 라이브러리를 이용해서 복잡한 분석이나 좀 더 디테일한 시각화 작업을 하는 것도 용이하다.

성능을 위해 많은 부분을 C 또는 Cython으로 작성하였으며 내부적으로 NumPy를 사용하고 있는데, NumPy에서 제공하는 단일 데이터 타입의 N-차원 배열(ndarray)에 추가로 여러 가지 데이터 타입을 사용할 수 있는 기능이라던가, 인덱스뿐만 아니라 이름으로 데이터에 접근할 수 있는 기능을 제공하여 데이터를 보다 쉽게 담고 확인할 수 있도록 구현되어 있다.

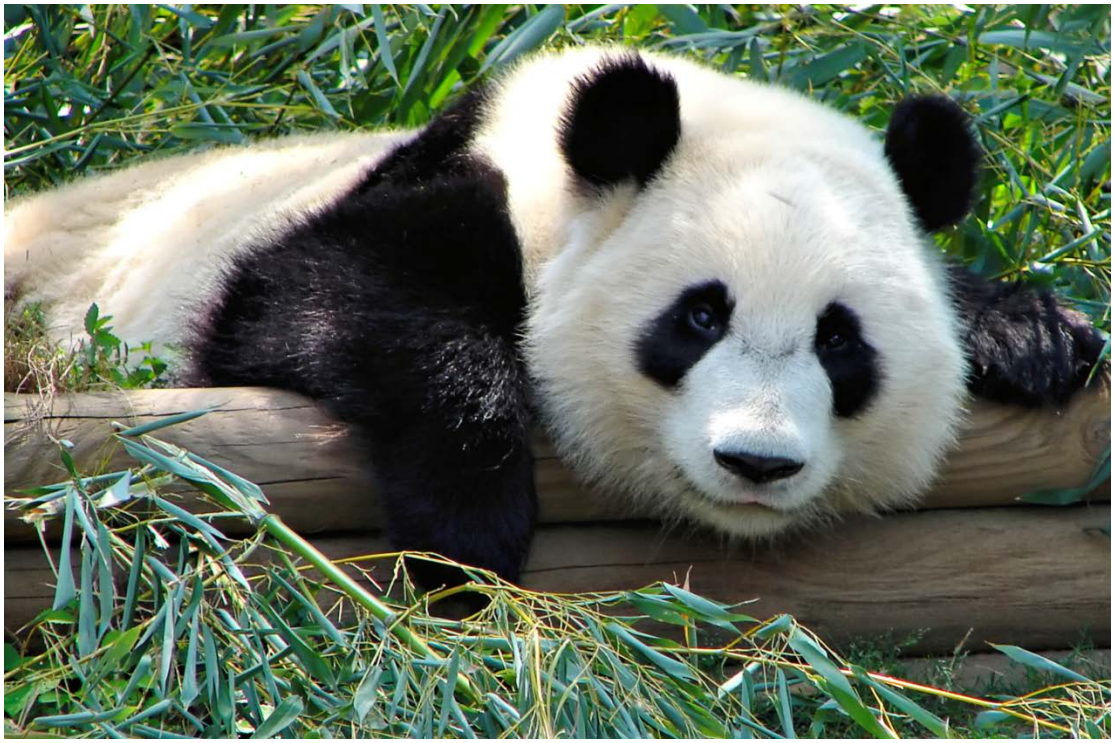
이런 기능은 모두 pandas의 기본 자료구조인 Series와 DataFrame을 통해 제공되며 빠른 처리를 위한 벡터 연산을 지원하고 Group-by 연산과 merge, join, concat 같은 데이터 재성형(Re-shaping) 편의 기능 역시 제공하고 있다. 다양한 외부 파일 포맷(csv, json, excel, SQL, msgpack, Google Big Query 등)의 읽기/쓰기를 제공하여 현실에서 만나게 되는 보편적인 파일에 담긴 데이터를 쉽게 불러오고 또 반대로 결과를 해당 포맷으로 저장하는 것도 가능하다.

pandas에 대해서 알아보려면 Series 와 DataFrame, 이 두 가지 자료 구조에 익숙해질 필요가 있다. Series는 일련의 객체를 담을 수 있는 1차원 배열 같은 자료 구조다(어떤 NumPy 자료형이라도 담을 수 있다). 그리고 인덱스에는 배열의 데이터에 연관된 이름을 가지고 있다. 가장 간단한 Series 객체는 배열로부터 생성할 수 있다.

DataFrame은 표 같은 스프레드시트 형식의 자료 구조로 여러 개의 칼럼이 있는데, 각 칼럼은 서로 다른 종류의 값(숫자, 문자열, 불리언 등)을 담을 수 있다. DataFrame은 로우와 칼럼에 대한 인덱스가 있는데, 이 DataFrame은 같은 인덱스를 가지는 Series 객체를 담고 있는 파이썬 사전으로 생각하면 편하다. R의 data.frame 같은 다른 DataFrame과 비슷한 자료 구조와 비교했을 때, DataFrame에서의 로우 연산과 칼럼 연산은 거의 대칭적으로 취급된다.



[그림 2] pandas의 자료구조



[그림 3] pandas와 관련 없음

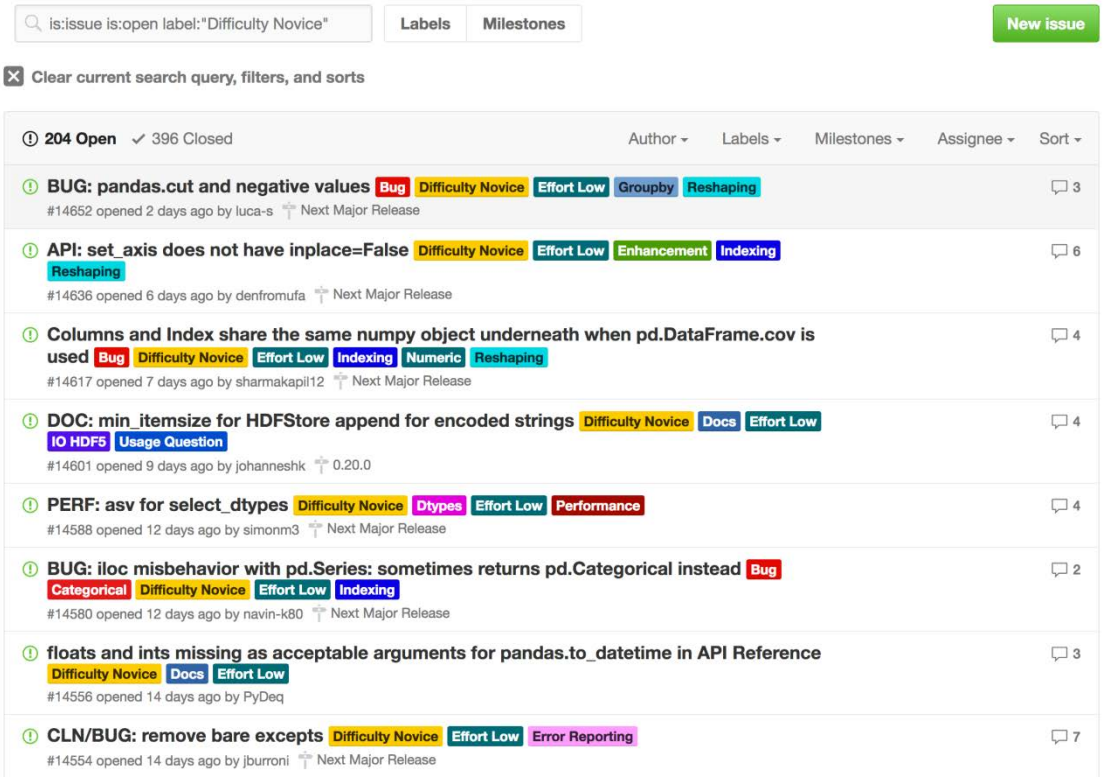
프로젝트의 이름인 pandas는 **PAN**el **DA**ta **S**ystem에서 따온 것이며 팬더와는 전혀 아무런 관련이 **없다**. BSD 라이선스를 채택하고 있으며 2016년 11월 현재, Github에서 7,449개의 스타를 얻었고 포크 횟수가 3000을 넘겼다. 100개 이상의 커밋을 보낸 코어 개발자는 11명이지만 전체 기여자는 640명을 넘어섰고, API 문서와 튜토리얼을 제공하는 공식 홈페이지(http://pandas.pydata.org)는 월간 순수 방문자 수가 5만에서 7만에 이를 정도로 인기 있는 프로젝트다.

버그 리포팅 및 이슈 관리는 Github에서 관리하고 있으며 최근에는 프로젝트가 커짐에 따라 기존에 pydata 그룹 아래에 있던 리퍼지토리를 pandas-dev 그룹으로 옮겼다.

개발에 대한 논의는 메일링과 Gitter 채널을 주로 이용하며 파이썬 개발 커뮤니티의 특징인 스프린트를 주기적으로 진행하고 있다. 스프린트는 파이썬 컨퍼런스인 파이콘의 숨은 메인 행사로, 파이썬 프로젝트 메인테이너와 사용자가 함께 모여 밀린 이슈를 처리하거나 밀도 높은 협업을 통해 큰 기능을 개발하는 자리이다. 또한, 스프린트는 해당 프로젝트에 기여하고 싶은 사용자들에게 기여 경험을 제공해주기 위한 목적도 있는데, 이를 통해 새로운 기여자를 발굴하고 프로젝트가 건강하게 유지될 수 있도록 하는 연료 역할을 하고 있다.

pandas 이슈 페이지에서는 모든 이슈에 대해서 카테고리와 주제뿐만 아니라 해당 이슈의 난이도와

소요 시간에 따라 상세한 라벨로 나눠서 관리하고 있으며 해결하기 쉬운 이슈는 새로운 기여자를 위해 남겨두고 기여 경험을 할 수 있도록 배려하고 있다.



[그림 4] pandas 프로젝트 이슈 페이지

2 pandas 현재

pandas는 버저닝을 보수적으로 해왔는데 상황에 따라 조금씩 연기되기도 하지만 기본적으로는 2개월마다 메이저 릴리즈를 원칙으로 하고 있으며 현재 최신 버전은 0.19.1이다. 다음 릴리즈는 0.20.0이 될 예정이지만, 1.0 릴리즈에 대한 논의(https://github.com/pandas-dev/pandas/issues/10000)가 이미 시작되었으며 1.0을 기점으로 1.x/2.x 브랜치를 나누어 유지보수와 새 버전 개발을 독립적으로 관리하게 된다.

현재 개발팀은 그동안 쌓인 기술 부채를 청산하지 않으면 안 될 상황이라 판단하고 있으며 1.x 개발 진행과 병행하여 기본이자 핵심 자료구조인 Series와 DataFrame 내부 구조에 대한 대대적인 재설계를 진행하고 있다.

현재까지의 버전은 코드가 제멋대로 늘어나서 20만 라인을 넘어섰다. 함수 개수만도 6,000개에 육

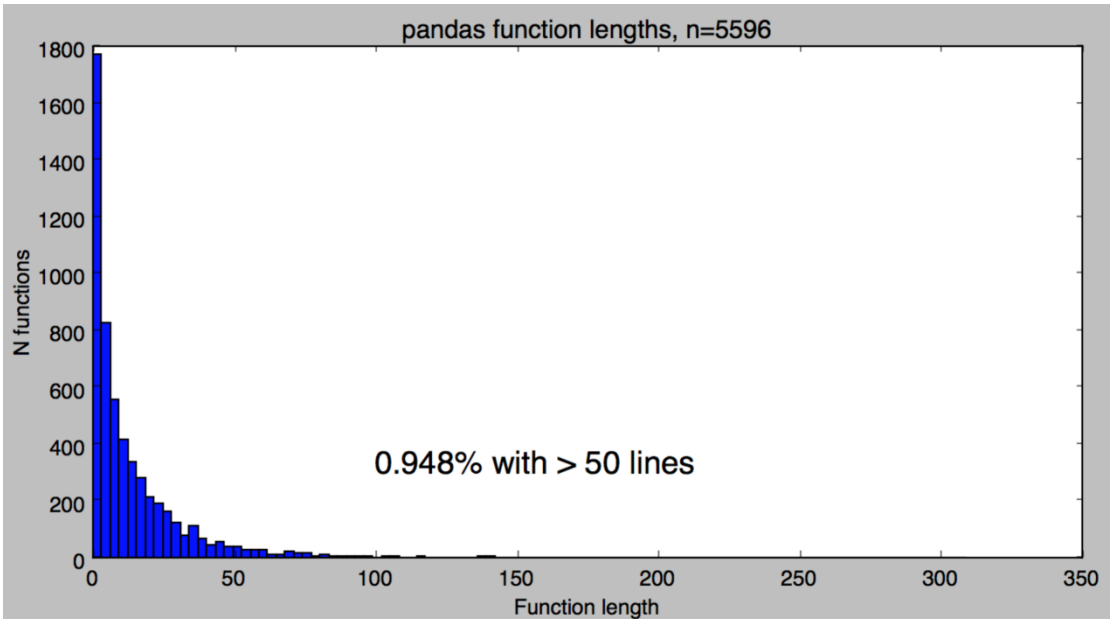
박하며 새로운 데이터 타입을 추가하기 어려운 상황이며, 특수한 상황을 제외하고는 여전히 단일 스레드로 동작하며 일부 코드는 Cython 수준에서 GIL을 해제하기도 하지만 여전히 많은 부분이 GIL로부터 자유롭지도 않다. 메모리 사용량을 예측하기 쉽지 않은 문제와 다른 시스템과 연동이 효율적이지 않게 구성되어 있는 등, 누적된 기술 부채로 인해 칼을 들지 않으면 안 되는 상황이다.

사실 이런 상황은 많은 오픈소스 프로젝트에서도 발생하는 일인데, 어떤 프로젝트가 좋은 아이디어와 어느 정도 쓸만한 퀄리티로 공개되면서 입소문을 탄다. 사용자가 늘고 컨트리뷰터도 늘면서 프로젝트는 호황을 누리게 되고 건강한 문화가 자리 잡으면서 원저자가 프로젝트에서 빠지게 되더라도(실제로 Wes를 비롯한 초기 개발자가 창업을 이유로 프로젝트를 한동안 떠나 있었다.) 다른 컨트리뷰터들이 프로젝트를 계속 이끌어 간다.

프로젝트가 점점 더 인기를 얻고 더 많은 개발자와 사용자층이 생기게 되고, 컨티넘애널리틱스 같은 오픈소스를 지원하는 회사에서 프로젝트 전담 개발자를 투입하여 프로젝트를 체계적으로 돕기 시작한다. 여기까지는 인기 오픈소스 프로젝트의 성공적인 성장 스토리처럼 들리지만 사실 그 이면에는 기술 부채가 쌓일 수밖에 없는 이유가 있다.

기업이 오픈소스에 개발자를 지원하는 이유는 여러 가지가 있겠으나 자사 또는 고객이 해당 프로젝트를 사용하고 있으면 버그 수정이나 사용자들의 요구사항이 제때 반영되지 않아서 곤란한 상황을 방지하기 위한 이유도 있다. 컨티넘애널리틱스의 경우 자체 개발하여 오픈소스로 공개한 훌륭한 프로젝트도 많이 가지고 있고 PyData 커뮤니티에 우호적인 기업으로, 위에서 얘기한 초기 개발자가 개인적인 이유로 프로젝트에 시간을 거의 쓰지 못하던 시기에 pandas 프로젝트를 거의 먹여 살렸다고 할 수 있을 정도로 많은 이바지를 했다.

하지만 기술 트렌드 변화에 따라 적절한 기술적인 의사 결정을 해야 하는 시기에 초기 개발자의 부재로 인해 설계 의도에 대한 커뮤니케이션을 충분히 할 수 없어서 코드가 계속 낡아가고 또, 주로 고객들의 요구 사항에 맞춰 급하게 수정을 해야 하는 기업의 오픈소스 개발자 입장에서는 충분한 고민을 하지 못하고 코드를 추가하다 보면 알게 모르게 작은 기술 부채들이 계속 쌓이게 된다.



[그림 5] pandas와 matplotlib을 이용해서 그린 pandas 함수 길이 분포

실제로 pandas 프로젝트 초기 시절에 Wes가 강하게 지키고자 했던 철학 중 하나는 함수를 작게 나눈다는 원칙이었는데 현재는 덩치가 큰 함수들이 상당수 출현하고 있다. (그래도 여전히 6,000여 함수 중에서 50줄을 넘어가는 함수는 1%가 되지 않는다.)

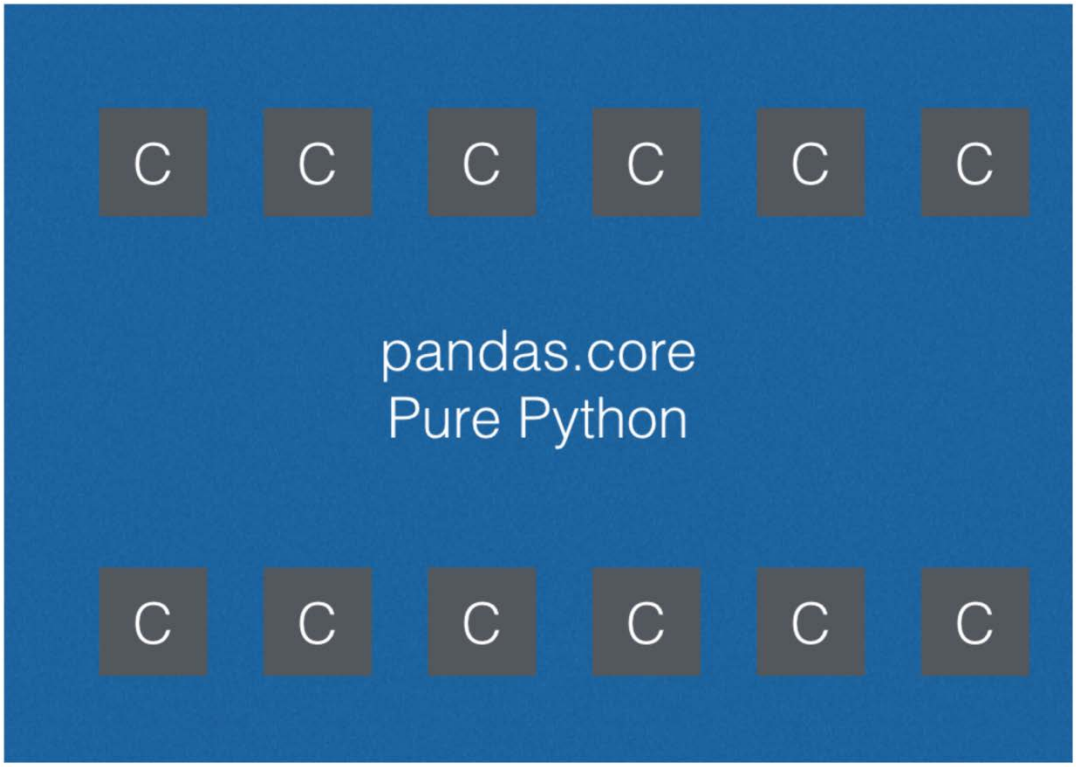
3 pandas 미래

pandas의 코드 베이스는 작성된 지 8년이 넘었다. 0.1 버전에서 겨우 1만 라인에 불과했던 소스 코드는 현재 20만 라인을 넘어섰고 기술 부채가 발목을 잡는 횟수가 눈에 띄게 늘었다. 이에 따라 현재 브랜치는 기능 개선보다는 버그 수정과 안정성에만 주력하여 향후 1.x 버전 릴리즈로 사용하기로 했고 새로운 2.0버전을 위한 설계가 진행 중이다.

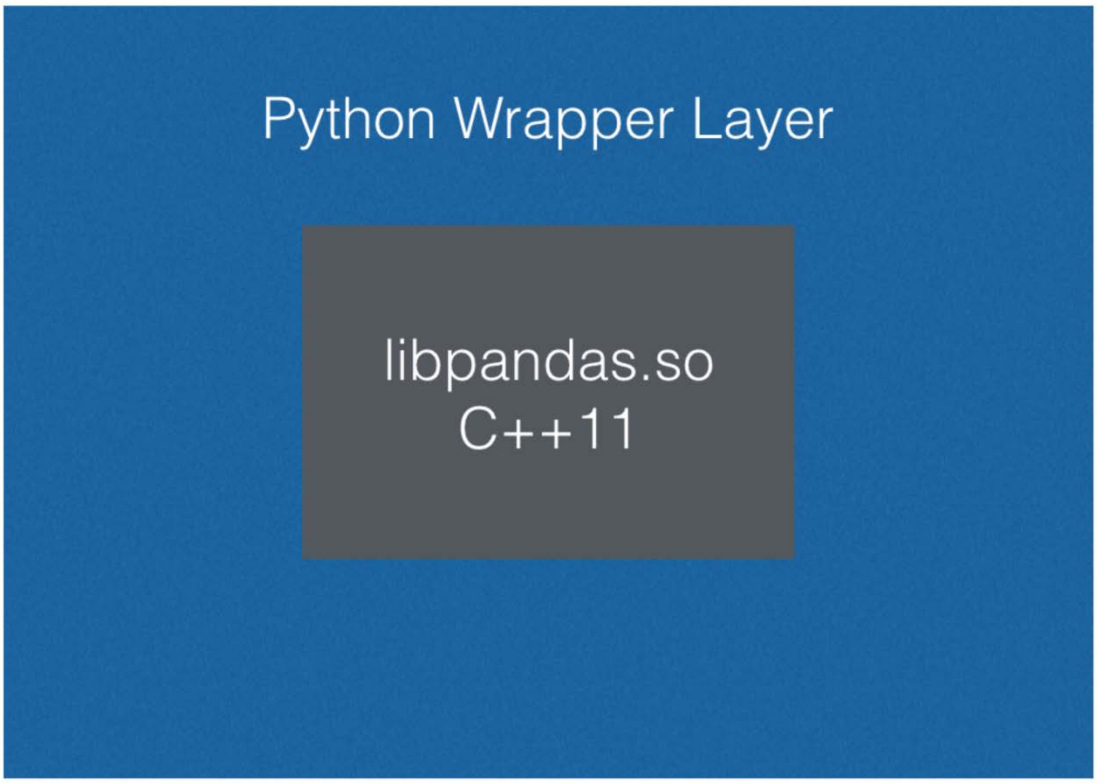
pandas를 처음 개발했을 때 가장 많이 어필했던 부분은 속도이다. Wes 스스로도 많은 pandas 관련 발표에서 기존 도구에 비해서 빠르다는 점을 강조했으나 2.0에서는 더 빠르게 동작하도록 성능 개선에 많은 노력을 기울일 예정이다. 또한, 멀티코어 같은 하드웨어 자원을 효율적으로 활용할 수 있도록 하는 개선 작업과 꾸준히 문제로 제기되었던 메모리 사용량을 예측할 수 없는 문제도 개선될 예정이다.

이를 위해 순수 파이썬으로 구현된 pandas.core 모듈에 기능별로 Cython 혹은 C 구현을 안고 있는 현재 상태에서 pandas 2.0부터는 코어 기능을 C++로 작성할 libpandas.so 라이브러리로 분리하고

이를 파이썬 wrapper 라이브러리로 감싸게 된다. libpandas는 다른 파이썬 라이브러리 개발자들을 위한 C/C++ API로 제공할 계획이며 이를 통해 pandas의 Series나 DataFrame의 내부 구조를 직접 활용할 수 있게 된다. 이렇게 되면 현재 C 혹은 Cython 레벨의 인터페이스를 요구하는 scikit-learn 같은 다른 프로젝트에서도 pandas 객체 데이터에 쉽게 접근할 수 있게 된다.



[그림 6] pandas 0.x/1.x



[그림 7] pandas 2.0

또한, 파이썬 커뮤니티의 움직임에 맞춰 파이썬 2 지원을 종료하고 파이썬 3.5 이상 만을 지원하려는 논의도 진행 중이다. 이렇게 함으로써 asyncio와 같은 파이썬 3에서만 사용 가능한 기능의 이점을 취할 수도 있을 것이다.