About the chart - Histogram: distributes observations into bins along the x-axis; useful to visualize frequency and skewness.

Graphics environment setup.

# installation   
#install.packages("daltoolbox")  
  
# loading DAL  
library(daltoolbox)

library(ggplot2)  
library(RColorBrewer)  
  
# color palette  
colors <- brewer.pal(4, 'Set1')  
  
# setting the font size for all charts  
font <- theme(text = element\_text(size=16))

Generate variables with distinct distributions (exponential, uniform, normal).

# Exemplos com distribuições de dados  
# Usamos variáveis aleatórias para facilitar a visualização de diferentes distribuições.  
  
# example: dataset to be plotted   
example <- data.frame(exponential = rexp(100000, rate = 1),   
 uniform = runif(100000, min = 2.5, max = 3.5),   
 normal = rnorm(100000, mean=5))  
head(example)

## exponential uniform normal  
## 1 1.0704920 2.594031 4.798013  
## 2 1.8113712 2.669831 3.009860  
## 3 0.3842084 3.219861 5.790757  
## 4 1.1711771 2.982447 3.901110  
## 5 1.2598766 2.627410 6.064325  
## 6 1.5955480 2.768846 6.027973

# Histogram

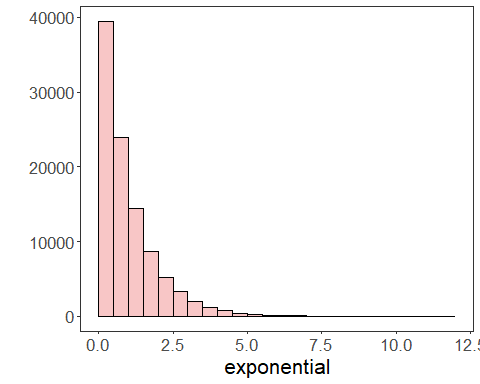
Visualize the distribution of a continuous variable by binning the x-axis and counting observations per bin. geom\_histogram() displays counts as bars. More info: ?geom\_histogram (R documentation)

Build histograms and arrange multiple charts side by side.

library(dplyr)  
  
grf <- plot\_hist(example |> dplyr::select(exponential),   
 label\_x = "exponential", color=colors[1]) + font

## Using as id variables

options(repr.plot.width=5, repr.plot.height=4)  
plot(grf)



# Chart arrangement

Use grid.arrange to place the generated charts side by side.

grfe <- plot\_hist(example |> dplyr::select(exponential),   
 label\_x = "exponential", color=colors[1]) + font

## Using as id variables

grfu <- plot\_hist(example |> dplyr::select(uniform),   
 label\_x = "uniform", color=colors[1]) + font

## Using as id variables

grfn <- plot\_hist(example |> dplyr::select(normal),   
 label\_x = "normal", color=colors[1]) + font

## Using as id variables

library(gridExtra)   
  
options(repr.plot.width=15, repr.plot.height=4)  
grid.arrange(grfe, grfu, grfn, ncol=3)

