EE 046202 - Technion - Unsupervised Learning & Data Analysis

Tal Daniel (https://taldatech.github.io)

Tutorial 02 - Classical Methods in Statistical Inference - Confidence Intervals

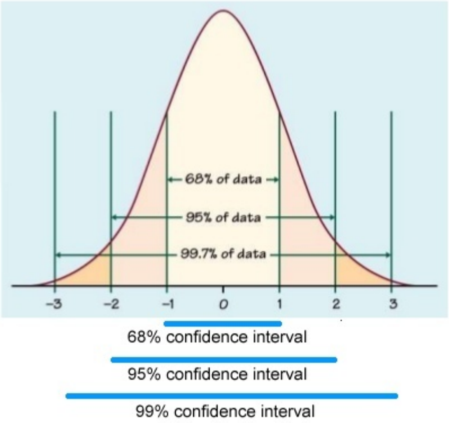
Agenda

- Interval Estimation (Confidence Intervals))
  - Formulation
  - Finding Interval Estimators
- Bootstrap Approach
  - Boostrap for the Median Estimator ()
- Confidence Interval with Chebyshev Inequality
- Confidence Interval with DKW Inequality
- Recommended Videos
- Credits

In [1]:
```
# imports for the tutorial
import numpy as np
import pandas as pd
from helpers.cdf_bounds import plot_cdf_confidence
from scipy.stats import norm
import time
import matplotlib.pyplot as plt
%matplotlib notebook
```

Confidence Intervals (Interval Estimation)

- What is the problem with the **point estimation** for θ, $\hat{\theta}$?
  - $\hat{\theta}$ alone does not give much information about θ!
  - Without any additional information, we cannot tell how close is $\hat{\theta}$ to the real θ.

- **Interval Estimation** - an interval that is *likely* to include the true value of θ.
  - Instead of saying $\hat{\theta} = 8.14$, we report: $[\hat{\theta}_l, \hat{\theta}_h] = [6.65, 9.87]$, which we hope includes the real value of θ.
  - We actually produce 2 estimates for θ: *high* estimate $\hat{\theta}_h$ and *low* estimate $\hat{\theta}_l$.
  - We define two important properties of a confidence interval:
    - **Length** - the precision which we can estimate θ. Defined as $\hat{\theta}_h - \hat{\theta}_l$.
    - **Confidence Level** - how confident are we in the interval. It is the *probability* that the real θ lies in the interval. We wish to have high confidence (usually 90-95%).
    - 

      [Image Source (https://www.quora.com/Why-is-a-99-confidence-level-wider-when-constructing-confidence-intervals)](https://www.quora.com/Why-is-a-99-confidence-level-wider-when-constructing-confidence-intervals)

Formulation

Let $X_1, X_2, .., X_n$ be a random sample from a distribution with a parameter θ that is to be estimated.

The goal is to find *two* (point) estimators for θ:

1. The low estimator:
$$\hat{\theta}_l = \hat{\theta}_l(X_1, X_2, \ldots, X_n)$$
2. The high estimator:
$$\hat{\theta}_h = \hat{\theta}_h(X_1, X_2, \ldots, X_n)$$

- The interval estimator is: $[\hat{\theta}_l, \hat{\theta}_h]$.
- **Confidence Level** - defined to be $1 - \alpha$.
- The estimators $\hat{\theta}_l, \hat{\theta}_h$ are chosen such that the **probability** that the interval $[\hat{\theta}_l, \hat{\theta}_h]$ includes θ is larger than the confidence level $1 - \alpha$. Thus, we wish α to be **small**.
  - Common values for α are: $0.1, 0.05, 0.01$ which correspond to confidence levels 90%, 95% and 99% respectively.
- Formally: find $\hat{\theta}_l, \hat{\theta}_h$ such that
$$P(\hat{\theta}_l \leq \theta \leq \hat{\theta}_h) \geq 1 - \alpha$$
  or
$$P(\theta \in [\hat{\theta}_l, \hat{\theta}_h]) \geq 1 - \alpha$$

Finding Interval Estimators

- Recall that for a continuous random variable X with **CDF** (Cumulative Distribution Function): $F_X(x) = P(X \leq x)$.
- We wish to find probability between bounds such that:

$$P(x_l \leq X \leq x_h) = 1 - \alpha$$
$$\rightarrow 1 - P(x_l \leq X \leq x_h) = \alpha$$
$$P(X \leq x_l \cup X \geq x_h) = \alpha = P(X \leq x_l) + P(X \geq x_h)$$

We can choose (notice the different inequality signs):

$$P(X \leq x_l) = \frac{\alpha}{2}, \text{and } P(X \geq x_h) = \frac{\alpha}{2}$$

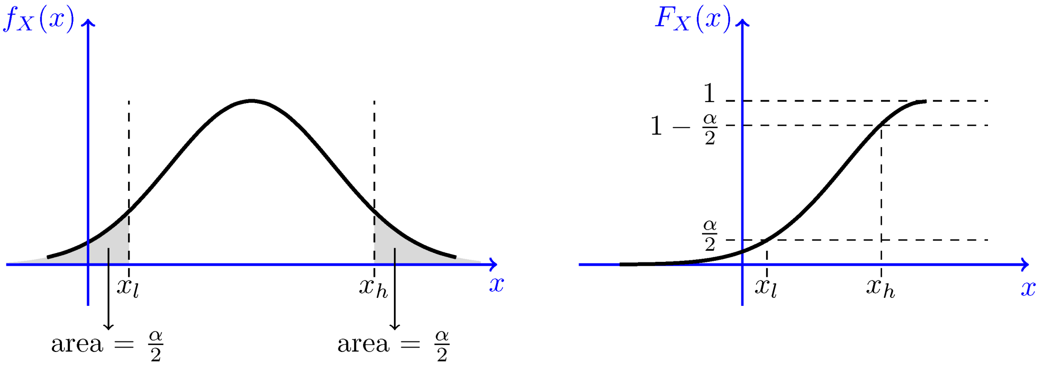
Or, equivalently

$$F_X(x_l) = \frac{\alpha}{2}, \text{and } F_X(x_h) = 1 - \frac{\alpha}{2}$$

$$\rightarrow x_l = F_X^{-1}(\frac{\alpha}{2}), x_h = F_X^{-1}(1 - \frac{\alpha}{2})$$

We call the interval $[x_l, x_h]$ a $(1-\alpha)$ interval for X

- Note: why do we divide α by 2 and not by 4 (and then pick $\frac{\alpha}{4}, \frac{3\alpha}{4}$ instead of $\frac{\alpha}{2}$)?
  - Since we are calculating *both* upper and lower bounds. Had we been interested in a single sided interval, then we would not have needed this division. Moreover, we would like our confidence level to be symmetrical in relation to both upper and lower bounds, there is no reason to give more weight to one over the other.



[Image Source (https://www.probabilitycourse.com/chapter8/8_3_2_finding_interval_estimators.php)](https://www.probabilitycourse.com/chapter8/8_3_2_finding_interval_estimators.php)

Exercise - Confidence Intervals for Normal Distribution (Using CLT)

1. (Probability Exercise) Let $Z \sim \mathcal{N}(0,1)$. Find x_l, x_h such that

$$P(x_l \leq Z \leq x_h) = 0.95$$

   - [Normal Table (https://www.math.arizona.edu/~jwatkins/normal-table.pdf)](https://www.math.arizona.edu/~jwatkins/normal-table.pdf)

2. Let $X_1, X_2, \ldots, X_n$ be a random sample from a normal distribution $\mathcal{N}(\theta, 1)$ ($\mathbb{E}[X] = \theta$). Find a 95% confidence interval for θ.

💡 **Solution**

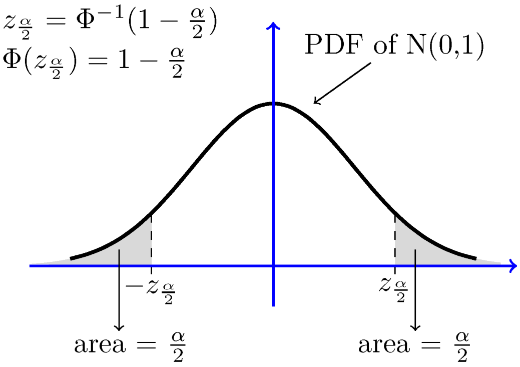
Section 1

$\alpha = 0.05$, and ϕ denotes the CDF of Z. Thus, we can choose:

$$x_l = \phi^{-1}(\frac{\alpha}{2}) = \phi^{-1}(0.025) = -1.96$$
$$x_h = \phi^{-1}(1 - \frac{\alpha}{2}) = \phi^{-1}(1 - 0.025) = 1.96$$

More generally, if we assume $Z \sim \mathcal{N}(0, 1)$, for any $p \in [0, 1]$, we define z_p as the real value for which $P(Z > z_p) = p$. Therfore, $\phi(z_p) = 1 - p \rightarrow z_p = \phi^{-1}(1 - p)$. In our case

$$P(-z_{\frac{\alpha}{2}} \leq Z \leq z_{\frac{\alpha}{2}}) = 1 - \alpha$$


$z_{\frac{\alpha}{2}} = \Phi^{-1}(1 - \frac{\alpha}{2})$
$\Phi(z_{\frac{\alpha}{2}}) = 1 - \frac{\alpha}{2}$
PDF of N(0,1)
$-z_{\frac{\alpha}{2}}$
$z_{\frac{\alpha}{2}}$
area $= \frac{\alpha}{2}$
area $= \frac{\alpha}{2}$

[Image Source (https://www.probabilitycourse.com/chapter8/8_3_2_finding_interval_estimators.php)](https://www.probabilitycourse.com/chapter8/8_3_2_finding_interval_estimators.php)

Section 2

We begin with finding the **point estimator** for θ, which is the *mean* of the distribution. Recall from the previous tutorial that: $\hat{\theta} = \overline{X} = \frac{X_1 + X_2 + ... + X_n}{n}$ and since $X_i \sim \mathcal{N}(\theta, 1)$ and the samples are i.i.d. then: $\overline{X} \sim \mathcal{N}(\theta, \frac{1}{n})$. We now perform **standartization** (normalization in our case):

$$\frac{\overline{X} - \theta}{\frac{1}{\sqrt{n}}} = \sqrt{n}(\overline{X} - \theta) \sim \mathcal{N}(0, 1)$$

- Why do we perform standartization? Because it is easier to work with $\mathcal{N}(0, 1)$ and we are familiar with its CDF and can easily find the values with the table.
- Standartization in the general case where $\sigma^2 \neq 1$:

$$\frac{\overline{X} - \theta}{\frac{\sigma}{\sqrt{n}}}$$

Thus:

$$P\big(-1.96 \leq \sqrt{n}(\overline{X} - \theta) \leq 1.96\big) = 0.95$$
$$\rightarrow P\big(\overline{X} - \frac{1.96}{\sqrt{n}} \leq \theta \leq \overline{X} + \frac{1.96}{\sqrt{n}}\big) = 0.95$$

Therefore, the confidence interval is:

$$[\hat{\theta}_l, \hat{\theta}_h] = [\overline{X} - \frac{1.96}{\sqrt{n}}, \overline{X} + \frac{1.96}{\sqrt{n}}]$$

🥾 Bootstrap

- Bootstrap is a **sampling** method which we use to do estimations (and even train ML algorithms, as in Boosting/Bagging method).
- The bootstrap approach relies on the assumption that drawing random samples **with replacement** from the data, simulates well the actual distribution of the data.
- In that case, several **statistics** (e.g. the variance) can be calculated using random samples from the data.

- **Motivation** - in general, we cannot compute the confidence interval. Thus, we need a numerical scheme to estimate it. This is where bootstrapping helps us.
- In sampling **with replacement**, each sample unit of the population can occur one or more times in the sample.
  - In statistics, resampling with replacement is called *bootstrapping*.

- **Bootstrap Algorithm**:
  - Denote the original sample: $L_N = (x_1, x_2, \dots, x_N)$
  - Repeat M times:
    - Generate a sample L_k of size k from L_N by sampling *with replacement* (samples can contain multiple copies of the same point).
    - Compute h from L_k (that is, calculate an estimation h using L_k).
  - Denote the bootstrap values $H = (h^1, h^2, \dots, h^M)$
    - Use these values for calculating all the quantities of interest.

Exercise - Bootstrap for the Median Estimator

Given n i.i.d. samples, $\{X_i\}_{i=1}^n$, from $\mathcal{D}$ consider the median estimator denoted by $\hat{\mu}_M$ and calculate the **standard deviation** using the bootstrap approach. Write down the boostrap algorithm for that case.

Solution

The algorithm:

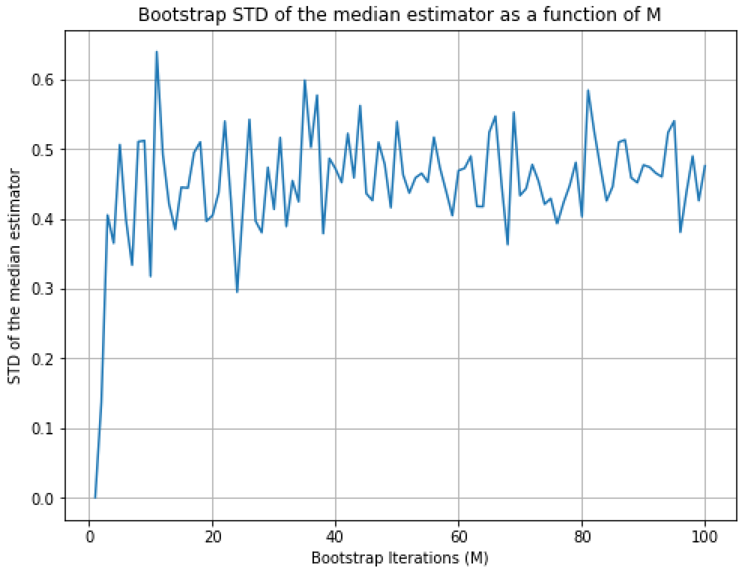
- Denote the original sample: $\mathcal{D} = X_n = (x_1, x_2, \dots, x_n)$
- Calculate the median estimator T: $T \leftarrow median(\mathcal{D})$
- Repeat M times:
  - Generate a sample L_k of size k from L_N by sampling *with replacement*.
  - Compute T^k_{boot} from L_k (that is, calculate an estimation for the median T_{boot} using L_k).
    - $T^k_{boot} = median(L_k)$
- Denote the bootstrap values $\hat{T} = (T^1_{boot}, T^2_{boot}, \dots, T^M_{boot})$
  - $Median \leftarrow T$
  - $\hat{\sigma}^2 \leftarrow Variance(\hat{T})$

In [2]:

```python
# Let's see the magic
def std_bootstrap(x, k, m, estimator=np.mean):
    estimate_value = estimator(x)
    estimations = []
    for _ in range(m):
        # sample K with replacement
        x_k = np.random.choice(x, size=k, replace=True)
        t_k = estimator(x_k)
        estimations.append(t_k)
    return estimate_value, np.std(estimations)

# generate real data
n = 300  # number of totals samples
K = 100  # number of bootstrap samples
mu , sigma = 1, 4
x = np.random.normal(mu, sigma, size=n)

M = np.arange(1, 101)  # number of boostrap iterations

stds = []
for m in M:
    stds.append(std_bootstrap(x=x, k=K, m=m, estimator=np.median)[1])
    # [1] because we don't care about the estimate, just the std
```

In [3]:
```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(1,1,1)
ax.plot(M, stds)
ax.grid()
ax.set_xlabel("Bootstrap Iterations (M)")
ax.set_ylabel("STD of the median estimator")
ax.set_title("Bootstrap STD of the median estimator as a function of M")
```

Out[3]:
```
Text(0.5, 1.0, 'Bootstrap STD of the median estimator as a function of M')
```



- Bootstrap is a general sampling method that can be used to estimate other values other than the STD.
- It has no convergence guarantees (unlike the sample mean for example). Thus, one should consider that values estimated with Bootstrap may be misleading.
- Usually, Bootstrap is used for non-trivial estimators like the median or the correlation estimators (estimators that cannot be easily defined as an expectancy $\mathbb{E}$ of an R.V.).

Confidence Interval with Chebyshev Inequality

- Recall the **Chebyshev Inequality**: if X is a random variable with finite mean and variance σ^2, then
$$P(|X - \mathbb{E}[X]| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$
for all $\epsilon > 0$.
  - Also, letting $\epsilon = k\sigma$:
$$P(|X - \mathbb{E}[X]| \geq k\sigma) \leq \frac{1}{k^2}$$
    - This is useful when we want to measure the distance in "standard deviations" (e.g., a confidence interval of 2 standard deviations).
- Chebyshev inequality allows us to calculate CIs given the variance (which can be estimated, for instance, with bootstrapping) of a random variable.

- For the sample mean $\overline{X} = \frac{1}{N}\sum_{k=1}^{N} X_k$, Chebyshev inequality tells us:
$$P(|\overline{X} - \mu| \geq \epsilon) \leq \frac{\hat{\sigma}^2}{\epsilon^2} = \frac{\sigma^2}{N\epsilon^2}$$
  - Recall that $\hat{\sigma}$ is the normalized empirical variance for a sample of size N.
  - For example, for a 95% CI, $\frac{\sigma^2}{N\epsilon^2} = \alpha = 0.05 \rightarrow \epsilon = \frac{\sigma}{\sqrt{0.05N}}$, which leads to:
$$\overline{X} - \frac{\sigma}{\sqrt{0.05N}} \leq \mu \leq \overline{X} + \frac{\sigma}{\sqrt{0.05N}}$$
with probability at least 0.95.
- In practice, if you are not given σ, then you can't calculate the CI.

Confidence Interval with DKW Inequality

The **Dvoretzky–Kiefer–Wolfowitz** (DKW) inequality bounds how close an empirically determined distribution function will be to the distribution function from which the empirical samples are drawn.

Let $X_1, X_2, \dots, X_n$ be real-valued independent and identically distributed (i.i.d.) random variables with cumulative distribution function (CDF) $F(\cdot)$ (that is, they are sampled from F: $X_1, X_2, \dots, X_n \sim F$). Let F_n denote the associated **empirical distribution** function defined by:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{X_i \leq x\}}, x \in \mathbb{R}$$

- $F(x)$ - the probability that a *single* random variable X is smaller than x
- $F_n(x)$ - the *fraction* of random variables that are smaller than x

The DKW inequality bounds the probability that the random function F_n differs from F by more than a given constant $\epsilon > 0$ anywhere on the real line ($\mathbb{R}$):

$$Pr\Big(\underset{x\in\mathbb{R}}{sup}|F_n(x) - F(x)| > \epsilon\Big) \leq 2e^{-2n\epsilon^2}, \forall\epsilon > 0$$

- This strengthens the **Glivenko–Cantelli** theorem by quantifying the *rate of convergence* as n tends to infinity.
  - But it holds for any n.
- Reminder: **Supremum**-sup - the least element in a (partially ordered) set P that is greater or equal to all elements of a subset S, if such element exists.

Producing CDF-based Confidence Bounds

- The DKW inequality is one method for generating **CDF-based confidence bounds** and producing a confidence band.
- The purpose of this confidence interval is to contain the entire CDF at the specified **confidence level** .
  - Alternative approaches attempt to only achieve the confidence level on each *individual point* which can allow for a tighter bound.
- The DKW bounds runs parallel to, and is equally above and below, the empirical CDF.

The interval that contains the **true CDF** $F(x)$, with probability $1-\alpha$:

$$F_n(x) - \epsilon \leq F(x) \leq F_n(x) + \epsilon$$

$$\alpha = 2e^{-2n\epsilon^2} \rightarrow \epsilon = \sqrt{\frac{ln\frac{2}{\alpha}}{2n}}$$

- What is so special about the DKW condifence bounds?
  - It is **distribution free** - therefore, it is very loose in comparison to the bootstrap estimates, which are usually tighter.
  - **Faster** to calculate.

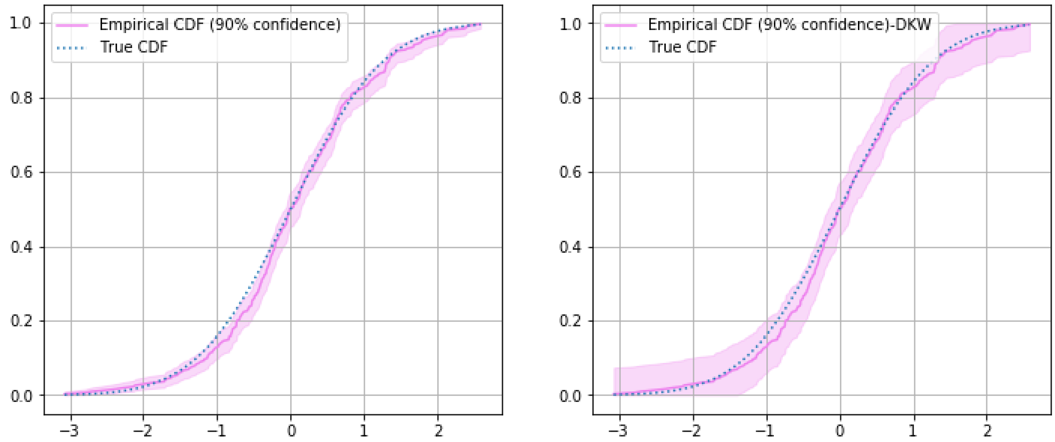
In [4]:

```
def compare_cdf_bounds():
    # generate random data
    x = np.random.randn(300)
    # plot
    fig = plt.figure(figsize=(12,5))
    ax1 = fig.add_subplot(1,2,1)
    ax1.grid()
    s_time = time.time()
    # Pointwise estimate (for each point!) confidence bounds using Beta distribution
    plot_cdf_confidence(data=x, confidence=0.9, color='violet', label='Empirical CDF (90% confidence)', ax
=ax1)
    beta_time = time.time() - s_time
    x_sorted = np.sort(x)
    ax1.plot(x_sorted, norm.cdf(x_sorted), ':', label='True CDF')
    ax1.legend()

    ax2 = fig.add_subplot(1,2,2)
    ax2.grid()
    s_time = time.time()
    # DKW (not pointwise!) confidence bounds
    plot_cdf_confidence(data=x, confidence=0.9, color='violet', label='Empirical CDF (90% confidence)-DKW'
,
                        estimator_name='DKW', ax=ax2)
    dkw_time = time.time() - s_time
    x_sorted = np.sort(x)
    ax2.plot(x_sorted, norm.cdf(x_sorted), ':', label='True CDF')
    ax2.legend()
    print("pointwise: {:.3} sec, DKW: {:.3} sec".format(beta_time, dkw_time))
```

In [5]:

```
compare_cdf_bounds()
```

```
pointwise: 0.121 sec, DKW: 0.00598 sec
```



Exercise - Confidence Intervals Calculation

- Reminder: **Hoeffding Inequality**: Let $X_1, \dots, X_n$ be i.i.d. random variables, bounded by the intervals $a_i \leq X_i \leq b_i$. Let the empirical mean be defined according to $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. It holds that:

$$P(|\overline{X} - \mathbb{E}[\overline{X}]| \geq \epsilon) \leq 2e^{-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(a_i - b_i)^2}}$$

Derive a 95% confidence band for the mean ($\mathbb{E}[X] = \mu$) using **DKW** (through the CDF bound) and **Hoeffding** inequalities.

Assume:

- True CDF: $F(x)$, estimated CDF: $F_n(x)$
- Estimated mean: $\overline{X}$
- Number of samples $n = 60$
- Prior knowledge: the range of the random variables is $X \in [0, 45]$
  - No specific distribution.

Solution

Section 1

Using DKW:

$$P(\underset{x\in\mathbb{R}}{sup}|F_n(x) - F(x)| > \epsilon) \leq 2e^{-2n\epsilon^2} = 0.05$$
$$\rightarrow \epsilon = \sqrt{\frac{-\ln(0.025)}{2*60}} = 0.175$$
$$\rightarrow F(x) \in [F_n(x) - 0.175, F_n(x) + 0.175]$$

- Deriving a CI for the mean: we know that $\mathbb{E}[X] = \int_0^{45}(1 - F(x))dx$ (from the Tail Sum formula), therefore:
$$\int_0^{45}(1 - F_n(x) \pm 0.175)dx = \overline{X} \pm 7.875$$
  - Why is $\mathbb{E}[X] = \int(1 - F(x))dx$? Recall the Tail Sum formula from previous tutorial ($\mathbb{E}[X] = \sum_{i=1}^{n} P(X \geq i)$). In the continuous space it is completely analogous!
$$\mathbb{E}[X] = \int_x P(X \geq x)dx = \int_x (1 - P(X \leq x))dx = \int_x (1 - F(x))dx$$
    - Proof (https://en.wikipedia.org/wiki/Expected_value#General_case_2)

Section 2

Using Hoeffding:

$$P(|\overline{X} - \mathbb{E}[\overline{X}]| \geq \epsilon) \leq 2e^{-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(a_i - b_i)^2}} = 2e^{-\frac{2\cdot 60^2\epsilon^2}{\sum_{i=1}^{60}(45-0)^2}} = 0.05$$
$$\rightarrow \epsilon = 7.89 \rightarrow \mathbb{E}[X] \in [\overline{X} - 7.89, \overline{X} + 7.89]$$

- We see that both approaches yield similar CI (same range) for the mean. In this sense, the estimators are equivalent.

Recommended Videos

Warning!

- These videos do not replace the lectures and tutorials.
- Please use these to get a better understanding of the material, and not as an alternative to the written material.

Video By Subject

- Confidence Intervals - CrashCourse (https://www.youtube.com/watch?v=yDEvXB6ApWc&t=600s)
- Bootstrap - Confidence Intervals: Intro to bootstrapping proportions (https://www.youtube.com/watch?v=655X9eZGxls)

Credits

- Examples, exercises and definitions from Introduction to Probability, Statistics and Random Processes (https://probabilitycourse.com/) - https://probabilitycourse.com (https://probabilitycourse.com)
- Icons from Icon8.com (https://icons8.com/) - https://icons8.com (https://icons8.com)
- Datasets from Kaggle (https://www.kaggle.com/) - https://www.kaggle.com/ (https://www.kaggle.com/)