Дәріс 8. Python бағдараламалау тілінде сипаттамалық статистика

Сипаттамалық статистика — бұл статистиканың деректерді сипаттау және талдау бойынша жұмыс істейтін саласы. Python тілінде сипаттамалық статистиканы орындау үшін жиі pandas, numpy, matplotlib және seaborn кітапханалары қолданылады.

Pandas — бұл Python тілінде деректерді өңдеу және талдау үшін арналған қуатты кітапхана. Ол әсіресе кестелік деректермен жұмыс істеуге ыңғайлы және деректерді фильтрациялау, агрегациялау, манипуляциялау сияқты тапсырмаларды жеңілдетеді. **Pandas** негізінен екі негізгі құрылыммен жұмыс істейді:

1. **DataFrame** — екі өлшемді, яғни жолдар мен бағандардан тұратын кесте түріндегі деректер құрылымы.
2. **Series** — бір өлшемді деректер құрылымы, ол негізінен деректердің бағанын немесе тізбегін білдіреді.

```
import pandas as pd

# Series мысалы
data = [1, 2, 3, 4, 5]
series = pd.Series(data)
print(series)

# DataFrame мысалы
data = {'Аты': ['Аян', 'Бекжан', 'Гүлнәр'], 'Жасы': [23, 25, 30]}
df = pd.DataFrame(data)
print(df)
```

2. Негізгі функциялар

Pandas кітапханасында деректермен жұмыс істегенде қолданылатын негізгі функциялар:

- **read_csv()**: CSV форматындағы файлды оқып, DataFrame-ге айналдырады.
- **to_csv()**: DataFrame-ді CSV файлға жазып сақтайды.
- **head()**: DataFrame немесе Series-тің алғашқы бірнеше жолын көруге мүмкіндік береді.
- **tail()**: DataFrame немесе Series-тің соңғы бірнеше жолын көруге мүмкіндік береді.
- **info()**: DataFrame туралы негізгі ақпаратты көрсетеді.
- **describe()**: Сандық бағандар үшін сипаттамалық статистиканы көрсетеді.

Pandas кітапханасының артықшылықтары

- **Жоғары өнімділік**: Pandas үлкен деректермен жұмыс істеуде өте тиімді.
- **Қарапайым және ыңғайлы API**: Деректермен жұмыс істеу оңай, себебі кітапхананың интерфейсі түсінікті және тиімді.
- **Кең мүмкіндіктер**: Pandas деректерді оқудан бастап, оларды түрлендіруге, талдауға, өңдеуге және визуализациялауға дейін көптеген операцияларды орындауға мүмкіндік береді.
- **Жақсы интеграцияланған**: Pandas басқа ғылыми кітапханалармен (мысалы, NumPy, Matplotlib) өте жақсы жұмыс істейді.

Pandas кітапханасының негіздері

1. DataFrame және Series

DataFrame — бұл кестелік деректерді сақтауға арналған негізгі құрылым. Ол жолдар мен бағандардан тұрады, әр баған түрлі деректер типтерін (сандық, мәтіндік, уақыттық және т.б.) сақтай алады.

Series — бұл бір бағанды деректер құрылымы. Ол DataFrame-нің әрбір бағаны ретінде қарастырылады.

```
python
Копировать
import pandas as pd

# Series мысалы
data = [1, 2, 3, 4, 5]
series = pd.Series(data)
print(series)
```

Python тілінде сипаттамалық статистиканың негізгі әдістері мен олардың жүзеге асырылуы келесідей:

1. Деректердің негізгі сипаттамалары

Орташа мән (Mean)

Орташа мән (немесе арифметикалық орташа) — барлық мәндердің қосындысы мен олардың санының бөлінісі.

```
import numpy as np
data = [10, 20, 30, 40, 50]
mean_value = np.mean(data)
print("Орташа мән:", mean_value)
```

Медиана (Median)

Медиана — деректерді сұрыптаған кезде екі тең бөлікке бөлетін мән. Егер деректер саны тақ болса, медиана — орталық мән. Егер жұп болса, медиана — орталық екі мәннің орташа мәні.

```
median_value = np.median(data)
print("Медиана:", median_value)
```

Мода (Mode)

Мода — жиі кездесетін мән.
Моданы табу үшін scipy.stats.mode пайдаланылады:

```
from scipy import stats
mode_value = stats.mode(data)
print("Мода:", mode_value.mode[0])
```

Стандартты ауытқу (Standard Deviation)

Стандартты ауытқу — деректердің орташа мәннен қаншалықты алшақ екендігін өлшейді.

```
std_dev = np.std(data)
print("Стандартты ауытқу:", std_dev)
```

Дисперсия (Variance)

Дисперсия — стандартты ауытқудың квадраты, деректердің таралуын өлшейді.

```
variance = np.var(data)
print("Дисперсия:", variance)
```

Минимум/Максимум (Min/Max)

Минималды және максималды мәндер.

```
min_value = np.min(data)
max_value = np.max(data)
print("Минимум:", min_value)
print("Максимум:", max_value)
```

2. pandas кітапханасын пайдалану арқылы сипаттамалық статистика

pandas кітапханасы деректердің негізгі статистикалық сипаттамаларын есептеу үшін өте ыңғайлы әдістерді ұсынады.
pandas пайдалану мысалы:

```
import pandas as pd

# DataFrame құру
data = {'Жас': [23, 25, 23, 27, 30, 25, 29, 31, 35, 27]}
df = pd.DataFrame(data)

# Негізгі статистикалық көрсеткіштер
print(df.describe())
```

describe() әдісі деректердің сандық бағандары үшін статистиканы көрсетеді:

- count — мәндер саны

- mean — орташа мән
- std — стандартты ауытқу
- min — минималды мән
- 25%, 50%, 75% — 25%-шы, 50%-шы және 75%-шы перцентильдер
- max — максималды мән

Шығу мысалы:

matlab

```
       Жас
count  10.000000
mean   27.300000
std     3.227208
min    23.000000
25%    25.000000
50%    27.000000
75%    30.000000
max    35.000000
```

3. Корреляция

Корреляция — екі айнымалының арасындағы байланысты өлшейді. Python тілінде бұл үшін pandas-тің corr() әдісін қолдануға болады.

```
# Екі деректер жиынының мысалы
data1 = [1, 2, 3, 4, 5]
data2 = [5, 4, 3, 2, 1]
df = pd.DataFrame({'X': data1, 'Y': data2})
correlation = df.corr()
print("Корреляция:", correlation)
```

df.corr() әдісі барлық сандық бағандар арасындағы корреляцияны есептейді.

4. Деректерді визуализациялау

Сипаттамалық статистиканы визуализациялау үшін matplotlib және seaborn кітапханалары қолданылады.

Гистограмма

Гистограммаларды салу үшін Matplotlib кітапханасында bar() және barh() функциялары қолданылады, олар сәйкесінше вертикальді және горизонтальді гистограммаларды салады. Бұл функциялар, басқа сурет салу функциялары сияқты, matplotlib.pyplot немесе pyplot модулінен импортталады. bar және barh функциялары қосымша параметрлермен көптеген баптауларды ұсынады, бірақ біз осы мақалада тек гистограммалардың сыртқы көрінісін баптауда жиі қолданылатын мүмкіндіктерді қарастырамыз.

Көрінісін баптаусыз график

Алдымен тек міндетті параметрлер қолданылатын ең қарапайым мысалды қарастырайық, олардың саны екеу ғана.

X осіндегі (немесе Y осіндегі barh() функциясы үшін) бағаналардың орнын анықтайтын координаттар тізімі.

Бағаналардың биіктігін (ұзындығын) көрсететін мәндер.

Бұл екі тізімнің ұзындықтары тең болуы керек.

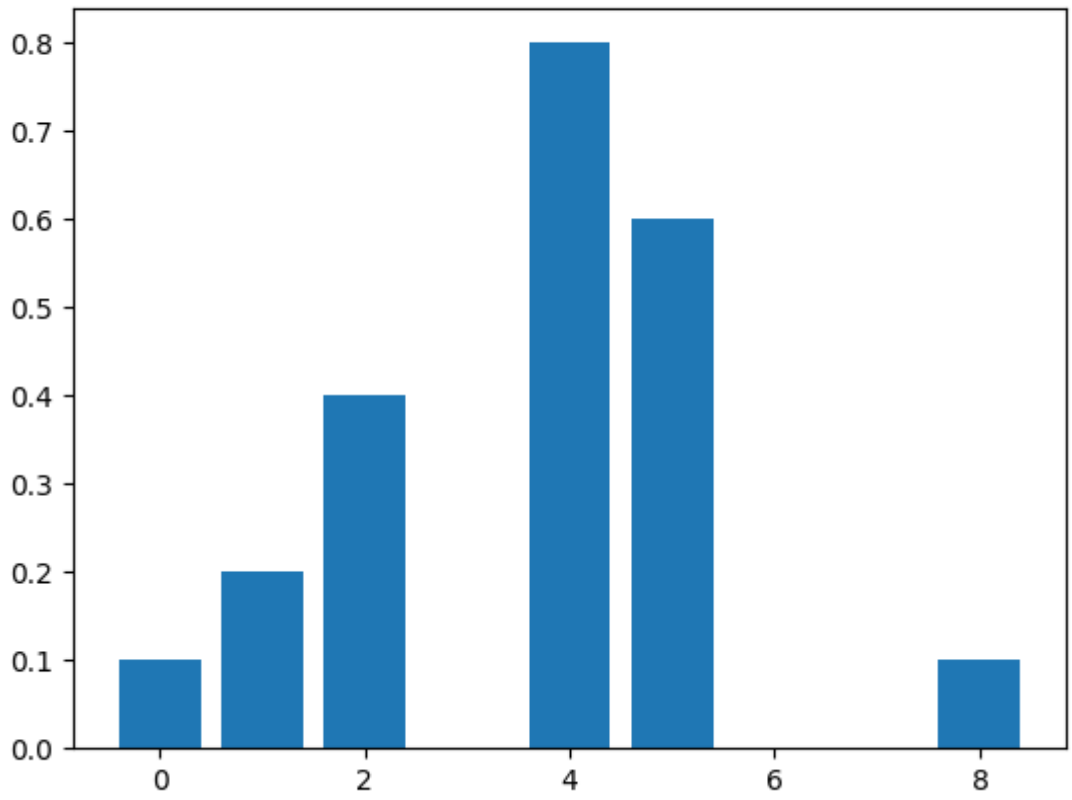
Біздің алғашқы мысалымыз осындай болады:

```
import matplotlib.pyplot as plt
```

```
if __name__ == '__main__':
    xdata = [0, 1, 2, 4, 5, 8]
    ydata = [0.1, 0.2, 0.4, 0.8, 0.6, 0.1]

    plt.bar(xdata, ydata)
    plt.show()
```

Бағдарламаны іске қосқаннан кейін келесі гистограмманы көруге болады:



Қорапты диаграмма (Boxplot)

Қорапты диаграмма деректердің медианасын, квартильдерін және аномалияларды көруге мүмкіндік береді.

```
import seaborn as sns
sns.boxplot(data=data)
plt.title('Қорапты диаграмма')
plt.show()
```

Корреляция матрицасы

Бірнеше айнымалы арасындағы байланысты зерттеу үшін корреляция матрицасы пайдалы.

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Корреляция матрицасы')
plt.show()
```

5. Толық деректерді талдау мысалы

```
python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Кездейсоқ деректерді генерациялау
np.random.seed(0)
data = np.random.randn(1000)

# Деректерді DataFrame-ге түрлендіру
df = pd.DataFrame(data, columns=['Мәндер'])

# Сипаттамалық статистика
print(df.describe())

# Деректерді визуализациялау
plt.figure(figsize=(12, 6))

# Гистограмма
plt.subplot(1, 2, 1)
plt.hist(df['Мәндер'], bins=30, edgecolor='black')
plt.title('Гистограмма')

# Қорапты диаграмма
plt.subplot(1, 2, 2)
sns.boxplot(data=df['Мәндер'])
plt.title('Қорапты диаграмма')

plt.tight_layout()
plt.show()
```

Бұл код кездейсоқ деректерді генерациялап, сипаттамалық статистиканы есептейді және деректерді гистограмма мен қорапты диаграмма арқылы визуализациялайды.

Қорытынды

Python тілінде сипаттамалық статистика деректерді талдау мен визуализациялаудың көптеген құралдарын ұсынады. numpy, pandas, matplotlib және seaborn кітапханалары деректермен жұмыс істегенде өте пайдалы болады.

Тапсырма

1. **Қатардың модасын анықта**

```
# importing the statistics library
import statistics
# creating the data set
my_set = [10, 20, 30, 30, 40, 40, 40, 50, 50, 60]
# estimating the mode of the given set
my_mode = statistics.mode( my_set)
# printing the estimated mode to the users
print("Mode of given set of data values is", my_mode)
```

2. **Қатардың медианасын тап**

```
from statistics import median
# Пример данных
numbers = [3, 1,2, 3, 5, 2]
# Получаем значение медианы
med = median(numbers)
print(med)
```

3. **Стандартты ауытқуды анықта**

```
import statistics
data = range(1,10)
res_std = statistics.stdev(data)
print(res_std)
```

4. **Арифметикалық ортасын анықта**

```
import statistics
numbers = [4, 8, 6, 5, 3, 2]
average = statistics.mean(numbers)
print(average)
```

Қосымша

В таблице pandas:statistics:tbl:1 представлен полный список методов сводной статистики и связанных с этим методов:

Таблица 6. Описательная и сводная статистика

Метод	Описание
count	Количество нечисловых значений
describe	Вычисляет сводную статистику для ряда или для каждого столбца объекта DataFrame
min, max	Вычисляет минимальное и максимальное значение
argmin, argmax	Возвращают индекс (целое число), где расположено минимальное или максимальное значение
idxmin, idxmax	Возвращают метку индекса, где расположено минимальное или максимальное значение
quantile	Вычисляет квантиль выборки от 0 до 1
sum	Сумма значений
mean	Среднее значение

Метод	Описание
median	Медиана (50-процентная квантиль) значений
mad	Среднее абсолютное отклонение от среднего значения
prod	Произведение значений
var	Дисперсия множества выборки значений
std	Стандартное отклонение выборки значений
skew	Асимметрия (третий момент) выборки значений
kurt	Эксцесс (четвертый момент) выборки значений
cumsum	Накопленная сумма значений
cummin, cummax	Совокупный минимум и максимум
cumprod	Накопленное произведение значений

Метод	Описание
diff	Вычисляет первую арифметическую разность (полезно для временных рядов)
pct_change	Вычисляет процентные изменения