

6장_19대 대선결과분석

발표자: 최경희

greenwave35@gmail.com

2018.06.13

스터디 참조 교재: 파이썬으로 데이터 주무르기, 민형기, BJPUBLIC

19대 대선결과 분석 : 작업 순서

데이터 획득 준비작업

(Selenium, BeautifulSoup 이용)



데이터 획득



데이터 정리



시각화

주요개념 **Summary**

install 라이브러리 목록

| | |
|--|---|
| pip install googlemaps | geocoding API 키 가져오기 : 자신의 키값으로 import |
| pip install seaborn | seaborn을 import 해 사용할 땐, matplotlib도 같이 import 해 사용해야 함 |
| pip install folium | 지도 시각화 |
| conda install -c conda-forge tqdm | 진행바 |
| pip install selenium | Selenium 설치후 브라우저에 맞춰 웹드라이버 다운로드 해야함. Chrome Driver 검색해 다운로드 |

그외 : Folium으로 표현 어려운 것 해결. 대한민국 지도 행정구역 경계선 그려주는 json파일 구하기/
LucyPark 의 github 에 접속하여 얻기. <https://goo.gl/xi5pKD>

파이썬 크롤링 시, 주로 import 하는 것들

```
from selenium import webdriver #webdriver사용 (가상크롬 설정)
```

```
from bs4 import BeautifulSoup #웹문자 추출 beautifulsoup
```

```
import urllib.request #url가져오기
```

```
import re #정규식
```

```
import time
```

```
import numpy as np #다차원 배열처리, 과학계산 등
```

```
import csv #csv파일 저장용
```

개념

정보제공

platform

- 파이썬의 플랫폼 모듈은 기본 플랫폼의 데이터에 액세스하는 데 사용
- 하드웨어, 운영 체제 및 인터프리터 버전 정보와 같은 정보를 제공
- 플랫폼의 하드웨어를 확인하고 시스템 및 인터프리터 버전 정보를 제공

예)

platform.system ()

'Linux', 'Windows' 또는 'Java'와 같은 시스템 / OS 이름을 반환

개념

시각화

Seaborn

- matplotlib을 기반으로하는 **Python 시각화 라이브러리**
- 매력적인 통계 그래픽을 그리기위한 고급 인터페이스를 제공
- <http://seaborn.pydata.org/>
- <http://seaborn.pydata.org/tutorial/aesthetics.html> 시각화 스타일 종류

설치)

pip install seaborn

개념

시각화

Folium

- 대화 형 리플릿 맵에서 Python으로 조작 된 데이터를 쉽게 시각화 가능
- Python으로 데이터를 조작 한 다음 Folium을 통해 리플릿 맵에서 시각화
- <http://folium.readthedocs.io/en/latest/>

설치)

pip install folium

Selenium 이란

-여러 플랫폼에서 웹 브라우저를 자동화 하는 도구 세트 www.seleniumhq.org/

-접근주소가 없으면 BeautifulSoup에서는 처리불가능. 그래서 사용하는 것이 Selenium.
save_screenshot 명령으로 화면을 캡처 할 수 있음

-동적으로 구성된 페이지를 크롤링 하거나 사이트트 캡처 등을 할때 자주 사용

-아나콘다에 포함된 모듈이 아니므로 별도 설치해야 함
pip install selenium

브라우저에 맞춰 드라이버를 다운로드 받아야 함.

예) Chrome Driver 검색해 다운로드,
경로는 source_code폴더 위치한 경로에 폴더 만들어 다운로드 후 설치

```
from selenium import webdriver
```

Beautiful Soup란

Beautiful Soup transforms a complex HTML document into a complex tree of Python objects.

Beautiful Soup Document 사이트

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

- HTML document안에 있는 **HTML 태그들을 Python 객체 형태로 만들어 줌**
- 모든 HTML 태그들을 객체화 했으므로 쉽게 호출 가능

예) `dom.html`
`dom.html.head`

개념

파싱

Re

정규식은 특수 문자와 일반 문자를 모두 포함 할 수 있습니다.

<https://docs.python.org/2/library/re.html#regular-expression-syntax>

정규식 (또는 RE)은 일치하는 문자열 세트를 지정합니다.

이 모듈의 함수는 특정 문자열이 주어진 정규 표현식과 일치하는지 또는 주어진 정규 표현식이 동일한 문자열에 일치하는지 확인할 수 있습니다.

'\''또는 같은 일부 문자 '('는 특별합니다.

특수 문자는 일반 문자의 클래스를 나타내거나 그 주변의 정규 표현식이 해석되는 방식에 영향을 줍니다.

데이터 획득 준비작업

데이터 획득 준비작업

준비

획득

정리

시각화

import

```
import pandas as pd
import numpy as np

import platform
import matplotlib.pyplot as plt

%matplotlib inline
```

그래프의 한글
폰트
문제해결

```
path = "c:/Windows/Fonts/malgun.ttf"
from matplotlib import font_manager, rc
if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
elif platform.system() == 'Windows':
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system... sorry~~~~~')

plt.rcParams['axes.unicode_minus'] = False
```

selenium의
webdriver
import

```
from selenium import webdriver
import time
```

데이터 획득 준비작업

준비

획득

정리

시각화

driver.get으로
사이트 호출

```
driver = webdriver.Chrome('../driver/chromedriver')  
driver.get("http://info.nec.go.kr/main/showDocument.xhtml?electionId=0000000000&topMenuId=VC&secondMenuId=VCCP09")
```

<명령>

크롬브라우저 열이라.

“ ” 사이트를 가져와라.



<실행>

작업표시줄에 크롬 아이콘이 새로 생김(브라우저 나타남).

열어보면, 해당 사이트 가져옴.

데이터 획득 준비작업

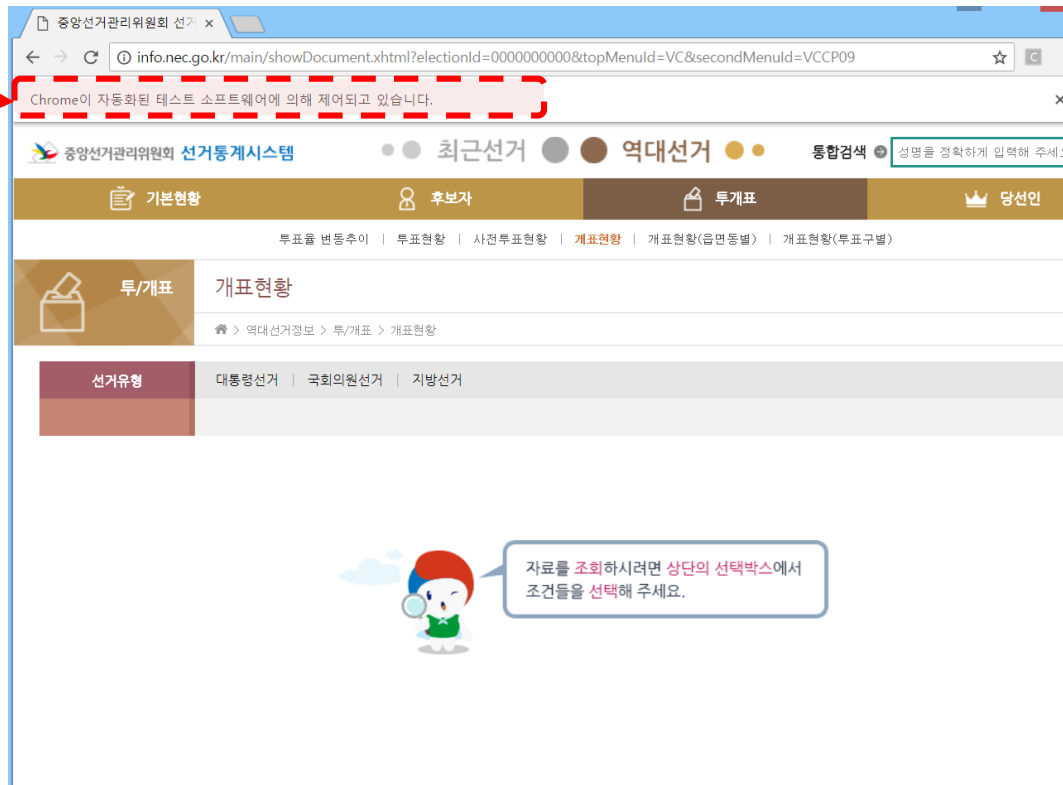
준비

획득

정리

시각화

Selenium의
driver.get으로 호출된
사이트라 표시 됨



데이터 획득 준비작업

준비

획득

정리

시각화

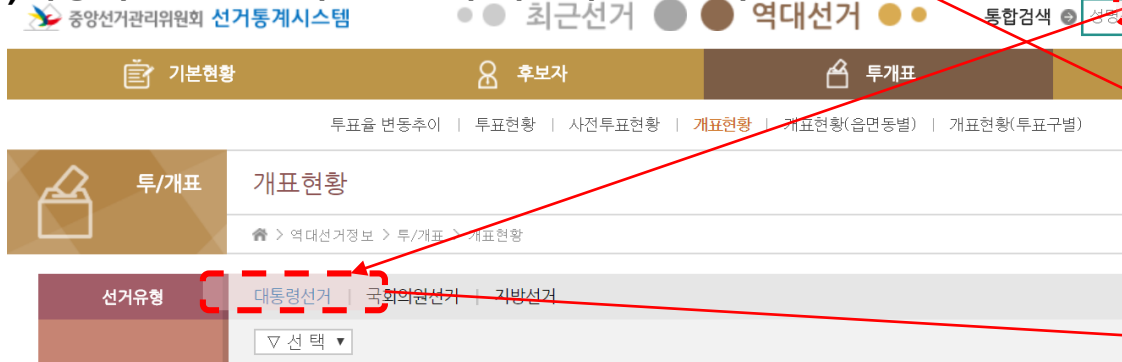
불러온 사이트에서
대통령선거 클릭해라!

대통령선거라는 글자부분을 클릭합니다. 해당 글자에서 앞 장에서 설명한 크롬 개발자 도구로 코드를 확인하면 id가 electionType1으로 나타납니다.

```
driver.find_element_by_id("electionType1").click()
```

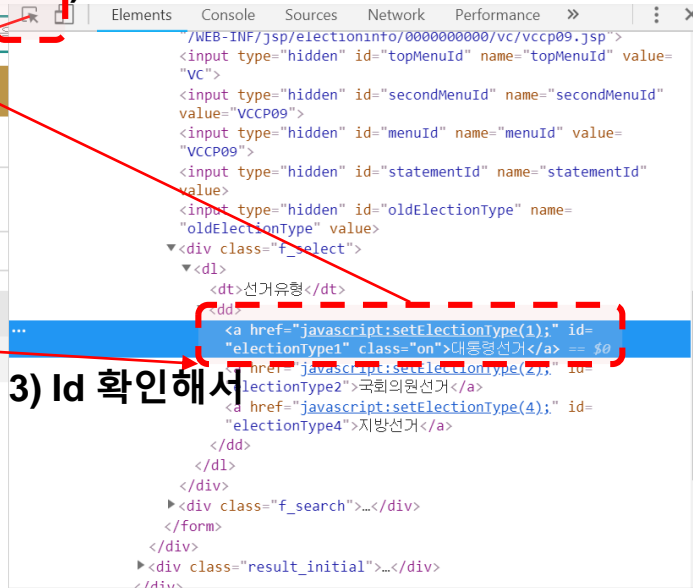
4) id로 요소를 찾아서 클릭

1) 해당사이트 열어서 F12눌러 개발자모드에서



3) 대통령선거 클릭하면

2) 모양 누른 후



3) Id 확인해서



자료를 조회하시려면 상단의 선택박스에서
조건들을 선택해 주세요.

데이터 획득 준비작업

준비

획득


정리

시각화

‘제19대’를 선택하라!

그리고 나타나는 선택항목에서 제19대를 선택하도록 합니다. 역시 크롬 개발자 도구에서 해당 위치의 id는 electionName이라는 것을 찾을 수 있습니다.

```
driver.find_element_by_id("electionName").send_keys("제19대")
```

 투/개표

개표현황

🏠 > 역대선거정보 > 투/개표 > 개표현황

선거유형

대통령선거 | 국회의원선거 | 지방선거

제19대 ▼ 선거 ▾ 선택 ▾

```
Elements Console Sources Network Performance >> ⋮ ×
<a href="javascript:setElectionType(4);" id=
  "electionType4">지방선거</a>
</dd>
</dl>
</div>
<div class="f_search">
  <dl>
    <dt></dt>
    <dd>
      <span id="wub_select" style>
        <div class="wub_select1">
          <select name="electionType" id="electionType" style=
            "display:none;" >...</select>
          <span id="spanElectionName" style="display: inline;
            ">
            <select name="electionName" id="electionName">...
              </select> == $0
            </span>
            <span id="spanElectionCode" style>...</span>
            <span id="spanCityCode" style="display: none;">...
              </span>
            <span id="spanTownCode" style="display:none">...
              </span>
            <span id="spanSggCityCode" style="display:none">...
              </span>
            <span id="spanSubmit" style="display:none;">...</span>
          </div>
        </dd>
      </dl>
    </div>
  </form>
</div>
```

데이터 획득 준비작업

준비

획득

정리

시각화

"대통령선거"를 선택
하라!

그리고 나타나는 선택에서 대통령선거를 선택하도록 합니다.

```
driver.findElementById("electionCode").sendKeys("대통령선거")
```



아이디 electionType

아이디로 찾음
electionName

아이디로 찾음
electionCode

Xpath로 찾아 데이터 가져오기
//*[@id="cityCode"]

데이터 획득 준비작업

준비

획득

정리

시각화

큰 따옴표 세개

“””

여러줄의 문자열을 출력할 때 이스케이프코드 \n 사용하지 않아도 여러줄로 보여줌.

XPath 찾기.

이제 나타나는 시도 항목에서 선택 부분의 XPath를 찾고, 해당 리스트를 가져옵니다

```
sido_list_raw = driver.find_element_by_xpath("//*[ @id="cityCode"]")
```

XPath 찾는 방법!

중영선거관리위원회 선거통계시스템

최근선거 ● 역대선거 ●

통합검색 ● 성명

기본현황 후보자 투개표

투표율 변동추이 | 투표현황 | 사전투표현황 | **개표현황** | 개표현황(읍면동별) | 개표현황(투표구별)

투/개표 개표현황

홈 > 역대선거정보 > 투/개표 > 개표현황

선거유형 대통령선거 | 국회의원선거 | 지방선거

select#cityCode | 121.33x71

제19대 선거 대통령선거 시도 **선택**

자료를 조회하시려면 상단의 선택박스에서 조건들을 선택해 주세요.

//*[@id="cityCode"]

Elements Console Sources Network Performance

Copy

Copy XPath

데이터 획득 준비작업

준비

획득

정리

시각화

XPath 찾아,
해당 리스트 가져오기

이제 나타나는 시도 항목에서 선택 부분의 XPath를 찾고, 해당 리스트를 가져옵니다

```
sido_list_raw = driver.find_element_by_xpath("//*[id='cityCode']")
sido_list = sido_list_raw.find_elements_by_tag_name("option")
sido_names_values = [option.text for option in sido_list]
sido_names_values = sido_names_values[2:]
sido_names_values
```

sido_list_raw에 크롬드라이브 해당 경로를 찾아 넣어라.

sido_list에 sido_list_raw에서 "option"태그를 찾아 넣어라.

sido_names_values에 sido_list안의 option 태그가 반복되는 동안 option태그의 text 값을 가져와 넣어라.

sido_names_values 리스트 항목을
2항목부터 끝까지 슬라이스 하여 넣어라.

sido_names_values를 실행.

주피터노트북
output
:데이터 가져와 보
여줌.

크롬브라우저에서
화면상에서는 더
이상
작동안함.

```
['서울특별시',  
'부산광역시',  
'대구광역시',  
'인천광역시',  
'광주광역시',  
'대전광역시',  
'울산광역시',  
'세종특별자치시',  
'경기도',  
'강원도',  
'충청북도',  
'충청남도',  
'전라북도',  
'전라남도',  
'경상북도',  
'경상남도',  
'제주특별자치도']
```

```
<span id="spanCityCode" style=>  
  <label id="cityCodeLabel" class="label" for=  
    "cityCode">시도</label>  
  <select id="cityCode" name="cityCode"> == $0  
    <option class="o_01" value="-1" selected=  
      "selected">> 선택</option>  
    <option value="0">> 전 제</option>  
    <option value="1100">서울특별시</option>  
    <option value="2600">부산광역시</option>  
    <option value="2700">대구광역시</option>  
    <option value="2800">인천광역시</option>  
    <option value="2900">광주광역시</option>  
    <option value="3000">대전광역시</option>  
    <option value="3100">울산광역시</option>  
    <option value="5100">세종특별자치시</option>  
    <option value="4100">경기도</option>  
    <option value="4200">강원도</option>  
    <option value="4300">충청북도</option>  
    <option value="4400">충청남도</option>  
    <option value="4500">전라북도</option>  
    <option value="4600">전라남도</option>  
    <option value="4700">경상북도</option>  
    <option value="4800">경상남도</option>  
    <option value="4900">제주특별자치도</option>  
  </select>  
</span>
```

데이터 획득

준비

획득

정리

시각화

import re

19대 대선 개표 결과 데이터 획득하기

결과보기

[제19대] [대통령선거] [서울특별시]

| 구시군명 | 선거인수 | 투표수 | 후보자별 득표수(득표율) | | | | | | | |
|------|------------------------|------------------------|----------------------|----------------------|----------------------|-------------------|-------------------|-----------------|---------------|-------------------|
| | | | 더불어민주 당 문재인 | 자유한국당 홍준표 | 국민의당 안철수 | 바른정당 유승민 | 정의당 심상정 | 새누리당 조원진 | 경제야국당 오영국 | 국민대통합 당 장성민 |
| 합계 | 8,382,399 (867,043) | 6,590,646 (843,459) | 2,781,345 (42.34) | 1,365,285 (20.78) | 1,492,767 (22.72) | 476,973 (7.26) | 425,459 (6.47) | 9,987 (0.15) | 789 (0.01) | 3,554 (0.05) |
| 종로구 | 133,769 (15,511) | 102,566 (14,822) | 42,512 (41.59) | 22,325 (21.84) | 22,313 (21.83) | 7,412 (7.25) | 7,113 (6.95) | 228 (0.22) | 5 (0.00) | 78 (0.07) |
| 중구 | 109,836 (12,066) | 82,852 (11,486) | 34,062 (41.23) | 17,901 (21.67) | 19,372 (23.45) | 5,874 (7.11) | 4,993 (6.04) | 158 (0.19) | 12 (0.01) | 53 (0.06) |
| 용산구 | 197,962 (22,059) | 148,157 (21,066) | 58,081 (39.33) | 35,230 (23.85) | 32,109 (21.74) | 11,825 (8.00) | 9,773 (6.61) | 299 (0.20) | 17 (0.01) | 68 (0.04) |
| 성동구 | 259,009 (26,997) | 203,175 (26,267) | 86,686 (42.80) | 40,566 (20.03) | 45,674 (22.55) | 15,859 (7.83) | 12,936 (6.38) | 304 (0.15) | 16 (0.00) | 94 (0.04) |
| 광진구 | 305,172 (31,439) | 240,030 (30,667) | 105,512 (44.10) | 46,368 (19.38) | 52,824 (22.08) | 17,114 (7.15) | 16,540 (6.91) | 312 (0.13) | 24 (0.01) | 120 (0.05) |
| 대대구 | 304,972 | 236,092 | 98,958 | 51,631 | 53,359 | 15,129 | 15,107 | 360 | 41 | 150 |

- 위 그림이 지역을 선택하면 나타나는 화면중 하나인데 득표수에 득표율이 괄호()로 함께 나타납니다.
- 이를 제거하고 (를 기준으로 왼쪽 숫자만 얻어서, 콤마(,)를 제거하고, float형으로 변경하는 함수를 get_num으로 준비해 둡니다

```
import re
```

```
def get_num(tmp):
```

```
    return float(re.split('₩(', tmp)[0].replace(',',''))
```

데이터 획득

준비

획득

정리

시각화

split 알아보기

문자열 나누기(split)

```
>>> a = "Life is too short"
>>> a.split()
['Life', 'is', 'too', 'short']
>>> a = "a:b:c:d"
>>> a.split(':')
['a', 'b', 'c', 'd']
```

`a.split()`처럼 괄호 안에 아무런 값도 넣어 주지 않으면 공백(스페이스, 탭, 엔터등)을 기준으로 문자열을 나누어 준다. 만약 `a.split(':')`처럼 괄호 안에 특정한 값이 있을 경우에는 괄호 안의 값을 구분자로 해서 문자열을 나누어 준다. 이렇게 나눈 값은 리스트에 하나씩 들어가게 된다. `['Life', 'is', 'too', 'short']`나 `['a', 'b', 'c', 'd']`처럼 된다.

참조자료

https://wikidocs.net/13#_9

<https://wikidocs.net/4308>

```
: import re
```

```
def get_num(tmp):
```

```
    return float(re.split('₩(', tmp)[0].replace(',','.'))
```

tmp를 '\('를 구분자로 하여 문자열 나눔.
문자열 나눔 리스트 중 항목0을 반환함.
그 값을 float 형으로 변환함.

데이터 획득

준비

획득

정리

시각화

re.split 알아보기

“나누기 기능”

정규 표현식 처리 메서드

파이썬은 정규 표현식을 처리하기 위한 여러 기능을 `re` 모듈로 묶어 두었다. `re` 모듈은 정규 표현식 패턴을 이용해 패턴과 매치하는 텍스트를 발견하는 기능을 제공한다. 이 모듈에서 자주 사용되는 함수를 표 11-6에 정리해 두었다.

```
import re

def get_num(tmp):
    return float(re.split('W(', tmp)[0].replace('.', ''))
```

| 함수 | 값 또는 기능 |
|--|--|
| <code>re.compile(pattern)</code> | <code>pattern</code> 을 패턴 객체로 컴파일한다 |
| <code>re.search(pattern, string)</code> | <code>string</code> 에서 <code>pattern</code> 과 매치하는 텍스트를 탐색한다 (임의 지점 매치) |
| <code>re.match(pattern, string)</code> | <code>string</code> 에서 <code>pattern</code> 과 매치하는 텍스트를 탐색한다 (시작점 매치) |
| <code>re.fullmatch(pattern, string)</code> | <code>string</code> 에서 <code>pattern</code> 과 매치하는 텍스트를 탐색한다 (전체 매치) |
| <code>re.sub(pattern, repl, string)</code> | <code>string</code> 에서 <code>pattern</code> 과 매치하는 텍스트를 <code>repl</code> 로 치환한다 |
| <code>re.split(pattern, string)</code> | <code>string</code> 을 <code>pattern</code> 을 기준으로 나눈다 |

참조 <https://python.bakyeono.net/chapter-11-2.html>

데이터 획득

준비

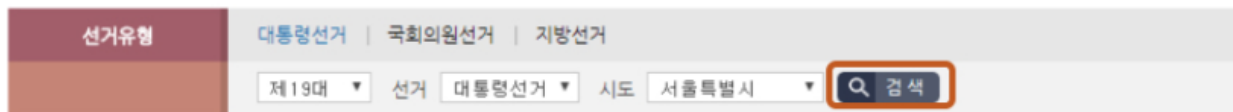
획득

정리

시각화

해당지역의
개표결과
검색하기

- 책에서는 특별히 다루고 있지 않지만, 아래 모듈을 import해서 `wait.until` 함수를 사용할 수 있습니다.



- 이 함수는 검색버튼이 클릭가능할 때 까지 기다리는 기능을 합니다.
- 그리고 `move_sido` 함수는 광역시도 이름을 리스트에 전송하고 검색 버튼을 누르는 역할을 합니다.

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

```
wait = WebDriverWait(driver, 10)
```

```
def move_sido(name):
    element = driver.find_element_by_id("cityCode")
    element.send_keys(name)
    make_xpath = "//*[@id='searchBtn']"
    wait.until(EC.element_to_be_clickable((By.XPATH, make_xpath)))
    driver.find_element_by_xpath(make_xpath).click()
```


데이터 획득

준비

획득

정리

시각화

칼럼 추가하기

리스트 찾아놓은 항목 0에서 5번째 항목을 데이터 프레임 칼럼 각각에 추가한다.

```
def append_data(df, sido_name, data):  
    for each in df[0].values[1:]:  
        data['광역시도'].append(sido_name)  
        data['시군'].append(each[0])  
        data['pop'].append(get_num(each[2]))  
        data['moon'].append(get_num(each[3]))  
        data['hong'].append(get_num(each[4]))  
        data['ahn'].append(get_num(each[5]))
```

- `append_data` 함수는 빈 내용으로 미리 준비한 `DataFrame`에 `append` 명령으로 읽은 데이터를 하나씩 추가하는 기능입니다.

결과보기

[제19대] [대통령선거] [서울특별시]

| 구시군명 | 선거인수 | 투표수 | 후보자별 득표수(득표율) | | | | | | | |
|------|------------------------|------------------------|----------------------|----------------------|----------------------|-------------------|-------------------|-----------------|---------------|-------------------|
| | | | 더불어민주 문재인 | 자유한국당 홍준표 | 국민의당 안철수 | 바른정당 유승민 | 정의당 심상정 | 새누리당 조원진 | 경제자유당 오영국 | 국민대통합 당 정성민 |
| 합계 | 8,382,999 (867,043) | 6,590,646 (843,459) | 2,781,345 (42.34) | 1,365,285 (20.78) | 1,492,767 (22.72) | 476,973 (7.26) | 425,459 (6.47) | 9,987 (0.15) | 789 (0.01) | 3,554 (0.05) |
| 종로구 | 133,769 (15,511) | 102,566 (14,822) | 42,512 (41.59) | 22,325 (21.84) | 22,313 (21.83) | 7,412 (7.25) | 7,113 (6.95) | 228 (0.22) | 5 (0.00) | 78 (0.07) |
| 중구 | 109,836 (12,066) | 82,852 (11,486) | 34,062 (41.23) | 17,901 (21.67) | 19,372 (23.45) | 5,874 (7.11) | 4,993 (6.04) | 158 (0.19) | 12 (0.01) | 53 (0.06) |
| 용산구 | 197,962 (22,059) | 148,157 (21,066) | 58,081 (39.33) | 35,230 (23.85) | 32,109 (21.74) | 11,825 (8.00) | 9,773 (6.61) | 299 (0.20) | 17 (0.01) | 68 (0.04) |
| 성동구 | 259,009 (26,997) | 203,175 (26,267) | 86,686 (42.80) | 40,566 (20.03) | 45,674 (22.55) | 15,859 (7.83) | 12,936 (6.38) | 304 (0.15) | 16 (0.00) | 94 (0.04) |
| 광진구 | 305,172 (31,439) | 240,030 (30,667) | 105,512 (44.10) | 46,368 (19.38) | 52,824 (22.08) | 17,114 (7.15) | 16,540 (6.91) | 312 (0.13) | 24 (0.01) | 120 (0.05) |
| 노원구 | 304,972 | 236,092 | 98,958 | 51,631 | 53,359 | 15,129 | 15,107 | 360 | 41 | 150 |

- 이 함수가 실행되면 전체 투표수, 문재인후보, 홍준표후보, 안철수후보의 득표수가 추가됩니다

데이터 획득

준비

획득

정리

시각화

데이터프레임의 열의 값을 변수로 만들기.

- 딕셔너리 형으로 만들어 줌.
- key에 해당 Value 가짐.

미리 변수를 하나 만들어 둡니다.

```
election_result_raw = { '광역시도' : [],  
                        '시군' : [],  
                        'pop' : [],  
                        'moon' : [],  
                        'hong' : [],  
                        'ahn' : [] }
```

파일저장

```
In [ ]: election_result = pd.DataFrame(election_result_raw,  
                                       columns=['광역시도', '시군', 'pop', 'moon', 'hong', 'ahn'])  
election_result
```

```
In [ ]: election_result.to_csv('../data/05. election_result.csv', encoding='utf-8', sep=',')
```

```
In [ ]: driver.close()
```