

In [1]:

```
#!/usr/bin/env python
#-*-coding:utf-8 -*-

import pathlib
import random
from functools import reduce
from collections import defaultdict

import pandas as pd
import geopandas as gpd
import folium
import shapely
import numpy as np
from IPython.display import display
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm
import xgboost
import sklearn.cluster
import tensorflow as tf
import matplotlib as mpl
import seaborn as sns

from pandas import DataFrame
from geoband import API
```

## 제공된 데이터

In [2]:

```
input_path = pathlib.Path('./input')
if not input_path.is_dir():
    input_path.mkdir()

from geoband.API import *
GetCompasData('SBJ_2102_001', '1', '1.세종시 전유부.csv')
GetCompasData('SBJ_2102_001', '2', '2.세종시 표제부.csv')
GetCompasData('SBJ_2102_001', '3', '3.세종시 아파트(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '4', '4.세종시 연립다세대(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '5', '5.세종시 단독다가구(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '6', '6.세종시 오피스텔(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '7', '7.세종시 아파트(전월세)_실거래가.csv')
GetCompasData('SBJ_2102_001', '8', '8.세종시 연립다세대(전월세)_실거래가.csv')
GetCompasData('SBJ_2102_001', '9', '9.세종시 단독다가구(전월세)_실거래가.csv')
GetCompasData('SBJ_2102_001', '10', '10.세종시 오피스텔(전월세)_실거래가.csv')
GetCompasData('SBJ_2102_001', '11', '11.세종시 상업업무용(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '12', '12.세종시 토지(매매)_실거래가.csv')
GetCompasData('SBJ_2102_001', '13', '13.세종시 분양권_실거래가.csv')
GetCompasData('SBJ_2102_001', '14', '14.세종시 상권정보.csv')
GetCompasData('SBJ_2102_001', '15', '15.세종시 상권정보_업종코드.csv')
GetCompasData('SBJ_2102_001', '16', '16.세종시 사업체_매출정보.geojson')
GetCompasData('SBJ_2102_001', '17', '17.세종시 사업체_매출정보.csv')
GetCompasData('SBJ_2102_001', '18', '18.세종시 개별공시지가(2017~2020).csv')
GetCompasData('SBJ_2102_001', '19', '19.세종시 연령별_거주인구정보_격자.geojson')
GetCompasData('SBJ_2102_001', '20', '20.세종시 전입자수.csv')
GetCompasData('SBJ_2102_001', '21', '21.세종시 전출자수.csv')
GetCompasData('SBJ_2102_001', '22', '22.세종시 연령별_인구현황.csv')
GetCompasData('SBJ_2102_001', '23', '23.세종시 도로명주소_건물.geojson')
GetCompasData('SBJ_2102_001', '24', '24.세종시 건축물연면적_격자.geojson')
GetCompasData('SBJ_2102_001', '25', '25.세종시 가구 월평균 소비지출액.csv')
GetCompasData('SBJ_2102_001', '26', '26.세종시 거주기간.csv')
GetCompasData('SBJ_2102_001', '27', '27.세종시 월평균 가구소득.csv')
GetCompasData('SBJ_2102_001', '28', '28.세종시 지역별_세대원수별_세대수.csv')
GetCompasData('SBJ_2102_001', '29', '29.세종시 거주 의사(향후).csv')
GetCompasData('SBJ_2102_001', '30', '30.세종시 법정경계(시군구).geojson')
GetCompasData('SBJ_2102_001', '31', '31.세종시 법정경계(읍면동).geojson')
GetCompasData('SBJ_2102_001', '32', '32.세종시 행정경계(읍면동).geojson')
GetCompasData('SBJ_2102_001', '33', '33.세종시 지적도.geojson')
GetCompasData('SBJ_2102_001', '34', '34.세종시 건물분포_이미지.zip')

for path in list(input_path.glob('*.*csv')) + list(input_path.glob('*.*geojson')):
```

```
print(path)
```

```
input/1.세종시_전유부.csv
input/10.세종시_오피스텔(전월세)_실거래가.csv
input/11.세종시_상업업무용(매매)_실거래가.csv
input/12.세종시_토지(매매)_실거래가.csv
input/13.세종시_분양권_실거래가.csv
input/14.세종시_상권정보.csv
input/15.세종시_상권정보_업종코드.csv
input/17.세종시_사업체_매출정보.csv
input/18.세종시_개별공시지가(2017~2020).csv
input/2.세종시_표제부.csv
input/20.세종시_전입자수.csv
input/21.세종시_전출자수.csv
input/22.세종시_연령별_인구현황.csv
input/28.세종시_지역별_세대원수별_세대수.csv
input/25.세종시_가구_월평균_소비지출액.csv
input/26.세종시_거주기간.csv
input/27.세종시_월평균_가구소득.csv
input/29.세종시_거주의사(향후).csv
input/3.세종시_아파트(매매)_실거래가.csv
input/4.세종시_연립다세대(매매)_실거래가.csv
input/5.세종시_단독다가구(매매)_실거래가.csv
input/6.세종시_오피스텔(매매)_실거래가.csv
input/7.세종시_아파트(전월세)_실거래가.csv
input/8.세종시_연립다세대(전월세)_실거래가.csv
input/9.세종시_단독다가구(전월세)_실거래가.csv
input/16.세종시_사업체_매출정보.geojson
input/19.세종시_연령별_거주인구정보_격자.geojson
input/23.세종시_도로명주소_건물.geojson
input/24.세종시_건축물연면적_격자.geojson
input/30.세종시_법정경계(시군구).geojson
input/31.세종시_법정경계(읍면동).geojson
input/32.세종시_행정경계(읍면동).geojson
input/33.세종시_지적도.geojson
```

## 지역별 매매 평균값 그래프

In [3]:

```
# 자료 불러오기
아파트_매매 = pd.read_csv(input_path.joinpath('3.세종시_아파트(매매)_실거래가.csv'), encoding='cp949')
아파트_매매

# 세종 특별자치시 없애기
for i in range(len(아파트_매매['시군구'])):
    아파트_매매['시군구'][i] = 아파트_매매['시군구'][i][7:]

# col 확인
아파트_매매.columns

# 주소별 거래금액 평균 구하기
data_아파트 = 아파트_매매.groupby('시군구')['거래금액(만원)'].mean()
data_아파트

# 그래프 한글깨짐 방지
import matplotlib
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

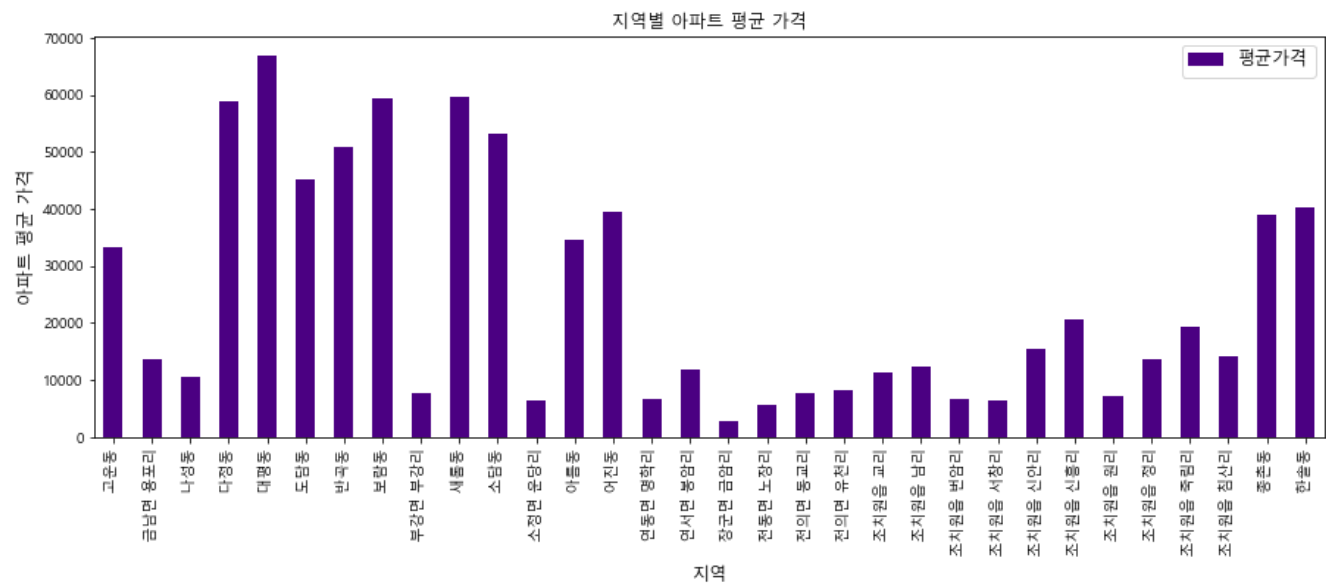
# 지역 별 평균 가격 그래프
ax = data_아파트.plot(kind='bar', title='지역별 아파트 평균 가격', figsize=(15, 5), legend=True,
fontsize=10, color='indigo')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('아파트 평균 가격', fontsize=12)
ax.legend(['평균가격'], fontsize=12)
```

/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
import sys

Out [3]:

<matplotlib.legend.Legend at 0x7f2b19a449e8>



In [4]:

```
# 자료 불러오기
연립다세대_매매 = pd.read_csv(input_path.joinpath('4.세종시_연립다세대 (매매)_실거래가.csv'), thousands =
',', encoding='cp949')
연립다세대_매매

# 세종 특별자치시 없애기
for i in range(len(연립다세대_매매['시군구'])):
    연립다세대_매매['시군구'][i] = 연립다세대_매매['시군구'][i][7:]

# col 확인
연립다세대_매매.columns

# 주소별 거래금액 평균 구하기
data_연립다세대 = 연립다세대_매매.groupby('시군구')['거래금액 (만원)'].mean()
data_연립다세대

# 그래프 한글깨짐 방지
import matplotlib
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

# 지역 별 평균 가격 그래프
ax = data_연립다세대.plot(kind='bar', title='지역별 연립다세대 평균 가격', figsize=(15, 5), legend=True,
e, fontsize=10, color='deepskyblue')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('연립다세대 평균 가격', fontsize=12)
ax.legend(['평균가격'], fontsize=12)
```

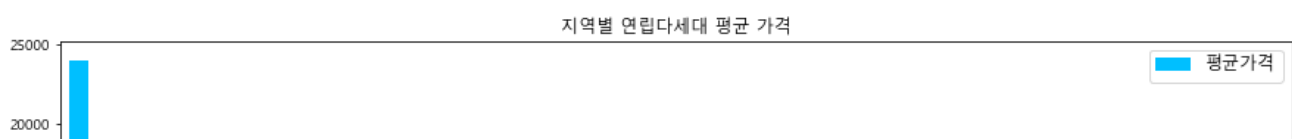
/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:7: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

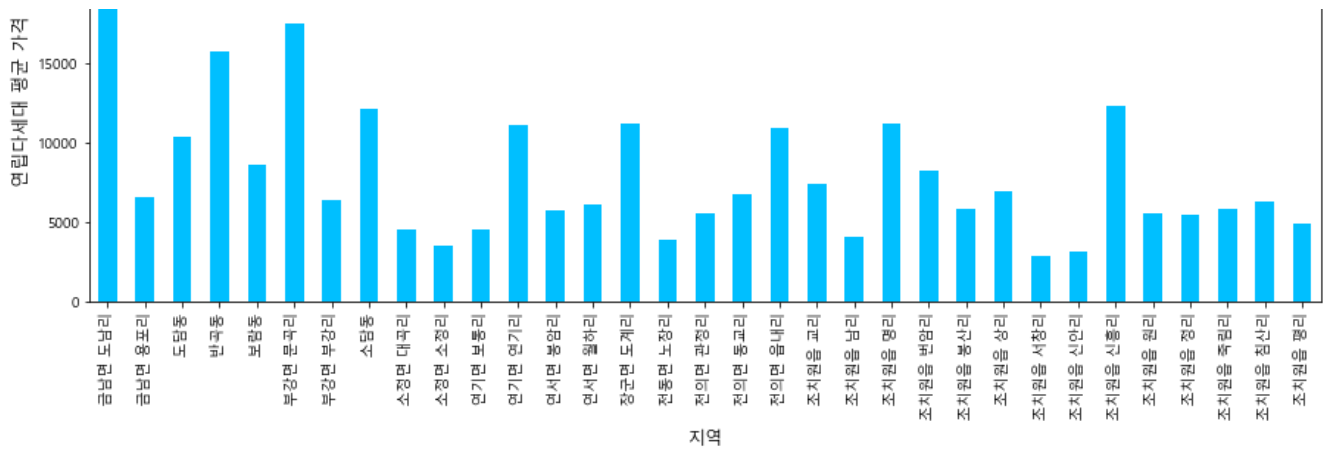
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
import sys
```

Out [4]:

<matplotlib.legend.Legend at 0x7f2b3cebf278>





In [5]:

```
# 자료 불러오기
단독다가구_매매 = pd.read_csv(input_path.joinpath('5.세종시_단독다가구 (매매)_실거래가.csv'), encoding='cp949')
단독다가구_매매

# 세종 특별자치시 없애기
for i in range(len(단독다가구_매매['시군구'])):
    단독다가구_매매['시군구'][i] = 단독다가구_매매['시군구'][i][7:]

# col 확인
단독다가구_매매.columns

# 주소별 거래금액 평균 구하기
data_단독다가구 = 단독다가구_매매.groupby('시군구')['거래금액 (만원)'].mean()
data_단독다가구

# 그래프 한글깨짐 방지
import matplotlib
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

# 지역 별 평균 가격 그래프
ax = data_단독다가구.plot(kind='bar', title='지역별 단독다가구 평균 가격', figsize=(80,15), legend=True,
                           fontsize=32, color='midnightblue')
ax.set_xlabel('지역', fontsize=30)
ax.set_ylabel('단독다가구 평균 가격', fontsize=30)
ax.legend(['평균 가격'], fontsize=30)
```

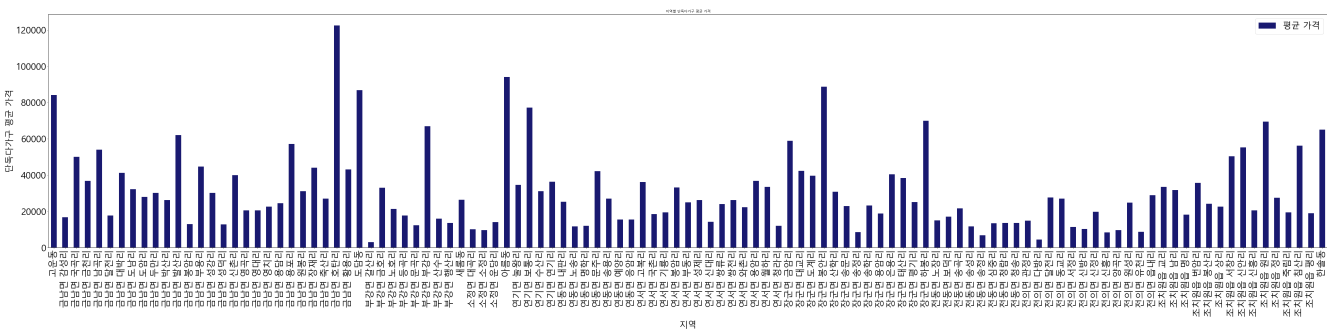
/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:7: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
import sys
```

Out[5]:

<matplotlib.legend.Legend at 0x7f2a87b3b160>



In [6]:

```
# 자료 불러오기
오피스텔_매매 = pd.read_csv(input_path.joinpath('6.세종시_오피스텔 (매매)_실거래가.csv'), encoding='cp949' )
오피스텔_매매

# 세종 특별자치시 없애기
for i in range(len(오피스텔_매매['시군구'])):
    오피스텔_매매['시군구'][i] = 오피스텔_매매['시군구'][i][7:]

# col 확인
오피스텔_매매.columns

# 주소별 거래금액 평균 구하기
data_오피스텔 = 오피스텔_매매.groupby('시군구')['거래금액 (만원)'].mean()
data_오피스텔

# 그래프 한글깨짐 방지
import matplotlib
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

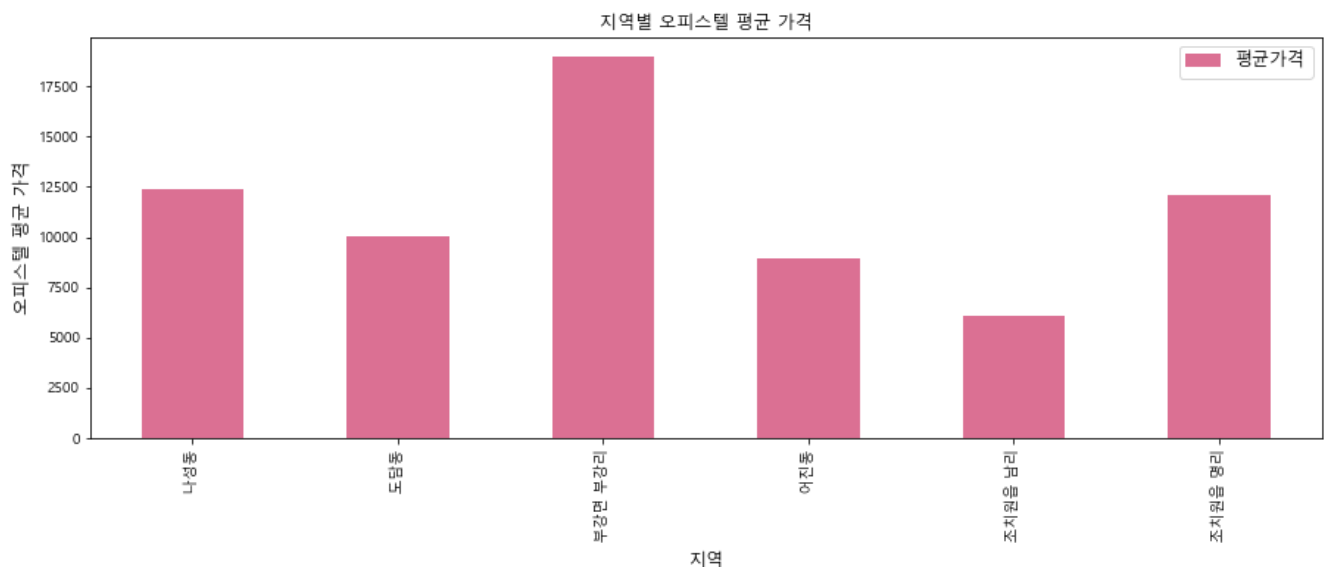
# 지역 별 평균 가격 그래프
ax = data_오피스텔.plot(kind='bar', title='지역별 오피스텔 평균 가격', figsize=(15, 5), legend=True,
fontsize=10, color='palevioletred')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('오피스텔 평균 가격', fontsize=12)
ax.legend(['평균가격'], fontsize=12)

/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
import sys
```

Out[6]:

<matplotlib.legend.Legend at 0x7f2a890e95c0>



In [ ]:

## 지역별 전,월세 평균값 그래프

In [7]:

```
# 자료 불러오기
아파트 전월세 = pd.read_csv(input_path.joinpath('7.세종시_아파트 (전월세)_실거래가.csv'), thousands = ',', encoding='cp949' )
아파트 전월세
```

```

# 행 확인
아파트_전월세.columns

# 세종특별자치시 없애기
for i in range(len(아파트_전월세['시군구'])):
    아파트_전월세['시군구'][i] = 아파트_전월세['시군구'][i][7:]

# 지역마다 전, 월세 평균
data_아파트_전월세 = 아파트_전월세.groupby(['시군구', '전월세구분'])[['보증금 (만원)', '월세 (만원)']].mean()
data_아파트_전월세 = data_아파트_전월세.unstack().fillna(0)
data_아파트_전월세

data_아파트_전월세_월세보증금 = data_아파트_전월세['보증금 (만원)', '월세']
data_아파트_전월세_월세 = data_아파트_전월세['월세 (만원)', '월세']
data_아파트_전세 = data_아파트_전월세['보증금 (만원)', '전세']

#data_아파트_전월세['보증금 (만원)', '전세'] = data_아파트_전세
del(data_아파트_전월세['월세 (만원)', '전세'])
del(data_아파트_전월세['보증금 (만원)', '전세'])

round(data_아파트_전월세, 2)

# 그래프 한글깨짐 방지
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

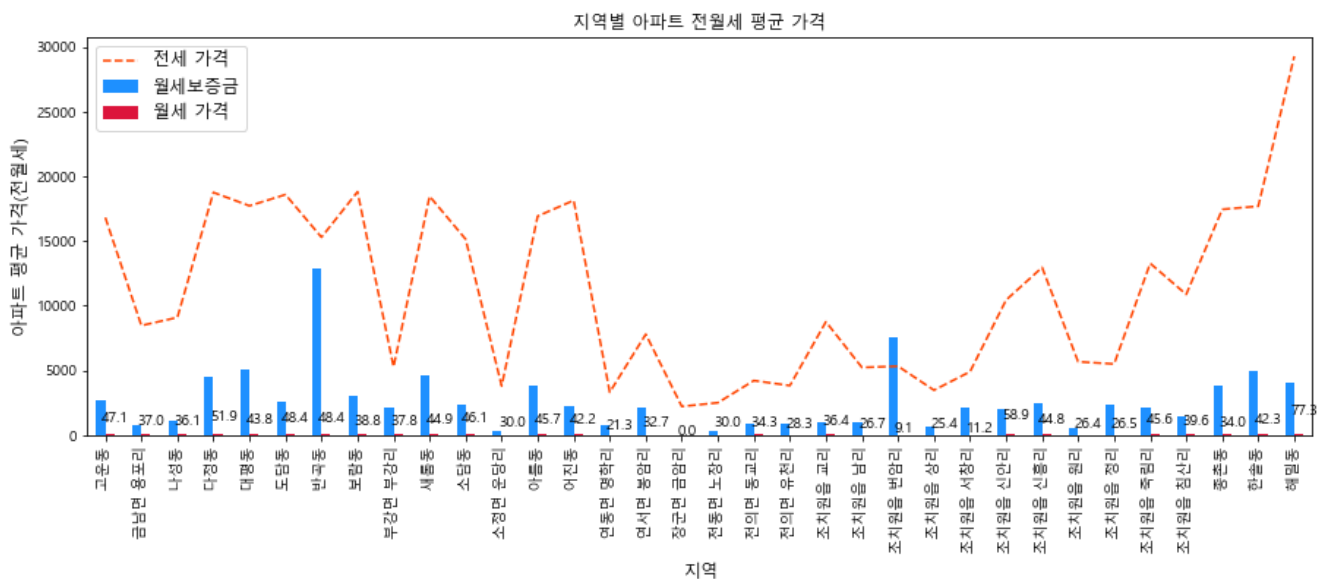
# 지역 별 평균 가격 그래프
ax = data_아파트_전월세.plot(kind='bar', title='지역별 아파트 전월세 평균 가격', figsize=(15, 5),
legend=True, fontsize=10, color=['dodgerblue', 'crimson'])
plt.plot(data_아파트_전세, linestyle='--', color='orangered')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('아파트 평균 가격(전월세)', fontsize=12)
ax.legend(['전세 가격', '월세보증금', '월세 가격'], fontsize=12)

for i, v in enumerate(data_아파트_전월세['월세 (만원)', '월세']):
    ax.text(i-0.1, v*v*20, str(round(v, 1)))

```

/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:10: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
# Remove the CWD from sys.path while we load stuff.



In [8]:

```

# 자료 불러오기
연립다세대_전월세= pd.read_csv(input_path.joinpath('8.세종시_연립다세대 (전월세)_실거래가.csv'), thousands =
',', encoding='cp949' )
연립다세대_전월세

```

```
# 행 확인
연립다세대_전월세.columns

# 세종특별자치시 없애기
for i in range(len(연립다세대_전월세['시군구'])):
    연립다세대_전월세['시군구'][i] = 연립다세대_전월세['시군구'][i][7:]

# 지역마다 전, 월세 평균
data_연립다세대_전월세 = 연립다세대_전월세.groupby(['시군구', '전월세구분'])[['보증금 (만원)', '월세 (만원)']].mean()
data_연립다세대_전월세 = data_연립다세대_전월세.unstack().fillna(0)
data_연립다세대_전월세

data_연립다세대_전월세_월세보증금 = data_연립다세대_전월세['보증금 (만원)', '월세']
data_연립다세대_전월세_월세 = data_연립다세대_전월세['월세 (만원)', '월세']
data_연립다세대_전세 = data_연립다세대_전월세['보증금 (만원)', '전세']

#data_연립다세대_전월세['보증금 (만원)', '전세'] = data_연립다세대_전세
del(data_연립다세대_전월세['월세 (만원)', '전세'])
del(data_연립다세대_전월세['보증금 (만원)', '전세'])

round(data_연립다세대_전월세, 2)

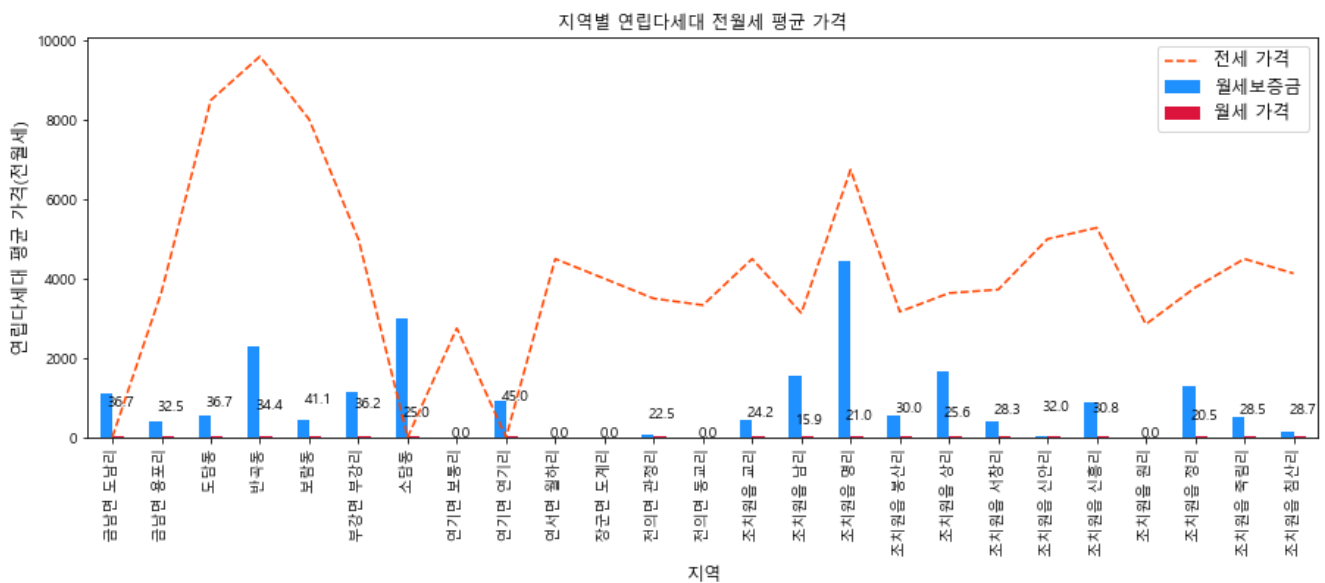
# 그래프 한글깨짐 방지
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

# 지역 별 평균 가격 그래프
ax = data_연립다세대_전월세.plot(kind='bar', title='지역별 연립다세대 전월세 평균 가격', figsize=(15, 5),
    legend=True, fontsize=10, color=['dodgerblue', 'crimson'])
plt.plot(data_연립다세대_전세, linestyle='--', color='orangered')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('연립다세대 평균 가격 (전월세)', fontsize=12)
ax.legend(['전세 가격', '월세보증금', '월세 가격'], fontsize=12)

for i, v in enumerate(data_연립다세대_전월세['월세 (만원)', '월세']):
    ax.text(i-0.1, v+v*20, str(round(v, 1)))
```

/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:10: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
 # Remove the CWD from sys.path while we load stuff.



In [9]:

```
# 자료 불러오기
단독다가구_전월세 = pd.read_csv(input_path.joinpath('9.세종시_단독다가구 (전월세)_실거래가.csv'), thousands =
    ',', encoding='cp949')
단독다가구_전월세
```

```
# 행 확인
단독다가구_전월세.columns

# 세종특별자치시 없애기
for i in range(len(단독다가구_전월세['시군구'])):
    단독다가구_전월세['시군구'][i] = 단독다가구_전월세['시군구'][i][7:]

# 지역마다 전, 월세 평균
data_단독다가구_전월세 = 단독다가구_전월세.groupby(['시군구', '전월세구분'])[['보증금(만원)', '월세(만원)']].mean()
data_단독다가구_전월세 = data_단독다가구_전월세.unstack().fillna(0)
data_단독다가구_전월세

data_단독다가구_전월세 월세보증금 = data_단독다가구_전월세['보증금(만원)', '월세']
data_단독다가구_전월세 월세 = data_단독다가구_전월세['월세(만원)', '월세']
data_단독다가구_전세 = data_단독다가구_전월세['보증금(만원)', '전세']

#data_단독다가구_전월세['보증금(만원)', '전세'] = data_단독다가구_전세
del(data_단독다가구_전월세['월세(만원)', '전세'])
del(data_단독다가구_전월세['보증금(만원)', '전세'])

round(data_단독다가구_전월세, 2)

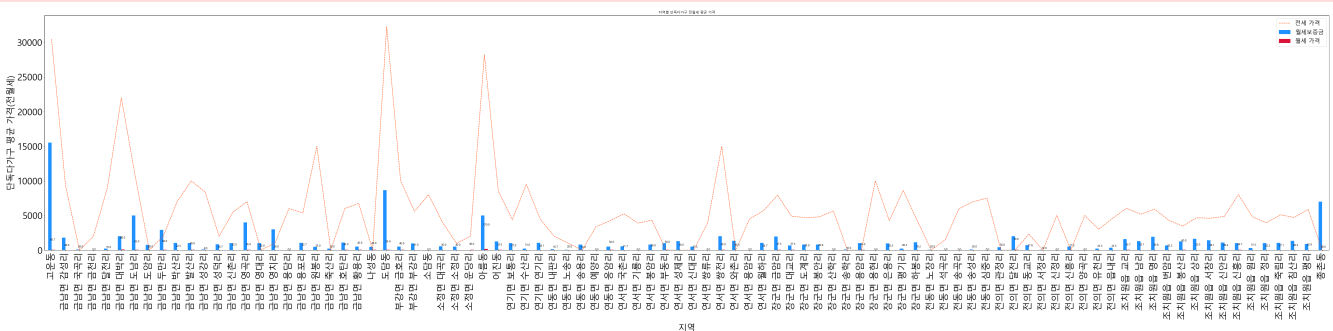
# 그래프 한글깨짐 방지
matplotlib.font_manager._rebuild()
plt.rc("font", family="Malgun Gothic")

# 지역 별 평균 가격 그래프
ax = data_단독다가구_전월세.plot(kind='bar', title='지역별 단독다가구 전월세 평균 가격', figsize=(80, 15),
    legend=True, fontsize=32, color=['dodgerblue', 'crimson'])
plt.plot(data_단독다가구_전세, linestyle='--', color='orangered')
ax.set_xlabel('지역', fontsize=30)
ax.set_ylabel('단독다가구 평균 가격(전월세)', fontsize=30)
ax.legend(['전세 가격', '월세보증금', '월세 가격'], fontsize=20)

for i, v in enumerate(data_단독다가구_전월세['월세(만원)', '월세']):
    ax.text(i-0.1, v+v*20, str(round(v, 1)))
```

/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:10: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
# Remove the CWD from sys.path while we load stuff.



In [10]:

```
# 자료 불러오기
오피스텔_전월세 = pd.read_csv(input_path.joinpath('10.세종시_오피스텔(전월세)_실거래가.csv'), thousands = ',', encoding='cp949')
오피스텔_전월세

# 행 확인
오피스텔_전월세.columns

# 세종특별자치시 없애기
for i in range(len(오피스텔_전월세['시군구'])):
    오피스텔_전월세['시군구'][i] = 오피스텔_전월세['시군구'][i][7:]

# 지역마다 전, 월세 평균
data_오피스텔_전월세 = 오피스텔_전월세.groupby(['시군구', '전월세구분'])[['보증금(만원)', '월세(만원)']].mean()
()
```



```

data_오피스텔_전월세 = data_오피스텔_전월세.unstack().fillna(0)
data_오피스텔_전월세

data_오피스텔_전월세_월세보증금 = data_오피스텔_전월세['보증금 (만원)', '월세']
data_오피스텔_전월세_월세 = data_오피스텔_전월세['월세 (만원)', '월세']
data_오피스텔_전세 = data_오피스텔_전월세['보증금 (만원)', '전세']

#data_오피스텔_전월세['보증금 (만원)', '전세'] = data_오피스텔_전세
del(data_오피스텔_전월세['월세 (만원)', '전세'])
del(data_오피스텔_전월세['보증금 (만원)', '전세'])

round(data_오피스텔_전월세, 2)

# 그래프 한글깨짐 방지
matplotlib.font_manager.rebuild()
plt.rc("font", family="Malgun Gothic")

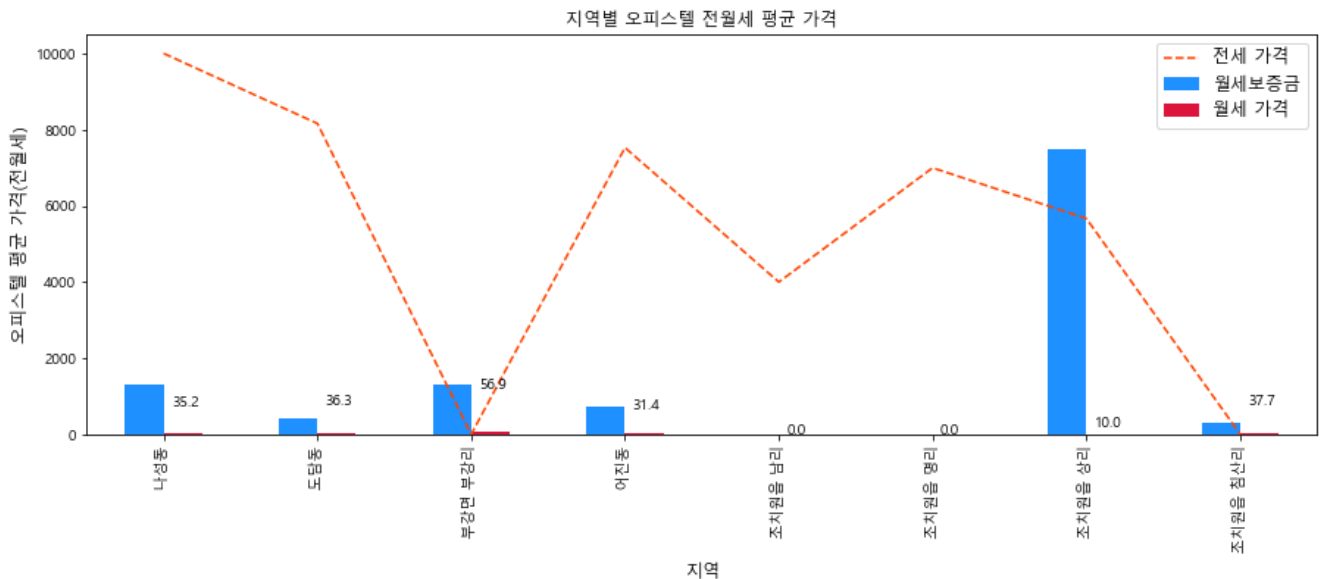
# 지역 별 평균 가격 그래프
ax = data_오피스텔_전월세.plot(kind='bar', title='지역별 오피스텔 전월세 평균 가격', figsize=(15, 5), 1
legend=True, fontsize=10, color=['dodgerblue', 'crimson'])
plt.plot(data_오피스텔_전세, linestyle='--', color='orangered')
ax.set_xlabel('지역', fontsize=12)
ax.set_ylabel('오피스텔 평균 가격(전월세)', fontsize=12)
ax.legend(['전세 가격', '월세보증금', '월세 가격'], fontsize=12)

for i, v in enumerate(data_오피스텔_전월세['월세 (만원)', '월세']):
    ax.text(i+0.05, v+v*20, str(round(v, 1)))

```

/opt/app-root/lib/python3.6/site-packages/ipykernel\_launcher.py:10: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
# Remove the CWD from sys.path while we load stuff.



In [ ]:

## geojson 자료 확인

In [11]:

```

연령별_거주인구 = gpd.read_file(input_path.joinpath('19.세종시_연령별_거주인구정보_격자.geojson'))
연령별_거주인구

```

Out[11]:

gid	201710-30 대-거주인 구주	201810-30 대-거주인 구주	201910-30 대-거주인 구주	202010-30 대-거주인 구주	201710-30 대-거주인 구주	201810-30 대-거주인 구주	201910-30 대-거주인 구주	202010-30 대-거주인 구주	201710-40 대-거주인 구주	...	202010-00 대-거주인 구주
0	다바 667567	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	다바 667568	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	다바 668566	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	다바 668567	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	다바 668568	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
47391	다바 919332	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
47392	다바 919333	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
47393	다바 919334	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
47394	다바 920332	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
47395	다바 920333	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

47396 rows × 38 columns



In [12]:

```
도로명주소 = gpd.read_file(input_path.joinpath('23.세종시_도로명주소_건물.geojson'))
도로명주소
```

Out[12]:

	BDTYP_CD	BULD_NM	BULD_NM_DC	BULD_SE_CD	BUL_MAN_NO	EMD_CD	GRO_FLO_CO	LNBR_MNNM	LNBR_SLNO	UNI
0	10299	None	None	0	55874	350	1	177	4	
1	10299	None	None	0	55795	340	1	42	23	
2	04401	None	None	0	55266	107	7	723	0	
3	04001	None	None	0	60761	107	9	722	0	
4	03999	None	None	0	55331	107	8	721	0	

BDTYP_CD	BULD_NM	BULD_NM_DC	BULD_SE_CD	BUL_MAN_NO	EMD_CD	GRO_FLO_CO	LNBR_MNNM	LNBR_SLNO	UNI
...	...	...	...	...	...	...	...	...	...
54952	01001	None	None	0	36537	340	1	135	0
54953	14001	None	None	0	40633	340	1	135	0
54954	01001	None	None	0	19892	340	1	135	0
54955	14001	None	None	0	19893	340	1	135	0
54956	01001	금남면 달 전리 119-1 단독주택	None	0	54174	340	1	119	1

54957 rows × 11 columns



In [13]:

```
건축물연면적 = gpd.read_file(input_path.joinpath('24.세종시_건축물연면적_격자.geojson'))
건축물연면적
```

Out[13]:

	gid	val	geometry
0	다바667567	NaN	MULTIPOLYGON (((127.12716 36.70773, 127.12716 ...
1	다바667568	NaN	MULTIPOLYGON (((127.12716 36.70863, 127.12716 ...
2	다바668566	NaN	MULTIPOLYGON (((127.12829 36.70683, 127.12828 ...
3	다바668567	NaN	MULTIPOLYGON (((127.12828 36.70773, 127.12828 ...
4	다바668568	NaN	MULTIPOLYGON (((127.12828 36.70863, 127.12828 ...
...	...	...	...
47391	다바919332	NaN	MULTIPOLYGON (((127.40956 36.49642, 127.40956 ...
47392	다바919333	NaN	MULTIPOLYGON (((127.40956 36.49732, 127.40956 ...
47393	다바919334	NaN	MULTIPOLYGON (((127.40956 36.49823, 127.40955 ...
47394	다바920332	NaN	MULTIPOLYGON (((127.41067 36.49642, 127.41067 ...
47395	다바920333	NaN	MULTIPOLYGON (((127.41067 36.49732, 127.41067 ...

47396 rows × 3 columns

In [14]:

```
법정경계_시군구 = gpd.read_file(input_path.joinpath('30.세종시_법정경계(시군구).geojson'))
법정경계_시군구
```

Out[14]:

	SIG_CD	SIG_KOR_NM	geometry
0	36110	세종특별자치시	MULTIPOLYGON (((127.17841 36.59687, 127.17839 ...

In [15]:

```
법정경계_읍면동 = gpd.read_file(input_path.joinpath('31.세종시_법정경계(읍면동).geojson'))
법정경계_읍면동
```

Out[15]:

	EMD_CD	EMD_KOR_NM	geometry
0	36110101	반곡동	MULTIPOLYGON (((127.30833 36.47977, 127.30818 ...
1	36110102	소담동	MULTIPOLYGON (((127.30480 36.48845, 127.30480 ...
2	36110103	보람동	MULTIPOLYGON (((127.29383 36.48508, 127.29381 ...
3	36110104	대평동	MULTIPOLYGON (((127.27215 36.46388, 127.27197 ...
4	36110105	가람동	MULTIPOLYGON (((127.24631 36.46300, 127.24626 ...
5	36110106	한솔동	MULTIPOLYGON (((127.25661 36.47033, 127.25653 ...
6	36110107	나성동	MULTIPOLYGON (((127.26603 36.49182, 127.26744 ...
7	36110108	새롬동	MULTIPOLYGON (((127.23978 36.48556, 127.23966 ...
8	36110109	다정동	MULTIPOLYGON (((127.24003 36.48809, 127.24004 ...
9	36110110	어진동	MULTIPOLYGON (((127.25544 36.49534, 127.25543 ...
10	36110111	종촌동	MULTIPOLYGON (((127.25357 36.50688, 127.25357 ...
11	36110112	고운동	MULTIPOLYGON (((127.23895 36.53520, 127.23960 ...
12	36110113	아름동	MULTIPOLYGON (((127.25238 36.51029, 127.25218 ...
13	36110114	도담동	MULTIPOLYGON (((127.25357 36.50688, 127.25348 ...
14	36110115	산울동	MULTIPOLYGON (((127.26859 36.53949, 127.26858 ...
15	36110116	해밀동	MULTIPOLYGON (((127.27076 36.52337, 127.27005 ...
16	36110117	합강동	MULTIPOLYGON (((127.34569 36.51897, 127.34628 ...
17	36110118	집현동	MULTIPOLYGON (((127.31732 36.48175, 127.31730 ...
18	36110250	조치원읍	MULTIPOLYGON (((127.30663 36.60062, 127.30662 ...
19	36110310	연기면	MULTIPOLYGON (((127.28524 36.55518, 127.28605 ...
20	36110320	연동면	MULTIPOLYGON (((127.32202 36.58302, 127.32209 ...
21	36110330	부강면	MULTIPOLYGON (((127.36891 36.48959, 127.36891 ...
22	36110340	금남면	MULTIPOLYGON (((127.36098 36.49088, 127.36059 ...
23	36110350	장군면	MULTIPOLYGON (((127.23952 36.48582, 127.23966 ...
24	36110360	연서면	MULTIPOLYGON (((127.19288 36.60110, 127.19298 ...
25	36110370	전의면	MULTIPOLYGON (((127.19609 36.60448, 127.19604 ...
26	36110380	전동면	MULTIPOLYGON (((127.27948 36.63350, 127.27948 ...
27	36110390	소정면	MULTIPOLYGON (((127.20791 36.71901, 127.20781 ...

In [16]:

```
행정경계_읍면동 = gpd.read_file(input_path.joinpath('32.세종시_행정경계(읍면동).geojson'))
행정경계_읍면동
```

Out[16]:

	ADM_DR_CD	ADM_DR_NM	geometry
0	2901011	조치원읍	MULTIPOLYGON (((127.29172 36.63614, 127.29203 ...
1	2901031	연기면	MULTIPOLYGON (((127.24092 36.46741, 127.24087 ...
2	2901032	연동면	MULTIPOLYGON (((127.35572 36.55450, 127.35568 ...
3	2901033	부강면	MULTIPOLYGON (((127.37706 36.56788, 127.37711 ...
4	2901034	금남면	MULTIPOLYGON (((127.36424 36.51353, 127.36375 ...
5	2901035	장군면	MULTIPOLYGON (((127.21187 36.57860, 127.21233 ...
6	2901036	연서면	MULTIPOLYGON (((127.21134 36.62035, 127.21169 ...
7	2901037	전의면	MULTIPOLYGON (((127.20998 36.71848, 127.21068 ...
8	2901038	전동면	MULTIPOLYGON (((127.25304 36.69594, 127.25410 ...
9	2901039	소정면	MULTIPOLYGON (((127.14980 36.72605, 127.14981 ...
10	2901053	도담동	MULTIPOLYGON (((127.26461 36.52069, 127.26459 ...
11	2901060	고운동	MULTIPOLYGON (((127.23853 36.53540, 127.23895 ...
12	2901056	종촌동	MULTIPOLYGON (((127.24233 36.50869, 127.24234 ...
13	2901065	소담동	MULTIPOLYGON (((127.31207 36.50321, 127.31209 ...
14	2901062	새롬동	MULTIPOLYGON (((127.26244 36.47868, 127.26244 ...
15	2901059	아름동	MULTIPOLYGON (((127.25182 36.52672, 127.25193 ...
16	2901061	한솔동	MULTIPOLYGON (((127.26244 36.47868, 127.26181 ...
17	2901064	대평동	MULTIPOLYGON (((127.27743 36.47628, 127.27743 ...
18	2901066	보람동	MULTIPOLYGON (((127.29381 36.48496, 127.29381 ...

In [17]:

```
지적도 = gpd.read_file(input_path.joinpath('33.세종시_지적도.geojson'))
지적도
```

Out[17]:

	PNU	JIBUN	geometry
0	3611037031101950001	195-1 잡	MULTIPOLYGON (((127.17929 36.65234, 127.17930 ...
1	3611034029102230001	223-1 답	MULTIPOLYGON (((127.30726 36.46622, 127.30716 ...
2	3611034028101540000	154 답	MULTIPOLYGON (((127.30510 36.45196, 127.30502 ...
3	3611034029200630000	산63 임	MULTIPOLYGON (((127.31121 36.47124, 127.31127 ...



10772	상권 업종 대분 류코드	학문/ 취업 대분 류명	R05 상권 중분 류코드	학원-음 식 상권 중분 류명	R05A02 상권 중소 분류 코드	노/바 이 상권 중분 류명	P85620 표준 업분 류코드	예술 학원 표준 산업 분류 명	3611054000 행정 동코드	중 행 정 동 명	충청 남도 3로 160, (충 청, 가재마을센 트럴빌딩주상 복합) 충청 남도 3로 160, (충 청, 가재마을센 트럴빌딩주상 복합)	127.243716	36.506156
												lon	lat

10773 rows × 13 columns

In [19]:

```
# 지역별 각 상점 개수와 합계
data_상권정보 = pd.crosstab(상권정보.행정동명, 상권정보.상권업종대분류명, margins=True)
data_상권정보
```

Out[19]:

상권업종대분류명	관광/여가/오락	부동산	생활서비스	소매	숙박	스포츠	음식	학문/교육	All
행정동명									
고운동	7	25	74	61	0	3	136	47	353
금남면	13	98	102	184	2	1	284	15	699
대평동	2	2	32	25	0	1	38	11	111
도담동	13	51	83	217	0	3	388	48	803
보람동	5	19	69	61	0	5	230	31	420
부강면	11	14	69	135	8	0	189	10	436
새롬동	21	157	159	313	0	4	349	115	1118
소담동	0	11	44	38	0	0	81	38	212
소정면	1	4	22	43	1	0	87	7	165
아름동	3	13	38	45	0	6	115	57	277
연기면	1	28	37	56	3	0	67	2	194
연동면	0	15	14	46	3	0	71	7	156
연서면	5	40	50	162	7	0	164	16	444
장군면	10	78	40	128	3	1	203	2	465
전동면	2	5	22	53	6	0	51	4	143
전의면	7	9	61	139	5	0	182	14	417
조치원읍	97	105	574	1133	36	3	1402	203	3553
종촌동	1	20	48	87	0	3	131	56	346
한솔동	1	89	49	77	0	0	174	71	461
All	200	783	1587	3003	74	30	4342	754	10773

In [20]:

```
# 지역별 상권 수
data_상권정보_all = data_상권정보['All']
data_상권정보_all
```

Out[20]:

행정동명	
고운동	353
금남면	699
대평동	111
도담동	803
보람동	420
부강면	436
새롬동	1118
소담동	212
소정면	165
아름동	277
연기면	194
연동면	156
연서면	444
장군면	465
전동면	143
전의면	417

```
조치원읍      3553
종촌동        346
한솔동        461
All           10773
Name: All, dtype: int64
```

In [21]:

```
# 지역별 위도와 경도
행정경계_읍면동 = gpd.read_file(input_path.joinpath('32.세종시_행정경계(읍면동).geojson'))
행정경계_읍면동

행정경계_읍면동['lon'] = (행정경계_읍면동['geometry'].bounds['maxx'] + 행정경계_읍면동['geometry'].bounds['minx'])/2
행정경계_읍면동['lat'] = (행정경계_읍면동['geometry'].bounds['maxy'] + 행정경계_읍면동['geometry'].bounds['miny'])/2
data_상권정보_lon_lat = 행정경계_읍면동[['ADM_DR_NM', 'lon', 'lat']]
data_상권정보_lon_lat=data_상권정보_lon_lat.rename(columns={'ADM_DR_NM':'행정동명'})
data_상권정보_lon_lat
```

Out[21]:

	행정동명	lon	lat
0	조치원읍	127.282863	36.607199
1	연기면	127.279812	36.507771
2	연동면	127.330061	36.550175
3	부강면	127.381030	36.528753
4	금남면	127.290863	36.462920
5	장군면	127.207249	36.507231
6	연서면	127.247954	36.583444
7	전의면	127.199439	36.655099
8	전동면	127.253810	36.656332
9	소정면	127.167851	36.711410
10	도담동	127.265374	36.508349
11	고운동	127.236059	36.517374
12	종촌동	127.247568	36.503792
13	소담동	127.307823	36.490650
14	새롬동	127.252152	36.488832
15	아름동	127.251252	36.517544
16	한솔동	127.249586	36.473352
17	대평동	127.274502	36.470178
18	보람동	127.288813	36.478255

In [22]:

```
# 지역별 위도와 경도, 총 상점 수 합치기
data_상권정보_info = pd.merge( data_상권정보_all,data_상권정보_lon_lat, on ="행정동명")
data_상권정보_info
```

Out[22]:

	행정동명	All	lon	lat
0	고운동	353	127.236059	36.517374
1	금남면	699	127.290863	36.462920
2	대평동	111	127.274502	36.470178
3	도담동	803	127.265374	36.508349
4	보람동	420	127.288813	36.478255
5	부강면	436	127.381030	36.528753



6	새롬동	1118	127.252152	36.488832
7	소담동	212	127.307823	36.490650
8	소정면	165	127.167851	36.711410
9	아름동	277	127.251252	36.517544
10	연기면	194	127.279812	36.507771
11	연동면	156	127.330061	36.550175
12	연서면	444	127.247954	36.583444
13	장군면	465	127.207249	36.507231
14	전동면	143	127.253810	36.656332
15	전의면	417	127.199439	36.655099
16	조치원읍	3553	127.282863	36.607199
17	종촌동	346	127.247568	36.503792
18	한솔동	461	127.249586	36.473352

In [23]:

```
# 지역별 중심위도와 중심경도, 총 상권 수 계산
data_상권정보_info = pd.merge( data_상권정보['All'], data_상권정보_lon_lat, on ="행정동명")
data_상권정보_info
```

Out[23]:

	행정동명	All	lon	lat
0	고운동	353	127.236059	36.517374
1	금남면	699	127.290863	36.462920
2	대평동	111	127.274502	36.470178
3	도담동	803	127.265374	36.508349
4	보람동	420	127.288813	36.478255
5	부강면	436	127.381030	36.528753
6	새롬동	1118	127.252152	36.488832
7	소담동	212	127.307823	36.490650
8	소정면	165	127.167851	36.711410
9	아름동	277	127.251252	36.517544
10	연기면	194	127.279812	36.507771
11	연동면	156	127.330061	36.550175
12	연서면	444	127.247954	36.583444
13	장군면	465	127.207249	36.507231
14	전동면	143	127.253810	36.656332
15	전의면	417	127.199439	36.655099
16	조치원읍	3553	127.282863	36.607199
17	종촌동	346	127.247568	36.503792
18	한솔동	461	127.249586	36.473352

In [24]:

```
# 상권업종대분류명 사전순 정렬
data_상권정보 = 상권정보
data_상권정보 = data_상권정보.sort_values(by='상권업종대분류명')
data_상권정보

data_상권정보 = data_상권정보[['상권업종대분류명', '도로명주소', 'lon', 'lat']]

# 상권업종대분류명을 상권종류로 이름 변경
data_상권정보.rename(columns={'상권업종대분류명':'상권종류'}, inplace = True)
data_상권정보
```

Out[24]:

Out [24]:

	상권종류	도로명주소	lon	lat
6091	관광/여가/오락	세종특별자치시 만남로 16, (고운동)	127.236487	36.501126
153	관광/여가/오락	세종특별자치시 장군면 장척로 565, (봉안리)	127.220519	36.499410
9564	관광/여가/오락	세종특별자치시 고운서길 19, (고운동)	127.233984	36.505851
2597	관광/여가/오락	세종특별자치시 조치원읍 으뜸길 233, (원리)	127.296533	36.601936
3054	관광/여가/오락	세종특별자치시 장군면 장척로 385-3, (도계리)	127.203729	36.496271
...	...	...	...	...
8580	학문/교육	세종특별자치시 한누리대로 157, (나성동)	127.262479	36.479318
1639	학문/교육	세종특별자치시 조치원읍 대첩로 17, (신흥리)	127.294190	36.595325
1642	학문/교육	세종특별자치시 조치원읍 문화2길 64, (침산리)	127.289099	36.602117
8606	학문/교육	세종특별자치시 새롬중앙로 62-15, (새롬동)	127.251201	36.487016
10772	학문/교육	세종특별자치시 도움3로 160, (종촌동, 가재마을센트레빌4단지)	127.243716	36.506156

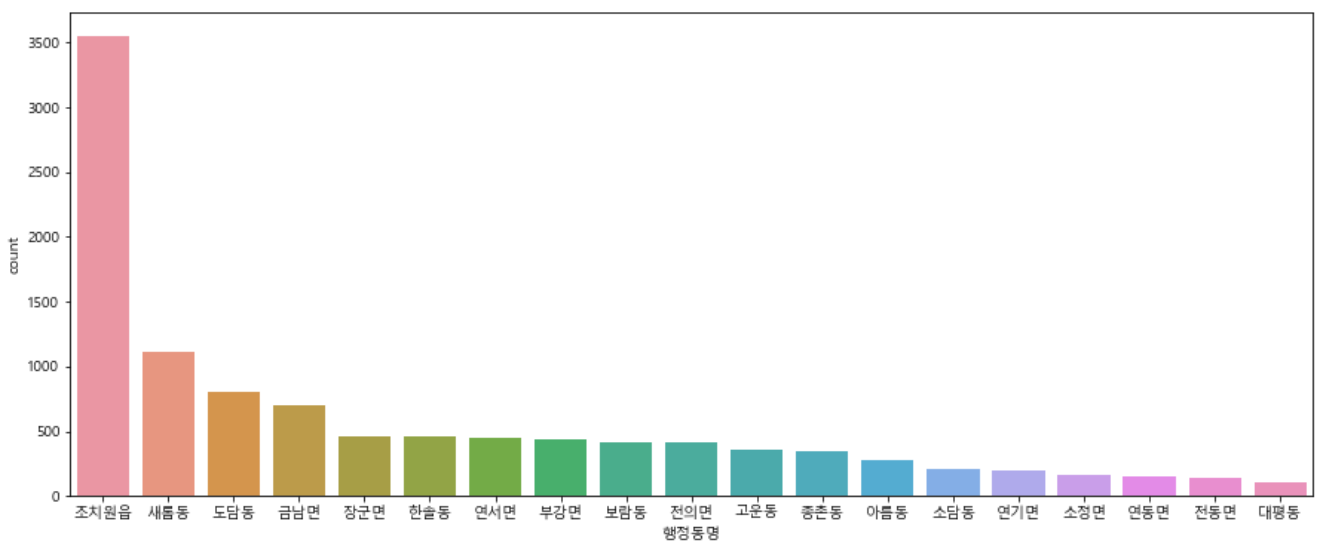
10773 rows × 4 columns

In [25]:

```
plt.figure(figsize= (15,6))
sns.countplot(data = 상권정보, x="행정동명", order=상권정보['행정동명'].value_counts().index)
```

Out [25]:

<AxesSubplot:xlabel='행정동명', ylabel='count'>



In [ ]:

In [ ]:

In [ ]:

## 상권 수 지도에 표시

In [26]:

```
# 위도, 경도, 총 상권 개수를 float 형으로 바꿈
data_상권정보_info['lon'] = data_상권정보_info.lon.astype(float)
data_상권정보_info['lat'] = data_상권정보_info.lat.astype(float)
data_상권정보_info['All'] = data_상권정보_info.All.astype(float)
data_상권정보_info

# 상권 수 시각화
상권수_map = folium.Map(location = [data_상권정보_info['lat'].mean(), data_상권정보_info['lon'].mean()], zoom_start = 11)

for item in data_상권정보_info.index:
    latitude = data_상권정보_info.loc[item, 'lat']
    longitude = data_상권정보_info.loc[item, 'lon']
    popups = folium.Popup(data_상권정보_info.loc[item, '행정동명'], max_width=100)
    folium.CircleMarker([latitude, longitude],
                        radius = data_상권정보_info.loc[item, 'All']/100,
                        popup = popups,
                        color = 'red',
                        fill = True).add_to(상권수_map)

상권수_map
```

Out [26]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [187]:

```
data_상권정보_info
```

Out [187]:

	행정동명	All	lon	lat
0	고운동	353.0	127.236059	36.517374
1	금남면	699.0	127.290863	36.462920
2	대평동	111.0	127.274502	36.470178
3	도담동	803.0	127.265374	36.508349
4	보람동	420.0	127.288813	36.478255
5	부강면	436.0	127.381030	36.528753
6	새롬동	1118.0	127.252152	36.488832
7	소담동	212.0	127.307823	36.490650
8	소정면	165.0	127.167851	36.711410

9	행정동명	277.0	127.251252	36.517544
10	연기면	194.0	127.279812	36.507771
11	연동면	156.0	127.330061	36.550175
12	연서면	444.0	127.247954	36.583444
13	장군면	465.0	127.207249	36.507231
14	전동면	143.0	127.253810	36.656332
15	전의면	417.0	127.199439	36.655099
16	조치원읍	3553.0	127.282863	36.607199
17	종촌동	346.0	127.247568	36.503792
18	한솔동	461.0	127.249586	36.473352

## 상권 수 지도에 표시2

In [27]:

```
# 자료 불러오기
행정경계_읍면동 = gpd.read_file(input_path.joinpath('32.세종시_행정경계(읍면동).geojson'))
행정경계_읍면동

data = data_상권정보_info[['행정동명', 'All']]
data

# 지도
상권수_map2 = folium.Map(location = [data_상권정보_info['lat'].mean(), data_상권정보_info['lon'].mean()], zoom_start = 11)

상권수_map2.choropleth(
    geo_data = 행정경계_읍면동,
    data = data,
    columns=['행정동명', 'All'],
    key_on = 'feature.properties.ADM_DR_NM',
    fill_color = 'BuPu',
    legend_name = '상권 수',
)

상권수_map2
```

/opt/app-root/lib/python3.6/site-packages/folium/folium.py:415: FutureWarning: The choropleth method has been deprecated. Instead use the new Choropleth class, which has the same arguments. See the example notebook 'GeoJSON\_and\_choropleth' for how to do this.  
FutureWarning

Out [27]:

Make this Notebook Trusted to load map: File -> Trust Notebook

```
In [ ]:
```

```
In [ ]:
```

## 상권 정보 지도에 표시

```
In [28]:
```

```
data_상권정보 = 상권정보
data_상권정보 = data_상권정보[['상권업종대분류명', '도로명주소', 'lon', 'lat']]

# 상권업종대분류명을 상권종류로 이름 변경
data_상권정보.rename(columns={'상권업종대분류명': '상권종류'}, inplace = True)
data_상권정보

# 위도, 경도 값 형 변환
data_상권정보['lon'] = data_상권정보.lon.astype(float)
data_상권정보['lat'] = data_상권정보.lat.astype(float)

# 지도
상권종류_map = folium.Map(location = [data_상권정보['lat'].mean(), data_상권정보['lon'].mean()], zoom_start = 11)

for item in data_상권정보.index:
    latitude = data_상권정보.loc[item, 'lat']
    longitude = data_상권정보.loc[item, 'lon']

    if data_상권정보.loc[item, '상권종류'] == '관광/여가/오락':
        colors = 'dodgerblue'

    elif data_상권정보.loc[item, '상권종류'] == '부동산':
        colors = 'burlywood'

    elif data_상권정보.loc[item, '상권종류'] == '생활서비스':
        colors = 'gold'

    elif data_상권정보.loc[item, '상권종류'] == '소매':
        colors = 'darkolivegreen'

    elif data_상권정보.loc[item, '상권종류'] == '학문/교육':
        colors = 'slategrey'

    elif data_상권정보.loc[item, '상권종류'] == '음식':
        colors = 'salmon'

    elif data_상권정보.loc[item, '상권종류'] == '스포츠':
        colors = 'blueviolet'

    elif data_상권정보.loc[item, '상권종류'] == '숙박':
        colors = 'lightpink'
    popups = folium.Popup(data_상권정보.loc[item, '도로명주소'], max_width=150)
    folium.CircleMarker([latitude, longitude],
                        popup = popups,
                        color = colors,
                        fill = True).add_to(상권종류_map)

상권종류_map
```

```
/opt/app-root/lib/python3.6/site-packages/pandas/core/frame.py:4308: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
errors=errors,
/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
# Remove the CWD from sys.path while we load stuff.
/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
# This is added back by InteractiveShellApp.init_path()
```

Out[28]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:

## 아파트 시세 지도

In [206]:

```
아파트_매매

# 법정경계에 맞게 시군구 편집

data_아파트 = 아파트_매매[['시군구', '거래금액(만원)']]
for i in range(len(data_아파트['시군구'])):
    if len(data_아파트.loc[i, '시군구']) == 9:
        data_아파트['시군구'][i] = data_아파트['시군구'][i][:5]
    elif len(data_아파트.loc[i, '시군구']) == 8:

        if data_아파트.loc[i, '시군구'][1] == '조':
            data_아파트['시군구'][i] = data_아파트['시군구'][i][:5]
        else:
            data_아파트['시군구'][i] = data_아파트['시군구'][i][:5]
```

data\_아파트

```
/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
/opt/app-root/lib/python3.6/site-packages/IPython/core/interactiveshell.py:3343:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
exec(code\_obj, self.user\_global\_ns, self.user\_ns)

```
/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
/opt/app-root/lib/python3.6/site-packages/ipykernel_launcher.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
if sys.path[0] == '':

Out[206]:

시군구 거래금액(만원)		
0	보람동	8600
1	종촌동	27300
2	고운동	32000
3	고운동	30500
4	아름동	31300
...	...	...
21135	조치원읍	50000
21136	조치원읍	23500
21137	도담동	68000
21138	부강면	6500
21139	조치원읍	37000

21140 rows × 2 columns

In [207]:

```
# 시군구별 거래금액(만원) 평균
data_아파트 = data_아파트.groupby(data_아파트.시군구)['거래금액(만원)'].mean()
data_아파트 = data_아파트.reset_index()
data_아파트
```

Out[207]:

시군구 거래금액(만원)		
0	고운동	33226.134474
1	금남면	13544.158965
2	나성동	10566.632948
3	다정동	58885.662667
4	대평동	66912.200820
5	도담동	45208.741341
6	바곡동	50896.721893

	시군구	거래금액(만원)
7	신말동	59299.212299
8	부강면	7824.732143
9	새롬동	59670.444348
10	소담동	53189.771694
11	소정면	6329.310345
12	아름동	34519.918834
13	어진동	39438.960880
14	연동면	6573.083333
15	연서면	11860.162602
16	장군면	2718.142857
17	전동면	5652.380952
18	전의면	7884.500000
19	조치원읍	16883.180214
20	종촌동	38950.013474
21	한솔동	40368.280587

In [208]:

```
법정경계_시군구 = gpd.read_file(input_path.joinpath('31.세종시_법정경계(읍면동).geojson'))
법정경계_시군구
```

Out [208]:

	EMD_CD	EMD_KOR_NM	geometry
0	36110101	반곡동	MULTIPOLYGON (((127.30833 36.47977, 127.30818 ...
1	36110102	소담동	MULTIPOLYGON (((127.30480 36.48845, 127.30480 ...
2	36110103	보람동	MULTIPOLYGON (((127.29383 36.48508, 127.29381 ...
3	36110104	대평동	MULTIPOLYGON (((127.27215 36.46388, 127.27197 ...
4	36110105	가람동	MULTIPOLYGON (((127.24631 36.46300, 127.24626 ...
5	36110106	한솔동	MULTIPOLYGON (((127.25661 36.47033, 127.25653 ...
6	36110107	나성동	MULTIPOLYGON (((127.26603 36.49182, 127.26744 ...
7	36110108	새롬동	MULTIPOLYGON (((127.23978 36.48556, 127.23966 ...
8	36110109	다정동	MULTIPOLYGON (((127.24003 36.48809, 127.24004 ...
9	36110110	어진동	MULTIPOLYGON (((127.25544 36.49534, 127.25543 ...
10	36110111	종촌동	MULTIPOLYGON (((127.25357 36.50688, 127.25357 ...
11	36110112	고운동	MULTIPOLYGON (((127.23895 36.53520, 127.23960 ...
12	36110113	아름동	MULTIPOLYGON (((127.25238 36.51029, 127.25218 ...
13	36110114	도담동	MULTIPOLYGON (((127.25357 36.50688, 127.25348 ...
14	36110115	산울동	MULTIPOLYGON (((127.26859 36.53949, 127.26858 ...
15	36110116	해밀동	MULTIPOLYGON (((127.27076 36.52337, 127.27005 ...
16	36110117	합강동	MULTIPOLYGON (((127.34569 36.51897, 127.34628 ...
17	36110118	집현동	MULTIPOLYGON (((127.31732 36.48175, 127.31730 ...



	EMD_CD	EMD_KOR_NM	geometry
18	36110250	조치원읍	MULTIPOLYGON (((127.30663 36.60062, 127.30662 ...
19	36110310	연기면	MULTIPOLYGON (((127.28524 36.55518, 127.28605 ...
20	36110320	연동면	MULTIPOLYGON (((127.32202 36.58302, 127.32209 ...
21	36110330	부강면	MULTIPOLYGON (((127.36891 36.48959, 127.36891 ...
22	36110340	금남면	MULTIPOLYGON (((127.36098 36.49088, 127.36059 ...
23	36110350	장군면	MULTIPOLYGON (((127.23952 36.48582, 127.23966 ...
24	36110360	연서면	MULTIPOLYGON (((127.19288 36.60110, 127.19298 ...
25	36110370	전의면	MULTIPOLYGON (((127.19609 36.60448, 127.19604 ...
26	36110380	전동면	MULTIPOLYGON (((127.27948 36.63350, 127.27948 ...
27	36110390	소정면	MULTIPOLYGON (((127.20791 36.71901, 127.20781 ...

In [209]:

```
data_아파트['lon'] = (법정경계_시군구['geometry'].bounds['maxx'] + 법정경계_시군구['geometry'].bounds['min
x']) / 2
data_아파트['lat'] = (법정경계_시군구['geometry'].bounds['maxy'] + 법정경계_시군구['geometry'].bounds['mi
ny']) / 2
data_아파트
```

Out[209]:

	시군구	거래금액(만원)	lon	lat
0	고운동	33226.134474	127.310382	36.491414
1	금남면	13544.158965	127.301194	36.485012
2	나성동	10566.632948	127.288813	36.478255
3	다정동	58885.662667	127.274502	36.470178
4	대평동	66912.200820	127.246671	36.471256
5	도담동	45208.741341	127.254392	36.477017
6	반곡동	50896.721893	127.264276	36.486736
7	보람동	59299.212299	127.249013	36.484422
8	부강면	7824.732143	127.243697	36.493832
9	새롭동	59670.444348	127.265374	36.500819
10	소담동	53189.771694	127.247569	36.503792
11	소정면	6329.310345	127.236093	36.517375
12	아름동	34519.918834	127.251275	36.517544
13	어진동	39438.960880	127.262456	36.515875
14	연동면	6573.083333	127.253918	36.534591
15	연서면	11860.162602	127.271144	36.527016
16	장군면	2718.142857	127.333018	36.525097
17	전동면	5652.380952	127.329770	36.496421
18	전의면	7884.500000	127.282863	36.607327
19	조치원읍	16883.180214	127.279813	36.507784
20	종촌동	38950.013474	127.330061	36.550718
21	한솔동	40368.280587	127.381030	36.528753

In [210]:

```
data_아파트.columns
```

Out [210]:

```
Index(['시군구', '거래금액 (만원)', 'lon', 'lat'], dtype='object')
```

In [213]:

```
data = data_아파트[['시군구', '거래금액 (만원)']]

# 지도
아파트시세_map = folium.Map(location = [data_아파트['lat'].mean(), data_아파트['lon'].mean()], zoom_start = 11)

아파트시세_map.choropleth(
    geo_data = 법정경계_시군구,
    data = data,

    columns=['시군구', '거래금액 (만원)'],
    key_on = 'feature.properties.EMD_KOR_NM',
    fill_color = 'BuPu',
    legend_name = '아파트시세',
)

아파트시세_map
```

Out [213]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: