****

**PROGRAMA PROFESIONAL:**

CIENCIA DE LA COMPUTACIÓN

**TÍTULO DEL TRABAJO:**

MALDICIÓN DE LA DIMENSIONALIDAD

**CURSO:**

ESTRUCTURAS DE DATOS AVANZADAS

**ESTUDIANTE:**

Becerra Sipiran, Cledy Elizabeth

**SEMESTRE:** VI

**AÑO:** 2022

**“El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo.”**

1. **CÓDIGO**

#include <iostream>

#include <cmath>

#include <vector>

#include <algorithm>

#include <cstdlib>

#include <fstream>

#include <string>

using namespace std;

vector<double> distances;

vector<vector<double>> finaldata;

int Lim(){

return rand()%1000;

}

bool Find(double n){

for(auto i = finaldata.begin(); i != finaldata.end(); i++)

if((\*i)[0] == n) return true;

return false;

}

template <class T>

T Distance(vector<T> a,vector<T> b){

T sum = 0;

for(int i = 0; i < a.size(); i++)

sum += pow(a[i] - b[i], 2);

return sqrt(sum);

}

double Count(vector<double>::iterator i, double n){

int count = 0;

while(i != distances.end()){

if(\*i == n) count++;

i++;

}

return count;

}

int main() {

for(int nelem = 10; nelem <= 50; nelem += 10){

for(int nvectors = 10000; nvectors <= 50000; nvectors += 10000){

vector<double> A(nelem);

generate(A.begin(), A.end(), Lim);

for(int j = 0; j < nvectors; ++j){

vector<double> B(nelem);

generate(B.begin(), B.end(), Lim);

distances.push\_back(Distance<double>(A,B));

}

for (auto i = distances.begin(); i != distances.end(); i++) {

if(!Find(\*i)){

vector<double> tmp;

tmp.push\_back(\*i); tmp.push\_back(Count(i,\*i));

finaldata.push\_back(tmp);

}

}

fstream file;

file.open(to\_string(nelem) + "\_" + to\_string(nvectors/1000) + "k.txt", ios::out);

for(auto i = finaldata.begin(); i != finaldata.end(); i++)

file << (\*i)[0] << " " << (\*i)[1] << endl;

file.close();

distances.clear();

finaldata.clear();

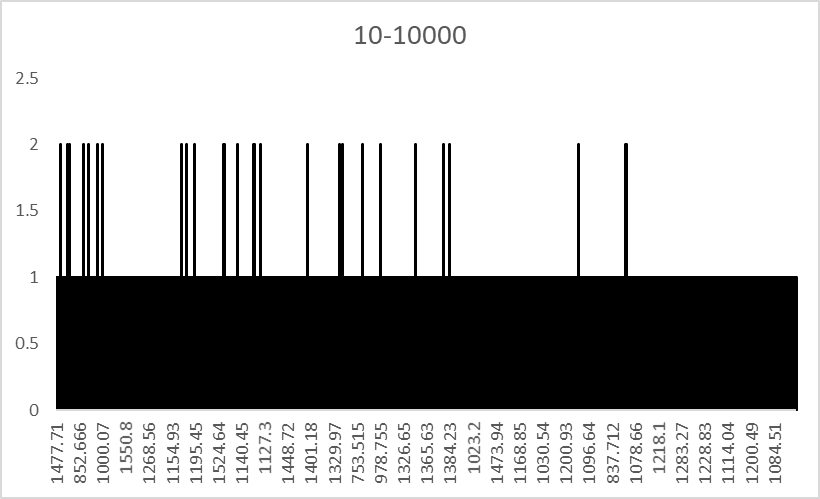
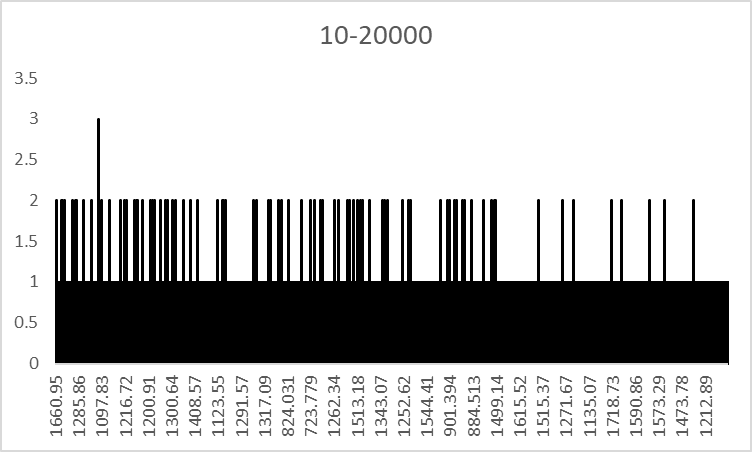
}

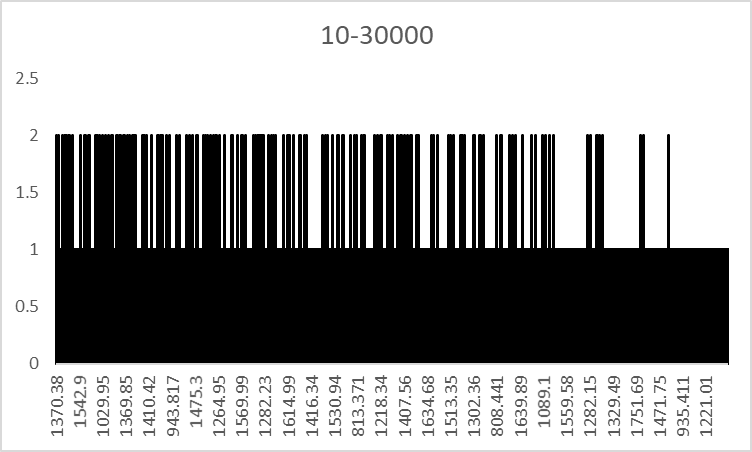
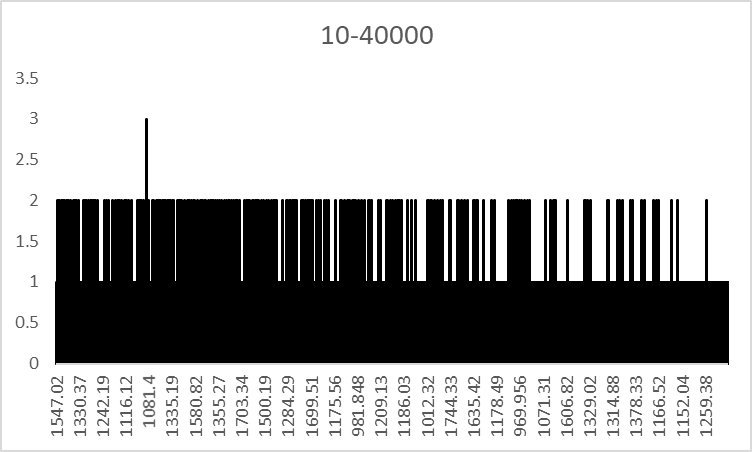
}

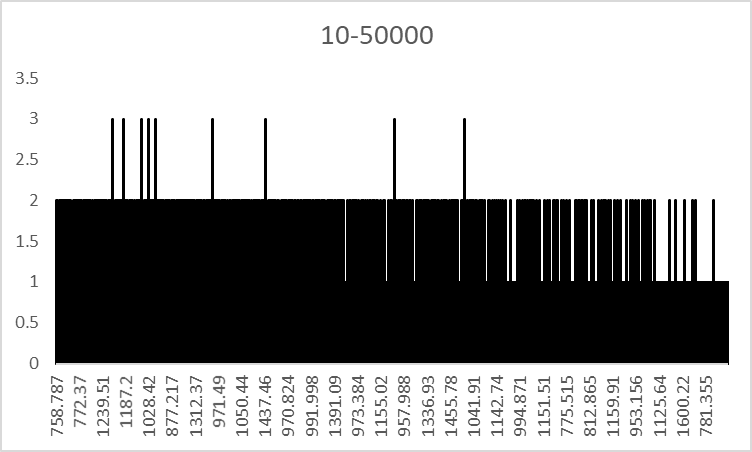
return 0;

}

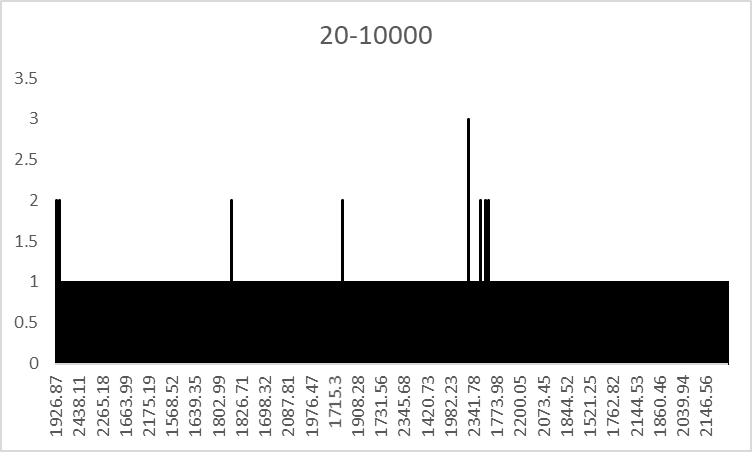
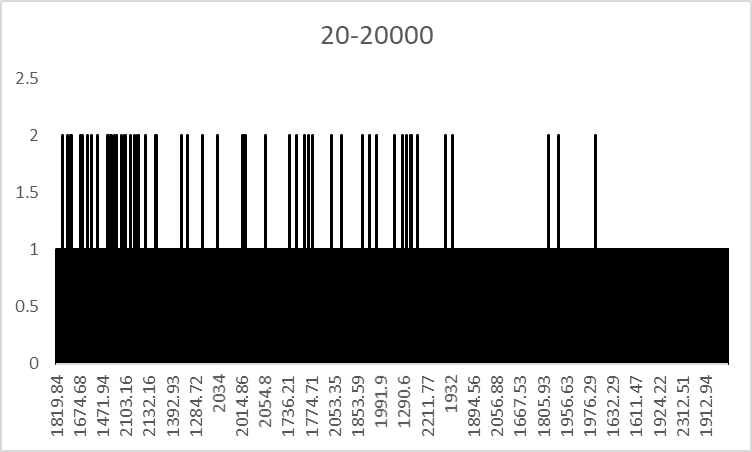
1. **HISTOGRAMAS**
2. 10 dimensiones.

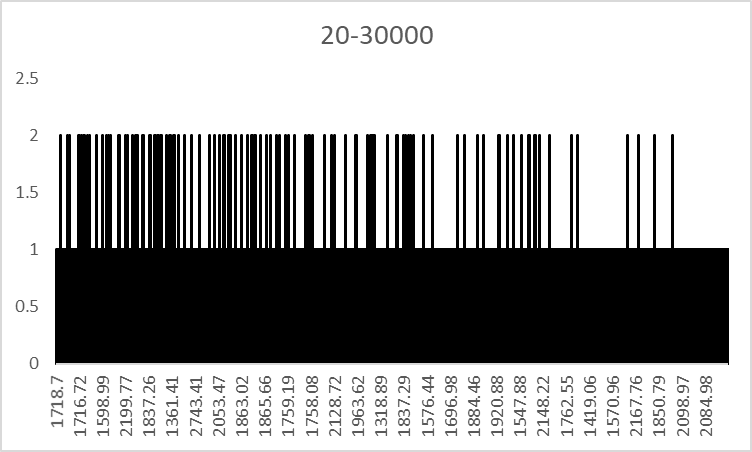
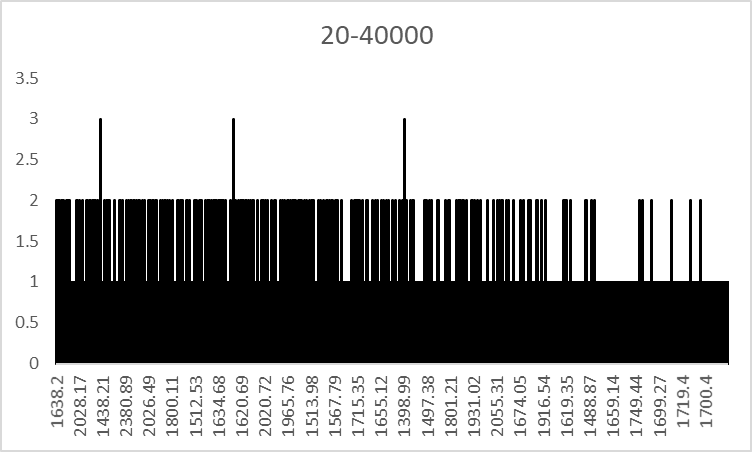
 

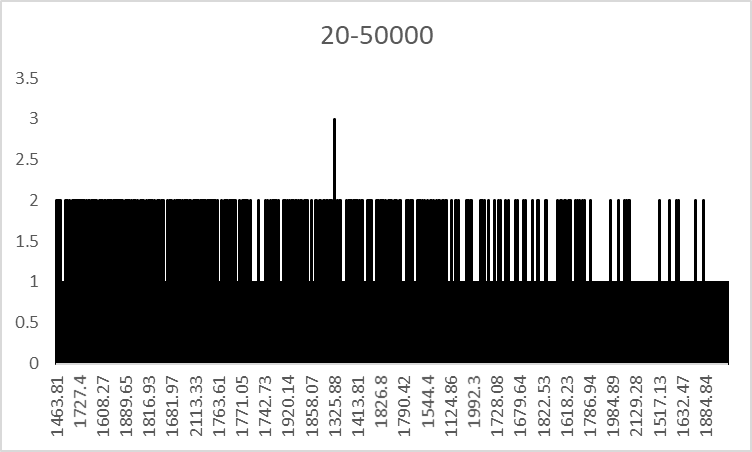
 



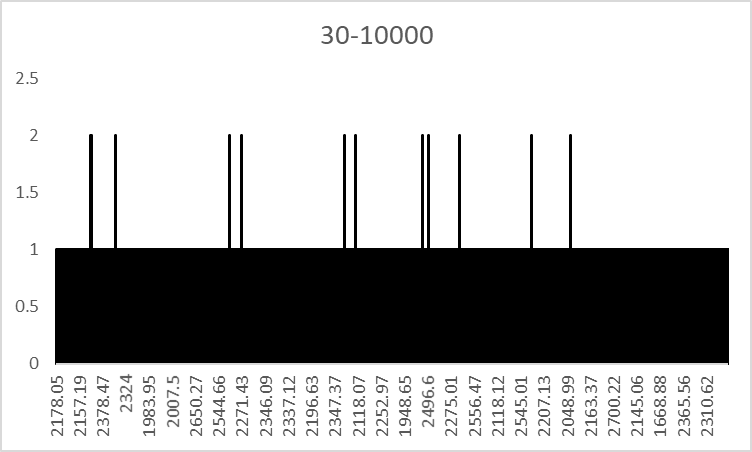
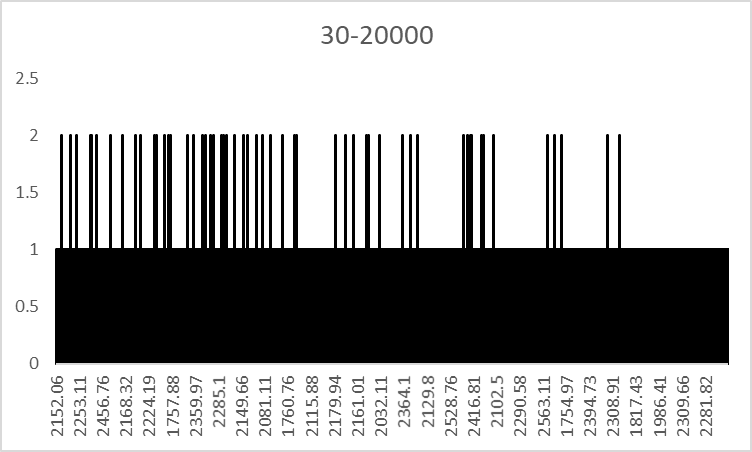
1. 20 dimensiones.

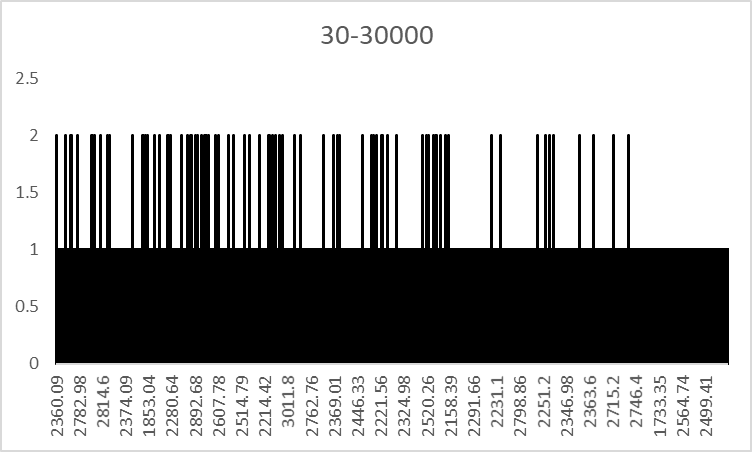
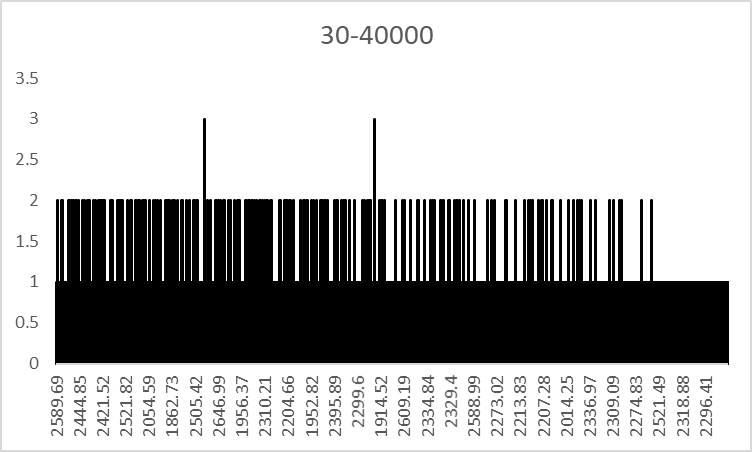
 

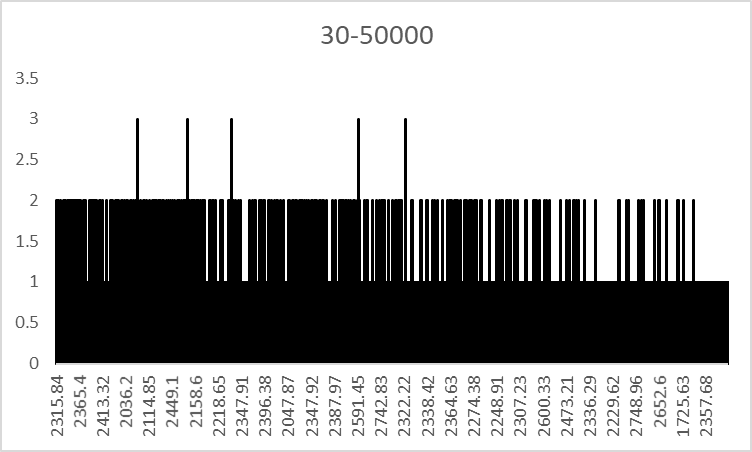
 



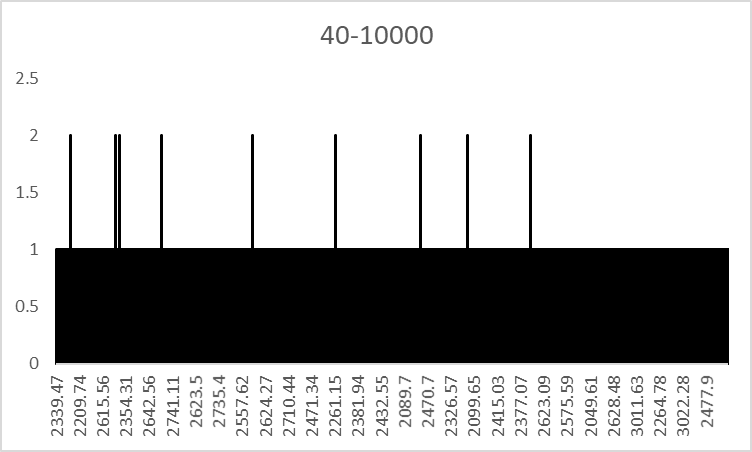
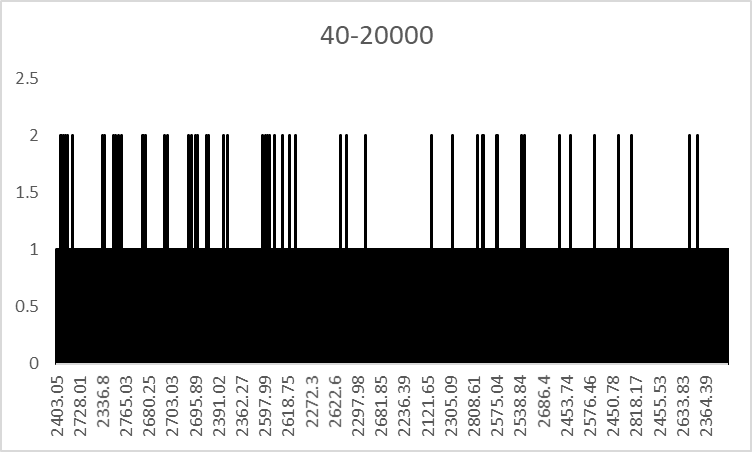
1. 30 dimensiones.

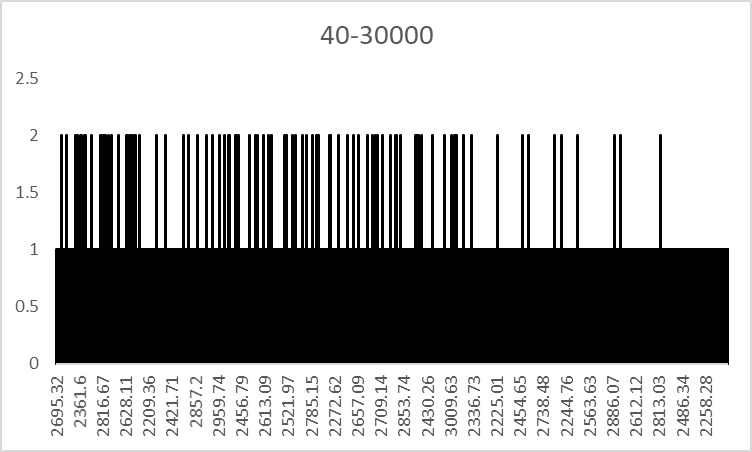
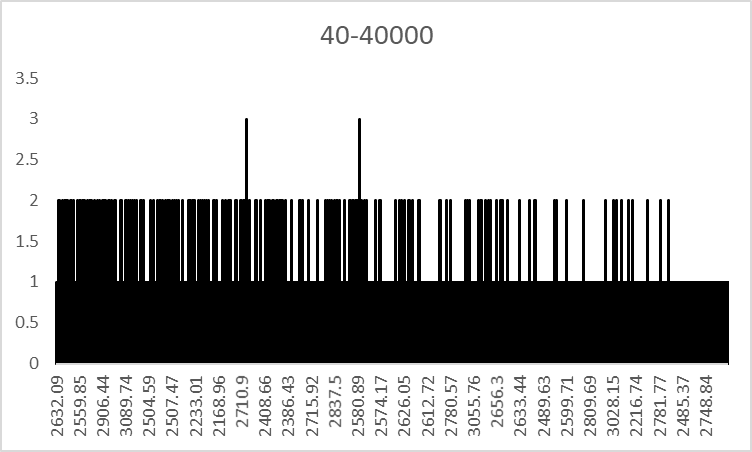
 

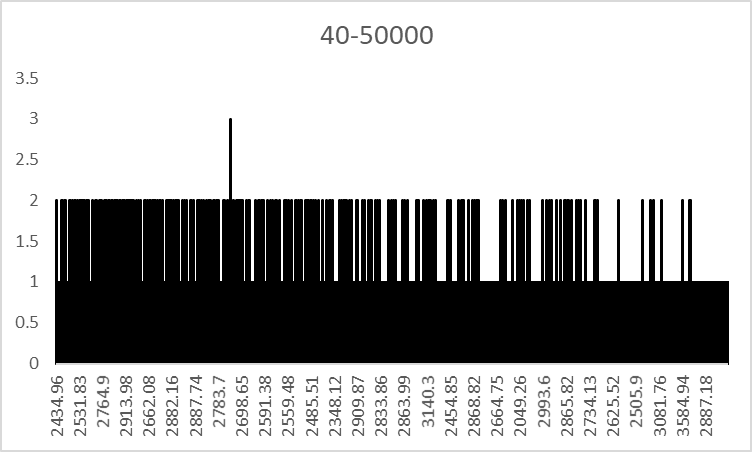
 



1. 40 dimensiones.



1. 50 dimensiones.

