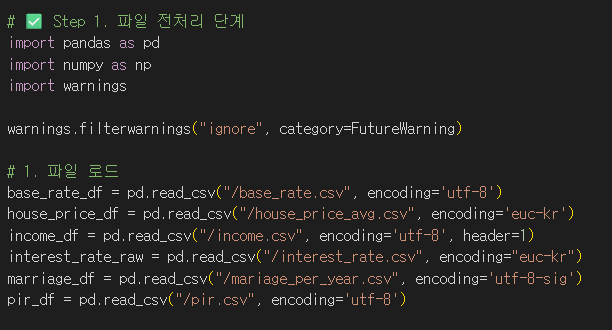
# **빅데이터 융합 개론 – 프로젝트 코드 설명**

작업 환경: Google Colab

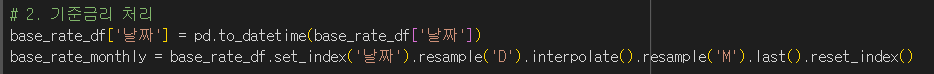
**Step1. 파일 전처리**

1. 파일 업로드 및 불러오기



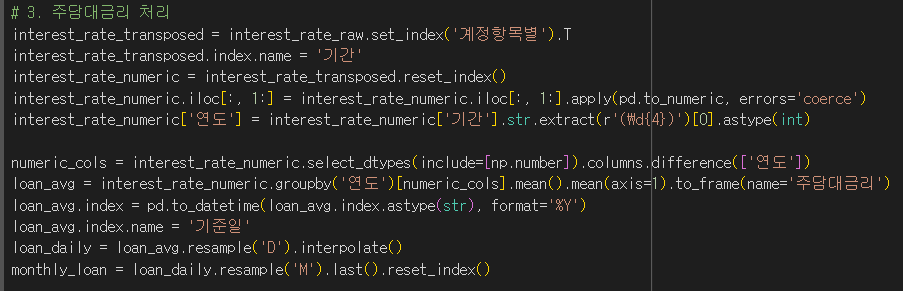
: 세션 저장소에 업로드된 csv 파일들을 불러오는 코드. (warnings 구문은 월보간 시, resample(‘M’) 대신 resample(‘ME’)를 쓰라고 경고하는 메시지 방지 위함)

1. 기준금리 리샘플링



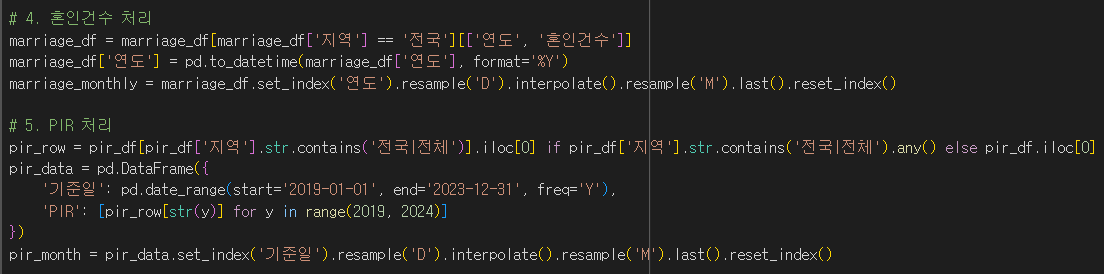
: 불러온 base\_rate.csv (기준금리 데이터)는 연간 데이터이므로, 분석 향상을 위해 .resample(‘M’) 메서드로 월 보간 진행 (매년 -> 매월) 🡪 월 보간 시, 선형 보간을 통해 결측치 처리(월말 기준값으로 인덱스 리셋)

1. 주택담보대출 리샘플링



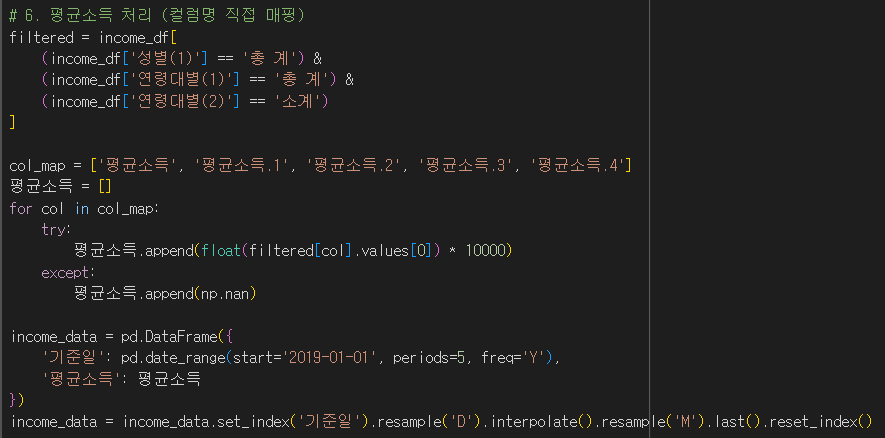
: 마찬가지로 주담대 또한 연데이터이므로, 월보간 진행해줌 (전치 -> 인덱스 리셋 -> 연도 추출 및 연도별 평균 계산 -> 일별 보간 후 월별 리샘플링)

1. 혼인건수 및 PIR지수 리샘플링



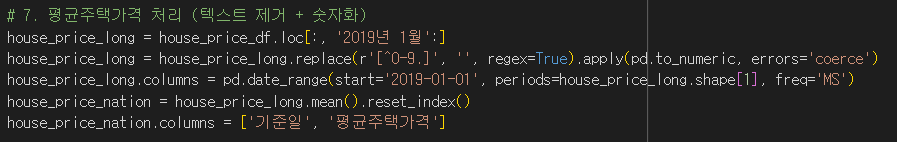
: 혼인건수는 동일. PIR지수는 전국 또는 전체 행을 선택(없으면 첫 행 선택) 후, 연도별 PIR 지수 구성 후 월별로 변환

1. 평균소득 리샘플링



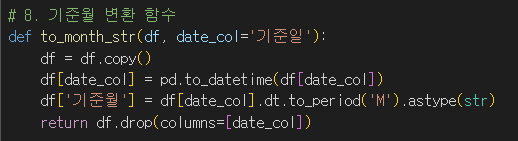
: 총계/소계에 해당하는 행을 필터링 후 평균소득의 단위인 만원을 곱해주고, 예외 시 NaN 처리 후, 연도별 평균소득 df 생성, 이 데이터로 월 보간 진행

1. 평균주택 매매가 리샘플링



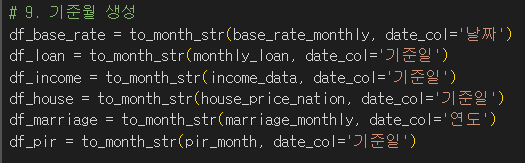
: 첫 줄에서 연속된 월별 데이터만 추출(데이터는 월별로 이미 전처리됨). 숫자 외의 문자는 제거하고 숫자형으로 전부 변환해줌(text -> numeric). 컬럼 이름을 날짜로 지정(start=””, periods=””, freq=’’)한 후에 인덱스를 리셋(reset\_index())하고 사용할 평균주택가격 데이터 구성(마지막 줄)

1. 기준월 변환 함수 정의( to\_month\_str )



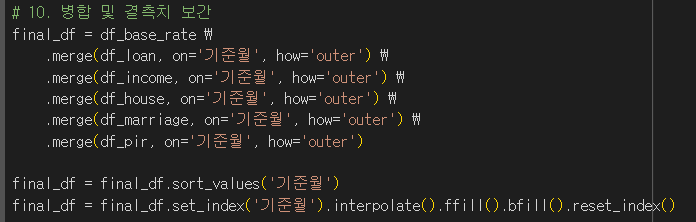
: 데이터프레임과 데이터프레임의 ‘기준일’ 컬럼을 인자값으로 리시브. ‘df’라는 변수에다가 받은 데이터프레임을 복사한 후, df의 ‘기준일’에 해당하는 컬럼에 기준월이라는 컬럼으로 전부 형식화(기준일 삭제 및 ‘2020-01’ 형식으로 변환, 기준월만 반환-> return)

1. 보간된 데이터들 전부 기준월 변환



: to\_month\_str() 함수를 통해 ‘기준월’ 컬럼으로 형식화 된 데이터 얻음

1. 데이터 병합 및 정렬 후 전체 선형 보간(결측치 처리)



: ‘final\_df’라는 데이터 프레임을 모든 보간된 데이터를 병합시킨 데이터 프레임으로 구성.

기준월을 기준으로 월별 정렬을 한번 한 후에, 전체에 대해 선형 보간을 진행(앞으로 채우고(.ffill()) 뒤로 채우고(.bfill()) 인덱스 리셋(.reset\_index())

1. 최종 통합 데이터 프레임 출력해서 확인해보기

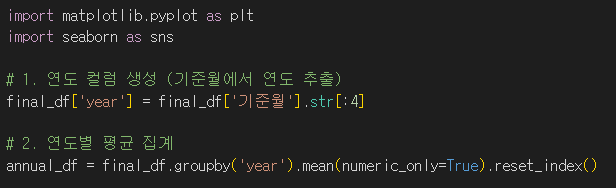




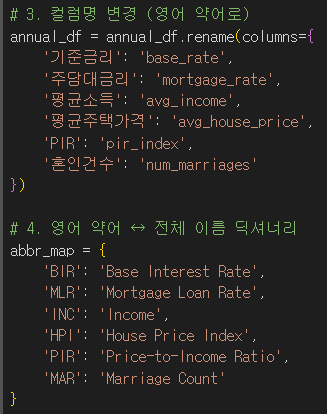
: 6개 행(2019 – 2024) 에서 72개 행(6 years \* 12 months)으로 잘 보간됨.

**Step2. 데이터 시각화(데이터 추이 확인용) – matplotlib과 seaborn 활용**

1. 연도별 추이를 위한 연도 추출

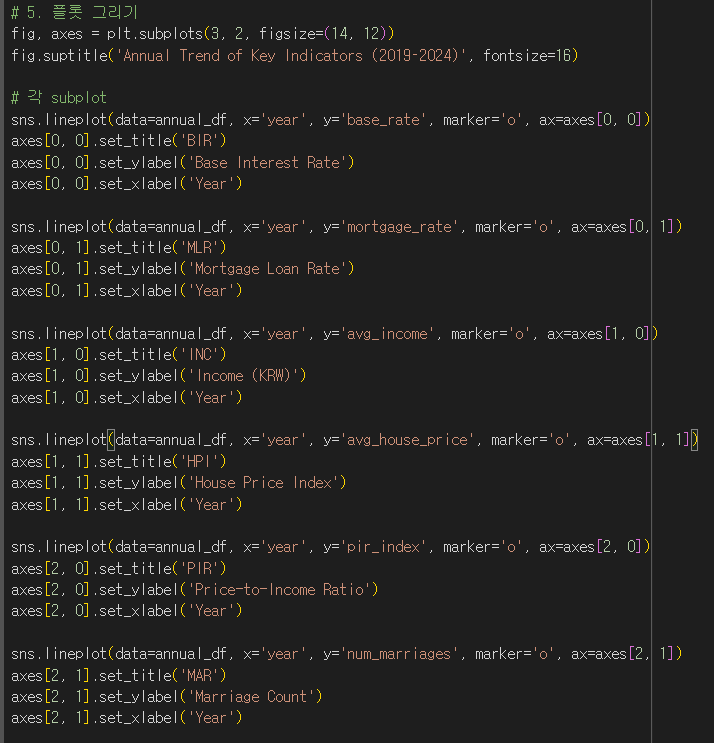


: final\_df 에는 기준월 데이터로 정렬되어있음 -> year이라는 컬럼을 생성하여 연도별 평균을 칼럼값으로 (numeric data로) 집계 -> annual\_df 로 생성

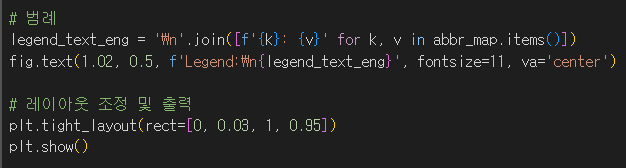
1. 코랩환경에서의 한글 깨짐 문제 현상 -> 영어 약어로 정의

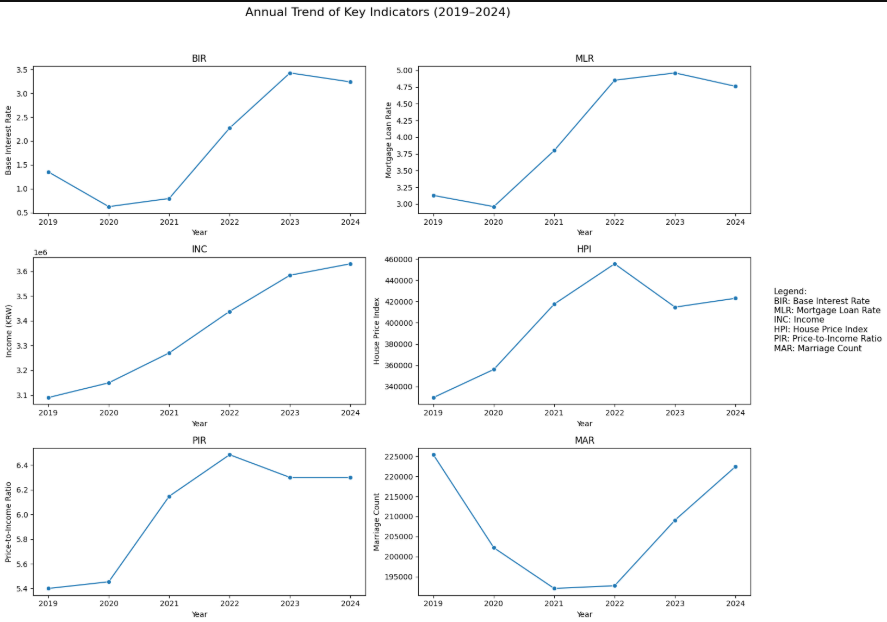
: annual\_df의 컬럼명을 영어로 변환(한글 오류 해소) 및 약어 딕셔너리 생성(플롯 및 범례에 사용)

1. 시각화 – subplot을 통해 6개의 그래프(3행 2열) 생성



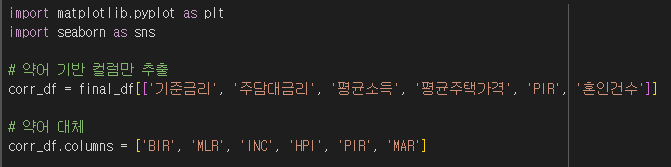
1. 범례 생성 및 레이아웃 조정 -> 출력





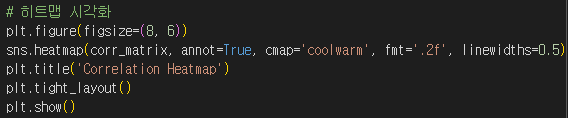
**Step3. 상관관계 히트맵 – 변수 관계 파악하기**

1. 약어 추출 및 간략화

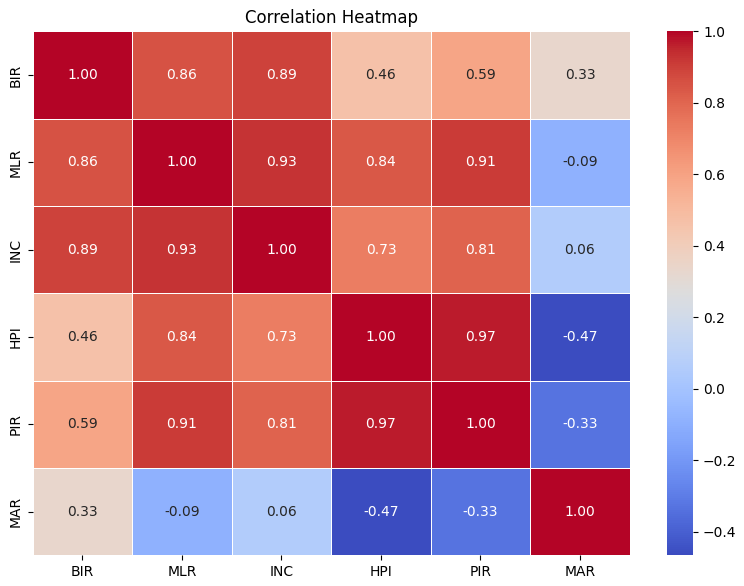


: final\_df 으로부터 각 컬럼을 추출 -> 약어컬럼의 기반이 됨 -> corr\_df 생성 후 각 컬럼명을 더 간략화된 약어로 대체

1. 상관행렬 계산 및 히트맵 출력



: 히트맵의 경우, seaborn의 .heatmap() 함수 활용

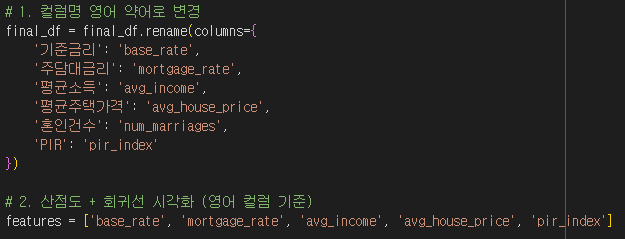


1. 분석결과 분석

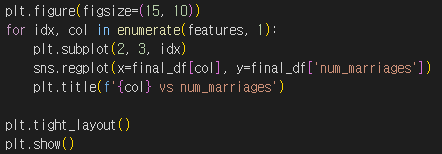


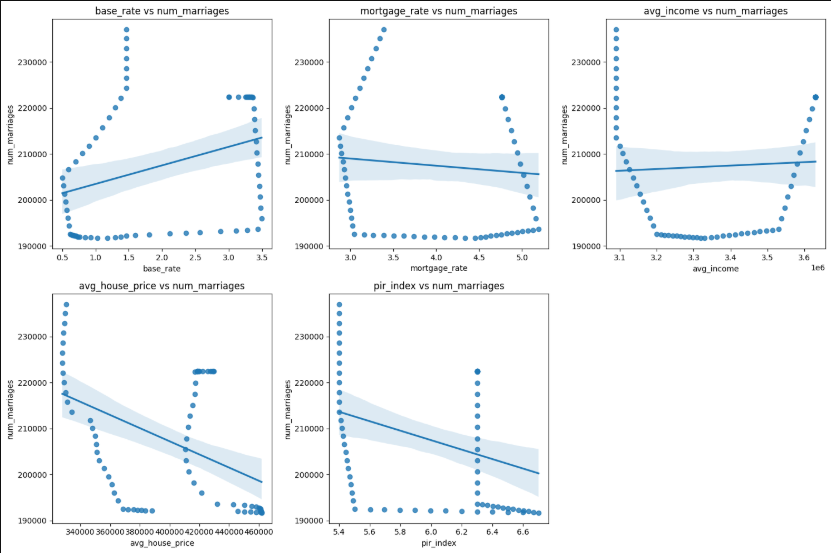
**Step3. 상관관계 분석 시각화**

1. 컬럼 영어 약어 작업 및 특징 컬럼 행벡터 취하기

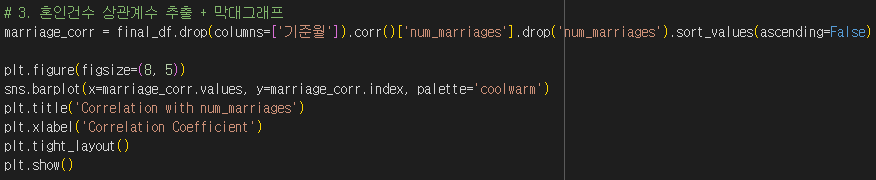


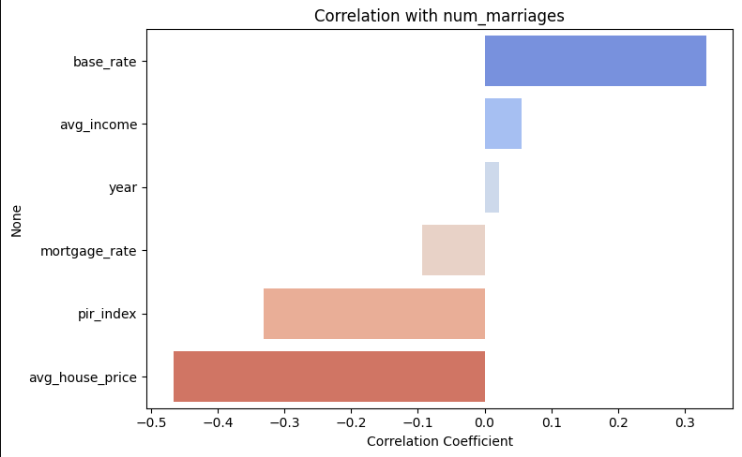
1. 산점도와 회귀선을 통해 각 변수간 상관관계 시각화





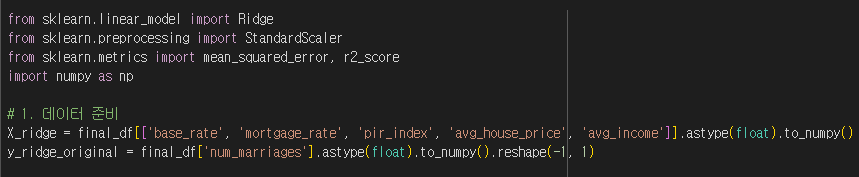
1. 혼인건수와의 상관관계를 막대그래프로 시각화(hot vs cold)





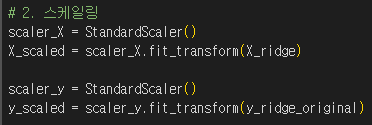
Step4. 스케일링 및 회귀분석 -1. 릿지회귀

1. 스케일링 준비



: 릿지회귀 모델 및 정규화모델(StandardScaler)을 sklearn에서 import하고 데이터 정의

1. 스케일링

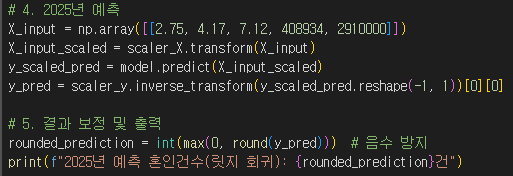


: x(입력값)과 y(출력값)에 따로 스케일링(정규화)을 적용

1. 릿지회귀 모델 학습

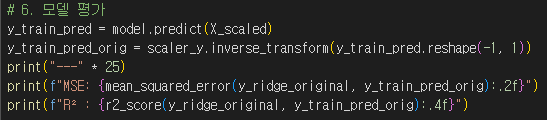


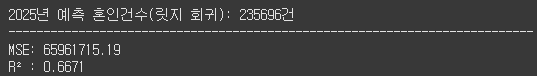
1. 모델로 2025년 혼인건수 예측 및 결괏값 역정규화(보정)

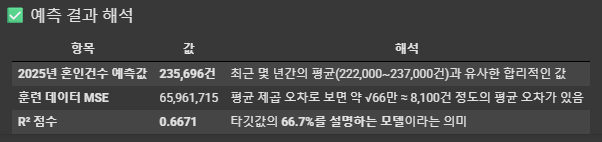


: 2025년의 데이터(현재 기준)를 입력값으로 정의(정답 없음)하여 예측(model.predict())하고, y\_pred에 inverse\_transtorm()하여 보정한 값을 y\_prediction에 저장

1. 결괏값 평가

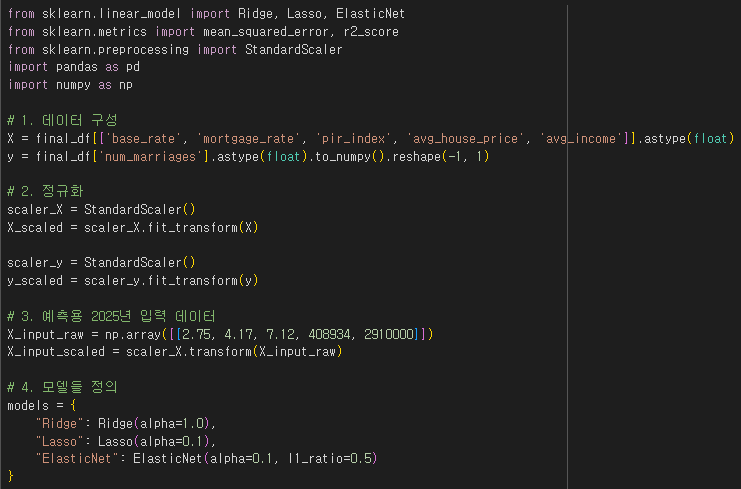






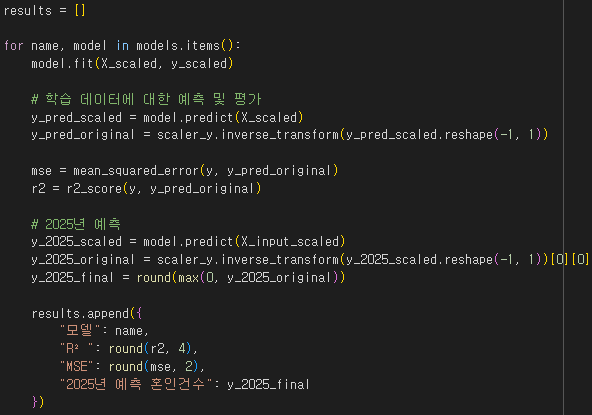
**Step4-2. 규제회귀 비교해보기**

1. 데이터 및 모델 정의



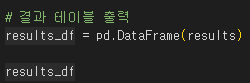
: 릿지, 라쏘, 엘라스틱넷을 서로 비교해보기

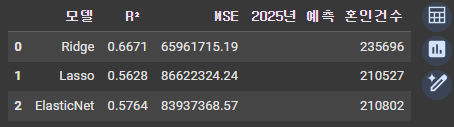
1. 학습 및 평가



: predict(), scaled -> inverse\_transform(), result.append() 하여 결괏값 테이블 생성

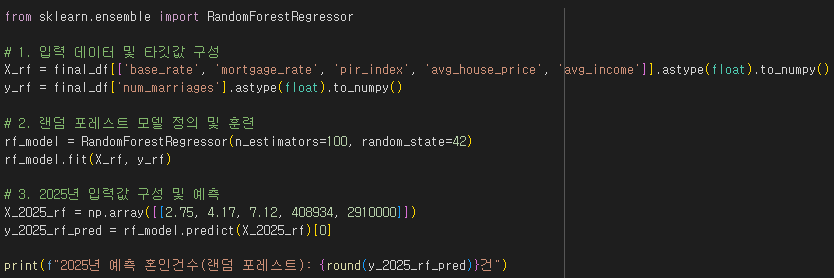
1. 출력





: 릿지 회귀가 가장 적절함

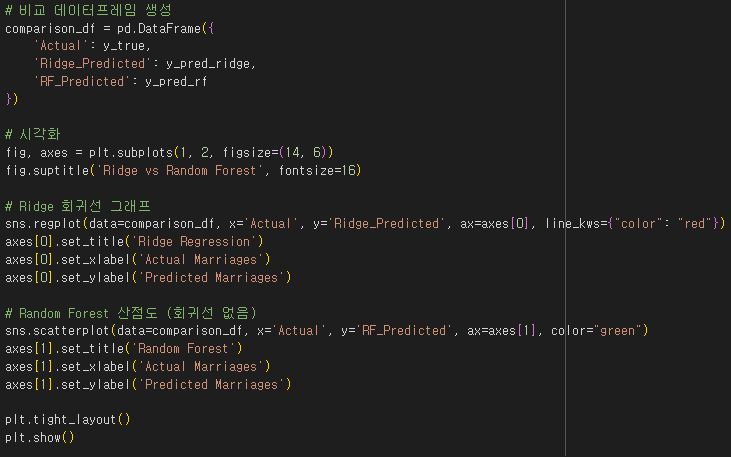
**Step4-3. 랜덤포레스트 적용**



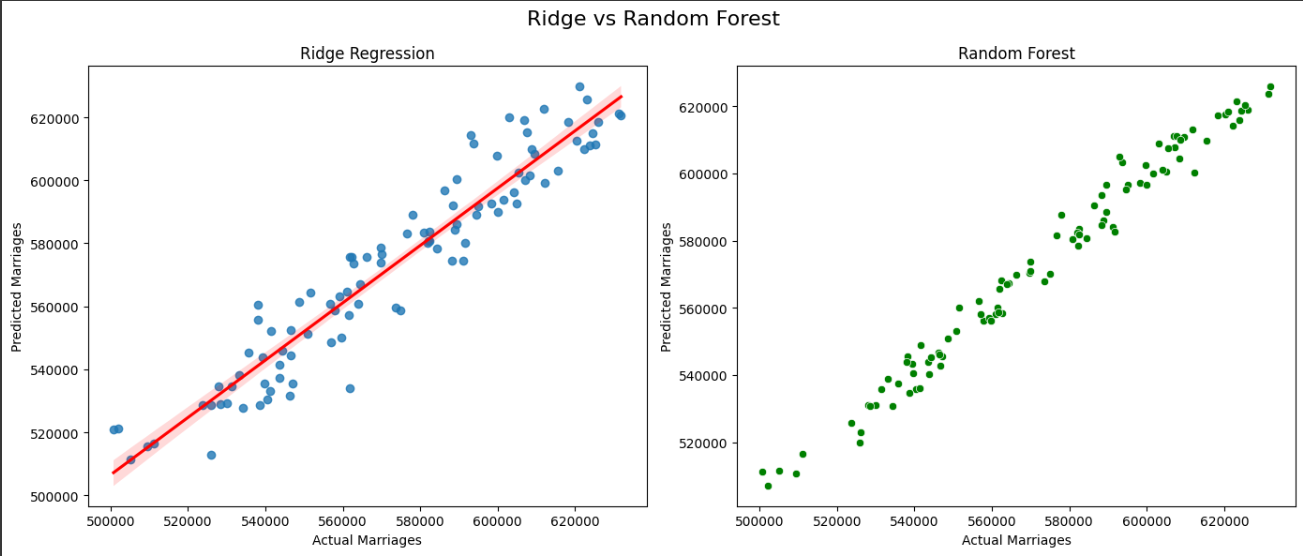


: 릿지와 랜덤포레스트의 값은 약 2만건 정도 차이나는 것을 확인

**Step5. 두 분석단계를 시각화하여 비교**

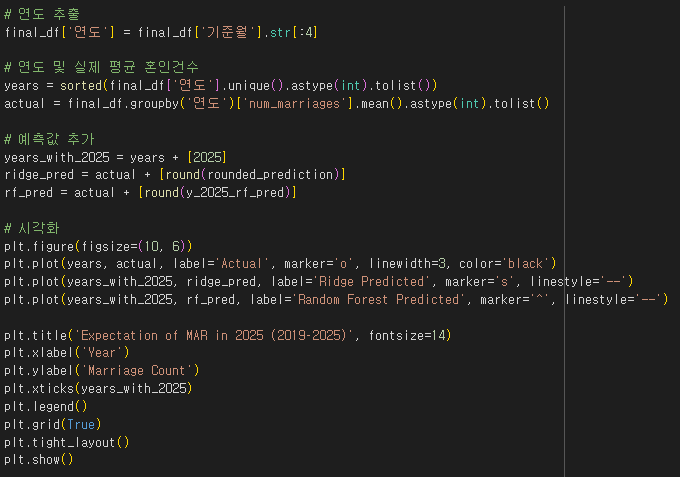


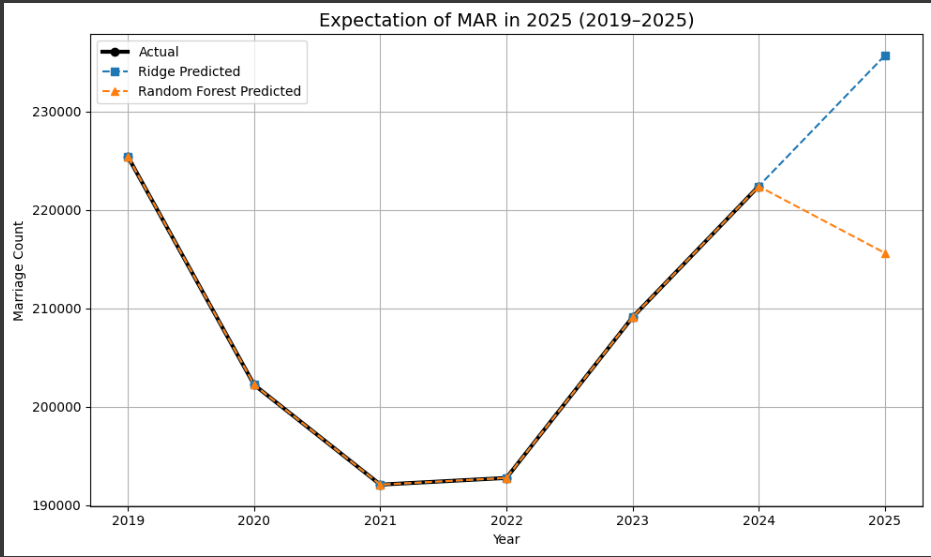
: comparison\_df에 2019~2024 + 25예측값들을 전부 넣어 실측값, 릿지회귀값, 랜덤포레스트값으로 데이터프레임 구성. 이 데이터프레임을 시각화



: 이렇게 비교해서는 딱히 차이점을 못느낌을 알게됨. 🡪 다르게 시각화 시도

**Step6. 예측값을 추이기반 꺾은선 그래프로 비교**

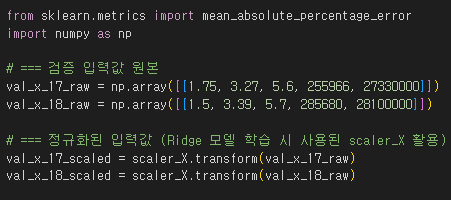




: 릿지회귀는 추이를 따라가고, 랜덤포레스트는 그렇지 않음을 확인. 그러나, 현실에서의 상황을 고려한다면, 증가도, 감소도 큰폭이 이루어지지 않고, 감소한다면 했지, 증가는 안할거 같음(경험적 데이터). 그러므로 일단 두 방식의 검증이 필요함.

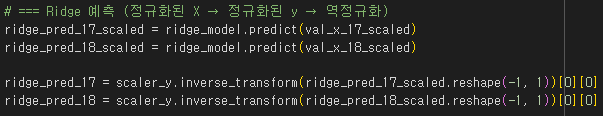
**Step7. 검증단계(MAPE 사용)**

1. 입력값 정의



: 검증데이터는 정답이 있는 2017년과 2018년도 데이터를 활용(이 역시 정규화 진행)

1. 릿지회귀



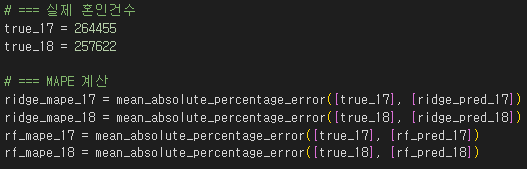
: 정규화된 x에 대해 예측 진행하고, 정규화된 예측 y를 다시 역정규화로 복원

1. 랜덤포레스트



: 얘는 정규화가 필요없어서 바로 predict()

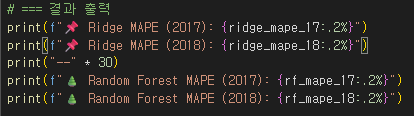
1. 정답 정의 및 MAPE 계산

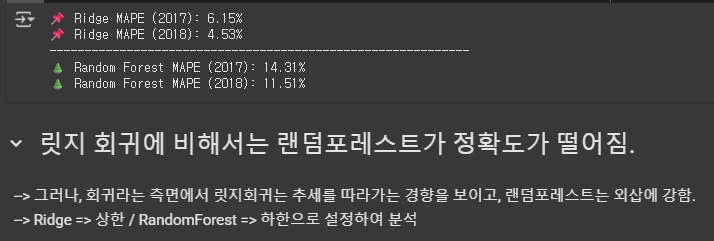


: 2017년과 2018년의 정답 데이터를 기반으로 mean\_absolute\_percentage\_error() 진행

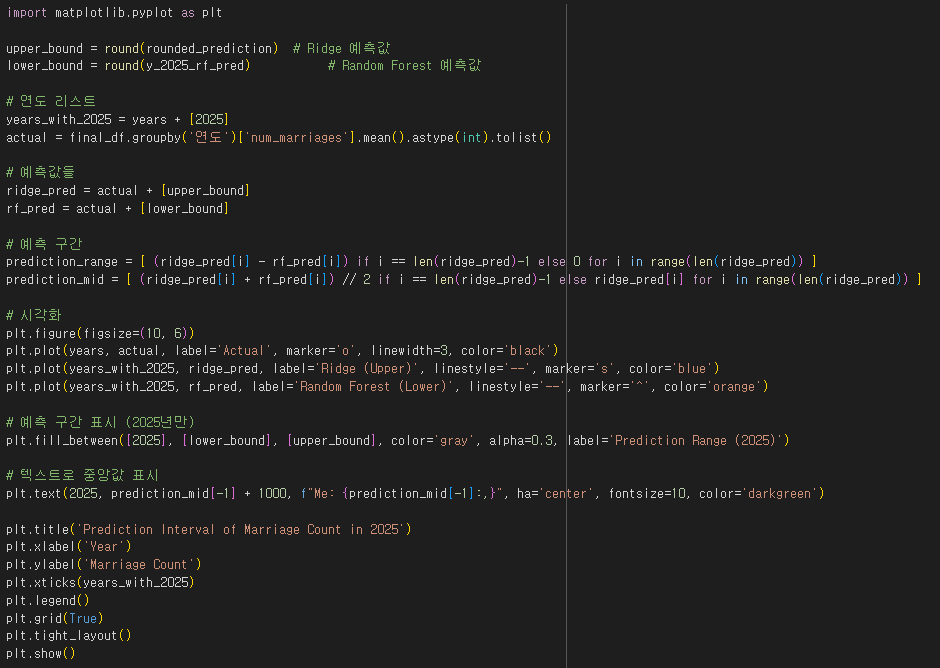
* 사용이유: 비율로 한눈에 파악하기 쉬움.

1. 검증 출력

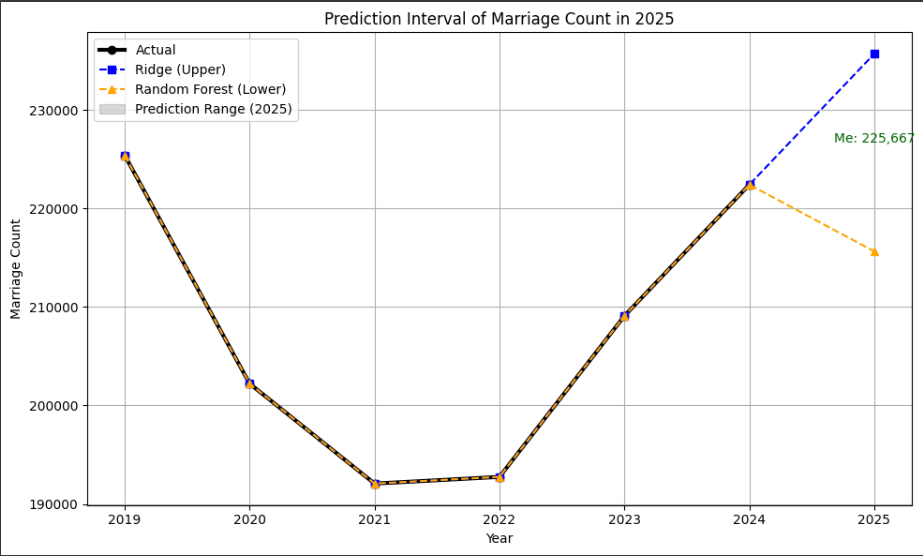




Step8. 최종 결괏값 출력 및 시각화



: 코드에 관해서는 위의 내용들을 참조할 것. (비슷한 과정 위에 똑 같은 과정 존재)  
🡪 예측값 시각화단계인 step6 : ref.



: 상한/하한을 기점으로 중앙값(Me)를 책정하여 그래프위에 텍스트 기반으로 표시

* Plt.text() 구문 확인
* 최종 예측값은 225,667(건)으로, 전년도 기준 약 1.5% 정도 증가할 것으로 예상됨.