Tutorial4

2023-09-03

Moment, Measure of Central Tendencies

Moments and Mean

Mean is the first moment(m = 0). General form of r-th moment is given by:

$$E[X^r] = \int_{-\infty}^{+\infty} x^r f_X(x) dx$$

in the case of continuous distributions. When r = 0, what we have is just the integral of the probability function, which would always result in 1.

$$E[X^0] = \int_{-\infty}^{+\infty} x^0 f_X(x) dx = \int_{-\infty}^{+\infty} f_X(x) dx = 1$$

which is something we have learned in probability theory, that the probability of all possible events should add up to 1. Now, let's consider the first moment.

$$E[X^0] = \int_{-\infty}^{+\infty} x^0 f_X(x) dx = \int_{-\infty}^{+\infty} x f_X(x) dx = \mu$$

But, what is this thing? Why is this called as mean? Let's descretize it, let's re write the integral! :) Let's assume that, the distribution can be effectively confined between -T and T, as any numbers larger than that would have a negligible probability. Let's divide this space into N intervals, and the set of all such intervals be denoted by I. Now, we have a probability for an item belonging to each set, which we can call as the probability of that set, p(i), where I be the value for that particular interval.

$$E[X] = \int_{-T}^{+T} x f_X(x) dx = \sum_{i \in I} i p(i) = \mu$$

We can effectively consider the probabilities as weights, and thus this sum would become a weighted average. But wait, something is WRONG! have you noticed it?

```
library(tidyr)
library(dplyr)
```

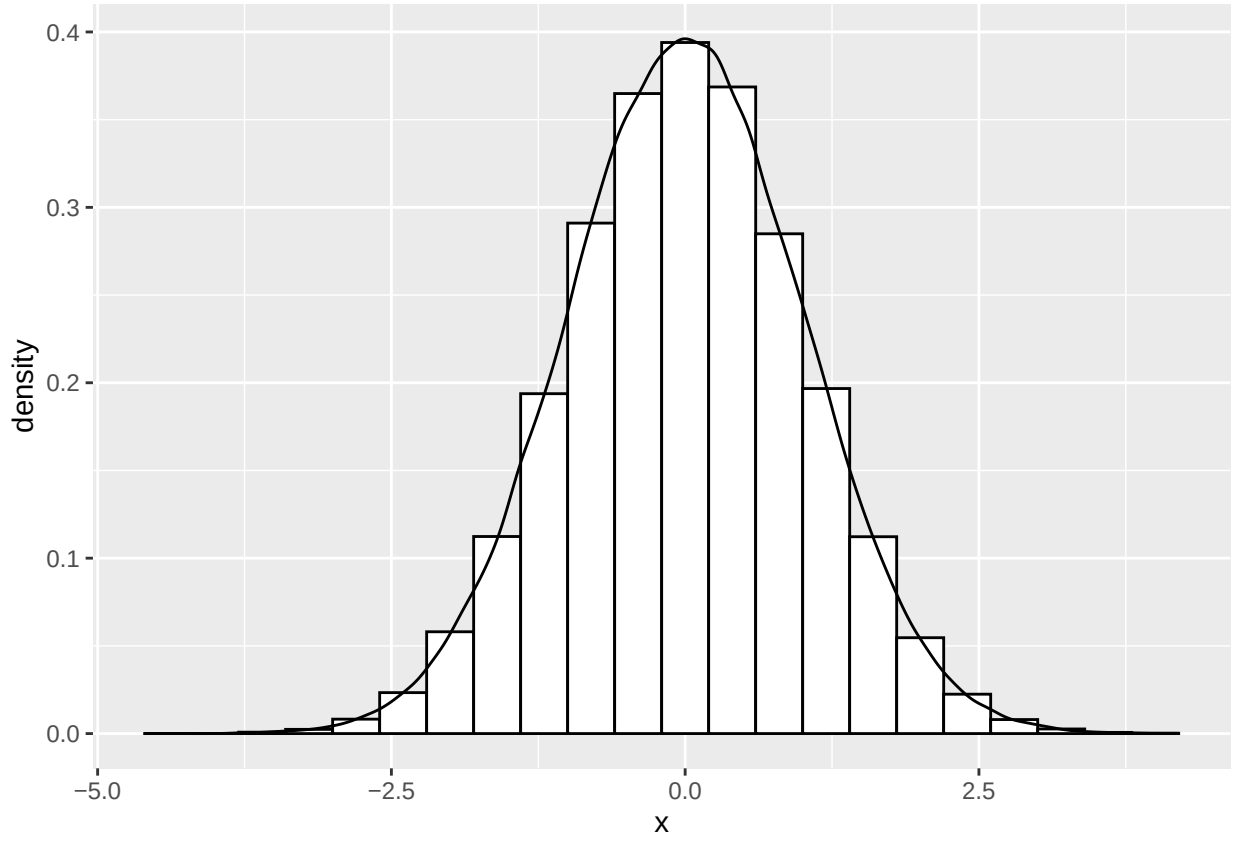
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

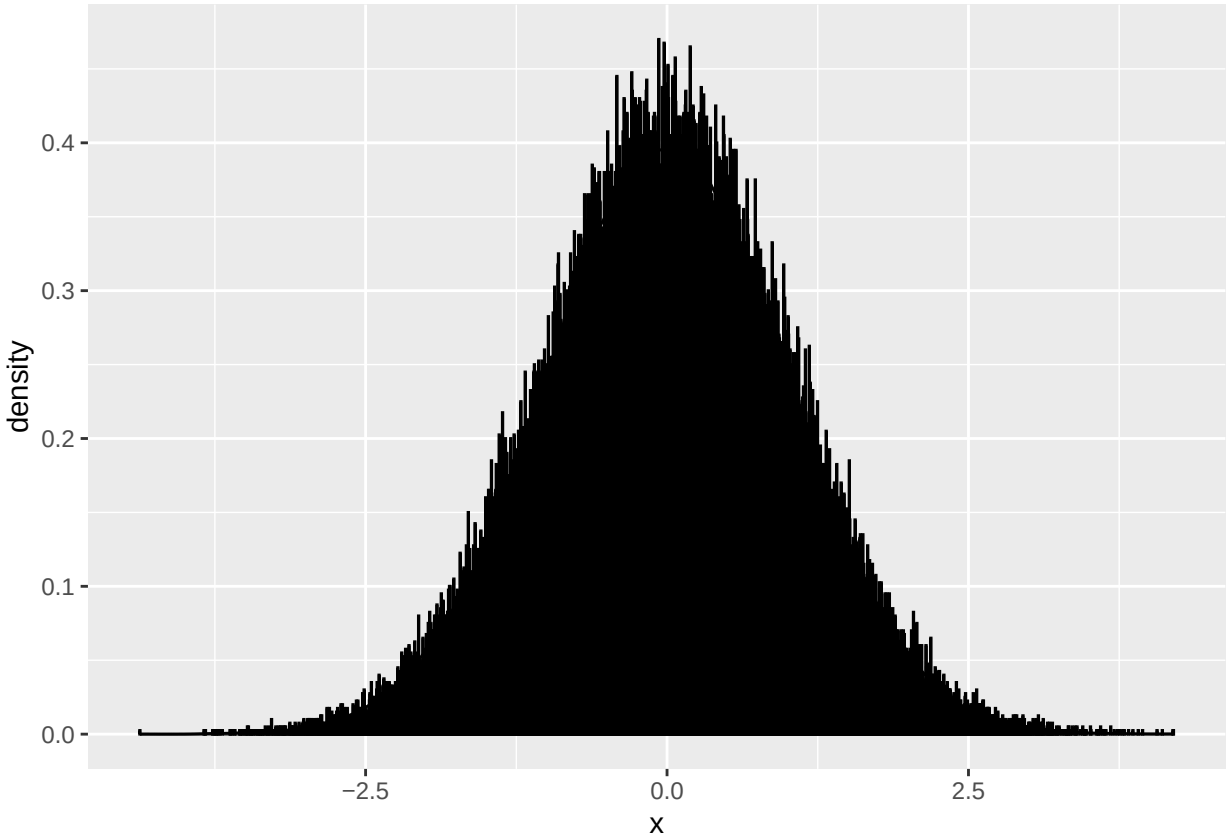
```
library(ggplot2)

NorDist <- rnorm(100000, mean = 0, 1)
data <- data.frame(x = NorDist)
ggplot(data, aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white", binwidth = 0.4) +
  geom_density()
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



We can clearly see that the approximation, that is the bins are not exacly capturing the actual distribtuion. What if we are making the bin size 1000 times smaller?

```
ggplot(data, aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white", binwidth = 0.004) +
  geom_density()
```



It better approximates with the actual distribution. Mathematically, when we limit the size of each interval towards zero, we would get better estimates.

$$E[X] = \int_{-T}^{+T} x f_X(x) dx = \lim_{N \to \infty} \sum_{i \in I} ip(i) = \mu$$

or if we are considering size of such intervals, then it can be written as:

$$E[X] = \int_{-T}^{+T} x f_X(x) dx = \lim_{\Delta x \to 0} \sum_{i \in I} ip(i) = \mu$$

So, integral is another version of sum! Now, let's make it more intuitive. We have seen that we can imagine the integral as a sum. But, why is it called a moment? It is analogous to the concept of center of mass?

Central Moments and variance

Now, let's consider the expectation of the deviation from the first moment, given by:

$$E[(X - \mu)^r] = \int_{-\infty}^{+\infty} (x - \mu)^r f_X(x) dx$$

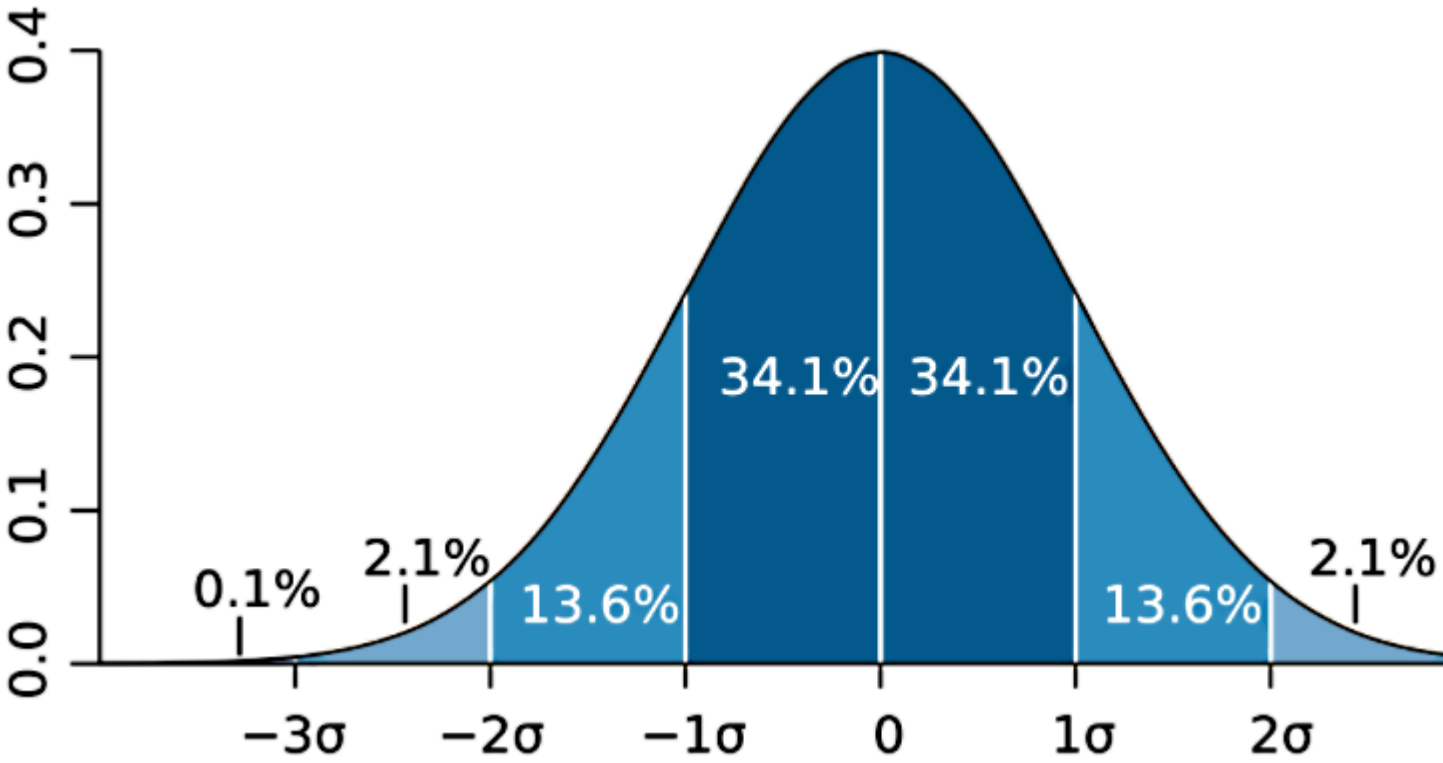
Let's consider the case where r = 2.

$$E[(X - \mu)^2] = \int_{-\infty}^{+\infty} (x - \mu)^2 f_X(x) dx$$

Let's only consider the LHS

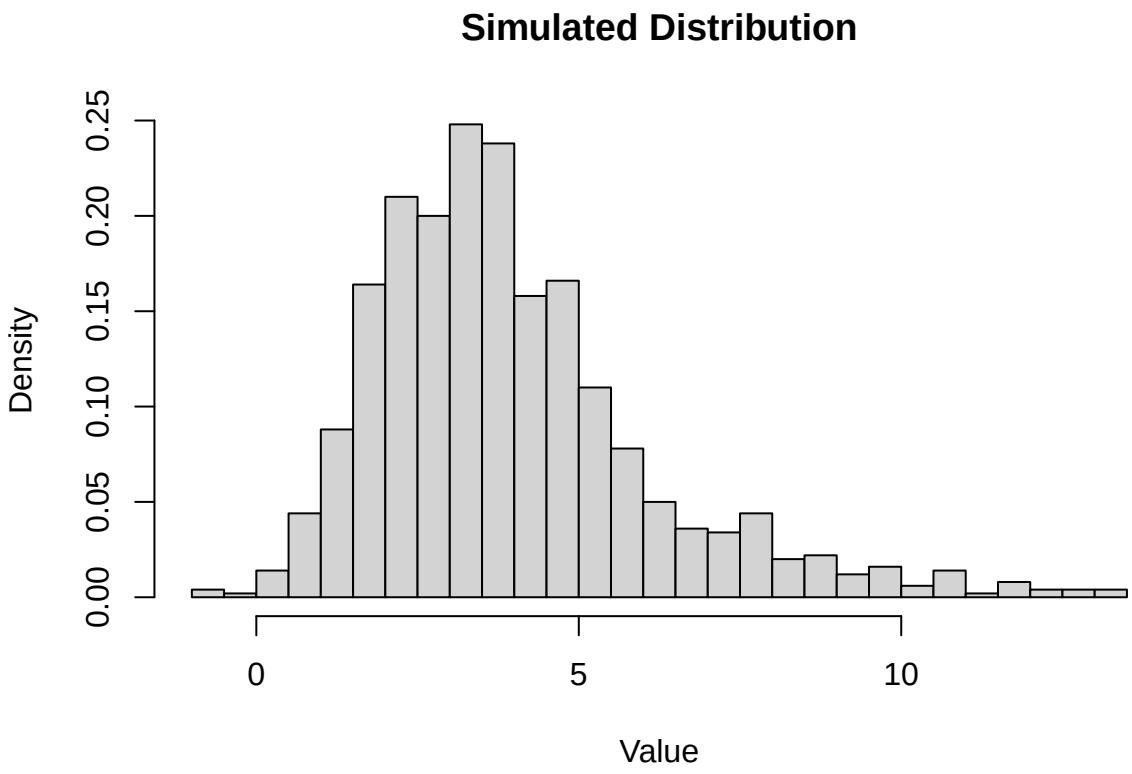
$$E[(X-\mu)^2] = E[X^2 - 2\mu X + \mu^2] = E[X^2] - 2\mu^2 + \mu^2 = E[X^2] - \mu^2$$

In r, function "mean()" gives you the first moment and "sd()" gives you the square root of second central moment. This measure called standard deviation is used for quantifying the spread of the data.



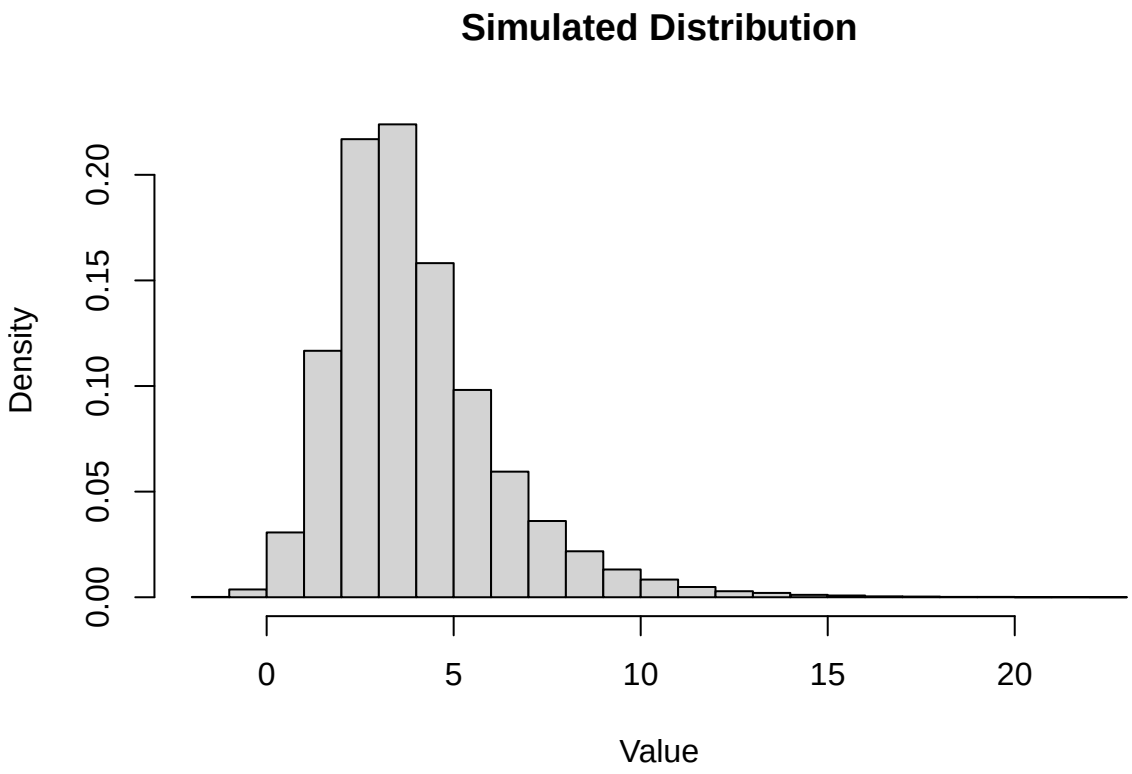
Let's consider a distribution, as specified below:

```r
generate_exgaussian_data <- function(n, mu, sigma, lambda) {
  # Generate random data from the Ex-Gaussian distribution
  gaussian_samples <- rnorm(n, mean = mu, sd = sigma)
  exponential_samples <- rexp(n, rate = lambda)

  exgaussian_samples <- gaussian_samples + exponential_samples

  return(exgaussian_samples)
}

# Usage:
set.seed(42)  # Set seed for reproducibility
n <- 1000
mu <- 2.0
sigma <- 1.0
lambda <- 0.5

exgaussian_data <- generate_exgaussian_data(n, mu, sigma, lambda)

# Plot the histogram of the simulated Ex-Gaussian data
hist(exgaussian_data, breaks = 30, probability = TRUE,
     main = "Simulated Distribution", xlab = "Value")
```



now, let's increase n and see what would happen

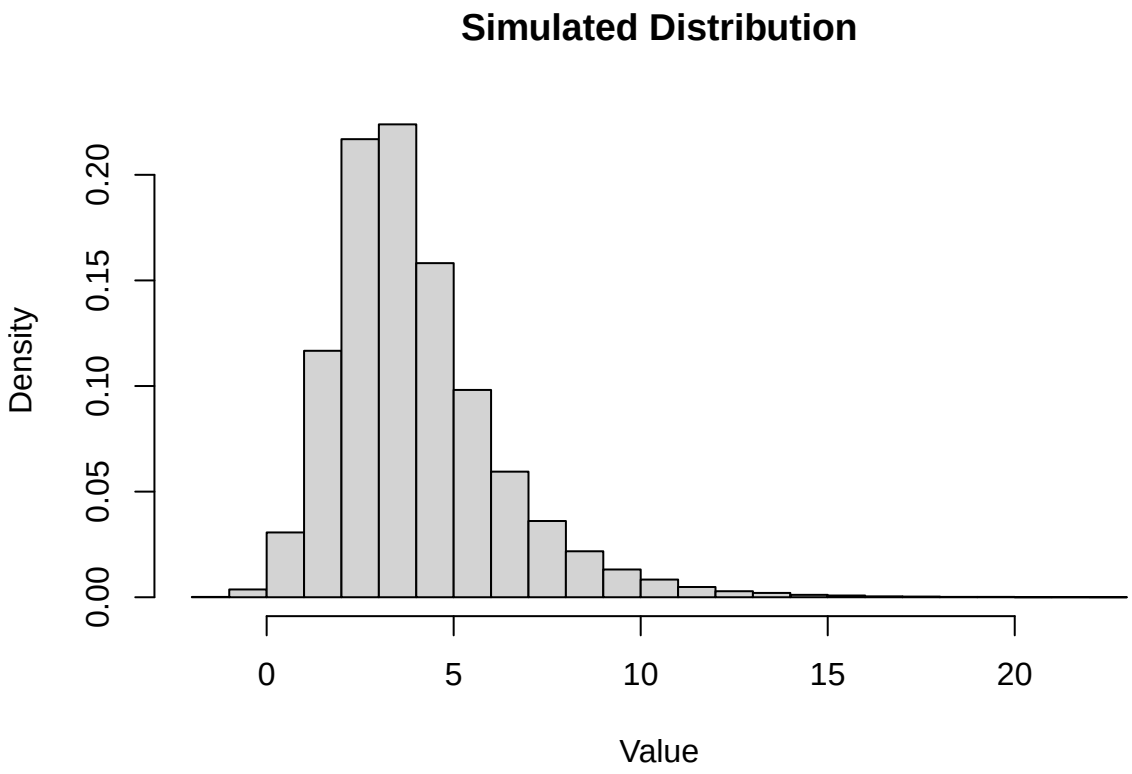
```
set.seed(42)  # Set seed for reproducibility
n <- 100000
mu <- 2.0
sigma <- 1.0
lambda <- 0.5

exgaussian_data <- generate_exgaussian_data(n, mu, sigma, lambda)

# Plot the histogram of the simulated Ex-Gaussian data
hist(exgaussian_data, breaks = 30, probability = TRUE,
     main = "Simulated Distribution", xlab = "Value")
```



Skewness

We can clearly see that the distribution has an elongated tail. Skewness is a measure, that gives us an idea about how much the distribution is deviating from normality. It tells about the position of the majority of data values in the distribution around the mean value. A fundamental statistical notion called skewness quantifies the asymmetries in data distributions. For this we can use moments library

```
library(moments)

set.seed(42)  # Set seed for reproducibility
n <- 100000
mu <- 2.0
sigma <- 1.0
lambda <- 0.5

exgaussian_data <- generate_exgaussian_data(n, mu, sigma, lambda)

# Plot the histogram of the simulated Ex-Gaussian data
hist(exgaussian_data, breaks = 30, probability = TRUE,
     main = "Simulated Distribution", xlab = "Value")
```




```
print(skewness(exgaussian_data))
```

```
## [1] 1.451948
```

But, what does this value mean? The asymmetry of data distributions where the tail extends towards higher values is known statistically as positive skewness. If the coefficient of skewness is greater than 0 i.e. $\gamma_1 > 0$, then the graph is said to be positively skewed with the majority of data values less than the mean. Most of the values are concentrated on the left side of the graph.

Kurtosis

A statistical measure known as kurtosis measures the flatness, and weight of the tails of data distributions

There exist 3 types of Kurtosis values on the basis of which the sharpness of the peak is measured. These are as follows: * Platykurtic

Data distributions having flattened tails compared to the normal distribution are referred to statistically as platykurtic distributions.

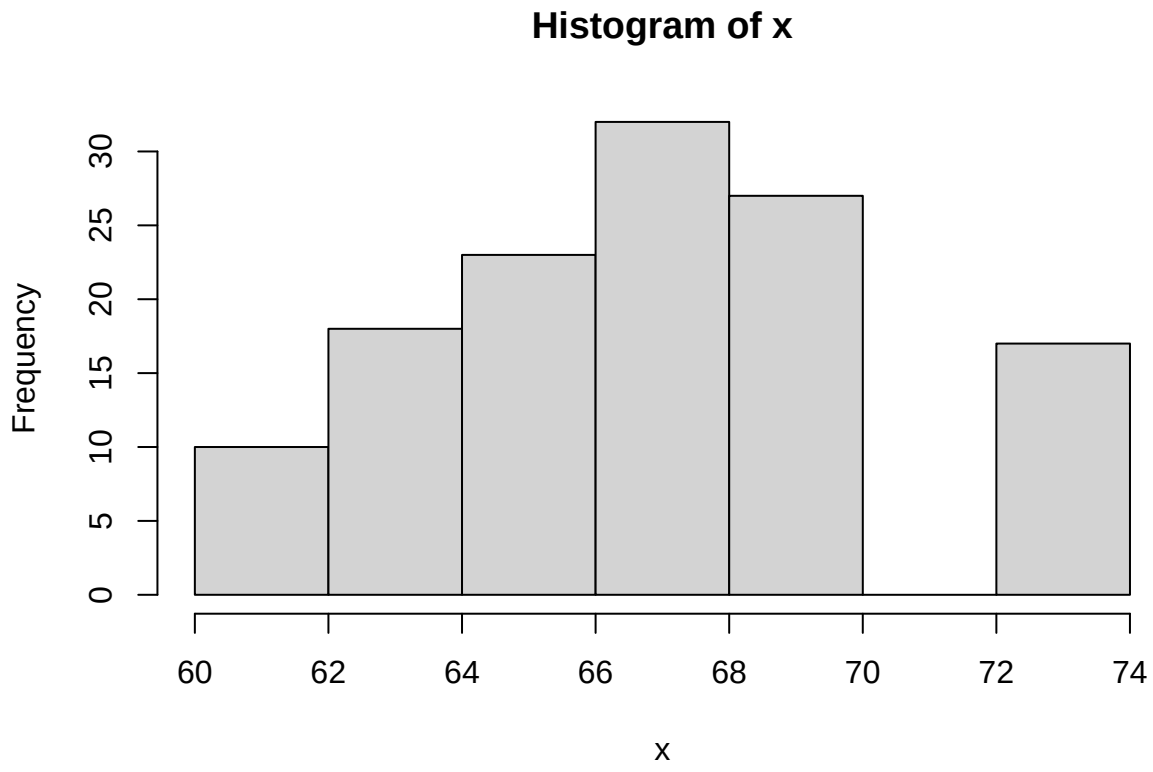
If the coefficient of kurtosis is less than 3 i.e. $\gamma_2 < 3$, then the data distribution is platykurtic. Being platykurtic doesn't mean that the graph is flat-topped.

```
x <- c(rep(61, each = 10), rep(64, each = 18),
rep(65, each = 23), rep(67, each = 32), rep(70, each = 27),
rep(73, each = 17))
```

```r
print(kurtosis(x))
```

```
## [1] 2.258318
```

```r
hist(x)
```



* Mesokurtic

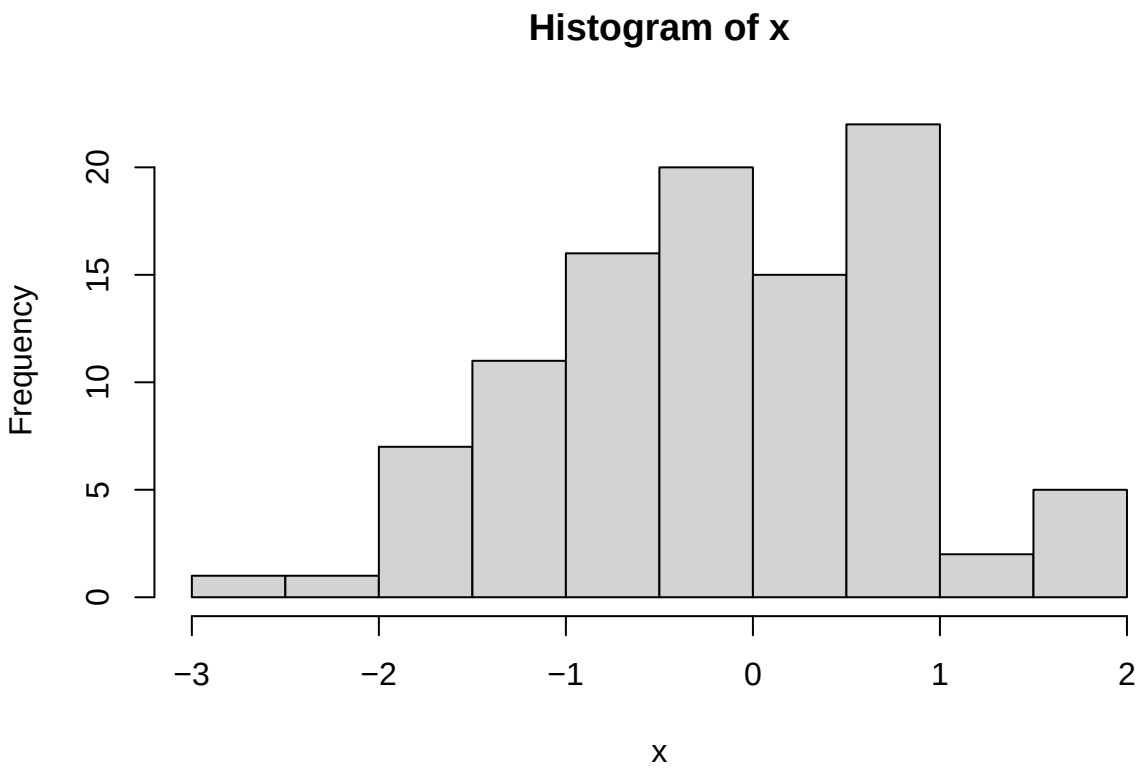
Data distributions with tails that are similar in thickness to the normal distribution are known statistically as mesokurtic distributions.

If the coefficient of kurtosis is equal to 3 or approximately close to 3 i.e. $\gamma_2 = 3$, then the data distribution is mesokurtic. For the normal distribution, the kurtosis value is approximately equal to 3

```r
x <- rnorm(100)
print(kurtosis(x))
```

```
## [1] 2.451632
```

```r
hist(x)
```



- Leptokurtic

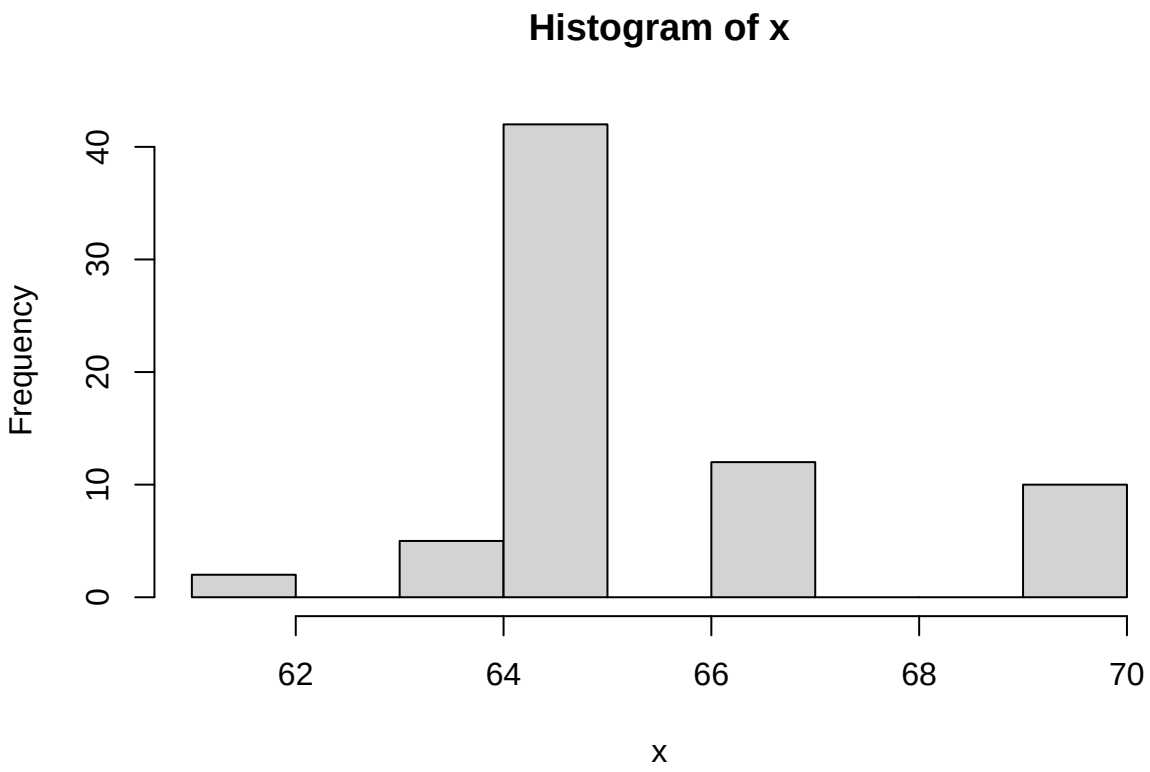
Data distributions having hefty tails compared to the normal distribution are referred to statistically as leptokurtic distributions.

If the coefficient of kurtosis is greater than 3 i.e. $\gamma_2 > 3$, then the data distribution is leptokurtic and shows a sharp peak on the graph.

```
x <- c(rep(61, each = 2), rep(64, each = 5),
rep(65, each = 42), rep(67, each = 12), rep(70, each = 10))
print(kurtosis(x))
```

```
## [1] 3.696788
```

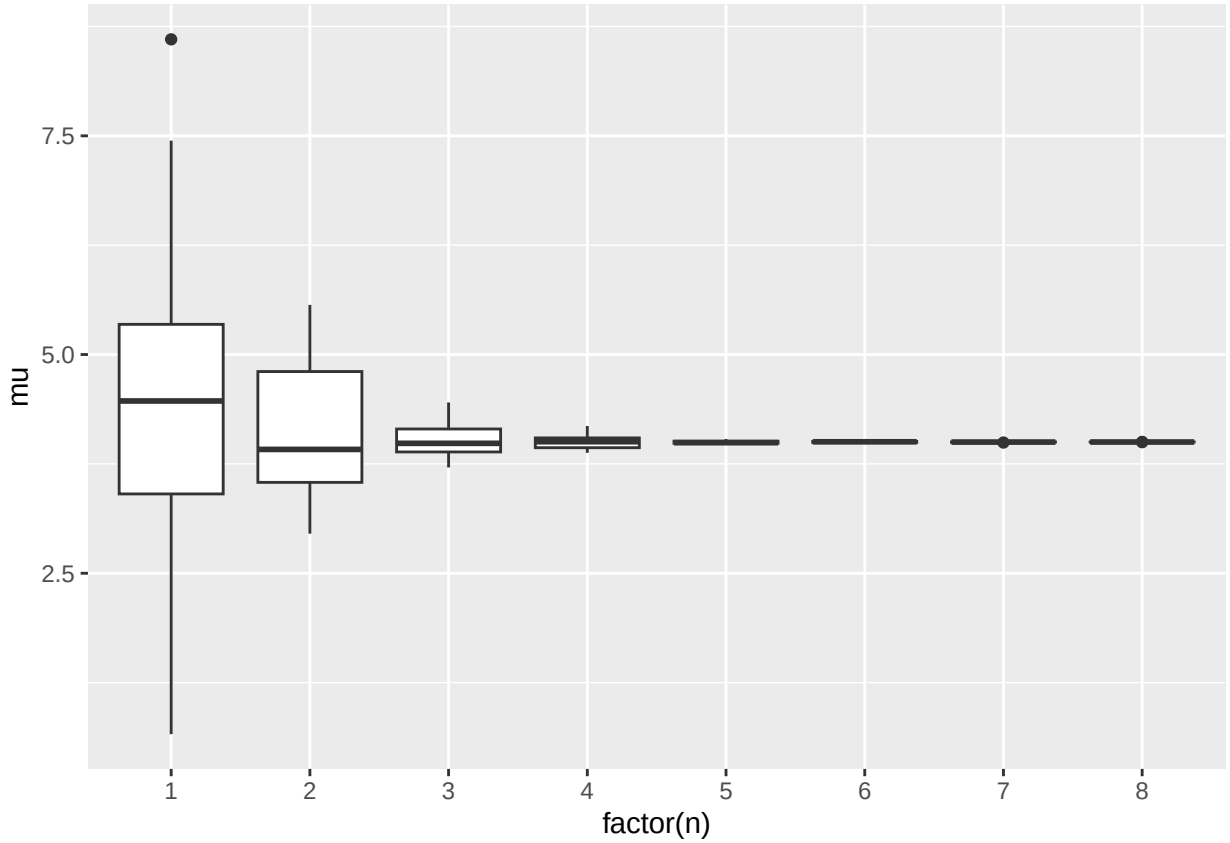
```
hist(x)
```



LLN and CLT

So, if we are better approximating the distribution, how many samples we take, the sample means would have less variations. Is that what we can see? Let's consider the case of mean.

```
data <- data.frame(mu = c(), n = c(), sample = c())
mu <- 2.0
sigma <- 1.0
lambda <- 0.5
for (n_exp in 1:8){
  for(rep in 1:30){
    n <- 10^(n_exp - 1)
    exgaussian_data <- generate_exgaussian_data(n, mu, sigma, lambda)
    df <- data.frame(mu = mean(exgaussian_data))
    df['n'] <- n_exp
    df['sample'] <- rep
    data <- rbind(data, df)
  }
}
my_plot <- ggplot(data)
my_plot+ geom_boxplot(aes(factor(n), y = mu))
```



That shows that the overall variability in the distribution becomes low as we increases the number of draws. We can clearly see that, given the data generating process is the same, as we increases the draws, we are better approximating the distribution to the "true" distribution, which eliminates the variations. Which is the essence of law of large numbers, larger the number of samples we are taking, better would be the approximation to the underlying distribution. Here, the distributions we are looking at are called sampling distributions of mean. Now, let's see the implications of CLT. We know that the sample of means would be normally distributed. But, let's see the standard error of mean as a function of both n and sample size.

```
#data <- data.frame(mu = c(), n = c(), sample = c())
#mu <- 2.0
#sigma <- 1.0
#lambda <- 0.5
#SE <- c()
#REP <- c()
#N<- c()
#for (n_exp in 1:8){
  #for (reps in 1:5){
  #curr_rep = reps*10
  #MEANS <- c()
  #for(rep in 1:curr_rep){
   # n <- 10^(n_exp - 1)
    #exgaussian_data <- generate_exgaussian_data(n, mu, sigma, lambda)
    #MEANS[length(MEANS)+1] <- mean(exgaussian_data)

  #}
  #SE[length(SE)+1]<- sd(MEANS)
```

```
  #REP[length(REP)+1] <- curr_rep
  #N[length(N)+1] <- n_exp
# }
#}
#data <- data.frame( SEM = SE, sample_size = REP, log_draws = N)
#my_plot <- ggplot(data)
#my_plot+ geom_point(aes(x=log_draws, y = sample_size, color = SEM))
```