충남 데이터 청년 인턴십 프로그램

구내식당 식수 인원 예측

태안발전본부

박준혁, 조서영

CONTENTS

1. 과제개요

과제명

구내식당 식수인원 예측모델 생성을 위한 데이터 분석

데이터

한국토지주택공사에서 제공한 구내식당 데이터 활용(데이콘), 날씨데이터 수집

목적

일일 예상 식수인원 산출

정확한 식수 예측을 통해 원재룟값 절감을 통항
식사 품질개선 및 음식물 쓰레기 절감을 위한
환경오염 예방

주요 분석 내용

약 3개년 식수인원 보유데이터 기반 일별/주별/월별 식수 인원을 예측하고, 추가로 날씨에 따른 구내식당 이용률을 분석

1-2. 분석 데이터 설명

5년간 회사 인원 데이터

출근 정원수 , 재택근무자수, 휴가자수, 출장자수, 점심,저녁 메뉴 데이터, 당일 중식을 먹은 인원, 당일 석식을 먹은 인원

날씨 데이터

불쾌지수 , 체감온도 , 날씨 동향 데이터

2-1. 수집 데이터 설명

1. 날씨 데이터

- 기상청 기상자료개방포털 - 종관기상관측(ASOS)자료
- 경상남도 진주시의 2016-02-01 ~ 2021-01-26기간의 기온, 강수량, 습도, 풍속 데이터 수집

2. 미세먼지 데이터

- Air경남 대기환경정보 - 통계정보 -> 확정자료 페이지
- 경상남도 진주시 상대동(측정소 위치)
- 측정망 위치 - 도시대기
- 2016-02-01 ~ 2021-01-26기간의 미세먼지(PM-10) 데이터 수집

3. 강수형태 데이터

- 기상청 기상자료개방포털 - 기상예보 - 초단기실황 페이지
- 경상남도 진주시 충무공동
- 2016-02-01 ~ 2021-01-26 기간의 강수형태 데이터 수집

2-2. 전처리 과정

[날씨 데이터]

```
# 날씨 파일 불러오기
w1 = pd.read_csv('날씨1.csv', encoding='cp949')
w2 = pd.read_csv('날씨2.csv', encoding='cp949')
w3 = pd.read_csv('날씨3.csv', encoding='cp949')
w4 = pd.read_csv('날씨4.csv', encoding='cp949')
w5 = pd.read_csv('날씨5.csv', encoding='cp949')
```

	지점	지점명	일시	기온(°C)	강수량(mm)	풍속(m/s)	습도(%)
0	192	진주	2016-02-01 12:00:00	3.0	NaN	1.9	24.0
24	192	진주	2016-02-02 12:00:00	1.7	NaN	1.6	24.0

```
# 점심시간(12시)의 데이터만 추출
w1 = w1[w1['일시'].dt.hour == 12]
w2 = w2[w2['일시'].dt.hour == 12]
w3 = w3[w3['일시'].dt.hour == 12]
w4 = w4[w4['일시'].dt.hour == 12]
w5 = w5[w5['일시'].dt.hour == 12]
```

- 경상남도 진주시의 날씨 데이터 중 점심시간인 12시의 데이터만 추출
- 결측값 처리
- 필요없는 컬럼 삭제
- date컬럼에서 시간 삭제
- 폭염, 체감온도, 불쾌지수 컬럼을 생성

2-2. 전처리 과정

[날씨 데이터] _폭염, 불쾌지수, 체감온도 컬럼 생성

```
# 기온컬럼 이용 -> 폭염인 경우 1 / 폭염 아닌 경우 0으로 분류

hw = 33         # 폭염 기준 온도값 초기화
w_all['폭염'] = w_all['기온(° C)'].apply(lambda x: 1 if x >= hw else 0) # 33도 이상이면 1 아니면 0
w_all.head(2)
```

```
w_all['불쾌지수'] = 1.8 * w_all['기온(° C)'] - 0.55 * (1 - w_all['습도(%)'] / 100) * (1.8 * w_all['기온(° C)'] - 26) + 32
w_all['체감온도'] = 13.12 + 0.6215 * w_all['기온(° C)']- 11.37 * w_all['풍속(m/s)']**0.16 + 0.3965 * w_all['기온(° C)'] * w_all['풍속(m/s)']**0.16
```

	date	기온(° C)	강수량(mm)	풍속(m/s)	습도(%)	폭염	불쾌지수	체감온도
0	2016-02-01	3.0	0.0	1.9	24.0	0	46.01080	3.702926
24	2016-02-02	1.7	0.0	1.6	24.0	0	44.64892	2.645242

2-2. 전처리 과정

[미세먼지 데이터]

```
# 미세먼지 데이터셋 불러오기 (경상남도 진주시 상대동(측정소))
m1 = pd.read_csv('미세먼지_160201_170126.csv', encoding='cp949')
m2 = pd.read_csv('미세먼지_170127_180126.csv', encoding='cp949')
m3 = pd.read_csv('미세먼지_180127_190126.csv', encoding='cp949')
m4 = pd.read_csv('미세먼지_190127_200126.csv', encoding='cp949')
m5 = pd.read_csv('미세먼지_200127_210126.csv', encoding='cp949')
```

```
# 년도별 미세먼지 데이터 concat
m_all = pd.concat([m1, m2, m3, m4, m5])
m_all.head(3)
```

	일자	PM-10
0	2016-02-01	36.0
1	2016-02-02	34.0
2	2016-02-03	54.0

	date	미세먼지
0	2016-02-01	36.0
1	2016-02-02	34.0
2	2016-02-03	54.0
3	2016-02-04	78.0

- NaN값에는 월 평균 값을 적용
- 컬럼명 수정

2-2. 전처리 과정

[강수형태 데이터]

```
# 강수형태 데이터 불러오기
r1 = pd.read_csv('충무공동_강수형태_201602_201701.csv')
r2 = pd.read_csv('충무공동_강수형태_201702_201801.csv')
r3 = pd.read_csv('충무공동_강수형태_201802_201901.csv')
r4 = pd.read_csv('충무공동_강수형태_201902_202001.csv')
r5 = pd.read_csv('충무공동_강수형태_202002_202101.csv')
```

```
# 10시 ~14시(1000 ~ 1400기간)데이터만 추출
rain1 = r_all[r_all['시간'] >= 1000]  # 1000(10시)이상 데이터 추출 후 저장
rain2 = rain1[rain1['시간'] <= 1400]  # 10시 이상 데이터 중 14시 이하의 데이터
rain_use = rain2.copy()                # 복사본으로 저장
#rain_use.head(4)
```

	일자	시간	강수형태
0	1	1000.0	0.0
1	1	1100.0	0.0
2	1	1200.0	0.0
3	1	1300.0	0.0
4	1	1400.0	0.0

- 10시 ~ 12시의 데이터만 추출
- 일자 컬럼 -> 년/월/일로 변경

2-2. 전처리 과정

[강수형태 데이터]_rain, snow 컬럼 생성

- rain과 snow컬럼 추가

```
# rain 컬럼 추가 -> 강수형태가 1 or 2이면 -> 1 / 그 외(-1 or 3)이면 -> 0
rain_use['rain'] = rain_use['강수형태'].apply(lambda x: 1 if x in [1, 2] else 0)

# snow 컬럼 추가 -> 강수형태가 3이면 -> 1 / 그 외에는 0
rain_use['snow'] = rain_use['강수형태'].apply(lambda x: 1 if x == 3 else 0)

# 중복된 날짜 제거 (첫 번째 중복된 행을 유지)
rain_use = rain_use.drop_duplicates(subset=['일자'])
rain_use.head()
```

	date	시간	강수형태	rain	snow
0	2016-02-01	1000.0	0.0	0	0
1	2016-02-01	1100.0	0.0	0	0

- rain 변수 추가 -> 강수형태가 0이거나 1일때 비가 왔다고 판단
- snow 변수 추가 -> 강수형태가 3일때 눈이 왔다고 판단

2-3. train.csv와 병합한 데이터셋

	date	dow	employees	dayoff	bustrip	ovtime	remote	target_ln	target_dn	year	...	강수량(mm)	풍속(m/s)	습도(%)	폭염	불쾌지수	체감온도	미세먼지	강수형태	rain	snow
0	2016-02-01	1	2601	50	150	238	0.0	1039.0	331.0	2016	...	0.0	1.9	24.0	0	46.01080	3.702926	36.0	0.0	0	0
1	2016-02-02	2	2601	50	173	319	0.0	867.0	560.0	2016	...	0.0	1.6	24.0	0	44.64892	2.645242	34.0	0.0	0	0
2	2016-02-03	3	2601	56	180	111	0.0	1017.0	573.0	2016	...	0.0	1.5	31.0	0	45.44108	4.330567	54.0	0.0	0	0
3	2016-02-04	4	2601	104	220	355	0.0	978.0	525.0	2016	...	0.0	0.7	30.0	0	46.10590	6.065918	78.0	0.0	0	0
4	2016-02-05	5	2601	278	181	34	0.0	925.0	330.0	2016	...	0.0	1.0	17.0	0	50.03229	8.163400	51.0	0.0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1200	2021-01-20	3	2983	75	198	4	391.0	1093.0	421.0	2021	...	0.0	1.2	31.0	0	47.22812	6.356172	34.0	0.0	0	0
1201	2021-01-21	4	2983	92	231	462	351.0	832.0	353.0	2021	...	0.0	0.4	52.0	0	50.91968	12.072254	30.0	1.0	1	0
1202	2021-01-22	5	2983	255	248	1	303.0	579.0	217.0	2021	...	0.0	0.5	94.0	0	47.47904	11.145135	32.0	0.0	0	0
1203	2021-01-25	1	2983	107	153	616	327.0	1145.0	502.0	2021	...	0.0	2.6	54.0	0	53.77198	12.115334	21.0	0.0	0	0
1204	2021-01-26	2	2983	69	183	551	362.0	1015.0	480.0	2021	...	4.2	0.5	97.0	0	43.40486	8.997105	11.0	0.0	0	0

1205 rows × 23 columns

```
train.columns
```

```
Index(['date', 'dow', 'employees', 'dayoff', 'bustrip', 'ovtime', 'remote',
       'target_ln', 'target_dn', 'year', 'month', 'day', '기온(°C)', '강수량(mm)',
       '풍속(m/s)', '습도(%)', '폭염', '불쾌지수', '체감온도', '미세먼지', '강수형태', 'rain',
       'snow'],
      dtype='object')
```

3-1. 데이터 정제

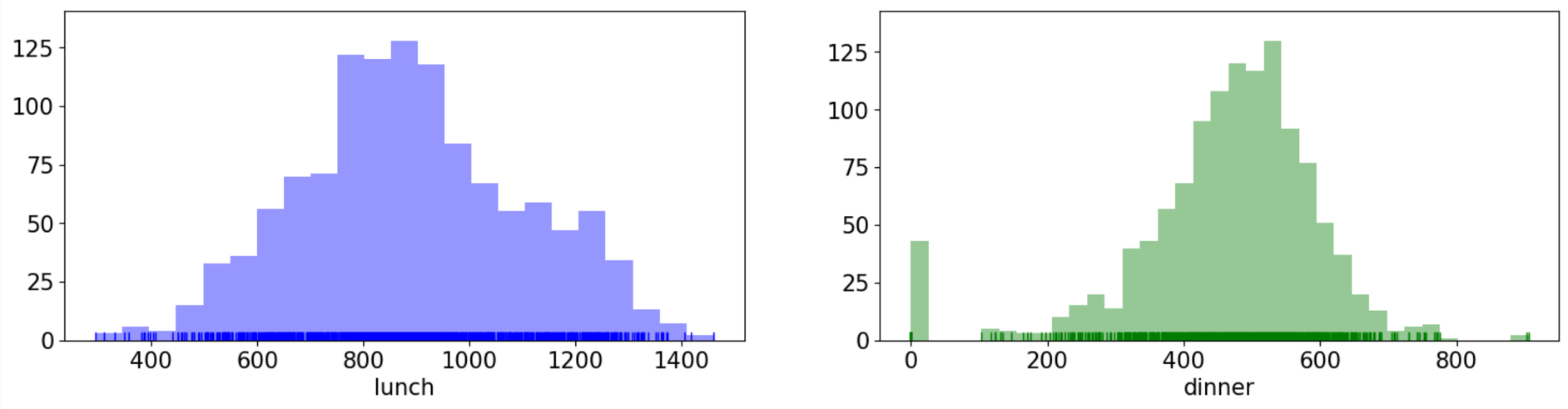
date	dow
2021-01-27	수
2021-01-28	목

date	dow
2016-02-01	1
2016-02-02	2

3-1. 데이터 정제

	date	dayoff	bustrip	ovtime	remote	dow	dn	target_dn
204	2016-11-30	68	207	0	0.0	3	*	0.0
224	2016-12-28	166	225	0	0.0	3	*	0.0
244	2017-01-25	79	203	0	0.0	3	*	0.0
262	2017-02-22	75	252	0	0.0	3	*	0.0
281	2017-03-22	53	235	0	0.0	3	*	0.0
306	2017-04-26	45	304	0	0.0	3	*	0.0
327	2017-05-31	43	265	0	0.0	3	자기계발의날	0.0
346	2017-06-28	58	259	0	0.0	3	*자기계발의날*	0.0
410	2017-09-27	70	265	0	0.0	3	쌀밥/잡곡밥 (쌀:국내산) 된장찌개 미니함박조림 계란말이 비름나물 포기김치 ...	0.0
412	2017-09-29	214	248	22	0.0	5	*	0.0
424	2017-10-25	75	289	0	0.0	3	*	0.0
449	2017-11-29	78	261	0	0.0	3	*	0.0
468	2017-12-27	169	255	0	0.0	3	*	0.0
492	2018-01-31	56	223	0	0.0	3	*	0.0
502	2018-02-14	418	159	0	0.0	3	쌀밥/잡곡밥 (쌀:국내산) 쇠고기무국 고추잡채*꽃빵 계란찜 오이무침 포기김치...	0.0

점심,저녁 이용자수의 히스토그램 분포도

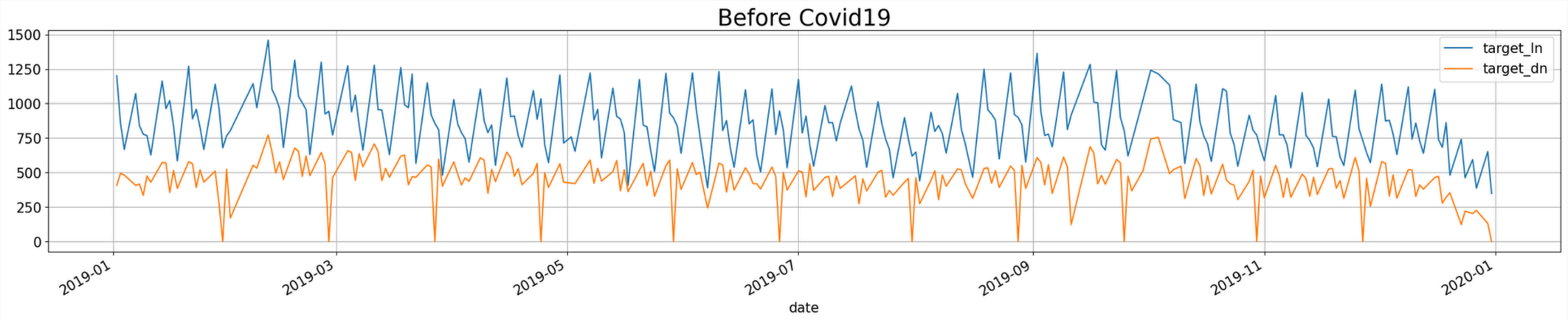


점심식사인원

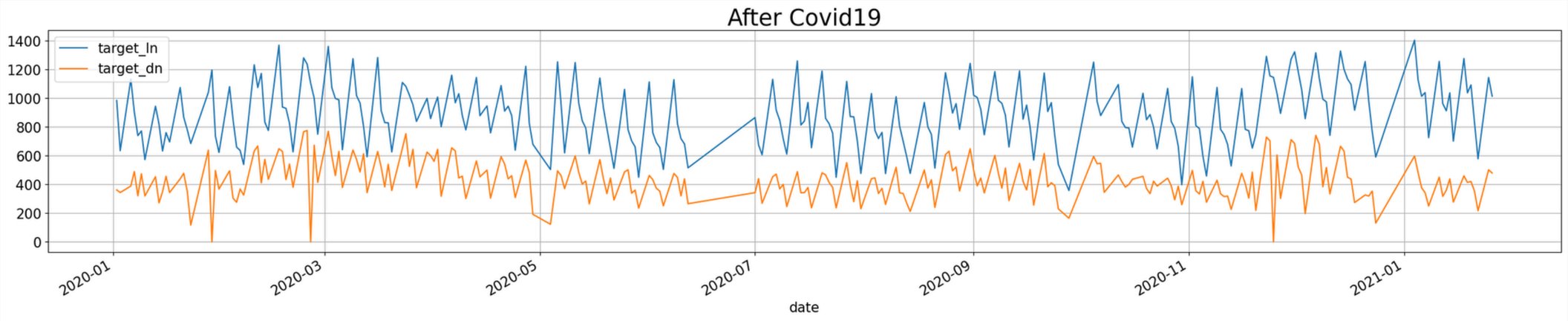
저녁식사인원

점심저녁이용자수 시계열데이터

코로나 전



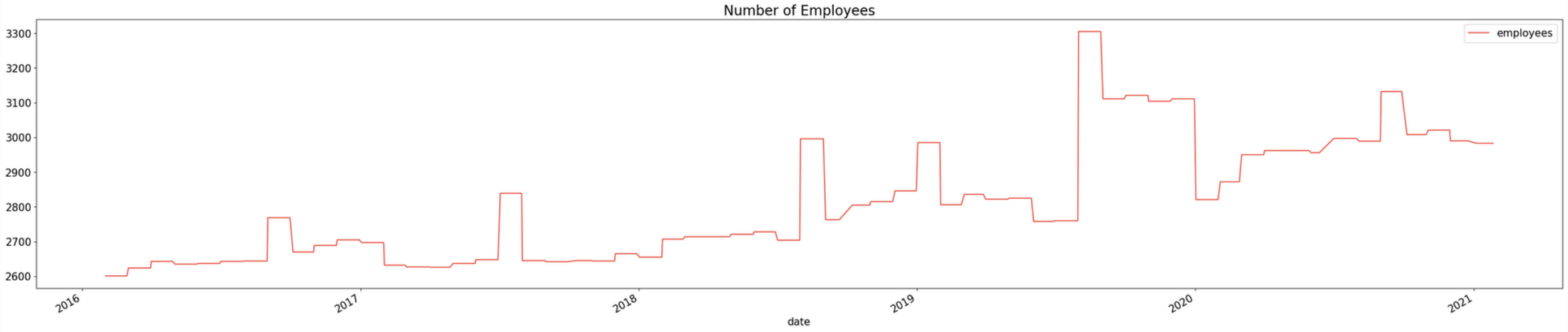
코로나 후



코로나 전후 점심과 저녁 이용자수 평균

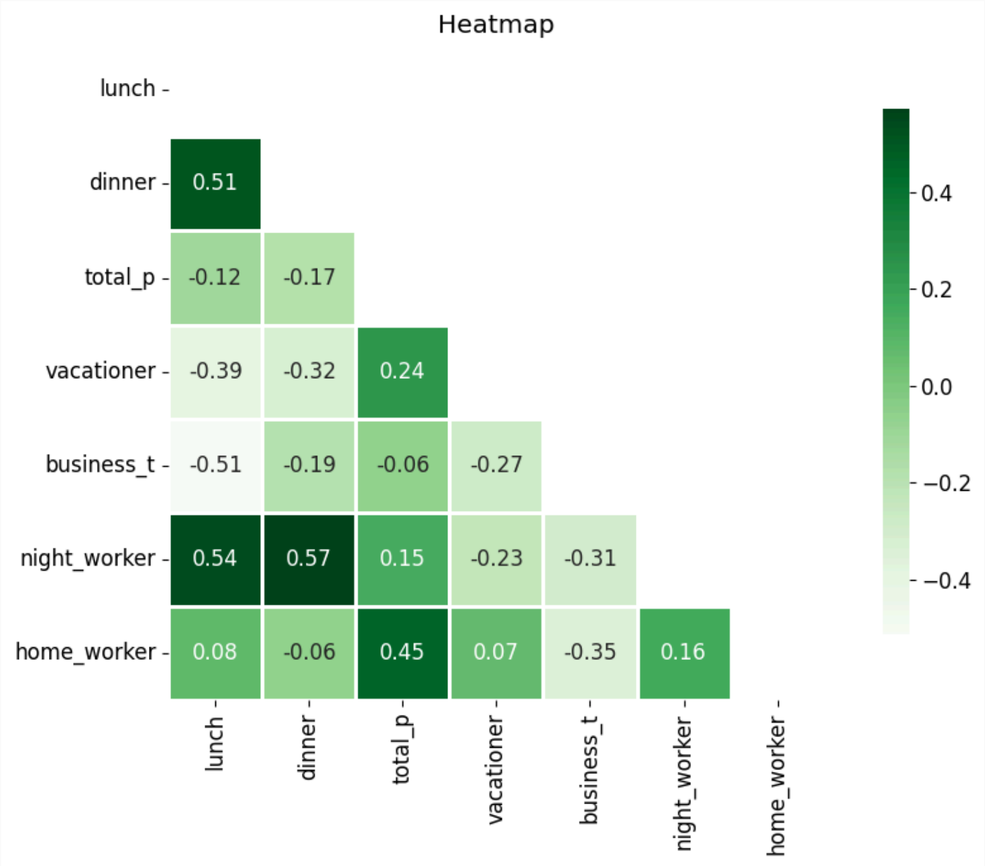
```
점심: 2019년에는 850.51 , 2020년에는 890.97
저녁: 2019년에는 445.39 , 2020년에는 428.34
```

총 직원수 시계열 추이

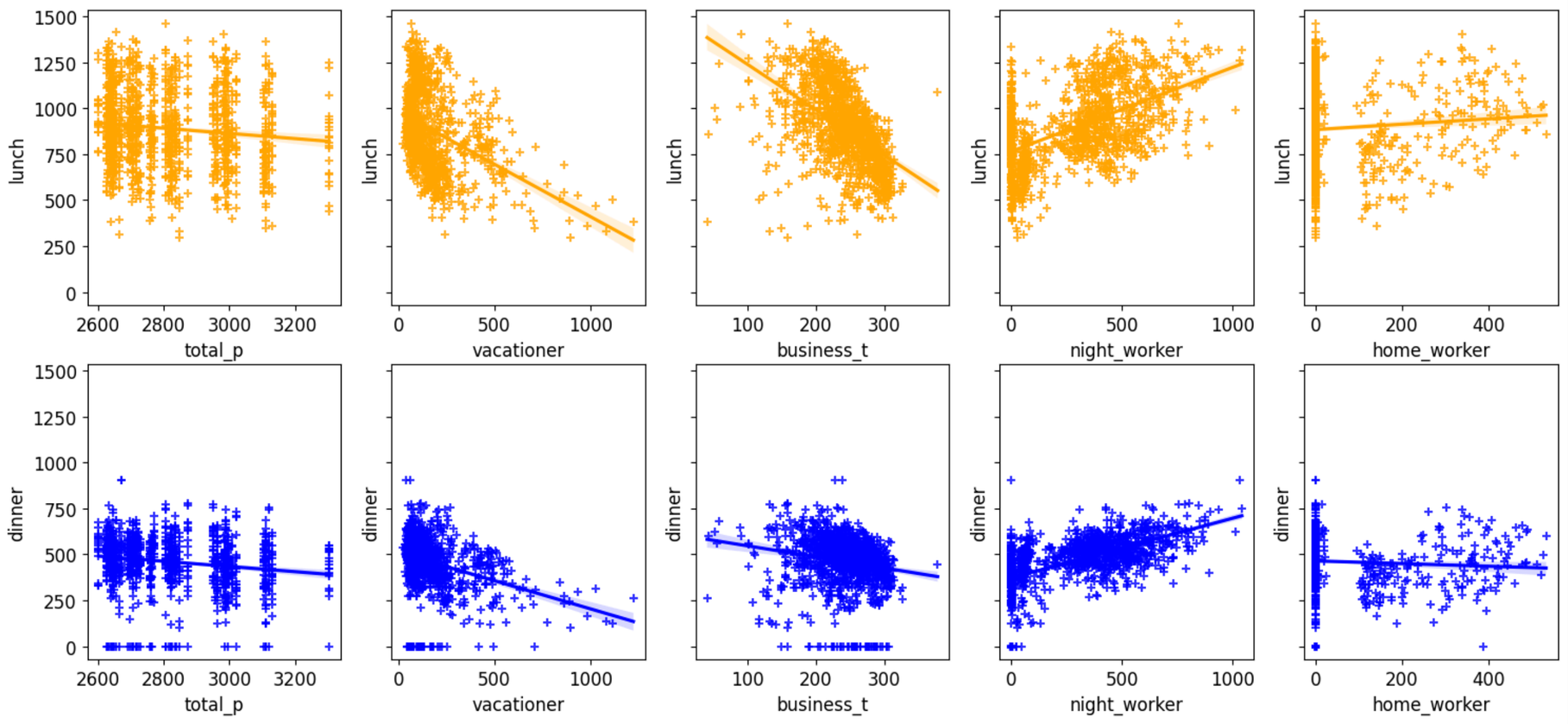


3-3. 상관관계 분석

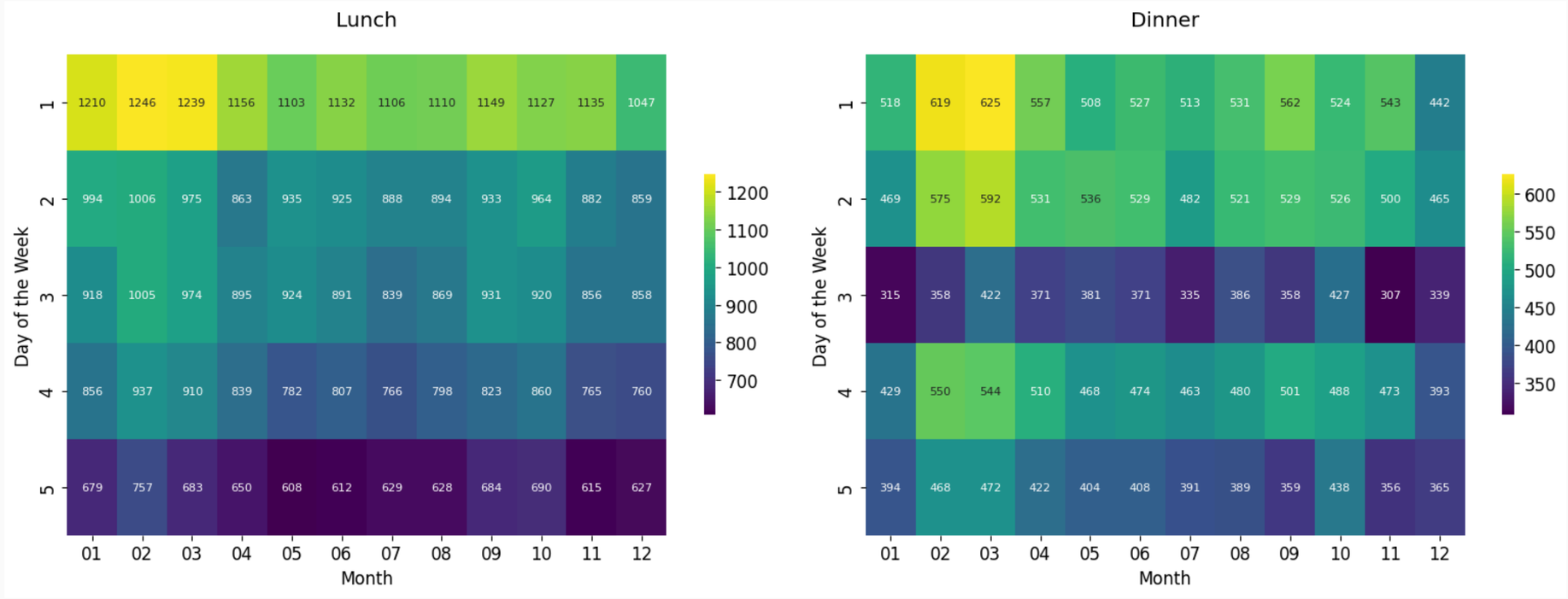
변수간의 상관관계 분석



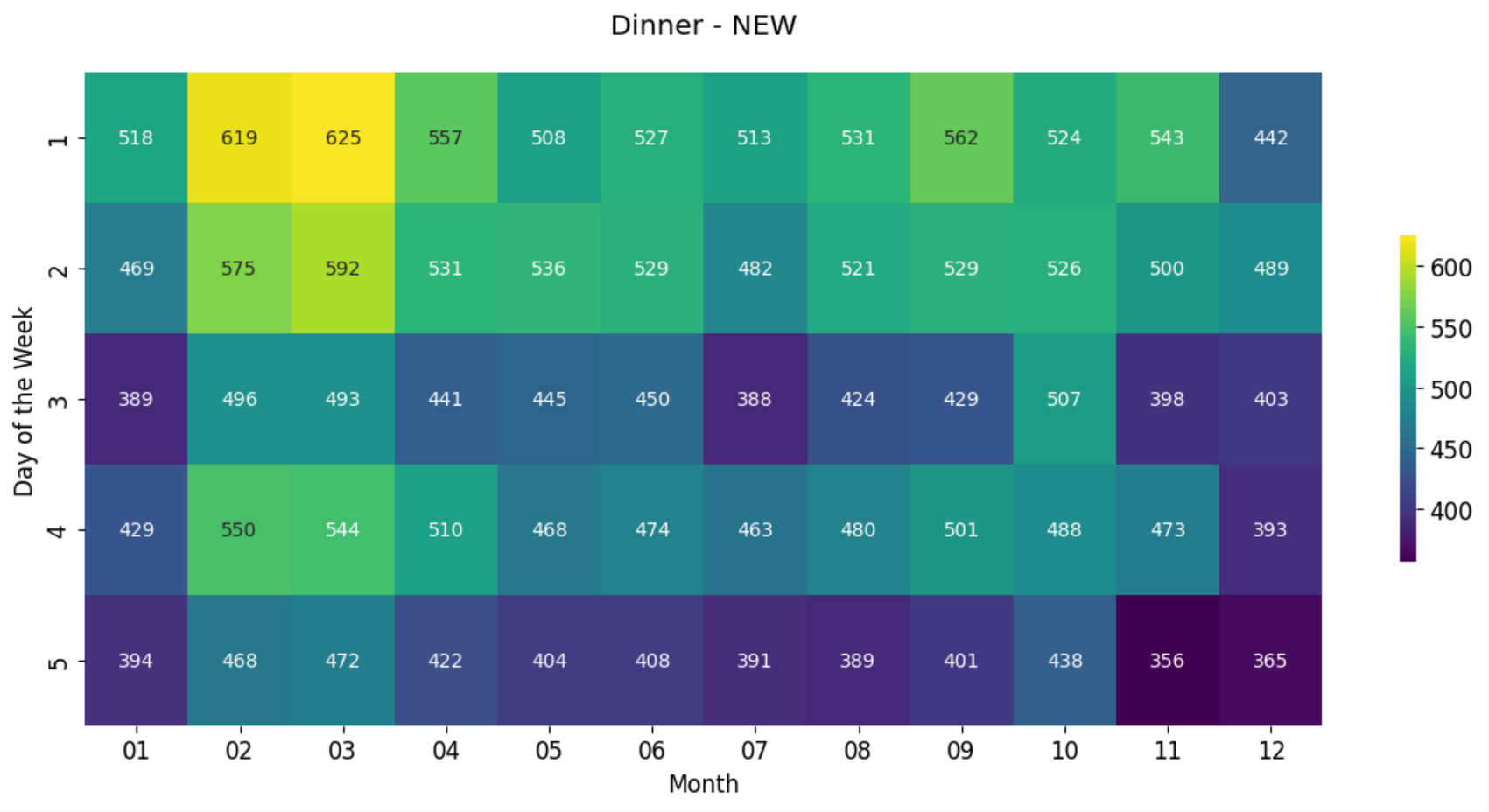
간단한 선점도


lunch
dinner
total_p
vacationer
business_t
night_worker
home_worker
1500
1250
1000
750
500
250
0
2600
2800
3000
3200
100
200
300
400

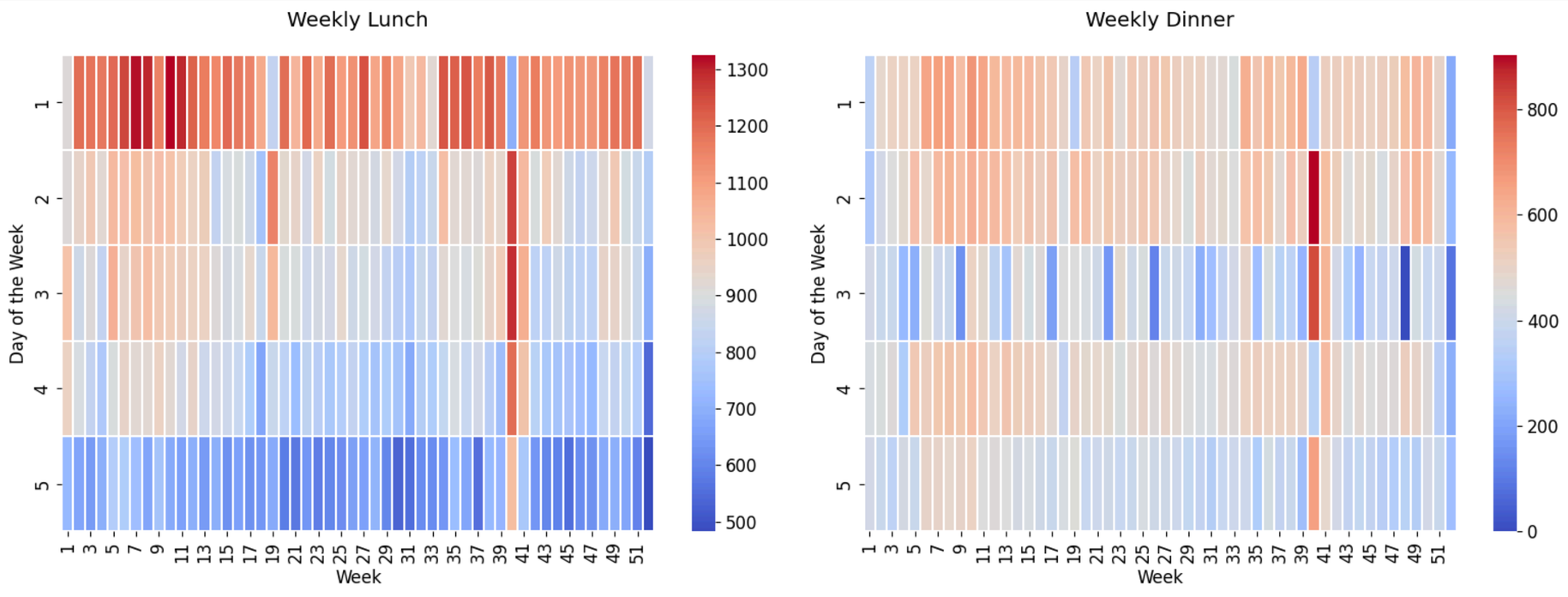
월별 요일별 패턴 분석


Lunch
Day of the Week
Month
1 1210 1246 1239 1156 1103 1132 1106 1110 1149 1127 1135 1047
2 994 1006 975 863 935 925 888 894 933 964 882 859
3 918 1005 974 895 924 891 839 869 931 920 856 858
4 856 937 910 839 782 807 766 798 823 860 765 760
5 679 757 683 650 608 612 629 628 684 690 615 627
01 02 03 04 05 06 07 08 09 10 11 12
1200 1100 1000 900 800 700
Dinner
Day of the Week
Month
1 518 619 625 557 508 527 513 531 562 524 543 442
2 469 575 592 531 536 529 482 521 529 526 500 465
3 315 358 422 371 381 371 335 386 358 427 307 339
4 429 550 544 510 468 474 463 480 501 488 473 393
5 394 468 472 422 404 408 391 389 359 438 356 365
01 02 03 04 05 06 07 08 09 10 11 12
600 550 500 450 400 350

저녁만 다시 시각화

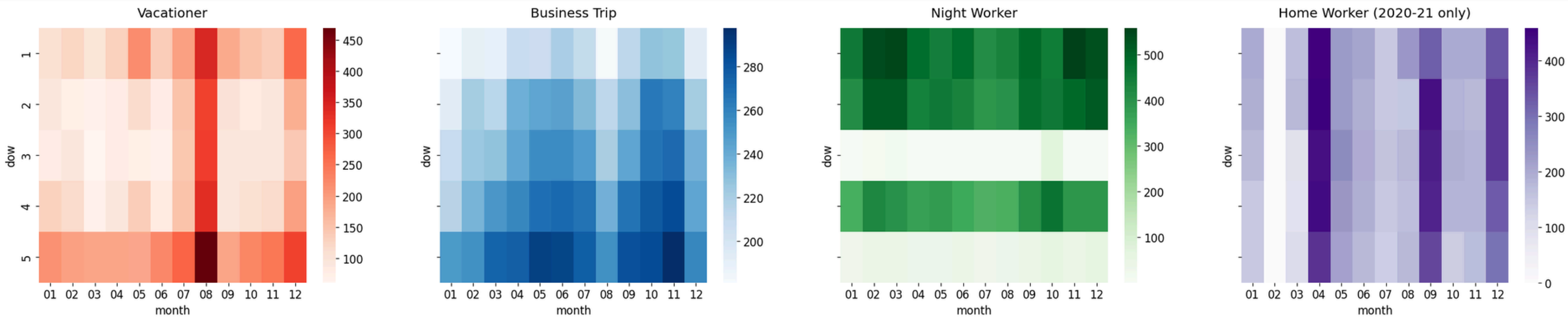

Dinner - NEW
Day of the Week
1
2
3
4
5
518 619 625 557 508 527 513 531 562 524 543 442
469 575 592 531 536 529 482 521 529 526 500 489
389 496 493 441 445 450 388 424 429 507 398 403
429 550 544 510 468 474 463 480 501 488 473 393
394 468 472 422 404 408 391 389 401 438 356 365
01 02 03 04 05 06 07 08 09 10 11 12
Month
600
550
500
450
400

주별 요일별 패턴 분석

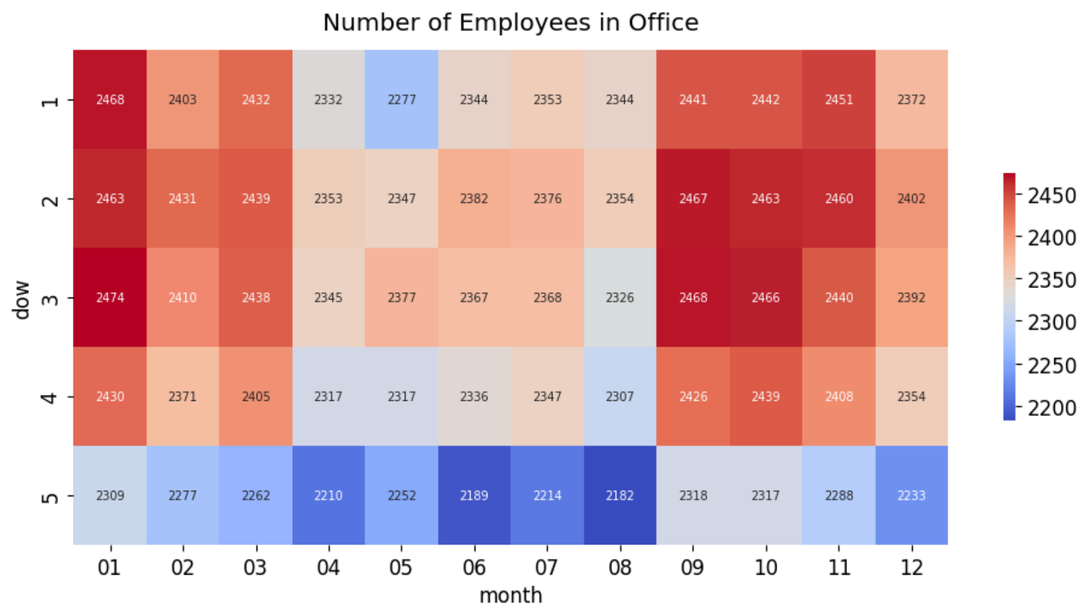

Weekly Lunch
Day of the Week
Week
1300
1200
1100
1000
900
800
700
600
500
Weekly Dinner
Day of the Week
Week
800
600
400
200
0

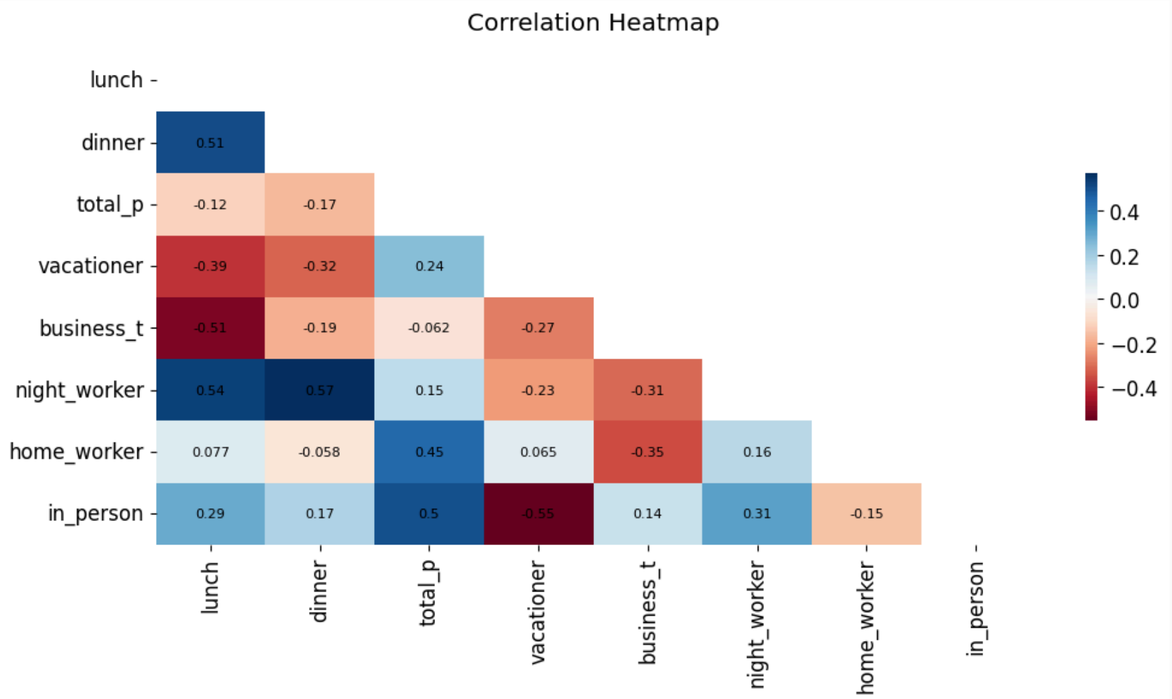
회사내에 있는 직원수 = 본사정원수 - (휴가자수 + 출장자수 + 재택근무자수)

회사에 있는 직원수 월별&요일별

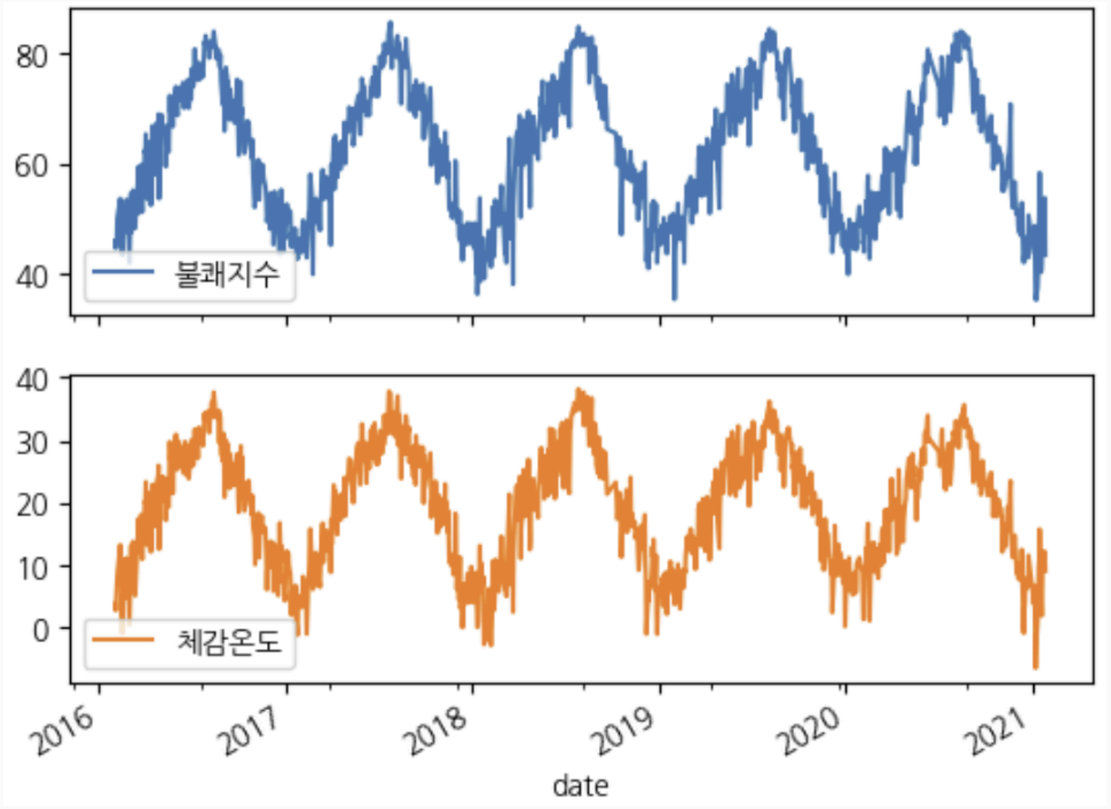


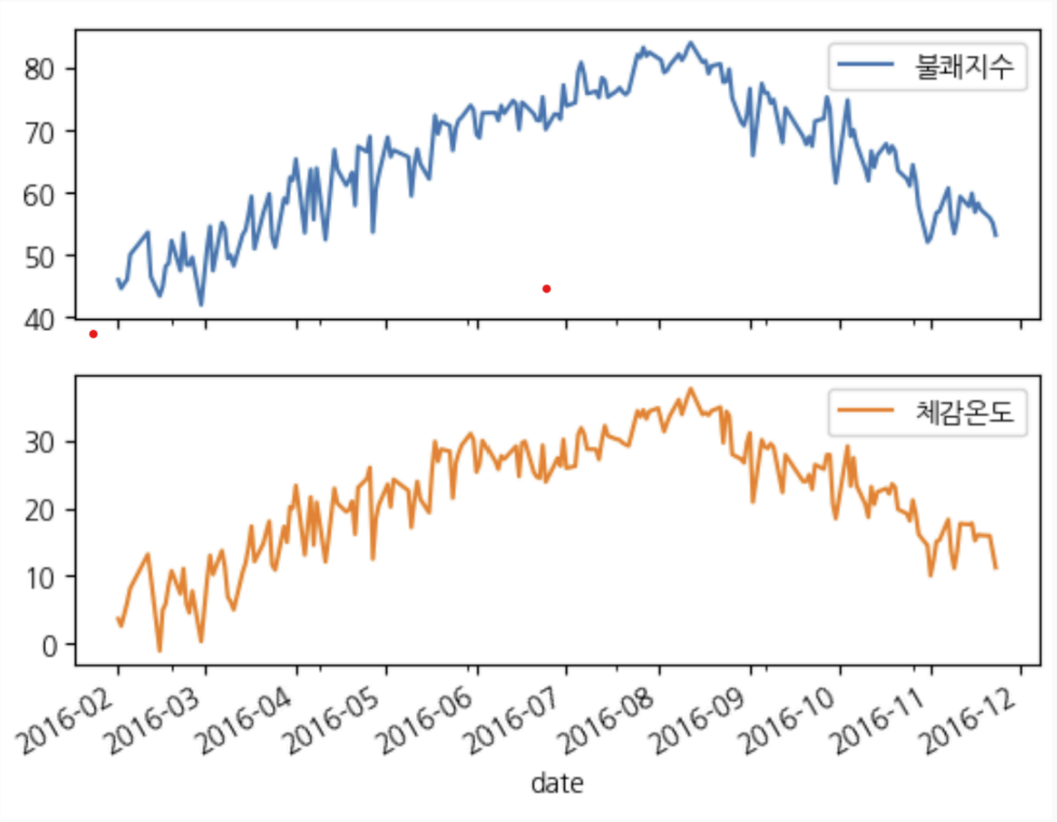
회사내 직원수 히트맵 및 변수간의 상관관계



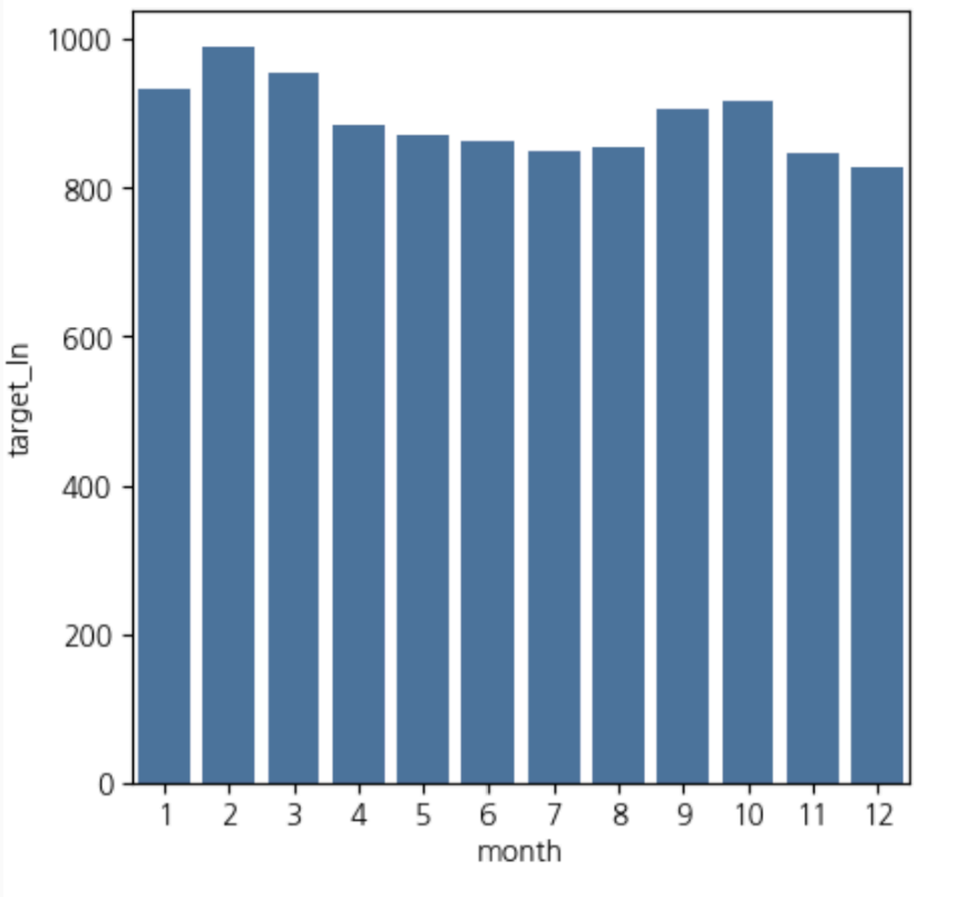


불쾌지수 & 체감온도 시계열 시각화

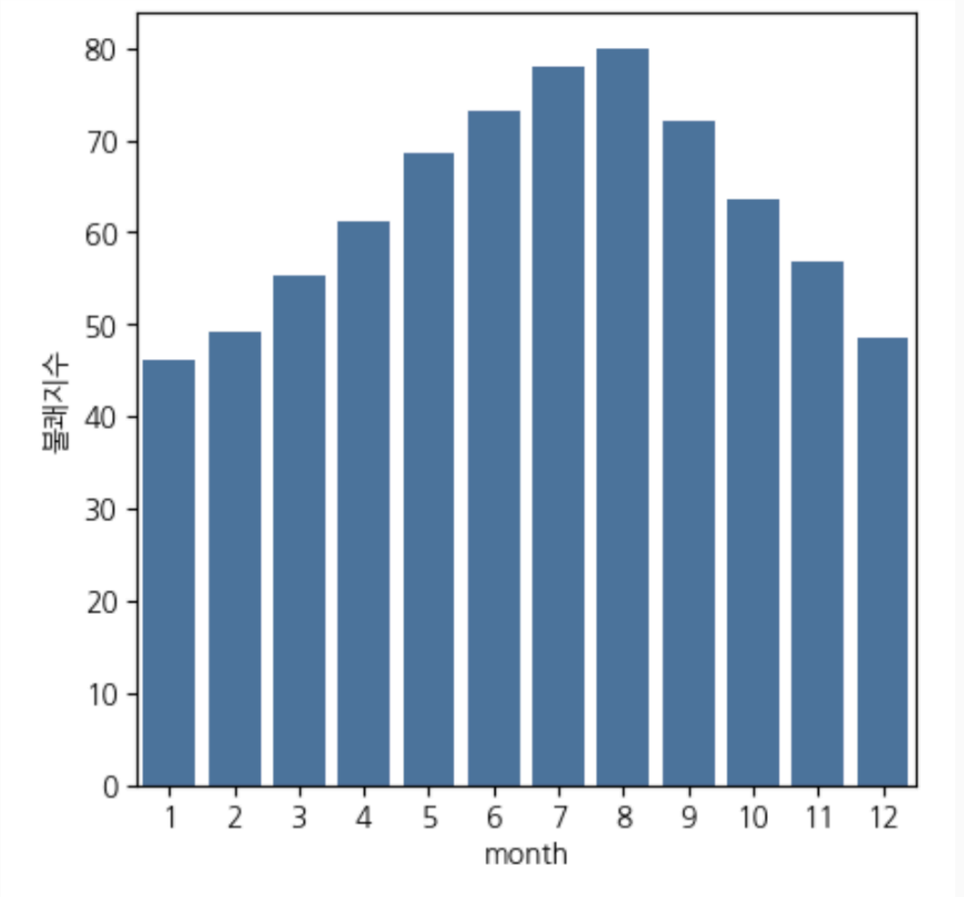




월별 중식계 평균과 불쾌지수 평균

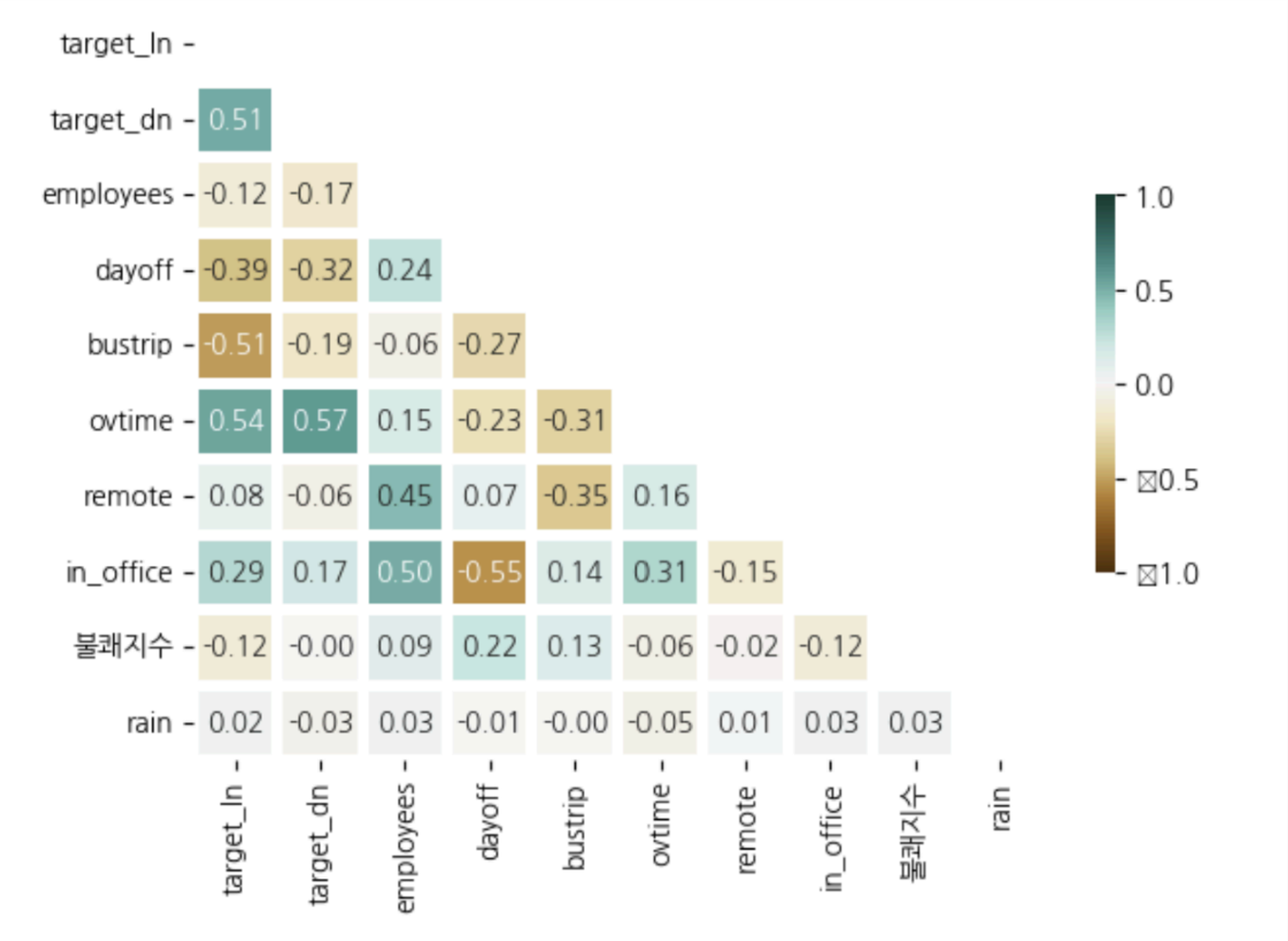


월별 중식계 평균

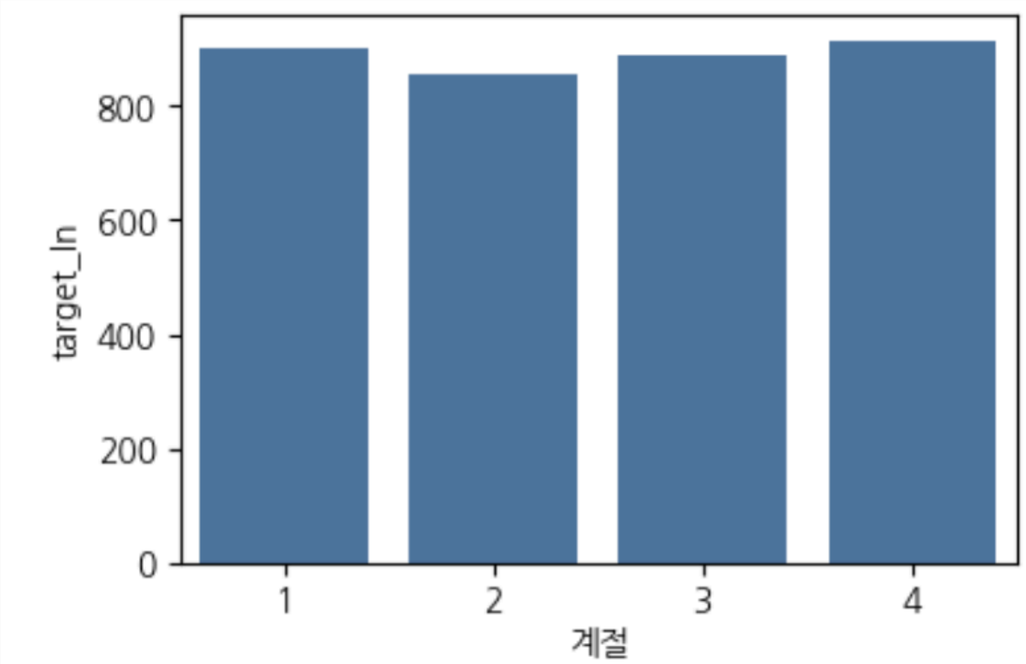


월별 불쾌지수 평균

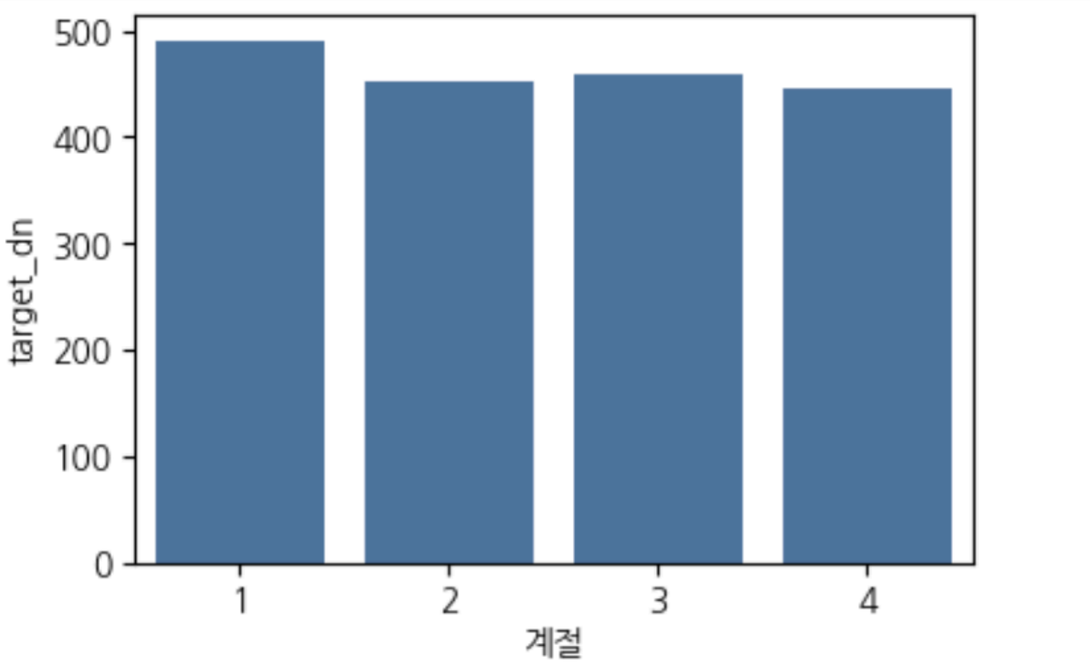
불쾌지수 & 비의 상관관계 분석



계절별 중식계와 석식계 평균

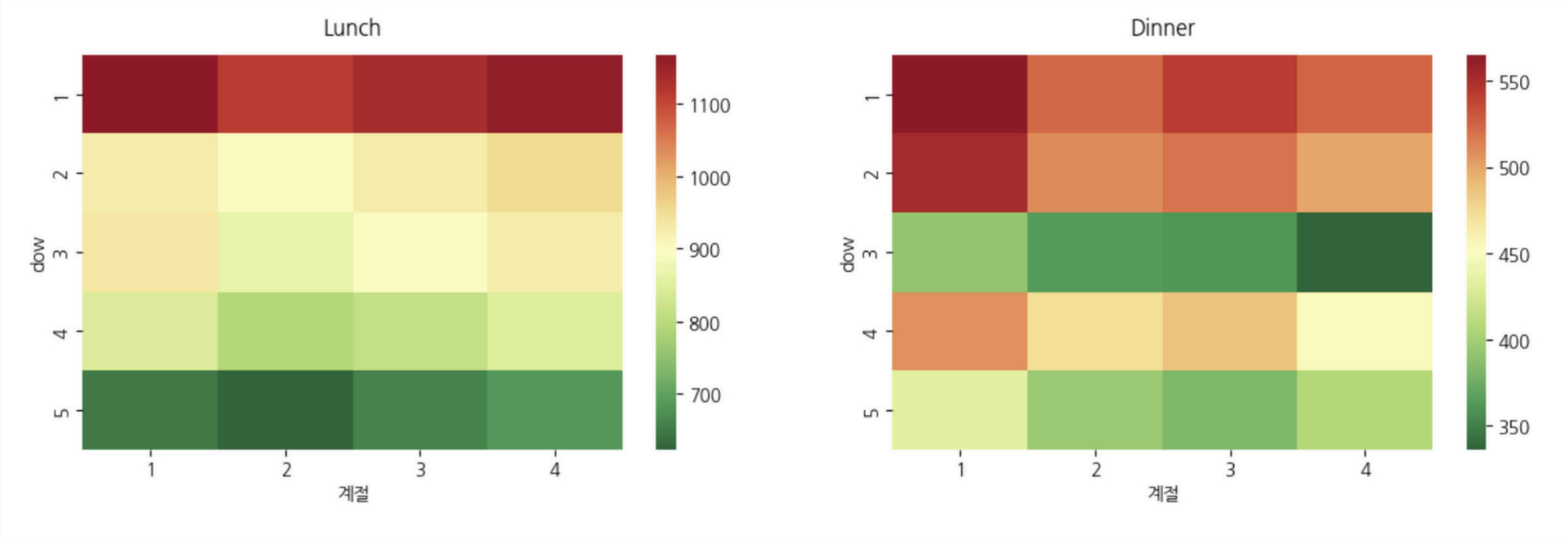


계절별 중식계 평균



계절별 석식계 평균

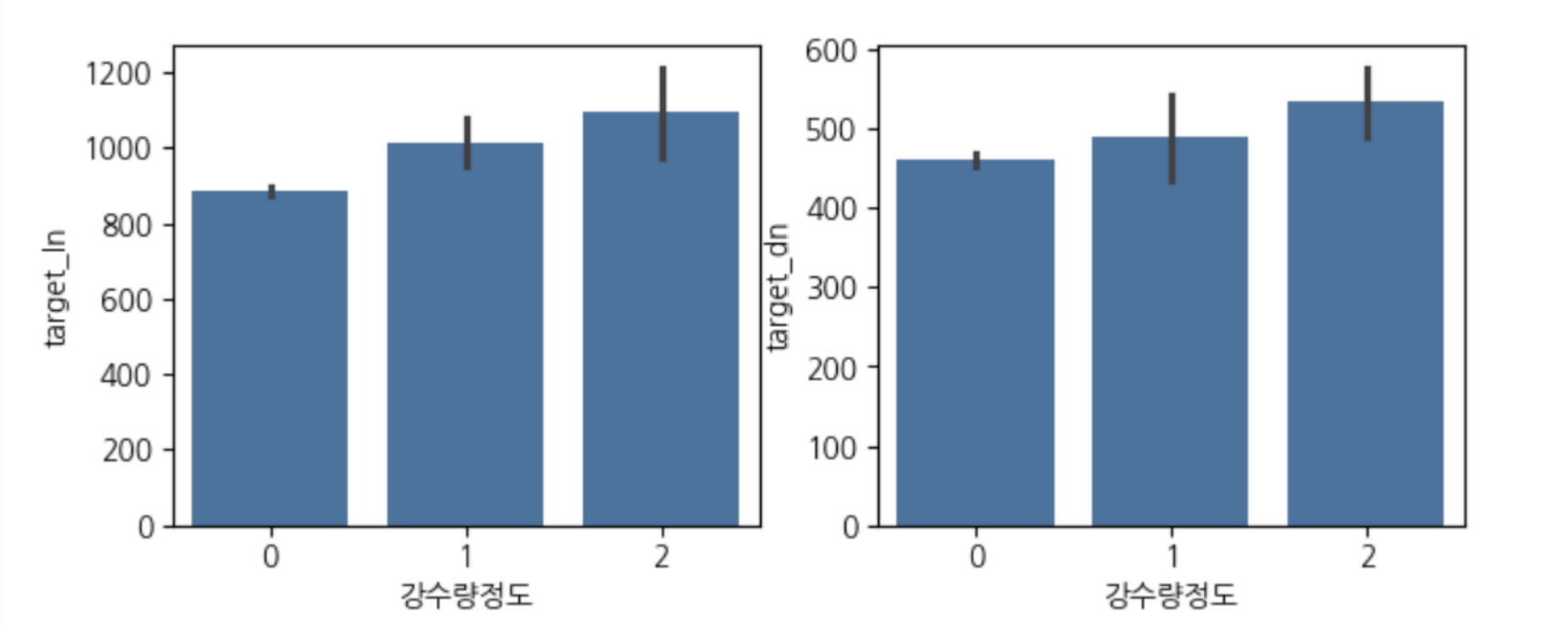
계절과 요일에 따른 중식계와 석식계 평균



강수량 정도별 중식계 평균

강수량 정도별 석식계 평균

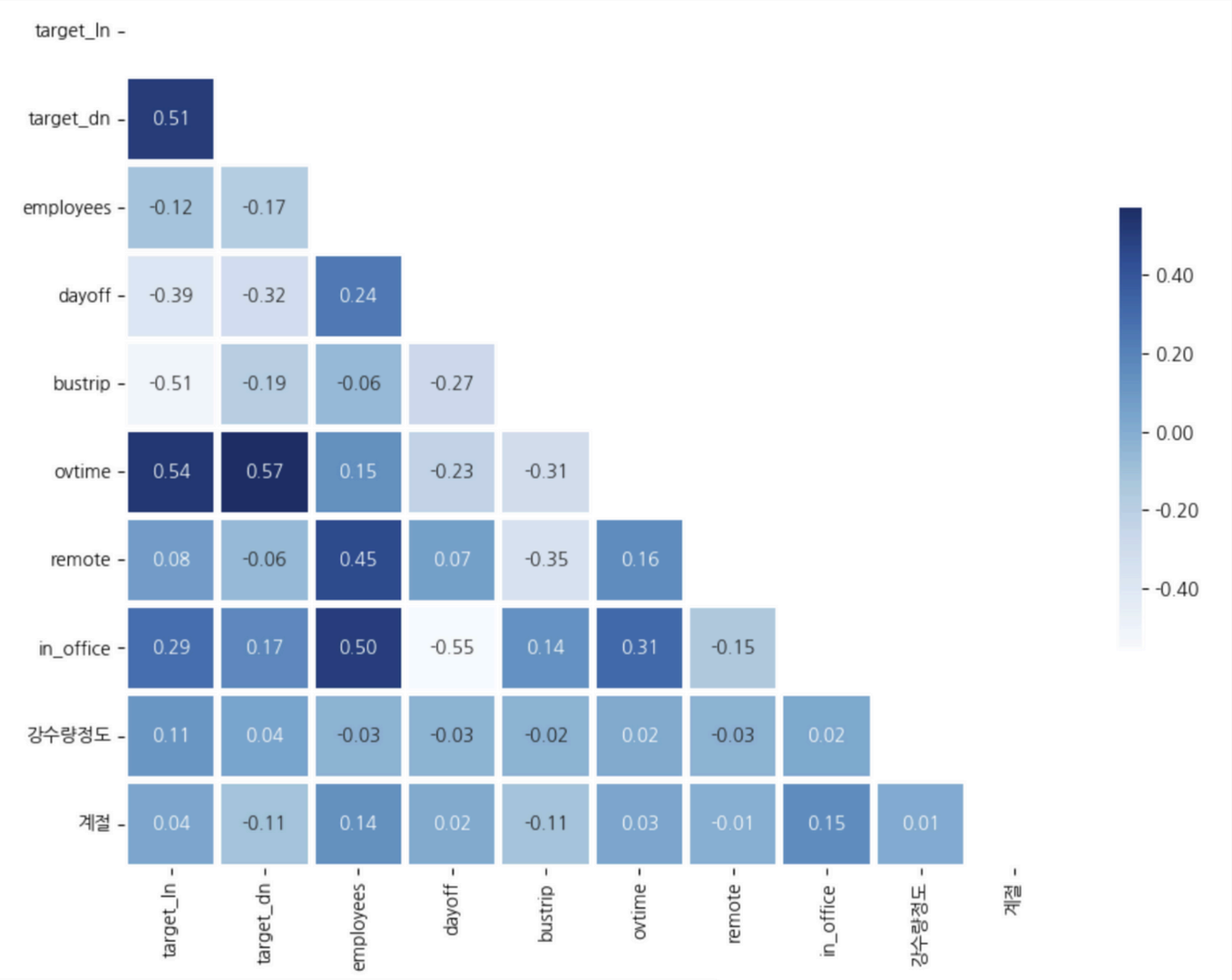
강수량 정도별 중식계와 석식계 평균



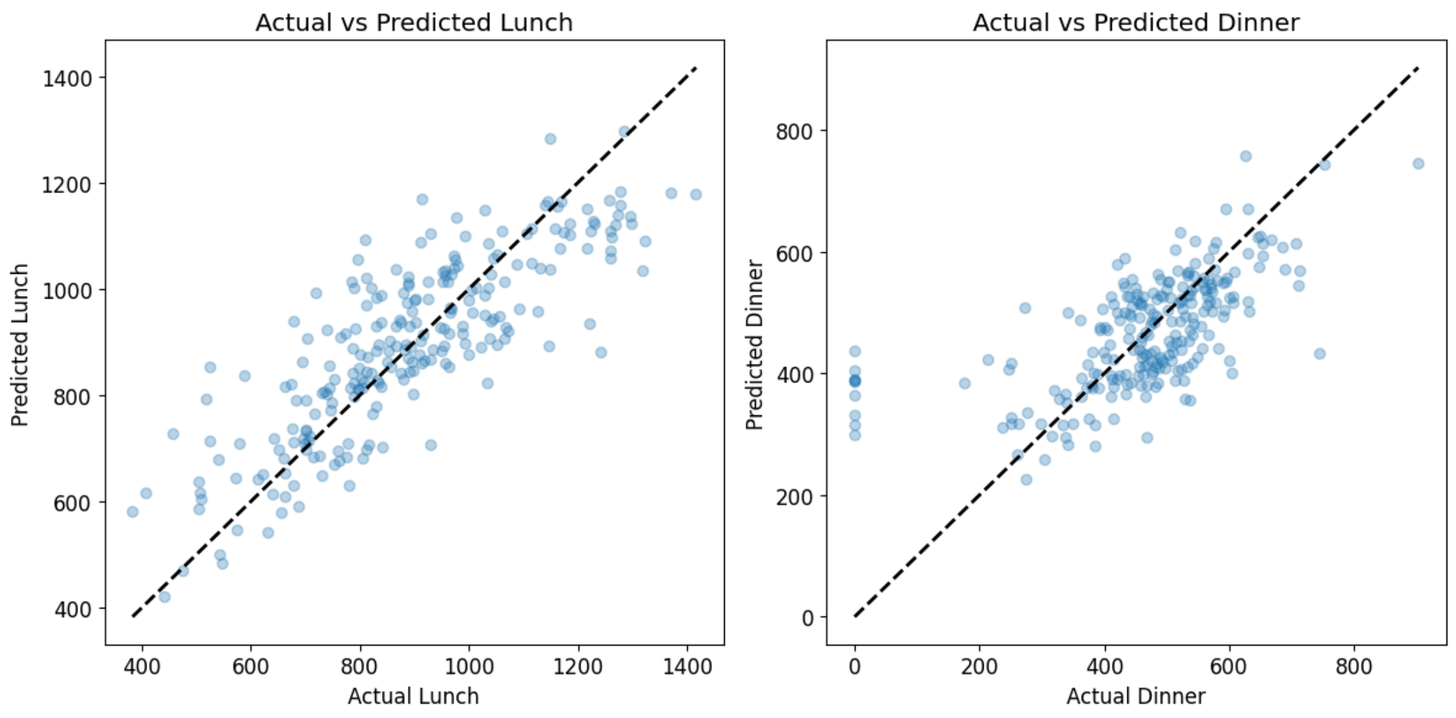
강수량 정도별 중식계 평균

강수량 정도별 석식계 평균

강수량정도 & 계절 변수의 상관관계 분석

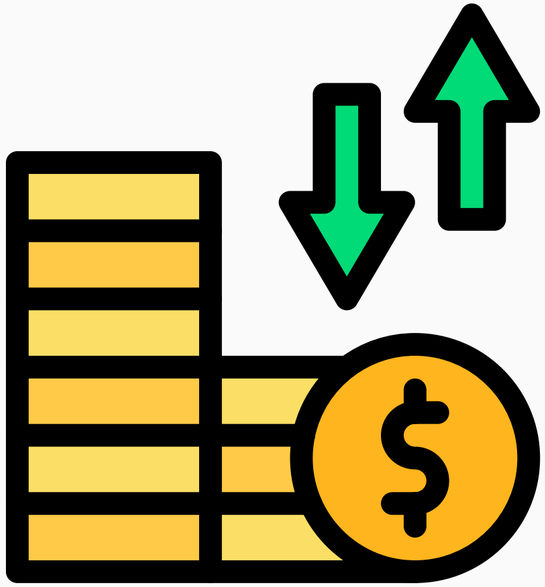


모델 성능결과



Lunch RMSE: 116.07876254981991
Lunch R^2: 0.6819167861816471
Dinner RMSE: 108.67797049744767
Dinner R^2: 0.40448264488756813

결론



참고 논문

고려대학교 - 머신러닝을 이용한 단체급식 다중코너 식수 예측 모델 연구
경희대학교 - 기계학습을 활용한 집단 급식소의 食數 豫測 모델 開發 연구
-S 시청 구내 직원 식당의 실데이터를 기반으로-

THANK YOU