Repetisjon/oppgaver i BED-1304 Python-Lab

Markus J. Aase

Forelesning 5 - Pandas og databehandling

Informasjon

Dette dokumentet er ment for å repetere og teste forståelsen av kjernepensum i **BED-1304 Python-Lab**. Denne gangen handler det om **Forelesning 5**, som dekker følgende temaer:

- Pandas biblioteket
- Databehandling

I tillegg til definisjoner og eksempler, vil du få oppgaver som kombinerer temaene fra tidligere forelesninger (variabler, uttrykk, funksjoner, lister, numpy) med dagens tema.

Repetisjon og teori

Pandas

Pandas er et av de mest brukte bibliotekene i Python for dataanalyse og behandling av tabulære data. Det bygger på `NumPy` og gjør det enkelt å jobbe med datasett i tabellform (ligner på Excel, men mye kraftigere).

Installasjon og import

For å installere Pandas kan du kjøre følgende i en Jupyter Notebook:

```
!pip install pandas
```

For å importere pakken (standard praksis er å bruke alias `pd`):

```
import pandas as pd
```

Lese inn data fra fil

For å lese inn en CSV-fil bruker vi funksjonen `read_csv`. Hvis du skal bruke en filbane på en Windows-maskin, kan det være lurt å bruke `'r'` foran filbanen, slik at spesialtegn (som `\n`) ikke feiltolkes.

OBS: For noen trenger dere ikke 'r', det avhenger av hvilke pc/mac du har.

```
data = pd.read_csv(r'DERES_PATH/schooling-gdp.csv')
data.head()
```

OBS: Hvis det over gir deg feilmelding, og sier at du mangler `openpyxl`, da kan dere kjøre linja under. Hvis dere får en annen feilmelding, kommer nok feilen fra **filbanen** dere oppga. Altså det som står mellom anførselstegnene.

```
!pip install openpyxl
```

Eksempel på data

Datasettet kan inneholde kolonner som:

Land	Landkode	BNP_per_capita	Utdanning	Befolkning
Norway	NOR	73262.680	11.65	4886000
Kuwait	KWT	67029.523	6.39	2992000
Luxembourg	LUX	57882.809	11.33	508000
Switzerland	CHE	55688.020	12.92	7809000
United States	USA	49500.629	13.24	309011008

Her har vi for eksempel land, landkode, BNP per innbygger, utdanningsnivå og befolkning. Dataene kan analyseres videre med Pandas. Dette datasettet er det vi skal se på i forelesning (Kilde: Our World In Data).

Denne filen inneholder data om BNP (bruttonasjonalprodukt) og utdanningsnivå for ulike land i 2010.

Hva er en DataFrame?

En **DataFrame** er hovedstrukturen i Pandas. Den kan sees som en tabell (lignende Excel) hvor radene har indekser og kolonnene har navn. Det gir fleksibel tilgang til data, enkel filtrering, gruppering, og statistiske beregninger.

Operasjoner på DataFrames

Vi antar at vi har kalt datarammen `data` (et annet vanlig navn dere vil se på internett er `df` for *dataframe*). Tabellen under viser noen nyttige operasjoner man kan gjøre med Pandas:

Operasjon	Formål	Syntaks i Python
Vise kolonner	Viser navnene på kolonnene	`data.columns`
Velge kolonne	Hente en spesifikk kolonne	`data['kolonnenavn']`
Filtrere rader	F.eks. land med utdanning > 10 år	`data[data['Utdanning'] > 10]`
Statistisk oversikt	Gir gjennomsnitt, min, maks osv.	`data.describe()`
Sortere data	Sortere på kolonne	`data.sort_values('kolonnenavn')`
Sjekke manglende verdier	Teller manglende data	`data.isnull().sum()`
Lage ny kolonne	Regne ut forhold mellom to kolonner	`data['BNP_per_innb'] = data['BNP_per_capita'] / data['Befolkning']`
Se datatyper	Viser typer for hver kolonne	`data.dtypes`
Informasjon om DataFrame	Antall rader, kolonner og minnebruk	`data.info()`

Dette er bare et utvalg — Pandas har svært mange funksjoner som gjør det mulig å analysere data raskt og effektivt.

Lurer dere på flere funksjoner i Pandas, kan dere enkelt søke opp Pandas sin dokumentasjon.

Eksempler

La oss se noen eksempler på hvordan `pandas` kan brukes i praksis.

```python
import pandas as pd

# Laste inn CSV-fil
data = pd.read_csv("schooling-gdp.csv")

# Vise de første radene
print(data.head())

# Velge en enkelt kolonne
print(data["BNP_per_capita"])

# Filtrere rader: land med utdanning over 10 år
print(data[data["Utdanning"] > 10])

# Lage en ny kolonne: BNP per person delt på utdanning
data["BNP_per_utdanning"] = data["BNP_per_capita"] / data["Utdanning"]
print(data.head())
```

Hva skjer her?

- `pd.read_csv(...)`: Leser en datafil (CSV-format).
- `.head()`: Viser de første 5 radene i datasettet.
- `data[BNP_per_capita"]`: Henter en bestemt kolonne.
- `data[data["Utdanning"] > 10]`: Filtrerer ut rader basert på en betingelse.
- Vi kan lage nye kolonner ut fra beregninger.

```python
# En statistisk oversikt
print(data.describe())

# Sortere land etter høyest BNP per capita
print(data.sort_values("BNP_per_capita", ascending=False).head(5))
```

Pandas gir oss dermed verktøy til å:

- Lese og lagre datafiler.
- Utforske og filtrere data.
- Gjøre beregninger og lage nye variabler.

- Få statistiske oppsummeringer.
- Sortere og analysere datasett raskt og effektivt.

Prøv selv

Gå til *«data til forelesningsnotater»* på hjemmesiden til kurset, last ned `schooling-gdp.csv` filen og prøv selv!
Test gjerne funksjonene listet opp i tabellen over.

Del 1: Flervalgsoppgaver

1. Hvilken metode brukes for å vise de første radene i en DataFrame?

   a) `data.top()`
   b) `data.head()`
   c) `data.start()`
   d) `data.show()`

2. Hvilken metode gir en statistisk oppsummering (gjennomsnitt, min, maks osv.)?

   a) `data.summary()`
   b) `data.stats()`
   c) `data.describe()`
   d) `data.info()`

3. Hvordan kan man hente ut en kolonne som heter `A`?

   a) `data[A]`
   b) `data["A"]`
   c) `data.get("A")`
   d) b) og c)

4. Hva gjør `data.isnull().sum()`?

   a) Sletter alle tomme rader
   b) Teller antall manglende verdier per kolonne
   c) Returnerer `True/False` for hver celle
   d) Ingenting, gir feil

5. Hvilken metode brukes for å sortere en DataFrame på en kolonne?

   a) `data.order()`
   b) `data.sort('kolonne')`
   c) `data.sort_values('kolonne')`
   d) `data.arrange('kolonne')`

6. Hva gir `data.shape`?

   a) Antall rader
   b) Antall kolonner
   c) Tuple med (antall rader, antall kolonner)
   d) Datatypene i kolonnene

7. Hvordan lager man en ny kolonne `C` som er summen av `A` og `B`?

   a) `data["C"] = A+B`
   b) `data.C = data.A + data.B`
   c) `data["C"] = data["A"] + data[B"]`
   d) `data.newcol("C", A+B)`

8. Hva gjør `data.dtypes`?

    a) Viser antall rader

    b) Viser antall kolonner

    c) Viser datatype for hver kolonne

    d) Konverterer alt til tekst

9. Hvordan henter man de siste 5 radene i en DataFrame?

    a) `data.last()`

    b) `data.tail()`

    c) `data.bottom()`

    d) `data.end()`

10. Hva gjør `data["A"].mean()`?

    a) Regner ut gjennomsnittet av kolonnen `A`

    b) Viser første rad i kolonnen `A`

    c) Teller antall elementer i `A`

    d) Sletter NaN-verdier i `A`

Del 2: Hva blir output?

```
1 import pandas as pd
2 data = pd.DataFrame({"A":[1,2,3], "B":[4,5,6]})
3 print(data.head())
```

```
2 print(data["A"])
```

```
3 print(data.shape)
```

```
4 print(data["A"].mean())
```

```
5 print(data.describe())
```

```
6 print(data["B"].max())
```

```
7 print(data.isnull().sum())
```

```
8 data["C"] = data["A"] * data["B"]
2 print(data)
```

Del 3: Praktiske oppgaver

OBS: Før dere kan laste inn filer, må dere ha lastet ned biblioteket `openpyxl`.
Kjør denne `!pip install openpyxl`, og `import pandas as pd` før dere kan laste inn filen.

1. Les inn en CSV-fil som heter `schooling-gdp.csv` i en DataFrame. Vis de første 5 radene.
2. Sjekk hvor mange rader og kolonner DataFrame-en har.
3. Legg til en ny kolonne som heter `GDP_thousands`, der du deler BNP per capita på 1000.
4. Finn gjennomsnittet av utdanningsår (`Utdanning`)-kolonnen.
5. Sorter dataene etter `BNP_per_capita`, og vis de 10 landene med høyest BNP.

Del 4: Utfordring

Kjør kodeblokken under:

```
import pandas as pd

# Datafilen
url = "https://raw.githubusercontent.com/uit-bed-1304-h25/uit-bed-1304-h25.
    github.io/main/data/StudentPerformanceFactors.csv"

# Leser excel fila rett fra Markus' github repo
df = pd.read_csv(url)

df # Se på dataene
```

1. Forklar med ord hva du tror dette datasettet handler om.
2. Finn ut hvilke datatype de ulike kolonnene er.
3. Sjekk om noen rader mangler verdier.

   Hint: Hvordan funker `df.isnull().sum()`?
4. Sjekk hvilke verdier kolonnen School_Typebestår av

   Hint: Bruk `df['School_Type'].value_counts()`
5. Finn hvor stor prosent av disse verdiene kommer fra privat skole, og hvor mange prosent er fra offentlig skole.

Løsningsforslag

Del 1: Flervalgsoppgaver

1: b, 2: c, 3: d, 4: b, 5: c, 6: c, 7: c, 8: c, 9: b, 10: a

Del 2: Hva blir output?

1:

```
   A  B
0  1  4
1  2  5
3  3  6
```

2:

```
0    1
1    2
2    3
Name: A, dtype: int64
```

3:

```
(3, 2)
```

4:

```
2.0
```

5:

```
              A         B
count  3.000000  3.000000
mean   2.000000  5.000000
std    1.000000  1.000000
min    1.000000  4.000000
25%    1.500000  4.500000
50%    2.000000  5.000000
75%    2.500000  5.500000
max    3.000000  6.000000
```

6:

```
6
```

7:

```
A    0
B    0
dtype: int64
```

8:

```
   A  B   C
0  1  4   4
1  2  5  10
2  3  6  18
```

Del 3: Praktiske oppgaver - Løsningsforslag

Oppgave 1

```
import pandas as pd

data = pd.read_csv("schooling-gdp.csv")
print(data.head())
```

OBS: Her må dere laste ned fila fra hjemmesiden til kurset, lokalt på PC/Mac-en deres. Deretter må dere erstatte `schooling-gdp.csv"` med den lokasjonen dere har lagret fila på. Det vil typisk se ut noe som dette: `"/Din_MAC/Skrivebord/BED-1304/schooling-gdp.csv"`

Dataene finner dere på venstre siden under *Data til forelesningsnotater"*.

Oppgave 2

```
print(data.shape)  # (rader, kolonner) - skal bli en tuple (106, 5)
```

Oppgave 3

```
data["GDP_thousands"] = data["BNP_per_capita"] / 1000
print(data.head())
```

Oppgave 4

```
print(data["Utdanning"].mean())
```

Oppgave 5

```
print(data.sort_values("BNP_per_capita", ascending=False).head(10))
```

Del 4: Utfordring - Løsningsforslag

Oppgave 1

```
import pandas as pd

# Datafilen
url = "https://raw.githubusercontent.com/uit-bed-1304-h25/uit-bed-1304-h25.
    github.io/main/data/StudentPerformanceFactors.csv"

# Leser excel fila rett fra Markus' github repo
df = pd.read_csv(url)

df
```

Når vi kjører dette får vi opp et datasett som handler om hvor mye stuendter studerer, oppmøte, sover, også videre. I tillegg, får vi en kolonne som er *Exam_ Score* som viser hva de får på eksamen (0-100 poengsum).

Oppgave 2

```
df.info()
```

Se selv, hva står under kolonne Dtype"?

Oppgave 3

```
# Sjekker hvilke verdier kolonnen "School_Type" består av
df["School_Type"].value_counts()
```

Da ser vi at det er 4598 kolonner, hvor `School_Type` er **Public**, mens i 2009 kolonner er `School_Type` satt til **Private**.

Oppgave 4

I oppgaven over fant vi hvor mange som var public, og hvor mange som var private.

```
# Finner prosentandelen av hver verdi i kolonnen "School_Type"
df["School_Type"].value_counts(normalize=True) * 100
```

I steden for at jeg forklarer hva som står over her, still deg følgende spørsmål:

1. Hva gjør `normalize=True`?
2. Hva skjer hvis du skriver `normalize=False`?
3. Hvorfor ganger jeg med 100?