편상관 분석 (Partial Correlation)

- **두 변수 간의 순수한 상관 관계**를 파악하기 위해, 다른 제3의 변수(통제 변수)가 미치는 영향을 제거(통제)한 후 두 변수의 상관 관계를 측정하는 분석 방법

예를 들어, **"아이스크림 판매량"과 "익사 사고 발생 건수"**는 **강한 양의 상관 관계**를 보일 수 있습니다. 하지만 이는 두 변수 모두 **"기온"이라는 제3의 변수의 영향**을 받기 때문일 수 있습니다. **편상관 분석은 이 "기온"의 영향을 제거**하고 아이스크림 판매량과 익사 사고 발생 건수 간의 실제 관계를 분석할 수 있게 해줍니다.

적용 가능한 상황

- 여러 변수들이 서로 얽혀 있을 때, 특정 두 변수 사이의 직접적인 관계를 확인하고 싶을 때
- 회귀 분석에서 다중공선성이 의심될 때, 특정 변수들 간의 순수한 관계를 파악하여 변수 선택에 활용할 때
- 가짜 상관(spurious correlation) 관계를 식별하고 싶을 때
- **용도**: 다변량 데이터에서 특정 변수들의 영향을 배제하고 두 변수 간의 상관성을 보고자 할 때 사용
- **주의사항/가정**:
    - **선형성**: 변수들 간의 관계가 선형적이라고 가정합니다.
    - **정규성**: 데이터가 다변량 정규 분포를 따른다고 가정합니다.
    - 통제 변수를 무엇으로 설정하는지에 따라 결과가 크게 달라지므로, 이론적 배경이나 도메인 지식에 기반한 신중한 선택이 필요합니다.

1. Pingouin 사용 (권장)

- `pingouin.partial_corr()`
    - 데이터프레임, x, y, 그리고 통제 변수(covar)를 지정하여 쉽게 편상관계를 계산

```
import pandas as pd
import numpy as np
import pingouin as pg

# 예제 데이터 생성
# X: 운동 시간, Y: 스트레스 지수, Z: 업무 시간
# 업무 시간(Z)이 많을수록 운동 시간(X)이 줄고, 스트레스 지수(Y)는 높아지는 관계
np.random.seed(42)
n = 100
data = pd.DataFrame({
    'X (운동 시간)': np.random.normal(5, 1, n),
    'Y (스트레스)': np.random.normal(50, 5, n),
    'Z (업무 시간)': np.random.normal(8, 2, n)
})

# Z의 영향을 X와 Y에 추가
data['X (운동 시간)'] = data['X (운동 시간)'] - 0.5 * data['Z (업무 시간)']
```

```
data['Y (스트레스)'] = data['Y (스트레스)'] + 2 * data['Z (업무 시간)']

# 단순 상관 분석 수행
print("--- 단순 상관 분석 (Z 통제 전) ---")
print(data.corr(method='pearson'))
'''
--- 단순 상관 분석 (Z 통제 전) ---
          X (운동 시간)  Y (스트레스)  Z (업무 시간)
X (운동 시간)   1.000000 -0.539691  -0.714711
Y (스트레스)   -0.539691  1.000000   0.657874
Z (업무 시간)  -0.714711  0.657874   1.000000
'''

# 편상관 분석 수행 (Z의 영향을 통제)
print("\n--- 편상관 분석 (Z 통제 후) ---")
# x: 'X (운동 시간)', y: 'Y (스트레스)', covar: 'Z (업무 시간)'
partial_corr_result = pg.partial_corr(data=data, x='X (운동 시간)', y='Y (스트레
스)', covar='Z (업무 시간)', method='pearson')
print(partial_corr_result)
'''
--- 편상관 분석 (Z 통제 후) ---
           n         r          CI95%        r2    adj_r2     p-val    BF10
power
pearson  100 -0.131943  [-0.32, 0.07]  0.017409 -0.002851  0.190679  0.291
0.259133
'''
```

2. Statsmodels를 이용한 직접 계산

편상관은 두 변수를 각각 통제 변수에 대해 선형 회귀를 수행한 후, 그 잔차(residual)들 간의 상관 관계를 구하는 것과 동일합니다.

```
import pandas as pd
import numpy as np
import statsmodels.api as sm

# 예제 데이터 (위와 동일)
np.random.seed(42)
n = 100
data = pd.DataFrame({
    'X': np.random.normal(5, 1, n),
    'Y': np.random.normal(50, 5, n),
    'Z': np.random.normal(8, 2, n)
})
data['X'] = data['X'] - 0.5 * data['Z']
data['Y'] = data['Y'] + 2 * data['Z']

# 1. X를 Z로 회귀분석 후 잔차 구하기
X_model = sm.OLS(data['X'], sm.add_constant(data['Z'])).fit()
X_residuals = X_model.resid
```

```
# 2. Y를 Z로 회귀분석 후 잔차 구하기
Y_model = sm.OLS(data['Y'], sm.add_constant(data['Z'])).fit()
Y_residuals = Y_model.resid

# 3. 두 잔차 간의 상관관계 계산
partial_corr_manual = np.corrcoef(X_residuals, Y_residuals)[0, 1]

print(f"--- Statsmodels를 이용한 수동 계산 ---")
print(f"X와 Y의 편상관 계수 (Z 통제): {partial_corr_manual:.4f}") # -0.1319
```

결과 해석 방법

- `pingouin` 결과 테이블:
  - **n**: 관측치 수
  - **r**: 편상관 계수. 해석은 일반 상관 계수와 동일합니다.
  - **CI95%**: 상관 계수의 95% 신뢰 구간
  - **p-val**: 편상관 계수가 통계적으로 유의미한지에 대한 p-value. `p < 0.05` 이면 통계적으로 유의미하다고 봅니다.
- **결과 비교**:
  - 예제 코드에서, 단순 상관 분석 결과 X와 Y는 강한 음의 상관(-0.7 부근)을 보입니다. 이는 업무 시간(Z)이 많을수록 운동 시간(X)은 줄고 스트레스(Y)는 높아지는 공통된 영향 때문입니다.
  - 하지만 업무 시간(Z)의 영향을 통제한 편상관 분석 결과, X와 Y의 상관 계수는 0에 가까워지며 통계적 유의성도 사라집니다(p-value 증가). 이는 운동 시간과 스트레스 지수 간의 직접적인 관련성은 거의 없음을 시사합니다.

장단점 및 대안

장점

- 변수들 간의 복잡한 관계 속에서 특정 두 변수 간의 순수한 관계를 규명할 수 있습니다.
- 다중공선성 문제의 원인을 파악하거나, 회귀 모델의 변수 선택에 대한 통찰력을 제공합니다.
- 가짜 상관 관계를 식별하여 잘못된 결론을 내리는 것을 방지할 수 있습니다.

단점

- 어떤 변수를 통제 변수로 선택할지에 대한 이론적 근거가 중요하며, 자의적으로 선택할 경우 잘못된 결과를 얻을 수 있습니다.
- 변수 간의 관계가 비선형일 경우, 편상관 분석만으로는 관계를 제대로 파악하기 어렵습니다.
- 통제 변수가 너무 많아지면 결과의 해석이 복잡해지고 불안정해질 수 있습니다.

대안

- **다중 회귀 분석 (Multiple Regression)**: 여러 독립 변수들이 종속 변수에 미치는 영향을 동시에 고려하므로, 각 변수의 계수는 다른 변수들이 통제된 상태에서의 효과를 나타냅니다. 편상관 분석과 유사한 해석을 제공하면서도 예측 모델링까지 가능합니다.
- **구조 방정식 모델링 (SEM, Structural Equation Modeling)**: 변수들 간의 복잡한 인과 관계 및 경로를 설정하고 검증할 수 있는 고급 통계 모델입니다. 편상관보다 더 복잡한 변수 관계 구조를 분석할 수 있습니다.