데이터사이언스 _ Data Summary 201911278 정경은

목차

1. 소스 코드 및 내용 설명

파일 dataframe 으로 읽어들이기

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

# excel file -> dataframe 으로 읽어들이기
xlfile = 'db_score.xlsx'
df = pd.read_excel(xlfile)

print(df)
```

제공된 excel file 을 dataframe 으로 읽어들여 사용하였고, 데이터를 시각화 하기 위한 라이브러리인 matplolib 을 사용하였습니다.

1. mean, median – midterm, final, score 각각

```
""" 1. mean, median – midterm, final, score 에 대하여 """

midterm = df['midterm']
final = df['final']
score = df['score']

print("midterm | mean: {0}, median: {1}".format(midterm.mean(),
midterm.median()))
print("final   | mean: {0}, median: {1}".format(final.mean(),
final.median()))
print("score   | mean: {0}, median: {1}".format(score.mean(),
score.median()))
```

mean 은 데이터의 평균을, median 은 중앙값을 의미합니다. 평균은 모든 값을 더한 후, 데이터의 개수로 나눈 값이고, 중앙값은 데이터를 정렬했을 때 가운데 있는 값입니다. Pandas 라이브러리에서 dataframe 자료형에 사용할 수 있도록 제공하는 mean, median 함수를 사용하여 값을 출력하였습니다.

2. mode - grade

```
""" 2. mode – grade 에 대하여 """

grade = df['grade']

print("grade | mode: {0}".format(grade.mode().tolist()))
```

mode 는 빈도수가 가장 높은 값을 의미합니다. Pandas 라이브러리에서 제공하는 mode 함수를 사용하여 값을 출력하였습니다.

3. variance, standard deviation – midterm, final, score 각각

```
""" 3. variance, standard deviation – midterm, final, score """

print("midterm | variance: {0}, standard deviation:
{1}".format(midterm.var(), midterm.std()))
print("final   | variance: {0}, standard deviation:
{1}".format(final.var(), final.std()))
print("score   | variance: {0}, standard deviation:
{1}".format(score.var(), score.std()))
```

variance 는 분산을, standard deviation 은 표준 편차를 의미합니다. 분산은 각 값에 평균을 뺀 값을 제곱하여 모두 더하고 데이터의 개수로 나눈 값이며, 표준 편차는 분산에 루트를 씌운 값입니다. Pandas 라이브러리에서 제공하는 var(분산), std(표준편차)함수를 사용하여 값을 출력하였습니다.

4. percentile plot – midterm, final, score 각각

```
""" 4. percentile plot - midterm, final, score"""

midterm_np = midterm.to_numpy()
final_np = final.to_numpy()
score_np = score.to_numpy()

standard = np.arange(0, 101, 10)

# midterm
midterm_percentile_plot = np.percentile(midterm_np, standard)
plt.plot(standard, midterm_percentile_plot, marker='o', linewidth=1,
label='midterm')
plt.xticks(standard)
plt.yticks(np.linspace(midterm_percentile_plot[0],
midterm_percentile_plot[len(midterm_percentile_plot)-1], 11))
plt.legend()
plt.show()

# final
final_percentile_plot = np.percentile(final_np, standard)
plt.plot(standard, final_percentile_plot, marker='o', linewidth=1,
label='final')
plt.xticks(standard)
plt.yticks(np.linspace(final_percentile_plot[0],
final_percentile_plot[len(final_percentile_plot)-1], 11))
plt.legend()
plt.show()

# score
score_percentile_plot = np.percentile(score_np, standard)
plt.plot(standard, score_percentile_plot, marker='o', linewidth=1,
label='score')
plt.xticks(standard)
plt.yticks(np.linspace(score_percentile_plot[0],
score_percentile_plot[len(score_percentile_plot)-1], 11))
plt.legend()
plt.show()
```

percentiles 은 백분위수를 의미합니다. Numpy 라이브러리에서 제공하는 percentile 함수를 사용하여, 0%, 10%, 20%, ..., 100%에 해당하는 값을 구하고, `matplotlib.pyplot` 라이브러리를 사용하여 그래프로 출력하였습니다.

- xticks, yticks: x, y 축 최솟값, 최댓값, 간격 지정
- legend: 범례 표시

5. boxplot – midterm, final, score 각각

```
""" 5. boxplot - midterm, final, score """

# midterm
midterm_box_plot = df.boxplot(column=['midterm'])
midterm_box_plot.plot()
plt.show()

# final
final_box_plot = df.boxplot(column=['final'])
final_box_plot.plot()
plt.show()

# score
score_box_plot = df.boxplot(column=['score'])
score_box_plot.plot()
plt.show()
```

boxplot 은 box 형태로 표현되는 방식을 의미합니다. 데이터는 Five-number summary 방식에 의해 계산됩니다. 박스로 나타나는 부분은 25% percentil(Q1)부터 75% percentile(Q3)까지이며, IQR 을 Q3-Q1 이라고 할 때, 선으로 표시되는 부분은 Q1-1.5*IQR ~ Q1, Q3 ~ Q3+1.5*IQR 입니다. Pandas 라이브러리에서 제공하는 boxplot 함수를 사용하여 결과를 출력하였습니다.

6. histogram – midterm, final, score 각각

```
""" 6. histogram - midterm, final, score """

# midterm
midterm.plot.hist()
plt.title('midterm')
plt.show()

# final
```

```
final.plot.hist()
plt.title('final')
plt.show()

# score
score.plot.hist()
plt.title('score')
plt.show()
```

histogram 은 막대로 표시된 표를 의미합니다. 이는 각 범주에 대하여 빈도의 비율에 대하여 보여줍니다. hist 함수를 사용하여 결과를 출력하였습니다.

7. scatter plot – midterm, final, score 각각

```
""" 7. scatter plot - midterm, final, score """

scatter_plot=df.plot.scatter(x='midterm', y='final')
scatter_plot.plot()
plt.show()

scatter_plot=df.plot.scatter(x='midterm', y='score')
scatter_plot.plot()
plt.show()

scatter_plot=df.plot.scatter(x='final', y='score')
scatter_plot.plot()
plt.show()
```

scatter plot 은 두 variable 사이의 관계를 보여줍니다. midterm, final, score 에 대하여 가능한 조합인 (midterm, final), (midterm, score), (final, score)에 대하여, scatter 함수를 사용하여 결과를 출력하였습니다.

2. 결과 캡처

1. mean, median – midterm, final, score 각각

In [2]:
```
""" 1. mean, median - midterm, final, score에 대하여 """

midterm = df['midterm']
final = df['final']
score = df['score']

print("midterm | mean: {0}, median: {1}".format(midterm.mean(), midterm.median()))
print("final   | mean: {0}, median: {1}".format(final.mean(), final.median()))
print("score   | mean: {0}, median: {1}".format(score.mean(), score.median()))
```

```
midterm | mean: 23.560978260869547, median: 24.5
final   | mean: 14.148369565217388, median: 13.65
score   | mean: 64.0441304347826, median: 67.58500000000001
```

2. mode – grade

In [3]:
```
""" 2. mode - grade에 대하여 """

grade = df['grade']

print("grade | mode: {0}".format(grade.mode().tolist()))
```

```
grade | mode: ['A', 'B', 'C', 'D']
```

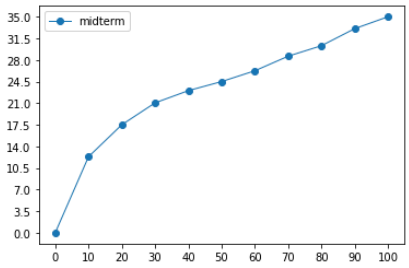
3. variance, standard deviation – midterm, final, score 각각

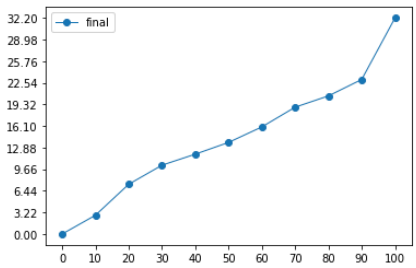
In [4]:
```
""" 3. variance, standard deviation - midterm, final, score """

print("midterm | variance: {0}, standard deviation: {1}".format(midterm.var(), midterm.std()))
print("final   | variance: {0}, standard deviation: {1}".format(final.var(), final.std()))
print("score   | variance: {0}, standard deviation: {1}".format(score.var(), score.std()))
```

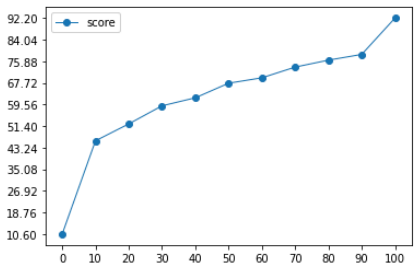
```
midterm | variance: 62.99816057095081, standard deviation: 7.9371380592094285
final   | variance: 60.56947533444818, standard deviation: 7.782639355286109
score   | variance: 231.29331242236023, standard deviation: 15.208330362743974
```

4. percentile plot – midterm, final, score 각각

In [14]:
```
""" 4. percentile plot - midterm, final, score"""

midterm_np = midterm.to_numpy()
final_np = final.to_numpy()
score_np = score.to_numpy()

standard = np.arange(0, 101, 10)

# midterm
midterm_percentile_plot = np.percentile(midterm_np, standard)
plt.plot(standard, midterm_percentile_plot, marker='o', linewidth=1, label='midterm')
plt.xticks(standard)
plt.yticks(np.linspace(midterm_percentile_plot[0], midterm_percentile_plot[len(midterm_percentile_plot)-1], 11))
plt.legend()
plt.show()

# final
final_percentile_plot = np.percentile(final_np, standard)
plt.plot(standard, final_percentile_plot, marker='o', linewidth=1, label='final')
plt.xticks(standard)
plt.yticks(np.linspace(final_percentile_plot[0], final_percentile_plot[len(final_percentile_plot)-1], 11))
plt.legend()
plt.show()

# score
score_percentile_plot = np.percentile(score_np, standard)
plt.plot(standard, score_percentile_plot, marker='o', linewidth=1, label='score')
plt.xticks(standard)
plt.yticks(np.linspace(score_percentile_plot[0], score_percentile_plot[len(score_percentile_plot)-1], 11))
plt.legend()
plt.show()
```

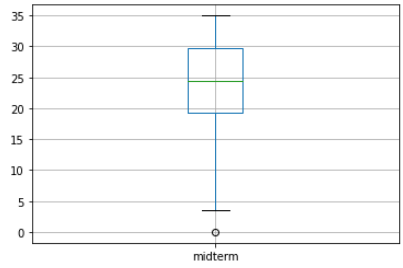


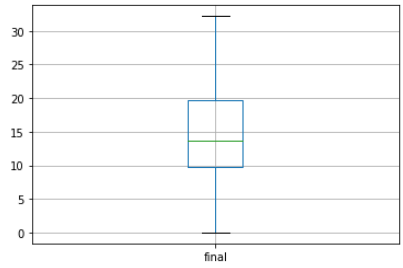


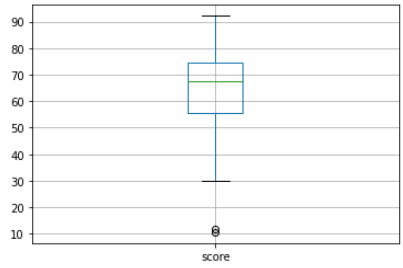


5. boxplot – midterm, final, score 각각

In [6]:
```
""" 5. boxplot - midterm, final, score """

# midterm
midterm_box_plot = df.boxplot(column=['midterm'])
midterm_box_plot.plot()
plt.show()

# final
final_box_plot = df.boxplot(column=['final'])
final_box_plot.plot()
plt.show()

# score
score_box_plot = df.boxplot(column=['score'])
score_box_plot.plot()
plt.show()
```

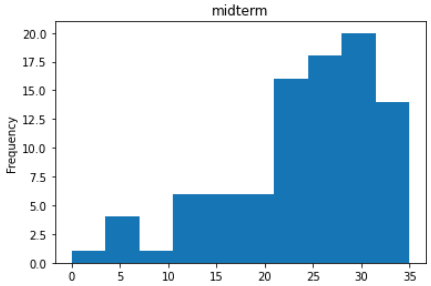


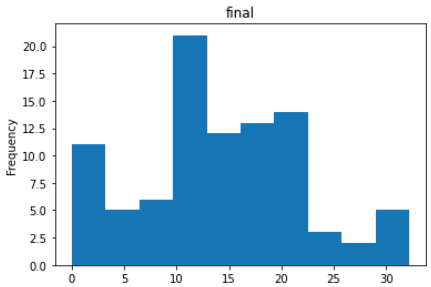


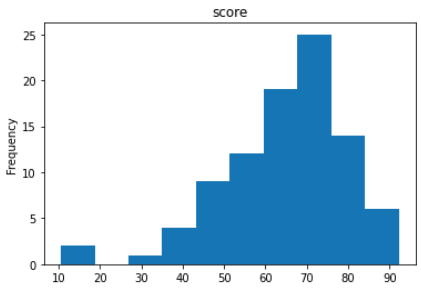


6. histogram – midterm, final, score 각각

In [7]:
```
""" 6. histogram - midterm, final, score """

# midterm
midterm.plot.hist()
plt.title('midterm')
plt.show()

# final
final.plot.hist()
plt.title('final')
plt.show()

# score
score.plot.hist()
plt.title('score')
plt.show()
```







7. scatter plot – midterm, final, score 각각

In [8]:
```
""" 7. scatter plot - midterm, final, score """

scatter_plot=df.plot.scatter(x='midterm', y='final')
scatter_plot.plot()
plt.show()

scatter_plot=df.plot.scatter(x='midterm', y='score')
scatter_plot.plot()
plt.show()

scatter_plot=df.plot.scatter(x='final', y='score')
scatter_plot.plot()
plt.show()
```