서울시 따릉이 대여량 예측 프로젝트

BRAIN 2조 고민서, 김은교, 송민수, 신민호

목차

1. 주제 설명 및 목표
2. 베이스라인 코드 설명
3. 모델 설명
4. 데이터 설명
5. 변수에 대한 EDA 결과
6. 차후 계획
7. 질의응답

1. 주제 설명 및 팀 목표

- 진행 대회: Dacon의 ‘따릉이 대여량 예측 프로젝트’
  - 서울 시의 공공 자전거인 따릉이 대여 수를 예측하는 프로젝트
- 기간: ~5월 13일
- 목표: 베이스라인 코드 대로 진행한 예측 모델의 성능보다 더 좋은 결과 도출하기
- 1차 진행 상황
  - 베이스라인 코드 분석
  - 팀원 별로 변수를 2개씩 할당하여 각 변수에 대해 EDA 진행해보기

2. 베이스라인 코드 설명

```
# 라이브러리 불러오기
import pandas as pd # import as : 라이브러리를 불러오고 이를 as이하 이름으로 명명함
                    # pandas : 데이터 분석시 표형식의 데이터를 쉽쉽게 다룰수있도록 도움
import numpy as np  # numpy : 수치 계산과 배열 처리를 빠르게 수행하는 라이브러리
from sklearn.linear_model import LinearRegression # scikit-learn이라는 라이브러리에
                                                  # 이렇게 가져오는 이유는 sklearn는
                                                  # import sklearn만 하면 실제 모델

from sklearn.model_selection import train_test_split #scikit-learn라이브러리의 mod
                                                     # 왜냐하면 머신러닝 모델을 사용

from sklearn.metrics import mean_absolute_error #scikit-learn라이브러리의 머신러닝
```

〈사용한 라이브러리〉

- Pandas: csv형태의 데이터를 데이터프레임 형태로 읽어들이기 위함
- Numpy: 데이터 수치 계산과 배열처리
- sklearn.linear_model의 LinearRegression : 선형회귀모델 사용
- sklearn.model_selection의 train_test_split : train set와 test set을 나눠주기 위한 함수

  → 해당 코드에서는 train set에서 valid set을 떼어 내는 작업을 하기 위해 사용
- sklearn.metrics의 mean_absolute_error : 머신러닝 모델의 평가를 위해 절대 평균 오차 사용

- 전처리 방법:
  1. X(사용할 feature) : date time, number of rentals 제외한 칼럼 선택
     - Date time: object형으로 되어있어 제외
     - Number of rentals: 예측 대상인 값이므로 제외
  2. Y(target) : number of rentals으로 지정
  3. Train set를 valid set과 train set 으로 나눔 (valid set으로 성능 평가를 위함)

- 베이스라인 코드에서 사용한 모델: LinearRegression(선형회귀 모델)

- 평가 지표: NMAE(절대 평균 오차)로 계산

$$\mathrm{NMAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n \times (y_{\max} - y_{\min})}$$

n: 데이터 샘플의 개수

y_i: 실제 값 (Ground truth)

$\hat{y}_i$: 예측 값

$y_{\max}$: 실제 값의 최댓값

$y_{\min}$: 실제 값의 최솟값

✓예측 값의 단위에 영향을 받지 않고 모델 성능을 비교해야 할 때 많이 사용

3. 모델 설명

1. 선형회귀
2. XGBoost
3. 랜덤 포레스트

선형회귀 모델

$$y = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

특성 x에 대한 각각의 **가중치 w**와 **절편 b** 학습

- 장점
  - 데이터가 선형성을 가질 때 좋은 성능을 가짐
  - 연속적인 수치를 예측할 때 주로 사용
  - 직관적으로 해석 가능
  - 빠르고 효율적
  - 가중치 값을 통해 특성의 중요도 확인 가능
- 단점
  - 비선형적 관계에서 성능이 낮음
  - 데이터의 스케일, 이상치에 민감
  - 차원이 너무 많을 때 overfitting(과적합) 문제 발생할 수 있음
  - 다중공선성 문제

    :변수들 간 높은 상관관계를 가질 때 모델의 안정성과 성능 저하

XGBoost

특징

- Gradient Boosting 기반: 여러 약한 트리를 순차적으로 학습하여 오류를 보완하는 부스팅 구조
- 정규화 내장 (L1/L2): 모델 복잡도 제어로 과적합을 효과적으로 방지
- 결측값 자동 처리: 결측값이 있어도 별도 전처리 없이 최적 분기 방향을 자동 탐색

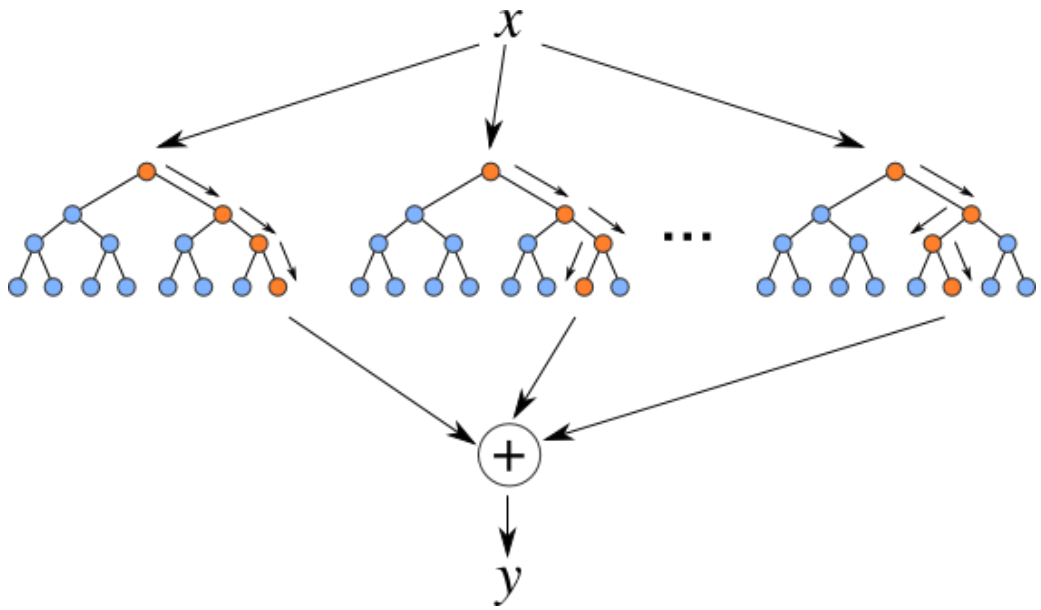
장점

- 예측 성능 우수
- 과적합 방지
- 빠른 학습속도

단점

- 낮은 해석력
- 하이퍼파라미터 튜닝복잡
- 높은 메모리사용

랜덤 포레스트 장단점



장점	높은 정확도	과적합 방지	다양한 데이터에 적용 가능	특성 중요도 평가	병렬 처리 가능
단점	모델 해석 어려움	메모리 사용량 많음	느린 예측 속도	과다한 트리 수로 인한 처리 속도 저하	특성 간 상호작용을 반영하기 어려움

Image 출처: https://medium.com/@vk.viswa/surface-level-understanding-of-random-forest-regression-3bac37e06f3e

모델 선택

- 베이스라인 코드에서 예측할 변수가 이산형 수치변수에 해당하므로, 선형회귀 모델을 사용

우리가 예측하려는 값은 '따릉이 대여 수' 이므로, 수치형 데이터에 해당함.

→ baseline 코드에서는 주어진 feature(x)과 target(y)인 대여 수 간의 관계를 학습하여 예측 값과 실제 값의 차이가 최소가 되는 line을 학습하는 선형회귀 모델을 선택하였음.

- 현재 데이터가 복잡한 상호작용이나 비선형 관계를 포함할 수 있기때문에 XGBoost나 랜덤 포레스트 모델을 선택하는 것이 좋다고 생각

=〉 차후 비교를 통해 성능이 더 좋은 모델을 선택할 예정

4. 데이터 설명

변수 설명

- **date_time:** 일별 날짜(18~20년까지 3개년의 4~6월 데이터)
- **Wind_direction:** 풍향(degree)
- **sky_condition:** 하늘 상태

  2019년 6월 4일까지 : 맑음(1), 구름조금(2), 구름많음(3), 흐림(4)

  2020년 : 맑음(1), 구름많음(3), 흐림(4)
- **precipitation_form:** 강수형태

  없음(0), 비(1), 진눈깨비(2), 눈(3), 소나기(4)
- **wind_speed:** 풍속 (m/s)
- **Humidity:** 습도(%)
- **low_temp:** 최저 기온 (`C)
- **high_temp:** 최고 기온 (`C)
- **Precipitation_Probability:** 강수 확률 (%)
- **number_of_rentals:** 따릉이 대여량 → **예측할 변수!**

변수별 자료형 확인

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 273 entries, 0 to 272
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   date_time                   273 non-null    object 
 1   wind_direction              273 non-null    float64
 2   sky_condition               273 non-null    float64
 3   precipitation_form          273 non-null    float64
 4   wind_speed                  273 non-null    float64
 5   humidity                    273 non-null    float64
 6   low_temp                    273 non-null    float64
 7   high_temp                   273 non-null    float64
 8   Precipitation_Probability   273 non-null    float64
 9   number_of_rentals           273 non-null    int64  
dtypes: float64(8), int64(1), object(1)
memory usage: 21.5+ KB
None
```

df.info()로 자료형 확인

✓결과

Date_time: object형으로 되어 있음

→datetime 형식으로 변환 후 분석 진행

Number_of_rentals: int형

그 외의 feature들은 float형으로 되어 있어 그대로 사용

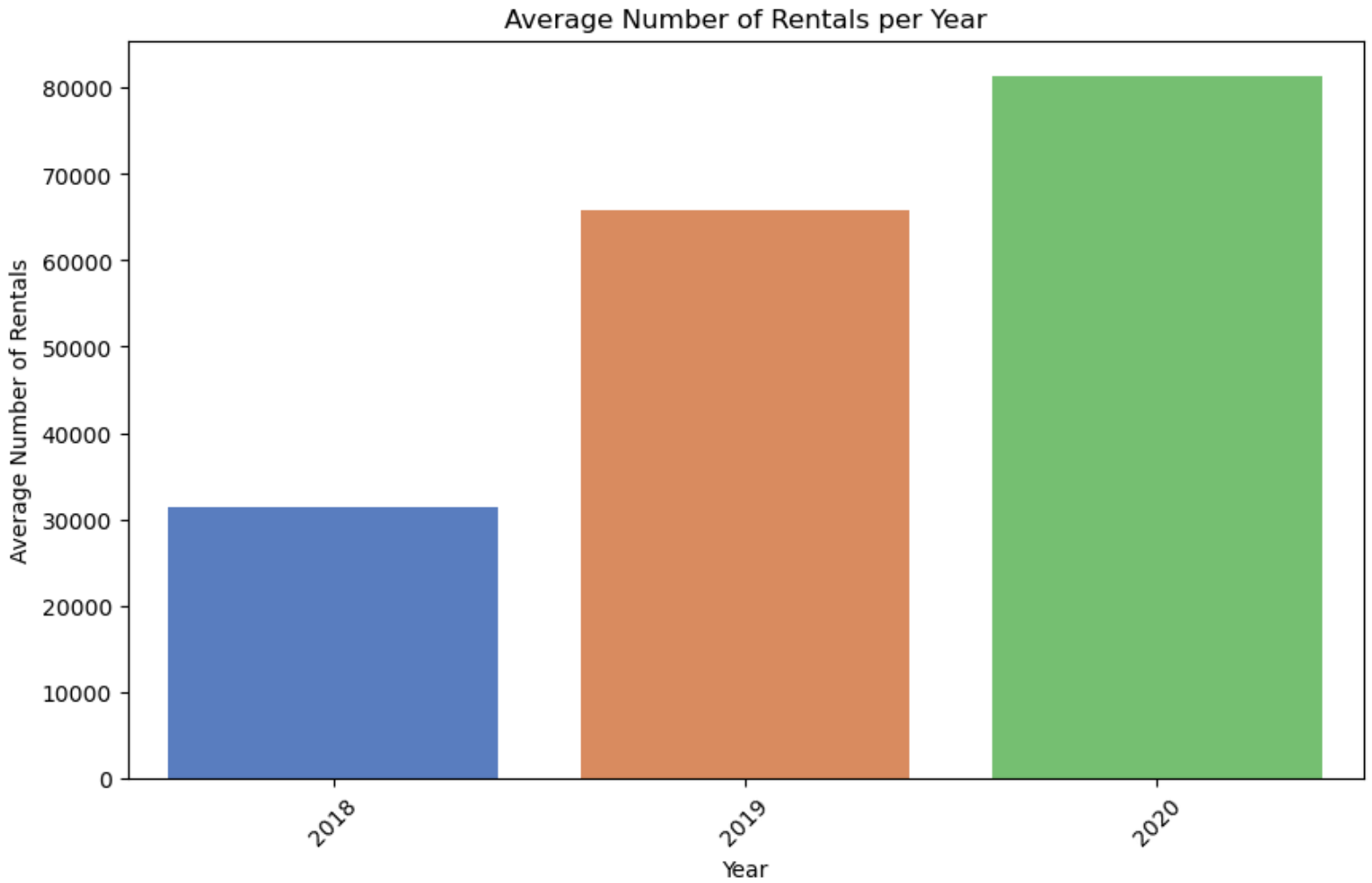
결측치 조사

```
date_time                    0
wind_direction               0
sky_condition                0
precipitation_form           0
wind_speed                   0
humidity                     0
low_temp                     0
high_temp                    0
Precipitation_Probability    0
number_of_rentals            0
dtype: int64
```

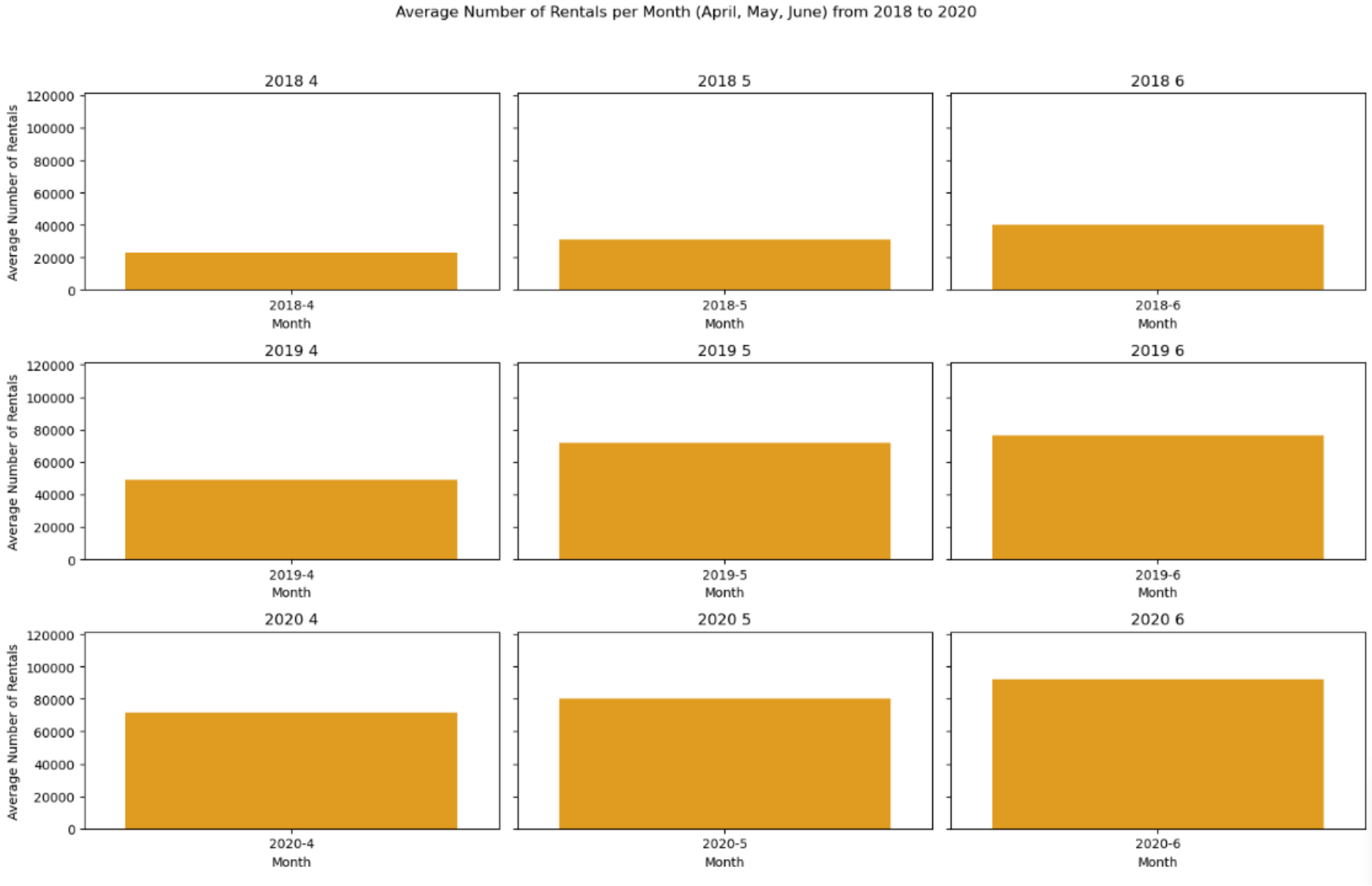
df.isnull()로 결측치 확인
✓결과: 없음

5. 각 변수 별 EDA 결과

연도별 자전거 대여량 평균



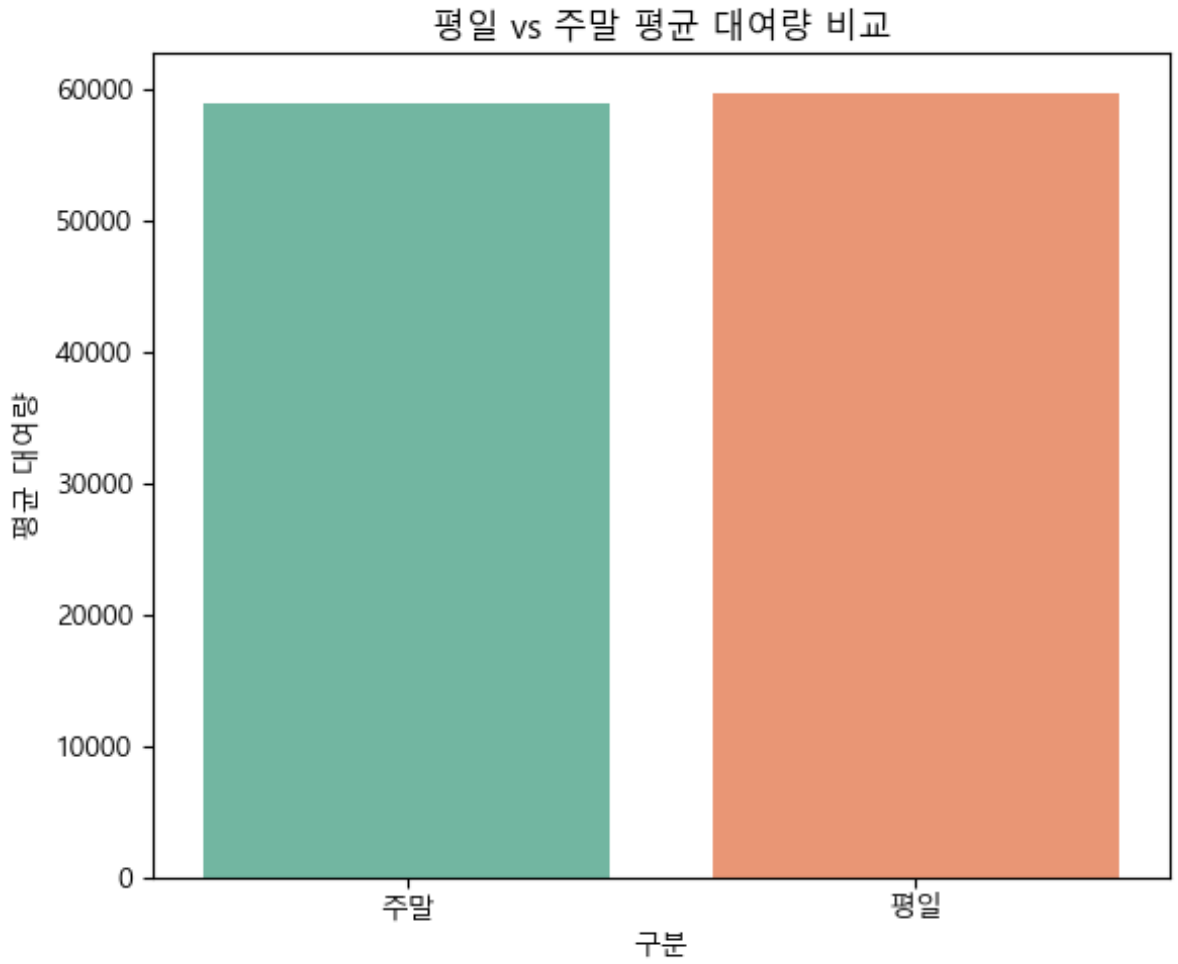
월별 자전거 대여량 평균



-매년 마다 대여량 수는 증가하는 추세를 보임

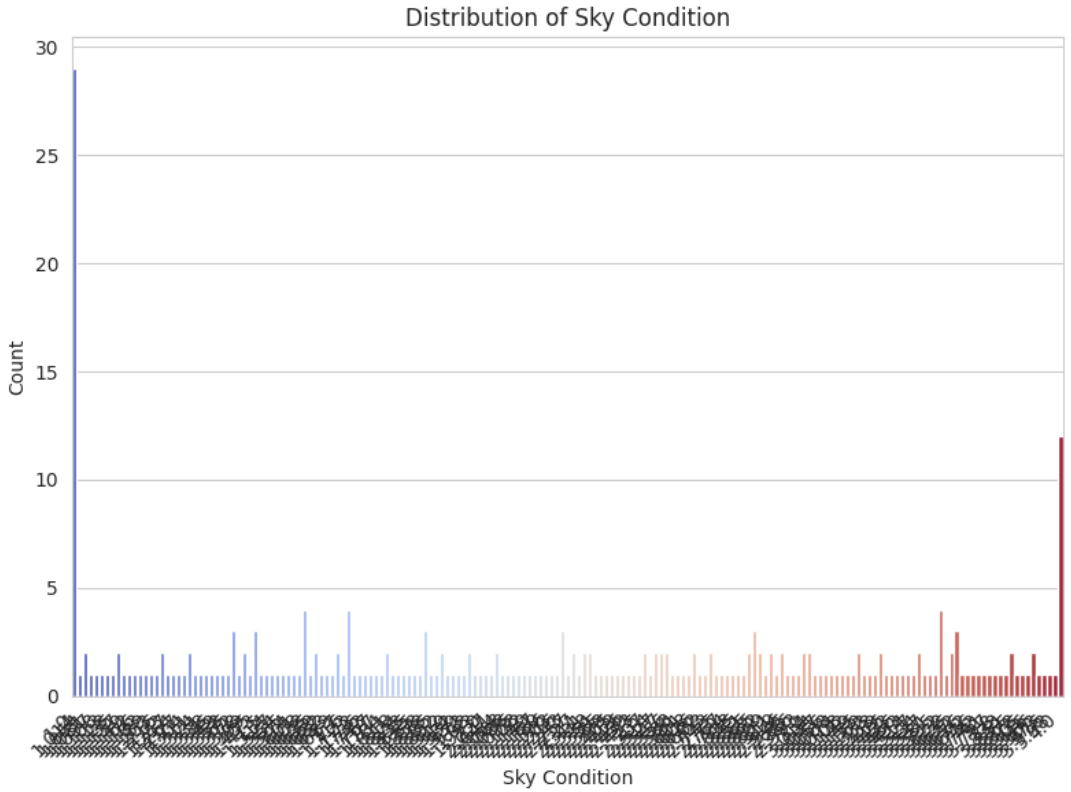
-4월에서 6월로 갈 수록 대여량이 점차 증가하는 양상을 보임

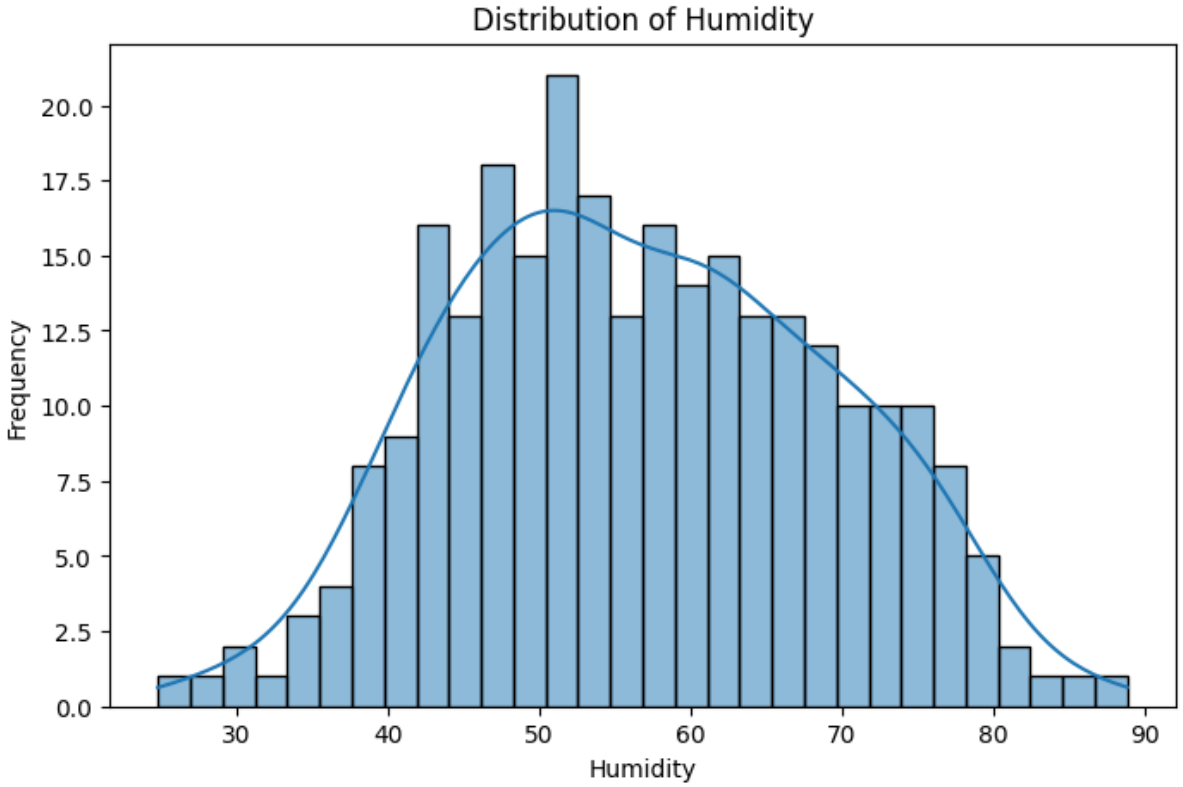
평일과 주말 평균 대여량 비교



✓ 유의미한 차이가 나타나지 않는 것으로 확인

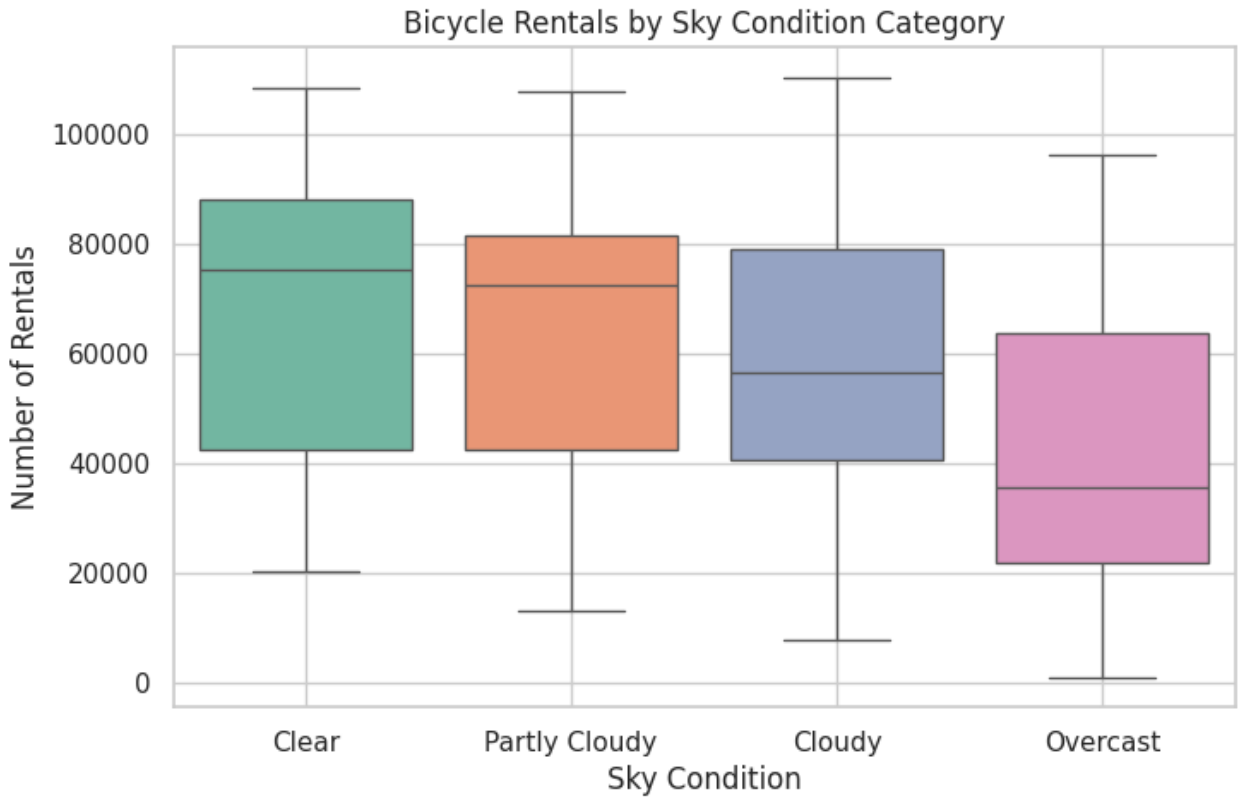
Sky condition/humidity 분포도



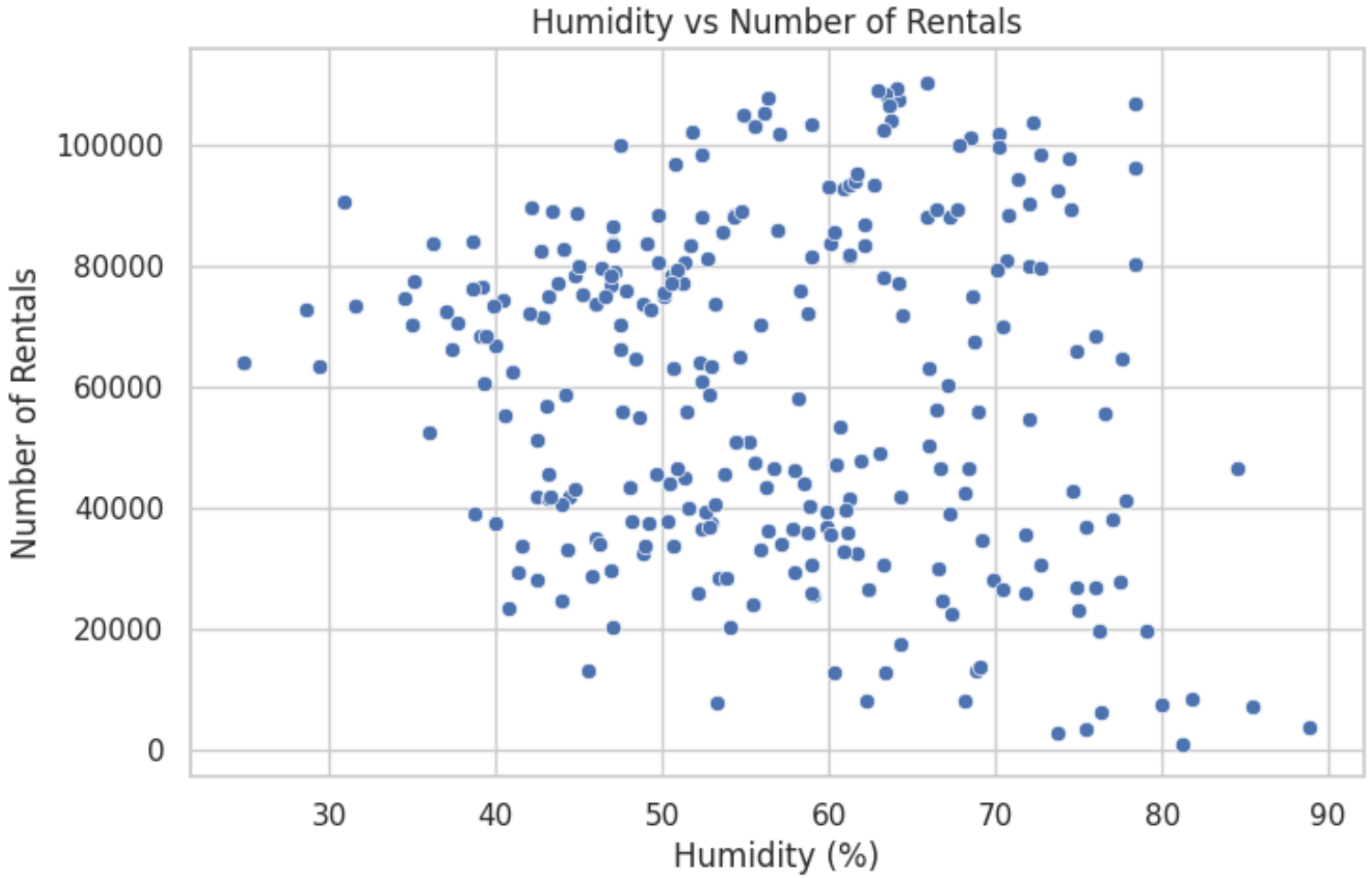


하늘 상태와 대여량 (그룹화) - .head() 사용

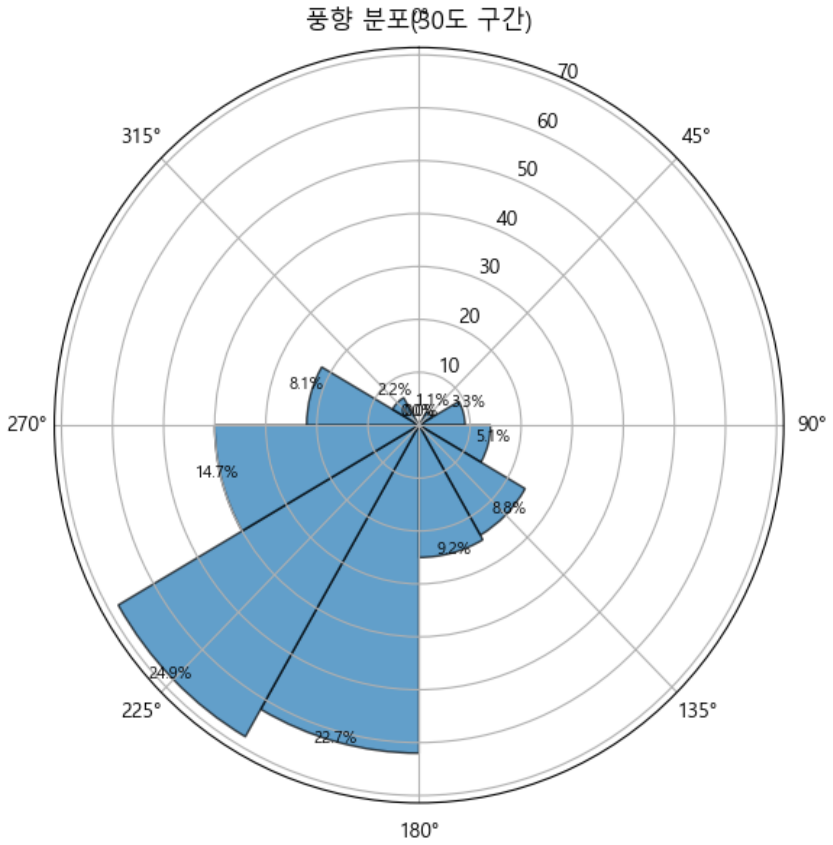
	sky_condition	sky_bin
0	4.000	Overcast
1	2.950	Cloudy
2	2.911	Cloudy
3	3.692	Overcast
4	4.000	Overcast



습도와 대여량 산점도


Humidity vs Number of Rentals
Number of Rentals
100000
80000
60000
40000
20000
0
30
40
50
60
70
80
90
Humidity (%)

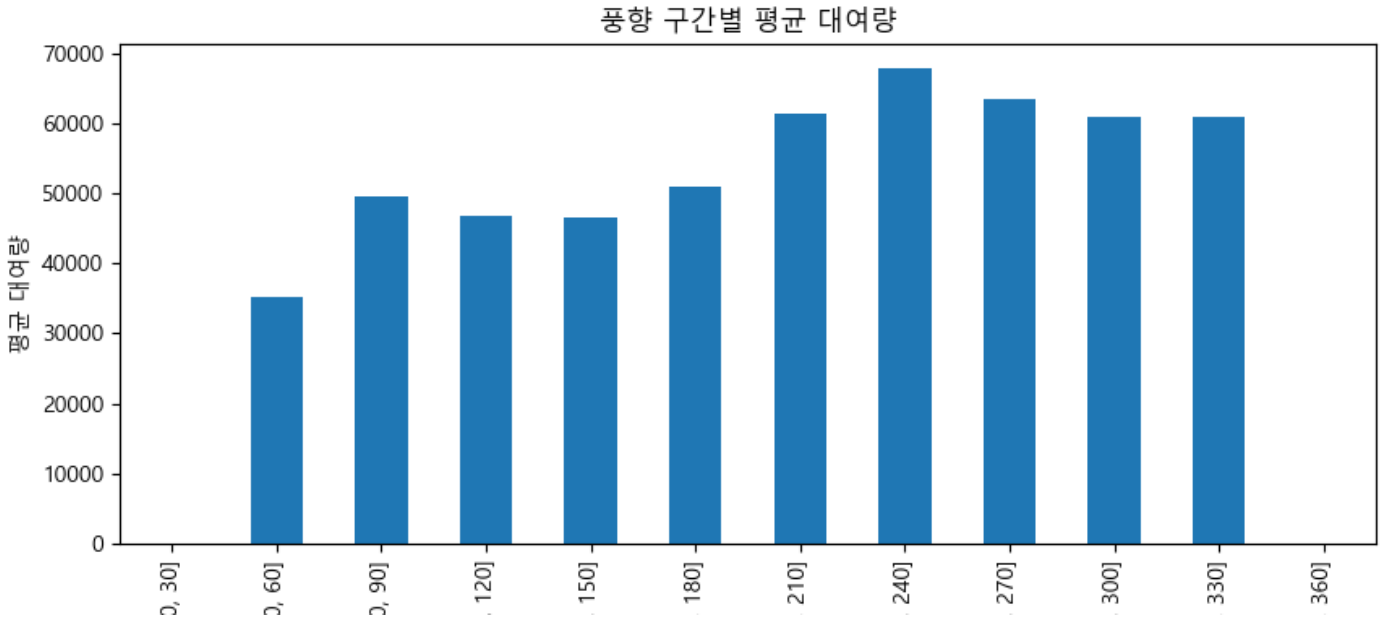
풍향 데이터 분포

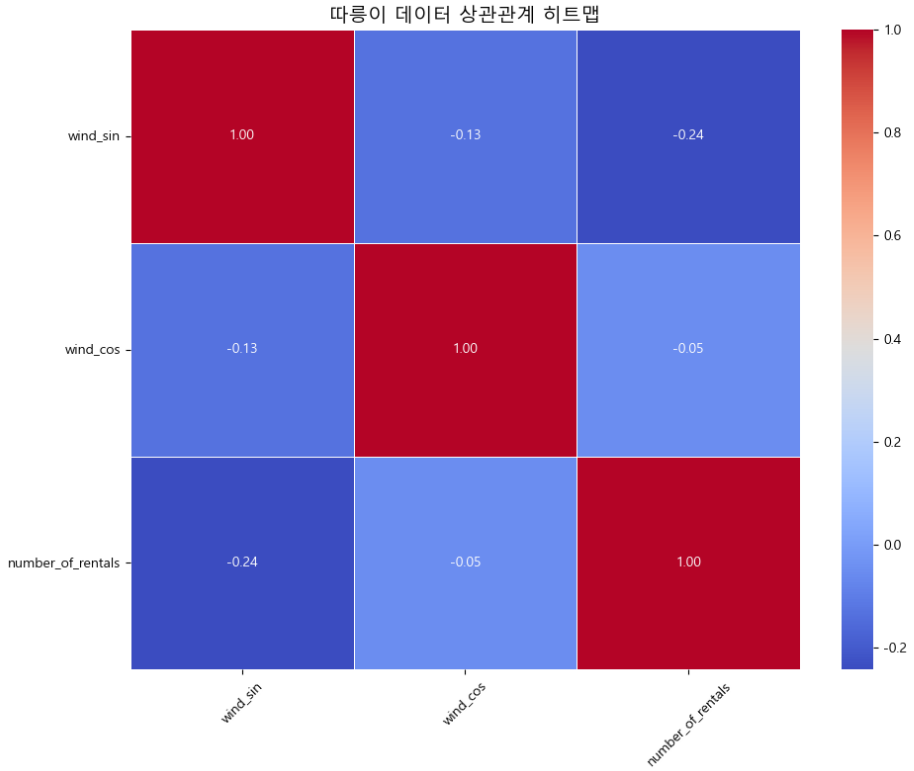


풍향 구간	빈도 수	퍼센트 (%)
0°-30°	0	0.0
30°-60°	3	1.1
60°-90°	9	3.3
90°-120°	14	5.13
120°-150°	24	8.79
150°-180°	25	9.16
180°-210°	62	22.71
210°-240°	68	24.91
240°-270°	40	14.65
270°-300°	22	8.06
300°-330°	6	2.2
330°-360°	0	0.0

남서풍, 남풍, 서풍의 비율이 약 60%정도를 차지함 → 4~6월 봄~초여름 풍향의 특징

풍향 구간 별 평균 대여량 조사



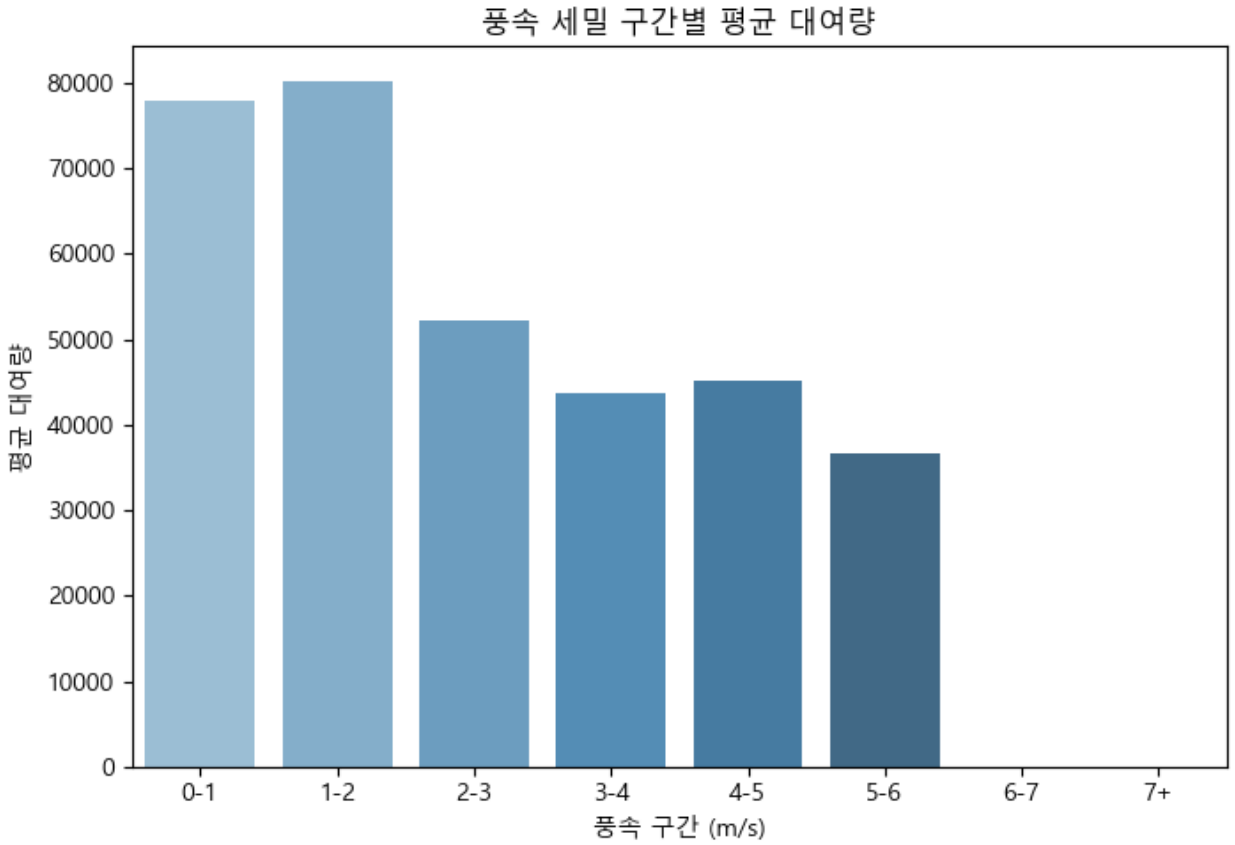


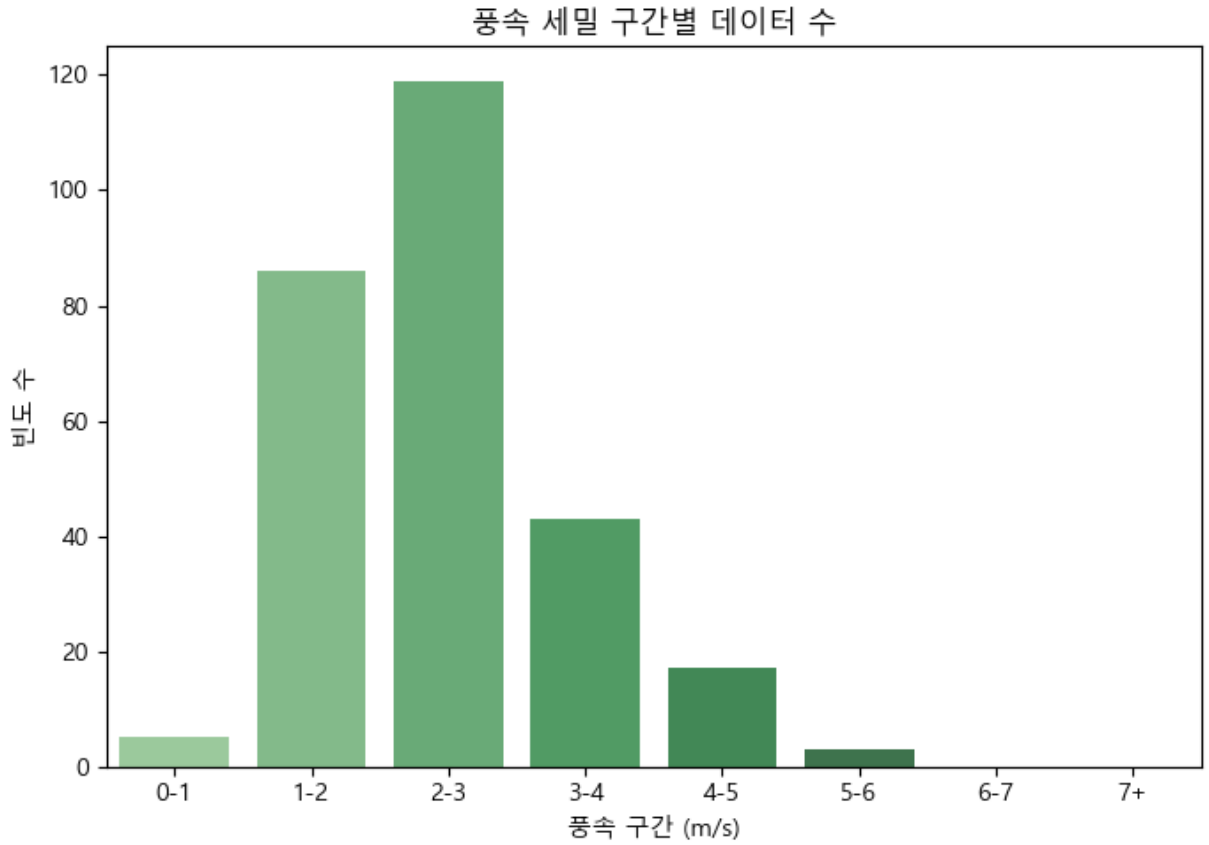
풍향 각도가 증가할수록 평균 대여량이 증가하는 추세를 보여

sin, cos 변환을 진행하여 조사

→ 각각 sin= −0.24, cos= −0.05로 유의미한 상관관계라고 보기 어려움

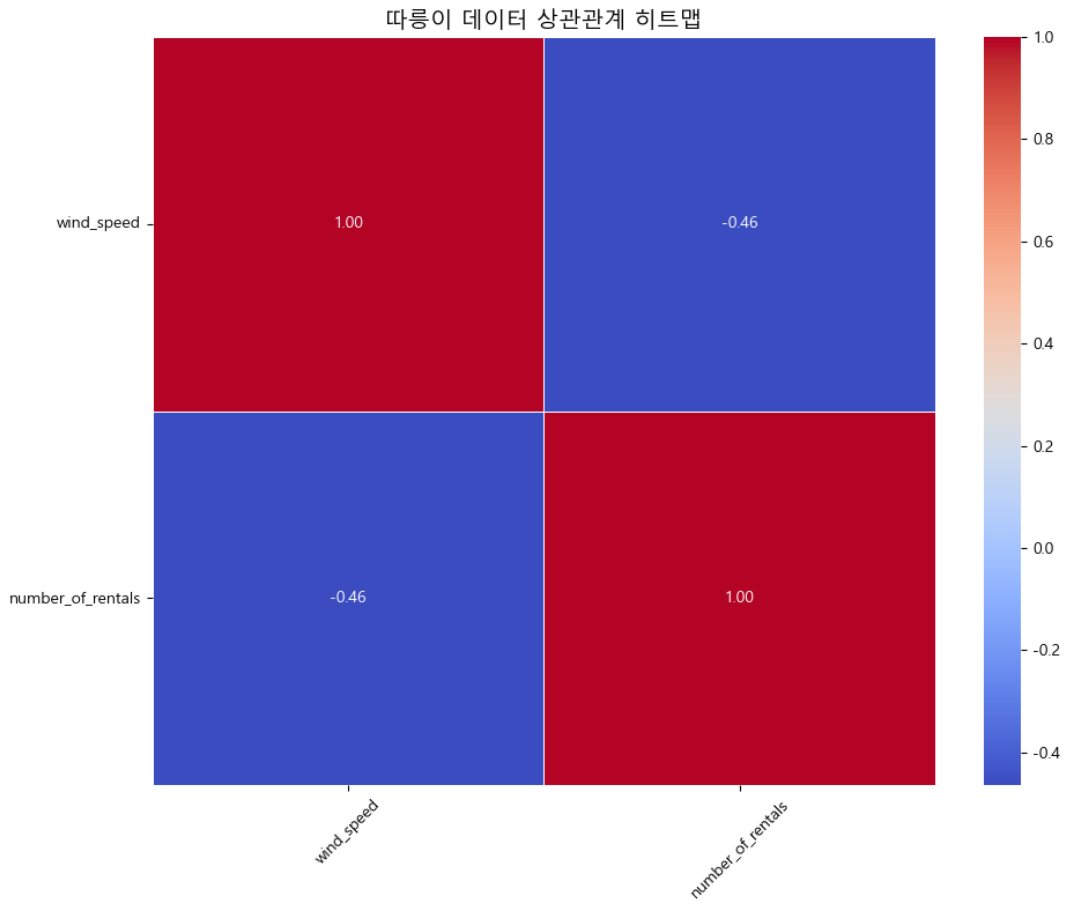
풍속 별 대여량과 데이터 빈도 수





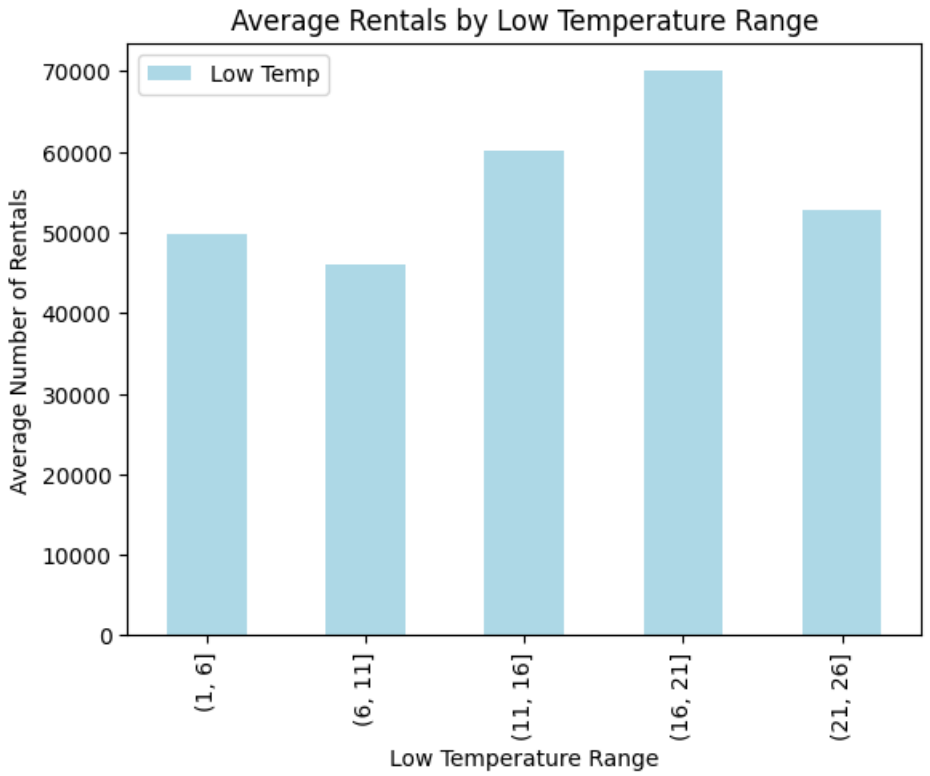
풍속은 1~3구간에서 가장 데이터 빈도가 많았고 풍속이 증가할수록 평균 대여량이 감소하는 경향을 보임.
또한 풍속과 대여량 간의 상관관계분석 결과 중간정도의 음의 관계(-0.46)를 가지고 있음
→ 풍속이 약할 때 (0~2m/s) 대여 수가 많음을 확인

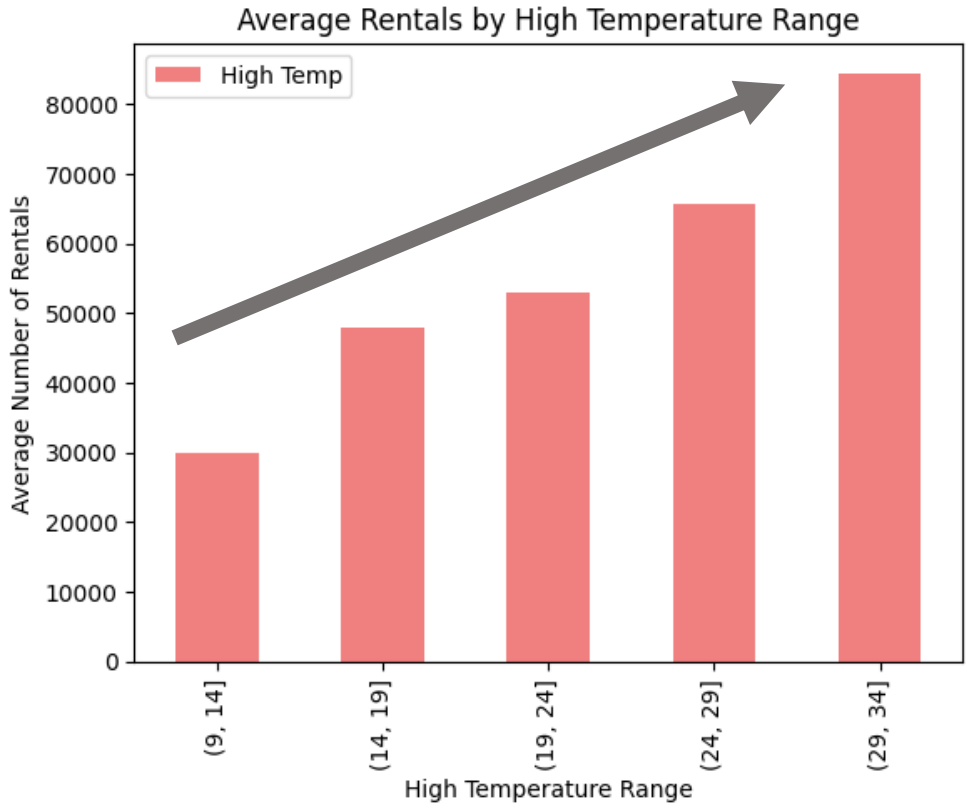
풍속과 대여량 간의 상관관계 분석 히트맵



Low temp & High temp 조사

방법: low temp, high temp 각각 최솟값부터 최댓값까지 구간을 5도씩 나눠서 시각화
목표: 온도가 가장 높은 구간에서 대여 수가 가장 많은지, 증가하는 추세를 띄는지 확인하기 위함





✓ 결과: low temp에서는 유의미한 상관관계를 보이지 않음
하지만 High temp에서 온도가 가장 높은 구간에서 대여 수가 가장 많음을 확인할 수 있었음
또한 최고기온이 증가할수록 대여 수도 많은 경향을 보임

일교차 조사

방법: high temp - low temp로 새로운 칼럼을 생성하고,
해당 칼럼의 최솟값부터 최댓값까지 1도 간격으로 구간을 나눠 시각화

✓결론

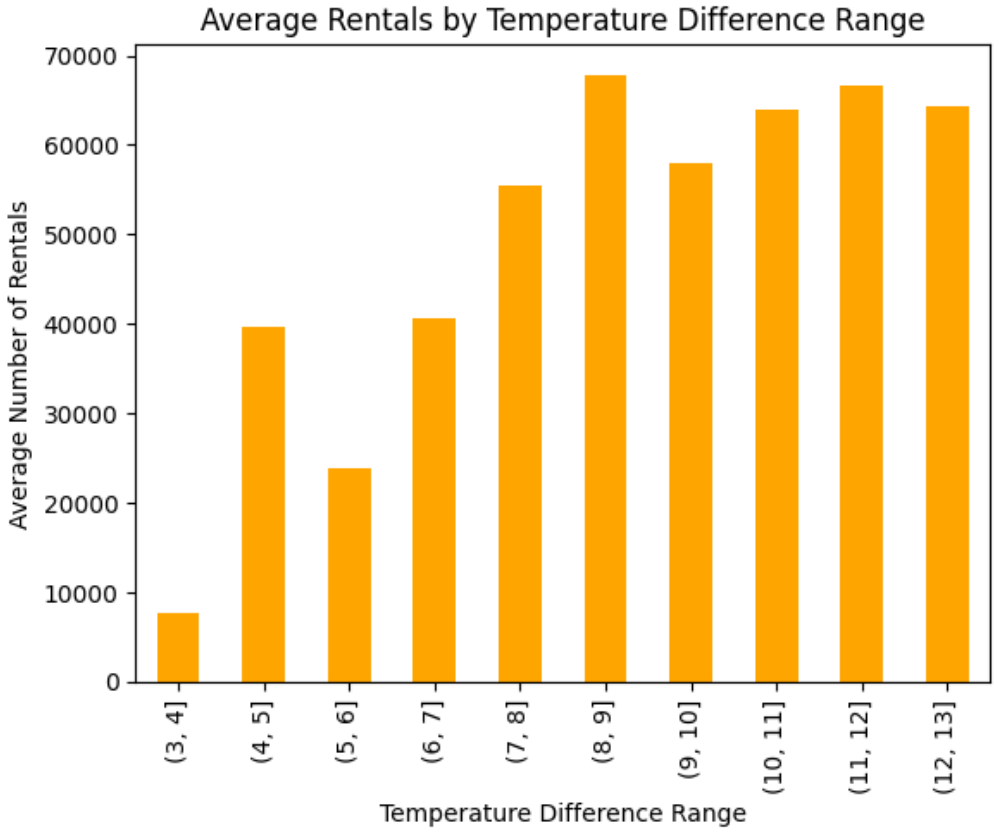
일교차와 대여 수사이의 유의미한 상관관계는 찾지 못하였으나,
일교차가 적을 때 대여 수가 확연히 적은 것을 확인

→ 일교차와 다른 변수들 간의 추가 조사 진행

: 일교차와 강수 확률이 뚜렷한 음의 상관관계를 나타냄 (-0.59)

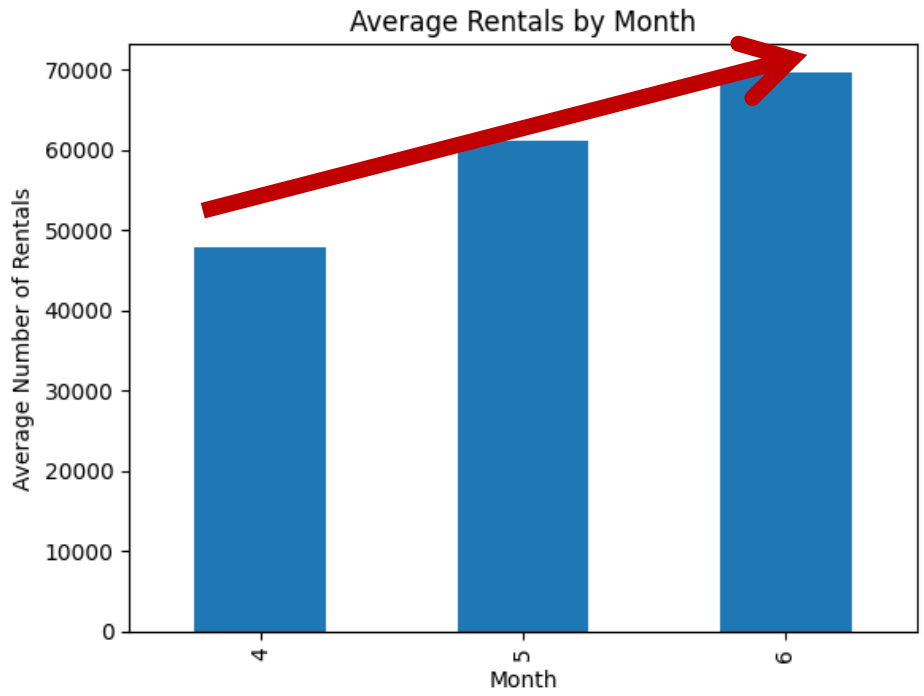
∴ 일교차가 작을 때는 강수 확률이 높은 날이었을 것으로 예상

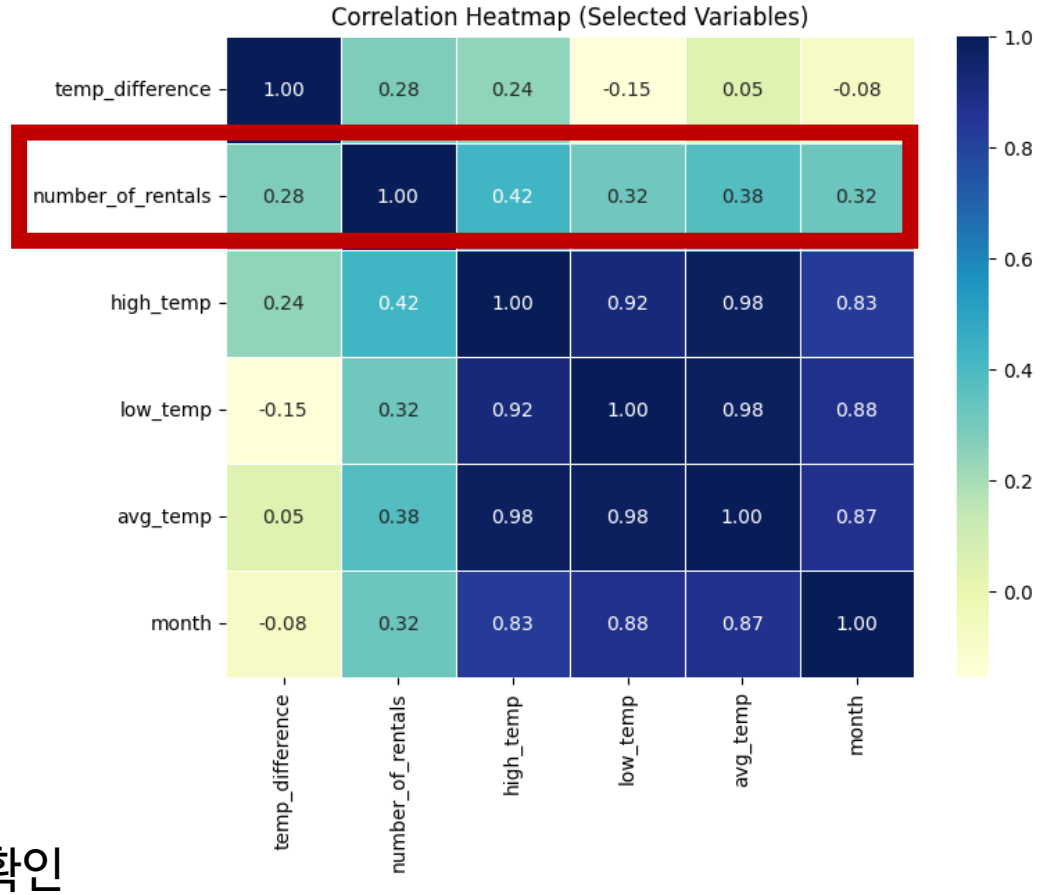
→ 강수확률과 대여 수에 유의미한 상관관계가 있음을 조사할 필요성이 있음.



월별 평균 대여량 조사

방법: 주어진 데이터를 월별로 그룹화하여 평균 대여 수를 시각화

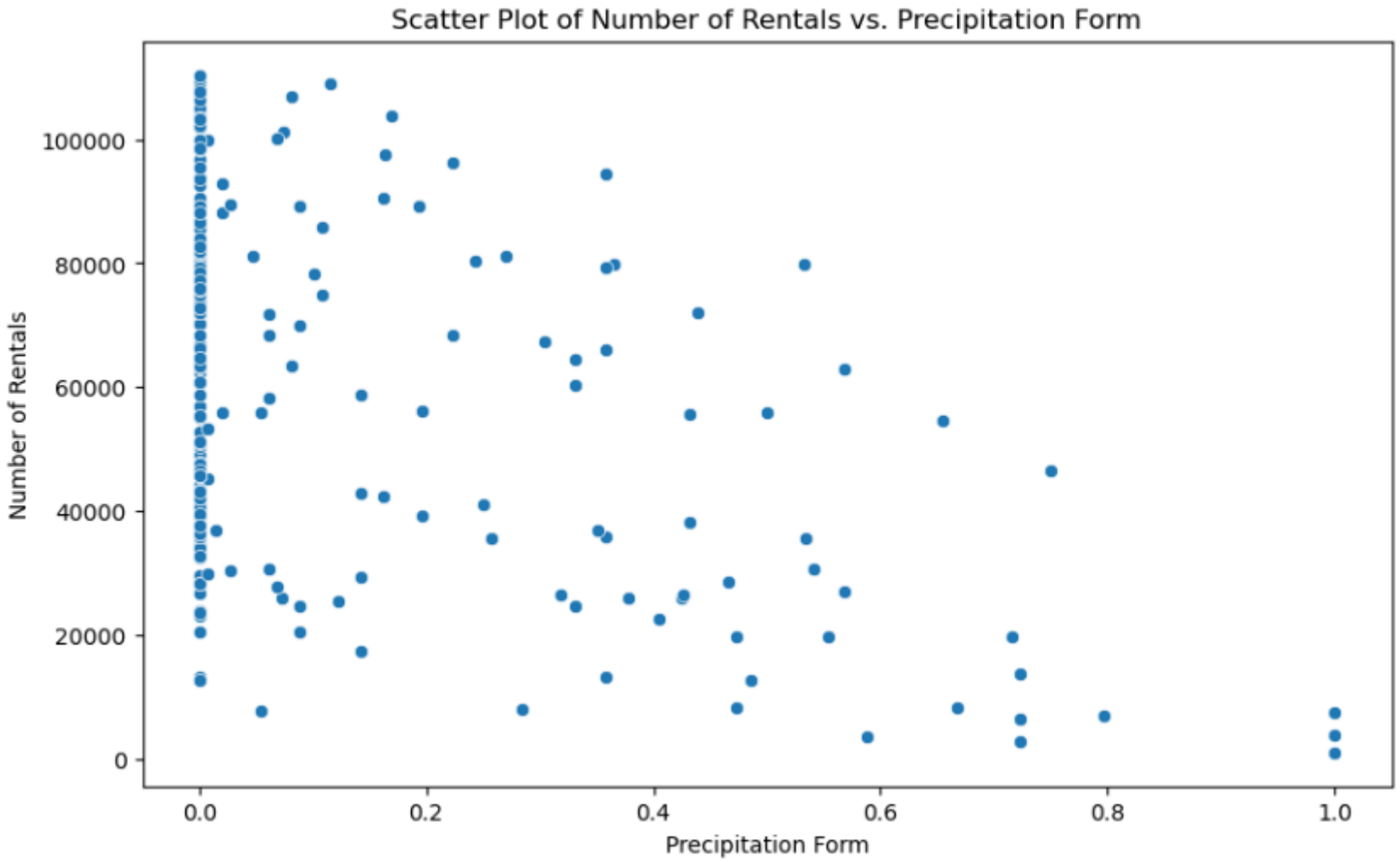




Heatmap에서 기온 관련 데이터와 강한 양적 상관관계를 나타냄을 확인

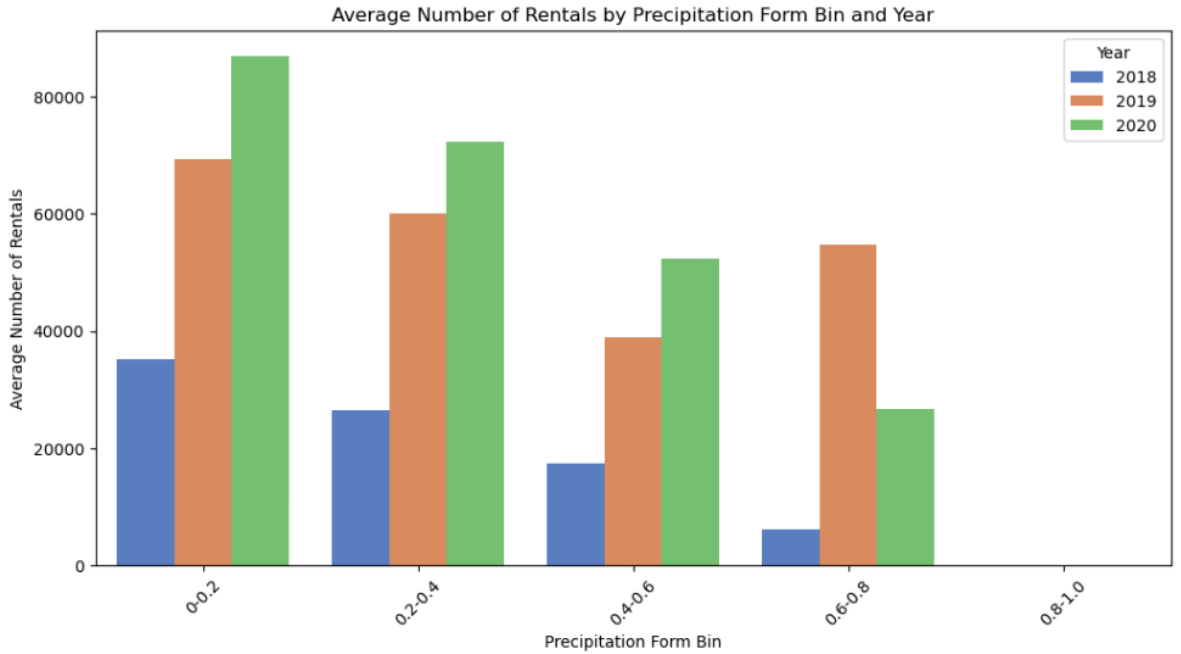
→ 6월로 갈수록 대여 수가 증가하는 이유는 기온이 상승하기 때문일 것으로 생각 가능

강수형태와 대여량 산점도



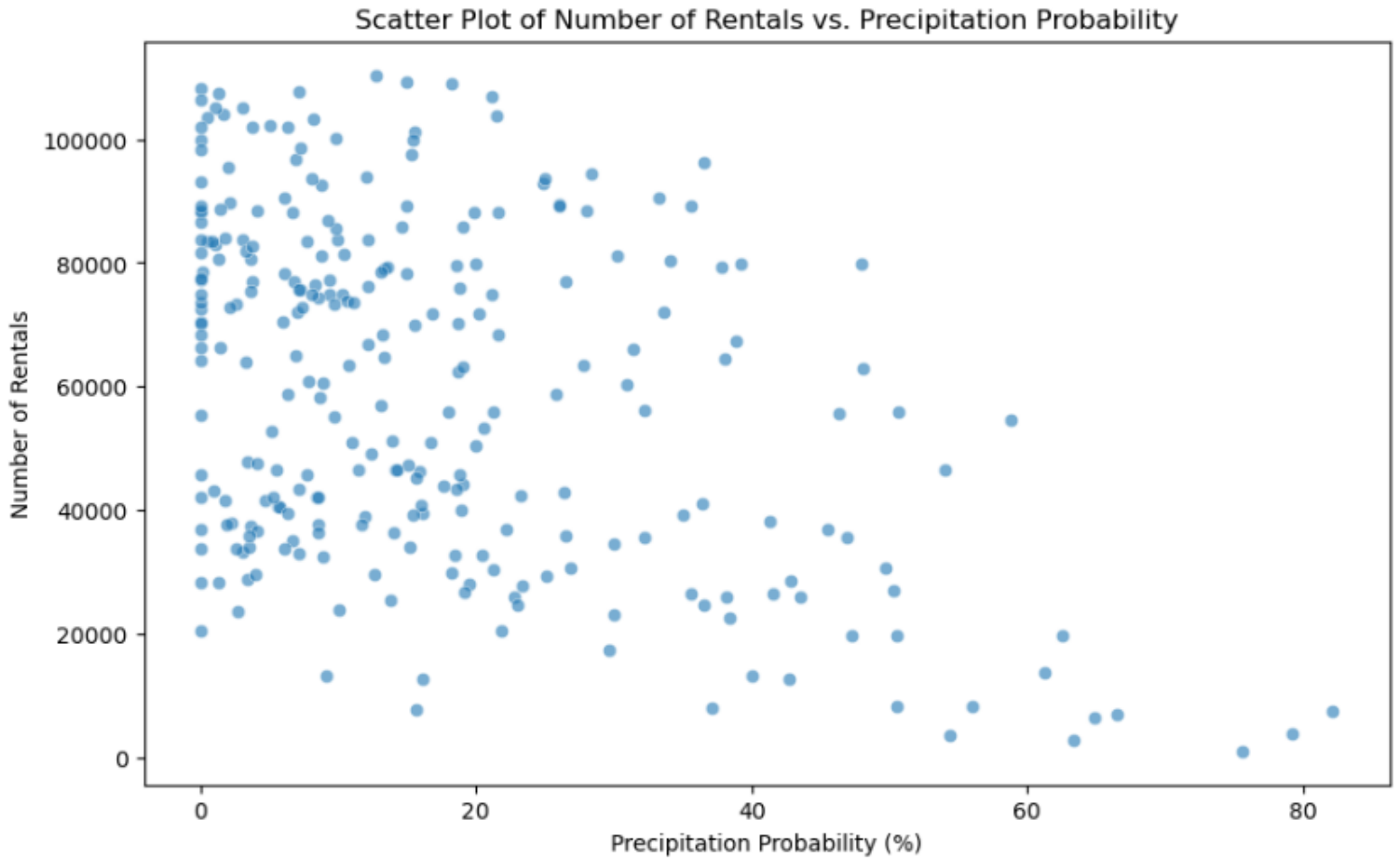
강수형태에 따른 대여량

0: 맑음 ~ 1: 비 =〉 0.2로 그룹화

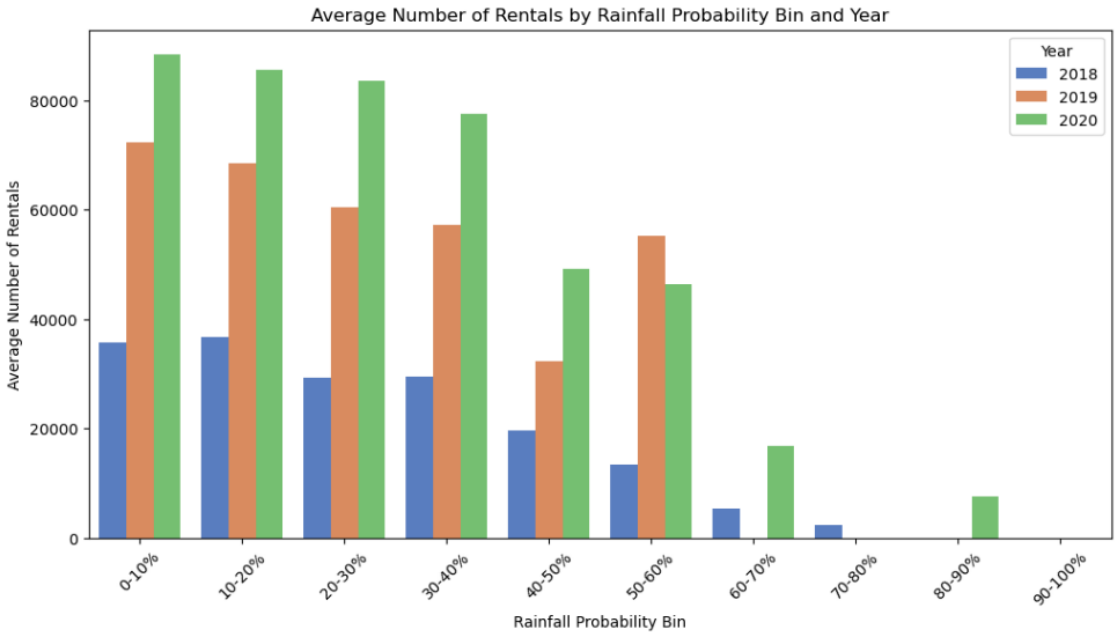


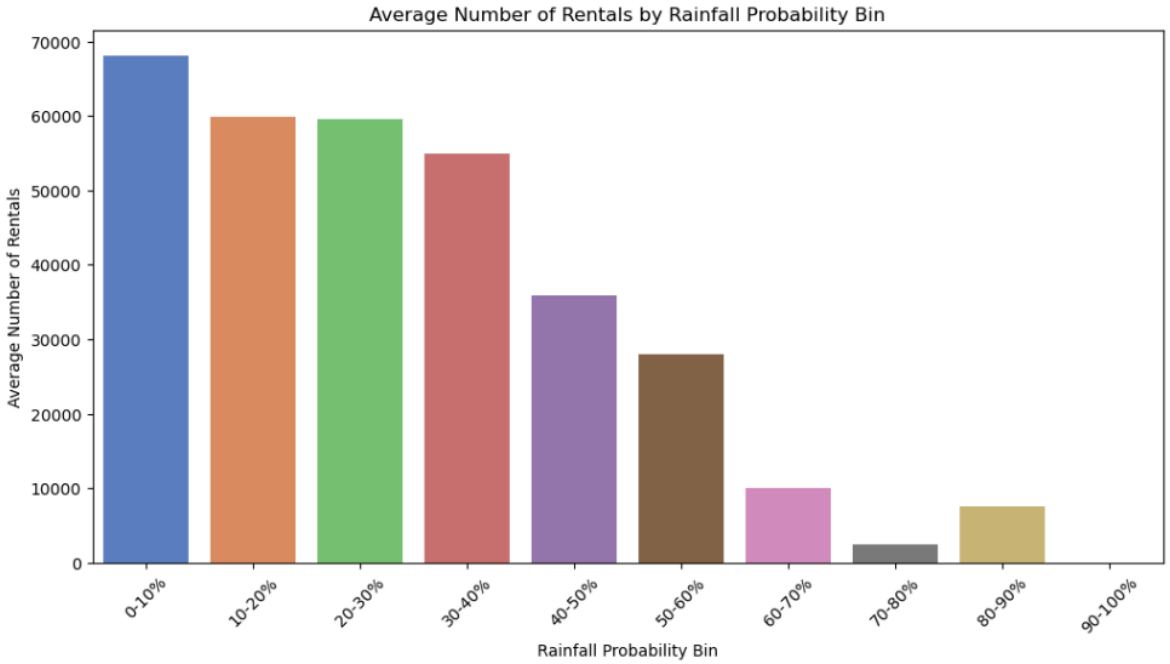
상관계수: -0.707 상관계수: -0.469 상관계수: -0.669

강수확률과 대여량 산점도



강수확률 구간별 대여량





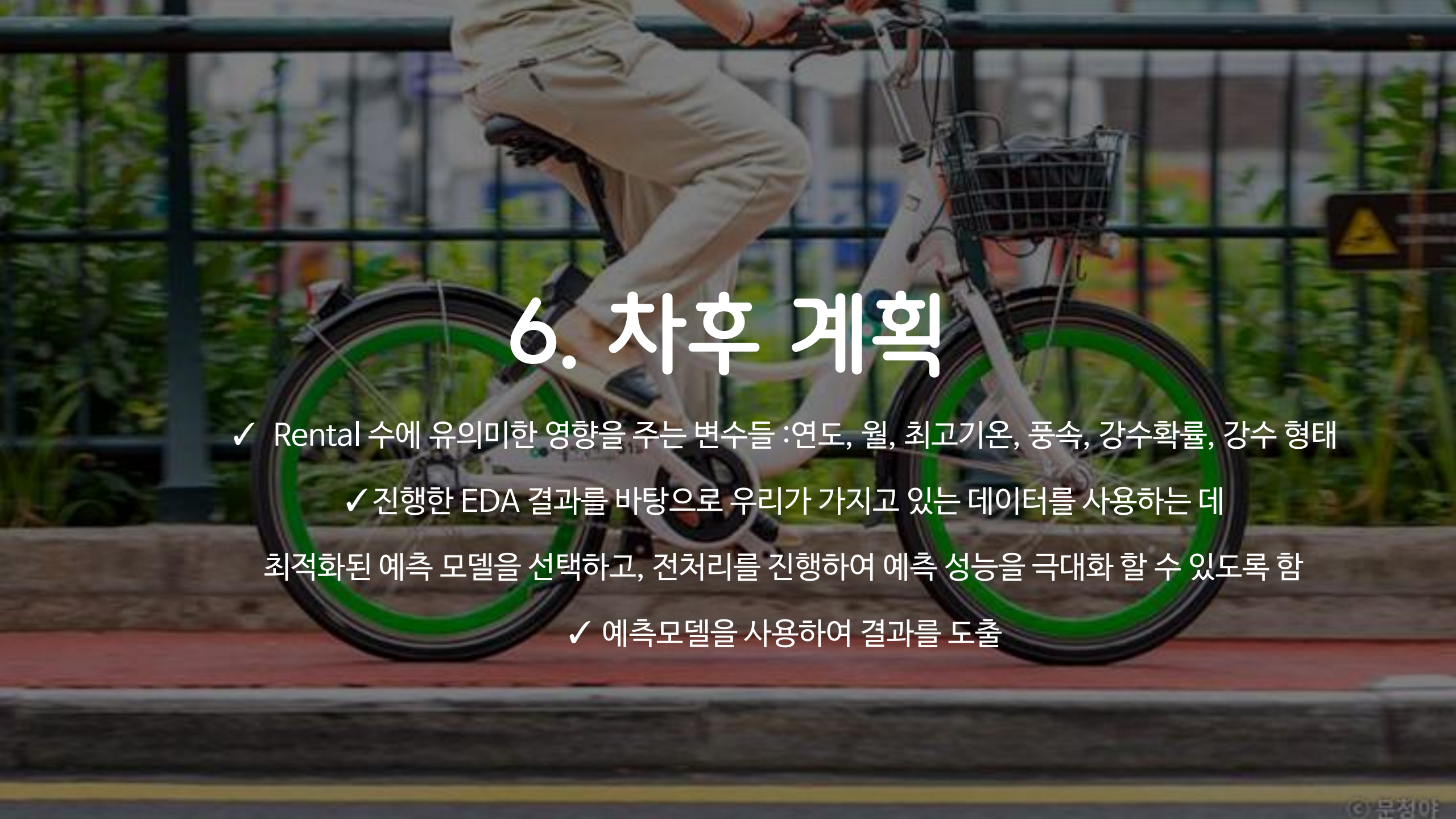
2018년
상관계수: -0.673

2019년
상관계수: -0.483

2020년
상관계수: -0.624

강수확률과 자전거 대여량 수 사이의 상관계수

: - 0.449



6. 차후 계획

✓ Rental 수에 유의미한 영향을 주는 변수들 :연도, 월, 최고기온, 풍속, 강수확률, 강수 형태

✓진행한 EDA 결과를 바탕으로 우리가 가지고 있는 데이터를 사용하는 데

최적화된 예측 모델을 선택하고, 전처리를 진행하여 예측 성능을 극대화 할 수 있도록 함

✓ 예측모델을 사용하여 결과를 도출

7. 질의응답