Table of Contents

References

- scipy stats (WARNING: in `scipy stats t interval` alpha actually means confidence level `1 - alpha`)

Confidence interval

A confidence interval for a mean is a range of values that is likely to contain a population mean with a certain level of confidence.

It is calculated as:

```
Confidence Interval = xbar  +/-  t_crit * (s/√n) for small sample size
n<30
Confidence Interval = xbar  +/-  z_crit * (s/√n) for larger sample size
n>=30

where:

xbar: sample mean
t_crit : t-value that corresponds to the confidence level
z_crit : z-value at corresponds to the confidence level, z_crit = 1.96 for
5% significance level two tailed normal distribution.
s: sample standard deviation (unbiased)
n: sample size
sem = s/sqrt(n) # NOTE: np.std np.nanstd uses biased ddof=0 but stats.sem
uses unbiased ddof=1 by default.
```

```
ci = (phat - margin_of_error, phat + margin_of_error)

moe = z_crit * standard_error

z_crit = 1.96 for two-tailed test at 5% significance level.
se = sigma / sqrt(n)

For coin toss:
Z = (phat - p0) / sqrt(p0q0/n)
pval = 1 - stats.norm.cdf(Z)

if pval < alpha:
    Reject Null Hypothesis that coin is unbiased and say we have enough
evidence to call it biased.
```

Formula

$$\mathrm{MOE}_\gamma = z_\gamma \times \sqrt{\frac{\sigma^2}{n}}$$

MOE = margin of error
γ = confidence level
z_γ = quantile
σ = standard deviation
n = sample size

The confidence level tells you how sure you can be and is expressed as a percentage.

- The 95% confidence level means you can be 95% certain.
- The 99% confidence level means you can be 99% certain.

α (Alpha) is called the significance level, and is the probability of rejecting the null hypothesis when it is true.

- It is usually set at or below 5%.
- If your significance level is 0.05, the corresponding confidence level is 95%.
- If your significance level of 0.05, there's a 5% risk of concluding that a difference exists when there is no actual difference.

Confidence interval using Python vs R

- reference
- reference

```
import numpy as np, scipy.stats as st

# returns confidence interval of mean
def confIntMean(a, conf=0.95):
  mean, sem, m = np.mean(a), st.sem(a), st.t.ppf((1+conf)/2., len(a)-1)
  return mean - m*sem, mean + m*sem

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
    return m, m-h, m+h

a = np.array([1,2,3,4,4,4,5,5,5,5,4,4,4,6,7,8])

# better method
confIntMean(a, 0.68) # (3.9974214366806184, 4.877578563319382)

# scipy method
st.norm.interval(0.68, loc=np.mean(a), scale=st.sem(a))
#(4.0120010966037407, 4.8629989033962593)

# ====================== Example 02 ============================
# using statsmodels
# https://stackoverflow.com/questions/15033511/compute-a-confidence-
interval-from-sample-data/15034143#15034143
import statsmodels.stats.api as sms
sms.DescrStatsW(a).tconfint_mean()

a = list(range(10,14))

mean_confidence_interval(a)
# (11.5, 9.4457397432391215, 13.554260256760879)

st.t.interval(0.95, len(a)-1, loc=np.mean(a), scale=st.sem(a))
# (9.4457397432391215, 13.554260256760879)

# sms.DescrStatsW(a).tconfint_mean()
# (9.4457397432391197, 13.55426025676088)

# And finally, the incorrect result using st.norm.interval():
st.norm.interval(0.95, loc=np.mean(a), scale=st.sem(a))
# (10.23484868811834, 12.76515131188166)
```

Bootstrapping for confidence interval

```
import numpy as np
import tqdm
from scipy import stats

alpha = 0.05
a = np.array([1,2,3,4,4,4,5,5,5,5,4,4,4,6,7,8])


reps = 1_000

sample = a # suppose that a is sample drawn from big population
ci_points = [] # point estimate
for _ in tqdm.trange(reps):
    bootsample = np.random.choice(sample, size=len(sample),replace=True)

    # make sure to use bootsample, not sample!
    ci_lo, ci_hi = stats.t.interval(1-alpha,
                                    df = len(bootsample)-1,
                                    loc=np.mean(bootsample),
                                    scale=stats.sem(bootsample))
    ci_point = (ci_lo+ci_hi)/2
    ci_points.append(ci_point)

ci_point = np.mean(ci_points)
ci_lo, ci_hi = np.percentile(ci_points, [alpha/2*100,100-alpha/2*100]) #
alpha/2*100 is 2.5
```

Z critical

```
import numpy as np
from scipy import stats

# alpha to critical
alpha = 0.05
n_sided = 2 # 2-sided test
z_crit = stats.norm.ppf(1-alpha/n_sided)
print(z_crit) # 1.959963984540054

# critical to alpha
alpha = stats.norm.sf(z_crit) * n_sided
print(alpha) # 0.05
```

Confidence interval for t-distribution

- statology

```
import numpy as np
from scipy scipy import stats

# define sample data
data = [12, 12, 13, 13, 15, 16, 17, 22, 23, 25, 26, 27, 28, 28, 29]

# create 95% confidence interval for population mean weight
# sem is standard error of mean
# significance level alpha = 0.05 and confidence level gamma = 0.95
# even though scipy stats t interval has first parameter called alpha it
is 1-alpha
alpha = 0.05
stats.t.interval(1-alpha, df=len(data)-1, loc=np.mean(data),
scale=st.sem(data))
# (17.40, 21.08)

# create 99% confidence interval for same sample
alpha = 0.01
stats.norm.interval(1-alpha, loc=np.mean(data), scale=st.sem(data))
# (16.82, 21.66)
# if confidence level is high, confidence level is wider
```

Confidence interval for z-distribution

- statology

```
import numpy as np
import scipy.stats as stats

#define sample data
np.random.seed(0)
data = np.random.randint(10, 30, 50)

#create 95% confidence interval for population mean weight
# significance level alpha = 0.05 and confidence level gamma = 0.95
stats.norm.interval(alpha=0.95, loc=np.mean(data), scale=st.sem(data))
```

Confidence interval example

- https://vedexcel.com/how-to-calculate-confidence-intervals-in-python/

```
import numpy as np
from scipy import stats

# define given sample data
data = [45, 55, 67, 45, 68, 79, 98, 87, 84, 82]
```

```
# Calculate the sample parameters
significanceLevel   alpha = 0.05          # 5%
confidenceLevel     CL    = 0.95          # 95% CI given
degrees_freedom     df    = len(data)-1   # degree of freedom = sample
size-1
sampleMean          xbar  = np.mean(data) # sample mean
sampleStandardError sem   = stats.sem(data,ddof=1)  # sample standard
error (default is already ddof=1 unlike numpy)

# create 95% confidence interval for the population mean
# scipy stats interval function has parameter alpha actually equal to 1-
alpha
confidenceInterval = stats.t.interval(alpha=confidenceLevel,
df=degrees_freedom, loc=sampleMean, scale=sampleStandardError)

# print the 95% confidence interval for the population mean
print('The 95% confidence interval for the population mean
:',confidenceInterval)
```

Confidence interval example

- https://vedexcel.com/how-to-calculate-confidence-intervals-in-python/

```
import numpy as np
import scipy.stats as st

# define given sample data
data = [87,80,68,72,56,58,60,63,82,70,58,55,48,50,77]

# Calculate the sample parameters
confidenceLevel     = 0.98          # 98% CI given
degrees_freedom     = len(data)-1   # degree of freedom = sample size-1
sampleMean          = np.mean(data) # sample mean
sampleStandardError = st.sem(data)  # sample standard error

#create 98% confidence interval for the population mean
confidenceInterval = st.t.interval(alpha=confidenceLevel,
df=degrees_freedom, loc=sampleMean, scale=sampleStandardError)

#print the 98% confidence interval for the population mean
print('The 98% confidence interval for the population mean weight
:',confidenceInterval)
```

CI using bootstrapping

- https://stackoverflow.com/questions/44392978/compute-a-confidence-interval-from-sample-data-assuming-unknown-distribution/66008548#66008548

```
def bootstrap_ci(
    data,
    statfunction=np.average,
    alpha = 0.05,
    n_samples = 100):

    """inspired by https://github.com/cgevans/scikits-bootstrap"""
    import warnings

    def bootstrap_ids(data, n_samples=100):
        for _ in range(n_samples):
            yield np.random.randint(data.shape[0], size=(data.shape[0],))

    alphas = np.array([alpha/2, 1 - alpha/2])
    nvals = np.round((n_samples - 1) * alphas).astype(int)
    if np.any(nvals < 10) or np.any(nvals >= n_samples-10):
        warnings.warn("Some values used extremal samples; results are
probably unstable. "
                      "Try to increase n_samples")

    data = np.array(data)
    if np.prod(data.shape) != max(data.shape):
        raise ValueError("Data must be 1D")
    data = data.ravel()

    boot_indexes = bootstrap_ids(data, n_samples)
    stat = np.asarray([statfunction(data[_ids]) for _ids in boot_indexes])
    stat.sort(axis=0)

    return stat[nvals]

# usage
# simulate some data from a pareto distribution

np.random.seed(33)
data = np.random.pareto(a=1, size=111)
sample_mean = np.mean(data)

plt.hist(data, bins=25)
plt.axvline(sample_mean, c='red', label='sample mean'); plt.legend()

# generate confidence intervals for the SAMPLE MEAN with bootstrapping
low_ci, up_ci = bootstrap_ci(data, np.mean, n_samples=1000)
#plot the resuts

plt.hist(data, bins=25)
plt.axvline(low_ci, c='orange', label='low_ci mean')
plt.axvline(up_ci, c='magenta', label='up_ci mean')
plt.axvline(sample_mean, c='red', label='sample mean'); plt.legend()
```

```
#generate confidence intervals for the DISTRIBUTION PARAMETERS with
bootstrapping
from scipy.stats import pareto

true_params = pareto.fit(data)
low_ci, up_ci = bootstrap_ci(data, pareto.fit, n_samples=1000)
# low_ci[0] and up_ci[0] are the confidence intervals for the shape param
# low_ci[0], true_params[0], up_ci[0] ---> (0.8786, 1.0983, 1.4599)
```

CI using bootstrapping for two groups

- https://github.com/SUN-Wenjun/bootstrapping

```
import numpy as np

def bootstrap_ci_two_groups(df, variable, classes, repetitions = 1000,
alpha = 0.05, random_state=None):
    # df: a data frame that includes observations of the two sample
    # variable: the column name of the column that includes observations
    # classes: the column name of the column that includes group
assignment (This column should contain two different group names)
    # repetitions: number of times you want the bootstrapping to repeat.
Default is 1000.
    # alpha: likelihood that the true population parameter lies outside
the confidence interval. Default is 0.05.
    # random_stata: enable users to set their own random_state, default is
None.

    df = df[[variable, classes]]
    bootstrap_sample_size = len(df)

    mean_diffs = []

    for i in range(repetitions):
        bootstrap_sample = df.sample(n = bootstrap_sample_size, replace =
True, random_state = random_state)
        mean_diff = bootstrap_sample.groupby(classes).mean().iloc[1,0] -
bootstrap_sample.groupby(classes).mean().iloc[0,0]
        mean_diffs.append(mean_diff)

    # confidence interval
    left = np.percentile(mean_diffs, alpha/2*100)
    right = np.percentile(mean_diffs, 100-alpha/2*100)

    # point estimate
    point_est = df.groupby(classes).mean().iloc[1,0] -
df.groupby(classes).mean().iloc[0,0]

    print('Point estimate of difference between means:',
```

```
round(point_est,2))
    print((1-alpha)*100,'%','confidence interval for the difference
between means:', (round(left,2), round(right,2)))
```

Images

$$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Confidence interval formula

Normal Approximation Method of the Binomial Confidence Interval

The equation for the Normal Approximation for the Binomial CI is shown below.

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

- where p = proportion of interest
- n = sample size
- α = desired confidence
- $z_{1-\alpha/2}$ = "z value" for desired level of confidence
- $z_{1-\alpha/2}$ = 1.96 for 95% confidence
- $z_{1-\alpha/2}$ = 2.57 for 99% confidence
- $z_{1-\alpha/2}$ = 3 for 99.73% confidence

286) What is the relationship between significance level and confidence level?

A) Significance level = Confidence level
B) Significance level = 1- Confidence level
C) Significance level = 1/Confidence level
D) Significance level = sqrt (1 – Confidence level)

Solution: (B)

Significance level is 1-confidence interval. If the significance level is 0.05, the corresponding confidence interval is 95% or 0.95. The significance level is the probability of obtaining a result as extreme as, or more extreme than, the result actually obtained when the null hypothesis is true. The confidence interval is the range of likely values for a population parameter, such as the population mean. For example, if you compute a 95% confidence interval for the average price of an ice cream, then you can be 95% confident that the interval contains the true average cost of all ice creams.

The significance level and confidence level are the complementary portions in the normal distribution.

3. Formula for confidence interval varies with statistics

For confidence interval of mean

$$\text{C.I.}_{\text{mean}}: \quad \mu \pm (t_{\frac{\alpha}{2},df} \times \frac{s}{\sqrt{n}})$$

For confidence interval of difference in mean

$$\text{C.I.}_{\Delta\text{mean}}: \quad (\mu_1 - \mu_2) \pm (t_{1-\frac{\alpha}{2},df} \times \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}})$$

For confidence interval of proportion

$$\text{C.I.}_{\text{proportion}}: \quad \hat{p} \pm (t_{\frac{\alpha}{2},df} \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}})$$

For confidence interval of variance

$$\text{C.I.}_{\text{variance}}: \frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2}}} \leq \sigma^2 \leq \frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2}}}$$

For confidence interval of standard deviation

$$\text{C.I.}_{\text{standard deviation}}: \sqrt{\frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2}}}} \leq \sigma \leq \sqrt{\frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2}}}}$$

Different analytical solutions exist for different statistics. However, confidence interval for many other statistics cannot be analytically solved, simply because there are no formulas for them. If the statistic of your interest does not have an analytical solution for its confidence interval, or you simply don't know it, numerical methods like boostrapping can be a good alternative (and its powerful).

Misunderstandings [edit]

See also: § Counter-examples, and Misunderstandings of p-values

Confidence intervals and levels are frequently misunderstood, and published studies have shown that even professional scientists often misinterpret them.[7][8][9][10][11]

- A 95% confidence level does not mean that for a given realized interval there is a 95% probability that the population parameter lies within the interval (i.e., a 95% probability that the interval covers the population parameter).[12] According to the strict frequentist interpretation, once an interval is calculated, this interval either covers the parameter value or it does not; it is no longer a matter of probability. The 95% probability relates to the reliability of the estimation procedure, not to a specific calculated interval.[13] Neyman himself (the original proponent of confidence intervals) made this point in his original paper:[5]

  > "It will be noticed that in the above description, the probability statements refer to the problems of estimation with which the statistician will be concerned in the future. In fact, I have repeatedly stated that the frequency of correct results will tend to α. Consider now the case when a sample is already drawn, and the calculations have given [particular limits]. Can we say that in this particular case the probability of the true value [falling between these limits] is equal to α? The answer is obviously in the negative. The parameter is an unknown constant, and no probability statement concerning its value may be made..."

  Deborah Mayo expands on this further as follows:[14]

  > "It must be stressed, however, that having seen the value [of the data], Neyman–Pearson theory never permits one to conclude that the specific confidence interval formed covers the true value of 0 with either $(1 - \alpha)$100% probability or $(1 - \alpha)$100% degree of confidence. Seidenfeld's remark seems rooted in a (not uncommon) desire for Neyman–Pearson confidence intervals to provide something which they cannot legitimately provide; namely, a measure of the degree of probability, belief, or support that an unknown parameter value lies in a specific interval. Following Savage (1962), the probability that a parameter lies in a specific interval may be referred to as a measure of final precision. While a measure of final precision may seem desirable, and while confidence levels are often (wrongly) interpreted as providing such a measure, no such interpretation is warranted. Admittedly, such a misinterpretation is encouraged by the word 'confidence'."

- A 95% confidence level does not mean that 95% of the sample data lie within the confidence interval.
- A confidence interval is not a definitive range of plausible values for the sample parameter, though it may be understood as an estimate of plausible values for the population parameter.
- A particular confidence level of 95% calculated from an experiment does not mean that there is a 95% probability of a sample parameter from a repeat of the experiment falling within this interval.[11]

269) What happens to the confidence interval when we introduce some outliers to the data?

A) Confidence interval is robust to outliers

B) Confidence interval will increase with the introduction of outliers.

C) Confidence interval will decrease with the introduction of outliers.

D) We cannot determine the confidence interval in this case.

Solution: (B)

We know that confidence interval depends on the standard deviation of the data. If we introduce outliers into the data, the standard deviation increases, and hence the confidence interval also increases.

Formula

$$\mathrm{MOE}_\gamma = z_\gamma \times \sqrt{\frac{\sigma^2}{n}}$$

MOE = margin of error

γ = confidence level

z_γ = quantile

σ = standard deviation

n = sample size

If $X \sim \text{Pois}(\lambda)$. then $E(X) = \lambda$ and $SD(X) = \sqrt{\lambda}$. For sufficiently large λ, the random variable X is approximately normally distributed. Then one says that $Z = \frac{X-\lambda}{\sqrt{\lambda}}$ is approximately standard normal, so that

$$P\left(-1.96 < \frac{X-\lambda}{\sqrt{\lambda}} < 1.96\right) \approx 0.95.$$

This gives rise to $P(X - 1.96\sqrt{\lambda} < \lambda < X + 1.96\sqrt{\lambda}) \approx 0.95$. Again, for sufficiently large λ, one says that $1.96\sqrt{\lambda} \approx 1.96\sqrt{X}$. So finally, an approximate 95% confidence interval for λ is of the form

$$(X - 1.96\sqrt{X},\ X + 1.96\sqrt{X}).$$

This type of interval was proposed by Wald as asymptotically accurate for $\lambda \to \infty$. It works reasonably well for $\lambda > 50$. For smaller λ, a confidence interval with somewhat closer to 95% coverage is

$$(X + 2 - 1.96\sqrt{X+1},\ X + 2 + 1.96\sqrt{X+1}).$$

> *Rationale:* This adjusted 95% interval for smaller λ is based on 'inverting' a standard test for $H_0 : \lambda = \lambda_0$ vs. $H_a : \lambda \neq \lambda_0$, with test statistic $Z = \frac{X-\lambda_0}{\sqrt{\lambda_0}}$, which rejects at the 5% level for $|Z| \geq 1.96$. Specifically for given X, the adjusted interval is found by solving a quadratic inequality for values λ_0 with $|Z| < 1.96$ and conflating 1.96 with 2 to obtain the terms with $X + 2$ and $X + 1$. In effect, the adjusted CI consists of non-rejectable hypothetical values of λ_0. (One still assumes that Z is approximately standard normal, but the additional assumption that $1.96\sqrt{\lambda} \approx 1.96\sqrt{X}$ is no longer required.)

For both styles of CIs, an approximate 90% confidence interval is shorter, using ± 1.645 instead of ± 1.96.

$$(\bar{x}_1 - \bar{x}_2) \pm t\ S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Use the t-table with degrees of freedom = n_1+n_2-2

Confidence Interval = mean +/- t-score * standard error *(see above)*

mean = new mean — old mean = 3–5 = -2

t-score = 2.101 given df=18 (20–2) and confidence interval of 95%

$$SE(\bar{x}_1 - \bar{x}_2) = S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

$$S_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

standard error = sqrt((0.^{62}*9+0.^{682}*9)/(10+10–2)) * sqrt(1/10+1/10)
standard error = 0.352

confidence interval = [-2.75, -1.25]

Assuming we subtract in this order (New System — Old System):

$$(\bar{x}_1 - \bar{x}_2) \pm z\, S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Use Z table for standard normal distribution

confidence interval formula for two independent samples

mean = new mean — old mean = 4–6 = -2

z-score = 1.96 confidence interval of 95%

$$SE(\bar{x}_1 - \bar{x}_2) = S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

$$S_p = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$$

st. error = sqrt((0.^{52}*99+^{22}*99)/(100+100–2)) * sqrt(1/100+1/100)
standard error = 0.205061
lower bound = -2–1.96*0.205061 = -2.40192
upper bound = -2+1.96*0.205061 = -1.59808

confidence interval = [-2.40192, -1.59808]

$$p = \frac{29}{50} = .58$$

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

$$CI = .58 \pm 1.96 \sqrt{\frac{.58(1-.58)}{50}}$$

$$CI = .58 \pm .14$$

Equal or unequal sample sizes, unequal variances ($s_{X_1} > 2s_{X_2}$ or $s_{X_2} > 2s_{X_1}$) [edit]

Main article: Welch's t-test

This test, also known as Welch's *t*-test, is used only when the two population variances are not assumed to be equal (the two sample sizes may or may not be equal) and hence must be estimated separately. The *t* statistic to test whether the population means are different is calculated as:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{\Delta}}}$$

where

$$s_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}.$$

Here s_i^2 is the unbiased estimator of the variance of each of the two samples with n_i = number of participants in group i (1 or 2). In this case $(s_{\bar{\Delta}})^2$ is not a pooled variance. For use in significance testing, the distribution of the test statistic is approximated as an ordinary Student's *t*-distribution with the degrees of freedom calculated using

$$\mathrm{d.f.} = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}.$$

This is known as the Welch–Satterthwaite equation. The true distribution of the test statistic actually depends (slightly) on the two unknown population variances (see Behrens–Fisher problem).

Independent two-sample *t*-test [edit]

Equal sample sizes and variance [edit]

Given two groups (1, 2), this test is only applicable when:

- the two sample sizes (that is, the number n of participants of each group) are equal;
- it can be assumed that the two distributions have the same variance;

Violations of these assumptions are discussed below.

The *t* statistic to test whether the means are different can be calculated as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \sqrt{\frac{2}{n}}}$$

where

$$s_p = \sqrt{\frac{s_{X_1}^2 + s_{X_2}^2}{2}}.$$

Here s_p is the pooled standard deviation for $n = n_1 = n_2$ and $s_{X_1}^2$ and $s_{X_2}^2$ are the unbiased estimators of the variances of the two samples. The denominator of *t* is the standard error of the difference between two means.

For significance testing, the degrees of freedom for this test is $2n - 2$ where n is the number of participants in each group.

Equal or unequal sample sizes, similar variances ($\frac{1}{2} < \frac{s_{X_1}}{s_{X_2}} < 2$) [edit]

This test is used only when it can be assumed that the two distributions have the same variance. (When this assumption is violated, see below.) The previous formulae are a special case of the formulae below, one recovers them when both samples are equal in size: $n = n_1 = n_2$.

The *t* statistic to test whether the means are different can be calculated as follows:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where

$$s_p = \sqrt{\frac{(n_1 - 1)\, s_{X_1}^2 + (n_2 - 1)\, s_{X_2}^2}{n_1 + n_2 - 2}}$$

is an estimator of the pooled standard deviation of the two samples: it is defined in this way so that its square is an unbiased estimator of the common variance whether or not the population means are the same. In these formulae, $n_i - 1$ is the number of degrees of freedom for each group, and the total sample size minus two (that is, $n_1 + n_2 - 2$) is the total number of degrees of freedom, which is used in significance testing.

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

Questions

36) From a sample the 95% confidence interval is already computed. What is the probability that the population parameter lies in the interval?

a)0.95

b)0.5

c)0.05

d)None of these

Ans: **b) 0.5**

This is a tricky question which requires a comprehensive explanation. The 'misunderstandings' section in Wikipedia has a good explanation. Read more for detailed explaination.

Example 1: Uncertainty in rock porosity

(Borrowed from Dr. Michael Pyrcz's Geostatistics class)

A reservoir engineer in the oil & gas industry wants to know the rock porosity of a formation to estimate the total oil reserve 9,500 ft underground. Due to the high cost of obtaining rock core samples from the deep formations, he could acquire only 12 rock core samples. Since the uncertainty of a point estimation scales inversely with a sample size, his estimation is subject to non-negligible uncertainty. He obtains 14.5% average rock porosity with 4.3% standard deviation. Executives in the company wants to know the worst-case scenario (P10) and the best-case scenario (P90) to make business decisions. You can convey your estimation of average porosity with uncertainty by constructing the confidence interval of mean.

Assuming that you have a reason to believe that the rock porosity follows normal distribution, you can construct its 80% confidence interval, with the procedure described below:

```
stats.t.interval(1 - 0.2, 12 - 1, loc=14.5, scale= 4.3 / np.sqrt(12))
```

```
(12.807569748569543, 16.19243025143046)
```

The above range of uncertainty was acquired from the 12 rock core samples. In the worst-case scenario, the rock formation at 9,500 ft underground has 12.8% porosity. In the best-case scenario, the oil reservoir has 16.2% porosity. The same procedures can be applied for the core samples collected at different depths, which give us the confidence interval plot of rock porosities shown in figure (2).

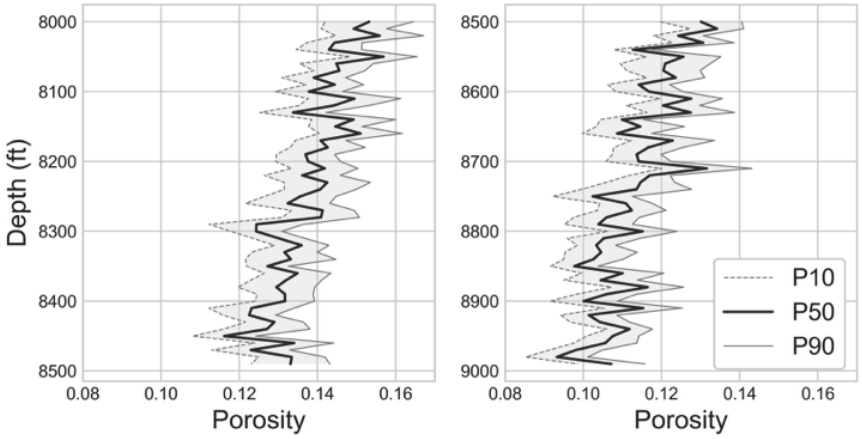

Figure 2: Confidence interval of core samples porosities along depths

Example 2: Purity of methamphetamine (crystal) in Breaking Bad

21 batches of crystal cooked by Mr. White shows 99.1% average purity with 3% standard deviation. 18 batches of crystal cooked by Mr. Pinkman shows 96.2% average purity with 4% standard deviation. Does Mr. White always cook better crystal than Mr. Pinkman, or is it possible for Mr. Pinkman to beat Mr. White in purity of cooked crystals, by luck? We can construct 95% confidence interval assuming normal distribution, with the procedure described below:

```
# Mr. White's

stats.t.interval(1 - 0.05, 21 - 1, loc=99.1, scale= 3 / np.sqrt(21))
```

```
(97.73441637228476, 100.46558362771523)
```

```
# Mr. Pinkman's

stats.t.interval(1 - 0.05, 18 - 1, loc=96.2, scale= 4 / np.sqrt(18))
```

```
(94.21084679714819, 98.18915320285181)
```

There's a small overlap between the confidence intervals of Mr. White's and Mr. Pinkman's. Although it is true that Mr. White is a better cooker, Mr. Pinkman can cook a purer batch of crystals by a small chance, if he has the luck. Comparing the means of two sample data sets is closely related to constructing confidence interval of difference in mean.

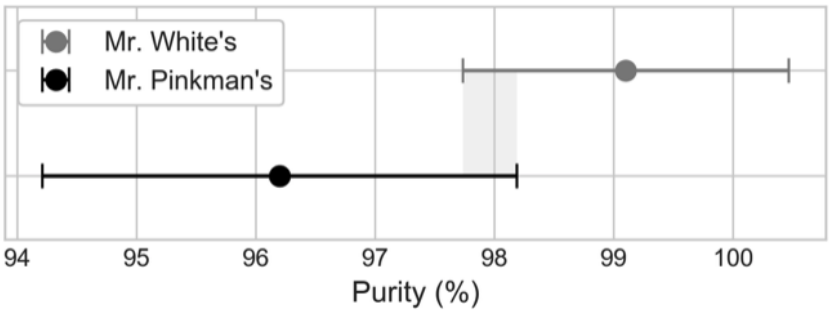

Figure 3: Overlap in the 95% confidence interval of two samples