Utforske filmkvalitet med statistikk i Python

Et bilde som inneholder tekst, mat, innendørs, kopp

Automatisk generert beskrivelse

I dette opplegget skal vi se på hvordan vi kan ut gjennomsnitt og standardavvik i Python. I **del 1** undersøker vi vi kan bruke gjennomsnitt og standardavvik til å si noe om hvor troverdig gjennomsnittskarakteren til en film er og i **del 2** utvider vi programmet slik at det laster ned et datasett med filmvurderinger til over 2000 filmer og bruker statistikk for å utforske dette datasettet.

Del 1: Statistikk med lister

Frøy vil bestemme hvilken film hun skal se. Valget står mellom «Pythons on a boat» og «A debuggers life». For å finne hvilken film som er best spør hun alle i klassen som har sett begge filmene om å gi dem et terningkast, og nå vil Frøy regne ut gjennomsnitterningkastet til filmene. Hun har startet med denne koden, som summer opp terningkastene:

from pylab import axhline, legend, plot, show, sqrt, ylabel   
  
alle\_pythons\_on\_a\_boat\_terningkast = [6, 2, 5, 4, 5, 3, 4, 5]  
alle\_a\_debuggers\_life\_terningkast = [4, 5, 2, 4, 3, 4, 3, 5]  
antall\_terningkast = len(alle\_pythons\_on\_a\_boat\_terningkast)  
  
pythons\_on\_a\_boat\_sum = 0  
for terningkast in alle\_pythons\_on\_a\_boat\_terningkast:  
 pythons\_on\_a\_boat\_sum += terningkast  
  
a\_debuggers\_life\_sum = 0  
for terningkast in alle\_a\_debuggers\_life\_terningkast:  
 a\_debuggers\_life\_sum += terningkast  
  
print(f"Det er {antall\_terningkast} som har sett begge filmene.")  
print(f"Terningkastsummen for Pythons on a boat er: {pythons\_on\_a\_boat\_sum}")  
print(f"Terningkastsummen for A debuggers life er: {a\_debuggers\_life\_sum}")

1. Kjør koden og svar på følgende spørsmål
2. Hvor mange er det som har sett begge filmene?
3. Hva skjer på linjen hvor det står pythons\_on\_a\_boat\_sum += terningkast?
4. Modifiser koden slik at den også regner ut gjennomsnittsterningkastet til filmene og lagrer det i variablene pythons\_on\_a\_boat\_snitt og a\_debuggers\_life\_snitt, og skriver ut resultatet i terminalvinduet. (**Hint**: Du kan regne ut gjennomsnittet ved å ta terningkastsummen og dele på antall terningkast).
5. Hvilken film likte klassen best?

Det er ofte nyttig å se på hvordan dataen fordeler seg og det gjør vi gjerne med et plott. Nedenfor er en kodesnutt som plotter terningkastene til de forskjellige filmene.

plot(alle\_pythons\_on\_a\_boat\_terningkast, "o", label="Pythons on a boat", color="orange")  
plot(alle\_a\_debuggers\_life\_terningkast, "o", label="A debuggers life", color="teal")  
ylabel("Terningkast")  
legend()

1. Kopier koden til slutten av koden du skrev i forrige oppgave, og kjør den.
2. Hvis “Pythons on a boat” er klart best, vil de oransje prikkene for det meste ligge langt over de gråblå prikkene. Er dette tilfelle?

For å lettere kunne sammenlikne filmscorene kan det være lurt å plotte gjennomsnittet til filmene som en strek. axhline(4, color="red") vil tegne en rød strek som krysser y-aksen i .

1. Bruk axhline-funksjonen til å tegne en oransje strek som treffer y-aksen i og en gråblå strek i .

Vi ser at det er veldig mye variasjon i dataene! Da kan vi jo lure på om det er noen særlig forskjell i hvor godt likt filmene er. Hvis en film får alt fra terningkast 2 til terningkast 6, er det vanskelig å vite om det er en god film eller ei, mens hvis en film stort sett alltid får terningkast 5 eller 6, kan vi mest sannsynlig stole på at det er en bra film.

Det kan med andre ord være nyttig å regne ut et mål på hvor mye datasettet varierer – hvor forskjellige er terningkastene? Hvis det ikke er noe variasjon vil alle terningkastene være lik gjennomsnittet. Kanskje vi kan måle hvor forskjellige hvert terningkast er fra gjennomsnittet? Nedenfor er en kodesnutt som forsøker akkurat dette:

pythons\_on\_a\_boat\_variasjon = 0  
for terningkast in alle\_pythons\_on\_a\_boat\_terningkast:  
 pythons\_on\_a\_boat\_variasjon += pythons\_on\_a\_boat\_snitt - terningkast

pythons\_on\_a\_boat\_variasjon /= antall\_terningkast  
print(f"Variasjonen i terningkast for Pythons on a boat er {pythons\_on\_a\_boat\_variasjon}")

1. Kopier koden over inn i scriptet du lagde tidligere i oppgaven, kjør programmet og svar på følgende spørsmål:
2. Hvor stor variasjon sier programmet at det er i terningkast for «Pythons on a boat»?
3. Stemmer det overens med det vi så i plottet?

Problemet med koden over er at vi har både «positiv variasjon» og «negativ variasjon», og når de legges sammen vil vi få null. Vi trenger en måte å passe på at variasjonen alltid er positiv, og den vanligste måten å gjøre det på er ved å kvadrere differansene før vi legger dem til pythons\_on\_a\_boat\_variasjon, og så ta kvadratroten helt til slutt. Dette gir oss det som heter for *standardavviket*.

pythons\_on\_a\_boat\_variasjon = 0  
for terningkast in alle\_pythons\_on\_a\_boat\_terningkast:  
 pythons\_on\_a\_boat\_variasjon += (pythons\_on\_a\_boat\_snitt - terningkast)\*\*2

pythons\_on\_a\_boat\_variasjon /= antall\_terningkast  
pythons\_on\_a\_boat\_standardavvik = np.sqrt(pythons\_on\_a\_boat\_variasjon  
  
print("Standardavviket av terningkast for Pythons on a boat er {pythons\_on\_a\_boat\_standardavvik}")

1. Kopier koden over inn i scriptet du lagde tidligere i oppgaven og kjør programmet. Hvor stor variasjon (standardavvik) sier programmet at det er i terningkast for Pythons on a boat? Stemmer dette resultatet bedre overens med det vi så i plottet?
2. Ta utgangspunkt i koden over og lag kode som regner ut standardavviket til terningkastene for «A debuggers life».
3. Regn ut differansen mellom gjennomsnittsvariablene pythons\_on\_a\_boat\_snitt og a\_debuggers\_life\_snitt. Er denne avstanden stor eller liten sammenliknet med standardavvikene?

Del 2: Databehandling med Pandas

Nå har vi analysert et lite datasett, men det som er fint med Python er at vi kan se på veldig store datamengder på en gang. Nedenfor er et eksempel som laster ned vurderinger som ca 600 brukere har gitt til over 2000 forskjellige filmer! Dette datasettet er en variant av MovieLens-datasettet som ble utviklet av forskere ved University of Minnesota for å lage nye anbefalingsalgoritmer.

import pandas as pd

url = "https://github.com/kodeskolen/viken\_h22/raw/dag4/filmanmeldelser-movielens.csv"  
filmkarakterer = pd.read\_csv(url, index\_col="userId")  
antall\_brukere, antall\_filmer = filmkarakterer.shape  
print(f"Vi har informasjon om hvor godt {antall\_brukere} brukere likte ulike filmer i en database med {antall\_filmer} filmer.")  
  
print("Utdrag fra filmene:")  
print(filmkarakterer)  
  
karakter\_gjennomsnitt = filmkarakterer.mean()  
fem\_beste\_filmer = karakter\_gjennomsnitt.nlargest(5)  
print("De fem filmene med høyest karaktersnitt er:")  
print(fem\_beste\_filmer)

1. Kjør kodesnutten over og svar på følgende spørsmål:
2. Hva gjør linje 3 i programmet over?
3. Hva gjør linjen karakter\_gjennomsnitt = filmkarakter.mean()?
4. Hva gjør linjen fem\_beste\_filmer = karakter\_gjennomsnitt.nlargest(5)?
5. Modifiser koden slik at du også bruker nsmallest-funksjonen til å finne de fem filmene med lavest karaktersnitt og skriv disse ut i terminalvinduet.
6. Bruk filmkarakter.std()-funksjonen til å regne ut standardavviket til karakteren til hver film og lagre dette i en variabel karakter\_standardavvik
7. Bruk karakter\_standardavvik.nlargest-funksjonen til å finne hvilke to filmer hvor filmpublikummet var mest uenig
8. Bruk karakter\_standardavvik.nsmallest-funksjonen til å finne hvilke to filmer hvor filmpublikummet var mest enig

**Utforske filmkvalitet med statistikk i Python**

Om opplegget

Kompetansemål

Dette opplegget er hovedsakelig rettet mot følgende kompetansemål fra matematikk 2P:

* bruke og vurdere val av formålstenlege sentralmål og spreiingsmål for statistisk datamateriale

Lisens

Dette opplegget er delt under en *Creative Commons Navngivelse-IkkeKommersiell-DelPåSammeVilkår 3.0* (CC-BY-NC-SA) lisens. Du kan altså dele opplegget fritt så lenge du

* + Oppgir hvor det kommer fra (Kodeskolen og Simula Learning)
  + Ikke bruker det til kommersielle formål
  + Deler videre med samme lisens (selv om du gjør endringer)

Mer informasjon om lisensen er tilgjengelig her: <https://creativecommons.org/licenses/by-nc-sa/3.0/no/>

Bilder: Pexels.com