24.11.30(EDA&Preprocessing)

- 60191682 임준용

1. **호텔 유형별 예약 개수 & 예약 상태별 개수 계산**

[60]:
```python
# hotel 컬럼의 각 범주별 개수 계산
hotel_counts = df['hotel'].value_counts()

print("=== 호텔 유형별 예약 개수 ===")
for hotel_type, count in hotel_counts.items():
    print(f"{hotel_type}: {count}건")
```

```
=== 호텔 유형별 예약 개수 ===
City Hotel: 79330건
Resort Hotel: 40060건
```

[61]:
```python
# is_canceled(호텔 예약 취소 여부) 컬럼의 범주 개수 계산
is_canceled_counts = df['is_canceled'].value_counts()

print("=== 예약 상태별 개수 ===")
print(f"호텔 예약 유지: {is_canceled_counts.get(0, 0)}")
print(f"호텔 예약 취소: {is_canceled_counts.get(1, 0)}")
```

```
=== 예약 상태별 개수 ===
호텔 예약 유지: 75166
호텔 예약 취소: 44224
```

2. **데이터프레임 내 각 컬럼별 결측치 합계 및 비율 계산**

[62]:
```python
# 각 컬럼별 결측치 합계 계산
missing_values = df.isnull().sum()

print("=== 컬럼별 결측치 합계 ===")
for column, missing in missing_values.items():
    if missing > 0:
        print(f"{column}: {missing}건")
```

```
=== 컬럼별 결측치 합계 ===
children: 4건
country: 488건
agent: 16340건
company: 112593건
```

[63]:
```python
# 각 컬럼별 결측치 비율 계산
missing_ratios = df.isnull().mean() * 100

print("=== 컬럼별 결측치 비율 (%) ===")
for column, ratio in missing_ratios.items():
    if ratio > 0:
        print(f"{column}: {ratio:.4f}%")
```

```
=== 컬럼별 결측치 비율 (%) ===
children: 0.0034%
country: 0.4087%
agent: 13.6862%
company: 94.3069%
```

3. children 컬럼 결측치 처리 과정

[64]:
```python
# children 결측치 처리 - 1
df['children'].value_counts()
```

[64]:
```
children
0.0     110796
1.0       4861
2.0       3652
3.0         76
10.0         1
Name: count, dtype: int64
```

[65]:
```python
# children 결측치 처리 - 2
# children의 결측치의 합계가 4개이므로 크게 의미 없을 것이라고 판단, 해당 결측치를 결측치 값 중 압도적으로 많은 0으로 치환함
df['children'] = df['children'].fillna(0)
```

4. country 컬럼 결측치 처리 과정

[66]:
```
# country 결측치 처리 - 1
df['country'].value_counts()
```

[66]:
```
country
PRT    48590
GBR    12129
FRA    10415
ESP     8568
DEU     7287
       ...
DJI        1
BWA        1
HND        1
VGB        1
NAM        1
Name: count, Length: 177, dtype: int64
```

[67]:
```
# country 결측치 처리 - 2
# country는 결측치가 많지 않기 때문에 어떤 국가에도 명확히 속하지 않는 것으로 간주하고 "Unknown"로 치환
df['country'] = df['country'].fillna('Unknown')
```

5. agent 컬럼 결측치 처리 과정

[68]:
```
# agent 결측치 처리 - 1
df['agent'].value_counts()
```

[68]:
```
agent
9.0      31961
240.0    13922
1.0       7191
14.0      3640
7.0       3539
         ...
289.0        1
432.0        1
265.0        1
93.0         1
304.0        1
Name: count, Length: 333, dtype: int64
```

[69]:
```
# agent 결측치 처리 - 2
agent_9_count = df['agent'].value_counts().get(9.0, 0)
total_agent_count = df['agent'].notnull().sum()
percentage_9 = (agent_9_count / total_agent_count) * 100
print(f"agent에서 9.0이 차지하는 비율: {percentage_9:.2f}%")

agent_240_count = df['agent'].value_counts().get(240.0, 0)
total_agent_count = df['agent'].notnull().sum()
percentage_240 = (agent_240_count / total_agent_count) * 100
print(f"agent에서 240.0이 차지하는 비율: {percentage_240:.2f}%")
```

```
agent에서 9.0이 차지하는 비율: 31.02%
agent에서 240.0이 차지하는 비율: 13.51%
```

```
# agent 결측치 처리 - 3
# agent 컬럼의 범주를 이름 순으로 정렬
sorted_agent_counts = df['agent'].value_counts().sort_index()
print(sorted_agent_counts)
```

```
agent
1.0      7191
2.0       162
3.0      1336
4.0        47
5.0       330
          ...
510.0       2
526.0      10
527.0      35
531.0      68
535.0       3
Name: count, Length: 333, dtype: int64
```

```
# agent 결측치 처리 - 4
# agent의 결측값 비율 13.6%는 상당히 크며 최빈값 9.0이 agent 데이터 중 압도적으로 많은 값은 아니라고 판단,
# 결측값을 Unknown으로 치환
df['agent'] = df['agent'].fillna('Unknown')
```

6. company 컬럼 결측치 처리 과정

[72]:
```
# company 결측치 처리 - 1
df['company'].value_counts()
```

[72]:
```
company
40.0     927
223.0    784
67.0     267
45.0     250
153.0    215
         ...
104.0      1
531.0      1
160.0      1
413.0      1
386.0      1
Name: count, Length: 352, dtype: int64
```

[73]:
```
# company 결측치 처리 - 2
# 결측치의 비율이 너무 높아 company 컬럼 존재 자체가 크게 의미 없을 것이라 판단, 컬럼을 삭제함
df = df.drop('company', axis = 1)
```

7. 결측치 처리 후 각 컬럼별 결측치 존재 여부 확인

```python
# 결측치 처리 후 각 컬럼별 결측치 합계 계산
missing_values = df.isnull().sum()

print("=== 컬럼별 결측치 합계 ===")
any_missing = False

for column, missing in missing_values.items():
    if missing > 0:
        print(f"{column}: {missing}건")
        any_missing = True

if not any_missing:
    print("모든 컬럼에 결측치가 없습니다.")
```

```
=== 컬럼별 결측치 합계 ===
모든 컬럼에 결측치가 없습니다.
```