Estatística Computacional II

Trabalho 1 - Extremos

Luiz Francisco - GRR20213026

Mateus Souza - GRR20207154

outubro/2023

1. Extremos

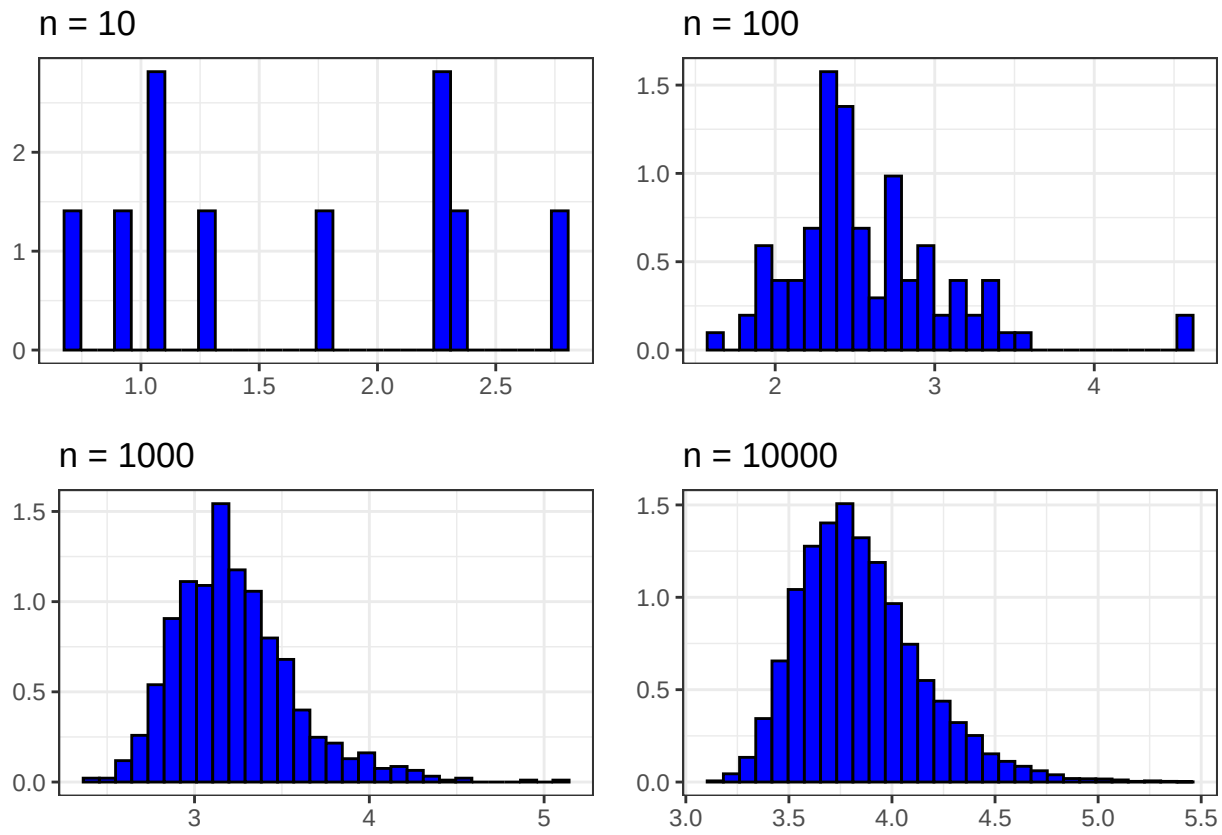
Seja uma amostra aleatória $(y_1, y_2, ..., y_n)$ de uma v.a. Y. Sabemos que para amostras *suficientemente* grandes e com algumas condições de regularidade, a distribuição amostral da média $\overline{Y}$ converge para uma distribuição normal, indiferentemente da distribuição $F_Y(y)$ da variável Y. Mas o que acontece com o máximo $Y_M = max(y_1, y_2, y_n)$? Estudar o comportamento do máximo de uma variável pode ser útil em diversos contextos como ambientais (precipitação, temperatura, poluição), financeiros, etc. Esse tema é tratado pela aréa da *teoria dos valores extremos*. No presente trabalho, serão realizadas explorações computacionais desses conceitos.

Seja $Y \sim N(0, 1)$.

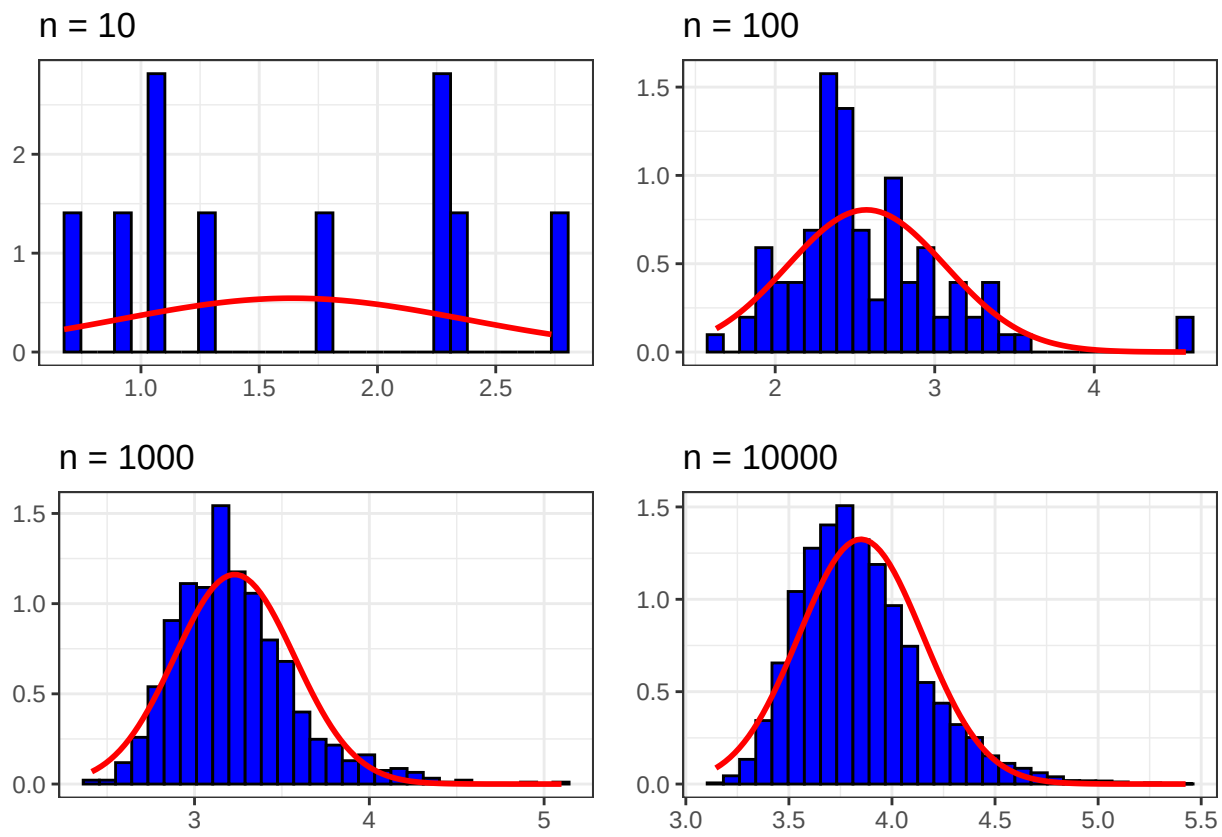
1. Tome diversas amostras para cada tamanho $n = 10$, $n = 100$, $n = 1000$, $n = 10000$ e verifique o comportamento das ditribuições amostrais. As distribuições amostrais parecem normais? Se aproximam da normalidade para valores crescentes de n? Discuta os resultados.

Para esse exercício precisamos simular diversas amostras e então verificar a distribuição amostral do valor máximo de cada amostra. Utilizaremos aqui a função *rnorm* para simular as amostras de uma distribuição normal e os pacotes *ggplot2* e *gridExtra* para mostrar visualmente os resultados.

```
### Carregando pacotes necessários
library(ggplot2)
library(gridExtra)

### Definindo a semente para reprodução de resultados
set.seed(2023)

### Criação das amostras
y_10 <- data.frame(x=replicate(10, max(rnorm(10))))
y_100 <- data.frame(x=replicate(100, max(rnorm(100))))
y_1000 <- data.frame(x=replicate(1000, max(rnorm(1000))))
y_10000 <- data.frame(x=replicate(10000, max(rnorm(10000))))

### Verificando como são as distribuições amostrais
d_10 <- ggplot(y_10)+
  geom_histogram(aes(x = x, y = after_stat(density)), color = "black", fill = "blue")+
  labs(x = element_blank(), y = element_blank(), title = "n = 10")+
  theme_bw()
d_100 <- ggplot(y_100)+
  geom_histogram(aes(x = x, y = after_stat(density)), color = "black", fill = "blue")+
  labs(x = element_blank(), y = element_blank(), title = "n = 100")+
  theme_bw()
d_1000 <- ggplot(y_1000)+
  geom_histogram(aes(x = x, y = after_stat(density)), color = "black", fill = "blue")+
  labs(x = element_blank(), y = element_blank(), title = "n = 1000")+
  theme_bw()
d_10000 <- ggplot(y_10000)+
  geom_histogram(aes(x = x, y = after_stat(density)), color = "black", fill = "blue")+
  labs(x = element_blank(), y = element_blank(), title = "n = 10000")+
  theme_bw()

### Exibindo os 4 gráficos na mesma tela
grid.arrange(d_10, d_100, d_1000, d_10000, ncol = 2)
```




```
### Verificando normalidade via gráficos
g_10 <- d_10 +
  stat_function(fun = dnorm, args = list(mean = mean(y_10$x), sd = sd(y_10$x)), color = "red", size = 1
g_100 <- d_100 +
  stat_function(fun = dnorm, args = list(mean = mean(y_100$x), sd = sd(y_100$x)), color = "red", size =
g_1000 <- d_1000 +
  stat_function(fun = dnorm, args = list(mean = mean(y_1000$x), sd = sd(y_1000$x)), color = "red", size
g_10000 <- d_10000 +
  stat_function(fun = dnorm, args = list(mean = mean(y_10000$x), sd = sd(y_10000$x)), color = "red", si

### Exibindo os 4 gráficos na mesma tela
grid.arrange(g_10, g_100, g_1000, g_10000, ncol = 2)
```




```r
# Função para verificar normalidade e retornar a conclusão com p-valor
normalidade <- function(data, alpha = 0.05) {
  ks_result <- ks.test(data, "pnorm")
  p_value <- ks_result$p.value
  if (p_value < alpha) { return(paste("Amostra não normal (p-valor =", p_value, ")"))}
  else { return(paste("Amostra normal (p-valor =", p_value, ")"))}}

# Aplicando a função para diferentes tamanhos de amostra
ks_10 <- normalidade(y_10)
ks_100 <- normalidade(y_100)
ks_1000 <- normalidade(y_1000)
ks_10000 <- normalidade(y_10000)

# Imprimindo os resultados
cat(" n = 10:", ks_10, "\n",
    "n = 100:", ks_100, "\n",
    "n = 1000:", ks_1000, "\n",
    "n = 10000:", ks_10000)
```

```
##  n = 10: Amostra não normal (p-valor = 2.46716962293281e-06 )
##  n = 100: Amostra não normal (p-valor = 0 )
##  n = 1000: Amostra não normal (p-valor = 0 )
##  n = 10000: Amostra não normal (p-valor = 0 )
```

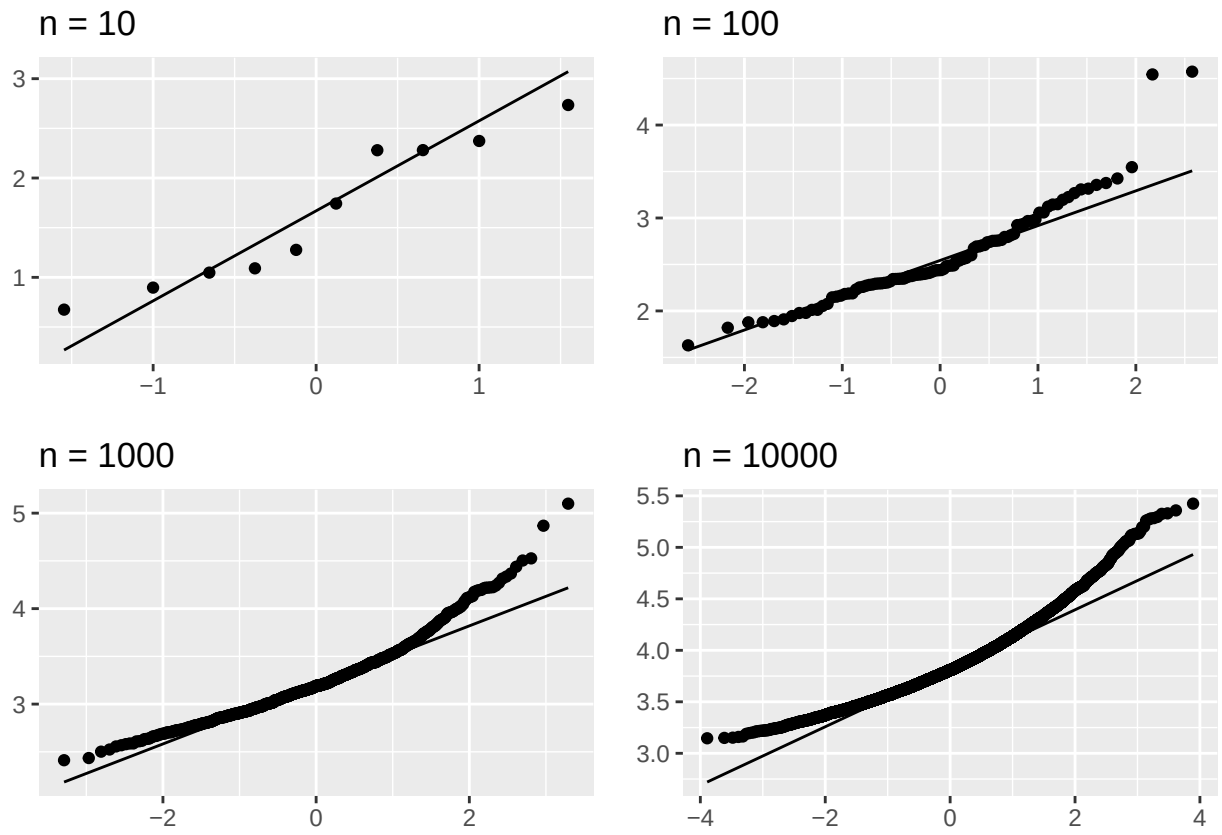
Ao observarmos a distribuição amostral por si só, é possível notar que ela não se parece com uma normal, porém aparenta se aproximar um pouco mais conforme o aumento de n. Quando adicionamos a curva da

normal, é notavel que há diversas fugas do padrão, mesmo com o aumento de n. Aplicou-se também o teste Kolmogorov-Smirnov para normalidade, o qual nos retornou a rejeição da hipótese de que a $Y_M \sim N(\mu, \sigma)$, mesmo a nível de 1% para todas os tamanhos de n.
Então, temos de conclusão que, com bases em testes e simulações computacionais, a distribuição do máximo de uma amostra normal, não segue uma distribuição normal, como a média segue, por exemplo.

2. Investigue o comportamento das distribuições amostrais utilizando qq-plots (qqnorm). Para isso voce pode usar função pronta do R (qqnorm(), por exemplo) ou se basear no seguinte. Sejam as estimativas das diversas amostras para um determinado tamanho ordenada em (y(1), ..., y(n)). Os quantis esperados correspondentes a cada dado são obtidos por $G^{-1}(\frac{(i-0.5)}{n})$ em que G é a distribuição contra a qual deseja-se confrontar os dados. Note que o intercepto e inclinação do gráfico de quantis fornece estimativas de média e desvio padrão.

O gráfico quantil-quantil é um gráfico que compara os quantis teóricos da distribuição teórica suposta e os compara com os valores observados, em geral ele é usado em conjunto com o teste Kolmogorov-Smirnov, por exemplo, para verificar se nossos dados seguem aquela distribuição em específico. Para isso o R tem sua função padrão, mas aqui iremos utilizar a visulização do *ggplot*.

```
### Criando os gráficos qqplot
q_10 <- ggplot(y_10, aes(sample = x))+
  geom_qq()+
  geom_qq_line()+
  labs(x = element_blank(), y = element_blank(), title = "n = 10")
q_100 <- ggplot(y_100, aes(sample = x))+
  geom_qq()+
  geom_qq_line()+
  labs(x = element_blank(), y = element_blank(), title = "n = 100")
q_1000 <- ggplot(y_1000, aes(sample = x))+
  geom_qq()+
  geom_qq_line()+
  labs(x = element_blank(), y = element_blank(), title = "n = 1000")
q_10000 <- ggplot(y_10000, aes(sample = x))+
  geom_qq()+
  geom_qq_line()+
  labs(x = element_blank(), y = element_blank(), title = "n = 10000")

### Exibindo os 4 gráficos na mesma tela
grid.arrange(q_10, q_100, q_1000, q_10000)
```



Aqui é nítido a falta de ajuste em relação à distribuição normal, podemos perceber as fugas sistemáticas dos dados comparados aos quantis teóricos, e ao aumentar o tamanho de amostra essa fica cada vez mais perceptível essa fuga. Em geral nossos dados estão sendo subestimados pelo ajuste. Isso então só confirma o que vimos anteriormente com o teste Kolmogorov-Smirnov e com o histograma dos dados.

Nos itens anteriores deve-se obter que a distribuição amostral de Y_M não é normal. Uma aproximação melhor é dada pela distribuição de *Gumbel*:

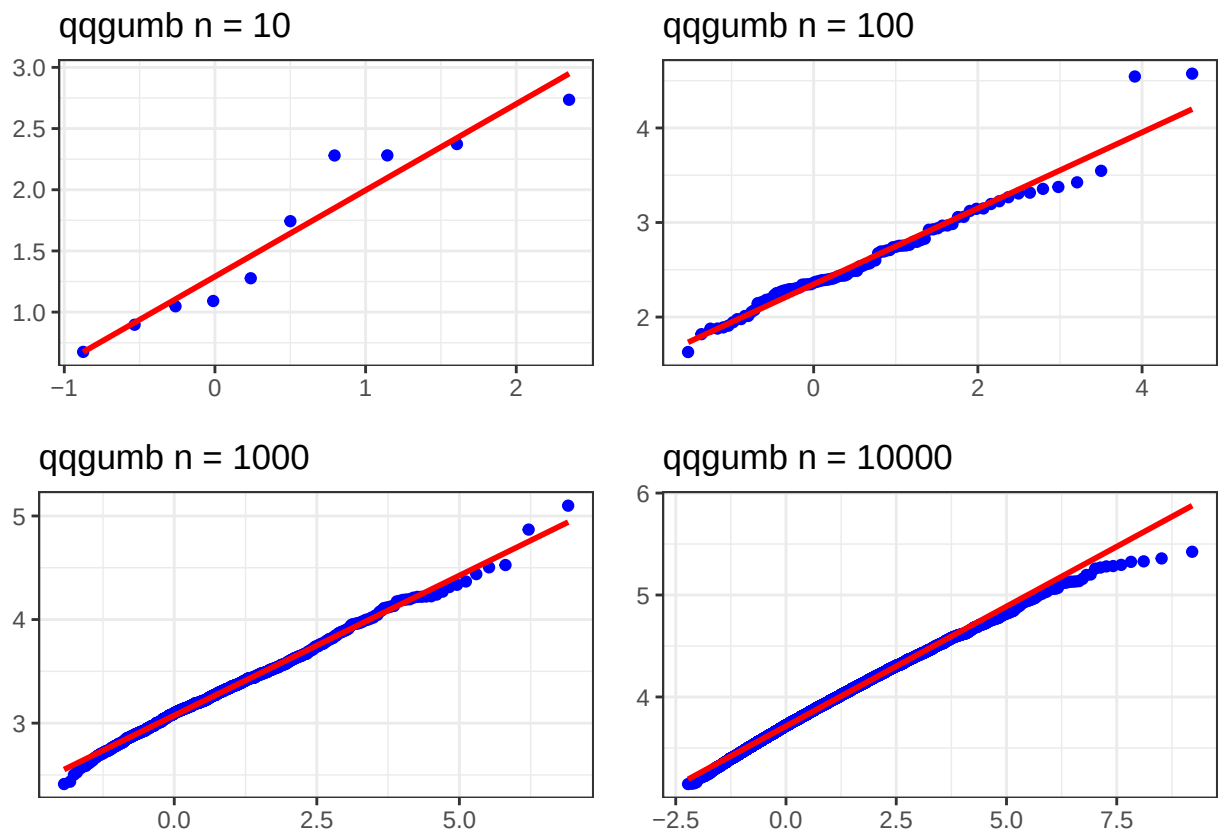
$$F_G(t;\mu,\sigma) = exp\{-exp\{-\frac{(t-\mu)}{\sigma}\}\},$$

em que μ e σ são parâmetros de locação e escala análogos à média e desvio padrão da normal. Avalie esta opção de distribuição amostral. No gráfico de quantis utilize $F_G^{-1}(\frac{i}{n+1})$. Note que o intercepto e a inclinação do gráfico de quantis fornecem estimativas dos parâmetros μ e σ de forma análoga ao caso normal.
Ilustre com simulações que a distribuição de *Gumbel* se aproxima melhor como distribuição amostral, embora possa ter problemas para capturar valores extremos para grandes tamanhos de amostra.

Aqui para ilustramos o *qqplot* dos dados em relação à distribuição *Gumbel*. Para isso iremos utilizar a inversa oferecia do enunciado para definirmos os quantis da distribuição e então usar uma regressão linear para aplicar a reta identidade. Aqui também ao invés de utilizarmos o *plot* padrão do R, usaremos o *ggplot*. Além disso, para encontrar os quantis da distribuição será utilizado a função *qgumbel* do pacote *extraDistr*.

```
### Carregando pacote para distribuição de Gumbel
library(extraDistr)

### Criando data frames dos dados com quantil x dados
df_10 <- data.frame(q = qgumbel(((1:10)/(10+1))), y = sort(y_10$x))
df_100 <- data.frame(q = qgumbel(((1:100)/(100+1))), y = sort(y_100$x))
df_1000 <- data.frame(q = qgumbel(((1:1000)/(1000+1))), y = sort(y_1000$x))
```

```r
df_10000 <- data.frame(q = qgumbel(((1:10000)/(10000+1))), y = sort(y_10000$x))

### Criando os qqplots dos dados em relação aos quantis da Gumbel teórica
qg_10 <- ggplot(df_10, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb n = 10")+
  theme_bw()
qg_100 <- ggplot(df_100, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb n = 100")+
  theme_bw()
qg_1000 <- ggplot(df_1000, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb n = 1000")+
  theme_bw()
qg_10000 <- ggplot(df_10000, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb n = 10000")+
  theme_bw()

### Exibindo os 4 gráficos na mesma tela

grid.arrange(qg_10, qg_100, qg_1000, qg_10000)
```



Aqui vemos que a distribuição de *Gumbel* é uma boa opção para podermos modelar a distribuição do máximo de variáveis oriundas de uma Normal. Porém, como dito no enunciado, para grandes amostras a distribuição começa ter certa dificuldade em captação de valores mais extremos.

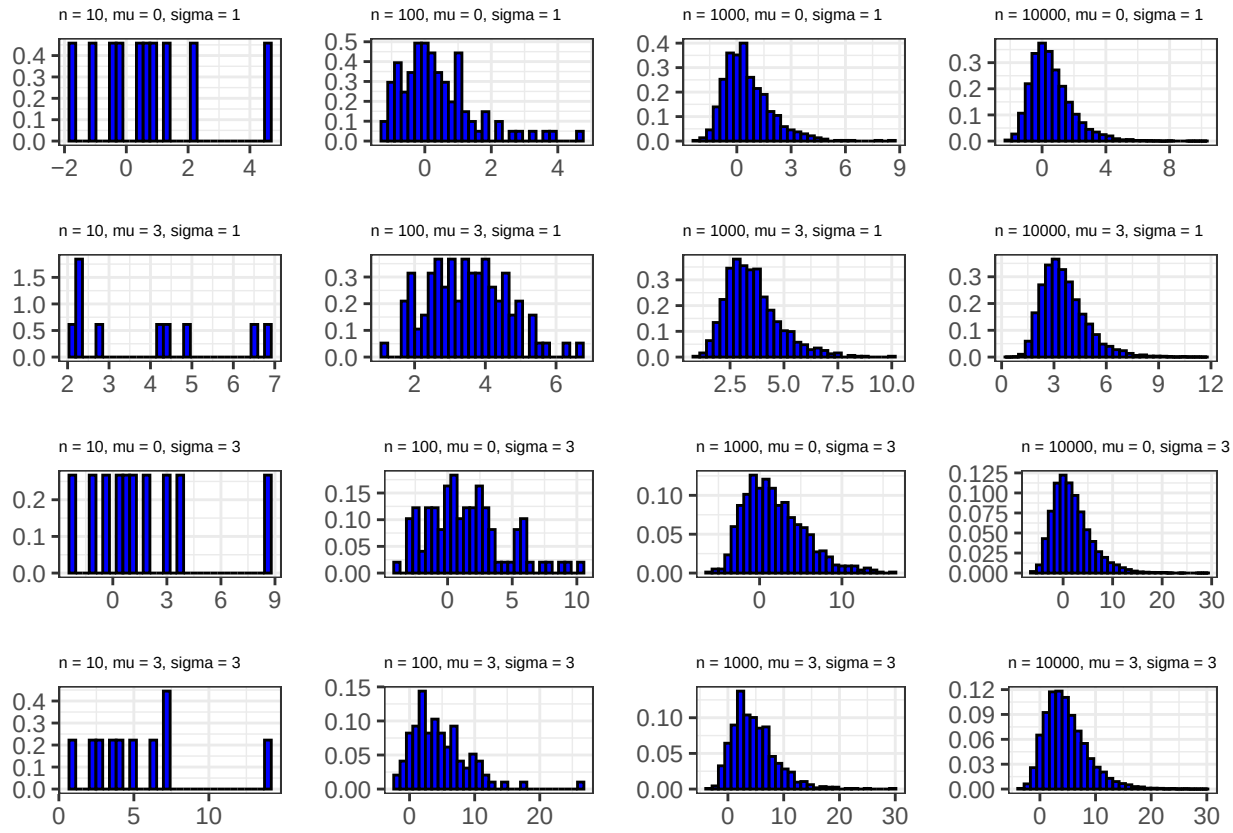
Explore ainda:

- Verificar como μ e σ variam em função do tamanho de amostra n.

Aqui iremos simular para os mesmos valores de n das questões anteriores, além de modificar indidivualmente cada parâmetro e depois conjuntamente para ver como a distribuição muda. Aqui utilizaremos a função *rgumbel* do pacote importado.

```
### Simulando valores com n = 10, 100, 1000 e 10000 e para mu = 0 e 3 e sigma = 1 e 3
g_10_0_1 <- data.frame(x = rgumbel(10))
g_100_0_1 <- data.frame(x = rgumbel(100))
g_1000_0_1 <- data.frame(x = rgumbel(1000))
g_10000_0_1 <- data.frame(x = rgumbel(10000))
g_10_3_1 <- data.frame(x = rgumbel(10, 3, 1))
g_100_3_1 <- data.frame(x = rgumbel(100, 3, 1))
g_1000_3_1 <- data.frame(x = rgumbel(1000, 3, 1))
g_10000_3_1 <- data.frame(x = rgumbel(10000, 3, 1))
g_10_0_3 <- data.frame(x = rgumbel(10, 0, 3))
g_100_0_3 <- data.frame(x = rgumbel(100, 0, 3))
g_1000_0_3 <- data.frame(x = rgumbel(1000, 0, 3))
g_10000_0_3 <- data.frame(x = rgumbel(10000, 0, 3))
g_10_3_3 <- data.frame(x = rgumbel(10, 3, 3))
g_100_3_3 <- data.frame(x = rgumbel(100, 3, 3))
```

```r
g_1000_3_3 <- data.frame(x = rgumbel(1000, 3, 3))
g_10000_3_3 <- data.frame(x = rgumbel(10000, 3, 3))

# Função para criar gráficos
create_histogram_plot <- function(data, title) {
  ggplot(data) +
    geom_histogram(aes(x = x, y = after_stat(density)), color = "black", fill = "blue") +
    labs(x = element_blank(), y = element_blank(), title = title) +
    theme_bw() +
    theme(plot.title = element_text(size = 6))  # Altere o tamanho da fonte aqui
}

# Criando os gráficos com diferentes configurações
h_10_0_1 <- create_histogram_plot(g_10_0_1, "n = 10, mu = 0, sigma = 1")
h_100_0_1 <- create_histogram_plot(g_100_0_1, "n = 100, mu = 0, sigma = 1")
h_1000_0_1 <- create_histogram_plot(g_1000_0_1, "n = 1000, mu = 0, sigma = 1")
h_10000_0_1 <- create_histogram_plot(g_10000_0_1, "n = 10000, mu = 0, sigma = 1")
h_10_3_1 <- create_histogram_plot(g_10_3_1, "n = 10, mu = 3, sigma = 1")
h_100_3_1 <- create_histogram_plot(g_100_3_1, "n = 100, mu = 3, sigma = 1")
h_1000_3_1 <- create_histogram_plot(g_1000_3_1, "n = 1000, mu = 3, sigma = 1")
h_10000_3_1 <- create_histogram_plot(g_10000_3_1, "n = 10000, mu = 3, sigma = 1")
h_10_0_3 <- create_histogram_plot(g_10_0_3, "n = 10, mu = 0, sigma = 3")
h_100_0_3 <- create_histogram_plot(g_100_0_3, "n = 100, mu = 0, sigma = 3")
h_1000_0_3 <- create_histogram_plot(g_1000_0_3, "n = 1000, mu = 0, sigma = 3")
h_10000_0_3 <- create_histogram_plot(g_10000_0_3, "n = 10000, mu = 0, sigma = 3")
h_10_3_3 <- create_histogram_plot(g_10_3_3, "n = 10, mu = 3, sigma = 3")
h_100_3_3 <- create_histogram_plot(g_100_3_3, "n = 100, mu = 3, sigma = 3")
h_1000_3_3 <- create_histogram_plot(g_1000_3_3, "n = 1000, mu = 3, sigma = 3")
h_10000_3_3 <- create_histogram_plot(g_10000_3_3, "n = 10000, mu = 3, sigma = 3")

# Exibir gráficos
grid.arrange(h_10_0_1, h_100_0_1, h_1000_0_1, h_10000_0_1,
             h_10_3_1, h_100_3_1, h_1000_3_1, h_10000_3_1,
             h_10_0_3, h_100_0_3, h_1000_0_3, h_10000_0_3,
             h_10_3_3, h_100_3_3, h_1000_3_3, h_10000_3_3, ncol = 4)
```



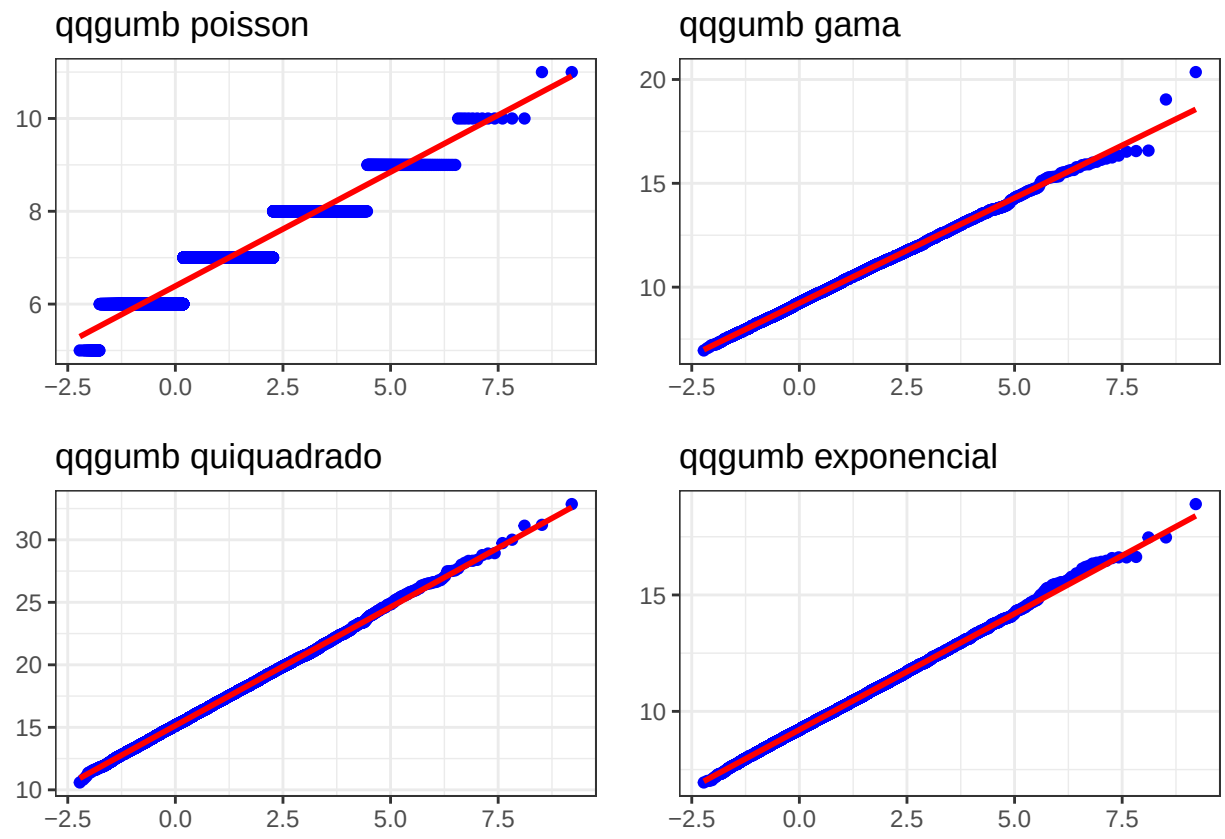
Com a visualização dos gráficos das diferentes *Gumbels* é possível entender porque os parâmetros se chamam de posição e escala, pois com a mudança dos parâmetros apenas é mudado aonde a curva é alocada e qual a escala do eixo x, mantendo sempre a mesma forma.

- A aproximação pela *Gumbel* vale para dados provenientes de outra distribuição que não a Normal?

Para verificar a suposição de que a aproximação da distribuição do máximo pela *Gumbel* é valida para outras distribuições akém da normal, iremos aqui comparar com outras 4 distribuições: *Poisson*, *Gama*, *Qui-Quadrado* e *Exponencial*, todas elas com n = 10000. Aqui iremos também visualizar como anteriormente com o gráfico gerado pelo *ggplot2* e usaremos as funções de simulação de dados para cada distribuição.

```
### Simulado dados do máximo para poisson, gama, qui-quadrado e beta
poisson <- data.frame(x = replicate(10000, max(rpois(10000, 1))))
gama <- data.frame(x = replicate(10000, max(rgamma(10000, 1, 1))))
quiquadrado <- data.frame(x = replicate(10000, max(rchisq(10000, 1))))
exponencial <- data.frame(x = replicate(10000, max(rexp(10000, 1))))

### Criando data frames para montar os qqplots
df_poisson <- data.frame(q = qgumbel(((1:10000)/(10000+1))), y = sort(poisson$x))
df_gama <- data.frame(q = qgumbel(((1:10000)/(10000+1))), y = sort(gama$x))
df_quiquadrado <- data.frame(q = qgumbel(((1:10000)/(10000+1))), y = sort(quiquadrado$x))
df_exponencial <- data.frame(q = qgumbel(((1:10000)/(10000+1))), y = sort(exponencial$x))

### Montando gráficos qq-plot
q_poisson <- ggplot(df_poisson, aes(x = q, y = y))+
```

```
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb poisson")+
  theme_bw()
q_gama <- ggplot(df_gama, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb gama")+
  theme_bw()
q_quiquadrado <- ggplot(df_quiquadrado, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb quiquadrado")+
  theme_bw()
q_exponencial <- ggplot(df_exponencial, aes(x = q, y = y))+
  geom_point(color = "blue")+
  geom_smooth(method = "lm", se = FALSE, color = "red")+
  labs(x = element_blank(), y = element_blank(), title = "qqgumb exponencial")+
  theme_bw()

### Exibindo os 4 gráficos na mesma tela
grid.arrange(q_poisson, q_gama, q_quiquadrado, q_exponencial, ncol = 2)
```



Com esses resultados é possível notar que a aproximação da distribuição do máximo pela *Gumbel* funciona para outras distribuições além da Normal, incluindo para discretas, como o caso da *Poisson* que também foi bem aproximada. Porém isso só é válido para distribuições que possuem o suporte com limite superior

infinito, pois a distribuição do máximo para distribuições limitadas superiormente será esse o valor limitante, como o caso da *Beta*, por exemplo, que é limitada em 1. Também existe o caso da distribuição do mínimo, que também pode ser aproximada pela Gumbel, apenas invertendo alguns sinais da densidade dela, nesse caso para a distribuição do mínimo temos que a aproximação vale para casos ontem não há limite inferior.