

서울 날씨 변화 분석(1994-2024)

팀명: 채정현팀 2조

팀원: 임동혁, 주지훈

기간: 2024.9.24(화) ~ 2024.9.30(월)

깃합: <https://github.com/dev-jhjoo/SeoulWeatherAnalysis>

목차

1. 수행배경
2. 데이터 수집
3. 데이터 전처리
4. 데이터 시각화
5. 결론 및 제언

1. 수행배경

서울은 매년 더워 진다. 무더위가 찾아오는 주기가 존재할 것이다. - 임동혁

봄, 가을은 짧아졌다. - 주지훈

올여름은 역대급 무더위라는 말을 많이 접했다. 개인적으로 추석 연휴에도 이렇게 더웠던 적이 있었을까 싶을 정도로 더위를 체감했다. 뉴스에서는 “여름철 기온 1위, 열대야 일수 1위, 시간당 강수량 1위, 해수면 온도 1위”라며 역대 최악의 여름이라고 보도했다. 이에 실제

올여름이 얼마나 더웠는지 데이터 분석을 통해 알아보고 이 결과를 바탕으로 활용할 수 있는 방향을 알아보고자 한다.

2. 데이터 수집

원본 데이터 정보:

- Kaggle | <https://www.kaggle.com/datasets/alfredkondoro/seoul-historical-weather-data-2024/data>

데이터 설명:

- 총 10972 row
- 용량 3.73MB

데이터 세트에는 다음과 같은 열이 포함.

- 날짜(datetime): 기록된 날씨 데이터의 날짜.
- 최고 기온(tempmax): 그날 기록된 가장 높은 기온(°F).
- 최저 기온(tempmin): 그날 기록된 가장 낮은 기온(°F).
- 평균 기온(temp): 해당 날짜에 기록된 평균 기온(°F).
- 체감 온도(feelslike): 습도와 바람을 고려하여 감지되는 온도(°F).
- 이슬점(이슬): 이슬이 형성되는 온도(°F).
- 습도(Humidity): 공기 중 습도의 백분율.
- 강수량 (precip): 기록된 총 강수량(mm).
- 눈(snow) : 기록된 총 강설량(mm).
- 풍속(windspeed): 평균 풍속(km/h).
- 풍향(winddir): 바람이 부는 방향(도).
- 해수면 기압(sealevelpressure): 해수면에서의 대기압(hPa).
- 구름 덮개(cloudcover): 구름으로 덮인 하늘의 비율.

- 가시성(visibility): 가시거리(km).
- 태양 복사선(solarradiation): 표면이 받는 태양 복사선(W/m²).
- 자외선 지수(uvindex): 햇볕에 타는 자외선의 강도를 측정하는 자외선 지수.
- 상태(conditions): 날씨 상태에 대한 설명(예: 맑음, 구름 약간).
- 설명(description): 그날의 날씨에 대한 텍스트 설명.

계절 정의:

현재 기상청에서는 이병설 박사(1976 우리나라 자연계절에 따른 연구)의 기준을 따라 계절을 정의하고있다.

- 봄: 일평균기온이 5°C 이상 올라간 후 다시 떨어지지 않는 첫날 부터
- 여름: 일평균기온이 20°C 이상 올라간 후 다시 떨어지지 않는 첫날 부터
- 가을: 일평균기온이 20°C 미만으로 떨어진 후 다시 올라가지 않는 첫날 부터
- 겨울: 일평균기온이 5°C 미만으로 떨어진 후 다시 올라가지 않는 첫날 부터

출처: <https://kscc.re.kr/2020kscc/papers/Oral/A-03.pdf>

3. 데이터 전처리

원본 데이터에서 필요한 데이터는 날짜 및 일 평균 기온으로 2개 컬럼을 사용한다. 따라서, 2개 컬럼을 새로운 DataFrame으로 구성하여 사용한다.

원본 데이터 합치기

캐글에서 제공하는 데이터셋은 2년 주기로 총 15개 csv 파일을 제공한다. 그중 두번째 파일(seoul 1996-01-01 to 1998-01-01.csv)부터 연도별로 1월 1일 데이터가 겹치는걸 확인. 이에 중복되는 첫번째 row를 제외하기 위해 슬라이싱 처리 후 하나의 DataFrame으로 구성했다.

합쳐진 원본 데이터 확인

```
<class 'pandas.core.frame.DataFrame'>
Index: 10958 entries, 0 to 730
```

Data columns (total 33 columns):

#	Column	Non-Null Count	Dtype
0	name	10958 non-null	object
1	datetime	10958 non-null	object
2	tempmax	10958 non-null	float64
3	tempmin	10958 non-null	float64
4	temp	10958 non-null	float64
5	feelslikemax	10958 non-null	float64
6	feelslikemin	10958 non-null	float64
7	feelslike	10958 non-null	float64
8	dew	10958 non-null	float64
9	humidity	10958 non-null	float64
10	precip	10958 non-null	float64
11	precipprob	10958 non-null	int64
12	precipcover	10958 non-null	float64
13	preciptype	4045 non-null	object
14	snow	8035 non-null	float64
15	snowdepth	8242 non-null	float64
16	windgust	3642 non-null	float64
17	windspeed	10958 non-null	float64
18	winddir	10958 non-null	float64
19	sealevelpressure	10958 non-null	float64
20	cloudcover	10958 non-null	float64
21	visibility	10958 non-null	float64
22	solarradiation	5114 non-null	float64
23	solarenergy	5114 non-null	float64
24	uvindex	5114 non-null	float64
25	severerisk	722 non-null	float64
26	sunrise	10958 non-null	object
27	sunset	10958 non-null	object
28	moonphase	10958 non-null	float64
29	conditions	10958 non-null	object
30	description	10958 non-null	object
31	icon	10958 non-null	object
32	stations	10958 non-null	object

dtypes: float64(23), int64(1), object(9)

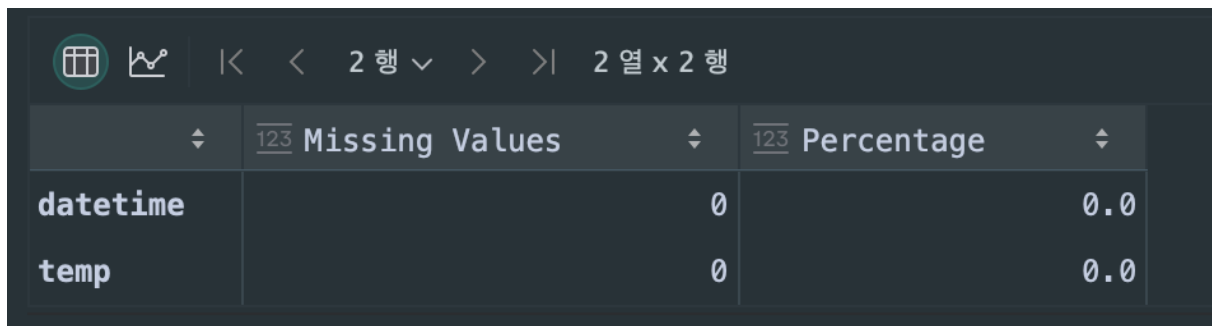
memory usage: 2.8+ MB

분석에 필요한 feature들로 새로운 DataFrame 구성

```
<class 'pandas.core.frame.DataFrame'>
Index: 10958 entries, 0 to 730
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   datetime    10958 non-null  object 
 1   temp        10958 non-null  float64
dtypes: float64(1), object(1)
memory usage: 256.8+ KB
```

이번 데이터 분석시 필요한 feature들은 날짜(datetime), 평균기온(temp) 2개. 2개 컬럼으로 새로운 DataFrame을 구성하여 원본데이터 크기에서 1/10로 줄였다.

결측치 분석



	Missing Values	Percentage
datetime	0	0.0
temp	0	0.0

처리해야 할 결측치는 없는 것으로 확인했다.

일 평균 온도 화씨에서 섭씨로 변환

전처리 중 2022-01-01 부터는 이미 섭씨로 변환된 데이터가 들어간것을 확인하여 화씨에서 섭씨로 변경할 필요가 없는것을 확인했다. 2022-01-01 날짜를 기준으로 화씨 변환을 진행하고, temp_celsius 컬럼으로 추가했다.

봄, 여름, 가을, 겨울 구분

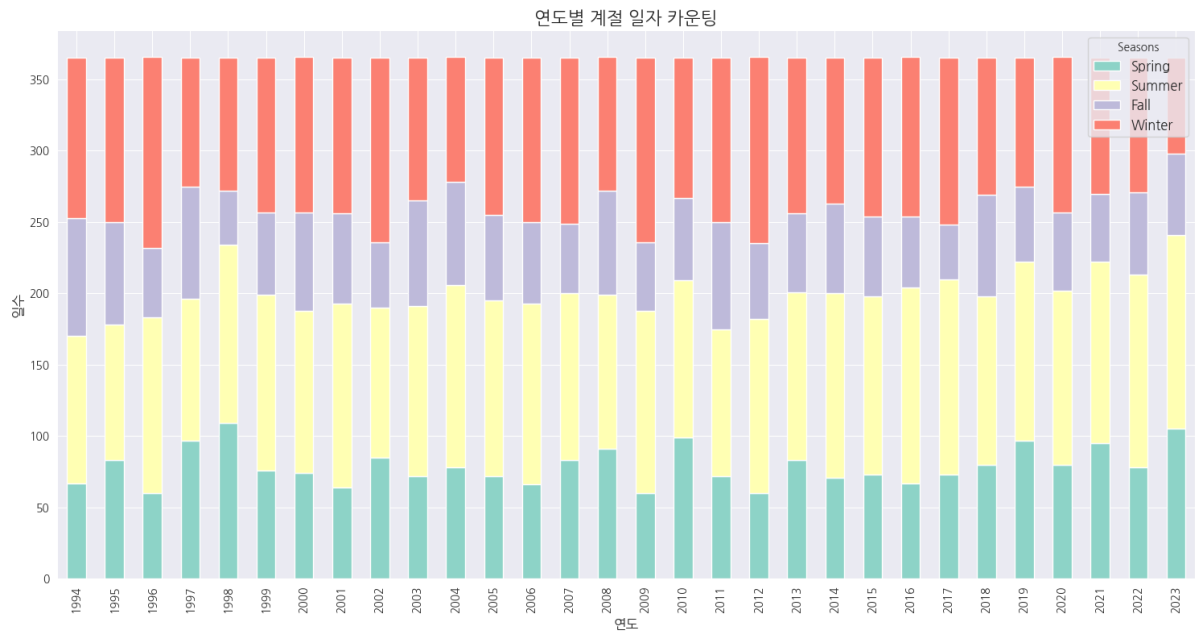
```
<class 'pandas.core.frame.DataFrame'>
Index: 10958 entries, 0 to 730
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   datetime        10958 non-null  object
1   temp            10958 non-null  float64
2   temp_celsius    10958 non-null  float64
3   seasons         10958 non-null  object
dtypes: float64(2), object(2)
memory usage: 428.0+ KB
```

데이터 수집부분 계절 정의에 따라 첫째날 기준 앞으로 2주간(14일)의 기온 중 조건을 만족하는 일수가 10일 이상이라면 해당 계절을 정의한다.

- 봄: 기준일 앞으로 2주간(14일)의 온도가 5도 이상인 날이 10일 이상이라면 봄으로 정의
- 여름: 기준일 앞으로 2주간(14일)의 온도가 20도 이상인 날이 10일 이상이라면 여름으로 정의
- 가을: 기준일 앞으로 2주간(14일)의 온도가 20도 미만인 날이 10일 이상이라면 가을로 정의
- 겨울: 기준일 앞으로 2주간(14일)의 온도가 5도 미만인 날이 10일 이상이라면 겨울로 정의

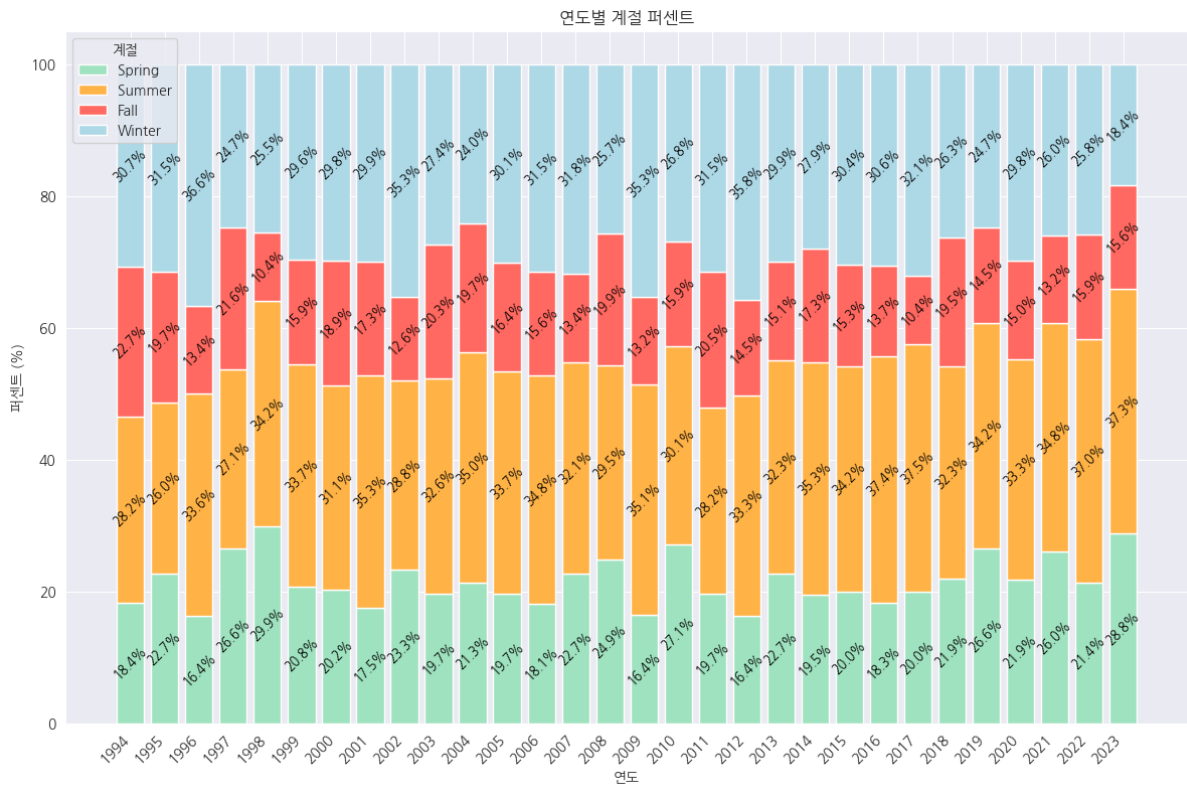
4. 데이터 분석 및 시각화

연도별 계절일수 카운트



연도별로 계절일수를 시각화한 결과이다. 큰 변화를 구분하기 어려웠다.

연도별 계절 일자를 퍼센트로 확인



좀 더 시각적으로 구분하기 쉽도록 계절별 일수를 퍼센트로 추가했다. 전체적인 연도별로 계절의 비율은 알수 있었으나, 계절별 일수의 변화를 알아보기 쉽지 않았다.

계절별 일수 변화



각 계절별로 연도별 일수를 나타내고 그 위에 추세선을 추가했다. 이를 통해 여름의 일자가 상승세 인걸 확인할 수 있었다.

계절별 시작된 일자 확인

seasons	Spring	Summer	Fall	Winter
1994	1994-03-24	1994-05-30	1994-09-10	1994-12-02
1995	1995-03-12	1995-06-03	1995-09-06	1995-11-17
1996	1996-03-25	1996-05-24	1996-09-24	1996-11-12
1997	1997-02-25	1997-06-02	1997-09-09	1997-11-27
1998	1998-02-16	1998-06-05	1998-10-08	1998-11-15
1999	1999-03-12	1999-05-27	1999-09-27	1999-11-24
2000	2000-03-09	2000-05-22	2000-09-13	2000-11-21
2001	2001-03-12	2001-05-15	2001-09-21	2001-11-23
2002	2002-03-05	2002-05-29	2002-09-11	2002-10-27
2003	2003-03-10	2003-05-21	2003-09-17	2003-11-30

연도별로 각 계절의 시작일을 구했다. 봄, 여름, 가을에 경우 1년 내에 존재하기에 min 함수를 통해 구할 수 있었다. 겨울의 경우 연도가 변하는 구간이 존재하기 때문에 min 함수를 사용할 경우 해당 연도 1월 1일을 나타냈다. 이를 해결하기 위해 max 함수를 통해 가을의 마지막 날을 구하고 +1 day를 통해 겨울의 시작일을 구했다.

계절별 시작된 일자변화에 대해 시각화

연도별 각 계절의 시작일

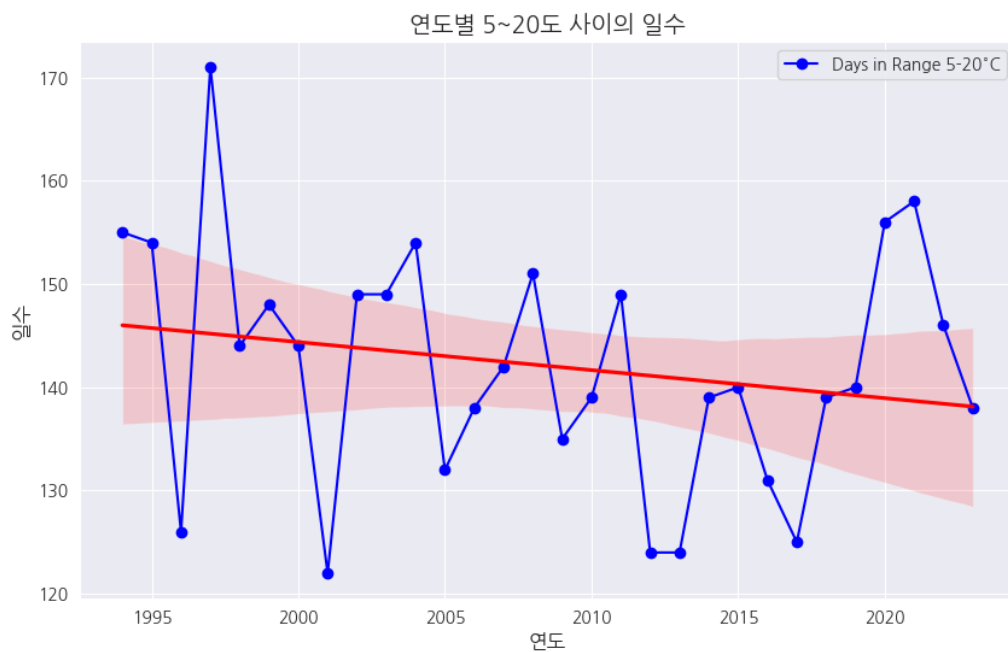


연도별로 각 계절의 시작일을 시각화 했다. 봄, 여름의 시작일이 점점 앞당겨 지는 추세였다. 가을의 경우 이에따라 가을의 시작일이 점점 늦어지는걸 확인했다. 결과적으로 봄, 여름이 빨리 찾아오며 가을은 늦게 찾아온다는 결과를 알수 있다.

온도의 따른 일수 변화 확인

계절의 정의에 따라 봄, 가을은 섭씨 5~20도 사이의 온도를 의미한다. 따라서, 연도별로 5~20도 사이의 일수 변화를 알아보았다.

datetime	123 <unnamed>
1994	155
1995	154
1996	126
1997	171
1998	144
1999	148
2000	144
2001	122
2002	149
2003	149



그래프를 통해 봄, 가을에 해당하는 5~20도 사이의 일수가 크게는 아니지만 점점 적어지는 경향을 볼 수 있다.

5. 결론 및 제언

날씨 변화에 대한 결과를 통해 활용할 수 있는 부분에 대해서도 생각해봤다. 이커머스 산업에서는 시즌별 상품판매 전략에 활용할 수 있을것이다. 또한, 계절별 상품과 관련하여 소비 패턴 분석에서도 날씨 변화와 연관지어 패턴화 할 수 있을 것이다. 마지막으로 한여름 열대야에 맞춰 기후변화에 대해 자연친화적인 회사라는 인식을 심어줄 광고를 진행한다면 더 큰 효과를 얻을 수 있을 것 같다. 그래서 추후 연관된 분석을 진행할때도 해당 날씨 변화 데이터를 활용할 예정이다.

이번 데이터 분석 프로젝트는 올 여름이 역대급 무더위였다는 사실에서 시작했다. 그중 봄, 가을이 짧아졌다는 증명하기 위해 원본 데이터를 통해 계절을 구분해야했다. 이부분이 특히 복잡했다. 그중 기상청에서 기온에 변화에 따른 계절 구분방법을 찾았고, 기상청에서 정의내린 계절 구분법이라는점에서 신뢰성을 얻었다. 이후 이를 함수로 구현하여 계절정보를 얻을 수 있었다.

온도와 계절 정보를 활용하여 데이터 분석을 진행하며, 실제로 서울의 평균기온이 점점 올라가고 있다는점과 실제로 역대급 무더위였다는 사실을 확인할 수 있었다. 물론, 30년간의 데이터라는 점과 모든 추세선이 급격한 변화를 보이진 않았지만 꾸준히 그리고 확실하게 더워지고 있었다. 데이터를 통해 사실을 확인하니 현실을 살아가는 사람으로서 기온변화에 대해 경각심을 느꼈다.