Exercise 1

The permutation test is a non-parametric method to test the null hypothesis.
By shuffling and recalculating the chi-squared statistic, we create a distribution of the statistic under the null hypothesis.

In the tables provided we have the joint probability distribution of two (discrete) random variables X and Y. What each table shows is the probability of each possible pair of values for X and Y simultaneously. The joint probability is given by: **P(X = x, Y = y)**.
The tables can be interpreted as follows, using table 1 as a reference:

- There is a 5% probability that both X=0 and Y=0 at the same time.
  P(X = 0, Y = 0) = 0.05
- There is a 10% probability that both X=0 and Y=1 at the same time.
  P(X = 0, Y = 1) = 0.1
- There is a 25% probability that both X=1 and Y=0 at the same time.
  P(X = 1, Y = 0) = 0.25
- There is a 60% probability that both X=1 and Y=1 at the same time.
  P(X = 1, Y = 1) = 0.6

Since the above probabilities represent all possible outcomes for X and Y, the summation of these probabilities will equal to 1. The rest of the tables can be inferred in the same manner.

To check for independence of a random variable, we will use a statistical test (in our case chi-squared independence test). For that, we need a null hypothesis (**H0**) and an alternative hypothesis (**H1**).

The H0 in our case is that X is independent of Y, for alpha = 0.05.
X and Y are independent if the joint probability equals the product of the individual probabilities for all values of x and y.
The above can be written as **H0: P(X = x, Y = y) = P(X = x)*P(Y = y)** (for all x,y values)

Our alternative hypothesis H1 is that there is some dependence between the variables. This means that the outcome of one variable influences the outcome of the other and happens when the product of the individual probabilities does not equal the joint probability.
The above can be written as **H1: P(X = x, Y = y) not equal to P(X = x)*P(Y = y)** (for some x,y values).
Note that for dependence we don't need the condition to be true for all x,y values.

We will sample values for X and Y using different sample sizes, in our case n = [25, 100, 500]. The possible outcomes given X and Y are discrete random variables taking values {0,1} are: (X=0, Y=0), (X=0, Y=1), (X=1, Y=0), (X=1, Y=1) or as tuples: (0,0), (0,1), (1,0), (1,1).

First we will create the possible outcomes and then using the probabilities as provided in each table, we will randomly sample (for each table) with said probability. This can be done using the function ***np.random.choice(a, size, p)*** and setting ***p*** to correspond to the probabilities of encountering a certain combination. For table 1 it will be an array [0.05, 0.1, 0.25, 0.6].

After doing so, we will perform the chi-squared independence test, using the formula for the statistic **T:**
T = Σ_{i,j} [(O_{i,j} - E_{i,j})^2 / E_{i,j}] (i,j: values of X and Y, in our case {0,1})

- **O_{i,j}**: Observed frequencies for x and y values (as generated from our sampling)
- **E_{i,j}**: Expected frequencies for x and y values (considering H0 is true, so X and Y are independent)

To make a decision if we will reject or accept the null hypothesis, we need a threshold, which is usually indicated by ***α*** which we compare with our p-value. We will be using ***α = 0.05***.
The p-value is given by : **p-value = 1 − P (T ≤ t obs)**
In general, the p-value is the probability of obtaining a test statistic at least as extreme as the one observed, assuming the null hypothesis is true. This means very low p-values indicate a very low probability, thus the comparison to a threshold (***α***).
We then compare p-value to ***α***, and if **p-value < α** then we can say we probably reject the null hypothesis. We are never certain of the outcome, so we need to specifically say that we probably reject or probably accept the null hypothesis.

In the script, when calculating expected values, it is common practice to add a very small number (chosen value is 1e-10) to avoid having expected values equal to zero, which will lead to division by zero issues when calculating the **T** statistic.

For the purpose of **Bi)**, we create a dictionary to store information about the **T** statistic, p-value, observed and expected frequencies, while samples can be included as well by removing the comment from **samples** during dictionary appending, showing the results for **A)**.

By analyzing the results, we notice a consistency at rejecting the null hypothesis when sample size increases for Tables 4,5 and 6. This is true in some cases for smaller sample sizes, but not always, which is normal when the sample size is not big enough to capture the variability. This means that there is evidence suggesting a dependency between X and Y for Tables 4,5 and 6 (we reject the null hypothesis that X and Y are independent).
Note that we do encounter a few cases of p-value = 0, due to python's floating-point arithmetic precision but it just indicates a very low p-value and not that we are completely certain of an outcome.

B.ii)

Permutation test is a non-parametric method to test the null hypothesis. It shuffles the data and recalculates the chi-squared statistic, creating a distribution of the statistic under the null hypothesis.

Steps for permutation:

1) Randomly shuffle the observed frequencies (This is done in case there is some potential dependency between variables)
2) Recalculate chi-squared statistic (**T**) for each shuffled observed dataset (using the previous formula, where instead of **O_{i,j}** we have **O_shuffled_{i,j}**
3) Compare the new statistic (**T'**) with the previously calculated **T**. We use a counter, where if **T' > T** we add 1 to the counter.
4) Repeat as many times as the permutation loops we initially chose (in our case 200)
5) Calculate the p-value as : (Number of times (**T'>T**) + 1) / (Total permutations +1)

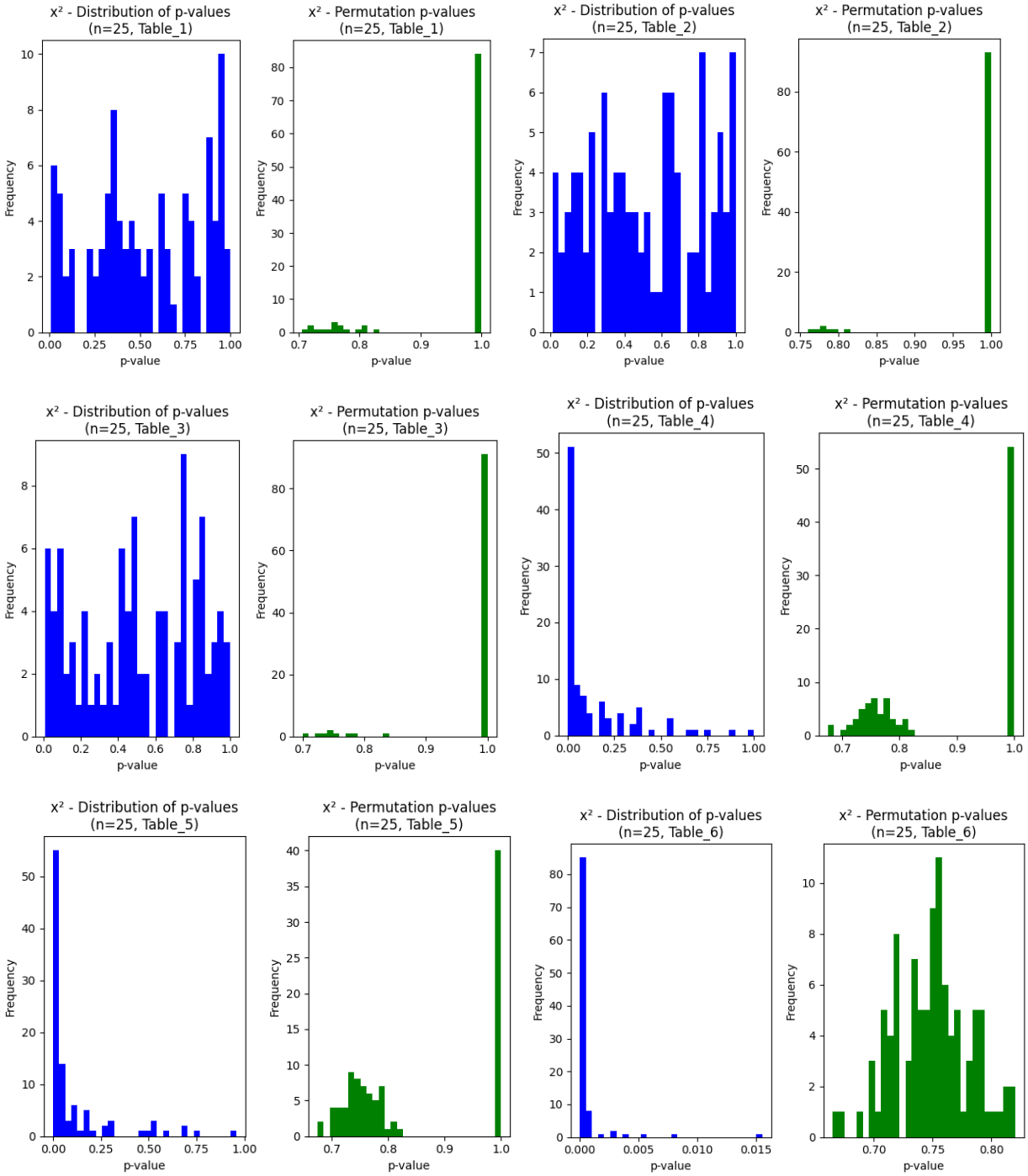
The reason we used Permutations - 1 as the initial loop counter is to make the division more efficient (200 instead of 201).

Note: We add +1 to the numerator and denominator to ensure that the p-value never reaches absolute zero (suggesting complete certainty) and provide a more conservative estimate.

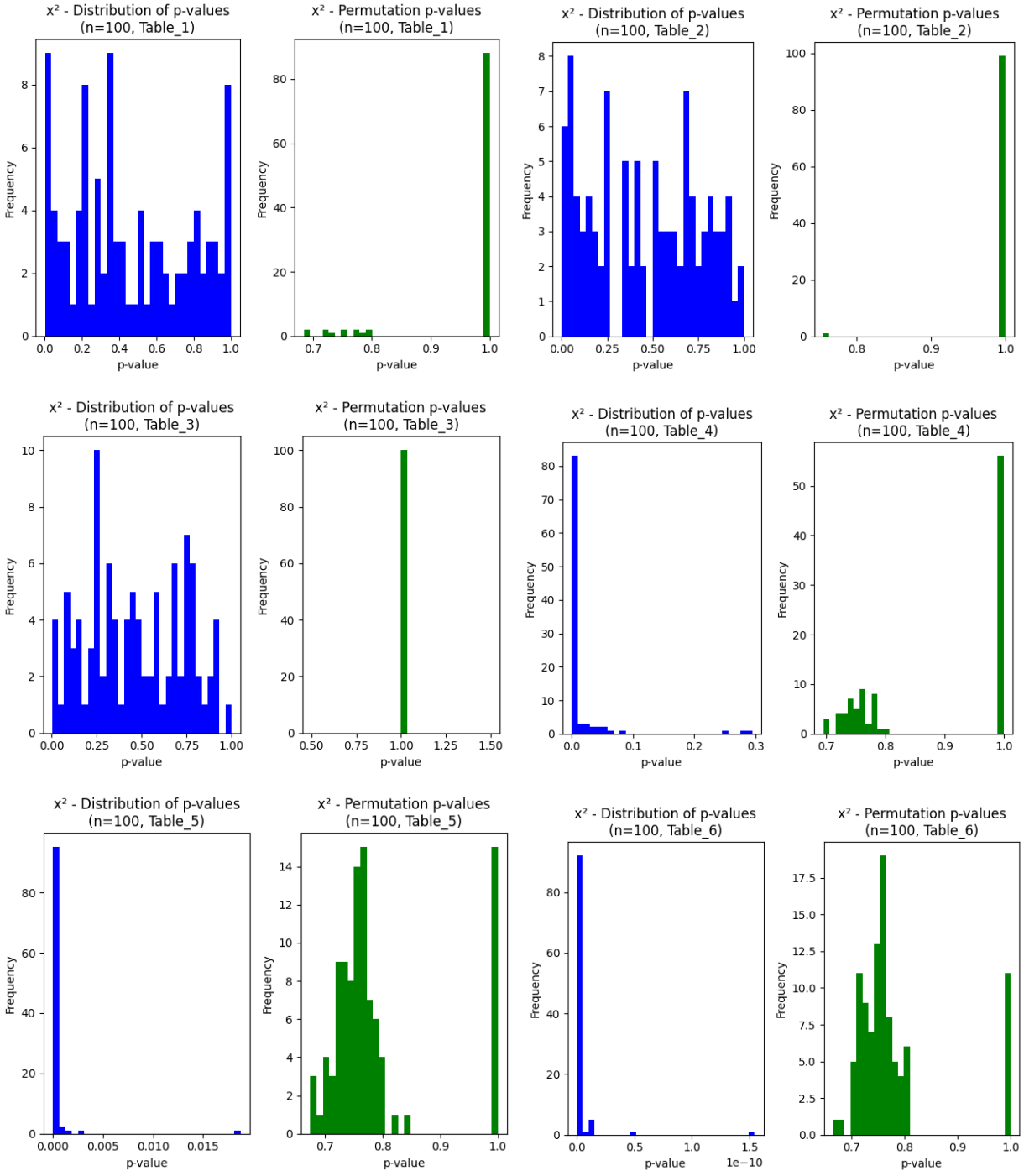
For each sample size and then for each table, we will calculate a chi-squared statistic **T** and use that to compare with the permutation statistic **T'**. We will do this process 100 times as indicated in the exercise, creating a list of p-values that come from 100 different re-samplings. This will provide us with a distribution accounting up to a point for the randomness. This means higher re-samplings, higher permutations and higher samples will provide more precise conclusions. We can then plot the distributions and examine the consistency of the outcomes.
Histograms for all samples sizes and all tables are shown below. On the left are the p-values for the normal chi-squared independence test, while on the right the results of the permutation test (using the p-values of the chi-squared test).

Looking at the non-permutation histograms, we notice an inconsistency concerning the p-values for Tables 1-3 for all sample sizes, meaning we cannot be certain of whether there is dependence between the two variables. On the other hand we notice a consistent pattern for Tables 4-6, a p-value << 0.05 (values close to 0), suggesting that we have strong evidence to reject the null hypothesis. This is more concrete when increasing the sample size, but still true for n=25. When looking at each respective permutation histogram, we notice that not a single p-value is below 0.6 (mostly clustering close to 0.7 or 1). This is in total disagreement to their respective non-permutation p-values, suggesting permutation might not be a proper test given our data is discrete, meaning there are not a lot of possible ways to rearrange the dataset given our 4 different possible outcomes. This lack of variability could explain why the permutation test is not in accordance with the chi-squared test.

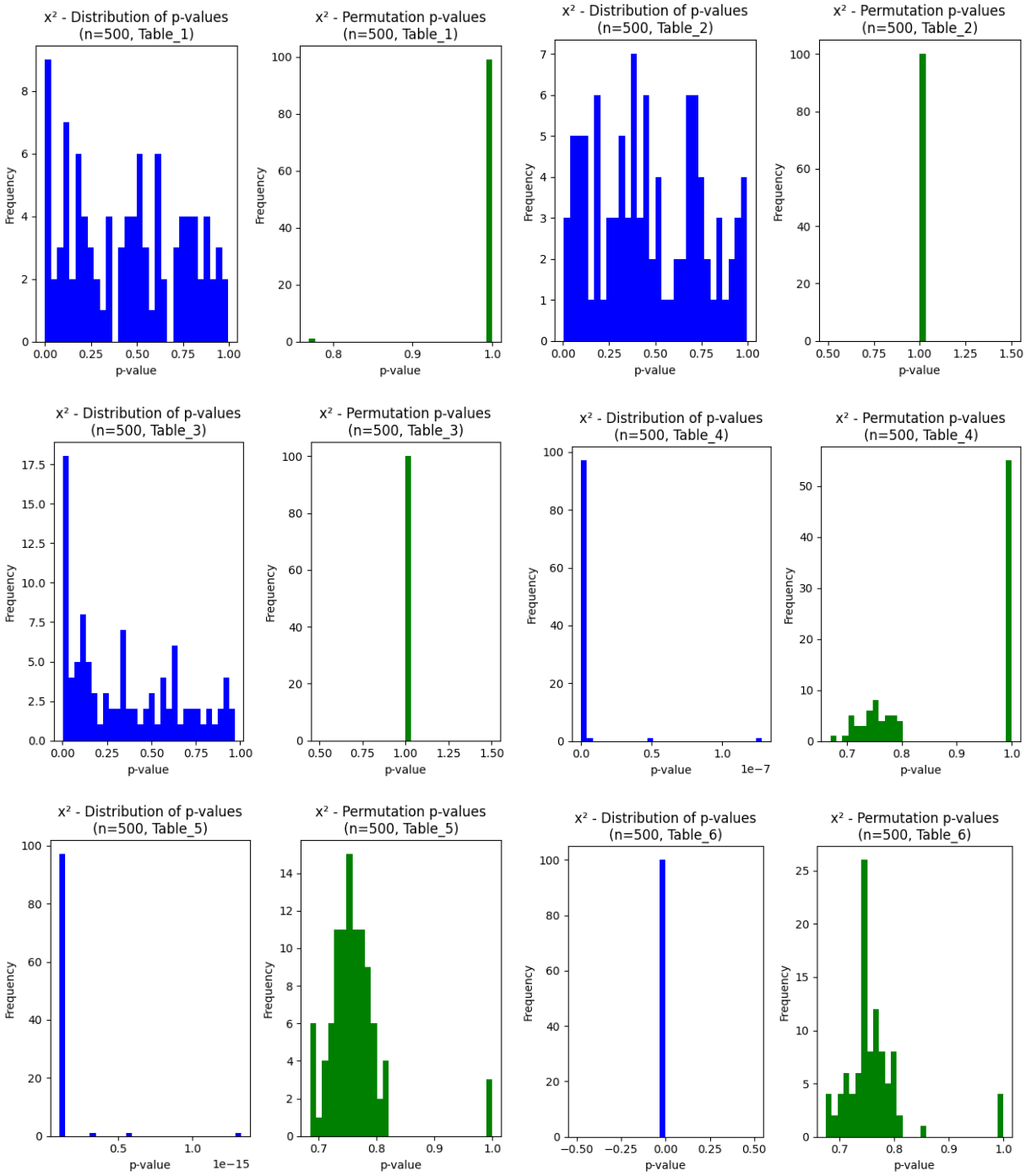
N = 25


x² - Distribution of p-values (n=25, Table_1)
x² - Permutation p-values (n=25, Table_1)
x² - Distribution of p-values (n=25, Table_2)
x² - Permutation p-values (n=25, Table_2)
x² - Distribution of p-values (n=25, Table_3)
x² - Permutation p-values (n=25, Table_3)
x² - Distribution of p-values (n=25, Table_4)
x² - Permutation p-values (n=25, Table_4)
x² - Distribution of p-values (n=25, Table_5)
x² - Permutation p-values (n=25, Table_5)
x² - Distribution of p-values (n=25, Table_6)
x² - Permutation p-values (n=25, Table_6)
Frequency
p-value

N = 100


x² - Distribution of p-values (n=100, Table_1)
x² - Permutation p-values (n=100, Table_1)
x² - Distribution of p-values (n=100, Table_2)
x² - Permutation p-values (n=100, Table_2)
x² - Distribution of p-values (n=100, Table_3)
x² - Permutation p-values (n=100, Table_3)
x² - Distribution of p-values (n=100, Table_4)
x² - Permutation p-values (n=100, Table_4)
x² - Distribution of p-values (n=100, Table_5)
x² - Permutation p-values (n=100, Table_5)
x² - Distribution of p-values (n=100, Table_6)
x² - Permutation p-values (n=100, Table_6)
Frequency
p-value
1e−10

N = 500


x² - Distribution of p-values (n=500, Table_1)
x² - Permutation p-values (n=500, Table_1)
x² - Distribution of p-values (n=500, Table_2)
x² - Permutation p-values (n=500, Table_2)
x² - Distribution of p-values (n=500, Table_3)
x² - Permutation p-values (n=500, Table_3)
x² - Distribution of p-values (n=500, Table_4)
x² - Permutation p-values (n=500, Table_4)
x² - Distribution of p-values (n=500, Table_5)
x² - Permutation p-values (n=500, Table_5)
x² - Distribution of p-values (n=500, Table_6)
x² - Permutation p-values (n=500, Table_6)
Frequency
p-value
1e−7
1e−15