

Velkommen!

Statistikk for ingeniører

Digital plenumstime UKE 8

Vi starter klokka 15:15

Skriv til oss i chat om du har spørsmål underveis

Det blir gjort opptak av denne timen :)

Thea Bjørnland, Institutt for matematiske fag, NTNU (Trondheim)

Merk!

Neste uke er *siste* uke i fellesmodul (uke 1-9)

Mandag neste uke (14. oktober) blir det derfor

- Informasjon om prosjektmoduler
- Gjennomgang av noen oppgaver i forberedende prosjekt
- Og litt om lineær regresjon (uke 9 fellesmodul)

Agenda for i dag: hypotesetesting

Hypotesetesting kan være en 'uvant' måte å tenke på,

men dere har gjort det før!

-> Repetisjon av oppgaver i øving 4 og 6, nå 'avslørt' som hypotesetester

Eksempel 1: Binomisk eksakt test (øving 4)

En farmasøytisk bedrift påstår at deres medisin har bivirkninger hos høyst 20 % av alle pasientene. En lege har imidlertid behandlet 14 pasienter med et spesifikt sykdomsbilde, og 6 av disse fikk bivirkninger.

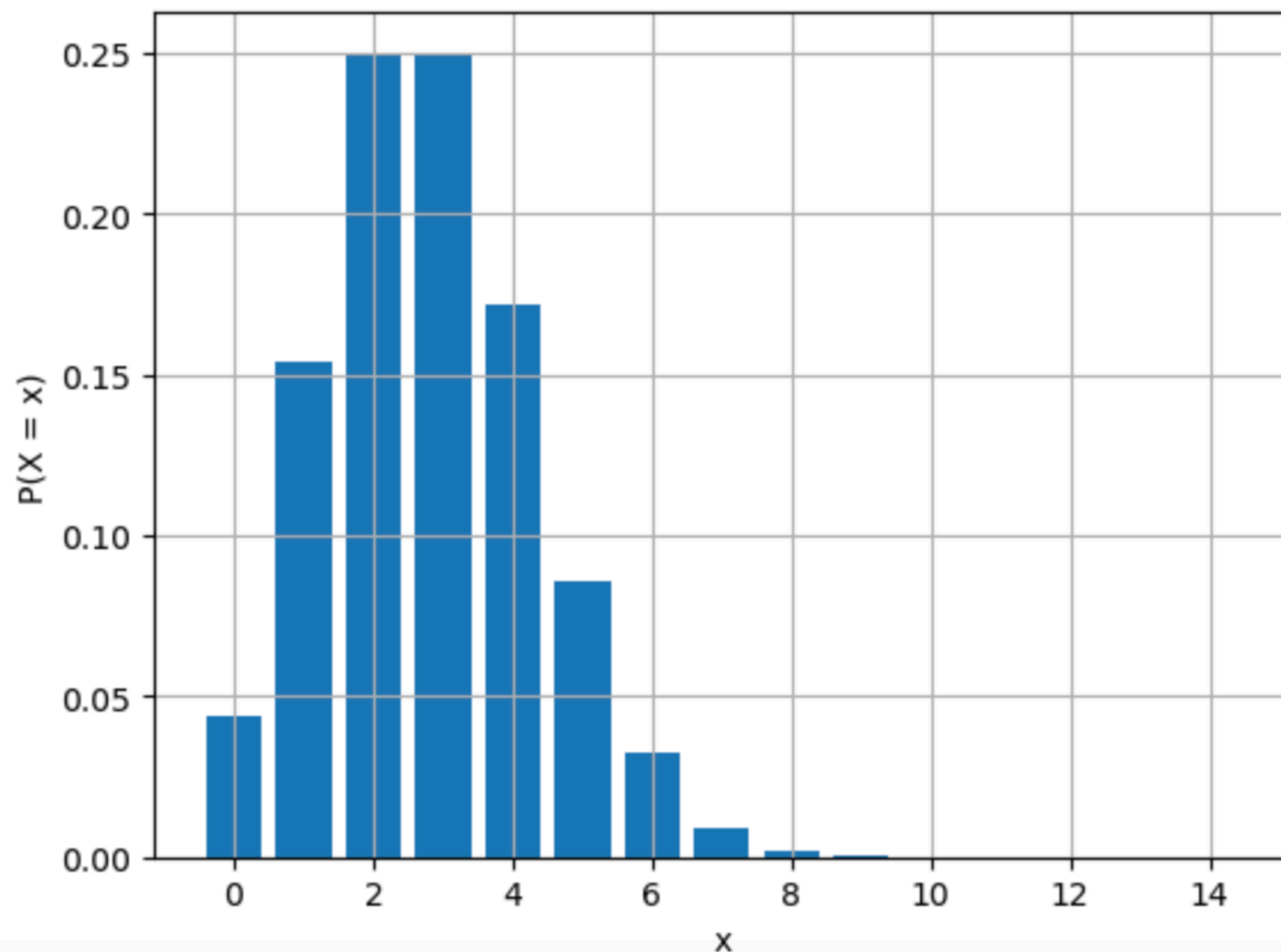
Hvor stor sannsynlighet er det for at 6 eller flere av 14 tilfeldig utvalgte pasienter får bivirkninger om sannsynligheten for bivirkninger er 20 %? Oppgi svaret som et desimaltall med fire desimaler, for eksempel 0.1234 eller 0.0001.

Eksempel 1: Binomisk eksakt test (øving 4)

Hvordan ser 'verden' ut når H_0 er sann?

```
from scipy import stats # statistikk-modulen i scipy-pakken
import matplotlib.pyplot as plt # plotting
```

```
n = 14 # antall forsøk
p = 0.2 # sannsynligheten for suksess
x_max = n+1 # hvor langt ut på x-aksen vi skal plote
plt.bar(range(0, x_max), stats.binom.pmf(range(0, x_max), n, p))
plt.grid(); plt.ylabel("P(X = x)"); plt.xlabel("x")
plt.show()
```

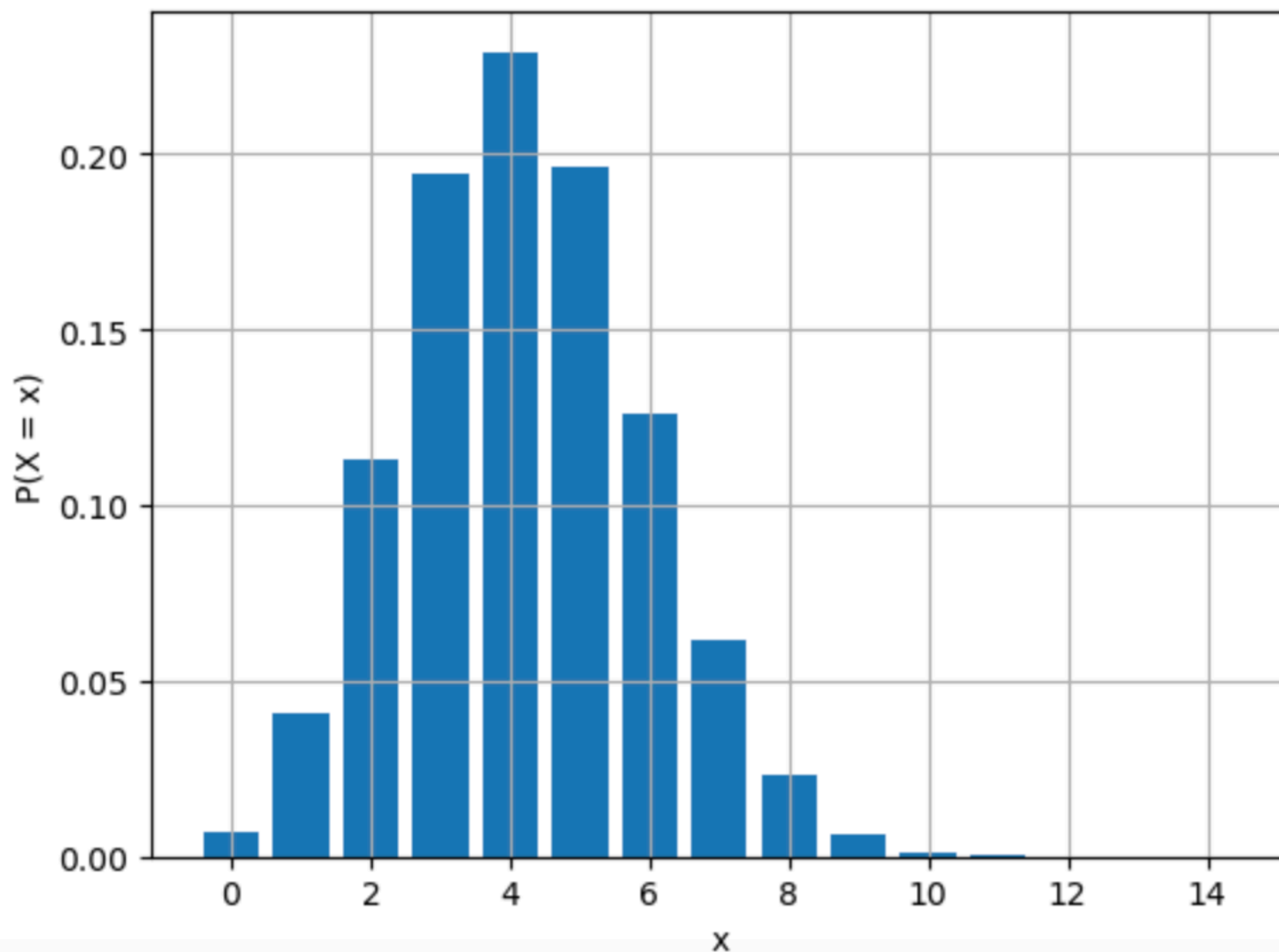


Eksempel 1: Binomisk eksakt test (øving 4)

Hvordan ser 'verden' ut når H_1 er sann?

```
: from scipy import stats # statistikk-modulen i scipy-pakken
import matplotlib.pyplot as plt # plotting

: n = 14 # antall forsøk
p = 0.3 # sannsynligheten for suksess
x_max = n+1 # hvor langt ut på x-aksen vi skal plote
plt.bar(range(0,x_max), stats.binom.pmf(range(0, x_max), n,p))
plt.grid(); plt.ylabel("P(X = x)"); plt.xlabel("x")
plt.show()
```



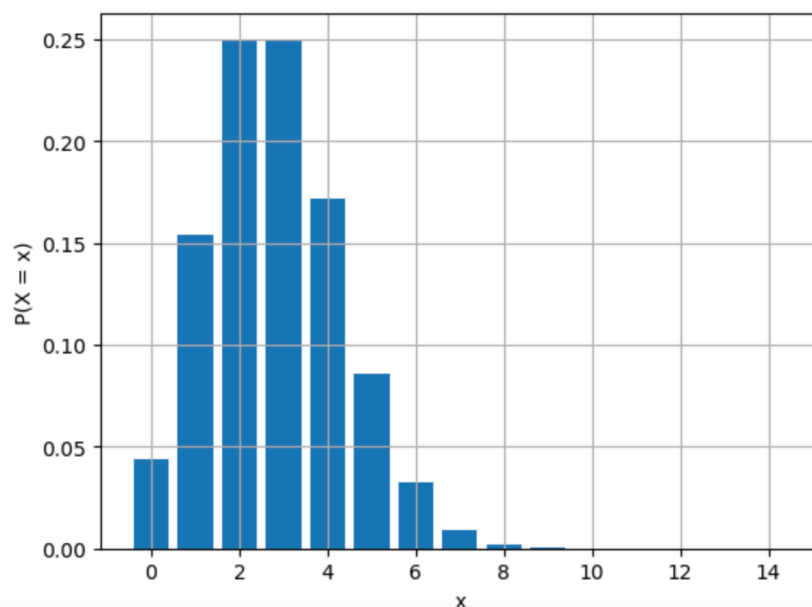
Eksempel 1: Binomisk eksakt test (øving 4)

En farmasøytisk bedrift påstår at deres medisin har bivirkninger hos høyst 20 % av alle pasientene. En lege har imidlertid behandlet 14 pasienter med et spesifikt sykdomsbilde, og 6 av disse fikk bivirkninger.

Hvor stor sannsynlighet er det for at 6 eller flere av 14 tilfeldig utvalgte pasienter får bivirkninger om sannsynligheten for bivirkninger er 20 %? Oppgi svaret som et desimaltall med fire desimaler, for eksempel 0.1234 eller 0.0001.

```
from scipy import stats # statistikk-modulen i scipy-pakken
import matplotlib.pyplot as plt # plotting
```

```
n = 14 # antall forsøk
p = 0.2 # sannsynligheten for suksess
x_max = n+1 # hvor langt ut på x-aksen vi skal plotte
plt.bar(range(0, x_max), stats.binom.pmf(range(0, x_max), n, p))
plt.grid(); plt.ylabel("P(X = x)"); plt.xlabel("x")
plt.show()
```



Eksempel 2: Oppdage kvalitetsavvik (øving 6)

Det vanligste kvalitetsavviket i en produksjonsprosess skyldes at prosessens forventningsverdi endrer seg underveis, bort fra den ønskede verdien. En slik forandring kan skyldes verktøyslitasje. For å oppdage slike avvik er det naturlig å måle gjennomsnittet $\bar{X}$ av små utvalg med gitte mellomrom^(*).

Anta at en bestemt produksjonsparameter er normalfordelt med forventningsverdi 78 cm og standardavvik 0.55 cm så lenge prosessen er under kontroll, og anta videre at resultater fra påfølgende målinger kan antas å være uavhengige.

Gitt at prosessen er under kontroll, hva er sannsynligheten for at gjennomsnittet av et utvalg på størrelse 10 er et tall mellom 77.7 og 78.3 cm?

Oppgi svaret som et desimaltall med tre desimaler, for eksempel 0.451 eller 0.123.

^(*) *Gunnar Løvås: Statistikk for universiteter og høyskoler, 4. utgave, kapittel 11.1.1*

Eksempel 2: Oppdage kvalitetsavvik (øving 6)

Hvordan ser 'verden' ut når H_0 er sann?

```
import numpy as np
mu = 78      # forventningsverdi
sigma = 0.55/np.sqrt(10) # standardavvik

xval = np.linspace(mu - 4*sigma, mu + 4*sigma, 100) # x-verdier
fx = stats.norm.pdf(xval, mu, sigma)                # f(x)
plt.plot(xval, fx)
plt.grid(); plt.ylabel("f(x-bar)"); plt.xlabel("x-bar")
plt.show()
```

