Prime Number Gap Pattern

Daniel Dos Santos Bossle, João Pedro Borges Pereira, Waqas Imtiaz

January 19, 2017

Motivation

Prime numbers have been a mystery for centuries. Finding a pattern for these numbers has proven to be such a complex task that it has become the base of modern cryptography. To encode the data, people use 2 very large prime numbers, multiplying each other and then executing the remaining operations necessary to encode. In order to decode, the hacker would need to find the original primes that form the current number. However, no pattern has been found for prime numbers, making the task very difficult if the primes are relatively large. Nevertheless, a new pattern was discovered, which was shown to be related to the distribution of the differences between consecutive primes. These differences are called "prime gaps". The paper by Robert J. Lemke Oliver and Kannan Soundararajan [1] demonstrates that the pattern is related to the Hardy-Littlewood conjecture [2], from 1923, which gives an approximation to that distribution.

Being a pseudo-random function, we thought it would be interesting to investigate this matter, analyzing with our own means while confirming or rejecting conjectures done by mathematicians. You can look at a pseudo-random function in 2 different ways:

- You can consider it as a random function and try to analyze its distribution.
- You can check why it is not random by looking for patterns.

We adopted the first approach.

Since prime numbers are a very popular topic in pure mathematics, there are many theorems and conjectures involving them. The most important theorem for this work is the Prime Number Theorem [3]: it says that the number of primes up to x, $\pi(x)$, tends to $\frac{x}{ln(x)}$, for x large enough. Note that this number is also the number of prime gaps up to x, since each prime is followed by exactly one prime gap. Also, very important for this work is the Hardy-Littlewood conjecture, also called the k-tuples conjecture. It gives an estimation to the amount of prime constellations up to a number x. In particular, it says the amount of gaps of size $2k$ tends to $C(k) * \int_2^x \frac{1}{ln\check{s}(t)} \partial t$.. We thus formulated some hypothesis based on those two formulas:

1. The Prime Number Theorem [4] says that, over time, the frequency of prime numbers becomes lower. That would mean that the average prime gap becomes larger with a larger input.
2. More specifically, if we have a prime gap from a to b, we have that $\pi(b) - \pi(a) = 1$ and $b - a = gap$, so $\frac{\pi(b)-\pi(a)}{b-a} = \frac{1}{gap}$. The left side of that equation can be seen as similar to the derivative of $\pi(x)$. Since the Prime Number Theorem [5] gives the approximation $\pi(x) \approx \frac{x}{ln(x)}$, we have that $\frac{1}{gap} \approx \frac{1}{ln(x)} - \frac{1}{ln\check{s}(x)}$. But what is this $\frac{1}{gap}$? It is the average of $\frac{1}{gap}$ with the Waiting Paradox bias, or the inverse of the average of the gap with the bias correction. The first one is since considering $\frac{1}{gap}$ for each possible input between a and b, we get sum equal to the number of gaps between a and b, however since it is for each input we get more samples from larger gaps. The second one is due to that considering the gaps between a and b, the sum of their lengths is $b - a$, and so their average is $\frac{b-a}{\pi(b)-\pi(a)}$. That is, for inputs close to x, the average of the gaps should be approximately $\frac{1}{\frac{1}{ln(x)} - \frac{1}{ln\check{s}(x)}} = \frac{ln\check{s}(x)}{ln(x)-1} \approx ln(x)$.
3. We can similarly make the hypothesis that other statistical measures, such as the median and quantiles, also scale proportionally to $ln(x)$.

[1] https://arxiv.org/abs/1603.03720

[2] Hardy, G. H. and Littlewood, J. E. "Some Problems of 'Partitio Numerorum.' III. On the Expression of a Number as a Sum of Primes." Acta Math. 44, 1-70, 1923.

[3] http://mathworld.wolfram.com/PrimeNumberTheorem.html

[4] http://mathworld.wolfram.com/PrimeNumberTheorem.html

[5] http://mathworld.wolfram.com/PrimeNumberTheorem.html

4. The Lemke Oliver and Soundarajan [6] observation is that consecutive prime numbers have some correlation, so we decided to investigate if consecutive prime gaps are too. Our initial hypothesis is that there is not, since they are regarded as being very "random".
5. Derivating the Hardy-Littlewood formula [7], we get that the probability of a gap of size $2k$ is proportional to $\frac{C(k)}{ln\check{s}(x)}$. One hypothesis we can look into is that this is true, since the mathematical community strongly believes in the conjecture.
6. We can also go beyond checking the average number of gaps of a given size, we can ask if they also behave randomly. One hypothesis is that they follow a Poisson distribution.

Generation of the data

In this project we worked with very large integers. This is necessary to provide significance to sublinear factors - factor smaller than n, such as $log(n)$. To generate large integers we used the very popular GNU Multiple Precision library (GMP) with the C language. This library is used by Python, for example, to provide integers with arbitrary magnitude. The library implements the Miller-Rabin probabilistic primality test [8]. It uses it for a function called 'nextprime', that computes the smallest prime larger than the given input. By altering this function we can compute the 'previous prime'. The difference between these two primes is the prime gap surrounding the input. For example, if we take an input of value 4, we know that the previous prime of 4 is 3 and the next prime is 5. The difference (i.e. prime gap) between 3 and 5 is 2. This gap of 2 is surrounding the input of 4.

Generating prime gaps in this way can have the issue of the (Bus) Waiting Paradox. This so-called paradox is mostly known from Poisson distributions, but may affect any other distribution if the sampling approach is similar. It seems contradictory that if buses arrive in a Poisson distribution and the average time between them is 15 minutes, then the average time you must wait in a bus stop is also 15 minutes, regardless of how long ago the previous bus left. The contradiction is that the average time since the last bus is also 15 minutes. This means you observe an average gap of 30 minutes. This is explained by the fact that it is more probable to arrive at the bus stop during a long gap. This is a bias approach towards longer gaps. As an example, if one gap is 10 minutes long and another is 20 then the average gap is 15 minutes. However the average observed gap is 16 minutes and 40 seconds, since it is more probable for an observer to arrive during the longer gap.

We solve this problem during generation by adding an extra step to it. In this step, if a gap is found with size of $2k$ then it is accepted only with $\frac{1}{k}$ probability. This is because each gap of size $2k$ has $2k$ possibilities of getting selected. By adding an inversely proportional weight to it, we correct this bias. This bias could also be solved by considering that the factor has an importance weight, where a weight of 2 would mean the same as having a second identical observation. However, altering the generation was much simpler for us, and does not significantly impact the amount of time needed to generate the data. The GMP library and C language are very fast, originally outputting thousands of samples each second, so rejecting a portion does not impact the significance of our analysis. Note that doing this changes the probability distribution of the input, since, as we could see with the rest of our work, larger inputs have larger gaps. This means that weighting larger gaps down, or rejecting them randomly, means weighting down larger inputs more than smaller ones. Thus, it is very important to keep this bias in mind, as now the number of samples is proportional to the number of gaps, not the size of the input. We also generate the next gap, to test the hypothesis that there is no correlation between consecutive prime gaps. To generate uniform random numbers for inputs we use the GMP library functions. We focused on generating input numbers in a logarithmic scale. This is to be able to notice the impact of sublinear factors. The generation could be done by first generating a uniformly random real number and then using it as exponent to a fixed base. However, with very large numbers this is hard to do due to concerns of precision. We instead generate an integer exponent n, which gives a range of possible

[6] https://arxiv.org/abs/1603.03720

[7] Hardy, G. H. and Littlewood, J. E. "Some Problems of 'Partitio Numerorum.' III. On the Expression of a Number as a Sum of Primes." Acta Math. 44, 1-70, 1923.

[8] Rabin, Michael O. (1980), "Probabilistic algorithm for testing primality", Journal of Number Theory, 12 (1): 128-138, doi:10.1016/0022-314X(80)90084-0

numbers, from 2^n to $2^{(n+1)}$. We have that the probability of choosing a number x should be proportional to $log(x) - log(x-1) \approx \frac{1}{x}$, so we generate a uniformly random number in that range and accept it with $\frac{2^n}{x}$ chance.

Analysis of the data

In a first instance, we made a graph displaying the general distribution of the gaps of the prime number throught the whole input dataset and the result is the following:

```
library(ggplot2)
library(reshape2)
library(Hmisc)
```

```
## Loading required package: lattice
```

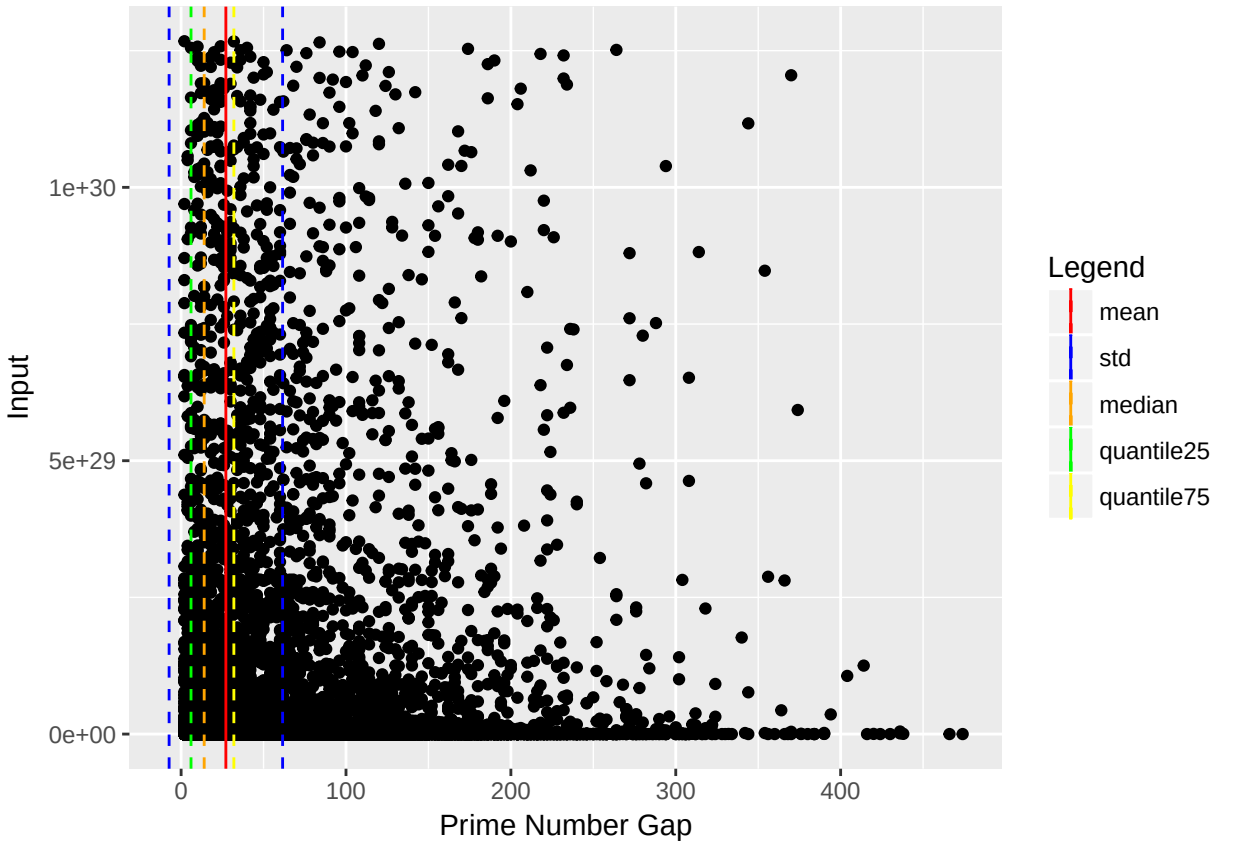
```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

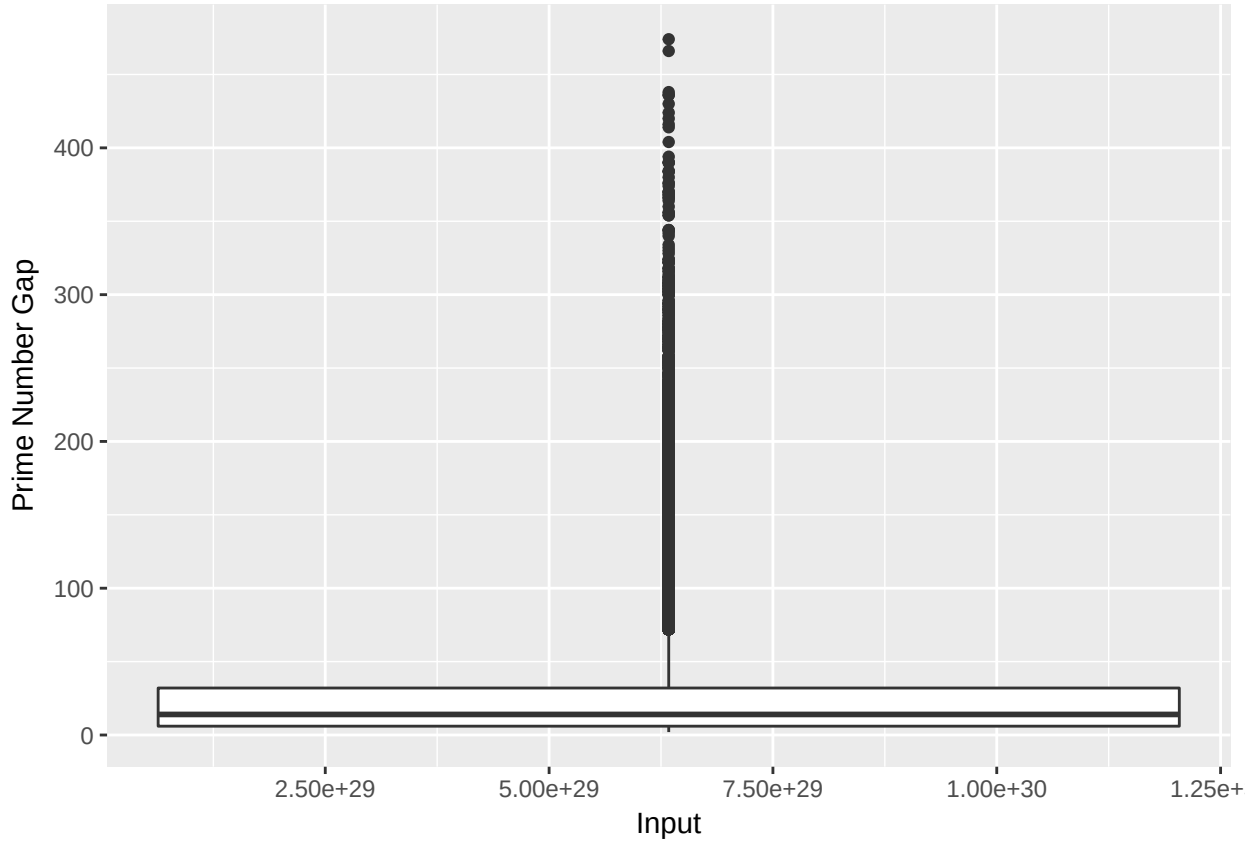
```
load(file="data/primeNumber.Rdata")
meanE <- mean(prime$PrimeGap)
std <- sd(prime$PrimeGap)
meanPlot <- ggplot(data= prime, aes(prime$PrimeGap, prime$Input)) +
  geom_point() + labs(x = "Prime Number Gap", y = "Input") +
  geom_vline(aes(xintercept = meanE, colour = "mean")) +
  geom_vline(aes(xintercept = (meanE + std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = (meanE - std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = median(prime$PrimeGap),
                 colour = "median"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(prime$PrimeGap, 0.25),
                 colour = "quantile25"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(prime$PrimeGap, 0.75),
                 colour = "quantile75"), linetype = "dashed") +
  scale_colour_manual(name = "Legend",
                      breaks = c("mean", "std","median", "quantile25", "quantile75"),
                      values= c(mean = "red", std = "blue", median = "orange", quantile25 = "green", qua
meanPlot
```



As you can see most of the gap values seem to be concentrated between 0 and 150. We then decided to draw a box plot with the overall average of the gap value.

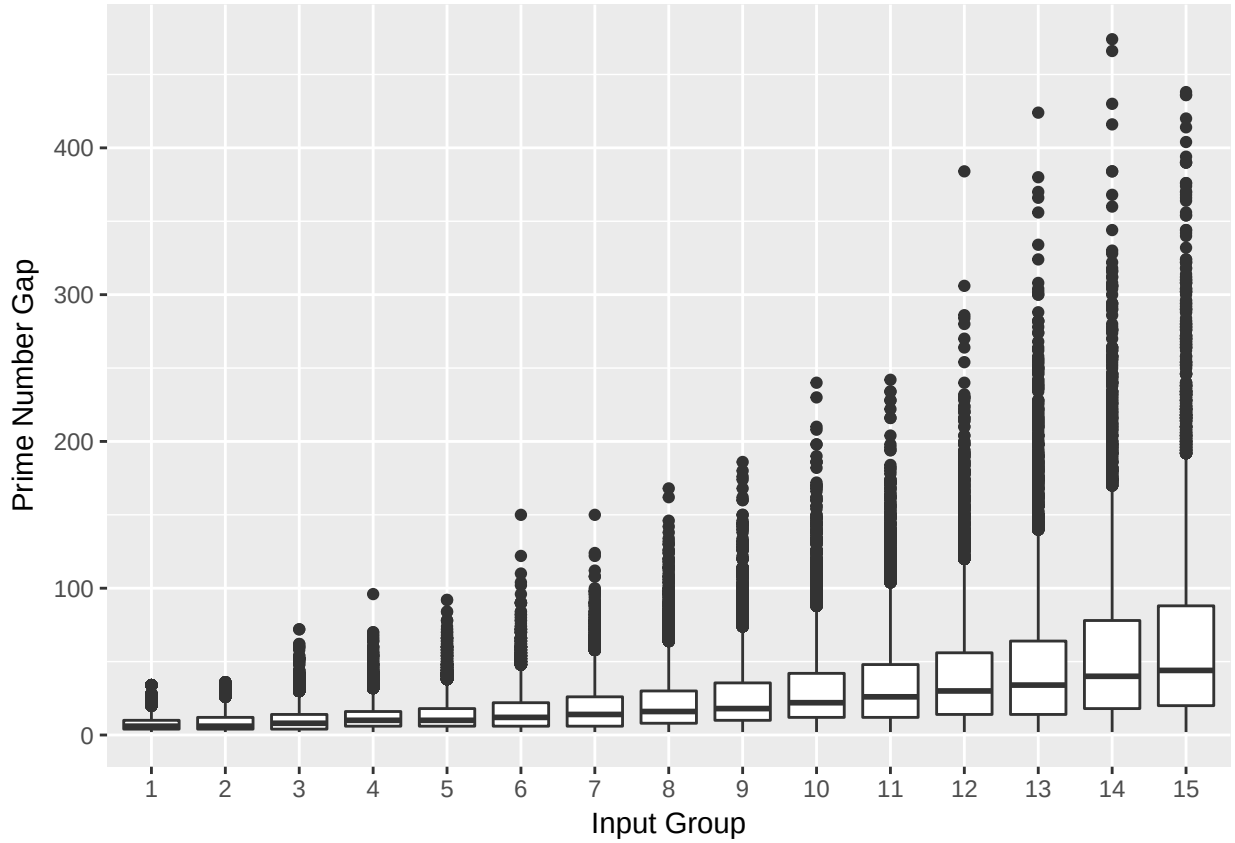
```
plot = ggplot(data= prime, aes(prime$Input, prime$PrimeGap)) +
  geom_boxplot() + labs(x = "Input", y = "Prime Number Gap")
plot
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



This seems to be consistent with the previous graph. Having drawn the box plot for the whole dataset, we then divided the latter into 15 equally distributed groups in terms of inputs values. Next, for each different group we drew a box plot similar to the one of the previous graph in order to confirm or reject hypothesis (1).

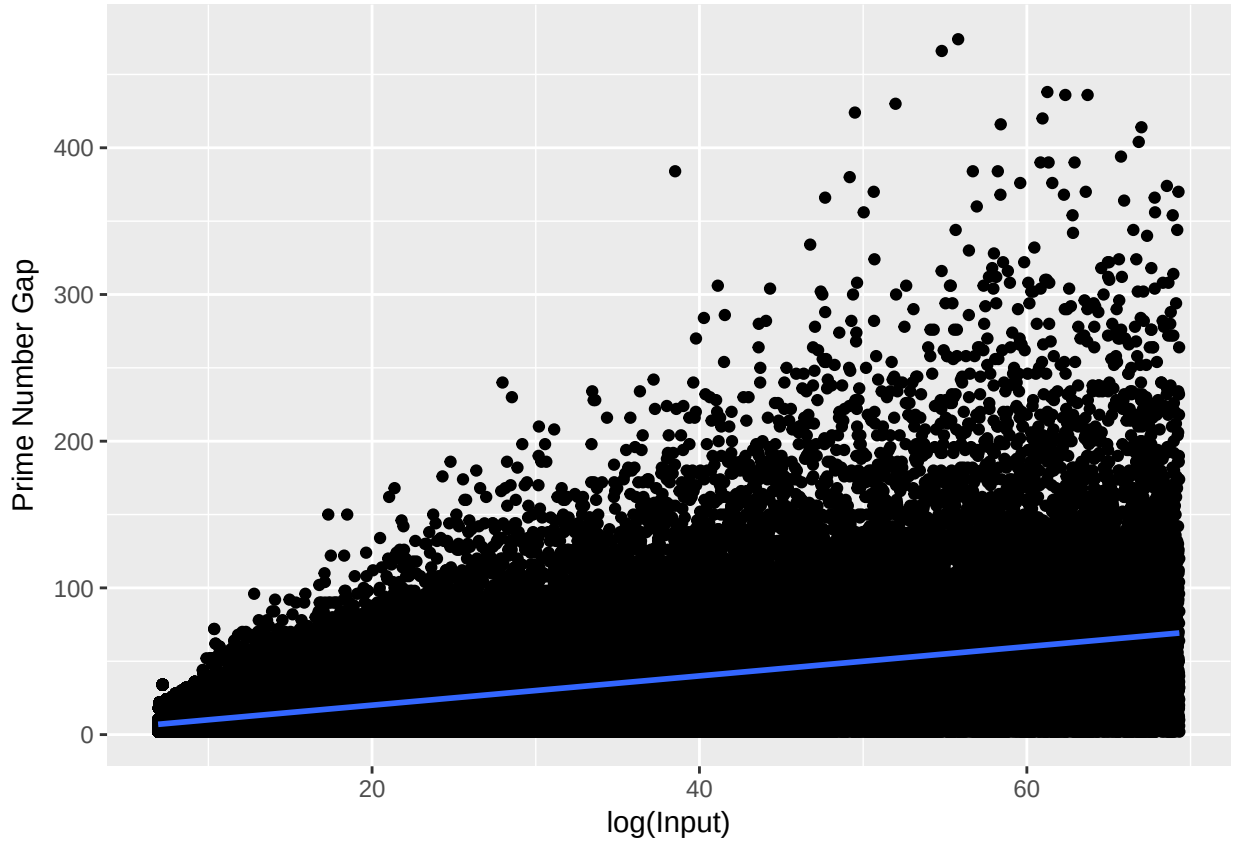
```
primeAux <- prime
primeAux$group <- as.numeric(cut2(primeAux$Input, g=15))
plot = ggplot(data= primeAux, aes(factor(group), prime$PrimeGap)) +
  geom_boxplot() + labs(x = "Input Group", y = "Prime Number Gap")
plot
```



As you can see, the value of the average seems to increase as we increase the group number, meaning that as we go into groups with larger numbers, the gap average increases. As such, we can confirm hypothesis (1) as true. However, since the x axis is proportional to the logarithm of the input value, we can see that the median and quartiles grow at a rate not linear in that logarithm. This goes heavily against Hypothesis (3), and so we can consider it to be very likely invalid.

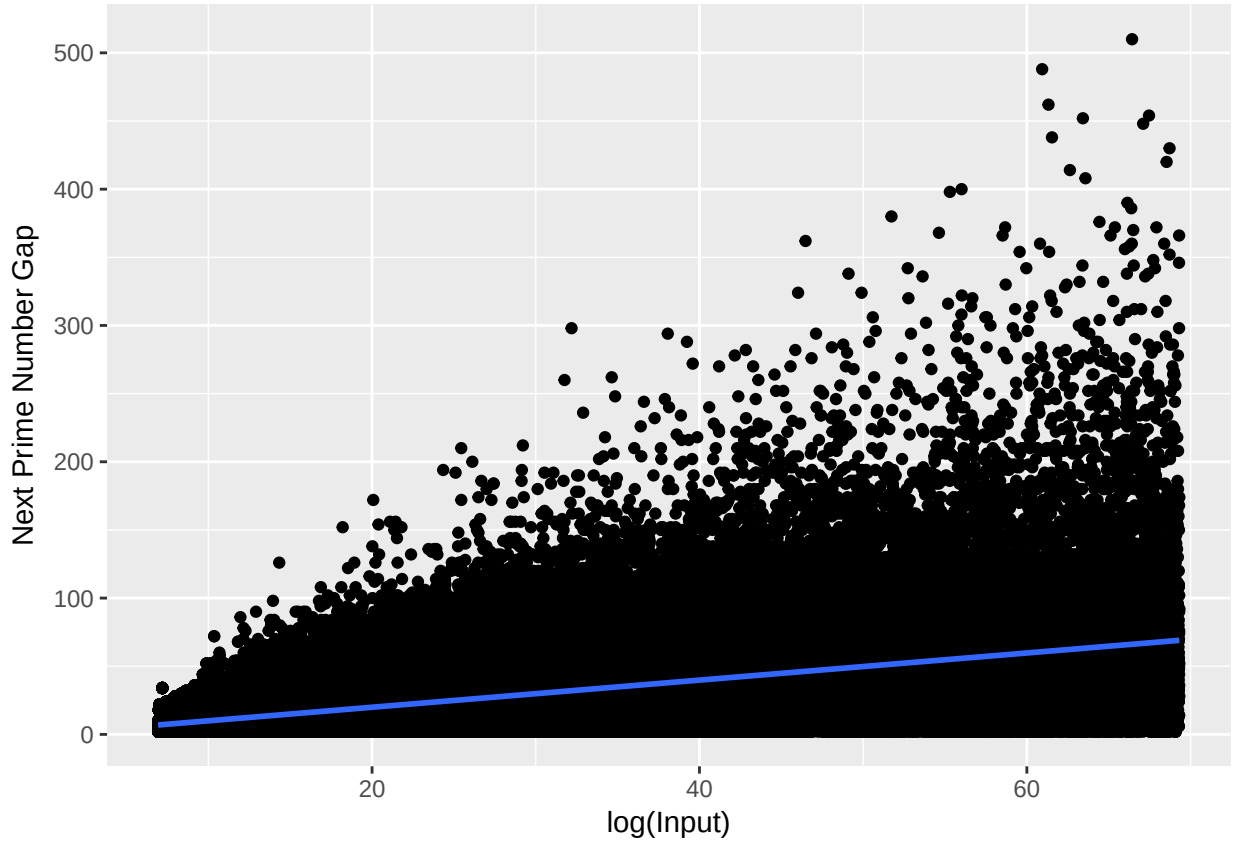
At that point, we started wondering if the gap values followed a logarithmic distribution. We started by ploting the $log(Input)$ against the prime number gap.

```
lnPlot <- ggplot(data= prime, aes(log(prime$Input), prime$PrimeGap)) +
  geom_point() + labs(x = "log(Input)", y = "Prime Number Gap") + geom_smooth(method = "lm")
lnPlot
```



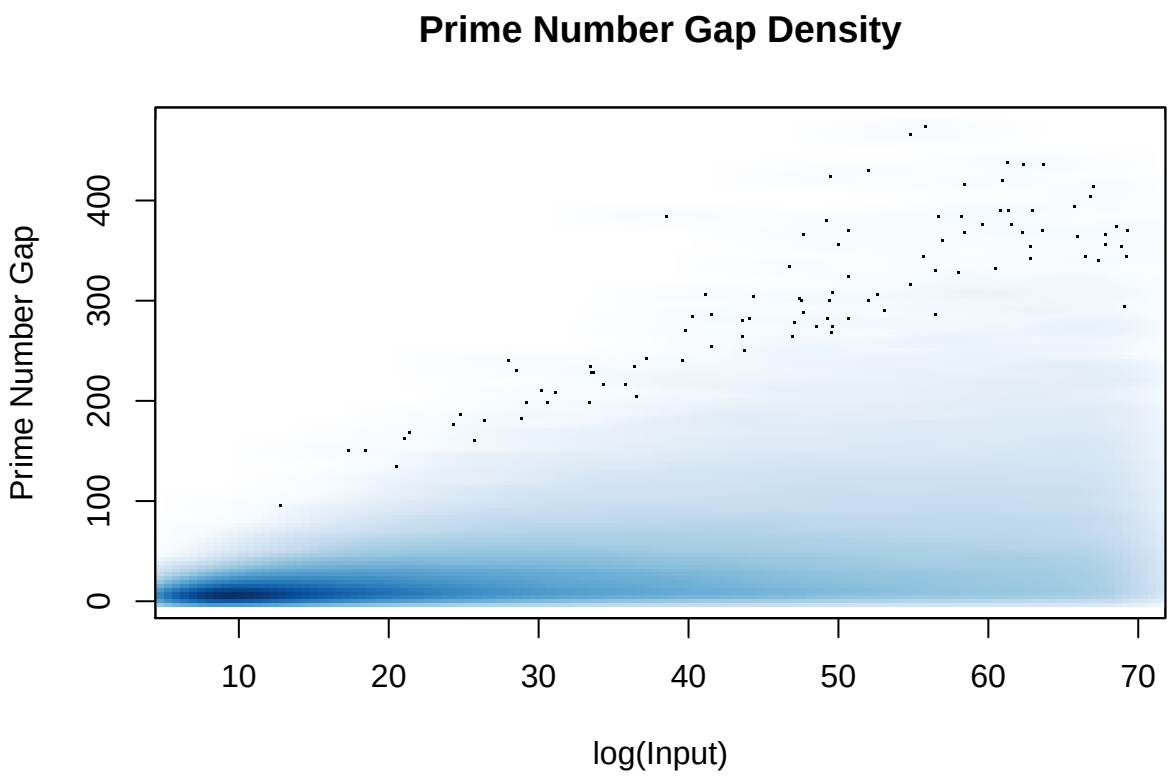
For redundancy reasons and to validate our data generation we ploted $log(Input)$ against the next prime number gap of the current input.

```
lnPlot <- ggplot(data= prime, aes(log(prime$Input), prime$NextGap)) +
  geom_point() + labs(x = "log(Input)", y = "Next Prime Number Gap") + geom_smooth(method = "lm")
lnPlot
```



Finally, we did a density plot to have a visual notion of the distribution of the gaps while still maintaining $log(Input)$ as the x axis.

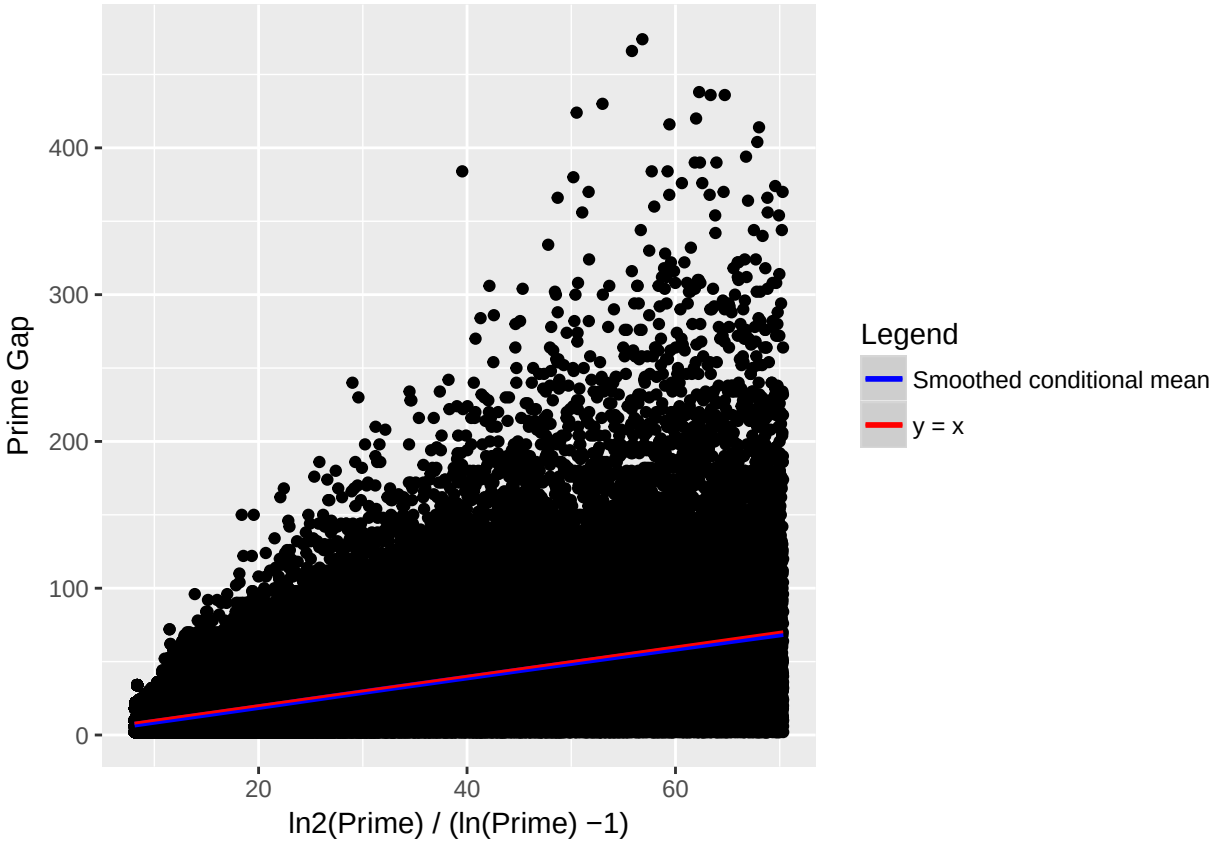
```
smoothScatter(log(prime$Input),prime$PrimeGap,
              xlab="log(Input)",ylab="Prime Number Gap",main="Prime Number Gap Density")
```



The blue line in the first 2 graphs show the smooth local average of the different data points. Both graphs seem to indicate the logarithmic distribution to be true and the density plot seem also to indicate a more concetrated number of gap in the lower values. To confirm the information conveyed by the first 2 graphs, we drew a 4^{th} one with the smooth conditional mean and the theoritical truth using the $\frac{log^2(x)}{log(x)-1}$ formula.

```
meanE <- mean(prime$PrimeGap)
plot = ggplot(data= prime, aes(prime$ln2x/(prime$lnx - 1), prime$PrimeGap)) +
  geom_point() + labs(x = "ln2(Prime) / (ln(Prime) -1)", y = "Prime Gap") +
  geom_smooth(aes(colour ="Smoothed conditional mean")) +
  stat_function(fun = function(x)(x), aes(colour="y = x")) +
  scale_colour_manual(name="Legend", values=c("blue", "red"))
plot
```

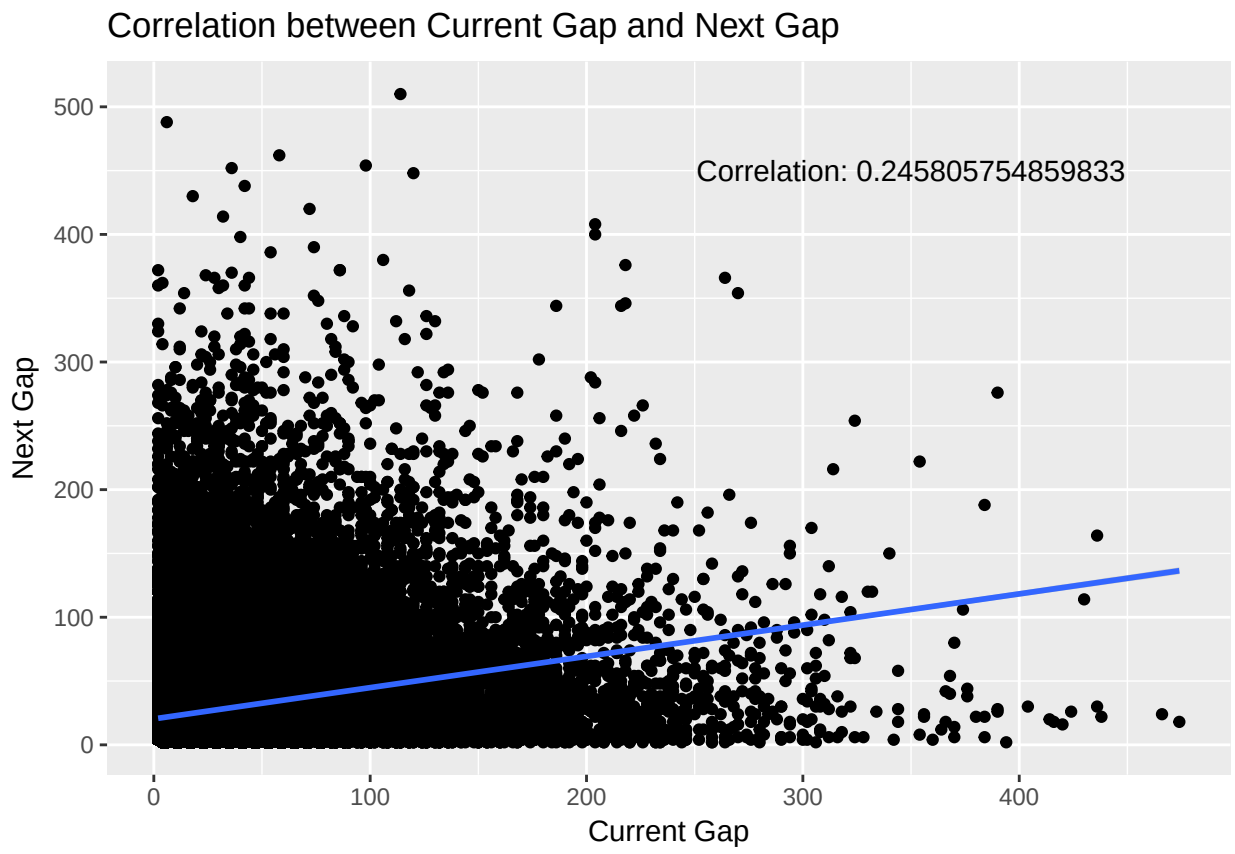
```
## `geom_smooth()` using method = 'gam'
```



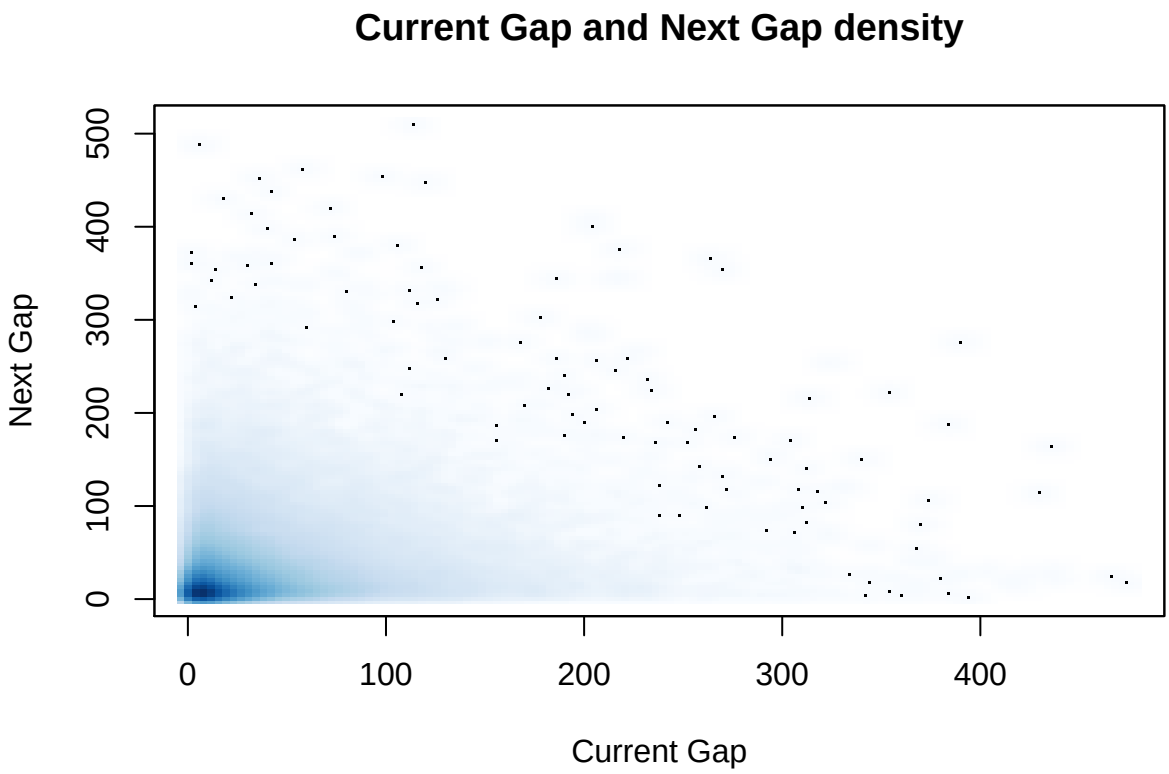
As we excepted, the smooth conditional mean alligns perfectly with the theoritical truth, meaning that our hypothesis (2) is confirmed as well.

Another hypothesis says that there is no correlation between consecutive prime gaps. We can use Pearson correlation coefficient to check the linear correlation between consecutive gaps. R provides a function to calculate the linear correlation, *cor*(). It takes two vectors as inputs and returns a value of linear correlation. The value of correlation coefficient varies between 0 and 1. The 0 value means that there is no correlation and 1 is otherwise. We input first and second prime gaps as first and second vectors, respectively. The return value (i.e. 0.24 as shown in graph) is equal to zero because of the following reason.

```
correlation <- cor.test(x<-prime$PrimeGap, y<-prime$NextGap, method = c("pearson"))
cor_graph <- ggplot(data = prime, aes(prime$PrimeGap, prime$NextGap)) +
    geom_point() + labs(x = "Current Gap", y = "Next Gap") +
    geom_smooth(method = "lm")
cor_graph + ggtitle("Correlation between Current Gap and Next Gap") +
          annotate("text", x = 350, y = 450, label = paste("Correlation:", correlation$estimate))
```




```
smoothScatter(prime$PrimeGap,prime$NextGap,
              xlab="Current Gap",ylab="Next Gap",main="Current Gap and Next Gap density")
```

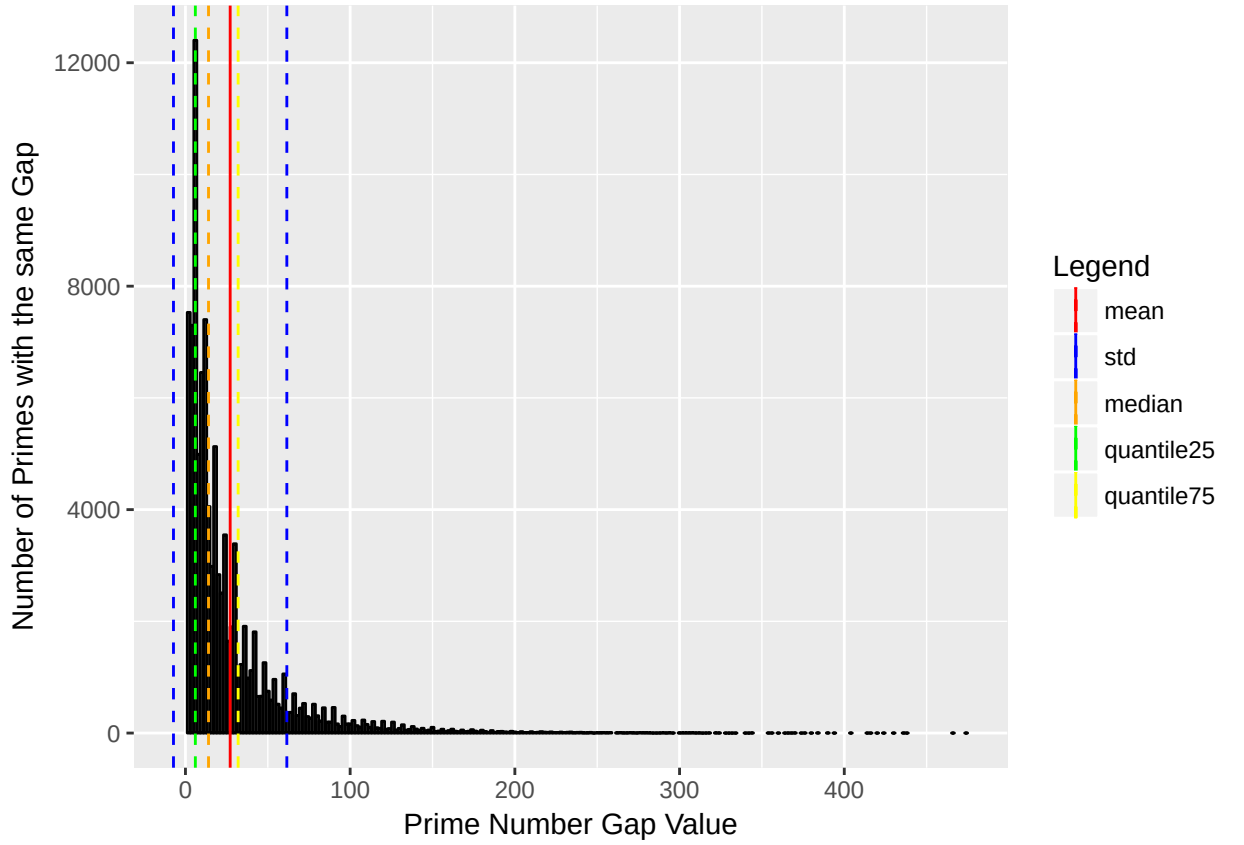


We found a plausible explanation for this fact in that we have larger gaps if the input is larger. That means that if the first gap is larger, the next gap has a higher probability of being relatively large too. Considering this, indeed we should expect some correlation. However, we could perform a separate experiment without this bias, by considering only similar inputs. But it was limited by time constraint. Hypothesis (4) is, thus, unconclusive due to lack of data.

Up until now, we analyzed the distrubtion of the gap values, but not their frequency. To do that, we ploted a histogram of each prime gap value in order to check their frequency throught the whole input dataset.

```
meanE <- mean(prime$PrimeGap)
std <- sd(prime$PrimeGap)
plotHist = ggplot(data = prime, aes(prime$PrimeGap)) +
  geom_bar(fill="white", colour="black") +
      labs(x = "Prime Number Gap Value", y = "Number of Primes with the same Gap") +
  geom_vline(aes(xintercept = meanE, colour = "mean")) +
  geom_vline(aes(xintercept = (meanE + std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = (meanE - std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = median(prime$PrimeGap),
                 colour = "median"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(prime$PrimeGap, 0.25),
                 colour = "quantile25"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(prime$PrimeGap, 0.75),
                 colour = "quantile75"), linetype = "dashed") +
  scale_colour_manual(name = "Legend",
                 breaks = c("mean", "std","median", "quantile25", "quantile75"),
                 values= c(mean = "red", std = "blue", median = "orange",
                           quantile25 = "green", quantile75 = "yellow"))
```

```
plotHist
```

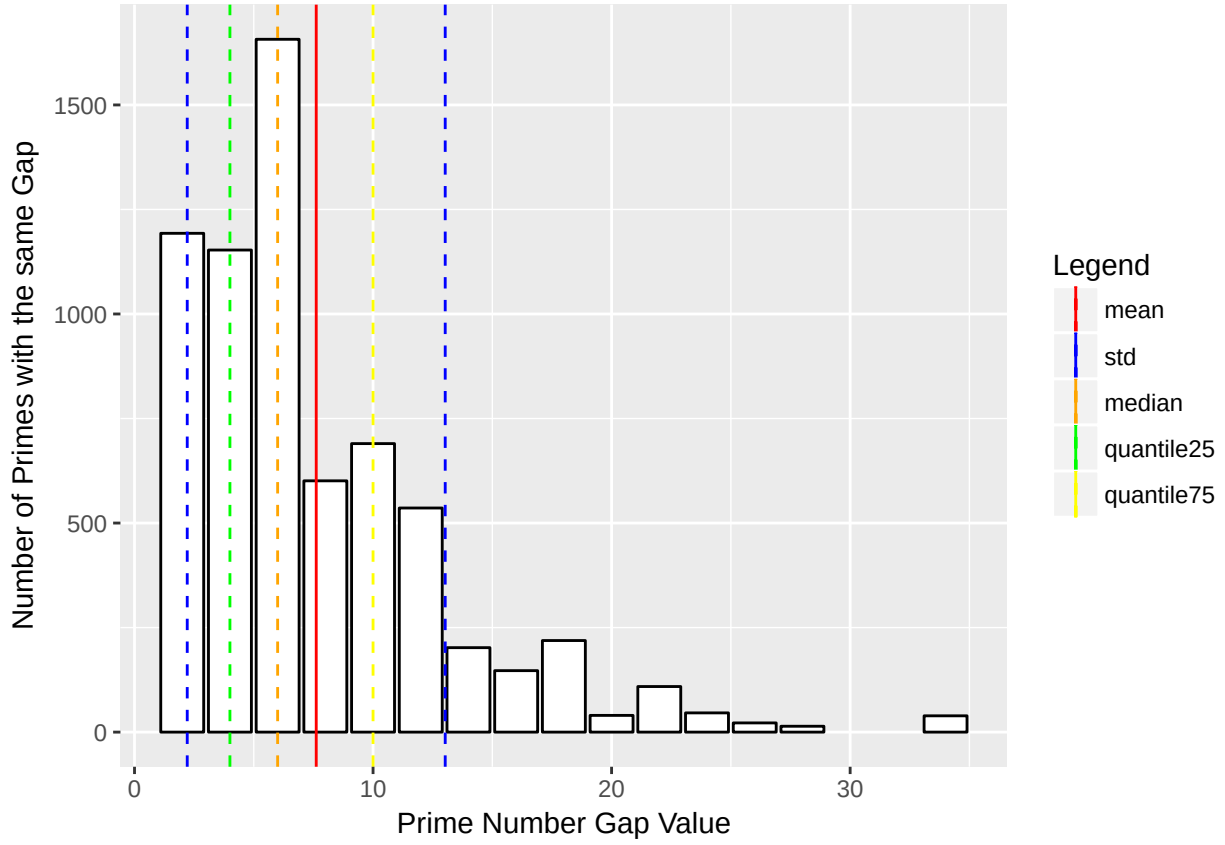


As we can see, the most common the lower gap value seem to be the most frequent ones. However, we pondered about the local distribution of the gaps since the graph shown here shows for the whole dataset. Therefore, we divided, again, the input numbers into 15 equally distributed groups in terms of inputs values. For each group we drew a graph similar to the previous one. The code for each plot remains the same, except the 3^{rd} line where you replace the value of `primeAux$group` by the number of the group desired.

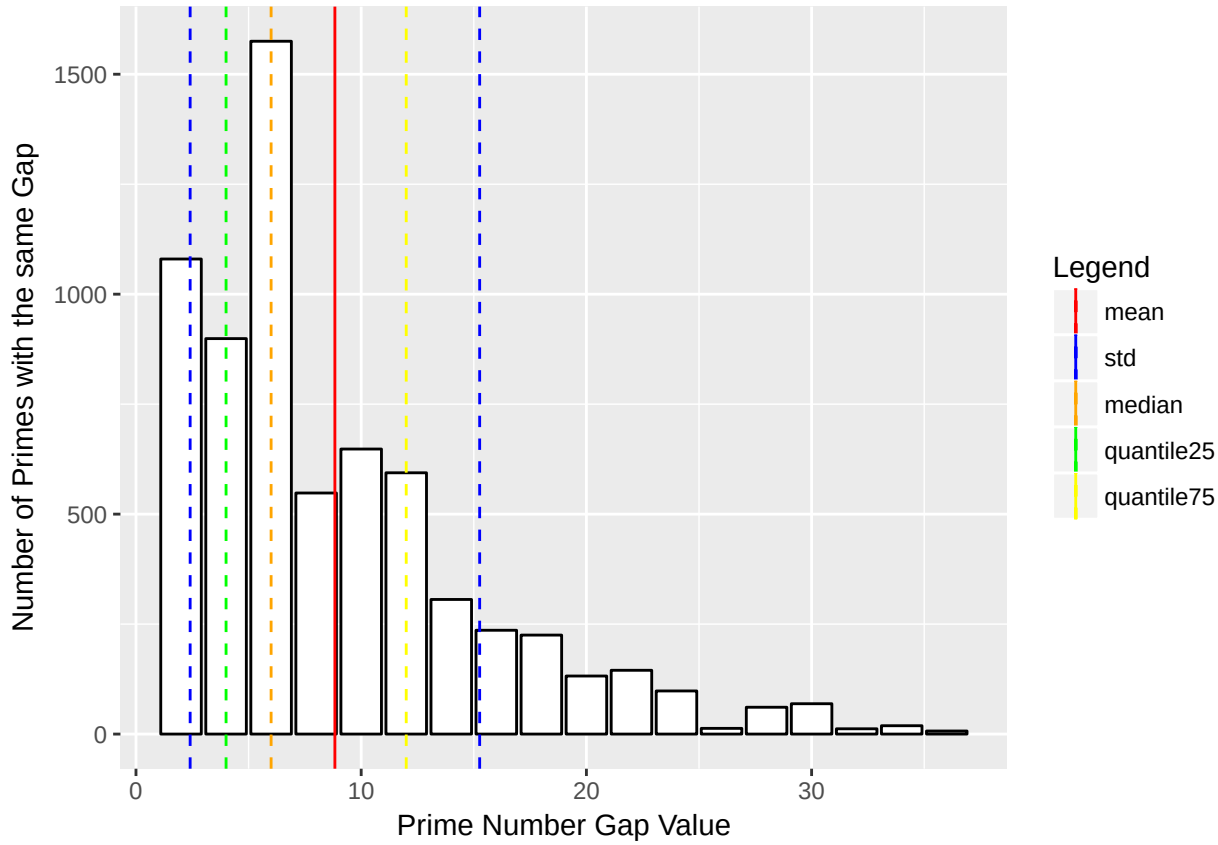
Group 1

```
primeAux <- prime
primeAux$group <- as.numeric(cut2(primeAux$Input, g=15))
primeAuxHist <- subset(primeAux, primeAux$group == 1)
meanE <- mean(primeAuxHist$PrimeGap)
std <- sd(primeAuxHist$PrimeGap)
plotHist = ggplot(data = primeAuxHist, aes(primeAuxHist$PrimeGap)) +
  geom_bar(fill="white", colour="black") +
      labs(x = "Prime Number Gap Value", y = "Number of Primes with the same Gap") +
  geom_vline(aes(xintercept = meanE, colour = "mean")) +
  geom_vline(aes(xintercept = (meanE + std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = (meanE - std), colour = "std"), linetype = "dashed") +
  geom_vline(aes(xintercept = median(primeAuxHist$PrimeGap),
                 colour = "median"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(primeAuxHist$PrimeGap, 0.25),
                 colour = "quantile25"), linetype = "dashed") +
  geom_vline(aes(xintercept = quantile(primeAuxHist$PrimeGap, 0.75),
                 colour = "quantile75"), linetype = "dashed") +
```

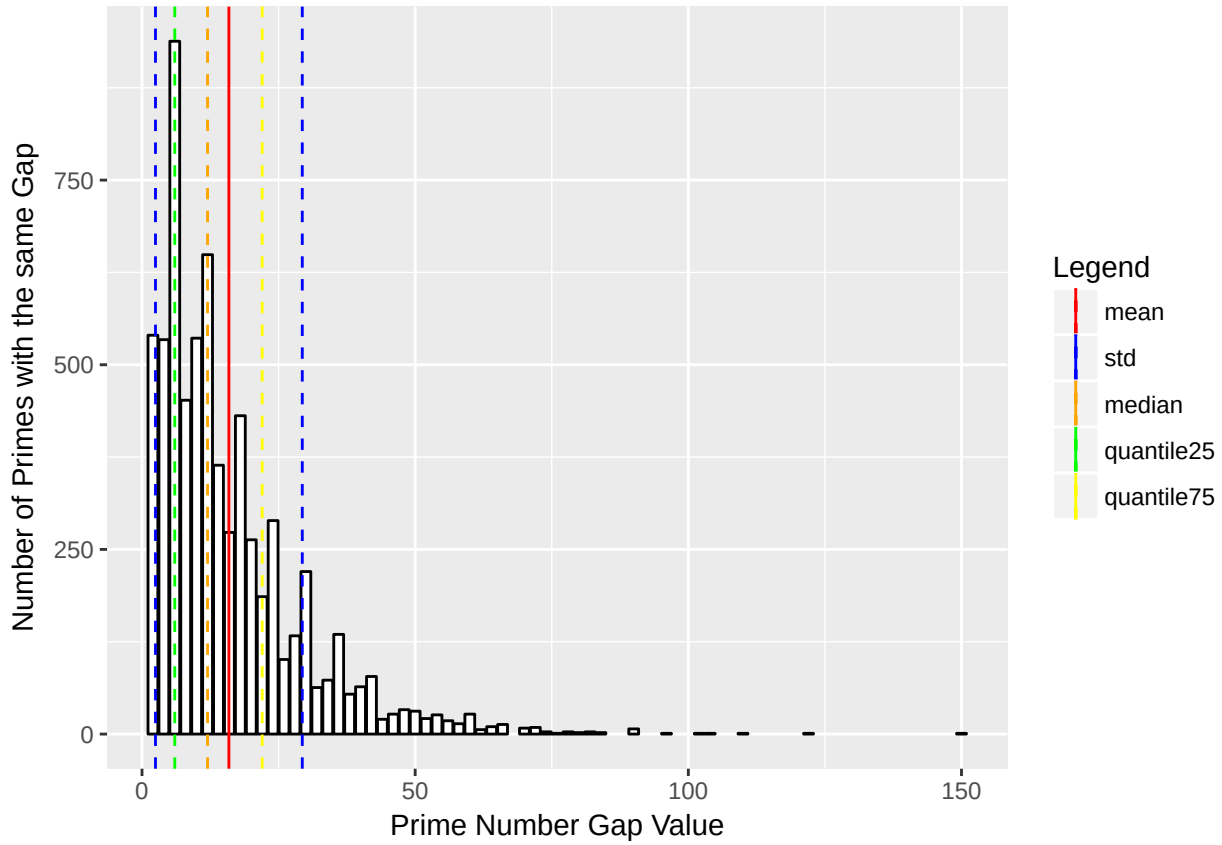
```
  scale_colour_manual(name = "Legend",
                      breaks = c("mean", "std","median", "quantile25", "quantile75"),
                      values= c(mean = "red", std = "blue", median = "orange",
                                quantile25 = "green", quantile75 = "yellow"))
plotHist
```



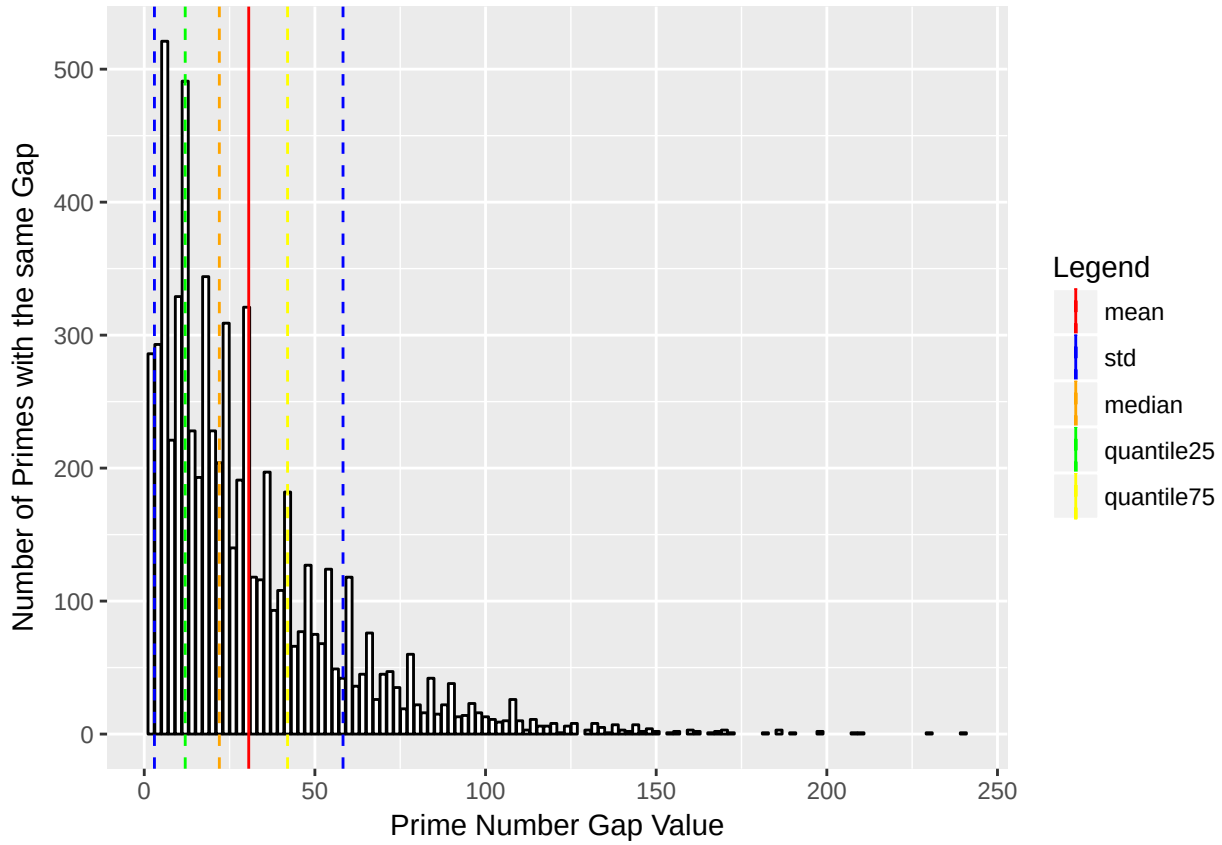
Group 2


Number of Primes with the same Gap
1500
1000
500
0
0
10
20
30
Prime Number Gap Value
Legend
mean
std
median
quantile25
quantile75

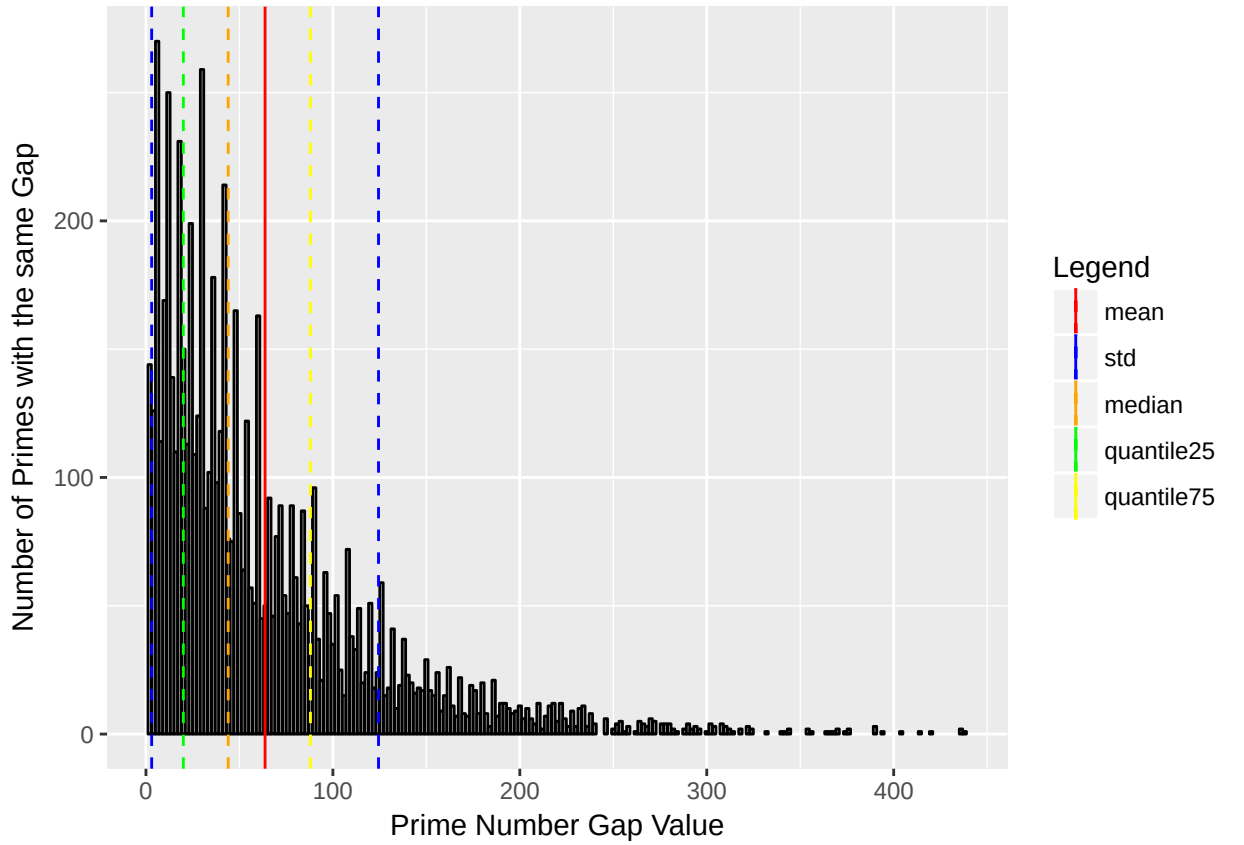
Group 6


Number of Primes with the same Gap
750
500
250
0
0
50
100
150
Prime Number Gap Value
Legend
mean
std
median
quantile25
quantile75

Group 10


Number of Primes with the same Gap
500
400
300
200
100
0
0
50
100
150
200
250
Prime Number Gap Value
Legend
mean
std
median
quantile25
quantile75

Group 15

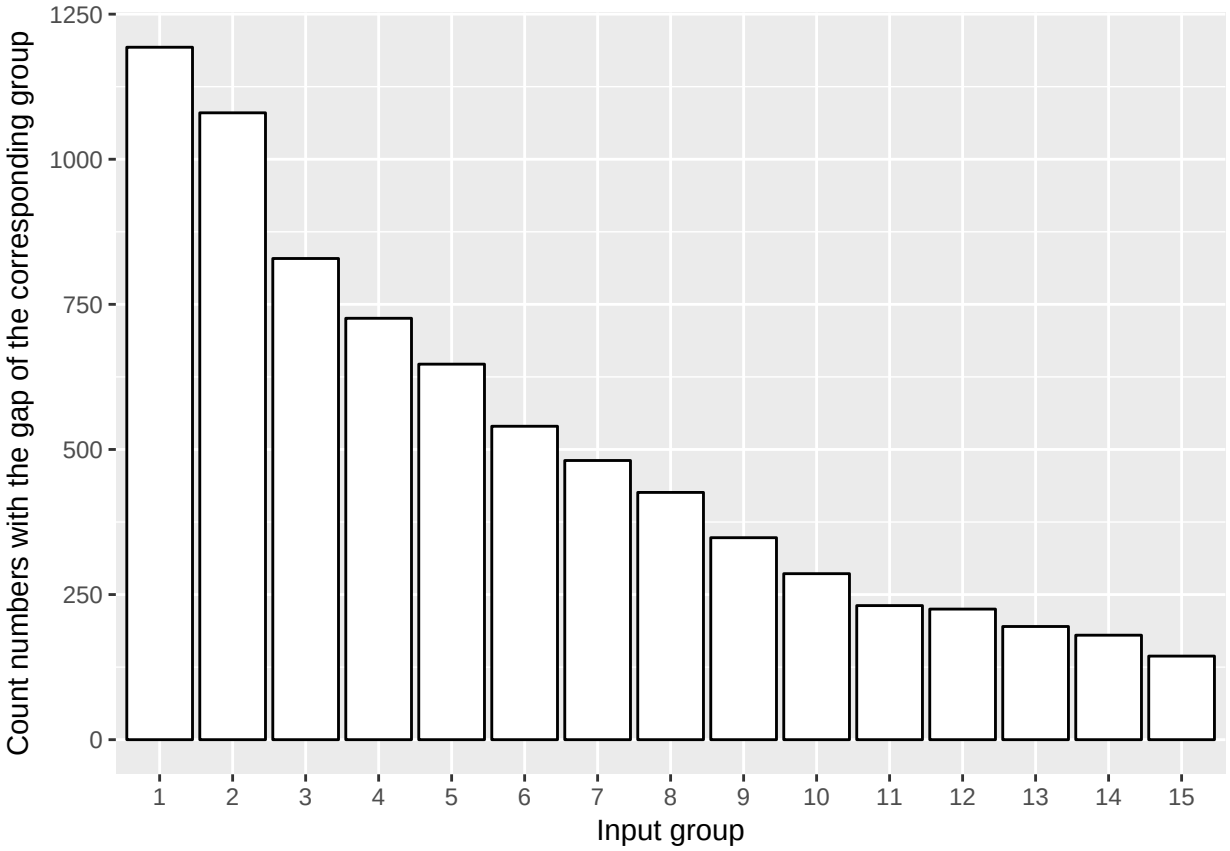


The results shown are quite curious; the graph always presents a long tail shape, however for lower input values we have a very smal variety of gap values, whereas for higher inputs values, the contrary is true.

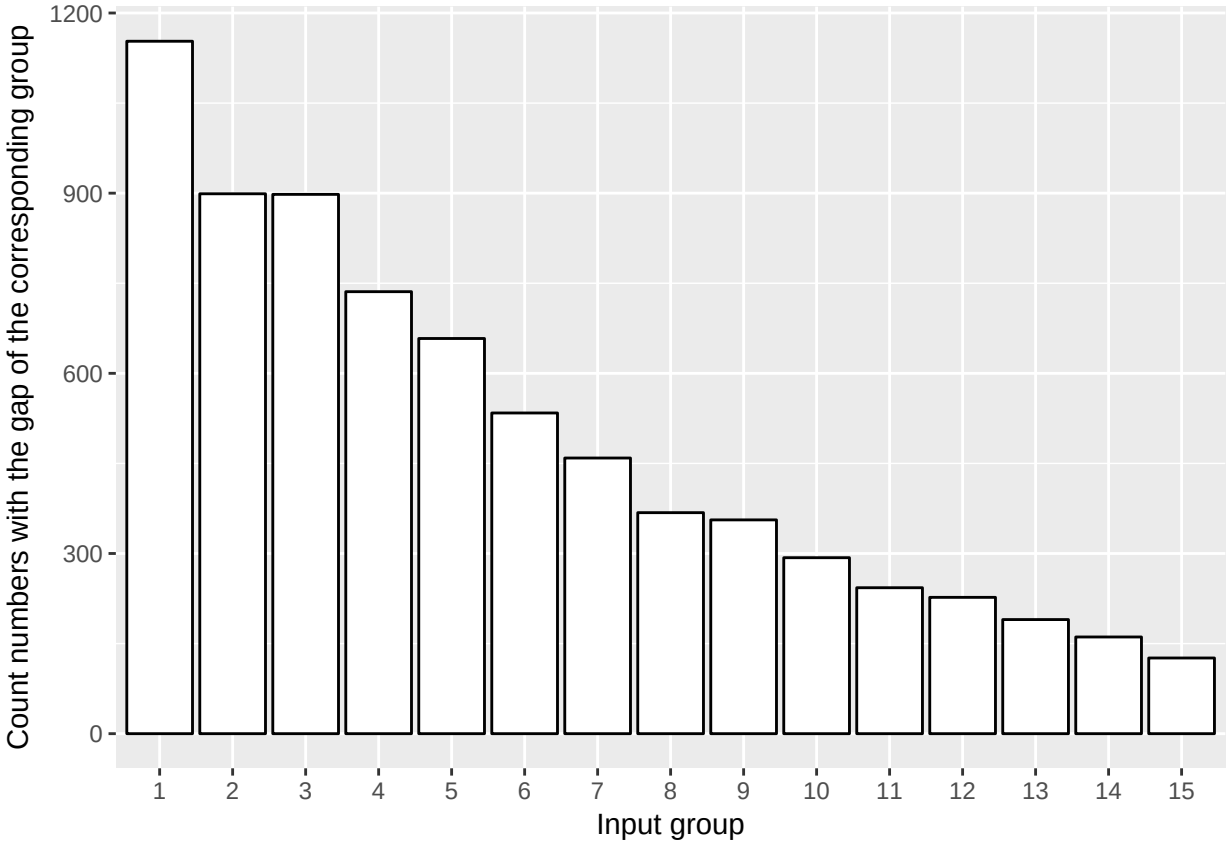
Nevertheless, even if we can conclude that lower gap values are more frequent, what would be the evolution of a specific gap value overtime? To do that, we not only divided, again, the input values into 15 equally distributed groups, but we also divided the gap values in very small groups. By fixing the gap group in `primeAux$groupGap` in the 3^{rd} line we obtain the following plots

$Gap = 2$

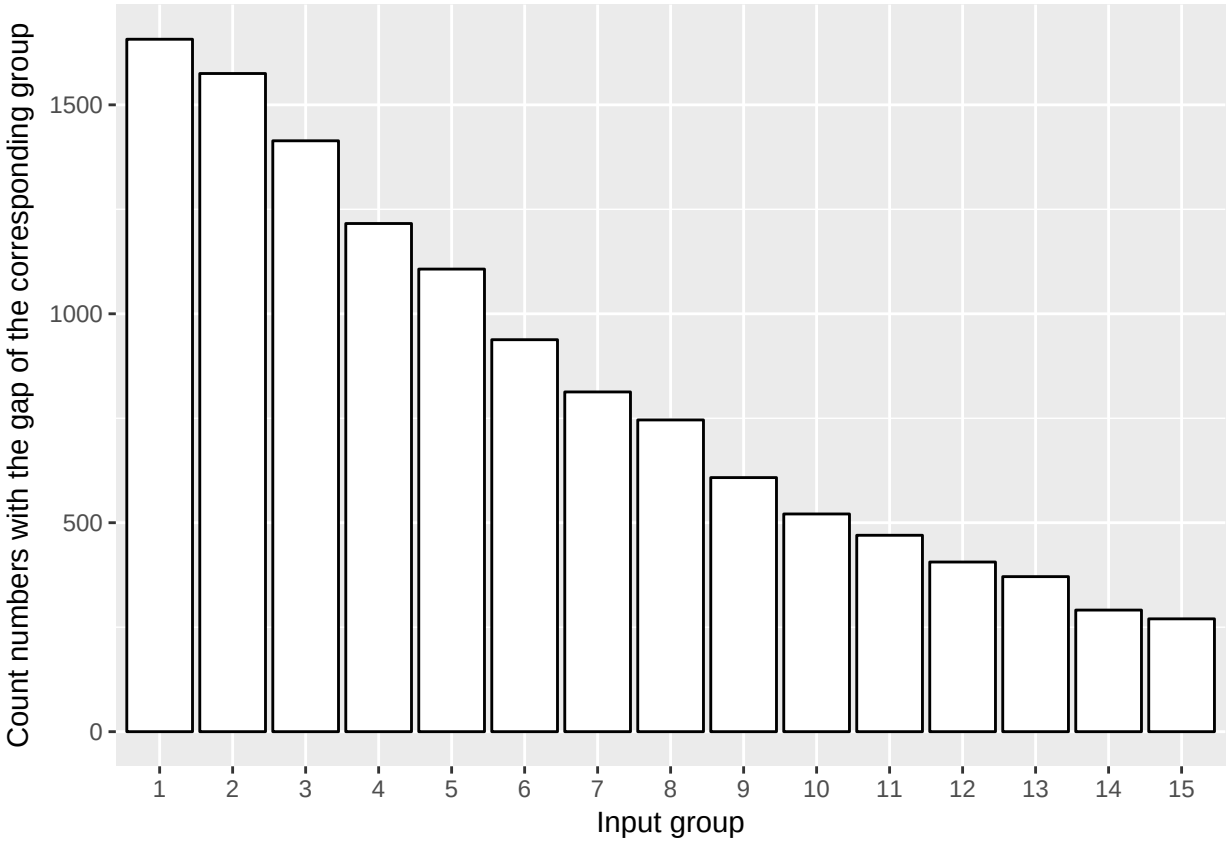
```
primeAux <- prime
primeAux$group <- as.numeric(cut2(primeAux$Input, g=15))
primeAux$groupGap <- as.numeric(cut2(primeAux$PrimeGap))
primeAuxHist <- subset(primeAux, primeAux$groupGap == 1)
plotHist = ggplot(data = primeAuxHist, aes(factor(primeAuxHist$group))) +
  geom_bar(fill="white", colour="black") +
      labs(x = "Input group", y = "Count numbers with the gap of the corresponding group")
plotHist
```


Count numbers with the gap of the corresponding group
1250
1000
750
500
250
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

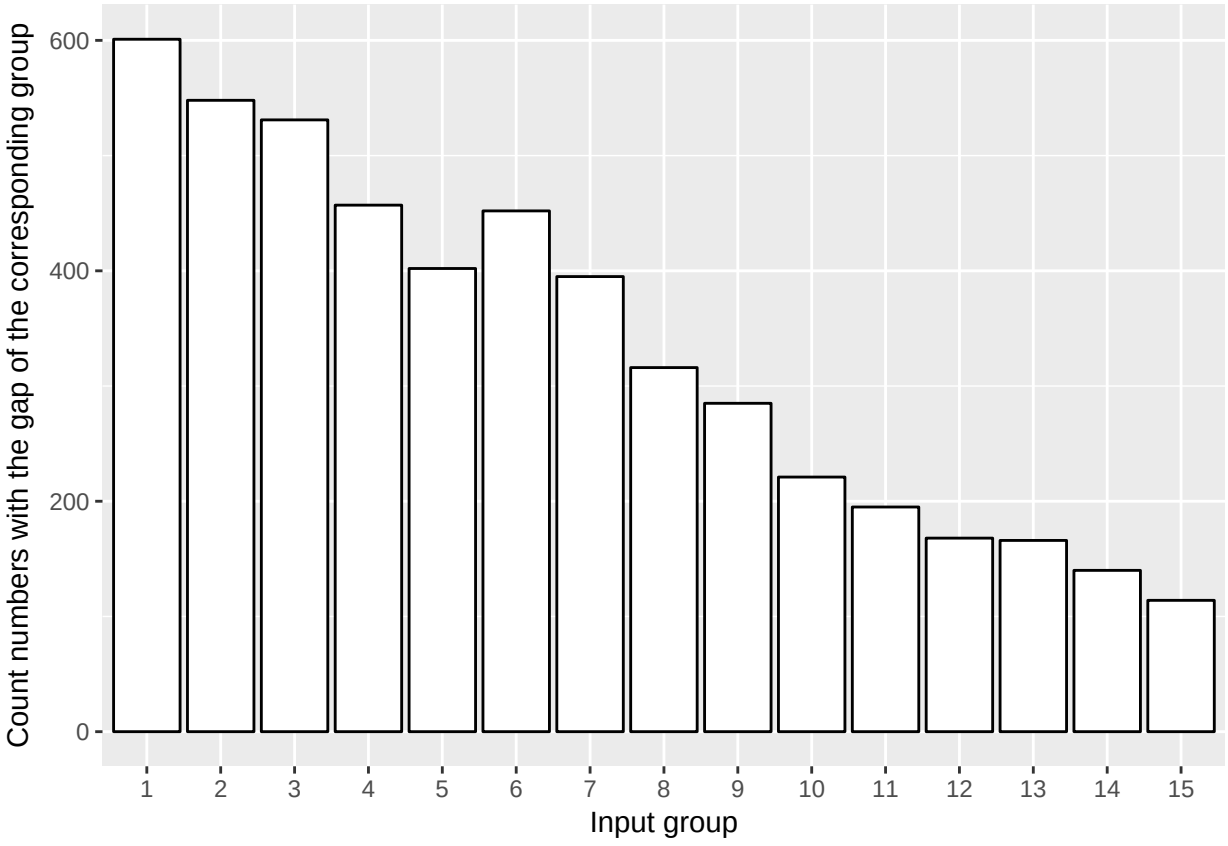
$Gap = 4$


Count numbers with the gap of the corresponding group
1200
900
600
300
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

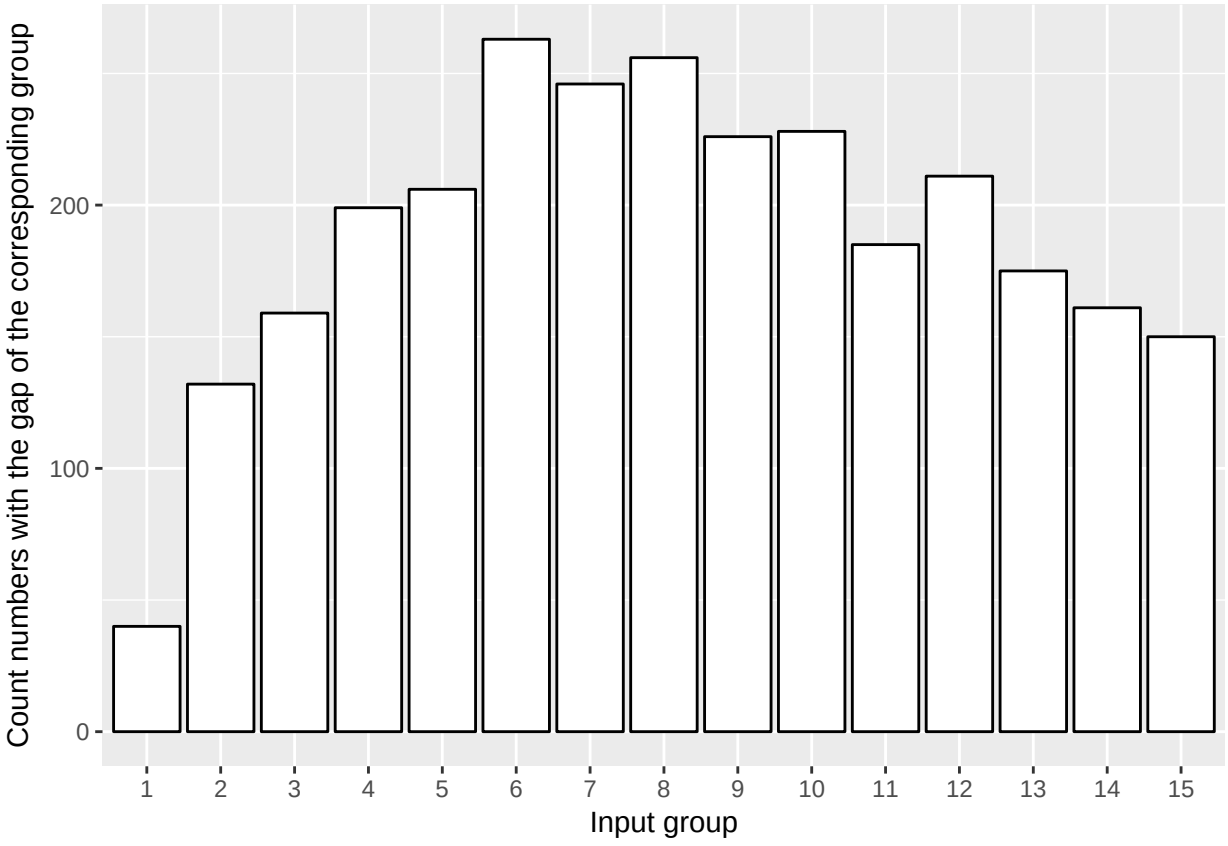
$Gap = 6$


Count numbers with the gap of the corresponding group
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

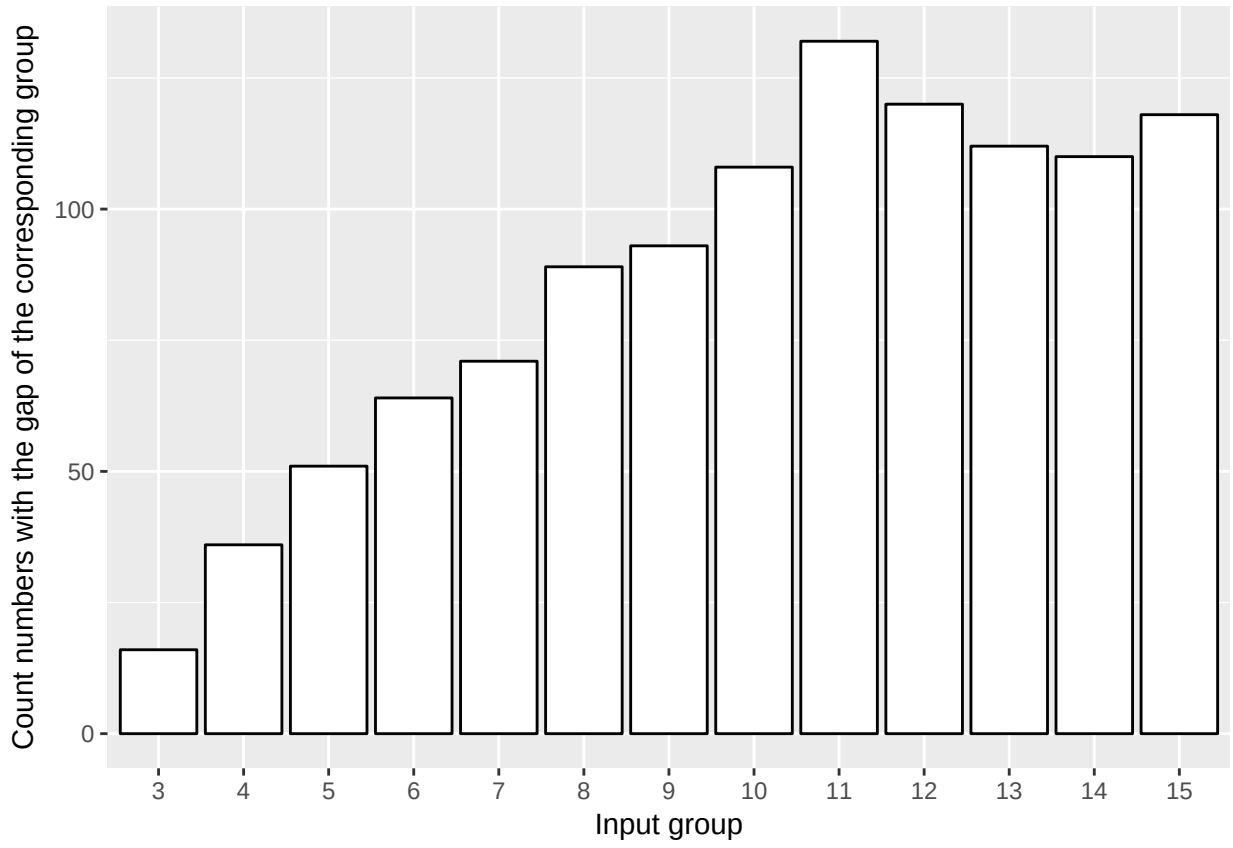
$Gap = 8$


Count numbers with the gap of the corresponding group
600
400
200
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

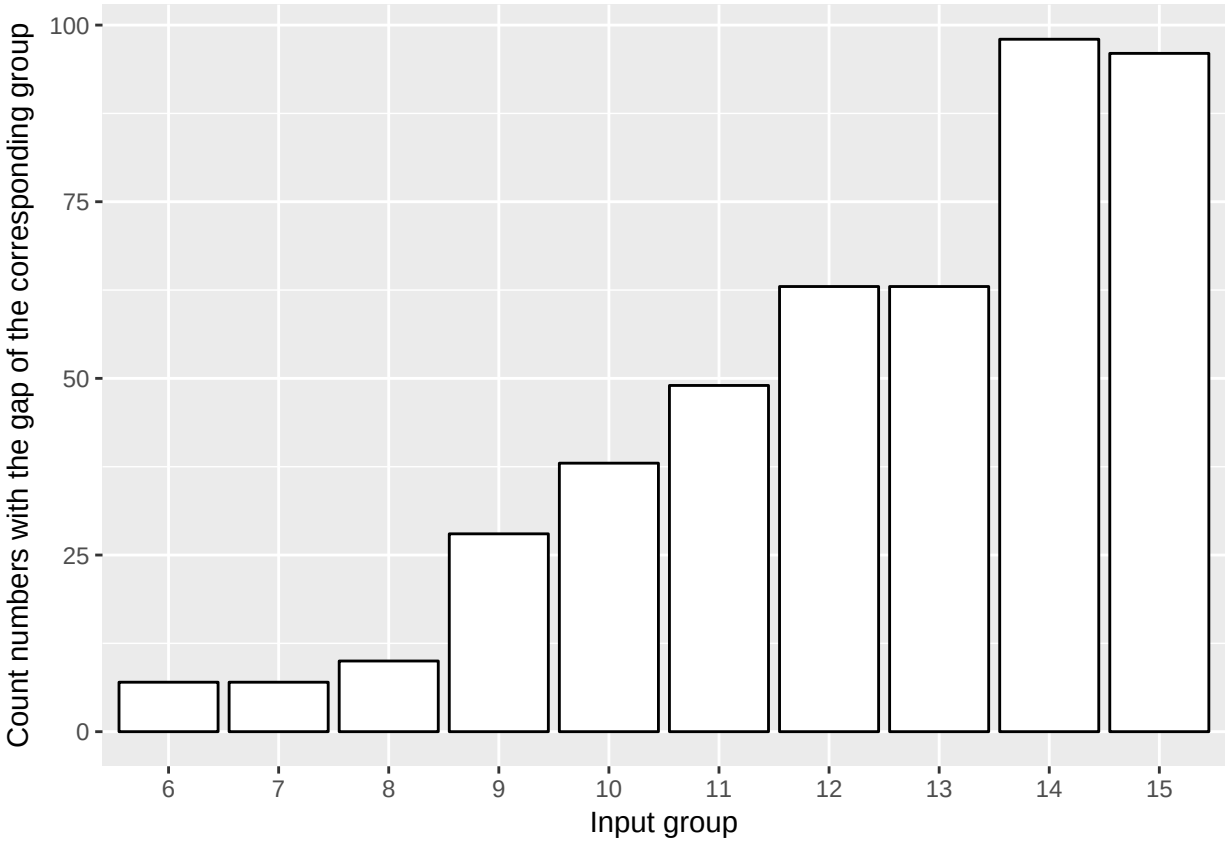
$Gap = 20$


Count numbers with the gap of the corresponding group
200
100
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

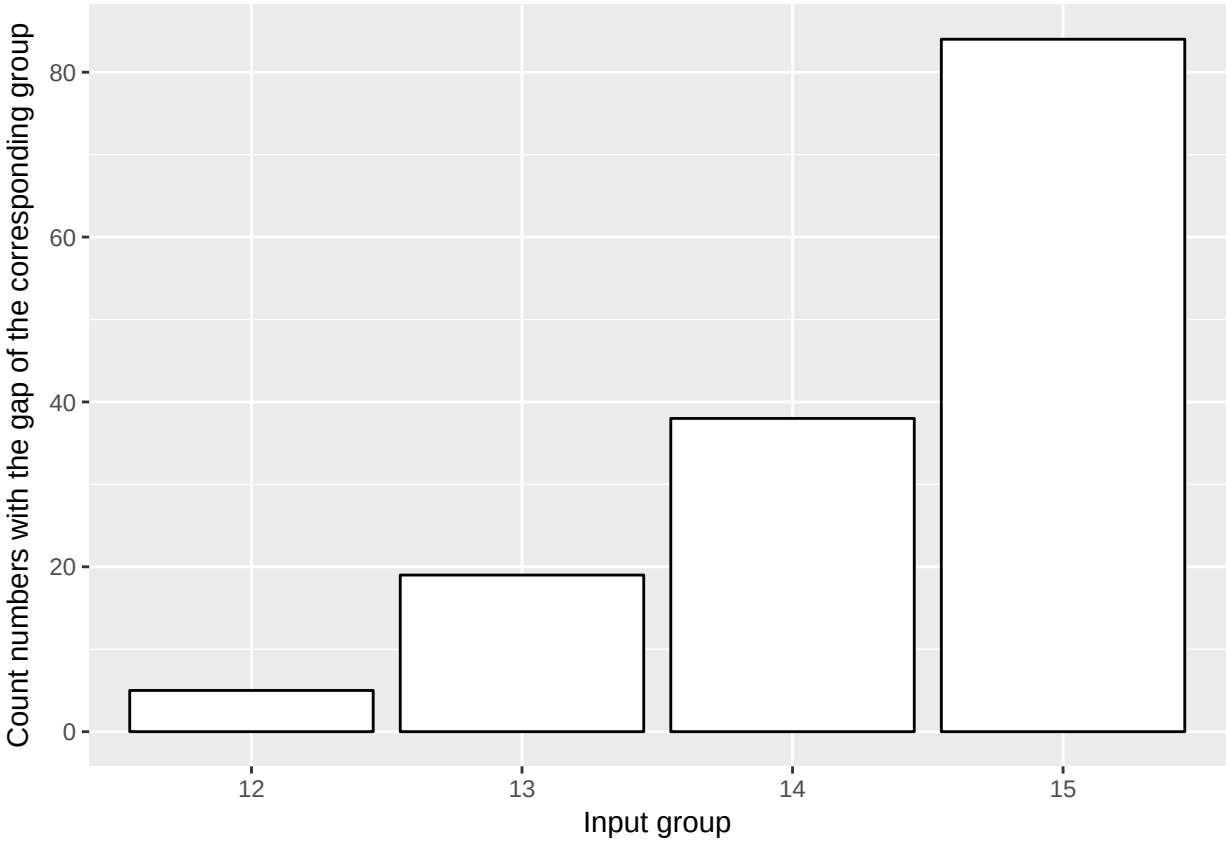
$Gap = 40$


Count numbers with the gap of the corresponding group
100
50
0
3
4
5
6
7
8
9
10
11
12
13
14
15
Input group

$Gap = 90$


Count numbers with the gap of the corresponding group
100
75
50
25
0
6
7
8
9
10
11
12
13
14
15
Input group

$Gap = [272; 474]$



Important to notice that as the gap value grows larger, the latter has no longer any occurences for the smaller groups and are, hence, not displayed. We can observe that, for large enough inputs, the frequencies start decreasing. This is expected, and explained by the Hardy-Littlewood conjecture [9]: as explained in the hypothesis (5), these frequencies are expected to become $\frac{1}{ln^2(x)}$. We can also note that, for larger gaps, the frequencies start low and have a build-up period before hitting their peak. The exact shape of this curve seems very interesting, but we did not find any theoretical results about it in our research.

Conclusion and Future Work

To sum up, we did an extensive analysis confirming 2 of our hypothesis and rejecting one. Important to mention that the $p - value$ for the correlation between consecutive gaps was $2.2 * 10^{-16}$, which is largely inferior to 0.005, meaning that we are almost 100% sure of our result. We made sure as well to generate enough data so that certainty of our results remains high. However, hypothesis (6), due to lack of time, was not explored and could be explored as future work as many other theories and conjectures not mentioned in this report. It would have been also interesting to run tests to validate our data mathematically, like the chi-squared test.

References

[9] Hardy, G. H. and Littlewood, J. E. "Some Problems of 'Partitio Numerorum.' III. On the Expression of a Number as a Sum of Primes." Acta Math. 44, 1-70, 1923.