INDIAN INSTITUTE OF TECHNOLOGY KANPUR

MTH516A: Non Parametric Inference

A PROJECT REPORT ON

Non Parametric Tests on Mobile Price Dataset

Submitted by :

Harshit Pandey

Contents

1 Introduction

Non-parametric tests are statistical tests that do not make assumptions about the underlying distribution of the data. Unlike parametric tests, non-parametric tests are not based on specific assumptions about the population, such as normality or equal variances. Non-parametric tests are often used when the data do not meet the assumptions of parametric tests, such as when the data are not normally distributed, or when the sample size is small. Some common non-parametric tests include the Sign test, Wilcoxon signed rank test, Mann-Whitney U test, Kruskal-Wallis test, and the Friedman test. These tests are used to compare two or more groups of data, and to test for differences between groups. Non-parametric tests are widely used in many fields, including biology, psychology, sociology, and economics, among others. They provide a powerful tool for analysing data when parametric assumptions are not met, and are often used in situations where the data are difficult or impossible to measure using traditional statistical methods.

2 Problem statement

In this project, we have applied various non parametric tests for testing location, scale, variability and for checking association between the variables. Along with that, we have also calculated the confidence interval for various parameters.

3 Dataset Description

The dataset is of an anonymous mobile company which sells phones in different price ranges according to their specifications such as RAM, internal memory, camera, battery power, etc.

No. of rows	2000
No. of columns	6
Missing values	NA

Table 1: Dataset descriptions

The description of the columns are as follows:

battery power - Total energy a battery can store in one time measured in mAh

Price range- price range of mobile phones

fc - Front Camera mega pixel

int memory - Internal Memory in Gigabytes

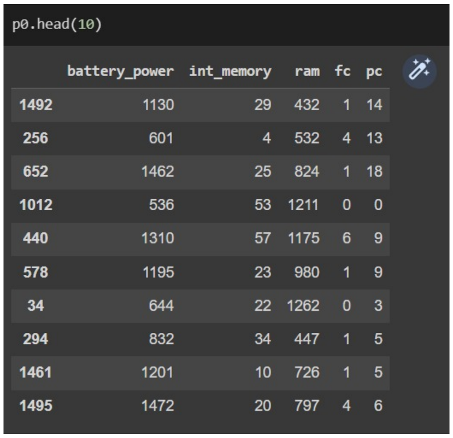
pc - Primary Camera mega pixels

ram - Random Access Memory in Megabytes

In the price column we have four different types of price ranges from zero to three.

We have partitioned the dataset into four different populations consisting of 500 observations each with respect to the price range of mobiles. They are denoted by Price_0, Price_1, Price_2, Price_3.

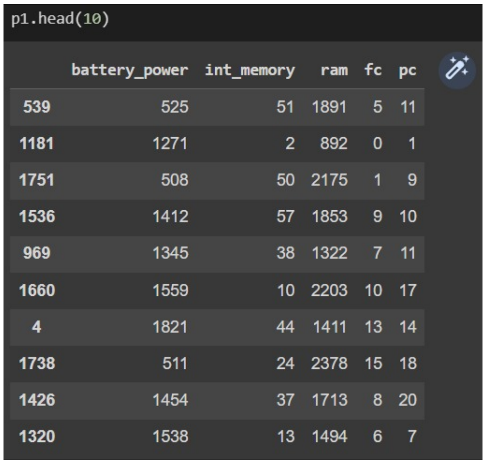
Now we will take samples of size 30 from each of the four populations. The samples are denoted by P_0,P_1,P_2,P_3.



```
p0.head(10)
```

	battery_power	int_memory	ram	fc	pc
1492	1130	29	432	1	14
256	601	4	532	4	13
652	1462	25	824	1	18
1012	536	53	1211	0	0
440	1310	57	1175	6	9
578	1195	23	980	1	9
34	644	22	1262	0	3
294	832	34	447	1	5
1461	1201	10	726	1	5
1495	1472	20	797	4	6

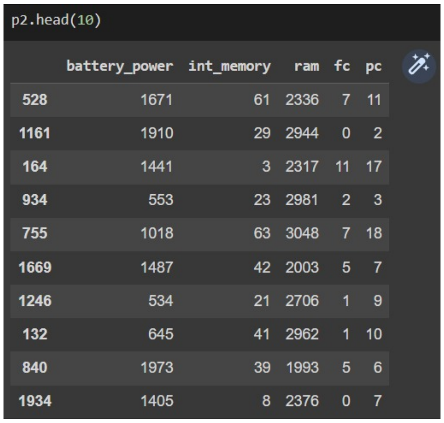
Figure 1: sample of population 1



```
p1.head(10)
```

	battery_power	int_memory	ram	fc	pc
539	525	51	1891	5	11
1181	1271	2	892	0	1
1751	508	50	2175	1	9
1536	1412	57	1853	9	10
969	1345	38	1322	7	11
1660	1559	10	2203	10	17
4	1821	44	1411	13	14
1738	511	24	2378	15	18
1426	1454	37	1713	8	20
1320	1538	13	1494	6	7

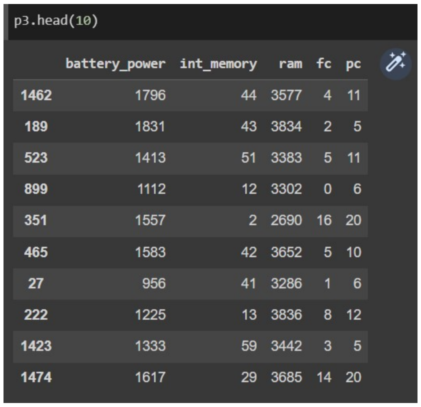
Figure 2: sample of population 2



```
p2.head(10)
```

	battery_power	int_memory	ram	fc	pc
528	1671	61	2336	7	11
1161	1910	29	2944	0	2
164	1441	3	2317	11	17
934	553	23	2981	2	3
755	1018	63	3048	7	18
1669	1487	42	2003	5	7
1246	534	21	2706	1	9
132	645	41	2962	1	10
840	1973	39	1993	5	6
1934	1405	8	2376	0	7

Figure 3: sample of population 3



```
p3.head(10)
```

	battery_power	int_memory	ram	fc	pc
1462	1796	44	3577	4	11
189	1831	43	3834	2	5
523	1413	51	3383	5	11
899	1112	12	3302	0	6
351	1557	2	2690	16	20
465	1583	42	3652	5	10
27	956	41	3286	1	6
222	1225	13	3836	8	12
1423	1333	59	3442	3	5
1474	1617	29	3685	14	20

Figure 4: sample of population 4

4 NON-PARAMETRIC TESTS PERFORMED ON DATASET

4.1 Kruskal Wallis Test

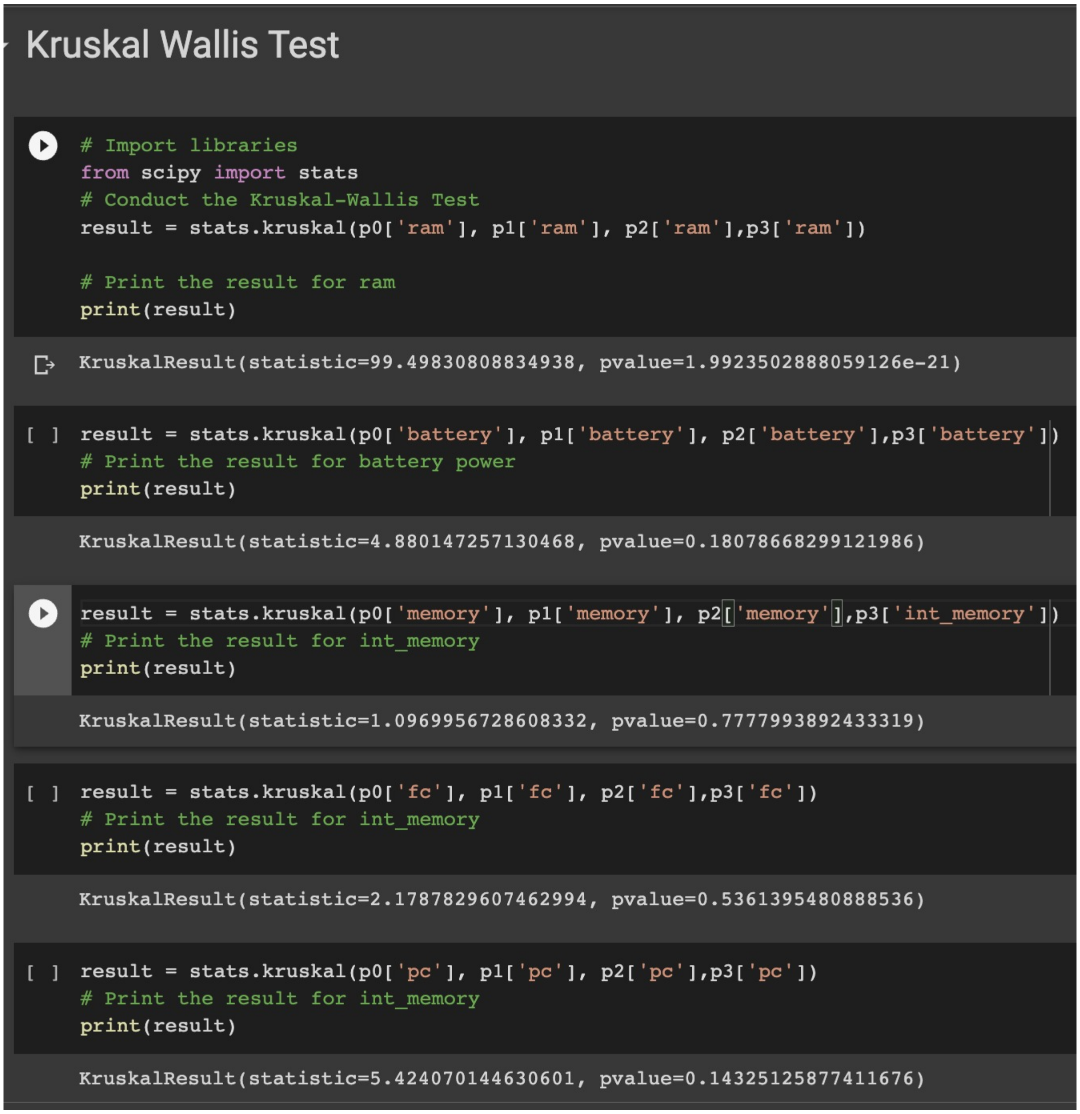
$X_{i1}, X_{i2}, ...X_i n_i$ i.i.d observation from F_i which is absolutely continuous and k sets of independent samples from each population, i=1(1)k

Total no. of observations $\sum_{i=1}^{k} n_i = N$
The null hypothesis is $H_0 : F_1(x) = F_2(x) = ... = F_k(x)\ \forall x$
and the alternative hypothesis is H_A : not equal in any one case

The Kruskal Wallis statistic (H) is

$$\begin{aligned} H &= (N-1)\frac{SSB}{SST} \\ &= \frac{12}{N(N+1)}\sum \frac{R_i^2}{n_i} - 3(N+1) \end{aligned}$$

For our dataset ,we have four different population of price ranges.
H_0 : Distribution of different variables(ram, internal memory, fc, battery power,pc) from each price population(0,1,2,3) are identical.
H_A : At least one is not identical in respect to location parameter.



Kruskal Wallis Test

```
# Import libraries
from scipy import stats
# Conduct the Kruskal-Wallis Test
result = stats.kruskal(p0['ram'], p1['ram'], p2['ram'],p3['ram'])

# Print the result for ram
print(result)
```

```
KruskalResult(statistic=99.49830808834938, pvalue=1.9923502888059126e-21)
```

```
result = stats.kruskal(p0['battery'], p1['battery'], p2['battery'],p3['battery'])
# Print the result for battery power
print(result)
```

```
KruskalResult(statistic=4.880147257130468, pvalue=0.18078668299121986)
```

```
result = stats.kruskal(p0['memory'], p1['memory'], p2['memory'],p3['int_memory'])
# Print the result for int_memory
print(result)
```

```
KruskalResult(statistic=1.0969956728608332, pvalue=0.7777993892433319)
```

```
result = stats.kruskal(p0['fc'], p1['fc'], p2['fc'],p3['fc'])
# Print the result for int_memory
print(result)
```

```
KruskalResult(statistic=2.1787829607462994, pvalue=0.5361395480888536)
```

```
result = stats.kruskal(p0['pc'], p1['pc'], p2['pc'],p3['pc'])
# Print the result for int_memory
print(result)
```

```
KruskalResult(statistic=5.424070144630601, pvalue=0.14325125877411676)
```

Figure 5: Kruskal Wallis test for checking population of various variables

Here we can see for ram the H_0 is getting rejected .

Hence, RAM has significant difference with respect to different prices.

Hence it is safe to assume that it is the most important regressors for the classification of different price ranges.

4.2 Two Sample K-S Test

The two-sample Kolmogorov-Smirnov (KS) test is a nonparametric statistical test used to compare two samples or datasets to determine whether they come from the same underlying distribution or not. The KS test is sensitive to differences in both location and shape of the distributions being compared.

The null hypothesis of the two-sample KS test is that the two samples come from the same underlying distribution. The alternative hypothesis is that the two samples come from different underlying distributions.

The test statistic for the two-sample KS test is the maximum difference between the cumulative distribution functions (CDFs) of the two samples.

The kolmogorov smirnov test is a goodness of fit test. It is used to compare empirical distribution function of a random sample with a hypothesized cumulative distribution , in the two sample case ,the comparision is made between the comparision is made between the empirical distribution functions of the two samples.

The order statistics corresponding to two random samples of size m and n from continuous populations F_x and F_Y, are $X_1, X_2, ..., X_m$ and $Y_1, Y_2, ...Y_n$

Their respective empirical distribution functions, denoted by $S_m(x)$ and $S_n(x)$ are defined as follows

$$S_m(x) = \begin{cases} 0, & \text{if } x < X_1 \\ k/m, & \text{if } X_k \leq x < X_{k+1} \\ 1, & \text{if } x \geq X_m \end{cases},$$

$$S_n(x) = \begin{cases} 0, & \text{if } y < Y_1 \\ k/n, & \text{if } Y_k \leq y < Y_{k+1} \\ 1, & \text{if } y \geq Y_n \end{cases},$$

The two sided K-S two sample test criterion, denoted by $D_{m,n}$ is based on maximum absolute difference between the two empirical distributions

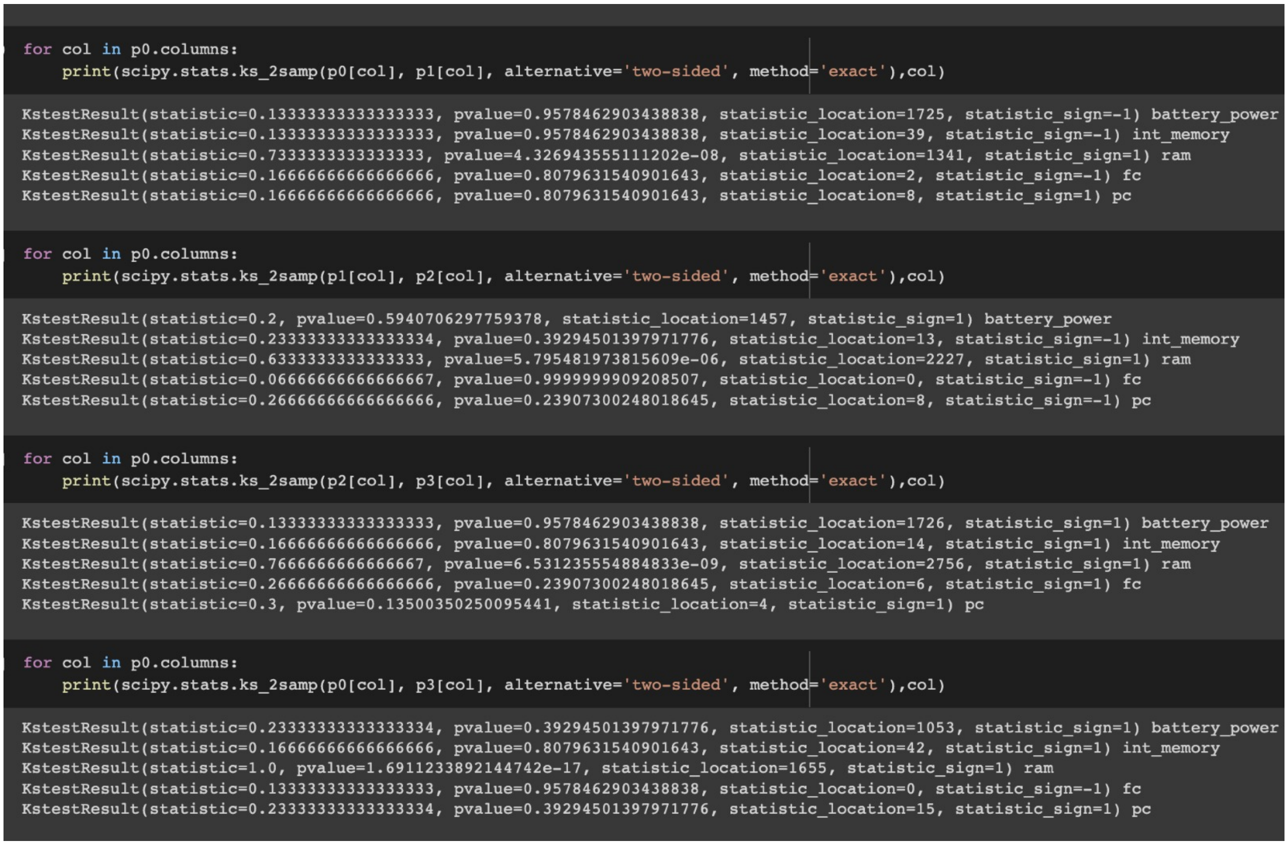
$D_{m,n} = \underset{x}{max} |S_m(x) - S_n(x)|$

The hypothesis to be tested is

$H_0 : F_Y(x) = F_X(x)$

against the null hypothesis
$H_A : F_Y(x) \neq F_X(x)$ for some x
If the alternative hypothesis is upper tailed, it is defined by
$D_{m,n} \geq c_\alpha$

Here, We test the equality of two populations for all the variables.



```
for col in p0.columns:
    print(scipy.stats.ks_2samp(p0[col], p1[col], alternative='two-sided', method='exact'),col)
```

```
KstestResult(statistic=0.13333333333333333, pvalue=0.9578462903438838, statistic_location=1725, statistic_sign=-1) battery_power
KstestResult(statistic=0.13333333333333333, pvalue=0.9578462903438838, statistic_location=39, statistic_sign=-1) int_memory
KstestResult(statistic=0.7333333333333333, pvalue=4.326943555111202e-08, statistic_location=1341, statistic_sign=1) ram
KstestResult(statistic=0.16666666666666666, pvalue=0.8079631540901643, statistic_location=2, statistic_sign=-1) fc
KstestResult(statistic=0.16666666666666666, pvalue=0.8079631540901643, statistic_location=8, statistic_sign=1) pc
```

```
for col in p0.columns:
    print(scipy.stats.ks_2samp(p1[col], p2[col], alternative='two-sided', method='exact'),col)
```

```
KstestResult(statistic=0.2, pvalue=0.5940706297759378, statistic_location=1457, statistic_sign=1) battery_power
KstestResult(statistic=0.23333333333333334, pvalue=0.39294501397971776, statistic_location=13, statistic_sign=-1) int_memory
KstestResult(statistic=0.6333333333333333, pvalue=5.795481973815609e-06, statistic_location=2227, statistic_sign=1) ram
KstestResult(statistic=0.06666666666666667, pvalue=0.9999999909208507, statistic_location=0, statistic_sign=-1) fc
KstestResult(statistic=0.26666666666666666, pvalue=0.23907300248018645, statistic_location=8, statistic_sign=-1) pc
```

```
for col in p0.columns:
    print(scipy.stats.ks_2samp(p2[col], p3[col], alternative='two-sided', method='exact'),col)
```

```
KstestResult(statistic=0.13333333333333333, pvalue=0.9578462903438838, statistic_location=1726, statistic_sign=1) battery_power
KstestResult(statistic=0.16666666666666666, pvalue=0.8079631540901643, statistic_location=14, statistic_sign=1) int_memory
KstestResult(statistic=0.7666666666666667, pvalue=6.531235554884833e-09, statistic_location=2756, statistic_sign=1) ram
KstestResult(statistic=0.26666666666666666, pvalue=0.23907300248018645, statistic_location=6, statistic_sign=1) fc
KstestResult(statistic=0.3, pvalue=0.13500350250095441, statistic_location=4, statistic_sign=1) pc
```

```
for col in p0.columns:
    print(scipy.stats.ks_2samp(p0[col], p3[col], alternative='two-sided', method='exact'),col)
```

```
KstestResult(statistic=0.23333333333333334, pvalue=0.39294501397971776, statistic_location=1053, statistic_sign=1) battery_power
KstestResult(statistic=0.16666666666666666, pvalue=0.8079631540901643, statistic_location=42, statistic_sign=1) int_memory
KstestResult(statistic=1.0, pvalue=1.6911233892144742e-17, statistic_location=1655, statistic_sign=1) ram
KstestResult(statistic=0.13333333333333333, pvalue=0.9578462903438838, statistic_location=0, statistic_sign=-1) fc
KstestResult(statistic=0.23333333333333334, pvalue=0.39294501397971776, statistic_location=15, statistic_sign=1) pc
```

Figure 6: two sample Kolmogorov Smirnov test

In the above output statistic_location: Value from data1 or data2 corresponding with the KS statistic; i.e., the distance between the empirical distribution functions is measured at this observation.

statistic_sign : +1 if the empirical distribution function of data1 exceeds the empirical distribution function of data2 at statistic_location, otherwise -1.

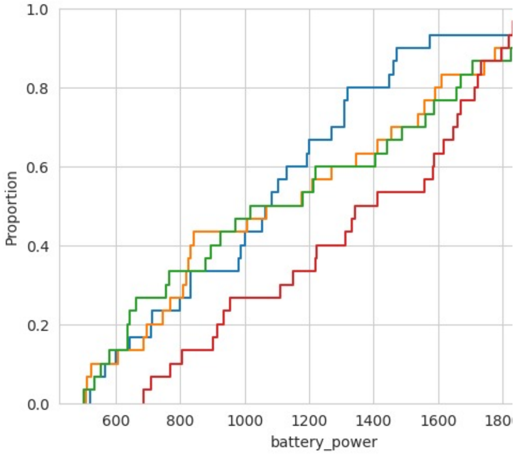


Figure 7: ecdf of battery power

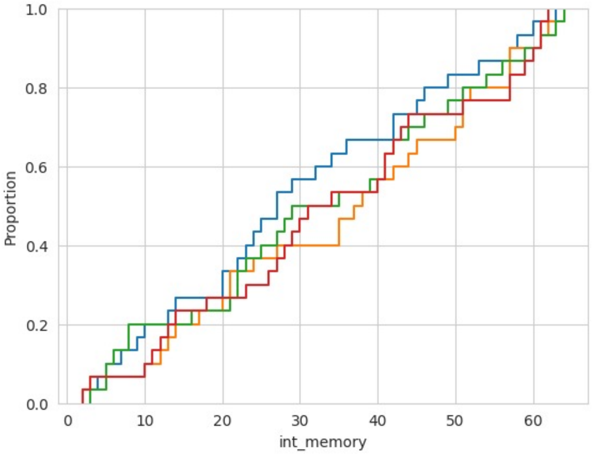


Figure 8: ecdf of internal memory

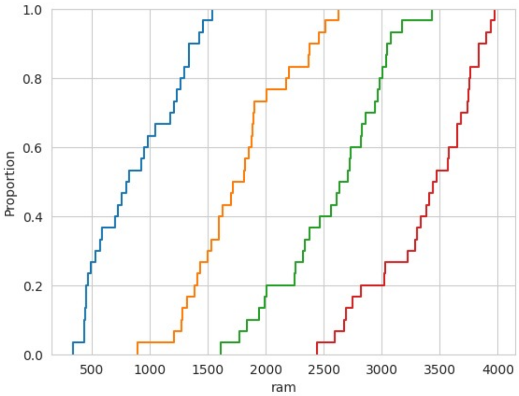


Figure 9: ecdf of ram

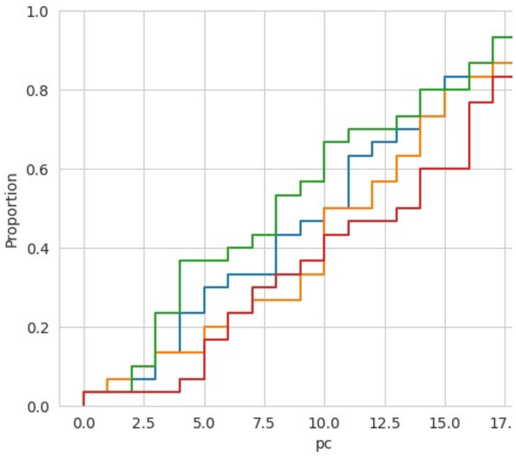


Figure 10: ecdf of primary camera

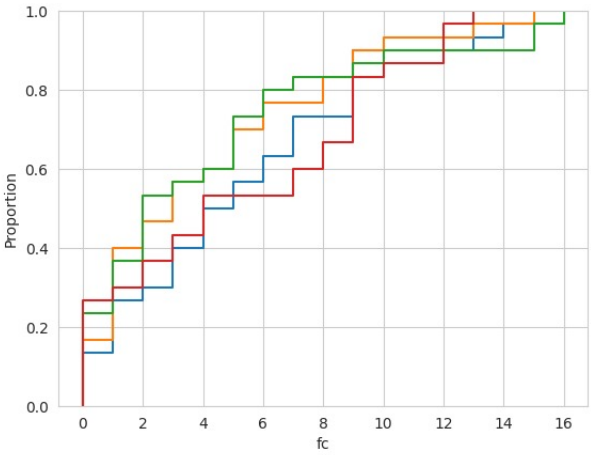


Figure 11: ecdf of front camera

From the K-S test and the ECDF we can observe that the location parameter of different population of ram is significantly different and $ram[pop0] < ram[pop1] < ram[pop2] < ram[pop3]$.

To check if there is any association between regressors

4.3 Kendall's Tau test

The Kendall rank correlation coefficient, also known as Kendall's tau, is a statistical measure that assesses the ordinal association between two variables. It measures the similarity of the orderings of the data when ranked by each of the two variables.

The Kendall tau coefficient is calculated by comparing the number of concordant pairs and discordant pairs in the data. A concordant pair is a pair of observations that have the same order for both variables, while a discordant pair is a pair of observations that have opposite order for the two variables.

The formula for Kendall tau is:

$T = (number of concordant pairs - number of discordant pairs)/n(n-1)/2)$

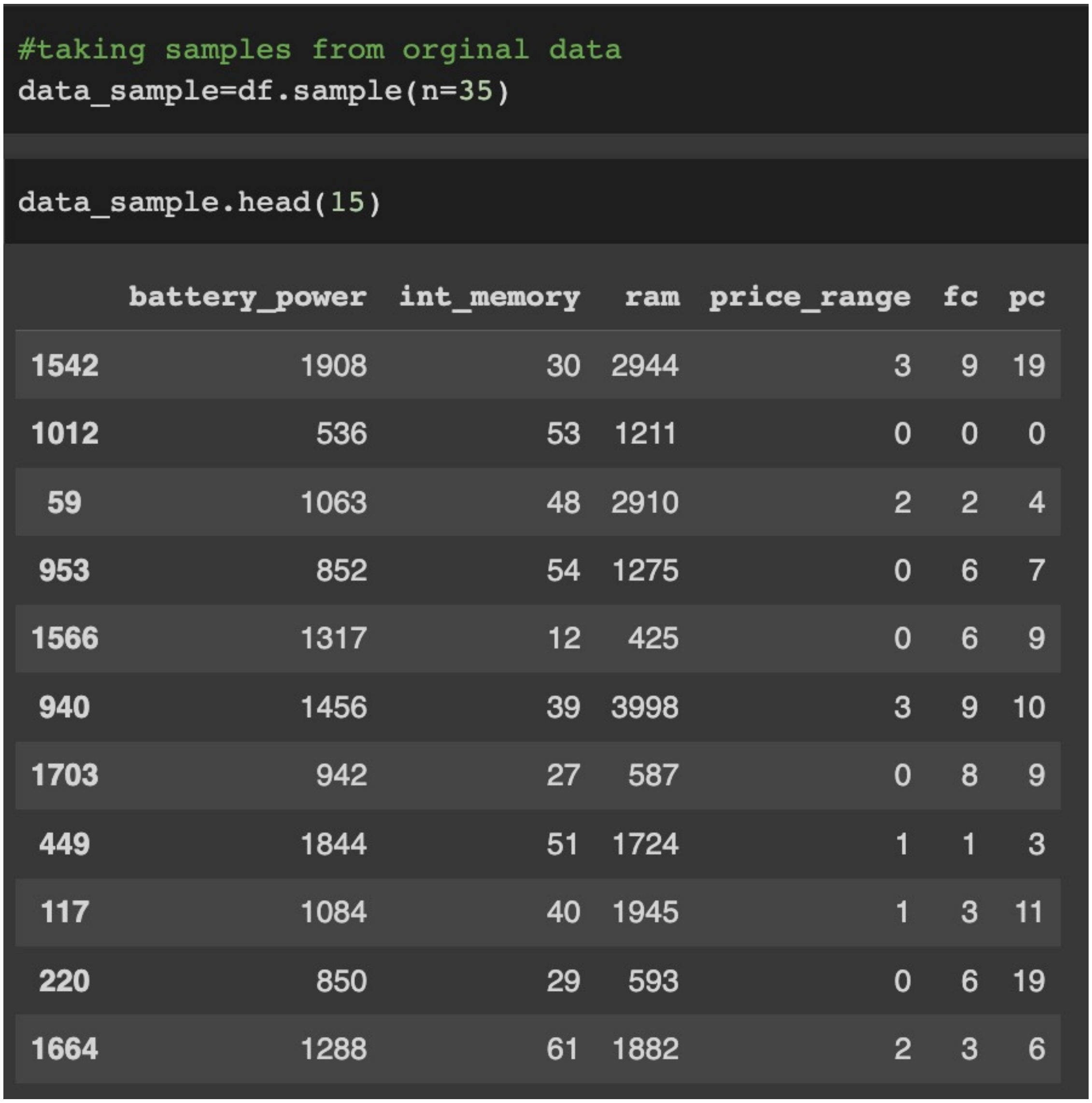
where n is the number of observations.

Kendall tau can range from -1 to 1, where -1 indicates a perfect negative association (i.e., as one variable increases, the other decreases), 0 indicates no association, and 1 indicates a perfect positive association (i.e., as one variable increases, the other also increases).

The null hypothesis $H_0 : F_{X,Y}(x, y) = F_X(x) \cdot F_Y(y)$

and the alternate hypothesis $H_A : F_{X,Y}(x, y) \neq F_X(x) \cdot F_Y(y)$

We took 35 samples from original regressors population battery_power , int_mem, ram , fc , pc then calculated the Kendall tau sample correlation coefficient for different regressors population.

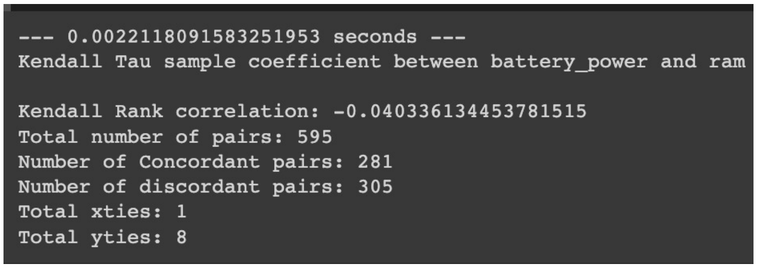


```
#taking samples from orginal data
data_sample=df.sample(n=35)
```

```
data_sample.head(15)
```

	battery_power	int_memory	ram	price_range	fc	pc
1542	1908	30	2944	3	9	19
1012	536	53	1211	0	0	0
59	1063	48	2910	2	2	4
953	852	54	1275	0	6	7
1566	1317	12	425	0	6	9
940	1456	39	3998	3	9	10
1703	942	27	587	0	8	9
449	1844	51	1724	1	1	3
117	1084	40	1945	1	3	11
220	850	29	593	0	6	19
1664	1288	61	1882	2	3	6

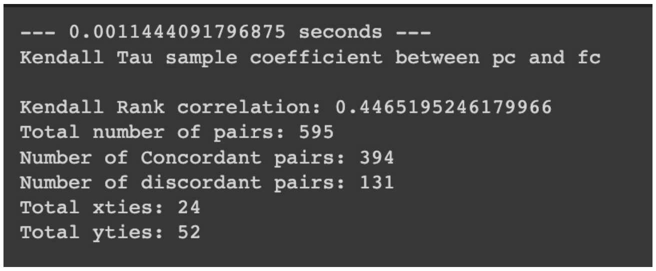
Figure 12: sample data



```
--- 0.0022118091583251953 seconds ---
Kendall Tau sample coefficient between battery_power and ram

Kendall Rank correlation: -0.040336134453781515
Total number of pairs: 595
Number of Concordant pairs: 281
Number of discordant pairs: 305
Total xties: 1
Total yties: 8
```

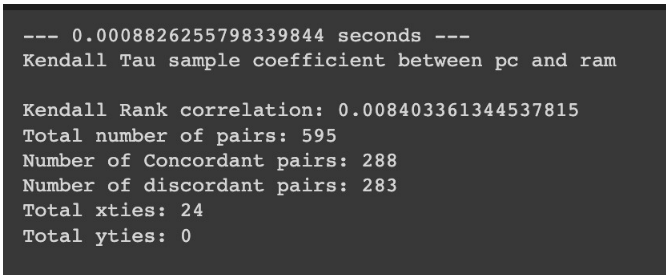
Figure 13: kendall tau sample corr coeff between ram and battery power



```
--- 0.0011444091796875 seconds ---
Kendall Tau sample coefficient between pc and fc

Kendall Rank correlation: 0.4465195246179966
Total number of pairs: 595
Number of Concordant pairs: 394
Number of discordant pairs: 131
Total xties: 24
Total yties: 52
```

Figure 14: kendall tau sample corr coeff between pc and fc



```
--- 0.0008826255798339844 seconds ---
Kendall Tau sample coefficient between pc and ram

Kendall Rank correlation: 0.008403361344537815
Total number of pairs: 595
Number of Concordant pairs: 288
Number of discordant pairs: 283
Total xties: 24
Total yties: 0
```

Figure 15: kendall tau sample corr coeff between ram and pc

4.4 Spearman's Rank correlation coefficient

Spearman's rank correlation coefficient is a statistical measure used to assess the strength and direction of the relationship between two variables. It is a non-parametric measure, which means that it does not make any assumptions about the distribution of the variables.

Spearman's rank correlation coefficient is calculated by first ranking the values of each variable separately, from smallest to largest. Then, the difference between the ranks of each pair of corresponding values is calculated. The correlation coefficient is then calculated as the ratio of the covariance of the ranks to the product of the standard deviations of the ranks.

$(X_1, Y_1), (X_2, Y_2), \ldots\ldots\ldots\ldots\ldots\ldots, (X_n, Y_n)$ are paired observations from some unknown distribution with CDF $F_{x,y}$.

Define $R_i = rank(X_i)$ and $S_i = rank(Y_i)$ Then, the Spearman's Rank Correlation coefficient is given by

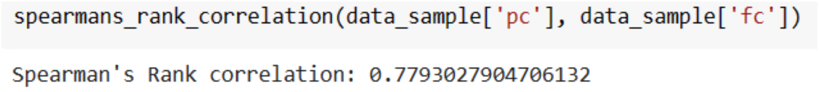
$$R = \frac{12\sum_{i=1}^{n}(R_i - \bar{R})(S_i - \bar{S})}{n(n^2-1)}$$

$$R = 1 - \frac{6\sum_{i=1}^{n} D_i^2}{n(n^2-1)}$$

where $D_i = R_i - S_i$
and $-1 \leq R \leq 1$

The null hypothesis $H_0 : F_{X,Y}(x, y) = F_X(x) \cdot F_Y(y)$

In our case we want to test whether there is any association between primary camera and front camera.



```
spearmans_rank_correlation(data_sample['pc'], data_sample['fc'])
```

```
Spearman's Rank correlation: 0.7793027904706132
```

Figure 16: Spearman's rank correlation coefficient test

As we can observe from the above figure that the R=0.779 which indicates front camera and primary camera are positively correlated

To check dispersion in ram and battery power in different price ranges

4.5 Mood Test

Moods Linear Rank Test is a non-parametric statistical test used to compare two independent samples of ordinal data. It is useful when the data is not normally distributed or the sample size is small. The test is based on the ranks of the observations in the combined sample, and the test result is reported as a p-value.

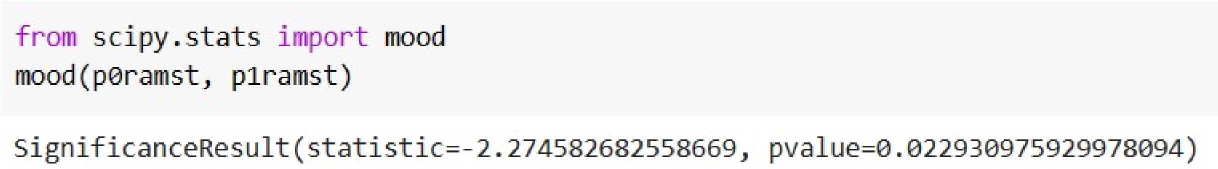
Large weights in extreme.

$$M_N = \sum_{i=1}^{N} \left(i - \frac{N+1}{2} \right)^2 Z_i$$

The null and alternative hypothesis is

$H_0 : F_Y(x) = F_X(x)$

$H_A : F_Y(x) = F_X(\theta x)$ where $\theta \neq 1$

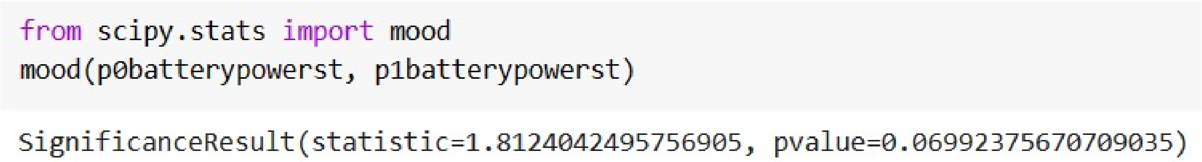


```
from scipy.stats import mood
mood(p0ramst, p1ramst)
```

```
SignificanceResult(statistic=-2.274582682558669, pvalue=0.022930975929978094)
```

Figure 17: Mood's test for checking the scale shift in the ram

as we can observe that the p-value is less than 0.05 hence reject our null hypothesis that ram has same dispersion for price range 0 and price range 1.



```
from scipy.stats import mood
mood(p0batterypowerst, p1batterypowerst)
```

```
SignificanceResult(statistic=1.8124042495756905, pvalue=0.06992375670709035)
```

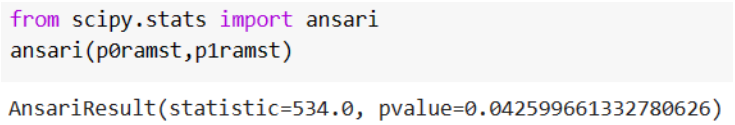
Figure 18: Mood's test for checking the scale shift in the battery power

We fail to reject the null hypotheses that battery power has same dispersion for price range 0 and price range 1

4.6 Ansari test

The test is based on the ranks of the observations in each group. The test statistic is calculated as the difference between the sums of the absolute differences between the ranks of the two samples, divided by a scaling factor. The scaling factor depends on the sample size and the number of ties in the data. Large weights at middle

$$F_N = \sum_{i=1}^{N} \left(\frac{N+1}{2} - \left| i - \frac{N+1}{2} \right| \right) Z_i$$

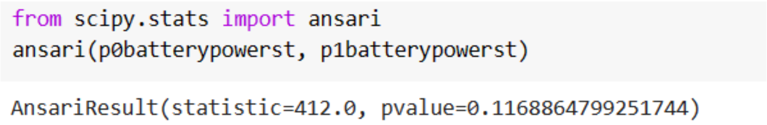


```
from scipy.stats import ansari
ansari(p0ramst,p1ramst)
```

```
AnsariResult(statistic=534.0, pvalue=0.042599661332780626)
```

Figure 19: Ansari test for checking the scale shift in the ram

as we can observe that the p-value is less than 0.05 hence reject our null hypothesis that ram has same dispersion for price range 0 and price range 1.



```
from scipy.stats import ansari
ansari(p0batterypowerst, p1batterypowerst)
```

```
AnsariResult(statistic=412.0, pvalue=0.1168864799251744)
```

Figure 20: Ansari test for checking the scale shift in the battery power

Fail to reject the null hypotheses that battery power has same dispersion for price range 0 and price range.

4.7 SIGN TEST :

A random sample of size N i.e., $X_1, X_2, \ldots, X_N$ is drawn from a population F_X with an unknown median M, where F_X is assumed to be continuous and strictly increasing, at least in the vicinity of M. The hypothesis to be tested concerns the value of the population median

$$H_0 : M = M_0$$

where M_0 is a specified value, against a corresponding one or two sided alternative.

The number of observations larger than M_0, denoted by K, can be used to test the validity of the null hypothesis.

$$K = \sum_{i=1}^{n} T_i,$$

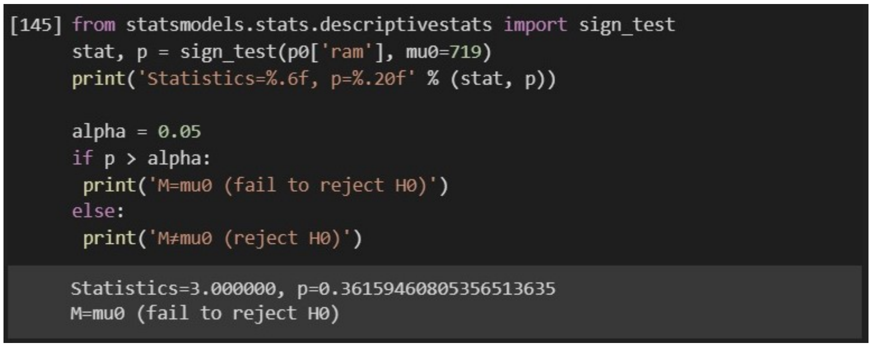
where $T_i \sim ber(1/2)$ and

$$K \sim bin(N, 1/2)$$

If the alternative is two-sided,

$H_1 : M \neq M_0$ and the rejection region is $K \geq k_{\alpha/2}$ or $K \leq k'_{\alpha/2}$

We will test whether the median is equal to a specific value or not for different variables.



```
[145] from statsmodels.stats.descriptivestats import sign_test
      stat, p = sign_test(p0['ram'], mu0=719)
      print('Statistics=%.6f, p=%.20f' % (stat, p))

      alpha = 0.05
      if p > alpha:
       print('M=mu0 (fail to reject H0)')
      else:
       print('M≠mu0 (reject H0)')
```

```
      Statistics=3.000000, p=0.36159460805356513635
      M=mu0 (fail to reject H0)
```

Figure 21: Sign test for checking median

Hence we fail to reject the null hypothesis using sign test that median of the ram for price range 0 mobiles is 719.

4.8 Wilcoxon Signed Rank Test

A random sample of size N i.e., $X_1, X_2, \ldots, X_N$ is drawn from a population F_X with an unknown median M, where F_X is assumed to be continuous and strictly increasing, at least in the vicinity of M.

The hypothesis to be tested concerns the value of the population median

$$H_0 : M = M_0$$

where M_0 is a specified value, against a corresponding one or two sided alternative.

The sum of all positively signed ranks, denoted by T^+ can be used to test the validity of the null hypothesis.

$$T^+ = \sum_{i=1}^{n} \Psi_i R_i, \text{ where}$$

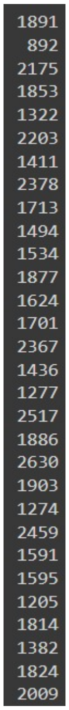
$$\Psi_i = \begin{cases} 1, & \text{if } z_i > 0 \\ 0, & \text{if } z_i \leq 0 \end{cases},$$

$$Z_i = X_i - M_0$$

$$R_i = Rank(|Z_i|)$$

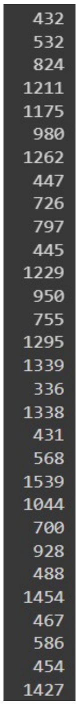
If the alternative is two-sided,
$H_1 : M \neq M_0$ and the rejection region is $T^+ \geq t_{\alpha/2}$ or $T^+ \leq t'_{\alpha/2}$

Now, we will test if there is any significant difference between different variables among two distinct populations.



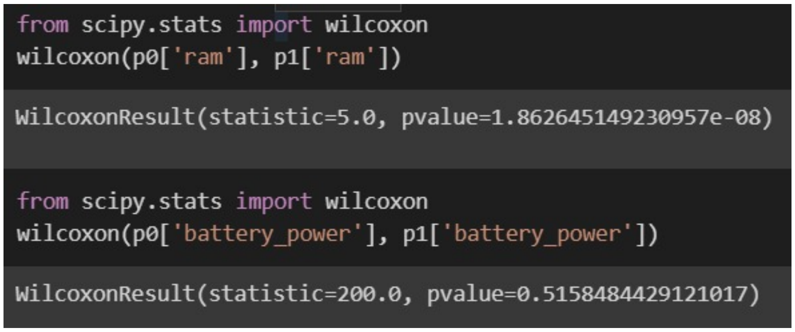
```
1891
 892
2175
1853
1322
2203
1411
2378
1713
1494
1534
1877
1624
1701
2367
1436
1277
2517
1886
2630
1903
1274
2459
1591
1595
1205
1814
1382
1824
2009
```

Figure 22: ram of p_1



```
 432
 532
 824
1211
1175
 980
1262
 447
 726
 797
 445
1229
 950
 755
1295
1339
 336
1338
 431
 568
1539
1044
 700
 928
 488
1454
 467
 586
 454
1427
```

Figure 23: ram of p_0



```
from scipy.stats import wilcoxon
wilcoxon(p0['ram'], p1['ram'])
```

```
WilcoxonResult(statistic=5.0, pvalue=1.862645149230957e-08)
```

```
from scipy.stats import wilcoxon
wilcoxon(p0['battery_power'], p1['battery_power'])
```

```
WilcoxonResult(statistic=200.0, pvalue=0.5158484429121017)
```

Figure 24: Wilcoxon signed rank test for checking median

We can interpret from Wilcoxon signed rank test that only the Ram variable differs significanlty in population P_0 and P_1. Also battery power differs significantly in population P_0 and P_3.

4.9 Mann Whitney U test

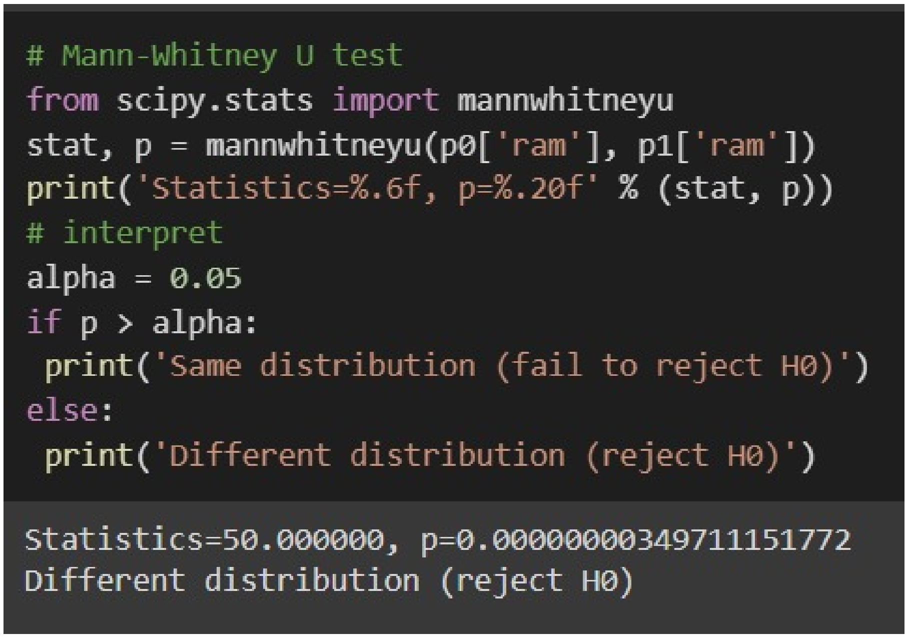
The Mann Whitney U test statistic is defined as the number of times a Y precedes a X in combined ordered arrangement of two independent populations $X_1, X_2, ...X_m \; and \; Y_1, Y_2, ...Y_n$

The hypothesis to be tested is

$$H_0 : F_Y(x) = F_X(x)$$

against the alternative hypothesis

$H_A : F_Y(x) \neq F_X(x)$ and the rejection region is $U \geq u_{\alpha/2}$ or $U \leq u'_{\alpha/2}$



```
# Mann-Whitney U test
from scipy.stats import mannwhitneyu
stat, p = mannwhitneyu(p0['ram'], p1['ram'])
print('Statistics=%.6f, p=%.20f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
 print('Same distribution (fail to reject H0)')
else:
 print('Different distribution (reject H0)')
```

```
Statistics=50.000000, p=0.00000000349711151772
Different distribution (reject H0)
```

Figure 25: Mann Whitney U test for checking median

We can interpret from Mann Whitney U test that only the distribution of the Ram variable differs significantly in population P_0 and P_1. Also the distribution of battery power differs significantly in population P_0 and P_3