

# Portfolio maker

---

# Containing

---

create\_mapper: CIK, ticker mapper 생성

---

collect\_financial\_data: 재무 정보 정리

---

collect\_price\_and\_volume: 가격, 거래량 정보 수집

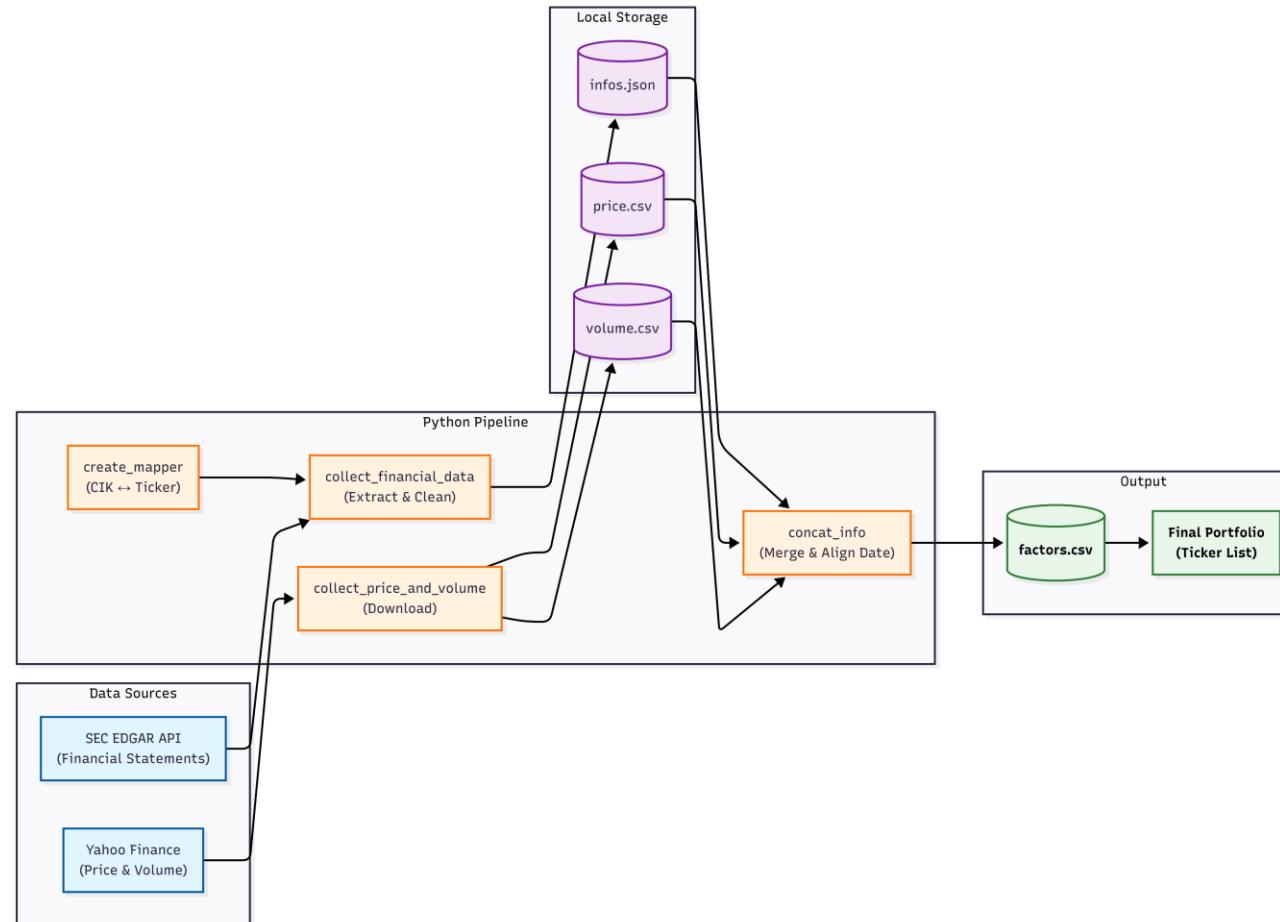
---

concat\_info: 가격과 재무 정보 날짜에 맞게 결합

---

make\_simple\_portfolio: 결합된 데이터를 바탕으로 간단한 factor에 기반한 포트폴리오 생성

# Project Overview



# 재무 데이터

---

SEC의 EDGAR system 이용하여 SEC에 기업이 제출한 재무  
데이터 수집

---

EDGAR에서는 기업의 이름을 CIK라는 고유한 숫자 일련번호로  
관리합니다.

---

CIK <-> ticker 변환기 필요. (create\_mapper() 함수를 통해 생성)

# 재무 데이터

<https://www.sec.gov/search-filings/edgar-application-programming-interfaces> 사이트에 접속하여 재무 파일 다운로드

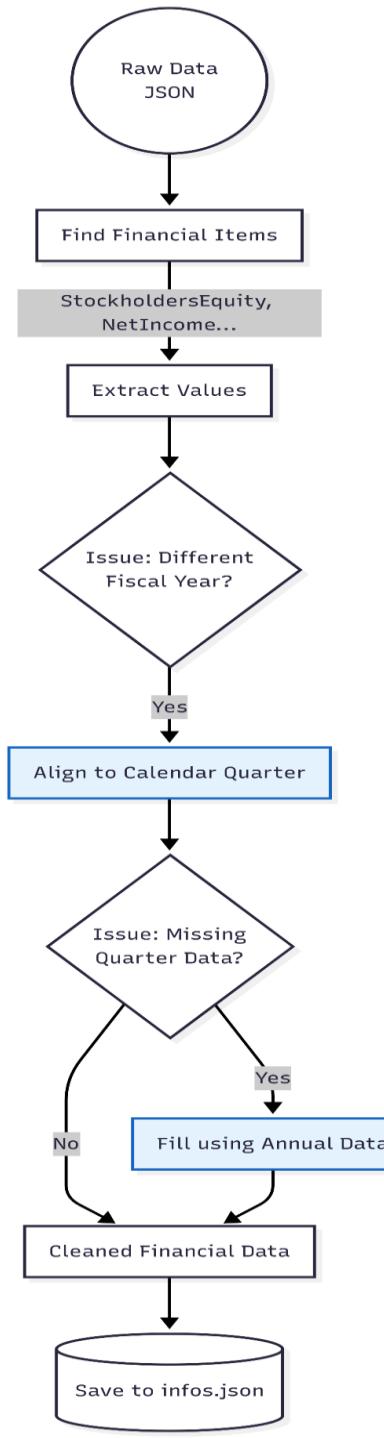
## Bulk Data

The most efficient means to fetch large amounts of API data is the bulk archive ZIP files, which are recompiled nightly.

- The [companyfacts.zip file](https://www.sec.gov/Archives/edgar/daily-index/xbrl/companyfacts.zip) contains all the data from the XBRL Frame API and the XBRL Company Facts API  
<https://www.sec.gov/Archives/edgar/daily-index/xbrl/companyfacts.zip>
- The [submission.zip file](https://www.sec.gov/Archives/edgar/daily-index/bulkdata/submissions.zip) contains the public EDGAR filing history for all filers from the Submissions API  
<https://www.sec.gov/Archives/edgar/daily-index/bulkdata/submissions.zip>

이 파일은 json 형식으로 되어 있습니다. 구체적인 재무 데이터 정보는 'facts' 항목의 'us-gaap' or 'ifrs-full' 항목에서 해당하는 재무 항목을 찾으면 제출 날짜, 시작(duration이 필요한 값들만), 마감 날짜, 수치값 등등의 정보를 얻을 수 있습니다.

# 재무 데이터 처리 과정



# 재무 데이터

재무 항목의 종류는 생각보다 매우 많으며 SEC에서는 그것들을 따로 정리하여 나누지 않기 때문에 어떤 이름에 해당되는지를 직접 찾아야 합니다.

Ex)

총자본 = StockholdersEquity

당기순이익 = NetIncomeLoss

```
'UnrecognizedTaxBenefitsIncomeTaxPenaltiesAndInterestExpense',
'UnrecognizedTaxBenefitsIncreasesResultingFromCurrentPeriodTaxPositions',
'UnrecognizedTaxBenefitsIncreasesResultingFromPriorPeriodTaxPositions', 'UnrecognizedTaxBenefitsPeriodIncreaseDecrease',
'UnrecognizedTaxBenefitsReductionsResultingFromLapseOfApplicableStatuteOfLimitations',
'UnrecognizedTaxBenefitsThatWouldImpactEffectiveTaxRate',
'UnrecordedUnconditionalPurchaseObligationBalanceOnFifthAnniversary',
'UnrecordedUnconditionalPurchaseObligationBalanceOnFirstAnniversary',
'UnrecordedUnconditionalPurchaseObligationBalanceOnFourthAnniversary',
'UnrecordedUnconditionalPurchaseObligationBalanceOnSecondAnniversary',
'UnrecordedUnconditionalPurchaseObligationBalanceOnThirdAnniversary',
'UnrecordedUnconditionalPurchaseObligationBalanceSheetAmount',
'UnrecordedUnconditionalPurchaseObligationDueAfterFiveYears', 'VariableLeaseCost',
'WeightedAverageNumberDilutedSharesOutstandingAdjustment', 'WeightedAverageNumberOfDilutedSharesOutstanding',
'WeightedAverageNumberOfSharesOutstandingBasic', 'DeferredTaxAssetsOtherComprehensiveLoss',
'EffectivelyIncomeTaxRateReconciliationFdiiAmount', 'DebtSecuritiesAvailableForSaleRestricted',
'InventoryWorkInProcessAndRawMaterialsNetOfReserves',
'UnrecordedUnconditionalPurchaseObligationDueInRemainderOfFiscalYear',
'InventoryRawMaterialsAndPurchasedPartsNetOfReserves', 'HedgedAssetFairValueHedge', 'HedgedLiabilityFairValueHedge',
'DeferredIncomeTaxAssetsNet', 'DeferredTaxLiabilitiesPropertyPlantAndEquipment',
'IncrementalCommonSharesAttributableToShareBasedPaymentArrangements', 'OtherAssetsMiscellaneousNoncurrent',
'OtherComprehensiveIncomeLossCashFlowHedgeGainLossAfterReclassificationAndTax',
'OtherComprehensiveIncomeLossCashFlowHedgeGainLossBeforeReclassificationAfterTax', 'GeneralAndAdministrativeExpense',
'IncomeTaxReconciliationChangeInDeferredTaxAssetsValuationAllowance', 'SellingAndMarketingExpense'])
```

재무 데이터 항목 예시

# 재무 데이터

- 해당하는 항목을 찾은 후에는 각 기업별로 해당 항목의 수치값을 년도, 분기별로 정리를 해야 합니다.
  - > 이 과정에서 몇가지 문제가 있는데, 분기 구분이 모든 기업이 전부 통일되어 있는 우리나라와는 다르게 미국의 기업들은 기업들마다 분기 구분이 다릅니다. 이를 해결하기 위해 마감 날짜를 기준으로 우리나라식으로 분기를 구분을 하는 작업이 필요합니다.
  - > 또한 1년 기준 재무제표가 제출되어 있는 경우 인접한 quarter의 데이터가 생략된 경우가 빈번하게 발생하기 때문에 annual data를 이용하여 비어있는 quarter 데이터를 채워주는 작업도 필요합니다.
  - > 또한 기업들마다 재무 항목을 지칭하는 명칭이 다른 경우가 있으므로 이것도 반영을 해 주어야 합니다.
- 위의 모든 작업들을 수행하여 정리된 형태로 만들어주는 것이 `collect_financial_data()` 함수입니다. 작업을 완료한 후에는 해당 정보를 `infos.json` 파일에 저장합니다.
  - > 현재 코드에서 해당 함수는 2개의 재무 항목(총자본, 당기순이익)과 발행 주식 수에 대해서만 진행을 하였습니다.

# 가격 데이터

- `collect_price_and_volume()` 은 yahoo finance에서 각 기업들의 일일 수정 종가와 거래량 정보를 다운받습니다.
- 정보를 각각 `price.csv`, `volume.csv` 파일에 저장합니다.
- 각 파일은 날짜-행, 종목-열 형식으로 되어 있습니다.

# 결합

- concat\_info 함수는 가격 데이터와 재무 데이터를 결합합니다.
- 가격 데이터는 매일 업데이트가 되는데 각 날짜별로 현재 접근할 수 있는 재무 데이터 정보를 확인하여 반영해 줍니다.  
: 재무 데이터의 마감일이 아니라 데이터가 제출되어 시장에 공개된 날짜를 기준으로 해 주어야 합니다.
- 결합된 데이터를 factors.csv 파일에 저장합니다.

# 간단한 factor 실험

- experiments.ipynb 파일은 factors.csv파일에 저장된 데이터를 바탕으로 간단한 전처리를 진행한 후 간단한 factor 실험을 진행합니다.
- 현재 코드에서는 pbr, per를 기준으로 투자를 진행하였을 경우 각 팩터(pbr, per)값이 높은 종목들과 낮은 종목들로 포트폴리오를 구성하여 투자했을 경우의 백테스트 수익률을 보여줍니다.
- Rebalancing 주기는 1년이며 매 달마다 투자를 진행합니다. 수익률을 기하평균 수익률을 이용하여 계산하여야 합니다. (메트릭 스튜디오 참고)

# 포트폴리오 구성

- `make_simple_portfolio` 함수는 pbr을 기준으로 pbr 값이 하위 10% 이하인 종목들을 이용하여 포트폴리오를 구성하고 해당 종목들을 알려줍니다.
- 종목들은 `ticker_list.txt` 파일로 저장합니다.