

Лабораторная работа №3

Клейменов Владимир Евгеньевич

БПМ-19-3

Вариант №3

```
In [4]: import numpy as np
import scipy as sc
```

Критерий Крускала-Уоллиса

Вариант №3

H0: Различия между выборками не являются статистически достоверными и носят случайный характер

H1: Различия являются статистически достоверными

```
In [ ]: A = np.array([35.6, 36.0, 34.9, 34.7, 35.8, 37.2, 38.9, 35.8, 36.7, 37.7,
B = np.array([35.7, 36.5, 34.8, 34.5, 35.9, 37.7, 38.8, 35.6, 36.8, 38.3,
```

Посчитаем ранги и проверим их

```
In [27]: def getRanks(arr):
        ranks = sc.stats.rankdata(arr)
        return ranks

ranksA = getRanks(A)
ranksB = getRanks(B)
print("Ranks A : ", ranksA)
print("Ranks B : ", ranksB)
```

```
Ranks A : [ 3.   6.   2.   1.   4.5  9.  13.   4.5  7.  11.  12.   8.
10.  14. ]
Ranks B : [ 4.   6.   2.   1.   5.  10.5 13.   3.   8.  12.  10.5  7.
9.  14. ]
```

```
In [19]: def checkRanks(ranks, a):
        sum = np.sum(ranks)
        N = len(a)
        result = (N * (N + 1)) / 2
        return sum == result

isValidRanksA = checkRanks(ranksA, A)
isValidRanksB = checkRanks(ranksB, B)
print("Ранги A корректны: ", isValidRanksA)
print("Ранги B корректны: ", isValidRanksB)
```

```
Ранги A корректны: True
Ранги B корректны: True
```

Для двух выборок критерий Крускала-Уоллиса по сути сводится к критерию Манна-Уитни

Посчитаем эмпирическое значение U

```
In [22]: def getU(ranksA, ranksB):
lenA = len(ranksA)
lenB = len(ranksB)
sumA = np.sum(ranksA)
sumB = np.sum(ranksB)
maxSum = max(sumA, sumB)
left = lenA * lenB
right = (lenA * (lenA + 1)) / 2
result = left + right - maxSum
return result

U = getU(ranksA, ranksB)
U_table = 55
```

196.0

Сравним эмпирическое значение U с табличным:

```
In [ ]: print("U эмпирическое = ", U)
print("U(14,14) критическое при уровне значимости 0.05 = ", U_table)
```

Так как эмпирическое значение U больше критического ($U > U_{кр}$) при уровне значимости 0.05, то принимается гипотеза H_0 : Различия между выборками не являются статистически достоверными и носят случайный характер

Критерий Уилкоксона

Вариант №3

H_0 : отсутствие статистической значимости изменений показателя H_1 : статистическая значимость изменений показателя в типичную сторону

```
In [23]: A = np.array([35.6, 36.0, 34.9, 34.7, 35.8, 37.2, 38.9, 35.8, 36.7, 37.7,
B = np.array([35.7, 36.5, 34.8, 34.5, 35.9, 37.7, 38.8, 35.6, 36.8, 38.3,
```

Посчитаем разности показателей и проранжируем

```
In [30]: differences = A - B
absDifferences = abs(differences)
print("Differences : ", differences)
print("Absolute differences : ", absDifferences)
diffRanks = getRanks(differences)
print("Ranks : ", diffRanks)

print("Is ranks valid :", checkRanks(diffRanks, differences))
```

```

Differences : [-0.1 -0.5  0.1  0.2 -0.1 -0.5  0.1  0.2 -0.1 -0.6  0.1
0.2 -0.1 -0.5]
Absolute differences : [0.1 0.5 0.1 0.2 0.1 0.5 0.1 0.2 0.1 0.6 0.1 0.
2 0.1 0.5]
Ranks : [ 6.   3.  10.5 14.   6.   3.  10.5 12.5  8.   1.   9.  12.5
6.   3. ]
Is ranks valid : True

```

Пусть, типичным сдвигом будет считаться $A[i] - B[i] < 0$

Выберем индексы рангов соответствующих сдвигам в нетипичном направлении

```

In [32]: def markRanks(diff, negative: bool):
          rankIndices = np.array([])
          for i in range(len(diff)):
              if ((diff[i] < 0) and negative) or (diff[i] > 0) and not negative:
                  rankIndices = np.append(rankIndices, i)
          return rankIndices

indices = markRanks(differences, False)
print("Индексы рангов с нетипичным сдвигом: ", indices)

```

```

Индексы рангов с нетипичным сдвигом: [ 2.  3.  6.  7. 10. 11.]
14

```

Расчитаем Т-критерий Уилкоксона

```

In [35]: def getEmpT(rankIndices, ranks):
          sum = 0
          for i in range(len(rankIndices)):
              sum += ranks[i]
          return sum

T_emp = getEmpT(indices, diffRanks)
T_table = 25

```

Сравним эмпирическое значение Т с табличным $T(14)_{0.05}$

```

In [36]: print("Т эмпирическое : ", T_emp)
          print("T(14)_0.05 : ", T_table)
          print("Тэмп > T(14) = ", T_emp > T_table)

```

```

Т эмпирическое : 42.5
T(14)_0.05 : 25
Тэмп > T(14) = True

```

Так как эмпирическое значение Т больше табличного $T(14)$ принимается нулевая гипотеза об отсутствии статистической значимости изменений показателя.

Задача №8 H_0 : расход топлива остался прежним: $x = 10$ л H_1 : расход топлива уменьшился: $x < 10$ л

Воспользуемся t-критерием Стьюдента для сравнения средних

```

In [37]: a = np.array([9.67, 10.1, 9.9, 9.35, 10.2, 9.45, 10.5, 9.56, 9.96])
          s = 10

```

Расчитаем сумму, выборочное среднее и квадраты отклонений

```
In [45]: lenA = len(a)
sum = np.sum(a)
xArange = sum / lenA
powDiffs = pow(a - xArange, 2)
powDiffsSum = np.sum(powDiffs)

[0.03401975 0.06029753 0.00207531 0.2544642  0.11940864 0.16357531
 0.41674198 0.08669753 0.01114198]
```

Посчитаем выборочную и исправленную дисперсии

```
In [46]: Dv = powDiffsSum / lenA
s2 = (lenA / (lenA - 1)) * Dv
std = pow(s2, 1 / 2)
```

Расчитаем статистику критерия t

```
In [50]: t = ((xArange - s) * pow(lenA, 1 / 2)) / std
t_table = 2.26
print(t)
```

-1.152508774777006

Сравним абсолютное значение t с табличным значением $t(9)_{0.05} = 2,26$

```
In [51]: print("|t| = ", abs(t))
print("t(9)_0.05 = ", t_table)
print("|t| < t(9)_0.05 =", abs(t) < t_table)
```

```
|t| = 1.152508774777006
t(9)_0.05 = 2.26
|t| < t(9)_0.05 = True
```

Так как абсолютное значение t с табличным значением меньше $t(9)_{0.05}$, принимается гипотеза H_0 - расход топлива не изменился