UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FIME
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

PORTAFOLIO DE EVIDENCIAS

MODELOS PROBABILISTICOS APLICADOS

Encargada: Dra. Elisa Schaeffer.

Alumna: Mayra Cristina Berrones Reyes.

Matricula: 1646291

Practice 1 - Basic concepts of probability.

Environment statistics

Mayra Cristina Berrones Reyes

September 7, 2020

1 Data description

The data used for this work comes from a government page called INEGI [] that according to their web site is an "autonomous public body responsible for regulating and coordinating the National System of Statistical and Geographical Information, as well as for capturing and disseminating information on Mexico regarding the territory, resources, population and economy, which allows to publicize the characteristics of our country and help decision making" []. Here we selected the subject of environmental data to draw some important information with basic concepts of probability.

Among all of the subjects inside the environmental material, we decided to work with the topic of water[1], and see how it is used across all 32 federative states encompassed in Mexico.

Taking into consideration the pros and cons of using arrays or lists, we demonstrate two problems to see if this features are correct.

1.1 Palindrome

For this first example we have the problem of a palindrome. Remembering the differences we described in Table 1 we have that arrays are more well suited for this type of problem, since we can search and compare the contents of the first and last item on the array using the the index numbers, as we can see in Figure **??**. Then in Figure **??** we can see that we will have to pass the list several times to check between the first and the last items on the list, since each element is pointing to the direction were the next item is stored (it is not a fixed parameter).

[1] https://www.inegi.org.mx/temas/agua/

Table 1: Main differences of lists and arrays in python.

Lists	**Arrays**
• Lists can be can be flexible and hold arbitrary data.	• Arrays need to be imported from other libraries.
• They are a part of the python syntaxes so they do not need to be declared first.	• Arrays can not be resized, it needs to be copied to another larger array.
• Can be resized quickly in a time efficient manner. Python initializes some elements in the list at the initialization.	• Arrays can only store items with values of uniform data types.
• Lists can hold different types of data.	• They are specially optimized for arithmetic computations.
• Mathematical functions can not be directly applied to all the list, it needs to be applied to each item individually.	• Since arrays stay the size that they were initialized with, they are compact
• They consume more memory as they are allocated a few extra elements to allow for quicker appending.	• In the array you can find an item easily by their index.
• To search for an item, you need to start from the first element, and go through all of the other items until you reach the one you want.	• Operations like delete and insertion take a lot of computational time.
• Delete and insertion are easy.	

It took me a while to grasp the concept of this practice, because I am very used to handle python, and every time I saw something referenced as a list or an array, I thought they meant the same. Now, after reading more about the rules, I figured that I have been using lists too liberally, and that it may not seem very significant the computational time, because despite their drawbacks, both structures perform very fast, but I realize now that some of the problems I had in the past, programing simple things that resulted in errors or leaked memory could have been because I was not using the proper structure to store my data.

Practice 2: Gutenbergr project.

Frequency in texts

Mayra Cristina Berrones Reyes 6291

September 15, 2020

1 Introduction

The Gutenberg Project is an online library that offers its users free access to more than 60,000 free books in different formats that go from ebooks, html, plain text, etc. It began with creator Michael Hart in 1971 at the Materials Research Lab at University of Illinois.

All of the books listed on this site follow the public domain copyrighted work of the USA, which was originally 14 years after publication. They work to their eventual goal of providing public domain editions after a short time of publication [1].

2 Books

Jane Austen (16 December 1775 — 18 July 1817) was an English novelist, acclaimed to this day for her various novels, that make use of irony, humor, and realism, giving us a peek of social commentary of her era, with subjects that expose the dependence of woman on marriage for the pursuit of a favorable social standing in 18th century society.

In this practice we take two of the her most famous works and earliest publications. *Pride and prejudice* [3] and *Sense and Sensibility* [4] published in 1813 and 1811 respectively.

3 Data exploration

As for both of the books used in this practice, we are very familiar with their plot, having read them several times, seen the movies and series adaptations. But for this first plots, we focus on the more frequent words and letters to see if there is a discernible pattern that tells something related to the plot of both books.

Using the library of `gutenbergr` we downloaded both books to different variables and with the help of other libraries such as `dplyr` and `tidytext` we organized some of its information. In the case of sub figure 1 we started with the comparison of the most frequent letters. In both sub figures 1a) and 1b), we had to put a threshold of more than 10 on the frequency, because all of the

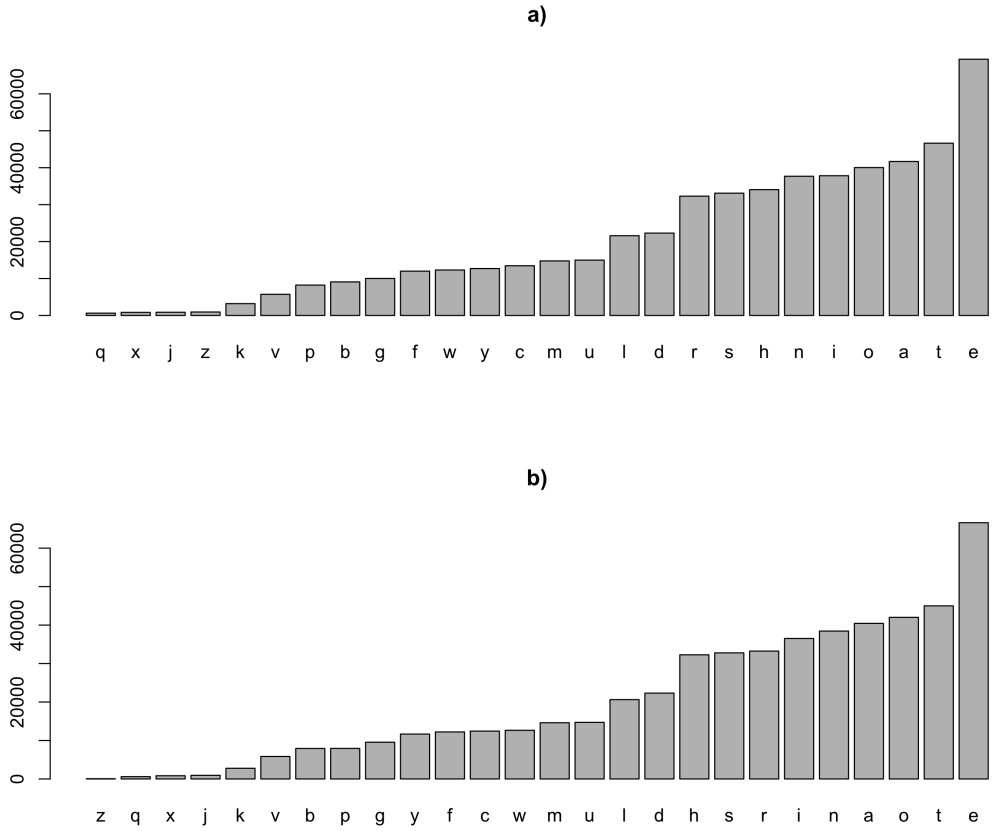


Figure 1: Data exploration by letters of the book Pride and Prejudice (sub figure a)) and Sense and sensibility (sub figure b)) by Jane Austen.

occurrences bellow that were only numbers. We suspect that they represented the chapter numbers. In both books, the placement of the vocals on the frequency is almost identical, and there are only a few swaps of positions on the consonants. The reason of the similarity can be because both are written in English and are from the same author, so it tends to have similar forms of expression.

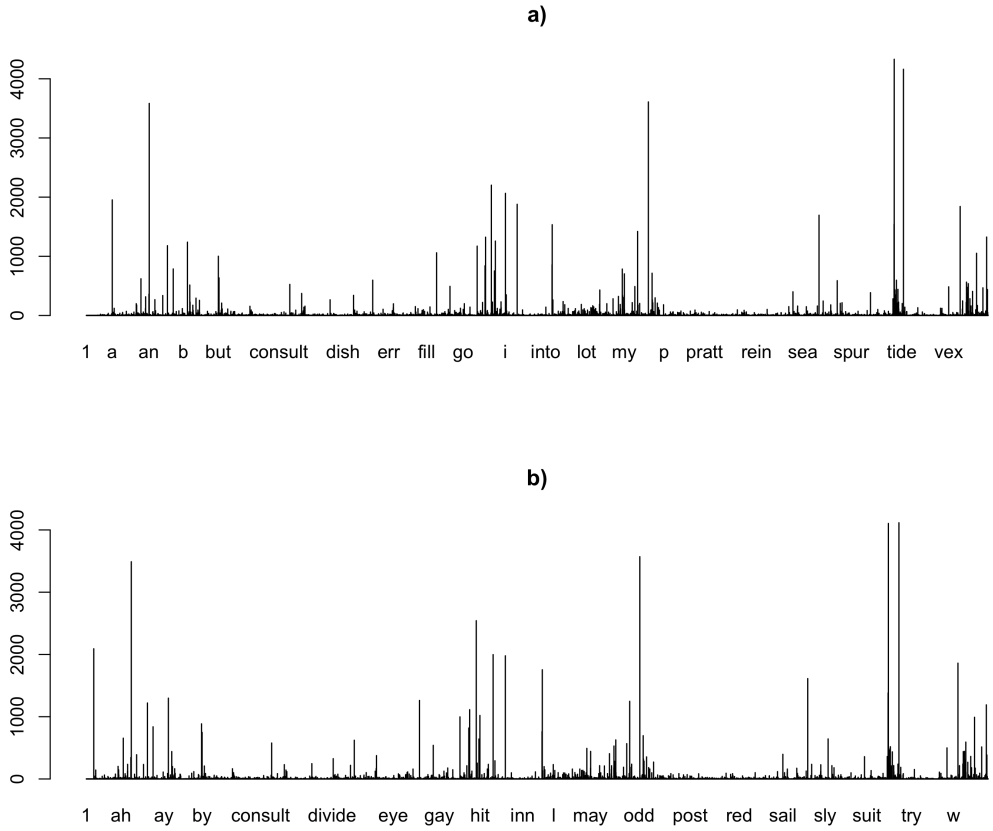


Figure 2: Data exploration by words of the book Pride and Prejudice (sub figure a)) and Sense and sensibility (sub figure b)) by Jane Austen.

We move on to see the frequency of the words. In this case, in figure 2 we expected some similarities on the most frequent words but without giving extra instructions to order them and filter the ones with low frequency, it is not very clear the information that we can gather from this plot.

Something that jumped out while reviewing figure 2 was that the sub figure 2b) had the word *gay* in it. As mentioned before, both books are personal favorites, so it was a little strange that this word was here, because no character was described in that way on the Spanish version. So after some research we found out that the word *gay* in the context of the 13th to 18th century meant something different than what it means in present day. Back then it meant happy or bright

and lively looking. In a video made by PBS [2] it exposes that the word *gay* was later adopted as the meaning of same sex attraction because the other word to describe it, homosexual, was meant with a bad psychological connotation, and the word gay was a way to avoid the criminalization of same sex relationships.

In a not so serious article some people speculate that Jane Austen used this term deliberately, mainly because some of her social commentary on her books where often obscured, and also because by the time she published her books, the use of the word gay was slowly catching up to the general public.

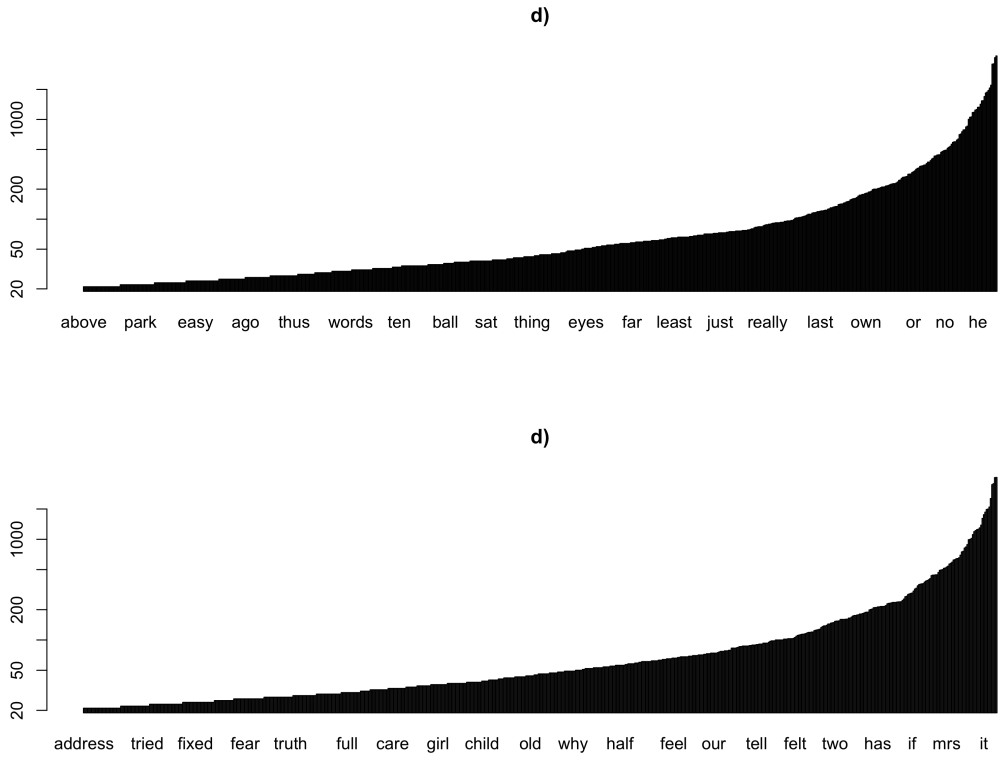


Figure 3: Data exploration by words, ordering the data from lower to higher and filtering frequency to more than 20 occurrences. Pride and Prejudice (sub figure a)). Sense and sensibility (sub figure b))

To be able to watch properly the distribution of the frequency on the words, we ordered them from lowest to highest number, taking out all the words with less than 20 recurrence in the data. Then, because the highest words where mostly connectors such as “the, a, as, for, by,” etc. we used the `log` value to even out the data, so it can be distinguishable. We can appreciate the results on figure 3.

In sub figure 3a) there is a variety of words. For someone who has not read the book, it

may be hard to make sense of it, but analysing the word, *park* is interesting because almost a third of the books, we talk about great states that the main male characters have or are related somehow, for example, Rossings, Pemberly, Longbourn, Netherfield and Haye-Park (all of them fictional). The importance of the places are there to remained the reader that this are not only very wealthy man, but also how it will benefit our protagonists to end up with one of them.

The word *ball* also stands out, because, as our protagonist Elizabeth Bennet often states, the best way to meet people and encourage affection is by dancing with said partner. In the structure of the books, the balls take such importance, because each builds a big event for Elizabeth. Meeting Darcy for the first time. Being rejected by him in a rude manner. Getting close with Mr. Wickham. Each dance she forms different opinions on Darcy, that help her build a rather misguided opinion on his future husband.

The word *eyes* just sparkle joy, because one can assume is there because Darcy keeps talking about Elizabeth's eyes for half of the book.

As for sub figure 3b) there is a combination of words that immediately steers attention to the main plot. In this case, the combination of the words *girl, child,* and *truth* could be alluding to the scandal on Willoughby side of the plot, where he neglects to tell the sister's protagonist Marianne that he has an illegitimate child that he has not claimed, while he is courting her. This is a big part of the plot, that helps Marianne grow from idealistic romantic to a more mature young lady.

The other word that it is very interesting to see on this plot is *old*, because one of the main reasons Marianne seems reluctant to accept the affections of Colonel Brandon is that he is 34, and she is 16. It can also allude to the fact that our protagonist Elinor is reaching an age in which it is deemed a bit old not to be married, and it is constantly reminded of that by Mrs Jennings.

Tired, fear, and *care* can also be attributed to the side of the plot in which the older half brother of the Dashwood sisters is often neglectful in taking care of them and his step mother, because of the poisonous advice of his wife, Fanny.

4 Plot comparison

This two books where chosen to be analysed because of its many similarities on their plot. In both cases we have a main protagonist that has only sisters, and is at the mercy of a good marriage to salvage the situation, that in that period was that a woman could not hold possessions, let that be money or state. In the case of Elizabeth, the absence of a brother leaves them at the mercy of their cousin, Mr Collins. And in Elinor case, they are dependent on their half brother, who resents his sisters because they were closer with his father.

They both have a favorite sister, Jane and Marianne respectively. And all of them have different good prospects for marriage that get complicated as the plot advances and their low social status gets in the way of forming a relationship.

Having established that, the next plots show the influence that the important characters have by measuring the frequency that they appear in the story.

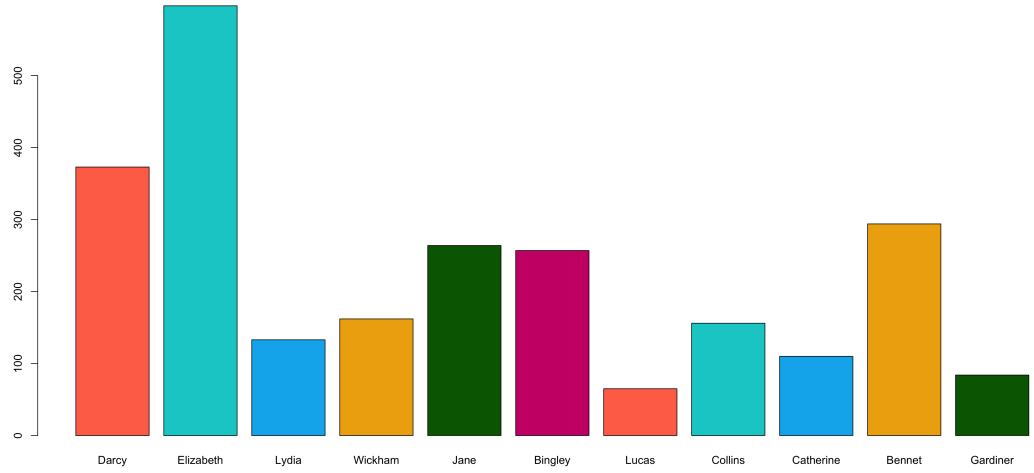


Figure 4: Bar plot of the frequency of characters in the book Pride and prejudice

In figure 4 we quickly realize that Elizabeth is our main protagonist. Darcy is the main love interest, and Wickham is the secondary. Wickham represents one of the main reasons why Elizabeth and Darcy do not get together at first. As the story progresses his character gets known to be a liar and a cheat, but he ends up getting married to Elizabeth's younger sisters, Lydia by convincing them to run away. Jane is the favorite sister, and is cute that she gets mentioned almost the same amount as Bingley, which ends up being her husband.

Collins gets mentioned almost as much as Wickham, and that is curious, because he was also (in his mind) a contender for Elizabeth's affection. He tried to propose to her offering to save their family from destitution of their house, but Elizabeth rejected him.

Bennet is the last name of our protagonist, and in this case it gets mentioned so much, because there are 5 miss Bennet, a Mrs. Bennet and a Mr. Bennet. And they interact in the plot quite a lot. The Gardiners are the uncles of Elizabeth, and they are in the figure, because they helped a lot in all the problems the family encountered.

In figure 5 we also see clearly that our protagonist is Elinor. But in this case, Marianne, her sister is almost as active in the plot as her. Whilst Elinor only has one love interest, Edward, Marianne has two. Colonel Brandon and Willoughby. It is quite sad that Brandon gets mentioned so much less than Willoughby since he is the one who ends up married to Marianne, but as we stated before, Marianne favored Willoughby up until the day he left her to marry some other richer girl.

Dashwood is the last name of our protagonists. And Jennings is the name of the meddler in the story. She tried her best to accommodate Marianne and Elinor with their respective partners, but always managed to make things worst.

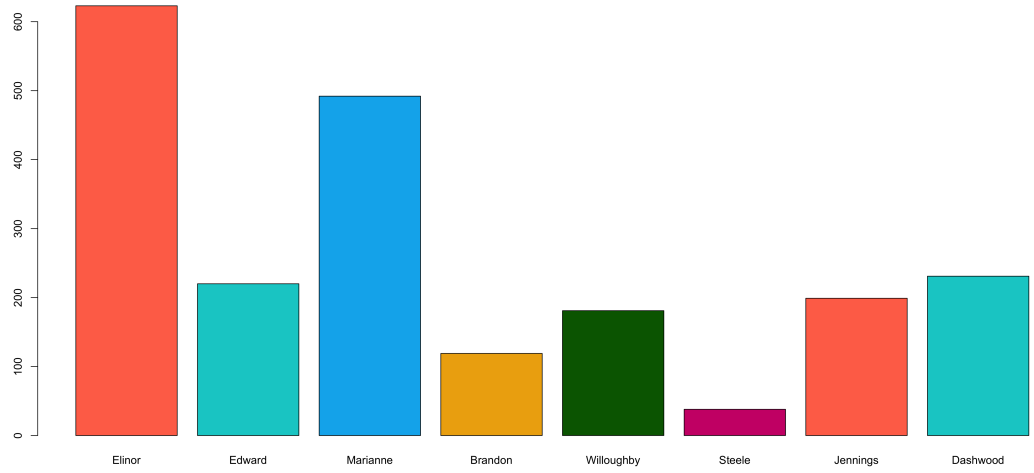


Figure 5: Bar plot of the frequency of characters in the book Sense and sensibility by Jane Austen.

5 Other experiments

As we played with the different variables and forms we could use the libraries in R, we created a variable with only the non repeated words of the story of Pride and prejudice. Just for fun, we tried to see how far in the story it stoped making sense without the repetition of the words. So here is the snippet:

"pride and prejudice by jane austen chapter 1 it is a truth universally acknowledged that single man in possession of good fortune must be want wife however little known the feelings or views such may on his first entering neighbourhood this so well fixed minds surrounding families he considered rightful property some one other their daughters my dear mr bennet said lady to him day have you heard netherfield park let at last replied had not but returned she for mrs long has just been here told me all about made no answer do know who taken cried impatiently you tell i objection hearing was invitation enough why says young large from north england came down monday chaise four see place much delighted with agreed morris immediately take before michaelmas servants are house end next week what name bingley married oh sure five thousand year fine thing our girls how can affect them tiresome am thinking marrying design settling nonsense"

Nonsense seemed like the right word to stop.

6 Conclusions

As a general conclusion, I had more fun with this practice that I thought I would have. Jane Austen is one of my favorite authors, and I really enjoyed getting a peek at what this plots would show me. In the end, I could have written so much more, because I love the plot of both books and the social commentary that the era reflects on marriage customs and the position of woman

in society. It came as a surprise some of the information in the later plot, because of who ends up with who in the end.

Regarding the Gutenberg project, it was very interesting seeing all of those books available for download. Shakespeare, the Bronte sisters and Frankenstein have always been in my to read list, so now I could take a look at those works.

References

[1] Gutenberg project. https://www.gutenberg.org/about/background/history_and_philosophy.html. Accessed: 2020-09-14.

[2] Origin of everything — history of the word "Gay". https://www.pbs.org/video/history-of-the-word-gay-bcbiuu/. Accessed: 2020-09-14.

[3] Gutenberg project — pride and prejudice. https://www.gutenberg.org/ebooks/1342. Accessed: 2020-09-14.

[4] Gutenberg project — sense and sensibility. https://www.gutenberg.org/ebooks/161. Accessed: 2020-09-14.

Practice 3: Gutenbergr project.

Histograms

Mayra Cristina Berrones Reyes 6291

September 22, 2020

1 Introduction

The Gutenberg Project is an online library that offers its users free access to more than 60,000 free books in different formats that go from ebooks, html, plain text, etc. It began with creator Michael Hart in 1971 at the Materials Research Lab at University of Illinois [3].

In previous work, we explored some of the features of libraries for text analysis such as `gutenbergr`, `tidytext`, `tm`, and, `dplyr`. Going deeper in analysis we now explore the sentiment analysis of texts using the library of `tidytext` to perform other experiments using histograms as a tool.

2 Books

Again for this project we use a book from the acclaimed novelist Jane Austen [1], *Pride and Prejudice* [4] published in 1813. And to make contrast in the present study, we are going to be comparing results with another book from another famous female author from the 19th century.

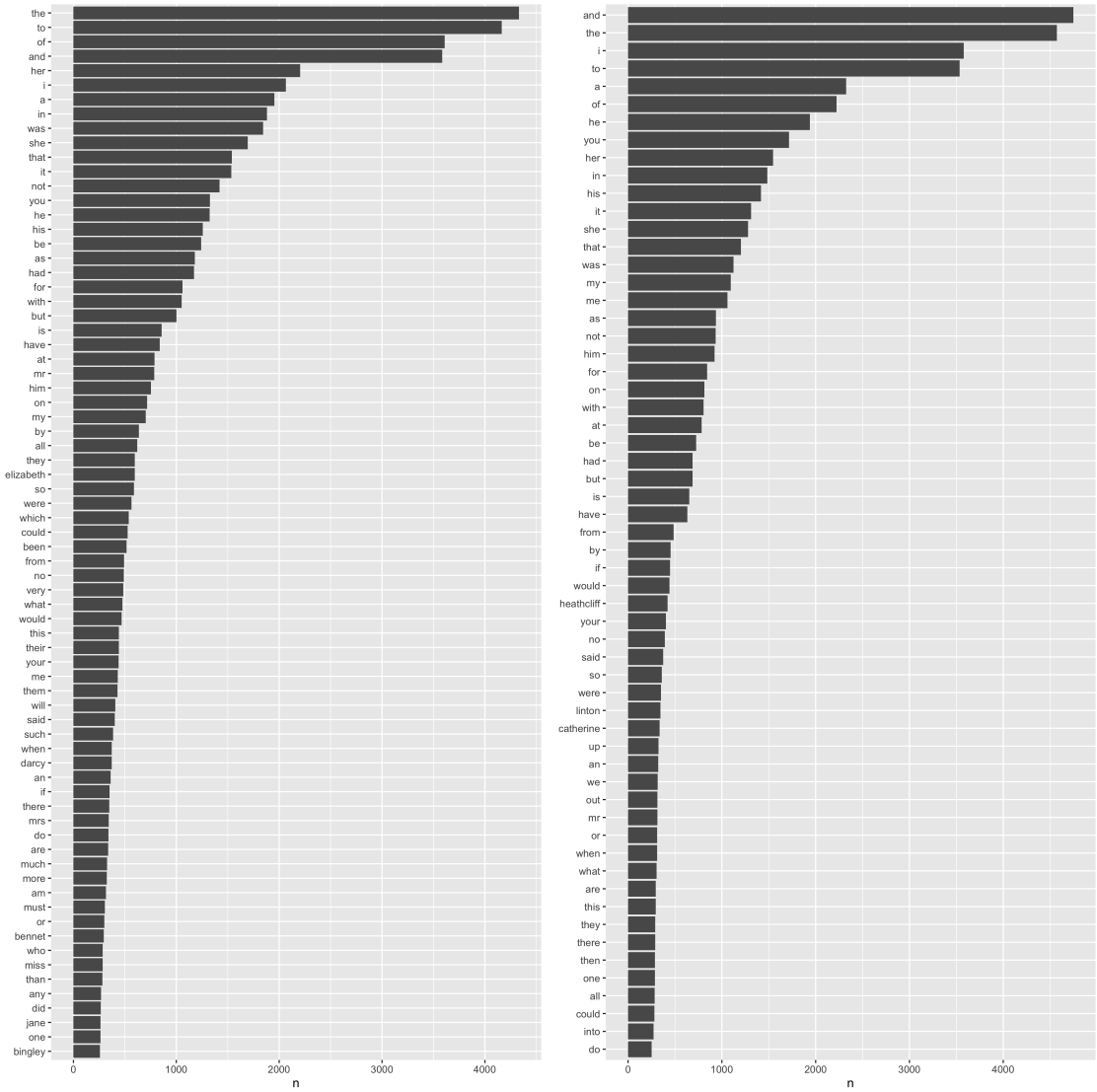
Wuthering Heights [8] is a novel by the author Emily Bontë [2] published in 1847. Her work is now regarded as classic in English literature, but in her time, the way she depicted mental and physical cruelty so attached to the love story she constructed in her book, challenged several Victorian ideas about religion, morality and class.

This two books are so alike in the way it changed public perception about class and the woman's place in society in their time, but their themes and tone of writing sound and feel to the reader so polarizing. So it gives us the focus for this experiment. Using the sentiment analysis of the library `tidytext`, we work to find the frequency of positive and negative words in both books.

3 Data exploration

To begin the exploration of the data, we start by tokenizing the words of each work. In Figure 1 we see that there is not much difference in the use of what is known as stop words. The main

distinct feature of this, is that in the work of Brontë Subfigure 1b we have more of this connective words than in Subfigure 1a which is the Austen book.



(a) Pride and Prejudice.

(b) Wuthering Heights

Figure 1: Frequency of all words in both books.

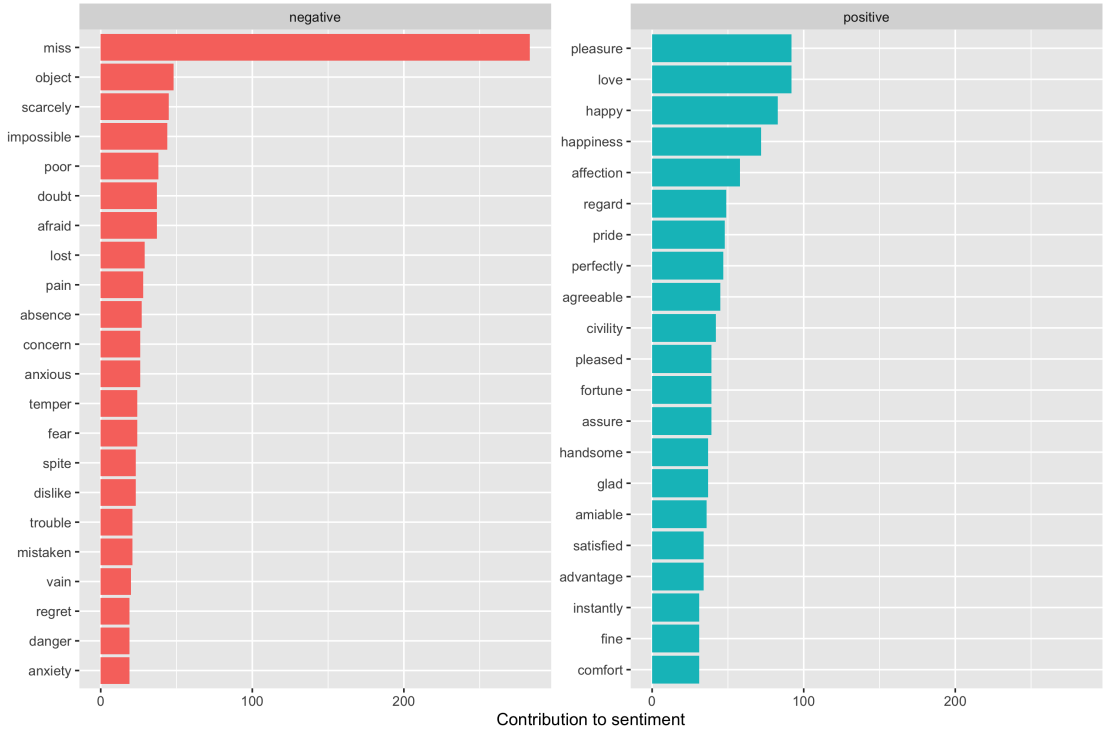
With this Figure 1 we can assess that Brontë is more prone to use connective words in her work. We can easily see that most of the words used on both books are similar in nature, except for the names of the characters. This can be associated with the fact that both authors are from the same era, and they both use British English.

Stop words however are not the main interest of the experimentation. The next step, in order to use the sentiment library is to create another variable without the stop words. This can be accomplished with the same library we used to tokenize the words on the downloaded document, only this time, we add the `anti join` variable. Inside we put `stop words` which is a data frame that contains pre loaded connective words in the English language.

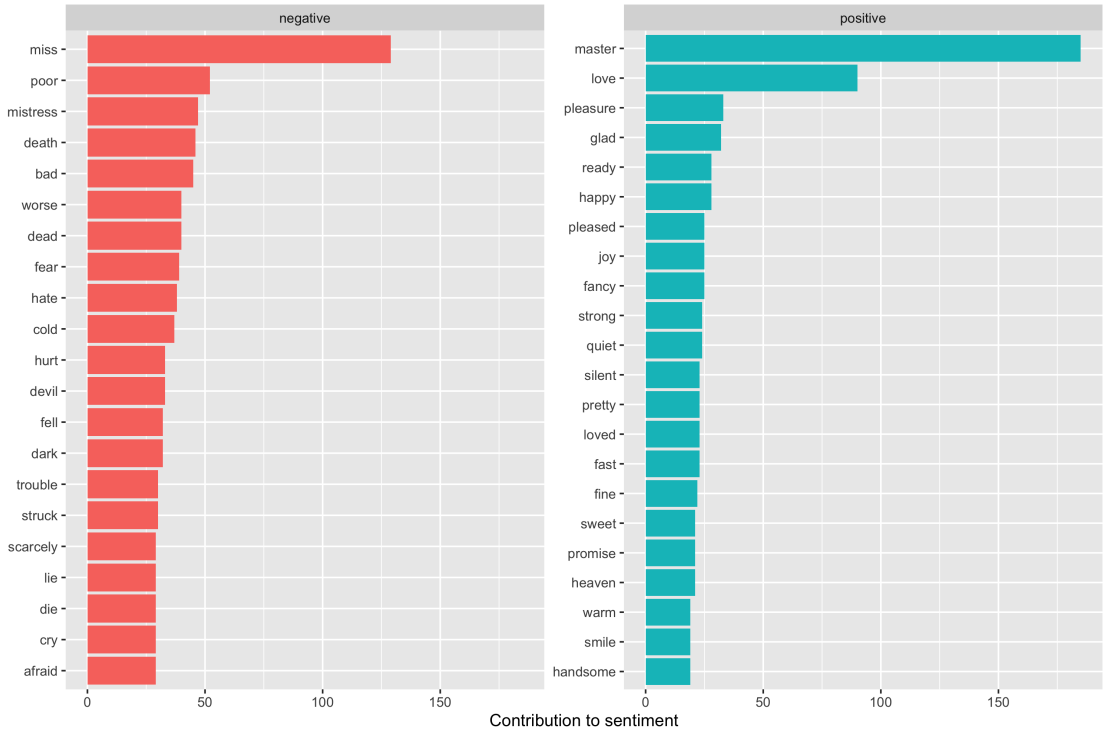
Having the data free of stop words, we can now use the library of sentiment to analyse both books. There are a few libraries of sentiment in the `tidytext` library, but in this case we choose the `bing` because it manages the words into positive or negative emotions. In Figure 2 the Subfigure 2a represents the positive and negative words of the book *Pride and Prejudice*, and the same is represented in Subfigure 2b for the *Wuthering Heights* book.

Right away when comparing the two Subfigures we notice that the negative words form the Austen novel have the feeling of being quite benign as opposed to the negative words in the Brontë book. For instance, in textitPride and Prejudice we have words like danger, poor, lost, absence, concern, dislike. And on the other hand in *Wuthering Heights* we have words like dead, hate, dark, hurt, devil.

In the side of the positive words, we see how the distribution of the frequency of the words is smaller in the Brontë side. Also, the top word *master* can be interpreted differently inside the book.

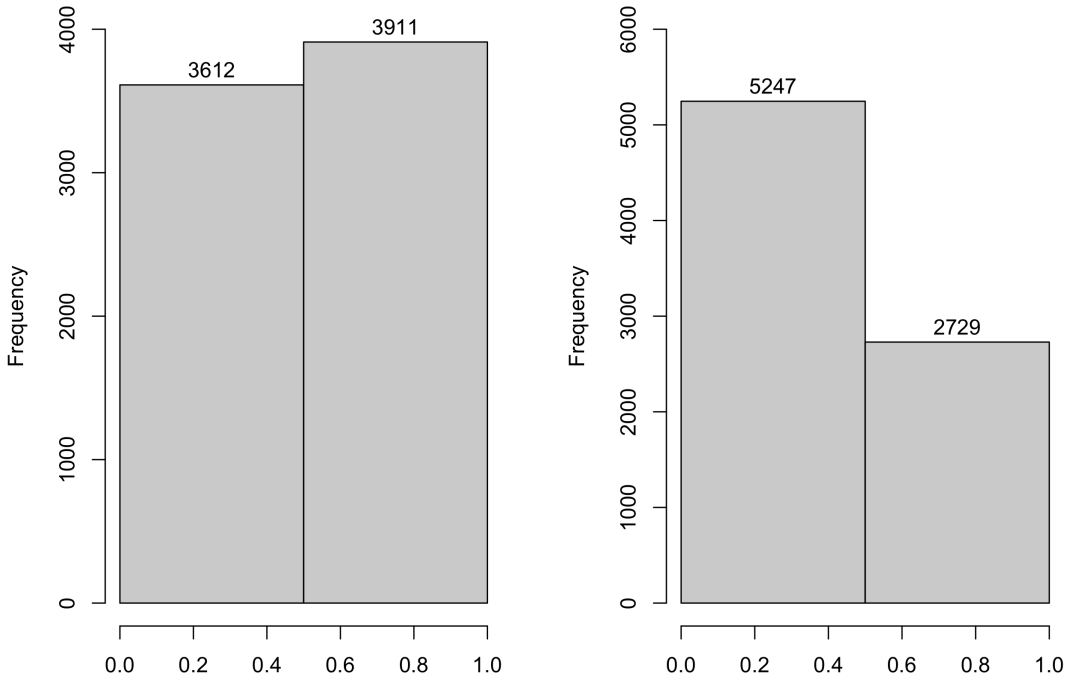


(a) Pride and Prejudice



(b) Wuthering Heights

Figure 2: Sentiment analysis of both books.



(a) Pride and Prejudice

(b) Wuthering Heights

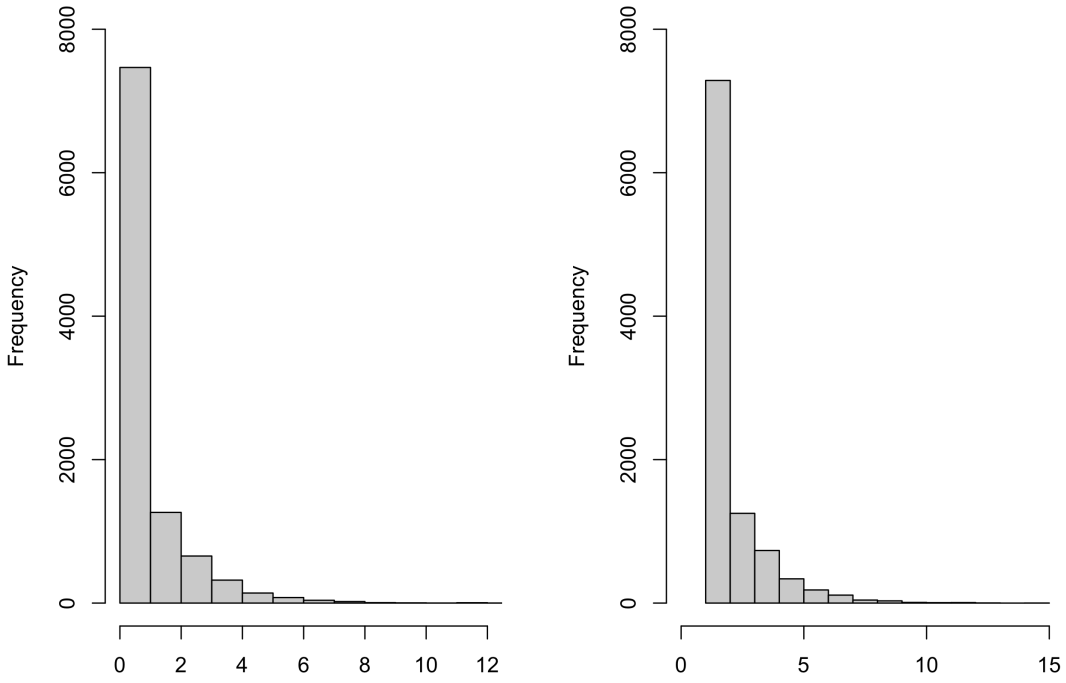
Figure 3: Frequency of positive and negative words. Negative are represented in 0 and positive in 1.

Having this outline of the sentiment analysis, we need to transform the data we have into something we can use for the histograms for the second part of this work. The reason we used the library `bing` for the sentiment analysis is because it divides the words into positive and negative emotions. With this, we transform them into binary data.

The first thing to know after this transformation is the distribution of positive versus negative words in both works. In Figure 3 we used histograms to represent that distribution. If we add the positive and negative words in each book, we find that the difference between total of words is not that great. *Pride and Prejudice* holds 7,523 words, while *Wuthering Heights* has 7,976.

The main and visible difference is that in Subfigure 3a the positive and negative words are almost even, with positive words being slightly higher. In the case of Subfigure 3b we see a stark difference, being that the negative words represent almost two thirds of the total of words.

In this case, Figure 3 goes appropriately with the previous knowledge of the books. While both authors are female, lived in the 19th century and both are novelists, the feeling of the book



(a) Histogram with rgeom tool.

(b) Histogram with made code.

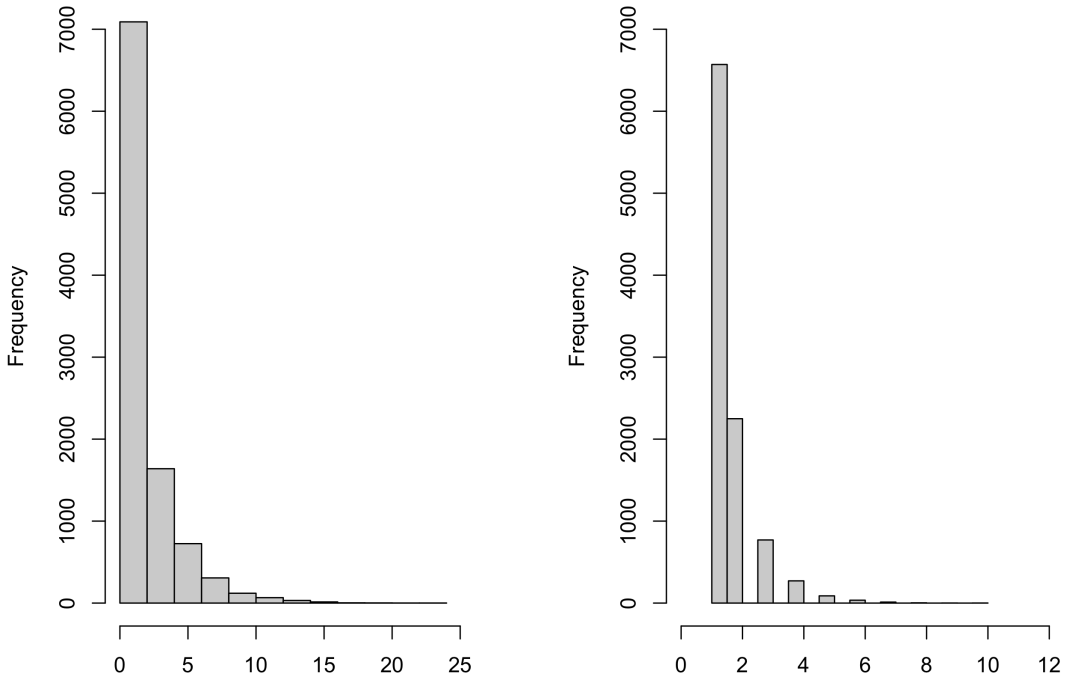
Figure 4: Comparison between the use of the tool `rgeom` and a made code on the book Pride and Prejudice.

is quite different. While in *Pride and Prejudice* our main protagonist Elizabeth is guilty of being biased and proud in her view of Darcy, and he in turn was spoiled and prideful as well, they are quite tamed compared with how volatile and prone to violence Heathcliff is. Also, the main protagonist Catherine is very selfish and cruel to everyone around her.

For this, the difference in characters behaviour, can be interpreted as one of the main reasons for the disparity in distribution of positive and negative words.

Having confirmed our first idea of the difference in tone in the books, we now use the information of the positive and negative words to perform distribution experiments. In Figure 4 we first perform an experiment with the tool `rgeom` [6] with the same variables we have in our data, to see if we can recreate something similar with our interpretation of the problem in a made code.

The `rgeom` tool takes a number of experiments and the percentage of success we have. In both cases we take as success if they have a positive word, so the percentage we give is the ones we calculated from our data. In the case of *Pride and Prejudice*, we have a percentage of 51%



(a) Histogram with rgeom tool

(b) Histogram with made code

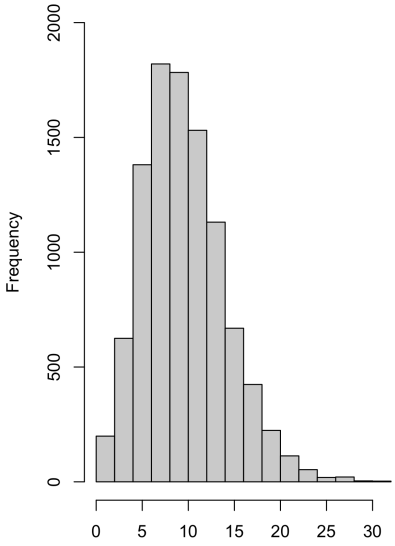
Figure 5: Comparison between the use of the tool `rgeom` and a made code on the book Wuthering Heights

positive words, and in *Wuthering Heights* we have a 34%.

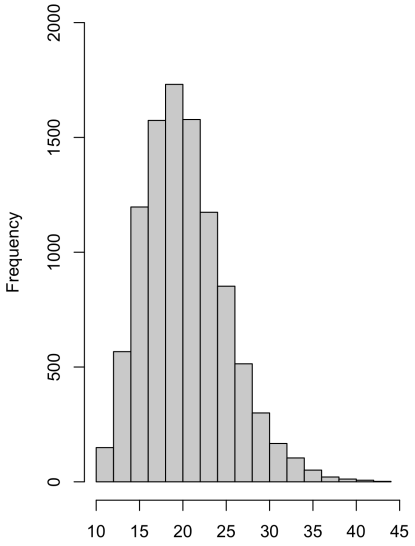
In Subfigure 4a we have the result of a 10,000 repetition of the experiment with a 0.51 probability of success. We replicated the experiments using our data of binary information, and with the tool `sample` we take a "word" and we see if it is a success (positive) or not (negative). As we can see, the behaviour of both plots is almost the same.

In Figure 5 we have the same experimentation, but with the book *Wuthering Heights*. In this case the difference between the subfigures are a bit more noticeable than in the previous experiment. In Subfigure 5a we have the result of a 10,000 repetition of the experiment with a 0.34 probability of success. Subfigure 5b uses the same made code than in Subfigure 4b changing only the parameters of probability and the data used.

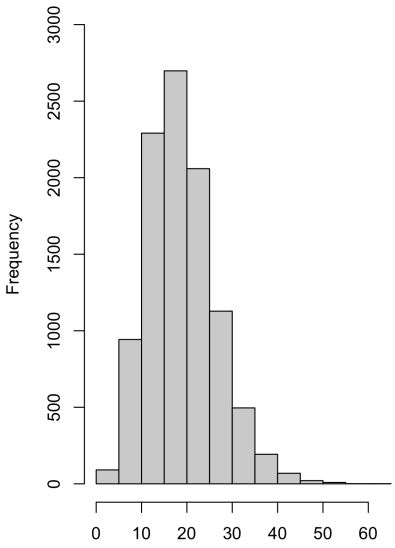
In Figure 6 we use now the `rbinom` [5] tool to further our experiments. This tool adds a variable inside the parameters. Same as with the `rgeom` tool, we have the number of experiments, then we have a number (k) which determines the amount of successful events before it can move



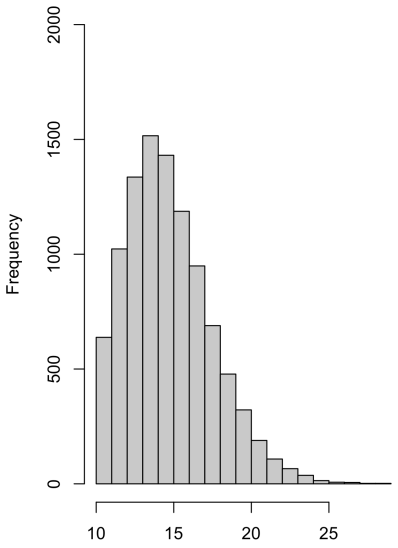
(a) Using rbinom, Pride and Prejudice.



(b) Using made code, Pride and Prejudice.



(c) Using rbinom, Wuthering Heights.



(d) Using made code, Wuthering Heights.

Figure 6: Comparison between the use of the tool `rbinom`

Table 1: Table of results from the `rhyper` tool with book Pride and Prejudice.

11	12	13	14	15	16	17	18	19
3	2	10	26	49	96	184	278	419
20	21	22	23	24	25	26	27	28
619	799	984	1036	1084	1101	892	818	596
29	30	31	32	33	34	35	36	38
429	276	143	84	43	24	2	2	1

Table 2: Table of results from the `rhyper` tool with book Wuthering Heights

20	21	22	23	24	25	26	27	28
2	2	8	27	41	86	145	258	417
29	30	31	32	33	34	35	36	37
566	786	977	1171	1126	1176	988	800	646
38	39	40	41	42	43	44	45	
375	213	109	50	17	8	5	1	

to the next iteration. Then it needs the parameter of probability of success.

In Subfigures 6a and 6c we use the parameter of 10,000 iterations, with k number equal to 10. The probability of success is 0.51 and 0.34 respectively. Then for the made code in both books we use the same parameters, but using our binary data of each book.

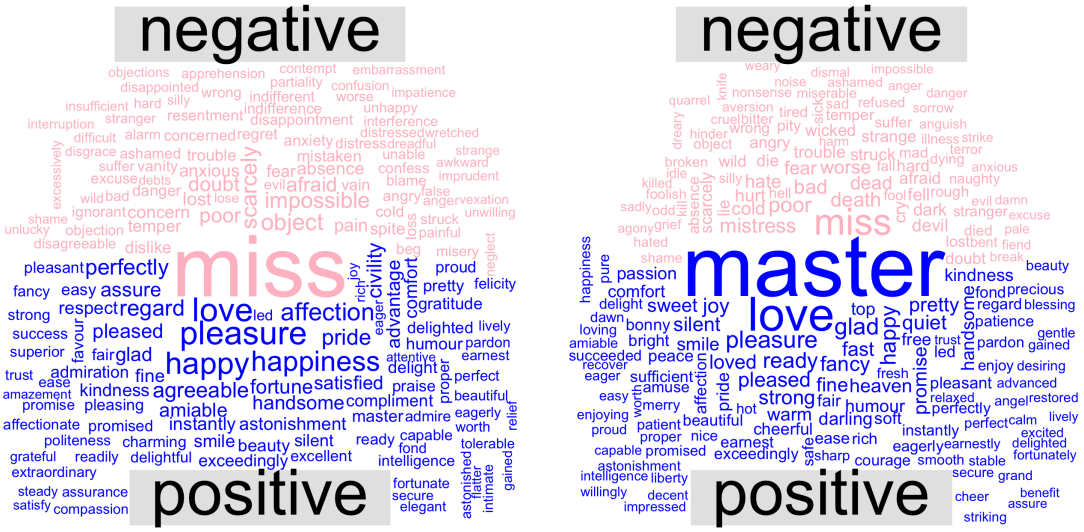
As we can appreciate in the comparison in Figure 6 the behaviour of the plots is very similar.

For the last experimentation with distribution, we take the tool `rhyper` [7]. In this case, it requires the amount of iterations, the amount of "white balls", the amount of "black balls" and the size of our sample. The white balls are represented as our positive words. The black balls as the negative ones. We set our iterations at 10,000 and our sample as 50.

Both tables have a gray and a white area. The gray area represents the number of bad words found in our sample of 50. The white area represents the number of times that many negative words repeated it self in our 10,000 iterations. In Table 1 we see the results of using the `rhyper` tool with the book Pride and Prejudice. Doing some calculations with our percentage of positive and negative words of this book, we have as a limit of negative words 24. So we add up all the frequencies after 24, and divide them for 10,000. The result of this is 0.4411.

Moving to Table 2 we have a different limit parameter. In this case is 33. So we add up all of the elements after the 33 and divide them by 10,000. The result is 0.4388.
.

Now, for our self made code, we use the same parameters for each book, 10,000 iterations, k of 10, and a sample of 50. The only variants are the positive and negative words for each book. Something else we added was the limit variable, which is the same as we used to make our



(a) Pride and Prejudice

(b) Wuthering Heights

Figure 7: Word cloud of positive and negative words in both books

calculation with the tables, 24 for Austen and 33 for Brontë. As a result we have, in the case of Austen a probability of 0.448 and 0.435 for Brontë. Comparing this to our results with the calculation of the tables we have 0.4411 and 0.4388, we have very similar results.

4 Other experiments

In the previous practice I struggled with word cloud plots. Here in Figure 7 I make a comparison of positive and negative words in both books. I find it quite funny that the main words for each book are miss and master.

5 Conclusions

With our first comparison of positive and negative distribution of words we were able to identify that Wuthering Heights has a more dark, more macabre tone of writing than Pride and Prejudice. The following plots with the R tools such as `rgeom, rbinom` and `rhyper` were interesting to develop in self made code. It was also curious to see the last experiment, and be able to prove again our impression on the difference of writing in both books.

In Tables 1 and 2 what sparked attention was the begining and end frequency of each. In Table 1 we begin with a small number and en with the number 38, with very low frequencies.

This tells us that there is a small amount of negative words in this data. And in Table 2 we begin with frequency of 20 and end up in the number 45. This is also aligned with the percentage of the positive and negative words we know of the book.

References

[1] Biografia de jane austen. hhttps://www.biografiasyvidas.com/biografia/a/austen.htm. Accessed: 2020-09-22.

[2] Biografia de emily bronte. https://www.biografiasyvidas.com/biografia/b/bronte_emily.htm. Accessed: 2020-09-22.

[3] Gutenberg project. https://www.gutenberg.org/about/background/history_and_philosophy.html. Accessed: 2020-09-22.

[4] Gutenberg project – pride and prejudice. https://www.gutenberg.org/ebooks/1342. Accessed: 2020-09-22.

[5] The binomial distribution. https://www.rdocumentation.org/packages/stats/versions/3.3/topics/Binomial. Accessed: 2020-09-22.

[6] Geometric distribution. https://rpubs.com/mpfoley73/458721. Accessed: 2020-09-22.

[7] The hypergeometric distribution. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Hypergeometric. Accessed: 2020-09-22.

[8] Gutenberg project – wuthering heights. http://www.gutenberg.org/ebooks/768. Accessed: 2020-09-22.

Practice 4: The Poisson distribution

Mayra Cristina Berrones Reyes 6291

September 29, 2020

1 Introduction

For this practice, we explore the properties of several `R` distributions, such as `rpois, runif,` and `rexp`, taking note in their difference as well as the similarities under certain circumstances. Before beginning with any experimentation, a brief description of each distribution will assist in the understanding of the problem.

1.1 Poisson distribution

A Poisson random variable can be used to model the number of times an event happened in a certain interval of time. It is described in terms of the rate in which this events happen, because an event can occur several times in a single interval. To be able to use this distribution we have to know the average number of events in an interval, which is designated by the sign of lambda (λ) [2].

Lambda is also called the rate parameter. The probability of k events happening in an interval is represented in Equation 1:

$$P(k \text{ events in interval}) = e^{-\lambda}\frac{\lambda^k}{k!} \tag{1}$$

In `R` we can generate this distribution with the module `rpois`. The description for this module in `R help` says that is a "Density, distribution function, quantile function and random generation for the Poisson distribution with parameter λ" [4]. The parameters it takes are stated in Equation 2:

$$\texttt{rpois}(\text{n, lambda}) \tag{2}$$

where

- **n** is the number of random values to return.
- **lambda** is the vector of non negative means.

1.2 Exponential distribution

The exponential distribution describes the time between events in a process in which they occur in a continuous and independent manner at a constant average rate. This distribution also

uses the parameter lambda (λ) and it is called the rate parameter [1]. The equation for this distribution can be seen in Equation 3:

$$f(x;\lambda) = \begin{cases} \lambda \exp^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3}$$

In `R` we can generate this distribution with the module `rexp`. The description for this module in `R help` says that is a "Density, distribution function, quantile function and random generation for the exponential distribution with rate (i.e., mean 1/rate)" [3]. This is a very important distinction, because we do not have to make the division in the variable ourselves, only add the *rate* value as it is [1].

The parameters it takes are stated in Equation 4:

$$\texttt{rexp}(\text{n, rate} = 1) \tag{4}$$

where

- **n** is the number of observations. If length(n) > 1, the length is taken to be the number required.
- **rate** is the vector of rates.

1.3 Uniform distribution

The uniform distribution is one of the most simple and commonly used distribution. In the continuous uniform distribution we have the probability in Equation 5 as:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \tag{5}$$

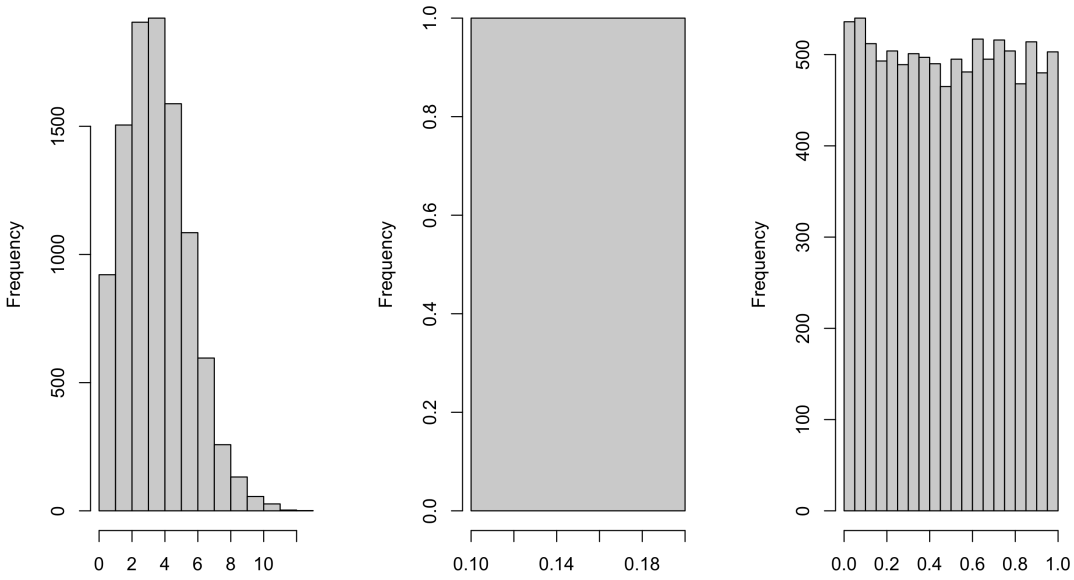
In this distribution, any interval of numbers of equal width have an equal probability of being observed. The curve describing the distribution is a rectangle with a constant height across interval and 0 height elsewhere. We can use in `R` a tool to see the this distribution, called `runif`. The description for this module in `R help` says that "These functions provide information about the uniform distribution on the interval from `min` to `max`. `runif` generates random deviates" [5].

The parameters it takes are stated in Equation 6:

$$\texttt{runif}(\text{n, min} = 0, \text{max} = 1) \tag{6}$$

where

- **n** is the number of observations. If length(n) > 1, the length is taken to be the number required.
- **min, max** are the lower and upper limits of the distribution. Must be finite.



(a) Histogram of `rpois`(10000, 4). (b) Histogram of `rexp`(1, 4) (c) Histogram of `runif`(10000)

Figure 1: Histograms showing the behaviours of each distribution (Poisson, Exponential and Uniform).

2 Comparisons between distributions

In Figure 1 we can appreciate how all of the described distributions behave. None of them look similar to one another. The goal of this practice is to show with which parameters we can make all of this distributions look similar, and explain why that is possible.

In order to search for ways to make them behave in a similar manner, we first needed an example of how the Poisson distribution works. Say we live near an airport, and we can see as an average, 4 planes in the span of one hour. Now as an experiment we record the number of planes we see every hour for 10,000 hours. In Subfigure 1a we can see such behaviour. As expected, the majority of the plane occurrences are 3, 4 or 5 inside the interval of an hour. There is a better chance of no planes showing than there is to have more than 7 in this interval. Understanding this, we begin now to translate this performance to the exponential and uniform distributions [7].

For the Poisson distribution it is possible to develop several generators that help us make a simply uniformly fast Poisson distribution. Such generators can be classified in several groups:

1. Generators that are based on the connection with homogeneous Poisson process. [6]. These type of generators are simple and run in expected time proportional to λ.

2. Inversion methods, that use sequential search starting at 0 and run in expected time

proportional to λ. If the sequential search starts here, the expected time is $\mathcal{O}(\sqrt{\lambda})$ [8]

3. Generators that are based on recursive properties of the distribution.
4. Rejection methods.
5. The acceptance complement method with the normal distribution as a starting distribution.

In this case, one of the simple generators we have the exponential and uniform distribution. Beginning with the connection between the Poisson distribution and exponential arrival times in a homogeneous point process we have the Lemma 2.1 for the Poisson distribution [7].

Lemma 2.1. *If $E_1, E_2, \ldots$ are exponential random variables, and X is the smallest integer such that*

$$\sum_{i=1}^{X+1} E_i > \lambda$$

then X is Poisson (λ).

As proof of this Lemma 2.1 we are presented with Proof 2:

Proof. Let f_k be the gamma (k) density.

$$P(X \leq k) = P(\sum_{i=1}^{k+1} E_i > \lambda) = \int_{\lambda}^{\infty} f_{k+1}(y)dy$$

Thus, by partial integration,

$$\begin{aligned} P(X = k) &= P(X \leq k) - P(X \leq k-1) \\ &= \int_{\lambda}^{\infty} (f_{k+1}(y) - f_k(y))dy \\ &= \int_{\lambda}^{\infty} (y-k)\frac{y^{k-1}}{k!}e^{-y}dy \\ &= \frac{1}{k!}\int_{\lambda}^{\infty} d(-y^k e^{-y}) \\ &= e^{-\lambda}\frac{\lambda^k}{k!}. \end{aligned}$$

□

In the end of this proof we see the same equation as we explained in the introduction in Equation 1 on the Poisson distribution. The algorithm that was based upon this proof is shown in Algorithm 1:

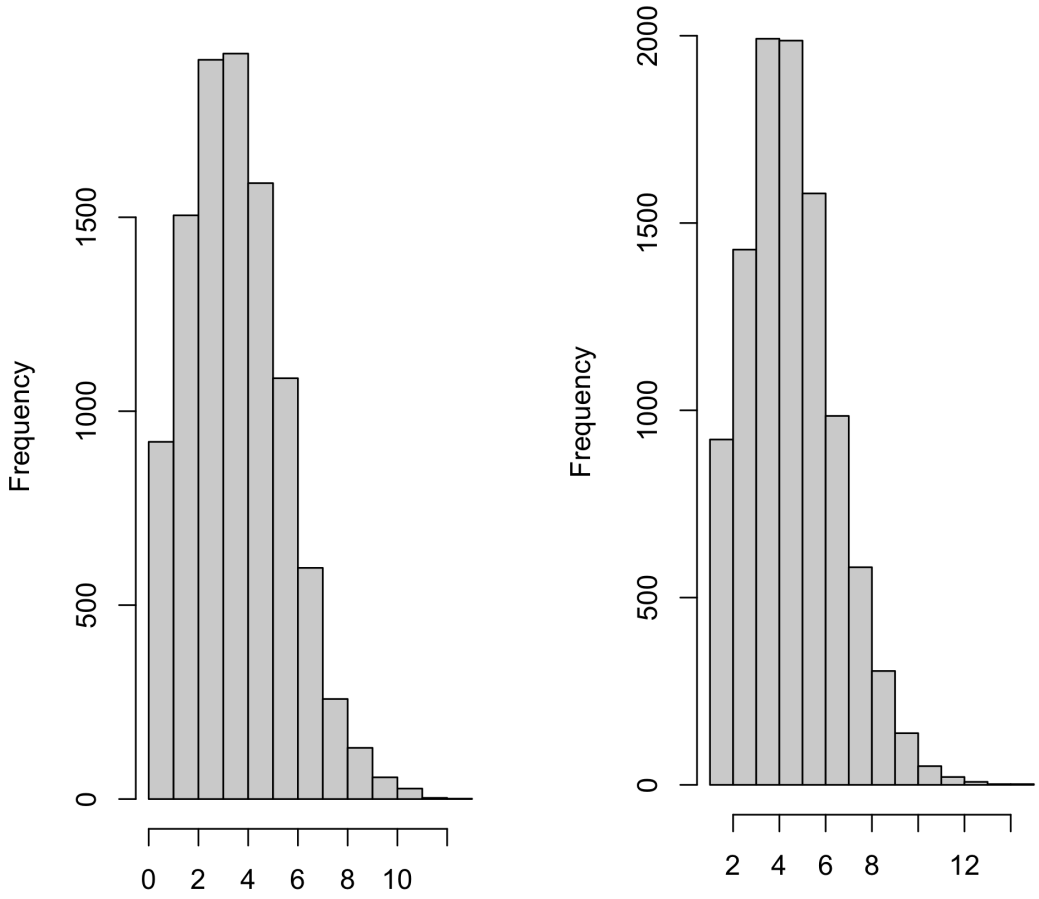
If we translate this to the example of the planes, we randomly generate 10,000 distributed random variables X, interpreting X as the waiting time from the passing of one plane to another measured by the hour. If we know the value of λ is 4 as the average of planes, the time that each event happens can be obtained from the cumulative sum of the waiting times from event to event. We can then aggregate the number of events that happen per unit time, and make an histogram

```
X ← 0;
Sum ← 0;
while TRUE do
    Generate an exponential random variate E;
    Sum ← Sum + E;
    if Sum < λ then
        X ← X + 1;
    else
        RETURN X;
    end
end
```

Algorithm 1: Poisson generator based upon exponential inter-arrival times

out of them.

Remembering the information of `R help` we put as a parameter the λ and not $1/\lambda$. Using the same parameters as the Poisson example we now have the results shown in Figure 2. Further analysis of both Subfigures 2a and 2b we can see that the behavior is very similar, in contrast with the first experimentation in Subfigure 1b.



(a) Histogram of `rpois`(10000, 4).

(b) Histogram of experimentation with cumulative sum of `rexp`

Figure 2: Histograms showing the behaviour of distribution Poisson and cumulative exponential experimentation.

For the uniform distribution we take on the fact that a uniform random variable is distributed as e^{-E}, we can equate the Lemma 2.1 to the next Lemma 2.2 [7].

Lemma 2.2. *Let $U_1, U_2, ...$ be uniform $[0, 1]$ random variables, and X is the smallest integer such that*

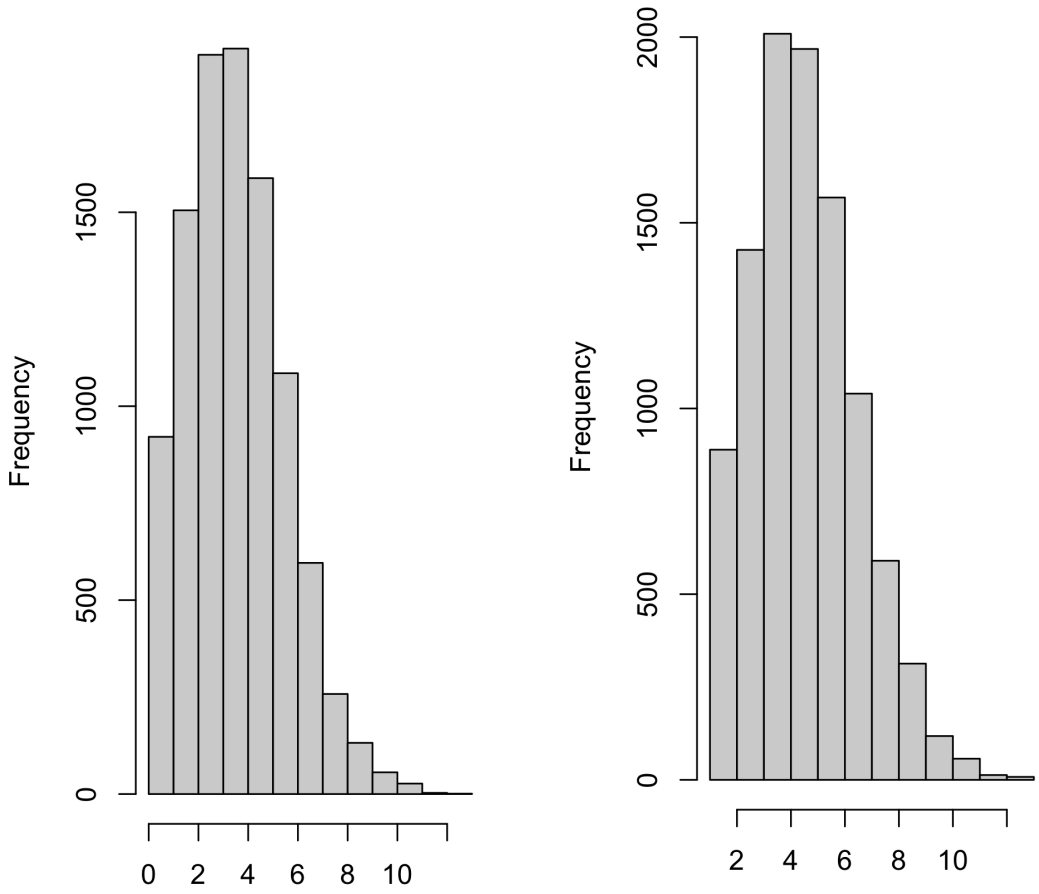
$$\prod_{i=1}^{X+1} U_i < e^{-\lambda}$$

then X is Poisson (λ).

The resulting algorithm of this Lemma is equivalent as Algorithm 1

```
X ← 0;
Prod ← 1;
while TRUE do
    Generate a uniform [0, 1] random variate U;
    Prod ← Prod U;
    if Prod > e^{-λ} (the constant should be computed only once) then
        X ← X + 1;
    else
        RETURN X;
    end
end
```

Algorithm 2: Poisson generator based upon multiplication of uniform random variates.



(a) Histogram of `rpois`(10000, 4).

(b) Histogram of experimentation with multiplication of uniform random variates `runif`

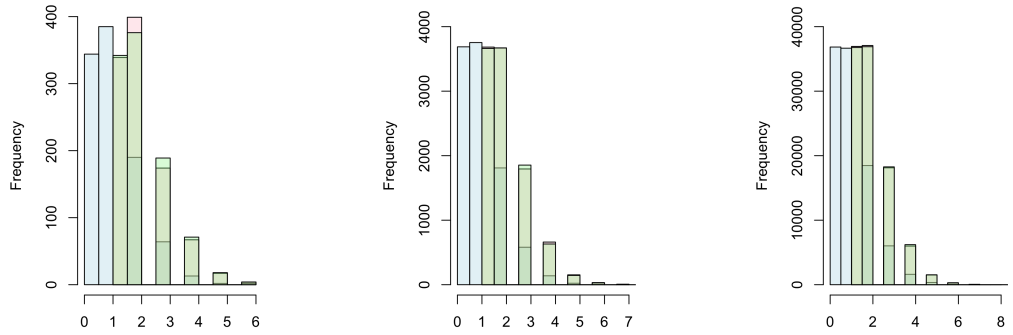
Figure 3: Histograms showing the behaviours of distribution poisson and uniform random variates experimentation.

In this case, the experiment of the planes was not so clear in mind, because the involvement of the exponential part in the code did not make much sense. So we went to the `R` code to see what was the result of each of the experiments, exponential and uniform.

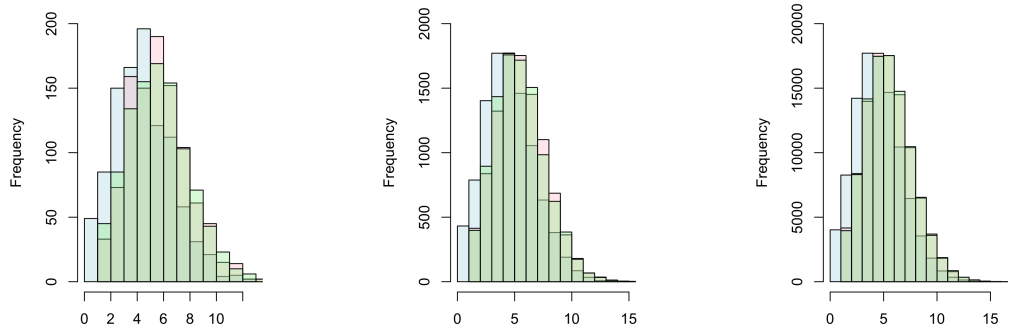
We printed the last iteration of each distribution, and in the case of the exponential, the cumulative sum was clear, as the results were 0.0614 and 1.4261. If the goal set is 1, those two events were enough to complete the interval. In the case of the uniform distribution the last iteration numbers where 0.2350, 0.8153, 0.4806, and 0.0748. At first this did not make sense, but after calculating the product of all this numbers and comparing them to the result of $1/-\lambda$ it made sense why we had to use multiplication instead of cumulative sum.

3 Experimentation with different parameters

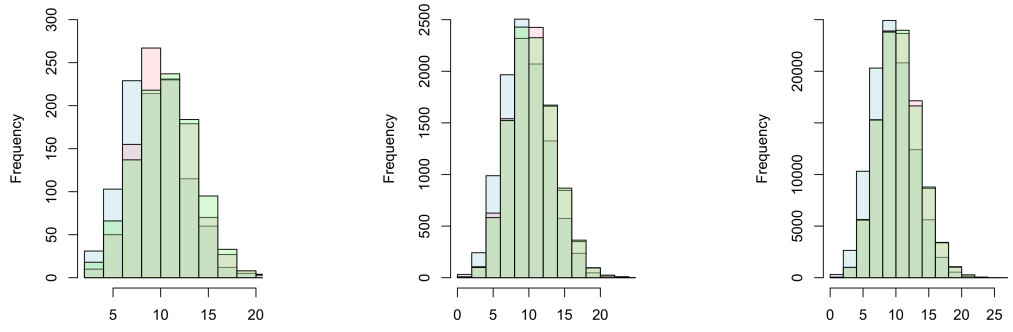
In Figure 4 there we can see all the distributions in different parameters. It is very curious that the lowest parameter on lambda in all three different iterations of Subfigures 4a, 4b and 4c, they all show a behaviour similar to the exponential that we saw in the introduction of this work. Subfigures 4d, 4e and 4f, all with lambda = 5, resemble the Poisson distribution and Subfigures 4g, 4h and 4i with lambda = 10 to the normal distribution.



(a) $\lambda = 1$, and replica = 1,000 (b) $\lambda = 1$, and replica = 10,000 (c) $\lambda = 1$, and replica = 100,000



(d) $\lambda = 5$, and replica = 1,000 (e) $\lambda = 5$, and replica = 10,000 (f) $\lambda = 5$, and replica = 100,000



(g) $\lambda = 10$, and replica = 1,000 (h) $\lambda = 10$, and replica = 10,000 (i) $\lambda = 10$, and replica = 100,000

Figure 4: Histograms of all the distributions with different parameters. The blue represents the Poisson distribution, the pink is the exponential distribution, and the green the uniform distribution.

Figure 5: Example made by Dr. Elisa as a visual aid of how the Poisson distribution works.

4 Conclusions

The conclusions arrived in this practice where more clear in the exact order we mentioned them. Seeing the Poisson distribution first as an example made easier to understand. Then to transform the exponential distribution to the behaviour of the Poisson one was easy because the concept of cumulative sum as depicted on the Figure 5 Dr. Elisa gave us in class was simple. After all of that, the uniform distribution using multiplication to lower the final result to fit inside the exponential condition was a bit harder to understand, but the experimentation in `R` made it easier to visualize and digest.

Figure 4 is also something interesting, how changing the parameters of lambda, we can change the behaviour of the distribution, making it seem like other distributions.

References

[1] Probability distributions in python. https://www.datacamp.com/community/tutorials/probability-distributions-python. Accessed: 2020-09-28.

[2] The poisson distribution and poisson process explained. https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459. Accessed: 2020-09-28.

[3] The exponential distribution. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Exponential. Accessed: 2020-09-28.

[4] The poisson distribution. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Poisson. Accessed: 2020-09-28.

[5] The uniform distribution. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/Uniform. Accessed: 2020-09-28.

[6] AC Atkinson. The computer generation of poisson random variables. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):29–35, 1979.

[7] Luc Devroye. Random variate generators for poisson – poisson and related distributions. *Computational Statistics and Data Analysis*, (8):247–278, 1989.

[8] CD Kemp. New algorithms for generating poisson variates. *Journal of computational and applied mathematics*, 31(1):133–137, 1990.

Practice 5: Pseudo Random Numbers

Mayra Cristina Berrones Reyes 6291

October 6, 2020

1 Introduction

A pseudo random number is the term used for the computer generated random numbers. The prefix "pseudo" is used be differentiate this type of number to a truly random number that is generated by a random physical process such as radioactive decay [3].

A pseudo random number generator (PRNG) is refereeing to an algorithm that uses mathematical formulas to create sequences that approximate as much as posible random numbers. This type of numbers are very important in the area of computational science, because more often than not, we find the need to use randomness in a computer program to be able to perform experiments. It is quite difficult however, to get a computer to do something by chance, because a computer follows instructions in a way that can be predicted one way or another [1].

Truly random numbers are not possible to generate from a deterministic method, so we use PRNG techniques to develop random numbers using a computer. A PRNG sequence is completely determined by an initial value called seed.

2 Generators based on linear recurrences

A great discovery in pseudo random generators was the introduction of techniques based on linear recurrences. They where used as standard in the second half of the 20th century. Their quality was known to be inadequate, but there were no better methods available.

A linear congruential generator (LCG) in an algorithm used to make pseudo randomized numbers with a discontinuous piecewise linear equation. This generator is defined by the recurrence relation represented in Equation 1:

$$X_{n+1} = (aX_n + c) \ \texttt{mod} \ m \tag{1}$$

where:

m $0 < m-$ the "modulus"

a $0 < a < m-$ the "multiplier"

c $0 \leq c < m-$ the "increment"

X_0 $0 \leq X_0 < m-$ the "seed"

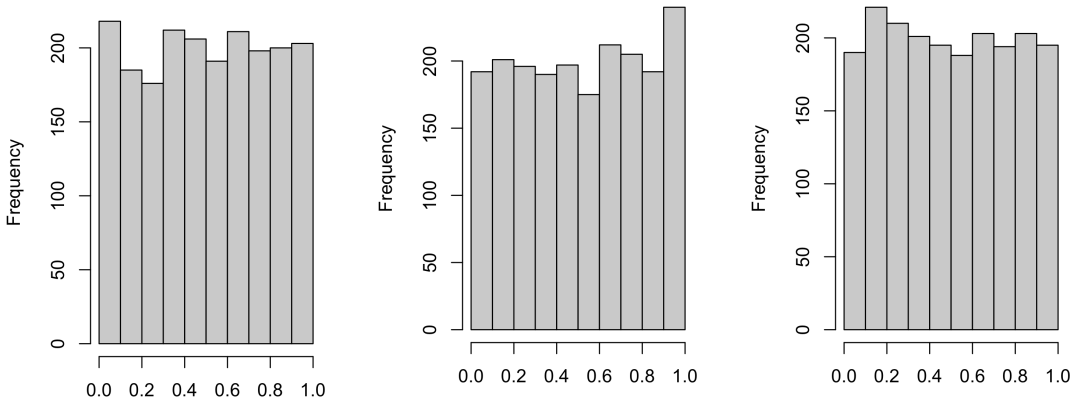
Given this generator, for this experimentation we are working with three different examples of LCG. First we have the standard generator of R for uniform random numbers `runif`. The other one is one generated by code shown in 1, where the variables `a, c,` and `m` in Equation 1 are prime numbers. The seed used is fixed to 27.

Listing 1: Code for uniform random numbers in R

```
> uniforme = function(n, semilla) {
>     a = 11551
>     c = 27077
>     m = 39709
>     datos = numeric()
>     x = semilla
>     while (length(datos) < n) {
+         x = (a * x + c) %% m
+         datos = c(datos, x)
>     }
>     return(datos / (m - 1))
>     }
```

The third generator uses the same structure as 1, but the values of `a, c,` and `m` are taken from the experimentation in the ANSI experimentation by Saucier [4], $\texttt{m} = 2^{32}$, $\texttt{a} = 1103515245$ and $\texttt{c} = 12345$. Another main difference is the quality of the seed used. In this case we set the seed using the current system time in microseconds.

On all three we perform a uniform test with a Chi - squared. In all three the `p value` gives us a correct uniformity, as shown in Figure 1.



(a) Histogram of `runif` (2000) (b) Histogram of function `uniforme`(2000, 27) (c) Histogram of function `lcg.rand`(2000)

Figure 1: Histograms showing the behaviours of each experimentation.

3 Gaussian distribution

The Gaussian distribution, also known as normal distribution, in probability theory is a type of continuous probability distribution for a real valued random variable. The form of its probability density is shown in Equation 2.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2}$$

where:

μ is the mean or expectation of the distribution.

σ is the standard deviation.

σ^2 is the variance of the distribution.

A random variable with this Gaussian distribution is called normal deviate.

Using the standard Box – Muller transform we are experimenting using the three generators of uniform random numbers mentioned in Section 2 and moving the values of Z. If Z is a standard normal deviate then $X = Z\sigma + \mu$ will have a normal distribution [2].

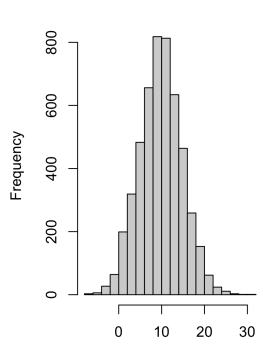
Listing 2: Code for Gaussian distribution in R

```
> gaussian = function(mu, sigma) {
>     u = runif(2);
>     z0 = sqrt(-2 * log(u[1])) * cos(2 * pi * u[2]);
>     z1 = sqrt(-2 * log(u[1])) * sin(2 * pi * u[2]);
>     datos = c(z0, z1);
>     return (sigma * datos + mu);
>  }

cat(gaussian(0, 1))
```

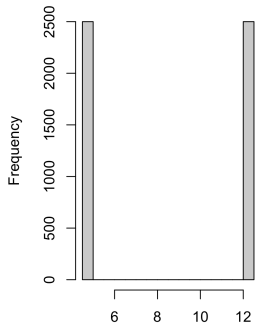
In Figure 2 we see all the distributions for all the experimentations with the different parameters using the code in Listing 2. On Figures 2a, 2b and 2c we use both parameters Z_0 and Z_1. As we can appreciate Figures 2a and 2c behave as a normal distribution. Changing the parameter used to only Z_0 made little difference in those two distributions, as we can see in Figures 2d and 2f, but in the case of Figure 2e it changes quite a bit.

In an effort to normalize whatever was happening on Figures 2b and 2e we used the two random generated numbers as separated variables `u1` and `u2` with the two Z variables. In Figure 2g the parameter for `u1` is `runif`(1) and `u2` we use the function `uniforme` (1, `u1` $*$ 1000).

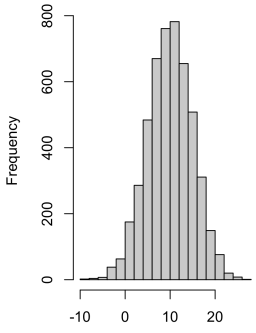
Then in Figure 2h we tried the same using now `u1` as the function `lcg.rand` (1) and `u2` `uniforme` (1, `u1` $*$ 1000). Both of this figures behave as normal distributions. Thinking that the solution to the `uniforme` function was the variables, we made a last attempt. In Figure 2i we used `u1` as the function `uniforme` (1, 27) and `u2` `uniforme` (1, `u1` $*$ 1000). This however yielded the same results as using the `uniforme` function by itself.



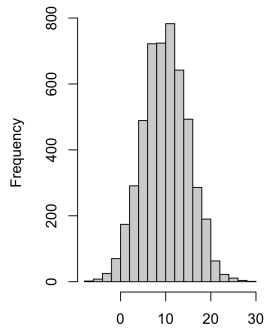
(a) With Z_0 and Z_1 using `runif`



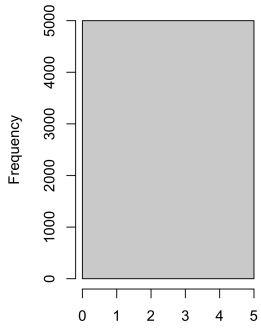
(b) With Z_0 and Z_1 using `uniforme` function



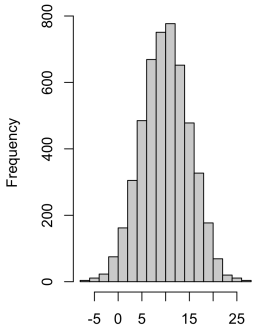
(c) With Z_0 and Z_1 using `lcg` function



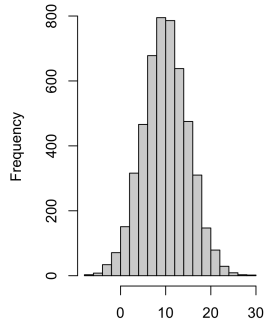
(d) With Z_0 using `runif`



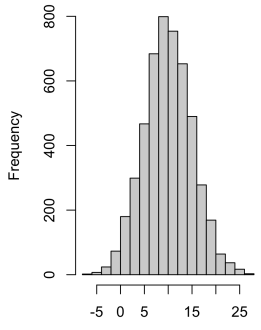
(e) With Z_0 using `uniforme` function



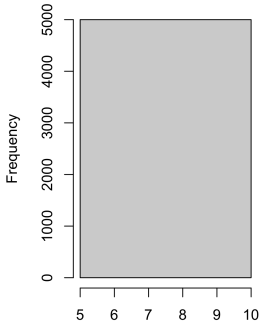
(f) With Z_0 using `lcg` function



(g) With Z_0 and Z_1 using `runif` as u1 and `uniforme` as u2

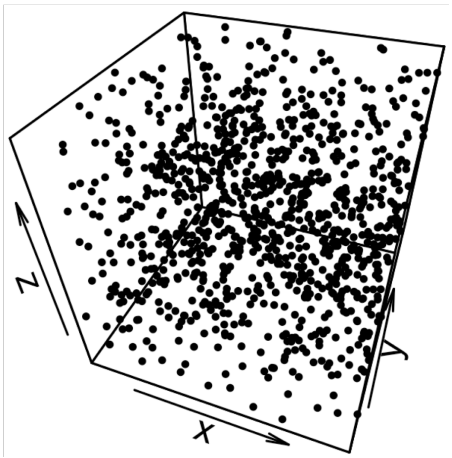


(h) With Z_0 and Z_1 using `lcg` as u1 and `uniforme` as u2

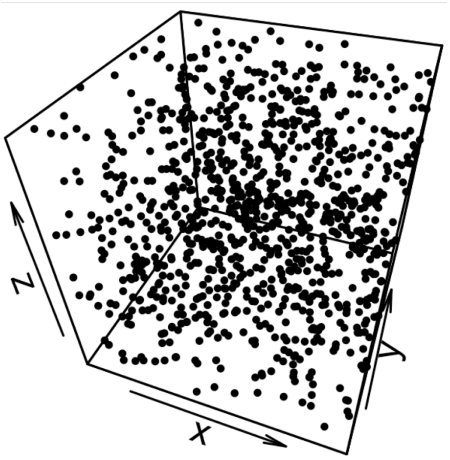


(i) With Z_0 and Z_1 using `uniforme` as u1 and `uniforme` as u2

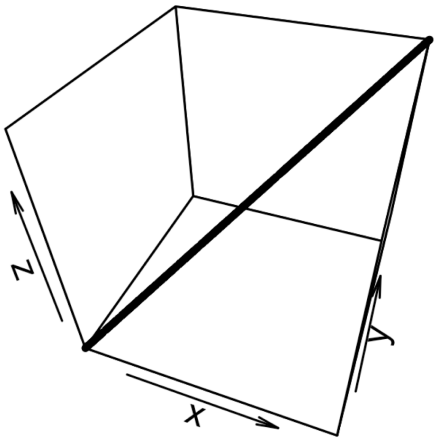
Figure 2: Histograms of all the distributions with different parameters.



(a) 3D of `runif` (2000)



(b) 3D of function `lcg.rand`(2000)



(c) 3D of function `uniforme`(2000, 27)

Figure 3: 3D depiction of each uniform random number generator

4 Other experiments

Looking for a better seed for the experimentation of the uniform distribution, we arrived to an example of a different way of seeing the randomness of each experimentation. In Figure 3 we can see in each point the behaviour of the `runif`, the `lcg` and the `uniforme` functions.

Same as with the distributions plot, in this figure we see an odd behaviour on the `uniforme` function.

5 Conclusions

It was really interesting learning more about random numbers, since we use them so much when doing experimentations. It was also compelling seeing the difference between using a good

seed and a bad one in the generation of uniform numbers, because in the first figures when we prove their uniformity there was no big sign that the results in the Gaussian experiment where going to behave that way.

There was also the expectation that in the experiments of changing the independent variables into dependent ones, that the histograms would look a lot more weird than the ones with independent variables, if only because in class there was the idea that taking away the independence of the variables would ruin the outcome of the Gaussian experiment.

References

[1] Pseudo random number generator. https://www.geeksforgeeks.org/pseudo-random-number-generator-prng/?ref=rp. Accessed: 2020-10-05.

[2] Box – muller transform. https://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform. Accessed: 2020-10-05.

[3] M. Luby. Pseudorandom number. *Pseudorandomness and Cryptographic Applications.*, page 266, 1992.

[4] Richard Saucier. *Computer Generation of Statistical Distributions.* Army Research Laboratory, 2000.

Practice 6: Statistical Tests

Mayra Cristina Berrones Reyes 6291

October 13, 2020

1 Activity

Please answer the next questions, taking into consideration the shared links.

a) Describe the relationship between hypothesis testing and statistical tests.

To understand the relationship between these two concepts, first we describe them separately.

A statistical test gives us a mechanism to be able to make quantitative decisions about processes. The goal of this decisions is to determine if there is enough evidence to "reject" a conjecture or hypothesis about our process. In this case, the conjecture is called null hypothesis [8]. A null hypothesis proposes that no significant difference exists in a set of given observations [1].

A classical use of the statistical tests occurs in process control studies.

Hypothesis testing is a way we can test if the results of an experiment really have meaningful results. This means that we test to see if the results we have are valid by checking out the odds that said results could have happened by chance [6].

It is very common in statistics to estimate a parameter from sample data. For example, a sample of the mean of the data is then used as the point estimate of the population mean. An hypothesis test addresses the uncertainty of the sample estimate, and instead of providing an interval, it attempts to refute a specific claim about a population parameter based on the sample data we took for the experiment [8].

A common format for hypothesis test is shown in Listing 1.

- H_0: Null hypothesis.
- H_a: Alternative hypothesis.
- Statistic test: This is based on the specific hypothesis test.
- Significance level: Commonly known as α, defines the sensitivity of the test.
- Critical value: This encompasses the values of the statistic test that lead to a rejection of the null hypothesis.

Table 1: Names and types of errors of the significance level.

	Truth about population	
Decision based on sample	H_0 is true	H_0 is false
Fail to reject H_0	Correct decision (Probability = $1-\alpha$)	**Type II error** Fail to reject H_0 when it is false (Probability = β)
Reject H_0	**Type I error** Rejecting H_0 when is true (Probability = α)	Correct decision (Probability = $1-\beta$)

To answer the original question, seeing the common format for a hypothesis test, we can now say that they are both needed for a good analysis of the experimentation, because we should not just trust that the sample we took for our statistical analysis is the correct one, so we should combine engineering judgement with statistical analysis.

b) What would indicate to reject the null hypothesis?

To reject a null hypothesis we perform a statistical test, and then we compare its results with the critical value. If it is greater than the critical value, the hypothesis is rejected. A great explanation Jonathan Christensen [3] says "In a theoretical underpinning, hypothesis tests are based on the notion of critical regions: the null hypothesis is rejected if the test statistic falls in the critical region. The critical values are the boundaries of the critical region. If the test is one sides, then there will only be one critical value, but in other cases there will be two."

So we reject the null hypothesis if this falls in the critical value.

c) How is the output of a statistical test interpreted?

In this case, the output of the statistical test can be that we accept the null hypothesis, or we reject it. If ithe null hypothesis falls in the critical value, we reject it. The null hypothesis is a statement about belief [8], so the test procedure is done in a way that the risk of rejecting the null hypothesis is small when it is in fact true.

d) How to select the alpha value?

The risk we mentioned in previous questions, α, is often referred as value significance level of the test. When we use a small value of α it is often said that it actually proves something when the null hypothesis is rejected.

There is no magic significance level that will give us a 100% accuracy results. The most common α values are 0.05 and 0.01, but they are mainly used based on tradition. Because this test are based in probability, there is always a chance of a wrong conclusion.

When doing this experiments, there are 2 types of errors, which are related and determined to the level of significance and the test used. So we should determine which error has worst concecuences for the experiment, before defining the risks.

In Table 1 we name these type of errors.

Table 2: Examples of different parametric and non parametric procedures for the same type of analysis.

Analysis Type	**Parametric**	**Non parametric**
Compare means between 2 distinct/independent groups	Two sample t-test	Wilcoxon rank sum test
Compare 2 quantitative mesurements taken from the same individual	Paired t-test	Wilcoxon signed rank test
Compare means between 3 or more distinct/independent groups	Analysis of variance (ANOVA)	Kruskal-Wallis test
Estimate the degree of association between two quantitative variables	Pearson coefficient of correlation	Spearman rank correlation

e) What are the most frequent misinterpretations of the p-value?

The p-value is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct. It is the measure pf the probability that an observed difference may have occurred by random [2].

The incorrect interpretation of the p-value is very common. One of the most frequent is to interpret it as the probability of making a mistake by rejecting the true null hypothesis, which as we can see in Table 1 is a Type I error.

Misinterpreting the p-value as the error rate creates the illusion that there is more evidence against the null hypothesis. But if we base the whole experiment on a study of p-value of 0.05, the difference observed in the sample may not exist in the whole population [4].

f) What is the statistical power and what is it for?

The statistical power of an experiment refers to how likely it is to distinguish an actual effect from one chance. It is the likelihood that the test we perform is correctly rejecting the null hypothesis, which looking at Table 1 is the probability of avoiding Type II error [11].

A high statistical power means that the results we have are likely valid, and as the power increases, the probability of making a Type II error decreases. It can be used as a tool to estimate the sample size required in order to detect an effect in an experiment [5].

g) Examples of parametric and non-parametric statistical tests.

The definition of non parametric is very convoluted, and better explained by examples. In Table 2 we see examples of the type of analysis and the parametric and non parametric procedure.

h) Summarize THE GUIDE to find the statistical test you are looking for.

It is importan to select correctly the type of statistical test we are going to use to analyze our data. There is a very helpful flowchart [9] shown in Figure 1 that helps to find the most suitable statistical test, depending on the type of data to analyze.

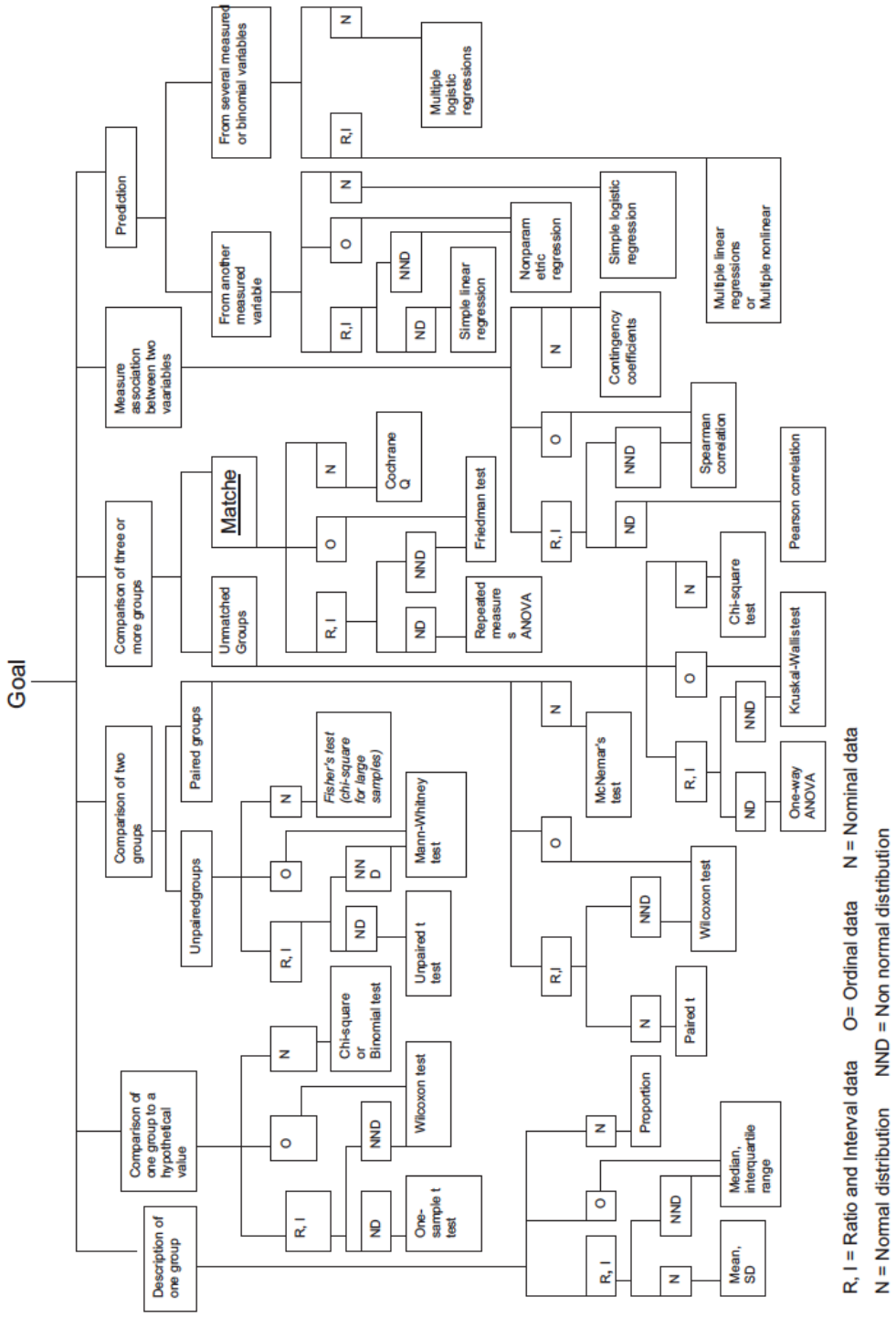


Figure 1: Flowchart for the selection of appropriate statistical tests

i) What are the assumptions to apply parametric techniques?

There are four important assumptions when it comes to the use of parametric tests in the analysis of data [10], which are listed in Listing 1.

- **Normal distribution data:** The p-value depends on a normal sampling distribution. If the sample size is big enough and the sample data point value are aproximately normally distributed, then the central limit ensures a normally distributed distribution.
- **Homogeneity of variance:** The variable in the population where we took the samples have been taken in similar variance of these populations.
- **Interval data:** The data point values should be for numerical variable and measured at this level.
- **Independence:** Data point values for variables for different groups should be independent of each other. In regression analysis, the errors should likewise be independent.

2 Experimentation

Using the same topic as practice 1, "Enviromental practices" from the page of INEGI [7] we attempt to replicate as many statistical test as possible. In this case, we still use the tables from the water section, but because the behavior of the previous data was to fractured and not compatible with any of the tests, we are focusing now on the quality of certain services on urban spaces. In this case, they give a 100 percent, and distributed in bad, regular and good quality.

In R we made a variable with each distribution, hoping to get a more normally distributed data than the one that we used before. In Figure 2 we see the distribution of each variable.

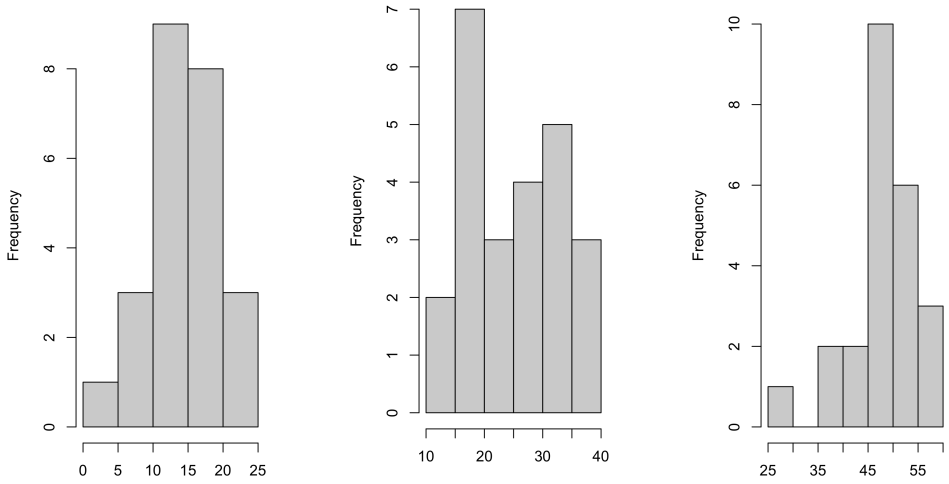
At plain sight we can speculate that Figure 2a and 2b are more or less resembling a normal distribution. But to be certain, we can perform some experimentation on the data.

2.1 One sample t-test

This is a parametric test to prove if the mean of a sample from a normal distribution could reasonably be a specific value. This test needs as values the name of the data, and a number that represents the posible mean. Seeing all three histograms, we set this parameter in each one as 15, 25, and 45, because is the most central in each one.

The results of the test are shown in Table 3.

Interpreting this results, we put our null hypothesis in each case that the mean of each data variable was 15, 25, and 45 for `bad`, `regular` and `good` respectively. Seeing the results on Table 3 we can see that in all three cases we accept the null hypothesis. In column mean we also can see that our first assumption to use the medium value of the histogram as mean was not so far off. The farthest one was the `good` variable, and it was off only by 2.51.



(a) Distribution of bad reviews (b) Distribution of regular reviews (c) Distribution of good reviews

Figure 2: Distribution of the public opinion about the water service of urban spaces.

Table 3: Output in R of the One sample t-test.

	One sample t-test variables				
Data	t	df	p-value	95% confidence interval	Mean
Bad	-0.4531	23	0.6547	12.4916 to 16.6067	14.54
Regular	0.0600	23	0.9527	21.8560 to 28.3318	25.09
Good	1.7898	23	0.0866	44.6074 to 50.4312	47.51

Table 4: Output in R of the Wilcoxon signed rank test.

	Wilcoxon variables			
Data	V	p-value	95% confidence interval	Pseudo median
Bad	141	0.8115	12.4177 to 16.8965	14.77
Regular	151	0.9888	21.6393 to 28.4672	25.15
Good	234	0.0150	45.7382 to 50.6462	48.25

Table 5: Output in R of the Shapiro test.

	Shapiro test variables	
Data	W	p-value
Bad	0.9664	0.5814
Regular	0.9485	0.2515
Good	0.8776	0.0074

2.2 Wilcoxon Signed Rank Test

We move to a experimentation to test the mean of a sample when normal distribution is not assumed of the data. In Table 4 we see the results of this test on each variable.

In this case, without the assumption of normality in our distribution the p-value, we reject the null hypothesis on the `good` variable. The difference in the media is also a bit larger in all three variables.

2.3 Shapiro test

This test is used to see if our sample follows a normal distribution. Same as the experiments before, we are going to check if all of our variables follow a normal distribution. Table 5 shows the results of this test.

As expected, variables `bad` and `regular` pass the test as normal distribution, but the `good` variable does not.

2.4 Kolmogorov and Smirnov test

This test helps find out if two samples follow the same distribution. From Figure 2 we can see that the distribution of our variables are different, but in Table 6 we prove it further than just analyzing a figure.

2.5 Fisher F-test

This test can be used to check if two samples have the same variance. We used the same parings as the Kolmogorov-Smirnov test. The results can be seen in Table 7. In this case, all three variables pass the null hypothesis, and the ratio of variances gives us also a positive result

Table 6: Output in R of the Kolmogorov-Smirnov test.

	Kolmogorov-Smirnov variables	
Data	D	p-value
Bad and Good	1.0000	6.195×10^{-14}
Regular and Good	0.9166	6.996×10^{-11}
Bad and Regular	0.6250	0.0001

Table 7: Output in R of the Fisher F-test.

	Fisher F-test variables					
Data	F	num df	denom df	p-value	95% confidence interval	Ratio of variances
Bad and Good	0.4993	23	123	0.1028	0.2159 to 1.1541	0.4992
Regular and Good	1.2364	23	23	0.6150	0.5348 to 2.8582	1.2364
Bad and Regular	0.4038	23	23	0.0343	0.1746 to 0.9334	0.4038

between them.

2.6 Correlation

This test gives us as a result the linear relationship of the two continuous variables. The null hypothesis in this case is that the true correlation between both variables is zero. The results of this tests are shown in Table 8.

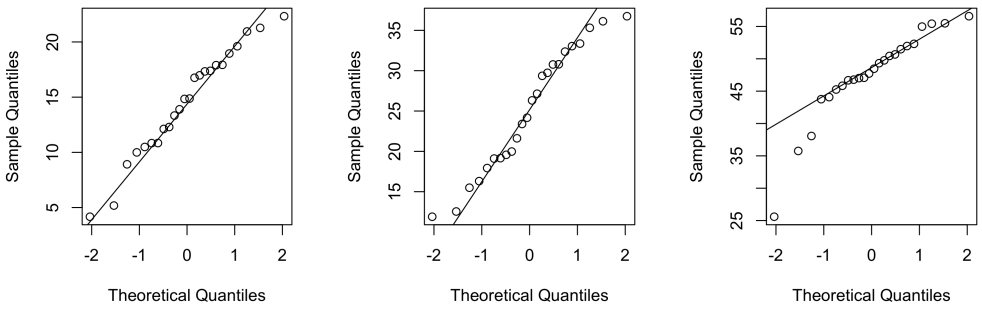
In this case the null hypothesis of the correlation could only be accomplished by the pair of `bad` and `good` variables.

3 Other experimentation

We have talked and experimented with different known statistical tests. And even if now when, after all the research they seem a lot more understandable, visual queues remain a favorite.

Table 8: Output in R of the correlation test.

	Correlation variables				
Data	t	df	p-value	95% confidence interval	Correlation
Bad and Good	1.5459	22	0.1364	-0.1034 to 0.6360	0.3130
Regular and Good	2.7261	22	0.0123	0.1242 to 0.7532	0.5024
Bad and Regular	2.9768	22	0.0069	0.1689 to 0.7723	0.5358



(a) Normal Q-Q plot of bad reviews (b) Normal Q-Q plot of regular reviews (c) Normal Q-Q plot of good reviews

Figure 3: Normal Q-Q plot of all the variables

So, after some searching, we found other ways to show our data. In Figure 3 we have the Q-Q plot, or more commonly known as the quantile-quantile plot. It helps to assess if a set of data came from some distribution such as normal or exponential.

This is the visual check of something that the t-test does, which is assume that the variable is normally distributed, so it is not a complete proof, but something to help visualize the data. In the case of Figure 3 we compare our results with Table 3, and we find similar results.

Another visual queue besides the histograms if Figure 2 to help find the mean of our data are the box plots. In Figure 4 we show the box plots of all the variables. Here we can see more clearly the mean of each variable, as well as the odd behavior of some of the data in variable `Good`.

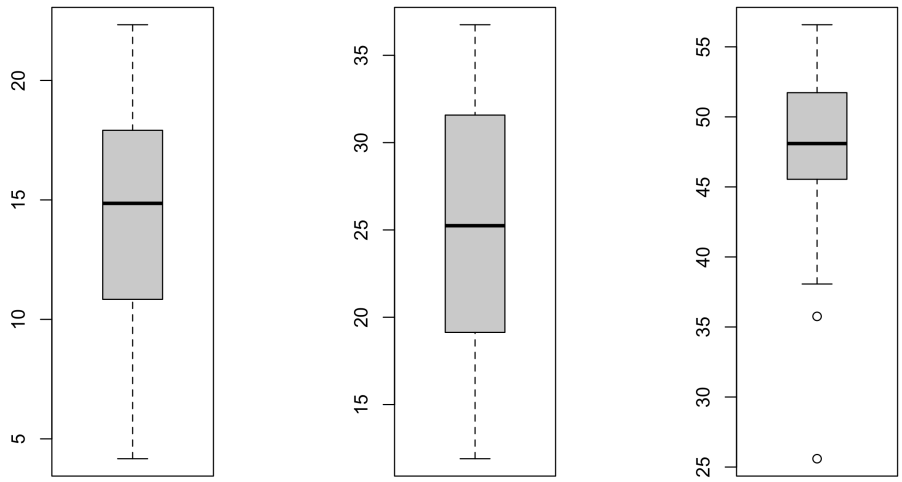
4 Conclusions

So far I was understanding fine the histograms and plots made in class, but in the last practice one of the mayor faults in my work was that I misinterpreted some data. With this practice the concepts of the tests and their results are clearer. They can help us move from just guessing our answers from the plots and actually proving some points, because many of the articles and books researched for this practice say that plots are only visual aids, and should require a bit more experimentation to back them up.

References

[1] Statistical tests: When to use which. https://towardsdatascience.com/statistical-tests-when-to-use-which-704557554740. Accessed: 2020-10-12.

[2] Brian Beers. P value definition. https://www.investopedia.com/terms/p/p-value.asp. Accessed: 2020-10-12.



(a) Box plot of bad reviews (b) Box plot of regular reviews (c) Box plot of good reviews

Figure 4: Box plots of all the variables

[3] Jonathan Christensen. What is a critical value in statistics. https://math.stackexchange.com/questions/281940/what-is-a-critical-value-in-statistics. Accessed: 2020-10-12.

[4] Minitab Blog editor. How to correctly interpret p values. https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-correctly-interpret-p-values. Accessed: 2020-10-12.

[5] Statistics for everyone. Statistical power: What it is, how to calculate it. https://www.statisticshowto.com/statistical-power/. Accessed: 2020-10-12.

[6] Stephanie Glen. Hypothesis testing. https://www.statisticshowto.com/probability-and-statistics/hypothesis-testing/. Accessed: 2020-10-12.

[7] INEGI. Medio ambiente. https://www.inegi.org.mx/temas/practicas/default.html#Tabulados. Accessed: 2020-10-12.

[8] Douglas C Montgomery, George C Runger, and Norma F Hubele. *Engineering statistics*. John Wiley & Sons, 2009.

[9] DISTRIBUTION OR NOT. How to select appropriate statistical test? *Journal of Pharmaceutical Negative Results| October*, 1(2):61, 2010.

[10] JP Verma and Abdel-Salam G Abdel-Salam. *Testing statistical assumptions in research*. John Wiley & Sons, 2019.

[11] Angela L.E. Walmsley. What is power? https://www.statisticsteacher.org/2017/09/15/what-is-power/. Accessed: 2020-10-12.

Practice 7: Curve fitting

Mayra Cristina Berrones Reyes 6291

October 27, 2020

1 Introduction

Part of the subject of curve fitting is the concept of correlation. In the field of statistics, correlation or dependence is any statistical relationship that two random variables have between each other. In commonly refers to the degree in which the pair is linearly related [1].

Correlations are useful because they can indicate the predictive side of a relationship, that help explore the data even further. The most familiar correlation measure is the Pearson coefficient, commonly known as the correlation coefficient. The product of this correlation attempts to establish a line of the best fit between two variables.

There is also a way to calculate the correlation between two variables, reveling non-linear interactions. They are called transformations. In correlation coefficient, there is no need for its values to have a normal shape, but it certainly helps to make them more clear to understand if the data is rearranged. This is where some transformations come in handy, depending on the type of data we are working on [2].

One transformation that we rely on for this experimentation, is the Tukey ladder of powers. A brief summary of this transformation is that we assume we have a collection of data, and we are interested to know the relationship between this variables. As we said before, a good way to understand our data is to re expressing its variables, in this case, using the power transformation [2].

Table 1 gives an example of the Tukey ladder of transformations.

2 Experimentation

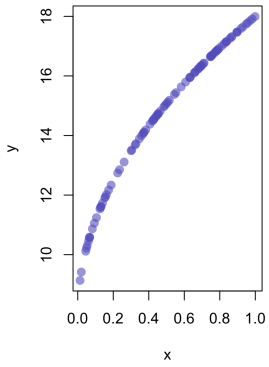
For this experimentation we are working with 4 different equations that have an `x` of random uniform numbers, with a `y` dependent of the value of x. Then we have a fifth experiment with an `x` with a different distribution, and a `y` dependent on that `x` but with a random value.

Table 1: Tukey ladder of transformations.

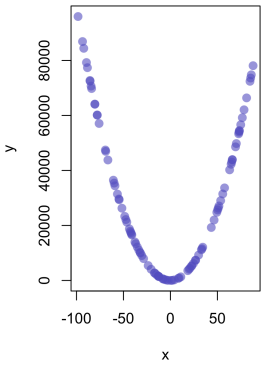
λ	-2	-1	-1/2	0	1/2	1	2
y	$\frac{1}{x^2}$	$\frac{1}{x}$	$\frac{1}{\sqrt{x}}$	$\log x$	$\sqrt{x}$	x	x^2

In Figure 1 we can see the scatter plots of all of this equations. In Figure 1a we have a polynomial equation. Figure 1b is a quadratic equation. Figure 1c is a exponential equation and Figure 1d is a logarithmic equation. Figure 1e is the one experimenting with the value of `x` and `y` .

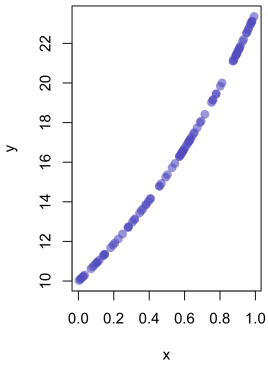
Visualization is a great tool to understand how some transformations work, so we use the `mosaic` and `manipulate` libraries in R to plot the behavior in each equation of the Tukey ladder of powers.



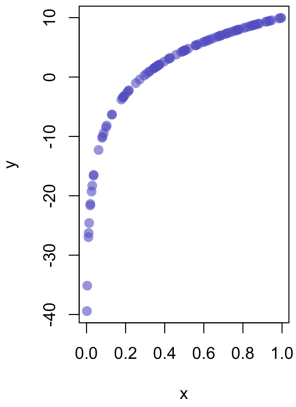
(a) $y = a * sqrt(x) + b$



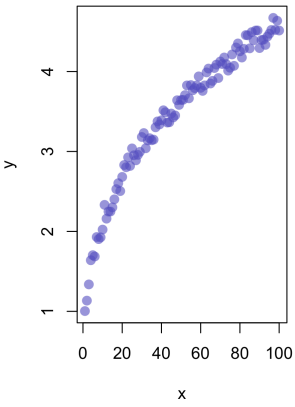
(b) $y = a * x^2 + b * x + c$



(c) $y = a * (exp(p * x))$

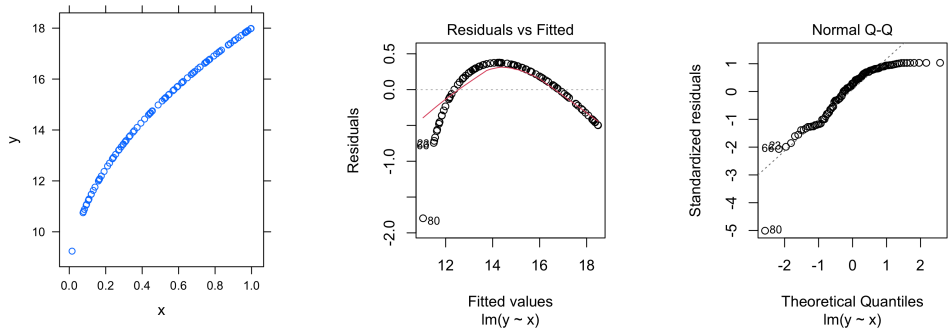


(d) $y = a + b * log(x)$



(e) y = jitter($x^p, factor = length(x)/2$)

Figure 1: Scatter plots of each equation.



(a) (b) (c)

Figure 2: Scatter plot of equation (a), Residuals vs. Fitted plot (b), and Normal Q-Q plot (c) of polynomial equation.

2.1 Polynomial

Polynomial equations are the most flexible tool to describe linear parameters, and can also be fitted with linear regression. The equation used in this part is Equation 1,

$$y = a * sqrt(x) + b \tag{1}$$

where:

`a` and `b` are fixed variables,

`x` random uniform variable.

In Figure 2 we can see the scatter plot of the equation 2a, the residuals versus fitted values 2b, and the normal Q-Q plot 2c. We used both values of `x, y` to build a data frame. Then, we pretend that we do not now the fixed variables we put in the equation. With the plots in 2 we can asees if the fit for a linear model is appropriate or not.

As stated before, we are using the Tukey ladder of powers to determine an approximate transformation of the `y` variable. With the `manipulate` library in R we automate the process of searching for an appropriate value for lambda. Figure 3 shows 13 iterations.

The best fit for a linear model is Figure 3a with λ= -3.

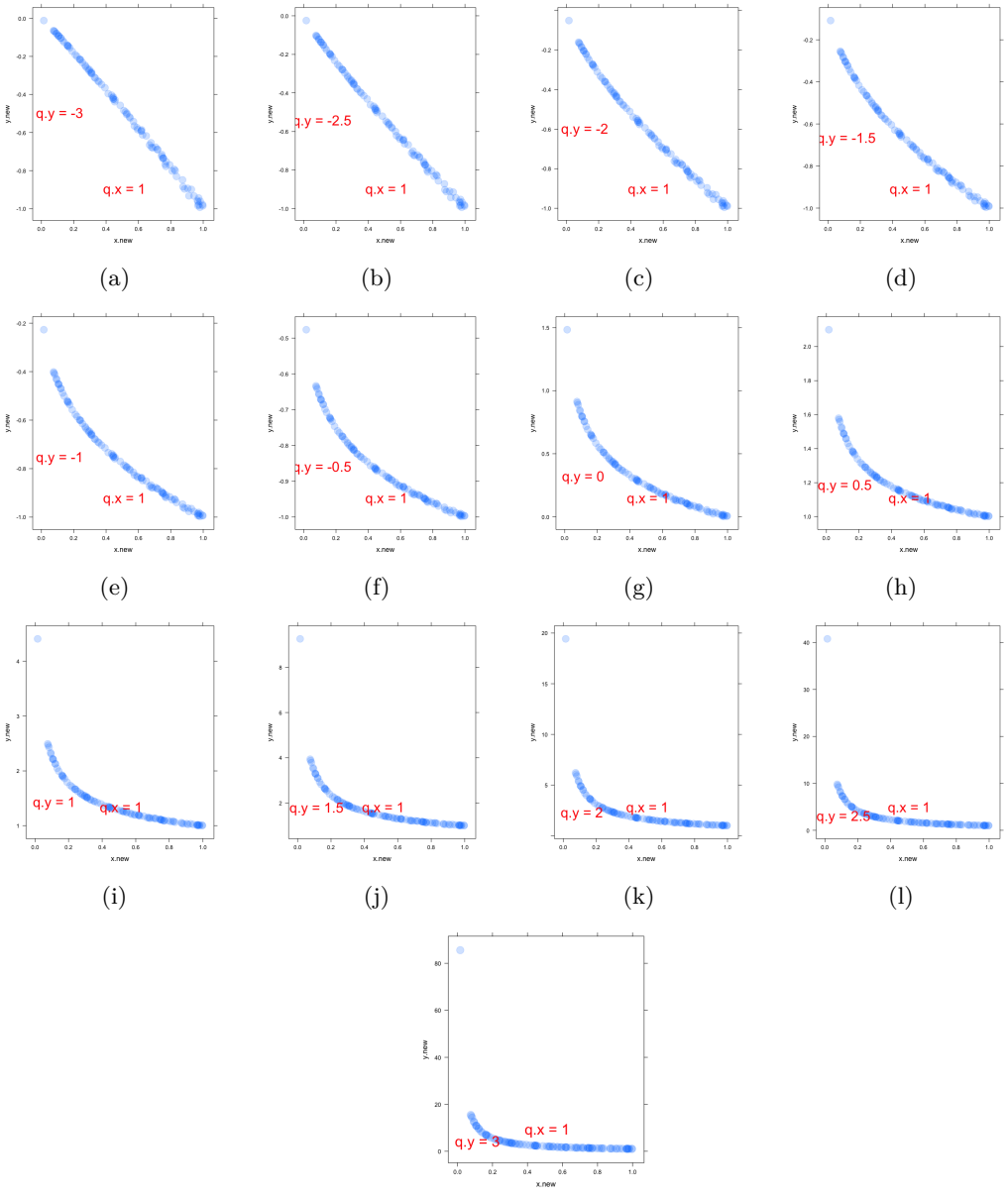
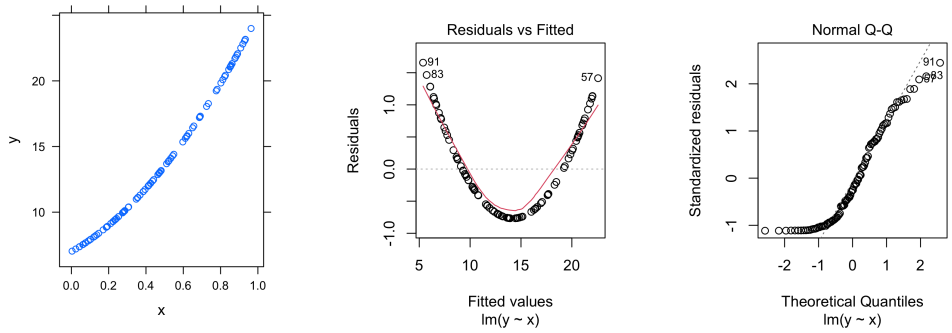


Figure 3: Iterations of the different values of lambda for the Tukey ladder of powers for the polynomial equation.



(a) (b) (c)

Figure 4: Scatter plot of equation (a), Residuals vs. Fitted plot (b), and Normal Q-Q plot (c) of quadratic equation.

2.2 Quadratic

The equation used for this is Equation 2.

$$y = a * x^2 + b * x + c \tag{2}$$

where all the fixed values are the same as the one used in Section 2.1.

In Figure 5 we have the iterations of the Tukey ladder. Examining this plot, Figure 5i is the one that shows a linear behaviour. In this case, $\lambda = 1$.

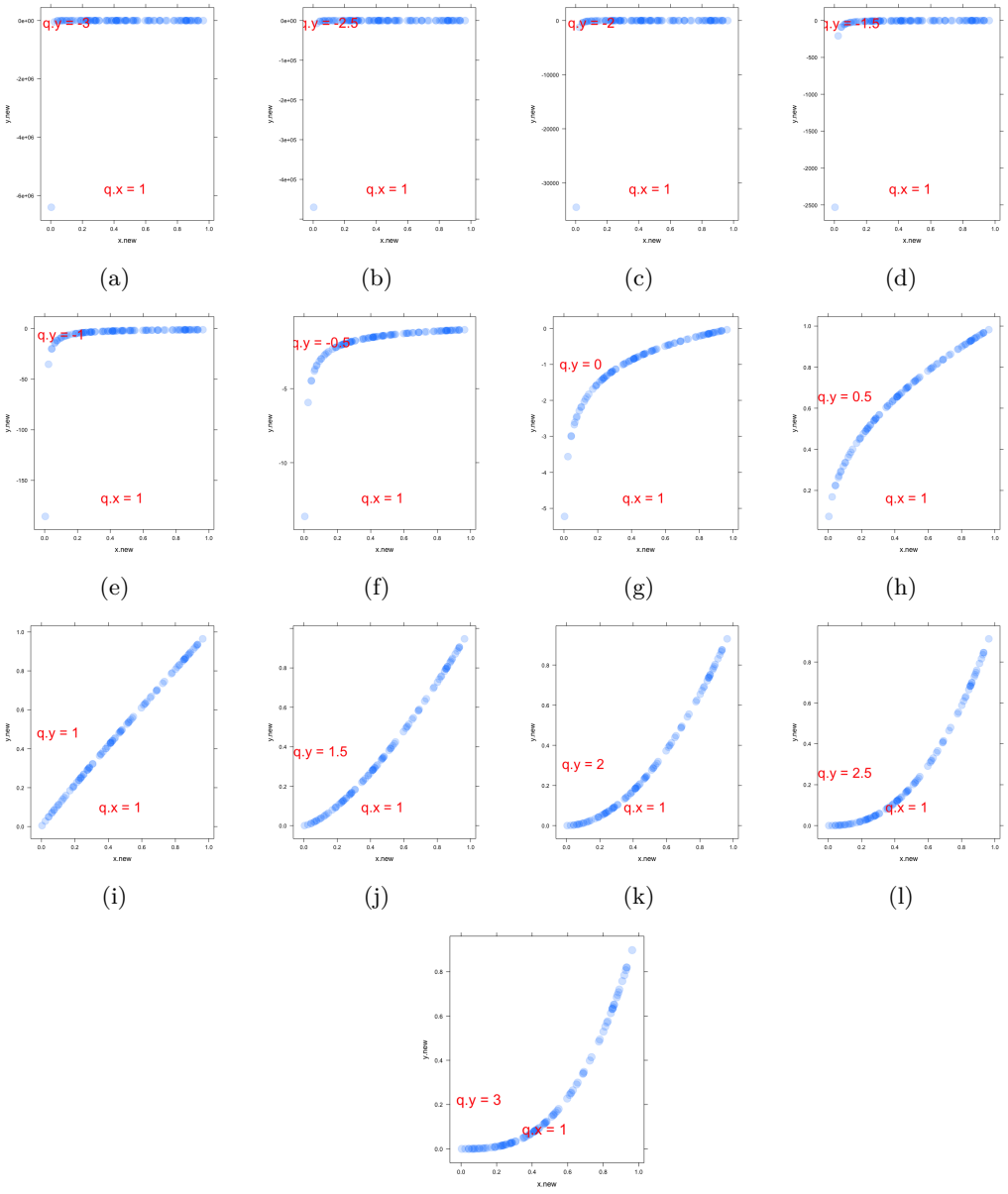
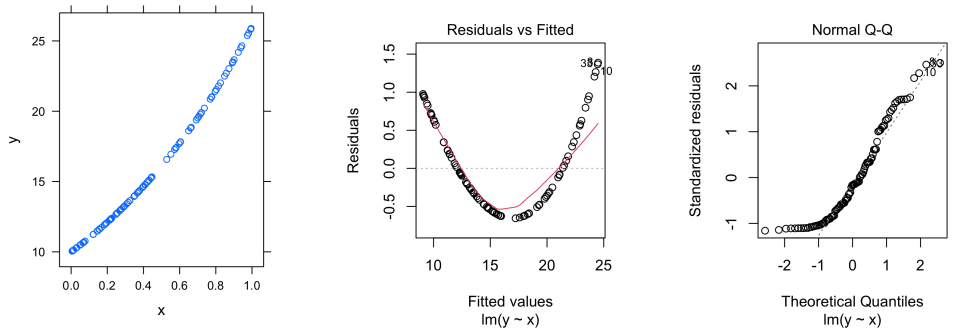


Figure 5: Iterations of the different values of lambda for the Tukey ladder of powers for the quadratic equation.



(a) $y = a * sqrt(x) + b$ (b) $y = a * x^2 + b * x + c$ (c) $y = a * (exp(p * x))$

Figure 6: Scatter plot of equation (a), Residuals vs. Fitted plot (b), and Normal Q-Q plot (c) of exponential equation.

2.3 Exponential

In the case of the exponential equation, it describes an increasing or decreasing trend, with a constant relative rate. Equation 3 shows the parameters used,

$$y = a * (exp(p * x)) \tag{3}$$

where all the constant variables are the same used in the polinomial and quadratic equations. The value of `p` is a standard normal random variable.

Seeing Figure 7 we see that the closer linear plot is between the values of Figure 7f and Figure 7g, so instead of showing the value of $\lambda = 3$, we plot the value between λ = -0.5 and λ = 0, and in Figure 7m we use the $\lambda = 0.25$. This shows to be the more linear in counts of behavior.

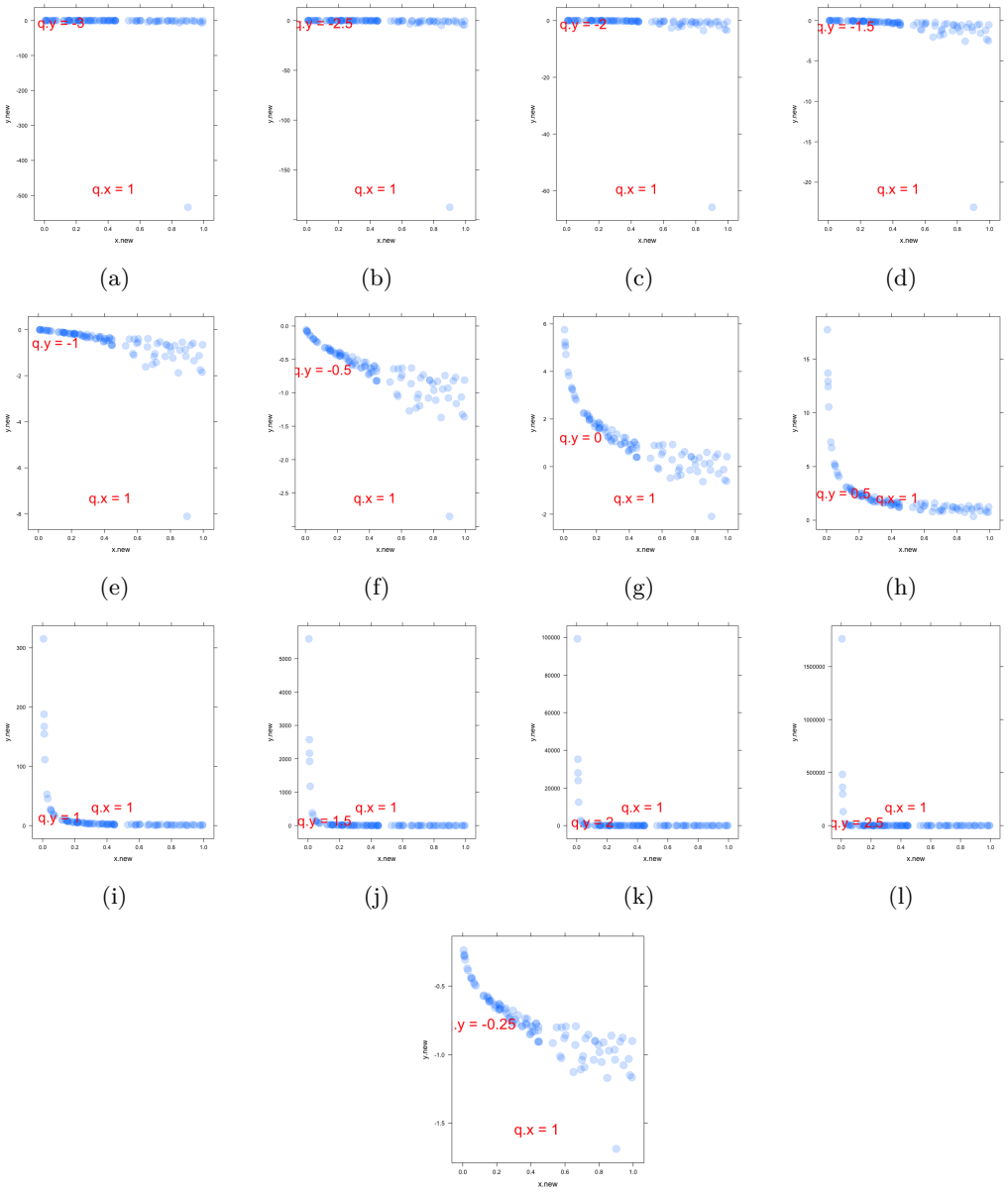
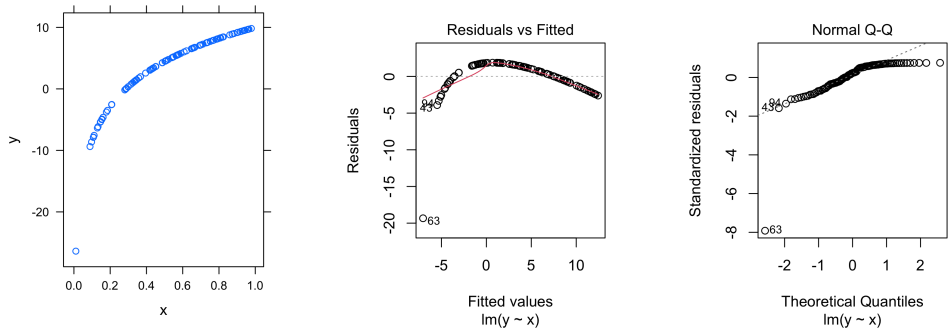


Figure 7: Iterations of the different values of lambda for the Tukey ladder of powers for the exponential equation.



(a) (b) (c)

Figure 8: Scatter plot of equation (a), Residuals vs. Fitted plot (b), and Normal Q-Q plot (c) of logarithmic equation.

2.4 Logarithmic

In this case, because of the logarithmic properties, `x` must be > to 0. Equation 4 shows the parameters used,

$$y = a + b * log(x) \tag{4}$$

where b is the parameter that determines the shape of the plot, and there are the same fixed variables used in the polynomial, quadratic and exponential experiments.

For the logarithmic equation, in Figure 9 we analyze the iterations of lambda and see that Figure 9e is the one with a linear behaviour, being λ = -1.5.

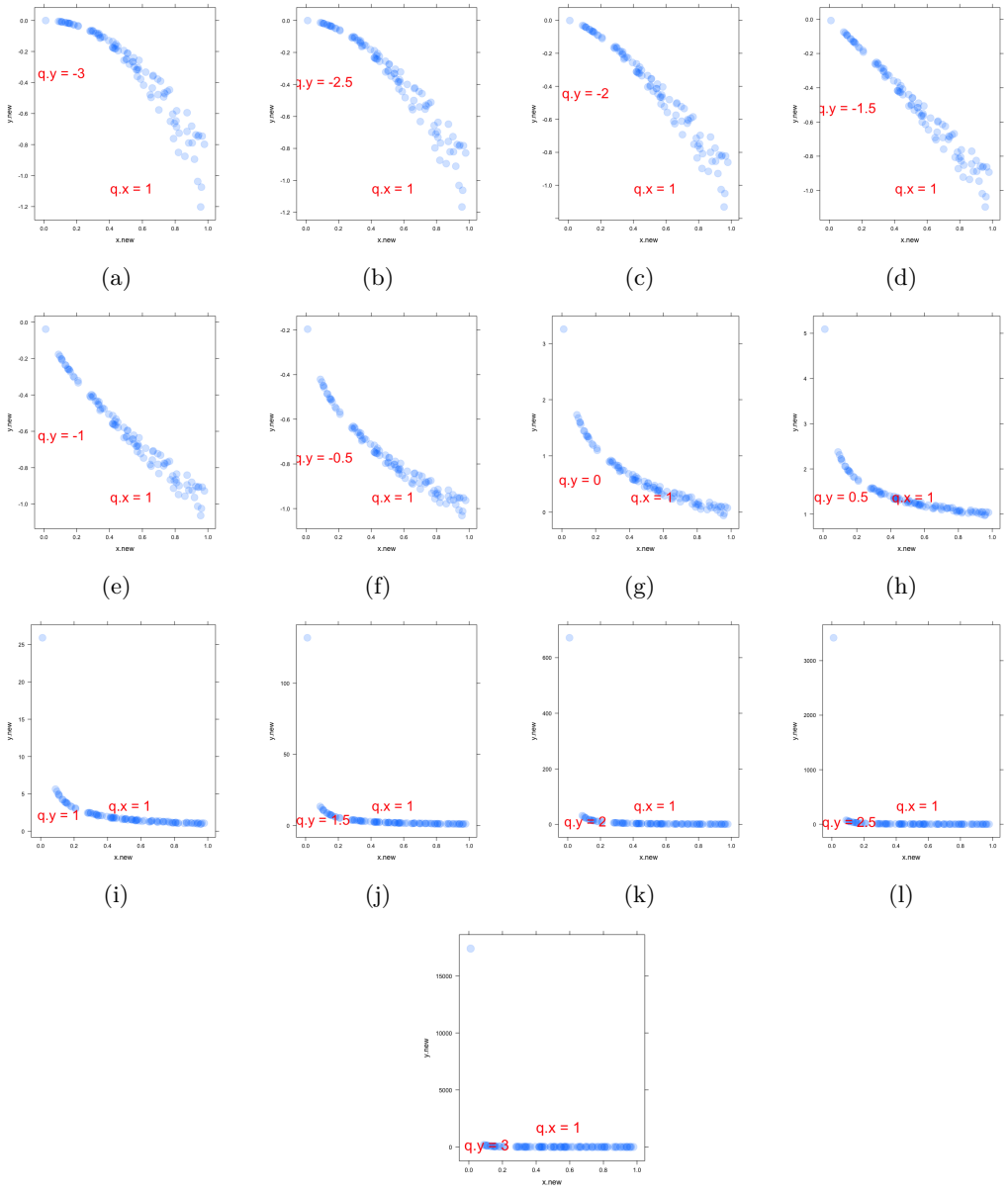
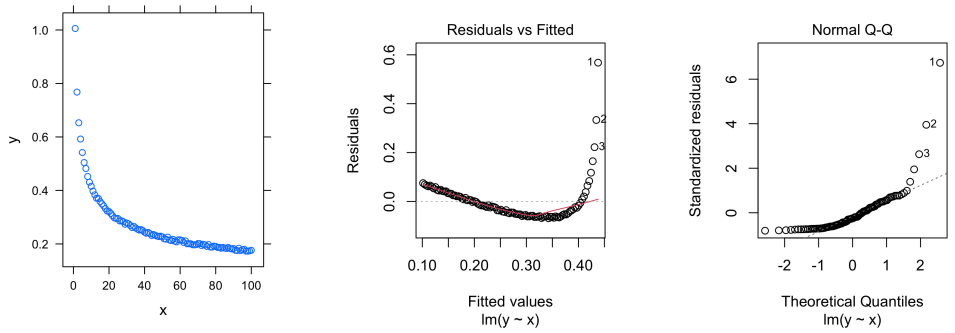


Figure 9: Iterations of the different values of lambda for the Tukey ladder of powers for the logarithmic equation.



(a) (b) (c)

Figure 10: Scatter plot of equation (a), Residuals vs. Fitted plot (b), and Normal Q-Q plot (c) of experiment with different `x`, and `y` with a random variable.

2.5 X with a different distribution.

For this experimentation we use the Equation 5,

$$y = jitter(x^p, factor = length(x)/2) \tag{5}$$

`p` is a standard normal random variable,

`x` is a number of 1:100,

`jitter` returns a numeric value of the same lenght as `x`, but with an amount of noise added in order to break ties.

In Figure 11 we apreciate that from all the iterations, Figure 11c is the one showing linear behaviour, with a λ = -2.

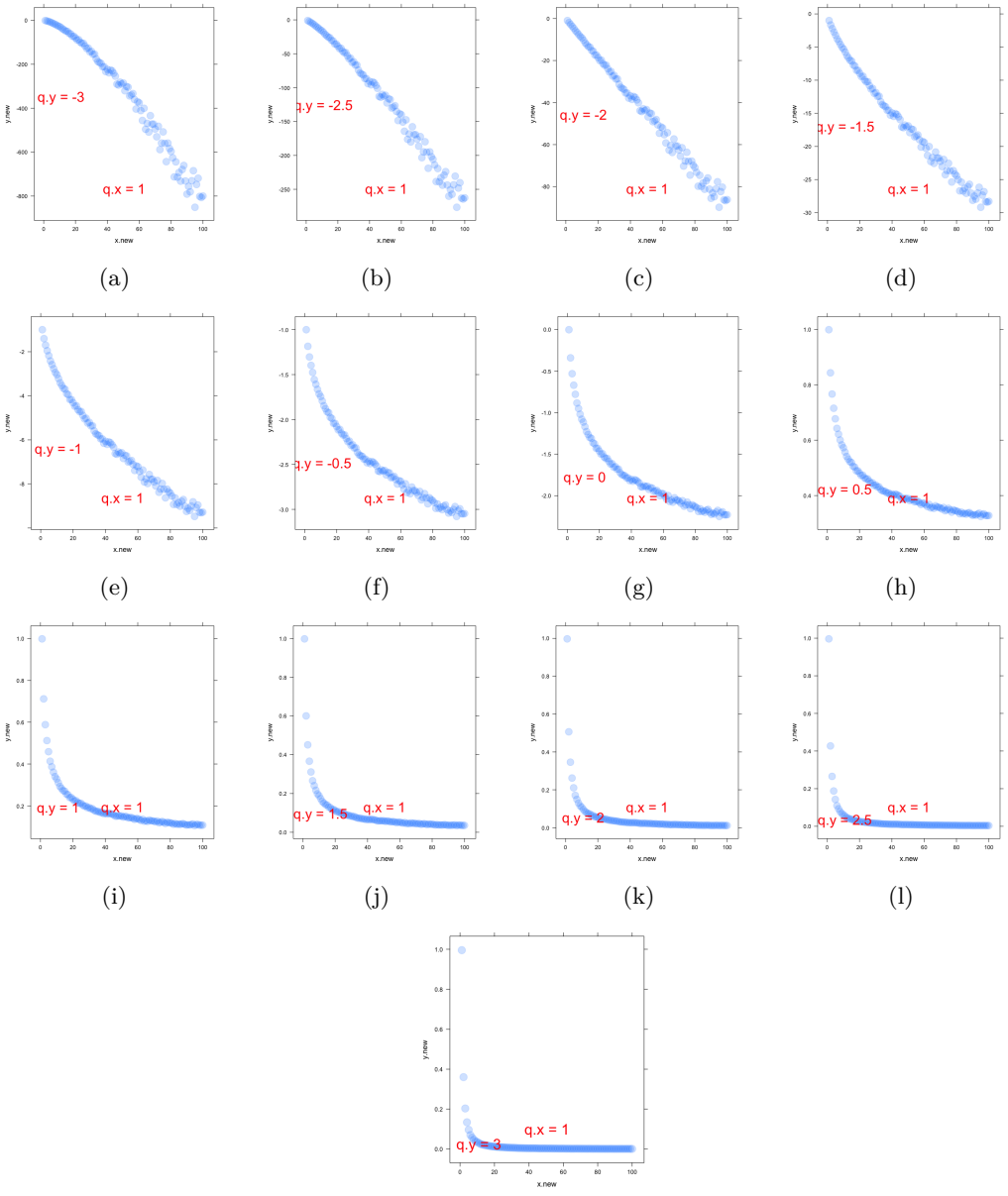
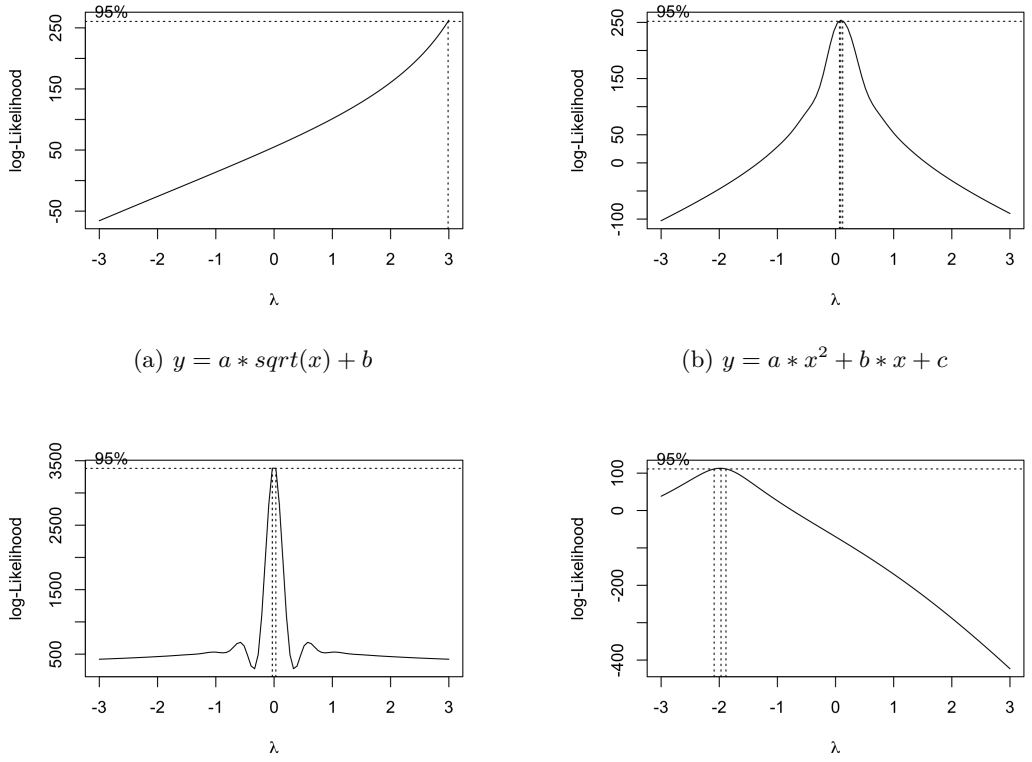


Figure 11: Iterations of the different values of lambda for the Tukey ladder of powers for the equation with different `x` distribution, and `y` with added noise.



(a) $y = a * sqrt(x) + b$

(b) $y = a * x^2 + b * x + c$

(c) $y = a * (exp(p * x))$

(d) $y = jitter(x^p, factor = length(x)/2)$

Figure 12: Correlations with the different values of λ.

Figure 12 shows the correlations of the different values of lambda working with the different equations used for this work.

3 Conclusions

Starting this work was a bit complicated for me, because I did not fully understand the transformations, but seeing them as an example first helped me understand what it was happening on each iteration. I had to use a built in library to work with the Tukey ladder, and did not use the box-cox transformation, but I think I could grasp the concept of either one a bit better having finished this work.

References

[1] Correlation test between two variables in r. http://www.sthda.com/english/wiki/correlation-test-between-two-variables-in-r. Accessed: 2020-10-20.

[2] David M. Lane. *Online Statistics Education: An Interactive Multimedia Course of Study*. Rice University.

Practice 8: Bayesian Theorem

Mayra Cristina Berrones Reyes 6291

October 27, 2020

1 Introduction

In this work, we will discuss a lot about the bayesian theorem, specifically aimed to review some of the information available to the general public, and some to data enthusiast, about the current COVID-19 pandemic.

Before we begin with the discussion of some of the links shared in class, and associated information within, it is very important to remark on some of the subjects that we are going to be using recurrently along this work, and that is the use of appropriate metrics to rate the results of the experiment at hand.

Figure 1 shows the confusion matrix for each of the classical metrics. In all cases, TP (True positives) represents if the result of the test is positive, and the patient actually has COVID. The same applies to TN (True negatives) in which, the test turns out negative, so the patient is healthy. FP (False positives) are the cases when the patient is healthy and the test says he has COVID. FN (False negatives) are when the test shows no sign of the disease, but the patient has it [7].

The first metric we have is the accuracy, which confusion matrix is depicted as a) in Figure 1. It computes the percentage of correct predictions over all kinds of predictions made, see Equation (1). This is a good indicator of performance when the data we have is balanced. In our case, if the number of cases positive with COVID and without them is nearly the same.

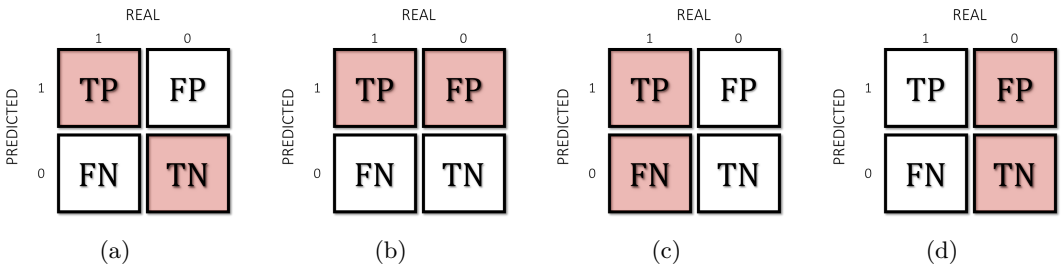


Figure 1: This figure represents the confusion matrix for: a) Accuracy, b) Precision, c) Recall or Sensitivity, and d) Specificity. In each confusion matrix, the colored boxes represent the features in which the metrics relay more heavily.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

The next metric is precision (Equation (2)) which computes the proportion of positive predictions being positive. It tells us about the success probability of making a correct positive test.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The recall, Equation (3), explains how sensitive the model is towards identifying the positive class of the tests. It is the proportion of true positives cases that were classified as positive, it is also known as the true positive rate.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

Specificity, Equation (4), is the opposite of Recall and focuses on the proportion of the negative cases that were classified as negative by the network; thus, it is a measure of how well the classifier identifies negative cases. It is also known as the true negative rate.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

Lastly, we have the F1 score.This metric combines precision and recall relative to a specific positive class, as seen in Equation (5). The F1 score can be seen as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{5}$$

2 Preliminary discussion

Now that we are clear about the different metrics used to evaluate the results, we begin the discussion of the multiple links shared in class. Without going in a specific order, this articles show some very interesting information about how we interpret test results from COVID-19, and introduces the highly important notion that the current way of testing and gathering results, may not be the optimal solution to accurately portray the behavior of the virus.

Testing for COVID is a privilege not many country have, and even in places where they are available, they are scarce, and not accessible enough for the general public because of the pricing. The present way of testing for COVID is shown in Table 1.

In this case, the standard reference for the diagnosis of COVID are the RT-PCR or molecular tests. As shown in Table 1 they have better accuracy than the other two. The alternative tests

Table 1: COVID testing basic information.

	Molecular test	Antigen test	Antibody test
Also known as:	Diagnostic test, RT-PCR	Rapid diagnostic test	Serological test, blood test
How is taken:	Nasal or throat swab, saliva	Nasal or throat swab	Finger stick or blood draw
Is another test needed:	This test is usually highly accurate and usually does not need repeat	Positive results are highly accurate. May need confirmation of molecular test.	Sometimes a second antibody test is required
What it shows:	Diagnosis active COVID infection	Diagnosis active COVID infection	shows if you have been infected by COVID in the past
What it does **not** do:	Show if you ever had COVID or were infected in the past.	Definitively rule out active COVID infection. Antigen test are more likely to miss an active virus infection compared to molecular tests.	Diagnose active COVID infection at the time of the test, or show that you do not have COVID

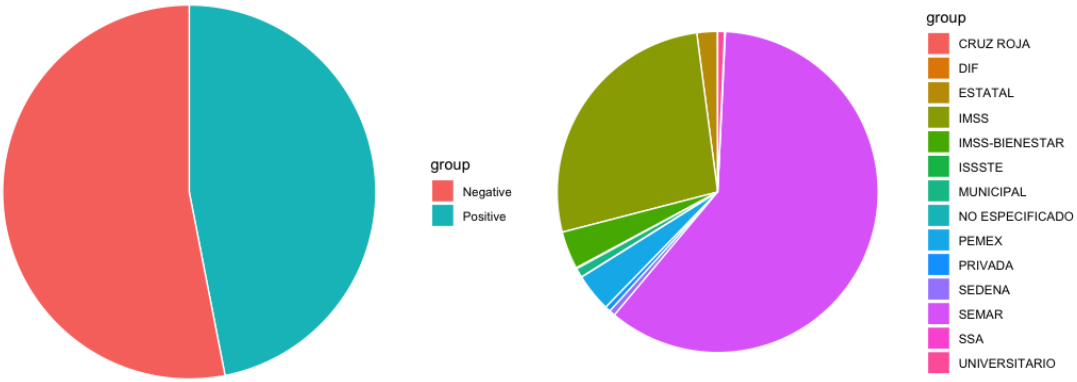
to this are the Antigen test, and antibody test. Both of this test have fairly accurate positive results, but if they come back negative, the general recommendation is to have a molecular test done.

Both molecular and antigen tests are diagnostic tests, which means that they show in their results if you have an active infection of COVID. The difference between the two is that the molecular test help to detect the genetic material of the virus, whilst the antigen tests detect specific proteins on the surface of the virus [2].

So what does that mean for the regular folk? It means that generally speaking, the antigen test can give a faster diagnostic on an active infection of COVID than a molecular test, but they have a higher probability of having a false negative.

Same goes with the antibody tests. They offer a fast and cheaper version of the test, but it should not be used to diagnose an active infection, since they only can detect if our immune system has developed antibodies to fight the virus, not the virus itself. This is tricky, because the human body takes up several weeks to develop enough antibodies to be detected in this test. If one is a target person for COVID, they may not have the luxury of waiting a few weeks to see their results.

With all of this instances we are presented with another point of view in regard of the result of the test and their reliability. In México it is common to find the mentality of, do not fix it if it is not broken, so when it comes to health, many of the people testing for COVID are likely to have shown symptoms before going to the hospital. Some of the articles presented in class mention the notion of pre-test probability, which encircle the cases that are more likely or less likely to show positive results on a COVID test, comparing between cases of people highly exposed to the virus, versus people that respected quarantine, but still show symptoms.



(a) (b)

Figure 2: Total of positive and negative cases in Mexico (a) and distribution of all cases in different health institutions (b).

As for the probability that the test are reliable, some articles mention that the sensitivity of the molecular test is of 63%, and in the case of antigen test is 31% for detecting an active infection. As stated before, an antibody test does not test active infections, but in order to be effective, the patient has to wait at least a week before the symptoms to perform this test, and up to 15 days, for it to perform optimally. It makes an excellent test to have post disease because it shows if your body is correctly fighting the virus, and in time could give us pointers in case of a new outbreak [8].

Because this last test is too dependent on time and various others factors, it is hard to find a source that tells us with certainty the percentage of reliability of this test. Performed after the 15 days or more after the symptoms, the accuracy can go from 75 to 90 percent [1].

In all this cases they use the bayesian theorem to revise the result of probability that a test may show in the results. The metric used generally is the sensitivity or recall.

3 Experimentation

For the experimentation in this work, we took a data base from Kaggle [4], that represents the cases reported in Mexico up to the month of October of 2020.

Using some of the columns in the file, in Figure 2a we can appreciate a plot with the distribution of all positive and negative tests results, and on Figure 2b the distribution of the total of cases in each hospital institution. The hospital plot helped us with the investigation of the COVID tests permitted in Mexico.

The two we found some information of were the Architect SARS CoV-2 IgG of Abbot Laboratories Inc [6], with a sensitivity of 95.8% and specificity of 99% [3]. This is the more expensive

Table 2: Table of predictions for the Architect SARS CoV-2 IgG test.

	1	0
1	355,144	15,570
0	4,194	415,157

one, and as far as we could search, not available in Tamaulipas (state where we researched more because of personal reasons). The other, more available is the MAGLUMI 2019-nCoV IgG (CLIA) of Shenzhen New Industries Biomedical Engineering [6] with 69.9% on sensitivity and 97.5% of specificity [3].

In the case of antibody tests, we could not find the right one, because they are done in private labs, since is blood work. The only thing we could find was its price, and it rounds between 1,000 and 2,000 Mexican pesos, depending on the urgency of the results and the laboratory it is made on (Again, this information could be only be verified in Tamaulipas).

Taking into consideration the experiment made in one of the links shared in class [5] we could replicate the procedure, using the data we could gather from the data set, and the reliability of the tests found.

According to our data set, we have 370,713 positive cases and 419,350 negative cases of all tested subjects. In total we have 790,063 tests. Giving this data the probability of specificity and sensibility of the Architect SARS CoV-2 IgG, we have the results of Table2.

- Accuracy: 0.9749
- Precision: 0.9580
- Sensitivity: 0.9883
- Specificity: 0.9638

Then we have the Bayes theorem:

$$\begin{aligned} P(A \mid B) &= \frac{P(A)P(B \mid A)}{(P(A)P(B \mid A) + P(notA)P(B \mid notA))} \\ &= \frac{0.454822 * 0.958}{(0.454822 * 0.958) + (0.545178 * 0.042)} \\ &= \frac{0.4357195}{0.458017} = 0.95007 \end{aligned} \tag{6}$$

Following the experiment, we calculate the other metrics with the information from Table 3 for the MAGLUMI 2019-nCoV IgG (CLIA) test.

- Accuracy: 0.8454

Table 3: Table of predictions for the MAGLUMI 2019-nCoV IgG (CLIA) test.

	1	0
1	259,128	111,584
0	10,483	408,866

- Precision: 0.6990
- Sensitivity: 0.9611
- Specificity: 0.7856

Then we have the Bayes theorem for the second test:

$$\begin{aligned} P(A \mid B) &= \frac{P(A)P(B \mid A)}{(P(A)P(B \mid A) + P(notA)P(B \mid notA))} \\ &= \frac{0.341252533 * 0.6990}{(0.341252533 * 0.6990) + (0.658747467 * 0.3010)} \\ &= \frac{0.2385355206}{0.4368185082} = 0.54607466 \end{aligned} \tag{7}$$

After this experiments we can clearly see the difference in reliability on both tests. In the Architect SARS CoV-2 IgG test, if we test positive, there is a probability of 95% that we have COVID. In change, with the MAGLUMI 2019-nCoV IgG (CLIA) test, if we test positive, there is a 54% probability that that diagnosis is correct.

4 Conclusions

The results of both experiments match the description that we had in mind, because we asked a health professional about the difference in testing, and, although he did not know the exact probability, the numbers are close enough to what he told us where the accuracy of the test.

In Tamaulipas, the standard way of testing is via antibodies if you are not a person in risk (Age, pre existing conditions, pregnant, etc) because in public hospitals, such as the IMSSS, ISSSTE, IPSET, etc., they have a very limited amount of testing material. The expert we contacted for information, can verify that they only have kits for rapid testing, so their results would be more like the MAGLUMI 2019-nCoV IgG (CLIA) test.

This work can help highlight the importance of having better testing equipment, since we are approaching the time of year when influenza and flu cases arise, as well as cases of dengue, because of all the rains of following months. Hospitals will be once again overflowed with people worried that they may be sick from COVID, and with test with poor reliability, resources like respirators and medicine, will become scarce again from diseases that may not even be COVID.

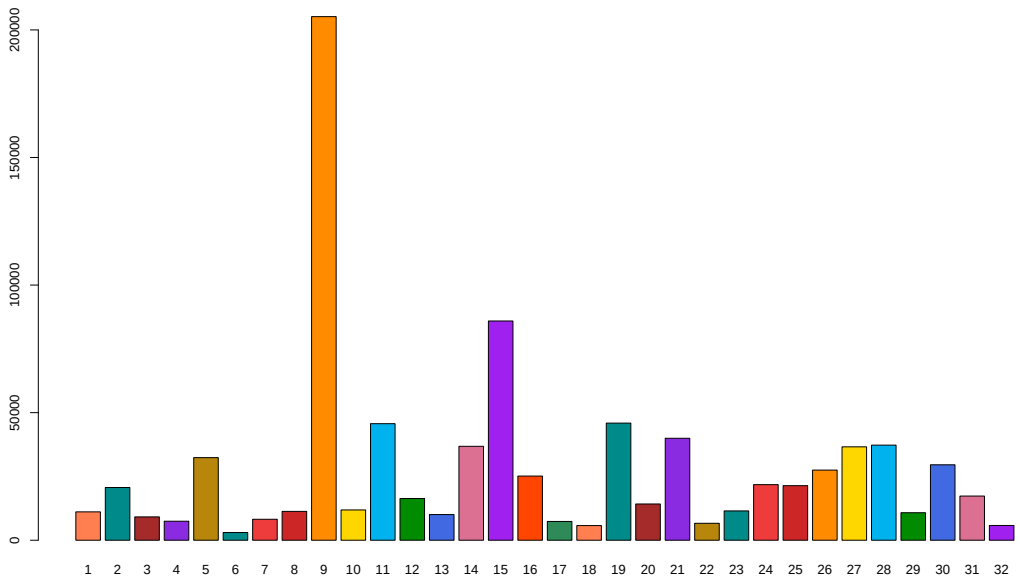


Figure 3: This plot represents all of the states in Mexico, and the total of cases reported in each one. The number represents a different federative state.

As a final thought, this work helped clear some things about COVID. In all the investigation we stumbled on, we found out that getting sick and recover does not mean that you are immune now (contrary to what some foreign president may say). And although it was quite an extensive work comparing sources and fishing everywhere for some information, I am glad that in most of the easy to find data does not have alarming head lines like, testing for COVID is unreliable.

If nothing else, this will help me explain to my family why and what is useful information about COVID, so they do not believe every gossip they pass on group chats.

5 Extras

Since we began experimenting a little bit with R and data mining (Python was the preferred method) we made two plots that did not quite fit with the rest of the experimentation, but they showed some important information. For example, in Figure 3 we see the distribution of positive cases of COVID.

In Figure 4 we were curious about the new information in the news, that not only older people were at risk of contracting COVID. We made an histogram with intervals of ages, and found that the most recurrent age of positive cases lies in the interval of 30 to 40 years old. This of course can be because that is the demographic that is more likely to have to go to work, and thus has more chances of getting infected.

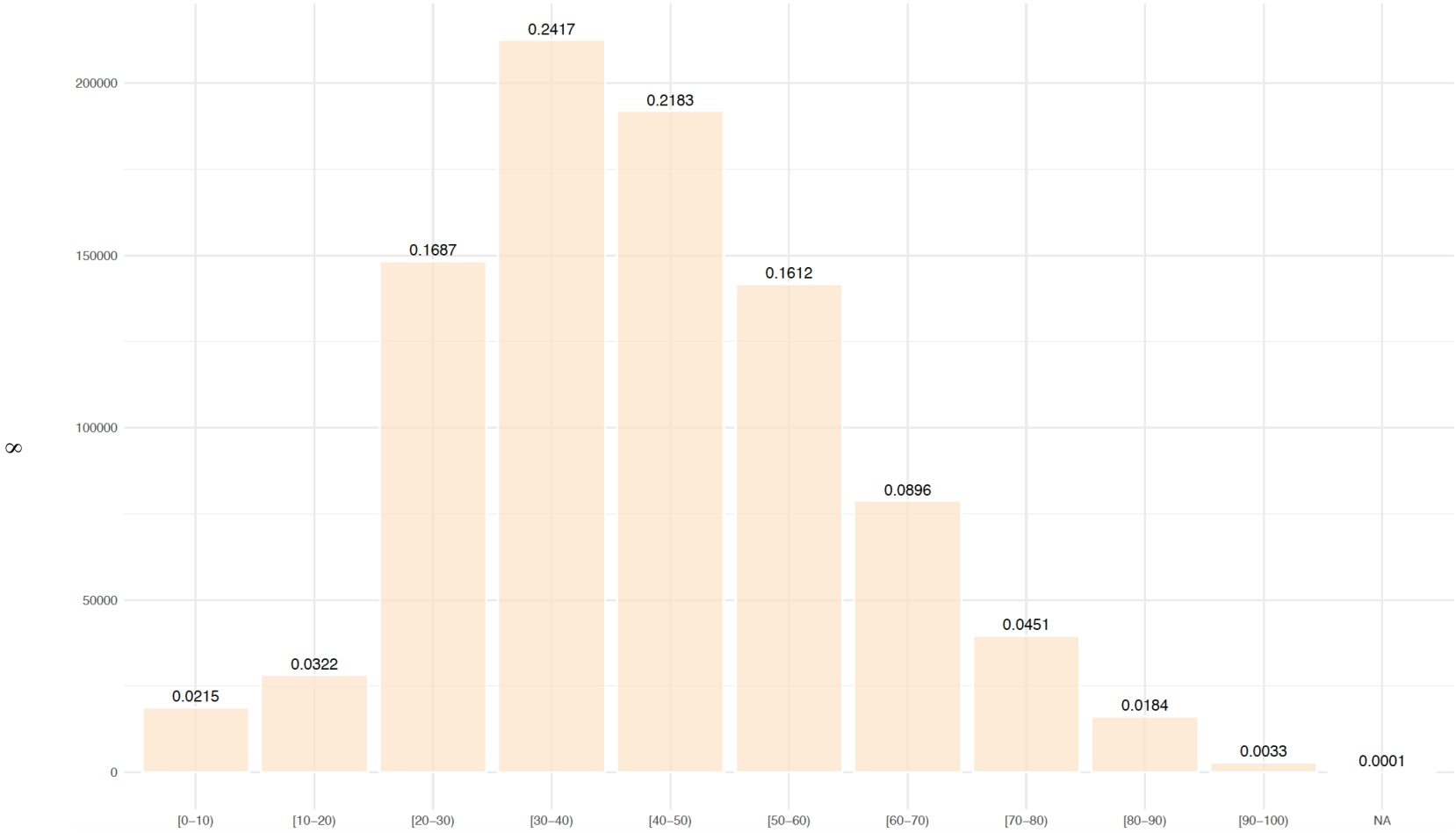


Figure 4: Distribution of cases by age groups in Mexico.

References

[1] Studies find varying accuracy of covid19 antibody tests. https://www.contagionlive.com/view/studies-find-varying-accuracy-of-covid19-antibody-tests. Accessed: 2020-10-27.

[2] Conceptos basicos para las pruebas del coronavirus. https://www.fda.gov/media/138239/download, . Accessed: 2020-10-27.

[3] Eua authorized serology tests performance. https://www.fda.gov/medical-devices/coronavirus-disease-2019-covid-19-emergency-use-authorizations-medical-devices/eua-authorized-serology-test-performance, . Accessed: 2020-10-27.

[4] Covid-19 mx. https://www.kaggle.com/lalish99/covid19-mx. Accessed: 2020-10-27.

[5] Covid-19 bayes theorem and taking probabilistic decisions. https://towardsdatascience.com/covid-19-bayes-theorem-and-taking-data-driven-decisions-part-1-b61e2c2b3bea. Accessed: 2020-10-27.

[6] Pruebas de covid aprobadas en mexico. https://www.xataka.com.mx/medicina-y-salud/pruebas-rapidas-covid-aprobadas-mexico-estas-19-test-avalados-cofepris-para-saber-he-tenido-c Accessed: 2020-10-27.

[7] Mayra C. Berrones-Reyes. Clasificador de mamografias por medio de redes neuronales convolucionales. http://eprints.uanl.mx/17656/.

[8] Gar Ming Chan. Bayes' theorem, covid19, and screening tests. *The American Journal of Emergency Medicine*, 2020.

Practice 9: Exercises of expected value and variance of random variables

Mayra Cristina Berrones Reyes 6291

November 3, 2020

1 Exercises

The exercises of this work where taken from the book Introduction to probability by Charles M. Grinstead and J. Laurie Snell [1].

1.1 Exercise 1, page 247

A card is drawn at random form a deck consisting of cards numbered from 2 to 10. A player wins 1 dollar if the number on the card is odd and loses 1 dollar if the number is even. What is the expected value of his winnings?

- **Answer:**

There are in total 9 cards numbered from 2 to 10, so we have a $n = 9$. The sample space we have for odd and even cards are $n\_odd = (3, 5, 7, 9)$ and $n\_even = (2, 4, 6, 8, 10)$.

We want to know the expected value of a winning. For that we need the probability of winning and losing, as we can see in Equation 1 and 2.

$$w\_prob = \frac{n\_odd}{n} = \frac{4}{9}, \tag{1}$$

$$l\_prob = \frac{n\_even}{n} = \frac{5}{9}. \tag{2}$$

Now we can calculate the expected value of the winnings in Equation 3.

$$E(X) = 1 * \left(\frac{4}{9}\right) - 1 * \left(\frac{5}{9}\right) = -\frac{1}{9} \tag{3}$$

■

Table 1: Calculation of probabilities for the expected value of X.

X value	2	3	4	5	6	7	8	9	10	11	12
Probabilities of each one	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
Simplifying fractions	$\frac{1}{36}$	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{9}$	$\frac{5}{36}$	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$

Table 2: Calculation of probabilities for the expected value of Y.

Y value	0	-1	-2	-3	-4	-5	1	2	3	4	5
Probabilities of each one	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
Simplifying fractions	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$

1.2 Exercise 6, page 247

A die is rolled twice. Let X denote the sum of the two numbers that turn up, and Y the difference of the numbers (specifically, the number on the first roll minus the number of the second). Show that $E(XY) = E(X)E(Y)$. Are X and Y independent?

- **Answer:**

So if we suppose that the dice is fair, our set of possible values are $values = 6^2 = 36$, with each possible outcome equally likely.

X could have the values shown in Table 1.

We can calculate the expected value of X in Equation 4.

$$\begin{aligned} E(X) =& 2 * \left(\frac{1}{36}\right) + 3 * \left(\frac{1}{18}\right) + 4 * \left(\frac{1}{12}\right) + 5 * \left(\frac{1}{9}\right) + 6 * \left(\frac{5}{36}\right) + 7 * \left(\frac{1}{6}\right) + \\ & 8 * \left(\frac{5}{36}\right) + 9 * \left(\frac{1}{9}\right) + 10 * \left(\frac{1}{12}\right) + 11 * \left(\frac{1}{18}\right) + 12 * \left(\frac{1}{36}\right) = 7. \end{aligned} \tag{4}$$

For Y we could have the values in Table 2.

With these probabilities we calculate the expected value of Y in Equation 5.

$$\begin{aligned} E(Y) =& 0 * \left(\frac{1}{6}\right) - 1 * \left(\frac{5}{36}\right) - 2 * \left(\frac{1}{9}\right) - 3 * \left(\frac{1}{12}\right) - 4 * \left(\frac{1}{18}\right) - 5 * \left(\frac{1}{36}\right) + \\ & 1 * \left(\frac{5}{36}\right) + 2 * \left(\frac{1}{9}\right) + 3 * \left(\frac{1}{12}\right) + 4 * \left(\frac{1}{18}\right) + 5 * \left(\frac{1}{36}\right) = 0. \end{aligned} \tag{5}$$

For the first part of the question, we need to show that $E(XY) = E(X)E(Y)$. We already have that $E(X)E(Y) = 0$, se we need to calculate the expected value of $E(XY)$. In Table 3 we

Table 3: Calculation of probabilities for the expected value of XY.

XY value	0	-3	-8	-15	-24	-35	3	-5	-12	-21	-32	8
Probabilities of each one	$\frac{6}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
XY value	5	-7	-16	-27	15	12	7	-9	-20	24	21	16
Probabilities of each one	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
XY value	9	-11	35	32	27	20	11					
Probabilities of each one	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$					

have the probabilities of all possible $E(XY)$.

$$\begin{aligned} E(XY) =& 0 * \left(\frac{6}{36}\right) - 3 * \left(\frac{1}{36}\right) - 8 * \left(\frac{1}{36}\right) - 15 * \left(\frac{1}{36}\right) - 24 * \left(\frac{1}{36}\right) - 35 * \left(\frac{1}{36}\right) + \\ & 3 * \left(\frac{1}{36}\right) - 5 * \left(\frac{1}{36}\right) - 12 * \left(\frac{1}{36}\right) - 21 * \left(\frac{1}{36}\right) - 32 * \left(\frac{1}{36}\right) + 8 * \left(\frac{1}{36}\right) + \\ & 5 * \left(\frac{1}{36}\right) - 7 * \left(\frac{1}{36}\right) - 16 * \left(\frac{1}{36}\right) - 27 * \left(\frac{1}{36}\right) + 15 * \left(\frac{1}{36}\right) + 12 * \left(\frac{1}{36}\right) + \\ & 7 * \left(\frac{1}{36}\right) - 9 * \left(\frac{1}{36}\right) - 20 * \left(\frac{1}{36}\right) + 24 * \left(\frac{1}{36}\right) + 21 * \left(\frac{1}{36}\right) + 16 * \left(\frac{1}{36}\right) + \\ & 9 * \left(\frac{1}{36}\right) - 11 * \left(\frac{1}{36}\right) + 35 * \left(\frac{1}{36}\right) + 32 * \left(\frac{1}{36}\right) + 27 * \left(\frac{1}{36}\right) + 20 * \left(\frac{1}{36}\right) + \\ & 11 * \left(\frac{1}{36}\right) = 0. \end{aligned} \tag{6}$$

So we can conclude now that $E(XY) = E(X)E(Y)$, because both equal 0.

The second part of the question asks if X and Y are independent values. For this there is a theorem of independence in the book mentioned at the beginning [1] that says that "*If X and Y are two random variables it is not true in general that $E(XY) = E(X)E(Y)$. However, this is true if X and Y are independent.*".

Since our X and Y are both random variables, and we just proved that $E(XY) = E(X)E(Y)$, we can conclude that our variables are independent.

■

1.3 Exercise 15, page 249

A box contains two gold balls and three silver balls. You are allowed to choose successively from the box at random. You win 1 dollar each time you draw a gold ball and lose 1 dollar each time you draw a silver ball. After a draw, the ball is not replaced. Show that, if you

Table 4: Different options of outputs on the different scenarios

1	0	-1
5	2	3

draw until you are ahead by 1 dollar, or until there are no more gold balls, this is a favorable game.

- **Answer:**

Since the balls can not be replaced in this experiment, we can only draw 5 balls in a certain order. When we order all possible scenarios of this draws, we come up with:

- When we have the first draw being a gold ball, we have four scenarios in which we comply with the instruction of ending the game by being ahead 1 dollar.
- When the draw of the first gold ball is in the second place, there is only one scenario in which we win 1 dollar, by drawing both gold balls, one when we end up with no winnings, but ends because we drew the two gold balls, and one last scenario, when we lose 1 dollar, because the last ball was gold.
- If the first gold ball is in the third place, we have one scenario when we have no winnings, and another when we lose 1 dollar.
- Lastly, if the first gold ball is in the fourth place, there is only one scenario where we lose 1 dollar.

Making a summary of this calculations, we have Table 4. There is only three options in output of winnings and loses. Either we win 1 dollar, we end up with nothing, or we lose 1 dollar.

So we have as an expected value Equation 7. All probabilities must sum up to 1, so we have all the probabilities multiplied by $\frac{1}{10}$.

$$E(X) = 1 * \left(\frac{5}{10}\right) + 0 * \left(\frac{2}{10}\right) - 1 * \left(\frac{3}{10}\right) = \frac{2}{10} = \frac{1}{5} \tag{7}$$

Since our expected value is positive, we can conclude that the second statement in the question, to wether it is a favorable game when there is no more gold balls, is correct. Another thing that can support this is, if we sum up the winnings with the neutral results, we have a $\frac{7}{10}$ of winnings versus $\frac{3}{10}$ probability of losses.

■

1.4 Exercise 18, page 249

Exactly one of six similar keys opens a certain door. If you try the keys, one after another, what is the expected number of keys that you will have to try before success?

- **Answer:**

For this experiment, we need to first calculate the probability of the process of elimination of each key. The probability of finding the right key in:

- First draw $= \frac{1}{6}$
- Second draw $= \frac{5}{6} * \frac{1}{5} = \frac{1}{6}$
- Third draw $= \frac{5}{6} * \frac{4}{5} * \frac{1}{4} = \frac{1}{6}$
- Fourth draw $= \frac{5}{6} * \frac{4}{5} * \frac{3}{4} * \frac{1}{3} = \frac{1}{6}$
- Fifth draw $= \frac{5}{6} * \frac{4}{5} * \frac{3}{4} * \frac{2}{3} * \frac{1}{2} = \frac{1}{6}$

In the sixth draw is a certainty that we will draw the correct key, because there is no other one. For the expected value of this experiment we have Equation 8

$$E(X) = 1 * \left(\frac{1}{6}\right) + 2 * \left(\frac{1}{6}\right) + 3 * \left(\frac{1}{6}\right) + 4 * \left(\frac{1}{6}\right) + 5 * \left(\frac{1}{6}\right) = \frac{5}{2} \tag{8}$$

■

1.5 Exercise 19, page 249

A multiple choice exam is given. A problem has four possible answers, and exactly one answer is correct. The student is allowed to choose a subset of four possible answers as his answer. If his chosen subset contains the correct answer, the student receives three points, but he loses one point for each wrong answer in his chosen subset. Show that if he just guesses a subset uniformly and randomly his expected score is zero.

- **Answer:**

For this, the way we understand the problem is that the universe of choice are the 4 multiple answers. In this case, if our whole set contains 4 elements ($n = 4$), then the number of subsets is $2^n = 16$. In Table 5 we describe the winning and losing of points depending on the subset we are working on. In this experiment, we will pretend that the correct answer is 1.

If we calculate the expected value of each case in Table 5 then we have Equations 9, 10, 11, 12 and 13.

$$E(X) = 0 \tag{9}$$

$$E(X) = 3 * \left(\frac{1}{4}\right) - 1 * \left(\frac{3}{4}\right) = 0 \tag{10}$$

Table 5: Different options of outputs on the different scenarios

0	{ }	Does not win points, so its value is 0
1	{1} {2} {3} {4}	We win 3 points $\frac{1}{4}$ and lose 1 point $\frac{3}{4}$ of the choices.
2	{1, 2} {1, 3} {1, 4} {2, 3} {2, 4} {3, 4}	We win 3 points $\frac{3}{6}$ of the times, and $\frac{3}{6}$ we lose 1 point.
3	{1, 2, 3} {1, 2, 4} {2, 3, 4} {1, 3, 4}	We win 1 point $\frac{3}{4}$ of the times, because of the 3 points, we need to extract 2 points for the wrong answers accompanying the right answer. And we loose 3 points $\frac{3}{4}$ of the times because of the three mistakes.
4	{1, 2, 3, 4}	We do not win any points because we have the 3 points for the right answer, but we have to extract 3 points for the wrong ones.

$$E(X) = 2 * \left(\frac{3}{6}\right) - 2 * \left(\frac{3}{6}\right) = 0 \tag{11}$$

$$E(X) = 1 * \left(\frac{3}{4}\right) - 3 * \left(\frac{1}{4}\right) = 0 \tag{12}$$

$$E(X) = 3 - 3 = 0 \tag{13}$$

With this we prove that if we chose any of the subsets, we have an expected value of zero.

■

1.6 Exercise 1, page 263

A number is chosen at random from the set $S = \{-1, 0.1\}$. Let X be the number chosen. Find the expected value, variance, and standard deviation of X.

- **Answer:**

For the expected value, we have a set of 3 elements, so each one has a probability of $\frac{1}{3}$. Our expected value is then shown in Equation 14,

$$E(X) = -1 * \left(\frac{1}{3}\right) + 0 * \left(\frac{1}{3}\right) + 1 * \left(\frac{1}{3}\right) = 0 \tag{14}$$

In the case of the variance, as a concept we have the variance as the expectation of the squared deviation of random variable from its mean. The formula for variance is in Equation 15

$$V(X) = E[X^2] - E[X]^2 = -1^2 * \left(\frac{1}{3}\right) + 0^2 * \left(\frac{1}{3}\right) + 1^2 * \left(\frac{1}{3}\right) - 0^2 = \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \tag{15}$$

The concept of standard deviation is the measure of the amount of variation or dispersion of a set of values. The formula is on Equation 16

$$D(X) = \sqrt{E[X^2] - E[X]^2} = \sqrt{-1^2 * \left(\frac{1}{3}\right) + 0^2 * \left(\frac{1}{3}\right) + 1^2 * \left(\frac{1}{3}\right) - 0^2} = \sqrt{\frac{1}{3} + \frac{1}{3}} = \sqrt{\frac{2}{3}} \quad (16)$$

■

1.7 Exercise 9, page 264

A die is loaded so that the probability of a face coming up is proportional to the number on that face. The die is rolled with outcome X. Find $V(X)$ and $D(X)$.

- **Answer:**

For this experiment, we need to first calculate the expected value of X. It says that the probability of each faace coming up is the number of the face. The sum of all probabilities must sum up to 1, so we can calculate probability k as Equation 17

$$\begin{aligned} (1,2,3,4,5,6)k &= 1 \\ 21k &= 1 \\ k &= \frac{1}{21}. \end{aligned} \quad (17)$$

We then calculate the expected value in Equation 18

$$E(X) = 1 * \left(\frac{1}{21}\right) + 2 * \left(\frac{2}{21}\right) + 3\left(\frac{3}{21}\right) + 4\left(\frac{4}{21}\right) + 5\left(\frac{5}{21}\right) + 6\left(\frac{6}{21}\right) = 13 \quad (18)$$

With the expected value we can now calculate the variance in Equation 19 and standard deviation in Equation 20.

$$\begin{aligned} V(X) &= E(X^2) - E(X)^2 \\ &= 1 * \left(\frac{1}{21}\right) + 2 * \left(\frac{2}{21}\right) + 3 * \left(\frac{3}{21}\right) + 4 * \left(\frac{4}{21}\right) + 5 * \left(\frac{5}{21}\right) + 6 * \left(\frac{6}{21}\right) - \left(\frac{13}{3}\right)^2 \\ &= 21 - \left(\frac{3}{21}\right)^2 = 21 - \left(\frac{169}{9}\right) = \frac{20}{9} \end{aligned} \quad (19)$$

$$D(X) = \sqrt{V(X)} = \sqrt{\frac{20}{9}} = \frac{2\sqrt{5}}{3} \quad (20)$$

■

1.8 Exercise 12, page 264

Let X be a random variable with $\mu = E(X)$ and $\sigma^2 = V(X)$. Define $X^* = (X - \mu)/\sigma$. The random variable X^* is called the standard random variable associated with X. Show that this standardized random variable has expected value of 0 and variance 1.

- **Answer:**

A standardized variable is sometimes called `Z-score` or standard score. Is a variable that has been rescaled to have a mean of 0 and a standard deviation of one. Using the properties of expectation that we are given in the problem, we have have Equation 21,

$$E(X^*) = E\left[\frac{X-\mu}{\sigma}\right] = \frac{1}{\sigma}[E(X) - \mu] = \frac{1}{\sigma}[\mu - \mu] = 0 \tag{21}$$

Then we calculate the variance in Equation 22,

$$\begin{aligned} V(X^{*2}) = E(X^{*2}) - E(X^*)^2 = E\left[\left(\frac{X-\mu}{\sigma}\right)^2\right] - 0^2 \\ = \frac{1}{\sigma}^2 [E(X^2) - 2\mu E(X) + \mu^2] \\ = \frac{1}{\sigma}^2 [E(X^2) - E^2(X) + E^2(X) - 2\mu E(X) + \mu^2] \\ = \frac{1}{\sigma}^2 [V(X) + \mu^2 - 2\mu^2 + \mu^2] \\ = \frac{1}{\sigma}[\sigma^2 + 0] = 1. \end{aligned} \tag{22}$$

■

References

[1] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

Practice 10: Simulations of expected value and variance of random variables

Mayra Cristina Berrones Reyes 6291

November 10, 2020

1 Exercises

The exercises of this work where taken from the book Introduction to probability by Charles M. Grinstead and J. Laurie Snell [1].

1.1 Exercise 1, page 247

A card is drawn at random form a deck consisting of cards numbered from 2 to 10. A player wins 1 dollar if the number on the card is odd and loses 1 dollar if the number is even. What is the expected value of his winnings?

- **Experimentation:**

For this experiment, we used the `sample` function in `R`. Inside a loop of 10,000 repetitions, we made a variable that gives us at random a number between 2 and 10. Then we divided the results in even and odd numbers. In Figure 1a we can see a pie plot of one iteration of the 10,000 repetition of the card experiment. In this case, the red represents the number of times the card was an even number, and the blue represents the odd ones. At first sight we can see that the losses are greater than the winnings.

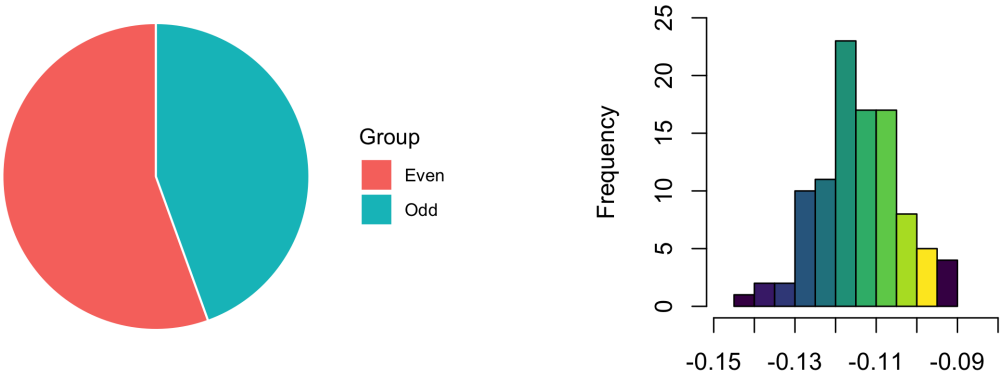
The expected value of this single experiment was $E(X) = -0.116$. The value resembles the result of our analytic experiment. To test it further, we performed 100 iterations of the 10,000 repetitions. In this case, Figure 1b is the histogram showing how the results are distributed. As we can see, the majority of the experiment land in the range of -0.12 and -0.10, proving with experimentation, that our analysis was correct.

■

1.2 Exercise 6, page 247

A die is rolled twice. Let X denote the sum of the two numbers that turn up, and Y the difference of the numbers (specifically, the number on the first roll minus the number of the second). Show that $E(XY) = E(X)E(Y)$. Are X and Y independent?

- **Experimentation:**



(a) Pie plot of the one iteration of the 10,000 repetitions of the card experiment.

(b) Histogram of the 100 iterations of the 10,000 repetitions of the card experiment.

Figure 1: Pie plot and histogram of the experiment.

Table 1: Calculation of probabilities for the expected value of X.

X value	2	3	4	5	6	7	8	9	10	11	12
Probabilities of each one	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
Simplifying fractions	$\frac{1}{36}$	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{9}$	$\frac{5}{36}$	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$

Table 2: Calculation of probabilities for the expected value of Y.

Y value	0	-1	-2	-3	-4	-5	1	2	3	4	5
Probabilities of each one	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
Simplifying fractions	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$

Table 3: Calculation of probabilities for the expected value of XY.

XY value	0	-3	-8	-15	-24	-35	3	-5	-12	-21	-32	8
Probabilities of each one	$\frac{6}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
XY value	5	-7	-16	-27	15	12	7	-9	-20	24	21	16
Probabilities of each one	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
XY value	9	-11	35	32	27	20	11					
Probabilities of each one	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$					

In this case, we replicated the experiment made in Exercise 1.1. In Figure 2a, 2b, and 2c we can see a pie plot with only one experimentation of each expected value experiment. If we compare the results we have from Table 1, 2 and 3 we can see that the distribution in the pie plot resembles our analitical results.

Keeping with the example of Exercise 1.1 we then made an experimentation of 100 iterations of 10,000 repetitions of the same experiment, to see if the result behaves like we expected. In Figure 3 we have the histograms of each expected value.

In Figure 3a we have the experiment for $E(X)$ and according to our analysis, the result should be 7. The histogram clearly shows a majority of the distribution in a range of 6.90 to 7.05, so our calculation can be deemed correct. In Figure 3b we have the experiment of the $E(Y)$. Here the result is 0 in our analysis, and as we can see, the interval of the distribution for the results is from -0.06 to 0.06, so we can again say our analysis is correct. Lastly for Figure 3c we have the experimentation for the $E(XY)$. Here the result should also be 0 according to our analysis. In this case, the interval is a bit more wide, but the majority of the distribution on this histogram still focuses on -0.2 to 0.2

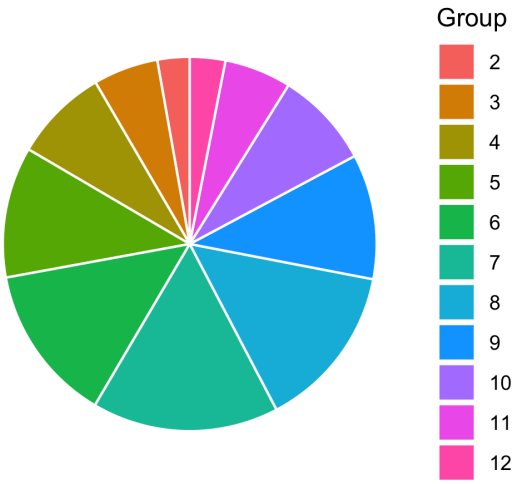
■

1.3 Exercise 15, page 249

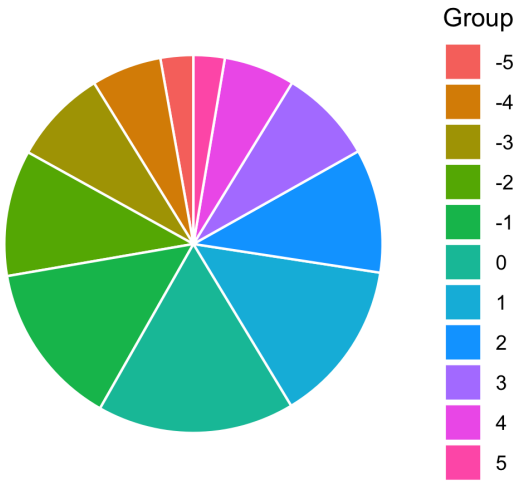
A box contains two gold balls and three silver balls. You are allowed to choose successively from the box at random. You win 1 dollar each time you draw a gold ball and lose 1 dollar each time you draw a silver ball. After a draw, the ball is not replaced. Show that, if you draw until you are ahead by 1 dollar, or until there are no more gold balls, this is a favorable game.

- **Experimentation:**

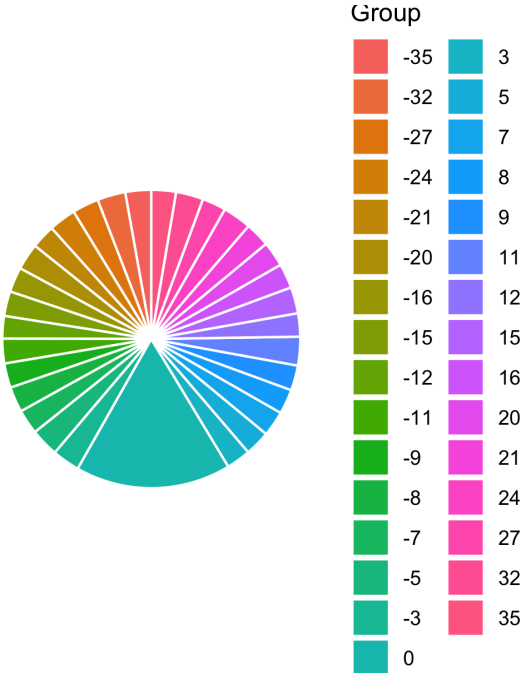
For this experimentation, first we replicated the rules that the exercise dictates. In `R` we used the library `arrangements` to make the different scenarios of drawing the balls. Table 4 gives us the result of this library. We represented the golden and silver balls according to their colors, and the value they give us when we draw them. The las column called winnings represents the final result, following the rules of ending the game if we are ahead by 1 dollar, or if we draw both silver balls.



(a) Pie plot of the one iteration of the 10,000 repetitions of the dice experiment for $E(X)$.

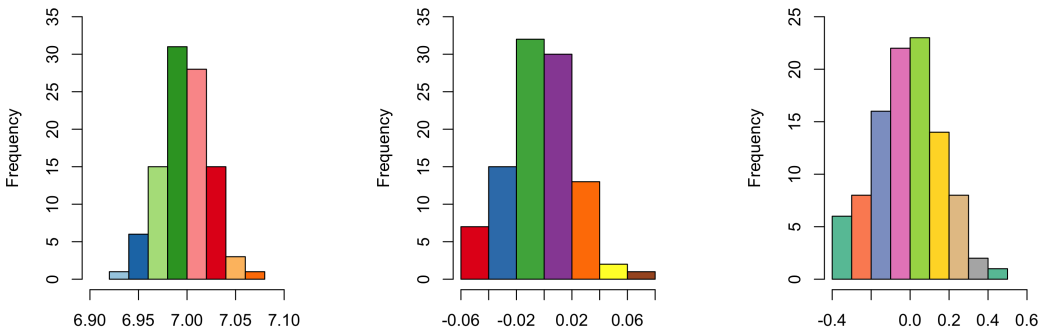


(b) Pie plot of the one iteration of the 10,000 repetitions of the dice experiment for $E(Y)$.



(c) Pie plot of the 100 iterations of the 10,000 repetitions of the dice experiment for $E(XY)$.

Figure 2: Pie plot of the different expected values of the experiment.



(a) Histogram of the dice experiment for $E(X)$.

(b) Histogram of the dice experiment for $E(Y)$.

(c) Histogram of the dice experiment for $E(XY)$.

Figure 3: Histograms of the 100 iterations of the 10,000 repetitions from the expected value of all the experiments.

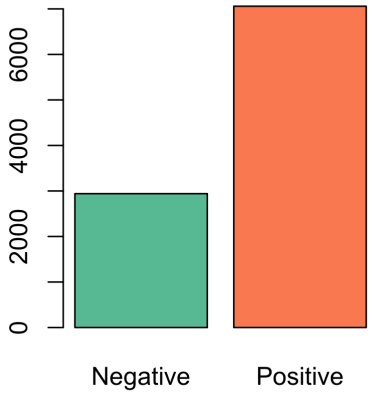
Table 4: Different resulting winnings on the different scenarios

	(,1)	(,2)	(,3)	(,4)	(,5)	Winnings
(1,)	-1	-1	-1	1	1	-1
(2,)	-1	-1	1	-1	1	-1
(3,)	-1	-1	1	1	-1	0
(4,)	-1	1	-1	-1	1	-1
(5,)	-1	1	-1	1	-1	0
(6,)	-1	1	1	-1	-1	1
(7,)	1	-1	-1	-1	1	1
(8,)	1	-1	-1	1	-1	1
(9,)	1	-1	1	-1	-1	1
(10,)	1	1	-1	-1	-1	1

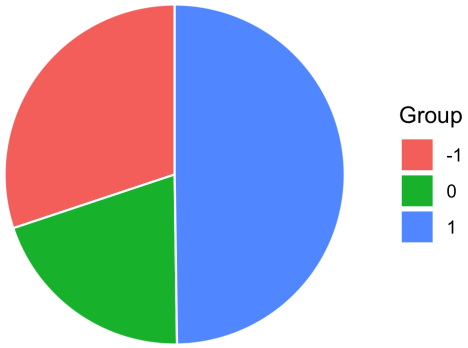
In Figure 4 we have the graphical representation of the the experiment, in which we draw a different configuration 10,000 times. The problem asks us to prove that if we draw until we are ahead by one dollar or there are no more gold balls, that the resulting winnings are still a favorable game. In our analysis we concluded that, if we count as a favorable result when we end up without losses nor winnings, $\frac{7}{10}$ of the times, we end up with favorable results.

In the experiment of 10,000 we have that 0.7059 of the times, we have a favorable result, proving our analysis correct.

■



(a) Bar plot of the positive and negative outcomes of the experiment.



(b) Pie plot of the distribution of the different results of the experiment (-1, 0 and 1).

Figure 4: Graphic representation of the 10,000 repetition of the experiment.

1.4 Exercise 18, page 249 (Extra)

Exactly one of six similar keys opens a certain door. If you try the keys, one after another, what is the expected number of keys that you will have to try before success?

- **Experimentation:**

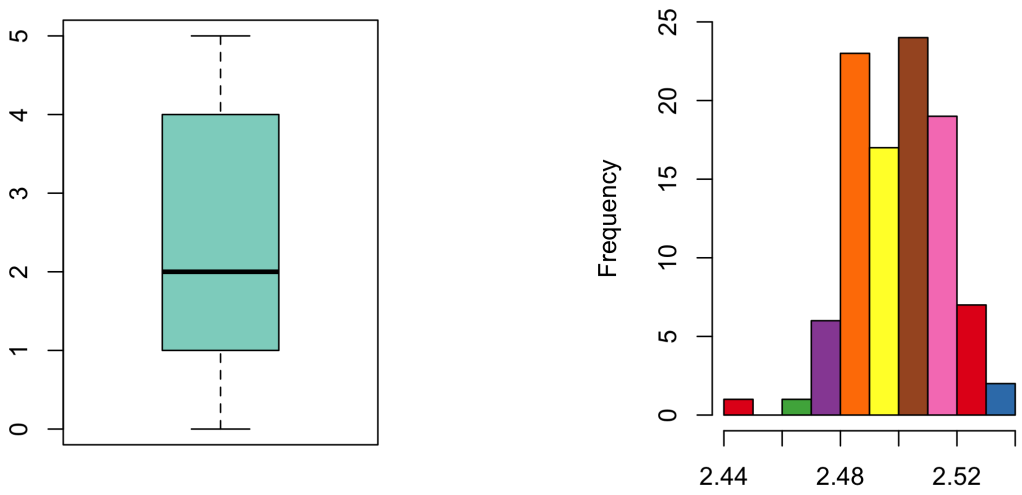
For this exercise, we used again the library of `sample`. We selected at random a number between 1 and 6, for a 10,000 repetitions. This number was the correct key. We then checked how many tries it took to find the right key. All these tries where saved in another variable. Using the `mean` option in `R` we calculated a mean of 2.4849.

In Figure 5a we have the box plot of one iteration of 10,000 repetitions. The mean in this plot goes to 2, because we used integer numbers, and in the experiment, we can not draw half a key. In Figure 5b we can see the histogram where we made 100 iterations of the 10,000 repetitions. The interval of medians is between 2.44 to 2.54, being the majority of the frequency in the 2.5 mark.

■

1.5 Exercise 27, page 252 (Extra)

It has been said that a Dr. B. Muriel declined a cup of tea stating that she preferred a cup into which milk had been poured first. The famous statician R. A. Fisher carried out an experiment to see if she could tell whether milk was poured in before or after the tea. Assume that for the



(a) Box plot of one iteration of 10,000 repetitions.

(b) Histogram of the 100 iterations of the 10,000 repetitions.

Figure 5: Graphic representation of the keys experiment.

test Dr. Bristol was given 8 cups of tea, four in which the milk had been poured before the tea, and four in which the milk was put in after the tea.

a) What is the expected number of correct guesses the lady would make if she had no information after each test and was just guessing.

- **Experimentation:**

This is just half of the experimentation, because since Practice 9 I started with some of the analysis of this problem, and I found it kind of funny that these staticians had such an important discovery in probability from an argument about how much do they know of tea. In this case, after some investigation, we found out that Fisher, the one who posed the argument, was so adamant to show that he was right, that he calculated how many cups of tea could Muriel guess correctly if she was in fact just guessing. The proposed hypothesis then was that if she managed to guess all 8, then she indeed must know something about the taste of the tea composition.

For this experimentation, we used the `dhyper` function in `R` to fill all the information we had about the problem. In this case, we discovered that the chance of Dr. B. Muriel to guess all 8 of them correctly if she was just guessing, was of **0.013**. In Figure 6 we can see the declining probability if the experiment where to continue. After the 14 cup, the probability becomes 0.

As a conclusion to this problem, I say that if she managed to guess correctly all of the cups, then she must really know something about the tea.

■

2 Conclusion

In this practice, it was easier to develop the experimentation, because we already had the outcome in mind. As a side note, the last extra experiment was a fun conversation in dinner with my family, because now my mother claims she can tell the same difference but with coffee.

References

[1] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

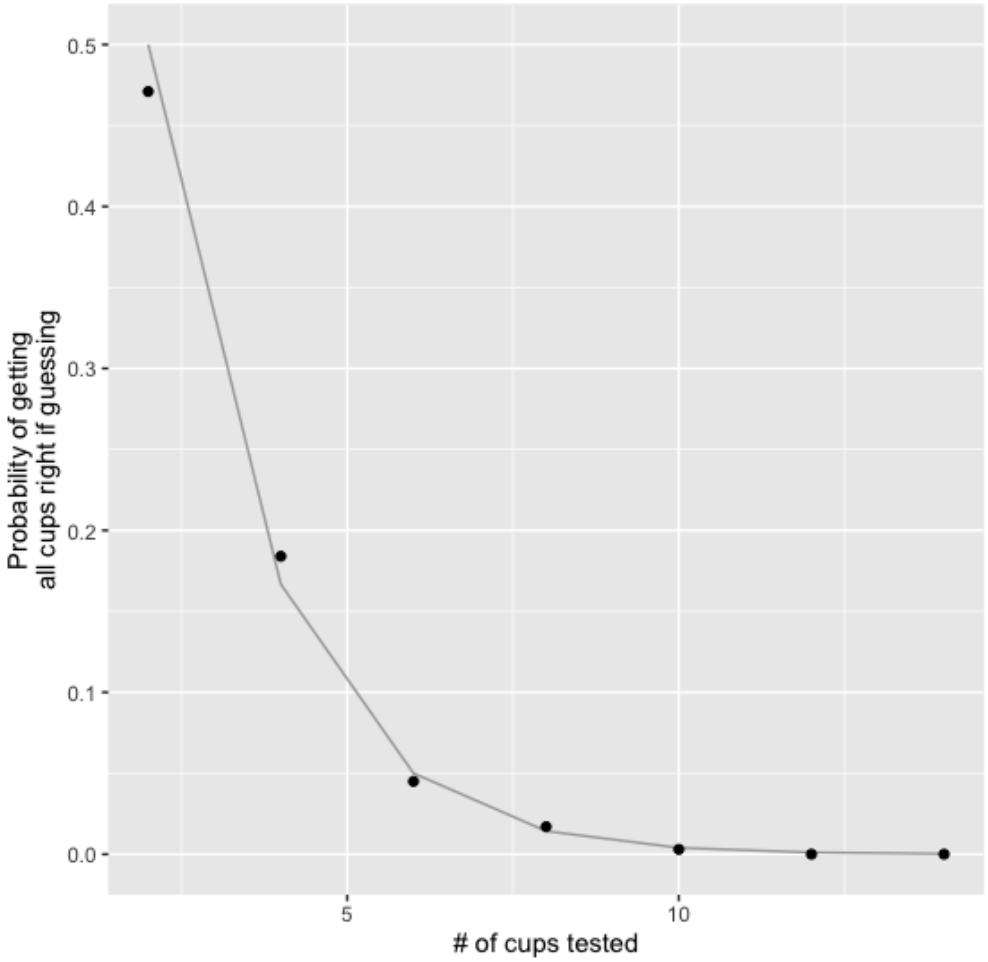


Figure 6: Graphic representation of the diminishing probability of guessing correctly different number of cups

Practice 11: Convolutions, Chi square and covariance.

Mayra Cristina Berrones Reyes 6291

November 17, 2020

1 Introduction

In this work we are going to discuss in three sections the subjects of convolution, chi squared and covariance with different examples. But first, we want to give a brief introduction into these three items.

1.1 Convolutions

Convolution refers to a mathematical operation on two functions (f and g) that produce a third function ($f * g$) that expresses how the shape of one is modified by the other. The term is often used both as the result function we mentioned, as well as the process of computing the function itself.

Convolutions can be find in several applications such as probability, statistics, computer vision, natural language processing, image and signal processing, engineering, etc.

The common engineering notational convention for this formula is shown in Equation 1.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{1}$$

■

1.2 Chi square test

The Pearson chi squared test is a statistical test applied to show the relationship between two categorical variables. This statistic is a single number that tells how much difference exists between the observed counts and the the expected counts if there was no relationship at all in the population.

There are two types of chi squared tests:

- A chi squared goodness fit test, that determines if a sample data matches a population.
- A chi squared test to test for independence, comparing two variables in a contingency table to see if they are related, testing whether distributions differ from each other.

The formula for the chi squared statistic test used in this type of tests is shown in Equation 2.

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2}$$

■

Table 1: Some popularly known kernels for image processing.

Boxblur	Gaussian blur	Identity	Sharpen
$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$	$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix}$

1.3 Covariance

In mathematics and statistics, covariance is a measure of the relationship that two random variables have with each other. It evaluates how much the variables change together, but they do not measure the dependency between variables. The variance can take any positive or negative value. This values can be interpreted as:

- **Positive:** Indicates that two variables tend to move in the same direction.
- **Negative:** Reveals that two variables tend to move in an inverse direction.

The formula for the covariance between two random variables X and Y can be calculated using Equation 3,

$$Cov(X,Y) = \frac{\sum(X_i - \overline{X})(Y_j - \overline{Y})}{n} \tag{3}$$

■

2 Section 1: Convolutions

Identify an application of your interest from the subject of convolution and present some related contribution (theoretical or numerical)

Convolutions, as we see in the introduction of this work, are an extremely general idea. The expression used in Equation 1 can also be used in a higher number of dimensions. Just like with one dimensional convolutions, the concept of two dimensional convolutions can be imagined as sliding one function on top of another, multiplying and adding [7].

Our subject of interest, and one of the common application of convolutions is image processing. In this case, we think of the images as two dimensional functions, in the form of value of pixels. Many important image transformations such as noise additions, sharpening of the image, etc. are convolutions where we convolve the image function with a very small local function called "kernel". The kernel slides to every position of the image and computes a new pixel as a weighted sum of pixels as it floats over.

In Table 1 we see some examples of popularly kernels used for different types of image precessing. Depending on the scale it wants to work, we can change the size of its matrix and some of the numbers inside the matrix.

The easiest way of representing how exactly a kernel works, can be appreciated in Figure 1 where we can see the kernel sweeping over the image matrix. We multiply a batch of the image

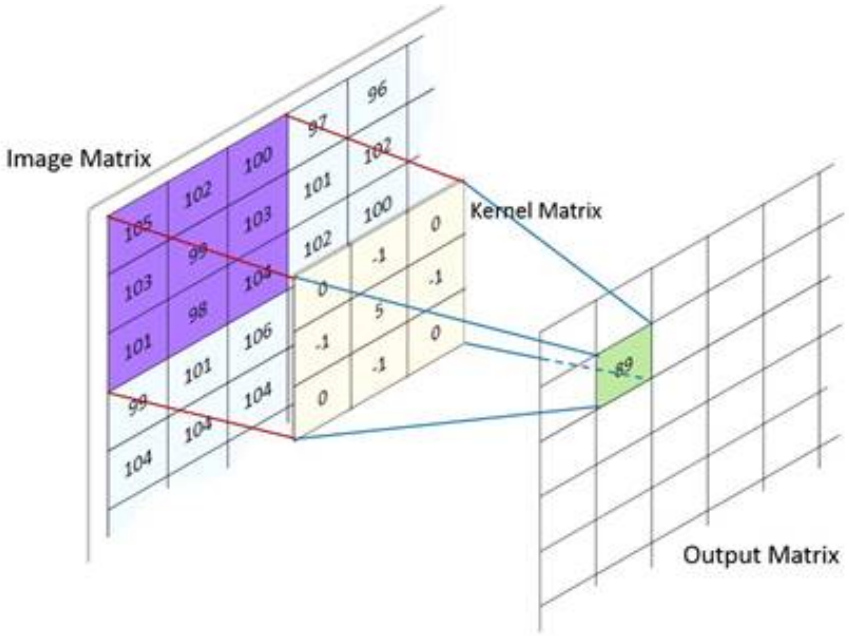


Figure 1: Graphical example of how a kernel works in a convolution of an image [6].

with the kernel, and then calculate the cumulative sum to add it to another matrix.

As an example of the use of convolutions in image processing, we can use the library of `magick` [4] in `R` that already has some integrated values for the kernels to add noise, edge detection, sharpen image, etc. In Figure 2 we see some of this examples. In this figure we also add a hand made kernel to blur the image, based on the values we used in Table 1.

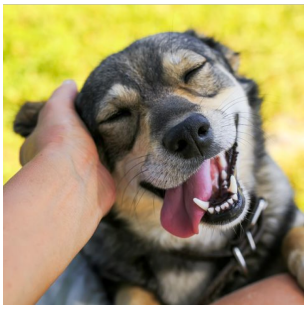
3 Section 2: Chi square

Identify an application of your interest (preferably of your thesis topic, if you have one) for the chi square test and apply it.

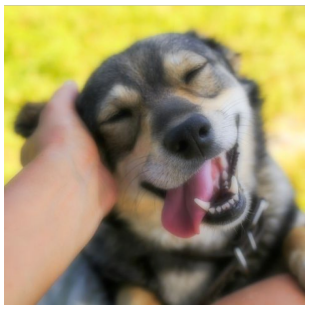
For this application, following the subject of convolution, one of the main topics set for our thesis is the use of Convolutional Neural Networks (CNNs). For this type of Neural Network (NN) we use the same concept that we explained in the convolution section. We take the formula shown in Equation 4 and use it to calculate the index and rows of our resulting matrix.

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \tag{4}$$

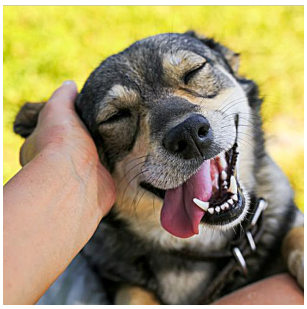
In the case of Convolutional Neural Networks (CNNs) we form a subsequent map of features in each convolution. The kernel used starts with random variables that get fitted in every iteration,



(a) Original image



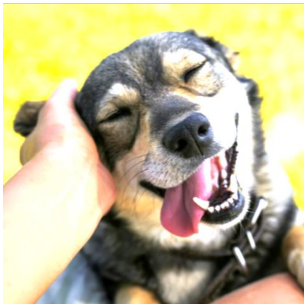
(b) Convolution with the gaussian kernel



(c) Convolution with the sharpen kernel

(d) Convolution with the sobel kernel (positive values)

(e) Convolution with the sobel kernel (negative values)



(f) Convolution with the blur kernel (hand picked)

Figure 2: Use of convolution to change the an image of a dog.

Figure 3: Some images form the data set Mnist Fashion.

depending on the optimizer that we are using.

As we can see in Figure 1 if we, for example, started with a matrix of 6×6 with a kernel of 3×3, we will get a feature map of 4×4. Since our image shrinks every time we perform a different convolution, we can only perform them a limited number of times before our image disappears completely. That is why there are many architectures of CNN and we experiment to find the best fit for our data.

For the experimentation part of this section, first we have to produce data that can be evaluated by the chi squared test. Since the images and and algorithm we are currently working on is a little bit too heavy for the computer we are currently working on, we are using a famous (and digestible for regular computers) data set of images called MNIST Fashion [5], which is widely used as a benchmark in the data science community to validate their algorithms [3].

We run a simple CNN with the help of the library of `keras`. In Figure 2 we show some of the images contained in this data set, and in Table 2 we have the results of the model testing, and in Table 3 we show the confusion matrix for these classification.

To be able to use the confusion matrix represented in Table 3 we accommodate the data in a more compact table, as the one shown in Table 4, where we put the right and wrong predictions of each class in two different rows.

Table 2: Metric results of the CNN model for the classes form the MNIST Fashion data set.

Class	0	1	2	3	4	5	6	7	8	9
Sensitivity	88.5%	98.3%	82.4%	90.8%	75.5%	98.3%	67.5%	93.9%	94.4%	93.7%
Specificity	97.4%	99.7%	97.5%	98.9%	98.4%	99.2%	97.0%	99.3%	99.9%	99.7%
Pos Pred Value	79.0%	97.0%	78.6%	90.4%	84.1%	93.7%	71.3%	93.8%	98.6%	97.4%
Neg Pred Value	98.7%	99.8%	98.0%	99.0%	97.3%	99.8%	96.4%	99.3%	99.4%	99.3%
Prevalence	9.9%	10.1%	10.0%	9.9%	10.1%	10.1%	10.0%	9.9%	9.8%	10.2%
Detection Rate	8.8%	9.9%	8.3%	9.0%	7.6%	10.0%	6.7%	9.3%	9.3%	9.5%
Detection Prevalence	11.1%	10.2%	10.5%	10.0%	9.0%	10.6%	9.5%	9.9%	9.4%	9.8%
Balanced Accuracy	92.9%	99.0%	89.9%	94.9%	86.9%	98.8%	82.3%	96.6%	97.1%	96.7%

Table 3: Confusion matrix of prediction results for the CNN model. The principal diagonal represent the correct predictions, and all the other are the wrong predictions.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	1052	1	26	29	1	0	209	0	13	1
1	5	1188	3	18	5	0	5	0	1	0
2	15	0	990	2	156	0	90	0	7	0
3	18	17	11	1083	37	0	23	0	9	0
4	3	1	67	33	910	0	54	0	14	0
5	1	0	0	0	0	1195	0	50	11	19
6	91	1	102	28	97	0	809	0	6	0
7	0	0	0	0	0	13	0	1113	4	57
8	4	0	3	0	0	0	8	1	1115	0
9	0	0	0	0	0	8	0	21	1	1141

Table 4: Table simplifying the data of the confusion matrix made in `R`.

Class	(, 0)	(, 1)	(, 2)	(, 3)	(, 4)	(, 5)	(, 6)	(, 7)	(, 8)	(, 9)
True predictions	1052	1188	990	1083	910	1195	809	1113	1115	1141
False predictions	277	37	270	115	172	81	325	72	16	30

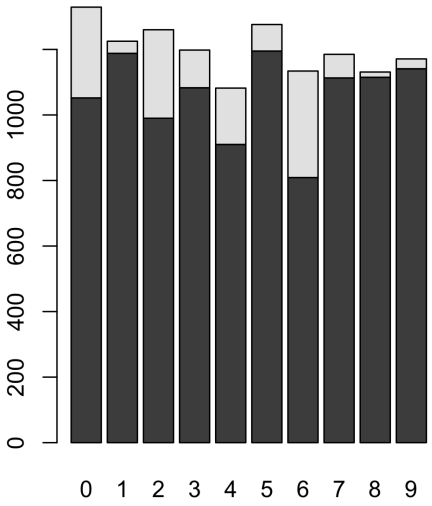


Figure 4: Barplot of the correct and incorrect predictions of the CNN model on each class.

Using the `chisq.test` [1] function in `R` we get as a result:

- **X squared:** 938.25
- **df:** 9
- **p value:** $< 2.2 \times 10^{-16}$

So as a conclusion we can say that, in our example, the row and the column variables are statistically significantly associated. And finally, Figure 4 shows a bar plot of how the different classes are distributed, being the darker color the correct predictions, and the grey the wrong predictions.

4 Section 3: Covariance

Validate the two properties related to the covariance that comes in the class page (preferably, first establish numerically that they are true and then prove them analytically

Both Equation 5 and 6 come form the example in the class page indicated in the instructions. For Equation 6 first we attempt a practical experiment in `R` using the functions of variance and covariance already in the program, as well as the function of `sample` to generate the random variables.

$$Cov[aX + b, cY + d] = acCov[X, Y] \tag{5}$$

$$Var[X + Y] = Var[X] - Var[Y] + 2Cov[X, y] \tag{6}$$

In Figure 5 we have the different number of iterations. We went from 10, to 100 to 1000 iterations, in each one we tried the equation with a different number of random variables inside the variance and covariance functions, which can be seen in the `x` axis of all the plots.

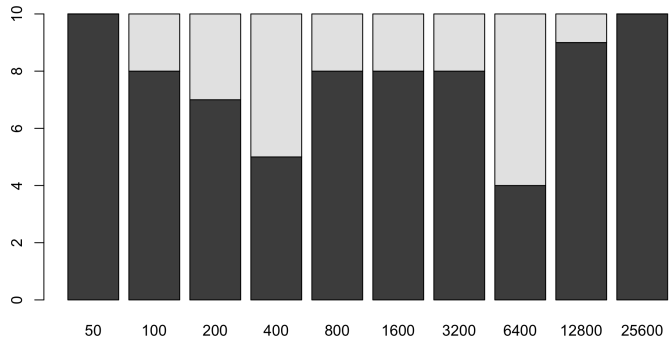
The darker color is for all the times the condition of equal was accomplished, and the gray color is when this condition was not met.

The analytic view of the formula can be described by Equation 7 [2],

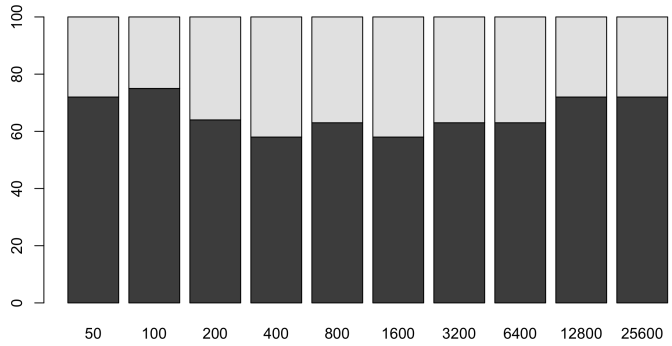
$$\begin{aligned}
&Var[X + Y] \\
&\quad = \sum_x \sum_y (x + y)^2 P_{XY}(x, y) - (E(X + Y))^2 \\
&\quad = \sum_x \sum_y x^2 P_{XY}(x, y) + \sum_x \sum_y 2xy P_{XY}(x, y) + \\
&\quad \sum_x \sum_y y^2 P_{XY}(x, y) - (E(X))^2 - 2E(X)E(Y) - (E(Y))^2 \\
&\quad = \sum_x x^2 P_X(x) - (E(X))2 + \sum_y y^2 P_Y(y) - (E(Y))^2 + \\
&\quad \sum_x \sum_y 2xy P_{XY}(x, y) - 2E(X)E(Y) \\
&\quad = E(X^2) - (E(X))^2 + E(Y^2) - (E(Y))^2 + 2(E(XY) - E(X)E(Y)) \\
&\quad = Var(X) + Var(Y) + 2Cov(X, Y).
\end{aligned} \tag{7}$$

5 Conclusion

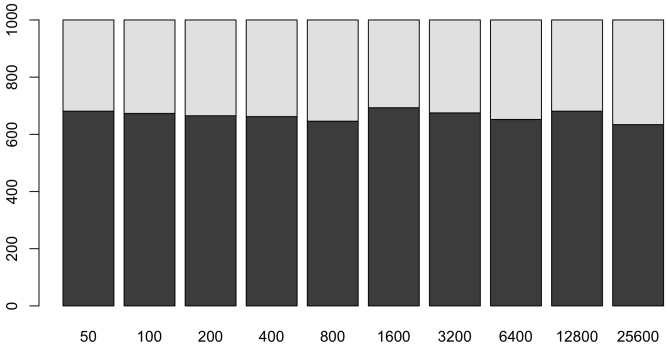
As always, I understand better all of this concepts when they come in the form of a practical example. In the reading part leading up to the class I was really confused by the concepts, and the explanations in class where not enough to dispel my doubts. It wasn't until some videos and examples later that I landed on the application of image processing. I thought this convolution and the one seen in class where two different concepts (one for statistics and the other for computer



(a) Barplot from 10 iterations of the equation comparision



(b) Barplot from 100 iterations of the equation comparision



(c) Barplot from 1000 iterations of the equation comparision

Figure 5: Barplots from the different iterations of the experiment of covariance.

science) but after seeing the application on the example, it became so easy to understand the equations seen class and its uses.

In the section of the Chi squared we used the MNIST dataset because the computer used for the experiments is of average processing capacity, and the images used in the real thesis problem are too big for it to be able to compile the program.

References

[1] Chi square test of independence in r. https://www.statsandr.com/blog/chi-square-test-of-independence-in-r/. Accessed: 2020-11-17.

[2] Sums of random variables. http://www.milefoot.com/math/stat/rv-sums.htm. Accessed: 2020-11-17.

[3] Lbb neural network. https://www.kaggle.com/stvenl/lbb-neural-network. Accessed: 2020-11-17.

[4] Image convolution in r using magick. https://ropensci.org/technotes/2017/11/02/image-convolve/. Accessed: 2020-11-17.

[5] Fashion mnist. https://www.kaggle.com/zalando-research/fashionmnist. Accessed: 2020-11-17.

[6] Understanding convolutions. https://colah.github.io/posts/2014-07-Understanding-Convolutions/. Accessed: 2020-11-17.

[7] Gentle dive into math behind convolutional neural networks. https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9. Accessed: 2020-11-17.

Practice 12: Exercises of generating functions

Mayra Cristina Berrones Reyes 6291

November 24, 2020

1 Exercises

The exercises of this work where taken from the complementary reading in section [1] of the class materials.

1.1 Exercise 1, page 393

Let $Z_1, Z_2, ..., Z_N$ describe a branching process in which each parent has j offspring with probability p_j. Find the probability d that the process eventually dies out if:

(a) $p_0 = \frac{1}{2}, p_1 = \frac{1}{4}$ and $p_2 = \frac{1}{4}$.

(b) $p_0 = \frac{1}{3}, p_1 = \frac{1}{3}$ and $p_2 = \frac{1}{3}$.

(c) $p_0 = \frac{1}{3}, p_1 = 0$ and $p_2 = \frac{2}{3}$.

(d) $p_j = \frac{1}{2}^{j+1}$ for $j = 0, 1, 2, ...$

(e) $p_j = \left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^j$ for $j = 0, 1, 2, ...$

(f) $p_j = e^{-2}\frac{2^j}{j!}$ for $j = 0, 1, 2, ...$ (estimate j numerically).

- **Answer:**

(a) $p_0 = \frac{1}{2}, p_1 = \frac{1}{4}$ and $p_2 = \frac{1}{4}$.

Taking into consideration the example form the complementary material, we have that if $p_0 > p_2$ then $m < 1$, then we have a second root that is > 1 For this item, we have $p_0 = \frac{1}{2}$ and $p_2 = \frac{1}{4}$, so we can say that in this case that $m < 1$ and the second root is > 1.

(b) $p_0 = \frac{1}{3}, p_1 = \frac{1}{3}$ and $p_2 = \frac{1}{3}$.

In this item we have that $p_0 = p_2$. For this case we say that we have a double root, where $d = 1$.

(c) $p_0 = \frac{1}{3}, p_1 = 0$ and $p_2 = \frac{2}{3}$.

For this case, we have take the same concept that we used in item (a), whew we compare p_0 and p_2. In this item, $p_0 < p_2$, which means that $m > 1$, so the second root here represents that d is **less than 1** as the probability of that the process will die out.

(d) $p_j = \frac{1}{2}^{j+1}$ for $j = 0, 1, 2, ...$

For the following items, we have a different format, in which j represents the number of probabilities we have. In these cases we also relied on `R` as a calculator to see to which number it converges. In this case, it goes to 1. We also did the same thing as item (a), comparing p_0 and p_2, and we got that $p_0 > p_2$, so our $m < 1$ and the second root is > 1.

(e) $p_j = \left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^j$ for $j = 0, 1, 2, ...$

This one was kind of confusing, because when we did the experimentation in `R` the number it converges is 2. But if we make the same experiment that we have been doing for all of the items before, we get that $p_0 > p_2$, so our $m < 1$ and the second root is > 1.

(f) $p_j = e^{-2}\frac{2^j}{j!}$ for $j = 0, 1, 2, ...$ (estimate j numerically).

We mentioned that item (e) was confusing because this experiment also converges to 2, but in this case the value is $p_0 < p_2$, which means that $m > 1$, so the second root here represents that d is **less than 1**.

■

1.2 Exercise 3, page 393

In the chain letter problem (see Example 10.14) find your expected profit if:

(a) $p_0 = \frac{1}{2}, p_1 = 0$ and $p_2 = \frac{1}{2}$.

(b) $p_0 = \frac{1}{6}, p_1 = \frac{1}{2}$ and $p_2 = \frac{1}{3}$.

- **Answer:**

(a) $p_0 = \frac{1}{2}, p_1 = 0$ and $p_2 = \frac{1}{2}$.

Reading the example 10.14, we follow the same steps and rules they tell us. For instance, we know that $m = p_1 + 2p_2$ is the expected number of letters that we sold. Then we follow the calculation of $50m + 50m^{12} > 100$ or if $m * m^{12} > 2$ to know if our profit is favorable. So, for the experiment in item (a) we have in Equation 1:

$$m = p_1 + 2p_2 = 0 + 2\left(\frac{1}{2}\right) = 1 \tag{1}$$

With the results in Equation 1 we have that $m * m^{12} = 2$. Since our formula is a strict $>$, then we expect not to have profit in this one. Doing the calculation we have $50(2) - 100 = 0$, proving our first assumption.

(b) $p_0 = \frac{1}{6}, p_1 = \frac{1}{2}$ and $p_2 = \frac{1}{3}$.

For this one we repeat the procedure we made in item (a). So, we have the result of m in Equation 2:

$$m = p_1 + 2p_2 = \frac{1}{2} + 2\left(\frac{1}{3}\right) = \frac{7}{6} \tag{2}$$

With the results in Equation 2 we have that $m * m^{12} = 7.5252$. With this result, we have that $m * m^{12} > 2$, so we do expect to be having some profit. Doing the calculation we have $50(7.5252) - 100 \approx 276$, proving our assumption.

- Show that if $p_0 > \frac{1}{2}$, you cannot expect to make a profit.

For this last one, we can deduce from the item (a) where we had no profit, that if p_0 goes any higher than $\frac{1}{2}$, then the value of m will be smaller, making the final calculation have negative numbers. To prove this, we played with the value of p_0, making it as closer to $\frac{1}{2}$ but always slightly higher. The value of m does not get to be 1 in any of them. This is done, of course, with the assumption that the sum off all p is equal to 1.

■

1.3 Exercise 1, page 402

Let X be a continuous random variable with values in $[0, 2]$ and density f_X. Find the moment generating function $g(t)$ for X if:

(a) $f_X(x) = \frac{1}{2}$.

(b) $f_X(x) = \left(\frac{1}{2}\right) x$.

(c) $f_X(x) = 1 - \left(\frac{1}{2}\right) x$.

(d) $f_X(x) = |1 - x|$.

(e) $f_X(x) = \left(\frac{3}{8}\right) x^2$.

Hint: Use the integral definition, as in Examples 10.15 and 10.16.

- **Answer:**

(a) $f_X(x) = \frac{1}{2}$.

$$\begin{aligned} g(t) = &\int_0^2 \frac{1}{2} e^{tx} dx = \frac{1}{2} \int_0^2 e^{tx} dx \\ =&\frac{1}{2} \left[\frac{e^{tx}}{t} \right]_0^2 = \frac{1}{2} \left(\frac{e^{2t} - e^{0t}}{t} \right) \\ =&\frac{e^{2t} - 1}{2t}. \end{aligned} \tag{3}$$

(b) $f_X(x) = \left(\frac{1}{2}\right) x$.

$$
\begin{aligned}
g(t) &= \int_0^2 \frac{1}{2} x e^{tx} dx = \frac{1}{2} \int_0^2 x e^{tx} dx \\
u &= x \qquad v = \frac{e^{tx}}{t} \\
du &= 1dx \qquad dv = e^{tx} dx \\
&= \int_0^2 x e^{tx} dx = \frac{x e^{tx}}{t}\Big|_0^2 - \int_0^2 \frac{e^{tx}}{t} dx \\
&= \int_o^2 x e^{tx} dx = \frac{x e^{tx}}{t}\Big|_0^2 - \frac{e^{tx}}{t^2}\Big|_0^2 \\
&= \frac{2e^{2t}}{t} - \frac{e^{2t}-1}{t^2} \\
\therefore g(t) &= \frac{1}{2}\left(\frac{2e^{2t}}{t} - \frac{e^{2t}-1}{t^2}\right).
\end{aligned} \tag{4}
$$

(c) $f_X(x) = 1 - \left(\frac{1}{2}\right) x$.

$$
\begin{aligned}
g(t) &= \int_0^2 \left(1 - \frac{1}{2}x\right) e^{tx} dx \\
&= \int_0^2 e^{tx} dx - \frac{1}{2} \int_0^2 x e^{tx} dx \\
&= \int_0^2 e^{tx} dx = \frac{e^{tx}}{t}\Big|_0^2 = \frac{e^{2t}-1}{t} \\
g(t) &= \frac{e^{2t}-1}{t} - \frac{1}{2}\left(\frac{2e^{2t}}{t} - \frac{e^{2t}-1}{t^2}\right) \\
&= \frac{e^{2t}-1}{2t^2} - \frac{1}{t}.
\end{aligned} \tag{5}
$$

(d) $f_X(x) = |1 - x|$.

(e) $f_X(x) = \left(\frac{3}{8}\right) x^2$.

$$
\begin{aligned}
g(t) &= \int_0^2 \frac{3}{8} x^2 e^{tx} dx = \frac{3}{8} \int_0^2 x^2 e^{tx} dx \\
u &= x^2 \qquad v = \frac{e^{tx}}{t} \\
du &= 2x dx \qquad dv = e^{tx} dx \\
&= \int_0^2 x^2 e^{tx} dx = \frac{x^2 e^{tx}}{t}\Big|_0^2 - \int_0^2 \frac{2x e^{tx}}{t} dx \\
&= \frac{x^2 e^{tx}}{t}\Big|_0^2 - \frac{2}{t} \int_0^2 x e^{tx} dx \\
&= \frac{4e^{2t}}{t} - \frac{2}{t}\left(\frac{2e^{2t}}{t} - \frac{e^{2t}-1}{t^2}\right).
\end{aligned} \tag{6}
$$

■

1.4 Exercise 6, page 403

Let X be a continuous random variable whose characteristic function $k_X(\tau)$ is:

$$k_x(\tau) = e^{-|\tau|},\ -\infty < \tau < +\infty.$$

Show directly that density f_X of X is:

$$f_X(x) = \frac{1}{\pi(1+x^2)}.$$

- **Answer:**

For this we have the start of this equation from the recommended reading, and we begin by replacing the $k_x(\tau) = e^{-|\tau|}$ value . In Equation 7 we can see how it develops to the formula of the Cauchy density.

$$\begin{aligned}
f_X(x) =& \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ix\tau}e^{-|\tau|}d\tau \\
=& \frac{1}{2\pi}\left(\int_{-\infty}^{0} e^{-ix\tau}e^{-}-\tau d\tau + \int_{\infty}^{0} e^{-ix\tau}e^{-\tau}d\tau\right) = \frac{1}{2\pi}\left(\int_{-\infty}^{0} e^{-ix\tau+\tau}d\tau + \int_{\infty}^{0} e^{-ix\tau-\tau}d\tau\right) \\
=& \frac{1}{2\pi}\left(\int_{-\infty}^{0} e^{\tau(1-i\tau)}d\tau + \int_{\infty}^{0} e^{-\tau(1+ix)}d\tau\right) = \frac{1}{2\pi}\left(\left(\frac{e^{\tau(1-ix)}}{(1-ix)}\right)\Big|_{-\infty}^{0} + \left(-\frac{e^{-t(1+ix)}}{(1+ix)}\right)\Big|_{0}^{-\infty}\right) \\
=& \frac{1}{2\pi}\left(\left(\frac{e^{0(1-ix)}}{(1-ix)} - \frac{e^{-\infty(1-ix)}}{(1-ix)}\right) - \left(\frac{e^{-\infty(1+ix)}}{(1+ix)}\frac{e^{0(1+ix)}}{(1+ix)}\right)\right) = \frac{1}{2\pi}\left(\frac{1}{(1-ix)} + \frac{1}{(1+ix)}\right) \\
=& \frac{1}{2\pi}\left(\frac{(1+ix)+(1-ix)}{(ix+1-(ix)^2)-ix}\right) = \frac{1}{2\pi}\left(\frac{2}{(1-(ix^2))}\right) = \frac{1}{\pi}\left(\frac{1}{(1-(-x^2))}\right) \\
=& \frac{1}{\pi(1+x^2)}
\end{aligned} \tag{7}$$

■

1.5 Exercise 10, page 404

Let $X_1, X_2, ..., X_n$ be an independent trials process with density:

$$f(x) = \frac{1}{2}e^{-|x|},\ -\infty < x < +\infty.$$

(a) Find mean and variance of $f(x)$.

(b) Find the moment generating function fo X_1, S_n, A_n, and S_n^*.

(c) What can you say about the moment generating function of S_n^* as $n \to \infty$.

(d) What can you say about the moment generating function of A_n as $n \to \infty$.

- **Answer:**

(a) Find mean and variance of $f(x)$.

For this item (a) we can say that the mean is zero by symmetry. A symmetric probability distribution is a probability distribution which is unchanged when its probability density function is reflected around a vertical line at some value of the random variable represented by the distribution. This probability of being any given distance on one side of the value about which symmetry occurs is the same as the probability of being the same distance on the other side of that value [2]. Since we have $-\infty < x < +\infty$ we can assume this is the case.

The variance can be obtained by integrating X^2 multiplied by the density, which can be performed as an integral performed using integration by parts (to accommodate the different limits and the absolute value of the exponent x).

$$\begin{aligned} &\int_{-\infty}^{0} x^2 \frac{e^x}{2} dx + \int_{0}^{\infty} x^2 \frac{e^{-x}}{2} dx \\ &\int_{-\infty}^{0} x^2 \frac{e^x}{2} dx = 1 \\ =&1 + \int_{0}^{\infty} x^2 \frac{e^{-x}}{2} dx = 1 + 1 = 2. \end{aligned} \tag{8}$$

(b) Find the moment generating function fo X_1, S_n, A_n, and S_n^*.

(c) What can you say about the moment generating function of S_n^* as $n \to \infty$.

For this, the answer kind of comes from the recommended reading. In Equation 9 we have the reference, to the standardized sum S_n^* generated function.

$$g_n^*(t) = \left(g\left(\frac{t}{\sqrt{n}} \right) \right)^2 \tag{9}$$

From that, using the LH́opital rule twice, we have that $g_n^*(t) \to e^{\frac{t^2}{2}}$ as $n \to \infty$. This rule tells us that the S_n^* generated function must converge to that distribution function of $e^{\frac{t^2}{2}}$.

(d) What can you say about the moment generating function of A_n as $n \to \infty$.

In this one, seeing the Equation 10 taken from the book notes, we think it converges to 1. They are taking an average for all the numbers, and, following the process of Exercise 1 in this work, the numbers can keep getting smaller and smaller, but as n tends to infinity, then my guess is that in some point it is going to converge to 1.

$$A_n = \frac{X_1 + X_2 + ... + X_n}{n} \tag{10}$$

■

References

[1] Chapter 10 - generating functions. https://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter10.pdf. Accessed: 2020-11-17.

[2] Symmetric probability distribution. https://en.wikipedia.org/wiki/Symmetric_probability_distribution. Accessed: 2020-11-17.

Practice 13: Law of large numbers.

Mayra Cristina Berrones Reyes 6291

December 1, 2020

1 Introduction

The Law of Large Numbers (LLN) is a theorem of probability theory, which describes the results of performing the same experiment a large number of times. In broad terms, this law states that the average of the results by a large number of trials should be close to the expected value, and as more trials are performed, this result should appear closer and closer [4].

Explained in a more mathematical approach, we have Theorem 1.1 for the LLN [5].

Theorem 1.1 (Law of Large Numbers). *Let $X_1, X_2, ..., X_n$ be an independent trial process, with finite expected value $\mu = E(X_j)$ and finite variance $\sigma^2 = X(X_j)$. Let $S_n = X_1 + X_2 + ... + X_n$. Then for any $\epsilon > 0$,*

$$P\left(\left|\frac{S_n}{n} - \mu\right| \geq \epsilon\right) \to 0$$

as $n \to \infty$. Equivalently,

$$P\left(\left|\frac{S_n}{n} - \mu\right| < \epsilon\right) \to 1$$

as $n \to \infty$.

■

1.1 Example: Die roll

To better understand the LLN we also have an example 1 from the same book [5], replacing some things from the Theorem 1.1 we have,

Example 1. *Consider n rolls of a die. Let X_j be the outcome of the jth roll. Then $S_n = X_1 + X_2 + ... + X_n$ is the sum of the first n rolls. This is an independent trials process with $E(X_j) = 7/2$. Thus, by the Law of Large Numbers, for any $\epsilon > 0$*

$$P\left(\left|\frac{S_n}{n} - \frac{7}{2}\right| \geq \epsilon\right) \to 0$$

as $n \to \infty$. An equivalent way to state this is that, for any $\epsilon > 0$,

$$P\left(\left|\frac{S_n}{n} - \frac{7}{2}\right| < \epsilon\right) \to 1$$

as $n \to \infty$.

■

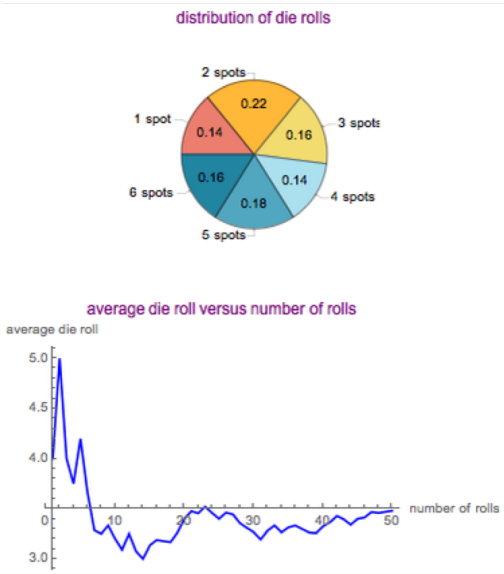
This example tells us that, according to the LLN if a large number of six sided fair dice are rolled, then the average of their values is likely to be close tho 3.5, with an increasing precision the more dice are rolled. Thanks to a tool from Wolfram we are able to visualize the different stages of this process in Figure 1.

As we can see in Figure 1 the mean is set to be 3.5, and as the iterations keep building, the line representing the average sum gets closer to converge to the mean line.

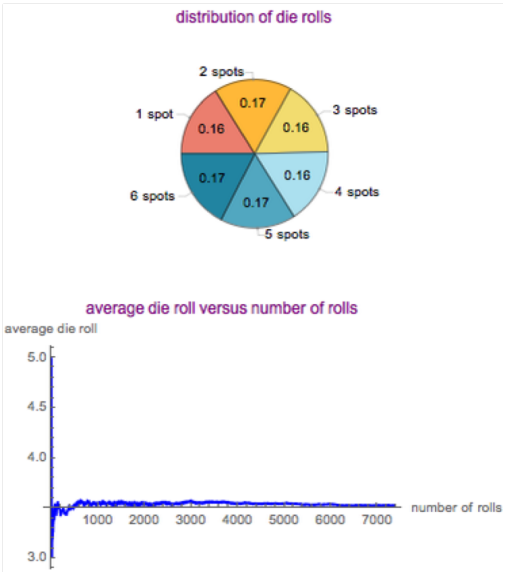
We can also develop a similar experiment in R, with simple functions such as `sample`. In this case, we take the average sum of two rolled dice. In Figure 2 we represent with a lollipop plot the behavior of the different sizes of iterations of the experiment.

```r
two.dice <- function(){
  dice <- sample(1:6, size = 2, replace = TRUE)
  return(sum(dice))
}

two.dice()
replicate(n = 20, expr = two.dice())

sims <- replicate(100, two.dice())
table(sims)
df = as.data.frame(table(sims)/length(sims))
barplot(table(sims)/length(sims), xlab = 'Sum', ylab = 'Relative Frequency', main
    = '100 Rolls of 2 Fair Dice')

# Libraries
library(ggplot2)

# Create data
data <- data.frame(x=df$sims,y=df$Freq)

# Plot
png("Ej13_dice1.png", width = 1000, height = 1300, res = 300)
ggplot(data, aes(x=x, y=y)) +
  geom_segment( aes(x=x, xend=x, y=0, yend=y), color="grey") +
  geom_point( color="orange", size=4) +
  theme_light() +
  theme(panel.grid.major.x = element_blank(),
    panel.border = element_blank(),
    axis.ticks.x = element_blank()) +
  xlab("Sum") +
  ylab("Relative Frequency")
dev.off()
```

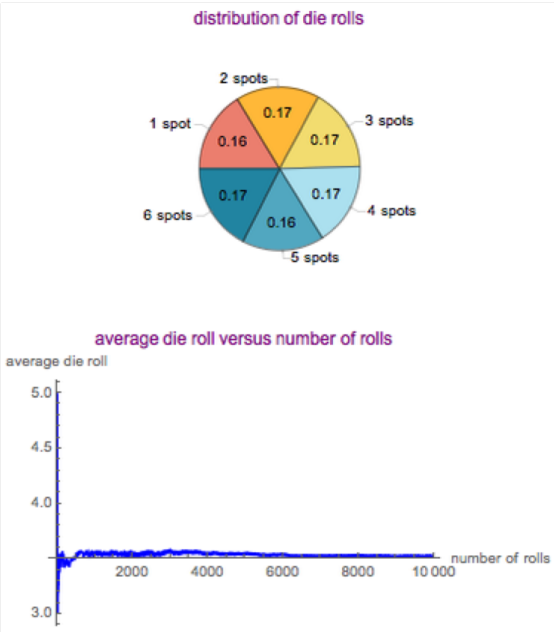
Listing 1: R extract of the code used to perform the dice experiment



(a) 50 rolls of the dice

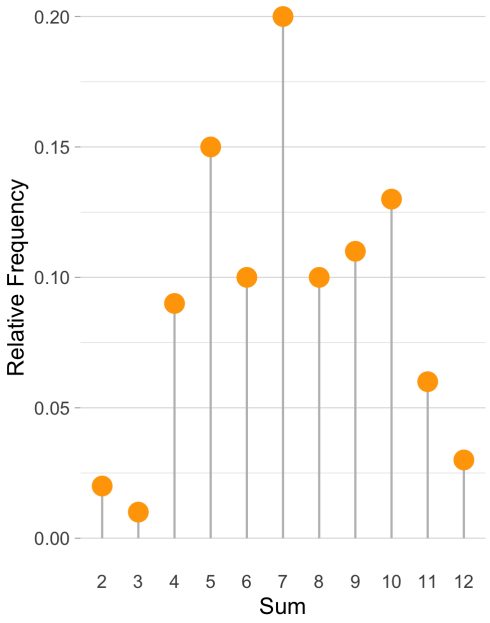


(b) 7,500 rolls of the dice

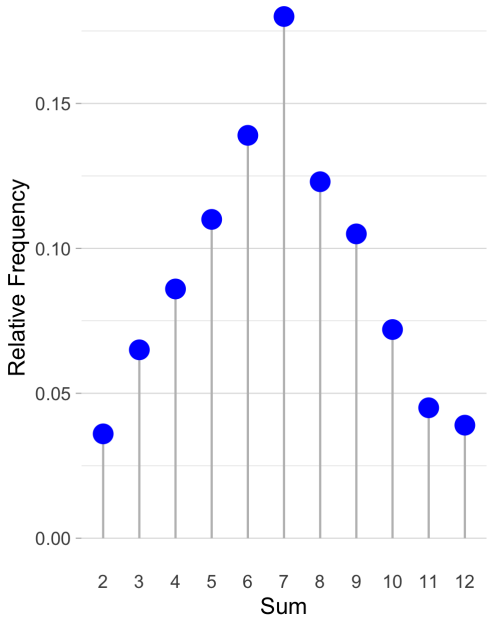


(c) 10,000 rolls of the dice.

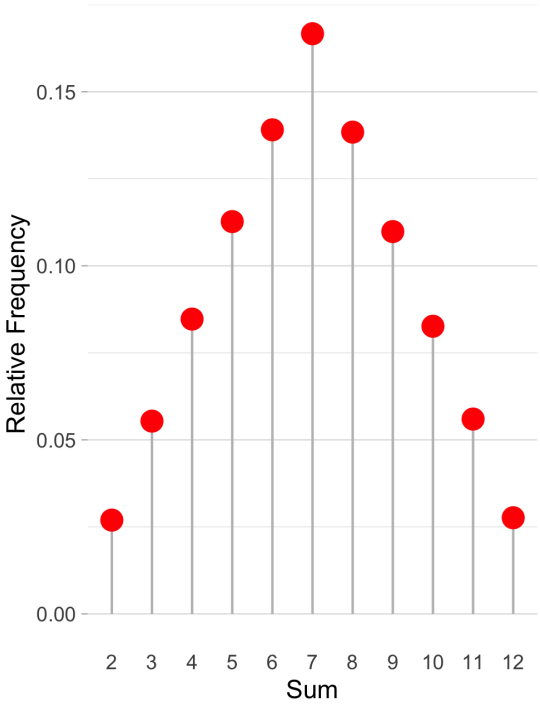
Figure 1: Different parameters for the dice experiment in the Wolfram page.



(a) 100 rolls of the dice



(b) 1,000 rolls of the dice



(c) 10,000 rolls of the dice.

Figure 2: Different parameters for the dice experiment performed in R.

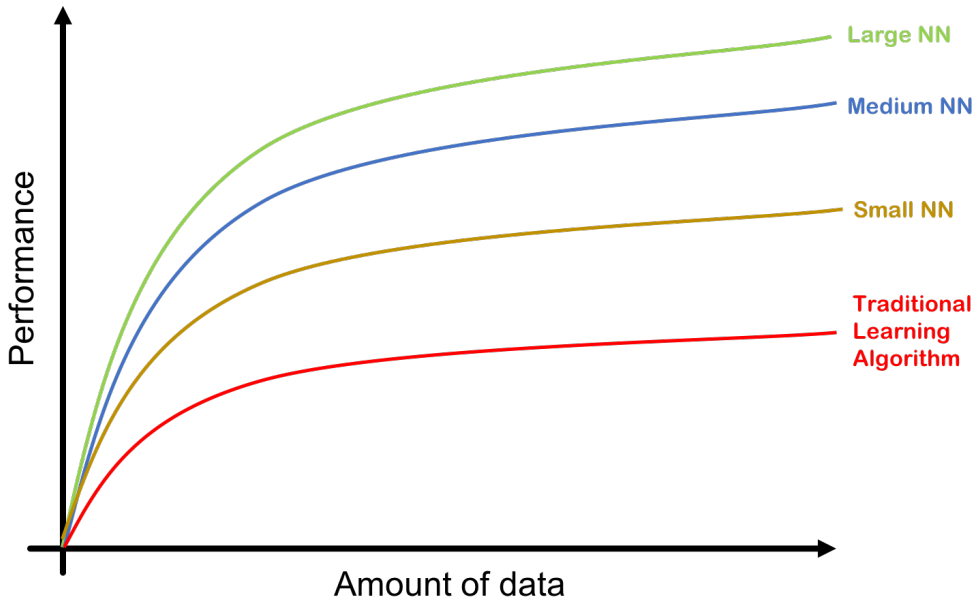


Figure 3: Representation of the learning curve with more amount of data.

2 Applications of Law of Large Numbers in Machine Learning

Now that the concept of the law of large number is presented, we searched for a topic related to the work of our thesis. In this case, there is a question that has always been present when working with Machine Learning tools such as Neural Networks (NN), and in this case is, how much data do I need to properly train my NN.

It all comes down to accuracy. When a experiment is performed, an accurate and precise result is desired. Accuracy of an experiment is defined as to whether or not the result of a measurement conforms to the correct value or expected value. Precision refers to the degree to which these values agree and can have a repeatable outcome if we performed the experiment again. The basic notion of this comes from the Law of Large Numbers (LLN), one of the basic principles of experimental physics and statistics. The law states that the average of an experiment performed many times converges to its true or expectation value.

In ML applications, this can translate as to training our NN with a defined set of data, and then testing this model with some more sets of data. The performance of the model we create depends upon the amount of data we use. When the data increases, we force the algorithm to fit the data, and with that, we minimize the error [1].

In Figure 3 we see the representation often used in articles to show the same behavior represented in the example of the dice 1 we saw in the introduction, where we can see how the learning curve slowly flattens the more data we add to our model.

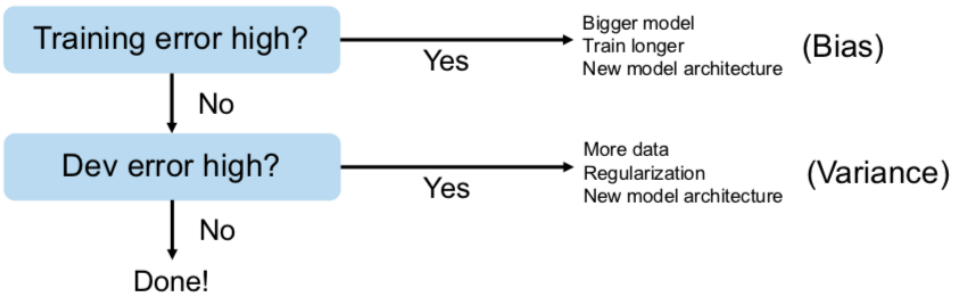


Figure 4: Basic recipe for machine learning [6]

Since our main goal is that our model gives us the best accuracy possible, we also have to consider the distribution of our data. Many cases have shown that the expected result of our test set will not match the accuracy performance of our training and developing sets if we use different distribution of data [6]. This also comes around to having a great amount of data to use for our model. Figure 4 shows the "basic recipe" for machine learning.

As one Google Translate engineer put it, "when you go from 10,000 training examples to 10 billion training examples, it all starts to work. Data trumps everything".

In our case, at the beginning of our Masters thesis [3], we started to develop toy CNN with the available data we had, which was not much. The maximum accuracy we could achieve for our image detection model, was of 65% to 70% in the test set. Later, when we were able to get more data sets, our accuracy improved to a 89% accuracy.

As a final note, some of the things we learned along the way, is that is not always possible to have a large data set available to train our models, and in this case, there has been several application of transfer learning, that basically uses data form a different distribution (in our case, normal images, not medical ones) to pre train the wights of the NN, and improve the results of our training and testing. We are currenty looking for more examples of this in the medical field, since the ones we already found are lacking in accuracy performance.

3 Conclusion

The theory of Large Numbers applied in Machine Learning was something we heard of in a conference of the ENOAN in Zacatecas that we were able to attend. In this case, it was a really interesting topic for us, because in our mentioned masters thesis, we had a little idea when it came to one of the optimizers we used to perform our experiments in the architecture of the classification model for our CNN.

The SGD or Stochastic Gradient Decent optimizer, was one of the more robust algorithms used, since in all of the models we trained, it consistently kept upgrading the accuracy in the model, ending up in a steady 85% to 95%, and the difference in percentage from the training and

tests sets where not so far apart. Even the optimizer we ended up using for the final experiment had some models in which the accuracy went bellow 40%.

In the conference we mentioned, Dr. Hugo Estrada Esquivel with his lecture titled "Ciencia de datos en la era del internet de las cosas" [2] commented about the experiments he had developed trying to stabilize the SGD optimizer. In his work, he found out, as we did, that the SGD optimizer is the most steady one, but also that it need significantly more time to converge in a optimal solution. Similar to the experiment in Figure 1 it gets close to the desired output, but eventually swerves again.

We did not follow up with this investigation, because it was very costly (computational wise) to perform this type of experiments, but it would be very interesting to dig up again some information about it, to see if it has made some progress.

References

[1] The law of large numbers in ai and big data. https://discover.bot/bot-talk/law-of-large-numbers-ai/. Accessed: 2020-11-30.

[2] Ciencia de datos en la era del internet de las cosas. http://smcca.org.mx/enoan2019/plenariaHugoEstrada.html. Accessed: 2020-11-30.

[3] Clasificación de mamografías mediante redes neuronales convolucionales. http://eprints.uanl.mx/17656/. Accessed: 2020-11-30.

[4] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how.* Springer Science & Business Media, 2005.

[5] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability.* American Mathematical Soc., 2012.

[6] Andrew Ng. Machine learning yearning. *URL: http://www. mlyearning. org/(96)*, 2017.

Practice 14: Central Limit Theorem.

Mayra Cristina Berrones Reyes 6291

December 8, 2020

1 Introduction

The Central Limit Theorem (CLT) tells us that in many situations, when independent random variables are added with unknown distribution (it could be uniform, binomial or completely random), the sample means will approximate the normal distribution, such as a bell curve.

Often, the CLT is confused with "the law of large numbers". This law states that as the size of a sample increases, the sample mean will become a more accurate estimate of the population mean. The main difference between the two theorems is that the law of large numbers pertains to a single sample, meanwhile, the CLT pertains to the distribution of sample means [2].

As statistical significance goes, the CLT [1]:

- Helps us analyze data like hypothesis testing and constructing confidence intervals, because these methods assume the population is normally distributed, so we can treat the sampling distribution as normal according to the CLT.
- When we increase the samples form the population, the standard deviation of sample means will decrease. This helps us estimate the population mean much more accurately.

The mean of the sample means is denoted as Equation 1 and the standard deviation of the sample mean is denoted as Equation 2

$$\mu_{\overline{x}} = \mu, \tag{1}$$

where,

- $\mu_{\overline{x}}$ is the mean of the sample means.
- μ is the population mean.

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}}, \tag{2}$$

where,

- $\sigma_{\overline{x}}$ is the standard deviation of the sample mean.
- σ is the population standard deviation.
- n is the sample size.

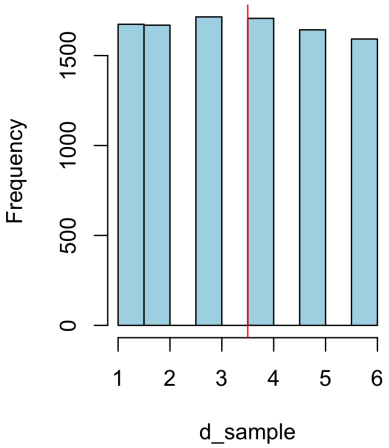


Figure 1: Histogram of the frequency of each outcome.

1.1 Example: Die roll

Following similar steps as the practice of Law of Large Numbers (LLN), to further analyze and understand the CLT, we use the example of the dice and with code in R 1.

A fair die can be modeled with a discrete random variable with outcome 1 through 6, each with the equal probability of $\frac{1}{6}$. The expected value is $\frac{1+2+3+4+5+6}{6} = 3.5$. In this experiment we are going to simulate throwing a fair die 10,000 times where we first have as a result Figure 1.

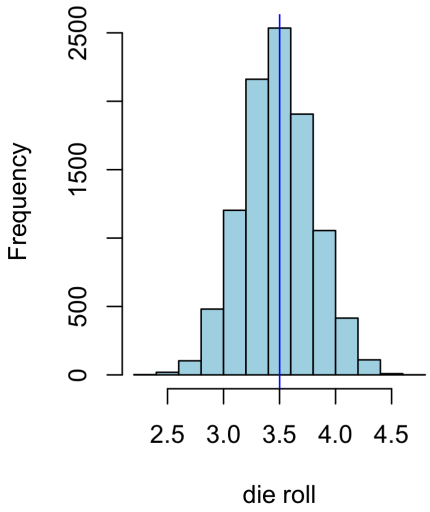
```
png("Ej14_dice.png", width = 1000, height = 1300, res = 300)
d_sample <- sample(1:6,10000, replace= TRUE)
hist(d_sample, col = "light blue",main="")
abline(v=3.5, col = "red",lty=1)
dev.off()

x30 <- c()
x100 <- c()
x1000 <- c()
k =10000
for ( i in 1:k){
  x30[i] = mean(sample(1:6,30, replace = TRUE))
  x100[i] = mean(sample(1:6,100, replace = TRUE))
  x1000[i] = mean(sample(1:6,1000, replace = TRUE))
}

png("Ej14_dice1.png", width = 1000, height = 1300, res = 300)
hist(x30, col ="light blue",main="",xlab ="die roll")
abline(v = mean(x30), col = "blue")
dev.off()

png("Ej14_dice2.png", width = 1000, height = 1300, res = 300)
hist(x100, col ="pink", main="",xlab ="die roll")
```

```
abline(v = mean(x100), col = "red")
dev.off()

png("Ej14_dice3.png", width = 1000, height = 1300, res = 300)
hist(x1000, col ="orange",main="",xlab ="die roll")
abline(v = mean(x1000), col = "red")
dev.off()
```

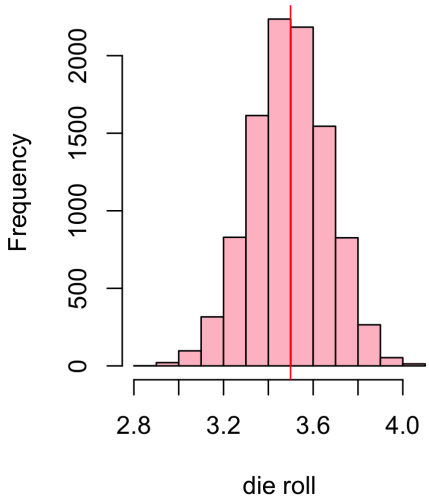
Listing 1: R extract of the code used to perform the dice experiment

We will take samples of size 10, from the above 10,000 observations of the outcome of die roll. From this, we will take the arithmetic mean and try to plot the mean of sample. This procedure will be reproduced k times (in this case k= 10,000).

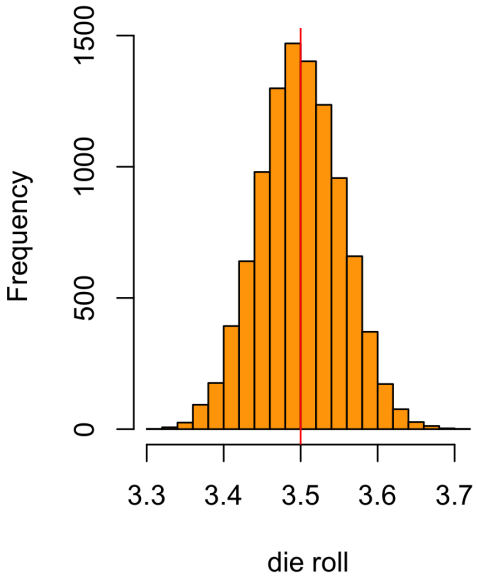
By theory , we know as the sample increases, we get better bell shaped curve. As the n approaches infinity , we get a normal distribution. We will achieve this by increasing the sample size to 30, 100 and 1,000, and the results can be seen in Figure 2.



(a) n = 30



(b) n = 100



(c) n = 1,000

Figure 2: Different parameters for the dice experiment performed in R.

Table 1: Calculation of the standard error of the dice example.

Sample size		
10	100	1,000
$\sigma_{\overline{x}} = \frac{1.71}{\sqrt{10}} = 0.54$	$\sigma_{\overline{x}} = \frac{1.71}{\sqrt{100}} = 0.17$	$\sigma_{\overline{x}} = \frac{1.71}{\sqrt{1000}} = 0.05$

2 Applications of Central Limit Theorem in Machine Learning

The CLT has important implications in applied machine learning. The theorem does inform the solution to linear algorithms such as linear regression, but not exotic methods like artificial neural networks that are solved using numerical optimization methods. Instead for these type of methods, we must use experiments to observe and record the behavior of the algorithms and use statistical methods to interpret their results.

For the significance test, we can make comparisons of the performance of one model against another model. In this case, we can use the tools of statistical significance to estimate the likelihood that the two samples of model skill scores were taken from the same or a different distribution of model scores. If it looks like the samples were drawn from the same population, then we assume there is no difference between the models, and any differences are due to statistical noise.

Another example is when we have trained a final model in machine learning, we want to make a guess about how good the model is expected to be in practice. This can be called a confidence interval. We can develop multiple independent evaluations of a model accuracy to result in a population of model accuracy estimates. The mean of these estimates will is regarded as the error of the true estimate of the model accuracy on the problem.

To illustrate this, we can actually take the example of the dice, and verify the standard error on each of the experiments seen on Figure ?2.

For that example we have the variance of rolling a fair die equal to 2.92 and the standard deviation to 1.71, so we have the results in Table 1. In that table we can see, as the sample increases, the error gets smaller and smaller.

3 Conclusion

In this case with the CLT definition, a part of the process of training a Neural Network comes to mind. When training the models used for our investigation of the thesis, we focused on comparing the efficiency and accuracy of the models. In the training process, there is a sort of monitoring of the results and development of the model. Similar to what we did here, we chose a set number of iterations in which our models are to be trained. This iterations are also divided in epochs, that are a form of larger way to see the iterations and the slow process of improving the accuracy of your model.

The way it improves is dependent of the loss function you choose. In our case, we chose the

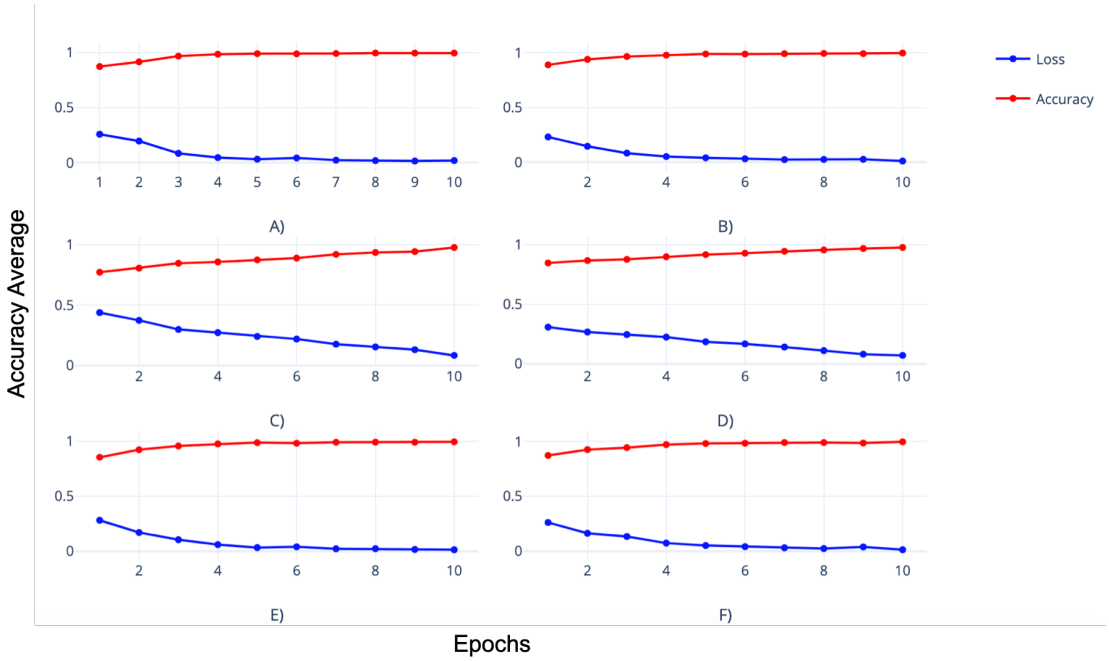


Figure 3: Loss and accuracy of the best models.

binary cross-entropy (specifically for the type of data we had) and the main goal is to reduce the loss function a close to zero as we can, and have the accuracy as close to 1. This is the part of the process that most resonated while reading for examples, because of the way we look for one part of the process to be as close as zero and the other to converge to 1 (this part more so in the Law of Large Numbers).

In Figure 3 we have a plot of the loss and accuracy pf the best models trained for the experimentation of our thesis. As we can see, in the best models, the accuracy is close to one in the last epoch, and the loss is closest to zero.

In conclusion, understanding the central limit theorem is important when it comes to trusting the validity of the results of the models we train and assessing the precision of the estimated accuracy it gives.

References

[1] Statistics for data science: Introduction to the central limit theorem. https://www.analyticsvidhya.com/blog/2019/05/statistics-101-introduction-central-limit-theorem/. Accessed: 2020-12-08.

[2] Why the central limit theorem in data science. https://towardsdatascience.com/why-the-central-limit-theorem-in-data-science-be8997b95f3. Accessed: 2020-12-08.

Practice 15: Ideas for the final project.

Mayra Cristina Berrones Reyes 6291

December 14, 2020

1 Central Limit Theorem.

For this subject, the idea was to make some experiments with an available dataset (we are thinking the MNIST, since it is large enough) and experiment with different size of the test set, to see if the accuracy told by the trained model is consistent with the test set with all the different sizes of data.

This experiment comes in line with some of the questions we had when we finalized our work in the masters thesis. One of the revisers asked if we had tried to see if changing the data sets size, our model improved or declined in accuracy.

2 Bayesian Theorem.

For the investigation we made of the Bayesian Theorem, we had the idea to evaluate several known Neural Networks used for transfer learning with the data set of Mini MIAS, which is a free data set of mammography's of that are already annotated. This is a very small data set (only 362 images in it) but as researched before, transfer learning thrives with small datasets because they are already pre trained with some weights.

This could show us a stepping point to use with our actual images, conformed of a larger dataset, and with greater resolution (heavier to train).

3 Law of large numbers.

For this subject, we wanted to explore some of the qualities of different optimizers used in training convolutional neural networks. Each one of them has different features that try to correct the failings of its predecessors. And it is because all of this different versions that there is not one optimizer that is perfect for a certain problem. With the LLN we want to see if the optimizers are affected in a good or a bad way. If it helps them converge to the closest to optimal value, or if in some cases it becomes flawed and lands in over training.

4 Convolutions.

Convolutions are often used in types of Neural Networks. They are also used to enhance or add noise to some images. In our case, there are very few Kernels that are allowed in articles to

be used in medical images, because they ofter distort the image in a way that adds features that can impair the classification and diagnostic process.

In many articles for normal images, noise is added to the dataset to make the model more robust. Here we propose an experiment with two different datasets. The Mini MIAS and the MNIST to see if the same Kernels that help the MNIST dataset to be more robust, help improve or diminish the accuracy on the Mini MIAS dataset.

Practice 16: Reviews of my classmates.

Mayra Cristina Berrones Reyes 6291

December 15, 2020

1 Alberto M.

Inferencia y estadísticas bayesianas para la imputación de datos en datasets Los datos faltantes son problemas muy comunes encontrados en los datasets de la vida real, estos pueden perturbar el análisis de datos dado que disminuyen el tamano de las muestras y en consecuencia la potencia de las pruebas de contraste de hipótesis, además hace que no se puedan utilizar directamente técnicas y modelos de machine learning, deep learning. Los anterior nos lleva a la necesidad de rellenar o imputar datos en datasets, existen diversas técnicas para lograr este objetivo como la sustitución por la media, la sustitución por constante, imputación por regresión, entre otras. Dichas técnicas tienen ciertas deficiencias, por lo que en este trabajo se utilizará la estadística inferencial y bayesiana para la imputación flexible de datos faltantes en algunos datasets.

(Mayra) Este trabajo también me parece bastante interesante. Como mencionas, la parte de rellenar datos faltantes es muy importante cuando estas trabajando temas de minería de datos, sobre todo cuando la indicación general para tratar con datos faltantes es trabajar alrededor del problema o de plano eliminarlos de tu investigación. Me gustaría bastante ver que técnicas encuentras para poder subsanar este problema. Un pequeno detalle es nada mas el uso que le das a la palabra de Imputación. La palabra rellenar creo que se entiende perfectamente. Imputación se refiere a algo más como dar la culpa de algo. No se si querías usar más o menos ese significado.

2 Palafox.

Networks arise in many scientific and technological fields [Newman, 2018]. The internet, social networks, electrical net- works, are among many available examples. To study network processes, sometimes it is convenient to have a model which preserves the essential characteristics of the network. A random graph is a model network in which the values of certain properties are fixed, but the network is in other respects random [Newman, 2018]. For example, a number n nodes and m edges could be fixed, but edges between any two nodes placed at random. The aim of this project is to do a theoretical and computational study of random graphs, and analyze how closely some of these resemble real world networks [Leskovec and Krevl, 2014].

(Mayra) La teoría de gráfos es un tema que en lo personal me parece bastante interesante. Entiendo el enfoque que le quieres dar al final a tu tema, que es explicar de manera teórica el comportamiento de los gráfos. Pero si estaría super interesante si, en caso de hacer este es el trabajo que quieras desarrollar, intentes con datos reales. Si no me equivoco, escuchamos algo de esto en una conferencia acerca de las redes eléctricas. También hay algunas aplicaciones en el

area de transporte.

3 Gabriela.

Comparación de soluciones: Como parte del trabajo de tesis se tienen datos sobre las soluciones obtenidas con dos formulaciones diferentes, se desea analizar dichas soluciones para verificar si hay diferencias significativas entre ambas. Como primera instancia se pretende usar estadística descriptiva y después verificar con pruebas de hipótesis que sean aplicables a los datos.

(Mayra) Esto es porque conozco tu tema, y si me gustaría bastante ver la diferencia que hay entre los dos modelos que tienes. Lo que no me queda muy claro es la parte en donde describes la primera instancia que vas a utilizar en tu experimento. Entiendo la parte en que vas a realizar estadística descriptiva. Lo demás me gustaría escucharte explicarlo.

Use law of large numbers to measure Neural Networks optimizers performance[☆]

Mayra Cristina Berrones Reyes[1]

Universidad Autónoma de Nuevo León. Facultad de Ingeniería Mecánica y Eléctrica

Abstract

In this work, we want to explore some of the qualities of different optimizers used in training Neural Networks. With the Law of Large Numbers we want to see if the optimizers are affected in a good or a bad way when given more iterations of training. If it helps them converge to the highest average of accuracy percentage, or if in some cases it becomes flawed and lands in over training.

Keywords: Neural Networks, Statistics, Optimizers, Law of Large Numbers

2020 MSC: 00-01, 99-00

1. Introduction

In the subject of problem solving and finding an optimal solution to issues we encounter in our day to day lives, a popular item is the topics of Machine Learning (ML) and Artificial Intelligence (AI), which mainly promises to find a faster and acceptable solution. Some of this methods, however, rely heavily on a random factor, that allow them to sometimes find an answer faster than in the case of traditional statistical methods.

So the question becomes here, is there a difference between statistics and ML? In this case, the answer can be that they are certainly similar in some aspects, as the two fields are converging more and more in different subjects, for example, both ML and statistics are used nowadays on techniques of pattern recognition, data mining, knowledge discovery, etc.

[☆].

[1]San Nicolás de los Garza, Nuevo León. México.

Defining them separately, we have that ML is a sub field of computer science. It concentrates on building systems that can learn from data, instead of relaying on explicit instruction of programs. A statistical model on the other hand has a more mathematical background. The main difference between the two is that ML focuses on optimization, performance and finding generalizable predictive patterns while statistics is more concerned with the inferences it can make from a sample of the population [1].

For a ML model, any prior assumptions about the underlying relationships between the variables we are studying are not needed in order to start building your model. In some instances, we can just give all of the data we have, and the algorithm can process the data and find patterns in it. Up until a few years back, all the process and calculations that the algorithm of ML made in its way to learn and find a suitable answer where treated as a *black box*, with a mentality that, as long as it works, we do not have to concern ourselves with how it got to the final result.

This practice stoped being acceptable when more and more scientist wanted to use models based on ML and AI to solve real life problems, but got rejected because they could not prove with detail why their model worked the way it did. In direct contrast, in the field of statistics first we need to collect and understand certain features of the data we are working with. How it was collected, the statistical properties, the distribution of the population we are using, etc.

Here we find another difference between these two fields. Generally speaking, ML modeling thrives on high dimensional data sets. The more data you give your model, the more accurate your prediction ends up being. In the case of statistical modeling, they are usually applied on low dimensional data sets.

In a comprehensive comparison between these two fields [2] we find an example of the main difference at interpreting results from each model in List 1. In this example we see that statistical models offer a better chance at reproducibility of the experiments than the models of ML. This is a highly valued feature when publishing an article of these sort of investigations.

- **Machine Learning model:** "The model is 85% accurate in predicting Y, given a, b and c."

- **Statistical model:** "The model is 85% accurate in predicting Y given a, b and c, and I am 90% certain that you will obtain the same results."

Overall, regarding all of these points, it may seem that ML and statistical modeling are two separate branches of predictive techniques. These differences however, have been reduced significantly over the last decade, where statistical models have adopted some methods from machine learning, creating an emerging field called statistical learning. ML in return tries to implement statistical strategies to justify the behavior of the model, its results, and dispel the idea of the *black box* when it comes to the calculations of the model.

Following the idea of these two fields working together, in this work we take a closer look at one of the most popular uses of ML modeling, which is Neural Networks (NN) and its derivates.

2. Background

As mentioned in the Introduction section 1 the main goal for this article is to approach a machine learning issue with a statistical set of mind, and see if we are able to arrive to a reasonable answer. In this case, we have a theoretical problem for a machine learning algorithm known as Convolutional Neural Network (CNN).

In previous works [3] (article in revision) we have worked with a self made architecture of a CNN model that will help us classify medical images of mammograms, and help medical professionals unload their pre processing work by labeling the data of images with and without anomalies. In this works, we trained and modeled about 200 different architectures, changing parameters such as optimizers, size of the image, number of neurons, number of hidden layers, etc.

Finishing all of those experimentations with the different parameters of the CNN took us almost a whole year, even with the use of powerful hardware, and parallel programing. At the end of the experimentation, we concluded that one of the most important feature that helped us determine which model architecture worked better was the optimizer we used. Adam and SGD where the more robust, having all of their models at a steady percentage of accuracy above the 95%, and Adadelta

was the one that had two of the models with best accuracy out of all of them.

Part of the training process of a NN model is to change some of the wights to try to minimize our loss function and maximize the accuracy of our predictions. This is where the optimizers come in. They help with the update of parameters in response to the loss function, and depending on the type of optimizer we choose, it determines how big the changes must be in each iteration. This is known as a learning rate. We now have a brief introduction to each of the three optimizers that we are working on this article. (For a more in depth explanation we suggest the following articles [4] [5] [6])

2.1. SGD

We begin by discussing the SGD optimizer, since it is one of the oldest approaches to an optimization algorithm. It stands for stochastic gradient descent. Starting form a initial value, this algorithm runs iteratively to find the optimal value given by the cost function. It is very simple, but perhaps for its simplicity, it finds several problems, for example, compared to other optimizers, it converges at a slower rate. It also has problems with being stuck in a local minimum. Newer approaches have outperformed SGD in optimizing the cost function, so some boosters have been implemented to correct its disadvantages, like momentum [7], nesterov [8] or a combination of both.

2.2. Adam

Adam stands for adaptive moment estimation. It is one of the most popular optimizers used in machine learning, as it is the one that performs best on average. It uses the same concept of the SGD plus momentum and adaptive learning rates to converge faster to the optimal value. The concept of adaptive learning rates can be pictured as a match of golf. We are allowed to move faster initially, but as the learning rate decays, we take smaller and smaller steps, allowing a fast convergence, since there is less chance of overstepping our goal.

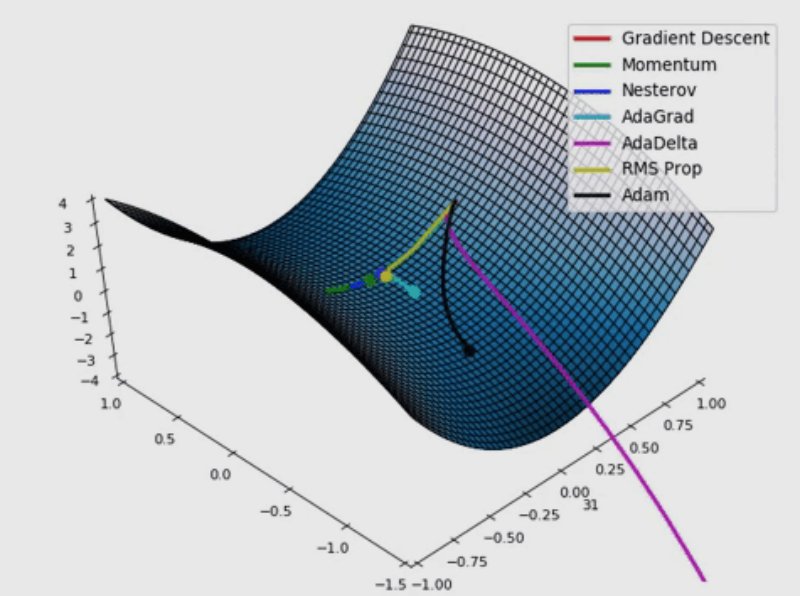


Figure 1: Comparison of other optimizers [9]

2.3. Adadelta

Now with Adadelta, we can say that it is like a progression of the previous optimizers. As Adam, this is an algorithm based in an adaptive learning rate. It more closely resembles Adagrad (another optimizer of the same distribution as Adam) and in this case, it seeks to reduce the aggressive decreasing learning rate. Compared with all the other optimizers, Adadelta is the one that seems to converge the fastest, as shown in Figure 1.

Selecting this three optimizer we continued to do experiments with the other parameters of the CNN that we could change. An important distinction was that, for all of the changeable parameters we had, we stablished some that remained fixed for the entire experiment. One of these parameters was the number of iterations and epochs we used to train all the models. In our case we had 10,000 small iterations, and 10 epochs (large iterations). For the validation set we had 2,000 small iterations.

A quick way to explain the difference between these mentioned small and large iterations is as follows. In the first epoch or large iteration, we train our model with 10,000 small iterations. At the end of this first epoch, the model is expected to have a small percentage in accuracy,

and when the small iterations end, the model is tested against the developing set for 2,000 small iterations. This helps the model modify some of its weights in preparation for the next epoch. When all of the epochs end, the model has finished its training, and it is expected that the resulting accuracy percentage closely resembles the accuracy we will get when we use the model on the test set.

When we made the comparison of these three optimizers we found that, for some models of the Adam and Adadelta optimizers, their performance on the test set was not consistent with the result on their training, while in the case of the optimizer SGD all of their accuracy remained in a steady range. Robustness in a model and reproducibility is something well sought out by researchers, and SGD provided that for our models. However, none of the models ever reached an accuracy percentage above of 96%. Doing a thorough scan of the behavior of all of the models from these three optimizers we realized that Adam and Adadelta reach a high accuracy percentage by the eighth epoch of training, and all the accuracy achieved after that comes in very small intervals.

In contrast, we saw that the models with a SGD optimizers had a very slow climb of their percentage accuracy, but it continued its constant pace, til it ends in the tenth epoch (as all the other models). This behavior made us believe that, if we gave more iterations to these models, than the accuracy will improve, and maybe even surpass the accuracy of all other models eventually.

This can also be said in favor of the other two optimizers, so, in order to compare fairly these experimentation, we will need to reach a point in which there is no discernible improvement in the performance of any of the optimizers, and see if they average to an accuracy above that of what we registered so far.

The problem now is that we do not have the time or the computational resources to do an experimentation of that scale with the same dataset, specially with the amount of epochs we are trying to achieve, so in this experiment we will be using a toy dataset of images of cats and dogs, which can come close to the binary experimentation we used for out previous models. In the next section we see a brief explanation of the inner workings of our experiment, and a discussion of some statistical themes that will help us strengthen our results.

2.4. Law of large numbers

The idea behind this experiment, comes with the basis of the Law of Large Numbers (LLN). The LLN states that if you repeat an experiment a large number of times, averaging the results, then your answer should be as close as the expected value. In Theorem 2.1 we see the common notation for the sample mean [10].

Theorem 2.1 (**Law of Large Numbers**). *For i.i.d. random variables X_1, X_2, ..., X_n the sample mean, denoted by $\overline{X}$, is defined as*

$$\overline{X} = \frac{X_1 + X_2 + ... + X_n}{n}$$

Another common notation for the sample mean is M_n. If the X_is have CDF $F_x(x)$, we might show the sample mean by $M_n(X)$ to indicate the distribution of the X_is.

■

Since we have that the LLN often deals with any sort of trial with a probabilistic outcome, the application in our case becomes clearer. Machine learning systems, such as a CNN adapt very quickly to a large amount of data being feed into it. The data entries will in this case represent the trials, and the resulting average of accuracy of the training process are the patterns and features that the CNN has to make a classification.

The basic nature of the learning process of the CNN is exactly what the LLN represents as a mathematical approach, only translated to a more operational format. In other words, the process of each epoch in our model can translate loosely at what the LLN represents. So in this experiment we give all three optimizers the chance to even out their learning rate by giving them more iterations to train the data [11].

3. Methods and materials

As we stablished in the Background section, there are many parameters in a CNN that can be tuned to try to improve the performance of our model. The main propose of this experiment is to

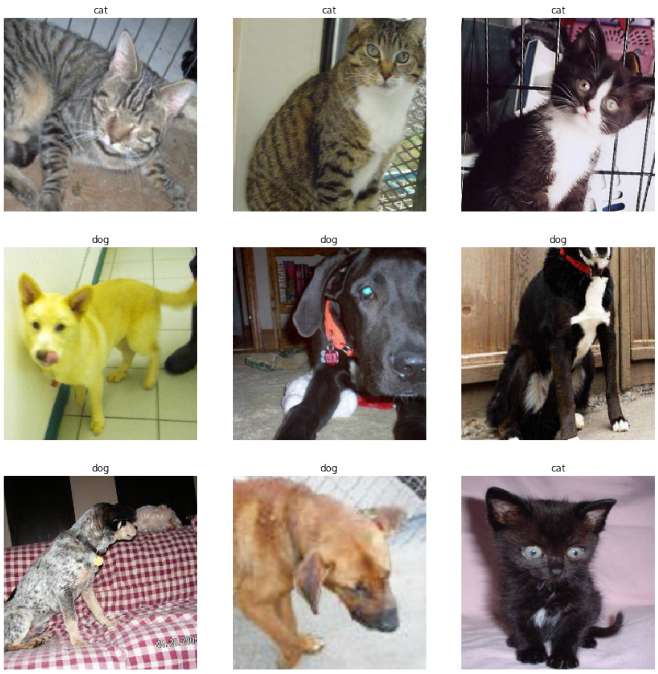


Figure 2: Example of some of the images on our dataset

establish if it is worth it to spend the time and computational processing to give more epochs to the models, to see if they can reach a higher accuracy percentage. We test this hypothesis with a toy data set of images of dogs and cats, and pre trained weights pulled from the ResNet50 neural network to accelerate the training process.

All of the experimentation was carried out in a Google Colab notebook, in a MacBook Pro 13-inch 2016.

3.1. Dataset and models

For the experiment we are using a dataset consisting on images of cats and dogs. The data sets are divided by 23,000 images on the training set, 2,000 images for the validation or developing set, and lastly 12,500 images for the test set. In Figure 2 we have an example of some of the images of our toy dataset. The resolution of all of them is kept at 150×150 pixels.

The libraries used on the experiment we have Keras, Matplotlib, sklearn and h5py.

We mentioned before that, in order to keep the time of the training process as minimal as possible, we began with pre trained weights of the ResNet50 using as input imagenet [2]. We then assigned the paths to the different datasets, and added some image augmentation by using the data generator that the library Keras has installed.

As discussed in previous sections, we are giving the models more iterations to train. In this case we have 300 epochs for each model. We add a restriction to each epoch instead of small iterations, that monitors the loss function, and when it detects that there is no improvement for more than 5 iterations, with a slope of 0.02 on the learning rate, gives a break and passes to the next epoch. This is a common practice to avoid overfitting your model.

4. Experiments with Adam, Adadelta and SGD

In Figure 3 we see the accuracy performance of the training set. We can appreciate the same feature explained in the Background section, in which thanks to the adaptive learning rate, the optimizers Adam and Adadelta move to a very high accuracy since the first epochs, and slowly even out to a more horizontal behavior. Still, it is barely visible, but comparing these two we see that, despite the little bumps along the line to the 300 epoch, the Adam optimizer is still climbing the accuracy.

In the case of the SGD optimizer, as we predicted, the curve does not even out until more than half of the epochs have passed. Again, it is almost imperceptible because of the little bumps, but we can see that the line has a small tendency of going upward.

In Figure 4 we have the accuracy of the validation set. In this case, for the Adadelta and the SGD optimizers we do reach a line where no improvement is made throughout the rest of the epochs.

[2] Repository with experimentation. https://github.com/mayraberrones94/Probabilidad/blob/master/ProyectoFinal/Code/Elisa_Exp.ipynb

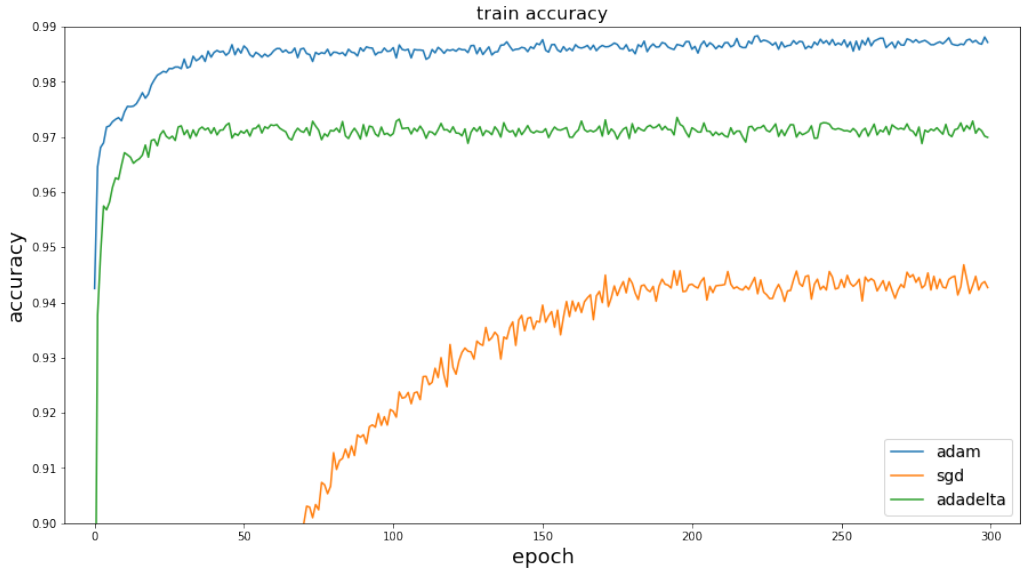


Figure 3: Plot of the accuracy of the models after 300 epochs in the training dataset

Adadelta reaches it in roughly the same time that in Figure 3 the curve evens out. These two optimizers also improved their accuracy in comparison with the training.

In the Adam optimizer, the spikes on the line become more prominent than in the training, and when it finally evens out on the last epochs, we can see that it does not reach the same accuracy as the training.

In Figure 5 and 6 we have the loss results for the training and validation sets respectively. As we can see, the behavior is similar to the plots for the training and validation accuracy, in the sense that the loss for the training appears to be less stable, and overall in all the optimizers the slope of the validation loss seems to behave better than in the training.

4.1. Experiments with SGD boosters

In the Background section we mentioned that SGD is one of the oldest optimizers, so it is the more simple of all. That is why there are some boosters that improve the accuracy of the SGD optimizer. We mentioned them as momentum and Nesterov. In the case of Nesterov, in the code you

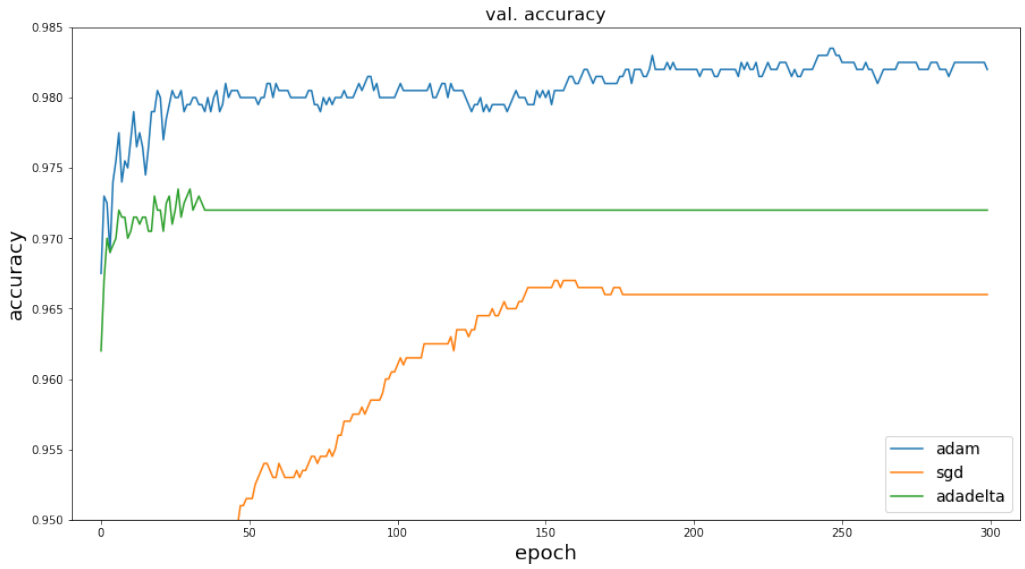


Figure 4: Plot of the accuracy of the models after 300 epochs in the validation dataset

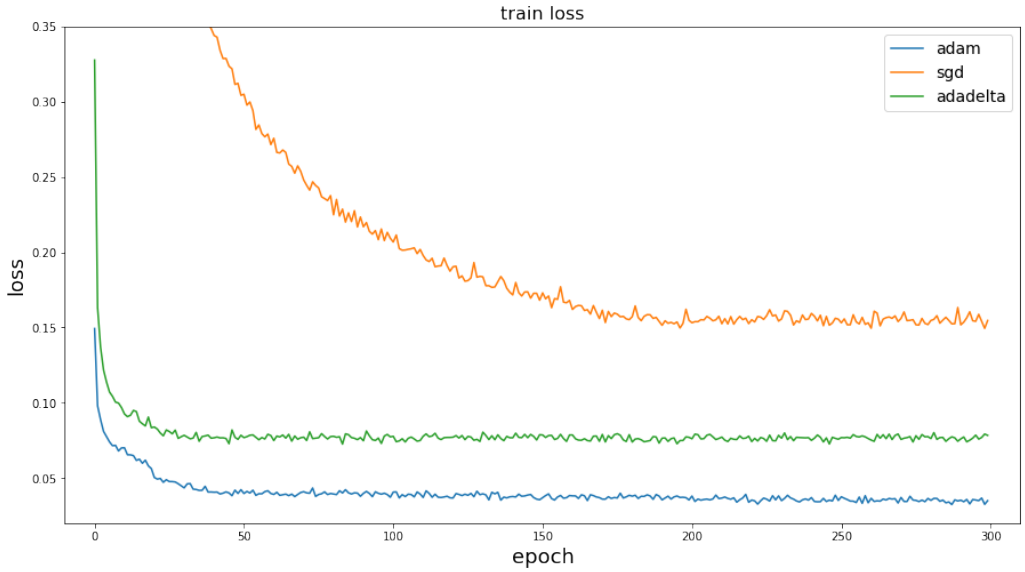


Figure 5: Plot of the loss of the models after 300 epochs in the training dataset

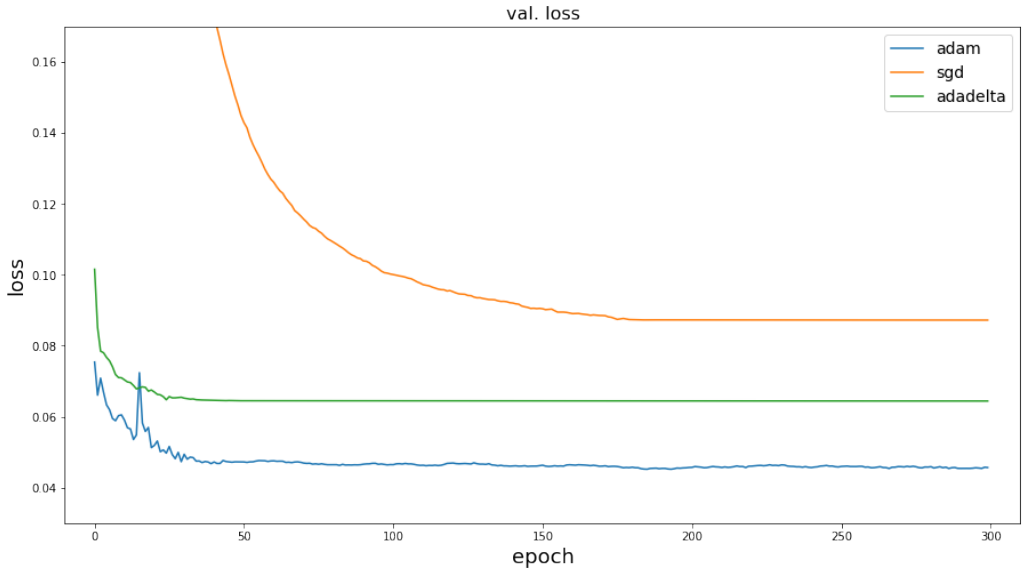


Figure 6: Plot of the loss of the models after 300 epochs in the validation dataset

only have to declare it as *True* when you call for the SGD optimizer. In the case of the momentum, we used a 0.9 as its feature, even when it combines with Nesterov.

As we can see Figure 7 and Figure 8 we have the same parameters for SGD, Adam and Adadelta that we had in previous plots, and now we added the accuracy performance of the SGD optimizer with the different boosters.

In both cases, Adam is still the optimizer that reaches a higher accuracy percentage. But we also notice that all three of the SGD plus booster models are better than the Adadelta model. As we can see, the boosters help the SGD algorithm to reach a high accuracy in the first epochs, and then evens out the curve fairly quickly.

In Figure 8 of the validation accuracy, we see however, that the spikes at the beginning of the line behave similar to the Adam optimizer, and same as Adam, in this plot, it reaches a lower accuracy than in training.

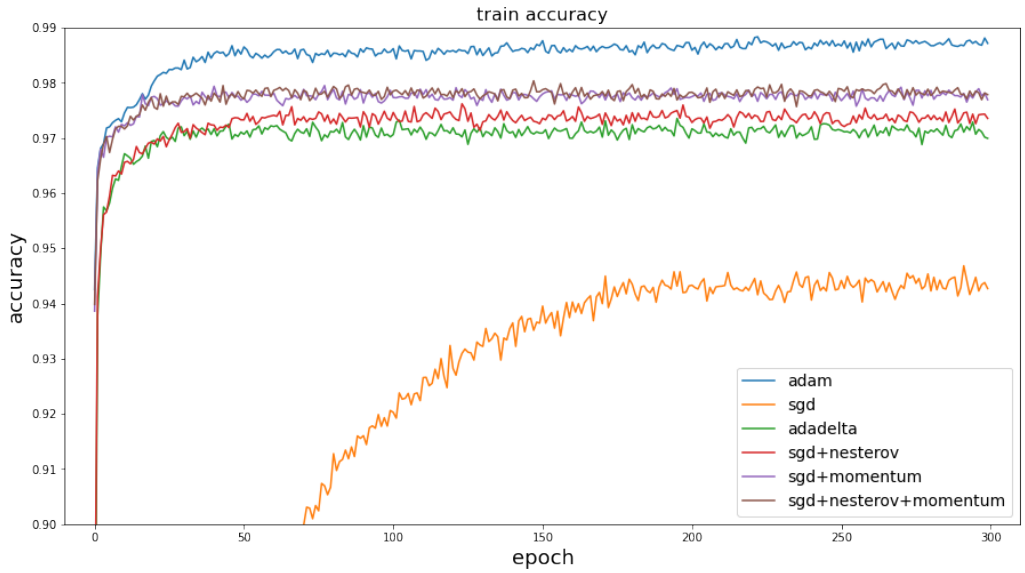


Figure 7: Plot of the accuracy of the models after 300 epochs in the training dataset

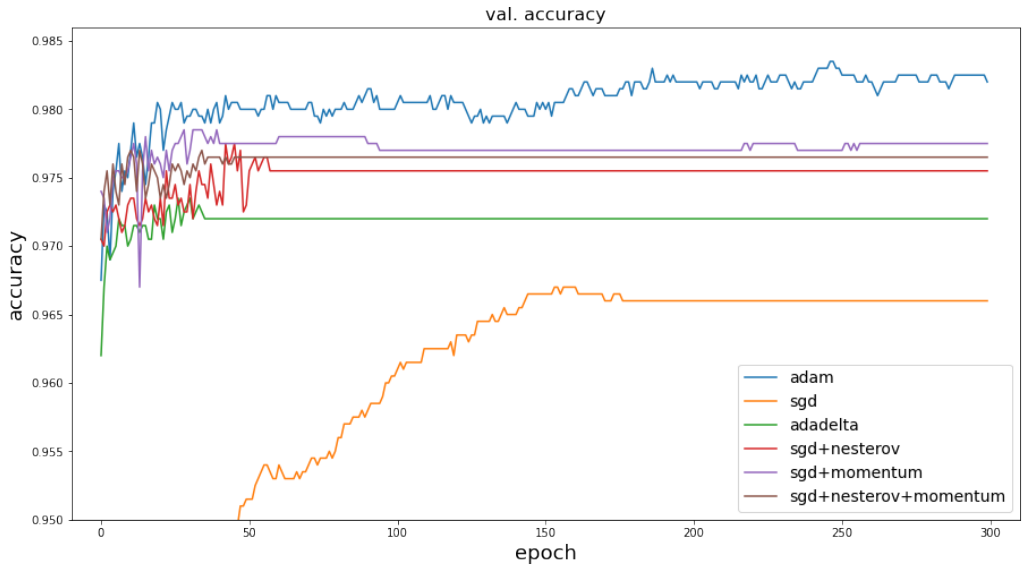


Figure 8: Plot of the accuracy of the models after 300 epochs in the validation dataset

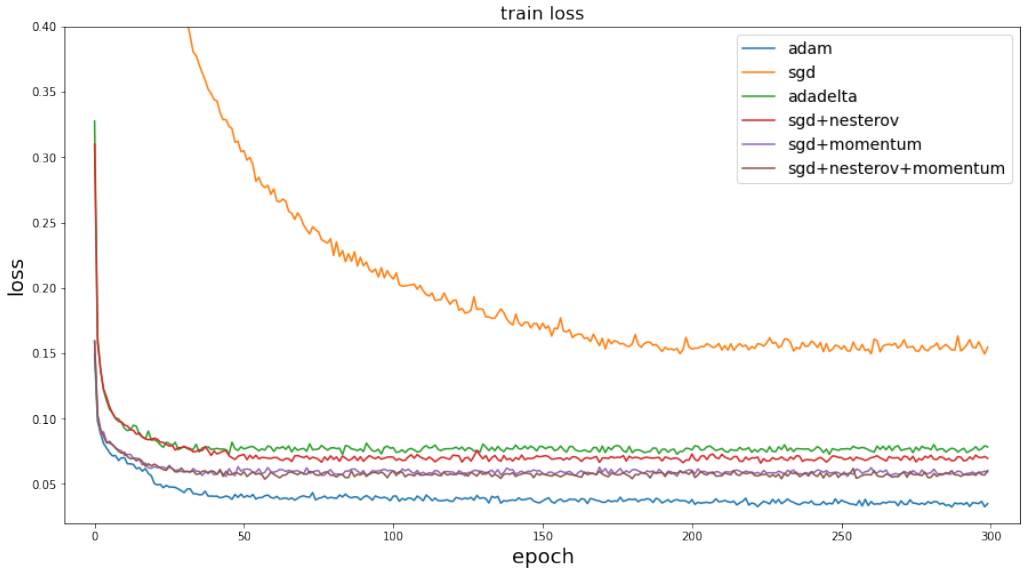


Figure 9: Plot of the loss of the models after 300 epochs in the training dataset

For both loss plots in Figure 9 and Figure 10 we see a clear improvement in the loss rate, but same as with the accuracy plots, Adam keeps being the one with the best results.

5. Conclusions and discussion

We mentioned before that the main goal for this experimentation was to determine if improving the accuracy of our models was worth the computational processing and the time it would take to train more iterations to the CNN architectures we mentioned at the beginning. And the answer to that is no.

Thanks to the Law of large numbers we where able to identify that the average of the SGD optimizer compared to the others is not going to come close to their accuracy percentage in a feasible time.

In the case of SGD by its own, it is not feasible to give it so many iterations to have a result that we achieve with other optimizers easily. More if we take into consideration, that when we see the results when the slope finally evens out, the accuracy does not reach above of 96% of accuracy.

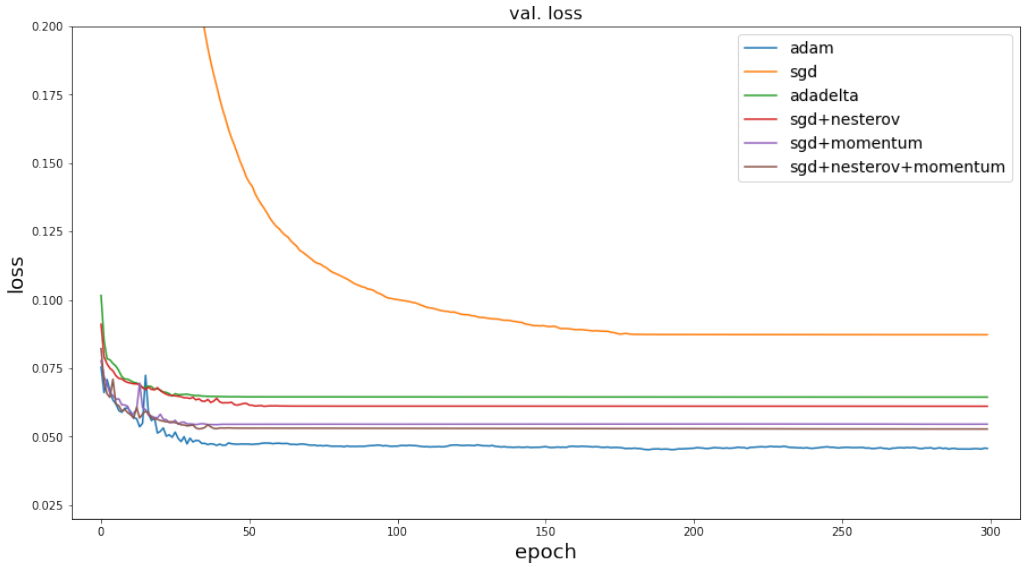


Figure 10: Plot of the loss of the models after 300 epochs in the validation dataset

The boosters definitely helped improve the results of the SGD, but even so, Adam remains a better option than the boosters, given that they take way more time to compile.

The one optimizer that we were considering giving it more iterations because of its behaviors on the plot, was Adam, but when we finally compare the training plot with the validation plot, we see that there is a bit of a gap between the two, where the training performs way better than the validation set.

In the Methods and materials section we discussed the use of a restriction on the iterations, in which the epoch changes when the learning rate gets stuck after several iterations. This was thought out to avoid overfitting, which is what we think is happening with the Adam optimizer. Now, despite the restriction of the learning rate, we see signs of overfitting in the behavior of the training and validation sets of this optimizer, which is why, if we do experiment with only the Adam optimizer with our real images, we will not be using 300 epochs, and instead try at maximum 40 epochs, where we can see the curve of our model even out.

References

[1] What is statistics ans why is important in machine learning (2017).
URL https://machinelearningmastery.com/what-is-statistics/

[2] Machine learning vs statistics (Jan 2016).
URL https://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html

[3] M. C. B. Reyes, Clasificacion de mamografias mediante redes neuronales convolucionales (2019).
URL http://eprints.uanl.mx/17656/

[4] D. P. Kingma, J. A. Ba, A method for stochastic optimization. arxiv 2014, arXiv preprint arXiv:1412.6980 434.

[5] M. D. Zeiler, Adadelta: An adaptive learning rate method. arxiv 2012, arXiv preprint arXiv:1212.5701 1212.

[6] An overview of gradient descent optimization algorithms (2016).
URL https://ruder.io/optimizing-gradient-descent/

[7] Optimizers explained. adam, momentum and sgd (2019).
URL https://mlfromscratch.com/optimizers-explained/#/

[8] A. Botev, G. Lever, D. Barber, Nesterov's accelerated gradient and momentum as approximations to regularised update descent, arXiv preprint arXiv:1607.01981.

[9] Visualising stochastic optimizers (2017).
URL https://rnrahman.com/blog/visualising-stochastic-optimisers/

[10] Law of large numbers (2018).
URL https://www.probabilitycourse.com/chapter7/7_1_1_law_of_large_numbers.php

[11] J. Sirignano, K. Spiliopoulos, Mean field analysis of neural networks: A law of large numbers, SIAM Journal on Applied Mathematics 80 (2) (2020) 725–752.