Solutions_exercises

Giovanni Saraceno

Contents

Exercise 1: Probability calculation

```r
# (a) P(X > 3), where X follows N(0, 1)
1 - pnorm(3, mean = 0, sd = 1)
```

```
## [1] 0.001349898
```

```r
# (b) P(X > 42), where X follows N(35, 36)
1 - pnorm(42, mean = 35, sd = 6)
```

```
## [1] 0.1216725
```

```r
# (c) P(X = 10), where X follows Bin(10, 0.8)
dbinom(10, size = 10, prob = 0.8)
```

```
## [1] 0.1073742
```

```r
# (d) P(X < 0.9), where X follows N(0, 1)
pnorm(0.9, mean = 0, sd = 1)
```

```
## [1] 0.8159399
```

```r
# (e) P(X > 6.5), where X follows chi-squared with 2 dof
1 - pchisq(6.5, df = 2)
```

```
## [1] 0.03877421
```

Exercise 2: Normal Distribution Intervals

```r
# Calculate the z-values for 1%, 0.5%, and 0.1%
intervals <- c(0.01, 0.005, 0.001)
z_values <- qnorm(1 - intervals / 2) # Symmetric about the mean
z_values
```

```
## [1] 2.575829 2.807034 3.290527
```

```r
# Convert z-values to intervals in terms of standard deviations
intervals_in_s <- cbind(
  Lower = -z_values,
  Upper = z_values
)
intervals_in_s
```

```
##          Lower    Upper
## [1,] -2.575829 2.575829
## [2,] -2.807034 2.807034
## [3,] -3.290527 3.290527
```

Exercise 3: Success Probability with Binomial Distribution

```r
# Probability of 10 successes with p = 0.2
dbinom(10, size = 10, prob = 0.2)
```

```
## [1] 1.024e-07
```

Exercise 4: Simulating Coin Toss Using rbinom

```r
# Simulating 10 tosses of a fair coin using rbinom
set.seed(123)
# Output will be 0 for tails and 1 for heads
rbinom(n = 10, size = 1, prob = 0.5)
```

```
##  [1] 0 1 0 1 1 0 1 1 1 0
```

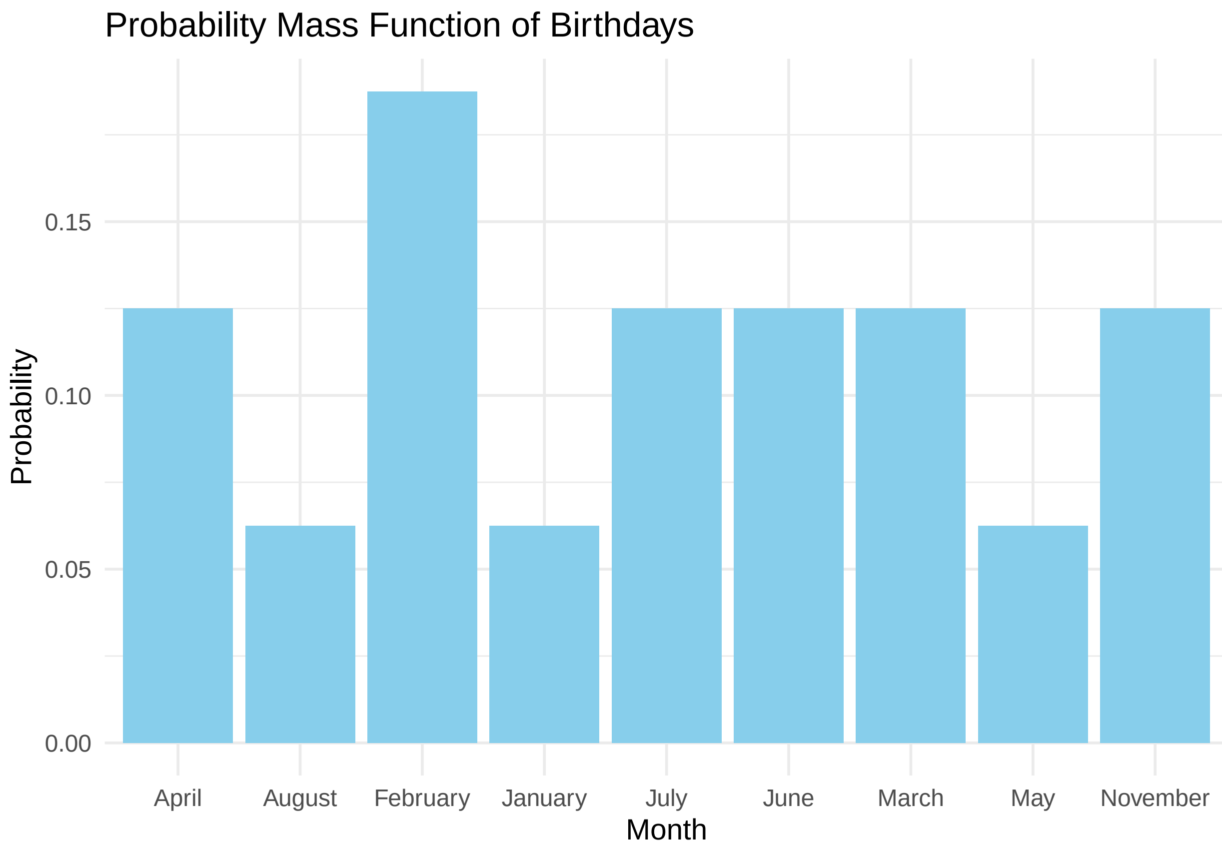
Exercise 5: Birthdays

```r
# Define the months and birthdays
birthdays <- c("January", "February", "February", "February", "March", "March",
               "April", "April", "May", "June", "June", "July", "July", "August",
               "November", "November")

# Create a frequency table
birthday_table <- table(birthdays)

# Convert to a probability table
birthday_prob <- prop.table(birthday_table)

# Plot the probability mass function
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.3.3
```

```r
birthday_df <- as.data.frame(birthday_prob)
names(birthday_df) <- c("Month", "Probability")

ggplot(birthday_df, aes(x = Month, y = Probability)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(
    title = "Probability Mass Function of Birthdays",
    x = "Month",
```

```
    y = "Probability"
  ) +
  theme_minimal()
```



- Input Data: Birthdays are grouped by their months.
- Frequency Calculation: A table is created to calculate the frequency of birthdays in each month.
- Probability Calculation: The frequencies are normalized to calculate probabilities.
- Visualization: A bar plot is generated using ggplot2 to show the probability mass function.

Exercise 6a: Skewness Calculation

```
skewness <- function(x, method = "g1") {
  N <- length(x)
  mean_x <- mean(x)
  sd_x <- sd(x) * sqrt((N - 1) / N)
  numerator <- sum((x - mean_x)^3) / N
  denominator <- sd_x^3

  if (method == "g1") {
    return(numerator / denominator)
  } else if (method == "G1") {
    adjusted_factor <- sqrt(N * (N - 1)) / (N - 2)
    return(adjusted_factor * numerator / denominator)
  } else {
    stop("Invalid method. Choose 'g1' or 'G1'.")
  }
```

```r
}

# Example usage
set.seed(123)
x <- rnorm(100)
skewness(x, method = "g1")
```

```
## [1] 0.06049948
```

```r
skewness(x, method = "G1")
```

```
## [1] 0.06142471
```

Exercise 6b: Kurtosis Calculation

```r
kurtosis <- function(x, type = "k1") {
  N <- length(x)
  mean_x <- mean(x)
  sd_x <- sd(x) * sqrt((N - 1) / N) # Use N for consistency with definition
  numerator <- sum((x - mean_x)^4) / N
  denominator <- sd_x^4

  if (type == "k1") {
    return(numerator / denominator)
  } else if (type == "k2") {
    return(numerator / denominator - 3)
  } else {
    stop("Invalid type. Choose 'k1' or 'k2'.")
  }
}

# Example usage
set.seed(123)
x <- rnorm(100)
kurtosis(x, type = "k1")
```

```
## [1] 2.838947
```

```r
kurtosis(x, type = "k2")
```

```
## [1] -0.161053
```

Exercise 7: Analyze and Compare Fund Types

```r
# Reading data from a text file
fondi <- read.table("fondi.txt", header = TRUE, sep=",")
summary(fondi)
```

```
##     GruppoA          GruppoB
##  Min.   :1.546   Min.   :0.8157
##  1st Qu.:1.851   1st Qu.:1.6927
##  Median :1.982   Median :2.0192
##  Mean   :2.033   Mean   :2.0572
##  3rd Qu.:2.193   3rd Qu.:2.4085
##  Max.   :2.636   Max.   :3.4349
```

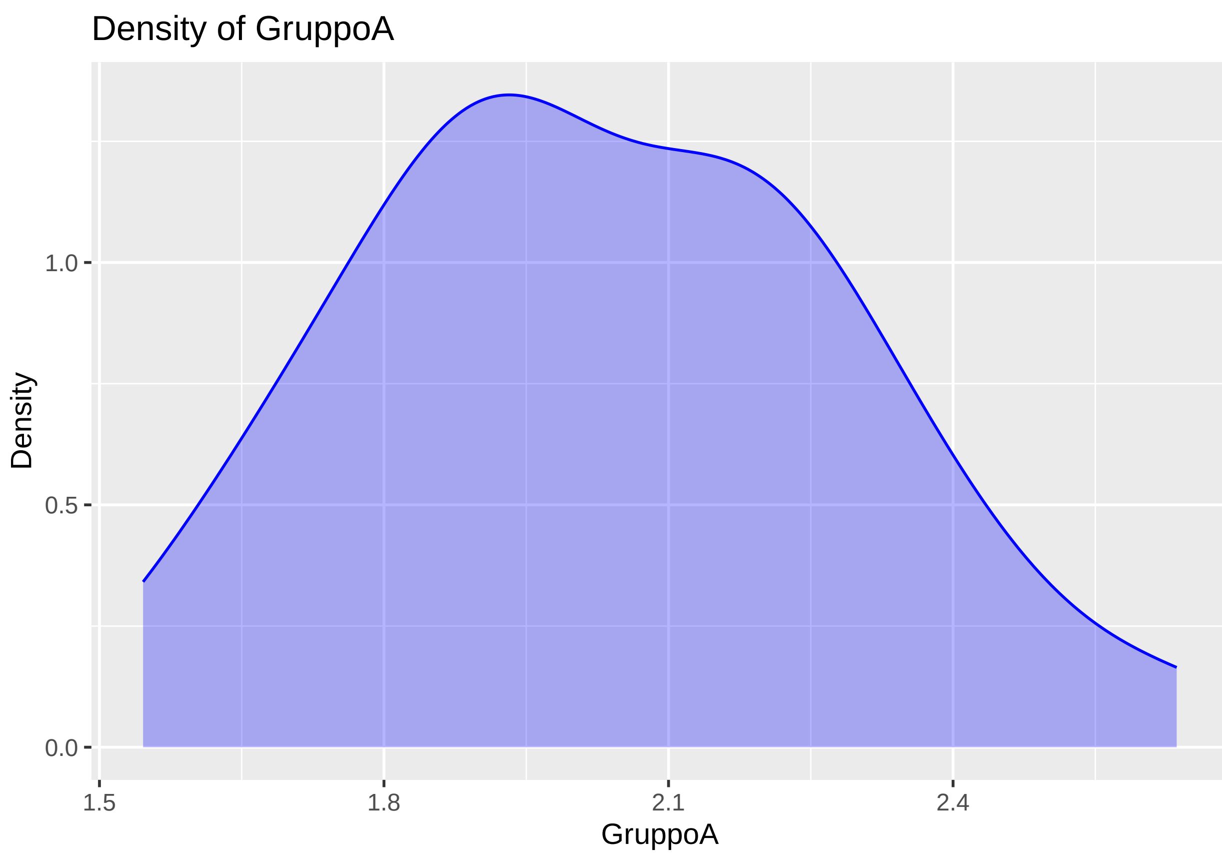
```r
# Calculating summary statistics
library(dplyr)
```

```
##
## Caricamento pacchetto: 'dplyr'

## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag

## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

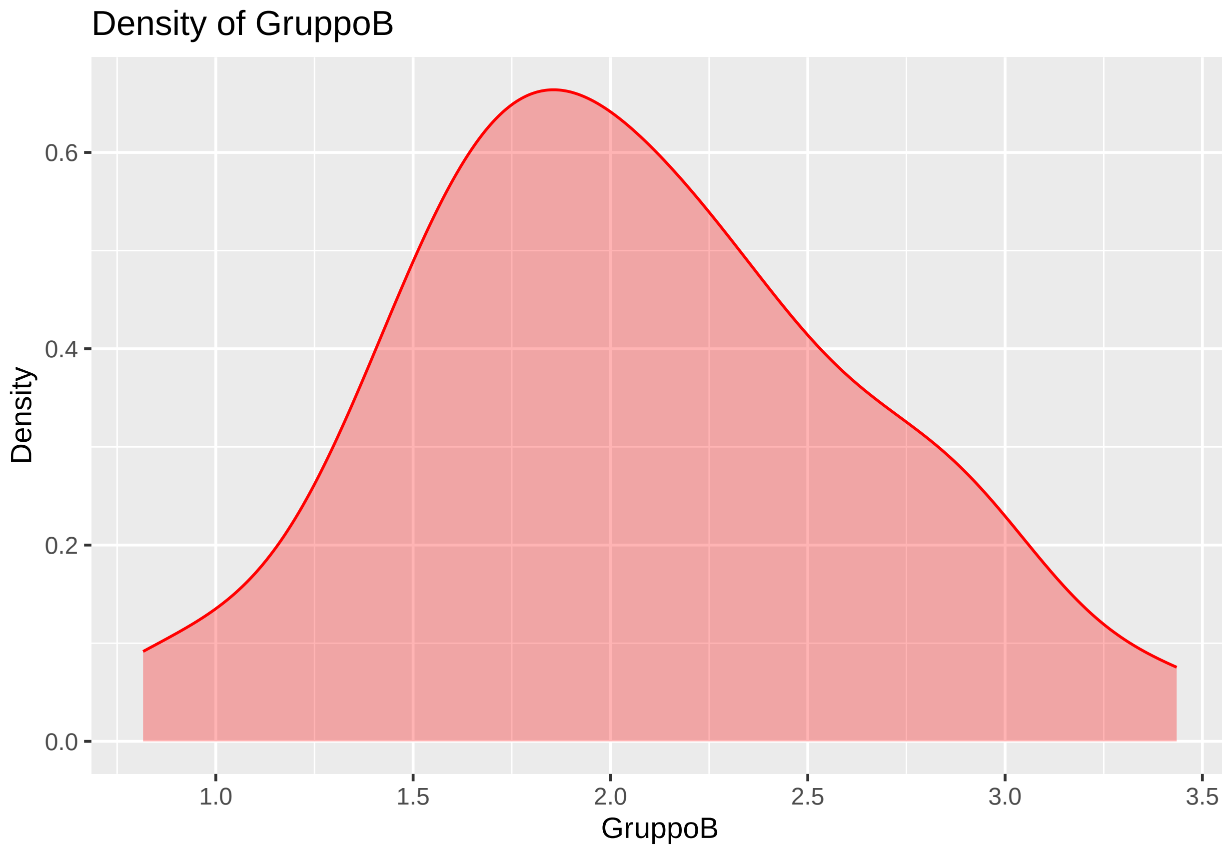
```r
fondi_summary <- fondi %>%
  summarise(
    A_min = min(GruppoA), A_Q1 = quantile(GruppoA, 0.25), A_mean = mean(GruppoA),
    A_median = median(GruppoA), A_Q3 = quantile(GruppoA, 0.75), A_max = max(GruppoA),
    A_sd = sd(GruppoA), A_IQR = IQR(GruppoA),
    B_min = min(GruppoB), B_Q1 = quantile(GruppoB, 0.25), B_mean = mean(GruppoB),
    B_median = median(GruppoB), B_Q3 = quantile(GruppoB, 0.75), B_max = max(GruppoB),
    B_sd = sd(GruppoB), B_IQR = IQR(GruppoB)
  ) %>%
  round(2)

fondi_summary
```

```
##   A_min A_Q1 A_mean A_median A_Q3 A_max A_sd A_IQR B_min B_Q1 B_mean B_median
## 1  1.55 1.85   2.03     1.98 2.19  2.64 0.26  0.34  0.82 1.69   2.06     2.02
##   B_Q3 B_max B_sd B_IQR
## 1 2.41  3.43 0.59  0.72
```

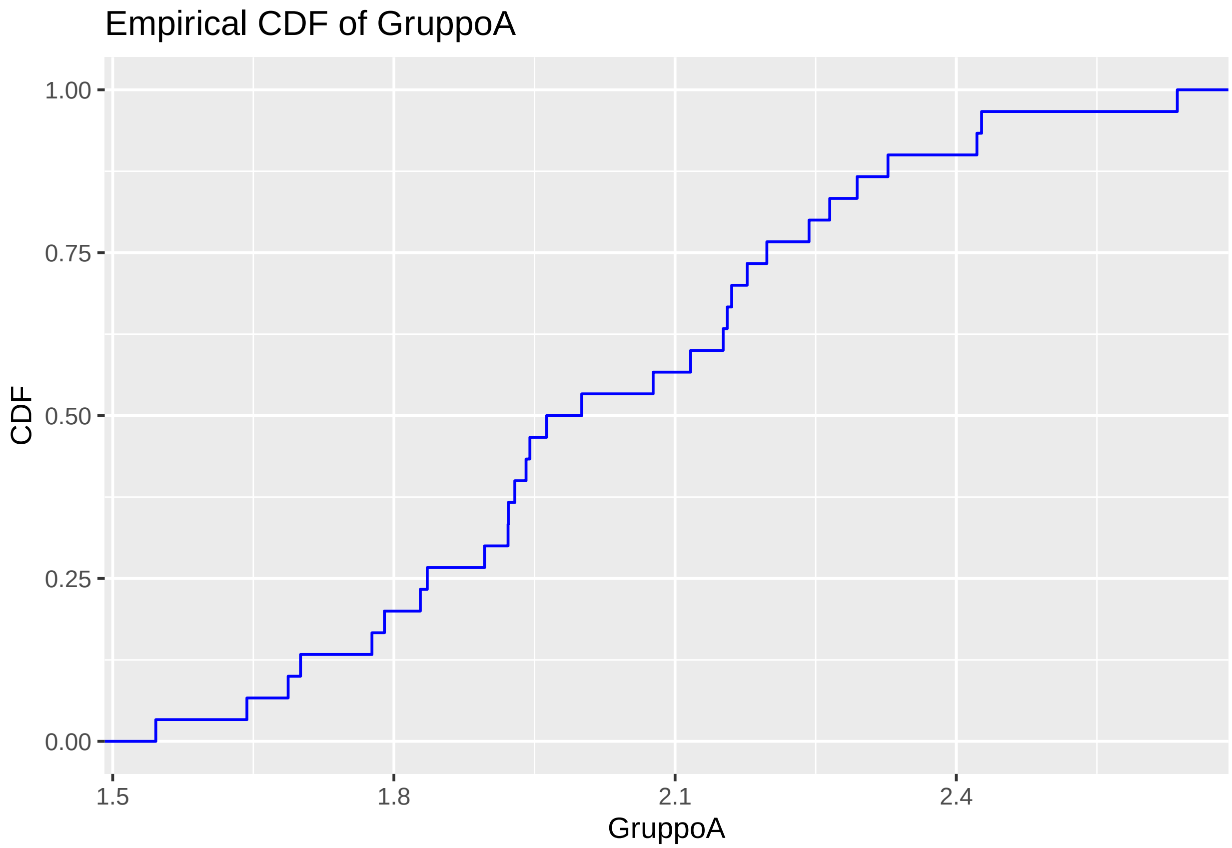
```r
# Density and ECDF plots
library(ggplot2)
ggplot(fondi, aes(x = GruppoA)) +
  geom_density(color = "blue", fill = "blue", alpha = 0.3) +
  labs(title = "Density of GruppoA", x = "GruppoA", y = "Density")
```



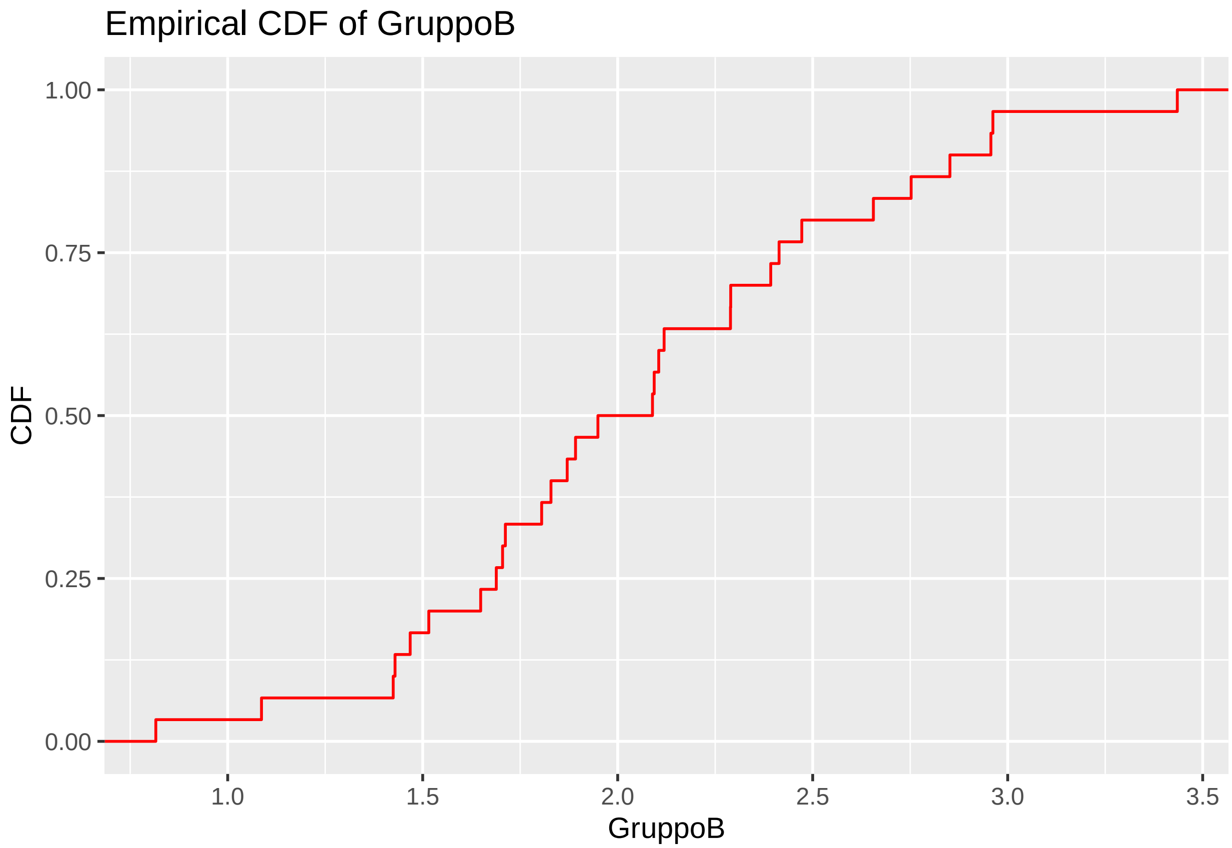

```r
ggplot(fondi, aes(x = GruppoB)) +
  geom_density(color = "red", fill = "red", alpha = 0.3) +
  labs(title = "Density of GruppoB", x = "GruppoB", y = "Density")
```




```r
ggplot(fondi, aes(x = GruppoA)) +
  stat_ecdf(geom = "step", color = "blue") +
  labs(title = "Empirical CDF of GruppoA", x = "GruppoA", y = "CDF")
```




```r
ggplot(fondi, aes(x = GruppoB)) +
  stat_ecdf(geom = "step", color = "red") +
  labs(title = "Empirical CDF of GruppoB", x = "GruppoB", y = "CDF")
```




```r
# Skewness and Kurtosis
library(moments)
```

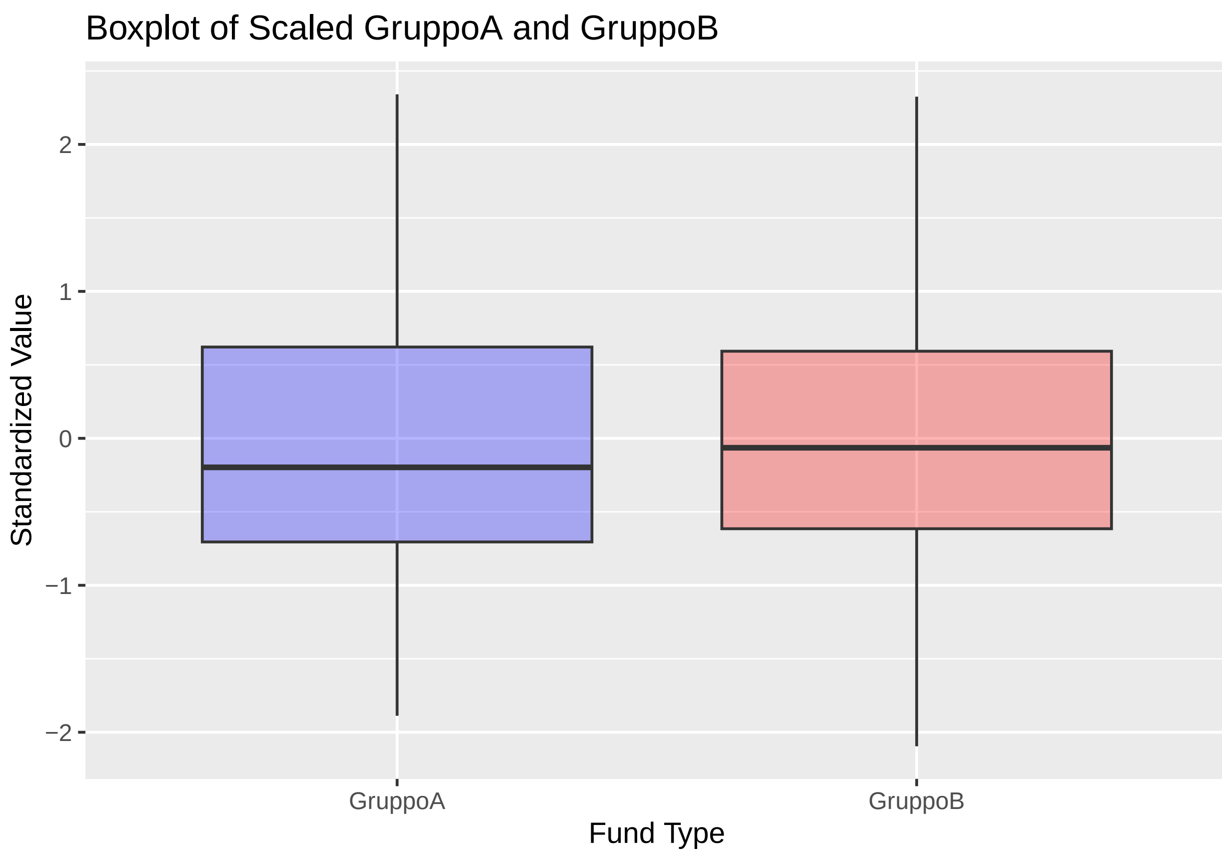
```
##
## Caricamento pacchetto: 'moments'

## I seguenti oggetti sono mascherati _da_ '.GlobalEnv':
##
##     kurtosis, skewness
```

```r
A_skewness <- skewness(fondi$GruppoA)
A_kurtosis <- kurtosis(fondi$GruppoA)
B_skewness <- skewness(fondi$GruppoB)
B_kurtosis <- kurtosis(fondi$GruppoB)

data.frame(
  Variable = c("GruppoA", "GruppoB"),
  Skewness = c(A_skewness, B_skewness),
  Kurtosis = c(A_kurtosis, B_kurtosis)
)
```

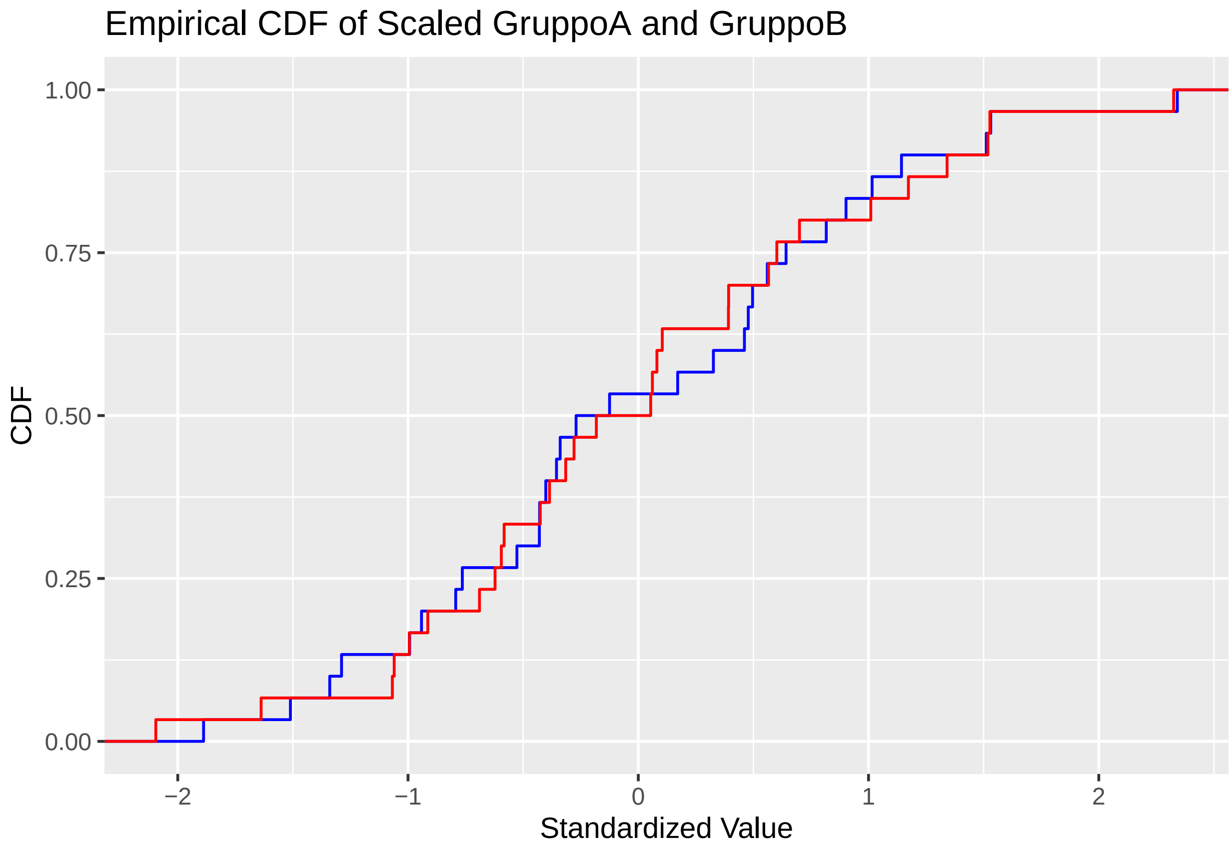
```
##   Variable  Skewness Kurtosis
## 1  GruppoA 0.2294457 2.556130
## 2  GruppoB 0.2325419 2.784963
```

```r
# Standardize the data
fondi_scaled <- fondi %>%
  mutate(A_scaled = scale(GruppoA), B_scaled = scale(GruppoB))
```

```r
# Comparing using Boxplot
ggplot(fondi_scaled) +
  geom_boxplot(aes(x = "GruppoA", y = A_scaled), fill = "blue", alpha = 0.3) +
  geom_boxplot(aes(x = "GruppoB", y = B_scaled), fill = "red", alpha = 0.3) +
  labs(title = "Boxplot of Scaled GruppoA and GruppoB", x = "Fund Type", y = "Standardized Value")
```




```r
# Comparing using ECDF
ggplot() +
  stat_ecdf(data = fondi_scaled, aes(x = A_scaled), geom = "step", color = "blue") +
  stat_ecdf(data = fondi_scaled, aes(x = B_scaled), geom = "step", color = "red") +
  labs(title = "Empirical CDF of Scaled GruppoA and GruppoB", x = "Standardized Value", y = "CDF")
```



Empirical CDF of Scaled GruppoA and GruppoB
CDF
1.00
0.75
0.50
0.25
0.00
−2
−1
0
1
2
Standardized Value