# Imagry – Home Assignment

Statistics:

1. **2d centers histogram**

I generated 1000 inputs: . Each time R was chosen first, its coordinates were generated by:

And then coordinates were generated uniformly inside R boundaries.

Hence, R significantly determines the locations of centers, so we expect the centers to distribute similarly to R - a normal behavior of the histogram:

Chart

Description automatically generated

Figure 1 – 2d centers histogram. Statistics over 1e3 inputs

As can be seen, the histogram seems to be x-y symmetric, as expected, with zero mean and std of ~4 for each axis.

Plotting the very same histogram for 100,000 inputs, I got a smoother histogram with a clearer normal distribution.

Chart

Description automatically generatedFigure 1 – 2d centers histogram. Statistics over 1e5 inputs

1. **2d coverage histogram**

To see what the spatial distribution of rectangles areas is, I created a manually histogram. To do so I created a mask which contains '1' for `pixels` that are in rectangle, and zero otherwise. To fill in this grid I needed a quantization - each coordinate was rounded. For this process to work well, I wanted a fine quantization, hence I increased R\_maximal\_size very much, causing the round process to be less significant, and I did not run this with big N for the same reason (large N cause [x1,x2] segments to be small), and then the round operation can nullify them.

For sanity check, I looked at the density of 1 epoch with many rectangles – I expected to see vertical lines since my algorithm generates non-overlapped rectangles in x-axis. The whole displayed image is of the maximal possible size of R, but in this case (and many others) R was smaller so the lines are located in a patch of the overall figure only.

Chart, histogram

Description automatically generated

When I decreased the number of rectangles (5), I expected to see wider shapes, in compared to the same number of epochs with more rectangles (50), and this is what I got:

Chart

Description automatically generated with low confidence

Chart

Description automatically generated

1. **Relative coverage Vs number of rectangles**

Now let's look on the coverage area out of R's area – what is the behavior of the rectangles' relative coverage. To compute it, we need to calculate the sum of rectangles areas, divided by R's area, for each epoch.

Those are the results for 1000 epochs with N=5, 1e5 epochs with N=5, and 1e5 epochs with N=1e5, respectfully.

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

As expected, the greater the number of epochs, the smoother the distribution and it goes to normal distribution.

Interesting thing is that if N goes bigger, the relative area goes to zero. It seems odd at first but it does make sense, because of the way I created the rectangles in first place – I pick 2\*N random variables from , **sort** them and then take pairs which are the rectangle (x1, x2). As N became bigger, I generate more x values from a uniform distribution hence those values are covering the segment . When that's happening, each rectangle has near-to-zero area, and so is the sum of all areas.

It means that when N goes bigger, my generator creates thinner rectangles. I could change it by ensure non overlapping based on both x and y segments, not only one of them.

Bonus: if R is not aligned to axis – it is rotated in some way in 2d.

Where is the rotation matrix. We need to find rotate and all the smaller rectangles – if we do so, we can apply the algorithm we constructed previously.

To find we only need to find . Having R's coordinates, we can compute from the dot product: where .

X1,Y1

X2,Y2