# Inferență statistică în ML

## Cap 3. Distribuții importante. Asimptotics. Intervale de confidență

March 27, 2019

# Distribuția Bernoulli

- **distribuția Bernoulli** apare ca urmare al unui rezultat așteptat de tip binar (aruncarea monedei, rezultat = Head sau Tail)
- variabilele aleatoare Bernoulli iau doar valori 1 sau 0 cu probabilitățile p și (1-p)
- Probability Mass Function:

$$P(X = x) = p^x(1 - p)^{1-x}$$

- media distribuției $\mu = p$
- dispersia $\sigma^2 = p(1 - p)$
- rezultat = 1 este denumit 'succes' iar rezultat = 0 - 'failure'

# Distribuția binomială

- o variabilă aleatoare binomială e obținută ca însumând mai multe variabile i.i.d. de tip Bernoulli
- ex. o variabilă binomială e numărul total de Heads care se obține aruncând de $n$ ori cu o monedă trucată
- fie $X_1, X_2 \dots X_n$ variabile iid Bernoulli(p)
- atunci $X = \sum_{i=1}^{n} X_i$ este o variabilă aleatoare binomială
- PMF:

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

- unde $\binom{n}{x} = \frac{n!}{x!(n-x)!}$ și $\binom{n}{0} = \binom{n}{n} = 1$
- selectarea a $x$ obiecte din $n$, fără replacement, fără a ține seama de ordinea obiectelor

# Exemplu de distribuție binomială

- un tată are 8 copii, din care 7 fete
- dacă fiecare gen are 50% probabilitate la fiecare naștere, care e probabilitatea de a avea 7 sau mai multe fete din 8 nașteri?

$$\binom{8}{7}.5^7(1-.5)^1 + \binom{8}{8}.5^8(1-.5)^0 \approx 0.04$$

```
1 print(binom(8, 7)*.5**8 + binom(8, 8)*.5**8)
2 print(stats.binom.pmf(8, n=8, p=.5) + stats.binom.pmf(7, n=8, p=.5))
3 print(1-stats.binom.cdf(6, n=8, p=.5))
```

```
0.03515625
0.035156250000000014
0.03515625
```

# Distribuția normală

- o variabilă aleatoare urmează **distribuție normală**, sau **Gaussiană** de medie $\mu$ și dispersie $\sigma^2$ dacă densitatea de probabilitate este:

$$F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- media $E[X] = \mu$
- dispersia $Var(X) = \sigma^2$
- $X \sim N(\mu, \sigma^2)$
- pentru $\mu = 0$ și $\sigma = 1$, distribuția se numește **standard normal distribution**

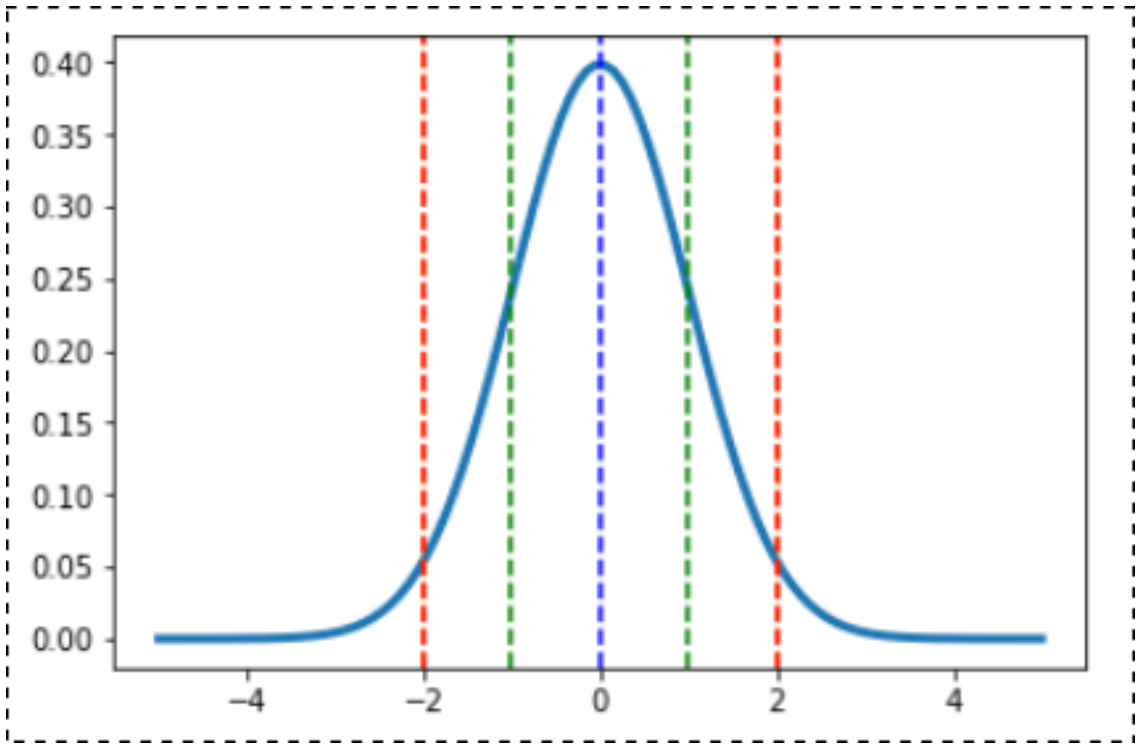
# Distribuții normale

- orice distribuție normală poate fi scalată cu media și dispersia sa (Z-scoring, normalizare)
- ca atare orice distribuție se poate reduce la distribuția normală standard

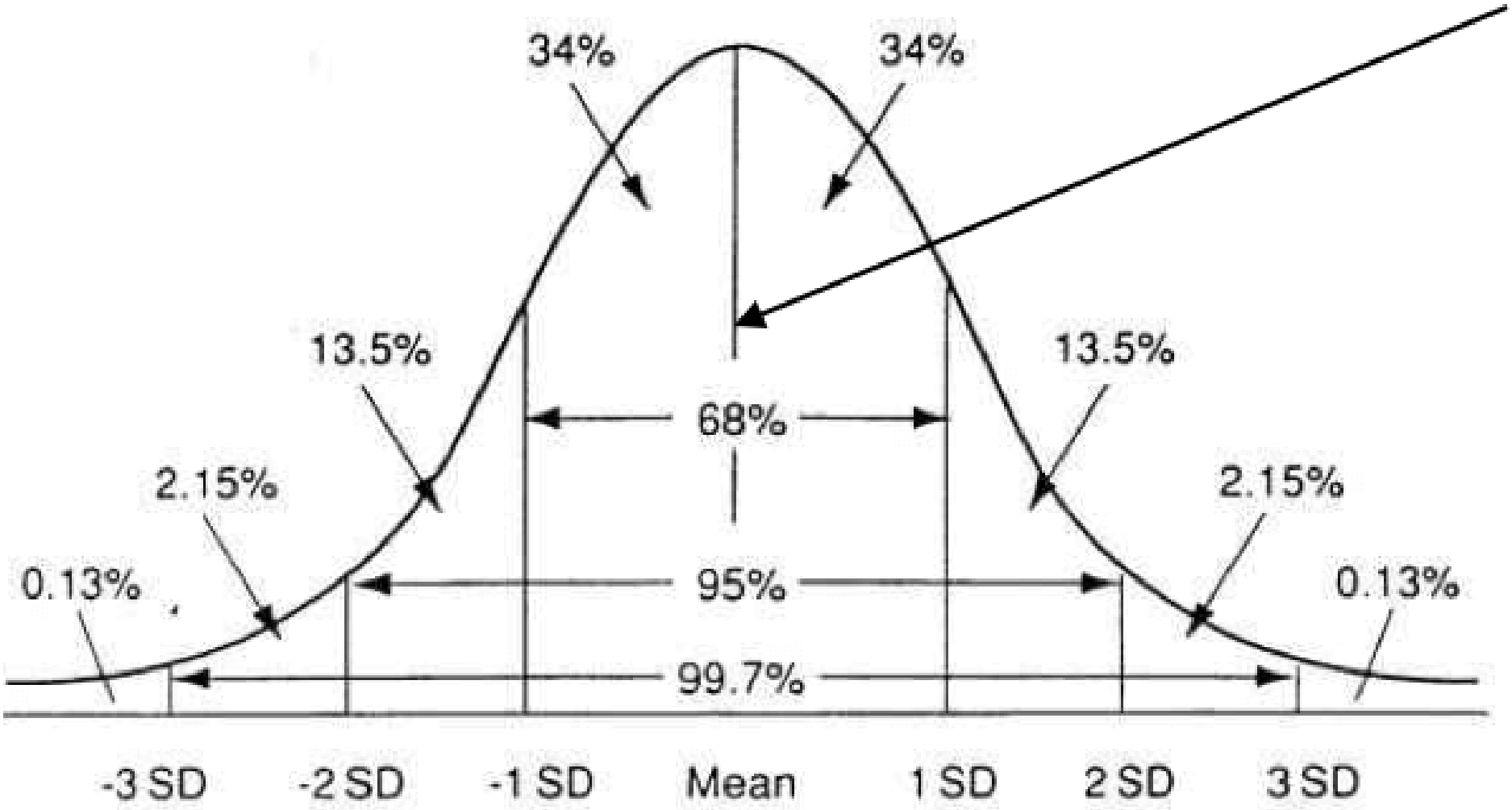
$$X \sim N(\mu, \sigma^2)$$

$$X_n \sim N(0, 1)$$

$$x_n = \frac{x - \mu}{\sigma}$$



# Percentile în distribuția gaussiană




from `http://tenmien.store/normal_graph_improvement_in.php`

# Conversii la distribuția normală standard

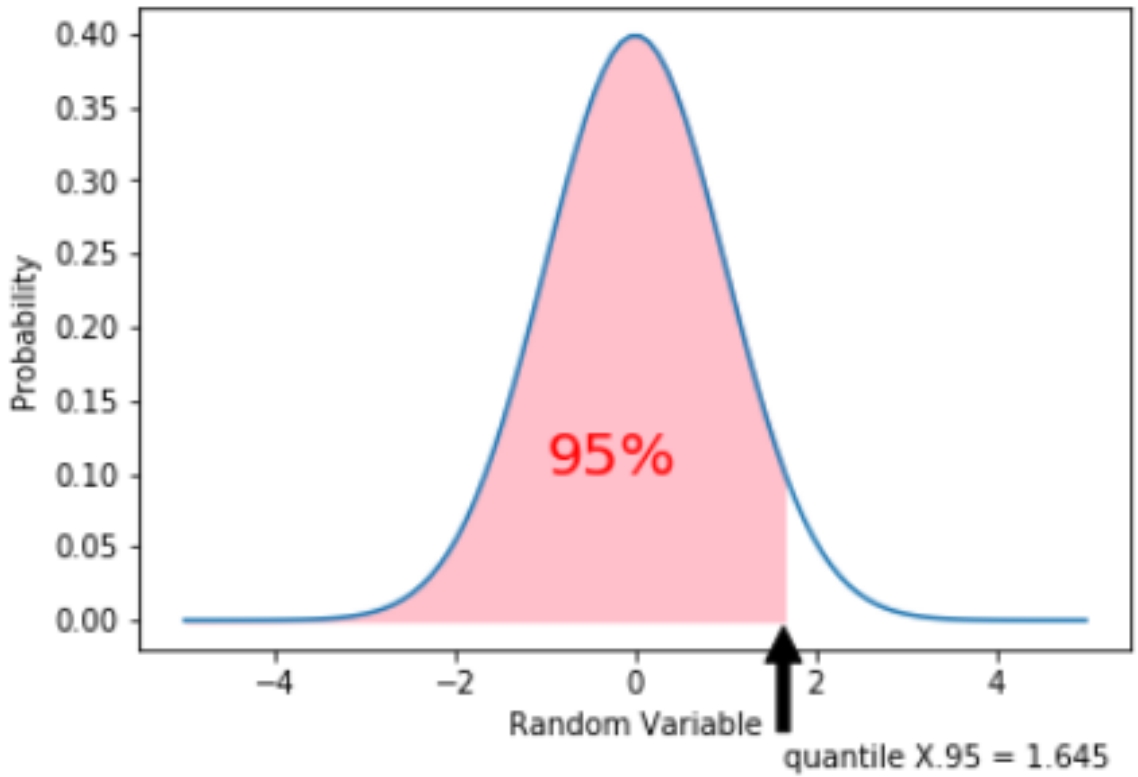
- dacă $X \sim N(\mu, \sigma^2)$, atunci
  $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$

- dacă Z este o distribuție normală standard,
  $X = \mu + \sigma Z \sim N(\mu, \sigma^2)$

- aproximativ 68%, 95% și 99% din densitatea normală se află la 1, 2 sau 3 deviații standard față de medie
- -1.28, -1.645, -1.96 și -2.33 sunt percentilele 10, 5, 2.5 și 1
- prin simetrie, 1.28, 1.645, 1.96 și 2.33 sunt percentilele 90, 95, 97.5 și 99

# Exemplu: quantile

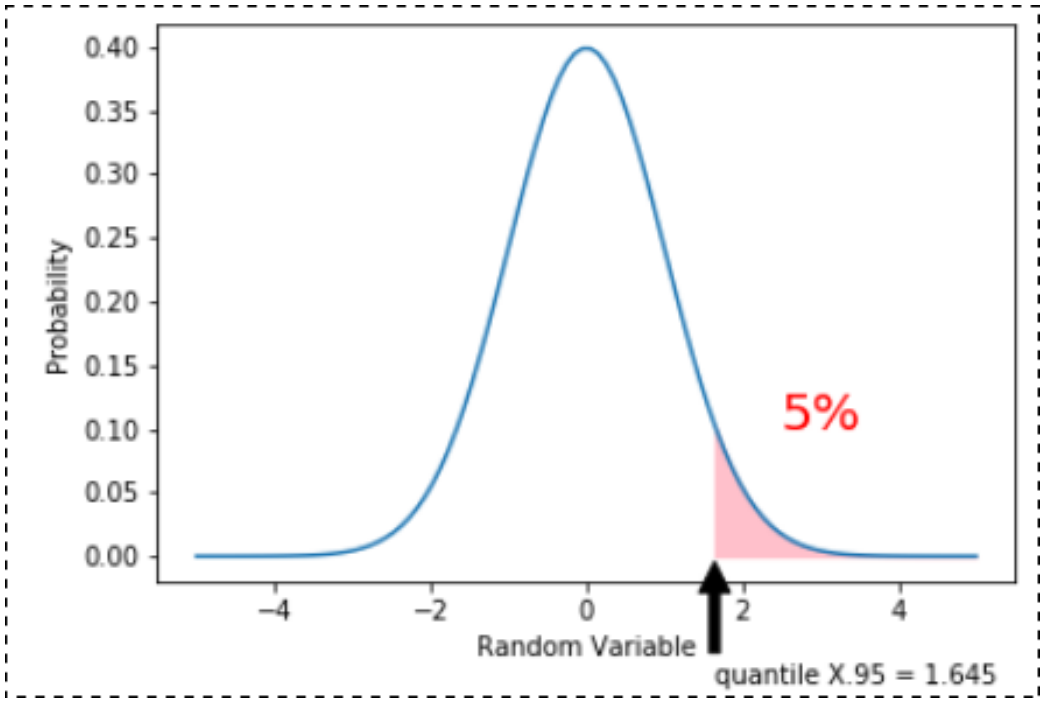
- care este percentila 95 pentru o distribuție $N(\mu, \sigma^2)$?




```
>> stats.norm.ppf(.95, loc=mu, scale=sigma)
1.6448536269514722
```

# Exemplu: quantile (2)

- care este probabilitatea ca o variabilă aleatoare $N(\mu, \sigma^2)$ să fie mai mare ca x?




```
>> 1-stats.norm.cdf(1.645, loc=mu, scale=sigma)
0.049984905539121 38
```

- se numește 'upper tail'

# Exemplu: quantile (3)

- Numărul zilnic de click-uri pentru o companie este distribuit normal cu medie 1020 și dispersie (deviație standard) 50. Care este probabilitatea ca să primească mai mult de 1160 click-uri pe zi?
- aceasta este 'upper tail'
- vezi slide 'Percentile în distribuția gaussiană'

```
>> print('cate deviatii standard fata de medie:',
          (1160-1020)/50)
>> print(1 - stats.norm.cdf(1160, loc=1020, scale=50))
>> print(1 - stats.norm.cdf(2.8))

cate deviatii standard fata de medie: 2.8
0.0025551303304279793
0.0025551303304279793
```

# Exemplu: quantile (3)

- Numărul zilnic de click-uri pentru o companie este distribuit normal cu medie 1020 și dispersie (deviație standard) 50. Care este numărul minim de click-uri zilnice corespunzător unei probabilități de cel puțin 70%?
- aceasta este 'lower tail'
- vezi slide 'Percentile în distribuția gaussiană'

```
>> print('1 standard deviation: ', 1020+50)
>> print(stats.norm.ppf(.75, loc=1020, scale=50))

1 standard deviation:  1070
1053.724487509804
```

# Distribuția Poisson

- folosită la modelarea numărării evenimentelor care apar

$$P(X = x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

- $x = 0, 1, 2 \ldots$
- media $\mu = \lambda$
- dispersia $\sigma = \lambda$
- egalitatea mediei cu dispersia este un test de verificare că datele urmează o distribuție Poisson

# Utilizări ale distribuției Poisson

- modelarea numărării apariției unor evenimente, în mod special dacă numărul aparițiilor este nelimitat
- modelarea timpului de supraviețuire, de exemplu pentru un medicament, timpul de apariție al unor simptome anume pe parcursul studiului
- modelarea contingency table; contingency table conține numărul de candidați care au, cumulat, mai multe caracteristici (date de axele tabelei)
- aproximarea distribuțiilor binomiale pentru $n$ foarte mare și $p$ foarte mic (multe trials dar probabilitate foarte mică să se întâmple evenimentul); populație numeroasă dar incidența bolii mică

# Modelarea ratelor de apariție cu distribuție Poisson

$$X \sim Poisson(\lambda t)$$

pentru care:

- $\lambda$ este numărul mediu de evenimente pe unitatea de timp
- $\lambda = E[X/t]$ este numărul așteptat de evenimente pe unitatea de timp
- $t$ este timpul total pentru care se face studiul, exprimat în secunde, ore, sau zile (de exemplu)

# Poisson: exemplu

- numărul de oameni care apar la stația de autobuz este distribuit Poisson cu o medie de 2.5 oameni pe oră
- dacă monitorizăm stația de autobuz pentru 4 ore, care e probabilitatea ca 3 sau mai puțini oameni să apară în stație pe durata celor 4 ore?

```
>> print(stats.poisson.cdf(3, mu=2.5 * 4))

0.010336050675925726
```

# Aproximarea distribuției Poisson cu o distribuție binomială

- dacă $n$ este mare și $p$ mic, distribuția Poisson e o aproximare destul de precisă pentru distribuția binomială

  $X \sim Binomial(n, p)$

  $\lambda = np$

  $n$ se mărește

  $p$ se micșorează

# Exemplu

- monedă cu probabilitatea de succes de 0.01, un număr de 500 de aruncări
- care e probabilitate de 2 sau mai puține succese?

```
>> print(stats.binom.cdf(2, n=500, p=0.01))
>> print(stats.poisson.cdf(2, mu=500 * 0.01))

0.12338577435354905
0.12465201948308108
```

# Asymptotics

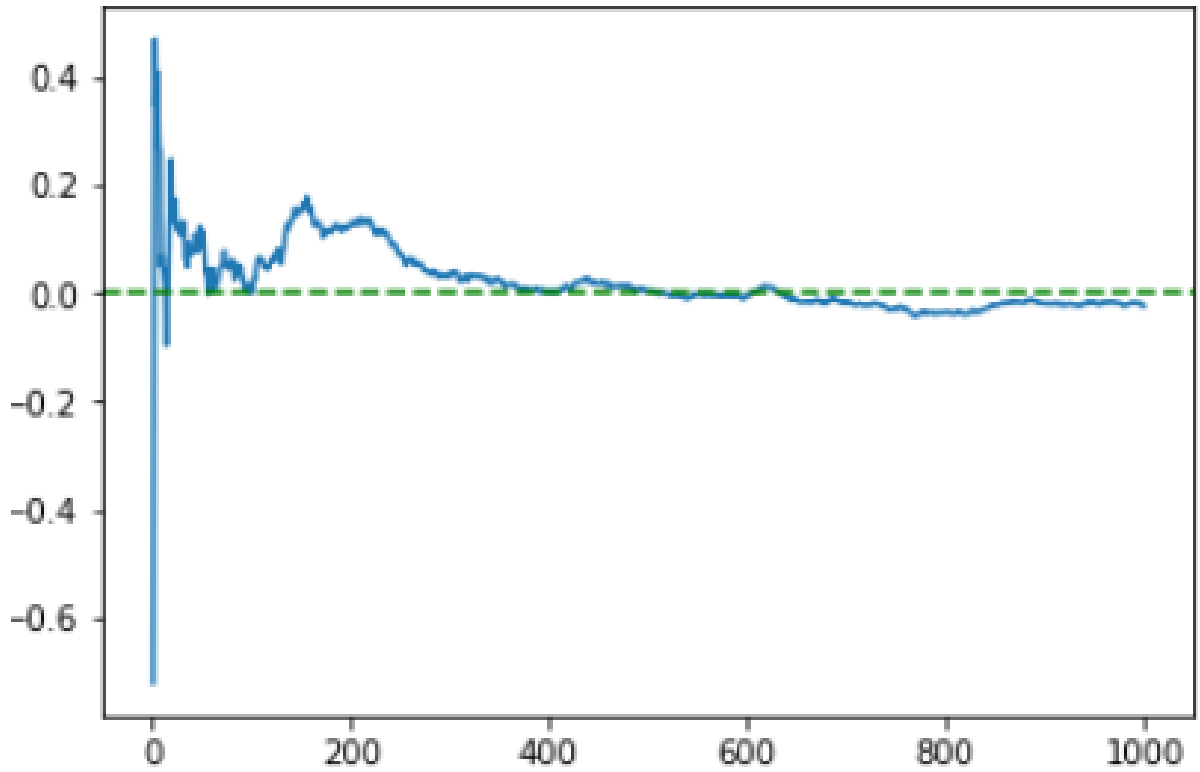
- comportamentul asimptotic descrie comportamentul statistic pe măsură ce sample size (sau altă cantitate relevantă) tinde la infinit (sau la altă limită relevantă)
- comportamentul este util de studiat pentru aproximări sau pentru a face inferențe
- investigarea proprietăților statistice fără a recurge la agregări masive
- un exemplu banal este convergența mediei pentru multe aruncări ale unei monede ideale (Law of Large Numbers)

# Variabile aleatoare la limită

- caracterizarea distribuțiilor sample means pentru colecții de observații iid
- LLN: Law of Large Numbers
  - media tinde la limită la ceea ce dorește să estimeze, adică media populației
- de exeplu, $\bar{X}_n$ poate fi rezultatul mediu ce indică proporția de Heads
- pe măsură ce realizăm mai multe aruncări, media converge la probabilitatea de a da Heads

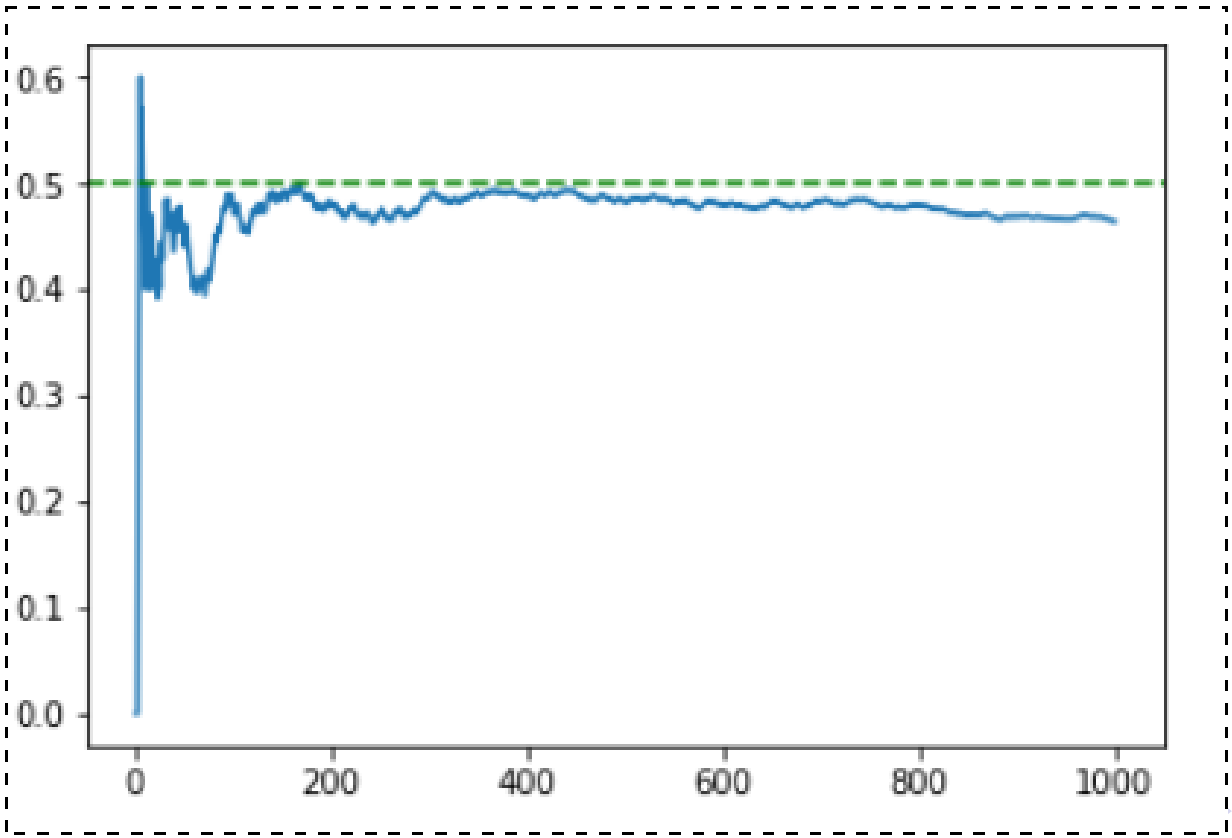
# LLN pentru distribuția normală

```
n = 1000
x = np.random.randn(n)
means = np.cumsum(x) / np.array(range(1, n+1))
plt.plot(np.array(range(1, n+1)), means)
plt.axhline(0, c='g', linestyle='--')
```



# LLN pentru distribuția binomială

```
n = 1000
x = np.random.randint(low=0, high=2, size=(n))
means = np.cumsum(x) / np.array(range(1, n+1))
plt.plot(np.array(range(1, n+1)), means)
plt.axhline(0.5, c='g', linestyle='--')
```



# Estimatori și consistență

- un estimator este **consistent** dacă el converge la ceea ce încearcă să estimeze
- LLN afirmă că sample mean pentru sample-uri iid este consistentă cu population mean
- sample variance precum și sample standard deviation pentru variabile aleatoare iid sunt de asemenea consistente

# Central Limit Theorem

- rezultat important în statistică
- CLT afirmă că distribuția mediilor variabilelor aleatoare iid (normalizate corespunzător) devine cea a unei distribuții normale standard pe măsură ce mărimea sample-ului crește
- se aplică în foarte multe contexte, datorită enunțului generic

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} = \frac{\textit{Estimate} - \text{Mean of estimate}}{\text{Standard error of estimate}}$$

- are o distribuție ca aceea a unei distribuții normale standard pentru $n$ mare
- înlocuirea deviației standard a populației ($\sigma$, necunoscută) cu deviația standard a sample-ului (cunoscută), nu schimbă rezultatul teoremei
- important: CLT zice că $\bar{X}_n$ e aproximativ $N(\mu, \sigma^2/n)$

# CLT: exemple

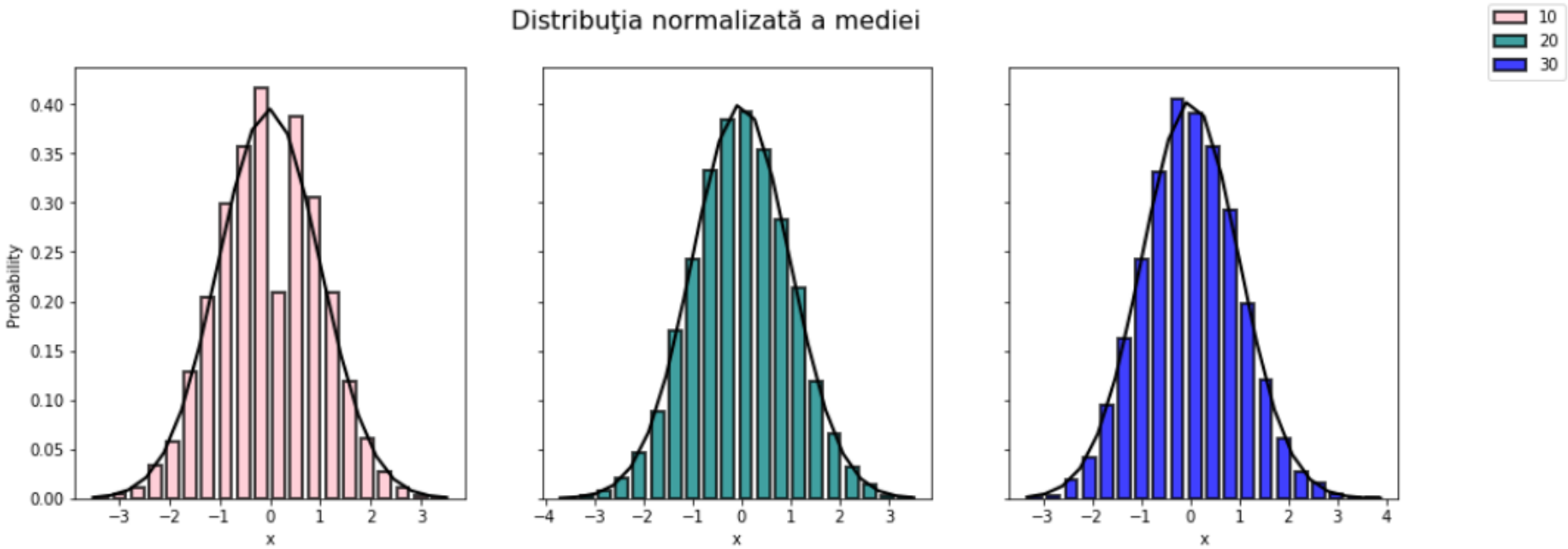
- simularea unei variabile standard normale prin distribuția binomială (zar), experiment repetat de n ori
- fie $X_i$ rezultatul pentru aruncarea cu zarul $i$

  $\mu = E[X_i] = 3.5$

  $Var(X_i) = 2.92$

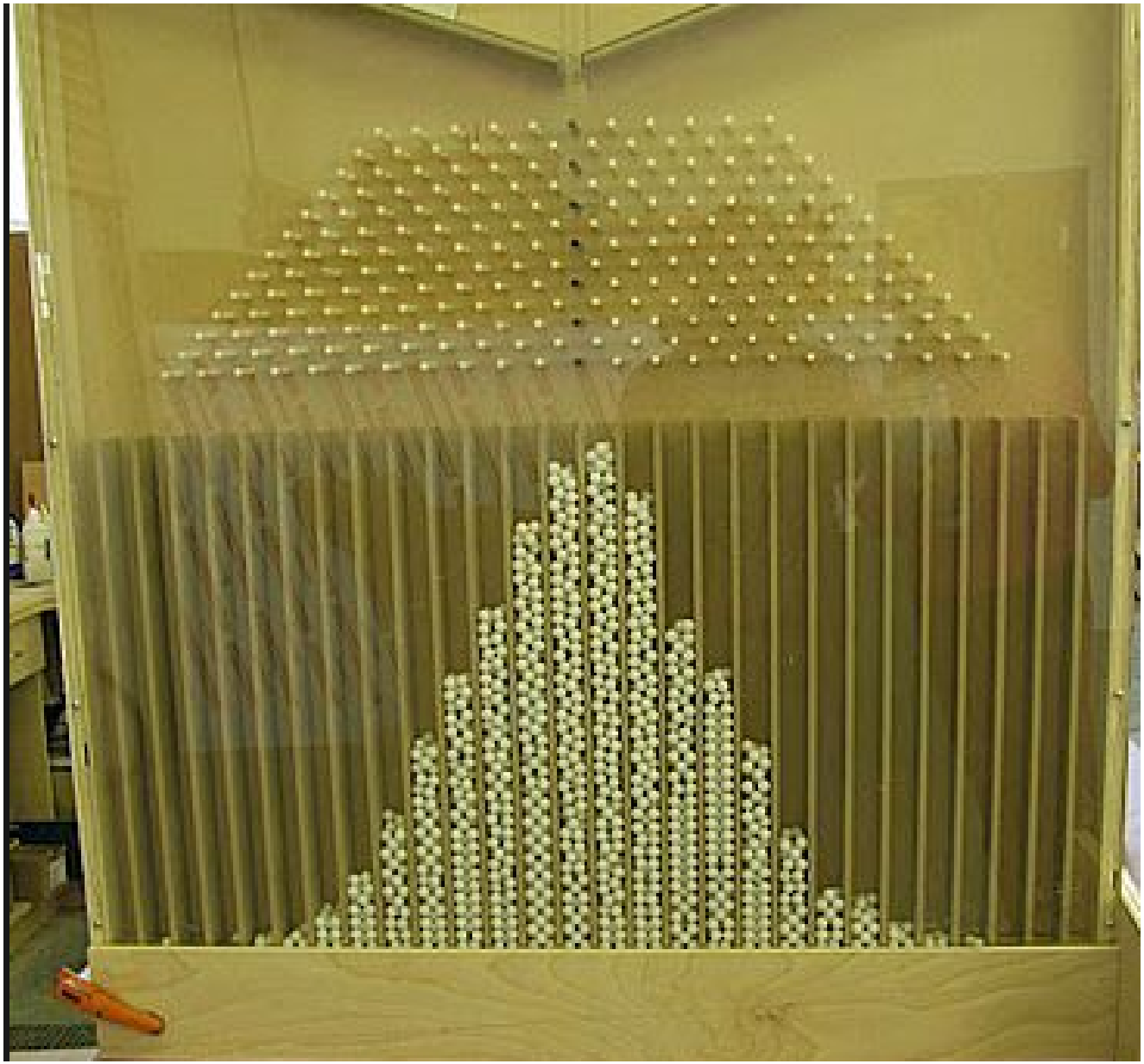
  standard error $\sqrt{2.92/n} = 1.71/\sqrt{n}$
- dăm de $n$ ori, aflăm media, scădem 3.5 și împărțim cu $1.71/\sqrt{n}$

# CLT: exemple (2)




- se aruncă cu zarul de 10 ori și se calculează media; aceasta se normalizează prin scăderea valorii așteptate și împărțirea la standard error
- se repetă procesul de 10000 de ori
- ne așteptăm ca procesul să fie centrat în 0 din cauza normalizării
- aproximarea e destul de bună

# Quincunx machine (Galton board)



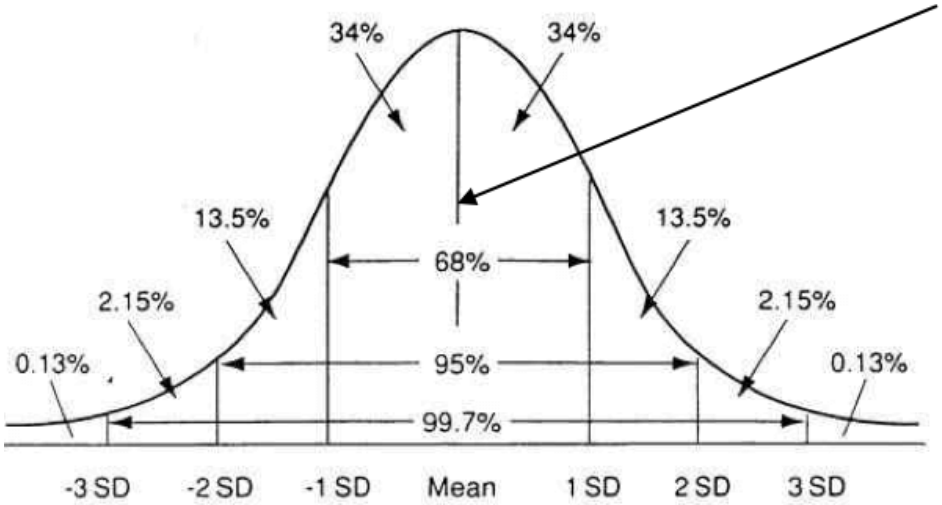
# Quincunx machine și CLT

- imaginea de la `https://nauka.metodolog.pl/wp-content/uploads/2017/01/Deska-Galtona.png`
- bilele se duc spre stânga/dreapta cu probabilitatea 0.5
- CLT ne spune că media variabilei aleatoare binomiale (distribuția Heads) tinde la o distribuție normală
- dacă înmulțim cu $n$, rezultă că și suma variabilelor aleatoare binomiale tinde tot la distribuția normală

1 Distribuții importante

# Intervale de confidență

- CLT ne spune că distribuția sample mean $\bar{X}$ e o distribuție aproximativ normală de medie $\mu$ și deviație standard $\sigma/\sqrt{n}$




- probabilitatea ca $\bar{X}$ să fie mai mare ca $\mu + 2\sigma/\sqrt{n}$ sau mai mică decât $\mu - 2\sigma/\sqrt{n}$ este 5%

$$P(\bar{X} < \mu - 2\sigma/\sqrt{n}) \text{ sau } P(\bar{X} > \mu + 2\sigma/\sqrt{n}) \sim 5\%$$

# Intervale de confidență (2)

- mai interesantă este relația lui 95%:

$$P(\mu - 2\sigma/\sqrt{n} < \bar{X} < \mu + 2\sigma/\sqrt{n}) \sim 95\%$$

- putem exprima și pe $\mu$ în funcție de $\bar{X}$, inversând rolurile:

$$P(\mu < \bar{X} + 2\sigma/\sqrt{n}) \text{ și } P(\mu > \bar{X} - 2\sigma/\sqrt{n}) \sim 95\%$$

- ceea ce se scrie mai compact:

$$P(\bar{X} - 2\sigma/\sqrt{n} < \mu < \bar{X} + 2\sigma/\sqrt{n}) \sim 95\%$$

- intervalul $\bar{X} \pm 2\sigma/\sqrt{n}$ este denumit intervalul 95% pentru $\mu$ (probabilitatea ca intervalul să conțină media populației)
- **interpretare**: dacă luăm sample-uri de dimensiune $n$ din populație și construim intervalul de confidență de fiecare dată, 95% din intervalele pe care le dăm conțin media populației

# Intervale de confidență (3)

```
1 father_son = pd.read_csv('father_son.csv')
2 father_son.head()
```

| | Unnamed: 0 | fheight | sheight |
|---|---|---|---|
| **0** | 1 | 65.04851 | 59.77827 |
| **1** | 2 | 63.25094 | 63.21404 |
| **2** | 3 | 64.95532 | 63.34242 |
| **3** | 4 | 65.75250 | 62.79238 |
| **4** | 5 | 61.13723 | 64.28113 |

```
1 # rezultatul generat in feet,
2 # dataset-ul este in inches
3 x = father_son['sheight'].values
4 ( np.mean(x) + np.array([-1, 1]) * stats.norm.ppf(0.975)
5       * np.std(x) / np.sqrt(len(x)) )/12
```

```
array([5.70967698, 5.73766797])
```

# Interval de confidență în cazul monezii

- pentru aruncarea cu moneda, $X_i = 1$ este evenimentul succes (head) și $X_i = 0$ evenimentul fail (tail)
- dacă probabilitatea de succes este $p$, dispersia $\sigma = p(1-p)$
- intervalul de confidență:

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}}$$

- formularea de mai sus dă intervalul de confidență Wald pentru media $p$ a distribuției Bernoulli (monedă)
- pentru o monedă fair, $p(1-p)$ are valoarea maximă 0.25, iar intervalul de confidență 95% se scrie astfel:

$$\hat{p} \pm 2\sqrt{\frac{1/4}{n}} \qquad \text{sau} \qquad \hat{p} \pm \frac{1}{\sqrt{n}}$$

- relația ne dă rapid estimarea intervalului de confidență pentru $p$ (dacă moneda nu e fair, intervalul acesta e mai larg decât cel real)

# CI: exemplu

- șeful de campanie afirmă că într-un sample random de 100 de votanți potențiali, 56 intenționează să voteze cu candidatul X
- putem să afirmăm că sunt șanse de 95% ca X să câștige alegerile?
- distribuție Bernoulli, putem calcula standard error a mediei ca $1/2\sqrt{n}$
- intervalul de confidență Wald:

$$\hat{p} \pm \frac{1}{\sqrt{n}} = 0.56 \pm \frac{1}{\sqrt{100}} = (0.46, 0.66)$$

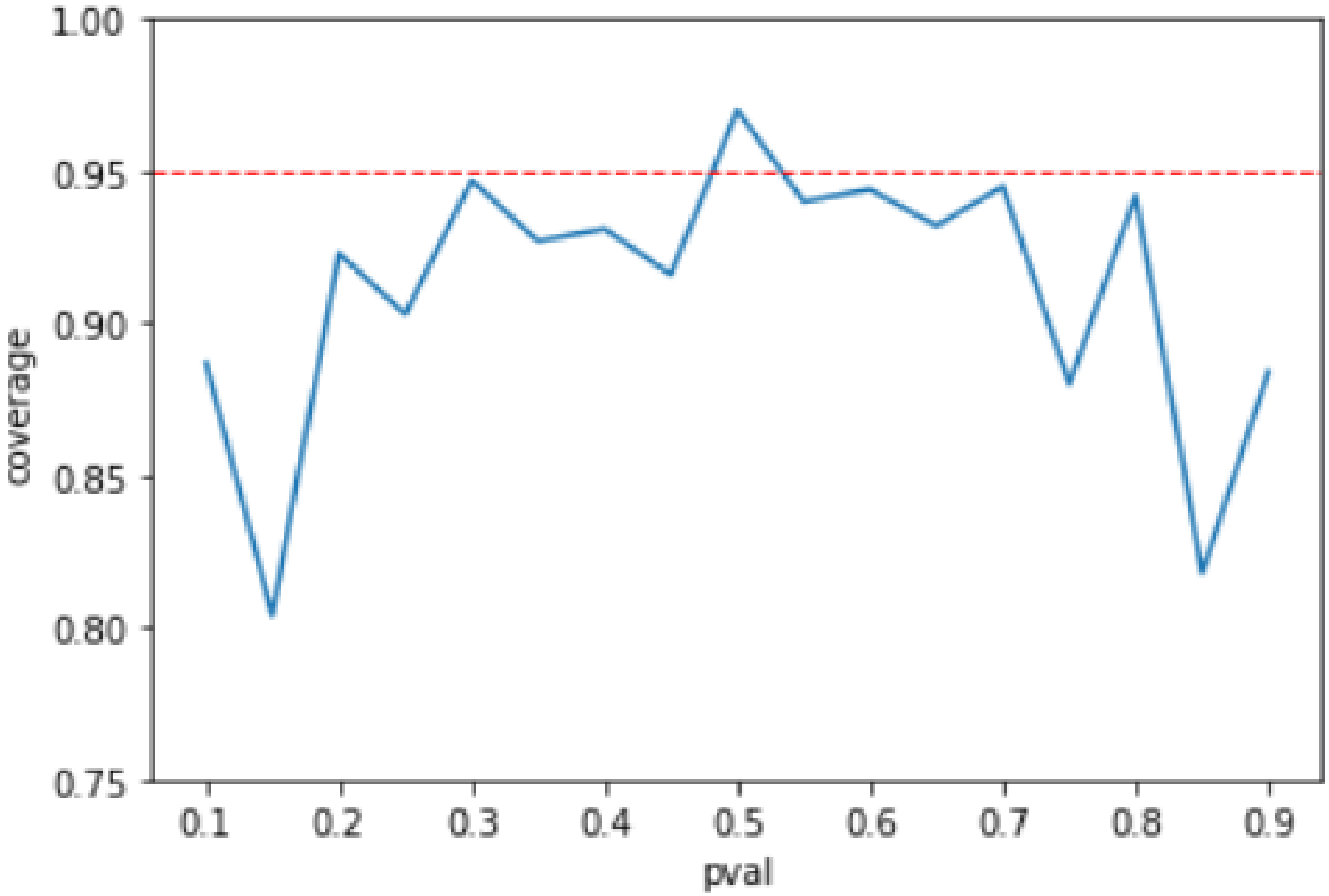
- este posibil ca X să piardă alegerile, nu suntem 95% siguri că media 0.5 nu se află în acest interval

```
>> 0.56 + np.array([-1, 1]) * stats.norm.ppf(0.975)
      * np.sqrt(0.56*(1-0.56)/100)
array([0.46270995, 0.65729005])
```

# Simulare

```
# what happens with CI when coin is strongly biased
n = 20
pvals = np.linspace(start=0.1, stop=0.9, num=17)
nosim = 1000

def coverage(p):
    phats = np.random.binomial(n, p=p, size=nosim)/n
    sde = np.sqrt(phats * (1-phats)/n)
    ll = phats - norm.ppf(0.975) * sde
    ul = phats + norm.ppf(0.975) * sde
    return np.mean(np.logical_and(ll < p, ul > p))

plt.plot(pvals, [coverage(p) for p in pvals])
plt.axhline(0.95, c='r', lw=1, linestyle='--')
plt.show()
```

# Simulare (2)



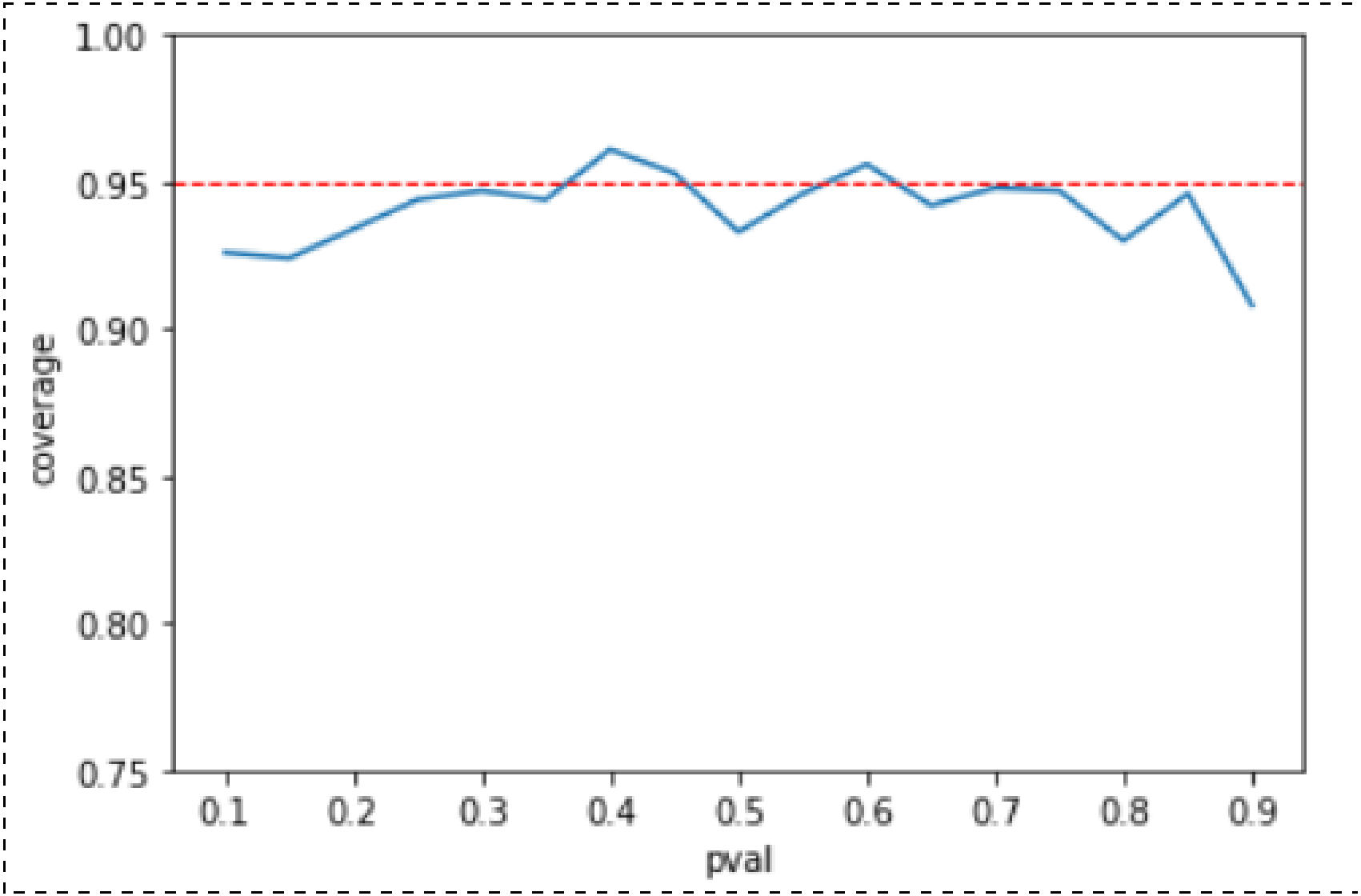
# Problema simulării

- $n$, mărimea sample-ului, nu este suficient de mare pentru ca CLT să se aplice pentru multe din valorile lui $p$

$$\hat{p} = \frac{X + 2}{n + 4}$$

- se adaugă 2 atât la nr. de succese cât și la cel de failures
- procedura se numește Agresti-Coull interval

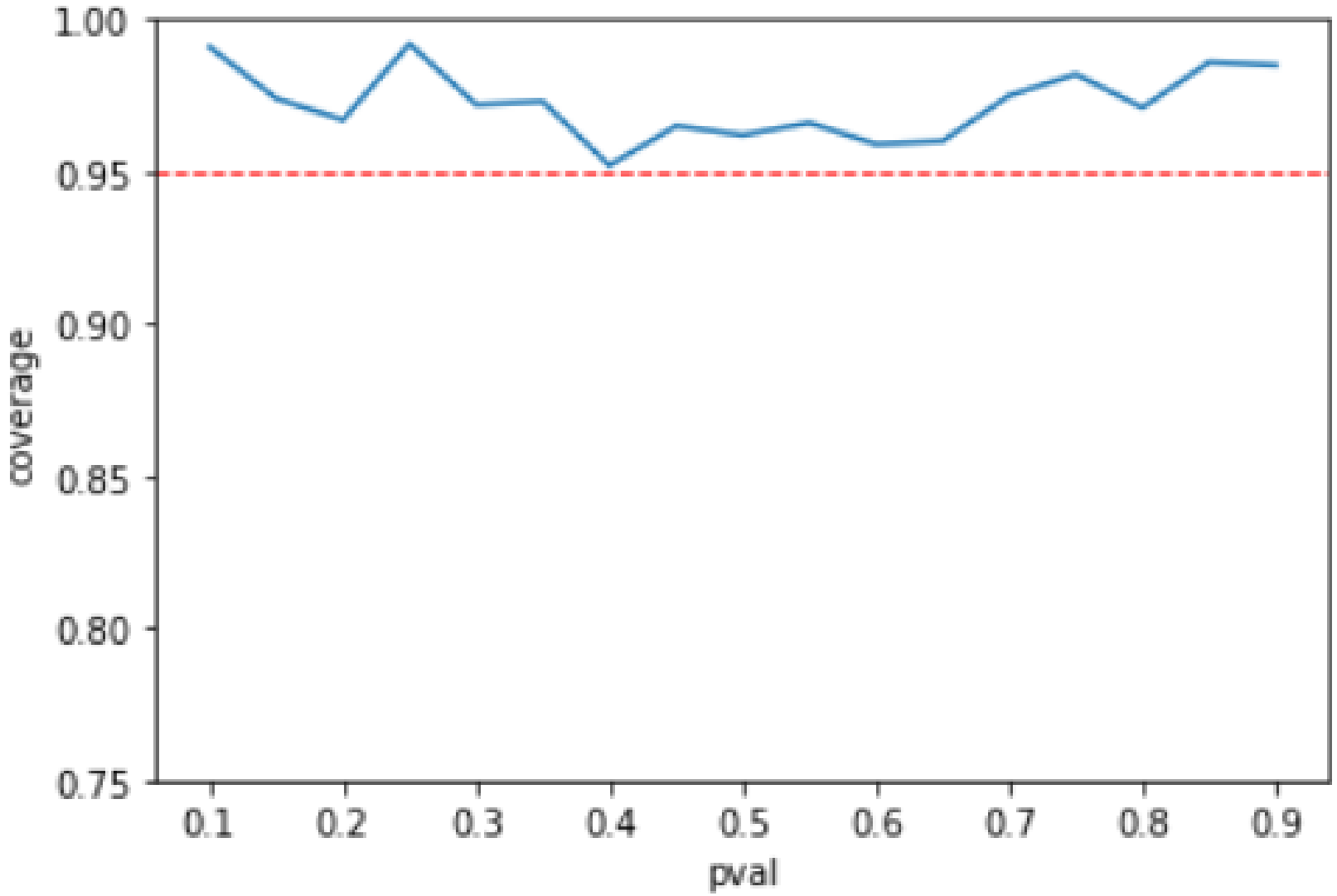
# Coverage pentru n = 100



# Corecția Agresti/Coull

```
# what happens with CI when coin is strongly biased
n = 20
pvals = np.linspace(start=0.1, stop=0.9, num=17)
nosim = 1000

def coverage(p):
    phats = (binomial(n, p=p, size=nosim)+2) / (n+4)
    sde = np.sqrt(phats * (1-phats)/n)
    ll = phats - norm.ppf(0.975) * sde
    ul = phats + norm.ppf(0.975) * sde
    return np.mean(np.logical_and(ll < p, ul > p))

plt.plot(pvals, [coverage(p) for p in pvals])
plt.axhline(0.95, c='r', lw=1, linestyle='--')
plt.show()
```

# Corecția Agresti/Coull (2)



# Confidence Interval pentru Poisson: exemplu

- o pompă nucleară a picat de 5 ori în 94.32 zile
- dați un interval de confidență de 95% pentru rata de defectare zilnică
- $X \sim Poisson(\lambda t)$
- estimarea ratei de defectare este $\hat{\lambda} = X/t$, numărul de defectări supra timpul total monitorizat
- dispersia $Var(\hat{\lambda}) = \lambda/t$ [1]
- $\hat{\lambda}/t$ este estimarea empirică a dispersiei
- intervalul de confidență pentru media populației:

$$\hat{\lambda} \pm z_{1-\alpha/2} \underbrace{\sqrt{\frac{\lambda/t}{t}}}_{\text{standard error}}$$

- $z_{1-\alpha/2}$ este quantila relevantă (97.5%) din distribuția normală standard

[1] similar cu $Var(\bar{X}) = \sigma^2/n$

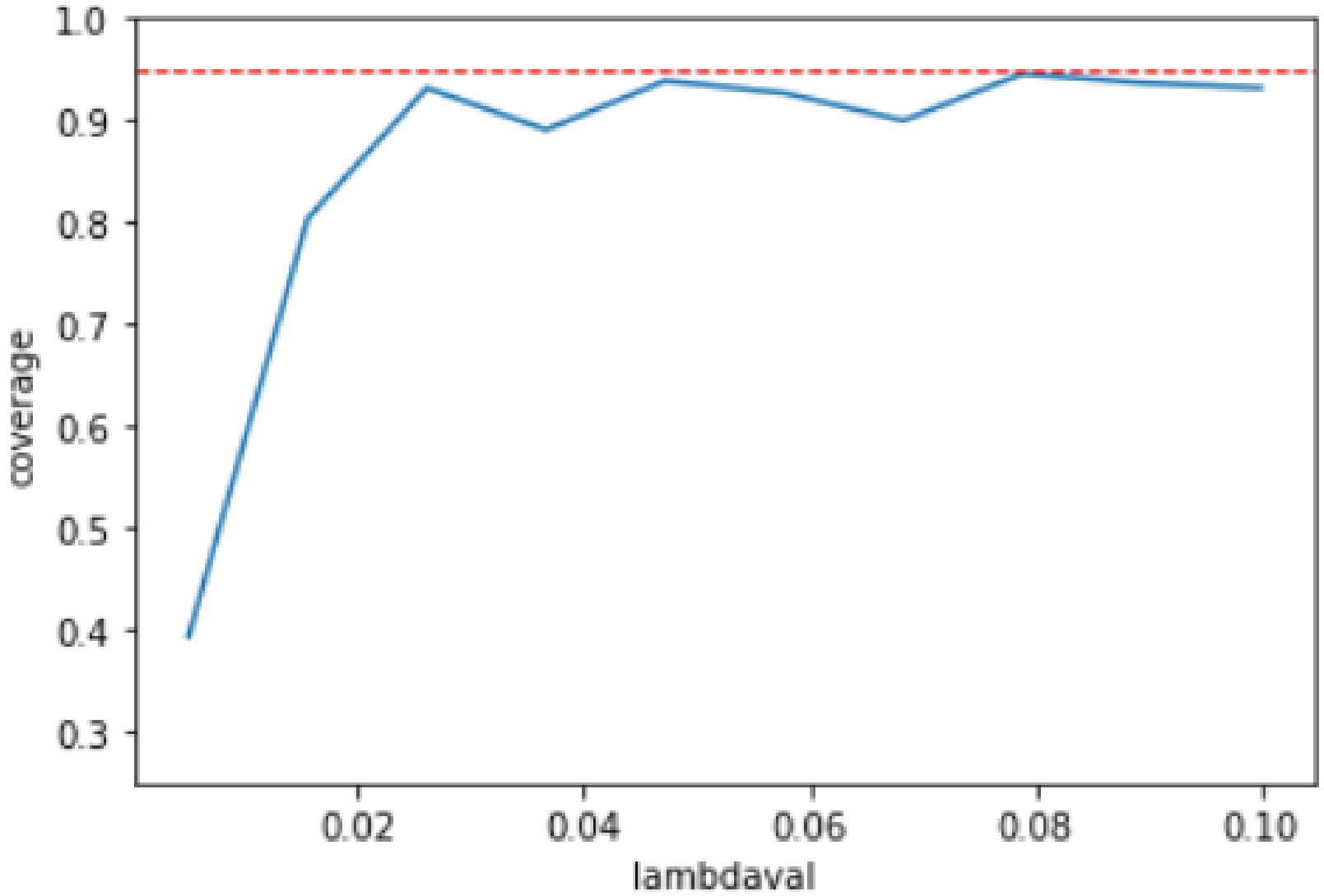
# Codul Python

```
>> x = 5          # numarul de defecte
>> t = 94.32      # durata monitorizata
>> lmbda = x/t # estimarea ratei (mediei)
>> np.round(lmbda + np.array([-1, 1])
                    * norm.ppf(0.975)
                    * np.sqrt(lmbda/t), 3
            )
array([0.007, 0.099])
```

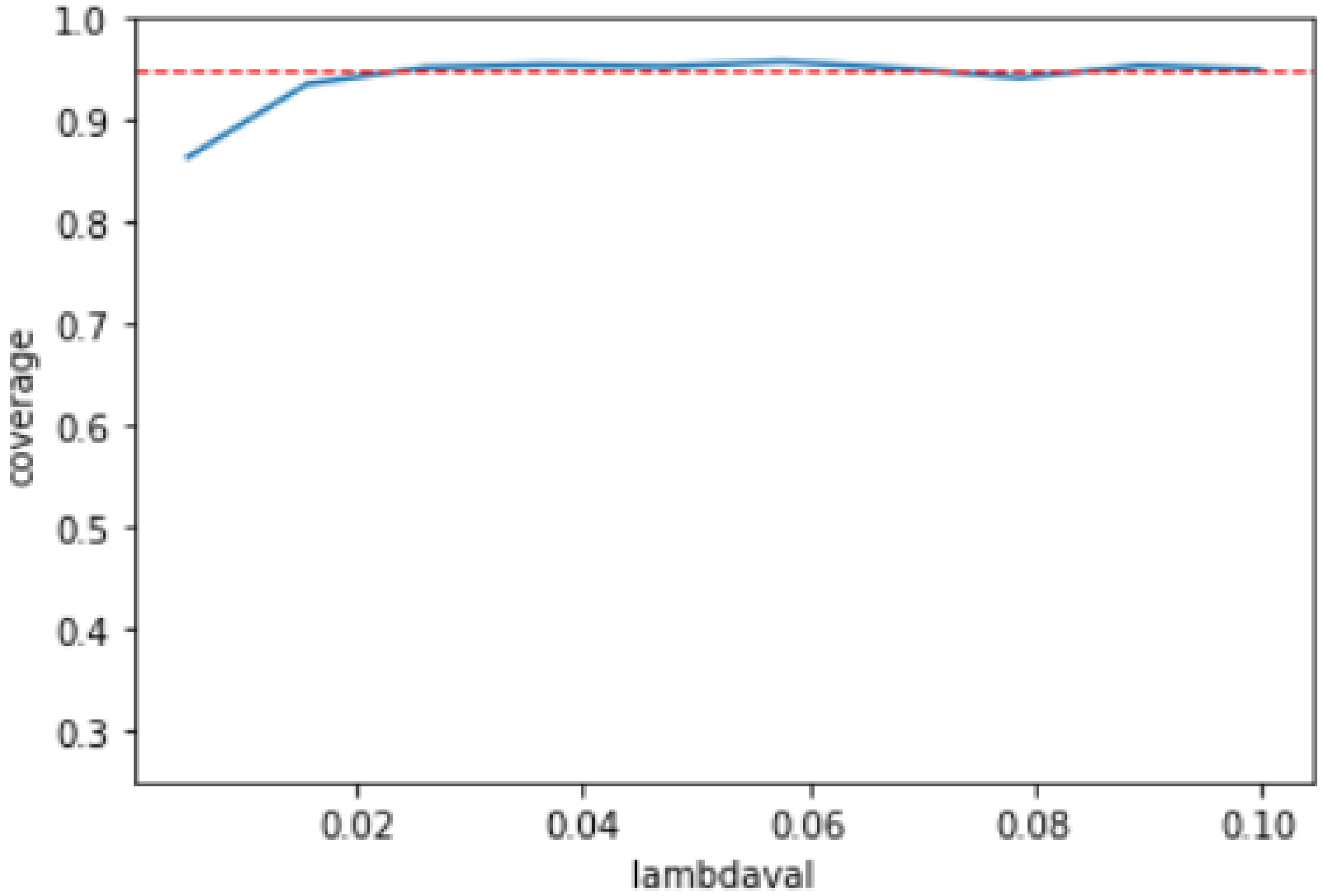
# Coverage pentru CI în cazul Poisson

```
t = 100
lmbdavals = np.linspace(start=0.005, stop=0.1, num=10)
nosim = 1000

def coverage(l):
    lhats = (np.random.poisson(lam=l*t, size=nosim))/t
    ll = lhats - norm.ppf(0.975) * np.sqrt(lhats/t)
    ul = lhats + norm.ppf(0.975) * np.sqrt (lhats/t)
    return np.mean(np.logical_and(ll < l, ul > l))

plt.plot(lmbdavals, [coverage(l) for l in lmbdavals])
plt.ylim(0.25, 1)
plt.axhline(0.95, c='r', lw=1, linestyle='--')
plt.xlabel('lambdaval')
plt.ylabel('coverage')
plt.show()
```

# Coverage pentru Poisson



# Timpul de monitorizare crește la t=1000



# Sumar

- LLN: mediile sample-urilor iid converg către media populației pe care o estimează, când dimensiunea sample-ului crește
- echivalent, ratele Poisson tind către rata pe care o estimează, când timpul de monitorizare tinde la infinit
- CLT: mediile sunt distribuite aproximativ normal, cu distribuția:
  - centrată la media populației
  - deviația standard egală cu standard error a mediei
  - CLT nu dă garanții, anume dacă $n$ e suficient de mare

# Sumar (2)

- folosind media estimată, prin adunare și scădere a quantilei normale relevante înmulțite cu standard error a mediei dă intervalul de confidență pentru medie $\longrightarrow$ intervale Wald
  - $\pm$ 2 * standard error se folosește pentru intervalele 95%
- intervalele de confidență devin mai mari când coverage crește; un interval pentru 99% va fi mai larg decât unul de 95%, care la rândul lui va fi mult mai mare ca unul de 50%
- pentru valori mici ale mediei în cazul distribuțiilor binomială și Poisson, intervalele generate nu garantează în 95% din cazuri că vor conține media căutată:
  - corecția Agresti/Coull sau mărirea dimensiunii sample-ului
  - folosirea unui timp de monitorizare cu un ordin de mărime mai mare