

데이터 분석 라이브러리 개발 로드맵

BOK Data Analysis Library

2024.06.11

디지털신기술팀 이창훈 과장

데이터 분석 라이브러리

- 데이터 분석 라이브러리란?

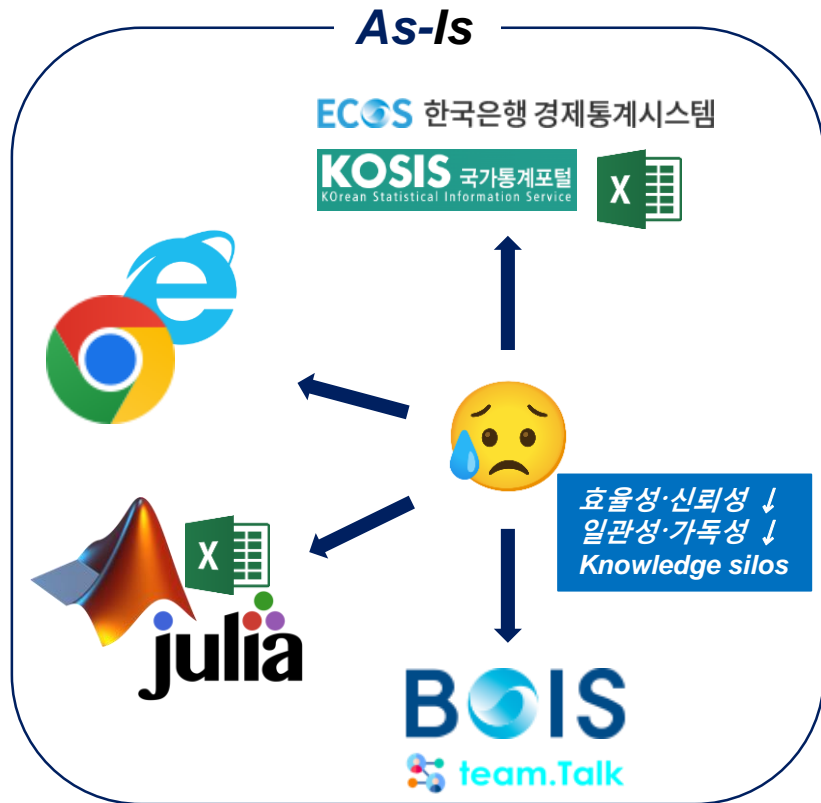
- 연구자가 필요한 논문을 도서관에서 찾아 읽듯이 데이터 분석에 필요한 코드를 라이브러리에 불러와 사용
- 일반적으로 사용 목적에 맞게 특화하여 개발: Pandas(데이터 처리 분석), NumPy, Matplotlib, scikit-learn, TensorFlow, PyTorch(AI/ML) 등

- 데이터 분석 라이브러리의 목적

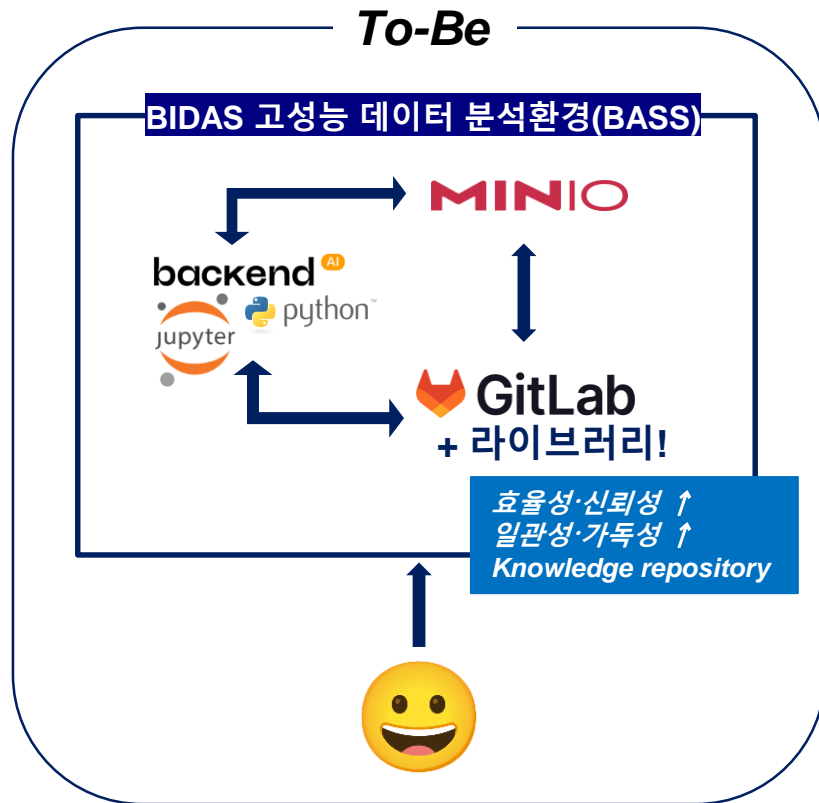
- 검증된 코드를 재사용함으로써 데이터 분석 업무의 효율성, 신뢰성 제고
- 코딩이 익숙하지 않은 사용자도 손쉽게 데이터 분석 수행 가능(튜토리얼 제공)

라이브러리를 통한 데이터 분석 업무 효율화

As-Is



To-Be



한국은행 데이터 분석 라이브러리

● 추진 배경 및 개요

- 당행 데이터 분석 업무 성과와 노하우를 지적자산으로 축적하여 업무 효율성·역량 제고
 - 행내 수요가 높은 계량경제 데이터 분석 라이브러리 개발을 우선 추진
 - 기존 데이터 분석 코드를 표준화·최적화하여 수록 예정
- 데이터 처리, 분석, 시각화, 공유까지 통합된 워크플로우 구축
 - 파이썬 기반 라이브러리 개발을 통해 BAAS내에서 효율적으로 연계

● 계량경제 데이터 분석 라이브러리 개발 현황

- 고려대학교 경제학과 연구진(한치록 교수, 한국계량경제학회 회장)과 협업
- 선형회귀, 시계열, 패널, 베이지안 등 ‘기본적인’ 계량경제 분석 및 시각화 도구 개발

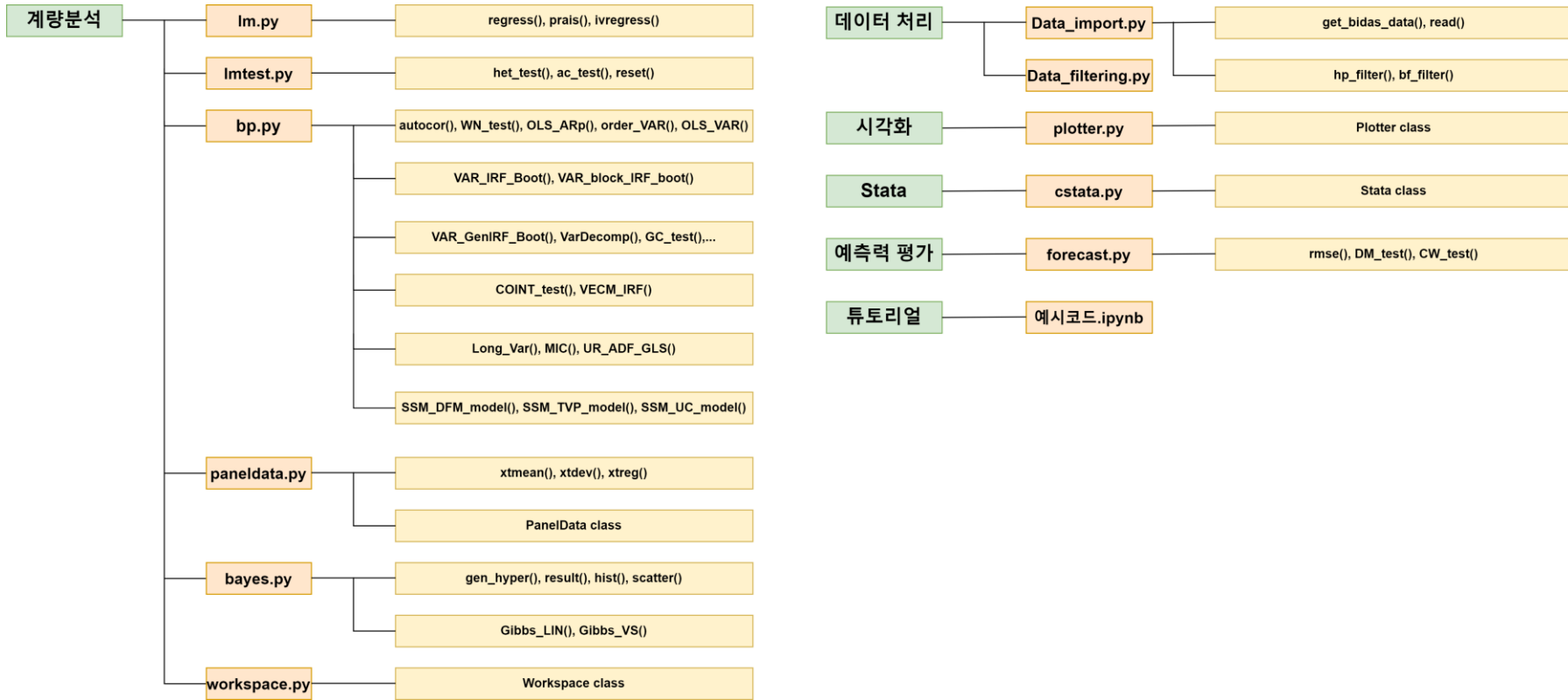
계량경제 데이터 분석 라이브러리 개발 중점 포인트

- **기능적 우수성** *정확성, 확장성, 호환성*
 - 업계 표준 데이터 분석 도구(stata) 수준의 정확성, 안정성 확보
 - 지속적인 업데이트 + 자연어 기반 데이터 분석 모듈로 활용
 - 기존 파이썬 라이브러리와 결합하여 작동
- **사용자 경험** *사용성, 가독성, 활용성, 한글 매뉴얼 및 튜토리얼*
 - 프로그래밍 언어에 익숙하지 않은 사용자도 매뉴얼 이용해 간편하게 사용
 - 코드 및 분석 결과 핵심정보를 쉽게 파악
 - 접근성·학습 효율성 향상, 행내 데이터 분석 생태계 활성화
- **인프라 활용 및 커뮤니티** *BIDAS 데이터 분석 자원 활용, 오픈소스*
 - Backend.AI, Gitlab, Object storage
 - Gitlab 통한 행내 공유 및 Github 이용한 대외공개

계량경제 데이터 분석 라이브러리 주요 기능

- **데이터 입수 및 처리** 데이터 입수, 전처리, 변환, 필터링
- **계량경제 분석** 선형회귀, 시계열, 패널, 베이지안 분석, workspace
- **시각화** 간단한 코드로 스냅샷 스타일 그림 그리기(향후 보고서별 그림 스타일을 옵션으로)
- **Stata 모듈** 파이썬에서 stata의 모든 기능 활용
- **예측력 평가** RMSE, MAE, DM test, CW test 등 예측력 평가
- **튜토리얼** Markdown 형식의 매뉴얼 및 라이브러리 사용 예시 코드

계량경제 데이터 분석 라이브러리 구조



총 6개 모듈, 12개 이상 클래스, 335개 이상 함수

라이브러리 불러오기

Python, R

① 사양 및 버전 선택 후 분석환경을 선택하세요.

분석환경 기동 중...

사양: (권장) 학습용 4vCPU, 8G RAM
버전: (기본) Python 3.9, R 4.3



Jupyterlab 터미널에서

git clone <https://bidas-gitlab.boknet.intra/digitaltech/bedal.git>

./INSTALL_LINUX.sh

The screenshot shows the Jupyterlab interface. On the left is a file explorer with a list of files and folders. The file 'DFM_example.ipynb' is selected. The main area shows the code editor with two code cells. The first cell contains a pandas DataFrame creation and a plot. The second cell contains a plot using matplotlib. The terminal at the bottom shows the output of the code execution.

File Explorer:

Name	Last Modified
bok	7 days ago
bok_python	7 days ago
fig	7 days ago
fonts	7 days ago
img	7 days ago
test_data	7 days ago
02 BOK library.ipynb	7 days ago
07 BOK Library.ipynb	7 days ago
08 BOK Workspace.ipynb	7 days ago
1 data import.ipynb	7 days ago
ACF_AR_manual_array.ipynb	7 days ago
Bayes_LinearReg.ipynb	7 days ago
bok_python_w_var_w_linux.7z	7 days ago
coint_VECM_manual_array.ipynb	7 days ago
DFM_example.ipynb	7 days ago
INSTALL_LINUX.sh	7 days ago
INSTALL_Window.bat	7 days ago
LV_UR_manual_array.ipynb	7 days ago
Main_SSM_DFM.ipynb	7 days ago
Main_SSM_TVP.ipynb	7 days ago
Main_SSM_UC.ipynb	7 days ago
matplotlib-3.9.0-cp39-cp39-manylinux2_17_x86_...	7 days ago
README.md	7 days ago
VAR_example.ipynb	7 days ago
VAR_manual_array.ipynb	7 days ago

Code Editor:

```
[10]: Factors = pd.DataFrame({'f1':Res.Global_factor,...,'f2':Res.R...})
```

```
[11]: mpl.rcParams.update(mpl.rcParamsDefault)
plt.rcParams["font.family"] = ["NanumGothicCoding"]
plt.rcParams["axes.unicode_minus"] = False
```

```
[12]: pp = bok.Plotter(figsize=(10,5),xmargin=0)
pp.line(Factors.index, [Factors.f1, Factors.f2, Factors.f3])
pp.set_xaxis('year')
pp.legend()
#pp.export('af.pdf')
```

Terminal:

4. 분석결과 시각화

4-1. 그림 그리고자 하는 결과 데이터프레임으로

4-2. Plotter 이용해 그림 그리기

한글 폰트 및 마이너스 표시 오류 수정

Legend: Global (blue line), R (orange line)

Plot: A line graph showing two data series over time. The x-axis is labeled 'year' and the y-axis ranges from 5 to 15. The blue line represents 'Global' and the orange line represents 'R'. The blue line shows a significant peak around year 10, while the orange line remains relatively flat.

데이터 입수: get_bidas_data()

```
import bok
import pandas as pd

data_id = ['NECOS-200U008-Q-10601', 'NECOS-704U001-D-1010301', 'NECOS-901U009-M-0']
id = ['rgdp', 'cd91', 'cpi']

df = bok.get_bidas_data(data_id, id, start_d='2000-01-01', end_d='2024-05-01', freq='Q')

df[df.index >= '2022-03-31']
```

[1]:

	rgdp	cd91	cpi
period			
2022-03-31	0.660558	1.465254	3.911844
2022-06-30	0.750469	1.796885	5.377679
2022-09-30	0.233727	2.733175	5.840742
2022-12-31	-0.302968	3.910645	5.214290
2023-03-31	0.329875	3.641290	4.599045
2023-06-30	0.608653	3.633115	3.261611
2023-09-30	0.616073	3.735968	3.133239
2023-12-31	0.624853	3.829836	3.404684
2024-03-31	1.279769	3.693770	3.003747
2024-06-30	0.000000	3.574762	2.675194

계량분석: regress(), summary()

데이터 입수



모형 및 하이퍼파라미터 설정



실행



결과보기

```
import bok
import pandas as pd

df = pd.read_csv('../data/Death.csv')
fm = 'deathrate~smoke+drink+aged+I(smoke+aged)+C(year)'

res = bok.regress(fm, data=df, vce='cl', cluster='region')

res.summary(slim=True)
```

note: I(smoke - aged) omitted because of collinearity.

note: I(smoke + aged) omitted because of collinearity.

OLS Regression Results

```
=====
Dep. Variable:      deathrate    R-squared:      0.9205
Model:              OLS          Adj. R-squared:  0.9189
No. Observations:   258          F-statistic:   585.2
Covariance Type:    cluster      Prob (F-statistic): 0.0000
=====
               coef    std err          z      P>|z|      [0.025    0.975]
-----
Intercept      0.27932    0.89277      0.31     0.754    -1.47048    2.02913
C(year)[T.2009] -0.36224    0.07250     -5.00     0.000    -0.50433   -0.22015
C(year)[T.2010] -0.30478    0.07896     -3.86     0.000    -0.45955   -0.15001
smoke           0.04025    0.02118      1.90     0.057    -0.00126    0.08175
drink           0.00506    0.01469      0.34     0.731    -0.02374    0.03386
aged            0.39885    0.01354     29.46     0.000     0.37231    0.42539
=====
```

Notes:

1. Standard Errors are robust to cluster correlation (cluster)
2. I(smoke - aged), I(smoke + aged) omitted because of collinearity.

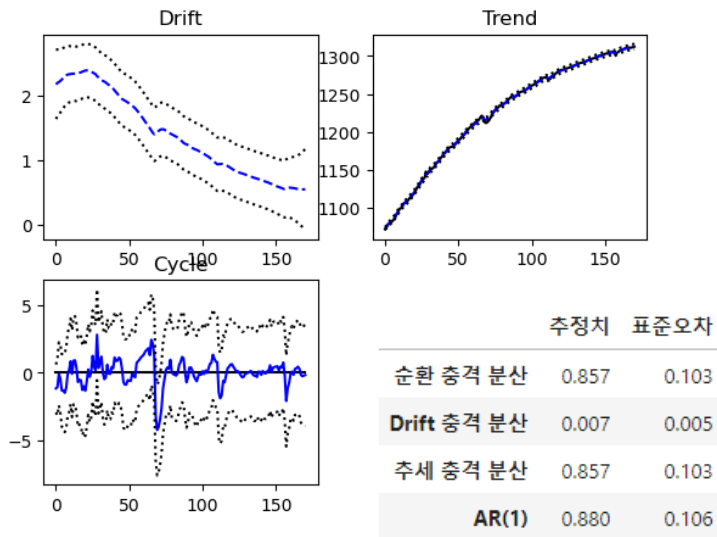
계량분석: ssm_UC()

```
import bok
import pandas as pd
import numpy as np

df = pd.read_excel('./test_data/Data_BayesEcon.xlsx', sheet_name='GDP')
df = 100*np.log(df)

res = bok.ssm_UC(df, P = 1, lamb = 1, drift = 'time-varying')

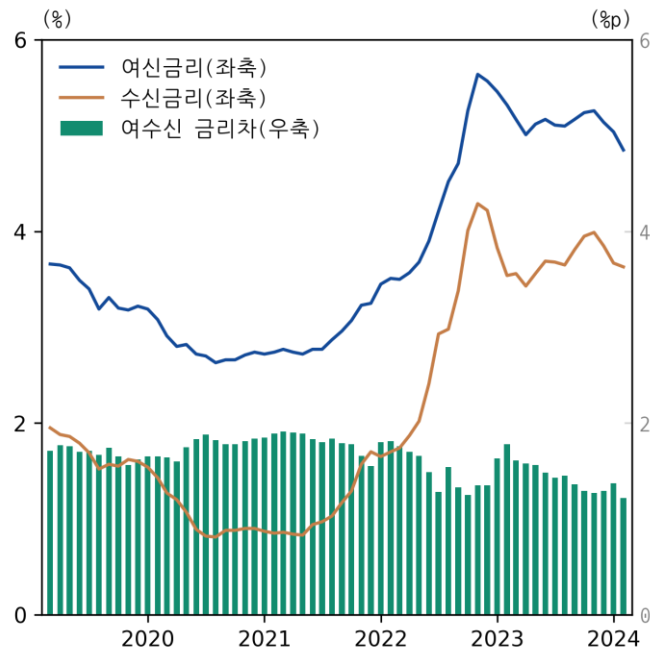
res.table_para
```



시각화: Plotter 클래스, line(), bar(),...

```
import bok
```

```
pp = bok.Plotter(xmargin = 0.01, figsize = (5,5))  
pp.line(df.x, [df.y1 df.y2], label = ['여신금리(좌축)', '수신금리(좌축)'])  
pp.bar(df.x, df.y3, label='여수신 금리차(우축)', axis = 1, width = 18)  
pp.annotate(['(%)', '(%p)'])
```



데이터 분석 라이브러리 향후 개발 방향

- 지속적인 업데이트로 분석 도구 다각화
 - 행내 수요가 많은 고급 모형 및 분석 도구 추가(예: 중립금리 추정 모형, 금리 기간구조, LBVAR 등)
 - 현재 개발중인 분석도구를 라이브러리에 포함
- GitHub 통한 대외 공개
 - 국내 경제·금융 부문 데이터 분석 라이브러리의 기준(커뮤니케이션국 협의중)
- 생성형 AI와 연계한 자연어 기반 데이터 분석 시스템 개발
 - 프롬프트 방식으로 간단하게 데이터 분석 및 시각화
- 보완할 것들
 - 라이브러리 검증 방안, 호환성 문제(파이썬 버전), data id 검색, 통일된 인터페이스 등

자연어 기반 데이터 분석 시스템



이창훈/디지털신기술팀 @2310490 2024-05-28 9:45:49

2020년 1월부터 현재까지 전년동월대비와 전월대비 한국 소비자물가지수상승률 그림을 그려줘. 단, 전년동월대비는 선그림으로, 전월대비는 막대그림으로 부탁해.

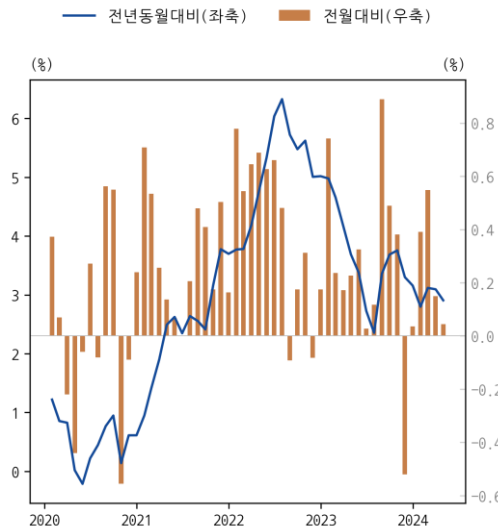


이창훈/디지털신기술팀 @2310490 2024-05-30 17:32:14

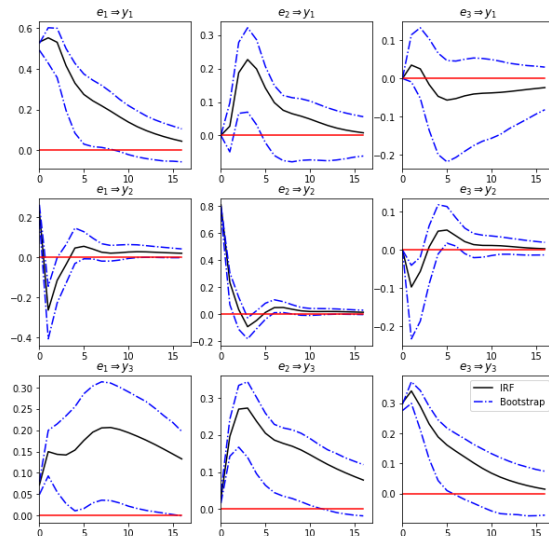
2000년 1월부터 현재까지의 분기별 소비자물가지수, 실질GDP, CD(91일) 변수로 구성된 VAR 모형에서 단기제약하에서 충격반응함수 그려줘. VAR 모형의 시차는 2로 설정해줘.



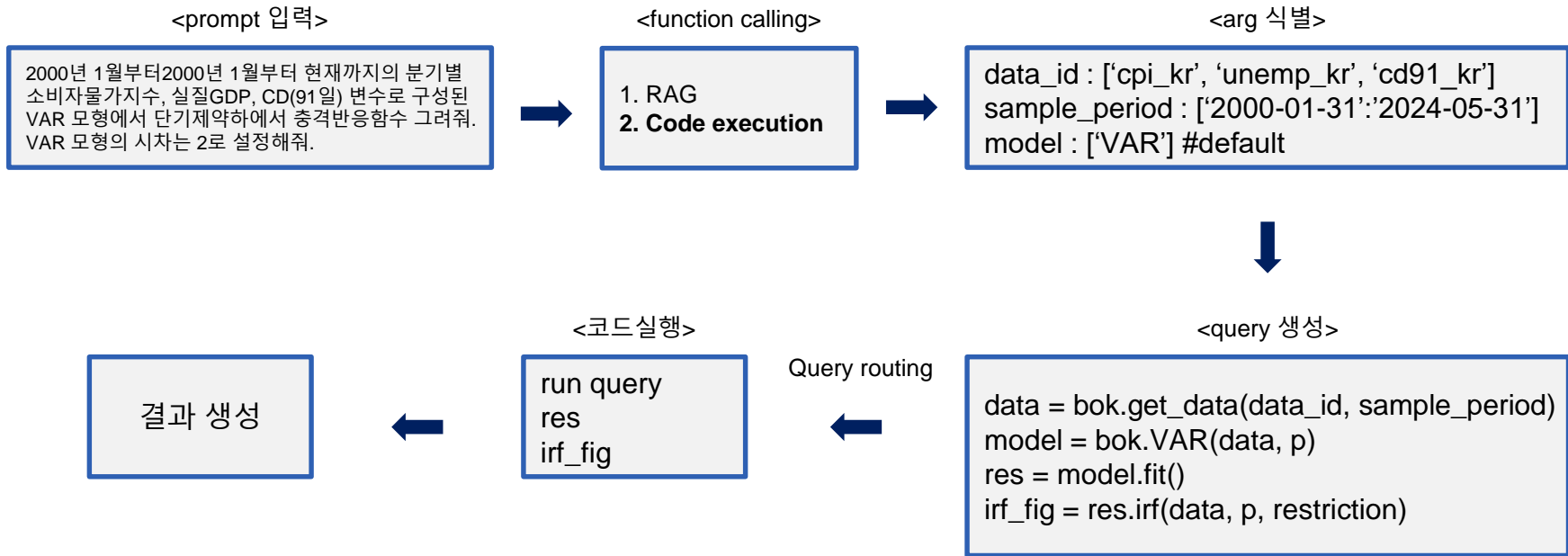
복실이 @boksiri 실시간상당원 시스템계정(로봇) 2024-05-28 9:47:10



복실이 @boksiri 실시간상당원 시스템계정(로봇) 2024-05-30 17:32:55



자연어 기반 데이터 분석: 백그라운드 프로세스



감사합니다.