Aufgabe 6.1 Konfidenzintervalle

(A) In eine fiktiven Studie im frühen 20. Jahrhundert wurden 50 Seiten in einer Shakespeare-Gesamtausgabe durch zufälliges Aufschlagen ausgewählt und die Type-Token-Ratios (TTR) bestimmt. Der Vergleich mit 50 Zufällig aufgeschlagenen Seiten im Werk seines Zeitgenossen Christopher Marlowe ergab, daß Shakespeares Texte einen signifikant höheren TTR-Wert haben. Anfang des 21. Jahrhunderts wurde eine zweite Studie durchgeführt in der die TTR-Werte der digitalisierten Gesamtwerke beider Autoren bestimmt wurden. Die zweite Studie ergab, daß die TTR-Werte beider Autoren gleich sind. Welcher Fehler (1. oder 2. Art) ist der früheren Studie unterlaufen?

Es handelt sich um einen Fehler zweiter Art. Aufgrund des fehlerhaften TTR-Wertes, der in der alten Studie höher ist, wird H0 zwar angenommen, jedoch trifft sie nicht zu. Der neuere Test widerlegt Studie 1 und beweist somit eine falschnegative Schlussfolgerung. Es liegt der Fall vor, dass existierender Effekt nicht erfasst wird, wie beim Münzbeispiel im Skript (S.3 Mitte), denn es besteht immer noch eine Restwahrscheinlichkeit von 50:50, dass die Texte bei Shakespeare nicht größer sind.

(B) Bestimmen Sie mit dem bootstrapping-Verfahren (10000 subsamples) das empirische 95%-Konfidenzintervall für das mittlere Alter der Autorinnen/Autoren in Ihrem Datensatz.

In [1]:

```python
import numpy as np

age_data = [63,42,58,73,62,30,50,71,79,59,72,82,75,69,79,71,79,76,86,78]

sample_size = len(age_data)
reps = 10000
result = []
for i in range(reps):
    subsample = np.random.choice(age_data, size=sample_size, replace=True)
    result.append(subsample.mean())
#print(result)

print(np.mean(age_data)) #ergebnis mittelwert
print(np.quantile(result, 0.025)) #untere grenze
print(np.quantile(result, 0.975)) #obere grenze

print(f'Der Wert {np.mean(age_data)} liegt genau im Intervall zwischen {np.quantile(result, 0.025)} bis {np.quantile(res
```

```
67.7
61.15
73.35
Der Wert 67.7 liegt genau im Intervall zwischen 61.15 bis 73.35.
```

6.2 Verteilungen und Differenzen

Nun gehen Sie folgendermaßen vor:

1. Bestimmen Sie die Mittelwertdifferenz der beiden vorliegenden Stichproben.
2. Zum Vergleich erzeugen Sie nun viele vergleichbare, aber simulierte Stichprobenpaare aus dem Nullmodell (der Funktion np.random.normal().
3. Bestimmen Sie für jedes dieser zufälligen Stichprobenpaare aus ein und derselben Grundgesamtheit die Mittelwertdifferenz.
4. Untersuchen Sie, in wie vielen Fällen eine zufällige Mittelwertdifferenz größer ist als die real gemessene.
5. Können Sie durch den Vergleich der gemessenen Mittelwertdifferenz mit den Quantilen der simulierten, zufälligen Mittelwertdifferenzen etwas über die statistische Signifikanz aussagen?

1)

In [2]:
```
#ergebnis mittelwert aus 6.1
print(np.mean(age_data))
```

```
67.7
```

In [3]:
```
#ergebnis liste aus angabewerten von A6.2
authors_es = [48, 68, 70, 60, 77, 68, 58, 58, 73, 81, 77, 70, 49, 85, 64, 71, 66, 61, 47, 98]
print(np.mean(authors_es))
```

```
67.45
```

In [4]:
```
#mittelwertdifferenz x1-x2
mw_dif = abs(np.mean(authors_es) - np.mean(age_data))
print(mw_dif)
```

```
0.25
```

2)

In [5]:

```
#nur für 1 zufallsbeispiel
#output= np.random.normal(size = 20)
#print(output)

#für 10 beispiele à 20
out= np.random.normal(loc = 67.7, scale = 10, size = (10,20))
rund = np.round(out)
print(rund)
```

```
[[ 81.  80.  87.  67.  58.  60.  73.  84.  54.  81.  60.  77.  64.  60.
   72.  77.  57.  77.  72.  66.]
 [ 64.  70.  70.  74.  72.  60.  70.  68.  82.  68.  54.  67.  67.  67.
   79.  80.  75.  48.  62.  86.]
 [ 77.  60.  71.  61.  55.  45.  93.  46.  69.  67.  82.  72.  73.  69.
   79.  74.  75.  50.  68.  56.]
 [ 67.  51.  69.  75.  67.  67.  55.  73.  50.  89.  67.  79.  74.  65.
   72.  73.  63.  68.  66.  72.]
 [ 75.  70.  55.  68.  79.  46.  62.  60.  75.  73.  68.  64.  66.  79.
   59.  83.  83.  57.  75.  80.]
 [ 69.  64.  72.  76.  69.  63.  62.  80.  87.  70.  60.  43.  71.  57.
   60.  64.  70. 100.  65.  86.]
 [ 59.  83.  47.  58.  79.  83.  65.  69.  78.  61.  61.  71.  55.  69.
   69.  69.  68.  78.  63.  45.]
 [ 60.  74.  58.  73.  68.  77.  74.  77.  81.  64.  76.  93.  72.  63.
   75.  67.  79.  94.  62.  76.]
 [ 77.  85.  74.  69.  62.  69.  87.  55.  66.  77.  75.  75.  80.  62.
   77.  69.  68.  79.  65.  59.]
 [ 70.  75.  47.  70.  68.  75.  76.  66.  56.  69.  66.  82.  79.  53.
   56.  69.  80.  65.  81.  62.]]
```

3)

In [6]:
```
liste = []
for i in rund:
    liste.append(np.mean(i))
print(liste)



from itertools import combinations


res = [abs(x - y) for x, y in combinations(liste, 2)]

print("\nAlle möglichen Differenzen : " + str(res))

print(f'\nAnzahl aller Differenzen:', len(res))
```

```
[70.35, 69.15, 67.1, 68.1, 68.85, 69.4, 66.5, 73.15, 71.5, 68.25]

Alle möglichen Differenzen : [1.1999999999999886, 3.25, 2.25, 1.5, 0.9499999999999886, 3.8499999999999943, 2.8000000000000
114, 1.1500000000000057, 2.0999999999999943, 2.0500000000000114, 1.0500000000000114, 0.3000000000001137, 0.25, 2.6500000
00000057, 4.0, 2.3499999999999943, 0.9000000000000057, 1.0, 1.75, 2.3000000000000114, 0.5999999999999943, 6.0500000000001
1, 4.400000000000006, 1.1500000000000057, 0.75, 1.3000000000000114, 1.5999999999999943, 5.050000000000011, 3.4000000000000
057, 0.15000000000000568, 0.5500000000000114, 2.3499999999999943, 4.300000000000011, 2.6500000000000057, 0.599999999999994
3, 2.9000000000000057, 3.75, 2.0999999999999943, 1.1500000000000057, 6.650000000000006, 5.0, 1.75, 1.6500000000000057, 4.9
00000000000006, 3.25]

Anzahl aller Differenzen: 45
```

4)

In [7]:
```
count = []
for i in res:
    if i > mw_dif:
        count.append(i)
print(f'\nIn {len(count)} Fällen ist eine zufällige MW-Differenz größer als die reale.')
```

```
In 43 Fällen ist eine zufällige MW-Differenz größer als die reale.
```

5) ¶

Nein, da die einfache Mittelwertdifferenz die Variabilität innerhalb der Stichproben nicht berücksichtigt. Man müsste die Mittelwertdifferenz noch ins Verhältnis zur Standardabweichung bringen und die Qunatile der daraus entstehenden T-Verteilungen berechnen.

6.3 Det t-Test nun richtig

(A) Implementieren Sie den t-Test nach der oben beschriebenen Formel in NumPy. Der Output ihrer Implementierung sollte der t-Wert sein, um den resultierenden P -Wert müssen Sie sich erst einmal keine Gedanken machen. Berechnen Sie den t-Wert zwischen den erreichten Lebensaltern der Autorinnen/Autoren in Ihrem eigenen Datensatz und den Daten für eine Reihe von spanischen Autoren:

In [102]:
```
import numpy as np

example = np.array([48, 68, 70, 60, 77, 68, 58, 58, 73, 81, 77, 70, 49, 85, 64, 71, 66, 61, 47, 98])
my_data = np.array([63, 42, 58, 73, 62, 30, 50, 71, 79, 59, 72, 82, 75, 69, 79, 71, 79, 76, 86, 78])


difference = my_data - example
s = np.std(difference)

t1 = np.mean(difference)/(s*np.sqrt((2/(len(example)+len(my_data)))))
print(abs(t1))
```

```
0.05861801670667292
```

(B) Führen Sie mit Hilfe der SciPy-Implementierung einen t-Test durch um zu bestimmen, ob sich die Alterswerte in Ihrem Datensatz signifikant von denen der spanischen Autoren unterscheiden.

In [11]:
```
import numpy as np
import scipy.stats as stats

print(stats.ttest_ind(a=example, b=my_data, equal_var=True))
```

```
Ttest_indResult(statistic=-0.05853682435948148, pvalue=0.9536277823527071)
```

(C) Formulieren Sie das Ergebnis Ihrer in (B) durchgeführten Untersuchung so, wie Sie es in einem wissenschaftlichen Text (Hausarbeit, Thesis, Fachartikel) beschreiben sollten.

Die Personen der Stichprobe A (example) sind nahezu identisch hinsichtlich der Alterswerte mit der Stichprobe B (eigene Daten, my_data). Es liegt eine Differenz von lediglich 0.25 beim Durchschnitt vor (t-Test: $t40 = 0.058$, $p < 0.05$).