기초 통계 과제 **report**

1. **Iris** 데이터 셋 로드 결과 및 구조 확인

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# 1. 데이터 로드 및 구조 확인
iris = sns.load_dataset('iris')

print("=== Head ===")
print(iris.head())
print("\n=== Info ===")
print(iris.info())
```

위의 코드를 실행한 결과는 다음과 같다.

```
=== Head ===
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa

=== Info ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

info()를 통해 5개의 데이터들을 출력한 결과, 150개의 데이터가 5개의 특성값을 가지고 있는 데이터셋이며
각각 speal_length(꽃받침 길이), sepal_width(꽃받침 너비), petal_length(꽃잎 길이), petal_width(꽃잎 너비), species(종)이다. 이러한 특성값들 중 speal_length, sepal_width, petal_length, petal_width은 실수형 데이터로 species는 개체형 데이터 임을 알 수 있다.

2. **Petal length(**꽃잎 길이**)**의 기술통계량

```
# 2. 기술통계량 (Species별 Petal Length을 분석)
# 평균, 개수, 표준편차 및 최소, 최대, 사분위수 확인
desc_stats =
iris.groupby('species')['petal_length'].describe()
print("\n=== Descriptive Statistics (Petal Length by Species)
===")
print(desc_stats)
```

위의 코드를 실행한 결과는 다음과 같다.

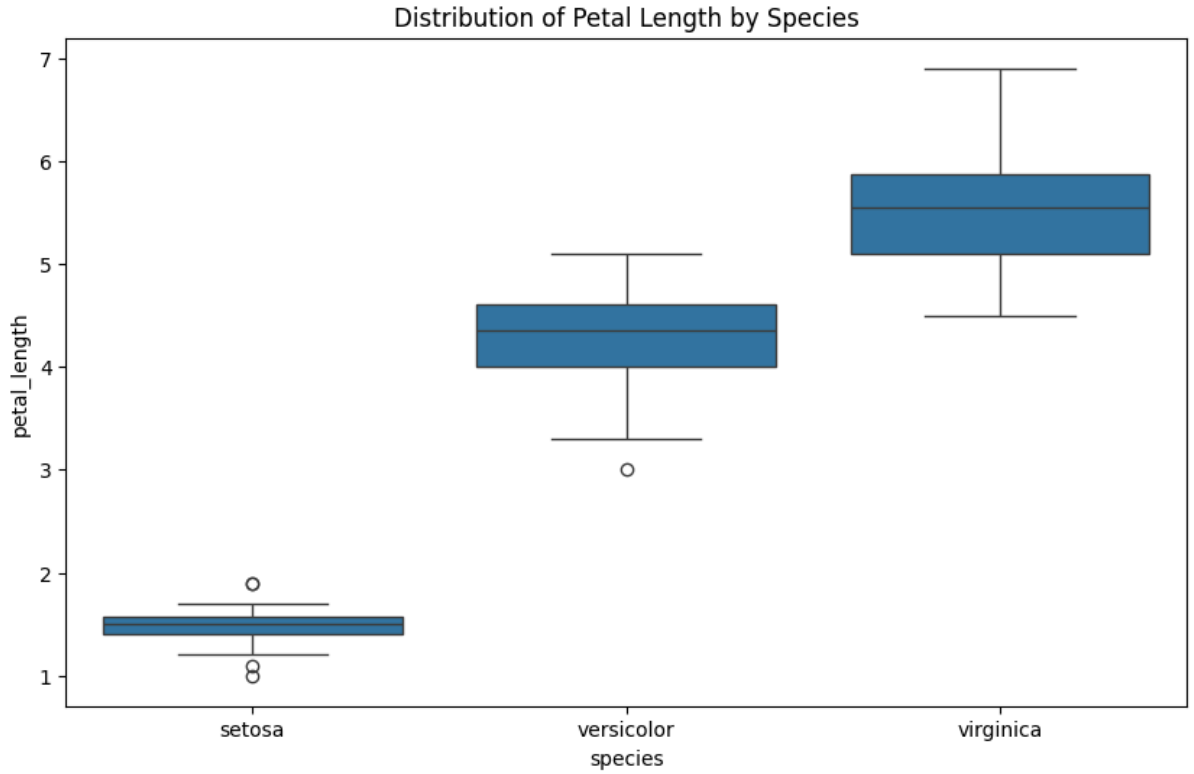
```
...
    === Descriptive Statistics (Petal Length by Species) ===
                count   mean       std  min  25%   50%    75%  max
    species
    setosa       50.0  1.462  0.173664  1.0  1.4  1.50  1.575  1.9
    versicolor   50.0  4.260  0.469911  3.0  4.0  4.35  4.600  5.1
    virginica    50.0  5.552  0.551895  4.5  5.1  5.55  5.875  6.9
```

각 종별로 50개의 데이터들을 가지고 있음을 알 수 있고, 각각의 평균값(mean), 표준편차(std), 최솟값(min), 최댓값(max), 사분위수(25%, 50%, 75%) 을 위의 결과값을 통해 알 수 있다.

3. **Petal length(**꽃잎 길이**)**의 기술 통계량 시각화 및 분석

```
# 3. 시각화 (Boxplot 포함)
plt.figure(figsize=(10, 6))
sns.boxplot(x='species', y='petal_length', data=iris)
plt.title('Distribution of Petal Length by Species')
plt.show()
```

위의 코드를 실행한 결과는 다음과 같다.



위의 기술통계량과 그래프를 분석하였을 때 setosa 종의 꽃잎 길이의 평균이 가장 작고 virginica 종의 꽃잎 길이의 평균이 가장 크며, setosa 종의 분포가 가장 조밀하며 virginica 종의 분포가 가장 넓은 것을 알 수 있다.

4. 정규성 검정

코드를 실행하기에 앞서 이 보고서에서는 Iris 데이터셋에 있는 각 종들의 데이터 들이 petal_length의 특성값에 대하여 정규 분포를 따를 것이다라는 명제를 귀무 가설로 설정한다.

```
from scipy import stats

# 4. 정규성 검정 (Shapiro-Wilk)
print("\n=== Shapiro-Wilk Test (Normality) ===")
species_list = iris['species'].unique()

for sp in species_list:
    stat, p_val = stats.shapiro(iris[iris['species'] ==
sp]['petal_length'])
    print(f"Species: {sp}, p-value: {p_val:.5f}")
```

위의 코드를 실행한 결과는 다음과 같다.

```
•••
    === Shapiro-Wilk Test (Normality) ===
    Species: setosa, p-value: 0.05481
    Species: versicolor, p-value: 0.15848
    Species: virginica, p-value: 0.10978
```

scipy.stats.shapiro()함수를 통해 얻은 p-value 값이 모든 종에서 0.05보다 크기 때문에 petal_length 데이터가 정규 분포를 따른다는 것을 알 수 있다.

5. 등분산성 검정

코드를 실행하기에 앞서 이 보고서에서는 Iris 데이터셋에 있는 각 종들의 데이터 들이 petal_length의 특성값에 대하여 분포가 동일하다라는 명제를 귀무 가설로 설정한다.

```
# 5. 등분산성 검정 (Levene)
print("\n=== Levene Test (Homoscedasticity) ===")
# 각 종별 데이터 분리
setosa = iris[iris['species'] == 'setosa']['petal_length']
versicolor = iris[iris['species'] ==
'versicolor']['petal_length']
virginica = iris[iris['species'] ==
'virginica']['petal_length']

stat, p_val = stats.levene(setosa, versicolor, virginica)
print(f"Levene Result: p-value: {p_val:.10f}")
```

위의 코드를 실행한 결과는 다음과 같다.

```
...
    === Levene Test (Homoscedasticity) ===
    Levene Result: p-value: 0.0000000313
```

scipy.stats.levene()함수를 통해 얻은 p-value 값이 0.0000000313으로로 유의 수준인 0.05보다 작기 때문에 귀무가설을 기각하고 각 종의 petal_length 데이터가 비슷한 분포를 따른지 않는다는 것을 알 수 있다.
하지만 과제에서 제시하였으므로 이후 분석은 등분산성을 만족한다고 가정하고 분석을 진행한다.

6. ANOVA 가설 수립

이 보고서에서 '세 species 사이에 petal length의 평균 차이는 없다'를 귀무 가설로 설정한다.
이에 따른 대립 가설은 '적어도 하나 이상의 집단의 평균은 다르다'이다.

7. One-way ANOVA

위에서 설정한 귀무가설을 검증하고자 ANOVA를 검증하고자 아래의 코드를 실행하고자 한다.

```
# 7. One-way ANOVA
f_stat, p_val = stats.f_oneway(setosa, versicolor, virginica)
print(f"\n=== One-way ANOVA ===\nF-statistic: {f_stat:.5f},
p-value: {p_val:.5e}")
```

위의 코드를 실행한 결과는 다음과 같다.

```
...
    === One-way ANOVA ===
    F-statistic: 1180.16118, p-value: 2.85678e-91
```

위의 결과를 분석하였을 때 p-value 값이 유의 수준인 0.05보다 작게 나왔으므로 적어도 하나 이상의 집단의 평균은 다르다라는 결론을 얻을 수 있다. 즉, 귀무가설인인 '세 species 사이에 petal length의 평균 차이는 없다'를 기각한다.

8. 사후검정 **(using Tukey HSD)**

위에서 ANOVA의 결과가 유의하다고 나왔기 때문에 Tukey HSD를 진행한다.

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# 8. 사후검정 (Tukey HSD)
print("\n=== Tukey HSD Post-hoc Test ===")
tukey = pairwise_tukeyhsd(endog=iris['petal_length'],
groups=iris['species'], alpha=0.05)
print(tukey)
```

위의 코드를 실행한 결과는 다음과 같다. 이때 유의수준을 설정하는 변수 alpha에 0.05로 설정한다.

```
...
    === Tukey HSD Post-hoc Test ===
       Multiple Comparison of Means - Tukey HSD, FWER=0.05
    ===================================================
      group1     group2   meandiff p-adj lower  upper  reject
    ---------------------------------------------------
        setosa versicolor    2.798   0.0 2.5942 3.0018   True
        setosa  virginica     4.09   0.0 3.8862 4.2938   True
    versicolor  virginica    1.292   0.0 1.0882 1.4958   True
    ---------------------------------------------------
```

각 specie의 사이에서 reject값을 보면 모두 True로 모든 값들이 비슷한 평균값을 가지고 있다라는 명제를 기각한다는 것을 알 수 있고, 모든 종 평균들 사이에 통계적으로 유의미한 차이가 있음을 알 수 있다. 평균값의 차이를 나타내는 meandiff를 보면 (virginica와 versicolor 사이 평균 차이)<(setosa와 versicolor 사이 평균 차이)<(setosa와 virginica 사이 평균 차이)이므로 가장 큰 차이는 setosa와 virginica 사이 임을 알 수 있다.

9. 결과 요약

ANOVA의 결과를 분석하였을 때 species 간 petal length 평균값들은 유의미한 차이가 있고, boxplot의 그래프를 참고하였을 때 virginica>versicolor>setosa 순으로 큰 평균값을 가짐을 알 수 있다. 이에 Tukey HSD 검사를 진행하였을 때 어떠한 종이 나머지 종과 유의미한 차이를 갖는 것이 아닌 모든 종이 서로 다른 종과 유의미한 차이를 가지고 있음을 알 수 있다. 또한 (virginica와 versicolor 사이 평균 차이)<(setosa와 versicolor 사이 평균 차이)<(setosa와 virginica 사이 평균 차이)이므로 virginica와 versicolor 사이의 차이보다 versicolor과 setosa 사이의 값 차이가 더 큼을 알 수 있다.

10. 회귀 분석

다른 특성들과 petal length의 상관 관계를 회귀분석하고 다른 특성값들을 바탕으로 petal length를 예측하는 모델을 학습시켜보고자 한다.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# 10. 회귀 분석
# 입력: sepal_length, sepal_width, petal_width
# 타겟: petal_length

X = iris[['sepal_length', 'sepal_width', 'petal_width']]
y = iris['petal_length']

# Train/Test 분리
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# 모델 학습
model = LinearRegression()
model.fit(X_train, y_train)

# 예측
y_pred = model.predict(X_test)

# 평가 및 해석
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
coef = model.coef_

print("\n=== Linear Regression Results ===")
print(f"MSE: {mse:.5f}")
print(f"R^2 Score: {r2:.5f}")
print(f"Coefficients: {coef}")
print(f"Features: {X.columns.tolist()}")
```

위의 코드를 실행한 결과는 다음과 같다.

```
...
    === Linear Regression Results ===
    MSE: 0.13002
    R^2 Score: 0.96033
    Coefficients: [ 0.72281463 -0.63581649  1.46752403]
    Features: ['sepal_length', 'sepal_width', 'petal_width']
```

MSE는 0.13002, R^2는 0.96033, 회귀계수는 (sepal length: 0.72281463), (sepal width: 0.63581649), (petal width: 1.46752403)이다. 이 때 MSE는 0.13002로 예측 오차가 크지 않고, R^2 또한 0.96033으로 96%이상의 정확도를 가지고 있음을 알 수 있다.
회귀 계수를 보았을 때 sepal length와 petal width는 양의 값을 가지므로 양의 상관 관계를 sepal width는 음의 값을 가지므로 음의 상관 관계를 가지는 것을 확인할 수 있었다.