상관 분석 (Correlation Analysis)

- 두 변수 간의 선형적인 관계의 강도와 방향을 측정하는 통계적 방법
- 상관 계수(Correlation Coefficient)를 사용하여 관계를 수치화
- **값의 범위**: -1 ~ 1
  - **1**: 완벽한 양의 선형 관계 (한 변수가 증가하면 다른 변수도 증가)
  - **-1**: 완벽한 음의 선형 관계 (한 변수가 증가하면 다른 변수는 감소)
  - **0**: 선형 관계 없음
- **주의**: 상관 관계 ≠ 인과 관계
  - 상관 관계와 인과 관계는 서로 다르다.

적용 가능한 상황

- 두 연속형 변수 간의 관계를 파악하고 싶을 때 (예: 키와 몸무게의 관계)
- 다중공선성 등 회귀 분석의 가정 충족 여부를 확인할 때
- 변수 선택 과정에서 종속 변수와 관련성이 높은 독립 변수를 찾을 때

1. Pearson 상관 분석

- **사용 상황**
  - 두 변수가 모두 연속형(수치형)
  - 정규 분포를 따름
  - 선형적인 관계를 가짐
  - 가장 널리 사용되는 상관 계수
- **주의사항/가정**:
  - **선형성**: 두 변수 간의 관계가 선형적이어야 합니다. 산점도를 통해 확인할 수 있습니다.
  - **정규성**: 각 변수가 정규 분포를 따라야 합니다. (데이터가 많을 경우 중심극한정리에 의해 완화될 수 있습니다.)
  - **등분산성**: 오차의 분산이 일정해야 합니다.
  - **이상치 민감성**: 이상치에 민감하므로, 분석 전 이상치 처리가 필요할 수 있습니다.

```
import pandas as pd
import numpy as np
from scipy.stats import pearsonr
import pingouin as pg

# 예제 데이터 생성
np.random.seed(42)
data = pd.DataFrame({
    'X': np.random.normal(0, 1, 100),
    'Y_linear': 2 * np.random.normal(0, 1, 100) + 3,
    'Y_nonlinear': np.random.normal(0, 1, 100)**2,
    'Z': np.random.normal(0, 1, 100)
})
data['X_Y_linear_corr'] = data['X'] + data['Y_linear']

# 1. Pandas 사용
```

```
print("--- Pandas ---")
# 데이터프레임 전체 상관 행렬
corr_matrix = data.corr(method='pearson')
print(corr_matrix)
'''
--- Pandas ---
                        X  Y_linear  Y_nonlinear         Z  X_Y_linear_corr
X                1.000000 -0.136422     0.074647 -0.170227         0.324385
Y_linear        -0.136422  1.000000     0.016339 -0.017613         0.892828
Y_nonlinear      0.074647  0.016339     1.000000 -0.038892         0.049540
Z               -0.170227 -0.017613    -0.038892  1.000000        -0.094211
X_Y_linear_corr  0.324385  0.892828     0.049540 -0.094211         1.000000
'''
# 두 변수 간의 상관 계수
corr_xy = data['X'].corr(data['Y_linear'], method='pearson')
print(f"\nX와 Y_linear 간의 Pearson 상관 계수: {corr_xy:.4f}") # -0.1364

# 2. Scipy 사용
print("\n--- Scipy ---")
# pearsonr(x, y) -> (상관계수, p-value) 반환
corr, p_value = pearsonr(data['X'], data['Y_linear'])
print(f"상관 계수: {corr:.4f}") # -0.1364
print(f"P-value: {p_value:.4f}") # 0.1759

# 3. Pingouin 사용
print("\n--- Pingouin ---")
# pg.corr(x, y) -> 상세한 통계 결과 반환
corr_result = pg.corr(data['X'], data['Y_linear'], method='pearson')
print(corr_result)
'''
--- Pingouin ---
           n         r          CI95%        r2    adj_r2     p-val   BF10
power
pearson  100 -0.136422  [-0.32, 0.06]  0.018611 -0.001624  0.175926  0.308
0.273868
'''
```

결과 해석 방법

- **상관 계수 (r)**:
  - 0.7 ~ 1.0: 매우 강한 양의 상관
  - 0.4 ~ 0.7: 강한 양의 상관
  - 0.2 ~ 0.4: 뚜렷한 양의 상관
  - 0.0 ~ 0.2: 거의 없는 상관
  - (음수 값은 반대 방향의 관계를 의미)
- **p-value**:
  - 상관 계수가 통계적으로 유의미한지를 나타냅니다.
  - 일반적으로 `p-value < 0.05` 이면, "두 변수 간의 상관 관계가 없다"는 귀무가설을 기각하고, 상관 관계가 통계적으로 유의미하다고 해석합니다.

2. Spearman 상관 분석

- **사용 상황**:
  - 두 변수 중 하나라도 서열 척도(순위형 데이터)일 때
  - 데이터가 정규 분포를 따르지 않거나, 선형 관계가 아닐 때
    - 단, 단조 관계는 있어야 함 (동시에 증가/감소 하지만 일정 비율은 아닌 관계)
  - 이상치에 덜 민감하여 robust한 분석이 필요할 때 유용
- **주의사항/가정**:
  - 변수의 실제 값 대신 순위를 매겨 상관 관계를 계산합니다.
  - **단조성(Monotonicity)**: 한 변수가 증가할 때 다른 변수도 계속 증가하거나 계속 감소하는 관계를 가정합니다. (반드시 선형일 필요는 없음)

```
import pandas as pd
import numpy as np
from scipy.stats import spearmanr
import pingouin as pg

# 예제 데이터 생성 (비선형 단조 관계)
np.random.seed(42)
x = np.arange(1, 101)
y = np.log(x) + np.random.normal(0, 0.2, 100)
data = pd.DataFrame({'X': x, 'Y': y})

# 1. Pandas 사용
print("--- Pandas ---")
corr_spearman = data.corr(method='spearman')
print(corr_spearman)
'''
--- Pandas ---
        X       Y
X  1.0000  0.9552
Y  0.9552  1.0000
'''

# 2. Scipy 사용
print("\n--- Scipy ---")
corr, p_value = spearmanr(data['X'], data['Y'])
print(f"상관 계수: {corr:.4f}")  # 0.9552
print(f"P-value: {p_value:.4f}") # 0.0000

# 3. Pingouin 사용
print("\n--- Pingouin ---")
corr_result = pg.corr(data['X'], data['Y'], method='spearman')
print(corr_result)
'''
--- Pingouin ---
             n       r         CI95%        r2  adj_r2         p-val  power
spearman   100  0.9552  [0.93, 0.97]  0.912406  0.9106  1.276212e-53    1.0
'''
```

결과 해석 방법

- Pearson과 동일하게 상관 계수와 p-value를 해석합니다.
- Spearman 상관 계수는 변수들의 순위 사이의 선형 관계를 나타냅니다.

3. Kendall's Tau (켄달 토)

- **사용 상황**:
  - Spearman과 유사한 상황에 사용
    - 서열 척도나 비정규 분포 데이터
  - 표본 크기가 작거나, 데이터 내에 동일한 순위(tie)가 많을 때 Spearman보다 더 정확한 경향이 존재
- **주의사항/가정**:
  - 모든 데이터 쌍(pair)에 대해 일치하는지(concordant) 불일치하는지(discordant)를 계산하여 상관관계를 측정합니다.
  - 계산 과정이 복잡하여 데이터가 많을 경우 계산 속도가 느릴 수 있습니다.

```
import pandas as pd
import numpy as np
from scipy.stats import kendalltau
import pingouin as pg

# 예제 데이터 생성
np.random.seed(42)
data = pd.DataFrame({
    'X': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Y': [2, 1, 4, 3, 6, 5, 8, 7, 10, 9] # 순위가 뒤바뀐 경우
})

# 1. Pandas 사용
print("--- Pandas ---")
corr_kendall = data.corr(method='kendall')
print(corr_kendall)
'''
--- Pandas ---
          X         Y
X  1.000000  0.777778
Y  0.777778  1.000000
'''

# 2. Scipy 사용
print("\n--- Scipy ---")
corr, p_value = kendalltau(data['X'], data['Y'])
print(f"상관 계수: {corr:.4f}")  # 0.7778
print(f"P-value: {p_value:.4f}") # 0.0009

# 3. Pingouin 사용
print("\n--- Pingouin ---")
corr_result = pg.corr(data['X'], data['Y'], method='kendall')
print(corr_result)
'''
--- Pingouin ---
          n         r         CI95%        r2    adj_r2     p-val     power
```

```
kendall  10  0.777778  [0.29, 0.94]  0.604938  0.492063  0.000946  0.814667
'''
```

결과 해석 방법

- Pearson, Spearman과 동일하게 상관 계수와 p-value를 해석합니다.
- Kendall's Tau는 두 변수의 순위 관계가 얼마나 일치하는지를 나타냅니다.

장단점 및 대안

분석 방법	장점	단점	대안
Pearson	- 가장 널리 쓰이고 해석이 직관적임 - 통계적 검정(p-value)이 명확함	- 선형 관계만 측정 가능 - 정규성, 등분산성 가정이 필요 - 이상치에 민감함	- 비선형 관계나 이상치가 의심될 때 **Spearman** 또는 **Kendall** 사용
Spearman	- 비선형(단조) 관계 측정 가능 - 서열 척도에 적용 가능 - 이상치에 덜 민감함 (로버스트함)	- Pearson보다 검정력이 낮을 수 있음 - 연속형 데이터의 정보를 일부 손실함	- 데이터가 정규성을 만족하고 선형 관계이면 **Pearson**이 더 효율적임 - 표본이 작거나 동일 순위가 많으면 **Kendall** 고려
Kendall	- Spearman과 장점 공유 - 작은 표본, 동일 순위가 많은 데이터에 더 적합	- 계산 비용이 높아 대용량 데이터에 느림 - 일반적으로 Spearman보다 값이 작게 나오는 경향이 있음	- 일반적인 상황에서는 계산이 빠른 **Spearman**을 더 선호함

대안적 접근:

- **편상관 분석 (Partial Correlation)**: 두 변수 간의 순수한 상관 관계를 알고 싶을 때, 제3의 변수의 영향을 통제한 후 상관 관계를 분석합니다.
- **거리 상관 (Distance Correlation)**: 선형, 비선형 관계를 모두 잡아낼 수 있으며, 두 변수가 독립일 경우 상관 계수가 0이 나오는 특징이 있습니다. (계산이 복잡함)