

# ECS132 Term Project Report

Jonathan Tran

Alex Din

Haosen Cao

June 10, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Analysis</b>	<b>3</b>
2.1	Normal . . . . .	4
2.2	Exponential . . . . .	7
2.3	Gamma . . . . .	12
2.4	Beta . . . . .	13
<b>3</b>	<b>Conclusion</b>	<b>14</b>

# **Chapter 1**

## **Introduction**

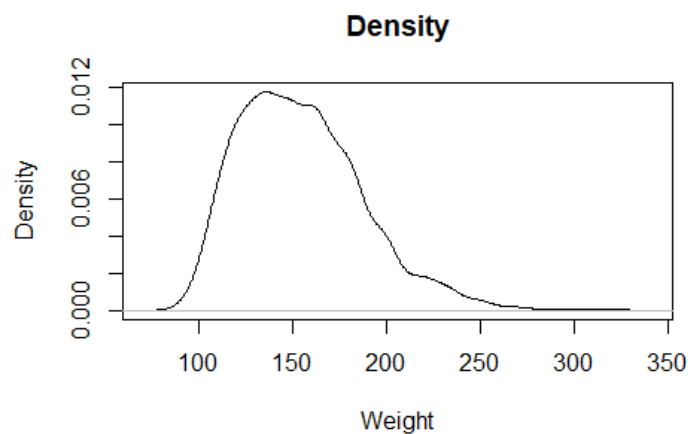
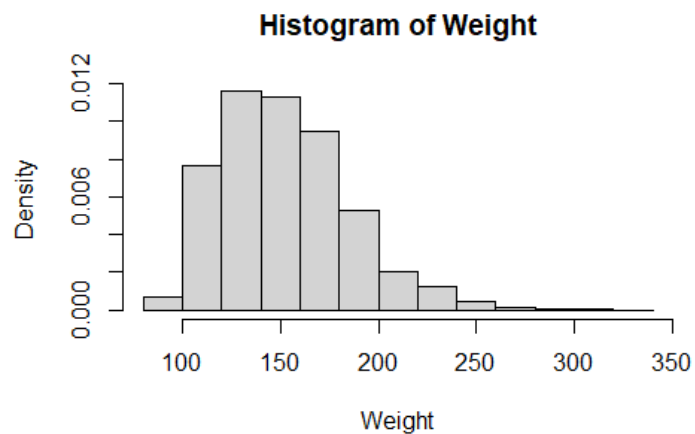
# **Chapter 2**

## **Analysis**

Part 1 content

## 2.1 Normal

To model the normal family of distributions we decided to base our model on the weight column of the national.longitudinal.survey data, which contains data of 4908 individuals' statistics like age, race, height, and weight. The weight is distributed in the range from 87 to 325. When we plotted the histogram, most of the data are in the range from 100 to 200, and as we go to the left or right of that, the number of points decreases. But most importantly the data looked similar to a bell curve.



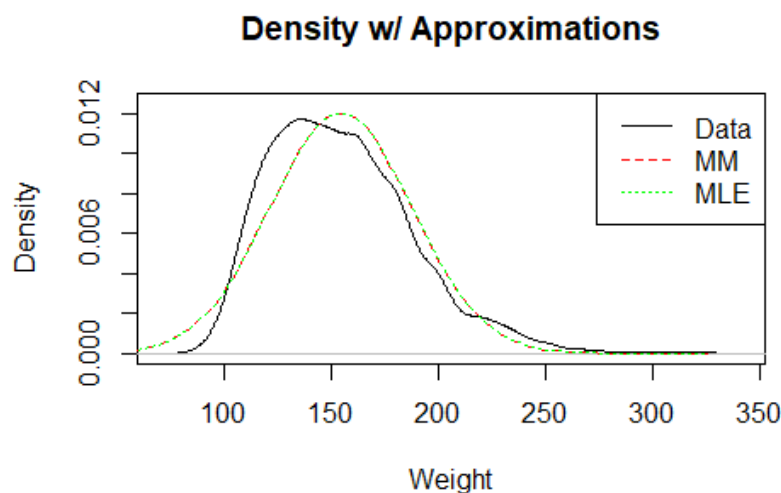
The normal family of distributions has two parameters, mean and standard deviation. We are assuming that the weights in this dataset can be modeled with a normal distribution. Therefore we use two methods to find estimates for the mean and standard deviation parameters, which we will call  $M$  and  $SD$ , respectively.

For maximum likelihood, we used R's `mle()` function to find  $M$  and  $SD$ , which takes a negative log-likelihood function. While the likelihood function is the product of density functions on the given data, the log-likelihood is the sum of the logarithm of those density functions. Therefore we use this line as our negative log-likelihood function:

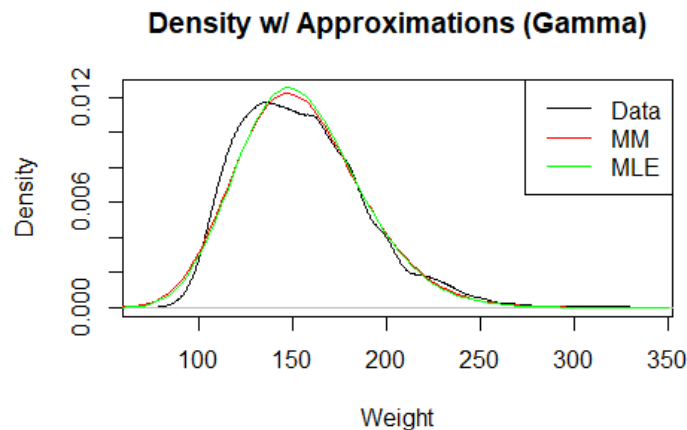
```
loglik <- sum(dnorm(weights, M, SD, log = TRUE))
```

Where `weights` is our weight data, and  $M$  and  $SD$  are our parameter estimates. Plugging this into the `mle()` function gives us  $M = 154.5827$  and  $SD = 33.18112$ .

The method of moments for a normal distribution is relatively simple. The two parameters of the normal distribution are the mean and standard deviation, which we can generate estimates for using our sample data directly from R.  $M$  is just  $\bar{A}$ , our sample mean from calling `mean(weights)`, and  $SD$  is just  $\sqrt{S^2}$  which can be given by calling `sqrt(var(weights))`. The difference between using  $S^2$  and  $s^2$  is negligible. Using this method, we get  $M = 154.5827$  and  $SD = 33.18453$ .



We would say that the normal family is a suitable estimator for the weights in this dataset. A normal distribution, according to the central limit theorem, is what results from a set of summations of random variables. The weight of a person can be thought of as a sum of the various parts of their body, whose sizes can be thought of as random. However, based on the graph, which has a slightly steeper slope on the left side than on the right, it can be argued that a gamma distribution would better model this data.

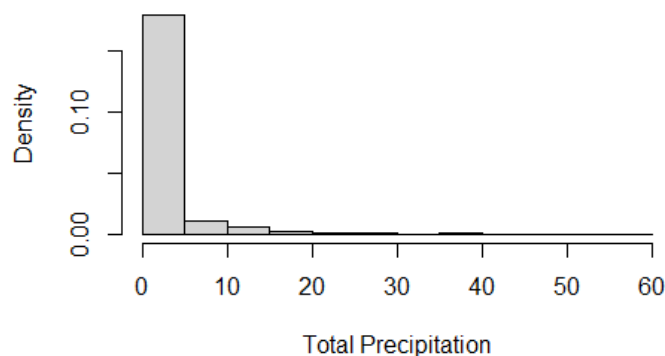


And from the graph, we must concede that the gamma family may model this data better than the normal family. But if the central limit theorem says that sums of random variables look normal, and the gamma distribution is a sum of exponentially distributed random variables, it's no wonder that the graphs look pretty similar. If we're summing a large enough number of exponential variables, it's possible that the gamma distribution could converge to a normal distribution. From this line of thinking, the normal and gamma distribution are likely both good estimators for the weights in the dataset.

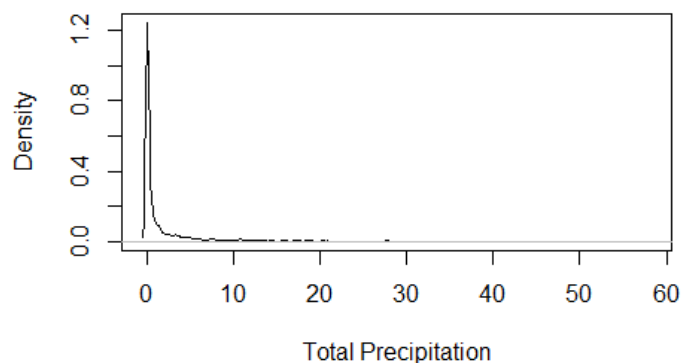
## 2.2 Exponential

To model the exponential family of distributions we decided to base our model on the PRECTOT column of the weatherTS data, which looks like it took weather data from Eastern Australia for 10 years (1985–1995). Of this data, the one we were most interested in was the PRECTOT column, which we interpreted as the total precipitation on a given day. When we plotted the histogram, the data was heavily skewed left, tapering off more to the right, which resembled that of an exponential graph. The most important thing was that the peak of the data was at  $x = 0$ , which is where exponential graphs peak.

**Default Histogram**



**Default Density**





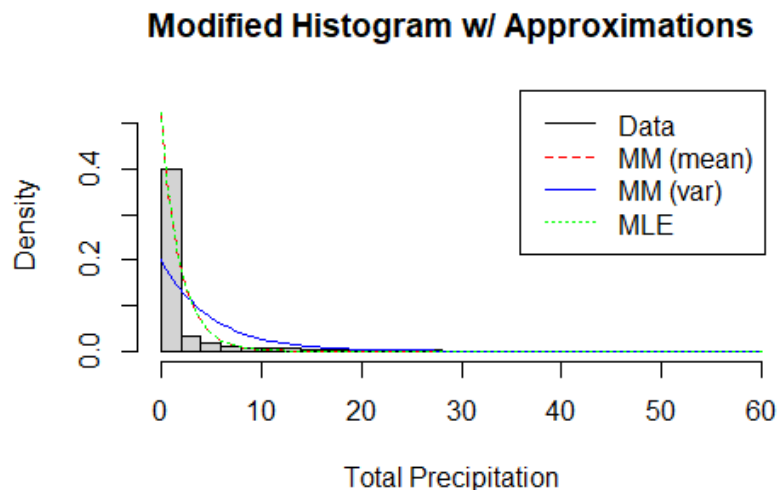
Just from looking at these graphs, while the density function doesn't look quite exponential, the histogram definitely does. We will discuss possible reasons why later on. But for now, we will find estimators for the parameters of the exponential family. The exponential family is a one-parameter family, that being lambda. Using the two methods, we are trying to find an estimator for lambda, which will be referred to as  $L$ . We use this line of code to get the negative log-likelihood function for our maximum likelihood.

```
loglik <- -sum(dexp(pectot, L, log = TRUE))
```

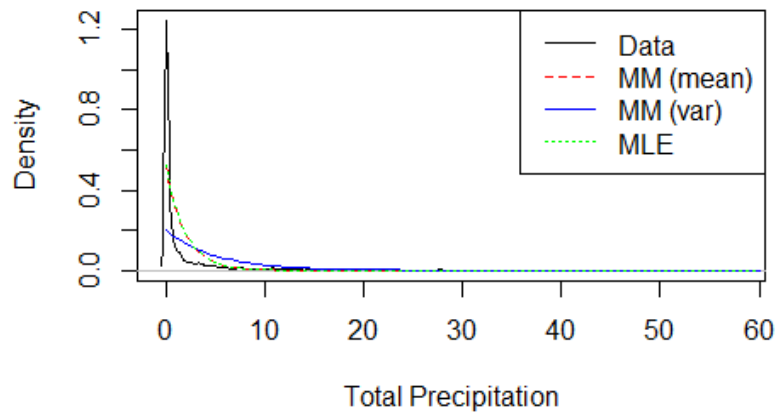
Where pectot is our precipitation data and  $L$  is our estimator for lambda. Plugging this into the `mle()` function we get  $L$  to be 0.5230325.

Next, for method of moments, we algebra to estimate  $L$ . For a given exponentially distributed variable  $X$ , the expected value is given by  $E(X) = \frac{1}{\lambda}$ . Assuming the mean of our sample data,  $\bar{A}$ , is an unbiased estimator for the true mean of the population data,  $\bar{A} = \frac{1}{L}$ , and therefore  $L = \frac{1}{\bar{A}}$ . Plugging this into R we get  $L$  to be 0.5230367.

Because this is a one-parameter family, we can also derive  $L$  from our sample variance  $s^2$ . The variance of an exponentially distributed variable  $X$  is given by  $Var(X) = 1/\lambda^2$ . Therefore  $L$  is given by  $L = \sqrt{1/s^2}$ . With this method,  $L$  is shown to be 0.2002137, which is very different from the other two methods above. From the graphs below, this particular estimate for lambda isn't very good.

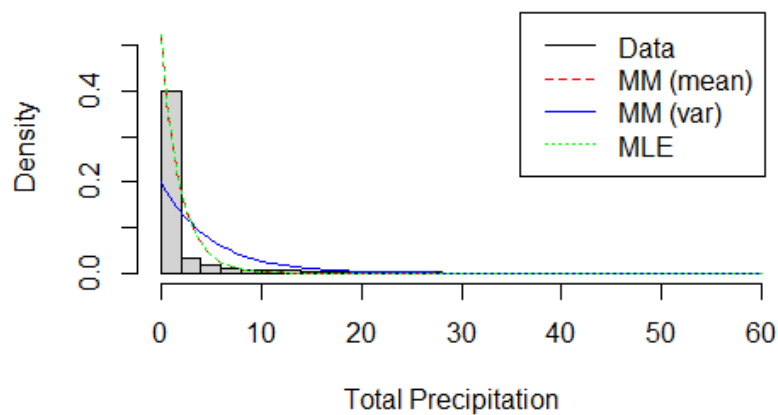


### Default Density w/ Approximations

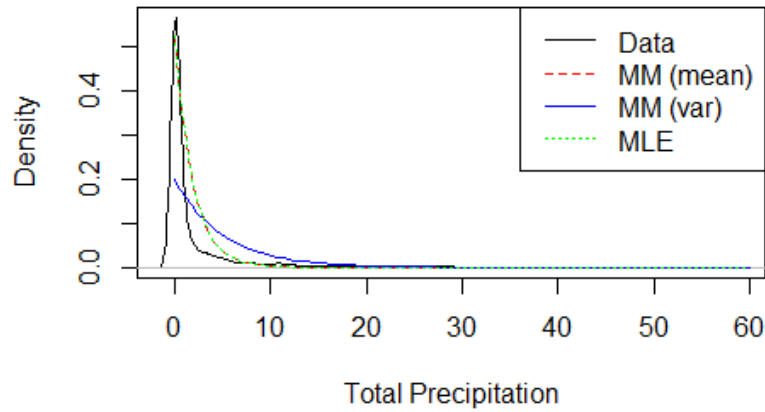


It is important to note that our exponential distribution plots don't quite match those of the `hist()` or `density()` functions on our sample data, at least with their default arguments. These discrepancies can be attributed to a bandwidth that doesn't accurately reflect our sample. By decreasing the bandwidth for the `density()` function and increasing it for the `density()` function, we get graphs that better match our plots.

### Modified Histogram w/ Approximations

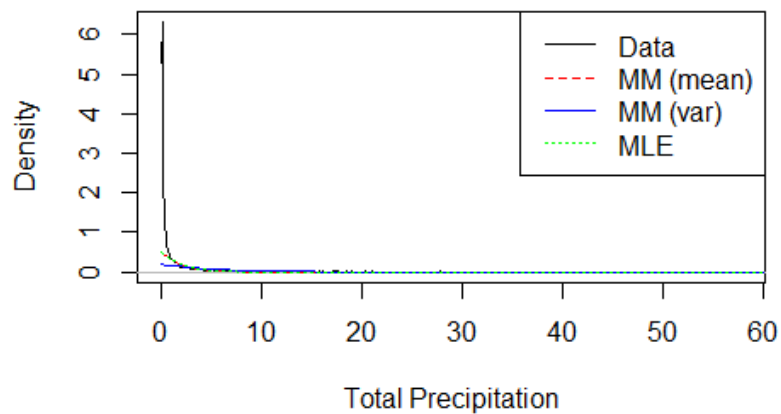


### Modified Density w/ Approximations



The histogram looks good, but the density function is arguably less exponential-looking as a result. Lowering the bandwidth of the density function does make it more exponential, but likely at the cost of being less accurate due to how few points can actually fit into an individual bandwidth.

### Small BW Density w/ Approximations



But even with these discrepancies, given our plots of the exponential distribution, we would say this family is a suitable estimator for the total precipitation on a given day in Australia. The exponential distribution is said to be the continuous analog of the geometric distribution. How much precipitation you get in the day

is pretty much the same as how long it rains on a given day. And how long it rains in a given day can be thought of as successive failures (the rain continuing) until a single success (the rain stopping), which is like a geometric distribution. Obviously, this is an oversimplification but it shows how an exponential distribution can model the density function of total precipitation.

## **2.3 Gamma**

## **2.4 Beta**

## **Chapter 3**

## **Conclusion**