Mersenne Primes
and their uniform natural distribution of digits

Matei Octavian Mihai[1]

[1]Technical University of Cluj-Napoca, Department of Computer Science, Cluj-Napoca, Cluj

The aim of the study was to analyze the distribution of digits in the first 20 largest Mersenne Primes and to establish a connection between these and the natural probability of occurrences of digits in random situations according to the Law of Large Numbers, proving that even the most ordered situations can and will obey the simple and elegant rules of entropy in our universe. Combining a series of computer programs and data manipulation software we can observe the hypothesis proving to be right computing the average values for the percentage of appearances of digits in the numbers and their error margin, making a curve that seems a straight line, with small variations along the range of decimal base.

INTRODUCTION

In mathematics, a Mersenne prime is a prime number that is one less than a power of two. That is, it is a prime number of the form $Mn = 2^n - 1$ for some integer n. They are named after Marin Mersenne, a French Minim friar, who studied them in the early 17th century.
If n is a composite number, then so is $2^n - 1$. ($2^{ab} - 1$ is divisible by both $2^a - 1$ and $2^b - 1$.) This definition is therefore equivalent to the definition as a prime number of the form $Mp = 2^p - 1$ for some prime p.

More generally, numbers of the form $Mn = 2^n - 1$ without the primality requirement may be called Mersenne numbers. Sometimes, however, Mersenne numbers are defined to have the additional requirement that n be prime. The smallest composite Mersenne number with prime exponent n is $2^{11} - 1 = 2047 = 23 \times 89$.

Another theoretical concept to be known is the the law of large numbers (LLN) which describes the result of performing the same experiment a large number of times. According to the law, the average of the results obtained from a large number of trials should be close to the expected value and will tend to become closer to the expected value as more trials are performed.

The main question of this paper is if ordered sequences of digits taking the form of Mersenne prime numbers can respect the Law of Large Numbers (from now on, shortened to LLN) or if they respect the same approximate distribution as they grow.

Procedure of finding the data

The research was done in two parts, the computational one and the observatory one.

In the computational one, I extracted from the internet the first 20 largest Mersenne numbers, built a simple C computer program to find the frequency of each digit and store it in a file to be further processed. The code below parses the current file and outputs to the storing file the frequency of each digit in the forma *digit : frequency:*

```
void ParseFile(FILE * fileIn, FILE * fileOut,int __power){
   fprintf(fileOut,"%d\n",__power);                          //  Dispaying the power of the current number for indexing
   int * frequencyDigits= (int*)malloc(sizeof(int)*10);             //  Allocate memory for the frequency of digits
   memset(frequencyDigits,0,10);                             //  Set the initial frequency to 0

   char lineDigit;                                   //  Declaring the character representing each digit
   while((lineDigit=getc(fileIn))!=EOF)                      //  Read each digit of the number from current file
   {
      if(( lineDigit >= '0' && lineDigit <= '9' ))                //  check if the character is a digit
      {
         frequencyDigits[lineDigit-'0']++;                    //  if it is, increase the frequency of that digit
      }
   }

   for(int i=0;i<10;i++)
      fprintf(fileOut,"       %d : %d\n",i,frequencyDigits[i]);    //  print each frequency along with an indentation to
index easier

   free(frequencyDigits);                                 //  free the memory allocated for frequency
}
int main(){
   FILE *fpIn  = fopen("List.txt","r");              //  input stream containing all the numbers with a specific power
   FILE *fpOut = fopen("OutputFile.txt","w");        //  output stream
   char * lineFile = (char*)malloc(sizeof(char)*500);

   while ( fgets ( lineFile, 500, fpIn ) != NULL )    //  read each power until the end of file
   {
      int __power;
      printf("%s\n",lineFile);                   //  print to the console to verify the output directly
      __power = atoi(lineFile);                   //  print into the folder the power to be able to format the Excel file
      char * actualFilePath = (char*)malloc(sizeof(char)*150);

                                    //  the whole procedure bellow builds the path of the following file
      strcpy(actualFilePath,"Resources/M");
      strcpy(actualFilePath+strlen(actualFilePath),lineFile);
      strcpy(actualFilePath+strlen(actualFilePath)-1,".txt");
                                    //  end of procedure

      FILE * fileIn = fopen(actualFilePath,"r");
      ParseFile(fileIn,fpOut,__power);              //  Call the function listed above to parse the file
      fclose(fileIn);
   }

   fclose(fpIn);
   fclose(fpOut);

   return 0;
}
```

The whole procedure lasted for 7 seconds and after the storing file was completed, I export the result to an Excel spreadsheet to further analyze the raw data. After that, I computed the following:

The average occurrence of digits with the formula:

=AVERAGE(CELLS_RANGE) *(1)*

The error margin:

=ABS(AVERAGE, EACH_CELLS) *(2)*

Percentage of each digit:

*=FLOOR.MATH(FREQUENCY/TOTAL,0.00001) *100* *(3)*

Error Percentage of each digit:

*=FLOOR.MATH(ERROR_MARGIN_DIGIT/TOTAL_MARGIN,0.00001) *100* (4)

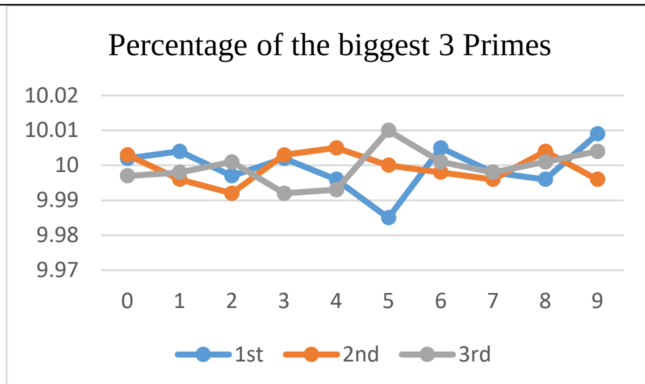
After the cells were fed with data, I drew the graph for percentage of the first 3 largest numbers, error percentage and frequency error for them. On a separate sheet, the total percentage per digit was calculated and, also, the total error percentage for each digit. Furthermore, it was computed the total absolute percentage and total absolute error percentage and drew each one of these, on the other hand, to observe the miniscule difference between the LLN and the reality, it was also necessary a graph with total percentage relative to 10%.

Results

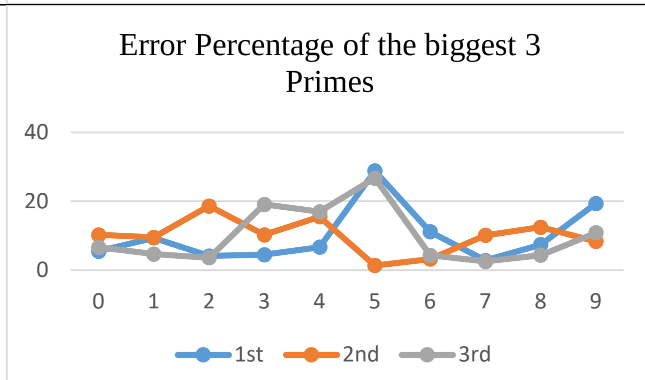
To my surprise, the set of data resulted from the algorithms and procedures resembles the probability of frequency of a digit as told by the LLN. This law says that if we take a large number, n, of primes, where n tends to infinity, and apply the frequency test, each digit should appear with a probability of 1/10, or 10% of cases, meaning we should get a linear constant distribution of each digit.

For the largest Mersenne 3 numbers, the distribution is as follows:

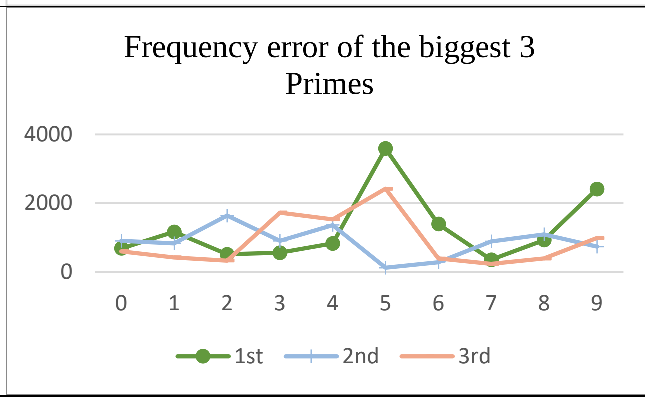
The percentage of occurrence of each digit was sampled and drawn a graph showing the relative distribution around 10, with an error of 0.015% from the actual LLN.



The error percentage of each digit was sampled and drawn a graph showing a Gaussian-like distribution, a pure mathematical concept applying on a real set of numbers and resulting some fascinating graphs.

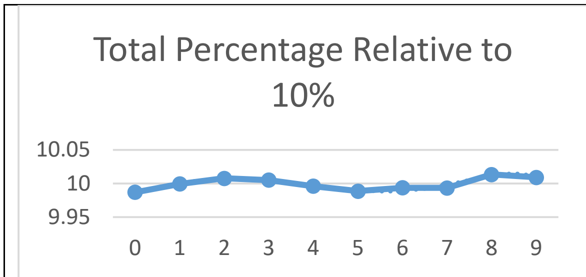


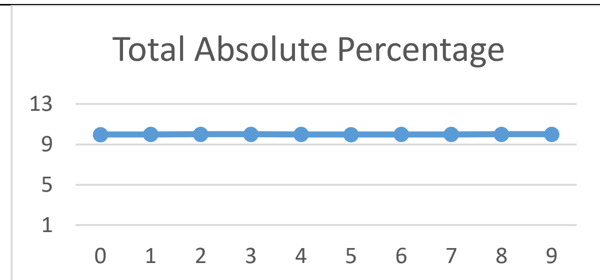
The frequency of the error margin was sampled and drawn a graph showing a similar distribution as the error percentage, that mean a Gaussian-like distribution.

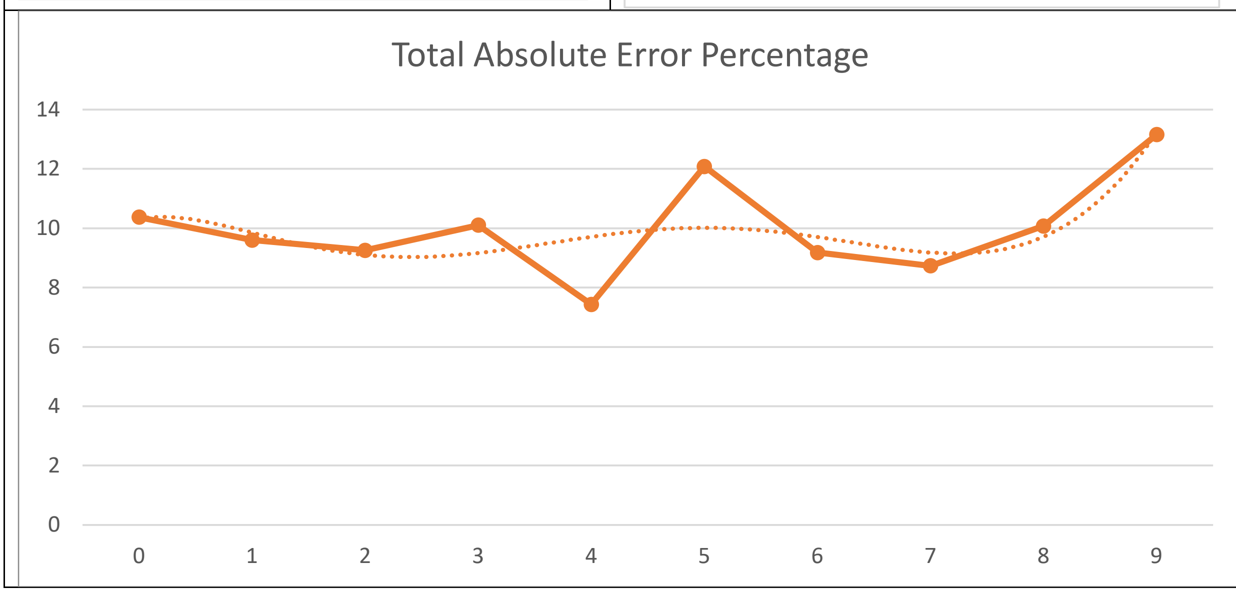


It is remarkable that the LLN could predict the distribution of an extremely ordered sequence of digits, the primes, moreover, in their error distribution, it is clear that a Gaussian Bell can be drawn, therefore, the Gaussian distribution appears.

Calculating the total occurrence percentage and total error percentage of the first 20 biggest Mersenne primes, the LLN appears to be so accurate, that there is an error margin of approximately ±0.015%(meaning a difference of frequency of around 2000), corresponding to the peaks of the previous graphs. On the other hand, the total absolute error percentage flattens down, sign that the relative frequency of each digit approaches the same value, amount predicted by the LLN.







What it means?

The finding of such relationship between a pure mathematical law and the most ordered and strict sequence built by mankind probably is a coincidence or it means that even this kind of structure obeys the laws of the universe, fueling our curiosity to find and research all the possible formations of information in the cosmos, but currently I do not possess such abilities and I let others more capable to formulate more elaborate hypotheses.

Bibliography

Mersenne Primes	https://en.wikipedia.org/wiki/Mersenne_prime
Law of Large Numbers	https://en.wikipedia.org/wiki/Law_of_large_numbers
List of Mersenne Primes	https://www.mersenne.org/primes/
Gaussian Distribution	https://en.wikipedia.org/wiki/Normal_distribution
The code discussed above including the input and output files	https://github.com/tavisit/ParsingListOfFiles/
Microsoft Excel	https://products.office.com/en/excel

Appendix and the whole data set

Bellow there is all the data analyzed in this paper to promote self-discovery and curiosity.

Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
Number is : 2^ 756839 -1									
0	22511	227832	22783.2	0	272.2	0	9.88	0	15.434
1	22840			1	56.8	1	10.024	1	3.22
2	23106			2	322.8	2	10.141	2	18.303
3	22876			3	92.8	3	10.04	3	5.261
4	22621			4	162.2	4	9.928	4	9.197
5	22391			5	392.2	5	9.827	5	22.238
6	22889			6	105.8	6	10.046	6	5.999
7	22728			7	55.2	7	9.975	7	3.129
8	22835			8	51.8	8	10.022	8	2.937
9	23035			9	251.8	9	10.11	9	14.277
Number is : 2^ 859433 -1									
0	25799	258716	25871.6	0	72.6	0	9.971	0	4.846
1	25928			1	56.4	1	10.021	1	3.765
2	25899			2	27.4	2	10.01	2	1.829
3	26093			3	221.4	3	10.085	3	14.779
4	25846			4	25.6	4	9.99	4	1.708
5	26113			5	241.4	5	10.093	5	16.114
6	25766			6	105.6	6	9.959	6	7.049
7	25357			7	514.6	7	9.801	7	34.352
8	25841			8	30.6	8	9.988	8	2.042
9	26074			9	202.4	9	10.078	9	13.511
Number is : 2^ 1257787 -1									
0	37562	378632	37863.2	0	301.2	0	9.92	0	18.433
1	37823			1	40.2	1	9.989	1	2.46
2	38229			2	365.8	2	10.096	2	22.386
3	38003			3	139.8	3	10.036	3	8.555
4	37940			4	76.8	4	10.02	4	4.7
5	37446			5	417.2	5	9.889	5	25.532
6	37856			6	7.2	6	9.998	6	0.44
7	38036			7	172.8	7	10.045	7	10.575
8	37925			8	61.8	8	10.016	8	3.782
9	37812			9	51.2	9	9.986	9	3.133
Number is : 2^ 1398269 -1									
0	42275	420921	42092.1	0	182.9	0	10.043	0	13.635
1	42000			1	92.1	1	9.978	1	6.865
2	42058			2	34.1	2	9.991	2	2.542

Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
3	42073			3	19.1	3	9.995	3	1.423
4	42225			4	132.9	4	10.031	4	9.907
5	42050			5	42.1	5	9.989	5	3.138
6	41878			6	214.1	6	9.949	6	15.96
7	42088			7	4.1	7	9.999	7	0.305
8	42447			8	354.9	8	10.084	8	26.457
9	41827			9	265.1	9	9.937	9	19.762
Number is : 2^ 2976221 -1									
0	89417	895932	89593.2	0	176.2	0	9.98	0	6.288
1	89346			1	247.2	1	9.972	1	8.822
2	89219			2	374.2	2	9.958	2	13.354
3	89693			3	99.8	3	10.011	3	3.561
4	89364			4	229.2	4	9.974	4	8.179
5	89600			5	6.8	5	10	5	0.242
6	89219			6	374.2	6	9.958	6	13.354
7	89995			7	401.8	7	10.044	7	14.339
8	89951			8	357.8	8	10.039	8	12.769
9	90128			9	534.8	9	10.059	9	19.086
Number is : 2^ 3021377 -1									
0	90508	909526	90952.6	0	444.6	0	9.951	0	24.482
1	91095			1	142.4	1	10.015	1	7.841
2	91259			2	306.4	2	10.033	2	16.872
3	91013			3	60.4	3	10.006	3	3.325
4	91065			4	112.4	4	10.012	4	6.189
5	90808			5	144.6	5	9.984	5	7.962
6	91239			6	286.4	6	10.031	6	15.77
7	90825			7	127.6	7	9.985	7	7.026
8	90914			8	38.6	8	9.995	8	2.125
9	90800			9	152.6	9	9.983	9	8.403
Number is : 2^ 6972593 -1									
0	210190	2098960	209896	0	294	0	10.014	0	8.818
1	210744			1	848	1	10.04	1	25.434
2	209678			2	218	2	9.989	2	6.538
3	209382			3	514	3	9.975	3	15.416
4	209832			4	64	4	9.996	4	1.919
5	209863			5	33	5	9.998	5	0.989
6	210356			6	460	6	10.021	6	13.797
7	209314			7	582	7	9.972	7	17.456
8	209961			8	65	8	10.003	8	1.949
9	209640			9	256	9	9.987	9	7.678

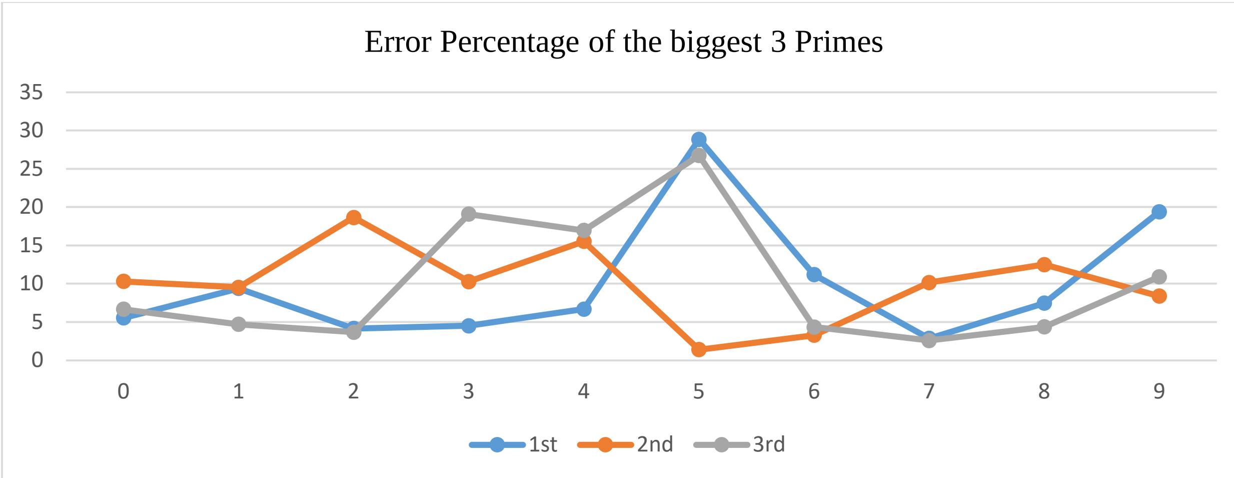
Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
Number is : 2^ 13466917 -1									
0	405083	4053946	405394.6	0	311.6	0	9.992	0	7.326
1	405614			1	219.4	1	10.005	1	5.158
2	405068			2	326.6	2	9.991	2	7.678
3	405928			3	533.4	3	10.013	3	12.541
4	405491			4	96.4	4	10.002	4	2.266
5	404915			5	479.6	5	9.988	5	11.276
6	405154			6	240.6	6	9.994	6	5.656
7	405308			7	86.6	7	9.997	7	2.036
8	406672			8	1277.4	8	10.031	8	30.033
9	404713			9	681.6	9	9.983	9	16.025
Number is : 2^ 20996011 -1									
0	631705	6320430	632043	0	338	0	9.994	0	3.932
1	632720			1	677	1	10.01	1	7.875
2	630989			2	1054	2	9.983	2	12.261
3	631467			3	576	3	9.99	3	6.7
4	632004			4	39	4	9.999	4	0.453
5	633283			5	1240	5	10.019	5	14.425
6	630929			6	1114	6	9.982	6	12.959
7	633503			7	1460	7	10.023	7	16.984
8	632964			8	921	8	10.014	8	10.714
9	630866			9	1177	9	9.981	9	13.692
Number is : 2^ 24036583 -1									
0	722613	7235733	723573.3	0	960.3	0	9.986	0	13.413
1	723188			1	385.3	1	9.994	1	5.382
2	722754			2	819.3	2	9.988	2	11.444
3	722181			3	1392.3	3	9.98	3	19.448
4	723758			4	184.7	4	10.002	4	2.579
5	724196			5	622.7	5	10.008	5	8.698
6	723856			6	282.7	6	10.003	6	3.948
7	724543			7	969.7	7	10.013	7	13.545
8	723551			8	22.3	8	9.999	8	0.311
9	725093			9	1519.7	9	10.021	9	21.227
Number is : 2^ 25964951 -1									
0	782138	7816230	781623	0	515	0	10.006	0	13.131
1	782118			1	495	1	10.006	1	12.621
2	781551			2	72	2	9.999	2	1.835
3	781856			3	233	3	10.002	3	5.94
4	780817			4	806	4	9.989	4	20.55
5	781588			5	35	5	9.999	5	0.892

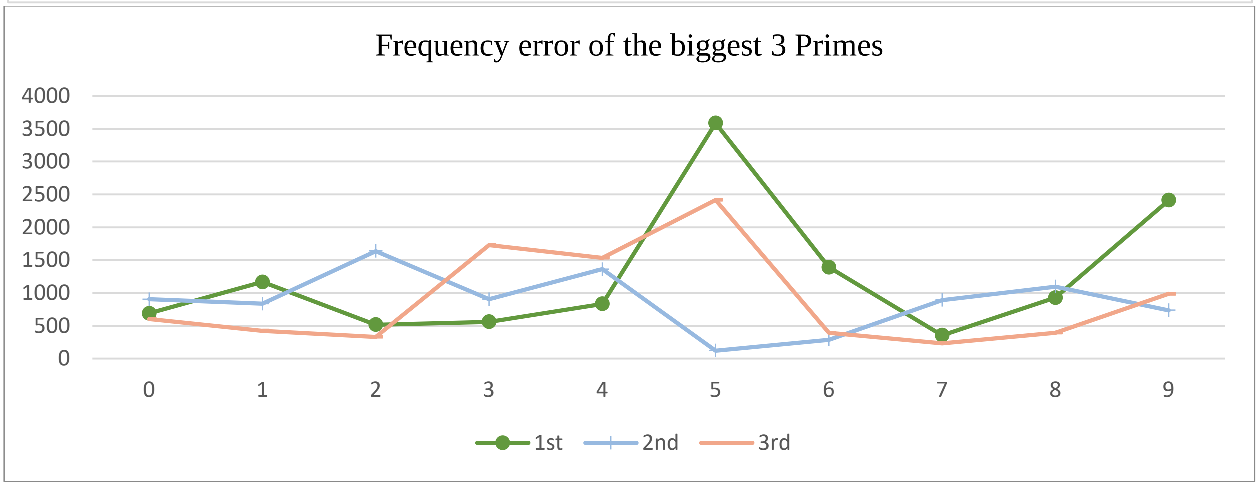
Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
6	780774			6	849	6	9.989	6	21.647
7	781662			7	39	7	10	7	0.994
8	781424			8	199	8	9.997	8	5.073
9	782302			9	679	9	10.008	9	17.312
Number is : 2^ 30402457 -1									
0	913468	9152052	915205.2	0	1737.2	0	9.981	0	16.444
1	914272			1	933.2	1	9.989	1	8.833
2	916362			2	1156.8	2	10.012	2	10.95
3	913997			3	1208.2	3	9.986	3	11.436
4	914191			4	1014.2	4	9.988	4	9.6
5	916441			5	1235.8	5	10.013	5	11.698
6	915744			6	538.8	6	10.005	6	5.1
7	915905			7	699.8	7	10.007	7	6.624
8	916856			8	1650.8	8	10.018	8	15.626
9	914816			9	389.2	9	9.995	9	3.684
Number is : 2^ 32582657 -1									
0	981284	9808358	980835.8	0	448.2	0	10.004	0	6.003
1	981525			1	689.2	1	10.007	1	9.231
2	980761			2	74.8	2	9.999	2	1.001
3	978652			3	2183.8	3	9.977	3	29.249
4	980519			4	316.8	4	9.996	4	4.243
5	981645			5	809.2	5	10.008	5	10.838
6	979697			6	1138.8	6	9.988	6	15.253
7	980817			7	18.8	7	9.999	7	0.251
8	982176			8	1340.2	8	10.013	8	17.95
9	981282			9	446.2	9	10.004	9	5.976
Number is : 2^ 37156667 -1									
0	1117011	11185272	1118527	0	1516.2	0	9.986	0	15.673
1	1116735			1	1792.2	1	9.983	1	18.526
2	1120012			2	1484.8	2	10.013	2	15.348
3	1118247			3	280.2	3	9.997	3	2.896
4	1119048			4	520.8	4	10.004	4	5.383
5	1118786			5	258.8	5	10.002	5	2.675
6	1117279			6	1248.2	6	9.988	6	12.903
7	1119760			7	1232.8	7	10.011	7	12.743
8	1119468			8	940.8	8	10.008	8	9.725
9	1118926			9	398.8	9	10.003	9	4.122
Number is : 2^ 42643801 -1									
0	1285401	12837064	1283706	0	1694.6	0	10.013	0	14.159
1	1281802			1	1904.4	1	9.985	1	15.912

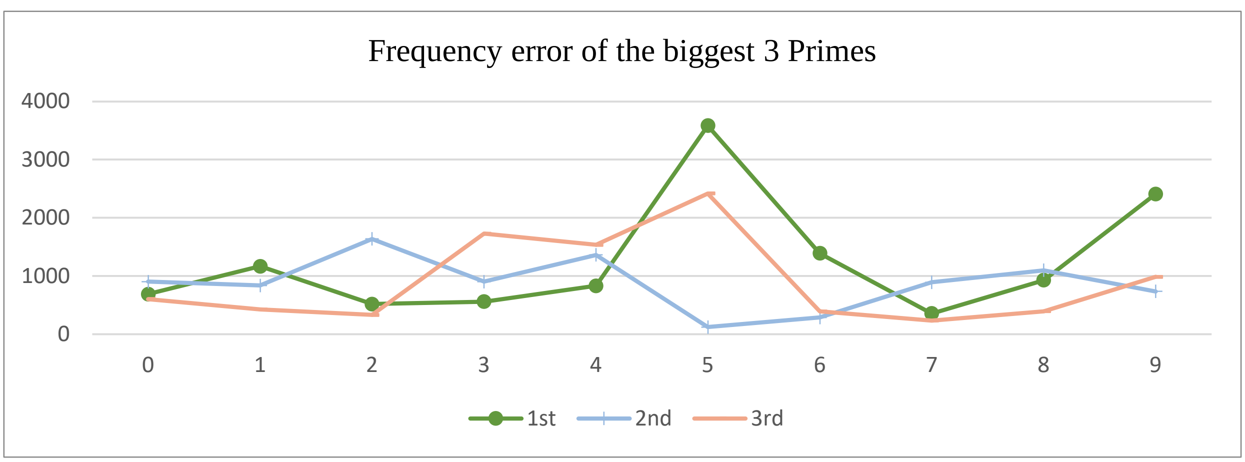
Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
2	1284339			2	632.6	2	10.004	2	5.285
3	1283552			3	154.4	3	9.998	3	1.29
4	1283482			4	224.4	4	9.998	4	1.874
5	1281665			5	2041.4	5	9.984	5	17.057
6	1284838			6	1131.6	6	10.008	6	9.455
7	1284005			7	298.6	7	10.002	7	2.494
8	1285933			8	2226.6	8	10.017	8	18.604
9	1282047			9	1659.4	9	9.987	9	13.865
Number is : 2^ 43112609 -1									
0	1297824	12978189	1297819	0	5.1	0	10	0	0.054
1	1296042			1	1776.9	1	9.986	1	18.861
2	1298077			2	258.1	2	10.001	2	2.739
3	1297701			3	117.9	3	9.999	3	1.251
4	1299093			4	1274.1	4	10.009	4	13.524
5	1296907			5	911.9	5	9.992	5	9.679
6	1296763			6	1055.9	6	9.991	6	11.207
7	1296971			7	847.9	7	9.993	7	9
8	1298243			8	424.1	8	10.003	8	4.501
9	1300568			9	2749.1	9	10.021	9	29.18
Number is : 2^ 57885161 -1									
0	1739652	17425170	1742517	0	2865	0	9.983	0	17.025
1	1743497			1	980	1	10.005	1	5.823
2	1739844			2	2673	2	9.984	2	15.884
3	1745602			3	3085	3	10.017	3	18.332
4	1743528			4	1011	4	10.005	4	6.007
5	1739641			5	2876	5	9.983	5	17.09
6	1742677			6	160	6	10	6	0.95
7	1743436			7	919	7	10.005	7	5.461
8	1743298			8	781	8	10.004	8	4.641
9	1743995			9	1478	9	10.008	9	8.782
Number is : 2^ 74207281 -1									
0	2233259	22338618	2233862	0	602.8	0	9.997	0	6.665
1	2233437			1	424.8	1	9.998	1	4.697
2	2234193			2	331.2	2	10.001	2	3.662
3	2232135			3	1726.8	3	9.992	3	19.093
4	2232328			4	1533.8	4	9.993	4	16.959
5	2236279			5	2417.2	5	10.01	5	26.727
6	2234254			6	392.2	6	10.001	6	4.336
7	2233628			7	233.8	7	9.998	7	2.585
8	2234257			8	395.2	8	10.001	8	4.369

Frequency of the digit		Total digits	Average	Error Margin per digit		Percentage of each digit		Error Percentage of each digit	
9	2234848			9	986.2	9	10.004	9	10.904
Number is : 2^ 77232917 -1									
0	2325846	23249425	2324943	0	903.5	0	10.003	0	10.296
1	2324106			1	836.5	1	9.996	1	9.532
2	2323306			2	1636.5	2	9.992	2	18.649
3	2325845			3	902.5	3	10.003	3	10.284
4	2326305			4	1362.5	4	10.005	4	15.527
5	2325065			5	122.5	5	10	5	1.396
6	2324655			6	287.5	6	9.998	6	3.276
7	2324051			7	891.5	7	9.996	7	10.159
8	2326039			8	1096.5	8	10.004	8	12.495
9	2324207			9	735.5	9	9.996	9	8.381
Number is : 2^ 82589933 -1									
0	2486893	24862048	2486205	0	688.2	0	10.002	0	5.536
1	2487371			1	1166.2	1	10.004	1	9.382
2	2485688			2	516.8	2	9.997	2	4.157
3	2486765			3	560.2	3	10.002	3	4.506
4	2485373			4	831.8	4	9.996	4	6.691
5	2482620			5	3584.8	5	9.985	5	28.839
6	2487595			6	1390.2	6	10.005	6	11.184
7	2485850			7	354.8	7	9.998	7	2.854
8	2485278			8	926.8	8	9.996	8	7.456
9	2488615			9	2410.2	9	10.009	9	19.39

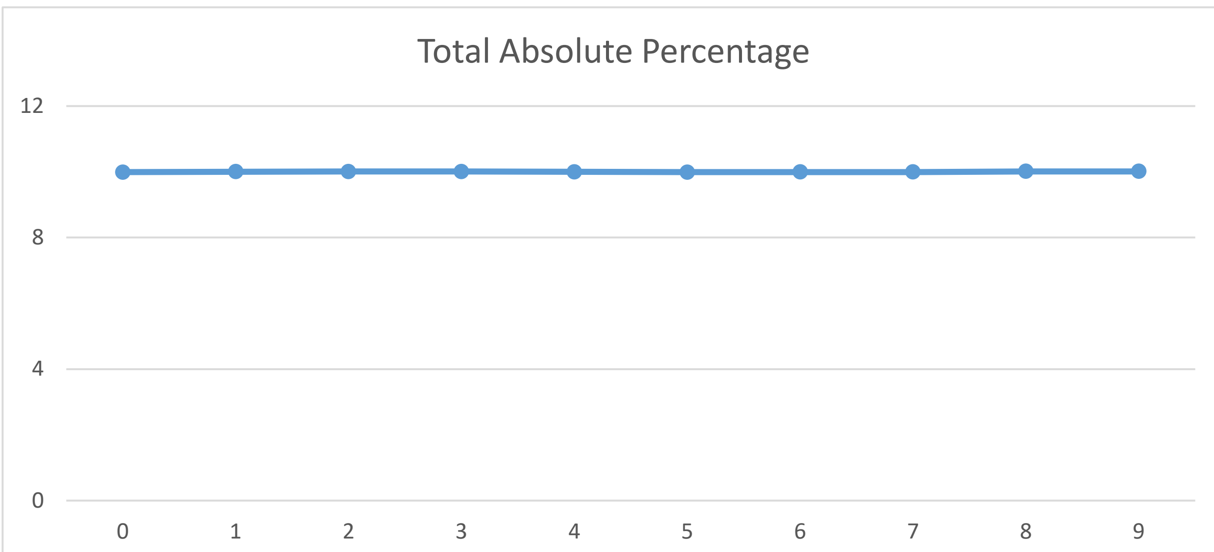
Bellow are 3 graphs illustrating the biggest 3 Mersenne prime numbers

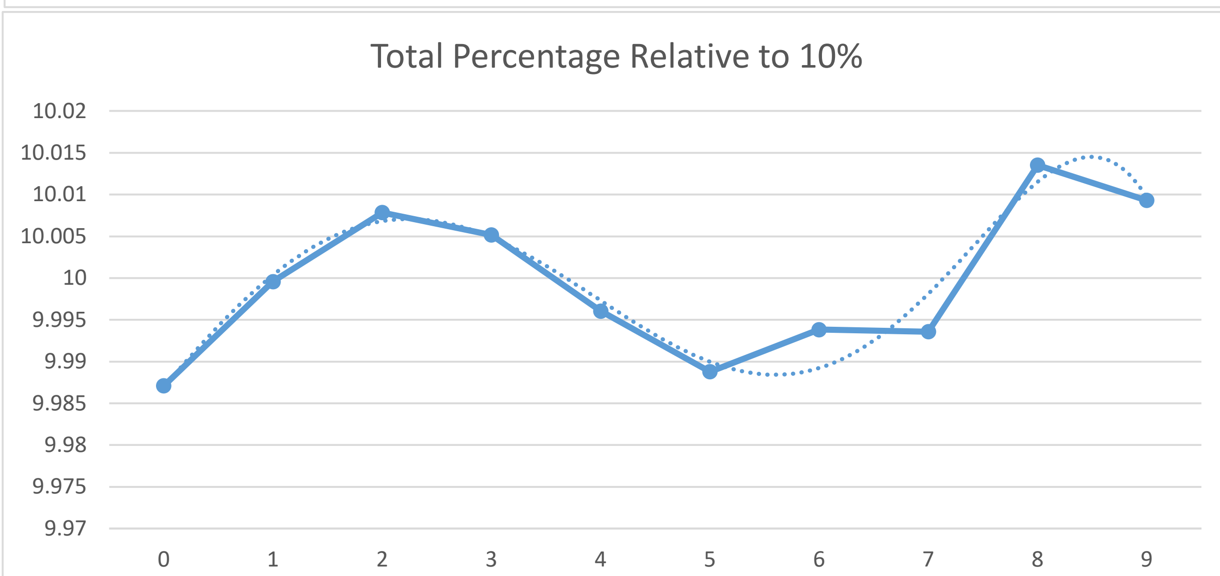


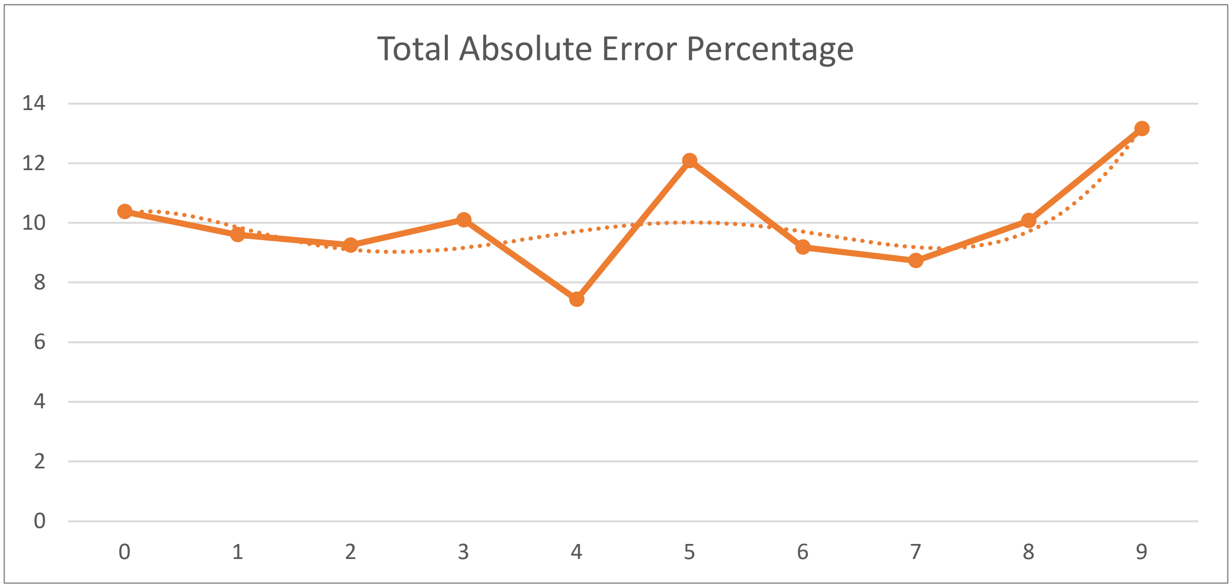




For the first 20 Prime numbers we can conclude:		
	Total Percentage	Total Error Percentage
0	9.987105263	10.37426316
1	9.999578947	9.599947368
2	10.00784211	9.255
3	10.00515789	10.10321053
4	9.996052632	7.435052632
5	9.988789474	12.08121053
6	9.993842105	9.182789474
7	9.993578947	8.730842105
8	10.01352632	10.07547368
9	10.00931579	13.15721053






Total Absolute Error Percentage
14
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
9