

데이터 역량 개발 스터디그룹 3회차

데이터 분석 라이브러리 개발

김태완
디지털혁신실 디지털신기술반

2022.11.23

Outline

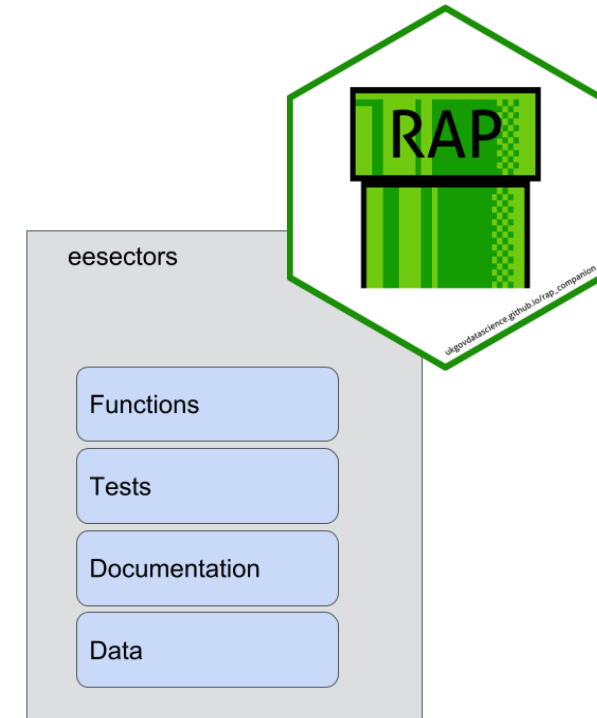
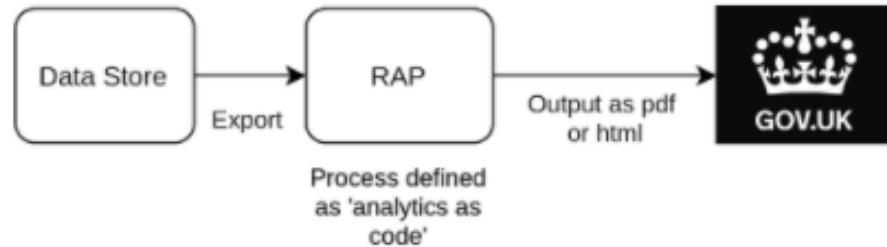
1. 개요 및 예시
2. 라이브러리 개발 내용
3. 라이브러리 개발 방식
4. 향후 일정

데이터 분석 라이브러리 개요 및 예시

RAP (Reproducible Analytical Pipeline)

- Analysis as code
- Using R and Python as languages
- Open source
- Identifying different users' needs

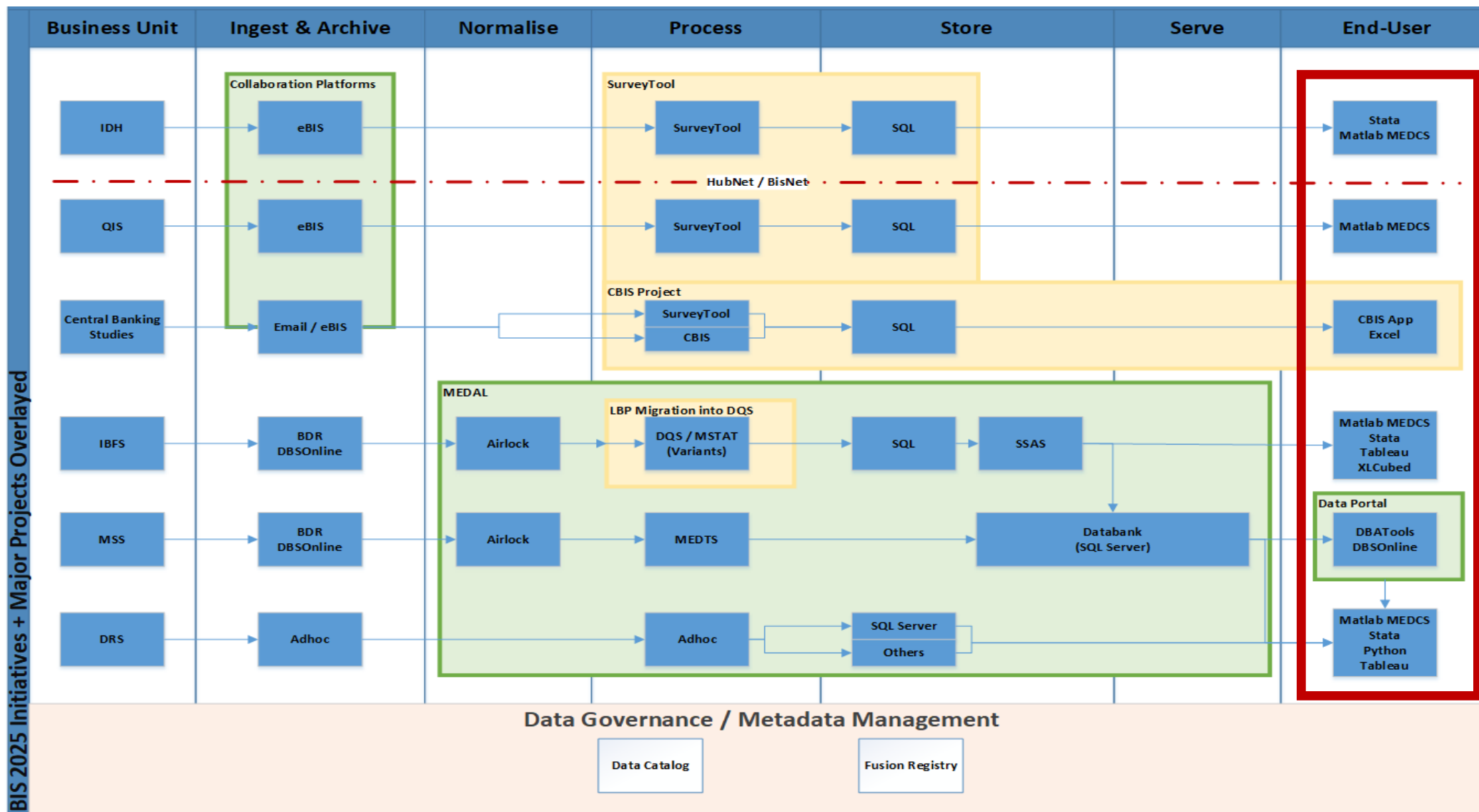
RAP - 영국정부 통계편제용 패키지



Source: <https://dataingovernment.blog.gov.uk/2017/03/27/reproducible-analytical-pipeline/>

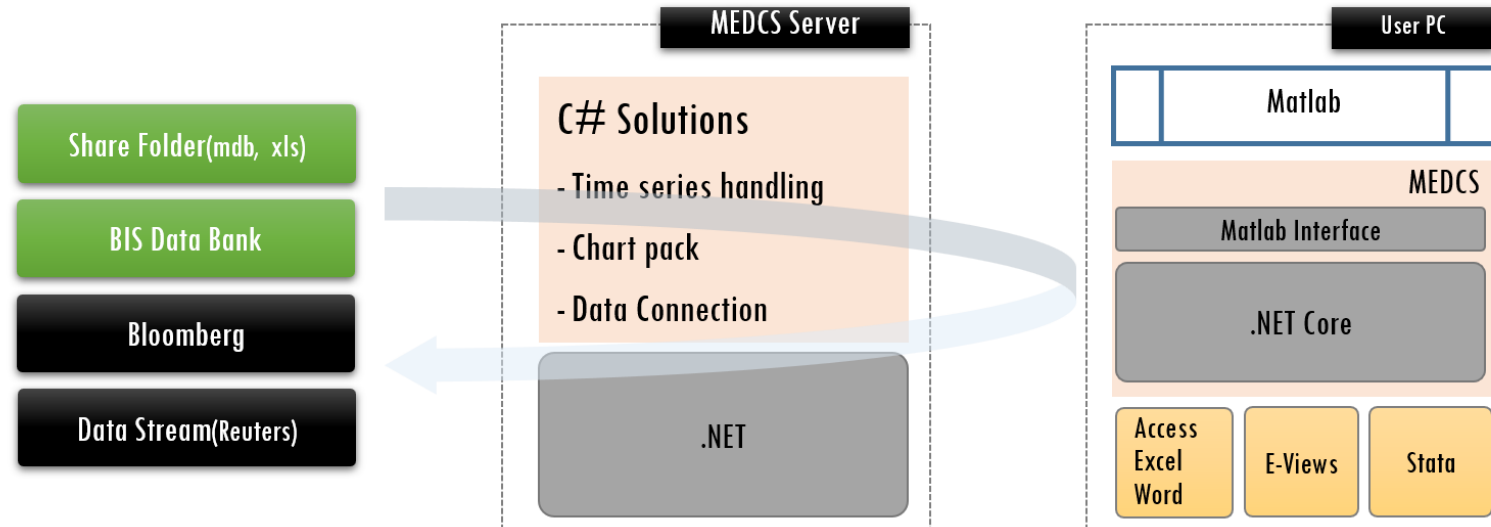
https://ukgovdatascience.github.io/rap_companion/exemplar.html

RAP - BIS 데이터 플랫폼

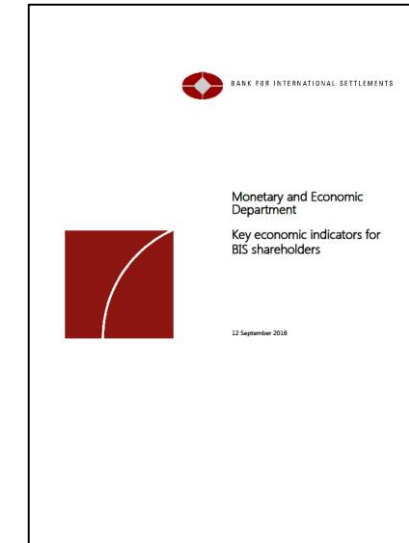


RAP - BIS 데이터 분석 라이브러리(MEDCS/MEDAU)

- Optimized statistical analysis environment for Matlab users
- Specialized time series collection
- Useful Econometric formulas(regressions, interpolation)
- Access various internal/external data sources
- One stop process from raw data to the publication(Chart, PDF)



Analysis performed	Function
Absolute value	ABS
Annualized percent changes (see Simple annualized percent changes)	ANNPCT
AR forecast	AR
ARIMA forecast	ARIMA
ARMA forecast	ARMA
Arctangent	ATAN
Autocorrelation (see Partial autocorrelation)	ACF
Average (see Centered moving average, Cumulative averages, and Moving averages)	AVE LAVE MEAN
Average growth rate	AGR
Base 10 logarithm	LOG10
Centered moving averages	MAVEC
Correlation (see Autocorrelation, Moving correlation, and Partial autocorrelation)	CORR

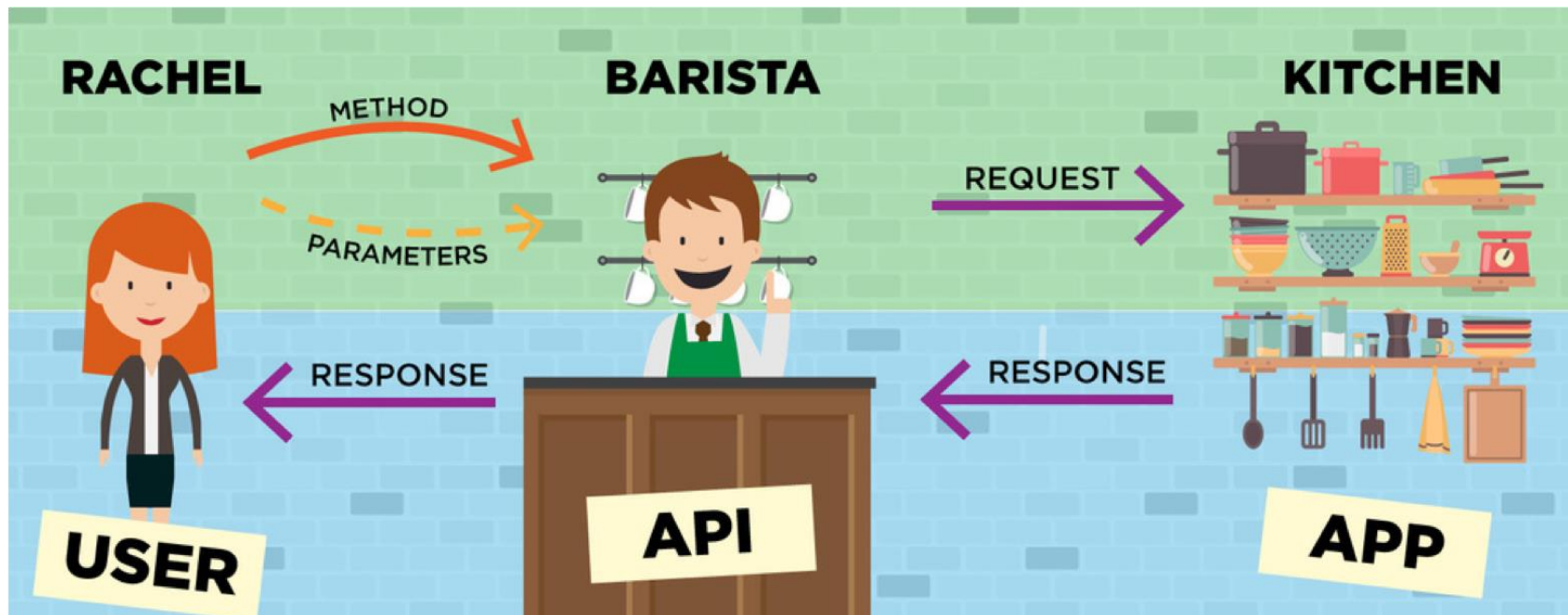


데이터 분석 단계별 라이브러리 / 패키지

- 데이터 입수
 - 데이터 원천기관별 API*를 이용한 다양한 라이브러리
- 데이터 전처리
 - NumPy, Pandas / dplyr, reshape
- 모형 구축 및 평가
 - Scikit-learn, statsmodels, SciPy / caret, glm, knn, etc
- 데이터 시각화
 - Matplotlib, Seaborn / ggplot2

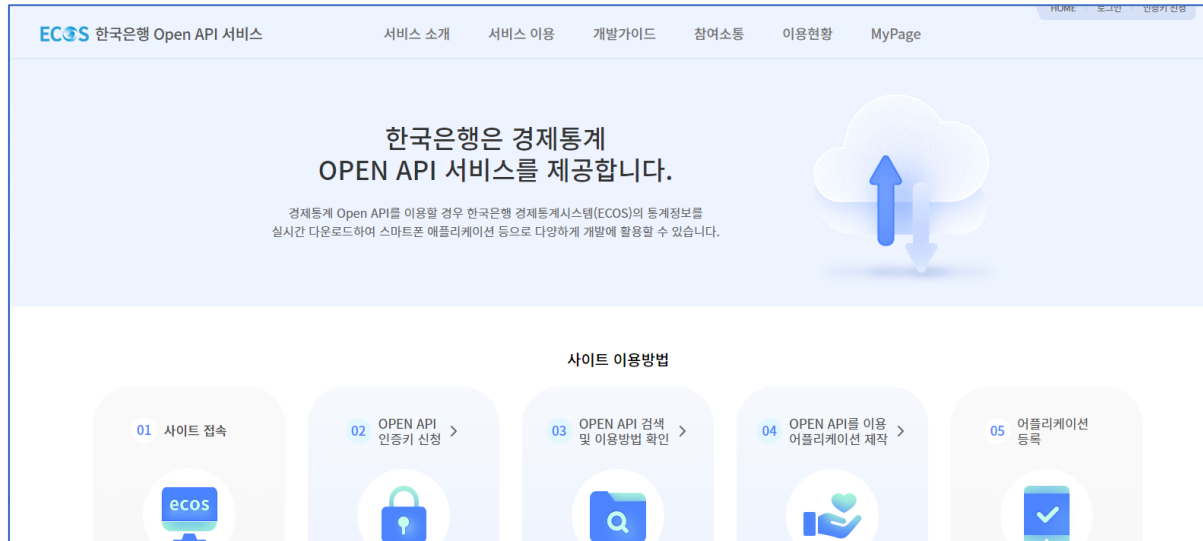
* API (Application Programming Interface)

- API는 '인터페이스'에 불과
- REST API(web), 일반 라이브러리 형식 등

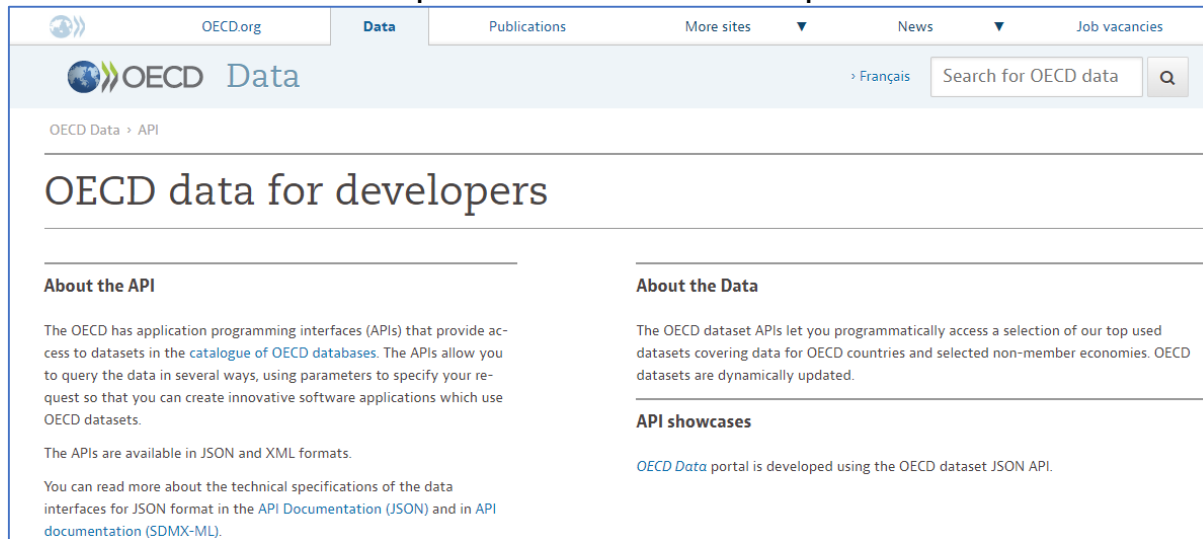


Source: <https://joshuadull.github.io/APIs-for-Libraries/>

데이터 입수 - APIs



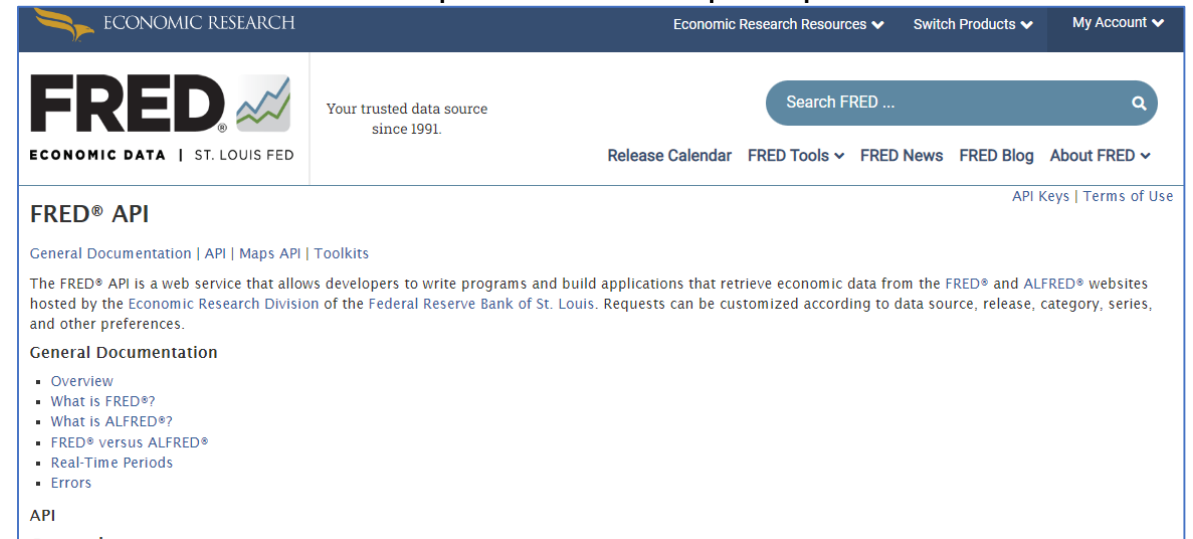
<https://ecos.bok.or.kr/api/>



<https://data.oecd.org/api/>

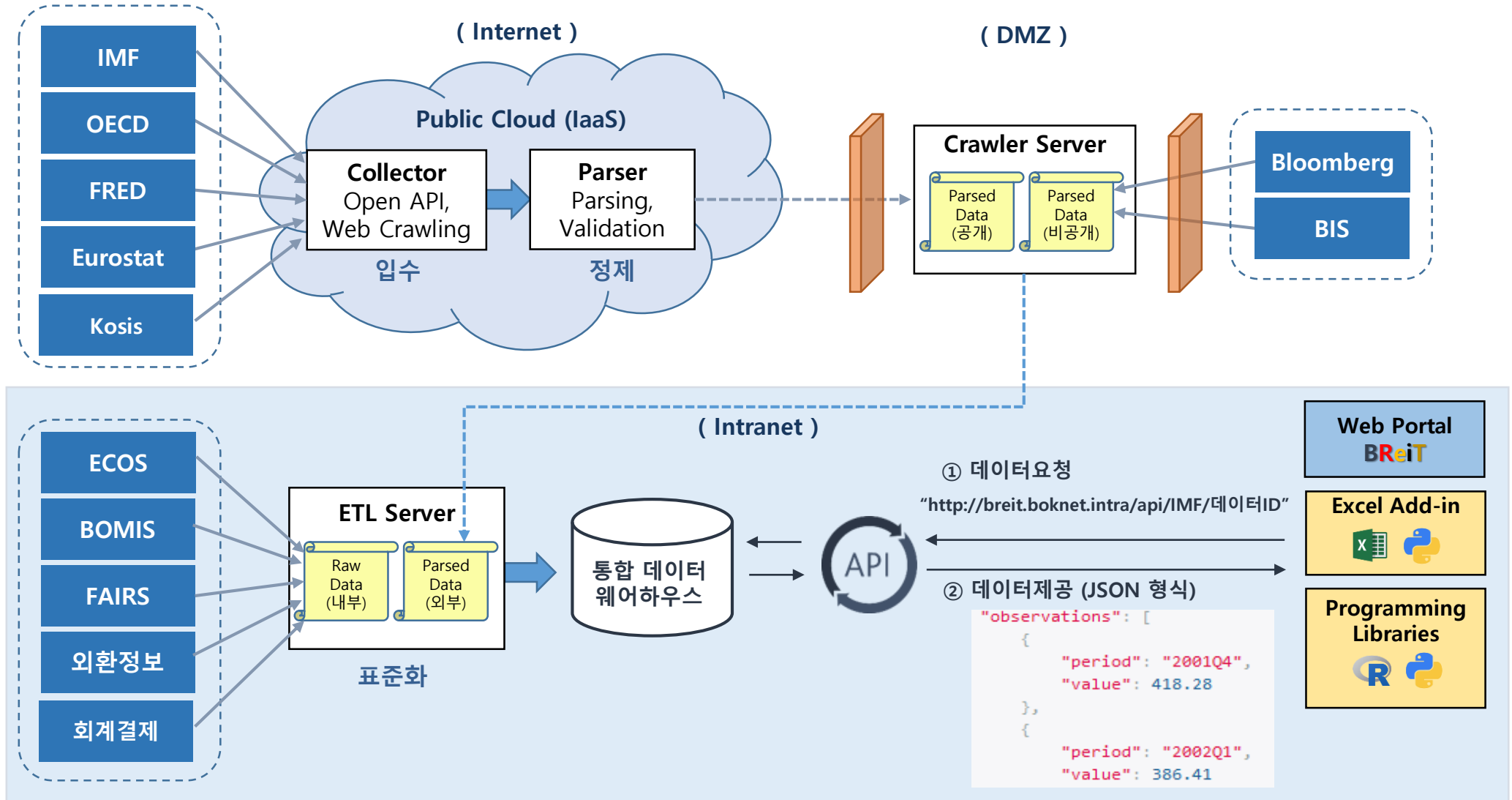


<https://kosis.kr/openapi>



<https://fred.stlouisfed.org/docs/api/fred/>

데이터 입수 - BReiT API



데이터 전처리

- 테이블 만들기(행과 열 라벨 지정)
- 데이터 정제(결측치 처리, 변수타입 변환 등)
- 데이터 구조화(시계열, 멀티인덱스 등)
- 빈도 변환(일 ↔ 월 ↔ 분기 ↔ 연)
- 데이터 결합

country	year	cases	population
Afghanistan	1999	175	19997071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

variables

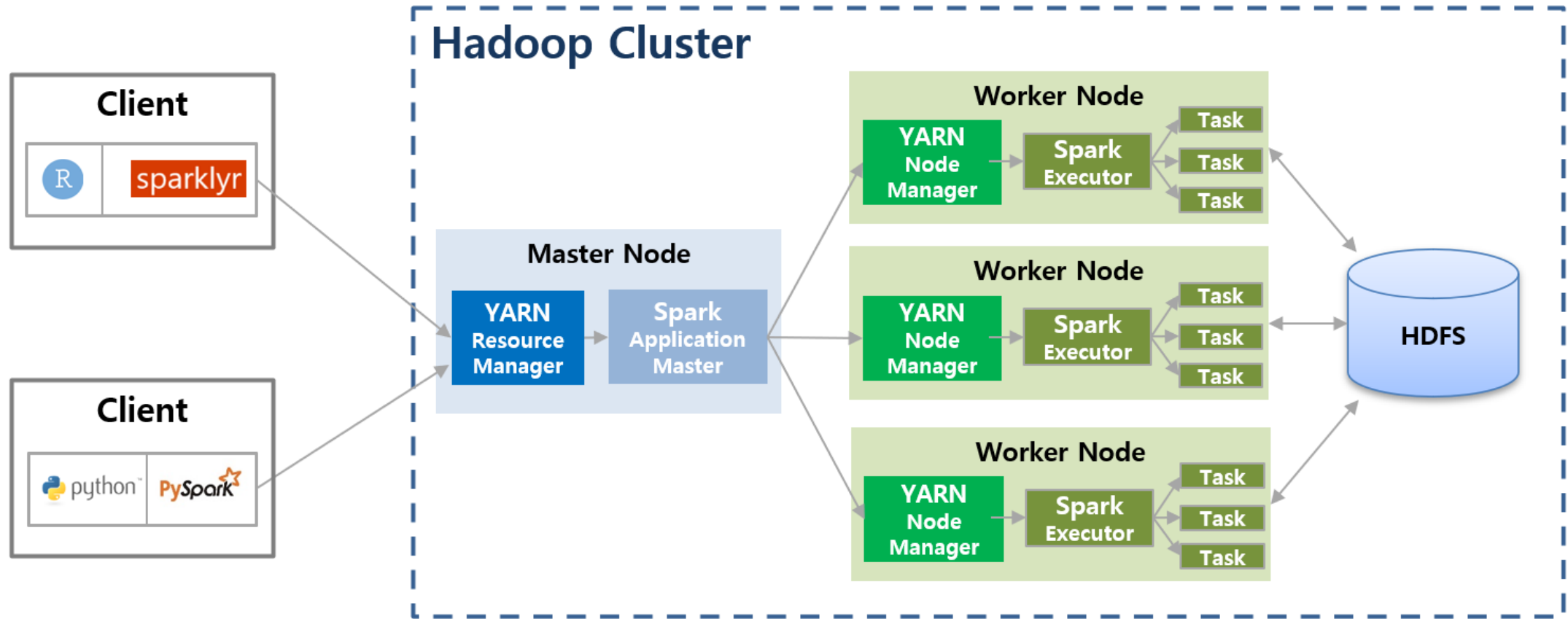
country	year	cases	population
Afghanistan	1999	175	19997071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

observations

country	year	cases	population
Afghanistan	1999	175	19997071
Afghanistan	2000	2666	20095360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

values

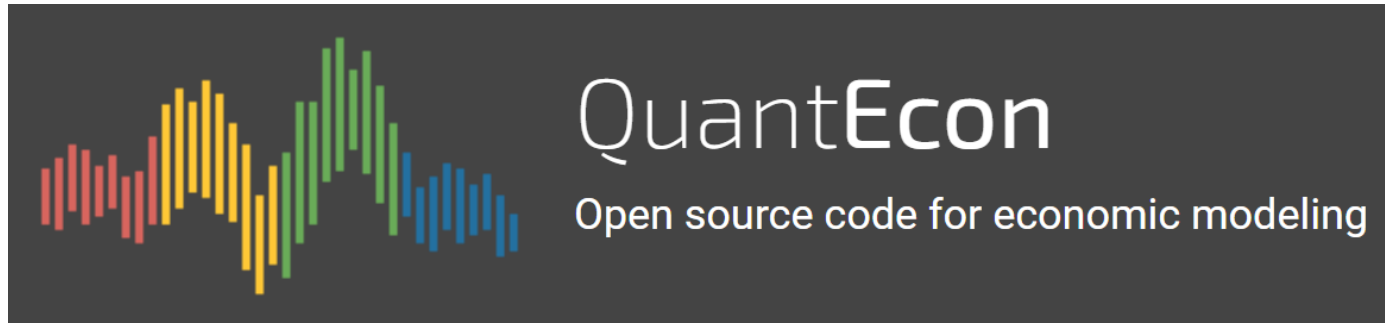
데이터 전처리- APIs for Big Data Processing



모형 구축 및 평가

- Descriptive/Predictive Modeling
 - 계량경제모형
 - 인공지능/기계학습 모형
- Model Evaluation
 - 교차검증
 - 회귀/분류 성능 평가
 - 모형간 성능 비교검증

모형 구축 및 평가 - Econometrics



<https://quantecon.org/>



<https://www.statsmodels.org/>



<https://www.oselab.org>

모형 구축 및 평가 - AI / ML



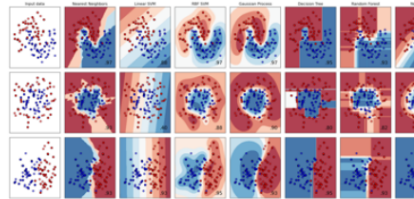
<https://scikit-learn.org/>

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

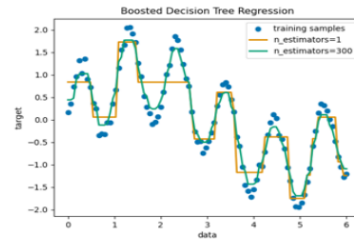


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...

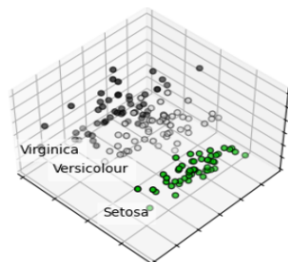


Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

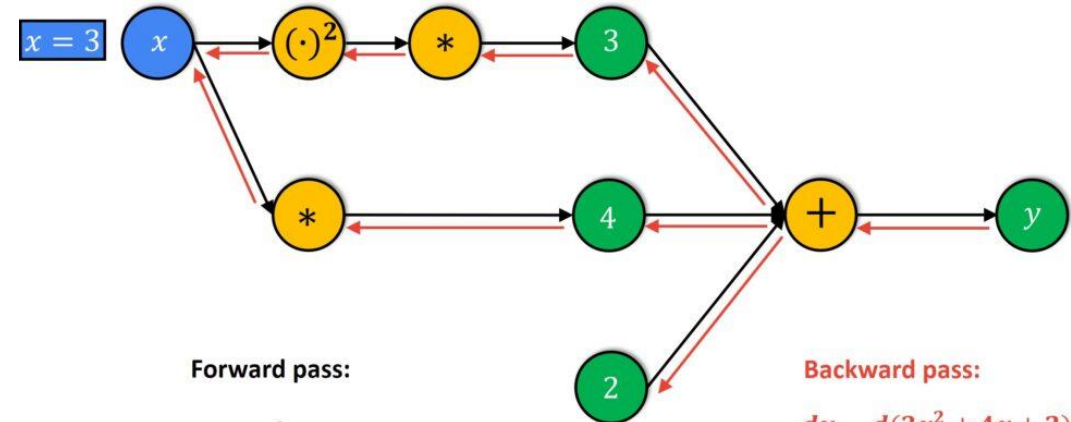
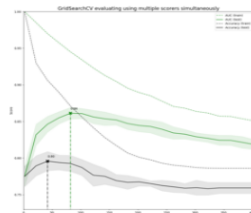


Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...



Forward pass:

$$y = 3x^2 + 4x + 2$$

$$y = 3 \cdot 9 + 4 \cdot 3 + 2$$

$$y = 41$$

Backward pass:

$$\frac{dy}{dx} = \frac{d(3x^2 + 4x + 2)}{dx}$$

$$\frac{dy}{dx} = 2 \cdot 3x + 4 = 6x + 4$$

$$\frac{dy}{dx} = 18 + 4 = 22$$

<https://datahacker.rs/004-computational-graph-and-autograd-with-pytorch/>

데이터 분석 라이브러리 내용

데이터 분석 라이브러리 구성

■ 라이브러리

- 데이터 입수: (외부) API/크롤링, (내부) BReiT API 연동
- 데이터 전처리: 필터링, 텍스트·대용량 데이터 전처리 등
- 모형 구축 및 평가: 주요 계량모형, 인공지능, XAI기법 등
- 데이터 시각화: 행내 스타일 테이블, 차트 자동 생성 등

■ 사용자 매뉴얼

- 라이브러리 목록·설명 / best practice

데이터 분석 라이브러리 사용례

QuantEcon
latest

Search docs

Game theory
Markov
Optimize
Random

Tools

- arma
- ce_util
- compute_fp
- discrete_rv
- distributions
- dle
- ecdf
- estspec
- filter
- graph_tools
- gridtools
- inequality
- ivp
- kalman**
 - References
 - kalman**
 - lae
 - lqcontrol
 - lqnash
 - lss

Docs » Tools » kalman

kalman

Implements the Kalman filter for a linear Gaussian state space model.

References

<https://lectures.quantecon.org/py/kalman.html>

`class quantecon.kalman.Kalman(ss, x_hat=None, Sigma=None)` [\[source\]](#)

Bases: `object`

Implements the Kalman filter for the Gaussian state space model

$$\begin{aligned}x_{t+1} &= Ax_t + Cw_{t+1} \\ y_t &= Gx_t + Hv_t\end{aligned}$$

Here x_t is the hidden state and y_t is the measurement. The shocks w_t and normals. Below we use the notation

$$Q := CC'R := HH'$$

Parameters:

- `ssinstance of LinearStateSpace`
An instance of the `quantecon.lss.LinearStateSpace` class
- `x_hat`scalar(float) or array_like(float), optional(default=None)
An $n \times 1$ array representing the mean \hat{x} of the prior/pred zero if not supplied.
- `Sigma`scalar(float) or array_like(float), optional(default=None)
An $n \times n$ array representing the covariance matrix Sigma of the density. Must be positive definite. Set to the identity if not supplied.

References

▼ 데이터 입수

```
[ ] indicators = ['empi', 'bankbeta', 'stockr', 'stockv', 'sr']
countries = ['KR']

df = get_breit_series('ECOS', indicators, countries, 'M', '1995-06', '2011-12')
```

▼ 데이터 전처리

```
[ ] df_processed = HP_filter(df)
```

▼ 데이터 시각화

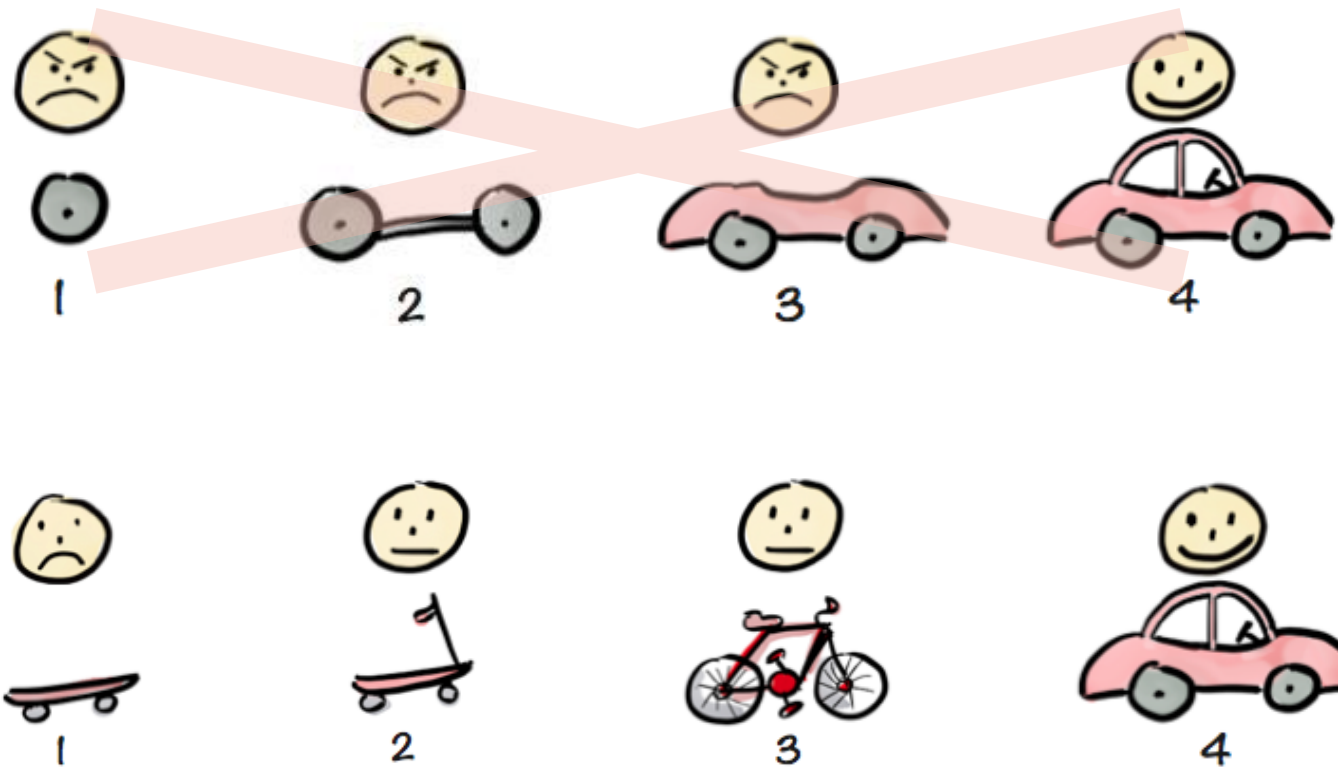
```
[ ] plot_term(df_processed.sum(axis=1), k=1)
```

라이브러리 개발 방식

라이브러리 개발 방식

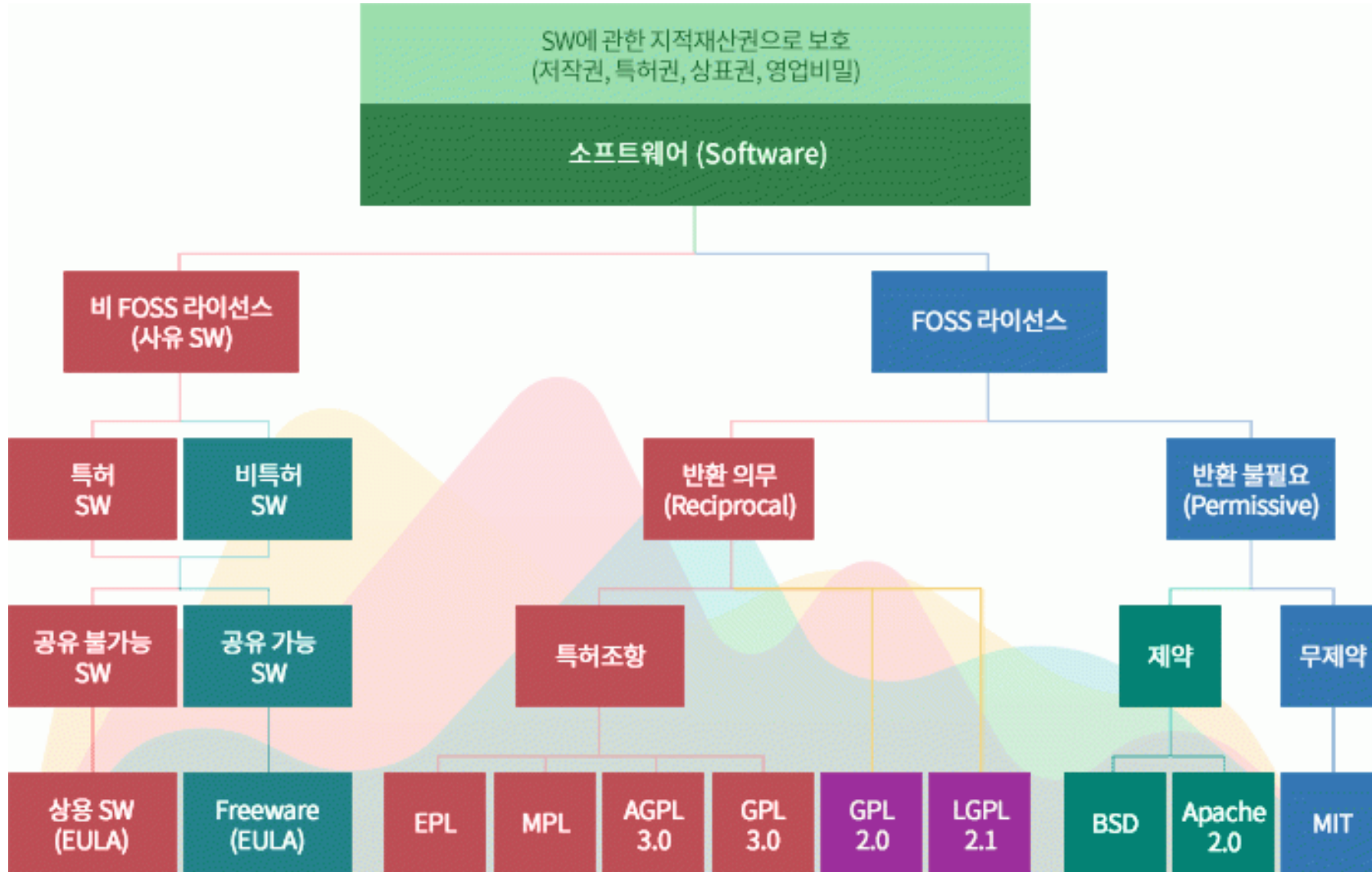
- 개발방법론
- 라이선스 유형
- 개발 언어
- 코딩 스타일
- 테스트 방식
- 협업 도구/방식
- 문서화 도구/방식

개발방법론 - MVP (Minimum Viable Product)



Source: <https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>

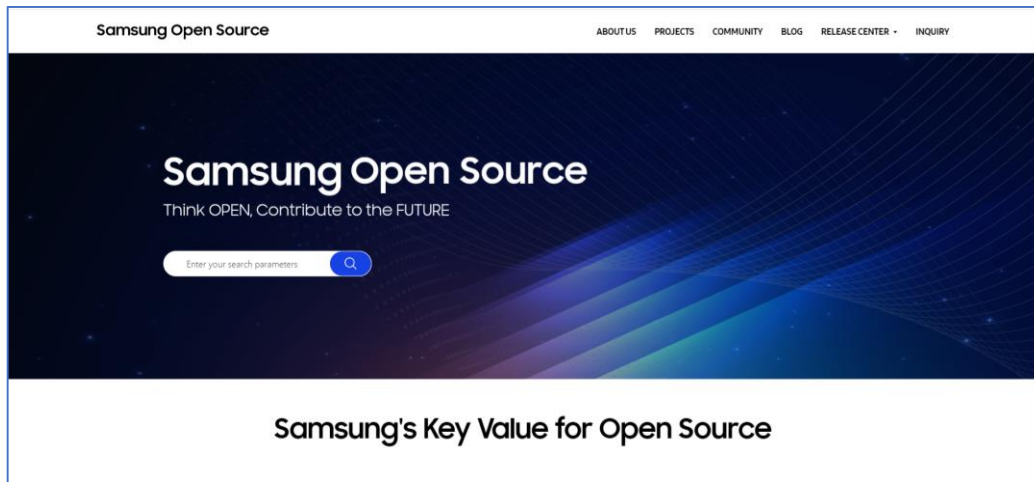
라이선스 유형



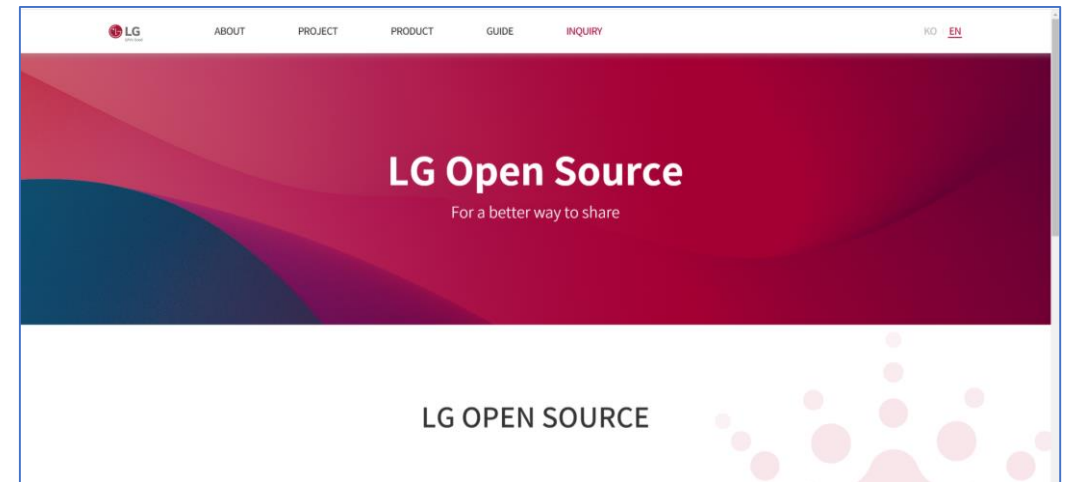
Source: https://www.oss.kr/oss_license

라이선스 유형 - 오픈소스

- 저작권, 개발자 및 기여자 명시
- 라이선스 정보 제공
- 코드 수정 정보 명시 (Apache 2.0)
- 동일 라이선스로 재배포 등 (GPL 등)



<https://opensource.samsung.com/main>

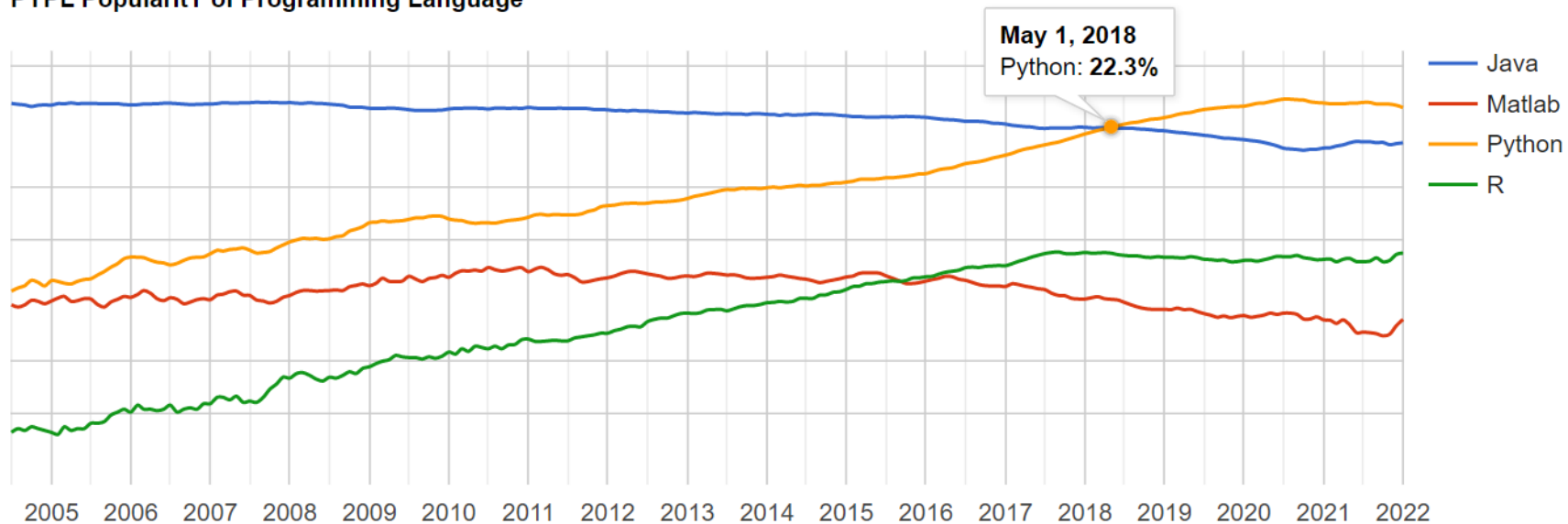


<https://opensource.lge.com/>

개발 언어

Worldwide, Python is the most popular language, Python grew the most in the last 5 years (13.4%) and Java lost the most (-5.1%)

PYPL Popularity of Programming Language



Source: <https://pypl.github.io/PYPL.html>

Rank	Language	Share
1	Python	28.74 %
2	Java	18.01 %
3	JavaScript	9.07 %
4	C/C++	7.4 %
5	C#	7.27 %
6	PHP	6.06 %
7	R	4.19 %
8	Objective-C	2.27 %
9	Swift	1.91 %
10	TypeScript	1.74 %
11	Matlab	1.74 %
12	Kotlin	1.71 %

개발 언어 - Python vs R

	Python	R
General	Python is a general-purpose programming language for data analysis and scientific computing.	R is a functional programming environment and language for statistical computing and graphics.
Objective	Data Science, Web Development, Embedded Systems	Data Science & Statistical Modeling
IDE	iPython, Pycharm, Jupyter Notebook, Spyder	Rstudio, R GUI, R KWARD
Data Collection	Supports CSV files, SQL , JSON , and web scraping with BeautifulSoup .	Can also import csv files with built-in readr library. R's library RCurl provides a simple way to make API requests, similar to Python's requests package.
Data Analysis	Organize dataframes with Pandas filtering, sorting. Python takes a more streamlined approach for data science projects.	Complex data visualization tools make the exploratory data analysis (EDA) process much more complex than Python.
Essential Packages & Libraries	Numpy , Pandas , matplotlib , scipy , scikit-learn , TensorFlow	caret , stringr , ggplot2 , knitr , tidyverse , markdown , shiny , forcats , haven
Database Handling Capacity	Can easily handle large data because there are less constraints for memory usage	R computes everything in memory, so its capabilities are limited by RAM size. A major downfall of R is the inability to handle massive amounts of data
Data Visualization	Despite the capabilities of data visualization tools like Matplotlib and Seaborn , Python fails to measure up to data visualization features of R.	Developed by and for statisticians, R has complex data visualization features.
Syntax	The 'zen of python' is that there's a proper way to write code.	R doesn't have this set of rules. Also indexing starts at 1, which can be considered unconventional for general programmers.
Learning Curve	Simple and readable code structure makes it easier for beginners to learn. It also allows for object-oriented programming. It also offers a wide range of data structures that you wouldn't expect from a general-purpose language.	R's functional syntax isn't easy for beginners, but not too challenging for those well versed in programming. It also offers a few data structures, but fails to handle large amounts of data.

Source: <https://towardsdatascience.com/python-vs-r-the-basics-d754c45c1596>

개발 언어 - Polyglot Programming



<https://rpy2.github.io/>

```
#python  
print("Hello World!")
```

Hello World!

```
[4] %%R  
#R using Cell Magic  
"Hello World!"
```

[1] "Hello World!"

```
[6] # using Line Magic  
x = 5  
%R y <- 3  
print(x)  
%R y
```

5
array([3.])



<https://github.com/rstudio/reticulate/>

reticulate is powerful!

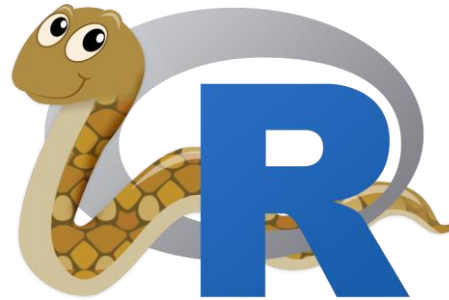
```
8 {r setup}  
9 library(reticulate)  
10 use_condaenv(condaenv = "r-reticulate", conda = "e:/mini  
11 {r}
```

```
14 {r}  
15 df <- data.frame(c(1, 2, 3), c(4, 5, 6))  
16 dim(df)  
17
```

[1] 3 2

```
18 {python}  
19 import pandas as pd  
20 df = pd.DataFrame([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  
21 df.shape  
22  
23
```

(3, 3)



코딩 스타일 - PEP 8

<https://peps.python.org/pep-0008/>

- 변수명 부여규칙
- 들여쓰기(indentation)
- 띄어쓰기 / 줄바꿈
- 주석 규칙 등

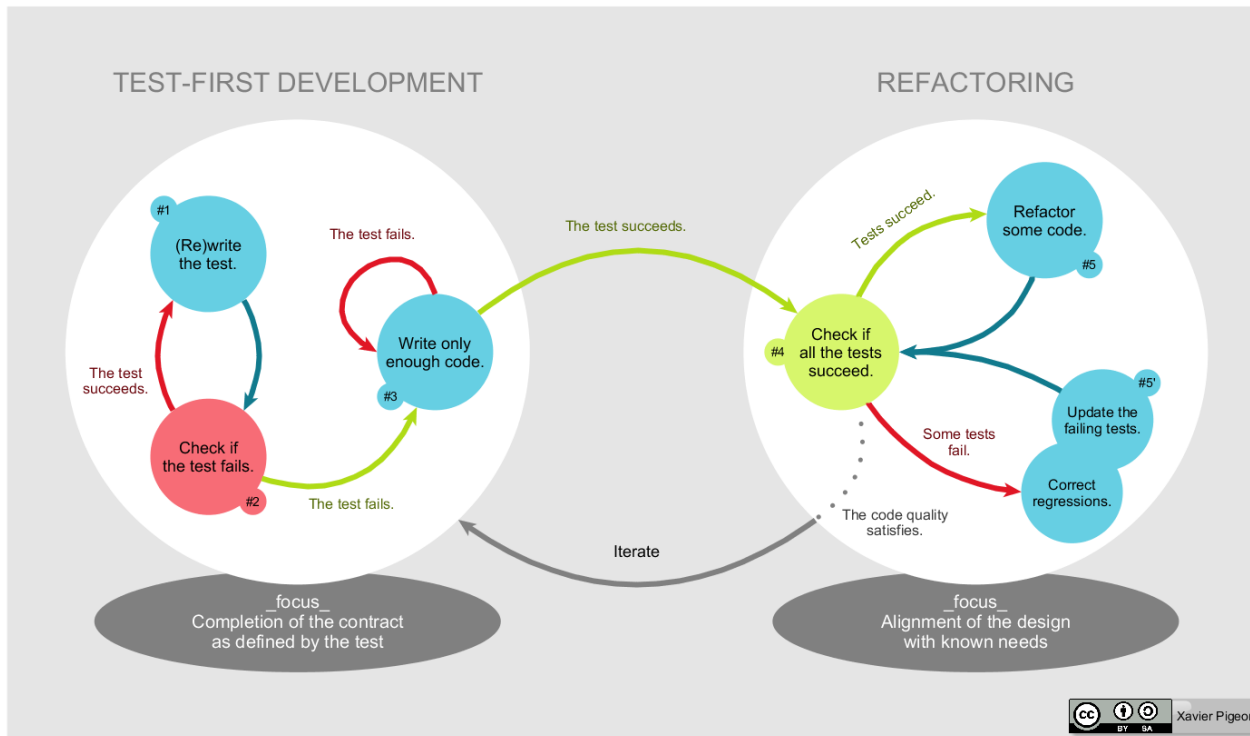
```
import math, sys;

def example1():
    """This is a long comment. This should be wrapped to fit within 72 characters.
    some_tuple=( 1,2, 3,'a' );
    some_variable={'long':'Long code lines should be wrapped within 79 characters.',
    'other':[math.pi, 100,200,300,9876543210,'This is a long string that goes on'],
    'more':{'inner':'This whole logical line should be wrapped.',some_tuple:[1,
    20,300,40000,500000000,6000000000000000000]}}
    return (some_tuple, some_variable)
```

```
import math
import sys

def example1():
    # This is a long comment. This should be wrapped to fit within 72
    # characters.
    some_tuple = (1, 2, 3, 'a')
    some_variable = {
        'long': 'Long code lines should be wrapped within 79 characters.',
        'other': [
            math.pi,
            100,
            200,
            300,
            9876543210,
            'This is a long string that goes on'],
        'more': {
            'inner': 'This whole logical line should be wrapped.',
            some_tuple: [
                1,
                20,
                300,
                40000,
                500000000,
                6000000000000000000]}}
    return (some_tuple, some_variable)
```

테스트 방식 - Test-Driven Development

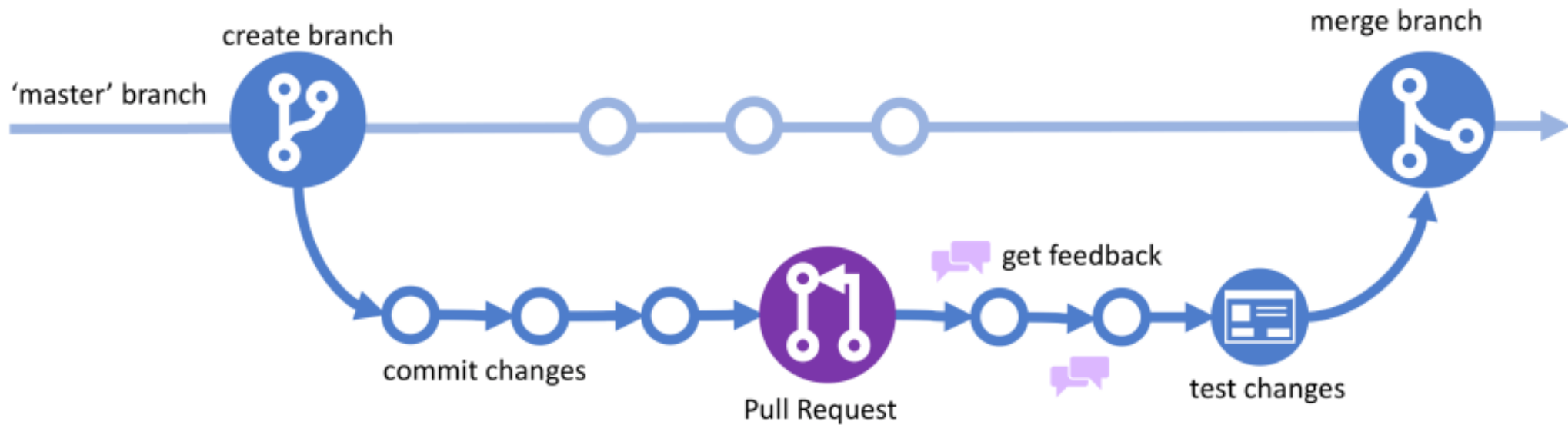


```
100 df_table[df_table$sector != 'perc_of_UK', paste0('X', x$years)] <- roundf(
101 df_table[df_table$sector == 'perc_of_UK', paste0('X', x$years)] <- sprintf
102
103 # Finally set
104
105 df_table$sector <- factor(unname(x$sectors_set[df_table$sector]))
106
107 # Print to html or as dataframe ----
108
109
110 if (html == TRUE) {
111
112     df_table <- xtable::xtable(
113         x = df_table,
114         ...
115     )
116
117     print(
118         df_table,
119         type = 'html'
120     )
121
122 } else {
```

https://ukgovdatascience.github.io/rap_companion/code-cover.html

협업 도구/방식

GitHub Flow



Copyright © 2018 Build Azure LLC

<http://buildazure.com>

Source: <https://build5nines.com/introduction-to-git-version-control-workflow/>

문서화 도구/방식

- 마크다운 사용 규칙
- 사용자 매뉴얼 - Read the Docs, Jupyter Book 등



<https://readthedocs.org/>



<https://jupyterbook.org/>

향후 일정

예상 일정

- 라이브러리 개발 내용/방식 설계('23.1)
 - 의견 수렴
 - 유사 프로젝트 벤치마크
 - 개발 표준 수립
- 협업 개발환경 마련 ('23.2)
 - github, 내부망 gitlab 등
 - 관련 사업결의
- 라이브러리 구축('23.3 ~)

감사합니다