

# 대주제: 경제 지표에 따른 사회 현상

## 세부 주제: 경기에 따른 부동산 매매가격, 전세가격 변화

- 1조 이소정
- 1인당 국민총소득(GNI)는 경기를 나타내는 지표 중 하나로, GNI가 변화하는 것에 따른 부동산 매매가격 지수, 전세가격 지수의 변화가 나타나는 가에 대해 알아보았습니다.
- 데이터

- GNI 데이터: 우리나라 국민이 국내는 물론 국외에서 벌어들인 소득의 실질구매력을 나타내는 지표로 실질 GDP에서 외국인이 국내에서 벌어들인 실질소득은 차감하고 우리 국민이 국외에서 벌어들인 실질소득은 더하여 산출했다.  
출처: 통계청 <https://www.index.go.kr/unify/idx-info.do?idxCd=4221>  
(<https://www.index.go.kr/unify/idx-info.do?idxCd=4221>)
- 주택 매매가격 지수 및 전세가격 지수: 전국 주택시장의 매매, 전세, 월세가격을 조사하여 주택시장의 평균적인 가격변화를 측정, 2021년 6월을 데이터를 100으로 두고 계산하였다.  
출처: 한국부동산원 [https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE\\_21111](https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE_21111) ([https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE\\_21111](https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE_21111))

In [1]:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

In [2]:

```
1 import warnings
2 warnings.filterwarnings('ignore')
```

## GNI 데이터 가져오기

In [3]:

```
1 df1 = pd.read_excel('./data/GNI_excel.xlsx', header=2, index_col=1, nrows =  
4) # 2번 째 행에서 1행부터 4개열 들고오기  
2 df1
```

Out[3]:

	Unnamed: 0	2004	2005	2006	2007	2008	2009	2010	2011
명목/ 원화	1인 당 GNI 2)	1,886.1	1,973.0	2,070.0	2,232.6	2,353.5	2,440.7	2,673.0	2,798.0
명목/ 달러 화	NaN	16,476.9	19,261.8	21,663.7	24,026.8	21,345.3	19,122.1	23,117.7	25,255.0
실질/ 원화	NaN	2,331.5	2,383.0	2,464.9	2,592.0	2,582.2	2,632.6	2,808.3	2,832.0
증가 율 3)	NaN	3.9	2.2	3.4	5.2	-0.4	2.0	6.7	0.8



In [4]:

```
1 # df1 정보확인
2 print(df1.info(), df1.index, df1.columns, sep = '\n\n')
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, 명목/원화 to 증가율 3)
Data columns (total 19 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0    1 non-null      object
1   2004          4 non-null      object
2   2005          4 non-null      object
3   2006          4 non-null      object
4   2007          4 non-null      object
5   2008          4 non-null      object
6   2009          4 non-null      object
7   2010          4 non-null      object
8   2011          4 non-null      object
9   2012          4 non-null      object
10  2013          4 non-null      object
11  2014          4 non-null      object
12  2015          4 non-null      object
13  2016          4 non-null      object
14  2017          4 non-null      object
15  2018          4 non-null      object
16  2019          4 non-null      object
17  2020          4 non-null      object
18  2021          4 non-null      object
dtypes: object(19)
memory usage: 640.0+ bytes
None
```

```
Index(['명목/원화', '명목/달러화', '실질/원화', '증가율 3'], dtype='object')
```

```
Index(['Unnamed: 0', '2004', '2005', '2006', '2007', '2008', '2009', '2010',
      '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019',
      '2020', '2021'],
      dtype='object')
```

- 연도에 따른 실질 GNI 지수 저장

In [5]:

```
1 gni_df = df1.iloc[2, 1:].reset_index().rename(columns = {'index' : 'year',  
  '실질/원화' : 'GNI'})  
2 gni_df
```

Out[5]:

	year	GNI
0	2004	2,331.5
1	2005	2,383.0
2	2006	2,464.9
3	2007	2,592.0
4	2008	2,582.2
5	2009	2,632.6
6	2010	2,808.3
7	2011	2,832.1
8	2012	2,899.7
9	2013	2,997.5
10	2014	3,082.8
11	2015	3,260.2
12	2016	3,390.7
13	2017	3,492.5
14	2018	3,530.9
15	2019	3,532.4
16	2020	3,529.5
17	2021	3,656.2

GNI 데이터가 object로 수치 데이터로 변경

In [6]:

```
1 gni_df.GNI = gni_df.GNI.apply(lambda x: float(x.replace(',', '')))
2 gni_df
```

Out[6]:

	year	GNI
0	2004	2331.5
1	2005	2383.0
2	2006	2464.9
3	2007	2592.0
4	2008	2582.2
5	2009	2632.6
6	2010	2808.3
7	2011	2832.1
8	2012	2899.7
9	2013	2997.5
10	2014	3082.8
11	2015	3260.2
12	2016	3390.7
13	2017	3492.5
14	2018	3530.9
15	2019	3532.4
16	2020	3529.5
17	2021	3656.2

부동산 매매가격지수 데이터 가져오기

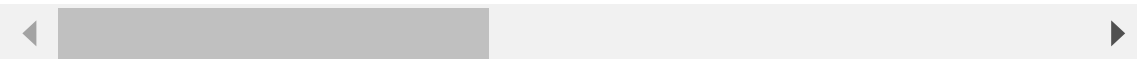
In [7]:

```
1 df2 = pd.read_excel('./data/월간_매매가격지수_종합.xlsx',header=10)
2 df2.head(5)
```

Out[7]:

	지 역	Unnamed: 1	Unnamed: 2	Unnamed: 3	2003년 11 월	2003년 12 월	2004년 01 월	2004년 02 월	2004년 03 월	2004년 04 월
0	전 국	NaN	NaN	NaN	61.451824	60.963185	60.652009	60.743829	60.842376	60.942376
1	수 도 권	NaN	NaN	NaN	55.765538	55.359195	55.116437	55.228958	55.376448	55.476448
2	지 방 권	NaN	NaN	NaN	68.358856	67.760797	67.3565	67.416319	67.441927	67.541927
3	6 대 광 역 시	NaN	NaN	NaN	60.555931	59.922991	59.557162	59.574504	59.636041	59.736041
4	5 대 광 역 시	NaN	NaN	NaN	61.78014	61.119917	60.744738	60.787542	60.892069	60.992069

5 rows × 233 columns



부동산 전세가격지수 데이터 가져오기

In [8]:

```
1 df3 = pd.read_excel('./data/월간_전세가격지수_종합.xlsx', header = 10)
2 df3.head(5)
```

Out[8]:

	지 역	Unnamed: 1	Unnamed: 2	Unnamed: 3	2003년 12 월	2004년 01 월	2004년 02 월	2004년 03 월	2004년 04 월	2004년 05 월
0	전 국	NaN	NaN	NaN	59.575331	59.209691	59.30727	59.466954	59.514041	59.551041
1	수 도 권	NaN	NaN	NaN	53.381388	52.998164	53.078503	53.244122	53.249467	53.254467
2	지 방 권	NaN	NaN	NaN	66.518315	66.193285	66.312755	66.457653	66.564598	66.564598
3	6 대 광 역 시	NaN	NaN	NaN	60.423033	59.92439	59.967838	60.110615	60.176017	60.176017
4	5 대 광 역 시	NaN	NaN	NaN	62.114685	61.649051	61.717534	61.88707	61.986271	61.986271

5 rows × 232 columns

In [9]:

```
1 # 전국의 주택 매매 가격 지수 가져오기
2 sale = df2.iloc[0, 4:].to_frame().reset_index().rename(columns =
3 {'index':'year', 0:'Sales'})
4 sale.head(5)
```

Out[9]:

	year	Sales
0	2003년 11월	61.451824
1	2003년 12월	60.963185
2	2004년 01월	60.652009
3	2004년 02월	60.743829
4	2004년 03월	60.842376

In [10]:

```
1 # 전국의 주택 전세 가격 지수 가져오기
2 charter = df3.iloc[0, 4:].to_frame().reset_index().rename(columns =
  {'index': 'year', 0: 'Jeonse'})
3 charter.head(5)
```

Out[10]:

	year	Jeonse
0	2003년 12월	59.575331
1	2004년 01월	59.209691
2	2004년 02월	59.30727
3	2004년 03월	59.466954
4	2004년 04월	59.514041

In [11]:

```
1 # 주택 가격에 관한 데이터를 하나의 데이터프레임으로 변환
2 house = pd.merge(sale, charter)
3 house.head(5)
```

Out[11]:

	year	Sales	Jeonse
0	2003년 12월	60.963185	59.575331
1	2004년 01월	60.652009	59.209691
2	2004년 02월	60.743829	59.30727
3	2004년 03월	60.842376	59.466954
4	2004년 04월	60.935649	59.514041

In [12]:

```
1 # 정보확인
2 house.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 228 entries, 0 to 227
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   year    228 non-null      object
1   Sales   228 non-null      object
2   Jeonse  228 non-null      object
dtypes: object(3)
memory usage: 7.1+ KB
```



In [13]:

```
1 # 연도 뽑기
2 house.year = house.year.apply(lambda x: x[:4])
3
4 # 매매 및 전세 데이터 타입 변경
5 house[['Sales', 'Jeonse']] = house[['Sales', 'Jeonse']].astype('float64')
6 house
```

Out[13]:

	year	Sales	Jeonse
0	2003	60.963185	59.575331
1	2004	60.652009	59.209691
2	2004	60.743829	59.307270
3	2004	60.842376	59.466954
4	2004	60.935649	59.514041
...	...	...	...
223	2022	104.758261	103.123228
224	2022	104.459050	102.837952
225	2022	103.949855	102.327543
226	2022	103.148521	101.425319
227	2022	101.736171	99.856381

228 rows × 3 columns

In [14]:

```
1 # 정보확인
2 house.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 228 entries, 0 to 227
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   year    228 non-null     object
1   Sales   228 non-null     float64
2   Jeonse  228 non-null     float64
dtypes: float64(2), object(1)
memory usage: 7.1+ KB
```

예시	▪ 주택A,B,C,D의 '20년 가격이 10,000 / 10,000 / 10,000 / 20,000 이고 '21년 가격이 10,000 / 10,000 / 10,000 / 50,000 라고 할 때, 각 지수별 변동률 계산		
	기하평균 변동률 (주택가격동향)	산술평균 변동률 (KB)	시가총액 변동률 (R114,공시가격)
	$(1.0 \times 1.0 \times 1.0 \times 2.5)^{1/4}$ $= \underline{1.257}$ 가격 변동률: 25.7%	$(1.0 + 1.0 + 1.0 + 2.5) / 4$ $= \underline{1.375}$ 가격 변동률: 37.5%	$(8.0 / 5.0)$ $= \underline{1.600}$ 가격 변동률: 60.0%

[https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE\\_21111](https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE_21111) ([https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE\\_21111](https://www.reb.or.kr/r-one/statistics/statisticsViewer.do?menuId=HOUSE_21111))

주택 매매 및 전세 가격 지수는 월별 데이터로 연간 데이터로 변경 시, 한국부동산원에서 권장한 주택가격동향 계산 방법은 기하평균이다.

In [15]:

```
1 import statistics as st
2 # st.geometric_mean(list_values) 기하평균 값들을 곱한 뒤 루트를 씌운 값
3
4 y, s, c = [], [], [] # year, sale, charter
5 for i in range(2004, 2022):
6     y.append(str(i))
7     h = house[house.year == str(i)]
8     s.append(st.geometric_mean(h.Sales))
9     c.append(st.geometric_mean(h.Jeonse))
```

In [16]:

```
1 # 리스트 정보확인
2 print(y, len(y))
3 print(s, len(s))
4 print(c, len(c))
```

```
['2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
'2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021'] 18
[60.429505728866744, 60.72042461374813, 64.32713379369024, 71.22541433947217, 7
6.23084137539591, 77.10885695066271, 78.83645953119704, 82.41983923079086, 83.96
065325795139, 82.89956036590208, 84.10434609628173, 86.33847894304232, 87.948797
076428, 88.945111925013, 90.20689448382032, 90.02018907547335, 92.5655139817435
6, 100.39539319016923] 18
[58.2377054692459, 56.779507864791874, 59.628052098694596, 63.20105989041464, 6
5.9782106254169, 66.39513670731888, 70.43309908951335, 78.02215289454371, 81.927
72058059438, 84.31912825565601, 88.16471055810071, 91.70571656932626, 94.3535426
3187776, 95.25793495168736, 94.57702245564961, 92.67553429344306, 94.23219961426
753, 100.39136684515964] 18
```

In [17]:

```
1 # 연도별 지수를 계산한 데이터로 데이터 프레임 만들기
2 house_3 = pd.DataFrame({'year': y, 'Sales':s, 'Jeonse': c})
3 house_3
```

Out[17]:

	year	Sales	Jeonse
0	2004	60.429506	58.237705
1	2005	60.720425	56.779508
2	2006	64.327134	59.628052
3	2007	71.225414	63.201060
4	2008	76.230841	65.978211
5	2009	77.108857	66.395137
6	2010	78.836460	70.433099
7	2011	82.419839	78.022153
8	2012	83.960653	81.927721
9	2013	82.899560	84.319128
10	2014	84.104346	88.164711
11	2015	86.338479	91.705717
12	2016	87.948797	94.353543
13	2017	88.945112	95.257935
14	2018	90.206894	94.577022
15	2019	90.020189	92.675534
16	2020	92.565514	94.232200
17	2021	100.395393	100.391367

In [18]:

```
1 house_3.info()
2 gni_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  ---
0   year     18 non-null    object
1   Sales    18 non-null    float64
2   Jeonse   18 non-null    float64
dtypes: float64(2), object(1)
memory usage: 560.0+ bytes
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 2 columns):
#   Column   Non-Null Count  Dtype
---  ---
0   year     18 non-null    object
1   GNI      18 non-null    float64
dtypes: float64(1), object(1)
memory usage: 416.0+ bytes
```

In [19]:

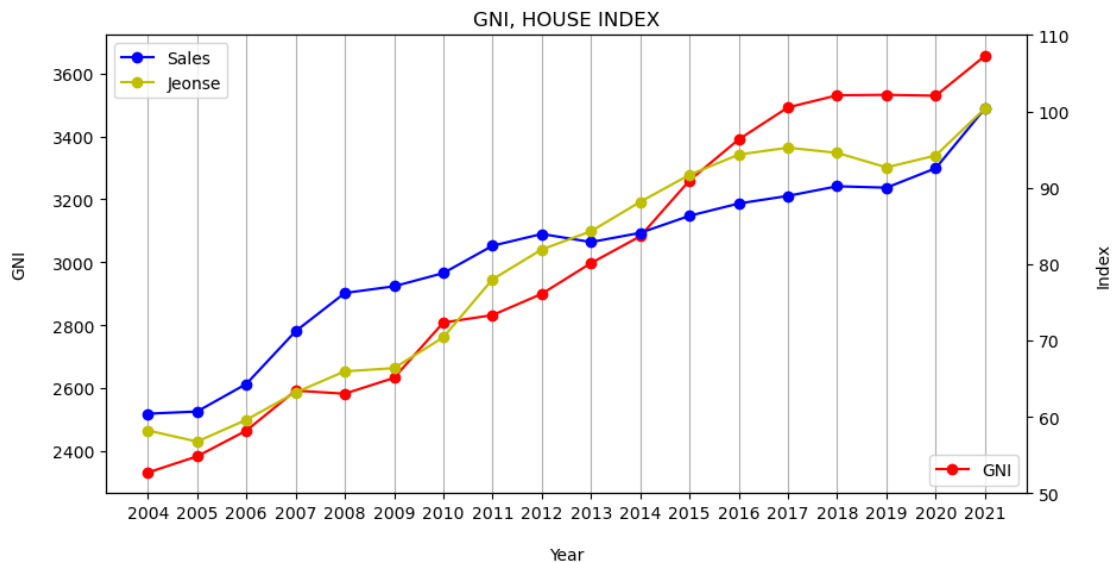
```
1 # gni와 주택 지수의 데이터 합치기
2 total = pd.merge(gni_df, house_3)
3 total
```

Out[19]:

	year	GNI	Sales	Jeonse
0	2004	2331.5	60.429506	58.237705
1	2005	2383.0	60.720425	56.779508
2	2006	2464.9	64.327134	59.628052
3	2007	2592.0	71.225414	63.201060
4	2008	2582.2	76.230841	65.978211
5	2009	2632.6	77.108857	66.395137
6	2010	2808.3	78.836460	70.433099
7	2011	2832.1	82.419839	78.022153
8	2012	2899.7	83.960653	81.927721
9	2013	2997.5	82.899560	84.319128
10	2014	3082.8	84.104346	88.164711
11	2015	3260.2	86.338479	91.705717
12	2016	3390.7	87.948797	94.353543
13	2017	3492.5	88.945112	95.257935
14	2018	3530.9	90.206894	94.577022
15	2019	3532.4	90.020189	92.675534
16	2020	3529.5	92.565514	94.232200
17	2021	3656.2	100.395393	100.391367

In [20]:

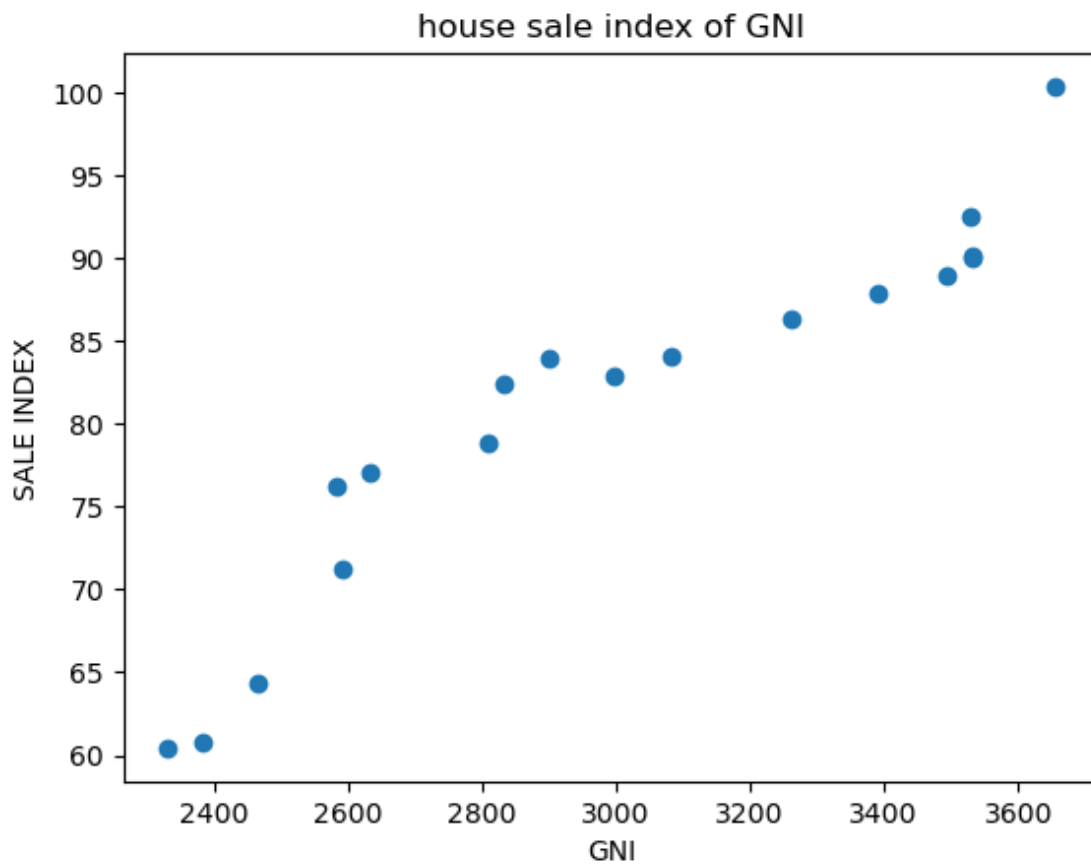
```
1 # GNI, 주택매매가격지수, 주택전세가격지수의 분포
2 fig, ax1 = plt.subplots(figsize = (10, 5))
3 ax2 = ax1.twinx()
4
5 ax1.plot(total.year, total.GNI, 'ro-', label = 'GNI')
6 ax2.plot(total.year, total.Sales, 'bo-', label = 'Sales')
7 ax2.plot(total.year, total.Jeonse, 'yo-', label = 'Jeonse')
8
9 ax1.set_ylabel('GNI', labelpad=15)
10 ax2.set_ylabel('Index', labelpad=15)
11
12 ax1.set_xlabel('Year', labelpad=15)
13
14 ax2.set_ylim(50, 110)
15
16 ax1.grid(axis = 'x')
17
18 ax1.legend(loc = 4)
19 ax2.legend(loc = 'upper left')
20
21 plt.title('GNI, HOUSE INDEX')
22
23 plt.show()
```



- gni에 따른 부동산 가격 변동

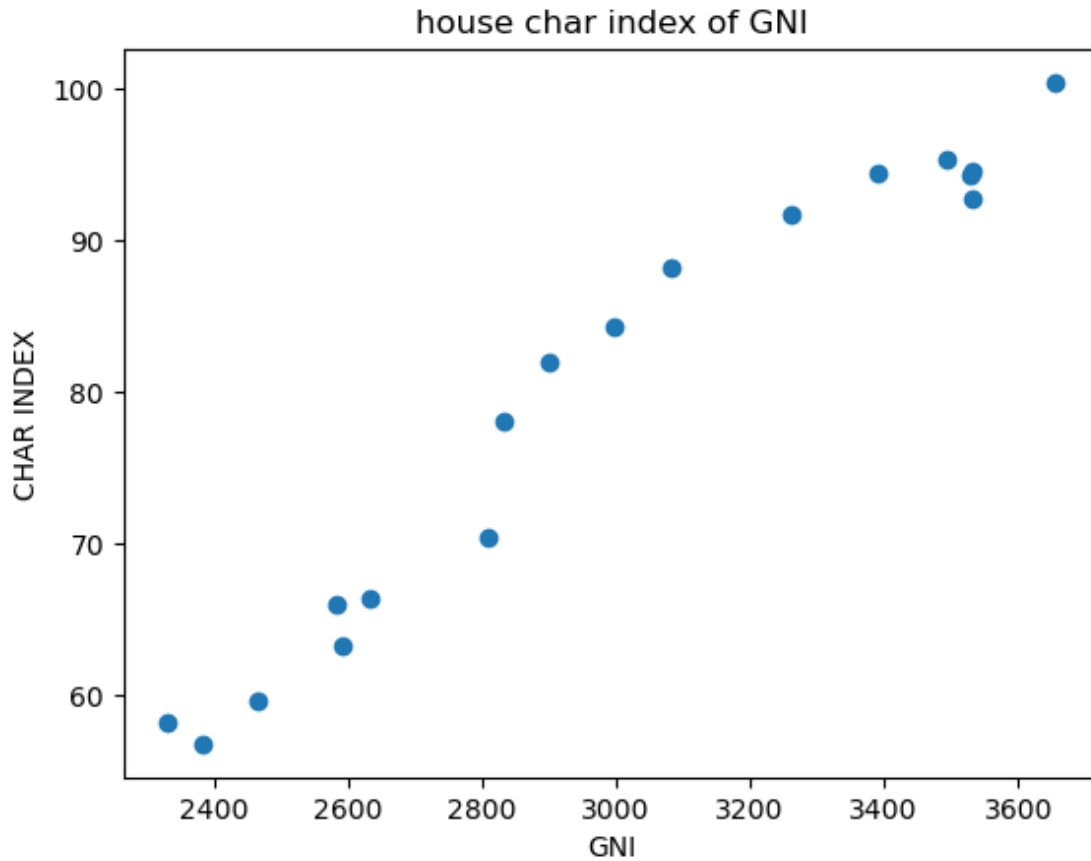
In [21]:

```
1 # GNI와 주택전세가격 지수의 산점도
2 import matplotlib.pyplot as plt
3
4 plt.scatter(total.GNI, total.Sales)
5
6 plt.xticks()
7 plt.xlabel('GNI')
8 plt.ylabel('SALE INDEX')
9 plt.title('house sale index of GNI')
10 plt.show()
```



In [22]:

```
1 # GNI와 주택전세가격 지수의 산점도
2 plt.scatter(total.GNI, total.Jeonse)
3 plt.xlabel('GNI')
4 plt.ylabel('CHAR INDEX')
5 plt.title('house char index of GNI')
6 plt.show()
```



In [23]:

```
1 # 상관관계 분석
2 cor = total.corr(method='pearson')
3 cor
```

Out[23]:

	GNI	Sales	Jeonse
GNI	1.000000	0.938932	0.975619
Sales	0.938932	1.000000	0.949660
Jeonse	0.975619	0.949660	1.000000



In [24]:

```
1 sns.heatmap(cor, annot=True, fmt=".2f", cmap = 'YlGnBu')
2 plt.title('Correlation\nGNI, housing sales price index, housing jeonse price
3 plt.show()
index')
```



- **GNI와 주택매매가격지수**의 상관관계를 분석한 결과 상관계수가 0.938932으로 상관정도가 매우 높게 나왔습니다.
  - GNI가 높아질수록 주택매매가격지수 또한 높아집니다.
- **GNI와 주택전세가격지수**의 상관관계를 분석한 결과 상관계수가 0.975619 상관정도가 매우 높게 나왔습니다.
  - GNI가 높아질수록 주택전세가격지수 또한 높아집니다.

▶ 경기가 안좋으면 부동산을 매매와 전세 시 부담높고, 경기가 좋아지면 상대적으로 매매, 전세 시 부담이 적어진다

범죄, 자살 상관X 물가, 자동차,