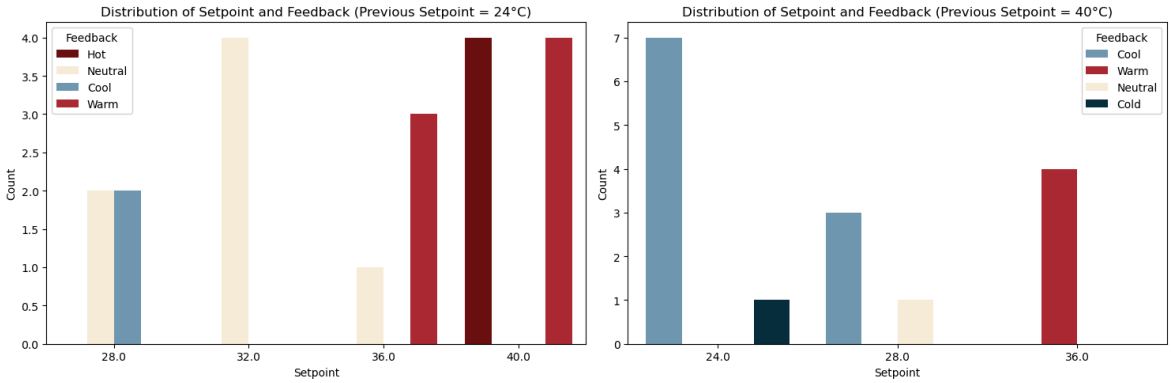
Out[ ]:

	Name	Setpoint	Achieved Temperature	Time Interval (s)	Time to Reach (s)	Feedback	Previous Setpoint
0	eunchae	40.0	40.08	19	19.23	Hot	32.0
1	eunchae	24.0	29.81	19	19.52	Cool	40.0
2	eunchae	40.0	36.17	19	19.08	Hot	24.0
3	eunchae	36.0	35.68	19	19.21	Warm	40.0
4	eunchae	40.0	39.59	19	19.42	Hot	36.0
5	eunchae	28.0	29.81	19	19.02	Cool	40.0
6	eunchae	24.0	NaN	19	19.29	Cold	28.0
7	eunchae	32.0	NaN	19	19.22	Neutral	24.0
8	eunchae	24.0	NaN	19	19.00	Cool	32.0
9	eunchae	40.0	37.63	19	19.25	Hot	24.0
10	eunchae	24.0	NaN	19	19.37	Cool	40.0
11	eunchae	28.0	26.39	19	19.17	Cool	24.0
12	eunchae	24.0	25.42	19	19.10	Cold	28.0
13	eunchae	36.0	NaN	19	19.30	Warm	24.0
14	jiyoung	40.0	NaN	19	19.23	Warm	32.0
15	jiyoung	24.0	NaN	19	19.27	Cool	40.0
16	jiyoung	40.0	NaN	19	19.02	Warm	24.0
17	jiyoung	36.0	NaN	19	19.48	Warm	40.0
18	jiyoung	40.0	40.57	19	19.19	Warm	36.0
19	jiyoung	28.0	29.81	19	19.28	Neutral	40.0
20	jiyoung	24.0	NaN	19	19.38	Cool	28.0
21	jiyoung	32.0	29.33	19	19.37	Neutral	24.0
22	jiyoung	24.0	NaN	19	19.21	Cool	32.0
23	jiyoung	40.0	NaN	19	19.32	Warm	24.0
24	jiyoung	24.0	26.39	19	19.40	Cool	40.0
25	jiyoung	28.0	NaN	19	19.41	Neutral	24.0
26	jiyoung	24.0	NaN	19	19.35	Cool	28.0
27	jiyoung	36.0	NaN	19	19.38	Neutral	24.0
28	minju	40.0	39.59	19	19.19	Hot	32.0
29	minju	24.0	28.35	19	19.36	Cool	40.0
30	minju	40.0	NaN	19	19.33	Warm	24.0
31	minju	36.0	35.68	19	19.19	Warm	40.0

	Name	Setpoint	Achieved Temperature	Time Interval (s)	Time to Reach (s)	Feedback	Previous Setpoint
32	minju	40.0	NaN	19	19.00	Hot	36.0
33	minju	28.0	30.30	19	19.12	Cool	40.0
34	minju	24.0	25.42	19	19.48	Cold	28.0
35	minju	32.0	28.84	19	19.03	Neutral	24.0
36	minju	24.0	NaN	19	19.23	Cold	32.0
37	minju	40.0	38.61	19	19.07	Hot	24.0
38	minju	24.0	NaN	19	19.25	Cool	40.0
39	minju	28.0	26.88	19	19.29	Neutral	24.0
40	minju	24.0	NaN	19	19.10	Cool	28.0
41	minju	36.0	33.24	19	19.17	Warm	24.0
42	taeyoon	40.0	NaN	19	19.42	Hot	32.0
43	taeyoon	24.0	29.33	19	19.33	Cold	40.0
44	taeyoon	40.0	35.68	19	19.51	Warm	24.0
45	taeyoon	36.0	35.68	19	19.47	Warm	40.0
46	taeyoon	40.0	NaN	19	19.42	Hot	36.0
47	taeyoon	28.0	NaN	19	19.33	Cool	40.0
48	taeyoon	24.0	26.39	19	19.22	Cold	28.0
49	taeyoon	32.0	28.35	19	19.01	Neutral	24.0
50	taeyoon	24.0	NaN	19	19.43	Cold	32.0
51	taeyoon	40.0	35.19	19	19.39	Hot	24.0
52	taeyoon	24.0	NaN	19	19.30	Cool	40.0
53	taeyoon	28.0	NaN	19	19.42	Cool	24.0
54	taeyoon	24.0	NaN	19	19.44	Cool	28.0
55	taeyoon	36.0	NaN	19	19.43	Warm	24.0

In [ ]:
```
# List of Previous Setpoint values to analyze
previous_setpoints = [24, 40]

# Set up the plotting area
plt.figure(figsize=(15, 5))

# Define a custom color palette for Feedback categories
custom_palette = {
    'Hot': '#780000',  # Red shade for Hot
    'Warm': '#c1121f',  # Lighter red (salmon) for Warm
    'Neutral': '#fdf0d5',
    'Cool': '#669bbc',  # Light blue for Cool
```

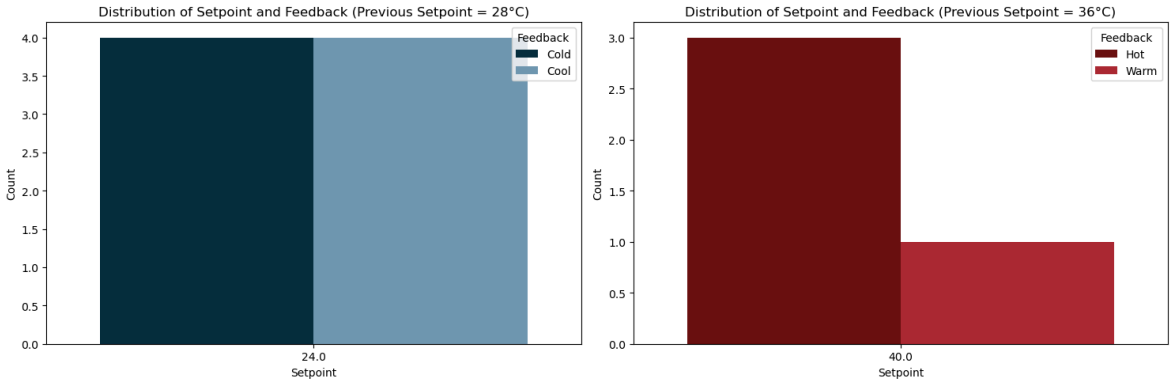
```
    'Cold': '#003049'   # Darker blue for Cold
}

# Iterate over each Previous Setpoint value
for i, setpoint in enumerate(previous_setpoints, 1):
    plt.subplot(1, len(previous_setpoints), i)

    # Filter the DataFrame for the current Previous Setpoint
    filtered_df = combined_df[combined_df['Previous Setpoint'] == setpoint]

    # Plot the relationship between Setpoint and Feedback
    sns.countplot(data=filtered_df, x='Setpoint', hue='Feedback', palette=custom

    plt.title(f'Distribution of Setpoint and Feedback (Previous Setpoint = {setp
    plt.xlabel('Setpoint')
    plt.ylabel('Count')
    plt.legend(title='Feedback')

plt.tight_layout()
plt.show()
```



In [ ]:

```
# List of Previous Setpoint values to analyze
previous_setpoints = [28, 36]

# Set up the plotting area
plt.figure(figsize=(15, 5))

# Define a custom color palette for Feedback categories
custom_palette = {
    'Hot': '#780000',  # Red shade for Hot
    'Warm': '#c1121f',  # Lighter red (salmon) for Warm
    'Neutral': '#fdf0d5',
    'Cool': '#669bbc',  # Light blue for Cool
    'Cold': '#003049'   # Darker blue for Cold
}

# Iterate over each Previous Setpoint value
for i, setpoint in enumerate(previous_setpoints, 1):
    plt.subplot(1, len(previous_setpoints), i)

    # Filter the DataFrame for the current Previous Setpoint
    filtered_df = combined_df[combined_df['Previous Setpoint'] == setpoint]

    # Plot the relationship between Setpoint and Feedback
    sns.countplot(data=filtered_df, x='Setpoint', hue='Feedback', palette=custom

    plt.title(f'Distribution of Setpoint and Feedback (Previous Setpoint = {setp
    plt.xlabel('Setpoint')
```

```
    plt.ylabel('Count')
    plt.legend(title='Feedback')

plt.tight_layout()
plt.show()
```



In [ ]:

In [ ]:
```
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(combined_df['Previous Setpoint'], combined_df['F

print(contingency_table)
print('\n\n')

# 카이제곱 검정 수행
chi2, p, dof, expected = chi2_contingency(contingency_table)

# 결과 출력
print("Chi-Square Statistic:", chi2)
print("p-value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies Table:")
print(expected)

# p-value를 기반으로 유의미한 차이 파악
alpha = 0.05  # 일반적으로 사용하는 유의 수준
if p < alpha:
    print("p-value가 유의 수준보다 작으므로, 'Previous Setpoint'와 'Feedback' 간
else:
    print("p-value가 유의 수준보다 크므로, 'Previous Setpoint'와 'Feedback' 간의
```

Feedback	Cold	Cool	Hot	Neutral	Warm
Previous Setpoint					
24.0	0	2	4	7	7
28.0	4	4	0	0	0
32.0	2	2	3	0	1
36.0	0	0	3	0	1
40.0	1	10	0	1	4

```
Chi-Square Statistic: 49.78886752136752
p-value: 2.4769447442344714e-05
Degrees of Freedom: 16
Expected Frequencies Table:
[[2.5        6.42857143 3.57142857 2.85714286 4.64285714]
 [1.         2.57142857 1.42857143 1.14285714 1.85714286]
 [1.         2.57142857 1.42857143 1.14285714 1.85714286]
 [0.5        1.28571429 0.71428571 0.57142857 0.92857143]
 [2.         5.14285714 2.85714286 2.28571429 3.71428571]]
```

p-value가 유의 수준보다 작으므로, 'Previous Setpoint'와 'Feedback' 간의 관계가 통계적으로 유의미합니다.

In [ ]: