2024 3학년 1학기 빅데이터 기초 Term 프로젝트

Economic Status

건국대학교 글로컬캠퍼스
202020881 소프트웨어전공 나성곤

CONTENTS

시연 + 프로젝트 구조

시연 페이지

Google Colab

GitHub

Streamlit

개요

● 프로젝트 목적

- 자신의 경제적 상태를 객관적으로 파악할 수 있도록 합니다.
- 내 자산을 과거의 경제적 흐름을 파악함으로써 경제적인 흐름을 읽을 수 있도록 합니다.

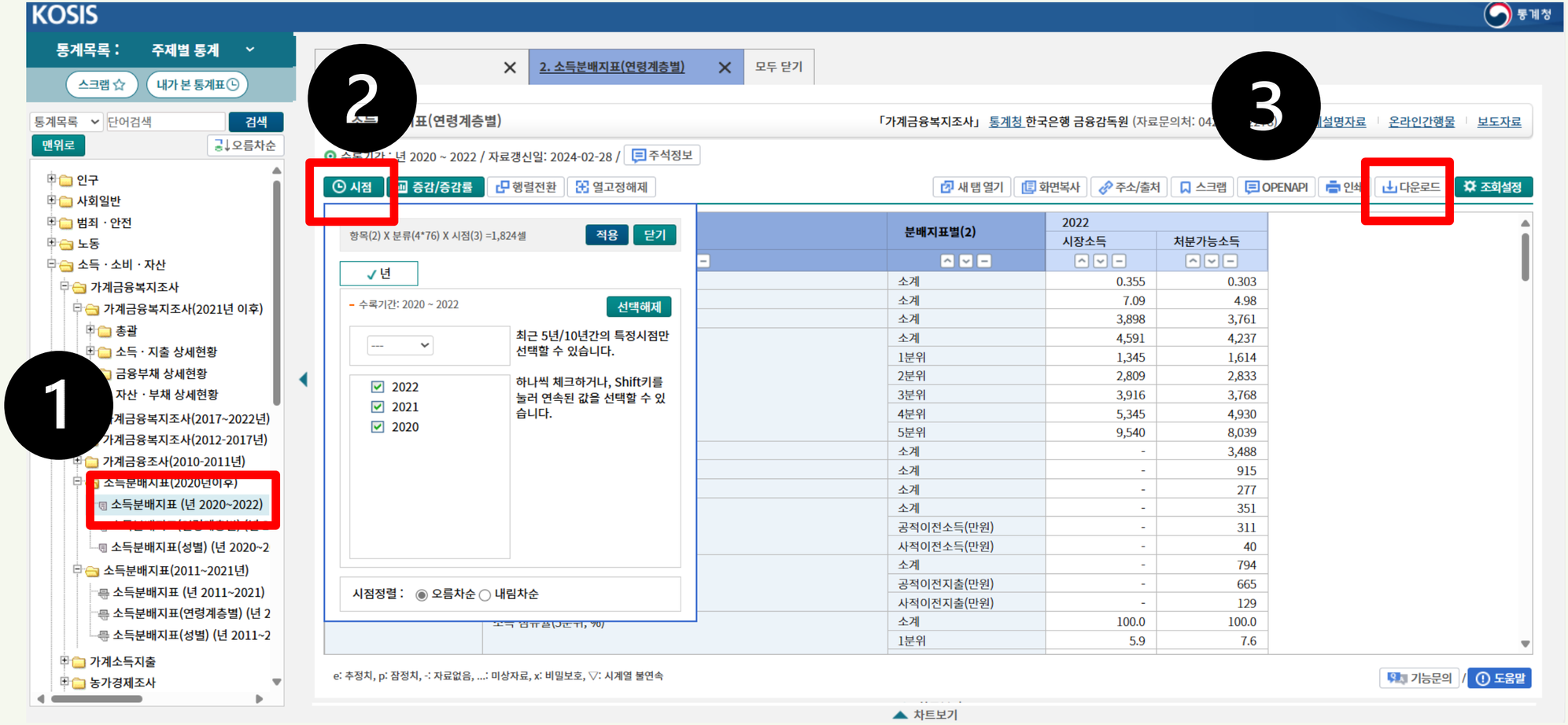
● 프로젝트 주요 과정

- 데이터 수집 : KOSIS를 활용하여 필요한 데이터를 수집합니다.

- 데이터 분석 접근법 : 상향식 분석 접근법을 이용하여 데이터들의 관계를 파악하여 기준을 정합니다.

- 분석 방법
  1. 서술 분석 : 과거의 데이터를 기준으로 사용자의 입력데이터가 어떤 경제적 위치를 가지고 있는지 시각화를 하여 보여줍니다.
  2. 예측 분석 : 귀 분석을 통해 도출된 회귀식을 사용하여 예측 모델을 만들고, 이를 통해 미래의 자산규모나 소비 습관에 따른 소득 분위를 예측합니다.

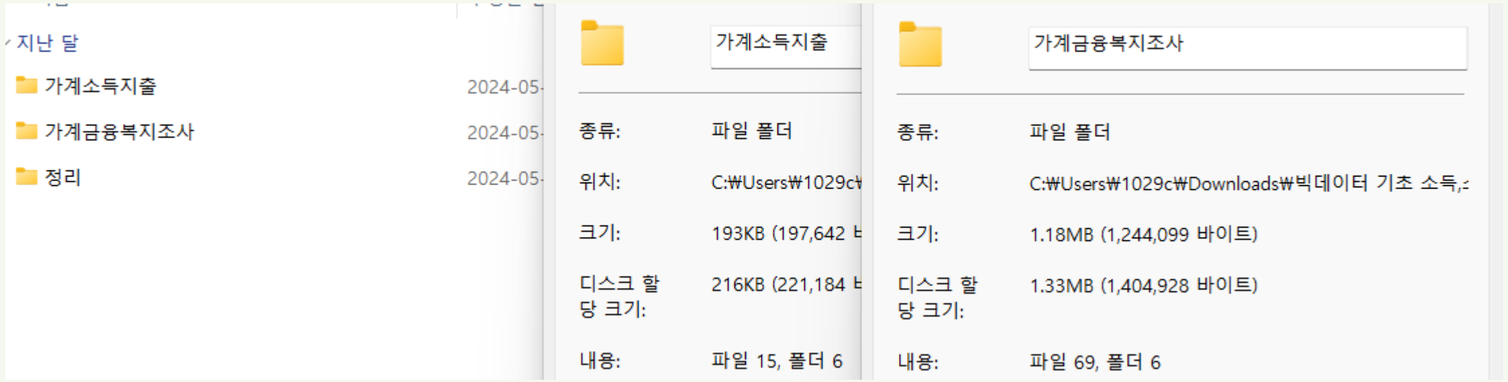
데이터 수집



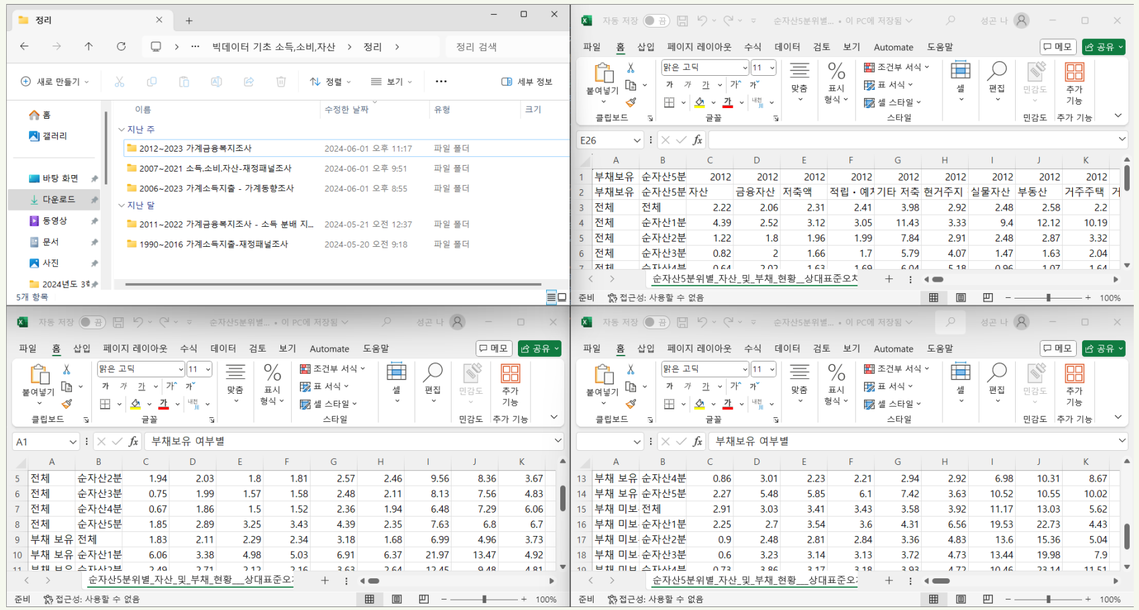
csv 파일로 다운을 받습니다.

주제에 맞는 데이터를 찾습니다. 소득 소비 자산에서 직,간접적으로 연관이 있는 자료들을 모두 찾습니다.

데이터 통합 (1)



주제와 관련된 약 84개의 csv 파일을 다운을 받아 분류로 저장을 하였습니다.

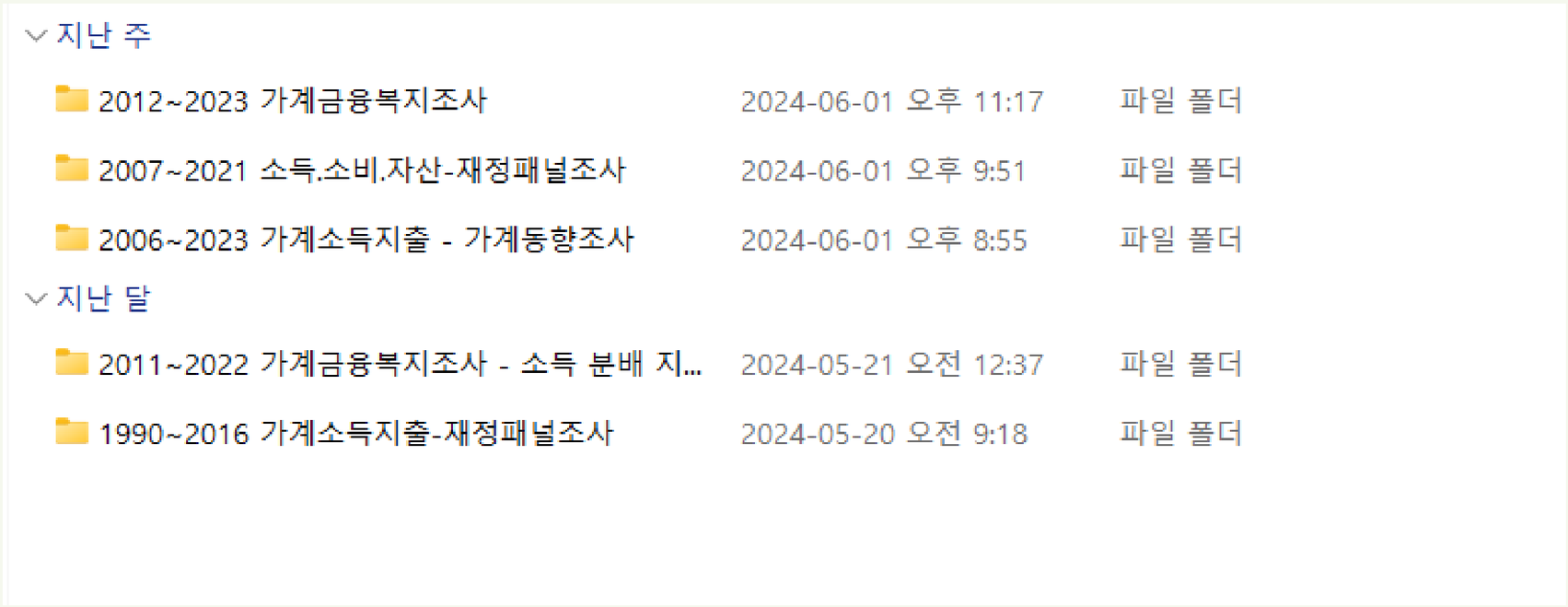


<-파일 통합할 때 노트북 화면

연도에 따른 데이터를 같은 데이터로 합쳐 하나의 데이터로 만들었습니다.

엑셀을 사용한 이유
1.데이터 구성이 멀티 인덱스로 되어 있으며, 파일마다 인덱스의 개수가 다릅니다.
2.하나의 파일로 합치는데 코드를 사용하는 것보다 엑셀을 사용하여 합치는 것이 시간적으로 빠르게 처리할 수 있습니다.

데이터 통합 (2)



하나의 파일 제목을 토대로 중복되는 연도를 제외하면서 파일을 합친 결과 입니다.

데이터 통합 (3) - 전체 데이터목록 (35개)

소득5분위별_신고유형별_소득공제액_특별공제__조특법상_공제

소득5분위별_연간_비소비지출

소득5분위별_연간_소비지출_

소득5분위별_자산_및_부채_순자산__

소득5분위별_연간_소득원천별_가구소득_

소득5분위별_가구주_특성_

소득5분위별__가구당_가계수지__전국_1인이상_실질

가구원수별__가구당_월평균_가계수지__전국_1인이상_실질

가구당_월평균_가계수지__전국_1인이상_실질

소득분배지표

소득분배지표_연령계층별

소득분배지표_성별

소득분배지표_전체가구__성별_및_연령구분별__

소득분배지표_

가구원수별_자산__부채__소득_현황

가구주_성별_자산__부채__소득_현황

가구주연령계층별_10세__자산__부채__소득_현황

가구주_혼인상태별_자산__부채__소득_현황

가구주_교육정도별_자산__부채__소득_현황

가구주_종사상지위별_자산__부채__소득_현황

소득5분위별_자산__부채__소득_현황

자산5분위별_자산__부채__소득_현황

순자산5분위별_자산__부채__소득_현황

소득5분위별_가구주_종사상지위별_자산__부채__소득_현황

소득5분위별_자산5분위별_자산__부채__소득_현황

소득5분위별_순자산5분위별_자산__부채__소득_현황

가구주연령계층별_10세__가계재무건전성

가구주종사상지위별_가계재무건전성

소득5분위별_가계재무건전성

자산5분위별_가계재무건전성

순자산5분위별_가계재무건전성

순자산__가구소득의_분위별_평균__점유율_및_경계값

소득5분위별_자산_및_부채_현황__상대표준오차

자산5분위별_자산_및_부채_현황__상대표준오차

순자산5분위별_자산_및_부채_현황__상대표준오차__

데이터 관계 분석

	순자산	자산	부채	소득	소비	
연령	○■	○■	○■	○■		순자산 5분위
종사자 지위	○	○■	○■	○■		자산 5분위
성별	○■	○■	○■	○■	■	소득 5분위
학력	○	○	○	○		
가구원 수	○	○	○	○		
혼인유무	○	○	○	○		

데이터 관계 분석을 통하여 어떤 데이터를 사용자에게 설명하기 좋은 인지 찾는 과정을 거치고
이 과정을 통하여 객관적인 기준을 정하는 것은 소득 5분위로 결정을 하게 되었습니다.

데이터 탐색

사용할 데이터 정리

시각화 구현

소득5분위별_자산_및_부채_순자산__20240520010244
소득5분위별_연간_소득원천별_가구소득_20240520010229
소득5분위별_연간_소비지출_20240520010258

소득5분위별_가구주_특성_20240520010215

소득5분위별_연간_소비지출_20240520010258
소득5분위별_가계재무건전성

1 <- 소득분위별 순자산,부채,소득,소비 (서술분석)

2 <- 소득 분위 및 조건을 통한 경제적 위치 확인 (서술분석)

3 <- 자산,소비에 따른 소득분위 예측 (서술 + 예측 분석)

소득 5분위을 기준으로 하여 데이터를 나눕니다.
1.순자산,자산,부채,소득,소비지출을 통해 각 소득분위별 확인하고, 2.내 소득분위가 소득이 있는 근로자 중 어느정도 비중을 차지하는지 확인하고, 3.자산과 소비를 입력하여 미래의 소득을 확인하고 그때의 소득분위를 확인하도록 합니다.

데이터 탐색 - 정제 [소득 분위 및 조건을 통한 경제적 위치 확인 / 자산,소비에 따른 소득분위 예측]

데이터 정제를 거침으로써 필요한 부분만 데이터를 가져오도록 합니다.
예시: 소득분위 마다 얼마의 소득이 있는지 중요하지만, 소득 분위의 평균값은 필요하지 않습니다.

소득5분위별_가구주_특성_20240520010215.c... 2024-06-01 오후 3:04

```
"소득분위별(1)","특성별(1)",특성별(2),2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021
"가구전체","성별",남성,77.8,77.7,77.6,77.6,73.1,72.5,71.9,71.6,71.1,69.6,69.0,68.8,68.1,68.2,66.3
"가구전체","성별",여성,22.2,22.3,22.4,22.4,26.9,27.5,28.1,28.4,28.9,30.4,31.0,31.2,31.9,31.8,33.7
"가구전체","연령별",20대 이하,7.7,6.7,5.6,4.6,4.7,3.9,4.0,4.0,4.1,4.1,4.9,5.5,6.1,6.9,7.2
"가구전체","연령별",30대,24.3,24.1,24.8,24.4,23.3,19.5,18.3,16.8,17.1,16.7,15.9,15.6,15.6,14.8,14.3
"가구전체","연령별",40대,26.4,26.3,25.6,25.7,24.8,27.5,27.5,27.7,26.5,27.4,25.8,24.8,23.0,21.6,22.3
"가구전체","연령별",50대,19.1,20.0,20.2,21.0,22.0,17.8,18.0,18.3,18.4,17.6,17.4,17.5,19.7,17.3,16.8
"가구전체","연령별",60대 이상,22.5,22.9,23.9,24.3,25.2,31.4,32.2,33.1,33.9,34.2,36.0,36.5,35.6,39.3,39.5
"가구전체","학력별",중졸이하,30.2,27.8,25.6,24.7,25.2,26.9,26.5,25.7,25.3,24.2,24.3,23.3,21.3,22.5,21.1
"가구전체","학력별",고교재학/고졸,33.6,34.7,34.6,34.2,34.3,33.4,33.1,33.3,32.4,32.0,32.1,31.6,32.6,32.4,32.3
"가구전체","학력별",대재이상,36.1,37.4,39.8,41.1,40.5,39.7,40.5,41.0,42.3,43.8,43.6,45.1,46.1,45.1,46.6
"가구전체","종사자지위별",임금 근로자,43.5,46.1,47.2,48.2,48.8,45.3,47.6,46.5,48.4,48.5,48.8,49.3,50.9,49.7,50.8
"가구전체","종사자지위별",일용 근로자,8.7,7.8,7.7,7.3,6.3,7.0,5.5,6.3,6.2,6.2,5.3,5.7,6.0,5.4,5.5
"가구전체","종사자지위별",고용원이 없는 자영업자,23.5,21.7,22.5,21.3,21.2,22.0,21.1,21.4,20.2,20.3,20.4,18.8,19.4,19.0,19.0
"가구전체","종사자지위별",고용원을 둔 사업주,4.8,4.5,4.3,4.7,3.5,3.7,3.8,3.2,3.3,3.0,2.8,3.3,3.9,3.2,2.9
"가구전체","종사자지위별",무급가족 종사자,0.4,0.7,0.5,0.5,0.5,0.7,0.5,0.6,0.5,0.4,0.5,0.7,0.6,0.6,0.6
"가구전체","종사자지위별",전업주부/학생/무직,19.1,19.1,17.9,18.0,19.7,21.4,21.4,22.1,21.4,21.6,22.2,22.2,19.1,22.0,21.2
"1분위","성별",남성,55.5,55.3,57.0,56.7,51.0,48.0,49.5,47.3,46.1,45.6,44.0,41.7,44.6,46.7,44.0
"1분위","성별",여성,44.5,44.7,43.0,43.3,49.0,52.0,50.5,52.7,53.9,54.4,56.0,58.3,55.4,53.3,56.0
"1분위","연령별",20대 이하,3.4,2.9,2.7,2.2,1.4,1.3,1.4,1.3,2.1,1.4,2.6,3.0,4.1,3.6,5.2
"1분위","연령별",30대,9.1,8.8,10.5,8.3,7.5,5.7,5.0,5.9,5.3,4.5,4.4,6.3,4.4,5.2,4.0
"1분위","연령별",40대,12.1,14.8,15.4,15.1,15.4,13.8,12.9,12.1,11.9,14.2,10.6,10.1,9.2,8.4,9.3
"1분위","연령별",50대,16.3,14.6,14.2,13.4,12.0,9.5,9.5,10.0,9.3,10.3,8.9,7.8,10.2,8.9,8.5
```

(정제)소득5분위별_가구주_특성_20240520010... 2024-06-01

```
"소득분위별(1)","특성별(1)",특성별(2),2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021
"1분위","성별",남성,55.5,55.3,57.0,56.7,51.0,48.0,49.5,47.3,46.1,45.6,44.0,41.7,44.6,46.7,44.0
"1분위","성별",여성,44.5,44.7,43.0,43.3,49.0,52.0,50.5,52.7,53.9,54.4,56.0,58.3,55.4,53.3,56.0
"1분위","연령별",20대 이하,3.4,2.9,2.7,2.2,1.4,1.3,1.4,1.3,2.1,1.4,2.6,3.0,4.1,3.6,5.2
"1분위","연령별",30대,9.1,8.8,10.5,8.3,7.5,5.7,5.0,5.9,5.3,4.5,4.4,6.3,4.4,5.2,4.0
"1분위","연령별",40대,12.1,14.8,15.4,15.1,15.4,13.8,12.9,12.1,11.9,14.2,10.6,10.1,9.2,8.4,9.3
"1분위","연령별",50대,16.3,14.6,14.2,13.4,12.0,9.5,9.5,10.0,9.3,10.3,8.9,7.8,10.2,8.9,8.5
"1분위","연령별",60대 이상,59.2,58.9,57.2,61.0,63.7,69.7,71.2,70.7,71.3,69.5,73.5,72.8,72.2,73.9,73.1
"1분위","학력별",중졸이하,66.9,61.3,57.5,59.4,59.7,60.9,61.9,61.9,59.9,58.1,59.2,59.6,57.6,56.6,54.3
"1분위","학력별",고교재학/고졸,22.2,26.6,27.6,25.9,28.3,26.7,24.8,24.0,25.8,25.8,26.3,21.5,25.7,26.7,27.2
"1분위","학력별",대재이상,10.9,12.1,14.9,14.6,11.9,12.3,13.2,14.1,14.3,16.0,14.5,18.9,16.7,16.7,18.6
"1분위","종사자지위별",임금 근로자,8.9,9.5,12.1,13.7,10.7,9.9,10.1,9.6,12.4,12.6,11.1,9.9,16.9,15.1,16.3
```

데이터 탐색 - 확인 (소득분위별 순자산,부채,소득,소비)

데이터 확인을 통해 데이터의 정보와 결측치를 파악합니다.

```
file_path1 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2007~2021/소득5분위별_자산_및_부채_순자산__20240520010244.csv")
file_path2 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2007~2021/소득5분위별_연간_소득원천별_가구소득_20240520010229.csv")
file_path3 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2007~2021/소득5분위별_연간_소비지출_20240520010258.csv")
```

```
import pandas as pd

# 데이터프레임 생성
df1 = pd.read_csv(file_path1,encoding='cp949')
df2 = pd.read_csv(file_path2,encoding='cp949')
df3 = pd.read_csv(file_path3,encoding='cp949')

# 구조 파악
print(df1.shape)
print(df2.shape)
print(df3.shape)

# 데이터프레임 정보 출력
print("₩ndf1 정보:")
print(df1.info())

print("₩ndf2 정보:")
print(df2.info())

print("₩ndf3 정보:")
print(df3.info())


# 결측치 확인
print("df1 결측치 확인:")
print(df1.isnull().sum())

print("₩ndf2 결측치 확인:")
print(df2.isnull().sum())

print("₩ndf3 결측치 확인:")
print(df3.isnull().sum())
```

데이터 정보

```
df1 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   소득분위별(1)   30 non-null     object
 1   자산및부채별(1)  30 non-null     object
 2   2007        30 non-null     float64
 3   2008        30 non-null     float64
 4   2009        30 non-null     float64
 5   2010        30 non-null     float64
 6   2011        30 non-null     float64
 7   2012        30 non-null     float64
 8   2013        30 non-null     float64
 9   2014        30 non-null     float64
 10  2015        30 non-null     float64
 11  2016        30 non-null     float64
 12  2017        30 non-null     float64
 13  2018        30 non-null     float64
 14  2019        30 non-null     float64
 15  2020        30 non-null     float64
 16  2021        30 non-null     float64
dtypes: float64(15), object(2)
memory usage: 4.1+ KB
None
```

```
df2 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   소득분위별(1)   42 non-null     object
 1   소득원천별(1)   42 non-null     object
 2   2007        42 non-null     float64
 3   2008        42 non-null     float64
 4   2009        42 non-null     float64
 5   2010        42 non-null     float64
 6   2011        42 non-null     float64
 7   2012        42 non-null     float64
 8   2013        42 non-null     float64
 9   2014        42 non-null     float64
 10  2015        42 non-null     float64
 11  2016        42 non-null     float64
 12  2017        42 non-null     float64
 13  2018        42 non-null     float64
 14  2019        42 non-null     float64
 15  2020        42 non-null     float64
 16  2021        42 non-null     float64
dtypes: float64(15), object(2)
memory usage: 5.7+ KB
None
```

```
df3 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   소득분위별(1)   6 non-null      object
 1   2007        6 non-null      float64
 2   2008        6 non-null      float64
 3   2009        6 non-null      float64
 4   2010        6 non-null      float64
 5   2011        6 non-null      float64
 6   2012        6 non-null      float64
 7   2013        6 non-null      float64
 8   2014        6 non-null      float64
 9   2015        6 non-null      float64
 10  2016        6 non-null      float64
 11  2017        6 non-null      float64
 12  2018        6 non-null      float64
 13  2019        6 non-null      float64
 14  2020        6 non-null      float64
 15  2021        6 non-null      float64
dtypes: float64(15), object(1)
memory usage: 896.0+ bytes
None
```

데이터 결측치 (없음)

```
df1 결측치 확인:
소득분위별(1)      0
자산및부채별(1)     0
2007          0
2008          0
2009          0
2010          0
2011          0
2012          0
2013          0
2014          0
2015          0
2016          0
2017          0
2018          0
2019          0
2020          0
2021          0
dtype: int64
```

```
df2 결측치 확인:
소득분위별(1)      0
소득원천별(1)      0
2007          0
2008          0
2009          0
2010          0
2011          0
2012          0
2013          0
2014          0
2015          0
2016          0
2017          0
2018          0
2019          0
2020          0
2021          0
dtype: int64
```

```
df3 결측치 확인:
소득분위별(1)      0
2007          0
2008          0
2009          0
2010          0
2011          0
2012          0
2013          0
2014          0
2015          0
2016          0
2017          0
2018          0
2019          0
2020          0
2021          0
dtype: int64
```

데이터 탐색 - 확인 (소득분위별 순자산,부채,소득,소비)

데이터 확인을 통해 데이터의 정보와 결측치를 파악합니다.

```
[3]  file_path1 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2007~2021/(정제)소득5분위별_가구주_특성_20240520010215.csv")
```

```
import pandas as pd

# 데이터프레임 생성
df1 = pd.read_csv(file_path1,encoding='cp949')
print(df1.to_dict())
# 구조 파악
print(df1.shape)

# 데이터프레임 정보 출력
print("₩ndf1 정보:")
print(df1.info())


# 결측치 확인
print("df1 결측치 확인:")
print(df1.isnull().sum())
```

데이터 정보

```
df1 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 80 entries, 0 to 79
Data columns (total 18 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   소득분위별(1)  80 non-null     object
 1   특성별(1)    80 non-null     object
 2   특성별(2)    80 non-null     object
 3   2007       80 non-null     float64
 4   2008       80 non-null     float64
 5   2009       80 non-null     float64
 6   2010       80 non-null     float64
 7   2011       80 non-null     float64
 8   2012       80 non-null     float64
 9   2013       80 non-null     float64
 10  2014       80 non-null     float64
 11  2015       80 non-null     float64
 12  2016       80 non-null     float64
 13  2017       80 non-null     float64
 14  2018       80 non-null     float64
 15  2019       80 non-null     float64
 16  2020       80 non-null     float64
 17  2021       80 non-null     float64
dtypes: float64(15), object(3)
memory usage: 11.4+ KB
None
```

데이터 결측치 (없음)

```
df1 결측치 확인:
소득분위별(1)    0
특성별(1)      0
특성별(2)      0
2007         0
2008         0
2009         0
2010         0
2011         0
2012         0
2013         0
2014         0
2015         0
2016         0
2017         0
2018         0
2019         0
2020         0
2021         0
```

데이터 탐색 - 확인 (자산,소비에 따른 소득분위 예측)

데이터 확인을 통해 데이터의 정보와 결측치를 파악합니다.

```
[3]  #소비
     file_path1 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2007~2021/(정제)소득5분위별_연간_소비지출_20240520010258.csv")

     #자산,소득 정제된 자료를 사용 - 이 데이터를 사용한 이유 자산과 소득이 같은 기준으로 정리된 데이블이 필요하기 때문 또한 표본이 인구주택총조사에서 따온것이며 이 자료는 모집단을 선정한 자료이기 때문에 연관성이 높다.
     file_path2 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2012~2023(자산,부채,소득)/(정제-자산)소득5분위별_가계재무건전성.csv")
     file_path3 = ("/content/drive/MyDrive/Bigdata_TermProject_data/2012~2023(자산,부채,소득)/(정제-소득)소득5분위별_가계재무건전성.csv")
```

```
import pandas as pd

# 데이터프레임 생성
df1 = pd.read_csv(file_path1,encoding='cp949')
df2 = pd.read_csv(file_path2, encoding='cp949')
df3 = pd.read_csv(file_path3, encoding='cp949')

# 구조 파악
print(df1.shape)
print(df2.shape)
print(df3.shape)


# 데이터프레임 정보 출력
print("₩ndf1 정보:")
print(df1.info())

print("₩ndf2 정보:")
print(df2.info())

print("₩ndf3 정보:")
print(df3.info())
```

데이터 정보

```
df1 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 16 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   소득분위별(1)  5 non-null      object
 1   2007       5 non-null      float64
 2   2008       5 non-null      float64
 3   2009       5 non-null      float64
 4   2010       5 non-null      float64
 5   2011       5 non-null      float64
 6   2012       5 non-null      float64
 7   2013       5 non-null      float64
 8   2014       5 non-null      float64
 9   2015       5 non-null      float64
 10  2016       5 non-null      float64
 11  2017       5 non-null      float64
 12  2018       5 non-null      float64
 13  2019       5 non-null      float64
 14  2020       5 non-null      float64
 15  2021       5 non-null      float64
dtypes: float64(15), object(1)
memory usage: 768.0+ bytes
None
```

```
df2 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   소득5분위별  5 non-null      object
 1   2012       5 non-null      int64
 2   2013       5 non-null      int64
 3   2014       5 non-null      int64
 4   2015       5 non-null      int64
 5   2016       5 non-null      int64
 6   2017       5 non-null      int64
 7   2018       5 non-null      int64
 8   2019       5 non-null      int64
 9   2020       5 non-null      int64
 10  2021       5 non-null      int64
 11  2022       5 non-null      int64
 12  2023       5 non-null      int64
dtypes: int64(12), object(1)
memory usage: 648.0+ bytes
None
```

```
df3 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   소득5분위별  5 non-null      object
 1   2012       5 non-null      int64
 2   2013       5 non-null      int64
 3   2014       5 non-null      int64
 4   2015       5 non-null      int64
 5   2016       5 non-null      int64
 6   2017       5 non-null      int64
 7   2018       5 non-null      int64
 8   2019       5 non-null      int64
 9   2020       5 non-null      int64
 10  2021       5 non-null      int64
 11  2022       5 non-null      int64
 12  2023       5 non-null      int64
dtypes: int64(12), object(1)
memory usage: 648.0+ bytes
None
```

데이터 결측치 (없음)

```
df1 결측치 확인:
소득분위별(1)    0
2007        0
2008        0
2009        0
2010        0
2011        0
2012        0
2013        0
2014        0
2015        0
2016        0
2017        0
2018        0
2019        0
2020        0
2021        0
dtype: int64
```

```
df2 결측치 확인:
소득5분위별    0
2012        0
2013        0
2014        0
2015        0
2016        0
2017        0
2018        0
2019        0
2020        0
2021        0
2022        0
2023        0
dtype: int64
```

```
df3 결측치 확인:
소득5분위별    0
2012        0
2013        0
2014        0
2015        0
2016        0
2017        0
2018        0
2019        0
2020        0
2021        0
2022        0
2023        0
```

데이터 분석 - 데이터 내용 확인

1

```
year = ['2007','2008','2009','2010','2011','2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021']

#자산 및 부채별
columns_of_interest1 = ['소득분위별(1)', '자산및부채별(1)']+year

df_f1 = df1[columns_of_interest1]

income_levels = ['1분위', '2분위', '3분위', '4분위', '5분위']
data_types1 = ['전체(순자산)','부채']

#소득
columns_of_interest2 = ['소득분위별(1)', '소득원천별(1)']+year

df_f2 = df2[columns_of_interest2]
#전체  : 근로소득,사업소득,부동산임대소득,이자및 배당소득,사적이전소득,공적이전소득
#사적이전소득이란 기초생활 수급자가 아는 사람에게 받은 돈 : 쉽게 누군가에게 받은 돈
#공적이전소득은 국가에서 정기적 또는 일시적으로 지원하는 각종 수당, 연금, 급여 등
data_types2 = ['전체','근로소득']


#소비
columns_of_interest3 = ['소득분위별(1)']+year

df_f3 = df3[columns_of_interest3]
```

데이터 프레임을 만들고 그 안에 순자산, 부채, 전체소득, 근로소득, 소비 데이터를 파악하여 소득 분위를 기준으로 NumPy 배열을 만드는 과정의 일부 입니다.

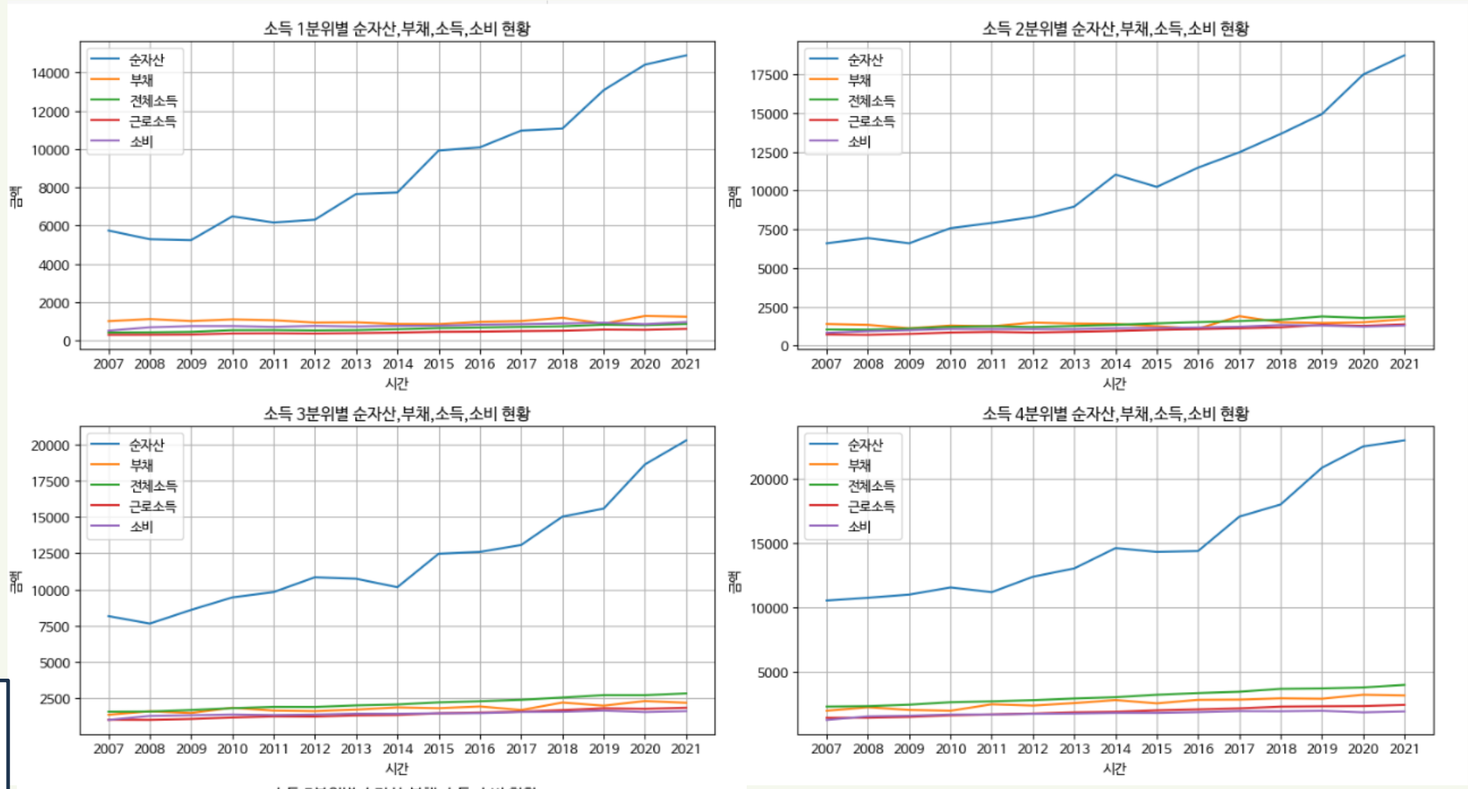
```
#자산 및 부채
data_dict1 = {}

for income_level in income_levels:
    for data_type in data_types1:
        key = f"{income_level}_{data_type}"
        data = df_f1[
            (df_f1['소득분위별(1)'] == income_level) &
            (df_f1['자산및부채별(1)'] == data_type)
        ]
        if not data.empty:
            data_dict1[key] = data.iloc[:, 2:].values.flatten()

# 결과 출력
for key, value in data_dict1.items():
    print(f"{key}: {value}")
print()

#소비
data_dict2 = {}

for income_level in income_levels:
    for data_type in data_types2:
        key = f"{income_level}_{data_type}"
        data = df_f2[
            (df_f2['소득분위별(1)'] == income_level) &
            (df_f2['소득원천별(1)'] == data_type)
        ]
        if not data.empty:
            data_dict2[key] = data.iloc[:, 2:].values.flatten()

# 결과 출력
for key, value in data_dict2.items():
    print(f"{key}: {value}")
print()

#소비
data_dict3 = {}

for income_level in income_levels:
    key = f"{income_level}"
    data = df_f3[
        (df_f3['소득분위별(1)'] == income_level)
    ]
    if not data.empty:
        data_dict3[key] = data.iloc[:, 1:].values.flatten()

# 결과 출력
for key, value in data_dict3.items():
    print(f"{key}: {value}")
```

```
1분위_전체(순자산): [ 5747.7  5294.9  5245.5  6488.7  6169.4  6314.5  7654.8  7736.1  9925.3
 10092.5 10966.8 11073.2 13078.6 14410.8 14897.2]
1분위_부채: [1017.1 1125.7 1021.8 1108.2 1064.8  940.3  953.3  868.   860.7  977.3
 1020.8 1193.7  885.4 1286.4 1248.5]
2분위_전체(순자산): [ 6599.3  6932.6  6599.3  7573.9  7912.8  8293.8  8967.5 11029.7 10236.4
 11480.3 12470.1 13653.4 14931.7 17484.3 18726.2]
2분위_부채: [1387.6 1334.  1115.2 1291.1 1243.1 1479.7 1417.6 1394.9 1246.3 1089.
 1894.7 1498.6 1453.1 1495.1 1700.3]
3분위_전체(순자산): [ 8170.3  7653.2  8596.2  9458.9  9826.7 10847.5 10747.6 10164.5 12470.4
 12597.7 13074.4 15030.  15583.1 18632.5 20284.7]
3분위_부채: [1369.8 1601.9 1488.6 1850.3 1655.6 1617.4 1724.8 1874.7 1821.7 1941.1
 1693.7 2208.9 1994.6 2315.3 2193.7]
4분위_전체(순자산): [10521.1 10737.3 10984.1 11540.5 11174.8 12376.3 13024.7 14605.6 14308.5
 14376.2 17056.6 18001.8 20859.2 22519.2 22989.6]
4분위_부채: [1939.9 2206.3 2003.6 1937.3 2443.  2342.2 2535.8 2773.2 2515.4 2791.8
 2811.7 2906.6 2875.3 3187.6 3133.9]
5분위_전체(순자산): [20275.  19735.9 19580.9 23210.8 22546.  22137.1 22919.1 23206.2 27076.2
 28804.4 30539.3 31307.3 34303.2 43489.2 43927.8]
5분위_부채: [3866.1 3655.3 3674.5 4612.9 4564.8 4638.9 5362.1 4647.5 5201.  5167.6
 4833.6 5280.4 5278.4 6304.4 6995.1]

1분위_전체: [424.2 433.  455.1 543.1 548.8 531.9 547.6 592.9 652.9 681.4 717.  747.7
 832.5 811.1 868.5]
1분위_근로소득: [139.7 109.2 135.7 155.2 129.5 117.  114.5 126.2 133.5 156.2 140.2 149.1
 205.1 147.9 179.7]
2분위_전체: [1031.1 1022.9 1096.8 1206.5 1247.  1198.1 1263.5 1337.8 1430.3 1508.9
 1577.1 1666.7 1876.2 1780.4 1877.6]
2분위_근로소득: [ 580.6  558.4  606.2  671.3  699.8  585.2  636.5  652.6  758.5  793.2
  802.4  876.9 1060.2  825.2  941.7]
3분위_전체: [1576.7 1589.2 1693.9 1822.9 1910.7 1905.4 2013.4 2078.3 2222.2 2300.1
 2399.1 2555.9 2720.8 2719.6 2841.3]
3분위_근로소득: [1014.6  987.1 1069.3 1169.1 1246.5 1182.3 1187.8 1296.2 1378.3 1444.5
 1546.2 1702.1 1847.6 1728.5 1826. ]
4분위_전체: [2262.7 2296.4 2411.5 2603.4 2673.6 2763.9 2893.7 2994.2 3185.5 3317.4
 3420.4 3642.7 3677.8 3755.4 3952.5]
4분위_근로소득: [1617.5 1595.6 1656.7 1841.1 1892.9 1866.4 1990.6 2027.5 2203.1 2336.9
 2378.  2579.2 2512.4 2696.1 2805.7]
5분위_전체: [4173.8 4179.1 4315.5 4750.1 4889.3 5150.6 5228.7 5418.6 5795.8 6214.6
 6102.4 6549.  6311.6 6485.3 6959.8]
5분위_근로소득: [2743.6 2981.3 2963.3 3123.9 3468.2 3468.3 3809.1 4056.  4269.5 4493.4
 4503.6 4878.  4417.4 4484.3 4851.4]

1분위: [518.  694.9 758.5 760.6 719.3 768.8 739.3 772.3 783.9 830.8 857.5 894.1
 933.2 859.6 964.3]
2분위: [ 824.6  925.9  991.4 1064.7 1055.3 1056.1 1053.5 1116.5 1134.2 1161.6
 1214.4 1322.7 1294.7 1218.6 1295.2]
3분위: [1010.2 1287.5 1333.1 1385.3 1347.6 1380.2 1436.8 1432.3 1465.7 1479.8
 1559.2 1571.2 1650.9 1552.2 1616.4]
4분위: [1223.8 1501.9 1553.7 1646.3 1652.1 1700.7 1722.9 1771.9 1771.8 1828.7
 1907.6 1896.4 1933.  1814.2 1884.3]
5분위: [1604.4 1962.  1955.  2173.  2035.1 2180.1 2288.6 2242.9 2342.9 2377.4
 2405.6 2485.3 2478.5 2348.6 2384.9]
```

데이터 분석 - 데이터 분포 확인

1

```
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 12))

for i in range(5):
    row = i // 2
    col = i % 2

    ax = axes[row, col]

    ax.plot(years, net_worth_data['순자산'][i], label='순자산')
    ax.plot(years, debt_data['부채'][i], label='부채')
    ax.plot(years, total_income_data['전체소득'][i], label='전체소득')
    ax.plot(years, income_data['근로소득'][i], label='근로소득')
    ax.plot(years, consume_data['소비'][i], label='소비')

    ax.set_title(f'소득 {i+1}분위별 순자산,부채,소득,소비 현황')
    ax.set_xlabel('시간')
    ax.set_ylabel('금액')
    ax.legend()
    ax.grid(True)

plt.tight_layout()
plt.show()
```



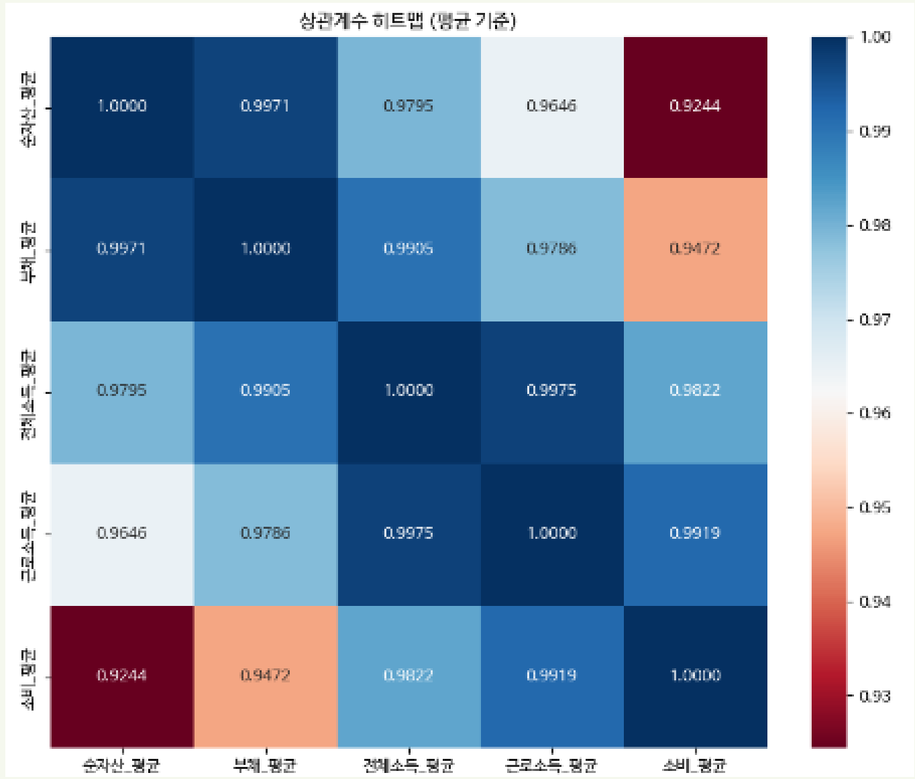
서술 분석 하는 그래프들을 먼저 만듭니다.
그래프는 소득 5분위를 기준으로 순자산, 부채,전체소득, 근로소득,소비를 나타내는 그래프를 그립니다.

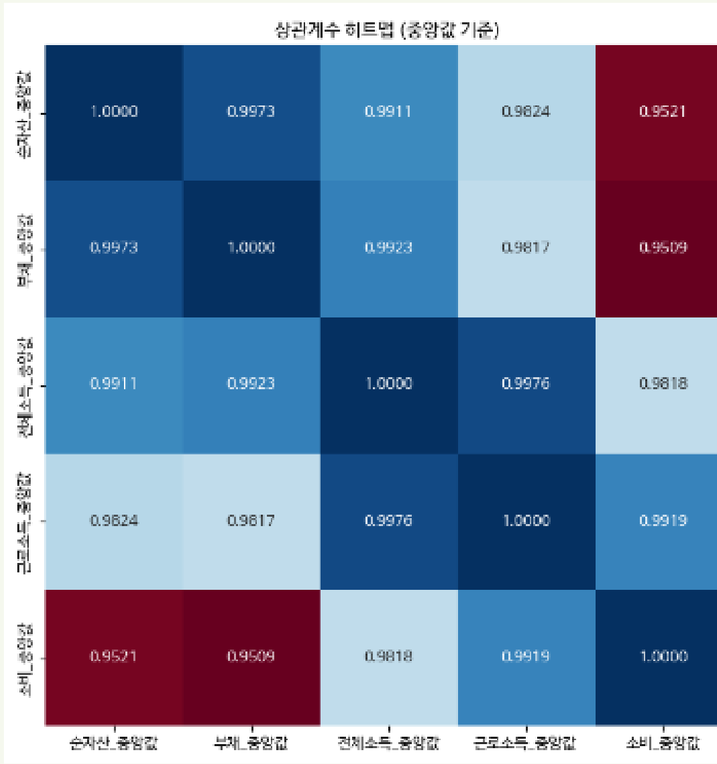
데이터 분석 - 히트맵 (상관계수 분석)

1

```
[21] net_worth_avg = [arr.mean() for arr in net_worth_data['순자산']]
     debt_avg = [arr.mean() for arr in debt_data['부채']]
     total_income_avg = [arr.mean() for arr in total_income_data['전체소득']]
     income_avg = [arr.mean() for arr in income_data['근로소득']]
     consume_avg = [arr.mean() for arr in consume_data['소비']]

     # Pandas DataFrame 생성
     data1 = {
         '순자산_평균': net_worth_avg,
         '부채_평균': debt_avg,
         '전체소득_평균': total_income_avg,
         '근로소득_평균': income_avg,
         '소비_평균': consume_avg
     }
     df1 = pd.DataFrame(data1)

     # 상관계수 계산
     correlation_matrix1 = df1.corr()

     print(correlation_matrix1)


     # 평균 대신 중앙값을 계산
     net_worth_median = [np.median(arr) for arr in net_worth_data['순자산']]
     debt_median = [np.median(arr) for arr in debt_data['부채']]
     total_income_median = [np.median(arr) for arr in total_income_data['전체소득']]
     income_median = [np.median(arr) for arr in income_data['근로소득']]
     consume_median = [np.median(arr) for arr in consume_data['소비']]

     # Pandas DataFrame 생성
     data2 = {
         '순자산_중앙값': net_worth_median,
         '부채_중앙값': debt_median,
         '전체소득_중앙값': total_income_median,
         '근로소득_중앙값': income_median,
         '소비_중앙값': consume_median
     }
     df2 = pd.DataFrame(data2)  # years를 인덱스로 설정

     # 상관계수 계산
     correlation_matrix2 = df2.corr()

     print(correlation_matrix2)
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# 상관계수 행렬 시각화
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix1, annot=True, fmt='.4f', cmap=plt.cm.RdBu)
plt.title('상관계수 히트맵 (평균 기준)')
plt.show()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix2, annot=True, fmt='.4f', cmap=plt.cm.RdBu)
plt.title('상관계수 히트맵 (중앙값 기준)')
plt.show()
```





결론

순자산,자산과 소비의 상관관계가
상대적으로 약한 상관관계를 가지고 있습니다.
다중공선성을 생각해서 자산&소비를 선택합니다

데이터 분석 - 비중 분석 2

1

```
years =['2007','2008','2009','2010','2011','2012','2013','2014','2015','2016','2017','2018','2019','2020','2021']

columns_of_interest = ['소득분위별(1)', '특성별(1)','특성별(2)']+years
income_levels = ['1분위', '2분위', '3분위', '4분위', '5분위']
categories = ['성별', '연령별', '학력별', '종사자지위별']
genders = ['남성','여성']
ages = ['20대 이하', '30대', '40대', '50대', '60대 이상']
educations = ['중졸이하','고교재학/고졸', '대재이상']
employee_statuses = ['임금 근로자','일용 근로자','고용원이 없는 자영업자','고용원을 둔 사업주','무급가족 종사자','전업주부/학생/무직']
# 사용자 입력 받기
years_input = input("연도별 (2007~2021): ")
income_level_input = input("소득분위 (1-5): ") + "분위"
gender_input = input("성별 (남성/여성): ")

age_input = input("연령별 (20대 이하/30대/40대/50대/60대 이상): ")
education_input = input("학력별 (중졸이하/고교재학/고졸/대재이상): ")
employee_status_input = input("종사자지위별 (임금 근로자/일용 근로자/고용원이 없는 자영업자/고용원을 둔 사업주/무급가족 종사자/전업주부,학생,무직): ")
years_index = years.index('2007')+3
```

비중 분석은 전체 가구 중에서 연도, 소득 분위, 성별, 연령별, 학력별, 종사자 지위별에 따른 비율을 알려주고 있습니다. 대상은 2005년 ~ 2018년 15,707,929 가구 / 2018년 이후 20,364,055 가구를 기준으로 비율을 파악합니다.

2

```
# 데이터 추출 및 배열화
data_dict = {}

for income_level in income_levels:
  if income_level == income_level_input:
    for category in categories:
        for gender in genders:
            if gender == gender_input:
                key = f"{category}_{income_level}_{gender}"
                data = df1_filtered[
                    (df1_filtered['소득분위별(1)'] == income_level) &
                    (df1_filtered['특성별(1)'] == category) &
                    (df1_filtered['특성별(2)'] == gender)
                ]

                # 선택된 연도에 대한 데이터만 추출
                if not data.empty:
                    data_dict[key] = data[years_input].values.flatten()


        for age in ages:
            if age == age_input:
                key = f"{category}_{income_level}_{age}"
                data = df1_filtered[
                    (df1_filtered['소득분위별(1)'] == income_level) &
                    (df1_filtered['특성별(1)'] == category) &
                    (df1_filtered['특성별(2)'] == age)
                ]
                if not data.empty:
                    data_dict[key] = data[years_input].values.flatten()

        for education in educations:
            if education == education_input:
                key = f"{category}_{income_level}_{education}"
                data = df1_filtered[
                    (df1_filtered['소득분위별(1)'] == income_level) &
                    (df1_filtered['특성별(1)'] == category) &
                    (df1_filtered['특성별(2)'] == education)
                ]
                if not data.empty:
                    data_dict[key] = data[years_input].values.flatten()

        for employee_status in employee_statuses:
            if employee_status == employee_status_input:
                key = f"{category}_{income_level}_{employee_status}"
                data = df1_filtered[
                    (df1_filtered['소득분위별(1)'] == income_level) &
                    (df1_filtered['특성별(1)'] == category) &
                    (df1_filtered['특성별(2)'] == employee_status)
                ]
                if not data.empty:
                    data_dict[key] = data[years_input].values.flatten()
```

데이터 분석 - 비중 분석 2

3

```
연도별 (2007~2021): 2021
소득분위 (1-5): 1
성별 (남성/여성): 남성
연령별 (20대 이하/30대/40대/50대/60대 이상): 20대 이하
학력별 (중졸이하/고교재학/고졸/대재이상): 중졸이하
종사자지위별 (임금 근로자/일용 근로자/고용원이 없는 자영업자/고용원을 둔 사업주/무급가족 종사자/전업주부,학생,무직): 임금 근로자


성별_1분위_남성: [44.]
연령별_1분위_20대 이하: [5.2]
학력별_1분위_중졸이하: [54.3]
종사자지위별_1분위_임금 근로자: [16.3]
2021년도에 가구원: 가구 내 작년 한해 동안 소득이 있거나 소득 활동을 한 가구원 중
소득1분위
성별 : 남성
연령대 :  20대 이하
교육수준 : 중졸이하
종사자지위 : 임금 근로자

해당하는 비율
비율 :0.00202509%
해당 연도 조사 기준 총 가구 수 :  20364055
해당 특성의 가구 수: 412
```

비중 분석을 통하여 소득분위,연도에 해당하는 기준에 따른 비율을 확인하고 해당 특성의 가구 수를 파악하는 분석 결과입니다.

데이터 분석 - 예측 그래프 (1)

3 - 자산,소비를 통해 소득을 예측하고 소득 분위를 확인하는 예측 모델

```
# 독립 변수와 종속 변수 선택
X = merged_df[['자산', '소비']]
y = merged_df['소득']
```

```
14] # 결과를 저장할 DataFrame 준비
    results_df = pd.DataFrame(columns=['Test Size', 'R2 Score', 'MSE', 'P-Value'])

    for test_size in np.arange(0.1, 0.51, 0.01):
        # 데이터 분할
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=0)

        # 선형 회귀 모델 생성 및 훈련
        model = LinearRegression()
        model.fit(X_train, y_train)

        # 예측 및 평가
        y_pred = model.predict(X_test)
        r2 = r2_score(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)

        # F-검정 유의확률 계산
        X_train_sm = X_train  # 상수항이 이미 포함되어 있으므로 sm.add_constant() 불필요
        model_sm = sm.OLS(y_train, X_train_sm).fit()
        f_pvalue = model_sm.f_pvalue

        # 결과 저장
        results_df.loc[len(results_df)] = [test_size, r2, mse, f_pvalue]

    # 결과 출력
    print(results_df)

    # r2가 가장 높은 행 출력
    best_r2_row = results_df.loc[results_df['R2 Score'].idxmax()]
    print("₩nBest R2 Score:")
    print(best_r2_row)

    # mse가 가장 낮은 행 출력
    best_mse_row = results_df.loc[results_df['MSE'].idxmin()]
    print("₩nLowest MSE:")
    print(best_mse_row)

    # f_pvalue가 가장 낮은 행 출력
    best_pvalue_row = results_df.loc[results_df['P-Value'].idxmin()]
    print("₩nLowest P-Value:")
    print(best_pvalue_row)
```

	Test Size	R2 Score	MSE	P-Value
0	0.10	0.990393	24548.013309	2.555360e-48
1	0.11	0.990742	20919.982368	3.881761e-47
2	0.12	0.990742	20919.982368	3.881761e-47
3	0.13	0.994022	17946.452482	6.175227e-46
4	0.14	0.994022	17946.452482	6.175227e-46
5	0.15	0.988876	30410.181875	6.329647e-45
6	0.16	0.988876	30410.181875	6.329647e-45
7	0.17	0.988887	27051.031620	8.893144e-44
8	0.18	0.988887	27051.031620	8.893144e-44
9	0.19	0.987550	53641.146997	1.389883e-42
10	0.20	0.987550	53641.146997	1.389883e-42
11	0.21	0.988307	49902.629164	1.531859e-41
12	0.22	0.988307	49902.629164	1.531859e-41
13	0.23	0.989788	46408.662309	1.109870e-40
14	0.24	0.989788	46408.662309	1.109870e-40
15	0.25	0.989866	43950.791587	1.301490e-39
16	0.26	0.989866	43950.791587	1.301490e-39
17	0.27	0.993338	49761.824596	5.672192e-38
18	0.28	0.993338	49761.824596	5.672192e-38
19	0.29	0.993091	48425.027228	4.552639e-37
20	0.30	0.993091	48425.027228	4.552639e-37
21	0.31	0.993073	45989.687945	5.300404e-36
22	0.32	0.993073	45989.687945	5.300404e-36
23	0.33	0.994095	43954.810283	1.880421e-34
24	0.34	0.994095	43954.810283	1.880421e-34
25	0.35	0.994005	45146.026909	8.745665e-34
26	0.36	0.994005	45146.026909	8.745665e-34
27	0.37	0.994010	44023.492144	1.932055e-32
28	0.38	0.994010	44023.492144	1.932055e-32
29	0.39	0.989064	95955.276498	4.063145e-32
30	0.40	0.989064	95955.276498	4.063145e-32
31	0.41	0.989876	91020.310547	2.195303e-31
32	0.42	0.989876	91020.310547	2.195303e-31
33	0.43	0.990178	89243.963188	1.026134e-30
34	0.44	0.990178	89243.963188	1.026134e-30
35	0.45	0.990085	86242.499540	1.751896e-29
36	0.46	0.990085	86242.499540	1.751896e-29
37	0.47	0.990315	82528.874960	2.339638e-28
38	0.48	0.990315	82528.874960	2.339638e-28
39	0.49	0.990296	79385.076014	4.097630e-27
40	0.50	0.990296	79385.076014	4.097630e-27

```
Best R2 Score:
Test Size    3.300000e-01
R2 Score     9.940952e-01
MSE          4.395481e+04
P-Value      1.880421e-34
Name: 23, dtype: float64

Lowest MSE:
Test Size    1.300000e-01
R2 Score     9.940218e-01
MSE          1.794645e+04
P-Value      6.175227e-46
Name: 3, dtype: float64

Lowest P-Value:
Test Size    1.000000e-01
R2 Score     9.903928e-01
MSE          2.454801e+04
P-Value      2.555360e-48
Name: 0, dtype: float64
```

예측 분석을 하기 위하여 선형 회귀 모델을 만듭니다.

자산과 소비를 통하여 소득을 예측하고 소득 5분위_소득 자료에 넣어서 소득 5분위의 위치를 파악하는 것입니다.

관련하여 testsize, p-value, MSE를 반복문으로 통해 가장 좋은 모델을 찾는 과정입니다.

데이터 분석 - 예측 그래프 (1)

3

- 자산,소비를 통해 소득을 예측하고 소득 분위를 확인하는 예측 모델

```python
# 데이터 분할 (학습 데이터와 테스트 데이터)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.13, random_state=0)

# 선형 회귀 모델 생성
model = LinearRegression()

# 모델 훈련
model.fit(X_train, y_train)

# 예측
y_pred = model.predict(X_test)

# R² 점수 계산
r2 = r2_score(y_test, y_pred)
adj_r2 = 1 - (1 - r2) * (len(y_train) - 1) / (len(y_train) - X_train.shape[1] - 1)

# F-검정 유의확률 계산
f_statistic = model_sm.fvalue
f_pvalue = model_sm.f_pvalue

print(f"R-squared:₩t{r2:.3f}")
print(f"Adj. R-squared:₩t{adj_r2:.3f}")
print(f"F-statistic:₩t{f_statistic:.2f}")
print(f"P - Value:₩t{f_pvalue}")

# 모델 평가 (평균 제곱 오차)
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE):", mse)
# 평균 제곱근 오차 (RMSE) 계산
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f"Root Mean Squared Error (RMSE): {rmse:.3f}")

# 회귀 계수 확인
print("Intercept (절편):", model.intercept_)
print("Coefficients (계수 : 자산, 소비):", model.coef_)
```

```
R-squared:      0.994
Adj. R-squared: 0.994
F-statistic:    2254.50
P - Value:      4.097630314653060 6e-27
Mean Squared Error (MSE): 17946.45248196038
Root Mean Squared Error (RMSE): 133.964
Intercept (절편): -1800.9159542720563
Coefficients (계수 : 자산, 소비): [0.08095987 1.89094071]
```

관련하여 MSE가 가장 낮은 케이스인 testsize = 0.13 을 선택하였습니다.

관련하여 절편과 계수입니다.

```
R-squared:      0.994
Adj. R-squared: 0.994
F-statistic:    2254.50
P - Value:      4.097630314653060 6e-27
Mean Squared Error (MSE): 17946.45248196038
Root Mean Squared Error (RMSE): 133.964
Intercept (절편): -1800.9159542720563
Coefficients (계수 : 자산, 소비): [0.08095987 1.89094071]
```

데이터 분석 - 예측 그래프 (2)

3

-연도와 소득을 가지고 2026년까지 소득 분위를 예측하는 모델

```
years = [year for year in income_data.keys() if year.isnumeric()]
income_values = [income_data[year] for year in years]

X = np.array([int(year) for year in years]).reshape(-1, 1)
models = []

for i in range(5):
    y = np.array([income[i] for income in income_values])
    model = LinearRegression()
    model.fit(X, y)
    models.append(model)


for i, model in enumerate(models):
    y_true = np.array([income[i] for income in income_values])
    y_pred = model.predict(X)
    r2 = r2_score(y_true, y_pred)
    print(f"소득{i+1}분위 모델 R2 스코어: {r2:.6f}")
    print(f"소득{i+1}분위 모델 계수:")
    print(f"  기울기: {model.coef_[0]:.2f}")
    print(f"  절편: {model.intercept_:.2f}")
```

```
소득1분위 모델 R2 스코어: 0.933416
소득1분위 모델 계수:
  기울기: 49.52
  절편: -98987.30
소득2분위 모델 R2 스코어: 0.981345
소득2분위 모델 계수:
  기울기: 88.18
  절편: -175692.84
소득3분위 모델 R2 스코어: 0.988346
소득3분위 모델 계수:
  기울기: 147.50
  절편: -293935.14
소득4분위 모델 R2 스코어: 0.970787
소득4분위 모델 계수:
  기울기: 228.73
  절편: -456034.87
소득5분위 모델 R2 스코어: 0.935679
소득5분위 모델 계수:
  기울기: 385.77
  절편: -768245.39
```

```
future_years = list(range(2022, 2027))
future_incomes = []

for i in range(5):
    future_income = []
    for year in future_years:
        income = models[i].predict([[year]])
        future_income.append(income[0])
    future_incomes.append(future_income)


for i in range(5):
    print(f"소득 {i+1}분위 예측 결과:")
    for j, year in enumerate(future_years):
        print(f"{year}년도 예상 소득: {future_incomes[i][j]:.2f} 만 원")
    print()
```

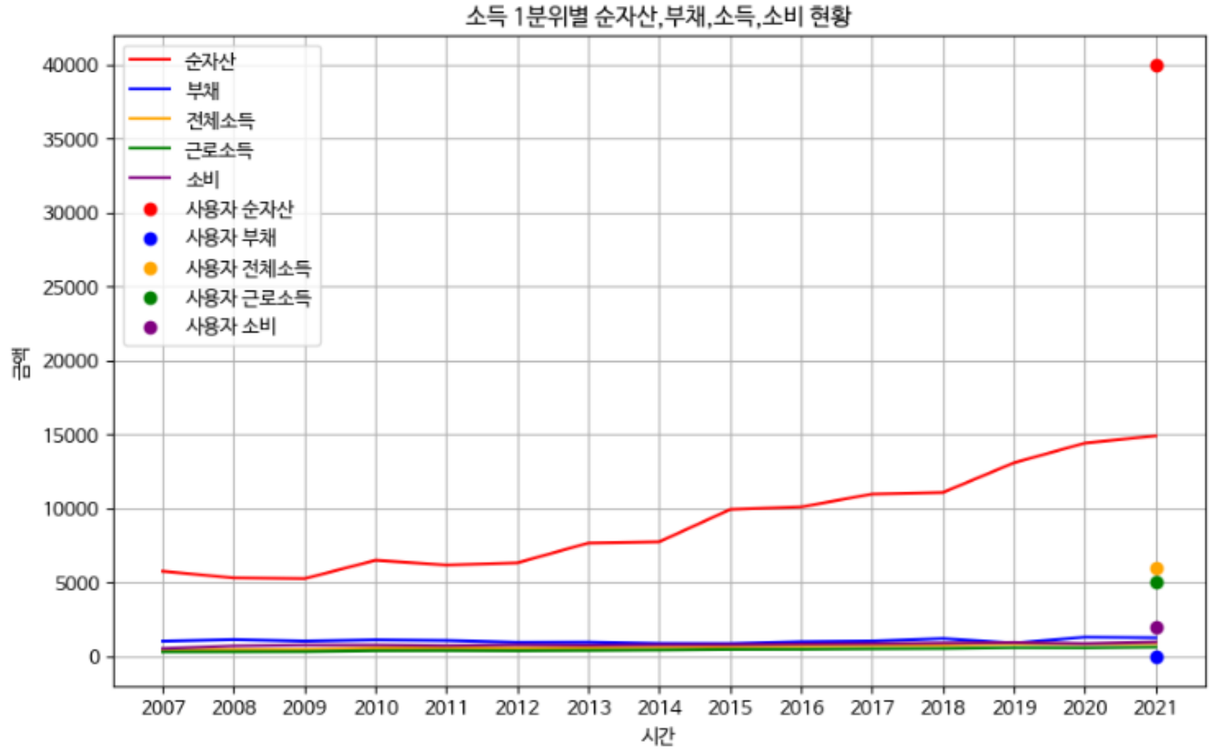
```
소득 1분위 예측 결과:
2022년도 예상 소득: 1132.33 만 원
2023년도 예상 소득: 1181.85 만 원
2024년도 예상 소득: 1231.36 만 원
2025년도 예상 소득: 1280.88 만 원
2026년도 예상 소득: 1330.39 만 원

소득 2분위 예측 결과:
2022년도 예상 소득: 2610.80 만 원
2023년도 예상 소득: 2698.98 만 원
2024년도 예상 소득: 2787.16 만 원
2025년도 예상 소득: 2875.35 만 원
2026년도 예상 소득: 2963.53 만 원

소득 3분위 예측 결과:
2022년도 예상 소득: 4303.73 만 원
2023년도 예상 소득: 4451.23 만 원
2024년도 예상 소득: 4598.73 만 원
2025년도 예상 소득: 4746.22 만 원
2026년도 예상 소득: 4893.72 만 원

소득 4분위 예측 결과:
2022년도 예상 소득: 6463.93 만 원
2023년도 예상 소득: 6692.67 만 원
2024년도 예상 소득: 6921.40 만 원
2025년도 예상 소득: 7150.13 만 원
2026년도 예상 소득: 7378.87 만 원

소득 5분위 예측 결과:
2022년도 예상 소득: 11780.93 만 원
2023년도 예상 소득: 12166.70 만 원
2024년도 예상 소득: 12552.47 만 원
2025년도 예상 소득: 12938.24 만 원
2026년도 예상 소득: 13324.01 만 원
```

연도와 기존의 소득 분위를 이용하여 2022년부터 2026년까지 소득 분위를 예측하는 회귀입니다.
회귀 모델을 사용하여 연도에 따른 소득을 예측하여 출력합니다.

데이터 시각화 - 순자산, 부채, 전체 소득, 근로소득, 소비 (그래프, 서술 분석)

1

```
모든 순자산을 입력하세요 (단위: 만 원): 40000
모든 부채를 입력하세요(단위 : 만 원) : 0
모든 전체소득을 입력하세요(단위 : 만 원) : 6000
한 해 근로소득을 입력하세요(단위 : 만 원) : 5000
한 해 소비를 입력하세요(단위 :만 원) : 2000
```




```
입력한 값과 순자산의 차이 비율 (높을수록 순자산이 소득분위표에 비해서 많다는 의미) :          168.5068%
입력한 값과 부채의 차이 비율 (낮을수록 부채가 소득분위표에 비해서 적다는 의미) :            -99.9199%
입력한 값과 전체 소득의 차이 비율(높을수록 전체 소득이 소득분위표에 비해서 많다는 의미) :    590.8463%
입력한 값과 근로 소득의 차이 비율(높을수록 근로 소득이 소득분위표에 비해서 많다는 의미) :    717.5278%
입력한 값과 소비의 차이 비율(높을수록 소비가 소득분위표에 비해서 많다는 의미) :            107.4043%
------------------------------------------------
```

사용자가 순자산,부채,전체소득,근로소득을 입력하면

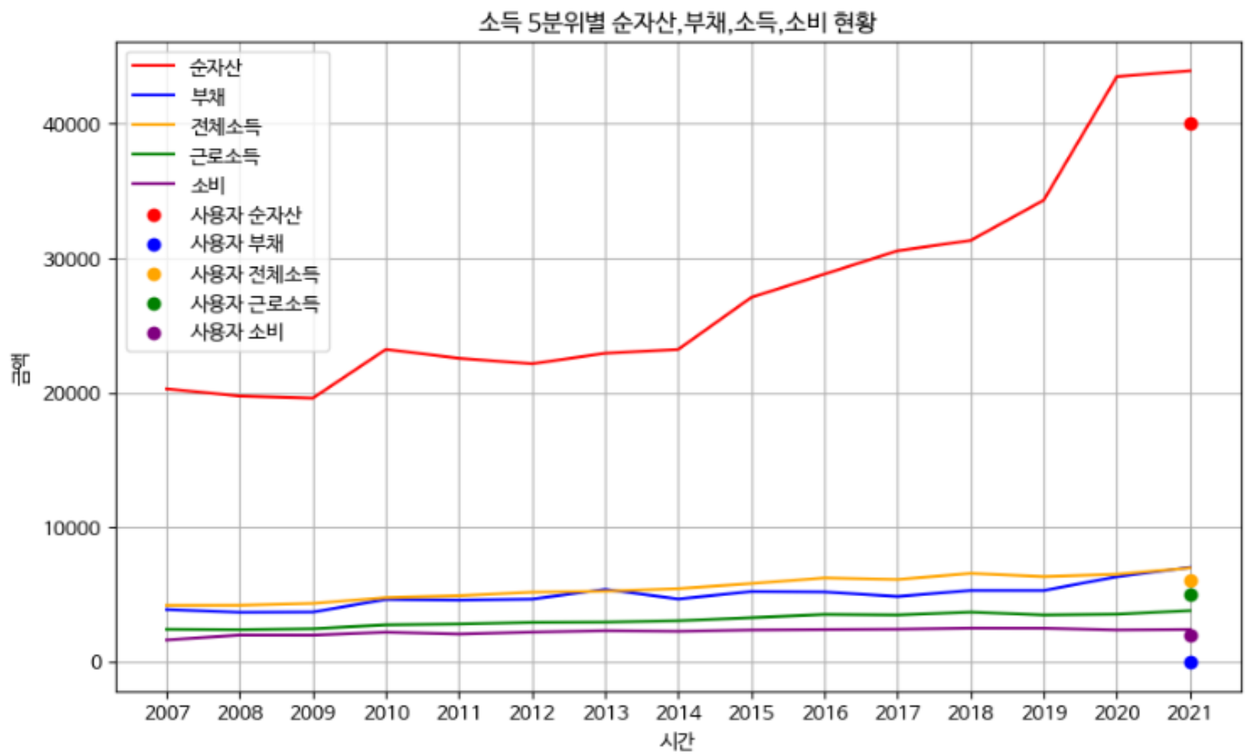
가장 최신인 2021년을 기준으로 위치를 알 수 있습니다.
소득 분위별로 얼마나 차이가 나는지 알 수 있도록 시각화를 진행하였습니다.

다음은 소득 1분위일때 그래프 입니다.

데이터 시각화 - 순자산, 부채, 전체 소득, 근로소득, 소비 (그래프, 서술 분석)

1

```
모든 순자산을 입력하세요 (단위: 만 원): 40000
모든 부채를 입력하세요(단위 : 만 원) : 0
모든 전체소득을 입력하세요(단위 : 만 원) : 6000
한 해 근로소득을 입력하세요(단위 : 만 원) : 5000
한 해 소비를 입력하세요(단위 :만 원) : 2000
------------------------------------------------
```




```
입력한 값과 순자산의 차이 비율 (높을수록 순자산이 소득분위표에 비해서 많다는 의미) :          -8.9415%
입력한 값과 부채의 차이 비율 (낮을수록 부채가 소득분위표에 비해서 적다는 의미) :              -99.9857%
입력한 값과 전체 소득의 차이 비율(높을수록 전체 소득이 소득분위표에 비해서 많다는 의미) :    -13.7906%
입력한 값과 근로 소득의 차이 비율(높을수록 근로 소득이 소득분위표에 비해서 많다는 의미) :    32.2297%
입력한 값과 소비의 차이 비율(높을수록 소비가 소득분위표에 비해서 많다는 의미) :              -16.139%
------------------------------------------------
```

다음은 소득 5분위일때 그래프 입니다.

-%는 내가 부족한 정도를 나타내는 의미이며
+%는 내가 소득 분위에 비해서 얼마나 가지고 있는지 알 수 있습니다.

3 데이터 시각화 - 미래의 예측 소득과 소득 분위의 위치를 파악 (그래프, 예측분석)

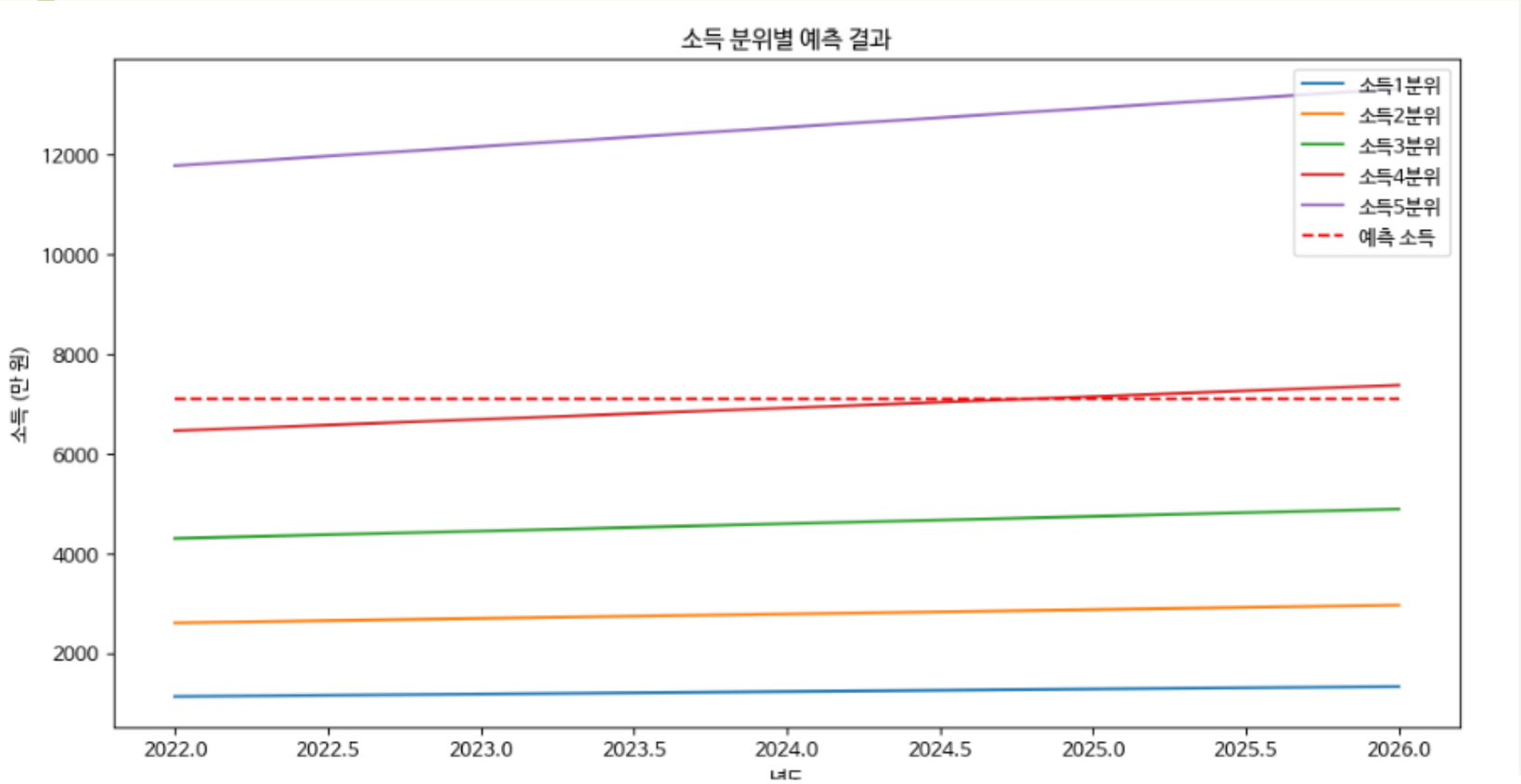
```
future_years = list(range(2022, 2027))
future_incomes = []

for i in range(5):
    future_income = []
    for year in future_years:
        income = models[i].predict([[year]])
        future_income.append(income[0])
    future_incomes.append(future_income)

future_incomes_T = np.array(future_incomes).T.tolist()
print(future_incomes_T)
for i in range(5):
    print(f"소득 {i+1}분위 예측 결과:")
    for j, year in enumerate(future_years):
        print(f"{year}년도 예상 소득: {future_incomes[i][j]:.2f} 만 원")
    print()

predicted_incomes = [predicted_income_data for _ in future_years]


# 차트 생성
plt.figure(figsize=(12, 6))
for i in range(5):
    plt.plot(future_years, future_incomes[i], label=income_data['소득5분위'][i])
plt.plot(future_years, predicted_incomes, label='예측 소득', linestyle='--', color='red')
plt.xlabel('년도')
plt.ylabel('소득 (만 원)')
plt.title('소득 분위별 예측 결과')
plt.legend()
plt.show()



for year, incomes, future_income in zip(future_years, predicted_incomes, future_incomes_T):
    print(f"{year}년도 소득 분위별 차이 (%):")
    min_difference = float('inf')
    min_difference_percentage = 0
    for i in range(5):
        for j in range(5):
            difference = future_income[i] - incomes
            difference_percentage = (difference / incomes) * 100
            if abs(difference_percentage) < abs(min_difference):
              min_difference = difference_percentage
              min_difference_percentage = difference_percentage
        print(f"  소득 {i+1}분위: {float(difference_percentage):.2f}%")
    print(f"  이 년도 중 0에 가장 가까운 차이: {float(min_difference_percentage):.2f}%")
    print()
```

회귀 분석을 시각화 하는 것에 대하여 2가지 회귀를 시각화 하는 과정입니다.

1. 자산과 소비를 통한 소득을 예측하여 점선으로 표시
2. 2022년부터 2026년 소득 분위에 따른 소득 증가율을 그래프로 표시

이로 인하여 내가 가진 자산과 소비 금액에 따른 소득을 예측하고 소득이 시간에 따른 소득 분위의 변화량을 간접적으로 알 수 있습니다.

3 데이터 시각화 - 미래의 예측 소득과 소득 분위의 위치를 파악 (그래프, 예측분석)



소득 5분위 기준으로 내 자산과 어떤 차이를 가지는지 보여줍니다.

-%이면 기준을 넘었다는 의미이며
+%이면 기준에 못 미친다는 의미입니다.

그리고 각 부분에서 가장 가까운 소득 분위 %를 알려줍니다.

2022년도 소득 분위별 차이 (%):
소득 1분위: -84.07%
소득 2분위: -63.28%
소득 3분위: -39.47%
소득 4분위: -9.09%
소득 5분위: 65.69%
이 년도 중 가장 가까운 차이: -9.09%

2023년도 소득 분위별 차이 (%):
소득 1분위: -83.38%
소득 2분위: -62.04%
소득 3분위: -37.40%
소득 4분위: -5.87%
소득 5분위: 71.11%
이 년도 중 가장 가까운 차이: -5.87%

2024년도 소득 분위별 차이 (%):
소득 1분위: -82.68%
소득 2분위: -60.80%
소득 3분위: -35.32%
소득 4분위: -2.66%
소득 5분위: 76.54%
이 년도 중 가장 가까운 차이: -2.66%

2025년도 소득 분위별 차이 (%):
소득 1분위: -81.99%
소득 2분위: -59.56%
소득 3분위: -33.25%
소득 4분위: 0.56%
소득 5분위: 81.96%
이 년도 중 가장 가까운 차이: 0.56%

2026년도 소득 분위별 차이 (%):
소득 1분위: -81.29%
소득 2분위: -58.32%
소득 3분위: -31.17%
소득 4분위: 3.78%
소득 5분위: 87.39%
이 년도 중 가장 가까운 차이: 3.78%