2nd Delivery

Eduardo Alarcón & Alfonso Pineda

2022-12-03

```r
library(readxl)
SpotifySongs <- read_excel("songstats.xlsx")
```

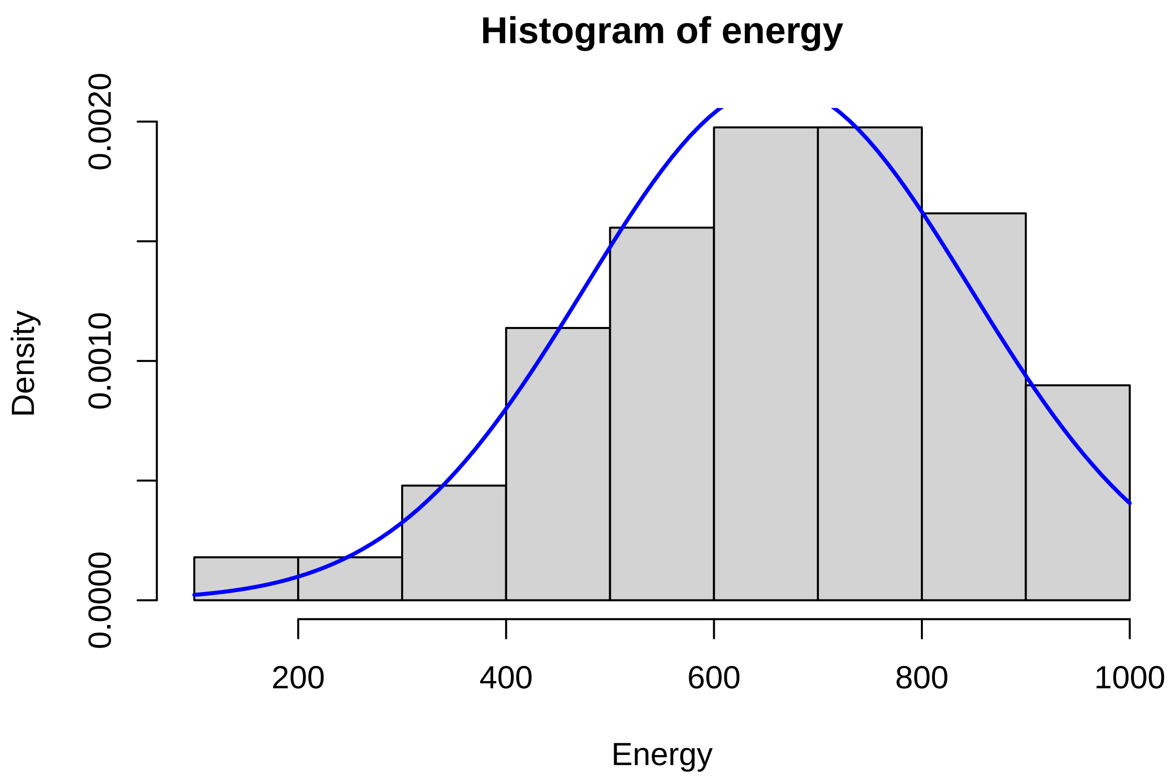
Model Fitting.

By just watching the histogram, we can suppose that our main variable, the energy, will follow a normal distribution

```r
SpotifySongs <- read_excel("songstats.xlsx")
suppressWarnings(library(summarytools))
energy <- SpotifySongs$energy
descr(energy)
```

```
## Descriptive Statistics
## energy
## N: 167
##
##                       energy
## ----------------- --------
##              Mean   660.92
##           Std.Dev   185.68
##               Min   104.00
##                Q1   551.00
##            Median   667.00
##                Q3   804.00
##               Max   993.00
##               MAD   188.29
##               IQR   249.50
##                CV     0.28
##          Skewness    -0.44
##       SE.Skewness     0.19
##          Kurtosis     0.07
##           N.Valid   167.00
##         Pct.Valid   100.00
```

```r
hist(energy, probability = TRUE, xlab = "Energy")
curve(dnorm(x, mean(energy), sd(energy)), col="blue", lwd=2, add=TRUE, yaxt="n")
```




```
Partition <- hist(energy, plot=FALSE)
Partition
```

```
## $breaks
##  [1]  100  200  300  400  500  600  700  800  900 1000
##
## $counts
## [1]  3  3  8 19 26 33 33 27 15
##
## $density
## [1] 0.0001796407 0.0001796407 0.0004790419 0.0011377246 0.0015568862
## [6] 0.0019760479 0.0019760479 0.0016167665 0.0008982036
##
## $mids
## [1] 150 250 350 450 550 650 750 850 950
##
## $xname
## [1] "energy"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

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

```r
normalfit <- fitdist(energy, "norm")
normalfit
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##      estimate Std. Error
## mean 660.9222   14.32480
## sd   185.1206   10.12919
```

```r
CummulativeProbabilities = pnorm(c(-Inf, Partition$breaks[c(-1, -10)], Inf),
                                 normalfit$estimate[1], normalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(energy)*Probabilities
chisq.test(Partition$counts, p=Probabilities)
```

```
## Warning in chisq.test(Partition$counts, p = Probabilities): Chi-squared
## approximation may be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 5.9441, df = 8, p-value = 0.6535
```

```r
pchisq(5.9441, 6, lower.tail = FALSE)
```

```
## [1] 0.4294811
```

```r
pchisq(99, 90, lower.tail=FALSE)
```

```
## [1] 0.2422774
```

As we can see from the previous test, the normal distribution model can be used as the p-value is less than 0.05. We can confirm this data by using the following texts

```r
library(nortest)
ad.test(energy)
```

```
##
##  Anderson-Darling normality test
##
## data:  energy
## A = 0.46126, p-value = 0.2563
```

```r
cvm.test(energy)
```

```
##
##  Cramer-von Mises normality test
##
## data:  energy
## W = 0.053358, p-value = 0.4607
```

```r
lillie.test(energy)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  energy
## D = 0.040216, p-value = 0.7325
```

```
pearson.test(energy)
```

```
##
##  Pearson chi-square normality test
##
## data:  energy
## P = 10.725, p-value = 0.6339
```

```
sf.test(energy)
```

```
##
##  Shapiro-Francia normality test
##
## data:  energy
## W = 0.98224, p-value = 0.03169
```

```
plot(normalfit)
```