강원지역혁신플랫폼

기계학습

Machine Learning

K-평균 군집 실습(2)

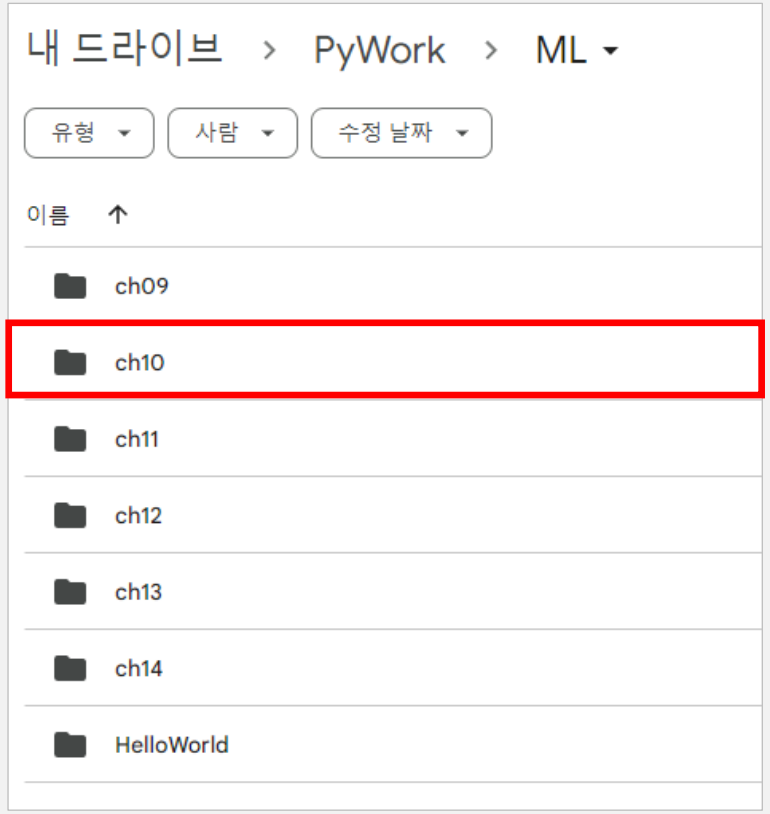
학습목표

Mall Customers Clustering Analysis
데이터 집합으로 군집 분석을 구현할 수 있습니다.

01 | 실습

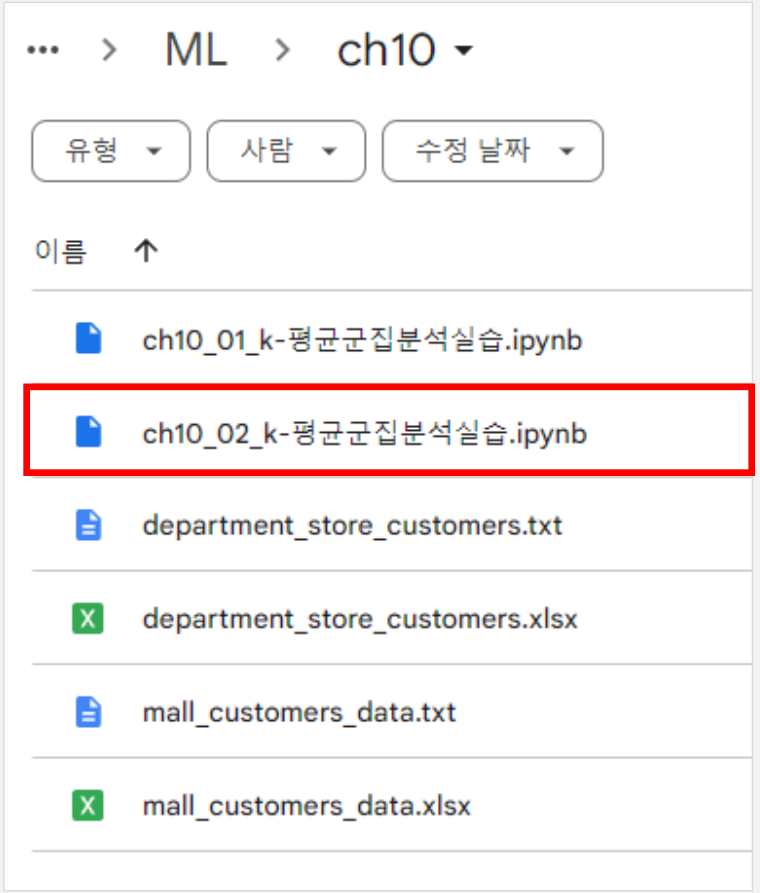
(권장) 아래와 같은 경로에 실행 소스가 존재하면 환경 구축 완료

- 구글 드라이브 "PyWork > ML" 폴더로 이동함
  - 아래의 [ch10] 폴더를 클릭하면 됨



01 | 실습

- “ML > ch10 >” 폴더를 클릭함
  - 아래의 [ch10_02_k-평균군집분석실습.ipynb] 스크립트를 클릭함



02 | 비지도 학습: K-평균 군집 분석

Kaggle의 Mall Customers Clustering Analysis 데이터 집합으로 고객 분류하기

- 다음은 Mall Customers Clustering Analysis 데이터 집합 200개로 마케팅을 위한 타겟 고객 분류를 수행해 보자.
  - Mall Customers Clustering Analysis 데이터 집합은 다음과 같음
    - Mall Customers Clustering Analysis 데이터 집합은 200개의 관측치와 5개의 속성으로 구성됨

No	속성	속성 설명
1	고객ID	고객ID
2	성별	성별 (여성, 남성)
3	나이	나이
4	연간소득	연간소득 (달러를 원으로 환산한 금액)
5	지출 점수 (1-100)	지출 점수 (1-100)

02 | 비지도 학습: K-평균 군집 분석

- 다음은 Mall Customers Clustering Analysis 데이터 집합을 읽어오는 코드이다.
  - 실행 결과 데이터의 형상이 (200, 5)인 것을 볼 수 있음
    - 관측치가 200개, 5개의 속성으로 구성됨

```
df = pd.read_excel(os.getcwd()+'/mall_customers_data.xlsx')
print(df.shape)   # (200, 5)
```

02 | 비지도 학습: K-평균 군집 분석

- 다음은 Mall Customers Clustering Analysis 데이터 집합을 앞에서 5개 행을 출력하는 코드이다.
  - 실행 결과 아래와 같은 데이터로 구성된 것을 볼 수 있음

```
df.head()
```

	고객ID	성별	나이	연간소득	지출점수(1-100)
0	1	남자	19	19500000	39
1	2	남자	21	19500000	81
2	3	여자	20	20800000	6
3	4	여자	23	20800000	77
4	5	여자	31	22100000	40

02 | 비지도 학습: K-평균 군집 분석

- 다음은 Mall Customers Clustering Analysis 데이터 집합의 "성별", "나이", "연간소득", "지출점수(1-100)" 속성으로 K-평균 군집 분석을 수행해 보자.

  - Mall Customers Clustering Analysis 데이터 집합에서 "성별", "나이", "연간소득", "지출점수(1-100)" 속성을 선택하여 변수 X에 할당함

```
X = df[['성별','나이','연간소득','지출점수(1-100)']]
```

02 | 비지도 학습: K-평균 군집 분석

- 다음은 **범주형 변수** “성별” 의 **데이터**를 **숫자형**으로 **변환**하는 코드이다.
  - 예를 들어 성별의 경우 “여성” → 1, “남성” → 0으로 변환됨

```
label_encoder = LabelEncoder()
X['성별'] = label_encoder.fit_transform(X['성별'])
```

02 | 비지도 학습: K-평균 군집 분석

다음은 표준화 함수를 이용한 데이터 정규화를 수행하는 코드이다.

- 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled
```

	성별	나이	연간소득	구매액
0	0	19	19500000	39
1	0	21	19500000	81
2	1	20	20800000	6
3	1	23	20800000	77
4	1	31	22100000	40

정규화 이전

```
array([[-1.12815215, -1.42456879, -1.73899919, -0.43480148],
       [-1.12815215, -1.28103541, -1.73899919,  1.19570407],
       [ 0.88640526, -1.3528021 , -1.70082976, -1.71591298],
       [ 0.88640526, -1.13750203, -1.70082976,  1.04041783],
       [ 0.88640526, -0.56336851, -1.66266033, -0.39597992],
       [ 0.88640526, -1.20926872, -1.66266033,  1.00159627],
       [ 0.88640526, -0.27630176, -1.62449091, -1.71591298],
       [ 0.88640526, -1.13750203, -1.62449091,  1.70038436],
```

정규화 이후

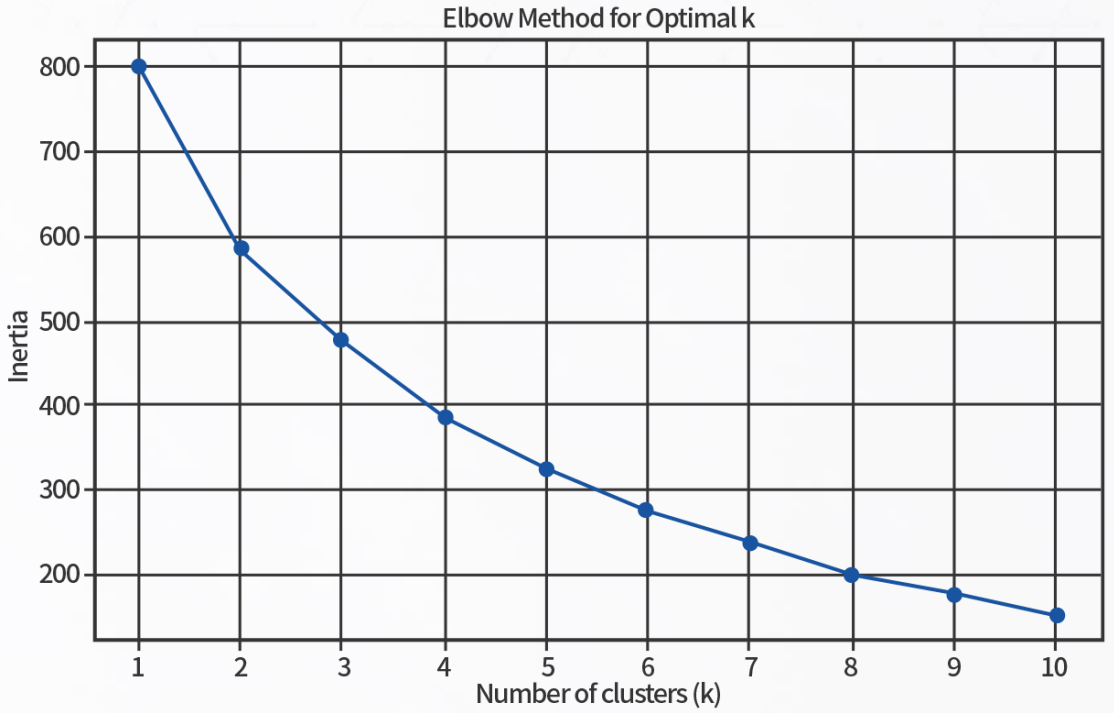
02 | 비지도 학습: K-평균 군집 분석

- 다음은 엘보우(elbow)방법을 이용해 최적의 K값을 찾는 코드이다.
  - 최적의 K값을 찾기 위해 K=1~10까지 범위로 오차 제곱합(SSE)의 변화를 시각화함
    - 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
# 엘보우 방법을 사용하여 최적의 k 값 찾기
inertia = []
k_range = range(1, 11)

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# 그래프로 그리기
plt.figure(figsize=(10, 6))
plt.plot(k_range, inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_range)
plt.grid(True)
plt.show()
```

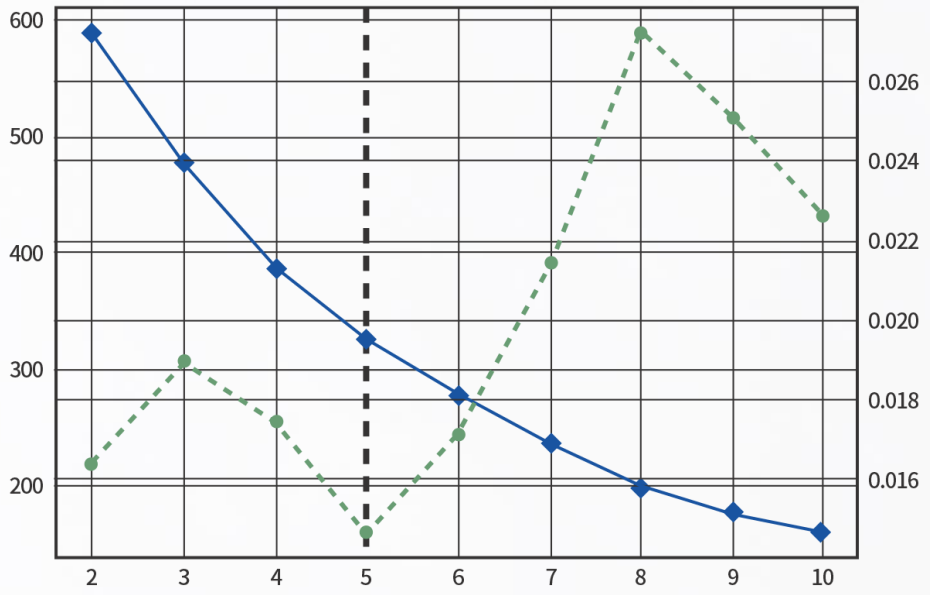


02 | 비지도 학습: K-평균 군집 분석

- 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.
  - 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
    - 검정색 점선의 위치를 보았을 때, 여기서는 K=5인 경우를 추천해주고 있음
    - 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,11))
visualizer.fit(X_scaled)
```



02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 5로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.

- 아래와 같이 “성별”, “나이”, “연간소득”, “지출점수(1-100)” 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit_predict(X_scaled)

# 각 군집의 중심값 확인
centroids = kmeans.cluster_centers_
centroids_df = pd.DataFrame(centroids, columns=X.columns)
centroids_df
```

	성별	나이	연간소득	지출점수(1-100)
0	0.886405	-0.753876	0.009161	0.716434
1	-1.128152	-0.760727	0.054964	0.833693
2	0.886405	0.708404	-0.548646	-0.408619
3	-1.128152	1.223854	-0.449858	-0.442315
4	-0.218352	0.112627	1.123708	-1.337716

02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 5로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.

아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=5, random_state=42)
X['군집'] = kmeans.fit_predict(X_scaled)
df['군집'] = kmeans.fit_predict(X_scaled)
```

	성별	나이	연간소득	지출점수(1-100)	군집
0	0	19	19500000	39	1
1	0	21	19500000	81	1
2	1	20	20800000	6	2
3	1	23	20800000	77	0
4	1	31	22100000	40	2

X 변수

	고객ID	성별	나이	연간소득	지출점수(1-100)	군집
0	1	남자	19	19500000	39	1
1	2	남자	21	19500000	81	1
2	3	여자	20	20800000	6	2
3	4	여자	23	20800000	77	0
4	5	여자	31	22100000	40	2

df 변수

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 고객 수를 확인하는 코드이다.

- 아래와 같이 5개 군집의 고객 수를 확인할 수 있음

```
cluster_counts = df['군집'].value_counts()
print(cluster_counts)
```

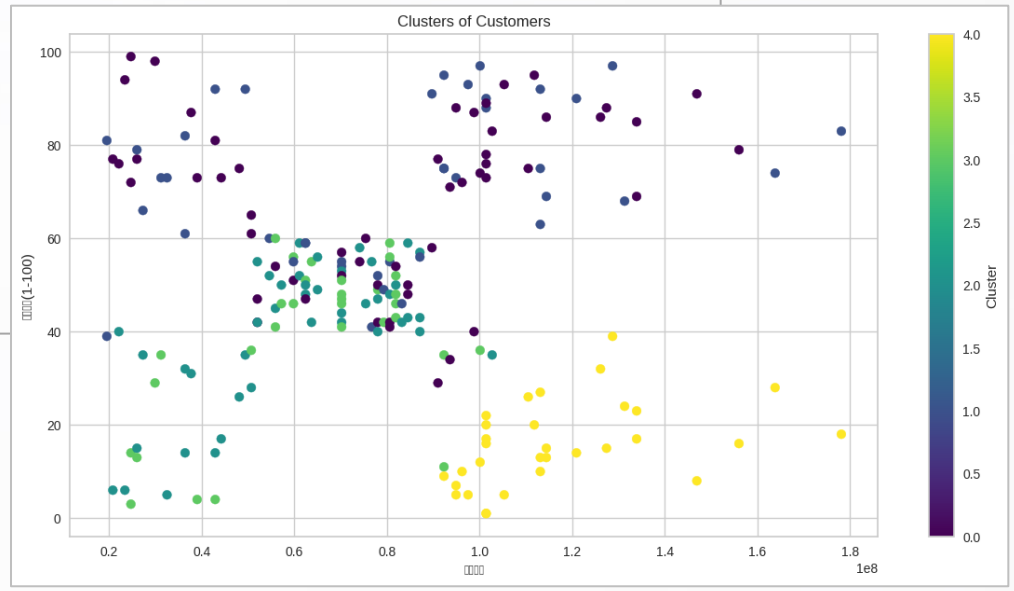
```
군집
0    55
2    43
1    40
3    31
4    31
Name: count, dtype: int64
```

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 시각화를 구현하는 코드이다.

- 아래 그림은 “성별”, “나이”, “연간소득”, “지출점수(1-100)” 속성, K=5 값으로 K-평균 군집 결과 “연간소득” 과 “지출점수(1-100)” 군집별 시각화 결과임
  - 시각화 결과에서 **5개 군집**으로 **잘 분류되지 않는 것**을 볼 수 있음

```
plt.figure(figsize=(14, 7))
plt.scatter(df['연간소득'], df['지출점수(1-100)'], c=df['군집'], cmap='viridis',
marker='o')
plt.title('Clusters of Customers')
plt.xlabel('연간소득')
plt.ylabel('지출점수(1-100)')
plt.colorbar(label='Cluster')
plt.show()
```



02 | 비지도 학습: K-평균 군집 분석

다음은 Mall Customers Clustering Analysis 데이터 집합의 "나이", "연간소득", "지출점수(1-100)" 속성으로 K-평균 군집 분석을 수행해 보자.

- Mall Customers Clustering Analysis 데이터 집합에서 "나이", "연간소득", "지출점수(1-100)" 속성을 선택하여 변수 df_selected에 할당함

```
# 필요한 속성 선택
df_selected = df[['나이', '연간소득', '지출점수(1-100)']]

df_selected
```

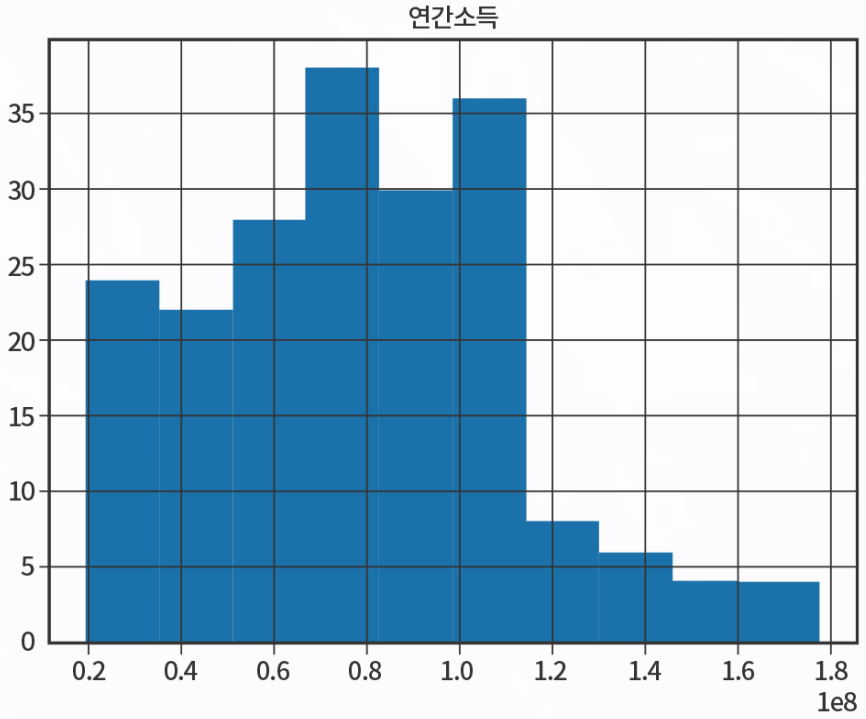
	나이	연간소득	지출점수(1-100)
0	19	19500000	39
1	21	19500000	81
2	20	20800000	6
3	23	20800000	77
4	31	22100000	40
...	...	...	...
495	35	156000000	79

02 | 비지도 학습: K-평균 군집 분석

다음은 "연간소득" 변수로 히스토그램을 그리는 코드이다.

- 아래의 히스토그램 그래프와 같이 연간소득은 0 ~ 180,000,000범위에 있음

```
pd.DataFrame(df_selected, columns=['연간소득']).hist()
plt.subplots_adjust(hspace=1)
plt.show()
```



02 | 비지도 학습: K-평균 군집 분석

- 다음은 **표준화 함수**로 **데이터 정규화**를 **수행**하는 코드이다.
  - 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_selected)
df_scaled
```

```
array([[-1.42456879, -1.73899919, -0.43480148],
       [-1.28103541, -1.73899919,  1.19570407],
       [-1.3528021 , -1.70082976, -1.71591298],
       [-1.13750203, -1.70082976,  1.04041783],
       [-0.56336851, -1.66266033, -0.39597992],
       [-1.20926872, -1.66266033,  1.00159627],
       [-0.27630176, -1.62449091, -1.71591298],
       [-1.13750203, -1.62449091,  1.70038436],
       [ 1.80493225, -1.58632148, -1.83237767],
       [-0.6351352 , -1.58632148,  0.84631002],
       [ 2.02023231, -1.58632148, -1.4053405 ],
```

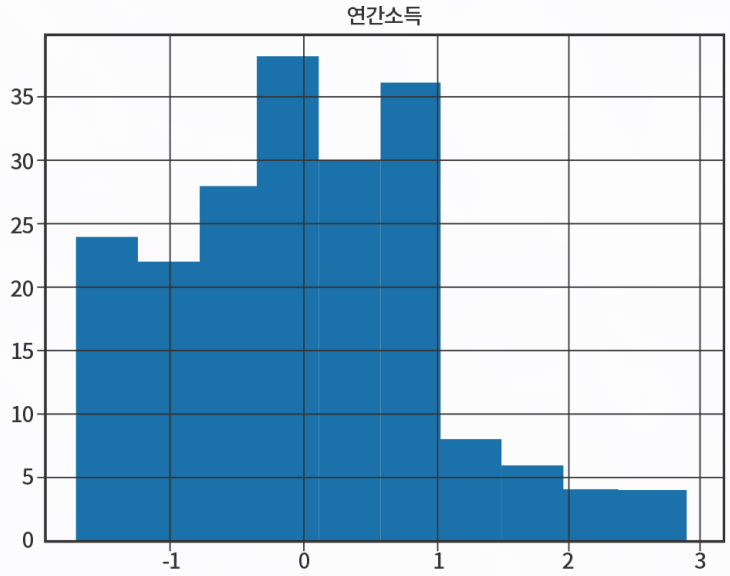
02 | 비지도 학습: K-평균 군집 분석

다음은 데이터 정규화된 "연간소득" 변수로 히스토그램을 그리는 코드이다.

- 아래의 히스토그램 그래프와 같이 구매액은 약 -2.0 ~ 3.0범위에 있음

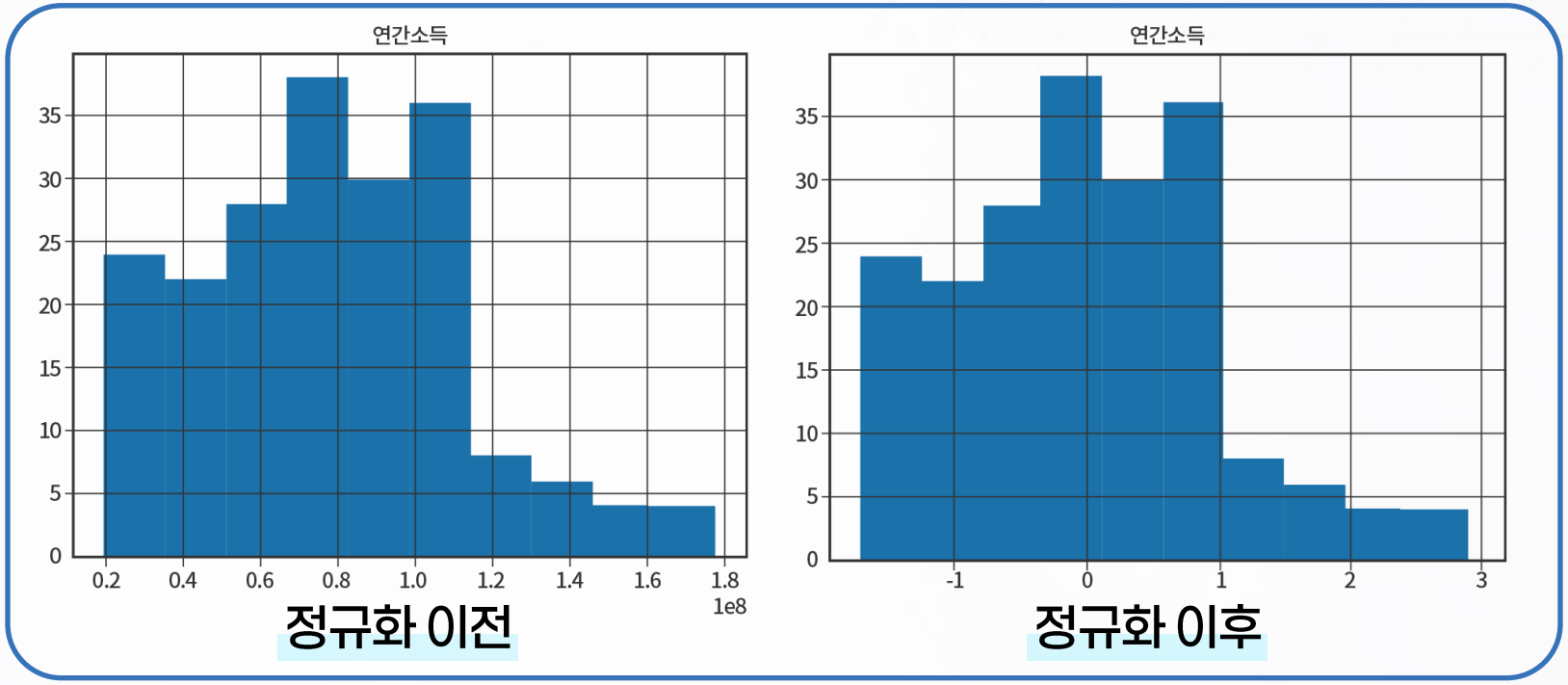
```
df_scaled_data = pd.DataFrame(df_scaled, columns=['나이','연간소득','지출점수(1-100)'])

pd.DataFrame(df_scaled_data, columns=['연간소득']).hist()
plt.subplots_adjust(hspace=1)
plt.show()
```



02 | 비지도 학습: K-평균 군집 분석

- 아래의 그림은 "연간소득" 변수의 히스토그램을 정규화 이전과 이후를 비교한 것임
  - 아래 그래프와 같이 데이터의 규격이 달라진 것을 알 수 있음
    - 정규화 이전 구매액: 0 ~ 180,000,000범위의 값
    - 정규화 이후 구매액: 약 -2.0 ~ 3.0범위의 값

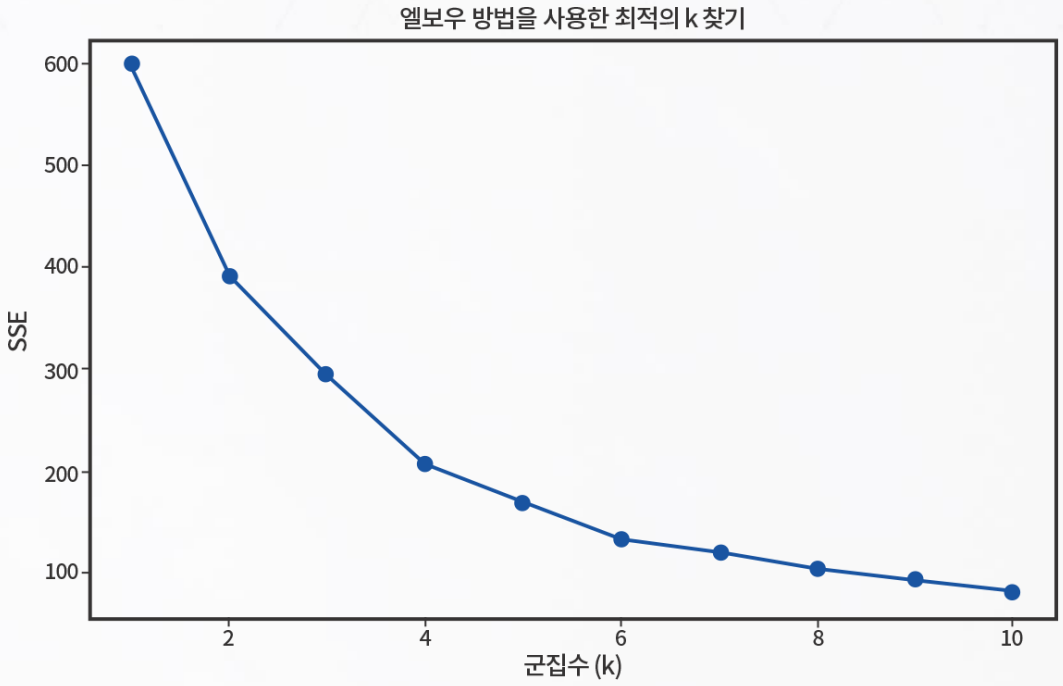


02 | 비지도 학습: K-평균 군집 분석

- 다음은 엘보우(elbow) 방법을 이용해 최적의 K값을 찾는 코드이다.
  - 최적의 K값을 찾기 위해 K=1~10까지 범위로 오차 제곱합(SSE)의 변화를 시각화함
    - 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_scaled)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse, marker='o')
plt.title('엘보우 방법을 사용한 최적의 k 찾기')
plt.xlabel('군집 수 (k)')
plt.ylabel('SSE')
plt.show()
```

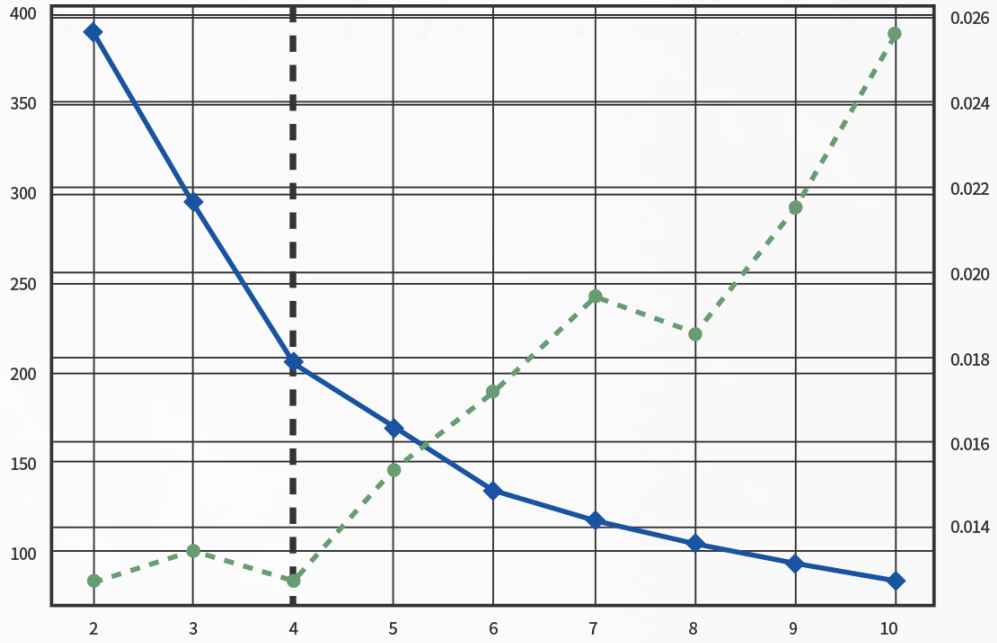


02 | 비지도 학습: K-평균 군집 분석

- 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.
  - 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
    - 검정색 점선의 위치를 보았을 때, 여기서는 K=4인 경우를 추천해주고 있음
    - 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,11))
visualizer.fit(df_scaled)
```



02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 4로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.

- 아래와 같이 “나이”, “연간소득”, “지출점수(1-100)” 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 4로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans.fit_predict(df_scaled)

# 각 군집의 중심값 확인
centroids = kmeans.cluster_centers_
centroids_df = pd.DataFrame(centroids,
columns=df_selected.columns)
centroids_df
```

	나이	연간소득	지출점수(1-100)
0	0.037205	0.990115	-1.188757
1	-0.428806	0.974847	1.216085
2	-0.962492	-0.784763	0.392030
3	1.086161	-0.490565	-0.397174

02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 4로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.

✦ 아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 4로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=4, random_state=42)
df['군집'] = kmeans.fit_predict(df_scaled)
df_selected['군집'] = kmeans.fit_predict(df_scaled)
```

	고객ID	성별	나이	연간소득	지출점수(1-100)	군집
0	1	남자	19	19500000	39	2
1	2	남자	21	19500000	81	2
2	3	여자	20	20800000	6	2
3	4	여자	23	20800000	77	2
4	5	여자	31	22100000	40	2

df 변수

	나이	연간소득	지출점수(1-100)	군집
0	19	19500000	39	2
1	21	19500000	81	2
2	20	20800000	6	2
3	23	20800000	77	2
4	31	22100000	40	2

df_selected 변수

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 고객 수를 확인하는 코드이다.

- 아래와 같이 **4개 군집**의 **고객 수**를 **확인**할 수 있음

```
cluster_counts = df['군집'].value_counts()
print(cluster_counts)
```

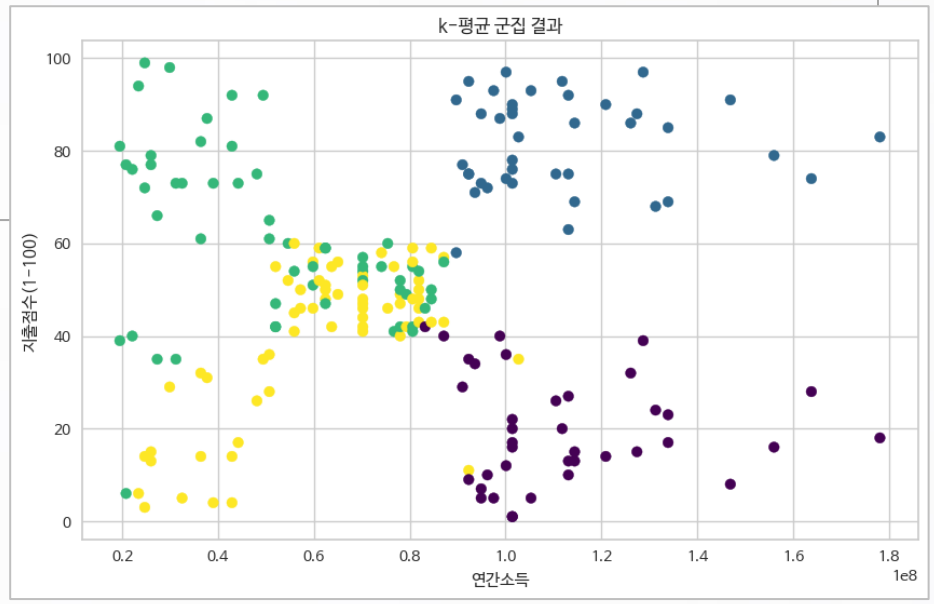
```
군집
3    65
2    57
1    40
0    38
Name: count, dtype: int64
```

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 시각화를 구현하는 코드이다.

- 아래 그림은 “나이”, “연간소득”, “지출점수(1-100)” 속성, K=4 값으로 K-평균 군집 결과 “연간소득” 과 “지출점수(1-100)” 군집별 시각화 결과임
  - 시각화 결과에서 **4개 군집**으로 **잘 분류되지 않는 것**을 볼 수 있음

```
plt.figure(figsize=(10, 6))
plt.scatter(df['연간소득'], df['지출점수(1-100)'], c=df['군집'], cmap='viridis')
plt.title('k-평균 군집 결과')
plt.xlabel('연간소득')
plt.ylabel('지출점수(1-100)')
plt.show()
```



02 | 비지도 학습: K-평균 군집 분석

다음은 Mall Customers Clustering Analysis 데이터 집합의 "연간소득", "지출점수(1-100)" 속성으로 K-평균 군집 분석을 수행해 보자.

- Mall Customers Clustering Analysis 데이터 집합에서 "연간소득", "지출점수(1-100)" 속성을 선택하여 변수 df_selected2에 할당함

```
# 필요한 속성 선택
df_selected2 = df[['연간소득','지출점수(1-100)']]

df_selected2
```

	연간소득	지출점수(1-100)
0	19500000	39
1	19500000	81
2	20800000	6
3	20800000	77
4	22100000	40
...	...	...
195	156000000	79

02 | 비지도 학습: K-평균 군집 분석

- 다음은 **표준화 함수**로 **데이터 정규화**를 **수행**하는 코드이다.
  - 아래와 같이 데이터 정규화 작업이 수행된 것을 볼 수 있음

```
scaler = StandardScaler()
df_scaled2 = scaler.fit_transform(df_selected2)
df_scaled2
```

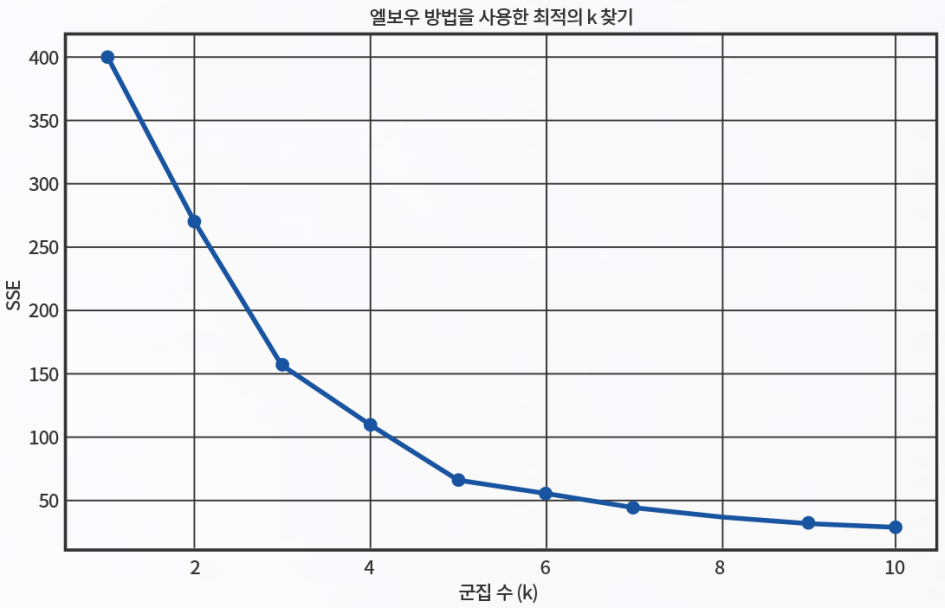
```
array([[-1.73899919, -0.43480148],
       [-1.73899919,  1.19570407],
       [-1.70082976, -1.71591298],
       [-1.70082976,  1.04041783],
       [-1.66266033, -0.39597992],
       [-1.66266033,  1.00159627],
       [-1.62449091, -1.71591298],
       [-1.62449091,  1.70038436],
```

02 | 비지도 학습: K-평균 군집 분석

- 다음은 엘보우(elbow) 방법을 이용해 최적의 K값을 찾는 코드이다.
  - 최적의 K값을 찾기 위해 K=1~10까지 범위로 오차 제곱합(SSE)의 변화를 시각화함
    - 아래 그림에서는 최적의 K값을 추정하기가 조금 모호함을 알 수 있음

```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df_scaled2)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse, marker='o')
plt.title('엘보우 방법을 사용한 최적의 k 찾기')
plt.xlabel('군집 수 (k)')
plt.ylabel('SSE')
plt.show()
```

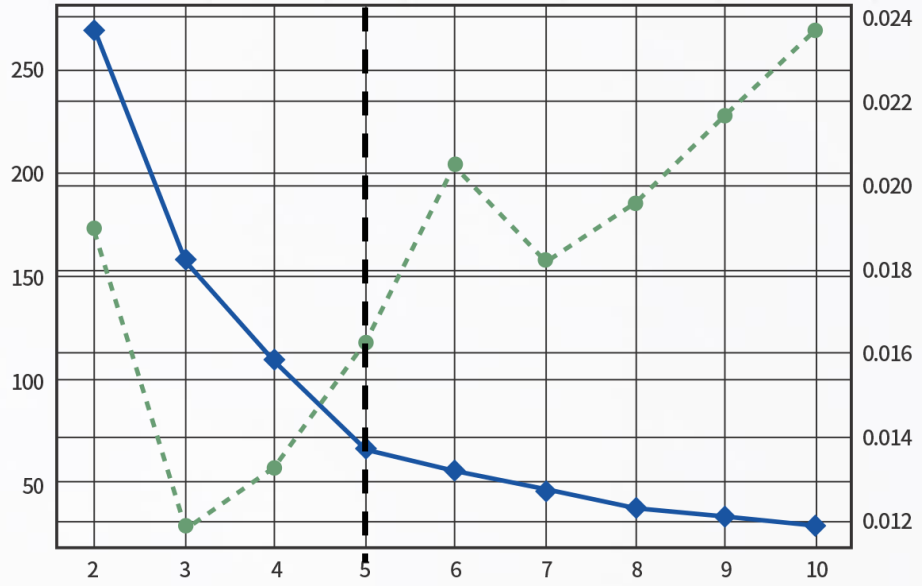


02 | 비지도 학습: K-평균 군집 분석

- 다음은 KElbowVisualizer 클래스로 최적의 K값을 선택하는 코드이다.
  - 실행결과에서 파란색이 각 데이터들의 군집 중심과의 평균 거리, 초록색은 학습 시간을 나타냄
    - 검정색 점선의 위치를 보았을 때, 여기서는 K=5인 경우를 추천해주고 있음
    - 절대적인 정답이 있는 것은 아니지만 하나의 평가 지표로 활용하면 좋을 것임

```
from yellowbrick.cluster import KElbowVisualizer

model = KMeans()
visualizer = KElbowVisualizer(model, k=(2,11))
visualizer.fit(df_scaled2)
```



02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 5로 가정하여 K-평균 군집을 수행하고, 각 군집의 중심값을 확인하는 코드이다.

- 아래와 같이 "연간소득", "지출점수(1-100)" 속성으로 각 군집의 중심값을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit_predict(df_scaled2)

# 각 군집의 중심값 확인
centroids = kmeans.cluster_centers_
centroids_df = pd.DataFrame(centroids,
columns=df_selected2.columns)
centroids_df
```

	연간소득	지출점수(1-100)
0	-0.200913	-0.026456
1	1.055003	-1.284439
2	-1.307519	-1.136965
3	-1.329545	1.132178
4	0.991583	1.239503

02 | 비지도 학습: K-평균 군집 분석

다음은 최적의 K값을 5로 가정하여 K-평균 군집 모델로 예측을 수행하는 코드이다.

✦ 아래와 같이 학습된 모델로 예측된 것을 확인할 수 있음

```
# 최적의 k 값을 5로 가정하고 k-평균 군집 수행
kmeans = KMeans(n_clusters=5, random_state=42)
df['군집'] = kmeans.fit_predict(df_scaled2)
df_selected2['군집'] = kmeans.fit_predict(df_scaled2)
```

	고객ID	성별	나이	연간소득	지출점수(1-100)	군집
0	1	남자	19	19500000	39	2
1	2	남자	21	19500000	81	3
2	3	여자	20	20800000	6	2
3	4	여자	23	20800000	77	3
4	5	여자	31	22100000	40	2

df 변수

	연간소득	지출점수(1-100)	군집
0	19500000	39	2
1	19500000	81	3
2	20800000	6	2
3	20800000	77	3
4	22100000	40	2

df_selected2 변수

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 고객 수를 확인하는 코드이다.

- 아래와 같이 5개 군집의 고객 수를 확인할 수 있음

```
cluster_counts = df['군집'].value_counts()
print(cluster_counts)
```

```
군집
0    81
4    39
1    35
2    23
3    22

Name: count, dtype: int64
```

02 | 비지도 학습: K-평균 군집 분석

다음은 군집별 시각화를 구현하는 코드이다.

- 아래 그림은 “연간소득”, “지출점수(1-100)” 속성, K=5 값으로 K-평균 군집 결과 “나이” 와 “구매액” 군집별 시각화 결과임
  - 시각화 결과에서 **5개 군집**으로 **잘 분류된 것**을 볼 수 있음

```
plt.figure(figsize=(10, 6))
plt.scatter(df['연간소득'], df['지출점수(1-100)'], c=df['군집'],
cmap='viridis')
plt.title('k-평균 군집 결과')
plt.xlabel('연간소득')
plt.ylabel('지출점수(1-100)')
plt.show()
```