

1 유형 문제집: 문제 + 정답코드 + 실행결과 + 최종질문

문제 1 (기본 그룹별 평균/결측치)

사용 데이터: type1_prob01.csv

문제

Industry 와 Age_Group 을 기준으로 Spend 의 평균을 구하시오.

Spend 의 결측치를 해당 그룹의 평균으로 채우시오.

결측치 처리 후 City 별 Spend 총합을 구하시오.

정답 코드

```
df['Spend'] = df['Spend'].fillna(df.groupby(['Industry',  
'Age_Group'])['Spend'].transform('mean'))  
df.groupby('City')['Spend'].sum()
```

실행 결과

City	
Busan	29497.228769
Incheon	30116.546395
Seoul	30386.285863

최종 질문

Spend 총합이 가장 높은 도시(City)의 이름은?

최종 코드

```
df.groupby('City')['Spend'].sum().idxmax()
```

정답

Seoul

문제 2 (기본 카테고리별 결측치 처리)

📁 사용 데이터: type1_prob02.csv

❓ 문제

Category 별 Revenue 평균을 구하시오.

해당 평균으로 결측치를 채우시오.

연도(Year)별 Revenue 평균을 출력하시오.

✓ 정답 코드

```
df['Revenue'] = df['Revenue'].fillna(df.groupby(['Category'])['Revenue'].transform('mean'))  
df.groupby('Year')['Revenue'].mean()
```

❖ 실행 결과

Year

2020 511.326495

2021 511.850439

2022 518.593494

🔍 최종 질문

Revenue 평균이 가장 높은 연도는?

💡 최종 코드

```
df.groupby('Year')['Revenue'].mean().idxmax()
```

⌚ 정답

2022

문제 3 (transform: 그룹 내 편차)

📁 사용 데이터: type1_prob03.csv

❓ 문제

Dept, Grade 별 평균 Salary 를 계산하고,

각 개인의 Salary 에서 해당 그룹의 평균을 빼서 'Diff' 컬럼을 만들고,

Diff 가 0 보다 큰 사람 수를 출력하시오.

✓ 정답 코드

```
df['Diff'] = df['Salary'] - df.groupby(['Dept', 'Grade'])['Salary'].transform('mean')  
df[df['Diff'] > 0].shape[0]
```

❖ 실행 결과

112

🔍 최종 질문

평균보다 급여가 높은 사람은 몇 명인가요?

💡 최종 코드

```
(df['Diff'] > 0).sum()
```

⌚ 정답

112



문제 4 (agg: 그룹별 평균/표준편차)

📁 사용 데이터: type1_prob04.csv

복수 카테고리의
평균과 표준편차를
구하는 문제입니다.

❓ 문제

Category, Month 별로 Sales 의 평균과 표준편차를 모두 구하시오.

평균은 'avg', 표준편자는 'std' 컬럼으로 나오게 하시오.

✓ 정답 코드

```
df.groupby(['Category', 'Month'])['Sales'].agg(avg='mean', std='std')
```



❖ 실행 결과

		avg	std	
Category	Month	Category	Month	
		Feb	962.447619	350.414411
		Jan	1013.907143	343.118075
	Mar	1076.921053	303.292945	
B	Feb	1061.363158	219.540460	
	Jan	975.709524	361.188675	
	Mar	1038.243478	279.565160	
C	Feb	950.541176	268.421907	

```
Jan 1065.906250 295.517937
Mar 1006.484211 310.972384
D Feb 1026.742105 288.375793
Jan 1159.033333 306.078196
Mar 1002.745000 370.213798
```

🔍 최종 질문

평균 매출이 가장 높은 카테고리/월 조합은?

💡 최종 코드

```
df.groupby(['Category', 'Month'])['Sales'].mean().idxmax()
```

☑ 정답

('D', 'Jan')

문제 5 (transform: 그룹 내 표준화)

📁 사용 데이터: type1_prob05.csv

❓ 문제

Product, Region 별로 Revenue 를 표준화하시오.

표준화된 값은 'Z_Score' 컬럼에 저장하시오.

✓ 정답 코드

```
df['Z_Score'] = (df['Revenue'] -
df.groupby(['Product','Region'])['Revenue'].transform('mean')) /
df.groupby(['Product','Region'])['Revenue'].transform('std')
```

❖ 실행 결과

```
Product Region Z_Score
0 P1 East 0.523695
1 P2 West -0.229028
2 P3 West -0.067149
3 P2 East 1.384337
4 P1 West -0.910405
```

🔍 최종 질문

Z_Score 의 절댓값이 가장 큰 데이터의 Product 와 Region 은?

💡 최종 코드

```
df.loc[df['Z_Score'].abs().idxmax(), ['Product', 'Region']]
```

⌚ 정답

```
{'Product': 'P2', 'Region': 'East'}
```

문제 6 (pivot_table 활용)

📁 사용 데이터: type1_prob06.csv

❓ 문제

pivot_table 을 사용하여 Month 와 Product 별 평균 Amount 를 구하시오.

✓ 정답 코드

```
pd.pivot_table(df, index='Month', columns='Product', values='Amount', aggfunc='mean')
```

❖ 실행 결과

Product	A	B	C
Month			
Feb	102.521622	98.306452	103.295833
Jan	97.821429	106.152174	95.100000
Mar	100.206667	102.277419	100.326829

🔍 최종 질문

Amount 평균이 가장 높은 월은?

💡 최종 코드

```
pivot_table.mean(axis=1).idxmax()
```

⌚ 정답

Feb

문제 7 (crosstab 활용)

📁 사용 데이터: type1_prob07.csv

? 문제

Age 와 Usage 별 고객 수를 crosstab 으로 구하시오.

비율을 포함한 테이블도 같이 출력하시오.

✓ 정답 코드

```
pd.crosstab(df['Age'], df['Usage'], margins=True)  
pd.crosstab(df['Age'], df['Usage'], normalize='index')
```

❖ 실행 결과

Usage High Low Medium All

Age

20 대 27 17 27 71

30 대 36 25 20 81

40 대 22 31 25 78

All 85 73 72 230

Usage High Low Medium

Age

20 대 0.380282 0.239437 0.380282

30 대 0.444444 0.308642 0.246914

40 대 0.282051 0.397436 0.320513

🔍 최종 질문

가장 높은 사용 비율을 보인 연령대(Age)는?

💡 최종 코드

```
pd.crosstab(df['Age'], df['Usage'], normalize='index').sum(axis=1).idxmax()
```

⌚ 정답

20 대

문제 8 (rolling 평균)

📁 사용 데이터: type1_prob08.csv

❓ 문제

Sales 컬럼에 대해 7 일 이동평균(rolling mean)을 구하시오.

결과는 'Rolling_Mean' 컬럼에 저장하시오.

✓ 정답 코드

```
df['Rolling_Mean'] = df['Sales'].rolling(window=7).mean()
```

❖ 실행 결과

	Date	Sales	Rolling_Mean
195	2023-07-15	577.1	450.614286
196	2023-07-16	423.4	468.928571
197	2023-07-17	341.7	461.042857
198	2023-07-18	600.4	475.257143
199	2023-07-19	496.1	480.242857

🔍 최종 질문

Rolling_Mean 0| 가장 높은 날짜는?

💡 최종 코드

```
df.loc[df['Rolling_Mean'].idxmax(), 'Date']
```

⌚ 정답

2023-07-19

문제 9 (diff 함수)

📁 사용 데이터: type1_prob09.csv

❓ 문제

Price 컬럼에 대해 전일 대비 차이(Diff)를 계산하시오.

결과는 'Diff' 컬럼에 저장하시오.

✓ 정답 코드

```
df['Diff'] = df['Price'].diff()
```

❖ 실행 결과

```
Date  Price  Diff
0 2023-03-01  971.8  NaN
1 2023-03-02 1366.1  394.3
2 2023-03-03  858.9 -507.2
3 2023-03-04 1092.1  233.2
4 2023-03-05 1110.3  18.2
```

🔍 최종 질문

가장 가격이 많이 하락한 날짜는?

💡 최종 코드

```
df.loc[df['Diff'].idxmin(), 'Date']
```

⌚ 정답

2023-03-03

문제 10 (pct_change 함수)

📁 사용 데이터: type1_prob10.csv

❓ 문제

Visitors 컬럼에 대해 전주 대비 변화율(pct_change)을 계산하시오.

결과는 'Change' 컬럼에 저장하시오.

✓ 정답 코드

```
df['Change'] = df['Visitors'].pct_change()
```

❖ 실행 결과

```
Week  Visitors  Change
0 2023-01-01  1065.7  NaN
1 2023-01-08  840.8 -0.211035
2 2023-01-15 1280.6  0.523073
3 2023-01-22  690.4 -0.460878
4 2023-01-29 1233.3  0.786356
```

🔍 최종 질문

가장 방문자 수 변화율이 큰 주는?

💡 최종 코드

```
df.loc[df['Change'].abs().idxmax(), 'Week']
```

⌚ 정답

2023-01-29